

27
2j.



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

DESARROLLO DE UN SISTEMA INTEGRAL DE
INFORMACION DE TESORERIA PARA
FERROCARRILES NACIONALES DE MEXICO

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A N :
FELIPE CRUZ CHAVEZ
LOURDES NIETO AGUILAR



DIRECTOR DE TESIS: ING. MARIA JAQUELINA LOPEZ BARRIENTOS

MEXICO, D. F.

1997

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.

Los sueños son la esperanza perenne de nuestra vida y la energía que nos hace vivir; ¡hay que aferrarse a ellos!, ¡podemos lograrlo!, más aun, nosotros, los mexicanos, tenemos sobradas razones para enorgullecernos de nuestra nacionalidad.

Somos un pueblo que tiene una historia y una cultura ancestrales.

Tenemos también una tradición y una historia de valores humanos.

Actualmente se ha olvidado lo que fuimos, y no proyectamos tampoco lo que podemos ser.

Usando nuestra razón e inteligencia, y aprovechando nuestros recursos intelectuales y humanos, podemos llegar a ser una nación culta y consciente de su responsabilidad en el mundo.

A DIOS:

Por permitirme llegar hasta este momento.

DE MANERA MUY ESPECIAL:

A mis padres

Por esa vida de lucha, sacrificio y esfuerzo constante, por ese apoyo moral y estímulos brindados con infinito amor y confianza. Gracias.

A MIS COMPAÑEROS DE TESIS:

Lourdes Nieto, Miguel Hernández, Mauricio González, Oscar Hernández por su apoyo y paciencia para terminar ésta tesis.

A TODAS AQUELLAS PERSONAS:

Que de una u otra manera formaron parte de este proyecto. Gracias.

FELIPE CRUZ CHÁVEZ.

A MIS PADRES:

Conchita y Felipe

Por todo el apoyo y comprensión que me brindaron para lograr todas mis metas. A ustedes, por todo su esfuerzo para darme una preparación profesional, dedico ésta tesis con mucho cariño. Muchas gracias.

A MI FAMILIA:

Muy en especial a mi abuelita Meche por todo el cariño y ayuda que me ha brindado durante toda mi vida. Muchas gracias.

A MIS COMPAÑEROS:

Miguel, Felipe, Mauricio y Oscar

Por el empeño y dedicación que pusimos para el desarrollo de esta tesis.

A LA ING. JAQUELINA LÓPEZ BARRIENTOS:

Por darnos su apoyo y asesoría como directora de esta tesis.

A TODAS LAS PERSONAS:

Que me apoyaron y confiaron en mí durante toda mi formación profesional. Muchas Gracias.

LOURDES NIETO AGUILAR.

ÍNDICE

INTRODUCCIÓN

Capítulo I. Generalidades

1.1 Antecedentes	1
1.1.1 Historia	1
1.1.2 Informática	2
1.1.3 Reestructuración	9
1.2 Organización de la empresa	19
1.3 Organización de la Gerencia de Tesorería	22

Capítulo II. Estudio de viabilidad

2.1 Metodología Yourdon	24
2.2 Identificación de la problemática	26
2.3 Situación actual	27
2.4 Propuesta de solución	29
2.5 Características de operación	30
2.5.1 Adquisición de compromisos para pago	30
2.5.2 Registro de movimientos bancarios	31
2.5.3 Explotación de información	32
2.6 Análisis costo/beneficio	32
2.6.1 Costo de desarrollo	33
2.6.2 Costos de operación	33
2.6.3 Comparación operación manual vs. Operación automatizada	36

Capítulo III. Análisis

3.1 Recursos	37
3.1.1 Hardware	37
3.1.2 Software	38
3.2 Diagrama de contexto	41
3.3 Especificación de requerimientos	43
3.4 Diagrama de flujo de datos	43
3.4.1 Definición de un DFD	43
3.4.2 Clasificación de DFD's	43

ÍNDICE

3.5 Diagrama de flujo operativo	50
3.5.1 Definición de DFO	50
3.5.2 Componentes de un DFO	50
3.5.3 Recomendaciones generales para la elaboración de DFO	50
3.6 Árbol de menús	62
Capítulo IV. Diseño	
4.1 Conceptos básicos	64
4.2 Diagrama de Entidad-relación	69
4.3 Diagrama de flujo de control	72
4.4 Especificación de procesos	72
4.5 Esquema de la base de datos	87
Capítulo V Desarrollo e implantación	
5.1 Calendario de actividades	104
5.2 Identificación de programas	104
5.3 Creación de programas	107
5.4 Pruebas	127
5.5 Implantación	128
5.5.1 Capacitación	128
5.5.2 Conversión	130
Capítulo VI. Mantenimiento	
6.1 Tipos de mantenimiento	132
6.1.1 Correctivo	132
6.1.2 Preventivo	133
6.1.3 Perfectivo	133
6.1.4 Aumentativo	133
6.1.5 Tecnológico	133
6.1.6 Adaptativo	133
6.2 Administración del mantenimiento	134
6.3 Factores que afectan a los costos de mantenimiento	134
6.4 Problemas de mantenimiento	136
6.5 Características del mantenimiento	137
6.6 Mantenimiento estructurado frente al no estructurado	137
6.7 Efectos secundarios del mantenimiento	139

ÍNDICE

CONCLUSIONES	141
BIBLIOGRAFÍA	143
GLOSARIO DE TÉRMINOS	145
APÉNDICES	
Apéndice A. Manual de Usuario	A1
Apéndice B. Muestras de Programas	B1
Apéndice C. Muestras de Productos	C1

INTRODUCCIÓN

INTRODUCCIÓN

La captación de recursos monetarios así como la mejor aplicación financiera que de ellos se haga, con ayuda de los actuales instrumentos de inversión, resulta ser uno de los pilares para una economía sana en las empresas.

No menos importante es el cumplimiento de los compromisos de pago con entidades externas, así como la negociación de créditos con las grandes firmas y, por supuesto, la solvencia para liquidar los requerimientos de pago del personal.

Es por ello que se hace patente la conveniencia y necesidad de contar con un sistema eficiente, sencillo y ágil que controle tanto la disponibilidad de efectivo como los compromisos de pago de la empresa.

En consecuencia la Gerencia de Tesorería de Ferrocarriles Nacionales de México (FNM), como el área responsable administrativa de dichas funciones, consciente del reto actual de modernización que requiere el organismo, ha solicitado la colaboración de la Coordinación de Descentralización Informática para desarrollar un Sistema Integral de Información que controle el flujo de efectivo y los movimientos financieros que se efectúan.

En apoyo a este requerimiento la Coordinación de Descentralización Informática (CDI) ha tomado el compromiso de realizar este Sistema Integral de Información, el cual dará respuesta a las necesidades de la Gerencia de Tesorería dentro de un ambiente de desarrollo informático Institucional, buscando que las áreas constitutivas de la empresa puedan explotar la información generada, transformándola según sus propias necesidades de forma más adecuada para la toma de decisiones.

Además de asegurar una mayor transparencia en el manejo de los fondos y valores de la institución, pues la Gerencia de Tesorería es la responsable, en primera instancia, de concentrar y custodiar la recepción del efectivo; y en segunda instancia, liquidar los documentos que amparan las obligaciones contraídas por las diversas áreas de la entidad con proveedores, contratistas y prestadores de servicio de acuerdo con la normatividad establecida.

Así mismo, es la encargada de transferir a cuentas bancarias los importes de las percepciones de sueldos y salarios del personal activo y jubilados; invertir los fondos remanentes transitorios en instrumentos de renta fija autorizados; obtener los subsidios otorgados por el Gobierno Federal de acuerdo a los calendarios establecidos en relación a las disponibles financieras y, tramitar y obtener los boletos de transportación aérea nacional e internacional del personal del organismo en comisiones del servicio.

Por lo anterior, como integrantes de la (CDI) se nos encomendó la tarea de crear el sistema que responda a las necesidades resultantes del ejercicio de las funciones de la Tesorería de (FNM), mismo que hemos elegido como tema de tesis para lograr el grado de licenciatura. A éste sistema hemos convenido denominarlo (SHTES).

INTRODUCCION

Para llevar a cabo esto, en el capítulo I de este trabajo se presenta la situación del desarrollo histórico de la empresa y su organización, la situación actual en el área y las necesidades de la misma, en el capítulo II se define metodología que se utiliza para el modelo del sistema, así también las características, alcances y limitaciones, así como el análisis costo/beneficio del sistema propuesto, en el capítulo III se describen las características del ambiente de operación (Hardware y Software) para el desarrollo del sistema, se definen las necesidades y se obtienen los diagramas de contexto, de flujo operativo, de flujo de datos y árbol de menús; en el capítulo IV se define la estructura de la base de datos, la organización de las tablas y datos; y finalmente el diagrama de entidad-relación; en el capítulo V se definen todas las especificaciones de los programas y se realiza su programación y pruebas, se especifican y crean todas las tablas requeridas por el sistema; se elabora la documentación requerida, tanto del sistema en general, como de los programas que lo integran, se realizan las pruebas necesarias de cada uno de los módulos del sistema por separado; se realizan las pruebas del sistema integral en paralelo con el sistema manual; se realiza la implantación y liberación de éste; en el capítulo VI se definen los pasos a seguir en caso de surgir la necesidad de algún tipo de mantenimiento como puede ser correctivo, preventivo, perfectivo, y/o aumentativo; y por último se presentan los comentarios y conclusiones acerca de la realización de este proyecto.



CAPÍTULO I

CAPÍTULO I GENERALIDADES**1.1 Antecedentes****1.1.1 Historia**

La empresa ferroviaria tiene una historia amplia que data del siglo pasado en que se inició este servicio, y a la fecha cuenta con una estructura organizacional que se rige dentro de un marco jurídico completo y definido.

Ferrocarriles Nacionales de México es una institución del sector público catalogada como paraestatal y normada por la Secretaría de Comunicaciones y Transportes, y por lo tanto tiene como función proporcionar servicio de transporte y comunicación.

Su historia se inicia en 1837, cuando se otorgó la primera concesión ferrocarrilera, pero fue hasta 1850 cuando recorre los primeros 13.6 Km de la ciudad de Veracruz a el Molino.

El primero de enero de 1873, el Presidente Sebastián Lerdo de Tejada inaugura el Ferrocarril Mexicano, propiedad de la Compañía Limitada, con un viaje de 423.7 Km entre la Ciudad de México y el Puerto de Veracruz.

En 1884 una compañía norteamericana termina 1,970 Km de vía para enlazar, por medio de su Ferrocarril Central Mexicano, a la Ciudad de México con la frontera norte en lo que ahora es Ciudad Juárez y diversas ciudades de los Estados Unidos.

En 1908 el Presidente Porfirio Díaz firma un convenio con empresarios estadounidenses para formar juntos Ferrocarriles Nacionales de México, cuya administración se ubicaría en nuestro país. En esta mexicanización de la empresa, juega un papel relevante la Gran Liga Mexicana de Empleados del Ferrocarril, y en especial el Sr. Felipe Pescador.

Durante la Revolución se inutilizan más del 40% de locomotoras y carros, lo cual ocasiona un alto desempleo y muchos otros problemas durante años.

Para salvar a la empresa, en 1937 el Presidente Lázaro Cárdenas decreta su Expropiación y en 1938 la entrega a los trabajadores para su administración. No se resuelve la crisis económica y en 1940 termina la experiencia de autogestión. El 30 de diciembre de 1948 el Presidente Miguel Alemán decreta la existencia jurídica de Ferrocarriles Nacionales de México (FNM) como Organismo Descentralizado.

Hechos como los anteriores muestran la complejidad que ha tenido el desarrollo de esta empresa.

Múltiples problemas y contratiempos han sido sorteados; sin embargo ha crecido y sigue enfrentando retos. Actualmente cuenta con 24,000 Km de vía, 40,000 carros de carga y 1,200 locomotoras. Desde 1992 se ha impulsado un proceso de modernización y de cambio estructural para llegar a ser una empresa competitiva y con autosuficiencia financiera.

1.1.2 Informática

Con la incorporación de Sistemas Informáticos se espera mejorar la estructura de la empresa, y la interconexión de estos sistemas mediante las actuales tecnologías de redes de datos, es un paso adelante en el proceso de modernización.

La Subdirección General de Finanzas, a través de la Coordinación de Descentralización Informática, ha planteado el Proyecto Institucional de Desarrollo Informático Administrativo (PIDIA), mismo que se expresa a través de 4 puntos, que en forma general se enuncian:

- Desarrollo del Sistema Integral de Información Distribuida (S.I.I.D), que lo conforman 15 macrosistemas administrativos a nivel institucional, compartiendo información distribuida a lo largo de todo el territorio nacional.
- Definición de los estándares de desarrollo institucional a ser usados para desarrollar el (SIID). Para este punto se determinaron todas las normas y lenguajes de mayor conveniencia para el desarrollo de los sistemas basados en computadora que conforman el (SIID).
- Determinación del modelo de Equipamiento para cada área de la empresa. Es necesario definir la infraestructura presente y futura, sobre la cual funcionarán los componentes del (SIID). De esta forma se asegura el éxito de los mismos, al momento de su implantación.
- Integración del potencial de cómputo, a través de los medios de comunicación de datos (FERROPAC) (Red Integral de Información Computarizada). Los puntos anteriores sugieren de manera natural la necesidad de integrar el potencial de cómputo, y con ello las bases de datos a través de una red de comunicaciones que soporte la utilización compartida de la información con eficiencia suficiente para soportar la modernización administrativa de (FNM).

Entre los sistemas que tiene conceptualizados el (SIID) está el Sistema de Tesorería, el cual nos permitirá controlar los egresos e ingresos, además de los diversos movimientos bancarios que se manejan a diario.

PROYECTO INSTITUCIONAL DE DESARROLLO INFORMÁTICO ADMINISTRATIVO (PIDIA)

Es evidente, que se deben sentar bases sólidas para resolver la problemática que en el campo informático vive Ferrocarriles Nacionales de México. Así pues, surge el Proyecto Institucional de Desarrollo Informático Administrativo (PIDIA), como respuesta a este proceso de modernización, mismo que propone de manera interrelacionada los cuatro puntos básicos antes referidos como esquema de composición, y que conformarán una plataforma importante en la que se estimule en gran medida la evolución y modernización de la Empresa.

En el pasado se realizaron un gran número de esfuerzos para la sistematización de los procesos técnicos, operativos y administrativos, sin embargo estos esfuerzos se han realizado en forma aislada, por lo tanto existe duplicidad de trabajo; ya que al no existir una normatividad en la planeación de cada uno de los procesos, la información que se genera de estos resulta incompatible entre sí.

En el año de 1991 se practica el estudio sobre las condiciones prevalentes en el campo de la informática en (FNM) dicho estudio conduce al siguiente diagnóstico:

- Desarrollos aislados.
- Aplicación y esfuerzos duplicados.
- Deficiencia de oportunidad de la información.
- Inconsistencia.
- Centralización.
- Falta de estándares.
- Escasez de normatividad.
- Sistemas obsoletos.
- Alto volumen de operaciones manuales.
- Mal uso de los recursos de cómputo.
- Desintegración de la información.

En conclusión, la problemática es una **ausencia de planeación informática.**

Para resolver la problemática que en el campo informático se tiene en Ferrocarriles Nacionales de México, surgió el Proyecto Institucional de Desarrollo Informático Administrativo (PIDIA).

El (PIDIA) estableció los siguientes objetivos, alcances y metas para dar solución a la problemática detectada:

Objetivos

- Aplicación de recursos informáticos de acuerdo a las necesidades detectadas.

- Aplicación de técnicas para el desarrollo de sistemas, operando procesos de información distribuida y de compartición de datos a diferentes niveles de acceso.
- Incorporación del uso de la tecnología de redes de computadoras.
- Aplicación de metodologías de análisis y diseño estructurado, para el desarrollo de una nueva generación de sistemas informáticos.

Alcances

- Uso de elementos actuales y de tecnología moderna, que aseguren la permanencia de la infraestructura informática, y que además garanticen su actualización, de acuerdo con los avances tecnológicos.
- Uso del nuevo enfoque de construcción de sistemas con herramientas de cuarta generación, que aseguren calidad, alta producción y mantenimiento simple.
- Desarrollos integrales, estandarizados y de uso general, tanto en su operación, como en su consumo y producción.
- Disponibilidad de información a todos los niveles, bajo el concepto de proceso distribuido, el de consulta y concurrencia, sobre las bases de una infraestructura de equipo de cómputo de capacidades diferentes, localizados remotamente e interconectados a través de una red combinada de satélite, fibra óptica, microondas y señales telefónicas.

Metas

- Composición de la Red de Cómputo Distribuida sobre 112 estaciones, 22 secciones de tiempo, 25 divisiones y 5 ferrocarriles; con un total de 164 nodos.
- Desarrollo de los nuevos sistemas de facturación, nómina, presupuestos, adquisiciones, almacenes que se implantarán en las 122 estaciones, 22 secciones de tiempo, 25 divisiones y 5 ferrocarriles.
- Desarrollo de 15 macrosistemas administrativos, conformando así el Sistema Integral de Información Distribuida (SIID) de Ferrocarriles Nacionales de México.

DESCRIPCIÓN DE COMPONENTES BÁSICOS.

El Proyecto Institucional de Desarrollo Informático Administrativo (PIDIA) se integra, como ya se mencionó, de cuatro componentes que se describen a continuación.

1.- Desarrollo del Modelo de Sistemas Integrales de Información Distribuida.

Ferrocarriles Nacionales de México tiene una gran necesidad de Sistemas apoyados en computadora, tanto en el marco de su operación técnica, como en su proceso administrativo.

Sin embargo, como ya se mencionó, esta necesidad debe ser atendida bajo un esquema de modelo, aplicando el concepto de desarrollos integrales que aseguren la compartición de los datos, evitando duplicidad y el mal aprovechamiento de recursos, tanto humanos como tecnológicos; bajo este concepto, será posible distribuir la información desde las localidades donde se produce, y complementarla desde lugares distantes, ejecutando programas ó procesos referidos a requerimientos de diferentes niveles y especificaciones.

Esto es lo que se ha denominado Sistema Integral de Información Distribuida (SIID), y que agrupará un conjunto de macrosistemas a desarrollar, como primera respuesta a la problemática detectada.

El (SIID) pretende cubrir mediante la automatización de funciones, la simplificación de todos los trámites, tanto internos como externos, incrementando: el control, la veracidad y la oportunidad en la información entre todas y cada una de las subdirecciones.

Al conceptualizar el (SIID) se tienen como premisas fundamentales:

- Simplificar las funciones administrativas de las áreas.
- Eliminar la redundancia de funciones afines.
- Compartir la información entre los diversos sistemas.
- Incrementar la veracidad de la información.
- Responsabilizar en forma específica a cada una de las áreas sobre el mantenimiento y actualización de la información que le corresponda.
- Distribución del proceso en los diferentes sectores del organismo.

El (SIID) contempla diversos sistemas para apoyar el proceso operativo de Ferrocarriles Nacionales de México, los cuales están interrelacionados entre sí, compartiendo la información que se genera en cada uno de estos sistemas en forma particular; para esto se utilizará la teoría de las bases de datos relacionales, y que será accesada mediante un lenguaje estándar de cuarta generación llamado INFORMIX 4GL.

Este tipo de software de cuarta generación tiene la capacidad de ser migrable a cualquier tipo de equipo que actualmente tiene la empresa, esto implica, que si se realiza un desarrollo para el equipo Hewlett Packard con el manejador de base de datos Informix, esa misma aplicación puede implantarse en cualquier equipo compatible.

2.- Determinación de un Software de Alto Nivel para ser usado como estándar en el desarrollo de sistemas a nivel Institucional.

Se ha detectado dentro de la problemática, la ausencia de un estándar para el desarrollo de los sistemas, por lo cual actualmente Ferrocarriles Nacionales de México utiliza diferentes lenguajes manejadores de bases de datos y varios sistemas operativos.

Bajo tal esquema, resulta evidente la imposibilidad de un desarrollo integral, pues existen incompatibilidades a veces insalvables en estas plataformas. Con el objeto de crear un estándar de trabajo en la creación e implantación de sistemas de información a nivel central y regional, Ferrocarriles Nacionales de México (FNM), a través de la Coordinación de Descentralización Informática (CDI), realizó en Marzo de 1991 un estudio que permitió evaluar los paquetes de software para desarrollo más reconocidos hasta esa fecha, los resultados que se obtuvieron en dichas pruebas de comparación permitieron elegir un Manejador de Bases de Datos como el estándar en el desarrollo de sistemas a nivel institucional.

Como es de suponerse, la decisión final implicó un análisis profundo, tanto de carácter técnico como costo-beneficio que pudiera asegurar un soporte real y adecuado a las características particulares de la institución.

3.- Equipamiento Racional, Planeado y Modelado para cada Área de la Empresa.

Es esencial definir un modelo para equipar cada una de las áreas que conforman Ferrocarriles Nacionales de México, y así establecer una plataforma en la que se asienten los desarrollos determinados en el (SIID).

De la misma manera que el (PIDIA) se integró como una necesidad primaria el contar con un estándar de software para el desarrollo de los sistemas, se consideró también la necesidad de definir un estándar de equipamiento de las áreas que desarrollan todas las funciones administrativas dentro de la empresa.

A finales de 1990 y antes de la actual administración se realizó la adquisición de 6 minicomputadoras HP-9000/835 de la siguiente configuración:

- Procesador PA-RISK de HP.
- Memoria RAM de 24 MB.
- Disco duro de 1.964 GB.
- Multiplexor de puertos con capacidad para 32 terminales en conexión serial.
- Consola de operación HP-700/92.
- 32 Terminales modelo HP-700/44.

Los equipos HP-9000 se encuentran ubicados uno en cada una de las sedes regionales y otro en las oficinas centrales. Estos equipos fueron instalados y puestos en operación hasta el mes de Abril de 1992 por personal de la C.D.I.

4.- Integración del Potencial de Cómputo, a través de los Métodos de Comunicación de datos (FERROPAC).

La unificación de los Ferrocarriles en una sola organización exige acompañarse de servicios administrativos eficientes y de control de información de tipo general a fin de normalizar todas las áreas de la empresa.

Ferrocarriles Nacionales de México crea su propia red de comunicación de datos, la cual le permitirá dar respuesta a todas las demandas que exige el país, basando su estructura y características, en tecnología acorde a las necesidades propias de la empresa.

Los pasos que se han dado en cuanto a la adquisición de equipos de cómputo para los ferrocarriles de las regiones y el establecimiento de una red de estaciones terrenas para comunicaciones vía satélite, facilitan la implantación de sistemas informáticos con la filosofía de concentración de operaciones con su escaso riesgo de pérdida de información con mayor control y consistencia.

Se planeó establecer la Red de Telecomunicaciones de Datos (FERROPAC), ésta es necesaria para soportar el gran intercambio de información entre los puntos de concentración con una gran eficiencia y seguridad.

Personal con experiencia se ha dedicado al diseño e implantación de redes de comunicación de datos, a fin de garantizar que la estructura de la Red permita soportar el tráfico de las distintas etapas de incorporación de los sistemas distribuidos y la conexión, en su momento, de otras redes a la Red principal.

La red de (FERROPAC) tiene como objetivo enlazar inteligentemente a las distintas unidades administrativas y operativas de esta organización, a fin de soportar la operación del organismo bajo un enfoque de eficiencia, confiabilidad y modernización.

Por lo anterior, para el diseño de la red, se contemplaron las siguientes consideraciones:

- Disponer de la infraestructura necesaria para transferir información entre todos los puntos por apartados que estén, considerando también el esquema funcional entre estaciones, secciones de tiempo y sus divisiones, éstas con las Gerencias Regionales y éstas últimas con las Oficinas Centrales.
- Manejo de los distintos sistemas de información que se desarrollen, atendiendo a los diversos esquemas de comunicación que se propongan entre unidades administrativas. Al contar con redes de máquinas UNIX interconectadas y utilizando bases de datos distribuidas, se debe tener presente el concepto de proceso distribuido, ya que deben ser capaces de buscar, ejecutar o mostrar datos a cada usuario conectado en cualquier terminal de la red en forma transparente.
- Que exista flexibilidad para soportar la conexión de otras redes que en un futuro se puedan incorporar.

El esquema de la red (FERROPAC) se encuentra formado de la siguiente manera:

- Un sistema de redes LAN tipo Ethernet, interconectadas por equipo de comunicación Bridge/Routher con enlaces digitales a 64 kbps y líneas dedicadas de microondas bajo el protocolo TCP/IP.
- Como sistema de respaldo, una red de conmutadores X.25 con enlaces vía microondas a una velocidad de 9.6 kbps, ofreciendo redundancia a la red principal, en los enlaces entre las Oficinas Centrales y los Ferrocarriles.

De esta manera, la arquitectura queda formada por un conjunto de LAN's tipo Ethernet, interconectadas a través de Brouters (Puentes-Ruteadores), los cuales aumentan las facilidades de las LAN's hacia áreas extendidas, que en este caso es a nivel nacional.

Para esta interconexión se requirió de líneas de comunicación conmutadas y privadas. Las líneas conmutadas de (FNM), son aquellas líneas que utilizan la red telefónica, enrutadas por el propio conmutador, satélite y fibra óptica.

Las demás estaciones y secciones de tiempo quedan conectadas a través de concentradores de manera indistinta a la red LAN divisional o al ferrocarril dependiendo de la distancia y de las facilidades de comunicación que existan.

Este diseño contempla el Plan de Desarrollo Informático Administrativo (PIDIA) y prevve un crecimiento hasta de 10 veces dicho tráfico.

En cuanto a la cobertura de la red, el diseño permite la conexión de las 185 unidades administrativas que se definieron en tres niveles.

Primer Nivel.- Conexión entre los ferrocarriles y Oficinas Centrales de alta capacidad (64 kbps). En relación de los interprocesos que actuarán entre éstas. Ver figura 1.1.

Segundo Nivel.- Comunicación entre los ferrocarriles y sus divisiones. Ver de la figura 1.2 a 1.6

Tercer Nivel.- Es el nivel de estaciones, en donde se establece la comunicación entre las estaciones y/o secciones de tiempo y sus divisiones. Ver de la figura 1.2 a 1.6.

Servicios que ofrece TCP/IP

Con el objeto de facilitar las operaciones de transferencia de datos TCP/IP ofrece:

- Transferencia de archivos (FTP).
- Servicios de terminal virtual (telnet).

- Abrir sesiones remotas (rlogin).
- Correo electrónico (sendmail).
- Sistema de archivos en red (NFS).
- Monitoreo de la red.
- Enlaces de PC's.
- Ambientes de ventanas.
- Facilidades de comunicación al sistema operativo UNIX.
- Distribución de aplicaciones INFORMIX.
- Conectividad hacia IBM (SNA), Token Ring (IEEE 802.5).

La red (FERROPAC) está integrada por cuatro tipos de equipo de cómputo:

- Computadora HP-T500.
- Computadoras HP-9000/835.
- Computadoras HP-9000/807.
- Computadoras Personales.

Conectados mediante Brouters, Concentradores, Xyplex, Weefflet y equinox.

1.1.3 Reestructuración

Así, desde su inicio el (PIDIA) concibió a la informática en (FNM) en un ambiente descentralizado pero controlado, distribuido pero cooperativista, autónomo pero integral.

Dicho proyecto ha alcanzado diferentes grados de madurez en cada uno de sus puntos, pero fundamentalmente ha avanzado en la desconcentración de la informática, para lo cual su enfoque ha dirigido los esfuerzos a la creación y el fortalecimiento de las áreas informáticas regionales.

Hoy en día (FNM) vive un profundo cambio y enfrenta el compromiso de la privatización, para lo cual se reestructura en principio en 5 unidades o ferrocarriles básicos y 62 líneas cortas.

- FERROCARRIL NORESTE.
- FERROCARRIL PACIFICO-NORTE.
- FERROCARRIL SURESTE.
- FERROCARRIL VALLE DE MÉXICO.
- FERROCARRIL CHIHUAHUA-PACIFICO.

El principio y objetivo de todas esas unidades es la búsqueda de una autonomía y autosuficiencia en el corto plazo de todas sus funciones tanto operativas como administrativas.

Bajo este panorama la informática tal y como lo concibe el (PIDIA), se asienta de manera natural sobre la nueva reestructuración de la empresa, posibilitando con ello el objeto buscado.

El (PIDIA) ha desarrollado todos sus sistemas de forma tal, que su operación es descentralizada, versátil, movable, ajustable, sin cambios técnicos significativos. Del mismo modo, dentro del proyecto se ha creado el ambiente en un marco técnico estándar, los locales de cómputo y el personal técnico capacitado para que el funcionamiento de los sistemas sea óptimo en cada nuevo ferrocarril; así mismo el modelo de equipamiento diseñado por el (PIDIA) garantiza cualquier cambio o traslado y crecimiento de equipo sin trastorno alguno.

Por último se cuenta con la comunicación entre todas las computadoras internas de los nuevos ferrocarriles y las correspondientes y necesarias entre los diferentes ferrocarriles y el corporativo.

La situación actual y resultado del (PIDIA) sobre los puntos anteriores son los siguientes:

Sistemas implantados:

Bajo el concepto descentralizado se han implantado y están operando e intercambiando información sobre todos los puntos de la red los siguientes sistemas:

SIIPAD	Pago descentralizado al personal (cinco nóminas diferentes).
SIPTAL	Control del presupuesto.
SIISEC	Servicios de carga, guía de embarque, ingresos, facturas y tarifas.
SIICALL	Control de llamadas a trenistas.
SISEDI	Control y cálculo de demoras.
SIICOFIN	Control de financiamientos (carta de crédito).
SIIRVOL	Cálculo y pago del retiro voluntario.
SIUREM	Control de adquisiciones y almacenes.
SIIMANEQ	Control y cálculo de mantenimiento de locomotoras.
SITREN	Apoyo a la reestructuración.
SISEF	Seguros y fianzas.
SIIDOF	Sistema de documentación de flete.

Sistemas desarrollados y sin implantar:

SIIAPI	Control de bienes inmuebles.
SIICOF	Contabilidad.
SIIPAD	Módulo de trenistas valor viaje.
SIICLOC	Asignación y control de locomotoras.

Sistemas pendientes de desarrollo:

SIICRECO	Crédito y cobranzas.
SHAUDI	Control de auditorías.
SHAVA	Inventarios y avalúos.

Los sistemas desarrollados y en operación pueden funcionar automáticamente en cada ferrocarril, en muy corto plazo, y concentrarán la información que se requiera.

Se deberán implantar los proyectos terminados y pendientes ya que existe el esfuerzo concluido y una completa compatibilidad para operar en las nuevas unidades de reestructuración, por último, se debe reafirmar y valorar en su caso, el compromiso para los desarrollos pendientes.

El (PIDIA) ha propiciado desde su creación la desconcentración de funciones y procesos de información administrativa, para lo cual fue necesario establecer las condiciones para la operación de los 27 macrosistemas en desarrollo, ellos fueron diseñados para operar directamente por el personal responsable de cada función administrativa, en las propias áreas de trabajo.

Los proyectos mencionados se han desarrollado por el propio personal informático de (FNM) usando lenguaje y tecnología actual y de alto nivel, para ello se ha capacitado intensamente a 126 Ingenieros de Sistemas sobre 61 temas, se cuenta con estándares, metodología y normatividad informática cuidadosamente seleccionados; se buscó la autosuficiencia técnica y de soporte en cada región de las siguientes áreas informáticas:

Una central normativa y alto desarrollo (desde donde se han desarrollado e implantado los sistemas mencionados).

Cinco regionales de operación y bajo desarrollo (de soporte y apoyo a los sistemas después de su implantación y desarrollos locales).

Se adecuaron y se dotaron de computadoras, impresoras, periféricos y nodo de conexión a la red de datos, a los siguientes centros de cómputo.

- Una central de alta capacidad (200 usuarios)
- Seis pequeñas centrales (16 usuarios)
- Cinco medianas regionales (64 usuarios) y
- 32 pequeños divisionales y/o estaciones (16 usuarios)

Se instaló una red de datos que cuenta con 44 nodos y 504 equipos que permite conectar a 900 usuarios de sistemas y equipos de cómputo para intercambiar información.

ESTRUCTURA DEL FERROCARRIL**Organización, estándares y normatividad**

La organización regional del personal informático marcada por el (PIDIA) es válida para los nuevos ferrocarriles, ya que en su mayoría coinciden con las sedes anteriores, sin embargo, es necesaria la creación de una nueva área de informática: la del VALLE DE MÉXICO, ésta tomará parte del personal de la ex-sede regional QUERÉTARO. Valle de México representa una demanda de personal informático en esta primera etapa, en el caso de CHIHUAHUA es necesario particionar la organización regional anterior, una parte para el nuevo ferrocarril OJINAGA/TOPOLOBAMPO y otra apoyando al área de informática del pacífico norte, ante la ampliación de la extensión territorial de ésta última.

Los estándares y normatividad son vigentes, es de insistir en su observancia y actualización, es necesario una doble capacitación al personal informático de cada ferrocarril, por un lado tendiente a la actualización tecnológica y por otro lado la autosuficiencia para el soporte técnico a los desarrollos institucionales que quedaran a su cargo.

Equipamiento de cómputo y red de datos

El equipo de cómputo y de red de datos está instalado en los lugares donde se requieren, de acuerdo a las necesidades de proceso y al volumen de información actual, por lo que de inicio serán necesarios cambios mínimos, solamente en el caso de QUERÉTARO dónde el equipo mayor será trasladado al nuevo ferrocarril VALLE DE MÉXICO, dejando en la nueva división QUERÉTARO de NORESTE suficiencia para las necesidades locales, adicionalmente se ha diseñado un programa que demandará inversión para actualizar la capacidad instalada tanto de equipo de cómputo como de la propia red de datos, en él se pretende la incorporación de 8 nuevos pequeños centros divisionales de cómputo y 6 aplicaciones para igual número de medianos centros de cómputo. Todo lo anterior necesario para soportar las nuevas cargas de trabajo.

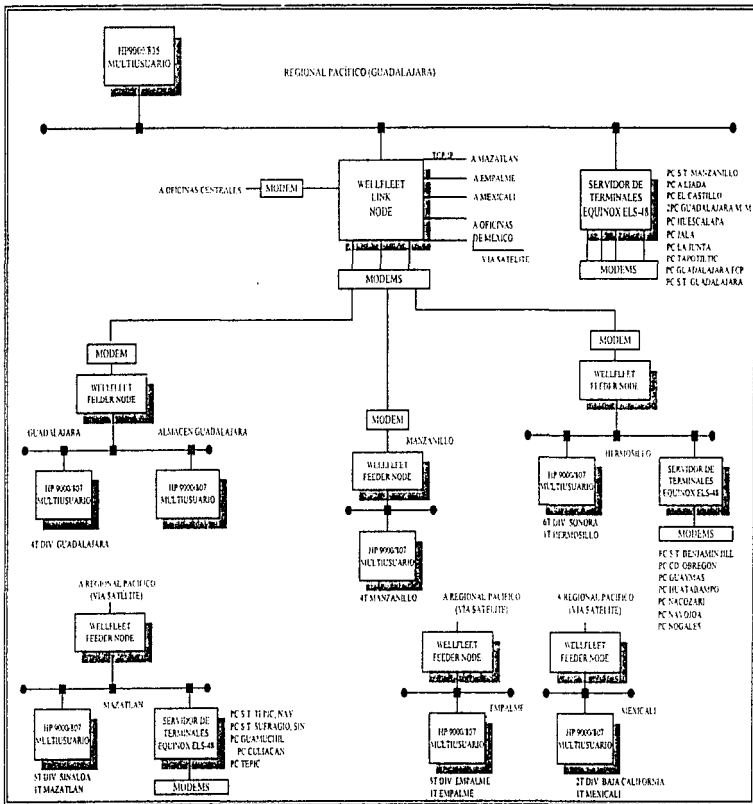


Figura 1.3 Regional Pacifico

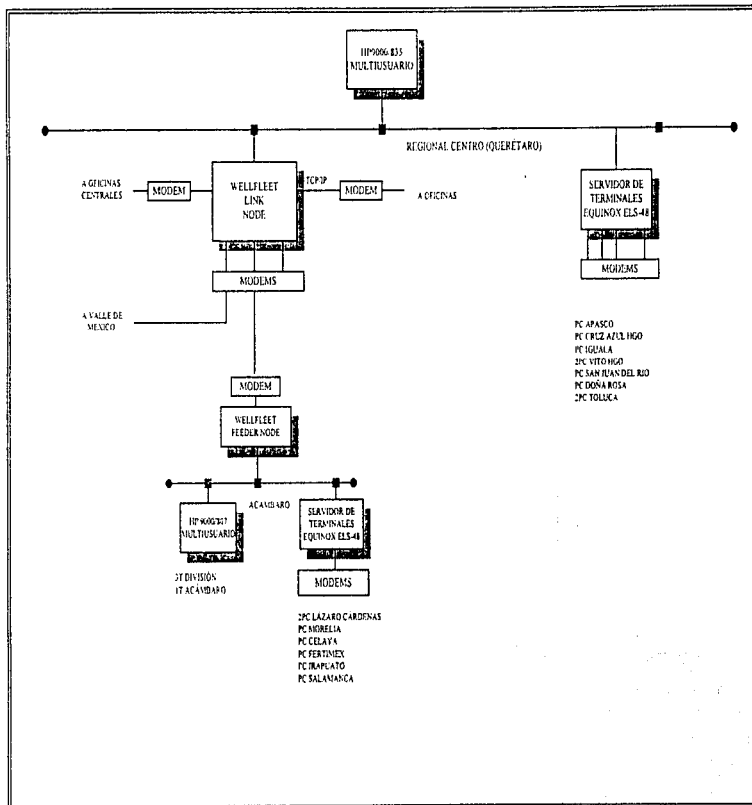


Figura 1.4 Regional Centro

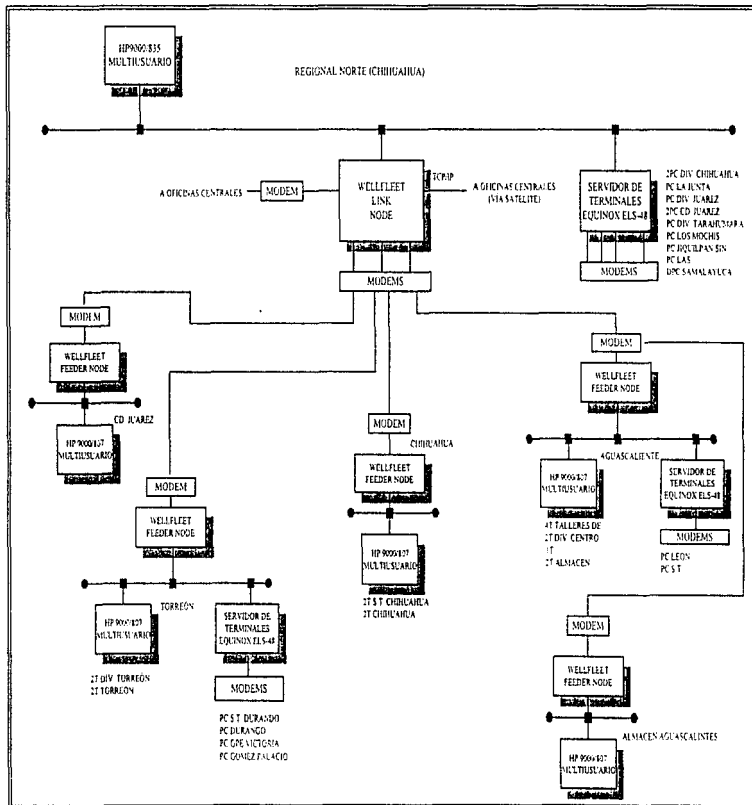


Figura 1.5 Regional Norte

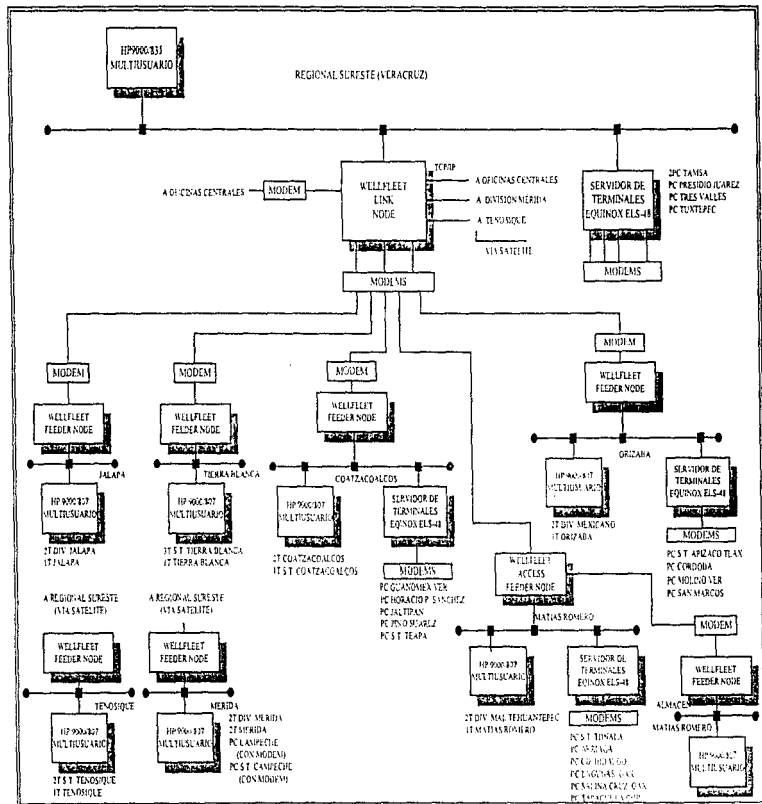


Figura 16 Regional Sureste

1.2 Organización de la empresa

La estructura directiva de Ferrocarriles Nacionales de México es la siguiente:

Un Consejo de Administración, una Dirección General de la que dependen las Subdirecciones Generales de Finanzas, Comercial y de Servicios, de Operación, de Fuerza Motriz y Equipo de Arrastre, Infraestructura y Telecomunicaciones, de Recursos Humanos y Relaciones Laborales, de Recursos Materiales, de Planeación y Sistemas, así como una Contraloría Interna y una Coordinación Ejecutiva a la cual reportan los Ferrocarriles Regionales del Centro, del Noreste, del Norte, del Pacífico y del Sureste. Finalmente una Gerencia de Comunicación Social que reporta directamente a la Dirección General.

En la figura 1.7 se muestra gráficamente la estructura orgánica de (FNM).

ESTRUCTURA Y FUNCIONES DE LA SUBDIRECCIÓN GENERAL DE FINANZAS

La Subdirección General de Finanzas está formada por una Coordinación General y las siguientes gerencias:

- Gerencia de Ingresos.
- Gerencia de Presupuesto.
- Gerencia de Tesorería.
- Coordinación de Descentralización Informática.
- Gerencia de Contabilidad.

Una de las funciones encomendadas a la Subdirección General de Finanzas, es la de llevar a cabo el Proyecto de Desarrollo Informático Administrativo (PIDIA), con el fin de administrar la información del manejo de los recursos financieros del organismo de manera eficiente, contar en forma oportuna con la información necesaria, integrar y presentar correctamente el presupuesto institucional del organismo ante las autoridades correspondientes y establecer el control de las erogaciones financieras; todo esto a través de la Coordinación de Descentralización Informática.

En la figura 1.8 se muestra gráficamente la estructura orgánica de la Subdirección General de Finanzas.

Funciones de la Coordinación de Descentralización de Informática

Normar, planear y coordinar la realización de los diversos proyectos informáticos, para la modernización de la empresa, además de administrar el uso racional del hardware y software.

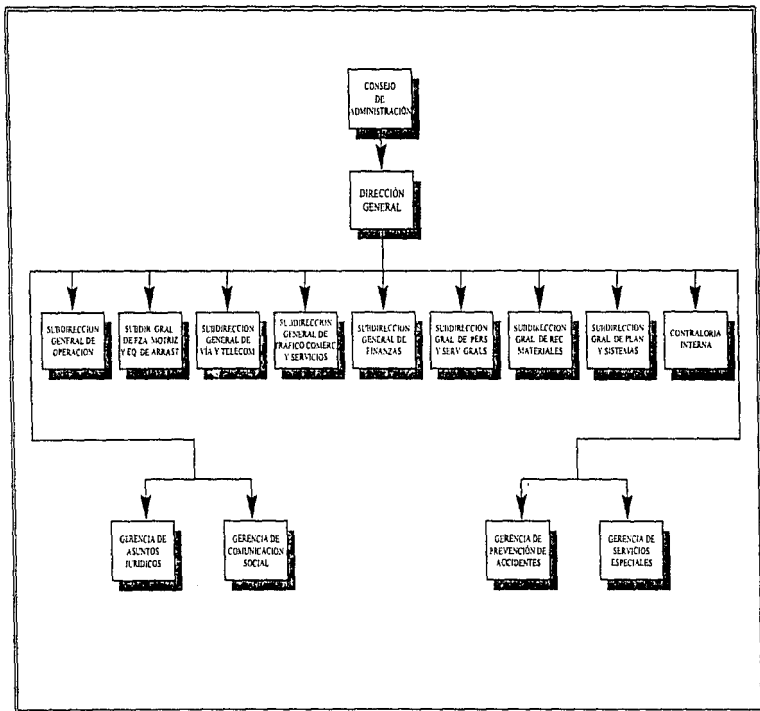


Figura 1.7 Estructura Orgánica de FNM

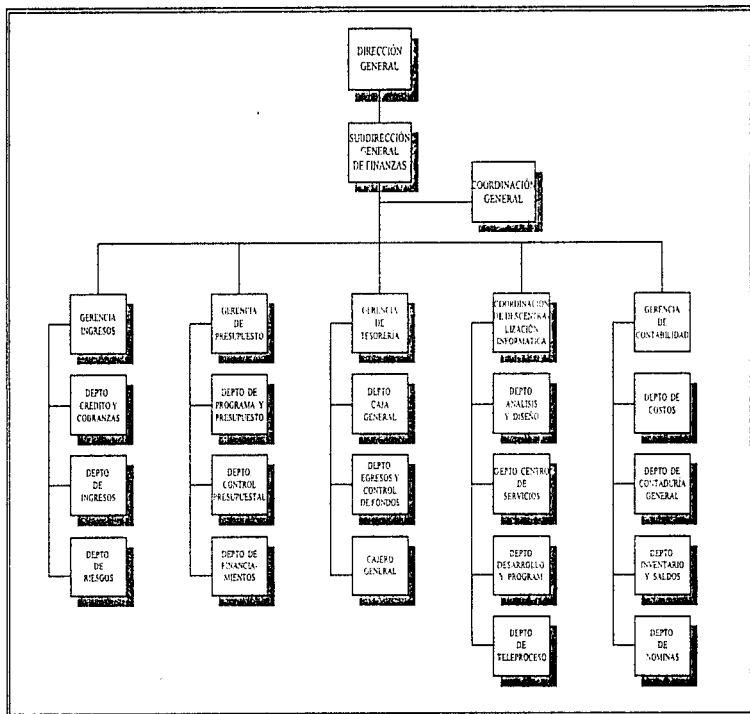


Figura 1.8 Estructura Orgánica de la Subdirección General de Finanzas

1.3 Organización de la Gerencia de Tesorería

La Gerencia de Tesorería dependiente de la Subdirección General de Finanzas, será la responsable, en primera instancia, de concentrar y custodiar el efectivo producto de las operaciones efectuadas por el organismo y, como segunda instancia liquidar los documentos que amparan las obligaciones contraídas por las diversas áreas de la entidad con su propio personal, proveedores, contratistas y prestadores de servicios de acuerdo con la normatividad establecida.

Asimismo, será la encargada de transferir a cuentas bancarias los importes de las percepciones de sueldos y salarios del personal activo y jubilado; invertir los fondos remanentes transitorios en instrumentos de renta fija autorizados; obtener los subsidios otorgados por el Gobierno Federal de acuerdo a los calendarios establecidos en relación a las disponibilidades financieras y; tramitar y obtener los boletos de transportación aérea nacional e internacional del personal del organismo en comisiones de servicio.

A continuación se muestra el organigrama de la Gerencia de Tesorería, Figura 1.9

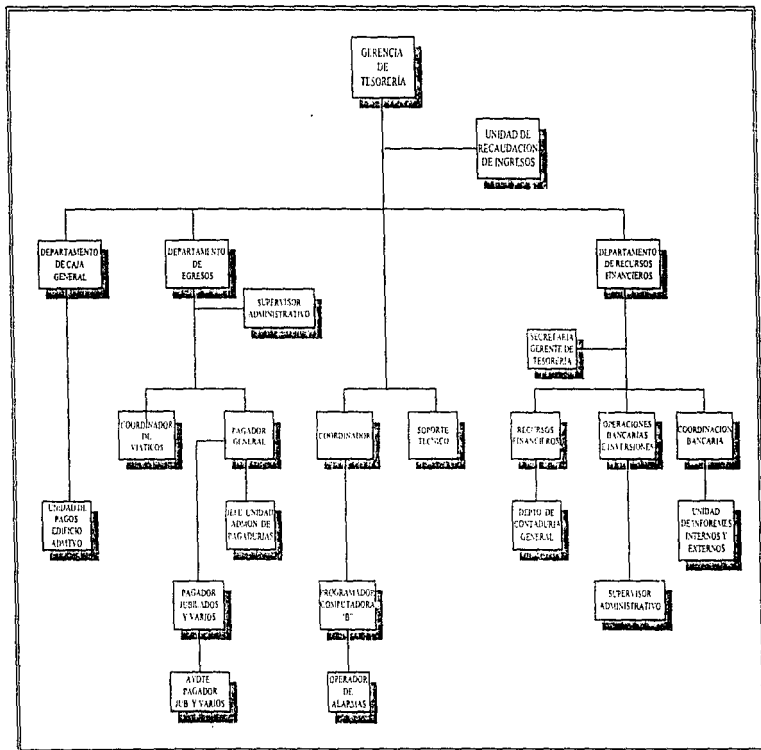


Figura 1.9 Estructura Orgánica de la Gerencia de Tesorería



CAPÍTULO II

CAPÍTULO II ESTUDIO DE VIABILIDAD**2.1 Metodología de Yourdon**

Para el ciclo de vida de un proyecto existen tres objetivos principales:

1. Definir las actividades a llevarse a cabo en un proyecto de desarrollo de sistemas.
2. Lograr congruencia entre la multitud de proyectos de desarrollo de sistemas en una misma organización.
3. Proporcionar puntos de control y revisión administrativos de las decisiones sobre continuar o no con un proyecto.

CICLO DE VIDA ESTRUCTURADO DEL PROYECTO

A continuación se describen las actividades del ciclo de vida estructurado de un proyecto.

Actividad 1: La encuesta

Esta actividad también se conoce como el estudio de factibilidad o como el estudio inicial de negocios. Por lo común, empieza cuando el usuario solicita que una o más partes de su sistema se automaticen. Los principales objetivos de la encuesta son los siguientes:

- Identificar a los usuarios responsables y crear un "campo de actividad" inicial del sistema. Esto puede comprender la conducción de una serie de entrevistas para determinar que usuarios estarán comprendidos en (o serán afectados por) el proyecto propuesto.
- Identificar las deficiencias actuales en el ambiente del usuario. Esto en general comprenderá la lista de funciones que hacen falta o que se están llevando a cabo insatisfactoriamente en el sistema actual.
- Establecer metas y objetivos para un sistema nuevo. Esto puede ser también una simple lista narrativa que contenga las funciones existentes que deben reimplantarse, las nuevas que necesitan añadirse y los criterios de desempeño del nuevo sistema.
- Determinar si es factible automatizar el sistema y de ser así, sugerir escenarios aceptables. Esto implicará algunas estimaciones bastante rudimentarias y aproximadas del costo y el tiempo necesarios para construir un sistema nuevo y los beneficios que se derivarán de ello. Aunque a estas alturas la administración y los usuarios usualmente querrán una estimación precisa y detallada, el analista tendrá mucha suerte si logra determinar el tiempo, los recursos y los costos con un error menor del 50% en esta etapa tan temprana del proyecto.

- Preparar el esquema que se usará para guiar el resto del proyecto. Este esquema incluirá toda la información que se lista anteriormente, además de identificar al administrador responsable del proyecto. También pudiera describir los detalles del ciclo de vida que seguirá el resto del proyecto.

En general, la encuesta ocupa sólo del 5 al 10 por ciento del tiempo y los recursos de todo el proyecto, y para los proyectos pequeños y sencillos pudiera ni siquiera ser una actividad formal. Sin embargo, aun cuando no se consuma mucho del tiempo y de los recursos del proyecto, es una actividad verdaderamente importante: al final de la encuesta, la administración pudiera decidir cancelar el proyecto si no parece atractivo desde el punto de vista de costo-beneficio.

Actividad 2: El análisis de sistemas

El propósito principal de la actividad de análisis es transformar sus entradas o insumos o factores principales, las políticas del usuario y el esquema del proyecto, en una especificación estructurada. Esto implica modelar el ambiente del usuario con diagramas de flujo de datos, diagramas de entidad-relación, diagramas de transición de estado y demás herramientas existentes.

Obviamente, como analista del sistema, en esto pasará la mayor parte de su tiempo.

Actividad 3: El diseño

La actividad de diseño se dedica a asignar porciones de la especificación (también conocida como modelo esencial) a procesadores adecuados (sean máquinas o humanos) y a labores apropiadas (tareas, particiones, etc.) dentro de cada procesador. Dentro de cada labor, la actividad de diseño se dedica a la creación de una jerarquía apropiada de módulos de programas y de interfaces entre ellos para implantar la especificación creada en la actividad de análisis. Además, la actividad de diseño se ocupa de la transformación de modelos de datos de entidad-relación en un diseño de base de datos.

Actividad 4: Implantación

Esta actividad incluye la codificación y la integración de módulos en un esqueleto progresivamente más completo del sistema final. Por eso, ésta actividad incluye tanto programación estructurada como implantación descendente.

Actividad 5: Generación de pruebas de aceptación

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

Dado que el desarrollo de las pruebas de aceptación puede suceder al mismo tiempo que las actividades de diseño e implantación, pudiera ser que al análisis le sea asignada esta labor al término del desarrollo.

Actividad 6: Descripción del procedimiento

En esta actividad se realiza la generación de una descripción formal de las partes del sistema que se harán en forma manual, lo mismo que la descripción de cómo interactuarán los usuarios con la parte automatizada del nuevo sistema. El resultado de esta actividad es un Manual para el Usuario.

Actividad 7: Conversión de bases de datos

En algunos proyectos, la conversión de bases de datos involucraba más trabajo que el desarrollo de programas de computadora para el nuevo sistema. En otros casos, pudiera no haber existido una base de datos que convertir. En el caso general, esta actividad requiere como entrada la base de datos actual del usuario, al igual que la especificación del diseño.

Actividad 8: Instalación

La actividad final, desde luego, es la instalación; sus entradas son el manual del usuario, descripción del procedimiento, la base de datos convertida, y el sistema aceptado. En algunos casos, sin embargo, la instalación pudiera significar simplemente un cambio de la noche a la mañana al nuevo sistema, sin mayor alboroto; en otros casos, la instalación pudiera ser un proceso gradual, en el que un grupo tras otro de usuarios van recibiendo manuales y entrenamiento y comenzando a usar el nuevo sistema.

2.2 Identificación de la problemática

Con base en el análisis efectuado con el área de Tesorería, definimos la problemática en término de los siguientes puntos:

- Debido al tamaño de la empresa, se maneja un alto volumen de movimientos de pago e ingresos, con un método administrativo que ya no puede soportar las presiones de información actuales de la empresa.
- No se cuenta con ningún sistema automatizado de cómputo que apoye el control de movimientos, es decir, todos los registros son manuales.
- El requerimiento de información para resúmenes o informes es sumamente lento.
- La información no es confiable ni precisa.

2.3 Situación actual

A continuación se describen las actividades fundamentales que se llevan a cabo diariamente en la Gerencia de Tesorería para cumplir con los compromisos que se adquieren:

CONTROL DE PAGO DE DOCUMENTOS.

- Se reciben los documentos y se valida la información, es decir, se verifican sumas, firmas, sellos y claves asignadas a cada documento. En caso de duda, se contacta por teléfono, al área que pueda aclarar la situación.
- Se autoriza el documento, es decir, una vez revisados los registros se define la forma de pago; ésta puede ser a través de cheque, efectivo o transferencia.
- Se dividen los documentos en base a la siguiente clasificación:

A) Cuando el pago se realiza en efectivo:

- Se consideran los documentos a favor de personas físicas con importe menor a \$1,000, así mismo las órdenes de viáticos cuyo importe no excede la cantidad de \$2,000.00.
- Se registra el pago en efectivo.
- Se afectan saldos bancarios y se lleva un control de las cuentas contables.

B) Cuando el pago se realiza por cheque:

- Se genera el cheque y la póliza, lo que implica realizar la distribución contable y presupuestal del documento.
- Se pasa a firma el cheque.
- Se lleva un control general de cheques.
- Se entregan los cheques firmados al cajero para que a su vez lleve un control de lo que se ha entregado y lo que conserva en caja.
- Se lleva un control de los poderes notariales autorizados.
- Se afectan saldos bancarios y se lleva un control de las cuentas contables.

C) Cuando el pago es a través de transferencia:

- Si el pago es a un proveedor directamente, después de generar el cheque, éste se manda por correspondencia con oficio que notifique el envío.

- Si se debe ordenar una transferencia, también se genera un oficio, pero esta vez dirigido al banco.

- Se integra toda la información y se envía a un área contable para que efectúe la póliza de caja y el resumen contable.
- Se archivan los documentos y resúmenes.
- Cuando se requiere algún tipo de información, se hace un rastreo en todo el archivo o se realizan sumas sobre los conceptos requeridos.
- Es importante mencionar que Ferrocarriles Nacionales de México trabaja con diferentes Bancos los cuales son:

INTERNACIONAL 2 cuentas.
BANCOMER 3 cuentas.
BANAMEX 4 cuentas.
SERFIN 1 cuenta.
INVERLAT 1 cuenta.

CONTROL DE INGRESOS (DIARIO)

- Se solicitan los recibos de depósito al área de Financiamiento de Ferrocarriles Nacionales de México.
- Se conservan los recibos generados por traspasos.
- Se registran los ingresos por inversiones.
- Se registran los ingresos por negociación con proveedores.
- Se confronta la información contra la proporcionada por el banco.
- Se agrupan los totales por los conceptos más importantes.
- Se efectúan sumas para realizar estadísticas de los ingresos recibidos en el mes.

CONTROL DE MOVIMIENTOS BANCARIOS

- Como complemento al registro de ingresos, se consulta todos los días de forma global el disponible del día, así como los depósitos y retiros registrados.
- Evalúa el compromiso de pagos y disponibilidades.
- Se realizan traspasos entre cuentas.
- Se realizan cotizaciones e inversiones.
- Se efectúan reportes diarios, semanales y mensuales de las inversiones y los pagos efectuados.
- Se consolida la información de los bancos.
- Se concilia información con los bancos.

CONTROL PRESUPUESTAL Y CONTABLE

- Se registra la información en el libro de caja.
- Se registra la información en el formato de resumen presupuestal.
- Se efectúan ajustes y correcciones.

Todo este control se efectúa bajo procedimientos manuales, ya que actualmente la Gerencia de Tesorería no cuenta con ningún sistema de cómputo que brinde apoyo a la operación diaria de los departamentos; esto aunado a actividades que desarrolla en forma paralela con las Gerencias de Contabilidad y Presupuesto, torna crítico el desempeño a nivel nacional disminuyendo la eficiencia en el tiempo de respuesta de requerimientos de información.

2.4 Propuesta de solución

En base al análisis y la problemática detectada y con base en los requerimientos planteados por la Gerencia de Tesorería se propone la creación de un Sistema Integral de Información de Tesorería (SIITES), y permita la incorporación de la Gerencia de Tesorería al Proyecto Institucional de Desarrollo Informático Administrativo (PIDIA).

Las metas por alcanzar son:

- Prever la comunicación del (SIITES) con el sistema de Presupuestos (SIIPTAL) y el sistema de Nómina (SIIPAD).
- Compartir información con el sistema de banca electrónica.
- Tener un seguimiento de los documentos en trámite para su pago.
- Emitir automáticamente cheques con sus respectivas pólizas.
 - Control de cheques.
 - Control de pagos en efectivo.
 - Control de transferencias.
- Se tendrá la facultad de dar como pagado los compromisos de pago registrados en fechas anteriores y/o los registrados el mismo día.
- Tener un control rápido, veraz y eficiente de los diferentes movimientos bancarios.

Dicho proyecto se llevará a cabo de acuerdo a la metodología estructurada de diseño y programación con los cuales se cubrirán los siguientes requisitos.

- El sistema deberá realizarse de manera que el usuario interactúe con él en tiempo real.
- El procesamiento será utilizado tiempo compartido, es decir, no habrá procesos en lote.
- Se desarrollará sobre una base de datos relacional que maneja el producto INFORMIX.

- El sistema debe ser modular.
- Se dará prioridad al desarrollo de los siguientes puntos:
 - Control sobre el pago de documentos.
 - Control sobre las operaciones bancarias.
 - Control de cheques.
- En segundo término se cubrirán:
 - Conciliación bancaria.
 - Emisión de la póliza de caja.
- Otro aspecto importante es el nivel de seguridad que se mantendrá en la operación general del sistema.

2.5 Características de operación

A efecto de lograr funcionalidad y facilidad a la operación, esta se divide como sigue:

- *Adquisición de compromisos para pago (documentos)*
- *Registro de movimientos bancarios*
- *Explotación de la información.*

2.5.1 Adquisición de compromisos para pago

El manejo del documento contará con dos claves para su registro; la primera clave se denominará número de (SIIPTAL) que al ser ingresada no permitirá la captura de la segunda clave que se denominará egreso, de no contar con número de (SIIPTAL) el programa permitirá la captura de la segunda clave.

De contar con el número de (SIIPTAL), se procede a explotar la información generada por el sistema (SIIPTAL) (Sistema Integral de Información Presupuestal, Gerencia de Presupuesto). El módulo obtendrá automáticamente los siguientes datos:

- Nombre del beneficiario
- Concepto del pago
- Número de dependencia
- Fecha de captura del documento en la Gerencia de Tesorería
- Fecha programada de pago
- Importe del documento
- Cuenta contable de cargo

El resto de datos serán capturados por el usuario:

- Claves concentradoras

- Forma de pago
- Cuenta bancaria
- Cuenta contable de abono

Una vez capturado el documento, la forma de pago indicará el camino para registrar el pago del mismo. Esta se puede realizar de tres formas:

- **Efectivo.** Cuando se registra un documento y la forma de pago es efectivo el módulo encargado de registrar el pago simplemente encontrará el documento y el usuario ingresará la fecha de pago del mismo, la afectación de saldos se hará en forma automática.
- **Transferencia.** Cuando se registra un documento y la forma de pago es transferencia, el módulo encargado de registrar el pago simplemente encontrará el documento, una vez localizado este la transferencia se hará en forma automática y la afectación de saldos también. Además este módulo contará con una opción para que en caso necesario el usuario pueda retractarse de alguna afectación incorrecta de pago.
- **Cheque.** Cuando se registra un documento y la forma de pago es cheque el módulo encargado de registrar el pago simplemente encontrará el documento, una vez localizado éste, a través del módulo asignación de cheques se asigna número de cheque, automáticamente se relaciona el documento con un número de cheque de una cuenta bancaria específica; en el módulo generación de cheque, se genera el cheque; en caso de ser necesario se contará con un módulo que cancele el cheque, que automáticamente liberará el documento relacionado al cheque, por lo tanto el documento podrá estar disponible para ser pagado con cualquier forma de pago y al mismo tiempo el número de cheque queda marcado para no poder ser reasignado. El módulo de registro de pago de cheques simplemente encontrará el documento a pagar, el usuario ingresará la fecha de pago, y automáticamente se afectarán saldos bancarios.

2.5.2 Registro de movimientos bancarios.

El registro de movimientos bancarios se divide en cuatro:

- Cotización e inversión.
- Traspasos.
- Ingresos.
- Ajustes.

Cotización e inversión. Se contará con un módulo para registrar hasta cinco cotizaciones diarias por banco, el usuario necesita ingresar la cuenta bancaria, la tasa de interés y el plazo fijo; y automáticamente se le asignará un número consecutivo. El módulo de

inversiones hará uso de la información capturada en el módulo de cotizaciones, para esto el usuario necesita ingresar la cuenta bancaria, el número de la cotización elegida, la cantidad a invertir y los días a invertir dicha cantidad. El sistema despliega automáticamente el movimiento a obtener con dicha inversión, y proporcionará un número consecutivo que identificará cada una de las inversiones.

Trasposos. El módulo registrará los trasposos de las diferentes cuentas fuentes de FNM a las cuentas llamadas concentradoras del mismo organismo. El usuario necesita ingresar la clave de la cuenta bancaria origen y destino, el importe a traspasar, el número de cheque o el número de transacción que otorga el banco para registrar el movimiento, y el sistema desplegará las claves que intervengan en el movimiento, además automáticamente se desplegará un número que identifique el movimiento dentro del sistema (SIITES).

Ingresos. El módulo registrará los ingresos que el organismo tiene, por ejemplo el subsidio que recibe del gobierno, los productos propios de su operación, recuperación por indemnizaciones de accidentes e intereses generados por inversiones y automáticamente se desplegará un número que identifique el movimiento dentro del sistema (SIITES).

Ajustes. Este módulo se encarga de registrar los ajustes resultado de eventos, por ejemplo la cancelación de un cheque que implique la recuperación de un egreso y por tanto la afectación de saldos en forma inversa es decir, las cuentas que en un principio se cargaban, ahora se abonan y las que se abonaban se cargan. El usuario ingresará la cuenta bancaria y el importe a ajustar, también ingresará las cuentas contables, y el sistema automáticamente afectará saldos y proporcionará un número consecutivo que identifique a cada ajuste.

2.5.3 Explotación de información.

El sistema (SIITES) contará con diferentes módulos destinados a emitir reportes que impliquen rapidez y veracidad en la información que ellos contengan. Se reportará información de gran importancia para la misma Gerencia de Tesorería como lo es la emisión de la póliza de caja y de los diferentes boletines que respaldan dicha póliza. También reportará estadísticos de las diferentes cuentas bancarias desglosadas por cuenta bancaria y contable.

2.6 Análisis costo/beneficio

Dentro del análisis costo/beneficio se requiere la relación de una evaluación técnica del hardware y software, buscando las opciones más adecuadas con base en las necesidades del sistema; sin embargo, en esta ocasión, la asignación de recursos tanto de hardware y software ya está establecida por la institución.

Razón por la cual se utilizará como hardware equipo HP-9000 y en cuanto a software INFORMIX bajo ambiente UNIX.

2.6.1 Costo de desarrollo

- En esta etapa es importante una evaluación técnica y económica en cuanto a recursos humanos, necesarios para el desarrollo del (SIITES).
- De acuerdo a los requerimientos y necesidades del (SIITES) se estima un tiempo de desarrollo distribuido de la siguiente manera:

ETAPA	TIEMPO
- Estudio de viabilidad y analisis	1 mes
- Diseño	1 mes
- Programación	3 semanas
- Pruebas	1 semana
- Implantación, capacitación y liberación	2 semanas
Tiempo total de desarrollo	<u>14 semanas</u>

- El personal requerido para este desarrollo, así como sus sueldos mensuales es el siguiente:

PERSONAL	SUELDO	TIEMPO
1 Coordinador del proyecto	\$9,000.00	14 semanas
1 Analista	\$7,000.00	14 semanas
3 Programadores	\$6,000.00	6 semanas

- De lo anterior se concluye que para el desarrollo el tiempo será de 14 semanas con un costo aproximado de \$83,000.00.

2.6.2 Costo de operación

a) Nómina de la Gerencia de Tesorería en condiciones actuales

El número de personas que intervienen en la atención de las funciones de tesorería, separadas por áreas son las siguientes:

Caja General

1 Cajero General	\$14,000.00
1 Jefe de unidad	\$7,000.00
2 Secretarías	\$5,000.00
Subtotal	<u>\$26,000.00</u>

Subjefatura de Operaciones Bancarias e Inversiones.

1 Subjefe	\$12,000.00
1 Jefe de unidad	\$7,000.00
Subtotal	<u>\$19,000.00</u>

Subjefatura de Recursos Financieros

1 Subjefe	\$12,000.00
2 Jefes de unidad	\$14,000.00
1 Empleado de cheques	\$2,500.00
1 Secretario	\$2,500.00
1 Dependiente de comprobantes	\$2,500.00
Subtotal	<u>\$33,500.00</u>

Subjefatura Departamento Coordinación Bancaria

1 Subjefe	\$12,000.00
1 Jefe de Unidad	\$7,000.00
1 Jefe de sección	\$5,000.00
Subtotal	<u>\$24,000.00</u>
Total	<u>\$97,500.00</u>

b) Costo esperado de la nómina de la Gerencia de Tesorería con el apoyo del sistema (SIITES)

Con el (SIITES) las actividades de pago, registros y movimientos de bancos se realizarán de acuerdo a los siguientes puntos, proponiendo una solución a la situación actual y que permita la comunicación con el sistema de Presupuesto (SIPTAL) y el sistema de Nóminas (SIIPAD) :

- Registro de documentos para pago
- Registro de movimientos bancarios
- Emisión de productos

Una vez instalado el sistema (SIITES) se espera un ahorro en recursos humanos con la siguiente estimación:

Caja General

1 Cajero General	\$14,000.00
1 Secretaria	\$2,500.00
Subtotal	<u>\$16,500.00</u>

Subjefatura de Operaciones Bancarias

1 Subjefe	<u>\$12,000.00</u>
-----------	--------------------

Subjefatura de Recursos Financieros

1 Subjefe	\$12,000.00
1 Jefe de unidad	\$7,000.00
1 Secretario	\$2,500.00
Subtotal	<u>\$21,500.00</u>

Subjefatura Departamento Coordinación Bancaria

Subjefe	\$12,000.00
Jefe de Unidad	\$7,000.00
Subtotal	<u>\$19,000.00</u>
Total	<u>\$69,000.00</u>

2.6.3 Comparación operación manual vs. Operación automatizada

De lo anterior se concluye que se tiene un ahorro de \$28,500.00 de la nómina actual de la Gerencia de Tesorería, con lo cual el costo del sistema se pagaría en solo tres meses.

Si se toma en cuenta que el promedio de inversiones diarias son del orden de hasta \$117,000,000.00 y con una tasa anual de 28 %, se obtiene una ganancia en intereses de \$91,000.00 diarios, los cuales podrían perderse de no contar con un sistema rápido veraz y oportuno, lo cual justifica por demás la necesidad y el costo de desarrollo del sistema propuesto.

Además se puede notar que con tan solo un día de inversión se cubre el monto total del costo del desarrollo del sistema.

Finalmente y quizás invaluable podría citarse la calidad de la información en cuanto a su veracidad, confiabilidad, disponibilidad y oportunidad.



CAPÍTULO III

CAPÍTULO III ANÁLISIS**3.1 Recursos****3.1.1 Hardware**

De igual forma que en el (PIDIA) se consideró la necesidad de definir un modelo y, dentro de lo posible, un estándar de equipamiento de las áreas que desarrollan todas las funciones administrativas dentro de (FNM)

La situación del equipo de cómputo de la Coordinación de Descentralización de Informática a inicios de 1991 era la siguiente:

Se contaba con 77 equipos PC/XT de las siguientes características:

- Procesador 8086.
- Memoria principal (RAM) de 640 KB.
- Unidad de disco duro de 30 MB.
- Unidad de disco flexible de 5 ¼" de 360 KB.
- Monitor monocromático de 12".
- Además, se contaba con 77 impresoras de la marca ATI, Modelo 2330.

En el año de 1990 se realizó la adquisición de 6 minicomputadoras HP-9000/835 de las siguientes características:

- Procesador PA-RISK de HP.
- Memoria RAM de 24 MB.
- Disco Duro de 1.964 GB.
- Multiplexor de puertos con capacidad para 32 terminales con conexión Serial.
- Consola de operación HP-700/92.
- 32 Terminales Modelo HP-700/44.
- Unidad de cinta de carrete abierto HP Modelo 7980A.
- Sistema Operativo Unix HP-UX versión 7.0.
- Impresora de Matriz de puntos HP Modelo 2567C de 1600 LPM.

Los equipos HP-9000 se encuentran ubicados uno en cada sede regional y otro en oficinas centrales. Estos equipos fueron instalados y puestos en operación hasta el mes de Abril de 1992 por personal de la (CDI)

Detectando la problemática a continuación se presentan las posibles alternativas de solución y la elección de una de ellas para su implantación:

- Existencia de equipo obsoleto como equipos PC/XT.

- Carencia de programas de mantenimiento y actualización del equipo.
- Falta de políticas para la administración de operación del equipo.
- Falta de procedimientos para la evaluación de equipos adquiridos y por adquirirse.

Como consecuencia de la situación antes descrita y para dar cumplimiento al tercer punto del (PIDIA) se plantearon lineamientos que se enuncian a continuación:

- Definición de las características de los equipos de cómputo con los que debe contar la empresa para una plena modernización en sus áreas informáticas y administrativas que necesiten de una computadora para su integración en la sistematización distribuida a nivel central y a nivel sede regional.
- Definición de las normas y políticas de instalación, seguridad y adquisición de los equipos.
- Determinación del tipo de equipos (modelo de equipamiento) de cómputo para el manejo de información distribuida de (FNM)

3.1.2 Software

Para establecer una solución a la problemática detectada, se hizo necesaria una evaluación comparativa de los diferentes productos de software para desarrollo existentes en el mercado. Con la participación directa tanto del personal de la Coordinación de Descentralización Informática, como los Departamentos de Informática de las Sedes Regionales con experiencia en el uso de Manejadores de Bases de Datos, para homogeneizar la información, se realizaron las siguientes actividades para la selección de proveedores y productos.

- Se realizó una convocatoria a diversas compañías especializadas en el área de software a quienes se les plantearon las necesidades de la empresa, donde se mostraron las características del equipo de cómputo que se quiera adquirir.
- La selección del software de las compañías se hizo considerando los productos más conocidos en el mercado a nivel nacional e internacional y que reunían las siguientes características:
 - Estar conceptualizado bajo el esquema de "Manejador de Bases de Datos Relacionales".
 - Soportar un lenguaje estructurado en el estándar SQL.
 - Brindar facilidades para el manejo de aplicaciones en ambiente cliente/servidor.
 - Ser compatible con la plataforma con la que se va a contar.
- El proveedor debe garantizar el soporte técnico, capacitación, así como tener amplia cobertura para asegurar la entrega del producto eficazmente a nivel nacional.

- Una vez integrada toda la información, se realizaron una serie de visitas a empresas donde se manejaban los diversos productos, posteriormente se evaluó su facilidad de aprendizaje, uso, confiabilidad, consumo de recursos, transportabilidad, calidad de la capacitación y servicios de soporte técnico.

Como resultado de las visitas y en base a los requerimientos de (FNM) se consideró a los manejadores *ORACLE*, *PROGRESS* e *INFORMIX* como únicos candidatos elegibles, de lo cual se concluyó lo siguiente:

ORACLE: Necesita de una infraestructura muy grande para funcionar óptimamente.

PROGRESS: No contiene interfaces naturales con hojas de cálculo y graficadores que se utilicen de apoyo al desarrollo de sistemas.

INFORMIX: Presenta una gama de herramientas de ayuda para agilizar el desarrollo e implantación de los sistemas.

Como conclusión final, después de las entrevistas y conocer las ventajas y desventajas técnicas de los productos, se determinó que el Manejador de Bases de Datos *INFORMIX* era el que más se apegaba a las necesidades de la empresa, cumpliendo con las características esenciales para su modernización y desarrollo a nivel nacional; esta decisión marcaría la pauta para el establecimiento del estándar institucional en la construcción e interacción de sistemas de aplicación administrativa en (FNM)

INFORMIX. Es un manejador de bases de datos que integra un SQL estándar y un lenguaje de cuarta generación (*INFORMIX 4GL*), los cuáles pueden ser utilizados en los ambientes de los sistemas operativos *UNIX*, *ULTRIX* y *MS-DOS*, corriendo de manera natural en equipos *HP-9000*, *PC'S AT*, aún en otras plataformas con que no cuenta (FNM)

INFORMIX permite tanto el proceso distribuido como el manejo de bases de datos distribuidas, así como también permite que la información de la base de datos sirva, para realizar aplicaciones de hojas de cálculo, graficadores y procesamiento de imágenes.

A continuación se presenta un cuadro comparativo entre los manejadores *ORACLE*, *PROGRESS* e *INFORMIX*.

CONCEPTO EN ORDEN DE IMPORTANCIA	POSICIÓN OBTENIDA EN EL ESTUDIO		
	1a.	2a.	3a.
Soporte a nivel nacional	Informix	Oracle	Progress
Adecuado a características del equipo actual	Progress Informix	Oracle	
Calidad del lenguaje de 4a. generación	Progress Informix	Oracle	
Menor costo situación actual	Informix	Oracle	Progress
Menor costo situación propuesta	Informix	Oracle	Progress
Incorporación en planes de estudios superiores	Informix	Oracle	Progress
Consumo de recursos	Informix	Oracle	Progress
Facilidad de comprensión	Progress	Informix	Oracle
Calidad de la capacitación	Informix Progress	Oracle	

Cuadro comparativo de lo manejadores de Bases de Datos

3.2 Diagrama de contexto

El diagrama de contexto representa varias características importantes del sistema:

- Los Terminadores que representan las personas, organizaciones y sistemas con los que se comunica el sistema.
- Los datos que el sistema recibe del mundo exterior y que deben procesarse de alguna forma.
- Los datos que el sistema produce y que se envían al mundo exterior.
- Los almacenes de datos que el sistema comparte con los terminadores. Estos almacenes de datos se crean fuera del sistema para su uso, o bien son creados en él y usados fuera.
- La frontera entre el sistema y el resto del mundo.

En este caso (SIITES) mantiene una estrecha relación con los siguientes sistemas externos:

- **SIPTAL:** Sistema Integral de Información Presupuestal. Este proporciona toda la información necesaria que se genera en la Gerencia de Presupuestos.
- **SIIPAD:** Sistema Integral de Información de Pago Descentralizado. Este proporciona los importes quincenales que son pagados a todos los empleados que laboran en FNM.

A su vez atiende o se relaciona con los siguientes terminadores:

- Áreas
- Bancos
- Externos

Los datos de entrada son la información que se obtiene del (SIPTAL) (devengado), dentro del (SIITES) se registra, y una vez que el documento se paga se obtiene como salida el documento con status de pagado que afectará al (SIPTAL). De las áreas llegan documentos urgentes para pago, y al igual que con el documento que llega de (SIPTAL), una vez que se paga se manda un informe del documento pagado. De los bancos se obtienen cotizaciones y saldos bancarios con los que el (SIITES) podrá hacer las inversiones, traspasos o consultar el disponible de las cuentas bancarias de FNM para los compromisos de pago que ésta tiene. Por último se obtiene la información del pago quincenal del personal que labora en FNM para hacer los movimientos necesarios de las cuentas donde se toma el dinero y así generar el pago.

A continuación se muestra en la figura 3.1 el diagrama de contexto del (SIITES).

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE CONTEXTO

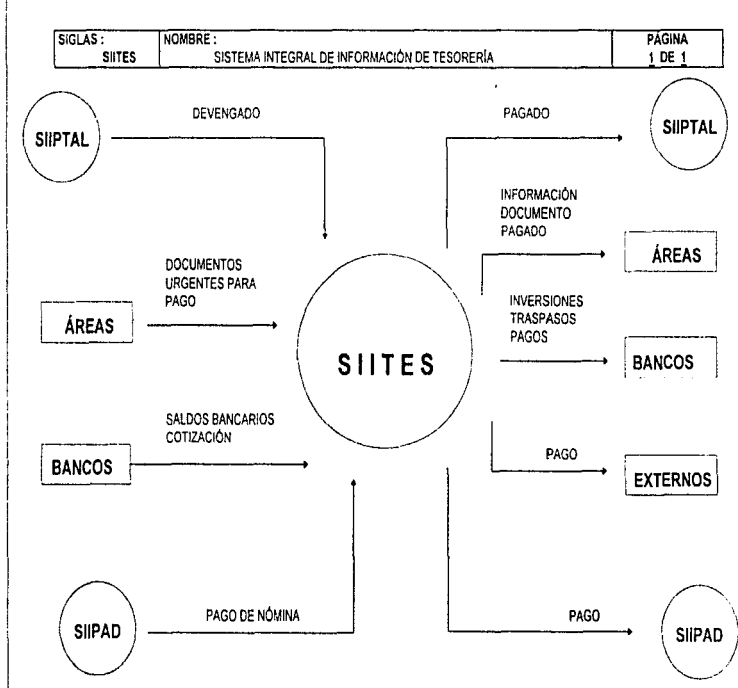


FIGURA 3.1

3.3 Especificación de requerimientos

- Almacenar la información de los documentos referente a los ingresos, clasificando el origen de los mismos, ya sea por subsidio del gobierno, por operación de los ferrocarriles e intereses generados por inversiones del remanente diario.
- Emisión de los informes de ingresos diarios.
- Almacenar la información de los egresos diarios, como son pagos, depósitos, transferencias a cuentas, etc.
- Producir los reportes de los egresos generados en el día junto con los informes que se envían a las diferentes áreas.
- Almacenar la información de las inversiones efectuadas en el día junto con los movimientos de traspaso de las cuentas receptoras hacia las cuentas concentradoras, así como también la cotización del día para el cálculo de los intereses.
- Emitir informes de los movimientos de cuentas, traspasos, inversiones y saldos.
- Realizar un control adecuado de las chequeras tanto de generación manual, como de generación automática.
- Llevar un control de acceso en todo el sistema, esto significa asignar niveles de acceso a ciertos módulos del sistema debido a que es información de mucha importancia para el organismo.
- Permitir el enlace del (SIITES) con otros sistemas del (SIID) (Sistema Integral de Información Distribuida), como son los sistemas SIIPAD (Sistema Integral de Pago Descentralizado), y (SIIPAL) (Sistema Integral de Información Presupuestal).

3.4 Diagrama de flujo de datos

Para representar el flujo de información de un sistema se aplican los Diagramas de Flujo de Datos (DFD's).

3.4.1 Definición de un DFD.

Un DFD es un conjunto de símbolos o componentes gráficos. Cada uno de ellos representa una cierta función dentro del sistema; las entidades involucradas; las actividades (procesos) que se realizan y los flujos de información entre éstos.

3.4.2 Clasificación de DFD's.

A la versión más simple de DFD se le conoce como "Diagrama de Contexto", en el sólo aparece un proceso y todos los flujos que representan las entradas y salidas del sistema, así como las entidades, productores y consumidores de la información de los procesos principales y los flujos que los interconectan.

A la primera derivación de un DFD de Contexto se le conoce como "Diagrama Conceptual" o "Diagrama de Nivel Cero (0)", en este diagrama, hacen su aparición las entidades, productores y consumidores de la información.

A las versiones resultantes, al descomponer o detallar a un "DFD Conceptual", se les conoce como "Diagrama de Nivel Uno (1)". Este a su vez se puede descomponer y se producen "Diagramas de Nivel Dos (2)". Así sucesivamente hasta llegar al "Diagrama de Nivel 'ene' (n)".

Cuando un proceso ya no es posible detallarlo o descomponerlo se le conoce como "Primitiva" o "Miniespecificación".

A continuación se presentan los Diagramas de Flujo del (SIITES). Ver figura 3.2 a 3.6.

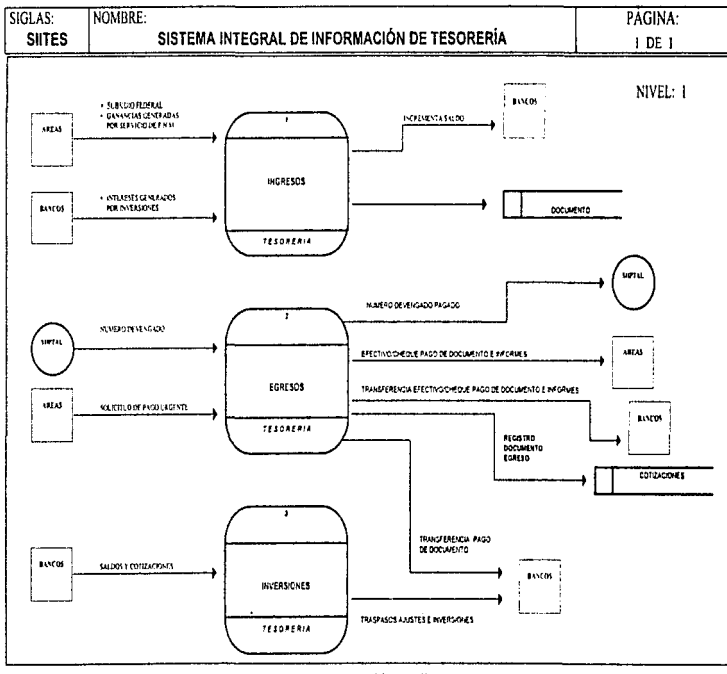


FIGURA 3.2

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

SIGLAS: SIITES	NOMBRE: SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PAGINA: 1 DE 1
--------------------------	--	-------------------

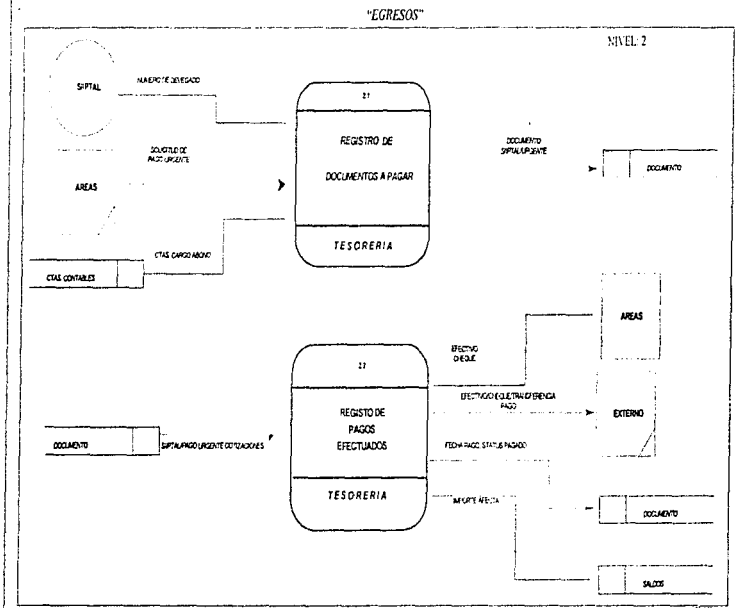


FIGURA 3.3

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

SIGLAS: SIITES	NOMBRE: SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PÁGINA: 1 DE 1
--------------------------	--	--------------------------

* REGISTRO DE PAGOS EFECTUADOS *

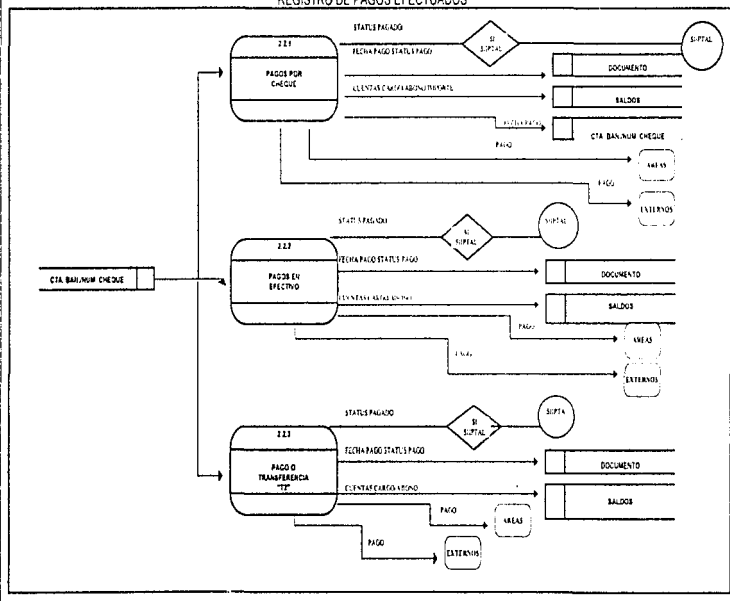


FIGURA 3.4

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

SIGLAS: SITES	NOMBRE: SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PAGINA: I DE I
-------------------------	--	--------------------------

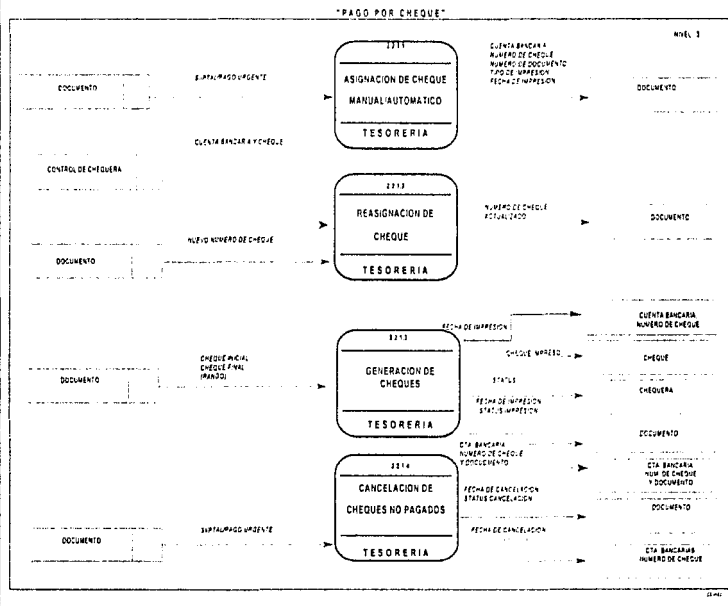


FIGURA 3.5

SIGLAS: SITES	NOMBRE: SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PAGINA: 1 DE 1
------------------	---	-------------------

* INVERSIONES *

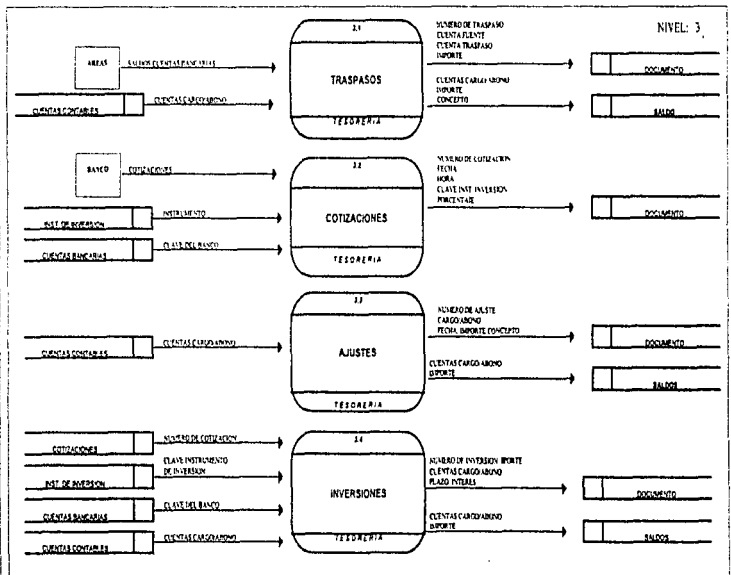


FIGURA 3.6

3.5 Diagrama de flujo operativo

3.5.1 Definición de un Diagrama de Flujo Operativo (DFO).

Un DFO es una herramienta para la representación gráfica del flujo de actividades, formada por un conjunto de columnas en donde se esquematiza el flujo de las actividades necesarias para el cumplimiento de una función, cada columna representa a la entidad o instancia responsable de la actividad.

Cada actividad puede estar representada en forma gráfica, y/o reforzada por una leyenda descriptiva.

3.5.2 Componentes de un DFO.

- Columnas: Deben tener un título indicando el nombre de la entidad o instancia responsable del desarrollo de ciertas actividades.
- Actividades: Son figuras representativas de las acciones que debe realizar una entidad. Pueden ser bloques rectangulares, con texto, con figuras o con texto y figura.
- Flujos: Son flechas que indican la ruta a seguir entre actividades, desde el inicio hasta el término de las mismas.

3.5.3 Recomendaciones generales para la elaboración de DFO.

- Los flujos, preferentemente, deberán ser: horizontales de izquierda a derecha y verticales de arriba a abajo.
- Las actividades que sean paralelas deben estar al mismo nivel.
- Para pasar de una página a otra utilice conectores de página.

A continuación se presentan los diagramas de flujo operativo del (SITES). Ver figuras 3.7 a 3.17.

SIGLAS: SIITES	NOMBRE: SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PAGINA: 1 DE 1
MODULO: EGRESOS		FUNCION: PAGO DE COMPROMISOS

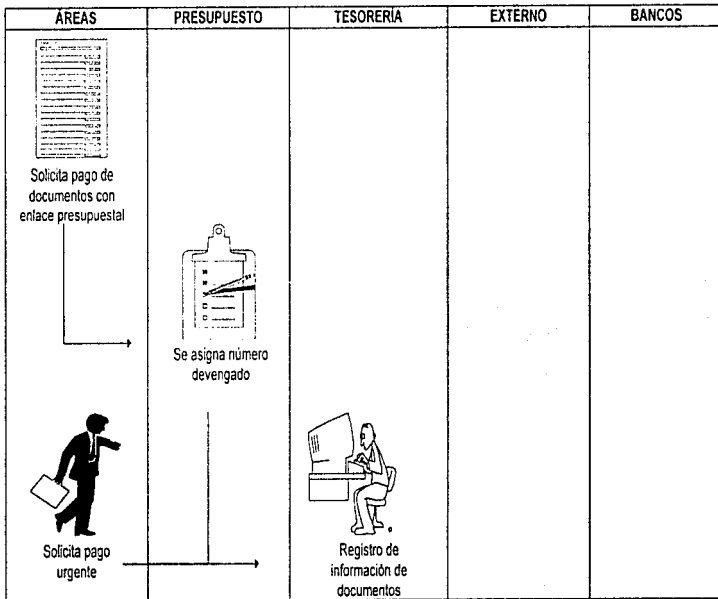


FIGURA 3.7

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

SIGLAS:	NOMBRE:	PAGINA:
SIITES	SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	2 DE 1
MODULO: EGRESOS		FUNCION: PAGO DE COMPROMISOS

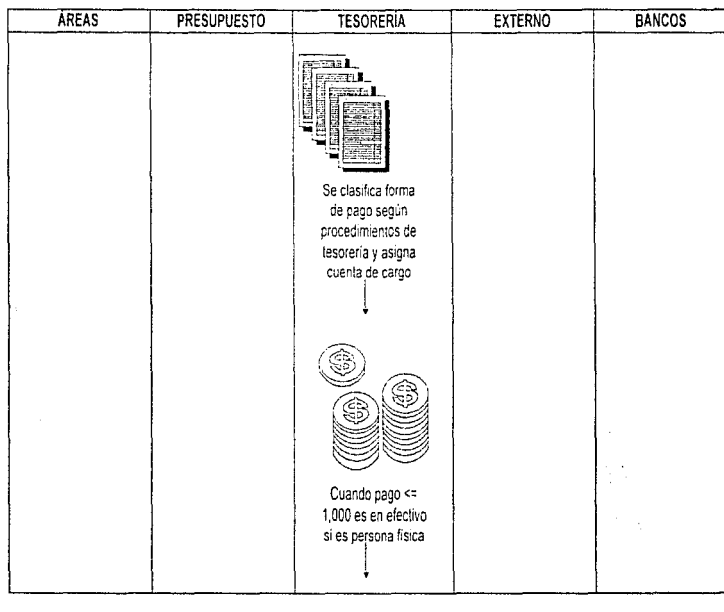


FIGURA 3 B

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

SIGLAS: SIITES	NOMBRE: SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PAGINA: 3 DE 1
MODULO: EGRESOS		FUNCIÓN: PAGO DE COMPROMISOS

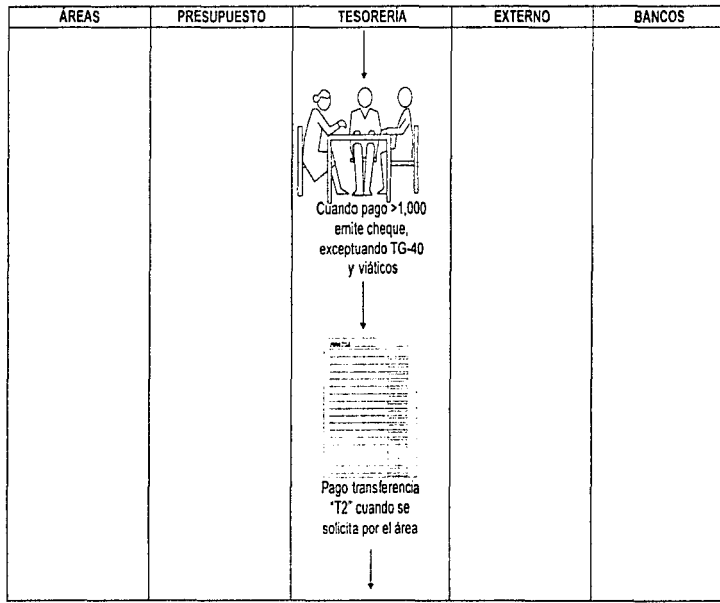


FIGURA 3.9

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

SIGLAS: SITES	NOMBRE: SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PAGINA: 4 DE 1
MODULO: EGRESOS		FUNCIÓN: PAGO DE COMPROMISOS

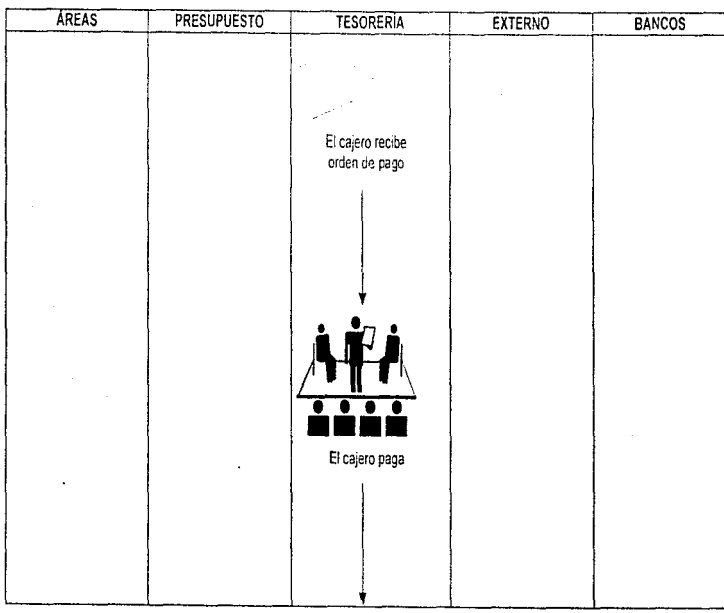


FIGURA 3.10

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

SIGLAS: SIITES	NOMBRE: SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PAGINA: 5 DE 1
MODULO: EGRESOS		FUNCIÓN: PAGO DE COMPROMISOS

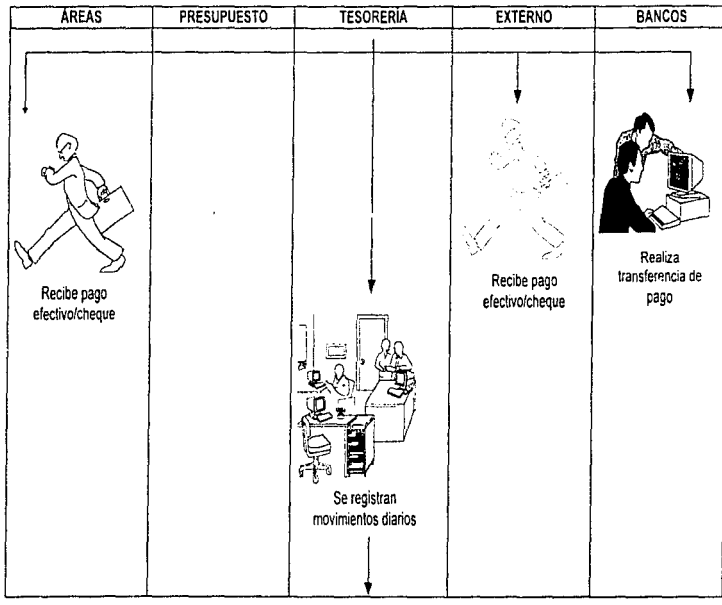


FIGURA 3.11

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

SIGLAS:	NOMBRE:	PÁGINA:
SIITES	SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	6 DE 1
MODULO: EGRESOS		FUNCIÓN: PAGO DE COMPROMISOS

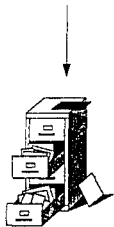
ÁREAS	PRESUPUESTO	TESORERÍA	EXTERNO	BANCOS
		 <p style="text-align: center;">Se afectan saldos</p>		

FIGURA 3.12

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

SIGLAS: SIITES	NOMBRE: SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PÁGINA: 1 DE 1
MODULO: INGRESOS		FUNCIÓN: REGISTRO DE INGRESOS

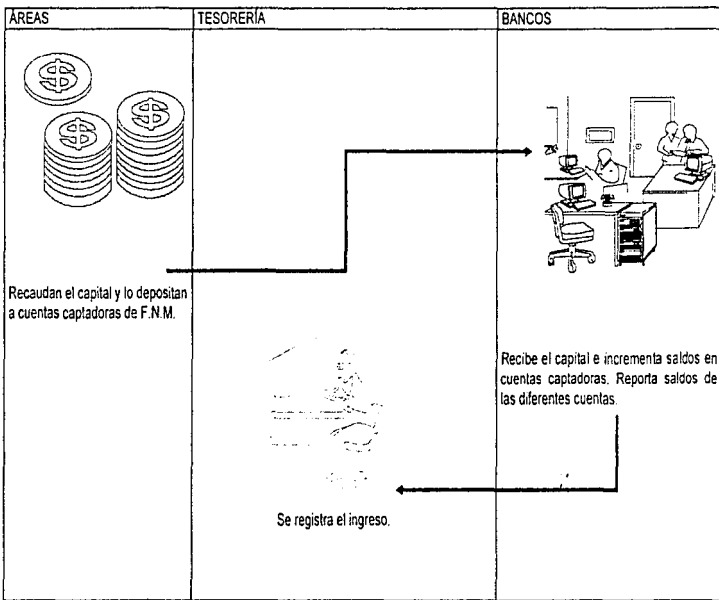


FIGURA 3.13

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

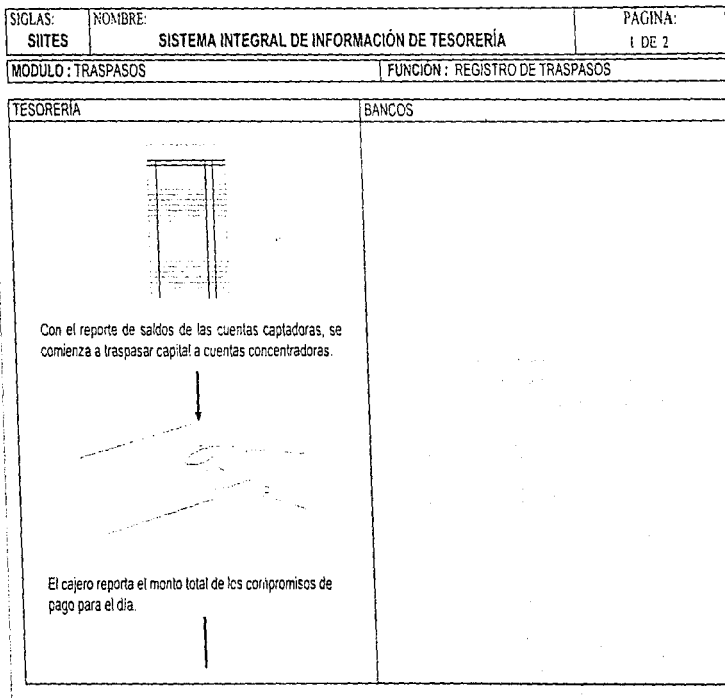


FIGURA 3.14

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

SIGLAS:	NOMBRE:	PAGINA:
SIITES	SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	2 DE 2

MODULO : TRASPASOS	FUNCIÓN : REGISTRO DE TRASPASOS
--------------------	---------------------------------

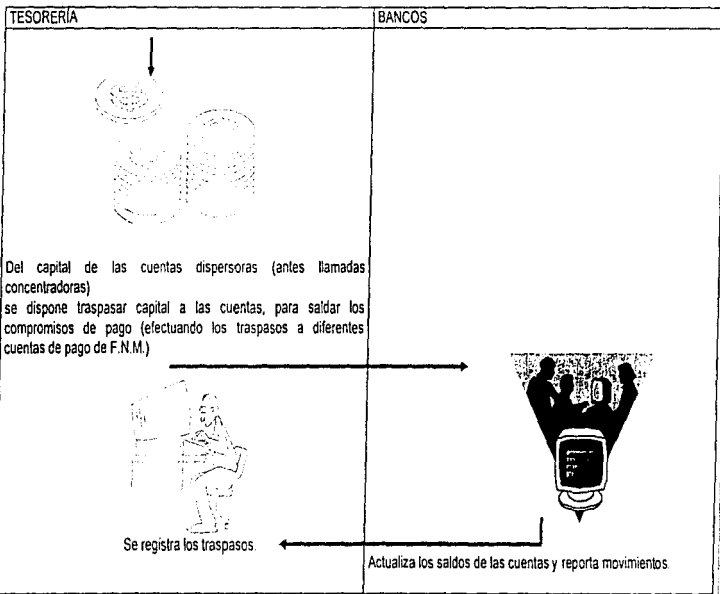


FIGURA 3.15

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

SIGLAS: SITES	NOMBRE: SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PAGINA: I DE 2
-------------------------	--	-------------------

MODULO : INVERSIONES	FUNCIÓN : REGISTRO DE INVERSIÓN
----------------------	---------------------------------

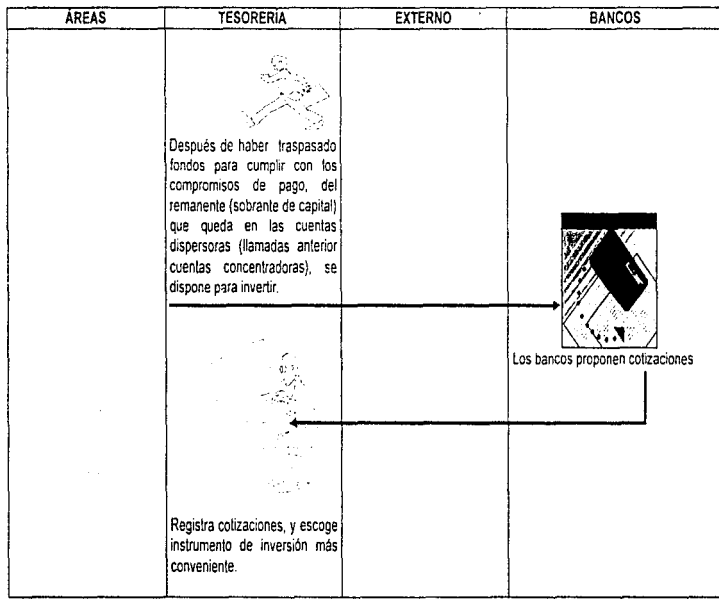


FIGURA 3.16

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

DIAGRAMA DE FLUJO DE DATOS

SIGLAS: SITES	NOMBRE: SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PAGINA: 2 DE 2
------------------	---	-------------------

MODULO: INVERSIONES	FUNCION: REGISTRO DE INVERSIÓN
---------------------	--------------------------------

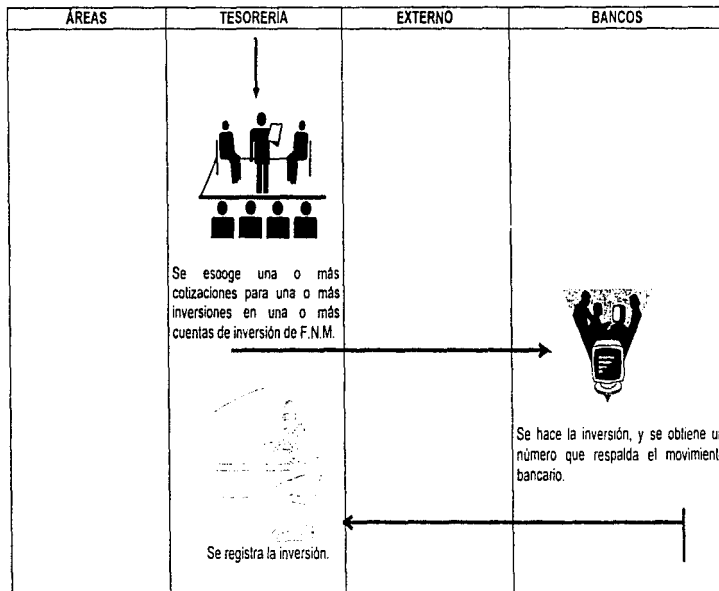


FIGURA 3.17

3.6 Árbol de menús

Para representar gráficamente el árbol de menús se esquematizará en un rectángulo el nombre del sistema y la descripción de sus módulos. Las funciones de cada módulo se especificarán en forma de lista vertical.

En la figura 3.18 se muestra el árbol de menús del (SITES).

SIGLAS: SIITES	NOMBRE: SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PÁGINA: 1 DE 1
-------------------	---	-------------------

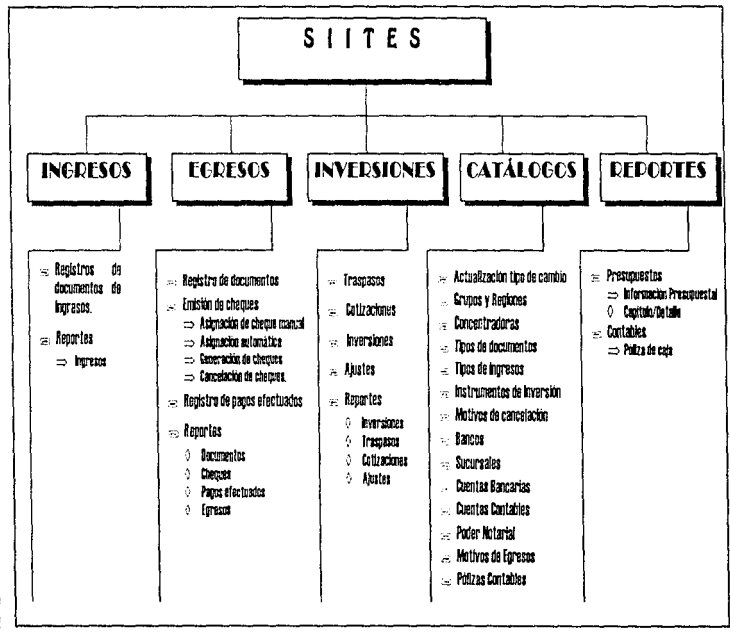


FIGURA 3.18



CAPÍTULO IV

CAPÍTULO IV DISEÑO

4.1 Conceptos Básicos

- **Bases de datos.** Es una colección de datos interrelacionados, almacenados juntos sin redundancia innecesaria, para servir a múltiples aplicaciones; los datos son almacenados de tal manera que son independientes de los programas que los usan, y además estructurados de tal manera que sirven para aplicaciones futuras.
- **Entidad.** Es una persona, un lugar, una cosa, un evento o un concepto acerca del cual se registra información.
- **Registro.** Es una colección de datos o valores tomados por atributos relacionados entre sí.
- **Atributo.** Es la unidad mínima o elemento de datos que describen a las entidades.
- **Llaves.** Existen dos tipos de llaves:
 - **Llave primaria.** Es un atributo o combinación de atributos que identifican como única una instancia de una entidad.
 - **Llave secundaria o foránea.** Es un atributo que debe hacer referencia a una llave primaria en otra entidad, para establecer relación entre entidades. Esta puede contener nulos y duplicados.
- **Diccionario de datos.** Es el diseño referente al uso, relaciones y significado de los datos. Y a la herramienta que da la posibilidad de controlar y manejar la información sobre los datos en las fases de diseño, implantación, operación y expansión de una base de datos, se llama diccionario de datos.
- **Sistema de Manejo de la Base de Datos (DBMS).** Un DBMS hace posible cesar datos integrados que cruzan límites operacionales, funcionales u organizacionales dentro de una empresa. Para reconocer el potencial de un DBMS debe satisfacer los siguientes requisitos:
 - Que el DBMS atienda de manera efectiva las diferentes funciones de la empresa.
 - Minimizar la cantidad de redundancia en los datos almacenados.
 - Suministrar información consistente al proceso que toma decisiones.
 - Aplicar controles de seguridad.
 - Desarrollar, mantener y mejorar los programas de aplicación de manera más rápida y económica, con menos personal calificado.
 - Facilitar la reorganización física de los datos almacenados.
 - Posibilitar el control centralizado de la base de datos.

- **Administrador de Base de Datos (DBA).** Es la persona responsable de administrar la base de datos. Se encarga de la definición de la estructura de almacenamiento, la creación de métodos de acceso apropiados, la concesión de autorización a los usuarios para el proceso a los datos y la especificación de personas autorizadas para efectuar ciertas operaciones.
- **Transacción.** Es una secuencia de operaciones que deben ser completadas totalmente o de lo contrario no se hace ninguna, la transacción no es meramente una protección contra errores, es una rutina que regresa todo a su estado anterior de los elementos involucrados al detectar un error lógico.
- **Integridad.** Las restricciones de integridad aseguran que la información no pueda ser borrada y que los datos insertados cumplan con las especificaciones definidas, ya que algunas tablas o entidades dependen de otras haciendo referencia a su información. Entre las restricciones de integridad disponibles se pueden mencionar las siguientes:
 - **Integridad de entidades.** Una entidad es el objeto fundamental que se registrará en la base de datos, la cual está representada por una tabla. Para identificar cada registro en la tabla, ésta debe tener una llave primaria (valor único de identificación del registro) a este requerimiento se le llama restricción de integridad de entidad.
 - **Integridad semántica.** Asegura que los datos de los registros reflejen un valor permitido dentro de un dominio, es decir, dentro de un conjunto de valores para esa columna. Para forzar la integridad semántica, se usan las siguientes restricciones:
 - ♦ Tipo de datos: Define los tipos de valores que se almacenarán en una columna.
 - ♦ Valores por default: Son los valores insertados en una columna cuando no se especifica el valor explícito.
 - ♦ Chequeo de constraint: Especifica condiciones en la inserción de datos a una columna, cada registro insertado en la tabla debe cumplir esas condiciones.
 - **Integridad referencial.** La integridad referencial es una dependencia lógica de una llave foránea sobre una llave primaria. Para preservar la integridad referencial, INFORMIX requiere que, para poder afectar el registro con la llave primaria, primero se deben afectar los registros que dependan del primero a través de una llave foránea.

La integridad referencial permite a los usuarios definir y forzar relaciones entre columnas. Por ejemplo, puede garantizarse mediante una

definición de integridad referencial que un registro de una tabla maestra no pueda ser borrado si existe información dependiente de éste registro en la tabla detalle.

- **Normalización.** Es un proceso de depuración de las relaciones y especificaciones de los archivos que forman el esquema, de tal manera que se garantice que las relaciones eviten en lo más posible la redundancia entre las diferentes entidades; y de esta manera lograr eficacia y eficiencia, de acuerdo a las aplicaciones.

La teoría de la normalización está basada en el concepto de formas normales, se dice que una relación está en una forma normal particular si satisface cierto conjunto específico de restricciones como las que a continuación se detallan.

Primera forma normal. Incluye la eliminación de grupos repetidos y la identificación de la llave que define al criterio primario.

Segunda forma normal. Asegura que todos los atributos no llaves sean completamente dependientes de la llave del criterio primario.

Tercera forma normal. Elimina cualquier dependencia transitoria, una dependencia transitoria es aquella en la cual sus atributos no llaves son dependientes de otros atributos no llave.

Cuarta forma normal. No se cumple con esta forma normal cuando una relación contiene dos o más datos multivaluados de una entidad.

Quinta forma normal. Establece que un registro no puede estar estructurado de tal forma que sus contenidos se puedan reconstruir a partir de registros mas pequeños. Por supuesto siempre es posible reestructurar un registro en otros más pequeños con el mismo campo clave. Este caso es una excepción de la restricción anterior.

Las formas normales se aplican secuencialmente, sobre un diseño de base de datos, primero debe comprobarse la primera, luego la segunda, seguida de la tercera, y así sucesivamente.

Las relaciones normalizadas tienden a tener un pequeño número de atributos. De hecho, si se siguen cuidadosamente todas las reglas de normalización, probablemente el diseñador acabará con muchas relaciones binarias. Los teóricos hacen un gran énfasis en procesos de normalización como herramienta para el diseño de una buena base de datos. El hecho es que la *sobrenormalización* del diseño puede degradar el rendimiento. La normalización debería utilizarse sólo como una de las herramientas para el diseño de bases de datos. El diseñador debe familiarizarse con el sistema y las necesidades de los usuarios finales. Las decisiones de agrupar campos deben tener en cuenta las necesidades de los usuarios. Las bases de datos estáticas, que no están sujetas a muchas actualizaciones,

empeoran su rendimiento si están sobrenormalizadas. Sólo se puede conocer el hecho de que una base de datos sea estática hablando con los usuarios finales.

La normalización, si se aplica ciegamente, consigue la eliminación de los datos redundantes. Aunque raramente, la redundancia de los datos es deseable y minimiza tiempos de acceso.

Como se mencionó en el Capítulo II, el desarrollo del sistema se hará bajo la metodología Yourdon, por esto, en la fase de análisis hablaremos de los Modelos de Bases de Datos.

- **Modelos de Bases de datos.** Un modelo de Bases de Datos se constituye con las herramientas que permiten describir la estructura de una base de datos. Estos modelos deben cumplir con la tarea de describir a los datos, sus relaciones y sus limitantes.

Existen 3 modelos de bases de datos:

Los modelos jerárquicos y de red han sido utilizados como estructuras básicas de los sistemas de manejo de base de datos hasta el principio de la década de los setenta, donde las características principales son que los datos se organizan por medio de un conjunto de árboles y las relaciones a través de apuntadores; mientras que en el modelo relacional las entidades y sus relaciones se representan con tablas bidimensionales, compuestas por diferentes atributos.

Por ser este último el modelo más reciente y con el cual se desarrollará el (SIITES), ya que está bajo la metodología Yourdon, describiremos los conceptos más importantes del modelo relacional.

El modelo relacional surgió a principios de la década de los setenta, con un nuevo planteamiento relacional en su simplicidad, es decir, la facilidad de comprensión por parte del usuario final. Los usuarios finales no se tienen que preocupar por la estructura de almacenamiento físico; se les puede orientar hacia el contenido de la información de sus datos.

El concepto matemático sobre el que está basado el modelo relacional es el de "relación" de la teoría de conjuntos. Una relación es un subconjunto del producto cartesiano de una lista de dominios. Un dominio es simplemente un conjunto de valores.

– Ventajas del modelo relacional

Simplicidad. Sus solicitudes se formulan en términos del contenido de la información y no reflejan la complejidad de los aspectos relacionados con el sistema.

Consultas no planeadas. Debido a que no hay una dependencia de posición entre las relaciones, las consultas no tienen que reflejar ninguna estructura preferida y por lo tanto pueden ser de tipo procedimientos no estándar de consulta.

Independencia de los datos. Esto debe constituir uno de los principales objetivos de cualquier sistema de manejo de bases de datos. El modelo relacional elimina los detalles relativos a la estructura del almacenamiento y la estrategia del acceso de la interfase con el usuario. Este modelo proporciona un grado de independencia de los datos relativamente más alta que los otros dos modelos antes mencionados.

Fundamentos teóricos. El modelo teórico está basado en la bien desarrollada teoría matemática de las relaciones. El riguroso método del diseño de una base de datos usando la normalización, da a este modelo un fundamento sólido. Esta clase de fundamento no existe para los otros modelos.

El campo del diseño estructurado ofrece guías para ayudar al diseñador a determinar los módulos, y sus interconexiones, que mejor realizarán los requerimientos especificados por el analista. Las dos reglas más importantes son las referentes al acoplamiento y la cohesión.

Cohesión. Grado en el cual los componentes de un módulo (típicamente las instrucciones individuales que conforman un módulo) son necesarias y suficientes para llevar a cabo una sola función bien definida. En la práctica, esto significa que el diseñador debe asegurarse de no fragmentar los procesos esenciales en módulos, y también de no juntar procesos no relacionados (que se representan por burbujas en DFD) en módulos sin sentido. Los mejores módulos son aquellos que son funcionalmente cohesivos (es decir, módulos en los cuales cada instrucción es necesaria para poder llevar a cabo una sola tarea bien definida) Los peores módulos son los que son coincidentemente cohesivos (es decir, aquellos cuyas instrucciones no tienen una relación significativa entre uno y otro)

Acoplamiento. Grado en el cual los módulos se interconectan o se relacionan entre ellos. Entre más fuerte sea el acoplamiento entre módulos en un sistema, más difícil es implantarlo y mantenerlo, pues entonces se necesitará un estudio cuidadoso para la modificación de algún módulo o módulos. En la práctica, esto significa que cada módulo debe tener interfaces sencillas y limpias con otros, y que se debe compartir un número mínimo de datos entre módulos. También significa que un módulo dado no debe modificar la lógica interna o los datos de algún otro módulo; lo que se conoce como una conexión patológica.

Tamaño del módulo. De ser posible, cada módulo debe ser lo suficientemente pequeño como para caber en una sola página (o para que pueda desplegarse en una sola pantalla). Desde luego, a veces no es posible determinar que tan grande va a ser un módulo hasta haberlo escrito, pero las actividades iniciales de diseño a menudo darán al diseñador una buena pista de que el módulo va a ser grande y complejo. Si es así, debe partirse en uno o más submódulos.

Alcance del control. El número de subordinados inmediatos que un módulo administrador puede llamar se conoce como el alcance del control. Un módulo no debe poder llamar a más de una media docena de módulos de nivel inferior. La razón es evitar la complejidad: si el módulo tiene digamos, 25 módulos de nivel inferior, entonces probablemente contendrá tanta lógica compleja de programa (HACER-MIENTRAS anidadas, etc.) que nadie lo podrá entender. La solución es introducir un nivel intermedio de módulos administradores, como haría un administrador de una organización humana, así se ve en la necesidad de tratar de supervisar directamente a 25 subordinados.

Alcance del efecto/alcance del control. Esta regla sugiere que cualquier módulo afectado por el resultado de alguna decisión debe ser subordinado (aunque no necesariamente un subordinado inmediato) del módulo que toma la decisión. Es un tanto a la regla de administración que dice que cualquier empleado afectado por los resultados de la decisión de algún administrador (es decir, dentro del alcance del efecto de la decisión) debe estar dentro del alcance de control de el administrador (es decir trabajando entre la jerarquía de personas que se reportan con el administrador). Violar esta regla en un ambiente de diseño estructurado usualmente lleva paso innecesario de banderas y condiciones (lo cual incrementa el acoplamiento entre módulos), la toma redundante de decisiones o (en el peor de los casos) conexiones patológicas entre módulos.

4.2 Diagrama Entidad-relación

Entre las estrategias de diseño se utiliza el diagrama de entidad-relación u otras formas de diagramas de estructura de datos como punto de partida para obtener el diagrama de estructura apropiado.

Todos los sistemas almacenan y usan información acerca del ambiente en el cual interactúan; a veces, la información es mínima, pero en la mayoría de los sistemas actuales es bastante compleja. No sólo deseamos conocer en detalle que información hay en cada agregado de datos, sino que también queremos conocer la relación que existe entre agregados. Este aspecto del sistema no es resultado por el diagrama de flujo de datos, pero si lo hace el diagrama de entidad-relación.

El diagrama de entidad-relación consta de cuatro elementos principales:

- **Entidades.** Se representan por medio de un rectángulo en el diagrama. Esto representa una colección o conjunto de objetos (cosas) del mundo real cuyos miembros juegan algún papel en el desarrollo del sistema.
- **Atributos.** Representan las características de los objetos que participan en el modelo de datos, generalmente columnas de tablas.
- **Relaciones.** Son líneas que representan la asociación que existe entre dos o más entidades de un modelo de datos. Una relación puede a su vez llegar a ser una entidad misma, por lo que incluso puede tener atributos.
- **Ligas.** Son líneas que conectan a las relaciones con sus entidades y estas con sus atributos y sirven para representar la cardinalidad de las relaciones.

En la figura 4.1 se muestra el diagrama de entidad-relación del (SIITES).

4.3 Diagrama de flujo de control

En un diagrama de flujo de datos se describen simplemente los conductores a lo largo donde viajan los paquetes de datos entre procesos y almacenes. Similarmenre, las burbujas en los DFD pudieran considerarse como procesadores de datos. Para una amplia clase de sistemas, sobre todo de negocios, existen sólo dos tipos de flujos necesarios en el modelo del sistema. Pero para otra clase de sistemas, los de tiempo real, necesitamos alguna manera de modelar flujos de control. Y se requiere una manera de mostrar procesos de control (esto es, burbujas cuya única labor es de coordinar y sincronizar las actividades de otras burbujas del DFD).

Un flujo de control puede imaginarse como un conducto que porta una señal binaria (esto es, está encendido o está apagado), el flujo de control se manda de un proceso a otro (o de un terminador externo a un proceso).

Un proceso de control puede considerarse como una burbuja supervisora o ejecutiva, cuya labor es coordinar las actividades de otras burbujas en el diagrama; sus entradas y salidas consisten sólo de flujos de control. Los flujos de control salientes del proceso de control se utilizan para despertar a otras burbujas; los flujos de control entrantes generalmente indican que una de las burbujas ha terminado su labor o que se ha presentado alguna situación extraordinaria, de la cual necesita informarse a la burbuja de control. Por lo tanto sólo hay un proceso de control de estos en un DFD dado.

Sin embargo, el comportamiento interno de un proceso de control es diferente; aquí es donde el comportamiento dependiente del tiempo del sistema se modela con detalle. El interior del proceso de control se modela con un diagrama de transición de estados, que muestra los varios estados en los que se puede encontrar todo el sistema y las circunstancias que lo llevan a cambiar de estado.

4.4 Especificación de procesos

La especificación de procesos, es la descripción de qué es lo que sucede en cada burbuja primitiva de nivel más bajo en un DFD. La especificación de procesos define lo que debe hacerse para transformar entradas en salidas. Es una descripción detallada de la política de negocios del usuario que cada burbuja lleva a cabo. Existen una variedad de herramientas que se pueden utilizar para la especificación de procesos: tablas de decisiones, lenguaje estructurado, pre/pos condiciones, diagramas de flujo, diagramas de Nassi/Shneiderman, etc. Es importante tener presente que en cualquiera de estas herramientas se deben considerar dos requerimientos cruciales:

- La especificación del proceso debe expresarse de una manera que puedan verificar tanto el usuario como el analista. Precisamente por esta razón se evita utilizar el lenguaje narrativo como herramienta de especificación.

- El proceso debe especificarse en una forma que pueda ser comunicada efectivamente al público amplio que este involucrado. A pesar de que el analista es típicamente quien describe la especificación del proceso, habitualmente será un público bastante diverso de usuarios, administradores, auditores, personal de control de calidad y otros, el que leerá la especificación del proceso. Una especificación pudiera expresarse tal vez con un cálculo de predicados, o en Pascal, o en enfoque de diagramación formal.

La mayoría de los analistas se inclinan por utilizar el lenguaje estructurado como método favorito para escribir especificación de procesos, pero también se puede utilizar una combinación de herramientas de especificación.

Como conclusión de una especificación de procesos se dice que representan la mayor parte del trabajo detallado que se tiene en la construcción de un modelo de sistemas; también la actividad de escribir especificaciones de proceso sirve como "prueba de cordura" para los DFD que ya se hallan desarrollado.

Así, podría descubrirse que la especificación del proceso requiere flujos de datos de entrada o salida adicionales (es decir flujos que no aparecieron en el DFD).

Por lo tanto se pueden esperar cambios, revisiones y correcciones del modelo de DFD, basadas en el trabajo detallado de la escritura de las especificaciones del proceso.

PROCESO INGRESOS

El proceso de consiste en el registro de todos los ingresos que se tienen en FNM, que pueden ser generados por:

1. Subsidio del gobierno
2. Operación de los ferrocarriles
3. Recuperación por indemnizaciones
4. Intereses generados por inversión

INICIO

Asignar número de documento de ingreso según le corresponda

Registrar fecha automático del día

Registrar el tipo de documento según exista en catálogo

Si no existe en catálogo

desplegar "Clave errónea"

de otra manera

desplegar la descripción del documento

fin si

Registrar clave de región según exista en catálogo

Si no existe clave de región en catálogo

desplegar "clave errónea"
de otra manera
desplegar descripción de la región
fin si
Registrar número de cuenta bancaria
Si no existe cuenta bancaria en catálogo
desplegar "clave errónea"
de otra manera
desplegar la descripción del banco y la sucursal
fin si
Registrar clave del motivo de ingreso
Si no existe tipo de ingreso en catálogo
desplegar "clave errónea"
de otra manera
desplegar la descripción del motivo de ingreso
fin si
Si motivo de ingreso es inversión
encontrar número de inversión en inversiones
Si existe número de inversión
obtener monto de interés de la inversión
de otra manera
desplegar "error la inversión no existe"
fin si
fin si
Registrar el importe del ingreso
Si el importe del ingreso menor o igual a cero
desplegar "El campo requiere ser capturado y mayor que cero"
fin si
Llamar proceso afectación contable
Almacenar información en ingresos
FIN INICIO

PROCESO EGRESOS

El siguiente proceso tiene las siguientes entradas:

- El registro de los egresos que pueden ser generados por :
 - Pago de documentos a proveedores.
 - Pago de la nomina del personal de (FNM)
 - Pago de otros servicios
 - etc.

Estos pagos a documentos se registran de la siguiente forma:

Al capturar una alta de egreso se le va asignar automáticamente un número único de egreso. El número de documento se le generara automáticamente, ya que es un campo serial.

Se registra el motivo del pago o del egreso, con una clave que debe existir en catálogo de tipos de egresos.

Se valida el registro de la cuenta concentradora donde se va tomar el importe para pagar el documento.

Se registra el tipo de documento a pagar.

Se registra la cuenta de banco, de donde se tomará para pagar el documento, y esto afectará en la tabla de los saldos de las cuentas bancarias.

Se valida el banco donde debe existir la cuenta bancaria.

Se registra el beneficiario al que se le va a pagar el documento, si es persona física..

Se registra la dependencia del organismo que tuvo la relación cliente-servicio, y que le esta pagando por los servicios prestados a la dependencia de este organismo.

Por lo tanto también se registra el proveedor.

Se registra la fecha programada para el pago al proveedor.

Se registra y se valida la fecha del pago por parte de tesorería.

Se le registra el número contable que va afectar este egreso o pago del documento.

Se registra si se va a pagar con moneda nacional o con moneda extranjera, si es con moneda nacional, se registra el importe con ese tipo de moneda, si no es así se registra el importe con la moneda definida de pago del documento, al registrarse el importe ya sea nacional o extranjero se registra el monto del pago, la forma de pago, el expediente y el trámite.

Estas son las entradas del proceso de Egresos, al terminar de registrarse todas las entradas se llama al proceso y genera las siguientes salidas y afectaciones.

Genera el reporte de los egresos diarios pagados por la Gerencia de Tesorería, afecta a la tabla de saldos y la tabla de afectaciones.

Inicio del proceso de información de Egresos
Asignar número de documento automático
Crea número de registro según le corresponda

- Si el número de registro de egreso tiene asignado un número de (SHIPTAL) entonces
 asignalo al número de egreso.
Fin del sí (crear).
- Se registra número de sucursal bancaria
 Si el número de la sucursal es nulo entonces
 Despliega "se requiere la sucursal"
De otra manera
 Si el número de sucursal no está en el catálogo de bancos entonces
 Despliega "número de sucursal no existe"
 De otra manera
 Despliega "número de sucursal correcta".
 Fin del sí (catálogo)
Fin del sí (nulo)
- Se registra número de banco
 Si el número del banco es nulo entonces
 Despliega "se requiere el banco"
 De otra manera
 Si el número de banco no está en el catálogo de bancos entonces
 Despliega "número de banco no existe"
 Fin del sí (catálogo)
 Fin del sí (nulo)
- Se registra número de compañía proveedora
 Si el número de la compañía proveedora entonces
 Despliega "se requiere la sucursal"
De otra manera
 Si el número de compañía no está en el catálogo de compañía de proveedores entonces
 Despliega "número de compañía no existe"
 Fin del sí (nulo)
- Se registra el RFC de la compañía
 Si el RFC es nulo entonces
 Despliega "se requiere el RFC"
 Fin del sí (nulo)
- Se registra el motivo del egreso
 Si este dato es nulo entonces
 Despliega "se requiere este dato"
De otra manera
 Si el motivo no está en el catálogo de motivos válidos entonces
 Despliega " motivo no válido"
 Fin del sí (catálogo)
 Fin del sí (nulo)
- Se registra la clave concentradora
 Si el número de la clave es nulo entonces
 Despliega "se requiere la clave"
De otra manera
 Si el número de clave no está en el catálogo de concentradoras entonces

- Despliega "número de clave no existe"
- Fin del sí (catálogo)
- Fin del sí (nulo)
- Se registra el tipo de documento a pagar
 - Si el tipo de documento es nulo entonces
 - Despliega "se requiere el dato"
 - De otra manera
 - Si el tipo de documento no está en el catálogo de los tipos de documentos entonces
 - Despliega "tipo de documento no existe"
 - Fin del sí (catálogo)
- Fin del sí (nulo)
- Se registra número de cuenta bancaria
 - Si el número de cuenta es nulo entonces
 - Despliega "se requiere la cuenta"
 - De otra manera
 - Si el número de cuenta no está en el catálogo de cuentas bancarias entonces
 - Despliega "número de cuenta no existe"
 - Fin del sí (catálogo)
- Fin del sí (nulo)
- Se registra número de dependencia de (FNM)
 - Si el número de la dependencia es nula entonces
 - Despliega "se requiere la dependencia"
 - De otra manera
 - Si el número de dependencia no está en el catálogo de dependencias entonces
 - Despliega "número de dependencia no existe"
 - Fin del sí (catálogo)
- Fin del sí (nulo)
- Se registra el proveedor
 - Si el campo del proveedor es nulo entonces
 - Despliega "se requiere el dato"
 - Fin del sí (nulo)
- Se registra la fecha programada para el pago
 - Si la fecha es nula entonces
 - Despliega "se requiere la fecha"
 - Fin del sí (nulo)
- Se registra la fecha programada de tesorería
 - Si la fecha es nula entonces
 - Despliega "se requiere la fecha"
 - Fin del sí (nulo)
- Se registra número contable
 - Si el número contable es nulo entonces
 - Despliega "se requiere el número"
 - Fin del sí (nulo)

- Se registra el tipo de moneda
 - Si el tipo de moneda es nulo entonces
 - Despliega "se requiere el tipo de moneda"
 - De otra manera
 - Si el tipo de moneda no está en el catálogo de tipo de moneda entonces
 - Despliega "tipo de moneda no existe"
 - Fin del sí (catálogo)
 - Fin del sí (nulo)
 - Se registra el importe en moneda nacional
 - Si el importe es nulo entonces
 - Despliega "se requiere este dato"
 - Fin del sí (nulo)
 - Se registra el importe en moneda extranjera
 - Si el importe es nulo entonces
 - Despliega "se requiere este dato"
 - Fin del sí (nulo)
 - Se registra la fecha del pago nacional
 - Si el dato es nulo entonces
 - Despliega "se requiere el dato"
 - Fin del sí (nulo)
 - Se registra la fecha del pago extranjero
 - Si el dato es nulo entonces
 - Despliega "se requiere el dato"
 - Fin del sí (nulo)
 - Se registra el monto del pago
 - Si el dato es nulo entonces
 - Despliega "se requiere el dato"
 - Fin del sí (nulo)
 - Se registra la forma de pago
 - Si el dato es nulo entonces
 - Despliega "se requiere el dato"
 - Fin del sí (nulo)
 - Se registra el Expediente
 - Si el dato es nulo entonces
 - Despliega "Se requiere el dato"
 - Fin del sí
 - Se registra el Trámite
 - Si el dato es nulo entonces
 - Despliega "Se requiere el dato"
 - Fin del sí
 - Llamar proceso afectación contable
 - Almacenar información en egresos
- Fin de Inicio del Proceso de Egresos

PROCESO TRASPASO

Este proceso consiste en que Ferrocarriles Nacionales de México cuenta con diferentes cuentas bancarias, las cuales se clasifican en cuentas captadoras que son donde se realizan ingresos a la institución y las cuentas concentradoras que son donde se concentra todo el dinero para solventar los compromisos de pagos. El mecanismo que normalmente se lleva a cabo es el siguiente:

1. Se consultan todas las cuentas captadoras y se registra el ingreso por cuenta.
2. Se traspasa todo el dinero de estas cuentas captadoras a las cuentas concentradoras de cada banco.
3. En el caso de pagos con depósitos a otras cuentas, se efectúa realizando una transferencia de cuenta a cuenta.
4. Se registran los traspasos.

Definiendo de una manera más técnica este proceso nos ayudaremos de la especificación haciendo uso de un lenguaje estructurado Pseudocódigo.

INICIO

Asignar número de documento según le corresponda

Registrar número de cuenta bancaria fuente

Si no existe cuenta bancaria en catálogo

desplegar "cuenta bancaria errónea"

de otra manera

desplegar descripción de banco

fin si

Registrar número de cuenta bancaria de traspaso

Si no existe cuenta bancaria de traspaso en catálogo

desplegar "cuenta bancaria errónea"

de otra manera

desplegar descripción del banco

fin si

Registrar clave de moneda

Si clave de moneda no existe en catálogo

desplegar "error no existe clave de moneda"

de otra manera

desplegar descripción de moneda

fin si

Registrar importe del traspaso

Si importe es nulo o importe menor igual a cero

desplegar "importe erróneo"

fin si

Registrar de tipo de traspaso

Si tipo de traspaso es diferente de "C" o "B"

desplegar "tipo de traspaso erróneo"

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

Fin si

Si tipo de traspaso es "C"

Asignar status = 1 (Listo para asignarle cheque)

de otra manera

Registrar clave de banca electrónica

fin si

Llamar proceso afectación contable

Almacenar información de traspaso

FIN INICIO

PROCESO INVERSIÓN

Las inversiones se generan a partir del remanente del día esto es, que una vez cubierto los compromisos el día, el dinero que sobra se puede disponer para invertirse en los diferentes instrumentos de inversión y según la cotización del día. Esto implica los siguientes pasos.

1. Verifica cuanto es el remanente disponible en el día.
2. Consulta las cotizaciones del día en los diferentes instrumentos de inversión:
 - a) Cetes
 - b) Tesobonos
 - c) etc.
3. Una vez consultados las cotizaciones del día se elige el instrumento de inversión con un plazo de un día
4. Se obtiene el porcentaje y monto que se obtendrá de la inversión del día
5. Se efectúa la inversión
6. Se registra inversión del día

Definiendo de una manera mas técnica este proceso nos ayudaremos de la especificación haciendo uso de un lenguaje estructurado Pseudocódigo.

INICIO

Asignar número de documento según le corresponda

Registrar clave de grupo financiero

Si clave de grupo financiero no existe en catálogo

desplegar "error no existe grupo financiero

de otra manera

desplegar descripción del grupo financiero

fin si

Registrar clave de cotización

Si clave de cotización no existe en catálogo

desplegar "error no existe cotización del día"

fin si

Registrar número de banco

Si número de banco no existe en catálogo

desplegar "error no existe número de banco
de otra manera
desplegar descripción del banco
fin si
Registrar número de sucursal
Si número de sucursal no existe del banco asignado en catálogo
desplegar "error no existe sucursal del banco asignado"
de otra manera
desplegar descripción de la sucursal
fin si
Registrar número de cuenta bancaria
Si número de cuenta no existe de la sucursal y banco asignado anteriormente
desplegar "error no existe cuenta bancaria para la sucursal y banco"
fin si
Registrar tipo de inversión
Si tipo de inversión es diferente de "C" o "B"
desplegar "tipo de inversión erróneo"
Fin si
Si tipo de inversión es "C"
Asignar status = 1 (Listo para asignarle cheque)
de otra manera
Registrar clave de banca electrónica
fin si Registrar clave de la inversión
Registrar clave de moneda
Si clave de moneda no existe en catálogo
desplegar "error no existe clave de moneda
de otra manera
desplegar descripción de moneda
fin si
Registrar importe de la inversión
Calcular interés generado por el importe de la inversión dependiendo de la cotización
Almacenar información de inversión
Llamar proceso afectación contable

FIN INICIO

PROCESO AFECTACIÓN SALDOS

INICIO

Caso documento = traspaso o inversión

Actualizar saldo de la cuenta fuente de la forma:

cargo por traspaso = cargo por traspaso + importe

saldo final = saldo final - importe

Actualizar saldo de la cuenta destino de la forma:

abono por traspaso = abono por traspaso + importe

saldo final = saldo final + importe
 fin de caso documento = inversión o egreso
Caso documento = ingreso
 Si motivo ingreso = "19", "20", "21", o "23" (devoluciones)
 Actualizar saldo de la cuenta destino de la siguiente manera:
 abono por devolución = abono por devolución + importe
 de otra manera
 Actualizar saldos de la forma:
 abono adicional = abono adicional + importe
 fin si
 Actualizar saldo final = saldo final + importe
 fin caso ingreso
Caso documento = egreso
 Si el documento esta presupuestado
 Actualizar cheque entregado = cheque entregado + importe
 de otra manera
 Actualizar saldos de la forma:
 carga adicional = carga adicional + importe
 fin si
 Actualizar saldo final = saldo final - importe

 fin caso egreso

FIN INICIO

PROCESO INGRESAR AJUSTES

Este es un proceso auxiliar que enmienda los errores del usuario, los cuales implican afectaciones tanto contables como de saldos que no son reales, en caso de un error de esta naturaleza, se pase por alto en la emisión de la carátula y los boletines de la póliza de caja; el error saldrá a la vista con la conciliación bancaria. Tomando en cuenta que los movimientos del día se cierran con la emisión de la póliza de caja, y no se permiten modificaciones a documentos de fechas anteriores, se hace necesario corregir la diferencia de dinero ocasionado por el error del usuario por medio de esta opción.

INICIO

Asignar la fecha del día corriente
 Asignar la bandera de tipo de documento.
 Se registra la clave de región
 Encontrar clave de región en Catálogo de regiones
 Si no se encuentra registro
 desplegar "Clave errónea de región"
 De otra manera
 desplegar la descripción de la región.
 Fin si

Se registra el concepto del documento

- Si el concepto es nulo
desplegar " el concepto no puede ser nulo"
- De otra manera
continua la captura

Fin si

Se registra el importe del documento

- Si el importe es nulo
desplegar " el importe debe ser mayor a cero"
- De otra manera
llamar proceso de afectación contable

Fin si

Si no hay interrupción

- Bloquear la tabla de documento
- Obtener el número serial correspondiente para este registro
- Añadir registro (número, fecha y tipo de documento, así como el usuario que esta haciendo la captura) en tabla documento.
- Desbloquear la tabla de documento.
- Añadir registro (número, concepto e importe del documento) en tabla de ajustes.
- Mientras haya registros de afectación contable
Añadir el registro en la tabla de afectaciones contables con el número de documento obtenido anteriormente.
- fin de mientras

De otra manera

- No procede la captura

Fin si (interrupción)

- Llamar proceso afectación contable ajustes
- Almacenar información en ajustes

FIN INICIO

Termina proceso de ingresar

PROCESO AFECTACIÓN CONTABLE

En el caso de inversiones o traspasos se tiene bien identificado de donde se toma el dinero y a donde se va a depositar, el proceso de afectación contable es automático y sólo se procede a:

- Obtener la cuenta contable asociada al banco donde se toma el dinero y se afecta contablemente como abono.
- Obtener la cuenta contable asociada a la entidad a donde se va a depositar el dinero y se registra como cargo.

En cualquier otro caso, se procede como a continuación se indica:

INICIO

Mientras haya más registros

Se registra la clave contable

Si el grupo cuenta, subcuenta, subsubcuenta, y subsubsubcuenta no están en el catálogo de cuentas

Desplegar "clave inválida "

No procede la captura

De otra manera

continúa la captura

Fin si

Se registra el tipo de movimiento

El tipo de movimiento sólo puede ser cargo o abono

Si el tipo de movimiento es inválido

Desplegar "tipo de movimiento inválido"

De otra manera

Continúa captura

Fin si

Se registra el importe

Si el importe es nulo o cero

Desplegar "el importe tiene que ser mayor a cero "

De otra manera

Guardar los datos en registros temporales

Fin si

Fin de mientras

Si la suma de movimientos capturados cargos - abonos no es igual cero

Desplegar "error en la captura"

No procede la captura

De otra manera

Si la suma de cargos es diferente al importe capturado en el maestro

Desplegar "error en la captura, suma no es igual al importe del ajuste"

No procede la captura

De otra manera

Los importes son correctos

Fin si (importe)

Fin si (movimientos)

FIN INICIO

PROCESO CANCELACIÓN DE CHEQUE DE UN (SIPTAL)

INICIO

Verificar que el cheque no haya sido cancelado previamente

Se registra la fecha de cancelación

Si la fecha es nula

Desplegar "La fecha es inválida"

No procede la captura

De otra manera

Se registra el motivo de cancelación

Si el motivo es nulo

Desplegar "Registre el motivo de cancelación"

La captura no procede

De otra manera

Se registra el responsable

Si el responsable es nulo

Desplegar "Registre el responsable de cancelación"

La captura no procede

De otra manera

Agregar este registro en tabla de cancelaciones

Actualizar el status del cheque en la tabla de cheques

Afectar sumando el importe del cheque al concepto de abono adicional en la tabla de saldos diarios

Mientras haya documentos pagados por este cheque en la tabla de cheque_documento

Actualizar el status="listo para asignar cheque"

a este documento en tabla de egresos

Mientras haya registros de afectación contable en la tabla de afectaciones contables

Agregar este registro pero lo que era abono ahora es cargo y lo que era cargo ahora es abono

Fin de mientras (afectaciones)

Fin de mientras (doctos)

Fin si (responsable)

Fin si (motivo)

Fin si (fecha)

Llamar proceso de baja de flujo de efectivo

FIN INICIO

PROCESO BAJA DE FLUJO DE EFECTIVO

INICIO

En (SIIPTAL)

Mientras haya registro en la tabla pt28flu_efe para este devengado y éste cheque

Buscar el tipo de documento (t16_tip_doc en pt26devengado) de este devengado

Casos

Caso 1 El egreso es viático (con t16_tip_doc = "VT")

Si este devengado tiene relevo (buscar en pt16a_vtcos con

t16_status = "R")

desplegar "Primero dar de baja el relevo"

```
De otra manera
  llamar proceso de borra_flujo
Fin si
Caso 2 El egreso es TG40 (con t16_tip_doc = "TG")
Si este devengado tiene relevo (buscar en ptt24mif con
t24_relevo = "R"
  desplegar "Primero dar de baja el relevo"
De otra manera
  llamar proceso de borra_flujo
Fin si
Caso 3 El egreso es aviso de adeudo (con t16_tip_doc = "AA")
Si este devengado tiene relevo (buscar en ptt21_a_avisos con
t21_a_relevo = "R"
  desplegar "Primero dar de baja el relevo"
De otra manera
  llamar proceso de borra_flujo
Fin si
Fin de casos
Fin de mientras
FIN INICIO
```

PROCESO BORRA FLUJO

INICIO

```
Mientras haya registros con este devengado y con este cheque en la tabla
pt28a_fe_cve
  Actualizar el campo t02_pagado de la tabla ptt02pa de esta clave presupuestal
  restándole al campo t02_pagado el importe del campo t28_importe
Fin de mientras
Actualizar al devengado en ptt16devengado con t16_pagado = t16_pagado +
t28_importe
Borrar todos los registros de la tabla flujo de efectivo de este devengado pagado con
este cheque
Borrar todos los registros de a_fe_cve de este devengado pagado con este cheque
FIN INICIO
```

4.5 Esquema de la base de datos

El esquema de Base de datos se refiere a toda la organización lógica o diagrama lógico de todas las entidades, es decir como el diseñador visualiza a los archivos y sus relaciones, también se le conoce como Diagrama de Estructura de Datos.

Cabe mencionar que para la definición del esquema de la base de datos se someterá al proceso de normalización utilizando un ejemplo de alguna de las tablas del sistema. La normalización es el proceso de transformación de las complejas presentaciones de usuarios y de los almacenamientos de datos en conjuntos estables de estructuras de datos de menor tamaño. Además de ser más sencillas, tales estructuras de datos son más estables. Las estructuras de datos normalizados son más fáciles de mantener.

En el diseño del diccionario de datos el analista debe normalizar las estructuras de datos en tres pasos, cada paso involucra un importante procedimiento de simplificación de la estructura de datos. La relación derivada de la presentación del usuario o del almacenamiento de datos, generalmente se encontrará no normalizada.

El segundo paso asegura que todos los atributos no llave, o sin llave, sean completamente dependientes de la llave del criterio primario. Todas las dependencias normales se eliminan y se colocarán en otra relación.

El tercer paso elimina cualquier dependencia transitoria. Una dependencia transitoria es aquella en la cual sus atributos no llave son dependientes de otros atributos no llave.

La normalización de una relación entre entidades se realiza en tres grandes pasos como se muestra en la figura 4.2

PASOS DE LA NORMALIZACIÓN

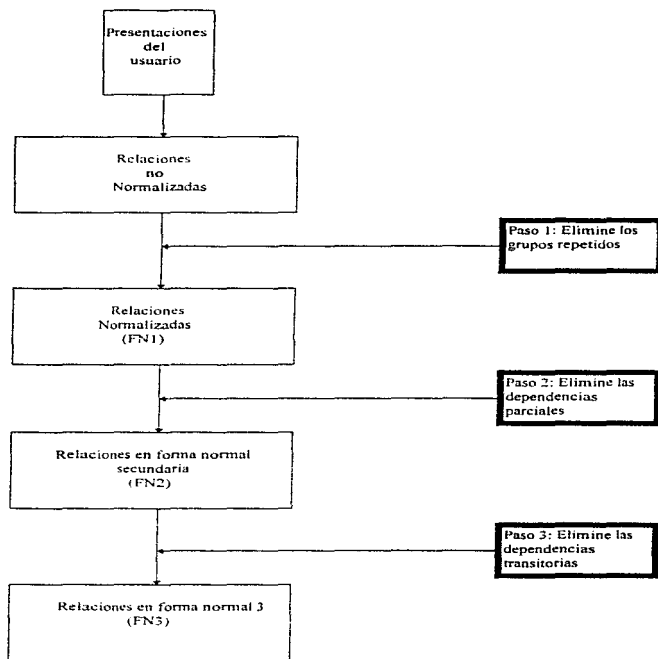


FIGURA 4.2

RELACION DE TABLAS DEL (SIITES)

CATÁLOGOS DEL SISTEMA

TEC01_MONEDA	CATÁLOGO DE TIPOS DE MONEDAS
TEC02_TIPOCAM	CATÁLOGO DE TIPOS DE CAMBIOS
TEC03_CONCEN	CATÁLOGO DE CLAVES DE CUENTAS CONCENTRADORAS
TEC04_CTAB	CATÁLOGO DE CUENTAS BANCARIAS
TEC05_TIPODO	CATÁLOGO DE TIPOS DE DOCUMENTOS
TEC06_TIPOING	CATÁLOGO DE LOS TIPOS DE INGRESOS
TEC08_REGION	CATÁLOGO DE REGIONES
TEC09_BANCO	CATÁLOGO DE BANCOS
TEC10_TIPCANCHE	CATÁLOGO DE LOS TIPOS DE CANCELACION DE CHEQUES
TEC11_SUCUR	CATÁLOGO DE LOS TIPOS DE SUCURSALES BANCARIAS
TEC12_CTASCONT	CATÁLOGO DE CUENTAS CONTABLES
TEC13_INSTRUM	CATÁLOGO DE INSTRUMENTOS DE INVERSION
TEC14_TIPOPO	CATÁLOGO DE TIPOS DE POLIZAS
TEC15_COMPANIAS	CATÁLOGO DL COMPANIAS
TEC16_TIPOTRAS	CATÁLOGO DE TIPOS DE TRASPASOS
TEC20_GPOSFINA	CATÁLOGO DE GRUPOS FINANCIEROS

TABLAS DEL SISTEMA

TET01_INGRESOS	TABLA DE INGRESOS
TET02_TRASPASOS	TABLA DE TRASPASOS
TET03_AJUSTES	TABLA DE AJUSTES
TET04_INVERSIONES	TABLA DE INVERSIONES
TET05_EGRESOS	TABLA DE EGRESOS
TET06_DOCUMENTO	TABLA DE DOCUMENTOS
TET07_AFECTA	TABLA DE AFECTACIONES A CUENTAS
TET08_CHEQUES	TABLA DE CHEQUES
TET09_COTIZACION	TABLA DE COTIZACIONES
TET10_CANCHEQUE	TABLA DE CANCELACIONES DE CHEQUES
TET11_BENEFICIARIO	TABLA DE BENEFICIARIOS O PODER NOTARIAL DE CIAS.
TET12_CANDUCTO	TABLA DE CANCELACION DE DOCUMENTOS
TET13_SALDOS	TABLA DE SALDOS
TET14_CHEDOC	TABLA DE CHEQUES DOCUMENTOS
TET16_AUDITOR	TABLA DE AUDITOR
TET17_COBRADOR	TABLA DE COBRADORES DE CHEQUES
TET19_CTLCHEQ	TABLA DE CONTROL DE LA CHEQUERA
TET21_MOVDOC	TABLA DE MOVIMIENTO DE DOCUMENTOS

CATÁLOGOS DE SEGURIDAD DEL SISTEMA

AUC01USUARIOS	CATÁLOGO DE USUARIOS
AUC02TITULO	CATÁLOGO DE TITULOS
AUC03MENUS	CATÁLOGO DE MENUS
AUC05OPCIONES	CATÁLOGO DE OPCIONES
AUC06INSTRUCC	CATÁLOGO DE INSTRUCCIONES
AUC07FUNCIONES	CATÁLOGO DE FUNCIONES

De acuerdo al diagrama entidad-relación se define el esquema de base de datos del (SIITES) de la siguiente manera.

ESQUEMA DE BASE DE DATOS (SIITES)

```
CREATE TABLE TEC01_MONEDA
(C011MONEDA CHAR(2) NOT NULL,
C01_DESCRIP CHAR(30));

CREATE UNIQUE CLUSTER INDEX XPKTEC05MONEDA ON TEC01_MONEDA (C011MONEDA);
(C011MONEDA) CONSTRAINT U100_1 ;

CREATE TABLE TEC02_TIPOCA
(C02IFECHA DATE NOT NULL,
C021MONEDA CHAR(2) NOT NULL,
C02_TIPOCAM MONEY(9,4));

CREATE UNIQUE CLUSTER INDEX "XPKTET07".SIT04TIPO ON TEC02_TIPOCA
(C02IFECHA,C021MONEDA);

CREATE TABLE TEC03_CONCEN
(C031CONCEN CHAR(2) NOT NULL,
C03_DESCRIP CHAR(20));

CREATE UNIQUE CLUSTER INDEX "XPKTEC06".STC03CONC ON TEC03_CONCEN (C031CONCEN);

CREATE TABLE TEC04_CTAB
(C04IBANCO CHAR(2) NOT NULL,
C041SUCUR SMALLINT NOT NULL,
C041NUMCTA CHAR(18) NOT NULL,
C04ICTACON INTEGER,
C041MONEDA CHAR(2),
C04_DESCRIP CHAR(30),
C04_SALDOMIN DECIMAL(15,2),
C04_TIPO CHAR(1),
C04_TIPOMOV CHAR(1));

CREATE UNIQUE CLUSTER INDEX "XPKTEC06".STC06CTAB ON TEC04_CTAB
(C04IBANCO,C041SUCUR,C041NUMCTA);

CREATE TABLE TEC05_TIPODO
(C051TIPODOC CHAR(2) NOT NULL,
C05_DESCRIP CHAR(20));

CREATE UNIQUE CLUSTER INDEX "XPKTEC05".STC05TIPO ON TEC05_TIPODO (C051TIPODOC);

CREATE TABLE TEC06_TIPOING
(C061MOTIVO SMALLINT NOT NULL,
C06_DESCRIP CHAR(60));

CREATE UNIQUE CLUSTER INDEX XPKTEC06_TIPOING ON TEC06_TIPOING (C061MOTIVO);
```

```
CREATE TABLE TEC08_REGION
(C08IREGION CHAR(1) NOT NULL,
C08_DESCRIP CHAR(20));

CREATE UNIQUE CLUSTER INDEX XPKTEC08_REGION ON TEC08_REGION (C08IREGION);

CREATE TABLE TEC09_BANCO
(C09IBANCO CHAR(2) NOT NULL,
C09_DESCRIP CHAR(30));

CREATE UNIQUE CLUSTER INDEX XPKTEC09_BANCO ON TEC09_BANCO (C09IBANCO);

CREATE TABLE TEC10_TIPCANCHE
(C10IMOTIVO CHAR(1) NOT NULL,
C10_DESCRIP CHAR(30));

CREATE UNIQUE CLUSTER INDEX XPKTEC10_TIPCAN ON TEC10_TIPCANCHE (C10IMOTIVO);

CREATE TABLE TEC11_SUCUR
(C11IBANCO CHAR(2) NOT NULL,
C11ISUCUR SMALLINT NOT NULL,
C11_DESCRIP CHAR(60),
C11_DOMICILIO CHAR(50),
C11_DELEGMUN CHAR(20),
C11_EDOCIUD CHAR(30),
C11_PAIS CHAR(15),
C11_TEL CHAR(10),
C11_ATENDIO CHAR(30),
C11_REGION CHAR(1));

CREATE UNIQUE CLUSTER INDEX XPKTEC11_SUCUR ON TEC11_SUCUR
(C11IBANCO,C11ISUCUR);

CREATE TABLE TEC12_CTASCONT
(C12ICTACON SERIAL NOT NULL,
C12ICTA SMALLINT NOT NULL,
C12ISCTA SMALLINT NOT NULL,
C12ISSCTA SMALLINT NOT NULL,
C12ISSSCTA SMALLINT,
C12_DESCRIP CHAR(50));

CREATE UNIQUE CLUSTER INDEX XPKTEC12_CTASCONT ON TEC12_CTASCONT (C12ICTACON);

CREATE TABLE TEC13_INSTRUM
(C13INSTRU CHAR(1) NOT NULL,
C13_DESCRIP CHAR(60));

CREATE UNIQUE CLUSTER INDEX XPKTEC13_INSTRUM ON TEC13_INSTRUM (C13INSTRU);

CREATE TABLE TEC14_TIPOPO
(C14ITIPOL SMALLINT NOT NULL,
C14_DESCRIP CHAR(10));

CREATE UNIQUE CLUSTER INDEX XPKTEC14_TIPOPO ON TEC14_TIPOPO (C14ITIPOL);
```

```
CREATE TABLE TEC15_COMPANIAS
(C15ICIA CHAR(5) NOT NULL,
 C15_DESCRIP CHAR(140));
```

```
CREATE UNIQUE CLUSTER INDEX XPKTEC15COMPANIAS ON TEC15_COMPANIAS (C15ICIA);
```

```
CREATE TABLE TEC16_TIPOTRAS
(C16TIPOTRAS CHAR(1) NOT NULL,
 C16_DESCRIP CHAR(40));
```

```
CREATE UNIQUE CLUSTER INDEX XPKTEC16_TIPOTRAS ON TEC16_TIPOTRAS (C16TIPOTRAS);
```

```
CREATE TABLE TEC20_GPOSFINA
(C20ICVEGPO CHAR(2) NOT NULL,
 C20_DESCRIP CHAR(30) NOT NULL,
 C20_CONTRATO CHAR(30) NOT NULL,
 C20ICVECON INTEGER NOT NULL);
```

```
CREATE UNIQUE INDEX IX149_1 ON TEC20_GPOSFINA (C20ICVEGPO);
```

```
CREATE TABLE TET01_INGRESOS
(T01INUM_DOCTO INTEGER NOT NULL,
 T01IMOV SMALLINT NOT NULL,
 T01IBANCO CHAR(2),
 T01ISUCUR SMALLINT,
 T01INUMCTA CHAR(18),
 T01IMOTIVO SMALLINT,
 T01_NUMINV INTEGER,
 T01_IMPORTE MONEY(20,4));
```

```
CREATE UNIQUE CLUSTER INDEX XPKTET01_INGRESOS ON TET01_INGRESOS
(T01INUM_DOCTO, T01IMOV);
```

```
CREATE TABLE TET02_TRASPASOS
(T02INUM_DOCTO INTEGER NOT NULL,
 T02IMOV SMALLINT,
 T02IBANCO CHAR(2),
 T02ISUCUR SMALLINT,
 T02INUMCTA CHAR(18),
 T02IBANCOTRAS CHAR(2),
 T02ISUCURTRAS SMALLINT,
 T02INUMCTATRAS CHAR(18),
 T02_IMPTRASPASO MONEY(20,4),
 T02ITIPOTRAS CHAR(1),
 T02_BAN_CHE CHAR(18));
```

```
CREATE UNIQUE CLUSTER INDEX XPKTET02_TRASPASOS ON TET02_TRASPASOS
(T02INUM_DOCTO, T02IMOV);
```

```
CREATE TABLE TET03_AJUSTES
(T03INUM_DOCT SERIAL NOT NULL,
 T03IMOV SMALLINT,
 T03_IMPORTE MONEY(20,4));
```

CREATE UNIQUE CLUSTER INDEX XPKTET03_AJUSTES ON TET03_AJUSTES
(T03INUM_DOCTO,T03IMOV);

CREATE TABLE TET04_INVERSIONES
(T04INUM_DOCTO INTEGER NOT NULL,
T04IMOV SMALLINT NOT NULL,
T04IGPOFIN CHAR(2),
T04IFECHACOT DATE,
T04_COTIZACION SMALLINT,
T04IBANCO CHAR(2),
T04ISUCUR SMALLINT,
T04INUMCTA CHAR(18),
T04_PLAZO SMALLINT,
T04_FORMAINV CHAR(1),
T04_BAN_CHE CHAR(18),
T04_IMPINVERSION MONEY(20,4),
T04_IMPINTERES MONEY(20,4));

CREATE UNIQUE CLUSTER INDEX XPKTET04_INVERSION ON TET04_INVERSIONES
(T04INUM_DOCTO,T04IMOV);

CREATE TABLE TET05_EGRESOS
(T05INUM_DOCTO INTEGER NOT NULL,
T05IMOV SMALLINT NOT NULL,
T05IDEVENGADO SMALLINT,
T05IBANCO CHAR(2),
T05ISUCUR SMALLINT,
T05INUMCTA CHAR(18),
T05_NOMBEN CHAR(80),
T05_EXPEDIENTE CHAR(20),
T05ICONCEN SMALLINT,
T05ITIPODOC SMALLINT,
T05_DEPENDENCIA CHAR(11),
T05_FECPROGPAGO DATE,
T05_URCA CHAR(20),
T05_IMPACIONAL MONEY(20,4),
T05_IMPEXTRANJERA MONEY(20,4),
T05_PROPAGMN MONEY(20,4),
T05_PROPAGME MONEY(20,4),
T05_MONTOPAGO MONEY(20,4),
T05_FORMAPAGO CHAR(2),
T05_BENEFIC CHAR(1));

PRIMARY KEY (T05INUM_DOCTO,T05IMOV) CONSTRAINT U127_27);

CREATE UNIQUE CLUSTER INDEX XPKTET05_EGRESOS ON TET05_EGRESOS
(T05INUM_DOCTO,T05IMOV);

CREATE TABLE TET06_DOCUMENTO
(T06INUM_DOCTO SERIAL NOT NULL,
T06ICVEUJUARIO CHAR(18),
T06IREGION CHAR(1),
T06_FECDOCTO DATE,
T06_STATUS CHAR(1));

CREATE UNIQUE CLUSTER INDEX XPKTET06_DOCUMENTO ON TET06_DOCUMENTO
(T06INUM_DOCTO);

CREATE TABLE TET07_AFECTA
(T07INUM_DOCTO INTEGER NOT NULL,
T07IMOV SMALLINT,
T07ICTACON INTEGER,
T07_CAR_ABO CHAR(1),
T07_IMPORTE MONEY(20,4));
CREATE UNIQUE INDEX XPKTET07_AFECTA ON
TET07_AFECTA(T07INUM_DOCTO,T07IMOV,T07ICTACON)

CREATE TABLE TET08_CHEQUES
(T08IBANCO CHAR(2) NOT NULL,
T08ISUCUR SMALLINT NOT NULL,
T08INUMCTA CHAR(18) NOT NULL,
T08NUMCHEQUE INTEGER NOT NULL,
T08IMONEDA CHAR(2),
T08_FECHAIMP DATE,
T08_BENEFICIARIO CHAR(80),
T08_FECENT DATE,
T08_STATUS CHAR(1),
T08_IMPORTE MONEY(20,4));

CREATE UNIQUE CLUSTER INDEX XPKTET08_CHEQUES ON TET08_CHEQUES
(T08IBANCO,T08ISUCUR,T08INUMCTA,T08NUMCHEQUE);

CREATE TABLE TET09_COTIZACION
(T09IGPOFIN CHAR(2) NOT NULL,
T09IFECHA DATE NOT NULL,
T09INUM_COT SMALLINT NOT NULL,
T09INSTRU CHAR(1),
T09_HORA DATETIME HOUR TO MINUTE,
T09_PORCENTAJE SMALLFLOAT);

CREATE UNIQUE CLUSTER INDEX XPKTET09_COTIZACION ON TET09_COTIZACION
(T09IGPOFIN,T09IFECHA,T09INUM_COT);

CREATE TABLE TET10_CANCHEQUE
(T10IBANCO CHAR(2) NOT NULL,
T10ISUCUR SMALLINT NOT NULL,
T10INUMCTA CHAR(18) NOT NULL,
T10NUMCHEQUE INTEGER NOT NULL,
T10MOTIVO CHAR(1),
T10_FECHA DATE,
T10_RESPONSABLE CHAR(60));

CREATE UNIQUE CLUSTER INDEX XPKTET10_CANCHEQUE ON TET10_CANCHEQUE
(T10IBANCO,T10ISUCUR,T10INUMCTA,T10NUMCHEQUE);

CREATE TABLE TET11_BENEFICIARIO
(T11ICIA CHAR(5) NOT NULL,
T11IRFC CHAR(13) NOT NULL,
T11_NOMBRE CHAR(40),

```
T11_TELEFONO CHAR(18),
T11_PUESTO CHAR(40),
T11_STATUS CHAR(1);
```

```
CREATE UNIQUE CLUSTER INDEX XPKTET11_BENEFICIA ON TET11_BENEFICIARIO
(T11ICIA,T11IRFC);
```

```
CREATE TABLE TET12_CANDOCTO
(T12INUM_DOCTO INTEGER NOT NULL,
T12IMOV SMALLINT NOT NULL,
T12_FECHA DATE,
T12_MOTIVO TEXT,
T12_RESPONSABLE CHAR(30));
```

```
CREATE UNIQUE CLUSTER INDEX XPKTET12_CANDOCTO ON TET12_CANDOCTO
(T12INUM_DOCTO);
```

```
CREATE TABLE TET13_SALDOS
(T13IBANCO CHAR(2) NOT NULL,
T13ISUCUR SMALLINT NOT NULL,
T13INUMCTA CHAR(18) NOT NULL,
T13IFECHA DATE NOT NULL,
T13_SALDOINI MONEY(20,4),
T13_SALDOFIN MONEY(20,4),
T13_ABO_ADI MONEY(20,4),
T13_ABO_TRA MONEY(20,4),
T13_ABO_DEV MONEY(20,4),
T13_CAR_ADI MONEY(20,4),
T13_CAR_TRA MONEY(20,4),
T13_CHE_ENT MONEY(20,4),
T13_COMPDIA MONEY(20,4),
T13_INVERSION MONEY(20,4));
```

```
CREATE UNIQUE CLUSTER INDEX XPKTET13_SALDOS ON TET13_SALDOS
(T13IBANCO,T13ISUCUR,T13INUMCTA,T13IFECHA);
```

```
CREATE TABLE TET14_CHEDOC
(T14IBANCO CHAR(2) NOT NULL,
T14ISUCUR SMALLINT NOT NULL,
T14INUMCTA CHAR(18) NOT NULL,
T14INUMCHQUE INTEGER NOT NULL,
T14INUM_DOCTO INTEGER NOT NULL,
T14IMOV SMALLINT NOT NULL);
```

```
CREATE UNIQUE CLUSTER INDEX XPKTET14_CHEDOC ON TET14_CHEDOC
(T14INUM_DOCTO,T14ISUCUR,T14INUMCTA,T14INUMCHQUE,T14INUM_DOCTO);
```

```
CREATE TABLE TET16_AUDITOR
(T16ICVTRANSA CHAR(1) NOT NULL,
T16ITABLA CHAR(18) NOT NULL,
T16IFECHA DATE NOT NULL,
T16IHORA DATE TIME HOUR TO MINUTE NOT NULL,
T16ICVEUSUARIO CHAR(18) NOT NULL,
T16_TERMINAL CHAR(6));
```

```

T16_DATO          CHAR(1024));

CREATE UNIQUE CLUSTER INDEX XPKTET16_AUDITOR ON TET16_AUDITOR
(T16ICVETRAMSA,T16ITABLA,T16IFECHA,T16IHORA,T16ICVEUSUARIO);

CREATE TABLE TET17_COBRADOR
(T17INUM_DOCTO  INTEGER      NOT NULL,
 T17IMOV       SMALLINT    NOT NULL,
 T17ICIA       CHAR(5)      NOT NULL,
 T17IRFC       CHAR(13)     NOT NULL);

CREATE UNIQUE INDEX XPKTET17_COBRADOR ON TET17_COBRADOR
(T17INUM_DOCTO,T17IMOV);

CREATE TABLE TET19_CTLCHEQ
(T19IBANCO     CHAR(2)      NOT NULL,
 T19ISUCUR     SMALLINT    NOT NULL,
 T19INUMCTA    CHAR(4)      NOT NULL,
 T19ICHEQUERA  INTEGER     NOT NULL,
 T19_FECHA     DATE,
 T19_INICIAL   INTEGER,
 T19_FINAL     INTEGER,
 T19_INVENTARIO INTEGER,
 T19_STATUS    CHAR(1));

CREATE UNIQUE CLUSTER INDEX XPKTET19_CTLCHEQ ON TET19_CTLCHEQ
(T19IBANCO,T19ISUCUR,T19INUMCTA,T19ICHEQUERA);

CREATE TABLE TET21_MOVDOC
(T21INUM_DOCTO INTEGER NOT NULL,
 T21IMOV       SMALLINT,
 T21ITIPOL     SMALLINT,
 T21_CONCEPTO TEXT IN FNMP3,
 T21MONEDA    CHAR(2),
 T21_FECPOLIZA DATE,
 T21_CLASEDOC CHAR(1),
 T21_STATUS    SMALLINT);

CREATE UNIQUE CLUSTER INDEX IX_T21_MOVDOC ON TET21_MOVDOC
(T21INUM_DOCTO,T21IMOV);

CREATE TABLE AUC01USUARIOS
(C01USUARIO    CHAR(18)    NOT NULL,
 C01_NOMBRE_US CHAR(35),
 C01_FEC_ALTA  DATE,
 C01_AUTORIZA CHAR(18),
 C01_NODO      CHAR(18),
 C01_SISTEMA   CHAR(70));

CREATE UNIQUE INDEX IDX_USU1 ON AUC01USUARIOS (C01USUARIO);

CREATE TABLE AUC02TITULO
(C02_TITULO1   CHAR(60)    NOT NULL,
 C02_TITULO2   CHAR(76));

```

```
C02_TITULO3 CHAR(76));

CREATE TABLE AUC03MENUS
(C03ICLAVE INTEGER NOT NULL,
C03_NOMBRE CHAR(78) NOT NULL);

CREATE UNIQUE INDEX IDX_MEN1 ON AUC03MENUS (C03ICLAVE);

CREATE TABLE AUC05OPCIONES
(C05ICLMENU INTEGER NOT NULL,
C05IOPCION INTEGER NOT NULL,
C05_NOMBRE CHAR(70) NOT NULL,
C05_TIPO CHAR(4) NOT NULL,
C05_INSTRUCCION CHAR(10) NOT NULL,
C05_LETRA CHAR(1),
C05_MENSAJE CHAR(74));

CREATE INDEX IDX_OPC1 ON AUC05OPCIONES (C05ICLMENU);
CREATE INDEX IDX_OPC2 ON AUC05OPCIONES (C05IOPCION);
CREATE UNIQUE INDEX IDX_OPC3 ON AUC05OPCIONES (C05ICLMENU,C05IOPCION);

CREATE TABLE AUC06INSTRUCC
(C06INS_NUM CHAR(10) NOT NULL,
C06_RUTA CHAR(70) NOT NULL,
C06_PROG4GE CHAR(9),
C06_PARAMETROS CHAR(80));

CREATE UNIQUE INDEX IDX_INSI ON AUC06INSTRUCC (C06INS_NUM);

CREATE TABLE AUC07FUNCIONES
(C07INOMBRE CHAR(18) NOT NULL,
C07ICLMENU INTEGER NOT NULL,
C07IOPCION INTEGER NOT NULL,
C07_FUNCIO CHAR(1) NOT NULL);

CREATE UNIQUE INDEX IDX_FUN1 ON AUC07FUNCIONES
(C07INOMBRE,C07ICLMENU,C07IOPCION);

ALTER TABLE TEC01_MONEDA
ADD CONSTRAINT PRIMARY KEY(C011MONEDA);

ALTER TABLE TEC02_TIPOCA
ADD CONSTRAINT PRIMARY KEY (C02IFECHA,C021MONEDA) CONSTRAINT U101_2 ;

ALTER TABLE TEC03_CONCEN
ADD CONSTRAINT PRIMARY KEY (C031CONCEN) CONSTRAINT U102_3 ;

ALTER TABLE TEC04_CTAB
ADD CONSTRAINT PRIMARY KEY (C041BANCO,C041SUCUR,C041NUMCTA) CONSTRAINT U103_4 ;

ALTER TABLE TEC05_TIPODO
ADD CONSTRAINT PRIMARY KEY (C051TIPODOC) CONSTRAINT U104_5 ;

ALTER TABLE TEC06_TIPOING
```



```
ADD CONSTRAINT PRIMARY KEY (C06IMOTIVO) CONSTRAINT U105_6 ;
ALTER TABLE TEC08_REGION
ADD CONSTRAINT PRIMARY KEY (C08IREGION) CONSTRAINT U107_8 ;
ALTER TABLE TEC09_BANCO
ADD CONSTRAINT PRIMARY KEY (C09IBANCO) CONSTRAINT U108_9 ;
ALTER TABLE TEC10_TIPCANCHE
ADD CONSTRAINT PRIMARY KEY (C10IMOTIVO) CONSTRAINT U109_10 ;
ALTER TABLE TEC11_SUCUR
ADD CONSTRAINT PRIMARY KEY (C11IBANCO,C11ISUCUR) CONSTRAINT U110_11 ;
ALTER TABLE TEC12_CTASCONT
ADD CONSTRAINT PRIMARY KEY (C12ICTACON) CONSTRAINT U111_12 ;
ALTER TABLE TEC13_INSTRUM
ADD CONSTRAINT PRIMARY KEY (C13IINSTRU) CONSTRAINT U112_13 ;
ALTER TABLE TEC14_TIPOPO
ADD CONSTRAINT PRIMARY KEY (C14ITIPOL) CONSTRAINT UTEC14 ;
ALTER TABLE TEC15_COMPANIAS
ADD CONSTRAINT PRIMARY KEY (C15ICIA) CONSTRAINT U114_15 ;
ALTER TABLE TEC16_TIPOTRAS
ADD CONSTRAINT PRIMARY KEY (C16ITIPOTRAS) CONSTRAINT U116_17 ;
ALTER TABLE TEC20_GPOSFINA
ADD CONSTRAINT PRIMARY KEY (C20ICVEGPO) CONSTRAINT U149_49 ;
ALTER TABLE TET01_INGRESOS
ADD CONSTRAINT PRIMARY KEY (T01INUM_DOCTO,T01IMOV) CONSTRAINT U123_23 ;
ALTER TABLE TET02_TRASPASOS
ADD CONSTRAINT PRIMARY KEY (T02INUM_DOCTO,T02IMOV) CONSTRAINT U124_24 ;
ALTER TABLE TET03_AJUSTES
ADD CONSTRAINT PRIMARY KEY (T03INUM_DOCTO,T03IMOV) CONSTRAINT U125_25 ;
ALTER TABLE TET04_INVERSIONES
ADD CONSTRAINT PRIMARY KEY (T04INUM_DOCTO,T04IMOV) CONSTRAINT U126_26 ;
ALTER TABLE TET06_DOCUMENTO
ADD CONSTRAINT PRIMARY KEY (T06INUM_DOCTO) CONSTRAINT U128_28 ;
ALTER TABLE TET08_CHEQUES
ADD CONSTRAINT PRIMARY KEY (T08IBANCO,T08ISUCUR,T08INUMCTA,T08INUMCHEQUE)
CONSTRAINT U130_30 ;
ALTER TABLE TET09_COTIZACION
ADD CONSTRAINT PRIMARY KEY (T09IGPOFIN,T09IFECHA,T09INUM_COT) CONSTRAINT
U131_31 ;
```

ALTER TABLE TET10_CANCHEQUE
ADD CONSTRAINT PRIMARY KEY (T10IBANCO,T10ISUCUR,T10INUMCTA,T10INUMCHEQUE)
CONSTRAINT U132_32 ;

ALTER TABLE TET11_BENEFICIARIO
ADD CONSTRAINT PRIMARY KEY (T11ICIA,T11IRFC) CONSTRAINT U133_33 ;

ALTER TABLE TET12_CANDOCTO
ADD CONSTRAINT PRIMARY KEY (T12INUM_DOCTO,T12IMOV) CONSTRAINT U134_34 ;

ALTER TABLE TET13_SALDOS
ADD CONSTRAINT PRIMARY KEY (T13IBANCO,T13ISUCUR,T13INUMCTA,T13IFECHA)
CONSTRAINT U135_35 ;

ALTER TABLE TET15_ACUMULADO
ADD CONSTRAINT PRIMARY KEY (T15IANIO,T15IMES,T15ITIPO,T15IMOTIVO,T15ITPOMOV)
CONSTRAINT U137_37 ;

ALTER TABLE TET19_CTLCHEQ
ADD CONSTRAINT PRIMARY KEY (T19IBANCO,T19ISUCUR,T19INUMCTA,T19ICHEQUERA)
CONSTRAINT U139_39 ;

ALTER TABLE TET21_MOVDOC
ADD CONSTRAINT PRIMARY KEY (T21INUM_DOCTO,T21IMOV) CONSTRAINT U124_21 ;

ALTER TABLE TEC02_TIPOCA
ADD CONSTRAINT (FOREIGN KEY (C02IMONEDA) REFERENCES TEC01_MONEDA CONSTRAINT
R_26);

ALTER TABLE TEC04_CTAB
ADD CONSTRAINT (FOREIGN KEY (C04ICTACON) REFERENCES TEC12_CTASCONT CONSTRAINT
R_74);

ALTER TABLE TEC04_CTAB
ADD CONSTRAINT (FOREIGN KEY (C04IBANCO,C04ISUCUR) REFERENCES TEC11_SUCUR
CONSTRAINT R_47);

ALTER TABLE TEC04_CTAB
ADD CONSTRAINT (FOREIGN KEY (C04IMONEDA) REFERENCES TEC01_MONEDA CONSTRAINT
R_28);

ALTER TABLE TET02_TRASPASOS
ADD CONSTRAINT (FOREIGN KEY (T02IBANCO,T02ISUCUR,T02INUMCTA) REFERENCES
TEC04_CTAB CONSTRAINT R_75);

ALTER TABLE TEC11_SUCUR
ADD CONSTRAINT (FOREIGN KEY (C11IBANCO) REFERENCES TEC09_BANCO CONSTRAINT
R_46);

ALTER TABLE TEC20_GPOSFINA
ADD CONSTRAINT (FOREIGN KEY (C20ICVECON) REFERENCES TEC12_CTASCONT CONSTRAINT
U149_50);

ALTER TABLE TET01_INGRESOS
ADD CONSTRAINT (FOREIGN KEY (T01IBANCO,T01ISUCUR,T01INUMCTA) REFERENCES
TEC04_CTAB CONSTRAINT R_21);

ALTER TABLE TET01_INGRESOS
ADD CONSTRAINT (FOREIGN KEY (T01IMOTIVO) REFERENCES TEC06_TIPOING CONSTRAINT
R_5);

ALTER TABLE TET02_TRASPASOS
ADD CONSTRAINT (FOREIGN KEY (T02ITIPOTRAS) REFERENCES TEC16_TIPOTRAS CONSTRAINT
R_66);

ALTER TABLE TET03_AJUSTES
ADD CONSTRAINT (FOREIGN KEY (T03INUM_DOCTO) REFERENCES TET06_DOCUMENTO
CONSTRAINT R_16);

ALTER TABLE TET04_INVERSIONES
ADD CONSTRAINT (FOREIGN KEY (T04IBANCO,T04ISUCUR,T04INUMCTA) REFERENCES
TEC04_CTAB CONSTRAINT R_22);

ALTER TABLE TET05_EGRESOS
ADD CONSTRAINT (FOREIGN KEY (T05IMONEDA) REFERENCES TEC01_MONEDA CONSTRAINT
R_78);

ALTER TABLE TET05_EGRESOS
ADD CONSTRAINT (FOREIGN KEY (T05IBANCO,T05ISUCUR,T05INUMCTA) REFERENCES
TEC04_CTAB CONSTRAINT R_49);

ALTER TABLE TET05_EGRESOS
ADD CONSTRAINT (FOREIGN KEY (T05ICIA,T05IRFC) REFERENCES TET11_BENEFICIARIO
CONSTRAINT R_44);

ALTER TABLE TET05_EGRESOS
ADD CONSTRAINT (FOREIGN KEY (T05IMOTIVO) REFERENCES TEC07_TIPOEGR CONSTRAINT
R_38);

ALTER TABLE TET05_EGRESOS
ADD CONSTRAINT (FOREIGN KEY (T05ICONCEN) REFERENCES TEC03_CONCEN CONSTRAINT
R_33);

ALTER TABLE TET05_EGRESOS
ADD CONSTRAINT (FOREIGN KEY (T05ITIPODOC) REFERENCES TEC05_TIPODO CONSTRAINT
R_30);

ALTER TABLE TET04_INVERSIONES
ADD CONSTRAINT (FOREIGN KEY (T04INUM_DOCTO) REFERENCES TET06_DOCUMENTO
CONSTRAINT R_17);

ALTER TABLE TET05_EGRESOS
ADD CONSTRAINT (FOREIGN KEY (T05INUM_DOCTO) REFERENCES TET06_DOCUMENTO
CONSTRAINT R_18);

ALTER TABLE TET06_DOCUMENTO
ADD CONSTRAINT (FOREIGN KEY (T06ICVEUSUARIO) REFERENCES TEC17_USUARIOS
CONSTRAINT R_61);

ALTER TABLE TET06_DOCUMENTO
ADD CONSTRAINT (FOREIGN KEY (T06IREGION) REFERENCES TEC08_REGION CONSTRAINT
R_19);

ALTER TABLE TET107_AFECTA
ADD CONSTRAINT (FOREIGN KEY T07ICTACON) REFERENCES TEC12_CTASCONT CONSTRAINT
R_73);

ALTER TABLE TET08_CHEQUES
ADD CONSTRAINT (FOREIGN KEY (T08IMONEDA) REFERENCES TEC01_MONEDA CONSTRAINT
R_35);

ALTER TABLE TET09_COTIZACION
ADD CONSTRAINT (FOREIGN KEY (T09IINSTRU) REFERENCES TEC13_INSTRUM CONSTRAINT
R_42);

ALTER TABLE TET09_COTIZACION
ADD CONSTRAINT (FOREIGN KEY (T09IGPOFIN) REFERENCES TEC20_GPOSFINA CONSTRAINT
R_39);

ALTER TABLE TET10_CANCHEQUE
ADD CONSTRAINT (FOREIGN KEY (T10IMOTIVO) REFERENCES TEC10_TIPCANCHE CONSTRAINT
R_40);

ALTER TABLE TET11_BENEFICIARIO
ADD CONSTRAINT (FOREIGN KEY (T11ICIA) REFERENCES TEC15_COMPANIAS CONSTRAINT
R_43);

ALTER TABLE TET12_CANDOCTO
ADD CONSTRAINT (FOREIGN KEY (T12INUM_DOCTO) REFERENCES TET06_DOCUMENTO
CONSTRAINT R_45);

ALTER TABLE TET13_SALDOS
ADD CONSTRAINT (FOREIGN KEY (T13IBANCO,T13ISUCUR,T13INUMCTA) REFERENCES
TEC04_CTAB CONSTRAINT R_52);

ALTER TABLE TET14_CHEDOC
ADD CONSTRAINT (FOREIGN KEY (T14INUM_DOCTO) REFERENCES TET06_DOCUMENTO
CONSTRAINT R_54);

ALTER TABLE TET21_MOVDOC
ADD CONSTRAINT (FOREIGN KEY (T21IMONEDA) REFERENCES TEC01_MONEDA CONSTRAINT
R__21);

ALTER TABLE TET21_MOVDOC
ADD CONSTRAINT (FOREIGN KEY (T21INUM_DOCTO) REFERENCES TET06_DOCUMENTO
CONSTRAINT R__22);

```
ALTER TABLE TET21_MOVDOC  
ADD CONSTRAINT (FOREIGN KEY (T21ITPOL) REFERENCES TEC14_TIPOPO CONSTRAINT R_23);
```



CAPÍTULO V

CAPÍTULO V DESARROLLO E IMPLANTACIÓN

La programación y pruebas normalmente comienzan, como pudiera esperarse cuando termina la actividad de diseño. La fase de programación e implantación de un proyecto típico involucra la escritura de instrucciones en el lenguaje de programación a desarrollar para implantar lo que el analista ha especificado y el diseñador ha organizado en módulos. La prueba, como el nombre implica, involucra ejecutar el sistema para asegurar que produzca las salidas apropiadas y exhiba el comportamiento adecuado para una gama amplia de entradas.

Sin tomar en cuenta el lenguaje de programación que se use, hay cuestiones comunes que todos los programadores enfrentan. Como analista, debe estar familiarizado con ellas y las más comunes se mencionan a continuación:

- **Productividad:** probablemente, la cuestión más importante de la programación actual sea la productividad, escribir más software, más rápidamente. La principal razón de esto es la enorme cantidad de sistemas y aplicaciones que siguen en espera en las grandes organizaciones. Por ello se deben alentar los lenguajes y técnicas de programación que promueven la productividad; exceptuando en algunos casos, la productividad se considera más importante que la eficiencia.
- **Eficiencia:** en algunas aplicaciones, la eficiencia sigue siendo de importancia. Esto sucede en muchos sistemas de tiempo real, y puede darse en otros tipos de sistemas que procesan grandes volúmenes de datos. Para estas aplicaciones, usualmente resulta importante minimizar la cantidad de tiempo de CPU requerido por el programa; también puede ser importante minimizar la utilización de memoria, al igual que la de otros recursos como el disco.
- **Corrección:** en el caso que alguna aplicación no satisfaga las necesidades planteadas o no funcione correctamente, es válido e imprescindible realizar correcciones sobre el mismo. Después de todo, si el programa no funciona correctamente, no importa qué tan eficiente sea.
- **Portabilidad:** en algunos ambientes esto es importante; el usuario puede desear ejecutar el mismo sistema en varios tipos de computadoras. Algunos lenguajes de programación son más portátiles que otros; siempre hay forma de que el programador aproveche las características especiales de una computadora o un sistema operativo específico. Por ello, además del lenguaje de programación debemos preocuparnos por el estilo de programación, pues la portabilidad es un factor importante.
- **Mantenibilidad:** finalmente, debemos recordar que gracias al mantenimiento del sistema, se prolonga la vida del mismo durante mucho tiempo.

5.1. Calendario de Actividades

En él se especifican las actividades a realizar para el desarrollo del sistema, como se muestra en la figura 5.1

5.2 Identificación de Programas

La relación de programas que han sido definidos para la elaboración de todo el sistema se presentan a continuación, considerando que existe la posibilidad de ampliar la lista de programas.

Es importante mencionar que para la definición de los nombres o identificadores de los programas se hizo de la siguiente manera:

Los 2 primeros caracteres identifican al sistema, la tercera letra identifica si es una función o forma y los siguientes caracteres identifican la operación del programa, ejemplo:

```

T E F B A N C O
| | | |
| | | | ----- Tabla de bancos(hasta 9 caracteres)
| | | | ----- Función de Mantenimiento a la tabla.
| | | |
| | | |
|----- Tesoreria

T E P B A N C O
| | | |
| | | | ----- Tabla de bancos (hasta 9 caracteres)
| | | | ----- Pantalla de captura.
| | | |
| | | |
|----- Tesoreria
  
```

Bajo esta definición de nomenclatura se listan los programas:

Programa	Descripción
tefbanco.4gl	Registra información del catálogo de bancos
tefcompan.4gl	Registra información del catálogo de compañías
tefconcn.4gl	Registra información del catálogo de cuentas concentradoras
tefcotiza.4gl	Registra información de cotizaciones del día
tefctrche.4gl	Registra información del control de la chequera
tefcheq.4gl	Registra información de la generación e impresión de cheques

tefexi.4gl	Programa de librería para la validación de llaves primarias
tefdescr.4gl	Programa de librería para las descripciones de llaves primarias
teflib.4gl	Programa de librería para ayuda de catálogos
tefmoneda.4gl	Registra información del catálogo de monedas
teftipdoc.4gl	Registra información del catálogo de tipos de documentos
tefregión.4gl	Registra información del catálogo de regiones
tefsucur.4gl	Registra información del catálogo de sucursales
teftiping.4gl	Registra información del catálogo de tipos de ingresos
teftipoca.4gl	Registra información del catálogo de tipos de cambio
teftipegr.4gl	Registra información del catálogo de tipos de egresos
teftpoliz.4gl	Registra información del catálogo de tipos de pólizas
teftiptra.4gl	Registra información del catálogo de tipos de traspasos
tefsuari.4gl	Registra información del catálogo de usuarios
teftacon.4gl	Registra información del catálogo de cuentas contables
teftctab.4gl	Registra información del catálogo de cuentas bancarias
teftinstru.4gl	Registra información del catálogo de instrumentos de inversión
teftgposf.4gl	Registra información del catálogo de grupos financieros
teftcanche.4gl	Registra información de cancelación de cheques
teftingreso.4gl	Registra información de ingresos
teftegreso.4gl	Registra información de egresos
teftinvers.4gl	Registra información de inversiones
tefttraspa.4gl	Registra información de traspasos
teftsaldo.4gl	Registra información de saldos bancarios

F N M COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA

PLAN DE ACTIVIDADES

SIGLAS:	SISTEMA INTEGRAL DE INFORMACIÓN DE TESORERÍA	PÁGINA:
SITES		1 DE 1

ACTIVIDADES	SISTEMA	PERÍODO DE EJECUCIÓN DE LAS ACTIVIDADES														
		OCTUBRE				NOVIEMBRE				DICIEMBRE						
		1	2	3	4	1	2	3	4	1	2	3	4			
ESPECIFICACIÓN DE PROCESOS				X												
CREACIÓN DE PROGRAMAS								X								
PRUEBAS A LOS PROGRAMAS								X								
PRUEBAS AL SISTEMA														X		
IMPLANTACIÓN														X		
CAPACITACIÓN															X	
LIBERACIÓN																

FIGURA 5.1

5.3 Creación de Programas

Para la realización de programas y formas se ha definido un esquema estándar de trabajo de la siguiente manera:

- Todo programa debe iniciar con un breve texto que nos informe el nombre del sistema, módulo, objetivo, programa, analista, versión, etc.
- Todas las palabras reservadas del lenguaje deben ser escritas en mayúsculas.
- Debe ser modular, para la localización y modificación de las funciones sin afectar todo el funcionamiento del sistema.
- La definición de las funciones debe iniciar con "f" luego un consecutivo que se lleva en ese módulo y luego un subrayador "_" y finalmente la descripción de la función, ejemplo:

```
FUNCTION f01_alta()
```

- La definición de los reportes es igual a la definición de funciones solo con la variante que en lugar de una "f" lleva una "r".
- La definición de variables será de la siguiente manera:

- Globales

- ◆ gt_variable
- ◆ ga_arreglo
- ◆ gr_registro

donde:

g = global
t = trabajo
a = arreglo
r = registro

- Locales

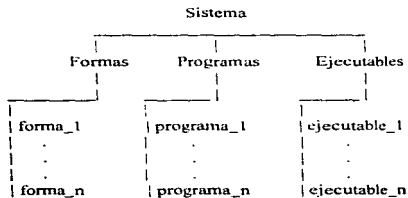
- ◆ lt_variable
- ◆ la_arreglo
- ◆ lr_registro
- ◆ pt_variable
- ◆ pr_registro

donde:

l = local
t = trabajo
a = arreglo
r = registro
p = paso de parámetro

Después del underscore "_" el identificador debe ser significativo al uso de la variable.

- Todas las fechas se manejarán con formato día/mes/año de la siguiente manera dd/mm/aaaa.
- Todas las formas deben desplegar información en modo inverso a la pantalla.
- Todos los títulos de la forma deben ser con mayúsculas.
- Todos los programas y formas se crearán en el subdirectorio correspondiente según la siguiente organización:



```

# Base de datos:      SIITESdb
# Sistema:
# Modulo:             Generacion de Mantenimientos.
# Pantalla:          tepinvers.per
# Prog. fuente:      tefinvers.4gl
# Prog. objeto:      tefinvers.4go
# Prog. ejecutable:  tefinvers.4gi
# Archivo de ayuda:  teainvers.msg
# Proceso:
# Objetivo:
#
# Analista:          Felipe Cruz Chavez
# Programador:       Felipe Cruz Chavez
# Fecha creacion:    11-16-1996
# Fecha ultima modificacion: 11-16-1996
# No. de version:    1

DATABASE SIITESdb

GLOBALS
DEFINE
    gt_cont2,
    gt_cont3          SMALLINT,
    gt_tabla          CHAR(20),
    gt_ban132        CHAR(1),
    gt_cd             CHAR(80)
END GLOBALS

DEFINE
    gr_old_reg,
    gr_consul,
    gr_bajmod,
    gr_sig,
    gr_pre,
    gr_record        RECORD
    t04inum_docto    LIKE tet04_inversiones.t04inum_docto,
    t06_fecdocto     LIKE tet06_documento.t06_fecdocto,
    t06region        LIKE tet06_documento.t06region,
    t04ibanco        LIKE tet04_inversiones.t04ibanco,
    t04isucur        LIKE tet04_inversiones.t04isucur,
    t04inumcta       LIKE tet04_inversiones.t04inumcta,
    t04_cotizacion   LIKE tet04_inversiones.t04_cotizacion,
    t04_plazo        LIKE tet04_inversiones.t04_plazo,
    t04_formainv     LIKE tet04_inversiones.t04_formainv,
    t04_ban_che      LIKE tet04_inversiones.t04_ban_che,
    t04_impinversion LIKE tet04_inversiones.t04_impinversion,
    t04_impinteres   LIKE tet04_inversiones.t04_impinteres

END RECORD.
gt_concepto LIKE tet04_inversiones.t04_concepto

MAIN
DEFINE
    pt_pasw CHAR(18).

```

```

pt_iton CHAR(78)

DEFER INTERRUPT
LET pt_pasw = ARG_VAL(1)
LET pt_iton = ARG_VAL(2)

CALL f0_menu(pt_pasw, pt_iton)
END MAIN

#####
# Muestra el primer nivel de menus #
#####
FUNCTION f0_menu(pt_nomb,pt_instr)
DEFINE
    lt_num_act .
    lt_num_reg INTEGER,
    lt_letra CHAR,
    lt_bandera SMALLINT,
    pt_nomb CHAR(18),
    pt_instr CHAR(78),
    lt_bcon SMALLINT,
    lt_npro CHAR(8),
    lt_ntit CHAR(80)

OPTIONS
INPUT ATTRIBUTE (RED,MAGENTA),
DISPLAY ATTRIBUTE (RED,MAGENTA),
HELP FILE "teainvers.hlp"

LET lt_npro = "TEPINVERS"
LET lt_ntit = "REGISTRO DE INVERSIONES"

PREPARE ejecproc FROM
"EXECUTE PROCEDURE (52_valperus(?,?,?))"
DECLARE curstop CURSOR FOR ejecproc

OPEN FORM TEPINVERS FROM "tepinvers"
DISPLAY FORM TEPINVERS
CALL f1_linea(lt_npro, lt_ntit, 32)

SELECT * FROM tet07_afecta
WHERE rowid = -1
INTO TEMP tmp_afecta07

MENU "MENU"
COMMAND "Alta" "Alta de un nuevo registro"
HELP 11

IF lt_bcon THEN
    CALL f3_closcurt)
    LET lt_bcon = FALSE
END IF

```

```
LET lt_letra = 'A'
OPEN curstop USING pt_nomb,pt_instr,lt_letra
  FOREACH curstop INTO lt_bandera
  END FOREACH
CLOSE curstop

IF lt_bandera THEN
  CALL f5_alta()
ELSE
  ERROR " USTED NO TIENE PERMISO PARA ACCESAR ESTA OPCION "
END IF

COMMAND "Bajas " "Elimina un registro."
HELP 12
IF lt_bcon THEN
LET lt_letra = 'B'
OPEN curstop USING pt_nomb,pt_instr,lt_letra
  FOREACH curstop INTO lt_bandera
  END FOREACH
CLOSE curstop

IF lt_bandera THEN
  CALL f6_baja(lt_num_act,lt_num_reg)
  RETURNING lt_bcon,lt_num_act,lt_num_reg
ELSE
  ERROR " USTED NO TIENE PERMISO PARA ACCESAR ESTA OPCION "
END IF
ELSE
  ERROR " PARA DAR DE BAJA UN REGISTRO PRIMERO DEBE CONSULTAR "
  NEXT OPTION "Consulta"
END IF

COMMAND "Modificacion " "Actualiza un registro."
HELP 13

IF lt_bcon THEN
LET lt_letra = 'M'

OPEN curstop USING pt_nomb,pt_instr,lt_letra
  FOREACH curstop INTO lt_bandera
  END FOREACH
CLOSE curstop

IF lt_bandera THEN
  CALL f7_modif(lt_num_act,lt_num_reg)
  RETURNING lt_num_act,lt_num_reg

  CALL f3_limpia()
  CALL f2_despl(lt_num_act,lt_num_reg)
ELSE
  ERROR " USTED NO TIENE PERMISO PARA ACCESAR ESTA OPCION "
END IF
ELSE
  ERROR " PARA MODIFICAR UN REGISTRO PRIMERO DEBE CONSULTAR "
```

```

NEXT OPTION "Consulta"
END IF

COMMAND "Consulta" "Consulta de registros"
HELP 14

IF It_bcon THEN
  CALL f9_clocsur()
  LET It_bcon = FALSE
END IF

CALL f8_consul()
RETURNING It_bcon,It_num_act,It_num_reg

COMMAND "Siguiente" "Ver siguiente registro en la lista"
HELP 15

IF It_bcon THEN
  CALL f9_sigreg(It_num_act,It_num_reg)
  RETURNING It_num_act,It_num_reg
ELSE
  ERROR " PARA VER SIGUIENTE REGISTRO PRIMERO DEBE CONSULTAR "
  NEXT OPTION "Consulta"
END IF

COMMAND "Previo" "Ver registro anterior en la lista"
HELP 16

IF It_bcon THEN
  CALL f9_prereg(It_num_act,It_num_reg)
  RETURNING It_num_act,It_num_reg
ELSE
  ERROR " PARA VER REGISTRO ANTERIOR PRIMERO DEBE CONSULTAR "
  NEXT OPTION "Consulta"
END IF

COMMAND "Concepto" "Editar el concepto del documento"
CALL desp_concep()

COMMAND "2a_pantalla" "Ver registro anterior en la lista"
HELP 16

IF It_bcon THEN
  CALL detalle(gr_record.t04_impinversion, "2", gr_record.t04inum_ducto,
gr_record.t04ibanco, gr_record.t04isucur, gr_record.t04inumcta,
gr_record.t04_cotizacion, gr_record.t04_plazo,
gr_record.t04_formainv, gr_record.t04_ban_che,
gr_record.t04_impinversion, gr_record.t04_impinteres)
ELSE
  ERROR " PARA VER REGISTRO ANTERIOR PRIMERO DEBE CONSULTAR "
  NEXT OPTION "Consulta"
END IF

COMMAND "Fin" "Fin de captura. Regresa a menu anterior"
IF It_bcon THEN

```



```

        CALL f3_closcur()
    END IF
    EXIT MENU
END MENU
END FUNCTION

```

```

#####
# Facilidad para registrar mas de un documento      =
#####

```

```

FUNCTION f5_alta()
    CALL f2_limpia()
    DISPLAY "DIGITE <ESC> PARA ACEPTAR REGISTRO" AT 1,1
    ATTRIBUTE (RED,REVERSE)
    DISPLAY "<CTRL-C> PARA ABORTAR OPERACION " AT 2,1
    ATTRIBUTE (RED,REVERSE)

    WHILE TRUE
        CASE f5_regis()
            WHEN 0
                ERROR "ALTA EFECTUADA..."
                SLEEP 2
            WHEN 1
                ERROR "ERROR EN LA CAPTURA. VERIFIQUE SUS DATOS"
                SLEEP 2
            WHEN 2
                ERROR "ALTA ABANDONADA. ULTIMOS DATOS IGNORADOS"
                EXIT WHILE
        END CASE
    END WHILE

    CLEAR FORM
END FUNCTION

```

```

#####
# Registrando documentos
#####

```

```

FUNCTION f5_regis()
    DEFINE
        lt_orden    CHAR(500),
        lt_intentos SMALLINT,
        lt_banco    CHAR(30),
        lt_sucursal CHAR(30),
        lt_region   CHAR(30),
        lt_cuenta   CHAR(30),
        lt_cla      CHAR(30),
        lt_tipodoc  CHAR(30),
        lt_concen   CHAR(30),
        lt_usuario  CHAR(15),
        lt_docto    INTEGER,
        lt_instru   LIKE tec13_instrum.c13iinstru,
        lt_instrumdesc LIKE tec13_instrum.c13_descrip,
        lt_instrumhor DATETIME HOUR TO MINUTE,
        lt_porcentaje LIKE tet09_cotizacion.t09_porcentaje,
        lt_interes  LIKE tet04_inversiones.t04_impinteres,

```

```
It_status SMALLINT
```

```
CLEAR FORM
```

```
LET gr_record.t06_fechacto = TODAY
DISPLAY BY NAME gr_record.t06_fechacto
LOCATE gr_concepto IN MEMORY
INPUT
```

```
#gr_record.t04inum_docto.
gr_record.t06_fechacto.
gr_record.t06iregion.
gr_record.t04ibanco.
gr_record.t04isucur.
gr_record.t04inumcta.
gr_record.t04_cotizacion.
gr_record.t04_plazo.
gr_record.t04_formainv.
gr_record.t04_ban_che.
gr_record.t04_impinversion.
gr_record.t04_impinteres.
gr_concepto
```

```
WITHOUT DEFAULTS
FROM
```

```
#tet04_inversiones.t04inum_docto.
tet06_documento.t06_fechacto.
tet06_documento.t06iregion.
tet04_inversiones.t04ibanco.
tet04_inversiones.t04isucur.
tet04_inversiones.t04inumcta.
tet04_inversiones.t04_cotizacion.
tet04_inversiones.t04_plazo.
tet04_inversiones.t04_formainv.
tet04_inversiones.t04_ban_che.
tet04_inversiones.t04_impinversion.
tet04_inversiones.t04_impinteres.
tet04_inversiones.t04_concepto
```

```
AFTER FIELD t06_fechacto
IF gr_record.t06_fechacto IS NULL THEN
  ERROR "LA FECHA NO PUEDE SER NULA"
  NEXT FIELD tet06_inversiones.t06_fechacto
ELSE
  END IF
```

```
BEFORE FIELD t06iregion
DISPLAY "[ZOOM]" AT 6,05 ATTRIBUTE(REVERSE)
```

```
AFTER FIELD t06iregion
IF It_intentos = 2 THEN
  IF gr_record.t06iregion IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET It_intentos = It_intentos + 1
  NEXT FIELD tet06_documento.t06iregion
```

```
ELSE
  SELECT c08_descrip INTO lt_region FROM tec08_region
  WHERE c08iregion = gr_record.t06iregion
  IF STATUS = NOTFOUND THEN
    ERROR "NO EXISTE REGION EN CATALOGO"
    NEXT FIELD tet06_documento.t06iregion
  ELSE
    DISPLAY BY NAME lt_region
  END IF
  LET lt_intentos = 0
END IF
ELSE
  CALL initia()
  EXIT INPUT
END IF

BEFORE FIELD t04ibanco
  DISPLAY "¡ZOOM!" AT 6,65 ATTRIBUTE(REVERSE)

AFTER FIELD t04ibanco
  IF lt_intentos < 2 THEN
    IF gr_record.t04ibanco IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET lt_intentos = lt_intentos + 1
      NEXT FIELD tet04_inversiones.t04ibanco
    ELSE
      SELECT c09_descrip INTO lt_banco FROM tec09_banco
      WHERE c09ibanco = gr_record.t04ibanco
      IF STATUS = NOTFOUND THEN
        ERROR "NO EXISTE BANCO EN CATALOGO"
        NEXT FIELD tet04_inversiones.t04ibanco
      ELSE
        DISPLAY BY NAME lt_banco
      END IF
      LET lt_intentos = 0
    END IF
  ELSE
    CALL initiat()
    EXIT INPUT
  END IF

BEFORE FIELD t04isucur
  DISPLAY "¡ZOOM!" AT 6,65 ATTRIBUTE(REVERSE)

AFTER FIELD t04isucur
  IF lt_intentos < 2 THEN
    IF gr_record.t04isucur IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET lt_intentos = lt_intentos + 1
      NEXT FIELD tet04_inversiones.t04isucur
    ELSE
      SELECT c11_descrip INTO lt_sucursal FROM tec11_sucur
      WHERE c11ibanco = gr_record.t04ibanco
      AND c11isucur = gr_record.t04isucur
```

```
IF STATUS = NOTFOUND THEN
  ERROR "NO EXISTE SUCURSAL DE ESE BANCO EN CATALOGO"
  NEXT FIELD tet04_inversiones.t04isucur
ELSE
  DISPLAY BY NAME It_sucursal
END IF
LET It_intentos = 0
END IF
ELSE IF
  CALL inicia()
  EXIT INPUT
END IF

AFTER FIELD t04inumcta
IF It_intentos < 2 THEN
  IF gr_record.t04inumcta IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET It_intentos = It_intentos + 1
    NEXT FIELD tet04_inversiones.t04inumcta
  ELSE
    SELECT c04_descrip INTO It_cuenta FROM tec04_ctab
    WHERE c04ibanco = gr_record.t04ibanco
    AND c04isucur = gr_record.t04isucur
    AND c04inumcta = gr_record.t04inumcta
    IF STATUS = NOTFOUND THEN
      ERROR "NO EXISTE CUENTA BANCARIA"
      NEXT FIELD tet04_inversiones.t04inumcta
    ELSE
      DISPLAY BY NAME It_cuenta
    END IF
    LET It_intentos = 0
  END IF
ELSE
  CALL inicia()
  EXIT INPUT
END IF

AFTER FIELD t04_cotizacion
IF It_intentos < 2 THEN
  IF gr_record.t04_cotizacion IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET It_intentos = It_intentos + 1
    NEXT FIELD tet04_inversiones.t04_cotizacion
  ELSE
    IF b_cotiza(gr_record.t04ibanco, gr_record.t06_fecdocto, gr_record.t04_cotizacion) THEN
      CALL d_cotiza(gr_record.t04ibanco, gr_record.t06_fecdocto,
        gr_record.t04_cotizacion)
      RETURNING It_instru, It_instrumdesc,
        It_porcentaje, It_instrumhor
      DISPLAY BY NAME It_instrumdesc, It_porcentaje
    ELSE
      ERROR "La cotizacion no existe para este dia.",
```

```
                " favor de registrarla"
                SLEEP 2
                NEXT FIELD tet04_inversiones.t04_cotizacion
            END IF
            LET It_intentos = 0
        END IF
    ELSE
        CALL inicia()
        EXIT INPUT
    END IF

AFTER FIELD t04_plazo
IF It_intentos < 2 THEN
    IF gr_record.t04_plazo IS NULL THEN
        ERROR "EL CAMPO NO PUEDE SER NULO!!!"
        LET It_intentos = It_intentos + 1
        NEXT FIELD tet04_inversiones.t04_plazo
    ELSE
        LET It_intentos = 0
    END IF
ELSE
    CALL inicia()
    EXIT INPUT
END IF

AFTER FIELD t04_formainv
IF It_intentos < 2 THEN
    IF gr_record.t04_formainv IS NULL THEN
        ERROR "EL CAMPO NO PUEDE SER NULO!!!"
        LET It_intentos = It_intentos + 1
        NEXT FIELD tet04_inversiones.t04_formainv
    ELSE
        LET It_intentos = 0
    END IF
ELSE
    CALL inicia()
    EXIT INPUT
END IF

AFTER FIELD t04_ban_che
IF It_intentos < 2 THEN
    IF gr_record.t04_ban_che IS NULL THEN
        ERROR "EL CAMPO NO PUEDE SER NULO!!!"
        LET It_intentos = It_intentos + 1
        NEXT FIELD tet04_inversiones.t04_ban_che
    ELSE
        LET It_intentos = 0
    END IF
ELSE
    CALL inicia()
    EXIT INPUT
END IF
```

```
AFTER FIELD t04_impinversion
IF It_intentos < 2 THEN
  IF gr_record.t04_impinversion IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET It_intentos = It_intentos + 1
    NEXT FIELD tet04_inversiones.t04_impinversion
  ELSE
    CALL cal_intereses(It_porcentaje, gr_record.t04_impinversion,
                      gr_record.t04_plazo)
    RETURNING gr_record.t04_impinteres
    DISPLAY BY NAME gr_record.t04_impinteres
    LET It_intentos = 0
  END IF
ELSE
  CALL inicia()
  EXIT INPUT
END IF
```

```
AFTER FIELD t04_impinteres
IF It_intentos < 2 THEN
  IF gr_record.t04_impinteres IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET It_intentos = It_intentos + 1
    NEXT FIELD tet04_inversiones.t04_impinteres
  ELSE
    LET It_intentos = 0
  END IF
ELSE
  CALL inicia()
  EXIT INPUT
END IF
```

```
ON KEY (CONTROL-W,16)
CASE
  WHEN INFIELD(t04inum_docto)
    CALL SHOWHELP( 10)
  WHEN INFIELD(t04isucur)
    CALL SHOWHELP( 20)
  WHEN INFIELD(t04ibanco)
    CALL SHOWHELP( 30)
  WHEN INFIELD(t04inumcta)
    CALL SHOWHELP( 40)
  WHEN INFIELD(t04_cotizacion)
    CALL SHOWHELP( 50)
  WHEN INFIELD(t04_plazo)
    CALL SHOWHELP( 60)
  WHEN INFIELD(t04_formainv)
    CALL SHOWHELP( 70)
  WHEN INFIELD(t04_ban_che)
    CALL SHOWHELP( 80)
  WHEN INFIELD(t04_impinversion)
    CALL SHOWHELP( 90)
  WHEN INFIELD(t04_impinteres)
    CALL SHOWHELP( 100)
```

```
END CASE

ON KEY (CONTROL-Z)
CASE
  WHEN INFIELD(t06iregion)
    CALL f1_region()
    RETURNING gr_record.t06iregion, lt_region
    DISPLAY BY NAME gr_record.t06iregion, lt_region
  WHEN INFIELD(t04ibanco)
    CALL f1_banco()
    RETURNING gr_record.t04ibanco, lt_banco
    DISPLAY BY NAME gr_record.t04ibanco, lt_banco
  WHEN INFIELD(t04isucur)
    CALL f2_sucursal(gr_record.t04ibanco)
    RETURNING gr_record.t04isucur, lt_sucursal
    DISPLAY BY NAME gr_record.t04isucur, lt_sucursal
  WHEN INFIELD(t04inumcta)
    CALL f4_cuentas(gr_record.t04ibanco, gr_record.t04isucur)
    RETURNING gr_record.t04inumcta, lt_cuenta
    DISPLAY BY NAME gr_record.t04inumcta, lt_cuenta
  WHEN INFIELD(t04_cotizacion)
    CALL f11_cotiza(gr_record.t04ibanco, gr_record.t06_fecha)
    RETURNING gr_record.t04_cotizacion, lt_instru, lt_instrumdesc,
              lt_instrumhor, lt_porcentaje
    DISPLAY BY NAME gr_record.t04_cotizacion, lt_instrumdesc,
                  lt_porcentaje

  END CASE

ON KEY (CONTROL-C)
  LET INT_FLAG = TRUE
  EXIT INPUT

AFTER INPUT
  IF INT_FLAG = TRUE THEN
    EXIT INPUT
  ELSE
    CALL cal_intereses(lt_porcentaje, gr_record.t04_impinversion,
                      gr_record.t04_plazo)
    RETURNING gr_record.t04_impinteres
    DISPLAY BY NAME gr_record.t04_impinteres
  END IF
END INPUT

IF INT_FLAG THEN
  LET INT_FLAG = FALSE
  DISPLAY " " AT 6,69
  RETURN 2
END IF

LET lt_usuario = 'SITES'

CALL detalle(gr_record.t04_impinversion, "1", gr_record.t04inum_docto,
```

```

        gr_record.t04ibanco, gr_record.t04isucur, gr_record.t04inumcta,
        gr_record.t04_cotizacion, gr_record.t04_plazo,
        gr_record.t04_formainv, gr_record.t04_ban_che,
        gr_record.t04_impinversion, gr_record.t04_impinteres)
RETURNING lt_status
IF lt_status = 0 THEN
    BEGIN WORK
    INSERT INTO tet06_documento
    VALUES (0,lt_usuario,gr_record.t06iregion,gr_record.t06_feedocto,'I','P')
    LET lt_docto = SQLCA.SQLERRD[2]
    LET gr_record.t04inum_docto = lt_docto
    DISPLAY lt_docto TO t04inum_docto SLEEP 1

    INSERT INTO tet04_inversiones
    VALUES (lt_docto,
            gr_record.t04ibanco,
            gr_record.t04isucur,
            gr_record.t04inumcta,
            gr_record.t04_cotizacion,
            gr_record.t04_plazo,
            gr_record.t04_formainv,
            gr_record.t04_ban_che,
            gr_record.t04_impinversion,
            gr_record.t04_impinteres,
            gr_concepto)

    UPDATE tmp_afecta07 SET t07inum_docto = lt_docto
    INSERT INTO tet07_afecta SELECT * FROM tmp_afecta07
    IF STATUS = 0 THEN
        COMMIT WORK
    ELSE
        ROLLBACK WORK
    END IF

END IF
IF STATUS <> 0 THEN
    RETURN 1
ELSE
    RETURN 0
END IF
END FUNCTION

#####
# Dando de baja a un registro #
#####
FUNCTION t6_baja(lt_num_act,lt_num_reg)
    DEFINE
        lt_bcon SMALLINT,
        lt_num_act ,
        lt_num_reg ,
        lt_num_rec INTEGER,
        lt_resp CHAR(1)

```



```
FETCH CURRENT cugr_consul INTO lt_num_rec

OPEN cugr_consulta USING lt_num_rec
FETCH FIRST cugr_consulta INTO

    gr_bajmod.t04inum_docto,
    gr_bajmod.t04ibanco,
    gr_bajmod.t04iscur,
    gr_bajmod.t04inumcta,
    gr_bajmod.t04_cotizacion,
    gr_bajmod.t04_plazo,
    gr_bajmod.t04_formainv,
    gr_bajmod.t04_ban_che,
    gr_bajmod.t04_impinversion,
    gr_bajmod.t04_impinteres

OPEN cugr_consult2 USING gr_bajmod.t04inum_docto
FETCH FIRST cugr_consult2 INTO
    gr_bajmod.t06iregion,
    gr_bajmod.t06_fecdocto

CALL f1_display(gr_bajmod.*)
CALL f2_limpiar()

OPEN WINDOW sdwbaja AT 13,15
WITH 5 ROWS,40 COLUMNS
ATTRIBUTE(BORDER,MENU LINE LAST - 1)

DISPLAY " DESEA DAR DE BAJA ESTE REGISTRO " AT 2,4
ATTRIBUTE(RED)
MENU ""
BEFORE MENU
    DISPLAY " " AT 3,3

COMMAND KEY ("N") " No "
    LET lt_resp = "N"
    EXIT MENU

COMMAND KEY ("S") " Si "
    LET lt_resp = "S"
    EXIT MENU

END MENU

CLOSE WINDOW sdwbaja

IF lt_resp = "S" THEN
    LET lt_num_reg = lt_num_reg - 1

#borrando en inversiones
DELETE FROM tet04_inversiones
WHERE ROWID = lt_num_reg
#borrando en documentos (el maestro)
DELETE FROM tet06_documento
```

```
WHERE t06inum_docto = gr_bajmod.t04inum_docto
FETCH NEXT cugr_consul INTO lt_num_rec
IF STATUS = NOTFOUND THEN
  FETCH PRIOR cugr_consul INTO lt_num_rec
  IF STATUS = NOTFOUND THEN
    LET lt_num_rec = 0
  ELSE
    LET lt_num_act = lt_num_act - 1
  OPEN cugr_consulta USING lt_num_rec
  FETCH FIRST cugr_consulta INTO
    gr_bajmod.t04inum_docto,
    gr_bajmod.t04ibanco,
    gr_bajmod.t04isucur,
    gr_bajmod.t04inumcta,
    gr_bajmod.t04_cotizacion,
    gr_bajmod.t04_plazo,
    gr_bajmod.t04_formainv,
    gr_bajmod.t04_ban_che,
    gr_bajmod.t04_impinversion,
    gr_bajmod.t04_impinteres
  OPEN cugr_consult2 USING gr_bajmod.t04inum_docto
  FETCH FIRST cugr_consult2 INTO
    gr_bajmod.t06iregion,
    gr_bajmod.t06_fecdocto
  CALL f11_display(gr_bajmod *)
END IF
ELSE
  LET lt_num_act = lt_num_act
  OPEN cugr_consulta USING lt_num_rec
  FETCH FIRST cugr_consulta INTO
    gr_bajmod.t04inum_docto,
    gr_bajmod.t04ibanco,
    gr_bajmod.t04isucur,
    gr_bajmod.t04inumcta,
    gr_bajmod.t04_cotizacion,
    gr_bajmod.t04_plazo,
    gr_bajmod.t04_formainv,
    gr_bajmod.t04_ban_che,
    gr_bajmod.t04_impinversion,
    gr_bajmod.t04_impinteres
  OPEN cugr_consult2 USING gr_bajmod.t04inum_docto
  FETCH FIRST cugr_consult2 INTO
    gr_bajmod.t06iregion,
    gr_bajmod.t06_fecdocto
```

```

CALL f1_display(gr_bajmod.*)
END IF

ERROR "REGISTRO ELIMINADO"
ELSE
ERROR "BAJA ABANDONADA"
END IF

IF lt_num_reg = 0 THEN
LET lt_bcon = FALSE
CLOSE cugr_consulta
CLOSE cugr_consul
CLEAR FORM
DISPLAY " " AT 23,60
ELSE
LET lt_bcon = TRUE
CALL f3_limpiar()
CALL f12_desplii(lt_num_act,lt_num_reg)
END IF

SLEEP 2
RETURN lt_bcon,lt_num_act,lt_num_reg
END FUNCTION

#####
# Desplegando el numero de registros encontrados #
#####
FUNCTION f12_desplii(lt_num_act,lt_num_reg)
DEFINE
lt_num_act ,
lt_num_reg INTEGER

DISPLAY "REGISTRO ".lt_num_act," DE ".lt_num_reg AT 23,60
ATTRIBUTE(RED,REVERSE)
END FUNCTION

#####
# Desplegando el encabezado #
#####
FUNCTION f1_linea(pt_nompro, pt_titpro, pt_coltit)
DEFINE
pt_nompro CHAR(8),
pt_titpro CHAR(80),
pt_coltit SMALLINT,
pt_ferro CHAR(80),
pt_apli CHAR(80),
lt_fec DATE,
lt_hor DATETIME HOUR TO MINUTE

LET lt_fec = TODAY
LET lt_hor = CURRENT HOUR TO MINUTE
LET pt_coltit = ((80 - LENGTH(pt_titpro)) / 2)

CALL fgl_drawbox(5, 79, 3, 1)

```

```

CALL fgl_drawbox(20, 79, 3, 1)
DISPLAY "r, lt_rec USING "dd/mm/yy", "]" AT 4, 4 ATTRIBUTE (MAGENTA)
DISPLAY "[", lt_hor, "]" AT 4, 70 ATTRIBUTE (MAGENTA)
DISPLAY pt_nompro CLIPPED AT 6, 5
DISPLAY pt_titpro CLIPPED AT 6, pt_coltit
LET pt_apli = "S I T E S"
LET pt_coltit = ((80 - LENGTH(pt_apli)) / 2)
DISPLAY pt_apli CLIPPED AT 5, pt_coltit
LET pt_ferro = "FERROCARRILES NACIONALES DE MEXICO"
LET pt_coltit = ((80 - LENGTH(pt_ferro)) / 2)
DISPLAY pt_ferro CLIPPED AT 4, pt_coltit
END FUNCTION

```

```

#####
# Encontrando el siguiente registro      #
#####
FUNCTION f9_sigreglt_num_act,lt_num_reg)

```

```

    DEFINE
        lt_num_act ,
        lt_num_reg ,
        lt_num_rec INTEGER

```

```

CALL f3_limpiar()
FETCH NEXT cugr_consul INTO lt_num_rec

```

```

OPEN cugr_consulta USING lt_num_rec
FETCH cugr_consulta INTO
    gr_sig.t04inum_docto,
    gr_sig.t04ibanco,
    gr_sig.t04isucur,
    gr_sig.t04inumcta,
    gr_sig.t04_cotizacion,
    gr_sig.t04_plazo,
    gr_sig.t04_formainv,
    gr_sig.t04_ban_che,
    gr_sig.t04_impinversion,
    gr_sig.t04_impinteres

```

```

LET lt_num_act = lt_num_act + 1

```

```

IF STATUS = NOTFOUND THEN
    ERROR "ESTE ES EL ULTIMO REGISTRO DE LA LISTA"
    FETCH CURRENT cugr_consul INTO lt_num_rec
    OPEN cugr_consulta USING lt_num_rec

```

```

    FETCH LAST cugr_consulta INTO
        gr_sig.t04inum_docto,
        gr_sig.t04ibanco,
        gr_sig.t04isucur,
        gr_sig.t04inumcta,
        gr_sig.t04_cotizacion,
        gr_sig.t04_plazo,
        gr_sig.t04_formainv,
        gr_sig.t04_ban_che,

```

```

gr_sig.t04_impinversion,
gr_sig.t04_impinteres

OPEN cugr_consult2 USING gr_sig.t04inum_docto
FETCH FIRST cugr_consult2 INTO
gr_sig.t06iregion,
gr_sig.t06_fecdocto

LET lt_num_act = lt_num_reg
END IF
OPEN cugr_consult2 USING gr_sig.t04inum_docto
FETCH FIRST cugr_consult2 INTO
gr_sig.t06iregion,
gr_sig.t06_fecdocto

IF lt_num_act > lt_num_reg THEN
LET lt_num_act = lt_num_reg
END IF

CALL f12_despli(lt_num_act,lt_num_reg)
CALL f11_display(gr_sig.*)

RETURN lt_num_act,lt_num_reg
END FUNCTION

#####
# Cerrando cursors                                u
#####
FUNCTION f3_closcur()
CLOSE cugr_consul
CLOSE cugr_consulta
END FUNCTION

#####
# Inicializando variables                          #
#####
FUNCTION inicia()
ERROR "YA REALIZO MAS DE TRES INTENTOS INFRUCTUOSOS. TRATE DESPUES"
SLEEP 2
INITIALIZE gr_record TO NULL
LET int_flag = TRUE
END FUNCTION

#####
# FUNCION : Validacion de existencia de cotizacion #
#####
FUNCTION b_cotiza(pt_banco, pt_fecha, pt_cotiza)
DEFINE
lt_cotiza SMALLINT,
pt_banco LIKE tec09_banco.c09ibanco,
pt_fecha LIKE tet09_cotizacion.t09ifecha,
pt_cotiza LIKE tet09_cotizacion.t09inum_cot
LET lt_cotiza = 0

```

```

SELECT COUNT(*) INTO It_cotiza FROM tet09_cotizacion
WHERE t09ibanco = pt_banco
AND t09ifecha = pt_fecha
AND t09inum_cot = pt_cotiza
IF It_cotiza = 0 THEN
    RETURN FALSE
ELSE
    RETURN TRUE
END IF

END FUNCTION

#####
# Calcula los intereses a recibir a partir de la      #
# la inversion y el porcentaje de rendimiento      #
#####
FUNCTION cal_intereses(pt_porcentaje, pt_monto, pt_periodos)
DEFINE
    pt_periodos LIKE tet04_inversiones.t04_plazo,
    pt_porcentaje LIKE tet09_cotizacion.t09_porcentaje,
    pt_monto LIKE tet04_inversiones.t04_impinversion,
    It_interes LIKE tet04_inversiones.t04_impinteres

    LET It_interes = (((pt_porcentaje/100) * pt_monto) / 360 ) * pt_periodos

RETURN It_interes
END FUNCTION

#####
# Captura de concepto      #
#####
FUNCTION desp_concep()

INPUT gt_concepto WITHOUT DEFAULTS FROM t04_concepto
BEFORE FIELD t04_concepto
    DISPLAY "Teclee ! para consultar concepto " AT 23,1
END INPUT

END FUNCTION

```

5.4 Pruebas

En esta fase de pruebas se puede llevar mucho tiempo si es que no se hicieron eficientemente, el análisis, el diseño y la programación, por otro lado, si no se realizó un trabajo perfecto de estas fases, puede volverse iterativo; la primera tanda de pruebas muestra la presencia de errores, y las posteriores verifican si los programas corregidos funcionan correctamente. El proceso de desarrollar casos de prueba de aceptación puede llevarse a cabo en paralelo con las actividades de implantación del diseño y de la programación, para que cuando los programadores terminen de escribir sus programas y de realizar sus propias pruebas locales, el equipo del analista/usuario esté listo con sus propios casos de prueba.

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

Además de estos conceptos básicos, cabe mencionar la importancia de los siguientes tipos de prueba:

- **Prueba Funcional:** Esta es la forma más común de prueba; su propósito es asegurar que el sistema realiza sus funciones normales de manera correcta. Así, los casos de prueba se desarrollan y se alimentan al sistema, las salidas se examinan para ver si son correctas.
- **Prueba de Recuperación:** El propósito de este tipo de prueba es asegurar que el sistema pueda recuperarse adecuadamente de diversos tipos de fallas. Esto es de particular importancia en los sistemas en línea grandes, al igual que en varios tipos de sistemas de tiempo real que controlan dispositivos físicos y/o procesos de fabricación.
- **Prueba de Desempeño:** El propósito de este tipo de prueba es, asegurar que el sistema pueda manejar el volumen de datos y transacciones de entrada especificados en el modelo de implantación del usuario, además de asegurar que tenga el tiempo de respuesta requerido.

Existe un último concepto del que se debe estar al tanto: la noción de prueba exhaustiva. En el proyecto ideal, se generarían casos de prueba para cubrir cada entrada posible y cada combinación posible de situaciones que el sistema pudiera enfrentar alguna vez; luego se probaría de manera exhaustiva para asegurar que su comportamiento sea perfecto.

5.5 Implantación

Esta actividad incluye la codificación y la integración de módulos en un esqueleto progresivamente más completo del sistema final. Por eso, la actividad de implantación incluye tanto la programación como la implantación.

La puesta en marcha exitosa posiblemente no garantice mejoras en la empresa (esta es una situación del diseño), pero la instalación inadecuada evitará las mejoras.

A continuación se analizan tres aspectos de la puesta en marcha que incluyen la capacitación del personal, los procedimientos de conversión y la revisión posterior a la implantación.

5.5.1 Capacitación

Incluso los sistemas bien diseñados y técnicamente elegantes pueden tener éxito o fallar debido a la forma en que se operan y se utilizan, por lo tanto, la calidad de la capacitación del personal involucrado con el sistema en varias de sus facetas, ayuda o dificulta y puede obstaculizar por entero el éxito de la implantación de un sistema de información. Las personas que trabajan con el sistema deben conocer con detalle las funciones que desempeñarán, cómo utilizarán el sistema de lo que éste hará o no, tanto los operadores como los usuarios necesitan capacitación.

Capacitación de operadores de sistemas. Muchos sistemas dependen del personal del centro de cómputo, quien tiene la responsabilidad de mantener el equipo en buenas condiciones, así como de proporcionar el servicio necesario. Su capacitación debe garantizar que están en condición para manejar todas las operaciones posibles, tanto las de rutina como las extraordinarias. Si el sistema requiere la instalación del equipo nuevo, por ejemplo, de un nuevo sistema de cómputo de terminales especiales o de equipo diferente de entrada de datos, la capacitación del operador deberá incluir aspectos específicos) o la obtención de respuestas a las preguntas y borrar registros de datos. La parte más importante de los sistemas incluye éste conjunto de actividades; por lo tanto, la mayor parte del tiempo de capacitación se dedicará a ésta área.

De vez en cuando los usuarios deberán preparar discos, cargar papel o cambiar las cintas en las impresoras. Ningún programa de capacitación estará completo si no dedica algún tiempo a las actividades de mantenimiento de sistemas. Si un sistema de microcomputadoras o de entrada de datos emplea discos, los usuarios deberán recibir instrucción respecto al método de formatear y probar discos. Así mismo, deberán llevar a cabo cambios de cintas, limpieza del equipo, etc. No es suficiente incluir simplemente esta información en un manual aunque es esencial para proporcionar materiales de referencias posteriores.

Cómo lo demuestra el análisis anterior, existen dos aspectos para la capacitación del usuario: uno, la familiaridad con el sistema de procesamiento (es decir, el equipo usado para los datos de entrada ó su procesamiento), y dos, la capacitación en el empleo de la aplicación, es decir, el software que acepta los datos los procesa y produce los resultados.

- **Métodos fundamentales.** Es la manera en que deben de encender el equipo o la forma de operarlo. Los operadores también deben recibir instrucciones sobre de los errores comunes que pueden ocurrir cómo detectarlos y tomar las medidas necesarias cuando se presentan, deben tener a la mano los nombres y los números de teléfonos de las personas con quien deberán ponerse en contacto cuando surjan problemas inesperados o poco comunes. Los operadores se familiarizan con las acciones que deben llevar a cabo como montar los discos magnéticos o las cintas, copiar los archivos, cambiar las formas de impresión o activar sistemas de comunicación.
- **Capacitación de usuarios.** La capacitación del usuario puede incluir el empleo del equipo, en especial, por ejemplo, en el caso de que se emplee una microcomputadora y el individuo sea simultáneamente operador y usuario. En estos casos los usuarios deben recibir instrucciones primero, respecto a la operación del equipo. Durante la capacitación del usuario también deben instruirlo en la solución del problema dentro del sistema determinando, si cuando surge alguno se origina en el equipo, por el software o en alguna acción que se haya llevado a cabo al operar el sistema, no hay nada más frustrante de trabajar con su sistema, detectar un problema y no ser capaz de determinar si se originó por una falla del operador o en el sistema mismo. Esta frustración se debe evitar desde el principio, es decir, desde la capacitación.

La mayor parte de la capacitación de los usuarios radica en la operación del sistema mismo. La capacitación en la codificación de datos destaca los métodos que han de seguirse en la captación de datos a partir de transacciones, o en la preparación de datos necesarios para las actividades en la toma de decisiones. Las actividades del manejo de datos reciben la máxima atención dentro de la capacitación de usuarios son la entrada de nuevo datos (cómo almacenar nuevas transacciones), la edición de datos (cómo cambiar datos almacenados anteriormente), la formulación de consultas (la búsqueda de registro específicos).

- **Método de capacitación.** La capacitación de operar y usuarios se pueden lograr mediante diferentes formas. Las actividades se pueden llevar a cabo en las mismas instalaciones de venta de equipos, en lugares alquilados, por ejemplo, en hoteles, universidades o instalaciones que pertenezcan a la compañía donde labora el empleado. Los métodos y el contenido de la capacitación varía a menudo y dependen del origen y de la ubicación de la capacitación.

5.5.2 Conversión

La conversión es el proceso de cambio al sistema antiguo al nuevo, existen cuatro métodos para manejar la conversión de sistemas como son sistemas paralelos, cambio directo, sistema piloto y método de etapas. En general, la conversión de sistemas se debe realizar tan rápido como sea posible; los periodos largos de conversión aumentan al posible frustración de la dificultad de la tarea para todas las personas que participan, incluyendo tanto a los analistas, como a los usuarios.

- **Sistemas paralelos.** El método más seguro de conversión de sistema viejo a uno nuevo radica en operar ambos sistemas en forma paralela, lo que significa que los usuarios continúan operando el viejo sistema de la manera acostumbrada, pero también empiezan a emplear el nuevo sistema, permite la puesta en marcha más segura, en caso de que haya dificultades, pero no se pueden ignorar los costos y riesgos de no ensayar adecuadamente el nuevo sistema
- **Cambio directo.** El método de cambio directo convierte el sistema viejo al nuevo de manera repentina, en ocasiones el lapso de un fin de semana o incluso de un día para otro. El sistema viejo se emplea hasta el día de conversión planeado y entonces se reemplaza por el sistema nuevo. No existen actividades paralelas. El cambio directo requiere de una planificación cuidadosa por adelantado. Las sesiones de capacitación se deben programar y mantener, la instalación de todo el equipo debe ser puntual con fechas precisas dentro del programa, para corregir las dificultades que puedan ocurrir.
- **Enfoque piloto.** Cuando los sistemas nuevos implican también técnicas nuevas o cambios drásticos en el desempeño de la empresa se prefiere el enfoque piloto. Con frecuencia en este método una versión práctica del sistema se pone en marcha en una parte de la empresa; por ejemplo, una sola área o departamento de trabajo. Los usuarios de ésta área saben por lo general que están llevando a cabo una prueba piloto de un nuevo sistema y que pueden practicarse cambio para mejorarlo. Cuando se calcula que el sistema está completo se instala en toda la compañía ya sea de una sola vez (cambio directo) o en forma gradual (por etapas).
- **Método por etapas.** El método por etapas se emplea cuando no es posible instalar un sistema nuevo en toda la compañía de manera simultánea. La conversión de archivos, la capacitación de personal o la llegada del equipo pueden forzar a la puesta en marcha gradual que se realiza en semanas o en meses. Algunos usuarios obtendrán ventajas del sistema nuevo antes que otros, cuando los sistemas se utilicen los métodos por etapas deben trabajar adecuadamente en la primera conversión y en las que siguen.



CAPÍTULO VI

CAPÍTULO VI MANTENIMIENTO

Históricamente en el área de sistemas el término "mantenimiento" se ha aplicado al proceso de modificar un programa cuando ya se ha entregado y está en uso. Esas modificaciones pueden implicar cambios sencillos para corregir errores de codificación, cambios mayores para corregir errores de diseño o reescrituras drásticas para corregir errores de especificación o introducir nuevos requisitos. A la actividad de hacer cambios se le da el nombre de fase de mantenimiento.

Una vez que el sistema ha sido instalado, existe un período de garantía durante el cual el proveedor es responsable del mantenimiento. Usualmente es de 90 días, aunque los términos específicos están sujetos a la negociación del contrato. Después de ese tiempo, el comprador tiene la opción de obtener el mantenimiento de varias fuentes.

También se puede obtener servicio por medio de compañías especializadas. Las compañías de mantenimiento por parte de terceros, como se llama a estas empresas, con frecuencia dan servicio a comunidades pequeñas, donde para los fabricantes no es costable mantener oficinas. Por otra parte, los proveedores de sistemas integrales que entregan e instalan combinaciones de hardware y software, pero que no fabrican dicho equipo, sugieren la contratación de empresas de mantenimiento por parte de terceros con las que trabajan directamente.

Una vez que el sistema es declarado operativo, es poco probable que funcione de forma satisfactoria durante mucho tiempo. La experiencia ha demostrado que lo normal es que en los sistemas en uso vayan surgiendo sugerencias para mejorar su funcionamiento.

Durante la vida del sistema, sus requisitos originales se modificarán para reflejar necesidades cambiantes, el ambiente del sistema cambiará y surgirán errores ocultos, no descubiertos durante la comprobación del sistema. El mantenimiento es inevitable, por eso los sistemas se deben diseñar y aplicar de modo que se minimicen los problemas en este aspecto.

Por otra parte, los requisitos de los sistemas varían con el tiempo. Por esta razón los sistemas necesitan cierto mantenimiento periódico. El mantenimiento requiere reespecificar una parte del sistema con corrección a los programas y a la documentación de acuerdo a estas modificaciones. También es necesario cierta cantidad de pruebas adicionales, a fin de comprobar que ni el hardware ni el software se han degenerado con el tiempo.

Los costos de mantenimiento son muy difíciles de estimar con anticipación. La evidencia muestra que los costos de mantenimiento son lo más cuantiosos del desarrollo y uso de los sistemas.

A medida que los sistemas envejecen, se debe dedicar relativamente más esfuerzo a su mantenimiento. Uno de los principales problemas que surgen es que esos sistemas pueden

estar escritos en lenguajes de programación que ya no se utilizan en el desarrollo de los nuevos sistemas, es decir se deben actualizar tecnológicamente.

En verdad vale la pena invertir tiempo y esfuerzo cuando se diseña y aplica un sistema para reducir el mantenimiento y con ello los costos totales.

Cuando el sistema entra en su fase operativa se realizan las auditorías postimplantación para verificar si satisface los criterios de diseño. Dentro de los primeros tres meses de haberse adoptado un nuevo sistema, se realiza una auditoría en la cual se compara el funcionamiento del sistema con sus objetivos originales, para comprobar si éste está cumpliendo con lo esperado.

Puesto que es una violación al concepto del control que un grupo se haga una auditoría a sí mismo, la auditoría la hace un equipo diferente al que desarrolló el sistema. Algunas empresas asignan esa responsabilidad al departamento de auditoría interna dependiente de la división de contraloría; otras establecen un grupo dentro del departamento de computación para que realicen las auditorías.

En la fase del mantenimiento se hacen 2 tipos de cambios a los sistemas en funcionamiento:

- Los cambios que se hacen para mejorar la eficiencia del sistema.
- Los cambios necesarios por actualización tecnológica (la reprogramación como resultado de adquirir nuevo equipo periférico).

La fase del mantenimiento no se debe considerar como la siguiente etapa del desarrollo. Los sistemas se planean para ser implantados en varias etapas con el fin de lograr niveles de complejidad crecientes; sin embargo hay ocasiones en que se deben hacer cambios menores en cada etapa.

El mantenimiento del software es mucho más que la "corrección de errores". A continuación se describen los diferentes tipos de mantenimientos que existen.

6.1. Tipos de mantenimiento.

6.1.1. Correctivo

La primera actividad del mantenimiento se da ocasionalmente cuando la prueba del software no haya descubierto todos los errores latentes de un sistema. Durante el uso del sistema se encontrarán errores, los cuales deben ser informados al equipo de desarrollo. El proceso que incluye el diagnóstico y corrección de uno o más errores se denomina mantenimiento correctivo.

6.1.2 Preventivo

Este tipo de mantenimiento es para prevenir errores. Se da cuando cambia el software para mejorar una futura facilidad de mantenimiento. También se puede considerar el mantenimiento a la información que se maneja para actualizarla y los resultados dados por el sistema sean los correctos.

6.1.3 Perfectivo

Esta actividad de mantenimiento se da cuando un paquete de software tiene éxito. A medida que se usa el software, se reciben de los usuarios recomendaciones sobre nuevas posibilidades acerca de modificaciones a funciones ya existentes. Para satisfacer estas peticiones se lleva a cabo el mantenimiento perfectivo.

El mantenimiento perfectivo comprende también los cambios solicitados al programador del sistema.

6.1.4 Aumentativo

Este tipo de mantenimiento se da cuando se incluyen nuevas funciones que no se contemplaron al inicio del desarrollo del sistema y surgen como una necesidad del usuario.

6.1.5 Tecnológico

Esta actividad que contribuye al mantenimiento se da debido a todo cambio importante en la informática. Si en un ciclo de 36 meses surgen nuevas generaciones de hardware, regularmente aparecen nuevos sistemas operativos o nuevas versiones de los antiguos; y frecuentemente se mejoran o modifican los equipos periféricos y otros elementos de sistemas.

6.1.6 Adaptativo

La vida útil estimada del software de aplicación puede fácilmente sobrepasar los diez años, pero considerando la evolución del ambiente, en la práctica éste puede volverse obsoleto. Por lo tanto, el mantenimiento adaptativo es una actividad que modifica al software para que las interacciones sean adecuadamente con su entorno cambiante.

El mantenimiento adaptativo se debe a cambios en el ambiente del programa y a la adaptación de nuevas unidades o módulos.

Un estudio hecho por Lientz y Swanson (1980) descubrió que alrededor del 65% del mantenimiento era perfectivo, el 18% adaptativo, y el 17% correctivo.

6.2 Administración del mantenimiento.

Uno de los principales objetivos de la administración de sistemas de información es desarrollar un sistema que presente un alto grado de mantenibilidad. A continuación se indican algunas de las causas por las cuales es preciso cambiar los programas, archivos, documentación y procedimientos generales de un sistema de operación. Con ello se da solución o se mejora.

6.3 Factores que afectan a los costos de mantenimiento.

Los costos de mantenimiento se pueden minimizar si se tienen en cuenta los requisitos del encargado de dar mantenimiento al programa en el momento en que éste se diseña y desarrolla. Estimar tales costos de mantenimiento de un programa determinado, es muy difícil. Las dificultades se deben a que los costos están relacionados con varios factores relativamente impredecibles que no tienen que ver con ninguna característica técnica del sistema de software. Estos incluyen:

- **La Aplicación que se apoya.** Si la aplicación del programa está definida con claridad y se comprende bien, los requisitos del sistema pueden ser definitivos y el mantenimiento perfecto debido a cambios en ellos, mínimo. Si, por otro lado, la aplicación es nueva, es probable que los requisitos iniciales se modifiquen a medida que los usuarios adquieren experiencia con el sistema.
- **Estabilidad del personal.** Suele ser más fácil para el escritor original comprender y modificar el programa que para cualquier otra persona que tenga que comprenderlo estudiando su documentación y el listado del código. Así pues, si el programador de un sistema también le da mantenimiento, los costos por tal concepto se reducirán. En la práctica, la naturaleza de la profesión de programador es tal que los individuos cambian de trabajo con frecuencia, y es poco usual que una persona desarrolle y dé mantenimiento a un programa durante su vida útil.
- **El tiempo de vida del programa.** Es obvio que la vida útil de un programa depende de su aplicación. El programa quedará anticuado al mismo tiempo que la aplicación o si se cambia su hardware original y los costos de conversión superan a los de reescritura. La evidencia histórica indica que los tiempos de vida de los programas son mucho mayores de lo que se espera; algunos sistemas todavía operativos fueron diseñados a principios de la década de 1960.

Como a un programa se le da mantenimiento de manera continua durante su vida útil, los costos de este aspecto tienden a crecer. Esto es resultado del deterioro inevitable de la estructura del programa, debido a múltiples modificaciones que se hacen.

- **La dependencia del programa de su ambiente externo.** Si un programa depende en gran medida de su ambiente externo, se debe modificar con los cambios de ambiente.
- **Estabilidad del hardware.** Si un programa se diseña para operar en una configuración particular de hardware y ésta no cambia durante el tiempo de vida del programa, no se incurrirá en ningún costo de mantenimiento debido a cambios de hardware. Sin embargo, los desarrollos en el hardware son tan rápidos que esta situación es relativamente rara y el hardware queda anticuado con mucha rapidez. El programa se debe modificar para utilizar el nuevo hardware que reemplace al equipo obsoleto.

Además de tales consideraciones, los costos de mantenimiento también están regidos por factores técnicos menos predecibles. Algunos factores técnicos que afectan al mantenimiento de los programas son:

- **La interdependencia de los módulos.** Debe ser posible modificar una unidad de programa de un sistema sin afectar a ninguna otra unidad.
- **El lenguaje de programación.** Los programas escritos en un lenguaje de alto nivel por lo general son más fáciles de comprender (y, por lo tanto, de mantener) que los programas escritos en lenguaje de bajo nivel.
- **El estilo de programación.** La manera en que está escrito un programa contribuye claramente a su comprensión y, por tanto, a la facilidad con que se puede modificar.
- **La comprobación y prueba del programa.** En general, cuanto más tiempo y esfuerzo se dedique a la comprobación del diseño y la prueba del programa, menos errores tendrá éste y, en consecuencia, los costos de mantenimiento debidos a la corrección de errores serán menores. Los costos de mantenimiento originados por la corrección de errores están influidos por el tipo de error que se va a corregir. La corrección de los errores de codificación por lo general es relativamente barata, mientras que los errores de diseño son mucho más caros porque pueden implicar la reescritura de una o más unidades del programa. Los errores en la especificación de requisitos suelen ser los más caros de corregir por el drástico rediseño que casi siempre implican.
- **La calidad y cantidad de la documentación del programa.** Si un programa está apoyado por documentación clara, completa y concisa, la tarea de comprenderlo puede ser de relativa sencillez. En consecuencia, los costos de mantenimiento del programa tienden a ser menores en sistemas bien documentados que en aquellos con documentación pobre o incompleta.

6.4 Problemas de mantenimiento.

A nadie podrá sorprender el hecho de que los sistemas estén sujetos a un cambio casi constante. Por desdicha las causas del mantenimiento pueden causar algunos problemas, que son los siguientes:

- **Costo.-** El cambio de los sistemas de aplicación puede ser muy costoso y en algunos casos absorber la mitad del costo de los sistemas de la organización. Muchas empresas han adoptado el principio de contar con un presupuesto para las actividades de mantenimiento, ejecutando tareas de mayor prioridad que se puedan realizar sin salirse de dicho presupuesto. Esto se basa en la noción de que los costos del software y de mantenimiento dependen del tamaño de los sistemas y de otros factores que incluyen la confiabilidad requerida, el tipo de software, las restricciones del hardware y la habilidad del personal de desarrollo del software.
- **La Moral del Personal Informático.-** El personal que labora en el sistema de información, sobre todo los programadores, protestan a menudo por la cantidad de mantenimiento que se les pide realizar. No les agrada pasar la mitad de su tiempo manteniendo o tratando de remendar sistemas que se instalaron mucho tiempo atrás.
- **Fallas.-** La programación de mantenimiento tiene fama de causar fallas catastróficas que la programación original de desarrollo. Si el programador de mantenimiento no está familiarizado con el programa o si la documentación es insuficiente, es posible que algunos cambios ocasionen fallas muy severas. En el primer caso, es obvio que la administración asignará los problemas mas complejos de mantenimiento a las personas más capacitadas, en el segundo, corresponde a la administración cuidar de que los programas estén debidamente documentados, ya que la documentación constituye en realidad la información necesaria para modificar un programa.
- **Costos Adicionales de Capacitación.-** Para mantener los viejos sistemas de aplicación usando lenguajes de programación y equipos anticuados, se requiere un entrenamiento adicional de las personas que lleguen a últimas fechas a trabajar con tales sistemas.
- **Circunstancias Impredecibles.-** La administración puede verse ante múltiples problemas complejos y difíciles de manejar, estas circunstancias provienen de factores como los siguientes:
 - Programas que difieren en diseño.
 - Escasa documentación del sistema.
 - Variedad de equipos y programas (incompatibilidad).
 - El uso de equipo y procedimientos anticuados.

La fase de mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas por la evolución del entorno del software y a las

modificaciones debidas a los cambios de los requisitos del cliente dirigidos a reforzar o ampliar el sistema. La fase de mantenimiento vuelve a aplicar los pasos de la fase de definición de desarrollo, pero en contexto del software ya existente.

6.5. Características del mantenimiento

El mantenimiento de software ha sido hasta hace muy poco la fase negra del proceso de ingeniería del software. Las características del mantenimiento de software se consideran desde tres puntos de vista diferentes:

- Las actividades requeridas para cubrir la fase de mantenimiento y el impacto de un enfoque de ingeniería de software (o de su ausencia) sobre la eficacia de tales actividades.
- Los costos asociados con la fase del mantenimiento.
- Los problemas que se encuentran frecuentemente cuando se lleva a cabo el mantenimiento.

6.6. Mantenimiento estructurado frente al no estructurado.

Si sólo se dispone del código fuente como elemento de configuración, la actividad de mantenimiento comienza con una dolorosa evaluación del código, la cual es más complicada por la pobre documentación interna. Las características delicadas, tales como: la estructura del programa, las estructuras de datos globales, las interfaces del sistema, el rendimiento y/o las limitaciones del diseño, son difíciles de descubrir y, frecuentemente, mal interpretadas, lo que se hace es un mantenimiento no estructurado, ya que es difícil asegurar cuales son las ramificaciones de los cambios realizados finalmente sobre el código.

Si existe una configuración completa del software, la tarea de mantenimiento comienza con una evaluación de la documentación del diseño. Se determinan las características estructurales importantes de rendimiento y de interfaz de software. Se estudia el impacto de las correcciones o modificaciones requeridas y se traza un plan de actuación. Se modifica el diseño y se revisa, se desarrolla nuevo código fuente, se realizan pruebas de regresión mediante la información contenida en la especificación de procesos y se vuelve a lanzar el software. Esta secuencia de sucesos constituye el mantenimiento estructurado y aparece como resultado de una anterior aplicación de una metodología de ingeniería de software.

La facilidad de mantenimiento se puede definir cualitativamente como la facilidad de comprender, corregir, adaptar y/o mejorar el software, y se ve afectada negativamente por muchos factores que a continuación se listan:

- Falta de cuidado en el diseño.
- Falta de cuidado en la codificación.

- Una fase de prueba muy pobre.
- Una fase de prueba muy limitada.

Todo esto, tiene un impacto obviamente negativo sobre nuestra capacidad de mantener fácilmente el software. Además de estos factores se definen otros relacionados con el entorno del desarrollo:

- Disponibilidad de una plantilla de software calificada.
- Estructura del sistema comprensible.
- Facilidad de manejo del sistema.
- Uso de lenguajes de programación estandarizados.
- Uso de sistemas operativos estandarizados.
- Estructura de la documentación estandarizada.
- Disponibilidad de los casos de prueba.
- Facilidades de depuración incorporadas.
- Disponibilidad de una computadora apropiada para llevar a cabo el mantenimiento.

Para cuantificar la facilidad del mantenimiento, se tiene un número de métricas para medir el esfuerzo gastado durante el mantenimiento:

- Tiempo de reconocimiento del problema.
- Tiempo de retraso administrativo.
- Tiempo de recolección de herramientas de mantenimiento.
- Tiempo de análisis del problema.
- Tiempo de especificación de cambios.
- Tiempo activo de corrección o modificación.
- Tiempo de prueba local.
- Tiempo de prueba global.
- Tiempo de revisión del mantenimiento.
- Tiempo total de recuperación.

Además de esas medidas orientadas al tiempo, se puede medir indirectamente la facilidad de mantenimiento considerando medidas de la estructura del diseño.

Debido a que la facilidad del mantenimiento debe ser una característica esencial de cualquier software, se deben considerar los factores antes señalados en la etapa del desarrollo. En cada nivel del proceso de revisión de la ingeniería del software ha de ser considerada la facilidad del mantenimiento.

Durante las revisiones del diseño deben evaluarse el diseño de la base de datos y los procedimientos. En las revisiones del código se debe cuidar el estilo y la documentación interna, dos factores que tienen influencia sobre la facilidad de mantenimiento.

6.7. Efectos secundarios del mantenimiento

Cuando se usa el concepto del mantenimiento de software, el término "EFECTOS SECUNDARIOS" implica un error u otro comportamiento indeseable aparecido como resultado de una modificación. Se definen tres grandes categorías de efectos secundarios, que se tratan a continuación.

- **Efectos Secundarios sobre el Código.** Un cambio sencillo sobre una sola sentencia puede, a veces, tener resultados desastrosos. Las posibilidades de efectos secundarios tiene mayor probabilidad de inducir errores con los siguientes cambios:
 - Un subprograma eliminado o cambiado.
 - Eliminación o modificación de una sentencia de etiqueta.
 - Eliminación o modificación de un identificador.
 - Cambios para mejorar el rendimiento de ejecución.
 - Modificación de apertura o cierre de archivos.
 - Modificación de operadores lógicos.
 - Traducción de cambios de diseño en importantes cambios en el código.
 - Cambios sobre las pruebas de límites.

Los efectos secundarios sobre el código van desde los insulsos errores que se detectan y se remedian durante las pruebas de regresión, hasta los problemas que pueden hacer que falle la operación del software.

- **Efectos Secundarios Sobre los Datos.** Durante el mantenimiento a menudo se hacen cambios sobre elementos de una estructura de datos. Cuando cambian los datos, el diseño del software puede no cuadrar con los datos y aparecen errores, también aparecen como resultado de las modificaciones sobre la estructura de información del software. Los cambios de datos a menudo producen efectos secundarios como los siguientes:
 - Redefinición de constantes locales y globales.
 - Redefinición de formatos de registros o de archivos.
 - Aumento o disminución del tamaño de arreglos o de otras estructuras de datos de mayor orden.
 - Modificación de datos globales.
 - Reinicialización de indicadores de control o de punteros.
 - Reorganización de argumentos de E/S de subprogramas.

Los efectos secundarios sobre los datos se pueden limitar mediante una profunda documentación de diseño que describa las estructuras de datos y de una referencia cruzada que asocie los elementos de datos, los registros, los archivos y otras estructuras a los módulos de software.

- **Efectos Secundarios sobre la Documentación.** El mantenimiento se debe centrar en la configuración completa del software y no solo en las modificaciones del código fuente. Los efectos secundarios sobre la documentación se dan cuando no se reflejan los cambios del código fuente en la documentación de diseño y en los manuales orientados al usuario.

CONCLUSIONES

El desarrollo del (SIITES) ha cumplido con sus objetivos: registrar los ingresos, egresos, trasposes e inversiones oportunamente de los movimientos generados en el área de Tesorería de Ferrocarriles Nacionales de México. Con el desarrollo de la presente tesis se comprueba el cumplimiento de objetivos de este sistema en el marco del Proyecto Institucional de Desarrollo Informático Administrativo (PIDIA), utilizando la tecnología de redes de computadoras (FERROPAC) como infraestructura para el desarrollo del sistema, construyendo una base de datos distribuida, programada con el lenguaje de cuarta generación, utilizando metodologías de análisis y diseño estructurado.

Es importante reiterar que el sistema (SIITES), es de mucha importancia para la Gerencia de Tesorería y ha tenido una excelente aceptación por los usuarios, debido a que es de fácil manejo, ha mejorado considerablemente los tiempos en el desempeño de las actividades manuales y tediosas, en fin todas sus ventajas han implicado obtener información confiable y oportuna. Esto significa que se han cumplido con los objetivos planteados al inicio de esta tesis.

Como se ha observado, el (SIITES), como sistema distribuido ha funcionado correctamente con ayuda del sistema de comunicaciones confiable con que se cuenta (red FERROPAC), porque no hay extravío de información cuando se procesa de punto a punto, ni se duplica o altera.

La realización del presente sistema surge como una necesidad para que se complementen todas las funciones operativas y técnicas que desempeña la Gerencia de Tesorería de Ferrocarriles Nacionales de México, este sistema abarca las áreas básicas para el adecuado control y administración de la información de esta Gerencia.

El sistema (SIITES) desarrollado presenta un diseño abierto, para el punto de vista del acoplamiento de sistemas, lo cual es la incorporación de las nuevas áreas operativas de F.N.M.

El (SIITES) representa en su operación, facilidad de trabajo a bajo costo financiero y poco esfuerzo humano.

El (SIITES) muestra características como la de formar una estandarización para el desarrollo de sistemas de información, así como todos los formatos y gráficas de interfaz con el usuario. Otra característica es la de proporcionar información única de los catálogos perfectamente validados y delimitados para cada área.

Por todo esto, podemos concluir que, a partir del planteamiento del problema inicial, la solución implantada resulta ser innovadora debido a que en la Gerencia de Tesorería no tenían un sistema que les proporcionara información veraz, confiable y oportunamente.

CONCLUSIONES

También, hemos podido observar que el desarrollo del Sistema Integral de Información para Tesorería, utilizando una herramienta de cuarta generación, hace posible que su implantación sea realizada en un tiempo menor. Esto confirma que el rediseño de los sistemas ya existentes con herramientas de este tipo son de mayor eficiencia que cualquier otra herramienta diseñada para el desarrollo de software.

De acuerdo a las ventajas planteadas para el (SITES), actualmente con el presente trabajo se obtuvo lo siguiente:

- Se agilizó el manejo de todo tipo de información.
- Quedó eliminado el alto volumen de papeles utilizados anteriormente.
- La forma de organización de la información facilita la generación de informes, tales como la carátula de la póliza de caja, así como los boletines que respaldan la información de la misma; nos permite generar informes estadísticos que pueden ser acumulados en periodos semanales, quincenales o mensuales los cuales nos dan una visión clara, veraz y oportuna de los ingresos y egresos de la empresa
- Facilita las auditorías que pueden ser de dos tipos:
 - Administrativas.- Las que le practican a la Gerencia de Tesorería (internas o externas)
 - Informativas.- La que practica el sistema mismo para deslindar responsabilidades.

Por lo tanto, se han cumplido con los objetivos propuestos al principio de este proyecto de tesis con base en los requerimientos de la Gerencia de Tesorería.

BIBLIOGRAFÍA**MANUALES DEL MANEJADOR INFORMIX**

- A Beginners Guide to HP-UX Reference Volume I
- Command Reference
- DB-Library
- HP-UX Reference Volume I
- Manual Técnico de la Red de Datos FERROPAC
- Manual 4GL-Reference
- SQL Guide
- System Administration Guide.
- Using Triggers

LIBROS

Análisis Estructurado Moderno
Edward Yourdon
Prentice Hall Hispanoamericana S.A.

Aplicación SQL
James R. Groff, Paul N. Weynberg
Mc Grall Hill, 1991

Database Management
Ozkarahan. Esen
Iberoamericana, 1991

Fundamentos de Bases de Datos
Henry F Korth
Mc Graw Hill 1994

Ingeniería de Software
Harvey M. Deitel
Adisson-Wesley Iberoamericana, 1987.

Ingeniería de Software
Pressman
Prentice-Hall, 1988.

Introducción a los Sistemas de Bases de Datos
C.J. Date
Adisson-Wesley Iberoamericana, 1986.

BIBLIOGRAFÍA

Principal of Relation Database Systems
Sitansu S. Mitra
Prentice Hall, 1991.

Relation Database Design
Paul Winsberg
Winsberg, 1990

Procesamiento de Datos en Unix
R. S. Tare
Mc Graw-Hill 1990

Técnicas de Bases de Datos
Estructuración en diseño y administración
Shakuntala Atre
Trillas, 1988

GLOSARIO DE TÉRMINOS

AJUSTE.- Es el registro de un evento originado por la cancelación de un documento que tuvo afectación contable y por ello es necesario recuperar ese egreso.

CDI.- Coordinación de Descentralización Informática.

DBA.- Administrador de Bases de Datos

DBMS.- Sistema Manejador de Bases de Datos.

DEVENGADO.- Documento con compromiso de pago y con clave presupuestal.

DFD.- Diagrama de Flujo de Datos.

DFO.- Diagrama de Flujo Operativo.

FERROCARRIL.- Empresa ferrocarrilera producto de la reestructuración de la empresa Ferrocarriles Nacionales de México para enfrentar el compromiso de la privatización.

FERROPAC.- Red Integral de Información Computarizada, red para soportar el intercambio de información entre las diferentes entidades de FNM.

FNM.- Ferrocarriles Nacionales de México.

HOSTS.- Son máquinas destinadas a correr programas de aplicación, también reciben el nombre de sistemas terminales.

PIDIA.- Proyecto Institucional de Desarrollo Informático Administrativo planteado por la Coordinación de Descentralización Informática a la Subdirección General de Finanzas.

PÓLIZA DE CAJA.- Documento de carácter contable que consta de dos partes, la primera es una carátula que resume los movimientos de las diferentes cuentas contables que utiliza el organismo, y la segunda que son boletines en donde se detalla la carátula mencionada.

SERVIDOR.- Es la computadora central que permite compartir recursos y es donde se encuentra alojado el sistema operativo. El servidor será el corazón de una red. Ya que provee el acceso controlado a los archivos, permite compartir impresoras y otros recursos dentro de la red.

GLOSARIO DE TÉRMINOS

Esta máquina debe tener la suficiente capacidad de procesamiento para llevar a cabo las tareas de la red y contar con suficientes ranuras de expansión (tarjetas de expansión, tarjetas de interfaz, etc.). El disco duro utilizado en el servidor debe ser soportado por el sistema operativo de la red, ser del tamaño necesario para que nuestra información pueda guardarse en él y además debe contemplar un posible crecimiento.

SERVIDOR DE LA BASE DE DATOS.- Es una porción del sistema administrador de bases de datos, que se encarga de manipular los archivos de la misma. De las tareas que se encarga de realizar un servidor de bases de datos al momento de recibir una instrucción por parte de una aplicación se encuentran las siguientes:

- Recibe las instrucciones del SQL por parte de una aplicación (programa).
- Optimiza el acceso a los datos.
- Recupera información de la base de datos; y
- Regresa los datos a la aplicación

SIID.- Sistema Integral de información Distribuida conformado por 15 macrosistemas.

SIIPAD.- Sistema Integral de Información de Pago Descentralizado.

SIIPTAL.- Sistema Integral de Información Presupuestal.

SISTEMA OPERATIVO.- Un sistema operativo es un programa que se encarga de aprovechar de manera óptima los recursos de una computadora. Estos recursos son, por ejemplo, el hardware (los dispositivos periféricos, dispositivos de entrada y salida, memoria, etc.).

SISTEMA OPERATIVO DE RED.- Es el software que se encarga de administrar los recursos que se están compartiendo (discos duros, impresoras, etc.) entre los usuarios.

El sistema operativo se selecciona según las necesidades de control de la información. Existen algunas consideraciones como son: el tipo de información que se estará compartiendo, los programas que se utilizarán, quién tendrá acceso a cierta información, etc. El sistema operativo que se haya seleccionado debe dar toda la seguridad que se requiere dentro de la red. Esta debe ir desde qué máquina se pueda usar, a qué hora se puede entrar a la red y que días se puede trabajar, hasta qué clave de acceso tendremos, los archivos que se podrán compartir o los programas que se ejecutaran.

TRASPASO.- Se refiere a la acción de mover capital de una cuenta bancaria a otra.

TRANSFERENCIA. Es una forma de pago que se realiza al depositar los pagos de los proveedores en las cuentas de éstos.



APÉNDICE A

**FERROCARRILES NACIONALES DE MÉXICO
SUBDIRECCIÓN GENERAL DE FINANZAS
COORDINACIÓN DE DESCENTRALIZACIÓN INFORMÁTICA**



FEBRERO 1997

CONTENIDO

	Pág.
I. Introducción	A3
II. SIITES	A4
II.1 Las características del SIITES son:	A4
II.2 Los beneficios que ofrece el SIITES son:	A4
II.3 ¿Como entrar al SIITES?	A5
III. Pantallas de menús	A7
IV. Pantallas para registro y actualización de datos	A8
V. Pantallas de apoyo para registro de datos	A17
VI. Procedimientos de operación	A19
VI.1 Administración	A19
VI.2 Ingresos	A24
VI.3 Egresos	A26
VI.4 Inversiones	A37
VI.5 Catálogos	A42
VI.6 Póliza de caja	A51

I. INTRODUCCIÓN

La captación de recursos monetarios así como la mejor aplicación financiera que de ellos se haga, con ayuda de los actuales instrumentos de inversión, resulta ser uno de los pilares para una economía sana en las empresas.

No menos importante es el cumplimiento de los compromisos de pago con entidades externas, así como la negociación de créditos con las grandes firmas y, por supuesto, la solvencia para liquidar los requerimientos de pago del personal.

Es por ello que se hace patente la conveniencia y necesidad de contar con un sistema eficiente, sencillo y ágil que controle tanto la disponibilidad de efectivo como los compromisos de pago de la empresa.

En consecuencia la Gerencia de Tesorería, como el área administrativa responsable de tales funciones y en el marco actual de modernización que requiere el organismo, ha solicitado la colaboración de la Coordinación de Descentralización Informática para desarrollar un Sistema Integral de Información que controle el flujo de efectivo y los movimientos financieros que se efectúan.

En apoyo a este requerimiento la Coordinación de Descentralización Informática ha tomado el compromiso de realizar este Sistema Integral de Información de Tesorería (SIITES), el cual dará respuesta a las necesidades de la Gerencia de Tesorería dentro de un ambiente de desarrollo informático institucional, buscando que las áreas constitutivas de la empresa puedan explotar la información generada, transformándola en forma más eficiente para la toma de decisiones.

El presente documento tiene como objetivo brindar al usuario del SIITES, una guía para el manejo apropiado de las opciones que se tienen en él.

En primer término, el índice y contenido de este documento se marcan los principales procedimientos que se deben realizar para obtener resultados y cubrir así las funciones administrativas que se relacionan con el registro de documentos de pago o ingreso y control bancario en la Gerencia de Tesorería.

II. SIITES***II.1 LAS CARACTERÍSTICAS DEL SIITES SON:***

- AMIGABLE.-** Cada una de las funciones o tareas que realiza el sistema se seleccionan mediante opciones de menús. Además, en la captura de datos presenta mensajes de ayuda.
- EN LINEA.-** Cada vez que se realiza una operación de manera inmediata y por consecuencia se actualiza la información.
- MODULAR.-** Se le pueden agregar, suprimir o modificar funciones sin alterar las demás ya existentes.
- CONSISTENTE.-** Todas sus funciones se ejecutan de la misma forma provocando resultados semejantes.
- SEGURIDAD.-** El acceso al sistema es mediante una cuenta y clave secreta para cada usuario.

II.2 LOS BENEFICIOS QUE OFRECE EL SIITES SON:***PARA OFICINAS ADMINISTRATIVAS:***

Apoyo al proceso de toma de decisiones y la facilidad para elaborar información estadística y evaluaciones, así como la formulación de análisis e informes relativos al manejo de los mismos movimientos del organismo.

Reducción de tiempos y costos para el registro, almacenamiento y control de documentos, así como para el procesamiento de los datos contenidos en ellos.

Reducción de los tiempos para producción de información de carácter estadístico.

Disponibilidad de la herramienta de consulta de datos y elaboración de informes en forma versátil y al alcance de cada usuario autorizado para su uso.

11.3 ¿COMO ENTRAR AL SIITES?

Para operar el SIITES previamente, el Administrador de Recursos de Cómputo debe de haber creado una cuenta o clave de usuario, de UNIX, que permita el uso de las terminales del equipo de cómputo. Asimismo, para ejecutar el programa principal del sistema, también previamente, el Administrador del Sistema debe haber registrado el nombre de usuario y password para cada uno de los empleados que utilizarán el SIITES. Es decir, hay tres niveles de seguridad, uno a nivel de sistema operativo de la computadora y dos a nivel del sistema SIITES este último caso posibilita otorgar permisos para el sistema y para las opciones de operación del sistema. En los siguientes apartados, se indica el procedimiento para entrar al SIITES.

1. Entrando a la computadora (1er. nivel de seguridad).

1.1. Encender la terminal oprimiendo el interruptor que se encuentra en la parte frontal de la misma.

1.2. Obtener en la pantalla de la terminal la palabra "login:" oprimiendo la tecla "Return" o "Intro".

1.3. Teclear la cuenta de UNIX, por ejemplo:

login: SIITES

2. Entrando al SIITES (2º nivel de seguridad).

2.1 Una vez realizados los pasos anteriores, el sistema despliega la pantalla de presentación del SIITES como se muestra en la Figura No. 11.1.

2.2 Teclear el password del usuario como se ilustra en la Figura No. 11.2 (aparecerán asteriscos en lugar de la clave, a manera de seguridad), Después teclear una vez el "Return" o "Intro"

3. Dentro del SIITES (3er nivel de seguridad).

3.1 Dependiendo de los privilegios asignados al password, el usuario tendrá acceso a determinados módulos del sistema, y a su vez, dentro de éstos acceso a determinadas funciones o actividades.

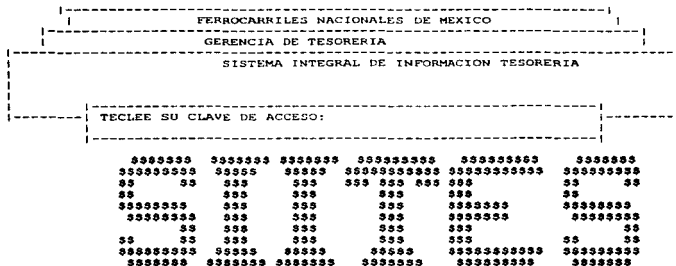


Figura No. II.1

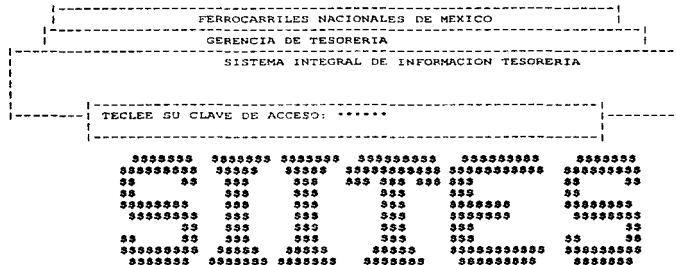


Figura No. II.2

III. PANTALLAS DE MENUS

Las pantallas de menús sirven para navegar en el SIITES utilizando sus opciones.

La pantalla del menú principal del SIITES se presenta en la Figura No. III.1.

1996/07/05	FERROCARRILES NACIONALES DE MEXICO	11:28		
MENU PRINCIPAL				
1. ADMINISTRACION 2. INGRESOS 3. EGRESOS 4. INVERSIONES 5. CATALOGOS 6. REPORTES				
MENU DE ADMINISTRACION DEL SIITES				
CTRL. PANTALLA	F MENU	F MENU	F TERMINAR	F ENTRADA
W DE AYUDA	5 ANTERIOR	6 PRINCIPAL	7 SESION	8 RAPIDA

Figura No. III.1

En el encabezado, el primer renglón en la parte central se refiere a FERROCARRILES NACIONALES DE MÉXICO propietario del sistema "SIITES", al margen izquierdo se indica la fecha del sistema, al margen derecho la hora del mismo. En los renglones subsecuentes se indica el nombre del menú, y sus opciones con un mensaje de ayuda para cada opción. Y en la parte inferior funciones que complementan el sistema.

El funcionamiento de las pantallas de menú es igual en todas ellas, para hacer efectiva una opción simplemente se teclaea el número de la opción deseada, o bien, navegando con las teclas del cursor (flechas) hasta la opción deseada y luego teclrear "Retum" o "Intro".

Si por alguna razón se teclaea una letra distinta a las del menú, el sistema no responde y espera hasta un intento correcto.

IV. PANTALLAS PARA REGISTRO Y ACTUALIZACIÓN DE DATOS

Las pantallas de datos, también conocidas como pantallas de trabajo, sirven para registrar, consultar y actualizar información en el SITÉS.

A continuación se presenta una pantalla de datos, la correspondiente al Catálogo de Regiones.

Todas las pantallas de datos, como la que se ilustra en la Figura No. IV.1 presentan en la parte superior un menú horizontal mediante el cual se pueden elegir acciones.

```
MENU: ALTA BAJAS MODIFICACION CONSULTA SIGUIENTE PREVIO REPORTE FIN
Alta de un nuevo registro
```

[05/11/96]	FERROCARRILES NACIONALES DE MEXICO	[18:57]
	S I T E S	
	CATALOGO DE REGIONES	
REGION :		
DESCRIPCION :		

Figura No. IV.1

En todas las pantallas de trabajo, para seleccionar cualquiera de las opciones se procede de dos formas:

- Tecleando la primera letra mayúscula de la opción.
- Navegando con las teclas del cursor (flechas) hasta la opción deseada y luego teclear "Return" o "Intro".

Acción Alta: Sirve únicamente para registrar datos por primera vez, es decir, mediante esta acción se crean e insertan registros a los archivos del SITÉS.

Al elegir la opción "Alta" el cursor (cuadro parpadeando) se mueve al área de datos colocándose en el primer campo de la pantalla para registro de datos. Para cambiar a otro campo de la pantalla se utiliza la tecla "Return" o "Intro" y se proporciona el dato; y así sucesivamente con los demás datos si los hubiera. Al término de esto se tienen dos alternativas.

Primera. La tecla "Esc" sirve para confirmar y ejecutar la acción de registro (Alta) de los datos en el SIITES. Una vez realizado lo anterior, el sistema reafirma con el mensaje que se ilustra en la figura No. IV.2.

DIGITE <ESC> PARA ACEPTAR REGISTRO
<CTRL-C> PARA ABORTAR OPERACION

[05/11/96]	FERRCARRILES NACIONALES DE MEXICO S I I T E S CATALOGO DE REGIONES	[18:57]
REGION :	1	
DESCRIPCION :	CORPORATIVO	

ALTA EFECTUADA...

Figura No IV.2

Segunda: Con la combinación de las teclas "Control-C" se puede cancelar la acción de registro (Altas) de datos en el SIITES. Una vez realizado lo anterior, el sistema reafirma con el mensaje que se ilustra en la figura No. IV.3.

DIGITE <ESC> PARA ACEPTAR REGISTRO
<CTRL-C> PARA ABORTAR OPERACION

[05/11/96]	FERRCARRILES NACIONALES DE MEXICO S I I T E S CATALOGO DE REGIONES	[19:26]
REGION :		
DESCRIPCION :		

ALTA ABANDONADA, ULTIMOS DATOS IGNORADOS

Figura No IV.3

Acción Consulta: Esta opción sirve para buscar o encontrar datos previamente registrados en el SIITES, Figura No. IV.4.

MENU: ALTA BAJAS MODIFICACION **CONSULTA** SIGUIENTE PREVIO REPORTE FIN
Alta de un nuevo registro

[05/11/96]	FERROCARRILES NACIONALES DE MEXICO	(18:57)
	SIITES	
	CATALOGO DE REGIONES	
REGION	: 2	
DESCRIPCION	: NORESTE	

REGISTRO 2 DE 6

Figura No IV.4

Al elegir la opción "Consulta" el cursor se mueve del área del menú de acciones, al área de datos concretamente al primer campo de la pantalla. Cada uno de los datos que aparecen en la pantalla se pueden utilizar para formular la condición o criterio selección para que el sistema encuentre registros.

En este caso se utilizó como ejemplo la pantalla de trabajo para registro de REGIONES y como se observa en la Figura No. IV.4 el mensaje que despliega el sistema en la parte inferior derecha de la pantalla significa que, de acuerdo con los datos proporcionados, encontró 2 de 6 registros que reúnen características similares a las que se proporcionaron de acuerdo al criterio de búsqueda, y que se encuentran desplegados los datos del registro número 2.

Acción Siguiente: Esta opción sirve para avanzar registro por registro del total de los encontrados por el sistema. En la pantalla que se ilustra en la Figura No. IV.5, el mensaje desplegado en la parte inferior derecha de la misma indica que se están visualizando los datos del registro número 3.

MENU: ALTA BAJAS MODIFICACION CONSULTA SIGUIENTE PREVIO REPORTE FIN
Alta de un nuevo registro

```
[05/11/96]          FERROCARRILES NACIONALES DE MEXICO          [18:57]
                    S I I T E S
                    CATALOGO DE REGIONES

REGION              :      3
DESCRIPCION         :      PACIFICO
```

REGISTRO 3 DE 6

Figura No IV.5

Para avanzar de registro en registro teclear la letra "S" o "Return" si ya esta seleccionada la opción.

Acción Previo: Esta opción sirve para retroceder registro por registro del total de los encontrados por el sistema. En la pantalla que se ilustra en la Figura No. IV.6, el mensaje desplegado en la parte inferior derecha de la misma indica que se está visualizando el registro número 1.

MENU: ALTA BAJAS MODIFICACION CONSULTA SIGUIENTE PREVIO REPORTE FIN
Alta de un nuevo registro

```
[05/11/96]          FERROCARRILES NACIONALES DE MEXICO          [18:57]
                    S I I T E S
                    CATALOGO DE REGIONES

REGION              :      1
DESCRIPCION         :      BUENAVISTA
```

REGISTRO 1 DE 6

Figura No IV.6

Para retroceder de registro en registro teclear la letra "P" o "Return" si ya está seleccionada la opción.

Acción Modificación: Al elegir la opción "Modificación" el cursor se mueve al área de datos colocándose en el primer dato de la pantalla; se elige el campo deseado, se modifica el dato y se ejecuta o cancela el cambio, de forma similar a las altas, se tienen dos alternativas.

Primera: Con la tecla "Esc" se ejecuta la acción de actualizar (Modificación) los datos en el SIITES. Una vez realizado lo anterior, el sistema confirma con el mensaje que se ilustra en la Figura No. IV.7.

MENU: ALTA BAJAS MODIFICACION CONSULTA SIGUIENTE PREVIO RESCATE FIN
Alta de un nuevo registro

[05/11/96]	FERROCARRILES NACIONALES DE MEXICO	[18:57]
	SIITES	
	CATALOGO DE REGIONES	
REGION	: 2	
DESCRIPCION	: MONTERREY	

MODIFICACION EFECTUADA

REGISTRO 2 DE 6

Figura No. IV.7

Segunda: Con la combinación de las teclas "Control-C" se cancela la acción de actualizar (Modificación) datos en el SIITES. Una vez realizado lo anterior, el sistema confirma con el mensaje que se ilustra en la Figura No. IV.8.

MENU: ALTA BAJAS MODIFICACION CONSULTA SIGUIENTE PREVIO REPORTE FIN
 Alta de un nuevo registro

[05/11/96]	FERROCARRILES NACIONALES DE MEXICO S I I T E S CATALOGO DE REGIONES	[18:57]
REGION	: 2	
DESCRIPCION	: NORESTE	

MODIFICACION ABANDONADA, ULTIMOS DATOS IGNORADOS

REGISTRO 2 DE 6

Figura No IV.8

Acción Bajas: Al elegir la opción "Bajas" el sistema despliega en la parte central de la pantalla un mensaje como se ilustra en la Figura No. IV.9

MENU: ALTA BAJAS MODIFICACION CONSULTA SIGUIENTE PREVIO REPORTE FIN
 Alta de un nuevo registro

[05/11/96]	FERROCARRILES NACIONALES DE MEXICO S I I T E S CATALOGO DE REGIONES	[18:57]
REGION		
DESCRIPCION	DESEA DAR DE BAJA ESTE REGISTRO	
	No	SI

REGISTRO 2 DE 6

Figura No. IV.9

Si se tecldea la letra "S" el sistema elimina el registro. Si se tecldea la letra "N" el sistema no elimina el registro.

En el caso de teclear "S", el sistema confirma la acción de borrar el registro, desplegando en la parte inferior izquierda de la pantalla, el mensaje que se ilustra en la Figura No. IV.10.

MENU: ALTA BAJAS MODIFICACION CONSULTA SIGUIENTE PREVIO REPORTE FIN
Alta de un nuevo registro

[05/11/96] FERROCARRILES NACIONALES DE MEXICO [18:57]
S I T E S
CATALOGO DE REGIONES

REGION : 2
DESCRIPCION : NORESTE

REGISTRO ELIMINADO

REGISTRO 2 DE 6

Figura No. IV.10

Acción Reporte: Al elegir la opción "Reporte" el sistema despliega en la parte central de la pantalla un menú como el que se ilustra en la Figura No. IV.11

MENÚ: ALTA BAJAS MODIFICACION CONSULTA SIGUIENTE PREVIO **REPORTE** FIN
 Genera reporte

[05/11/96]	FERROCARRILES NACIONALES DE MEXICO S I I T E S CATALOGO DE REGIONES	[18:57]	
REGION	: 2		
<table border="1"> <tr> <td>SALIDA DEL REPORTE: Archivo Impresora Pantalla Fin Se genera el archivo</td> </tr> </table>			SALIDA DEL REPORTE: Archivo Impresora Pantalla Fin Se genera el archivo
SALIDA DEL REPORTE: Archivo Impresora Pantalla Fin Se genera el archivo			

REGISTRO 2 DE 6

Figura No. IV.11

Como se observa se pueden seleccionar cualquiera de las opciones del menú de impresión, los cuales pueden ser :

Pantalla: Sirve para desplegar en la pantalla de la terminal los registros previamente encontrados con la opción "Consulta".

Impresora: Sirve para imprimir los resultados de las consultas en la impresora que el usuario desee, para esto se despliega un submenú como el de la figura No. IV.12. La explicación de cada opción es la siguiente:

Impresora por Default: En esta opción el usuario tan solo necesita indicar que la impresora esta lista y el impreso saldrá en la impresora principal.
 Por ejemplo:

Esta lista la impresora? <S> Si o <N> No : S

Direccionar a Puerto: En esta opción el usuario necesita indicar le puerto en donde se encuentra conectada la impresora, después debe indicar si la impresora esta lista, y el impreso saldrá en la impresora del puerto preestablecido; Esta es una impresora no definida.
 Por ejemplo:

Proporcione la dirección de la impresora : tty1p9

Esta lista la impresora? <S> Si o <N> No : S

Direccionar a Impresora: En esta opción el usuario necesita indicar el nombre de la impresora, después debe indicar si la impresora esta lista, y el impreso saldrá en la impresora pre_establecida; Esta es una impresora predefinida Por.ejemplo:

Proporcione el nombre de la impresora : lpana

Esta lista la impresora? <S> Si o <N> No : S

Impresora esclava: En esta opción el usuario tan solo necesita indicar que la impresora esta lista, y el impreso saldrá en la impresora que esta conectada a la terminal (la terminal que esta usando en este momento puede tener una impresora esclava). Por.ejemplo:

Esta lista la impresora? <S> Si o <N> No : S

Archivo: Sirve para guardar en un archivo los registros previamente encontrados con la opción "Encuentra". El nombre se asigna en forma automática.

Acción Fin: Sirve para salir del menú y regresar al menú anterior.

MENU: ALTA BAJAS MODIFICACION CONSULTA SIGUIENTE PREVIO **REPORTE** FIN
 Genera reporte

[05/11/96]	FERROCARRILES NACIONALES DE MEXICO	[18:57]
SELECCION DE IMPRESORA		

A) IMPRESORA POR DEFAULT REGION B) DIRECCIONAR A PUERTO DESCRIPCION C) DIRECCIONAR A IMPRESORA D) IMPRESORA ESCLAVA S) SALIR		
OPCION []		

REGISTRO 2 DE 6

Figura No. IV.12

V. PANTALLAS DE APOYO PARA REGISTRO DE DATOS

Existe la posibilidad de invocar ayuda o apoyo para el registro de datos en algunos campos de las pantallas, para activarla se oprimen al mismo tiempo las teclas "Control-Z" y el sistema despliega la pantalla de apoyo correspondiente al catálogo del dato solicitado. En dicha pantalla solamente aparecen los datos más significativos por los cuales se puede hacer la búsqueda de información. Estas pantallas apoyan fundamentalmente las acciones de MODIFICACIÓN. Un ejemplo de las pantallas de apoyo se muestra en la Figura No. V.1.

DIGITE <ESC> PARA ACEPTAR REGISTRO <CTRL-C> PARA ABORTAR OPERACION																		
[05/11/96]	FERROCARRILES NACIONALES DE MEXICO	[20:42]																
	S I I T E S																	
TEFEGRESO	REGISTRO DE EGRESOS	[ZOOM]																
DOCUMENTO	STIPTAL	FECHA 05/10/96																
BANCO		REGION																
SUCURSAL	<table border="1"> <tr> <td colspan="2">SELECCION Y PRESIONE <ESC>, CANCEL</td> </tr> <tr> <td colspan="2">CATALOGO DE REGIONES</td> </tr> <tr> <td>REGION</td> <td>DESCRIPCION</td> </tr> <tr> <td>[1]</td> <td>[BUENAVISTA]</td> </tr> <tr> <td>[2]</td> <td>[MORESTE]</td> </tr> <tr> <td>[3]</td> <td>[PACIFICO-NORTE]</td> </tr> <tr> <td>[4]</td> <td>[CHIRIHUAHUA-PACI]</td> </tr> <tr> <td>[5]</td> <td>[VALLE DE MEXICO]</td> </tr> </table>		SELECCION Y PRESIONE <ESC>, CANCEL		CATALOGO DE REGIONES		REGION	DESCRIPCION	[1]	[BUENAVISTA]	[2]	[MORESTE]	[3]	[PACIFICO-NORTE]	[4]	[CHIRIHUAHUA-PACI]	[5]	[VALLE DE MEXICO]
SELECCION Y PRESIONE <ESC>, CANCEL																		
CATALOGO DE REGIONES																		
REGION			DESCRIPCION															
[1]			[BUENAVISTA]															
[2]	[MORESTE]																	
[3]	[PACIFICO-NORTE]																	
[4]	[CHIRIHUAHUA-PACI]																	
[5]	[VALLE DE MEXICO]																	
NUMERO DE CUENTA																		
TIPO DE DOCUMENT																		
FECHA PROC.																		
NUMERO CONTA																		
MONEDA																		
IMPORTE NACIONAL																		
IMPORTE EXTRANJE																		
MONTO DE PAGO																		
FORMA DE PAGO	EXPEDIENTE	TRAMITE																
CONCEPTO																		

Figura No. V.1

En el caso que aquí se ilustra aparecen solamente los datos "CLAVE" y "NOMBRE" mediante los cuales se buscará en el catálogo de "REGION" las claves que corresponden a las diferentes regiones.

Para buscar la información en las pantallas de apoyo o ayuda hay dos formas:

Primera. Oprimiendo directamente la tecla "Esc", como lo indica el mensaje. De esta manera el sistema busca todos los registros que contenga el catálogo respectivo (en este caso el catálogo de "REGION") y los despliega en la pantalla para poder elegir la clave o dato deseado mediante las teclas de flechas. Una vez seleccionado el dato, se vuelve a oprimir la tecla "Esc" para que el sistema lo tome y lo deje en el campo solicitado desde la pantalla de datos de donde se activo el "Control-Z".

Segunda: Colocando el cursor en algunos de los campos de datos de la pantalla de ayuda y precisando el dato de interés, es decir proporcionando un criterio de selección. En éste caso que se ilustra se eligió el campo "NOMBRE" y se tecló "PACIF*", como se ve en la Figura No. V.2, para que el sistema busque la palabra PACIFICO en los registros del catálogo de "GRUPO REGION". Enseguida volver a teclar "Esc" para que el sistema tome el dato y lo deje en el campo solicitado desde la pantalla de datos de donde se activo el "Control-Z".

DIGITE <ESC> PARA ACEPTAR REGISTRO
<CTRL-C> PARA ABORTAR OPERACIÓN

(05/11/96)	FERROCARRILES NACIONALES DE MEXICO	(20:42)
	S I I T E S	
TEFEGRESO	REGISTRO DE EGRESOS	(ZOOM)
DOCUMENTO	SIIPITAL	FECHA 05/10/96
BANCO		REGION
SECURSAL	SELECCION Y PRESIONE <ESC>, CANCEL	
NUMERO DE CUENTA	CATALOGO DE REGIONES	
	REGION	DESCRIPCION
TIPO DE DOCUMENT	[]	[*PACIF*]
FECHA PROG.	[]	[]
NUMERO CONTA	[]	[]
MONEDA	[]	[]
IMPORTE NACIONAL	[]	[]
IMPORTE EXTRANJE	[]	[]
MONTO DE PAGO		
FORMA DE PAGO	EXPEDIENTE	TRAMITE
CONCEPTO		

Figura No. V.2

El carácter "*" sirve como comodín de búsqueda, para substituir uno o más caracteres contiguos, cuando se requiere encontrar datos de una manera más general.

En el criterio de selección se emplean los asteriscos; además, es posible emplear el signo de interrogación "?", en cuyo caso precisa la substitución de un solo carácter por cada signo de interrogación.

Ambos, el asterisco o el signo de interrogación, se pueden utilizar al principio, en medio o al final de un dato o un conjunto de ellos.

NOTA: Cabe mencionar que la operación de las pantallas de datos antes descrita es la misma en todas las que se utilizan en el SIITES, es por esto que la explicación dada en éste capítulo es de suma importancia para operar adecuadamente el sistema.

PROCEDIMIENTO DE OPERACIÓN

Considerando que en las primeras páginas de este apéndice, se explico de manera detallada la forma de navegar por las pantallas de menús, el como invocar a las pantallas de ayuda, y de la actividad que realizan todas y cada una de las funciones que aparecen en las pantallas de trabajo.

Se describirá de manera breve y concisa el significado de cada campo en las diferentes pantallas de trabajo, así como la forma de llegar a las mismas.

VI.1 ADMINISTRACIÓN**VI.1.1 USUARIOS**

1. En el menú principal, elegir la opción 1) ADMINISTRACIÓN.
2. En el menú ADMINISTRACIÓN, elegir la opción 1) USUARIOS.
3. Los datos considerados son:

USUARIO	Clave del usuario. Hasta 10 caracteres alfanuméricos. Dato requerido. No existen dos usuarios con la misma clave.
NOMBRE	Nombre del usuario. Hasta 40 caracteres alfanuméricos. Dato requerido.
UTL. ACT.	Fecha de la última actualización. Formato dd/mm/aaaa. Dato opcional.
AUTORIZA	Responsable del control de usuarios Dato requerido.
NODO	Máquina en donde radica el sistema. Dato opcional.
SISTEMA	Sistema que accesa el usuario. Dato requerido.

VI.1.2 MENÚS

1. En el menú principal, elegir la opción 1) ADMINISTRACIÓN.
2. En el menú ADMINISTRACIÓN, elegir la opción 2) MENÚS.

3. Los datos considerados son:

CLAVE	Clave del menú. Dato requerido. No deben existir dos menús con la misma clave.
NOMBRE	Nombre del menú. Dato requerido.

VI.1.3 TÍTULOS

1. En el menú principal, elegir la opción 1) ADMINISTRACIÓN.
2. En el menú ADMINISTRACIÓN, elegir la opción 3) TÍTULOS.
3. Los datos considerados son:

TÍTULO PRINCIPAL	Letrero del primer renglón. Dato opcional.
SUBTÍTULO 1	Letrero del segundo renglón. Dato opcional.
SUBTÍTULO 2	Letrero del tercer renglón. Dato opcional.

VI.1.4 OPCIONES

1. En el menú principal, elegir la opción 1) ADMINISTRACIÓN.
2. En el menú ADMINISTRACIÓN, elegir la opción 4) OPCIONES.
3. Los datos considerados son:

CLAVE DEL MENÚ	Clave del menú. Dato requerido.
OPCIÓN	Clave de la opción. Dato requerido.
DESCRIPCIÓN	Descripción de la opción. Dato requerido.

TIPO	Que tipo de opción es Dato requerido.	MENU / PROG.
INSTRUCCIÓN	Instrucción que se lleva a cabo. Dato requerido.	
LETRA	Letra o número que corresponde a la opción dentro del menú en particular. Dato requerido.	
MENSAJE	Mensaje de ayuda que se despliega para la opción. Dato opcional.	

VI.1.5 RELACIONES

- En el menú principal, elegir la opción 1) ADMINISTRACIÓN.
- En el menú ADMINISTRACIÓN, elegir la opción 5) RELACIONES.
- Los datos considerados son:

USUARIO	Clave del usuario. Dato requerido.
MENÚ	Clave del menú. Dato requerido.
OPCIÓN	Opción que puede acceder el usuario. Dato requerido.
DESCRIPCIÓN	Descripción de la opción. Dato requerido.
PERMISO	*X* permite acceso. * * nulo niega acceso. Dato requerido.

VI.1.6 FUNCIONES

- En el menú principal, elegir la opción 1) ADMINISTRACIÓN.
- En el menú ADMINISTRACIÓN, elegir la opción 6) FUNCIONES.

3. Los datos considerados son:

USUARIO	Clave del usuario. Dato requerido.								
MENÚ	Clave del menú. Dato requerido.								
OPCIÓN	Opción que puede accesar el usuario. Dato requerido.								
DESCRIPCIÓN	Descripción de la opción. Dato requerido.								
PERMISO	<table> <tr> <td>1 = Altas.</td> <td>2 = Bajas</td> </tr> <tr> <td>3 = Altas, Bajas</td> <td>4 = Modificaciones</td> </tr> <tr> <td>5 = Altas, Modificaciones</td> <td>6 = Bajas, Modificaciones</td> </tr> <tr> <td>7 = Altas, Bajas, Modificaciones</td> <td></td> </tr> </table> <p>Si la opción es reporte:</p> <p>1 = Vídeo 2 = esclava 3 = imprimir 4 = Vídeo, esclava</p> <p>Dato requerido.</p>	1 = Altas.	2 = Bajas	3 = Altas, Bajas	4 = Modificaciones	5 = Altas, Modificaciones	6 = Bajas, Modificaciones	7 = Altas, Bajas, Modificaciones	
1 = Altas.	2 = Bajas								
3 = Altas, Bajas	4 = Modificaciones								
5 = Altas, Modificaciones	6 = Bajas, Modificaciones								
7 = Altas, Bajas, Modificaciones									

VI.1.7 INSTRUCCIONES

- En el menú principal, elegir la opción 1) ADMINISTRACIÓN.
- En el menú ADMINISTRACIÓN, elegir la opción 7) INSTRUCCIONES.
- Los datos considerados son:

INSTRUCCIÓN	Clave de la instrucción. Dato requerido.
DESCRIPCIÓN	Descripción de la instrucción. Dato requerido.
RUTA	Dirección de la instrucción. Dato requerido.

PROGRAMA 4GE	Nombre del programa si existe. Dato opcional.
PARÁMETROS	Parámetros que puede necesitar. Dato opcional.

VI.2 INGRESOS

VI.2.1 INGRESOS

1. En el menú principal, elegir la opción 2) INGRESOS.
2. En el menú de INGRESOS, elegir la opción 1) INGRESOS.
3. Los datos considerados son:

No. DE INGRESO	Clave del ingreso. Generado por el sistema. Dato requerido.
REGIÓN	Clave de la región. Dato requerido Debe existir en catálogo. Permite consulta al catálogo.
FECHA	Dato requerido. Default = Fecha del día. Formato = dd/mm/aaaa.
MOTIVO	Clave que identifica al tipo de ingreso. Debe existir en catálogo. Permite consulta al catálogo.
BANCO	Clave del banco. Puede consultar catálogo Dato requerido.
SUCURSAL	Clave de la sucursal. Puede consultar catálogo Dato requerido.

NÚMERO DE CUENTA	Clave de la cuenta bancaria. Puede consultar catálogo Dato requerido.
No. DE INVERSIÓN	Número del documento de inversión. Dato requerido o no, que depende del tipo de ingreso.
IMPORTE A REGISTRAR	Ingreso recibido. Dato requerido.
CONCEPTO	Dato opcional.
DETALLE CONTABLE	Captura el detalle contable del ingreso. Dato requerido.

PANTALLA DETALLE CONTABLE

FECHA DE PÓLIZA	Default = Fecha de proceso. Formato dd/mm/aaaa. Dato desplegado.
CUENTA, SUB-CUENTA, SUB-SUB-CUENTA, SUB-SUB-SUBCUENTA	Registro contable. Permite consulta al catálogo. Debe existir en el catálogo. Dato desplegado.
MOVIMIENTO	Dato requerido. Valores válidos: <A> Abono. <C> Cargo.
CARGO	Importe del cargo. Dato requerido.
ABONO	Importe del abono. Dato requerido.
TOTAL DE AFECTACIÓN	La suma de cargos y de abonos deben ser igual al importe de la inversión. La diferencia entre el importe de cargos menos el de abonos debe ser cero. Dato calculado.
TOTAL DEL IMPORTE	Importe registrado como ingreso. Dato calculado.

AFECTACIÓN DE
SALDOS

- a) Si se trata de devoluciones.
A los "abonos por devolución" se les suma el importe del ingreso cancelado, para la fecha de proceso.
- b) Otros ingresos.
Al rubro - "abonos adicionales" se les suma el importe del ingreso cancelado, para la fecha del proceso.
Dato calculado.

VI.3 EGRESOS

VI.3.1 REGISTRO DE DOCUMENTOS

1. En el menú principal, elegir la opción 3) EGRESOS
2. En el menú EGRESOS, elegir la opción 1) REGISTROS DE DOCUMENTOS
3. Los datos considerados son:

- a) Alta de un sólo documento.

PANTALLA PRINCIPAL

DOCUMENTO

Número de DOCUMENTO.
Dato Calculado.
No existen dos documentos con el mismo número.

REGIÓN

Clave de región.
Dato requerido.
Debe existir en catálogo
Permite consulta al catálogo.

SIIPTAL

Número de SIIPTAL.
Dato opcional.
Si el número de SIIPTAL es ingresado se obtiene de forma automática datos para el registro del documento a pagar, dichos datos se obtienen del enlace con el sistema SIIPTAL

FECHA	Fecha del Documento. Dato requerido. Formato dd/mm/aaaa. Proporcionado por el sistema de manera automática.
NOMBRE	Nombre del beneficiario. Dato requerido. Dato desplegado si existe No. de SIIPTAL.
CONCEPTO	Descripción del concepto. Dato opcional. Si existe No. de SIIPTAL, dato proporcionado por el sistema de presupuesto.
DEPENDENCIA	Clave de la dependencia. Si existe No. de SIIPTAL, dato proporcionado por el sistema de presupuesto.
No. PROVEEDOR	Clave del proveedor o beneficiario. Dato opcional. Si existe No. de SIIPTAL, dato proporcionado por el sistema de presupuesto.
FECHA PROCESO EN TESORERÍA.	Fecha de proceso en tesorería. Default = Fecha del día. Formato dd/mm/aaaa.
FEC. PROG. PAGO	Fecha programada de pago. Proporcionada por el sistema de presupuesto o fecha del día más uno. Formato dd/mm/aaaa.
TIPO DE DOCUMENTO	Tipo de documento de tesorería. Dato requerido. Consulta a catálogo. La clave debe existir en el catálogo de tipos de Documento
NÚMERO CONTABLE	Número de registro contable. Dato opcional.
CLAVE CONCENTRADORA	Clave concentradora. Dato requerido. Consulta a catálogo. La clave debe existir en el catálogo.

<p>IMPORTE A PAGARSE</p>	<p>Importe a pagarse en moneda nacional. Dato requerido.</p>
<p>MONEDA NACIONAL</p>	<p>Si existe No. de SIIPAL, dato proporcionado por el sistema de presupuesto.</p>
<p>MONEDA EXTRANJERA</p>	<p>Importe a pagarse en moneda extranjera. Igual a cero o proporcionado por el sistema de presupuesto.</p>
<p>DEDUCCIONES PRONTO PAGO : MONEDA NACIONAL</p>	<p>Deducción por pronto pago en moneda nacional. Default = 0.</p>
<p>MONEDA EXTRANJERA</p>	<p>Deducción por pronto pago en moneda extranjera. Default = 0.</p>
<p>IMPORTE FINAL</p>	<p>Importe a pagarse menos las deducciones por pronto pago. Importe neto a pagar en moneda nacional.</p>
<p>MONEDA NACIONAL</p>	<p>Calculado por el programa.</p>
<p>MONEDA EXTRANJERA</p>	<p>Importe neto a pagar en moneda extranjera. Calculado por el programa</p>
<p>MONEDA</p>	<p>Moneda utilizada para pagar al beneficiario. Dato requerido. <MN> Moneda Nacional. <DL> Dólares.</p>
<p>FORMA DE PAGO</p>	<p>Forma de pago. Dato requerido. La clave debe existir en el catálogo. < CH > Pago en cheque. < EF > Pago en efectivo. < T2 > Transferencias a banco.</p>
<p>EXPEDIENTE</p>	<p>Número de expediente. Dato requerido. Se captura para forma de pago < T2 >.</p>
<p>BANCO</p>	<p>Clave Banco. Dato requerido. Consulta a catálogo. La clave debe existir en el catálogo.</p>

SUCURSAL	Clave de sucursal. Dato requerido. Consulta a catálogo. La clave debe existir en el catálogo.
CUENTA BANCARIA	Cuenta Bancaria. Dato requerido. Consulta a catálogo. La clave debe existir en el catálogo.

PANTALLA 2 (AFECTACIONES CONTABLES)

FECHA DE PÓLIZA	Fecha de pago. Default = Nula. Formato dd/mmm/aaaa
CUENTA, SUB-CUENTA, SUB-SUB-CUENTA, SUB-SUB-SUBCUENTA	Registro contable. Permite consulta al catálogo. Debe existir en el catálogo. Dato desplegado.
MOVIMIENTO	Tipo de movimiento. < C > Cargo < A > Abono Dato requerido.

b) Alta de un conjunto de documentos a pagar con un mismo número de cheque.

PANTALLA ADICIONAL (DOCUMENTOS CON MISMO BENEFICIARIO).

Para la forma de pago en cheque existe la opción de registrar varios documentos de un mismo beneficiario (siempre y cuando los documentos estén presupuestados), proporcionando:

NÚMERO DE CHEQUE	Número de cheque común a todos los documentos. Dato requerido.
SIIPTAL	Número de SIIPTAL. Dato requerido. No se puede dar de alta dos documentos con un mismo número de SIIPTAL.

NÚMERO CONTABLE	Número de registro contable. Dato requerido.
IMPORTE EN MONEDA NAL.	Importe en moneda nacional. Proporcionado por el sistema de presupuesto.
IMPORTE EN MONEDA EXT.	Importe en moneda extranjera. Proporcionado por el sistema de presupuesto.
TOTAL MONEDA NACIONAL	Importe total en moneda nacional de los documentos a pagarse con un cheque. Dato calculado.
TOTAL MONEDA EXTRANJERA	Importe total en moneda extranjera de los documentos a pagarse con un cheque. Dato calculado.

VI.3.2 EMISIÓN DE CHEQUES

VI.3.2.1 ASIGNACIÓN MANUAL DE CHEQUES

1. En el menú principal, elegir la opción 3) EGRESOS .
2. En el menú EGRESOS, elegir la opción 2) EMISIÓN DE CHEQUES.
3. En el menú EMISIÓN DE CHEQUES, elegir la opción 1) ASIGNACIÓN DE CHEQUE MANUAL.
4. Los datos considerados son:

PANTALLA PRINCIPAL

BANCO	Consulta al catálogo. La clave debe existir en el catálogo. Dato requerido.
SUCURSAL	Consulta al catálogo. La clave debe existir en el catálogo. Dato requerido.
CUENTA BANCARIA	Consulta al catálogo. La clave debe existir en el catálogo. Dato requerido.

No. CHEQUE	No debe haberse registrado o pagado anteriormente. Dato requerido.
MONEDA	Moneda de pago < MN > Moneda Nacional. < DL > Dólares. Debe ser igual a moneda de pago de todos los documentos a registrarse. Dato requerido.
TIPO DE IMPRESIÓN	< M > Mecánica o Manual. < A > Automatizada. Dato requerido.
FECHA DE IMPRESIÓN	Solo se captura para impresión Mecánica (< M >). Formato dd/mm/aaaa.
BENEFICIARIO	Nombre del beneficiario. Se obtiene de los documentos capturados en el detalle, se puede modificar el nombre. Dato desplegado.
IMPORTE	Suma de los importes a pagar por cada documento. Calculado por el programa.

PANTALLA DETALLE

No. DOCUMENTO	Registra el número de documento a pagar. Debe haber sido capturado previamente y no debe estar pagado o asignado a otro cheque. Debe coincidir con la descripción del cheque. Es decir, debe ser del mismo tipo de moneda de pago e importe del documento.
BENEFICIARIO	Beneficiario del documento relacionado. Dato desplegado.
IMPORTE MN	Importe en moneda nacional del documento proporcionado. Dato desplegado.

VI.3.2.2 ASIGNACIÓN AUTOMÁTICA DE CHEQUES

1. En el menú principal, elegir la opción 3) EGRESOS .

2. En el menú EGRESOS, elegir la opción 2) EMISIÓN DE CHEQUES.
3. En el menú EMISIÓN DE CHEQUES, elegir la opción 2) ASIGNACIÓN AUTOMÁTICA.
4. Los datos considerados son:

BANCO	Consulta a catálogo. La clave debe existir en el catálogo. Dato requerido.
SUCURSAL	Consulta a catálogo. La clave debe existir en el catálogo. Dato requerido.
CUENTA BANCARIA	Consulta a catálogo. La clave debe existir en el catálogo. Dato requerido.
NUMERACIÓN INICIAL DE LA CHEQUERA	A partir de que número se asignarán cheques. Dato requerido.
NUMERACIÓN FINAL DE LA CHEQUERA	Hasta que número se asignarán cheques. Este número debe ser mayor al anterior. Dato requerido.

VI.3.2.3 GENERA CHEQUES

1. En el menú principal, elegir la opción 3)EGRESOS .
2. En el menú EGRESOS, elegir la opción 2) EMISIÓN DE CHEQUES.
3. En el menú EMISIÓN DE CHEQUES, elegir la opción 3) GENERACIÓN DE CHEQUES.
4. Los datos considerados son:

NÚMERO DE CHEQUE INICIAL	Este número debe coincidir con el primer cheque montado en la impresora y debió haber sido asignado previamente. Dato requerido.
-----------------------------	---

NÚMERO DE CHEQUE FINAL	Este número debe ser mayor al cheque inicial. Debe estar dentro de los cheques asignados previamente. Dato requerido.
BANCO	Clave del banco de la chequera montada. Dato requerido.
SUCURSAL	Clave de la sucursal de la chequera montada. Dato requerido.
CUENTA BANCARIA	Clave de la cuenta bancaria de la chequera montada. Dato requerido.

Reimpresión de cheques

Quando se mandan GENERAR CHEQUES y por alguna circunstancia es interrumpida esta impresión (por ejemplo: Se ató el papel, se defaso, etc.), se puede volver a mandar la impresión de los cheques que quedan pendientes indicando con esta opción el rango que no fue impreso y volviendo a GENERAR CHEQUES.

CHEQUE INICIAL	A partir de que cheque se mandará la reimpresión. Dato requerido.
CHEQUE FINAL	Hasta que cheque se manda la reimpresión. Debe ser mayor al cheque inicial. Dato requerido.
BANCO	Banco de la chequera. Dato requerido.
SUCURSAL	Sucursal de la chequera. Dato requerido.
CUENTA BANCARIA	No. de Cuenta de la chequera. Dato requerido.

VI.3.2.4 CANCELACIÓN DE CHEQUES

1. En el menú principal, elegir la opción 3) EGRESOS .
2. En el menú EGRESOS, elegir la opción 2) EMISIÓN DE CHEQUES.
3. En el menú EMISIÓN DE CHEQUES, elegir la opción 4) CANCELACIÓN DE CHEQUES.

4. Los datos considerados son:

BANCO	Clave del banco. Debe existir en el catálogo. Dato requerido.
SUCURSAL	Clave sucursal. Debe existir en el catálogo. Dato requerido.
CUENTA BANCARIA	Clave de la cuenta bancaria. Debe existir en el catálogo. Dato requerido.
No. DE CHEQUE	Debe estar registrado el cheque. Dato requerido.
BENEFICIARIO	Nombre o razón social de quien recibe el cheque. Dato desplegado.
IMPORTE	Cantidad por el cual sale el cheque. Dato desplegado.
STATUS DE CANCELACIÓN	Dato requerido. < 2 > Cancelado. Solo despliega el valor.
FECHA DE CANCELACIÓN	Dato requerido. Default = Fecha del día Formato dd/mm/aaaa.
MOTIVO DE CANCELACIÓN	Dato requerido. < 1 > Error de impresión. < 2 > Cancelación documento. < 3 > No usado. < 4 > Otros.
No. DE DOCUMENTO	No. del documento a pagar. Dato requerido.
BENEFICIARIO	Nombre o razón social de quien va a recibir el cheque. Dato desplegado.
IMPORTE DETALLE	Importe del documento a cancelar. Dato desplegado.

VI.3.3 REGISTRO DE PAGOS EFECTUADOS

1. En el menú principal, elegir la opción 3) EGRESOS.
2. En el menú EGRESOS, elegir la opción 3) REGISTRO DE PAGOS EFECTUADOS.
3. Los datos considerados son:

TIPO DE DOCUMENTO	Tipo de Documento es el campo en donde vamos a distinguir si el pago es por medio de Cheque, Traspaso o en Efectivo. Dato requerido.
BENEFICIARIO	Dato desplegado. Nombre del beneficiario o razón social.
No. DE CHEQUE	Dato desplegado. No. de la chequera.
BANCO	Clave del banco. Dato desplegado.
SUCURSAL	Clave de la sucursal. Dato desplegado.
CUENTA BANCARIA	No. de cuenta bancaria. Dato desplegado.
MONEDA	Clave de la moneda. Dato desplegado.
IMPORTE	Dato desplegado. Importe del cheque.
FECHA DE ENTREGA	Fecha de pago = Fecha de póliza contable. Default = Fecha del día. Formato dd/mm/aaaa.
PODER NOTARIAL	Clave que identifica al Poder Notarial presentado. Debe existir en catálogo. <CTRL-Z> para consultar catálogo.
No. DE SIIPTAL	No. del documento asignado por presupuesto. Dato opcional.

BENEFICIARIO	Nombre o razón social de quien va a recibir el pago. Dato requerido.
IMPORTE POR SIIPTAL	Valor en moneda nacional o extranjera del documento. Dato requerido.

VI.4 INVERSIONES

VI.4.1 TRASPASOS

1. En el menú principal, elegir la opción 4) INVERSIONES.
2. En el menú INVERSIONES, elegir la opción 1) TRASPASOS.
3. Los datos considerados son:

PANTALLA PRINCIPAL

No. DE TRASPASO	Default = Consecutivo que corresponde al traspaso. Dato requerido.
REGIÓN	Clave de la región. Dato requerido.
FECHA	Fecha en que se realiza el traspaso. Dato requerido. Default = Fecha del día. Formato = dd/mm/yy.
CUENTA FUENTE	Cuenta bancaria con saldo disponible para traspaso. Debe existir en catálogo. En la ayuda muestra todas las cuentas que tienen saldo disponible.
NOMBRE DE LA CTA. FUENTE	Dato desplegado.
SALDO DISPONIBLE	Dato disponible. Utilizando CTR-I muestra los movimientos globales de la cuenta. La obtención del saldo disponible se describe más adelante.
CUENTA DE TRASPASO	Cuenta con compromisos de pago. Debe existir en catálogo. En ayuda muestra todas las cuentas por su saldo.

NOMBRE DE LA CUENTA DE TRASPASO	Dato desplegado.
COMPROMISO	Pagos pendientes de efectuar para esta cuenta.
IMPORTE A TRASPASAR	El importe a traspasar está restringido por el saldo disponible.
TIPO DE TRASPASO	Dato requerido. < C > Cheque. < B > Banca Electrónica.
CLAVE DE BANCA	Clave de control proporcionada por el banco. Se captura si el tipo de traspaso = B.

VI.4.2 COTIZACIONES

1. En el menú principal, elegir la opción 4) INVERSIONES.
2. En el menú registro de INVERSIONES, elegir la opción 2) COTIZACIONES.
3. Los datos considerados son:

BANCO	Grupo financiero. Dato requerido. Debe existir en el catálogo. Permite consultar el catálogo.
FECHA	Fecha de la cotización. Dato requerido. Default = Fecha del día. Formato dd/mm/aaaa.
COTIZACIÓN	Clave que identifica el número de cotizaciones en el día. Se asigna automáticamente. Dato desplegado.
HORA	Hora en que se recibe la cotización. Default = Hora de captura. Dato desplegado.
PORCENTAJE	Porcentaje anual cotizado. Dato requerido.

INSTRUMENTO	Instrumento sobre el que se realizó la cotización. Dato requerido. Debe existir en catálogo. Permite consulta al catálogo.
-------------	---

VI.4.3 INVERSIONES

1. En el menú principal, elegir la opción 4) INVERSIONES.
2. En el menú INVERSIONES, elegir la opción 3) INVERSIONES.
3. Los datos considerados son:

No. DE INVERSIÓN	Dato desplegado. Clave que identifica al documento.
REGIÓN	Clave de la región. Permite consulta al catálogo. Dato requerido
FECHA	Fecha en que se realiza la inversión. Default = Fecha del día. Formato = dd/m/aaaa. Dato requerido.
BANCO	Dato requerido. Grupo financiero donde se realizará la inversión. Debe existir en catálogo. Deben existir cotizaciones previamente registradas para ese grupo financiero. Muestra los grupos financieros con cotizaciones registradas
NÚMERO DE COTIZACIÓN	Es el número que corresponde a la cotización registrada para el grupo financiero elegido. Dato requerido.
INSTRUMENTO	Corresponde a la cotización elegida. Dato desplegado.
PORCENTAJE RENDIMIENTO	Corresponde a la cotización elegida. Dato desplegado.
HORA	Hora en la que se registró la cotización. Dato desplegado.

BANCO	Clave del banco con saldo disponible para inversión. Debe existir en catálogo. Muestra todas las cuentas que tienen saldo disponible
SUCURSAL	Dato requerido. Clave de la sucursal.
CUENTA BANCARIA	Dato requerido. No. de cuenta bancaria.
SALDO DISPONIBLE	Dato desplegado. El total disponible en la cuenta bancaria.
IMPORTE A INVERTIR	El importe no está restringido por el saldo disponible. Pero para la correcta afectación de los saldos bancarios es recomendable que se restrinja la cantidad a invertir al saldo disponible, el cual debe ser el importe real. Dato requerido.
INTERESES	Dato calculado a partir del importe a invertir y el porcentaje de rendimiento.
PLAZO	Duración de la inversión. Dato requerido
INTERÉS POR PLAZO	Dato calculado a partir de los intereses esperados por el plazo. Dato desplegado.

VI.4.4 AJUSTES

1. En el menú principal, elegir la opción 4) INVERSIONES.
2. En el menú INVERSIONES, elegir la opción 4) AJUSTES.
3. Los datos considerados son:

No. DE AJUSTE	Número de Ajuste. No se puede dar de alta dos documentos con el mismo número de Ajuste. Dato requerido.
REGIÓN	Clave de la región. Dato desplegado.

FECHA	Fecha del movimiento de Ajuste. Dato requerido. Despliega en forma automática la fecha por default. Formato dd/mm/aaaa.
CONCEPTO	Descripción del concepto. Motivo por el cual se lleva a cabo el movimiento. Dato opcional.
IMPORTE A REGISTRAR	Importe del movimiento. Dato requerido.

VI.5 CATÁLOGOS

VI.5.1 ACTUALIZA TIPO DE CAMBIO

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 1) ACTUALIZA TIPO DE CAMBIO.
3. Los datos considerados son:

FECHA DE ACTUALIZACIÓN	Fecha de registro. Formato dd/mm/aaaa. Default = Fecha del día. Dato requerido.
CLAVE DE MONEDA	Dato requerido. Debe existir en catálogo. Muestra el catálogo.
TIPO DE CAMBIO	Dato requerido. Siete enteros y cuatro decimales.

VI.5.2 REGIONES

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 2) REGIONES.
3. Los datos considerados son:

CLAVE	Clave de la región. Longitud = 1. Tipo = Alfabético. Dato requerido
DESCRIPCIÓN	Nombre de la región. Longitud = 20. Tipo = Alfanumérico. Dato requerido

VI.5.3 CONCENTRADORAS

1. En el menú principal, elegir la opción 5) CATALOGOS.
2. En el menú CATÁLOGOS, elegir la opción 3) CONCENTRADORAS.
3. Los datos considerados son:

CLAVE	Identificador de la clave concentradora. Numérico. Dato requerido.
DESCRIPCIÓN	Descripción de la clave. Alfanumérico. Dato requerido.

VI.5.4 TIPOS DE DOCUMENTO

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 4) TIPOS DE DOCUMENTOS.
3. Los datos considerados son:

CLAVE	Clave del tipo. Numérico. Dato requerido.
DESCRIPCIÓN	Descripción del tipo de documento. Alfanumérico. Dato requerido.

VI.5.5 TIPOS DE INGRESO

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 5) TIPOS DE INGRESO.
3. Los datos considerados son:

CLAVE	Clave del ingreso. Numérico. Dato requerido.
DESCRIPCIÓN	Descripción del ingreso. Alfanumérico. Dato requerido.

VI.5.6 INSTRUMENTOS DE INVERSIÓN

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 6) INSTRUMENTOS DE INVERSIÓN.
3. Los datos considerados son:

CLAVE	Clave del instrumento. Numérico. Dato requerido.
DESCRIPCIÓN	Descripción del instrumento. Alfanumérico. Dato requerido.

VI.5.7 MOTIVOS DE CANCELACIÓN

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 7) MOTIVOS DE CANCELACIÓN.
3. Los datos considerados son:

CLAVE	Clave del Motivo. Numérico. Dato requerido.
DESCRIPCIÓN	Descripción del motivo. Alfanumérico. Dato requerido.

VI.5.8 TIPOS DE MONEDA

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 8) TIPOS DE MONEDA.
3. Los datos considerados son:

CLAVE	Clave de la moneda. Alfabético. Dato requerido.
DESCRIPCIÓN	Descripción de la moneda. Alfanumérico. Dato requerido.

VI.5.9 BANCOS

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 9) BANCOS.
3. Los datos considerados son:

CLAVE	Clave del banco. Alfabético. Dato requerido.
DESCRIPCIÓN	Descripción del banco. Alfanumérico. Dato requerido.

VI.5.10 SUCURSALES

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 10) SUCURSALES
3. Los datos considerados son:

BANCO	Clave del banco. Alfabético. Acceso al catálogo. Clave debe existir en el catálogo. Dato requerido.
SUCURSAL	Clave de la sucursal. Numérico. Dato requerido.
DESCRIPCIÓN SUCURSAL	Descripción de la sucursal. Alfanumérico. Dato requerido.
DOMICILIO	Alfanumérico. Dato opcional.
DELEG. Y MPIO.	Alfanumérico. Dato opcional.
POB., EDO. Y C.P.	Alfanumérico. Dato opcional.
PAÍS	Alfanumérico. Dato opcional.
TELÉFONO	Numérico. Dato opcional.
ATENCIÓN A CARGO	Nombre del gerente o ejecutivo de cuenta. Alfanumérico. Dato opcional.
REGIÓN	Clave de la región. Alfabético. Dato requerido.

VI.5.11 CUENTAS BANCARIAS

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 11) CUENTAS BANCARIAS.
3. Los datos considerados son:

BANCO	Dato requerido. Alfabético. Acceso al catálogo de bancos. Debe existir en el catálogo.
SUCURSAL	Dato requerido. Numérico. Acceso a catálogo. Debe existir en el catálogo.
NÚMERO DE CUENTA	Dato requerido. Alfanumérico.
DESCRIPCIÓN DE LA CUENTA	Dato requerido. Alfanumérico.
CUENTA CONTABLE	Numérico. Dato requerido
SUBCUENTA CONTABLE	Numérico. Dato opcional.
SUB-SUBCUENTA CONTABLE	Numérico. Dato opcional.
SUB-SUB-SUBCUENTA CONTABLE	Numérico. Debe existir en el catálogo la cuenta contable completa. Dato opcional.
SALDO MÍNIMO	Es el mínimo importe que debe existir en la cuenta para no sobre girar la chequera. Numérico. Dato requerido.

TIPO DE CUENTA Dato requerido.
 Numérico.
 < 0 > Gasto.
 < 1 > Inversión.
 < 3 > Ambas.

TIPO DE MOVIMIENTO Dato requerido.
 Alfabético.
 < C > Captación.
 < P > Pago.
 < A > Ambas.

VI.5.12 CUENTAS CONTABLES

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 12) CUENTAS CONTABLES.
3. Los datos considerados son:

REGIÓN	Alfabético Dato requerido. Clave de la región.
ENTIDAD	Alfanumérico. Dato requerido. Clave de la entidad.
FUNCIÓN	Alfanumérico. Dato requerido. Clave de la función
TRANSACCIÓN	Alfanumérico. Dato requerido Clave de la función
CUENTA CONTABLE	Numérico.
SUBCTA	Cuenta contable completa
SUBSUBCTA	Formato = #####
SUBSUBSUBCTA	Dato requerido.
PARTIDA	Alfanumérico. Clave de la partida Dato opcional.

LÍNEA	Alfanumérico. Línea del negocio Dato opcional.
ÁREA	Alfanumérico. Clave del área Dato opcional.
MOVIMIENTO	Númérico. Clave del movimiento Dato opcional.

VI.5.13 PODER NOTARIAL

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 12) PODER NOTARIAL.
3. Los datos considerados son:

No. CONSECUTIVO	Dato requerido. Está formado por la clave de la compañía y un consecutivo. Compañía: Longitud = 4. , Tipo = Alfanumérico. Consecutivo: Tipo = numérico.
COMPAÑÍA	Dato requerido. Descripción de la compañía. Longitud = 140. Tipo = Alfanumérico.
R.F.C.	Dato requerido. Longitud = 15. Tipo = Alfanumérico.
NOMBRE	Dato requerido. Longitud = 15. Tipo = Alfanumérico.
APELLIDO PATERNO	Dato requerido. Longitud = 15. Tipo = Alfanumérico.

APELLIDO MATERNO	Dato requerido. Longitud = 15. Tipo = Alfanumérico.
------------------	---

VI.5.14 PÓLIZAS CONTABLES

1. En el menú principal, elegir la opción 5) CATÁLOGOS.
2. En el menú CATÁLOGOS, elegir la opción 14) PÓLIZAS CONTABLES.
3. Los datos considerados son:

No. DE PÓLIZA	Dato requerido. Numérico.
---------------	------------------------------

DESCRIPCIÓN	Dato requerido. Alfanumérico.
-------------	----------------------------------

VI.6 REPORTES

VI.6.1 PÓLIZA DE CAJA

1. En el menú principal, elegir la opción 6) REPORTES.
2. En el menú REPORTES, elegir la opción 3) PÓLIZA DE CAJA..
3. Los datos considerados son:

FECHA INICIAL	Dato requerido. Fecha del día de elaboración de la póliza de caja. No puede ser mayor a la fecha final. Default fecha del día. Formato dd/mm/aaaa.
---------------	--

FECHA FINAL	Dato requerido. Fecha del día de elaboración de la póliza de caja. No puede ser menor a la fecha inicial. Default fecha del día. Formato dd/mm/aaaa.
-------------	--



APÉNDICE B

PANTALLA DE REGISTRO DE TRASPASOS

DATABASE siitesdb
 SCREEN SIZE 24 BY 80
 {

```

DOCUMENTO {f005 } FECHA {f007 } REGION {c}d ]
CVE.BANCO FUENTE {a0} ] {e } ]
CVE.SUCURSAL FUENTE {f000 } ] {f } ]
CVE.CUENTA FUENTE {f001 } ] {g } ]
CVE.BANCO TRASPASO {a1} ] {h } ]
CVE.SUCURSAL TRASPASO {f006 } ] {i } ]
CVE.CUENTA TRASPASO {f002 } ] {j } ]
CVE.MONEDA {k } ] {l } ]
IMPORTE TRASPASO {f003 } ] {m } ]
TIPO DE TRASPASO {a} ] {n } ]
CVE.TIPO DE TRASP. {f004 } ] CONCEPTO {f008 } ]
}
end
tables
tet02_traspasos
tet06_documento
tet21_movdoc
attributes
f005 = tet02_traspasos.f02inum_documento,REVERSE,UPSHIFT;
f007 = tet06_documento.f06_fecha_documento,REVERSE,REQUIRED,FORMAT="dd/mm/yyyy";
c = tet06_documento.f06_region,REVERSE,UPSHIFT;
comments = "Teclee region del documento";
a0 = tet02_traspasos.f02ibanco,UPSHIFT,REVERSE,REQUIRED;
comments = "Teclee cve.banco fuente";
f000 = tet02_traspasos.f02isucur,REVERSE;
comments = "Teclee cve.sucursal fuente";
f001 = tet02_traspasos.f02inumcuenta,UPSHIFT,REVERSE,REQUIRED;
comments = "Teclee el No.de cuenta fuente";
a1 = tet02_traspasos.f02ibancotras,UPSHIFT,REVERSE,REQUIRED;
comments = "Teclee cve.banco a donde se hara el traspaso";
f006 = tet02_traspasos.f02isucurtras,REVERSE;
comments = "Teclee cve.sucursal a donde se hara el traspaso";
f002 = tet02_traspasos.f02inumctatras,UPSHIFT,REVERSE,REQUIRED;
comments = "Teclee el No. de cuenta a donde se hara el traspaso";
k = tet21_movdoc.f21moneda,UPSHIFT,REVERSE;
comments = "Teclee cve.de moneda";
f003 = tet02_traspasos.f02_imptraspaso,REVERSE,REQUIRED;
FORMAT = "$$$,$$$,$$$,$$$,$$$,.&&&&";
comments = "Teclee el importe del traspaso";
a = tet02_traspasos.f02itipotras,UPSHIFT,REVERSE,REQUIRED,INCLUDE="(","B");
comments = "Teclee el tipo de traspaso 'C'cheque... 'B'anca electronica";
f004 = tet02_traspasos.f02_ban_che,UPSHIFT,REVERSE;

```

```
comments = "Teclee el No. de cheque o Cve. de banca electronica";
f008 = tet21_movdoc.t21_concepto.PROGRAM=vi.REVERSE,
comments = "Teclee el concepto del documento, teclee < SHIFT-I >";
b = FORMONLY.lt_tpotras.REVERSE;
d = FORMONLY.lt_region.REVERSE;
e = FORMONLY.lt_bancof.REVERSE;
f = FORMONLY.lt_sucurf.REVERSE;
g = FORMONLY.lt_ctaf.REVERSE;
h = FORMONLY.lt_bancof.REVERSE;
i = FORMONLY.lt_sucurf.REVERSE;
j = FORMONLY.lt_ctaf.REVERSE;
l = FORMONLY.lt_moneda.REVERSE;
end
INSTRUCTIONS
DELIMITERS " "
```

PROGRAMA DE REGISTRO DE TRASPASOS

```

#
#                               No. 1
#
# Base de datos:      sitesdb
# Sistema:           SITES
# Modulo:            Generacion de Mantenimientos.
# Pantalla:         tetraspa.per
# Prog. fuente:     tetraspa.4gl
# Prog. objeto:     tetraspa.4go
# Prog. ejecutable: tetraspa.4gl
# Archivo de ayuda: teatraspa.msg
# Proceso:          Registro de traspasos
# Objetivo:
#
# Analista:         Lourdes Nieto Aguilar
# Programador:      Lourdes Nieto Aguilar
# Fecha creacion:   11-13-1996
# Fecha ultima modificacion: 11-13-1996
# No. de version:   1
#
DATABASE sitesdb
# GLOBALS "globales.4gl"

GLOBALS
DEFINE
  gt_cont2,
  gt_cont3          SMALLINT,
  gt_tabla          CHAR(20),
  gt_ban132        CHAR(1),
  gt_cd             CHAR(80),
  gt_docto         LIKE tet02_traspasos.t02inum_docto,
  gt_mov           LIKE tet02_traspasos.t02imov
END GLOBALS
DEFINE
  old_reg,
  r_consul,
  r_bajmod,
  r_sig,
  r_pre,
  r_record         RECORD
  t02inum_docto   LIKE tet02_traspasos.t02inum_docto,
  t02imov         LIKE tet02_traspasos.t02imov,
  t06_fecdocto   LIKE tet06_documento.t06_fecdocto,
  t06iregion     LIKE tet06_documento.t06iregion,
  t02ibanco      LIKE tet02_traspasos.t02ibanco,
  t02isucur      LIKE tet02_traspasos.t02isucur,
  t02inumccta    LIKE tet02_traspasos.t02inumccta,
  t02ibancotras  LIKE tet02_traspasos.t02ibancotras,
  t02isucurtras  LIKE tet02_traspasos.t02isucurtras,
  t02inumctatras LIKE tet02_traspasos.t02inumctatras,
  t21imonedata   LIKE tet21_movdoc.t21imonedata,

```

```

t02_imptraspaso LIKE tet02_traspasos.t02_imptraspaso,
t02itipotrasas LIKE tet02_traspasos.t02itipotrasas,
t02_ban_che LIKE tet02_traspasos.t02_ban_che
END RECORD,
gt_concepto LIKE tet21_movdoc.t21_concepto

MAIN
DEFINE
  pt_pasw CHAR(18),
  pt_iton CHAR(78)

DEFER INTERRUPT
LET pt_pasw = ARG_VAL(1)
LET pt_iton = ARG_VAL(2)
LET gt_docto = NULL
LET gt_mov = NULL

CALL fd_menu(pt_pasw, pt_iton)
END MAIN

#####
# Muestra el primer nivel de menus #
#####
FUNCTION fd_menu(pt_nomb, pt_instr)
DEFINE
  num_act,
  num_reg INTEGER,
  ictra CHAR,
  lt_movmax,
  bandera SMALLINT,
  pt_nomb CHAR(18),
  pt_instr CHAR(78),
  lt_bcon SMALLINT,
  lt_npro CHAR(9),
  lt_ntit CHAR(80)

OPTIONS
  INPUT ATTRIBUTE (RED,MAGENTA),
  DISPLAY ATTRIBUTE (RED,MAGENTA),
  HELP FILE "teatraspa.blp"

LET lt_npro = "TEPTRASPA"
LET lt_ntit = "REGISTRO DE TRASPASOS"

PREPARE ejecproc FROM
"EXECUTE PROCEDURE f52_valperas(?,?)";
DECLARE curstop CURSOR FOR ejecproc

OPEN FORM TEPTRASPA FROM "teptraspa"
DISPLAY FORM TEPTRASPA
CALL fl_linea2(lt_npro, lt_ntit, 32)

SELECT * FROM tet07_afecta
WHERE ROWID = -1

```



```
INTO TEMP tmp_afecta07
```

```
MENU "MENU"
```

```
COMMAND "Alta" "Alta de un nuevo registro"  
HELP 11
```

```
IF lt_bcon THEN  
CALL f3_closetur()  
LET lt_bcon = FALSE  
END IF
```

```
LET letra = 'A'
```

```
OPEN curstop USING pt_nomb.pt_instr.letra  
FOREACH curstop INTO bandera  
END FOREACH  
CLOSE curstop
```

```
IF bandera THEN  
CALL f5_altat()  
ELSE  
ERROR " USTED NO TIENE PERMISO PARA ACCESAR ESTA OPCION "  
END IF
```

```
COMMAND "Bajas" "Elimina un registro."  
HELP 12
```

```
IF lt_bcon THEN  
LET letra = 'B'
```

```
OPEN curstop USING pt_nomb.pt_instr.letra  
FOREACH curstop INTO bandera  
END FOREACH  
CLOSE curstop
```

```
IF bandera THEN  
CALL f6_baja(num_act.num_reg.lt_movmax)  
RETURNING lt_bcon.num_act.num_reg.lt_movmax  
ELSE  
ERROR " USTED NO TIENE PERMISO PARA ACCESAR ESTA OPCION "  
END IF  
ELSE  
ERROR " PARA DAR DE BAJA UN REGISTRO PRIMERO DEBE CONSULTAR "  
NEXT OPTION "Consulta"  
END IF
```

```
COMMAND "Modificacion" "Actualiza un registro."  
HELP 13
```

```
IF lt_bcon THEN  
LET letra = 'M'
```

```
OPEN curstop USING pt_nomb.pt_instr.letra  
FOREACH curstop INTO bandera
```

```
END FOREACH
CLOSE curstop

IF bandera THEN
  CALL f7_modif(num_act,num_reg,lt_movmax)
  RETURNING num_act,num_reg,lt_movmax

  CALL f3_limpiar()
  CALL f2_despl(num_act,num_reg)
ELSE
  ERROR " USTED NO TIENE PERMISO PARA ACCESAR ESTA OPCION "
END IF
ELSE
  ERROR " PARA MODIFICAR UN REGISTRO PRIMERO DEBE CONSULTAR "
NEXT OPTION "Consulta"
END IF

COMMAND "Consulta" "Consulta de registros"
HELP 14

IF lt_bcon THEN
  CALL f3_closcurt()
  LET lt_bcon = FALSE
END IF

CALL f8_consult()
RETURNING lt_bcon,num_act,num_reg,lt_movmax

COMMAND "Siguiente" "Ver siguiente registro en la lista"
HELP 15

IF lt_bcon THEN
  CALL f9_sigreg(num_act,num_reg,lt_movmax)
  RETURNING num_act,num_reg,lt_movmax
ELSE
  ERROR " PARA VER SIGUIENTE REGISTRO PRIMERO DEBE CONSULTAR "
NEXT OPTION "Consulta"
END IF

COMMAND "Previo" "Ver registro anterior en la lista"
HELP 16

IF lt_bcon THEN
  CALL f9_prereg(num_act,num_reg,lt_movmax)
  RETURNING num_act,num_reg,lt_movmax
ELSE
  ERROR " PARA VER REGISTRO ANTERIOR PRIMERO DEBE CONSULTAR "
NEXT OPTION "Consulta"
END IF

COMMAND "Concepto" "Editar el concepto del documento" # HELP 17
IF lt_bcon THEN
  CALL desp_concept(gt_concepto)
ELSE
```

```

ERROR "PRIMERO DEBE CONSULTAR ALGUN REGISTRO"
NEXT OPTION "Consulta"
END IF

COMMAND "2a_pantalla " "Ver registro anterior en la lista"
HELP 16

IF It_bcon THEN
    CALL pantalla2a(gt,docto,gt,mo)
ELSE
    ERROR "PRIMERO DEBE CONSULTAR ALGUN REGISTRO"
    NEXT OPTION "Consulta"
ENDIF

COMMAND "Fin" "Fin de captura. Regresa a menu anterior"
IF It_bcon THEN
    CALL B_closcart()
ENDIF
EXIT MENU
END MENU
END FUNCTION

#####
# Facilidad para registrar mas de un documento
#####
FUNCTION IS_alta()
CALL I2_limpiar()
DISPLAY "DIGITE <ESC> PARA ACEPTAR REGISTRO" AT 1,1
ATTRIBUTE(RED,REVERSE)
DISPLAY "<CTRL-C> PARA ABORTAR OPERACION " AT 2,1
ATTRIBUTE(RED,REVERSE)

WHILE TRUE
CASE IS_regis()
WHEN 0
    ERROR "ALTA EFECTUADA..."
    SLEEP 2
WHEN 1
    ERROR "ERROR EN LA CAPTURA, VERIFIQUE SUS DATOS"
    SLEEP 2
WHEN 2
    ERROR "ALTA ABANDONADA, ULTIMOS DATOS IGNORADOS"
EXIT WHILE
END CASE
END WHILE

CLEAR FORM
END FUNCTION

#####
# Registro de documentos
#####
FUNCTION IS_regis()
DEFINE

```

orden	CHAR(500),
intentos	SMALLINT,
lt_numdocto	INTEGER,
lt_bancof.	
lt_bancos.	
lt_bancot.	
lt_sucurf.	
lt_sucurt.	
lt_ctaf.	
lt_ctat.	
lt_moneda.	
lt_tpotras.	
lt_region	CHAR(30),
lt_usuario	CHAR(10),
lt_status	SMALLINT,
lt_saldoof.	
lt_saldot.	
lt_impdjet	DECIMAL(15,2),
lt_mon1	LIKE tec04_ctab.c04imoned,
lt_salmin.	
lt_chsal	MONEY(18,4),
lt_tipcam	LIKE tec02_tipoca.c02_tipocam,
lt_cta1.lt_cta2	LIKE tec12_ctascont.c12ctacon

CLEAR FORM

LOCATE gt_concepto IN MEMORY

INPUT

r_record.t06 fecdocto,
 r_record.t06iregion,
 r_record.t02ibanco,
 r_record.t02isucur,
 r_record.t02inumcta,
 r_record.t02ibancotras,
 r_record.t02isucurtas,
 r_record.t02inumctatras,
 r_record.t21imoned,
 r_record.t02_imptraspaso,
 r_record.t02itipotras,
 r_record.t02_ban_che,
 gt_concepto

FROM

tet06_documento.t06_fecdocto,
 tet06_documento.t06iregion,
 tet02_traspasos.t02ibanco,
 tet02_traspasos.t02isucur,
 tet02_traspasos.t02inumcta,
 tet02_traspasos.t02ibancotras,
 tet02_traspasos.t02isucurtas,
 tet02_traspasos.t02inumctatras,
 tet21_movdoc.t21imoned,
 tet02_traspasos.t02_imptraspaso,
 tet02_traspasos.t02itipotras,
 tet02_traspasos.t02_ban_che,
 tet21_movdoc.t21_concepto

```
BEFORE FIELD t06_fecdocto
  LET r_record.t06_fecdocto = TODAY
  DISPLAY BY NAME r_record.t06_fecdocto
BEFORE FIELD t06iregion
  DISPLAY "[ZOOM]" AT 0.65 ATTRIBUTE:REVERSE,BLINK)

AFTER FIELD t06iregion
  IF intentos < 2 THEN
    IF r_record.t06iregion IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos + 1
      NEXT FIELD tet06_documento.t06iregion
    ELSE
      CALL f06_d_region(r_record.t06iregion) RETURNING it_region
      IF STATUS = NOTFOUND THEN
        ERROR "NO EXISTE REGION EN EL CATALOGO"
        NEXT FIELD tet06_documento.t06iregion
      ELSE
        DISPLAY BY NAME it_region
      END IF
      LET intentos = 0
    END IF
  ELSE
    CALL inicia()
    EXIT INPUT
  END IF

AFTER FIELD t02ibanco
  IF intentos < 2 THEN
    IF r_record.t02ibanco IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos + 1
      NEXT FIELD tet02_traspasos.t02ibanco
    ELSE
      CALL f01_d_banco(r_record.t02ibanco) RETURNING it_bancof
      IF STATUS = NOTFOUND THEN
        ERROR "NO EXISTE EL BANCO EN EL CATALOGO"
        NEXT FIELD tet02_traspasos.t02ibanco
      ELSE
        DISPLAY BY NAME it_bancof
      END IF
      LET intentos = 0
    END IF
  ELSE
    CALL inicia()
    EXIT INPUT
  END IF

AFTER FIELD t02isucur
  IF intentos < 2 THEN
    IF r_record.t02isucur IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos + 1
      NEXT FIELD tet02_traspasos.t02isucur
```

```
ELSE
CALL t07_d_sucursal(r_record.t02ibanco.r_record.t02isucur)
RETURNING lt_sucurf
IF STATUS = NOTFOUND THEN
ERROR "NO EXISTE LA SUCURSAL EN EL CATALOGO"
NEXT FIELD tet02_traspasos.t02isucur
ELSE
DISPLAY BY NAME lt_sucurf
END IF
LET intentos = 0
END IF
ELSE
CALL initia()
EXIT INPUT
END IF
AFTER FIELD t02inumeta
IF intentos < 2 THEN
IF r_record.t02inumeta IS NULL THEN
ERROR "EL CAMPO NO PUEDE SER NULO!!!"
LET intentos = intentos + 1
NEXT FIELD tet02_traspasos.t02inumeta
ELSE
CALL t13_d_ctaban(r_record.t02ibanco.r_record.t02isucur,
r_record.t02inumeta) RETURNING lt_ctaf
IF STATUS = NOTFOUND THEN
ERROR "NO EXISTE CUENTA BANCARIA"
NEXT FIELD tet02_traspasos.t02inumeta
ELSE
DISPLAY BY NAME lt_ctaf
END IF
LET intentos = 0
END IF
ELSE
CALL initia()
EXIT INPUT
END IF
AFTER FIELD t02ibancotras
IF intentos < 2 THEN
IF r_record.t02ibancotras IS NULL THEN
ERROR "EL CAMPO NO PUEDE SER NULO!!!"
LET intentos = intentos + 1
NEXT FIELD tet02_traspasos.t02ibancotras
ELSE
CALL t01_d_banco(r_record.t02ibancotras) RETURNING lt_bancof
IF STATUS = NOTFOUND THEN
ERROR "NO EXISTE EL BANCO EN EL CATALOGO"
NEXT FIELD tet02_traspasos.t02ibancotras
ELSE
DISPLAY BY NAME lt_bancof
END IF
LET intentos = 0
END IF
```

```
ELSE
  CALL initia()
  EXIT INPUT
END IF

AFTER FIELD t02isucurtras
  IF intentos < 2 THEN
    IF r_record.t02isucurtras IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos + 1
      NEXT FIELD tet02_traspasos.t02isucurtras
    ELSE
      CALL t07_d_sucursali(r_record.t02ibancotras,
        r_record.t02isucurtras) RETURNING it_sucurt
      IF STATUS = NOTFOUND THEN
        ERROR "NO EXISTE LA SUCURSAL EN EL CATALOGO"
        NEXT FIELD tet02_traspasos.t02isucurtras
      ELSE
        DISPLAY BY NAME it_sucurt
      END IF
      LET intentos = 0
    END IF
  ELSE
    CALL initia()
    EXIT INPUT
  END IF

AFTER FIELD t02inumctatras
  IF intentos < 2 THEN
    IF r_record.t02inumctatras IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos + 1
      NEXT FIELD tet02_traspasos.t02inumctatras
    ELSE
      CALL t13_d_ctaban(r_record.t02ibancotras,r_record.t02isucurtras,
        r_record.t02inumctatras) RETURNING it_ctat
      IF STATUS = NOTFOUND THEN
        ERROR "NO EXISTE CUENTA BANCARIA"
        NEXT FIELD tet02_traspasos.t02inumctatras
      ELSE
        IF r_record.t02inumeta = r_record.t02inumctatras THEN
          ERROR "NO PUEDEN REALIZARSE TRASPASOS SOBRE LA MISMA CUENTA"
          NEXT FIELD tet02_traspasos.t02ibancotras
        ELSE
          DISPLAY BY NAME it_ctat
        END IF
      END IF
      LET intentos = 0
    END IF
  ELSE
    CALL initia()
    EXIT INPUT
  END IF
```

```

BEFORE FIELD t21moneda
SELECT c041moneda INTO r_record.t21moneda
FROM tec04_ctab
WHERE c04ibanco = r_record.t02ibanco
AND c04isucur = r_record.t02isucur
AND c04inumeta = r_record.t02inumeta
DISPLAY BY NAME r_record.t21moneda

AFTER FIELD t21moneda
IF intentos > 2 THEN
IF r_record.t21moneda IS NULL THEN
ERROR "EL CAMPO NO PUEDE SER NULO!!!"
LET intentos = intentos + 1
NEXT FIELD tet21_movdoc.t21moneda
ELSE
CALL t05_d_monedastr_record(t21moneda) RETURNING lt_moneda
IF STATUS = NOTFOUND THEN
ERROR "NO EXISTE MONEDA EN EL CATALOGO"
NEXT FIELD tet21_movdoc.t21moneda
ELSE
DISPLAY BY NAME lt_moneda
END IF
LET intentos = 0
END IF
ELSE
CALL initiat
EXIT INPUT
END IF

BEFORE FIELD t02_imptraspaso
DISPLAY " AT 6,65

AFTER FIELD t02_imptraspaso
IF intentos < 2 THEN
IF r_record.t02_imptraspaso IS NULL OR
r_record.t02_imptraspaso <= 0 THEN
ERROR "EL CAMPO NO PUEDE SER NULO NI MENOR A CERO!!!"
LET intentos = intentos + 1
NEXT FIELD tet02_traspasos.t02_imptraspaso
ELSE
SELECT c04ictacon.c041moneda,c04_saldomin
INTO lt_cta1.lt_mon1.lt_salmin
FROM tec04_ctab
WHERE c04ibanco = r_record.t02ibanco
AND c04isucur = r_record.t02isucur
AND c04inumeta = r_record.t02inumeta

SELECT c04ictacon INTO lt_cta2
FROM tec04_ctab
WHERE c04ibanco = r_record.t02ibanco
AND c04isucur = r_record.t02isucur
AND c04inumeta = r_record.t02inumeta

LET lt_mon1 = r_record.t21moneda

```



```

IF r_record.t21imoneda != "MN" THEN
  LET lt_impdet = 0
  SELECT c02_tipocam INTO lt_tipcam
  FROM tet02_tipoca
  WHERE c02imoneda = r_record.t21imoneda
  AND c02fecha = r_record.t06_fecdocto
IF STATUS = NOTFOUND THEN
  ERROR "LA PARIDAD DEL DIA PARA LA MONEDA ",
  r_record.t21imoneda," NO SE HA DADO DE ALTA. VERIFIQUE!!!"
  SLEEP 4
  RETURN 2
ELSE
  LET lt_impdet = r_record.t02_imptraspaso * lt_tipcam
  END IF
ELSE
  LET lt_impdet = r_record.t02_imptraspaso
  END IF

SELECT t13_saldofin INTO lt_saldof
FROM tet13_saldos
WHERE t13ibanco = r_record.t02ibanco
AND t13isucur = r_record.t02isucur
AND t13inmcta = r_record.t02inmcta
AND t13ifecha = r_record.t06_fecdocto

LET lt_chsal = lt_saldof - lt_impdet
IF lt_chsal < lt_salmin THEN
  ERROR "NO SE DEBE SOBREPASAR EL SALDO MINIMO DE LA CUENTA.",
  " QUE ES ",lt_salmin
  NEXT FIELD tet02_traspasos.t02_imptraspaso
  END IF
  LET intentos = 0
  END IF
ELSE
  CALL inicia()
  EXIT INPUT
  END IF

AFTER FIELD t02itipotras
IF intentos < 2 THEN
  IF r_record.t02itipotras IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos + 1
    NEXT FIELD tet02_traspasos.t02itipotras
  ELSE
    IF r_record.t02itipotras <> "C" AND
    r_record.t02itipotras <> "B" THEN
      ERROR "ESTE VALOR NO ES VALIDO"
      NEXT FIELD t02itipotras
    ELSE
      IF r_record.t02itipotras = "C" THEN
        LET lt_ipotras = "VIA CHEQUE" CLIPPED
      ELSE
        LET lt_ipotras = "VIA BANCA ELECTRONICA" CLIPPED
      END IF
    END IF
  END IF

```

```
        END IF
        DISPLAY BY NAME It_ptoras
    END IF
    LET intentos = 0
    END IF
ELSE
    CALL initia()
    EXIT INPUT
END IF

AFTER FIELD t02_ban_che
IF intentos < 2 THEN
    IF r_record.t02_ban_che IS NULL THEN
        ERROR "EL CAMPO NO PUEDE SER NULO!!!"
        LET intentos = intentos + 1
        NEXT FIELD tet02_traspasos.t02_ban_che
    ELSE
        LET intentos = 0
    END IF
ELSE
    CALL initia()
    EXIT INPUT
END IF

ON KEY (CONTROL-Z)
CASE
    WHEN INFIELD(t06iregion)
        CALL f3_region() RETURNING r_record.t06iregion,It_region
        IF r_record.t06iregion IS NULL THEN
            ERROR "NO SELECCIONO REGION"
        ELSE
            DISPLAY BY NAME r_record.t06iregion,It_region
        END IF
    WHEN INFIELD(t02ibanco)
        CALL f1_banco() RETURNING r_record.t02ibanco,It_bancof
        IF r_record.t02ibanco IS NULL THEN
            ERROR "NO SELECCIONO BANCO"
        ELSE
            DISPLAY BY NAME r_record.t02ibanco,It_bancof
        END IF
    WHEN INFIELD(t02isucur)
        CALL f2_sucursal(r_record.t02ibanco)RETURNING r_record.t02isucur,
            It_sucurf
        IF r_record.t02isucur IS NULL THEN
            ERROR "NO SELECCIONO SUCURSAL"
        ELSE
            DISPLAY BY NAME r_record.t02isucur,It_sucurf
        END IF
    WHEN INFIELD(t02inumcta)
        CALL f4_cuentas(r_record.t02ibanco,r_record.t02isucur)
        RETURNING r_record.t02inumcta,It_ctaf
        IF r_record.t02inumcta IS NULL THEN
            ERROR "NO SELECCIONO CUENTA FUENTE"
        ELSE
```

```
        DISPLAY BY NAME r_record.t02inumcta.lt_ctaf
    END IF
    WHEN INFIELD(t02ibancotras)
        CALL fl_banco() RETURNING r_record.t02ibancotras.lt_banco
        IF r_record.t02ibancotras IS NULL THEN
            ERROR "NO SELECCIONO BANCO TRASPASO"
        ELSE
            DISPLAY BY NAME r_record.t02ibancotras.lt_banco
        END IF
    WHEN INFIELD(t02isucurtras)
        CALL t2_sucursal(r_record.t02ibancotras)
            RETURNING r_record.t02isucurtras.lt_sucrt
        IF r_record.t02isucurtras IS NULL THEN
            ERROR "NO SELECCIONO SUCURSAL TRASPASO"
        ELSE
            DISPLAY BY NAME r_record.t02isucurtras.lt_sucurt
        END IF
    WHEN INFIELD(t02inumctatras)
        CALL f4_cuentas(r_record.t02ibancotras,r_record.t02isucurtras)
            RETURNING r_record.t02inumctatras.lt_ctat
        IF r_record.t02inumctatras IS NULL THEN
            ERROR "NO SELECCIONO CUENTA TRASPASO"
        ELSE
            DISPLAY BY NAME r_record.t02inumctatras.lt_ctat
        END IF
    WHEN INFIELD(t21imoneda)
        CALL f8_moneda() RETURNING r_record.t21imoneda.lt_moneda
        IF r_record.t21imoneda IS NULL THEN
            ERROR "NO SELECCIONO MONEDA"
        ELSE
            DISPLAY BY NAME r_record.t21imoneda.lt_moneda
        END IF
    END CASE

ON KEY (CONTROL-W,f6)
CASE
    WHEN INFIELD(t02inum_docto)
        CALL SHOWHELP( 10)
    WHEN INFIELD(t02inumcia)
        CALL SHOWHELP( 20)
    WHEN INFIELD(t02inumctatras)
        CALL SHOWHELP( 30)
    WHEN INFIELD(t02_imptrasaso)
        CALL SHOWHELP( 40)
    WHEN INFIELD(t02itipotras)
        CALL SHOWHELP( 50)
    WHEN INFIELD(t02_cvebanelec)
        CALL SHOWHELP( 60)
END CASE

    ON KEY (CONTROL-C)
        LET INT_FLAG = TRUE
        EXIT INPUT
END INPUT
```

```

IF INT_FLAG THEN
LET INT_FLAG = FALSE
DISPLAY " " AT 6,65
RETURN 2
END IF

LET lt_usuario = "sities"

CALL detalle(lt_impdet,"1",r_record.t02inum_docto,1,lt_cta1,lt_cta2,"T")
RETURNING lt_status
IF lt_status = 0 THEN
BEGIN WORK
INSERT INTO tet06_documento VALUES(0,lt_usuario,r_record.t06iregion,
r_record.t06_fechacto,"A")
LET lt_numdocto = SOLCA.SOLERRD[2]
DISPLAY lt_numdocto TO t02inum_docto
SLEEP 3
INSERT INTO tet21_movdoc VALUES(lt_numdocto,1,3,gt_concepto,
r_record.t21moneda,r_record.t06_fechacto,"T",1)
INSERT INTO tet02_traspasos VALUES(lt_numdocto,1,r_record.t02ibanco,
r_record.t02isucur,r_record.t02inumeta,r_record.t02ibancotras,
r_record.t02isucurtras,r_record.t02inumettras,
r_record.t02_imptraspaso,r_record.t02itipotras,r_record.t02_ban_che)
UPDATE imp_afecta07
SET t07inum_docto = lt_numdocto
INSERT INTO tet07_afecta SELECT * FROM imp_afecta07
SELECT t13_saldofin INTO lt_saldof
FROM tet15_saldos
WHERE t13ibanco = r_record.t02ibancotras
AND t13isucur = r_record.t02isucurtras
AND t13inumeta = r_record.t02inumettras
AND t13ifecha = r_record.t06_fechacto

LET lt_saldof = lt_saldof - lt_impdet
LET lt_saldot = lt_saldof + lt_impdet

UPDATE tet13_saldos
SET t13_saldofin = lt_saldof,
t13_car_tra = t13_car_tra + lt_impdet
WHERE t13ibanco = r_record.t02ibanco
AND t13isucur = r_record.t02isucur
AND t13inumeta = r_record.t02inumeta
AND t13ifecha = r_record.t06_fechacto
UPDATE tet13_saldos
SET t13_saldofin = lt_saldot,
t13_abo_tra = t13_abo_tra + lt_impdet
WHERE t13ibanco = r_record.t02ibancotras
AND t13isucur = r_record.t02isucurtras
AND t13inumeta = r_record.t02inumettras
AND t13ifecha = r_record.t06_fechacto
IF STATUS = 0 THEN
COMMIT WORK
ELSE
ROLLBACK WORK

```

```

END IF
ELSE
RETURN 2
END IF

IF STATUS <> 0 THEN
RETURN 1
ELSE
RETURN 0
END IF
END FUNCTION

```

```

#####
# Búsqueda de registros
#####
FUNCTION fB_consul()

```

```

DEFINE
    num_act,
    num_reg,
    num_rec          INTEGER,
    txt_cons,
    con_cons        CHAR(1000),
    lt_codig        SMALLINT,
    lt_movmax       LIKE lt21_movdoc.21imov

```

```
CALL f2_limpiar()
```

```
MESSAGE * INDIQUE LAS CONDICIONES DE CONSULTA Y PRESIONE <ESC> *
ATTRIBUTE(RED,REVERSE)
CLEAR FORM

```

```
CONSTRUCT BY NAME con_cons ON

```

```

t02inum_docto,
t06_fecdocto,
t06iregion,
t02ibanco,
t02isucur,
t02inumcta,
t02bancotras,
t02isucurtras,
t02inumctatras,
c21imoneda,
t02_imprtraspaso,
t02itipotras,
t02_ban_che

```

```
END CONSTRUCT

```

```
IF INT_FLAG THEN

```

```

LET num_act = 0
LET num_reg = 0
LET INT_FLAG = FALSE
ERROR * CONSULTA ABANADONADA *
CLEAR FORM
RETURN lt_codig,num_act,num_reg,lt_movmax
END IF

```

```

LET txt_cons = "SELECT tet06_documento,t06inum_docto,max(t02imov)",
  " FROM tet02_traspasos,tet06_documento,tet21_movdoc ",
  " WHERE t06inum_docto = t21inum_docto ",
  " AND t21inum_docto = t02inum_docto ",
  " AND t21imov = t02imov ",
  " AND t02inum_docto = t21inum_docto ",
  " AND ", con_cons CLIPPED,
  " GROUP BY tet06_documento,t06inum_docto "

```

```

PREPARE set_cons FROM txt_cons
DECLARE cur_consul SCROLL CURSOR WITH HOLD FOR set_cons

```

```
LET num_reg = 0
```

```

FOREACH cur_consul
  LET num_reg = num_reg + 1
END FOREACH

```

```

LET num_act = 1
OPEN cur_consul
FETCH FIRST cur_consul INTO num_rec,lt_movmax

```

```
IF STATUS = NOTFOUND THEN
```

```

  LET num_act = 0
  LET num_reg = 0
  ERROR "NINGUN REGISTRO CUMPLE LAS CONDICIONES, VERIFIQUE "
  CLOSE cur_consul
  CLEAR FORM
  RETURN lt_codig,num_act,num_reg,lt_movmax

```

```
ELSE
```

```

  LET txt_cons = "SELECT t02inum_docto,t02imov,t06_fecdocto, ",
    "t06Region,t02ibanco,t02isucur,t02inumcta, ",
    "t02ibucotras,t02isucotras,t02inumctatras,t21imonedada, ",
    "t02_imptraspaso,t02tipotras,t02_ban_che ",
    "FROM tet02_traspasos,tet06_documento,tet21_movdoc ",
    "WHERE t06inum_docto = t02inum_docto ",
    "AND t06inum_docto = t21inum_docto ",
    "AND t21imov = t02imov ",
    "AND tet02_traspasos.t02inum_docto = ? ",
    "AND t02imov = ? ",
    "ORDER BY ",
    "t02inum_docto"

```

```
PREPARE set_consulta FROM txt_cons
```

```

DECLARE cur_consulta SCROLL CURSOR FOR set_consulta
END IF

```

```

CALL ft2_despli(num_act,num_reg)
DISPLAY "REGISTRO ",num_act," DE ",num_reg AT 23,60
ATTRIBUTE(RED,REVERSE)

```

```

OPEN cur_consulta USING num_rec,lt_movmax
FETCH FIRST cur_consulta INTO r_consul.*

```

```

LET lt_codig = TRUE
CALL f11_display(r_consul.*)

MENU "Consulta"
COMMAND "Siguiente" "Ver siguiente registro en la lista de consulta"
HELP 15
CALL f9_sigreg(num_act,num_reg,lt_movmax)
RETURNING num_act,num_reg,lt_movmax

COMMAND "Previo" "Ver registro previo en la lista de consulta" HELP 16
CALL f9_prereg(num_act,num_reg,lt_movmax)
RETURNING num_act,num_reg,lt_movmax

COMMAND "Inicial" "Ver registro inicial en la lista de consulta"
HELP 17
FETCH FIRST cur_consul INTO num_rec,lt_movmax
OPEN cur_consulta USING num_rec,lt_movmax
FETCH FIRST cur_consulta INTO r_consul.*
LET num_act = 1
CALL f3_limpiar()
DISPLAY "REGISTRO ".num_act." DE ".num_reg AT 23.60
ATTRIBUTE(RED,REVERSE)
CALL f11_display(r_consul.*)
NEXT OPTION "Siguiente"

COMMAND "Ultimo" "Ver ultimo registro en la lista de consulta" HELP 18
FETCH LAST cur_consul INTO num_rec,lt_movmax
OPEN cur_consulta USING num_rec,lt_movmax
FETCH LAST cur_consulta INTO r_consul.*
LET num_act = num_reg
CALL f3_limpiar()
CALL f12_despli(num_act,num_reg)
CALL f11_display(r_consul.*)
NEXT OPTION "Previo"

COMMAND "Fin" "Salir de menu de consulta"
EXIT MENU
END MENU
LET gt_docto = r_consul.t02inum_docto
LET gt_mov = r_consul.t02imov
RETURN lt_codig,num_act,num_reg,lt_movmax
END FUNCTION

```

```

#####
# Baja de registros #
#####
FUNCTION f6_baja(num_act,num_reg,lt_movmax)
DEFINE
lt_movmax,
lt_bcon SMALLINT,
num_act,
num_reg,
num_rec INTEGER,
lt_impdet like tet07_afecta.t07_importe,

```

```

it_resp CHAR(1)

FETCH CURRENT cur_consul INTO num_rec,it_movmax
OPEN cur_consulta USING num_rec,it_movmax
FETCH FIRST cur_consulta INTO r_bajmod.*

CALL p1_display(r_bajmod.*)
CALL f2_limpiar()
IF r_bajmod.t06_fecdocto = TODAY THEN
OPEN WINDOW sdwbaja AT 13,15
WITH 5 ROWS,40 COLUMNS
ATTRIBUTE(BORDER,MENU LINE LAST - 1)

DISPLAY " DESEA DAR DE BAJA ESTE REGISTRO " AT 2,4
ATTRIBUTE(RED)
MENU ""
BEFORE MENU
DISPLAY " " AT 3,3

COMMAND KEY ("N") " No "
LET it_resp = "N"
EXIT MENU

COMMAND KEY ("S") " Si "
LET it_resp = "S"
EXIT MENU
END MENU
CLOSE WINDOW sdwbaja

IF it_resp = "S" THEN
LET num_reg = num_reg - 1
BEGIN WORK
SELECT t07_importe INTO it_impdet
FROM tet07_afecta
WHERE t07inum_docto = r_bajmod.t02inum_docto
AND t07imov = r_bajmod.t02imov
AND t07_car_abo = "A"

UPDATE tet13_saldos
SET t13_saldofin = t13_saldofin + it_impdet,
t13_car_tra = t13_car_tra - it_impdet
WHERE t13ibanco = r_bajmod.t02ibanco
AND t13isucur = r_bajmod.t02isucur
AND t13inumcta = r_bajmod.t02inumcta
AND t13ifecha = r_bajmod.t06_fecdocto

UPDATE tet13_saldos
SET t13_saldofin = t13_saldofin - it_impdet,
t13_abo_tra = t13_abo_tra - it_impdet
WHERE t13ibanco = r_bajmod.t02ibancotras
AND t13isucur = r_bajmod.t02isucurtras
AND t13inumcta = r_bajmod.t02inumctatras
AND t13ifecha = r_bajmod.t06_fecdocto

```



```
DELETE FROM tet02_traspasos
WHERE t02inum_docto = r_bajmod.t02inum_docto
AND t02imov = r_bajmod.t02imov
DELETE FROM tet07_afecta
WHERE t07inum_docto = r_bajmod.t02inum_docto
AND t07imov = r_bajmod.t02imov
DELETE FROM tet21_movdoc
WHERE t21inum_docto = r_bajmod.t02inum_docto
AND t21imov = r_bajmod.t02imov

IF STATUS = 0 THEN
  COMMIT WORK
  LET num_reg = 0
ELSE
  ROLLBACK WORK
  LET num_reg = 0
END IF
FETCH NEXT cur_consul INTO num_rec.lt_movmax
IF STATUS = NOTFOUND THEN
  FETCH PRIOR cur_consul INTO num_rec.lt_movmax
  IF STATUS = NOTFOUND THEN
    LET num_reg = 0
  ELSE
    LET num_act = num_act - 1
    OPEN cur_consulta USING num_rec.lt_movmax
    FETCH FIRST cur_consulta INTO r_bajmod.*
    CALL f11_display(r_bajmod.*)
  END IF
ELSE
  LET num_act = num_act
  OPEN cur_consulta USING num_rec.lt_movmax
  FETCH FIRST cur_consulta INTO r_bajmod.*
  CALL f11_display(r_bajmod.*)
END IF
ERROR "REGISTRO ELIMINADO"
ELSE
  ERROR "BAJA ABANDONADA"
END IF
ELSE
  ERROR "NO SE PUEDEN DAR DE BAJA DOCUMENTOS DE DIAS ANTERIORES"
END IF

IF num_reg = 0 THEN
  LET lt_bcon = FALSE
  CLOSE cur_consulta
  CLOSE cur_consul
  CLEAR FORM
  DISPLAY " " AT 23,60
ELSE
  LET lt_bcon = TRUE
  CALL f3_limpia()
  CALL f12_despli(num_act,num_reg)
END IF
SLEEP 2
```

```

RETURN lt_bcon,num_act,num_reg,lt_movmax
END FUNCTION

#####
# Modificacion de registros
#####
FUNCTION f7_modif(num_act,num_reg,lt_movmax)
DEFINE
    num_reg,
    num_act,
    num_rec      INTEGER,
    intentos     SMALLINT,
    orden        CHAR(500),
    lt_bancof,
    lt_bancof,
    lt_bancos,
    lt_sucurf,
    lt_sucurt,
    lt_ctaf,
    lt_ctat,
    lt_moneda,
    lt_region    CHAR(30),
    lt_movmax,
    lt_status    SMALLINT,
    lt_saldof,
    lt_saldot,
    lt_impdet    DECIMAL(15,2),
    lt_salmin,
    lt_chsal,
    lt_nwon1     MONEY(18,4),
    lt_tipcam    LIKE tec04_ctab.c04imoneda,
    lt_eta1,lt_eta2 LIKE tec02_tipoca.c02_tipocam,
    lt_eta1,lt_eta2 LIKE tec12_ctasecont.c12ietacan

FETCH CURRENT cur_consul INTO num_rec,lt_movmax

OPEN cur_consulta USING num_rec,lt_movmax
FETCH FIRST cur_consulta INTO r_record.*

CALL f11_display(r_record.*)
IF r_record.t06_fecdocto >= TODAY THEN
    LET intentos = 0
    LET old_reg.* = r_record.*

INPUT
    r_record.t06iregion,
    r_record.t02ibanco,
    r_record.t02isucur,
    r_record.t02inumcta,
    r_record.t02ibancotras,
    r_record.t02isucurtras,
    r_record.t02inumctatras,
    r_record.t21imoneda,
    r_record.t02_imptraspaso,
    r_record.t02itipotras,

```

```

RETURN lt_bcon,num_act,num_reg,lt_movmax
END FUNCTION

```

```

#####
# Modificación de registros
#####
FUNCTION f7_modif(num_act,num_reg,lt_movmax)
  DEFINE
    num_reg,
    num_act,
    num_rec      INTEGER,
    intentos     SMALLINT,
    orden        CHAR(500),
    lt_bancof,
    lt_bancot,
    lt_bancos,
    lt_sucurf,
    lt_sucurt,
    lt_ctaf,
    lt_ctat,
    lt_moneda,
    lt_region    CHAR(30),
    lt_movmax,
    lt_status    SMALLINT,
    lt_saldof,
    lt_saldot,
    lt_impdet    DECIMAL(15,2),
    lt_salmin,
    lt_chsal     MONEY(18,4),
    lt_mon1      LIKE tec04_ctab.c041moneda,
    lt_tipeam    LIKE tec02_tipoca.c02_tipocam,
    lt_cta1,lt_cta2 LIKE tec12_ctascont.c121ctacon
  #####

```

```

FETCH CURRENT cur_consul INTO num_rec,lt_movmax

```

```

OPEN cur_consulta USING num_rec,lt_movmax
FETCH FIRST cur_consulta INTO r_record.*

```

```

CALL f11_display(r_record.*)
IF r_record.t06_fechacto = TODAY THEN
  LET intentos = 0
  LET old_reg.* = r_record.*

```

```

INPUT

```

```

r_record.t06region,
r_record.t02banco,
r_record.t02isucur,
r_record.t02numcta,
r_record.t02ibancotras,
r_record.t02isucurtas,
r_record.t02inumetatra,
r_record.t21moneda,
r_record.t02_imprtraspaso,
r_record.t02itipotras.

```

```

r_record.t02_ban_che.
gt_concepto
WITHOUT DEFAULTS
FROM
tet06_documento t06region.
tet02_traspasos.t02ibanco.
tet02_traspasos.t02isucur.
tet02_traspasos.t02inumcta.
tet02_traspasos.t02ibancoctras.
tet02_traspasos.t02isucurtras.
tet02_traspasos.t02inumctatras.
tet21_movdoc.t21movmeda.
tet02_traspasos.t02_impra-paso.
tet02_traspasos.t02tipotras.
tet02_traspasos.t02_ban_che.
tet21_movdoc.t21_concepto

```

```

BEFORE FIELD t06region
DISPLAY "[ZOOM]" AT 6,65 ATTRIBUTE(REVERSE,BLINK)

```

```

AFTER FIELD t06region

```

```

IF intentos < 2 THEN

```

```

  IF r_record.t06region IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos + 1
    NEXT FIELD tet06_documento t06region
  ELSE

```

```

    CALL t06_d_region(r_record.t06region) RETURNING lt_region
    IF STATUS = NOTFOUND THEN
      ERROR "NO EXISTE REGION EN EL CATALOGO"
      NEXT FIELD tet06_documento t06region
    ELSE
      DISPLAY BY NAME lt_region
      END IF
    LET intentos = 0
  END IF

```

```

END IF

```

```

ELSE

```

```

  CALL initac()
  EXIT INPUT
END IF

```

```

BEFORE FIELD t02ibanco

```

```

DISPLAY "[ZOOM]" AT 6,65 ATTRIBUTE(REVERSE,BLINK)

```

```

AFTER FIELD t02ibanco

```

```

IF intentos < 2 THEN

```

```

  IF r_record.t02ibanco IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos + 1
    NEXT FIELD tet02_traspasos.t02ibanco
  ELSE

```

```

    CALL t01_d_banco(r_record.t02ibanco) RETURNING lt_bancof
    IF STATUS = NOTFOUND THEN
      ERROR "NO EXISTE EL BANCO EN EL CATALOGO"

```

```

        NEXT FIELD tet02_traspasos.t02ibanco
    ELSE
        DISPLAY BY NAME !:_bancof
    END IF
    LET intentos = 0
END IF
ELSE
    CALL .initia()
    EXIT INPUT
END IF

BEFORE FIELD t02isucur
    DISPLAY " [ZOOM]" AT 6.65 ATTRIBUTE(REVERSE,BLINK)

AFTER FIELD t02isucur
    IF intentos < 2 THEN
        IF r_record.t02isucur IS NULL THEN
            ERROR "EL CAMPO NO PUEDE SER NULO!!!"
            LET intentos = intentos + 1
            NEXT FIELD t02isucur
        ELSE
            CALL t07_d_sucursale_r_record.t02ibanco.r_record.t02isucur)
                RETURNING lt_sucurf
            IF STATUS = NOTFOUND THEN
                ERROR "NO EXISTE LA SUCURSAL EN EL CATALOGO"
                NEXT FIELD tet02_traspasos.t02isucur
            ELSE
                DISPLAY BY NAME lt_sucurf
            END IF
            LET intentos = 0
        END IF
    ELSE
        CALL .initia()
        EXIT INPUT
    END IF

BEFORE FIELD t02inumcta
    DISPLAY " " AT 6.65

AFTER FIELD t02inumcta
    IF intentos < 2 THEN
        IF r_record.t02inumcta IS NULL THEN
            ERROR "EL CAMPO NO PUEDE SER NULO!!!"
            LET intentos = intentos + 1
            NEXT FIELD tet02_traspasos.t02inumcta
        ELSE
            CALL t13_d_ctaban(r_record.t02ibanco.r_record.t02isucur,
                r_record.t02inumcta) RETURNING lt_ctaf
            IF STATUS = NOTFOUND THEN
                ERROR "NO EXISTE CUENTA BANCARIA"
                NEXT FIELD tet02_traspasos.t02inumcta
            ELSE
                DISPLAY BY NAME lt_ctaf
            END IF
        END IF
    END IF

```

```
    LET intentos = 0
  END IF
ELSE
  CALL inicia()
  EXIT INPUT
END IF
```

```
BEFORE FIELD t02ibancotras
  DISPLAY "[ZOOM]" AT 6.65 ATTRIBUTE(REVERSE,BLINK)
```

```
AFTER FIELD t02ibancotras
  IF intentos > 2 THEN
    IF r_record.t02ibancotras IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos + 1
      NEXT FIELD tet02_traspasos.t02ibancotras
    ELSE
      CALL t01_d_banco(r_record.t02ibancotras) RETURNING lt_banco
      IF STATUS = NOTFOUND THEN
        ERROR "NO EXISTE EL BANCO EN EL CATALOGO"
        NEXT FIELD tet02_traspasos.t02ibancotras
      ELSE
        DISPLAY BY NAME lt_banco
      END IF
      LET intentos = 0
    END IF
  ELSE
    CALL inicia()
    EXIT INPUT
  END IF
```

```
BEFORE FIELD t02isucurtras
  DISPLAY "[ZOOM]" AT 6.65 ATTRIBUTE(REVERSE,BLINK)
```

```
AFTER FIELD t02isucurtras
  IF intentos < 2 THEN
    IF r_record.t02isucurtras IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos + 1
      NEXT FIELD tet02_traspasos.t02isucurtras
    ELSE
      CALL t07_d_sucursal(r_record.t02ibancotras,
        r_record.t02isucurtras) RETURNING lt_sucur
      IF STATUS = NOTFOUND THEN
        ERROR "NO EXISTE LA SUCURSAL EN EL CATALOGO"
        NEXT FIELD tet02_traspasos.t02isucur
      ELSE
        DISPLAY BY NAME lt_sucur
      END IF
      LET intentos = 0
    END IF
  ELSE
    CALL inicia()
    EXIT INPUT
  END IF
```

```

END IF

BEFORE FIELD t02inumctatras
  DISPLAY " " AT 6.65

AFTER FIELD t02inumctatras
  IF intentos < 2 THEN
    IF r_record.t02inumctatras IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos - 1
      NEXT FIELD tet02_traspasos.t02inumctatras
    ELSE
      CALL t13_d_ctabanr_record.t02ibancotras,
        r_record.t02isucurtras,r_record.t02inumctatras)
      RETURNING lt_ctat
      IF STATUS = NOTFOUND THEN
        ERROR "NO EXISTE CUENTA BANCARIA"
        NEXT FIELD tet02_traspasos.t02inumctatras
      ELSE
        IF r_record.t02inumcta = r_record.t02inumctatras THEN
          ERROR "NO PUEDEN REALIZARSE TRASPASOS SOBRE ",
            "LA MISMA CUENTA"
          NEXT FIELD tet02_traspasos.t02inumctat
        ELSE
          DISPLAY BY NAME lt_ctat
        END IF
      END IF
      LET intentos = 0
    END IF
  ELSE
    CALL initia()
    EXIT INPUT
  END IF

BEFORE FIELD t21imonedas
  SELECT c04imonedas INTO r_record.t21imonedas
  FROM tec04_ctab
  WHERE c04ibanco = r_record.t02ibanco
  AND c04isucur = r_record.t02isucur
  AND c04inumcta = r_record.t02inumcta
  DISPLAY BY NAME r_record.t21imonedas

AFTER FIELD t21imonedas
  IF intentos < 2 THEN
    IF r_record.t21imonedas IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos - 1
      NEXT FIELD tet21_movdoc.t21imonedas
    ELSE
      CALL t05_d_monedast_r_record.t21imonedas) RETURNING lt_moneda
      IF STATUS = NOTFOUND THEN
        ERROR "NO EXISTE MONEDA EN EL CATALOGO"
        NEXT FIELD tet21_movdoc.t21imonedas
      ELSE

```

```

      DISPLAY BY NAME lt_moneda
    END IF
    LET intentos = 0
  END IF
ELSE
  CALL iniciat )
  EXIT INPUT
END IF

AFTER FIELD t02_imptraspaso
IF intentos < 2 THEN
  IF r_record.t02_imptraspaso IS NULL OR
  r_record.t02_imptraspaso = 0 THEN
    ERROR "EL CAMPO NO PUEDE SER NULO NI MENOR A CERO!!!"
    LET intentos = intentos + 1
    NEXT FIELD t02_imptraspaso
  ELSE
    SELECT c04ietacon.c04imoneda.c04_saldomin
    INTO lt_cta1.lt_mon1.lt_salmin
    FROM tec04_ctab
    WHERE c04ibanco = r_record.t02ibanco
    AND c04isucur = r_record.t02isucur
    AND c04inumeta = r_record.t02inumeta

    SELECT c04ietacon INTO lt_cta2
    FROM tec04_ctab
    WHERE c04ibanco = r_record.t02ibanco
    AND c04isucur = r_record.t02isucur
    AND c04inumeta = r_record.t02inumetatras

    LET lt_mon1 = r_record.t21imoneda

  IF r_record.t21imoneda != "MN" THEN
    LET lt_impdet = 0
    SELECT c02_tipocam INTO lt_tipcam
    FROM tec02_tipoca
    WHERE c02imoneda = r_record.t21imoneda
    AND c02ifecha = r_record.t06_fecdocto
    IF STATUS = NOTFOUND THEN
      ERROR "LA PARIDAD DEL DIA PARA LA MONEDA ",
      r_record.t21imoneda,
      " NO SE HA DADO DE ALTA. VERIFIQUE!!!"
    SLEEP 4
    RETURN 2
  ELSE
    LET lt_impdet = r_record.t02_imptraspaso * lt_tipcam
  END IF
ELSE
  LET lt_impdet = r_record.t02_imptraspaso
END IF

SELECT t13_saldofin INTO lt_saldof
FROM tet13_saldos
WHERE t13ibanco = r_record.t02ibanco

```



```
AND t13isucur = r_record.t02isucur
AND t13inumeta = r_record.t02inumeta
AND t13ifecha = r_record.t06_fecdocto

LET lt_chsal = lt_saldo - lt_impdet
IF lt_chsal < lt_salmin THEN
  ERROR "NO SE DEBE SOBREPASAR EL SALDO MINIMO DE LA CUENTA."
  " QUE ES ",lt_salmin
  NEXT FIELD t02_tra-pasos t02_imptaspaso
END IF
LET intentos = 0
END IF
ELSE
  CALL initial()
  EXIT INPUT
END IF

AFTER FIELD t02tipotras
IF intentos < 2 THEN
  IF r_record.t02tipotras IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos - 1
    NEXT FIELD t02tipotras
  ELSE
    LET intentos = 0
  END IF
ELSE
  CALL initial()
  EXIT INPUT
END IF

AFTER FIELD t02_ban_che
IF intentos < 2 THEN
  IF r_record.t02_ban_che IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos - 1
    NEXT FIELD t02_ban_che
  ELSE
    LET intentos = 0
  END IF
ELSE
  CALL initial()
  EXIT INPUT
END IF

ON KEY (CONTROL-Z)
CASE
  WHEN INFIELD(t06iregion)
    CALL f3_region() RETURNING r_record.t06iregion.lt_region
  IF r_record.t06iregion IS NULL THEN
    ERROR "NO SELECCIONO REGION"
  ELSE
    DISPLAY BY NAME r_record.t06iregion.lt_region
  END IF
```

```
WHEN INFIELD(t02ibanco)
  CALL f1_banco() RETURNING r_record.t02ibanco.lt_bancof
  IF r_record.t02ibanco IS NULL THEN
    ERROR "NO SELECCIONO BANCO"
  ELSE
    DISPLAY BY NAME r_record.t02ibanco.lt_bancof
  END IF
WHEN INFIELD(t02isucur)
  CALL f2_sucursal(r_record.t02ibanco)
  RETURNING r_record.t02isucur.lt_sucurf
  IF r_record.t02isucur IS NULL THEN
    ERROR "NO SELECCIONO SUCURSAL"
  ELSE
    DISPLAY BY NAME r_record.t02isucur.lt_sucurf
  END IF
WHEN INFIELD(t02inumeta)
  CALL f4_cuentas(r_record.t02ibanco,r_record.t02isucur)
  RETURNING r_record.t02inumeta.lt_ctaf
  IF r_record.t02inumeta IS NULL THEN
    ERROR "NO SELECCIONO CUENTA FUENTE"
  ELSE
    DISPLAY BY NAME r_record.t02inumeta.lt_ctaf
  END IF
WHEN INFIELD(t02ibancotras)
  CALL f1_banco() RETURNING r_record.t02ibancotras.lt_bancot
  IF r_record.t02ibancotras IS NULL THEN
    ERROR "NO SELECCIONO BANCO TRASPASO"
  ELSE
    DISPLAY BY NAME r_record.t02ibancotras.lt_bancot
  END IF
WHEN INFIELD(t02isucurtras)
  CALL f2_sucursal(r_record.t02ibanco)
  RETURNING r_record.t02isucurtras.lt_sucurt
  IF r_record.t02isucurtras IS NULL THEN
    ERROR "NO SELECCIONO SUCURSAL TRASPASO"
  ELSE
    DISPLAY BY NAME r_record.t02isucurtras.lt_sucurt
  END IF
WHEN INFIELD(t02inumctatras)
  CALL f4_cuentas(r_record.t02ibancotras,r_record.t02isucurtras)
  RETURNING r_record.t02inumctatras.lt_ctat
  IF r_record.t02inumctatras IS NULL THEN
    ERROR "NO SELECCIONO CUENTA TRASPASO"
  ELSE
    DISPLAY BY NAME r_record.t02inumctatras.lt_ctat
  END IF
WHEN INFIELD(t21imoneda)
  CALL f8_moneda() RETURNING r_record.t21imoneda.lt_moneda
  IF r_record.t21imoneda IS NULL THEN
    ERROR "NO SELECCIONO MONEDA"
  ELSE
    DISPLAY BY NAME r_record.t21imoneda.lt_moneda
  END IF
END CASE
```

```

ON KEY (CONTROL-C)
  LET INT_FLAG = TRUE
  EXIT INPUT
END INPUT

IF INT_FLAG THEN
  LET INT_FLAG = FALSE
  ERROR "MODIFICACION ABANDONADA, ULTIMOS DATOS IGNORADOS"
  CALL f11_display(r_record.*)
  RETURN num_sct_num_reg.lt_movmax
END IF
BEGIN WORK
  CALL detalle(ht_impdet,"1",r_record.t02inum_docto,1,
    ht_cta1.lt_cta2,"T") RETURNING ht_status
  UPDATE tet21_movdoe
  SET t21_status = 0
  WHERE t21inum_docto = r_record.t02inum_docto
  AND t21imov = r_record.t02imov

  LET r_record.t02imov = r_record.t02imov + 1
  INSERT INTO tet21_movdoe VALUES(r_record.t02inum_docto,
    r_record.t02imov,3,gt_concepto,r_record.t21imoneda,
    r_record.t06_fecdocto,"T",4)
  INSERT INTO tet02_traspasos VALUES(r_record.t02inum_docto,
    r_record.t02imov,r_record.t02ibanco,r_record.t02isucur,
    r_record.t02inumcta,r_record.t02ibancotras,r_record.t02isucurtras,
    r_record.t02inumctatras,r_record.t02_imprtraspaso,
    r_record.t02itipotras,r_record.t02_ban_che)

  SELECT t13_saldofin INTO ht_saldof
  FROM tet13_saldos
  WHERE t13ibanco = r_record.t02ibancotras
  AND t13isucur = r_record.t02isucurtras
  AND t13inumcta = r_record.t02inumctatras
  AND t13ifecha = r_record.t06_fecdocto

  LET ht_saldof = ht_saldof - ht_impdet
  LET ht_saldot = ht_saldof + ht_impdet

  UPDATE tet13_saldos
  SET t13_saldofin = ht_saldof,
    t13_car_tra = t13_car_tra + ht_impdet
  WHERE t13ibanco = r_record.t02ibanco
  AND t13isucur = r_record.t02isucur
  AND t13inumcta = r_record.t02inumcta
  AND t13ifecha = r_record.t06_fecdocto
  UPDATE tet13_saldos
  SET t13_saldofin = ht_saldot,
    t13_abo_tra = t13_abo_tra + ht_impdet
  WHERE t13ibanco = r_record.t02ibancotras
  AND t13isucur = r_record.t02isucurtras
  AND t13inumcta = r_record.t02inumctatras
  AND t13ifecha = r_record.t06_fecdocto
  COMMIT WORK

```

```

IF STATUS <> 0 THEN
  ERROR "ERROR EN LOS DATOS, MODIFICACION ABANDONADA",
    " ULTIMOS DATOS IGNORADOS"
ELSE
  ERROR "MODIFICACION EFECTUADA"
END IF
ELSE
  ERROR "NO SE PUEDEN MODIFICAR DOCUMENTOS DE DIAS ANTERIORES"
END IF
RETURN num_act,num_reg,it_movmax
END FUNCTION

```

```

#####
# Limpia pantalla #
#####
FUNCTION f2_limpia()
  DISPLAY "" AT 1, 1
  DISPLAY "" AT 2, 1
END FUNCTION

```

```

FUNCTION f3_limpia()
  DISPLAY "" " AT 23,60
END FUNCTION

```

```

#####
# Despliega el numero de registros encontrados #
#####
FUNCTION f12_despli(num_act,num_reg)

```

```

  DEFINE
    num_act ,
    num_reg INTEGER
  DISPLAY "REGISTRO ",num_act," DE ",num_reg AT 23,60
  ATTRIBUTE(RED,REVERSE)
END FUNCTION

```

```

#####
# Encontrando el siguiente registro #
#####
FUNCTION f9_sigreg(num_act,num_reg,it_movmax)

```

```

  DEFINE
    it_movmax LIKE tet02_traspasos.t02imov.
    num_act ,
    num_reg ,
    num_rec INTEGER

```

```

CALL f3_limpia()
FETCH NEXT cur_consul INTO num_rec,it_movmax
OPEN cur_consulta USING num_rec,it_movmax
FETCH NEXT cur_consulta INTO r_sig.*

```

```

LET num_act = num_act + 1

```

```

IF STATUS = NOTFOUND THEN
  ERROR "ESTE ES EL ULTIMO REGISTRO DE LA LISTA"

```

```

FETCH CURRENT cur_consul INTO num_rec.lt_movmax
OPEN cur_consulta USING num_rec.lt_movmax

  FETCH LAST cur_consulta INTO r_sig.*

  LET num_act = num_reg
END IF

IF num_act > num_reg THEN
  LET num_act = num_reg
END IF

CALL ft2_despli(num_act,num_reg)
CALL ft1_display(r_sig.*)
LET gt_docto = r_sig.t02inum_docto
LET gt_mov = r_sig.t02imov

RETURN num_act,num_reg.lt_movmax
END FUNCTION

#####
# Enccontrando el registro previo #####
#####
FUNCTION f9_prepeg(num_act,num_reg,lt_movmax)
DEFINE
  resp CIAR,
  lt_movmax LIKE tet02_traspasos.t02imov,
  num_act,
  num_reg,
  num_rec INTEGER

CALL f3_limpia()
FETCH PRIOR cur_consul INTO num_rec.lt_movmax

IF num_rec = 0 THEN
  ERROR "ESTE ES EL PRIMER REGISTRO DE LA LISTA"
  FETCH CURRENT cur_consul INTO num_rec.lt_movmax
  OPEN cur_consulta USING num_rec.lt_movmax
  LET num_act = 1
  FETCH FIRST cur_consulta INTO r_pre.*
ELSE
  OPEN cur_consulta USING num_rec.lt_movmax
  LET num_act = num_rec - 1
  FETCH FIRST cur_consulta INTO r_pre.*
  IF num_act <= 0 THEN
    LET num_act = 1
  END IF
END IF
CALL ft1_display(r_pre.*)
CALL ft2_despli(num_act,num_reg)
LET gt_docto = r_pre.t02inum_docto
LET gt_mov = r_pre.t02imov
RETURN num_act,num_reg.lt_movmax
END FUNCTION

```

```

#####
# Cierra cursores
#####
FUNCTION f3_closcur()
  CLOSE cur_consul
  CLOSE cur_consulta
END FUNCTION

#####
# Inicializa variables
#####
FUNCTION inicia()
  ERROR "YA REALIZO MAS DE TRES INTENTOS INFRUCTUOSOS, TRATE DESPUES"
  SLEEP 2
  INITIALIZE r_record TO NULL
  LET int_flag = TRUE
END FUNCTION

#####
# Despliega informacion de los registros
#####
FUNCTION f11_display(r_record)
  DEFINE
    lt_region,
    lt_bancof,
    lt_bancot,
    lt_sucurf,
    lt_sucurt,
    lt_moneda,
    lt_ctaf,
    lt_ctat,
    lt_ipotras CHAR(30),
    r_record RECORD
    t02inum_docto LIKE t02_traspasos.t02inum_docto,
    t02tmov LIKE t02_traspasos.t02imov,
    t06_fecdocto LIKE t06_documento.t06_fecdocto,
    t06iregion LIKE t06_documento.t06iregion,
    t02ibanco LIKE t02_traspasos.t02ibanco,
    t02isucur LIKE t02_traspasos.t02isucur,
    t02inumcia LIKE t02_traspasos.t02inumcia,
    t02ibancostras LIKE t02_traspasos.t02ibancostras,
    t02isucurtras LIKE t02_traspasos.t02isucurtras,
    t02inumctatras LIKE t02_traspasos.t02inumctatras,
    t21moneda LIKE t021_movdoc.t21moneda,
    t02_imprtraspaso LIKE t02_traspasos.t02_imprtraspaso,
    t02tipotras LIKE t02_traspasos.t02tipotras,
    t02_ban_che LIKE t02_traspasos.t02_ban_che
  END RECORD

  LOCATE gt_concepto IN MEMORY
  SELECT t21_concepto INTO gt_concepto
  FROM t021_movdoc
  WHERE t21inum_docto = r_record.t02inum_docto
  AND t21imov = r_record.t02imov

```

```

IF (r_record.t02inum_docto <> 0) THEN
  DISPLAY
    r_record.t02inum_docto,
    r_record.t06_fecdocto,
    r_record.t06iregion,
    r_record.t02ibanco,
    r_record.t02isucur,
    r_record.t02inumcta,
    r_record.t02ibancotras,
    r_record.t02isucurtras,
    r_record.t02inumctatras,
    r_record.t21lmoneda,
    r_record.t02_imptraspaso,
    r_record.t02tipotras,
    r_record.t02_ban_che,
    gt_concepto
  TO
    tet02_traspasos.t02inum_docto,
    tet06_documento.t06_fecdocto,
    tet06_documento.t06iregion,
    tet02_traspasos.t02ibanco,
    tet02_traspasos.t02isucur,
    tet02_traspasos.t02inumcta,
    tet02_traspasos.t02ibancotras,
    tet02_traspasos.t02isucurtras,
    tet02_traspasos.t02inumctatras,
    tet21_movdoc.t21lmoneda,
    tet02_traspasos.t02_imptraspaso,
    tet02_traspasos.t02tipotras,
    tet02_traspasos.t02_ban_che,
    tet21_movdoc.t21_concepto
  END IF

  CALL t06_d_region(r_record.t06iregion) RETURNING lt_region
  CALL t01_d_banco(r_record.t02ibanco) RETURNING lt_bancof
  CALL t01_d_banco(r_record.t02ibancotras) RETURNING lt_bancof
  CALL t07_d_sucursalf(r_record.t02ibanco,r_record.t02isucur)
    RETURNING lt_sucurf
  CALL t07_d_sucursalf(r_record.t02ibancotras,r_record.t02isucurtras)
    RETURNING lt_sucurf
  CALL ft3_d_ctaban(r_record.t02ibanco,r_record.t02isucur,r_record.t02inumcta)
    RETURNING lt_ctaf
  CALL ft3_d_ctaban(r_record.t02ibancotras,r_record.t02isucurtras,
    r_record.t02inumctatras) RETURNING lt_ctaf
  CALL t05_d_monedas(r_record.t21lmoneda) RETURNING lt_moneda
  CALL ft7_d_tiptra(r_record.t02tipotras) RETURNING lt_tpotras

  DISPLAY BY NAME lt_region,lt_bancof,lt_bancof,lt_sucurf,lt_sucurf,
    lt_ctaf,lt_ctaf,lt_moneda,lt_tpotras
END FUNCTION

```

PANTALLAS DE REGISTRO DE CHEQUES

DATABASE sitesdb

SCREEN SIZE 24 BY 80

(

```

DOCUMENTO          [a3 ]   FECHA [a4 ]   REGION [b] [b1 ]
NUM. DE CHEQUE     [f000 ]
BANCO              [a0] [a0]
SUCURSAL          [f001] [f0001
NUM. CUENTA       [f002] [f0002
MONEDA            [a1] [a10
FECHA DE IMPRES.  [f003 ]
BENEFICIARIO      [f004 ]
                  [f005 ]
FECHA DE ENTREGA  [f006 ]
STATUS            [a]

```

END

TABLES

tet08_cheques tet06_documento

ATTRIBUTES

```

a3 = tet06_documento.t06inum_docto.reverse;
a4 = tet06_documento.t06_fecha.reverse; PICTURE = "###/###/####";
b = tet06_documento.t06region.reverse;
b1 = FORMONLY.t_region.reverse; noentry;
f000 = tet08_cheques.t08inumcheque.reverse;
      comments = "PROPORCIONA EL NUMERO DE DOCUMENTO";
a0 = tet08_cheques.t08ibanca.reverse;
      comments = "PROPORCIONA EL NUMERO DE BANCO";
f001 = FORMONLY.t_banco.reverse; noentry;
f001 = tet08_cheques.t08sucur.reverse;
      comments = "PROPORCIONA LA SUCURSAL";
f001 = FORMONLY.t_sucur.reverse; noentry;
f002 = tet08_cheques.t08numcta.reverse;
      comments = "PROPORCIONA LA CUENTA DEL BANCO";
f002 = FORMONLY.t_ctaban.reverse; noentry;
a1 = tet08_cheques.t08moneda.reverse;
      comments = "PROPORCIONA LA MONEDA ";
a10 = FORMONLY.t_moneda.reverse; noentry;
f003 = tet08_cheques.t08_fechaimp.reverse; PICTURE = "###/###/####";
      comments = "PROPORCIONA LA FECHA DE IMPRESION";
f004 = tet08_cheques.t08_beneficiario[1,40].reverse; UPSHIFT;
f005 = tet08_cheques.t08_beneficiario[41,80].reverse; UPSHIFT;
      comments = "PROPORCIONA EL BENEFICIARIO";
f006 = tet08_cheques.t08_fechaent.reverse; PICTURE = "###/###/####";
      comments = "PROPORCIONA LA FECHA DE ENTREGA";
a = tet08_cheques.t08_status.reverse;

```

END

INSTRUCTIONS

delimiters "

PROGRAMA DE REGISTRO DE CHEQUE

```

#                               No. 1
#
# Base de datos:                sitedb
# Sistema:                      SIITES
# Modulo:                       Generacion de Mantenimientos
# Pantalla:                     tepcheq per
# Prog. fuente:                 tefcheq.4gl
# Prog. objeto:                 tefcheq.4go
# Prog. ejecutable:            tefcheq.4gi
# Archivo de ayuda:            teacheq.msg
# Proceso:                      Impresion de Cheques
# Objetivo:                     Generacion e impresion de Cheques
#
# Analista:                     Felipe Cruz Chavez
# Programador:                  Felipe Cruz Chavez
# Fecha creacion:               11-13-1996
# Fecha ultima modificacion:    11-13-1996
# No. de version:               1
#

```

```

DATABASE sitedb

```

```

DEFINE

```

```

old_reg,
r_consul,
r_pre,
r_sig,
r_bajmod,
r_record

```

```

RECORD

```

```

i06inum_docto LIKE tet06_documento.i06inum_docto,
i06_fecdocto  LIKE tet06_documento.i06_fecdocto,
i06iregion    LIKE tet06_documento.i06iregion,
i08inumcheque LIKE tet08_cheques.i08inumcheque,
i08ibanco     LIKE tet08_cheques.i08ibanco,
i08isucur     LIKE tet08_cheques.i08isucur,
i08inumcta    LIKE tet08_cheques.i08inumcta,
i08imonedas   LIKE tet08_cheques.i08imonedas,
i08_fechaimp  LIKE tet08_cheques.i08_fechaimp,
i08_beneficiario LIKE tet08_cheques.i08_beneficiario,
i08_fecent    LIKE tet08_cheques.i08_fecent,
i08_status    LIKE tet08_cheques.i08_status

```

```

END RECORD

```

```

MAIN

```

```

DEFINE

```

```

pt_pasw      CHAR(18),
pt_itor      CHAR(78)

```

```

DEFER INTERRUPT

```

```

LET pt_pasw = ARG_VAL(1)

```

```

LET pt_itor = ARG_VAL(2)

```

```
CALL f0_menu(pt_pasw, pt_iton)
END MAIN
```

```
FUNCTION f0_menu(pt_nomb, pt_instr)
```

```
DEFINE
```

```
    num_act,      INTEGER,
    num_reg       CHAR,
    letra         SMALLINT,
    bandera       CHAR(18),
    pt_nomb       CHAR(78),
    pt_instr      SMALLINT,
    lt_bcon       CHAR(8),
    lt_npro       CHAR(80),
    lt_nttt
```

```
OPTIONS
```

```
INPUT ATTRIBUTE (RID,MAGENTA),
DISPLAY ATTRIBUTE (RID,MAGENTA),
HELP FILE "techeq.hlp"
```

```
LET lt_npro = "TEPCHEQ"
LET lt_nttt = "REGISTRO DE CHEQUES"
```

```
OPEN FORM TEPCHQEQ FROM "tepchq"
DISPLAY FORM TEPCHQEQ
CALL f1_linea(lt_npro, lt_nttt, 32)
```

```
MENU "MENU"
```

```
COMMAND "Alta" "Alta de un nuevo registro" HELP 11
```

```
IF lt_bcon THEN
    CALL f1_closeurl()
    LET lt_bcon = FALSE
END IF
```

```
LET letra = 'A'
```

```
IF bandera THEN
    CALL f5_ajust()
```

```
ELSE
    ERROR " USTED NO TIENE PERMISO PARA ACCESAR",
          " ESTA OPCION "
END IF
```

```
COMMAND "Bajas" "Elimina un registro." HELP 12
```

```
IF lt_bcon THEN
    LET letra = 'B'
```

```
IF bandera THEN
    CALL f6_baja(num_act, num_reg)
    RETURNING lt_bcon, num_act, num_reg
```

```
ELSE
    ERROR " USTED NO TIENE PERMISO PARA ACCESAR",
```

```
      " ESTA OPCION "
    END IF
  ELSE
    ERROR " PARA DAR DE BAJA UN REGISTRO PRIMERO",
      " DEBE CONSULTAR "
    NEXT OPTION "Consulta"
  END IF

COMMAND "Modificacion " "Actualiza un registro."  HELP 13

IF !t_bcon THEN
  LET letra = 'M'

  IF bandera THEN
    CALL f7_modif(num_act,num_reg)
    RETURNING num_act,num_reg
    CALL f3_limpiar()
    CALL f12_despl1(num_act,num_reg)
  ELSE
    ERROR " USTED NO TIENE PERMISO PARA ACCESAR",
      " ESTA OPCION "
  END IF
ELSE
  ERROR " PARA MODIFICAR UN REGISTRO PRIMERO",
    " DEBE CONSULTAR "
  NEXT OPTION "Consulta"
END IF

COMMAND "Consulta" "Consulta de registros"  HELP 14

IF !t_bcon THEN
  CALL f3_clocurt()
  LET !t_bcon = FALSE
END IF

CALL f8_consul()
RETURNING !t_bcon,num_act,num_reg

COMMAND "Siguiente" "Ver siguiente registro en la lista"  HELP 15

IF !t_bcon THEN
  CALL f9_sigreg(num_act,num_reg)
  RETURNING num_act,num_reg
ELSE
  ERROR " PARA VER SIGUIENTE REGISTRO PRIMERO",
    " DEBE CONSULTAR "
  NEXT OPTION "Consulta"
END IF

COMMAND "Previo" "Ver registro anterior en la lista"  HELP 16

IF !t_bcon THEN
  CALL f9_prereg(num_act,num_reg)
  RETURNING num_act,num_reg
```

```
ELSE
  ERROR " PARA VER REGISTRO ANTERIOR PRIMERO",
    " DEBE CONSULTAR "
  NEXT OPTION "Consulta "
END IF

COMMAND "Fin" "Fin de captura. Regresa a menu anterior"

IF It_bcon THEN
  CALL f3_closcur()
END IF
EXIT MENU
END MENU
END FUNCTION

FUNCTION f5_alta()
CALL I2_impis()
DISPLAY "DIGITE <ESC> PARA ACEPTAR REGISTRO" AT 1,1
ATTRIBUTE(RED,REVERSE)
DISPLAY "- CTRL-C- PARA ABORTAR OPERACION " AT 2,1
ATTRIBUTE(RED,REVERSE)

WHILE TRUE
CASE f5_regis()
  WHEN 0
    ERROR "ALTA EFECTUADA ."
    SLEEP 2
  WHEN 1
    ERROR "ERROR EN LA CAPTURA. VERIFIQUE SUS DATOS"
    SLEEP 2
  WHEN 2
    ERROR "ALTA ABANDONADA, ULTIMOS DATOS IGNORADOS"
    EXIT WHILE
END CASE
END WHILE
CLEAR FORM
END FUNCTION

FUNCTION f5_regis()
DEFINE
orden          CHAR(500),
intentos       SMALLINT,
old_reg.,
It_banco      CHAR(30),
It_sucur      CHAR(30),
It_ctaban     CHAR(30),
It_moneda     CHAR(30),
It_region     CHAR(20),
It_usuario    CHAR(10),
It_numdocto   INTEGER

CLEAR FORM
INPUT BY NAME r_record.*
```

```

BEFORE FIELD t08inumcheque
  IF r_record.t08inumcheque IS NULL THEN
    MESSAGE ""
  END IF

BEFORE FIELD t06_fecdocto
  LET r_record.t06_fecdocto = TODAY
  DISPLAY BY NAME r_record.t06_fecdocto

AFTER FIELD t06iregion
  IF intentos < 2 THEN
    IF r_record.t06iregion IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos + 1
      NEXT FIELD tet06_documento.t06iregion
    ELSE
      CALL t06_d_region(r_record.t06iregion) RETURNING lt_region
      IF STATUS = NOTFOUND THEN
        ERROR "ESTA REGION NO EXISTE"
      NEXT FIELD t06iregion
    ELSE
      DISPLAY BY NAME lt_region
      END IF
      LET intentos = 0
    END IF
  ELSE
    CALL inicia()
    EXIT INPUT
  END IF

AFTER FIELD t08inumcheque
  IF intentos < 2 THEN
    IF r_record.t08inumcheque IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos + 1
      NEXT FIELD tet08_cheques.t08inumcheque
    ELSE
      LET intentos = 0
      END IF
  ELSE
    CALL inicia()
    EXIT INPUT
  END IF

BEFORE FIELD t08ibanco
  DISPLAY "[ZOOM]" AT 6.65 ATTRIBUTES(REVERSE,BLINK)

AFTER FIELD t08ibanco
  IF intentos < 2 THEN
    IF r_record.t08ibanco IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos + 1
      NEXT FIELD tet08_cheques.t08ibanco
    ELSE

```

```
CALL t01_d_banco(r_record.t08ibanco) RETURNING lt_bancof
IF STATUS = NOTFOUND THEN
  ERROR "ESTE BANCO NO EXISTE"
  NEXT FIELD t08ibanco
ELSE
  DISPLAY BY NAME lt_banco
END IF
LET intentos = 0
END IF
ELSE
  CALL initia()
  EXIT INPUT
END IF

BEFORE FIELD t08isucur
  DISPLAY "[ZOOM]" AT 6.65 ATTRIBUTE(REVERSE,BLINK)

AFTER FIELD t08isucur
  IF intentos < 2 THEN
    IF r_record.t08isucur IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos + 1
      NEXT FIELD tet08_cheques.t08isucur
    ELSE
      CALL t07_d_sucursal(r_record.t08ibanco,r_record.t08isucur)
      RETURNING lt_sucur
      IF STATUS = NOTFOUND THEN
        ERROR "ESTA SUCURSAL DEL BANCO NO EXISTE"
        NEXT FIELD t08isucur
      ELSE
        DISPLAY BY NAME lt_sucur
      END IF
      LET intentos = 0
    END IF
  ELSE
    CALL initia()
    EXIT INPUT
  END IF

BEFORE FIELD t08inumeta
  DISPLAY "[ZOOM]" AT 6.65 ATTRIBUTES(REVERSE,BLINK)

AFTER FIELD t08inumeta
  IF intentos < 2 THEN
    IF r_record.t08inumeta IS NULL THEN
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"
      LET intentos = intentos + 1
      NEXT FIELD tet08_cheques.t08inumeta
    ELSE
      CALL despliega() returning lt_ctaban
      SELECT c04_descrip INTO lt_ctaban FROM tec04_ctab
      WHERE c04ibanco = r_record.t08ibanco
      AND c04isucur = r_record.t08isucur
      AND c04inumeta = r_record.t08inumeta
    END IF
  ELSE
    CALL initia()
    EXIT INPUT
  END IF
```

```

IF STATUS = NOTFOUND THEN
  ERROR "ESTA CUENTA NO EXISTE EN ESTE BANCO"
  NEXT FIELD t08inumcta
ELSE
  DISPLAY BY NAME It_ctaban
END IF
LET intentos = 0
END IF
ELSE
  CALL inicia()
  EXIT INPUT
END IF

BEFORE FIELD t08imoneda
DISPLAY ["ZOOM]" AT 6.65 ATTRIBUTES(REVERSE,BLINK)

AFTER FIELD t08imoneda
IF intentos < 2 THEN
  IF r_record.t08imoneda IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos + 1
    NEXT FIELD tet08_cheques.t08imoneda
  ELSE
    CALL desplegar() returning It_moneda
    SELECT c01_moneda INTO It_moneda FROM tec01_moneda
    WHERE c01_moneda = r_record.t08imoneda
    IF STATUS = NOTFOUND THEN
      ERROR "ESTA MONEDA NO EXISTE EN EL CATALOGO"
      NEXT FIELD t08imoneda
    ELSE
      DISPLAY BY NAME It_moneda
    END IF
  END IF
  LET intentos = 0
END IF
ELSE
  CALL inicia()
  EXIT INPUT
END IF

AFTER FIELD t08_fechaimp
IF intentos < 2 THEN
  IF r_record.t08_fechaimp IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos + 1
    NEXT FIELD tet08_cheques.t08_fechaimp
  ELSE
    LET intentos = 0
  END IF
ELSE
  CALL inicia()
  EXIT INPUT
END IF

AFTER FIELD t08_beneficiario
IF intentos < 2 THEN

```

```
IF r_record.t08_beneficiario IS NULL THEN
  ERROR "EL CAMPO NO PUEDE SER NULO!!!"
  LET intentos = intentos - 1
  NEXT FIELD tet08_cheques.t08_beneficiario
ELSE
  LET intentos = 0
END IF
ELSE
  CALL initial()
  EXIT INPUT
END IF

AFTER FIELD t08_fecent
IF intentos < 2 THEN
  IF r_record.t08_fecent IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos - 1
    NEXT FIELD tet08_cheques.t08_fecent
  ELSE
    LET intentos = 0
  END IF
ELSE
  CALL initial()
  EXIT INPUT
END IF

AFTER FIELD t08_status
IF intentos < 2 THEN
  IF r_record.t08_status IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos - 1
    NEXT FIELD tet08_cheques.t08_status
  ELSE
    LET intentos = 0
  END IF
ELSE
  CALL initial()
  EXIT INPUT
END IF

ON KEY (CONTROL-/)
CASE
  WHEN INFIELD(t08ibanco)
    CALL f1_banco() RETURNING r_record.t08ibanco,lt_banco
  IF r_record.t08ibanco IS NULL THEN
    ERROR "NO SELECCIONO OPCION"
    NEXT FIELD t08ibanco
  ELSE
    DISPLAY BY NAME r_record.t08ibanco,lt_banco
  END IF
  WHEN INFIELD(t08isucur)
    CALL f2_sucursal(r_record.t08ibanco)
    RETURNING r_record.t08isucur,lt_sucur
```



```
        ERROR "NO SELECCIONO OPCION"
        NEXT FIELD t08isucur
    ELSE
        DISPLAY BY NAME r_record.t08isucur.lt_sucur
    END IF
    WHEN INFIELD(t08inumcta)
        CALL f4_cuentas(r_record.t08ibanco,r_record.t08isucur)
        RETURNING r_record.t08inumcta.lt_ctaban
        IF r_record.t08inumcta IS NULL THEN
            ERROR "NO SELECCIONO OPCION"
            NEXT FIELD t08inumcta
        ELSE
            DISPLAY BY NAME r_record.t08inumcta.lt_ctaban
        END IF
    WHEN INFIELD(t06iregion)
        CALL f3_region() RETURNING r_record.t06iregion.lt_region
        IF r_record.t06iregion IS NULL THEN
            ERROR "NO SELECCIONO OPCION"
            NEXT FIELD t06iregion
        ELSE
            DISPLAY BY NAME r_record.t06iregion.lt_region
        END IF
    WHEN INFIELD(t08_moneda)
        CALL f8_moneda() RETURNING r_record.t08imoneda.lt_moneda
        IF r_record.t08imoneda IS NULL THEN
            ERROR "NO SELECCIONO OPCION"
            NEXT FIELD t08imoneda
        ELSE
            DISPLAY BY NAME r_record.t08imoneda.lt_moneda
        END IF
    END CASE

ON KEY (CONTROL-W,f6)
CASE
    WHEN INFIELD(t08inumcheque)
        CALL SHOWHELP( 10)
    WHEN INFIELD(t08ibanco)
        CALL SHOWHELP( 20)
    WHEN INFIELD(t08isucur)
        CALL SHOWHELP( 30)
    WHEN INFIELD(t08inumcta)
        CALL SHOWHELP( 40)
    WHEN INFIELD(t08imoneda)
        CALL SHOWHELP( 50)
    WHEN INFIELD(t08_fechaimp)
        CALL SHOWHELP( 60)
    WHEN INFIELD(t08_beneficiario)
        CALL SHOWHELP( 70)
    WHEN INFIELD(t08_fecent)
        CALL SHOWHELP( 80)
    WHEN INFIELD(t08_status)
        CALL SHOWHELP( 90)
    END CASE
```

```
ON KEY (CONTROL-C)
  LET INT_FLAG = TRUE
  EXIT INPUT
END INPUT

IF INT_FLAG THEN
  LET INT_FLAG = FALSE
  DISPLAY " " A1 6.69
  RETURN 2
END IF

LET lt_usuario = "siles"

INSERT INTO tet06_documento
VALUES (0,lt_usuario,r_record.t06iregion,r_record.t06_fecdocto,"C","P")

LET lt_numdocto = SQLCA.SQLERRD[2]
DISPLAY lt_numdocto TO t06inum_docto
SLEEP 2

INSERT INTO tet08_cheques
VALUES (r_record.t08numcheque,r_record.t08ibanco,
       r_record.t08isucur,r_record.t08inumeta,
       r_record.t08imoneda, r_record.t08_fechaiamp,
       r_record.t08_beneficiario, r_record.t08_fecent,
       r_record.t08_status)

INSERT INTO tet14_chequeo
VALUES (r_record.t08numcheque,r_record.t08ibanco,
       r_record.t08isucur,r_record.t08inumeta,lt_numdocto)

LET old_reg.* = r_record.*

SELECT * FROM tet08_cheques
WHERE t08numcheque = old_reg.t08numcheque
  AND t08ibanco = old_reg.t08ibanco
  AND t08isucur = old_reg.t08isucur
  AND t08inumeta = old_reg.t08inumeta
  AND t08imoneda = old_reg.t08imoneda
  AND t08_fechaiamp = old_reg.t08_fechaiamp
  AND t08_beneficiario = old_reg.t08_beneficiario
  AND t08_fecent = old_reg.t08_fecent
  AND t08_status = old_reg.t08_status

IF STATUS = NOTFOUND THEN
  INSERT INTO tet08_cheques VALUES (r_record.*)
  IF STATUS <> 0 THEN
    RETURN 1
  ELSE
    RETURN 0
  END IF
ELSE
  ERROR "ESTE REGISTRO YA EXISTE"
  SLEEP 2

```

```

RETURN 2
END IF
END FUNCTION

=====
# Búsqueda de registros #
=====
FUNCTION f8_consul()
  DEFINE
    num_act,
    num_reg,
    num_rec,
    r_consul,
    txt_cons,
    con_cons,
    lt_codig
    INTEGER,
    RECORD LIKE tet08_cheques.,
    CHAR(1000),
    SMALLINT

  CALL f2_limpia()

  MESSAGE " INDIQUE LAS CONDICIONES DE CONSULTA Y PRESIONE",
    " <ESC>"
  ATTRIBUTE(RED,REVERSE)
  CLEAR FORM

  CONSTRUCT BY NAME con_cons ON
    t06inum_docto,
    t06_fecdocto,
    t06iregion,
    t08inumcheque,
    t08ibanco,
    t08isucur,
    t08inumcta,
    t08imoneda,
    t08_fechaimp,
    t08_beneficiario,
    t08_tocent,
    t08_status
  END CONSTRUCT

  IF INT_FLAG THEN
    LET num_act = 0
    LET num_reg = 0
    LET INT_FLAG = FALSE
    ERROR " CONSULTA ABANADONADA "
    CLEAR FORM
    RETURN lt_codig,num_act,num_reg
  END IF

  LET txt_cons = "SELECT ROWID ",
    "FROM tet08_cheques ",
    "WHERE ".con_cons CLIPPED

  PREPARE set_cons FROM txt_cons
  DECLARE cur_consul SCROLL CURSOR FOR set_cons

```

```

LET num_reg = 0

FOREACH cur_consul
  LET num_reg = num_reg + 1
END FOREACH

LET num_act = 1
OPEN cur_consul
FETCH FIRST cur_consul INTO num_reg
IF STATUS = NOTFOUND THEN
  LET num_act = 0
  LET num_reg = 0
  ERROR "NINGUN REGISTRO CUMPLE LAS CONDICIONES, VERIFIQUE "
  CLOSE cur_consul
  CLEAR FORM
  RETURN lt_codig,num_act,num_reg
ELSE
  LET txt_cons = "SELECT t06inum_docto,t06_fecdocto,t06iregion,",
    "t08inumcheque,t08ibanco,t08isucur,t08inumeta,",
    "t08imonedat,t08_fechainp,t08_beneficiario,",
    "t08_fecent,t08_status",
    "FROM ter09_documento.tet14_cheque.tet08_cheques ",
    "WHERE t06inum_docto = t14inum_docto ",
    "AND t08inumcheque = t14inumcheque ",
    "AND t08ibanco = t14ibanco ",
    "AND t08isucur = t14isucur ",
    "AND t08inumeta = t14inumeta ",
    "ORDER BY ",
    "t08inumcheque."

  PREPARE set_consulta FROM txt_cons
  DECLARE cur_consulta SCROLL CURSOR FOR set_consulta
  END IF

  CALL t12_despl1(num_act,num_reg)
  DISPLAY "REGISTRO ",num_act," DE ",num_reg AT 23,60
  ATTRIBUTE:RED,REVERSE)

OPEN cur_consulta
FETCH FIRST cur_consulta INTO r_consul.*

LET lt_codig = TRUE
CALL t11_display(r_consul.*)

MENU "Consulta"
COMMAND "Siguiente" "Ver siguiente registro en la lista de consulta"
  HELP 15
  CALL f9_sigreg(num_act,num_reg)
  RETURNING num_act,num_reg

COMMAND "Previo" "Ver registro previo en la lista de consulta" HELP 16
  CALL f9_prereg(num_act,num_reg)
  RETURNING num_act,num_reg

```

```

COMMAND "Inicial" "Ver registro inicial en la lista de consulta"
HELP 17
FETCH FIRST cur_consul INTO num_rec
OPEN cur_consulta
FETCH FIRST cur_consulta INTO r_consul.*
LET num_act = 1
CALL f3_limpiar()
DISPLAY "REGISTRO ", num_act, " DE ", num_reg AT 23,60
ATTRIBUTE(RED,REVERSE)
CALL f11_display(r_consul.*)
NEXT OPTION "Siguiente"

COMMAND "Ultimo" "Ver ultimo registro en la lista de consulta" HELP 18
FETCH LAST cur_consul INTO num_rec
OPEN cur_consulta #####USING num_rec
FETCH LAST cur_consulta INTO r_consul.*
LET num_act = num_reg
CALL f3_limpiar()
CALL f12_desplz(num_act,num_reg)
CALL f11_display(r_consul.*)
NEXT OPTION "Previo"

COMMAND "Fin" "Salir de menu de consulta"
EXIT MENU
END MENU
RETURN lt_codig,num_act,num_reg
END FUNCTION

```

```

#####
# Baja de registros
#####
FUNCTION f6_baja(num_act,num_reg)
DEFINE
    lt_bcon          SMALLINT,
    num_act,         INTEGER,
    num_reg,         INTEGER,
    old_reg,         RECORD LIKE tet08_cheques.*,
    r_bajmod         CHAR(1)

FETCH CURRENT cur_consul INTO num_rec
OPEN cur_consulta USING num_rec
FETCH FIRST cur_consulta INTO r_bajmod.*
CALL f11_display(r_bajmod.*)
CALL f2_limpiar()

OPEN WINDOW edwbaja AT 13,15 WITH 5 ROWS,40 COLUMNS
ATTRIBUTE(BORDER,MENU LINE LAST - 1)
DISPLAY " DESEA DAR DE BAJA ESTE REGISTRO " AT 2,4
ATTRIBUTE(RED)
MENU ""
BEFORE MENU

```

```

    DISPLAY " " AT 3,3
    COMMAND KEY ("N") " No "
    LET lt_resp = "N"
    EXIT MENU

    COMMAND KEY ("S") " Si "
    LET lt_resp = "S"
    EXIT MENU
END MENU
CLOSE WINDOW sdwbaja

IF lt_resp = "S" THEN
    LET old_reg.* = r_bajmod.*
    LET num_reg = num_reg - 1

    DELETE FROM tet08_cheques
    WHERE ROWID = num_rec

    FETCH NEXT cur_consul INTO num_rec
    IF STATUS = NOTFOUND THEN
        FETCH PRIOR cur_consul INTO num_rec
        IF STATUS = NOTFOUND THEN
            LET num_reg = 0
        ELSE
            LET num_act = num_act - 1
            OPEN cur_consulta USING num_rec
            FETCH FIRST cur_consulta INTO r_bajmod.*
            CALL fl1_display(r_bajmod.*)
        END IF
    ELSE
        LET num_act = num_act
        OPEN cur_consulta USING num_rec
        FETCH FIRST cur_consulta INTO r_bajmod.*
        CALL fl1_display(r_bajmod.*)
    END IF
    ERROR "REGISTRO ELIMINADO"
ELSE
    ERROR "BAJA ABANDONADA"
END IF

IF num_reg = 0 THEN
    LET lt_bcon = FALSE
    CLOSE cur_consulta
    CLOSE cur_consul
    CLEAR FORM
    DISPLAY " " AT 23,60
ELSE
    LET lt_bcon = TRUE
    CALL fl_limpia()
    CALL fl2_despli(num_act,num_reg)
END IF
SLEEP 2
RETURN lt_bcon,num_act,num_reg
END FUNCTION

```

```
#####  
# Modificación de registros  
#####  
FUNCTION i7_modif(num_act,num_reg)
```

```
  DEFINE  
    num_reg,  
    num_act,  
    num_rec          INTEGER,  
    intentos        SMALLINT,  
    orden           CHAR(500)  
    old_reg,  
    r_record        RECORD LIKE tet08_cheques.*
```

```
  FETCH CURRENT cur_consul INTO num_rec
```

```
  OPEN cur_consulta USING num_rec  
  FETCH FIRST cur_consulta INTO r_record.*  
  CALL f1_display(r_record.*)
```

```
  LET intentos = 0  
  LET old_reg.* = r_record.*
```

```
  INPUT BY NAME r_record.*
```

```
  AFTER FIELD i08inumcheque  
  IF intentos < 2 THEN  
    IF r_record.i08inumcheque IS NULL THEN  
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"  
      LET intentos = intentos + 1  
      NEXT FIELD tet08_cheques.i08inumcheque  
    ELSE  
      LET intentos = 0  
    END IF  
  ELSE  
    CALL inicia()  
    EXIT INPUT  
  END IF
```

```
  AFTER FIELD i08ibanco  
  IF intentos < 2 THEN  
    IF r_record.i08ibanco IS NULL THEN  
      ERROR "EL CAMPO NO PUEDE SER NULO!!!"  
      LET intentos = intentos + 1  
      NEXT FIELD tet08_cheques.i08ibanco  
    ELSE  
      LET intentos = 0  
    END IF  
  ELSE  
    CALL inicia()  
    EXIT INPUT  
  END IF
```

```
  AFTER FIELD i08isucur  
  IF intentos < 2 THEN
```

```
IF r_record.t08isucur IS NULL THEN
  ERROR "EL CAMPO NO PUEDE SER NULO!!!"
  LET intentos = intentos + 1
  NEXT FIELD tet08_cheques.t08isucur
ELSE
  LET intentos = 0
END IF
ELSE
  CALL initia()
  EXIT INPUT
END IF

AFTER FIELD t08inumeta
IF intentos < 2 THEN
  IF r_record.t08inumeta IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos + 1
    NEXT FIELD tet08_cheques.t08inumeta
  ELSE
    LET intentos = 0
  END IF
ELSE
  CALL initia()
  EXIT INPUT
END IF

AFTER FIELD t08imoneda
IF intentos < 2 THEN
  IF r_record.t08imoneda IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos + 1
    NEXT FIELD tet08_cheques.t08imoneda
  ELSE
    LET intentos = 0
  END IF
ELSE
  CALL initia()
  EXIT INPUT
END IF

AFTER FIELD t08_fechaimp
IF intentos < 2 THEN
  IF r_record.t08_fechaimp IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos + 1
    NEXT FIELD tet08_cheques.t08_fechaimp
  ELSE
    LET intentos = 0
  END IF
ELSE
  CALL initia()
  EXIT INPUT
END IF
```



```
AFTER FIELD t08_beneficiario
IF intentos < 2 THEN
  IF r_record.t08_beneficiario IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos + 1
    NEXT FIELD tet08_cheques.t08_beneficiario
  ELSE
    LET intentos = 0
  END IF
ELSE
  CALL initia()
  EXIT INPUT
END IF

AFTER FIELD t08_fecent
IF intentos < 2 THEN
  IF r_record.t08_fecent IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos + 1
    NEXT FIELD tet08_cheques.t08_fecent
  ELSE
    LET intentos = 0
  END IF
ELSE
  CALL initia()
  EXIT INPUT
END IF

AFTER FIELD t08_status
IF intentos < 2 THEN
  IF r_record.t08_status IS NULL THEN
    ERROR "EL CAMPO NO PUEDE SER NULO!!!"
    LET intentos = intentos + 1
    NEXT FIELD tet08_cheques.t08_status
  ELSE
    LET intentos = 0
  END IF
ELSE
  CALL initia()
  EXIT INPUT
END IF

ON KEY (CONTROL-W.(6))
CASE
WHEN INFIELD(t08inumcheque)
  CALL SHOWHELP( 10)
WHEN INFIELD(t08ibanco)
  CALL SHOWHELP( 20)
WHEN INFIELD(t08isucur)
  CALL SHOWHELP( 30)
WHEN INFIELD(t08inumcfa)
  CALL SHOWHELP( 40)
WHEN INFIELD(t08imoneda)
  CALL SHOWHELP( 50)
```

```

WHEN INFIELD(t08_fechinip)
  CALL SHOWHELP( 60)
WHEN INFIELD(t08_beneficiario)
  CALL SHOWHELP( 70)
WHEN INFIELD(t08_fecent)
  CALL SHOWHELP( 80)
WHEN INFIELD(t08_status)
  CALL SHOWHELP( 90)
END CASE
ON KEY (CONTROL-C)
  LET INT_FLAG = TRUE
  EXIT INPUT
END INPUT

IF INT_FLAG THEN
  LET INT_FLAG = FALSE
  ERROR "MODIFICACION ABANDONADA. ULTIMOS DATOS IGNORADOS"
  LET r_record.* = old_reg.*
  CALL f11_display(r_record.*)
  RETURN num_act,num_reg
END IF

```

```

UPDATE t08_cheques
SET t08_cheques.* = r_record.*
WHERE ROWID = num_rec

```

```

IF STATUS <> 0 THEN
  ERROR "ERROR EN LOS DATOS, MODIFICACION ABANDONADA."
  " ULTIMOS DATOS IGNORADOS"
ELSE
  ERROR "MODIFICACION EFECTUADA"
END IF
RETURN num_act,num_reg
END FUNCTION

```

```

#####
# Limpia pantalla
#####

```

```

FUNCTION f2_limpia()
  DISPLAY "" AT 1, 1
  DISPLAY "" AT 2, 1
END FUNCTION

```

```

FUNCTION f3_limpia()
  DISPLAY " " AT 23,60
END FUNCTION

```

```

#####
# Despliega el numero de registros encontrados
#####

```

```

FUNCTION f12_despli(num_act,num_reg)
DEFINE
  num_act,
  num_reg  INTEGER

```

```

DISPLAY "REGISTRO ",num_act," DE ",num_reg AT 23.60
ATTRIBUTE(RED,REVERSE)
END FUNCTION

```

```

#####
# Encontrando el siguiente registro #
#####
FUNCTION f9_sigreg(num_act,num_reg)
DEFINE
    num_act,
    num_reg,
    num_rec      INTEGER
    r_sig        RECORD LIKE tet08_cheques.*

```

```

CALL f3_limpia()
FETCH NEXT cur_consul INTO num_rec
OPEN cur_consulta USING num_rec
FETCH NEXT cur_consulta INTO r_sig.*
LET num_act = num_act + 1

IF STATUS = NOTFOUND THEN
    ERROR "ESTE ES EL ULTIMO REGISTRO DE LA LISTA"
    FETCH CURRENT cur_consul INTO num_rec
    OPEN cur_consulta USING num_rec
    FETCH LAST cur_consulta INTO r_sig.*
    LET num_act = num_reg
END IF

IF num_act > num_reg THEN
    LET num_act = num_reg
END IF

CALL f12_despli(num_act,num_reg)
CALL f11_display(r_sig.*)
RETURN num_act,num_reg
END FUNCTION

```

```

#####
# Encontrando el registro previo #
#####
FUNCTION f9_prereg(num_act,num_reg)
DEFINE
    resp      CHAR,
    num_act ,
    num_reg,
    num_rec   INTEGER
    r_pre     RECORD LIKE tet08_cheques.*

```

```

CALL f3_limpia()
FETCH PRIOR cur_consul INTO num_rec
IF num_rec = 0 THEN
    ERROR "ESTE ES EL PRIMER REGISTRO DE LA LISTA"
    FETCH CURRENT cur_consul INTO num_rec
    OPEN cur_consulta USING num_rec
    LET num_act = 1
    FETCH FIRST cur_consulta INTO r_pre.*

```

```

ELSE
  OPEN cur_consulta USING num_rec

  LET num_act = num_act - 1

  FETCH FIRST cur_consulta INTO r_pre.*

  IF num_act <<= 0 THEN
    LET num_act = 1
  END IF
END IF

CALL f11_display(r_pre.*)
CALL f12_desplnum_act.num_reg

RETURN num_act.num_reg
END FUNCTION

#####
# Cierra cursores
#####
FUNCTION f3_closcur()
  CLOSE cur_consulta
END FUNCTION

#####
# Inicializa variables
#####
FUNCTION inicia()
  DEFINE
    r_record RECORD LIKE tet08_cheques.*

  ERROR "YA REALIZO MAS DE TRES INTENTOS INFRUCTUOSOS.",
    "TRATE DESPUES"

  SLEEP 2
  INITIALIZE r_record TO NULL
  LET in_flag = TRUE
END FUNCTION

#####
# Despliega informacion de los registros
#####
FUNCTION f11_display(r_record)
  DEFINE
    r_record RECORD
    r06num_docto LIKE tet06_documento.t06num_docto,
    r06_fecdocto LIKE tet06_documento.t06_fecdocto,
    r06iregion LIKE tet06_documento.t06iregion,
    r08inumcheque LIKE tet08_cheques.t08inumcheque,
    r08ibanco LIKE tet08_cheques.t08ibanco,
    r08isucur LIKE tet08_cheques.t08isucur,
    r08inumcta LIKE tet08_cheques.t08inumcta,
    r08imonedas LIKE tet08_cheques.t08imonedas,

```

```
i08_fechaimp      LIKE tet08_cheques.i08_fechaimp.  
i08_beneficiario LIKE tet08_cheques.i08_beneficiario.  
i08_fecent       LIKE tet08_cheques.i08_fecent,  
i08_status       LIKE tet08_cheques.i08_status
```

END RECORD.

```
lt_banco         CHAR(30),  
lt_sucur        CHAR(30),  
lt_ctaban       CHAR(30),  
lt_moneda       CHAR(30),  
lt_region       CHAR(20),  
lt_usuario      CHAR(10),  
lt_numdocto     INTEGER
```

INITIALIZE lt_banco TO NULL

```
let lt_banco = lt_sucur  
let lt_banco = lt_ctaban  
let lt_banco = lt_region  
let lt_banco = lt_moneda  
let lt_banco = lt_usuario
```

```
CALL f06_d_region(r_record.i06iregion) RETURNING lt_region  
CALL f01_d_banco(r_record.i08ibanco) RETURNING lt_banco  
CALL f07_d_sucursall(r_record.i08ibanco,r_record.i08isucur) RETURNING lt_sucur  
CALL f13_d_ctaban(r_record.i08ibanco,r_record.i08isucur,r_record.i08inumcta) RETURNING lt_ctaban  
CALL f05_d_monedas(r_record.i08imoneda) RETURNING lt_moneda
```

DISPLAY BY NAME lt_region, lt_banco, lt_sucur, lt_ctaban, lt_moneda

END FUNCTION



APÉNDICE C

APÉNDICE C

MUESTRAS DE PRODUCTOS

PÓLIZA DE CAJA

FECHA : 01-May-1997

No. DE PAGINA : 1

FERROCARRILES NACIONALES DE MEXICO
 SUBDIRECCION GENERAL DE FINANZAS
 S I I T E S
 POLIZA DE CAJA TG-23 DE 03-Feb-1997
 C A R G O S

	IMPORTE SSCTA.	IMPORTE SCTA	IMPORTE CTA.	TOTAL
531 GASTOS DE OPERACION				233,798.55
1 CONSERVACION DE VIA Y ESTRUCTURAS			497.00	
201 SUPERINTENDENCIA		497.00		
7 GASTOS GENERALES			233,301.55	
471 SUELDOS Y GASTOS DE FUNCIONARIOS GRALES	14,839.00			
472 SUELDOS Y GASTOS DE EMPLEADOS Y SEVIDUMB	35,508.00			
478 GASTOS DIVERSOS	7,500.00			
488 CONCESIONES DIVERSAS AL PERSONAL	175,454.55			
701 CAJA Y BANCOS				256,114,821.88
20 C A J A			21,807.11	
4 EFECTIVO Y CHEQUES POR DEVOLUCION	12,890.66			
5 NUMERARIO NO CLASIFICADO	8,916.45			
62 030318-8 BANAMEX, S.N.C. CTA. M.N. REMES		5,429,016.57		
66 0820278-8 COMERMEX,SNC,M.N. REMESAS		6,335,447.21		
70 0390838-0 V.COBRANZA RECIBIDA EN MULTIBA		5,176,549.75		
71 1008367-3 BANCO INTERNACIONAL,SNC,CTA.M.		393,460.87		
74 BANAMEX, INGRESOS DIVISION DEL PACIFICO		776,804.96		
75 COMERMEX, INGRESOS DIVISION CHIHUAHUA-PA		104,986.27		
79 0876950-7 BANCOMER, SNC. M.N. CUENTA GEN	149,857,642.64			
92 10-08450-7 BANCO INTERNACIONAL, SNC PROV		364,554.52		
101 COMERMEX CUENTA ESPECIAL NO.550597-6		87,654,551.98		
702 INVERSIONES TEMPORALES			231,000,000.00	
16 VALORES EN BANCO INVERLAT		87,800,000.00		
42 CUENTA DE MULTIVALORES, S.A.		43,200,000.00		
45 ABACO CASA DE BOLSA, S.A.		100,000,000.00		
710 FONDOS DE TRABAJO			16,140.88	
607 TG-40 MINISTRACION INMEDIATA DE FONDOS		16,140.88		
713 OTRO ACTIVO CIRCULANTE			48,500.00	
345 ADEUDOS EMPLEADOS ANTICIPO SUELDO, E G R		20,500.00		
416 FUNCIONARIOS Y EMPLEADOS		28,000.00		
748 CARGOS DIVERSOS PENDIENTES DE AJUSTE			4.03	
312 DEPOSITO EN TRANSITO			4.03	

FECHA : 01-May-1997

No. DE PAGINA : 2

FERROCARRILES NACIONALES DE MEXICO
 SUBDIRECCION GENERAL DE FINANZAS
 S I I T E S
 POLIZA DE CAJA TG-23 DE 03-Feb-1997
 C A R G O S

		IMPORTE IMPORTE IMPORTE			TOTAL
		SSCTA.	SCTA.	CTA.	
753	CUENTAS POR PAGAR, GLOSADAS				186,693.18
	1 COMPROBANTES				186,693.18
755	SUELDOS POR PAGAR				787,498.94
	2 BOLETAS DE TIEMPO				59,200.00
	5 LISTAS DE RAYA				64,145.71
	42 SEGUROS DE MEXICO				381,260.71
	62 FINIQUITOS POR PAGAR				20,067.98
	65 LIQ. PROGRAMA RACIONALIZACION POR PAGAR				262,824.54
788	CREDITOS DIVERSOS PENDIENTES DE AJUSTE				4,314,581.03
	18 ENTEROS EN SUSPENSO				4,314,581.03

\$					492,702,038.49

APÉNDICE C

MUESTRAS DE PRODUCTOS

FECHA : 01-May-1997

FERROCARRILES NACIONALES DE MEXICO
SUBDIRECCION GENERAL DE FINANZAS
S I I T E S
POLIZA DE CAJA TG-23 DE 03-Feb-1997
A B O N O S

No. DE PAGINA : 3

	IMPORTE SSCTA.	IMPORTE SCTA.	IMPORTE CTA.	TOTAL
515 PRODUCTOS DE TITULOS Y CUENTAS A CORTO P				287,485.42
701 CAJA Y BANCOS			255,207,311.65	
20 C A J A			51,098.39	
1 NUMERARIO DISPONIBLE	51,098.39			
62 030318-8 BANAMEX, S.N.C. CTA. M.N. REMES		5,500,000.00		
66 0820278-8 COMERMEX, SNC, M.N. REMESAS		6,335,447.21		
70 0390838-0 V. COBRANZA RECIBIDA EN MULTIBA		5,176,549.75		
71 1008367-3 BANCO INTERNACIONAL, SNC, CTA. M		282,892.52		
74 BANAMEX, INGRESOS DIVISION DEL PACIFICO		776,804.96		
75 COMERMEX, INGRESOS DIVISION CHIHUAHUA-PA		104,986.27		
79 0876950-7 BANCOMER, SNC, M.N. CUENTA GEN	148,587,966.20			
92 10-08450-7 BANCO INTERNACIONAL, SNC PROV		591,566.35		
101 COMERMEX CUENTA ESPECIAL NO.550597-6		87,800,000.00		
702 INVERSIONES TEMPORALES			220,100,000.00	
16 VALORES EN BANCO INVERLAT		76,000,000.00		
42 CUENTA DE MULTIVALORES, S.A.		90,000,000.00		
45 ABACO CASA DE BOLSA, S.A.		54,100,000.00		
707 CUENTAS POR COBRAR			5,176,553.78	
30 DOCTOS. PARA SU COBRO EN PODER M. BANCO C	5,176,553.78			
755 SUELDOS POR PAGAR			28,906.35	
62 FINIQUITOS POR PAGAR		1,810.20		
63 LIQ. PROGRAMA RACIONALIZACION POR PAGAR		27,096.15		
760 IMPUESTOS POR PAGAR			10,524.13	
3 IMPUESTO SOBRE SUELDO		10,524.13		
788 CRÉDITOS DIVERSOS PENDIENTES DE AJUSTE			11,009,465.93	
18 ENTEROS EN SUSPENSO		21,807.11		
312 DEPÓSITO EN TRANSITO		10,987,658.82		
789 CUENTA DE ENLACE ACREEDORA ENTRE REGIONE			881,791.23	
1 EFECTIVO RECIBIDO Y PAGOS HECHOS POR OTR		776,804.96	881,791.23	
85 REGION PACIFICO	776,804.96			
86 REGION NORTE	104,986.27			
			\$	492,702,038.49

APÉNDICE C

MUESTRAS DE PRODUCTOS

STRCHCAJ

FECHA : 01-May-1997
HOJA : 001

FERROCARRILES NACIONALES DE MEXICO
S.I.I.T.E.S
REPORTE DE CHEQUES ENTREGADOS POR CAJERO

FECHA DE PAGO : 02-03-1997
BANCO : BANCOMER. S.A.
CUENTA : CUENTA GENERAL 876950-7

NO. CHEQUE	BENEFICIARIO	MONEDA	IMPORTE
8245	RANGEL SALCEDO LUIS	MN	6,000.00
8311	SANCHEZ SANTISTEBE JOSE	MN	713.10
8314	GONZALEZ ALARCON FELIX	MN	4,376.56
8315	ORTIZ GARCIA SOTERO	MN	1,507.64
8334	JUAREZ SANCHEZ MARIA ROSALINA	MN	8,260.36
8335	MENDOZA REYES F. CORNELIO	MN	4,237.65
8342	X GARCIA JAIME	MN	3,246.98
8346	GARCIA MARTINEZ OGARITA BEATRIZ	MN	14,197.60
8350	ALVARADO GUZMAN SOCORRO	MN	9,056.12
8352	GARCIA BERNAL SAULA LAURA	MN	4,124.97
8356	MARIO ROMERO TREJO	MN	3,750.00
8357	JORGE GONZALEZ MARTINEZ	MN	3,750.00
8367	ISAAC RON PEREZ	MN	5,000.00
8368	VICTOR JAVIER BERUMEN RUIZ DE CHAVEZ	MN	9,000.00
8369	MA. DEL ROSARIO DE LEON GARCIA	MN	14,000.00
8370	MUNOZ AGUILAR MARIA EUGENIA MIREYA	MN	1,400.00
8371	ARELLANO GUTIERREZ CLEMENCIA	MN	2,400.00
8373	CANTO RAMIREZ LUIS FERNANDO	MN	5,000.00
8374	CRUZ UGALDE MARGARITA	MN	4,500.00
8376	FIESCO PEREZ JORGE	MN	4,500.00
8377	FUENTES FLORES JESUS	MN	5,000.00
8378	GUTIERREZ HERNANDEZ ENRIQUE	MN	5,000.00
8379	LNARIES Y BARAGAN MA. TERESA	MN	4,500.00
8380	MARTINEZ GOMEZ GUILLERMO	MN	5,000.00
8381	MARTINEZ PEREZ JUAN MANUEL	MN	5,000.00
8382	RODRIGUEZ TENORIO RUBEN	MN	5,000.00
8383	SOSA ALTAMIRANO ARTURO	MN	5,000.00
8384	TAPIA JIMENEZ MIGUEL	MN	5,000.00
8385	VAZQUEZ JIMENEZ HUGO ACOSTA	MN	1,900.00
8386	CALZADA DURAN MARIA ELENA DE LA CRUZ	MN	2,400.00
8399	LOURDES NIETO AGUILAR	MN	64,347.29
8400	GUADALUPE MALDONADO CERVANTES	MN	14,542.20
8401	ARMANDO L. RAMIREZ FLORES	MN	6,500.00
8402	ROBERTO RUIZ NEGRETE	MN	8,000.00
8403	OLGUIN CHAPARRO YOLANDA	MN	4,749.60
8409	VERA CASTILLO MARIA DE LOURDES	MN	10,522.11
8410	JOSE TORRES ALTAMIRANO	MN	86,040.93

Sub-Total de la Cuenta CUENTA GENERAL 876950-7

Cheques : 37

347,523.11

STRCHCAJ

FECHA : 01-May-1997

HOJA : 002

**FERROCARRILES NACIONALES DE MEXICO
S.I.T.T.E.S
REPORTE DE CHEQUES ENTREGADOS POR CAJERO**

Total de Banco BANCOMER. S.A

M

347.523.11

STRCHCAJ

FECHA : 01-May-1997

HOJA : 003

FERROCARRILES NACIONALES DE MEXICO

S.I.T.E.S

REPORTE DE CHEQUES ENTREGADOS POR CAJERO

FECHA DE PAGO : 02/03/1997
 BANCO : BANCO INTERNACIONAL, S.A
 CUENTA : PRIMAS DE ANTIGUEDAD 10-08367-J

NO. CHEQUE	BENEFICIARIO	MONEDA	IMPORTE
358676	FINIQUITO DE TRABAJO MODULO M-9707	MN	7,231.60
358677	LIQUIDACION DE TRABAJO MODULO M-9707-1	MN	70,306.05
358679	LIQUIDACIONES DE TRABAJO DEL FC CH-P M-9 708D1	MN	114,090.24
358690	LIQUIDACION DEL DEPTO.SERV.ESP. M-9709C1	MN	78,419.25
358691	FINIQUITO DEL DEPTO.SERV.ESP. M-9709C1	MN	12,836.38
Sub-Total de la Cuenta PRIMAS DE ANTIGUEDAD 10-08367-J			
Cheques : 5			282,892.52

APÉNDICE C

MUESTRAS DE PRODUCTOS

STRCHCAJ

FECHA : 01-May-1997

HOJA : 004

**FERROCARRILES NACIONALES DE MEXICO
S.I.L.T.E.S.
REPORTE DE CHEQUES ENTREGADOS POR CAJERO**

FECHA DE PAGO : 02/03/1997
BANCO : BANCO INTERNACIONAL S.A.
CUENTA : PROVEEDORES 10-08450-7

NO. CHEQUE	BENEFICIARIO	MONEDA	IMPORTE
14476	GRAFO CINTAS, S.A. DE C.V.	MN	378.58
14618	IVORY TOURS, S.A. DE C.V.	MN	2,012.44
14623	GRAFO CINTAS, S.A. DE C.V.	MN	860.32
14701	MORA PANIAGUA JORGE OCTAVIO	MN	5,853.50
14710	ANA GARCIA VILLEGAS	MN	1,500.00
14719	MA. GUADALUPE AMESCUA GONZALEZ	MN	1,500.00
14728	MA. GUADALUPE AMESCUA DE RAMOS	MN	4,770.56
14734	DIGITAL DESIGN, S.A. DE C.V.	MN	21,068.00
14790	ILUSTRACION Y DISEÑO Y O MARIO RAMIREZ D IAZ	MN	13,800.00
14791	ILUSTRACION Y DISEÑO Y O MARIO RAMIREZ D IAZ	MN	17,250.00
14792	ILUSTRACION Y DISEÑO Y O MARIO RAMIREZ D IAZ	MN	9,200.00
14799	SPG ROMÉ, S.A. DE C.V.	MN	10,067.10
14805	RADIOMOVIL DIPSА, S.A. DE C.V.	MN	53,559.14
14809	GRAFO CINTAS, S.A. DE C.V.	MN	1,845.97
14816	SEGURON GENESIS, S.A.	MN	381,260.71
14818	MA. ESTELA GARCIA HERLIDIA	MN	8,340.88
14819	RAMON VILLEGAS GALLEGOS	MN	5,325.00
14820	ANTONIO GUTIERREZ VAZQUEZ	MN	4,800.00
14821	EL MUNDO ES TUYO S.A. DE C.V.	MN	22,454.15
14822	IVONNE VELAZQUEZ BELMONT	MN	2,853.15
14843	CARLOS MARIO DURAN CLEVAS	MN	1,378.85
14849	ALEJANDRO ALMADA SUAREZ	MN	4,012.00
14850	JASCENCION BARAJAS ACEVES	MN	4,240.00
14852	JOSE CRUZ RODRIGUEZ RODRIGUEZ	MN	4,240.00
14853	CESAR MEJICANO SANCHEZ	MN	4,240.00
14854	JULIO HERNANDEZ AGUILAR	MN	4,756.00
Sub-Total de la Cuenta PROVEEDORES 10-08450-7			
Cheques : 26			591,566.35

STRCHCAJ

FECHA : 01-May-1997
HOJA : 005FERROCARRILES NACIONALES DE MEXICO
S.I.I.T.E.S
REPORTE DE CHEQUES ENTREGADOS POR CAJERO

Total de Banco BANCO INTERNACIONAL, S.A. =874.458.87

STRCHCAJ

FECHA : 01-May-1997

HOJA : 006

FERROCARRILES NACIONALES DE MEXICO
S.I.T.E.S
REPORTE DE CHEQUES ENTREGADOS POR CAJERO

TOTAL GENERAL

=

1,157,351.39

FRANCISCO SANCHEZ GOMEZ
SUBJEFE DEPTO. EGRESOS