

29.51



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

LA INGENIERIA DE SOFTWARE COMO
METODOLOGIA PARA EL DESARROLLO DE
UN SISTEMA DE CHEQUES BANCARIO

T E S I S
QUE PARA OBTENER EL TITULO DE:
A C T U A R I O
P R E S E N T A :
VICTOR GABRIEL TORRES VARGAS



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 está 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.

INTRODUCCION

Este trabajo tiene como objetivo principal el desarrollo de un Sistema de Cheques, empleando para ello, las herramientas y los métodos que proporciona la Ingeniería de Software.

En el Capítulo I se da una breve explicación del entorno de la Ingeniería de Software y las partes que la conforman. Asimismo se describen las causas más importantes que favorecen su aparición.

En el Capítulo II se detallan cada una de las etapas que forman el "Ciclo de Vida" de un proyecto de Software y se mencionan algunas de las herramientas más importantes que lo auxilian en su desarrollo.

En el Capítulo III se desarrolla un Sistema de Cheques Bancario con el auxilio de la teoría descrita en los 2 capítulos anteriores. Este sistema se planea, analiza y diseña, aportando las bases para su implementación.

Por último, el Capítulo IV contiene las tablas y los apéndices creados durante el tiempo en que se desarrolla el Sistema.

I N D I C E

PAG

CAP I. LA INGENIERIA DE SOFTWARE

| | | |
|-----|--------------------------------------------|---|
| 1.1 | ¿ QUE ES LA INGENIERIA DE SOFTWARE ? | 1 |
| 1.2 | ¿ PORQUE SURGE LA INGENIERIA DE SOFTWARE ? | 4 |

CAP II. EL CICLO DE VIDA DEL SOFTWARE.

| | | |
|---------|--------------------------------------------------|----|
| 2.1 | LA ETAPA DE PLANEACION | 15 |
| 2.2 | LA ETAPA DE ANALISIS | 18 |
| 2.3 | LA ETAPA DE DISEÑO | 33 |
| 2.3.1 | EL DISEÑO ORIENTADO AL FLUJO DE DATOS | 35 |
| 2.3.1.1 | EL ANALISIS DE TRANSFORMACIONES | 39 |
| 2.3.1.2 | EL ANALISIS DE TRANSACCIONES | 41 |
| 2.3.2 | CONCEPTOS Y HERRAMIENTAS FUNDAMENTALES DE DISEÑO | 53 |
| 2.4 | LA ETAPA DE IMPLEMENTACION | 57 |
| 2.4.1 | LA CODIFICACION ESTRUCTURADA | 58 |
| 2.4.2 | EL ESTILO DE CODIFICACION | 59 |
| 2.4.3 | LA DOCUMENTACION DE APOYO | 61 |

CAP III. DESARROLLO DE UN SISTEMA DE CHEQUES.

| | | |
|-----|----------------------------------|-----|
| 3.1 | LA DEFINICION DEL SISTEMA | 63 |
| 3.2 | EL ANALISIS DEL SISTEMA | 70 |
| 3.3 | EL DISEÑO PRELIMINAR DEL SISTEMA | 87 |
| 3.4 | EL DISEÑO DETALLADO DEL SISTEMA | 100 |

CAP. IV TABLAS Y APENDICES DEL SISTEMA.

| | | |
|-----|------------------------------------------------|-----|
| - - | TARLA DE REFERENCIAS CRUZADAS ARCHIVOS/MODULOS | 108 |
| - - | TABLA DE REFERENCIAS CRUZADAS DATOS/MODULOS | 109 |
| - - | APENDICE A: DICCIONARIO DE DATOS | 110 |
| - - | APENDICE B: LAS TARJETAS "IPO" | 124 |
| - - | BIBLIOGRAFIA | 154 |

CAP I. LA INGENIERIA DE SOFTWARE

1.1. ¿QUE ES LA INGENIERIA DE SOFTWARE?

Antes de formular alguna definición, es necesario senalar brevemente el marco de referencia en el que se maneja el concepto de Ingeniería de Software.

La Ingeniería de Software (tema que se maneja en esta tesis) y la Ingeniería de Hardware son actividades englobadas dentro de una categoría principal que llamaremos "Ingeniería de los Sistemas de Cómputo", en donde cada una de estas disciplinas, representa un intento por desarrollar en forma ordenada, sistemas basados o enfocados hacia el uso de las computadoras.

De esta manera, la Ingeniería de los Sistemas de Cómputo es una actividad encaminada a la resolución de los problemas planteados en el ambiente del manejo del software y el hardware; se intenta descubrir, analizar, e implementar aquellas funciones que se requiere para la solución correcta del problema, utilizando la computadora.

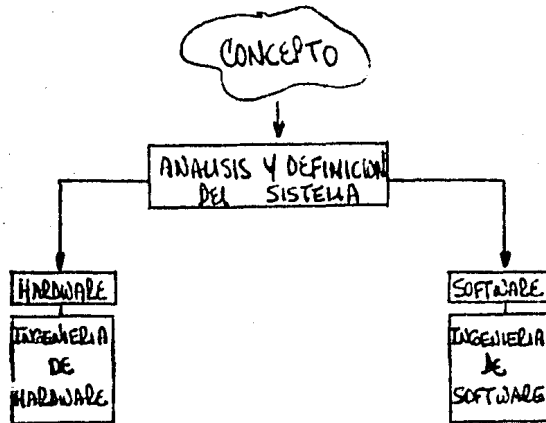


FIG # 1 ENTORNO DE LA INGENIERIA DE SOFTWARE

En particular, en lo referente a la Ingeniería de Software, podemos señalar las siguientes definiciones :

"Ingeniería de Software es la disciplina tecnológica y administrativa dedicada a la producción sistemática de productos de programación que son desarrollados y modificados a tiempo y dentro de un presupuesto definido." (1)

"Ingeniería de Software es una aplicación práctica del conocimiento científico en el diseño y construcción de programas para computadoras y la documentación asociada requerida para desarrollarlos, operarlos y mantenerlos." (8)

"La Ingeniería de Software es un enfoque sistemático para el desarrollo, operación, mantenimiento y eliminación del software; software en este caso se define como aquellos programas, procedimientos, reglas y documentación posible asociada con la computación, así como los datos pertenecientes a la operación de un sistema de cómputo." (6)

De una manera general, podemos afirmar que la Ingeniería de Software se ocupa de los métodos, herramientas y técnicas empleadas para el desarrollo y mantenimiento de software.

En este entendido, podemos ampliar el concepto afirmando que los principales objetivos de la Ingeniería de Software son los siguientes :

- Manejar una metodología bien definida que coordine el "ciclo de vida" del software, que se encuentra formado por las etapas de planeación, desarrollo y mantenimiento del proyecto.
- Establecer un conjunto de componentes del software que documenten cada etapa de su ciclo de vida y permitan un seguimiento paso a paso.
- Establecer un conjunto de eventos significativos durante el ciclo de vida, que pueden ser revisados a intervalos regulares.

El ciclo de vida es un concepto a largo plazo del proyecto de software que engloba las actividades que ocurren antes de que se inicie el desarrollo y después de que el software se encuentra en operación formal.

A continuación se describen brevemente las etapas que conforman el mencionado ciclo de vida, los documentos que se producen y las revisiones que se llevan a cabo :

Fase de Planeación. Como se ha notado, el software es siempre parte de un sistema mas grande basado en la computadora y por tanto, el análisis y la definición deben ocurrir de manera previa o

En este entendido, podemos ampliar el concepto afirmando que los principales objetivos de la Ingeniería de Software son los siguientes :

- Manejar una metodología bien definida que coordine el "ciclo de vida" del software, que se encuentra formado por las etapas de planeación, desarrollo y mantenimiento del proyecto.

- Establecer un conjunto de componentes del software que documenten cada etapa de su ciclo de vida y permitan un seguimiento paso a paso.

- Establecer un conjunto de eventos significativos durante el ciclo de vida, que pueden ser revisados a intervalos regulares.

El ciclo de vida es un concepto a largo plazo del proyecto de software que engloba las actividades que ocurren antes de que se inicie el desarrollo y después de que el software se encuentra en operación formal.

A continuación se describen brevemente las etapas que conforman el mencionado ciclo de vida, los documentos que se producen y las revisiones que se llevan a cabo :

Fase de Planeación. Como se ha notado, el software es siempre parte de un sistema mas grande basado en la computadora y por tanto, el análisis y la definición deben ocurrir de manera previa o

conjuntamente con la planeación del proyecto.

La fase de Planeación en la Ingeniería de Software se inicia con la etapa de la planeación del software. En esta etapa se desarrolla una descripción limitada del alcance del proyecto, se realiza una estimación de los recursos que se requieren para su desarrollo y de los costos y programas de trabajo

El propósito de esta etapa es obtener una indicación preliminar de la viabilidad del proyecto en relación a restricciones de costo y recursos que han sido establecidos de antemano

Se produce un documento llamado "Plan del Software" que contiene los criterios arriba mencionados y es revisado por la Dirección que maneja el proyecto.

La siguiente etapa en la fase de planeación es el análisis y la definición de los requerimientos del software, en donde el sistema es definido en detalle. El flujo y la estructura de la información proporcionan la definición de las interrelaciones del sistema y sus características funcionales.

El análisis y la definición de los requerimientos del software es un esfuerzo conjunto realizado por el desarrollador del proyecto y por los usuarios.

Esta fase de planeación culmina con una revisión técnica de

la "Especificación de los Requerimientos del Software" y una vez que es aceptada, se revisan el alcance, recursos, costo y programa de trabajo establecidos en el "Plan de Software" para su posible redefinición.

Fase de Desarrollo. En esta fase se translada un conjunto de requerimientos al elemento operacional del sistema, es decir, se desarrolla una estructura modular, se definen sus interrelaciones y su estructura de datos.

Se utilizan heurísticos de diseño (guías) para evaluar con corrección la calidad del software. Esta etapa preliminar de diseño es revisada para checar su consistencia con los requerimientos. Se genera un primer "Documento de diseño" que forma parte de la configuración del proyecto.

Los aspectos de procedimiento de los módulos se consideran durante la siguiente etapa de la fase de desarrollo: el diseño detallado, en donde se utilizan herramientas específicas para proporcionar una descripción detallada del software.

Finalmente, después de haber concluido las dos etapas de diseño, se realiza la codificación, es decir, la generación de programas con el uso de un lenguaje de programación apropiado. La codificación es revisada en sus aspectos de estilo y claridad y se producen listados del lenguaje fuente por cada módulo del sistema.

Fase de Mantenimiento. Cuando el software se encuentra en uso activo, debe ser mantenido. Se revisa la configuración del sistema para asegurar que toda la documentación esta disponible y es adecuada para las actividades de mantenimiento que se realizarán en el futuro.

Las tareas asociadas con el mantenimiento del software dependen del tipo de mantenimiento que se efectue, pero en cualquier caso, la modificación del sistema debe incluir una configuración completa (todos los documentos desarrollados en las fases de planeación y desarrollo) no solo la codificación.

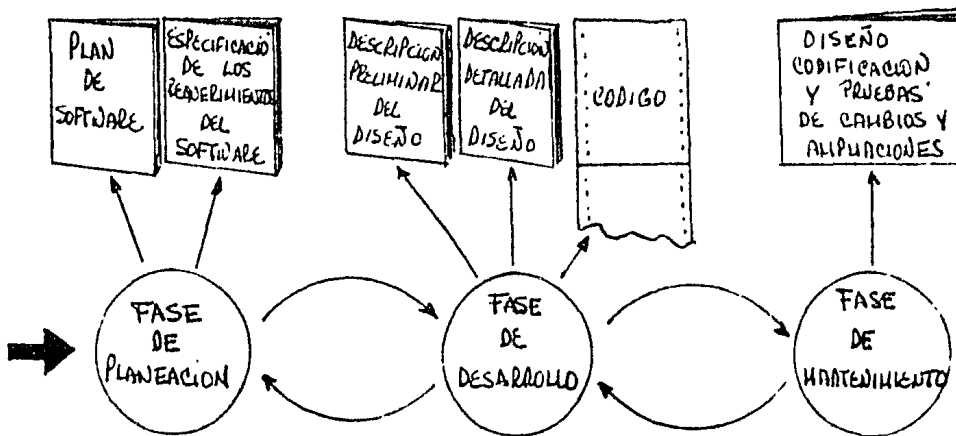


FIG # 2 EL CICLO DE VIDA DEL SOFTWARE

Cabe apuntar como complemento, algunas diferencias que existen entre los conceptos de Ingeniería de Software y los que tradicionalmente se manejan como análisis-programación.

La Ingeniería de Software difiere de la programación tradicional, en que la primera utiliza técnicas de ingeniería para especificar, diseñar, instrumentar, validar y mantener los productos dentro del tiempo y el presupuesto establecido para el proyecto.

Los ingenieros en software, además de detallar el código en la instrumentación, el empaquetado y modificación de los algoritmos y la estructura de los datos codificados en un lenguaje de programación, se ocupan de aspectos como el análisis, diseño, verificación y prueba de programas, la documentación, el mantenimiento y la administración del proyecto.

Por otra parte, el término Software suele considerarse como sinónimo de código fuente. Aquí se tomará como sinónimo de producto de programación, de modo que el software incluye el código fuente y todos los manuales asociados, así como la documentación propia del producto: los manuales de requisitos, especificaciones de diseño, el código fuente, planes de prueba, principios de operación, controles de calidad, informes de problemas encontrados, manuales de mantenimiento y de usuario, documentos para la instalación y guías de entrenamiento.

La Ingeniería de Software es una disciplina profesional nueva y de rápida evolución, que requiere de profesionales con capacitación avanzada, basada en fundamentos intelectuales bien establecidos y un periodo de aprendizaje, a fin de obtener experiencia práctica dentro de la profesión.

1.2 ¿ PORQUE SURBE LA INGENIERIA DE SOFTWARE ?

A partir de la década de los 70's se ha dedicado cada vez mas atención a la tecnología para el desarrollo de proyectos de software.

A medida que los sistemas de cómputo se multiplican, se hacen mas complejos y penetran con mayor profundidad en las sociedades modernas, se hace evidente la necesidad de enfoques sistemáticos para el desarrollo del software, así como para su mantenimiento. La Ingeniería de Software es el campo de estudio relacionado con esta nueva tecnología.

La necesidad de enfoques sistemáticos para el desarrollo y mantenimiento de proyectos de software se patentizó en la década de los 60's; durante esta década aparecieron las computadoras de 3a. generación y se desarrollaron técnicas de programación como la multiprogramación y el "tiempo compartido".

Estas nuevas capacidades aportaron la tecnología necesaria para establecer sistemas de cómputo interactivo, de multiusuarios, en línea y en tiempo real; de esta manera surgieron aplicaciones nuevas como sistemas de reservaciones aéreas, bancos de información médica, control de procesos, dirección y control de equipos militares, etc.

De 1970 en adelante y a pesar del sensible aumento en la capacidad de almacenamiento de las computadoras, ha surgido un

crecimiento exponencial en la información que se necesita manejar en las diferentes organizaciones modernas como consecuencia del desarrollo mencionado, resulta por tanto, de máxima importancia la creación de criterios para manejarlo porque, aunque muchos de estos sistemas fueron desarrollados y producidos adecuadamente, gran parte de los sistemas intentados nunca se produjeron, y de aquellos realizados, muchos quedaron sujetos a sobrecosto, entregas tardías, poca confiabilidad, ineficiencia y sobre todo, poca aceptación por parte de los usuarios.

Para mejorar al respecto, debemos entender el proceso de desarrollo del software y encontrar criterios de medición de su calidad y confiabilidad para su operación futura.

Los problemas asociados al desarrollo del software han sido originados tanto por su naturaleza propia, como por errores de criterio de los encargados de dirigir su desarrollo, aunque es de justicia mencionar que la experiencia en el área lleva apenas poco más de 30 años.

Los directivos de alto o regular nivel son casi siempre responsables del desarrollo de los productos y es esencial que mantengan buenos niveles de comunicación dentro del ambiente de creación del software, con el solicitante del proyecto (persona física o moral), el staff de diseño, los usuarios, etc. para evitar que la naturaleza especial del software y los problemas asociados con su desarrollo no sean bien entendidos.

Otro aspecto importante, es el hecho de que la mayor parte de nosotros resistimos los cambios en la forma de pensar, y quizá sea esta la razón mas importante de los problemas en el ambiente de producción del software, pues mientras el hardware ha experimentado enormes cambios, la gente responsable de manejar ese potencial, se opone al cambio cuando este se plantea y se resiste al mismo cuando este cambio se ha llevado a cabo.

Los problemas mas importantes que surgen durante la realización de un proyecto de software pueden enumerarse de la siguiente manera :

- Los programas de trabajo y las estimaciones de costos son inadecuados. La demora ocasionada de semanas y meses producen como consecuencia altos pagos por retrasos en el programa de trabajo.

- No se evalúan adecuadamente la eficacia de nuevas herramientas y técnicas. Puede suceder también que el equipo técnico tenga poco entrenamiento formal en nuevas técnicas de desarrollo, y cada individuo enfoca la actividad de "escribir programas" solo con su experiencia de trabajo, tal vez con malos hábitos que dan por resultado, un software de poca calidad y difícil mantenimiento.

Y aunque un conjunto formal de estas herramientas de desarrollo no existe todavía, muchas de ellas están disponibles y las más están en diferente estado de desarrollo; aquellas que están disponibles deben ser evaluadas, seleccionadas y utilizadas

como parte de una solución.

- La insatisfacción del cliente con el sistema terminado o en operación. Esto puede deberse a que no existió una indicación correcta de los requerimientos del cliente, posiblemente a consecuencia de una comunicación pobre entre el cliente y el desarrollador del software.

La calidad del software es puesta en duda porque no existe una validación sistemática y técnicamente completa o no se aplican conceptos cuantitativos de confiabilidad.

- El software producido es difícil de mantener y nos encontramos con que, tan grandes como los costos de desarrollo son los costos de mantenimiento.

El costo de mantenimiento se refiere al trabajo necesario para detectar errores en el software que ha sido formalmente liberado y se encuentra en operación. Además, casi todos los grandes sistemas que han sido hechos para operar durante años, han sido modificados de una manera u otra.

Indicados los criterios anteriores, podemos mencionar que, algunas veces, al no aplicar una metodología adecuada, damos rienda suelta a nuestro entusiasmo, desarrollando sistemas cuya

sofisticación alcanza costos inmensos y arrastran un nivel de irrealidad que sobrepasa sus beneficios, a veces desplazando personal por computadoras, que tienen un menor costo, pero con altos índices de error.

También es importante señalar que, aunque es muy posible que el costo del hardware llegue a mantenerse uniforme por la altísima producción que existe, el costo del software continuará incrementándose por varias razones :

- Con cada nueva aplicación, aparecerán nuevos programas, diseños, etc.

- Las computadoras modernas que sustituirán a sus antecesoras, requerirán de nuevo software o modificaciones al software existente.

- La programación es una labor muy técnica y es afectada fácilmente por la inflación, y a pesar de que poseemos técnicas novedosas que incrementarán la productividad, es poco probable que compensen las grandes cargas de trabajo futuras.

Siendo conscientes de esta situación, planteada sobre los problemas a los que se enfrentan desarrolladores y consumidores de software. Que criterios se pueden seguir para abordarlos ?

Las respuestas podrían ser :

- Aplicación de todas las técnicas disponibles de ingeniería y de dirección para desarrollar el proyecto considerando tanto costo como efectividad.

- Desarrollo de técnicas adecuadas, aunque al principio imperfectas, en actividades donde no existen tales.

- Implementación de un programa amplio y bien establecido de investigación.

Por otra parte, existen nuevos métodos y tecnologías, como por ejemplo, la evolución de las técnicas de programación estructurada, con sus tres estructuras de control :

Secuenciación, Selección e Iteración

Y también, el avance mas importante en cuanto a dirección, que es la adopción casi universal, del control en la configuración del software en proyectos grandes y la documentación adecuada durante el diseño, utilizando métodos de representación gráficos, como los diagramas de flujo de datos o las cartas HIPN.

En los capítulos siguientes se describen con mas detalle las fases del "ciclo de vida" del software, así como las técnicas y herramientas que se aplican en mayor medida a fin de alcanzar un nivel óptimo en el desarrollo de este tipo de proyectos.

CAP II. EL CICLO DE VIDA DEL SOFTWARE

2.1 LA ETAPA DE PLANEACION EN LA INGENIERIA DE SOFTWARE

Durante la etapa de planeación de un proyecto de software, se desarrollará una descripción inicial de su alcance, es decir, definido el problema y conocidas las restricciones que existen para conseguir una solución, se pueden formular metas y requisitos preliminares.

Las metas son logros por alcanzar y sirven para establecer un marco de referencia para el desarrollo del proyecto. Algunas metas se aplican a todos los productos, por ejemplo, todo proyecto de programación debe ser útil, comprensible, eficiente y debe producir el sistema a tiempo.

Los requisitos especifican las capacidades que debe tener el sistema para lograr la solución del problema por el que fue creado. Estos se deben establecer para definir la funcionalidad, el rendimiento del equipo y las interrelaciones con el usuario.

En esta etapa de planeación, también se describen los mecanismos para el manejo de recursos humanos. La mayoría de los proyectos comerciales y gubernamentales involucran a mucha gente. La tensión, la interacción y en general la dinámica de la relación interpersonal, debe ser tomada en cuenta.

Es importante estimar la tasa esperada de gastos que

originen el uso de recursos y crear un programa apropiado de disponibilidad de personal ya que, por ejemplo, reducir el personal de programación durante el proyecto, puede causar problemas si la reducción es prematura.

Los recursos humanos son el corazón del desarrollo de un proyecto, pues se encargan de la dirección, comunicación y control del ambiente donde se manejará el software.

Asimismo la cantidad de personal requerida a través de un proyecto de desarrollo, no es constante, por lo regular, la fase de planeación y el análisis lo efectúa un grupo pequeño de individuos, el diseño preliminar un grupo mayor (aunque todavía pequeño) y el diseño detallado un grupo grande de personas. La fase inicial de mantenimiento puede requerir un número considerable de personas, aunque este número debiera disminuir en poco tiempo.

2.1.1 EL ESTUDIO DE FACTIBILIDAD

De entre varias estrategias de solución la gente encargada de la planificación, debe revisar cada una de ellas, para realizar estudios de factibilidad y estimaciones de costos preliminares.

La factibilidad de un proyecto de software se puede establecer a través del análisis sobre las restricciones de la solución escogida. Estas restricciones establecen las fronteras del espacio de solución (alcance), y el análisis de factibilidad

determinará si el proyecto es posible dentro de dichas fronteras.

Un proyecto será factible si las metas y los requisitos del proyecto se pueden satisfacer dentro de las restricciones de tiempo, recursos y tecnología disponible.

2.1.2 EL PLAN DEL SOFTWARE

El documento que se produce en esta etapa de planeación es el "Plan de Software", que contiene el modelo del ciclo de vida que se utilizará, la estructura organizacional del proyecto, el programa preliminar del desarrollo, estimados preliminares de costos y recursos, así como la descripción de las técnicas y herramientas que se aplicaran y los estándares que se seguirán.

Cuando se ha preparado el documento, se realiza una revisión de la factibilidad del proyecto, para determinar el inicio del desarrollo o su rediseño dentro del "Plan del Software", incluso si se da otra dirección al proyecto, se pueden requerir cambios y se debe realizar otra revisión de factibilidad.

2.2 EL ANALISIS DE SISTEMAS EN LA INGENIERIA DE SOFTWARE

Una especificación técnica y completa de los requerimientos del software es esencial para lograr un desarrollo exitoso del producto; su objetivo es especificar total y consistentemente los requerimientos técnicos del proyecto, de manera concisa y sin ambigüedades.

Las metas ó alcance del proyecto que son establecidas durante la etapa de planeación, se revisan en detalle y son utilizadas como indicadores, en el análisis de requerimientos, para alcanzar los siguientes objetivos :

- Proporcionar una base para el desarrollo del software, determinando el flujo y la estructura de la información que manejará el sistema.

- Dar una descripción del propio sistema, identificando los detalles de las interfaces, elaborando una descripción detallada de las funciones, determinando restricciones de diseño y definiendo los criterios de validación.

- Establecer y mantener canales de comunicación con el usuario o el contratante, de modo que los dos objetivos mencionados previamente, puedan ser satisfechos.

Este último objetivo es quizá la clave de un buen análisis; el analista debe establecer contacto con la dirección y el staff técnico de la organización que manejará el producto, y con la organización que lo desarrollará.

Para que este análisis cumpla sus objetivos lo mejor posible, la comunicación entre las partes involucradas debe ser muy buena. En muchas ocasiones aparecen "lagunas" por desinformación o mala interpretación en las pláticas.

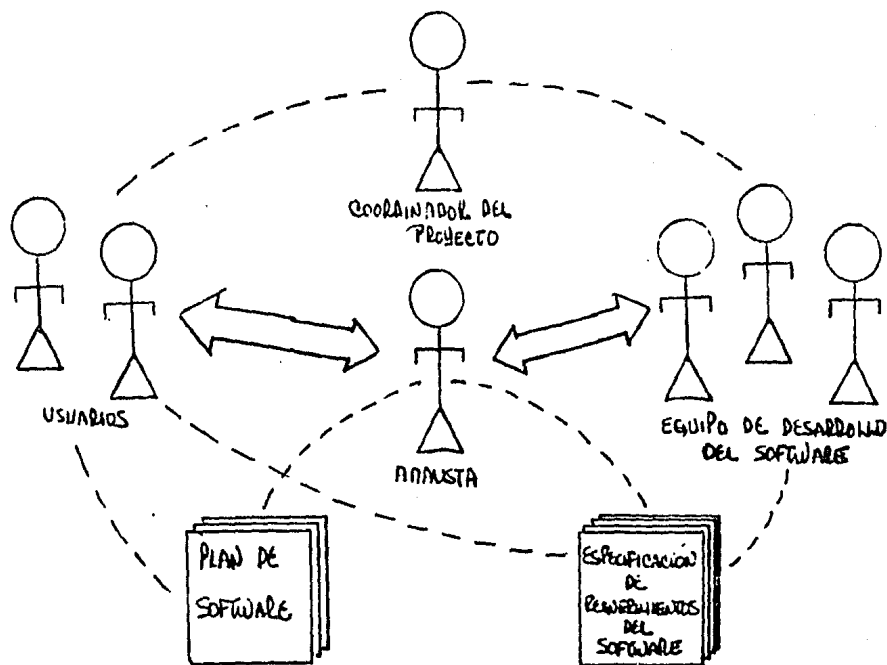


FIG # 3 CANALES DE COMUNICACION PARA EFECTUAR EL ANALISIS DE REQUERIMIENTOS

Para reforzar esta comunicación durante las actividades de análisis, se desarrolla un "Manual Preliminar del Usuario" y aunque puede parecer prematuro crear un documento de esta naturaleza en esta fase del ciclo de vida del proyecto, dicho manual servirá para alcanzar dos propósitos

- Fuerza al analista a considerar el punto de vista del usuario sobre el diseño del proyecto, que es particularmente importante en sistemas interactivos y

- Pretende también que el usuario quede motivado para hacer una revisión de los requerimientos que el analista le presenta en un documento "activo", que define las interrelaciones computadora-operador de manera explícita.

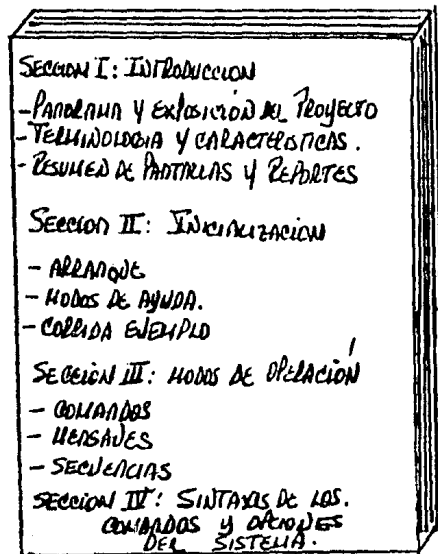


FIG 4 UN EJEMPLO DE BOSQUEJO DEL MANUAL DEL USUARIO

El análisis de requerimientos del software puede ser dividido en :

- Reconocimiento del problema. Como inicio, el analista estudia el "Plan del Software", ya que es necesario entender el proyecto en el sentido de que es un elemento de un sistema mas general, y revisa el alcance (metas) que fue utilizado para generar estimaciones durante la fase de planeación. Cuando el analista tiene pleno conocimiento del problema, se establecen los canales de comunicación para llevar a cabo el análisis.

- Evaluación y síntesis de las soluciones. Durante esta importante actividad, el analista debe evaluar el flujo y la estructura de la información y realizar una definición de las interrelaciones del sistema y sus características funcionales, además de descubrir restricciones de diseño.

Cada una de estas actividades sirve para describir el problema de manera que pueda sintetizarse un enfoque completo o solución.

Evaluando cada problema, el analista sintetiza una o mas soluciones. El proceso de evaluación y síntesis continua hasta que, tanto el analista como el usuario, se sienten seguros de que el software puede ser especificado adecuadamente, para iniciar el diseño.

Estas tareas de análisis, representan un esfuerzo para conseguir una representación del software que será revisado y aprobado por la organización contratante, aunque generalmente, las especificaciones de requisitos, son hechas en un esfuerzo conjunto entre analista y usuarios.

El documento desarrollado como parte del análisis de requerimientos, es la "Especificación de Requerimientos del Software", que amplía el alcance del proyecto identificado en el "Plan del Software", estableciendo una descripción completa de la información, una descripción detallada del funcionamiento, criterios de validación apropiados y otros aspectos desarrollados con los requerimientos.

Una vez que la "Especificación de Requerimientos del Software" y el "Manual Preliminar del Usuario" han sido elaborados, se debe realizar un revisión conjunta entre analista y usuarios.

También se revisa el impacto de los requerimientos sobre estimaciones de costos, recursos, programa de trabajo y, si llega a ser necesario, se modifican estos conceptos, para reflejar más conocimiento y se actualiza el "Plan de Software".

Debido a que la especificación es la base de la fase de desarrollo, la revisión debe realizarse con precaución; en esta revisión deben considerarse temas como :

1. ¿ Son consistentes las metas y objetivos del software con los del sistema ?
2. ¿ Han sido descritas todas las interfaces para todos los elementos del sistema ?
3. ¿ Estan bien definidos el flujo y la estructura de información para el sistema ?
4. ¿ Son realistas las restricciones de diseño ?
5. ¿ Se han establecido en detalle los criterios de validación ?
6. ¿ Existen inconsistencias, omisiones o redundancia ?
7. ¿ Como se han afectado las estimaciones descritas en el "Plan del Software" ?

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1.- INTRODUCCION</p> <p>2.- DESCRIPCION DE LA INFORMACION</p> <ul style="list-style-type: none"> - DIAGRAMAS DE FLUJO DE DATOS - GLOSARIO DE DATOS. - DESCRIPCION DE LAS INTERFACES DEL SISTEMA. <p>3.- DESCRIPCION FUNCIONAL</p> <ul style="list-style-type: none"> - FUNCIONES - PROCESOS - RESTRICCIONES DE DISEÑO | <p>4.- CRITERIOS DE VALIDACION</p> <ul style="list-style-type: none"> - TIPOS DE PRUEBAS - RESPUESTAS ESPERADAS DE SOFTWARE. - CONSIDERACIONES ESPECIALES. <p>5.- BIBLIOGRAFIA.</p> <p>6.- APENDICES.</p> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

FIG 5 UN BOSQUEJO PARA LA "ESPECIFICACION DE LOS REQUERIMIENTOS DEL SOFTWARE

Todos estos criterios servirán de base para verificar el diseño del software.

2.2.1 LAS HERRAMIENTAS PARA HACER EL ANALISIS

Para que el analista consiga una "Especificación de Requerimientos del Software" aceptable, bien fundamentada y confiable, es necesario que haga uso de las técnicas y herramientas de que dispone la Ingeniería de Software.

Anteriormente se habló de las actividades de evaluación y síntesis que realiza el analista, a fin de determinar la mejor solución a su problema.

Estas actividades, que forman la base del análisis de requerimientos, son desarrolladas con ayuda de herramientas como los "Diagramas de Flujo de Datos", para describir el flujo de la información y los "Diagramas Jerárquicos" para describir la estructura de la información.

2.2.1.1 DIAGRAMAS DE FLUJO DE DATOS

La información que se mueve a través de un sistema de software, es modificada por una serie de transformaciones.

Un diagrama de flujo de datos es una técnica gráfica que describe el flujo de información y las transformaciones que son

aplicadas cuando los datos se mueven de la entrada a la salida del sistema; cualquier sistema puede ser representado como un diagrama de flujo de datos.

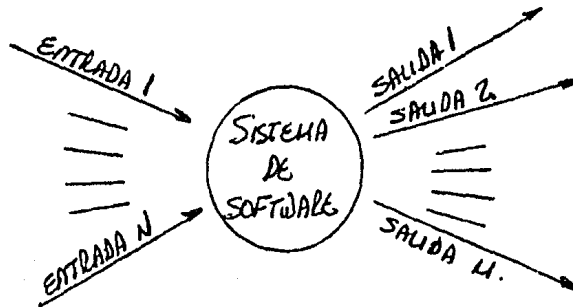


FIG 6 MODELO FUNDAMENTAL DEL SISTEMA

El sistema completo es ilustrado como una simple burbuja, con los datos de entrada y salida indicados por flechas que entran y salen de la burbuja respectivamente. En general este modelo fundamental es refinado en una serie de burbujas, cada una representando un transformación que ocurre a lo largo del recorrido del flujo, desde la entrada hasta la salida del sistema.

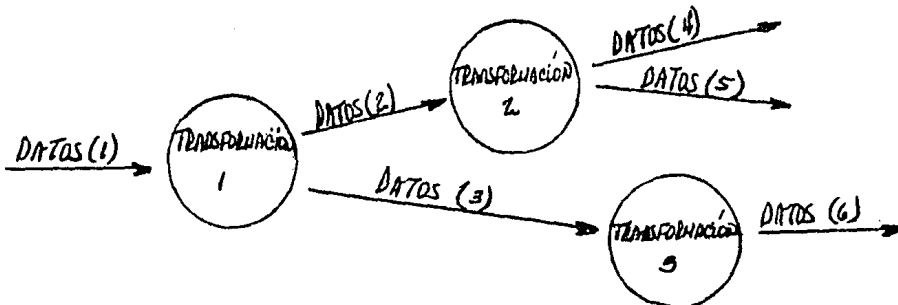


FIG 7 DIAGRAMA DE FLUJO DE DATOS REFINADO

Es importante notar que en estos diagramas, no hay indicación explícita de las secuencias de los eventos; las secuencias pueden estar implícitas, pero una representación de procedimientos se hace hasta la etapa de diseño del proyecto.

Los diagramas de flujo de datos tienen tres características importantes :

- Puede ser representado el flujo de cualquier sistema (manual, automatizado o híbrido).

- Cada burbuja puede requerir refinamiento adicional hasta que se alcance un entendimiento completo del proceso.

- Se enfatiza el flujo de datos más que el flujo de control.

Las herramientas gráficas para construir un diagrama de flujo de datos, son sencillas :

- El flujo de datos se representa por flechas etiquetadas.

- Las transformaciones o procesos se representan por burbujas, también etiquetadas.

Los orígenes y los destinos de la información, son anotados como cajas etiquetadas, y

- La información almacenada se representa como rectángulos abiertos por uno de sus extremos.

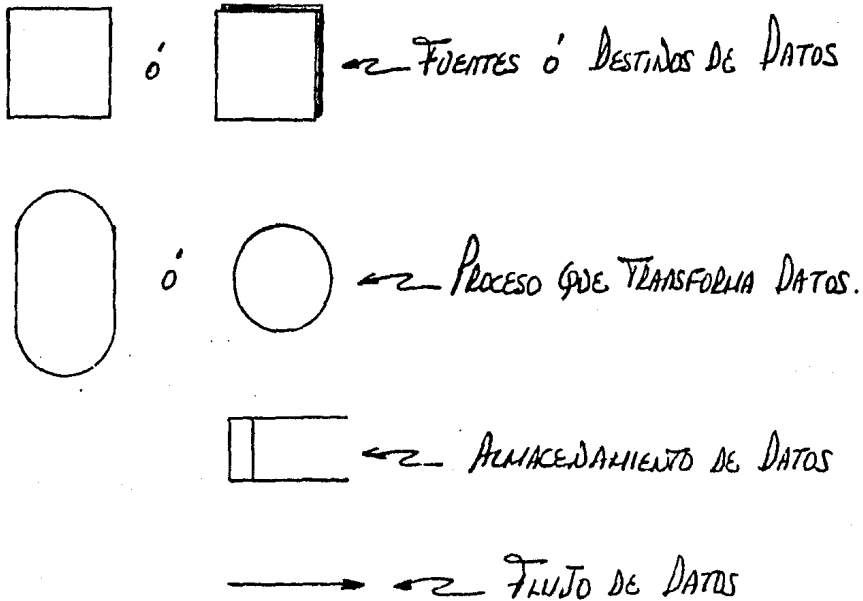


FIG 8 HERRAMIENTAS GRAFICAS EN LOS DIAGRAMAS
DE FLUJO DE DATOS

Estos diagramas son herramientas gráficas muy valiosas durante el análisis de los requerimientos del software, sin embargo, pueden provocar confusión, si su utilidad se confunde con la de un diagrama de flujo de control. Un diagrama de flujo de datos describe el flujo de información sin ninguna notación explícita de control.

Existen algunos consejos que ayudan en la construcción de diagramas de flujo adecuados :

- El diagrama de flujo de datos inicial debe ser siempre el sistema fundamental, es decir, el sistema con una sola burbuja que indica la transformación general y las entradas y salidas principales.

- Los archivos de entrada/salida ^{deben} (deben) ser anotados cuidadosamente.

- Todas las flechas y burbujas deben ser anotados cuidadosamente (con nombres significativos).

- La continuidad en el flujo de la información debe ser mantenida y

- Las burbujas deben ser refinadas una a la vez.

Existe una tendencia a complicar el diagrama de flujo de datos. Esto ocurre cuando el analista intenta mostrar procedimientos de control en lugar de flujo de información.

2.2.1.2. REPRESENTACION DE LA ESTRUCTURA DE DATOS.

Se mencionó también como herramienta de ayuda en el análisis de requerimientos a los Diagramas de Estructuras de Información, que

son representaciones de la relación lógica que existe entre los datos elementales del sistema.

La estructura de datos coordina la organización, los métodos de acceso, el grado de asociatividad y las alternativas de procesamiento de información.

Existe un número limitado de estructuras clásicas de datos, que forman la base para representaciones más sofisticadas, las cuales se mencionan a continuación :

- La más sencilla de estas estructuras es el dato "escalar"; representa el elemento más simple de información y puede ser accedido tan solo con un identificador que indique su lugar de almacenamiento en memoria.

- Cuando los escalares se organizan en forma contigua, como por ejemplo una lista, se forma un vector, que constituye una de las estructuras más comunes.

- Cuando el vector se extiende a 2,3 o un número mayor de dimensiones, se crea un espacio n-dimensional. El espacio n-dimensional más común es el espacio bidimensional.

Los escalares, vectores y espacios pueden ser organizados en una gran variedad de formas, por ejemplo, como una lista eslabonada, que es una estructura que organiza escalares no contiguos, vectores

o espacios de manera que puedan ser procesados como una lista.

Cada nodo dentro de esta lista contiene una organización apropiada (por ejemplo un vector) y uno o más apuntadores que indican la dirección de almacenamiento del siguiente nodo en la lista.

Se pueden adicionar y eliminar nodos en cualquier lugar de la lista, redefiniendo apuntadores para definir la nueva lista.

Existen otras estructuras de datos que pueden ser construidas, usando las estructuras descritas anteriormente, por ejemplo, una estructura de datos jerárquica es implementada utilizando listas multiligadas que contengan escalares, vectores y posiblemente espacios n-dimensionales.

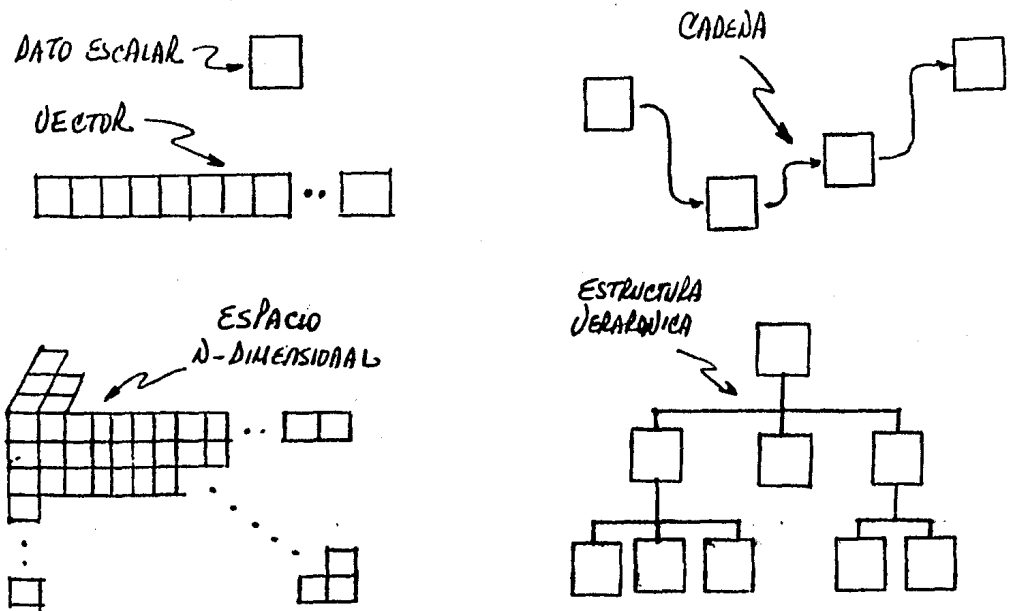


FIG 9 ESTRUCTURAS DE DATOS BASICAS

Una estructura jerárquica será comúnmente encontrada en aplicaciones que requieren la información por categorías y por criterios de asociación.

La categorización se refiere a un agrupamiento de información que tiene alguna categoría genérica.

La asociatividad se refiere a la capacidad para asociar información de diferentes categorías.

Durante el análisis de requerimientos de software se encuentra con frecuencia una estructura de datos jerárquica y, debido a que la estructura de información puede tener un impacto significativo en los requerimientos para el diseño, el analista debe representar a la jerarquía de una manera clara y sin ambigüedades.

El método siguiente puede ser utilizado para representar estructuras de datos jerárquicos y puede describir tanto datos como software.

DIAGRAMAS JERARQUICOS POR BLOQUES

Estas representaciones muestra la información como una serie de bloques en varios niveles o como una estructura de árbol.

En el primer nivel de la estructura, se utiliza un único bloque para representar la jerarquía completa y enseguida, los

bloques de niveles subsiguientes que representan diferentes categorías de información.

En el nivel más bajo del diagrama cada bloque contiene datos individuales (por ejemplo 'cantidades', 'nombres', etc.).

En la medida en que el diagrama jerárquico por bloques sea refinado, se describirá lógicamente mayor detalle.

Este modo de representación se ajusta bien al análisis de requerimientos. El analista comienza por categorizar la información del nivel más alto y el refinamiento continúa a lo largo de cada ramificación hasta que todos los detalles se describen.

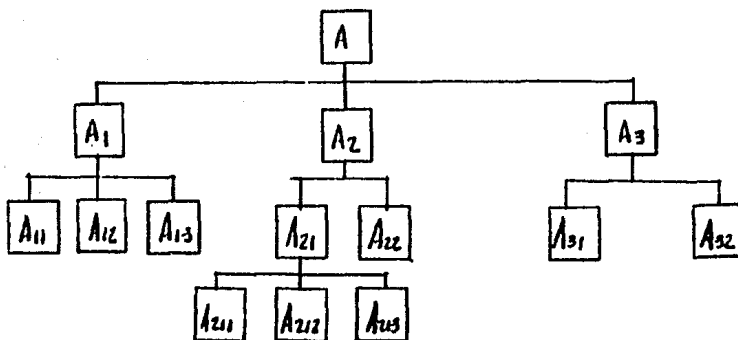


FIG 10 EJEMPLO DE UN DIAGRAMA JERARQUICO

La asociatividad entre categorías de información en el árbol jerárquico se implica, aunque no se muestra de manera técnica; estos diagramas no especifican indicadores de registro, formato de datos o atributos internos.

2.3 EL DISEÑO DE SISTEMAS EN LA INGENIERIA DE SOFTWARE

El diseño se puede describir como un proceso de varias etapas, en el cual las representaciones de la estructura y los procedimientos del software se sintetizan a partir de los requerimientos del sistema.

Las metodologías de diseño de software han sido derivadas de consideraciones sobre el flujo y la estructura de la información obtenida durante la etapa del análisis.

Es posible definir el diseño como el puente entre los requisitos de software y la instrumentación o implantación que satisfaga estos requisitos.

El diseño está involucrado en la identificación de los componentes del software (funciones, flujo de datos, almacenamientos, etc), especificando las relaciones entre ellos, la estructura del sistema, manteniendo un registro de las decisiones y proporcionando un documento base para realizar la implementación.

Podemos hablar del diseño englobándolo dentro de dos categorías principales : el diseño preliminar y el diseño detallado.

El diseño preliminar comprende la identificación de los componentes de la programación y su descomposición en módulos de

procesamiento, estructuras de datos y la especificación de las interrelaciones entre los componentes.

El diseño detallado se refiere a aspectos sobre la construcción de algoritmos, de las estructuras de datos y las interconexiones del sistema.

Otras tareas de diseño detallado podrían ser, la adaptación de código existente, modificaciones de algoritmos comunes, creación de nuevos algoritmos, diseño de representaciones de datos e integración del producto final.

Es importante saber cuando el diseño esta lo suficientemente detallado y se debe iniciar la implementación. La regla práctica es considerar que un diseño esta completo si los programadores pueden implementar el sistema trabajando solo a partir de las especificaciones de diseño.

Hay dos revisiones durante el diseño : La revisión preliminar, en donde se revisan los requisitos del software y la revisión final donde se revisan todos los productos de trabajo verificados durante las revisiones previas, aunque se pone especial énfasis en las especificaciones del diseño detallado y en un plan de pruebas.

En relacion a la metodología del diseño, a continuación se

mencionan dos corrientes importantes, que se encuentran estrechamente relacionadas con la información que se manejará en el sistema; estas corrientes son :

- EL diseño orientado al flujo de datos y
- El diseño orientado a la estructura de datos.

2.3.1. EL DISEÑO ORIENTADO AL FLUJO DE DATOS

El flujo de información es un concepto clave durante el análisis de requerimientos del software.

Comenzando con el modelo fundamental del sistema (ya revisado), la información puede ser representada como un flujo continuo que sufre una serie de transformaciones a medida que recorre el flujo de la entrada a la salida del Sistema.

El diagrama de flujo de datos es usado como una herramienta gráfica para ilustrar el flujo de información, mediante este método, se definen un número de gráficas, que transformaran el flujo de información en una estructura de software.

Esta metodología da un transición conveniente de las representaciones de información : el diagrama de flujo de datos que se representó durante el análisis de requerimientos a una estructura de software. Esta transición del flujo de información a la estructura, se realiza mediante un proceso que abarca varias actividades :

- Se establece un flujo de información.
- Los límites del flujo son establecidos.
- El diagrama de flujo de datos se transforma en la estructura del sistema.
- El control de la estructura se define por factorización.
- La estructura resultante se refina con el uso de medidas de diseño y heurísticos.

Para describir algunas corrientes de diseño orientada al flujo de datos es necesario definir dos tipos de flujo.

A) "El Flujo de Transformaciones"

Recordando el modelo del sistema fundamental, la información debe entrar y salir del sistema de manera "externa" (por ejemplo, como datos introducidos a través de una terminal); estos datos deben ser convertidos a una forma interna para procesarlos.

La información entra al sistema a través de un recorrido que transforma los datos externos en datos internos. El flujo a lo largo de tales recorridos se nombre "aférente"; en el núcleo del sistema ocurre una "transición" : los datos de entrada son pasados por un centro de transformación y entonces se mueven en otro recorrido que

lo "secará" del sistema; el flujo en este último recorrido es llamado "eferente".

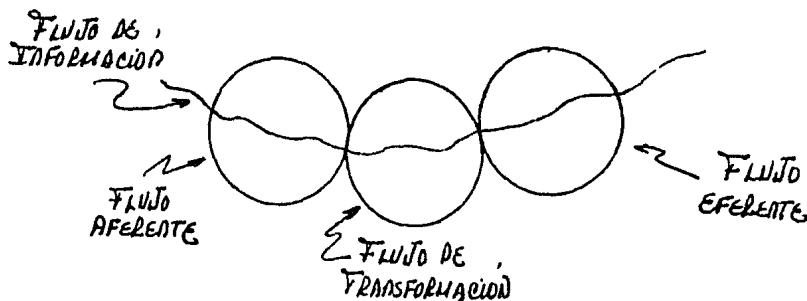


FIG 11 DIAGRAMA DE UN FLUJO DE TRANSFORMACION

Cuando un segmento de un diagrama de flujo de datos muestra estas tres características, se puede hablar de flujo de transformación.

R) El Flujo de Transacciones.

Dentro de un diagrama de flujo de datos, el flujo de información se caracteriza frecuentemente por un simple dato llamado "transacción", el cual manda el flujo a uno de muchos recorridos posibles.

Cuando en un diagrama de flujo de datos, se verifica esta situación, se puede hablar de flujo de transacciones.

El flujo de transacciones se caracteriza por el movimiento de datos a través de un "recorrido de recepción", que convierte la

información externa en una transacción; la transacción es evaluada y, dependiendo de su valor, se inicia el flujo a lo largo de uno de muchos "recorridos de acción".

El centro del flujo de información por el cual los diferentes recorridos parten, se llama "centro de transacciones".

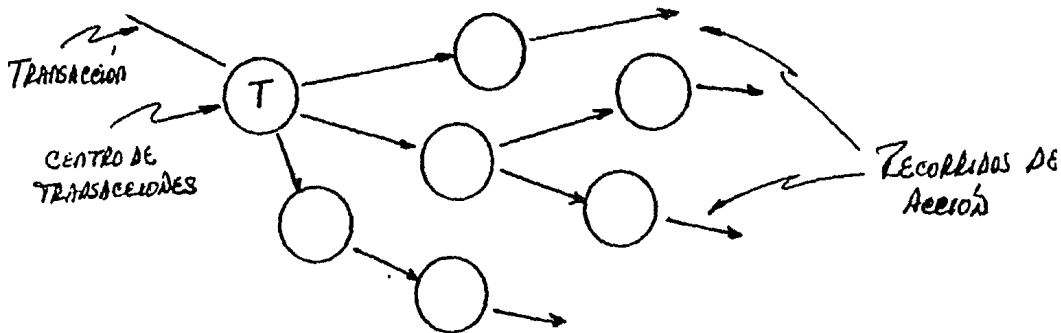


FIG 12 DIAGRAMA DE UN FLUJO DE TRANSACCION

Es importante notar que dentro de un diagrama de flujo extenso, se pueden presentar tanto el flujo de transformaciones como el de transacciones.

El diseño orientado al flujo de datos se inicia con una evaluación del diagrama de flujo de datos y se establece el tipo de flujo (transformación o transacción), además se definen los límites del flujo que delimitan la transformación o transacción central.

Basado en la ubicación de estos límites, las transformaciones (representadas por burbujas en el diagrama) son definidas como módulos en la estructura del software. La definición

precisa de estos módulos se determina factorizando la estructura y aplicando medidas de diseño y heurísticas.

2.3.1.1. EL ANALISIS DE TRANSFORMACIONES

El análisis de transformaciones es una estrategia para derivar diseños estructurales iniciales a partir de un diagrama de flujo de datos con características de flujo de transformaciones, que consiste en los siguientes cuatro pasos principales :

- Plantear el problema como una gráfica de flujo de datos.
- Identificar los elementos aferentes y eferentes.
- Hacer la factorización de primer nivel.
- Hacer la factorización de las partes aferente, eferente y la transformación de la gráfica.

1ER. PARR. Definición del problema como un Diagrama de Flujo de Datos.

No obstante que ya se habló sobre el modelo fundamental y los diagramas de flujo de datos, se detallará un poco más sobre la generación de un diagrama de flujo, porque esta actividad inicia el análisis de transformación y es necesario estudiar el flujo de datos a través del sistema.

Ahora se desea saber como el diagrama de flujo se concibe. Existen varias maneras para enfocar este proceso analítico, los diseñadores que tienen experiencia en el uso de gráficas de flujo de datos y que trabajan con lógica no enfocada a procedimientos, generalmente inician la gráfica con las entradas físicas, manejándolas a través de transformaciones sucesivas hasta obtener las salidas físicas, o trabajan el flujo de las salidas a las entradas.

Cualquiera que sea el enfoque para desarrollarlo, la cantidad de detalle que se muestre en la gráfica del flujo final varía de sistema a sistema y de diseñador a diseñador, pero es recomendable describir la mayor cantidad de detalle hasta estar seguro de que ya no se requiere mayor información.

Generalmente es más fácil formar la gráfica a través de las trayectorias de ciertas entradas primarias; las entradas y salidas menores pueden ser ignoradas inicialmente.

Se comentó también que no deben mostrarse aspectos de control o toma de decisiones explícitamente; en la gráfica debe mostrarse el proceso como una operación continua y sin fin, es decir, se debe ignorar tanto la iniciación como la terminación.

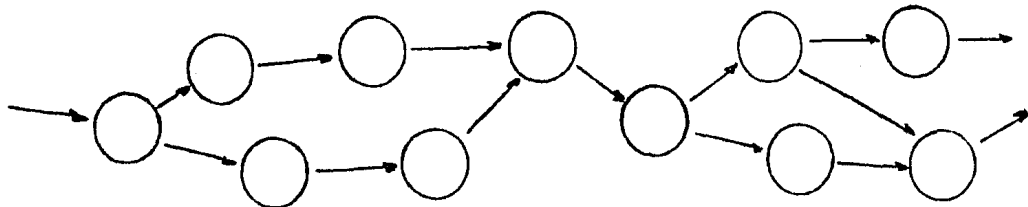


FIG 13 DIAGRAMA DE FLUJO SIN IDENTIFICACION DE ZONAS

2DO PASO. Identificar los elementos aferentes y
eferentes en la gráfica.

Es necesario definir en primer término los elementos aferentes y eferentes. Los elementos aferentes son aquellos datos de alto nivel que se encuentran lo mas alejado posible del manejo físicos en el sistema, por tanto estos elementos representan el nivel maximo de abstracción del término "entrada del sistema".

Para identificar los elementos aferentes se toma como punto de partida la entrada física del sistema y entonces se sigue la trayectoria de los datos desde la entrada hasta que se detecta una parte de dicha trayectoria que no puede ser considerada más como flujo aferente y, aunque esto representa una aspecto subjetivo para el diseñador, la idea es llegar tan lejos como sea posible.

Este proceso debe realizarse por cada trayectoria de datos de entrada.

Por otro lado, se definen en la gráfica a los elementos eferentes, es decir, aquellos datos de alto nivel, que estan lo mas alejado posible de la salida física y que pueden ser identificados como salidas del sistema.

Estos elementos pueden ser definidos como datos lógicos de salida, producidos por un proceso principal dentro del sistema y se

deben determinar, al igual que los datos aferentes, por cada una de las trayectorias de salida.

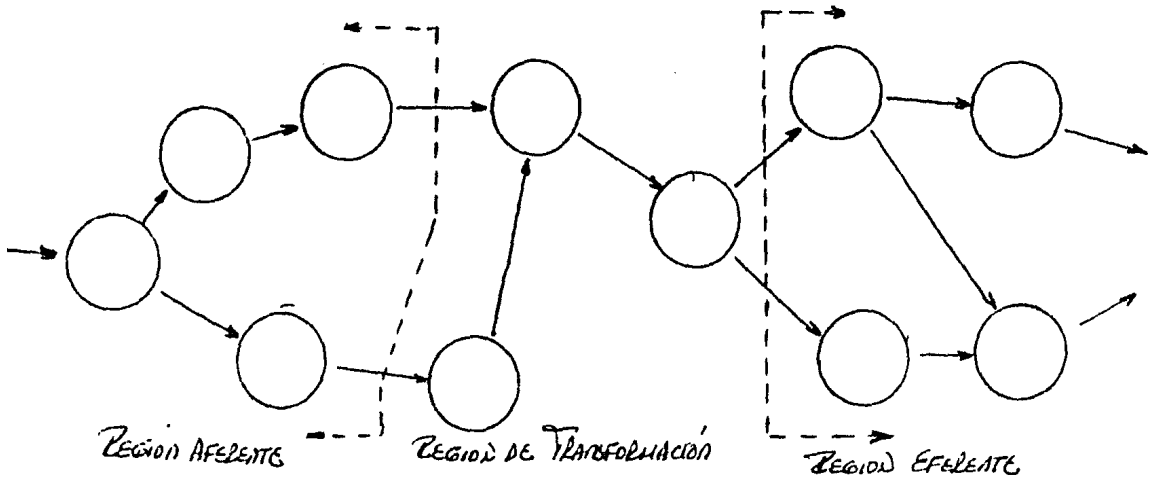


FIG 14 IDENTIFICACION DE REGIONES EN UN DIAGRAMA DE FLUJO

La definición de los elementos aferentes y eferentes usualmente deja alguna transformación en la parte media de la gráfica, que es designada como la(s) transformación(es) central(es), que realizara(n) el trabajo importante dentro del sistema y transformaran las entradas principales en las salidas principales. Algunas veces los elementos aferentes y eferentes son los mismos, en cuyo caso no existe ninguna transformación.

3ER PASO. La factorización de primer nivel.

Quando se han identificado los elementos aferentes y eferentes, se especifica un módulo principal, que al ser activado realiza el proceso completo del sistema, llamando a sus subordinados.

Por cada elemento aferente que alimenta una transformación central, se especifica un módulo aferente como subordinado inmediato al módulo principal. Estos módulos aferentes obtienen sus entradas llamado a los módulos subordinados de mas bajo nivel y transmiten esas entradas hacia arriba.

De manera similar, por cada elemento eferente que emerge de cualquier transformación central, se define un módulo eferente subordinado, que acepta al elemento eferente y lo transforma en una salida física.

Finalmente por cada transformación central o composición de transformaciones, se especifica un módulo de transformación subordinado, el cual acepta información del módulo principal y la transforma en datos correctos de salida, de donde son regresados al módulo principal.

Por tanto, podemos establecer una correspondencia uno-a-uno, entre el diagrama de flujo de datos y la factorización inicial de primer nivel.

El módulo principal representa el control global del sistema; su función es controlar y coordinar los módulos aferentes, eferentes y de transformación, que manejan los datos de mas alto nivel del sistema, es decir, llama a sus módulos aferentes (subordinados) inmediatos, para obtener las entradas principales; esta información es transferida a los módulos de transformación subordinados

correspondiente y entrega los resultados finales a los módulos eferentes que a su vez proporcionan las salidas principales.

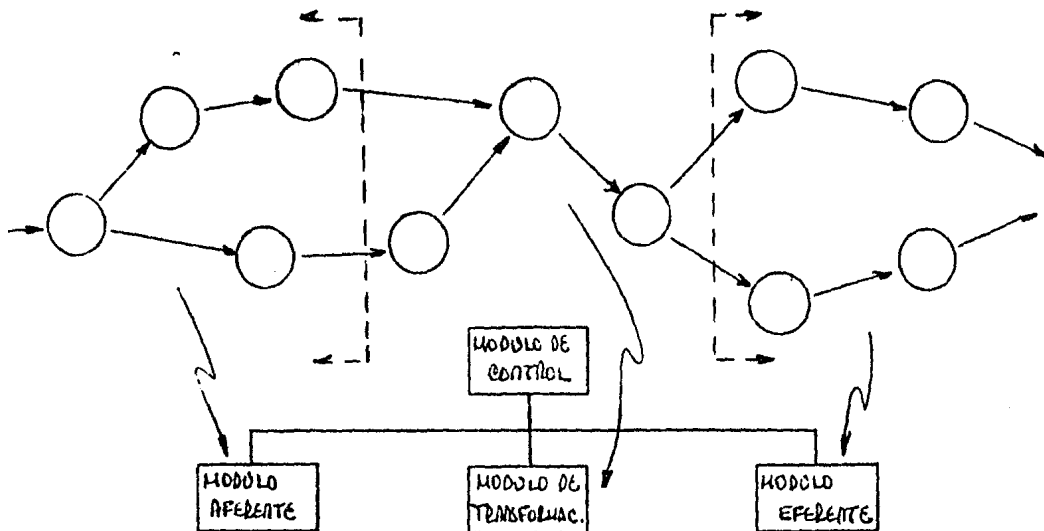


FIG 15 FACTORIZACION DE PRIMER NIVEL

4o. PASO. Factorización de grupos de elementos aferentes, eferentes y de transformación

Se utilizan tres estrategias diferentes para factorizar los 3 tipos de módulos subordinados en otros subordinados de menor nivel. No existe alguna razón particular para iniciar con la porción aferente, aunque muchos diseñadores piensan que es la manera natural para hacerlo. Tampoco es necesario factorizar completamente la porción aferente hasta el mas bajo nivel de detalle, antes de trabajar en la otra porción, aunque es importante identificar todos los subordinados inmediatos de cualquier módulo dado antes de pasar a cualquier otro.

Supongamos que hemos identificado a los módulos "Principal", "A", "T" y "E" como una factorización de primer nivel y que el siguiente paso es comenzar con la factorización del módulo "A"; también hemos definido uno de sus subordinados como "A1", entonces podemos continuar definiendo e identificando a los otros subordinados de "A" (a los que podemos llamarles "A2" y "A3") antes de explorar los subordinados de "T", "E" o "A1".

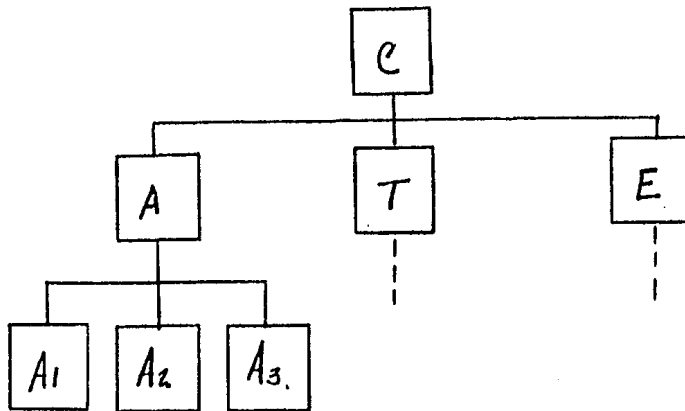


FIG 16 FACTORIZACION DE LOS MODULOS PRINCIPALES DEL SISTEMA

Para conocer como un módulo aferente puede ser factorizado, consideremos al módulo "A", cuya función, desde el punto de vista del módulo principal, es la ejecución de un proceso. La labor consiste en identificar las transformaciones o procesamientos necesarios para realizar ese proceso. Estas transformaciones son función de un nuevo módulo de transformación subordinado de manera inmediata a "A". Cada uno de estos nuevos módulos aferentes de nivel mas bajo, son factorizados recursivamente de igual forma, hasta que la última entrada física es alcanzada o el proceso termina.

La factorización de los módulos eferentes es esencialmente simétrica a la de los módulos eferentes. Para un módulo eferente dado se debe localizar la siguiente transformación que será aplicada y que llevara la información a una forma mas próxima a la salida física.

El módulo de transformación que realiza esa transformación será un subordinado al módulo eferente de mas alto nivel en el sistema.

Con respecto a la factorización óptima de los módulos de transformación, se conoce poco relativamente. Obviamente para cada transformación, se deben identificar las subtransformaciones que conforman a la transformación total o la composición de funciones que forman la transformación central en el diagrama de flujo de datos original y que son insertados como módulos intermedios, entre el nivel mas alto y las funciones de las cuales esta compuesto.

El propósito principal de esta segunda factorización, es asegurar que los subordinados del módulo principal representen los niveles mas altos de procesamiento y que los detalles menos importantes queden en los niveles inferiores.

El juicio y experiencia del diseñador son guiados durante el proceso por consideraciones importantes como el acoplamiento y la cohesión y otros heurísticos que se trataran posteriormente.

La factorización de los módulos eferentes es esencialmente simétrica a la de los módulos eferentes. Para un módulo eferente dado se debe localizar la siguiente transformación que será aplicada y que llevara la información a una forma mas próxima a la salida física.

El módulo de transformación que realiza esa transformación será un subordinado al módulo eferente de mas alto nivel en el sistema.

Con respecto a la factorización óptima de los módulos de transformación, se conoce poco relativamente. Obviamente para cada transformación, se deben identificar las subtransformaciones que conforman a la transformación total o la composición de funciones que forman la transformación central en el diagrama de flujo de datos original y que son insertados como módulos intermedios, entre el nivel mas alto y las funciones de las cuales esta compuesto.

El propósito principal de esta segunda factorización, es asegurar que los subordinados del módulo principal representen los niveles mas altos de procesamiento y que los detalles menos importantes queden en los niveles inferiores.

El juicio y experiencia del diseñador son guiados durante el proceso por consideraciones importantes como el acoplamiento y la cohesión y otros heurísticos que se trataran posteriormente.

50. PASO. Refinar la estructura del software
usando heurísticos de diseño

La estructura resultante de los pasos anteriores, puede ser refinada complementariamente, aplicando conceptos de independencia modular. Los elementos de la estructura se particionan o unen con buena cohesión y bajo acoplamiento y lo mas importante : el resultado debe ser una estructura que puede ser implementada sin dificultad, probada sin confusión y mantenida sin problemas.

Los refinamientos en la estructura se realizan en base a consideraciones prácticas y sentido común, por ejemplo, habrá ocasiones en que un controlador aferente sea totalmente innecesario; o cuando algún proceso con la información de entrada se requiera en algún módulo subordinado al controlador de la transformación, o cuando exista un alto acoplamiento debido a que los datos globales no pueden ser evitados, etc.

El juicio del diseñador y los requerimientos de software, son los que determinan finalmente el producto terminado.

El objetivo de los pasos anteriores en el análisis de transformación, es obtener una representación global del software, así, una vez que la estructura quedo definida, se puede evaluar y refinar la arquitectura del software contemplandolo en forma general. Las modificaciones hechas en el software durante este lapso, requieren poca labor adicional y pueden tener un profundo impacto en

la calidad y el mantenimiento del software.

2.3.1.2. EL ANALISIS DE TRANSACCIONES.

El flujo de información representa frecuentemente a un grupo de datos los cuales, dependiendo de su valor, provocan flujo adicional a lo largo de una o varias trayectorias definidas de antemano en el diagrama.

Este grupo de datos o información, llamado transacción, son el punto importante en un diagrama de flujo orientado a transacciones, que será transformado a una estructura de software por medio del análisis transaccional.

Los pasos para desarrollar el análisis transaccional son similares, y en casos idénticos a los del análisis de transformación.

1er. PASO. Definición del problema como un diagrama de flujo de datos.

Este paso es similar al expuesto en el análisis de transformación, con la diferencia de que aquí se habla de flujo de transacciones en lugar de flujo de transformaciones.

2o. PASO Identificar el centro de transacciones
y el flujo de cada trayectoria de acción.

La localización del centro de transacción puede ser percibido inmediatamente del diagrama de flujo de datos. Este centro se encuentra ubicado al inicio de un número de trayectorias de información que fluyen de él.

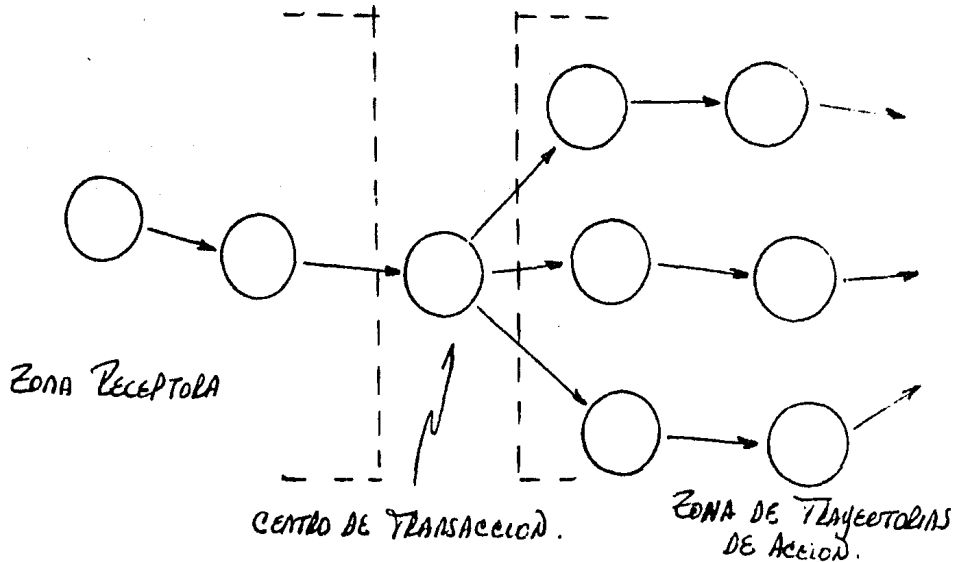


FIG 17 IDENTIFICACION DE ZONAS EN UN
DIAGRAMA DE FLUJO TRANSACCIONAL

La trayectoria receptora (es decir, la trayectoria de flujo a lo largo de la cual se recibe una transacción) y todas las trayectorias de acción deben ser aisladas y sus límites deben ser establecidos. Cada trayectoria de acción debe ser evaluada para conocer su tipo de flujo (que puede ser de transformación o transaccional).

3er PASO. Traducir el diagrama de flujo de datos en una estructura de software.

El flujo de transacción es transformado en una estructura de software que contiene una porción receptora y una porción de despacho

La estructura de la porción receptora es desarrollada de la misma manera que en el análisis de transformación. Iniciando en el "centro de transacciones", las burbujas que se encuentran ubicadas a lo largo de la trayectoria de recepción son convertidas en módulos. La estructura de la porción de envío contiene un módulo despachador que controla todos los módulos de acción subordinados. Cada trayectoria de flujo de acción del diagrama de flujo de datos, es transformada en una estructura que corresponde a sus características específicas de flujo.

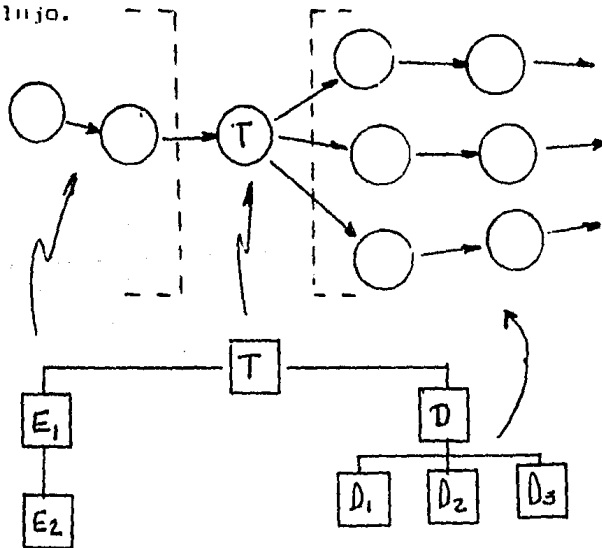


FIG 1B CONVERSION DEL DIAGRAMA DE FLUJO TRANSACCIONAL EN UNA ESTRUCTURA DE SOFTWARE

4o. PASO. Factorizar y refinar la estructura de software y la estructura de cada trayectoria de acción.

Cada trayectoria de acción tiene sus propias características. Se pueden encontrar subflujos tanto de transformación como transaccionales y la subestructura relacionada con las trayectorias de acción se desarrolla utilizando los pasos correspondientes.

5o. PASO. Refinar la estructura del software utilizando heurísticos de diseño.

Este paso del análisis transaccional es idéntico al correspondiente del análisis de transformación.

Como actividad final, el diseñador puede guiarse por los principios de cohesión, acoplamiento y heurísticos de diseño, para elaborar un diseño práctico y eficaz en su implantación y pruebas.

Una aplicación exitosa del análisis de transformación o transaccional debe ser complementado con documentación adicional que se requiere como parte del diseño preliminar. Después de que la estructura del software ha sido desarrollada y refinada, las siguientes actividades se deben realizar :

- Se debe desarrollar un proceso narrativo para cada módulo, sin ambigüedades, que describa en forma limitada las actividades de

procesamiento que ocurren dentro de cada módulo, como son las actividades más importantes, las decisiones, etc.

Este proceso narrativo sirve como una descripción de procedimiento de alto nivel que será refinada durante el diseño detallado.

- Se debe realizar una descripción de las interrelaciones de cada módulo que proporcione una lista de todos los datos que entren y salgan del módulo y en donde se incluya además, el nombre de los módulos subordinados y superordinados.

- Se deben definir estructuras de datos locales y globales, utilizando por ejemplo, diagramas jerárquicos. El detalle de estas estructuras tienen un profundo impacto sobre la estructura del software y los detalles de procesamiento de cada módulo.

- Se deben anotar todas las restricciones y limitaciones de diseño, por ejemplo, de tiempo y memoria, valores límite de cantidades en la estructura de datos y características especiales de cada módulo.

- Una vez que la documentación del diseño preliminar esté completa, se realiza una revisión para enfatizar el seguimiento de los requerimientos del software, la calidad de la estructura, la descripción de interfaces y la descripción de la estructura de datos.

2.3.2 CONCEPTOS Y HERRAMIENTAS FUNDAMENTALES DEL DISEÑO.

Los conceptos básicos en el diseño del software incluyen a la abstracción, la estructura, la modularidad y la verificación.

La abstracción es la herramienta intelectual que permite trabajar con los conceptos independientemente de la naturaleza de estos; durante la definición de los requerimientos y el diseño, la abstracción permite la separación de los elementos conceptuales de un sistema con los que será más tarde instrumentado.

Durante el diseño de la programación, la abstracción permite organizar y dirigir los procesos de pensamiento al posponer las consideraciones estructurales y algorítmicas hasta que las características funcionales y el almacenamiento de los datos hayan quedado definidos. Las consideraciones estructurales son entonces estudiadas antes de ver los aspectos de la instrumentación.

Este enfoque reduce la complejidad a la que nos enfrentamos en cualquier momento durante el diseño.

La "estructura" es una característica fundamental de los proyectos de software. El uso de una estructura permite que un sistema grande sea definido en término de unidades más pequeñas y manejables, con una clara definición de las relaciones entre las diferentes partes del sistema.

En particular, la estructura jerárquica aísla los componentes del software y promueve la facilidad del entendimiento, de instrumentación, de depuración y de pruebas, integración y modificaciones.

Por otra lado, los sistemas modulares consisten en unidades claramente definidas y manejables, con sus interrelaciones planamente identificadas entre sí.

La modularidad, ciertamente mejora la calidad del diseño, que a su vez, facilita la instrumentación, la depuración, las pruebas, la documentación y el mantenimiento de un producto de programación. Además, permite al diseñador descomponer un sistema en sus unidades funcionales, con el fin de imponer un ordenamiento jerárquico en el uso de las funciones.

Una meta fundamental en el diseño de los productos de software es la de estructurar el proyecto de tal forma que el número y la complejidad de las interacciones entre los diversos módulos sea mínima; los conceptos necesarios en esta optimización incluyen el acoplamiento y la cohesión.

La fuerza del acoplamiento entre dos módulos está influido por la complejidad de su interrelación (y se pretende que sea mínima). La modificación de un bloque común de datos o de control puede requerir de modificaciones en todas las rutinas que se encuentran acopladas a ese bloque (lo cual significa que quizá, se

En particular, la estructura jerárquica aísla los componentes del software y promueve la facilidad del entendimiento, de instrumentación, de depuración y de pruebas, integración y modificaciones.

Por otro lado, los sistemas modulares consisten en unidades claramente definidas y manejables, con sus interrelaciones plenamente identificadas entre sí.

La modularidad, ciertamente mejora la calidad del diseño, que a su vez, facilita la instrumentación, la depuración, las pruebas, la documentación y el mantenimiento de un producto de programación. Además, permite al diseñador descomponer un sistema en sus unidades funcionales, con el fin de imponer un ordenamiento jerárquico en el uso de las funciones.

Una meta fundamental en el diseño de los productos de software es la de estructurar el proyecto de tal forma que el número y la complejidad de las interacciones entre los diversos módulos sea mínima; los conceptos necesarios en esta optimización incluyen el acoplamiento y la cohesión.

La fuerza del acoplamiento entre dos módulos está influido por la complejidad de su interrelación (y se pretende que sea mínima). La modificación de un bloque común de datos o de control puede requerir de modificaciones en todas las rutinas que se encuentran acopladas a ese bloque (lo cual significa que quizá, se

tiene un grado alto de acoplamiento); por otro lado, si los módulos se comunican solamente por los parámetros y si las interfaces permanecen constantes, los detalles internos de los módulos pueden ser modificados sin tener que modificar las rutinas que usan a los módulos modificados (con lo que se consigue un grado bajo de acoplamiento).

La comunicación entre módulos puede incluir, el paso de datos, de elementos de control y de modificaciones de código de un módulo a otro. El grado de acoplamiento es menor, cuando se comunican datos, mayor cuando se comunican los conceptos de control, y mucho mayor, en el caso de módulos que modifican el código de otros módulos.

La cohesión interna de un módulo se mide en términos de la fuerza de unión de los elementos dentro del módulo; puede referirse a elementos que no tienen relación aparente (cohesión débil), hasta aquella en que todos los elementos se encuentren relacionados al desempeño de una función.

La meta de la modularización de un sistema de software, por el uso de los criterios de acoplamiento y cohesión, es la de producir sistemas, que tengan mínimo acoplamiento de datos entre módulos y, además, que cuenten con fuerte cohesión en los elementos de cada módulo.

La verificación es otro concepto fundamental en el diseño del software. Un diseño es verificable, si puede demostrarse que el

diseño generará el producto que satisface los requerimientos del cliente y se desarrolla comunmente en dos etapas :

- Verificación de que la definición de los requisitos de programación satisfacen las necesidades del usuario (verificación de requerimientos).

- Verificación de que el diseño satisface la definición de los requisitos (verificación del diseño).

Es necesario subrayar que los planes de prueba, son un producto del proceso del diseño; el plan de pruebas debe estar acoplado a las especificaciones comprobables del sistema. Se debe estar seguro de que el sistema esta estructurado para que el proceso interno pueda ser observado y probado y que los resultados tengan relacion con los requerimientos.

2.4 LA IMPLEMENTACION DE SISTEMAS EN LA INGENIERIA DE SOFTWARE.

La fase de la instrumentación o implementación del software tiene que ver con la transformación de las especificaciones de diseño a código fuente, y su objetivo principal es el de escribir el código y la documentación interna, de modo que la concordancia del código con sus especificaciones sea fácil de verificar y que se faciliten la depuración, las pruebas y las modificaciones.

Este objetivo puede alcanzarse haciendo el código fuente, tan claro y sencillo como sea posible, pues la claridad, sencillez y elegancia son los sellos de los buenos programas y la obscuridad, la ingeniosidad y la complejidad, son indicaciones de un diseño inadecuado y un pensamiento mal orientado.

La claridad del código fuente se mejora mediante técnicas de codificación estructurada, buen estilo de codificación, documentos adecuados de apoyo, buenos comentarios internos y por las características que proporcionan los lenguajes de programación modernos.

El equipo que realiza la implementación, debe contar con un conjunto de requisitos de programación bien definidos, una especificación de diseño estructural preliminar y una descripción de diseño detallado.

2.4.1. LA CODIFICACION ESTRUCTURADA.

El objetivo de la codificación estructurada es linealizar el flujo de control a través de un programa de computadora, de modo que la secuencia de ejecución siga a la secuencia en que esta escrito el código.

Si la estructura dinámica de un programa se parece a la escritura estática del texto escrito, se mejora la legibilidad del código, y esto a su vez, facilita la comprensión, depuración, prueba, documentación y modificación de programas.

La creación de la programación estructurada aparece en 1966, cuando se demuestra que la secuenciación, la selección entre alternativas de acción y la iteración forman un conjunto suficiente de construcciones para describir el flujo de control de todo algoritmo concebible, con el argumento de que una maquina de Turing puede escribirse de esa manera, y puesto que la maquina de Turing es el modelo fundamental de la computacion, se sigue que todo algoritmo puede escribirse usando secuenciación, selección e iteración.

Una crítica frecuente a este tipo de codificación, es que da como resultado una utilización ineficiente del espacio de memoria y del tiempo de ejecución, pues con frecuencia se cita la necesidad de variables auxiliares, segmentos de código repetido y un excesivo

llamado de subprogramas, aunque las ventajas son mucho mayores : mejora la claridad y legibilidad de los programas fuente y fortalece el estilo de programación.

2.4.2. EL ESTILO DE CODIFICACION.

En la programación de computadoras, el estilo de programación se manifiesta en los caminos que usa el programador para expresar una acción o un resultado deseado. Se ha reconocido que un buen estilo de codificación puede superar muchas de las deficiencias de un lenguaje de programación primitivo, mientras que un estilo pobre puede frustrar los propósitos de un lenguaje excelente. El objetivo de un buen estilo de codificación es proporcionar un código fácil de comprender, sencillo y elegante.

A continuación se mencionan algunas acciones para mejorar el estilo y otras que deben evitarse :

- Usar construcciones comunes de control. Si el lenguaje de instrumentación no proporciona construcciones para codificación estructurada, los programadores que instrumentan el sistema deben usar unos cuantos patrones de estilo; esto hará mas uniforme el estilo de codificación.

- Utilizar la estructura "goto" de manera disciplinada. En los lenguajes de programación primitivos, la propocisión "goto" puede utilizarse junto con la propocisión "if" para lograr el formato y el

efecto de las construcciones de codificación estructurada.

En particular, los usos aceptables de la propocision "goto" son casi siempre, transferencias de control hacia adelante dentro de una zona local de código. El uso del "goto" para hacer una transferencia de control hacia atrás y hacer un salto hacia adelante a una zona remota del programa, rara vez se garantiza.

- Proporcionar prólogos comunes de documentación para cada programa. Un prólogo de documentación de un programa contiene información que no resulta obvia al leer el texto fuente del programa como : autor, fecha de compilación, función realizada, rutinas que invoca, restricciones de tiempo, etc.

- Utilizar sangrías, paréntesis, espacios, líneas en blanco, y márgenes alrededor de los bloques de comentarios, para aumentar la legibilidad.

- No codificar de manera complicada.

- No anidar en forma muy profunda, pues el anidamiento excesivo también es una indicación de pensamiento confuso. Como principio general, el anidamiento de las construcciones de un programa a profundidades mayores de 3 o 4 niveles, deben evitarse.

- No se debe suboptimizar. Situación que ocurre cuando se

dedican esfuerzos desmesurados para refinar una situación que tiene poco efecto en el resultado final.

- Examinar cuidadosamente las rutinas que tengan mas de 5 parámetros formales. Dan como resultado códigos excesivamente complejos que son difíciles de mantener o usar; pocos parámetros y pocas variables globales mejoran la calidad y sencillez de los programas.

- No se debe utilizar una variable para propósitos múltiples, que podría usarse por eficiencia de memoria, ya que con esta práctica, no es posible darle un nombre significativo que sugiera su propósito, podrían aparecer códigos con baja cohesión y lo hace muy sensible a modificaciones futuras.

2.4.3. LA DOCUMENTACION DE APOYO

La programación incluye el código fuente de un sistema y todos los documentos de apoyo generados durante el análisis, diseño, implementación, pruebas y mantenimiento del sistema.

Un enfoque sistemático al desarrollo de la programación, garantiza que los documentos de apoyo se desarrollen de una manera ordenada y que se encuentren disponibles cuando se necesiten.

Estos documentos deben desarrollarse como un producto natural paralelo al proceso de desarrollo. Las necesidades y

restricciones del usuario se registran en la especificación de requisitos, los requisitos proporcionan el marco de trabajo para el diseño preliminar, el diseño detallado surge a partir del diseño preliminar, el código fuente se desarrolla a partir de los diseño preliminar y detallado. Los planes de prueba, manuales de usuario, programas de entrenamiento, instrucciones de instalación y procedimientos de mantenimiento, evolucionan a través de todo el ciclo de vida del sistema. La calidad, cantidad, vigencia y utilidad de los documentos de apoyo son las principales medidas de la bondad de un proyecto de software.

Otro aspecto importante en la documentación durante la etapa de implementación, son las notas que se originan al desarrollar unidades de código fuente, que pueden ser probadas en forma aislada por el programador que las desarrolló o modificó. Estas notas pasan de programador a programador a medida que se va transfiriendo la responsabilidad.

Finalmente, podemos mencionar a la documentación interna, que consiste en comentarios para cada programa, intercalados en la porción ejecutable del código y nombres de identificadores significativos, para denotar tipos de datos definidos por el usuario, variables, parámetros y subprogramas que pueden mejorar ampliamente la legibilidad del código fuente y minimizar la necesidad de complementos internos en la porción ejecutable de los programas.

CAP III DESARROLLO DE UN SISTEMA DE CHEQUES

3.1 DEFINICION DEL SISTEMA

La primera actividad a realizar durante la planeación de un proyecto de desarrollo de software, es la especificación en un lenguaje no técnico, de un enunciado breve del problema que se pretende solucionar y de las restricciones que existen para su solución, además del entendimiento cabal del dominio del problema y de su entorno.

Para conseguir una definición completa del sistema, se debe proporcionar en primer lugar, una definición del problema. En este caso particular, el problema se plantea a continuación :

La organización que constituye el entorno del Sistema es una Sociedad Nacional de Crédito (perteneciente a la Banca Comercial Nacional) que cuenta con oficinas foráneas de servicio al público en cada una de las capitales de los estados de la República Mexicana.

Con el fin de simplificar relaciones de información dentro de la organización, se consideran, dentro de las oficinas mencionadas, un departamento de Cheques, un departamento de Contabilidad y las Cajas de Servicio al Público, en donde se reciben las operaciones de los Clientes.

La problemática detectada en estas oficinas, se encuentra

estrechamente relacionada con las actividades de atención al público, que consisten principalmente, en el pago de Cheques, Depositos y otras actividades relacionadas.

Las transacciones que se manejan diariamente en las oficinas, se procesan mediante un sistema híbrido, es decir, un sistema cuyos componentes son, por un lado, la actividad manual del personal y por otro, la actividad de una máquina electromecánica de registro directo.

El sistema consiste en primer lugar, en el registro manual de las operaciones y la generación de reportes diarios y mensuales mediante el mismo procedimiento.

Por otro lado, los estados de cuenta que se entregan a los clientes, son actualizados diariamente en el equipo de registro directo.

Las desventajas principales del sistema descrito, son las siguientes :

- No existe un procedimiento confiable para verificar cheques con fondos insuficientes o reportados como extraviados y que son presentados para cobro.

- Las Cifras de Control (totales del día) del Departamento de Cheques que se deben consolidar con las cifras de Caja y con las

cifras de Contabilidad, son determinadas con cierto retraso, debido sobre todo, a que deben ser calculadas manualmente, retardando las actividades finales de gran parte del personal contable.

- No se puede proporcionar el saldo actualizado a los cuentahabientes en el momento que lo solicitan, pues se actualizan hasta el final del día, por lo que solamente se les proporciona el del día anterior.

- Al no actualizarse los saldos de las cuentas de cheques, pueden ocurrir omisiones al autorizarse para pago, dos o mas cheques si se utiliza el mismo saldo, en el momento de la autorización.

- La elaboración de los reportes diarios o eventuales consumen altos porcentajes de horas-hombre, que pueden ser utilizadas convenientemente en otras actividades.

- Es posible que los reportes mencionados presenten alteraciones en la información provocadas por fallas humanas, cuando se manejan volúmenes considerables de información.

- Los estados de cuenta de cheques son actualizados hasta que finalizan las operaciones normales de atención al público, por lo que no es posible proporcionar al cliente este documento, en el momento que se solicita, sino hasta el día siguiente.

- El equipo electromecánico presenta fallas constantes

debido a que es obsoleto y carece, por tanto, de un mantenimiento adecuado.

Al ser considerada esta problemática por las autoridades del Banco y tomando en cuenta la posibilidad de aumentar la infraestructura informática de la Institución, se tomo la decisión de crear un Sistema Automatizado de Cheques, planeado, analizado y diseñado a partir de las necesidades del propio Banco.

Los requerimientos que debe cumplir el Sistema que se pretende desarrollar son los siguientes :

- Debe ser posible la captura de depositos y la autorización para el pago de cheques en el momento en que se efectue la transacción.

- Se deben actualizar de inmediato los saldos de las cuentas de cheques, cuando se realiza una transacción.

- Se debe verificar la corrección del pago de cheques, verificando si estos no ha sido reportados como extraviados.

- Debe existir la consulta de saldos y ^{de} otros datos importantes cuando lo solicite el cliente.

- Debe tenerse la posibilidad de calcular las Cifras de Control del Departamento de Cheques, en el momento que se requieran,

sin necesidad de esperar al cierre de ventanillas.

- Que sea capaz de proporcionar los siguientes reportes, basado en la información que maneja :

- Reporte Diario de Cheques Pagados.
- Reporte Diario de Depositos Recibidos.
- Reporte Diario de Saldos por Clientes.
- Reporte de Cuentas por Sector Económico.
- Reporte de Cuentas por Rangos de Saldos.
- Estados de Cuentas de Cheques, mensuales o a solicitud del Cliente.

La justificación para el desarrollo del sistema, es decir, la determinación de que resulta apropiada una solución computarizada para abatir la problemática mencionada, se consigue al considerar el concepto "costo", pues proporcionará los mismos servicios y la misma información que el sistema que se pretende substituir, utilizando para ello una menor cantidad de tiempo. Además el sistema queda justificado socialmente, por no provocar desplazamiento de empleados, sino al contrario, contribuye al correcto desempeño de sus funciones.

El sistema será desarrollado en la Dirección de Informática del propio Banco, es decir, se utilizará equipo y personal de la propia Institución y no se contratará ningún servicio externo.

Además del equipo humano para el desarrollo del proyecto, se cuenta con los recursos materiales necesarios, que configuran el hardware del Sistema, con que contará cada Oficina de Atención al Público y que consiste en :

- Un microcomputador con las siguientes características :
 - C.P.U. con procesador 8086 (16 bits de long.)
 - Disco fijo con 20 megabytes de capacidad.
 - Unidad de cinta magnetica de 10 megabytes de capacidad
 - Funciona como intérprete; no tiene compilador del lenguaje de implementación : Basic
- Tres terminales con pantalla monocromática, conectadas en serie al computador.
- Una impresora con velocidad de 300 líneas por minuto, conectada en paralelo al computador.

Los recursos humanos de que se dispone es el personal de los Departamentos de Planeación, Analisis, Diseño y Programación, por lo que el proyecto se considera exclusivamente interno y los costos extraordinarios que pudiesen aparecer, se reducen a los aspectos de

implantación y mantenimiento.

Los objetivos que se pretenden alcanzar al desarrollar el proyecto, además de un ahorro considerable en tiempo, son :

- Se aumentará el nivel de presentación en los reportes que se generen, así como en los Estados de Cuenta.

- Se reducirá el tiempo que el cliente debe permanecer en la oficina.

-Se proporcionará al Cliente información debidamente actualizada.

-Se aumentará el nivel de confiabilidad en el manejo de la información, por parte de los empleados y de la Gerencia.

En las secciones siguientes, se continua el análisis del problema y se realiza el diseño del sistema.

3.2 ANALISIS DE LOS REQUERIMIENTOS DEL SISTEMA

En base a la situación que se expuso en la sección anterior, sobre la necesidad de desarrollar un sistema automatizado para reducir la problemática de las Oficinas Foráneas del Banco en sus áreas de 'Atención al Público', se determinaron los requerimientos que debía cumplir el Sistema, es decir, las funciones principales que debe realizar.

En esta sección se realiza el análisis de tales requerimientos para conseguir, tomando como base la definición inicial del Sistema, una especificación concreta, que servirá como base fundamental para el desarrollo del software que se tiene en mente.

Se integra también una descripción detallada del problema que el software debe solucionar, utilizando para ello, la documentación del flujo y la estructura de la información que manejará el Sistema.

3.2.1 DESCRIPCION DE LA INFORMACION

Con la información que se tiene de la Definición del Sistema, es posible iniciar la descripción del sistema, utilizando para ello, los diagramas de flujo de datos.

Se comenzará por presentar la gráfica de lo que se llama el modelo fundamental del Sistema y que consiste en describirlo en forma general, indicando solamente los orígenes y los destinos de los datos.

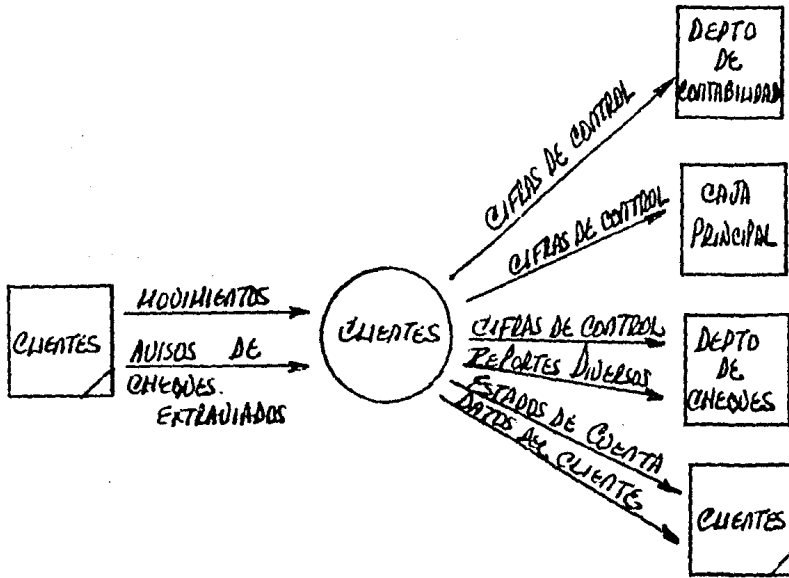


FIG 19 MODELO FUNDAMENTAL DEL SISTEMA DE CHEQUES

A partir de este primer diagrama, se inicia la descripción del Sistema y para ello, se deberán determinar sus componentes primarios : procesos, almacenamientos de datos y el flujo de los mismos.

Se menciona que el Sistema que se desea desarrollar, tiene como objetivos principales, satisfacer las demandas de la clientela, en lo que se refiere a la atención que se les brinda y mejorar el

nivel operativo del personal que maneja el Departamento de Cheques; por tanto, los procesos principales del Sistema quedan directamente definidos :

- Registro de Operaciones Diarias.
- Consulta de Saldos e Información de Clientes
- Generación de Estados de Cuenta y Reportes Diversos.
- Cálculo de Cifras de Control.

Asimismo, se deben considerar dos funciones complementarias que tienen como objetivo, la alimentación de la información básica :

- Registro de Clientes.
- Registro de Cheques extraviados.

Estos seis procesos se consideran como las seis funciones principales que desarrollará el Sistema.

A continuación se debe analizar lo que en un diagrama de flujo de datos se llama "Almacenamiento de Datos", que se refiere a los datos que deben ser mantenidos en algún medio de almacenamiento (como cintas, discos, o algún otro medio), ya que serán requeridos por algún proceso durante la operación del Sistema.

De la definición del problema se concluye la existencia de los siguientes almacenamientos :

- Movimiento Diario. Es la información base del Sistema y servirá para calcular las cifras de control del día que transcurre.

- Movimiento Mensual. Es el movimiento diario que se acumula en forma mensual y que servirá para generar los Estados de Cuenta del mes que transcurre.

- Cheques Extraviados. Es la información básica para detectar pagos de cheques que no proceden durante el registro diario de movimiento.

- Clientes. Es la información básica del Sistema; registra los datos principales de los cuentahabientes para consultas posteriores.

- Técnicamente, también se consideran los Reportes y los Estados de Cuenta, como otro tipo de Almacenamientos.

Por último, es necesario identificar los flujos de datos del Sistema, es decir, aquellos datos que se mueven a través del Sistema; por el momento, de la definición del problema, se pueden identificar los siguientes :

- Movimientos Diarios.
- Cheques extraviados.
- Cifras de Control.
- Datos del Cliente.
- Estados de Cuenta y
- Reportes Varios.

Como ya se tiene una lista más o menos completa de los componentes que integran el Sistema, es posible definir un segundo diagrama de flujo, que ilustre el avance que se lleva en el análisis del Sistema.

El diagrama se muestra a continuación :

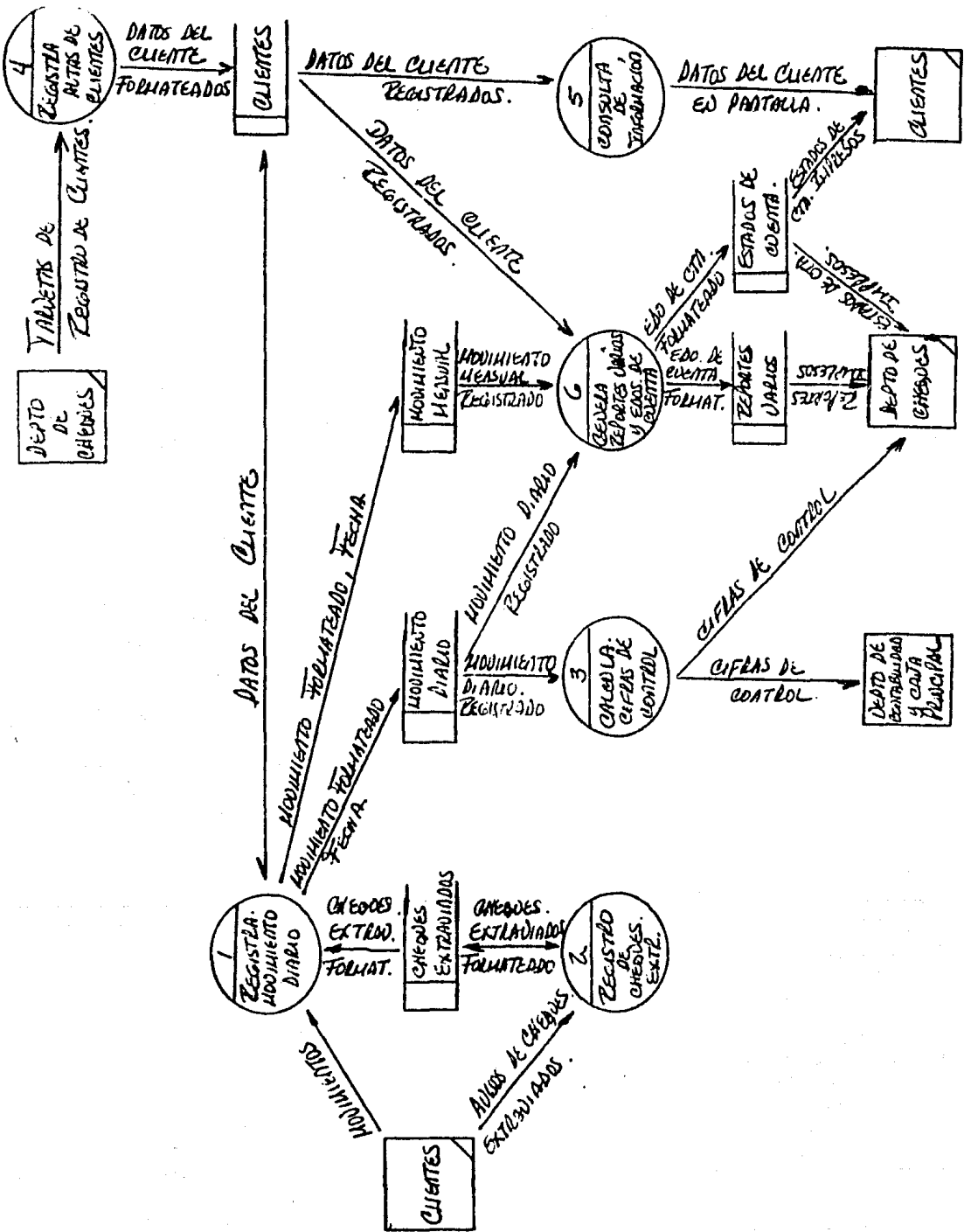


FIG 19 DIAGRAMA QUE DESCRIBE LOS ORIGENES, DESTINOS, ALMACENAMIENTOS Y FLUJOS DEL SISTEMA

Es importante notar, la relación que se establece en el Sistema entre el departamento de Contabilidad, el departamento de Cheques y la Caja Principal, lo cual representa uno de los requerimientos que se deben cumplir.

Por otra parte, al revisar el diagrama de flujo, se pueden notar las relaciones de tiempo que existen en algunos procesos, por ejemplo, el Registro de Movimientos, parece ser continuo, mientras que la Generación de Reportes se realiza al finalizar las operaciones o de manera eventual.

Esta particularidad será de utilidad más adelante al establecer el criterio, en el diseño, sobre la manera más provechosa de manejar la información.

En este momento, el análisis puede continuar si se considera el flujo de datos que se describe en el diagrama de la Fig 19, es decir, se debe definir el flujo y el almacenamiento de datos, que existe bajo este primer nivel elemental, pues en general, no todos pueden ser extraídos directamente de la descripción del problema.

Al quedar establecida la problemática de las Oficinas Foráneas del Rancho, la situación se describió de manera general y por tanto, para continuar nuestro análisis se deben establecer canales de comunicación con el personal que trabaja directamente en las oficinas.

A través de comunicación oral, escrita y documentos proporcionados por estos usuarios potenciales del Sistema, se obtiene la siguiente información, que complementa la reunida hasta ahora.

Al estudiar físicamente un Estado de Cuenta se observaron los siguientes datos :

- Núm. de Cuenta.

- Nombre del Cliente

- Dirección

- Población

- Saldo del Mes Anterior

- Movimientos que se realizaron en el Mes

- Saldo del Mes que transcurre

- Saldo Promedio

Los Reportes Diarios que se mencionaron durante la definición del problema y que aparecen en el diagrama de flujo de datos contienen la siguiente información :

- Reporte Diario de Saldos por Cliente. Se detalla Núm. de Cuenta, Nombre del Cliente, Saldo del Cliente y se imprime un total de saldos al final del reporte.

- Reporte de Cuentas por Rangos de Saldos. Muestra el No. de Cuentas y el importe total de grupos de cuentas, seleccionados por el usuario.

- Reporte de Cuentas por Sector Económico. Muestra el No. de Cuentas y el importe global de cada uno de los siguientes sectores económicos :

- Sector "01" de personas físicas.
- Sector "02" de personas morales o empresas
- Sector "03" de instituciones de gobierno

Se revisaron los cheques que se presentan a cobro y se observaron los datos : Núm. de Cuenta, Num de Cheque e Importe de Cheque; estos datos también se encuentran en los Avisos de Cheques Extraviados.

Se revisaron también las fichas de depósito y se observaron los datos : Núm. de Cuenta e Importe de Depósito.

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

También se observó que los datos que solicitan los Clientes son : Saldo Disponible, Fecha de Ultimo Movimiento y Estados de Cuenta.

Las Cifras de Control se calculan, utilizando el importe total de cheques recibidos y el importe total de depósitos recibidos.

Una de las actividades importantes durante el análisis, es la documentación de la información que se ha ido recopilando en este proceso, y debe realizarse de la manera más completa posible, puesto que será de utilidad dentro del mismo análisis o durante el diseño.

Se puede alcanzar tal propósito mediante la creación de un Diccionario de Datos. En el se indicará la definición, estructura y uso de cada unidad elemental que utilice el Sistema, y servirá al mismo tiempo, para normalizar la información que pudiesen manejar al mismo tiempo, diferentes personas dentro del equipo de desarrollo.

En el apendice "A" se puede encontrar un Diccionario de Datos para el sistema que se desarrolla en esta tesis.

Con el Diccionario de Datos y el Diagrama de Flujo de Datos que se construyó de la definición del problema se deben investigar las fuentes de cada dato, es decir, se deben localizar los orígenes de los datos dentro del Sistema, con la ayuda de Usuarios y todos los medios al alcance.

Cuando se ha llegado a un nivel de detalle satisfactorio, es posible describir los componentes del diagrama de flujo, de manera mas completa; a continuación se describe esta información.

1. Fuentes y Destinos de Información.

- Clientes
- Departamento de Cheques.
- Departamento de Contabilidad.
- Caja Principal.

2. Procesos o Funciones Principales.

- Registra Movimientos.
- Registra Clientes.
- Registra Cheques Extraviados.

3. Almacenamientos de Datos.

- Movimiento Dierio.
- Núm. de Cuenta.
- Importe de Movimiento.
- Núm. de Cheque o Núm. de Depósito.

- Movimiento Mensual.

- Núm. de Cuenta.

- Importe de Movimiento.

- Núm. de Cheque o de Depósito.

- Fecha de Movimiento. *

- Cheques Extraviados.

- Núm. de Cuenta.

- Núm. de Cheque.

- Importe de Cheque.

- Clientes.

- Núm. de Cuenta.

- Nombre.

- Dirección.

- Población.

- Saldo Anterior.

- Saldo Actual.

- Sector Económico.

- Estados de Cuenta.

- Núm. de Cuenta.
- Nombre del Cliente.
- Dirección.
- Población.
- Saldo Anterior.
- Saldo Actual.
- Núm. de Cheque o de Depósito.
- Importe de Movimiento.
- Fecha de Movimiento.

- Reportes varios.

- Datos contenidos en almacenamiento de Clientes.

El contar con la información anterior, permite analizar el diagrama de flujo a través de una descomposición funcional, es decir, por medio del refinamiento de cada proceso en sus subfunciones correspondientes.

Estas nuevas funciones de menor nivel, se representan como nuevos procesos en el diagrama de flujo.

Consideremos en primer término, el proceso "Registra Movimiento" que se ilustra en el siguiente diagrama de flujo.

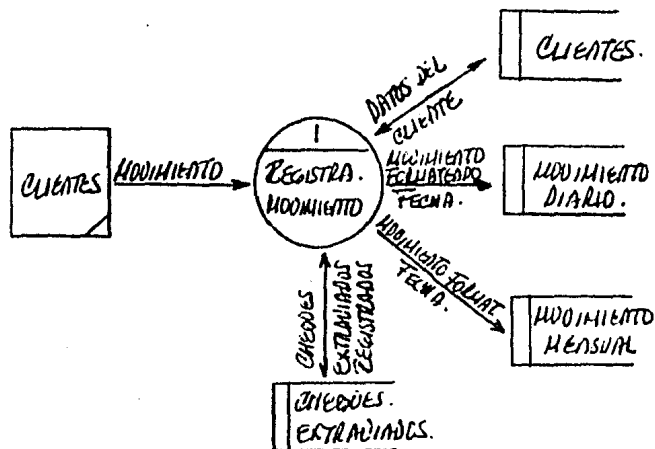


FIG 20 DIAGRAMA DE FLUJO DEL PROCESO "REGISTRA MOVIMIENTO.

Al realizar un análisis complementario del proceso, se identifican las siguientes sub-funciones :

- Recibe Datos del Movimiento.
- Verifica los datos del Movimiento.
- Actualiza los Datos del Cliente.
- Registra Movimiento en almacenamiento "Movimiento Diario"
- Registra Movimiento en almacenamiento "Movimiento Mensual"

Estos cinco sub-procesos pueden ser representados graficamente por medio de un diagrama de flujo de datos parcial, en donde se muestra solo el proceso "Registra Movimiento". El diagrama se muestra a continuación.

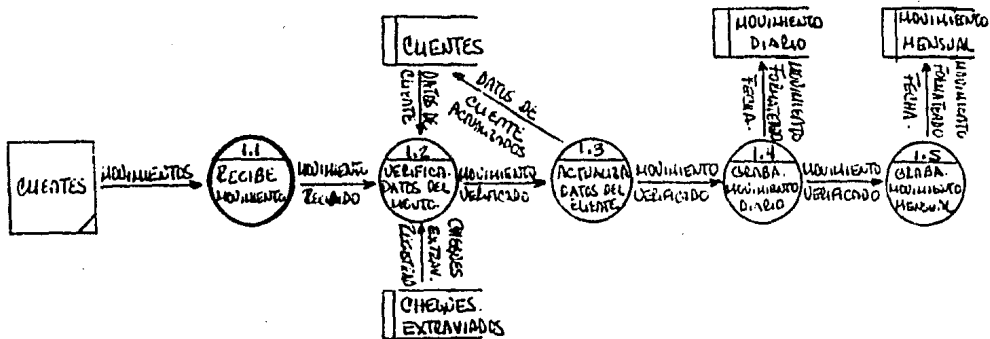


FIG 21 EL PROCESO "REGISTRA MOVIMIENTO" YA REFINADO.

Análogamente se puede manejar el proceso "Genera Reportes Varios y Estados de Cuenta". Este proceso agrupa una serie de sub-funciones que corresponden a los diferentes reportes que el Sistema debe generar. Este nuevo refinamiento se presenta gráficamente en el siguiente Diagrama.

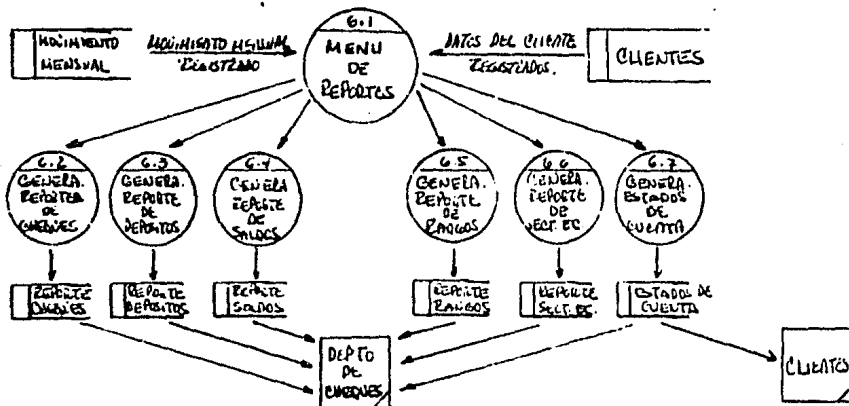


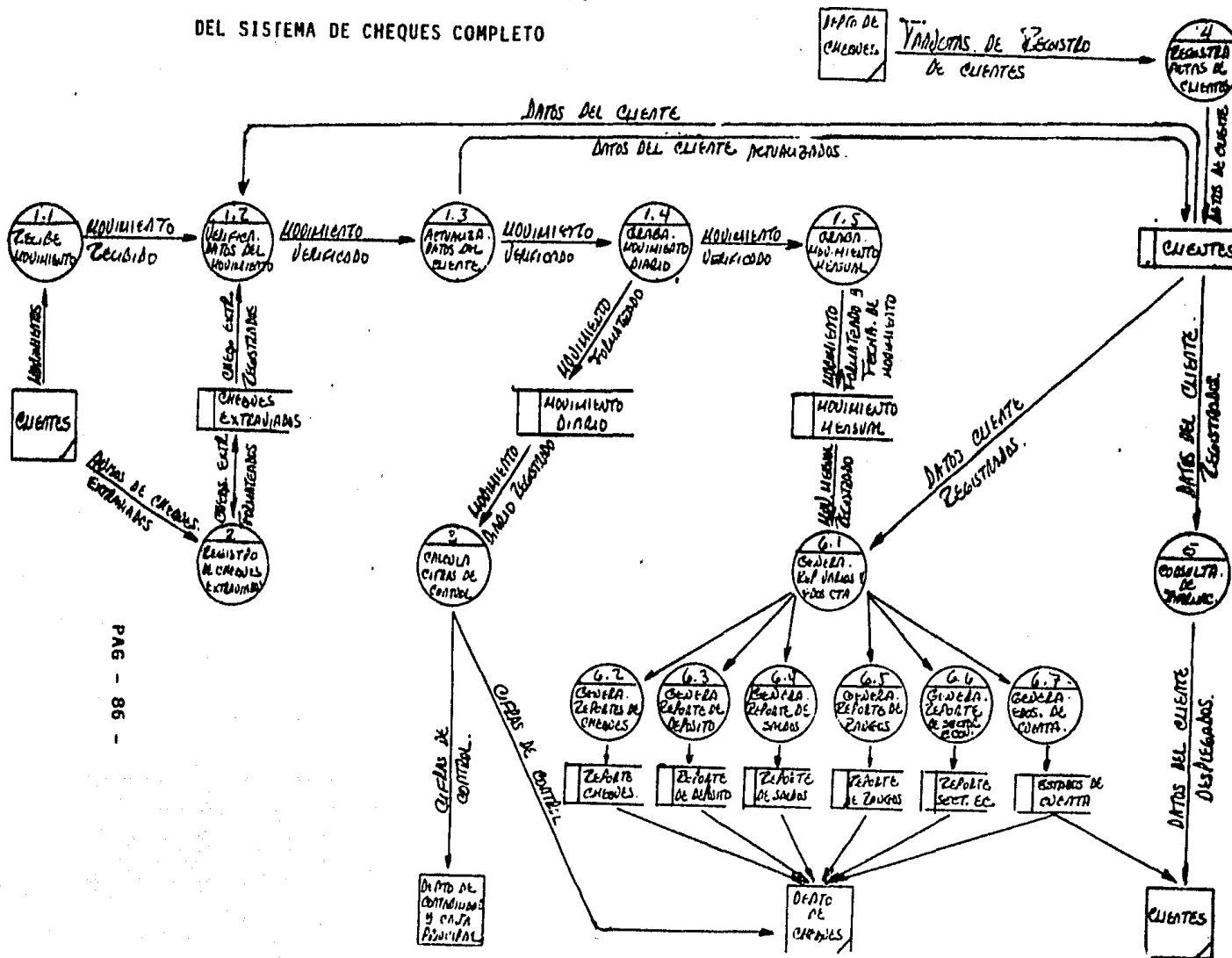
FIG 22 EL PROCESO "GENERA REPORTES VARIOS Y ESTADOS DE CUENTA YA REFINADO

Los procesos principales restantes, no requieren de refinamiento, como en los casos anteriores, por ser funciones de menor complejidad.

Los refinamiento de los procesos mencionados, se representan a su vez en el diagrama de flujo de datos general, el cual representa al Sistema en su totalidad. De esta manera se tiene una versión nueva con un nivel de detalle mayor (Ver Diagrama 23).

Con este último diagrama, se puede dar por concluido el análisis del proyecto. Tanto el Diccionario de Datos como los Diagramas de Flujo que se desarrollaron, podran utilizarse, por un lado, como herramientas de comunicación, proporcionando a Usuarios y Gerentes Involucrados una mejor comprensión del punto de vista del Analista, y por otro, como valioso soporte par el futuro desarrollo del Sistema.

FIG. 23 DIAGRAMA DE FLUJO DE DATOS REFINADO
DEL SISTEMA DE CHEQUES COMPLETO



3.3 DISEÑO PRELIMINAR DEL SISTEMA

Con base en la información generada durante la etapa del análisis del Sistema (en donde se crea un Diagrama de Flujo general del Sistema y se inicia un Diccionario de Datos), se inician las actividades de diseño del proyecto.

En esta sección se deriva una estructura inicial del software a partir del diagrama de flujo de datos, para refinarlo posteriormente, y se identifica cada módulo componente y sus interrelaciones dentro del Sistema.

Para desarrollar una estructura inicial de software, se utiliza como base el diagrama de flujo de datos, de manera que se pueda aplicar el análisis de transformación o transaccional (según se requiera) que se describieron en la sección teórica correspondiente.

Del diagrama de flujo general, es posible observar algunos procesos que pueden ser agrupados, por tener los mismos requerimientos de ejecución en relación al factor "tiempo", es decir, aquellos procesos que se realizan simultáneamente o que pueden ser agrupados para ser ejecutados de esa manera.

Consideremos en primer lugar los procesos 1.1 a 1.5 que forman parte del diagrama principal. Se observa que estos procesos ocurren continuamente, y en particular, el proceso 1.1 "Recibe Movimiento" debe ser considerado como un proceso en línea.

No existe inconveniente para que los procesos 1.2 a 1.5 se realicen de manera simultanea, cada vez que se recibe un movimiento. Podemos por tanto, agruparlos dentro de un macroproceso, al que podemos llamar "Registro de Movimientos".

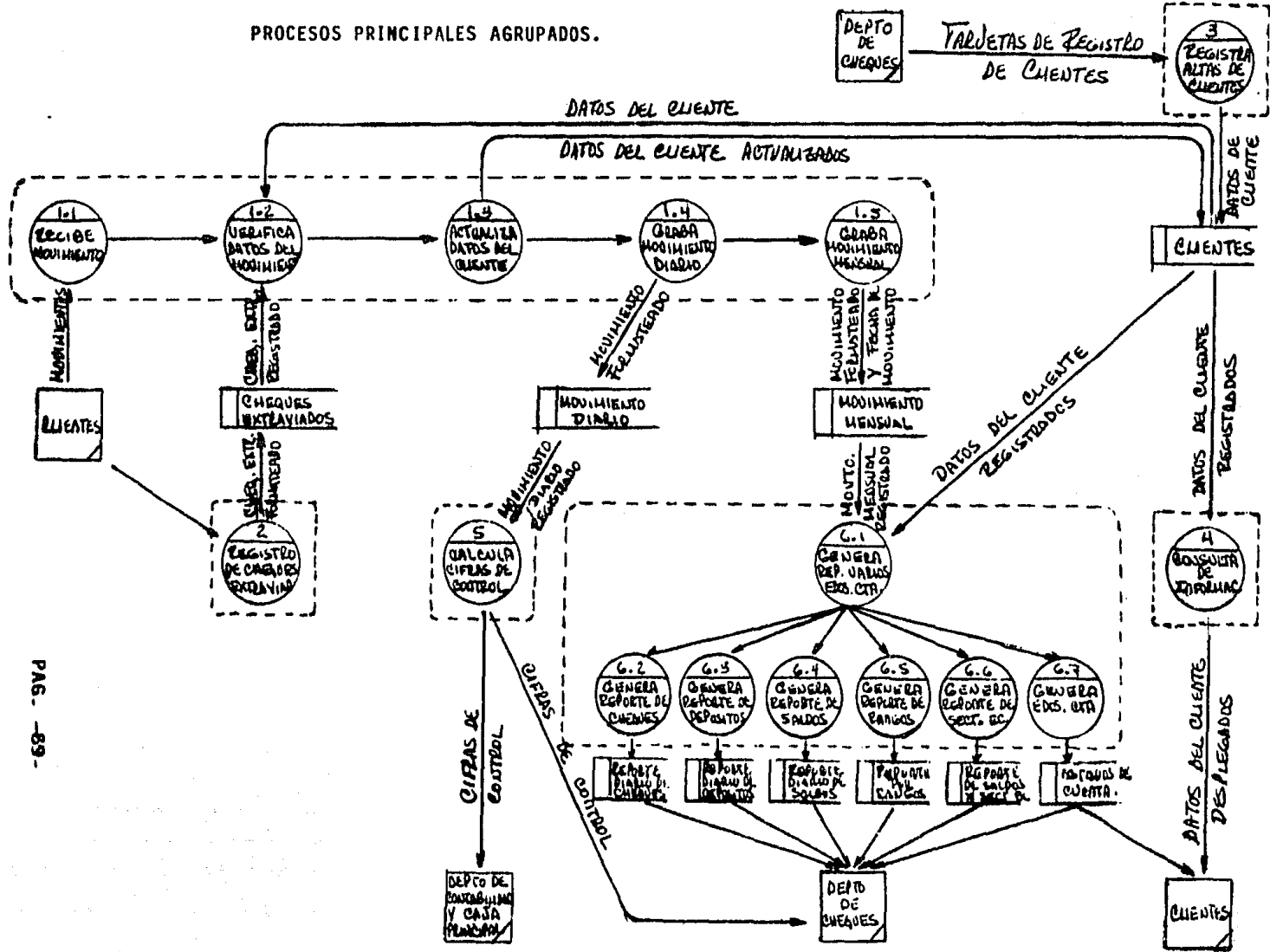
De igual forma, se pueden considerar los procesos 6.1 a 6.7 que forman un agrupamiento de Reportes, los cuales se procesan normalmente al finalizar las actividades del dia.

Estos procesos pueden ser operados en su totalidad, en modo "batch", y por lo tanto, tambien existe la posibilidad de aislar este grupo, que podemos identificar como "Generación de Reportes".

Los procesos señalados con los numeros 2,3,4 y 5 se deben considerar de manera independiente, porque su ejecución es eventual y por ende, no existe la posibilidad de agruparlos funcionalmente (Ver diagrama 24).

Con la configuración del Sistema definida de esta manera, podemos generar ya, una estructura inicial del software. Esta estructura estará formada por un Módulo Principal que realice la función de controlador del Sistema completo, y seis módulos subordinados (4 de ellos de control tambien) que representan a los seis procesos principales que efectuará el Sistema (Diagrama 25).

FIG 24 DIAGRAMA DE FLUJO DEL SISTEMA CON SUS PROCESOS PRINCIPALES AGRUPADOS.



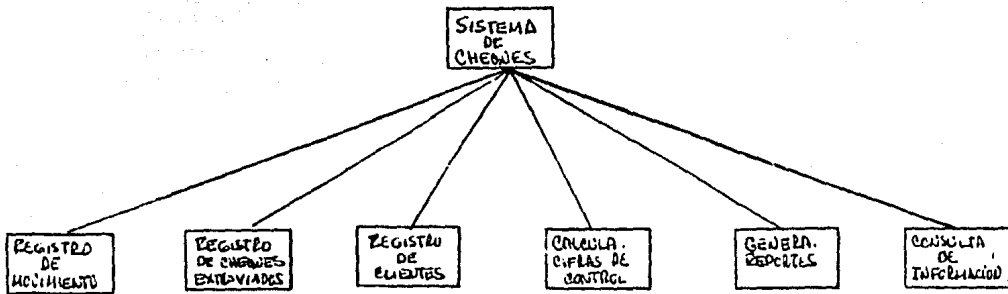


FIG 25 ESTRUCTURA INICIAL DEL SOFTWARE

Consideremos en primer lugar, el subdiagrama que contiene los procesos 1.1 a 1.5) entonces se identifican en él las regiones aferente y de transformación (por la naturaleza del proceso no existe propiamente una zona eferente)

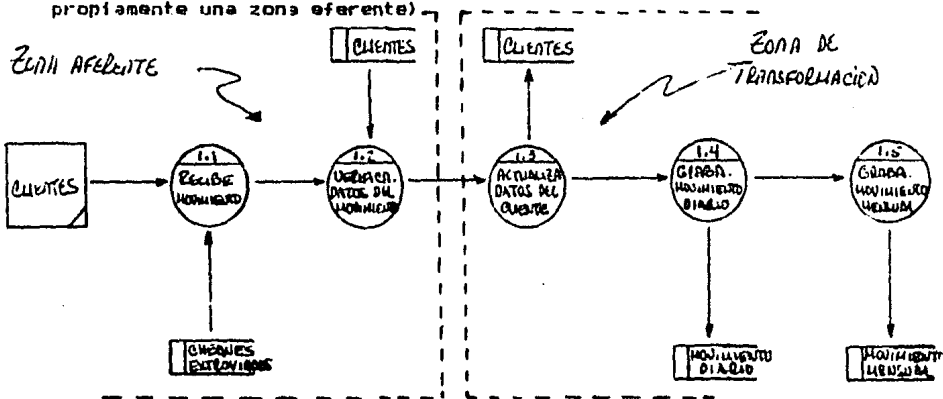


FIG 26 IDENTIFICACION DE LAS ZONAS AFERENTE Y DE TRANSFORMACION EN EL SUBDIAGRAMA DE FLUJO DE LOS PROCESOS 1.1 A 1.5

De este subdiagrama se deriva una segunda estructura de software. Del módulo de control que designamos como "Registro de Movimientos" se originan dos módulos subordinados a los que se designaran como "Recibe Movimiento" (modulo aferente) y "Actualiza Movimiento" (modulo de transformación) respectivamente.

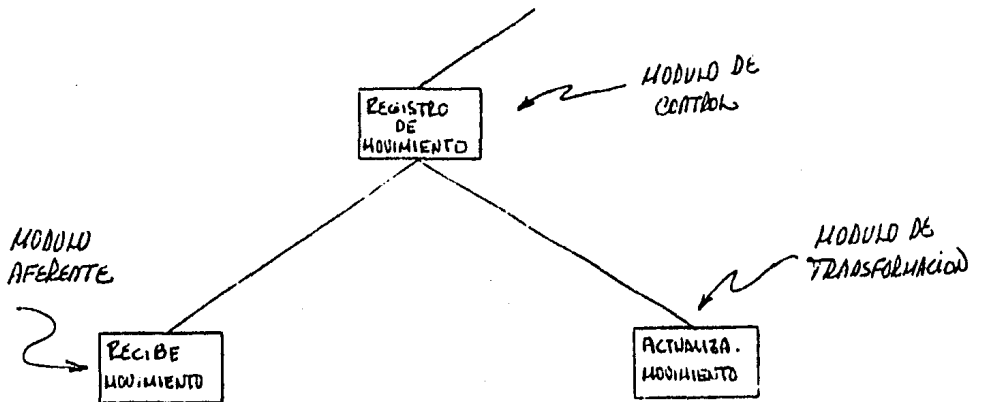


FIG 26 ESTRUCTURA INICIAL DE SOFTWARE DEL PROCESO PRINCIPAL DEL SISTEMA DE CHEQUES.

El módulo "Recibe Movimiento" queda definido al añadirsele los módulos subordinados que se identifican con los procesos 1.1 y 1.2 en el diagrama de flujo de datos; estos módulos reciben el nombre de "Acepta Movimiento" y "Valida Movimiento", respectivamente y realizarán las subfunciones que sus nombres indican.

Análogamente, el módulo "Actualiza Movimiento" queda totalmente factorizado al añadirsele los módulos que corresponden a los procesos 1.3, 1.4, y 1.5 cuyos nombre son "Actualiza Datos del

Ciente", "Registra Movimiento Mensual" y "Registra Movimiento Diario", respectivamente.

La factorización de la primera función principal del Sistema queda entonces definida; se observa que existe una correspondencia entre el subdiagrama de flujo de datos y la subestructura jerárquica.

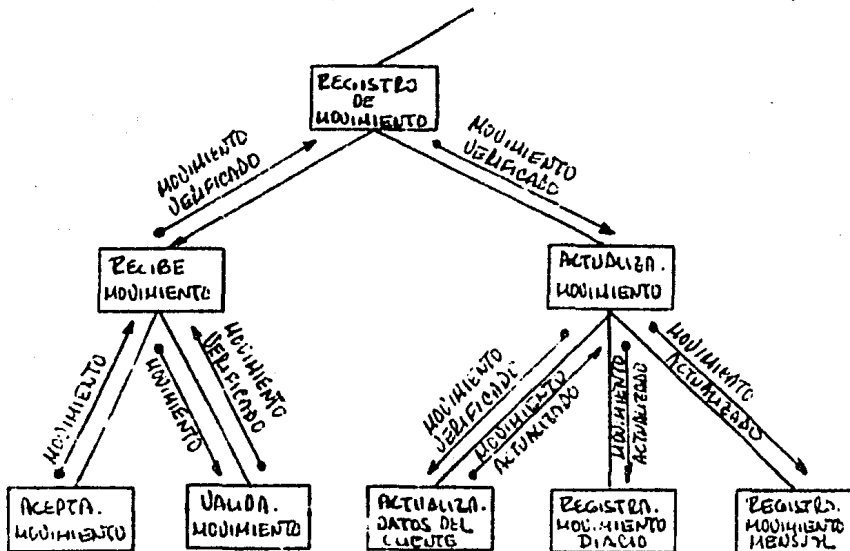


FIG 27 ESTRUCTURA DE SOFTWARE DE LA PRIMERA FUNCIÓN DEL SISTEMA YA REFINADA.

Una característica importante de las subestructuras de software que se desarrollan en esta etapa de diseño, es la especificación de la información que se comparte entre los módulos; los nombres de los datos de entrada y salida están anotados junto a las líneas que enlazan los módulos.

Aunque estas anotaciones no se deben emplear con fines de documentación, ya que su objetivo es meramente informativo (es engorroso actualizar la notación de estas estructuras) proporcionan un método para rastrear el flujo de datos en una estructura de software.

Cada uno de los módulos que se illustren en este trabajo, serán descritos complementariamente con ayuda de las tarjetas "IPO", en donde se indica :

- El sistema al que pertenecen.
- El autor de la información.
- El nombre del módulo.
- La fecha en que se detalla.
- La manera en que esta interrelacionado con sus módulos supra u subordinados.

Durante la etapa del diseño preliminar estos datos serán vaciados a su respectivas tarjetas "IPO", conforme la estructura del software se define. En el apéndice "R" se describen las tarjetas de los módulos que componen el Sistema. Otros conceptos que aparecen en las tarjetas que componen este apéndice serán explicados en su oportunidad.

Las tarjetas "IPO", el Diccionario de Datos y otras herramientas complementan poco a poco la documentación del Sistema y constituyen una de las ayudas mas importantes en revisiones futuras o posibles mejoras y ampliaciones.

La segunda función principal del Sistema que se diseña, es la que se encuentra integrada por los procesos 6.1 a 6.7 que componen la "Generación de Reportes". Este diagrama tiene características de un flujo de datos transaccional (por su forma especial), por tanto, en este caso, se utilizará el análisis transaccional para generar una subestructura de software adecuado.

En primer término se definen las regiones "Receptora" y de "Trayectorias de Acción" y su "Centro de Transacciones"; estos conceptos forman los elementos básicos iniciales para desarrollar el análisis.

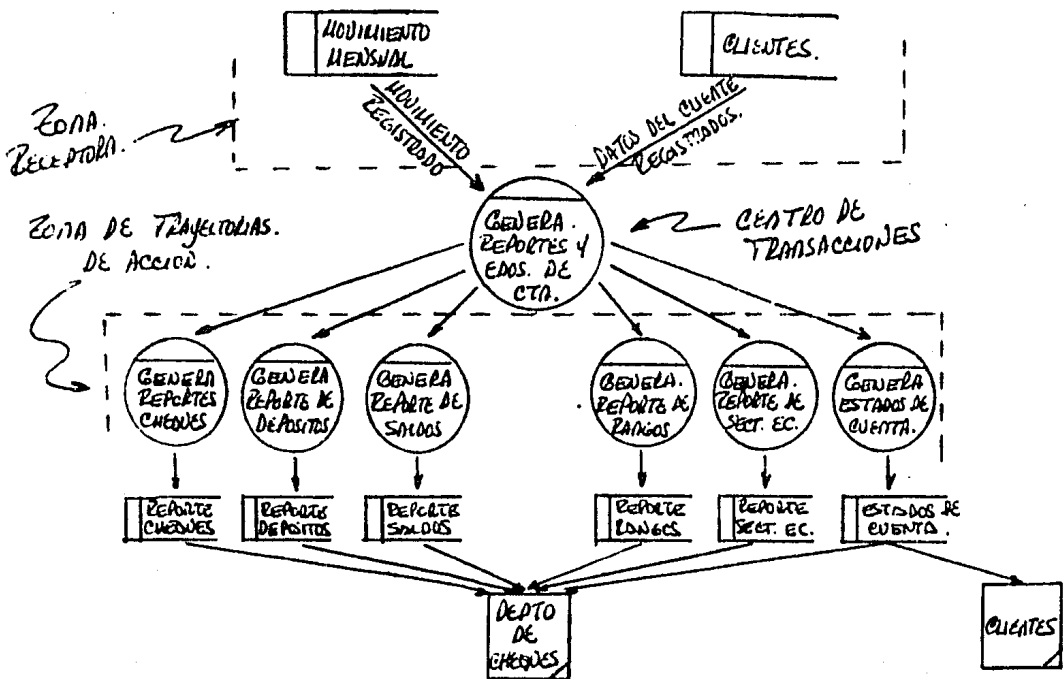


FIG 28 DEFINICION DE LOS ELEMENTOS DEL DIAGRAMA DE FLUJO TRANSACCIONAL

Del módulo subordinado "Generación de Reportes", que se definió con anterioridad, se generan dos módulos subordinados a este.

El primero de ellos es el módulo de "Envío", que se identifica con la porción receptora del diagrama de flujo y se denominará "Menú". El segundo se corresponde con el Centro de Transacciones, que se muestra en el diagrama con el proceso 6.1; a este módulo se le llamará "Despachador".

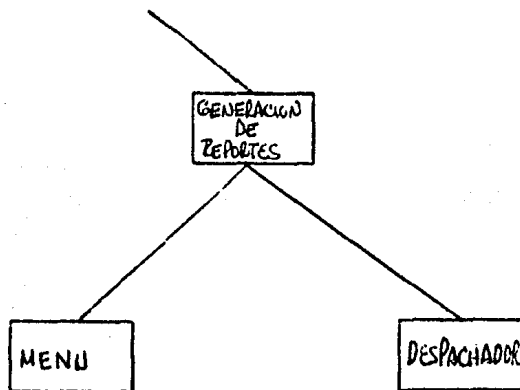


FIG 29. ESTRUCTURA INICIAL DE SOFTWARE
PARA LA 6A. FUNCION PRINCIPAL

El módulo "Despachador" queda definido al añadirsele los módulos subordinados correspondientes a los procesos 6.2 a 6.7 cuyos nombres son : "Genera Reporte Diario de Cheques Pagados", "Genera Reporte Diario de Depositos Recibidos", "Genera Reporte Diario de Saldos por Cuenta", "Genera Reporte de Cuentas por Sector Económico", "Reporte de Cuentas por Rangos de Saldos" e "Imprime Estados de Cuenta", respectivamente.

De esta forma, queda definida una segunda subestructura de software que corresponde a otra función primaria del Sistema: La Generación de Reportes.

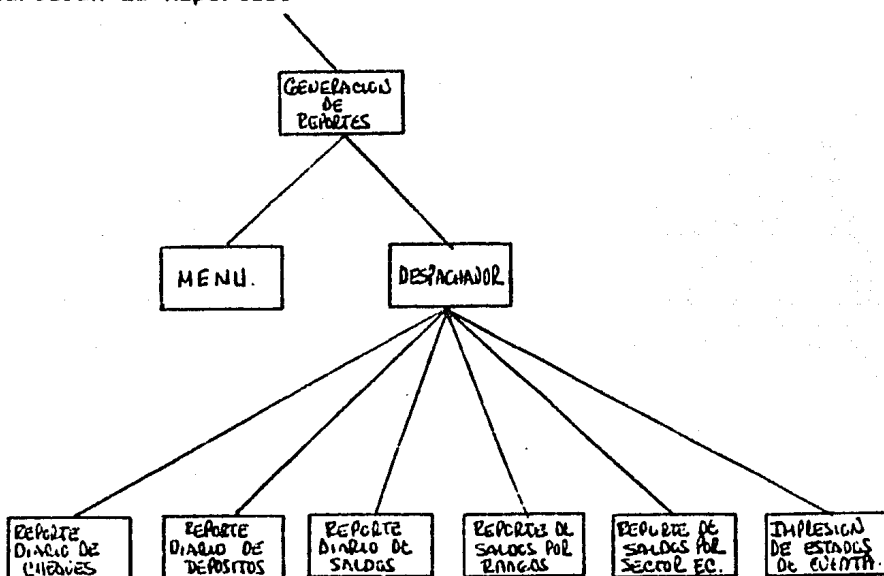


FIG 30 ESTRUCTURA FINAL DE SOFTWARE PARA EL
60. PROCESO PRINCIPAL DEL SISTEMA

Como se comentó, estos nuevos módulos tendrán sus respectivas cartas "IPO", que servirán como su identificación dentro del Sistema general.

Los procesos restantes 2,3,4 y 5 se definen como procesos independientes (se comentó la razón con anterioridad). Los módulos "Registro de Cheques Extraviados" y "Registro de Altas a Clientes", se estructuran con dos módulos subordinados cada uno. Los primeros tienen como función recibir la información y validarla; los segundos efectúan el proceso propiamente.

La subestructura jerárquica resultante de estos dos procesos se muestra a continuación :

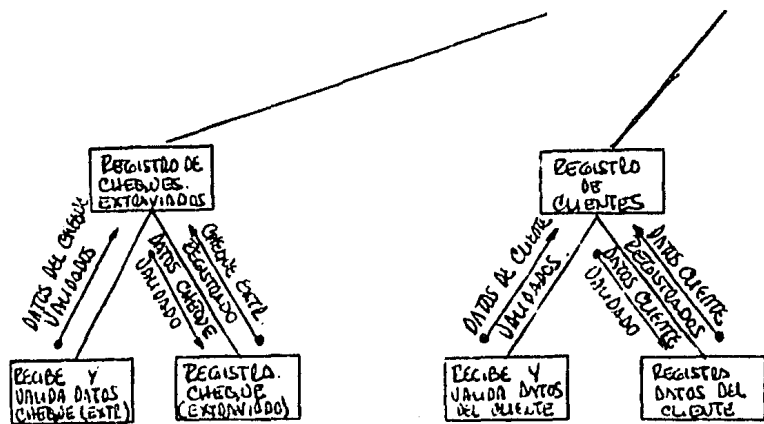


FIG 33 ESTRUCTURA FINAL DE SOFTWARE PARA
LOS PROCESOS PRINCIPALES 2 Y 3

Finalmente se deriva la estructura jerárquica para los procesos "Consulta de Información" y "Calcula Cifras de Control", los cuales debido a que su complejidad de procesamiento no es muy elevada, pueden estructurarse como módulos simples. Estos módulos los describimos en la siguiente figura.

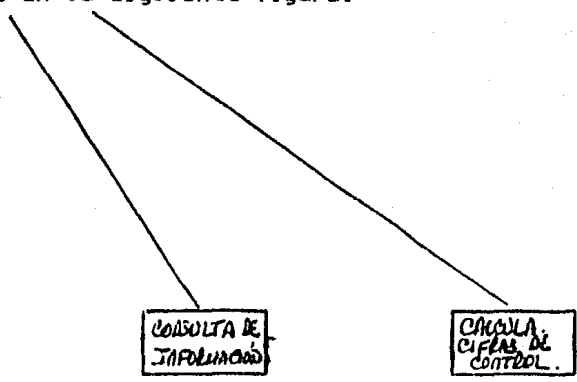


FIG 34 ESTRUCTURA FINAL DE SOFTWARE PARA
LOS PROCESOS PRINCIPALES 4 Y 5

En esta forma se puede dar como finalizada la factorización inicial del Sistema y un refinamiento de segundo nivel.

La necesidad de refinamientos complementarios puede surgir durante las actividades del diseño detallado, mientras tanto, para determinar si la estructura jerárquica resultante representa un diseño completo, se debe checar cuidadosamente.

Se debe revisar el diagrama de flujo, verificando los límites que indican la independencia o dependencia de los procesos. El flujo lógico del diagrama se sigue a través de la estructura jerárquica para determinar si falta algún proceso, si los almacenamientos son accesados en alguna parte o si todos los flujos de datos están representados.

Si se relaciona el diagrama de flujo con la estructura de software, se tendrá la seguridad de que las funciones del Sistema están incluidas.

Concluye esta sección con la descripción de la estructura de software terminada para el Sistema (Diagrama 35).

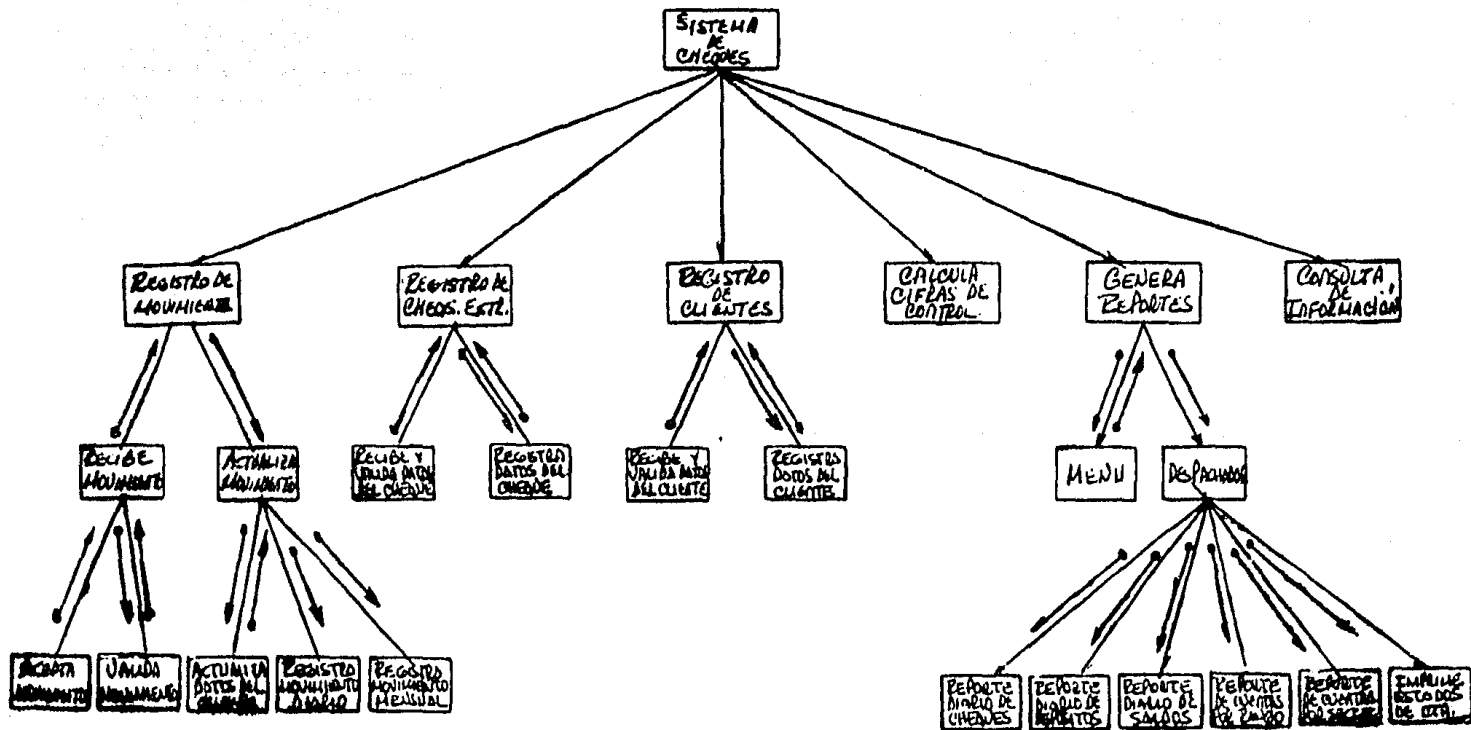


FIG 35 ESTRUCTURA DE SOFTWARE CONCLUIDA
 PARA EL SISTEMA DE CHEQUES.

3.4 EL DISEÑO DETALLADO DEL SISTEMA

Las actividades finales en el proceso de diseño, se relacionan con la especificación de los aspectos de procedimiento de cada módulo y la descripción narrativa que los ubica dentro del Sistema general.

También se da la posibilidad de un refinamiento complementario de la estructura del software, si se detecta un incremento en la complejidad del código, al desarrollar los algoritmos correspondientes.

Se crea asimismo, una especificación para desarrollar las estructuras de datos que se manejan durante la operación del Sistema.

Finalmente se efectúa una revisión técnica, para determinar si las especificaciones de diseño que se proporcionan son suficientes para iniciar la implementación.

En relación a las descripciones narrativas de los módulos, estas se encuentran señaladas en la parte inferior de cada tarjeta "IPO", en el anexo "A", en donde se da una breve descripción de las funciones del módulo correspondiente. De manera análoga se hace la especificación de los aspectos de procedimiento (algoritmos) que permiten obtener los resultados deseados dentro de cada módulo.

La especificación de los algoritmos, se realiza con la ayuda de lenguaje estructurado o pseudocódigo, que indique con precisión las rutinas de trabajo que deben efectuarse.

En esta sección, se da un ejemplo de tales especificaciones con el módulo "Calcula Cifras de Control", considerando el hecho de que este proceso se encuentra clasificado dentro del grupo de procesos con menor grado de complejidad lógica.

Este proceso, como se comenta en la parte de la descripción narrativa (Anexo "R"), debe considerar los importes de los cheques recibidos y registrados y los importes de los depósitos realizados y calcular la suma total por cada grupo. Los resultados deben desplegarse en la pantalla del video en el que se corra el proceso.

Con la descripción anterior debe elaborarse el algoritmo correspondiente para conseguir el resultado deseado.

Una de las posibles redacciones del algoritmo podría ser la siguiente :

INICIALIZACION.

'TOTALCHEQUES' = 0, 'TOTALDEPOSITOS' = 0

PROCESO.

ACCESA ARCHIVO DE "MOVIMIENTO DIARIO"

LEE 'DATOS DEL MOVIMIENTO' HASTA TERMINAR ARCHIVO

SI 'NUMDOC' TIENE RANGO DE CHEQUES.

SUMAR 'IMPMOV' A 'TOTALCHEQUES'

SI 'NUMDOC' TIENE RANGO DE DEPOSITOS

SUMAR 'IMPMOV' A 'TOTALDEPOSITOS'

DESPLIEGA 'TOTALCHEQUES'

DESPLIEGA 'TOTALDEPOSITOS'

CIERRA ARCHIVO

FIN DE PROCESO.

Los algoritmos correspondientes a los módulos restantes se describen en sus respectivas tarjetas "IPO", en el anexo correspondiente.

Uno de los aspectos más relevantes durante la etapa de diseño detallado es la definición de las estructuras de datos que conservarán la información del Sistema.

En el caso particular del Sistema que se desarrolla en este trabajo, se recordará que durante el Análisis se realizaron actividades de evaluación y solución de los detalles que conforman el Sistema.

Con la ayuda de los diagramas de flujo de datos, se definió poco a poco, cada uno de los elementos que forman parte del sistema. Estos elementos fueron recopilándose en un Diccionario de Datos, que forma parte de la documentación y que se continuó complementando hasta la misma etapa de diseño.

Como se tienen identificados los 'Almacenamientos de Datos', (actividad que se desarrolló durante el análisis) ahora se deben especificar sus contenidos. Con la ayuda del diccionario, se puede compilar la lista de elementos que componen los 'Almacenamientos', de donde la Base de Datos o los Archivos se construyen en relación a las estructuras resultantes.

Por ejemplo, si se considera el archivo "Clientes", los elementos que deben almacenarse por cada cliente son :

ARCHIVO : CLIENTES

| NOMBRE DEL ELEMENTO | LONGITUD | NAT. | FORMATO |
|----------------------------|----------|---------|--------------|
| NUMERO DE CUENTA | 6 | ALFNUM. | XXXXXX |
| NOMBRE | 20 | ALFNUM. | XX..XX |
| DIRECCION | 20 | ALFNUM. | XX..XX |
| POBLACION | 20 | ALFNUM. | XX..XX |
| SALDO ACTUAL | 12 | NUM. | 999999999.99 |
| SALDO DEL MES ANTERIOR | 12 | NUM. | 999999999.99 |
| FECHA DE ULTIMO MOVIMIENTO | 6 | NUM. | AAMDD |
| SECTOR ECONOMICO | 2 | NUM. | 99 |

LONGITUD = 98

La longitud total del registro en la estructura es de 98 caracteres; el archivo de Clientes consta de una serie de estos registros lógicos, uno para cada cliente.

La definición de archivos incluye también la selección de organización que tendrán. Existen tres posibilidades de selección :

- Organización Secuencial. Cuando se crea un archivo secuencial, los registros se añaden uno tras otro como 'eslabones'. Cuando se desean procesar, deben ser leídos en la misma secuencia en que entraron; por tanto, la velocidad con que un registro puede ser accedido depende de su ubicación física dentro del archivo.

- Organización Directa o Random. Cuando se utiliza un archivo directo, a cada registro lógico se le asigna una "llave" que corresponde, de alguna manera, a su localización física dentro del disco. En consecuencia es posible almacenar o recuperar cualquier registro, utilizando el mismo tiempo de acceso, sin importar su ubicación física dentro del archivo.

- Organización Indexada. En un archivo indexado, a medida que los datos son almacenados, se crea un "índice" que encadena la llave lógica del registro con su dirección física en el disco. Estos índices se almacenan de manera independiente; cuando los datos son procesados, se extrae la dirección lógica del registro usando el "índice" y los registros son leídos directamente. Consecuentemente un archivo indexado puede ser leído también secuencialmente.

Un objetivo importante de los diseñadores es seleccionar un tipo de organización de datos, que permita al sistema ejecutar sus funciones dentro del menor costo (tiempo) posible.

En particular, se determina que la organización correcta del archivo de "Clientes" debe ser "Directa" o "Random", que permitirá reducir al máximo, el tiempo de respuesta del Sistema, cuando se realicen las operaciones diarias y las consultas de información.

No existe una razón importante para que la organización de los otros archivos del Sistema sea Directa o Indexada (en todo caso, existirían serias desventajas si se construyeran como archivos secuenciales). La definición de cada archivo debe continuarse hasta que el diseñador sea capaz de implementarlos físicamente en el ambiente de software que se utilizará.

A continuación se describen los archivos restantes; se indican el nombre de los elementos, su longitud, naturaleza y formato.

ARCHIVO : CHEQUES EXTRAVIADOS

| NOMBRE DEL ELEMENTO | LONGITUD | NAT. | FORMATO |
|------------------------------|----------|---------|-------------|
| NUMERO DE CUENTA | 6 | ALFNUM. | XXXXXX |
| NUMERO DE CHEQUE EXTRAVIADO | 8 | ALFNUM. | XXXXXXXX |
| IMPORTE DE CHEQUE EXTRAVIADO | 12 | NUM. | 99999999.99 |

LONGITUD = 26

ARCHIVO : MOVIMIENTO DIARIO

| NOMBRE DEL ELEMENTO | LONGITUD | NAT. | FORMATO |
|-----------------------------|----------|---------|-------------|
| NUMERO DE CUENTA | 6 | ALFNUM. | XXXXXX |
| NUMERO DE CHEQUE O DEPOSITO | 8 | ALFNUM. | XXXXXXXX |
| IMPORTE DE MOVIMIENTO | 12 | NUM. | 99999999.99 |
| FECHA DE MOVIMIENTO | 6 | NUM. | AAMMDD |

LONGITUD = 34

ARCHIVO : MOVIMIENTO MENSUAL

| NOMBRE DEL ELEMENTO | LONGITUD | NAT. | FORMATO |
|-----------------------------|----------|---------|-------------|
| NUMERO DE CUENTA | 6 | ALFNUM. | XXXXXX |
| NUMERO DE CHEQUE O DEPOSITO | 8 | ALFNUM. | XXXXXXXX |
| IMPORTE DE MOVIMIENTO | 12 | NUM. | 99999999.99 |
| FECHA DE MOVIMIENTO | 6 | NUM. | AAMMDD |

LONGITUD = 34

Con estas definiciones de archivos podemos dar por concluidas las actividades de diseño. Se debe entonces efectuar una revisión técnica, que permita asegurar que se ha concluido satisfactoriamente y se puede iniciar la implementación del Sistema.

Se revisa la Estructura Jerárquica y las Tarjetas "IPO" contra el Diccionario de Dato para asegurar que la documentación es

consistente (puede ocurrir que el diccionario deba complementarse o que faltan elementos de diseño en el Sistema) y se deben evaluar los módulos para conocer su grado de cohesión y acoplamiento. Las herramientas que nos daran ayuda para realizar lo anterior son las tablas conocidas como "Referencias Cruzadas", en donde se relacionan diferentes conceptos entre sí.

A continuación se presentan dos tablas de referencias cruzadas. La primera establece la relación que existe entre los archivos y los módulos del Sistema. La segunda relaciona los datos elementales del Sistema con los módulos.

La segunda tabla, nos proporciona mayor información sobre el acoplamiento, la cohesión y la independencia de los módulos. Si se observan demasiadas variables que se transfieren entre módulos, se presume que existe un acoplamiento excesivo, lo cual implica dependencia modular. En este caso deberán considerarse rediseños locales en la estructura jerárquica.

Por otra parte, si dentro de un módulo, existen variables innecesarias o cuya relación es dudosa, se puede pensar que la cohesión no es buena y deberá entonces hacerse una revisión de la estructura de control para corregir el problema.

También se checarán las descripciones de los algoritmos y su localización dentro de la estructura de software.

CAP IV TABLAS Y APENDICES DEL SISTEMA

**REFERENCIAS CRUZADAS DE LOS ARCHIVOS Y
LOS MODULOS DEL SISTEMA DE CHEQUES**

| | CLIENTES | MONTO. DIARIO | MONTO. MENSUAL | CHEQUES EXTR. |
|---------------------------------|----------|---------------|----------------|---------------|
| ACEPTA MOVIMIENTO. | ● | | | |
| VALIDA MOVIMIENTO. | ● | | | ● |
| ACTUALIZA DATOS CLIENTE. | ● | | | |
| REGISTRA MONTO. DIARIO | | ● | | |
| REGISTRA MONTO. MENSUAL | ● | | ● | |
| RECIBE/VALIDA DATOS CHEQ. EXTR. | | | | ● |
| REGISTRA DATOS CHEQUE EXTRA. | | | | ● |
| RECIBE/VALIDA DATOS DEL CLIENTE | ● | | | |
| REGISTRA DATOS DEL CLIENTE | ● | | | |
| CALCULA CIFRAS DE CONTROL | | ● | | |
| CONSULTA DE INFORMACION | ● | | | ● |
| REPORTE DIARIO DE CHEQUES | | ● | | |
| REPORTE DIARIOS DE DEPOSITOS | | ● | | |
| REPORTE DIARIO DE SALDOS | ● | | | |
| REPORTE DE CUENTAS POR PAGAR | ● | | | |
| REPORTE DE CUENTAS POR SECT EC | ● | | | |
| IMPRIME ESTADOS DE CUENTA | ● | | ● | |

**REFERENCIAS CRUZADAS DE LOS DATOS ELEMENTALES
Y LOS MODULOS DEL SISTEMA DE CHEQUES**

| | NUM. DE CUENTA | NOMBRE DEL CLIENTE | DIRECCION DEL CLIENTE | POBLACION DEL CLIENTE | SECTOR ECONOMICO | SALDO ACTUAL | SALDO DEL MES ANTERIOR | FECHA DE ULT. MOVIMIENTO | SALDO PROMEDIO | FECHA DE MOVIMIENTO | NUMERO DE DEPOSITO | IMPORTE DE MOVIMIENTO | NUMERO DE CHEQUE | NUMERO DE CHEQUE EXTRAV. | IMPORTE DE CHEQUE EXTRAV. |
|---------------------------------|----------------|--------------------|-----------------------|-----------------------|------------------|--------------|------------------------|--------------------------|----------------|---------------------|--------------------|-----------------------|------------------|--------------------------|---------------------------|
| ACEPTA MOVIMIENTO. | ● | | | | | | | | | | ● | ● | ● | | |
| VALIDA MOVIMIENTO. | ● | | | | | ● | | | | | ● | ● | ● | ● | ● |
| ACTUALIZA DATOS CLIENTE. | ● | | | | | ● | | ● | | | | ● | | | |
| REGISTRA MOVTO. DIARIO | ● | | | | | | | | | ● | ● | ● | ● | | |
| REGISTRA MOVTO. MENSUAL | ● | | | | | | | | | ● | ● | ● | ● | | |
| RECIBE/VALIDA DATOS CHEQ. EXTR. | ● | | | | | | | | | | | ● | ● | ● | ● |
| REGISTRA DATOS CHEQUE EXTRAV. | ● | | | | | | | | | | | ● | ● | ● | ● |
| RECIBE/VALIDA DATOS DEL CLIENTE | ● | ● | ● | ● | ● | | | | | | | | | | |
| REGISTRA DATOS DEL CLIENTE | ● | ● | ● | ● | ● | | | | | | | | | | |
| CALCULA CIFRAS DE CONTROL | | | | | | | | | | | ● | ● | ● | | |
| CONSULTA DE INFORMACION | ● | ● | ● | ● | ● | ● | | ● | | | | | | | |
| REPORTE DIARIO DE CHEQUES | ● | | | | | | | | | | | ● | ● | | |
| REPORTE DIARIOS DE DEPOSITOS | ● | | | | | | | | | | ● | ● | | | |
| REPORTE DIARIO DE SALDOS | ● | ● | | | | ● | | | | | | | | | |
| REPORTE DE CUENTAS POR RAMBOS | ● | | | | | ● | | | | | | | | | |
| REPORTE DE CUENTAS POR SECT EC | ● | | | | ● | ● | | | | | | | | | |
| IMPIME ESTADOS DE CUENTA | ● | ● | ● | ● | | ● | ● | | ● | ● | ● | ● | ● | | |

APENDICE A. DICCIONARIO DE DATOS DEL SISTEMA

La idea básica para formar un Diccionario de Datos, es tener información de la definición, estructura y uso de cada dato elemental que tenga relación con el Sistema que se desarrolla.

Otra razón importante para crear el Diccionario, es mejorar la comunicación entre usuarios y analistas, pues se establece un conjunto de definiciones consistentes.

Asimismo, si los programadores requirieran desarrollar descripciones de datos, un Diccionario común puede evitar problemas potenciales al integrar las diferentes interfaces del sistema.

Tal vez, la ventaja mas concreta en el uso de este tipo de diccionarios se derive del control de información que se lleva para cada dato. Normalmente todos aquellos programas que utilizan un elemento dado, son referenciados en el Diccionario por lo que llega a ser fácil establecer el impacto de un cambio en el dato.

Finalmente, un Diccionario de Datos es un valioso primer paso para desarrollar una Base de Datos del Sistema.

En este apéndice se detalla un pequeño Diccionario de Datos que contiene la información elemental que se maneja en el Sistema.

DICCIONARIO DE DATOS

NOMBRE DEL DATO : NUMERO DE CUENTA.

ALIAS : NUMCTA.

DESCRIPCION : Es el número o clave mediante el cual se identifican cada uno de los clientes que se dan de alta en el sistema. Este número es único, de modo que se puede considerar la clave física de acceso en el Sistema.

FORMATO : Longitud = 6 y alfanumérico (xxxxxx).

ARCHIVOS : Clientes
Cheques Extraviados.
Movimiento Diario.
Movimiento Mensual.

MÓDULOS : Todos, a excepción de "Calcula Cifras de Control" y los módulos de Control.

¡DICCIONARIO DE DATOS!

NOMBRE DEL DATO : NOMBRE DEL CLIENTE.

ALIAS : NOMBRE.

DESCRIPCION : Este dato es solicitado en el proceso de "Altas a Clientes", aparece tanto en consultas de información como en diferentes reportes e identifica al Cliente con su Número de Cuenta. Se deben especificar apellido materno, apellido paterno y nombre

FORMATO : Longitud = 20 y alfanumerico (xxx..xxx)

ARCHIVOS : Clientes.

MODULOS : "Recibe y Valida Datos del Cliente"
"Registra Datos del Cliente"
"Genera Reporte Diario de Saldos"
"Imprime Estados de Cuenta"
"Consulta de Información"

DICCIONARIO DE DATOS

NOMBRE DEL DATO : DIRECCION DEL CLIENTE.

ALIAS : DIRECCION.

DESCRIPCION : Este dato también se solicita al dar de alta los datos del cliente. La información además de utilizarse en consulta de saldos, se imprime en los Estados de Cuenta de Cheques, para ser enviado por correo; se debe indicar calle, número, colonia, C.P.

FORMATO : Longitud = 20 y alfanumerico (xxxx..xxxx)

ARCHIVOS : Clientes.

MODULOS : "Recibe y Valida Datos del Cliente"
"Registra Datos del Cliente"
"Imprime Estados de Cuenta"
"Consulta de Información."

DICCIONARIO DE DATOS

NOMBRE DEL DATO : POBLACION DEL CLIENTE.

ALIAS : POBLACION.

DESCRIPCION : Este dato tambien se solicita en la alta de datos del cliente; se utiliza en consulta de saldos y en la impresion de los Estados de Cuenta.

FORMATO : longitud = 20 y alfanumerico (xxx..xxx)

ARCHIVOS : Clientes

**MODULOS : "Recibe y Valida Datos del Cliente"
"Registra Datos del Cliente"
"Imprime Estados de Cuenta"
"Consulta de Información"**

DICCIONARIO DE DATOS

NOMBRE DEL DATO : SALDO ACTUAL.

ALIAS : SDOACT.

DESCRIPCION : Es el saldo real del Cliente. Este dato es actualizado continuamente mediante el registro de cheques y/o depósitos del Cliente. Cuando finaliza el mes, el saldo actual se convierte en Saldo del Mes Anterior, de modo que al iniciarse el mes siguiente, los dos saldos coinciden.

FORMATO : Longitud = 12 y numerico (999999999.99)

ARCHIVOS : Clientes.

MODULOS : "Consulta de Información"
"Valida Movimiento"
"Actualiza Datos del Cliente"
"Reporte Diario de Saldos"
"Reporte de Cuentas por Rangos"
"Imprime Estados de Cuenta"
"Reporte de Cuentas por Sector Económico"

¡DICCIONARIO DE DATOS!

NOMBRE DEL DATO : SALDO DEL MES ANTERIOR.

ALIAS : SDOANT.

DESCRIPCION : Es el saldo del cliente, que corresponde al último día hábil del mes anterior. Este saldo permanece fijo durante el transcurso del mes vigente y al finalizar este, es sustituido por el "Saldo Actual".

FORMATO : Longitud = 12 y numérico (999999999.99)

ARCHIVOS : Clientes.

MODULOS : "Imprime estados de Cuentas"

DICCIONARIO DE DATOS

NOMBRE DEL DATO : FECHA DE ULTIMO MOVIMIENTO.

ALIAS : FEULMOV

DESCRIPCION : Como su nombre lo indica, esta fecha señala el día, mes y año, en que el saldo actual del cliente fue actualizado por última vez al realizarse un movimiento en la Cuenta del Cliente.

FORMATO : Longitud = 6 y numerico (AAMDD)

ARCHIVOS : Clientes.

MODULOS : "Actualiza Datos del Cliente"
"Consulta de Informacion."

♦ DICCIONARIO DE DATOS ♦

NOMBRE DEL DATO : SALDO PROMEDIO

ALIAS : SDO PROM.

DESCRIPCION : Como su nombre lo indica, este dato representa el saldo promedio del Cliente por el movimiento que tuvo durante el mes. El saldo se obtiene al sumarse los saldos parciales (los saldos de los días en que se movió la cuenta) y dividirla entre el número de días del mes que transcurre.

FORMATO : Longitud = 12 y numerico (999999999.99)

ARCHIVOS : Es un dato local (se genera en el programa)

MODULOS : "Imprime Estados de Cuenta".

DICCIONARIO DE DATOS

NOMBRE DEL DATO : FECHA DE MOVIMIENTO

ALIAS : FEMOV

DESCRIPCION : Como su nombre lo indica, este dato proporciona el día, mes y año en que se registran los movimientos en el Sistema. El dato se genera automáticamente en el C.P.U. y por lo tanto no es necesario solicitarlo al usuario.

FORMATO : Longitud = 6 y numerico (AAMDD)

ARCHIVOS : Movimiento Diario
Movimiento Mensual

MODULOS : "Registra Movimiento Diario"
"Registra Movimiento Mensual"
"Imprime Estados de Cuenta"

DICCIONARIO DE DATOS

NOMBRE DEL DATO : NUMERO DE DEPOSITO.

ALIAS : NUMDOC.

DESCRIPCION : Este dato es el numero de identificación del depósito que registra el Cliente. El número se genera mediante programa y se le comunica al (Ciente) para llevar un control de los movimientos. Estos números tienen un rango diferente al de los cheques para distinguir el tipo de movimiento.

FORMATO : Longitud = 9 y alfanumerico (XXXXXXXX)

ARCHIVOS : Movimiento Diario.
Movimiento Mensual.

MODULOS : "Acepta Movimiento"
"Valida Movimiento"
"Registra Movimiento Diario"
"Registra Movimiento Mensual"
"Reporte Diario de Depositos"
"Imprime Estados de Cuenta"

DICCIONARIO DE DATOS

NOMBRE DEL DATO : IMPORTE DE MOVIMIENTO

ALIAS : IMPMOV

DESCRIPCION : Es la cantidad que aparece físicamente en un cheque o ficha de depósito y que nos indica el valor del documento.

FORMATO : Longitud = 12 y numerico (999999999.99)

ARCHIVOS : Movimiento Diario
Movimiento Mensual

MODULOS : "Acepta Movimiento"
"Valida Movimiento"
"Actualiza Datos del Cliente"
"Registra Movimiento Diario"
"Registra Movimiento Mensual"
"Recibe y Valida Datos del Cheque (Extrav.)"
"Registra Datos del Cheque (Extrav.)"
"Calcula Cifras de Control"
"Reporte Diario de Cheques"
"Reporte Diario de Depositos"
"Imprime Estdos de Cuenta"

★DICCIONARIO DE DATOS★

NOMBRE DEL DATO : NUMERO DE CHEQUE

ALIAS : NUMDOC

DESCRIPCION : Este dato es el numero de identificación del cheque que se registra (el numero que aparece físicamente en el documento). Estos números tienen un rango diferente al de los números de deposito para distinguir el tipo de movimiento. El alias del dato es el mismo que para "NUMERO DE DEPOSITO" porque ocupan el mismo lugar en el registro lógico del archivo.

FORMATO : Longitud = 8 y alfanumerico (xxxxxxxx)

ARCHIVOS : Movimiento Diario
Movimiento Mensual
Cheques Extraviados

MODULOS : "Acepta Movimiento"
"Valide Movimiento"
"Registra Movimiento Diario"
"Registra Movimiento Mensual"
"Recibe y Valida Datos del Cheque (Extrav.)"
"Registra Datos del Cheque" (Extrav.)
"Reporte Diario de Cheques"
"Imprime Estados de Cuenta "

DICCIONARIO DE DATOS

NOMBRE DEL DATO : SECTOR ECONOMICO

ALIAS : SECTEC.

DESCRIPCION : Este dato se solicita en la captura inicial de datos del cliente y nos indica el Sector Económico al que pertenece. Los sectores económicos que se manejan en el sistema son tres : personas físicas, personas morales y gobierno.

FORMATO : Longitud = 2 y numerico (99)

ARCHIVOS : Clientes.

MODULOS : "Recibe y Valida Datos del Cliente"
"Registra Datos del Cliente"
"Consulta de Información"
"Genera Reporte de Cuentas por Sector Ec."

APENDICE "B" LAS TARJETAS "IPO"

Este apéndice contiene documentación complementaria sobre el Sistema de Cheques que se desarrolla.

Las herramientas de diseño que se han utilizado en la construcción de este proyecto de software, incluyen a la técnica "HIPO", por medio de la cual se define una estructura (inicial) de software que gradualmente es refinada hasta alcanzar la descomposición funcional con el nivel de detalle que se desea.

Las tarjetas "IPO" (Input/Process/Output), se preparan entonces par describir el funcionamiento de cada módulo que se creó al definir la estructura de software."

La tarjetas "IPO" muestran los datos de entrada, los datos de salida, el proceso o rutina que realiza el módulo, los módulos que lo invocan y los módulos que llama.

La documentación resultante, generalmente es checada contra el diagrama de flujo de datos, el diccionario de datos y la descripción de algoritmos, que también forman parte de la documentación del Sistema.

Aquí se describen las 26 tarjetas "IPO" que identifican a los 26 módulos que conforman la estructura jerárquica final del proyecto.

TARJETA "IPD" NUM. 001

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : PRINCIPAL (SISTEMA DE CHEQUES)

FECHA : 15/08/88

LLAMADO O INVOCADO POR : - - -

LLAMA O INVOKA A : "REGISTRA MOVIMIENTO
"REGISTRA CHEQUES EXTRAVIADOS
"REGISTRA A CLIENTES"
"CALCULA CIFRAS DE CONTROL"
"CONSULTA DE INFORMACION"
"GENERA REPORTES"

DATOS DE ENTRADA : VARIABLES DE CONTROL

DATOS DE SALIDA : VARIABLES DE CONTROL

* * DESCRIPCION NARRATIVA DEL PROCESO * *

POR SER EL MODULO PRINCIPAL DEL SISTEMA, SE ENCARGA DEL CONTROL DE TODA LA ESTRUCTURA DE SOFTWARE. SU NIVEL SE PUEDE CONSIDERAR EL MAS ALTO DEL SISTEMA; EVENTUALMENTE LE PUEDE SER ADICIONADO CONTROL PARA CHEQUEO DE CLAVES DE ACCESO, NIVELES DE ACCESO, ETC.

* * DESCRIPCION DEL ALGORITMO * *

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 002

SISTEMA : CHEQUES PREPARADO POR : G.T.V.
MODULO : "REGISTRO DE MOVIMIENTOS" FECHA : 15/08/88
LLAMADO O INVOCADO POR : MODULO PRINCIPAL (SISTEMA DE CHEQUES)
LLAMA O INVOKA A : "RECIBE MOVIMIENTO"
"ACTUALIZA MOVIMIENTO"

DATOS DE ENTRADA : VARIABLES DE CONTROL

DATOS DE SALIDA : VARIABLES DE CONTROL

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO PRINCIPAL PARA TOMAR A SU VEZ EL CONTROL DE LA PRIMERA FUNCION PRINCIPAL DEL SISTEMA. ASIMISMO ACCESA LOS ARCHIVOS "CLIENTES" Y "CHEQUES EXTRAVIADOS" CUANDO PASA EL CONTROL AL MODULO "RECIBE MOVIMIENTO" Y LOS ARCHIVOS "CLIENTES", "MOVIMIENTO DIARIO" Y "MOVIMIENTO MENSUAL" CUANDO TRANSFIERE EL CONTROL AL MODULO "ACTUALIZA MOVIMIENTO"

* * DESCRIPCION DEL ALGORITMO * *

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 003

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "RECIBE MOVIMIENTO"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "REGISTRO DE MOVIMIENTO"

LLAMA O INVOCA A : "ACEPTA MOVIMIENTO"
"VALIDA MOVIMIENTO"

DATOS DE ENTRADA : CONTROL, 'NUMCTA', 'NUMDOC' Y 'IMPMOV'

DATOS DE SALIDA : CONTROL, 'NUMCTA', 'NUMDOC' Y 'IMPMOV'

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "REGISTRO DE MOVIMIENTOS" Y SE ENCARGA DE PASAR EL CONTROL ENTRE LOS MODULOS "RECIBE MOVIMIENTO" Y "VALIDA MOVIMIENTO" TRANSFIRIENDO LOS DATOS CORRESPONDIENTES

.....
.....
.....

* * DESCRIPCION DEL ALGORITMO * *

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 004

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : ACEPTA MOVIMIENTO

FECHA : 15/08/89

LLAMADO O INVOCADO POR : "RECIBE MOVIMIENTO"

LLAMA O INVOKA A : - - -

DATOS DE ENTRADA : - - -

DATOS DE SALIDA : 'NUMETA', 'NUMDOC' Y 'IMPMOV'

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "RECIBE MOVIMIENTO" Y SE ENCARGA DE RECIBIR DEL USUARIO LOS DATOS NECESARIOS PARA REALIZAR UNA TRANSACCION

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- DESPLIEGA PANTALLA PARA RECIBIR DATOS DEL MOVIMIENTO.
- SOLICITA Y ACEPTA 'NUMETA'
- SOLICITA Y ACEPTA 'NUMDOC'
- SOLICITA Y ACEPTA 'IMPMOV'
- TERMINA RUTINA.

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 005

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "VALIDA MOVIMIENTO"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "RECIBE MOVIMIENTO"

LLAMA O INVOCA A : - - -

DATOS DE ENTRADA : 'NUMCTA', 'NUMDOC' Y 'IMPMOV'

DATOS DE SALIDA : 'NUMCTA', 'NUMDOC' Y 'IMPMOV'

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "RECIBE MOVIMIENTO" Y SE ENCARGA DE VALIDAR LOS DATOS QUE LE MANDA EL MODULO "ACEPTA MOVIMIENTO"

.....
.....
.....
.....

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- RECIBE 'NUMCTA' Y VERIFICA QUE SEA NUMERICO Y DE LONG. CORRECTA SI 'NUMCTA' ES INCORRECTO DESPLIEGA MENSAJE "NUMERO DE CUENTA INCORRECTO, VERIFICAR" TERMINA RUTINA.
- RECIBE 'NUMDOC' Y VERIFICA QUE SEA NUMERICO Y DE LONG. CORRECTA SI 'NUMDOC' ES INCORRECTO DESPLIEGA MENSAJE "NUMERO DE DOCTO. INCORRECTO, VERIFICAR" TERMINA RUTINA.
- RECIBE 'IMPMOV' Y VERIFICA QUE SEA NUMERICO Y CON RANGO CORRECTO SI 'IMPMOV' ES INCORRECTO DESPLIEGA MENSAJE "IMPORTE INCORRECTO, VERIFICAR" TERMINA RUTINA.
- TERMINA RUTINA.

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 006

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "ACTUALIZA MOVIMIENTO"

FECHA : 15/09/88

LLAMADO O INVOCADO POR : "REGISTRO DE MOVIMIENTO"

LLAMA O INVOKA A : "ACTUALIZA DATOS DEL CLIENTE"
"REGISTRA MOVIMIENTO DIARIO"
"REGISTRA MOVIMIENTO MENSUAL"

DATOS DE ENTRADA : CONTROL, 'NUMCTA', 'NUMDOC' Y 'IMPMOV'

DATOS DE SALIDA : CONTROL, 'NUMCTA', 'NUMDOC' Y 'IMPMOV'

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "REGISTRO DE MOVIMIENTOS" Y SE ENCARGA DE PASAR EL CONTROL A LOS MODULOS "ACTUALIZA DATOS DEL CLIENTE", "REGISTRA MOVIMIENTO DIARIO" Y "REGISTRA MOVIMIENTO MENSUAL" TRANSFIRIENDO LOS DATOS CORRESPONDIENTES

.....
.....
.....

* * DESCRIPCION DEL ALGORITMO * *

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPD" NUM. 007

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "ACTUALIZA DATOS DEL CLIENTE"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "ACTUALIZA MOVIMIENTO"

LLAMA O INVOCA A : - - -

DATOS DE ENTRADA : 'NUMCTA', 'NUMDOC' Y 'IMPMOV'

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "ACTUALIZA MOVIMIENTO" Y SE ENCARGA DE ACTUALIZAR LOS DATOS DEL CLIENTE CADA VEZ QUE SE REALIZA UNA TRANSACCION

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- RECIBE 'NUMCTA' Y 'IMPMOV' Y 'NUMDOC'
- LEE DEL ARCHIVO DE "CLIENTES" EL REGISTRO QUE CORRESPONDE A 'NUMCTA'
- EL VALOR DE 'FEMOV' SE TRANSFIERE A 'FEULMOV'
- SI 'NUMDOC' PERTENECE A RANGO DE CHEQUES
 RESTAR AL VALOR DE 'SDOACT' EL VALOR DE 'IMPMOV'
- SI 'NUMDOC' PERTENECE A RANGO DE DEPOSITOS
 SUMAR AL VALOR DE 'SDOACT' EL VALOR DE 'IMPMOV'
- TERMINA RUTINA

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 008

SISTEMA : CHEQUES

PREPARADO POR : G.T.

MODULO : "REGISTRA MOVIMIENTO DIARIO"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "ACTUALIZA MOVIMIENTO"

LLAMA O INVOKA A : - - -

DATOS DE ENTRADA : 'NUMCTA', 'NUMDOC' Y 'IMPMOV'

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "ACTUALIZA MOVIMIENTO" Y SE ENCARGA DE REGISTRAR LOS DATOS DEL MOVIMIENTO EN EL ARCHIVO DE "MOVIMIENTO DIARIO" CADA VEZ QUE SE REALIZA UNA TRANSACCION

.....

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- RECIBE 'NUMCTA' Y 'IMPMOV' Y 'NUMDOC'
- GRABA EN EL ARCHIVO "MOVIMIENTO DIARIO" LOS DATOS 'FEMOV', 'IMPMOV', 'NUMDOC', 'NUMCTA'
- TERMINA RUTINA

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 009

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "REGISTRA MOVIMIENTO MENSUAL"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "ACTUALIZA MOVIMIENTO"

LLAMA O INVOC A : - - -

DATOS DE ENTRADA : 'NUMCTA', 'NUMDOC' Y 'IMPMOV'

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "ACTUALIZA MOVIMIENTO" Y SE ENCARGA DE REGISTRAR LOS DATOS DE LA TRANSACCION EN EL ARCHIVO "MOVIMIENTO MENSUAL"

.....
.....
.....

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- RECIBE 'NUMCTA' Y 'IMPMOV' Y 'NUMDOC'
- GRABA EN EL ARCHIVO "MOVIMIENTO MENSUAL" LOS DATOS 'FEMOV',
..... 'IMPMOV', 'NUMDOC', 'NUMCTA'
- TERMINA RUTINA

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 010

SISTEMA : CHEQUES PREPARADO POR : G.T.V.

MODULO : "REGISTRO DE CHEQUES EXTRAVIADOS" FECHA : 15/08/88

LLAMADO O INVOCADO POR : MODULO PRINCIPAL (SISTEMA DE CHEQUES)

LLAMA O INVOCA A : "RECIBE Y VALIDA DATOS DEL CHEQUE.(EXTRAV.)
"REGISTRA DATOS DEL CHEQUE (EXTRAV.)"

DATOS DE ENTRADA : VARIABLES DE CONTROL

DATOS DE SALIDA : VARIABLES DE CONTROL

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO PRINCIPAL PARA TOMAR A SU VEZ EL CONTROL DE LA SEGUNDA FUNCION PRINCIPAL DEL SISTEMA. ASIMISMO ACCESA LOS ARCHIVOS "CLIENTES" Y "CHEQUES EXTRAVIADOS" CUANDO PASA EL CONTROL AL MODULO "RECIBE Y VALIDA DATOS DEL CHEQUE" Y EL ARCHIVO "CHEQUES EXTRAVIADOS" CUANDO TRANSFIERE EL CONTROL AL MODULO "REGISTRA DATOS DEL CHEQUE"

* * DESCRIPCION DEL ALGORITMO * *

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 011

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "RECIBE Y VALIDA DATOS DEL CHEQUE" FECHA : 15/08/88

LLAMADO O INVOCADO POR : "REGISTRO DE CHEQUES EXTRAVIADOS"

LLAMA O INVOC A : - - -

DATOS DE ENTRADA : 'NUMCTA', 'NUMDOC' Y 'IMPMOV'

DATOS DE SALIDA : 'NUMCTA', 'NUMDOC' Y 'IMPMOV'

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "REGISTRO DE CHEQUES EXTRAVIADOS" Y SU FUNCION ES RECIBIR Y VALIDAR LOS DATOS PARA REGISTRAR POSTERIORMENTE CHEQUES EXTRAVIADOS

.....
.....
.....

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- DESPIEGA PANTALLA PARA SOLICITAR DATOS
- VERIFICA QUE 'NUMCTA' SEA NUMERICO Y DENTRO DE RANGO CORRECTO
SI 'NUMCTA' ES INCORRECTO
DESPIEGA MENSAJE "NUMERO DE CUENTA INCORRECTO, VERIFICAR"
TERMINA RUTINA.
- LEE EN ARCHIVO "CLIENTES" EL REGISTRO DE 'NUMCTA'
SI NO ENCUENTRA EL REGISTRO
DESPLIEGA MENSAJE "NUMERO DE CUENTA NO EXISTE, VERIFICAR"
TERMINA RUTINA.
- RECIBE 'NUMDOC' Y VERIFICA QUE SEA NUMERICO Y LONGITUD CORRECTA
SI 'NUMDOC' ES INCORRECTO
DESPLIEGA MENSAJE "NUMERO DE CHEQUE INCORRECTO, VERIFICAR"
TERMINA RUTINA.
- RECIBE 'IMPMOV' Y VERIFICA QUE SEA NUMERICO Y CON RANGO CORRECTO
SI 'IMPMOV' ES INCORRECTO
DESPLIEGA MENSAJE "IMPORTE INCORRECTO, VERIFICAR"
TERMINA RUTINA.
- TERMINA RUTINA.

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 012

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "REGISTRA DATOS DEL CHEQUE"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "REGISTRO DE CHEQUES EXTRAVIADOS"

LLAMA O INVOKA A : - - -

DATOS DE ENTRADA : 'NUMCTA', 'NUMDOC' Y 'IMPMOV'

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "REGISTRO DE CHEQUES EXTRAVIADOS" Y SU FUNCION ES REGISTRAR LOS DATOS DEL CHEQUE EXTRAVIADO

.....

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- RECIBE 'NUMCTA', 'IMPMOV' Y 'NUMDOC'
- LEE EN EL ARCHIVO DE "CHEQUES EXTRAVIADOS" EL REGISTRO QUE
.....CORRESPONDE A 'NUMCTA'+ 'NUMDOC'
- SI ENCUENTRA EL REGISTRO
DESPLIEGA MENSAJE "EL CHEQUE YA ESTA REGISTRADO, VERIFIQUE"
DESPLIEGA 'NUMCTA', 'NUMDOC', 'IMPMOV' DEL CHEQUE ENCONTRADO
TERMINA RUTINA
- GRABA EN EL ARCHIVO DE "CHEQUES EXTRAVIADOS" LOS DATOS 'NUMCTA',
.....'NUMDOC', 'IMPMOV'
- TERMINA RUTINA

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPD" NUM. 013

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "REGISTRO DE CLIENTES"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : MODULO PRINCIPAL (SISTEMA DE CHEQUES)

LLAMA O INVOKA A : "RECIBE Y VALIDA DATOS DEL CLIENTE"
"REGISTRA DATOS DEL CLIENTE"

DATOS DE ENTRADA : VARIABLES DE CONTROL

DATOS DE SALIDA : VARIABLES DE CONTROL

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO PRINCIPAL PARA TOMAR A SU VEZ EL CONTROL DE LA TERCERA FUNCION PRINCIPAL DEL SISTEMA. ASIMISMO ACCESA EL ARCHIVO "CLIENTES" CUANDO TRANSFIERE EL CONTROL A LOS MODULOS "RECIBE Y VALIDA DATOS DEL CLIENTE" Y "REGISTRA DATOS DEL CLIENTE"

* * DESCRIPCION DEL ALGORITMO * *

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 014

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "RECIBE Y VALIDA DATOS DEL CLIENTE" FECHA : 15/08/88

LLAMADO O INVOCADO POR : "REGISTRO DE CLIENTES"

LLAMA O INVOCA A : - - -

DATOS DE ENTRADA : - - -

DATOS DE SALIDA : 'NUMCTA', 'NOMBRE' Y 'DIRECCION', 'SECTEC',
'POBLACION',

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "REGISTRO DE CLIENTES" Y SU
FUNCION ES RECIBIR Y VALIDAR LOS DATOS PARA REGISTRAR POSTERIORMENTE
A LOS CLIENTES

.....

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- DESPLIEGA PANTALLA PARA SOLICITAR DATOS
- VERIFICA QUE 'NUMCTA' SEA NUMERICO Y DENTRO DE RANGO CORRECTO
SI 'NUMCTA' ES INCORRECTO
DESPLIEGA MENSAJE "NUMERO DE CUENTA INCORRECTO, VERIFICAR"
TERMINA RUTINA.
- LEE EN ARCHIVO "CLIENTES" EL REGISTRO DE 'NUMCTA'
SI ENCUENTRA EL REGISTRO
DESPLIEGA MENSAJE "NUMERO DE CUENTA YA EXISTE"
DESPLIEGA 'NOMBRE', 'DIRECCION', 'POBLACION', 'SECTEC'
TERMINA RUTINA.
- SOLICITA Y ACEPTA 'NOMBRE'
- VERIFICA QUE 'NOMBRE' TENGA LA LONGITUD CORRECTA
SI 'NOMBRE' ES INCORRECTO
DESPLIEGA MENSAJE "NOMBRE INCORRECTO, VERIFIQUE"
TERMINA RUTINA
- SOLICITA Y ACEPTA 'DIRECCION'
- VERIFICA QUE 'DIRECCION' TENGA LONGITUD CORRECTA
SI 'DIRECCION' ES INCORRECTA
DESPLIEGA MENSAJE "DIRECCION INCORRECTA, VERIFICAR"
TERMINA RUTINA.

(CONTINUA ALGORITMO EN LA SIG. PAG.)

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 014

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "RECIBE Y VALIDA DATOS DEL CLIENTE" FECHA : 15/08/88

* * CONTINUACION DEL ALGORITMO * *

- SOLICITA Y ACEPTA 'POBLACION'
- VERIFICA QUE 'POBLACION' TENGA LONGITUD CORRECTA
SI 'POBLACION' ES INCORRECTA
DESPLIEGA MENSAJE "POBLACION INCORRECTA, VERIFICAR
TERMINA RUTINA.
- SOLICITA Y ACEPTA 'SECTEC'
- VERIFICA QUE 'SECTEC' SEA "01", "02" O "03"
SI 'SECTEC' ES INCORRECTO
DESPLIEGA MENSAJE "SECTOR ECONOMICO INCORRECTO, VERIFICAR
TERMINA RUTINA
- TERMINA RUTINA.

TARJETA "IPO" NUM. 015

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "REGISTRA DATOS DEL CLIENTE"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "REGISTRO DE CLIENTES"

LLAMA O INVOCA A : - - -

DATOS DE ENTRADA : 'NUMCTA', 'NOMBRE' Y 'DIRECCION', 'SECTEC',
'POBLACION'

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "REGISTRO CLIENTES" Y SU
FUNCION ES REGISTRAR LOS DATOS DEL CLIENTE

.....
.....
.....

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- RECIBE 'NUMCTA', 'NOMBRE', 'DIRECCION', 'POBLACION' Y 'SECTEC'
- HACE 'SDOACT' = 0, 'SDOANT' = 0 Y 'FEULMOV' = 00000
- REGISTRA EN EL ARCHIVO "CLIENTES" LOS DATOS 'NUMCTA', 'NOMBRE',
..... 'DIRECCION', 'POBLACION', 'SECTEC', 'SDOANT', 'SDOACT', 'FEULMOV'
- TERMINA RUTINA

DATOS LOCALES :

NOTAS :

TARJETA "IPO" NUM. 016

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "CALCULA CIFRAS DE CONTROL"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : MODULO PRINCIPAL (SISTEMA DE CHEQUES)

LLAMA O INVOKA A : - - -

DATOS DE ENTRADA : - - -

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO PRINCIPAL Y SU FUNCION ES
CALCULAR LOS IMPORTES TOTALES DE LOS MOVIMIENTOS DE CHEQUES Y
DEPOSITOS EFECTUADOS DURANTE EL DIA
.....
.....
.....

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- DESPIEGA PANTALLA PARA MOSTRAR CIFRAS DE CONTROL
- ACCESA ARCHIVO "MOVIMIENTO DIARIO"
- LEE ARCHIVO "MOVIMIENTO DIARIO" SECUENCIALMENTE
 - SI 'NUMDOC' ESTA EN EL RANGO DE CHEQUES
 - HACER 'TOTALCHEQUES' = 'TOTALCHEQUES' + 'IMPMOV'
 - HACER 'CONT/CH' = 'CONT/CH' + 1
 - SI 'NUMDOC' ESTA EN EL RANGO DE DEPOSITOS
 - HACER 'TOTALDEPOSITOS' = 'TOTALDEPOSITOS' + 'IMPMOV'
 - HACER 'CONT/DEP' = 'CONT/DEP' + 1
 - SI NO ES FIN DE ARCHIVO
 - PASAR CONTROL PARA LEER OTRO REGISTRO
- DESPIEGA "IMPORTE TOTAL DE CHEQUES Y DEPOSITOS, NUM DE CHEQUES,
.....NUM DE DEPOSITOS", 'TOTAL CHEQUES', 'CONT/CH', 'TOTALDEPOSITOS',
.....'CONT/DEP'
- TERMINA RUTINA

DATOS LOCALES : 'TOTALCHEQUES', 'TOTALDEPOSITOS', 'CONT/CH', 'CONT/DEP'

NOTAS :
.....

TARJETA "IPO" NUM. 017

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "CONSULTA DE INFORMACION"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : MODULO PRINCIPAL (SISTEMA DE CHEQUES)

LLAMA O INVOKA A : - - -

DATOS DE ENTRADA : - - -

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO PRINCIPAL Y SU FUNCION ES DESPLEGAR EN PANTALLA LA INFORMACION DE LOS CLIENTES QUE SOLICITE EL USUARIO.....

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- DESPLIEGA PANTALLA PARA MOSTRAR DATOS DEL CLIENTE
- ACCESA ARCHIVO "CLIENTES"
- SOLICITA Y ACEPTA 'NUMCTA'
- VERIFICA QUE 'NUMCTA' SEA NUMERICO Y TENGA LONGITUD CORRECTA
SI 'NUMCTA' ES INCORRECTO
DESPLIEGA MENSAJE "NUMERO DE CUENTA INCORRECTO, VERIFIQUE"
TERMINA RUTINA
- LEE EN ARCHIVO "CLIENTES" EL REGISTRO QUE CORRESPONDE A 'NUMCTA'
SI NO EXISTE EL REGISTRO
DESPLIEGA MENSAJE "EL NUMERO DE CUENTA NO SE HA DADO DE ALTA"
TERMINA RUTINA
- DESPLIEGA EN PANTALLA LOS DATOS 'NUMCTA', 'NOMBRE', 'DIRECCION',
..... 'POBLACION', 'SECTEC', 'SDOACT', 'FEHLMOV'
- TERMINA RUTINA

DATOS LOCALES :

NOTAS :

TARJETA "IPO" NUM. 018

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "GENERA REPORTES"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : MODULO PRINCIPAL (SISTEMA DE CHEQUES)

LLAMA O INVOKA A : "MENU"
"DESPACHADOR"

DATOS DE ENTRADA : VARIABLES DE CONTROL

DATOS DE SALIDA : VARIABLES DE CONTROL

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO PRINCIPAL PARA TOMAR A SU VEZ EL CONTROL DE LA SEXTA FUNCION PRINCIPAL DEL SISTEMA, ASINISMO ACCESA LOS ARCHIVOS "CLIENTES", "MOVIMIENTO DIARIO" Y "MOVIMIENTO MENSUAL" CUNADO TRANSFIERE EL CONTROL A LOS MODULOS QUE GENERAN LOS REPORTES DEL SISTEMA.

* * DESCRIPCION DEL ALGORITMO * *

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 019

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "MENU"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "GENERA REPORTES"

LLAMA O INVOCA A : - - -

DATOS DE ENTRADA : VARIABLES DE CONTROL

DATOS DE SALIDA : VARIABLES DE CONTROL

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "GENERA REPORTES" Y DESPLIEGA LA INFORMACION SOBRE LAS OPCION DE IMPRESION DE REPORTES CON QUE CUENTA EL SISTEMA. EL USUARIO ESCOGE AQUI LA ALTERNATIVA QUE REQUIERE.

.....
.....

* * DESCRIPCION DEL ALGORITMO * *

DATOS LOCALES :

NOTAS :

.....

TARJETA "IPO" NUM. 020

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "DESPACHADOR" (DE OPCIONES)

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "GENERA REPORTES"

LLAMA O INVOKA A : "REPORTE DIARIO DE CHEQUES"
"REPORTE DIARIO DE DEPOSITOS"
"REPORTE DIARIO DE SALDOS"
"REPORTE DE CUENTAS POR RANGOS"
"REPORTE DE CUENTAS POR SECTOR EC."
"IMPRIME ESTADOS DE CUENTA"

DATOS DE ENTRADA : VARIABLES DE CONTROL

DATOS DE SALIDA : VARIABLES DE CONTROL

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "GENERA REPORTES" Y A SU VEZ TRANSFIERE EL CONTROL A LOS REPORTES CON QUE CUENTA EL SISTEMA, DE ACUERDO A LA ALTERNATIVA QUE ESCOGE EL USUARIO EN EL MODULO "MENU"

* * DESCRIPCION DEL ALGORITMO * *

DATOS LOCALES :

NOTAS :

TARJETA "IPO" NUM. 021

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "REPORTE DIARIO DE CHEQUES"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "DESPACHADOR"

LLAMA O INVOKA A : - - -

DATOS DE ENTRADA : - - -

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "DESPACHADOR" Y SU FUNCION ES
GENERAR EL REPORTE DE LOS CHEQUES QUE SE OPERARON DURANTE EL DIA

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- VERIFICA QUE LA IMPRESORA ESTA LISTA
SI LA IMPRESORA NO ESTA LISTA
DESPLEGA MENSAJE "LA IMPRESORA NO ESTA LISTA, VERIFICAR"
TERMINA RUTINA
- IMPRIME ENCAEZADOS DEL REPORTE
- LEE EL ARCHIVO DE "MOVIMIENTO DIARIO" SECUENCIALMENTE
SI 'NUMDOC' ESTA EN RANGO DE DEPOSITOS
PASA CONTROL PARA LEER OTRO REGISTRO
- IMPRIME 'NUMCTA', 'NUMDOC', 'IMPMOV'
- HACE 'TOTALCHEQUES' = 'TOTALCHEQUES' + 'IMPMOV'
- HACE 'CONT/CH' = 'CONT/CH' + 1
SI NO ES FIN DE ARCHIVO
LEE OTRO REGISTRO
- IMPRIME "IMPORTE TOTAL DE CHEQUES : ", 'TOTALCHEQUES'
- IMPRIME "NUMERO DE CHEQUES OPERADOS : ", 'CONT/CH'

DATOS LOCALES : 'TOTALCHEQUES', 'CONT/CH'

NOTAS :

.....

TARJETA "IPD" NUM. 022

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "REPORTE DIARIO DE DEPOSITOS"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "DESPACHADOR"

LLAMA O INVOCA A : - - -

DATOS DE ENTRADA : - - -

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "DESPACHADOR" Y SU FUNCION ES
GENERAR EL REPORTE DE LOS DEPOSITOS QUE SE OPERARON EN EL DIA

.....
.....
.....

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- VERIFICA QUE LA IMPRESORA ESTA LISTA
SI LA IMPRESORA NO ESTA LISTA
DESPLIEGA MENSAJE "LA IMPRESORA NO ESTA LISTA, VERIFICAR"
TERMINA RUTINA
- IMPRIME ENCAREZADOS DEL REPORTE
- LEE EL ARCHIVO DE "MOVIMIENTO DIARIO" SECUENCIALMENTE
SI 'NUMDOC' ESTA EN RANGO DE CHEQUES
PASA CONTROL PARA LEER OTRO REGISTRO
- IMPRIME 'NUMCTA', 'NUMDOC', 'IMPMOV'
- HACE 'TOTALDEPOSITOS' = 'TOTALDEPOSITOS' + 'IMPMOV'
- HACE 'CONT/DEP' = 'CONT/DEP' + 1
SI NO ES FIN DE ARCHIVO
LEE OTRO REGISTRO
- IMPRIME "IMPORTE TOTAL DE DEPOSITOS : ", 'TOTALDEPOSITOS'
- IMPRIME "NUMERO DE DEPOSITOS OPERADOS : ", 'CONT/DEP'

DATOS LOCALES : 'TOTALDEPOSITOS', 'CONT/DEP'

NOTAS :

.....

TARJETA "IPD" NUM. 023

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "REPORTE DIARIO DE SALDOS"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "DESPACHADOR"

LLAMA O INVOKA A : - - -

DATOS DE ENTRADA : - - -

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "DESPACHADOR" Y SU FUNCION ES
GENERAR EL REPORTE DE LOS SALDOS CON QUE TERMINAN EL DIA LOS CLIENTES

.....
.....
.....
.....

* * DESCRIPCION DE ALGORITMO * *

- INICIA RUTINA
- VERIFICA QUE LA IMPRESORA ESTA LISTA
SI LA IMPRESORA NO ESTA LISTA
DESPLIEGA MENSAJE "LA IMPRESORA NO ESTA LISTA, VERIFICAR"
TERMINA RUTINA
- IMPRIME ENCAPEZADOS DEL REPORTE
- LEE EL ARCHIVO DE "CLIENTES" SECUENCIALMENTE
- IMPRIME 'NUMCTA', 'NOMBRE', 'SDOACT'
- HACE 'TOTALSALDOS' = 'TOTALSALDOS' + 'IMPMOV'
- HACE 'CONT/CTA' = 'CONT/CTA' + 1
SI NO ES FIN DE ARCHIVO
LEE OTRO REGISTRO
- IMPRIME "IMPORTE TOTAL DE SALDOS : ", 'TOTALSALDOS'
- IMPRIME "NUMERO DE CUENTAS : ", 'CONT/CTA'

DATOS LOCALES : 'TOTALSALDOS', 'CONT/CTA'

NOTAS :

.....

TARJETA "IPO" NUM. 024

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "REPORTE DE CUENTAS POR RANGOS"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "DESPACHADOR"

LLAMA O INVOC A : - - -

DATOS DE ENTRADA : - - -

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "DESPACHADOR" Y SU FUNCION ES GENERAR EL REPORTE DE LAS CUENTAS AGRUPADAS POR LOS RANGOS DE SALDOS QUE SOLICITA EL USUARIO. LOS GRUPOS DE RANGOS SON TRES Y SON DETERMINADOS POR EL USUARIO.

.....

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- VERIFICA QUE LA IMPRESORA ESTA LISTA
SI LA IMPRESORA NO ESTA LISTA
DESPLIEGA MENSAJE "LA IMPRESORA NO ESTA LISTA, VERIFICAR"
- SOLICITA LA 1a. DISTRIBUCION DE SALDOS : 'LIMITE-SUPERIOR-1' Y
.....'LIMITE-INFERIOR-1'
- SOLICITA LA 2a. DISTRIBUCION DE SALDOS : 'LIMITE SUPERIOR-2' Y
.....'LIMITE-INFERIOR-2'
- SOLICITA LA 3a. DISTRIBUCION DE SALDOS : 'LIMITE SUPERIOR-3' Y
.....'LIMITE-INFERIOR-3'
- IMPRIME ENCABEZADOS DEL REPORTE
- LEE EL ARCHIVO DE "CLIENTES" SECUENCIALMENTE
SI 'LIMITE-SUPERIOR-1' < 'SDOACT' < 'LIMITE-INFERIOR-1'
HACE 'TOTALSALDOS-1' = 'TOTALSALDOS-1' + 'IMPMOV'
HACE 'CONT/CTA-1' = 'CONT/CTA-1' + 1
SI 'LIMITE-SUPERIOR-2' < 'SDOACT' < 'LIMITE-INFERIOR-2'
HACE 'TOTALSALDOS-2' = 'TOTALSALDOS-2' + 'IMPMOV'
HACE 'CONT/CTA-2' = 'CONT/CTA-2' + 1

(CONTINUA ALGORITMO EN LA SIGUIENTE HOJA)

DATOS LOCALES : 'LIMITE-SUPERIOR-1', 'LIMITE-INFERIOR-1',
'LIMITE-SUPERIOR-2', 'LIMITE-INFERIOR-2', 'LIMITE-SUPERIOR-3',
'LIMITE-INFERIOR-3', 'TOTALSALDOS1', 'TOTALSALDOS2', 'TOTALSALDOS3',
'CONT/CTA-1', 'CONT/CTA-2', 'CONT/CTA-3'

NOTAS :

.....

TARJETA "IPO" NUM. 024

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "REPORTE DE CUENTAS POR RANGOS"

FECHA : 15/08/88

* * CONTINUACION DEL ALGORITMO * *

SI 'LIMITE-SUPERIOR-3' < 'SDDACT' < 'LIMITE-INFERIOR-3'

HACE 'TOTALSALDOS-3' = 'TOTALSALDOS-3' + 'IMPMOV'

HACE 'CONT/CTA-3' = 'CONT/CTA-3' + 1

SI NO ES FIN DE ARCHIVO

LEE OTRO REGISTRO

- IMPRIME "ENCABEZADOS DEL REPORTE
- IMPRIME "CUENTAS CON SALDO ENTRE ", 'LIMITE-INFERIOR-1', "Y",
.....'LIMITE-SUPERIOR-1', "---->", 'TOTAL-SALDOS-1', 'CONT/CTA-1'
- IMPRIME "CUENTAS CON SALDO ENTRE ", 'LIMITE-INFERIOR-2', "Y",
.....'LIMITE-SUPERIOR-2', "---->", 'TOTAL-SALDOS-2', 'CONT/CTA-2'
- IMPRIME "CUENTAS CON SALDO ENTRE ", 'LIMITE-INFERIOR-3', "Y",
.....'LIMITE-SUPERIOR-3', "---->", 'TOTAL-SALDOS-3', 'CONT/CTA-3'
- TERMINA RUTINA

TARJETA "IPO" NUM. 025

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "REPORTE DE CUENTAS POR SECTEC."

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "DESPACHADOR"

LLAMA O INVOCA A : - - -

DATOS DE ENTRADA : - - -

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "DESPACHADOR" Y SU FUNCION ES
 GENERAR EL REPORTE DE LAS CUENTAS AGRUPADAS POR LOS SECTORES
 ECONOMICOS QUE MANEJA EL SISTEMA

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- VERIFICA QUE LA IMPRESORA ESTA LISTA
 - SI LA IMPRESORA NO ESTA LISTA
 - DESPLIEGA MENSAJE "LA IMPRESORA NO ESTA LISTA, VERIFICAR"
- IMPRIME ENCABEZADOS DEL REPORTE
- LEE EL ARCHIVO DE "CLIENTES" SECUENCIALMENTE
 - SI 'SECTEC' = 1
 - HACE 'TOTALSALDOS-1' = 'TOTALSALDOS-1' + 'IMPMOV'
 - HACE 'CONT/CTA-1' = 'CONT/CTA-1' + 1
 - SI 'SECTEC' = 2
 - HACE 'TOTALSALDOS-2' = 'TOTALSALDOS-2' + 'IMPMOV'
 - HACE 'CONT/CTA-2' = 'CONT/CTA-2' + 1
 - SI 'SECTEC' = 3
 - HACE 'TOTALSALDOS-3' = 'TOTALSALDOS-3' + 'IMPMOV'
 - HACE 'CONT/CTA-3' = 'CONT/CTA-3' + 1
 - SI NO ES FIN DE ARCHIVO
 - LEE OTRO REGISTRO
- IMPRIME "ENCABEZADOS DEL REPORTE"
 - IMPRIME "CUENTAS CON SECTOR ECONOMICO '01', "---->",
 -'TOTAL-SALDOS-1', 'CONT/CTA-1'
 - IMPRIME "CUENTAS CON SECTOR ECONOMICO '02', "---->",
 -'TOTAL-SALDOS-2', 'CONT/CTA-2'
 - IMPRIME "CUENTAS CON SECTOR ECONOMICO '03', "---->",
 -'TOTAL-SALDOS-3', 'CONT/CTA-3'
- TERMINA RUTINA

DATOS LOCALES : 'TOTALSALDOS-1', 'TOTALSALDOS-2', 'TOTALSALDOS-3',
 'CONT/CTA-1', 'CONT/CTA-2', 'CONT/CTA-3'

NOTAS :

TARJETA "IPO" NUM. 026

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "IMPRIME ESTADOS DE CUENTA"

FECHA : 15/08/88

LLAMADO O INVOCADO POR : "DESPACHADOR"

LLAMA O INVOCA A : - - -

DATOS DE ENTRADA : - - -

DATOS DE SALIDA : - - -

* * DESCRIPCION NARRATIVA DEL PROCESO * *

ESTE MODULO RECIBE CONTROL DEL MODULO "DESPACHADOR", SU FUNCION ES IMPRIMIR LOS ESTADOS DE CUENTA QUE SOLICITE EL USUARIO. ADEMAS TIENE LA OPCION DE BORRAR EL ARCHIVO "MOVIMIENTO MENSUAL" Y PASAR LOS SALDOS ACTUALES COMO SALDOS ANTERIORES

* * DESCRIPCION DEL ALGORITMO * *

- INICIA RUTINA
- VERIFICA QUE LA IMPRESORA ESTA LISTA
SI LA IMPRESORA NO ESTA LISTA
DESPLIEGA MENSAJE "LA IMPRESORA NO ESTA LISTA, VERIFICAR"
- SOLICITA Y ACEPTA EL 'NUMCTA'
- LEE EN ARCHIVO "CLIENTES" EL REGISTRO QUE CORRESPONDE A 'NUMCTA'
SI NO ENCUENTRA EL REGISTRO
DESPLIEGA MENSAJE "EL NUMERO DE CUENTA NO EXISTE, VERIFIQUE
TERMINA RUTINA
- IMPRIME 'NUMCTA', 'NOMBRE', 'DIRECCION', 'POBLACION', 'SDOANT'
- HACE 'SALDO' = 'SDOANT'
- LEE EN ARCHIVO "MOVIMIENTO MENSUAL" LOS REGISTROS QUE
.....CORRESPONDEN A 'NUMCTA', SECUENCIALMENTE
IMPRIME 'FEMOV', 'NUMDOC', 'IMPMOV'
SI 'NUMDOC' ESTA EN RANGO DE CHEQUES
HACE 'SALDO' = 'SALDO' - 'IMPMOV'
HACE 'TOTALCHEQUES' = 'TOTALCHEQUES' + 'IMPMOV'
SI 'NUMDOC' ESTA EN RANGO DE DEPOSITOS
HACE 'SALDO' = 'SALDO' + 'IMPMOV'
HACE 'TOTALDEPOSITOS' = 'TOTALDEPOSITOS' + 'IMPMOV'
SI 'FEMOV' CAMBIA CON RESPECTO AL REGISTRO ANTERIOR"
IMPRIME 'SALDO'
SI HAY MAS REGISTROS DE 'NUMCTA'
LEE OTRO REGISTRO DEL ARCHIVO "MOVIMIENTO MENSUAL"
- IMPRIME 'TOTALCHEQUES', 'TOTALDEPOSITOS', 'SDOACT'

(CONTINUA ALGORITMO EN LA PAG. SIG.)

DATOS LOCALES : 'SALDO', 'TOTALCHEQUES', 'TOTALDEPOSITOS'

NOTAS :

.....

TARJETA "IPO" NUM. 026

SISTEMA : CHEQUES

PREPARADO POR : G.T.V.

MODULO : "IMPRIME ESTADOS DE CUENTA"

FECHA : 15/08/88

* * CONTINUACION DEL ALGORITMO * *

SI 'SDOACT' ES DIFERENTE DE 'SDOACT'
IMPRIME "ESTADO DE CUENTA CON DIFERENCIA, VERIFICAR"
SI SE DESEA BORRAR LA INFORMACION DEL MES
BORRAR REGISTROS DE 'NUMCTA' DEL ARCHIVO "MOVIMIENTO MENSUAL"
HACER 'SDOANT' = 'SDOACT' EN ARCHIVO "CLIENTES"
- TERMINA RUTINA .

BIBLIOGRAFIA

- (1) "SOFTWARE ENGINEERING"
RICHARD E. FAIRLEY
Mc. GRAW HILL
- (2) "SYSTEMS ANALYSIS AND DESIGN
A STRUCTURED APPROACH"
WILLIAM S. DAVIS
ADDISON-WESLEY
- (3) "SOFTWARE ENGINEERING"
MARTIN I. SHOUMAN
Mc. GRAW HILL
- (4) "SOFTWARE ENGINEERING"
PRESSMAN
Mc. GRAW HILL
- (5) "THE SECOND STRUCTURED REVOLUTION"
EDWARD YOURDON
YOURDON INC.
- (6) "STANDARD GLOSSARY OF SOFTWARE ENGINEERING
TERMINOLOGY"
PUBLICACIONES IEEE"
- (7) "STRUCTURED DESIGN"
EDWARD YOURDON AND LARRY L. CONSTANTINE
PRENTICE HALL
- (8) "SOFTWARE ENGINEERING ECONOMICS"
BARRY BOYEM
PRENTICE HALL