



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES.

CAMPUS ARAGÓN

**“SISTEMA PARA LA AUTOMATIZACIÓN DE LA BIBLIOTECA
DEL CENTRO DE ESTUDIOS TECNOLÓGICOS Industrial y de
Servicios No. 92”.**

2001

T E S I S

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

P R E S E N T A :
J. MARTÍN ALBA MARTÍNEZ

ASESOR :
ING. LUIS LORENZO JIMÉNEZ GARCÍA





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

DEDICATORIA

Para el desarrollo profesional de toda persona se necesita el apoyo integral de familiares, profesores, amigos, compañeros de la escuela y del trabajo, tratar de nombrar a las personas más importantes sería imposible ya que sin la ayuda de alguna de ellas, no habría concluido la carrera, por consiguiente, un sincero reconocimiento para todas estas personas.

C O N T E N I D O

INTRODUCCION	I, II, III
1. DISEÑO DE BASES DE DATOS	
1.1. Descripción de los datos	3
1.2. Tipos de archivos	5
1.3. Modelos de bases de datos	6
1.3.1. Modelo relacional	6
1.3.2. Modelo jerárquico	7
1.3.3. Modelo de red	8
1.4. Organización de los archivos	9
1.4.1. Organización secuencial	9
1.4.2. Organización de acceso directo	10
1.4.3. Organización secuencial indexada	14
2. ANALISIS DEL SISTEMA ACTUAL	
2.1. Definición de sistema	16
2.2. Clasificación de los sistemas de información	18
2.3. Técnicas estructuradas	19
2.3.1. Programación estructurada	19
2.3.2. Diseño estructurado	21

Contenido

2.3.3. Modelización de datos	22
2.3.4. Ingeniería de información	23
2.3.5. Técnicas de prototipos y desarrollo rápido	23
2.4. Herramientas CASE	25
2.5. Principales razones para la construcción de un sistema	26
2.6. Ciclo de vida del desarrollo de sistemas	29
2.7. Técnicas para identificar requerimientos	31
2.8. Servicios que presta la biblioteca	32
2.9. Sistema de clasificación de libros	36
2.9.1. Sistema de clasificación Dewey	36
2.10. Adquisición de libros	38
2.11. Problemas identificados	39
2.12. Identificación de las necesidades de los usuarios	41
2.13. Justificación del sistema	43
2.14. Recursos disponibles	44
3. DISEÑO GENERAL DEL SISTEMA AUTOMATIZADO	
3.1. Selección del proyecto	46
3.2. Opciones del sistema	48
3.3. Diseño de la entrada y salida de datos	49
3.4. Diseño de los controles	54
3.5. Diseño de la base de datos	56

4. DISEÑO DETALLADO DEL SISTEMA AUTOMATIZADO

4.1. Características de las pantallas	61
4.2. Entrada de datos	64
4.3. Diseño de los programas	68
4.4. Salida de datos	69
4.5. Diseño de archivos	73

5. IMPLANTACION DEL SISTEMA

5.1. Capacitación del personal	79
5.2. Construcción y prueba de los programas	80
5.3. Pruebas al sistema	85
5.3.1. Prueba de entrada y salida	85
5.3.2. Prueba de la base de datos	86
5.3.3. Prueba de los controles	86
5.3.4. Prueba de los programas	86
5.4. Conversión del sistema	86
5.5. Soporte de sistemas	87

ANEXO 1	Programas fuentes del sistema	91
ANEXO 2	Tablas Cutter	107
CONCLUSION		109
BIBLIOGRAFIA		110

INTRODUCCION

El cálculo de la nomina, generar las estadísticas, la creación de horarios de grupos y el historial académico, son ejemplos de tareas tediosas y laboriosas que consumen demasiado tiempo por el alto volumen de información que es necesario procesar. Cuando estas son realizadas por personas por lo general contienen errores de omisión o de calculo incluso de proyección.

El servicio de biblioteca es uno de los más importantes en las empresas dedicadas a la educación. El saber cantidad, tipo y los libros que hacen falta es información primordial para emprender acciones en beneficio al mejoramiento del servicio. Este tipo de labores se tienen que automatizar para aumentar la productividad y lograr los objetivos de la organización. Se cuenta con la tecnología de computadoras y comunicaciones, que desde años anteriores se usa para ejecutar los trabajos rutinarios, peligrosos y para crear efectos especiales.

El propósito principal de este trabajo es que la biblioteca cuente con un sistema que permita, no solo realizar rápido las tareas que consumen más tiempo sino que además, hacer que estas se lleven a cabo de forma más eficiente y dinámica. Con la ayuda del sistema que se desea implantar, el encargado desempeñará sus funciones con eficiencia y en consecuencia hará sencillo el préstamo de libros, se tendrá un mejor control sobre los libros, en general el servicio y la administración será más ágil y eficiente.

La Dirección General de Educación Tecnológica Industrial (DGETI), es un organismo público del gobierno federal, imparte educación de nivel medio superior de carácter propedeúico y terminal en sistema escolarizado o sistema abierto. Cuenta con la modalidad de:

- Técnico profesional: esta modalidad está integrada por un área de formación general básica que permite al educando la comprensión de la naturaleza, la sociedad y el hombre, y un área tecnológica que permite colocar a los egresados en aptitud de incorporarse directamente al trabajo profesional.
- Bachillerato Básico: curricularmente está formado por un área de tronco común que permite a los egresados incorporarse a estudios superiores y al trabajo en el área tecnológica en calidad de técnico.

Actualmente la DGETI cuenta con un total de 662 planteles en toda la República Mexicana: 234 Centros de Estudios Científicos y Tecnológicos de los Estados (CECYTES), 261 Centros de Bachillerato Tecnológico Industrial y de Servicios (CBETIS) y 167 Centros de Estudios Tecnológicos Industrial y de Servicios (CETIS), organizados por 31 Coordinaciones Estatales y una Dirección ubicada en el Distrito Federal, esta cantidad de escuelas hacen posible que en México el 80% de los jóvenes que estudian a nivel bachillerato, lo hacen en alguna escuela de este organismo.

El CETIS número 92 es una escuela cuyo objetivo es formar bachilleres técnicos. pertenece a la Coordinación Estatal del Estado de México, cuenta con una población estudiantil de más de 900 alumnos divididos en dos carreras, Informática Administrativa y Computación.

Cuando existe una gran concentración de escuelas y además administradas por un organismo público, se presentan problemas complejos que afectan a las actividades de las escuelas. Un problema grave es la forma arbitraria en que se lleva a cabo la distribución de los recursos, originando principalmente por grupos de poder que se alternan la administración de la DGETI.

Por otro lado, la escuela enfrenta el aumento de población estudiantil y el costo de la educación cada vez más demandante en todos sus aspectos. También se encuentra en la necesidad de ofrecer servicios con calidad para ayudar a los alumnos a adquirir conocimientos adecuados, de esta manera ellos tendrán mayor oportunidad de continuar sus estudios o bien incursionar en el terreno laboral.

Cuando un maestro se jubila o pide permiso temporal, estas plazas generalmente son utilizadas por la Coordinación Estatal, dejando a la escuela sin este recurso, el cual es necesario ya sea para funciones de docencia, asesoría o cargo administrativo. Esta situación hace necesario que el uso de los recursos de la escuela sean eficientemente utilizados para cumplir con los objetivos de docencia y administrativos.

El equipo de cómputo constituye un recurso importante, ya que cuando se le utiliza de manera eficiente puede contribuir al ahorro de otros recursos por un lado y por otro ayuda a que el trabajo rutinario sea más eficaz.

Se cuenta con 40 computadoras 80386 y 80486 divididas en dos laboratorios, al llegar equipo nuevo éste es desplazado, debido a que los programas que se enseñan requieren de equipo moderno. Este deberá estar en buen estado para instalar en ellos programas adecuados para que puedan ser útiles.

Una de las tareas del ingeniero en computación es utilizar eficientemente el equipo de computó con que cuenta la empresa. Así que se tiene la responsabilidad de analizar y determinar si un equipo es funcional, y si así es, darle el debido uso. Las computadoras disponibles, tienen capacidad de procesar, el volumen de información que se maneja en este tipo de organizaciones, siempre y cuando se utilicen los programas adecuados.

Primero fue necesario hacer la conexión en red para los equipos del plantel, esta etapa se realizó con la instalación de un servidor de Novell 312, la instalación del programa Office 4.2 en red y se inició la creación de programas para estos equipos. Uno de estos programas es el que corresponde al Sistema para la Automatización de la Biblioteca del CETIS 92 que presento como trabajo de tesis.

Para la construcción del sistema de información se utiliza una metodología conocida como ciclo de vida del sistema o ciclo de desarrollo de los sistemas. No hay un número exacto de etapas que conforman el ciclo, sin embargo, es importante un enfoque sistemático. Para nuestro propósito lo dividimos en cinco etapas: Planificación de sistemas, Análisis del sistema, Diseño general del sistema, Diseño detallado del sistema e Implantación, en cada una de las etapas se detalla el trabajo que se debe realizar, después se muestran ejemplos relacionados con el sistema propuesto.

Es conveniente tener conocimientos de diseño, tipos y modelos de bases de datos, ya que el sistema las utilizará en todos los procesos, por lo tanto en el primer capítulo se estudian las bases de datos, ya que es el componente estructural más importante de los sistemas de información.

En el segundo capítulo se estudia el sistema que se está utilizando actualmente. Se explica teóricamente desde la planificación del sistema hasta la implantación, es decir, se reúne información que permite conocer las necesidades o requerimientos de los usuarios, los datos que va a emplear el sistema y herramientas para obtener dichos datos.

El diseño general del sistema es el tercer capítulo, aquí se estudia de manera general el diseño de bases de datos, de controles, de entradas y salidas, utilizando ejemplos del sistema a desarrollar, se encuentran los detalles que establecen la forma en la que el sistema debe satisfacer los requerimientos identificados durante la fase del análisis. Se analizan los puntos que determinan la continuación o cancelación del proyecto.

En el cuarto capítulo se describe completamente el formato y contenido de la salida del sistema. Las entidades de los datos se explican de acuerdo a sus atributos, valores y presentaciones. En otras palabras, se preparan especificaciones precisas para la fase de implementación, que es el quinto capítulo de este trabajo, en donde se verifica e instala el equipo y la aplicación, se entrena a los usuarios y se construyen todos los archivos necesarios para el funcionamiento del sistema.

1. DISEÑO DE BASES DE DATOS

Descripción de los datos

Tipos de archivos

Modelos de las bases de datos

Organización de los archivos

I. DISEÑO DE BASES DE DATOS

Los sistemas de información crean, mantienen y usan datos. Estos se guardan en archivos y bases de datos. En la figura 1.1.1 se observa la diferencia fundamental que existe. El entorno basado en archivos, dan más importancia al sistema. Conforme se desarrollan las aplicaciones, se construyen archivos adaptados a ellas. Por lo general, estos archivos no sirven para dar respuesta a las necesidades futuras. En el entorno de bases de datos, figura 1.1.2, las aplicaciones se construyen alrededor de la base de datos integrada, es decir, ésta no depende de las aplicaciones que harán uso de ella.

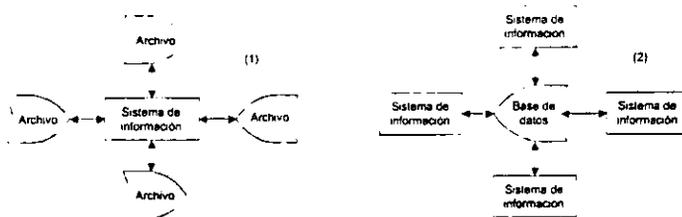


Figura 1.1 Entorno basado en archivos y base de datos.

Los archivos convencionales son relativamente fácil de diseñar e implantar, ya que se basan normalmente en una aplicación o un sistema de información único. Cuando se trata de manejar grandes volúmenes de transacciones, estos archivos tienen buen desempeño en el procesamiento. Sin embargo, se tienen inconvenientes; con el tiempo, el desarrollo múltiples aplicaciones se traduce en duplicación de datos en los archivos. Otra desventaja es la falta de flexibilidad, cuando se plantean nuevas necesidades, como pueden ser informes y consultas, requieren que se vuelvan a diseñar los archivos, ya que es difícil que los archivos existentes respondan bien a estos requerimientos. Si fuera necesario reestructurar los archivos, habría que volver a escribir todos los programas que hicieran uso de ellos.

Las bases de datos tienen capacidad de compartir y disminuir la duplicación de los datos. Hay un mal entendido con respecto a su uso, al creer que se puede construir solo una base de datos que contenga todos los datos de interés para la empresa. Esta noción, aunque deseable, no resulta práctica, ya que se llevaría años en construir una base tan completa. En la práctica, se utilizan varias bases con datos compartidos entre sistemas de información. Esto implica que habrá redundancia, mínima y controlada, en ellas. Gracias a que las bases de datos están definidas de modo independiente a los programas, se podrán usar los datos en formas no especificadas originalmente por los usuarios, en otras palabras, se tiene más flexibilidad.

Por otro lado, la tecnología de bases de datos es más compleja, requiere de un conjunto de programas especiales, llamado sistema de gestión de base de datos. La flexibilidad proporcionada por este sistema hace que el sistema funcione más lento, por lo tanto, se requieren computadoras con mayor velocidad de procesamiento. El costo del desarrollo aumenta, ya que se debe adquirir, aprender y dar mantenimiento al sistema. También, para

¹ Fuente: Análisis y diseño de sistemas de información Jeffrey L. Whitten.

lograr la independencia de los datos en las bases, se deben respetar los principios de diseño. A pesar de estas dificultades, el uso del entorno de bases de datos está aumentado gracias al mejoramiento continuo de la tecnología y, los métodos y las herramientas de diseño.

La base de datos se define como una colección de datos interrelacionados almacenados en conjunto sin redundancias innecesarias; su finalidad es la de servir a una aplicación o más, los datos se almacenan de modo que resulten independientes de los programas que los usan. La idea básica en la implantación de una base de datos se refiere a que los mismos datos deben ser aprovechados por todas las aplicaciones como sea posible. Sin embargo existen datos duplicados o redundantes, los cuales se encuentran simultáneamente almacenados en varios volúmenes con distinta finalidad y diferente fecha de actualización. En las bases de datos se admite cierta redundancia con el objeto de reducir los tiempos de acceso o simplificar los métodos de direccionamiento. Algunos registros se duplican para facilitar la reconstrucción de la base en caso de daño. Así que, existe la necesidad de controlar el grado de redundancia, de modo que es deseable hablar de redundancia mínima en lugar de no redundancia como criterio de diseño.

La independencia de los datos es uno de los más valiosos argumentos a favor de las bases de datos. Esta idea implica que los datos y los programas de aplicación son independientes, de manera que unos y otros pueden ser modificados sin ser afectados.

Cuando un determinado conjunto de datos sirve a una variedad de programas de aplicación, cada uno de éstos percibe diferentes relaciones entre aquéllos, es decir pueden existir múltiples interconexiones entre los datos. En estas condiciones el conocer donde se almacenan los datos y su seguridad, se presentan problemas más complejos que los que se encuentran en el caso de los archivos de datos no interconectados. La seguridad de los datos y la reconstrucción en caso de falla son aspectos de mayor importancia en el diseño de toda base de datos.

En las bases de datos dinámicas se llevan a cabo cuatro operaciones básicas, algunas aplicaciones utilizan todas las operaciones, pero ciertas aplicaciones lo harán más con un tipo que con otro.

1. La construcción de una base de datos incluye la obtención de los datos, su codificación y su captación.
2. La actualización de una base de datos permite colocar nuevos datos, modificar cuando sea necesario los valores de datos almacenados, y eliminar datos obsoletos o no válidos.

3. Recuperación de datos consiste en tomar un elemento específico para obtener un valor o en la recolección de una serie de elementos relacionados para obtener datos referentes a algunas relaciones que se manifiesten al unir los datos. Para tomar o traer un registro específico de datos se entrará a la base de datos mediante un argumento de búsqueda, el que se comparará con una llave en los registros de la base. Algunas veces el argumento de búsqueda se le conoce como llave de búsqueda.
4. La reducción de datos resulta necesaria cuando el volumen de datos importantes excede la capacidad de los solicitantes. En una base de datos estática este tipo de actividad tiende a ser dominante, mientras que en la base dinámica la reducción de datos se utiliza básicamente para resúmenes periódicos y análisis de tendencias. La reducción de datos es una actividad determinante en el desempeño de una base de datos.

Las características convenientes para los sistemas que almacenan datos son el rápido acceso para la recuperación, actualización y la economía de almacenamiento. El mantenimiento, la capacidad para representar estructuras de información del mundo real y la protección de la privacidad son otros criterios que también se deben tomar en cuenta en el diseño de las bases de datos. Para el éxito del sistema es vital que las capacidades proporcionadas por el sistema de archivos estén de acuerdo con las prioridades asignadas a los criterios, según lo determinen los objetivos para la base de datos.

1.1 Descripción de los datos

Los datos representan objetos físicos de la vida real. El bibliotecario frecuentemente necesita saber cuantos libros de un autor en particular existen. No sería práctico tener que ir hasta el lugar en donde se localizan y contar el número de libros cada vez que alguien preguntara. En lugar de esto, el empleado accesa los datos que representan los objetos.

Los objetos se denominan entidades. Una entidad puede ser un objeto tangible, como un empleado, un cliente o un artículo. Una entidad también puede ser un objeto intangible como un evento o un proyecto.

Una entidad tiene ciertos atributos que se desean registrar. Para un libro de la biblioteca se desea llevar un seguimiento de atributos tales como la clave del libro, autor del libro, título del libro, etc. Cada atributo tiene un valor que podría asociarse con él y una representación física de los datos de dicho valor del atributo. La figura 1.2 presenta un ejemplo de dos entidades usadas en el proyecto, además se observa cómo los atributos de datos se aplican en entidades reales.

ENTIDAD	ATRIBUTO DE DATO	VALOR DEL ATRIBUTO	REPRESENTACION DE DATOS
Libro	Clave	384001	6 dígitos
	Título	Programación en Clipper	40 caracteres
	Autor	Ramalho	40 caracteres
Usuario	Nombre	Erminio Alba Martinez	40 caracteres
	Dirección	Apaseo el Alto	30 caracteres
	Grupo	6 ^a	2 caracteres

*Figura 1.2. Entidades y atributos de datos.

Se deben identificar todos los atributos de datos para una entidad, los cuales se necesitarán para proporcionar información a los usuarios. Las relaciones son asociaciones entre entidades, se tienen tres diferentes tipos:

- Una a una. Una entidad en A está asociada únicamente con una entidad en B, y una entidad en B está asociada sólo con una entidad en A. Figura 1.3.1.
- Una a muchas. Una entidad en A está relacionada con cualquier número de entidades en B, pero una entidad en B puede asociarse únicamente con una entidad en A. Figura 1.3.2.
- Muchas a una. Una entidad en A está asociada únicamente con una entidad en B, pero una entidad en B está relacionada con cualquier número de entidades en A. Figura 1.3.3.

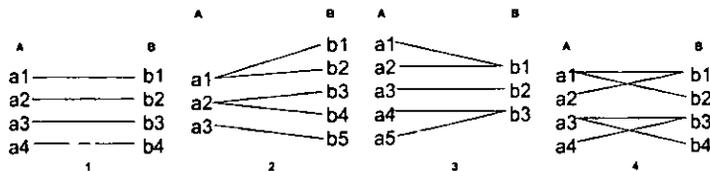


Figura 1.3 Tipos de relaciones.

- Muchas a muchas. Una entidad en A está asociada con cualquier número de entidades en B, y una entidad en B está vinculada con cualquier número de entidades en A. Figura 1.3.4.

La forma en que los datos se almacenan en una base de datos, es en registros y campos. Un registro es la descripción de una sola persona, cosa o actividad; el campo es una parte de la descripción almacenada en el registro, por otro lado la actualización de una base de datos se entiende como la adición de nuevos datos, tener capacidad para modificar los datos almacenados y eliminar datos no válidos.

* Fuente: propia.

¹ Fuente: Fundamento de bases de datos, Henry F. Korth.

1.2 Tipos de archivos

Las empresas y organizaciones utilizan ampliamente los archivos maestros, de transacciones, archivos de datos no activos y archivos temporales. Todos los archivos interactúan en el procesamiento de la información como se ve en la figura 1.4.

Un archivo maestro es un conjunto de registros acerca de un aspecto importante de las actividades de una organización. Puede contener datos que describan el estado actual de eventos específicos o indicadores. Los archivos maestros son útiles sólo mientras se mantengan exactos y actualizados. En otras palabras, antes de que los archivos puedan utilizarse, deben ser mantenidos para reflejar los eventos más recientes que afectan en ellos. Los campos de un archivo maestro ordinario, generalmente permanecen constantes. Sin embargo el contenido de los campos puede cambiar constantemente, por lo tanto estos archivos se actualizan constantemente utilizando los archivos de transacciones.

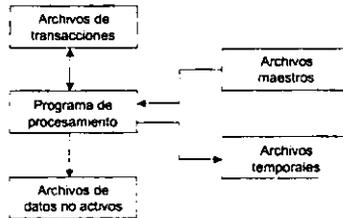


Figura 1.4 Relación de archivos y los programas.

Un archivo de transacciones tiene dos propósitos: acumular datos acerca de los eventos al momento que ocurran y actualizar los archivos maestros para reflejar los resultados de las transacciones actuales. El término transacción se refiere a cualquier evento que afecte a la organización y sobre el cual se calculan datos. Los archivos de transacciones pueden retenerse por meses, a veces incluso por años, después de que han sido creados, dependiendo de las necesidades de la organización. Por ejemplo, un registro de un libro prestado tiene utilidad, hasta que el libro es devuelto, o es sustituido, en caso de extravío por parte del usuario, este registro se conserva hasta que finaliza el año escolar, al inicio de cada año no debe haber libros fuera de la biblioteca, en consecuencia la base donde se hayan almacenados los registros de libros prestados estará vacía.

Los archivos de datos no activos. Después de un tiempo de usar los archivos de transacciones y maestros, algunos registros de estos archivos, dejan de ser requeridos o el acceso es poco frecuente, estos no se borran; más bien se incluyen en los archivos de datos no activos, posteriormente serán utilizados para un análisis o una auditoría.

Los archivos temporales, llamados también archivos de trabajo contienen réplicas, subconjuntos temporales o formas alternativas de ordenación de un archivo maestro o de transacciones. Generalmente, este tipo de archivos se crean, para ser utilizados por el

¹ Fuente: Diseño general de sistemas de información John G. Burch.

programa informático apropiado y a continuación se destruyen, es decir, se crean para su uso en una sola tarea, y deben volverse a generar cada vez que se efectúe la tarea. Los archivos de reportes son archivos temporales que se utilizan cuando el tiempo de impresión no está disponible para todos los archivos producidos, situación que surge con frecuencia en el procesamiento sobrepuesto. El procesamiento sobrepuesto es la capacidad de una computadora para llevar a cabo en forma simultánea entrada, procesamiento y salida.

La computadora escribe el reporte a un archivo, lo guarda en disco donde permanece hasta que pueda imprimirse. Este proceso se conoce como impresión por cola, esto quiere decir que la salida que no puede imprimirse cuando se produce forma una cola en un archivo de reportes. Después, en pocos minutos, dependiendo del volumen de trabajo y velocidad de las impresoras, se dará la instrucción al sistema para que lea el archivo de reportes e imprima la salida.

1.3 Modelos de bases de datos

Cuando se introduce información en una base de datos, dicha información se almacena en disco, en uno o más archivos. Esto se denomina almacenamiento físico de la información. Los programas que actualizan y manipulan el contenido de la base de datos manejan todos los aspectos del almacenamiento físico.

El almacenamiento lógico se refiere a la forma en que se puede relacionar la información de diferentes archivos. La disposición lógica de los archivos se denomina modelo o esquema de la base de datos. Los tres modelos, típicos son: el de redes, el jerárquico y relacional, los cuales se describen en los subtemas siguientes. Casi todos los programas de administración de bases de datos diseñados para microcomputadoras utilizan el modelo relacional.

1.3.1 Modelo relacional

El modelo relacional de datos se basa en una relación: una tabla bidimensional. Los renglones de la tabla representan los registros y las columnas muestran los atributos de la entidad. La figura 1.5 muestra tres registros con clave, título, autor y editorial como atributos de la entidad libro. El orden de los datos en la tabla no es significativo y tampoco implica un orden cuando los registros están incluidos en la relación. Análogamente los detalles físicos de almacenamiento no son de interés. Las tablas relacionales solo muestran las relaciones lógicas no físicas.

El modelo relacional es capaz de mostrar relaciones complicadas de datos. Sin embargo en comparación con el modelo de red, es fácil de entender.

CLAVE	TITULO DEL LIBRO	AUTOR	EDITORIAL
384075	Análisis y diseño de sistemas	Janes A. Senn	Mc Graw Hill
384094	Clipper 5.01 avanzado	José A. Ramalho	Mc Graw Hill
000027	Organización del CETIS	SEP	SEP

Figura 1.5 Registros y sus atributos.

1.3.2 Modelo jerárquico

El modelo jerárquico relaciona entidades por medio de una relación superior - subordinado o padre - hijo. En la figura 1.6 se muestra el modelo jerárquico de datos como un árbol volteado hacia arriba, el cual el nivel más alto se conoce como raíz. Los nodos del árbol representan las entidades.

Un modelo jerárquico de datos permite dos tipos de relación: uno a uno, una entidad en un nivel se relaciona con una entidad en el siguiente nivel. El segundo tipo de relación es uno a muchos, una entidad en un nivel se relaciona con una, muchas o ninguna entidad del siguiente nivel. En un organigrama, esto significa que ningún empleado puede tener más de un supervisor, pero un supervisor puede tener un número ilimitado de empleados.

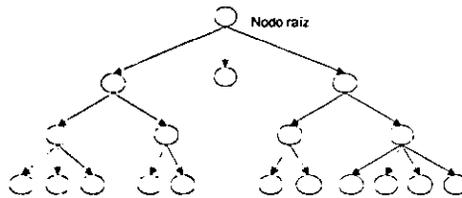


Figura 1.6

El diseño de una base de datos jerárquica afectará la facilidad de acceso. Por ejemplo, en la figura 1.7 muestra que los pedidos siempre están relacionados con un cliente específico.

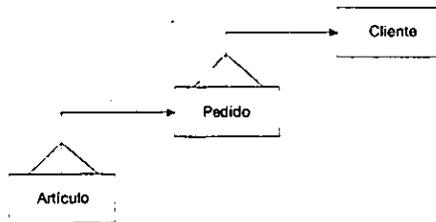


Figura 1.7

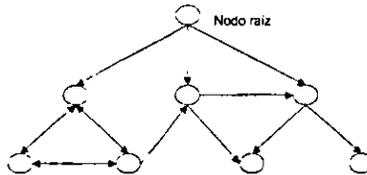
* Fuente: Análisis y diseño de sistemas de Información, Janes A. Senn.

Aparecen efectos colaterales indeseables bajo ciertos diseños de una base de datos. Las bases de datos jerárquicas involucran anomalías con respecto a lo siguiente:

- Inserción de registros: un registro dependiente no se puede añadir a una base de datos sin un padre. Por ejemplo, los artículos no se pueden añadir sin incluirlos en un pedido.
- Borrado de registros: al borrar un padre de la base de datos también se borran todos sus descendientes. Por ejemplo, al quitar un cliente también se quitan los pedidos pendientes.

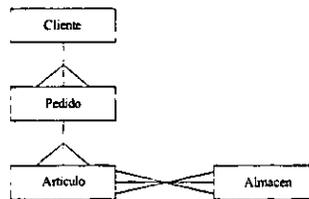
1.3.3 Modelo de red

El modelo de red es análogo al modelo jerárquico, excepto que una entidad puede tener más de un padre. Así, como se muestra en la figura 1.8, los miembros pueden pertenecer a más de una relación. Esta capacidad introduce el uso de un tipo adicional de relación entre los datos: muchos a muchos, una entidad se puede relacionar con una, muchas o ninguna entidad en otro nivel.



*Figura 1.8

La relación de red se muestra en la figura 1.9, donde la entidad artículo participa en más de una relación, es decir, un almacén guarda varios artículos y un artículo puede guardarse en muchos almacenes.



*Figura 1.9

* Fuente: Análisis y diseño de sistemas de información, James A. Sed.

En las bases de datos de tipo red, así como en las jerárquicas, se deben establecer las relaciones entre las entidades, al mismo tiempo que se establece el modelo de los datos y se crea la base de datos; en contraste con el modelo relacional, el cual no requiere rutas de acceso predefinido o relaciones entre las entidades.

Las bases de datos jerárquicas y de red son conceptualmente sencillas y parecen no ser complicadas a primera vista. Sin embargo, en un ambiente de una base de datos grande, puede evolucionar rápidamente hacia una telaraña complicada de interrelaciones que son difíciles de manejar al crecer la base de datos con el uso.

1.4 Organización de los archivos

Los registros se almacenan en archivos utilizando una organización de archivo la cual determina cómo se almacenan, localizan y recuperan los registros. El método secuencial y el directo están disponibles en todas las computadoras. El indexado, es posible sólo con software especial.

1.4.1 Organización secuencial

Cuando los registros se encuentran ordenados físicamente en un archivos, se dice que tal archivo es un archivo secuencial. Cuando se actualiza un archivo de este tipo, es necesario accederlo de manera completa, ya que los registros no pueden insertarse en la parte intermedia del archivo, un archivo secuencial generalmente se copia durante el proceso de la actualización.

La organización secuencial es la forma más simple de almacenar y recuperar registros en un archivo. En esta organización los registros se almacenan uno tras otro sin importar el valor real de los datos en los registros. El primer registro almacenado se coloca al principio del archivo. El segundo se almacena inmediatamente después del primero, el tercero después del segundo, etc. Este orden nunca cambia en la organización secuencial del archivo.

Para leer una archivo secuencial, el sistema siempre comienza al principio del archivo y lee un registro a la vez hasta llegar al registro deseado. Por ejemplo, para leer el décimo registro, el sistema comienza con el primer registro y lee hacia adelante un registro a la vez hasta llegar al décimo. No puede ir directamente al décimo registro sin partir del principio. Los archivos secuenciales no utilizan llaves del registro físico; los registros se accesan por su orden de aparición en el archivo.

Por ejemplo, si un archivo secuencial tiene las claves mostradas en la figura de 1.10 y queremos localizar y recuperar los datos del libro con la clave 384006. El registro 384001 es leído y se compara con la llave de búsqueda, es decir con 384006. Si el número de libro y la llave de búsqueda no coinciden, se lee el segundo registro y su número de libro se compara con la llave de búsqueda y si tampoco es el requerido se repite el proceso de lectura y comparación de registros hasta que el número de libro y la llave de búsqueda sean iguales. Si el archivo no contiene el libro con el número 384006, el proceso de lectura y comparación continúa hasta alcanzar el final del archivo.

CLAVE_A
384001
384002
384003
384004
384005
384006
384007
384008

*Figura 1.10

Los archivos secuenciales se utilizan cuando es necesario acceder cada registro en el archivo para una aplicación en particular. Si en promedio, alrededor de la mitad de los registros en el archivo se van a utilizar, la organización secuencial sigue siendo aceptable, ya que al menos uno de dos registros recuperados será utilizado y el tiempo necesario para examinar un registro adyacente es breve. Por otro lado, si el requerimiento es hallar un registro particular en un archivo muy grande, la organización secuencial del archivo se convierte en una desventaja.

1.4.2 Organización de acceso directo

La organización de acceso directo le pide al programa que diga al sistema dónde se almacena un registro antes de poderlo acceder. En contraste con la organización secuencial, el procesamiento de un archivo de acceso directo no requiere que el sistema comience en el primer registro.

Los archivos de acceso directo son archivos con llave. Asocian un registro con un valor llave específico y un lugar particular de almacenamiento. Todos los registros se almacenan mediante las llaves en las direcciones en lugar de por posición; si un programa conoce la llave del registro, se puede determinar la dirección de localización de un registro y recuperarlo en forma independiente de los demás registros del archivo.

• Fuente: Propia.

Siguiendo con el ejemplo de los libros, en la figura 1.11 se observa que en el acceso directo, los registros utilizan una área del almacenamiento que tiene un espacio reservado para cada número de libro. El sistema utiliza el número de libro como una llave de registro físico. El libro número 384004 se almacena en la dirección 384004, lugar reservado para el libro con ese número. Para recuperar ese libro del almacenamiento, se instruye al programa para que utilice el número 384004 como llave de búsqueda. Al conocer el programa que la llave sirve como la dirección, va directamente al lugar asignado para el registro con la llave de 384004 y recupera el registro.

El direccionamiento directo significa que la llave registro corresponde exactamente a una dirección de almacenamiento. Esta forma de organización requiere tener al conjunto de datos con las siguientes características:

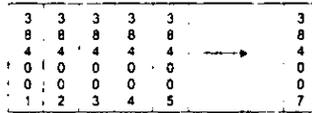


Figura 1.11.

- El conjunto de llaves tiene un orden denso y ascendente con pocos valores no utilizados. Por lo tanto, se desean pocos espacios abiertos entre valores de las llaves.
- Las llaves de los registros corresponden con los números de las direcciones de almacenamiento; existe una dirección de almacenamiento en el archivo para cada valor real o posible de la llave y no hay valores duplicados de la llave.

CLAVE_A	CLAVE_B
384001	384001
384002	384060
384003	384200
384004	384300
384005	384355
384006	400001
384007	400050
384008	400100

Figura 1.12

En una lista densa de llaves, para cada valor llave existe un espacio de almacenamiento con una dirección que es equivalente a la llave. Los valores llave están también en una secuencia cerrada. Esto es importante, ya que, cuando se asigna el espacio de almacenamiento al archivo, comienza en el valor más bajo de la llave y se extiende al valor más alto de la misma. Se observa en la figura 1.12, la lista de CLAVE_A es densa con pocos saltos en el rango de los valores. La CLAVE_B contiene muchos saltos, los valores

¹ Fuente: Diseño general de sistemas de información John G. Burch.

• Fuente: Propia.

no utilizados en el rango de los valores de la llave provocan un desperdicio en el espacio de almacenamiento. Se debe asignar el mismo aún cuando no sea utilizado.

Es común que las llaves de registros no cumplan con los requisitos anteriores, pero es necesario el direccionamiento directo, entonces, se especifica el método de acceso alternativo de hashing. Este método también se conoce como método transformación de llaves, se refiere al proceso de obtener una dirección de almacenamiento a partir de un campo llave. Se diseña un algoritmo para transformar el valor de la llave en otro valor que sirva como una dirección de almacenamiento.

Se pueden desarrollar infinidad de algoritmos. Por ejemplo, cuando se almacenan registros de información sobre el personal de una empresa, frecuentemente se elige el número del seguro social para utilizarlo como llave del registro. Sin embargo, ya que el número no cumple con los criterios mencionados anteriormente para las llaves de direccionamiento directo, el diseño debe especificar un procedimiento de hashing que convierta el número del seguro social en un valor correspondiente a una dirección de almacenamiento. Por ejemplo, se usa el siguiente algoritmo de hashing para transformar el número en una dirección de almacenamiento adecuado.

1. Se quitan los tres primeros dígitos del número del Seguro Social. El número 456821455 se convierte en 821455.
2. Se divide la nueva llave entre el número primo, el 41 se utiliza en este ejemplo. Se divide 821455 entre 41.
3. Se utiliza la división modular. Este procedimiento especifica que el cociente se descarta y se toma como dirección de almacenamiento el residuo: cociente 20035 y el residuo 20 así que el lugar de almacenamiento está dado por el residuo, 20 en este caso. Según el medio real de almacenamiento, la dirección 20 significará algo específico, por ejemplo, un bloque en un disco magnético.

Los algoritmos hashing tienen las siguientes características:

- Posibilidad de repetición. La capacidad de almacenar un registro mediante un algoritmo y recuperarlo, utilizando el mismo algoritmo, es un requerimiento importante, tomando en cuenta que en la división por dispersión, no se utilizan números aleatorios, ya que éstos cambian con el tiempo.
- Distribución uniforme. Si se va almacenar un archivo en un espacio que permite el almacenamiento de 10000 registros, éstos deben distribuirse de manera uniforme en todo el espacio asignado en vez de acumularse todos juntos. Esta distribución garantiza una recuperación más rápida de los registros y hace un mejor uso del espacio.

- Minimizar sinónimos. Cuando un procedimiento de hashing es deficiente e insuficiente se generan con frecuencia sinónimos. Por ejemplo, aplicando el algoritmo siguiente: dividir la llave entre 1000 y utilizar el residuo como dirección.

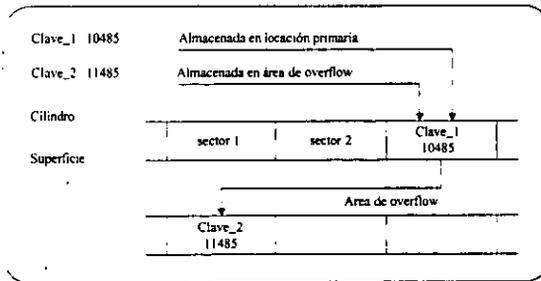
CLAVE_1	CLAVE_2
10485	11485

*Figura 1.13

Para la CLAVE_1 y la CLAVE_2 de la figura 1.13, el cociente es 10 y 11 respectivamente y el residuo en 845 para ambas claves. En la figura 1.14 muestra un área de overflow para proporcionar el almacenamiento de los registros cuando aparecen sinónimos.

Cuando se almacena un registro, el algoritmo hashing se lleva a cabo y se obtiene la dirección. El programa accesa esa área de almacenamiento y, si no está ocupada, el registro se almacena ahí. Si ya existe un registro almacenado en ese lugar, el nuevo registro se escribe en el área de overflow.

Si el sistema debe recuperar un registro, el algoritmo se lleva a cabo y se determina la dirección de almacenamiento. Entonces, se verifica el registro del área de almacenamiento. Si no es el correcto, significa que apareció un sinónimo, el sistema automáticamente va al área de overflow y recupera el registro para su procesamiento.



*Figura 1.14. Almacenamiento para claves duplicadas.

* Fuente: Propia.

* Fuente: Análisis y diseño de sistemas de Información, James A. Sedgwick.

1.4.3 Organización secuencial indexada

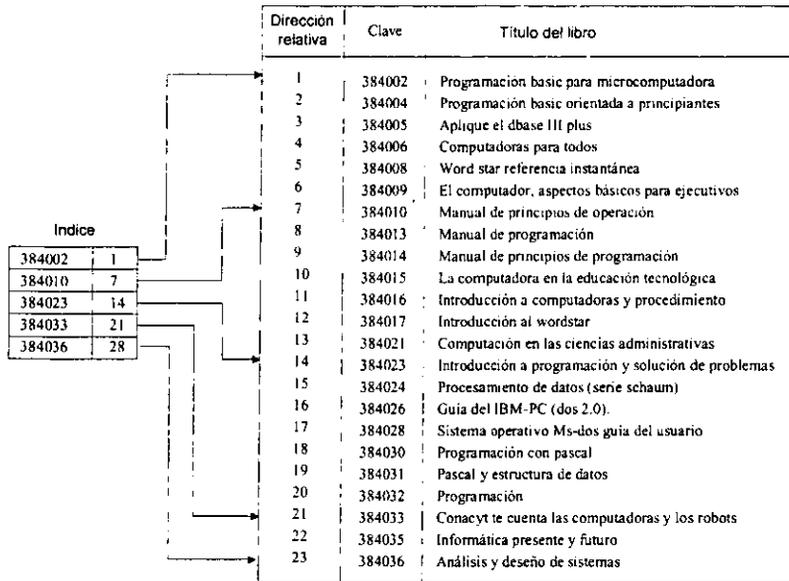
Un archivo secuencial indexado está formado en realidad por dos archivos: un archivo que contiene los datos clasificados y ordenados por un campo llave y un archivo que contiene una serie de índices utilizados para el acceso de datos.

La organización secuencial indexada de archivos permite el acceso a los registros del archivo tanto en forma secuencial como aleatoria. Permite que el procesamiento secuencial comience en cualquier registro del archivo y el proceso de tantos otros subsecuentes, como se quiera.

En forma alterna, la organización secuencial indexada también permite obtener aleatoriamente un registro seleccionado haciendo una búsqueda en una serie de índices.

Los datos se localizan con el método secuencial indexado creando un archivo físico separado, que está compuesto de una entrada de índice para cada enésimo registro de datos y no por cada registro de datos. Primero se hace una búsqueda secuencial hasta que se determina la ubicación aproximada del registro que se está buscando. Luego, a partir de la entrada del índice, se hace una búsqueda secuencial de la porción localizada del archivo de datos hasta que se encuentra una correspondencia con el valor llave deseado. Para comprender mejor como se aplica este procedimiento para la localización de datos se explicará la figura 1.15.

Las cinco entradas en el índice indican las claves de los libros ordenados en el segmento o sublista de datos. Si quisiéramos recuperar el registro de datos que tiene el valor llave "84021". Se empezaría haciendo una búsqueda secuencial del índice a partir del registro que tiene el valor llave "84002". El registro "84021" podría estar en la primer sublista. Se sabe que dicho registro, no está en la primer sublista sólo cuando se lee el segundo valor llave en el índice "84010". A medida que se continúa la búsqueda secuencial a través del índice, se determina que si existe el registro con la llave "84021", entonces éste debe estar contenido en la sublista encabezada por el registro de datos con el valor de la llave "84010" y "84023", es decir, "84010" < "84021" < "84023". El control se transfiere luego a la dirección relativa 7, que es la ubicación del registro de datos "84010". Se hace una búsqueda secuencial en la sublista del archivo de datos real hasta que, finalmente, se encuentra una correspondencia en el ubicación 13.



*Figura 1.15. Organización secuencial indexada.

Un archivo secuencial indexado no se podrá acceder secuencialmente tan rápido como en un archivo secuencial. Tampoco se podrá acceder aleatoriamente tan rápido como en un archivo de organización directa. La ventaja más grande de la organización secuencial indexada es su flexibilidad y deberá usarse cuando sea necesario tener acceso a los datos tanto en forma secuencial como aleatoria.

* Fuente: Diseño general de sistemas de información John G. Burch.

2. ANALISIS DEL SISTEMA ACTUAL

Definición de sistema

Clasificación de los sistemas de información

Técnicas estructuradas

Herramientas CASE

Principales razones para la construcción de un sistema

Ciclo de vida del desarrollo de sistemas

Técnicas para identificar requerimientos

Servicios que presta la biblioteca

Sistema de clasificación de libros

Adquisición de libros

Problemas identificados

Identificación de las necesidades de usuarios

Justificación del sistema

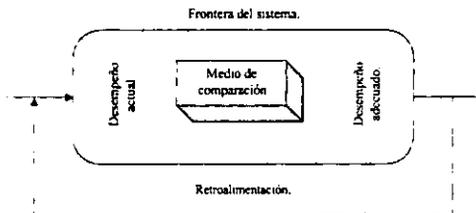
Recursos disponibles

II. ANALISIS DEL SISTEMA ACTUAL

Antes de iniciar el análisis es conveniente conocer los conceptos necesarios para comprender como surgen, se desarrollan y clasifican los sistemas.

2.1 Definición de sistema

Es importante definir lo que es sistema y sistema de información. Un sistema es un conjunto de componentes relacionados entre sí con objeto de conseguir un cierto resultado, un sistema pueden estar constituido por sistemas más pequeños, es decir, los sistemas pueden estar formados por varios niveles de sistemas o subsistemas. Los sistemas interaccionan con su medio ambiente, el cual está formado por todos los objetos que se encuentran fuera de la frontera del sistema. Los sistemas que interactúan con su medio ambiente, es decir reciben entradas y producen salidas, se llaman sistemas abiertos. Al contrario, de aquellos que no interactúan con su medio ambiente, se conocen como sistemas cerrados. Todos los sistemas tienen niveles aceptables de desempeño los cuales se comparan con el desempeño actual del sistema, ver la figura 2.1, la información obtenida, al comparar estos desempeños, junto con el proceso de reportar las diferencias a los elementos de control se conoce como retroalimentación. Recibir y evaluar es el trabajo de la retroalimentación, lo cual permite al sistema determinar qué tan bien está operando.



*Figura 2.1

Los sistemas cerrados sostienen su nivel de operación siempre y cuando posean información de control adecuada y no necesiten nada de su medio ambiente. Dado que esta condición no puede sostenerse por mucho tiempo, prácticamente no existen los sistemas cerrados.

Un sistema de información es una integración ordenada de personas, datos y tecnología con el objeto de apoyar y mejorar las labores cotidianas de la empresa. Así como satisfacer las demandas de información para la toma de decisiones y solución de problemas. En este tipo de sistemas, la información fluye de una persona o un departamento hacia otros, y puede ser desde una comunicación interna entre los diferentes componentes de una organización hasta un sistema de computo que genera reportes para los usuarios. Los sistemas de información dan servicio a los demás sistemas de la organización y combinan sus componentes para que trabajen adecuadamente para alcanzar un objetivo común.

* Fuente: Análisis y diseño de sistemas de Información, Janes A. Send.

Todos los sistemas de información grandes o sencillos están formados por los siguientes componentes estructurales:

- **Entrada:** representan a todos los datos, texto, voz e imágenes que entran al sistema de información, así como los métodos y medios por los cuales se capturan e introducen datos. En la actualidad, los medios más comunes para la entrada de datos son: lectoras de códigos de barras y el teclado. La tendencia es hacia los sistemas de reconocimiento de voz y escritura manual.
 - **Modelos:** es la forma de representar la realidad. En los sistemas de información se construyen varios modelos con propósitos diferentes. Un propósito es incluir los procedimientos que debe seguir el personal para llevar a cabo determinadas tareas. Otro es el desarrollo de modelos matemáticos y lógicos para convertir datos en información. Las técnicas de modelación más populares son: tablas y árboles de decisión, diagramas de flujo tradicionales, diagramas Nassi- Shneiderman, HIPO, diagramas de estructura, diagramas Warnier-Orr y el empleo de prototipos.
 - **Salida:** la información que producen los sistemas de información es considerada como la salida. Antes de convertirse en salida adecuada, ciertos datos requieren algunos procesos, otros, se almacenan y cuando se requieren se consideran salidas aunque no sean sometidos a procesos. Se puede presentar la salida en pantallas, impresoras, dispositivos de audio, sistemas de respuesta por voz y correo electrónico.
 - **Tecnología:** Consta de tres componentes principales: la computadora y almacenamiento auxiliar, las telecomunicaciones y los programas. Estos elementos son indispensables para trabajar en la creación de sistemas de información.
 - **Base de datos:** la base de datos es un lugar en donde se almacenan los datos más significativos, los cuales son compartidos por los usuarios para diversas aplicaciones. Los datos pueden ser una combinación de voz, imágenes, texto y números. La base de datos se considera desde dos puntos de vista: el lógico el cual determina cómo buscar, asociar y recuperar los datos almacenados para satisfacer las necesidades de información; por otro lado, el punto de vista físico hace referencia a los medios de almacenamiento, como cintas, discos, disquetes, etc.
 - **Controles:** la aplicación de controles correctamente aseguran la protección al sistema de información contra abusos por parte de personas, desastres naturales y servicios incompetentes y además, permite detectar tareas que realiza el sistema de información en forma deficiente.
 - **Personas:** son todos los participantes en el desarrollo de los sistemas de información, por ejemplo, los propietarios de los sistemas, usuarios de sistemas, analistas, programadores, etc.
-

La forma de estos componentes esta influenciada por fuerzas o aspectos a considerar en el diseño de sistemas. Entre los más importantes se encuentran: la integración, interfaz del usuario con el sistema, calidad y utilidad de la información, requerimientos del sistema y factibilidad. Se deben definir en detalle las fuerzas de diseño y determinar su nivel de impacto y su interacción entre si.

2.2 Clasificación de los sistemas de información

Los sistemas de información se desarrollan con diferentes propósitos, los cuales dependen de las necesidades de la empresa. Generalmente se clasifican en los siguientes tipos:

- **Sistemas de procesamiento de datos:** son sistemas de información computarizados, desarrollados para procesar información, el personal participa en la captura de la información, es decir, ejecutan actividades rutinarias de la empresa. Estos sistemas liberan el tedio y la rutina a las tareas que se realizan manualmente, sin embargo, el elemento humano sigue participando, al llevar a cabo la captura de la información requerida.
- **Sistemas de información para la administración:** soportan un amplio aspecto de tareas de la organización, además del procesamiento de datos, incluyen el análisis y la toma de decisiones. Los usuarios de estos sistemas utilizan una base de datos compartida para tener acceso a la información y utilizan modelos para la interpretación y uso de la información, además proporcionan informes periódicos para la planeación, el control y la toma de decisiones.
- **Sistema de apoyo para la toma de decisiones.** Se distingue del sistema de información para la administración, al enfatizar en el soporte en cada una de las etapas de la toma de decisiones. Sin embargo, la decisión depende de la persona responsable. Este tipo de sistemas se diseñan con una orientación hacia la persona o el grupo que hará uso del sistema.
- **Sistemas expertos:** son aquellos que capturan y utilizan el conocimiento de un experto, para la solución de un problema particular de la organización. En los sistemas de apoyo para la toma de decisiones el responsable finalmente toma de decisión, un sistema experto selecciona la mejor solución al problema. Los elementos básicos de un sistema experto son: la base de conocimientos y una máquina de inferencia que liga al usuario con el sistema procesando sus solicitudes mediante lenguajes de programación.

El sistema propuesto realiza básicamente tres procesos: la entrada de datos donde la información es introducida. El segundo tipo de proceso es la salida, aquí es donde se confirma una acción o se activa una respuesta. Por ejemplo, el préstamo de un libro o el

pago de una multa por no devolver el libro el día pactado. El tercer tipo de proceso es conocido como mantenimiento del sistema, aquí es donde se añaden nuevos registros, se dan de baja a usuarios, se corrigen datos, etc.

Este sistema puede clasificarse como de procesamiento de datos, sin olvidar que el módulo de estadísticas bien puede ubicarse en sistemas de información para la administración ya que genera reportes donde se muestra a los grupos que más utilizan libros, libros menos usados, esta información puede servir de apoyo para tomar decisiones.

2.3 Técnicas estructuradas

Las técnicas estructuradas son métodos formales de división de un problema de empresa en fragmentos y relaciones manejables, la unión consecutiva de estos fragmentos y relaciones dan como resultado una solución de empresa útil para resolver el problema. Estas técnicas utilizan el método divide y vencerás para resolver los problemas relacionados con el desarrollo de software de sistemas.

2.3.1 Programación estructurada

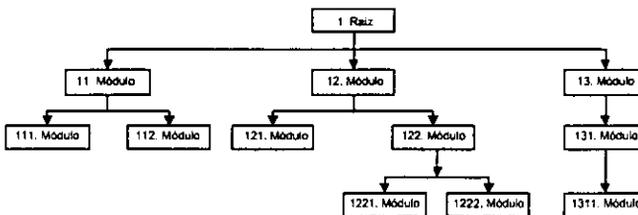


Figura 2.2

La programación estructurada es una técnica orientada a procesos para el diseño y la escritura de programas con mayor claridad y consistencia, es decir, tiene que ver sólo con la lógica y la codificación de los programas y sugiere que el esquema lógico de cualquier programa debería ser escrito de acuerdo a las siguientes reglas:

1. El programa tiene un diseño modular y descendente

En la programación modular el programa se divide en módulos (partes independientes), cada una de las cuales ejecuta una única actividad o tarea y se analiza, codifica y depura por separado. Cada programa contiene un módulo denominado programa principal que controla todo lo que sucede; transfiere el control a submódulos, de modo que ellos puedan ejecutar sus funciones; sin embargo, cada

¹ Fuente: Turbo Pascal, Luis Joyanes Aguilar.

submódulo, devuelve el control al módulo principal cuando se haya completado su tarea. El proceso sucesivo de subdivisión de módulos continúa hasta que cada módulo tenga solamente una tarea específica que ejecutar. Esta tarea puede ser entrada, salida, manipulación de datos, control de otro módulos o alguna combinación de éstos.

Los módulos son independientes en el sentido de que ningún módulo puede tener acceso directo a cualquier otro, un módulo transfiere el control temporalmente a otro submódulo, sin embargo, cada módulo debe devolver el control al módulo del cual se recibe originalmente el control. Los resultados producidos por el módulo pueden ser utilizados por cualquier otro módulo cuando se transfiere a ellos el control. Por ejemplo, la figura 2.2 se observa que el módulo 122 puede llamar al 1221 para que realice alguna tarea específica y cuando finalice, el control será devuelto al módulo 122. El módulo 13 no puede llamar al 1311 y 111 tampoco puede hacer el llamado al 112.

2. Cada módulo se codifica utilizando las tres estructuras de control básicas
 - a) Una secuencia de instrucciones o grupo de instrucciones. La secuencia consiste sencillamente en un paso seguido de otro, las figura 2.3.1 muestra que primero se ejecuta A y luego B, etc.
 - b) Una selección de instrucciones o grupo de instrucciones basadas en ciertos criterios de decisión. La estructura de selección requiere una prueba de cumplimiento de una condición, figura 2.3.2, seguida de dos trayectos de control de programa alternativos, el resultado que elija depende del resultado de la prueba.
 - c) Una iteración de instrucciones o grupo de instrucciones que se repite sobre la base de ciertos criterios. La estructura de ciclo, figura 2.3.3, hace que el programa ejecute una o varias operaciones mientras se cumpla una condición. Cuando la condición se deje de cumplir, el proceso de iteración se interrumpe.

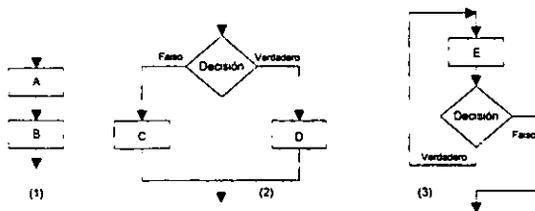


Figura 2.3

Fuente: informática presente y futuro, Donald H. Senders.

Las instrucciones del punto tres puede repetirse o anidarse en otras construcciones. Una característica importante de las estructuras en que cada construcción debe tener la propiedad de disponer de una sola entrada y una salida. El código estructurado se lee de arriba abajo, sin referencias hacia atrás. Se diseña la lógica de los programas mediante herramientas de modelización como organigramas, gráficos de cuadros (diagramas Nassi-schneiderman), pseudocódigo o diagramas de acción. Después se codifica dicha estructura, respetando las estructuras de control limitadas y las propiedades de una sola entrada y una sola salida. La programación estructurada no es una programación sin GOTO, busca sólo controlar su uso indiscriminado, es especial cuando un GOTO envía el control hacia atrás, es decir, de abajo arriba, dentro del programa, lo que crearía más de un punto de entrada en una secuencia de instrucciones. Como técnica estructurada, la programación estructurada sólo da apoyo a la implantación y soporte del ciclo de vida de desarrollo de sistemas. Por último, utilizando esta técnica, los programas son más fáciles de escribir, probar y depurar, y sencillos de mantener, debido a que se leen de principio a fin.

2.3.2 Diseño estructurado

Es una técnica orientada a procesos se utiliza para fragmentar un programa grande en un conjunto jerarquizado de módulos y obtener un programa informático más fácil de implementar y mantener. Se diseña el programa mediante una distribución jerárquica descendente de módulos. Donde, el módulo es un grupo de instrucciones, un bloque, un programa, subrutina o un párrafo. Se han desarrollado varias técnicas para conseguir diseños bien estructurados, los autores mas conocidos son:

- Ed Yourdon y Larry Constantine. Esta escuela obtiene la estructura de software ideal por el estudio de flujo de datos a través de las funciones de programa necesarias. Se describe la estructura descendente de módulos en forma de árbol invertido.
- Jean-Dominique Warnier y Ken Orr. Esta obtiene la estructura del software ideal mediante el estudio del contenido de las salidas y la entradas. Se muestra la posición jerárquica descendente en forma de serie de llaves de izquierda a derecha.
- Michael Jackson. Esta técnica obtiene la estructura de software por el estudio del contenido de las salidas y las entradas, pero muestra la disposición jerárquica descendente de los módulos en forma de árbol invertido.

Estas técnicas, su propósito es diseñar procesos, y en especial de software, la más conocida y utilizada es la Ed Yourdon y Larry Constantine, la cual divide un programa en un conjunto jerárquico de módulos descendente, con las siguiente características:

- Módulos con una fuerte cohesión, cada uno de los módulos deben generar una sola función, de esta manera los módulos son utilizables en otros programas.

- Los módulos deben tener un débil acoplamiento, es otras palabras, deberán tener una dependencia mínima unos de otros, para reducir los cambios en los módulos cuando éstos se modifiquen o se integren otros.

Se obtiene un modelo de software llamado diagrama de estructuras del diseño estructurado de Yourdon, este diagrama surge del estudio del flujo de datos a través del programa, y es utilizado en la fase de diseño del ciclo de vida del desarrollo de sistemas. Sin embargo, no cubre todos los aspectos del diseño, es decir, no se aplica en diseñar entradas, bases de datos o archivos. Se obtienen ventajas cuando se aplica esta técnica, las principales son: los programas que se desglosan conforme al diseño estructurado son fáciles de leer y probar por diferentes programadores, ya que las interfaces entre módulos están definidas y limitadas por las reglas, los módulos cuya prueba sea correcta debería de funcionar bien cuando se integre al sistema global.

2.3.3 Modelización de datos

La modelización de datos es un complemento de las técnicas orientadas a procesos, se define como la técnica orientada por los datos que representa un sistema en función de sus datos, independientemente de cómo se procesen dichos datos para producir información. El concepto es simple, si se capturan los datos y se almacenan en estructuras flexibles de archivos y bases de datos, todas las necesidades actuales y futuras de información pueden satisfacerse por medio del uso de éstos datos.

La modelización de los datos se lleva a cabo de la siguiente manera: se identifica las entidades en torno a las cuales captan datos las aplicaciones o la empresa. A continuación se identifica aquellos atributos que describen una presencia de cada entidad. Por ejemplo, los siguientes atributos describen una presencia de *usuario* (entidad): Número de control del usuario, nombre del usuario, dirección de su domicilio, etc. Posteriormente se diseñan entradas para capturar éstos atributos. También se podrá diseñar numerosas variedades de salidas para resumir informaciones sobre estos y otros atributos. Por último, se identifican las actividades de empresa que tienen lugar entre las entidades, por ejemplo, "*los usuarios se llevan libros prestados*", entonces libros y usuarios son entidades "*se prestan libros*" es una actividad de la empresa que asocia la presencia de las entidades libros y usuarios, esta asociación nos permite conocer otro tipo de información. Es decir, no solamente se almacenan datos de los libros y los usuarios, haciendo uso de las asociaciones se puede conocer que usuarios hacen uso de un determinado tipo de libros. Normalmente el modelo de datos queda definido con el descubrimiento de las entidades, relaciones y atributos.

Algunos métodos para modelar datos, utilizan técnicas formales para asegurar que el modelo de datos sea lo suficientemente flexible para adaptarse a las necesidades actuales y futuras que se basen en los mismos datos. Para lograr esto se utiliza una técnica denominada normalización, tratada en el capítulo tres.

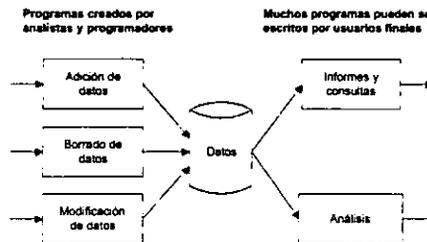
2.3.4 Ingeniería de información

La ingeniería de información es una técnica basada en los datos, pero también pone atención en los procesos, que se aplican en las empresas consideradas en su conjunto más que a proyectos circunstanciales concretos. Aunque la técnica propone un equilibrio entre los métodos orientados a datos y a procesos, se basa claramente en los datos; primero se elaboran los modelos de datos y después los procesos. Esta técnica estructurada cubre casi todo el ciclo de vida. En realidad, la única fase no incluida en la ingeniería de información es el soporte de sistemas.

En la figura 2.4 se observa que la ingeniería de la información se basa principalmente en los datos. Se elabora una base de datos integrada y flexible para cada área de empresa identificada. Las aplicaciones se construyen en torno a esta base. En la mayoría de los casos las aplicaciones que aparecen a la izquierda pueden ser creadas por los usuarios finales. Las aplicaciones críticas deberán ser desarrolladas por analistas y programadores.

2.3.5 Técnicas de prototipos y desarrollo rápido

El desarrollo de prototipos es una técnica de ingeniería utilizada para desarrollar modelos a escala o simulada de un producto o de alguno de sus componentes. En la creación de sistemas de información se usa para identificar con rapidez, las necesidades particulares de información del usuario, generalmente se realiza en la primeras etapas del ciclo de vida del desarrollo de sistemas. Los objetivos principales son:



¹Figura 2.4. La filosofía de la ingeniería de la información.

- Detectar las reacciones iniciales de los usuario, es decir, ajustar las necesidades planteadas con las características modeladas por el prototipo.
- Obtener las sugerencias de los usuarios sobre modificaciones y las posibles innovaciones al sistema para satisfacer mejor las necesidades.

¹ Fuente: Análisis y diseño de sistemas de información, Jeffrey L. Whitten.

- Detectar las prioridades que deberán considerarse próximamente para el desarrollo de prototipos y determinar aquellas partes del sistema que requieran implantarse primero.

El prototipo de rendimiento, está relacionado con la construcción de un sistema que si bien funciona, no llega a ser eficiente. Cuenta con todas las características necesarias, así que los usuarios se relacionan con el sistema, adaptándose a las interfaces y a los tipos de salidas disponibles. Los procesos de recuperación y almacenamiento son limitados, ya que los programas se escribieron de manera apresurada con el fin exclusivo de que fueran operativos, aunque no eficientes.

Modelo a escala no funcional, son aquellos que se construyen con el objeto de evaluar ciertos aspectos de diseño, por ejemplo, en un sistema que requiere una codificación extensa, por lo que se decide mostrar la utilidad del sistema con un prototipo que funcione únicamente con los procesos de entrada y salida. Omitiendo el resto de los procesos por ser costoso y requerir de tiempo.

Modelo a escala completa, en otras palabras, el prototipo es completamente funcional, es adecuado cuando se planea implantar el mismo sistema de información en varias instalaciones de la empresa.

En el uso de prototipos, los usuarios se hacen participantes más activos en el desarrollo del sistema, lo que simplifica la definición de las necesidades por el hecho de que muchos usuarios no comprenden o no son capaces de manifestar de manera detallada sus necesidades hasta que observan un prototipo. Sin embargo, los prototipos suelen pasar la fase de análisis y diseño con rapidez sin haber comprendido bien los problemas. También pueden desalentar la consideración de soluciones técnicas posibles, es común, seguir adelante con el primer prototipo que genere una reacción favorable en los usuarios. El prototipo no es un sustituto de las técnicas estructuradas, es decir, no es posible crear dicho prototipo sin especificaciones, más bien ellos actúan de complemento a las técnicas de especificaciones apropiadas.

El desarrollo rápido de aplicaciones es una combinación de diversas técnicas estructuradas, principalmente, la ingeniería de información basada en datos, con técnicas de prototipos y desarrollo conjunto de aplicaciones cuyo fin es acelerar el desarrollo de sistemas. El equipo de desarrollo primero construye los modelos previos de datos y de procesos de las necesidades. Los prototipos ayudan al analista y a los usuarios a verificar dichas necesidades, y a pulir formalmente éstos modelos.

Por ejemplo, si el sistema propuesto funciona de manera esperada, entonces se instalará en todos los CETIS, de tal forma que se le considera como modelo a escala completa.

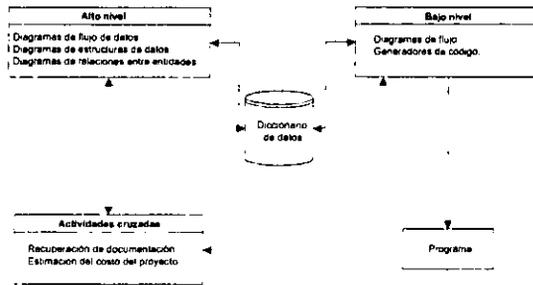


Figura 2.5 Clasificación de la herramientas CASE

2.4 Herramientas CASE

Durante años, se ha utilizado tecnología informática para resolver los problemas de empresa planteados por los usuarios, sin embargo, no se avanza con la misma rapidez a la hora de aplicar esta misma tecnología al problema de desarrollo de sistemas de información. Actualmente se comercializa una tecnología llamada *ingeniería de sistemas asistida por computadora (CASE)*, la cual se define como la aplicación de tecnología informática a las actividades, las técnicas y las metodología propias del desarrollo de sistemas y su objetivo es acelerar el proceso del desarrollo y mejorar la calidad de los sistemas. La figura 2.5 muestra la relación que existe entre las herramientas con el diccionarios de datos.

La tecnología CASE tiene un parecido con la combinación de *diseño asistido por computadora y fabricación asistida por computadora (CAD/CAM)*. Se aplican las herramientas CAD para diseñar y analizar nuevos productor y después las herramientas CAM crearán automáticamente los programas informáticos que harán funcionar la maquinaria de la planta de fabricación para producir el diseño. CASE pretende hacer lo mismo pero en el desarrollo de sistemas de información. En la actualidad no existe un programa que de el soporte a todas las fases del ciclo de vida, por lo general, un programa solamente se aplica a una fase o a una parte. Se pueden clasificar a estas herramientas de la siguiente manera:

- CASE de alto nivel para aquellas herramientas que automatizan o apoyan las fases de planificación, el análisis y el soporte de sistemas. Estas pretenden ayudar a los analistas y consultores a capturar, almacenar, organizar y analizar los modelos de la empresa para definir y establecer las prioridades. Por ejemplo, las herramientas de ADW Planing Workbench de KnowledgeWare, describe a la empresa en términos de sus metas, objetivos, factores críticos de éxito, problemas, estructura organizativa, ubicaciones geográficas y necesidades de información. Bachman Analyst y Bachman Designer de Bachman son utilizadas para expresar las necesidades y los diseños en

³ Fuente: Diseño general de sistemas de información, John G. Burch.

forma de modelos gráficos y descripciones asociadas, también suministran capacidad analítica para evaluar la calidad y la integridad de estos modelos.

- CASE de bajo nivel son herramientas que tienen su aplicación en el diseño detallado, la implantación y el soporte de sistemas, se pueden mencionar EASEL Workbench, de Easel Corporation para crear interfaz gráfica de usuario, APS/PW, de INTERSOLV para generar programas de aplicación a partir de las especificaciones del análisis y diseño, Cobol Structuring Facility, de IBM usado para leer programas no estructurados y convertirlos en código estructurado, a esto se conoce como reingeniería, el cual es un proceso por el cual se altera o se reestructura un sistema existente. Recientemente apareció Excelerator de INTERSOLV, herramienta de ingeniería inversa, analiza el código de los programas de aplicación y las bases de datos para crear representaciones de alto nivel, es decir, produce diagramas de estructura, organigramas, árboles de decisión e inclusive, ideas del programador y del analista de sistemas.
- CASE cruzado, apoyan las actividades que tienen lugar a lo largo de todo el ciclo de vida. Se incluyen actividades como gestión de proyectos y la estimación. HyperAnalyst de Rapid System gestiona los procesos, es decir, es capaz de invocar herramientas CASE en los momentos adecuados durante la aplicación de la metodología. Project Bridge de ABT intenta evaluar de forma precisa la envergadura de un proyecto y estimar el tiempo y el costo. Por último, la documentación del sistema es también una actividad cruzada, el programa ADW/DOC de KnowledgeWare permite recuperar automáticamente la información, las especificaciones o los diagramas apropiados. Puede incluso incorporar archivos de tratamiento de texto y hojas de cálculo.

2.5 Principales razones para la construcción de un sistema

Los usuarios y directivos se encuentran directamente implicados en las actividades de la organización, conocen en donde se requieren mejoras, generalmente son los encargados de impulsar proyectos de sistemas. Sin embargo, es recomendable que las empresas tengan una directiva de planificación de sistemas encargada de aplicar los métodos formales de planificación a largo plazo para diseñar proyectos de sistemas que reflejen y apoyen los planes de la empresa.

Los proyectos de sistemas de información, tienen su origen en cualquier parte de la empresa y se relacionan con problemas o carencias de la organización. Una buena razón para emprender el análisis de un sistema puede ser:

Resolver un problema: las actividades, procesos o funciones que en la actualidad, y quizá en el futuro, no satisfacen las necesidades de desempeño o expectativas, es un problema que debe ser considerado. Por lo que es necesario emprender una acción que resuelva las

dificultades. La forma de entender lo que es un problema y de cómo surgen, es imaginar las situaciones en que nunca se logran las metas o se logran parcialmente. La retroalimentación proporciona información referente al defasamiento entre el desempeño actual y el deseado. De esta forma la retroalimentación ayuda a detectar problemas. Los síntomas de problemas pueden ser: la persistencia y la cantidad de errores, el desarrollo lento, incompleto, incorrecto o la no realización del trabajo. En otras palabras, se puede decir que la revisión de las salidas, la observación de la conducta de empleados y la retroalimentación, son acciones que ayudan al analista a identificar problemas y oportunidades. El cuadro 2.6 muestra las acciones e indicios para detectar problemas en la empresa.

ACCION	INDICIO
Comparar las salidas con los criterios de desempeño.	<ul style="list-style-type: none"> • Exceso de errores. • Lentitud en el desarrollo del trabajo. • Incorrecto el desarrollo del trabajo. • Trabajo inconcluso.
Observación de la conducta de empleados.	<ul style="list-style-type: none"> • Ausentismo. • Inconformidad en el desempeño del trabajo. • Rotación de trabajadores.
Retroalimentación.	<ul style="list-style-type: none"> • Quejas. • Sugerencias. • Pérdidas de clientes.

Figura 2.6

- Aprovechar una oportunidad: buscar cambios para ampliar o mejorar el rendimiento económico de la empresa y su competitividad. El analista debe ser capaz de identificar el sitio donde los procesos pudieran mejorar, sin embargo, la gente que está usando continuamente el sistema constituye la mayor fuente de información para determinar las mejoras que se pueden hacer al sistema. La mejoría en los sistemas son cambios que otorgan beneficios considerables, entre estos cambios se encuentran la simplificación de un proceso, reducción de errores de captura y mejorar el sistema para que sea más aceptable.

Tablas de decisión: Es un modo de mostrar en forma de tabla las especificaciones de condiciones junto con sus acciones correspondientes, son de gran utilidad cuando se desea precisar las normas de la institución y las reglas de toma de decisiones. Las partes que la forman son:

- Condiciones que describen los factores que influyen sobre la decisión o la norma que se trate.
- Acciones, muestran en forma de oraciones, las posibles acciones o decisiones asociadas a la norma.
- Reglas, describen qué acciones deben tomarse para una combinación específica de condiciones.

¹ Fuente: Análisis y diseño de sistemas de información, Jeffrey L. Whitten.

		Reglas			
		1	2	3	4
C o n d i c i o n e s	Alumno inscrito actualmente	Si	Si	No	No
	El alumno tiene menos de dos libros prestados	Si	No	Si	No
A c c i o n e s	Prestar el libro	X			
	No prestar el libro		X	X	X

Figura 2.7

Norma: es todo nuevo requisito impuesto por la empresa, el gobierno o cualquier institución externa. Por ejemplo, actualmente en México se está anunciando modificaciones al impuesto al valor agregado, por lo que será necesario modificar los sistemas para incluir o modificar este impuesto. En nuestro caso una norma es la siguiente orden: *a partir de esta fecha únicamente se dará servicio de préstamo a domicilio a los alumnos inscritos en la escuela y solamente se autorizan dos libros por alumno. Quedan excluidos del servicio, aquellos que están de baja temporal.*

Se ha definido la norma sobre el préstamo de libros a domicilio, la figura 2.7 muestra la misma norma por medio de la tabla de decisiones. Las condiciones que influyen sobre la decisión que ha de tomarse en el préstamo de libros son dos: el alumno esta inscrito (Si, No) y si tiene un libro o ningún libro prestado (Si, No). Las acciones tomadas pueden ser prestar o no el libro.

La estructura ideada por James Wetherbe es una herramienta útil para agrupar e identificar los problemas, oportunidades y normas cuando se estudia a los sistemas de información. Esta estructura se conoce como PIECES, acróstico de las palabras en ingles: *performance, information, economics, control, efficiency y service*. A continuación se describe la estructura en forma más detallada.

PRESTACIONES

Productividad: cantidad de trabajo realizado en un periodo de tiempo.
 Tiempo de respuesta: tiempo medio transcurrido entre una transacción o solicitud y la respuesta a dicha solicitud.

INFORMACION

Salidas.
 Falta de alguna información.
 Falta de información necesaria o relevante.
 Información no útil.
 Información no exacta.

¹ Fuente: Análisis y diseño de sistemas de información, Jeffrey L. Whitten.

Información difícil de producir.
 Entradas.
 Datos no capturados.
 Datos capturados con errores.
 Datos capturados de forma redundante.
 Demasiados datos capturados.
 Datos almacenados.
 Datos almacenados de forma redundante.
 Datos no seguros en caso de accidentes.
 Datos mal organizados.
 Datos no flexibles.

ECONOMIA

Costos no conocidos.
 Costos elevados.

CONTROL

Seguridad o control demasiado bajo.
 Sabotaje a los datos.
 Datos almacenados de forma redundante.
 Regulaciones o directrices de privacidad de datos.
 Errores de procesos.
 Errores en la toma de decisiones.

Exceso de controles.
 Excesiva burocracia hace lento al sistema.
 Los controles causan molestias a clientes o empleados.
 Los excesivos controles provocan retrasos en el proceso.

EFICACIA

Desperdicio de tiempo de personas, máquinas o computadoras.
 Datos introducidos, copiados o procesados de forma redundante.
 Información generada en forma redundante.
 Desperdicio de materiales, personas y computadoras.

El esfuerzo requerido y los materiales son excesivos.

SERVICIOS

El sistema produce resultados inexactos, incoherentes y por tanto no confiables.
 El sistema es difícil de aprender y usar.
 El sistema es inflexible entre situaciones nuevas o excepcionales.
 El sistema es incompatible con otros sistemas.

La estructura PIECES es importante tenerla presente cuando se está analizando un sistema de información ya que permite tener un panorama global de los sucesos generadores de proyectos.

2.6 Ciclo de vida del desarrollo de sistemas

El ciclo de vida del desarrollo de sistemas es un proceso por el cual los analistas, ingenieros de software, programadores y usuarios finales elaboran sistemas de información y aplicaciones informáticas. También se puede decir, que es una herramienta de gestión de proyectos para planear, ejecutar y controlar los proyectos de desarrollo de sistemas. La figura 2.8 muestra las etapas. No existe un número exacto de etapas, tampoco un consenso con el nombre de ellas, sin embargo cualquiera que sea el número o nombres de ellas, todas las actividades, que la mayoría de los autores recomiendan están incluidas en alguna de las fases o bien en una fase adicional.



Figura 2.8

La planificación de sistemas pretende señalar y establecer prioridades sobre aquellas tecnologías y aplicaciones que producen un beneficio máximo para la empresa, los directivos de la organización deben estar de acuerdo con el desarrollo del plan estratégico, de lo contrario carece de utilidad. La metodología para la planificación de sistemas se puede resumir en tres fases:

1. Estudiar el cometido de la empresa

Idealmente la planificación debe ser para toda la empresa, sin embargo para la mayoría resulta imposible, por los costos; pero, es conveniente que el ámbito se reduzca a una división, una planta o a una unidad operativa. El cometido de la empresa de define generalmente en términos de clientes, productos, servicios, recursos humanos, materiales, lugares geográficos de operación, estructuras y filosofía de gestión, metas y objetivos corporativos, restricciones de empresa y factores críticos de éxito de la empresa. La figura 2.9 muestra que al estudiar el cometido de la empresa se descubren los planes de la empresa. También se observa un rombo que indica un punto de control, es decir, el esfuerzo de planificación se puede cancelar por falta de compromiso de los directivos.

2. Definir una arquitectura de información

Esta arquitectura se define como un plan para seleccionar la tecnología de información y el desarrollo de los sistemas de información necesarios para apoyar el cometido de la empresa. La entrada de información para esta fase, son los planes de la empresa y los detalles y limitaciones de los sistemas existentes, obtenidos de la documentación durante el soporte de sistemas.

3. Análisis de áreas de empresa

La arquitectura de información es ya un buen plan de sistemas de información, pero algunas empresas buscan afinar aún más su plan para definir proyectos específicos de desarrollo de sistemas. Así, la siguiente fase de planeación de sistemas consiste en evaluar y establecer prioridades sobre los proyectos. Todos los directivos de las áreas de una empresa específica, deben de participar, con el fin de identificar las aplicaciones necesarias y establecer prioridades de los proyectos para el desarrollo de las aplicaciones. También pueden verse implicados algunos usuarios del sistema, normalmente aquellos identificados con los

² Fuente: Análisis y diseño de sistemas, Kendall y Kendall.

métodos de trabajo de la empresa. Se observa en la figura 2.9, el principal resultado obtenido en la fase, son los proyectos planificados de desarrollo de aplicaciones, que serán eventualmente transmitidos al analista del sistema. Esta información sirve como útil borrador en las primeras fases del ciclo de vida de desarrollo del sistema.

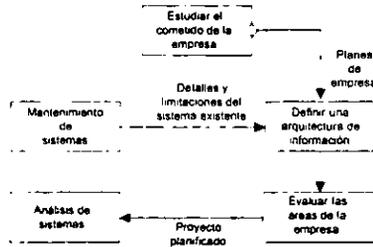


Figura 2.9

Después de haber definido algunos conceptos de sistemas de información, incluyendo la planificación de sistemas, entramos al estudio del análisis del sistema, el cual consiste en definir el ámbito o alcance del sistema, donde se incluye la identificación de los usuarios, administradores, la detección de problemas, oportunidades y la identificación de cualquier restricción técnica y las soluciones posibles. Una vez obtenida esta información se evalúa la viabilidad del proyecto. Después se estudia y se analiza el sistema actual, siempre hay uno previo, ya sea que utilice computadora o sea manual.

2.7 Técnicas para identificar requerimientos

La investigación de hechos es un proceso formal que utiliza procedimientos de búsqueda, entrevistas, cuestionarios, y otras técnicas para recoger toda la información disponible sobre los sistemas, sus necesidades y las preferencias mostradas.

Existen muchas ocasiones para investigar los hechos durante el ciclo de vida del desarrollo de sistemas. Sin embargo, es vital durante las fases de la planificación y el análisis. Es durante estas fases cuando se aprende el vocabulario de la empresa y el funcionamiento del sistema, así como sus problemas, oportunidades, limitaciones, necesidades y prioridades. Durante el diseño, la investigación adquiere un aspecto más técnico conforme se incrementan los conocimientos sobre la tecnología elegida para el nuevo sistema. Durante la fase de soporte, la investigación adquiere mayor importancia por la necesidad de determinar en qué medida alcanza el punto a partir del cual ha de plantearse la necesidad del desarrollo de un nuevo sistema.

Para identificar los requerimientos de información de los sistemas se utilizan los siguientes métodos: entrevista, cuestionario, revisión de los registros y la observación.

¹ Fuente: Análisis y diseño de sistemas de información, Jeffrey L. Whitten.

- La entrevista se emplea para reunir información proveniente de grupos de personas. Por lo general, los entrevistados son usuarios de los sistemas existentes o usuarios en potencia del sistema propuesto. Este método es la mejor fuente de información cualitativa. Los otros métodos son más útiles en la recolección de información cuantitativa. Las entrevistas permiten descubrir áreas mal comprendidas, expectativas poco realistas e incluso indicadores de resistencia hacia el sistema propuesto.
- El cuestionario es una técnica de recopilación de información que permite recoger opiniones, posturas, conductas y características de personas de una organización involucradas en la operación o implantación de un sistema. Las respuestas que se obtienen mediante cuestionarios de preguntas cerradas pueden cuantificarse. Las respuestas a cuestionarios que utilizan preguntas abiertas se analizan e interpretan de acuerdo a la sensibilidad y criterio del analista. El uso de cuestionarios sirve para determinar qué tan difundido se encuentra el sentimiento hacia el sistema. Además se puede sondear una gran muestra de usuarios con el fin de detectar problemas, o bien, tener presente aspectos importantes, antes de la programación de entrevistas. Este método puede proporcionar datos confiables ya que el anonimato de los encuestados puede conducir a respuestas más honestas. Sin embargo, no se puede observar las expresiones o reacciones de los encuestados y la respuesta puede ser limitada ya que es posible que no tenga mucha importancia, para los encuestados, llenar el cuestionario.
- Estudio de registros: se examina la información asentada en ellos y relacionada con los usuarios y el sistema. La revisión de los registros puede efectuarse al comienzo del estudio, como introducción o después y sirve de base para comparar las operaciones actuales, por lo tanto, los registros pueden indicar qué está sucediendo.
- La observación permite ganar información que no se puede obtener por otras técnicas. A través de la observación, la información es de primera mano sobre la forma en que se efectúan las actividades. Este método es más útil cuando se necesita conocer la forma en que se manejan los documentos y se llevan a cabo los procesos y; o si se siguen todos los pasos especificados.

2.8 Servicios que presta la biblioteca

Es evidente que no todas las bibliotecas de los planteles dependientes del subsistema funcionan de manera óptima; la mayoría de ellas no cuentan con lineamientos que normen su funcionamiento, no manejan sistemas de clasificación y carecen de la bibliografía básica necesaria para apoyar los programas de estudio de las carreras que se imparten en la escuela. Por lo anterior, la comunidad escolar tiene un concepto poco satisfactorio de la biblioteca, pues concibe la biblioteca como un lugar donde sólo se almacenan libros.

La biblioteca escolar es una institución al servicio de la comunidad, su meta es proporcionar en forma eficiente, a quien lo solicite, la consulta o préstamo de libros y de otros materiales que respondan a sus necesidades de información, es decir, debe ser un apoyo para todos los

estudiantes y docentes, y debe contribuir a su mejoramiento educativo, cultural, social y económico.

Las bibliotecas que forman parte de las Bibliotecas Escolares DGETI deben de prestar los siguientes servicios básicos:

- Consulta: Se proporciona ayuda personal a los usuarios en sus requerimientos de información.
- Orientación a usuarios: Se les informa sobre los contenidos de las colecciones y de los servicios que presta la biblioteca, de manera que puedan usarla de forma integral.
- Fomento a la lectura: Se ofrece a los usuarios diversas actividades culturales encaminadas a promover su acercamiento a la lectura y a fortalecer su vida cultural.
- Préstamo a domicilio: Se autoriza a los usuarios a llevar a su casa los materiales de la biblioteca. Se prestan hasta dos libros por un periodo renovable de una semana a quienes tienen credencial o alguna identificación específica de la biblioteca.

Además de los servicios deberán llevar un registro de los datos de los servicios que brindan. Estos datos se suman al final de cada mes, y con ellos se llena la hoja de estadística mensual, la cual se entrega a la Coordinación Estatal de Bibliotecas Escolares, que a su vez la hace llegar a la DGETI. Estos servicios se logran con las siguientes actividades:

- Se elabora un letrero con el horario de servicio de la biblioteca, y se coloca en un lugar visible desde el exterior.
- Se mantienen los tarjeteros de préstamo a domicilio siempre al día, pues constituye la única manera segura de saber quién tiene los libros y cuándo serán devueltos.
- Cultivar en los lectores el cuidado y buen uso de los libros.
- Fomentar el uso del préstamo a domicilio; ya que mientras más se lean los libros, es mejor. Todos los libros de la biblioteca pueden prestarse a domicilio, excepto los de consulta.
- Iniciar lo más pronto posible el registro de los datos para la estadística, considerando los puntos siguientes:
 - Número de usuarios atendidos.
 - Número de libros utilizados por los usuarios.
 - Número de libros prestados a domicilio.

- Actividades al fomento de la lectura.
- Credenciales expedidas de préstamo a domicilio.

Además de cumplir con una responsabilidad, el bibliotecario se puede dar cuenta de qué tanto y cómo se usa la biblioteca y en base en esta información saber si es necesario hacer algo para incrementar el servicio o corregir algún servicio. Por lo tanto, la labor del bibliotecario no se limita a ordenar los libros y esperar que lleguen lectores. Es necesario que los miembros de la comunidad estudiantil y docente sepan que en la biblioteca hay todo un conjunto de materiales para ellos. Por este motivo, es importante efectuar una campaña permanente de difusión, con el objeto de atraer incluso a aquellas personas que nunca habían pensado acudir a la biblioteca. Para alcanzar estas metas se realizan las actividades siguientes:

- Se da a conocer a la comunidad los servicios y las colecciones con que cuenta la biblioteca, mediante carteles y volantes que se pueden elaborar en forma sencilla, después se reparten y se pegan en el plantel.
- Organizar pláticas con maestros, estudiantes y trabajadores para que conozcan mejor los beneficios que reciben al acudir a la biblioteca escolar. Los estudiantes y docentes podrán utilizar los libros para hacer sus tareas escolares e investigaciones.
- Llevar a cabo actividades de fomento de lectura dirigidas a jóvenes y adultos. Estas pueden ser círculos de lectura, pláticas o exposiciones, entre otras. Así mismo, aprovechar para darles información sobre la biblioteca escolar.

Acceso a la biblioteca

Los investigadores, profesores y estudiantes tienen acceso a las instalaciones de la biblioteca, mediante el sistema de estantes abiertos o estantes cerrados. Lo primero quiere decir que el lector puede dirigirse libremente a la estantería para obtener las obras que le interesan. Cuando se habla de estantes cerrados quiere decir que el lector debe solicitar la obra a través de un empleado de la biblioteca.

Selección de libros

Ninguna biblioteca puede darse el lujo de adquirir todas las obras disponibles: se limita a coleccionar, entre ellas, las que pueden ser más idóneas desde el punto de vista de los objetivos que persigue la comunidad a la que sirve, por ejemplo, la biblioteca universitaria debe reflejar, en sus colecciones, las disciplinas y las áreas representadas en los programas de docencia e investigación de la universidad de la que forma parte.

Los procedimientos de selección varían, de biblioteca a biblioteca, de acuerdo con factores tan diversos como el tamaño de la institución, el número de bibliotecarios asignados a esta tarea, el monto del presupuesto de adquisiciones, y las facilidades o dificultades existentes para la comunicación entre la biblioteca y el usuario.

Clasificación

Entre los sistemas de clasificación de las obras, destacan la Clasificación Decimal Dewey, la Clasificación Decimal Universal y el sistema de la Biblioteca del Congreso de Estados Unidos. Los tres tienen por objetivo:

- Agrupar las obras por disciplinas o materias, de modo que el lector que tiene acceso directo a las colecciones pueda encontrar, en un mismo lugar, los materiales que le interesan y asignar a cada obra la clave que permita localizarla con rapidez y certeza.

Un sistema similar es el utilizado en la Universidad Nacional Autónoma de México, el cual tiene las siguientes características que lo hacen diferente:

1. Usa el modelo de estantería abierta.
2. Está instalado en la salida de la biblioteca un sistema para detectar los libros que no cumplieron el proceso de préstamo o faltó algún paso, el cual puede ser, el pasar el libro por la rampa para que sea revisada la fecha de devolución y evitar se active la alarma del detector.
3. El sistema computarizado tiene implementado dos dispositivos de entrada para leer el código de la clasificación del libro. El primero, es el lector óptico, el cual lee el código de barras y en caso de que lector tenga problemas, se usa el teclado para introducir el código en forma manual.

Es importante decir, que este sistema es más completo en comparación con el que estamos proponiendo, sin embargo, requiere de mayores recursos humanos, técnicos y materiales para implementarlo. Así que cuando se planteó como alternativa al director del CETIS la consideró adecuada, pero por falta de recursos se decidió desarrollar un sistema que utilizara los recursos disponibles, como un solo empleado para manejar la biblioteca y continuar utilizando el servicio de estantería cerrada.

El proceso del préstamo de libros en el CETIS es el siguiente: un usuario solicita un libro con un formato establecido para préstamo, el cual contiene los datos del libro que se solicita. En caso de que el libro solicitado no este disponible para préstamo a domicilio, el usuario debe llenar otro formato y posteriormente volver a solicitar otro libro. Este proceso, se repite hasta que el usuario obtiene el libro que le pueda ser útil o definitivamente cancelar la solicitud. En algunos casos el usuario pregunta al encargado si conoce un libro en donde pueda investigar un tema específico. Si el libro se presta, se archiva el formato hasta que es devuelto el libro. Después de dos días se devuelve el libro a la biblioteca, se busca el formato que corresponde al libro prestado y se

archiva dicho formato para que al finalizar el mes se calculen las estadísticas de los grupos que más utilizaron los servicios de la biblioteca.

El servicio de préstamo interno: únicamente el usuario solicita el libro y se retira a una de las mesas a consultarlo, cuando devuelve el libro también entrega el formato de préstamo de libros debidamente requisitado. El bibliotecario almacena el formato para ser usado cuando se realice la estadística de los libros consultados.

Los libros tienen un número de clasificación asignado de acuerdo al tema que tratan y con base en el sistema de clasificación decimal de Dewey, que divide a los libros en diez grupos. Este número, tiene tres dígitos enteros y, cuando la clasificación es más específica, uno o varios decimales. El cuidado de los libros se refiere en mantener en buen estado el espacio físico, efectuar la limpieza del acervo y de la estantería, y al buen manejo de libros tanto por el bibliotecario como de los usuarios. Los libros con hojas sueltas, es necesario separarlo de los demás para su reparación o mantenimiento.

2.9 Sistema de clasificación de los libros

Los materiales que conforman las diferentes colecciones de la biblioteca deben organizarse de modo que sean localizados con facilidad, rapidez y confiabilidad. Los libros pueden ordenarse de muchas maneras: por su tamaño, título, autor, o progresivamente en la medida en que se vayan adquiriendo. Sin embargo, estos arreglos resultan inadecuados para una biblioteca, pues dado el gran número de obras que poseen sería imposible recuperar con eficiencia la información contenida en ella.

Los bibliotecarios han encontrado que el mejor método para organizar las obras consiste en clasificarlas de acuerdo con el tema que tratan, porque los lectores usualmente buscan libros sobre determinada materia y prefieren tener juntas todas las obras de un tema particular. Se han desarrollado diversos sistemas de clasificación, los cuales agrupan a los materiales por su contenido. Estos sistemas facilitan la búsqueda de la información y muestran la relación que guarda un libro con el resto de la colección.

2.9.1 Sistema de clasificación Dewey

El sistema de clasificación bibliográfico más utilizado es el llamado decimal, creado por Melvin Dewey, aunque ha sido modificado para adaptarlo a las necesidades actuales. El sistema se llama decimal, porque divide el conocimiento humano en diez clases mostradas en la figura 2.10.

CLAVE	DESCRIPCION
000	Generalidades
100	Filosofía y disciplinas afines
200	Religión
300	Ciencias sociales
400	Lenguas
500	Ciencias puras
600	Ciencias aplicadas
700	Bellas artes
800	Literatura
900	Geografía e historia

Figura 2.10

Cada una de estas clases se fraccionan a su vez en diez subclases y cada una de éstas en diez divisiones; esto permite clasificar temas cada vez más específicos. Los tres primeros dígitos determinan la clase, subclase y división, a medida que aumentan los dígitos en los decimales la clasificación es más específica. Por ejemplo, en la figura 2.11 la clase Ciencias sociales, que es una de las diez primeras, se expresa con el número 300 y se subdivide a su vez en varias subclases, entre las cuales está la Educación, que se designa con el número 370, y la Educación a su vez se subdivide en sus diversas ramas, una de la cuales es la Educación de adultos que se identifica con el número 374; ésta se subdivide en Educación en grupos, a la que corresponde el número 374.2.

000	300	370	374.0	
100	310	371	374.1	
200	.	372	374.2	Educación en grupos
300	Ciencias sociales	373	.	
	370 Educación	374 Educación de adultos	.	
	.	.	.	
900	390	379	374.9	

Figura 2.11

Las obras de literatura escritas en español llevan, además del número de clasificación que corresponde al género literario, una o dos letras que indican el país de origen del autor. La figura 2.12.1 muestra la clasificación 863, la cual es la clave de la novela, figura 2.12.2 y como el autor es mexicano, entonces, de la figura 2.12.3 se toma el código del país, M para este ejemplo.

GENERO LITERARIO	AUTOR	CLASIFICACION	CODIGO
Novelística mexicana	Pacheco, José Emilio	863M	P32

Figura 2.12

^c Fuente: Manual de bibliotecología, Josefa E. Sabor.

^a Fuente: Lineamientos generales para el funcionamiento de la biblioteca escolar. Reyes Vilchis Maximino.

GENERO	CODIGO
Poesía	861
Teatro	862
Novelística	863
Ensayo	864
Oratoria	865

Figura 2.12.2

PAIS	CODIGO
España	Sin letra
Argentina	A
Cuba	Cu
México	M
Chile	Ch

Figura 2.12.3

Código del autor

El código del autor es el símbolo que permite diferenciar varias obras sobre el mismo tema escritas por diferentes autores y también cuando una persona ha escrito dos o más libros sobre el mismo tema, los números del código de autor de cada obra son distintos. Se obtiene combinando la inicial del primer apellido del autor con un número que representa a la segunda letra, el cual se toma de unas tablas numéricas diseñadas con ese propósito. Este número se llama también número Cutter, puesto que las tablas fueron creadas por el estadounidense Charles A. Cutter. Para la mayoría de los nombres se usa una letra y uno o dos números. Por ejemplo, la R es la letra inicial del primer apellido, como esta letra es consonante y la letra que le sigue es la O se pone el 6 ver las tablas en los anexos, el código del autor queda como R6, para el primer libro. El segundo, es del mismo tema y autor para ubicarlos en el estante se aumenta un número más en el código del autor, siguiendo las mismas reglas. Ahora D es consonante y la letra que sigue es R, por tanto se pone el 3, código queda como R63, tal como lo muestra el cuadro siguiente:

Número de clasificación:	641.865	Repostería mexicana
Código del autor	R6	Rodríguez María
Número de clasificación:	641.865	Los pasteles de la abuela
Código del autor	R63	Rodríguez María

En los casos en que no es factible identificar a un autor específico de la obra y, por tanto, carece de código de autor, se le asigna un código al título bajo las mismas normas que se emplean para el código de autor.

2.10 Adquisición de libros

Los libros se obtienen mediante donaciones que hacen los alumnos, en eventos especiales que se realizan para captarlos, los que adquiere la escuela con recursos propios y los que envía las autoridades de la Coordinación Estatal. Cuando ingresan los libros nuevos, por cualquiera de los medios mencionados, se elabora una acta en donde se menciona su procedencia y la descripción de los mismos, después se procede al llenado de la papeleta múltiple y al mismo tiempo se clasifica de acuerdo al tema. Los departamentos involucrados en el manejo de libros son:

- La oficina de Métodos Educativos determina las necesidades de libros requeridos para el cumplimiento de las labores encomendadas al departamento y presentarlo a la subdirección técnica, quien orienta al personal docente en el diseño y elaboración de apuntes y libros de texto. Mantiene actualizado el banco de material didáctico del plantel. Por último supervisa que el equipo didáctico del plantel se encuentren en condiciones de funcionalidad.
- La oficina de Bienes y Servicios propicia el desarrollo de acciones tendientes al autoequipamiento del plantel. Promueve acciones tendientes a generar bienes y servicios para apoyar el proceso formativo de los alumnos.
- Recursos Materiales desarrolla los métodos de adquisición y suministro del material. Vigila el uso adecuado de los materiales, equipo e instalaciones con que cuenta el plantel. Mantiene el control sobre los movimientos de resguardo, transferencia y baja de bienes.

La información generada por el sistema será usada por alumnos, maestros, personal de la escuela, pero principalmente por los departamentos sombreados en el organigrama, figura 2.13. La oficina de Recursos Materiales controla la información sobre las bajas del material bibliográfico, la oficina de Bienes y Servicios necesita conocer que títulos de libros hacen falta y para sugerir en sus eventos la donación de estos títulos de libros. La oficina de Métodos Educativos debe conocer todo el material de libros existente e incluirlos en la bibliografía de sus planes de estudio; también la oficina de Evaluación e Informática usará el sistema para integrar la información requerida para los procesos de planeación y toma de decisiones.

2.11 Problemas identificados

Algunos maestros entran hasta donde se encuentran los libros, se los llevan y en ocasiones no llenan el formato de control para el préstamo de libros, esto provoca que los libros más importantes estén fuera de control ya que estos usuarios los mantienen en su poder por un periodo prolongado de tiempo, llegando en ocasiones a no devolver estos libros.

Por falta de recursos humanos solo una persona realiza el trabajo bibliotecario, con apoyo de prestadores de servicio social. Esto hace que el servicio se vuelva lento y que exista el riesgo de que los alumnos salgan de la biblioteca sin registrar el libro ya que cuando el préstamo de libros es interno, el formato se llena hasta que es devuelto el libro.

Cuando los libros salen de la biblioteca en calidad de préstamo a domicilio, es necesario que el usuario entregue el formato con los datos del libro que solicita para poder obtenerlo. Sin embargo, en el préstamo interno dicho formato se llena después de que el usuario consultó el libro. El proceso usado en el préstamo de libros en forma interna, puede ocasionar los siguientes problemas:

- Robo de libros, cuando el usuario salga de la biblioteca sin registrar el libro.

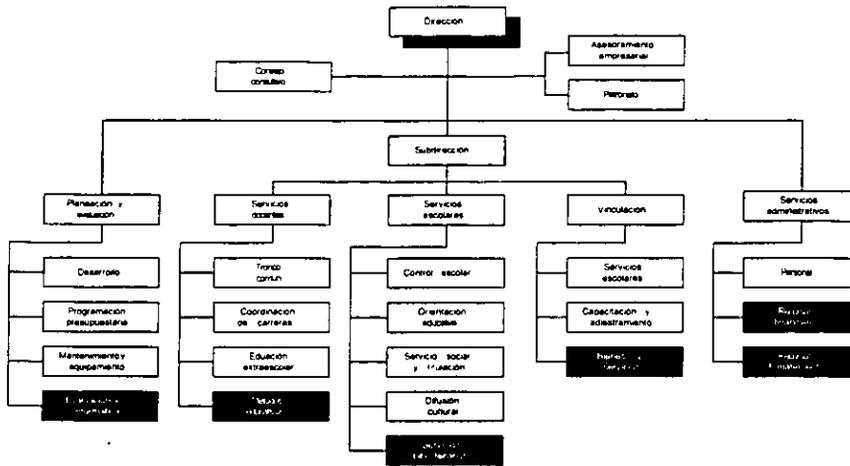


Figura 2.13

- Cambio de un libro robado por otro que tenga el usuario por medio del préstamo interno.
- En consecuencia la información generada en las estadísticas estará alterada o es poco confiable.

Para el responsable de la biblioteca es difícil detectar estas anomalías ya que por cada libro tendría que anotar datos, después compararlo con la papeleta para saber si el libro que están devolviendo es el mismo que fue prestado.

Cuando se realizan eventos para obtener libros o un usuario, por concepto de alguna multa, deba donar un libro, se les pide cualquier libro por la falta de una herramienta capaz de generar información que permita conocer que títulos de libros son en realidad los que hacen falta.

Otro problema grave es el diferente proceso que se hace en el préstamo de libros. Si no es alumno la persona que solicita un libro, entonces, puede ser maestro, principalmente, o empleado de la escuela a quien se le está prestando el libro, estas personas no llenan la papeleta, figura 2.14, así que estos libros no tienen ningún tipo de control. Esto es la causa principal de los libros extraviados.

La generación de las estadísticas es tediosa ya que se debe primero almacenar las papeletas hasta en final del mes, después se realiza el conteo y la clasificación manualmente. Esto ocasiona problemas como son: en el transcurso del mes se pueden perder papeletas o clasificar a los libros con criterios diferentes y los errores que se cometen al realizar el conteo, comunes en este tipo de trabajo.

⁷ Fuente: Manual de organización del centro de estudios tecnológicos y de servicios, y del centro de bachillerato tecnológico industrial y de servicios.

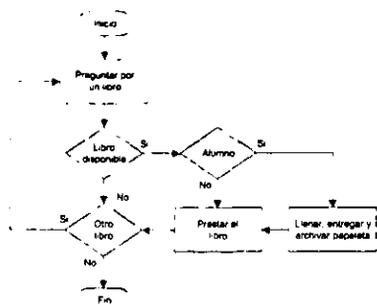


Figura 2.14

2.12 Identificación de las necesidades de los usuarios

La identificación de las necesidades se activa a partir de la aprobación para proseguir con el proyecto. Esta actividad se realiza conjuntamente por los usuarios del sistema, los dueños de los sistemas y el analista de sistemas. Se utilizan las técnicas de investigación de hechos para definir las necesidades o requerimientos de los usuarios.

Estudiar el sistema actual para conocer cómo trabaja y dónde es necesario ejecutar mejoras nos permite determinar los requerimientos del sistema. Los estudios arrojan una evaluación de la forma como trabajan los métodos empleados y si es necesario o posible realizar ajustes. Un requerimiento es una característica que debe incluirse en un nuevo sistema. Esta puede ser la inclusión de una determinada forma para capturar o procesar datos, producir información o controlar una actividad de la empresa. Es así como la determinación de requerimientos vincula el estudio de un sistema existente con la recopilación de detalles relacionados con él.

Los analistas de sistemas y los usuarios no tienen los mismos conocimientos de hechos y detalles del sistema. Por ello, el primer paso del analista es comprender la situación, aunque cierto tipo de requerimientos son tan fundamentales, que son comunes en casi todos los sistemas. Esta tarea se lleva a cabo a través de tres grandes actividades: anticipación, investigación y especificación de los mismos.

- La anticipación de los requerimientos es posible cuando el analista tiene experiencia en una área en particular puede anticipar problemas o características y requerimientos para un nuevo sistema. En esta actividad existe la posibilidad de introducir sesgos o atajos al conducir la investigación, entonces, deben darse lineamientos para estructurar una investigación en las cuestiones básicas, con la finalidad de evitar consecuencias indeseables.
- La investigación de los requerimientos es la actividad más importante del análisis de sistemas. Se estudia el sistema actual con la ayuda de herramientas y se documentan las características para iniciar el análisis.

• Fuente: Propia.

- Los datos obtenidos durante la recopilación de hechos se analizan para determinar las especificaciones de los requerimientos, es decir, la descripción de las características del nuevo sistema. Esta actividad tiene tres partes relacionadas entre sí:
 1. Análisis de datos basados en hechos reales: se examinan los datos recopilados durante el estudio.
 2. Identificación de requerimientos esenciales: características que deben incluir el nuevo sistema, cómo son los detalles de operación.
 3. Selección de estrategias para satisfacer los requerimientos: métodos que serán utilizados para alcanzar los requerimientos establecidos.

Dentro de los requerimientos de un sistema es importante determinar el tiempo de vida y el crecimiento que puede sufrir un sistema, en otras palabras, un sistema puede instalarse y trabajar muy bien durante cierto tiempo, pero un sistema que no tiene posibilidad de crecer, queda inservible cuando aumentan las necesidades de los usuarios. Por lo tanto, los sistemas deben diseñarse para satisfacer requerimientos cuando menos por cinco años y crecer si las necesidades, lo exigen.

En el estudio de un nuevo sistema se debe acumular información que ayude a comprender la forma en que se lleva a cabo el trabajo y así determinar cómo y dónde el sistema computarizado será benéfico para todos los usuarios y al mismo tiempo anticipar requerimientos que puedan ser necesarios. El sistema propuesto debe realizar lo siguiente:

1. El sistema debe de generar los reportes necesarios de la biblioteca, permitir la consulta de los datos disponibles, hacer las operaciones indispensables para el mantenimiento de las bases de datos.
2. Se requiere que el sistema permita a los usuarios de la biblioteca consultar todos los libros de forma rápida.
3. Se necesita que el sistema de apoyo a los maestros al momento de actualizar sus planes de estudio por medio de una consulta o un reporte de libros de alguna especialidad y así conocer los nuevos libros e incluirlos en la bibliografía.
4. El sistema debe emitir reportes con las estadísticas de uso de libros por grupo, libros más consultados y el total por materia de libros consultados.
5. Generar una constancia, para alumnos que por cualquier motivo dejan la escuela, donde indique que el alumno no debe libros a la biblioteca.

6. Se requiere que el sistema lleve el control de los libros que salen de la biblioteca en calidad de préstamo.
7. Es importante que el sistema realice los cálculos para cobrar una multa a todos aquellos usuarios que no devuelven los libros el día pactado.

2.13 Justificación del sistema

Debido al constante aumento de alumnos se ha dado un crecimiento general en la escuela, principalmente los libros de la biblioteca han aumentado de manera considerable, lo cual a su vez provoca que su manejo sea más complejo. El sistema manual utilizado está tardando mucho tiempo en generar la información necesaria para satisfacer las necesidades actuales de la escuela.

El sistema se desarrolla en Clipper por la facilidad que tiene este compilador para la obtención de programas directamente ejecutables a nivel DOS. Si el tiempo de ejecución de alguna parte de código fuera demasiado, Clipper aporta un sistema extendido que permite escribir programas y funciones en otros lenguajes, como C y ensamblador y llamarlos posteriormente transfiriéndoles parámetros y obteniendo como contrapartida los valores devueltos desde C. Además Clipper es un manejador de bases de datos eficiente, probado en bases de datos medianas en la plataforma MSDOS.

También se consideró la sugerencia del Jefe Académico para usar Clipper en la programación del sistema ya que este manejador de base de datos se enseña a los alumnos de programación y cuando el sistema requiera de una corrección, modificación o agregarle un módulo los alumnos podrían llevarlo a cabo.

Con el constante aumento de libros cada semestre y trabajando con el mismo sistema se perderá el control de los libros y será necesario contratar una persona que apoye al encargado en el mantenimiento a los libros y a la organización.

Con el nuevo sistema, antes de que salga un libro de la biblioteca primero se cargará a la cuenta del usuario y se archivará la credencial y el formato de préstamo. Para aquellos usuarios que por cualquier razón no llenan formato y tampoco dejan credencial, por ejemplo maestros, empleados, etc., también tendrán una clave de usuario; de tal manera que cuando se lleven un libro se pedirá tanto su clave como la del libro para introducir las al sistema, quedando el libro en asignado a una clave de usuario.

Tanto en préstamo como en la devolución de libros el encargado únicamente introducirá la clave y en la pantalla aparecerán los datos complementarios del libro, así se podrá comparar la información del libro físico con los de la base de datos. Con esta comparación será difícil que se devuelva o salga un libro diferente al que fue prestado.

Con el uso del programa se utilizarán aquellos libros que nunca han sido consultados ya que muchos de ellos, contienen temas de utilidad, pero al no contar con una herramienta que permita buscar temas, los usuarios solamente consultan los libros de la bibliografía proporcionada por el profesor.

2.14 Recursos disponibles

Dentro del análisis se definen los recursos necesarios para la implantación del sistema. Pero en este caso estos recursos ya se encuentran en la escuela, así que todo el esfuerzo de diseño se concentra en adaptar la aplicación a las características del sistema de cómputo existente. Sin embargo, cuando es necesario nuevo equipo de cómputo, la selección y adquisición del mismo produce una actividad más en el desarrollo del nuevo sistema. Se deben formular las especificaciones del sistema, recibir y revisar las propuestas de los proveedores de equipo de cómputo y hacer la selección adecuada. Esta última se centra alrededor de los costos de adquisición, operación, mantenimiento y características de procesamiento y comunicación.

Los recursos existentes donde se implantará el sistema es en un servidor con Novell 3.12 y se ejecutará en computadoras con procesador Intel 486, memoria RAM de 4 Mb, disco duro 100 Mb y una unidad de disco flexible de 3 1/2 de alta densidad.

3. DISEÑO GENERAL DEL SISTEMA AUTOMATIZADO

Selección del proyecto

Opciones del sistema

Diseño de la entrada y salida de datos

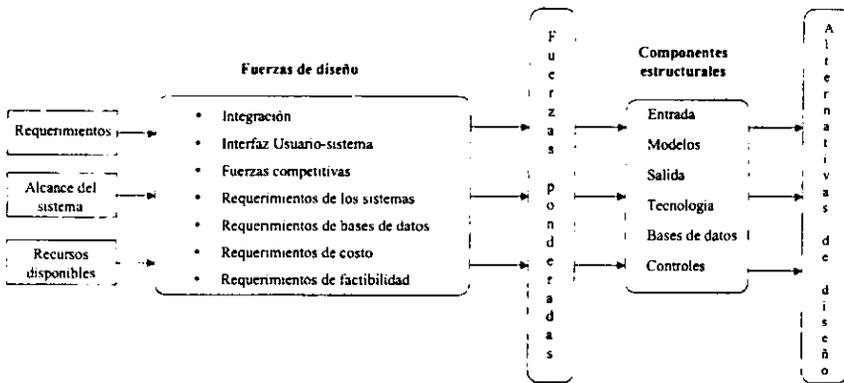
Diseño de los controles

Diseño de la base de datos

III. DISEÑO GENERAL DEL SISTEMA AUTOMATIZADO

Una vez concluida la fase de análisis se crean alternativas generales de diseño que satisfacen los requerimientos de los usuarios y las fuerzas del diseño se consideran dentro del proyecto del sistema.

A medida que se analizan los hechos de estudio y se revisan los hallazgos se ven más claros los conceptos de diseño. Estas ideas, se convierten en modelos de diseño empleando técnicas de modelación, como los diagramas de flujo de datos, los diagramas entidad relación y los diagramas de estructura.



³Figura 3.1

El proceso del diseño general de sistemas se observa en la figura 3.1. Conociendo los requerimientos de los usuarios, el alcance del sistema y los recursos disponibles. Las fuerzas de diseño se consideran y se ponderan para determinar su impacto sobre los componentes estructurales y de esta forma obtener alternativas de diseño. Se crean diferentes alternativas de diseño tomando en cuenta diferentes funciones de ponderación. Por ejemplo, una alternativa se puede dar mucho mayor peso a la integración, a la interfaz usuario-sistema y las fuerzas competitivas. Otra alternativa de diseño puede incluir un componente estructural de tecnología de avances recientes, mientras que otra sólo incluya la tecnología establecida.

La presentación de alternativas aumenta la probabilidad de que finalmente se implemente el diseño correcto, ya que en muchos casos los usuarios no saben lo que desean hasta que se les muestra.

³ Fuente: Diseño general de sistemas de información, John G. Burch.

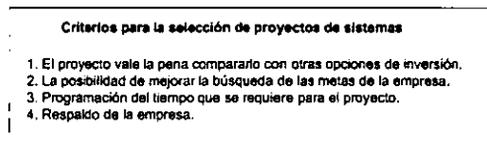
3.1 Selección de proyectos

Los proyectos tienen orígenes y motivos muy diversos. No todos ellos deben elegirse para estudios adicionales. Las razones que justifican todo estudio de sistemas, que parece solucionar un problema o traer mejoras, deben quedar muy claras. Los proyectos a estudiar deben considerarse desde un punto de vista de sistemas, de tal forma que se considere el impacto del proyecto propuesto sobre toda la organización. Los subsistemas de la empresa se encuentran interrelacionados e interdependientes, de forma que los cambios en uno ellos puede afectar a todos los demás. en otras palabras, no pueden contemplarse o seleccionarse de manera aislada de la organización. Existen cuatro criterios para la selección de proyectos, figura 3.2, el primordial será contar con el respaldo de la gente dueña de los sistemas, la disponibilidad de tiempo, es decir, todos los involucrados están dispuestos a ofrecer el tiempo necesario para la implantación del nuevo sistema o para el mejoramiento de alguno de los existentes.

Un tercer criterio para la elección reside en la posibilidad de mejorar la realización de las metas de la empresa, en otras palabras, dentro de los objetivos del proyecto debe estar contemplada la empresa y no desviarla de sus fines primarios.

Por último, para la elegir un proyecto deberá ser viable en función de los recursos y su capacidad y debe llegar a un acuerdo con la empresa respecto a las ventajas del proyecto de sistemas sobre cualquier otra alternativa de inversión. Conviene considerar a la proyectos como competidores por los recursos empresariales de tiempo, dinero y personal.

Una vez que el número de proyectos se ha reducido por la aplicación de los criterios vistos anteriormente, es todavía necesario determinar si estos proyectos seleccionados son viables, de acuerdo a tres principios básicos: operativo, técnico y económico. Este estudio sirve para recopilar datos relevantes y en base a ellos, deberá tomarse la decisión si procede un estudio de sistemas, el cual debe ser lo más conciso posible, realizándose simultáneamente varias actividades durante este estudio. Los objetivos del proyecto deben hacerse explícitos mediante la entrevista a las personas, el grupo o al departamento que lo propone.



¹Figura 3.2

Además, es útil contar con una revisión por escrito del trabajo, asociada al proyecto en cuestión. Los recursos para llevar a cabo los proyecto se analizan en relación con los tipos de viabilidad: técnica, económica y operativa.

¹ Análisis y diseño de sistemas, Kendall y Kendall.

Viabilidad técnica

Se debe investigar si los recursos técnicos usuales pueden actualizarse o complementarse, de tal manera que satisfagan la necesidad considerada. Sin embargo, si los sistemas existentes no pueden actualizarse, el siguiente paso a considerar será determinar si existe alguna tecnología que pueda satisfacer los requisitos. Es aquí donde la experiencia del analista de sistemas es de gran valor, ya que haciendo uso de sus conocimientos y de sus relaciones con otras empresas podrá solucionar el problema de viabilidad técnica. Por lo general la respuesta a si una tecnología se encuentra disponible y si llegaría a satisfacer las necesidades de los usuarios es afirmativa. luego entonces, el problema se convierte en otro, ahora de tipo económico. En la actualidad, prácticamente nada es imposible desde el punto de vista técnico. En consecuencia, la viabilidad técnica analiza lo que será práctico o razonable implantar. Algunos proyectos no requieren de esta evaluación. Por ejemplo, aquellos proyectos que proceden de una planificación formal de sistemas, los cuales ya han sido justificados por el equipo de planificación y la dirección de la empresa. De forma similar, la conveniencia de algunos proyectos puede estar predeterminada por provenir de una persona de alto rango en la empresa, como puede ser un vicepresidente o un directivo ejecutivo. Así, algunos proyectos pueden pasar directamente a la siguiente fase de estudio.

Viabilidad económica

También se conoce como viabilidad financiera, durante las primeras fases del proyecto es casi imposible calcular los costos en éstas etapas, ya que las necesidades los usuarios y las soluciones técnicas alternativas todavía han de identificarse. Sin embargo, tan pronto se hayan determinado los requisitos específicos y las soluciones, se continua con la valoración los costos y los beneficios de cada alternativa. Los recursos básicos que deben considerarse son: el costo de la realización integral del estudio del sistema, incluyendo el tiempo de los empleados que participarán en él; el costo del empleado para la empresa, el costo estimado del equipo y el costo estimando de los programas comerciales o de su desarrollo. Se tiene que tener la capacidad de establecer el valor de la inversión antes de comprometerse con el estudio del sistema completo. Si los costos a corto plazo no se compensan por las ganancias a largo plazo o no hay una reducción inmediata del costo de operación, entonces, desde el punto de vista económico, el sistema no será viable y el proyecto no debe trascender de esta etapa.

Viabilidad operativa

Suponemos que los recursos técnicos y económicos están disponible. Ahora la viabilidad operativa depende de los recursos humanos que participan durante la operación del proyecto. Esto se refiere al pronóstico de si, una vez instalado el sistema llegará a funcionar o a usarse. Si los usuarios están acoplados con el sistema actual y no le ven problema alguno, y no participan para perfeccionar o remplazarlo por otro nuevo sistema, es muy probable que la resistencia al cambio será enorme. En estas circunstancias, son muy pocas la oportunidades de que el sistema nuevo llegue a ser operativo. De manera alternativa, si

de los mismos usuarios surge el interés de contar con un sistema que sea más eficiente, entonces la oportunidad será mayor para que el sistema solicitado sea con el tiempo aceptado. En gran medida, la viabilidad operativa depende del tipo de interfaz que se elija.

Tomando en cuenta los programas educativos de las carreras, excluyendo a la del técnico en programación, las restantes no tienen materias adicionales donde se incluya el manejo de paquetes de computación. Para estas carreras su único contacto es al final por medio de dos materias básicas de computación por lo que la fuerza de diseño que se consideró fue la forma en que se comunica el usuario con el sistema, es decir, la interfaz usuario-sistema la cual se compone de dos elementos principales: el lenguaje de presentación que es parte de la relación entre la computadora y el hombre y el lenguaje de acción que caracteriza la parte del hombre y su relación con la computadora, ambos conceptos cubren el término de interfaz de usuario. Si existen varios tipos de interfaz, entre las cuales se encuentran: lenguaje natural, preguntas respuestas, manejo directo, dispositivos de señalamiento y reconocimiento y síntesis del lenguaje. Cada una de estas interfaces, cuenta con atributos especiales que las hacen únicas para aplicaciones particulares, para nuestro propósito se considera la interfaz dirigida por menús. Es necesario que con una capacitación mínima los usuarios puedan comunicarse con el sistema, ya que se cambia continuamente a las personas que manejan el sistema, generalmente son prestadores de servicio social.

3.2 Opciones del sistema

Cada vez que se evalúan los requerimientos de los usuarios se generan tres opciones básicas de diseño. Se puede recomendar que no se haga nada, que se modifique el sistema existente o que se diseñe un nuevo sistema.

Una pobre identificación de los requerimientos o la determinación de que no es factible desarrollar un sistema realmente significativo son razones suficientes para recomendar que se continúe trabajando con el sistema actual, es decir, que no se haga nada.

El nivel de desarrollo de sistemas de información que existe actualmente ha llegado a un punto en que las nuevas demandas de los usuarios con frecuencia requieren cambios relativamente pequeños, como puede ser la captura de un dato o algún formato de salida nuevo.

La opción del diseño de un nuevo sistema es la solución más compleja y difícil de implementar. Además, se debe tomar la decisión de si el sistema se va a desarrollar o si se va a comprar un sistema comercial.

En nuestro caso solamente se tienen dos opciones, hacer un nuevo sistema o no hacer cambios al sistema actual, ya que no existe sistema automatizado. Se decidió hacer el sistema porque el sistema manual usado hasta el momento no genera la información a

tiempo y por lo tanto no cubre los requerimientos de información que demanda la escuela. Por otro lado no se tienen recursos para adquirir un sistema comercial de biblioteca.

3.3 Diseño de la entrada y salida de datos

El proceso de entrada por lotes es el más antiguo y tradicional. Se reúnen los documentos originales y se entregan periódicamente a las personas que introducen los datos, quienes teclean dichos datos por medio de un dispositivo de entrada que los traduce a un formato comprensible por la computadora.

El soporte más común de introducción de datos por lotes fueron las tarjeta perforadas, han sido sustituidos por las estaciones de trabajo de teclado a disco y de teclado a cinta, los cuales transcriben los datos a discos magnéticos y cintas magnéticas, respectivamente. El registro de entrada de datos se ven en la pantalla mientras se teclea y se puede corregir, ya que los datos se sitúan en la zona de almacenamiento intermedio (buffer). En la figura 3.3 se observa el archivo final de entrada, obtenido por la fusión de la acción de varias estaciones de teclado a disco. En la figura A se observa la hoja con marcas ópticas, las cuales eliminan, en su mayor parte, la necesidad de teclear datos de entrada. El documento es leído directamente por un lector de marcas ópticas (OMR) o un lector de caracteres ópticos (OCR). Estos lectores son utilizados para actividades que requieren un elevado volumen de entradas.

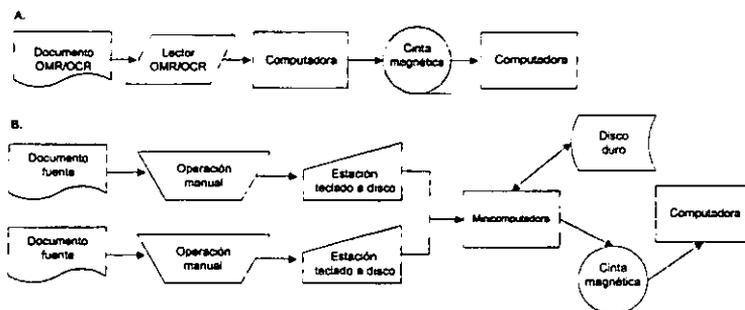


Figura 3.3

¹ Fuente: Análisis y diseño de sistemas. Kendall y Kendall.

El método de entrada en línea implica la captura de datos en el mismo lugar en donde se originan y se introducen directamente a la computadora. Actualmente la mayoría de los sistemas utilizan este método, el cual requiere de una pantalla de monitor y un teclado conectado directamente al sistema informático. El usuario introduce los datos cuando se originan. Con la tecnología actual, se puede eliminar gran parte de la intervención humana en la tarea de entrada de datos en línea. Por ejemplo, los centros comerciales usan lectores ópticos de caracteres y de código de barras, de esta forma eliminan, en gran parte, la necesidad de teclear datos por parte de los usuarios. En su lugar, lectores de láser leen el código de barras y envían los datos representados por dicho código directamente a la computadora para su proceso.

La entrada de datos es el punto de partida más importante de los sistemas de información, se deben especificar los métodos para capturar y validar la exactitud de los datos. Se debe comenzar por capturar sólo aquellos datos que en realidad son parte de la entrada. Los datos complementarios deben ser recuperados o calculados por el sistema. Existen varios tipos de datos que son considerados para proporcionarse como entradas cuando se procesan transacciones, los cuales se describen a continuación.

- Datos variables: son datos que cambian para cada transacción o toma de decisión. Por ejemplo, la identificación de cada usuario y libro cambia en cada solicitud de libros, por lo tanto debe formar parte de la entrada.
- Datos que el sistema puede recuperar: son datos almacenados y que son recuperados por el sistema con facilidad. Por ejemplo, al introducir la clave del usuario el sistema recupera el nombre, grupo, sexo, etc., o el nombre del autor, título de libro y la editorial al dar la clave del libro.
- Datos que el sistema debe calcular: Son resultados que pueden ser producidos indicando al sistema que utilice combinaciones de datos almacenados y proporcionados. Por ejemplo, la fecha de préstamo del libro se recupera del calendario de la computadora. La multa se calcula en base a la operación de fecha de préstamo del libro y fecha de devolución.

En el desarrollo de formas de entrada y pantallas es importante considerar que un buen diseño debe satisfacer los objetivos siguientes:

- Eficacia: significa que las formas y las pantallas de entrada satisfagan los propósitos específicos del sistema.
- Facilidad de uso: quiere decir que tanto las formas como las pantallas serán explícitas y no es necesario tiempo adicional para descifrar su contenido.

- Sencillez: se refiere a mantener en un mínimo los elementos indispensables que centren la atención del usuario.
- Atracción: implica que el usuario disfrutará del uso o tránsito a través de las formas y pantallas en la medida que los diseños sean más atractivos.

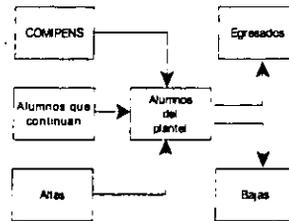


Figura 3.4

Cada año Comisión Metropolitana de Instituciones Públicas de Educación Media Superior. (COMIPENS) se encarga de realizar el concurso para los aspirantes a ingresar a las escuelas de nivel medio superior del gobierno. En un formato el aspirante enumera a las escuelas en forma ascendente, es decir, el número uno se maneja como primera opción, el dos corresponde a la segunda y así sucesivamente. Dependiendo del número de aciertos, el cupo en las escuelas y resultado en el sorteo, es como se ubican a los aspirantes a las escuelas. Por ejemplo, si existen 2000 aspirantes que seleccionaron al CETIS 92 como primera opción y con igual número de aciertos, pero la escuela solamente tiene un cupo de 900 alumnos, por lo tanto se sortean los lugares. Para aquellos que no obtuvieron lugar, son considerados en la escuela que marcaron como segunda opción. Posteriormente se publican y se envía el aviso de los aspirantes que ganaron un lugar en las escuelas respectivas. Por lo anterior la incorporación y salida de alumnos a la escuela se muestra en el figura 3.4.

Se tiene a los alumnos del plantel de todos los semestres, se agregan los enviados por COMIPENS, los que llegan de otro plantel y los que estaban con baja temporal y ahora se vuelven a inscribir. Se eliminan a egresados, aquellos que causaron baja temporal o definitiva y los que cambian de plantel. En los semestres pares no hay egresados ni concurso.

• Fuente: Propia.

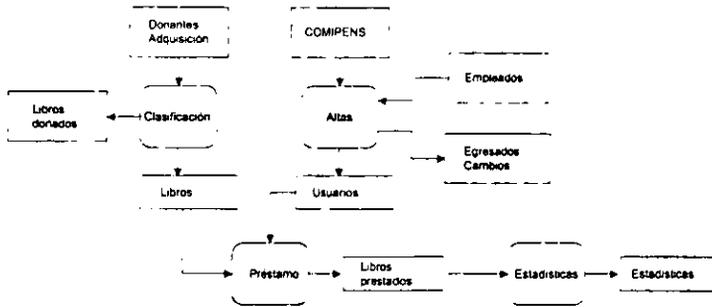
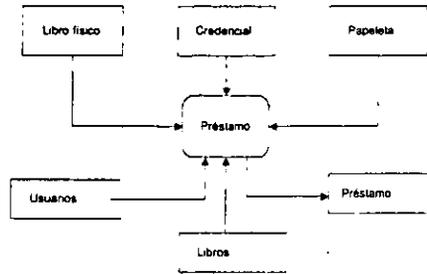


Figura 3.5

Se puede utilizar el diagrama de flujo de datos para modelar los procesos que representan el flujo de datos a través del sistema y los procesos o trabajos llevados a cabo por dicho sistema. Los símbolos más utilizados son: los cuadrados con esquinas redondeadas los cuales representan los procesos, los rectángulos con un lado abierto a un archivo o almacenamiento y los rectángulos a la entrada o salida de datos del sistema. Por ejemplo, la figura 3.5 muestra que la adquisición de libros y los donados, representan entradas de datos, posteriormente, se hace el proceso de clasificación para obtener el archivo de los libros. En otras palabras, se tiene a los alumnos del plantel de todos los semestres, se agregan los enviados por COMIPENS, los que llegan de otro plantel y los que estaban con baja temporal. Se eliminan a egresados, aquellos que causaron baja temporal o definitiva y los que cambian de plantel. En los semestres pares no hay egresados ni concurso.

Los diagramas de flujos de datos, también se puede utilizar para mostrar, de manera más detallada, las entradas necesarias en un proceso determinado. Por ejemplo, para el préstamo de libros hace uso de cinco entradas de datos: la credencial de la biblioteca, la cual se conserva hasta que se devuelve el libro y obtener el número de control del alumno, la papeleta para tomar la clasificación del libro, se usan dos archivos de bases de datos para recuperar los datos adicionales del libro y del usuario; y por último se verifican los datos contenidos en la papeleta con los datos que están impresos en el libro. Con la información de estos datos de entrada y realizando el proceso que se requiere para el préstamo, se genera un archivo llamado préstamo, figura 3.6, el cual contiene los datos más importante del usuario y del libro involucrados en el préstamo.

• Fuente: Propia.

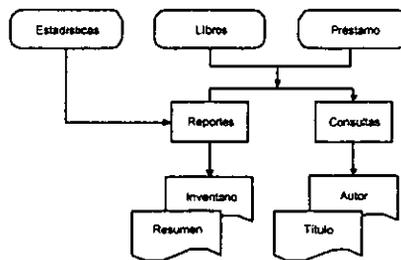


*Figura 3.6

La salida se refiere a los resultados e información generados por el sistema y debe alcanzar uno o más de los siguientes objetivos:

- Expresar la información relacionada con actividades pasadas, estado actual o proyecciones para el futuro de la empresa.
- Señalar los eventos importantes como son: problemas, advertencias, iniciar o confirmar una acción.

No se puede clasificar como adecuada a una salida atractiva o que haga uso de una tecnología a menos que satisfaga las necesidades de la organización y de los usuarios. De acuerdo con las circunstancias y contenidos, la salida puede ser impresa o presentada en pantalla. Sin importar si la salida es un reporte o un listado del contenido de un archivo, éste siempre es resultado de un proceso por computadora, así que la salida puede ser: un documento, reporte o mensaje.



*Figura 3.7

La figura 3.7 muestra de donde se obtiene la información para la salida. Los archivos de bases de datos del préstamo y de los libros son requeridos tanto para obtener listados como consultas, mientras que el archivo de estadísticas solamente se utiliza para listados.

• Fuente: Propia.

Las consultas de libros se pueden visualizar por los campos más significativos de la base de datos. En caso de las consultas de alumnos que adeudan libros se hace por clave del usuario. El reporte estadístico contiene información de los grupos de usuarios que más usaron el servicio de biblioteca.

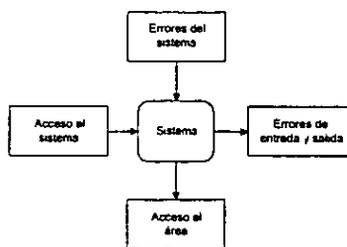
3.4 Diseño de los controles

Los controles son fundamentales para cualquier tipo de sistema de información basado en computadora, grande o pequeño, sofisticado o sencillo. Estos incluyen controles administrativos, de entrada, de procesamiento, de base de datos, de salida y de las operaciones de la computadora figura 3.8. Los controles proporcionan medios para:

- Asegurar que sólo usuarios autorizados tengan acceso al sistema.
- Validar los datos para comprobar su exactitud
- Determinar si se han omitido datos que son necesarios.

Se debe garantizar que los datos de entrada sean validados, para realizar esta tarea es importante tomar en cuenta las siguientes recomendaciones:

- Comprobar que se tienen todos los elementos necesarios, es decir, determinar si se han introducido realmente la información en todos los campos requeridos en la entrada.

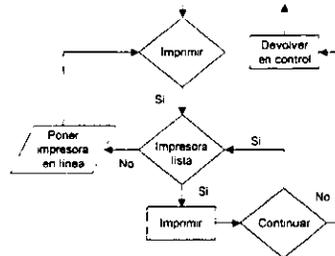


• Figura 3.8

- Hacer comprobaciones de límites e intervalos, para determinar si los datos de entrada de cada campo están comprendidos en el conjunto o intervalo correcto de valores definidos para dicho campo.

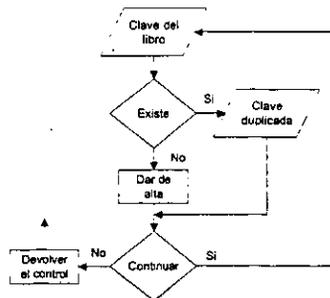
- Dígitos de autocomprobación, permiten describir los errores de introducción de datos en las claves primarias. Un dígito de comprobación es un número o caracter que se añade a un campo de clave primaria, se calcula aplicando una fórmula o un algoritmo.

Procedimientos para el manejo de errores: Son las acciones a seguir cuando se presentan errores inesperados. Por ejemplo, cuando una impresora se encuentra atascada, sin papel o simplemente no esta en línea ocurre un error. Lo mismo ocurre en un archivo dañado cuando el sistema intenta leer o escribir datos. Para estos tipos de errores, se tienen controles similares a los que se muestran en la figura 3.9. Si la impresora no esta lista se muestra el mensaje de error y da al usuario oportunidad de volver a mandar imprimir, en caso de que la impresora este lista se imprime y aún mientras se imprime se puede cancelar la impresión. Otro ejemplo de control se muestra en figura 3.10, sirve para evitar llaves duplicadas de libros.



• Figura 3.9

Los archivos de base de datos cada vez que sufren modificaciones se respaldan en el disco duro y en discos flexibles, como medida de seguridad, en caso de que se dañen los archivos se utiliza el respaldo del disco duro y se usa el respaldo de los discos flexibles en caso de que el disco duro sufra la pérdida de los datos, ya sea por algún virus informático o por daño físico.



• Figura 3.10

• Fuente: Propia.

El diagrama de flujo 3.11 muestra el control para el ingreso de datos. Si el campo no contiene información, pero es necesaria dicha información, entonces, no se puede avanzar en la captura. Por el contrario, se puede continuar con la entrada de datos, si el campo contiene información y además esta información es del tipo de datos adecuado; de lo contrario, nuevamente, no se avanza en la captura de datos.

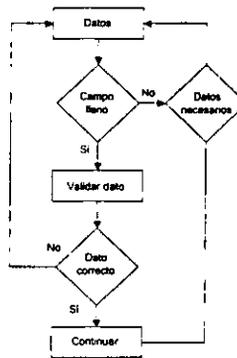


Figura 3.11

Se deben anticipar los errores que se puedan cometer al ingresar los datos al sistema o al solicitar la ejecución de ciertas funciones. Aunque exista la más mínima probabilidad de cometer un error, un buen diseño de sistemas ofrecerá los medios para detectar y manejar el error. De hecho un sistema bien diseñado debe tener la capacidad de modificar los errores sin interrumpir las demás actividades.

Además de los controles mencionados el sistema cuenta con una clave para tener acceso al sistema. Esta clave solamente la conocen las personas que manejan el sistema.

3.5 Diseño de la base de datos

El análisis de datos es una técnica que se utiliza para preparar un modelo de datos para implantarlo en forma de base de datos no redundante, flexible y adaptable. Para lograr esto, se hace uso del procedimiento llamado normalización. Se agrupan a los campos de datos en tablas que representan a las entidades y a sus relaciones. La teoría de la normalización está basada en la observación de que un cierto conjunto de relaciones tiene mejores propiedades en un medio de inserción, actualización y supresión que las que tendrían otros conjuntos de relaciones conteniendo los mismos datos. La razón de usar este procedimiento es asegurar que el modelo conceptual de la base de datos funcionará, esto no quiere decir que una estructura no normalizada no funcione, sino que puede tener problemas cuando los

* Fuente: Propia.

programadores de aplicaciones traten de modificar la base de datos. Los tres pasos para realizar la normalización son:

1. El primer paso de la normalización consiste en transformar los campos de datos en una tabla de dos dimensiones. Usualmente en este paso se necesita eliminar las ocurrencias repetidas de campos de datos, es decir todo atributo que se repite en realidad indica la presencia de una entidad independiente.
2. El segundo paso de la normalización es establecer las claves y relaciones con los campos de datos, es decir, se establecen los campos de datos que están relacionados con alguna parte de la clave. La división de la primera tabla normalizada en una serie de tablas en la que cada campo sólo depende de la clave completa, se le conoce como la segunda forma normalizada.
3. El tercer paso consiste en separar los campos de las segundas relaciones normales que, aunque dependan sólo de una clave, deben tener una existencia independiente en la base de datos. Una entidad está en tercera forma normal cuando los valores de sus atributos no clave no son dependientes de ningún otro atributo no clave.

Si se considera la información contenida en los libros de la biblioteca, de acuerdo a la figura 3.12. Cada libro se le debe asignar cinco temas de acuerdo al contenido del libro, entonces, en la base de datos se encuentran cinco registros con información repetida, esta se le conoce como grupos de repetición. Estos se deben de quitar y crear un archivo o relación aparte que contenga la información repetida, y de esta forma dejar un solo registro para cada libro. La figura 3.13 muestra las relaciones nuevas.

CLASI_LIB	TITULO	TEMAS	AREA	CLAVE
AA00BB00	Sistemas de información	Diseño de las entradas	Computación	384
		Selección del hardware	Computación	384
		Diseño del diálogo en línea	Computación	384
		Herramientas CASE	Computación	384
		Análisis estructurado	Computación	384
RR25II91	Lineamientos para la biblioteca	Las partes de libro	Biblioteca	025
		El catálogo público	Biblioteca	025
		Tipos de encuadernación	Biblioteca	025
		Estadísticas del servicio	Biblioteca	025
		Reglamento	Biblioteca	025

Figura 3.12

La segunda forma normal se alcanza cuando un registro está en la primera forma normal y cada campo depende totalmente de la llave del registro, tanto en el almacenamiento como en la recuperación. En otras palabras se busca la dependencia funcional: un campo es

• Fuente: Propia.

funcionalmente independiente si su valor está asociado de manera única con un campo único. Por ejemplo, en el control vehicular tratan de garantizar que a un vehículo específico se le asigne un número de placa identificada de manera única; en número de serie de un vehículo está asociado con uno y solo un número de placa. Así, si se conoce el número de serie, se puede determinar el número de placa del vehículo.

CLAS_LIB	TITULO	AREA	CLAVE
AA00BB00	Sistemas de información	Computación	384
RR25I191	Linkamientos para la biblioteca	Biblioteca	025

CLAS_LIB	TEMAS	AREA	CLAVE
AA00BB00	Diseño de las entradas	Computación	384
AA00BB00	Selección del hardware	Biblioteca	025
AA00BB00	Diseño del diálogo en línea		
AA00BB00	Herramientas CASE		
AA00BB00	Análisis estructurado		
RR25I191	Las partes de libro		
RR25I191	El catálogo público		
RR25I191	Tipos de encuadernación		
RR25I191	Estadísticas del servicio		
RR25I191	Reglamento		

•Figura 3.13

La tercera forma normal se alcanza cuando se quitan las dependencias transitivas de un diseño de registro. Para comprender mejor este concepto se debe considerar que:

- A, B y C son tres datos en un registro
- Si C es funcionalmente dependiente de B y
- B es funcionalmente dependiente de A,
- Entonces C es funcionalmente también de A.
- Por lo tanto existe una dependencia transitiva.

• Fuente: Propia.

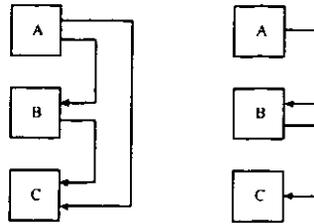


Figura 3.14

En la tercera forma normal se quitan las dependencias transitivas dividiendo la relación en dos relaciones separadas, tal como se muestra en la figura 3.14. Estas dependencias se deben evitar ya que los datos se pueden perder de manera inadvertida cuando la relación está oculta, en otras palabras, si se quita A, también se quita B y C.

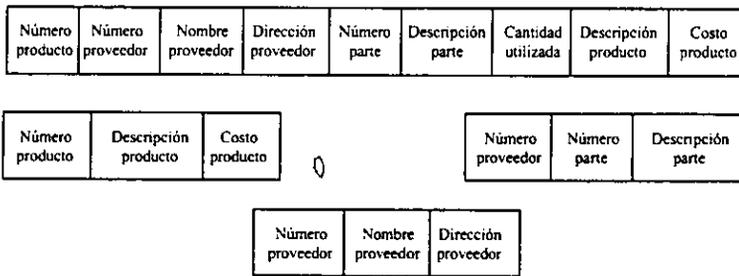


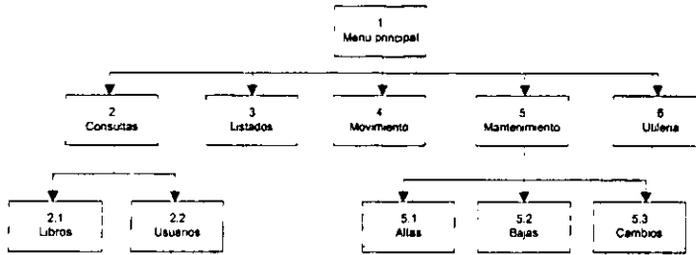
Figura 3.15

Nuestro registro de libros se encuentra en la segunda forma, y puesto que no contiene dependencias transitivas, también está en la tercera forma. Por lo tanto, se utiliza otro ejemplo para comprender mejor la eliminación de las dependencias transitivas, es necesario observar la figura 3.15 en donde PARTE depende de PRODUCTO, y PROVEEDOR depende de PARTE si se elimina un cierto producto de la base de datos, también retiran las partes y los proveedores, los cuales podrían estar asociados con otros productos. Para quitar las dependencias transitivas se crean los siguientes registros.

Los registros adicionales dan mayor flexibilidad para cubrir los requerimientos futuros, además de que eliminan los problemas que acarrea los registros con dependencia transitiva.

⁴ Análisis y diseño de sistemas de información. James A. Send.

Después de realizar todos los diseños anteriores es importante tener un panorama general de los módulos del sistema, es común que se utilice la tabla visual de contenido (VTOC), figura 3.16, esta tabla es un diagrama jerárquico que permite localizar un módulo del programa existente dentro del sistema.



*Figura 3.16

1. El menú principal es el programa encargado de realizar las llamadas a los módulos 2, 3, 4, 5 y 6. Además el control total del sistema lo realiza este programa.
2. Módulo que llama a dos submódulos, 2.1 y 2.2, para hacer las consultas tanto de libros y usuarios.
3. Módulo que sirve para llamar a los programas encargados de generar los reportes que el sistema emitirá.
4. Módulo para llamar a los programas que permiten hacer el préstamo y devolución de los libros.
5. La tarea de este módulo es llamar a otros submódulos para hacer los cambios, altas y bajas de libros y usuarios.
6. Módulo encargado de llamar a los programas necesarios para crear índices, actualizar o adicionar información a las bases.

* Fuente: propia.

4. DISEÑO DETALLADO DEL SISTEMA AUTOMATIZADO

Características de las pantallas

Entrada de datos

Diseño de los programas

Salida de datos

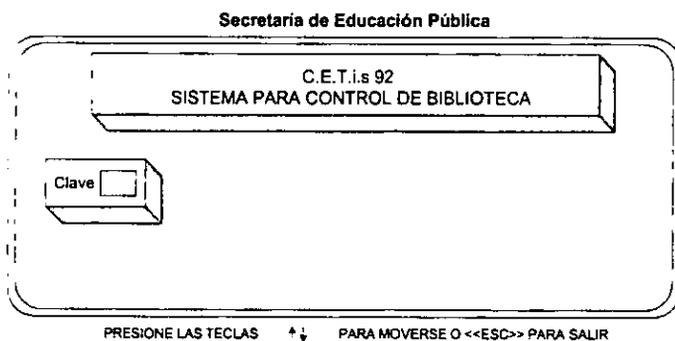
Diseño de archivos

IV. DISEÑO DETALLADO DEL SISTEMA AUTOMATIZADO

Uno de los objetivos de la fase del diseño detallado de sistemas es preparar especificaciones precisas para la fase de implementación del sistema, otro de los objetivos principales es diseñar un sistema con el cual los usuarios se sientan confiados y complacidos al trabajar. Este objetivo se logra considerando la interfaz usuario-sistema y los factores humanos. El diálogo o interfaz entre el usuario es la forma en la que el usuario interacciona con el sistema de cómputo y con la aplicación. Si un sistema es difícil de usar debido al diálogo, los usuarios tenderán a evitar el sistema, aunque pudieran ser más productivos si lo usaran.

Al preparar los elementos específicos acerca del contenido y formato de la entrada se consideró a las personas que recopilan e introducen los datos, y así determinar los formatos y pantallas de entrada más adecuados para estos usuarios. Puesto que los sistemas en línea proporcionan varias opciones de entrada y procesamiento a los usuarios, se requiere un método para mostrar a los usuarios las alternativas disponibles. Los menús cumplen este propósito. Los diálogos que utiliza este método de interrelación son conducidos por menú.

De acuerdo a los puntos planteados se diseñó las pantallas utilizando menús con mensajes para moverse por el sistema. Los menús reducen la necesidad de capacitación y memorización sintáctica y aumenta el conocimiento semántico relevante para las tareas de los usuarios. En otras palabras, la conversación entre el usuario y el sistema es conducida por menús. Un menú es una lista de funciones disponibles en el sistema, los cuales se muestran en el monitor de modo que el usuario puede elegir entre ellas.



*Figura 4.1 Programa menu.prg.

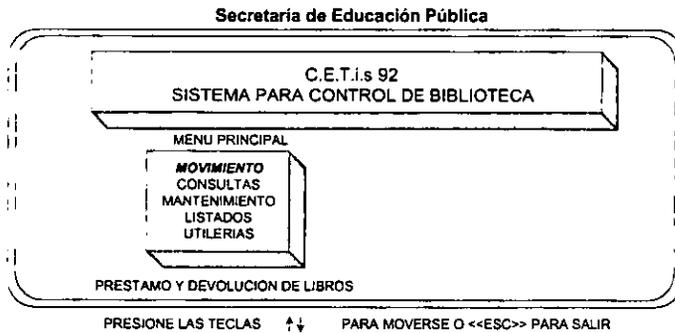
4.1 Características de las pantallas

La pantalla del menú principal tiene las ventanas necesarias para que los usuarios puedan navegar por todo el programa. Se recomienda que la construcción de las pantallas se realice de acuerdo a los siguientes puntos:

• Fuente: Propia.

- Ventana de título: identifica el título de la pantalla, la función a desarrollar o la aplicación en ejecución.
- Ventana de instrucciones: le dice al usuario cómo introducir datos, elegir un procedimiento o salir del sistema.
- Ventana principal de texto: es la porción más grande de la pantalla se reserva para la captura de datos o menús.
- Area de navegación y menús: instruye al usuario sobre cómo moverse en los menús.

Por ejemplo, cuando se ejecuta el programa principal del sistema, se observará por monitor el menú principal o la pantalla de inicio mostrada en la figura 4.1. Se observan mensajes para operar el sistema, flecha arriba, abajo para moverse. Para controlar el acceso de usuarios al sistema, se cuenta con una clave de entrada, al momento de teclear correctamente dicha clave se muestra la pantalla principal. Tal como se ve en la figura 4.2.



*Figura 4.2 Programa menú.prg.

La figura 4.3 muestra el desplegado de los menús, por ejemplo, si se selecciona la opción de consultas en el menú principal se muestra un submenú a la derecha y a la misma altura de la opción seleccionada del menú principal. Después se eligen usuarios del submenú y se despliega el último submenú.

Con la tecla ESC se puede volver al menú principal, pero ahora se observaría el borrado de la ventana de los submenús hasta llegar al menú principal. Si estando en el menú principal se tecldea de nuevo "ESC" se abre la ventana de salida. Esto se ve en la pantalla que se muestra en la figura 4.4. Cuando se selecciona la opción NO se devuelve al usuario al menú principal, de lo contrario se regresa la línea de órdenes del sistema operativo.

• Fuente: Propia.

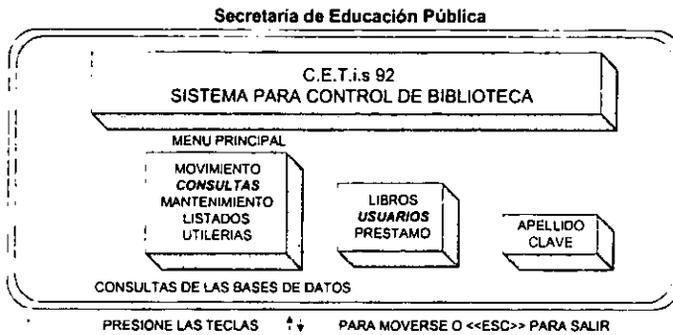


Figura 4.3 Programa menu.prg.

Los reportes tienen un control para evitar los errores de la impresora. Así que antes de enviar a imprimir cualquier tipo de reporte, se muestra el mensaje por pantalla, figura 4.5, para que prepare la impresora. El siguiente fragmento de programa es el que realiza este tipo de control.

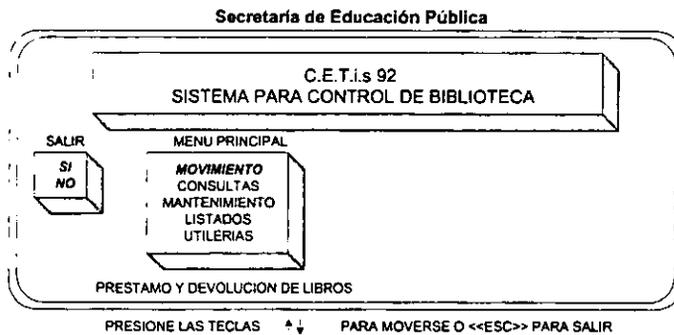


Figura 4.4 Programa menu.prg.

```
SET COLOR TO RB-+RB
@ 0.0,24,79 BOX *0000000000
SET COLOR TO W-+RB
@ 0.0 TO 24,79 DOUBLE
SET COLOR TO GR-+RB
```

```
SET COLOR TO R-+R
@ 0.0,24,79 BOX *0000000000
SET COLOR TO GR+R
@ 2,2 TO 22,77 DOUBLE
```

Fuente: Propia.

```

@ 4,18 TO 7,62 DOUBLE
@ 8,18 TO 11,62 DOUBLE
@ 5,19 SAY ' VERIFIQUE QUE LA
IMPRESORA TENGA PAPEL '
@ 6,19 SAY ' Y SE ENCUENTRE EN
LINEA '
@ 9,19 SAY ' NOTA: EL PAPEL DEBE DE
SER TAMAÑO CARTA'
@ 14,10 SAY 'EJEMPLO:'
@ 13,29 SAY 'AAAAAAA 132 AAAAAA'
@ 14,29 TO 19,48
@ 21,38 SAY 'presione alguna tecla para
continuar...'
wait "
TECLA = LASTKEY()
IF TECLA = 27
CLOSE DATABASES
RESTORE SCREEN FROM PA
RETURN
ENDIF
IF !SPRINTER()
SET COLOR TO B/B
@ 9,20,15,59 BOX '*****'
SET COLOR TO W~/B
@ 9,20 TO 15,59 DOUBLE
SET COLOR TO W~/B
@ 12,23 SAY 'LA IMPRESORA NO ESTA
PREPARADA'
@ 13,23 SAY ' VERIFIQUE LAS CONEXIONES'
@ 21,38 SAY 'presione alguna tecla para continuar...'
wait "
CLOSE DATABASES
RESTORE SCREEN FROM PA
RETURN
ENDIF

```

El sistema no se bloquea si ocurre algún error de impresión, el programa tiene la opción de corregir de los siguientes tipos de errores: impresora desconectada, falta de papel o impresora fuera de línea. Si ocurre el error, se muestra el aviso, similar al de la figura 4.5. para que el usuario lo corrija.

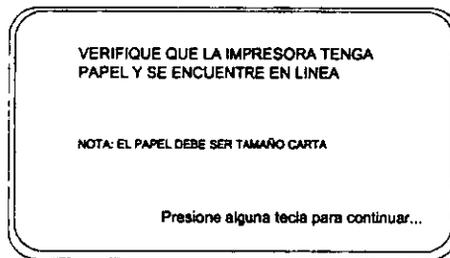


Figura 4.5 Programa rep_iv.prg.

4.2 Entrada de datos

Para dar de alta los libros, se diseñó la pantalla de la figura 4.6, cuando se introduce la clave se muestra automáticamente el nombre que corresponde a dicha clave, se introducen los datos que son requeridos, mismos que se toman principalmente de la portada y del reverso de la portada del libro, aunque también pueden aparecer en la anteportada, la contraportada o el colofón de la obra. Cuando se realiza la captura de libros, el bibliotecario ya tiene una

papeleta con datos bibliográficos, de control y de identificación, que previamente los obtuvo tomando en consideración los siguientes puntos:

Autor: El nombre del autor se escribe tal y como aparece en la portada del libro. No se incluirán los títulos o créditos profesionales. Es común que aparezcan dos o tres autores de una obra; si existieran más, se registran sólo tres. Cuando no aparece el nombre del autor personal, pero sí el de una institución, sociedad u organismo, entonces el autor es corporativo y se registra en el espacio correspondiente al autor, en la forma como esté escrito en la portada del libro. En caso de que no apareciera ningún autor, se dejará en blanco el espacio correspondiente en la papeleta.

Título: El título se escribe en la papeleta exactamente como aparece en la portada. En caso de que exista un subtítulo, éste también se anota completo, separándolo del título con dos puntos.

Lugar: Se refiere a la ciudad en la que se encuentra ubicada la casa editora o impresora del libro. Si aparecen citados dos o más lugares y entre ellos México, se anota México; de lo contrario, se toma sólo el primero.

ALTAS DE LIBROS

CLASIFICACION	020J85TU
AREA	COMPUTACION
AUTORPERSONAL	
EDITORIAL	
AUTORCORPORATIVO	
NUMERO DE PAGINAS	
EJEMPLARES	
AÑO DE PUBLICACION	
LUGAR DE PUBLICACION	
CLAVE ISBN	
TEMA 1:	
TEMA 2:	
TEMA 3:	
TEMA 4:	
TEMA 5:	
DATOS CORRECTOS (SN) 8	

Figura 4.6 Programa alt_lib.prg

Editor: Es el nombre de la institución o empresa responsable de la publicación de la obra. Se escribe completo y sin abreviar. En caso de que aparezcan dos o más editoriales, se anota sólo la primera. Se excluyen términos como “compañía”, “sociedad anónima” o “editorial”.

Año: se refiere al año en que se publicó el libro. Comúnmente se encuentra en el reverso de la portada. Puede aparecer año de edición, año de reimpresión y/o año de copy right o (derechos de autor). Si se encuentran los dos primeros, se anotan ambos en ese orden.

• Fuente: Propia.

separados por punto y coma (;). Cuando se tenga el año de copy right, se anota también, precedido por una "c". Si se encuentran varios años se toma el más reciente.

Páginas: se anota el número que aparece en la última página del libro. Si la obra consta de más de un volumen, se indicará el número total de volúmenes con la abreviatura "v". En caso de que la obra tenga dos numeraciones distintas, se anotarán ambas.

El número normalizado internacional que se le asigna a cada título (ISBN): Se localiza al reverso de la portada de algunos libros, si éste fuera el caso se anotará también en el espacio reservado.

Por último se cuenta con cinco campos de temas, la persona que clasifica el libro revisa el contenido del mismo y determina los cinco temas más completos que trata el libro.

Los libros tienen un número de clasificación asignado de acuerdo al tema que traten y con base en el sistema de clasificación decimal de Dewey, este número tiene tres dígitos enteros y, cuando la clasificación es más específica, uno o varios decimales, mismo que funciona como clave o llave. Como dato de control, se deja un campo en la base de datos libros.dbf para anotar, ya sea el número de oficio, circular o acta con lo cual se asignan de manera oficial los libros a la biblioteca. Los datos de la papeleta agilizan la captura de libros. En la figura 4.6 se tiene a la pantalla de captura de libros.

En la pantalla de préstamo de libros el sistema pide los datos sombreados, es decir, el número de control del alumno y la clasificación del libro, los datos adicionales los recupera el sistema de las bases de datos de libros y usuarios.

PRESTAMODELIBROS

CLAVE DEL ALUMNO	71512872
NOMBRE	CRUZ DIAZ ANGELICA
FECHA DE PRESTAMO	12-AGO-01
FECHA DE DEVOLUCION	14-AGO-01
GRUPO	6A
TURNO	M
CLAVE DEL LIBRO	0205785M
TITULO DEL LIBRO	NOVELL NETWORK VER 4
NUMERO CONSECUTIVO	001

DATOS CORRECTOS (SA)

Figura 4.7 Programa pre_lib.prg.

La entrada de solicitud de préstamo de libros a domicilio se muestra en la figura 4.7. La información al sistema se introduce por medio del teclado con tres validaciones; en primer lugar se introduce la clave del usuario que está solicitando el préstamo de libros y la clave del libro solicitado; se verifica que existan las dos claves en los archivos de base de datos. Si ocurre un problema, como una entrada de un número de clave inválido, entonces se produce un mensaje de aviso y la operación se repite con un nuevo número de clave o

• Fuente: Propia.

finaliza la operación si se quiere. Cuando un usuario solicita un libro el sistema verifica que dicho usuario no tenga más de dos libros prestados con anterioridad. En caso afirmativo el sistema manda el correspondiente mensaje y termina la operación.

DEVOLUCION DE LIBROS

CLAVE DEL LIBRO	0205788N
TITULO DEL LIBRO	NOVELL NETWORK VER 4
NUMERO CONSECUTIVO	001
FECHA DE PRESTAMO	12-AGO-01
FECHA DE DEVOLUCION	14-AGO-01
GRUPO	6A
TURNO	M
CLAVE DEL USUARIO	71512872
NOMBRE	CRUZ DIAZ ANGELICA

DATOS CORRECTOS (S/N)

Figura 4.8 Programa dev_lib.prg

Cuando los usuarios devuelven libros se despliega una pantalla igual a la de la figura 4.8, se utiliza la base de datos pre_lib.dbf. Como entrada solamente se pide la clave del libro, los datos complementarios son accedados por el sistema de este archivo, después se actualiza el archivo de estadísticas con los campos necesarios y por último se elimina el registro del archivo.

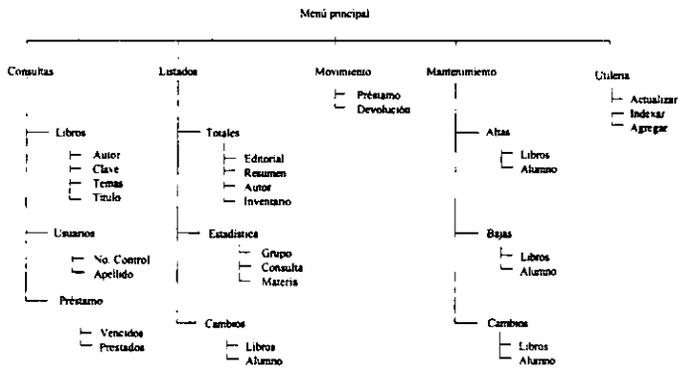


Figura 4.9

4.3 Diseño de los programas

La figura 4.9 muestra todas las opciones del sistema. Cada uno de los nombres localizados al final de las ramas del árbol representan a cada uno de los programas del sistema. Se observan los 29 programas, incluido el programa principal encargado del control de ellos.

En el diagrama funcional de la figura 4.10 se puede ver la entrada, proceso y salida del modulo llamado movimiento. Como entrada se tienen las bases de datos de usuarios y libros, el proceso lo hace el programa préstamo de libros y la salida es una actualización del archivo préstamo, mismo que pasa a ser entrada para el proceso de devolución de libros. Este proceso actualiza el archivo de estadísticas.



Figura 4.10

Las especificaciones para todos los programas del sistema préstamo de libros se describen bajo la forma de inglés estructurado, tal como se muestra en el siguiente ejemplo:

Abrir las bases de datos

```

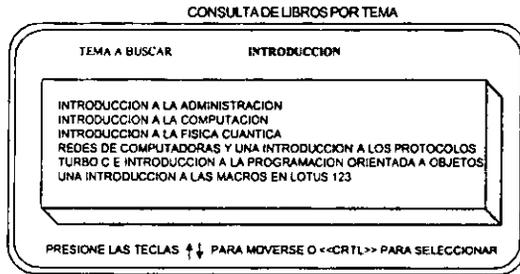
WHILE se requiera prestar libros
  pedir la clasificación del libro;
  IF la clasificación es válida
    pedir el número de control de alumno;
    IF número de control es válido
      prestar el libro;
      actualizar el archivo préstamo;
    ELSE
      mostrar el mensaje " el usuario no se encuentra";
    ENDIF;
  ELSE
    mostrar el mensaje " libro no se encuentra";
  ENDIF;
ENDDO.
  
```

³ Diseño general de sistemas de información, John G. Burch.

4.4 Salida de datos

En el diseño detallado de la salida se define el contenido y el formato de todos los documentos impresos, reportes y pantallas que serán producidos por el sistema, sin embargo, la calidad de la salida solo será tan buena en la medida que la entrada lo sea.

En la figura 4.11 se observa la pantalla de las consultas de libros se introduce cualquier tema o cualquier caracter y el sistema automáticamente coloca al puntero en la primer letra que coincida en la base de datos. Se puede mover el apuntador a otro tema si se quiere. Después de seleccionar el tema adecuado, se muestra la información correspondiente del libro que contiene el tema.



*Figura 4.11 Programa cos_lib.prg.

El formato de liberación o carta de no adeudo, figura 4.12, se imprime en tamaño carta. Este formato se obtiene al revisar la base de datos de libros prestados, si el usuario no se encuentra en dicha base, se le entrega la constancia de no adeudo de libros a la biblioteca, de lo contrario deberá devolver el libro o pagar el precio actual del libro que adeuda, para poder entregar dicha constancia.

El reporte del inventario se muestra en la figura 4.13. Se imprimen las cantidades por área o tema y el total de libros. La impresora debe hacer el salto de hoja cada vez que cambie el nombre del área, en la figura se muestran varias áreas, esto es únicamente para el ejemplo del total de libros.

* Fuente: Propia.

CENTRO DE ESTUDIOS TECNOLOGICOS Industrial y de servicios No. 92
Departamento de biblioteca

CARTA DE NO ADEUDO DE LIBROS

Después de haber revisado la base de datos correspondiente, se constató que el alumno(s):

IGNACIO MARTINEZ GUZMAN

No adeuda material bibliográfico a la biblioteca.

ATENTAMENTE

ENCARGADO DE LA BIBLIOTECA

Nombre y firma

*Figura 4.12. Constancia de no adeudo de libros

Los reportes de estadísticas muestran cantidades de libros por materia que fueron consultados por semestre Figura 4.14.

CENTRO DE ESTUDIOS TECNOLOGICOS Industrial y de servicios No. 92
DEPARTAMENTO DE BIBLIOTECA
Inventario de libros

COORDINACION ESTATAL: ESTADO DE MEXICO			05/25/01
TITULO	AUTOR	EDITORIAL	EJEMPLARES
AREA: COMUNICACION			
RELACIONES HUMANAS	BELO DAVID	DGETI	3
COMUNICACION ORAL	RANGEL INOJOSA MONICA	TRILLAS	4
TOTAL POR AREA:			7
AREA: COMPUTACION			
VISUAL BASICA 6	GREG PERRY	PRENTICE HALL	7
INTRODUCCION A PASCAL	RODNAY ZAKS	MARCOMBO	3
TOTAL POR AREA:			10
TOTAL:			17

*Figura 4.13

* Fuente: Propia.

En el reporte de libros prestados solamente se imprimen los cuatro campos que se observan en la figura 4.15. Esta información sirve para identificar al usuario que tenga en su poder algún libro. Se imprime el cuadro de la figura 4.16 para conocer la consulta de libros por grupo.

CENTRO DE ESTUDIOS TECNOLOGICOS Industrial y de servicios No. 92
DEPARTAMENTO DE BIBLIOTECA
 Consulta de libros por tema

							05/25/01
MATERIA	1	2	3	4	5	6	TOTAL
MATEMATICAS	500	200	250	450	235	450	2100
FISICA	800	450	147	126	711	581	2823
QUIMICA	257	285	548	254	247	265	1746
BIOLOGIA	125	147	1253	38	587	447	2595
HISTORIA	1587	1254	125	125	264	457	3812
COMPUTACION	68	25	145	514	144	145	1041
ADMINISTRACION	41	145	247	125	457	125	1140
TOTAL	3378	2494	2831	1638	2645	2471	15257

Figura 4.14

CENTRO DE ESTUDIOS TECNOLOGICOS Industrial y de servicios No. 92
DEPARTAMENTO DE BIBLIOTECA
 Listado de libros prestados

			05/25/01
CVE_LIB	TITULO	NOMBRE	
020S859101	DISEÑO DE SISTEMAS DE INFORMACION	ANGELICA ALBA MARTINEZ	
02501R5791	LINEAMIENTOS PARA LA ORGANIZACION	MARIA ELENA MARTINEZ GIRON	
020G785212	DIENO DE BASES DE DATOS	GISELA PIÑA ALBA	

*Figura 4.15

Para obtener datos estadísticos confiables, es necesario llevar un registro diario y sistemático de los aspectos de interés. En la biblioteca se usa un cuaderno exclusivo para concentrar diariamente las anotaciones de los usuarios atendidos, es decir, el número de personas que asistieron a la biblioteca para utilizar algún servicio. A continuación, en la figura 4.17, se observa la información solicitada a los usuarios.

• Fuente: Propia.

CENTRO DE ESTUDIOS TECNOLOGICOS industrial y de servicios No. 92
DEPARTAMENTO DE BIBLIOTECA

05/25/01			
NOMBRE	SEMESTRE	GRUPO	ESPECIALIDAD
ANGELICA ALBA MARTINEZ	5	A	COMPUTACION
MARIA ELENA MARTINEZ GIRON	3	A	ADMINISTRACION
GISELA PIÑA ALBA	3	E	COMPUTACION
PERLA SANCHEZ GARCIA	5	B	ADMINISTRACION

*Figura 4.16

Con el sistema el formato de la figura 4.17, queda obsoleto, ya que todos esos datos anotados en dicho formato, ahora se guardan en el archivo de base de datos de las estadísticas desde el momento en que el usuario solicita el libro para consulta interna o el préstamo a domicilio. Posteriormente se ejecutan los programas de reportes del módulo de estadísticas para generar el informe requerido.

CENTRO DE ESTUDIOS TECNOLOGICOS industrial y de servicios No. 92
DEPARTAMENTO DE BIBLIOTECA

05/25/01			
NOMBRE	SEMESTRE	GRUPO	ESPECIALIDAD
ANGELICA ALBA MARTINEZ	5	A	COMPUTACION
MARIA ELENA MARTINEZ GIRON	3	A	ADMINISTRACION
GISELA PIÑA ALBA	3	E	COMPUTACION
PERLA SANCHEZ GARCIA	5	B	ADMINISTRACION

*Figura 4.17

Las salidas impresas pueden utilizar papel, sin embargo, ocupa mucho volumen y requiere espacio considerable para su almacenamiento. Para resolver el problema se utiliza la película como soporte de almacenamiento de salida. También el video está aumentando su uso.

* Fuente: Propia.

- El microfilm es un carrete de película fotográfica que se utiliza para grabar información de salida en tamaño reducido.
- La microficha es una hoja única de tipo especial de película donde pueden almacenarse muchas páginas de información reducida.
- El video presenta la información en línea mediante un dispositivo de imagen como una terminal de tubo de rayos catódicos o el monitor de una computadora.

El uso de la película presenta problemas, ya que sólo puede grabarse o leer la información con equipo especial. Mientras que el video pierde la información cuando el sistema deja la pantalla y la recupera cuando se vuelve a visualizar. Si se requiere una copia permanente de la información, resulta más adecuado el papel y la película.

La mayoría de los sistemas manejadores de bases de datos y los lenguajes de programación actuales incluyen generadores de aplicaciones para el desarrollo rápido de prototipos de sistemas completamente funcionales. Generalmente durante el diseño de bases de datos se creó una base de datos, estos datos almacenados pueden utilizarse para hacer prototipos de salidas y pantallas. Esta cualidad está disponible en sistemas manejadores de bases de datos para microcomputadoras.

4.5 Diseño de archivos

El archivo maestro de libros contiene los datos relacionados con cada uno de los libros con que cuenta la biblioteca. Por medio de este archivo se puede recuperar la información acerca del libro, por ejemplo, autores de libros, títulos de libros, etc. La clave del libro es llave del registro. Este archivo se utiliza en la captura, consulta y en la generación de reportes.

Archivos de temporales

En el sistema se establece un archivo temporal que contiene una relación clave-carrera. Con este archivo se pretende que cuando se capture la clave de un nuevo libro el sistema informe el nombre de la carrera en relación a la clave dada. Otro archivo de este tipo se genera cuando se hace la consulta de libros por tema. Del archivo "temas.dbf" se selecciona las claves de libros que cumplen con el nombre del tema solicitado, estas claves se guardan en un archivo temporal para después relacionarlo con la base "libros.dbf" para recuperar los datos adicionales del libro.

Archivo de transacciones

Existen dos archivos de transacciones, el archivo de libros prestados guarda información de los libros que salen de la biblioteca por préstamo a domicilio. Se conservan los registros hasta que el libro es devuelto. El archivo coli_aux.dbf contiene claves de libros, necesarios para efectuar una relación con el archivo lib_bib.dbf cuando se hacen consultas por tema. El número de claves será igual al número de temas.

El archivo de estadística recopila los datos: clave del libro, grupo y semestre, necesarios para llevar a cabo la estadística.

Archivo maestro de usuarios tiene los datos de las personas autorizadas para llevar libros prestados a domicilio. Este archivo se encuentra ordenado por clave del usuario. Se utiliza al momento de prestar los libros para verificar que la clave sea vigente.

ESTADIST.DBF	PRE_LIB.DBF	LIBROS.DBF	USUARIOS.DBF
FEC_DEV	NOM_USU	NUM_AREA	CLA_USU
FEC_PRE	TIT_LIB	AREA	NDM_USU
CLA_USU	FEC_DEV	TITULO	SEM
CLASI_LI	FEC_PRE	AUT_PER	SEXO
NUM_AREA	CLA_USU	EDITORIAL	GPO
GPO	CLASI_LI	AUT_COR	TURNO
SEM	NUM_AREA	NO_PAG	EDAD
TURNO	GPO	CLASI_LI	COLONIA
SEXO	SEM	CANTIDAD	CALLE
	TURNO	ANO_PUB	NO_INT
	SEXO	LUGAR_PUB	NO_EXT
	NUM_CONS	CLA_ISBN	CP
			MUN
			EDO

*Figura 4.18

La figura 4.18 muestra los campos de las bases de datos que está utilizando el sistema para obtener los datos y procesarlos y generar la información tanto impresa como la que se visualiza en la pantalla o monitor de la computadora. También se observa que los archivos PRE_LIB.DBF Y ESTADIST.DBF son los llamados de transacciones por el constante movimiento de registros que se realiza en ellos. Los otros dos son considerados archivos maestros.

El sistema hace uso de más archivos de bases de datos, sin embargo estos son temporales, por ejemplo, en la generación de todos los reportes, se seleccionan los datos de la base adecuada y se guarda en un archivo, al finalizar el reporte con éxito, se elimina dicho archivo.

*Fuente: Propia.

5. IMPLANTACION DEL SISTEMA

Capacitación del personal

Construcción y prueba de los programas

Pruebas al sistema

Conversión del sistema

Soporte de sistemas

V. IMPLANTACION DEL SISTEMA

La implantación es la construcción del nuevo sistema, asegurar el buen funcionamiento del sistema de información e involucrar a los usuarios, previamente capacitados, en su operación. Este proceso consta de las fases mostradas en la figura 5.1.

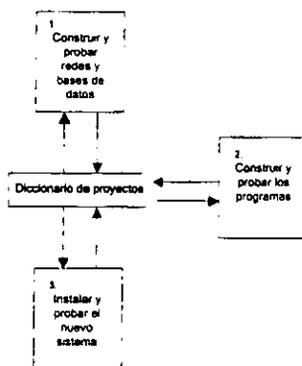


Figura 5.1.

También se observan los flujos de salida en las fases de implantación. Esto representa información almacenada en el diccionario de proyectos, la cual es requerida para desarrollar otra actividad. Se debe redactar los detalles y características técnicas de la nueva actividad y se guardan en el diccionario, este flujo es representado por las flechas que apuntan hacia el diccionario. Por ejemplo, se necesita la información del diseño técnico de las bases de datos y redes para construirlos (salida), la redacción de los detalles de esta tarea se almacena en el diccionario de proyectos (entrada).

Cuando se construyen aplicaciones nuevas o mejoradas en torno a redes y bases de datos ya existentes, la primera fase se omite, de lo contrario, deberán implantarse dichas redes y bases antes de escribir o instalar los programas informáticos.

El diccionario de proyecto, también conocido como diccionario de datos, es una herramienta para la documentación de los sistemas. En él se registra la información detallada sobre los modelos de datos, procesos, redes y los programas asociados a una aplicación o a un proyecto. El diccionario almacena estos modelos y se amplía la descripción de los objetos que aparecen en él.

Los objetos de datos (entidades de datos, flujo de datos y almacenes de datos) constan de atributos de datos que son las unidades mínimas de datos que tienen significado para el usuario. Los atributos se disponen según patrones específicos denominados estructuras de datos, que describen una única presencia del objeto. Estas estructuras de datos, junto con otros atributos lógicos, quedan registradas en el diccionario.

¹ Análisis y diseño de sistemas de información, Jeffrey L. Whitten.

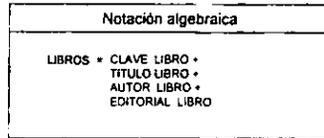


Figura 5.2

El contenido de una presencia cualquiera de un objeto de datos puede documentarse con la combinación de tres estructuras de datos, llamada notación algebraica booleana, sugerida por Tom DeMarco y además es la más utilizada. Simplemente, se utiliza un signo de igual, en la parte izquierda de este signo se coloca el nombre del flujo de datos, el almacén de datos o la entidad de datos, y del lado derecho del signo se colocan los atributos de datos.

1. La estructura de datos en secuencia. Contiene un grupo de atributos dispuestos en serie, por ejemplo en la figura 5.2 se lee como: una presencia de libros consta de los siguientes atributos: CLAVE DEL LIBRO y TITULO DEL LIBRO y, ETC. En la notación algebraica, el signo más se interpreta como la letra "y"
2. La estructura de datos selección. Permite mostrar situaciones en la que, dada una única presencia de flujo de datos, almacén de datos o una entidad de datos, ha de verificarse una de las siguientes afirmaciones:
 - Uno y sólo un atributo de la lista de atributos de datos adoptará un valor determinado, denominado selección de tipo o-exclusivo.
 - Existen uno o más atributos de datos de la lista que asumirán un valor determinado, denominado selección de tipo o-inclusivo.

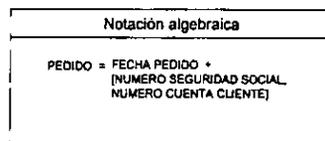


Figura 5.3

La figura 5.3 muestra un ejemplo de sentencia de tipo o-exclusivo. Un pedido dado puede solicitarlo una persona o una empresa, pero no por ambos a la vez. Por tanto, cualquier presencia dada en un pedido constará bien de FECHA DE PEDIDO y un NUMERO DE IDENTIFICACION o bien por una FECHA DE PEDIDO y un NUMERO DE CUENTA DE CLIENTE. Se observa que los atributos de selección se incluyen entre llaves y están separados por comas.

¹ Fuente: Análisis y diseño de sistemas de información, Jeffrey L. Whitten.

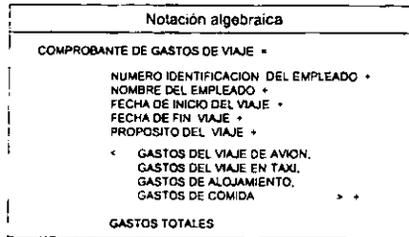


Figura 5.4

La figura 5.4 muestra un ejemplo de sentencia o-inclusivo, además se han combinado sentencias en secuencia y de selección para definir la estructura de datos. Quiere decir que una NOTA DE GASTOS DE VIAJE puede contener uno o más gastos de la lista. Siempre se tendrán los GASTOS TOTALES. Los atributos que se pueden seleccionar se sitúan entre signos de mayor y menor; y son separados por comas.

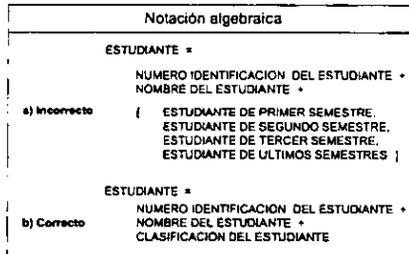


Figura 5.5

Es importante no cometer un error muy común al definir las estructuras, dicho error se observa en la figura 5.5 y el cual consiste en documentar valores de atributos de datos en vez de los atributos de datos.

La estructura de datos repetición. Se utiliza para establecer los atributos de datos o los grupos de atributos que se repetirán un número especificado de veces para dada presencia del flujo de datos, el almacén de datos o la entidad de datos. En forma algebraica se documenta del modo siguiente:

$$M\{\text{Lista de atributos o grupos de atributos}\}N$$

Donde M es el número mínimo de presencias y N es el número máximo de presencias de la estructura de datos. Si M=0, el grupo en su conjunto es opcional. En cualquier caso, el

¹ Fuente: Análisis y diseño de sistemas de información, Jeffrey L. Whitten.

grupo completo de datos se repite como tal grupo. Si el número de repeticiones fuera indefinido, no se le asigna valor numérico a N.

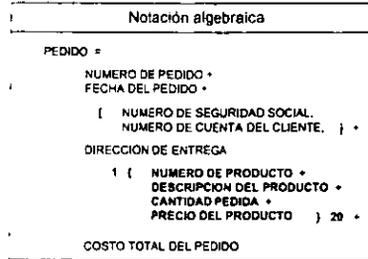


Figura 5.6

El contenido de un grupo repetitivo puede incluir una secuencia de atributos, una o varias sentencias de selección o incluso sentencias de repetición anidadas. La figura 5.6 muestra que la secuencia de atributos de datos relacionados se definió como grupo repetitivo. En la figura 5.7 se muestra otro ejemplo, se tiene que existen cuatro semanas en un periodo de pago, representando por el primer grupo repetitivo. En una cualquiera o en la totalidad de estas semanas, los empleados tendrán entre uno y siete valores de horas trabajadas, uno para cada día trabajado. Se trata, en este caso, de un grupo repetitivo anidado. La llave interior se produce de una a siete veces para cada presencia de periodo de pago, lo que también sucede de una a cuatro veces.

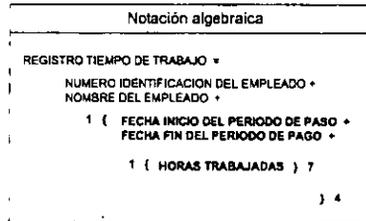


Figura 5.7

También en el diccionario de datos se especifican las normas, las cuales se documentan con tablas de decisión, descritas en el capítulo dos.

De acuerdo a las necesidades de diseño, se harán las modificaciones apropiadas en las redes existentes que puedan ser usadas por el nuevo sistema. Si es necesario, alternativamente, se desarrollan desde el principio las redes nuevas. Los detalles de éstas se guardan en el diccionario de proyectos para referencias futuras.

¹ Fuente: Análisis y diseño de sistemas de información. Jeffrey L. Whitten.

La construcción y prueba de archivos y bases de datos es una tarea que se realiza antes que las actividades de programación ya que este tipo de archivos son necesarios para probar los programas. El resultado de estas tareas son, los archivos vacíos y redes configuradas.

Una vez construidas las redes y las bases de datos, la siguiente fase refiere a la construcción y pruebas de los programas, lo cual se consigue con las siguientes tareas:

1. El plan de programación, se hace la revisión de las especificaciones de diseño, se organiza el equipo de programación y será necesario ubicar los módulos y programas de acuerdo a la prioridad, para su codificación.
2. Escribir y probar los nuevos programas, para esta actividad se hace uso del ciclo de vida del desarrollo de programas, es decir, se utiliza una metodología para la creación de programas.

Durante la instalación y prueba del sistema se realizan tres actividades: se instalan y se prueban los paquetes adquiridos de vendedores, se hace una prueba completa del sistema para asegurarse que los programas creados y adquiridos funcionan en conjunto de manera apropiada y se decide el plan de conversión del sistema. Las dos primeras actividades se realizan cuando se adquieren programas comerciales, si se construye todo el sistema con medios propios, es decir, no se compran programas, entonces, la dos primeras actividades no tienen razón de ser.

La tercer actividad implica realizar la instalación de los archivos y bases de datos que utilizará el nuevo sistema, proporcionar capacitación y documentación del sistema a los usuarios, evaluar el proyecto y por último establecer la estrategia para reemplazar el antiguo sistema por el nuevo.

5.1 Capacitación del personal

Algunos programas bien diseñados pueden fracasar si la forma en que se operan y usan no es la adecuada, por tal motivo es importante capacitar a todos los usuarios involucrados con el sistema, es decir, desde el capturista hasta aquellos que usan la salida para la toma de decisiones. Estos deben conocer en detalle cómo pueden hacer uso del sistema, qué hace o no hace el sistema. También se recomienda, para sistemas grandes, agrupar a los usuarios de acuerdo al uso que le darán al sistema, ejemplo, un grupo para aquellos que solamente lo manejarán, otro para los de soporte y se forma un grupo de usuarios que únicamente les importe interpretar los reportes del sistema.

Se utilizan diversos enfoques para llevar la capacitación a los usuarios, estos dependen de los objetivos que se desean lograr, los más empleados son:

- **Seminarios e instrucciones por grupo.** Este enfoque permite llegar a varias personas y es útil cuando se quiere presentar una visión general del sistema.
- **Capacitación de procedimientos.** El individuo tiene la oportunidad de hacer preguntas y plantear problemas referentes al procedimiento, ya sea en una sesión de grupo o individualmente. También se proporciona un documento escrito formal del sistema.

5.2 Construcción y pruebas de los programas

La creación de un programa se puede dividir en diferentes fases, figura 5.8, análisis, diseño, codificación, prueba y mantenimiento. Estas fases se conocen como ciclo de vida del software, y son los principios básicos en los que se apoya la ingeniería del software.



Figura 5.8 Ciclo de vida del software

El análisis se refiere a la definición de requerimientos y es el periodo durante el cual se especifican las características funcionales y operacionales. La entrada a esta fase son las necesidades declaradas o las tareas que deberá realizar el programa y la salida es un documento con los requerimientos, es decir, especifica lo que el sistema debe hacer pero no dice como hacerlo. El proceso de diseño inicia cuando están bien definidos los requerimientos. Una tarea importante en el diseño, es la descomposición del sistema en un conjunto de subsistemas. Cada subsistema se descompone posteriormente en un conjunto de programas y procedimientos más pequeños. Es recomendable verificar si cualquiera de los programas, funciones o procedimientos existentes se pueden reutilizar en el nuevo sistema. Otra parte importante del proceso del diseño es definir la representación interna de los datos.

La entrada a esta fase es un documento de requerimientos depurados y validados, la salida es un diseño expresado en alguna herramienta adecuada, por ejemplo, pseudocódigo.

² Fuente: Turbo Pascal, Luis Joyanes Aguilar.

Una vez que el diseño ha finalizado, éste se codifica en un lenguaje de programación específico. En esta fase intervienen diferentes programadores, por esta razón es necesario que el diseño del programa sea cuidadosamente documentado, basado en diagramas de estructura y pseudocódigo esencialmente.

Cuando no quedan errores se habrá finalizado la fase de codificación y depuración, y es momento de probar el programa. Esta prueba debe ser un proceso riguroso, que generalmente se ejecuta por un grupo distinto al de los programadores originales y por los usuarios del programa.

El mantenimiento es el proceso de actualizar o mantener un programa, consiste en corregir los nuevos errores que se produzcan o incorporar los cambios necesarios. El programa debe poder corregirse eficazmente durante el periodo de vida útil, incluso en entornos cambiantes.

La prueba de programas proporciona una base para asegurar que el sistema funcione como se requiere. La prueba de programas es la última oportunidad para asegurarse que el sistema cubra con los requerimientos de los usuarios. No obstante, una prueba de programas nunca puede demostrar que un programa sea correcto. La prueba sólo demuestra la presencia de errores; no puede probar la ausencia. Siempre es posible que existan errores no detectados aun después de realizar las pruebas más completas.

Como una prueba rigurosa no garantiza la ausencia de errores en los programas, el tiempo dedicado a las pruebas y los métodos empleados dependen de la aplicación particular del sistema.

En relación con las pruebas, Edsger Dijkstra ha escrito que las pruebas efectivamente muestran la presencia de errores, nunca pueden demostrar la ausencia. Una prueba con éxito significa que ningún error se descubrió en las circunstancias particulares probadas, pero no dice nada sobre las otras. En teoría, el único medio de comprobar que un programa es correcto es probar todos los casos posibles, situación técnicamente imposible, incluso para los programas más simples, no obstante, el análisis anterior no significa que la comprobación sea imposible, tomando en cuenta las siguientes consideraciones se puede llegar a eliminar los errores de los programas, no el cien por ciento, pero si en un alto porcentaje.

1. Suponer que el programa tiene errores hasta que las pruebas demuestren lo contrario.
2. Ninguna prueba simple de ejecución puede demostrar que un programa este libre de errores.
3. Someter al programa a pruebas duras. Un programa bien diseñado manipula entradas de manera concisa, es decir, no debería producir errores en tiempo de ejecución ni produce resultados incorrectos, por el contrario, el programa, en la mayoría de los casos visualiza un mensaje de error y solicita de nuevo los datos de entrada.

4. Comenzar con la comprobación antes de terminar con la codificación, esto quiere decir que importante comprobar cada módulo antes de incorporarlo al programa o al sistema.
5. Cambiar sólo una cosa cada vez.

Tipos de errores

Los errores de programación se pueden dividir en tres clases, de compilación o de sintaxis, errores en tiempo de ejecución y errores lógicos. La figura 5.9 muestra el proceso de los errores en la ejecución de un programa. Los errores de sintaxis se originan cuando no hay una analogía con las reglas de la gramática del lenguaje de programación y con la escritura del programa. El siguiente ejemplo presenta dos errores de sintaxis: la palabra reservada NEXT en la última línea hace falta para cerrar el FOR y el nombre de la función FIELNAME() debe ser FIELDNAME().

```

FUNCTION IGUA_VAR()
A=FCOUNT()
FOR I=1 TO A
  IF TYPE(FIELD(I)) <> "M"
    NOMCAM=FIELNAME(I)
    NOMVAR="M"+NOMCAM
    PUBLIC &NOMVAR
    &NOMVAR=&NOMCAM
  ENDIF

```

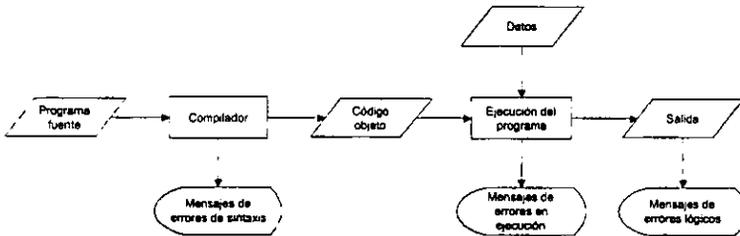


Figura 5.9 Errores que producen los programas.

Los errores en tiempo de ejecución suceden cuando el programa trata de hacer algo incongruente. Errores típicos son la división por cero, intentar utilizar un subíndice fuera de los límites definidos en el arreglo, etc.

$X=1/N$ Produce un error cuando $N=0$.

^E Fuente: Turbo pascal, Luis Joyanes Aguilar.

Los errores lógicos son difíciles de encontrar porque el compilador no produce ningún mensaje de error. Son errores de algoritmo o de lógica del programa y se producen cuando el programa es perfectamente válido y produce una respuesta. En el ejemplo siguiente la media está calculada equivocadamente ya que existen once números (0 al 10) y no diez como se ha escrito.

```
SUMA=0
@ 5.5 GET NUM
SUMA=SUUMA+NUM
MEDIA=SUMA/10
FOR J=0 TO 10
READ
NEXT
```

Las pruebas de los programas no son actividades que deban demorarse hasta que se haya escrito completamente el programa, es necesario realizarlas en los tres niveles de pruebas que se describen:

- Pruebas individuales, son realizadas sobre módulos particulares, que pueden ser programas, subrutinas, subprogramas o bloques.
- Prueba de unidades, donde se verifican todos los módulos codificados, ya probados individualmente, como una sola unidad integrada.
- Prueba del sistema, con esto se garantiza que los programas de aplicación escritos por separado funcionen de modo adecuado cuando se integran al sistema global.

Los módulos de nivel superior se pueden probar antes de haber codificado los de nivel inferior, mediante la simulación de los segundos, con un mensaje o con alguna señal que nos indique se han sido invocados correctamente y devuelven el control al módulo principal.

El siguiente fragmento del programa principal del sistema de biblioteca, es un ejemplo de un módulo superior, muestra la pantalla principal del sistema y aún cuando los programas a los que llama por medio del comando DO, no están codificados, este módulo se prueba simulando a estos programas utilizando mensajes. Es importante llevar a cabo las tres pruebas descritas anteriormente, ya que un programa puede funcionar bien por separado, pero cuando se integra al sistema general crea problemas, como son variables no reconocidas por el programa, la salida enviada de un programa a otro no es aceptada, etc.

```

DO CASE
  CASE SALIR=1
    RESTORE SCREEN FROM PA
    G=F.
  CASE SALIR=2
    SET COLOR TO
    CLEAR
    QUIT
  ENDCASE
ENDDO

CASE opc=1
  OPC_I:=MOVIMIENTO()
  DO CASE
    CASE OPC_I=1
      DO PRE_LIB
    CASE OPC_I=2
      DO DEV_LIB
  ENDCASE

CASE opc=2
  OPC_I2:=CONSULTA()
  DO CASE
    CASE OPC_I2=1
      DO COLI_TIT
    CASE OPC_I2=2
      DO COLI_TEM
    CASE OPC_I2=3
      DO COLI_AUT
    CASE OPC_I2=4
      DO COLI_CLA
    CASE OPC_I2=5
      DO CO_LIPRE
    CASE OPC_I2=6
      DO CO_LIVEN
    CASE OPC_I2=7
      DO CON_USU
  ENDCASE

```

Se utiliza el concepto de programación dirigida por los datos para el ahorro de tiempo en el desarrollo y mantenimiento de la aplicación. Para entender mejor este concepto se va a explicar el siguiente ejemplo:

```

use lib_lib
do while .t.
  Ncla_lib = cla_lib
  Nautor  = autor
  Ntitulo = titulo
  Neditoria = editorial
Enddo

```

En este ejemplo son inicializadas las variables de memoria con el contenido de los campos de un archivo para utilizarlas después. Cada nombre de campo y de variable han sido explícitamente especificados. Si se tuvieran cinco campos, tendrían cinco líneas de asignación. Teniendo treinta campos, serían necesarias treinta líneas de asignación.

Esta misma tarea se realiza por una rutina que utiliza el concepto de programación dirigida por los datos. El bucle FOR crea una variable para cada campo del archivo de bases de datos en uso. La función fcount() devuelve el número de campos de la base. Se asigna a la variable VAR del nombre el campo más una M al inicio, sentencia "M"+fieldname() y por último el contenido del campo se le asigna a la variable VAR.

```
use lib_bib
  for i=1 to fcount()
    var="M"+fieldname()
    &var=&(fieldname(i))
  next
```

Con sólo dos líneas, todas las variables de memoria son inicializadas. Tanto si existen 5 campos o 40, estas mismas dos líneas hacen esa función, de esta forma no es necesario preocuparse de teclear el nombre de los campos y de las variables, evitando errores cometidos al teclear los programas.

También se sustituye a las macros por bloques de código el cual es un tipo de dato que contiene una expresión compilada que puede ser evaluada. Esto quiere decir que cualquier error se identifica a nivel compilación. Mientras que las macros es un símbolo que informa al compilador para que asuma el contenido de una variable como si fuera una literal que solo puede ser conocido durante la ejecución. Por lo anterior, al hacer la sustitución de macros por códigos de bloque se reducen los errores de ejecución después de que el sistema ha sido implantado. Otra ventaja es la rapidez de ejecución ya que el bloque de código se compila solo una vez, mientras que la macro cada vez que se vaya a utilizar la expresión.

5.3 Pruebas al sistema

Las pruebas del sistema se hacen cada vez que se integra un módulo. Primero se cuida que la parte a integrar funcione individualmente de acuerdo a lo esperado, para después integrarlo y volverlo a probar con todas las partes del sistema. Esto probará que el módulo que se integra no ocasione alguna dificultad con el resto de los módulos.

5.3.1 Prueba de entrada y salida

La prueba de la entrada implica determinar si las formas de papel y las estructuras de codificación cumplen con las guías y las especificaciones del diseño. Además la salida por pantalla deben tener una distribución adecuada y no deben tener datos innecesarios. Se debe garantizar que los datos de entrada sean validados, para realizar esta tarea es importante tomar en cuenta las siguientes recomendaciones:

- Comprobar que se tienen todos los elementos necesarios, es decir, determinar si se han introducido realmente la información en todos los campos requeridos en la entrada.

- Comprobaciones de límites e intervalos, para determinar si los datos de entrada de cada campo están comprendidos en el conjunto o intervalo correcto de valores definidos para dicho campo.
- Dígitos de autocomprobación, permiten describir los errores de introducción de datos en las claves primarias. Un dígito de comprobación es un número o carácter que se añade a un campo de clave primaria, se calcula aplicando una fórmula o un algoritmo.

Una prueba importante de la salida consiste en la generación de reportes, mismos que se entregan al usuario para que él indique si dicho reporte satisface sus necesidades de información. Por lo general, una buena prueba para determinar si el formato de la salida es comprensible consiste en mostrar la salida a una persona que no esté involucrada con el sistema. Si la persona logra entender el reporte, entonces el formato probablemente pueda ser entendido por los usuarios apropiados.

5.3.2 Prueba de la base de datos

Para determinar si los contenidos de la base de datos satisfacen los requerimientos de los usuarios se pueden observar en la salida pero, adicionalmente se deben hacer pruebas de creación de un nuevo registro, intentar introducir registro incorrectos y verificar los totales predeterminados con los producidos por el sistema.

5.3.3 Prueba de los controles

El propósito de la prueba de los controles es asegurar que éstos se encuentren en el lugar adecuado y trabajen según se planeó. Por lo tanto se requiere observar y estudiar a los controles, realizar pruebas reales y evaluar la eficacia con que los controles satisfacen estas pruebas. Es importante hacer notar que al probar otros componentes estructurales, ya se habrán efectuado algunas pruebas de control. Por ejemplo, cuando se introducen datos se debe observar que algunos campos deben permitir datos en un rango específico, etc.

5.4 Conversión del sistema

Si las pruebas que se aplican al sistema son exitosas, el siguiente paso es el proceso de conversión. Cuatro métodos son los más utilizados para llevar a cabo la conversión del sistema. Cada método debe considerar las ventajas que ofrece y los problemas que puede ocasionar. En general, la conversión de sistemas debe realizarse lo más rápido posible.

Enfoque piloto

En este método se implanta una versión del sistema en una parte de la organización. Este enfoque tiene la ventaja de proporcionar una sólida base de prueba antes de la implantación total. Sin embargo, si no se conduce adecuadamente la implantación, los usuarios pueden llegar a pensar que el sistema sigue teniendo problemas y que no es posible confiar en él.

Método por etapas

Se implanta el sistema de manera gradual a todos los usuarios. Cuando los sistemas se desarrollan por etapas, deben de trabajar bien desde la primera conversión y las subsiguientes; permite la capacitación y la instalación con los recursos necesarios.

Conversión paralelo

Con el modo de conversión en paralelo, las salidas de cada sistema se comparan y las diferencias se corrigen. Las ventajas de este enfoque es la protección de la empresa ante el fracaso del nuevo sistema, en contrapartida se tiene un costo adicional al mantener duplicidad de equipo, instalaciones y personal, por tener en operación a los dos sistemas.

Conversión directa

El método de conversión directa es la implementación del nuevo sistema de manera inmediata. Este enfoque es el que se está considerando para implantar el sistema propuesto ya que es recomendable cuando el sistema no está reemplazando a ningún otro sistema o cuando el sistema nuevo no realiza tareas críticas.

En nuestro caso el sistema de biblioteca se instaló, utilizando el método de conversión directa ya que el sistema es nuevo, es decir, no se encontraba anteriormente uno automatizado. A pesar de que el alcance del sistema se limita al uso exclusivo de esta institución, se puede considerar al programa como piloto, al ser evaluado y considerarlo como viable para ser implementado en todos los CETIS de la DGETI.

5.5 Soporte del sistema

El soporte de sistemas es realizado por diseñadores y constructores de sistemas como apoyo a los usuarios y consta de cuatro actividades permanentes: corregir errores, recuperar el sistema, asistir a los usuarios, adaptar el sistema ante nuevas necesidades. En la figura 5.10, se observa que para cada actividad se tiene al menos una entrada, por ejemplo, en el proceso de corregir errores, la entrada la constituyen aquellos errores que el sistema pueda generar al ser utilizado por los usuarios.

Corregir errores

Independientemente del diseño, la construcción y pruebas realizadas, son inevitables los errores. Estos pueden aparecer por falta de comunicación de las necesidades, por defectos de diseño, por situaciones no previstas y en consecuencia no probadas; y por mal uso no considerado del programa. En cada una de estas situaciones se deben tomar medidas para la corrección, estas acciones son conocidas como mantenimiento del sistema, los puntos a considerar son:

- Realizar los cambios predecibles en los programas para la corrección de los errores que se cometieron en el diseño e implantación del sistema. Se deben excluir a las nuevas necesidades y a las actividades de mejoras.
- Evitar la posibilidad de que al arreglar un programa se originen nuevos problemas o funcionen de forma diferente.

Recuperación del sistema

Es inevitable que un sistema falle y en consecuencia, la posible pérdida de datos. Generalmente es el analista el encargado de arreglar el sistema o de actuar como intermediario entre los usuarios y quienes deben recuperarlo. Esta actividad se puede resumir de la siguiente manera:

- En ocasiones puede ser tan simple como pulsar una tecla o inicializar la computadora. También se pueden requerir instrucciones de corrección para evitar que pueda producirse un fallo generalizado.
- En algunos casos, se tendrá que recurrir a la administración de datos para recuperar archivos o bases de datos dañadas.
- Es posible que sea necesario pedir auxilio a los técnicos o representantes de los vendedores para arreglar un problema de hardware.
- Si se descubre el error que ha provocado el fallo, se deberá intentar aislar dicho error y bloquearlo automáticamente o indicar a los usuarios para que lo eviten manualmente y de esta forma asegurar que este error provoque otro fallo.

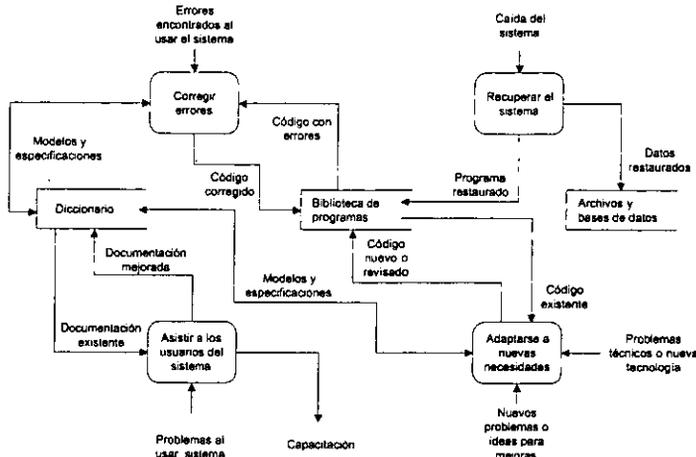


Figura 5.10

Asistencia al usuario

Independientemente de cómo haya sido la capacitación, los usuarios inevitablemente requerirán asistencia adicional. El analista debe estar a disposición de los usuarios para ofrecerles ayuda en el uso diario de aplicaciones específicas. Para poder dar dicha ayuda a los usuarios en forma eficiente, es importante llevar a cabo las siguientes tareas:

- Observación rutinaria del uso del sistema.
- Realizar estudios y reuniones para conocer el grado de satisfacción del usuario.
- Modificar los procedimientos para hacerlos más claros y guardarlos en el diccionario.
- Anotar en el diccionario las ideas y las solicitudes sobre posibles mejoras.

Adaptar el sistema

Normalmente la mayor parte de adaptaciones del sistema se hace como respuesta a la aparición de nuevos problemas, nuevas necesidades de información o ideas de mejoras y cambios en la tecnología. Los equipos que desarrollan sistemas de información no quieren esperar a que el sistema falle, actualmente, optan por analizar las bibliotecas de programas para determinar qué aplicaciones y programas son los más costosos de mantener o en cuáles

¹ Fuente: Análisis y diseño de sistemas de información, Jeffrey L. Whitten.

el mantenimiento resulta más difícil. Estos sistemas pueden ser adaptados para reducir costos y tiempo empleado en el mantenimiento. Estos ejemplos de adaptaciones, también se conocen como reingeniería y los objetivos son o bien adaptar el sistema ante un cambio tecnológico importante y arreglar el sistema antes de que falle o bien hacer un sistema más sencillo de manejar para cuando falle o tenga que ser adaptado.

La reingeniería de sistemas es un medio para ampliar la vida útil de las aplicaciones, sin embargo, todos los sistemas terminan por ser obsoletos, y se deberá volver a empezar la planificación o el análisis para construir completamente nuevas aplicaciones.

El ciclo de vida es presentado desde el punto de vista de la ingeniería directa. Primero, se analiza el problema y las necesidades, y después se diseña e implanta una solución. Pero, también es posible hacer las cosas al revés; es lo que se llama ingeniería inversa o ingeniería hacia atrás.

Dentro de cada programa existente, no importa lo nuevo o lo antiguo que pueda ser; hay almacenados conocimientos de diseño de todas las personas que han trabajado alguna vez en el programa, al extraer estos conocimientos, guardarlos en el diccionario y hacerlos disponibles, se reducen los esfuerzos de planificación, análisis y diseño del sistema. Sobre estas cuestiones tratará la recuperación de diseño y análisis. Al mismo tiempo, también es el objetivo de algunas herramientas CASE en auge. Actualmente el programa Excelerator para Recuperación de Diseño, de INTERSOLV puede leer un programa en COBOL y generar el modelo gráfico del sistema.

En otras palabras se está prestando cada vez más atención al desarrollo de métodos, herramientas y técnicas para dar mantenimiento a los sistemas existentes y la ingeniería hacia atrás es el proceso de crear especificaciones, en contraste con la ingeniería hacia adelante, en las que las especificaciones se crean de lo general a lo específico y que son seguidas por el desarrollo. La ingeniería hacia atrás es un concepto importante que se adecúa a la evolución de los sistemas de información. En la mayoría de las corporaciones, los sistemas no mueren; se les redirige, lo cual significa que los componentes se vuelven a desarrollar y volver a usar para crear nuevas aplicaciones.

La documentación que describe en forma completa a los sistemas de información, así como los formatos de archivos y bases de datos, y el código fuente de los programas, es donde se encuentran los detalles que explican cómo funciona un sistema, y en algunos casos por qué se les ha diseñado de esa manera. Los comentarios en el código del programa y la información que se incluye en reportes con cierto formato también proporcionan pistas importantes para hallar la lógica oculta en algunos programas.

ANEXOS

Tablas Cutter

Programas fuentes del sistema

Anexo 1

Lista de los programas fuente del sistema

```

* Programa para capturar nuevos libros
* Nombre del programa fuente: ALT_LIB PRG
* Modificado: 30-03-2001

CLEAR
SET ESCAPE OFF
SET CURSOR ON
SET SOFTSEEX ON
SAL='S'
FIL=3
COL=5
COLU=24

SELECT 1
USE TEMAS INDEX CLASIFI,TEM,TEM
SELECT 2
USE LIB_BIB INDEX CLASI,LI,AUT,PER,TIT,LIB
(IN) VARI()
SELECT 3
USE AREA INDEX NUM,AREA

@ 1.1 to 23.78 Double
@ 0.20 SAY ***** ALTAS DE LIBROS *****
@ 24.20 SAY 'PRESIONE LA TECLA ~ CHR(24)+CHR(25)+ ' PARA
MOVERSE'

        MNUM_AREA=0
        DO WHILE SAL='S'
                FIL=3
                MNUM_AREA=0
@ 2.5 SAY 'CLAVE DEL AREA ' GET MNUM_AREA PICT '999'
                READ

                CLEAR GETS
                IF MNUM_AREA < 0
                        SELECT 1
                        REINDEX
                        SELECT 2
                        REINDEX
                CLEAR
                CLOSE ALL
                RESTORE SCREEN FROM PA
                RETURN
                ENDIF

                SEEK MNUM_AREA
                IF NOT FOUND()
@ 22.7 SAY 'EL AREA NO EXISTE EN LA BASE..... OPRIMA CTRL
PARA CONTINUAR'
                        INKEY(0)
@ 22.7 CLEAR TO 22.77
                        LOOP
                        ENDIF

                MAREA=AREA
@ 3.5 SAY 'AREA ' + MAREA PICT '9'

                SELECT 2

@ 4.5 SAY 'INTRODUSCA LA CLASIFICACION ' GET MCLASI_LI PICT
'9' VALID REVISAR(MCLASI_LI)
                READ
                MCLASI_LI=ALLTRIM(MCLASI_LI)
                TNUM_AREA=TRANSFORM(MNUM_AREA,@X)
                TNUM_AREA=ALLTRIM(TNUM_AREA)
                MCLASI_LI=STUFF(MCLASI_LI,1,0,TNUM_AREA)

                SEEK MCLASI_LI
                IF FOUND()

@ 22.7 SAY 'LA CLASIFICACION YA EXISTE..... OPRIMA CTRL PARA
CONTINUAR'
                        INKEY(0)
                        RELEASE MCLASI_LI
                        MCLASI_LI=SPACE(7)
@ 2.5 CLEAR TO 22.77
                        SELECT 3
                        LOOP
                        ENDIF

                MTEMA1=SPACE(65)
                MTEMA2=SPACE(65)
                MTEMA3=SPACE(65)
                MTEMA4=SPACE(65)
                MTEMA5=SPACE(65)

@ FIL=2,COL SAY 'TITULO ' GET MTTITULO PICT '9'
@ FIL=3,COL SAY 'AUTOR PERSONAL ' GET MAUT_PER PICT '9'
@ FIL=4,COL SAY 'EDITORIAL ' GET MEDITORIAL PICT '9'
@ FIL=5,COL SAY 'AUTOR CORP. ' GET MAUT_CORP PICT '9'
@ FIL=6,COL SAY 'No PAGINAS ' GET MNQ_PAG PICT '9'
@ FIL=7,COL SAY 'CANTIDAD ' GET MCANTIDAD PICT '999'

```

```

@ FIL=8,COL SAY 'AÑO DE PUB ' GET MANO_PUB PICT '9'
@ FIL=9,COL SAY 'LUGAR DE PUB. ' GET MLUGAR PICT '9'
@ FIL=10,COL SAY 'CLAVE ISBN ' GET MCLA_ISBN PICT '9'
@ FIL=12,COL SAY 'TEMA 1 ' GET MTEMA1 PICT '9'
@ FIL=13,COL SAY 'TEMA 2 ' GET MTEMA2 PICT '9'
@ FIL=14,COL SAY 'TEMA 3 ' GET MTEMA3 PICT '9'
@ FIL=15,COL SAY 'TEMA 4 ' GET MTEMA4 PICT '9'
@ FIL=16,COL SAY 'TEMA 5 ' GET MTEMA5 PICT '9'

                READ
                SN=SPACE(1)
                SN='S'
@ 22.16 SAY ' GUARDAR LOS DATOS (S/N) ? ' GET SN PICT '9'
                VALID (SN='S')
                READ
                IF SN='S'
@ 22.7 CLEAR TO 22.77

                SELECT 2
                APPEND BLANK
                REPL,VAR()

                SELECT 1
                ADRUEBA=EMPTY(MTEMA1)
                BDRUEBA=EMPTY(MTEMA2)
                CPRUEBA=EMPTY(MTEMA3)
                DPRUEBA=EMPTY(MTEMA4)
                EPRUEBA=EMPTY(MTEMA5)
                IF ADRUEBA=.F.
                        APPEND BLANK
                        REPLACE CVE,AREA WITH MNUM_AREA
                        REPLACE TEMA WITH MTEMA1
                        REPLACE CLASI,LI WITH MCLASI_LI
                ENDIF
                IF BDRUEBA=.F.
                        APPEND BLANK
                        REPLACE CVE,AREA WITH MNUM_AREA
                        REPLACE TEMA WITH MTEMA2
                        REPLACE CLASI,LI WITH MCLASI_LI
                ENDIF
                IF CPRUEBA=.F.
                        APPEND BLANK
                        REPLACE CVE,AREA WITH MNUM_AREA
                        REPLACE TEMA WITH MTEMA3
                        REPLACE CLASI,LI WITH MCLASI_LI
                ENDIF
                IF DPRUEBA=.F.
                        APPEND BLANK
                        REPLACE CVE,AREA WITH MNUM_AREA
                        REPLACE TEMA WITH MTEMA4
                        REPLACE CLASI,LI WITH MCLASI_LI
                ENDIF
                IF EPRUEBA=.F.
                        APPEND BLANK
                        REPLACE CVE,AREA WITH MNUM_AREA
                        REPLACE TEMA WITH MTEMA5
                        REPLACE CLASI,LI WITH MCLASI_LI
                ENDIF
                ENDIF
                * LOS CINCO IF SON PARA NO GUARDAR LOS REGISTROS VACIOS O
                EN BLANCO

                SN='S'
@ 22.15 SAY ' CONTINUAR DANDO DE ALTAS (S/N) ' GET SAL PICT
'9' VALID(SAL='S')
                READ
@ 2.5 CLEAR TO 22.77

                SELECT 2
                LIB,VAR()
                INI,VAR()
                SELECT 3
                ENDDO

                SELECT 1
                REINDEX
                SELECT 2
                REINDEX
                CLOSE DATABASES
                RESTORE SCREEN FROM PA
                RETURN

* Programa para capturar usuarios.
* Nombre del programa fuente: ALT_USU PRG.
* Modificado: 30-03-2001.
* SET BELL ON
CLEAR
SAL=SPACE(1)
SAL='S'

USE USUARIO INDEX CVE_USI,NOM_USI

@ 1.1 to 23.78 Double
@ 0.20 SAY ***** ALTAS DE USUARIOS *****

```

Anexo I

```

@ 24.8 SAY 'PRESIONE LA TECLA '+' CHR(24)+CHR(25)+' PARA
MOVERSE+' O '<< ESC >>' PARA SALIR

      DO WHILE SAL="S"
@ 5.5 CLEAR TO 21.73
      INI VAR()
@ 5.5 SAY 'CLAVE DEL USUARIO:' GET MCLA USU PICTURE
'9999999999'
      READ
      IF EMPTY(MCLA, USU)
      REINDEX
      CLEAR
      CLOSE ALL
      RESTORE SCREEN FROM PA
      RETURN
      ENDIF

      SEEK MCLA_USU
      IF FOUND()
@ 21.7 SAY 'EL USUARIO YA EXISTE EN LA BASE... OPRIMA CTRL
PARA CONTINUAR'
      INKEY(0)
      LOOP
      ENDIF

@ 6.5 SAY 'NOMBRE' 'GET MNOM_USU PICTURE 'G'I'
@ 7.5 SAY 'SEMESTRE' 'GET MSEM PICTURE '9' RANGE 1:8
@ 8.5 SAY 'SEXO' 'GET MSEXO PICTURE '1' VALID (MSEXO
$ "M")
@ 9.5 SAY 'GRUPO' 'GET MGPO PICTURE '1' VALID (MGPO
$ "ABCDEFGHIJK")
@ 10.5 SAY 'TURNO' 'GET MTURNO PICTURE '1' VALID
(MTURNO $ "MAY")
      READ
      CLEAR GETS

SN=SPACE(1)
SN='S'
@ 21.15 SAY ' GUARDAR LOS DATOS (S/N) ? ' GET SN PICT '1'
VALID (SN $ "SN")
      READ
      IF SN='S'
      APPEND BLANK
      REPL VAR()
      ENDIF
@ 21.15 CLEAR TO 21.70

LIB VAR()
@ 21.15 SAY ' CONTINUAR DANDO DE ALTAS (S/N) ' GET SAL PICT
'1' VALID (SAL $ "SN")
      READ
      ENDDO
      REINDEX
      CLOSE ALL
      RESTORE SCREEN FROM PA
      RETURN

* Programa para dar de baja a libros.
* Nombre del programa fuente: BAJ_LIB.PRG.
* Modificado: 30-03-2001.

CLEAR
SET CURSOR ON
SET DELETED ON
SET ESCAPE OFF
SET WRAP ON

SAL=SPACE(1)
SAL='S'
FIL=2
COL=5
COL2=23
COL3=12

SELECT 1
USE TEMAS INDEX CLASIFI.TEM, TEM
SELECT 2
USE LIB_BIB INDEX CLASI_LI,AUT_PER,TIT_LIB

@ 1.1 to 23.78 Double
@ 0.20 SAY ***** BAJAS DE LIBROS *****
@ 24.20 SAY 'PRESIONE LA TECLA '+' CHR(24)+CHR(25)+' PARA
MOVERSE

      DO WHILE SAL="S"

      MCLASI_LI=SPACE(10)
@ 2.5 SAY 'INTRODUZCA LA CLASIFICACION:' GET MCLASI_LI PICT
'G'I'
      READ

```

Lista de los programas fuente del sistema

```

CLEAR GETS
IF EMPTY(MCLASI_LI)
CLEAR
SELECT 1
PACK
REINDEX
SELECT 2
PACK
REINDEX
CLOSE ALL
RESTORE SCREEN FROM PA
RETURN
ENDIF
      SEEK MCLASI_LI
      IF FOUND()
      IGUA VAR()
@ FIL+1,COL SAY 'AREA '
@ FIL+1,13 SAY AREA
@ FIL+2,COL SAY 'TITULO DEL LIBRO.'
@ FIL+2,COL2 SAY TITULO
@ FIL+3,COL SAY 'AUTOR'
@ FIL+3,COL2 SAY AUT PER
@ FIL+4,COL SAY 'AUTOR CORP'
@ FIL+4,COL2 SAY AUT COR
@ FIL+5,COL SAY 'EDITORIAL'
@ FIL+5,COL2 SAY EDITORIAL
@ FIL+6,COL SAY 'EJEMPLARES'
@ FIL+6,COL2 SAY CANTIDAD
@ FIL+7,COL SAY 'AÑO DE PUB.'
@ FIL+7,COL2 SAY AÑO DE PUB
@ FIL+8,COL SAY 'LUGAR DE PUB'
@ FIL+8,COL2 SAY LUGAR

SELECT 1

      SEEK MCLASI_LI
      IF FOUND()
      COMPARI=MCLASI_LI
      J=1
      I=11
      WHILE COMPARI=MCLASI_LI ' EN CASO DE QUE ALGUN LIBRO
SE HAYA CAPTURADO SIN '
@ FIL+1,COL SAY 'TEMA' ' SIN TEMAS, ENTONCES LOS
CAMPOS DE TEMAS NO APARECEN.'
@ FIL+1,10 SAY 'J PICTURE '9' ES DECIR, UNICAMENTE LOS
TEMAS QUE SE CAPTURARON SON '
@ FIL+1,COL3 SAY 'TEMA' ' SON LOS QUE APARECEN.

      SKIP
      J++
      I++
      COMPARI=CLASI_LI
      ENDDO
      ENDDO
      ENDDO
      SN=SPACE(1)
      SN='S'
@ 21.15 SAY ' DAR DE BAJA (S/N) ? ' GET SN PICTURE '1'
VALID (SN $ "SN")
      READ
      IF SN='S'
      SEEK MCLASI_LI
      IF FOUND()
      COMPARA=MCLASI_LI

      WHILE COMPARA=MCLASI_LI
      DELETE
      SKIP
      COMPARA=CLASI_LI
      ENDDO
      ENDDO
      ENDDO
      SELECT 2
      DELETE
      ENDDO
      ELSE
@ 21.5 SAY 'LA CLASIFICACION NO EXISTE EN LA BASE... OPRIMA
CTRL PARA CONTINUAR'
      INKEY(0)
@ 21.5 CLEAR TO 21.77
      ENDDO
      SELECT 2
      RELEASE MCLASI_LI
@ 21.5 CLEAR TO 21.77
@ 21.15 SAY ' CONTINUAR DANDO DE BAJA (S/N) ' GET SAL
PICTURE '1' VALID (SAL $ "SN")
      READ
@ 2.5 CLEAR TO 21.77
      ENDDO
      SELECT 1
      PACK
      REINDEX
      SELECT 2
      PACK

```

Anexo I

```

REINDEX
CLOSE ALL
RESTORE SCREEN FROM PA
RETURN

* Programa para dar de baja e usuarios libros
* Nombre del programa fuente. BAJ_USU.PRG
* Modificado. 30-03-2001

CLEAR

SET ESCAPE ON
SET WRAP ON

SAL=SPACE(1)
SAL='S'

USE USUARIO INDEK CVE, USI,NOM,USI

DO WHILE SAL='S'

  1 5.5 CLEAR TO 10,70
  2 1.1 to 23,78 Double
  3 0.20 SAY ***** BAJAS DE USUARIOS *****
  4 24.10 SAY ***** PRESIONE LA TECLA * CHR(24)*CHR(25)* * PARA
  MOVEVERSE'
  5 21.15 CLEAR TO 21,72

MCLA=0

  6 5.5 SAY 'CLAVE DEL USUARIO ' GET MCLA PICTURE '999999999'
  READ
  CLEAR GETS
  IF EMPTY(MCLA)
    REINDEX
    CLEAR
  CLOSE ALL
  RESTORE SCREEN FROM PA
  RETURN
  ENDIF
  SEEK MCLA
  IF FOUND()
    7 8.5 SAY 'NOMBRE DE ALUMNO '
    8 8.28 SAY 'NOM USU '
    9 7.5 SAY 'SEMESTRE '
    10 7.26 SAY 'SEM '
    11 8.5 SAY 'GRUPO '
    12 8.26 SAY 'GPO '
    13 9.5 SAY 'TURNO '
    14 9.26 SAY 'TURNO '
    READ

  SN=SPACE(1)
  SN='S'
  15 21.15 SAY ' DAR DE BAJA (S/N)? ' GET SN PICTURE 'T'
  VALID(SN $SN)
  READ
  IF SN='S'
    16 21.15 CLEAR TO 21,70

    DELETE
    PACK
    ENDIF
  ELSE
    17 21.15 SAY ' EL ASUARIO NO SE ENCONTRO'
    INKEY(2)
    ENDIF
    RELEASE MCLA
    18 21.15 CLEAR TO 21,65
    19 21.15 SAY ' CONTINUAR DANDO DE BAJA (S/N) ' GET SAL
    PICTURE '99'
    READ
    ENDDO
    REINDEX
    CLOSE ALL
    RESTORE SCREEN FROM PA
    RETURN

* Programa para modificar libros
* Nombre del programa fuente. CAM_LIB.PRG
* Modificado. 30-03-2001

CLEAR
SET CURSOR ON
SET DELETED ON
SAL=SPACE(1)
SAL='S'
FIL=2

```

Lista de los programas fuente del sistema

```

COL=5

SELECT 1
USE TEMAS INDEX CLASIFI,TEM,TEM
SELECT 2
USE LIB, BIB INDEX CLASI, LI,AUT, PER, TIT, LIB

  1 1.1 to 23,78 Double
  2 0.20 SAY ***** CAMBIOS DE LIBROS *****
  3 24.20 SAY 'PRESIONE LAS TECLAS * CHR(24)* * * * CHR(25)* *
  PARA MOVEVERSE'
  DO WHILE SAL='S'
  4 2.2 CLEAR TO 22,77

MCLASI,LI=SPACE(10)
  5 2.5 SAY 'INTRODUZCA LA CLASIFICACION: ' GET MCLASI, LI PICT
  6

  READ
  IF EMPTY(MCLASI,LI)
    CLEAR
    SELECT 1
    REINDEX
    SELECT 2
    REINDEX
    CLOSE DATABASES
    RESTORE SCREEN FROM PA
    RETURN
    ENDF
    SEEK MCLASI,LI
    IF NOT FOUND()
  7 22.7 SAY 'EL LIBRO NO EXISTE EN LA BASE... OPRIMA CTRL
  PARA CONTINUAR'
  INKEY(0)
  LOOP
  ENDF

  IGUA,VAR()
  8 3.5 SAY 'CLAVE DEL LIBRO ' GET MCLASI,LI PICTURE '999999'
  'para cambiar la clave'

  9 FIL+1,COL SAY 'AREA: ' GET MAREA PICTURE '9!'
  10 FIL+2,COL SAY 'TITULO ' GET MITITULO PICTURE '9!'
  11 FIL+3,COL SAY 'AUTOR PERSONAL ' GET MAUT,PER PICTURE
  '9!'
  12 FIL+4,COL SAY 'EDITORIAL ' GET MEDITORIAL PICTURE '9!'
  13 FIL+5,COL SAY 'AUTOR CORP. ' GET MAUT,COR PICTURE
  '9!'
  14 FIL+6,COL SAY 'No PAGINAS ' GET MNO,PAG PICTURE '9!'
  15 FIL+7,COL SAY 'CANTIDAD ' GET MCANTIDAD PICTURE '999'
  16 FIL+8,COL SAY 'AÑO DE PUB. ' GET MANO,PUB PICTURE '9!'
  17 FIL+9,COL SAY 'LUGAR DE PUB. ' GET MLUGAR PICTURE '9!'
  18 FIL+10,COL SAY 'CLAVE ISBN ' GET MCLA,ISBN PICTURE '9!'
  READ
  CLEAR GETS
  SELECT 1

  SEEK MCLASI,LI
  IF FOUND()
    COMPARE=MCLASI,LI
    K=CUENTA()
    DO CASE
      CASE K=1
        MTEMA2=SPACE(65)
        MTEMA3=SPACE(65)
        MTEMA4=SPACE(65)
        MTEMA5=SPACE(65)

        MTEMA1=TEMA
        19 FIL+13,COL SAY 'TEMA 1: ' GET MTEMA1 PICTURE '9!'
        20 FIL+14,COL SAY 'TEMA 2: ' GET MTEMA2 PICTURE '9!'
        21 FIL+15,COL SAY 'TEMA 3: ' GET MTEMA3 PICTURE '9!'
        22 FIL+16,COL SAY 'TEMA 4: ' GET MTEMA4 PICTURE '9!'
        23 FIL+17,COL SAY 'TEMA 5: ' GET MTEMA5 PICTURE '9!'
        READ
        CASE K=2
          MTEMA3=SPACE(65)
          MTEMA4=SPACE(65)
          MTEMA5=SPACE(65)

          MTEMA1=TEMA
          24 FIL+13,COL SAY 'TEMA 1: ' GET MTEMA1 PICTURE '9!'
          SKIP
          MTEMA2=TEMA
          25 FIL+14,COL SAY 'TEMA 2: ' GET MTEMA2 PICTURE '9!'
          26 FIL+15,COL SAY 'TEMA 3: ' GET MTEMA3 PICTURE '9!'
          27 FIL+16,COL SAY 'TEMA 4: ' GET MTEMA4 PICTURE '9!'
          28 FIL+17,COL SAY 'TEMA 5: ' GET MTEMA5 PICTURE '9!'
          READ
          CASE K=3
            MTEMA4=SPACE(65)
            MTEMA5=SPACE(65)

```

Anexo I

```

MTEMA1=TEMA
① FIL*13.COL SAY 'TEMA 1' GET MTEMA1 PICTURE '@!'
  SKIP
MTEMA2=TEMA
② FIL*14.COL SAY 'TEMA 2' GET MTEMA2 PICTURE '@!'
  SKIP
MTEMA3=TEMA
③ FIL*15.COL SAY 'TEMA 3' GET MTEMA3 PICTURE '@!'
④ FIL*16.COL SAY 'TEMA 4' GET MTEMA4 PICTURE '@!'
⑤ FIL*17.COL SAY 'TEMA 5' GET MTEMA5 PICTURE '@!'
  READ

CASE K=4
MTEMA5=SPACE(85)
MTEMA1=TEMA
① FIL*13.COL SAY 'TEMA 1' GET MTEMA1 PICTURE '@!'
  SKIP
MTEMA2=TEMA
② FIL*14.COL SAY 'TEMA 2' GET MTEMA2 PICTURE '@!'
  SKIP
MTEMA3=TEMA
③ FIL*15.COL SAY 'TEMA 3' GET MTEMA3 PICTURE '@!'
  SKIP
MTEMA4=TEMA
④ FIL*16.COL SAY 'TEMA 4' GET MTEMA4 PICTURE '@!'
⑤ FIL*17.COL SAY 'TEMA 5' GET MTEMA5 PICTURE '@!'
  READ

CASE K=5
MTEMA1=TEMA
① FIL*13.COL SAY 'TEMA 1' GET MTEMA1 PICTURE '@!'
  SKIP
MTEMA2=TEMA
② FIL*14.COL SAY 'TEMA 2' GET MTEMA2 PICTURE '@!'
  SKIP
MTEMA3=TEMA
③ FIL*15.COL SAY 'TEMA 3' GET MTEMA3 PICTURE '@!'
  SKIP
MTEMA4=TEMA
④ FIL*16.COL SAY 'TEMA 4' GET MTEMA4 PICTURE '@!'
  SKIP
MTEMA5=TEMA
⑤ FIL*17.COL SAY 'TEMA 5' GET MTEMA5 PICTURE '@!'
  READ
ENDCASE

ELSE
K=0
MTEMA1=SPACE(85)
MTEMA2=SPACE(85)
MTEMA3=SPACE(85)
MTEMA4=SPACE(85)
MTEMA5=SPACE(85)

① FIL*13.COL SAY 'TEMA 1' GET MTEMA1 PICTURE '@!'
② FIL*14.COL SAY 'TEMA 2' GET MTEMA2 PICTURE '@!'
③ FIL*15.COL SAY 'TEMA 3' GET MTEMA3 PICTURE '@!'
④ FIL*16.COL SAY 'TEMA 4' GET MTEMA4 PICTURE '@!'
⑤ FIL*17.COL SAY 'TEMA 5' GET MTEMA5 PICTURE '@!'
  READ

ENDIF
* K=0 no indica que no se encuentran temas guardados en la base, por es
necesario adicionar registros en la base temas.dbf. además el valor de * K=0
se utiliza en el siguiente caso.

SN=SPACE(1)
SN='S'
② 22.15 SAY ' GUARDAR LOS CAMBIOS (S / N) ? ' GET SN PICT 'T'
VALID SN $ 'SN'
  READ
  IF SN='S'
    APRUEBA=EMPTY(MTEMA1)
    BPRUEBA=EMPTY(MTEMA2)
    CPRUEBA=EMPTY(MTEMA3)
    DPRUEBA=EMPTY(MTEMA4)
    EPRUEBA=EMPTY(MTEMA5)

    SEEK MCLASI_LI
    IF FOUND()
      COMPARA=MCLASI_LI

      WHILE COMPARA=MCLASI_LI
        DELETE
        SKIP
        COMPARA=CLASI_LI
      ENDDO
      PACK
    ENDIF
    IF APRUEBA=F
      APPEND BLANK
      REPLACE CVE AREA WITH MNUM_AREA
      REPLACE TEMA WITH MTEMA1
  
```

Lista de los programas fuente del sistema

```

REPLACE CLASI_LI WITH MCLASI_LI
ENDIF
IF BPRUEBA=F
  APPEND BLANK
  REPLACE CVE AREA WITH MNUM_AREA
  REPLACE TEMA WITH MTEMA2
  REPLACE CLASI_LI WITH MCLASI_LI
ENDIF
IF CPRUEBA=F
  APPEND BLANK
  REPLACE CVE AREA WITH MNUM_AREA
  REPLACE TEMA WITH MTEMA3
  REPLACE CLASI_LI WITH MCLASI_LI
ENDIF
IF DPRUEBA=F
  APPEND BLANK
  REPLACE CVE AREA WITH MNUM_AREA
  REPLACE TEMA WITH MTEMA4
  REPLACE CLASI_LI WITH MCLASI_LI
ENDIF
IF EPRUEBA=F
  APPEND BLANK
  REPLACE CVE AREA WITH MNUM_AREA
  REPLACE TEMA WITH MTEMA5
  REPLACE CLASI_LI WITH MCLASI_LI
ENDIF

SELECT 2
  REPL_VAR()
② 22.2 CLEAR TO ZZ.77

LIB VAR()
RELEASE ALL LIKE MTEMA?
ENDIF

SELECT 2
② 22.15 SAY ' CONTINUAR MODIFICANDO (S / N) ' GET SAL PICT
'T' VALID (SAL $ 'SN')
  READ
  ENDDO

SELECT 1
  REINDEX
SELECT 2
  REINDEX
CLOSE DATABASES
RESTORE SCREEN FROM PA
RETURN

* En caso de que se modifique el título del libro se tiene que reindexar el
índice TIT_LIB, de no hacerlo al momento de hacer consultas por título del
libro se estaría utilizando el índice anterior.

* Programa para modificar a los usuarios
* Nombre del programa fuente: CAM_USU.PRG
* Modificado: 30-03-2001

CLEAR
SAL=SPACE(1)
SAL='S'

USE USUARIO INDEX CVE_USU;NDM_USU
① 1.1 TO 23.78 Double
② 0.20 SAY '***** CAMBIOS DE USUARIOS *****'
③ 24.20 SAY 'PRESIONE LAS TECLAS * CHR(24)* ** * CHR(25)* '
PARA MOVERSE'

DO WHILE SAL='S'
④ 5.5 CLEAR TO 21.73
MCLA_USU()
⑤ 5.5 SAY 'CLAVE USUARIO . ' GET MCLA_USU PICT '9999999999'
  READ
  IF EMPTY(MCLA_USU)
    REINDEX
    CLEAR
    CLOSE DATABASES
  RESTORE SCREEN FROM PA
  RETURN
  ENDF

  SEEK MCLA_USU
  IF .NOT FOUND()
⑥ 21.7 SAY 'EL USUARIO NO EXISTE EN LA BASE... OPRIMA CTRL
PARA CONTINUAR'
  INKEY(0)
  LOOP
  ENDF

  IGUA VAR()
⑦ 5.5 SAY 'CLAVE DEL USUARIO: ' GET MCLA_USU PICTURE
'9999999999 / (para cambiar la clave)'
⑧ 6.5 SAY 'NOMBRE ' GET MNUM_USU PICTURE '@!'

```

Anexo I

```

@ 7.5 SAY 'SEMESTRE' 'GET MSEM PICTURE 'Y'
@ 8.5 SAY 'SEDO' 'GET MSEXO PICTURE 'Q'
@ 9.5 SAY 'GRUPO' 'GET MGRO PICTURE 'Q'
@ 10.5 SAY 'TURNO' 'GET MTRNO PICTURE 'Q'
  READ
  CLEAR GETS
  SN=SPACE(1)
  SN='S'
@ 21.15 SAY ' GUARDAR LOS CAMBIOS (S/N) ? 'GET SN PICT 'T'
  VALID(SN $ 'SN')
  READ
  IF SN='S'
    REPL_VAR(1)
  ENDIF

@ 21.15 CLEAR TO 21.70

  LIB_VAR()
@ 21.15 SAY ' CONTINUAR MODIFICANDO (S/N) 'GET SAL PICT
'T' VALID(SN $ 'SN')
  READ
  ENDDO
  REINDEX
  CLOSE ALL
  RESTORE SCREEN FROM PA
  RETURN

* Programa consultar los libros prestados por usuario.
* Nombre del programa fuente: CO_LIPRE.PRG.
* Modificación: 30-03-2001.

SET SOFTSEEK ON
SET COLOR TO
CLEAR
SET WRAP ON
SAL=SPACE(1)
SAL='S'

USE PRE, LIB INDEX NOM, USU

@ 1.1 to 23.78 Double
@ 0.18 SAY '*** CONSULTAS DE LIBROS PRESTADOS ***'
@ 24.15 SAY 'PRESIONE LA TECLA <<< ESC >>> PARA SALIR'

  DO WHILE SAL='S'

RELEASE MNOM_USU
MNUM_USU=SPACE(40)

@ 4.5 CLEAR TO 21.77
@ 4.5 SAY 'NOMBRE DE USUARIO 'GET MNOM_USU PICTURE 'Q'
  READ

  IF EMPTY(MNOM_USU)
    CLEAR
    CLOSE ALL
  RESTORE SCREEN FROM PA
  RETURN

  ENDF

  IF LASTREC() <> 0

    CONSULTAS(MNOM_USU,1,1,4)

@ 4.5 CLEAR TO 4.77

@ 5.5 SAY 'TITULO'
@ 5.25 SAY 'TIT_LIB'
@ 6.5 SAY 'CLASIFICACION'
@ 6.25 SAY 'CLASI_LI'
@ 7.5 SAY 'NOMBRE'
@ 7.25 SAY 'NOM_USU'
@ 8.5 SAY 'CLAVE DE USUARIO'
@ 8.25 SAY 'CLA_USU'
@ 9.5 SAY 'FECHA DE PRESTAMO'
@ 9.25 SAY 'FEC_PRE'
@ 10.5 SAY 'DEVOLUCION'
@ 10.25 SAY 'FEC_DEV'
@ 11.5 SAY 'NUMERO CONSECUTIVO'
@ 11.25 SAY 'NUM_CONS'
@ 21.20 SAY 'CONTINUAR EN CONSULTAS (S/N) 'GET SAL PICTURE
'T' VALID(SAL $ 'SN')
  READ

  ELSE

SET CURSOR OFF

```

Lista de los programas fuente del sistema

```

@ 21.8 SAY 'NO HAY LIBROS PRESTADOS <<< CTRL >>> PARA
CONTINUAR'
  INKEY(0)
@ 21.8 CLEAR TO 21.77
  ENDF
  SET CURSOR ON
  ENDDO
  CLOSE ALL
  RESTORE SCREEN FROM PA
  RETURN

* Programa consultar los libros prestados vencidos.
* Nombre del programa fuente: CO_LIVEN.PRG.
* Modificación: 30-03-2001

SET COLOR TO
CLEAR
SET WRAP ON
SET SOFTSEEK ON

USE PRE, LIB INDEX NOM, USU
FEC_COM=DATE()-2
SET FILTER TO FEC_COM > FEC_PRE

SAL=SPACE(1)
SAL='S'

@ 1.1 to 23.78 Double
@ 0.15 SAY '*** CONSULTA DE LIBROS PRESTADOS VENCIDOS ***'
@ 24.15 SAY 'PRESIONE LA TECLA <<< ESC >>> PARA SALIR'

  DO WHILE SAL='S'

RELEASE MNOM_USU
@ 4.5 CLEAR TO 21.77
MNUM_USU=SPACE(40)

@ 5.5 SAY 'NOMBRE DEL USUARIO 'GET MNOM_USU PICTURE 'Q'
  READ

  IF EMPTY(MNOM_USU)
    CLEAR
    CLOSE ALL
  RESTORE SCREEN FROM PA
  RETURN
  ENDF

  COUNT TO CUENTA FOR FEC_COM > FEC_PRE
  IF CUENTA > 0
    CONSULTAS(MNOM_USU,1,1,6)

@ 5.5 CLEAR TO 5.77
@ 5.5 SAY 'NOMBRE'
@ 5.25 SAY 'NOM_USU'
@ 6.5 SAY 'TITULO'
@ 6.25 SAY 'TIT_LIB'
@ 7.5 SAY 'CLASIFICACION'
@ 7.25 SAY 'CLASI_LI'
@ 8.5 SAY 'NOMBRE'
@ 8.25 SAY 'NOM_USU'
@ 9.5 SAY 'CLAVE DE USUARIO'
@ 9.25 SAY 'CLA_USU'
@ 10.5 SAY 'FECHA DE PRESTAMO'
@ 10.25 SAY 'FEC_PRE'
@ 11.5 SAY 'DEVOLUCION'
@ 11.25 SAY 'FEC_DEV'
@ 12.5 SAY 'NUMERO CONSECUTIVO'
@ 12.25 SAY 'NUM_CONS'

  ELSE

@ 21.8 SAY 'NO HAY LIBROS VENCIDOS <<< CTRL >>> PARA
CONTINUAR'
  INKEY(0)
@ 21.8 CLEAR TO 21.77
  ENDF

@ 21.20 SAY 'CONTINUAR EN CONSULTAS (S/N) 'GET SAL PICTURE
'T' VALID(SAL $ 'SN')
  READ
  ENDDO

  CLOSE ALL
  RESTORE SCREEN FROM PA
  RETURN

* Programa consultar libros por autor.
* Nombre del programa fuente: COLI_AUT.PRG.
* Modificado: 30-03-2001.

SET COLOR TO
CLEAR

```

Anexo I

```

SET SOFTSEEK ON
SET WRAP ON
CLOSE ALL

SAL=SPACE(1)
SAL="S"

USE LIB_BIB INDEX AUT_PER

@ 1,1 to 23,78 Double
@ 0,18 SAY "" CONSULTAS DE LIBROS POR AUTOR ""
@ 24,15 SAY 'PRESIONE LA TECLA <<< ESC >>> PARA SALIR'

DO WHILE SAL="S"

@ 5,5 CLEAR TO 21,77

RELEASE MAUTOR
MAUTOR=SPACE(40)
@ 5,5 SAY 'NOMBRE DEL AUTOR 'GET MAUTOR PICTURE @I'
READ

IF EMPTY(MAUTOR)
CLEAR
CLOSE ALL
RESTORE SCREEN FROM PA
RETURN
ENDIF

* El primer parámetro es la clave por la que se busca, el segundo es el campo que se fija, el tercero y el cuarto son el rango de campos que se muestran.

CONSULTAS(MAUTOR 4,4,4)

@ 5,5 SAY 'NOMBRE DEL AUTOR '
@ 6,5 SAY 'AREA '
@ 7,5 SAY 'TITULO DEL LIBRO '
@ 8,5 SAY 'AUTOR PERSONAL '
@ 9,5 SAY 'EDITORIAL '
@ 10,5 SAY 'AUTOR CORP '
@ 11,5 SAY 'No PAGINAS '
@ 12,5 SAY 'EJEMPLARES '
@ 13,5 SAY 'AÑO DE PUB '
@ 14,5 SAY 'LUGAR DE PUB '
@ 5,28 SAY AUT_PER
@ 6,11 SAY AREA
@ 7,28 SAY CLASI_LI
@ 8,28 SAY TITULO
@ 9,28 SAY EDITORIAL
@ 10,28 SAY AUT_CDR
@ 11,28 SAY NO_PAG
@ 12,28 SAY CANTIDAD
@ 13,28 SAY AÑO_PUB
@ 14,28 SAY LUGAR

@ 21,20 SAY 'CONTINUAR EN CONSULTAS (S / N) 'GET SAL PICTURE
T VALID(SAL $ 'SN')
READ
ENDDO
CLOSE ALL
RESTORE SCREEN FROM PA
RETURN

* Programa consultar libros por clave
* Nombre del programa fuente: COJ_CLA.PRG
* Modificado: 30-03-2001.

SET COLOR TO
CLEAR
SET SOFTSEEK ON
SET WRAP ON
*SETCOLOR("B*B")

SAL=SPACE(1)

SAL="S"

FIL=2

COL=5

LIN=16

NUM_TEMA=1

SELECT 1

USE TEMAS INDEX CLASIFI
    
```

Lista de los programas fuente del sistema

```

SELECT 2

USE LIB_BIB INDEX CLASI_LI

@ 1,1 to 23,78 Double
@ 0,15 SAY "" CONSULTAS DE LIBROS POR CLASIFICACION ""
@ 24,15 SAY 'PRESIONE LA TECLA <<< ESC >>> PARA SALIR'

DO WHILE SAL="S"

RELEASE MCLASI_LI

@ 3,5 CLEAR TO 21,77
MCLASI_LI=SPACE(10)
@ 3,8 SAY 'CLASIFICACION 'GET MCLASI_LI PICTURE @I'
READ
IF MCLASI_LI=""
CLEAR
CLOSE ALL
RESTORE SCREEN FROM PA
RETURN
ENDIF

* El primer parámetro es la clave por la que se busca, el segundo es el campo que se fija, el tercero y el cuarto son el rango de campos que se muestran.

CONSULTAS(MCLASI_LI,1,1,8)

@ FIL+1,5 SAY 'CLAVE DE LIBRO '
@ FIL+2,5 SAY 'AREA '

@ FIL+3,5 SAY 'TITULO '
@ FIL+4,5 SAY 'AUTOR PERSONAL '
@ FIL+5,5 SAY 'EDITORIAL '
@ FIL+6,5 SAY 'AUTOR CORP '
@ FIL+7,5 SAY 'No PAGINAS '
@ FIL+8,5 SAY 'EJEMPLARES '
@ FIL+9,5 SAY 'AÑO DE PUB '
@ FIL+10,5 SAY 'LUGAR DE PUB '
@ FIL+1,28 SAY CLASI_LI
@ FIL+2,12 SAY AREA
@ FIL+3,28 SAY TITULO
@ FIL+4,28 SAY AUT_PER
@ FIL+5,28 SAY EDITORIAL
@ FIL+6,28 SAY AUT_CDR
@ FIL+7,28 SAY NO_PAG
@ FIL+8,28 SAY CANTIDAD
@ FIL+9,28 SAY AÑO_PUB
@ FIL+10,28 SAY LUGAR

MCLASI_LI=MCLASI_LI / ' EL USUARIO SE POSICIONA EL
DIFERENTE '

/ ' CLASIFICACION, POR LO QUE ES NECESARIO '

/ ' RESCATAR DICHA CLASIFICACION '

SELECT 1

* Se cambia a la base de datos donde se encuentran los temas de cada libro. Se tienen hasta 5 temas por libros.

SEEK MCLASI_LI

IF FOUND().

LIN=14

COMPARA=MCLASI_LI
WHILE COMPARA=MCLASI_LI
*SETCOLOR("B*B")

@ LIN,COL SAY 'TEMA '
@ LIN,COL+5 SAY NUM_TEMA PICTURE "9"
@ LIN,COL+7 SAY TEMA PICTURE @I
    
```

Anexo 1

```

NUM_TEMA=NUM_TEMA+1
LIN=LIN+1
SKIP
COMPARA=CLASI_LI
ENDDO
NUM_TEMA+1
LIN=14

Se cambia a la base de datos lib_bib.dbf ya que la base se encuentra temas.
es necesario cambiarse en ambos casos, es decir, si encuentra o no
encuentra temas, de lo contrario quedaria activada la base de datos
temas.dbf

SELECT 2
ENDIF

SELECT 2

@ 21.20 SAY ' CONTINUAR EN CONSULTAS (S / N) ' GET SAL PICTURE
T1 VALID(SAL $ 'SN')

READ

* SETCOLOR('B/B+')
ENDDO
CLOSE ALL
RESTORE SCREEN FROM PA
* RESTSCREEN(12,55,19,67,PAN,IV)
RETURN

* Programa consultar libros por tema.
* Nombre del programe fuente. COLI_TEM.PRG.
* Modificado 30-03-2001.

* SET COLOR TO
CLEAR
SET WRAP ON
SET SCOREBOARD OFF
SET CURSOR ON

# define ESCAPE 27
# define AVPAG 3
# define REPAG 18

SAL='S'
FIL=4
COL=5
NUM_TEMA+1

SELECT 1
USE LIB_BIB INDEX CLASI_LI

SELECT 2
USE COLI_AUX

SELECT 3
USE TEMAS INDEX TEM_TEM,CLASIFI

SET ORDER TO 1
SET ORDER TO 2

```

Lista de los programas fuente del sistema

```

@ 1.1 to 23.78 Double
@ 0.18 SAY *** CONSULTAS DE LIBROS POR TEMA ***
@ 24.3 SAY 'AV_PAG = Siguiente RE_PAG = Anterior ESC =
Salir

DO WHILE 'SAL#S'

RELEASE MTEMA

@ 4.5 CLEAR TO 21,77

MTEMA=SPACE(50)

@ 4.5 SAY 'TEMA . ' GET MTEMA PICTURE '@!'
READ
IF EMPTY(MTEMA)
CLEAR
SELECT 2
ZAP

SET RELATION TO
CLOSE ALL
RESTORE SCREEN FROM PA
RETURN

ENDIF

MTEMA=ALLTRIM(MTEMA)
NUM_CAR=LEN(MTEMA)

SET ORDER TO 1
SEEK MTEMA

I=0
K=0

COMPARA=MTEMA
MCLASI_LI=CLASI_LI
COMPARA=LEFT(MTEMA,NUM_CAR)
IF COMPARA=MTEMA

WHILE COMPARA=MTEMA
I++
K++
SELECT 2
APPEND BLANK
REPLACE CLASI_LI WITH MCLASI_LI
SELECT 3
SKIP
COMPA_III=TEMA
COMPARA=LEFT(COMPA_III,NUM_CAR)
MCLASI_LI=CLASI_LI
ENDDO

@ 2.68 SAY '
@ 2.69 SAY ' PICT 99'

```

Anexo 1

```

SELECT 2

GO TOP
J=0
K=1

SET RELATION TO CLASI_LI INTO LIB_BIB

WHILE I > J AND K <= 0

SET CURSOR OFF

@ 2.65 SAY K PICT '99'

@ FIL+1.5 SAY "CLAVE DE LIBRO "
@ FIL+2.5 SAY "AREA."
@ FIL+3.5 SAY "TITULO "
@ FIL+4.5 SAY "AUTOR PERSONAL "
@ FIL+5.5 SAY "EDITORIAL "
@ FIL+6.5 SAY "AUTOR CORP. "
@ FIL+7.5 SAY "No PAGINAS "
@ FIL+8.5 SAY "EJEMPLARES "
@ FIL+9.5 SAY "AÑO DE PUB. "
@ FIL+10.5 SAY "LUGAR DE PUB. "
@ FIL+1.26 SAY LIB_BIB-> CLASI_LI
@ FIL+2.11 SAY LIB_BIB-> AREA
@ FIL+3.25 SAY LIB_BIB-> TITULO
@ FIL+4.25 SAY LIB_BIB-> AUT.PER
@ FIL+5.25 SAY LIB_BIB-> EDITORIAL
@ FIL+6.25 SAY LIB_BIB-> AUT COR
@ FIL+7.25 SAY LIB_BIB-> NO_PAG

@ FIL+8.25 SAY LIB_BIB-> CANTIDAD
@ FIL+9.25 SAY LIB_BIB-> ANO_PUB
@ FIL+10.25 SAY LIB_BIB-> LUGAR

MCLASI_LI=CLASI_LI

SELECT 3

SET ORDER TO 2

SEEK MCLASI_LI

encuentran los temas de cada libro. Se banen hasta 5 temas por libros.
IF FOUND().

LIN=16

COMPA_II=MCLASI_LI

WHILE COMPA_II=MCLASI_LI

@ LIN.COL SAY "TEMA "
@ LIN.COL+5 SAY NUM_TEMA PICTURE "9"
@ LIN.COL+7 SAY TEMA PICTURE "I"

NUM_TEMA=NUM_TEMA+1

LIN=LIN+1

SKIP

COMPA_II=CLASI_LI

ENDDO

ENDIF

TECLA=INKEY(0)

DO CASE

```

Lista de los programas fuente del sistema

```

CASE Tecla ** AVPAG

SELECT 2
SKIP
J++
K++
@ 16.5 CLEAR TO 22.77
CASE Tecla ** REPAG
SELECT 2
SKIP-1
J-
K-
@ 18.5 CLEAR TO 22.77

CASE Tecla ** ESCAPE

SELECT 2

ZAP

SET RELATION TO

RESTORE SCREEN FROM PA

RETURN

OTHERWISE
RESTORE SCREEN FROM PA
RETURN

ENDCASE

NUM_TEMA=1

ENDDO

ELSE

SET CURSOR OFF

@ 21.10 SAY " NO EXISTEN TEMAS EN LA BASE << CTRL >> PARA
CONTINUAR

INKEY(0)

ENDIF

@ 21.7 CLEAR TO 21.75

@ 21.20 SAY " CONTINUAR EN CONSULTAS (S / N) " GET SAL PICTURE
(" VALID(SAL $ 'SN')

READ

@ 2.7 CLEAR TO 2.9
@ 2.11 CLEAR TO 2.12

SET CURSOR ON

SELECT 2

ZAP

SELECT 3

ENDDO

SET RELATION TO
CLOSE ALL
RESTORE SCREEN FROM PA

RETURN

* Programa consultar libros por título.
* Nombre del programa fuente: COLI_TIT.PRG.
* Modificado: 30-03-2001.

```

Anexo I

```

SET COLOR TO
CLEAR
SET SOFTSEEK ON
SET WRAP ON

SAL=SPACE(1)
SAL="S"

USE LIB BIB INDEX TIT LIB

@ 1.1 to 23.78 Double
@ 0.18 SAY "" CONSULTAS DE LIBROS POR TITULO ""
@ 24.15 SAY 'PRESIONE LA TECLA <<< ESC >>> PARA SALIR'

DO WHILE SAL="S"
@ 3.5 CLEAR TO 21,77

RELEASE MTITULO
MTITULO=SPACE(40)
@ 5.5 SAY TITULO 'GET MTITULO PICTURE '@'
READ

IF EMPTY(MTITULO)
CLEAR
CLOSE ALL
RESTORE SCREEN FROM PA
RETURN
ENDIF

*El primer parámetro es la clave por la que se busca, 1 el segundo es el
campo que se fija el tercero y el cuarto son el rango de campos que se
muestran.

CONSULTAS(MTITULO,3,3,3)

@ 5.5 SAY TITULO "
@ 6.5 SAY "AREA "
@ 7.5 SAY "CLASIFICACION "
@ 8.5 SAY "AUTOR PERSONAL "
@ 9.5 SAY "EDITORIAL "
@ 10.5 SAY "AUTOR CORP "
@ 11.5 SAY "No PAGINAS "
@ 12.5 SAY "EJEMPLARES "
@ 13.5 SAY "ANO DE PUB "
@ 14.5 SAY "LUGAR DE PUB. "

@ 5.28 SAY TITULO
@ 6.11 SAY AREA
@ 7.28 SAY CLASI LI
@ 8.28 SAY AUT PER
@ 9.28 SAY EDITORIAL
@ 10.28 SAY AUT CDR
@ 11.28 SAY NO PAG
@ 12.28 SAY CANTIDAD
@ 13.28 SAY ANO PUB
@ 14.28 SAY LUGAR
@ 21.20 SAY 'CONTINUAR EN CONSULTAS (S / N) 'GET SAL PICTURE
""VALIDISAL $ 'S'N'
READ
ENDDO
CLOSE ALL
RESTORE SCREEN FROM PA
RETURN

* Programa para consultar los usuarios de la biblioteca.
* Nombre del programa fuente. CON_USU.PRG
* Última modificación. 30-03-2001.

SET COLOR TO
CLEAR
SET WRAP ON
SET SOFTSEEK ON
USE USUARIO INDEX NOM_USI

SAL=SPACE(1)
SAL="S"

@ 1.1 to 23.78 Double
@ 0.15 SAY "" USUARIOS DE LA BIBLIOTECA ""
@ 24.15 SAY 'PRESIONE LA TECLA <<< ESC >>> PARA SALIR'
DO WHILE SAL="S"

RELEASE MCLA
@ 5.5 CLEAR TO 21,77
MCLA=SPACE(48)

@ 5.5 SAY 'NOMBRE DE USUARIO 'GET MCLA PICTURE '@'

```

Lista de los programas fuente del sistema

```

READ
IF EMPTY(MCLA)
CLEAR
CLOSE ALL
RESTORE SCREEN FROM PA
RETURN
ENDIF

CONSULTAS(MCLA,1,1,2)

@ 5.5 SAY 'NOMBRE "
@ 5.25 SAY NOM_USU
@ 6.5 SAY "CLAVE "
@ 6.25 SAY CLA_USU
@ 7.5 SAY "SEMESTRE "
@ 7.25 SAY SEM
@ 8.5 SAY "GRUPO "
@ 8.25 SAY GPO
@ 9.5 SAY "TURNO "
@ 9.25 SAY TURNO
@ 10.5 SAY "SEXO "
@ 10.25 SAY SEXO
@ 21.20 SAY 'CONTINUAR LAS CONSULTAS (S / N) 'GET SAL
PICTURE ""VALIDISAL $ 'S'N'
READ
ENDDO
CLOSE ALL
RESTORE SCREEN FROM PA
RETURN

* Programa para devolver los libros prestados
* Nombre del programa fuente. DEV_LIB.PRG
* Modificación: el 14 de agosto de 1997

```

```

SET COLOR TO
CLEAR

*SET DELETE ON

SET SOFTSEEK ON
SET WRAP ON

SAL=SPACE(1)
SAL="S"

* Cuando la base de datos LIB_PRE.DBF se queda vacía se daña su archivo
índice. Si después se adicionan nuevos registros (se prestan libros) se
necesario reindexar ya que de lo contrario al momento de utilizar el programa
DEV_LIB.PRG, el cual utiliza un archivo índice, no muestra ningún registro.

SELECT 1
USE STADIST
SELECT 2

USE PRE_LIB INDEX CVE_USU,NOM_USU

* REINDEX
* DO WHILE SAL="S"

@ 1.1 to 23.78 Double
@ 0.18 SAY "" DEVOLUCION DE LIBROS ""
@ 24.15 SAY 'PRESIONE LA TECLA <<< ESC >>> PARA SALIR'

DO WHILE SAL="S"
RELEASE MCLA_USU
@ 4.8 CLEAR TO 21,77

MCLA_USU=0
@ 5.8 SAY 'CLAVE DE USUARIO : 'GET MCLA_USU PICTURE
'9999999999'
READ
IF EMPTY(MCLA_USU)
CLEAR
PACK
REINDEX
CLOSE ALL
RESTORE SCREEN FROM PA
RETURN
ENDIF

* El primer parámetro es la clave por la que se busca, el segundo es el
campo que se fija el tercero y el cuarto son el rango de campos que se
muestran.

IF LASTREC() <<= 0
CONSULTAS(MCLA_USU,1,1,7)

IGUA_VAR()

```

Anexo 1

```

@ 5.8 SAY 'CLAVE DE USUARIO .'
@ 6.8 SAY 'TITULO .'
@ 7.8 SAY 'CLASIFICACION .'
@ 8.8 SAY 'NOMBRE USUARIO .'
@ 9.8 SAY 'FECHA DEVOLUCION .'
@ 10.8 SAY 'GRUPO .'
@ 11.8 SAY 'SEMESTRE .'
@ 12.8 SAY 'TURNO .'
@ 13.8 SAY 'SEXO .'

@ 5.28 SAY 'MCLA_USU'
@ 6.18 SAY 'MTIT_LIB'
@ 7.28 SAY 'MCLASI_LI'
@ 8.28 SAY 'MNOM_USU'
@ 9.28 SAY 'MPEC_DEV'
@ 10.28 SAY 'MGPO'
@ 11.28 SAY 'MSEM'
@ 12.28 SAY 'MTURNO'
@ 13.28 SAY 'MSEXO'

MULTA+DATE() - FEC_DEV
IF MULTA > 0
  TONE(500.9)
@ 15.45 SAY MULTA PICTURE '999' COLOR 'B+'
@ 15.50 SAY 'PESOS DE MULTA' COLOR 'B+'
ENDIF

@ 21.20 SAY ' DATOS CORRECTOS (S/N) 'GET SAL PICTURE '1'
VALID(SAL $ 'SN')
  READ
  IF SAL='S'

  SELECT 1
  APPEND BLANK
  REPLACE CLASI_LI WITH MCLASI_LI
  REPLACE CLA_USU WITH MCLA_USU
  REPLACE GPO WITH MGPO
  REPLACE SEM WITH MSEM
  REPLACE SEXO WITH MSEXO
  REPLACE NUM_AREA WITH MNUM_AREA
  REPLACE TURNO WITH MTURNO

  SELECT 2
  DELETE
  PACK
  ENDF

* Se vuelve asignar 'S' a la variable 'SAL' para sobrescribir en pantalla la 'S'
ya que permanece la 'N' cuando se lee la 'N' en la opcion datos no son
correctos (sin).

ELSE
@ 21.10 SAY ' BASE DE DATOS VACIA <<< CTRL >>> PARA
CONTINUAR
INKEY(0)
@ 21.8 CLEAR TO 21.77
ENDIF

SAL='S'

@ 21.20 SAY ' CONTINUAR LA DEVOLUCION (S/N) 'GET SAL PICTURE
'1' VALID(SAL $ 'SN')
  READ
  ENDDO
  PACK
  REINDEX
  CLOSE ALL
  RESTORE SCREEN FROM PA
  RETURN

* Programa para indexar la base de datos.
* Programa: INDEXA.PRG.
* Modificado: 30-03-01.

USE e:\c:\piper5.0\tareas\LIB.BIB
INDEX ON AUT_PER TO AUT_PER
INDEX ON TITULO TO TIT_LIB

INDEX ON CLASI_LI TO CLASI_LI

INDEX ON CLASI_LI + STR(LIB_CONS) TO CLASI_LI

USE e:\c:\piper5.0\tareas\USUARIO
INDEX ON CLA_USU TO CVE_USI
INDEX ON NOM_USU TO NOM_USI

USE e:\c:\piper5.0\tareas\PRE_LIB
INDEX ON CLA_USU TO CVE_USI
INDEX ON NOM_USU TO NOM_USI

INDEX ON NUM_LIB TO NUM_LIB
INDEX ON TIT_LIB TO TIT_LIB

```

Lista de los programas fuente del sistema

```

* INDEX ON STR(CLA_USU) + STR(NUM_LIB) TO DOBLE
* INDEX ON TIT_LIB + DTOC(FEC_DEV) TO FEC_DEV
* La función DTOC sirve para convertir una fecha a caracteres y contrano es
DTOS.

USE e:\c:\piper5.0\tareas\AREA

INDEX ON NUM_AREA TO NUM_AREA

USE e:\c:\piper5.0\tareas\TEMAS
INDEX ON TEMA TO TEM_TEM
INDEX ON CLASI_LI TO CLASIFI

USE e:\c:\piper5.0\tareas\COLL_AUX
INDEX ON CLASI_LI TO CLASI_CO
CLOSE ALL
RETURN

* Programa principal

* Nombre del programa fuente: LIBROS.PRG
* Sistema para la automatización de la biblioteca del celis 92.
* Modificado: 30-03-2001.

CLEAR
PRIVATE PA
PRIVATE PAN_IV
SET ESCAPE ON
SET WRAP ON
SET COLOR TO B/B+
@ 0.0,24.80 BOX '*****'
SET COLOR TO GR+/B
@ 1.1 to 23.78 Double
@ 0.20 SAY '*** SECRETARIA DE EDUCACION PUBLICA ***'
@ 24.10 SAY 'PRESIONE LA TECLA * CHR(24)+CHR(25)* PARA
MOVERSE O <<< ESC >>> PARA SALIR'
* @ 9.31 SAY ' MENU PRINCIPAL '
* PAN_IV=SAVESCREEN(11.49,19.70)

DO RRL

DO LLAVE

F=.T.
@ 9.32 SAY 'MENU PRINCIPAL'
DO WHILE F=.T

  SET COLOR TO N
  @ 11.34,19.47 BOX ' 00000 '
  SET COLOR TO bg/B,G+/BG+
  @ 10.32 to 18.45 DOUBLE
  SET COLOR TO gr+/B+,GR+/BG
  SET MESSAGE TO 22 CENTER
  @ 11.34 PROMPT 'MOVIMIENTO ' MESSAGE 'MOVIMIENTOS DE
  LIBROS'
  @ 12.34 PROMPT 'CONSULTA ' MESSAGE 'LIBROS DISPONIBLES'
  @ 13.34 PROMPT 'LISTADOS ' MESSAGE 'LISTADOS DE
  MOVIMIENTOS'
  @ 14.34 PROMPT 'ALTAS ' MESSAGE 'ALTA DE MATERIAL NUEVO'
  @ 15.34 PROMPT 'BAJAS ' MESSAGE 'BAJA DE MATERIAL'
  @ 16.34 PROMPT 'CAMBIOS ' MESSAGE 'MOVIMIENTO DE
  MATERIAL'
  @ 17.34 PROMPT 'UTILERIA ' MESSAGE 'ORDENAR BASE DE DATOS'
  MENU TO ope
  SAVE SCREEN TO PA

DO CASE
CASE LASTKEY()=27
  G=.T.
  DO WHILE G=.T.
  SET COLOR TO bg+/B
  @ 7.8 SAY 'SALIR'
  SET COLOR TO N
  @ 9.8,12.14 BOX ' 00000 '
  SET COLOR TO bg/B,G+/BG+
  @ 8.4 to 11.13 DOUBLE
  SET COLOR TO bg/B+,GR+/BG
  SET MESSAGE TO B
  @ 9.5 PROMPT ' NO ' / MESSAGE ' * CANCELAR LA SALIDA ' /
  @ 10.5 PROMPT ' SI ' / MESSAGE ' / * SALIR DEL SISTEMA ' /
  MENU TO SALIR

DO CASE
CASE SALIR=1
  RESTORE SCREEN FROM PA
  G=.F.
CASE SALIR=2
  SET COLOR TO
  CLEAR
  QUIT

```


Anexo I

```

SET COLOR TO bg/B.G+/BG+
16.55 to 19.64 double
SET COLOR TO gr+/B+.GR+/BG
17.58 PROMPT 'LIBROS '
18.58 PROMPT 'USUARIOS'
MENU TO OPC_VI
RESTSCREEN(18.55,20.65,PAN_IV)
RETURN OPC_VI
-----
function UTILERIA()
LOCAL OPC,VII
PAN_IV=SAVESCREEN(18.55,20.65)

SET COLOR TO N
17.57,20.65 BOX ' 00000 '
SET COLOR TO bg/B.G+/BG+
18.53 to 19.64 double
SET COLOR TO gr+/B+.GR+/BG

17.58 PROMPT 'INDEXAR '
18.58 PROMPT 'USUARIOS'
MENU TO OPC_VI

RESTSCREEN(16.55,20.65,PAN_IV)
RETURN OPC_VI
-----
FUNCTION wmshow(table,fa,col,tempo,control)
PRIVATE i,c

DO CASE
CASE UPPER(control)='A'
FOR c=1 TO LEN(table)
FOR i=1to c-1 TO fa
17.57 say table[i+c:fa]
NEXT
INKEY(!tempo)
NEXT
ENDCASE
RETURN[]
-----
* Inicializa variables de memoria con el contenido de los campos. las
variables de memoria tendran el mismo nombre de los campos mas la letra
"M".
FUNCTION IGUA_VAR()
A=FCOUNT()
FOR i=1 TO A
IF TYPE(FIELD(i)) <> "M"
NOMCAM=FIELDNAME(i)
NOMVAR="M"+NOMCAM
PUBLIC &NOMVAR
&NOMVAR=&NOMCAM
ENDIF
NEXT
-----
* Inicializa variables de memoria para cada campo del archivo. las variables
de memoria tendran el mismo nombre de los campos mas la letra "m".
FUNCTION ini_VAR()
A=FCOUNT()
FOR i=1 TO A
IF TYPE(FIELD(i)) <> "M"
NOMCAM=FIELD(i)
NOMVAR="M"+NOMCAM
PUBLIC &NOMVAR
DO CASE
CASE TYPE(FIELD(i))="C"
&NOMVAR=SPACE(LEN(&NOMCAM))
CASE TYPE(FIELD(i))="N"
&NOMVAR=""
CASE TYPE(FIELD(i))="D"
&NOMVAR=CTOD("")
CASE TYPE(FIELD(i))="L"
&NOMVAR=.F.
ENDIF
ENDIF
NEXT
-----
* Elimna variables de memoria creadas por los campos de archivo y que
fueron inicializadas o igualadas por los procedimientos igual_var o ini_var.
FUNCTION LIB_VAR()
A=FCOUNT()
FOR i=1 TO A
IF TYPE(FIELD(i)) <> "M"

```

Lista de los programas fuente del sistema

```

NOMVAR="M"+FIELD(i)
RELEASE &NOMVAR
ENDIF
NEXT
RELEASE NOMVAR
-----
* Sustituye los campos del archivo abierto por el contenido de los variables
correspondientes y que fueron creadas o igualadas por los procedimientos
igual_var o ini_var
FUNCTION REPL_VAR()
A=FCOUNT()
FOR i=1 TO A
IF TYPE(FIELD(i)) <> "M"
NOMCAM=FIELD(i)
NOMVAR="M"+NOMCAM
IF TYPE(&NOMVAR) <> "U"
REPLACE &NOMCAM WITH &NOMVAR
ENDIF
ENDIF
NEXT
-----
FUNCTION REVIS(PALABRA)
IF EMPTY(PALABRA)
RETURN(.F.)
ELSE
RETURN(.T.)
ENDIF
-----
function CONSULTA(CLAVE,FLIA,CAMPO_I,CAMPO_II)

# define INTRO 13
# define ESCAPE 27
# define FLECHAR 5
# define FLECHAB 24
# define FLECHAI 19
# define FLECHAD 4
# define AVPAG 3
# define REPAG 18
# define INICIO 1
# define FIN 8
# define CTRL_FLECHAI 26
# define CTRL_FLECHAD 2
# define CTRL_AVPAG 30
# define CTRL_REPAG 31
# define CTRL_INICIO 29
# define CTRL_FIN 23

Arriba=9
Izquierda=15
Abajo=19
Derecha=65

SEEK CLAVE

b.=TBrowseDB(Arriba,Izquierda,Abajo,Derecha)

b.headSep.:='T'
b.colSep.:=' '
b.colorSpec.:='NW,N/BG,BW,B/BG,BW,B/BG,RW,B/R'

21.17 SAY 'PRESIONE LA TECLA << CTRL >> PARA SELECCIONAR'

FOR n.=CAMPO_I TO CAMPO_II / FCOUNT() /
Columna.=TBColumnNew(FieldName(n),FieldWidth(FieldName(n),
Select(i))
b.addColumn(Columna)
NEXT

b.freeze.:='FLIA
CurColor:=SETCOLOR("N")

Arriba+1,Izquierda+1 CLEAR TO Abajo+1,Derecha+1
SETCOLOR("WW")
Arriba,Izquierda CLEAR TO Abajo,Derecha
SETCOLOR(CurColor)

CurCursor:=SETCURSOR(0)

Mes.=T.
WHILE (Mes)

IF b.colPos <= b.freeze + 1
b.colPos := b.freeze + 1
ENDIF

WHILE !b.stabize()
Tecla := INKEY()

```

Anexo I

```

IF Tecta != 0
EXIT
ENDIF
ENDOF
ENDDO

IF b.stable
IF b.htTop_OR_b.htBottom
TONE(125,0)
ENDIF
Tecta := INKEY(0)
ENDIF

DO CASE
CASE Tecta == FLECHAB
b.down()
CASE Tecta == FLECHAR
b.up()
CASE Tecta == AVPAG
b.pageDown()
CASE Tecta == REPAG
b.pageUp()
CASE Tecta == CTRL_REPAG
b.goTop()
CASE Tecta == CTRL_AVPAG
b.goBottom()
CASE Tecta == FLECHAD
b.right()
CASE Tecta == FLECHA
b.left()
CASE Tecta == INICIO
b.home()
CASE Tecta == FIN
b.end()
CASE Tecta == CTRL_FLECHA
b.pantLeft()
CASE Tecta == CTRL_FLECHAD
b.pantRight()
CASE Tecta == CTRL_INICIO
b.pantHome()
CASE Tecta == CTRL_FIN
b.pantEnd()
CASE Tecta == ESCAPE
Mas = T
CASE Tecta == INTRO
Mas = F
ENDCASE
ENDDO
SET COLOR TO
setcursor(1)
@ 9,13 CLEAR TO 21,77
RETURN

-----
FUNCTION CONSECU()
COMPARA=MCLASI_LI
WHILE COMPARA=MCLASI_LI
SKIP
COMPARA=CLASI_LI
ENDDO
RETURN

-----
FUNCTION CUENTA()
LOCAL K
K=0
COMPARA=MCLASI_LI
WHILE COMPARA=MCLASI_LI
SKIP
K++
COMPARA=CLASI_LI
ENDDO
SEEK MCLASI_LI
RETURN(K)

----- PROCEDIMIENTOS -----

PROCEDURE LLAVE
LOCAL
PAN_I:=SAVESCREEN(11,4,15,17),PAN_U:=SAVESCREEN(15,5,18,24)
CONTA=1
CLAVE=SPACE(4)
DO WHILE CONTA < 4
SET SCOREBOARD OFF
SET COLOR TO N
@ 12,6,14,17 box ' 00000 '
SET COLOR TO bp/B.G+BG+
@ 11,5 to 13,16 double
SET COLOR TO gr+B+GR+BG
@ 12,8 SAY "CLAVE "
@ 12,12 SET CLAVE PICT @1" COLOR W+W
READ

```

Lista de los programas fuente del sistema

```

@ 12,6 CLEAR TO 12,10

IF CLAVE == "PASO"
RESTSCREEN(11,4,15,17,PAN_I)
RETURN
ELSE
CONTA=CONTA+1
IF CONTA=4
SET COLOR TO
CLEAR
QUIT
ELSE

RESTSCREEN(11,4,15,17,PAN_U)
SET COLOR TO N
@ 16,6,18,23 box ' 00000 '
SET COLOR TO bp/B.G+BG+
@ 15,5 to 17,22 double
SET COLOR TO gr+B+GR+BG
SET CURSOR OFF

@ 16,8 SAY '¡NTE DE NUEVO!' COLOR 'R'
TONE(500,10)
INKEY(1)
RESTSCREEN(15,5,18,24,PAN_U)
ENDIF
ENDIF
ENDDO
SET COLOR TO
SET SCOREBOARD ON
CLEAR
QUIT

-----
PROCEDURE RRL
set color to n/r
@ 4,21,8,59 box ' 00000 '
SET COLOR TO n/r
DECLARE TABLA[S]
TABLA[1]=E"#####"
TABLA[2]= " C.E.T.I.S. 92 "
TABLA[3]= " SISTEMA PARA CONTROL DE BIBLIOTECA "
TABLA[4]= "
TABLA[5]=E"#####"
SET CURSOR OFF
window(TABLA,7,19,0.05,'A')
INKEY(1)
SET CURSOR ON
RETURN

* Programa para el préstamo de libros.
* Nombre del programa fuente: PRE_LIB.PRG.
* Modificación: 30-03-2001.

CLEAR

SET ESCAPE ON

SET WRAP ON

SAL=SPACE(1)

SAL="S"

SELECT 1

USE PRE_LIB INDEX CVE_USU|NOM_USU
SELECT 2
USE LIB_BIB INDEX CLASI_LI
SELECT 3
USE USUARIO INDEX CVE_USI

@ 1,1 to 23,78 Double
@ 0,20 SAY "----- PRESTAMO DE LIBROS -----"
@ 24,10 SAY " PRESIONE LA TECLA " CHR(24)+CHR(25)+ " PARA
MOVESE
DO WHILE SAL="S"

@ 5,5 CLEAR TO 21,77
MCLA_USU=0

@ 5,5 SAY "CLAVE DEL ALUMNO : GET MCLA_USU PICTURE
"9999999999
READ

```

Anexo I

```

* CLEAR GETS
  IF EMPTY(MCLA_USU)
    SELECT 1
    REINDEX
    CLEAR
    CLOSE ALL
  RESTORE SCREEN FROM PA
  RETURN
ENDIF

SELECT 1
SEEK MCLA_USU
  IF .NOT. FOUND()

    SELECT 3
    SEEK MCLA_USU
    IF FOUND()
      MNOM_USU=NOM_USU
      IF CDOW(DATE())="Viernes"
        MFEC_DEV=DATE()+3
      ELSE
        MFEC_DEV=DATE()+2
      ENDIF
      MSEM=SEM
      MSEXO=SEXO
      MGPO=GPO
      MTURNO=TURNO

      @ 8.5 SAY 'NOMBRE'
      @ 8.26 SAY 'MNOM_USU'
      @ 7.5 SAY 'FECHA DE PRESTAMO'
      @ 7.26 SAY 'DATE()'
      @ 8.5 SAY 'FECHA DE DEVOLUCION'
      @ 8.26 SAY 'MFEC_DEV'
      @ 9.5 SAY 'GRUPO'
      @ 9.26 SAY 'MGPO'
      @ 10.5 SAY 'TURNO'
      @ 10.26 SAY 'MTURNO'
      @ 11.5 SAY 'SEMESTRE'
      @ 11.26 SAY 'MSEM'

    SELEC 2

    MCONSE=0
    MCLASI_LI=SPACE(10) /* VARIABLE DE LIB_BIB */

    @ 12.5 SAY 'CLASIFICACION LIBRO:' GET MCLASI_LI PICT '99'
    READ

    SEEK MCLASI_LI
    IF FOUND()
      MTITULO=TITULO
      MNUM_AREA=NUM_AREA

      @ 13.5 SAY 'TITULO DE LIBRO' + MTITULO
      @ 14.5 SAY 'NUMERO CONSECUTIVO' GET MCONSE PICT '999'
      READ

    SN=SPACE(1)
    SN='S'
    @ 21.15 SAY ' DATOS CORRECTOS (S/N)? ' GET SN PICTURE '*'
    VALID(SN'SN')
    READ
    IF SN='S'

      SELECT 1
      @ 21.15 CLEAR TO 21,77

      APPEND BLANK

      REPLACE CLASI_LI WITH MCLASI_LI
      REPLACE TIT_LIB WITH MTITULO
      REPLACE CLA_USU WITH MCLA_USU
      REPLACE NOM_USU WITH MNOM_USU
      REPLACE GPO WITH MGPO
      REPLACE SEM WITH MSEM
      REPLACE TURNO WITH MTURNO
      REPLACE FEC_PRE WITH DATE()
      REPLACE FEC_DEV WITH MFEC_DEV
      REPLACE SEXO WITH MSEXO
      REPLACE NUM_AREA WITH MNUM_AREA
      REPLACE NUM_CONS WITH MCONSE

    ENDIF
  ELSE

    @ 21.15 SAY ' LA LIBRO NO EXISTE EN LA BASE'
    INKEY(2)
    ENDIF

  ELSE

    @ 21.15 SAY ' EL ALUMNO NO EXISTE EN LA BASE'

```

Lista de los programas fuente del sistema

```

INKEY(2)
ENDIF
ELSE

  TONE(500,9)
  @ 15.50 SAY 'EL ALUMNO DEBE UN LIBRO' COLOR "W*"
  INKEY(3)
  @ 15.50 CLEAR TO 15,77
  ENDIF

RELEASE ALL LIKE M /* Borrar de la memoria y de la pantalla todas las
*/
/* las variables que empiecen con M */

SELECT 3
@ 21.15 CLEAR TO 21,77
@ 21.15 SAY ' CONTINUAR PRESTANDO LIBROS (S/N) ' GET SAL
PICTURE ' * VALID(SAL,'SN')
READ
  ENDDO
  SELECT 1 /* SE DEBEN REORDENAR LOS INDICES YA
QUE DE DAN DE BAJA */
  REINDEX
  CLOSE DATABASES
  RESTORE SCREEN FROM PA
  RETURN

* function MENSAJE()
* PANTA=SAVESCREEN(14,48,17,76)
* SET COLOR TO N /*
@ 15.50,18,76 BOX ' 000000 '
* SET COLOR TO bg;B.G+BG+
@ 14,48 to 16,75 double
* SET COLOR TO gr;B+.GR+BG

@ 15.49 SAY 'EL ALUMNO DEBE UN LIBRO'
* RESTSCREEN(14,48,17,76,PANTA)
* RETURN

* Programa para imprimir el inventario de libros.
* Nombre del programa fuente: RELI_INV.PRG.
* Modificado: 30-03-2001.

CLEAR

SET PRINTER OFF

SET MARGIN TO 4

SET SCOREBOARD OFF

SET SOFTSEEK ON

USE LIB_BIB INDEX CLASS LIB
* INDEX ON CLA_LIB TO CLA_LIB

SET COLOR TO RB+RB
@ 0,0,24,79 BOX '*****'
SET COLOR TO W+RB
@ 0,0 TO 24,79 DOUBLE
SET COLOR TO GR+RB

SET COLOR TO R+R
@ 0,0,24,79 BOX '*****'
SET COLOR TO GR+R
@ 2,2 TO 22,77 DOUBLE
@ 4,18 TO 7,62 DOUBLE
@ 8,16 TO 11,62 DOUBLE
@ 5,19 SAY ' VERIFIQUE QUE LA IMPRESORA TENGA PAPEL '
@ 6,19 SAY ' Y SE ENCUENTRE EN LINEA '
@ 9,19 SAY ' NOTA: EL PAPEL DEBE DE SER CON EL FORMATO '
@ 10,19 SAY ' DE 132 CARACTERES '
@ 14,10 SAY 'EJEMPLO:'
@ 13,28 SAY 'AAAAAAA 132 AAAAAA '
@ 14,28 TO 19,48
@ 21,38 SAY 'presione alguna tecla para continuar...'
wait
TECLA = LASTKEY()
IF TECLA = 21
  CLOSE DATABASES
  RESTORE SCREEN FROM PA
  RETURN
ENDIF

IF !SPRINTER()
  SET COLOR TO B/B
  @ 9,20,15,59 BOX '*****'
  SET COLOR TO W+B
  @ 9,20 TO 15,59 DOUBLE

```

Anexo I

```

SET COLOR TO W+R
@ 12.23 SAY 'LA IMPRESORA NO ESTA PREPARADA'
@ 13.23 SAY ' VERIFIQUE LAS CONEXIONES'
@ 21.38 SAY 'presione alguna tecla para continuar...'
wait

CLOSE DATABASES
RESTORE SCREEN FROM PA
RETURN
ENDIF

SET TALK OFF
TOTAL=0
N_PAG=0

DO WHILE .NOT. EOF()
    K1=3
    N_PAG=N_PAG+1
    MAT_I=MATERIA
    SUB_TOT=CANTIDAD

    SET PRINTER ON

    ? S.E.P
    S.E.I.T.
    ? *
    * CENTRO DE ESTUDIOS TECNOLOGICOS Industrial
    y de Servicios 92*
    ? *
    * AREA DE ACERVO BIBLIOTECARIO
    FECHA " ,*(DATE)
    ? *
    * DISTRIBUCION DE LIBROS"
    ? *
    ? *COORDINACION ESTATAL, ESTADO DE MEXICO
    P g. No."N_PAG
    ?
    *
    -----
    ? *
    ? *
    ? * TITULO
    EDITORIAL EJEMP* AUTOR
    ?
    *
    -----
    ? *
    ? *
    ? * AREA DE "MAT_I
    ? *
    ? * "TITULO," "AUT_PER," "EDITORIAL," "CANTIDAD
    SKIP
    MAT_II=MATERIA

    DO WHILE MAT_II=MAT_II
    ? * "TITULO," "AUT_PER," "EDITORIAL," "CANTIDAD
    CAN_II=CANTIDAD
    SUB_TOT=SUB_TOT+CAN_II
    SKIP
    MAT_II=MATERIA
    **K1

    IF K1>41
    N_PAG=N_PAG+1
    K1=3
    EJECT

    ? S.E.P
    S.E.I.T.
    ? *
    * CENTRO DE ESTUDIOS TECNOLOGICOS Industrial
    y de Servicios 92*
    ? *
    * AREA DE ACERVO BIBLIOTECARIO
    FECHA " ,*(DATE)
    ? *
    * DISTRIBUCION DE LIBROS"
    ? *
    ? *COORDINACION ESTATAL, ESTADO DE MEXICO
    P g. No."N_PAG
    ?
    *
    -----
    ? *
    ? *
    ? * TITULO
    EDITORIAL EJEMP* AUTOR
    ?
    *
    -----
    ? *
    ? *
    ? * AREA DE "MAT_I
    ? *
    * ELSE

```

Lista de los programas fuente del sistema

```

LOOP
ENDIF
ENDDO
? *
? *
TOTAL=TOTAL+SUB_TOT
EJECT
ENDDO

? *
? *
? *
TOTAL POR AREA. "SUB_TOT

? *
? *
? *
TOTAL DE LIBROS DE LA BIBLIOTECA.

SET PRINTER OFF
CLOSE ALL
RESTORE SCREEN FROM PA
RETURN

*Programa para mostrar los totales de libros.
*Nombre del programa fuente: RELI_TOT.PRG.
*Modificación: 30-03-2001

CLEAR
SET TALK OFF
SET PRINTER OFF
SET MARGIN TO 25
SET SCOREBOARD OFF
SET SOFTSEEK ON

USE LIB, BIB index cta, lb
* INDEX ON CLA, LIB TO CLA, LIB

TOT=0
MATE =array(27)
MATE = 518,517,531,539,574,577,578,652,661,664,741,865,900,901)
SET COLOR TO RB+RB
@ 0.0,24.79 BOX "*****"
SET COLOR TO W+R
@ 0.0 TO 24.79 DOUBLE
SET COLOR TO GR+RB

SET COLOR TO R+R
@ 0.0,24.79 BOX "*****"
SET COLOR TO GR+R
@ 2.2 TO 22.77 DOUBLE
@ 4.18 TO 7.62 DOUBLE
@ 8.18 TO 11.92 DOUBLE
@ 5.19 SAY ' VERIFIQUE QUE LA IMPRESORA TENGA PAPEL '
@ 6.19 SAY ' Y SE ENCUENTRE EN LINEA
@ 9.19 SAY ' NOTA. EL PAPEL DEBE DE SER CON EL FORMATO '
@ 10.19 SAY ' DE 132 CARACTERES
@ 14.10 SAY 'EJEMPLO'
@ 13.29 SAY 'AAAAAAAAA 132 AAAAAA'
@ 14.29 TO 19.48
@ 21.38 SAY 'presione alguna tecla para continuar...'
wait
TECLA = LASTKEY()
IF TECLA = 27
CLOSE DATABASES
RESTORE SCREEN FROM PA
RETURN
ENDIF

IF !SPRINTER()
SET COLOR TO B/B
@ 8.20,15.59 BOX "*****"
SET COLOR TO W+R
@ 9.20 TO 15.59 DOUBLE
SET COLOR TO W+R
@ 12.23 SAY 'LA IMPRESORA NO ESTA PREPARADA'
@ 13.23 SAY ' VERIFIQUE LAS CONEXIONES'
@ 21.38 SAY 'presione alguna tecla para continuar...'
wait
CLOSE DATABASES
RESTORE SCREEN FROM PA
RETURN
ENDIF

SET PRINTER ON

? * CENTRO DE ESTUDIOS TECNOLOGICOS Industrial y de Servicios 92*
? * Totales de Libros por Carrera"
?
?
? *
? *
? *
? * CLAVE CARRERA CANTIDAD *

```

Anexo I

Lista de los programas fuente del sistema

```
?  
-----  
TOTAL=0  
I=1  
DO WHILE NOT EOF()  
TOT=TOT+CANTIDAD  
MAT_I=MATERIA  
K1= T.  
  
DO WHILE K1  
  
SKIP  
MAT_II=MATERIA  
IF MAT_I=MAT_II  
TOT=TOT+CANTIDAD  
LOOP  
  
ELSE  
? MATE[I], " ", MAT_I, " ", TOT  
  
TOTAL=TOTAL+TOT  
TOT=0  
++  
K1= F  
ENDIF  
ENDDO  
ENDDO .  
  
?  
-----  
?* TOTAL ", TOTAL  
  
EJECT  
SET PRINTER OFF  
CLOSE ALL  
RESTORE SCREEN FROM PA  
RETURN
```

1. Cuando la letra inicial es una vocal, si la letra que sigue es:

Usar el número	b	d,k	l,m	n	p	r	s	u
	2	3	4	5	6	7	8	9

2. Cuando la letra inicial es una consonante, si la letra que sigue es:

Usar el número	a	e	i	o	r,s	u	y
	3	4	5	6	7	8	9

3. Cuando la letra inicial es una S, si la letra que sigue es:

Usar el número	a	ch	e	h	m	t	u
	2	3	4	5	6	7,8	9

4. Cuando la letra inicial es una Q, si la letra que sigue es:

Usar el número	a	e,i	i	o	r	y
	3	4	5	6	7	9

5. Cuando la letra inicial es una CH, si la letra que sigue es:

Usar el número	a-z
	9

Cuando se considere necesario para la tercera letra del apellido del autor o título, se utiliza la siguiente tabla:

Usar el número	a-d	e-h	i-l	m	n-q	r-t	u-w	x-z
	2	3	4	5	6	7	8	9

Nota: Cuando no encuentre en las tablas el número equivalente de una letra, use el número de la letra anterior.

CONCLUSION

El Sistema para la automatización de la biblioteca tiene como su principal función controlar el préstamo de libros, el cuál tiene aceptación tanto por el personal de la biblioteca como por los alumnos. Se ha disminuido el tiempo que el alumno ocupaba en realizar el trámite del préstamo de un libro, además como el sistema determina si el libro solicitado está disponible o no, el usuario no pierde tiempo llenando papeletas con libros que no están disponibles, por otra parte, el encargado conoce en forma inmediata qué libros tiene en existencia. Esto ha reducido en forma importante el tiempo ocupado para obtener un libro en préstamo y el trabajo estadístico ahora se realiza con mayor rapidez y precisión.

Otra ventaja del sistema implantado, es la facilidad para buscar temas específicos ya que el sistema cuenta con la opción de consulta por temas, lo cual es de gran utilidad para los alumnos que frecuentemente preguntan al encargado sobre un tema específico, con esta alternativa se puede saber en qué otro libro se encuentra un tema similar o igual. También los profesores del plantel se han visto beneficiados ya que tienen la oportunidad de conocer fácilmente los títulos de libros existentes que corresponden a las asignaturas que imparten, para después incorporar, los que sean útiles, a la bibliografía que será entregada al alumno.

En relación con el control de libros se tiene avances importantes, ya que en cualquier momento se puede generar información de aquellos maestros o alumnos que tienen libros de la biblioteca en préstamo y de forma precisa invitarlos a que devuelvan los libros. Anteriormente había robo de libros ya que el extravío de papeletas de los libros o bien el encargado olvidaba a qué docente le prestó libros, esto provoca que no se haga el reclamo. Con el uso del sistema prácticamente ha desaparecido el problema.

El sistema se puede mejorar en las etapas de recuperación de datos, para lograr esto únicamente se requiere instalar un lector óptico de caracteres a la computadora y configurarlo con el sistema. Esta simple modificación puede disminuir de manera considerable el tiempo que se emplea para el préstamo y devolución de libros por el hecho de evitar introducir por teclado las claves del libro y de usuario, el lector se encargaría de leer dichas claves al sistema.

Es conveniente seguir con interés el desarrollo de las herramientas CASE ya que seleccionando los programas adecuados y utilizandolos en el ciclo de vida del desarrollo de sistemas, se tienen ventajas realmente significativas tanto en el diseño, como en la generación de código.

BIBLIOGRAFIA

Análisis y diseño de sistemas, Kendall y Kendall. Editorial Printice Hall.

Diseño general de sistemas de información, John G. Burch, Gary Grudnitski. Editorial Noriega-Megabyte.

Análisis y diseño de sistemas de Información, Janes A. Send. Editorial Mc Graw Hill.

Diseño de base de datos, Gio Wiederhold. Editorial Mc Graw Hill

Análisis y diseño de sistemas de información, Jeffrey L. Whitten, Lonnie D. Bentley, Victor M. Barlow. Editorial Mc Graw Hill

Manual de organización del centro de estudios tecnológicos y de servicios, y del centro de bachillerato tecnológico industrial y de servicios. Editorial SEP.

Lineamientos generales para el funcionamiento de la biblioteca escolar. Reyes Vilchis Máximo. Editorial DGETI.

Clipper 5.01 avanzado, José A. Ramalho. Editorial Mc Graw Hill.

Clipper5 referencia rápida, Francisco Quirós Marin, Antonio Quirós Casado. Editorial Rama.

Librerías de clipper 5, Francisco Quirós Marin, Antonio Quirós Casado, Antonio Torres Lozano. Editorial Ra-ma.

Manual de bibliotecología, Josefa E. Sabor, Roberto V. Cagnoli, Lidya H. Revello, Susana Soto. Editorial Kapelusz mexicana.

Manual de técnicas de investigación para estudiantes de ciencias sociales, Ario Garza Mercado. Editorial Harla.

Turbo Pascal, Luis Joyanes Aguilar. Editorial Mc Graw Hill.

Fundamento de bases de datos, Henry F. Korth, Abraham Silberschatz. Editorial Mc Graw Hill.

Informática presente y futuro, Donald H. Senders. Editorial Mc Graw Hill.