

52
2 g/4



Universidad Nacional Autónoma de México

Facultad de Contaduría y Administración

**TECNICAS PARA EL DESARROLLO
DE SISTEMAS DE INFORMACION
COMPUTARIZADOS.**



Seminario de Investigación Administrativa

Que en opción al grado de:
LICENCIADO EN ADMINISTRACION

P r e s e n t a n :

**MONICA CANTERO SANDOVAL
DANIEL MORENO LOPEZ**

Profesor del Seminario: L.A.E. y M.B.A. José Antonio Echenique



México, D. F.

1985

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.

I N D I C E

INDICE GENERAL

CONTENIDO	PAGINA
CAPITULO I.	
INTRODUCCION AL DESARROLLO DE SISTEMAS.	
1. EL CONCEPTO DE SISTEMAS EN EL PROCESAMIENTO DE LOS DATOS	2
1.1 El Concepto de Sistema	2
1.2 El Procesamiento de Datos	3
1.3 El Sistema de Procesamiento de Datos	3
1.4 Elementos Humanos que Intervienen en un Sistema de Procesamiento de Datos	6
1.5 Sistemas Amigables	13
2. CARACTERISTICAS BUSCADAS EN UN SISTEMA	15
2.1 Confiable	17
2.2 Mantenable	17
2.3 Accesible	18
2.4 Disponible	18
2.5 Flexible	18
2.6 General	18
2.7 Portable	18
2.8 Tolerante	18
2.9 Validable	18
3. EL DESARROLLO DE SISTEMAS	19
3.1 Concepto de Desarrollo de Sistemas	19
3.2 El Proceso General de Desarrollo de Sistemas ..	20
3.3 El Desarrollo de Sistemas Visto como un Proyecto	27
3.4 El Empleo de Modelos en el Desarrollo de Sistemas y su Similitud con la Metodología Ingenieril	29

4. PROBLEMATICA DEL DESARROLLO DE SISTEMAS EN MEXICO.	32
4.1 Contexto General	32
4.2 Disponibilidad de Recursos Humanos	34
4.3 Planeación del Cambio	35
4.4 Proyección de los Costos de Automatización ...	35
4.5 Efectos Psicológicos en la Organización	35
4.6 Ubicación del Departamento de Sistemas dentro de la Organización	36
4.7 La Expectancia de Resultados Inmediatos por Parte de los Funcionarios	37
4.8 Deficiencias Profesionales	37
5. NECESIDAD DE EMPLEAR UNA TECNICA EN EL DESARROLLO DE SISTEMAS	38

CAPITULO II.
LA TECNICA ESTRUCTURADA.

1. FUNDAMENTOS	40
2. DESCRIPCION DE LA TECNICA	40
3. CICLO DE VIDA	41
3.1 Definición del Problema	47
3.2 Estudio de Factibilidad	56
3.3 Análisis	71
3.4 Diseño	93
3.5 Diseño Detallado	110
3.6 Implantación	172
3.7 Mantenimiento	181

CAPITULO III.
LA TECNICA JSD (JACKSON SYSTEM DEVELOPMENT)

1. DESCRIPCION DE LA TECNICA	185
------------------------------------	-----

CONTENIDO

PAGINA

2. PASOS DEL DESARROLLO	188
3. ETAPA ACCION-ENTIDAD	189
4. ETAPA ESTRUCTURA-ENTIDAD	194
4.1 Estructura de los Diagramas	195
4.2 Secuencias	196
4.3 Iteraciones	196
4.4 Selecciones	197
4.5 Estructuras Mínimas	198
4.6 Estructuras Marsupiales	198
4.7 Estructura-Entidad de la Cía. "Lo Vendo Barato S.A."	200
4.8 Estructura-Entidad-2 "lo Vendo Barato S.A." ..	205
5. ETAPA EL MODELO INICIAL	206
5.1 Texto de la Estructura	206
5.2 Conexión de Procesos	210
5.3 Conexión del Flujo de Datos	213
5.4 Conexión Vector-Estado	218
6. ETAPA DE FUNCIONES	231
6.1 Marcas de Tiempo	232
6.2 Funciones 1 y 2 (nivel 2) "Lo Vendo Barato S.A."	233
6.3 Funciones Impuestas	236
6.4 Función 3 (impuesta) "Lo Vendo Barato S.A." ..	237
6.5 Función Especializada # 1	239
6.6 Función Especializada # 2	245
7. ETAPA DE SINCRONIZACION DEL SISTEMA	247
8. ETAPA DE IMPLANTACION	248
8.1 Coordinación sin Coordinador	251
8.2 Implantación de Fusiones con Inversión	253
8.3 Canales	254

CONTENIDO

PAGINA

8.4 Coordinando con Buffers	256
8.5 Separación de Vectores-Estado	259
8.6 Implantación-1 "Lo Vendo Barato S.A."	261
CONCLUSIONES	268
BIBLIOGRAFIA	274

I N T R O D U C C I O N

INTRODUCCION

En la actualidad se vive un desarrollo acelerado en el área de la informática dentro de nuestro país, lo cual implica la necesidad de contar con profesionales especializados en esta rama para el desarrollo de sistemas en aplicaciones específicas.

Hasta hoy se han venido desarrollando sistemas dentro de instituciones privadas y gubernamentales las cuales invierten grandes sumas de dinero y los resultados que éstas obtienen muchas veces no son del todo halagadores debido a que las personas que se dedican a esto, aún cuando dominan la parte técnica, no manejan un proceso bien establecido en el desarrollo del sistema (semejante al proceso administrativo) que oriente su trabajo, por lo que muchos de sus esfuerzos no son aprovechados en su totalidad restando eficiencia a los proyectos. Lo anterior se ha constatado algunas veces por medio de observaciones y experiencias tenidas en este medio de trabajo.

Por lo tanto, se hace necesario transmitir las técnicas más avanzadas en el desarrollo de sistemas, así como los elementos necesarios para discernir cual de estas es la más adecuada al aplicarla sobre problemas específicos.

Sobre el contenido de esta investigación se puede decir, en primer término, que en el Capítulo No. I "Introducción al Desarrollo de Sistemas" se exponen los conceptos básicos del tema, objeto de este trabajo. No se duda de que estos conceptos sean comunes para aquellas personas que dominan el tema y que además puedan ser encontrados en un libro especializado en la materia, sin embargo, lo que se pretende no es tratarlos ampliamente, ya que no es el objetivo de este trabajo, sino compilarlos a manera de introducción para después pasar a la parte medular: Las Técnicas para el Desarrollo de Sistemas así como su análisis.

En los Capítulos II y III se habla de la técnica estructurada y de la técnica JSD (Jackson System Development), correspondientemente. Ambas pueden considerarse como dos maneras diferentes de desarrollar sistemas de información compu-

tarizados, ya que cada una posee una metodología propia y un conjunto de fases o etapas diferentes. Se desarrolla en estos dos capítulos el ciclo de vida de cada técnica especificando sus características.

En el primer proyecto de este trabajo se consideró tratar aquí mismo otras técnicas para el desarrollo de sistemas que en nuestro medio son aún desconocidas: Isac y Datatran, con la finalidad de analizarlas entre sí y poder proporcionar mayores criterios de elección entre éstas, sin embargo, debido a limitaciones de tiempo y a otras razones, nuestro proyecto no se consumó en la forma en la que se había planteado en sus inicios y sólo se presentan en este trabajo dos técnicas de desarrollo de sistemas de información computarizados: La Técnica Estructurada y La Técnica JSD, de las cuales se hace un análisis y una comparación entre ambas.

Finalmente, se desea aclarar que en lugar de emplear en este trabajo el término "Análisis y Diseño de Sistemas", se ha utilizado el término "Desarrollo de Sistemas" por considerarlo más amplio que el primero ya que contempla todas las actividades y fases del ciclo de vida de un sistema sin importar la técnica que se emplee.

C A P I T U L O I

INTRODUCCION AL DESARROLLO DE SISTEMAS

CAPITULO I

INTRODUCCION AL DESARROLLO DE SISTEMAS

1. EL CONCEPTO DE SISTEMAS EN EL PROCESAMIENTO DE DATOS

El concepto de "sistema" es muy estudiado en nuestras universidades dentro de muchas carreras de tipo humanístico, económico, ingenieril, etc. Específicamente la administración lo estudia también y se habla así mismo del "enfoque de sistemas". Por otro lado, muchos autores definen el concepto de sistemas y recalcan la utilidad que tiene el comprenderlo.

En este trabajo se hablará también de los sistemas pero desde otro punto de vista: Los sistemas en el procesamiento de los datos de cualquier organización.

1.1 CONCEPTO DE SISTEMA.- El término "SISTEMA" es derivado de la palabra griega SYSTEMA la cual se refiere a una organización, funcionamiento en afinidad entre unidades o componentes (1).

Elias Awad define a un sistema como un grupo de componentes organizados (subsistemas) ligados entre sí y acordes a un plan para lograr un objetivo específico (2).

Un estudio de las definiciones formuladas por varios eruditos en el tema conjuntó la siguiente definición que es bastante completa: "Un sistema es una serie de elementos que forman una actividad, procedimiento o plan de procesamiento y que buscan una meta o metas comunes mediante el manejo de datos, energía o materia en una referencia de tiempo, para proporcionar información, energía o materia" (3).

Pueden citarse aquí otras definiciones al respecto pero no es el objeto; lo importante es recalcar que todos aquellos que definen a un sistema están de acuerdo en que se trata de un conjunto de partes coordinadas para lograr un conjunto de metas.

West Churchman cita algunas consideraciones básicas a tomar en cuenta cuando se razone acerca del significado de

(1) AWAD ELIAS. SYSTEM ANALYSIS AND DESIGN. PAG. 4.

(2) AWAD. OP.CIT. PAG.4.

(3) A.C. HOPKINS. A SYSTEMATIC PROCEDURE FOR SYSTEM DEVELOPMENT. PAG. 85.

un sistema. Estas consideraciones se anotan aquí ya que pueden aplicarse a todo tipo de sistemas incluyendo aquellos que son para el procesamiento de los datos. Tales consideraciones son:

- a) Objetivos del Sistema y Medidas de Actuación de éste.
- b) Medio Ambiente del Sistema.
- c) Recursos del Sistema.
- d) Componentes y Actividades del Sistema.
- e) Su Administración (4).

Antes de desarrollar estas consideraciones referidas al enfoque de sistemas en general y aplicarlas al concepto de sistemas en el procesamiento de los datos, veamos primero lo que es el procesamiento de los datos.

1.2 EL PROCESAMIENTO DE DATOS.- El procesamiento de datos puede definirse como el manejo y transformación de los datos por medio de un sistema, ya sea manual, mecánico o electrónico con la finalidad de generar cierta información que será útil:

"El procesamiento de datos con computadora consiste en una metódica y bien planeada secuencia de actividades. Cada actividad va encaminada a proporcionar un dato que será empleado en la preparación del resultado final. todo este proceso no es un agrupamiento al azar de varias ideas, sino un proceso cuidadosamente planeado en el cual la computadora juega un importante papel" (5).

Hasta aquí se ha definido lo que es un sistema y lo que es el proceso de datos. Conjuntando ambos conceptos se definirá lo que es el Sistema De Procesamiento De Datos.

1.3 EL SISTEMA DE PROCESAMIENTO DE DATOS.- Se considera que un sistema de procesamiento de datos puede ser definido desde dos puntos de vista:

- a) Tal y como lo define el American National Standards Comitee: Un sistema de procesamiento de datos es un conjunto de hombres, máquinas y métodos organizados para desempeñar un grupo de funciones específicas.
- b) También se puede entender este concepto, de una manera más específica, como el conjunto de programas ma-

(4) CHURCHMAN WEST. EL ENFOQUE DE SISTEMAS. PAG. 4.

(5) S. ORILIA LAWRENCE. INTRODUCCION AL PROCESAMIENTO DE LOS-DATOS EN LOS NEGOCIOS. PAG. 24-25.

estros, subprogramas y archivos de datos que están entrelazados unos con otros para que, por medio del procesamiento, arrojen información útil a la organización. Todo ello por medio de un computador.

Michael Jackson cita el concepto de sistema en el procesamiento de los datos como aquello que es originado por el mundo real externo el cual es el que le da el motivo de su razón de ser. Se coincide con él al afirmar que el sistema en sí mismo consiste en los procedimientos manuales y de programación así como el hardware que interviene (6).

Finalmente dentro de este inciso se volverán a tratar las 5 consideraciones de West Curchman las cuales pueden ser aplicables a los sistemas de procesamiento de datos y así mismo, tomarse como requisitos a cubrir en todo sistema de esta índole:

a) Objetivos del Sistema y Medidas de Actuación de Este. - Todo sistema de procesamiento de datos debe tener un objetivo claramente definido el cual valga la pena alcanzar, considerando su costo/beneficio, sin embargo, no es cosa fácil determinar los verdaderos objetivos de un sistema y en especial cuando éste no está muy de acuerdo con el medio que lo rodea: la organización y sus necesidades de información.

En cuanto a las medidas de actuación del sistema éstas deben marcarse como estándares del mismo: Que es lo que se espera del sistema de procesamiento de datos?, Cual debe ser su medida óptima de actuación?, Como se medirá su eficiencia al final de su implantación?, etc. Estas preguntas y muchas otras al respecto pueden contestarse por medio de una cuidadosa planeación.

b) El Medio Ambiente del Sistema. - Todo lo que está fuera del sistema es algo que no puede ser controlado por éste ya que el medio ambiente le impone restricciones y le determina como debe operar y que precauciones debe tomar. Por ejemplo, si deseamos desarrollar un sistema automatizado de biblioteca que vaya a estar al servicio de muchos estudiantes de alguna facultad de la Universidad, el sistema debe considerar, para su realización el tipo de usuarios a los que se destinará para así proporcionarles la información requerida, y porque no, para tomar las precauciones en cuanto al manejo y cuidado

del equipo; también debe tomarse en cuenta el espacio que se tiene disponible para la instalación del hardware y muchos otros factores externos que determinarán muchas decisiones las cuales a su vez le darán forma al nuevo sistema de procesamiento de datos.

West Curchman identifica al medio ambiente por medio de las respuestas a dos preguntas:

"El medio ambiente no es el aire que respiramos, el grupo social al cual pertenecemos, o la casa en la cual vivimos, sin importar que tanto de esto puede parecer estar fuera de nosotros. En cada caso debemos preguntar: Podría yo hacer algo acerca de ello?, influye en mis objetivos?. Si la respuesta a la primera pregunta es "NO" y "SI" a la segunda entonces sí se está en el medio ambiente" (7).

c) Recursos del Sistema.- Son aquellos elementos que fueron empleados al crear y organizar el sistema de procesamiento así como aquellos que se emplean constantemente para lograr su adecuado mantenimiento y funcionamiento.

Cuando se habla de recursos hablamos de dinero, horas-hombre, y equipos hardware y software.

Cuando se planea un sistema de procesamiento de datos es importante que el administrador del sistema decida cuanto tiene disponible de cada recurso y con ello podrá fijar las restricciones que tiene sobre la realización y actividades del sistema. La siguiente cita, aunque está generalizada en su contenido puede aplicarse al tipo de sistemas que aquí estamos tratando:

"Al observar y pensar acerca de un sistema el administrador científico pone atención no sólo a los recursos existentes, sino también a la forma en la que éstos pueden aumentarse... por medio de investigaciones y desarrollo en el caso de cierto tipo de equipo, o bien, mediante el entrenamiento y educación del personal..." (8).

Por tanto, un sistema de procesamiento de datos implantado puede estar constantemente enriqueciéndose tanto de manera tangible (compra de nuevos equipos de cómputo más modernos, mejoramiento del lugar de trabajo, etc.) como de manera

(7) CURCHMAN. OP.CIT. PAG. 55.

(8) CHURCHMAN. OP.CIT: PAG. 59.

intangibles (capacitación, eficiencia en los programas, etc.).

d) Componentes y Actividades del Sistema. - Es bien sabido que todo sistema se compone de partes o subsistemas. En un sistema de procesamiento de datos es importante conocer los subsistemas que forman parte de él (en éste contexto como actividades del sistema) para comprender las funciones y objetivos del sistema en general. Los subsistemas pueden desglosarse en función de las misiones, tareas o actividades básicas que el sistema de procesamiento realice, de esta manera se puede destinar el valor de cada actividad (un programa del computador que cumpla con "x" función, por ejemplo) para el logro del funcionamiento eficiente del sistema en general (contribución al objetivo del sistema total).

Churchman puntualiza de una manera cierta lo siguiente: "...La única razón de separar el sistema en componentes es para proporcionar al analista el tipo de información que necesita para poder decir si el sistema está operando adecuadamente y lo que se debe hacer a continuación" (9).

e) La Administración del Sistema. - Se refiere principalmente a la planeación del sistema (en este caso del sistema de procesamiento de datos) que abarcaría todo lo que ya se ha mencionado anteriormente: objetivos generales, medio ambiente, utilización de recursos y componentes y actividades del sistema. La administración del sistema es la que establece los objetivos, asigna los recursos, controla y evalúa al sistema, por ello es una actividad fundamental dentro de todo desarrollo de sistemas de procesamiento de datos específicamente, ello implica la necesidad de la existencia de un líder o administrador de proyectos de sistemas computarizados, que además de efectuar todas las actividades anteriores garantice que todos los planes se lleven a cabo de acuerdo con las ideas originales, y por último, otra actividad muy importante que señala Churchman para todo sistema: "...la planeación para el cambio de planes..." (10).

1.4 ELEMENTOS HUMANOS QUE INTERVIENEN EN UN SISTEMA DE PROCESAMIENTO DE DATOS.- "...las computadoras no pueden funcionar independientemente de los seres humanos. Los hombres y mujeres que trabajan con las computadoras son imprescindibles en el proceso de datos..." (11).

(9) CHURCHMAN. OP.CIT. PAG. 61.

(10) CHURCHMAN. OP.CIT. PAG. 65.

(11) ORILIA. OP.CIT. PAG. 37.

En efecto, es obvio que la computadora por si sola (por lo menos en la actualidad) no puede realizar mucho debido a que se requiere de un cuerpo de especialistas que planeen, organicen y coordinen su funcionamiento, por lo cual se considera importante mencionar el personal y los puestos que existen en el campo del procesamiento de datos y su relación entre sí. El personal que interviene en este campo es el siguiente:

a) El Usuario del Procesamiento de Datos.- En algún momento de nuestra vida todos llegamos a ser usuarios de un sistema de procesamiento de datos: cuando vamos al banco, cuando solicitamos alguna calificación en nuestra universidad (historiales académicos) o simplemente cuando hablamos por teléfono al servicio de LOCATEL por citar algunos ejemplos.

El usuario de un sistema de procesamiento de datos se caracteriza porque siempre demanda información; pero para que esta información le sea realmente útil debe saber comunicar con suma claridad sus necesidades a los "especialistas" en esta tarea.

Por otro lado, los usuarios son de muchos tipos, desde gente común y corriente, pasando por estudiantes, hasta administradores o directores de empresas los cuales deben poseer conocimientos de la materia para saber dar a conocer a los técnicos que es lo que necesitan (cuando se trate el tema de "Problemática del Desarrollo de Sistemas en México" se hablará más a fondo de este punto).

b) El Grupo de Operación.- En términos generales este grupo está compuesto por el operador de la computadora, el capturista, (en casos inoperantes el perforista), el cintotecario y el controlador de datos, principalmente.

Es este grupo el que mantiene un mayor contacto con la computadora y con la información que entra y sale de ésta. Las funciones teóricas de cada miembro se muestran en el CUADRO I.1.

CUADRO I.1

FUNCIONES QUE EFECTUA CADA UNA DE LAS PERSONAS QUE FORMAN PARTE DEL GRUPO DE OPERACION

ELEMENTO	FUNCIONES QUE DESEMPEÑA
OPERADOR	<ul style="list-style-type: none"> - PREPARACION Y LIMPIEZA DEL HARDWARE. - MANTENIMIENTO DE BITACORAS E INFORMES DE LA COMPUTADORA. - MANEJO EFICIENTE DE LOS RECURSOS REQUERIDOS EN LA SALA DE COMPUTO. - MANEJO DE LA CONSOLA, MONTAR Y QUITAR CINTAS Y/O DISCOS DURANTE EL PROCESO, MANEJO DEL PAPEL PARA LA IMPRESORA. - ES EL RESPONSABLE DEL EQUIPO.
CAPTURA	<ul style="list-style-type: none"> - SUMINISTRO DE DATOS AL COMPUTADOR DE ACUERDO A UN FORMATO ESTABLECIDO. - REALIZACION DE UNA AMPLIA GAMA DE TAREAS DE ENTRADA DE DATOS UTILIZANDO DIVERSOS TIPOS DE TECLADOS. - ES EL RESPONSABLE DE PROPORCIONAR AL COMPUTADOR LOS DATOS DE UNA MANERA CORRECTA.
CINTOTECARIO	<ul style="list-style-type: none"> - CATALOGACION Y ALMACENAMIENTO DE LAS CINTAS EMPLEADAS EN EL PROCESAMIENTO. - MANEJO DE LOS SUMINISTROS Y LOS MEDIOS DE ALMACENAMIENTO QUE PRODUCEN OTROS DISPOSITIVOS COMO LOS DISCOS MAGNETICOS.

(Continúa en la siguiente página)

ELEMENTO	FUNCIONES QUE DESEMPEÑA
C O N T R O L A D O R D E D A T O S	<ul style="list-style-type: none"> - CONTROL DE LA INMENSA CANTIDAD DE DATOS QUE ENTRAN Y SALEN EN UN CENTRO DE PROCESO DE DATOS. - REGISTRO DE LA ENTRADA DE DATOS QUE VAN A PROCESARSE, SOLICITUD DE LOS PROGRAMAS QUE REALIZARAN EL PROCESO ASI COMO DE LOS DATOS USADOS EN EL. - ORDENAMIENTO DE LOS RESULTADOS DE PROGRAMAS Y DE LOS REPORTES DE LA COMPUTADORA DETERMINANDO SU DESTINO. - PROPORCIONAR INFORMACION A TRAVES DE UNA TERMINAL ACERCA DE LOS DATOS RECIBIDOS POR EL DEPARTAMENTO. - USO DE LA COMPUTADORA PARA SEGUIR LA PISTA DE TRABAJOS PENDIENTES, TERMINADOS Y DE AQUELLOS QUE ESTEN LISTOS PARA PROCESARSE Y EN GENERAL PARA ALMACENAR INFORMACION DEL TRABAJO QUE SE ESTE PROCESANDO EN ESE MOMENTO.

Lawrence Orillia define a cada uno de los puestos del grupo de operación como sigue:

- Operador de la Computadora.- Persona del grupo de operación que opera físicamente la computadora y los equipos relacionados con ella.
- Capturista.- Persona del grupo de operación que emplea diversos tipos de teclado para proporcionar los datos directamente al sistema de computación.
- Cintotecario.- La persona del área de operación responsable de catalogar y almacenar los medios de almacenamiento de la computadora.
- Controlador de datos.- Persona del área de operación

que controla el flujo de datos a través del departamento de proceso de datos (12).

Finalmente, el tamaño del grupo de operación dependerá en gran medida de la magnitud del equipo hardware que se maneje, del sistema en juego y del tamaño de la empresa.

c) El Grupo de Programación .- Como su nombre lo indica forman parte de él aquellas personas que elaboran las instrucciones para que la computadora efectúe las funciones deseadas, es decir, escriben los programas que se ejecutan en la computadora.

Yourdon define a un programador de la siguiente manera: "Es el encargado de precisar las especificaciones de los módulos individuales las cuales incluyen información sobre entradas, salidas, interfaces con otras partes del sistema y el algoritmo a través del cual el módulo hará su trabajo" (13).

Las funciones principales del grupo de programación son las siguientes:

- Elaborar nuevos programas cuando se requiera.
- Modificar los programas existentes según sea necesario
- Asegurarse de que todos los programas operen correctamente.

En suma: escribir, probar y verificar que los programas estén listos para ser usados.

En algunas compañías grandes existen algunas clasificaciones dentro del grupo de programación en cuanto a la experiencia de cada uno de sus miembros. Estas clasificaciones son por lo general las siguientes:

- Programadores Junior o aprendices de programación.
- Programadores "B" (nivel medio).
- Programadores Senior.

Finalmente es común que en proyectos grandes el grupo de programación esté coordinado por un líder o programador principal. Cada miembro del equipo atiende en forma particular una parte del sistema y debe reportar al líder sobre el avance de su trabajo. Este líder del grupo de programación debe

(12) ORILIA. OP.CIT. PAG 66-67.

(13) YOURDON. STRUCTURED DESIGN. PAG. 7.

reportar a su vez al líder general del proyecto.

d) El Grupo de Sistemas.- Tiene por objetivo lograr un flujo eficiente de datos a través de toda la organización.

Este grupo es el que inicia el trabajo de desarrollo de sistemas solicitado por la alta gerencia; son los denominados proyectos en el área de informática y computación en los cuales interviene el analista de sistemas.

El analista de sistemas "...es la persona cuyo trabajo consiste en hablar con el usuario descubriendo así sus necesidades y requerimientos, para entonces expresarlos en términos bien organizados" (14).

El analista de sistemas debe elaborar una definición del sistema y del medio ambiente en el cual éste va a operar. Requerirá de elaborar una investigación de la estructura organizacional, del flujo de datos, del personal que será el usuario del sistema, del equipo de cómputo, del software, de las condiciones físicas de operación y otras consideraciones necesarias para lograr una definición, desarrollo y operación exitosa del sistema. Podrá así mismo, recurrir al proceso de desarrollo de sistemas para proveerse de un patrón o guía al desarrollar su investigación y documentación.

La idea del párrafo anterior acerca de las actividades del analista de sistemas es dada por Norman L. Enger quien también nos señala otros aspectos importantes al respecto:

"El analista seleccionará para la investigación aquellas funciones que probablemente generen los mejores beneficios del uso de los nuevos procedimientos y equipos, y estudiará esas funciones en detalle para determinar si debieran ser automatizadas... El punto de partida del analista es el estudio del sistema vigente y un análisis de dónde está la falla para alcanzar el objetivo del negocio... analizará la existencia de operaciones, explorará alternativas, métodos y procedimientos y considerará la posibilidad de eliminar actividades básicas e intermedias... deberá determinar el grado o rango del personal que podría ser afectado por el nuevo sistema..." (15).

(14) YOURDON. OP.CIT. PAG. 5.

(15) COTTERMAN. OP.CIT. PAG. 6.

Norman cita en el párrafo compactado anteriormente, los aspectos más importantes que deben caracterizar a un buen analista de sistemas tomando en cuenta así mismo que debe considerar a las personas que se verían afectadas por el nuevo cambio.

Por otro lado, las funciones que desempeña el grupo de sistemas son las siguientes:

- Detectar necesidades en cuanto a sistemas de información se refiere.
- Análisis de los flujos de información para proponer mejoras o cambios, o bien, crear éstos.
- Elección y cambio de equipos de cómputo y análisis del costo beneficio de ello.
- Estudios de factibilidad.
- Elaboración de programas de trabajo para la instalación de equipos y reentrenamiento de personal.
- Diseño de nuevos sistemas, implantación de los mismos, mantenimiento y vigilancia en su operación.

Los analistas de sistemas, al igual que los programadores pasan por diversos grados de experiencia, entrenamiento y educación:

- Analista Junior.
- Aprendiz de Sistemas.
- Analista Senior.

Lo anterior se da, por lo general, en empresas de gran magnitud que poseen una gerencia, área o departamento de proceso de datos para apoyar su giro (En la Banca por ejemplo) o bien en empresas cuyo giro es proporcionar servicios en el área de cómputo e informática.

Lo que sucede en organizaciones pequeñas es muy diferente; muchas veces una persona o dos son las que efectúan todo el trabajo del sistema de procesamiento de datos debido a que su magnitud es tal que no requiere de puestos tan definidos

como los dados a lo largo de este subtema, siendo así que una persona puede ser analista, programador y capturista simultáneamente y ser ayudado esporádicamente por un técnico que atiende el mantenimiento del equipo el cual no requerirá de excesivos cuidados pues en este caso generalmente se trata de microcomputadoras, aunque no siempre.

En la mayoría de los casos, el tipo de personas a las que nos referimos en el párrafo anterior, se desenvuelven en organizaciones pequeñas o de poco presupuesto y en donde no se requiere de tanto personal, además puede que lleguen a adquirir experiencia en todas las etapas del desarrollo de un sistema. No tienen un puesto definido pues hacen de todo.

1.5 SISTEMAS AMIGABLES.- Los sistemas amigables son aquellos en donde el usuario interactúa directamente con el sistema sin necesidad de una tercera persona para la captura de datos.

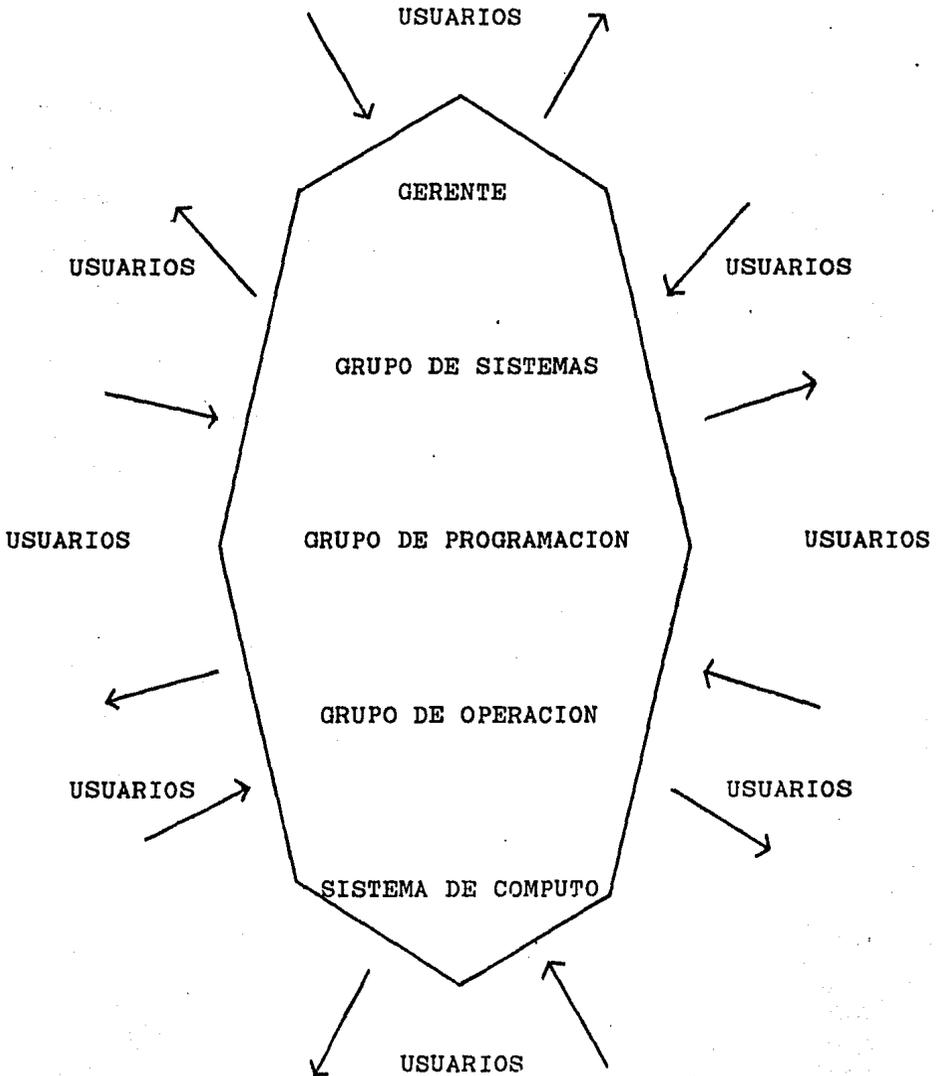
Un sistema amigable puede denominarse como un conjunto de programas realizados para una aplicación específica y que está dirigido a los usuarios finales. El sistema guía al usuario y le da las instrucciones que debe seguir para lograr lo que él desea del sistema; por ejemplo, como preparar la impresora para mandar un listado, que hacer para dar de baja un registro dentro de un archivo, que hacer cuando se equivoca al teclear un dato, etc.

El tipo de equipo en el que se desarrolla el sistema es independiente de la decisión de implantar un sistema amigable, lo importante es hacer notar que se considera principalmente al tipo de usuario a quien va dirigido, desde niños (sistemas de juegos educativos) hasta empleados bancarios que consultan o actualizan información.

Finalmente, es conveniente agregar que el desarrollo de sistemas amigables probablemente siga incrementándose tanto en las compañías que desarrollan software comercial como dentro de empresas que solicitan el desarrollo de sistemas particulares para alguna de sus áreas. La razón principal que se considera respecto al hecho anterior, es que si estos sistemas son bien empleados resultarán ser funcionales y accesibles a la mayoría de los usuarios.

GRAFICA I.1

ESQUEMA QUE MUESTRA LA RELACION ENTRE CADA UNO DE LOS ELEMENTOS QUE INTEGRAN EL SISTEMA DE PROCESAMIENTO DE DATOS



2. CARACTERISTICAS BUSCADAS EN UN SISTEMA

Dentro de la fase de "Diseño" se determinan las características generales del sistema a desarrollar, esto se hace a través de una serie de decisiones de carácter técnico tomadas por el diseñador en base a los requerimientos del usuario, recopilados por el analista, y que en la medida de su conjunto describirán la calidad y cualidad del futuro sistema.

Un diseñador profesional se caracteriza por su insatisfacción de sólo encontrar "una solución" al problema, él buscará la "mejor solución" en base a los términos establecidos y dentro de las limitaciones reconocidas. Por desgracia, cuando un sistema "parece que funciona", la mayoría de los diseñadores se limitan a dar por hecho que este sistema fue un "buen" trabajo.

Debe medirse que es lo que estamos haciendo, pero no sólo aquello fácilmente reconocible como es el espacio y el tiempo consumido, sino aquellos factores críticos, principalmente los relacionados con los recursos del talento humano que en los últimos tiempos ha cobrado gran interés debido a su escasez y altos costos comparándolos con los recursos de hardware, los cuales con los últimos adelantos técnicos se han hecho más accesibles y de bajo costo.

Dentro de una organización los gastos por concepto de desarrollo y mantenimiento de los sistemas han cobrado mayor interés que los concernientes a los equipos de cómputo, como lo menciona en su libro Edward Yourdon: "En 1971, la fuerza aérea de los Estados Unidos estimó que para 1980 sólo el 20% del presupuesto asignado para el proceso de datos sería gastado en equipos de cómputo, y el 80% restante sería gastado en la gente dedicada al desarrollo y operación de los sistemas. Similarmente, un estudio hecho en la General Motors encontró que en 1970, los gastos por concepto de equipo de cómputo fueron de sólo el 50% del presupuesto total asignado a esa área, y previendo que para un futuro ese porcentaje disminuyera" (16).

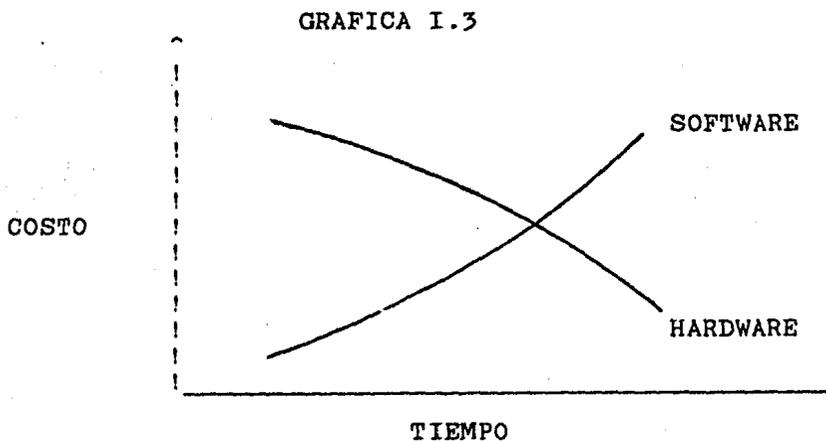
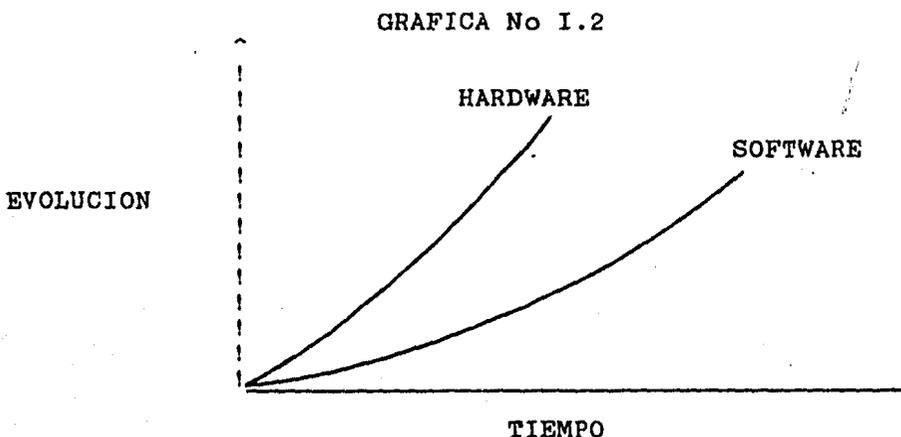
Tal ha sido el progreso de la computación, que se dice, por ejemplo que si la industria de la aviación hubiera evolucionado en los últimos 25 años (1960-1985) a la par de la computación, un Boeing 727 costaría 500 dls. y daría una

(16) YOURDON. OP.CIT. PAG. 11.

(17) CIFCA. HACIA EL DESARROLLO DE SISTEMAS CON HERRAMIENTAS AUTOMATICAS. FEB-1985. PAG. 1.

vuelta al mundo en 20 minutos con un consumo de combustible de 5 galones" (17).

Antiguamente el costo de un servicio computacional se basaba en el tiempo de procesador central que era utilizado para un proceso, actualmente, este parámetro ha sido relegado a segunda importancia por la evolución en el campo del hardware, estando por lo tanto, los factores de operación como el software, personal, periféricos, etc. en primer lugar debido a lo disparado de sus costos ocasionados por su nivel de evolución mas lenta. Todo esto puede verse reflejado en las GRAFICAS I.2 y I.3: DESARROLLO DEL HARDWARE Y SOFTWARE (18).



(18) CIFCA. FEB-85. OP.CIT. PAG. 2.

Ante la evolución más que proporcional del hardware con respecto al software, se ha incrementado el número de usuarios potenciales que adquieren un equipo de cómputo y que necesitan del apoyo lógico que satisfaga sus requerimientos, pero por la baja evolución del software, el desarrollo de éste por parte del personal calificado se ha vuelto escaso, resultando un incremento de la cola de espera de aplicaciones a realizar.

Si a lo anteriormente expuesto agregamos que gran parte del tiempo del personal del área de sistemas es dedicado a la detección y corrección de errores, mantenimiento y adecuación de los sistemas ya existentes, encontramos la importancia que tiene el buscar la mejor opción al definir las características del sistema que se está desarrollando. Estas características son las siguientes:

2.1 CONFIABLE.- Es la probabilidad de que un programa opere por un cierto período de tiempo sin que ocurra un error de lógica en la máquina para la que fue diseñado siendo usado dentro de los límites definidos.

Si ocurriese un "error", el sistema debe de ser capaz de recobrase automáticamente y continuar ejecutandose sin intervención humana. Podemos calcular la confiabilidad de un sistema a través de la siguiente fórmula:

$$\text{CONFIABILIDAD} = 1 - \frac{\text{NUMERO DE ENTRADAS CON FALLAS}}{\text{NUMERO TOTAL DE ENTRADAS}}$$

2.2 MANTENIBLE.- Es la probabilidad de que, cuando la acción de mantenimiento es iniciada bajo condiciones establecidas, un sistema con fallas pueda volver a ser operable en un tiempo especificado. Se mide en tiempo y está en función de factores primarios tales como:

- a) El diseño del sistema.
- b) El personal encargado.
- c) Facilidades de soporte como versiones alternativas del programa, archivos de respaldo, equipo de diagnóstico, etc.

2.3 ACCESIBLE.- Es el grado de facilidad o dificultad con el cual el sistema puede ser reparado.

2.4 DISPONIBLE.- Es la probabilidad de que un sistema esté operando satisfactoriamente en cualquier tiempo cuando es usado bajo condiciones establecidas. Está dado por la siguiente relación :

$$\text{DISPONIBILIDAD} = \frac{\text{TIEMPO MEDIO ENTRE FALLAS}}{\text{TIEMPO MEDIO ENTRE FALLAS} + \text{TIEMPO MEDIO PARA REPARACIONES}}$$

2.5 FLEXIBLE.- Es la capacidad que tiene un sistema de aceptar nuevas funciones, o de adaptarse a los cambios que se suceden en su medio ambiente.

2.6 GENERAL.- Se refiere a la posibilidad de que el sistema se pueda aplicar a "n" ambientes, ésta es una característica deseada, aunque muchas veces se llegan a desarrollar sistemas para aplicaciones específicas pues las circunstancias y el medio ambiente así lo requieren.

2.7 PORTABLE.- Es la capacidad que tiene un sistema para convertirse o adaptarse de un equipo a otro. En muchas ocasiones, dentro de una organización se llega a cambiar el equipo de cómputo por otro más moderno o de mejores características, pues las circunstancias así lo requieren, presentándose el problema de que los sistemas no pueden adaptarse fácilmente al nuevo equipo.

2.8 TOLERABLE.- Se refiere a la capacidad que tiene un sistema de aceptar datos de entrada con la misma información pero en forma diferente.

2.9 VALIDABLE.- Es la capacidad que debe tener un sistema de detectar cualquier circunstancia que pueda provocar un error o falla del mismo. En muchas ocasiones el usuario del sistema por desconocimiento, descuido, o intento de sabotaje llega a alimentar el sistema con datos inválidos, combinaciones de datos no previstos o manejar versiones incorrectas de archivos de datos, por lo que el sistema debe ser capaz de advertir este tipo de situaciones.

3. EL DESARROLLO DE SISTEMAS

3.1 CONCEPTO DE DESARROLLO DE SISTEMAS.- Michael Jackson define al desarrollo de sistemas de la siguiente manera:

"...es usado este término para englobar todas las actividades referentes al análisis y diseño de sistemas, diseño de programas y programación, excluyendo actividades como selección de proyectos, planeación, administración de los mismos y análisis de costo-beneficio" (19).

Esta definición es usada en la técnica JSD (Jackson System Development) de desarrollo de sistemas, sin embargo, no puede decirse que tal definición deba generalizarse ya que otras técnicas como la estructurada consideran algunas otras etapas no incluidas por ejemplo en JSD.

A grandes rasgos el desarrollo de sistemas es una metodología que se sigue para la creación de sistemas computarizados, esta metodología se refiere al ciclo completo de vida del sistema desde su concepción inicial hasta su última disposición, tal y como lo menciona G. F. Hice, Turner y Cashwell (20).

Existen algunos criterios básicos para lograr éxito en el desarrollo de todo sistema, dichos criterios son los siguientes:

a) Proyectar un Plan de Largo Alcance. - Este plan debe contener todas las actividades a realizar y el tiempo en el que se cubrirá cada una: determinación de horas de trabajo, hardware requerido, procedimientos, diagramas de flujo, personal requerido, etc.

b) Organizar el Plan. - Determinar el equipo de trabajo para el proyecto: Número de analistas, programadores, consultores, administradores, y usuarios que estarán en contacto con el nuevo sistema y que conocen las necesidades de la organización al respecto, así como definir y distribuir funciones individuales y tiempos de terminación de dichas funciones.

c) Desarrollo del Proyecto. - En este punto es conveniente hacer juntas durante todo el tiempo de realización del

(19) JACKSON. OP.CIT. PAG. IX.

(20) HICE, TURNER Y CASHWELL. SYSTEM DEVELOPMENT METHODOLOGY. PAG.1.

proyecto para mantener comunicados entre sí a todos los miembros del equipo, conocer sus inquietudes, dudas e ideas. Finalmente, un factor esencial en esta etapa es el control en todos los aspectos: en la calidad, el costo, diseño configuración y progreso paulatino en el trabajo.

3.2 EL PROCESO GENERAL DEL DESARROLLO DE SISTEMAS.- En este punto se dará a conocer cual es el proceso genérico que se sigue al desarrollar un sistema de información.

El proceso general de desarrollo de sistemas es conocido también con los nombres de: ciclo de vida del sistema, ciclo de ejecución ó ciclo de desarrollo. Philip W. Metzger menciona que el ciclo de desarrollo de la programación es una serie de actividades interrelacionadas y ordenadas que conducen a la terminación de una serie de programas, y que los períodos en los que estas actividades tienen lugar se denominan "fases" (21).

Es bien sabido que cada autor propone un proceso de desarrollo de sistemas distinto, con un número de etapas y con una denominación diferente de cada una. Por lo anterior puede decirse que la definición de las etapas y la organización de las actividades dentro de cada una de éstas por parte de los autores muchas veces es arbitraria. En el fondo el proceso de desarrollo de sistemas, (Ciclo de Vida del Sistema) es en su esencia inamovible. Expondremos algunos ejemplos citando a los siguientes autores del tema:

a) Norman L. Enger.- Este autor cita el ciclo de vida clásico o tradicional de un sistema (Proceso tradicional o clásico del desarrollo de sistemas) y lo divide en seis fases cada una de las cuales consta de un cierto número de actividades anotando así mismo los documentos que se generan en estas fases. Tal ciclo puede observarse en el CUADRO I.2.

b) Elias Awad.- Este autor también cita, según su criterio y experiencia, los pasos principales a seguir en todo desarrollo de sistemas. El proceso puede observarse en el CUADRO I.3.

c) Philip W. Metzger.- También propone el ciclo de desarrollo en seis fases las cuales son mostradas en el CUADRO I.4.

(21) W.METZGUER PHILIP. ADMINISTRACION DE UN PROYECTO DE PROGRAMACION. PAG. 17.

Los tres cuadros que se muestran en las páginas siguientes se explican por sí solos. Si los observamos con detenimiento nos daremos cuenta que en su esencia son iguales y que los tres tienen desde una fase de definición hasta una fase de instalación y operación sólo que hay una variación en la denominación de las fases y de las actividades dentro de cada una de éstas.

Por lo tanto, el ciclo de vida de un sistema de información será siempre el mismo.

CUADRO I.2
CICLO DE VIDA CLASICO DE UN SISTEMA

FASES	ACTIVIDADES PRINCIPALES	DOCTOS. GENERADOS
R E N U A E L R I I S M I I S E N D T E O S	- EVALUAR LOS REQUERIMIENTOS DEL USUARIO. - CONducIR EL ESTUDIO DE FACTIBILIDAD. - DEFINIR REQUERIMIENTOS DEL USUARIO. - PREPARAR EL PLAN DEL PROYECTO.	- REQUERIMIENTOS DEL USUARIO. - PROPUESTA Y ESTIMACION DE COSTOS. - ESTUDIO DE FACTIBILIDAD Y ANALISIS DE REQUERIMIENTOS. - PLAN DEL PROYECTO.
D L I O S G E I N C O O	- PREPARAR LAS ESPECIFICACIONES DEL DISEÑO GENERAL. - REFINAR LOS REQUERIMIENTOS DEL SISTEMA DEL USUARIO.	- DESCRIPCION FUNCIONAL. - DOCUMENTO DE REQUERIMIENTOS DE DATOS.

(Continúa en la siguiente página)

FASES	ACTIVIDADES PRINCIPALES	DOCTOS. GENERADOS
DISEÑO	<ul style="list-style-type: none"> - PREPARAR LAS ESPECIFICACIONES DEL DISEÑO DETALLADO. - DEFINIR LOS SUBSISTEMAS. - DISEÑAR LA ESTRUCTURA DE LA BASE DE DATOS. 	<ul style="list-style-type: none"> - ESPECIFICACIONES DE SISTEMAS Y SUBSISTEMAS. - ESPECIFICACION DE LA BASE DE DATOS. - ESPECIFICACIONES DE LOS PROGRAMAS.
DESARROLLO	<ul style="list-style-type: none"> - CODIFICAR LOS PROGRAMAS. - PROBARLOS. - DOCUMENTARLOS. 	<ul style="list-style-type: none"> - DOCUMENTACION DE PROGRAMAS.
IMPLEMENTACION	<ul style="list-style-type: none"> - EJECUTAR PRUEBAS DE SUBSISTEMAS. - EJECUTAR PRUEBAS DEL SISTEMA. - CAPACITACION DEL PERSONAL QUE SERA EL USUARIO DEL SISTEMA. - ESTABLECER CONTROLES DE CONVERSION. - EJECUTAR CONVERSION DE DATOS. 	<ul style="list-style-type: none"> - PLAN PRUEBA. - REPORTE DE PRUEBAS DE ANALISIS. - MANUAL PARA EL USUARIO DEL SISTEMA.
OPERACION	<ul style="list-style-type: none"> - OPERAR EL SISTEMA YA EXISTENTE. - MANTENERLO. - EVALUARLO. 	<ul style="list-style-type: none"> - MANUALES DE OPERACION. - MANUAL DE MANTENIMIENTO. - REPORTE DE EVALUACION.

CUADRO I.3

PROCESO DE DESARROLLO DE SISTEMAS SEGUN ELIAS AWAD

FASES	CARACTERISTICAS
I M P E L E T U C S A M P A B I O A	- LA IDEA DEL CAMBIO ES ORIGINADA POR DOS FACTORES: >>MEDIO AMBIENTE: PROVOCA UN IMPACTO DEFINITIVO SOBRE LA EXISTENCIA DEL SISTEMA.EJEMPLOS: AGENCIAS GUBERNAMENTALES,COMPETENCIA CONSUMIDORES,NUEVA TECNOLOGIA,NECESIDAD DE CAMBIO PARA ATENDER LAS DEMANDAS DEL MEDIO. >>ORGANIZACION:CAMBIOS EN EL VOLUMEN DE VENTAS,TASAS DE PRODUCCION,AVANCES EN LA TECNOLOGIA DE COMPUTADORES, SENTIMIENTO DEL ADMINISTRADOR DE LA NECESIDAD DE ACTUALIZAR APLICACIONES O MEJORAR PROCEDIMIENTOS.
I N S P E D E L C S I O N S T E M A I C I A L	- ES UNA FASE SEMEJANTE A UN ESTUDIO DE SISTEMAS COMPLETO EXCEPTO QUE ES DE MAS CORTA DURACION. - ES UNA RUTINA DE PREANALISIS A FIN DE DETERMINAR SI EL SISTEMA EN USO ES EFICIENTE. - INCLUYE UNA MIRADA A LA DUPLICACION DE LABORES,REDUNDANCIA EN LA EXISTENCIA DE PROCEDIMIENTOS O INEFICIENCIAS EN LA EJECUCION. - ES LLEVADA A CABO POR USUARIO Y ANALISTA, EL PRIMERO INTRODUCE AL SEGUNDO EN EL PROBLEMA. - AL FINAL SE INDICAN LAS VENTAJAS Y LIMITACIONES DEL SISTEMA EXISTENTE Y LA JUSTIFICACION PARA EL DISEÑO DE UN NUEVO SISTEMA. - LO ANTERIOR SE PLASMA EN UN REPORTE DENOMINADO "REPORTE DEL RECONOCIMIENTO INICIAL DEL SISTEMA".

(Continúa en la siguiente página)

FASES	CARACTERISTICAS
ANÁLISIS	<ul style="list-style-type: none"> - EVALUACION PLANEADA DE LA APLICACION DE LOS NEGOCIOS PARA DESCUBRIR Y ENTENDER LAS AREAS DEL PROBLEMA. - ES UN PREREQUISITO PARA EL DISEÑO. - SUS ETAPAS SON: <ol style="list-style-type: none"> 1) DEFINICION DEL PROBLEMA INICIAL. 2) RECOLECCION DE LOS DATOS POR MEDIO DE ENTREVISTAS, OBSERVACIONES DIRECTAS Y ANALISIS DE FORMAS Y PROCEDIMIENTOS. 3) ORGANIZACION DE LOS DATOS (DIAGRAMAS DE FLUJO Y TABLAS DE DECISION). 4) ANALISIS DEL COSTO/BENEFICIO. 5) DEFINICION FINAL DEL PROBLEMA. 6) REPORTE FINAL A LA ADMINISTRACION.
DISEÑO	<ul style="list-style-type: none"> - SE PROPONE UN METODO PARA LA IMPLANTACION DEL SISTEMA. - ES EL CONJUNTO DE ESPECIFICACIONES TECNICAS QUE SON APLICADAS EN SU IMPLANTACION. TALES COMO: <ol style="list-style-type: none"> 1) DISEÑO DE SALIDAS Y DE ENTRADAS. 2) DISEÑO DE PROCESOS Y DE ARCHIVOS. 3) DOCUMENTACION DE LOS DETALLES DEL DISEÑO DEL SISTEMA. 4) SOMETIMIENTO DE LA DOCUMENTACION A LA ALTA ADMINISTRACION.
IMPLEMENTACION	<ul style="list-style-type: none"> - LAS ACTIVIDADES EN ESTA FASE SON LAS SIGUIENTES: <ol style="list-style-type: none"> 1) DESARROLLO DE PROGRAMAS. 2) DESARROLLO DE INSTRUCCIONES DIRIGIDAS AL USUARIO PARA OPERAR EL NUEVO SISTEMA. 3) CAPACITACION AL USUARIO. 4) PRUEBAS A LOS PROGRAMAS. 5) PRUEBAS EN PARALELO. 6) DOCUMENTACION DEL SISTEMA.

FASES	CARACTERISTICAS
M A N T E N I M I E N T O	<ul style="list-style-type: none"> - ES LA MODIFICACION DEL SISTEMA DEBIDO A LOS CAMBIOS QUE SE SUSCITAN EN EL MEDIO A LO LARGO DEL TIEMPO. - LAS ACTIVIDADES DE ESTA FASE SON: <ul style="list-style-type: none"> 1) SOLICITUD PARA EL CAMBIO PARCIAL. 2) ESPECIFICACION DE LAS MODIFICACIONES REQUERIDAS POR LOS USUARIOS. 3) REALIZACION DE LOS CAMBIOS. 4) PRUEBAS A LOS CAMBIOS. 5) PRESENTACION DE LOS RESULTADOS DE LAS PRUEBAS A LOS USUARIOS PARA SU APROBACION. 6) MODIFICACION DE LA DOCUMENTACION DEL SISTEMA. 7) NOTIFICACION AL USUARIO DE LA TERMINACION DEL O LOS CAMBIOS SOLICITADOS AL SISTEMA.

CUADRO I.4

PROCESO DE DESARROLLO DE SISTEMAS SEGUN PHILIP W. METZGUER

FASES	ACTIVIDADES PRINCIPALES	DOCTOS. GENERADOS
D E F I N I C I O N	- DEFINIR EL PROBLEMA TECNICO. - DISCUTIR LAS SOLUCIONES! - EFECTUAR UN ANALISIS PRELIMINAR Y COMENZAR LA PLANEACION LA CUAL CULMINARA EN LA SIG. FASE.	- PLAN DEL PROYECTO. - ESPECIFICACION DEL PROBLEMA.
D I S E Ñ O	- DESCRIBIR UNA SOLUCION ACEPTABLE DEL PROBLEMA LA CUAL ES ESCOGIDA ENTRE VARIAS. - PREPARAR LA PRUEBA DE INTEGRACION. - EFECTUAR UN DISEÑO BASE!	- ESPECIFICACION DEL DISEÑO. - MANUAL DEL PROGRAMADOR. - ESPECIFICACION DE LA PRUEBA PRELIMINAR DE DE ACEPTACION.
P R O G R A M A C I O N	- CONSTRUIR UN PROGRAMA Y SOMETERLO A PRUEBA DE ACUERDO CON EL PROYECTO! - HACER UN DISEÑO DETALLADO. - CODIFICAR, PROBAR Y DOCUMENTAR PROGRAMAS. - EFECTUAR LA PRUEBA DE INTEGRACION. - PREPARAR LA PRUEBA DEL SISTEMA. - PREPARAR LA PRUEBA DE ACEPTACION.	- ESPECIFICACION DE LA PRUEBA DE INTEGRACION. - DOCUMENTACION PRELIMINAR DEL PROGRAMA. - ESPECIFICACIONES DE LA PRUEBA FINAL DE ACEPTACION Y DE LA PRUEBA DE CAMPO.
P R U E B A S	- REALIZAR PRUEBAS AL SISTEMA LO MAS REALES. - ADIESTRAR AL CLIENTE.	

(Continúa en la siguiente página)

FASES	ACTIVIDADES PRINCIPALES	DOCTOS. GENERADOS
A C E P T A C I O N	- MOSTRAR AL CLIENTE EL SISTEMA TERMINADO INCLUYENDO SU DOCUMENTACION A FIN DE OBTENER LA ACEPTACION FORMAL DE QUE EL SISTEMA SATISFACE EL CONTRATO. - SEGUIR ADIESTRANDO AL CLIENTE.	- CONVENIO DE ACEPTACION. - DOCUMENTACION FINAL DEL PROGRAMA.
I N S T A L A C I O N	- INSTALAR EL SISTEMA. - PROBARLO. - PONERLO EN OPERACION EN EL AMBIENTE REAL DEL CLIENTE.	

3.3 EL DESARROLLO DE SISTEMAS VISTO COMO UN PROYECTO.- La Organización de las Naciones Unidas (O.N.U.) define, en su manual de proyectos de desarrollo económico, lo que es un proyecto pero a nivel macro, nos dice: "El proyecto se define como el conjunto de antecedentes que permiten estimar las ventajas y desventajas económicas que se derivan de asignar ciertos recursos de un país para la producción de determinados bienes o servicios" (22).

En esta definición se observa que la palabra "Proyecto" se emplea a un nivel de índole nacional, no obstante, se puede retomar este concepto y definir la evaluación de un pro-

yecto de desarrollo de sistemas como el conjunto de antecedentes que permiten valorar las ventajas y desventajas económicas, organizacionales y de eficiencia que se derivarían de asignar ciertos recursos de la empresa para la creación y desarrollo de un sistema de procesamiento de datos.

Dentro de esta evaluación del proyecto se incluiría la previsión de los impactos que habría en el personal de la organización provocados por el cambio que implica el implantar un nuevo sistema de información computarizado (repercusiones).

También, si se está hablando de proyectos y de su evaluación en el área que nos compete, se debe considerar el riesgo al que se enfrenta el o los encargados de desarrollar el sistema. En el mismo manual citado anteriormente, se habla en los siguientes párrafos acerca de las estimaciones sobre el futuro y sus riesgos: "Las estimaciones sobre el futuro suponen riesgos en cuanto a la certeza de las previsiones. Algunos de los riesgos que enfrenta toda empresa, así como contingencias son asegurable pero no lo son los que derivan de errores de estimación en los varios aspectos que comprende el estudio de un proyecto, y estos pueden ser de tal cuantía que conduzcan al fracaso".

"La asignación de recursos en un proyecto implica hacer frente a lo que se llama RIESGO CALCULADO" (23).

Si se aplican los párrafos anteriores a nuestra materia, se puede apreciar que tales líneas nos muestran la necesidad de efectuar cálculos monetarios de los riesgos que podrían correrse, tanto riesgos normales (fallas técnicas en el equipo, falta de energía eléctrica, etc.) como riesgos derivados de errores de estimación en el estudio (mala utilización de la potencialidad del equipo cuando el sistema ya esté funcionando, no llenar los requerimientos del usuario, baja productividad del sistema y los costos que todo ello implica).

Por otro lado se considera que el estudio correcto de un proyecto de esta índole (proyecto de desarrollo de un sistema de información computarizado) requiere del trabajo armonioso de un equipo de técnicos y profesionistas cuyas aportaciones se complementen. El mismo manual de proyectos de desarrollo económico nos dice al respecto lo siguiente: "...si no se logra la mutua comprensión y entendimiento para llevar en

buen forma el trabajo en equipo, resultarán estériles las mejores técnicas para la elaboración de proyectos... es imprescindible que haya un jefe... que tenga autoridad jerárquica y profesional para tomar las decisiones" (24).

Por otro lado Elias Awad considera que los criterios a tomar en cuenta por la alta administración para la selección de un proyecto (de la índole que hemos venido manejando hasta aquí) son los siguientes:

- a) Criterios Técnicos. - Se refieren a la capacidad del departamento de sistemas para manejar el proyecto.
- b) Criterios Operacionales. - Son un factor crítico en la decisión de llevar o no a cabo el proyecto.
- c) Criterios Económicos. - Son los más importantes en la selección del proyecto. Si la tasa de retorno es solamente marginal no habrá un punto de iniciación ni tampoco un cambio en el sistema existente. Muchas organizaciones consideran que los proyectos de sistemas deben mostrar una promesa de por lo menos el 20% antes de que sean aprobados para su implantación (25).

De esta manera es como la administración debe aprobar los proyectos de estudio de sistemas, basándose en los criterios anteriores.

3.4 EL EMPLEO DE MODELOS EN EL DESARROLLO DE SISTEMAS Y SU SIMILITUD CON LA METODOLOGIA INGENIERIL.- En la teoría general de sistemas es útil el empleo de modelos para simular sistemas ya que facilitan la visualización y comprensión de los mismos. Sabemos así mismo que un modelo es la representación de la realidad o de un sistema planeado, por ejemplo un sistema de compras, el cual es desarrollado de acuerdo al proceso genérico de desarrollo de sistemas de una manera planeada, es decir, antes de su implantación y operación debe ser diseñado por medio de un modelo. Por tanto, es importante citar el concepto de "Modelo" dentro del área de la informática. Elias Awad nos dice que un analista de sistemas sin un modelo es semejante a un mecánico sin su llave inglesa. Con esta comparación el autor recalca la importancia de los modelos como una herramienta de base para poner en práctica un sistema de cómputo y para la solución de proble-

(24) O.N.U. OP.CIT. PAGS. 4 Y 12.

(25) AWAD. OP.CIT. PAGS. 39-40.

mas reales por medio de su entendimiento y representación. El modelo permite al analista de sistemas partir de algo ya planeado para seguir el camino con mayor seguridad sin improvisaciones durante el desarrollo del sistema.

El analista puede emplear los siguientes tipos de modelos:

- a) Modelos Esquemáticos. - Son planos en los cuales se describen los elementos del sistema y sus ligas.
- b) Modelos de Flujo de Sistemas. - Estos modelos muestran el flujo de materia, energía o información de los vínculos que existen entre los elementos del sistema y ayudan a explicar su comportamiento (diagramas de flujo por ejemplo).

Por otro lado, se considera que un sistema de información computarizado (Sistema de Procesamiento de Datos) es un modelo de sistema de tipo dinámico, ya que está siempre en constante cambio, es autorregulable, autodirigible y su conducta tiene un objetivo.

Elias Awad menciona que los modelos de sistemas dinámicos consisten de los siguientes elementos:

- a) Entradas del Sistema.
- b) Procesador del Sistema.
- c) Programas Requeridos para el Procesamiento.
- d) Salidas.
- e) Retroalimentación. (26)

En cuanto a la similitud que tiene la metodología ingenieril con el proceso de desarrollo de sistemas se menciona lo siguiente:

Un ingeniero es un profesional técnico que traslada las necesidades del usuario en soluciones tangibles. Tanto el trabajo del ingeniero como del analista de sistemas son paralelos, ya que el analista efectúa la misma función pero en el área de sistemas.

A continuación se enumeran las características de la metodología ingenieril las cuales pueden irse comparando simultáneamente con las del analista de sistemas:

a) Comienza primero con una imagen lógica del proyecto, con un diseño tosco. En esta fase no se toman muy en cuenta los detalles.

b) Construye los modelos y los dibuja cuando el diseño lógico ha sido aceptado. Los modelos pasarán a ser en esta fase un prototipo en miniatura del proyecto de la construcción. Aquí el modelo funge como un medio de comunicación excelente ya que a través de éste el ingeniero puede comunicar a los que no lo son su diseño físico.

c) Después se construirán los modelos gráficos de la construcción a escala. De esta manera los usuarios podrán tener una visión sobre el conjunto final y el contratista podrá leerlo y crear la construcción. Se desea agregar que el ingeniero que hace el trabajo anterior se puede equiparar con el analista de sistemas que es el que hace los modelos del sistema, y así mismo, el contratista puede igualarse al programador que es el que debe entender lo que desea realizarse.

d) Finalmente el proyecto es construido. El contratista será la primera persona que se referirá a los documentos gráficos para construir cuidadosamente la estructura física en base a las especificaciones lógicas (27).

Si analizamos detenidamente lo anterior puede concluirse que el desarrollo de sistemas de procesamiento de datos, para lograr éxito debe estar sujeto también a una metodología similar.

(27) WILIAM S. DAVIS. SYSTEM ANALYSIS AND DESIGN : A Structured Approach. PAGES. 6-7.

4. PROBLEMATICA DEL DESARROLLO DE SISTEMAS EN MEXICO

4.1 CONTEXTO GENERAL.- En la actualidad nos enfrentamos a un proceso de desarrollo tecnológico en el área de computación, en donde los países de vanguardia tecnológica viven una época que se caracteriza por un proceso continuo de cambio y de competencia acelerada para mantenerse a la cabeza del desarrollo de nuevos y más sofisticados equipos de computación.

Para los países en proceso de desarrollo, como es el caso de México, día a día se hace más grande la brecha y la dependencia tecnológica en el campo del hardware pues es cada vez más difícil disponer de los recursos materiales y humanos para poder desarrollar una tecnología que pudiera competir en éste mercado. Al respecto el gobierno mexicano, desde hace varios años, implantó una política de integración nacional de la industria de la computación, a fin de lograr el ensamblaje de minicomputadoras y microcomputadoras con una parte de piezas elaboradas en el país, con vista a que éste número de piezas se incremente progresivamente creando así nuevas industrias y fomentando la investigación en este campo.

Por el momento la tecnología utilizada para la construcción de estos aparatos sigue siendo de los países más avanzados, por lo que nuestro país aún cuando exporte parte de los equipos que ensambla se ve sumamente limitado en este campo a nivel mundial, pero por otro lado, se ha fortalecido el mercado nacional, difundiéndose el uso de las computadoras entre gran parte de las empresas del país, teniéndose así que en 1984 el número de microcomputadoras instaladas en México fue de 18500 unidades con un costo de 19,545 millones de pesos, y se pronostica para los próximos cuatro años las siguientes cifras: (28)

ANO	UNIDADES
1985	29,600
1986	42,400
1987	57,800
1988	72,500

(28) M EN C MANUEL ALVAREZ. PRIMERA CONFERENCIA INTERNACIONAL : "LAS COMPUTADORAS EN INSTITUCIONES DE EDUCACION SUPERIOR". PUC-BURROUGHS. 1985.

Del total de equipos instalados, su aplicación se divide de acuerdo a la siguiente tabla: (29)

CUADRO I.5

APLICACION NACIONAL DE EQUIPOS DE COMPUTO

%	COSTO (MILL DE \$)	APLICACION	NUMERO DE EQUIPOS
70.6	13,789	EMPRESAS Y ACT. PROF.	11,378
27.4	5,365	EDUCACION	6,771
2.0	391	INVESTIGACION	351
100.0	19,545	TOTAL:	18,500

Si tomamos en cuenta que la tabla anterior sólo muestra la distribución de las microcomputadoras en el año de 1984, y a ésto le aumentamos que históricamente, como es bien sabido, el mayor número de computadoras están instaladas en empresas, claramente se podrá comprender su importancia, cada vez más alta, para el buen desarrollo de las actividades empresariales al realizar a través de éstas el manejo de la información básica de sus operaciones. Es por esto que las principales aplicaciones de las computadoras en México para el tratamiento de información administrativa, financiera y de actividades profesionales, deben ser logradas con sistemas específicos desarrollados en su mayor parte por técnicos mexicanos.

Es en el área de Desarrollo de Sistemas en donde México puede llegar a tener un lugar destacado, pues se cuenta con los equipos y con los profesionistas idóneos; si se promoviera más esta área, encausándola adecuadamente para el buen aprovechamiento de los recursos disponibles, muy bien se podría llegar a satisfacer nuestra demanda interna y competir en calidad y costos a nivel internacional. Pero para poder lograr lo anterior hay que salvar primero obstáculos que ori-

ginan la ineficiencia en el desarrollo de sistemas.

A continuación se mencionarán brevemente algunos de los factores que obstaculizan el desarrollo de sistemas.

4.2 DISPONIBILIDAD DE RECURSOS HUMANOS.- Primeramente se puede mencionar la falta de personal capacitado en esta área, posiblemente el freno más poderoso para la computación. El déficit de personal preparado era en el año de 1983 de casi 15000 técnicos y se preveía que para 1986 esta cantidad se cuadruplicara.

En una etapa inicial la capacitación del personal técnico corría a cargo de las compañías manufactureras del equipo y de academias comerciales que proliferaron en este tiempo, posteriormente, y en la medida en que las instituciones de educación superior empezaron a incorporar materias aisladas en esta área y después planes de estudio en computación, compartieron de esta manera, en forma creciente la formación de recursos humanos en este campo. Los primeros planes de estudio se inician a mediados de los años setentas en instituciones como el Instituto Tecnológico de Monterrey, el IPN y posteriormente en la UNAM. Para el año de 1983 se tenían definidos 114 planes de estudio en computación (30).

De todas maneras la producción de estos profesionales, viene a ser una primera solución a los requerimientos del país, y con el tiempo tendrán que ir ocupando la posición que les corresponde en la empresa, pero como el problema subsiste y se agranda, es necesario resolverlo a fondo, y la solución debe enfocarse a los niveles inferiores al profesional, y aún cuando esto ya está abordándose con la creación de carreras cortas como las que creó el CONALEP recientemente y otros organismos del estado, los resultados aún tardarán en apreciarse y como opción inmediata está el capacitar al personal que labora en cada empresa. (31)

Lo anterior provoca que un profesional en esta área esté altamente cotizado en el mercado de trabajo y no es difícil encontrar que el personal de programación y operación de computadoras se le pague más que a otros técnicos de más o menos el mismo nivel.

(30) M. EN C. MANUEL ALVAREZ. OP.CIT. PAG. 26.

(31) LA CAPACITACION COMO UN RECURSO DE DIRECCION EN LOS CENTROS INFORMATICOS. CIFCA. ENERO DE 1983. PAG. 7

4.3 PLANEACION DEL CAMBIO.- Por otro lado, gran parte de los computadores que se instalan en empresas gubernamentales o privadas son adquiridos en forma compulsiva, es decir, sin estar considerados dentro de un plan orientado a una evolución de la empresa sino más bien orientados por presiones momentáneas que hacen ver a los directivos el espejismo de que la computadora les resolverá todos sus problemas sin tomar en consideración que el lograr que la información sea procesada por la máquina toma tiempo y requiere del esfuerzo y participación de muchas personas dentro de la organización, y no sólo de aquellas que estén a cargo de la operación y programación del computador, sino también de las áreas usuarias, además de que se tienen que tomar las medidas administrativas necesarias para permitir el desarrollo de sistemas y procedimientos y así evitar las interferencias que pudieran provocar la subutilización del computador.

4.4 PROYECCION DE LOS COSTOS DE LA AUTOMATIZACION.- La falta de planeación por parte de los directivos provoca inestabilidad en el desarrollo de las actividades de informática, partiendo de que en muchas ocasiones realmente no se presupuesta el costo total que implica el implantar un sistema, ya que sólo se piensa en los costos de la compra del equipo de cómputo dejando afuera aquellos que implican la adecuación del lugar donde estará el equipo e instalaciones especiales si fuera necesario; la capacitación del personal que tendrá contacto con el equipo; el material como papel continuo, cintas, discos, etc.; el contratar los servicios de personal especializado, que como se mencionaba anteriormente gana sueldos relativamente altos y que en ocasiones los funcionarios se resisten fuertemente a pagarlos; y finalmente la compra de paquetes de aplicación, o la contratación de los servicios de una compañía de profesionales del área para el desarrollo de los sistemas específicos, o en su defecto el tener un equipo de técnicos propio para el desarrollo de los mismos y el costo de mantenimiento tanto de los equipos como de los sistemas.

4.5 EFECTOS PSICOLOGICOS EN LA ORGANIZACION.- En muchas ocasiones cuando se introduce por primera vez un computador o cuando se automatizan funciones de alguna área, los analistas, al entrevistar a las personas cuyos trabajos están relacionados con las funciones que el computador reemplazará o complementará, se topan con una serie de problemas tales como:

- a) Frustración y aumento de errores en el personal de la oficina, creados por el temor de que serán remplazados o despedidos por la intervención del computador.
- b) Poca cooperación del personal hacia los analistas, a quienes entregan información falsa o incompleta.
- c) Desconfianza al cambio por parte del personal, ya que está acostumbrado a trabajar de una forma determinada.

Los problemas anteriores son en gran parte producto de una comunicación ineficiente entre el nivel ejecutivo y el nivel operativo. Para evitar dichos problemas es necesario comunicar explícitamente al personal cual es el objetivo de tal cambio, de que manera el computador afectará su trabajo, si será necesaria una readecuación de puestos, posibilidades de capacitación, etc. Para lograr la cooperación del nivel operativo es necesario explicarles de que manera el computador facilitará y agilizará sus funciones, además de enseñarles como podrían hacer un buen uso de éste.

Es función de los ejecutivos el crear un ambiente de cambio que facilite el trabajo de los analistas y prepare al personal para dichos cambios.

4.6 UBICACION DEL DEPARTAMENTO DE SISTEMAS DENTRO DE LA ORGANIZACION.- Antiguamente era normal ver que el computador o departamento de sistemas estuviese subordinado al área de finanzas y/o contabilidad porque era ella quien más utilizaba sus servicios. Poco a poco se ha ido diversificando la aplicación de la computadora para el tratamiento de la información de otras áreas funcionales, como es el caso actual de la mayoría de las empresas, por lo tanto no es conveniente que esté subordinado a ninguna de ellas, sino que exista como departamento o gerencia autónoma. Ahora bien, al implantar un sistema de información se hace necesario el modificar los procedimientos existentes y los flujos de información del área afectada, por lo que tal departamento deberá contar con la autoridad suficiente que le permita producir los cambios necesarios en las materias que le competen, y la manera de imponer ciertas normas y procedimientos que faciliten el flujo de información para su posterior procesamiento.

4.7 LA EXPECTANCIA DE RESULTADOS INMEDIATOS POR PARTE DE LOS FUNCIONARIOS.- A este respecto se ha escrito muy poco, pero es uno de los males que más aquejan al desarrollo de un sistema dentro de una organización. Por lo regular en las empresas todo trabajo debe de estar concluido para "ayer", esto traspasado al área de informática es ocasionado por el desconocimiento y falta de planeación por parte de los funcionarios de las áreas usuarias y del abundante trabajo que debe de realizarse para el desarrollo de un sistema. Lo anterior provoca que los trabajos deban realizarse al "vapor", perdiendo de vista la importancia que tiene el hacer un análisis y diseño a conciencia de los requerimientos del usuario, trayendo como consecuencia la producción de sistemas ineficientes e inadecuados que por el momento sacarán del apuro al usuario pero a largo plazo resultarán mas costosos por el mantenimiento, corrección de errores y modificaciones que se les tenga que hacer, además de que en su gran mayoría al momento de adecuarlos a nuevas necesidades serán inservibles y tendrán que ser programados nuevamente.

Si por el contrario los funcionarios estuviesen conscientes la importancia de cada una de las etapas del desarrollo de un sistema, podrían apoyar a estas actividades formando grupos temporales constituidos por personas que de alguna forma tendrán que ver con el sistema, coordinado por un líder para que de esta forma se pueda llevar a cabo un estudio detallado de las necesidades actuales y futuras, logrando así el diseño de un sistema capaz de soportar el advenimiento de nuevas necesidades sin grandes cambios.

4.8 DEFICIENCIAS PROFESIONALES.- Al principio del tema No. 4 se vió la problemática existente por falta de personal capacitado que se dedica al desarrollo de sistemas, a lo cual podemos aunar los siguientes puntos:

a) Inexperiencia del Personal que se Dedicar al Desarrollo de Sistemas.- En muchos casos las personas que están desarrollando sistemas carecen de experiencia suficiente porque o acaban de terminar sus estudios o bien porque se trata de personal sin formación profesional el cual es empleado rápidamente o movido de otros puestos de la empresa ya que ésta solicita la resolución de necesidades inmediatas.

b) Falta de Capacitación.- Muchas veces es personal de

otras áreas que sólo tomaron cursos de programación para dedicarse al desarrollo de sistemas.

c) Falta de Profesionalismo. - En ocasiones se encuentran profesionales capacitados que no hacen uso de los conocimientos que tienen de análisis y diseño y se avocan a desarrollar sistemas sin un mínimo de planeación y control de sus actividades.

Estos tres puntos anteriores son los males que más aquejan a nuestro medio. Son las mismas personas del área de computación los que o no conocen o pierden de vista la importancia de manejar una herramienta que les permita realizar un buen análisis y diseño en el cual basarse para la programación e implantación de los sistemas que desarrollan.

5. NECESIDAD DE EMPLEAR UNA TECNICA EN EL DESARROLLO DE SISTEMAS

Aquella persona que presume de ser un profesional del desarrollo de sistemas no debe de suponer que un problema es obvio y que todo mundo sabe en que consiste, así como tampoco comenzar a diseñar los programas inmediatamente después de que se le ha presentado un problema, pues si así lo hiciese solo generará una serie de programas que en forma incipiente resolverán los problemas momentáneos del usuario. Es tiempo de que las personas que se dedican al desarrollo de sistemas guíen sus acciones y conocimientos a través de una técnica que les permita crear un trabajo de calidad que resuelva de la mejor forma las necesidades de los usuarios para que en verdad se les pueda llamar "profesionales del desarrollo de sistemas".

Existen muchas técnicas que podemos utilizar, todas formuladas en base al ciclo de vida de un sistema pero con un punto de vista diferente que en algunos casos ayudarán a resolver de la mejor forma determinadas necesidades. Tales técnicas se verán en los siguientes capítulos.

C A P I T U L O I I
LA TECNICA ESTRUCTURADA

CAPITULO II

LA TECNICA ESTRUCTURADA

1. FUNDAMENTOS

No hace mucho que la industria de la computación ha sido invadida por nuevas técnicas y filosofías de como aprovechar de la mejor forma el uso del computador. El desarrollo de sistemas es relativamente una nueva profesión, su metodología envuelve muchas versiones de como efectuarlo de manera correcta. Dentro del ámbito profesional que aquí tratamos frecuentemente se escuchan términos tales como: "Programación Estructurada", "Diseño Estructurado", etc. Se considera que tales términos son instrumentos o técnicas que se usan de manera aislada sin tomar en cuenta más que la fase de desarrollo del sistema en la que se emplean: Diseño y Programación.

El objetivo de este capítulo es exponer un criterio más amplio y general al respecto. Se considera necesario conjuntar lo que podría llamarse el "Ciclo de Vida Estructurado" que no es más que el desarrollo de sistemas de una manera estructurada, es decir, de una manera paulatina, organizada y ordenada de tal forma que los resultados que se obtengan de un trabajo de esta naturaleza sean satisfactorios.

2. DESCRIPCION DE LA TECNICA

William S. Davis describe la Técnica Estructurada como una aproximación paulatina hacia el desarrollo de sistemas, comenzando con un diseño lógico el cual irá gradualmente trasladándose hacia el diseño físico. Cuando se usa la técnica estructurada, el analista de sistemas debe avanzar paso a paso, de una manera cuidadosa y metódica, hasta alcanzar los criterios definidos para cumplir con cada fase del ciclo y así lograr éxito en los resultados.

Los criterios definidos en cada fase no son más que indicadores que señalan cuando una fase ha sido terminada satisfactoriamente y puede comenzarse con la siguiente. Muchas veces lo anterior no se da en la realidad ya que por ejemplo, que pasaría si es descubierto un error en la etapa de defini-

ción del problema cuando ya estamos en la fase de análisis? o que dificultades habría en la implantación si se decidiera modificar el diseño?, lo que se haría en estos casos sería retroceder a la etapa precedente y volver a comenzarla. Para evitar los problemas anteriores están los "Criterios de Exito" definidos en cada una de las etapas del ciclo de vida estructurado, de los cuales se hablará posteriormente. Lo importante de todo esto es no olvidar que la técnica debe llevarse paso a paso, sin improvisaciones.

Algunos indicadores que podrían caracterizar a la técnica son los siguientes:

- a) Es llevada paso a paso de acuerdo al cumplimiento de ciertos criterios ya definidos en cada fase.
- b) La aproximación paulatina desde un modelo lógico hasta un modelo físico dan a la administración un claro control del proyecto.
- c) Usa una gran variedad de nuevas técnicas tales como el diseño estructurado, programación estructurada, diagramas de flujo de datos, cartas estructuradas para planear la calidad de los sistemas que serán implantados y diseñados (1).
- d) Cada etapa finaliza con una inspección (monitoreo del proceso), la cual asegura que el trabajo técnico sea bueno.
- e) Al final de cada etapa se generan ciertos documentos que sirven de base para iniciar la etapa subsiguiente así como un medio de comunicación e información tanto para los usuarios directos como para la administración.

3. CICLO DE VIDA

Anteriormente se aclaró que el ciclo de vida, al desarrollar un sistema de la naturaleza que aquí tratamos, es en su esencia el mismo. No obstante, la técnica estructurada propone, basándose en el ciclo de vida tradicional, un nuevo ciclo el cual podría denominarse Ciclo de Vida Estructurado. Este ciclo se caracteriza porque en cada una de sus fases

(1) COTTERMAN, ET.AL. OP.CIT. PAG. 1.

propone actividades bien determinadas y específicas que se verán más adelante con detenimiento.

Norman L. Enger enumera 7 fases del ciclo de vida estructurado, las cuales son:

- a) Análisis de Requerimientos.
- b) Diseño lógico.
- c) Diseño Físico (Diseño Estructurado).
- d) Implantación Top-Down.
- e) Aceptación de Pruebas.
- f) Calidad de Seguridad.
- g) Operación del Sistema (2).

Por otro lado, William S. Davis cita las siguientes fases como integrantes del ciclo de vida de un sistema con una aproximación estructurada:

- a) Definición del Problema..
- b) Estudio de Factibilidad.
- c) Análisis del Sistema.
- d) Diseño del Sistema.
- e) Diseño Detallado.
- f) Implantación.
- g) Mantenimiento. (3)

Si se observan los dos ciclos propuestos anteriormente, se puede afirmar que tienen gran similitud entre sí y además también se asemejan a lo que es el ciclo tradicional o clásico de vida de un sistema. Lo anterior no tiene nada de nuevo pues el ciclo de vida de un sistema en el área de estudio que tratamos es, por lo menos, familiar para las personas que se mueven en el campo de los sistemas, lo importante es que la técnica estructurada toma dicho ciclo y lo trata de una manera diferente, lo transforma y hasta se podría decir que lo estiliza dándole un nuevo enfoque: El Enfoque Estructurado.

Para desarrollar y explicar la técnica estructurada nos basaremos en el ciclo de vida propuesto por William S. Davis; la razón de ello fue que nos pareció bastante detallada, completa y además fue la que estuvo más a nuestro alcance, sin embargo, se desea hacer notar que varios autores proponen el desarrollo de sistemas estructurado de manera

(2) COTTERMAN, ET.AL. OP.CIT. PAG. 3-5.

(3) S. DAVIS WILLIAM. SYSTEMS ANALYSIS AND DESIGN: A Structured Approach. PAG. 8.

diferente siendo que en su esencia es siempre el mismo.

Antes de comenzar a ver la técnica estructurada, se desea agregar que existe una similitud entre el proceso de dicha técnica con el proceso que se sigue en la investigación social. Este hecho se da especialmente en las tres primeras etapas del ciclo de vida estructurado: Definición del Problema, Estudio de Factibilidad y Análisis del Sistema. Dichas etapas tienen por objetivo el conocer una realidad a través de la investigación del medio con la finalidad de desarrollar un sistema de información acorde a las necesidades de la organización.

En función de la observación anterior se considera importante exponer algunos preceptos tomados de la metodología científica para que el analista o el encargado del desarrollo de un sistema los tome en cuenta con la finalidad de realizar su investigación preliminar de una manera más exacta, veraz y objetiva, y que otra cosa mejor para lograr esto que el acercamiento a la metodología de la investigación científica?.

A continuación se exponen algunas ideas básicas de lo que es la investigación científica, las cuales pueden aplicarse al proceso estructurado de desarrollo de sistemas especialmente en sus tres primeras etapas:

a) "El progreso del hombre se finca en la búsqueda de respuestas a los fenómenos del medio. En el momento en el que el hombre encuentra esas respuestas podrá explotar racionalmente el medio ambiente a través de una adecuada organización social del trabajo". Si transportamos este razonamiento al ámbito del desarrollo de sistemas estructurado podemos concluir que el analista junto con su equipo de trabajo deben encontrar, durante la realización de las tres primeras fases del ciclo estructurado, las respuestas a los problemas existentes (fenómenos del medio) referentes a los flujos y procesos de información para así lograr desarrollar un sistema adecuado a las necesidades de la organización, esto a la larga generará mayor productividad y coadyuvará al progreso de la organización en cuanto al mejor alcance de sus objetivos.

b) "Las experiencias de los individuos debidamente sistematizadas juegan un papel importante en el desarrollo de conocimientos teóricos". Lo anterior significa que si el ana-

lista aprovecha la experiencia adquirida en el pasado y la aplica en las situaciones futuras esta experiencia podrá convertirse en un conocimiento sistematizado y teórico.

c) "La investigación científica pretende encontrar respuesta a los problemas trascendentales que el hombre se plantea lograr, pero, para que estos hallazgos sean consistentes y significativos, deben obtenerse mediante un proceso que implica la concatenación lógica de una serie de etapas". También, el desarrollo de sistemas estructurado se basa en una serie de etapas que están relacionadas entre sí, con la finalidad de cumplir con un objetivo: crear un sistema de información acorde a las necesidades de la organización.

d) "La investigación científica, en cualquier área del conocimiento humano se plantea como finalidad la descripción, explicación y predicción de los fenómenos". El ciclo de vida estructurado pretende por su parte, en sus tres primeras etapas, la descripción, explicación y comprensión de un sistema de información existente para encontrar sus ventajas, desventajas y problemas (apropiación de la realidad) con lo cual se logrará el diseño de un sistema de información más eficiente y productivo.

e) "La investigación es un proceso que se inicia con el planteamiento de un problema que requiere solución, y para encontrarla el experto social debe construir un diseño de investigación que le permita lograr descubrir, explicar y si es posible, predecir probabilísticamente determinadas situaciones y efectos". El proceso estructurado de desarrollo de sistemas también se inicia con la definición del problema. El líder o analista encargado del desarrollo del sistema de información debe idear la manera en la que será efectuada la "investigación" para conocer la situación de la organización en cuanto a sistemas de información se refiere. Para realizar tal investigación el analista puede valerse de algunas técnicas como son las entrevistas, cuestionarios, observación entre otras, las cuales son bastante útiles para diseñar una investigación directa.

Los aspectos a tomar en cuenta en toda investigación son los siguientes:

(Ver en la siguiente página)

- Magnitud.- Se refiere al tamaño del problema y a los individuos afectados por éste.
- Trascendencia.- Es la ponderación que la sociedad, grupo u organización hacen del problema de acuerdo a su gravedad y consecuencias.
- Vulnerabilidad.- Consiste en el grado en el cual un problema puede ser resuelto o atacado.
- Factibilidad.- Es la existencia de recursos y organización suficientes para solucionar o disminuir el problema.

Si analizamos los aspectos anteriores, nos damos cuenta que al realizar un proyecto de desarrollo de un sistema de información debemos tomar en cuenta también la magnitud o alcance del proyecto (áreas y personal que envolverá el posible cambio), la importancia o trascendencia del proyecto, su dificultad o vulnerabilidad así como su factibilidad.

g) Existen dos tipos de investigación, la directa y la documental. En la investigación directa la información para el análisis de la situación se obtiene de la realidad a través de técnicas como la observación, la entrevista estructurada, la encuesta y otras. En cambio, en la investigación documental se recurre a las fuentes escritas referentes al tema para efectuar el análisis del problema. El analista de sistemas debe valerse de estos dos tipos de investigación para lograr un mejor conocimiento de la situación. En el caso de la investigación documental puede solicitar manuales de sistemas y procedimientos, de flujos de información, reportes diversos ,etc. de tal manera que el estudio de tales documentos sirvan como complemento al análisis del problema.

Los tipos de investigación directa tal y como lo expresa Raul Rojas Soriano son los siguientes:

- Estudios Exploratorios o de Acercamiento a la Realidad Social.- Su propósito es recabar información para reconocer, ubicar y definir problemas; fundamentar hipótesis, recoger ideas o sugerencias que permitan afinar la metodología, depurar estrategias, etc. para formular con mayor exactitud el esquema de investigación definitivo.

- Estudios Descriptivos.- Su objetivo es obtener un panorama más preciso de la magnitud del problema o situación, jerarquizar los problemas, derivar elementos de juicio para estructurar políticas o estrategias operativas y conocer las variables que se asocian.

Estos dos tipos de estudios de investigación directa pueden ser tomados en cuenta por el analista en las tres primeras etapas del ciclo de vida estructurado con la finalidad de comprender perfectamente la situación que le compete estudiar.

Finalmente, para evitar análisis superficiales, fragmentarios y aislados, pueden tomarse en cuenta algunos puntos centrales que son importantes en cualquier investigación tal y como la que se efectúa al inicio de un proyecto de desarrollo de sistemas de información. Estos puntos son los siguientes:

- Todos los fenómenos de una realidad se encuentran en relación y dependencia mutua. Esta tesis (causa-efecto) nos ayuda a conocer las causas del los fenómenos.
- La realidad se encuentra en permanente movimiento, cambio y transformación. Los hechos, fenómenos o situaciones no son estáticos.
- La realidad se presenta a diferentes niveles. Nosotros sólo captamos los aspectos externos de los objetos y procesos. Los elementos y relaciones internas, la esencia, es posible conocerlos recurriendo a la abstracción (uso de modelos).
- No puede haber conocimiento sin práctica (4).

En las siguientes páginas se presentarán cada una de las etapas que conforman el ciclo de vida estructurado propuesto por William S. Davis.

(4) LAS IDEAS CENTRALES DE LOS INCISOS FUERON TOMADAS DEL LIBRO: ROJAS SORIANO RAUL. GUIA PARA REALIZAR INVESTIGACIONES SOCIALES. PAGES. 27-38.

3.1 DEFINICION DEL PROBLEMA.- Definir un problema es una actividad muy común que debe darse en la mayoría de las investigaciones, ya sean éstas de tipo social, administrativo, económico, mercadológico, etc. para así poder saber hacia donde se dirigirán los esfuerzos y todo el trabajo humano al resolver la cuestión definida.

El plantear un problema significa reducirlo a sus aspectos y relaciones fundamentales a fin de poder iniciar su estudio intensivo. Para definir un problema es necesario hacer una aprehensión de la realidad por medio de la abstracción y después plantearlo en términos concretos destacando aquellos elementos y vínculos importantes para una primera aproximación.

En la práctica la formulación del problema se inicia planteando en términos generales la cuestión que se pretende estudiar. A medida que se avanza en la indagación o investigación de la situación (lo cual ocurre en las 3 primeras etapas del ciclo de vida estructurado), el problema se empieza a plantear en forma más clara y precisa ya que se conocen cada vez más detalles.

Rojas Soriano nos señala que en el momento de plantear, aunque sea de manera rudimentaria un problema, el investigador adelanta una hipótesis preliminar llamada "Hipótesis de Trabajo", la cual tiene que ser superada conforme se profundiza en el planteamiento del problema (5). Lo mismo puede hacer el analista después de haber realizado una investigación preliminar y de haber tenido una plática informal con los representantes de la organización. Rojas Soriano continúa diciendo al respecto lo siguiente: "Puede suceder que una hipótesis de trabajo conduzca a nuevas interrogantes que permitan el planteamiento de otras hipótesis. El proceso de conocimiento se muestra como una relación de preguntas (problemas) y respuestas posibles (hipótesis)...esta complejidad del proceso de investigación puede llevar al investigador por sendas equivocadas que lo alejen del problema inicial, pero también puede conducir al descubrimiento de otras facetas que resulten básicas para una comprensión más amplia del problema" (6).

La definición del problema es un prerequisite para comenzar a realizar un análisis del sistema. La fuente princi-

(5) ROJAS. OP.CIT. PAG. 47.

(6) ROJAS. OP.CIT. PAGS.47-48.

pal de información será tanto el usuario como la administración, ambos darán la pauta para definir las necesidades de información y el problema junto con sus causas. Las dos partes ayudarán también a identificar el área afectada. Todo lo anterior se logrará conocer por medio de una plática informal. Si el problema es juzgado significativo, la administración y el usuario pedirán al analista y a su equipo de trabajo (ya sea éste contratado o que forme parte de la organización) que lo estudie. En éste momento surgirá el proyecto para desarrollar un nuevo sistema o modificar el ya existente.

Después de las pláticas informales, y una vez realizada una hipótesis de trabajo, el analista puede diseñar algunos cuestionarios, entrevistas y/o rutinas de observación dirigidas al personal envuelto en la situación para indagar más sobre ésta de una manera más formal.

El cuestionario puede contener los siguientes puntos:

- a) Planteamiento del Problema.
- b) Información que sustenta el Problema.
- c) Su Urgencia.
- d) Nombre del Entrevistado.
- e) Departamento o División.
- f) Fecha y Teléfono interno.

Si el usuario contesta ampliamente a los puntos anteriores se podrá determinar con tal información la naturaleza del problema.

Una vez realizadas las entrevistas formales y aplicados los cuestionarios, el analista estudiará la información recopilada y elaborará un documento por escrito que contendrá una descripción y opinión acerca del problema, así como un análisis de las necesidades del Usuario. Después revisará este documento con la administración y el usuario para discutir si realmente se ha encontrado la verdadera situación. En caso de que el analista haya comprendido realmente el problema y lo que el cliente desea (admon. y usuarios) procederá a plantear el alcance y objetivos del proyecto, los cuales quedarán especificados en un documento final (Ver CUADRO II.2).

En este momento es importante anotar que en el transcurso de las etapas subsiguientes del ciclo de vida estructurado

llegan a surgir problemas de diversa índole: modificaciones al diseño, retrasos, etc. cuyo origen es un planteamiento confuso e incorrecto. Por ello debe tenerse especial cuidado en las primeras etapas de la investigación, pues de lo contrario se corre el riesgo de nunca avanzar en la aprehensión de la realidad o hacerlo en forma incorrecta. El incurrir en el error anterior provocará el empleo de técnicas y diseño de instrumentos (cuestionarios por ejemplo) poco útiles para captar la información requerida; por consecuencia se obtendrán resultados y conclusiones impregnados de errores cometidos en esta fase, lo que impedirá que los resultados sean útiles.

Un planteamiento inadecuado del problema tendrá repercusiones en las distintas etapas de la investigación a tal grado que puedan entorpecer el desarrollo general de la misma y, por consiguiente retrasar la fecha de su terminación.

Por lo tanto, se considera pertinente exponer aquí algunos criterios básicos a tomar en cuenta cuando se plantee un problema. Tales criterios son los siguientes:

- a) Precisar las características del problema y conceptualizarlo, así como también dilucidar posibles conexiones entre distintos aspectos que están presentes en en la problemática.
- b) Establecer los límites espaciales del estudio (área de la organización que va a estudiarse).
- c) Definir las unidades de observación.
- d) Situar el problema en el contexto ambiental.
- e) Mantener una constante retroinformación entre responsables del estudio y solicitantes del mismo (7).

Las actividades enumeradas anteriormente son propuestas por Rojas Soriano para la realización de investigaciones de tipo social, no obstante éstas mismas pueden aplicarse cuando el analista se encuentre en la etapa de indagación del problema ya que esta actividad se asemeja mucho a una investigación de tipo social (a nivel micro).

(7) ROJAS. OP.CIT. PAG. 50.

El éxito de todo proyecto de investigación quedará en gran medida supeditado a la correcta definición del problema. Rusell L. Ackoff señala: Un problema bien planteado es la mitad de la solución (8). Así también, Eli de Gortari menciona al respecto lo siguiente: "En el planteamiento correcto del problema descansa la posibilidad de su solución" (9).

Algunos criterios para el planteamiento de buenos problemas de investigación son los siguientes:

- a) El problema deberá expresar una relación entre dos o más variables.
- b) El problema debe formularse claramente y sin ambigüedades (10).

Kelinger hace énfasis en que la mejor forma de plantear un problema, de la manera más simple es haciendo una pregunta acerca de la situación que vamos a estudiar.

Finalmente deberá hacerse una sencilla estimación de los fondos que se requerirán para financiar el proyecto, es decir, las estimaciones del costo de éste (análisis del costo beneficio). También deberán calcularse el costo y los materiales a emplear en la próxima etapa: Estudio de Factibilidad. Estas estimaciones darán una idea a la administración sobre el alcance del proyecto lo cual le permitirá decidir si se continúa o no con éste o bien si se le efectúan ciertas modificaciones dependiendo de su urgencia y de los fondos con los que se cuenten. Para efectuar lo anterior es muy importante considerar la disponibilidad de recursos financieros, humanos y materiales, los cuales determinarán en última instancia los alcances de la investigación.

Es muy importante aclarar que la definición del problema debe ser breve y que el único propósito de esta etapa (criterio de éxito) es definir los objetivos del sistema y el alcance del proyecto.

Los objetivos del sistema son los puntos de referencia o señalamientos que guían el desarrollo del proyecto y a cuyo logro se dirigen todos los esfuerzos. Cuando los objetivos sean fijados, éstos deben estar bien fundamentados y ser susceptibles de alcanzarse.

(8) ROJAS. OP.CIT. PAG. 52.

(9) ROJAS. OP.CIT. PAG. 52.

(10) ROJAS. OP.CIT. PAG. 53.

En el momento de fijar los objetivos deben tenerse presentes diversos aspectos o circunstancias que pueden limitar o impedir la realización de los mismos, por ejemplo: carencia de recursos humanos o financieros.

Por otro lado, al definir el alcance del proyecto, el analista, el usuario y la administración se encuentran normalmente con tres limitaciones:

- a) El Tiempo.
- b) El Financiamiento.
- c) La Materia de Estudio.

Estas mismas limitaciones van a definir el alcance del proyecto, es decir, el alcance puede estar dado en términos monetarios (financiamiento), en términos del tiempo que se tiene para terminar el proyecto o en términos de la cantidad de trabajo que se requiera realizar. Normalmente una de las variables anteriores (variable independiente) determinará a las demás. Por ejemplo, el tiempo puede ser una variable importante que podría determinar el alcance del proyecto y a la cual se supeditarán las otras dos (cuando se requiere que el proyecto se termine en X fecha) o bien puede ser variable independiente el tamaño del área de estudio (si será por ejemplo la automatización de toda la empresa o sólo una de sus áreas, departamentos o secciones), el mismo caso puede darse para el factor financiamiento.

En ésta etapa, un punto importante que no debe menospreciarse es la adecuada comunicación entre usuario-analista-administración.

En el CUADRO II.1 y en el DIAGRAMA II.1 se resume la etapa de definición del problema.

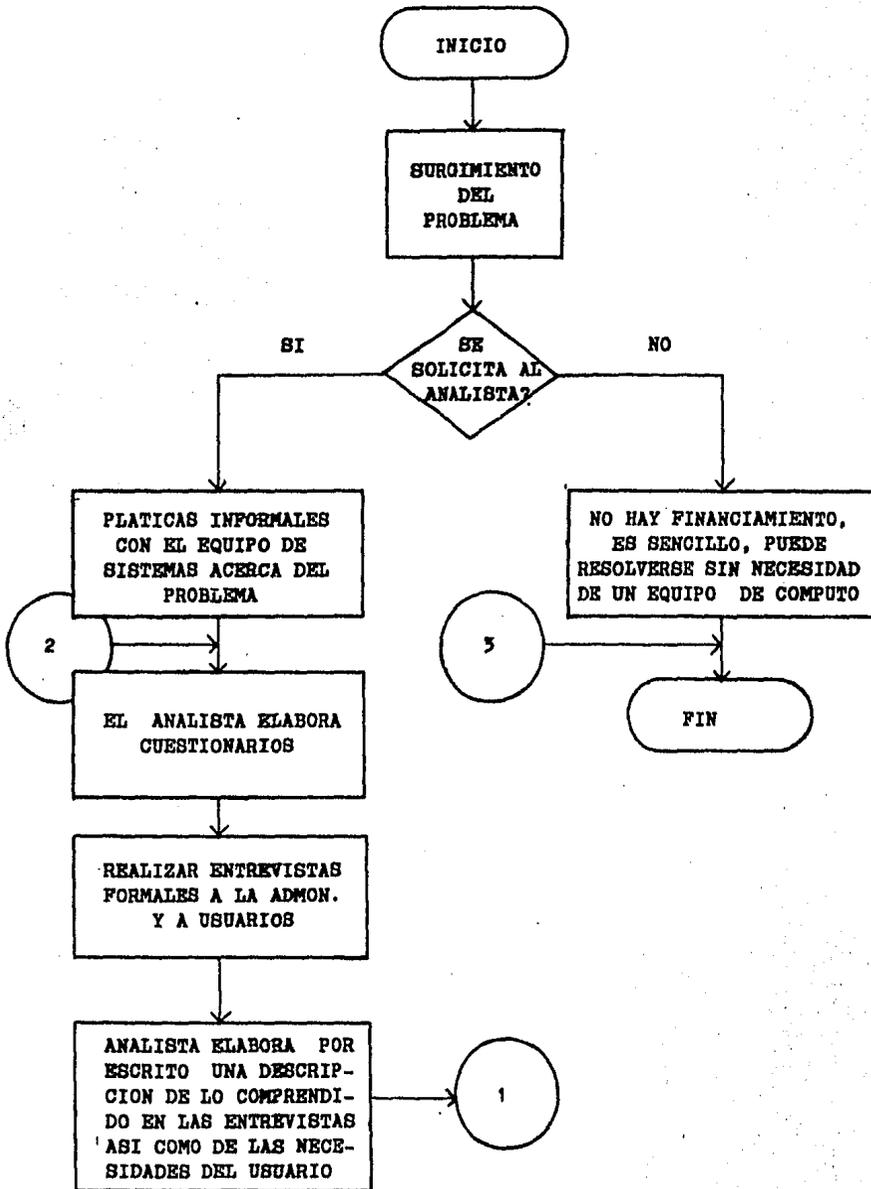
CUADRO II.1

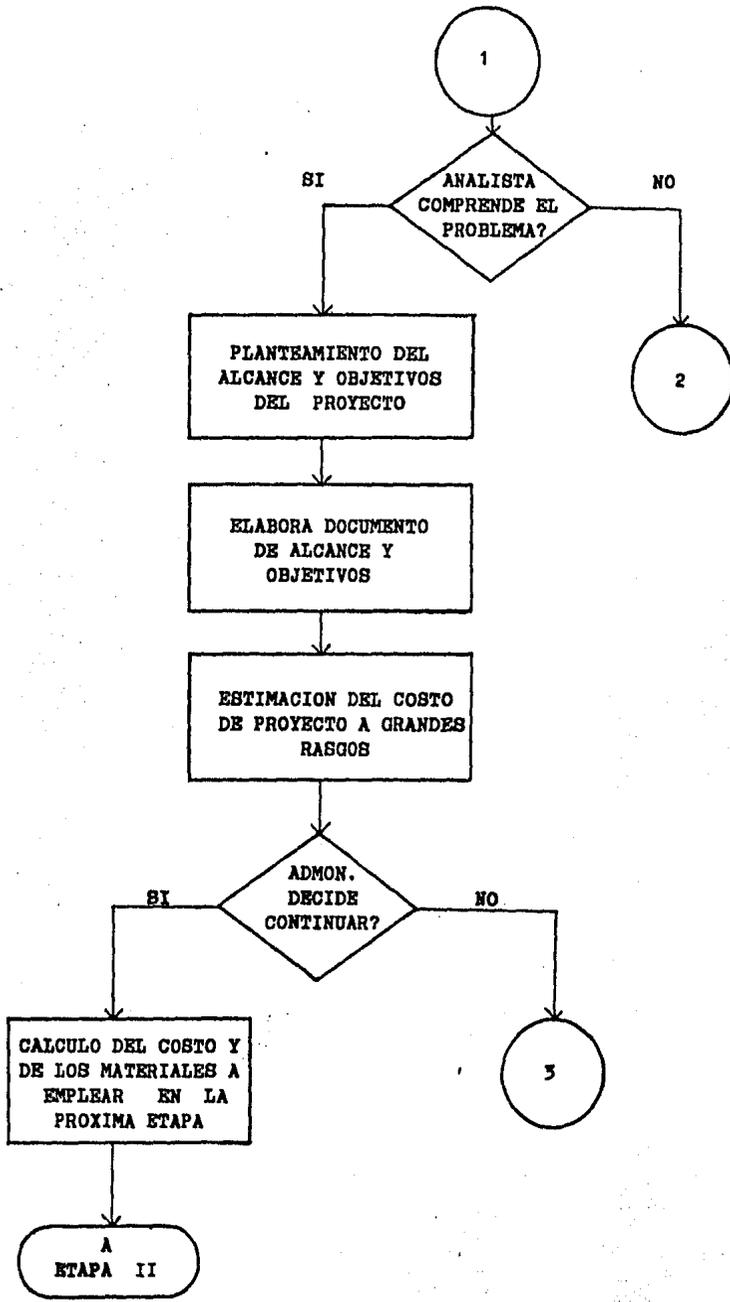
ETAPA: DEFINICION DEL PROBLEMA

! PREGUNTA A !	! CUAL ES EL PROBLEMA? !
! CONTESTAR !	! !
! !	! !
! A !	! - PLATICAS INFORMALES . !
! C !	! - SOLICITUD AL ANALISTA Y A SU EQUIPO DE TRABAJO !
! T !	! PARA QUE CONSIDERE EL PROBLEMA. !
! I !	! - ANALISTA ENTREVISTA A USUARIOS Y A ADMON. POR ME- !
! V !	! DIO DE CUESTIONARIOS Y/O EN FORMA ORAL (PLATICAS- !
! I !	! FORMALES). !
! D !	! - RETROALIMENTACION ANALISTA VS USUARIOS Y ADMON. !
! A !	! - ESTIMACION DE COSTOS GENERALES ASI COMO COSTOS DE !
! D !	! LA ETAPA SIGUIENTE. !
! E !	! !
! S !	! !
! !	! !
! DOCUMENTOS !	! - CUESTIONARIOS. !
! GENERADOS !	! - DESCRIPCION DEL PROBLEMA POR PARTE DEL ANALISTA. !
! !	! - DOCUMENTO DE ALCANCE Y OBJETIVOS DEL PROYECTO. !
! !	! !
! CRITERIO !	! PLANTEAR EL ALCANCE Y LOS OBJETIVOS !
! DE EXITO !	! DEL PROYECTO !
! !	! !

DIAGRAMA II.1

ETAPA: DEFINICION DEL PROBLEMA





CUADRO II.2

PUNTOS QUE DEBE CONTENER EL DOCUMENTO DE ALCANCE Y OBJETIVOS

FECHA
NOMBRE DEL PROYECTO
DEFINICION DEL PROBLEMA
OBJETIVOS DEL PROYECTO
ALCANCE DEL PROYECTO
IDEAS PRELIMINARES
DURACION Y COSTO DEL ESTUDIO DE FACTIBILIDAD

3.2 ESTUDIO DE FACTIBILIDAD.- La palabra "FACTIBILIDAD", tal y como se presenta aquí no existe en el diccionario, sin embargo, puede deducirse que deriva del término "FACTIBLE" cuyo origen está en la palabra latina "Fácere" (hacer). Por tanto, una cosa factible es algo que se puede hacer, algo posible, realizable o susceptible de ser hecho.

Existe también el término "VIABILIDAD" el cual si tiene un significado exacto y se define en el diccionario como algo viable, capaz de vivir. Este término se aplica a las cosas susceptibles de ser realizadas o a aquellas que por sus circunstancias tienen posibilidad de éxito. Su origen proviene del término "viable", de vie, vida, del latín vita, vivir.

Los dos términos anteriores, (Factibilidad y Viabilidad) además de ser usados en otras áreas dentro de la administración, son usados en el área de sistemas, en donde el más citado es el primero. Algunas personas incluso los manejan como dos términos distintos, pero si observamos el significado de ambas palabras en el diccionario llegaremos a la conclusión de que son sinónimos y es así como aquí las trataremos.

El estudio de viabilidad o factibilidad es aquel que se hace para decidir si se realiza o no un proyecto en base a ciertos parámetros o condiciones. El objetivo del estudio de factibilidad es determinar si hay alguna solución posible que aporte un beneficio a la organización, que pueda ser alcanzada desde todos los puntos de vista: financiera, técnica y operacionalmente. En caso de que no exista solución alguna, cualquier esfuerzo, tiempo y dinero consumido en el desarrollo del proyecto probablemente será derrochado. Por ello la razón de ser de este estudio preliminar.

Es necesario determinar rápidamente, con un mínimo de gastos, si el problema puede resolverse. El costo del estudio de factibilidad deberá representar aproximadamente entre el 5 y 10% del costo total estimado del proyecto. Por lo tanto, el analista debe recordar que generalmente el tiempo y el dinero son limitados, aunque estos dos factores dependerán del alcance del proyecto, es decir, de la magnitud del mismo. "El objetivo no es resolver el problema sino determinar si vale la pena resolverlo" (11).

(11) DAVIS. OP.CIT. PAG. 30.

La estimación de los costos en el estudio de factibilidad debe ser confiable ya que finalizando esta etapa se decidirá si se da o no marcha al proyecto y si los beneficios justifican la inversión. Por lo anterior podemos notar que el análisis del costo/beneficio de la alternativa que se proponga es una parte importante del estudio.

William S. Davis menciona tres diferentes tipos de estudios de viabilidad que son:

<u>TIPO</u>	<u>PREGUNTA A LA QUE DEBE CONTESTAR</u>
- TECNICO	- EL SISTEMA PUEDE SER IMPLANTADO EMPLEANDO LA TECNOLOGIA DISPONIBLE?
- OPERACIONAL U ORGANIZACIONAL	- EL SISTEMA PUEDE SER IMPLANTADO EN ESTA ORGANIZACION?
- ECONOMICO	- LOS BENEFICIOS SOBREPASAN LOS COSTOS?

Por otra parte, las actividades a realizar dentro de esta etapa son las siguientes:

- a) Clarificación del Alcance y Objetivos del Sistema.
- b) Estudio del Sistema ya Existente.
- c) Desarrollo de un Modelo Lógico de Alto Nivel del Sistema a proponer.
- d) Redefinición del Problema en Base a lo ya Estudiado.
- e) Desarrollo y Evaluación de Alternativas de Solución (Análisis del Costo/Beneficio).
- f) Recomendación de un Curso de Acción.
- g) Desarrollo de un Plan de Acción Tentativo.
- h) Elaboración del Documento y Presentación del Estudio.

A continuación se explica cada una:

a) Clarificación del Alcance y Objetivos del Sistema. - A pesar de que en la primera etapa el alcance y objetivos del sistema ya fueron clarificados, será necesario que el analista revise nuevamente lo que el usuario y la administración desean en realidad, es decir debe confirmar la definición del problema, el alcance anticipado del proyecto y los objetivos del sistema.

En estos momentos es conveniente entrevistarse con el personal clave y revisar los documentos que se tienen referentes a la situación de que se trate. El analista debe preguntarse si realmente se encuentra situado en el punto central del problema o bien si está divagando.

Es importante que el analista comprenda bien lo que realmente significan el alcance y los objetivos antes de comenzar a indagar. Si el sistema ya existe y sólo se desea estudiarlo para modificarlo y actualizarlo, debe recordarse que el objetivo no es documentar su existencia sino entenderlo para valorarlo. El llevar a cabo lo anterior requiere de gran preparación por parte del analista pues se necesita que primero comprenda el sistema y después emita su juicio y proponga mejoras.

b) Estudio del Sistema ya Existente.--Dependiendo de cada caso en particular y de la magnitud y tipo de la empresa, un sistema de información puede o no existir. En muchos casos, el sistema de este tipo no existe ya que los flujos de información y comunicación se llevan a cabo de una manera informal y en tal caso no puede decirse que se trata de un sistema de información ya que no cumple con los elementos que caracterizan a un sistema de esta índole: coordinación entre partes, o subsistemas, objetivos definidos, eficiencia en los flujos de información, etc. Esta situación sucede a menudo en organizaciones cuya administración se lleva a cabo en forma empírica.

Por otra parte, cuando el sistema de información si existe pueden estarse presentando en el algunas deficiencias y contradicciones, de ahí la utilidad de realizar un estudio al sistema ya existente antes de desarrollar uno nuevo, además de que este sistema actual representa una importante fuente de información.

Los tipos de sistemas de información que pueden existir en una organización son los siguientes:

<u>SISTEMA EXISTENTE</u>	<u>SISTEMA DESEADO</u>
- MANUAL	- MANUAL MEJORADO O ELECTRONICO
- MECANICO	- MECANICO MEJORADO O ELECTRONICO
- ELECTRONICO	- ELECTRONICO MEJORADO, AUMENTADO Y/O CORREGIDO

El estudio comienza entrevistando al personal idóneo que pueda describir al analista el funcionamiento del sistema actual. La idea es comenzar partiendo de lo que se sabe, hablar con la gente y estudiar las funciones que se dan; el analista debe permitir que la gente se explique ampliamente para que así se vayan encontrando las partes desconocidas del sistema, y los problemas que se tengan con el. Recordemos que la relación entre el analista de sistemas y el usuario es semejante a la relación entre un doctor y su paciente, por lo tanto debe haber continuas entrevistas en las cuales el usuario describirá el sistema existente y los síntomas del mismo para así conocer sus características, deficiencias y/o aciertos.

Cuando el analista realice el estudio del sistema existente debe:

- Analizar cuidadosamente los procedimientos y la documentación dibujando los flujos de trabajo.
- Construir los diagramas de flujo de datos y el diccionario de datos del sistema ya existente (elaboración de su modelo lógico) pero procurando no entrar en detalles: Algoritmos o Diagramas de Flujo de Programas (en caso de que el sistema actual sea electrónico).
- Estudiar la relación que hay entre el sistema y otros sistemas de información que afecten al primero.
- Investigar cuanto cuesta mantener en operación el presente sistema y los beneficios tangibles y/o intangibles que generará.

De ésta manera, el conocimiento del sistema actual (apropiación o extracción de la realidad) quedará simplificado en documentos que darán al analista un perfil completo acerca de éste.

La conclusión a la que puede llegarse como resultado del estudio del sistema existente puede ser alguna de las siguientes:

- Crear un nuevo sistema al que se le incorporen elementos útiles del actual.

- Corregir al sistema existente pero sólo en forma parcial, es decir estudiar únicamente sus fallas o deficiencias y corregirlas.
- Que no se requiera un nuevo sistema pues el actual si puede cubrir las necesidades presentes y las futuras. Se considera que esta conclusión debe ser la menos probable de llevarse a cabo ya que un sistema de información, al ser dinámico está siempre en constante cambio, lo cual provocará en un momento dado, aún cuando el sistema funciona bien en el presente, que en el futuro sea deficiente.

Si la última conclusión llega a tomarse se estaría dejando a un lado el término "Planeación a Largo Plazo" que no es más que la consideración de condiciones futuras que afectarán al presente sistema.

Finalmente, el costo de operación del sistema existente puede representar otro parámetro importante para decidir que se hará. Si resulta por ejemplo que un nuevo sistema no provee beneficios adicionales (ya sean tangibles o intangibles) y/o reduce costos, el viejo sistema obviamente se mantendrá.

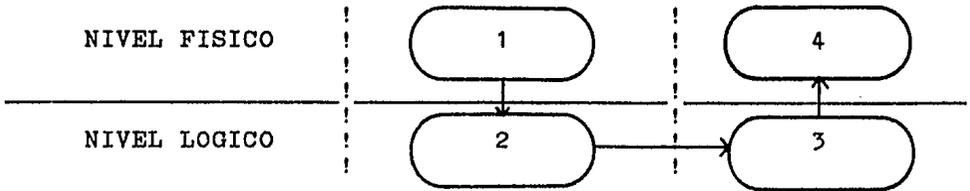
c) Desarrollo de un Modelo lógico de Alto Nivel del Sistema a Proponer.- Cuando hablamos de un modelo lógico de alto nivel nos referimos a aquellos modelos que se desarrollan en términos generales sin considerar sus detalles ya que únicamente son descritos a grandes rasgos.

A estas alturas, el analista debe tener ya una idea de las funciones que caracterizarán al nuevo sistema propuesto (en caso de que se decida generar uno nuevo).

El modelo lógico del nuevo sistema puede ser construido usando los diagramas de flujo de datos y el diccionario de datos (estas herramientas se explican en la etapa de análisis). Tal modelo podrá ser usado en el diseño del nuevo sistema. Un buen diseño comienza desde la existencia física del sistema, del cual se elaborará un modelo lógico, este modelo lógico será usado para construir otro nuevo modelo lógico que contendrá las características del sistema propuesto; tal modelo será la base para desarrollar el nuevo sistema físico (ver GRAFICA II.1).

GRAFICA II.1

RELACION ENTRE EL NIVEL FISICO Y EL NIVEL LOGICO



El nuevo modelo lógico que desarrolla el analista referente al sistema propuesto deberá explicar lo que el considera que hará tal sistema.

d) Redefinición del Problema en Base a lo ya Estudiado.- Después de que el analista ha desarrollado el modelo lógico del sistema que propone, es necesario platicar con el o los usuarios para saber si realmente están de acuerdo con el modelo propuesto.

Por otro lado, se debe resolver la cuestión de que si realmente los planteamientos iniciales del alcance y objetivos del proyecto así como de la definición del problema son válidos o por el contrario si éstos deben modificarse. Para resolver lo anterior es necesario retroalimentarse con el o los usuarios tomando como base de la discusión los diagramas de flujo de datos y el diccionario de datos.

El objetivo de esta actividad es encontrar aquellos aspectos que el analista y/o el usuario hayan pasado por alto para poder dejar bien sentados los planteamientos de la definición del problema, del alcance y de los objetivos del proyecto.

Las primeras cuatro actividades del estudio de factibilidad pueden considerarse como un ciclo constante (loop) que se repetirá hasta que no se logre definir exactamente lo deseado: El analista estudia y define el problema, desarrolla un modelo lógico del sistema físico existente, desarrolla una

solución tentativa, revisa la solución, redefine el problema, lo reanaliza, identifica algunos elementos no tomados en cuenta y repite el proceso hasta que el modelo lógico sea casi exacto, responda a las necesidades presentadas y hasta que el problema, alcance y objetivos estén bien claros.

Cuando este punto es alcanzado el analista podrá comenzar a considerar como deberá ser diseñado el sistema.

e) Desarrollo y Evaluación de Alternativas de Solución.- Dado el modelo lógico del sistema propuesto el analista puede comenzar a generar algunas alternativas físicas que podrían solucionar la cuestión. Para generar tales alternativas se tienen las siguientes opciones:

- Agrupar en los diagramas de flujo de datos diferentes procesos de diversas maneras cada una de las cuales generará un sistema físico diferente, una posible solución física. Todas las soluciones que se generen deberán ser estudiadas.
- Usar la conocida técnica de lluvia de ideas. Dado el modelo lógico del sistema propuesto y un tiempo adecuado para llevar a cabo la sesión, el analista y algunos técnicos colegiados podrán reunir una lista de alternativas. Las reglas a seguir al emplear esta opción son muy sencillas y son las siguientes:
 - >> Cada elemento del grupo deberá proponer una solución posible.
 - >> No deberá criticarse ni hacerse una evaluación personal de cada una de las sugerencias dadas por los demás.
 - >> El grupo de trabajo debe conocer perfectamente el problema, los objetivos y el alcance del proyecto, así como otros aspectos.
 - >> Terminada la sesión, el analista evaluará cada alternativa sugerida y seleccionará las que le parezcan más razonables. Nótese que aquí su criterio y experiencia son determinantes.

- Otra opción es crear una matriz o tabla en donde se crucen distintas variables que dan una serie de combinaciones o alternativas. Para realizar lo anterior deben considerarse los siguientes factores:

- >> Magnitud del Proyecto (Alcance).
- >> Tipo de Sistema a Desarrollar.
- >> Características del mismo.
- >> Magnitud de la Empresa.
- >> Perspectivas a Futuro de la misma en cuanto al proyecto.
- >> Fondos o Presupuestos que se destinan al proyecto.
- >> Tipo de Empresa y Características del Personal (tipo de persona, nivel de estudios, etc.)

En el CUADRO II.3 se ejemplifica la manera en la que pueden generarse las alternativas. Cada alternativa originada deberá analizarse considerando las condiciones generales tanto del proyecto como de la empresa. Por ejemplo si se requiere un tiempo de respuesta corto, la alternativa Batch será descartada, o bien si se pretende desarrollar un sistema integral de información (computarizar toda una empresa), probablemente lo mejor será adquirir una macrocomputadora con terminales conectadas a ella que estén situadas en cada una de las áreas físicas de la organización en donde se requieran. Este estudio es lo que ya denominamos al principio de esta etapa como estudio técnico de factibilidad.

En cuanto al estudio de operacional u organizacional se debe considerar por ejemplo, si el usuario se opone o no a un servicio externo de agencia o bien a un servicio de tiempo compartido, o si hay alguna tendencia que se oponga a un tipo de hardware en particular. En este tipo de decisiones intervienen las políticas y acuerdos de la compañía.

Básicamente, el analista debe revisar cada alternativa considerando al personal implicado, a los aspectos operacionales y de eficiencia, eliminando aquellas que no respondan a los requerimientos y condiciones.

En el estudio de factibilidad desde el punto de vista económico deben considerarse los siguientes aspectos para la evaluación de las alternativas propuestas:

- Costos de Desarrollo del Proyecto.
- Costos de Operación.
- Tasa de Retorno de la Inversión.
- Valor Presente y Período de Recuperación de la Inversión.

En una palabra, el análisis del costo beneficio de cada alternativa.

CUADRO II.3

GENERACION DE ALTERNATIVAS TECNICAS (12)

FUENTE	TIPO	MICRO	MINI	PROCESADOR CENTRAL	SERVICIOS D'AGENCIA	SERVICIOS TIEMPO COMP
	BATCH					
INTERNA	INTERACTIVO					
	TIEMPO REAL					
	BATCH					
EXTERNA	INTERACTIVO					
	TIEMPO REAL					
CONTRATAR	BATCH					
LA REALI-	INTERACTIVO					
ZACION DEL	TIEMPO REAL					
SISTEMA						

Cada alternativa pasará por una evaluación técnica, económica y operacional y se desarrollará para cada una un posible bosquejo de implantación, el cual no será muy detallado. A todo lo anterior se agregará un breve plan en el que se especificarán las fechas de terminación de cada una de las etapas del ciclo de vida del nuevo sistema.

(12) TOMADO DE: WILLIAM S. DAVIS. OP.CIT. PAG. 39.

El analista debe informar tanto a los usuarios como a la administración de un cierto número de alternativas estudiadas tales como:

- Un sistema de bajo costo con los posibles resultados que esto pueda traer.
- Un sistema de costo intermedio que efectúe solamente lo deseado y necesario.
- Un sistema de alto costo que sea de aplicación general y abierta del cual pueda obtenerse lo que se desee.
- El sistema existente (como parámetro de comparación).
- Compra de un sistema comercial.
- Un sistema manual, etc.

Resulta de más agregar que deben estudiarse las ventajas y desventajas de cada alternativa posible.

f) Recomendación de un Curso de Acción.- El analista indicará como producto de la evaluación de las alternativas, si es factible o no llevar a cabo el proyecto. En caso afirmativo él mismo seleccionará una alternativa, la mejor, justificando las razones Técnicas, Económicas y Operacionales de su elección.

Generalmente, una empresa que es administrada científicamente, y que su tamaño es considerable, siempre está buscando alternativas de inversión que generen un beneficio y el desarrollo de un nuevo sistema es una forma de invertir.

Si la mejor alternativa para el sistema propuesto ofrece, por ejemplo un retorno de la inversión proyectado en un 15% y una tasa de interés del 16% la administración escogerá tal vez poner el capital en el banco en vez de desarrollar el proyecto. Sin embargo, se considera que lo anterior no es muy válido ya que la justificación para realizar o no un proyecto no sólo se hace desde el punto de vista económico sino desde el punto de vista funcional, es decir, de los beneficios intangibles y a largo plazo que el sistema de información proporcionará a la organización.

Si se requiere de un sistema funcional y eficiente que remplace a un sistema obsoleto, tardado e ineficiente, y si se decide el desarrollo de este nuevo sistema, no precisamente por la rentabilidad que vaya a ofrecer sino por los beneficios intangibles, que a la larga pasarán a mejorar las utilidades de la organización como parte del giro de la misma, se estará probablemente sacrificando el dinero de una inversión que no generará beneficios a corto plazo sino a largo plazo. Esto es completamente válido y correcto, es administrar de una manera adecuada los recursos de la organización.

A pesar de lo anterior, generalmente las decisiones de inversión se basan en su tasa de recuperación por lo que siempre se incluye el análisis de costo/beneficio que es a final de cuentas lo que le interesa a la administración.

g) Desarrollo de un Plan de Acción Tentativo.- El plan estará basado en el curso de acción recomendado. A pesar de que a estas alturas es difícil estimar con seguridad los tiempos y gastos requeridos para desarrollar el proyecto, ya que no se conocen sus detalles, es necesario hacer algunos cálculos basándonos en el ciclo de vida del sistema (cálculos por etapas), con lo cual podremos lograr un plan de acción más o menos objetivo y cercano a la realidad. El plan contendrá para cada una de las etapas el tiempo, costo y personal requerido.

El plan proveerá a la administración la información que ésta requiera en cuanto a la duración del proyecto y el número de personas a emplear en él.

h) Elaboración del Documento y Presentación del Estudio.
- A pesar de que cada caso es diferente, los puntos generales que debe contener el documento, producto del estudio de factibilidad son los siguientes:

- Títulos: Nombre del proyecto, título del reporte, autores(es) y fecha.
- Índice o Contenido.
- Definición del Problema: Claro y conciso, de preferencia en una sola página.

- Resumen de lo Ejecutado: Debe ser claro y conciso, resumiendo en una página el estudio de factibilidad, los resultados y recomendaciones. Incluir autorizaciones, fuentes clave de información, alternativas consideradas, alternativas no consideradas, costos, beneficios y tiempos asociados con la alternativa recomendada.
- Método de Estudio: Describir los procedimientos usados en la realización del estudio de factibilidad. Mencionar las fuentes, las referencias y el personal clave. Describir brevemente el sistema existente.
- Análisis: Se analiza el sistema lógico propuesto, se describen los objetivos, el alcance del sistema de manera detallada, se incluyen los diagramas de flujo de datos, el diccionario de datos del sistema propuesto y se identifican las relaciones de éste con otros sistemas.
- Alternativas Consideradas: Para cada alternativa considerada se describe su estudio de factibilidad técnico, económico y operacional, así como un plan burdo de su implantación, un diagrama de flujo del sistema y otras descripciones del mismo. Los detalles irán en el apéndice.
- Recomendaciones: Recomendar un curso de acción de una manera clara y breve justificando tal recomendación con bases tales como el análisis de costo/beneficio.
- Desarrollo del Plan: Incluye materiales a emplear, tiempo y costos proyectados a erogar en cada fase del ciclo de vida del sistema, asumiendo que se seguirá el curso de acción recomendado.
- Apéndice: Puede incluir los siguientes documentos:

>> Cartas	>> Gráficas
>> Entrevistas	>> Estadísticas
>> Notas	>> Referencias
>> Contactos clave	>> Reconocimientos
>> Diagramas	>> Memorandums, etc.

Es decir, todos los detalles que le dan base a un estudio.

El estudio de factibilidad finaliza con la presentación formal de los resultados de dicho estudio tanto al usuario como a la administración. Tales resultados estarán incluidos, como ya se vió, en el informe final de esta etapa. Esta presentación será de crucial importancia para decidir si el proyecto sigue en marcha o no. Esta decisión será resuelta por la administración de la organización más no por el analista.

Muchos proyectos morirán en esta etapa y sólo quedarán aquellos que aseguren principalmente un promisorio retorno de la inversión. Si la administración aprueba el proyecto, el estudio de factibilidad representará una base excelente para el entendimiento del problema y un claro sentido de dirección para el desarrollo subsecuente del sistema.

En el CUADRO II.4 se resume la etapa del estudio de factibilidad y así mismo se muestra en el DIAGRAMA II.2 de la siguiente página la manera de llevar a cabo tal etapa.

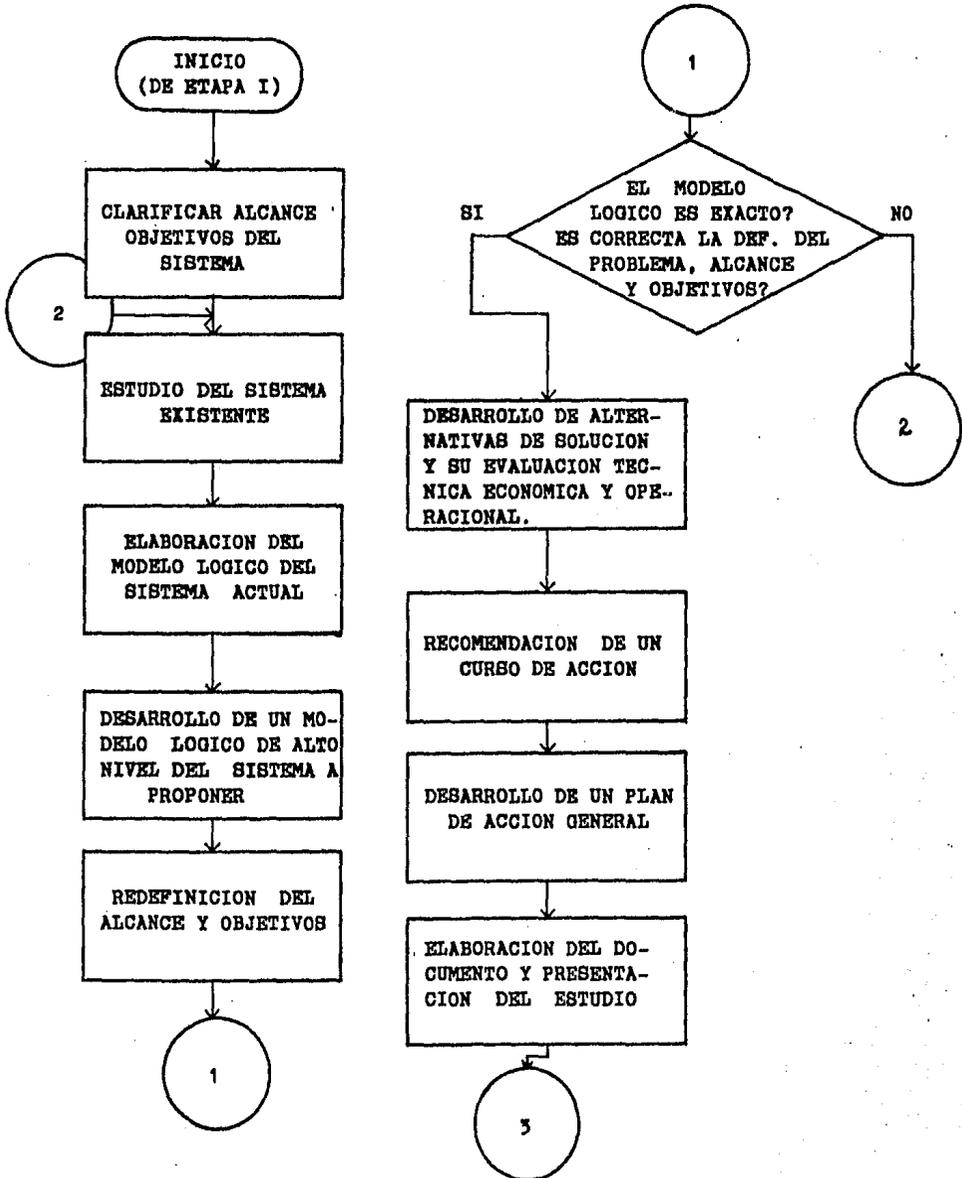
CUADRO II.4

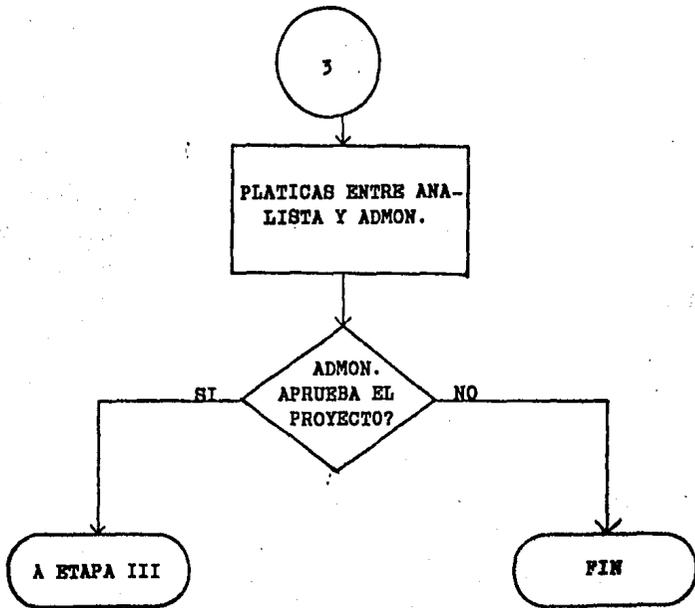
ETAPA: ESTUDIO DE FACTIBILIDAD

PREGUNTA A CONTESTAR	CUAL ES EL PROBLEMA?
A C T I V I D A D E S	<ul style="list-style-type: none"> - CLARIFICAR EL ALCANCE Y OBJETIVOS DEL SISTEMA. - ESTUDIO DEL SISTEMA YA EXISTENTE. - DESARROLLO DE UN MODELO LOGICO DE ALTO NIVEL DEL SISTEMA A PROPONER. - REDEFINICION DEL PROBLEMA EN BASE A LO YA ESTUDIADO. - DESARROLLO Y EVALUACION DE ALTERNATIVAS DE SOLUCION. - RECOMENDACION DE UN CURSO DE ACCION. - DESARROLLO DE UN PLAN DE ACCION TENTATIVO - ELABORACION DEL DOCUMENTO Y PRESENTACION DEL ESTUDIO.
DOCUMENTOS GENERADOS	<ul style="list-style-type: none"> - ESTUDIO DE FACTIBILIDAD. - PLAN DE ACCION A GROSSO MODO. - DIAGRAMAS DE FLUJO DE DATOS Y DICCIONARIO DE DATOS TANTO DEL SISTEMA EXISTENTE COMO DEL SISTEMA A PROPONER (A GROSSO MODO Y SIN ENTRAR EN DETALLES).
CRITERIO DE EXITO	<ul style="list-style-type: none"> - REALIZAR LOS TRES TIPOS DE ESTUDIOS: OPERACIONAL, TECNICO Y ECONOMICO (ANALISIS COSTO/BENEFICIO). - PLANTEAR UN CURSO DE ACCION.

DIAGRAMA II.2

ETAPA: ESTUDIO DE FACTIBILIDAD





3.3 ANALISIS.- Se le denomina análisis a "...la separación de un sistema en sus componentes y el estudio de cada componente en forma independiente así como su relación con los otros componentes. Las coacciones adicionales (internas o externas) tienen que ser incluidas en el análisis para determinar su impacto sobre las decisiones que afectan la existencia del sistema" (13).

Todas las actividades de la etapa de análisis se basan en la documentación generada durante el estudio de factibilidad, especialmente en los diagramas de flujo de datos.

El objetivo del análisis es determinar que es lo que debe hacer el sistema para así poder resolver el problema del usuario, el cual es el que lo conoce totalmente. Durante el análisis se debe desarrollar un esquema completo y funcional del sistema propuesto. El hecho no es determinar COMO deberá trabajar el sistema sino QUE es lo que debe hacer.

Se elaborarán con más detalle los diagramas de flujo de datos, el diccionario de datos o catálogo de datos y una serie de algoritmos los cuales se describen en una clase de documentos preliminares denominados IPO (Input/Proces/Output). De esta manera poco a poco se irá definiendo el diseño lógico del sistema propuesto con mayor detalle.

Seguidamente, la administración y el usuario aprobarán los resultados los cuales serán la base para el inicio de la próxima etapa. El análisis termina con una inspección técnica realizada por un equipo de trabajo ajeno al proyecto el cual investigará si los criterios de éxito de esta etapa se han cumplido. Al mismo tiempo, la administración revisará el trabajo del analista pero desde el punto de vista de los beneficios que podrán obtenerse.

La problemática que puede surgir en esta etapa es que muchas veces el técnico desea finalizar rápidamente el análisis para comenzar con el diseño; esta tentación debe ser desechada. La metodología estructurada nos ayuda a que el técnico no cree un modelo físico prematuro con base en un modelo lógico apresurado, incompleto y falto de objetividad. Para evitar problemas como el anterior, la metodología estructurada establece siempre en cada fase criterios de éxito, los cuales, mientras no sean cumplidos no se proseguirá con la

siguiente etapa. Los criterios de éxito de la etapa de análisis se muestran en el CUADRO II.7.

Las actividades principales que se realizan en la etapa de análisis, en la cual todas las funciones lógicas del sistema propuesto deben quedar definidas de una manera más completa, son las siguientes:

- a) Definición de los Datos.
- b) Estudio del Modelo Lógico del Sistema Propuesto.
- c) Análisis de los Diagramas de Flujo de Datos.
- d) Inspección Técnica.
- e) Revisión por la Administración.
- f) Documentación Formal.

a) Definición de los Datos. - La etapa de análisis respecto a los datos pretende hacer una definición de éstos así como de sus fuentes y sus flujos. Se comenzará por lo tanto, declarando los campos que serán objeto del procesamiento, ya que éstos son la clave para definir el tipo de procesos.

Se definirán posteriormente las salidas (resultados que debe generar el proceso global), razón de ser de todo sistema. Para que el analista conozca el origen de tales salidas, debe regresar a revisar los diagramas de flujo de datos que fueron realizados de manera general en la etapa del estudio de factibilidad. En caso de que el analista no conozca el origen de algunos datos, o bien el de todos los que participarán en el proceso, puede entrevistarse con los usuarios expertos con la finalidad de definir los detalles funcionales del sistema.

Es muy común que uno o más campos resulten de la operación o mezcla de otros, por lo tanto, el analista deberá estudiar el origen de cada dato.

b) Estudio del Modelo Lógico del Sistema Propuesto. - Debido a que en la etapa del estudio de factibilidad el modelo lógico del sistema propuesto se efectuó sin adentrar en detalles, a estas alturas, aún siguen existiendo dudas en el analista en cuanto a algunas especificaciones. El debe encontrar respuesta segura a las siguientes cuestiones:

- El Diccionario de Datos es Exacto?.

- Está Completo?
- Los Algoritmos están Completos y son Correctos?
- Hay Algún Detalle que Falte?
- Cuales son las Fuentes Verdaderas de los Datos?

Obsérvese que la respuesta a estas preguntas requiere que el analista sea una persona minuciosa, detallista y paciente que realice el trabajo de una manera lenta, pero segura para que de esta forma, puedan verse resultados finales satisfactorios y se evite caer en equivocaciones cuyas consecuencias generalmente se dejan ver en las etapas finales.

Por lo tanto debe analizarse minuciosamente el flujo de cada uno de los datos en los diagramas correspondientes. Al efectuar ésto suelen surgir cuestiones desconocidas o poco claras que den lugar a una entrevista con el usuario, fuente primaria de información. De esta manera, paulatinamente se agregarán detalles a los diagramas de flujo de datos existentes.

La "espiral" del análisis estructurado es como un ciclo que se lleva a cabo como sigue:

- Un análisis preliminar generará preguntas.
- Las respuestas a estas preguntas darán una comprensión más detallada del sistema y posiblemente se podrán generar más preguntas a partir de ello.
- Las respuestas a estas últimas preguntas más detalladas darán origen a un entendimiento cada vez mejor y quizá a más preguntas.
- Cada aproximación envolverá más detalles acerca del sistema lógico y posiblemente se descubrirá que se requieren de otros procesos nuevos para completar y/o complementar el sistema. De esta forma quedará armado el diseño lógico del sistema propuesto.

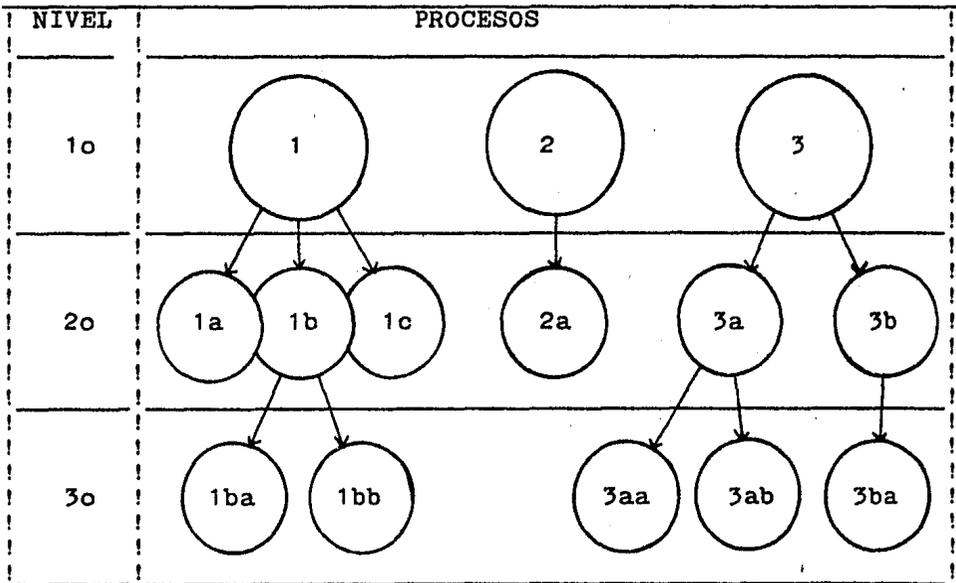
En esta espiral del análisis estructurado juega un importante papel la documentación formal de la cual se hablará posteriormente en el inciso f.

c) Análisis de los Diagramas de Flujo de Datos.- Como se ha mencionado ya, el analista debe aumentar su visión general acerca del flujo lógico del sistema propuesto, es decir debe profundizar en el por medio del análisis de los diagramas de flujo de datos.

Los diagramas de flujo de datos se analizan por medio de su descomposición funcional, por ejemplo, si escogemos uno de los tantos procesos que existen en un diseño lógico y después lo descomponemos en subfunciones, éstas subfunciones llegarán a ser procesos de un nuevo diagrama de flujo de datos. Lo mismo podrá hacerse para todos los demás procesos del 1o nivel (ver GRAFICA II.2).

GRAFICA II.2

ANALISIS DE LOS DIAGRAMAS DE FLUJO DE DATOS POR MEDIO DE LA DESCOMPOSICION DE LOS PROCESOS PRINCIPALES



Para cada proceso, subproceso y subsubproceso habrá un diagrama de flujo de datos, si el proceso principal así lo requiere. En el gráfico anterior sólo se explican los niveles de aproximación hacia el detalle partiendo de los procesos generales y principales los que darán origen a subprocesos y así sucesivamente. La operación anterior puede no ser necesaria cuando el sistema es sencillo y pequeño, pero cuando el sistema es complicado y de mayor tamaño, ésta aproximación o análisis resulta útil; en tal caso puede elaborarse una lista de los procesos más importantes de un algoritmo o diagrama de flujo de datos general, después analizar cada uno y describir las funciones que le corresponden haciendo de cada una de ellas un subproceso con su correspondiente diagrama de flujo de datos. Podrá haber procesos principales que no requieran subdividirse para analizarse, por ejemplo la emisión de un reporte.

Cuando el sistema se haya desmembrado, todas las funciones analizadas y diagramadas podrán incorporarse a un diagrama de flujo de datos general, así se tendrá una descripción gráfica más detallada que la que se tenía antes acerca del sistema. Sin embargo debe recordarse que no es conveniente hacer una división tan exhaustiva ya que la finalidad del análisis es entender que pasa y no como pasa.

Cuando se hace una descomposición demasiado específica o detallada, tanto que provoque que comencemos a pensar en la forma en la que se codificará para implantar el sistema, tal descomposición resultará innecesaria en esta fase ya que la codificación corresponde al desarrollo físico del sistema (Diseño Detallado) más no al lógico y por lo tanto en estos momentos no debemos preocuparnos por los detalles físicos.

d) Inspección Técnica.- El objetivo de ésta inspección es revisar los aspectos técnicos del análisis para determinar si el sistema puede ser diseñado.

La inspección es realizada por un equipo de técnicos en la materia: analista(s) y algunas personas que estén directamente en contacto con el sistema existente.

El equipo estudiará la documentación formal existente (ver inciso f), tomando en cuenta los diagramas de flujo de datos como documentos primarios y al resto como documentos de soporte.

El analista encargado del sistema responderá a cuestiones técnicas. El objetivo es encontrar algún error o hecho inadvertido y no el defender o atacar el trabajo. Si son encontrados algunos errores éstos serán corregidos por el analista. De esta manera se dará una retroalimentación constante que finalizará en una situación en la cual el equipo de técnicos estarán de acuerdo con el análisis. En estos momentos el analista ya conoce real y perfectamente el objetivo y el sistema a realizar.

Finalmente, los miembros del equipo de inspección firmarán un documento que haga patente el estudio técnico del trabajo del analista hasta esta etapa, así como su aprobación.

e) Revisión por la Administración. - Para finalizar la etapa de análisis y comenzar con el diseño, la administración hará una revisión en cuanto a las siguientes cuestiones:

- Podrá ser efectuado a tiempo el trabajo que resta? o habrá que modificar los planes y programas.
- Que hay de los costos?, se modificarán? o bien, aún es razonable la estimación hecha en el estudio de factibilidad.
- Se modificará el tiempo que se le asignó a la siguiente etapa? (Diseño)...etc.

Las actividades que engloban la etapa de análisis pueden observarse en el CUADRO II.7, así como en el DIAGRAMA II.4.

f) Documentación Formal. - Debido a que el desarrollo de un sistema envuelve en general a varias personas, es necesario que cada una de éstas conozca el trabajo de las demás, y la mejor forma para lograr esto es por medio de documentos. El analista no es el único que debe conocer el contenido de lo que hasta este momento se tiene en la documentación del sistema, es necesario que la administración y el usuario conozcan el trabajo del analista así como otras personas que puedan requerir después la información, por ejemplo los programadores.

Por las razones anteriores todo analista debe documentar su trabajo de una manera consistente y fácil de comprender.

La metodología estructurada propone tres documentos formales, necesarios en la etapa de análisis:

- f.1) Diagramas de Flujo de Datos.
- f.2) Diccionario de Datos.
- f.3) Descripción de Algoritmos en Forma General.

La utilidad que tienen estos documentos es la siguiente:

- Sirven para verificar y recordar hechos respecto a lo que sucede en el sistema lógico.
- Son auxiliares valiosos que minimizan los riesgos de equivocación por olvido o ignorancia de algún hecho.
- Son una excelente herramienta de comunicación.
- Sirven de base para iniciar la etapa de diseño, ya que es análoga a los diseños preliminares que efectúa un arquitecto o ingeniero.

La ausencia de documentación formal en la etapa de análisis puede ocasionar problemas en cuanto a que no se tendrá una visión clara de lo que se desea ni tampoco una base segura para efectuar un acertado diseño.

A continuación se describe cada uno de los documentos formales:

f.1) Diagramas de Flujo de Datos.-

- Descripción General.- Durante la etapa de análisis y diseño de procesos el analista recoge grandes cantidades de información tanto de entrevistas formales como casuales. Esta información no debe dejarse aislada, desordenada y en hojas sueltas, sino que debe recopilarse, resumirse y ordenarse ya que tal información, referente al flujo de los datos, sirve como herramienta de comunicación con el usuario, siendo así mismo bastante útil para las siguientes etapas del desarrollo del sistema.

Un diagrama de flujo de datos es un modelo lógico del sistema que no depende del hardware, software, estructura de datos u organización alguna de archivos, en una palabra, no tiene implicaciones físicas.

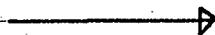
El diagrama de flujo de datos muestra el origen y destino de los datos, su flujo, almacenamiento y modificaciones en caso de que sean procesados por medio de un programa, rutina o proceso manual.

Al ser el diagrama tan sólo una gráfica del sistema lógico no conlleva dificultades para comprenderlo de ahí que el o los usuarios y/o programador(es) pueden estudiarlo y entenderlo, por ello es una buena herramienta de comunicación.

- Simbología.- Los símbolos que se usan para construir diagramas de flujo de datos se muestran en el CUADRO II.5, en donde también se hace una explicación de la función de cada uno.

CUADRO II.5

SIMBOLOGIA USADA EN UN DIAGRAMA DE FLUJO DE DATOS

SIMBOLO	FUNCION
	- FLUJO DE INFORMACION : INDICA HACIA DONDE FLUYEN LOS DATOS (1).
	- ORIGEN O DESTINO DE LOS DATOS.
	- PROCESO: INDICA TRANSFORMACION DE DATOS. CADA SIMBOLO PUEDE SE NUMERADO O ETIQUETADO EN LA PARTE POSTERIOR (2).
	- ALMACEN DE DATOS (3).

NOTAS REFERENTES AL CUADRO II.5:

(1) En los tradicionales diagramas de flujo la dirección es de arriba hacia abajo y de izquierda a derecha. Un buen diagrama de flujo de datos sigue un convencionalismo similar con el movimiento de los datos desde su fuente (extremo izquierdo) hasta su destino (extremo derecho).

(2) El proceso de transformación de datos no debe ser necesariamente un programa, puede representar una serie de programas, una rutina de alguno de éstos o bien, un proceso manual.

(3) Un almacén de datos no representa necesariamente un archivo, si no que puede representar o una parte de éste, elementos de una base de datos o bien una porción de un registro. Similarmente el almacén de datos puede representar un disco, una cinta magnética, la memoria principal, microfichas, tarjetas perforadas, e inclusive el cerebro humano. Gráficamente el símbolo puede enumerarse o etiquetarse en su extremo izquierdo.

(4) Los símbolos que presentamos aquí no son necesariamente los únicos para elaborar un diagrama de flujo de datos, sino que el analista puede emplear algunos otros que sean inventados por él o que sean tomados de otra simbología y que le sirvan para representar en tal diagrama, el flujo de los datos. Por ejemplo puede usarse aquí el diamante usado para elaborar diagramas de flujo comunes, etc. Todo estará en función de las necesidades de diagramación que se requieran para representar la realidad.

(5) El flujo de datos se diferencia del almacén de datos en que en el primer caso los datos se encuentran en movimiento y el segundo caso los datos están inmóviles, además el almacén es llenado por un flujo de datos y la fuente de este flujo es un almacén; varios flujos de datos pueden combinarse para formar un almacén de datos o uno sólo de éstos puede proveer diferentes elementos para cada uno de los flujos. Estos dos conceptos son justamente diferentes versiones de una misma cosa.

(6) Un número de suposiciones son representadas de manera resumida en el diagrama de flujo de datos. No se detallan

cuestiones, no se le pone atención a la manera en la que se procesan los datos. En resumen el objeto es describir que pasa sin preocuparse como pasa.

- Como Diagramar el Flujo de los Datos.- Para comenzar a diagramar se recomienda crear primero un bosquejo general el cual se desglosará y detallará poco a poco, descubriendo antes que nada, los orígenes y destinos de los datos así como los procesos principales. Una vez que los procesos principales han sido identificados e incorporados al diagrama, el analista podrá comenzar a descomponer cada uno de estos procesos a un nivel cada vez más detallado tal y como se hace cuando se analizan los diagramas de flujo de datos.

Pero cuando se podrá saber que se ha llegado hasta el último nivel de detalle lógico?, pues cuando hayamos obtenido el punto donde la última subdivisión requiera, para describirse, detalles físicos o de implantación del proceso.

Algunas veces el sistema será tan grande que los diagramas resultarán ser igual de grandes y enredados para quien los siga. Para facilitar su entendimiento, cada uno de los procesos principales podrán diagramarse aparte en subdiagramas, de esta manera se comprenderá el flujo de los datos siempre y cuando cada subdiagrama esté conectado con los demás.

- Revisión de los Diagramas de Flujo de Datos.- Algunas recomendaciones para asegurarnos de que nuestros diagramas están correctos son las siguientes:

- >> Cada vez que se elabore o se modifique un diagrama debe corroborarse con el usuario de que tal diagrama refleje realmente la realidad.
- >> Comparar en cada almacén de datos las afluencias y salidas de información, ya que todo dato que entra forzosamente debe salir en algún momento y si ello no sucede probablemente exista algún error.
- >> Revisar por medio del seguimiento del diagrama, de donde proviene cada campo y al mismo tiempo corroborar si hay alguna duplicación innecesaria de tales campos.

Toda retroalimentación que se haga con el usuario provocará un cambio en el diagrama, por lo tanto, es conveniente que los primeros diagramas que se elaboren sean sólo bocetos en sucio y una vez que ya no exista modificación alguna entonces si se procederá a realizar la versión final en forma presentable. De ésta forma se evitará la pérdida de tiempo, de recursos y de esfuerzos.

- Utilidad de este tipo de Diagramas.- Los diagramas de flujo de datos se consideran de gran utilidad debido a las siguientes razones:

- >> Ayudan al analista a organizar la información acerca del sistema con lo cual se logra su mejor entendimiento.
- >> Lo obligan a resumir la información extrayendo los puntos más importantes y considerando la relación entre ellos.
- >> Los elementos que requieren detallarse en forma amplia se explican fácilmente en tan sólo una porción de la estructura gráfica del diagrama.
- >> Los datos que fluyen en el diagrama así como aquellos que están almacenados, representan un base para el desarrollo de un diccionario de datos.
- >> Son una excelente herramienta en la comunicación. El número limitado de símbolos y la ausencia de detalles físicos y técnicos en éstos diagramas permiten su entendimiento por parte de la mayoría de los usuarios.
- >> Ayudan al diseño del sistema, pueden ser usados también en la etapa del diseño detallado cuando se elaboran los diagramas de flujo de cada proceso los cuales se basan en los diagramas de flujo de datos.
- >> Sirven para efectuar pruebas preliminares al diseño físico del sistema.
- >> Permiten descubrir un número razonable de alternativas para solucionar un problema ya que en ellos pueden agruparse los procesos de los datos y sus flujos

de diferente forma lo cual puede dar origen a un sistema físico diferente dependiendo de la forma de agrupación de tales procesos.

>> En conclusión, son una guía excelente para desarrollar un sistema.

- Ejemplo de un Diagrama de Flujo de Datos.- A continuación se presenta un ejemplo de un flujo de datos de tipo bibliográfico el cual se hace empleando los símbolos ya mencionados en el CUADRO II.5.

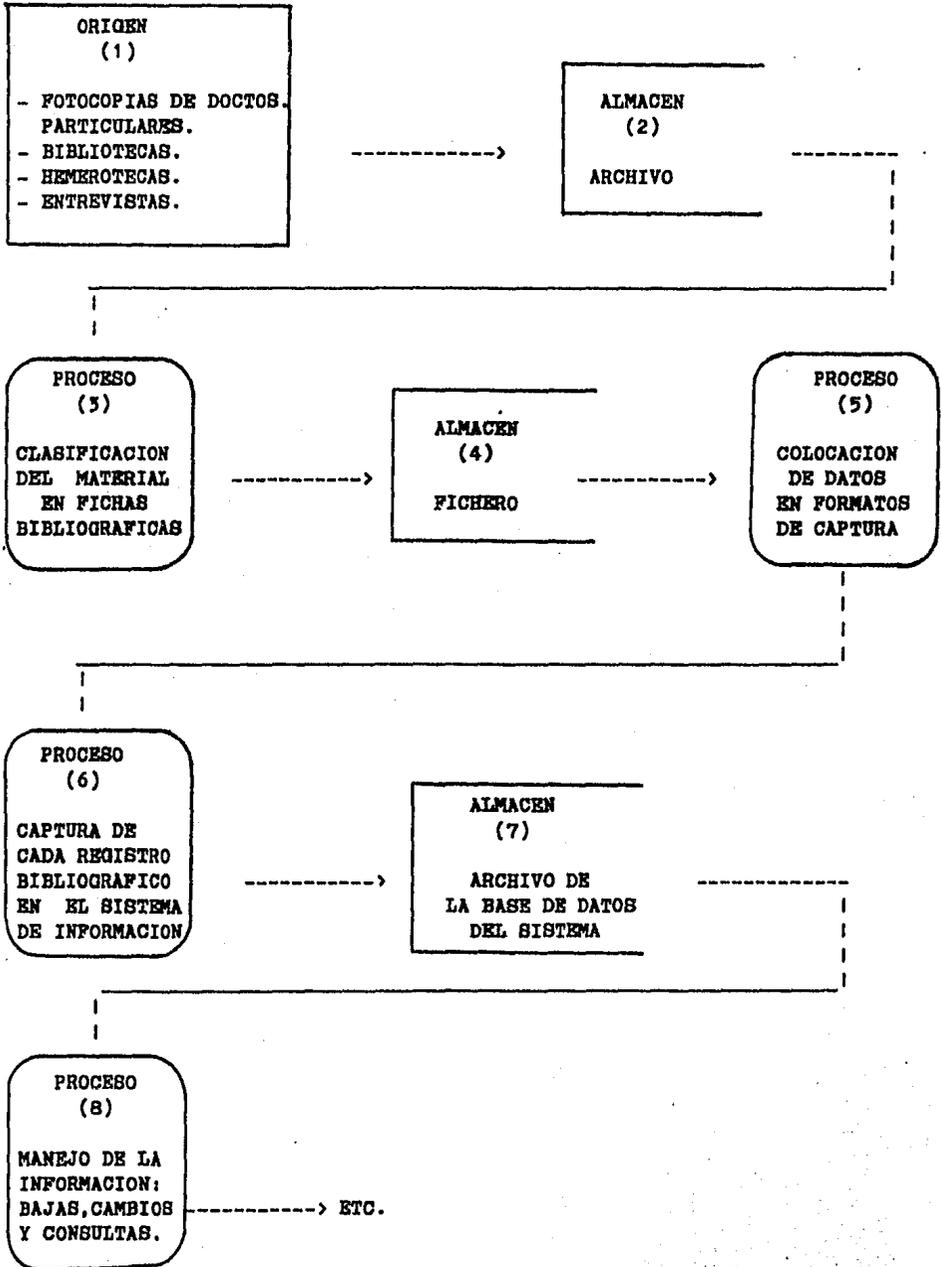
EJEMPLO II.1

El ejemplo forma parte de un sistema que tiene por objeto el mantenimiento de información bibliográfica empleando una base de datos. Antes de la captura física de los datos, es decir, antes de que los datos sean capturados en el computador se pretende realizar un procedimiento para su obtención y preparación. A lo largo de este procedimiento los datos fluyen de un lado a otro, dicho flujo se representa en el diagrama que se observa en la siguiente página, el cual explicamos a continuación:

- (1) ORIGEN: La información bibliográfica se obtiene de diversas fuentes.
- (2) ALMACEN: La información obtenida se almacena en un archivero.
- (3) PROCESO: La información se clasifica en fichas bibliográficas.
- (4) ALMACEN: Cada ficha se coloca en un fichero especial con la finalidad de tener la información bien ordenada y clasificada.
- (5) PROCESO: Cada ficha se pasa a un formato de captura especial similar a la pantalla de captura del sistema con la finalidad de evitar que el capturista se equivoque al teclear cada registro y lograr así mayor confiabilidad en la información. El formato de captura especial es simplemente una hoja de papel que contiene todos los campos que conformarán el registro bibliográfico.
- (6) PROCESO: Captura de los registros bibliográficos.
- (7) ALMACEN: El registro capturado pasa a ser almacenado a la base de datos del sistema.
- (8) PROCESO: Se da mantenimiento a la información capturada: bajas, modificaciones o consultas.

EJEMPLO II.1

FLUJO DE DATOS DE UN SISTEMA DE INFORMACION BIBLIOGRAFICA



Si suponemos que el diagrama anterior refleja el flujo actual de los datos, entonces, lo que el analista debe hacer es estudiarlo y tratar de optimizar dicho flujo generando uno nuevo, más eficiente y óptimo.

f.2) Diccionario de Datos.-

- Descripción General.- Un diccionario de datos es una colección de datos hecha por escrito. Tiene por objeto proveer información acerca de la definición, estructura y uso de cada dato. Un dato es una unidad de información que no puede ser descompuesta, es conocida también como "Campo".

- Información que debe contener un Diccionario de Datos.- La mínima información que debe contener el diccionario de datos respecto de cada campo es la siguiente:

- >> Nombre del Campo o Dato.
- >> Descripción.
- >> Formato.
- >> Fuente.
- >> Uso.

En el mercado existe software para controlar el manejo de un diccionario de datos, pero como no todos tienen acceso a dicho software puede hacerse una simulación valiéndose de una especie de fichero en el cual cada tarjeta contenga la descripción de un campo tal y como se muestra en el cuadro II.6.

CUADRO II.6

TARJETA QUE MUESTRA INFORMACION DE UN DATO

- | |
|--|
| - NOMBRE: [Nombre o clave de dato] |
| - DESCRIPCION BREVE: [Que es y su uso] |
| - DESCRIPCION GENERAL: [En donde se obtiene, si es o no producto de un proceso. Describir el dato] |
| - FORMATOS: [Tipo de dato: numérico o alfanumérico, longitud, máximos valores, rango, etc.] |
| - COLOCACION: [Documentos en los que se encuentran] |

Otra información adicional podrá anotarse al reverso de la tarjeta.

La razón de usar tarjetas en lugar de listas o tablas es la facilidad que se tiene al manejar cada ficha de manera independiente, sin necesidad de modificar y volver a escribir toda la lista o la tabla cuando sólo un dato o unos cuantos se están insertando o modificando. Por ello las tarjetas son fáciles de mantenerse actualizadas y por lo tanto son más prácticas y manejables.

Si se desea que el diccionario de datos quede más completo, el conjunto de información que se incluiría para cada dato sería la siguiente:

- >> Información General: Nombre, sinónimos, descripción.
- >> Formatos: Tipo de dato, longitud, tipo de unidades en las que se maneja.
- >> Características de Uso: Rango de valores que puede tomar, frecuencia en su uso, si es de entrada, de salida o local, valores condicionales.
- >> Información de Control: Fuente, fecha de origen, programas en los cuales es usado, autorizaciones para su cambio.
- >> Información de Grupo: Estructura que le da origen, estructuras secundarias, estructuras en las que se repite, localización física: registro, archivo o base de datos en la que se encuentra.

Será útil establecer reglas convencionales para nombrar cada campo o dato. Sucede con frecuencia que un mismo dato es denominado con distintos nombres, tales nombres o sinónimos podrán incluirse en la tarjeta para que así pueda haber una mayor identificación del mismo.

El tener un diccionario de datos "simulado", es decir que se lleve en forma manual y no en computadora, da lugar a ciertas desventajas:

- >> Si el sistema es grande y requiere de grandes canti-

dades de nombres y características de datos para funcionar, el diccionario de datos obviamente crecerá y las posibilidades de manejarlo eficientemente serán pocas.

>> Por lo anterior, en caso de que tal diccionario contenga demasiadas descripciones de datos éstas no podrán ser cruzadas fácilmente en caso de que se desee conocer alguna información específica de algún dato; por ejemplo: que programas emplearán tal dato, cuantos datos son empleados en "X" programa(s), etc. Este tipo de búsquedas sólo se podrán lograr en un computador de una manera rápida y eficiente por medio del cruce de información.

Se puede concluir que el diccionario de datos llevado en forma manual resulta eficiente en ciertos casos y en otros no lo es del todo. Esto dependerá del tipo y magnitud del sistema y del criterio del analista para decidir en el momento apropiado, si se desarrolla o no software para controlar el diccionario de datos por computador. Este tipo de software es usado en los Estados Unidos y probablemente en otros países. Se desconoce si en México existe.

- El Diccionario de Datos por Computador.- En los Estados Unidos existe un número considerable de paquetes comerciales para manejar diccionarios de datos. Algunos de ellos están asociados con un sistema de administración de base de datos particular, es decir, son más específicos. Otros son más generales y muchos ofrecen opciones para ligarse con una variedad de sistemas de administración de base de datos.

Las facilidades que deben esperarse al obtener un paquete de diccionario de datos son considerables. Cuando se crea un centro de cómputo muy grande al servicio de una institución y/o empresa resulta eficiente tener un sistema software que permita guardar nuevas descripciones de datos y actualizar o consultar éstas, por medio de su despliegue en pantalla. Lo anterior resulta de gran ayuda tanto a analistas, programadores y técnicos que deseen información de una manera rápida y práctica referente a los datos.

Sistemas más sofisticados de diccionarios de datos son diseñados para generar un código fuente con aplicación a pro-

gramas, es decir que haya una comunicación interna entre el programa y el diccionario de datos que permita verificar la existencia de cada dato, su nombre y características, si el dato es nuevo, éste puede ser sumado a la base de datos como una parte del proceso de verificación. En fin, pueden existir muchas opciones en software.

En cuanto a las salidas que genera el software comercial al que nos referimos, éstas están frecuentemente diseñadas de manera similar a los registros de la DATA DIVISION en COBOL o a las declaraciones dadas en el lenguaje PL/1 en cuanto a los datos.

Finalmente existe la posibilidad de usar información del diccionario de datos para generar listados que nos informen de lo que se desee acerca de los datos. Por ejemplo, generar un listado de todos los programas que usen en su ejecución "x" dato(s).

Consideramos que en México resulta aún sofisticado el llevar el control de un diccionario de datos por computador, empleando software comercial o desarrollado de manera particular.

Quizá muchas personas que se desenvuelven en el área de sistemas no lo conozcan y/o tal vez no manejen el concepto de diccionario de datos en la manera en la que aquí se ha tratado.

- Su Utilidad.- Independientemente de que el diccionario de datos sea llevado en forma manual o electrónica se desea exponer las razones fundamentales por las cuales es conveniente emplear tal documentación. Estas razones son las siguientes:

- >> Durante la etapa de análisis el diccionario de datos ayuda al analista a organizar la información acerca de los datos, así como a evitar redundancias al respecto. Pueden surgir problemas posteriores cuando por descuido un mismo dato o más son almacenados físicamente en varios lugares bajo diferentes formatos y con diferentes niveles de control.
- >> Frecuentemente un dato es definido o nombrado de ma-

nera diferente por varias personas. Esto crea confusiones y hará ver tanto al analista como al usuario de que al emplear distintos nombres probablemente no se esté hablando de la misma cosa. El diccionario de datos puede ayudar a mejorar la comunicación entre el analista y el usuario al establecer las definiciones de los datos lo más consistente posible.

- >> La utilidad del diccionario de datos también se deja ver en la etapa de programación cuando por ejemplo, se está programando de manera modular y cada programador está codificando un módulo. Por pequeño o grande que éste sea obviamente empleará varios nombres de datos los cuales también serán usados en el resto de los módulos. Si los programadores consultan el diccionario de datos para conocer la descripción y características de cada dato procesado en su módulo, se evitarán problemas en el momento de enlazar todos los módulos que componen el sistema ya que habrá una estandarización general en cuanto a los datos.
- >> A un nivel más general, el diccionario de datos es de gran ayuda cuando se requiere que varios sistemas sean ligados o relacionados entre sí y que tales sistemas manejen los mismos datos.
- >> En la etapa de implantación el diccionario de datos sirve de base a los programadores, los cuales trabajando sobre el sistema, pueden comparar las descripciones de datos.
- >> En la etapa de mantenimiento el diccionario de datos proporciona al o los programadores una idea clara de como un campo particular es usado, esto representa una información de valor cuando se requiera modificar el programa en un posible futuro.
- >> En general el diccionario de datos es usado como un documento de comunicación, como un medio de control de los datos y como una memoria auxiliar.

- Ejemplo.- Continuando con el ejemplo II.1 (diagrama de flujo de datos), se presentará la manera en la que quedaría el diccionario de datos si éste fuese llevado por tarjetas, en forma manual; se ejemplificará sólo con algunos campos (ver EJEMPLO II.2).

EJEMPLO II.2

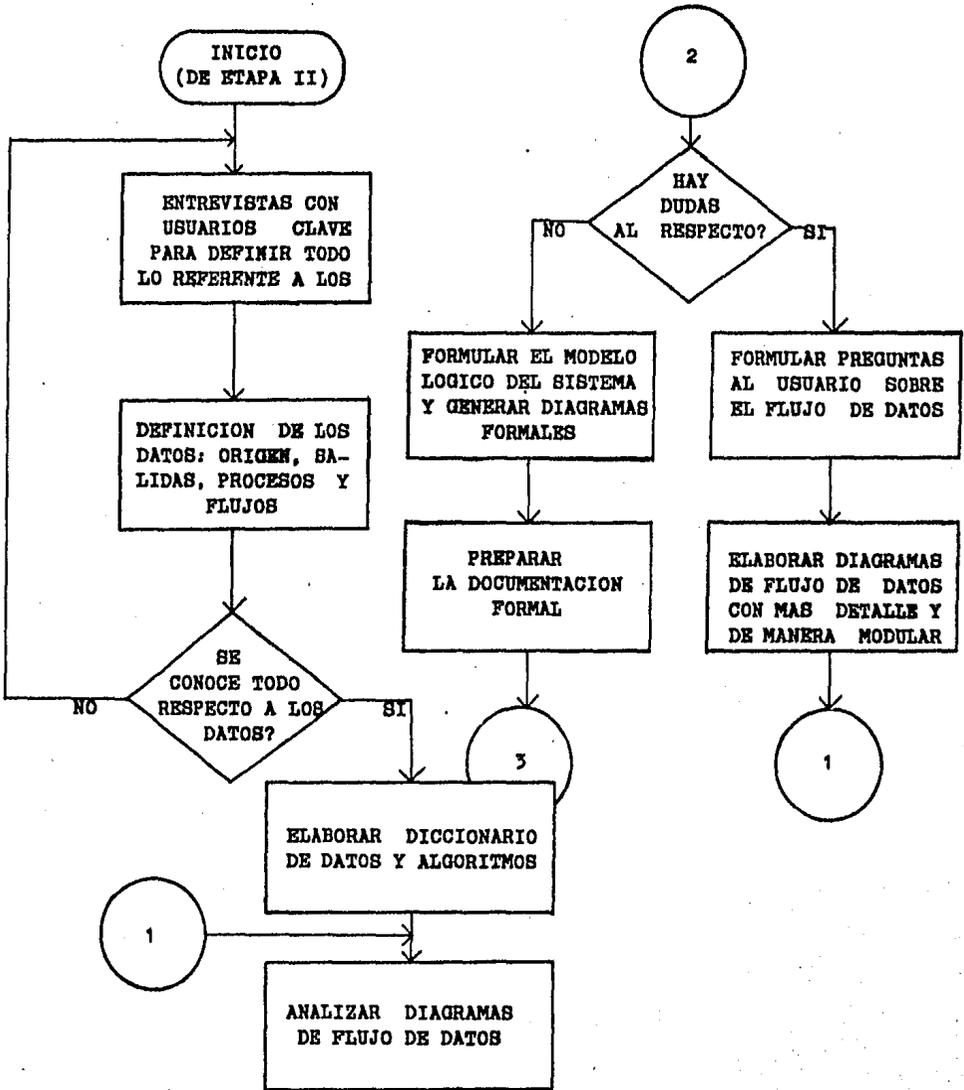
TARJETAS QUE FORMAN PARTE DE UN DICCIONARIO DE DATOS

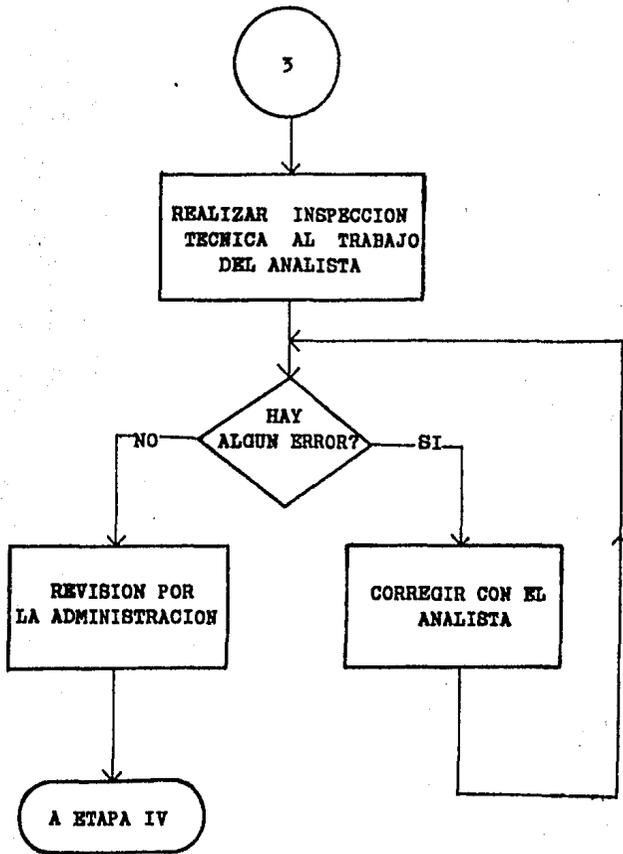
(3)
(2)
(1) NOMBRE: FBFOLIO. BREVE DESCRIPCION: NUMERO DE LA FICHA BIBLIOGRAFICA. DESCRIPCION GENERAL: DEBE GENERARSE INTERNAMENTE, POR TANTO SERA PRODUCTO DE UN PROCESO ESPECIAL. FORMATOS: DATO DE TIPO NUMERICO CUYA LONGITUD MAXIMA ES DE 6 CARACTERES. COLOCACION: HOJAS DE CAPTURA Y ARCHIVO DE FICHAS BIBLIOGRAFICAS.

(6)
(5)
(4) NOMBRE: FB TITULO. BREVE DESCRIPCION: TITULO DEL DOCUMENTO. DESCRIPCION GENERAL: NO SOLO SE REFIERE A TITULOS DE LIBROS SINO TAMBIEN A TITULOS DE PERIODICOS, REVISTAS, FOLLETOS, BOLETINES, FOTOGRAFIAS, PELICULAS, TESIS, TESINAS, TRANSPARENCIAS, VOLANTES, ETC. FORMATOS: ES UN DATO DE TIPO ALFANUMERICO CUYA LONGITUD MAXIMA ES DE 40 CARACTERES. COLOCACION: HOJAS DE CAPTURA Y ARCHIVO DE FICHAS BIBLIOGRAFICAS.

DIAGRAMA II.4

ETAPA: ANALISIS





3.4 DISEÑO.- Yourdon define la palabra "diseñar" como la actividad de planear la forma y método que nos llevarán a una solución, es decir, diseñar es el proceso o actividad por medio de la cual se determinan las características físicas que conformarán el sistema final (14).

La definición anterior aunque es específica del área de sistemas de información computarizados, coincide con la que da la Real Academia Española la cual define la palabra diseño como la "...descripción o bosquejo de alguna cosa..." (15).

El diseño de sistemas de información se refiere a las especificaciones técnicas que se llevarán a cabo para implantar un sistema y puede equipararse por ejemplo, con el diseño de un producto por un mercadólogo o bien con el diseño de una casa. En general para efectuar un diseño de cualquier tipo se requiere partir de lo abstracto y basarse en el medio circundante para así poder generar algo concreto, algo físico; ése algo quedará plasmado primeramente, en dibujos planos y/o descripciones redactadas por escrito (nivel lógico), las cuales pasarán a convertirse en entes físicos y tangibles (nivel físico).

Al principio de este capítulo se habló que el Diseño Estructurado y otras herramientas más son usadas dentro de la técnica estructurada para el desarrollo de sistemas. Este tipo de diseño se define como el arte de idear los componentes de un sistema así como la interrelación de éstos en la mejor forma posible; es el proceso de decidir la interconexión de componentes de tal forma que se pueda dar solución a problemas específicos (16).

En esta cuarta fase describiremos ampliamente como debe ser resuelto el problema. El objetivo es determinar como será implantado el sistema, pero de una manera general.

Se partirá de un modelo lógico para construir un modelo físico. Los diagramas de flujo de datos son el punto de partida de esta etapa. Lo único que el analista hará será plasmar gráficamente en el papel la configuración probable del sistema físico. Es importante que el analista no comience con la implantación del sistema sin antes haber terminado con el diseño; el uso de la metodología estructurada nos ayudará al respecto.

(14) YOURDON. OP.CIT. PAG. 8.

(15) REAL ACADEMIA ESPAÑOLA. DICCIONARIO DE LA LENGUA ESPAÑOLA PAG. 484.

(16) YOURDON. OP.CIT. PAG. 8.

La etapa del diseño principia con el desarrollo de un "X" número de alternativas físicas cada una de las cuales será representada en un diagrama de flujo que exprese una posible configuración del sistema en términos generales.

Por otro lado se listarán los componentes físicos que formarán al sistema en cada alternativa y se efectuará así mismo, un análisis del costo/beneficio para cada alternativa física.

Finalmente, el analista recomendará la alternativa que a su criterio sea la más conveniente y elaborará para ésta un programa de actividades que quedará representado en una red PERT. En forma opcional el analista puede elaborar, para el resto de las alternativas, un programa de actividades. De esta manera se tendrá otro parámetro: el tiempo, para tener más criterios de evaluación y elegir el camino más conveniente.

El diseño del sistema finaliza con una inspección hecha por un equipo de técnicos y con una revisión por la administración, al igual que como se hizo en la etapa de análisis.

De una manera ordenada, puede indicarse que el proceso que se sigue para elaborar el diseño de un sistema a un nivel general, es el siguiente:

Proceso del
Diseño de
Sistemas

- a) Búsqueda de Alternativas Físicas.
- b) Selección de Alternativas.
- c) Documentación de Cada Alternativa.
- d) Recomendación de un Curso de Acción.
- e) Inspección Técnica.
- f) Revisión por la Administración.

A continuación se explica cada una de las actividades.

a) Búsqueda de Alternativas Físicas. - El idear alternativas físicas requerirá del uso de los diagramas de flujo de datos como fuente de primera mano.

Para la generación de alternativas físicas podemos valernos de algunas herramientas útiles tales como la técnica lluvia de ideas y las listas de comprobación de las cuales ya se habló en la etapa del Estudio de Factibilidad (inciso e).

La lista de comprobación es una matriz o cuadro similar al usado en el Estudio de Factibilidad (ver CUADRO II.3 pag. 64) en donde se listan todas las alternativas y sus características.

Sin embargo, aunque se cuente con estas técnicas para buscar alternativas, se requiere un buen analista de sistemas que posea gran creatividad y experiencia para que pueda visualizar las alternativas posibles a realizarse.

Para efectuar la búsqueda de alternativas el analista, usando los diagramas de flujo de datos como referencia, podrá generar diferentes agrupaciones de procesos que sean posibles y eficientes, es decir, supondrá un "X" número de procesos que agrupará de diferentes maneras para encontrar la "mejor combinación", tomando como base los gráficos. Queremos decir que la mejor combinación será aquella que resulte más apropiada y eficiente al caso.

Así, cada proceso lógico del sistema puede agruparse, si el caso así lo requiere, con dos o más con la finalidad de hacer al sistema lo más eficiente y útil posible. Cuando hablamos de procesos nos referimos tanto a programas pequeños que cumplen un sólo objetivo (por ejemplo: emitir un reporte, ordenar un archivo, etc.), como a módulos o rutinas.

El objetivo en este punto es imaginar una serie de posibles opciones físicas sin detenernos a evaluar a cada una simplemente generarlas. Obsérvese que se trata ya de aspectos físicos del software que conformará al sistema.

Algunas de las alternativas planteadas presentarán el problema de decidir si determinado proceso o procesos se llevarán o no a cabo en forma manual o electrónica. Similarmente, se tratará de descubrir aquellos procesos que definitivamente

mente no podrán efectuarse de manera conjunta y/o simultánea, por ejemplo, consultar un registro y al mismo tiempo grabarlo. Por otro lado, será posible en algunas ocasiones, efectuar dos funciones al mismo tiempo dentro del sistema de información para así ahorrar tiempo y lograr optimizar el sistema.

b) Selección de Alternativas. - Se seleccionarán una o más alternativas de la lista de opciones factibles. La selección de éste conjunto de posibles estrategias de implantación dependerá del alcance y objetivos planteados al inicio, así como de los resultados de las etapas anteriores.

La identificación de alternativas posibles requerirá probablemente de entrevistas adicionales con los usuarios y la administración con la cual se decidirá por otra parte, si el software que requerirán las alternativas seleccionadas será comprado o creado internamente.

c) Documentación de cada Alternativa. - El documentar cada alternativa significa que para cada una de éstas el analista elaborará lo siguiente:

- c.1) Diagrama General de Bloque del Sistema.
- c.2) Lista de Componentes Físicos del Sistema.
- c.3) Análisis del Costo Beneficio
- c.4) Plan de Implantación.

Estos documentos representan los indicadores o criterios de éxito de esta etapa y en el momento en el que queden concluidos se identificará que la etapa de diseño ha sido llevada a cabo tal cual.

A continuación daremos una descripción de los puntos anteriores:

c.1) Diagramas de Bloques. - Los diagramas de bloques son una herramienta tradicional muy usada para describir un sistema físico. Cuando hablamos de un sistema físico nos referimos a un sistema de información cuyas características, diseñadas partiendo de la abstracción, han sido ya concretizadas en algo tangible.

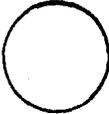
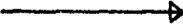
Los símbolos de un diagrama de bloques nos ayudan a re-

presentar cada componente del sistema: Programas, Procesos, Reportes, Archivos, etc. a un nivel general (caja negra). Un diagrama de bloques es un gráfico que muestra a nivel general sin detalles, un sistema físico y se diferencia de un diagrama de flujo común en que en el primero no se representan los detalles de programación sino sólo cuestiones generales, de ahí que los símbolos de ambos diagramas sean un tanto diferentes.

Los símbolos básicos de un diagrama de bloques se observan en el CUADRO II.8 que se muestra a continuación:

CUADRO II.8

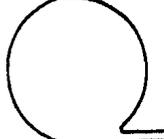
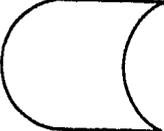
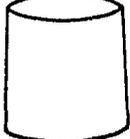
SIMBOLOGIA BASICA DE UN DIAGRAMA DE BLOQUES

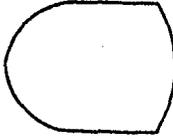
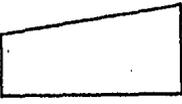
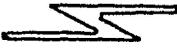
SIMBOLO	SIGNIFICADO
	- PROCESO O SUBPROCESO QUE MODIFICA EL VALOR O COLOCACION DE UN DATO. REPRESENTA UN PROGRAMA, RUTINA O PROCEDIMIENTO.
	- INDICA LA ENTRADA Y/O SALIDA DE INFORMACION
	CONECTOR: SENALA UNA SALIDA Y/O ENTRADA A OTRA PARTE DEL DIAGRAMA. GENERALMENTE ES USADO EN LA MISMA PAGINA.
	- INDICA EL FINAL DE UNA PAGINA Y LA CONTINUACION DEL DIAGRAMA EN LA SIGUIENTE.
	- ES USADO PARA LIGAR LOS SIMBOLOS CON EL FIN DE QUE EL DIAGRAMA TENGA UNA SECUENCIA LOGICA POR TANTO, DEFINE LA SECUENCIA Y LA DIRECCION DEL FLUJO.

Existen otros símbolos más específicos que sirven al analista para representar los medios por los cuales se llevará a cabo la entrada de información, su almacenamiento y/o su salida. Los símbolos secundarios se muestran en el CUADRO II.9.

CUADRO II.9

SIMBOLOS SECUNDARIOS EN LA DIAGRAMACION

SIMBOLO	SIGNIFICADO
	<p>- ENTRADA O SALIDA DE INFORMACION POR TARJETA PERFORADA. PUEDE REPRESENTAR TAMBIEN UN ARCHIVO DE DATOS EN TARJETAS.</p>
	<p>- ES USADO PARA SENALAR LA IMPRESION DE DOCUMENTOS: LISTADOS, REPORTES, ETC. PUEDE SENALAR TAMBIEN LA SALIDA DE DATOS POR TERMINAL.</p>
	<p>- ENTRADA O SALIDA DE INFORMACION POR CINTA MAGNETICA, O BIEN UN ARCHIVO ALMACENADO EN CINTA.</p>
	<p>- ALMACENAMIENTO EN LINEA, YA SEA EN DISCO, TAMBOR MAGNETICO, DISKETTE U OTRO RECURSO DE ALMACENAMIENTO EN MASA.</p>
	<p>- DISCO MAGNETICO. REPRESENTA LA E/S DE INFORMACION O BIEN UN ARCHIVO O UNA BASE DE DATOS ALMACENADA EN EL. COLOCADO HORIZONTALMENTE REPRESENTA UN TAMBOR MAGNETICO.</p>

SIMBOLO	SIGNIFICADO
	<p>- REPRESENTA UNA PANTALLA (CRT) O CUALQUIER OTRA UNIDAD DE DESPLIEGUE SIMILAR. PUEDE SER USADA TAMBIEN PARA IDENTIFICAR E/S.</p>
	<p>- ENTRADA DE INFORMACION EN FORMA MANUAL. PROCESO FUERA DE LINEA.</p>
	<p>- OPERACION MANUAL. REPRESENTA LA EJECUCION DE UN PROCESO EN FORMA MANUAL.</p>
	<p>- OPERACION AUXILIAR REALIZADA FUERA DE LINEA.</p>
	<p>- TRANSMISION DE LOS DATOS VIA TELECOMUNICACION.</p>

Los símbolos enumerados anteriormente, específicamente los básicos, definen a un nivel de "caja negra" cada uno de los componentes que conforman el sistema. Lo anterior es una característica que diferencia a tales símbolos de los que se usan para la construcción de diagramas de flujo, los cuales son usados en la etapa de diseño detallado para expresar básicamente la forma lógica de codificar cada uno de los programas.

- Reglas Generales Para la Construcción de un Diagrama de Bloques:

- >> Generalmente, la dirección del flujo de un diagrama de bloques es de arriba hacia abajo, sin embargo es posible realizar el flujo del diagrama de izquierda a derecha. Todo depende del técnico que los elabore.
- >> Los símbolos deben etiquetarse para fines de información, es decir, puede colocarse dentro de ellos un número o letra y en una página aparte hacer una explicación amplia, la cual muchas veces no cabe dentro de la figura. También en vez de un número o letra puede hacerse una explicación breve dentro de la figura, si así se desea.
- >> En algunos casos puede suceder que se requiera de un diagrama bastante grande para representar el sistema. Ante este caso, y a pesar de que existe un símbolo que funge como conector de página, si se llegan a tener muchos de éstos conectores en una misma página será difícil seguir el flujo del diagrama, por lo tanto, lo que puede hacerse en estos casos es dibujar el diagrama completo expresando en el únicamente las funciones principales. Después, cada función de éstas se dibujará en otro diagrama y así sucesivamente hasta llegar al nivel de detalle apropiado.

- Utilidad de los Diagramas de Bloque.-

- >> Por ser tan concretos (en ellos los procesos y archivos específicos son generalizados) proporcionan gran facilidad para poder visualizar la manera en la que será implantado el sistema, tal facilidad de comunicación permitirá al final de la fase de diseño responder al usuario y a la administración, con bases concretas, como se implantará el sistema.
- >> El diagrama identifica cada componente en forma concreta. Estos componentes serán analizados y descompuestos en el momento indicado hasta llegar a una aproximación de detalle adecuada. Por tanto, son un punto de partida para la próxima etapa: Diseño Detallado.

- >> Es una excelente herramienta de planeación ya que los costos estimados son más confiables cuando se basan en elementos físicos y concretos; además, el plan de implantación podrá ser proyectado y dividido en actividades concretas las cuales serán asignadas a varios grupos de la organización basándose tan sólo en un diagrama.
- >> El diagrama del sistema dará a cada grupo de trabajo independiente una idea de la forma en la que sus esfuerzos se reúnen y se disponen en un pequeño gráfico pudiendo conocer al mismo tiempo el objetivo al que estará encaminado su trabajo.
- >> En todo sistema representado por un diagrama de bloques es posible identificar cada proceso así como los dispositivos de entrada/salida que se le asocian.
- >> Finalmente, el diagrama de bloques es una memoria auxiliar que ayuda al analista a generar un trabajo seguro y concreto.

- Usos.-

- >> Para entender un sistema físico existente: El analista puede dibujar en un diagrama de bloques el sistema físico existente, de tal manera que pueda identificar la relación entre cada parte del mismo.
- >> Para representar la estructura física del sistema: Un diagrama de bloques también puede ser usado para representar el hardware en conjunto, es decir, la configuración física de todo un sistema, ya sea que éste exista o que sea un plano que explique como se proyectará conectar los equipos entre sí considerando el espacio: El computador con las unidades de cinta, la impresora, las unidades de disco, terminales, etc. En una palabra, puede usarse para planear la distribución y configuración del sistema hardware.
- >> Sistemas en diagramas de bloques y bases de datos: Generalmente los archivos físicos independientes tienden a desaparecer y a transformarse en un sistema de base de datos, ya que resulta más práctico hacer

acceso a una base de datos que a un número de archivos independientes para buscar información (aunque al respecto puede haber excepciones). Al ser mezclados la mayoría de los archivos en una base de datos, el diagrama de bloques del sistema es un tanto menos útil para hacer representaciones que cuando se usan los archivos tradicionales. Sin embargo, el diagrama de bloque del sistema puede proporcionar un esquema claro del uso de la base de datos mostrando los programas clave y el origen y destino de los datos que se relacionan con la base de datos.

- Ejemplificación de Un Diagrama de Bloques.- A continuación ofrecemos dos ejemplos de la utilización de los diagramas de bloques. En el EJEMPLO II.3 se cita el uso de la simbología de un diagrama de bloques para ilustrar la configuración clásica de un centro de cómputo.

Por su parte, el EJEMPLO II.4 ilustra el proceso general para el mantenimiento de un archivo de empleados. El sistema, aparte de permitir al usuario mantener el archivo (altas, bajas, cambios y consultas de registros), efectuará otras funciones tales como:

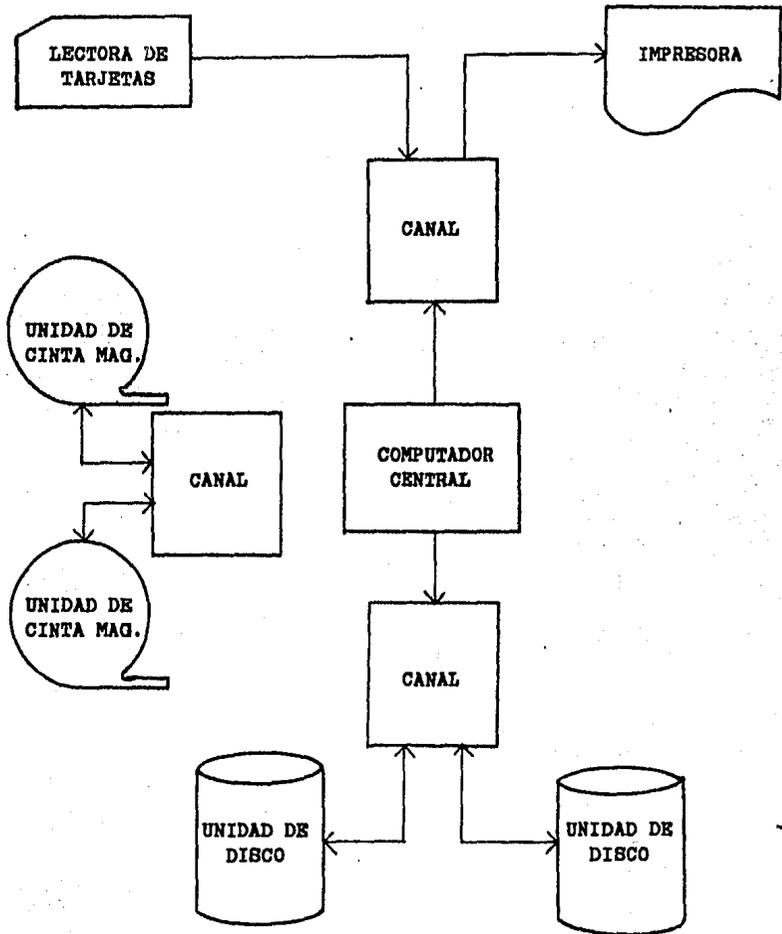
- >> Crear un Archivo Maestro Indexado.
- >> Crear un Archivo que contenga la Bitácora de los movimientos efectuados en una sesión.
- >> Emitir el Archivo de Empleados Actualizado.
- >> Calcular la Nómina y emitirla.
- >> Emitir un Reporte que contenga la Bitácora de la sesión correspondiente: No de Bajas realizadas, No de Cambios, No de Registros Capturados, No Total de Registros en el Archivo a la fecha, etc.

c.2) Lista de Componentes Físicos.- Los componentes físicos individuales deben ser identificados pero a un nivel general (caja negra). Por ejemplo si se trata de un archivo, éste sólo deberá nombrarse o representarse pero sin señalar su estructura, sus características de acceso, etc.

El contenido de los componentes será ideado hasta la etapa del diseño detallado. Los componentes físicos del sistema son:

EJEMPLO II.3

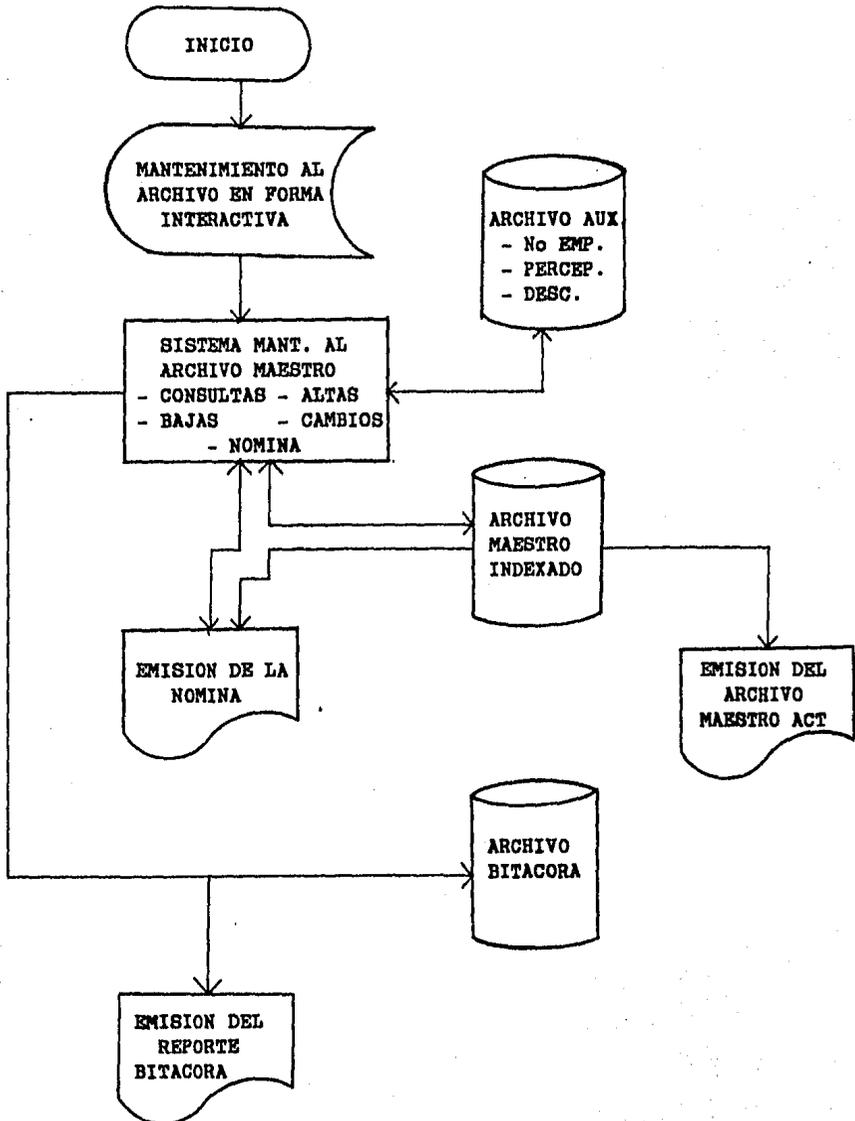
DIAGRAMA DE BLOQUES QUE MUESTRA LA CONFIGURACION TIPICA DE UN CENTRO DE COMPUTO (17)



(17) TOMADO DE: WILLIAM S. DAVIS. OP.CIT. PAG. 310.

EJEMPLO II.4

DIAGRAMA DE BLOQUES QUE MUESTRA UN SISTEMA DE MANTENIMIENTO A UN ARCHIVO MAESTRO DE EMPLEADOS



- Programas.
- Archivos.
- Reportes.
- Procedimientos Manuales e Instrucciones.

Como nota aparte se asume que el hardware ya existe.

Finalmente, gracias a la definición de los componentes físicos del sistema, el analista podrá comenzar a elaborar el plan de implantación para cada alternativa. Durante las etapas preliminares del diseño de procesos el analista estuvo trabajando con una imagen puramente lógica del sistema, ahora el tiempo y los costos estimados podrán basarse en componentes concretos y físicos, mejor que en funciones imaginarias.

c.3) Análisis de Costo Beneficio.- Este análisis probablemente enfrentará a la administración a decidir si se compra el software o se elabora internamente. Dentro de estas dos opciones para cada alternativa pueden analizarse aspectos tales como:

OPCION A
(COMPRAR)

- COSTO DEL SOFTWARE.
- COSTOS DE IMPLANTACION, OPERACION Y MANTENIMIENTO. (ACTUALIZACION DE VERSIONES).
- RETORNO DE LA INVERSION EN PORCENTAJE.

OPCION B
(HACER)

- COSTO DE DESARROLLO DEL SOFTWARE.
- COSTOS DE OPERACION Y MANTENIMIENTO.
- RETORNO DE LA INVERSION EN PORCENTAJE.

El resultado del análisis del costo beneficio de cada alternativa deberá compararse con el alcance dado en el estudio de factibilidad y con los costos del sistema calculados en esta etapa.

c.4) Plan de Implantación.- El plan de implantación debe elaborarse en una malla de actividades; esta malla de actividades puede realizarse para aquella alternativa elegida, no obstante, al considerar que el tiempo puede ser una variable importante que determine el elegir o no un curso de acción, será posible hacer un plan para cada alternativa y comparar cada uno.

Nótese que el programa de actividades es elaborado hasta esta etapa con mayor detalle que al principio ya que se conoce con especificidad las actividades particulares que van a realizarse, pues el modelo lógico planteado en las primeras etapas se está convirtiendo en un modelo tangible y físico. El plan que se elabora en la etapa de estudio de factibilidad es más general y plantea más incertidumbre ya que sólo es un simple cálculo.

d) Recomendación de un Curso de Acción.- Al elegir el analista un curso de acción descartará aquellas alternativas que no sean eficientes desde el punto de vista técnico, por ejemplo, aquellas cuyos procesos representen pérdidas de tiempo y dinero.

El analista sugerirá tanto a la administración como al usuario la alternativa que considere más apropiada tomando en cuenta los aspectos económicos, el objetivo y alcance inicial del proyecto.

Una vez que el curso de acción es recomendado se elaborará un plan de implantación detallado de tal alternativa. Este plan puede elaborarse en una malla de actividades que podrá incluir más o menos las siguientes actividades:

- Diseño de Archivos.
- Creación de Archivos.
- Diseño de Programas.
- Diseño de Reportes.
- Codificación.
- Capacitación.
- Pruebas.

Estas actividades se refieren a las etapas que faltan para concluir con el desarrollo del sistema: Diseño Detallado e Implantación.

La malla de actividades ilustra gráficamente el orden en el cual los eventos deben ocurrir así como la manera en la cual éstos se relacionan entre sí. Debe marcarse en tal malla cual es la ruta crítica, es decir cual es la combinación de actividades más óptima, considerando el tiempo y los costos.

Durante las etapas de diseño e implantación la malla de

actividades resultará una valiosa herramienta de control. Si un evento de la ruta crítica se retrasa en su realización todos los demás eventos que le siguen obviamente también se retrasarán.

e) Inspección Técnica. - Hasta este momento el analista ha trabajado sólo. Las próximas etapas requerirán de la presencia del programador/operador y de un considerable tiempo de computador. Por lo tanto, el costo del sistema comenzará a elevarse.

Antes de asignar fondos adicionales para realizar las etapas siguientes, la administración querrá estar segura de que el sistema propuesto realmente atiende las necesidades del usuario. De esta manera se plantea una inspección formal. Tanto el usuario como los representantes técnicos estarán en el equipo de inspección. Por medio del análisis que este equipo efectúe a los documentos, producto del diseño, se contestarán a las siguientes preguntas:

- El Sistema Efectúa el Trabajo que se Desea? (usuario)
- El Sistema Propuesto puede ser Implantado? (programador)
- Los Costos Estimados son Razonables?
- El Tiempo Projectado para Completar el Trabajo es Realista?

La recomendación del analista puede muchas veces estar basada en una predisposición o influencia subjetiva o bien en el mejor de los casos, en una cuidadosa consideración de los hechos.

Después de la inspección se decidirá si el analista ha realizado un buen trabajo técnico.

f) Revisión por la Administración. - La administración estudiará la recomendación dada por el analista y como producto de tal estudio o revisión la recomendación será aceptada o rechazada.

En caso de que la administración acepte la recomendación del analista sin ningún cambio, entonces se comenzarán a destinar los fondos necesarios para continuar con las últimas etapas del ciclo de vida estructurado.

Dados los resultados de la inspección la administración puede ignorar los detalles técnicos y concentrarse en los fondos financieros y los planes, puntos centrales de su interés.

En el CUADRO II.10 y en el DIAGRAMA II.5 pueden observarse los puntos más importantes de la etapa de diseño.

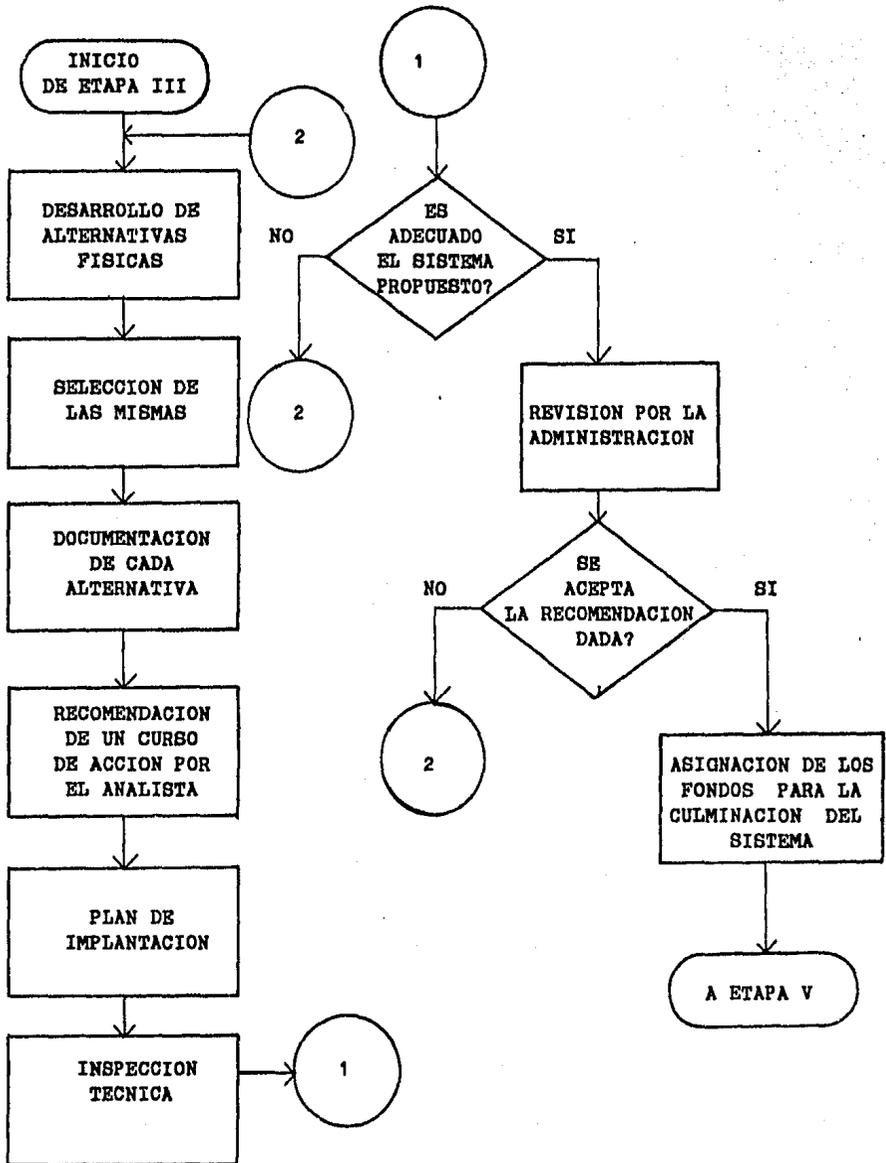
CUADRO II.10

ETAPA: DISEÑO

PREGUNTA A CONTESTAR	COMO SERA RESUELTO EL PROBLEMA?
A C T I V I D A D E S	- BUSQUEDA DE ALTERNATIVAS FISICAS. - SELECCION DE ALTERNATIVAS. - DOCUMENTACION DE CADA ALTERNATIVA. - RECOMENDACION DE UN CURSO DE ACCION - INSPECCION TECNICA. - REVISION POR LA ADMINISTRACION.
DOCUMENTOS GENERADOS	- DIAGRAMA DE BLOQUE DEL SISTEMA. - LISTA DE COMPONENTES FISICOS DEL SISTEMA. - ANALISIS DEL COSTO BENEFICIO. - PLAN DE IMPLANTACION.
CRITERIO DE EXITO	- GENERACION DE ALTERNATIVAS DE SOLUCION. - DIAGRAMA DE BLOQUE. - ANALISIS DEL COSTO/BENEFICIO DE CADA ALTERNATIVA. - PLAN DE IMPLANTACION.

DIAGRAMA II.5

ETAPA: DISEÑO



2.5 DISEÑO DETALLADO.- El objetivo de esta etapa es describir exactamente como será implantado el sistema por medio de la elaboración de una serie de documentos así como de un plan detallado de implantación.

En la etapa anterior (Diseño), los programas, archivos, procedimientos y hardware fueron identificados pero sólo a un nivel general; ahora, en esta etapa tales generalidades serán especificadas totalmente.

A grandes rasgos, las actividades a realizar serán entre otras, el diseño de archivos, el desarrollo de datos prueba, la elaboración de un modelo detallado del sistema, (por medio de descomposiciones funcionales) y el diseño de entradas, salidas y procesos usando la documentación HIPO. Todo esto se elaborará en base al diagrama de bloques general del sistema, realizado en la fase anterior.

Como es común, los documentos generados en la etapa anterior servirán de base para hacer el diseño detallado del sistema. Una vez realizado éste, se procederá a revisar toda la documentación que fue generada en esta etapa (la cual sustenta dicho diseño) y se comparará con la documentación que le precedió en las etapas anteriores: Diagramas de Flujo de Datos, Diccionario de Datos y Descripción de Algoritmos.

Después, se preparará un itinerario detallado que expresará la manera en la que será implantado el sistema. Dicho plan de implantación podrá desarrollarse, ya sea en Gráficas de Gantt o en una Malla de Actividades. Finalmente se procederá a efectuar una inspección técnica de los resultados de esta etapa.

En cuanto a la organización del equipo de sistemas para designar las funciones a cada miembro se puede agregar que la Malla de Actividades y el Diagrama de Bloques del Sistema ayudarán al analista a efectuar la división del trabajo en función de lo que se vaya a realizar y del como se vaya a realizar. Entonces, se puede decir que la Malla de Actividades del Proyecto y el Diagrama de Bloques General del Sistema apoyarán al analista en la coordinación de los esfuerzos ya que estos documentos expresan claramente las actividades concretas a realizar, la estructura física del sistema y los componentes físicos y sus interrelaciones, en forma respectiva.

El personal que participará necesariamente en el trabajo de esta fase será básicamente el diseñador de sistemas y los programadores. El primero, no necesariamente deberá ser otra persona diferente al analista de sistemas pues puede ser éste mismo el que se encargue de dirigir el diseño detallado; todo dependerá de la magnitud del proyecto y del número de elementos que formen parte del equipo de trabajo.

El diseñador de sistemas "... es aquella persona encargada de desarrollar el diseño estructural del sistema o programa maestro, definiendo los subsistemas apropiados, programas o módulos y su interconexión" (18).

Por su parte el programador "...es el encargado de precisar las especificaciones de los módulos individuales. Especificaciones que incluyen información sobre entradas, salidas e interfaces con otras partes del sistema y el algoritmo a través del cual el módulo hará su trabajo" (19).

Las actividades a realizar durante esta fase son las siguientes:

- a) Definición de los Datos.
- b) Diseño de Archivos.
- c) Preparación de Datos Prueba.
- d) Diseño de Salidas (Reportes, Formas y Pantallas).
- e) Diseño de Programas.
- f) Plan de Implantación.
- g) Inspección Técnica y Revisión por la Admon.
- h) Herramientas Empleadas.

a) Definición de los Datos para la Generación de Archivos. - La fuente principal para definir los datos que manejará el sistema es el diccionario de datos. Cada dato formará parte de uno o más archivos físicos y aparecerá ya sea en pantalla o bien en un reporte.

Cuando se habló del diccionario de datos en la etapa de análisis se mencionó que el definir un dato consiste en darle un nombre y en especificar su longitud y tipo. Para saber que datos o campos irán en que archivos, es necesario que el analista agrupe dichos datos de acuerdo a las relaciones que haya entre ellos para que así pueda formar estructuras.

(18) YOURDON. OP.CIT. PAG. 7.

(19) YOURDON. OP.CIT. PAG. 7.

Si el diccionario de datos se llevó en tarjetas, pueden hacerse agrupaciones de éstas en conjuntos separados (crear los registros para identificar que campos o datos serán almacenados en un determinado archivo). De esta manera se tendrá lista la estructura de un archivo con los datos que formarán parte de el.

Será conveniente realizar una reproducción de los documentos que contienen la definición de los datos con la finalidad de que los programadores tengan acceso a ella y no haya confusiones ni errores posteriores causados por una incongruente definición de los mismos.

b) Diseño de Archivos. - Un archivo es un conjunto de datos ordenados y clasificados de tal manera que cuando éstos sean usados, por un programa o sistema, puedan ser fácilmente procesados.

Los archivos son normalmente los primeros componentes del diagrama de bloques del sistema que deben ser diseñados pues ellos contendrán el elemento que ligará a todos los componentes principales del sistema: los datos.

El principal objetivo que debe perseguirse al diseñar un archivo es la minimización total de los costos del mismo. Aunque puede haber otros objetivos, quizá éste sea el más importante.

A grandes rasgos el diseño de archivos consiste en 3 puntos básicos a saber:

- Especificación del Contenido.
- Selección del Tipo de Organización.
- Estimación del Espacio Requerido.

- Especificación del Contenido.- Consiste en explicar el contenido de cada archivo del sistema y el describir la estructura de los datos (20) usando el diccionario de datos. El analista reunirá los campos que formarán parte de un registro lógico, de un determinado archivo, de tal forma que la estructura del registro quede constituida; así mismo deberá especificar el número aproximado de registros en cada archivo.

(20) UNA ESTRUCTURA DE DATOS DEFINE EL NUMERO DE CARACTERES POR REGISTRO Y EL NOMBRE Y NUMERO DE CAMPOS EN ESTE.

A continuación se presentan dos ejemplos de estructuras de datos:

EJEMPLO II.5

ESTRUCTURA DE LOS REGISTROS DE UN ARCHIVO BIBLIOGRAFICO

<u>NOMBRE</u>	<u>LONGITUD</u>	<u>FORMATO</u>	<u>TIPO</u>
FBAUTOR	30	X(30)	ALFABETICO
FBTITULO	40	X(40)	ALFABETICO
FBARTICULO	40	X(40)	ALFABETICO
FBLUGARED	18	X(1)	ALFABETICO
FBEDITORIA	15	X(15)	ALFABETICO
FBANIO	2	9(2)	NUMERICO
FBNUMERO	5	9(5)	NUMERICO
FBVOL	2	9(2)	NUMERICO
FBTOMO	2	9(2)	NUMERICO
FBPERIODI	2	9(2)	NUMERICO
FBTIPOMAT	2	9(2)	NUMERICO
FBFECHAED	8	X(8)	ALFANUMERICO
FBPAGINAS	9	X(9)	ALFANUMERICO

Esta estructura se refiere a un archivo que contendrá información bibliográfica y/o hemerográfica referente a un cierto tipo de tópicos universitarios. Las primeras dos letras significan "Ficha Bibliográfica" y el resto el nombre del campo. Lo anterior representa la estructura de un registro completo y un número determinado de estos registros formarán parte de un archivo.

EJEMPLO II.6

ESTRUCTURA DE LOS REGISTROS DE UN ARCHIVO DE EMPLEADOS

<u>NOMBRE</u>	<u>LONGITUD</u>	<u>FORMATO</u>	<u>TIPO</u>
NUM-EMP	5	9(5)	NUMERICO
NOMBRE	30	X(30)	ALFABETICO
RFC	10	X(10)	ALFANUMERICO
TURNO	1	9(1)	NUMERICO
SUELDO-DIARIO	8	9(8)	NUMERICO
HORAS-EXTRA	2	9(2)	NUMERICO

Un aspecto importante a tomar en cuenta es la creación de una nomenclatura para nombrar cada uno de los archivos que se van a emplear en el sistema. Lo anterior resulta útil cuando se van a manejar muchos archivos. Dicha nomenclatura debe ser clara y sencilla de tal manera que pueda reflejar el contenido de cada archivo para evitar confusiones. Por ejemplo si tenemos un sistema de ventas en el cual los archivos que se van a emplear son: Productos, Clientes y Distribuidoras; para nombrarlos se podrían usar los siguientes mnemotécnicos:

<u>ARCHIVO</u>	<u>NOMBRE</u>
PRODUCTOS	PROD/DAT
CLIENTES	CLIEN/DAT
DISTRIBUIDORAS	DISTRI/DAT

Y no como en algunas ocasiones se acostumbra nombrarlos de tal manera que no es posible adivinar cual será su contenido y uso:

GUS1/BAS
P1L2.TXT
JAF5/PAT

Es importante que desde ahora los archivos sean nombrados con una nomenclatura consistente y clara, antes de que sean creados físicamente.

- Selección del Tipo de Organización.- El analista debe estudiar y escoger cual es la manera más óptima (desde todos los puntos de vista) para organizar lógicamente los archivos.

Se considera que esta es una interrogante un tanto difícil ya que se debe estudiar detenidamente la forma en la que será más conveniente el acceso y uso de los datos tanto en eficiencia como en costos.

Los tipos de organización lógica que existen son:

- >> Secuencial.
- >> Directa o Random.
- >> Indexada.

>> Organización de Tipo Secuencial.- El diseño de la organización secuencial se basa en el concepto de sucesión de registros, es decir, solamente importa la colocación física de un registro detrás de otro.

Los registros son adheridos uno después de otro como en una cadena, es decir, se van almacenando en el orden en el que se van capturando o bien, pueden ser ordenados una vez capturados a través de una rutina de sorteo usando una llave que será uno de los campos del registro por el cual se desea ordenar el archivo.

Cuando los datos son procesados o consultados, éstos son leídos en la misma secuencia en la que se encuentran. La idea básica es que los registros del archivo son usados en el orden en el que se encuentran. Por ejemplo, si se desea acceder el registro No 10 el apuntador recorrerá del registro No 1 al No 9 hasta llegar al 10. Por lo tanto, la rapidez con la cual se encontrará el registro dependerá de su situación en el archivo.

>> Organización de Tipo Directa o Random.- "El diseño de la organización directa está basado en la definición de una dirección en la cual se almacena el registro y la cual constituye el factor de acceso" (21). Cuando se usa un archivo de acceso directo cada registro lógico tendrá una dirección física en el disco. Generalmente se creará un archivo que contendrá las direcciones de cada registro, este archivo normalmente es denominado como "Archivo de Apuntadores".

El medio de acceso será entonces una llave la cual no formará parte del registro sino que sólo estará asociada a éste, ya que contendrá la dirección física del mismo. Dada esta llave será posible almacenar o borrar un registro sin conocer siquiera su posición en el archivo. Por lo tanto este tipo de organización lógica permite tener acceso a cualquier registro sin necesidad de leer todo el archivo.

Generalmente este tipo de archivos se crean de igual forma que los archivos de tipo secuencial sólo que se le va incrementando el valor a la llave conforme se va grabando cada registro; dicho valor se asociará con su correspondiente registro.

>> Organización Indexada.- En este tipo de organización, la llave forma parte del registro, y tal registro resulta ser único. Un requisito fundamental es que al crear el archivo este debe estar ordenado por una llave.

La organización indexada "...permite acceder el archivo tanto de una manera secuencial como aleatoria a través de una llave primaria" (22)

>> Criterios de Selección.- El seleccionar un tipo de organización lógica dependerá de la aplicación, de las ventajas, de las desventajas y de las restricciones técnicas tanto del equipo como del software que se posean.

Se sabe perfectamente que los archivos de todo sistema generalmente son almacenados en memoria secundaria. La memoria secundaria se refiere a los dispositivos especiales de almacenamiento tales como cintas magnéticas, cassettes, disquettes, tambores magnéticos y discos magnéticos (23).

William S. Davis considera que el criterio más importante a considerar para seleccionar un tipo de organización lógica es el costo: "El primer criterio que el analista debe considerar al seleccionar el tipo de organización es aquel que permita el funcionamiento del sistema al menor costo posible" (24).

En base a tal criterio deben evaluarse los siguientes costos: costos de almacenamiento físico, costos de programación, costos de ejecución y costos de uso. Estos costos serán estudiados en cada alternativa (tipo de organización lógica) y lo importante de ello será el costo total.

Por ejemplo: Si en la ejecución, el tiempo de respuesta es un factor importante, el acceso secuencial no será el adecuado en cambio, resultará favorable el tipo de acceso directo o indexado, en el caso de que se esté manejando un archivo muy grande. Pero aquí lo que convendría evaluar también serían los costos de ambas alternativas.

Debido a que la elección del tipo de organización de un archivo, es tal vez la decisión más importante que el analista deberá tomar durante la etapa del diseño detallado, no se debe olvidar que dicha selección estará casi siempre dirigi-

(22) G. CLAYBROOK BILLY. FILE MANAGEMENT TECHNIQUES. PAG. 56.

(23) LAS CINTAS Y LOS CASSETTES SON LIMITADOS PARA ALMACENAMIENTOS DE TIPO SECUENCIAL. EL TAMBOR MAGNETICO ES DE GRAN RAPIDEZ Y BAJA CAPACIDAD, LOS DISCOS Y LOS DISKETTES SON LOS MAS USADOS.

(24) DAVIS. OP.CIT. PAG. 388.

da a la reducción de los costos de almacenamiento, de ejecución, de programación y de uso.

- Estimación del Espacio Requerido.- Una vez que ya están definidos los registros y se conoce el número estimado que habrá de ellos en cada archivo, entonces el analista podrá deducir el tamaño de los archivos.

Es importante que al realizar la estimación del espacio requerido se comparen las necesidades de almacenamiento que se requerirán con la capacidad física de la computadora. Este cálculo de los requerimientos de espacio será útil para determinar el tipo de hardware a emplear y sus costos.

William S. Davis considera al factor costo como algo muy importante que debe de aparecer en primer plano cuando se habla de la administración de la memoria. En este punto lo importante es el ahorro de la memoria ya que ello implica un ahorro en dinero. En efecto, quizá el costo si sea un factor importante pero quizá no en los casos en los que se decide sacrificar la eficiencia por el costo. De cualquier manera, a la administración siempre le interesará el factor costo.

En la práctica, el analista realiza la estimación del espacio requerido con tan sólo dos datos: el tamaño del registro lógico y el número de registros que se supone habrá en el archivo. Cuanto espacio en memoria secundaria deberá ser asignado para cada archivo?, esta es la interrogante a responderse.

Para la estimación del espacio requerido en memoria por cada archivo se emplean ciertas fórmulas. El analista debe estar familiarizado con las especificaciones y unidades de disco usadas en el centro de cómputo en el que trabaja. La capacidad de memoria en disco u otro dispositivo varía y por lo tanto deben identificarse las fórmulas propias a cada caso, ya que el calcular el espacio físico requerido en memoria secundaria es importante para determinar los costos de hardware.

c) Preparación de Datos Prueba. - Antes de que el analista comience con la generación de programas deberá preparar los datos experimentales o prueba.

Los datos prueba son una herramienta de gran utilidad que nos permitirán saber si el sistema completo o un programa del mismo es correcto antes de ponerlo formalmente en marcha.

La preparación de datos prueba es una medida preventiva que permite dilucidar si el sistema funciona tal y como se desea. Desafortunadamente, muchas veces los datos prueba son ignorados o no se les da la importancia que se merecen, la cual radica en que ayudan a minimizar los errores o circunstancias no consideradas dentro del sistema, siempre y cuando tales datos sean realistas, es decir, que consideren todas las situaciones posibles.

Al hablar de datos prueba realistas se desea expresar la conveniencia de usar como tales a aquellos datos que se están empleando en el sistema de información actual ya que serán datos verdaderos.

Al probar uno o más programas o rutinas, se podrá incluir, aparte de datos en condiciones normales, datos adicionales que representen condiciones extremas, o bien datos cuyos valores sean muy altos o muy bajos. El usuario podría sugerir algunas otras condiciones extremas, por ejemplo, incluir datos erróneos, poner una coma en medio de un campo numérico, etc.

Es importante que el usuario participe en la generación y definición de los datos prueba, ya que los sistemas son diseñados normalmente para los usuarios y no para los programadores o analistas. Al ser el usuario el cliente (en el caso de que se trate de un equipo de sistemas independiente) y al conocer los procedimientos de la empresa, él es el experto.

Los archivos ya deben haber sido diseñados para que una vez especificados los datos prueba, tales archivos puedan ser creados e inicializados con dichos datos. Entonces será cuando el analista podrá comenzar a diseñar el o los programas del sistema.

De cualquier manera, un buen sistema deberá ser capaz de manejar y contemplar cualquier situación que sea factible darse.

d) Diseño de Salidas. - Los medios básicos para la comunicación entre el computador y el usuario son las salidas que el sistema genera una vez que éste ha sido concluido, por tanto es menester que ahora el analista o diseñador del sistema dedique sus esfuerzos en la creación y diseño de tales salidas.

Los medios de comunicación entre el computador y el usuario son:

- Reportes Impresos.
- Pantallas.
- Formas de Recolección de Datos.

las cuales trataremos a continuación:

- Reportes Impresos.- Un reporte impreso es un medio de salida que provee grandes cantidades de datos en forma permanente, barata y, cuando es necesario, con copias, teniendo la desventaja de que la información que contiene es estática y puede rápidamente llegar a ser obsoleta.

Para diseñar como quedará formado un reporte es muy común el empleo de una hoja cuadrículada, tal y como la que se muestra en la siguiente página, la cual es formateada especialmente para designar en ella las líneas de impresión que formarán el reporte. En esta hoja cada cuadro representa simplemente una posición de impresión. Usándola, el analista describe en ella, línea por línea y posición por posición de impresión, la estructura del reporte.

La estructura de un reporte consta normalmente de lo siguiente:

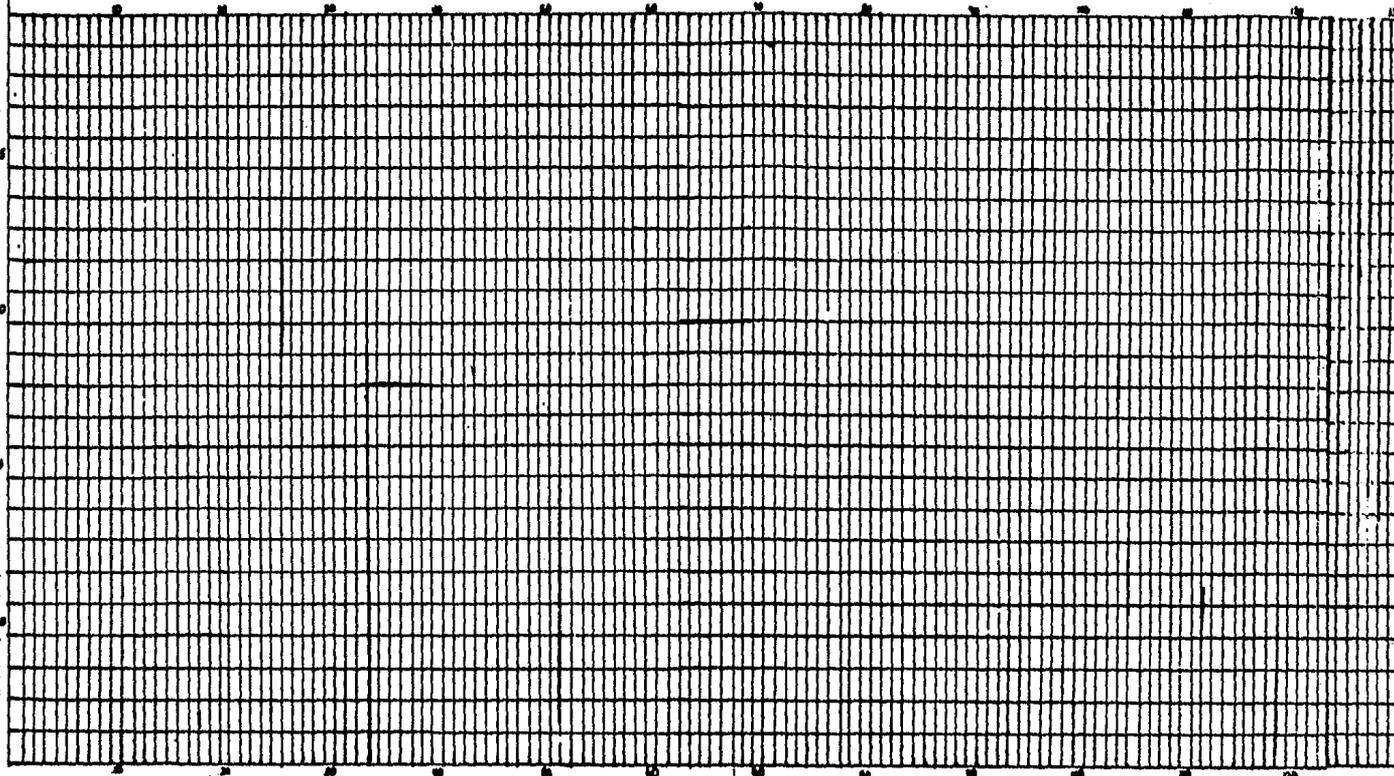
- >> Encabezados.
- >> Líneas de Detalle.
- >> Líneas de Pie de Página.

- >> Encabezados.- Son de tres tipos: Encabezados de Reporte que van al principio de todo reporte y generalmente identifican el nombre de la organización a la que se refieren; Encabezados de Página que identifican el nombre del reporte y se imprimen al principio de cada una de las páginas que se vayan a sacar, de-

HOJA PARA DISEÑO DE REPORTES

SISTEMA: _____ SUBSISTEMA: _____ PROCESO: _____

FUNCION: _____ FECHA: _____ DISEÑO: _____ REVISO: _____



pendiendo de la cantidad de datos, o después de un corte de control principal; y Encabezados de Columna que identifican el nombre de todos los campos listados en el reporte.

- >> Líneas de Detalle.- En ellas se escribe el valor de cada uno de los campos que aparecerán en el reporte. Son los datos los que son impresos aquí.
- >> Líneas de Pie de Página.- Se encargan de totalizar los campos numéricos de las líneas de detalle.

Como elementos necesarios, que también deben de ir en todo reporte están la fecha y el número de página.

El procedimiento que puede seguirse al diseñar un reporte es el siguiente:

- >> Listar todos los campos a incluir en el reporte.
- >> Agruparlos en clases de acuerdo a sus características
- >> Definir el número de posiciones de impresión requeridos por cada elemento escribiendo su máximo y su mínimo valor, incluyendo signos, y en caso de que se trate de campos no numéricos usar la longitud de campo.
- >> Observar si existe suficiente espacio para imprimir el reporte completo. Para ello deben sumarse todas las longitudes de encabezados y demás, para saber la longitud de cada renglón, recordando que debe haber como mínimo un espacio entre columna y columna. Por lo tanto, será necesario checar la suma total de columnas de impresión por cada renglón la cual no debe sobrepasar la longitud de caracteres por columna que permite una impresora, generalmente de 80, 120 y/o 132 caracteres.

El objetivo de todo esto es realizar un reporte práctico y fácil de leer que por lo mismo sea capaz de comunicar la información solicitada.

En el EJEMPLO II.7 se muestra el diseño del reporte empleando la misma estructura de datos del EJEMPLO II.6.

Así, el diseño preliminar del reporte en las hojas especiales ayudará a corroborar si éste es o no comprensivo para el usuario, por ello, antes de implantarlo formalmente deberá comentarse con él para que, en caso necesario, se le hagan las modificaciones pertinentes.

Por tanto, debe tomarse mucho en cuenta las necesidades de las personas a las cuales se dirigirá el reporte (demandantes de información o usuarios) para así decidir por ejemplo si se van a usar o no abreviaturas y nombres especiales o incluso formas de redacción específicas que probablemente no sean comprendidas por los solicitantes aunque para el analista si lo sean.

- Pantallas en Terminales o Microcomputadoras.- La pantalla puede usarse como un medio de entrada/salida. Una pantalla es como la ventana del sistema que permite al usuario comunicarse con éste. Si el sistema pretende ser de tipo amigable su diseño deberá girar en torno al concepto de listas o menús de opciones. La idea del menú es proporcionar al usuario una lista de opciones de las cuales el pueda elegir una, así como poder comunicarse con la computadora de una manera fácil.

Debido a lo expresado anteriormente, el analista o diseñador deberá dedicarse a pensar en el contenido de cada menú. Frecuentemente una pantalla se diseña en forma preliminar, al igual que un reporte, usando hojas cuadriculadas especiales como la que se observa en la siguiente página, las cuales permiten acomodar y centrar el contenido del menú (encabezados, nombres de datos, mensajes, etc.) para que así se pueda implantar posteriormente en una pantalla.

Para el diseño de una pantalla se debe conocer el número de caracteres que pueden desplegarse en una línea (No. de columnas) y el número posible de renglones disponibles. Estos dos datos: No. de columnas y No. de renglones, definirán el perímetro de la pantalla y las coordenadas renglón-columna de cada una de las posibles posiciones.

Es posible lograr posicionar el cursor en una determinada coordenada de la pantalla con el objeto de que a partir de ella se comience a teclear, ya sea una respuesta, un dato, o el valor de una opción. También puede colocarse un letrero en el lugar en el que se desee dentro de la pantalla. Sólo será

cuestión de revisar el manual del equipo que se trate en cuanto al uso de la terminal o pantalla y del teclado.

En el EJEMPLO II.8 se muestra el diseño de una pantalla en la cual se usan los posicionamientos del cursor.

Un proceso común para ejemplificar lo anterior sería el siguiente: Desplegar un letrero que pregunte por un dato específico, el cursor se colocará al extremo derecho de la pregunta para que el usuario alimente con tal dato al sistema y después, el valor será almacenado, tal y como se muestra en el EJEMPLO II.9.

Una pantalla que "conversa", por así decirlo, con el usuario a través de preguntas y respuestas, se dice que es una "máscara simulada" del sistema y del computador, que facilita el trabajo al usuario y que hace que el proceso interno sea invisible para él.

En terminales o pantallas de microcomputadoras más sofisticadas pueden diseñarse máscaras en colores diversos o bien con videos de intensidad reversa o media con lo cual se logran distinguir nombres, etiquetas o datos del resto del contenido de la pantalla.

También se habla ya de la existencia de pantallas sensibles al tacto. Esta nueva tecnología puede ser considerada por los diseñadores del sistema.

Entre las ventajas que podemos citar sobre el uso de una pantalla como medio de entrada/salida son:

- >> El despliegue de información por pantalla es un medio rápido de comunicación.
- >> Siempre presenta información actual, al día.
- >> Es un medio sencillo de comunicación para el usuario.

Por otro lado, el uso de estas "máscaras simuladas" presentan algunas desventajas no muy importantes como son:

- >> La información que presentan no es tan portátil como podría serlo un reporte.
- >> Si aparece en pantalla una salida o reporte, la información de éste irá desapareciendo línea por línea

DIRECCION DE COMPUTO PARA LA DOCENCIA
EJEMPLO II.8

PROGRAMA
UNIVERSITARIO DE
COMPUTO

FORMATO DE PANTALLA

SISTEMA Nómina SUBSISTEMA Mantenimiento Archivo PROCESO Altas

DIA	MEZ	AÑO
7	11	86

FUNCION Captura del Registro PANTALLA # 7 PREPARADO POR _____

	10	20	30	40	50	60	70
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							

"AROMATIZANTES QUIMICOS S.A."

ALTAS

1. N^o EMPLEADO: █

2. NOMBRE: █

3. R.F.C.: █

4. TURNO: █

5. SUELDO DIARIO: █

6. HORAS EXTRA: █

DIRECCION DE COMPUTO PARA LA DOCENCIA

EJEMPLO II.9

FORMATO DE PANTALLA

SISTEMA Nómina SUBSISTEMA Mantenimiento al Archivo PROCESO Bajas

DIA	MES	AÑO
7	II	86

FUNCION Captura de Llave para borrar Reg. PANTALLA # 9 PREPARADO POR _____

	10	20	30	40	50	60	70
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							

AROMATIZANTES QUIMICOS S.A.

BAJAS

DAME EL NUMERO DEL EMPLEADO PARA DAR DE BAJA SUS DATOS: 

al seguir entrando nuevas líneas de abajo hacia arriba. Generalmente esto ocurre cuando tal información ocupa un número mayor de renglones y columnas que los que tiene la pantalla. Lo anterior no permite que el reporte pueda visualizarse por entero teniendo que mandar a papel el reporte.

Como medida de alerta, se agrega que todo diseño de pantallas, a pesar de ser la actividad quizá más sencilla de esta etapa, debe estar cuidadosamente planeado, ya que si cada menú del sistema es diseñado con una mala calidad y en forma ambigua o poco clara probablemente el sistema sea subutilizado en un futuro.

- Formas de Recolección de Datos.- Una forma de este tipo es un documento usado para la captura de los datos en el computador. Los datos son recolectados en las formas y después introducidos tal cual al computador por cualquier medio.

En el EJEMPLO II.10 se muestra una forma de recolección de datos usada en el ya citado sistema de información bibliográfica.

Las ventajas de este tipo de formas son:

- >> Sus efectos pueden ahorrar costos ya que contribuyen en forma relativa a reducir el índice de errores de captura y de tiempo de máquina. Aunque toma tiempo el diseñar, reproducir y llenar una forma de entrada esta nos simplifica la colección, entrada y procesamiento de los datos. El tiempo de máquina para introducir los datos es menor ya que al ser preparados con anterioridad, hay menos posibilidades de cometer algún error ya que los datos están listos para ser capturados.
- >> Pueden usarse para cualquier caso si se diseñan de una manera estándar o general.

Su desventaja es que deben ser diseñadas y transcritas por personal específico, lo cual puede implicar un costo relativo en algunos casos.

EJEMPLO II.10

FORMA DE CAPTURA DE FICHAS BIBLIOGRAFICAS, HEMEROGRAFICAS Y DOCUMENTALES

NO FOLIO: _ _ _ _ _

AUTOR: _ _ _ _ _

TITULO: _ _ _ _ _

LUGAR DE EDICION: _ _ _ _ _ EDITORIAL: _ _ _ _ _

AÑO: _ _ EPOCA: _ _ _ _ _ NUMERO: _ _ _ _ _ VOL: _ _

PERIODICIDAD: _ _ _ _ _ TIPO DE MATERIAL: _ _

CONTENIDO: _ _ _ _ _ LENGUA: _ _ _ _ _

LUGAR DONDE SE ENCUENTRA: _ _ _ _ _ PAIS: _ _ PERIODO: _

FECHA DE EDICION: _ _ _ _ _ PAGINAS: _ _ _ _ _

ELABORO: _____ REVISO: _____ CAPTURO: _____

Dependiendo de la magnitud del caso y de la organización, las formas pueden imprimirse formalmente y ser de uso común en casos necesarios contratando a un diseñador gráfico para ello, o bien ser diseñadas y reproducidas por el propio analista.

Algunas formas llenas podrán ser leídas electrónicamente, otras, en cambio deberán ser introducidas al computador por capturistas. Antes, los datos eran tradicionalmente perforados en tarjetas y entregados al operador de la lectora, ahora, la opción más común es introducir los datos por medio de la terminal o pantalla.

En muchas ocasiones en la misma pantalla se despliega la forma de captura y el capturista simplemente llena los blancos con la información correspondiente a cada campo. En este caso, si se tienen formas impresas y al mismo tiempo una pantalla del sistema igual a las formas, lo que se estaría haciendo sería quizá una simple copia que podría representar una pérdida de tiempo, en otros casos la duplicidad anterior será necesaria siempre y cuando la preparación de los datos requiera de un largo proceso.

e) Diseño de Programas.— Quizá ésta sea una de las actividades más largas e importantes de la etapa del diseño detallado. El diseño de programas será tratado en este trabajo como una actividad que partirá de lo general a lo particular y consistirá de lo siguiente:

- Diseño del Programa Maestro del Sistema.
- Diseño de cada Módulo del Sistema.
- Revisión del Diseño de Programas.

- Diseño del Programa Maestro del Sistema.—

>> Concepto.— El programa maestro del sistema es aquel que se encarga de vigilar el acceso en línea del usuario al propio sistema, así como de controlar todas las subrutinas o módulos supeditados a él. Este programa puede solicitar al usuario, si así lo requiere la seguridad del sistema, una clave o identificación que garantice su derecho al acceso del mismo (programa objeto).

El programa maestro del sistema debe considerar que el usuario no tenga acceso a la estructura interna del sistema (programa fuente) ya que en algunos casos, si éste tiene conocimientos en la materia, podría modificar la estructura del mismo. El administrador o encargado del sistema si tendrá acceso a aquellos módulos que permitan modificar o arreglar el programa fuente que soporta al sistema, por razones de mantenimiento.

En el caso en el que dos tipos de personas vayan a manejar el sistema (usuario y encargado o administrador), dentro del mismo debe considerarse que éste sea capaz de identificar al tipo de persona para así darle oportunidad de acceso a determinadas funciones que desee manejar. Por ejemplo, en un sistema de mantenimiento de información bibliográfica y hemerográfica, incluyendo información de cintotecas, videotecas, discotecas, etc., cuyos usuarios serán investigadores, no sería lógico que tales usuarios tuvieran acceso a las rutinas de captura, de eliminación de registros o de cambios a los mismos, sino sólo a la rutina de consultas, que como investigadores es la que les interesa. En cambio, el administrador o encargado de tal sistema, si podrá tener acceso a las rutinas ya mencionadas por razones obvias.

Por lo tanto, el programa maestro del sistema deberá controlar, entre otras cosas, que el usuario tenga acceso solamente a aquellos módulos o procesos realizados exclusivamente para él, los cuales deben prever todas las posibles situaciones que se refieran al mal manejo del sistema, es decir, debe haber un control en el uso del mismo.

>> Elaboración del Diseño del Programa Maestro.- El diseño del programa maestro se inicia elaborando una lista de las funciones o proceso principales que ejecutará el sistema, basándose principalmente en el diagrama de bloques (25). Después, el siguiente paso será dibujar en una hoja los módulos de manera jerárquica, en donde cada uno de éstos representará una función principal de las listadas anteriormente. En pocas palabras, se elaborará una carta estructurada de tipo general (ver inciso h punto 1). Esta carta estructurada mostrará las funciones primarias tal y como se definieron en los diagramas de flujo de datos. De esta manera el control del sistema quedará representado en esta carta estructurada. En el EJEMPLO II.11 se muestra una carta estructurada general referente a

(25) EL DIAGRAMA DE BLOQUES DEL SISTEMA ES UNA REPRESENTACION FISICA DE LOS PROCESOS QUE SE EJECUTARAN EN EL SISTEMA, LOS CUALES PROVIENEN DEL ESTUDIO DETALLADO DE LOS DIAGRAMAS DE FLUJO DE DATOS. EL DIAGRAMA DE BLOQUE DEL SISTEMA CONTIENE LOS ARCHIVOS, LOS DISPOSITIVOS DE E/S ASI COMO LOS PROCESOS; TODO ELLO EN FORMA GRAFICA. COMO RECORDAREMOS, ESTE DOCUMENTO ES ELABORADO EN LA ETAPA DE DISEÑO.

un sistema de mantenimiento. El objetivo de este sistema es mantener actualizado un archivo de empleados y al mismo tiempo generar 3 reportes: la nómina, una bitácora, que contenga el número y tipo de movimientos realizados en una sesión, y el archivo actualizado. En la carta estructurada general se observan las funciones principales de tal sistema.

La carta estructurada inicial no es muy útil ya que sólo es un punto de partida bastante general. Lo importante será elaborar otra carta estructurada en donde se descompongan cada uno de los procesos generales para determinar sus funciones a detalle.

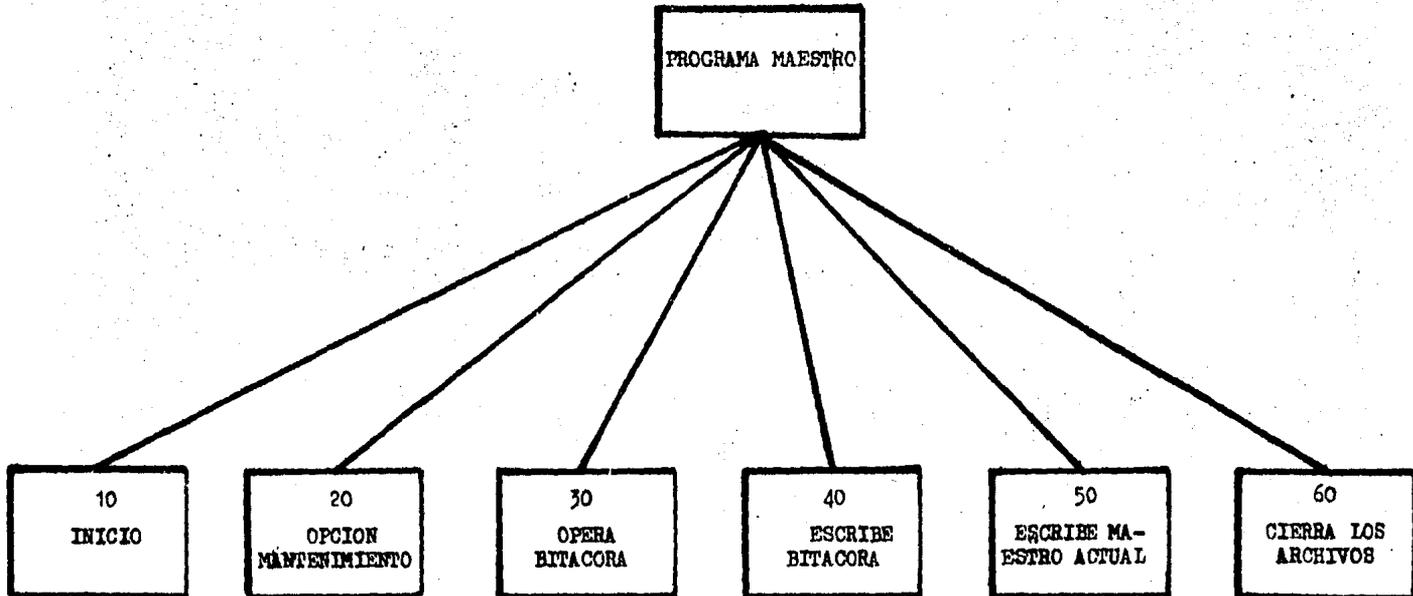
>> Recomendaciones.- Para diseñar el programa maestro el analista debe considerar las sugerencias y exigencias del usuario (aunque en algunas ocasiones éste no tiene noción alguna en la materia para poder sugerir) ya que cuando llega a probar un sistema ya existente, el espera encontrar reglas consistentes para hacer acceso a cualquier servicio por el computador y si sucede que cada servicio de estos requiere de una clase diferente de reglas de acceso, es decir, que sea complicado el manejo interactivo del sistema, entonces, el cliente o usuario encontrará a primera vista que el sistema es difícil de entenderse y manejarse.

El líder del desarrollo del sistema no debe permitir que los programadores realicen por separado el diseño de un programa o función ya que no es muy conveniente tener al final módulos independientes y separados entre los cuales no haya comunicación alguna, ya que lo que generalmente debe buscarse es la eficiencia que proporciona un sistema integral.

>> Filosofía de su Funcionamiento.- La estructura básica del programa maestro generalmente está conformada físicamente por el despliegue en pantalla de la presentación y descripción del sistema así como de una lista de opciones principales conocida como menú. El usuario seleccionará la opción por medio de un número o clave asignado a ella y oprimirá la tecla correspondiente (Return, New-Line, Enter, etc.) para que el sistema pueda identificar su elección. Este ciclo se repetirá tantas veces como el usuario lo requiera hasta llegar a las funciones más detalladas del sistema y que previamente fueron solicitadas por el usuario.

EJEMPLO II.11

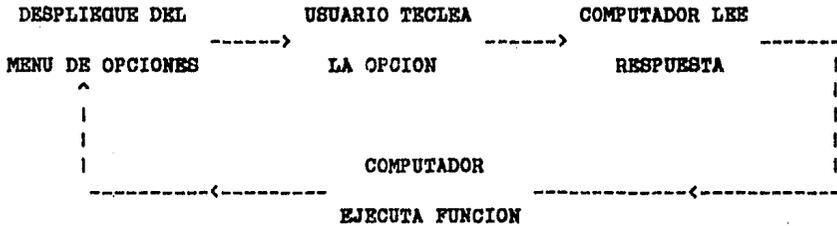
CARTA ESTRUCTURADA GENERAL DEL SISTEMA DE MANTENIMIENTO
AL ARCHIVO DE EMPLEADOS



EL ciclo de operación e interacción entre el usuario y el sistema será como sigue:

DIAGRAMA II.6

CICLO DE OPERACION USUARIO-SISTEMA



El proceso similar ocurre cuando el usuario desea regresar al menú de opciones principal.

En el EJEMPLO II.12 se observa una pantalla de presentación del sistema de información bibliográfica del que ya se ha hablado anteriormente.

Por su parte el EJEMPLO II.13 ilustra un menú de opciones principal bastante común, el cual se refiere al mantenimiento del archivo de empleados, ya citado con anterioridad.

- Diseño Detallado de cada Módulo del Sistema.- El diseño detallado del programa requerirá de la documentación generada en las etapas anteriores; entre esta documentación están principalmente, la descripción de algoritmos realizada en la etapa de análisis. También, por su parte las herramientas que se emplearán para diseñar los programas serán: Las cartas estructuradas y las cartas IPO (Entrada/Proceso/salida).

Una vez que las funciones del sistema han sido identificadas en la carta estructurada general, cada función representada en esta carta será desglosada poco a poco, es decir, se comenzará a realizar las llamadas "descomposiciones funcionales del sistema".

Una Descomposición Funcional es un proceso que consiste en desglosar "hacia abajo" cada nivel a un nivel de detalle más preciso. Las descomposiciones funcionales generarán numerosos módulos y cada uno de ellos representará un detalle lógico.

DIRECCION DE COMPUTO PARA LA DOCENCIA
EJEMPLO II.12

PROGRAMA
UNIVERSITARIO DE
COMPUTO (1983)

FORMATO DE PANTALLA

SISTEMA SIAPPEAL SUBSISTEMA PANTALLA PRESENTACION PROCESO _____

DIA	MESES	AÑO
7	II	86

FUNCION BIENVENIDA AL SISTEMA PANTALLA # 1 PREPARADO POR _____

	10	20	30	40	50	60	70	80
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								

FACULTAD DE FILOSOFIA Y LETRAS
Colegio de Pedagogia

¡¡BIENVENIDO!!

USTED ESTA EN EL SISTEMA DENOMINADO SIAPPEAL

Sistema de Información de las Alternativas
Pedagógicas y Prospectiva Educativa en
America Latina

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ACADEMICO
UNAM

DIRECCION DE COMPUTO PARA LA DOCENCIA
EJEMPLO II.13

PROGRAMA
UNIVERSITARIO DE
COMPUTO

FORMATO DE PANTALLA

SISTEMA Nómina SUBSISTEMA Mantenimiento Archivo PROCESO Menú Movimientos

DIA	MES	AÑO
7	II	86

FUNCION Saber tipo de mantenimiento PANTALLA # 2 PREPARADO POR _____

	10	20	30	40	50	60	70
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							

AROMATIZANTES QUIMICOS S.A.

MOVIMIENTOS AL ARCHIVO DE EMPLEADOS

1. CONSULTAS

2. ALTAS

3. BAJAS

4. CAMBIOS

5. EMISION DE NOMINA

6. FIN DE SESION

DAME TU OPCION: █

Deberá estudiarse que módulo o módulos podrán tener descomposiciones sucesivas y hasta que nivel de detalle será posible realizar tales descomposiciones. Hay módulos que por la función que realizan o por su misma sencillez no requieren de una descomposición exhaustiva. Lo más importante será tomar como guía el sentido común. Si la descomposición de un módulo favorece el seguimiento y comprensión de la lógica del proceso, entonces sí convendrá desglosarlo, en caso contrario no.

Podrán elaborarse tantas cartas estructuradas específicas como el nivel de detalle así lo requiera.

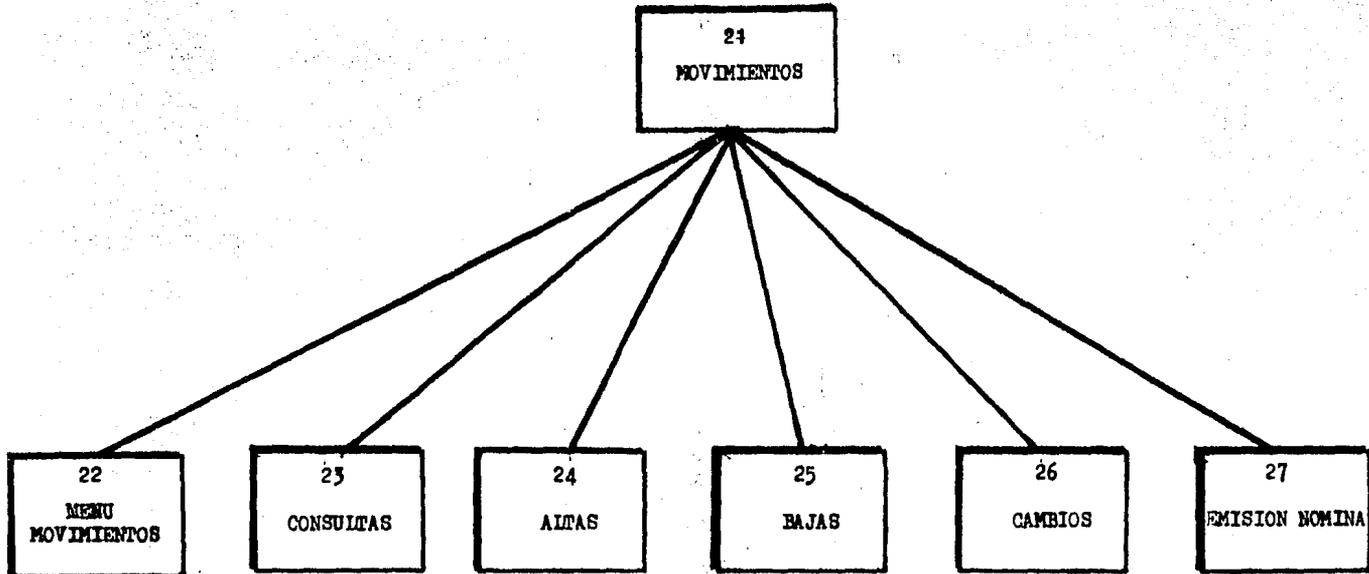
Para una correcta descomposición funcional pueden tomarse en cuenta las siguientes reglas:

- >> Principio de Cohesión.- Cada módulo de la carta estructurada deberá ejecutar una función completa, eficiente y sencilla, para que pueda haber una facilidad en el seguimiento lógico del programa. Los módulos con múltiples funciones pueden resultar confusos.
- >> Principio de Conexión.- Cada módulo debe ser independiente del resto del programa, esto se puede saber conociendo el número de entradas y salidas de datos que se efectuen en el módulo.
- >> No debe haber una modularización excesiva ni tampoco demasiadas llamadas, salidas o regresos a una determinada rutina o rutinas. Ya que esto puede provocar que la comprensión del proceso sea más difícil.
- >> Un módulo superior no debe controlar a demasiados módulos y/o niveles. El límite de control establecido es de 5 a 9 niveles.

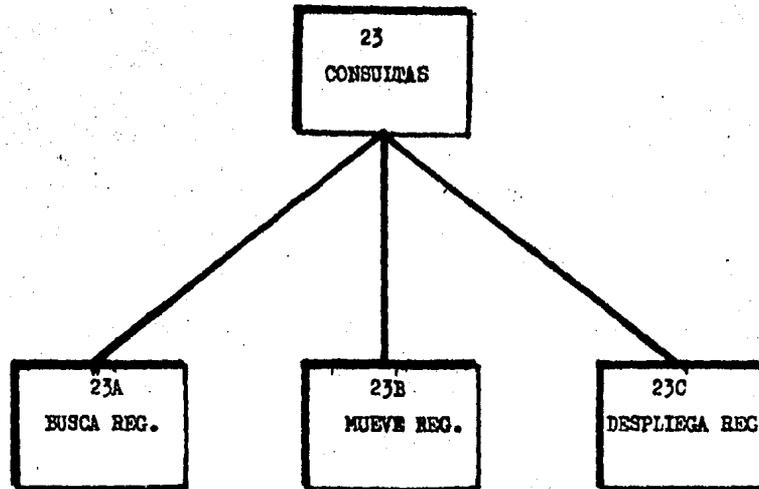
Como ilustración a lo anterior podemos observar en los EJEMPLOS II.14 y II.15 las cartas estructuradas específicas del Módulo 20-OPCION-MANTENIMIENTO y del Módulo 23-CONSULTAS del sistema de mantenimiento del archivo de empleados, producto de las descomposiciones funcionales.

Cuando la o las cartas estructuradas específicas han sido totalmente terminadas, podremos hacer uso de las cartas IPO (ver inciso h punto 1) para describir a detalle cada mo-

EJEMPLO II.14
SISTEMA DE MANTENIMIENTO AL ARCHIVO DE EMPLEADOS
MODULO 21-MOVIMIENTOS



EJEMPLO II.15
CARTA ESTRUCTURADA ESPECIFICA DEL SISTEMA DE MANTENIMIENTO
AL ARCHIVO DE EMPLEADOS
MODULO 23-CONSULTAS



dulo usando, para el diseño del proceso, la herramienta que más convenga al caso: Inglés Estructurado, Pseudocódigo, Diagramas de Flujo, Arbol de Decisiones o Tablas de Decisiones (ver inciso h puntos 2 a 6).

Para ejemplificar como se realiza una carta IPO tomaremos como base el Módulo de Consultas. Las cartas IPO se muestran en el EJEMPLO II.16

- Revisión del Diseño de Programas.- La carta estructurada del sistema en general así como las cartas IPO de cada módulo, representarán el diseño detallado completo del programa maestro del sistema. Faltará ver si es o no un buen diseño y el único camino para saber lo anterior es revisarlo cuidadosamente. Ahora es el momento adecuado para corregir las deficiencias del diseño y no durante la programación.

Los documentos que nos ayudarán en la revisión del diseño de programas serán, la lista de funciones prototipo, el diagrama de bloques del sistema y el diagrama de flujo de datos.

El analista deberá considerar lo siguiente:

- >> Todas las funciones listadas se encuentran incluidas en la carta estructurada?. En caso negativo probablemente el analista ignoró alguna.
- >> Hay funciones identificadas en la carta estructurada que sólo lo estén en la lista mencionada?. En caso afirmativo, esas nuevas funciones serán de utilidad?.

Una vez que se está seguro de que todas las funciones necesarias han sido incluidas, se procederá a revisar otros aspectos: El flujo de datos, la ausencia o presencia de datos útiles o no necesarios y la independencia de los mismos.

Todo lo anterior puede estudiarse dibujando el nombre y el flujo de cada dato usando flechas que indiquen la dirección de cada uno sobre la carta estructurada del sistema. La mejor fuente de información para dibujar el flujo de los datos son las cartas IPO las cuales contienen la relación entre cada módulo por medio de los datos así como el nombre de los mismos. Lo importante es dibujar lo más rápido y eficiente

CARTA IPO (INPUT / PROCESS / OUTPUT)

SISTEMA: NOMINA PREPARADO POR: _____

MODULO: 23-CONSULTAS FECHA: 7/II/86

EJEMPLO II.16

LLAMADO O INVOCADO POR:

21-MOVIMIENTOS

LLAMA O INVOCA A:

23A-BUSCA-REGISTRO
23B-MUEVE-REGISTRO
23C-DESPLIEGA-REGISTRO

ENTRADAS:

-VALOR DE "RESPUESTA" SI SE DESEA O NO
HACER UNA CONSULTA
-NUM-EMP

SALIDAS:

REGISTRO COMPLETO A PANTALLA

PROCESO: PSEUDOCODIGO

```
DO WHILE 23-CONSULTAS UNTIL RESPUESTA="NO"  
  WRITE PANTALLA "AROMATIZANTES QUIMICOS S.A."  
  READ RESPUESTA  
  WRITE "DAME EL NUMERO DE EMPLEADO A CONSULTAR: "  
  READ NUM-EMP  
  EJECUTA 23A-BUSCA-REGISTRO  
  EJECUTA 23B-MUEVE-REGISTRO  
  EJECUTA 23C-DESPLIEGA-REGISTRO  
END WHILE
```

DATOS LOCALES:

NINGUNO

OBSERVACIONES:

RESPUESTA ES UNA VARIABLE INDEPENDIENTE CUYO VALOR PUEDE SER "SI" O "NO".

posible el flujo de los datos tratando de no perder tiempo en operaciones absurdas.

Con lo anterior se tendrá una carta estructurada con el nombre y el flujo de cada dato hacia los módulos correspondientes. Sobre esta carta el analista podrá trabajar en lo siguiente:

- >> Que datos deben fluir a que módulo?. Sólo aquellos que son esenciales para la función del mismo.
- >> Que datos deben salir de que módulo?. Sólo aquellos que requieran trasladarse a otro módulo y se mantendrán aquellos que así se requieran.

Esta revisión del flujo de los datos dentro del diseño del sistema es muy importante debido a las siguientes razones:

- >> Si un determinado dato está fluyendo erróneamente, o bien, su trayectoria por el proceso no es lógica, probablemente se originarán errores de cálculo que podrán ser transmitidos al resto del programa. Si el valor de un parámetro es mal calculado cualquier módulo que emplee dicho parámetro arrojará datos erróneos.
- >> Un flujo innecesario generará una complicación innecesaria que a su vez dará lugar a que el tiempo de codificación y depuración de errores sea largo.
- >> Puede darse el caso de que un dato que sea innecesario genere más datos inservibles para los demás módulos del programa.

Todo lo anterior puede ocurrir en cualquier caso ya que cada módulo estará siempre ligado con los demás por el propio flujo de los datos.

Algunas normas básicas referentes a todo lo anterior se listan en la siguiente página.

- >> Sólo los campos que son necesarios para un determinado módulo deberán ser los que fluyan a ese módulo.
- >> Un módulo con muchas entradas y salidas de parámetros estará rigidamente ligado o asociado con el resto del programa. Por lo tanto a mayor asociación o relación del módulo con el resto del programa habrá mayor dependencia. Lo mejor es que cada módulo sea lo más independiente posible de los demás.
- >> El control del proceso debe fluir sobre la carta estructurada de arriba hacia abajo.
- >> Un módulo de menor nivel nunca indica a su superior lo que debe hacer, ni tampoco un módulo a otro que sea del mismo nivel.

Finalmente, quizá no resulte muy eficiente, para objetivos de documentación el incluir el flujo de los datos en la carta estructurada ya que este flujo será difícil de leer, no obstante, la carta estructurada tiene la finalidad de ayudar al analista a verificar el diseño.

De cualquier manera es preciso que el flujo de los datos entre los módulos de la carta jerárquica sea claro y entendible para los programadores ya que éstos serán los encargados de crear la parte final del proyecto.

La carta estructurada no es muy útil como medio para documentar el sistema, es tan sólo una herramienta de trabajo.

f) Plan de Implantación.-- El paso final del diseño detallado es la preparación de un plan de implantación el cual deberá resolver cuestiones tales como:

- Personas que Codificarán los Programas.
- Orden en el que serán Codificados los Programas.
- Fecha de Terminación del Sistema, etc.

A lo largo del proceso estructurado de desarrollo de sistemas se ha hablado mucho de la realización de planes, por ejemplo, durante el estudio de factibilidad, los tiempos estimados fueron expresados en años, en cambio, en la etapa an-

terior (diseño), los tiempos fueron estimados en meses. Ahora, estos tiempos podrán ser estimados en días simplemente porque el desarrollo del sistema ha progresado y el analista conoce más acerca de él, de tal manera que es capaz de hacer estimaciones más precisas.

Las estimaciones de tiempo que se hagan aquí podrán ser comparadas con las anteriores con la finalidad de revisar si la fecha de cumplimiento esperada en las etapas anteriores es aún válida.

El analista deberá estimar el tiempo de programación para cada módulo, usando como posible parámetro las líneas de codificación. Cada módulo de la carta estructurada representa una rutina del programa y si el diseño del sistema ha sido manejado convenientemente, cada una de estas rutinas tendrá no más de una hoja de codificación, máximo entre 50 y 60 líneas por módulo. Así, las líneas de codificación pueden convertirse directamente en días de programación asignando un número determinado de líneas por día. Lo anterior podrá programarse tal y como se muestra en el EJEMPLO II.17.

En cuanto a la distribución de las funciones programadas, es conveniente que el programa maestro sea asignado al programador más experimentado, por lo demás, el analista o el jefe del proyecto deberá decidir la mejor manera de distribuir el trabajo de programación en función del número de programadores y de la magnitud del sistema.

En caso de que dicho trabajo de programación sea considerable, puede elaborarse una tabla como la que se muestra en el EJEMPLO II.18 en la que se detalle el trabajo asignado a cada programador dentro del sistema ya citado anteriormente: Sistema de Mantenimiento al Archivo de Empleados.

También puede emplearse una Gráfica de Gantt en la cual se clasifique el trabajo del programador. Esta gráfica se muestra en el EJEMPLO II.19.

Es seguro que la administración solicitará un plan concreto en el que se muestren fechas realistas y que en efecto sea controlado su cumplimiento.

EJEMPLO II.17

PLAN DE IMPLANTACION DEL SISTEMA DE MANTENIMIENTO AL ARCHIVO DE EMPLEADOS

MODULO	LINEAS DE CODIF. PROBABLES	DIAS DE PROGRAMACION PROBABLES
- PROGRAMA MAESTRO	10	2
- 10-INICIO	4	1
- 20-OPCION-MANTENIMIENTO	12	2
- 21-MOVIMIENTOS	20	5
- 22-MENU-MOVIMIENTOS	11	3
- 23-CONSULTAS	40	7
- 24-ALTAS	68	10
- 25-BAJAS	33	5
- 26-CAMBIOS	31	5
- 27-OPERA-NOMINA	54	7
- 30-OPERA-BITACORA	32	7
- 40-ESCRIBE-BITACORA	33	7
- 50-ESCRIBE-MAESTRO-ACTUALIZADO	32	8
TOTALES:	382	69

NOTA: Los nombres de los módulos pueden tomarse de las cartas estructuradas General y Específicas del sistema que se hayan elaborado con anterioridad. Los cálculos en este ejemplo son aproximados. Todo dependerá del lenguaje de programación que se emplee.

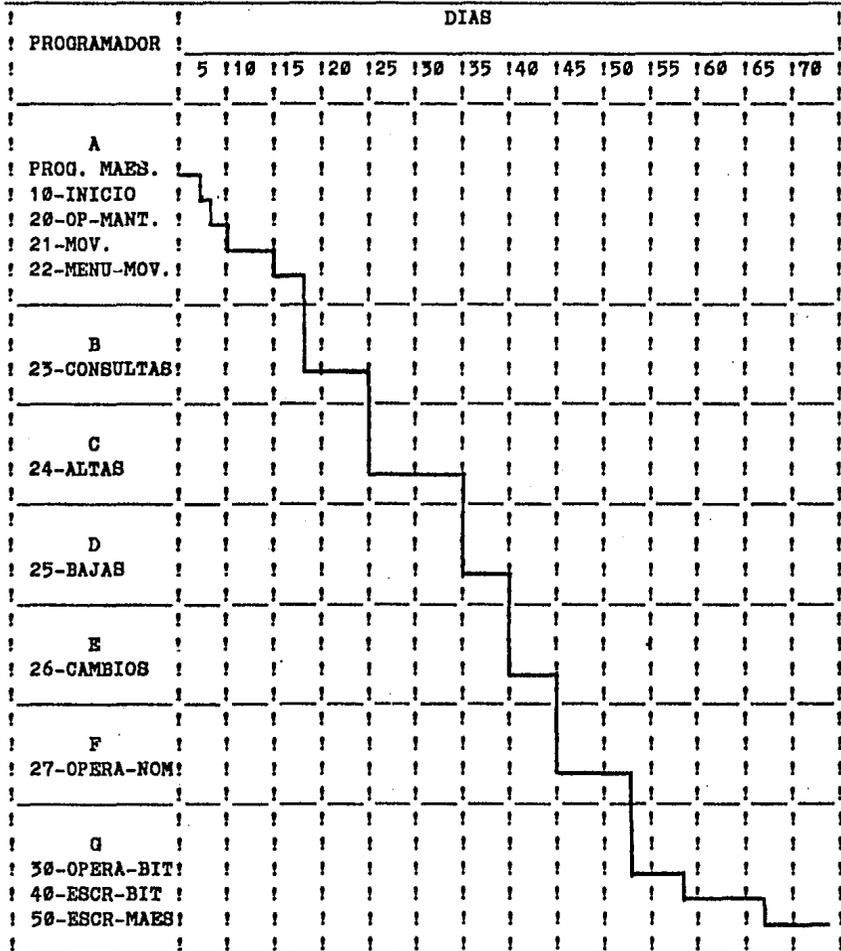
EJEMPLO II.18

TABLA DE DISTRIBUCION DEL TRABAJO DE PROGRAMACION

PROGRAMADOR	MODULOS A REALIZAR	TIEMPO
PROGRAMADOR A:	- PROGRAMA MAESTRO - 10-INICIO - 20-OPCION-MANTENIMIENTO - 21-MOVIMIENTOS - 22-MENU-MOVIMIENTOS	13 DIAS
PROGRAMADOR B:	- 23-CONSULTAS	7 DIAS
PROGRAMADOR C:	- 24-ALTAS	10 DIAS
PROGRAMADOR D:	- 25-BAJAS	5 DIAS
PROGRAMADOR E:	- 26-CAMBIOS	5 DIAS
PROGRAMADOR F:	- 27-OPERA-NOMINA	7 DIAS
PROGRAMADOR G:	- 30-OPERA-BITACORA - 40-ESCRIBE-BITACORA - 50-ESCRIBE-MAESTRO-ACTUALIZADO	22 DIAS

EJEMPLO II.19

GRAFICA DE GANTT QUE MUESTRA LA DISTRIBUCION Y EL TIEMPO DE DURACION DEL TRABAJO DE PROGRAMACION



NOTA: El tiempo de realización del sistema no será necesariamente de 69 días ya que el trabajo de cada programador puede hacerse en forma simultánea cuando el caso así lo permita. El encargado del proyecto deberá encontrar la mejor manera de coordinar el tiempo y el trabajo.

Cuando la fecha de inicio queda establecida, los días de programación específicos podrán ser definidos en relación a dicha fecha tomando en cuenta los días laborables y no laborables.

Por otro lado, es posible, en el caso en el que se desee aventajar en tiempo, realizar algunas funciones en forma simultánea, pero sólo aquellas que lo permitan ya que algunas requerirán ser efectuadas una tras otra por la naturaleza misma del trabajo.

Como comentario final podemos agregar que cualquier herramienta para la planeación de la implantación del sistema es buena, todo dependerá del caso y del líder del proyecto. La administración como ciencia emplea muchas otras herramientas de planeación que las que aquí mencionamos.

g) Inspección y Revisión por la Administración.- Tal y como se ha hecho al final de cada etapa del proceso estructurado, toca ahora al grupo de expertos en programación (independientes del equipo de trabajo) realizar una inspección técnica de cada programa del sistema en forma separada. De esta manera, los técnicos, al estar fuera del equipo de trabajo podrán visualizar con mayor objetividad la lógica de cada programa.

Por su parte, la revisión por la administración se enfocará hacia la aprobación del plan de implantación y a la asignación del presupuesto necesario para cumplir con el plan y concluir el sistema.

h) Herramientas Empleadas.- Este último inciso de la etapa del diseño detallado pretende ser una especie de anexo en el que se puedan explicar, en forma aislada del proceso estructurado, las herramientas que el analista o diseñador puede usar principalmente, dentro del diseño de programas.

Se ha visto que en cada una de las etapas del proceso estructurado se generan o se usan uno o más documentos específicos. Dentro de la fase en la que ahora estamos, los documentos a generar y las herramientas a usar en forma opcional serán:

- Documentación HIPO (Hierarchy Input Proces Output).

- Inglés Estructurado.
- Pseudocódigo.
- Diagramas de Flujo.
- Tablas de Decisión.
- Arboles de Decisión.

Se desea aclarar antes que nada, que el proceso es diferente a lo que son las herramientas anteriores. El proceso es una filosofía o un punto de vista de como se puede resolver el problema, es una progresión metódica que va desde lo lógico hasta lo físico. La habilidad para seguir el flujo de este proceso y completarlo con la documentación adecuada al caso debe ser una destreza esencial del analista de sistemas.

"Las herramientas pueden cambiar pero la filosofía no. Si nosotros aprendemos solamente las herramientas nuestro conocimiento se volverá rápidamente obsoleto" (26).

Se procederá entonces a explicar cada una de las herramientas mencionadas:

- Documentación HIPO.- HIPO es una herramienta usada en el diseño detallado, dentro del diseño de programas, para describir cada uno de los procesos expuestos en un diagrama de bloques.

El objetivo de HIPO es documentar un plan que va a indicar la manera de realizar la implantación. El uso de esta herramienta implica la generación de dos tipos de documentos: la carta estructurada y la carta IPO (Input Proces Output).

>> Carta Estructurada.- Es usada para representar la estructura jerárquica del sistema por medio del uso de cuadros en forma similar a un organigrama. Yourdon la denomina como "Diagrama Estructurado" definido por él mismo como "... un modelo independiente que muestra las relaciones jerárquicas de los módulos dentro de un programa o sistema" (27).

Una carta estructurada puede ser de dos tipos: General y Detallada. La Carta Estructurada de Tipo General es la que describe gráficamente el programa maestro del sistema y sus módulos principales. Los módulos de esta carta son generales y ejecutarán únicamente funciones de control. El módulo que representa el programa maestro se encuentra a la cabeza de la

(26) DAVIS. OP.CIT. PAG. 18.

(27) YOURDON. OP.CIT. PAG. 64.

carta estructurada y será el que determinará el orden en el cual los módulos subordinados al mismo serán ejecutados.

Por su parte, La Carta Estructurada más Específica representa la ejecución de las funciones más detalladas. Una vez que se desarrolla la carta de tipo general se le toma como base para generar la o las carta(s) estructurada(s) de tipo específico. Cada nivel menor representará la ejecución de una sencilla función que estará controlada por el o los módulos superiores. Por lo tanto, el control del programa regresará hacia los módulos superiores y las tareas específicas serán ejecutadas por los más inferiores.

Si se observa lo anterior, nos damos cuenta de que hay una similitud con el proceso de división del trabajo y el principio de la jerarquización de los que tanto se habla en la administración científica, ya que mientras más se tratan los detalles, se va observando que cada módulo de la carta estructurada representa la ejecución de una tarea cada vez más sencilla y que además cada una de las tareas está supervisada y controlada por un módulo superior.

Cada uno de los niveles y módulos de la carta estructurada pueden numerarse. En el caso de los módulos, éstos pueden designarse con un nombre, tal nombre quedará constituido por un prefijo (número) y un mnemotécnico que identifique la función que realiza. De esta manera, cuando el nombre del módulo sea referenciado en el programa fuente, el prefijo revelará con toda claridad su posición en la carta.

Por ejemplo si tenemos el nombre 10-ACTUALIZA-INVENTARIO, éste nos indicará que está en el nivel 1 y por tanto representa un proceso de tipo general.

>> Carta IPO.- Una carta IPO muestra las entradas y salidas (flujos de información) y procesos ejecutados por una rutina o módulo y su relación con los módulos superiores y/o inferiores a los que está ligado. En una carta IPO se describen cada uno de los módulos que se representan tanto en la carta estructurada general como en la detallada.

Para cada módulo descrito en la carta estructurada se usa una carta IPO en la cual se mencionan las entradas y/o salidas a un determinado módulo y el proceso ejecutado por éste.

Los documentos fuente para elaborar las cartas IPO son las cartas estructuradas (para conocer la dependencia y orden de cada módulo respecto a los demás), el diccionario de datos (para conocer las entradas y salidas de datos que hay en cada módulo) y la descripción de algoritmos (para definir los procesos).

A continuación se presenta en la siguiente página un formato IPO (28), el cual podemos clasificarlo de la siguiente manera para explicarlo:

- >> Líneas Superiores: Identifican al sistema, el nombre del módulo que va a describirse, la fecha y el autor.
- >> "Llamado por y Llama a": En estos dos primeros cuadros se muestra como se relaciona el módulo en cuestión con los demás. Con estos datos se puede conocer de quien depende y/o a quien controla este módulo.
- >> "Entradas y Salidas": En estos dos cuadros se identifican los datos que entran y salen de este módulo para ser procesados.
- >> "Proceso": En este último rectángulo se describe el proceso que ejecutará el módulo usando la herramienta más adecuada al caso: Inglés Estructurado, Pseudocódigo, Diagramas de Flujo, Tablas de Decisión y Árboles de Decisión (1).
- >> Finalmente se enumeran hasta abajo los datos o parámetros locales (2) y se escriben algunas notas en el extremo inferior derecho en caso de ser necesario.

NOTAS:

(1) Las herramientas usadas para describir los procesos pueden combinarse entre sí o usarse en forma independiente según el caso y las necesidades. La idea es describir en la carta IPO, de una forma clara y concisa, el proceso ejecutado en el módulo, no debiendo ajustar la descripción de cada proceso a una forma particular de documentación forzosa.

(28) TOMADO DE: WILLIAM S. DAVIS. OP.CIT. PAG. 328.

CARTA IPO (INPUT / PROCESS / OUTPUT)

SISTEMA: _____ PREPARADO POR: _____

MODULO: _____ FECHA: _____

LLAMADO O INVOCADO POR:

LLAMA O INVOCA A:

ENTRADAS:

SALIDAS:

PROCESO:

DATOS LOCALES:

OBSERVACIONES:

(2) Los datos locales son datos usados solamente dentro de un módulo en particular. Estos datos internos pueden ser ignorados en cuanto a su valor ya que sólo se transforman dentro del módulo, son temporales, su valor se mantiene mientras se ejecuta este módulo y se obtienen los resultados parciales.

Los resultados intermedios serán irrelevantes para el resto de los demás módulos. En cambio, las entradas y salidas del módulo serán empleadas por muchos otros módulos y por tanto estos datos serán globales.

Un dato global tiene participación en varios módulos o en todo el sistema, en cambio, los datos locales son definidos, almacenados y usados exclusivamente en el módulo al que pertenecen.

Una carta IPO deberá describir plenamente cada rutina del sistema sin dejar duda alguna.

En muchas organizaciones el uso de hojas IPO ha sido estandarizado, en otras en donde se desarrollan sistemas, probablemente no se usen ni se conozcan. En los Estados Unidos se usa en algunas empresas un software especial para la documentación IPO. El analista puede llenar las hojas IPO desplegándolas en la terminal y almacenar esta información en disco. Las cartas pueden ser modificadas simplemente por extracción selectiva, cambio y reestablecimiento en el archivo. Además, nuevas cartas pueden ser creadas y adheridas al conjunto. También se realizan cruces de valores para obtener, a través del sistema, una carta IPO en específico. Una vez que las cartas han sido completadas podrán ser impresas y usadas como documentación tradicional.

Algunos propósitos de HIPO son:

- >> Los diseñadores pueden evaluar y reafinar el diseño corrigiendo imperfecciones antes de que la implantación sea llevada a cabo.
- >> El usuario y la administración pueden seguir fácilmente la estructura del sistema debido a la naturaleza gráfica de HIPO.

- >> Los programadores pueden usar las cartas estructuradas y IPO para codificar, mantener o modificar el o los programas.

- Inglés Estructurado.- Es usado para describir la secuencia de un proceso dentro de una carta IPO. Es muy limitado. En muchos casos puede asemejarse a un lenguaje de programación y por tanto será fácil de entender por cualquier programador.

Aunque hay muchas versiones del inglés estructurado y ninguna se aproxima a un estándar, algunas guías para su uso son:

- >> Se usan tres tipos básicos de instrucciones: Secuencia, Decisión y Repetición.
- >> Una instrucción en inglés estructurado debe estar formada por una oración imperativa y corta.
- >> Las instrucciones de secuencia están seguidas por el nombre de los campos asociados. Algunos ejemplos son: CALCULATE, MOVE, GET, READ, COMPUTE, etc.
- >> Es conveniente agrupar varias instrucciones en un bloque y asignar un nombre a este bloque considerándolo como una oración secuencial simple.
- >> Las instrucciones de decisión están basadas en el conocido IF-THEN-ELSE. El vocablo IF está siempre seguido por una condición; si la condición es verdadera las instrucciones que se ejecutarán serán las que contiene el THEN. ELSE es la opción que se sigue cuando la condición resulta ser falsa. El IF-THEN-ELSE es una instrucción de decisión fácil de leer y comprender y como se sabe, este tipo de instrucciones pueden estar anidadas.
- >> Las instrucciones de repetición determinan que un bloque de instrucciones sea ejecutado varias veces hasta que la condición sea verdadera o falsa.

- >> Un bloque en inglés estructurado puede contener cualquier combinación de instrucciones, ya sea de secuencia, decisión o repetición.

- Pseudocódigo.- El pseudocódigo es una modalidad del inglés estructurado. "Pseudo" significa similar a. Por lo tanto, el pseudocódigo es similar a un código real.

La estructura del pseudocódigo se basa en algunos lenguajes de programación tales como COBOL, FORTRAN o PASCAL.

Cuando se usa el inglés estructurado muchos detalles son frecuentemente ignorados o dados por hecho tales como la apertura y cierre de archivos, la inicialización de contadores y el establecimiento de "banderas" (29). En cambio, al usar el pseudocódigo estos detalles son explícitamente codificados, aunque no exactamente con instrucciones de un lenguaje de programación, pero sí de manera que el programador pueda fácilmente trasladarlo al lenguaje de programación que decida emplearse.

Al igual que el inglés estructurado, no existe un pseudo código estándar sino que existen muchas versiones las que en su mayoría tienen incorporados en su filosofía las mismas 3 convenciones estructurales de programación que el inglés estructurado: Secuencia, Decisión y Repetición, que a continuación explicamos brevemente:

- >> Secuencia.- Las operaciones se llevan a cabo de acuerdo a un orden. Se dan reglas en el uso del paréntesis. Los nombres de los datos deben ser tomados del diccionario de datos y/o de las listas de E/S de las cartas IPO.
- >> Decisión.- La forma general para expresar una condición en pseudocódigo es el IF-THEN-ELSE:

```
IF Condición-1 THEN
  Block de Instrucciones 1
ELSE
  Block de Instrucciones 2.
ENDIF
```

(29) UNA "BANDERA", EN EL CAMPO DE LA PROGRAMACION SE DEFINE COMO UNA VARIABLE QUE FUNGE COMO INDICADOR Y CUYO VALOR (1 o 0) SEÑALA SI UNA CONDICION SE HA CUMPLIDO O NO.

Cada block se delimita claramente por medio de un sangrado progresivo y siempre comienza con un IF y termina con un ENDIF.

>> Repetición.- La idea básica del código repetitivo es que un block sea ejecutado una y otra vez hasta que la condición dada se cumpla. En el inglés estructurado no se hace una distinción entre varias formas de lógica repetitiva. En el pseudocódigo si. Las distintas formas de lógica repetitiva que se conocen son:

DO WHILE

```
WHILE <CONDICION> DO
  <INSTRUCCIONES>
END WHILE
```

La condición se da al principio del block de instrucciones y dicho block se efectúa mientras la condición sea verdadera.

REPEAT

```
REPEAT
  <INSTRUCCIONES>
UNTIL <CONDICION>
```

La condición es dada hasta el final del block de instrucciones, el cual deja de ejecutarse cuando tal condición es verdadera, es decir, el bloque se ejecuta mientras la condición es falsa.

DO o FOR

```
DO <INDICE INICIAL> TO <LIMITE FINAL>
  <INSTRUCCIONES>
END DO
```

El block de instrucciones se ejecuta el número de veces que indique el valor del límite final. El índice inicial se incrementa cada vez que se ejecuta el block de instrucciones y funge como un contador.

CASE

```
SELECT <VARIABLE>  
    CASE <VALOR-1> <BLOCK-1>  
    CASE <VALOR-2> <BLOCK-2>  
    CASE <VALOR-N> <BLOCK-N>  
    DEFAULT CASE <BLOCK-DEFAULT>  
END SELECT
```

El CASE resuelve los problemas comunes de la programación consistentes en la selección de una alternativa entre varias.

La decisión tomada por un CASE depende del valor de la variable especificada por el vocablo SELECT. Si ésta tiene el valor de 1 entonces se ejecutará el block número 1 y así sucesivamente. El DEFAULT CASE es ejecutado en caso de que el valor de la variable no sea ninguno de los listados dentro del SELECT. El CASE siempre estará delimitado por el END SELECT.

Al igual que en el inglés estructurado, es posible escribir un número de instrucciones en pseudocódigo y tratarlas como un block o conjunto de operaciones que realizan un proceso definido el cual cumple con un objetivo. Un block puede contener cualquier clase de convencionalismo estructurado de los mencionados anteriormente.

- Diagramas de Flujo.- Un diagrama de flujo es la representación gráfica de la lógica de un programa. Esta representación ayudará a concebir en un esquema un simple proceso; por tanto, el diagrama de flujo es la planeación previa de un programa.

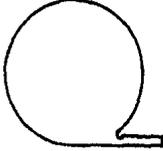
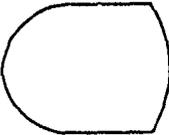
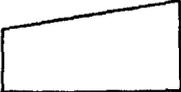
Para indicar el flujo de la lógica mencionada y los pasos del proceso se usan algunos símbolos que permiten describir lo que el programa, módulo o rutina va a realizar. El analista podrá diseñar sus propios símbolos, aunque esto no es necesario ya que la industria de computadoras ha estandarizado un conjunto de símbolos para este objeto y son los que se utilizan normalmente y por lo tanto, tienen una aceptación universal. En 1967 la ANSI (American National Standard Symbols) declara como oficiales los símbolos que se muestran en los CUADROS II.11, II.12, II.13, II.14 Y II.15.

CUADRO II.11
SIMBOLOS BASICOS

NOMBRE	SIMBOLO
ENTRADA/SALIDA	
PROCESO	
FLUJO	

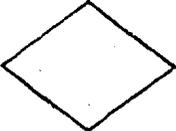
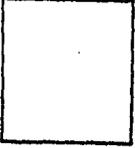
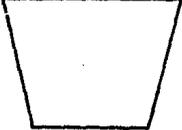
CUADRO II.12

SIMBOLOS ESPECIALES DE ENTRADA/SALIDA

NOMBRE	SIMBOLO
TARJETA PERFORADA	
CINTA MAGNETICA	
CINTA PERFORADA	
DOCUMENTO	
PANTALLA	
ENTRADA MANUAL	

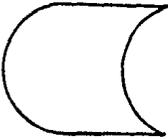
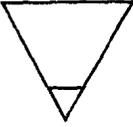
CUADRO II.13

SIMBOLOS ESPECIALES PARA PROCESO

NOMBRE	SIMBOLO
DESICION	
OPERACION AUXILIAR	
OPERACION MANUAL	
PROCESO DEFINIDO CON ANTERIORIDAD	

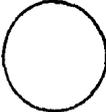
CUADRO II.14

SIMBOLOS ESPECIALES DE ENTRADA/SALIDA PARA ALMACENAMIENTO

NOMBRE	SIMBOLO
ALMACENAMIENTO EN LINEA	
ALMACENAMIENTO FUERA DE LINEA	

CUADRO II.15

SIMBOLOS ADICIONALES

NOMBRE	SIMBOLO
CONECTOR	
TERMINAL	

Los convencionalismos clásicos al elaborar diagramas de este tipo son que el flujo del proceso debe ir de arriba hacia abajo y se deben crear de izquierda a derecha para que haya facilidad en su lectura.

Los mismos convencionalismos estructurales de programación tratados tanto en el inglés estructurado como en el pseudocódigo pueden ser representados en este punto por medio de las figuras lógicas de la programación que presentamos en los siguientes diagramas:

DIAGRAMA II.6

FIGURA DE SECUENCIA

El patrón de secuencia implica que la lógica es ejecutada en una secuencia simple: un bloque después de otro. El bloque de secuencia se identifica en el diagrama de flujo por un rectángulo que puede representar una o más instrucciones:

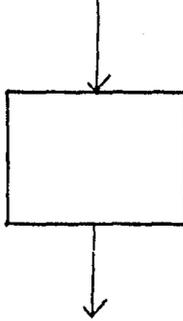


DIAGRAMA II.7

FIGURA DE DECISION

Un bloque de decisión implica el uso de la figura clásica del IF-THEN-ELSE:

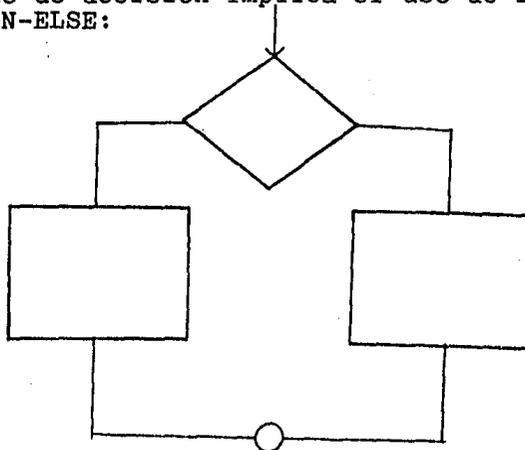


DIAGRAMA II.8

FIGURA DE REPETICION: DO WHILE

La pregunta o condición se prueba antes de ejecutar el bloque de instrucciones; si la condición es verdadera, entonces dicho bloque se ejecuta hasta el momento en que tal condición sea falsa y es en este momento cuando el control del proceso se transfiere a otro bloque.

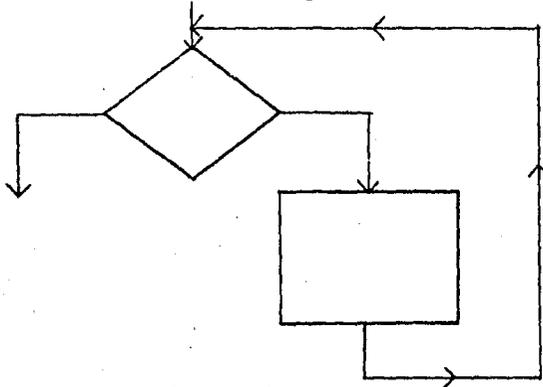


DIAGRAMA II.9

FIGURA DE REPETICION: DO UNTIL

La pregunta o condición se prueba después de haber ejecutado el bloque de instrucciones; si la condición es falsa, entonces dicho bloque se ejecuta hasta el momento en que tal condición sea verdadera y es en este momento cuando el control del proceso se transfiere a otro bloque.

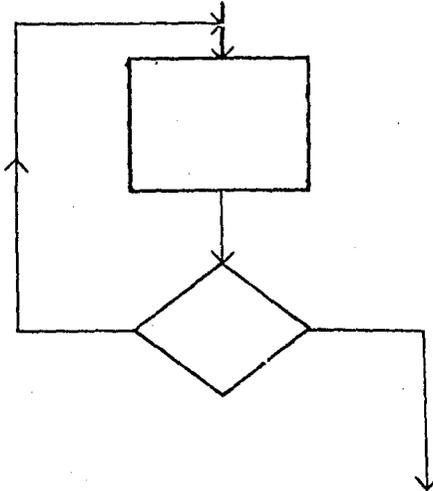


DIAGRAMA II.10

FIGURA DE REPETICION: CASE

En caso de tener varias opciones se evalua el valor de una variable y dependiendo de su valor se ejecutará la secuencia correspondiente.

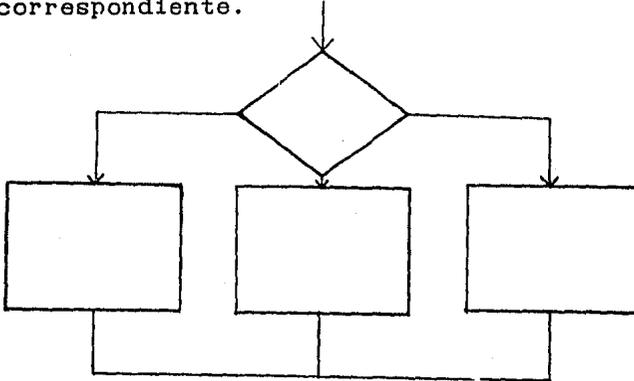
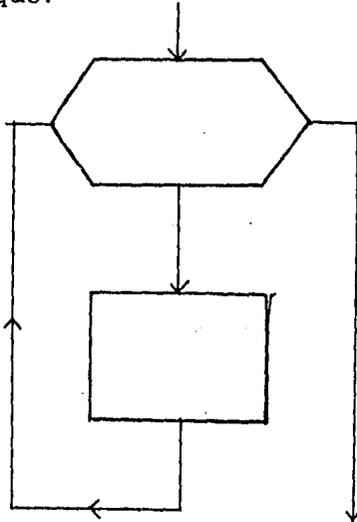


DIAGRAMA II.11

FIGURA DE REPETICION: FOR O DO CONTINUO

Una variable comienza con un límite inferior y hasta que llegue a su límite superior, por medio de incrementos constantes, deja de ejecutar el proceso o bloque de instrucciones. Es un ciclo fijo y determinado ya que se establece el número de veces que se van a ejecutar las instrucciones contenidas en el bloque.



Por años la documentación estándar que se ha usado para diseñar y expresar la estructura de un programa ha sido los diagramas de flujo. De hecho muchos textos usan únicamente los diagramas de flujo como herramientas especiales para diseñar programas. En cierta forma este hecho ha provocado que los diagramas de flujo ya no sean tan estimados y que en ocasiones ya no se les dé el valor que en realidad tienen. Muchas veces el diagrama de flujo es realizado hasta después de que el programa ha sido escrito o simplemente no se recurre a él sino que los programadores directamente construyen su programa en el computador lo cual sólo da lugar a que después se pierda mucho tiempo en depuraciones.

En casos sencillos es factible elaborar un simple programa sin necesidad de diseñar el flujo del proceso a través de un diagrama de flujo, pero, cuando el problema es de mayor magnitud, será necesario que exista una documentación formal o un modelo que nos permita crear un sistema eficiente de la misma manera que cuando se desea hacer un edificio sin planos.

Así como los planos de un edificio, los diagramas sirven para dos propósitos: describir la estructura del programa para que después éste pueda implantarse en el computador sin ningún problema y documentar la lógica del proceso para efectos de comunicación o bien, para usarla como referencia cuando el programa requiera de modificaciones.

- Tablas de Decisión.- Las tablas de decisión resultan útiles cuando alguno de los algoritmos del sistema contiene un gran número de decisiones anidadas.

Una tabla de decisión está dividida en cuatro partes que son:

- >> Sección de Condiciones.
- >> Sección de Respuesta a tales Condiciones.
- >> Sección de Acciones.
- >> Sección de Realización de las Acciones.

- >> Sección de Condiciones.- Se enumeran las condiciones o preguntas que deben considerarse para llevar o no a cabo la condición.

- >> Sección de Respuesta a tales Condiciones.- Se da la respuesta, ya sea negativa o afirmativa, del cumplimiento de las condiciones dadas en la sección anterior.
- >> Sección de Acciones.- Corresponde al número de acciones que son factibles de realizarse dependiendo del resultado de la acción.
- >> Sección de Realización de las Acciones.- Contiene los resultados finales, es decir, en esta sección se marcan las acciones a seguir para cada una de las situaciones dadas condicionalmente.

EJEMPLO II.20

TABLA DE DECISION PARA LA SELECCION DE INFORMACION

Se desea obtener, de un archivo con información bibliográfica, hemerográfica y demás, todos aquellos registros que cumplan con las siguientes condiciones:

- >> Que el Tipo de Material sea Artículo Periodístico.
- >> Que el Contenido de tales Artículos sea de Tipo Financiero.
- >> Que el Año al que corresponda sea 1986.

Veamos el caso en la tabla de decisiones de la siguiente página: Si el registro cumple con las condiciones requeridas entonces la acción No 1 (listar los datos) será ejecutada para que el investigador pueda obtener la información que desea. En el caso en el que el registro no cumpla con las 3 condiciones, tal registro no se tomará en cuenta. Debe aclararse en este ejemplo que el registro debe cumplir forzosamente con las 3 condiciones establecidas.

Las tablas de decisión sólo nos señalan las condiciones a las que debe sujetarse un determinado hecho así como las acciones que deben realizarse dependiendo de la respuesta a la o las condiciones dadas, pero no nos indica como realizar el proceso a nivel de programación.

Las tablas de decisión son bastante útiles cuando hay varias decisiones anidadas de las cuales depende la realización o no de un hecho. En cada tabla se observa claramente las condiciones a las que un hecho debe sujetarse así como las acciones correspondientes dependiendo de la respuesta. Podrán hacerse tantas tablas como lo requiera el caso y número de situaciones.

CONDICIONES	RESPUESTAS (SI/NO)						
	REGISTROS						
	1	2	3	4	5	6	7
1) ES ARTICULO PERIODISTICO?	SI	SI	NO	NO	SI	NO	SI
2) SU CONTENIDO ES FINANCIERO?	SI	SI	SI	NO	SI	NO	SI
3) EL AÑO ES 1986?	NO	SI	NO	NO	SI	SI	SI
ACCIONES	REALIZACION DE ACCIONES (SI: X)						
LISTAR: NOMBRE PERIODICO, LUGAR DONDE SE ENCUENTRA DOCTO., PAGS. Y FECHA.	X				X		X
DESCARTAR EL REGISTRO PORQUE NO CUMPLE CON LAS TRES CONDICIONES	X		X	X		X	

- Árboles de Decisión.- Los árboles de decisión son una herramienta más de la ciencia de la administración que puede ser empleada por el analista en los caso en los cuales se desee conocer y estudiar cada uno de los resultados derivados de la toma de varias decisiones o alternativas.

Gracias a esta herramienta se logra representar los andamiajes lógicos de un programa de computador. Un árbol de decisión se define como la representación gráfica de una situación que implica la toma de decisiones y el conocimiento de sus consecuencias correspondientes.

El árbol comienza a la izquierda con una raíz de la cual parten los caminos posibles que pueden ser 2 o más. Esta raíz puede estar representada por un pequeño cuadro. De cada rama primaria puede surgir una nueva bifurcación la cual se representa con un círculo que contendrá el número de la decisión o alternativa. De cada uno de estos círculos podrán surgir más ramas, alternativas o consecuencias. El final del árbol de decisiones (que termina en el extremo derecho) quedará terminado con óvalos los cuales contendrán el resultado de tomar o darse el camino correspondiente.

Si retomamos el ejemplo del archivo expuesto para explicar las tablas de decisión, podrá comprenderse mejor el uso de esta herramienta. Las condiciones dadas en tal ejemplo eran las siguientes:

- >> Que sea un Artículo Periodístico.
- >> Que el Contenido sea de Tipo Financiero.
- >> Que el Año al que pertenezca sea 1986.

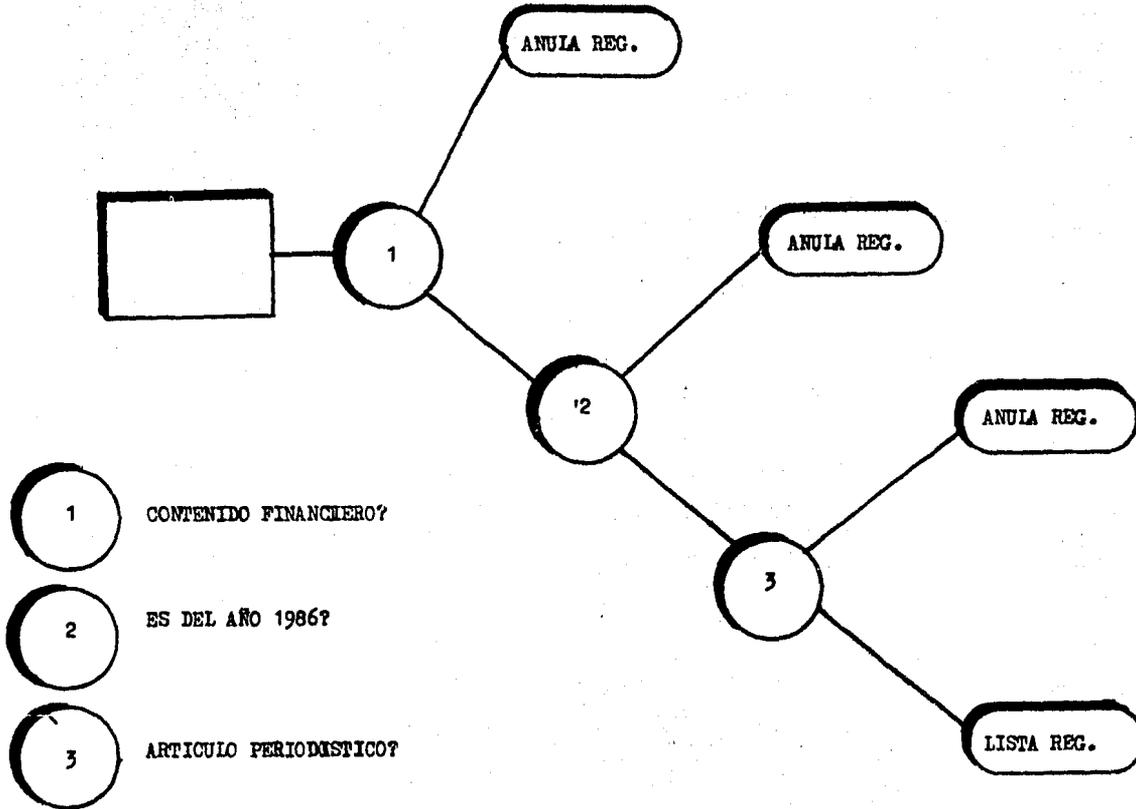
En este caso, por razones de eficiencia deberá considerarse primero aquella condición más selectiva y discriminatória que elimine a un número mayor de registros que no cumplan con tal condición. Por ejemplo, si suponemos que el archivo contiene escasos registros cuyo contenido sea de tipo financiero, convendrá primero sujetar el archivo a la segunda condición de tal manera que se elimine un número mayor de registros que no cumplan con ésta condición y así sucesivamente ir escogiendo las condiciones menos selectivas. De esta forma se evitarán algunas pérdidas de tiempo y de recursos.

Considerando la observación anterior, el árbol de decisión del ejemplo anterior se observa en la siguiente página. (EJEMPLO II.21).

En general, cuando se diseña un algoritmo de decisiones anidadas, siempre deberá ejecutarse la prueba más discriminatória. De esta manera tendremos como resultado un programa más eficiente y con un menor costo.

EJEMPLO II.21

ARBOLE DE DECISIONES PARA LA SELECCION DE UN REGISTRO EN UN ARCHIVO



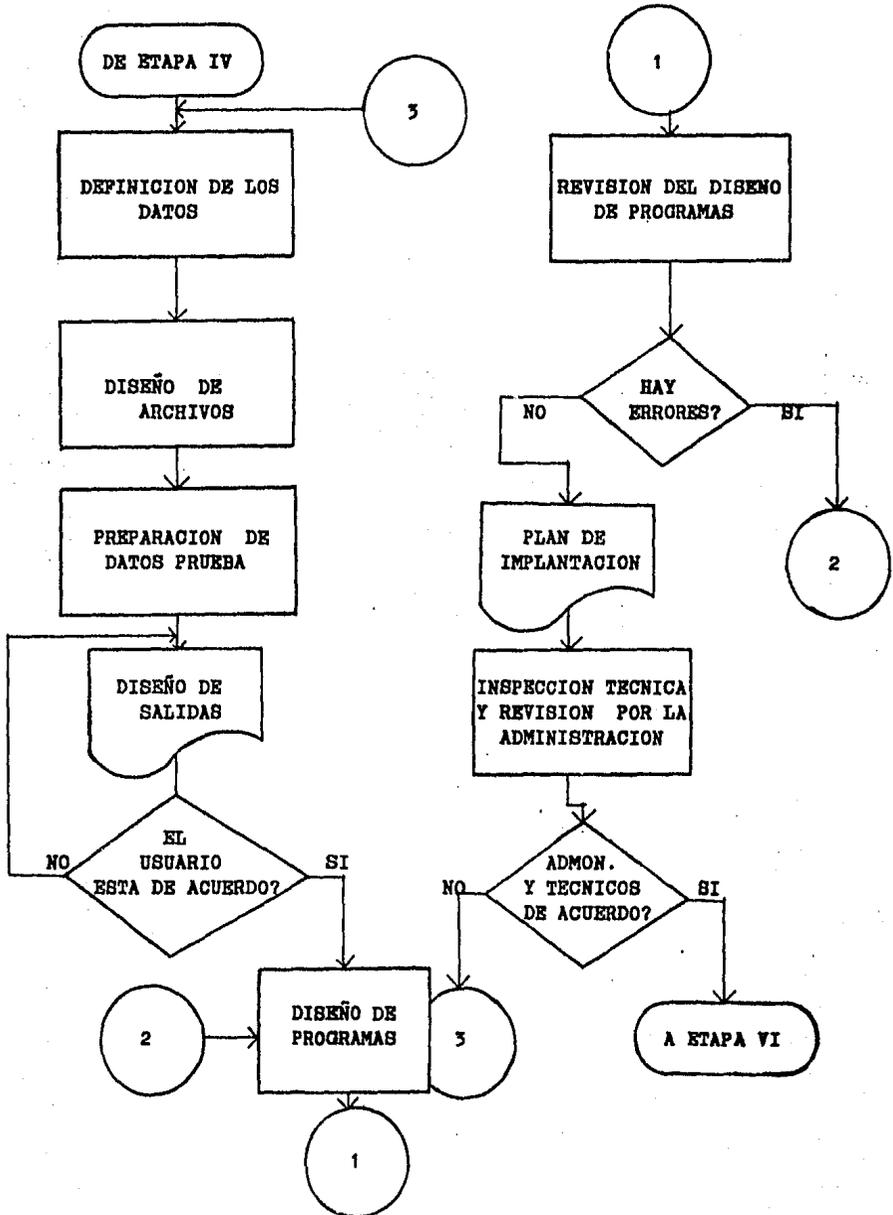
CUADRO II.15

ETAPA: DISEÑO DETALLADO

?	?	?	?
?	PREGUNTA A	?	?
?	CONTESTAR	?	?
?		?	?
?	A	?	?
?	C	?	?
?	T	?	?
?	I	?	?
?	V	?	?
?	I	?	?
?	D	?	?
?	A	?	?
?	D	?	?
?	E	?	?
?	S	?	?
?		?	?
?		?	?
?	DOCUMENTOS	?	?
?	GENERADOS	?	?
?		?	?
?	CRITERIO	?	?
?	DE EXITO	?	?
?		?	?
?		?	?

DIAGRAMA II.12

ETAPA: DISEÑO DETALLADO



2.6 IMPLANTACION.- La implantación puede definirse como el conjunto de actividades que permiten que un sistema de información computarizado se traslade de una etapa lógica (concepción abstracta plasmada en un modelo) a una etapa física, de tal manera que su funcionamiento se convierta en algo tangible y concreto. A grandes rasgos, en esta fase el sistema es creado físicamente. Los programas son codificados, depurados y documentados. Nuevo hardware es seleccionado, ordenado e instalado. Se desarrollan procedimientos de auditoría y seguridad y se establece la prueba formal del sistema, actividad con la cual finaliza esta etapa.

En la fase de implantación es necesario llevar a cabo las siguientes actividades:

- a) Preparación del Espacio Físico.
- b) Codificación.
- c) Documentación de los Programas.
- d) Depuración.
- e) Prueba Formal del Sistema.
- f) Capacitación al Usuario.
- g) Documentación del Sistema.

a) Preparación del Espacio Físico.- Es claro que el diseño y preparación del espacio físico para la instalación de un computador debe estar de acuerdo al tamaño y configuración del equipo que se halla adquirido: macrocomputador, minicomputador o microcomputador, así como también al número de personas que trabajarán con el sistema. Es importante hacer una planeación de como quedará distribuido el equipo y el personal en el espacio disponible, considerando que el ambiente de trabajo es un factor importante.

El diseño del espacio físico puede planearse en papel, dibujando o marcando sobre éste el lugar en el que quedará cada elemento del departamento: equipo y personal.

b) Codificación.- La codificación es la escritura de las instrucciones detalladas utilizando un lenguaje de alto nivel. La codificación consiste en transformar las operaciones y el proceso de cada módulo del sistema, reflejadas en las cartas IPO, en instrucciones de un determinado lenguaje de alto nivel (30).

(30) UN LENGUAJE DE ALTO NIVEL ES UN LENGUAJE DE PROGRAMACION QUE PERMITE CON UNA SOLA INSTRUCCION, HACER UNA SERIE DE OPERACIONES QUE EN OTRO TIPO DE LENGUAJES (ENSAMBLADOR O DE MAQUINA) DEBEN DESGLOSARSE Y ESPECIFICARSE UNA TRAS OTRA.

Para codificar adecuadamente cada uno de los programas resulta de utilidad conocer algunos aspectos de la programación estructurada.

El objetivo de la programación estructurada es realizar programas que sean fáciles de entender, de depurar y de mantener. Todo ello con la finalidad de minimizar los costos de depuración y mantenimiento de los programas.

"El principio básico de la programación estructurada es similar al de la estrategia militar de dividir y luego conquistar" (31). Las principales reglas de la programación estructurada son:

- Dividir el programa en pequeñas funciones independientes y sencillas con la finalidad de que sean fáciles de entender.
- Cada módulo no debe exceder de una hoja de codificación.
- Los nombres escogidos para cada módulo deben formarse con un verbo imperativo seguido de un sustantivo, de tal manera que se exprese con claridad la función del módulo así nombrado. Ejemplo: "Trae Registro", "Procesa Nómina", "Actualiza Inventario", etc.
- Los módulos principales deberán subdividirse en bloques o conjuntos de instrucciones que realicen una función específica.

Si el analista y el programador toman en cuenta las reglas anteriores, es probable que el costo de implantación del sistema sea minimizado, ya que los componentes del sistema (rutinas, módulos o programas) serán sencillos, manejables e independientes.

La elección del lenguaje en el que se programará es una decisión importante, y a la vez difícil, que el analista debe tomar. Cada lenguaje ofrece ventajas y desventajas para una determinada utilidad y por tanto el analista debe evaluar y escoger aquel lenguaje de programación que más resuelva las exigencias del futuro sistema ya que en esta decisión muchas

(31) DAVIS. OP.CIT. PAG. 100.

veces está en juego la eficiencia y por lo tanto el costo. Además, el hecho de que el analista deba decidir qué lenguaje de programación se empleará, implica que debe conocer a la mayoría de ellos y estar a la vanguardia en software de este tipo.

La captura de los programas ya codificados es una actividad obvia que debe contemplarse en este punto en cuanto al tiempo que llevará realizarla.

c) Documentación de los Programas.— La documentación de un programa está básicamente dirigida a los programadores. El documentar un programa significa realizar dentro de él algunos comentarios que expliquen, en forma breve, el funcionamiento y características específicas del mismo, con la finalidad de que algún otro programador nuevo, que en un futuro se encargue de mantener ese mismo programa, las pueda comprender con facilidad. En este sentido, los objetivos primarios serán la calidad y brevedad tanto de los comentarios como de las explicaciones que se hagan en el programa fuente.

De cualquier manera, aunque un programa no sea documentado, debe ser fácil de comprender ya que se supone que debe ser un programa bien estructurado que por sí mismo ofrecerá sencillez en su lectura.

d) Depuración.— Esta actividad tiene por objeto eliminar todos los errores de programación que pudieron haberse suscitado después de la compilación (32). Generalmente, los errores de compilación son sencillos de encontrar y corregir, en cambio, los errores de lógica no lo son tanto.

Si el programa fuente se diseñó en forma modular, el proceso de depuración será más fácil ya que en primera instancia, cada módulo se revisará en forma independiente para detectar los errores que se generen en él y después, se harán las pruebas generales.

(32) LOS TIPOS DE ERRORES QUE PUEDEN COMETERSE AL ESCRIBIR UN PROGRAMA SON:

- ERRORES DE SINTAXIS: RADICAN EN EL USO INCORRECTO DE LAS REGLAS QUE GOBIERNAN EL LENGUAJE.
- ERRORES DE LOGICA: SUELEN OCURRIR CUANDO EL DISEÑO DETALLADO DE LOS PROGRAMAS NO FUE REALIZADO CORRECTAMENTE. EL COMPILADOR NO DETECTA LOS ERRORES DE LOGICA NI MUCHO MENOS LO QUE UN PROGRAMADOR INTENTA HACER. CUANDO HAY UN ERROR DE LOGICA EL SISTEMA GENERA RESULTADOS EXTRAÑOS E IMPRESISOS ADEMÁS DE QUE NO SE COMPORTA TAL Y COMO SE PLANEÓ.
- ERRORES DE DATOS: SE PRESENTAN CUANDO INGRESAN AL PROCESO DATOS ERRONEOS.

Es común que en ocasiones el programador tenga dificultades al depurar su propio código. Frecuentemente cuando programamos, leemos en el programa fuente lo que pensamos hacer y no lo que realmente hicimos. No es de dudarse que algunas veces el programador se la ha pasado solo, buscando por horas en el programa fuente tratando inútilmente de encontrar el error. Lo anterior demuestra que en ocasiones se requiere del trabajo en equipo para que los integrantes del mismo analicen y discutan juntos el programa. De esta manera se logrará la generación de numerosas ideas, preguntas y respuestas que a la larga facilitarán, sobre todo, la detección de errores de lógica.

e) Prueba Formal del Sistema. - La prueba formal del sistema representa el momento crucial para comprobar si todo el trabajo del equipo de sistemas generó o no resultados fructíferos. Elias Awad opina el respecto afirmando lo siguiente: "Consiste en la prueba o revisión del sistema para verificar si realiza o no lo deseado. Incluye ejecuciones en paralelo para estar seguro de que las salidas del nuevo sistema son idénticas al viejo sistema" (33).

La prueba formal del sistema debe considerar todos los elementos que conformarán el sistema físico:

- Datos.
- Programas.
- Archivos.
- Reportes.
- Procedimientos Manuales.
- Participación del Personal.

La elaboración del plan para la prueba formal del sistema es esencial. La prueba debe ser ejecutada en dos niveles: Primero, cada módulo debe ser probado de manera independiente; después, cuando todos los módulos han pasado las pruebas independientes, deben combinarse y conectarse para que el sistema sea probado en su totalidad. En todas estas pruebas los datos, preparados para el objeto en la fase de diseño detallado, jugarán un importante papel. Las pruebas pueden ser de tres formas a saber: en condiciones extremas, en condiciones normales y en condiciones erróneas.

Una vez que el encargado del proyecto y su equipo de trabajo han probado el sistema en su totalidad, será conveniente que los usuarios, generalmente no expertos en el área, prueben el sistema y que se les permita usarlo por algunos días con la finalidad de que ellos mismos, a través del uso normal del mismo, encuentren sus posibles deficiencias.

Después de una exitosa prueba (que se realiza normalmente para la administración y los usuarios de manera formal), se procederá a la ejecución del sistema en paralelo. Una prueba en paralelo consiste en ejecutar de manera simultánea y durante un intervalo de tiempo (1 mes o 2 meses) los dos sistemas de información: El sistema antiguo y el nuevo, con la finalidad de detectar errores vía la comparación de los resultados generados por ambos sistemas.

Una vez que el usuario y la administración quedan satisfechos con la demostración final del nuevo sistema, se ha llegado ya a su feliz terminación, pero sólo en forma parcial, ya que requerirá de un mantenimiento constante.

f) Capacitación al Usuario. - Quizá la capacitación al usuario es un factor al cual no se le da la importancia que debiera y por lo mismo, se hace de una manera poco formal y breve.

El analista será el responsable de capacitar al personal que usará o que estará a cargo del nuevo sistema. Muchas veces se enfrentará con dos fenómenos muy comunes: El miedo y la aversión a la tecnología por parte del personal. Estos dos fenómenos son muy importantes ya que cuando se presentan en el personal, simple y sencillamente el sistema de información del computador no será aprovechado tal cual por muy bien que éste haya sido diseñado. Será inútil cualquier esfuerzo si la gente se rehusa a usarlo. Una buena capacitación podrá enfrentar tales sentimientos tan naturales en el hombre: el miedo y la duda. De la persona encargada de la capacitación dependerá todo esto.

Por otro lado, se puede hablar de una capacitación previa que comienza cuando el sistema se está diseñando. El analista, en la etapa de diseño, podrá organizar algunas entrevistas con los usuarios finales con el objeto de conocer sus ideas y sugerencias e incorporarlas en el sistema aún no im-

plantado. De esta manera los usuarios sentirán que ellos han intervenido en el diseño del sistema y que han tenido la oportunidad de participar en la realización del mismo. A la par de esto, podrán ir siendo informados sobre el avance del trabajo y por lo tanto, conocerán los futuros resultados del mismo.

De cualquier manera, cuando se trata de un equipo de sistemas ajeno a la empresa, el encargado del proyecto debe recordar las siguientes reglas: (34)

- Llegada la Aplicación el Usuario es el Experto.
- El Usuario es el Cliente y el Cliente tiene la Razón.

g) Documentación del Sistema.- Consiste en la obtención de todos los documentos relacionados con el sistema y la organización de éstos para elaborar un instructivo de trabajo y operación para el usuario y para las personas que se encargarán de mantener dicho sistema.

Pueden elaborarse dos tipos de documentos: el manual de operación para el usuario y el manual técnico para los administradores o encargados de dar mantenimiento al sistema. Los documentos fuente para elaborar estos dos manuales son todos aquellos documentos generados en cada una de las etapas del ciclo de vida estructurado los cuales ya fueron explicados a lo largo de este capítulo.

A continuación se hablará brevemente de los dos tipos de manuales existentes:

- Manual de Operación.- Tiene por objeto describir al usuario el funcionamiento del sistema a detalle para que, aparte de la capacitación que reciba, conozca las especificaciones y funciones de dicho sistema. Los futuros usuarios necesitan saber lo que el programa hace y lo que no hace, como preparar los datos e interpretar las salidas, etc. y para resolver estas cuestiones, el usuario podrá consultar el manual de operación.

La información indispensable que debe contener el manual es la siguiente:

- >> Detalles de los Datos de Entrada (Formatos, Longitud, etc.

- >> Descripción de las Salidas que Genera el Sistema.
- >> Instrucciones de Operación del Sistema en General, si es necesario con ejemplos.
- >> Software Necesario para poner en Marcha el Sistema.
- >> Instrucciones para el Manejo Adecuado del Equipo.

No es muy necesario que el usuario conozca la estructura y funcionamiento interno del sistema ya que este punto le compete a las personas que se encargarán de su mantenimiento.

- Manual Técnico.- Lo que pretende este manual es básicamente informar, a aquellas personas que se encargarán del mantenimiento del sistema, sobre las características físicas del mismo con el objeto de permitirles hacer las modificaciones y actualizaciones posteriores una vez que el sistema ya está en funcionamiento.

Este documento puede contener entre otras cosas los siguientes puntos:

- >> Identificación: Nombre del sistema, autor(es), versión, fecha(s), breve descripción del sistema, lenguaje de programación utilizado y razones de su elección.
- >> Especificaciones del Sistema: Archivos usados, estructura completa del sistema, programas que lo componen, restricciones y limitaciones del sistema, etc.
- >> Información sobre la Programación: Toda la documentación generada en las etapas de análisis, diseño y diseño detallado del sistema (diagramas de flujo de datos, algoritmos, diagrama de bloques del sistema, documentación HIPO, etc.), descripción de archivos, listado del programa fuente, descripción técnica del hardware y del software empleado, muestras de pruebas realizadas al sistema, instrucciones de operación, etc.

En el CUADRO II.16 y el DIAGRAMA II.13 se hace un resumen de la etapa de implantación del sistema.

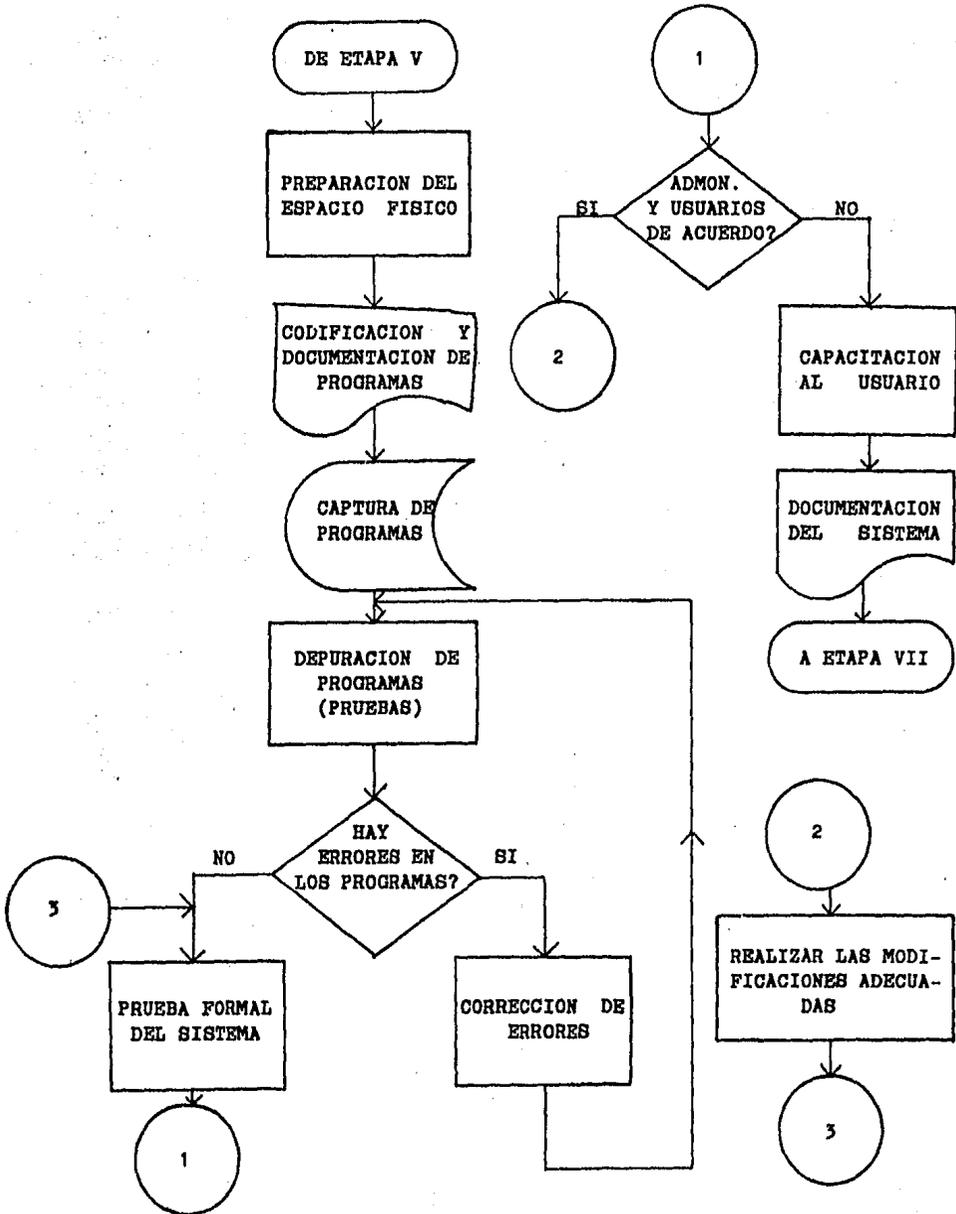
CUADRO II.16

ETAPA: IMPLANTACION

PREGUNTA A CONTESTAR	NINGUNA. SOLO IMPLANTAR EL SISTEMA
A	
C	
T	
I	- PREPARACION DEL ESPACIO FISICO.
V	- CODIFICACION.
I	- DOCUMENTACION DE LOS PROGRAMAS.
D	- DEPURACION.
A	- PRUEBA FORMAL DEL SISTEMA.
D	- CAPACITACION AL USUARIO.
E	- DOCUMENTACION DEL SISTEMA.
S	
DOCUMENTOS GENERADOS	- MANUAL DE OPERACION PARA EL USUARIO. - MANUAL TECNICO.
CRITERIO DE EXITO	- CODIFICACION Y DOCUMENTACION DE PROGRAMAS. - DESICIONES SOBRE HARDWARE. - OPERACION Y SEGURIDAD DE PROCEDIMIENTOS. - AUDITORIA DE LOS PROCEDIMIENTOS. - PRUEBA FORMAL DEL SISTEMA.

DIAGRAMA II.13

ETAPA: IMPLANTACION



2.7 MANTENIMIENTO.- La etapa de mantenimiento es precisamente la que viene a consolidar el funcionamiento de un sistema de información computarizado, ya que tiene por objeto vigilar que tal sistema funcione en forma adecuada y se mantenga actualizado.

El mantenimiento del sistema se inicia tan pronto como este empiece a usarse en forma normal. Muchas veces es fácil pensar que una vez implantado y probado, ya se ha terminado, pero tal pensamiento es falso ya que siempre habrán pequeños detalles o errores que sólo se llegarán a identificar con la práctica, en un tiempo probable que va desde semanas hasta años de uso del sistema.

Las causas principales que originan la necesidad de mantener un sistema de información computarizado son, entre otras, las siguientes:

- Algunos valores empleados por el sistema, que supuestamente son de carácter constante, serán modificados tarde o temprano y por alguna u otra razón, por el entorno de tal sistema (salario mínimo, sueldos base, impuestos por diversos conceptos, demanda, precios, cantidades de pedidos, etc.).
- La administración o el usuario constantemente tendrán nuevas necesidades de información: nuevos reportes, cálculos, controles, etc.
- Requerimientos de renovación de hardware por razones de obsolescencia.
- Desarrollo de nuevos sistemas de información dentro de la misma organización, lo cual requerirá, por razones de eficiencia en la información, entrelazar tales sistemas con el anterior para así generar un sistema de información integral.

En esta etapa podemos hablar de dos tipos de mantenimiento:

- a) Mantenimiento de Programas.
- b) Mantenimiento de la Documentación del Sistema.

a) Mantenimiento de Programas.- El mantener un programa significa actualizarlo y adecuarlo a las cambiantes necesidades de la organización, en cuanto a información se refiere, a lo largo del tiempo.

La economía del procesamiento de los datos está cambiando. Históricamente los costos de hardware eran tan altos que los programadores se preocupaban por el tiempo del CPU (Central Process Unit) y por el espacio en memoria que su programa ocupaba.

Ahora, dado el descenso en los costos del hardware, el costo del desarrollo y del mantenimiento de programas es frecuentemente mucho más alto que la ejecución de los mismos. Por esta razón, lo que se debe pretender es crear, desde un principio, programas que sean fáciles de mantener; esto se logra realizando un sistema cuyas partes sean independientes y que además cada una de ellas realice una sencilla y completa función, para que de esta manera el costo de mantenimiento de los programas del sistema logre minimizarse.

Yourdon nos señala respecto a lo anterior lo siguiente: "El costo de mantenimiento es minimizado cuando las partes del sistema sean pequeñas, manejables, independientes y sencillas en su comprensión" (35).

El mejor camino para minimizar el costo de mantenimiento de los programas es entonces, el diseño del sistema por medio de la modularización. Si cada módulo es independiente y ejecuta una sencilla y completa función, entonces, la mayoría de los cambios se limitarán sólo a la lógica del módulo a modificar. Con esto el efecto de "onda" (36) es minimizado.

El efecto de "onda" se observa claramente en el manejo de los datos locales y de los datos globales, conocidos ambos como parámetros. Los datos locales son definidos, almacenados y usados estrictamente dentro de un determinado módulo. Si se requiere modificar un dato de esta naturaleza, probablemente dicho cambio no afecte el resto del programa. Por otro lado, los datos globales, al ser normalmente definidos, almacenados a un alto nivel de la carta estructurada y usados en más de un módulo, pueden crear un efecto de "onda" cuando hay un cambio en ellos, causando dificultades en el seguimiento de las implicaciones de flujo de datos en los demás módulos, de

(35) YOURDON. OP.CIT. PAG. 19.

(36) EL EFECTO DE ONDA ES AQUEL QUE SE GENERA CUANDO SE MODIFICA UNA RUTINA, QUE AL SER TAN DEPENDIENTE DEL RESTO DEL PROGRAMA, TAL MODIFICACION LLEGA A AFECTAR EL FUNCIONAMIENTO DE LAS DEMAS RUTINAS. POR LO TANTO, LO QUE SE RECOMIENDA ES GENERAR RUTINAS LO MAS INDEPENDIENTE POSIBLES Y QUE SOLO ESTEN LIGADAS POR UN PROGRAMA MAESTRO QUE LAS CONTROLE.

tal manera que el realizar un cambio en un dato global es un hecho que debe ser cuidadosamente evaluado.

Si un dato local o global es separado, el programador podrá determinar rápidamente si el efecto de "onda" será un hecho importante a considerar para así generar un plan de mantenimiento. Simplemente, con identificar si una variable es local o global, con ello se tendrá una importante ayuda.

b) Mantenimiento de la Documentación. - El mantenimiento de los documentos que fundamentan y describen al sistema, normalmente estarán en función de las modificaciones físicas que se la hagan al mismo (modificación a programas).

El mantener la documentación de un sistema siempre actualizada representa un problema ya que normalmente lo que ocurre es que una vez realizada ésta al finalizar el sistema, jamás se vuelve a tocar y poco a poco se va convirtiendo en obsoleta. Los programadores generalmente no tendrán mucha espontaneidad para actualizar por ejemplo un diagrama de flujo, es decir, realmente no mantendrán la documentación a menos que el hacerlo sea verdaderamente de utilidad y fácil. Quizá el documentar sólo por documentar representa un gasto de tiempo y dinero. Por lo tanto debe haber una razón válida que justifique el mantener una documentación actualizada.

CUADRO II.17

ETAPA: MANTENIMIENTO

!	!	!	!
!	PREGUNTA A	!	QUE DETALLES DEL SISTEMA DEBERAN MODIFICARSE CON
!	CONTESTAR	!	EL TIEMPO?
!	!	!	!
!	A	!	!
!	C	!	!
!	T	!	!
!	I	!	!
!	V	!	- MANTENIMIENTO DE PROGRAMAS.
!	I	!	- MANTENIMIENTO DE LA DOCUMENTACION.
!	D	!	- MANTENIMIENTO DEL HARDWARE.
!	A	!	!
!	D	!	!
!	E	!	!
!	S	!	!
!	!	!	!
!	!	!	!
!	CRITERIO	!	- CONTINUAR REVISANDO EL FUNCIONAMIENTO DEL SISTEMA.
!	DE EXITO	!	- MODIFICAR EL SISTEMA TANTO COMO SEA NECESARIO.
!	!	!	!

C A P I T U L O I I I

T E C N I C A J S D

(J A C K S O N S Y S T E M D E V E L O P M E N T)

CAPITULO III

JSD

1. DESCRIPCION DE LA TECNICA

JSD es una técnica para especificar e implantar sistemas en una computadora. Se diferencia de otras técnicas desde los mismos conceptos que maneja pues las actividades tales como el análisis, diseño y programación las engloba dentro del término "desarrollo" y así mismo cualquier persona que intervenga en esta actividad le nombra "desarrollador". En esta técnica el desarrollo se divide en dos grandes actividades, la primera, que se compone de cuatro procedimientos, y es la que abarca lo referente a la especificación del sistema requerido, y la segunda compuesta de dos procedimientos, se refiere a la implantación de dicho sistema.

JSD no abarca todas las actividades concernientes al desarrollo de sistemas, excluye aquellas tales como la selección del proyecto, planeación y administración del mismo, estudios del costo-beneficio, aceptación e instalación. Pretende más bien garantizar que el sistema refleje correctamente al mundo real tal y como es, así como proveer las funciones requeridas por el.

JSD no parte de especificaciones dadas ni descompone al sistema en procesos secuenciales, sino que el desarrollo comienza creando las especificaciones del sistema, y construyendo a éste de partes que en sí sean procesos secuenciales. La actividad es de síntesis, más no de descomposición. Así mismo JSD provee de una técnica para la implantación al combinar los procesos secuenciales de las especificaciones dentro de un sistema que sea ejecutable en forma eficiente.

Tradicionalmente, el punto de inicio en el desarrollo de un sistema son los requerimientos del mismo. El desarrollador empieza por establecer las funciones del sistema determinando que es lo que deberá hacer y que reportes generará. Generalmente este es un buen punto de partida, pues si el propósito del sistema es que haga algo, es atractivo pensar que éste es

el punto más conveniente para iniciarlo.

JSD relega la consideración de las funciones del sistema a un punto posterior en el desarrollo y ubica en primera instancia la actividad de modelar el mundo real, siendo ésta un principio fundamental.

El sistema es considerado como un tipo de simulación de la realidad, un modelo que simule al mundo real que existe fuera del computador, más no comprender por modelo al sistema mismo o a los procedimientos usados para crearlo. La creación de un modelo en JSD del mundo real envuelve dos tareas:

- a) Hacer una descripción abstracta del mundo real.
- b) Construir la realización de esa descripción abstracta en una computadora.

La realización en la computadora será entonces el modelo del mundo real, así como una maqueta será el modelo de un edificio.

Una descripción abstracta de la realidad siempre será parcial, ya que se escogerán sólo aquellas partes importantes, y se omitirán las otras.

El modelo nos proporcionará las bases conceptuales y el contexto para establecer posteriormente las especificaciones funcionales del sistema, y las conexiones entre el mundo real y el modelo a través de las mismas palabras utilizadas para describirlo. El modelo en forma implícita definirá las funciones del sistema.

Si nosotros empezáramos por definir las funciones de un sistema, realmente estaríamos manejando un pseudo modelo de la realidad a la cual va a servir, pero al no tomar en cuenta un modelo completo, con el cual podamos prevenir futuras necesidades, estaríamos cayendo en uno de los problemas más generalizados: El cambio de las especificaciones funcionales aunque éstas sean mínimas, ocasionando con esto elevados costos de mantenimiento, ya que al querer adecuar el sistema nos daremos cuenta de que éste fue hecho en forma sumamente rígida y que será imposible cambiarlo, por lo que en muchas ocasiones se tendrá que volver a programar el sistema completo. Por otro lado, si se inicia el desarrollo de un sistema por

crear un modelo de la realidad a la cual servirá, deben tenerse presente las funciones a las que se aplicará, pues como se mencionaba anteriormente, el modelo creado sólo tomará en cuenta los factores importantes y representará parcialmente a la realidad. Podemos utilizar la siguiente analogía: Un mapa será en sí, un modelo del territorio al que representa, pero éste sólo mostrará una parcialidad de la realidad, dependiendo de nuestras necesidades, por ejemplo, se tienen mapas que sólo muestran la hidrología de un territorio, o su orografía, o su división política. Pero si lo que se desea es viajar en auto, se necesitará un mapa que muestre las carreteras por las que se puede circular, siendo así que el modelo cubrirá por completo las necesidades del usuario del mapa tan sólo por representar fielmente a la parte de la realidad que se necesita, y se podrán definir tantos recodos por carretera como se deseen sin tener que modificar los modelos. Esto demuestra que la implantación de nuevas funciones no afectan al modelo, pero si solamente hubiéramos armado un modelo para un sólo recorrido, que es lo análogo a iniciar el desarrollo de un sistema por sus funciones, al querer implantar otro recorrido, el modelo no lo hubiese podido soportar.

Las funciones que se modelen en JSD serán siempre aquellas en que la dimensión del tiempo es de central importancia, es decir, las actividades que los componen llevan un orden estricto en el tiempo, y ésta es en esencia la característica fundamental de las aplicaciones para las cuales JSD será empleado.

Una vez que se cuente con el modelo del mundo real, se generarán las especificaciones de las funciones del sistema basadas en ese modelo. En JSD, una especificación debe ser escrita en términos de los usuarios, estructurada como su mundo lo está y no en estricta concordancia con las necesidades de la implantación en la computadora, pero al mismo tiempo, ésta será en principio directamente ejecutable.

La decisión de usar una base de datos o un arreglo de registros, la descomposición del sistema en módulos, subprogramas y tareas, que generalmente se identifican con la etapa del diseño y éste con las especificaciones del sistema, en JSD son actividades dadas en la fase de implantación. La planeación y programación de los procesos es el problema central de la implantación en JSD.

2. PASOS DEL DESARROLLO

El proceso definido por JSD para el desarrollo de sistemas lo podemos esquematizar de la siguiente manera:

- a) Desarrollo de las Especificaciones
 - a.1) Especificar el Modelo de la Realidad
 - a.1.1) Describir la realidad en forma abstracta
 - Etapa Acción-Entidad
 - Etapa Estructura-Entidad
 - a.1.2) Definir un Modelo Realizado
 - Etapa del Modelo Inicial
 - a.2) Especificar las Funciones del Sistema
 - Etapa de Funciones
 - Etapa de Sincronización del Sistema
- b) Desarrollo de la Implantación
 - Etapa de la Implantación

Los niveles más bajos de la tabla representan los pasos a seguir en JSD, o sea:

a) Etapa Acción-Entidad.- En ésta, el desarrollador define el área de interés del mundo real, listando las entidades y las acciones que le concernen al sistema.

b) Etapa Estructura-Entidad.- Las acciones realizadas o sufridas por cada entidad, son arregladas por su orden en el tiempo. Este orden se representa a través de diagramas.

c) Etapa del Modelo Inicial.- Aquí, la descripción de la realidad en términos de entidades y acciones, es realizada en un modelo del proceso, así como con las debidas conexiones entre el mundo real y el modelo.

d) Etapa de Funciones.- Son especificadas las funciones para producir las salidas del sistema, así como los procesos adicionales que han sido sumados a las especificaciones como necesarios.

e) Etapa de Sincronización del Sistema.- Aquí el diseñador considerará ciertos aspectos para la sincronización de los procesos en el tiempo.

f) Etapa de Implantación.- En ésta, el desarrollador

considera que equipo y software, es o deberá ser provisto para ejecutar el sistema, así como las técnicas de transformación y programación junto con la definición de la base de datos que permitirá al sistema ejecutarse eficientemente.

3. ETAPA ACCION-ENTIDAD

Este es el paso inicial para el desarrollo de un sistema con JSD. En este punto el desarrollador sólo cuenta con una idea general de cual será el propósito o materia del sistema.

En la etapa de Acción-Entidad se inicia por definir la materia objeto, describiendo el mundo real del sistema en términos de las entidades y las acciones que éstas realizan o sufren. El desarrollador debe considerar a toda la gente, objetos y organización, las cuales pueden ser tomadas como entidades. El significado de la palabra "entidad" en JSD está asociada con la idea del proceso y con la idea de un grupo ordenado de acciones y por lo tanto diferente al significado usado en métodos de base de datos, donde una entidad puede no tener acciones en el mundo real. En JSD no existen entidades a menos de que éstas realicen o sufran acciones, por lo que todos los eventos que ocurran pueden ser considerados como acciones.

Para poder definir entidades y acciones podemos tomar en cuenta los siguientes criterios generales:

a) Criterios Generales Para Definir Acciones.-

- Una acción debe ser considerada como aquella que ocupa un punto en el tiempo, en vez de extenderse en un periodo en el tiempo.
- Una acción debe tomar un lugar en el mundo real fuera del sistema y no debe ser una acción del sistema mismo.
- Una acción debe ser considerada como atómica, y no puede ser descompuesta en subacciones.

b) Criterios Generales Para Definir Entidades.-

- Una entidad debe realizar o sufrir una acción guardando un orden en el tiempo.
- Una entidad debe existir en el mundo real fuera del sistema y no debe ser parte del mismo, o producto de éste.
- Una entidad debe ser considerada como individual, y si existe más de una entidad de un tipo, deben tener un nombre exclusivo cada una.

Si nos sujetamos a estos criterios generales, seremos libres de seleccionar cualquier entidad y acción que cubra el criterio para el cual el sistema se ha desarrollado. Es importante que el desarrollador incluya en la lista todas aquellas entidades y acciones que el sistema necesitará para producir o usar información referente a ellas. Se había hablado anteriormente de que el modelo representaría sólo una parte de la realidad, pues es esta lista de entidades y acciones la que nos especificará la parte de la realidad que se tomará en cuenta.

Adicionalmente a la lista de las acciones, es necesario darle a cada acción una descripción informal que nos diga que es lo que ocurre en el mundo real. Esta descripción será asociada al nombre que identifique a dicha actividad, por ejemplo, en una institución bancaria existe el servicio de cajeros automáticos a través de los cuales un cuentahabiente puede retirar efectivo. En este caso la entidad será el "cuentahabiente", pues es el que ejecutará la operación, y la acción será el "retiro" del efectivo, ahora bien, se nos pide dar una descripción informal de esa acción que bien podría ser algo como "sacar dinero de la cuenta bancaria". Una vez que se tiene identificadas a la entidad, a la acción y a su breve descripción, el desarrollador listará los atributos de la acción que no son otra cosa más que los datos necesarios para llevar a cabo dicha acción, siendo así que para nuestro ejemplo manejaríamos los atributos de NIP (Número de Identificación Personal con el cual se identifica el cuentahabiente en el sistema), y monto del retiro por supuesto. Una vez definidos la entidad y su acción tenemos que el sistema del ejemplo sólo será capaz de manejar retiros de efectivo por parte del cuentahabiente porque sólo fue esa la parcialidad de la realidad que se ha tomado en cuenta, pero si se quisieran inclu-

ir otras funciones como las de pagos de servicios, pago de tarjeta de crédito, manejo de valores, etc., deberíamos incluirlas como actividades propias de nuestro modelo con sus respectivas descripciones y atributos. Por el momento en el primer paso de la descripción abstracta de nuestra realidad, tendríamos lo siguiente:

ENTIDAD: Cuentahabiente
ACCION: Retiro
DESCRIPCION DE LA ACCION
RETIRO: Sacar dinero de la cuenta bancaria
atributos.- NIP, monto

Si se está preguntando porqué, a las acciones se les da una descripción y a la entidad sólo se le menciona, la respuesta es que a la entidad se le describe a partir de las acciones que ésta realiza o sufre.

Para hacer más realista y explícito el desarrollo de los temas, a partir de aquí iniciaremos un ejemplo un poco más completo para que el lector vaya identificando y ligando las diferentes etapas de JSD.

Imaginemos a la Compañía "LO VENDO BARATO S.A.", que se dedica a comercializar diferentes productos. Los clientes por lo regular hacen sus pedidos por teléfono, pero algunas veces por otros medios, como vía correo o visitando personalmente las instalaciones de la Compañía. Existe como regla el conocer que ordenes separadas son requeridas para productos separados.

Los clientes en ocasiones corrigen sus ordenes modificando las cantidades o la fecha requerida y en ocasiones las llegan a cancelar.

La Compañía emplea a un dependiente cuyo trabajo es el de tratar con los clientes, así como de asignar las existencias disponibles a las ordenes. El puede acceder la información correspondiente a las existencias disponibles de cada producto telefoneando al almacén, de donde recibirá los datos con aproximada realidad.

Lo que se desarrollará sólo será el sistema de ventas manejando las ordenes de los clientes. El propósito del sis-

tema será el de manejar la asignación de las existencias de los productos a las ordenes.

Analizando un poco el problema, podemos llegar a la conclusión de que nuestro modelo abarcará lo siguiente:

ENTIDADES: cliente, dependiente, orden, producto.

ACCIONES: ordenar, corregir, cancelar, retrasar, asignar, entregar.

DESCRIPCION DE LAS ACCIONES:

ORDENAR: Llevar una orden a la compañía para su asignación y entrega. Es una acción del cliente y de la orden
Atributos: folio-orden, producto-id, cantidad, fecha.

CORREGIR: Cambiar la cantidad o la fecha requerida de una orden. El producto no lo pueden cambiar. Es acción del cliente y de la orden.
Atributos: cifra (nueva cantidad o fecha), cantidad o fecha, folio-orden.

CANCELAR: Cancelar una orden. Acción del cliente y de la orden.
Atributos: folio-orden.

RETRASAR: Retardar una orden por no haber existencias disponibles del producto para asignarle. Acción del dependiente y de la orden.
Atributos: folio-orden.

ASIGNAR: Asignar la existencia de un producto a una orden. Acción del dependiente, orden y producto.
Atributos: cantidad.

ENTREGAR: Entregar el producto ordenado al cliente. Acción del dependiente, el producto y el cliente.
Atributos: cantidad y fecha.

Este resultado de la etapa acción-entidad es tentativo. Nosotros hemos hecho un primer esquema burdo de las fronteras del modelo, por ejemplo, concientemente se excluyó la reordenación de productos a los proveedores, así como el recoger la

existencia del almacén para su entrega. La lista de entidades y acciones debe ser reconsiderada en los siguientes pasos de la creación del modelo.

Para hacer la lista, nosotros necesitamos estudiar el mundo real teniendo en cuenta el propósito general del sistema, podemos estudiar el mundo real entrevistando a los usuarios, leyendo reportes u otros estudios, observando directamente el mundo real, etc.

El peligro mas común en esta etapa es el de saltar a las conclusiones sin hacer un estudio adecuado, pues aunque lo que se obtenga como resultado en este paso sea poco, sólo una lista de entidades y acciones, su descripción y argumentos, será el fundamento de nuestro modelo en la realidad.

Una forma comprensiva es el de listar todos los nombres sustantivos y verbos que encontremos en nuestro estudio de la realidad, pues cada nombre sustantivo es una posible entidad y cada verbo una posible acción, una vez que se cuente con la lista, se hará un escrutinio seleccionando sólo aquellos que estén dentro de los límites marcados para el sistema y rechazando los demás. Al final sólo se deberá contar con una lista pequeña y sencilla que describa claramente la parte de la realidad que el modelo abarcará.

Aún cuando la lista se haya basado en un estudio cuidadoso, sólo será tentativa pues es común que sea modificada durante los pasos de estructura-entidad y modelo inicial, algunas modificaciones pueden ser incluidas por:

- a) Rechazar una acción en la etapa de estructura-entidad ya que podría pertenecer más al sistema que al mundo real en sí.
- b) Rechazar una acción en la etapa de modelo inicial ya que puede no ser económicamente conveniente para el modelo.
- c) Sumar una entidad en la etapa de estructura-entidad por ser necesario para especificar el orden de las acciones de una entidad existente.
- d) Sumar una acción en la etapa de modelo inicial por ser necesaria para la interpretación de las acciones existentes.

Cuando nosotros seleccionamos o descartamos posibles entidades, estamos haciendo dos tipos de elecciones simultáneamente. Estamos escogiendo que cosas y que personas deberán ser contempladas en nuestra descripción abstracta de la realidad, así como la forma en que deben ser clasificadas.

Primero, un tipo de entidad es especificado en términos de las posibles acciones realizadas o sufridas, así como su orden.

Segundo, especificamos el tiempo de vida de una entidad, extendiéndose dentro de los límites del sistema; por ejemplo en una nómina el tiempo de vida de un empleado será desde su contratación hasta su renuncia o despido.

Una acción puede ser común a dos entidades, en el sentido de que las dos toman parte en un mismo evento, si dos autos chocan, es una acción común de dos entidades del mismo tipo, pero por otro lado, una cancelación de una orden de compra es común a dos tipos de entidades.

La decisión de considerar o no una acción común para varias entidades es nuestra en el sentido de la forma elegida para describir la realidad. Dentro de los límites somos libres de decidir la forma de apreciar la realidad. Podemos describir dos acciones como distintas cuando en la realidad sea una sola, en el caso de la cancelación, una entidad ejecuta la acción y la otra la sufre.

4. ETAPA ESTRUCTURA-ENTIDAD

Puesto que las acciones de una entidad se dan con un orden en el tiempo, necesitamos expresar ese orden. Aunque en ocasiones una entidad parece que realiza o sufre acciones sin un orden específico, por lo regular encontraremos restricciones en su ordenamiento.

En JSD esas restricciones en el ordenamiento de las acciones son representadas diagramáticamente por lo que el resultado de este paso será un grupo de diagramas mostrando el orden de las acciones de cada entidad.

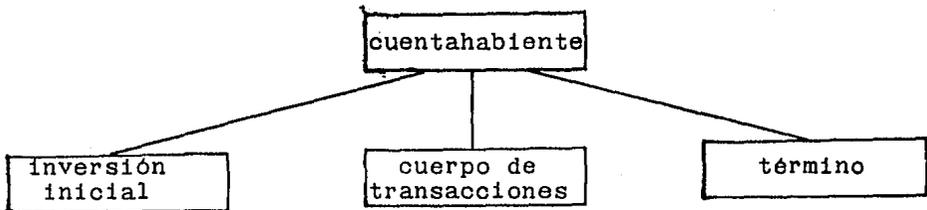
Cabe mencionar que la notación diagramática utilizada

difiere de los "diagramas de flujo" comúnmente utilizados, por lo que es necesario dejar atrás los prejuicios creados para la mejor comprensión de lo aquí expuesto, además de estar conciente de que lo que se diagramará no es la estructura de un sistema o base de datos, sino el orden en que una entidad sufre o realiza una acción.

Algunas veces encontraremos que para una entidad requeriremos más de un diagrama, y estos diagramas adicionales nos sugerirán entidades que tal vez no hayan aparecido en la lista original.

Un requerimiento fundamental al especificar las estructuras de las entidades es el de abarcar el tiempo de vida completo de la entidad del mundo real y no sólo una parte. Esta tentación de sólo especificar una parte del tiempo de vida de una entidad es particularmente fuerte cuando el desarrollador pone atención prematura a la implantación del sistema.

4.1 ESTRUCTURA DE LOS DIAGRAMAS.- Los diagramas utilizados en JSD son árboles, observemos el siguiente diagrama:



La localidad con el título cuentahabiente es llamada la raíz del árbol, las localidades con los títulos inversión inicial, movimientos y término son llamados las hojas del árbol.

La característica fundamental de un árbol es que existe un sólo camino de la raíz a cualquier hoja.

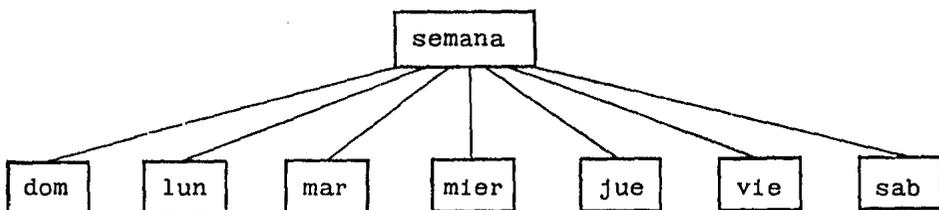
En los diagramas creados, cada hoja deberá ser una

acción, así que las acciones del cuentahabiente, llamadas inversión inicial, movimientos y término, son hojas del árbol y este mostrará como están ordenadas las acciones.

Si de una localidad dependen una o más localidades conectadas directamente en un nivel inferior, se les llamará "Padre" e "Hijos" respectivamente. Si a los hijos se les pone una marca en la esquina superior derecha, indicarán que su padre es una iteración, una selección o una secuencia. Cada localidad dentro de un diagrama que no sea una acción deberá ser cualquiera de esas tres. Las localidades que representan acciones son llamadas "elementales", así como los hijos de un padre son llamados "sus partes".

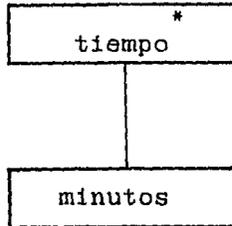
4.2 SECUENCIAS.- Una secuencia es indicada al dejar a las localidades hijas sin marca alguna. El significado de una secuencia es el de que para cada instancia de ésta se ocurrirán una sola vez cada uno de sus hijos de izquierda a derecha. No hay posibilidad de que algún hijo sea omitido o que ocurra más de una vez.

Una secuencia puede tener cualquier número de hijos, pero es necesario enumerar cada parte de la secuencia, esto es, se deben especificar cada uno de sus hijos con sus respectivos nombres, ejemplo:



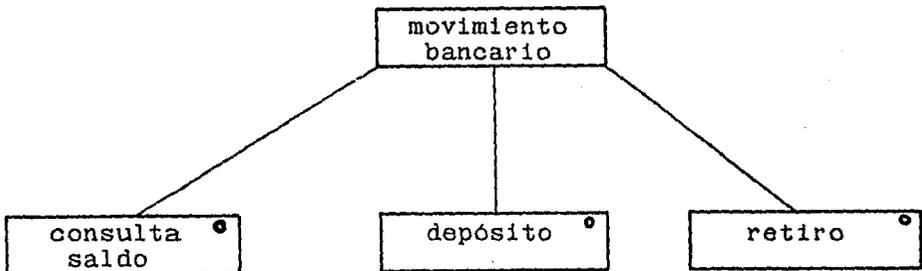
4.3 ITERACIONES.- Una iteración se indica marcando la localidad hija con un asterisco en la esquina superior derecha, por lo tanto, siempre existirá una localidad hija para cada iteración llamada "parte de la iteración".

El número de ocurrencias de la parte de una iteración debe ser limitada, pero estos límites no son mostrados en los diagramas de JSD, sino que se añaden notas al texto de la estructura donde sea necesario, ejemplo :



4.4 SELECCIONES.- Una selección es indicada marcando cada localidad hija por un pequeño círculo en su esquina superior derecha. El significado de una selección es el que por cada ocurrencia de la selección misma, ocurrirá uno de sus hijos sin importar su orden.

Una selección, como una secuencia, puede tener cualquier número de hijos y a diferencia de aquella, una selección no puede tener sólo un hijo. Ejemplo:



4.5 ESTRUCTURAS MINIMAS.- Al diagramar las estructuras de las entidades, debemos especificar las restricciones en el posible orden de las acciones en el mundo real. La estructura debe ser diferente, y aunársele a ésta las notas adicionales necesarias para expresar esas restricciones, pero no debe ser elaborado para servir a otros propósitos y en particular, no debe ser elaborado con requerimientos funcionales en mente.

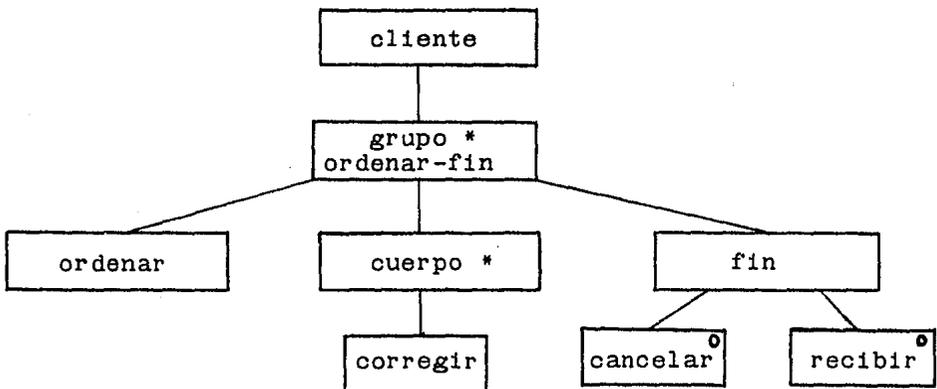
Debemos de diagramar la estructura más simple del mundo real en el paso estructura-entidad. Un diagrama más elaborado y preciso será el que hagamos en el paso de funciones.

4.6 ESTRUCTURAS MARSUPIALES.- Algunas veces encontraremos que a alguna entidad de la lista que generamos, pertenecen otras entidades que por el momento no puedan existir por separado, pero que al llegar al paso de estructura-entidad emerjan como entidades autónomas y a éstas se les denomina marsupiales. Para una mejor aclaración vayamos a nuestro ejemplo "LO VENDO BARATO S.A." y listemos las acciones de la entidad cliente que habíamos ya de terminado en el paso anterior:

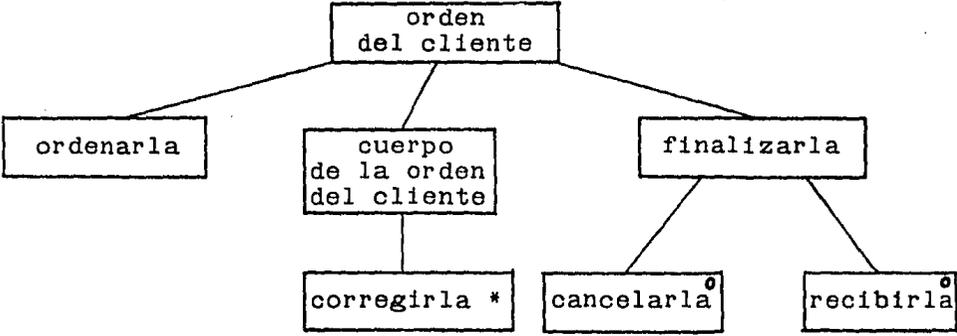
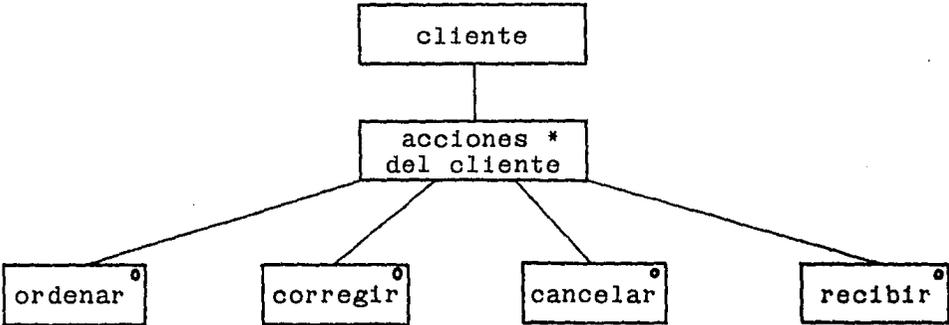
ENTIDAD: Cliente

ACCIONES: Colocar una orden, corregir esa orden, cancelarla, recibir la entrega.

Si creáramos el diagrama representativo de las acciones de esta entidad, tendríamos:



Analizándolo encontraremos que un cliente puede ubicar una orden, corregir esa orden cuantas veces sea necesario y cancelar o recibir la orden pedida, y todas estas acciones, desde ordenar hasta fin, las puede realizar "n" número de veces, pero sólo una orden a la vez. Nosotros habíamos dicho que un cliente debe hacer una orden por cada tipo de producto que él requiriese, por lo que con el diagrama anterior lo estamos restringiendo a que haga una orden siempre y cuando la anterior ya haya finalizado, pero eso no es cierto sino que un cliente puede tener varias ordenes en proceso al mismo tiempo, por lo que se hace necesario que esta entidad se divida en dos: el cliente y su orden, para así poder crear los siguientes diagramas:

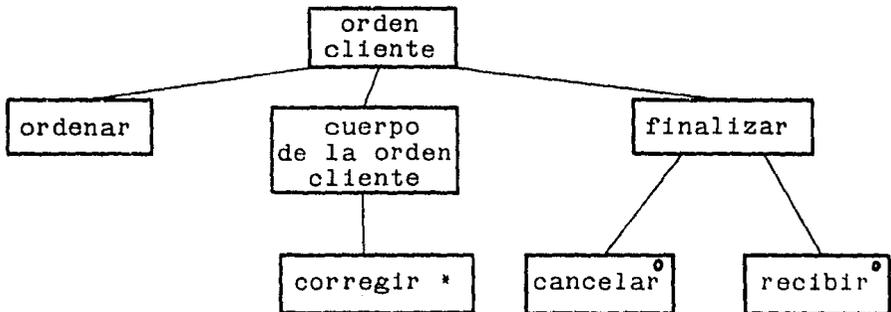
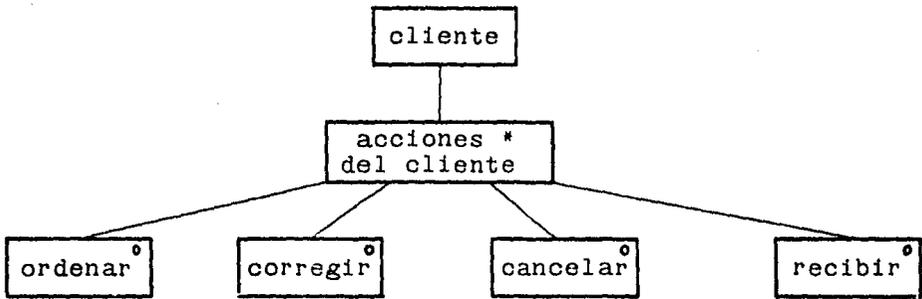


De esta forma el cliente queda en libertad absoluta de hacer varias ordenes y manejarlas al mismo tiempo, y cada orden seguirá su ciclo de vida normal como entidad nueva.

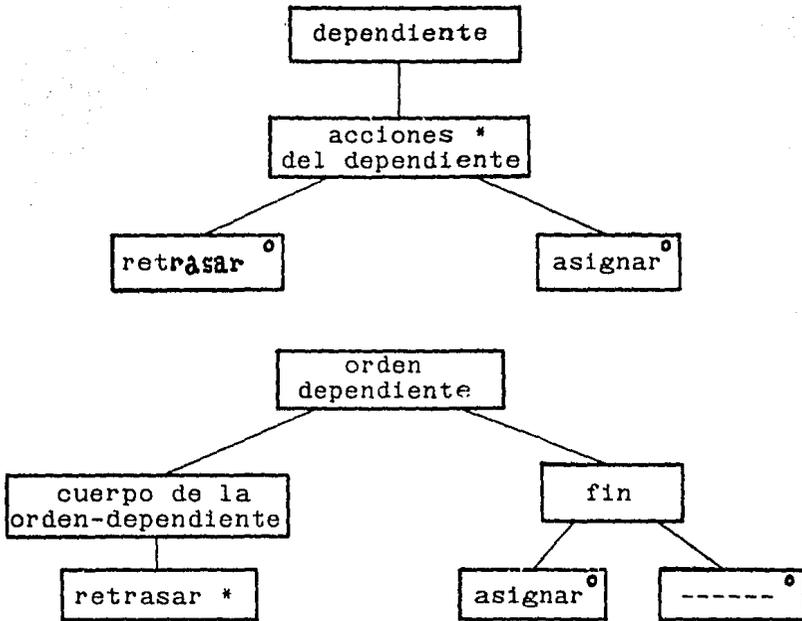
4.7 ESTRUCTURA-ENTIDAD DE LA CIA. "LO VENDO BARATO S.A.".- Siguiendo la lista inicial, tenemos que las acciones por entidad son:

- CLIENTE: Ordenar, corregir, cancelar, entregar.
- DEPENDIENTE: Retrasar, asignar.
- ORDEN: Ordenar, corregir, cancelar, entregar, retrasar, asignar.
- PRODUCTO: Asignar, entregar.

La estructura de la entidad cliente presenta una entidad marsupial, como ya lo analizamos en el punto anterior, que es la orden del cliente, por lo que tenemos los siguientes diagramas:



La estructura del dependiente es similar a la del cliente con su respectiva entidad marsupial:



No debemos perder de vista que en estos primeros pasos de JSD estamos desarrollando una especificación y que una vez que un cliente ha colocado una orden, él debe eventualmente modificarla, cancelarla o participar en la acción de la entrega del producto, no hay posibilidad de que el dependiente pueda cancelar la orden, sólo hemos especificado que él puede asignar los productos al final de la vida de la orden o una acción nula, así mismo por ejemplo, no es necesario que el dependiente busque la aceptación por parte de la Compañía de la cancelación hecha por un cliente.

La entidad orden es representada por una estructura compuesta formada por las estructuras respectivas de las entidades marsupiales ORDEN-CLIENTE y ORDEN-DEPENDIENTE. Para estar seguros que ésta composición sea llevada a cabo

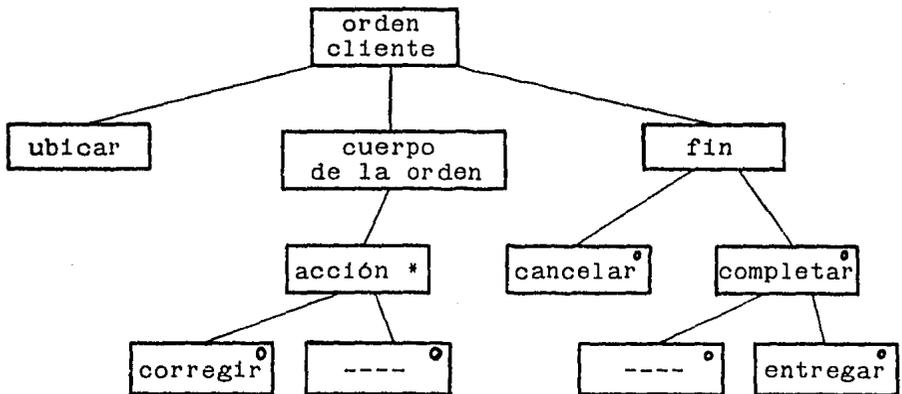
correctamente, debemos de satisfacer las siguientes condiciones para la simplificación de estas estructuras:

- a) Ningún caso posible de la entidad orden infringe las restricciones de la secuencia de las acciones: ubicar, corregir, cancelar y entregar. Impuestas por la ORDEN-CLIENTE o las restricciones en la secuencia de retrasar y asignar de ORDEN-DEPENDIENTE.
- b) Cualquier restricción impuesta por orden, de tal manera que se den ORDEN-CLIENTE y ORDEN-DEPENDIENTE, será una restricción válida en el mundo real.

Comenzaremos por revisar la primera condición de ORDEN-CLIENTE y ORDEN-DEPENDIENTE tomando en cuenta lo siguiente:

- a) Reemplazar por un nulo (---) aquellas hojas de ORDEN que no sean relevantes en la estructura.
- b) Aplicar sucesivamente cualquier operación permisible hasta que la estructura con la interrelación sea obtenida.

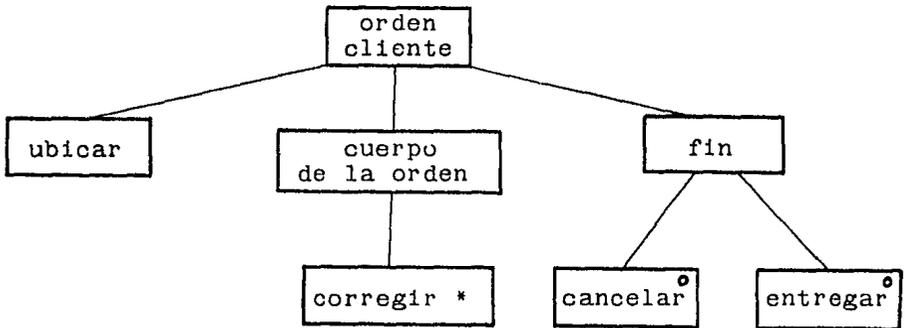
Tomando ORDEN-CLIENTE como la estructura a sintetizar y cuya representación aparece abajo:



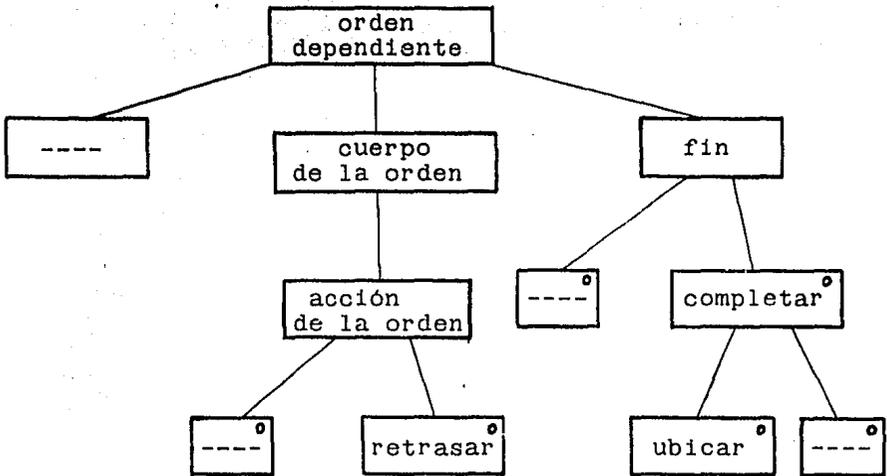
Las operaciones de síntesis permitidas son aquellas que no cambiarán las restricciones en el orden de las hojas de la estructura, y éstas incluyen:

- a) Remove una hoja nula que sea parte de una secuencia
- b) Remove una hoja nula que es parte de una selección, la cual es, en sí misma, parte de una iteración.
- c) Reemplazar una selección de una parte por ésta misma
- d) Reemplazar una secuencia de una parte por ésta misma.
- e) Remove una de dos partes idénticas en una selección
- f) Remove una de dos iteraciones idénticas que sean partes consecutivas de una secuencia.
- g) Reducir dos niveles de iteración a uno. Esto es, reemplazar B por C, donde A es una iteración de B, y B una iteración de A.

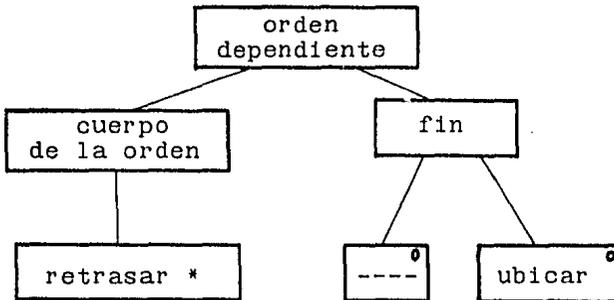
Si aplicamos las operaciones permisibles (a) y (b) de la lista anterior a la hoja COMPLETAR de nuestra estructura y (b) y (c) a ACCION DE LAS ORDENES, obtendremos la estructura siguiente, que es igual a la anterior excepto por el nombramiento de los componentes no-elementales de ORDEN-CLIENTE:



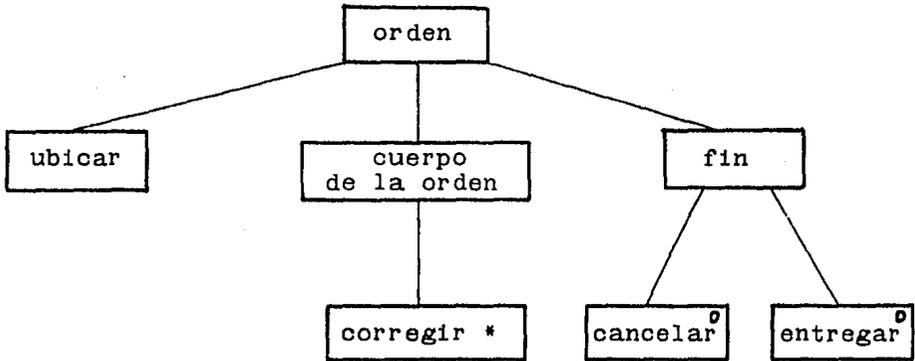
Ahora, sintetizémos la estructura ORDEN-DEPENDIENTE. Su representación se muestra en la siguiente figura:



Si aplicamos la operación (a) a ORDEN, (a) y (d) a COMPLETAR, y (b) y (c) a ACCION DE LA ORDEN, obtendremos la siguiente estructura sintetizada:



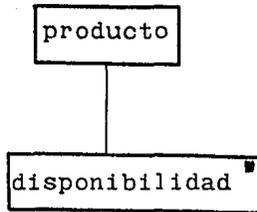
4.8 ESTRUCTURA-ENTIDAD-2 "LO VENDO BARATO S.A." .- En un sistema realista, seguramente se automatizarían las acciones de "asignar" y "retrasar" ordenes. La automatización significaría que las acciones de "asignar" y "retrasar" del dependiente, vendrían siendo parte de las funciones del sistema, por lo que el dependiente dejaría de ser una entidad en el modelo. Puesto que las acciones de "asignar" y "retrasar" son ahora parte del sistema tendrían que desaparecer de la estructura de la orden, la cual quedaría como la estructura marsupial ORDEN-CLIENTE.



La entidad cliente queda inafectada, a menos que la automatización de la asignación de la orden, requiera un comportamiento diferente por parte del cliente.

Si automatizamos la actividad del dependiente, es necesario modelar la entidad PRODUCTO. Nuestro ejemplo ya contaba con un sistema de control de inventarios, por lo que para nosotros la entidad no son los productos físicos dentro del almacén sino el registro de sus existencias dadas por el sistema de inventarios. Si consideramos ésto, podemos modelar el comportamiento de la entidad PRODUCTO en una sola acción:

Disponibilidad: Es la cantidad de productos disponibles para su asignación a una orden.
atributos: cantidad, ...



5 ETAPA DEL MODELO INICIAL

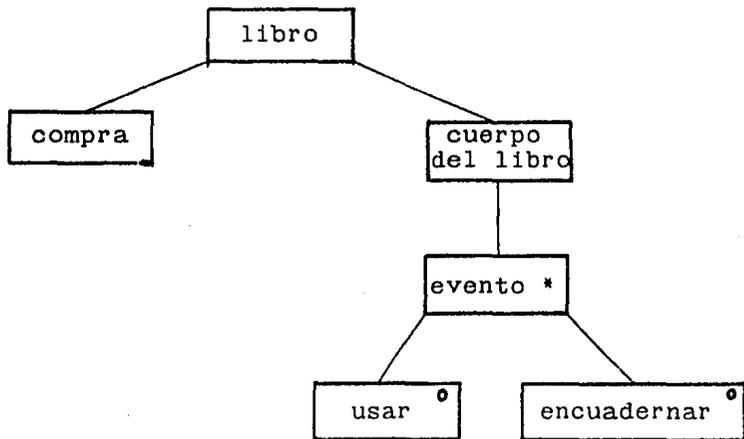
En las etapas de acción-entidad y estructura-entidad se hizo una descripción abstracta del mundo real en términos de entidades y el orden de las acciones que sufrían o realizaban. En la etapa del modelo inicial, empezaremos a construir nuestro sistema especificando un conjunto de procesos secuenciales que modelarán a las entidades del mundo real y su funcionamiento. Donde exista la estructura de una entidad en su descripción abstracta, existirá un proceso secuencial en el modelo.

Para asegurarnos de que el funcionamiento del modelo refleje los eventos del mundo real, necesitamos conectarlo al mundo real directa o indirectamente. Una vez conectados al mundo real los modelos secuenciales del modelo son, en principio, programas ejecutables por la computadora. No podremos ejecutarlos directamente, pues tenderá a poseer características que se alejan de algún lenguaje de programación en común.

5.1 TEXTO DE LA ESTRUCTURA .- Para muchos propósitos necesitaremos una notación especial para especificar los procesos secuenciales. JSD se refiere como texto de estructura, a un pseudocódigo que nos permita transcribir los diagramas a un proceso secuencial que se apegue más a lo que fuese un lenguaje de programación para su fácil codificación posterior al momento de implantarlo en una computadora.

A continuación especificaremos las reglas que debemos tomar en cuenta para poder representar una estructura con el pseudocódigo propuesto.

Primeramente, imaginémos que estamos especificando la vida de un libro en la biblioteca en donde sólo abarcamos las actividades de comprar, usar y encuadernar. La acción de compra es realizada cuando es comprado el libro inicialmente, la acción de usar es ejecutada cuando el libro es dado a uno de los miembros de la biblioteca para su lectura, (aquí asumiremos que siempre será devuelto al término del plazo dado al lector para no complicar nuestro ejemplo) y la acción de encuadernar es realizada cuando el libro es mandado a reencuadernar. Tomando en cuenta lo anterior, tendremos la siguiente estructura de la entidad libro:



A continuación mostramos el texto de la estructura correspondiente al diagrama anterior:

```

LIBRO sec
  COMPRA
  CUERPO DEL LIBRO itr
    EVENTO sel
      USAR;
    EVENTO alt
      ENCUADERNAR;
    EVENTO fin
  CUERPO DEL LIBRO fin
LIBRO fin.
  
```

Como se puede inferir fácilmente de este ejemplo, las hojas de los diagramas son representadas por sus nombres seguidos de punto y coma (;), una secuencia formando un bloque en cuya parte inicial se escriben el identificador de la secuencia y la abreviatura "sec", cerrando el bloque se pone el identificador de la secuencia mas la palabra "fin":

```
LIBRO sec
```

```
.
```

```
.
```

```
LIBRO fin
```

Es importante notar desde éste momento, que cuando abrimos, en este caso una secuencia, su finalización debe estar al mismo nivel que su inicio y todo lo que contenga esa secuencia deberá estar a un nivel mas interno representar que se encuentra contenido en un nivel mayor:

```
A sec  
  B sec
```

```
.
```

```
.
```

```
  B fin
```

```
A fin.
```

Una iteración se representa, al igual que una secuencia, con un bloque delimitado por un inicio y un fin, con la diferencia de que en el inicio el identificador estará seguido por la abreviatura "itr" :

```
CUERPO DEL LIBRO itr
```

```
.
```

```
.
```

```
CUERPO DEL LIBRO fin
```

Una selección será representada también con un inicio constituido por el identificador de la selección más la abreviatura "sel". Un fin y las alternativas que contenga entre éstos dos, formadas por el identificador de la selección más la abreviatura "alt" y debajo de cada una de éstas, en un nivel mas interno el identificador de la opción abarcada por esa alternativa. Así entonces una selección de dos opciones será representada así:

```
EVENTO sel
      USAR;
EVENTO alt
      ENCUADERNAR;
EVENTO fin.
```

Nótese que se apega a la estructura que nosotros conocemos como "IF-THEN-ELSE" .

Una selección de más de dos opciones la podemos representar como sigue:

```
DECIDIR sel
      UNO;
DECIDIR alt
      DOS;
DECIDIR alt
      TRES;
      .
      .
DECIDIR fin.
```

Cuando se desee escribir una condición dentro del texto de la estructura, deberemos escribirla después del encabezado de la iteración o de la parte de la selección a que corresponda:

```
LOOP itr mientras(sw=apagado)
      .
      .
      .
LOOP fin.
```

El significado de esta operación es el convencional de una figura "DO-WHILE", o sea, que las operaciones que se encuentran entre el inicio y el fin de la iteración se ejecutarán repetidas veces mientras la condición "sw=apagado" sea verdadera.

En el caso de una selección aparecería como sigue:

```
CASO sel (a=b)
    IGUAL;
CASO alt (a>b)
    MAYOR;
CASO alt (a<b)
    MENOR;
CASO fin.
```

En ciertos casos la condición de la última alternativa de la selección, se puede validar en forma lógica si ninguna de las anteriores fuese verdadera, como en el caso del ejemplo anterior, si "a" no es igual a "b" y "a" no es mayor que "b", entonces por deducción lógica "a" será menor que "b":

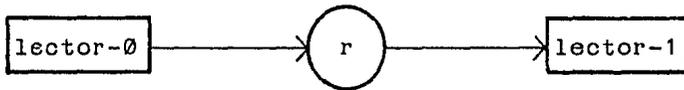
```
CASO sel (a=b)
    IGUAL;
CASO alt (a>b)
    MAYOR;
CASO alt (sino)
    MENOR;
CASO fin.
```

5.2 CONEXION DE PROCESOS.- En las etapas de modelo inicial y función de JSD, existen dos formas para conectar procesos: la conexión de flujo de datos y la conexión de vector-estado. En la conexión de flujo de datos, un proceso escribe una serie de datos secuencial consistente en un conjunto de mensajes o registros y el otro proceso leerá ésta serie. En la conexión de vector-estado, un proceso inspecciona directamente el vector-estado, que no es otra cosa sino un área de memoria o un conjunto de variables internas perteneciente al otro proceso.

Nosotros podremos utilizar ambos tipos de conexiones para conectar los procesos del modelo entre sí o con el mundo real.

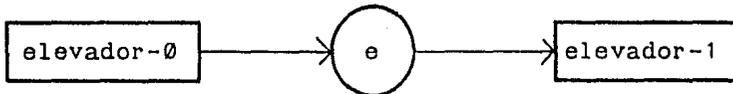
Podemos considerar la tarea de conectar el proceso del modelo al mundo real como el conectar dos procesos secuenciales entre sí. Nos referimos a los procesos del mundo real con la terminación "-0" (e.j. lector-0) y al proceso que lo modela dentro del sistema con la terminación "-1" (e.j. lector-1).

Para poder representar diagramáticamente la conexión entre diferentes procesos, se usa un diagrama de especificación de sistemas o SSD (System Specification Diagram). En el SSD, la conexión de flujo de datos se representa así:



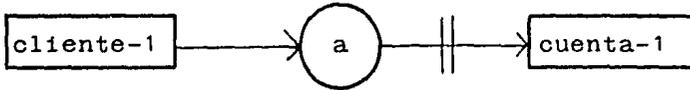
Los rectángulos en un SSD representan procesos secuenciales y los círculos un flujo de datos, las flechas indican la dirección por la cual la información se mueve. En el SSD anterior el lector-0 escribe un flujo de datos que es leído por el lector-1.

La conexión vector-estado se representa en la forma siguiente:



De nuevo los rectángulos representan procesos. El diamante indica que la conexión es por la inspección del vector-estado, en éste caso el elevador-1 inspecciona el vector-estado del elevador-0.

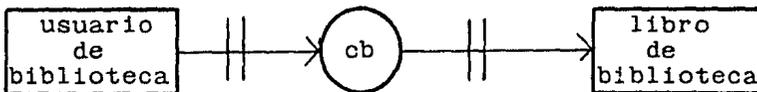
Las conexiones mostradas anteriormente tienen una relación de uno-a-uno, esto es, existe un proceso en cada lado de la conexión. Algunas veces una conexión puede tener una relación muchos-a-uno y esta multiplicidad se representa poniendo dos líneas paralelas entre sí y perpendiculares a la vez a la flecha del lado donde existan muchos procesos:



El SSD anterior representa que existen muchos procesos cuenta-1 conectados al proceso cliente-1. Esta multiplicidad la podemos interpretar como sigue:

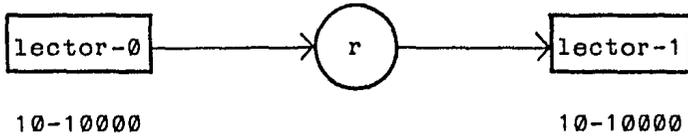
- a) A través de la vida de un proceso cliente-1 éste puede ser conectado a cualquier número (incluyendo 0) de procesos cuenta-1 y
- b) En cualquier tiempo, un proceso cliente-1 puede estar conectado a 0, 1, o más procesos cuenta-1.

Con la misma interpretación de multiplicidad, conexiones con una relación de muchos-a-muchos también son permitidas en un SSD:



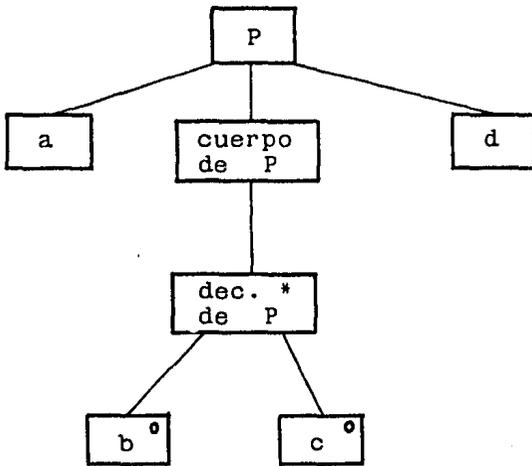
El SSD anterior nos representa que cada usuario de biblioteca está conectado a muchos libros de biblioteca y viceversa.

La multiplicidad representada por dos líneas es relativa, pues no nos dice nada acerca de cuantos procesos están conectados. Para mostrar una multiplicidad absoluta, podemos escribir los límites inferior y superior debajo de cada proceso:



Pero la multiplicidad absoluta no es de mucha importancia sino hasta la fase de implantación.

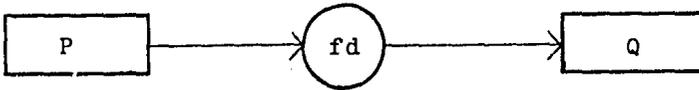
5.3 CONEXION DE FLUJO DE DATOS .- Supongase que se tiene un proceso secuencial P cuyo diagrama y texto son los siguientes:



```

P seq
a;
CUERPO DE P itr
DEC. DE P sel
  b;
DEC. DE P alt
  c;
DEC. DE P fin
CUERPO DE P fin
d;
P fin.
  
```

Imaginemos ahora que queremos construir otro proceso Q para modelar el funcionamiento de P en alguna manera. Inicialmente supondremos que Q es el modelo de la función de P exactamente, esto es, Q realiza las acciones Qa, Qb, Qc y Qd en el mismo orden que P realiza a,b,c y d respectivamente. Podremos conectar P y Q por un flujo de datos "fd" :

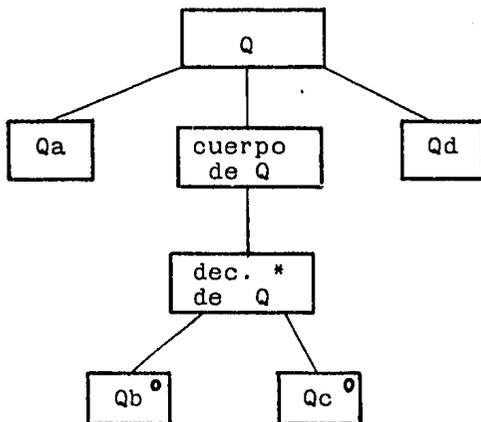


Podremos señalar que cada acción de P es dada a Q por un registro de "fd" : la acción "A" es dada por el registro "fA" y así sucesivamente. Estos registros deben ser escritos por P cuando la acción señalada sea realizada.

```

P  seq
  A; escribe fA a fd
  CUERPO DE P itr
    DEC. DE P sel
      B; escribe fB a fd
    DEC. DE P alt
      C; escribe fC a fd
    DEC. DE P fin
  CUERPO DE P fin
  D; escribe fD a fd
P  fin.
  
```

Siendo que Q es el modelo del funcionamiento de P exactamente, debe de tener su misma estructura. La función de Q puede ser descrita informalmente como sigue: "esperar por una señal que llegue de P, en la forma de un registro de fd, y cuando éste realice la acción correspondiente dentro de la estructura de "Q". Esta función es ejecutada por una operación de "lectura fd" y condiciones sumadas al texto de Q. La operación de "lectura fd" espera por la señal de P y la recibe cuando llega; las condiciones guían a Q a través de su estructura:

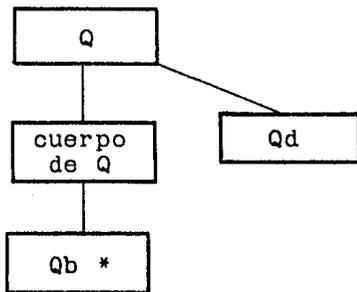


```

Q seq
lectura fd;
Qa; lectura fd
CUERPO DE Q itr
  mientras(fb o fc)
  DEC. DE Q sel (fb)
  Qb; lectura fd;
  DEC. DE Q alt (fc)
  Qc; lectura fd;
  DEC. DE Q fin
CUERPO DE Q fin
Qd;
Q fin
  
```

Hasta el momento sólo hemos considerado que Q modela exactamente el funcionamiento de P por lo que son idénticos, pero una conexión de flujo de datos permite otras relaciones entre P y Q, como sería la posibilidad de que Q modelara sólo una parte de las acciones de P. Imaginemos entonces que Q solo modelará las acciones b y d de P:

- a) En P sólo se ubicarán las operaciones de "escribe" para las acciones b y d.
- b) Por lo tanto Q tendrá la siguiente estructura:



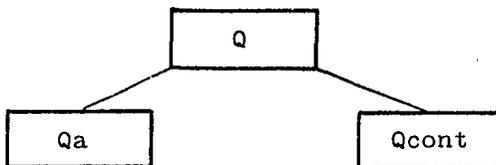
```

Q seq
lectura fd;
CUERPO DE Q itr
  mientras (fb)
  Qb; lectura fd;
CUERPO DE Q fin
Qd;
Q fin.
  
```

Otra posibilidad es la de que P escriba sólo un resumen de su funcionamiento simplificado a fd por ejemplo, la acción de "a" y un contador de las acciones "b". P sería entonces:

```
P seq
a; escribe fa a fd; cont:=0;
CUERPO DE P itr
  DEC. DE P sel
    b; cont:=cont+1
  DEC. DE P alt
    c;
CUERPO DE P fin
  escribe cont a fd;
d;
P fin.
```

Y la figura de Q podrá aparecer como sigue:

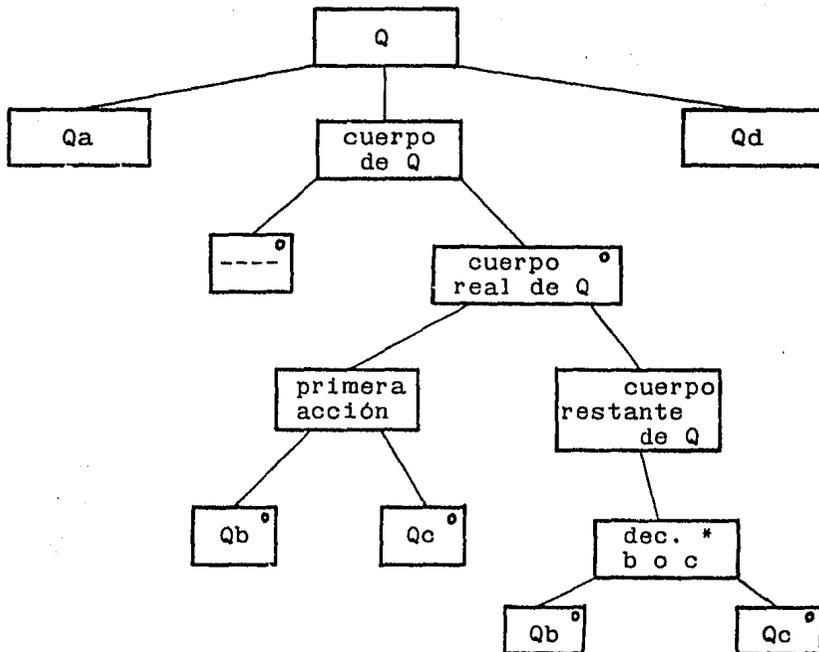


Otra posibilidad es que Q modele todas las acciones de P en una estructura más elaborada. Por ejemplo la estructura de P deberá de distinguir los tres siguientes casos:

- a) Que no se realice la acción b o c .
- b) Que la primera acción después de a sea b.
- c) Que la primera acción después de a sea c.

Entonces la estructura de Q aparecerá como sigue: (Ver el diagrama en la siguiente página).

Hasta este momento podría quedar poco comprensible el concepto de una conexión de flujo de datos, por lo que trataremos de visualizarlo en una forma más simple. Imaginemos al Video Juego "Asteroides" manejado por una persona a través de

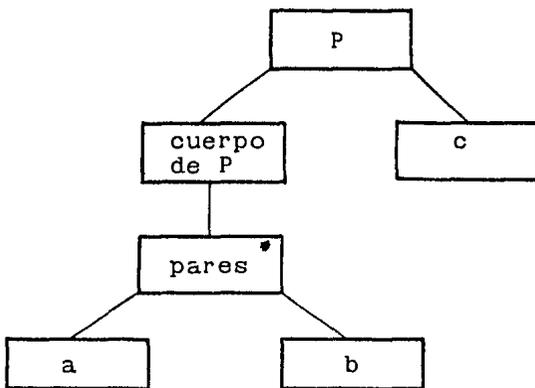


un control, el objeto del juego es el de desintegrar los asteroides que caen mediante los disparos de un cañón laser controlado por el usuario, el cual sólo puede ejecutar tres acciones : disparar, mover-izquierda y mover- derecha. En este caso el usuario que está en el mundo real, manipula el control a través del cual manda la señal de la acción que quiere ejecutar, ésto es que al momento en el que el usuario oprime el botón de disparo "escribe" ésta señal a un flujo

de datos (FD) que es transmitido en línea al proceso que realiza una acción de "lectura" del flujo de datos (FD) y ejecuta la acción correspondiente. En otras palabras es un sistema ON-LINE o en tiempo real. Si trabaja con una terminal, el usuario está conectado vía un flujo de datos al proceso. Pero no sólo nos limitemos a ver una terminal y un proceso, sino que este concepto es más amplio. Imaginemos una presa en la que tenemos un procesador cuyo propósito es el de medir la altura de las aguas y al momento en el que éstas rebasan una altura dada, manda un mensaje a otro procesador que se encarga de controlar las compuertas para que las abra más, o viceversa, en éste caso los dos procesadores están conectados a través de un flujo de datos.

5.4 CONEXION VECTOR-ESTADO .- Primero veamos su concepto en forma general para después pasar a ejemplos específicos.

Supongamos que tenemos un proceso "P" cuya estructura es la siguiente:

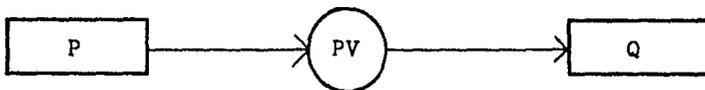


```

P seq
n:=0;
CUERPO DE P itr
  PARES seq
    a; n:=n+1;
    b;
  PARES fin
CUERPO DE P fin
c;
P fin
  
```

No nos preocupemos en este momento sobre cual es el objeto de P ni de sus acciones a, b y c. Observemos que se incluyó la variable "n" la cual cumple con la función de contar las acciones "a" que ocurran.

Supongamos también la existencia de un proceso Q cuya estructura no nos preocupa en este momento, sólo sabemos que Q está conectado con P a través de un vector-estado :



La conexión se da en algún momento en que alguna acción de Q inspecciona el vector-estado de P, que no es otra cosa sino sus variables locales y apuntadores, a través de una operación :

getsv PV o inspecciona PV

Una o más de estas operaciones deberán aparecer en el texto de Q. El resultado de éstas, es que el vector-estado de P sea ubicado en un área accesible para Q. En este caso Q tendría acceso a los valores de "n" y del apuntador a P.

Las características de una conexión vector-estado son las siguientes :

- a) La iniciativa en la comunicación entre los dos procesos depende enteramente a Q, o sea al proceso que invoca la operación getsv PV o inspecciona PV. El proceso P, o sea el que es inspeccionado no puede ejecutar ninguna operación que provoque que Q lo inspeccione.
- b) Las operaciones getsv o inspecciona ejecutadas por Q son determinadas por éste.
- c) Si la ejecución de Q es lenta o se detiene temporalmente, los valores que P pudiera tener en ese lapso se pierden.
- d) Las operaciones getsv o inspecciona no causan un bloqueo de Q o P.

El que un proceso pueda inspeccionar los valores de las variables de otro proceso independiente, puede traer problemas en la consistencia de los datos así obtenidos, esto es una versión microscópica de un problema de consistencia en base de datos. Imaginemos que al proceso P descrito anteriormente, no cuente el número de veces que ocurre la acción "a" sino también guarde el acumulado de su atributo "monto", para lo que tendríamos que incluir la operación correspondiente al sucederse "a" :

```
a; n:=n+1; acumulado:=acumulado+monto
```

Si el proceso P corre a una velocidad y Q en otra, es probable que al darse el getsv desde Q tome al proceso P después de ejecutar el incremento de "n" y antes del incremento de acumulado por lo que los valores que obtendría no fuesen correspondientes, dándose un problema de consistencia en los datos. Para evitar tener problemas de este tipo, podemos imponer ciertas restricciones para que esos valores sean obtenibles en aquellos puntos en la ejecución de P donde:

- a) Q acabe de hacer una operación "lee" a un flujo de datos.
- b) P acabe de hacer una operación "escribe" a un flujo de datos.
- c) Q acabe de hacer una operación getsv a otro proceso.

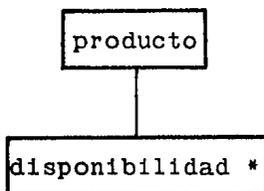
Analicemos el siguiente ejemplo : Un elevador hace parada sólo en aquellos pisos en los que se haya oprimido el botón que lo llama, el elevador puede determinar el piso en el que debe parar en dos formas diferentes:

- a) Al momento de oprimir un botón en un piso dado, se manda una señal al procesador del elevador (escribe la señal a un fd), el elevador recibe la señal (lee la señal del fd) y ejecuta las acciones correspondientes para llegar a ese piso. En éste caso el elevador y el panel de los pisos están conectados con un flujo de datos.
- b) Al momento de oprimir un botón en un piso dado, sólo se enciende un switch. El procesador en el elevador constantemente inspecciona el estado de los switches en los diferentes pisos, si los encuentra encendidos

se dirigirá hacia ellos. En este caso el elevador y el panel de los pisos están conectados con un vector-estado, pues el elevador ejecuta una operación getsv a los paneles en los pisos.

Regresemos a nuestro ejemplo de la Compañía "LO VENDO BARATO S.A.", donde hemos considerado dos versiones en sus especificaciones: en la primera las entidades eran cliente, dependiente y orden; y en la segunda eran cliente, orden y producto. En esta ocasión haremos referencia a la segunda versión, la versión automatizada.

La entidad producto tenía una estructura como sigue:

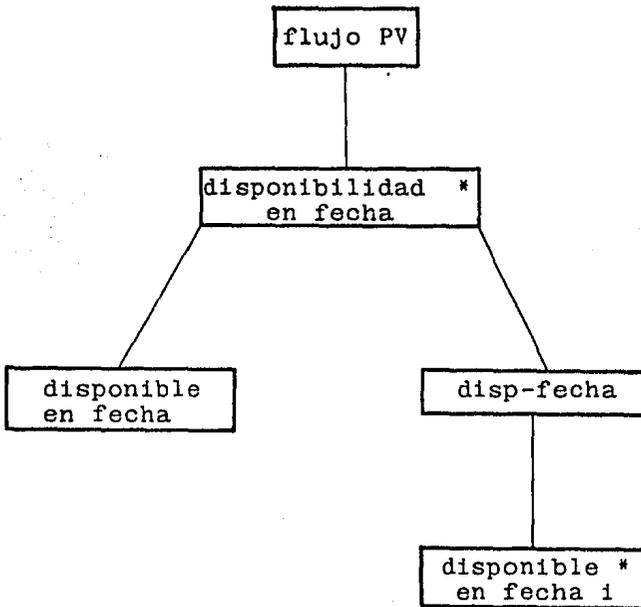


Sin embargo para nosotros ésta entidad existe en el mundo real como un registro generado por el sistema de compras e inventarios de la Compañía.

Si el otro sistema ejecutara una operación de "escribe" para poder obtener la disponibilidad del producto, podríamos utilizar una conexión de flujo de datos, pero en realidad lo que se nos está permitido es acceder el stock del producto grabado en un registro dentro de un archivo a través de una operación getsv, lo que provoca que pueden sucederse dos operaciones getsv con la que obtengamos el mismo valor del estado del producto-0, o sea la misma cantidad disponible para su asignación. Existen dos soluciones para éste problema, en la primera tendríamos que distinguir sucesivas operaciones de "disponible", no por sus valores sino por la intervención de una operación "no-disponible" que ocurriría en cualquier punto de la ejecución del sistema de inventa-

rios, éste punto sería aquel cuando se iniciara un cálculo para un producto en particular. la acción de "no-disponible" cerraría el registro de stock del producto contra cualquier acceso mientras éste fuese actualizado, algo similar a lo que realizan algunos sistemas de base de datos que mantienen "locked" el registro mientras se actualiza.

Otra solución es la de requerir que la acción de "disponible" sea asociada con alguna otra calificación tal como una fecha de identificación si la acción sólo puede ocurrir una vez al día. Esta fecha de identificación junto con la cantidad, es parte del estado del producto-0. La sucesión de vectores-estado en PV, considerados como un flujo de datos tendrían la siguiente estructura:



Agrupando, de acuerdo a la fecha de identificación, los valores sucesivos encontrados en el vector-estado de producto-0, producto-1 es capaz de asegurar que las acciones de disponible han ocurrido al menos una vez en cada caso del componente "disponibilidad en fecha", cuyo texto es :

```

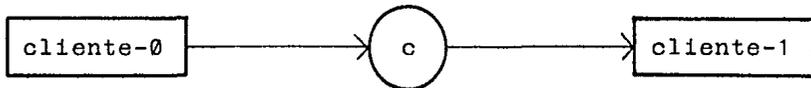
PRODUCTO-1 seq
  gets PV;
  CUERPO PRODUCTO-1 itr
    DISPONIBILIDAD EN FECHA seq
      disponible; i=j (donde PV = fecha);
      gets PV;
      CUERPO DISP-FECHA itr mientras (fecha i)
        gets PV;
      CUERPO DISP-FECHA fin
    DISPONIBILIDAD EN FECHA fin
  CUERPO PRODUCTO-1 fin
PRODUCTO-1 fin

```

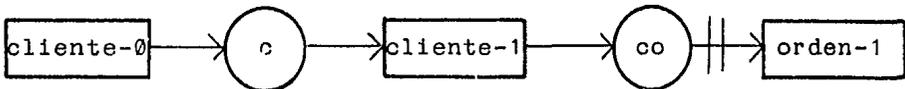
La ventaja de esta solución es que producto-1 necesita inspeccionar el estado de producto-0.

Las dos entidades restantes en nuestro sistema automatizado son cliente y orden.

Para cliente usaremos evidentemente un flujo de datos:



Orden es un marsupial de cliente. así que usaremos la misma entrada para conectarlo al mundo real:



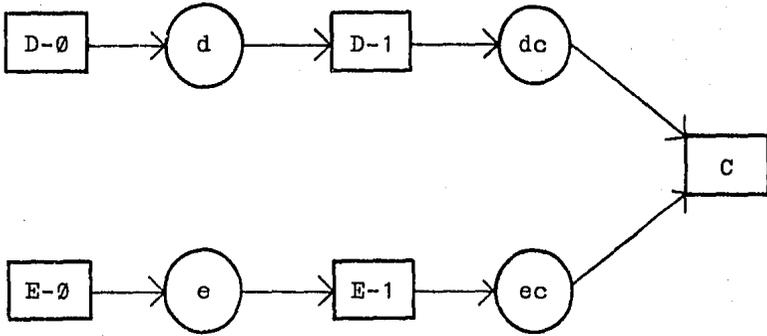
Las líneas paralelas indican que podrán existir varias ordenes para cada cliente. El proceso de cliente-1 aparecerá como sigue:

```
CLIENTE-1 seq
  lee C;
  CUERPO DE CLIENTE-1 itr
    ACCION DEL CLIENTE sel (ordenar(i))
      ordenar; escribe ordenar a CO(i); lee C;
    ACCION DEL CLIENTE alt (corregir(j))
      corregir; escribe corregir a CO(j); lee C;
    ACCION DEL CLIENTE alt (cancelar(k));
      cancelar; escribe cancelar a CO(k); lee C;
    ACCION DEL CLIENTE alt (entregar(l));
      entregar; escribe entregar a CO(l); lee C;
    ACCION DEL CLIENTE fin
  CUERPO DEL CLIENTE fin
CLIENTE-1 fin
```

El proceso de orden será:

```
ORDEN-1 seq
  lee CO;
  ordenar; lee CO;
  CUERPO DE ORDEN-1 itr mientras (corregir)
    corregir; lee CO;
  CUERPO DE ORDEN-1 fin
  FIN sel (cancelar)
    cancelar; lee CO;
  FIN alt (entregar)
    entregar; lee CO;
  FIN fin
ORDEN-1 fin
```

Hasta el momento no hemos considerado como un proceso puede manejar varios flujos de datos a la vez, por lo que para comprenderlo consideraremos un proceso C que recibe un flujo de datos de los procesos D y E, los cuales sólo pueden ejecutar dos acciones diferentes DA1, DA2, EA1 y EA2 respectivamente. El modelo de su interrelación aparecería como sigue:



a) Fusión Fija.- Imaginemos que D y E sólo pueden ejecutar sus acciones en un orden dado e invariable por lo que la estructura del proceso C deberá obedecer a este orden:

```

C seq
  lee dc; DA1
  lee ec; EA1
  lee dc; DA2
  lee ec; EA2
C fin
  
```

b) Fusión de Datos.- En este caso supondremos que D y E podrán realizar sus acciones en cualquier orden, pero cada uno de ellos podrán realizar a lo más una acción en un día cualquiera, a la que le sumaremos una identificación de fecha para lograr esto, por lo que el proceso C aparecería como sigue:

```

C seq
  lee dc;
  lee ec;
  CUERPO DE C itr
    FECHA ACTIVA seq
      fecha:=MIN(fecha-D, fecha-E)
      CUERPO DE FECHA ACTIVA sel(fecha-D=fecha<>fecha-E)
        DA; lee dc;
      CUERPO DE FECHA ACTIVA alt(fecha-D=fecha=fecha-E)
        ACCIONES CONJUNTAS seq
          DA; lee dc;
          EA; lee ec;
        ACCIONES CONJUNTAS fin
      CUERPO DE FECHA ACTIVA alt(fecha-D<>fecha=fecha-E)
        EA; lee ec;
      CUERPO DE FECHA ACTIVA fin
    FECHA ACTIVA fin
  CUERPO DE C fin
C fin

```

En el proceso anterior, la función MIN() regresa el valor menor de sus argumentos. Nótese que es necesario esperar a que existan dos registros disponibles (dc y ec) para poder ejecutar una acción durante todo el proceso, lo que ofrece una gran desventaja pues C estará bloqueado a menos de que existan los dos registros.

c) Fusión Tosca.- Consideremos una última solución en donde le sea permitido a C preguntar por los flujos de datos "dc y ec" para determinar cuando un registro está disponible para su lectura, teniendo así nuestra primera versión del proceso :

```

C itr
  POSIBLE REGISTRO sel (dc con registro)
    lee dc; DA
  POSIBLE REGISTRO alt (ec con registro)
    lee ec; EA
  POSIBLE REGISTRO alt (dc y ec vacios)
    (sin acción)
  POSIBLE REGISTRO fin
C fin

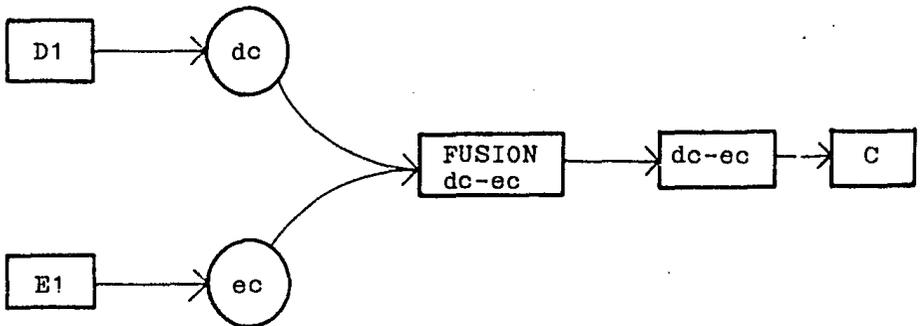
```

En éste podemos decir que la fusión que realiza no es imparcial, pues aún cuando existan registros en el flujo de datos "ec" no serán leídos si existen registros en "dc" pues estos tienen preferencia. Modificando un poco el proceso anterior llegaremos a tener una fusión más parcial en el sentido de que las acciones son ejecutadas casi en el momento en el que llegan :

```

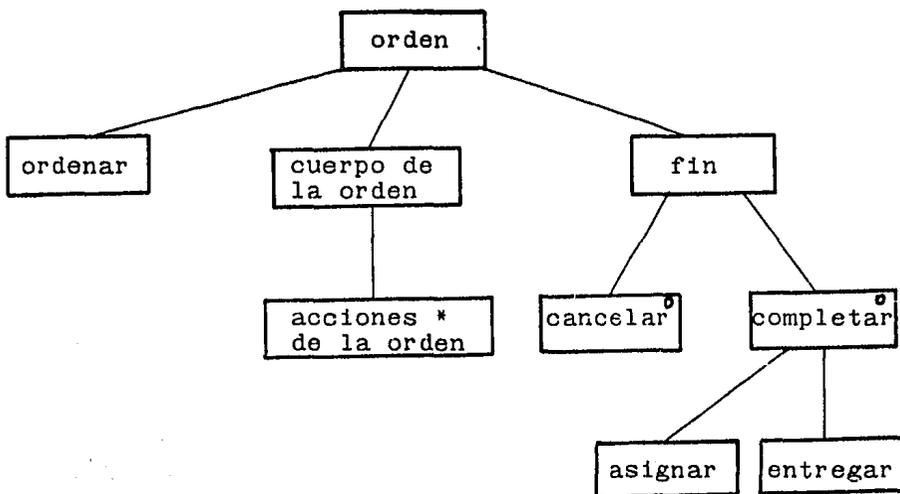
C itr
  POSIBLE REGISTRO-DC sel (dc con registro)
    lee dc; DA
  POSIBLE REGISTRO-DC alt (dc vacio)
    (sin acción)
  POSIBLE REGISTRO-DC fin
  POSIBLE REGISTRO-EC sel (ec con registro)
    lee ec; EA
  POSIBLE REGISTRO-EC alt (ec vacio)
    (sin acción)
  POSIBLE REGISTRO-EC fin
C fin
  
```

Para indicar en un diagrama que se hará una fusión de dos flujos de datos, se hace de la siguiente manera:

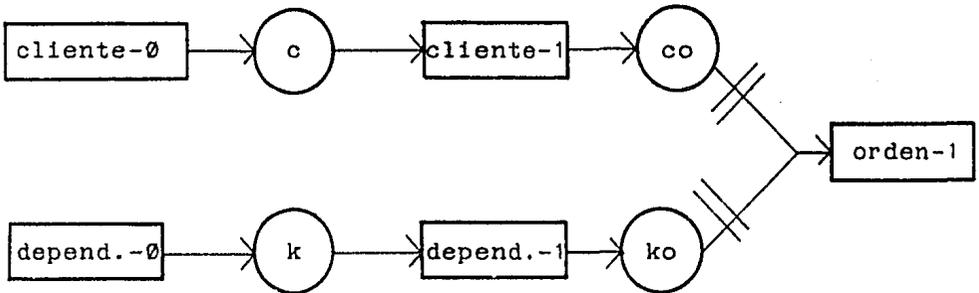


La versión sin automatización de nuestro ejemplo "LO VENDO BARATO S.A.", tiene las entidades cliente, dependiente y orden, donde orden está compuesta de dos entidades marsupiales : orden-cliente y orden-dependiente, en otras palabras, la entidad orden está compuesta de lo que el cliente le hace y lo que el dependiente le hace.

La siguiente es la estructura de la entidad orden en la que las acciones ordenar, corregir, cancelar y entregar son del cliente y las acciones retrasar y asignar pertenecen al dependiente:



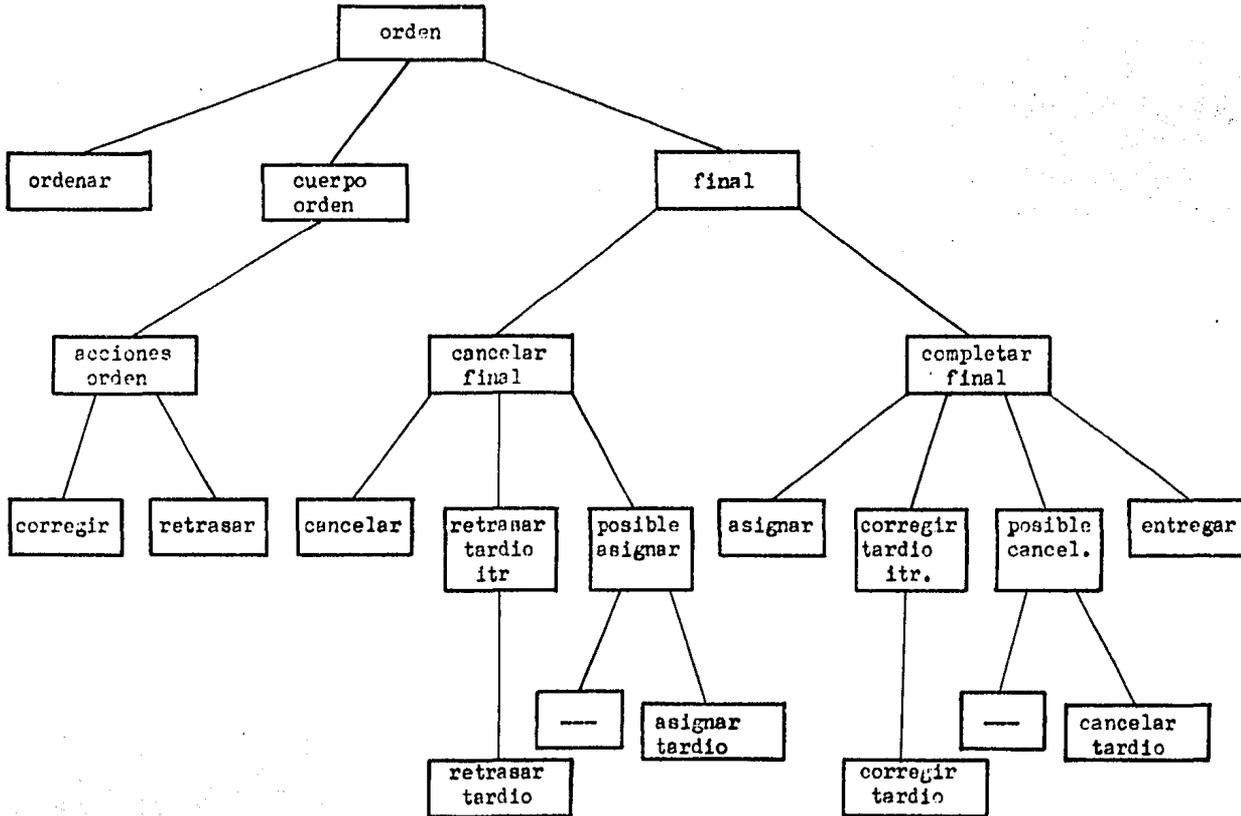
El proceso orden-1 debe fusionar los flujos de datos del cliente y del dependiente a través de una fusión tosca:



Se eligió una fusión tosca por existir la posibilidad de desorden. Ahora consideremos las siguientes suposiciones:

- a) El sistema proveerá información de salida, tanto al cliente como al dependiente, con algún tiempo variable de retraso que debe ser diferente para ambos.
- b) El cliente y el dependiente se comportarán correctamente bajo las bases de sus propias acciones pasadas y la información ahora disponible para él.

Una estructura que refleje estas suposiciones y la fusión tosca de orden-1 sería como la que se muestra en la siguiente hoja, en donde aquellos nombres que incluyen la palabra "tardío" son acciones que no deberían ser realizadas teniendo la apropiada información de salida disponible anteriormente:



6. ETAPA DE FUNCIONES

El resultado de la etapa de modelo inicial es un diagrama de las especificaciones del sistema mostrándolo como un conjunto de procesos que conectan cada una de las entidades con el mundo real que representan y un texto de cada uno. En este momento no existen salidas por parte del sistema y las entradas son solamente mensajes que señalan la ocurrencia de eventos en el mundo real para que puedan ser simulados por el modelo.

El propósito de la etapa de funciones es el de proveer al sistema de salidas. La mayoría de los sistemas realizan muchas funciones de las cuales algunas son especificadas cuando el sistema es originalmente desarrollado, y otras después, en JSD esto no es complicado, pues cada función es especificada en base al modelo y en forma aislada de otras funciones.

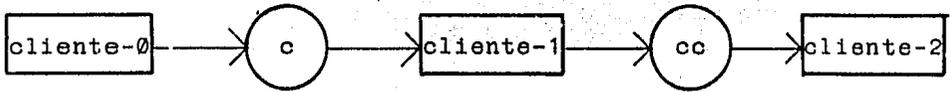
En la especificación de las funciones de un sistema, es necesario que el desarrollador sea capaz de usar su inventiva y elegir entre las diferentes alternativas que se le pueden presentar, siempre teniendo en mente las siguientes reglas:

a) Las funciones son siempre representadas como procesos con un largo tiempo de vida, y nunca como procedimientos. No podemos declarar un reporte semanal como una función si lo especificamos como un procedimiento que se ejecuta cada semana; en vez de eso, podemos especificar un proceso secuencial cuya salida es el conjunto de todos los reportes semanales producidos durante la vida del sistema.

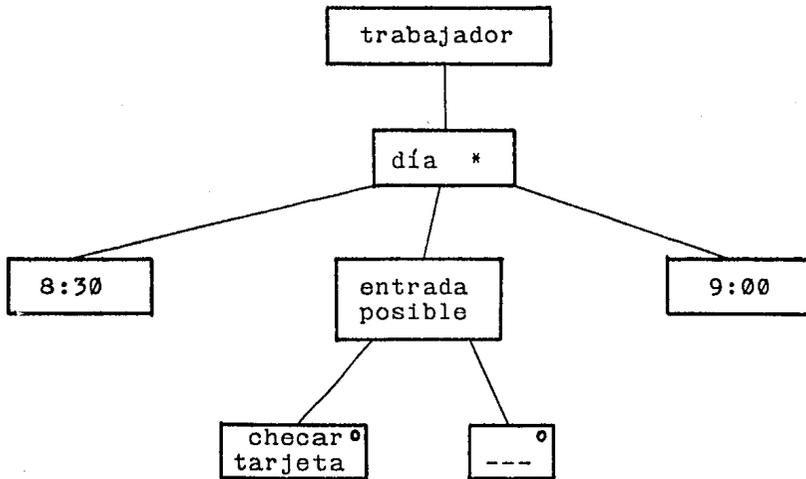
Especificando un proceso dejamos abierta la oportunidad y hacemos posible el mecanismo para detallar cualquier relación deseada durante la vida del proceso. Si al final un procedimiento debe ser adecuado, podemos fácilmente transformar el proceso en un procedimiento en la etapa de implantación.

b) El modelo de un proceso debe ser elaborado introduciendo variables y operaciones elementales. Si una elaboración estructural es requerida, el modelo de un nuevo proceso debe ser especificado y conectado al modelo del proceso original mediante un flujo de datos. Podremos hacer referencia al modelo del nuevo proceso como proceso "nivel-2".

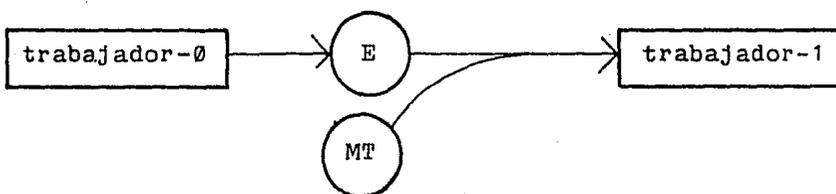
Para hacer más elaborado el proceso de cliente-1, introducimos cliente-2:



6.1 MARCAS DE TIEMPO .- En los modelos de procesos que hemos especificado, el paso del tiempo en la vida real es marcado sólo por la ocurrencia de acciones, dadas al modelo a través de registros en un flujo de datos y en el intervalo entre una acción y otra el sistema queda suspendido. Algunas ocasiones necesitaremos modelar un tipo de acción que consista en no hacer nada durante un período de tiempo, o que realice cierta acción durante un rango de tiempo, como lo sería el automatizar el reloj checador a la entrada de una fábrica en donde los trabajadores sólo pueden entrar entre las 8:30 y las 9:00 am:



El poder reconocer un punto en el tiempo y ser tratado en forma significativa dentro del modelo de un proceso es a través de una lectura a un flujo de datos especial llamado "marcador de tiempo" (MT) que requiere siempre de una fusión tosca con otra entrada del proceso :

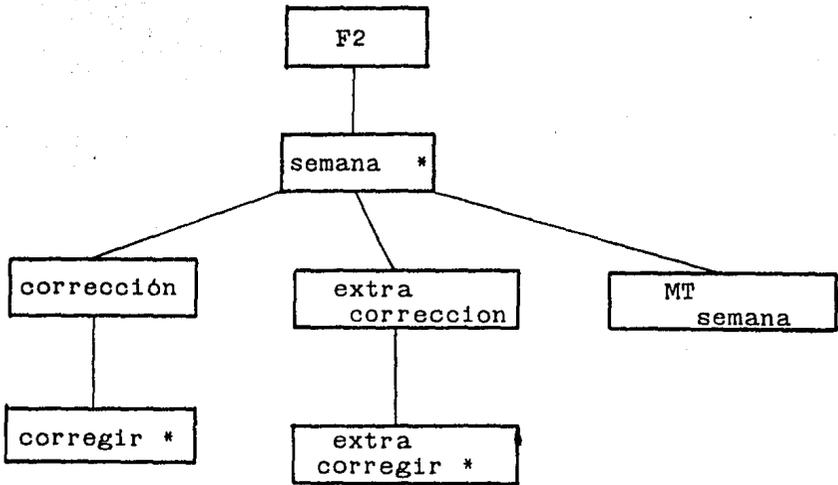
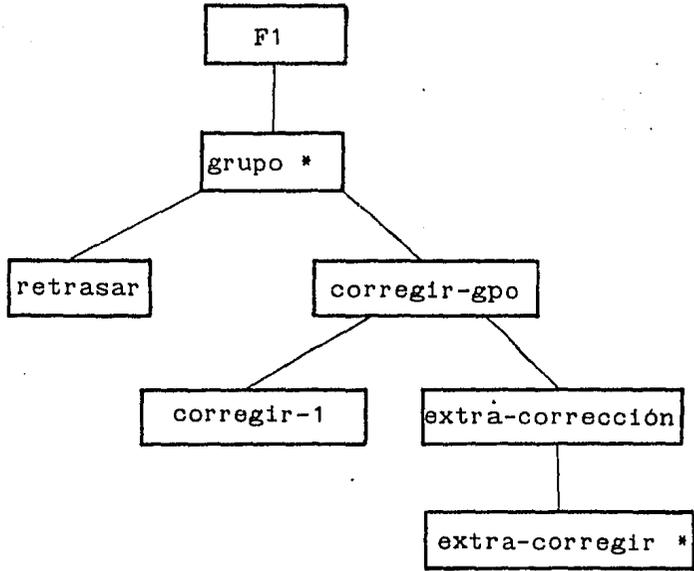


6.2 FUNCIONES 1 y 2 (nivel-2) "LO VENDO BARATO S.A." .- Tomando en cuenta el modelo inicial creado en el capítulo pasado para la versión no automatizada de nuestro sistema sumaremos las siguientes funciones.

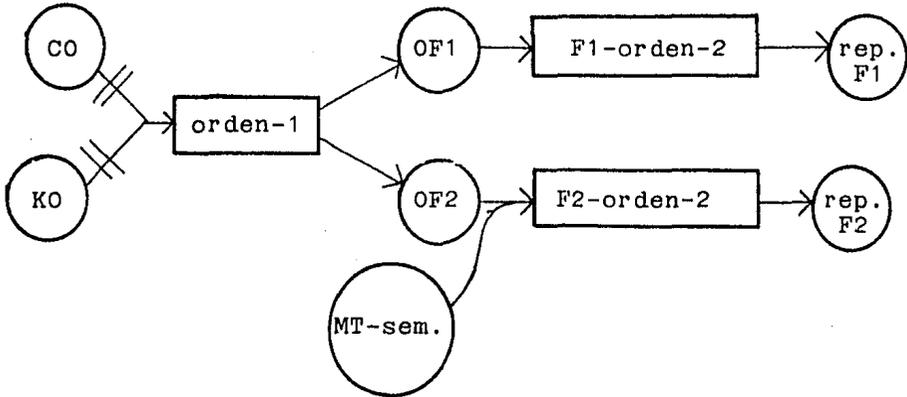
El gerente de ventas de la empresa está preocupado, ya que piensa que puede ser excesiva la actividad de "corregir" por parte del cliente, por lo que desea que el sistema ejecute las siguientes funciones :

- F1) Un reporte de excepción por cualquier orden que sea corregida más de una vez sin la intervención de una acción de retraso.
- F2) Un reporte de excepción por cualquier orden que sea corregida más de dos veces en una semana.

La estructura de orden-1 no contiene los componentes apropiados para meter las operaciones de "escribe reporte-F1 o reporte-F2" por lo que se necesitan las siguientes estructuras : (Ver en la siguiente página).



Estas estructuras son un tanto incompatibles con orden-1, por lo que debemos sumar la función de estos procesos al SSD (Diagrama de Especificación del Sistema) de orden-1, siendo cada uno de éstos un modelo a nivel-2 de la entidad orden. Cada función deberá estar conectada a su orden-1 vía un flujo de datos y F2 además requiere un flujo de datos con un MT (marcador de tiempo) semanal: (ver sección 5.2.5 para comprender mejor el siguiente diagrama)



En orden-1 ponemos las siguientes operaciones para escribir :

- a) Un registro de cada retrasar, corregir y corregir-tardío a OF1.
- b) Un registro de cada corregir y corregir-tardío a OF2

MT-sem. es un marcador de tiempo por semana, o sea que escribirá un registro cada semana al flujo de datos y será fusionado con el flujo de datos de OF2 para dar como entrada a F2-orden-2 la fusión OF2&MT-sem.

Existe un proceso F1-orden-2 y uno F2-orden-2 para cada orden-1. La estructura de F2-orden-2 es la que se muestra en la siguiente página:

```

F2-ORDEN-2 seq
  lee Of2&Mt-sem;
  CUERPO DE F2-ORDEN-2 itr
    SEMANA seq
      contador-corr:=0;
      CORRECCION itr mientras(corregir y contador-corr<2)
        contador-corr:=contador-corr+1; lee OF2&MT-sem;
      CORRECCION fin
      EXTRA CORRECCION itr mientras (corregir)
        imprimir el reporte F2;
        lee Of2&MT-sem;
      EXTRA CORRECCION fin
      MT SEMANAL seq
        lee OF2&MT-sem;
      MT SEMANAL fin
    SEMANA fin
  CUERPO DE F2-ORDEN-2 fin
F2-ORDEN-2 fin

```

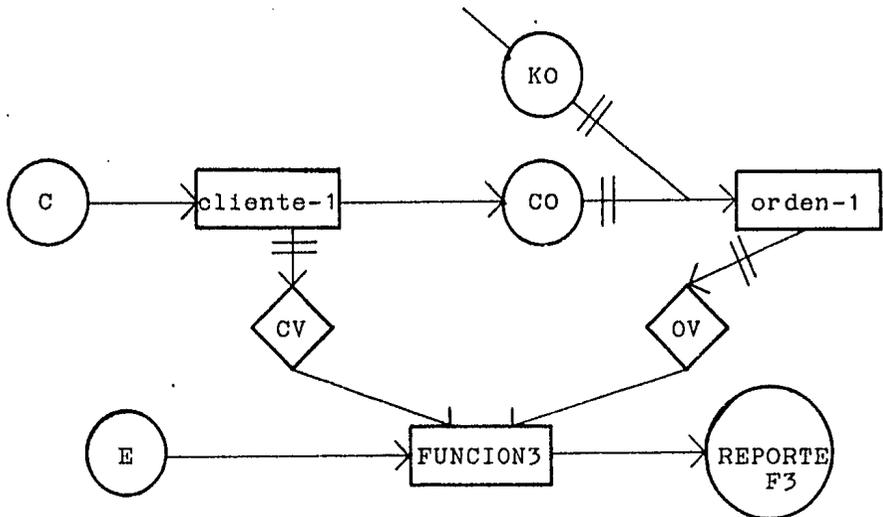
6.3 FUNCIONES IMPUESTAS .- Son aquellas con más complejidad que las funciones simples (como F1 y F2) pues para cada registro de entrada es necesario acceder varios vectores-estado de diferentes tipos de entidades y accederlos en un cierto orden y en forma selectiva. Por ejemplo en el sistema de "LO VENDO BARATO S.A.", podríamos desear las siguientes funciones:

- a) Listar todas las ordenes de un determinado producto.
- b) Listar todos los clientes que tienen ordenes que no han sido asignadas a un cierto producto.
- c) Listar todos los clientes que tienen ordenes de cualquier producto del cual un cierto cliente tiene una orden.
- d) Determinar si existe un producto ordenado por más de 10 clientes, y si es así, listarlos.

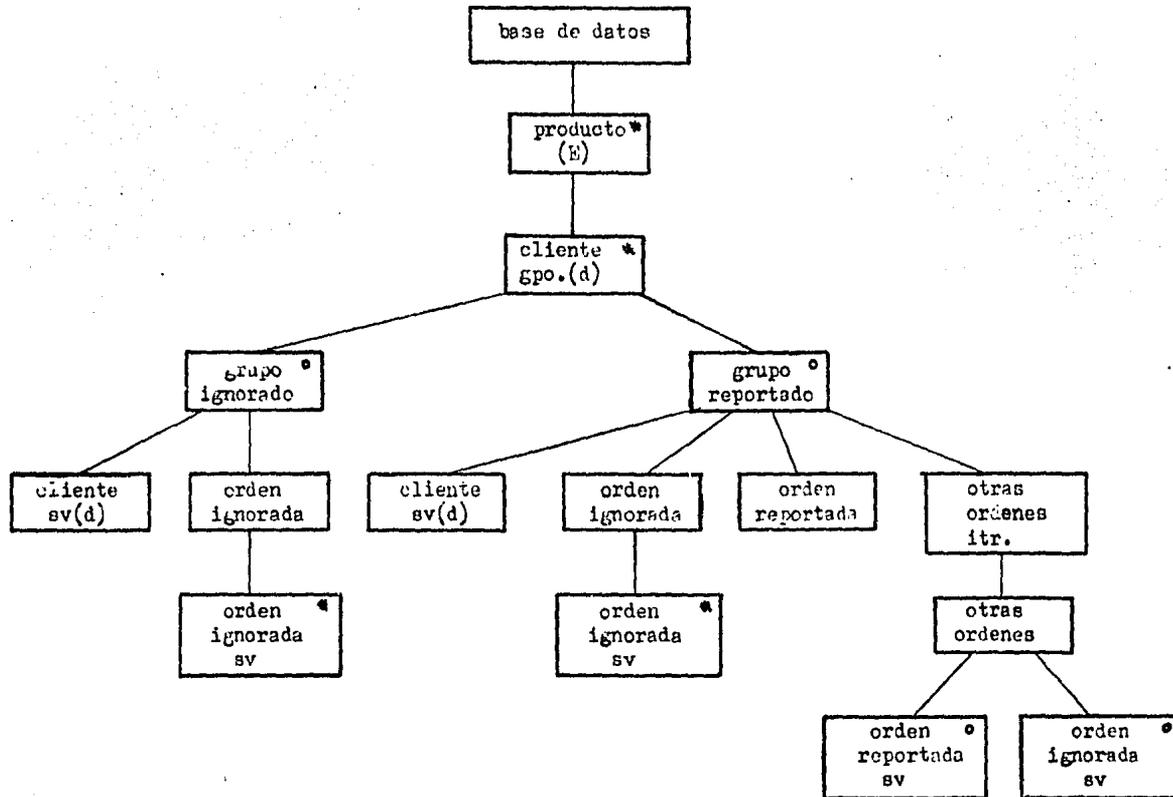
En algunos ambientes de implantación podríamos dejar estas funciones a un manejador de bases de datos de propósito general, pero en otros casos necesitamos tratarlos en la etapa de funciones, y aún cuando contamos con el software conveniente necesitamos hacer un análisis del acceso a la base de datos que es necesario para escoger una eficiente implantación de la base de datos.

En esta etapa determinaremos los caminos de acceso necesarios para las funciones especificadas.

6.4 FUNCION 3 (impuesta) "LO VENDO BARATO S.A."- El gerente de ventas necesita se tome en cuenta la siguiente función: para un producto determinado, listar los nombres de todos los clientes que tengan ordenes de ese producto pero que aún no han sido asignadas o canceladas. Para ésto es necesario especificar un proceso que maneje las consultas para todos los productos, el proceso deberá inspeccionar el estado actual de las ordenes, así como el de los clientes, asumiendo que el nombre del cliente es un atributo de cliente-1, por lo que el SSD para F3 aparecerá como sigue:



Para facilitarnos las cosas podemos introducir la variable "orden-activa" a orden-1, a la cual le asignaremos el valor de "si" cuando la acción ordenar ocurra y "no" al final del proceso de la orden.



F3 deberá obtener los vectores-estado de cliente-1 y orden-1 en el orden que le permita producir el reporte deseado, de modo que tomaremos las siguientes premisas para producir la estructura del proceso:

- a) Dado un producto "E", es posible obtener sucesivamente todos los vectores-estado de los clientes que al menos tienen una orden activa para el producto "E".
- b) Una vez teniendo el vector-estado del cliente "C", quien tiene al menos una orden activa para el producto "E", es posible obtener todas las demás ordenes activas de ese cliente por ese producto sucesivamente

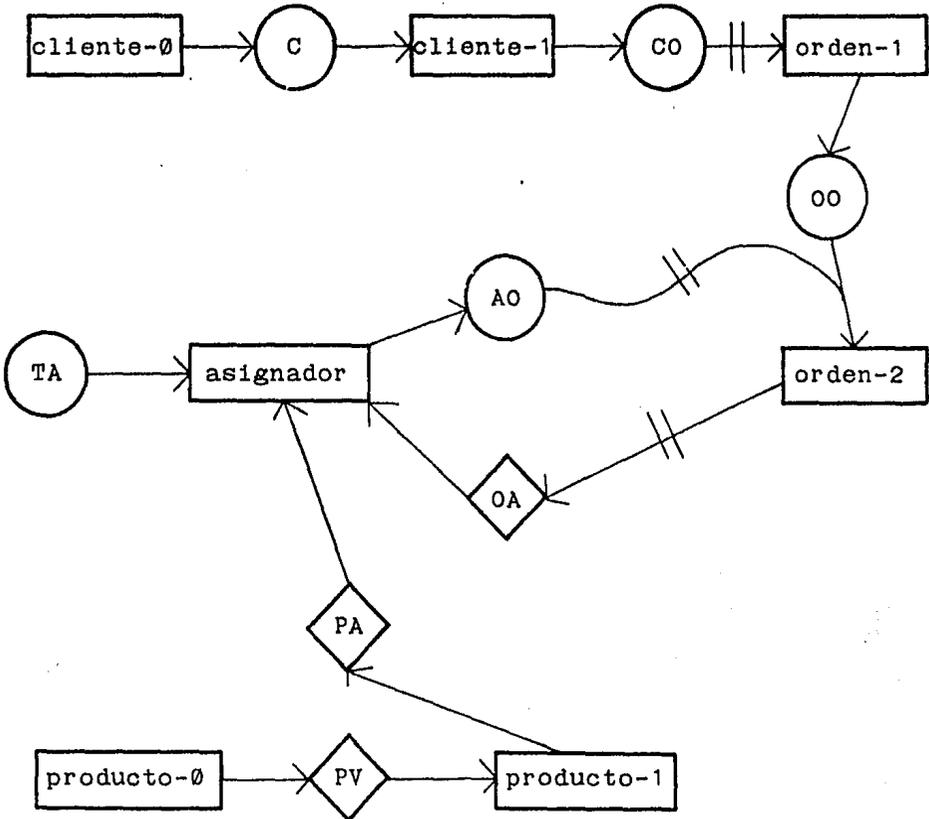
Tomando lo anterior como premisa podremos armar la estructura presentada en la siguiente página :

6.5 FUNCION ESPECIALIZADA #1 .- Especificaremos una función para la asignación automática de stock a las ordenes activas, un reporte será producido que muestre las asignaciones hechas así como los retrasos de ordenes a las que no se les pudo asignar la mercancía por falta de stock en el inventario. Para la especificación de esta función tomaremos la versión automatizada del sistema que hemos venido desarrollando.

Para poder especificar el proceso "asignador" es necesario tomar en cuenta las políticas de la compañía:

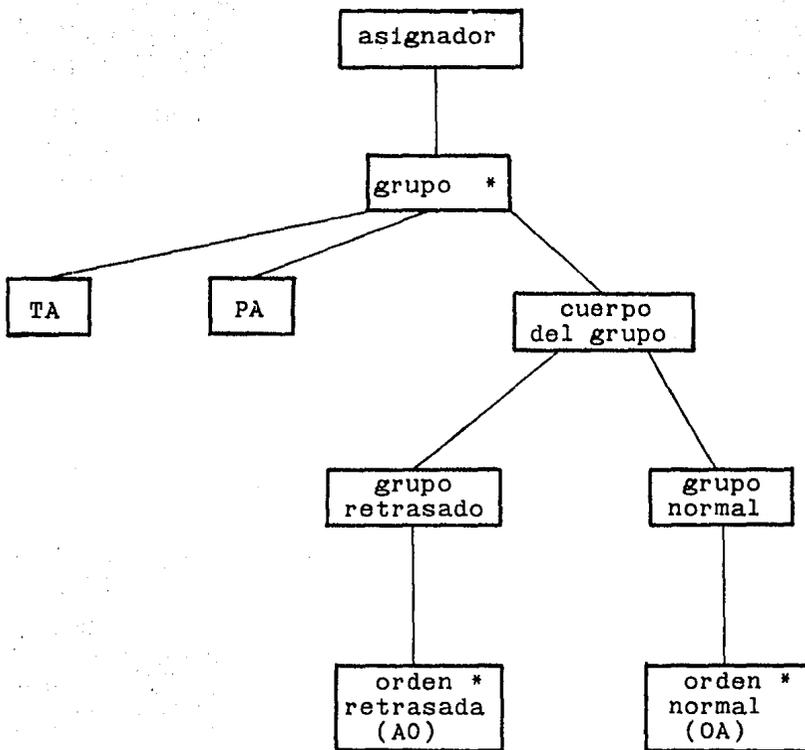
- a) No puede haber sustitución de productos al momento de su asignación, esto es, a una orden de lápices sólo se le pueden asignar lápices.
- b) Si una orden es retrasada al momento de la asignación se le adjudicará una mayor prioridad en el siguiente proceso de asignación .
- c) La prioridad de una orden no se verá afectada por lo que le ocurra a otra orden del mismo cliente.
- d) El proceso de asignación se realizará diariamente, después que se haya completado la actualización del archivo de inventarios por parte del sistema de compras e inventarios que existe actualmente.

El proceso asignador deberá conectarse al proceso producto-1 del cual obtendrá información relativa a la disponibilidad de stock del producto, así también deberá estar conectado interactivamente con el proceso orden-2, más no hay motivo para conectarlo con el proceso cliente-1, así que el SSD aparecerá como sigue:



En donde cada registro del flujo de datos TA al proceso asignador señala que la asignación es requerida para el producto manejado por asignador. Se ha asumido una conexión vector-estado entre asignador y producto-1 y entre asignador y orden-2. El flujo de datos AO contiene registros de retraso y asignación que son fusionados con el flujo de datos OO de orden-2.

La estructura detallada de asignador es la siguiente:



Cuando una marca de tiempo TA es leída, asignador obtiene el vector-estado de producto-1 para determinar que stock está disponible para asignación el vector-estado del proceso orden-2 para tomar el producto que está esperando para su asignación, primero el vector-estado de aquellos que se retrasaron y después los de las demás ordenes.

El texto de la estructura de asignador es:

```
ASIGNADOR itr
GRUPO seq
  lee TA;
  inspecciona PA; disponible:= cantidad en PA;
CUERPO DEL GRUPO seq
  GRUPO RETRASADO itr mientras(retrasados)
    ORDENES RETRASADAS sel (requerido <= disponible)
      disponible:=disponible-requerido;
      escribe asignar a AO(OA);
    ORDENES RETRASADAS alt (requerido > disponible)
      escribe retrasar a AO(OA);
    ORDENES RETRASADAS fin
      inspecciona OA;
  GRUPO RETRASADO fin
  GRUPO NORMAL itr mientras (NOT fin-de-OA)
    ORDEN NORMAL sel (requerido <= disponible)
      disponible:=disponible-requerido;
      escribe asignar a AO(OA);
    ORDEN NORMAL alt (requerido > disponible)
      escribe retrasar a AO(OA);
    ORDEN NORMAL fin
      inspecciona OA;
  GRUPO NORMAL fin
CUERPO DEL GRUPO fin
GRUPO fin
ASIGNADOR fin
```

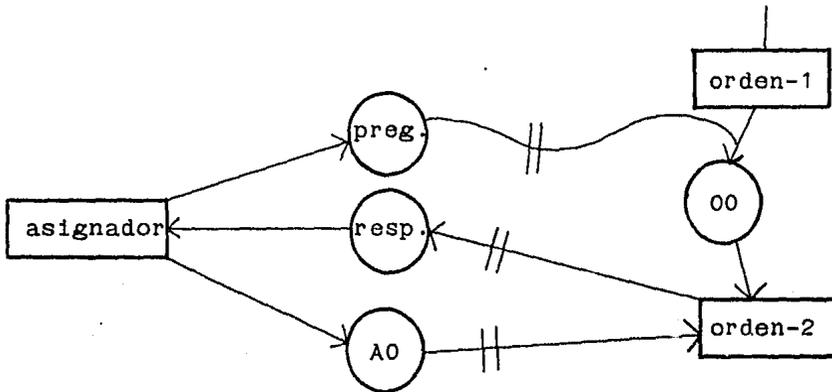
El punto débil en la conexión entre asignador y orden-2 se puede dar en la siguiente secuencia de eventos:

- a) Asignador obtiene el vector-estado de la orden y escribe un registro de asignar a AO.
- b) Antes de que el registro de asignar sea leído por orden-2, un registro cancelar es leído.

c) El registro asignar es entonces un registro de asignación tardía.

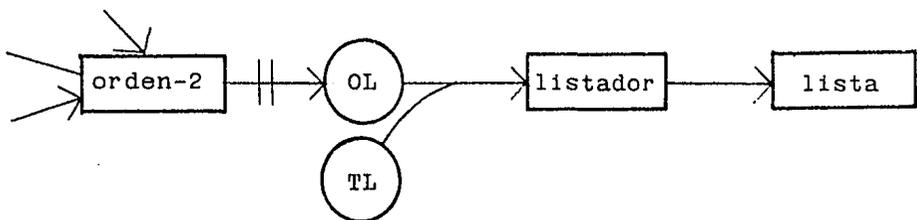
El resultado de esta secuencia de eventos es que el stock de mercancía que se le asignó a la orden cancelada, no será entregada al cliente, pero por el momento no está disponible para que sea asignada a otra orden.

Para evitar esto, podríamos cambiar el vector-estado OA por un flujo de datos:

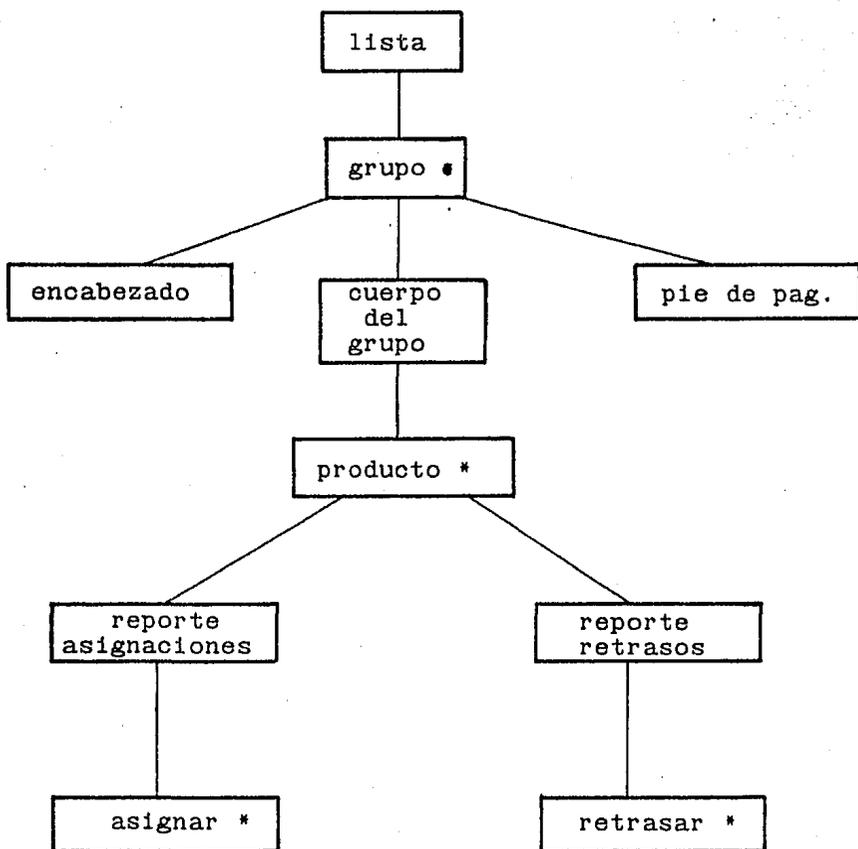


El proceso asignador escribe un registro a "preg.", preguntando cual es la cantidad requerida para su asignación. Orden-2 hace una fusión tosca del flujo preg. con el flujo 00. Al leer un registro preg. escribe uno resp. y espera por un asignar o retrasar en el flujo AO, esto provoca que sea imposible leer un registro 00 entre un preg y un AO, por lo que podríamos quitar de la estructura de la orden los componentes "retrasando tardíamente" y "posible asignación".

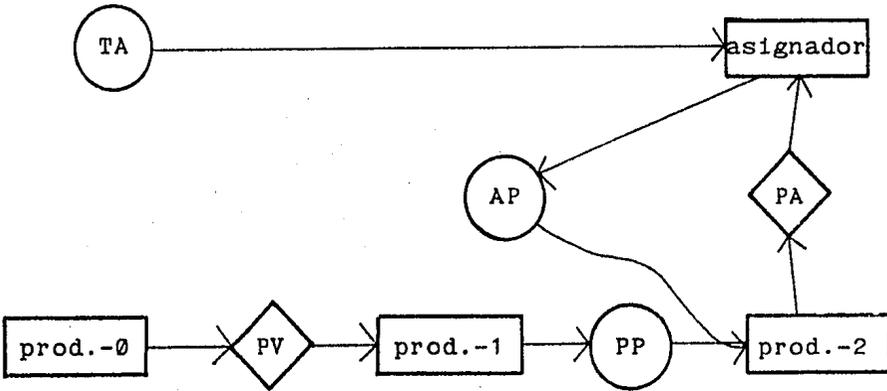
El reporte de asignaciones y retrasos no ofrece gran dificultad y si la lista tendrá una secuencia especial, tal vez por clave del producto, debemos incluir un proceso que reestructure la salida, por lo que el SSD aparecerá como se muestra en la siguiente página:



Y su estructura será:

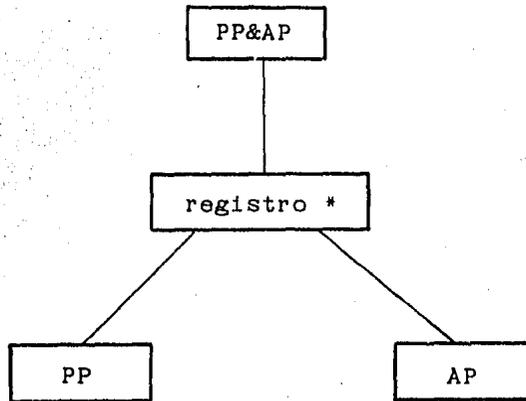


6.6 FUNCION ESPECIALIZADA #2 .- Hemos asumido que la asignación se hará sólo una vez al día, después de la actualización del archivo por el otro sistema. Cuando asignador lee un registro TA sabrá que cuando inspeccione el vector-estado PA de producto-1 mostrará la cantidad actualizada del stock disponible para su asignación:



Con las modificaciones al SSD anterior permitimos que la asignación sea realizada en demanda para productos individuales, especificando a asignador como función interactiva con respecto al modelo del producto.

Producto-2 está conectado a producto-1 por un flujo de datos PP, cada registro del cual indica una cantidad disponible, este flujo PP se fusiona con el flujo AP, cada registro del cual indica el monto total asignado por asignador. La estructura de producto-2 está basada en la estructura de la fusión PP&AP:



```

PRODUCTO-2 seq
  cantidad:=0;
  lee PP&AP;
  CUERPO DE PRODUCTO-2 itr
    REGISTRO DE PRODUCTO-2 sel (reg. PP)
      cantidad:=cantidad en reg. PP;
    REGISTRO DE PRODUCTO-2 alt (reg. AP)
      cantidad:=cantidad - lo asignado;
    REGISTRO DE PRODUCTO-2 fin
  lee PP&AP;
  CUERPO DE PRODUCTO-2 fin
PRODUCTO-2 fin
  
```

7. ETAPA DE SINCRONIZACION DEL SISTEMA .- En esta etapa, el desarrollador considera o reconsidera, las restricciones que el sistema debe satisfacer por la implantación si el sistema hace lo que es requerido.

Estas restricciones deben ser recolectadas y documentadas antes de la etapa de implantación, restricciones tales como :

- a) Ciertas salidas del sistema deben ser producidas dentro de un límite de tiempo después de llegadas las entradas de las que se derivará.
- b) Los procesos a nivel-1 que son conectados con las entidades en el mundo real a nivel-0 por una conexión vector-estado deberán ejecutar operaciones de "inspección" con una cierta frecuencia.
- c) Los modelos de procesos cuyos vectores-estado sean inspeccionados por otro proceso, deben tenerlos actualizados suficientemente cuando ésto ocurra.
- d) Los registros de MT (marcas de tiempo) deberán ser leídos dentro de un cierto período después de haber sido escritos.
- e) Ciertos pares de procesos, en donde la salida de uno es fusionada con otro para que sirva de entrada, deberán estar sincronizados.

En la mayoría de los casos la documentación de restricciones de sincronización es enteramente informal a la que se hace referencia en la etapa de implantación, pues podemos especificar si algún proceso correrá en línea (ON-LINE), y cuales otros en batch diariamente o semanalmente.

Es en ésta sección donde los elementos de mayor importancia son los MT (marcas de tiempo) pues especificamos sus características.

8. ETAPA DE IMPLANTACION

Se deben establecer ciertas reglas para la programación de los procesos, de tal forma que cada uno ocupe el tiempo de procesador suficiente.

En JSD escribimos las especificaciones de tal forma que asemejen al mundo real; en la etapa de implantación transformaremos estas especificaciones para que sean aptas a un ambiente de máquina.

A lo largo del desarrollo de las diferentes etapas, hemos considerado procesos separados y en esencia, cada uno de éstos con un procesador propio que los ejecuta, por lo que aunque en teoría se manejó un ambiente de multiproceso, en la realidad estos recursos serán escasos por lo que una manera de reducir el número de procesadores necesarios es simulándolos en un sólo procesador, esto puede hacerse con técnicas como la de multiprogramación. Otra manera de reducir el número de procesadores necesitados, es la de transformar las especificaciones de tal forma que existan sólo unos cuantos procesos secuenciales que ejecutar. Esta reducción es efectiva sólo desde el punto de vista de implantación, pues no existe manera de reducir el número de procesos especificados sin alterar el modelo, la función o ambos. Sin embargo, podemos hacer parecer a la máquina que sólo un proceso es ejecutado cuando en verdad está compuesto de un conjunto de éstos.

Los procesos deben de ser combinados de tal forma que parezcan ser un sólo proceso ejecutado por un sólo procesador.

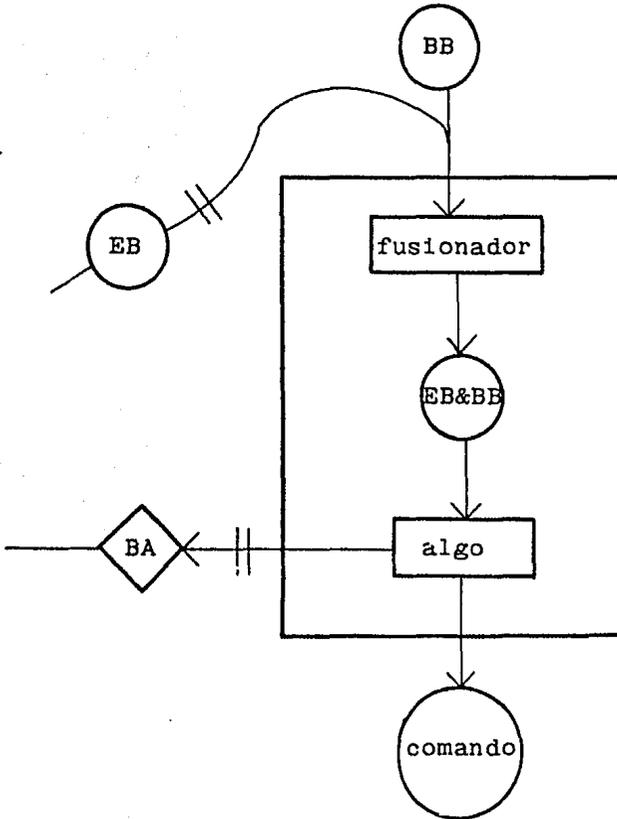
El primer paso es el de considerar cómo es que los procesos han de ser coordinados, esto es, cómo es que el tiempo de un procesador ha de ser compartido entre dos procesos. Para implantar esta compartición del procesador, necesitamos hacer unos cambios a los textos de los procedimientos para tenerlos bajo el control de un "coordinador", para lo cual convertiremos a los procesos en subrutinas que pueden ser llamadas por el coordinador usando la operación :

CALL <Proceso> (<registro>)

o

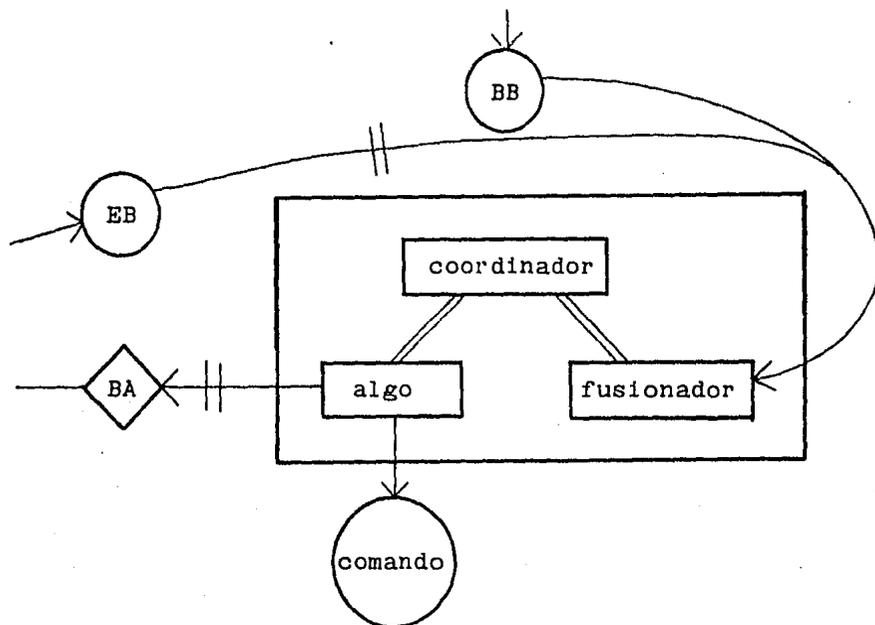
LLAMA A <proceso> (<registro>)

Podemos mostrar que los procesos están compartiendo un procesador si enmarcamos con una línea al SSD:



Así también podemos mostrar la configuración de la coordinación en el procesador distribuido dibujando un nuevo

tipo de diagrama llamado SID (Sistem Implementation Diagram, Diagrama de Implantación del Sistema).



La línea vertical a la derecha de coordinador le da esa característica especial, las dobles líneas que unen a algo y fusionador a coordinador, indican que cada uno está invertido con respecto a un flujo de datos y es una subrutina de coordinador, no es necesario escribir junto a las líneas paralelas el nombre del flujo de datos, pero es recomendable para una mejor visualización. El orden de las llamadas a las subrutinas no se muestra en el SID, éste sólo puede verse en el texto del proceso coordinador.

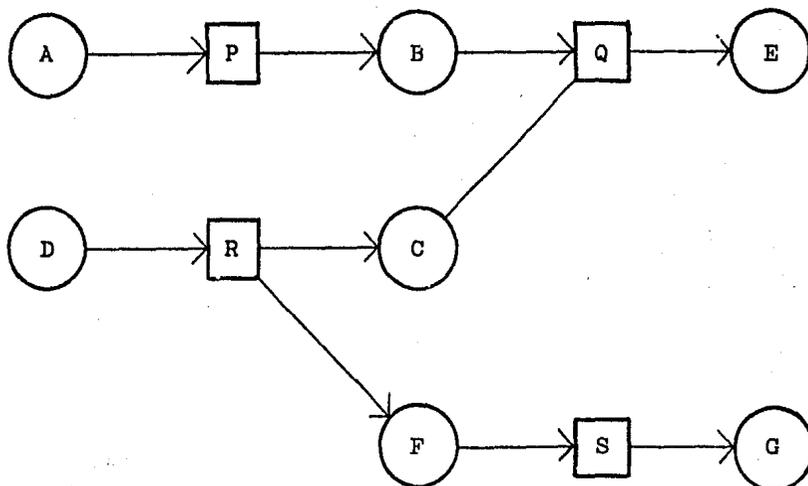
La conversión de "algo" y "fusionador" en las subrutinas requeridas, es llevado a cabo en el momento de codificarlo en algol, pascal, C o cualquier otro lenguaje de programación.

8.1 COORDINACION SIN COORDINADOR .- El concepto de inversión se refiere a aquella transformación en la cual un proceso secuencial es convertido en procedimiento, invocado una vez por cada registro de cada flujo de datos con respecto al cual el proceso está invertido.

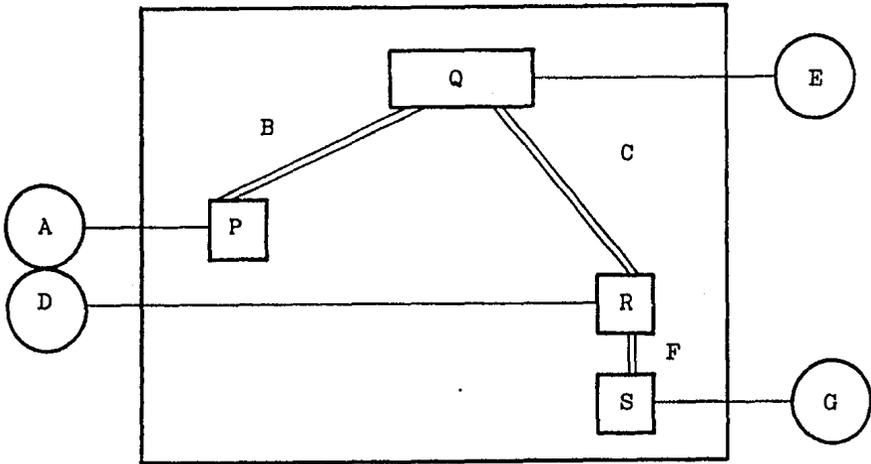
La técnica de combinar procesos por inversión sin añadir un proceso coordinador por separado, puede ser aplicado a aquel sistema que cumpla con las siguientes condiciones:

- a) Sólo es usada la conexión de flujo de datos.
- b) No hay fusiones toscas en el sistema.
- c) Cualquier par de procesos estarán conectados por una sola ruta.

Tomemos como ejemplo el siguiente SSD :



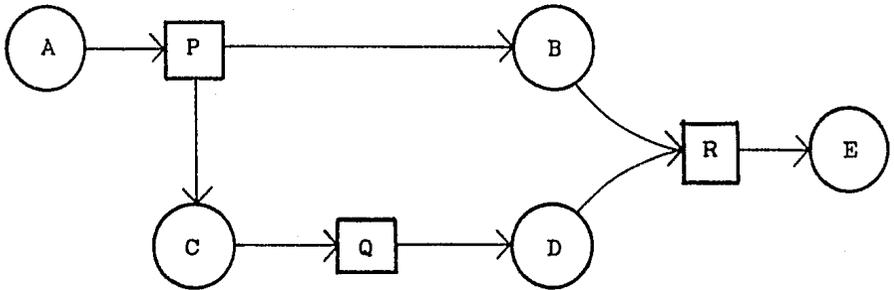
Y su respectivo SDI:



Q es efectivamente el proceso coordinador del sistema. P es invertido con respecto a B, R es invertido con respecto a C y S es invertido con respecto a F. El esquema coordinador implantado es:

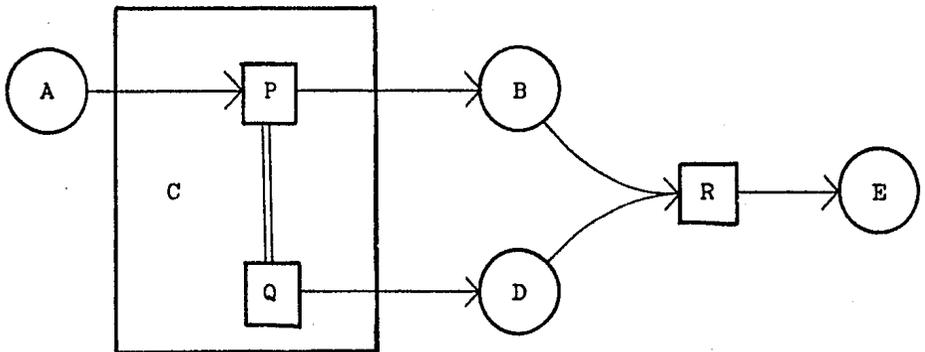
- Correr Q hasta que esté bloqueado esperando por un registro B o C.
- Si Q está bloqueado esperando por un registro B, correrá P hasta que P produzca el registro B, entonces se suspenderá P y continuará Q procesando. Un proceso similar ocurrirá si está Q bloqueado esperando un registro C por lo que se ejecutará el proceso R.
- Si R, mientras corre produce un registro F, se suspenderá R y se correrá S hasta que se haya generado el registro F.

8.2 IMPLANTACION DE FUSIONES CON INVERSION .- Imaginemos el siguiente sistema :

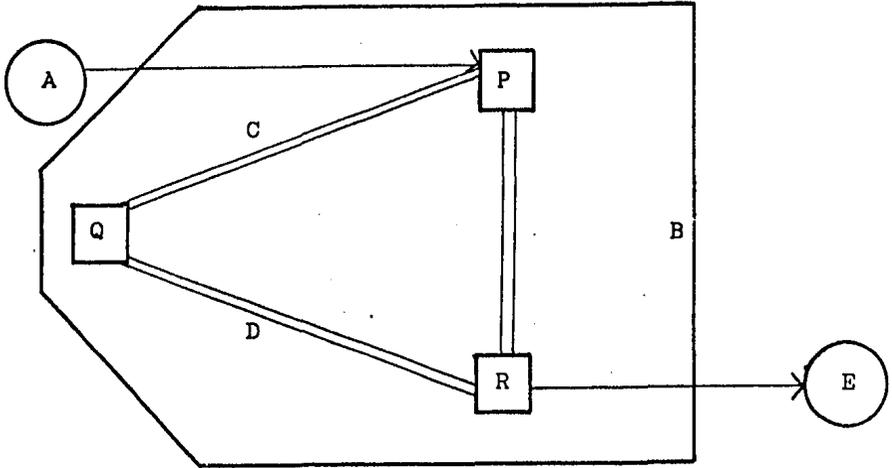


Este sistema no cumple con las condiciones para combinar todos los procesos por inversión sin añadir un proceso coordinador por separado. Los procesos P y R están conectados por dos rutas, sin embargo, como R hace una fusión tosca con sus flujos de datos, podemos coordinarlo en forma muy sencilla.

Podemos combinar P y Q e invirtiendo Q obtendremos:

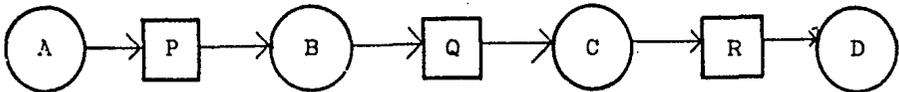


Los procesos combinados P Y Q podrán escribir los registros a los flujos de datos B y D en algún orden. Si este orden es aceptable como resultado de la fusión tosca en R podemos invertir R con respecto al flujo fusionado B&D y hacerlo una subrutina de los procesos combinados P y Q :



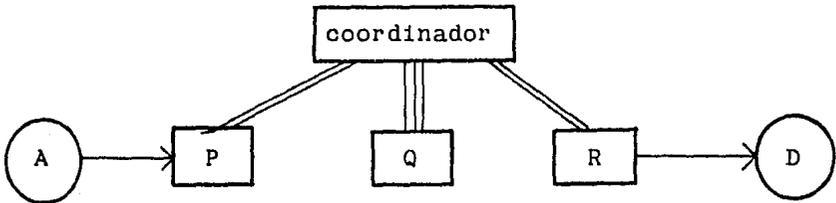
La condición esencial añadida es que sean aceptados los registros B&D en el orden en que son escritos por P y Q.

8.3 CANALES.- Supongamos que queremos utilizar un proceso coordinador por separado para implantar el siguiente sistema:



Podemos traer bajo el control del coordinador : a P invirtiéndolo con respecto a B, y R con respecto a C, pero Q presenta una dificultad. Necesitamos suspender la ejecución de Q cuando lea B y cuando escriba C, esto es sencillo si tratáramos la escritura de B en P y la lectura de C en R, pero el coordinador debe saber si Q está suspendido por leer B o escribir C de tal forma que maneje determinados parámetros y tomar la acción adecuada.

Podemos representar inversión con respecto a dos o más flujos de datos usando una unión de tres o más líneas en el SID :



A este tipo de conexión se le llama "conexión de canal". Q está conectado por un canal al coordinador y las tres líneas indican que Q está invertido a dos flujos de datos.

Para llamar a un proceso que está conectado por un canal se ejecuta una operación :

query <proceso>

o

pregunta <proceso>

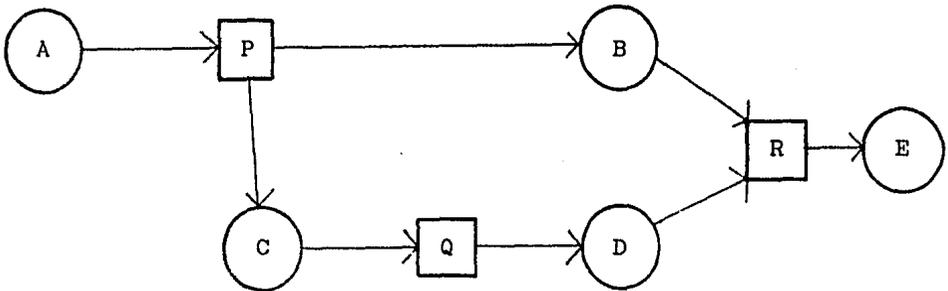
El texto del proceso coordinador será :

```

COORDINADOR itr
  REG-COORDINADOR seq
    pregunta Q;
    TRANSFERENCIA sel (lee B en Q)
      REG-B seq
        LLAMA A P (reg. B)
        LLAMA A Q (reg. B, reg. C)
      REG-B fin
    TRANSFERENCIA alt (escribe C en Q)
      REG-C seq
        LLAMA A Q (reg. B, reg. C)
        LLAMA A R (reg. C)
      REG-C fin
    TRANSFERENCIA fin
  REG-COORDINADOR fin
COORDINADOR fin

```

8.4 COORDINANDO CON BUFFERS.- Considerémos el siguiente sistema imaginario:



En esta ocasión R fusiona sus entradas pero no mediante una fusión tosca, sino por algún otro tipo. El orden en el que los registros B y D son leídos por R es determinado sólo por la estructura de R misma, y no puede asumir el orden en el que son escritos por P y Q. La supuesta idea del sistema es :

- a) El flujo de datos A contiene registros enteros en orden ascendente sin duplicados.
- b) P escribe registros alternos de A a B y C, escribiendo el primero a B.
- c) Q copia los registros de C a D, sumando 8 a cada entero.
- d) R fusiona los registros de B y D, eliminando duplicados y dando a la salida de flujo de datos E la lista de registros en orden ascendente.

Un ejemplo del contenido de los flujos de datos es :

```

A: 1, 5, 7, 10, 13, 17, 18, 25, 28, ...
B: 1, 7, 13, 18, 25, 28, ...
C: 5, 10, 17, 25, ...
D: 13, 18, 25, 33, ...
E: 1, 7, 13, 18, 25, 28, ...

```

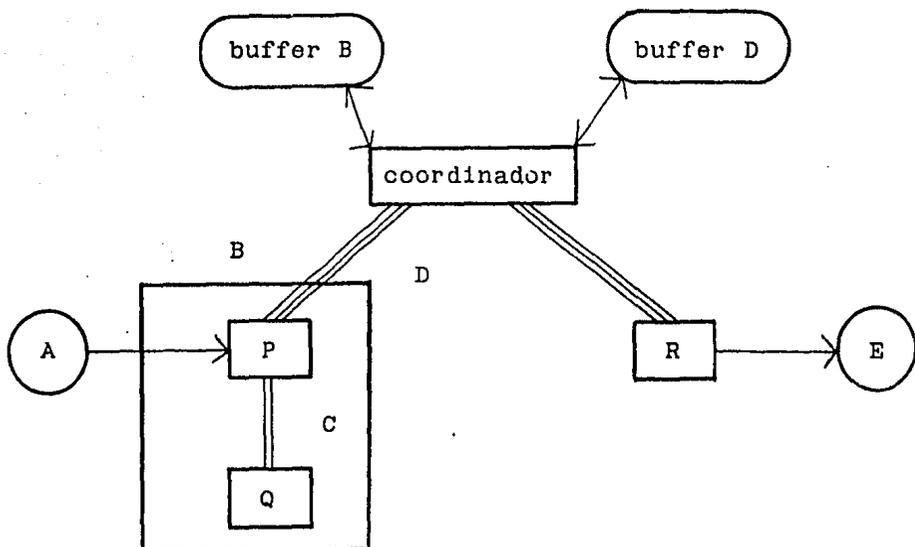
El orden de las operaciones de lectura ejecutadas en R, asumiendo que B y D son leídas antes y que inicialmente B es leída primero es :

lee B(1); lee D(13); lee B(7); lee B(13); lee B(18) ...

P no puede producir el registro B(18) hasta que no se hayan producido los registros C(5), C(10) y C(17). De esos registros, Q produce los registros D(13), D(18) y D(25). Al punto en la ejecución del sistema en donde R leerá el registro B(18) y no ha leído el registro D(13) es necesario que :

- a) Los registros C(10) y C(17) sean grabados en un buffer, o;
- b) Los registros D(18) y D(25) sean grabados en un buffer, o;
- c) Los registros C(10) y D(25) sean grabados en un buffer.

El tamaño del buffer requerido en cualquier sistema depende de la coordinación de los procesos. Podemos escoger que P y Q corran hasta que hayan producido 1000 registros en cada flujo de datos B y D antes de correr R, por lo que necesitaríamos un buffer para 2000 registros, pero podemos coordinar el proceso para ahorrar este espacio. El siguiente es el diagrama de una posible implantación para el sistema:



No hay necesidad de un buffer para C, pues sus registros son pasados directamente de P a Q. El texto del proceso será :

```

COORDINADOR itr
REGISTRO seq
pregunta R;
LEE-R sel (lee B en R)
  R-LEE-B sel (buffer B no vacio)
    lee buffer B (reg. B);
    LLAMA A R (reg. B);
  R-LEE-B alt (buffer B vacio)
    pregunta P&Q;
    B-DESDE-PQ sel (escribe B en P&Q)
      LLAMA A P&Q (reg. B);
      LLAMA A R (reg. B);
    B-DESDE-PQ alt (escribe D en P&Q)

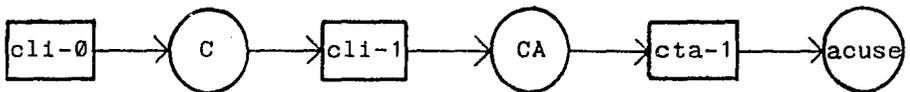
```

```

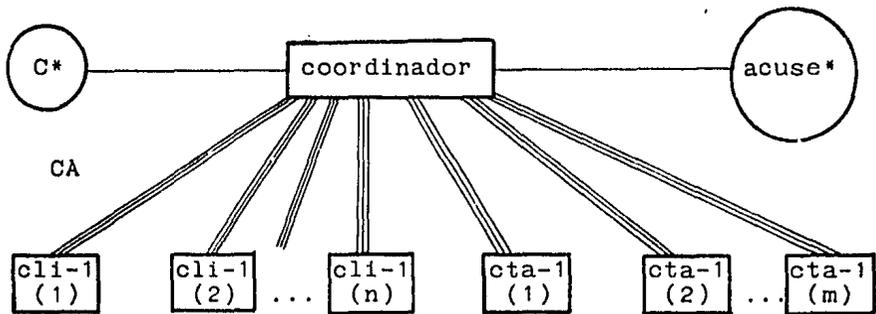
        LLAMA A P&Q (reg.D);
        escribe reg.D a buffer D;
    B-DESDE-PQ fin
    R-LEE-B fin
    LEE-R alt (lee D en R)
    R-LEE-D sel (buffer D no vacío)
        lee buffer D (reg. D);
        LLAMA A R (reg. D);
    R-LEE-D alt (buffer D vacío)
        pregunta P&Q;
        D-DESDE-PQ sel (escribe D en P&Q)
            LLAMA A P&Q (reg. D);
            LLAMA A R (reg. D);
        D-DESDE-PQ alt (escribe B en P&Q)
            LLAMA A P&Q (reg. B);
            escribe reg. B a buffer B
        D-DESDE-PQ fin
    R-LEE-D fin
    LEE-R fin
    REGISTRO fin
    COORDINADOR fin

```

8.5 SEPARACION DE VECTORES-ESTADO.- Un sistema sencillo de cuentas tiene el siguiente SSD :



Existen muchos clientes, cli-0, modelados por el proceso cli-1. Las transacciones en una cuenta son reconocidas por una función interna que escribe un flujo de salida "acuse". Queremos implantar este sistema en un sólo procesador. Un posible SID es el siguiente : (Ver diagrama en sig. página).

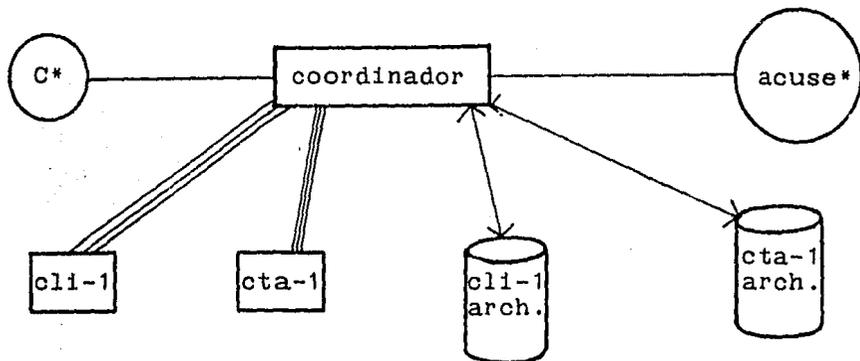


El proceso cli-1 está invertido con respecto a C y CA, el proceso cta-1 está invertido con respecto a CA y a accuse. Según el SDI anterior existe una subrutina cli-1 para cada cliente y una subrutina cta-1 por cada cuenta, por lo que es obvio que existe una falta de economía, realmente sólo necesitamos una copia de cli-1 y cta-1, y para lograrlo debemos :

- a) Separar los vectores-estado del texto del programa.
- b) Grabar los vectores-estado en algún lugar accesible por el controlador.
- c) Traer juntos el texto requerido y el vector-estado cuando un proceso es activado.
- d) Reemplazar el nuevo valor del vector-estado cuando el proceso es suspendido.

Tomando en cuenta lo anterior, podemos representar la implantación requerida como sigue :

(estructura en la siguiente página)



Cli-1 y cuenta-1 son los archivos maestros de clientes y cuentas respectivamente implantados usando un manejador de base de datos o algún software que nos permita los accesos que deseamos. Cuando el coordinador activa cli-1 para el cliente 1234, éste debe ejecutar las siguientes acciones :

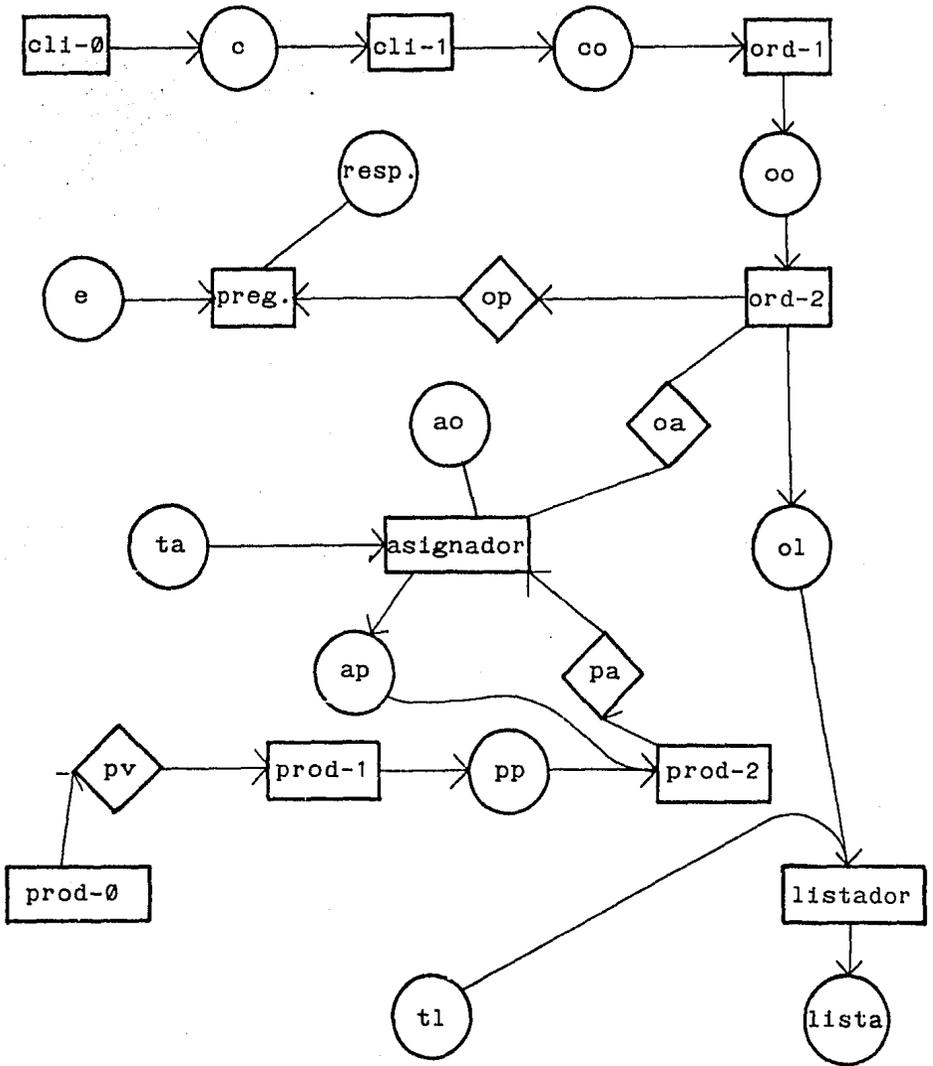
```

carga cli-1sv(1234) en CSV;
LLAMA A cli-1(CSV,...);
graba CSV en cli-1sv(1234);
  
```

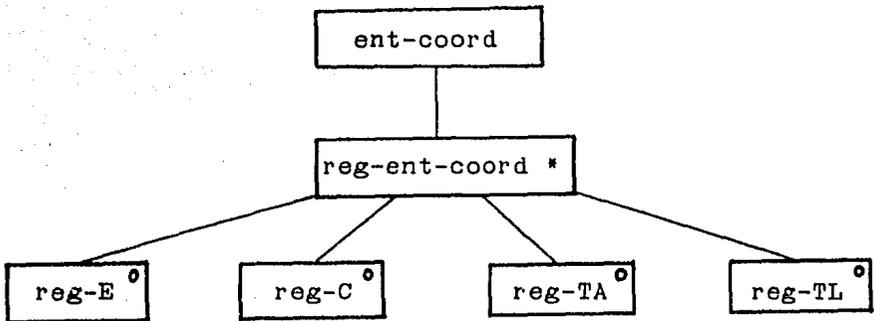
La operación de carga regresa el vector-estado contenido en cli-1 arch, la operación de graba reemplaza el nuevo valor después de la activación y suspensión de cli-1(1234). La identificación del cliente 1234 se supone que es un campo en el registro de entrada en C*, indicando que es parte del flujo de entrada C para el cliente 1234.

8.6 IMPLANTACION-1 "LO VENDO BARATO S.A.".- El SSD que implantaremos en un sólo procesador lo mostramos en la siguiente página.

Los flujos de datos de entrada al sistema son C, E, TA y TL. Suponemos que todas estas entradas son fusionadas en un sólo flujo de datos que será leído por el coordinador. Con un sólo procesador, es esencial que todos los flujos de datos en las fronteras del sistema sean leídos por el coordinador, teniendo entonces una estructura como la que mostramos a



continuación :

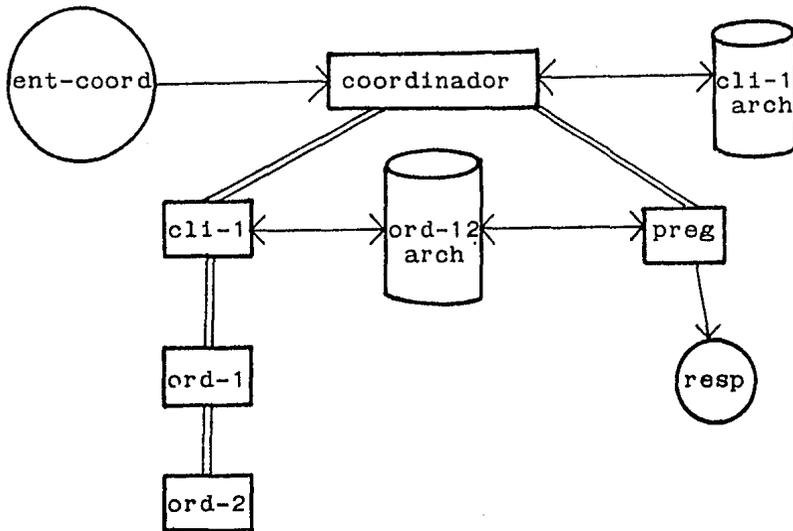


Cualquier restricción sobre ciertas entradas en ciertos tiempos es parte de la especificación y debe de ser tomada en forma explícita. Para nuestro sistema asumiremos que nuestros requerimientos son los siguientes :

- a) La contestación de resp debe de ser producida lo más pronto posible y deberá estar basada en un estado actualizado del modelo de orden.
- b) Asignar, debe ser ejecutada tan pronto sea posible después de un requerimiento (cuando reg-TA es recibido).
- c) La lista deberá ser producida tan pronto como sea posible, en forma consistente con los otros requerimientos.

Si la contestación de resp debe de ser producida lo antes posible, entonces el proceso preg deberá ser activado tan pronto como un registro reg-E esté disponible en el flujo ent-coord. Esto implica que preg deberá estar invertido y ser una subrutina del coordinador. Si la respuesta deberá estar basada en un estado actualizado del modelo de orden, entonces

los procesos cli-1, ord-1 y ord-2 deberán, similarmente, ser coordinados desde que la entrada esté disponible. La manera más directa de lograr esto es hacer a cli-1, invertido con respecto a C, una subrutina del coordinador, y permitir a cli-1 coordinar ord-1 e indirectamente a ord-2 cada vez que es producido un registro CO. Ord-2 es invertido con respecto a su flujo de datos fusionado OO&AO, la implantación de esto podría ser como sigue :

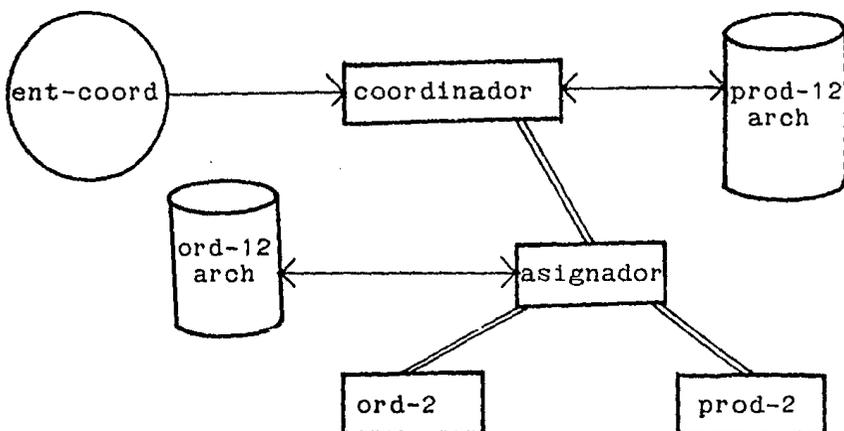


El flujo de salida OL de ord-2 no aparece en esta implantación parcial porque los registros de OL son asignar y retrasar y no pueden ser escritos como resultado del registro de entrada OO, que contiene ordenar, corregir, cancelar y entregar.

El proceso de asignador es más complejo, cuando recibe un registro reg-TA ejecuta las siguientes operaciones :

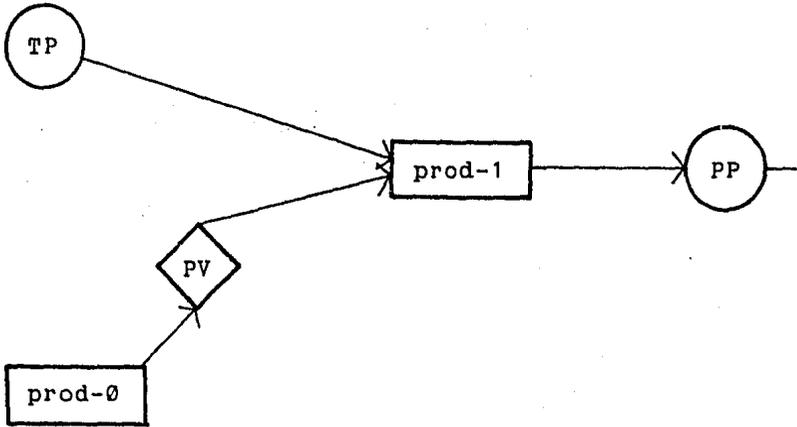
- a) Inspecciona PA, determinando el stock disponible para su asignación.
- b) Para aquellas ordenes para productos relevantes y están aguardando para su asignación :
 - Inspecciona OA para determinar la cantidad requerida.
 - Escribe retraso o escribe asignar a AO, dependiendo si la asignación se puede hacer o no.
- c) Escribe un registro AP, indicando el monto asignado

Mientras exista un asignador para cada instancia de prod-1 y prod-2, podremos combinar sus vectores-estado, como se hizo con ord-1 y ord-2, resultando un vector-estado prod-12. Invertiendo prod-2 con respecto a su fusión tosca de entrada AP&PP, podemos presentar su implantación parcial :



Prod-1, como lo especificamos en la etapa del modelo inicial, itera inspeccionando el vector-estado de prod-0. Cuando detecta que una acción de disponible ha ocurrido para

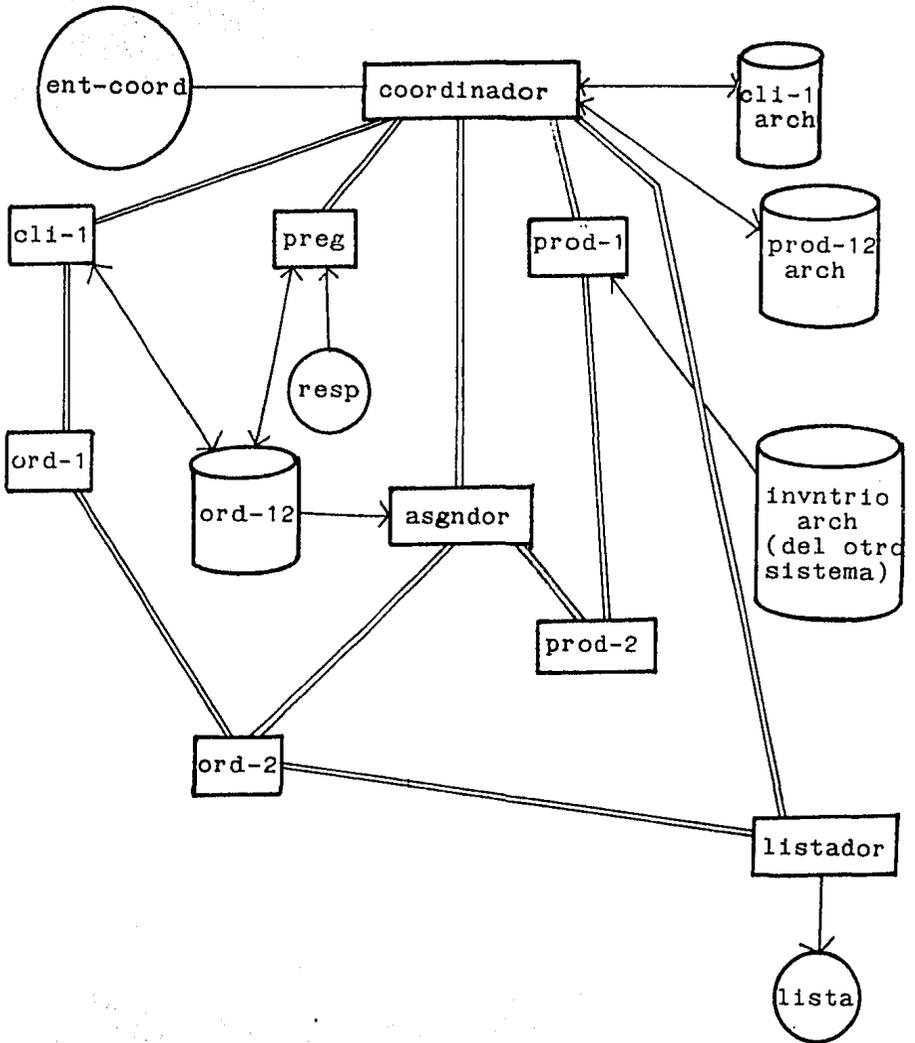
prod-0 escribe un registro PP. Si activamos prod-1 éste no se suspenderá por sí mismo antes de la siguiente acción de disponible por lo que monopolizará durante un lapso de tiempo inaceptable. Tomando en cuenta lo anterior necesitamos cambiar prod-1 de tal forma en se pueda suspender, esto lo podemos hacer dándole otro flujo de datos con respecto al cual esté invertido. Si llamáramos a este flujo de datos TP el SSD modificado aparecería como sigue :



Prod-1 es activado en cualquier momento en el que un registro TP sea escrito. Los registros TP no son parte de los registros de entrada al sistema, por lo que son escritos en forma interna de acuerdo con la activación deseada para prod-1. Parece razonable que prod-1 sea activado cuando asignador sea ejecutado, por lo que el registro TP deberá ser escrito por el coordinador cuando un registro reg-TA sea escrito al proceso de asignador. Prod-1 es invertido con respecto a su registro de entrada TP y llamará a prod-2 cuando escriba un registro PP. prod-2 es una subrutina de asignador y prod-1.

Sólo nos falta considerar al proceso listador, al cual debemos invertir con respecto a su fusión tosca de entrada OL&TL, haciéndolo una subrutina del coordinador y de ord-2.

El resultado de la implantación se muestra como sigue:



C O N C L U S I O N E S

CONCLUSIONES DE ESTA INVESTIGACION

Una vez expuestas dos de las técnicas existentes para el desarrollo de sistemas de información computarizados (Técnica Estructurada y Técnica JSD) se considera pertinente emitir tanto una opinión como una visión general acerca del tema tratado a lo largo de este trabajo.

Darémos a conocer entonces las principales conclusiones a las que se han llegado como producto del análisis realizado en este trabajo. Dichas conclusiones girarán en torno a los siguientes puntos:

1. Metodología Propuesta por cada Técnica para el Desarrollo de Sistemas de Información.
2. Analogías encontradas entre ambas Técnicas.
3. Diferencias.
4. Ventajas que ofrece el aprovechar ambas Técnicas.
5. Problemática Nacional de los Profesionales y Técnicos del área al desarrollar un Sistema de Información Computarizado.

A continuación se desarrolla cada punto:

1. METODOLOGIA PROPUESTA.- Al haber expuesto con relativa amplitud la metodología que sigue cada una de las técnicas para generar un sistema de información, queremos hacer notar que ambas técnicas tienen una forma muy opuesta de abordar una misma cuestión: Como generar un sistema eficiente.

Con lo anterior no queremos decir que una técnica sea mejor que la otra sino que únicamente son dos maneras de abarcar una misma cosa. Por tanto, es nuestro deseo dejar asentado que no debemos tomar como absoluta a una de las muchas técnicas que puedan existir para generar sistemas sino que es importante que el profesional del área posea una visión y un criterio más amplio para ir más allá del uso

constante de una sola técnica común y tradicional. Lo anterior sólo se logrará con la difusión y el conocimiento teórico y práctico de las demás técnicas existentes.

Con lo anterior, se deja abierta la inquietud de seguir investigando más acerca de este tema para esfumar así de nuestra mente la idea de que existe una sola forma de desarrollar sistemas de información, una sola forma tradicional (los diagramas de flujo, la programación estructurada, etc. por ejemplo, que son llevadas en forma aislada) sin ir más allá de esa frontera.

A lo largo de esta investigación, nuestro objetivo sufrió algunas modificaciones. En principio se intentaba proporcionar los elementos de juicio necesarios, por medio de la confrontación de las técnicas de desarrollo de sistemas de información, para así poder decidir cual de ellas sería la más adecuada en aplicaciones específicas; ahora, consideramos aún más importante la meta de crear una conciencia entre los presentes y futuros profesionales, que coordinarán o bien que formarán parte de un equipo de desarrollo de sistemas, de que no existe tan sólo una forma de desarrollar sistemas sino que tengan un criterio más amplio para aceptar nuevas ideas o técnicas de vanguardia que podrían mejorar la calidad de su trabajo y los frutos del mismo si se aplican adecuadamente. Lo anterior representa un reto pero vale la pena lograrlo.

2. ANALOGIAS.- Las dos técnicas propuestas en este trabajo coinciden en los siguientes puntos:

a) Tienen por objeto garantizar que el sistema refleje correctamente al mundo real, para lo cual emplean los modelos previos.

b) Del inciso anterior deriva la meta en común de ambas técnicas de crear un sistema cuya característica principal sea el prevenir las futuras necesidades con la finalidad de evitar cambios radicales en las especificaciones funcionales del sistema y con ello excesivos costos de mantenimiento. Ambas técnicas pretenden crear un sistema flexible que sea capaz de soportar la adhesión de más funciones en forma posterior.

c) El sistema y sus especificaciones deben realizarse en torno a las necesidades del usuario , es decir, ambas técnicas hacen hincapié en tomar en cuenta al usuario al definir las fronteras del sistema en base a la parcialidad del mundo real que se abarcará.

d) Por último, las dos técnicas remarcan la importancia de que el desarrollador no debe prestar atención prematura a la implantación del sistema sin antes haber realizado un estudio adecuado y un modelo objetivo que refleje la realidad. En términos más sencillos, lo anterior significa que no debe pensarse en la programación del sistema hasta que no se tenga bien definido el modelo del mismo y hasta que su desarrollo no esté bien planeado.

3. DIFERENCIAS.- Sólo basta con leer cada una de las técnicas para saber que son dos filosofías diferentes al atacar un mismo problema. El objeto de esta sección es el de enumerar algunos de los puntos más importantes en los que se diferencian:

a) La Técnica Estructurada hace mucho énfasis en el respaldo formal del desarrollo, esto es, que da mucha importancia a la documentación que se genera durante el proceso y a la comunicación del equipo de desarrolladores con la administración del sistema en cuanto a costos, avances, programación de actividades, eficiencia, etc., en cambio, la Técnica JSD no toma en cuenta el aspecto formal del desarrollo sino sólo se concreta a incluir a los usuarios al modelar la parcialidad del mundo real de interés sin detenerse mucho en la elaboración de reportes o documentos que comuniquen a una administración de los avances y planes.

b) Así como la técnica estructurada se preocupa mucho por generar reportes e informes a la administración y a cumplir con criterios de éxito establecidos, toma en cuenta por el contrario, en una forma relativamente superficial a las implicaciones del hardware y del software que conllevará el sistema en su desarrollo, describiendo en forma muy tradicional el aspecto del diseño detallado del sistema (pues sólo habla de entradas-procesos-salidas). En cambio, una vez que JSD estudia y analiza a fondo la parcialidad del mundo real, intenta crear un modelo en el que se combine la realidad con

el mundo del computador (canales de comunicación entre el mundo real y el proceso o entre diferentes procesos en un ambiente de multiproceso o proceso distribuido, comunicaciones a través de un flujo de datos y vector-estado respectivamente). También, considera la problemática de la consistencia de una base de datos en este tipo de ambientes.

c) Las herramientas de las que se valen ambas técnicas para dar forma a sus modelos son diferentes, pues mientras una maneja diagramas de flujo de datos, diagramas de bloques, diagramas de flujo tradicionales, cartas HIPO, etc. la otra hace uso de nuevos conceptos de diagramación y estructuración de los procesos.

4. VENTAJAS EN EL APROVECHAMIENTO DE AMBAS TECNICAS.- Tomando en cuenta que la Técnica Estructurada incluye en el trabajo de desarrollo a aquellos aspectos que circundan al equipo de desarrolladores, como es la parte administrativa de una empresa, y que la Técnica JSD se centra más en la esencia misma del sistema a desarrollar desde un punto de vista más técnico, resultaría interesante que el profesionista que se dedique al desarrollo de sistemas pueda combinar ambas técnicas siempre teniendo una mentalidad abierta para poder aprovecharlas sin cerrarse a una sola costumbre.

Lo anterior podría facilitarse si se formara a los futuros profesionales del área (desarrolladores) con una mentalidad más crítica y abierta que estuviera dispuesta al cambio.

5. PROBLEMATICA NACIONAL DEL PROFESIONAL EN EL AREA.- Actualmente en el ámbito de la informática, específicamente en el desarrollo de sistemas, nos encontramos con los siguientes problemas en los cuales se ven envueltas las personas que laboran en el área:

a) Algunas personas no conocen ninguna técnica para coordinar sus acciones a lo largo de la realización de un proyecto de sistemas y por tanto trabajan de manera empírica lo cual trae como consecuencia que los costos de desarrollo del sistema sean demasiado altos y que el resultado final no sea del todo satisfactorio.

b) En otros casos existen personas que conocen en cierta forma algunos conceptos de una determinada técnica pero sin dominarla en su totalidad, por lo cual no existe consistencia en sus acciones al coordinar o al elaborar un proyecto de sistemas.

c) Existen también otras personas que son reacias al cambio y demasiado conservadoras que no aceptan saber y conocer de nuevas ideas que puedan provocar a la larga una mejoría en su trabajo.

d) Debido a que la mayoría de las bases técnicas y conocimientos en computación son importados, y que por ende, para que los presentes y futuros profesionales del área en el país conozcan y manejen estos conocimientos, es necesario que dominen el idioma en el que están escritas estas bases documentales y que tengan el grado de preparación necesario y previo para así alcanzar un nivel de comprensión aceptable acerca de estos nuevos conocimientos importados con el fin de poder aportar, agregar o modificar tales bases y poder adaptarlas a nuestra realidad nacional, lo cual requiere de un esfuerzo aún mayor.

B I B L I O G R A F I A

BIBLIOGRAFIA

1. Awad Elías. SYSTEMS ANALYSIS AND DESIGN. U.S.A., 1979. - Richard D. Irwin Inc., 370 p.p.
2. CIFCA (Centro de Informática de la Facultad de Contaduría y Administración UNAM). HACIA EL DESARROLLO DE SISTEMAS CON HERRAMIENTAS AUTOMATICAS (Boletín).Febrero de 1985. Pag.1
3. CIFCA (Centro de Informática de la Facultad de Contaduría y Administración UNAM). LA CAPACITACION COMO UN RECURSO DE DIRECCION EN LOS CENTROS INFORMATICOS (Boletín). Enero de 1983.
4. Churchman West. EL ENFOQUE DE SISTEMAS. U.S.A. 1973. 1a Ed. 270 p.p.
5. Cotterman W. William,et.al. SYSTEM ANALYSIS AND DESIGN: A Foundation For the 1980's. U.S.A. 1981. North Holland . 553 p.p.
6. Hice Turner y Cashwell. SYSTEM DEVELOPMENT METHODOLOGY. U.S.A. 1981. North Holland. 553 p.p.
7. Instituto Nacional de Estadística Geografía e Informática, Secretaría de Programación y Presupuesto y Universidad - Nacional Autónoma de México. LA INFORMATICA A FUTURO EN MEXICO (Memorias del ciclo de conferencias 1983). PUC. UNAM. Feb. de 1983. 251 p.p.
8. Jackson Michael. SYSTEM DEVELOPMENT. U.S.A. 1983. Prentice Hall International. 418 p.p.
9. Murdick G. Robert. SISTEMAS DE INFORMACION BASADOS EN COMPUTADORAS PARA LA ADMINISTRACION MODERNA. México,1974 (Traducción al Español). Ed. Diana. 1a Ed. 638 p.p.
10. O.N.U. MANUAL DE PROYECTOS DE DESARROLLO ECONOMICO. O.N.U. México 1958. 246 p.p.
11. Ramos Elorduy Margarita. MANEJO DE ARCHIVOS EN COBOL. México 1983. Programa Universitario de Cómputo. 3a Ed. 251 p.p.

12. Rojas Soriano Raul. GUIA PARA REALIZAR INVESTIGACIONES SOCIALES.

13. S. Orilia Lawrence. INTRODUCCION AL PROCESAMIENTO DE DATOS PARA LOS NEGOCIOS. México 1984. (traducción al Español) Mac Graw-Hill. 2a Ed. 744 p.p.

14. W. Metzger Philip. ADMINISTRACION DE UN PROYECTO DE PROGRAMACION. México, 1978. Trillas 1a Ed. 226 p.p.

15. William S. Davis. SYSTEMS ANALYSIS AND DESIGN: A Structured Approach. Miami University, Oxford Ohio, 1983. Addison-Wesley Publishing Company. 415 p.p.

14. Yourdon Edward, et.al. STRUCTURED DESIGN: Fundamentals of a Discipline of Computer Program and System Design. U.S.A. 1979. Prentice Hall Inc. 473 p.p.