

7
29

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS



LA IMPORTANCIA DEL DISEÑO ESTRUCTURADO EN EL DESARROLLO DE SISTEMAS DE INFORMACION

T E S I S

QUE PARA OBTENER EL TITULO DE :

M A T E M A T I C O

P R E S E N T A :

MARIA DE LUZ GASCA SOTO

FALLA DE ORIGEN



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

INTRODUCCION	1
CAPITULO UNO: TEORIA GENERAL DE SISTEMAS	1
*Antecedentes	2
*Características	4
*Conceptos importantes	5
*Mejoramiento y diseño	10
CAPITULO DOS: SISTEMAS DE INFORMACION	15
*Conceptos importantes	16
*Sistema de Información	17
*Computación y Sistemas de Información	18
*Ciclo de vida	19
*Funciones básicas del ciclo de vida	22
— Viabilidad y planeación del proyecto	22
— Analisis y determinación de requerimientos	22
— Diseño	23
— Desarrollo e instalación	23
— Evaluación y mantenimiento	23
CAPITULO TRES: MODULARIDAD Y BASES DE DATOS	25
*Modularidad	26
— Conexión entre módulos	28
*Bases de Datos	29
— Estructura y características	30

INDICE

CAPITULO CUATRO: DISEÑO ESTRUCTURADO	35
*Introducción	36
*Técnicas del Diseño Estructurado	40
— Cajas Negras	40
— Diagrama Jerárquico Funcional	43
— Diagramas de Flujo de Datos	50
— Diagramas de Warnier /Orr	60
— Diagramas H. I. P. O.	67
CAPITULO CINCO: Desarrollo de un Sistema de Información	70
*Análisis de Requerimientos	71
*Metodología	74
*Diseño	75
— Diseño de la Base de Datos	75
— Modularización del Sistema	86
+Descripción del objetivo de cada módulo	87
— Diseño en detalle del Sistema	90
+Descripción detallada de los procesos	92
CONCLUSIONES	102
BIBLIOGRAFIA	106
APENDICE A: Diagramas de Warnier/Orr.	111
APENDICE B: Diagramas de Flujo de Datos.	127
APENDICE C: Cajas Negras.	143
APENDICE D: Ejemplos de Modificaciones.	149

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

INTRODUCCION

Ante el gran auge de la computación en el mundo actual y la enorme demanda de elaboración de programas que auxilien y faciliten el manejo de una organización, resulta más que una necesidad, una OBLIGACION la planeación detallada de programas o paquetes de programas (llamados sistemas) que se requieran.

Cuando se tiene un problema, de cualquier tipo sin importar la magnitud, generalmente, se aconseja: "pensar bien las cosas antes de hacerlas". Es decir, lo ideal es pensar bien el cómo resolverlo, ese es el secreto. Primero se debe analizar bien la situación, tanto la actual como la anterior, después es necesario identificar perfecta y detalladamente el problema para así saber por donde se podría empezar a atacarlo, esto proporcionará varias soluciones dando opción a elegir la que mejor convenga: cuál proporcionará más ganancias o, en el peor de los casos, menos pérdidas.

Es casi natural pensar en una planeación antes de una realización, pero en programación esto rara vez sucede. Usualmente se realizan los programas sin tener diseño alguno que indique lo que habrá de realizar para llegar a un objetivo dado, no se prevén casos extremos (raros o extraños), ni se toma en cuenta la posibilidad de ampliar, modificar o mejorar el proyecto para el futuro.

INTRODUCCION

Todos estos detalles que quedan por ahí 'flotando' traerán graves consecuencias para el usuario desde frecuentes fallas en diferentes 'ejecuciones' del sistema hasta la pérdida y alteración de la información.

Si antes de hacer un programa (o sistema) tomamos en cuenta qué tipo de herramientas se tienen y con cuáles se puede trabajar (tipo de computadora e impresora, lenguajes de programación y paquetes auxiliares así como el equipo humano, entre otras cosas) entonces podremos empezar a planear un sistema.

Para ello es necesario saber QUE hará tal proyecto, es decir, se deben identificar sus metas seleccionando las que tengan realmente utilidad (que no sean fantásticas o absurdas), aquí es vital la comunicación con el usuario. Una vez definidos los objetivos es preciso saber COMO se logrará "dar vida" a éstos (diseño del sistema), es decir, cómo lograremos que la computadora los realice eficientemente previendo cualquier tipo de situación extraña que pueda provocar un mal comportamiento del sistema. Será responsabilidad del programador la validación total de la información que maneje en el sistema. Finalmente se debe avisar que todo el proyecto funcione como el usuario lo espera.

Cuando un sistema ha sido planeado la posibilidad de que falle es casi nula porque durante la planeación (análisis y diseño) se han visto las posibles soluciones con sus consecuencias y de ahí se ha elegido la solución óptima.

Ahora bien, existen algunos programadores muy experimentados que quizá crean darse el lujo de no diseñar un sistema. En realidad, considero, nadie puede darse tal lujo. En primer lugar porque la computación cambia frecuentemente y siempre hay que estar al día y en segundo lugar para la gente con más experiencia ya es casi natural el diseño, tal vez no lo formalizan pero tienen bien claro qué harán, cómo lo harán y quiénes le ayudarán.

Es obvio que cualquiera puede jugar al arquitecto y construir una casa, pero... ¿cuánto tiempo podrá durar ésta, algunos años, unos meses o tan sólo unos cuantos días?

Así pues, la importancia de hacer un diseño es inmensa y el secreto para lograr un sistema óptimo es PLANEAR ANTES DE REALIZAR. El diseño estructurado es una metodología que ayuda a elaborar mejores diseños, pero no implica que evite los 'malos', la clave está en cómo manejar la metodología y sus técnicas, pero eso solo se logra con la experiencia.

El propósito de este trabajo es difundir el área de los Sistemas de Información entre los estudiantes interesados en la computación, así como dar apoyo a cursos relacionados con los sistemas (Metodología y Análisis de Sistemas, Sistemas de Información, Seminario de Bases de Datos, entre otros) e insistir en la importancia del diseño de sistemas independientemente del área de trabajo.

De lo anterior, en la tesis se plantea *La Importancia del Diseño Estructurado en el Desarrollo de Sistemas de Información*, pero para tratar sobre la importancia de algo primero debemos saber que es ese algo, dónde actúa y qué es eso sobre lo que actúa.

El presente trabajo ha sido dividido en varios capítulos y apéndices procurando identificar de manera individual los conceptos generales. La idea fundamental es presentar la teoría necesaria que dé apoyo a la teoría del Diseño Estructurado. Se continúa con un pequeño sistema que trata al máximo de ejemplificar la teoría dada y de clarificar su importancia. Finalmente la tesis queda estructurada de la siguiente forma.

INTRODUCCION

El capítulo I es acerca de la Teoría General de los Sistemas, ya que todo lo relacionado con el Diseño y Análisis de Sistemas de Información ha sido fundamentado en ésta. El capítulo II da una panorámica global de lo que son los Sistemas de Información y su Ciclo de vida definiendo sus diferentes etapas. El capítulo III trata de manera general los conceptos acerca de la Modularidad y Bases de Datos así como una simple metodología para crear bases de datos. El capítulo IV es sobre el Diseño Estructurado y algunas de sus técnicas, las cuales se definen e ilustran con ejemplos. En el capítulo V se describe el diseño del sistema propuesto para ejemplificar la teoría desarrollada. Los apéndices contienen algunos algoritmos y programas del sistema del capítulo V que se consideran importantes para ejemplificar el desarrollo.

C A P I T U L O

U N O

TEORIA GENERAL DE SISTEMAS

CAPITULO

U N O

TEORIA GENERAL DE SISTEMAS

ANTECEDENTES

El universo está organizado por sistemas, algunos simples y otros extremadamente complejos, como el universo mismo. Algunos de ellos han sido creados o estructurados por el ser humano y otros por la naturaleza.

Ante esto, es natural que vivamos rodeados de sistemas (como máquinas, seres vivos, el hombre mismo, instituciones y otros tipos de organizaciones, problemas sociales, tecnológicos y matemáticos, etcétera) los cuales requieren de ser reorganizados, rediseñados y hasta mejorados; otros deben ser resueltos y algunos otros deberán ser creados y organizados para ayudar a los ya existentes. Para realizar tales operaciones se requieren de técnicas complicadas que van más allá de los alcances de cualquier científico.

Con frecuencia escuchamos, en cualquier conversación, la palabra *sistema*, la cual es utilizada en diferentes campos, desde los tópicos industriales, no muy complicados, hasta los temas más selectos de la ciencia. Lo cual significa que tal concepto ha invadido todas las áreas de la ciencia y se ha integrado, sin dificultad alguna, al pensamiento y lenguaje popular, así como a los medios de comunicación. Todo este "complejo sistemático" hace necesaria la creación de una metodología con la cual dado un objetivo específico se pudiesen encontrar procedimientos para alcanzar tal objetivo y que además se pudiesen considerar varias soluciones factibles de donde se obtendría la óptima con máxima eficiencia y mínimo costo.

CAPITULO I : TEORIA GENERAL DE SISTEMAS

Hay, actualmente, modelos y principios aplicables a los sistemas sin importar su género particular, la naturaleza de sus componentes ni la relación que exista entre ellos. Tales modelos están envueltos en una disciplina llamada *Teoría General de Sistemas* la cual fórmula principios válidos para los sistemas en general e intenta evitar la inútil repetición de esfuerzos.

La *Teoría General de Sistemas* es un instrumento útil al otorgar modelos utilizables y transferibles en diferentes campos y evitar analogías que perjudiquen el progreso de tales campos.

De acuerdo con Bertalanffy [1]: "*La Teoría General de Sistemas es una ciencia general de la totalidad. Es una disciplina lógico-matemática puramente formal en sí misma pero aplicable a varias ciencias empíricas.*"

Sin embargo, el enfoque matemático no es el único ni el más amplio para poder generalizar la *Teoría General de Sistemas*, existen otros enfoques afines tales como: la teoría de la información, la cibernética, la teoría de juegos, la teoría de las decisiones, los procesos estocásticos y la investigación de operaciones

Pero en realidad, como asegura Bertalanffy [1], "... el problema de los sistemas es esencialmente el problema de las limitaciones de los procedimientos analíticos de la ciencia. Proceder analítico significa que una entidad investigada es resuelta en partes unidas, a partir de las cuales puede, por tanto, ser constituida y reconstruida, entendiéndose estos proceder en sus sentidos tanto material como conceptual."

Para resolver problemas se requiere de una amplia visión que abarque el panorama total del problema y no sólo una parte aislada del mismo. Es decir, es necesario también resolver los problemas decisivos de la organización y el orden que los unifica, los cuales son resultado de la interacción dinámica de las partes y hace posible un comportamiento de éstas cuando se estudian aisladas o dentro del todo.

CAPITULO I : TEORIA GENERAL DE SISTEMAS

Es indispensable darle a los problemas un enfoque general tomando en cuenta el medio que los rodea y su constante interacción con él para así hacer más factible la posibilidad de tomar decisiones apropiadas para su correcta solución.

Además de enfocarse sobre todas partes de un sistema la Teoría General de Sistemas ayuda a la comunicación entre los especialistas de diferentes campos.

Segun John P. van Gigch (7) "La complejidad de los sistemas es el resultado de la interacción del hombre en los sistemas." Generalizando se puede asegurar que la complejidad de un sistema está relacionada con su interacción sobre otros sistemas.

CARACTERISTICAS DE LOS SISTEMAS

Un sistema es un conjunto de componentes que interactúan para alcanzar un objetivo. El objetivo de un sistema es la razón de su existencia; para cumplir con sus propósitos, generalmente interactúan con su medio ambiente, es decir, son observados y analizados dentro de un sistema mayor que los contiene y, que de alguna forma, determina su desarrollo. El concepto de interacción dentro del medio ambiente, de algunos sistemas, es esencial para su control.

Si un sistema está constituido por conceptos, entonces se tratará de un sistema conceptual, por ejemplo, un lenguaje es un sistema conceptual.

Un sistema puede estar integrado por objetos, cualquier máquina está integrada por objetos — automóvil, radio, computadora, television, etcétera

Los elementos de un sistema pueden ser sujetos, como el personal de una empresa o los integrantes de un equipo deportivo.

CAPITULO I : TEORIA GENERAL DE SISTEMAS

Resumiendo, un sistema puede estar formado por conceptos, objetos, sujetos o cualquier combinación entre ellos

Un sistema hombre-máquina está compuesto por sujetos (el hombre), objetos (la máquina) y conceptos (el medio de comunicación entre el hombre y la máquina).

Es claro, además, que un sistema puede constituirse por otros sistemas más pequeños a los que se les llamará subsistemas, y varios de estos interactúan con otros (sistemas o subsistemas) para así alcanzar la meta propuesta, lo cual es común en casi todos los sistemas.

CONCEPTOS IMPORTANTES DE LOS SISTEMAS

Un sistema está rodeado de diferentes entes los cuales forman parte de su medio ambiente y que, además, determinan su control y sobrevivencia tales como: los elementos que lo forman, los recursos con que cuenta, los datos que requiere, el proceso de conversión, las estructuras, los estados y los flujos de los elementos, los resultados que deberán obtenerse, los objetivos que se deben alcanzar, las funciones y atributos del sistema, además de su administración, los autores y agentes responsables de las decisiones a tomar.

Cada uno de los conceptos anteriores deben ser considerados y perfectamente identificados para el buen desarrollo del sistema, la búsqueda de la solución óptima. Tales conceptos se definen a continuación.

* Elementos.

Son las componentes del sistema, que pueden ser objetos, sujetos, conceptos, sistemas o subsistemas. Generalmente los sistemas están constituidos por alguna combinación de los anteriores, como ya se ha mencionado.

CAPITULO I · TEORIA GENERAL DE SISTEMAS

Los elementos que entran al sistema, o que habrá de necesitar el sistema para laborar son llamados *Entradas*, y los que los dejan, o que son obtenidos después del proceso son denominados *Salidas* o *Resultados*.

El cuerpo humano es un sistema y consta de varios subsistemas como: el sistema nervioso, el digestivo, el respiratorio y el muscular entre otros. Cada uno de ellos a su vez tienen subsistemas. El hombre es, además, un subsistema de un *algo* más grande que lo posee: su medio ambiente, el universo.

Un automóvil es, también, un sistema cuyos subsistemas son: el sistema de arranque, el de calefacción y el de locomoción entre otros que interactúan para lograr el objetivo principal que es mover el auto y dar comodidad a su dueño.

* Entradas y recursos.

Existe tan sólo una sutil diferencia entre las *entradas* y los *recursos*, depende únicamente de las circunstancias y del punto de vista. Durante el proceso de conversión, generalmente, sobre las *entradas* se aplican los *recursos*.

Los recursos son todo lo que hace que el sistema este disponible para ejecutar las actividades necesarias para alcanzar sus metas. En contraste con el medio ambiente los recursos están dentro del sistema y están bajo control.

Por ejemplo en un sistema educativo, los alumnos son *entradas* y los profesores *recursos*, en un sistema deportivo los jugadores son *entradas* y los *recursos* son las estrategias y técnicas que utilizan para alcanzar su objetivo.

Quizá analizando en un contexto más amplio se puede contemplar que los estudiantes con educación se convierten en *recursos* para una sociedad y los jugadores con técnicas y estrategias llegan a ser *recursos* para un equipo.

CAPITULO I : TEORIA GENERAL DE SISTEMAS

Debe también especificarse si las entradas y recursos del sistema estarán o no bajo control del diseñador, para así determinar si son parte del sistema o de su medio ambiente.

* Proceso de conversión.

Es el proceso por medio del cual los elementos del sistema pueden cambiar de estado, en general cambia las entradas en salidas

En un sistema organizado es muy frecuente que los procesos de conversión proporcionen atributos extras a las entradas convirtiendolas así en salidas

* Salidas o resultados.

Las salidas son consecuencia del proceso de conversión del sistema y son también denominadas resultados, éxitos o beneficios.

* Medio ambiente.

Es indispensable reconocer y determinar los límites de los sistemas y su interrelación con otros sistemas y así determinar con precisión cuales sistemas estan bajo control del diseñador y cuales deben ser considerados fuera de su autoridad

El sistema y su medio ambiente están interrelacionados y son interdependientes. El medio ambiente es externo al sistema, abarca todo lo que está fuera del control del sistema, también determina su desempeño, control y superviuvencia.

Por ejemplo, el hombre es un sistema que depende íntegramente de su interrelación con su medio ambiente para su control y sobrevivencia; si existe una leve variación en la temperatura "normal", a la que él está habituado, quizá no le afecte mucho, pero aún así, el ambiente es diferente y cambia su funcionamiento.

Si el hombre se encuentra en un lugar donde la temperatura es muy extrema, entonces tendrá problemas de superviuvencia.

CAPITULO I : TEORIA GENERAL DE SISTEMAS

Los sistemas pueden ser mejor aprovechados cuando operan dentro de condiciones tolerables de rendimiento; si existen condiciones extremas durante un largo tiempo, el sistema puede fallar totalmente.

Existen sistemas que pueden ajustar sus actividades bajo algunas condiciones no deseables y seguirán funcionando, pero hay otros que no podrán hacer el ajuste y entonces se detendrán.

Como un ejemplo de lo anterior está la televisión que es un sistema que para su mejor aprovechamiento depende totalmente de las condiciones bajo las cuales trabaje: Si ésta se enciende en un lugar donde hay mucha luz solar, no se podrá ver bien la imagen pero si es encendida en algún lugar donde exista muy poca luz, entonces la imagen será mejor apreciada y esto no depende de que si la televisión es nueva o antigua, o de que sea de buena calidad o no.

* Metas y objetivos.

Un sistema tiene metas u objetivos los cuales con frecuencia son difíciles de observar. Los objetivos de una máquina, por ejemplo, son claros, pero ¿cuáles son los objetivos de un sistema social?

Es sumamente importante identificar de manera precisa las metas y objetivos de un sistema para su desarrollo, al ir eliminando el grado de abstracción de la definición del problema los objetivos estarán mejor definidos y serán más manejables. Implantar los objetivos del sistema está estrechamente ligado con el establecimiento de los límites del sistema, pues ambas tareas habrán de determinar su desempeño.

* Propósitos y funciones.

Son tareas específicas que debe realizar el sistema. Algunos sistemas, generalmente aquellos cuyos elementos son objetos, carecen de un propósito evidente y adquieren uno específico cuando se interrelacionan con otros sistemas.

CAPITULO I : TEORIA GENERAL DE SISTEMAS

Por ejemplo, una máquina de escribir no tiene una tarea por realizar hasta que algún ser humano, el cual es un sistema, la utilice y escriba algo por medio de ella.

* Control.

El control se ocupa de la ejecución de los planes o tareas. Asociado con el control está el flujo de la información y la retroalimentación de la misma, de tal modo que un sistema pueda evaluar sus planes.

* Atributos o propiedades.

Los sistemas, subsistemas y sus elementos están provistos de atributos o propiedades las cuales pueden ser cualitativos o cuantitativos. En algunos casos los atributos son usados como sinónimos de "mediciones de eficacia"

* Componentes, programas y misiones.

Un sistema está formado por componentes que son tareas, actividades o partes del sistema (programas o misiones) las cuales son desempeñadas para alcanzar los objetivos trazados anteriormente.

* Estructuras.

La estructura está relacionada con la interacción e interrelación que sostienen los elementos del sistema, puede ser simple o compleja, depende del número y tipo de interrelaciones y del nivel de la jerarquía.

* Estados y flujos.

El estado de un sistema es definido por las propiedades que muestran sus elementos en un punto fijo en el tiempo. Es decir, el estado de un sistema está dado por los atributos que adquieren sus elementos en alguna etapa del proceso de conversión.

Los cambios de un estado a otro por los que pasan los elementos de un sistema constituyen los flujos lo cuales se definen según los cambios del valor de los atributos.

CAPITULO I : TEORIA GENERAL DE SISTEMAS

* Conducta.

La conducta de un sistema es determinada por los cambios de los estados sobre el tiempo.

* Administración y planeación.

La administración de un sistema está formada por actividades dirigidas hacia la planeación y control del mismo.

La planeación abarca el fijar las metas, la utilización de recursos, el desarrollo de un programa encargado de realizar las diferentes actividades y de una estrategia que trate (se relacione) con el medio ambiente.

* Agentes y autores de decisiones

Las acciones y decisiones que suceden en un sistema son atribuidos y asignados a personas cuya responsabilidad es guiar al sistema hacia el logro de sus objetivos, tales individuos son conocidos como administradores, agentes y autores de decisiones.

MEJORAMIENTO Y DISEÑO DE SISTEMAS

Un problema usual, cuando se está trabajando con sistemas, es la incapacidad de diferenciar entre mejoramiento y diseño de sistemas.

El mejoramiento implica el cambio que lleva a un sistema a su condición de operación normal. El diseño, en cambio, es un proceso creativo que demanda una apartencia y enfoque totalmente nuevos.

* Mejoramiento de sistemas.

El mejoramiento de sistemas involucra determinar las razones de porqué el sistema no funciona como se esperaba, lo cual implica la existencia de un plan (diseño) anterior contra el cual puede compararse el funcionamiento real.

CAPITULO I : TEORIA GENERAL DE SISTEMAS

El mejoramiento que se haga a un sistema no ha de implicar que tal cambio sea, en cierto momento, bueno o malo, es decir, no tiene implicaciones éticas, es tan sólo un proceso que ha de asegurar que un sistema logre trabajar según las expectativas.

Ahora bien, un sistema habrá de modificarse porque no satisface los objetivos establecidos, no proporciona los resultados predichos y no opera como se planteó inicialmente.

Conociendo cual es el problema (es decir, qué parte del sistema está fallando) y cómo es que debe funcionar, se deben deducir algunas conclusiones tentativas de por qué esa parte del sistema está fallando o qué es exactamente lo que no funciona y cuáles podrían ser las formas de solucionar tales detalles.

Reorganizando lo anterior, se puede concluir que para el mejoramiento de sistemas se debe :

1. Definir el problema e identificar las componentes del del sistema que no funcionan correctamente, es decir, localizar de manera precisa las fallas.

2. Observar cuidadosamente los estados, condiciones y conductas actuales del sistema, para así determinar con exactitud su comportamiento en diferentes etapas del proceso.

3. Comparar, con enorme detalle, las condiciones y comportamientos reales del sistema con los esperados para determinar, de esta forma, el grado de desviación.

4. Obtener los supuestos argumentos del porque de la existencia de las fallas, todo ello de acuerdo con los límites de los subsistemas componentes.

5. Sacar conclusiones de los hechos conocidos, mediante un proceso de deducción y se debe desglosar el problema en subproblemas usando algún proceso de reducción.

CAPITULO I : TEORIA GENERAL DE SISTEMAS

* Diseño de sistemas

Diseñar un sistema significa crear una estructura que sea óptima, requiere de establecer los métodos que aseguren que él opere de acuerdo a la especificación del problema.

El enfoque de sistemas es básicamente una metodología de diseño que hace énfasis sobre el sistema total y no sobre sus componentes separadas, se esfuerza por optimizar la eficacia del sistema total, además provee el mejoramiento del mismo.

En el diseño de sistemas se debe :

1. Definir el problema en relación a sistemas en los cuales esté contenido el sistema en cuestión y en los cuales esté relacionado mediante metas comunes.

2. Establecer los objetivos del sistema en relación a sistemas mayores a él que lo contienen o al sistema total.

3. Evaluar los diseños de acuerdo con los costos de oportunidad o del grado de desviación del diseño óptimo del sistema.

4. Involucrar planeación, evaluación e implementación de nuevas alternativas que ofrezcan resultados innovadores y creativos para el sistema total, además de tomar en cuenta procesos de inducción y síntesis para así tener la factibilidad de obtener un diseño óptimo.

5. Concibir el planteamiento del sistema como un proceso por medio del cual el planificador asume el papel de líder (y no de seguidor, como en el mejoramiento).

En la figura 1.1 se presenta una tabla de comparación entre el mejoramiento y diseño de sistemas (7).

CAPITULO I : TEORIA GENERAL DE SISTEMAS

El enfoque de sistemas es la filosofía del manejo de sistemas en la cual se pretende que las soluciones tengan éxito para éstos en todas las situaciones posibles, en realidad tal enfoque no es más que la Teoría General de Sistemas Aplicada.

El análisis y diseño de sistemas para los de información está fundamentado en la Teoría General de Sistemas, la cual enfatiza la necesidad de examinar todas las partes de un sistema, frecuentemente el enfoque que se toma, al desarrollar un sistema, es sobre una o varias componentes del sistema y cuando alguna decisión es tomada, ésta puede ser ineficiente ya que varias componentes fueron ignoradas.

CAPITULO I : TEORIA GENERAL DE SISTEMAS

*Comparación de dos metodologías de cambio
Mejoramiento y Diseño de Sistemas*

	<i>Mejoramiento</i>	<i>Diseño</i>
<i>Condiciones del sistema</i>	<i>El diseño se implanta</i>	<i>El diseño se cuestiona</i>
<i>Intereses</i>	<i>Sustancia Contenido Causas</i>	<i>Estructura y proceso Método Propósitos y funciones</i>
<i>Paradigma (proceso)</i>	<i>Análisis de sistemas y subsistemas componentes. (Método analítico)</i>	<i>Diseño del sistema global (enfoque de sistemas)</i>
<i>Proceso de razonamiento</i>	<i>Deducción y Reducción</i>	<i>Inducción y Síntesis</i>
<i>Salida (Resultado)</i>	<i>Mejoramiento del sistema existente</i>	<i>Optimización del sistema global.</i>
<i>Método</i>	<i>Determinación de desviaciones entre operación intentada y real (Costos directos)</i>	<i>Determinación de entre el diseño real y el diseño óptimo (Costos de oportunidad)</i>
<i>Enfasis</i>	<i>Explicaciones de desviaciones del pasado.</i>	<i>Predicciones de resultados futuros.</i>
<i>Perspectiva</i>	<i>Introspectiva del sistema hacia el interior</i>	<i>Extrospectiva del sistema hacia el exterior</i>
<i>Papel del planificador</i>	<i>Seguidor satisfacer las tendencias reinantes.</i>	<i>Líder influir sobre las tendencias y modificarlas.</i>

Figura 1.1

C A P I T U L O

D O S

SISTEMAS DE INFORMACION

CAPITULO

D O S

SISTEMAS DE INFORMACION

CONCEPTOS IMPORTANTES.

Para hablar de sistemas de información primero debemos definir tres conceptos importantes relacionados con éstos, los cuales son: datos, información y procesamiento de datos.

Datos.

Los datos son elementos aisladas y abstractos que empleados en operaciones de procesamiento, permiten obtener resultados relacionados con la identificación y evaluación de eventos y objetos.

Es decir, los datos son materia prima para producir información, son "entes" aisladas que pueden llegar a ser un todo capaz de formar un concepto mediante modelos de procesamiento de datos y de toma de decisiones.

Información.

Es el resultado obtenido de la organización y orientación apropiada de los datos. Es el procesamiento de los datos que nos puede suministrar un conocimiento. Generalmente, la información es requerida, como algo auxiliar, para tomar decisiones.

La información es una serie de eventos que llevan un mensaje y que al ser percibidas por un receptor aumenta los conocimientos de éste. Es decir, la función principal de la información es ampliar el conocimiento del receptor o reducir su incertidumbre.

CAPITULO II : SISTEMAS DE INFORMACION

Procesamiento de Datos.

Siempre se ha tenido la necesidad de información como un apoyo en el control del medio ambiente. Una de las exigencias fundamentales para producir información es el tamaño, crecimiento, especialización y complejidad de las organizaciones.

Los avances teóricos y tecnológicos (impresión, radio, televisión, computadora, etc.) son parte importante de la evolución del procesamiento de datos, los principales factores que interviene en el procesamiento de datos modernos son los modelos lógicos-matemáticos y la computación.

Ningún método de procesamiento de datos (manual, computarizado o híbrido) ha de garantizar que los datos serán procesados según las necesidades del sistema o del usuario.

Sistema de información.

"Un sistema de información es un conjunto sistemático y formal de componentes, capaz de realizar operaciones de procesamiento de datos" (2)

Tomemos en cuenta que "Sistemas de información "

- Es una expresión muy extensa que implica un sinnúmero de información destinada tanto al manejo de un sistema organizativo como a eventos externos.
- No depende de una computadora, ni debe basarse exclusivamente en el uso de computadoras.
- Es la clave del manejo efectivo de todo sistema.

Resumiendo: Un sistema de información es el subsistema de una organización que se ocupa de las actividades de procesamiento de datos y, además, proporciona información.

CAPITULO II : SISTEMAS DE INFORMACION

Un sistema de información contiene básicamente:

- * Operaciones del procesamiento de datos.
- * Métodos del procesamiento
- * Análisis del sistema
- * Técnicas de elaboración de modelos.

Y puede ser analizado según:

- * Sus objetivos.
- * Los métodos específicos aplicados.
- * Su estructura dentro de su medio ambiente o un sistema general que lo contenga.
- * La tecnología que lo hace funcional.
- * El conjunto de datos que constituyen su base.
- * El flujo de la información.

COMPUTACION Y SISTEMAS DE INFORMACION.

Para ampliar la efectividad del sistema de información y otorgar los medios para realizar actividades que pudieran parecer irrealizables (imposibles de efectuar), la computación es un instrumento básico. No todos los sistemas de información requieren ser computarizados, ni todos los sistemas de información computarizados mejoran un sistema, es decir, no necesariamente amplían su efectividad.

Los sistemas de información por computadora están integrados por: datos (información) y procesamientos (métodos); y son auxiliados por:

- * Equipo de cómputo (Hardware)
- * Programas de aplicación (Software)
 - Lenguajes de programación,
 - Editores, Compiladores, etc. -
- * Especialistas y usuarios (Humanware).

CAPITULO II : SISTEMAS DE INFORMACION

Los cuales se interrelacionan para resolver problemas, apoyar las operaciones de problemas. Procedimientos y datos específicos describen un sistema específico, es decir, comprenden una aplicación de sistemas de información.

Los sistemas de información por computadora, en general, tienen como tarea:

- Recibir las entradas del sistema.
- Procesar las entradas.
- Mantener los archivos de datos en relación con el objetivo principal del sistema.
- Producir información sobre lo procesado, emitir resultados del proceso y demás salidas.

* CICLO DE VIDA DE UN SISTEMA DE INFORMACION.

Cuando se desea generar un sistema, usualmente, primero se estudia la posibilidad de crearlo, la pregunta a contestar, para diseñadores y usuarios, sería: ¿Es conveniente hacerlo?, la siguiente pregunta a contestar es: ¿Qué se desea del sistema?, o bien, ¿Qué funciones realizará el sistema, qué objetivos deberá satisfacer?,

Una vez contestadas éstas, sigue lo más importante: ¿Cómo se hará el sistema, cómo deberá desarrollarse la solución del sistema?, ya definido el diseño, en seguida se inicia la programación, después las pruebas y por último la implantación del mismo. Estos pasos lo podemos observar en la figura 2.1

CAPITULO II : SISTEMAS DE INFORMACION

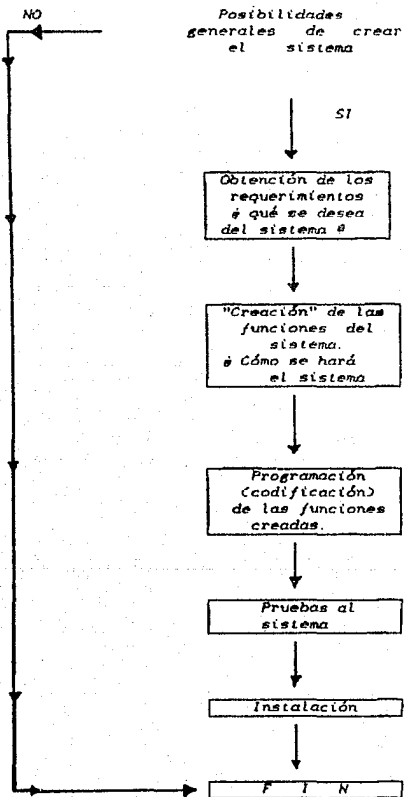


FIGURA 2.1

CAPITULO II : SISTEMAS DE INFORMACION

Para lograr desarrollar e implantar sistemas eficaces, eficientes y acordes con las metas primordiales, se deberá cumplir con los siguientes objetivos:

- Definir una metodología adecuada como soporte de las actividades de planeación, desarrollo y mantenimiento de los sistemas computarizados
- Determinar el ciclo de vida de los sistemas, para lograr su desarrollo y documentación en etapas muy bien definidas.
- Establecer los controles que se requieren en la revisión y aprobación de los documentos durante el ciclo de vida, en la liberación y utilización del sistema.

Tales objetivos permiten identificar las siguientes funciones básicas que se realizan dentro del ciclo de vida del sistema:

1. Viabilidad y planeación del proyecto.
2. Análisis y determinación de requerimientos.
3. Diseño del sistema.
4. Desarrollo e implementación.
5. Evaluación y mantenimiento.

Resumiendo: el ciclo de vida de un sistema es el conjunto de actividades necesarias para llevar a cabo el desarrollo y la implementación del sistema.

A continuación se describen brevemente las funciones básicas del ciclo de vida de los sistemas de información.

CAPITULO II : SISTEMAS DE INFORMACION

FUNCIONES BASICAS DEL CICLO DE VIDA DE LOS SISTEMAS DE INFORMACION

1. Viabilidad y Planeación del proyecto.

Es una fase de planeación y análisis sobre la posibilidad de realizar sistemas computacionales, reconociendo el objetivo del sistema, sus alcances y limitaciones, identificando sus requerimientos y sus relaciones con otros sistemas, de manera general.

Se deberá proporcionar información sobre la conveniencia del sistema en relación con sus costos y beneficios, también se deberán identificar y programar las actividades necesarias para llevar a cabo el sistema y se deberá estimar el costo y el tiempo de terminación y personal requerido.

2. Análisis y Determinación de requerimientos.

Aquí deberán identificarse detalladamente los requerimientos y limitaciones del sistema y así lograr una clara definición del mismo, obtendiéndose un diseño conceptual, que muestre tanto la estructura como el flujo de datos, procesos y salidas correspondientes. Establece qué objetivos deberá alcanzar el sistema.

En esta fase, se deberán analizar con gran detalle todos los aspectos que han de formar el sistema, justificandolos según los objetivos generales. Al final de esta etapa, el usuario deberá aprobar definitivamente el desarrollo del sistema.

Según James A. Senn [18], "... el punto clave del análisis de sistemas se consigue al adquirir un conocimiento detallado de todas las facetas importantes dentro del área que se investiga; se requiere consultar manuales y reportes, así como hacer una observación real de las actividades de trabajo y algunas veces se deberán recopilar formas y documentos para entender completamente al sistema".

CAPITULO II : SISTEMAS DE INFORMACION

3. Diseño del sistema.

Habr  de desarrollarse, en esta etapa, la estructura modular del sistema para definir las interfaces entre sus componentes incluyendo un dise o bien detallado del software con la especificaci n de cada programa que lo forma.

Se establece c mo cumplir  el sistema con los requerimientos identificados y describe los datos almacenados que se anexar n al sistema, adem s se determinan o identifican los archivos maestros y procedimientos.

En resumen, el dise o es una soluci n, es una traducci n de los requerimientos en formas que los satisfagan. Existen varias t cnicas para el dise o estructural del sistema, lo cual ser  mostrado y explicado posteriormente.

4. Desarrollo e instalaci n.

Se obtiene aqu , la codificaci n, programaci n y prueba de los programas ya identificados con anterioridad, los cuales se desarrollan con el uso de un lenguaje de programaci n previamente seleccionado (o definido). En cuanto a las pruebas, estas deber n ejecutarse tanto individuales (programa por programa, m dulo por m dulo) como globales (donde se integran todos los m dulos) para as  lograr su implementaci n.

Deber n de comentarse los programas e incluir los comentarios que especifiquen c mo y porqu  se utiliz  tal procedimiento y tal codificaci n. La documentaci n es un factor importante para las pruebas de los programas y el mantenimiento del sistema.

Al finalizar esta fase, se deber  tener un sistema en operaci n totalmente documentado y liberado.

5. Evaluaci n y mantenimiento.

Esta etapa, determina el nivel de satisfacci n de los usuarios, as  como las fallas y problemas existentes para poder llevar a cabo las modificaciones necesarias y lograr su operaci n eficiente.

CAPITULO II : SISTEMAS DE INFORMACION

Las evaluaciones se utilizan experimentalmente para asegurar que el sistema no falle: que cumpla con las especificaciones establecidas y que funcione como los usuarios esperan que lo haga.

C A P I T U L O

T R E S

MODULARIDAD Y

BASES DE DATOS

CAPITULO

T R E S

MODULARIDAD Y BASES DE DATOS

MODULARIDAD

"La complejidad de un sistema", como asegura J. P. van Gigch (7), "toma frecuentemente la forma de jerarquía o de sistema jerárquico". Los sistemas que son, o pueden ser, estructurados jerárquicamente tiene la ventaja de la modularización, la cual permite la organización de funciones o conjuntos de funciones que son usadas para separar el sistema en segmentos.

Un sistema modularizado es aquel que ha sido dividido en partes independientes que pueden ser instaladas o modificadas sin llegar a afectar a todo el sistema, lo cual es muy útil para el desarrollo de sistemas, en especial para el diseño.

A tales partes independientes se les llama *módulo* y este término es usado para hacer referencia a un conjunto de instrucciones contiguas de programa. Es decir, se usa el término *módulo* para un conjunto pequeño, pero integrado de programas que realizan alguna tarea específica y además posee interfaces bien definidas con otros módulos del sistema. Cada módulo deberá poseer un nombre (o etiqueta, generalmente único) por el cual otro módulo o partes de programa puedan hacer referencia a él.

El hecho de que un sistema esté dividido en segmentos no implica que sea eficiente, fácil de mantener o libre de errores. No se trata de hacer divisiones sólo por dividir sino para facilitar el trabajo y crear un sistema eficiente.

CAPÍTULO III : MODULARIDAD Y BASES DE DATOS

La forma más ineficiente de dividir un sistema es tomar como "regla de partición" un número determinado de líneas consecutivas de programa, pues se trabaja sin especificaciones explícitas de diseño, sin entender cómo manejar un objetivo o actividad.

Otra forma de dividir un sistema, quizá sólo un poco más eficiente que la anterior, es hacer un agrupamiento lógico. En cada módulo se manejan todos los tipos de funciones (actividades) similares, por ejemplo se puede crear un módulo que maneje todas las entradas de los datos, otro que maneje las salidas, otro que se encargue de todas las presentaciones en pantalla, etcétera.

En este tipo de agrupamiento podemos decir que tenemos actividades compartidas, ya que cada módulo tiene un "pedazo" de cada función lo cual implica que todas las instrucciones pueden ser ejecutadas al mismo tiempo, y esto es una ventaja sobre el agrupamiento anterior. Pero cuando hay que hacer modificaciones, éstas resultan difíciles pues el costo será compartido por cada actividad y el más insignificante cambio puede afectar a todo el sistema.

Una manera un poco mejor que las anteriores es generar agrupamientos que dependan de las estructuras de datos utilizados, es decir, cada módulo comparte un conjunto común de datos, variables o archivos. Por ejemplo, algún módulo podría tener las funciones de ordenamiento, y algunos otros las de modificaciones, consultas, copiado e impresión de datos de un archivo específico. Cuando hay que hacer modificaciones sólo se llega a afectar las actividades que dependen de tal módulo.

El mejor agrupamiento posible es cuando los módulos están determinados por la función que desempeñan, es decir, todos los elementos de un módulo tienen una sola meta o función, así es fácil determinar si el módulo está realizando su tarea correctamente.

CAPITULO III : MODULARIDAD Y BASES DE DATOS

Además permite que un módulo sea instalado y ejecutado sin depender de los demás lo cual ayuda a realizar pruebas más completas de los módulos y a que el mantenimiento sea más fácil, pues cada módulo es independiente de los otros. En los sistemas modularizados en forma apropiada, los módulos deben ser diseñados de tal manera que desarrollen una función específica.

Generalmente los módulos no deben relacionarse entre sí, es decir, sus conexiones deben ser limitadas y la interrelación entre sus datos debe ser mínima. Es preferible que cada módulo tenga su propio conjunto de variables. Casi todas las subrutinas de todos los tipos son módulos.

CONEXION ENTRE LOS MODULOS.

De acuerdo con Roger Pressman (16), "conexión es una medida del enlace entre los módulos en una estructura de programación", es decir es, una medida que se aplica a la relación que mantienen entre sí los módulos de un sistema (13).

El grado de conexión dependerá de la complejidad de los enlaces entre los módulos, de las llamadas entre ellos (es decir, si ésta es al módulo en total o a algún elemento de él) y de los parámetros (variables) recibidos o los resultados obtenidos por los módulos.

La comunicación entre los módulos deberá realizarse a través de el mínimo número posible de parámetros procurando entrar a los módulos únicamente por la parte inicial, evitando así módulos con entradas múltiples.

Los módulos deberán ser diseñados de tal forma que entre ellos se transmita solamente la información necesaria, la que pueda ser considerada como indispensable para el funcionamiento de los módulos que se están interrelacionando, y evitar que algún módulo sea responsable del funcionamiento interno de otro, previendo así la independencia intermodular.

CAPÍTULO III : MODULARIDAD Y BASES DE DATOS

James Senn (18) afirma que: " El uso de diseños descendentes y modulares producirá un sistema bien estructurado. Este enfoque da como resultado módulos sencillos en donde las conexiones con otros módulos son aparentes. Estas características facilitan el mantenimiento y al mismo tiempo aseguran que es más fácil desarrollarlo cuando se requiera"

Por otra parte, para las conexiones entre los módulos, se deberán respetar los niveles jerárquicos del sistema y las rutas que éstos determinan en la estructura general del sistema.

BASES DE DATOS

En algunas aplicaciones la flexibilidad del almacenamiento y consulta de datos es esencial. Algunas veces no es suficiente utilizar estructuras de archivos como las secuenciales, aleatorias o las secuenciales indexadas, por ello se recurre al manejo de bases de datos.

Una base de datos es un conjunto de archivos, cuyos datos están almacenados, organizados y relacionados entre sí.

Utilizar estructuras "comunes" de archivos (como las secuenciales o aleatorias) implica crear diferentes conjuntos de información, es decir, diferentes tipos de archivos, lo cual generalmente ocasiona redundancia de datos, además, no se tiene un fácil manejo de estos archivos, por ejemplo si se necesita agregar o cambiar nuevos atributos a un archivo de datos se deberá crear un nuevo archivo para esa nueva estructura, esto implicaría, quizá añadir nuevos datos a archivos ya existentes y por tanto se deberán revisar todos los archivos y programas existentes, esta "pequeña operación", finalmente además de ser laboriosa resulta ser costosa y muy propensa a errores.

CAPITULO III : MODULARIDAD Y BASES DE DATOS

En cambio, la estructura de una base de datos es diferente y su manejo también. En las bases de datos pueden extraerse diferentes conjuntos de información de un fondo común de datos, que puede ser único, además modificar o agregar nuevos atributos, generalmente, no afecta ni a las demás bases de datos, ni al sistema total, es decir, se tiene una independencia de datos.

Un sistema para manejo de bases de datos proporciona una facilidad de manipulación de los mismos, sin importar las características internas (definición lógica y almacenamiento físico).

ESTRUCTURA DE UNA BASE DE DATOS

Una base de datos, como se mencionó anteriormente, es un conjunto de archivos relacionados entre sí, en el cual cada archivo consta de una colección de registros de un mismo tipo y a su vez cada registro es una colección de campos afines. La figura 3.1 muestra la estructura de una Base de Datos.

Básicamente la estructura de una base de datos está compuesta por campos (o elementos dato), registros y archivos.

Donde:

Campo es un conjunto de caracteres con atributos, los cuales definen la clase y longitud de los datos. El campo es la "unidad elemental" de una base de datos, es la parte más pequeña de la estructura.

Registro es una colección de campos relacionados entre sí considerados como una unidad.

Archivo es una colección de registros del mismo tipo de datos.

CAPÍTULO III : MODULARIDAD Y BASES DE DATOS

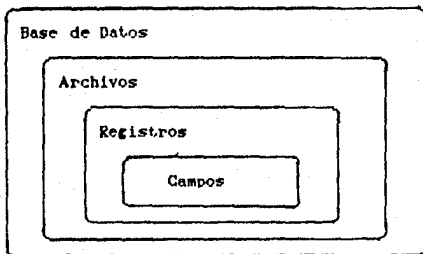


Figura 3.1
Estructura de una base de datos

En una base de datos los registros de un archivo contienen valores de datos relacionados con alguna entidad del mundo real.

Para describir una base de datos es necesario construir un modelo que represente su estructura y permita una total manipulación.

Una vez que se han recolectado todas las definiciones para la Base de Datos, es decir cuando se han especificado los campos, registros y archivos que habrán de constituirlos, se tiene, entonces, un Modelo Estructural o un Modelo de la Base de Datos. En el modelo estructural deberán describirse los datos y las relaciones o conexiones existentes entre los datos.

Una base de datos que sea usada para dos objetivos diferentes, puede tener varios modelos diferentes, según la visión de diversos usuarios. Una sola visión de una base de datos puede ser descrita mediante un modelo. Un modelo de visión representa un pequeño subconjunto de la realidad, el cual será el apropiado para alguna aplicación del contenido de la base de datos.

CAPITULO III : MODULARIDAD Y BASES DE DATOS

Un modelo de visión habrá de construirse a partir de los campos y sus vínculos.

La expresión básica de un vínculo es un vínculo binario, el cual conecta al Objeto (Clave) con el Valor del Atributo.

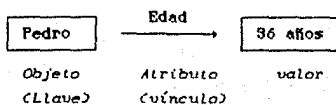


Figura 9.2
Vínculo Binario

Cuando los datos se recolectan y son organizados para su proceso, entonces, surgen múltiples vínculos binarios, los cuales conectan al Objeto (Clave) con varios Atributos.

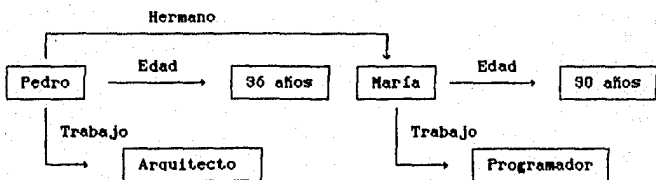


Figura 9.3
Vínculos Binarios Múltiples

CAPÍTULO III : MODULARIDAD Y BASES DE DATOS

Reorganizando los múltiples vínculos binarios, se generan *n-adas* (registros). Una *n-ada* representa múltiples atributos del objeto.

Nombre	Edad	Hermano	Edad_H	Trabajo
Pedro	36	María	30	Arquitecto

Nombre	Edad	Hermano	Edad_H	Trabajo
María	30	Pedro	36	Programador

Figura 3.4
N - adas

Ahora bien, un conjunto de *n-adas* semejantes forma una *Relación*, cada *n-ada* en una *relación* representa algún objetivo diferente y expresa vínculos semejantes entre sus atributos. Cada vínculo en un registro (*n-ada*) se define mediante el nombre que sirve de etiqueta a la columna de la *n-ada*, o sea el atributo.

PERSONA

Nombre	Edad	Hermano	Edad_H	Trabajo
María	30	Pedro	36	Programador
Pedro	36	María	30	Arquitecto
Juan	29	- - -	---	Ing Industrial
Hugo	28	Manuel	26	Abogado
Manuel	26	Hugo	28	Ingeniero E

Figura 3.5
Relación

CAPITULO III : MODULARIDAD Y BASES DE DATOS

Cada r-ada existe para describir un objeto específico, se puede tomar una o varias columnas de atributos para dar nombre o definir al objetivo; estos atributos forman una parte rectora, generalmente llamada *clave*. Los atributos restantes son la parte dependiente.

Resumiendo se tiene que en una base de datos se puede:

- Compartir la información por más de un módulo.
- Interrelacionar las bases de datos.
- Evitar más fácilmente la redundancia de datos.
- Disponer de los datos tanto de una base de datos como de los de una función específica que puede involucrar varias bases de datos.

C A P I T U L O

C U A T R O

DISEÑO ESTRUCTURADO

CAPITULO
C U A T R O
D I S E Ñ O E S T R U C T U R A D O

INTRODUCCION

Diseñar significa planear o trazar la forma y el método de una solución. El diseño es una etapa muy importante del ciclo de vida del desarrollo de sistemas pues en él se determinan las principales características del sistema final, se establecen los límites inferiores y superiores en ejecución, además, se generan los algoritmos que formarán parte del sistema y se puede determinar cuál será el costo final. El diseño es, finalmente, una presentación detallada del análisis del sistema.

El diseño, casi siempre, se caracteriza por un gran número de decisiones técnicas, es decir, es de naturaleza técnica; nos indica cómo realizar los requerimientos descritos en la fase del análisis, se pasa del qué se hará al cómo hacerlo.

Durante el proceso de diseño se deberán plantear soluciones alternativas y finalmente determinar cuál será la óptima, o sea que esta etapa requiere de mucha creatividad, por lo general, los diseñadores continuamente se preguntan: ¿Cómo habrá de hacerse ...?, ¿Qué ocurriría si ...?, ¿Por qué no usar ...? y muchas otras cuestiones similares que ayudarán a elaborar y concluir exitosamente el sistema.

† CAPITULO IV : DISEÑO ESTRUCTURADO

Diseñar un sistema implica conocer más de cerca a la organización para la cual se realizará el sistema; saber qué necesidades de información tiene, cómo la procesan y saber con qué equipo de trabajo se cuenta.

Al respecto J. Burch & F. Strater [2] afirman que:

"... para llevar adelante la fase del diseño es preciso conocer :

- 1) Los recursos con que cuenta la administración.
- 2) Las necesidades de información de los usuarios.
- 3) Las necesidades del sistema.
- 4) Los métodos de procesamiento de datos.
- 5) Las operaciones que deben efectuarse con los datos.
- 6) Los instrumentos del diseño.

En el proceso de diseño, el analista pone en práctica dos elementos más: el razonamiento y la creatividad. "

Generalmente, esta fase, ha sido menospreciada (y hasta ignorada o relegada) por muchas organizaciones sin tener en cuenta que de ella depende el buen funcionamiento de un sistema, es común que se conformen con un diseño, alguna solución cualquiera y no busquen (o no sean capaces de encontrar) el mejor diseño el cual lleve a la mejor solución, que implica la obtención de un muy buen sistema.

Para obtener buenos diseños y desechar, con relativa facilidad, los "malos" es necesaria una metodología como el Diseño Estructurado que es una colección de principios o normas que ayudan a distinguir entre un buen diseño y un mal diseño, es además, un conjunto de técnicas, estrategias y heurísticas que, regularmente, conducen a un buen diseño.

CAPITULO IV : DISEÑO ESTRUCTURADO

El Diseño Estructurado es una metodología que consolida, formaliza y hace visibles las actividades y decisiones del diseño, las cuales ocurren inevitable e ineludiblemente durante la elaboración de todos los proyectos de desarrollo de sistemas. NO es una etapa más del ciclo de vida, es tan sólo el proceso que decide cuáles componentes deberán interconectarse y cuáles deberán resolver un problema específico.

Algunas técnicas del diseño estructurado son:

- 1. Cajas Negras.*
- 2. Diagrama jerárquico funcional.*
- 3. Diagramas de flujo de datos.*
- 4. Diagramas de Warnier/Orr.*
- 5. Diagramas H. I. P. O.*

En este capítulo se trabajará con un problema específico, cuyo diseño será desarrollado utilizando las técnicas antes mencionadas, mostrando así que un problema no puede ser tan fácilmente representado y desarrollado con todas las técnicas.

CAPITULO IV : DISEÑO ESTRUCTURADO

Problema: Control de Alquiler de Trajes

Se desea un sistema que sea capaz de llevar el control de una tienda de alquiler de trajes: el control de los clientes y del almacén (inventario general del almacén acerca de los trajes: existencias, modelos, etc.).

En el control de Clientes se deberán obtener las altas (captura de la información, tanto de los trajes alquilados, como de los clientes), consultas (generales, por fecha de alquiler y de entrega), modificaciones y bajas (entrega del traje).

En el control del almacén se desea tener las altas, bajas, modificaciones, consultas y reportes. Estas tres últimas por claves, nombres, tallas o color. Se deberá tomar en cuenta que se cobrará una multa por cada día de retraso en la entrega del traje.

Dentro del sistema general usaremos una base de datos, la cual está formada por dos archivos: uno para el control de los clientes (Clientes) y otro para el Almacén (Trajes).

En seguida se describirán y ejemplificarán las técnicas del diseño estructurado, antes mencionadas.

CAPITULO IV : DISEÑO ESTRUCTURADO

1. CAJAS NEGRAS

Es un sistema que puede ser cien por ciento explotado sin saber qué contiene en su interior. En la vida diaria usamos (y hasta explotamos) diferentes tipos de cajas negras como lo son: televisores, radios, automóviles y hasta computadoras; de las cuales, generalmente, desconocemos "detalladamente" cómo es que funcionan, qué hay en su interior. Los sistemas de información por computadora son cajas negras para sus usuarios.

En una caja negra, frecuentemente, sólo se conocen las entradas, las salidas y cuál es la transformación que se hace a las entradas, pero no se conoce su contenido, es decir, la manera en que se hace la transformación. La figura 4.01 muestra la representación general de una caja negra: se tienen las entradas, el proceso (caja negra) y finalmente las salidas.



FIGURA 4.01

Reglas generales para crear una Caja Negra.

- | | |
|---|------------------------|
| <ul style="list-style-type: none">* Identificar Salidas.* Identificar Entradas.* Definir Funciones Principales | } Análisis del Sistema |
| <ul style="list-style-type: none">* Establecer la jerarquía con un diagrama.* Construir diversas alternativas de solución. | } Síntesis del Sistema |

† CAPITULO IV : DISEÑO ESTRUCTURADO

Las cajas negras son especialmente útiles en la fase del diseño, ya que pueden ser usadas sin preocuparse por su contenido; la caja negra usada debe ser diseñada y aunque es un concepto muy útil no es suficiente para permitir un buen desarrollo.

Los procedimientos y estructuras, de una caja negra, deberán determinarse, pero únicamente los necesarios, lo cual indica que toda componente estará disponible, además mantiene al diseñador trabajando sobre el resultado principal sin distraerlo por detalles innecesarios.

Esta técnica es adecuada como primer intento, en el desarrollo de un sistema, lo cual puede verse como una ventaja.

Ejemplos de Cajas Negras:

1) Desarrollando el sistema de control de alquiler de trajes se obtiene el diagrama 4.02, se tiene la estructura general del sistema: Entradas, Proceso (Sistema Control de alquiler de trajes) y Salidas. En este ejemplo es claro que la técnica no ayuda a resolver, al menos, a ilustrar mejor, este tipo de sistema, pues es necesario saber qué contiene el proceso en su interior, cómo funciona y que datos requiere para funcionar. Si se intenta desglosar más el diagrama 4.02, se llegará a la necesidad de conocer cómo funciona el sistema (cuál es la transformación) y eso contradice la definición de una Caja Negra.

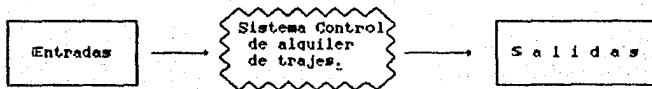


FIGURA 4.02

CAPITULO IV : DISEÑO ESTRUCTURADO

2) Al usar el sistema TeX para procesar algún texto, es necesario pasar por varias etapas, las cuales pueden verse como cajas negras, primero se debe tener listo el texto a procesar el cual con ayuda de un editor será guardado en un disco, ahora bien ese archivo estará compuesto del texto más las instrucciones de TeX, tal archivo deberá ser procesado por el compilador TeX, si el archivo está correcto, entonces TeX entregará un archivo listo para imprimir (que pasa por un manejador de la impresora) o para verse en pantalla (que pasa por un manejador de la pantalla). En la figura 4.03 se muestra la representación gráfica de este ejemplo.

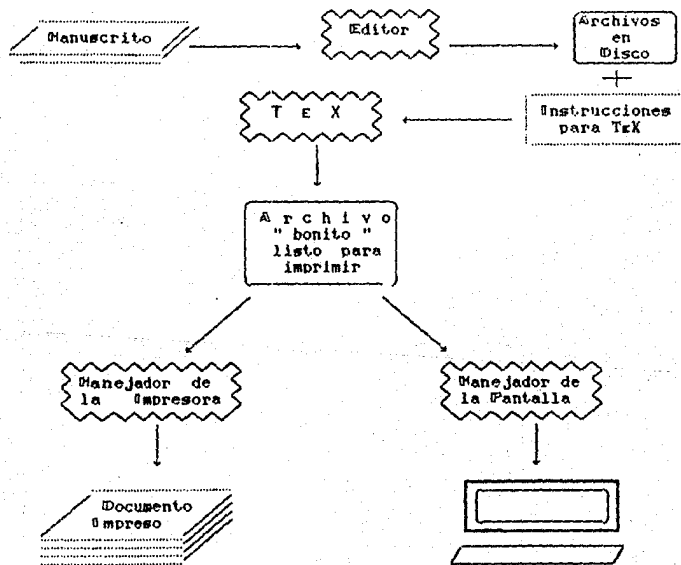


FIGURA 4.03

CAPITULO IV : DISEÑO ESTRUCTURADO

2. DIAGRAMA JERARQUICO FUNCIONAL

Tiene como objetivo proporcionar una estructuración del sistema que simplifique el entendimiento de sus funciones, su construcción está fundamentada en el principio del diseño de arriba hacia abajo (top-down).

El diagrama jerárquico funcional da una representación esquemática del sistema y una relación jerárquica de sus elementos (Sistema, subsistemas, módulos, submódulos, procesos, etc.). En la figura 4.04 la representación nos indica que el sistema consta de tres módulos (A, B, C); el módulo A tiene 2 submódulos (A1, A2), el módulo A1 está compuesto por dos procesos (A11, A12). Y así sucesivamente podemos ir identificando la estructura.

Esta estructura de diseño puede ser utilizada para describir sistemas en operación, tanto computarizados como manuales, así como para iniciar el diseño conceptual del nuevo sistema. Con esta técnica es muy fácil mostrar cualquier tipo de estructura (u organización) en que se presente alguna jerarquía, como lo son las organizaciones empresariales, gubernamentales, etc.

Ejemplos:

1) En la figura 4.05 se muestra una estructura, muy general, de la organización en una editorial. La dirección general es la parte más importante de la editorial y tiene a su cargo cuatro sub-direcciones las de Publicidad, Redacción, Fabricación y Administración, que a su vez están sub-divididas en otros departamentos, por ejemplo: Fabricación cuenta con tres departamentos Relaciones Exteriores (dividida en: Preparadores, Compras e Impresiones), Laboratorio y las Oficinas Técnicas (divididas en: Correctores, Bocetistas y Fotógrafos).

2) El diseño del problema "Control de alquiler de trajes", usando esta técnica, se muestra en la figura 4.06, donde claramente se ve son usados principalmente dos módulos: Control de Clientes y Control de Almacén como se había especificado en el breve análisis.

CAPITULO IV : DISEÑO ESTRUCTURADO

2. DIAGRAMA JERARQUICO FUNCIONAL

Tiene como objetivo proporcionar una estructuración del sistema que simplifique el entendimiento de sus funciones, su construcción está fundamentada en el principio del diseño de arriba hacia abajo (top-down).

El diagrama jerárquico funcional da una representación esquemática del sistema y una relación jerárquica de sus elementos (Sistema, subsistemas, módulos, submódulos, procesos, etc.). En la figura 4.04 la representación nos indica que el sistema consta de tres módulos (A, B, C); el módulo A tiene 2 submódulos (A1, A2), el módulo A1 está compuesto por dos procesos (A11, A12). Y así sucesivamente podemos ir identificando la estructura.

Esta estructura de diseño puede ser utilizada para describir sistemas en operación, tanto computarizados como manuales, así como para iniciar el diseño conceptual del nuevo sistema. Con esta técnica es muy fácil mostrar cualquier tipo de estructura (u organización) en que se presente alguna jerarquía, como lo son las organizaciones empresariales, gubernamentales, etc.

Ejemplos:

1) En la figura 4.05 se muestra una estructura, muy general, de la organización en una editorial. La dirección general es la parte más importante de la editorial y tiene a su cargo cuatro sub-direcciones las de Publicidad, Redacción, Fabricación y Administración, que a su vez están sub-divididas en otros departamentos, por ejemplo: Fabricación cuenta con tres departamentos Relaciones Exteriores (dividida en: Preparadores, Compras e Impresiones), Laboratorio y las Oficinas Técnicas (divididas en: Correctores, Bocetistas y Fotógrafos).

2) El diseño del problema "Control de alquiler de trajes", usando esta técnica, se muestra en la figura 4.06, donde claramente se ve son usados principalmente dos módulos: Control de Clientes y Control de Almacén como se había especificado en el breve análisis.

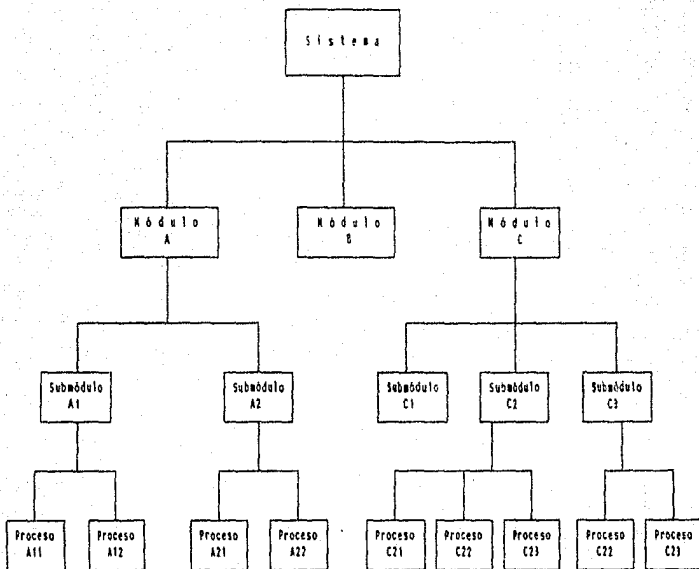


FIGURA 4.04

RELACION JERARQUICA DE LOS
ELEMENTOS DE UN SISTEMA

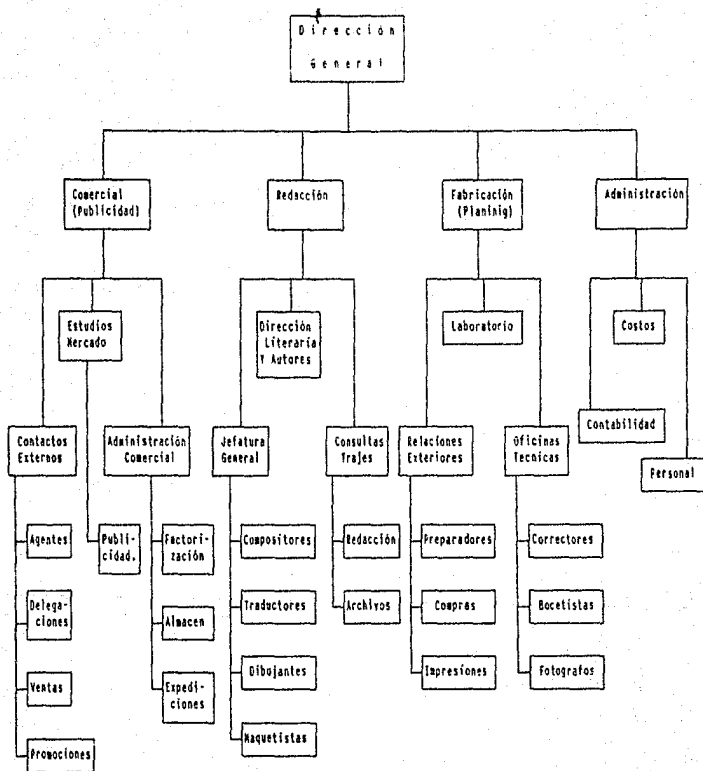


FIGURA 4.05

DIAGRAMA BASICO DE UNA EDITORIAL

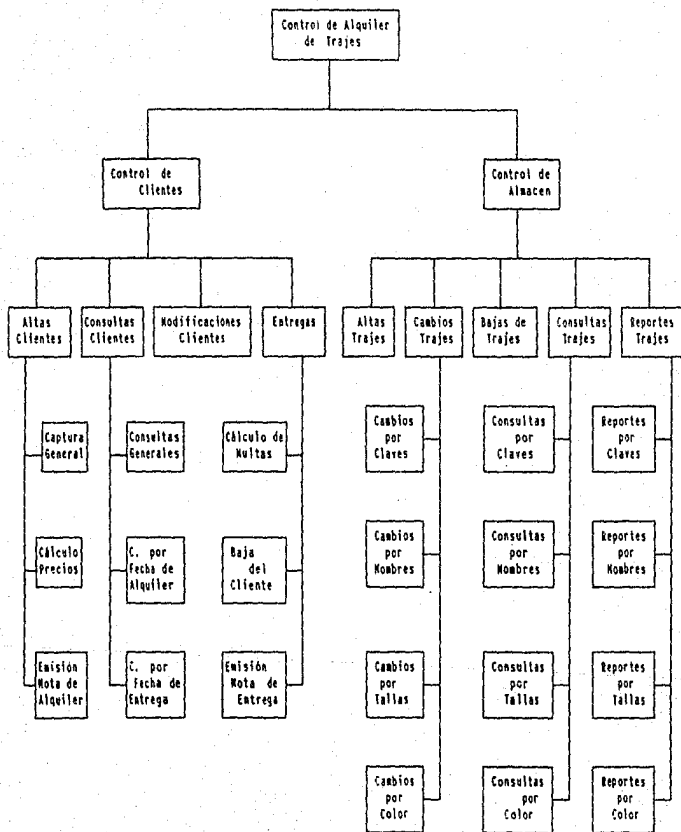


FIGURA 4.06

DIAGRAMA JERARQUICO FUNCIONAL
 SISTEMA CONTROL DE ALQUILER DE TRAJES

† CAPITULO IV : DISEÑO ESTRUCTURADO

En este tipo de estructuras es muy fácil localizar una función específica y comprender el sistema a cierto nivel.

Por ejemplo, en la figura 4.07 se muestra "el camino a recorrer" para poder realizar una consulta de trajes por talla de los trajes, estando en el sistema Control de Alquiler de Trajes se deberá pasar al módulo de Control de Almacén donde habrá de escogerse el proceso de Consultas de Trajes y finalmente seleccionar la opción de Consultas por Tallas.

En la figura 4.08 se indica el tercer nivel jerárquico del sistema, el nivel donde se puede elegir un proceso especial que se desee realizar; el sistema Control de Alquiler de Trajes (primer nivel) tiene dos módulos principales: Control de Clientes y Control de Almacén (segundo nivel) los cuales tienen sub-módulos (tercer nivel) que son las opciones a seleccionar para poder realizar un proceso específico del sistema.

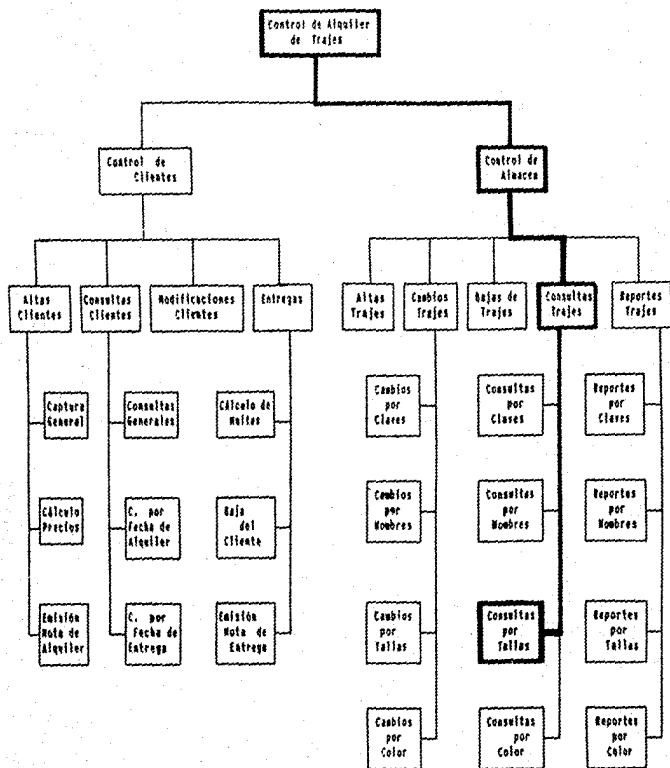


FIGURA 4.87

SISTEMA CONTROL DE ALQUILER DE TRAJES

LOCALIZACIÓN DE UNA FUNCIÓN ESPECÍFICA DEL SISTEMA

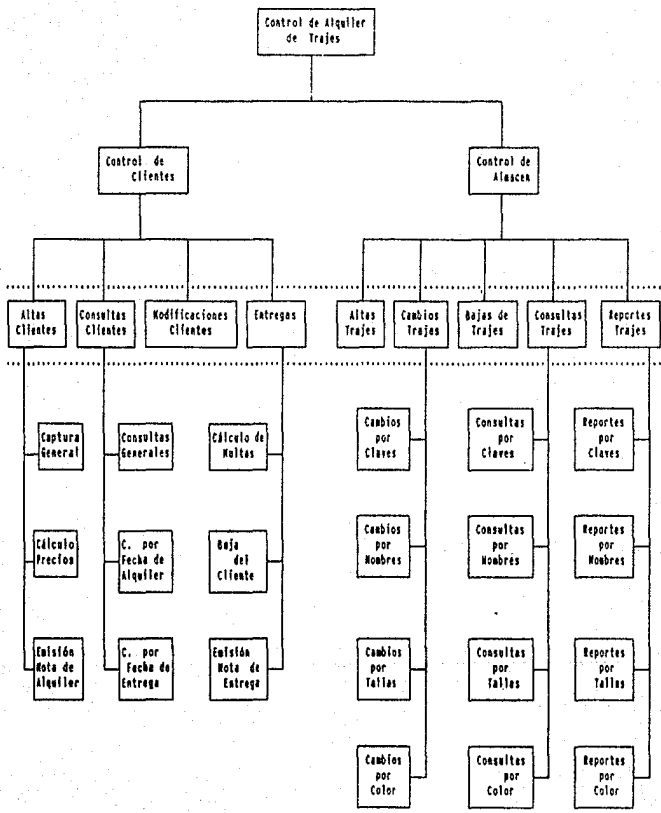


FIGURA 1.00
 SISTEMA CONTROL DE ALQUILER DE TRAJES
 LOCALIZACION DE UN NIVEL DEL SISTEMA

CAPITULO IV : DISEÑO ESTRUCTURADO

3. DIAGRAMA DE FLUJO DE DATOS

Su objetivo principal es mostrar gráficamente el flujo de los datos de un sistema, identificando sus entradas, archivos y salidas, pasando por todos los procesos de transformación.

Sus principales características son:

- Puede representar el flujo de datos de cualquier tipo de sistema (manual, automatizado o híbrido). Puede modelar un sistema sin importar su grado de automatización.
- Enfatiza principalmente el flujo de datos más que el del control.
- Todos los procesos de transformación son representados con sus entradas y salidas.
- Cada transformación se puede ir refinando hasta el último detalle, es decir, hasta que sea suficientemente clara.
- Ayuda a ilustrar las componentes esenciales de un proceso y la forma en que interactúan (o se interrelacionan).
- Cuenta con una notación simbólica especial, la cual es presentada a continuación:

Símbolo:



Representa:

Flujo de datos

Fuente o destino.

† CAPITULO IV : DISEÑO ESTRUCTURADO



Archivo de Datos.



Proceso de
Transformación.

En el diagrama, cada componente de flujo de datos deberá estar etiquetado con un nombre descriptivo y con un número que será utilizado como identificación.

Antes de crear un Diagrama de Flujo de Datos se deberán identificar: las entradas y salidas del sistema, así como los procesos. Además se deberá tener en cuenta la depuración del diagrama.

Reglas Generales Para Crear Un Diagrama de Flujo de Datos.

1. El primer diagrama debe ser el modelo fundamental del sistema.
2. Los archivos de entrada y salida primarios deben ser seleccionados con sumo cuidado.
3. Todas las flechas y burbujas deben ser etiquetadas apropiadamente.
4. Se debe mantener la continuidad en la información.
5. Los detalles del sistema se adquieren en un nivel a la vez. Sólo se debe definir una burbuja a la vez.

CAPITULO IV : DISEÑO ESTRUCTURADO

En el diagrama final se deberá minimizar el número de cruces de líneas de flujo, duplicando entidades externas o archivos, si es necesario, además, los nombres que están especificados en el Diagrama de Flujo de Datos deben representar a los datos que se mueven en el flujo de datos real.

Los diagramas de flujos de datos son también conocidos como: Gráficas de flujo de datos o Gráficas del programa

En la figura 4.09 se muestra en forma general el refinamiento de un sistema. Tenemos al sistema F con sus módulos de entrada y salida, en el primer refinamiento se despliegan las funciones de F (F₁, F₂, F₃ y F₄) y sus flujos de datos.

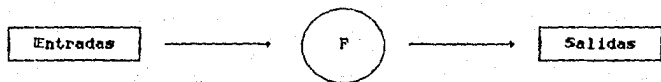
Si se recibe el flujo de datos y, de las Entradas, se realizará el proceso F₁ produciendo el flujo de datos x que será procesado por F₂ emitiendo sus resultados por el flujo de datos z los cuales deberán pasar por el proceso F₄ y el que, para finalizar, mandará los datos obtenidos a las Salidas. Si se recibe el flujo de datos w se realizará F₃ generando el flujo de datos v los cuales serán procesados por F₂ que continuará el proceso.

En el segundo refinamiento se detalla el proceso F₃ al igual que su flujo de datos.

Ejemplos:

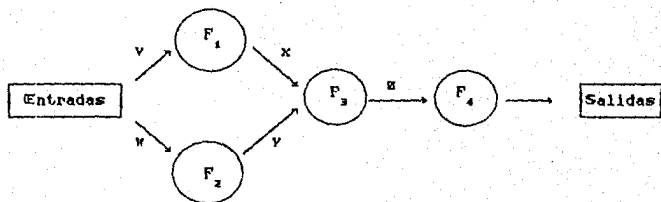
1) En la figura 4.10 se muestra un ejemplo de un sistema con sólo dos procesos generales y cinco flujos de datos diferentes. Si del Origen (módulo de entrada) se recibe el flujo de datos 1 entonces se hará el Proceso 1, se guardarán las actualizaciones de los datos en el archivo de Almacenamiento de datos y finalmente los resultados obtenidos serán mandados al Destino (módulo de salida) por medio del flujo de datos 4. Si del Origen se recibe el flujo de datos 2, entonces se realizará el Proceso 2 cuyas actualizaciones serán enviadas en el flujo de datos 3 al Proceso 1, el cual llevará a cabo sus objetivos.

† CAPITULO IV : DISEÑO ESTRUCTURADO



Detalles del
Proceso

Primer Refinamiento



Más detalles
del Proceso

Segundo Refinamiento

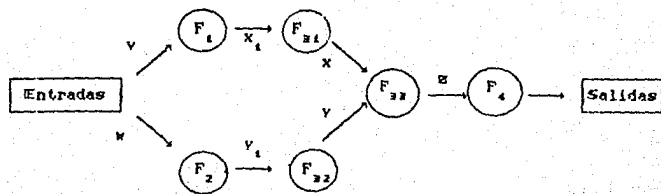


FIGURA 4.09

CAPITULO IV : DISEÑO ESTRUCTURADO

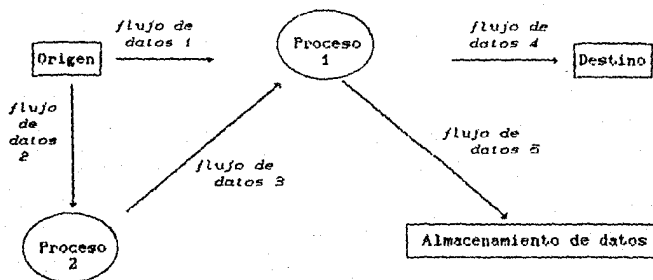


FIGURA 4.10

2) El sistema para control de alquiler de trajes está mostrado en las figuras 4.11-A y 4.11-B, utilizando estática.

El primer diagrama, figura 4.11-A, muestra la estructura muy general del sistema: Las Entradas al proceso (Control Alquiler de trajes) y las Salidas. El siguiente diagrama, muestra el flujo de datos de los módulos principales (clientes y almacén), si se recibe el flujo de datos para clientes, entonces será procesado el módulo Control de Clientes y si se reciben los datos para el almacén, entonces será ejecutado el módulo Control de Almacén.

Finalmente, en el tercer diagrama, figura 4.11-C, se muestra el flujo de datos necesario para elegir el proceso a realizar en el sistema. La figura 4.11-C resulta "muy difícil" de analizar, pues se ha llegado a un nivel de detalle donde se tienen múltiples procesos y accesos a la base de datos. Para evitar tener tantos cruces entre las flechas de flujos de datos, se separará el diagrama 4.11-C en dos. Uno que muestre cuales son los accesos a los archivos de datos (fig. 4.11-C1) y otro que señale los flujos de datos entre los procesos (fig. 4.11-C2)

CAPITULO IV : DISEÑO ESTRUCTURADO

Y para que el diagrama quede "mas legible" se separaran los diagramas. Las figuras 4.12-A ilustran detalladamente el proceso de control de clientes y las figuras 4.12-B el proceso de control de almacén.

Primer Diagrama: (Módulo general del sistema)



Segundo Diagrama:

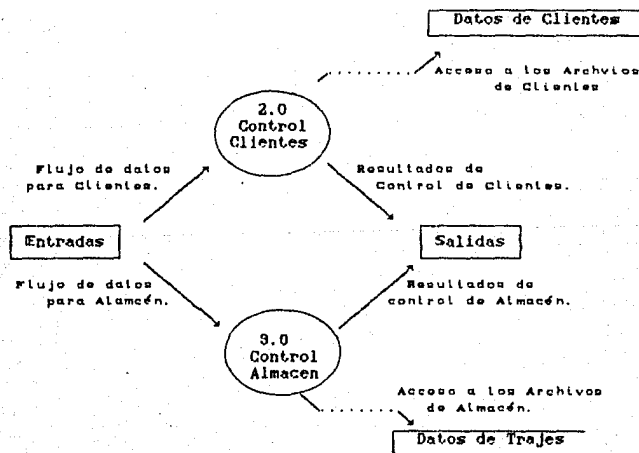


Figura 4.11 - A

CAPITULO IV : DISEÑO ESTRUCTURADO

Tercer Diagrama:

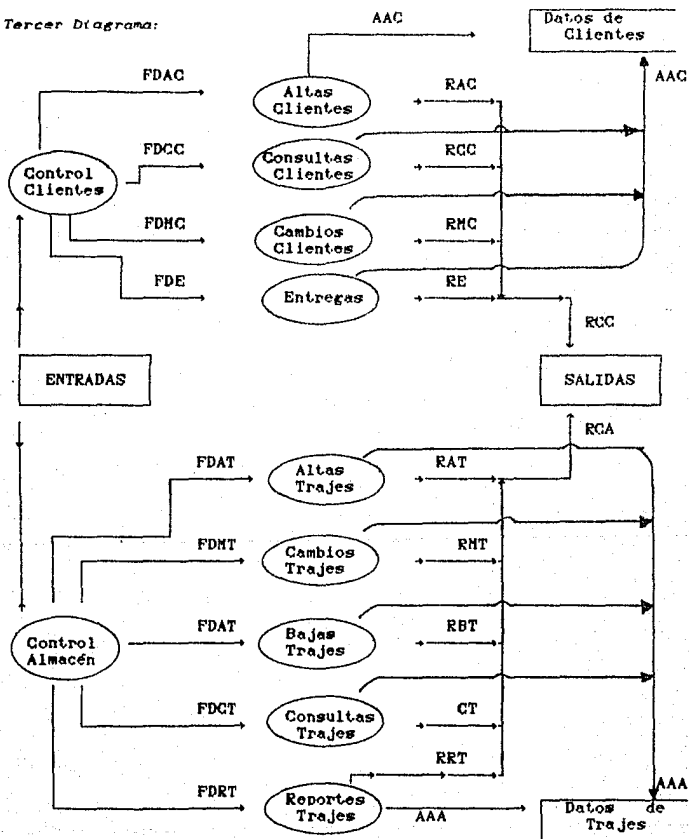


FIGURA 4.11 - B

CAPITULO IV : DISEÑO ESTRUCTURADO

Donde:

FDAC : Flujo de Datos para Altas de Clientes.
FDCC : Flujo de Datos para Consultas de Clientes.
FDMC : Flujo de Datos para Modificaciones de Clientes.
FDE : Flujo de Datos para Entregas

FDAT : Flujo de Datos para Altas de Trajes.
FDMT : Flujo de Datos para Modificaciones de Trajes.
FDBT : Flujo de Datos para Bajas de Trajes.
FDCT : Flujo de Datos para Cambios de Trajes.
FDRT : Flujo de Datos para Reportes de Trajes.

RAC : Resultados de Altas de Clientes.
RCC : Resultados de Consultas de Clientes.
RMC : Resultados de Modificaciones de Clientes.
RE : Resultados de Entregas.

RAT : Resultados de Altas de Trajes.
RMT : Resultados de Modificaciones de Trajes.
RBT : Resultados de Bajas de Trajes.
RCT : Resultados de Cambios de Trajes.
RRT : Resultados de Reportes de Trajes.

AAA : Acceso a los Archivos de Almacén.
AAC : Acceso a los Archivos de Clientes.

RA : Resultados de Almacén.
RC : Resultados de Clientes.

CAPITULO IV : DISEÑO ESTRUCTURADO

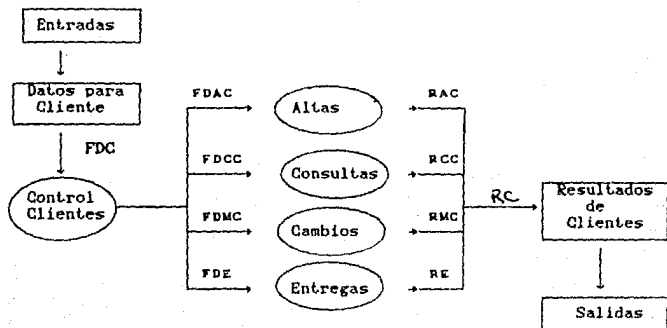


Figura 4.12-A1

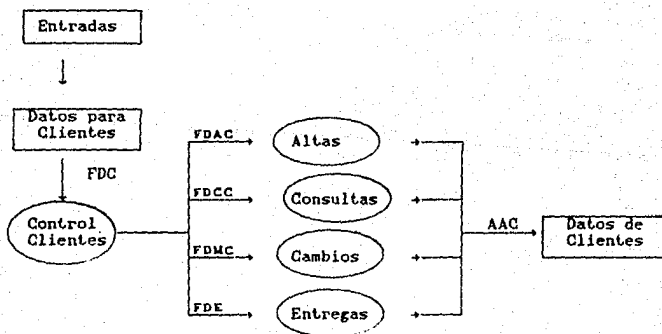


Figura 4.12-A2

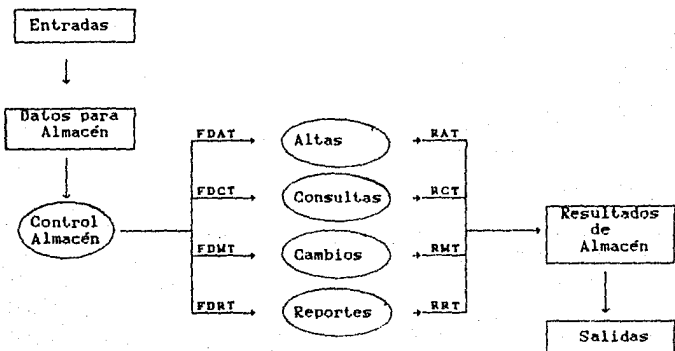


Figura 4.12-B1

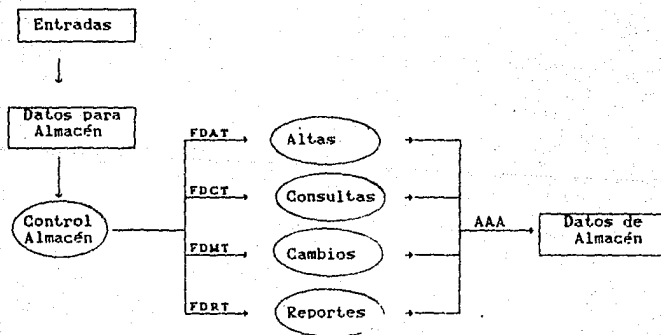


Figura 4.12-B2

CAPITULO IV : DISEÑO ESTRUCTURADO

4. DIAGRAMAS DE WARNIER / ORR

Esta técnica ayuda al diseño de las estructuras de un programa, inicia identificando la salida y los resultados del proceso, después trabaja para establecer los pasos y el acoplamiento que son necesarios para producirlos.

Los métodos gráficos y simples de esta técnica muestran de forma muy clara los niveles del sistema y el "movimiento" de datos hecho de un nivel al otro.

Este método utiliza un análisis orientado a la salida, primero se define la salida deseada o los resultados del proceso, después, en el siguiente nivel se definen los pasos necesarios a fin de producir tal salida; cada paso a su vez se define más adelante. Corchetes adicionales agrupan los procesos requeridos para generar el resultado en el siguiente nivel.

Algunas características de los diagramas de Warnier/Orr son:

- Muestra los procesos y las secuencias en que éstas se desempeñan.*
- Cada proceso se define en forma jerárquica, de izquierda a derecha y de arriba hacia abajo.*
- Cada nivel del proceso se muestra en corchetes que agrupan a sus componentes.*
- Muestran fácilmente las iteraciones e intercalaciones.*
- Cualquier sistema puede representarse.*

† CAPITULO IV : DISEÑO ESTRUCTURADO

Un diagrama completo de Warnier/Orr engloba tanto agrupamientos de procesos como requerimientos de datos. Cuando se termina la definición se documenta una estructura de datos para cada proceso, facilitando así la codificación del sistema.

Esta técnica cuenta con la siguiente notación :

- Las iteraciones son marcadas como: "(1,x)"
que indica que se hará desde 1 hasta x veces.
- "(0,1)" indica que puede suceder o no.
- "ø" representa alternativa.
- "ø" indica que no existe condición.

Ejemplos:

1) Representemos, con esta técnica, los días de clases en la semana. Durante la semana se tienen clases con los profesores titulares, con los profesores ayudantes y hay días de descanso.

Generalmente las clases con los titulares son los días: lunes, miércoles y viernes; con los ayudantes: martes y jueves; y los días de descanso son: sábados y domingo.

El diagrama queda;

Días de clases en la semana	{	Días de clase con el Titular	{ Lunes Miércoles Viernes
		Días de clase con el Ayudante	{ Martes Jueves
		Días de descanso	{ Sábados Domingos

CAPITULO IV : DISEÑO ESTRUCTURADO

2) Representemos, ahora, la estructura general del sistema político mexicano. El sistema político mexicano está dividido, principalmente, en Gobierno Federal y Gobierno Local, ambos cuentan con los Poderes: Legislativo, Ejecutivo y Judicial.

El Poder Legislativo Federal está compuesto por las Cámaras de Diputados y Senadores; el Poder Ejecutivo Federal está representado por el Presidente de la República, las Secretarías de Estado y los Departamentos de Estado; El Poder Judicial Federal está integrado por la Suprema Corte de Justicia, los Tribunales de Circuito y los Juzgados de Distrito.

El Poder Legislativo Local está formado por el Congreso Local; el Poder Ejecutivo Local esta representado por el gobernador del Estado y el Poder Judicial Local, está compuesto por la tribuna Superior de Justicia y los Juzgados de Primera Instancia.

La figura 4.12 muestra la representación en diagramas de Warner/Orr de la estructura general del sistema político nacional

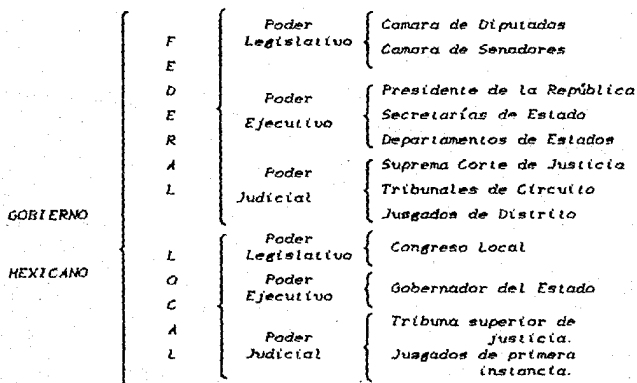


FIGURA 4.12

f CAPITULO IV : DISEÑO ESTRUCTURADO

3) Para el "Control de Alquiler de Trajes" en su fase inicial su estructura con esta técnica indicaría únicamente los módulos principales del sistema. El sistema Control de Alquiler de Trajes tiene dos módulos principales Control de Clientes y Control de Almacén, además de los procesos que dan inicio y finalización al sistema. La figura 4.13 muestra el diagrama que representa la fase inicial.



FIGURA 4.13

En una segunda etapa se muestran los procesos principales de los módulos, así como las funciones principales de los procesos de inicialización y finalización del sistema.

Continuando con el ejemplo 3, el primer módulo está integrado por los procesos de Altas, Consultas y Modificaciones de Clientes, y Entregas de trajes; el segundo módulo se tienen los procesos de Altas, Modificaciones, Bajas, Consultas y Reportes de trajes. Además incluir los procesos de inicialización y finalización de cada módulo.

Las funciones principales del proceso de inicio del sistema son: Inicialización de las variables globales, Actualización de la fecha, presentación del sistema y Autorización de entrada al sistema. Las funciones del proceso de finalización del sistema son: Obtención de respaldos, Reindexación de archivos y Cierre de Bases de Datos. La figura 4.14 muestra el diagrama correspondiente a ésta etapa.

CAPITULO IV : DISEÑO ESTRUCTURADO

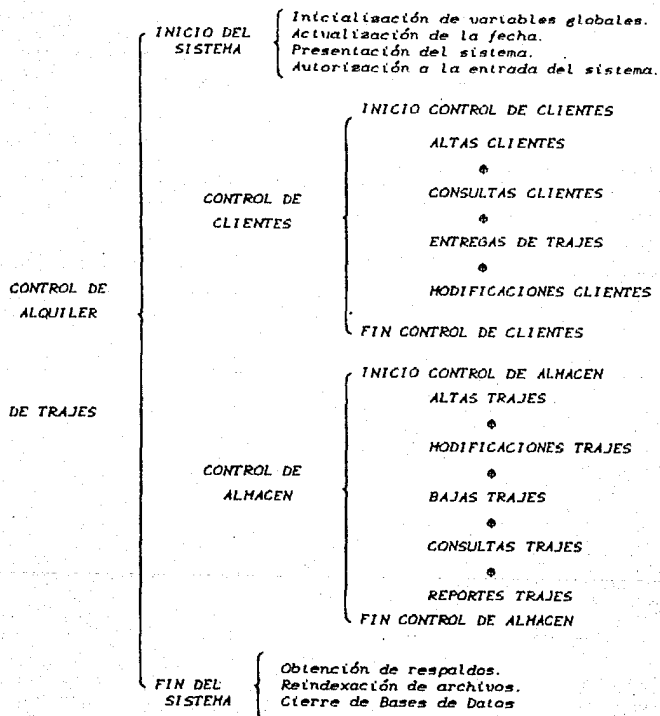


FIGURA 4.14

† CAPITULO IV : DISEÑO ESTRUCTURADO

En la siguiente fase , se desglosan las funciones principales de los procesos, por ejemplo *Alta*s Clientes deberá realizar las siguientes funciones: *Captura General de los datos del cliente y de los datos del traje a alquilar, el Cálculo de Precios del costo del alquiler del traje e Impresión de nota de alquiler.*

El proceso *Inicio Control de Clientes*, por ejemplo, hará la inicialización de las variables globales a ese módulo, abrirá los archivos para Clientes, realizará la presentación del módulo y presentará el menú para la elección de la operación a realizar. Operaciones similares serán ejecutadas por el proceso de inicio del modulo de control de almacén.

La figura 4.15 representa el diagrama obtenido en ésta fase, el diagrama 4.15-A muestra el proceso de inicialización de control de clientes.

INICIO
CONTROL DE
CLIENTES

Inicialización de variables
globales al módulo.
Presentación del Módulo.
Abrir los archivos correspondientes.
Presentación del menú.

FIGURA 4.15-A

Algunas características, de esta técnica, son:

- Muestra agrupamientos de procesos y los datos que se deben pasar de un nivel a otro.
- La secuencia de trabajar en retroceso asegura que el sistema estará bien orientado hacia los resultados esperados (lo cual es un proceso muy natural).
- Muestra los ciclos de una forma muy poderosa, pues están señalados en el mismo diagrama.
- Son simples en apariencia y fáciles de entender.
- Son muy laboriosos.

CAPITULO IV : DISEÑO ESTRUCTURADO

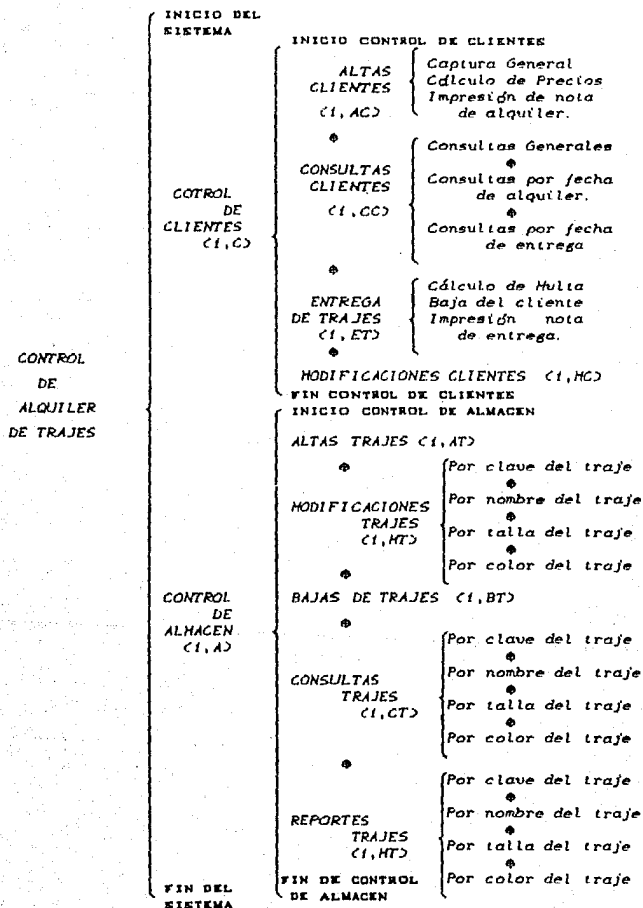


FIGURA 4.15

5. DIAGRAMAS H. I. P.O.

El propósito general, de los diagramas HIPO, es describir y documentar los módulos y su interrelación de manera que proporcione suficiente detalle, pero sin perder de vista el diagrama principal.

Los diagramas HIPO son más bien representaciones gráficas que descripciones narrativas de un sistema. HIPO son las iniciales de Hierarchical plus Input Process Output, que en español significa Jerarquía más Entrada Proceso Salida.

Una descripción HIPO para un sistema consiste de la tabla visual de contenidos (VTOC) y los diagramas funcionales.

Tabla visual de Contenidos:

Muestra la relación entre cada uno de los documentos que constituyen un paquete HIPO. Consiste en un diagrama jerárquico que identifica por número los procesos de un sistema y de una sección de contenidos, la cual proporciona una breve descripción de cada proceso; hay que tener muy en cuenta que los números en la sección de contenidos corresponden a los de la sección de organización. Se puede notar que el diagrama jerárquico funcional está contenido en los diagramas H. I. P. O.

Los procesos empezarán a definirse en detalle creciente, es decir, primero las funciones principales y después los procesos de las funciones con más detalles, es decir, el nivel más alto de la jerarquía describe la función global del sistema; luego se desarrollan niveles jerárquicos adicionales para describir el sistema con mayor detalle cada vez.

Diagrama funcional:

Deberá existir un diagrama para cada caja en la tabla visual de contenidos, cada diagrama deberá mostrar la entrada, la salida, los procesos principales, el movimiento de los datos y los puntos de control.

CAPITULO IV : DISEÑO ESTRUCTURADO

Algunos diagramas funcionales pueden contener otros diagramas intermedios y también mostrar datos externos, así como los datos desarrollados internamente (como el uso o modificación de variables).

Ejemplo:

Regresando al problema de "Control de alquiler de trajes", en las figuras 4.16 y 4.17 se muestra la tabla visual de contenidos HIPO.

La figura 4.16 muestra la sección de contenidos, aquí se describe cada función, proceso y subproceso del sistema, de acuerdo con la numeración de la sección de organización, a manera de ejemplo se describe el contenido de algunos procesos.

La figura 4.17 muestra la sección de organización del sistema, cada función del sistema, así como sus procesos y subprocesos están numerados; esta numeración nos ayuda a localizar el nivel del sistema.

Algunas características de los Diagramas HIPO :

- * Prácticos para documentar un sistema, lo cual es de gran ayuda para los diseñadores.
- * Obligan, a los diseñadores, a pensar sobre cómo se lograrán satisfacer los requerimientos y en dónde deben unirse (relacionarse) las actividades y las componentes.
- * Usan un conjunto de símbolos especializados, que requieren de alguna explicación para su uso, lo que implica menor comunicación con el usuario.
- * No es fácil usarlos para propósitos de comunicación.

CAPITULO IV : DISEÑO ESTRUCTURADO

1.0 Sistema Control de Alquiler de Trajes

Lleva todo el control del alquiler de trajes, llama a los programas que manejan las entradas, actualizaciones y consultas.

2.0 Control de Clientes

Controla todos los datos relacionados con los clientes, llama a los programas encargados de las actualizaciones y consultas.

2.1 Altas Clientes

Realiza la captura y validación de datos de entrada para las altas, además hace el cálculo del precio final del alquiler e imprime la nota de alquiler; y decrementa las las existencias del traje alquilado.

2.2 Consultas Clientes

Hace la validación y captura de datos para poder efectuar las consultas; invoca, según la opción, a los programas encargados de las consultas generales, por fecha de alquiler y por fecha de entrega.

2.3 Modificaciones Clientes

Realiza la captura y modificación de datos de entrada (el nombre del cliente) para llevar a cabo los cambios.

2.4 Entregas Trajes

Realiza la entrada y validación de datos (el nombre del cliente), hace el cálculo de multas, incrementa el número de existencias del traje entregado, dá de baja al cliente e imprime una nota de entrega.

3.0 Control de Almacén

Controla todos los programas relacionados con los trajes, llama a los programas encargados de las consultas (en pantalla y en papel — reportes) y actualizaciones (altas bajas y cambios).

FIGURA 4.16

* Punto de vista acerca de las técnicas presentadas:

Creo que de las técnicas presentadas en este trabajo la más completa es la de los diagramas H.I.P.O, ya que ayudan mucho a los diseñadores pues mientras se está elaborando el diseño del sistema se va documentando el mismo, además por su estructura proporcionan mucha información sobre el diseño del sistema.

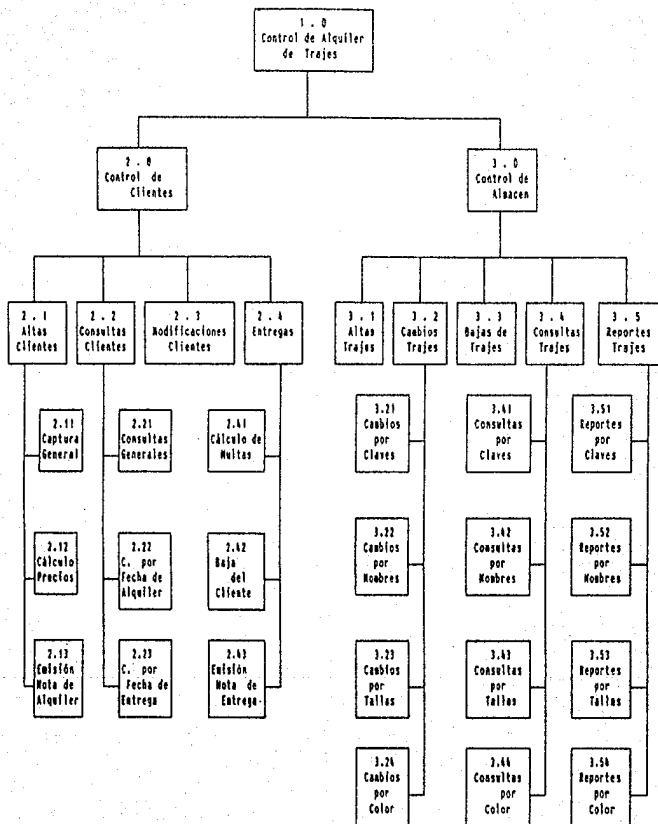


FIGURA 4.11

SISTEMA CONTROL DE ALQUILER DE TRAJES
SECCION DE ORGANIZACION DEL SISTEMA

C A P I T U L O

C I N C O

DESARROLLO DE UN
SISTEMA DE INFORMACION

CAPITULO

C I N C O

DESARROLLO DE UN SISTEMA DE INFORMACION

ANALISIS Y DETERMINACION DE REQUERIMIENTOS.

Problema:

Se tiene una colección de artículos sobre algún área particular, los cuales fueron obtenidos de diversas publicaciones.

Con el volumen de información disponible resulta casi imposible saber exactamente qué se tiene, es decir qué artículos hay de determinado tema o si existe algún artículo en especial.

Objetivo:

Realizar un sistema automatizado que dé apoyo al dueño de la colección, en el sentido de que le proporcione un acceso sencillo y flexible de la información y además le permita tener un adecuado control de la misma.

Limitantes:

No se tendrá el artículo completo, sólo información referente a la descripción general, como título, tema, autores, breve descripción del mismo, revista de donde fue obtenido.

Requerimientos Generales:

Se necesitan funciones que realicen actualizaciones y consultas de la información a manipular, de una manera eficiente y fácil que además proporcionen el mantenimiento necesario.

CAPITULO V : DESARROLLO DE UN SISTEMA

Los datos con que cuenta cualquier artículo son:

- Título
- Breve descripción
- Autor(es)
- Tema o área

además de los datos de la revista que lo contiene como:

- Nombre de la revista
- número
- volumen
- mes
- año
- descripción de la misma (Tipo de publicación, período de edición, país de edición);

y otros datos que han de servir para control como:

- saber si está o no prestado algún artículo

Para el manejo de la colección se requiere de funciones que permitan:

1. Consultar la información ya sea de algún artículo en especial o de varios que cumplan con ciertas características.

Las formas en que podrá consultarse la información son:

a) Dada una característica exacta del artículo:

- Título
- Autor(es)
- Tema
- Revista
- Subtema

b) Dada una combinación de características del artículo, por ejemplo:

- Título y autor
- Tema y autor
- Tema, subtema y revista

c) Dada una característica no exacta del artículo, ejemplos:

- Título incompleto
- Tema incompleto
- Subtema incompleto

d) Dadas varias características no necesariamente exactas del artículo, por ejemplo:

- Título incompleto y autor(es)
- Título incompleto y tema completo
- Tema completo, subtema incompleto

e) Todos los artículos existentes en la colección.

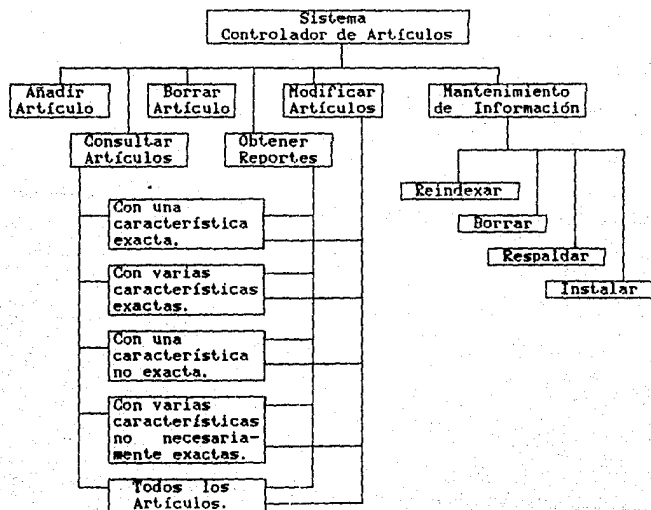
2. Obtener reportes impresos de la información consultada.

3. Añadir nuevos datos (artículos "nuevos" para la colección) a los ya almacenados.

CAPITULO V : DESARROLLO DE UN SISTEMA

4. *Borrar datos (eliminar artículos) que por alguna razón ya no existen en la colección.*
5. *Modificar la información de algún artículo específico o de varios que satisfagan ciertas características. De manera similar que en las consultas.*
6. *Ordenar los archivos periódicamente.*
7. *Obtener periódicamente respaldos de los archivos.*
8. *Borrar archivos que por alguna razón fueron dañados o ya no son necesarios.*
9. *Instalar por vez primera el sistema, generando la estructura de los archivos e índices necesarios.*

Dados los requerimientos anteriores, como primer diseño (diseño conceptual) se puede crear la siguiente estructura del sistema.



METODOLOGIA

Para el desarrollo de este sistema se utilizará una combinación de todas las técnicas antes mencionadas, tratando de aprovechar cada una de ellas al máximo. Esto significa que se usará:

- Diagrama de Flujo de Datos (D.F.D.) para mostrar procesos que involucren alguna interrelación con los archivos de la base de datos.
- Diagrama Jerárquico Funcional (D.J.F.) para ilustrar la jerarquía general del sistema y sus módulos. Estos diagramas serán complementados con la descripción de cada módulo.
- Cajas Negras cuando se requiera algún proceso simple que aún no haya sido diseñado del cual se conozcan los datos que va a usar y los resultados esperados.
- Diagramas de Warnier/Orr para especificar con más detalle algunos procesos o partes de ellos.

El sistema será realizado en un Sistema Manejador de Bases de Datos (DBMS), es decir, en un sistema que permite manipular la información de Bases de datos, desde crearlas y relacionarlas entre sí hasta emplear las relaciones para uno o varias aplicaciones al mismo tiempo. Clipper, dBase y Oracle, entre otros, son sistemas manejadores de bases de datos.

En este trabajo se utilizará Clipper para el desarrollo del sistema, ya que es muy versátil, cuenta con diversas funciones e instrucciones que facilitan el manejo de las Bases de datos y la creación de sistemas (programas o ambientes) que las manipulen. Clipper es un compilador con una gran variedad de utilerías que hacen "no tan difícil" su manejo. Permite además combinarlo con otros programas ya compilados en C, Pascal, Ensamblador, entre otros.

DISEÑO

DISEÑO DE LA BASE DE DATOS.

Al observar detalladamente un artículo cualquiera, es fácil encontrar que los datos que tienen en común son: el título, tema y/o subtema, una breve descripción, los autores, número y volumen de la revista, además del mes y año de edición.

Ejemplo :

TITULO : "On the criteria used in descomposing system into modules"

TEMA : Programming techniques.

SUBTEMA : Modularization.

AUTOR : Pranas, D.L.

BREVE DESCRIPCION :

This paper discusses modularization as a mechanism for improving the flexibility and comprehensibility of a system. The effectiveness of a "modularization" is dependent upon the criteria used in dividing the system into modules.

REVISTA : Communication of A.C.M.

NUMERO : 12

VOLUMEN : 15

MES : December.

AÑO : 1972

Pero además es necesario saber si han sido prestados y tener una breve descripción de la revista.

Hasta este momento se han recolectado los datos necesarios para identificar un artículo específico. Tales datos son vínculos binarios para el artículo (figura 5.01), los cuales formarán múltiples vínculos binarios (figura 5.02) que al reorganizarlos nos proporcionan un Registro (n-ada) con los atributos del artículo (figura 5.03).

CAPITULO V : DESARROLLO DE UN SISTEMA

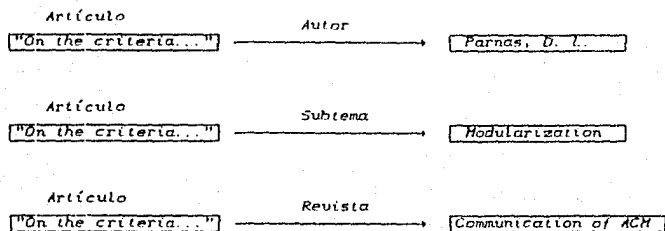


figura 5.01

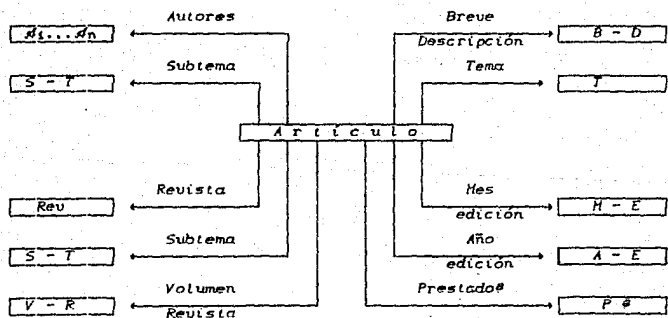


figura 5.02

CAPITULO V : DESARROLLO DE UN SISTEMA

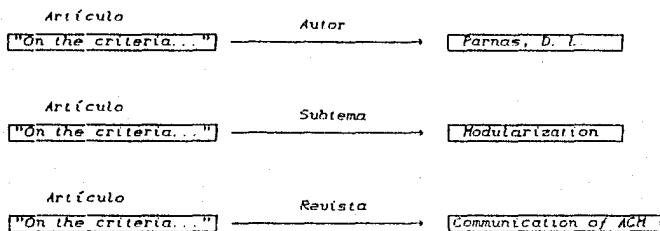


figura 5.01

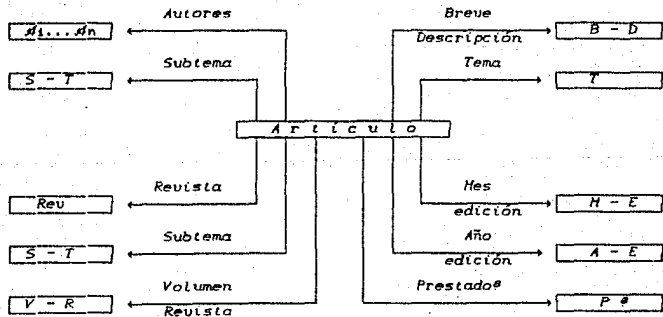


figura 5.02

CAPITULO V : DESARROLLO DE UN SISTEMA

Registro: *Artículos.*

Campos o Atributos	
NUM	Nombre
01	Título
02	Tema
03	Subtema
04	Descripción del artículo
05	Autores
06	Revista
07	Descripción de la revista
08	Número del ejemplar
09	Volumen del ejemplar
10	Mes-Edición de la revista
11	Año-Edición de la revista
12	Prestado #

figura 5.03

Pero si se declarará la Base de Datos de un sólo archivo cuya estructura esté formada por el registro anterior, se tendría información repetida. Supongase que se tienen dos o más artículos que fueron publicados por la misma revista, entonces toda la información acerca de tal revista (de los campos 06 al 11) estarían dos o más veces incluidas en el archivo de datos, esto es, obviamente, redundancia.

Separando los datos se tendrían, entonces, dos archivos para la Base de Datos: uno para los artículos y otro para las revistas. Pero además se debe tener alguna forma de relacionar a ambos, para ello se creará una clave única para la revista que no será manipulada por el usuario.

Bien, ahora, la Base de Datos es un conjunto de dos archivos, *Artículos* y *Revistas*, cuya estructura es mostrada por la figura 5.04.

CAPITULO V : DESARROLLO DE UN SISTEMA

Registro: *Artículos.*

Campos o Atributos	
Núm	Nombre
01	Título
02	Tema
03	Subtema
04	Descripción
05	Autores
06	Clave-Revista
07	Prestado #

Registro: *Revistas.*

Campos o Atributos	
Núm	Nombre
01	Clave-Revista
02	Nombre
03	Descripción
04	Número Ejemplar
05	Volumen Ejemplar
06	Mes Edición
07	Año Edición

figura 5.04

Aún con los cambios hechos en el archivo de *Revistas* hay la posibilidad de tener datos redundantes. Si existen más de dos artículos obtenidos exactamente del mismo ejemplar, entonces la clave, el nombre y la descripción de la revista serán datos, repetidos dos o más veces. Sería ideal, en este caso, tener un catálogo de *Revistas* que estuviese formado únicamente por tales atributos.

Al reorganizar los datos, se tiene que la *Base de Datos* consta de tres archivos: un catálogo de *Revistas*, el archivo de *Artículos* y el archivo de *Ejemplares*.

Para relacionar tales archivos se tendrán dos llaves, una que relacione al artículo con el ejemplar del que fue obtenido y otra que relacione al ejemplar con la revista. La clave del ejemplar (Cue_E) será generada después de obtener los datos particulares de la revista y será una concatenación de la clave de la revista (Cue_R) con el año de edición y el número de ejemplar. Se toman estos datos para identificar la revista a la cual pertenece tal ejemplar y evitar claves repetidas, pues es casi imposible que existan ejemplares diferentes de la misma revista con igual año y número de edición.

CAPITULO V : DESARROLLO DE UN SISTEMA

La nueva estructura de los registros de la Base de Datos se describe en la figura 5.05.

Registro: *Revistas.*

Campos o Atributos	
Núm	Nombre
01	Cue_R
02	Nombre-Revista
03	Descripción

Registro: *Ejemplares.*

Campos o Atributos	
Núm	Nombre
01	Cue_E
02	Nombre Ejemplar
03	Volumen
04	Mes Edición
05	Año Edición
06	Revista (Cue_R)

Registro: *Artículos.*

Campos o Atributos	
Núm	Nombre
01	Título
02	Tema
03	Subtema
04	Descripción
05	Autores
06	Ejemplar (Cue_E)
07	Préstado *

figura 5.05

Tal vez esta estructura para la Base de Datos del sistema estaría bien y funcionaría bien, pero aún existe un "pequeño problema": los autores. La mayoría de los artículos son escritos por más de un autor, es absurdo querer tener en un único campo los nombres de tres o cuatro autores, además, pensando en la posibilidad de tener una consulta por algún autor determinado tampoco sería recomendable declarar tres o más campos en la estructura del archivo (uno para cada autor) ya que existen artículos que han sido escritos sólo por uno o dos autores.

CAPITULO V ; DESARROLLO DE UN SISTEMA

En realidad gran parte de los artículos son escritos por uno o dos autores y muy pocos por más de dos. Si se declararán tres o más campos (uno para cada autor) tal estructura, después de un tiempo, puede ocasionar problemas, pues se estará desperdiciando espacio en la memoria física. Una buena opción es tener una lista de autores donde únicamente exista el nombre del autor y una clave que lo relacione con el artículo que ha escrito.

La llave podría estar dada por la clave del ejemplar (Cue_E) concatenada con la letra 'A' (de autor). Pero aunque es casi nula la posibilidad de que un mismo autor escriba dos o más artículos para un mismo ejemplar, se debe preveer tal situación. Además, no puede depender únicamente de Cue_E ya que de un ejemplar se pueden obtener varios artículos, y entonces, diferentes autores, que no hayan escrito un mismo artículo, tendrían la misma clave.

Ambas situaciones prevengan confusión en los datos, ya que la clave descrita con anterioridad no sería única. Lo mejor será construir una llave para el artículo y de ahí obtener la clave para los autores, la cual podría estar dada por la clave del artículo concatenada con la letra 'A' (de autor).

La Base de Datos estará formada por cuatro archivos: un catálogo de Revistas, Artículos, Ejemplares y Autores. La nueva estructura de los registros de la Base de Datos es descrita en la figura 5.06.

Ahora bien, la mayoría de las operaciones a realizar por el sistema son consultas, esto implica que se tendrá que hacer un constante acceso a los archivos realizando búsquedas las cuales serán, generalmente, sobre campos de más de veinte caracteres (como tema y subtema), esto implica hacer demasiadas comparaciones para localizar algún dato. Tratando de agilizar las búsquedas se crearán archivos que sólo contengan el nombre del atributo y una clave correspondiente.

Registro: *Revistas.*

Campos o Atributos		
Núm	Nombre	Descripción
01	Cue_R	Clave de la Revista
02	Nombre-Revista	Nombre de la Revista
03	Descripción	Datos Generales

Registro: *Ejemplares.*

Campos o Atributos		
Núm	Nombre	Descripción
01	Cue_E	Clave del Ejemplar
02	Número	Número del Ejemplar
03	Volumen	Volumen del Ejemplar
04	Mes Edición	Mes de Edición del Ejemplar.
05	Año Edición	Año de Edición del Ejemplar
06	Revista	Clave de la Revista

Registro: *Artículos.*

Campos o Atributos		
Núm	Nombre	Descripción
01	Título	Título del Artículo
02	Tema	Tema del Artículo
03	Subtema	Subtema del Artículo
04	Descripción	Descripción del Artículo
05	Autores	Clave de los Autores
06	Ejemplar	Clave del Ejemplar
07	Prestado a	Ha sido prestado el Art. a
08	Cue_A	Clave del Artículo

Registro: *Autores.*

Campos o Atributos		
Núm	Nombre	Descripción
01	Cue_W	Clave del Autor
02	Nombre_W	Nombre del Autor

figura 5.06

CAPITULO V : DESARROLLO DE UN SISTEMA

Se tendrá un archivo para clasificar a todos los temas y cuyos únicos campos serán: Nombre y clave del tema. Similarmente se creará un archivo para los subtemas.

Las claves serán creadas de la siguiente manera:

Cue_R (clave de la revista): Será una concatenación de la primera letra del nombre de la revista con una cadena de dos dos dígitos. Como ejemplo, supongase que el nombre de la "revista a guardar" es Communication of ACM y que es la primera revista cuyo nombre empieza con C, entonces su clave será C01; ahora bien supongase que la segunda revista con letra C sea Computing World, su clave será C02. Los datos son tomados de esta forma para así dar opción a tener una amplia colección de revistas, al menos 99 para cada letra del abecedario (25), es decir se podría tener una colección de casi 2500 Revistas; y se utilizan sólo dos dígitos tomando en cuenta que se trata de una colección particular. Si el sistema fuera implementado para una biblioteca o para una colección muy grande se tomarían, entonces, tres o más dígitos.

Cue_E (clave del ejemplar): Será la unión de la clave de la revista con el año de edición (los dos últimos dígitos) y el número de la revista (una cadena de tres dígitos).

Cue_A (clave del artículo) : Será la concatenación de la primera letra del título con una cadena de tres dígitos.

Cue_W (clave del autor) : Será la unión de la clave del artículo con el caracter 'A'.

Cue_T (clave del tema) : Será la unión de la primera letra del nombre del Tema con una cadena de dos dígitos. Por razones similares a las de Cue_R, se toman tales datos.

CAPITULO V : DESARROLLO DE UN SISTEMA

Cve_S (Clave del Subtema) : Será la unión de la primera letra del Subtema con una cadena de dos dígitos.

Cabe mencionar que las claves serán asignadas según se vayan incluyendo al sistema los datos, es decir, no importa el orden alfabético del atributo para asignarla y que éstas no serán manejadas por el usuario. Existen dos razones importantes por las cuales se evita esto.

En primer lugar porque es muy molesto aprenderse claves para tener que manejar un sistema, es decir, es muy incomodo tener que manejar un sistema por medio de claves que generalmente no tienen ningún sentido para el usuario, al menos no cuando empieza a conocer el sistema.

Y en segundo, porque podrían alterar (intencional o accidentalmente) la información. Por ejemplo si se modifica la clave de una revista podrían quedarse varios ejemplares y artículos sin "conexión" con su revista. Aunque esto pueda ser considerado por el programa y modificado internamente, no tiene caso alterar toda la base de datos simplemente por un accidente o una mala intención.

Finalmente, la Base de Datos estará formada por seis archivos: Revistas, Artículos, Ejemplares, Autores, Temas y Subtemas. Pero aun se puede simplificar ya que en el archivo Ejemplares se tiene el campo REVISTA el cual relaciona a este archivo con el de Revistas, pero por la forma como se está construyendo la clave del ejemplar (CVE_E) no es necesario tener este campo ya que REVISTA = CVE_R \subseteq CVE_E, y es relativamente fácil obtener una subcadena de una cadena. Ahora se tendrían dos o tres operaciones más en el programa y un campo menos en cada ejemplar. Para el archivo Artículos existe una situación similar con la clave del autor (Cve_A) y la clave del artículo (Clave_A).

La estructura final de los registros de la Base de Datos es la descrita a continuación.

CAPITULO V : DESARROLLO DE UN SISTEMA

Base de Datos del Sistema

Registro: *Revistas.*

Campos o Atributos		
Núm	Nombre	Descripción
01	Cue_R	Clave de la Revista
02	Nombre-Revista	Nombre de la Revista
03	Descripción	Datos Generales

Registro: *Ejemplares.*

Campos o Atributos		
Núm	Nombre	Descripción
01	Cue_E	Clave del Ejemplar
02	Número	Número del Ejemplar
03	Volumen	Volumen del Ejemplar
04	Mes Edición	Mes de Edición del Ejemplar
05	Año Edición	Año de Edición del Ejemplar
06	Revista	Clave de la Revista

Registro: *Artículos.*

Campos o Atributos		
Núm	Nombre	Descripción
01	Título	Título del Artículo
02	Tema	Clave del Tema
03	Subtema	Clave del Subtema
04	Descripción	Descripción del Artículo
05	Autores	Clave de los Autores
06	Ejemplar	Clave del Ejemplar
07	Prestado #	Ha sido prestado el Art. #
08	Cue_A	Clave del Artículo

CAPITULO V : DESARROLLO DE UN SISTEMA

Registro: *Autores*.

Campos o Atributos		
Núm	Nombre	Descripción
01	Cue_W	Clave del Autor
02	Nombre_W	Nombre del Autor

Registro: *Temas*.

Campos o Atributos		
Núm	Nombre	Descripción
01	Cue_T	Clave del Tema
02	Tema	Nombre del Tema

Registro: *Subtemas*.

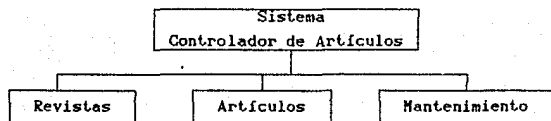
Campos o Atributos		
Núm	Nombre	Descripción
01	Cue_S	Clave del Subtema
02	Subtema	Nombre del Subtema

Durante todo este desarrollo la preocupación principal ha sido el evitar la redundancia de datos y las conexiones innecesarias e ineficientes entre los archivos de la Base de Datos, pero θ es que pasa con el tiempo de ejecución del programa θ es será más lenta la ejecución del sistema por el constante acceso a los archivos θ . Sinceramente creo que no, y en todo caso se está sacrificando tiempo de ejecución por eficiencia de programación, incluyendo la facilidad de uso que se espera tener de este sistema.

MODULARIZACION DEL SISTEMA.

Volviendo a analizar la Base de Datos resulta obvio que será "más fácil" el manejo de información si las funciones que han de manipularla son separadas, es decir, si se tienen tareas específicas para el manejo de las revistas y procesos especiales para los artículos.

Lo cual da lugar a la creación de un módulo para Revistas y otro para Artículos. Dada tal situación se tiene, para empezar, la siguiente estructura del sistema:



Ahora bien, para una revista se tienen datos que la describen de manera global y datos que determinan algún ejemplar en particular. Esto permite dividir el módulo de Revistas en dos submódulos, uno que maneje la información general de la revista como un Catálogo de Revistas y otro que organice los datos particulares de la Revista como Ejemplares.

La figura 5.07 ilustra la estructura jerárquica del sistema la cual incluye las funciones que fueron especificadas para el módulo Mantenimiento.

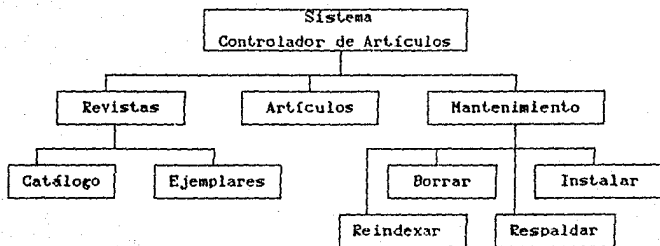


Figura 5.7

DESCRIPCION DEL OBJETIVO DE CADA MODULO.

* **Revistas.**

Llevar el control de la información acerca del Catálogo de revistas y de los Ejemplares.

* **Catálogo.**

Manejar la información general acerca de la revista realizando actualizaciones y consultas de la misma.

* **Ejemplares.**

Permitir actualizar y consultar los datos particulares de las revistas.

* **Mantenimiento.**

Realizar las operaciones necesarias para que los archivos de datos se mantengan en condiciones óptimas para la ejecución del sistema.

CAPITULO V : DESARROLLO DE UN SISTEMA

Nota:

En los módulos anteriores:

1. Las actualizaciones incluirán el anexar, eliminar y modificar la información.
2. Las consultas podran ser en pantalla o en impresora. Permitiendo consultar los datos según las características especificadas por el usuario.

Finalmente la estructura jerárquica del sistema queda ilustrada

en el diagrama 1.

Control de Artículos

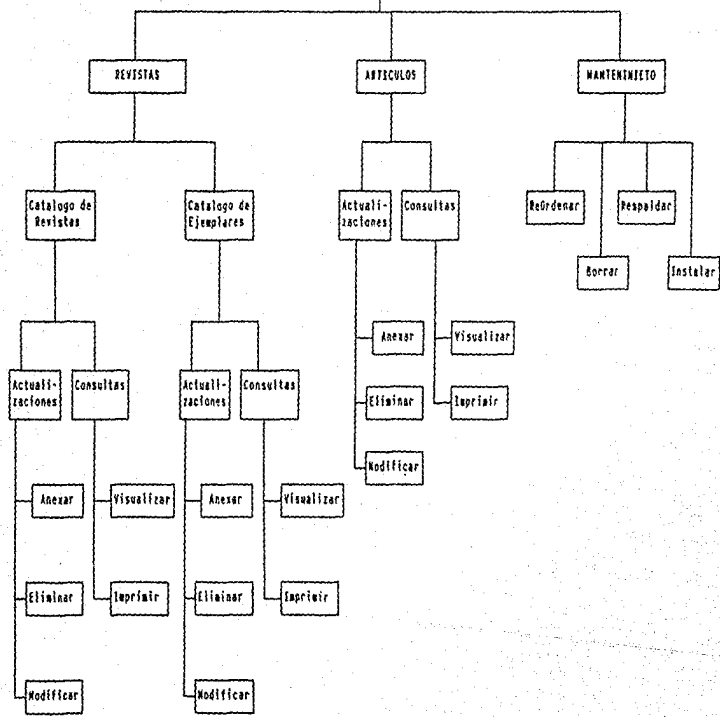


DIAGRAMA 1

DIAGRAMA JERARQUICO FUNCIONAL
SISTEMA CONTROL DE ARTICULOS

DISEÑO EN DETALLE.

Los primeros tres niveles del diagrama jerárquico funcional del sistema (Diagrama 1) son programas de control, es decir, son pequeños programas que generan los menús de selección y llaman a las rutinas necesarias según la opción elegida. En el módulo de revistas el cuarto nivel es, también, de menús.

El programa principal, P, muestra el menú para seleccionar uno de los tres módulos que constituyen el sistema (Revistas, Artículos, Mantenimiento) o bien para terminar con la ejecución del sistema.

El módulo de Revistas será manejado por un programa, PR, que contiene los menús para manipular los submódulos de Catálogo de Revistas y Ejemplares. Tal programa está integrado por varias funciones que serán activadas según la opción del menú.

El módulo de Artículos es controlado por un programa, PA, que está compuesto por los menús de los submódulos de Actualizaciones y Consultas. Este programa también estará integrado por diferentes funciones las cuales serán ejecutadas según la selección del menú.

El módulo de Mantenimiento será manipulado por un programa, PH, que presenta el menú para la selección de la utilería a realizar. Cabe mencionar que este módulo tendrá una clave de acceso como precaución dada la magnitud de los cambios que pueden hacerse en los archivos de datos y que pueden dañar internamente al sistema.

Notas Generales.

- El sistema contará con una amplia instalación de ayudas, las cuales se darán a casi todos los niveles, desde la descripción general de módulos o funciones [F1] hasta ayudas especiales que permiten consultar y/o elegir algún dato en especial [F2, F3, F4, F5, F6] para las Actualizaciones y consultas de Revistas y Artículos.

- Los menús generales de selección (de módulos, submódulos y procesos específicos) aparecerán en el centro de la pantalla.

CAPITULO V: DESARROLLO DE UN SISTEMA

- Siempre será posible cancelar las funciones en ejecución, [F10 o ESC], y regresar al nivel anterior, también será posible cancelar la ejecución total del sistema (regresar al sistema operativo, [F9]) desde cualquier nivel y en cualquier momento.

- Durante la ejecución del sistema siempre estará fija la pantalla que indicará, en el primer renglon, el nombre del sistema SCAR (Sistema Controlador de Artículos), en el segundo renglon, a la extrema izquierda, la fecha actual (la capturada por el sistema operativo). El último renglón tendrá las indicaciones necesarias para activar las ayudas. De ser necesario, según el módulo, el penúltimo y antepenúltimo renglón tendrán indicaciones extras auxiliares.

SCAR					
17/ v / 00					
— Menu Principal —					
<table border="1"><tr><td>REVI STAS</td></tr><tr><td>ARTICULOS</td></tr><tr><td>MANTENIMIENTO</td></tr><tr><td>FINALIZAR</td></tr></table>		REVI STAS	ARTICULOS	MANTENIMIENTO	FINALIZAR
REVI STAS					
ARTICULOS					
MANTENIMIENTO					
FINALIZAR					
[F11] AYUDAS ↑ ANT. ↓ SIG. [F10]/[ESC] REGRESAR [F9] FIN					

Pantalla Principal del Sistema SCAR

- Para las consultas y modificaciones, el nombre de la revista, tema o subtema podrá ser dado completo o incompleto (sin importar si la "frase" está al principio, en medio o al final del nombre).

CAPITULO V : DESARROLLO DE UN SISTEMA

DESCRIPCION DETALLADA DE LOS PROCESOS (PROGRAMAS).

Proceso P :

Como se indicó anteriormente el programa principal, P, es el que llama a las funciones necesarias para seleccionar un módulo. En general, las actividades de P son: validar la clave de acceso, declarar las variables globales del sistema, así como sus variables locales, instalar las rutinas generales del sistema, desplegar el menú principal y poner la línea de ayudas. Al finalizar la ejecución del sistema, cierra (por precaución) todos los archivos de datos, reinstaura el color "estándar", limpia la pantalla y regresa al sistema operativo. Las figuras 1 y 2 representan los diagrama de Warnier/Orr del proceso P.

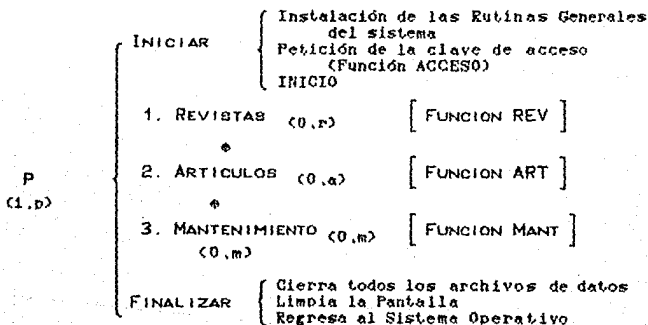


FIGURA 1

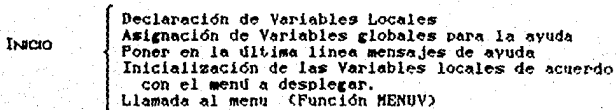


Figura 2

CAPITULO V : DESARROLLO DE UN SISTEMA

Función ART :

La función ART, la cual es la función principal del programa PA, es el proceso que manipula el módulo de Artículos. En términos generales, las instrucciones que realizará son: Instalar las rutinas del módulo, declarar e inicializar las variables locales, según el menú que presentará, asignar a las variables globales el valor necesario de acuerdo a esta función, pone, además la línea de ayudas y despliega su menú. Espera la selección y llama a la función correspondiente para que sea ejecutada. Las figuras 2, 3 y 4 ilustran una manera de representar la función en diagramas de Warnier / Orr.

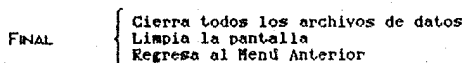


FIGURA 3

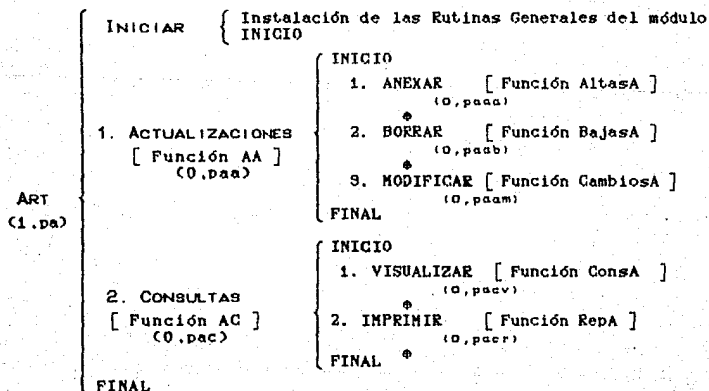


FIGURA 4

Función REV :

Es el proceso principal del módulo de revistas (y del programa PR). Sus actividades son: Instalar su miscelanea de rutinas, declarar e inicializar sus variables locales, asignar a las variables globales los valores necesarios de acuerdo a sí misma, poner además la línea de ayudas, mostrar su menú, el cual ofrece la opción a elegir cualquiera de sus dos submódulos, finalmente, espera la respuesta que activará el siguiente proceso a efectuar. Al finalizar, igual que los procesos anteriores, cerrará los archivos de datos, limpiará la pantalla y regresará al menú principal. Las figuras 2, 3 y 5 muestran una forma de representar esta función en la técnica de diagramas de Warnier / Orr.

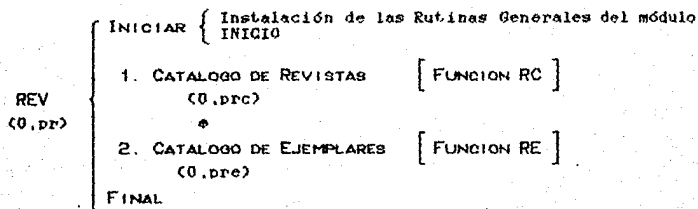


FIGURA 5

Función RC :

Esta función controlará todo el submódulo del catalogo de revistas. Instalará su miscelanea de rutinas, declarará sus variables locales, asignará los valores necesarios a las variables que manejará, pondrá la línea de ayudas y mostrará el menú, cuyas funciones principales serán realizar las actualizaciones (Altas, bajas y cambios) y consultas de las revistas. Finalmente, se asegurará de cerrar los archivos de datos, además limpiará la pantalla y regresará al menú anterior (función REV). Las figuras 2, 3 y 6 muestran en diagramas de Warnier / Orr una forma de representar la función RC.

Función RE :

Esta función manipulará el submódulo del catálogo de revistas. Instalará su miscelánea de rutinas, declarará sus variables locales, asignará los valores necesarios a las variables, pondrá la línea de ayudas y mostrará el menú, cuyas funciones principales serán realizar las actualizaciones (Altas, bajas y cambios de ejemplares) y consultas de las ejemplares. Finalmente, cerrará los archivos de datos, limpiará la pantalla y regresará al menú anterior (función REV). Las figuras 2, 3 y 7 ilustran en diagramas de Warnier / Orr la función RE.

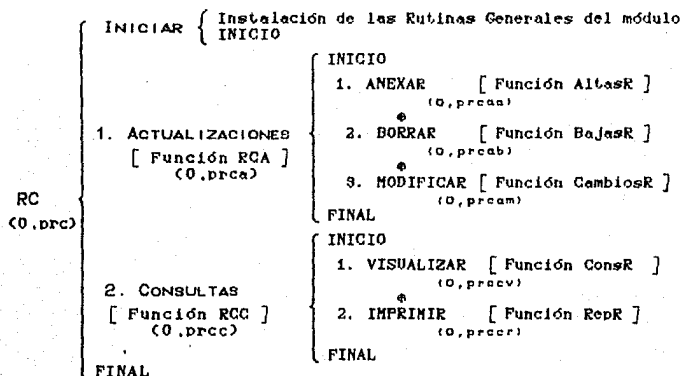


FIGURA 6

Función MANT :

Es la que manejará el módulo de Mantenimiento, programa PM, sus actividades serán: La instalación de sus rutinas, la petición de la clave de acceso, la declaración e inicialización de variables locales, asignación a las variables globales, la instalación de los mensajes para activar las ayudas, el despliegue del menú y la espera de la selección de la función a realizar, de acuerdo con el menú. Al finalizar cerrará los archivos de datos, limpiará la pantalla, y regresará al menú principal. Las figuras 2, 3 y 8 ilustran una representación de la función en diagramas de Warnier / Orr.

CAPITULO V : DESARROLLO DE UN SISTEMA

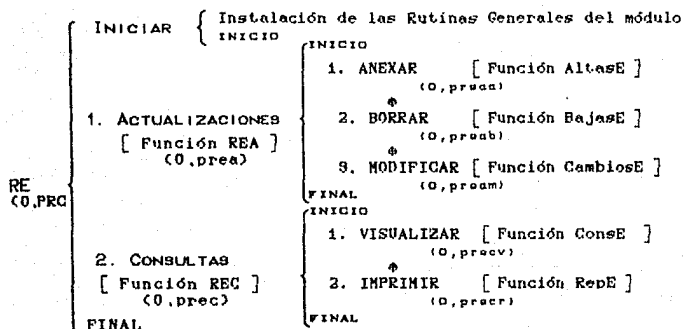


FIGURA 7

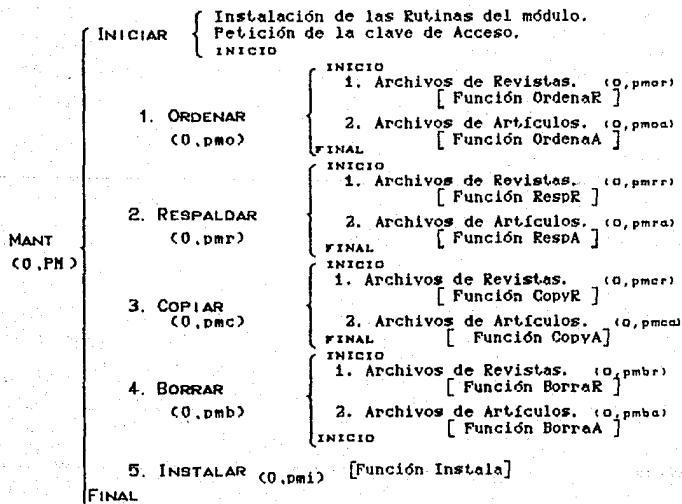


Figura 8

Función AltasR :

Dará la posibilidad de efectuar una, ninguna o varias altas. Pedirá el nombre de la revista, el cual debe ser no nulo, lo pasará a mayúsculas y le quitará los blancos, de esta forma, el dato quedará "estandarizado" para subsecuentes operaciones. Si ya existe la revista o el nombre es nulo (son sólo espacios en blanco o nada) cancelará la operación dando la opción de continuar con la función o regresar al menú anterior. En caso de ser un nuevo dato creará su clave y capturará la descripción, al terminar ésta (la captura de la descripción) pregunta si se desea dar otra alta o regresar al menú anterior (nivel anterior, menú de actualizaciones de Revistas).

Función BajasR :

Se podrán realizar una, ninguna o varias bajas. Pedirá el nombre de la revista, que validaremos y estandarizaremos, el cual podrá ser elegido por la ayuda F2 la cual permitirá seleccionar una revista del archivo de datos de revistas. Si no existe la revista o el nombre es nulo (solo espacios en blanco o nada) cancelará la operación dando la opción de continuar con la función o regresar al menú anterior, el de actualizaciones de Revistas.

Una vez obtenido el dato válido se verifica si existen o no ejemplares de tal revista. Si existen ejemplares de tal, se cancela la baja, en caso contrario se procede a eliminar la revista del archivo de datos. Realizada la operación pregunta si se desea borrar otro dato o regresar al menú anterior (nivel anterior).

Función CambiosR :

Se podrán hacer las modificaciones de las revistas a partir de un nombre dado (NomAux, el cual podrá ser seleccionado con la ayuda F2) hasta la última revista o bien todas las revistas en existencia. Ambas opciones presentarán las revistas en orden alfabético. Se usará la misma función para ambas opciones (Función CambiarR) la cual recibirá un parámetro lógico que indicará si son todas (en este caso el parámetro será verdadero) o si son sólo algunas (entonces el parámetro será falso) las revistas a modificar.

CAPITULO V : DESARROLLO DE UN SISTEMA

No hay que olvidar que se puede cancelar la función en cualquier momento, lo cual permite que se puedan realizar una o ninguna modificación (o tan sólo algunas cuantas).

En caso de no ser la opción que modifica a todas las revistas, se pedirá el nombre de la revista (que podrá ser elegido por la ayuda F2) una vez validado y estandarizado el nombre se construirá un archivo auxiliar que contenga a todas las revistas del archivo de datos cuyo nombre sea mayor o igual al nombre dado. Después checamos si el archivo es vacío, ya sea el auxiliar o el original, si lo es, entonces se manda un mensaje que ha de indicar que no existen datos. Si es no vacío, se mostrarán una a una las revistas del archivo auxiliar para que sean modificadas.

Función ConsultasR :

Esta función será similar a la función anterior (CambiosR) y también posee una subfunción (ConsulR) muy parecida a la subfunción CambiaR, sólo que no se permiten modificaciones, es decir, sólo se muestran los datos.

Función RepR :

Es idéntica a las funciones anteriores (CambiosR y ConsultasR), únicamente que el despliegue de los datos se manda a la impresora.

Función AltasE :

Se tendrá la posibilidad de realizar una, ninguna o varias altas, ya sea de la misma revista o de diferentes. Pedirá el nombre de la revista, que puede ser seleccionado por F2, se estarán capturando datos de ejemplares mientras el usuario lo desee, cada dato del ejemplar será validado, en caso de que sea un ejemplar ya existente, se cancelará esa alta y se podrá continuar dando otras.

Al finalizar las altas de un ejemplar se preguntará si se desea realizar otra alta con la misma revista o con otra. Y al terminar las altas de ejemplares de una revista dada se procederá a preguntar si se quiere dar más altas con otra revista o si se finaliza con la operación y se regresa al menú anterior (nivel anterior, al menú de actualizaciones de Ejemplares).

CAPITULO V : DESARROLLO DE UN SISTEMA

Función BajasE :

Dado el nombre de una Revista se despliegan uno a uno los ejemplares en existencia de tal revista. Se pregunta si se desea borrar el que acaba de ser desplegado, en caso de contestar afirmativamente, se verifica que no existan artículos de tal ejemplar. Si existen, se cancela la opción, se manda un mensaje indicando que no es posible realizar la baja y se pasa al siguiente dato. Si no existen artículos de tal ejemplar se le pone un mensaje de borrado en la pantalla y se borra internamente el dato.

Al finalizar se pregunta si se quiere continuar realizando bajas de ejemplares con otra revista o si se regresa al menú anterior.

Función CambiosE :

Las modificaciones podrán hacerse a partir del nombre de una revista o bien de todas las revistas. Para cada revista se despliegan los datos de sus ejemplares en "pantallas o páginas" de a lo más seis ejemplares cada una.

Al finalizar los cambios se pregunta al usuario si continúa con los cambios o si regresa al menú anterior. Al igual que en las funciones antes definidas, es posible suspender la función en cualquier momento, lo cual permite que se puedan realizar una o ninguna modificación (o sólo algunas cuantas).

Función ConsultasE :

La presente función será idéntica a la anterior (CambiosE) sólo que no se permiten modificaciones, es, decir, sólo es mostrada la información.

Función RepE :

Es muy similar a las dos funciones anteriores (CambiosE y ConsultasE), la diferencia es que la información de los datos se direcciona a la impresora.

NOTA: Para las funciones de ejemplares se tendrá en F3 una ayuda que muestra todos los ejemplares de la revista con la cual se está irabajando.

CAPITULO V : DESARROLLO DE UN SISTEMA

Función AltasA :

Dados los datos de la Revista (Nombre, Número, Volumen, Mes y Año) se pedirá y validará el nombre del artículo a anexar. Si el dato ya existe, se manda un mensaje de error. Si no existe se pedirán y validarán los demás datos, después se crearán las claves necesarias (para autores, temas, subtemas, artículo y ejemplar).

Si los temas y subtemas son nuevos se anexan a los archivos correspondientes. Los autores serán añadidos al archivo de datos de autores con los nombres dados y la clave creada.

Habrà la posibilidad de efectuar una, ninguna o varias altas, así como existirá la opción de cancelar en cualquier momento la ejecución de la operación.

Al finalizar la operación se preguntará si se desea realizar otra alta o si se finaliza con la operación y se regresa al menú anterior (nivel anterior, al de actualizaciones de Ejemplares).

Función BajasA :

Se dará la opción de realizar una, varias o ninguna baja, según algún dato específico del artículo (Título, Autor, Tema, Subtema) o de acuerdo con varios datos a la vez.

Se mostrarán uno por uno los artículos que satisfagan la información especificada y se preguntará al usuario si se desea o no borrar tal dato. En caso de contestar afirmativamente, se procede a borrar los datos necesarios.

Al finalizar se pregunta si se quiere continuar realizando bajas si se regresa al menú anterior.

Función CambiosA :

Las modificaciones serán realizadas de acuerdo con algún dato específico del artículo (Título, Autor, Tema, Subtema) o combinaciones de varios de ellos a la vez.

CAPITULO V : DEBARROLLO DE UN SISTEMA

Los artículos que cumplan con las condiciones dadas, serán mostrados uno a uno para que se realicen los cambios que se crean necesarios.

Al finalizar los cambios se pregunta al usuario si continua con el proceso de modificaciones o si regresa al menú anterior. Al igual que en las funciones antes definidas, es posible suspender la función en cualquier momento, lo cual permite que se puedan realizar una o ninguna modificación (o sólo algunas cuantías).

Función ConsultasA :

La presente función será similar a la anterior (CambiosA) sólo que no se permiten modificaciones, simplemente se muestran los datos.

Función RepA :

Es, en general, idéntica a las dos funciones anteriores (CambiosA y ConsultasA), la diferencia es que la información obtenida es mandada a la impresora.

En el apéndice A se ilustrarán estas funciones en diagramas de Warnier / Orr.

En el apéndice B serán mostrados los flujos de datos de algunas de las funciones aquí descritas.

En el apéndice C se mostrarán por medio de cajas negras algunos procesos que se utilizan en los diagramas de Warnier / Orr y que no fueron descritos ni en este capítulo ni en los apéndice anteriores.

En el apéndice D se ilustrará un ejemplo de modificación al sistema.

C O N C L U S I O N E S

CONCLUSIONES

Durante la elaboración del sistema aquí presentado, en particular, durante el diseño en detalle y la implementación, fue muy útil la organización inicial de los módulos y la descripción general y concisa de las funciones a realizar, sobre todo esta última, pues cuando se transcribían a diagramas de Warnier/Orr (o a los de flujo de datos) se podía notar que algunas funciones eran muy similares, en términos globales, y tal hecho indicaba que era factible obtener procesos comunes a ellas, auxiliándonos de algunos parámetros, en caso de ser necesario. Es decir las bases del diseño estructurado ayudaron a construir el diseño final, el cual también es estructurado. Ayudaron, además, a implementar el sistema, ya que era suficiente hacer las funciones de altas y las bajas, de cualquier módulo, las otras, eran muy similares.

Elaborar los programas, era simplemente "traducir" lo escrito en los diagramas de Warnier en instrucciones de Clipper, únicamente había que tener un poco de cuidado extra en la declaración e inicialización de variables y el despliegue de información, ya fuese en pantalla o en impresora, pues en los diagramas no se le dió la suficiente importancia, estos últimos detalles (mostrar los datos), provocaban, más que errores lógicos, errores "Estéticos", es decir, había que fijarse si el dato mostrado estaría mejor más a la derecha o a la izquierda, o quizá un renglón arriba o abajo, o si el color estaba bien o si fué alterado por algún proceso ejecutado con anterioridad, ... en fin, sólo estética!

CONCLUSIONES

En realidad, la mayoría de los errores que se tuvieron fueron lógicos (variables no inicializadas o mal inicializadas, mal paso de parámetros y ciclos con condiciones erróneas — sobre todo, cuando había contadores de renglones) y estéticos. Los errores de sintaxis, fueron más que nada "errores de dedo". Evaluando un poco tal situación, se puede decir que estos errores fueron "insignificantes" tomando en cuenta que el sistema es muy completo.

Ahora bien, detectar estos pequeños errores fue relativamente fácil, ya que cada proceso está bien identificado, gracias a los diagramas de Warnier, se sabe a que módulo, submódulo o función pertenece el proceso donde aparecía el error. Cabe mencionar que los programas que constituyen el sistema SCAR están organizados de tal forma que hay un programa controlador para cada módulo y un programa para cada función. Lo que significa que una vez hecho el diseño estructurado se tenía ya una amplia documentación del sistema.

Hubo otro tipo de errores lógicos que no dependieron ni del diseño ni de la programación del sistema, sino que fueron errores del "lenguaje" cuyo compilador no "respetaba" o revisaba eficientemente la gramática.

Por otro lado cuando se implementó la mayor parte del sistema hubo necesidad de cambiar y añadir algunas funciones al Módulo de Artículos y dada la modularidad del sistema sólo se modificaron unos cuantos programas, y en este caso, una línea por cada programa, la cual era la llamada a la nueva función a ejecutar.

La instalación de las ayudas estuvo muy relacionada con el esquema modular del sistema y la paquetería con que fue implementado. Aunque Clipper cuenta con varias funciones que auxilian la creación de procesos de ayuda, tener modularizado el sistema facilitó su implementación, ya que para activar las ayudas, basta saber qué función, o rutina, y en qué módulo se está ejecutando.

CONCLUSIONES

Lo anterior trata sobre el trabajo "preliminar" necesario para la instalación y programación del sistema, desde entonces, sobre la organización y documentación que se obtiene para el sistema, es posible aplicar, o bien, modificar totalmente algunas funciones a la estructura de los archivos de datos, de acuerdo con ciertos requerimientos.

Es importante hacer notar, también que el sistema es muy portable, puede ejecutarse en cualquier computador compatible con IBM cuyo sistema sea DOS. La capacidad de los archivos está relacionada con la capacidad de almacenamiento del disco duro o flexible donde está instalado el sistema.

CONCLUSIONES

Lo anterior trata sobre el trabajo "facilitado" durante la instalación y programación del sistema, pero, además, dada la organización y documentación que se obtuvo para el sistema, es posible ampliar, o bien, modificar totalmente algunas funciones o la estructura de los archivos de datos, de acuerdo con nuevos requerimientos.

Es importante hacer notar, también que el sistema es muy portable, puede ejecutarse en cualquier computadora compatible con IBH cuyo sistema sea DOS. La capacidad de los archivos está relacionada con la capacidad de almacenamiento del disco duro o flexible donde esté instalado el sistema.

B I B L I O G R A F I A

BIBLIOGRAFIA

1. Bertalanffy, Ludwig von
Teoría general de los sistemas.
Traducción de la 1a. edición en inglés,
por Juan Almela.
la edición en español, 7a. reimpresión,
Fondo de cultura económico.
México, 1989.
2. Burch, G. J. & Strater, R. F.
Sistemas de información. Teoría y práctica.
Traducción de la 1a. edición en inglés,
por Ricardo Calvet Pérez.
1a. edición en español, 8a. reimpresión.
Editorial Limusa.
México, 1989.
3. Constantine, Larry L. & Yourdon, Edward.
Structured design fundamentals of discipline
of computer program and system
design.
Ed. Prentice-Hall Inc.
New Jersey, U.S.A., 1979.
4. Constantine, L.L., Stevens, W.P. & Myers, G.J.
Structured Design
I.B.H. Syst. J. No. 2, 1974.

BIBLIOGRAFIA

5. Date, C. J.
An introduction to Database systems
Addison Wesley Publishing Company.
U.S.A., 1977.
6. Douglass, T. Ross & Kenneth, E. Shoman Jr.
Structured analysis for requirement definition
7. Giech, John P. van
Teoría General de sistemas.
Traducción de la 2a. edición en inglés.
Editorial Trillas.
México, 1989.
8. Lesk, Michael
Programación del tratamiento de la información.
9. Lucas Henry C. Jr.
*The analysis, design and implementation of
information Systems*
3a. edición, Editorial Mc Graw - Hill
Series in managment information System
E.U.A., 1985
10. Martin, James
Managing the Database Environment.
Ed. Prentice-Hall, Inc.
U.S.A., 1989.
11. Marquez Vite, Juan M.
Sistemas de información por computadora.
Metodología de desarrollo.
1a. edición, Ed. Trillas.
México, 1987.

12. Metzger, Philip W.
Administración de un proyecto de programación,
Traducción de la 1a. edición en inglés,
por Carlos Villegas García.
1a. Edición en español.
Ed. Trillas, México, 1978.
13. Murillo López, Agustín.
*Importancia de la conexión y cohesión en la
elaboración de sistemas de información.*
Tesis para obtener el título de Matemático.
México, 1989.
14. Parnas, D.L.
*On the criteria to be used in decomposing
system into Modules.*
Comm. of A.C.H., 15, 12 (Dec. 1972).
15. Parnas, D.L.
*A technique for software module specification
with examples.*
Comm. of A.C.H., Volúmen 15, número 5
Mayo 1972.
16. Pressman, Roger S.
Software engineering: a practitioner's approach.
Editorial Mc. Graw-Hill.
U.S.A., 1982
17. Schubert, Richard F.
Basic concepts in database management systems.
Byte, Vol 9, No 11, Julio 1972.
18. Seen, James
Análisis y diseño de Sistemas de información.
Mc. Graw-Hill.
México, 1987.

BIBLIOGRAFIA

17. Squire, Enid

Introducción al diseño de sistemas.

1a. edición.

Fondo educativo interamericano.

México 19

20. Wiederhold, Gio.

Diseño de base de datos.

Traducción de la 2a. edición en inglés
por Ma. de Lourdes Fournier García.

1a. Edición en español.

Editorial Mc. Graw-Hill de México.

México, 1980

A P E N D I C E A

DIAGRAMAS DE WARNIER/ORR

DIAGRAMA DE WARNIER / ORR PARA LA FUNCION ALTASR

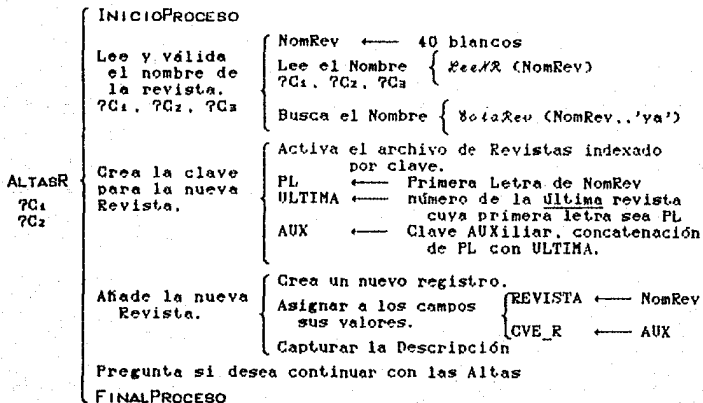


DIAGRAMA DE WARNIER / ORR PARA LA FUNCION BAJASR

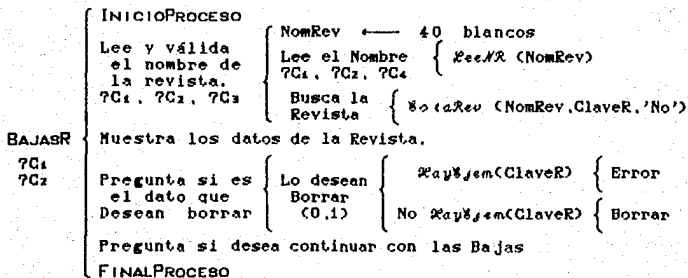
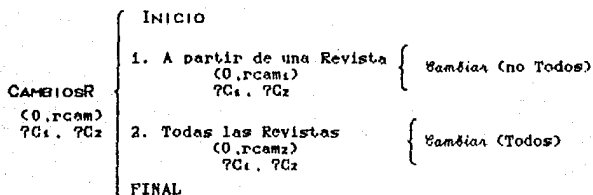
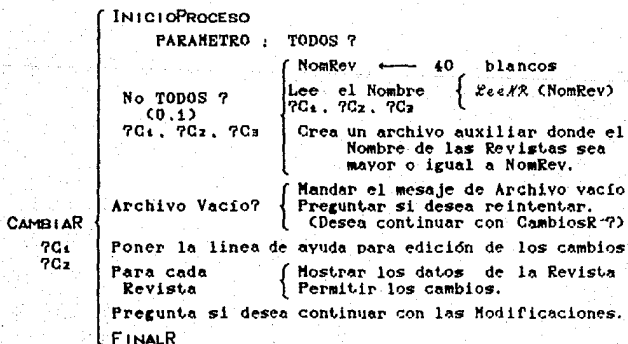


DIAGRAMA DE WARNER / ORR PARA LA FUNCION CAMBIOSR



Para las funciones *ConsR* y *RepR* los diagramas son similares y llaman a las funciones *ConsoulR* y *ReportR*, respectivamente.

DIAGRAMA DE WARNER / ORR PARA LA FUNCION CAMBIAR



El diagrama para la función *ReportR* es semejante al de *CambiaR* únicamente se omite la instrucción "Permitir los cambios".

DIAGRAMA DE WARNER / ORR PARA LA FUNCION REPORTR

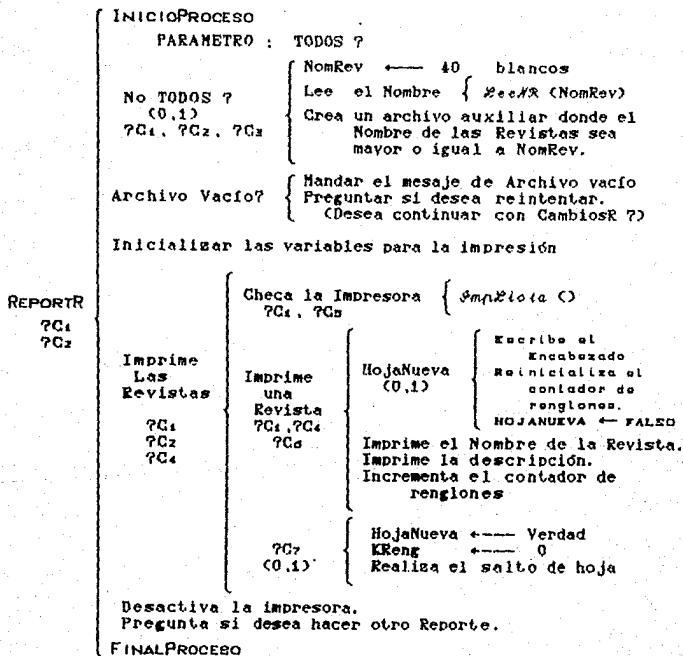


DIAGRAMA DE WARNER / ORR PARA LA FUNCION ALTASE

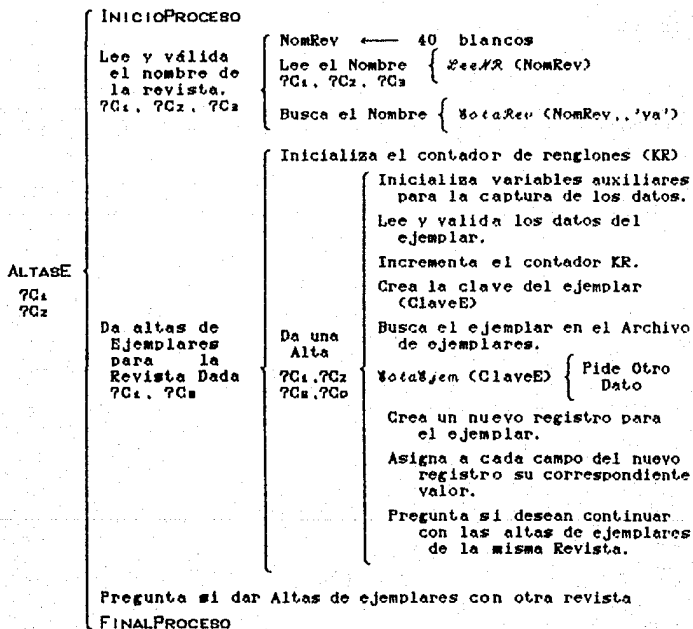


DIAGRAMA DE WARNER / ORR PARA LA FUNCION BAJASE

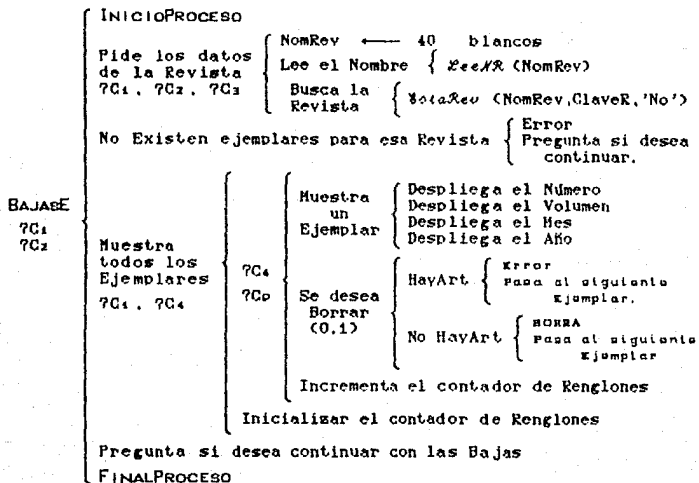
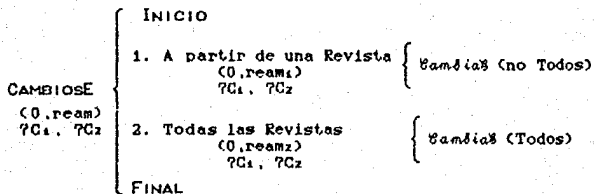


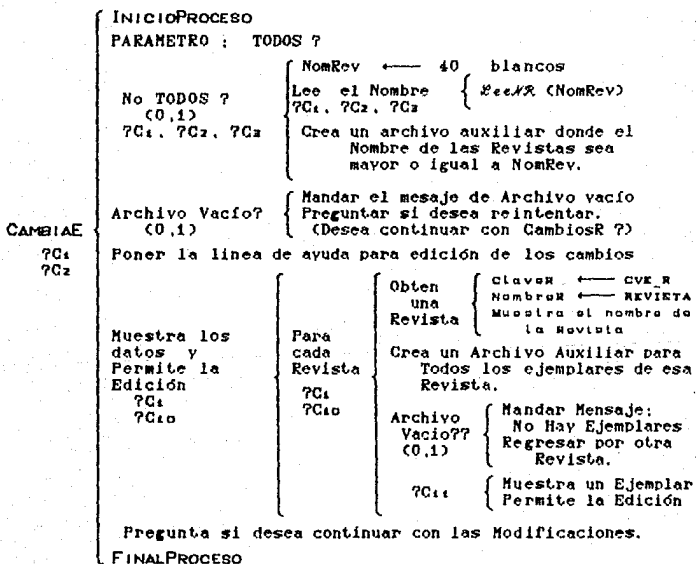
DIAGRAMA DE WARNER / ORR PARA LA FUNCION CAMBIOSE



APENDICE A

Los diagramas para las funciones de *ConoR* y *RepR* los diagramas son similares y las funciones que invocan son *ConoR* y *RepR*, respectivamente.

DIAGRAMA DE WARNER / ORR PARA LA FUNCION CAMBIAE



Para las funciones *ConoR* y *RepR* los diagramas son similares sólo que, para el primero se omite la instrucción "Permite Edición" y para el segun el despliegue de los datos es direccionado a impresora.

DIAGRAMA DE WARNIER / ORR PARA LA FUNCION ALTASA

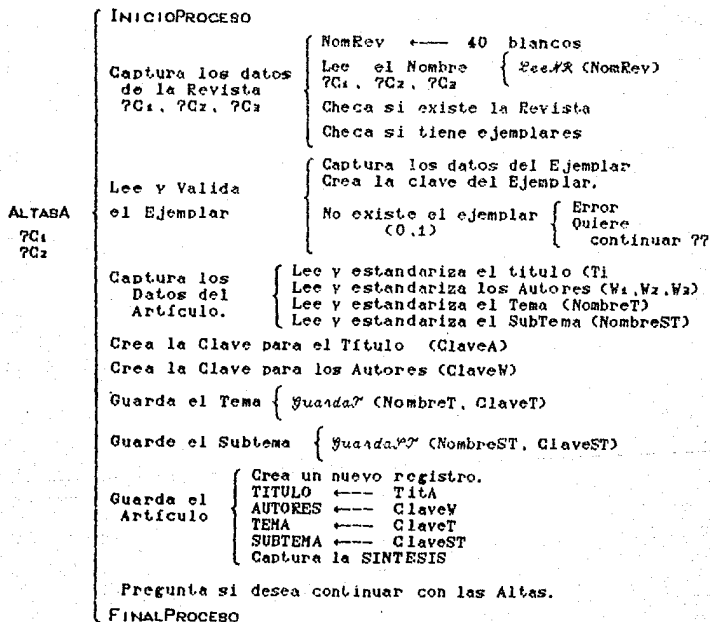
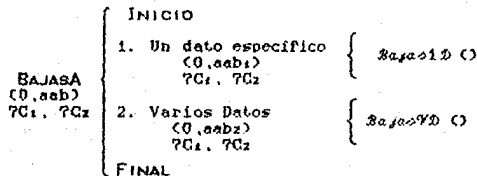
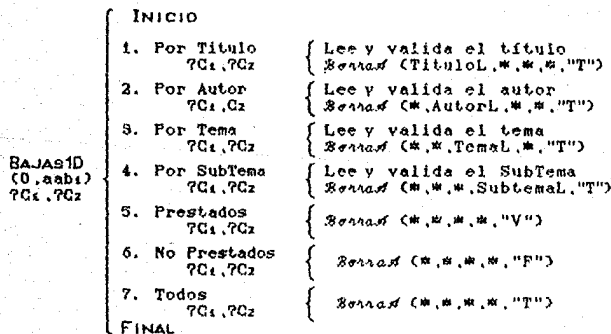


DIAGRAMA DE WARNIER / ORR PARA LA FUNCION BAJASA



Los procesos *CambiasID*, *ConsultID* y *ReportID*, los digramas son muy similares y las funciones que mandan ejecutar son *CambiaID*, *CambiaSID*; *ConsultID*, *ConsultSID*; *ReportID* y *ReportSID*;

DIAGRAMA DE WARNIER / ORR PARA LA FUNCION BAJASID



Los diagramas para las funciones *CambiasID*, *ConsultID* son muy parecidas al diagrama anterior, sólo que llaman a las funciones *CambiaID*, *MuestraID*; *MuestraID*, *EdutoresID*; *ImprimID*, *EdutoresID*.

DIAGRAMA DE WARNIER / ORR PARA LA FUNCION BORRAA

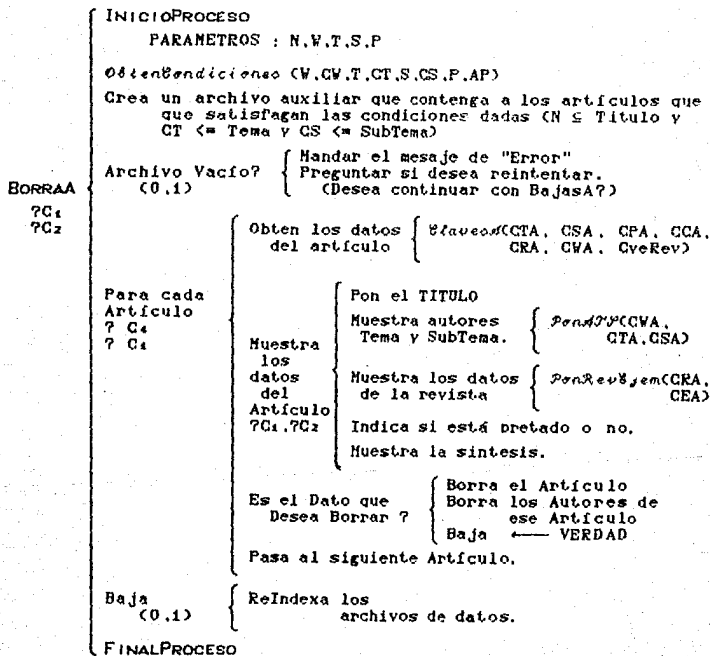


DIAGRAMA DE WARNER / ORR PARA LA FUNCION MUESTRAA

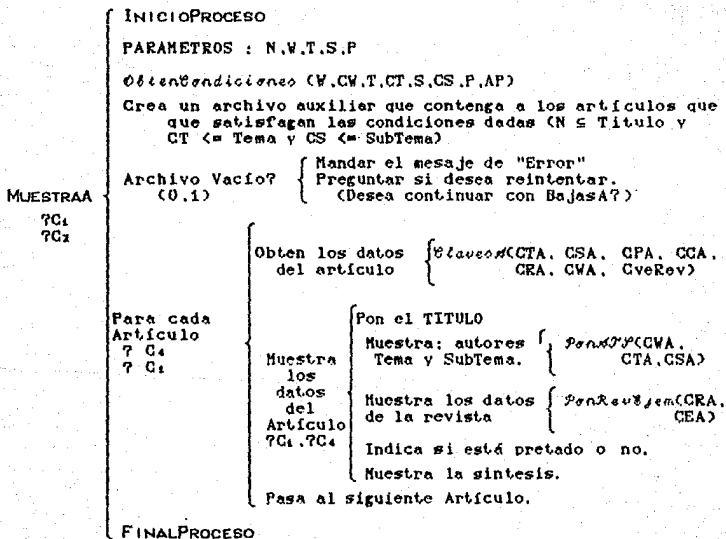


DIAGRAMA DE WARNIER / ORR PARA LA FUNCION CAMBIAA

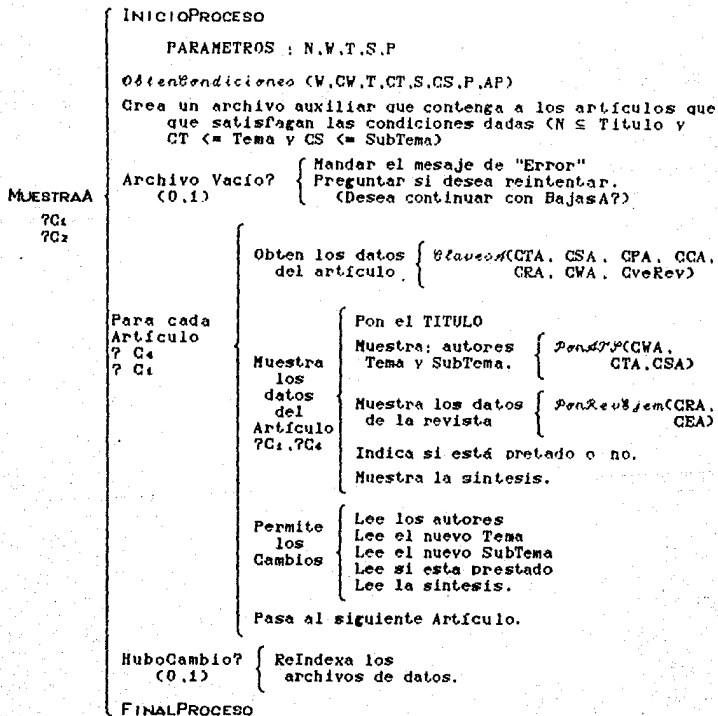


DIAGRAMA DE WARNIER / ORR PARA LA FUNCION IMPRIMEA

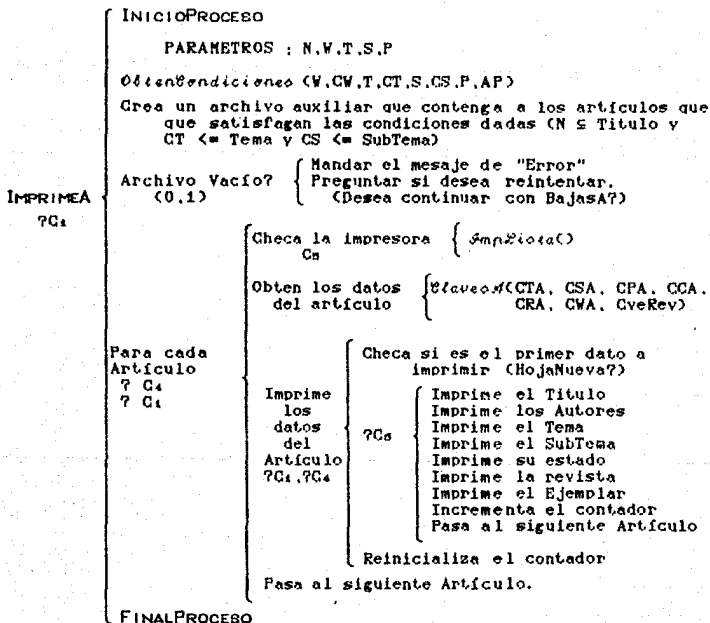
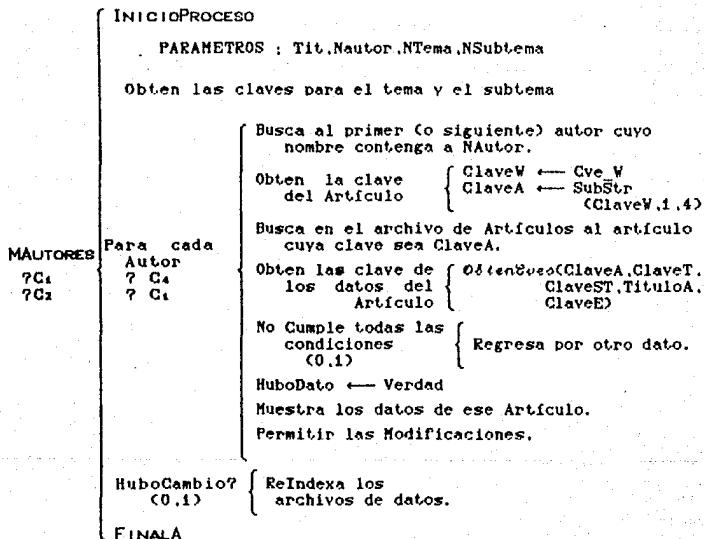
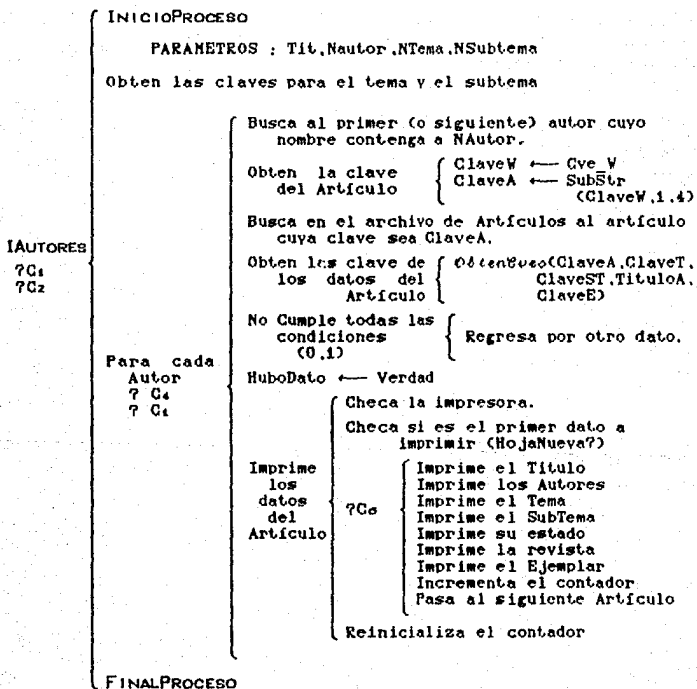


DIAGRAMA DE WARNIER / ORR PARA LA FUNCION MAUTORES



Para la funcion *Estadisticas* el diagrama es similar al anterior, las diferencias consisten en omitir la instrucción "Permitir las Modificaciones" y la verificación de HuboCambio.

DIAGRAMA DE WARNER / ORR PARA LA FUNCION IAUTORES



DIAGRAMAS AUXILIARES

INICIO PROCESO {
 Abre los archivos de Datos.
 Poner el título del proceso.
 Poner la línea de ayuda.
 Inicializar variables globales.
 Activar el proceso de ayudas.
 Inicializar las Variables Locales

FINAL PROCESO {
 Cierra los Archivos de Datos.
 Desactiva el Proceso de Ayudas.
 Regresar al Menú Anterior.

Muestra los Datos de la Revista {
 Despliega el Nombre de la Revista.
 Pon la Línea de Ayudas.
 Despliega la descripción de la Revista.

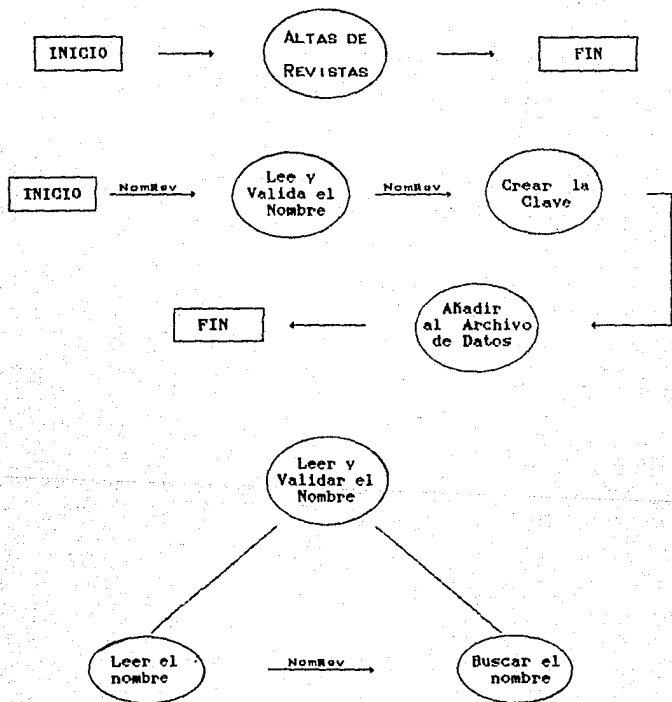
Condiciones de salida:

- C1 : Mientras no opriman las teclas [ESC] o [F9].
- C2 : Mientras el usuario quiera continuar con el proceso.
- C3 : Mientras el nombre de la Revista no sea válido.
- C4 : Mientras no sea fin de archivo.
- C5 : Mientras la impresora no esté prendida.
- C7 : Si el Contador de Renglones es mayor que el máximo de renglones de una hoja.
- C8 : Mientras el usuario desee continuar con altas de ejemplares con la misma Revista.
- C9 : Mientras el contador de renglones sea menor que el número máximo de renglones de una "Pantalla".
- C10 : Mientras existan Revistas.
- C11 : Mientras existan Ejemplares.

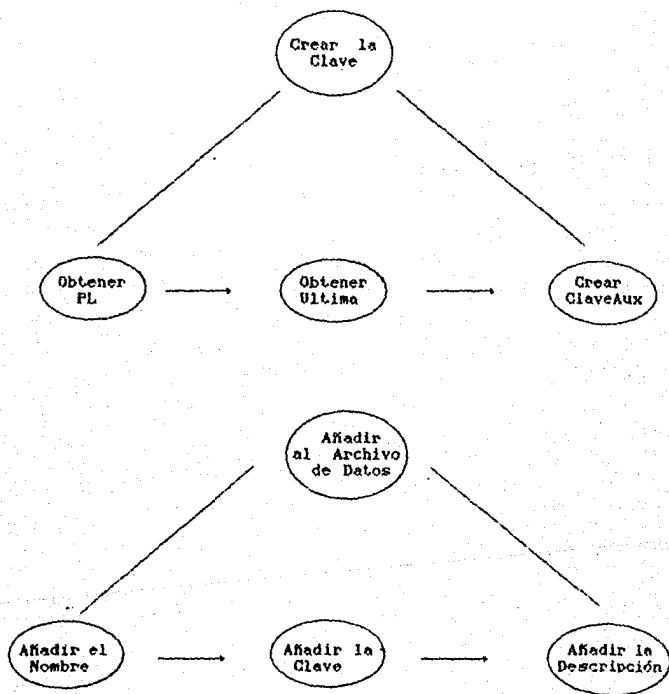
A P E N D I C E B

DIAGRAMAS DE
FLUJO DE DATOS

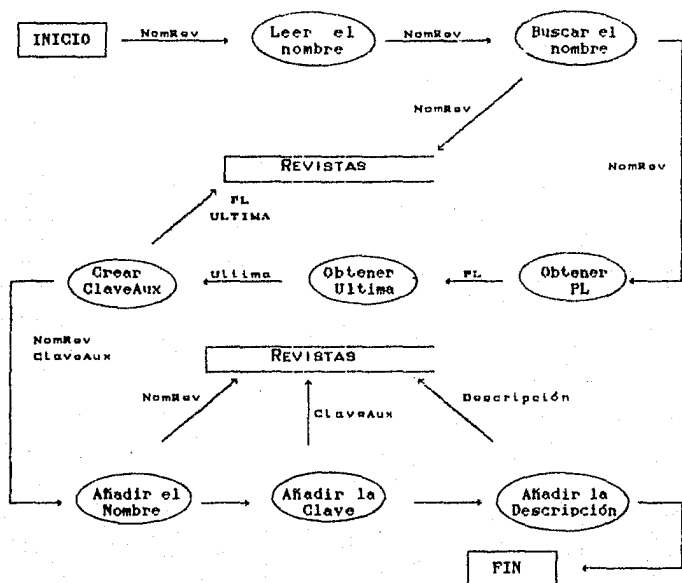
FLUJO DE DATOS DE ALTASR



FLUJO DE DATOS DE ALTASR ...



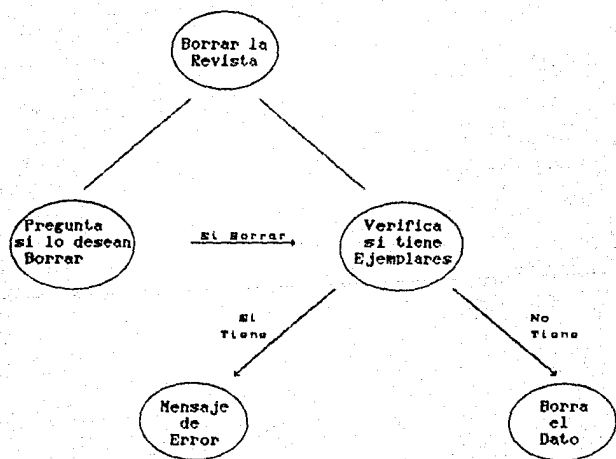
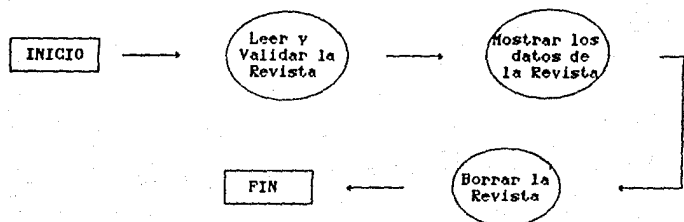
FLUJO DE DATOS DE ALTAAR ...



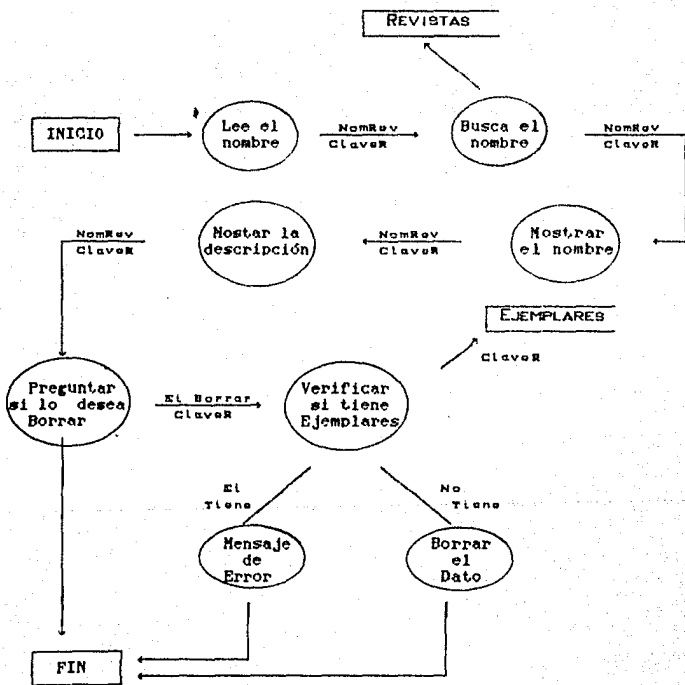
FLUJO DE DATOS DE BAJASR



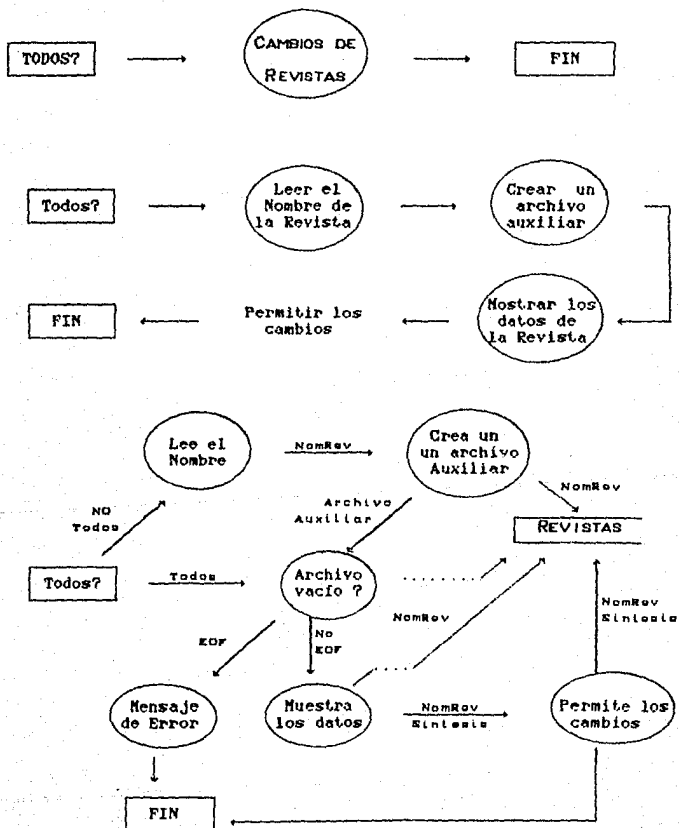
FLUJO DE DATOS DE BAJASR



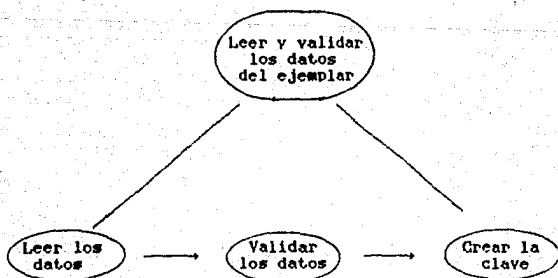
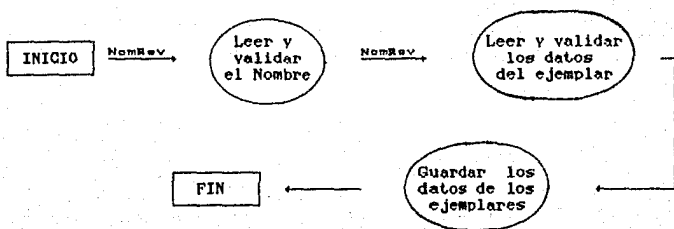
FLUJO DE DATOS DE BAJASR



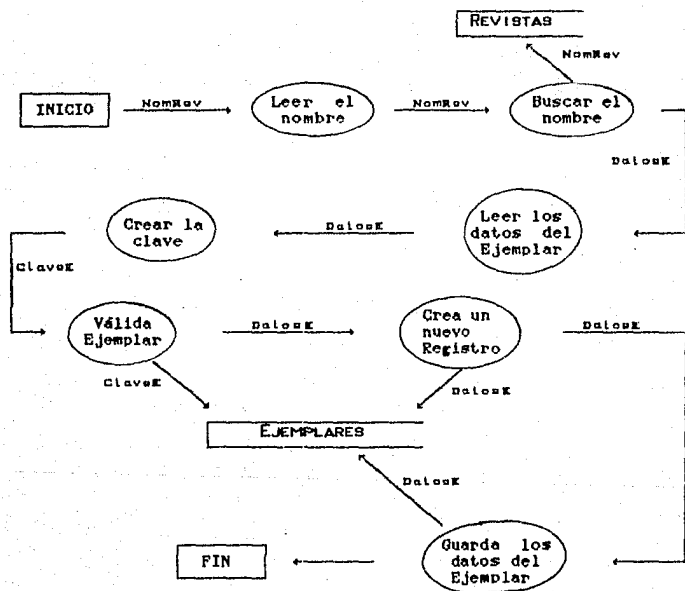
FLUJO DE DATOS DE CAMBIAR



FLUJO DE DATOS DE ALTAS E

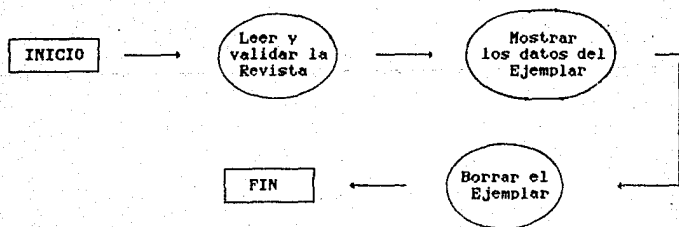


FLUJO DE DATOS DE ALTASÉ

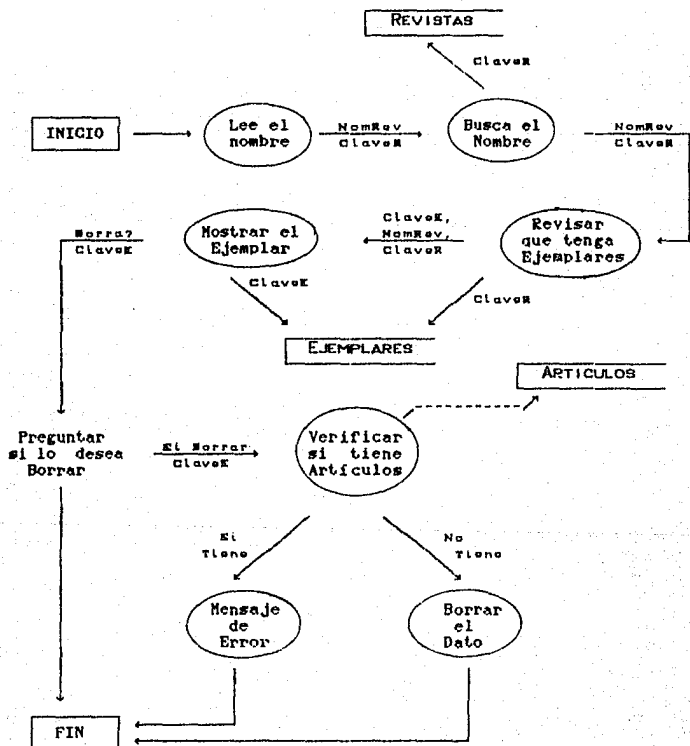


DatosK: NumK, VolK, MesK, AñoK

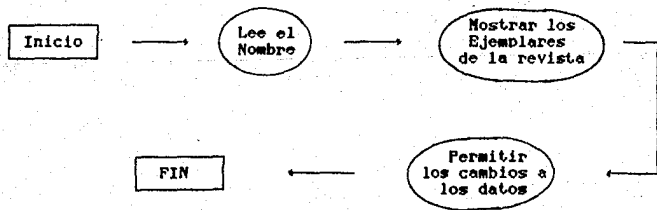
FLUJO DE DATOS DE BAJAS E



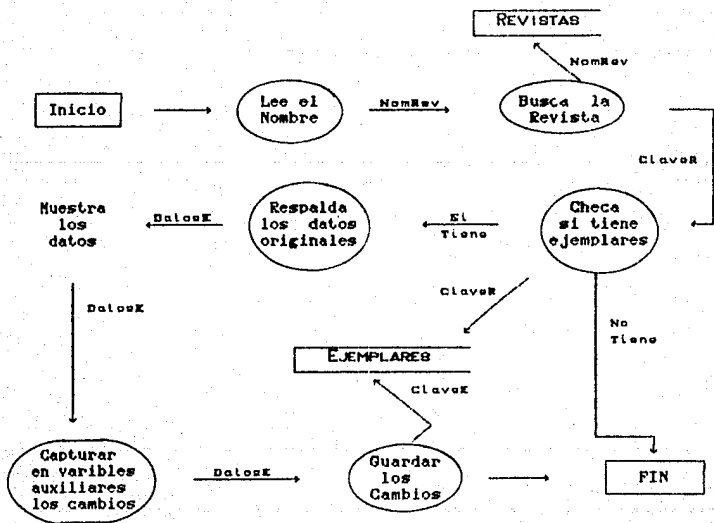
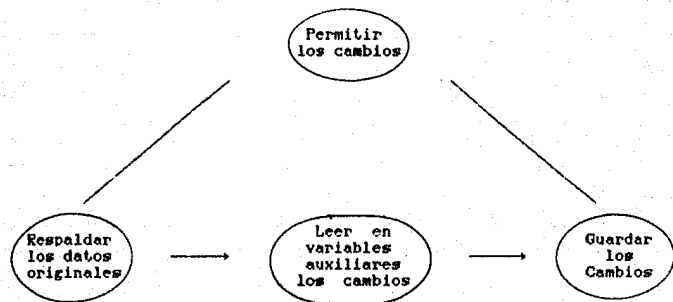
FLUJO DE DATOS DE BAJAS E



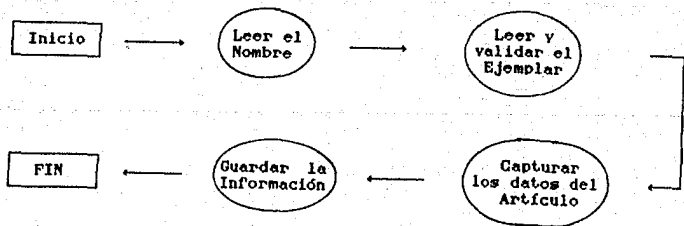
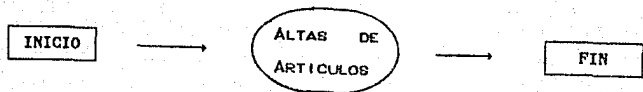
FLUJO DE DATOS DE CAMBIOS E



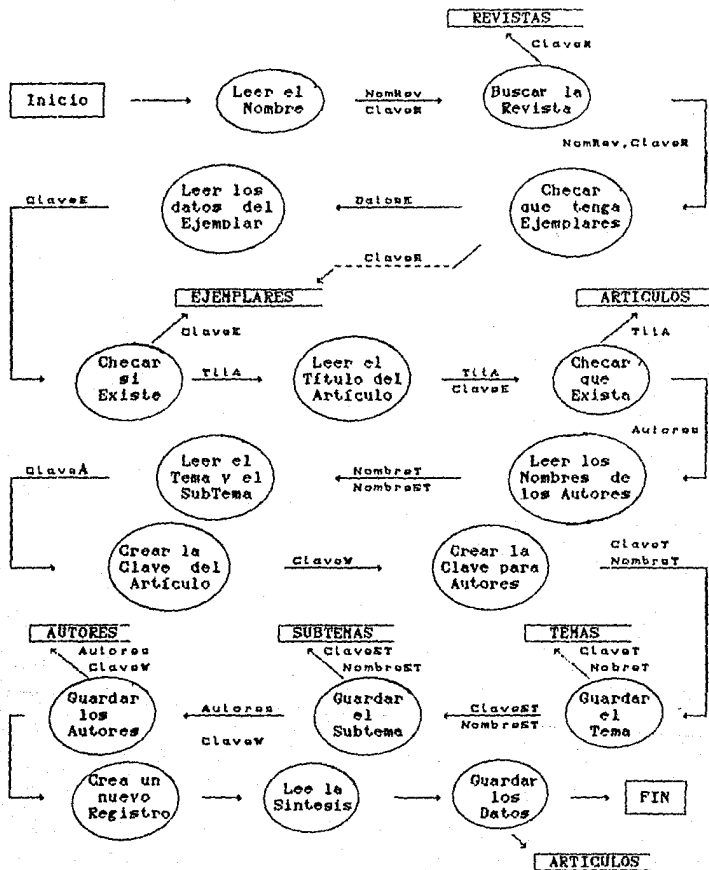
FLUJO DE DATOS DE CAMBIOS E



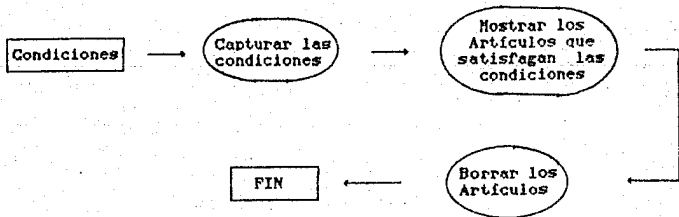
FLUJO DE DATOS DE ALTASA



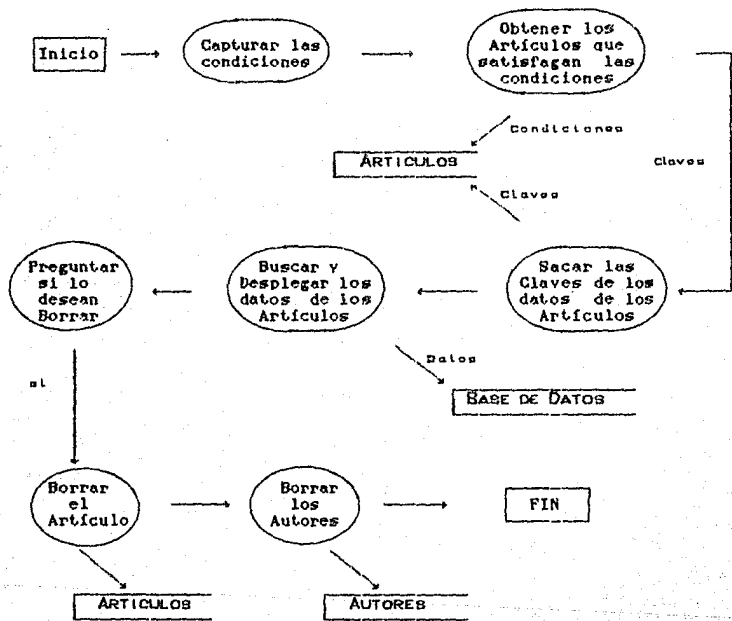
FLUJO DE DATOS DE ALTASA



FLUJO DE DATOS DE BAJASA



FLUJO DE DATOS PARA LA BAJA DE ARTICULOS



A P E N D I C E C

CAJAS NEGRAS

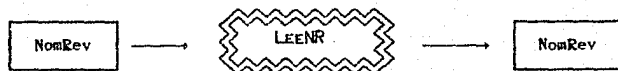
CAJAS NEGRAS

LEENR:

Entradas: Nombre de la Revista.

Proceso: Pedirá el Nombre de la Revista y capturará información.

Salidas: Nombre de la Revista.



ESTAREV:

Entradas: Nombre de la Revista, Clave de la Revista, Cadena.

Proceso: Buscará el Nombre de la Revista, en el archivo de datos y mandará un mensaje según la cadena dada y el resultado obtenido. Además, si existe la revista, asignará a la clave de la revista el valor correspondiente.

Salidas: Mensaje, Clave de la Revista y Valor Lógico.



IMPLISTA:

Entradas: No tiene.

Proceso: Revisará que la impresora esté lista para recibir los datos a imprimir. Regresará un valor lógico y mandará un mensaje dependiendo de la situación en que ésta se encuentre.

Salidas: Mensaje y Valor Lógico.



ESTAEJEM:

Entradas: Clave del Ejemplar.

Proceso: Verificará si existe o no un ejemplar cuya clave sea Clave del Ejemplar y regresará un mensaje según el resultado obtenido.

Salidas: Mensaje y Valor Lógico.

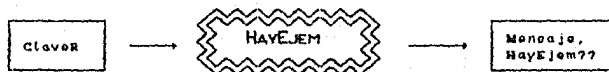


HAYEJEM:

Entradas: Clave de la Revista

Proceso: Revisará si hay ejemplares de la revista cuya clave sea Clave de la Revista. Regresará un valor lógico y mandará un mensaje dependiendo de la situación en que ésta se encuentre.

Salidas: Mensaje y Valor Lógico.



GUARDAT:

Entradas: Nombre y Clave del Tema.

Proceso: Verificará si el Nombre del Tema ya existe en el archivo de datos. Regresará la Clave del Tema, si existe, en caso contrario, además, creará la clave para el nuevo tema y guardará estos datos el Archivo de Temas.

Salidas: Clave del Tema.



GUARDAST:

Entradas: Nombre y Clave del Subtema.

Proceso: Será un proceso similar al anterior, pero trabajará archivo de Subtemas.

Salidas: Clave del Subtema.



PONATS:

Entradas: Clave de Autor, Tema y Subtema.

Proceso: Buscará y mostrará en pantalla los datos de los escritores, Tema y Subtema de un artículo.

Salidas: Datos en pantalla.



PONREVEJEM:

Entradas: Clave de Revista y Ejemplar.

Proceso: Buscará y mostrará en pantalla los datos de la revista y del ejemplar.

Salidas: Datos en pantalla.



A P E N D I C E D

**EJEMPLOS DE
MODIFICACIONES**

EJEMPLO DE MODIFICACION AL SISTEMA SCAR

En el presente apéndice se describirá un ejemplo de cómo se modificaría el desarrollo del sistema SCAR, descrito en el capítulo V, si se añade el siguiente requerimiento: Se desea llevar un control de información acerca de las personas a quienes se les prestan los artículos. Este requerimiento implica ampliar considerablemente el diseño del sistema.

MODIFICACIONES:

1. ANALISIS DEL REQUERIMIENTO.

Si el artículo está prestado, se debe tener la información de la persona y acceso a manipular tal información. A una persona pueden prestarse varios artículos, cuando esa persona regrese los artículos, sus datos serán eliminados de la base de datos.

Se necesitan funciones que realicen las actualizaciones y consultas de la información acerca de las personas a quienes se les prestaron artículos.

2. DISEÑO.

a) Diseño de la Base de Datos

La información que se tendrá de cada persona será: Nombre, Teléfono y Dirección particular, Datos del artículo que le fue prestado y algunas observaciones adicionales.

El nuevo archivo para la Base de Datos, podría estar dado por el registro *Personas*, el cual es ilustrado en la figura D-1.

Registro: *Personas.*

Núm	Nombre del Campo	Descripción del Campo
01	Nombre	Nombre de la Persona.
02	Teléfono	Teléfono particular de la Persona.
03	Dirección	Dirección particular de la Persona.
04	Artículo	Datos del artículo que le fué prestado.
05	Observaciones	Observaciones Adicionales.

Figura D-1

Ahora bien, como existe la posibilidad de que a una persona se le presten varios artículos, para evitar la redundancia de datos, serán separados los datos de la persona de los del artículo en archivos diferentes, los cuales estarán conectados por una clave que será dada por el Registro Federal de Causantes. Basta tener la clave del artículo para identificar el artículo que ha sido prestado. La figura D-2 ilustra la estructura de los registros que formarán los nuevos archivos para la Base de datos.

Registro: *Personas.*

Núm	Nombre del Campo	Descripción del Campo
01	Nombre	Nombre de la Persona.
02	RFC	Registro Federal de Causantes.
03	Teléfono	Teléfono particular de la Persona.
04	Dirección	Dirección particular de la Persona.
05	Observaciones	Observaciones Adicionales.

Registro: *Adeudos.*

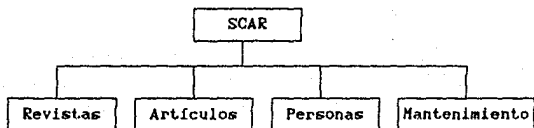
Núm	Nombre del Campo	Descripción del Campo
01	RFC	Registro Federal de Causantes.
02	Artículo	Teléfono particular de la Persona.

Figura D-2

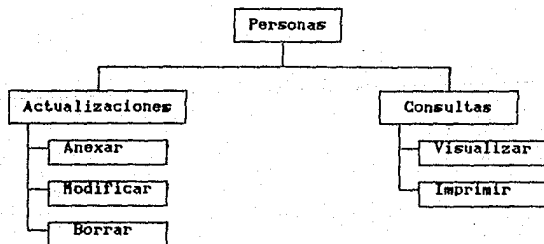
Finalmente, la Base de datos estará formada por los Archivos: *Revistas, Ejemplares, Artículos, Autores, Temas, Sistemas, Personas y Adeudos.*

b) Modularización.

Se tendrá un nuevo módulo para manejar los datos personales. El archivo de *Adeudo* será manipulado internamente por el sistema. La nueva estructura jerárquica del sistema SCAR será:



El diagrama jerárquico para el módulo de personas está ilustrado en la siguiente figura.



c) Descripción de los procesos.

Anexar: Añadir los datos personales de un nuevo individuo.

borrar: Permite borrar la información de una persona, siempre y cuando no "deba" Artículos.

Modificar: Permitirá hacer modificaciones al archivo de personas.

Visualizar: Dado el nombre de un individuo, mostrará los datos personales y en caso de que "deba" artículos, se mostrará la información de tales.

Imprimir: Similar al anterior, únicamente que la información será direccionada a impresora.

3. MODIFICACIONES AL MODULO DE ARTICULOS.

Los cambios que sufrirán los procesos del módulo de artículos son descritos a continuación.

Anexar y Modificar Artículos: En el proceso de altas el campo de *Prestado* es dado de antemano como FALSO, si se modifica indicará que el artículo sí está prestado, entonces, se preguntará al usuario quien es el deudor. Una vez dado el nombre y el registro federal de causantes del deudor, será creado un nuevo registro para el archivo de *Addeudo* en el cual se guardará la información correspondiente. Buscará los datos personales de tal deudor y si no existe, se pedirá la información necesaria para incluirla en el archivo de *Personas*.

Si al realizar una modificación se indica que el archivo ha sido devuelto, es decir, se cambia el campo *Prestado* de VERDADERO a FALSO, entonces se eliminará el registro respectivo en el archivo de *Addeudo*.

Borrar Artículos: No será permitido borrar un Artículo que esté prestado.

Consultar Artículos: Si el artículo está prestado mostrará el nombre del Deudor. De hecho esto lo hará cada vez que se desplieguen los datos de un artículo sin importar el proceso.

Los siguientes diagramas complementan los diagramas de Warnier para los procesos anteriores. Fue_Prestado? será anexado al final del dígrama de altas (cambios) de un artículo; Fue_Devuelto? se incluirá en la parte final del diagrama de cambios de artículos y Prestado? será incluido en el proceso de mostrar los datos.

Fue_Prestado?
(0,1)

{ Preguntar el Nombre del Deudor.
 Buscar los Datos del Deudor.
 No Existe el Deudor (0,1) { Preguntar Datos Personales.
 Crear el Registro para Personas.
 Guardar los Datos.
 Crear un Registro para Adeudos.
 Guardar los Datos del Deudor.

Fue_Devuelto?
(0,1)

{ Busca en el Archivo de Adeudos la Clave del Artículo.
 Borra el Registro.

Fue_Prestado?
(0,1)

{ Buscar en el Archivo de Adeudos la Clave del Artículo.
 Sacar el Registro Federal de Causantes del Deudor.
 Buscar en el Archivo de Personas al Deudor.
 Sacar el Nombre del Deudor.
 Mostrar el Nombre del Deudor.