

12
2ej.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Facultad de Ingeniería

SISTEMA INTEGRAL DE INFORMACION PARA LA MEDIATECA DEL
CENTRO DE ENSEÑANZA DE LENGUAS EXTRANJERAS

T E S I S

Que para obtener el Título de
INGENIERO EN COMPUTACION

presentan:

Ricardo Eduardo Coutiño Sánchez

Mauricio Soto Quintana



DIRECTOR DE TESIS: ING. ORLANDO ZALDIVAR ZAMORATEGUI

México D. F.

TESIS CON
FALLA DE ORIGEN

278642 1999



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

AGRADECIMIENTOS:

A la Universidad Nacional Autónoma de México,
en especial a la Facultad de Ingeniería por la
formación profesional que nos brindaron.

Al Centro de Enseñanza de Lenguas Extranjeras,
en particular a la Mediateca por la oportunidad de desarrollo
profesional.

Al Ing. Orlando Zaldívar Zamorategui por los conocimientos,
consejos y experiencias transmitidas.

A nuestros padres:

Daria Sánchez Anaya

María de Lourdes Quintana Ramos

Lázaro Soto Domínguez

por su cariño, comprensión y espíritu de superación que nos forjaron.

Y a nuestros hermanos y familiares por su apoyo.

Ricardo E. Coutiño Sánchez

Mauricio Soto Quintana

ÍNDICE.

INTRODUCCIÓN.	1
1. FUNDAMENTOS TEÓRICOS.	3
1.1 ESTUDIO INICIAL Y ANÁLISIS DEL SISTEMA.	3
1.1.1 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL.	9
1.1.2 ANÁLISIS DE LA PROBLEMÁTICA DE LA SITUACIÓN ACTUAL Y PROPUESTA DE SOLUCIÓN.	10
1.1.3 ANÁLISIS DE FACTIBILIDAD TÉCNICA DE LA PROPUESTA DE SOLUCIÓN.	10
1.2 ANÁLISIS Y ESPECIFICACIÓN ESTRUCTURADA.	11
1.2.1 EL DIAGRAMA DE FLUJO DE DATOS (DFD).	12
1.2.2 DICCIONARIO DE DATOS.	16
1.2.3 MINI-ESPECIFICACIONES.	18
1.2.4 TABLAS DE DECISIÓN.	18
1.3 DISEÑO ESTRUCTURADO.	19
1.3.1 LA CARTA DE ESTRUCTURA (CDE).	20
1.3.2 COHESIÓN.	23
1.3.3 ACOPLAMIENTO.	25
1.4 BASES DE DATOS.	26
1.4.1 DIAGRAMA ENTIDAD-RELACIÓN.	28
1.4.2 NORMALIZACIÓN.	31
1.5 LENGUAJES DE PROGRAMACIÓN.	31
1.6 MANUAL DEL USUARIO.	40
1.7 PRUEBAS Y CONFIABILIDAD DEL SISTEMA.	41
1.7.1 PRUEBA DE UNIDAD.	41
1.7.2 PRUEBA DE INTEGRACIÓN.	42
1.7.3 PRUEBA DE VOLUMEN.	42
1.7.4 PRUEBA DE VALIDACIÓN.	43
1.8 CALIDAD Y MANTENIMIENTO DEL SISTEMA.	44
1.7.5 CALIDAD DEL SISTEMA.	44
1.7.6 MANTENIMIENTO.	47
2. ESTUDIO INICIAL Y ANÁLISIS DEL <i>SIIMCELE</i>.	49
2.1 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL DE LA <i>MEDIATECA</i> .	49
2.2 ANÁLISIS DE LA PROBLEMÁTICA DE LA SITUACIÓN ACTUAL DE LA <i>MEDIATECA</i> Y PROPUESTA DE SOLUCIÓN.	54
2.3 ANÁLISIS DE FACTIBILIDAD TÉCNICA DE LA PROPUESTA DE SOLUCIÓN PARA LA <i>MEDIATECA</i> .	60

3. ANÁLISIS Y ESPECIFICACIÓN ESTRUCTURADA DEL SIIMCELE.	63
3.1 EL DFD DEL SIIMCELE.	63
3.2 DICCIONARIO DE DATOS DEL SIIMCELE.	71
3.3 MINI-ESPECIFICACIONES DEL SIIMCELE.	85
3.4 TABLAS DE DECISIÓN DEL SIIMCELE.	95
4. DISEÑO ESTRUCTURADO DEL SIIMCELE.	100
4.1 LA CDE DEL SIIMCELE.	100
4.2 COHESIÓN DEL SIIMCELE.	105
4.3 ACOPLAMIENTO DEL SIIMCELE.	106
5. BASE DE DATOS DEL SIIMCELE.	107
5.1 DIAGRAMA ENTIDAD-RELACIÓN EN LA ESTRUCTURA DE DATOS RELACIONALES DEL SIIMCELE.	107
5.2 NORMALIZACIÓN EN EL SIIMCELE.	112
6. LENGUAJE DE PROGRAMACIÓN EMPLEADO PARA DESARROLLAR EL SIIMCELE.	119
6.1 VISUAL FOXPRO Y OTROS LENGUAJES DE PROGRAMACIÓN.	122
6.2 CAPACIDADES DE VISUAL FOXPRO.	129
7. MANUAL DE USUARIO DEL SIIMCELE.	132
7.1 INTRODUCCIÓN AL SISTEMA DE INFORMACIÓN DE LA MEDIATECA DEL CELE (SIIMCELE).	132
7.2 USO DEL SIIMCELE.	133
8. PRUEBAS Y CONFIABILIDAD DEL SIIMCELE.	155
8.1 PRUEBAS DE UNIDAD DEL SIIMCELE.	155
8.2 PRUEBAS DE INTEGRACIÓN DEL SIIMCELE.	160
8.3 PRUEBAS DE VOLUMEN DEL SIIMCELE.	160
8.4 PRUEBAS DE VALIDACIÓN DEL SIIMCELE.	161

9. CALIDAD Y MANTENIMIENTO DEL SIIMCELE.	162
9.1 CALIDAD DEL SIIMCELE.	162
9.2 MANTENIMIENTO DEL SIIMCELE.	165
10. CONCLUSIONES.	166
APÉNDICE.	170
CÓDIGOS DE BARRAS.	170
CÓDIGOS DE BARRAS EN EL SIIMCELE.	171
BIBLIOGRAFÍA.	173

INTRODUCCIÓN.

En los siguientes párrafos, que constituyen este capítulo introductorio, exponemos la estructura general de la tesis que presentamos titulada *Sistema Integral de Información para la Mediateca del Centro de Enseñanza de Lenguas Extranjeras (SIIMCELE)*.

En el Capítulo 1 incluimos los fundamentos teóricos sobre los cuales se apoya el desarrollo de este proyecto.

En el Capítulo 2 nos enfocaremos en la situación problemática actual de la **MEDIATECA** del Centro de Enseñanza de Lenguas Extranjeras (CELE) de la UNAM, recopilando los requisitos necesarios para el desarrollo de una solución que tome en cuenta las restricciones y recursos existentes en ese centro.

En el Capítulo 3 se describe la metodología seguida para desarrollar el *SIIMCELE*, la cual se apoya en las herramientas del *análisis y especificación estructurada*, el diagrama de flujo de datos, el diccionario de datos, las mini-especificaciones y las tablas de decisiones.

En el Capítulo 4 empleamos el diseño estructurado para elaborar la carta de estructura del *SIIMCELE* y de esta manera, analizar la función y las relaciones de cada módulo del sistema, apoyándonos al mismo tiempo de los conceptos de cohesión y acoplamiento.

Los conceptos correspondientes al uso de una base de datos descritos en la parte teórica de este proyecto son aplicados al desarrollo del *SIIMCELE*, de la manera en que se muestra en el Capítulo 5.

La herramienta de programación empleada en el desarrollo del *SIIMCELE* es descrita en el Capítulo 6.

En el Capítulo 7, se encuentra el manual del *SIIMCELE*, que especifica al usuario el manejo de la aplicación, ofreciendo los pasos básicos de ejecución para cada una de las funciones del sistema.

Las pruebas del *SIIMCELE* se llevaron a cabo con objeto de garantizar su calidad. Las pruebas realizadas se conocen como Prueba de Unidad, Prueba de Integración, Prueba de Volumen, Prueba de Validación. Todas ellas desarrolladas en el Capítulo 8. Se anexa la carta de aceptación del *SIIMCELE* expedida por la coordinación de la *MEDIATECA*.

Desarrollamos el Capítulo 9, denominado Calidad y Mantenimiento del sistema, cuya primer parte enfatiza a las Técnicas de Verificación Formal, que nos ayudan a valorar la concordancia con los requisitos funcionales y de rendimiento obtenido del software, a medida que se desarrolla. Se trata también al mantenimiento del sistema, el aspecto más extenso del Ciclo de Sistemas en la segunda parte de ese capítulo.

El Capítulo 10 es el espacio para las Conclusiones; y al final de este proyecto, existe un Apéndice sobre códigos de barras.

1. FUNDAMENTOS TEÓRICOS.

1.1 ESTUDIO INICIAL Y ANÁLISIS DEL SISTEMA.

Un *sistema*, según el diccionario "Webster" corresponde a:

1. Un conjunto u ordenación de cosas relacionadas de tal manera que forman una unidad o un todo orgánico; 2. Un conjunto de hechos, principios, reglas, etc., clasificados y ordenados de tal manera que muestran un plan lógico uniendo las diferentes partes; 3. Un método o plan de clasificación u ordenación; 4. Una forma establecida de hacer algo; un método, un procedimiento...

Considerando las definiciones anteriores, podemos indicar que un *sistema basado en computadora* es:

"Un conjunto de elementos organizados para llevar a cabo algún método, procedimiento o control mediante el procesamiento de la información"¹.

Los elementos de un *sistema basado en computadora*, son²:

- *Software*. Incluye a los programas de computadora, las estructuras de datos y la documentación asociada.
- *Hardware*. Son los dispositivos electrónicos propios de la computadora y los dispositivos electromecánicos para la comunicación con el exterior.
- *Gente*. Involucra a los usuarios, operadores del software y los operadores del hardware.
- *Bases de datos*. Conjunto ordenado de información la cual es accesible mediante el software.

¹Pressman, Roger S., Ingeniería del software, Editorial Mc Graw-Hill. México. 1993, pp. 140-141

²Ibidem, pp. 142

- *Documentación.* Información escrita que explica la operación y uso del sistema.
- *Procedimientos.* Los pasos que definen el uso específico de cada elemento del sistema.

Para el desarrollo de un sistema basado en computadora, existen varias metodologías disponibles en la actualidad:

- Ciclo de vida clásico.
- Construcción de prototipos.
- Modelo en espiral o evolutivo.
- Ciclo de vida clásico modificado.

El proceso de desarrollo de un sistema (independientemente de la metodología que se ha seleccionado) contiene tres fases genéricas:

- Descripción y análisis de la situación actual.
- Diseño (que incluye el uso de bases de datos, codificación y empleo de un lenguaje de programación y pruebas del sistema).
- Mantenimiento.

Ciclo de vida clásico.

El ciclo de vida clásico presenta un enfoque sistemático o secuencial del desarrollo de un sistema que comienza en el nivel de la ingeniería de sistemas y progresa a través del análisis, diseño, codificación, prueba y mantenimiento.

Ingeniería de sistemas. Dado que el software forma parte de un sistema mayor,

se debe comenzar por establecer los requisitos de todos los elementos del sistema y luego asignarlos al software.

Análisis. Se recopilan los requisitos necesarios para el desarrollo del sistema, además se propone su función, su rendimiento y las interfaces requeridas.

Diseño. Traduce los requisitos en una representación de software. Se documenta y forma parte de la configuración del software.

Codificación. El diseño se traducirá en forma legible para la máquina; si el diseño se realizó detalladamente, la codificación podrá realizarse mecánicamente.

Prueba. La prueba se centra en la lógica interna del programa, comprobando sentencias, funciones externas, entradas y resultados esperados.

Mantenimiento. El mantenimiento del software aplica cada uno de los pasos precedentes del ciclo de vida a un programa existente en vez de a uno nuevo.

Construcción de prototipos.

La *construcción de prototipos* es un modelo que facilita la creación del sistema a construir mediante una de las tres formas siguientes:

- Un prototipo en papel o un modelo basado en computadora que describa la interacción hombre – máquina.
- Un prototipo que implemente algunos subconjuntos de la función requerida del programa deseado.
- Un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de

desarrollo.

Como en el ciclo de vida clásico, se comienza con la recolección de los requisitos, luego se produce un "diseño rápido", el cual se enfoca sobre la representación de los aspectos del software visibles al usuario, de esta manera, el "diseño rápido" conduce a la construcción del prototipo que será evaluado por el cliente/usuario, y que será utilizado para refinar los requisitos del software a desarrollar. Se produce un proceso interactivo en el que el prototipo se va afinando para satisfacer las necesidades del cliente, al mismo tiempo que va ayudando al que lo desarrolla.

Modelo en espiral.

El *modelo en espiral* se ha desarrollado para cubrir las mejores características tanto del ciclo de vida clásico, como de la creación de prototipos añadiendo un nuevo elemento: el análisis de riesgo.

Esta metodología define cuatro actividades:

- *Planificación.* Determinación de objetivos, alternativas y restricciones.
- *Análisis de riesgo.* Análisis de alternativas e identificación y/o resolución de riesgos.
- *Ingeniería.* Desarrollo del producto de "siguiente nivel".
- *Evaluación del cliente.* Valoración de los resultados de la ingeniería.

En cada iteración alrededor de una espiral se construyen sucesivas versiones del software, cada vez más completas. Durante la primera vuelta se definen los objetivos, las alternativas y las restricciones, y se analizan e identifican los riesgos. El cliente evalúa y basándose en los comentarios se produce la fase de planificación y de análisis de riesgo. Si el análisis de riesgo indica que hay una incertidumbre en los requisitos, se puede usar la creación de prototipos para refinar los requisitos. En cada

bucle alrededor de la espiral, la culminación del análisis de riesgo resulta en una decisión de "seguir o no seguir". Cada vuelta alrededor de la espiral requiere ingeniería que se puede llevar a cabo mediante el enfoque del ciclo de vida clásico o de la creación de prototipos.

Ciclo de vida clásico modificado.

Tomando en cuenta los elementos del ciclo de vida clásico (previamente expuestos) y con la intención de hacerlo aplicable a la mayoría de las situaciones, surge el *ciclo de vida clásico modificado* que proporciona más flexibilidad entre las fases, permitiendo iteraciones entre ellas, según sean necesarias; trata de acomodar cambios en los requerimientos y disminuir los efectos de errores en cada etapa del desarrollo.

¿Cuál metodología elegir?

Ciclo de vida clásico.

El ciclo de vida clásico proporciona un modelo que sirve para dos propósitos; el primero, permite representar los procesos de concepción y producción en una forma gráfica y lógica; y el segundo, proporciona un marco de trabajo alrededor del cual las actividades de aseguramiento de calidad pueden ser construidas en una manera disciplinada.

Construcción de prototipos.

La construcción de prototipos como paradigma, puede ser problemática porque el cliente ve funcionando lo que parece ser la primera versión del sistema e ignora que no se han considerado todos los aspectos de calidad, por lo que

solicita desesperadamente se apliquen todas las mejoras necesarias. También puede suceder que, para crear el prototipo se utilice un sistema operativo o un lenguaje de programación inapropiados, simplemente porque ya está disponible y es conocido.

A pesar de esto, el modelo funciona adecuadamente si se definen las reglas del juego desde el principio; el cliente y el desarrollador deben estar de acuerdo en que el prototipo sirve como un mecanismo de definición de requisitos y posteriormente será descartado para construir el sistema real.

Modelo en espiral.

El modelo en espiral utiliza tanto el enfoque basado en prototipos como los pasos sugeridos por el ciclo de vida de clásico pero incorporándolo dentro de un marco de trabajo interactivo.

Pero, por otro lado, puede resultar difícil convencer a los usuarios de que el modelo en espiral es controlable; se requiere una considerable habilidad para la valoración del riesgo y, por último, este modelo es el más reciente y no se ha usado tanto como el ciclo de vida o la creación de prototipos.

En la mayoría de los casos, los paradigmas pueden y deben combinarse de forma que sea posible utilizar las ventajas de cada uno en un mismo proyecto. Para el desarrollo de este proyecto, nos apoyaremos fundamentalmente en el **ciclo de vida clásico** (Fig. 1.1), porque es el paradigma más ampliamente usado, tiene un lugar definido e importante dentro del trabajo que se emplea en el desarrollo de sistemas, suministra una plantilla en la que pueden colocarse los métodos para el análisis, diseño, codificación, prueba y mantenimiento. Además, los pasos de este modelo son muy similares a los pasos genéricos aplicables a todos los paradigmas de la ingeniería de sistemas.

Dentro del análisis y el diseño, emplearemos el **enfoque estructurado** (explicado más adelante) que nos permitirá modelar los componentes del sistema por medio de símbolos gráficos, estableciendo una atención especial en los aspectos lógicos del sistema. Este enfoque formula las especificaciones funcionales para los módulos de software. Asimismo, incluye una descripción de la interacción entre los diferentes módulos, pero sin mostrar la codificación interna de cada uno de ellos.

Todas las definiciones de datos, procesos y demás información pertinente, se encuentran descritas en el diccionario de datos, que es un elemento central en el enfoque estructurado.

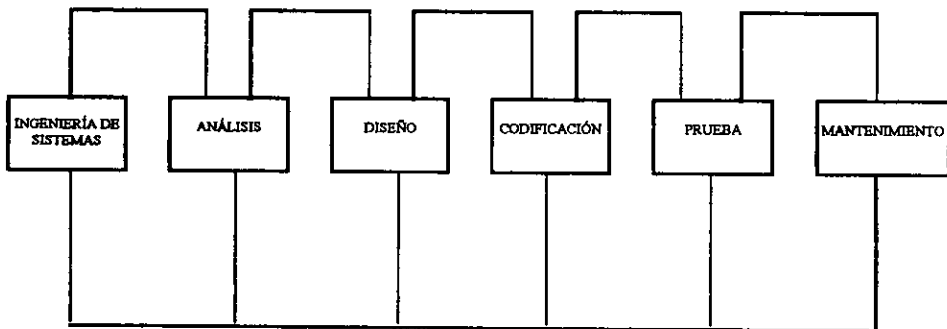


FIG. 1.1

1.1.1 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL.

La *descripción de la situación actual* del problema en estudio es indispensable para identificar las necesidades a considerar: el analista se entrevista con el cliente para definir los objetivos del sistema, la información que se desea obtener, la información que se va a suministrar, las funciones y el rendimiento requeridos. Es útil establecer la diferencia entre lo que "necesita" el cliente y lo que "quiere" el cliente.

1.1.2 ANÁLISIS DE LA PROBLEMÁTICA DE LA SITUACIÓN ACTUAL Y PROPUESTA DE SOLUCIÓN.

Analizar la problemática de la situación actual implica descubrir las funciones que debe realizar el sistema y asignarlas a elementos individuales del sistema que se propondrá; es decir, asignarlas al software, al hardware, a la gente; en una palabra, a los recursos. Es necesario considerar los objetivos y las restricciones definidas por el usuario para desarrollar una representación del alcance del sistema describiendo la función, rendimiento, restricciones, interfaces y estructura de datos.

La *función* conjunta a las acciones ejecutadas por el sistema; el *rendimiento* es el tiempo de respuesta y de procesamiento; y las *restricciones* son los límites del software debido al hardware externo, a la memoria disponible y a otros sistemas externos. Estos tres factores se evalúan juntos.

El desarrollo de una *interfaz* debe considerar al hardware (procesador, periféricos, máquinas, etc.), que es controlado por el software (rutinas de acceso a una base de datos, paquetes de subrutinas, sistemas operativos, etc.) que a su vez está enlazado a la gente responsable de su manejo.

La *estructura de datos* es una representación de la relación lógica existente entre los datos y determina la organización, los métodos de acceso, el grado de asociatividad y las alternativas de procesamiento para dichos datos.

1.1.3 ANÁLISIS DE FACTIBILIDAD TÉCNICA DE LA PROPUESTA DE SOLUCIÓN.

La *factibilidad técnica* se refiere a establecer las especificaciones exactas de los recursos necesarios para el funcionamiento del sistema, al mismo tiempo que se efectúa el análisis y definición del mismo.

Recursos.

La especificación de los recursos, incluye establecer las herramientas existentes (hardware y software).

Hay que considerar tres categorías de hardware:

1. *Sistema de desarrollo o sistema anfitrión.* Compuesto por la computadora que se utilizará en la fase de desarrollo del software junto con sus periféricos.
2. *Máquina objetivo.* Computadora en la cual se ejecutará el software desarrollado.
3. *Elementos de hardware del sistema que se desarrollará.* Dispositivos que se deberán incluir con el nuevo sistema a desarrollar.

Respecto al software, se definirán los programas y lenguajes de programación que servirán como herramienta para desarrollar al sistema, tales como: la programación orientada a objetos, los lenguajes de programación de cuarta generación y los sistemas avanzados de consulta a bases de datos, entre otros.

1.2 ANÁLISIS Y ESPECIFICACIÓN ESTRUCTURADA.

El *análisis y especificación estructurada* es una metodología de análisis empleada para el desarrollo de sistemas que examina los problemas mediante el modelo entrada-proceso-salida, es decir, mediante el establecimiento del flujo de la información. Dicha metodología crea una representación del problema real con el objeto de establecer una solución apoyándose en herramientas como: el diagrama de flujo de datos, el diccionario de datos, las miniespecificaciones y las tablas de decisiones; herramientas que serán explicadas a lo largo de este capítulo.

1.2.1 EL DIAGRAMA DE FLUJO DE DATOS (DFD).

El enfoque de flujo de datos permite crear una descripción gráfica del problema, esta descripción se llama *diagrama de flujo de datos (DFD)* y establece los caminos que seguirán los datos a través de los procesos del sistema a desarrollar.

Al utilizar el DFD es posible beneficiarse de tres maneras:

1. Contar con rapidez de una implantación técnica del sistema.
2. Comprender la relación existente entre los procesos del sistema.
3. La comunicación a los usuarios del estado actual del sistema, mediante los DFD's.

Para representar el flujo en un DFD emplearemos los cuatro símbolos que se muestran en la Fig. 1.2.

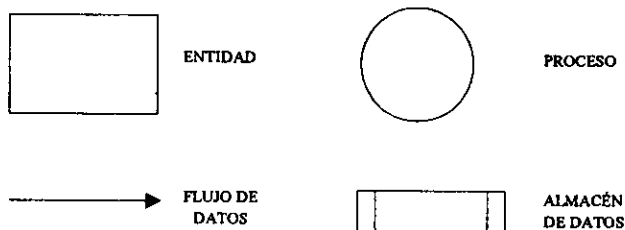


FIG. 1.2

La *entidad* representa algo que da y recibe datos del sistema; cada entidad se identifica por medio de un nombre apropiado.

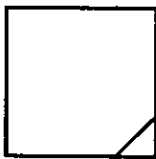
El *flujo de datos* representa el movimiento de los datos, con la punta se señala el destino de los datos, un flujo de información bidireccional puede representarse por medio de dos flechas paralelas. Cada flecha también se define con un nombre apropiado.

El *proceso* indica la existencia de una acción de transformación. Los procesos siempre denotan una transformación de los datos, por lo tanto, el flujo de información que sale de un proceso, siempre tendrá un nombre diferente del que tenía al entrar.

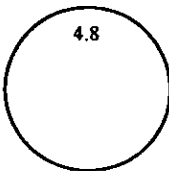
El *almacén de datos* representa el almacenamiento de la información. Se permiten anotaciones entre ambas líneas verticales del símbolo correspondiente.

Es conveniente señalar que los DFD's pueden particionarse para mostrar mayor detalle.

Algunas convenciones de uso común para evitar confusiones en los diagramas de flujo de datos son las mostradas en la Fig. 1.3.



Anotar una diagonal en la esquina para indicar que es una identidad duplicada.



Es conveniente utilizar decimales para indicar el nivel de detalle de los procesos relacionados.

FIG. 1.3

Características del DFD.

Para la elaboración eficaz de un DFD se debe atender a las siguientes características sistemáticas:

1. Desarrollar el DFD mediante el enfoque descendente (top-down).
 - Hacer una lista de las entidades, los flujos de datos, los procesos y los almacenes de datos; esto determina los límites del sistema que se desarrolla.
 - Dibujar un DFD básico que ilustre exclusivamente los aspectos generales.
2. Cubrir los detalles.
 - Añadir, por pasos, los detalles a cada proceso.
 - Indicar excepciones, cuando sea necesario.
3. Dibujar de nuevo los diagramas y volver a definir todos los símbolos por medio de nombres significativos.

Enfoque descendente.

Se puede iniciar un DFD a partir de la narración del sistema organizacional, haciendo uso de cuatro categorías: entidad, flujo de datos, proceso y almacén de datos. Cuando se haya compilado la lista de los datos elementales se puede elaborar un diagrama inicial.

El primer diagrama (nivel cero), incluirá lo básico de las entradas, los procesos y las salidas. Este será el diagrama más general.

Cubrir los detalles.

Posteriormente, los DFD's deberán incluir los detalles de cada uno de los procesos. Pero en el primer diagrama se deben definir las entradas y las salidas, éstas se

mantienen constantes en los diagramas consecutivos. El manejo de las excepciones se ignora en los primeros dos o tres niveles del dibujo del DFD.

El resto del diagrama original se descompone en diagramas que detallan los procesos, agregando en cada nivel inferior nuevos almacenes de datos y nuevos flujos de datos.

Dibujar de nuevo los diagramas.

Después que los diagramas de flujo se aclaran, lo que sigue es volver a dibujarlos y rotularlos de manera significativa. Es decir, los rótulos no deberán ser muy genéricos. También se debe pensar en que las personas poco familiarizadas con el sistema sean capaces de comprender su contenido.

La efectividad de los DFD's se debe a su consistencia: cada vez que sea posible, hay que unificar términos.

Los diagramas de flujo se emplean para el análisis y diseño de procesos. Utilizando DFD desde el principio se pueden establecer los requisitos de la información y de esta manera, obtener una visión global del flujo de los datos del sistema. En este punto, bastará con dibujar los DFD's a mano y de manera somera.

Al detallar los DFD's, éstos pueden ser útiles para la interacción con los usuarios, puesto que ellos pudieran opinar sobre la realización del proceso y así obtener una mayor precisión en los diagramas.

Al final, los DFD's sirven para la documentación del sistema ya que ayudan a concebir la lógica de los flujos de datos de la organización. Para evitar el dibujo manual de los DFD's se pueden emplear herramientas integradas (workbench) o instrumentos CASE (Computer Assisted Software Engineering).

1.2.2 DICCIONARIO DE DATOS.

El *diccionario de datos* es una referencia que sirve de guía durante el análisis y el diseño del sistema. Es un documento en el que se recopila, se coordina y se confirma lo que un término específico significa para el sistema.

Al elaborar un diccionario de datos, es necesario catalogar los diversos términos que se refieren al mismo dato con objeto de evitar duplicaciones.

Datos del Diccionario de Datos.

Como ya se mencionó, el diccionario de datos es una referencia útil para localizar nombres y atributos de los datos utilizados en el sistema y deberá actualizarse cada vez que se hagan cambios en el sistema. Los registros del diccionario de datos deben contener información referente a:

1. El nombre y el alias del dato.
2. Las descripciones del dato.
3. Los flujos de datos que se relacionan con el término.
4. La longitud disponible en caracteres.
5. Cualquier otra información pertinente.

1. *Nombres y Alias.* Es necesario incluir la manera más común de denominar el dato en los programas del sistema, además de su alias. El registro de estos datos, facilita la comunicación y la referencia cruzada entre esos programas.

2. *Descripción.* La descripción incluirá un texto breve, pero informativo del dato elemental.

3. *Flujos de datos.* Establecer el origen y el destino de la transferencia de datos.

4. *Longitud del dato.* En el diccionario de datos, también se establece la longitud permitida para el acceso de un dato. La longitud siempre se da en función del número

de caracteres impresos.

5. *Información adicional.* Deberá agregarse al diccionario de datos cualquier información útil extra.

La notación utilizada en el diccionario de datos es:

=	Está compuesto de.
+	Y.
()	Una de las alternativas.
[]	Opcional.
*(1 - 5)	Iteración (5 veces).
*(1 -)	Iteración (se ejecuta hasta concluir).

¿Cómo formar un diccionario de datos?

Se pueden observar los siguientes pasos en la elaboración de un diccionario de datos:

1. **Flujos de datos y almacenes de datos.** En este paso se catalogan los flujos que entren y salgan de los procesos ya señalados por los DFD's. Además, se catalogan los almacenes de datos empleados en cada proceso.
2. **Describir la estructura de datos.** Aquí, se describen los grupos relacionados de datos elementales que existan dentro del sistema. La estructura de datos es alimentada por los flujos y los almacenes de datos.
3. **Desglosar la estructura de los datos en datos elementales.** En esta parte, hay que trabajar con los significados más simples del sistema, es decir, con los datos elementales.

1.2.3 MINI-ESPECIFICACIONES.

Las *mini-especificaciones* son sentencias que existirán por cada proceso especificado en los DFD's. Se componen de verbos imperativos y de palabras reservadas para expresar la lógica del funcionamiento de cada proceso.

La sintaxis empleada se limita a oraciones declarativas, estructuras de decisión y estructuras de iteración:

- PROCEDIMIENTO (PROCEDURE).
- SI-ENTONCES-DE OTRA MANERA (IF-THEN-ELSE).
- HACER...MIENTRAS (DO...WHILE).
- HACER...HASTA (DO...UNTIL).

Una herramienta que puede emplearse para hacer las mini-especificaciones es la constituida por las tablas de decisiones.

1.2.4 TABLAS DE DECISIÓN.

Las *tablas de decisión* se emplean para traducir acciones y condiciones a una forma tabular. La organización de una tabla de decisión se muestra en la Fig. 1.4.

La tabla está dividida en cuatro cuadrantes; el superior izquierdo, contiene la lista de todas las condiciones; el inferior izquierdo, la lista de todas las acciones que se pueden producir basándose en combinaciones de las condiciones; en los cuadrantes de la derecha se indican las combinaciones de las condiciones y las correspondientes acciones que se han de producir para cada combinación específica. De esta manera, cada columna se interpreta como una regla de procesamiento.

Números de reglas		1	2	3	4	5	6	7	8	9	10
Filas de condiciones											
Filas de acción											

R E G L A S

FIG. 1.4

Para desarrollar una tabla de decisión se aplican los siguientes pasos:

1. Listar las acciones que puedan asociarse con un proceso.
2. Listar las condiciones o decisiones durante la ejecución del procedimiento.
3. Asociar conjuntos específicos de condiciones con acciones específicas, eliminando las combinaciones de condiciones imposibles.
4. Definir reglas indicando qué acciones ocurren para un conjunto de condiciones.

1.3 DISEÑO ESTRUCTURADO.

El *diseño estructurado* se apoya en las herramientas del análisis y especificación estructurada y en la carta de estructura para interconectar los datos con los procesos que en conjunto constituyen el sistema que soluciona la problemática de la situación en consideración. El diseño estructurado, además, tiene en cuenta conceptos clave como son: la modularidad, la cohesión y el acoplamiento; explicados más adelante.

1.3.1 LA CARTA DE ESTRUCTURA (CDE).

La *carta de estructura (CDE)* es un medio gráfico para mostrar la estructura de los componentes de un sistema, señalando la manera en que los elementos de un DFD pueden relacionarse. La CDE puede usarse como una descripción visual del programa junto con la información de control.

Características de la CDE.

La CDE cuenta con las siguientes características de representación: un componente funcional (proceso) se representa como un rectángulo, la jerarquía se despliega mediante rectángulos enlazados con líneas, las entradas y salidas a los componentes se indican mediante flechas.

Dado un DFD mostrando transformaciones lógicas, generalmente es posible derivar una CDE de él. Con el objeto de derivar la "mejor" CDE de un DFD es necesario atender a los siguientes conceptos:

1. *Entrada.* Esta unidad del programa es responsable de aceptar datos de una unidad de un menor nivel en la CDE y pasar esos datos de manera modificada a una de más alto nivel (Constantine y Yourdon emplean el término "aférente" para describir estas unidades).
2. *Salida.* Esta unidad del programa es responsable de aceptar datos de una unidad de más alto nivel y pasarlos a una unidad de más bajo nivel ("eferente" según Constantine y Yourdon).
3. *Transformar.* Una unidad del programa acepta datos de una unidad de un nivel más alto, transforma esos datos y los regresa a la unidad.
4. *Coordinar.* Una unidad es responsable de controlar y manejar otras unidades.

El primer paso para convertir un DFD a una CDE es identificar las unidades de salida y de entrada del más alto nivel. Esas unidades tienen que ver con el paso de datos hacia

arriba y hacia abajo en la jerarquía.

El primer nivel de la CDE se produce mediante la representación de los procesos como unas cajas sencillas. La caja en la raíz de la CDE es una unidad de coordinación. El proceso de elaboración puede ser repetido para las unidades del primer nivel en la CDE hasta que todos los procesos en el DFD sean representados en la estructura.

Cada nodo en la CDE de un diseño bien estructurado debe tener entre dos y siete subordinaciones. Si el nodo tiene sólo una subordinación, esto implica que la unidad representada por ese nodo puede tener un bajo grado de cohesión. Si el nodo tiene muchas subordinaciones, esto implica que el diseño ha sido desarrollado a demasiado bajo nivel en esa parte.

La derivación de la CDE es además un proceso de dos etapas. Desde el DFD, una estructura inicial puede derivarse y puede ser usada en la construcción de la descripción del diseño que incluye información de control y en la cual, adicionalmente, se pueden identificar las funciones de control. Siguiendo esta descripción del diseño, la CDE se modifica para incluir componentes de control adicionales.

Modularidad.

El concepto de *modularidad* implica que el software se divide en componentes con nombres y ubicaciones determinados, denominados *módulos* y que se integran para satisfacer los requisitos del problema. Es decir, los módulos son los componentes procedimentales (procesos) del sistema.

La *arquitectura del software* se refiere a la estructura jerárquica de los módulos y a la estructura de datos y se obtiene mediante un proceso de partición, que relaciona los elementos de una solución de software con partes de un problema del mundo real

definido durante el análisis de requisitos.

El software puede ser diseñado mediante dos enfoques:

1. *Diseño orientado al flujo de datos (diseño estructurado)*. Esta estructura considera la modularidad, el diseño descendente y la programación estructurada, integrando explícitamente el flujo de la información.
2. *Diseño orientado a los objetos*. Con este diseño se interconectan objetos de datos (elementos de datos) y operaciones de procesamiento en una forma que modulariza la información y el procesamiento, en lugar de dejar aparte el procesamiento. Esta estructura se basa en los conceptos de abstracción, ocultamiento de información y modularidad.

1.3.2 COHESIÓN.

La *cohesión* se refiere a la fuerza de unión entre las funciones que realiza un módulo y se mide considerando una escala que va desde la más débil hasta la más fuerte (la más deseada) en el siguiente orden³:

1. *Cohesión coincidental.* Ocurre cuando los elementos dentro de un módulo no tienen relación aparente entre cada uno de ellos; esto puede suceder cuando se segmenta arbitrariamente un programa en varios módulos.
2. *Cohesión lógica.* Este tipo de cohesión implica algunas relaciones entre los elementos de un módulo, como por ejemplo, en uno que desempeñe todas las funciones de entrada y salida, o en otro dedicado a la edición general de los datos. Un módulo unido lógicamente por lo común combina varias funciones relacionadas en una forma compleja e interrelacional.
3. *Cohesión temporal.* En los módulos que presentan cohesión temporal, todos los elementos son ejecutados en un momento dado sin requerir de ningún parámetro o lógica alguna para determinar qué elemento debe ejecutarse. Un ejemplo de esta cohesión se observa en un módulo dedicado a la inicialización de un sistema.
4. *Cohesión en la comunicación.* Los elementos de un módulo que contiene cohesión en la comunicación se refieren al mismo conjunto de datos de entrada y de salida; por ejemplo la instrucción "Imprima y perfore el archivo de salida" presenta una cohesión en la comunicación.
5. *Cohesión secuencial.* Esta clase de cohesión se presenta cuando la salida de un elemento es la entrada para el siguiente; por ejemplo, "lea la siguiente transacción y actualice el archivo maestro" se encuentra unida secuencialmente.
6. *Cohesión funcional.* Este tipo de cohesión representa un tipo fuerte, y por lo tanto deseable, de la relación entre los elementos de un módulo, debido a que todos ellos se encuentran encausados al desempeño de una sola función. Un

ejemplo de un módulo de este tipo, es el que ejecuta la instrucción "escriba el registro en el archivo de salida".

7. *Cohesión informacional*. Ocurre cuando un módulo contiene una estructura de datos compleja, así como varias rutinas que manejan dicha estructura; cada rutina del módulo presenta unión funcional; esta cohesión es la realización total de la abstracción de los datos. La cohesión informacional solamente requiere que el segmento con cohesión funcional sea ejecutado al ser llamado el módulo.

No es indispensable tener siempre el nivel más alto de cohesión, sin embargo, lo importante es intentar conseguir una cohesión alta y saber reconocer la cohesión baja, de forma que sea posible modificar el diseño del software para que disponga de una mayor independencia funcional. La cohesión de un módulo puede ser determinada atendiendo a los razonamientos propuestos por Constantine (desarrollador del análisis y diseño estructurado):

Escribir una proposición sobre el propósito del módulo y examinarla; si la proposición es compuesta y contiene más de una coma o más de un verbo, el módulo probablemente estará desempeñando más de una función, por lo que tal vez tenga una cohesión del tipo secuencial o de comunicación. Si el enunciado contiene palabras relacionadas con el tiempo, tales como "primero", "después", "entonces", etc., el módulo probablemente tiene una unión secuencial o temporal. Si el predicado de la oración no contiene un objeto específico y sencillo que le siga al verbo, el módulo probablemente se encuentra unido lógicamente; por ejemplo, "edita todos los datos" está unido lógicamente, mientras que "edita los datos fuente" puede tener una cohesión funcional. La existencia de palabras como "inicializa" y "limpia" denotan una unión temporal.

³ Fairley, Richard, Ingeniería de Software, Editorial Mc Graw-Hill, México 1994, pp. 156-160

1.3.3 ACOPLAMIENTO.

El *acoplamiento* es una medida de la interconexión entre los módulos de un sistema. El acoplamiento depende de la complejidad de las interfaces entre los módulos, del punto en el que se hace una entrada o referencia a un módulo y de los datos que pasan a través de esas interfaces.

En el diseño del software se busca el más bajo acoplamiento posible. La conectividad sencilla entre módulos da como resultado un software que es más fácil de comprender y menos propenso a errores que aparecen en una posición y que se propagan a lo largo del sistema. El acoplamiento entre módulos puede ser considerado dentro de una escala del más fuerte al más débil (el más deseable) de la siguiente manera:

1. *Acoplamiento del contenido.* Ocurre cuando un módulo modifica los valores locales o las instrucciones de algún otro módulo; este acoplamiento es posible en los programas en lenguaje ensamblador.
2. *Acoplamiento de zonas compartidas.* En este acoplamiento, los módulos son conjuntados por medio de zonas globales para las estructuras de datos, por ejemplo, cuando un programa se referencia a un bloque común de datos.
3. *Acoplamiento de control.* Este acoplamiento incluye el paso de banderas de control, ya sea como parámetros o en forma global entre los módulos, de tal forma que un módulo controla la secuencia de proceso de otro.
4. *Acoplamiento por zonas de datos.* Este tipo de acoplamiento es similar al de zonas compartidas, excepto que los elementos globales son compartidos en forma selectiva entre los diversos módulos que requieren los datos
5. *Acoplamiento de datos.* Incluye el uso de listas de parámetros para pasar los elementos entre los módulos.

1.4 BASES DE DATOS.

Es posible considerar dos tipos de almacenamiento de datos en un sistema de información basado en computadora: archivos individuales o elaborar una base de datos. Un sistema que hace uso de los archivos convencionales implica que los datos almacenados lleguen a ser redundantes, la actualización de los archivos lleva más tiempo y existirán problemas de integración de datos, ya que los cambios en un archivo, requerirán la modificación de ciertos datos en otros archivos.

Por otro lado, una *base de datos* es una fuente central de datos significativos que pueden ser compartidos por varios usuarios y por varias aplicaciones (si ese fuera el caso) y asegura que los datos requeridos estén siempre disponibles, además permite el mantenimiento preciso y consistente de los mismos.

Es oportuno en este momento, establecer los conceptos básicos necesarios en el manejo de una base de datos.

Entidades. Una *entidad* es cualquier objeto o evento, que sea fuente de datos.

Relaciones. Las *relaciones* son asociaciones de datos entre entidades. Existen tres tipos de relaciones: *relación de uno a uno* (1:1), para una entidad existe otra sola entidad; *relación de uno a muchos* (1:M), para una entidad existen muchas otras entidades, o para muchas entidades existe otra sola entidad; finalmente, la *relación de muchos a muchos* (M:N), para muchas entidades existen otras varias entidades.

Atributos. Un *atributo* es una característica de una entidad, y pueden describirse varios atributos para una entidad.

Registros. Un *registro* es una colección de datos elementales que tienen algo en común con la entidad correspondiente. Los datos son las unidades más pequeñas en una

base de datos. La palabra dato, puede aquí considerarse como sinónimo de atributo. los tipos de los datos pueden ser alfabéticos, numéricos o alfanuméricos (todos ellos de longitud variable). Ocasionalmente, un dato puede ser llamado como un campo, sin embargo, un campo representa algo físico y no lógico, por lo tanto esta nominación es incorrecta; además, un grupo de datos puede agruparse en un campo, por ejemplo, el campo fecha contiene los datos día, mes y año.

Llaves. Una llave es un dato en un registro que se emplea como identificador para éste. Pueden mencionarse dos tipos de llaves: 1) *llave primaria*, cuando una llave identifica de manera exclusiva a un registro; 2) *llave externa*, un atributo que no es llave primaria en una tabla pero si lo es en otra.

Un registro puede esquematizarse como se indica en la Fig. 1.5:

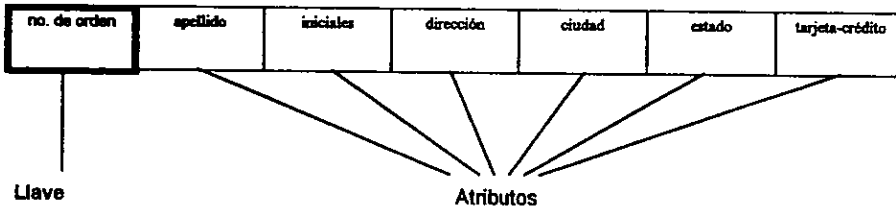


FIG. 1.5

Tipos de consulta.

El almacenamiento empleando bases de datos facilita la emisión de reportes adecuados a los usuarios mediante el empleo de consultas, las cuales son preguntas acerca de la información contenida en dicha base. Pueden mencionarse algunos tipos de consultas básicos⁴, involucrando a las entidades, atributos y valores, en cada uno de los casos, se conocen dos de esos elementos y se intenta obtener el tercero.

⁴Kendall, K. E., Análisis y diseño de sistemas, Editorial Prentice-Hall Hispanoamericana. México. 1991, p.p.701-

- Tipo 1. Se cuenta con la entidad E y uno de sus atributos A, se desea buscar el valor correspondiente: $V = (E, A)$.
- Tipo 2: $E = (V, A)$.
- Tipo 3: $A = (V, E)$.
- Tipo 4: $V = (E, \text{ todos los } A)$.
- Tipo 5: Todas las $E = (V, \text{ todos los } A)$.
- Tipo 6: Todos los $A = (V, \text{ todas las } E)$.

Empleando operadores aritméticos, comparativos y booleanos, las consultas pueden ser más complejas y precisas respecto a sus tipos básicos.

1.4.1 DIAGRAMA ENTIDAD-RELACIÓN.

Un *Diagrama Entidad-Relación* (Fig.1.6) es un diagrama que muestra los diversos tipos de relaciones entre las entidades y los atributos asociados empleando rectángulos para representar las entidades y líneas para representar las relaciones.

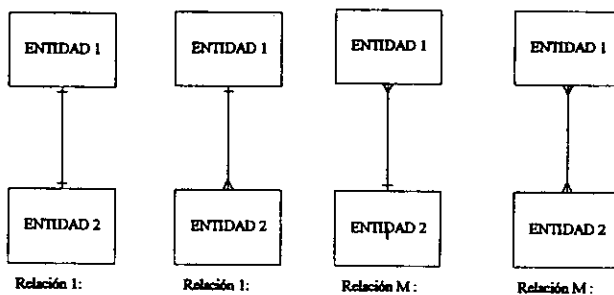


FIG. 1.6

Al diseñar una base de datos, es necesario atender a su estructuración lógica, la cual puede ser: jerárquica, en red y de relación.

Estructura de datos jerárquica.

La *estructura de datos jerárquica* (Fig.1.7) implica que las únicas relaciones permitidas son 1:1 y 1:M.

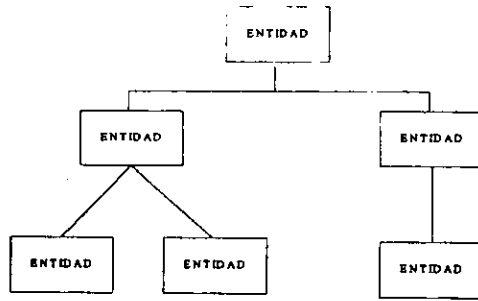


FIG. 1.7

Estructuras de red de datos.

Una *estructura de red de datos* (Fig. 1.8) permite que cualquier entidad cuente con cualquier número de entidades subordinadas o superiores. Las entidades se conectan mediante enlaces de red, que son datos comunes a ambas entidades conectadas.

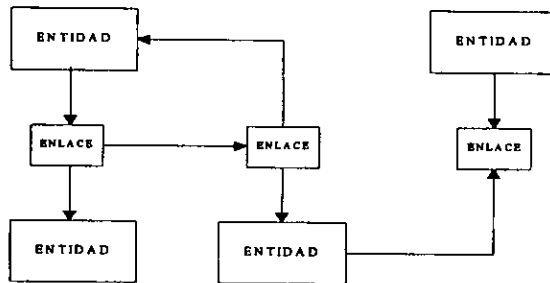


FIG. 1.8

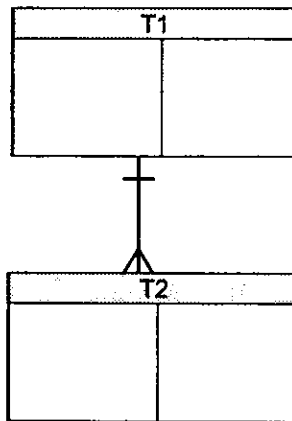
Estructura de datos relacional.

Una *estructura de datos relacional* consiste de tablas bidimensionales, en las cuales los renglones representan a los registros y las columnas a los atributos. Las relaciones entre estas tablas se esquematizan por medio de líneas, como las empleadas en el

Diagrama Entidad-Relación.

Este tipo de estructura presenta una ventaja importante respecto a los dos tipos anteriores: las consultas específicas se desarrollan eficientemente.

En la Fig. 1.9 se muestra la esquematización de este tipo de estructura. Se observa la relación existente entre dos tablas (una llamada T1 y la otra T2), en este caso, la relación es 1:M.

**FIG. 1.9**

1.4.2 NORMALIZACIÓN.

La normalización es la simplificación de las relaciones existentes entre los datos de las aplicaciones y los almacenamientos usados por los usuarios; de tal manera, que se establezcan estructuras de datos de menor tamaño. Se emplean tres pasos para lograr esta simplificación de los datos, llamados formas normales (FN).

La *primera forma normal (FN1)* elimina los grupos de datos repetidos e identifica la llave primaria, en este punto puede ser necesario desglosar alguna relación en dos o más relaciones.

La *segunda forma normal (FN2)* se aplica para asegurar que todos los atributos no llave sean dependientes de la llave primaria. Si existen dependencias se colocarán en otra relación.

La *tercera forma normal (FN3)* elimina cualquier dependencia transitoria (la dependencia en la cual sus atributos no llave son dependientes de otros atributos no llave).

1.5 LENGUAJES DE PROGRAMACIÓN.

La programación debe estar basada en una estructura lógica predecible, empleando las construcciones de secuencia, condición y repetición. La *secuencia* implementa los pasos de procesamiento esenciales de la especificación de cualquier algoritmo, la *condición* proporciona la posibilidad de seleccionar un procedimiento basado en alguna ocurrencia lógica y la *repetición* proporciona iteración.

Las herramientas de programación convencionales están integradas por compiladores,

editores, y depuradores. Pueden mencionarse además, los lenguajes de cuarta generación, los entornos de programación orientados a objetos (POO), los generadores de aplicaciones y los lenguajes de consulta a bases de datos.

Los lenguajes de programación, a través del tiempo han evolucionado y tradicionalmente se señalan hasta ahora, cuatro generaciones de desarrollo.

Primera generación de lenguajes.

Esta generación incluye la codificación a nivel de máquina y la hecha con lenguaje ensamblador. Representan el menor nivel de abstracción con el que se puede desarrollar un programa.

Existen tantos lenguajes ensambladores como arquitectura de procesadores con sus correspondientes conjuntos de instrucciones. Se recomienda usar estos lenguajes sólo cuando un lenguaje de alto nivel no satisfaga los requisitos o no esté disponible.

Segunda generación de lenguajes.

La segunda generación de lenguajes se caracteriza por su amplio uso, gran cantidad de bibliotecas de software, la familiaridad y la aceptación. Los ejemplos más representativos son: FORTRAN, COBOL, ALGOL y de alguna manera BASIC.

Tercera generación de lenguajes.

Los lenguajes de la tercera generación también se conocen como lenguajes de programación estructurada y se distinguen por sus posibilidades procedimentales y la

estructuración de datos. Los lenguajes de esta generación pueden ser clasificados como:

1. *Lenguajes de alto nivel de propósito general*: ALGOL, PL/1, PASCAL, Modula-2, C y Ada.
2. *Lenguajes orientados a objetos*: Dialectos de C como C++ y Objective-C, Smalltalk y Eiffel.
3. *Lenguajes especializados*: Estos lenguajes tienen una sintaxis no común diseñada especialmente para una aplicación en particular. Ejemplos: LISP, PROLOG, APL y FORTH.

Cuarta generación de lenguajes.

Los lenguajes de cuarta generación elevan los niveles de abstracción mediante una sintaxis diferente a la usada en las anteriores generaciones para la representación del control y de las estructuras de datos. Los lenguajes de cuarta generación (L4G) eliminan la necesidad de especificar los detalles algorítmicos; combinan características procedimentales y no procedimentales; ya que se pueden especificar condiciones con sus correspondientes acciones y, al mismo tiempo, se puede indicar el resultado deseado.

Martin desarrolla las siguientes categorías de los L4G⁵:

- *Lenguaje de petición*. Estos lenguajes permiten manipular la información de una base de datos previamente creada.
- *Generadores de programas*. Con estos lenguajes se pueden crear programas en un lenguaje de tercera generación usando menos sentencias, mediante un gran uso de abstracción de datos y de procedimientos.

⁵Martin, J., Fourth Generation Languages, Volume 1, Editorial Prentice-Hall, USA. 1985.

- *Otros L4G.* Lenguajes de soporte a la toma de decisiones, que permiten llevar análisis de posibilidades en los modelos de hojas de cálculo bidimensionales hasta modelos estadísticos y de investigación operativa. Y los lenguajes de prototipos, que facilitan la creación de interfaces de usuario y diálogos.

Con las herramientas de cuarta generación, se pretende la generación automática de código, es decir, la representación de sistemas en el nivel de abstracción más elevado que el posible con los lenguajes de programación convencionales. Idealmente, estas herramientas traducirán la descripción de un sistema a un programa operativo y además, ayudarán a verificar la corrección de la especificación del sistema.

Las herramientas de cuarta generación incluyen a los generadores de código, a los lenguajes de cuarta generación y a los generadores de aplicaciones.

Características generales de los lenguajes de programación.

Las características de ingeniería en los lenguajes de programación se enfocan en las necesidades de un proyecto específico. sin embargo, se pueden establecer los siguientes planteamientos generales:

1. *Facilidad de traducción del diseño al código.* Proporciona una indicación de cómo se aproxima un lenguaje de programación a la representación del diseño.
2. *Eficiencia del compilador.* Existen avances en este campo respecto a la velocidad del procesador y de la capacidad de memoria; sin embargo, algunos compiladores de lenguajes de alto nivel son incapaces de producir código ejecutable rápido y ajustado, para estos casos, los lenguajes con compiladores optimizados pudieran ser empleados.
3. *Portabilidad del código fuente.* Esta característica puede entenderse de las siguientes maneras:
 - El código fuente puede ser transportado de un procesador a otro y de un compilador a otro, sin ninguna o muy pocas modificaciones.
 - El código fuente permanece inalterado cuando cambia su entorno de funcionamiento (por ejemplo, si se instala una versión nueva del sistema operativo).
 - El código fuente puede ser integrado en diferentes paquetes de software sin que prácticamente se requieran modificaciones debidas a las características propias del lenguaje de programación.
4. *Disponibilidad de las herramientas de desarrollo.* Característica importante, ya que puede reducir el tiempo requerido para la generación del código fuente y

además, puede mejorar su calidad. Estas herramientas incluyen: compiladores con depuradores, ayudas, facilidades de edición, herramientas para el control del código fuente, extensas bibliotecas de subprogramas para una gran variedad de áreas de aplicación, correctores, etc.

5. *Facilidad del mantenimiento.* Para poder llevar a cabo el mantenimiento del software, es necesario considerar los siguientes aspectos: a) entender el software, ya que el código final debe ser leído y modificado de acuerdo con los cambios del diseño; b) facilidad de traducción y c) las características de documentación (longitud de identificadores no limitada, etiquetado, definición de estructuras y tipos de datos, etc.)

Las características técnicas de los lenguajes de programación se presentan en cuatro grupos:

1. Tipificación de datos.
2. Subprograma.
3. Estructuras de control.
4. Soporte para el enfoque orientado a objetos.

1. Tipificación de datos.

Es posible definir a los tipos de datos y a la tipificación como una clase de objetos de datos junto con un conjunto de operaciones para crearlos y manipularlos. Un objeto de datos puede tomar un valor que se encuentre dentro del rango de valores permitidos del tipo de datos y puede ser manipulado por operaciones que se apliquen a ese tipo de datos.

Los tipos de datos simples pueden ser:

- Numéricos.
- Alfabéticos.
- Alfanuméricos.

Los tipos de datos más complejos engloban estructuras de datos: desde vectores simples unidimensionales hasta listas, arreglos complejos y heterogéneos y registros.

Las operaciones que se pueden realizar sobre un tipo de datos particular y la forma en que se pueden manipular distintos tipos en una misma sentencia se controlan por la comprobación de tipos. La comprobación de tipos está incluida en el intérprete o en el compilador.

2. *Subprogramas.*

Un subprograma es un elemento de un programa compilado por separado que contiene datos y estructuras de control (de forma genérica es un módulo). Dependiendo del lenguaje de programación, el subprograma puede ser llamado: subrutina, procedimiento, función, etc. Las características generales de un subprograma son:

- Una sección de especificación que incluye su nombre y la descripción de la interfaz.
- Una sección de implementación que incluye los datos y las estructuras de control.
- Un mecanismo de activación que permite que el subprograma sea requerido desde cualquier otro punto del programa.

En los lenguajes de programación orientados a objetos, el concepto clásico de subprograma es reemplazado por el de objeto.

3. Estructuras de control.

Todos los lenguajes de programación modernos permiten las estructuras de control: secuencias, condiciones y repeticiones. Además de las construcciones procedimentales básicas, pueden existir otras estructuras de control:

- La recursividad crea una segunda activación de un subprograma durante la primera activación.
- La concurrencia soporta la creación de múltiples tareas, la sincronización de tareas y la comunicación de tareas.
- El manejo de excepciones para "atrapar" los errores del sistema o los definidos por el usuario.

4. Soporte para el enfoque orientado a objetos.

Teóricamente, la creación de objetos es realizable en cualquier lenguaje de programación convencional. Prácticamente, el soporte para la programación orientada a objetos debe estar incorporado en el lenguaje de programación que se vaya a utilizar.

Un lenguaje de programación orientado a objetos soportará directamente la definición de clases, eventos, métodos, herencia y encapsulamiento. Muchos lenguajes también implementan características adicionales como polimorfismo (diferentes objetos que pueden recibir mensajes con el mismo nombre).

Un *objeto* es un elemento de una clase. Una *clase* es una descripción de los datos y características de comportamiento de un grupo de objetos. Las clases y objetos están estrechamente relacionados, pero no son lo mismo. Una clase contiene información sobre cuál debe ser la apariencia y el comportamiento de un objeto. Un objeto tiene ciertas propiedades o atributos; cada objeto reconoce y puede responder a determinadas acciones denominadas *eventos*. Un evento es una actividad específica y predeterminada, indicada por el usuario o por el sistema. También relacionados con los

objetos se encuentran los *métodos*. Los métodos son procedimientos asociados a un objeto. Los eventos pueden tener métodos asociados o pueden existir independientemente.

Todas las propiedades, eventos y métodos de un objeto se especifican en la clase. Las clases, además, tienen las siguientes características que las hacen especialmente útiles para crear código reutilizable y fácil de mantener:

- Encapsulamiento.
- Subclases.
- Herencia.

La ventaja de ignorar los detalles internos de un objeto para poder centrarse en los aspectos del objeto que se necesitan utilizar se denomina *abstracción*. El *encapsulamiento* que empaqueta el código de métodos y propiedades en un mismo objeto, contribuye a la abstracción.

Una *subclase* puede tener toda la funcionalidad de una clase existente, además de la funcionalidad y los controles adicionales que se desea darle. La creación de subclases es un modo de reducir la cantidad de código que hay que escribir.

Con la *herencia*, si se realiza un cambio en una clase, ese cambio se reflejará en todas las subclases que se basan en ella. Esta actualización automática ahorra tiempo y trabajo.

Los entornos de programación orientados a objetos están unidos a lenguajes de programación específicos: C++, Eiffel, Objective-C o Smalltalk. Estos entornos incorporan características de las interfaces de tercera generación (ratón, ventanas, menús desplegables, operaciones sensibles al contexto, multitarea) con funciones que permiten examinar todos los elementos contenidos en una biblioteca de objetos para determinar si pueden ser reutilizados en la aplicación actual.

1.6 MANUAL DEL USUARIO.

El *manual del usuario* es un documento que describe la forma de operación del sistema y que explica los pasos necesarios para llevar a cabo cualquier función proporcionada por el sistema. El manual debe ser lo suficientemente comprensible incluso para personas sin conocimientos de computación y que requieran manejar el sistema.

Este documento, servirá también como base para capacitar a los usuarios y de referencia cuando el usuario esté operando el sistema. El manual incluirá los siguientes puntos:

- Introducción al sistema.
- Cómo entrar al sistema.
- Descripción de los módulos.
- Descripción de cada uno de los menús y pantallas del sistema.
- Procedimientos para realizar todas las opciones disponibles en el sistema.
- Cómo salir del sistema.

Es conveniente incluir en el manual los menús y las pantallas tal y como aparecen en la computadora, acompañadas de un breve descripción; además se debe emplear un lenguaje común evitando en la medida de lo posible el uso de palabras técnicas.

1.7 PRUEBAS Y CONFIABILIDAD DEL SISTEMA.

La prueba del proyecto es un elemento importantísimo para garantizar la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. La prueba debe considerar los siguientes objetivos:

1. La prueba es un proceso de ejecución de un programa con la intención de descubrir errores.
2. La prueba es adecuada si se cuenta con una alta probabilidad de mostrar un error no descubierto antes.

1.7.1 PRUEBA DE UNIDAD.

La *prueba de unidad* centra el proceso de verificación en la menor unidad del diseño, es decir, en el módulo. Las pruebas que se realizan son de:

- *Interfaz*. Para asegurar que la información fluye de forma adecuada hacia y desde la unidad del programa que está siendo probada.
- *Estructuras de datos*. Para asegurar que los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución del algoritmo.
- *Condiciones límite*. Para asegurar que el módulo funciona correctamente en los límites establecidos como restricciones de procesamiento.
- *Caminos independientes*. Para asegurar que todas las sentencias del módulo se ejecutan por lo menos una vez.

También deben diseñarse pruebas para detectar errores en las estructuras de datos locales de cada módulo, en las siguientes categorías:

1. Tipificación impropia o inconsistente.
2. Inicialización o valores implícitos erróneos.
3. Nombres de variables incorrectos.
4. Tipos de datos inconsistentes.
5. Desbordamiento de direccionamiento.

De la misma manera, deben establecerse caminos para detectar errores debidos a cálculos o comparaciones incorrectos o flujos de control inadecuados:

1. Comparaciones entre tipos de datos distintos.
2. Operadores lógicos o de precedencia incorrectos.
3. Variables o comparadores incorrectos.
4. Terminación inapropiada o inexistente de bucles.

1.7.2 PRUEBA DE INTEGRACIÓN.

La *prueba de integración*, se establece para construir la estructura del programa mientras que, simultáneamente se realizan pruebas para detectar errores asociados con la interacción. Para “poner juntos” a los módulos se debe seguir la técnica denominada integración incremental, que consiste en probar pequeños segmentos para que los errores sean más fáciles de aislar y de corregir.

1.7.3 PRUEBA DE VOLUMEN.

La *prueba de volumen* intenta detectar errores en el software debidos a condiciones anormales, demandando recursos en cantidad, frecuencia o volúmenes anormales. Algunos ejemplos de pruebas para estos casos son:

1. Generar diez interrupciones por segundo, cuando las normales son una o dos.
2. Incrementar las frecuencias de datos de entrada en magnitud.
3. Casos en los que se requieran el máximo de memoria o de otros recursos.
4. Casos para probar la memoria virtual.
5. Casos de excesiva búsqueda de datos en disco.

1.7.4 PRUEBA DE VALIDACIÓN.

La validación del software se consigue demostrándose la conformidad con los requisitos, es decir, que el software funciona de acuerdo con las expectativas razonables del cliente.

Cuando se ha realizado el software para un cliente, es necesario realizar *la prueba de validación* para que el cliente establezca conformidad del sistema con todos los requisitos.

Para realizar esta prueba, es necesaria la instalación del producto terminado, posteriormente, se lleva a cabo la capacitación, tanto al personal técnico como a los propios usuarios. Se entiende por producto terminado, al sistema listo para ser empleado en el equipo para el cual fue diseñado, junto con el manual de usuario. El sistema deberá estar listo para comenzar su operación de manera inmediata.

1.8 CALIDAD Y MANTENIMIENTO DEL SISTEMA.

1.8.1 CALIDAD DEL SISTEMA.

El factor más importante de la calidad de un sistema es su utilidad, en otras palabras, que el sistema satisfaga las necesidades del usuario; la planeación cuidadosa, el análisis y la participación del cliente son obligatorios para el desarrollo de productos de programación útiles.

El concepto de calidad implica que: los requisitos del software son la base de las medidas de la calidad, hay ciertos seguimientos especificados que definen los criterios de desarrollo y existe un conjunto de requisitos implícitos que frecuentemente no se mencionan, como el deseo de un buen mantenimiento. Se debe resaltar el hecho de que la garantía de la calidad es un proceso de protección que se aplica a lo largo de todo el ciclo de vida del sistema y que esta garantía se basa en *las revisiones técnicas formales* cuyo objetivo es encontrar errores durante un proceso para evitar que se propaguen al paso siguiente.

Las revisiones técnicas formales garantizan la calidad del software mediante:

- Revisar el producto, no al productor.
- Fijar una agenda y mantenerla.
- Limitar el debate y las impugnaciones.
- Enunciar áreas de problemas, pero no intentar resolver cualquier problema que se ponga de manifiesto.
- Tomar notas escritas.
- Limitar el número de participantes e insistir en la preparación anticipada.
- Desarrollar una lista de comprobación para cada módulo que haya de ser revisado.
- Disponer recursos para las revisiones técnicas formales.

- Llevar a cabo un buen entrenamiento de todos los revisores.
- Repasar las revisiones anteriores.

Según McCall⁶, la calidad es una combinación de factores que se concentran en tres aspectos importantes de un producto de software; sus características operativas, su capacidad de cambios y su adaptabilidad a nuevos entornos.

- *Corrección*. Hasta dónde satisface un programa su especificación y los objetivos de la misión del cliente.
- *Fiabilidad*: Hasta dónde se puede esperar que un programa lleve a cabo su función pretendida con la exactitud requerida.
- *Eficiencia*: La cantidad de recursos informáticos y código necesarios para que un programa realice su función.
- *Integridad*. Hasta dónde se puede controlar al acceso al software o a los datos por personas no autorizadas.
- *Usabilidad*. El esfuerzo necesario para aprender a operar, preparar los datos de entrada e interpretar los resultados del programa.
- *Facilidad de mantenimiento*. El esfuerzo necesario para localizar y arreglar un error en el programa.
- *Flexibilidad*. El esfuerzo necesario para modificar un programa.
- *Facilidad de prueba*. El esfuerzo necesario para probar un programa para asegurarse de que realiza su función pretendida.
- *Portabilidad*. El esfuerzo necesario para transferir el programa de un entorno de sistema de hardware y/o software a otro.
- *Reusabilidad*. Hasta dónde se puede volver a usar un programa (o alguna de sus partes) en otras aplicaciones, en relación al alcance de las funciones que realiza el programa.
- *Interoperatividad*. El esfuerzo necesario para acoplar un sistema con otro.

⁶ McCall, J., P. Richards y G. Walters, Factors in Software Quality, NTIS, USA, 1977. Tres Volúmenes.

Es difícil desarrollar medidas directas de los factores de calidad anteriores; no obstante, se han desarrollado las siguientes métricas que facilitan la estimación de la calidad de un producto de software:

- *Facilidad de auditoría.* La facilidad con la que se puede comprobar el cumplimiento de los estándares.
- *Exactitud.* El grado con el cual se realizan los cálculos y controles conforme a las reglas.
- *Estandarización de comunicaciones.* El grado de empleo de estándares de interfaces y protocolos.
- *Complitud.* El grado con que se ha logrado la implementación total de una función.
- *Concisión.* Lo compacto que es el programa en términos de líneas de código.
- *Consistencia.* El empleo de un diseño uniforme y de técnicas de documentación a lo largo del proyecto de desarrollo del software.
- *Estandarización de los datos.* El empleo de estructuras y tipos de datos estándares a lo largo del programa.
- *Tolerancia al error.* El daño causado cuando un programa encuentra un error.
- *Eficiencia de ejecución.* El rendimiento del funcionamiento de un programa.
- *Capacidad de expansión.* El grado con que se pueden ampliar el diseño de datos o procedimental.
- *Generalidad.* La amplitud de aplicación potencial de los componentes del programa.
- *Independencia del hardware.* El grado con que se desacopla el software del hardware donde opera.
- *Instrumentación.* El grado con que el programa vigila su propio funcionamiento e identifica los errores que ocurren.
- *Modularidad.* La independencia funcional de componentes de programa.
- *Operatividad.* La facilidad de operación de un programa.
- *Seguridad.* La disponibilidad de mecanismos que controlan o protegen los

programas y los datos.

- *Autodocumentación.* El grado en que el código fuente proporciona documentación significativa.
- *Simplicidad.* El grado de facilidad con que se puede entender un programa.
- *Independencia del sistema.* El grado de autosuficiencia del programa respecto a las características de lenguaje de programación no estándar y características del sistema operativo.
- *Trazabilidad.* La capacidad de seguir una representación del diseño o un componente real del programa hasta los requisitos.
- *Formación.* El grado en que ayuda el software a manejar el sistema a los nuevos usuarios.

1.8.2 MANTENIMIENTO.

El término *mantenimiento* se ha aplicado al proceso de modificar un programa después de que ha sido desarrollado y está en uso. Las modificaciones implican simples cambios para corregir errores de codificación, cambios más extensos para corregir errores de diseño o cambios drásticos para corregir errores de especificación o para acomodarse a nuevos requerimientos.

Es imposible producir sistemas de cualquier tamaño que no necesiten mantenimientos. A lo largo de la vida útil de un sistema, sus requerimientos originales serán modificados para reflejar las necesidades cambiantes. Ya que el mantenimiento es inevitable, los sistemas deben ser diseñados e implementados de tal manera que los problemas de mantenimiento sean minimizados.

El mantenimiento del software cae en tres categorías:

- Mantenimiento perfectivo.
- Mantenimiento adaptativo.
- Mantenimiento correctivo.

El *mantenimiento perfectivo* significa cambios que mejoran el sistema de alguna manera sin cambiar su funcionalidad. El *mantenimiento adaptativo* es el mantenimiento que se requiere debido a los cambios en el ambiente del programa. El *mantenimiento correctivo* modifica de alguna manera el funcionamiento de ciertas partes del programa.

El proceso de mantenimiento se inicia por un conjunto de solicitudes de cambio por parte de los usuarios del sistema o por parte de los administradores del mismo. Esta petición, generalmente involucra elementos del mantenimiento adaptativo, correctivo y perfectivo. Los cambios se implementan y son validados.

Finalmente, se libera una nueva versión del sistema. El proceso de esta manera se vuelve iterativo con un nuevo conjunto de cambios propuestos para la nueva liberación.

El mantenimiento es una iteración del proceso de desarrollo y se deben aplicar los estándares de comparación. Los nuevos requerimientos deben ser validados, los componentes del sistema deben ser rediseñados e implementados y parte, o la totalidad del sistema, debe ser probada.

2. ESTUDIO INICIAL Y ANÁLISIS DEL *SIIMCELE*.

2.1 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL DE LA *MEDIATECA*.

En el CELE se ha puesto en marcha el proyecto *MEDIATECA*, en el cual, se realizan las siguientes actividades académico-administrativas:

- 1. Inscripción y elaboración de credenciales de usuarios.*
- 2. Control de acceso de usuarios a la *MEDIATECA*.*
- 3. Registro de materiales del acervo.*
- 4. Bitácora de las actividades de aprendizaje realizadas por el usuario y del material empleado por éste.*
- 5. Asesorías de aprendizaje autodirigido, diversos talleres y elaboración de documentos didácticos.*
- 6. Proposición de compra de equipo y materiales y evaluación de los mismos.*
- 7. Coordinación de personal y prestadores de servicio social.*

Después de una entrevista con los administrativos de la *MEDIATECA*, se estableció que ellos deseaban la automatización de las actividades que se describen a continuación:

- 1. Inscripción y elaboración de credenciales de usuarios.*

La persona interesada en hacer uso de los servicios de la *MEDIATECA* debe pertenecer a una de las siguientes categorías:

- Alumno inscrito o tesista.
- Empleado:
 - Académico.
 - Administrativo.

Si es alumno:

- Ser estudiante de licenciatura o posgrado del Campus de Ciudad Universitaria, Escuela Nacional de Música, Escuela Nacional de Enfermería y Obstetricia, Escuela Nacional de Artes Plásticas.
- Ser tesista (deberá comprobar con documentos oficiales que se encuentra en etapa de desarrollo de tesis).
- Tener credencial de estudiante de la UNAM y tira de materias actualizada.
- Proporcionar dos fotografías tamaño infantil.
- Cubrir cuota de inscripción, la cual tendrá vigencia seis meses a partir de la fecha de pago.

Si es empleado:

- Ser trabajador académico o administrativo de la Ciudad Universitaria, Escuela Nacional de Música, Escuela Nacional de Enfermería y Obstetricia, Escuela Nacional de Artes Plásticas.
- Tener credencial de trabajador actualizada y/o último talón de pago.
- No se admiten trabajadores por honorarios.
- Proporcionar dos fotografías tamaño infantil.
- Cubrir cuota de inscripción, la cual tendrá vigencia de seis meses a partir de la fecha de pago.

La solicitud que debe llenar el interesado en hacer uso de los servicios de la *MEDIATECA*, contempla los siguientes datos de interés para la coordinación:

- Número de cuenta o RFC.
- Nombre.
- Categoría:
 - Si es alumno, proporcionará el nombre de su carrera.
 - Si es empleado, proporcionará el nombre de su dependencia.

- Idioma de autoaprendizaje (inglés o francés).
- Fecha de nacimiento (dd/mm/aa).
- Sexo.
- Domicilio particular del interesado:
 - Calle.
 - Número.
 - Colonia.
 - Delegación.
 - Código Postal.
 - Teléfono.
- Información adicional:
 - Si el interesado, además es alumno del CELE, proporcionará el nombre de su(s) idioma(s) y el semestre que cursa.

De esta manera, el interesado es inscrito en formatos de papel previstos para este proceso y se adiciona la fecha de inscripción; la *MEDIATECA* entonces, imprime una credencial al interesado para que pueda acceder a los servicios.

2. Control de acceso de usuarios a la MEDIATECA.

Esta actividad se lleva a cabo cuando el usuario ingresa a las instalaciones y presenta su credencial en el módulo de ingreso, la persona responsable de atender el módulo, verifica personalmente la vigencia de este documento de identificación y permite o niega la entrada.

3. Registro de materiales del acervo (libros de texto, diccionarios, publicaciones periódicas, CD-ROM's, audiocassettes, videocassettes, programas televisivos y fichas de trabajo).

Este proceso implica dar de alta el material adquirido, su catalogación y clasificación. Para registrar el acervo se consideran los siguientes atributos:

- Número de adquisición.
- ISBN.
- ISSN.
- Idioma del material.
- Clasificación del material.
- Autor personal y corporativo.
- Título del material.
- Edición.
- Pie de imprenta.
- Descripción física.
- Serie.
- Notas generales.
- Descriptores (sinónimo bibliotecónomo de temas).
- Fondo de origen.
- Resumen.
- Presencia de acento del idioma (inglés británico, inglés americano, francés canadiense, etc.).
- Referencia.
- Nivel de dificultad.
- Número de ejemplares.
- Fecha de ingreso.
- Catalogador.

El proceso se realiza de la siguiente manera:

- A) Se toman los materiales de los anaqueles y se trasladan al escritorio de

captura.

B) El número de clasificación, autor y título de los materiales son capturados en una hoja de cálculo (EXCEL). Esto significa que, el almacenamiento de datos se realiza en archivos individuales.

C) Los materiales cuyos datos fueron ya capturados son reubicados a los anaqueles.

D) Finalmente, se imprime el listado de los datos tal y como fueron capturados.

4. Bitácora de las actividades de aprendizaje realizadas por el usuario y del material empleado por éste.

Este proceso se realiza al terminar cada sesión de trabajo. Al elaborar la bitácora, el usuario contesta una serie de preguntas impresas en tres páginas, que tienen como fin permitir a los administradores cumplir los siguientes objetivos tendientes a mejorar el servicio:

- Adición de materiales sugeridos por el usuario al acervo.
- Eliminar materiales inútiles.
- Actualización de materiales.
- Elaboración de horarios de asesoría.
- Evaluación de aptitudes de los asesores.
- Reubicación de materiales.
- Evaluación del aprendizaje de los usuarios.
- Conocer nuevas organizaciones de aprendizaje por parte de los usuarios.

Las preguntas de la bitácora son:

- Nombre.

- **Materiales utilizados:**
 - ¿Por qué escogiste este material?
 - ¿Cuál fue la utilidad de este material para ti?
 - ¿Te fue fácil encontrar el material?
- ¿Qué actividades realizaste?
- ¿Tuviste algún problema?
 - ¿Cuáles?
- ¿Realizaste algún tipo de autoevaluación?
 - ¿De qué forma te autoevaluaste?
- ¿Tenías un objetivo al llegar?
 - ¿Lograste tu objetivo?
 - ¿Por qué no lo lograste?
- ¿Necesitas asesoría?
 - ¿Por qué?
 - ¿Cómo te resultó la asesoría?
- ¿Piensas que la *MEDIATECA* te está ayudando a cumplir tus objetivos?
- ¿Cómo piensas optimar tu desempeño?
- Describe brevemente cómo organizaste tu trabajo.
- ¿Tienes algún comentario adicional?

2.2 ANÁLISIS DE LA PROBLEMÁTICA DE LA SITUACIÓN ACTUAL DE LA *MEDIATECA* Y PROPUESTA DE SOLUCIÓN.

En este momento, es factible hacer las siguientes observaciones a la situación descrita anteriormente y siguiendo el orden ya establecido:

1. Inscripción y elaboración de credenciales de usuarios.

En el proceso de inscripción existe retraso en la elaboración de credenciales: primero, si la persona interesada en emplear los servicios de la *MEDIATECA* cumple con los requisitos enumerados, debe llenar manualmente una forma de inscripción; después, se capturan el nombre, idioma y número de cuenta/RFC y, finalmente, se imprimen en lo que será su credencial.

2. Control de acceso de usuarios a la *MEDIATECA*.

La persona encargada de verificar la validez de inscripción de cada usuario (la inscripción tiene vigencia de seis meses), debe consultar manualmente los registros de inscripción y establecer mentalmente, si el usuario tiene acceso o no a las instalaciones de la *MEDIATECA*.

3. Registro de materiales del acervo.

Tomando en cuenta el proceso con el que se realiza el registro de los materiales del acervo, resulta tedioso y complicado intentar una consulta a los registros de los materiales. Además, se carece de seguridad, porque es posible introducir datos erróneos; por ejemplo, se han introducido registros como el siguiente, en el cual se puede observar que el segundo registro posee un identificador nulo.

IDENTIFICADOR	AUTOR	TÍTULO
C00025	Roxanne Burnes	Americas Dictionary
	Edwin B. Williams	New World Dictionary

Y en este otro caso, existe un identificador duplicado:

IDENTIFICADOR	AUTOR	TÍTULO
C00025	Roxanne Burnes	Americas Dictionary
C00025	Edwin B. Williams	New World Dictionary

De la manera en que se emplea la hoja de cálculo en la *MEDIATECA*; ésta no facilita la manipulación de la información que permita conocer con precisión y rapidez la existencia del acervo. Asimismo, no se hacen ordenamientos, búsquedas, selecciones, y en general, ningún tipo de manipulación de datos que facilite la toma de decisiones académico-administrativas.

Cabe destacar, que la capacitación técnica para el uso de esa hoja de cálculo, requiere aproximadamente de un mes, período en el cual, el personal de la *MEDIATECA* desatiende otras actividades, tales como: atención al usuario, reubicación de los materiales y asesorías.

4. Bitácora de las actividades de aprendizaje realizadas por el usuario y del material empleado por éste.

Observando las preguntas de la bitácora, se establece que éstas no proporcionan información cuantificable, debido a su ambigüedad, y no facilitan una visión inmediata del aprovechamiento general de los usuarios. En consecuencia, es imposible realizar un informe que permita la observancia del cumplimiento de los objetivos de la *MEDIATECA*.

Observaciones adicionales.

Junto con la problemática ya descrita de las actividades académico-administrativas consideradas, en general, pueden mencionarse estas otras dificultades:

- Falta de generación de reportes que apoyen la planeación y toma de decisiones de la coordinación de la *MEDIATECA*.
- Debido al carácter manual del registro de usuarios, del registro de materiales del acervo, del inventario del acervo y de la bitácora de actividades del

usuario, se presentan errores que van desde omisión de datos hasta duplicación de registros, además, existe traspapeleo, falta de disponibilidad inmediata de la información y pérdida de la misma.

Propuesta de solución.

Considerando la situación ya presentada en la Problemática de la Situación Actual, surge en la *MEDIATECA* la necesidad de implantar un sistema automatizado integral de información, atendiendo a cada una de estas actividades:

1. *Inscripción y elaboración de credenciales de usuarios.*
2. *Control de acceso de usuarios a la MEDIATECA.*
3. *Registro de materiales del acervo (libros de texto, diccionarios, publicaciones periódicas, CD-ROM's, audiocassettes, videocassettes, programas televisivos y fichas de trabajo) y su posterior consulta por parte de los usuarios.*
4. *Bitácora de las actividades de aprendizaje realizadas por el usuario y del material empleado por éste.*

1. Para la fase de *inscripción y elaboración de credenciales de usuarios*, se propone el desarrollo de una forma de captura computarizada (formato o formulario), consistente en una interfaz amigable en su empleo, que hará innecesario el uso de papel, garantizará la introducción de datos válidos, la seguridad e integridad de esos datos y contemplará, los mismos campos de interés para la *MEDIATECA*.

La persona interesada en hacer uso de los servicios de la *MEDIATECA*, una vez que cumple los requisitos de inscripción, podrá ser dado de alta de manera inmediata.

Existirá un formato con el cual se seleccionarán los datos de los usuarios a quienes se les imprimirá su credencial.

2. En la fase de *control de acceso de usuarios a la MEDiateca*, la propuesta es incluir en las credenciales de usuario un código de barras que será leído mediante un lector de ranura, lo que agilizará la verificación de la vigencia de la inscripción. La pérdida de vigencia de alguna inscripción será indicada mediante un sonido distintivo. La lectura será capturada en un formato desarrollado con ese fin, y gracias a esta captura, será posible generar reportes que muestren la frecuencia de asistencia de los usuarios a la *MEDIATECA*.

3. En el proceso de *registro de materiales del acervo*, se desarrollará, de nueva cuenta, un formato, que sea sencillo de llenar y en el cual se puedan capturar todos los datos que caracterizan a los materiales de la *MEDIATECA* (ya mencionados en la descripción de la situación actual). Este formato, también ofrecerá la garantía de la introducción de datos válidos y la seguridad e integridad de los mismos. Adicionalmente, será posible imprimir los códigos de barras que identifiquen y clasifiquen a cada material, esto permitirá la captura de la información de los materiales mediante lectores de códigos de barra por parte de los usuarios al momento de responder la bitácora correspondiente. Se sustituirá el empleo de la hoja de cálculo y se evitará la carencia de algún dato en un registro o la duplicación de datos, que son los problemas más comúnmente detectados en este proceso al usar la hoja de cálculo como sucede actualmente en la *MEDIATECA*.

Debido a la facilidad de uso del nuevo formato a desarrollar, desaparecerá la capacitación técnica que se empleaba en la *MEDIATECA* para el manejo de la hoja de cálculo con la que realizaban su inventario.

Gracias al almacenamiento que se realizará de la información de los materiales, los usuarios serán capaces de consultar los datos del acervo con el objeto de localizar el material que necesiten, mediante una búsqueda por palabras o al menos mediante tres letras. La búsqueda del material considerará los atributos de título, autor y temas, eligiendo si el material a localizar se trata de un libro de texto, diccionario, publicación periódica, CD-ROM, audiocassette o videocassette, además del idioma

correspondiente a cada material.

4. Para el proceso de la *Bitácora de las actividades de aprendizaje realizadas por el usuario y del material empleado por éste*, la propuesta es, eliminar la ambigüedad en las preguntas actuales para poder plasmarlas en un formato, sin perder de vista los objetivos por los cuales se estableció la Bitácora; y de esta manera, capturar los datos de interés para la *MEDIATECA*.

La información introducida por los usuarios en la bitácora (computarizada), permitirá la generación de informes útiles en la toma de decisiones por parte de la coordinación de la *MEDIATECA*.

En general, la propuesta de solución contempla la captura de información en formatos muy fáciles de utilizar y la consulta a la base de datos por los administradores requiriendo un mínimo de conocimientos previos sobre el uso de la computadora.

El manejo del sistema propuesto será similar al manejo de una aplicación que trabaje bajo ambiente Windows, es decir, estarán disponibles menús e iconos.

La propuesta de solución surgirá del *análisis y especificación estructurada* y del *diseño estructurado* explicados anteriormente.

Con el objeto de tener una visión general de la problemática de la *MEDIATECA* se considerarán las restricciones y la factibilidad técnica del sistema.

Restricciones.

Para establecer las propuestas indicadas, se han considerado las siguientes restricciones:

- El sistema debe desarrollarse e implementarse, haciendo uso de los recursos de software y hardware existentes en la *MEDIATECA*.
- El sistema debe ser desarrollado de tal manera, que requiera mínima capacitación técnica sobre computación para poder hacer uso de él. Es decir, una interfaz gráfica que proporcione un ambiente de operación amigable y de fácil interacción.
- Incluir un módulo de ayuda en línea que permita al usuario en cuestión, contar con una guía que le auxilie durante la operación del sistema.

2.3 ANÁLISIS DE FACTIBILIDAD TÉCNICA DE LA PROPUESTA DE SOLUCIÓN PARA LA *MEDIATECA*.

Recursos de hardware disponibles en la *MEDIATECA*.

- Sistema de desarrollo o sistema anfitrión:

Computadora Personal con procesador Pentium a 60 Mhz. Disco duro de 540 Megabytes y 8 Megabytes de memoria RAM.

- Máquinas objetivo:

La implantación del sistema será en una red Novell, para que pueda ejecutarse en computadoras que presentan las siguientes especificaciones:

6 equipos:

*Procesador 486 DX a 80 Mhz.

*8 MB en RAM.

*Monitores VGA.

*Un promedio de 100 MB disponibles en disco duro.

Se ha planeado la siguiente disposición de las computadoras:

1. En la recepción, se ubicarán dos computadoras, en una de las cuales será posible, exclusivamente, controlar el acceso de los usuarios a la *MEDIATECA*. En la otra, se realizará el proceso de inscripción.
2. En el área de consulta, existirán dos computadoras, una que permitirá a los usuarios contestar la bitácora de las actividades de aprendizaje y del material empleado, y la otra, disponible para consultar el acervo de la *MEDIATECA*.
3. Otra computadora se ubicará en la coordinación, y con ella podrán realizarse consultas en general.
4. Una computadora más estará disponible para la captura de la información de los materiales del acervo.

- Elementos de hardware del nuevo sistema.

Adicional a las computadoras personales empleadas, se utilizarán lectores de código de barras: uno de ranura para el proceso de acceso de usuarios a la *MEDIATECA* y dos "WAND Reader" (de lápiz) para los procesos de bitácora y registro de materiales. Estos lectores eliminarán la necesidad de teclear los datos correspondientes, minimizando tiempo y evitando errores de captura manual.

Recursos de Software disponibles en la *MEDIATECA*.

- Sistema Operativo de Red *Novell Netware 4.1*
- Sistema Operativo *Windows 3.11*
- Lenguaje de programación y manejador de bases de datos *Visual FoxPro*.

Más adelante y como parte de los requisitos establecidos por la *MEDIATECA*, se definirán las características de la herramienta que servirá como desarrollador del sistema: *Visual FoxPro*.

3. ANÁLISIS Y ESPECIFICACIÓN ESTRUCTURADA DEL *SIIMCELE*.

3.1 EL DFD DEL *SIIMCELE*.

En el *SIIMCELE* se toman en cuenta las siguientes entidades:

1. Usuario.
2. Acervo.
3. Administrativo.
4. Dispositivo de salida (monitor o impresora).
5. Coordinador de la *MEDIATECA*.

Estas entidades interaccionan con los procesos:

1. Inscripción.
2. Control de acceso.
3. Llenado de bitácora.
4. Registro de acervo.
5. Consulta a la base de datos.
6. Reinscripción.
7. Registro de administrativo.
8. Consulta al acervo.

Los procesos se han establecido para automatizar las actividades académico-administrativas ya descritas.

Los datos obtenidos o procesados por el *SIIMCELE* serán almacenados en las siguientes tablas principales:

1. Usuarios.
2. Frecuencia de usuarios.
3. Bitácora:

- De usuarios.
 - De acervo.
4. Acervo.
 5. Privilegios.

Todo lo anterior queda claramente expresado en los Diagramas de Flujo de Datos, que han sido desarrollados desde un nivel 0 de detalle hasta un nivel 2 para una mejor representación del SIIMCELE. Todos estos diagramas se muestran en las siguientes figuras, iniciando con la 3.1 y terminando con la 3.10.

DFD. Nivel 0 del SIIMCELE.

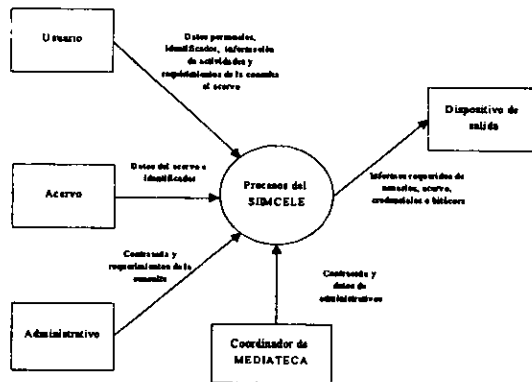


FIG. 3.1

DFD. Nivel 1 del *SIMCELE*.

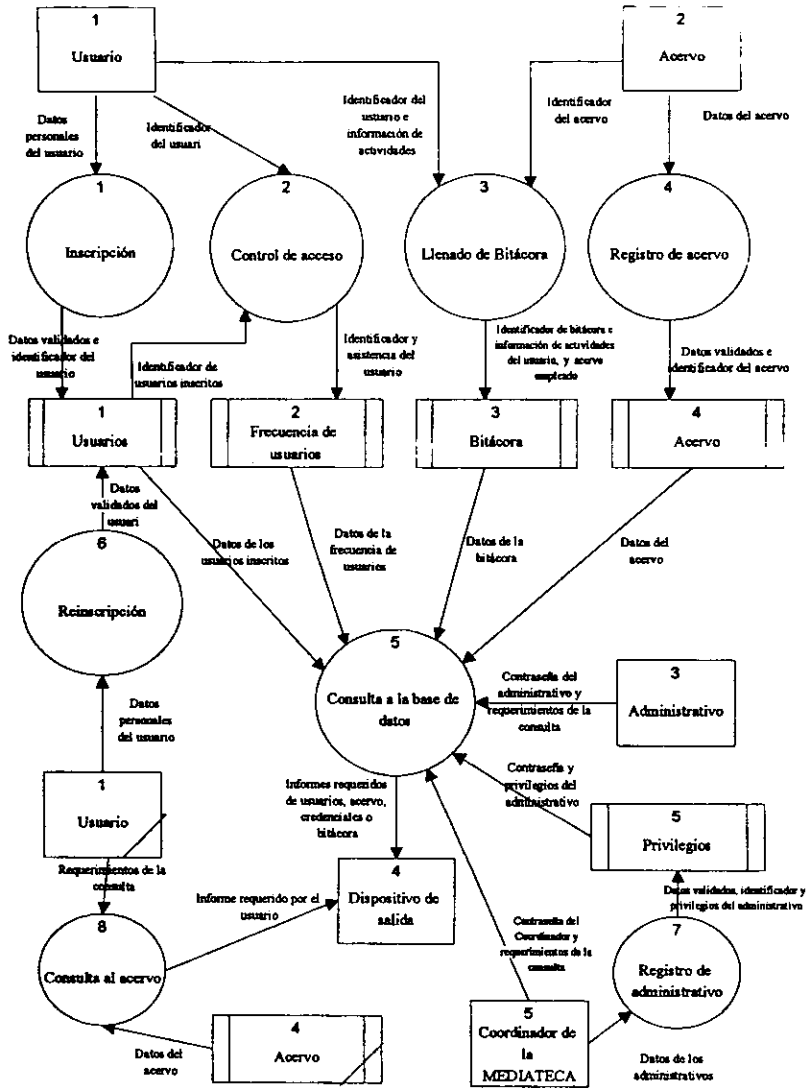


FIG. 3.2

DFD. Nivel 2 del proceso Inscripción.

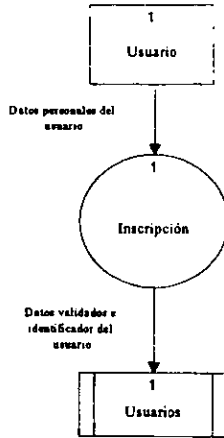


FIG. 3.3

DFD. Nivel 2 del proceso Control de Acceso.

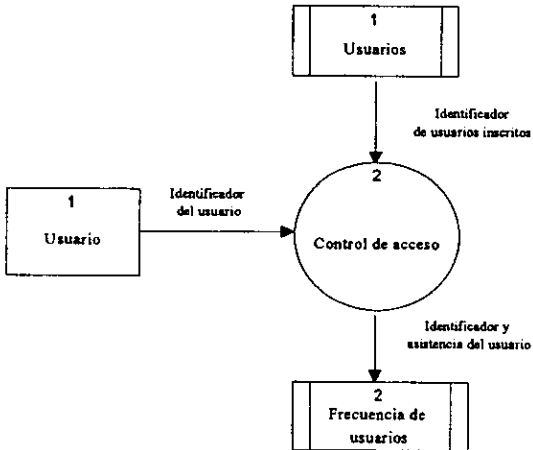


FIG. 3.4

DFD. Nivel 2 del proceso Llenado de bitácora.

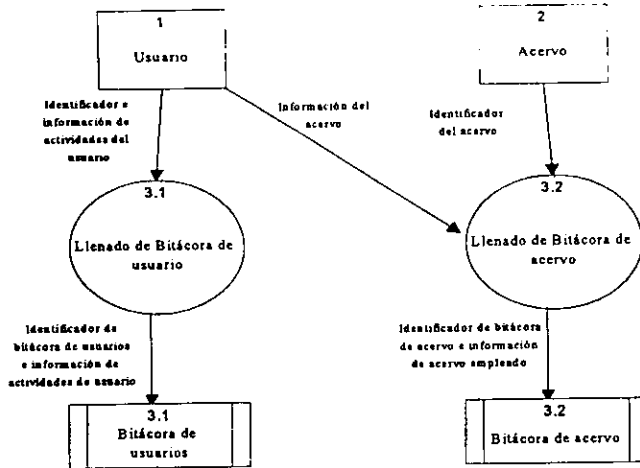


FIG. 3.5

DFD. Nivel 2 del proceso Registro de acervo.

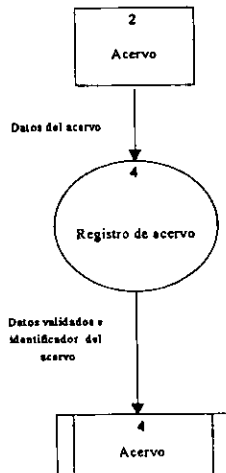


FIG. 3.6

DFD. Nivel 2 del proceso Consulta.

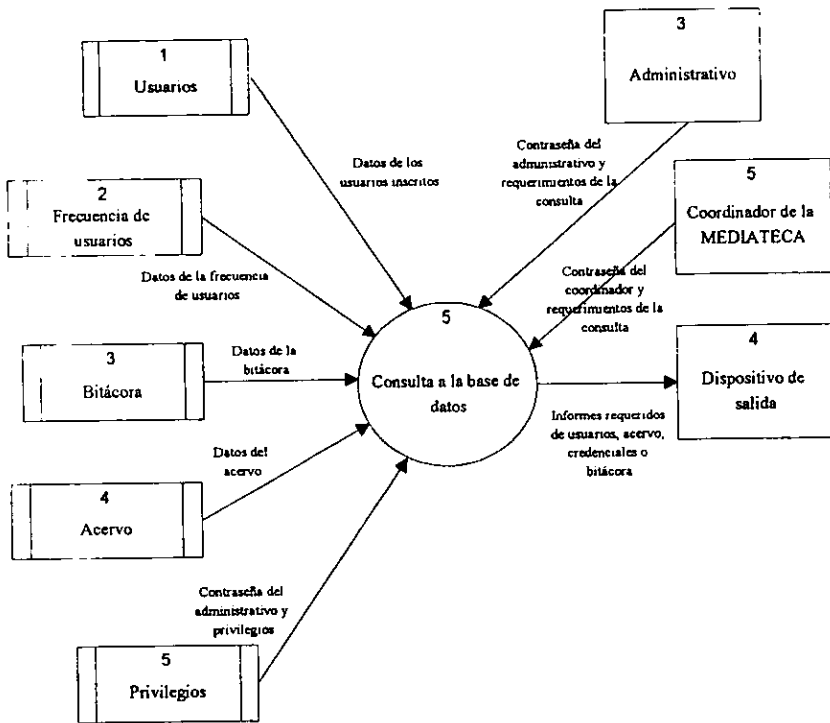


FIG. 3.7

DFD. Nivel 2 del proceso Reinscripción.

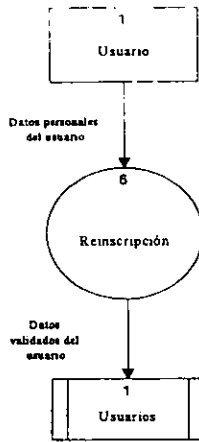


FIG. 3.8

DFD. Nivel 2 del proceso Registro de administrativo.

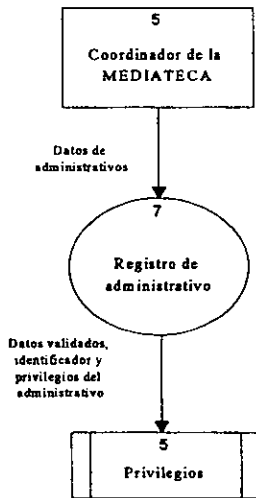


FIG. 3.9

DFD. Nivel 2 del proceso Consulta al acervo.

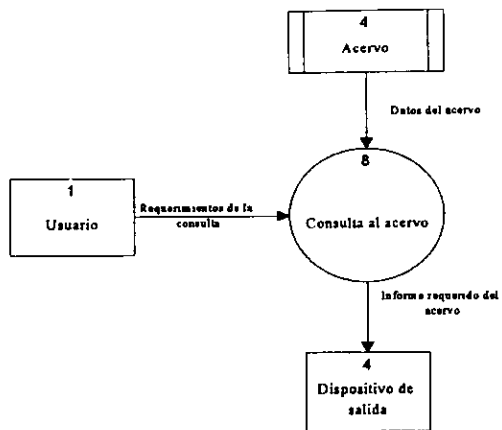


FIG. 3.10

3.2 DICCIONARIO DE DATOS DEL SIIMCELE.

El diccionario de datos del SIIMCELE recopila los nombres y atributos de los datos de cada proceso y que interactúan con las entidades y tablas de almacenamiento del sistema.

1. Inscripción.

Nombre	datos personales del usuario
Dónde y cómo se usa	entidad Usuario (salida) proceso Inscripción (entrada)
Descripción	nombre + (número_cuenta rfc) + identificador_categoria + identificador_idioma + número_inscripciones + fecha_registro + identificador_procedencia + fecha_inscripción + fecha_nacimiento + sexo + calle + colonia + delegación-municipio + código-postal + teléfono + lengua1 + semestre_lengua1 + lengua2 + semestre_lengua2 + observaciones + inscrito + no_inscripciones

Nombre	nombre
Alias	nombre
Descripción	apellido paterno + apellido materno + nombre
Longitud	40 (carácter)

Nombre	número_cuenta rfc
Alias	cta rfc
Descripción	número de cuenta de la UNAM o Registro Federal de Causantes
Longitud	15 (carácter)

Nombre	identificador_categoria
Alias	cat
Descripción	(01.- estudiante 02.- académico 03.- administrativo 04.- tesista otro)
Longitud	2 (carácter)

Nombre	identificador_idioma
Alias	idi
Descripción	(01.- inglés 02.- francés 03.- italiano 04.- alemán 05.- portugués 06.- ruso 07.- japonés 08.- árabe 09.- hebreo 10.- griego moderno 11.- chino 12.- coreano 13.- sueco 14.- catalán otro)
Longitud	2 (carácter)
Comentario	se refiere al idioma de aprendizaje en la <i>MEDIA TECA</i>

Nombre	número_inscripciones
Alias	no_ins
Descripción	número consecutivo
Longitud	2 (numérico)

Nombre	fecha_registro
Alias	f_r
Descripción	dd/mm/aaaa
Longitud	8 (fecha)
Comentario	Se refiere al día en que se realiza la captura de los datos del usuario.

Nombre	identificador_procedencia
Alias	pro
Descripción	(01.-Facultad de Arquitectura + 02.-Facultad de Ciencias + 03.-Facultad de Ciencias Políticas + ...)
Longitud	3 (carácter)
Comentario	Se refiere a escuelas, facultades, institutos, dependencias, etc. Aproximadamente 150.

Nombre	fecha_inscripción
Alias	f_ins
Descripción	dd/mm/aaaa
Longitud	8 (fecha)

Nombre	fecha_nacimiento
Alias	f_nac
Descripción	dd/mm/aa
Longitud	8 (fecha)

Nombre	sexo
Alias	sexo
Descripción	(f m)
Longitud	1 (carácter)

Nombre	calle
Alias	calle
Descripción	nombre de la calle, número, edificio, número interior, etc.
Longitud	memo

Nombre	colonia
Alias	col
Longitud	memo

Nombre	delegación-municipio
Alias	del
Longitud	20 (carácter)
Comentario	delegación o municipio

Nombre	código-postal
Alias	cp
Longitud	10 (carácter)

Nombre	teléfono
Alias	tel
Descripción	número telefónico particular del usuario
Longitud	10 (carácter)

Nombre	lengua1
Alias	1
Descripción	(01.- inglés 02.- francés 03.- italiano 04.- alemán 05.- portugués 06.- ruso 07.- japonés 08.- árabe 09.- hebreo 10.- griego moderno 11.- chino 12.- coreano 13.- sueco 14.- catalán 15.- otro)
Longitud	2 (carácter)
Comentario	Se refiere al idioma de estudio en el CELE que se está cursando.

Nombre	semestre lengua1
Alias	seml 1
Descripción	número del semestre correspondiente a la lengua 1
Longitud	2 (numérico)

Nombre	lengua2
Alias	2
Descripción	(01.- inglés 02.- francés 03.- italiano 04.- alemán 05.- portugués 06.- ruso 07.- japonés 08.- árabe 09.- hebreo 10.- griego moderno 11.- chino 12.- coreano 13.- sueco 14.- catalán 15.- otro)
Longitud	2 (carácter)
Comentario	Se refiere al idioma de estudio en el CELE y al semestre que se está cursando (en caso de estudiar un segundo idioma).

Nombre	semestre lengua2
Alias	seml 2
Descripción	número del semestre correspondiente a la lengua 2
Longitud	2 (numérico)

Nombre	observaciones
Alias	obs
Descripción	permite la introducción de texto, para comentarios sobre el usuario
Longitud	memo

Nombre	inscrito
Alias	ins
Descripción	(inscrito no inscrito)
Longitud	1 (lógico)
Nombre	no inscripciones
Alias	no_ins
Longitud	2 (numérico)

Nombre	datos validados e identificador del usuario
Dónde y cómo se usa	proceso Inscripción (salida) almacenamiento Usuarios (entrada)
Comentario	tipos de datos personales del usuario correctos e identificador del usuario

Nombre	identificador_usuario
Alias	idusr
Descripción	año de inscripción + número consecutivo + código de barras
Longitud	2 (numérico) + 4 (numérico)

Nombre	usuarios
Alias	usr
Dónde y cómo se usa	datos Datos validados e identificador de usuario (salida)
Descripción	tabla de almacenamiento

2. Control de acceso.

Nombre	identificador del usuario
Alias	idusr
Dónde y cómo se usa	entidad Usuario (salida) proceso Control de acceso (entrada)
Descripción	ya descrito
Longitud	ya establecida

Nombre	identificador de usuarios inscritos
Alias	idusr
Dónde y cómo se usa	almacenamiento Usuarios (salida) proceso Control de acceso (entrada)
Descripción	ya descrito
Longitud	ya establecida

Nombre	identificador y asistencia del usuario
Alias	idusr + consec
Dónde y cómo se usa	proceso Control de acceso (salida) almacenamiento Frecuencia de usuarios (entrada)
Descripción	identificador_usuario + consecutivo
Longitud	6 (numérico) + 3 (numérico)

Nombre	frecuencia del usuarios
Alias	frecusr
Dónde y cómo se usa	datos Datos Identificador y asistencia del usuario (salida)
Descripción	tabla de almacenamiento

3. Llenado de bitácora.

3.1 Bitácora de usuarios.

Nombre	Identificador e información de actividades del usuario
Dónde y cómo se usa	entidad Usuario (salida) proceso Llenado de bitácora de usuario
Descripción	usuario + identificador_utilidad_orientación + identificador_problemas_usuario + identificador_objetivo + identificador_orientación + identificador_autoevaluación + [comentarios]

Nombre	usuario
Alias	usr
Descripción	ya descrita (como idusr)
Longitud	6 (numérico)

Nombre	identificador_utilidad_orientación
Alias	util
Descripción	(01.-muy útil 02.-útil 03.-poco útil 04.-nada útil)
Longitud	2 (carácter)

Nombre	identificador_problemas_usuario
Alias	probu
Descripción	(01.-falta orientación para trabajar autónomamente 02.-falta de asesor 03.-problemas técnicos 04.-otro 05.-no tuve ningún problema)
Longitud	2 (carácter)

Nombre	identificador_objetivo
Alias	obj
Descripción	(01.-si tenía objetivo 02.-me faltó tiempo 03.- material inadecuado 04.-cambié el objetivo 05.- objetivo muy difícil 06.-objetivo poco claro 07 otro 08.-si logré el objetivo)
Longitud	2 (carácter)

Nombre	identificador_orientación
Alias	orien
Descripción	(01.-sobre materiales 02.-acerca de las fichas de *aprender a aprender 03.-para autoevaluación 04.- para organizar mi trabajo 05.-sobre lingüística 06.- ninguna)
Longitud	2 (carácter)

Nombre	identificador_autoevaluación
Alias	autoe
Descripción	(01.-cotejé con la clave de respuestas 02.-consulté la transcripción del video o de la cinta 03.-grabé mi voz y la comparé con la original 04.-usé libros de la sección de consulta 05.-comenté mis respuestas con un compañero 06.-comparé mi trabajo con un modelo del libro 07.-platicué con un asesor 08.-otra

ANÁLISIS Y ESPECIFICACIÓN ESTRUCTURADA DEL SIIMCELE

	09.-no realicé autoevaluación)
Longitud	2 (carácter)

Nombre	comentarios
Alias	comm
Descripción	texto
Longitud	memo

Nombre	identificador de bitácora de usuarios e información de actividades del usuario
Dónde y cómo se usa	proceso Llenado de bitácora de usuarios (salida) almacenamiento Bitácora de usuarios (entrada)
Descripción	identificador bitácora usuarios + información actividades usuario

Nombre	identificador bitácora usuarios
Alias	fec
Descripción	dd/mm/aaaa + hh:mm:ss
Longitud	8 (fecha)

Nombre	bitácora de usuario
Alias	bitu
Dónde y cómo se usa	datos Datos Identificador de bitácora de usuarios e información de actividades del usuario (salida)
Descripción	tabla de almacenamiento

3.2 Bitácora de acervo.

Nombre	identificador del acervo e información del acervo
Dónde y cómo se usa	entidades Acervo y Usuario (salida) proceso Llenado de bitácora de acervo (entrada)
Descripción	identificador_acervo + identificador_elección_acervo + identificador_utilidad_acervo + identificador_problemas_acervo

Nombre	identificador_acervo
Alias	ace
Descripción	en la tabla de acervo (idace)
Longitud	6 (numérico)

Nombre	identificador_elección_acervo
Alias	elec
Descripción	(01.-recomendado por un asesor 02.-recomendado por un compañero 03.-al azar 04.-ya lo conocía 05.-lo encontré en la base de datos 06.-me interesa el tema 07.-otro)
Longitud	2 (carácter)

Nombre	identificador_utilidad_acervo
Alias	util
Descripción	(01.-muy útil 02.-útil 03.-poco útil 04.-nada útil)
Longitud	2 (carácter)

Nombre	identificador_problemas_acervo
Alias	probm
Descripción	(01.-material inadecuado para mi nivel 02.-material inadecuado para mis propósitos 03.-no supe hacer con el material 04.-insuficientes ejemplares con el material 05.-insuficientes propuestas de actividades que acompañen a los materiales 06.-otro 07.-no tuve ningún problema)
Longitud	2 (carácter)

Nombre	identificador de bitácora de acervo e información de acervo empleado
Dónde y cómo se usa	proceso Llenado de bitácora de acervo (salida) almacenamiento Bitácora de acervo (entrada)
Descripción	identificador_bitácora_acervo + información_acervo

Nombre	identificador_bitácora_acervo
Alias	fec
Descripción	ya descrita
Longitud	8 (fecha)

Nombre	bitácora de acervo
Alias	Bitm
Dónde y cómo se usa	datos identificador de bitácora de acervo e información de acervo empleado (salida)
Descripción	tabla de almacenamiento

4. Registro de acervo.

Nombre	datos del acervo
Dónde y cómo se usa	entidad Acervo (salida) proceso Registro de acervo (entrada)
Descripción	isbn + issn + identificador_tipo_acervo + identificador_idioma + clasificación + autor-personal + autor-corporativo + título + edición + pie_imprenta + descripción-física + serie + notas-generales + fecha_ingreso + descriptores + consulta + fondo_origen + resumen + identificador_acento + referencia + nivel_dificultad + número_ejemplares + catalogador

Nombre	isbn
Alias	isbn
Descripción	international standard book number
Longitud	20 (carácter)

Nombre	issn
Alias	issn
Descripción	international standard serial number
Longitud	20 (carácter)

Nombre	identificador tipo acervo
Alias	mat
Descripción	(01.-libros de texto 02.-diccionarios 03.-publicaciones periódicas 04.- CD-ROM's 05.-audiocassettes 06.-videocassettes 07 fichas de trabajo 08.-otro)
Longitud	2 (carácter)

Nombre	identificador idioma
Alias	idi
Descripción	(01.- inglés 02.- francés 03.- italiano 04.- alemán 05.- portugués 06.- ruso 07.- japonés 08.- árabe 09.- hebreo 10.- griego moderno 11.- chino 12.- coreano 13.- sueco 14.- catalán 15.-otro)
Longitud	2 (carácter)

Nombre	clasificación
Alias	cla
Descripción	código de letras + código de números
Longitud	30 (carácter)

Nombre	autor-personal
Alias	aut_per
Descripción	apellido (s) + nombre (s) + fecha de nacimiento / muerte
Longitud	100 (carácter)

Nombre	autor-corporativo
Alias	aut_cor
Descripción	nombre de la institución + unidad subordinada + lugar + fecha
Longitud	memo

Nombre	título
Alias	tit
Descripción	título + subtítulo + autores
Longitud	200 (carácter)

Nombre	edición
Alias	edi
Descripción	(2a. edición 3a. edición 4a. edición . . .)
Longitud	20 (carácter)

Nombre	pie imprenta
Alias	pie_imp
Descripción	lugar (es) + editorial + fecha
Longitud	200 (carácter)

Nombre	descripción-física
Alias	desc
Descripción	(paginación + ilustraciones + dimensiones + material complementario unidad + tiempo + sonido + color + formato)
Longitud	150 (carácter)

Nombre	serie
Alias	serie
Descripción	título de la serie + (número parte sección) + volumen + issn de la serie
Longitud	80 (carácter)

Nombre	notas-generales
Alias	nota
Descripción	información que incluye elementos como: bibliografía, hojas de respuestas de ejercicios, encuadernación, etc.
Longitud	memo
Nombre	fecha ingreso
Alias	f_ing
Descripción	dd / mm / aaaa
Longitud	8 (fecha)

Nombre	descriptores
Alias	descri
Descripción	expresa el contenido significativo del material
Longitud	254 (carácter)

Nombre	consulta
Alias	cons
Descripción	(s n)

ANÁLISIS Y ESPECIFICACIÓN ESTRUCTURADA DEL SIIMCELE

Longitud	1 (carácter)
Comentario	Indica si el material estará disponible para su consulta por los usuarios de la <i>Mediateca</i> .

Nombre	fondo_origen
Alias	fondo
Descripción	nombre de la institución + unidad subordinada + lugar
Longitud	40 (carácter)

Nombre	resumen
Alias	resu
Descripción	notas hechas por un especialista acerca del material
Longitud	memo

Nombre	identificador_acento
Alias	acen
Descripción	(01.-británico 02.-americano ...)
Longitud	2 (carácter)

Nombre	referencia
Alias	ref
Descripción	notas sobre autores, materiales o títulos relacionados
Longitud	memo

Nombre	nivel_dificudad
Alias	dif
Descripción	(básico + intermedio + ...)
Longitud	30 (carácter)

Nombre	catalogador
Alias	cata
Descripción	iniciales de la persona que realizó catalogación
Longitud	4 (carácter)

Nombre	datos validados e identificador del acervo
Descripción	datos del acervo + identificador_acervo

Nombre	identificador_acervo
Alias	idace
Descripción	número consecutivo (también se conoce como número de adquisición)
Longitud	6 (numérico)

Nombre	Acervo
Alias	Ace
Dónde y cómo se usa	datos Datos validados e identificador del acervo (entrada)
Descripción	tabla de almacenamiento

5. Consulta a la base de datos.

Nombre	contraseña del administrativo y requerimientos de la consulta
Alias	passwd + requerimientos de la consulta
Dónde y cómo se usa	entidad Administrativo (salida) proceso Consulta a la base de datos (entrada)
Descripción	passwd + requerimientos de consulta a las tablas: usuarios, frecuencia de usuarios, bitácora, acervo o catálogos correspondientes

Nombre	contraseña del coordinador y requerimientos de la consulta
Alias	passwd + requerimientos de consulta
Dónde y cómo se usa	entidad Coordinador de la <i>MEDIATECA</i> (salida) proceso Consulta a la base de datos (entrada)
Descripción	idadm + requerimientos de consulta a las tablas: usuarios, frecuencia de usuarios, bitácora, acervo, administrativos o catálogos correspondientes

Nombre	datos de los usuarios inscritos
Alias	datos validados e identificador de usuario
Dónde y cómo se usa	almacenamiento Usuarios (salida) proceso Consulta a la base de datos (entrada)
Descripción	de los datos de los usuarios inscritos ya descritos se toman los correspondientes a los requerimientos de la consulta

Nombre	datos de la frecuencia de usuarios
Alias	identificador y asistencia del usuario
Dónde y cómo se usa	almacenamiento Frecuencia de usuarios (salida) proceso Consulta a la base de datos (entrada)
Descripción	de los datos de la frecuencia de usuarios ya descritos se toman los correspondientes a los requerimientos de la consulta

Nombre	datos de la bitácora
Alias	identificador de bitácora e información de actividades del usuario y acervo empleado
Dónde y cómo se usa	almacenamiento Bitácora (salida) proceso Consulta a la base de datos (entrada)
Descripción	de los datos de la bitácora ya descritos se toman los correspondientes a los requerimientos de la consulta

Nombre	datos del acervo
Alias	datos validados e identificador del acervo
Dónde y cómo se usa	almacenamiento Acervo (salida) proceso Consulta a la base de datos (entrada)
Descripción	de los datos del acervo ya descritos se toman los correspondientes a los requerimientos de la consulta

Nombre	contraseña y privilegios del administrativo
Alias	contraseña y privilegios validados del administrativo
Dónde y cómo se usa	almacenamiento Privilegios (salida) proceso Consulta a la base de datos (entrada)
Descripción	descrita en el proceso Registro de administrativo

Nombre	informes requeridos de usuarios, acervo, privilegios, credenciales o bitácora
Dónde y cómo se usa	proceso Consulta a la base de datos (salida) entidad Dispositivo de salida (entrada)
Descripción	resultados de las consultas

6. Reinscripción.

Nombre	datos personales del usuario
Dónde y cómo se usa	entidad Usuario (salida) proceso Reinscripción (entrada)
Descripción	ya descritos

Nombre	datos validados del usuario
Dónde y cómo se usa	proceso Reinscripción (salida) almacenamiento Usuario (entrada)
Descripción	ya descrito

7. Registro de administrativo.

Nombre	datos del administrativo
Dónde y cómo se usa	entidad Coordinador de la <i>MEDIATECA</i> (salida) proceso Registro de administrativo (entrada)
Descripción	nombre + cargo + teléfono + contraseña + fecha_acceso + nivel_acceso + notas
Nota	administrativo involucra coordinador, usuario y al propio administrativo

Nombre	nombre
Alias	nombre
Descripción	apellido paterno + apellido materno + nombre (s)
Longitud	40 (carácter)

Nombre	cargo
Alias	cargo
Descripción	(coordinador administrativo usuario)
Longitud	20 (carácter)

Nombre	télefono
Alias	tel
Descripción	número telefónico particular
Longitud	10 (carácter)

Nombre	contraseña
Alias	passw
Descripción	palabra clave
Longitud	8 (carácter)

Nombre	fecha_acceso
Alias	f_acc
Descripción	dd/mm/aaaa
Longitud	8 (fecha)

Nombre	nivel_acceso
Alias	nivel
Descripción	(1.-usuario 2.-operador 3.-coordinador)
Longitud	1 (carácter)

Nombre	notas
Alias	notas
Descripción	observaciones relacionadas con el registrado
Longitud	memo

Nombre	datos validados, identificador y privilegios del administrativo
Dónde y cómo se usa	proceso Registro de administrativo (salida) almacenamiento Privilegios (entrada)
Descripción	datos del administrativo + identificador administrativo + nivel_acceso

Nombre	identificador_administrativo
Alias	idadm
Descripción	número consecutivo
Longitud	2 (carácter)

Nombre	Privilegios
Alias	Adm
Dónde y cómo se usa	datos Datos validados, contraseña y privilegios del administrativos (salida)
Descripción	tabla de almacenamiento

8. Consulta al acervo.

Nombre	requerimientos de la consulta
Alias	requerimientos de consulta
Dónde y cómo se usa	entidad Usuario (salida) proceso Consulta al acervo (entrada)
Descripción	requerimientos de consulta a la tabla acervo

Nombre	datos del acervo
Alias	datos validados e identificador del acervo
Dónde y cómo se usa	almacenamiento Acervo (salida) proceso Consulta al acervo (entrada)
Descripción	de los datos del acervo ya descritos se toman los correspondientes a los requerimientos de la consulta

Nombre	informe requerido del acervo
Dónde y cómo se usa	proceso Consulta al acervo (salida) entidad Dispositivo de salida (entrada)
Descripción	resultados de los requerimientos de la consulta

3.3 MINI-ESPECIFICACIONES DEL SIIMCELE.

Atendiendo a los DFD's. Nivel 2 de los procesos que conforman al SIIMCELE y a las Tablas de Decisiones de los mismos procesos, surgen las miniespecificaciones correspondientes. Las mini-especificaciones están indicadas a partir de la Fig. 3.11 y hasta la Fig. 3.18.

1. Inscripción.

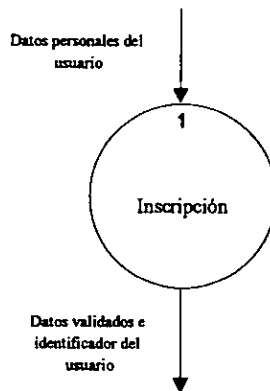


FIG. 3.11

PROCEDIMIENTO inscripción

SI datos personales del usuario disponibles
 ENTONCES permitir capturar datos personales del usuario
 validar datos personales del usuario
 establecer identificador de usuario
 registrar datos personales e identificador de usuario
 DE OTRA MANERA esperar datos personales del usuario
 FIN-SI

FIN-PROCEDIMIENTO

2. Control de acceso.

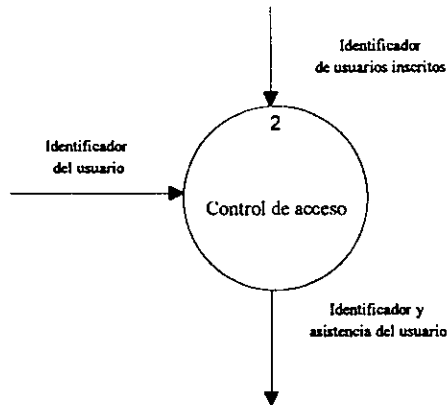


FIG. 3.12

PROCEDIMIENTO control de acceso

SI identificador del usuario disponible

ENTONCES leer identificador del usuario

comparar identificador del usuario con identificador de usuario inscrito

SI identificador del usuario = identificador de usuario inscrito

ENTONCES acceso permitido

registrar identificador y asistencia al usuario

DE OTRA MANERA acceso denegado

FIN-SI

DE OTRA MANERA esperar identificador del usuario

FIN-SI

FIN-PROCEDIMIENTO

3. Llenado de bitácora.

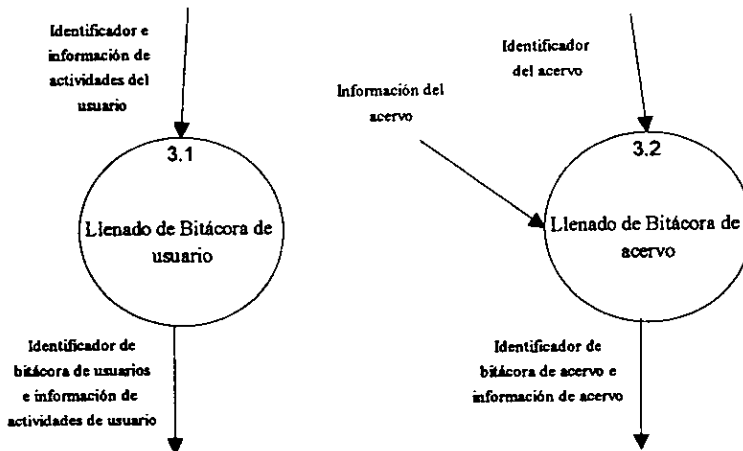


FIG. 3.13

PROCEDIMIENTO llenado de bitácora de usuario

SI identificador e información de actividades del usuario disponibles

ENTONCES leer el identificador del usuario

permitir capturar información de actividades del usuario

establecer identificador de bitácora de usuario

registrar información de actividades del usuario e identificador de bitácora de usuario

DE OTRA MANERA no procede llenado de bitácora de usuario

FIN-SI

FIN-PROCEDIMIENTO

PROCEDIMIENTO llenado de bitácora de acervo

SI identificador e información del acervo disponibles

ENTONCES leer el identificador del acervo

permitir capturar información del acervo

establecer identificador de bitácora de acervo

registrar identificador de bitácora de acervo e información del acervo

DE OTRA MANERA no procede llenado de bitácora de acervo

FIN-SI

FIN-PROCEDIMIENTO

4. Registro de acervo.



FIG. 3.14

PROCEDIMIENTO registro de acervo**SI datos del acervo disponibles****ENTONCES permitir capturar datos del acervo****validar datos del acervo****establecer identificador de acervo****registrar datos e identificador del acervo****DE OTRA MANERA esperar datos del acervo****FIN-SI****FIN-PROCEDIMIENTO**

5. Consulta a la base de datos.

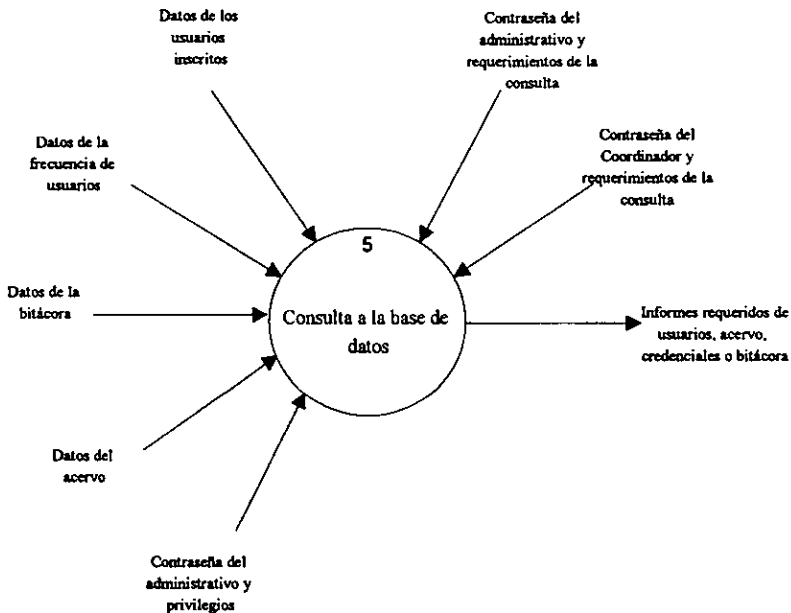


FIG. 3.15

PROCEDIMIENTO consulta a la base de datos

SI contraseña del administrativo y requerimientos de consulta disponibles

ENTONCES leer contraseña y requerimientos del administrativo

leer privilegios del administrativo

SI privilegios válidos

ENTONCES atender requerimientos de consulta pudiendo leer y/o

escribir en las tablas: usuarios inscritos, frecuencia de usuarios,

bitácora, acervo

canalizar el informe obtenido al dispositivo de salida

DE OTRA MANERA no procede consulta

DE OTRA MANERA continuar

SI contraseña del coordinador y requerimientos de consulta disponibles

ENTONCES leer contraseña y requerimientos del coordinador

leer privilegios del coordinador

SI privilegios válidos

ENTONCES atender requerimientos de consulta pudiendo leer y/o

escribir en las tablas: usuarios, frecuencia de usuarios, bitácora,

acervo, privilegios de los administrativos

canalizar el informe obtenido al dispositivo de salida

DE OTRA MANERA no procede consulta

DE OTRA MANERA esperar contraseña y requerimientos de administrativo o de coordinador

FIN-PROCEDIMIENTO

6. Reinscripción.

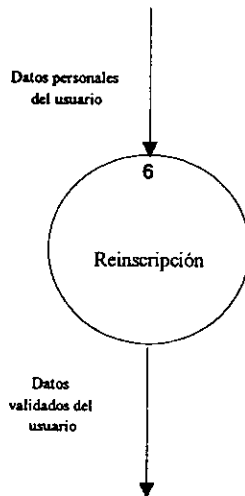


FIG. 3.16

PROCEDIMIENTO reinscripción

SI datos personales del usuario disponibles

ENTONCES leer status del usuario

SI status = no inscrito

ENTONCES validar datos personales del usuario

DE OTRA MANERA no procede reinscripción

FIN-SI

DE OTRA MANERA esperar datos personales del usuario

FIN-SI

FIN-PROCEDIMIENTO

7. Registro de administrativo.

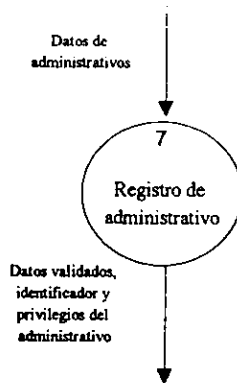


FIG. 3.17

PROCEDIMIENTO registro de administrativo**SI datos de administrativos disponibles****ENTONCES permitir captura de datos de administrativos****validar datos de administrativos****permitir establecer privilegios de administrativos****establecer identificador de administrativos****registrar datos validados, identificador y privilegios de administrativo****DE OTRA MANERA esperar datos de administrativos****FIN-SI****FIN-PROCEDIMIENTO**

8. Consulta al acervo.

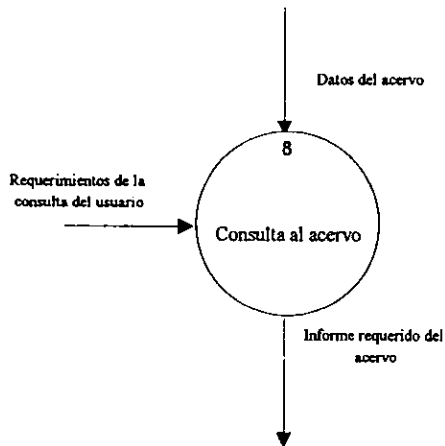


FIG. 3.18

PROCEDIMIENTO consulta al acervo

SI requerimientos de consulta al acervo del usuario disponibles

ENTONCES requerimientos del usuario

atender requerimientos de consulta leyendo los datos del acervo

canalizar el informe obtenido al dispositivo de salida

DE OTRA MANERA esperar requerimientos de consulta al acervo del usuario

FIN-SI

FIN-PROCEDIMIENTO

3.4 TABLAS DE DECISIÓN DEL SIIMCELE.

En esta sección se muestran las tablas de decisión (Tablas 3.1-3.11) asociadas a cada proceso que realiza el SIIMCELE y que sirven de apoyo en la elaboración de las mini-especificaciones.

1. Inscripción.

	1	2
Datos personales del usuario disponibles	T	F
Permitir capturar datos personales del usuario	X	
Validar datos personales del usuario	X	
Establecer identificador de usuario	X	
Registrar datos personales del usuario	X	
Esperar datos personales del usuario		X

TABLA 3.1

2. Control de acceso.

Tabla A

	1	2
Identificador del usuario disponible	T	F
Leer identificador del usuario	X	
Ejecutar tabla B	X	
Esperar identificador del usuario		X

TABLA 3.2

Tabla B

	1	2
Identificador del usuario = identificador de usuario inscrito	T	F
Acceso permitido	X	
Registrar identificador y asistencia al usuario	X	
Acceso denegado		X
Regresar a la tabla A		X

TABLA 3.3

3. Llenado de bitácora.

3.1 Llenado de bitácora de usuario.

	1	2
Identificador e información de actividades del usuario disponibles	T	F
Leer identificador del usuario	X	
Permitir capturar información de las actividades del usuario	X	
Establecer identificador de la bitácora del usuario	X	
Registrar información de las actividades del usuario e identificador de la bitácora del usuario	X	
No procede llenado de la bitácora del usuario		X

TABLA 3.4

3.2 Llenado de bitácora de acervo.

	1	2
Identificador e información del acervo disponibles	T	F
Leer identificador del acervo	X	
Permitir capturar información del acervo	X	
Establecer identificador de bitácora del acervo	X	
Registrar información del acervo e identificador de la bitácora del acervo	X	
No procede llenado de la bitácora del acervo		X

TABLA 3.5

4. Registro del acervo.

	1	2
Datos del acervo disponibles	T	F
Permitir capturar datos del acervo	X	
Validar datos del acervo	X	
Establecer identificador del acervo	X	
Registrar datos e identificador del acervo	X	
Esperar datos del acervo		X

TABLA 3.6

5. Consulta a la base de datos.

	1	2	3
Contraseña del administrativo y requerimientos de consulta disponibles	T		F
Contraseña del coordinador y requerimientos de consulta disponibles		T	F
Leer contraseña y requerimientos del administrativo	X		
Leer privilegios del administrativo	X		
Atender requerimientos de consulta pudiendo leer y/o escribir en datos de los usuarios inscritos, datos de la frecuencia de usuarios, datos de la bitácora, datos del acervo	X		
Canalizar el informe obtenido al dispositivo de salida	X		
Leer contraseña y requerimientos del coordinador		X	
Leer privilegios del coordinador		X	
Atender requerimientos de consulta pudiendo leer y/o escribir en los datos de los usuarios inscritos, datos de la frecuencia de usuarios, datos de la bitácora, datos del acervo, datos de los privilegios de administrativos		X	
Canalizar el informe obtenido al dispositivo de salida		X	
Esperar contraseña y requerimientos de consulta del usuario, del administrativo o del coordinador			X

TABLA 3.7

6. Reinscripción.

Tabla A

	1	2
Datos personales del usuario disponibles	T	F
Leer status del usuario	X	
Ejecutar tabla B	X	
Esperar datos personales del usuario		X

TABLA 3.8

Tabla B

	1	2
Status = no inscrito	T	F
Validar datos personales del usuario	X	
No procede reinscripción		X
Regresar a la tabla A		X

TABLA 3.9

7. Registro de administrativo.

	1	2
Datos de administrativos disponibles	T	F
Permitir captura de datos del administrativo	X	
Validar datos de administrativos	X	
Permitir establecer privilegios de administrativos	X	
Establecer identificador del administrativo	X	
Establecer contraseña del administrativo	X	
Registrar datos validados, privilegios e identificador del administrativo	X	
Esperar datos del administrativo		X

TABLA 3.10

8. *Consulta al acervo.*

	1	2
Requerimientos de consulta al acervo del usuario disponibles	T	F
Leer identificador y requerimientos del usuario	X	
Atender requerimientos de consulta leyendo los datos del acervo	X	
Canalizar el informe obtenido al dispositivo de salida	X	
Esperar requerimientos de consulta al acervo del usuario		X

TABLA 3.11

4. DISEÑO ESTRUCTURADO DEL SIIMCELE.

4.1 LA CDE DEL SIIMCELE.

La CDE del SIIMCELE, es el diagrama que consta de rectángulos numerados que representan los módulos del sistema; los rectángulos sin números representan las correspondientes funciones que ejecutan esos módulos y las líneas de conexión señalan un sistema modular descendente, como se observa en la Fig. 4.1.

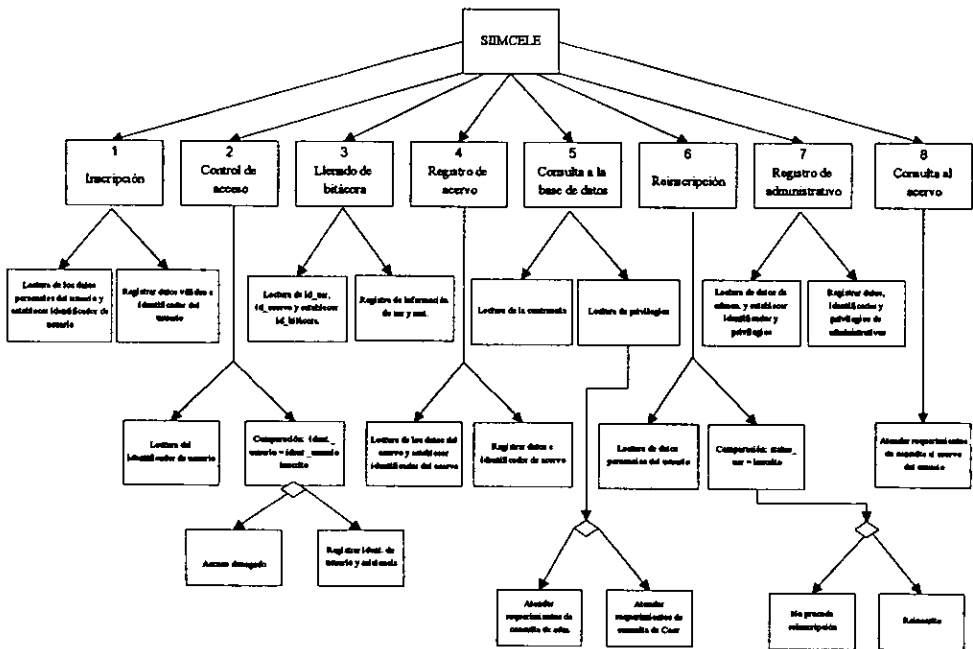


FIG. 4.1

Cada módulo representado en la CDE puede ser dibujado con mayor detalle con objeto de señalar los datos que se están manejando (Figuras 4.2 a 4.9):

1. Inscripción.

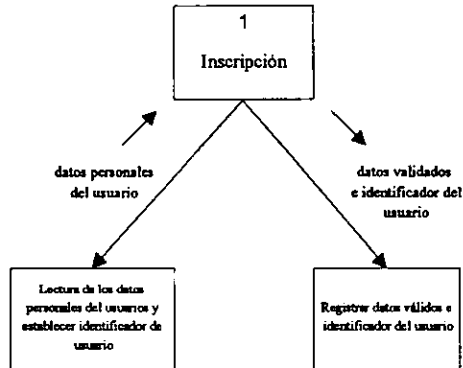


FIG. 4.2

2.- Control de acceso.

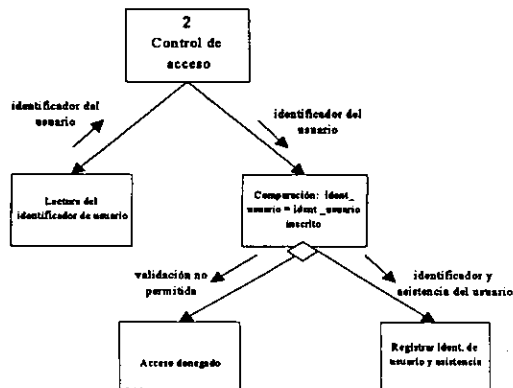


FIG. 4.3

3. Llenado de bitácora.

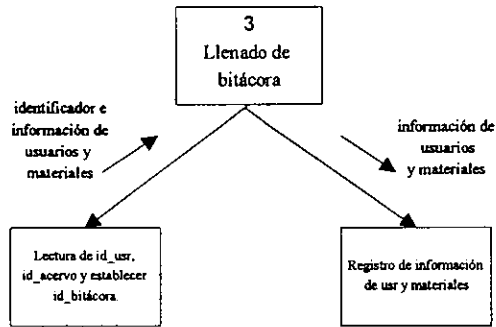


FIG. 4.4

4. Registro de acervo.

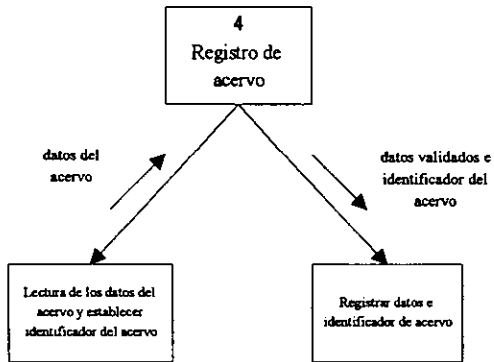


FIG. 4.5

5. Consulta a la base de datos.

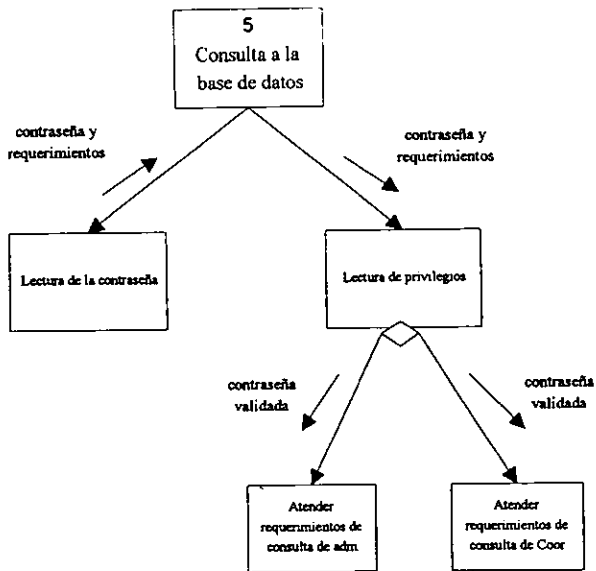


FIG. 4.6

6. Reinscripción.

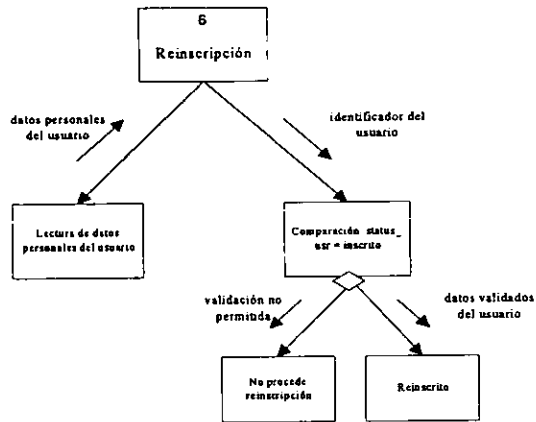


FIG. 4.7

7. Registro de administrativo.

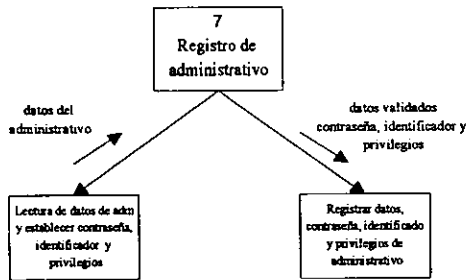


FIG. 4.8

8. Consulta al acervo.

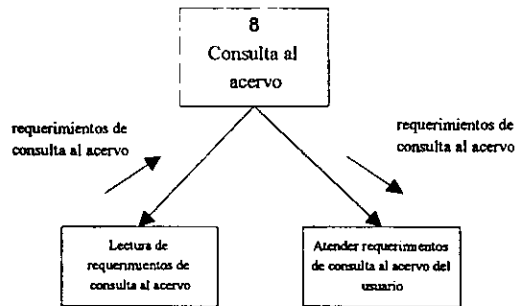


FIG. 4.9

4.2 COHESIÓN DEL SIIMCELE.

Para determinar la cohesión del SIIMCELE, nos apoyamos en los razonamientos de Constantine (explicados anteriormente), los cuales son aplicados a una proposición sobre la función de cada módulo:

MÓDULO	FUNCIÓN
1. <i>Inscripción.</i>	Leer los datos personales del usuario para registrarlos.
2. <i>Control de acceso.</i>	Leer el identificador del usuario para registrar su asistencia, en el caso en que la inscripción del usuario esté vigente.
3. <i>Llenado de bitácora.</i>	Leer el identificador del usuario y del acervo para registrar la información de las actividades del usuario y del material que utilizó.
4. <i>Registro de acervo.</i>	Leer los datos del acervo para registrarlos.
5. <i>Consulta a la base de datos.</i>	Leer la contraseña y privilegios del administrativo y/o coordinador para la atención a sus requerimientos de consulta

	a la base de datos del SIIMCELE.
6. <i>Reinscripción.</i>	Leer la vigencia de la inscripción del usuario para registrar sus datos por un nuevo período, si la reinscripción procede.
7. <i>Registro de administrativo.</i>	Leer los datos personales, contraseña, identificador y privilegios de los administrativos para registrarlos.
8. <i>Consulta al acervo.</i>	Atender los requerimientos de consulta inmediata al acervo por parte de los usuarios.

TABLA 4.1

Como se observa, las proposiciones expresan que los elementos de cada módulo se encuentran relacionados al desempeño de una sola función, esto implica una *cohesión funcional* de cada uno de ellos.

4.3 ACOPLAMIENTO DEL SIIMCELE.

Los módulos del SIIMCELE presentan *acoplamiento por zona de datos*, cada módulo tiene acceso a zonas previamente determinadas de la única base de datos existente y dichos módulos modifican los datos de acuerdo a una función establecida en particular. Identificando a cada zona como las tablas que constituyen nuestra base de datos, observamos la siguiente distribución de acoplamientos:

MÓDULOS	ZONA DE DATOS
<i>Inscripción, Control de acceso y Reinscripción.</i>	Tabla Usuarios y Tabla Frecuencia de usuarios.
<i>Bitácoras y Consulta a la base de datos.</i>	Tablas de Bitácoras, Tabla Usuarios y Tabla Acervo.
<i>Registro de acervo y Consulta al acervo.</i>	Tabla Acervo.
<i>Registro de administrativo.</i>	Tabla Privilegios.

5. BASE DE DATOS DEL SIIMCELE.

5.1 DIAGRAMA ENTIDAD-RELACIÓN EN LA ESTRUCTURA DE DATOS RELACIONALES DEL SIIMCELE.

Con objeto de establecer el diagrama Entidad-Relación del *SIIMCELE*, se toman en cuenta las siguientes tablas bidimensionales, entendiéndose como tales a las fuentes de datos que se establecen en las estructuras de datos relacionales:

1. Usuarios.
2. Frecuencia de usuarios.
3. Bitácora de usuarios y Bitácora de acervo.
4. Acervo.
5. Privilegios.

Estas tablas están complementadas con catálogos que facilitan la captura de los atributos correspondientes a cada una de ellas. Los catálogos son tablas con dos atributos y un número variado de registros, que establecen con una clave (identificador) y una descripción, el atributo de una tabla o entidad; por ejemplo: para el atributo idioma de la tabla Usuario existe el catálogo con el mismo nombre:

IDIOMAS	
Identificador	Idioma
01	Inglés
02	Francés
03	Alemán
04	Italiano

FIG. 5.1

En el *SIIMCELE* existen los siguientes catálogos:

- Acentos.
- Acervo.
- Autoevaluación.
- Categoría.
- Elección del acervo.
- Idiomas.
- Objetivos.
- Orientación.
- Problemas con el acervo.
- Problemas del usuario.
- Procedencia.
- Utilidad.

Los atributos de cada tabla y de cada catálogo de alguna manera fueron descritos en el diccionario de datos, sin embargo, en esta sección los enlistaremos, señalando el atributo llave primaria con un subrayado continuo doble y el atributo llave externa (un atributo que no es llave primaria en una tabla pero si lo es en otra) con un subrayado continuo sencillo.

• *Lista de tablas y correspondientes atributos.*

USUARIOS	<u>identificador_usuario</u> , nombre, número_cuenta-rc, <u>identificador_categoria</u> , <u>identificador_idioma</u> , número_inscripciones, fecha_registro, <u>identificador_procedencia</u> , fecha_inscripción, fecha_nacimiento, sexo, calle, colonia, delegación-municipio, código-postal, teléfono, lengua1, semestre_lengua1, lengua2, semestre_lengua2, observaciones, inscrito, no_inscripciones)
BITÁCORA DE USUARIOS	<u>identificador_bitácora_usuario</u> , <u>usuario</u> , <u>identificador_utilidad_orientación</u> , <u>identificador_problemas_usuario</u> , <u>identificador_objetivo</u> <u>identificador_orientación</u> , <u>identificador_autoevaluación</u> , comentarios
BITÁCORA DE ACERVO	<u>identificador_bitácora_acervo</u> , <u>identificador_acervo</u> , <u>identificador_elección_acervo</u> , <u>identificador_utilidad_acervo</u> , <u>identificador_problemas_acervo</u>
ACERVO	<u>identificador_acervo</u> , isbn, issn, <u>identificador_tipo_acervo</u> , <u>identificador_idioma</u> , clasificación, autor-personal, autor-corporativo, título, edición, pie_imprenta, descripción-física, serie, notas-generales, fecha_ingreso, descriptores, consulta fondo_origen, resumen, <u>identificador_acento</u> , referencia, nivel_dificultad, número_ejemplares, catalogador
PRIVILEGIOS	<u>identificador_administrativo</u> , nombre, cargo, teléfono, contraseña, fecha_acceso, nivel_acceso, notas

- *Lista de Catálogos y correspondientes atributos.*

acentos	<u>identificador acento</u> , descripción
acervo	<u>identificador tipo acervo</u> , descripción
autoevaluación	<u>identificador autoevaluación</u> , descripción
categoría	<u>identificador categoría</u> , descripción
elección de acervo	<u>identificador elección acervo</u> , descripción
idiomas	<u>identificador idioma</u> , descripción
objetivos	<u>identificador objetivo</u> , descripción
orientación	<u>identificador orientación</u> , descripción
problemas con el acervo	<u>identificador problemas acervo</u> , descripción
problemas con el usuario	<u>identificador problemas usuario</u> , descripción
procedencia	<u>identificador procedencia</u> , descripción
utilidad del acervo	<u>identificador utilidad acervo</u> , descripción

DIAGRAMA ENTIDAD-RELACION.

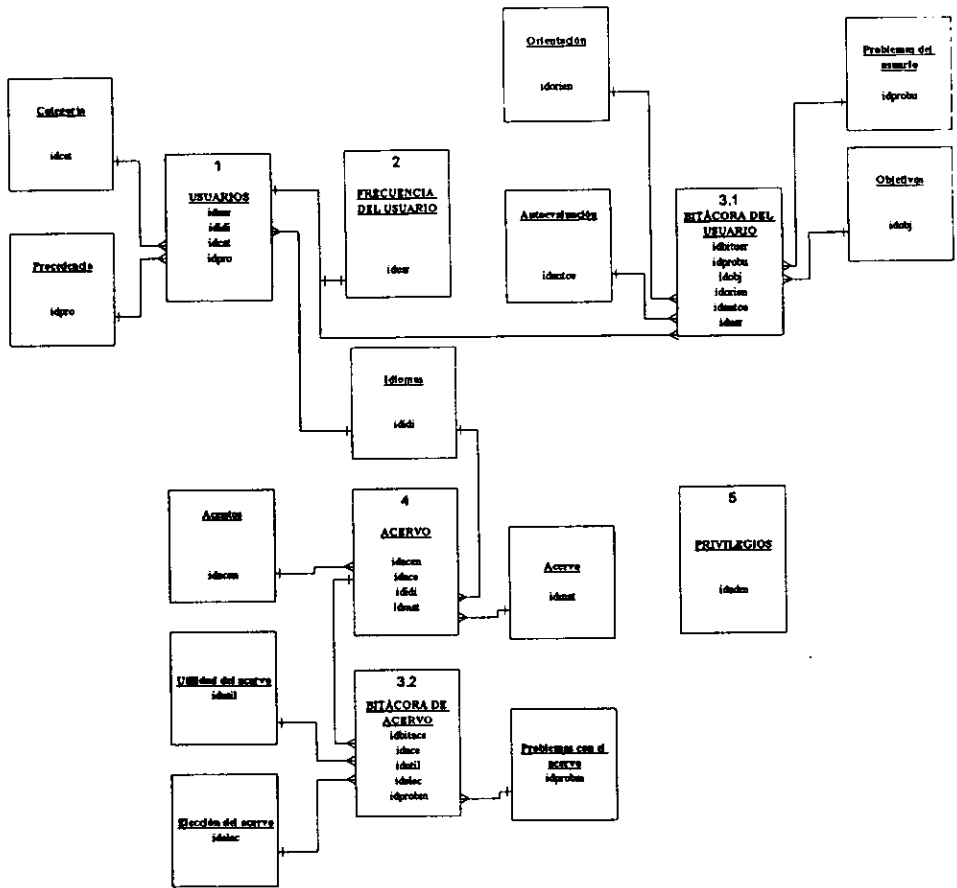


FIG. 5.2

5.2 NORMALIZACIÓN EN EL SIIMCELE.

El diagrama Entidad-Relación involucra las tablas de almacenamiento necesarias para el desempeño del SIIMCELE. Por otro lado, es adecuado señalar que el establecimiento de la tabla de la Bitácora y la de Usuarios surgió de la normalización de las tablas que se utilizaban en la MEDiateca.

BITÁCORA (tabla sin normalizar).

Atributos	Ejemplo
nombre	Soto Quintana Mauricio
fecha_registro	24 / Jul /1997
materiales_empleados	videocassettes
	libro
	CD-ROM
elección_material	azar
	recomendación
	lo ví en el estante
¿Utilidad_material?	muy útil
	poco útil
	útil
¿Problemas con los materiales?	no
	si
	si
¿Cuáles problemas con los materiales?	ninguno
	insuficientes ayudas
	deficientes recomendaciones
¿Problemas en general?	problemas técnicos
¿Realizaste autoevaluación?	si
¿Cuál autoevaluación?	con un amigo
¿Objetivos para esta sesión?	si
¿Lograste los objetivos?	no
¿Si no los lograste cuál es el motivo?	faltó tiempo
¿Asesoría?	si
¿Cómo resultó la asesoría?	muy útil
comentarios	

TABLA 5.1

Forma Normal 1.

1. Se eliminan los grupos repetidos: materiales empleados, elección_material, ¿Utilidad_material?, ¿Problemas con los materiales?, ¿Cuáles problemas?
- 2.- Debido a la existencia de grupos repetidos, la tabla Bitácora deberá descomponerse en dos tablas:

Bitácora de acervo.

Atributos	Ejemplo de datos	Ejemplo de datos	Ejemplo de datos
fecha de registro	24 / Jul / 1997		
material empleado	videocassettes	libro	CD-ROM
elección_material	azar	recomendación	lo ví en el estante
¿Utilidad_material?	muy útil	poco útil	útil
¿Problemas con los materiales?	no	si	si
¿Cuáles problemas con los materiales?	ninguno	insuficientes ayudas	deficientes recomendaciones

TABLA 5.2

Bitácora de usuarios.

Atributos	Ejemplos
Nombre	Soto Quintana Mauricio
fecha_registro	24 / Jul / 1997
¿Problemas en general?	problemas técnicos
¿Realizaste autoevaluación?	Si
¿Cuál autoevaluación?	con un amigo
¿Objetivos para esta sesión?	Si
¿Lograste los objetivos?	No
¿Si no los lograste cuál es el motivo?	faltó tiempo
¿Asesoría?	Si
¿Cómo resultó la asesoría?	muy útil
comentarios	

TABLA 5.3

Forma Normal 2.

Las dos tablas de Bitácoras ya se encuentran en FN2: todos los atributos son dependientes del criterio primario (nombre). Sin embargo, ciertos atributos son dependientes de otros atributos, por lo que se hace necesario emplear la FN3.

Forma Normal 3.

En la FN3 se establece que todos los atributos sean dependientes del criterio primario y no existan dependencias entre esos atributos, como es el caso de:

Bitácora de acervo.

¿Cuáles problemas con los materiales?	depende de	¿Problemas con los materiales?
---------------------------------------	------------	--------------------------------

TABLA 5.4

Bitácora de usuarios.

¿Cuál autoevaluación?	depende de	¿Realizaste autoevaluación?
¿Lograste los objetivos?	depende de	¿Objetivos para esta sesión?
¿Si no lograste los objetivos, cuál es el motivo?	depende de	¿Lograste los objetivos?
¿Cómo resultó la asesoría?	depende de	¿Asesoría?

TABLA 5.5

Las tablas de Bitácora normalizadas tendrían entonces los siguientes atributos con nombres más adecuados y además, la palabra "materiales" será sustituida por la de "acervo".

Bitácora de acervo. Se estableció la llave primaria: `identificador_bitácora_acervo` (considerándola como la fecha y hora de llenado de la bitácora de acervo).

Atributo anterior	Atributo actual
	<code>identificador_bitácora_acervo</code>
materiales empleados	<code>identificador_acervo</code>
elección material	<code>identificador_elección_acervo</code>
¿Utilidad material?	<code>identificador_utilidad_acervo</code>
¿Problemas con los materiales?	<code>identificador_problemas_acervo</code>

TABLA 5.6

Bitácora de usuarios. Se agregó la llave primaria: `identificador_bitácora` (considerándola como la fecha y hora del llenado de la bitácora de usuario) y el atributo: `identificador_orientación`.

Atributo anterior	Atributo actual
	<code>identificador_bitácora</code>
nombre	<code>identificador_usuario</code>
fecha registro	<code>fecha_registro</code>
¿Cómo resultó la asesoría?	<code>identificador_utilidad_orientación</code>
¿Problemas en general?	<code>identificador_problemas_usuario</code>
¿Si no lograste los objetivos cuál es el motivo?	<code>identificador_objetivo</code>
	<code>identificador_orientación</code>
¿Cuál autoevaluación?	<code>identificador_autoevaluación</code>
comentarios	<code>comentarios</code>

TABLA 5.7

USUARIOS (tabla sin normalizar).

Atributos	Ejemplo 1 de datos	Ejemplo 2 de datos
nombre	Soto Quintana Mauricio	Coutiño Sánchez Ricardo
fecha_inscripción	24 / Jul / 1997	10 / Ago / 1997
estudiante	si	
empleado		si
idioma	inglés	francés
facultad	Ingeniería	
instituto		Inst. de Biología
escuela		
dependencia		
número_cuenta	87256936	
rfc		cosr300172
fecha_nacimiento	22 / Sep / 1972	30 / Ene / 1972
sexo	m	m
calle	Rio Bravo	Orizaba
número_calle	17	30
colonia	Antares	Valle lunar
delegación / municipio	Iztapalapa	Atizapán
código-postal	09770	07750
teléfono	5688722	6581111
¿Estudia en el CELE?	no	si
lengua1		alemán
semestre-lengua1		3
lengua2		
semestre-lengua2		
observaciones	usuario frecuente	
status	inscrito	no inscrito

TABLA 5.8

Forma Normal 1.

La tabla ya se encuentra en esta forma, porque no existen grupos de datos repetidos.

Forma Normal 2.

Todos los atributos son dependientes de la llave primaria nombre, por lo tanto, también la tabla ya se encuentra en esta forma.

Forma Normal 3.

Se elimina la dependencia entre atributos:

Dependencia entre	Nombre actual del atributo sustituto
estudiante empleado	identificador_categoria
número_cuenta rfc	número_cuenta-rfc
facultad instituto escuela dependencia	identificador_procedencia
calle número_calle	calle
¿Estudia en el CELE? lengua1 semestre-lengua1	lengua1-semester
¿Estudia en el CELE? lengua2 semestre-lengua2	lengua2-semester

TABLA 5.9

La tabla de **Usuarios** normalizada tendría entonces los siguientes atributos con nombres más adecuados (se estableció la llave primaria `identificador_usuario` y se agregaron los atributos `número_inscripciones` y `fecha_registro`):

Atributo anterior	Atributo actual
	<code>identificador_usuario</code>
<code>nombre</code>	<code>nombre</code>
<code>número_cuenta/rfc</code>	<code>número_cuenta-rfc</code>
<code>estudiante/empleado</code>	<code>identificador_categoria</code>
<code>idioma</code>	<code>identificador_idioma</code>
	<code>número_inscripciones</code>
	<code>fecha_registro</code>
<code>facultad/instituto/escuela/dependencia</code>	<code>identificador_procedencia</code>
<code>fecha_inscripción</code>	<code>fecha_inscripción</code>
<code>fecha_nacimiento</code>	<code>fecha_nacimiento</code>
<code>sexo</code>	<code>sexo</code>
<code>calle/número_calle</code>	<code>calle</code>
<code>colonia</code>	<code>colonia</code>

delegación/municipio	delegación-municipio
código-postal	código-postal
teléfono	teléfono
¿Estudia en el CELE?/lengua1/semestre-lengua1	lengua1-semester
¿Estudia en el CELE?/lengua2/semestre-lengua2	lengua2-semester
observaciones	observaciones
status	status

TABLA 5.10

6. LENGUAJE DE PROGRAMACIÓN EMPLEADO PARA DESARROLLAR EL *SIIMCELE*.

Uno de los requisitos establecidos para el desarrollo del *SIIMCELE* es emplear Visual FoxPro como herramienta de desarrollo, por lo tanto, en este capítulo se señalan las características sobresalientes de este lenguaje de programación y manejador de bases de datos.

Visual FoxPro es una herramienta de programación de cuarta generación, como tal, permite la generación automática de código, facilitando el desarrollo de aplicaciones que emplean bases de datos.

Visual FoxPro presenta las siguientes características técnicas y de ingeniería:

1. Facilidad de traducción del diseño al código.
2. Compilación eficiente.
3. Portabilidad del código fuente.
4. Disponibilidad de elementos que apoyan el desarrollo de aplicaciones, tales como ayudas, facilidades de edición, correctores, etc.
5. Facilidades de mantenimiento, al presentar código sencillo de entender y ser modificado según se requiera en los cambios de diseño que pudieran presentarse.
6. Tipificación de datos.
7. Estructuras de control.
8. Acomodamiento de tablas. Visual FoxPro permite diseñar tablas que incluyan todos los campos pertinentes y además, modificar la estructura de éstas, incluyendo campos adicionales.
9. Importación de datos. Este DBMS (DataBase Management System) importa texto alineando los datos en los campos apropiados.
10. Crear formatos (pantallas de captura de datos). Aprovechando la interfaz gráfica de Windows, la aplicación de base de datos construida dará a los usuarios la

habilidad de introducir datos de la manera más fácil, amigable, interactiva y eficiente.

11. Construcción de Querys. Visual FoxPro ofrece gran velocidad de acceso a los datos, que es una característica importante en el desempeño de querys.
12. Crear Reportes. El DBMS no sólo accesa a los datos sino que permite manejarlos en reportes útiles, que pueden ser diseñados y ajustados para ver exactamente lo que se desea.
13. El programa incluye herramientas visuales de programación y controladores para el procesamiento cliente/servidor.
14. Visual FoxPro contiene los objetos necesarios para trazar la programación de aplicaciones de bases de datos. El lenguaje (XBase), y las herramientas de programación son totalmente visuales. Las ventajas de estas características son: código de programación mínimo y óptimo, tiempo de aprendizaje e implementación cortos (con respecto a otros lenguajes de tercera generación como C, Pascal, Clipper, etc.).
15. Para disminuir el tiempo de la codificación, se crean objetos del tipo arrastrar y soltar. Se pueden modificar estos objetos utilizando herramientas visuales.
16. El Administrador de proyectos es el "depósito" principal de FoxPro para todas las tablas de datos y objetos relacionados (formas, reportes, programas, clases de objetos, etc.). Para generar un programa ejecutable de un proyecto, es suficiente con un click en el botón Crear.
17. La base de datos soporta la conectividad de bases de datos abierta (ODBC), el estándar para ganar acceso a bases de datos más grandes. Ofrece soporte directo para bases de datos de empresas como Oracle y DB2 de IBM.
18. En cuanto a los requerimientos de hardware; Visual FoxPro se desempeña adecuadamente en equipos PC-486 y superiores, en cuanto a la cantidad de memoria de su PC se podrá obtener un mayor rendimiento de Visual FoxPro en memoria RAM si:
 - No se ejecuta ninguna otra aplicación Windows mientras Visual FoxPro esté en ejecución.

- No se utiliza un administrador de memoria expandida.
- Se utilizan sólo los programas residentes en memoria necesarios para el funcionamiento.
- Se utiliza un caché de disco no mayor de 1048 KB.

Para obtener el rendimiento óptimo en un equipo de 8 MB con RAM:

- Utilizar un archivo de intercambio permanente de 8 MB para la memoria virtual.
- Crear un caché pequeño (alrededor de 128 KB).

Para obtener un óptimo rendimiento en un equipo con 12 MB o más de RAM:

- No utilizar un archivo de intercambio para la memoria virtual y establecer la memoria virtual como "Ninguna".

Visual FoxPro presenta además, las siguientes ventajas:

- Empleo de XBase.
- Asistentes y generadores.
- Diseñadores visuales.
- Interfaces rápidas de aplicaciones.
- Programación orientada a objetos.
- Eventos.
- Tecnología Rushmore.
- Barras de herramientas.
- Uso compartido de datos con otras aplicaciones.
- Automatización OLE.
- Importación y exportación de datos.
- Diccionario de datos.
- Vistas remotas y heterogéneas.
- Transacciones.

- Soporte cliente-servidor.
- Distribución de aplicaciones.
- Ayudas.
- Herramientas y ejemplos adicionales.

6.1 VISUAL FOXPRO Y OTROS LENGUAJES DE PROGRAMACIÓN.

La mayoría de los lenguajes de programación tienen características básicas similares. Si se está familiarizado con un lenguaje de programación se puede traducir fácilmente estos conocimientos a otro lenguaje de programación.

Comparación entre Visual FoxPro, BASIC, C/C++ y Pascal:

- Distinción entre mayúsculas y minúsculas.
- Declaración de variables.
- Comentarios.
- Instrucciones de asignación.
- Instrucciones IF.
- Instrucciones CASE.
- Bucles FOR.
- Bucles WHILE.
- Transferencia de parámetros por valor.
- Transferencia de argumentos por referencia.

Distinción entre mayúsculas y minúsculas.

Sólo C/C++ es un lenguaje que distingue entre mayúsculas y minúsculas.

Visual FoxPro	No distingue mayúsculas de minúsculas.
BASIC	No distingue mayúsculas de minúsculas.
Pascal	No distingue mayúsculas de minúsculas.
C/C++	Distingue mayúsculas de minúsculas.

TABLA 6.1

Declaración de variables.

Cuando almacena un valor en una variable y la variable no existe, Visual FoxPro la declara implícitamente. Los lenguajes que requieren una declaración explícita de variables generarán un error.

Visual FoxPro	Las variables se declaran implícitamente.
BASIC	Las variables se pueden declarar implícitamente .
Pascal	Las variables se deben declarar explícitamente.
C/C++	Las variables se deben declarar explícitamente.

TABLA 6.2

Comentarios.

Cada lenguaje tiene su propio formato para agregar comentarios a los programas. Visual FoxPro tiene diferentes formatos para comentarios en una sola línea y para comentarios en líneas parciales.

Visual FoxPro	* línea completa USE && fin de línea
BASIC	' comentario
Pascal	{ comentario }
C/C++	// comentario

TABLA 6.3

Instrucciones de asignación.

Pascal requiere dos puntos en la instrucción de asignación.

Visual FoxPro	nVal = 7
BASIC	nVal = 7
Pascal	nVal := 7
C/C++	nVal = 7

TABLA 6.4

Nota: En Visual FoxPro también se puede asignar valores a campos de una tabla con el comando REPLACE.

Instrucciones IF.

Cada lenguaje ofrece diferentes instrucciones IF.

Visual FoxPro	IF nCnt < nMax nTot = nTot * nCnt nCnt = nCnt + 1 ENDIF
BASIC	If nCnt < nMax Then nTot = nTot * nCnt nCnt = nCnt + 1 End If
Pascal	if nCnt < nMax then begin nTot:=nTot * nCnt; nCnt:=nCnt + 1; end
C/C++	if(nCnt < nMax) { nTot *= nCnt; nCnt++; }

TABLA 6.5

Instrucciones CASE.

Únicamente Pascal no ofrece valores predeterminados en las instrucciones CASE.

Visual FoxPro	DO CASE CASE n = 0 ? 'Zero' CASE n > 0 ? 'Pos' OTHERWISE ? 'Neg' ENDCASE
BASIC	Select Case n Case 0 Print 'Zero' Case Is > 0 Print 'Pos' Case Else Print 'Neg' End Select

Pascal	case n of 0: writeln("Zero"); 1: writeln("One");end
C/C++	switch(n) { case 0: printf("Zero\n"); break; case 1: printf("One\n"); break; default: printf("?n");}

TABLA 6.6

Bucles FOR.

Cada lenguaje ofrece una instrucción FOR; C/C++ es el que ofrece mayor flexibilidad en las expresiones.

Visual FoxPro	FOR n = 1 TO 10 ? n ENDFOR
BASIC	For n = 1 to 10 Print n Next n
Pascal	for n := 1 to 10 do writeln(n);
C/C++	for(n=1; n<11; n++) printf("%d\n",n);

TABLA 6.7

Bucles WHILE.

Cada lenguaje ofrece instrucciones WHILE equivalentes.

Visual FoxPro	DO WHILE n < 100 n = n + n ENDDO
BASIC	While n < 100 n = n + n Wend
Pascal	while n < 100 do n := n + n;
C/C++	while(n < 100) n += n;

TABLA 6.8

Transferencia de parámetros por valor.

Cada lenguaje permite transferir parámetros por valor.

Visual FoxPro	=ABC(X)
BASIC	ABC ByVal X
Pascal	procedure ABC (x:integer);
C/C++	ABC(X);

TABLA 6.9

Transferencia de argumentos por referencia.

En Visual FoxPro puede transferir parámetros por referencia de dos maneras.

Visual FoxPro	=ABC(@X)DO ABC WITH X
BASIC	ABC X
Pascal	procedure ABC var x:integer);
C/C++	ABC(&VAR);

TABLA 6.10

Extensiones de archivo y tipos de archivo.

Esta lista presenta las extensiones y sus tipos de archivos asociados que se usan con Visual FoxPro.

- .ACT Diagrama de acción del Asistente de documentación.
- .APP Aplicación generada.
- .CDX Índice compuesto.
- .DBC Base de datos.
- .DBF Tabla.
- .DCT Memo de base de datos.
- .DCX Índice de base de datos.
- .DLL Biblioteca de vínculos dinámicos de Windows.
- .ERR Error de compilación.
- .EXE Programa ejecutable
- .FKY Macro.
- .FLL Biblioteca de vínculos dinámicos de FoxPro.
- .FMT Formato de archivo.
- .FPT Memo de tabla.
- .FRT Memo de informe.
- .FRX Informe.
- .FXP Programa compilado.
- .HLP Ayuda gráfica.
- .IDX Índice, índice compacto.
- .LBT Memo de etiqueta.
- .LBX Etiqueta.
- .LST Lista del Asistente de documentación.
- .MEM Almacenamiento de variables de memoria.
- .MNT Memo de menú.
- .MNX Menú.

- .MPR Programa de menú generado.
- .MPX Programa de menú compilado.
- .OCX Control OLE.
- .PJT Memo de proyecto.
- .PJX Proyecto.
- .PRG Programa.
- .QPR Programa de consulta generado.
- .QPX Programa de consulta compilado.
- .SCT Memo de formulario.
- .SCX Formulario.
- .SPR Programa de pantalla generado (sólo en versiones anteriores de FoxPro).
- .SPX Programa de pantalla compilado (sólo versiones anteriores de FoxPro).
- .TBK Copia de seguridad de memo.
- .TXT Texto.
- .VCT Memo de biblioteca de clases visuales.
- .VCX Biblioteca de clases visuales.
- .VUE Presentación de FoxPro 2.x.
- .WIN Archivo de ventana.

6.2 CAPACIDADES DE VISUAL FOXPRO.

Archivos de tabla y de índice.

Número máximo de registros por archivo de tabla:	1000 millones
Tamaño máximo de un archivo de tabla:	2 gigabytes
Número máximo de caracteres por registro	65.500
Número máximo de campos por registro:	255
Número máximo de tablas abiertas a la vez:	255 (1)
Número máximo de caracteres por campo de tabla:	254
Número máximo de bytes por clave de índice en un índice no compacto (2):	100
Número máximo de bytes por clave de índice en un índice compacto (2):	240
Número máximo de archivos de índice abiertos por tabla:	sin límite (1)
Número máximo de índices abiertos en todas las áreas de trabajo:	sin límite (1)
Número máximo de relaciones:	sin límite
Longitud máxima de expresiones relacionales:	sin límite

TABLA 6.11

Características de campo.

Tamaño máximo de campos de caracteres:	254
Tamaño máximo de campos numéricos (y float):	20
Número máximo de caracteres en nombres de campos de una tabla libre:	10
Número máximo de caracteres en nombres de campo de una tabla contenida en una base de datos:	128
Valor mínimo de un entero:	-2,147,483,647
Valor máximo de un entero:	2,147,483,647
Dígitos de precisión en cálculos numéricos:	16

TABLA 6.12

Variables de memoria y matrices.

Número predefinido de variables de memoria:	1,024
Número máximo de variables de memoria:	65,000
Número máximo de matrices:	65,000
Número máximo de elementos por matriz:	65,000

TABLA 6.13

Archivos de programa y procedimiento.

Número máximo de líneas en archivos de programa fuente:	sin límite
Tamaño máximo de módulos de programa compilados (3):	64 KB
Número máximo de procedimientos por archivo:	sin límite
Número máximo de llamadas DO anidadas:	128
Número máximo de niveles de anidamiento READ:	5
Número máximo de comandos de programación estructurada anidados:	384
Número máximo de parámetros transferidos:	27
Número máximo de transacciones:	5

TABLA 6.14

Capacidades del generador de informes.

Número máximo de objetos en una definición de informes:	sin límite (1)
Longitud máxima de una definición de informe:	20 pulgadas
Número máximo de niveles de agrupamiento:	128

TABLA 6.15

Otras capacidades diversas.

Número máximo de ventanas abiertas (de todos los tipos):	sin límite (1)
Número máximo de ventanas Examinar abiertas:	255
Número máximo de caracteres por cadena de caracteres:	2 gigabytes
Número máximo de caracteres por línea de comando:	8,192
Número máximo de caracteres por control de etiqueta en un informe:	252
Número máximo de caracteres por línea de subst. de macro:	8,192
Número máximo de archivos abiertos:	por el SO
Número máximo de pulsaciones en macro de teclado:	1,024
Número máximo de campos que se pueden seleccionar mediante una instrucción SQL SELECT:	255

TABLA 6.16

NOTAS:

(1) Limitado por la memoria y los controladores de archivo disponibles. Los archivos .CDX sólo usan un controlador de archivo.

(2) Si la secuencia de ordenación está establecida en **MACHINE**, cada carácter utiliza un byte. Si la secuencia de ordenación no está establecida como **MACHINE**, cada carácter empleará dos bytes. Si el campo indexado admite valores nulos, en la clave de índice se utiliza un byte adicional. Se puede observar que los índices que no utilizan la ordenación **MACHINE** son siempre compactos.

(3) Un módulo de programa es un procedimiento. Un programa o aplicación puede contener un número ilimitado de módulos de programa.

7. MANUAL DE USUARIO DEL SIIMCELE.

El manual del *SIIMCELE* especifica al usuario el manejo de la aplicación, ofreciendo los pasos básicos de ejecución para cada una de las funciones del sistema.

7.1 INTRODUCCIÓN AL SISTEMA DE INFORMACIÓN DE LA *MEDIATECA* DEL CELE (*SIIMCELE*).

En la Base de Datos del *SIIMCELE* se almacenan los datos de Usuarios, Acervo, Bitácoras y Catálogos necesarios para integrar la información requerida por el Sistema.

En este capítulo, se establecerán las instrucciones para el uso del *SIIMCELE*, en ellas se utiliza la palabra formulario para referirse a una pantalla en la computadora que permite tanto la entrada de datos como su visualización. Es oportuno hacer notar que, mientras se está trabajando con el sistema, en la pantalla aparecen pequeñas sugerencias correspondientes a lo que se está realizando y en cualquier momento está disponible la tecla de ayuda <F1>.

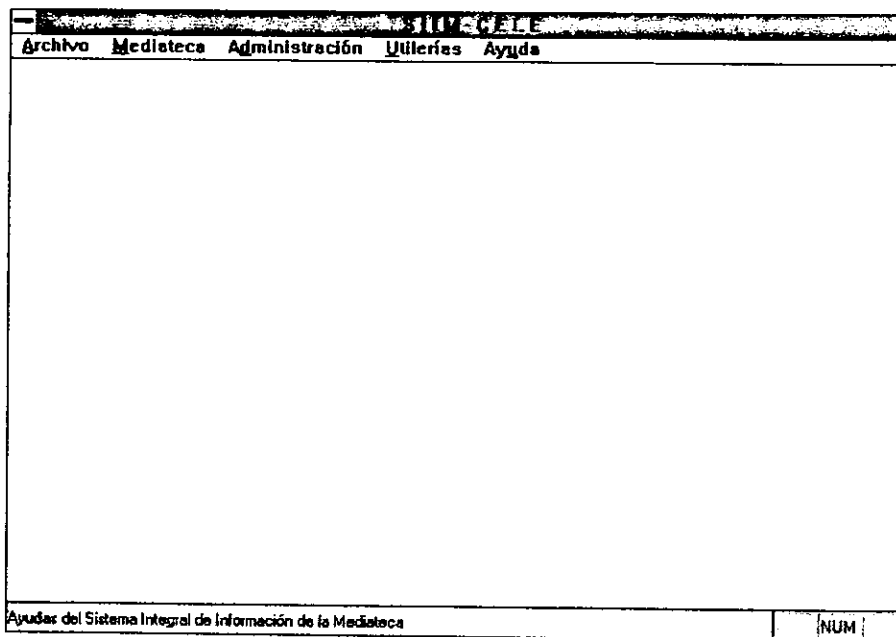
Dependiendo de su nivel de uso, tendrá acceso a distintas funciones del programa, que podrán incluir el ver, quitar, modificar y agregar nuevos registros a la Base de Datos.

Nivel de uso: el nivel de uso quedará identificado como uno de los siguientes:

1. Usuario de los servicios de la *MEDIATECA*.
2. Administrativo de la *MEDIATECA*.
3. Coordinador de la *MEDIATECA*.

7.2 USO DEL SIIMCELE.




















Esta es la pantalla principal del SIIMCELE :



Como se observa, se cuenta con los menús:

- Archivo.
- Mediateca.
- Administración.
- Utilerías.
- Ayuda.

Antes de describir las funciones de cada menú, se explicará la función de los iconos auxiliares en el manejo del *SIIMCELE*.

	Primer registro. Recorre los registros hasta el primero.
	Registro anterior. Recorre un registro hacia atrás.
	Registro siguiente. Recorre un registro hacia adelante.
	Último registro. Recorre los registros hasta el último.
	Buscar registros. Permite buscar registros en particular.
	Imprimir. Imprime la selección.
	Agregar registro. Permite la captura de un registro.
	Modificar registro. Permite hacer cambios en un registro.
	Eliminar registro. Borra un registro.
	Salir del formulario. Termina la sesión con un formulario.
	Aceptar. Valida la captura de datos en un registro.
	Salir. Termina la sesión con un formulario.
	Guardar. Almacena un registro.
	Revertir registro. deshace la captura de un registro.
	Presentación preliminar. Muestra la selección como se verá en impresión.
	Ordenación descendente. Ordena la selección descendentemente.
	Ordenación ascendente. Ordena la selección ascendentemente.
	Dispositivo de salida. Envía la selección hacia pantalla o archivo.
	Tabla. Permite la selección de una tabla de datos.

MENÚ	FUNCIONES
Archivo.	Registro. Salir.

Registro. El formulario Registro permite la entrada al *SIIMCELE*, habilitando las opciones del sistema dependiendo del *nivel de usuario*.

Datos Personales
 Nombre: LIC. MARINA CHAVEZ
 Contraseña:
 Cargo: COORDINADORA
 Organismo: COMANDO EN JEFE FUERZAS POLICIALES DE LA UNIDAD
 Aceptar Cancelar

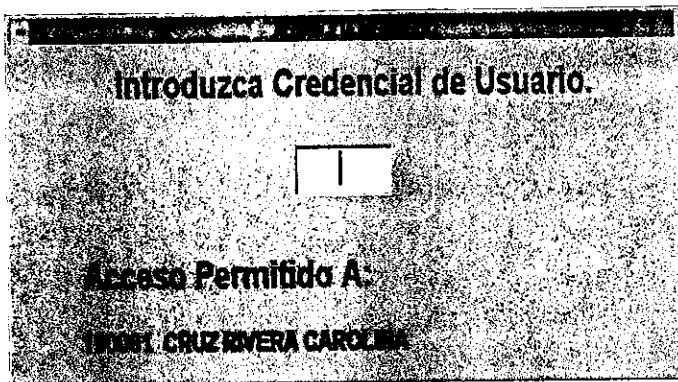
Cómo registrarse.

1. En el menú **Archivo**, elija **Registro** (al inicio de una sesión con el *SIIMCELE*, aparece automáticamente este formulario; en sesiones posteriores, será necesario seguir este paso).
2. En el cuadro "Nombre", escriba su nombre o selecciónelo de la lista desplegable.
3. En el cuadro "Contraseña", escriba su contraseña.
4. Haga click en el botón "Aceptar".

Salir. Termina la ejecución del *SIIMCELE*.

MENÚ	FUNCIONES
Mediateca.	Acceso a la Mediateca. Bitácora. Consulta al Acervo.

Acceso a la Mediateca. Este formulario indica si el usuario puede o no ingresar a la **MEDIATECA** considerando la vigencia de su inscripción. Además, posibilitará un control estadístico de accesos para cada usuario.

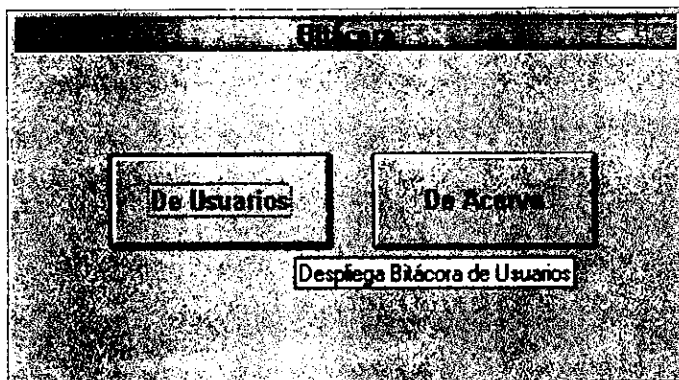


Cómo hacer un Acceso.

1. En el menú **Mediateca**, elija Acceso a la Mediateca.
2. Aparecerá un formulario en espera del valor del Identificador de Usuario, este valor es proporcionado por el mismo usuario al hacer pasar su credencial por el lector de ranura de códigos de barras.
3. El formulario Acceso a la Mediateca validará la operación anterior, verificando la existencia del identificador proporcionado, así como su vigencia en calidad de inscrito. Si se cumplen estas condiciones, el formulario Acceso a la Mediateca desplegará el mensaje: "Acceso permitido A:" *nombre del usuario*, en caso contrario, se desplegará el mensaje "Acceso denegado".
4. El formulario Acceso a la Mediateca estará en condiciones de espera de nuevos identificadores de usuario.

5. Para cerrar este formulario, se emplea la combinación: CTRL + Q.

Bitácora. Despliega el siguiente menú:



El botón "De usuarios" muestra el formulario que recopilará la información de actividades de los usuarios, así como sugerencias que hagan éstos con respecto al servicio.

 A screenshot of a user feedback form. At the top, it shows a date and time stamp "19/02/1999 08:43:31" and a user ID "190001". Below this is a table with four rows of feedback options. The first three rows have checkboxes that are unchecked, while the fourth row has a checked checkbox. At the bottom of the form, there is a button labeled "Agregar/Guardar registro".

NO TUVE NINGUN PROBLEMA	<input type="checkbox"/>
SOBRE LINGÜÍSTICA	<input type="checkbox"/>
MUY ÚTIL	<input type="checkbox"/>
PLATIQUE CON UN ASESOR	<input type="checkbox"/>
EL SERVICIO ES ADECUADO A MIS NECESIDADES	<input checked="" type="checkbox"/>

Cómo llenar una Bitácora de Usuario.

1. En el menú **Mediateca**, elija **Bitácora**.
2. En el cuadro siguiente seleccione "De usuarios".

3. Haga click en el icono "Agregar registro".
4. Pase el lector de códigos de barras sobre la credencial de usuario para transferir el identificador de usuario al formulario Bitácora, el cual mostrará el nombre del usuario. A continuación, haga las selecciones pertinentes en las listas desplegables y si existen comentarios, introdúzcalos en el espacio asignado para ellos.
5. Haga click en el icono "Guardar".
6. Salga del formulario con el icono "Salir del formulario".

El botón "De Acervo" muestra el formulario que recopilará la información que puedan proporcionar los usuarios acerca del acervo de la *MEDIATECA*.

Cómo llenar una Bitácora de Acervo.

1. En el menú **Mediateca**, elija Bitácora.
2. En el cuadro siguiente seleccione "De Acervo".
3. Haga click en el icono "Agregar registro".
4. Pase el lector de códigos de barras sobre las etiquetas identificadoras de los materiales. El formulario validará y mostrará el nombre del material, a continuación, haga las selecciones pertinentes en las listas desplegables.

5. Haga click en el icono "Guardar".
6. Salga de la selección de materiales con el icono "Salir del formulario". Aparecerá nuevamente el menú Bitácora y estará en espera de un nuevo llenado.

Consulta al acervo. El formulario de Consulta al acervo, permite al usuario de los recursos de la *MEDIATECA*, realizar búsquedas parametrizadas del material registrado.

Cómo hacer una Consulta al acervo.

1. En el menú **Mediateca**, elija Consulta al acervo.
2. En la lista desplegable "Tipo de material", seleccione uno de la lista (libro, videocassette, audiocassette, etc.).
3. En la lista desplegable "Idioma del material", elija uno de la lista, correspondiente al material que ha seleccionado en la lista anterior.
4. En el cuadro de texto "Palabra", escriba una descripción parcial (de al menos tres letras) o total de lo que desee buscar.
5. Haga click en el cuadro de selección "Contenido en" para elegir si desea que su palabra se busque en uno de los campos siguientes: Título, Autor, Temas o en

- todos los anteriores mediante la opción "Todo".
6. Opcionalmente puede hacer una búsqueda más selectiva, utilizando el cuadro de selección con las opciones "Y", "O", que permiten construir una nueva expresión.
 7. Haga click en el icono "Buscar", entonces aparecerá una lista mostrando los resultados.
 8. Además, es posible hacer una búsqueda de palabras sin considerar su acentuación, esto se logra habilitando con un click el cuadro "No distinguir acentuación".
 9. Si tiene alguna búsqueda adicional, presione la tecla ESC y repita nuevamente esta secuencia de ayuda.
 10. Para los niveles de uso: Coordinador de la Mediateca y Administrativo, existe el icono "Imprimir" para obtener los resultados de la búsqueda en papel. Para esto, se debe hacer un click sobre ese icono.

MENÚ	FUNCIONES
Administración.	Inscripción. Impresión de Credenciales. Registro de Acervo. Etiquetas de Acervo. Catálogos. Informes. Estadísticas.

Inscripción. Despliega el formulario que permite realizar inscripciones o bien, reinscripciones de usuarios.

Cómo hacer una inscripción o reinscripción.

Para la inscripción:

1. En el menú **Administración**, elija **Inscripción**.
2. Haga click en el icono "Agregar registro".
3. Capture los datos correspondientes en los cuadros de texto o haga las selecciones necesarias en las listas desplegadas en cada una de las tres páginas del formulario (se alterna entre las páginas, haciendo click en los nombres: Datos personales, Dirección y Lenguas en el CELE/Obs).
4. Haga click en el icono "Guardar". La fecha de registro, la vigencia y el Id. de Usuario (Identificador de usuario) se establecerán automáticamente.
5. Podrá seguir capturando registros siguiendo la secuencia anterior, o si lo desea, es posible salir del formulario con el icono "Salir del formulario".

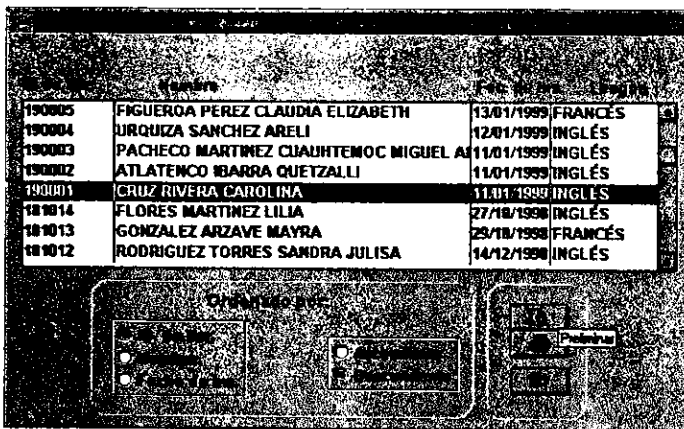
Nota: Es posible visualizar los datos previamente capturados de otros registros empleando los iconos de "Primer registro", "Registro anterior", "Registro siguiente", "Último registro" o "Buscar registro".

Para la reinscripción:

1. En el menú **Administración**, elija **Inscripción**.
2. Haga click en el icono "Modificar registro".

3. Haga click en el botón "Reinscribir". La vigencia se establecerá automáticamente.
4. Si es necesario, haga los ajustes correspondientes en los cuadros de texto y/o en las listas desplegables.
5. Haga click en el icono "Guardar".
6. Podrá seguir reinscribiendo siguiendo la secuencia anterior, o si lo desea, es posible salir del formulario con el icono "Salir del formulario".

Impresión de Credenciales. Este formulario permite imprimir las credenciales de los usuarios.



Cómo imprimir credenciales.

1. En el menú **Administración**, elija Impresión de Credenciales.
2. Seleccionar con un click el nombre de un usuario; o con la tecla SHIFT + Flechas de dirección, un grupo de nombres. El formulario permite ordenar la lista de los usuarios mediante su ID. de Usr. (Identificador de usuario), Nombre o Fecha de Ins. (fecha de inscripción) y de manera ascendente o descendente.
3. Si lo desea, haga click en el icono "Presentación preliminar" para observar la apariencia que tendrán las credenciales una vez impresas. Para imprimir las credenciales, haga click en el icono "Imprimir".
4. Podrá seguir imprimiendo credenciales siguiendo la secuencia anterior, o si lo desea, es posible salir del formulario con el icono "Salir del formulario".

Acervo. Le permite realizar el registro de los materiales del acervo de la *MEDIATECA*.

The screenshot shows a registration form with the following fields and values:

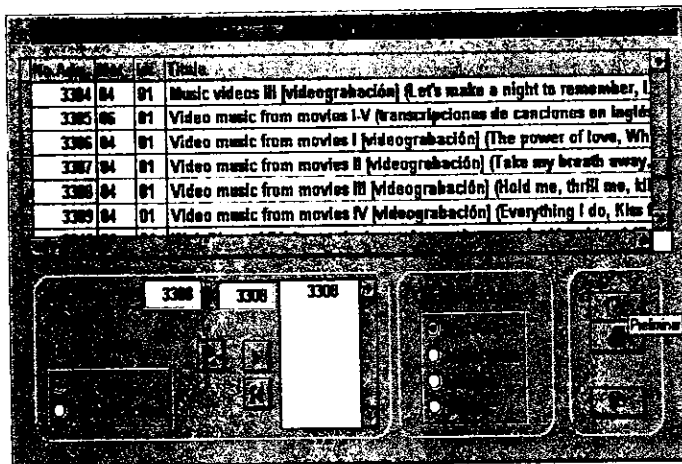
- Fecha de Ingreso:** 09/02/1999
- Forma de Adquisición:** UNIAM, CELE MEDIATECA
- Código de Adquisición:** 004409
- Categoría:** LIBRO
- Idioma:** INGLÉS
- Clasificación:** ICA1 A173Ej. 1
- Apellido del Autor:** Helgesen, Marc
- Título:** Active listening : Expanding understanding through context (Teacher's edition 3) / Marc Helgesen, Steven Brown, Dorothy Smith : ill. Adventure House ... [et al.]

Cómo registrar el acervo.

1. En el menú **Administración**, elija **Registro de Acervo**.
2. Haga click en el icono "Agregar registro".
3. Capture los datos correspondientes o haga las selecciones necesarias en las listas desplegadas, en cada una de las dos páginas del formulario (se alterna entre las páginas haciendo click en los nombres: **Página 1** y **Página 2**).
4. Haga click en el icono "Guardar". La fecha de ingreso y el número de adquisición se establecen automáticamente.
5. Podrá seguir capturando registros siguiendo la secuencia anterior, o si lo desea, están disponibles los iconos de "Copiar" y "Pegar" que permiten registrar información repetitiva sin necesidad de teclearla; para lograr esto, utilice los iconos de desplazamiento: "Primer registro", "Registro anterior", "Registro siguiente", "Último registro" o "Buscar registro" hasta encontrar el que contiene la información a copiar; haga click en el icono "Copiar" y ahora vuelva a emplear los iconos de desplazamiento para llegar al registro en el cual la información será copiada, en ese momento haga click en el icono "Pegar".

6. Existe una casilla de opciones llamada "¿Enviar a Consulta de Acervo?", en ella se eligirá "S" (Si) si se desea que los datos capturados estén a disposición de los usuarios para el módulo de Consulta al Acervo; en el caso contrario, la elección deberá ser "N" (No).
7. Es posible salir del formulario con el icono "Salir del formulario".

Etiquetas de Acervo. Permite generar las etiquetas con códigos de barras para los materiales del Acervo.



Cómo imprimir Etiquetas.

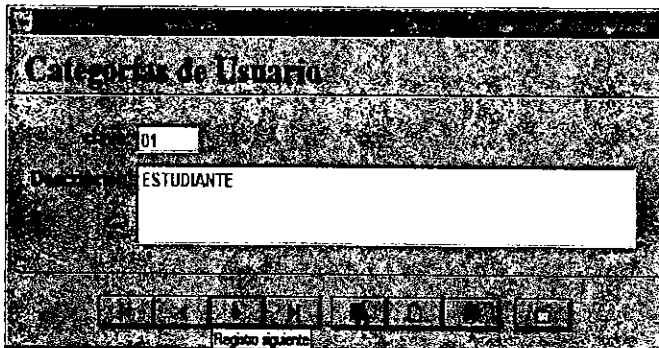
1. En el menú **Administración**, elija **Etiquetas de Acervo**.
2. Para seleccionar el número de etiquetas a imprimir, escriba el rango en cuadro de texto "No. de Adq." (Número de Adquisición) y a continuación haga un click en el icono de "Selección" (es una flecha que apunta hacia la derecha). También es posible seleccionar un rango haciendo click sobre el campo deseado de la lista mostrada y entonces un click sobre el icono de "Selección"; de esta manera, habrá seleccionado el número inicial del rango; para establecer el número final, repita lo anterior para otro de los campos mostrados en la lista. Si existe algún error en los números seleccionados, podrá cancelarlos empleando el icono de "Deselección"

(es una flecha que apunta hacia la izquierda).

3. La lista mostrada puede ser ordenada por: Número de adquisición, materiales, idioma o título, haciendo la selección correspondiente en el cuadro "Ordenado por".
4. Es posible visualizar la información completa del acervo haciendo un click en el icono "Muestra el Contenido de la Tabla Acervo".
5. Se cuenta con dos datos a imprimir: Número de Adquisición y Clasificación, seleccionados en el cuadro "Imprimir".
6. Puede tener una visión preliminar de las etiquetas a imprimir con el icono "Preliminar", o bien, obtener la impresión en papel con el icono "Imprimir".
7. Para salir del formulario, haga click en el icono "Salir".

Catálogos. En esta sección se encuentran agrupados los catálogos del sistema; el formulario del catálogo seleccionado permite realizar las tareas de altas y modificaciones a los registros correspondientes.

La imagen que se muestra a continuación es el ejemplo de un catálogo. Todos los catálogos existentes en el *SIIMCELE* son semejantes.



Cómo hacer altas y modificaciones (Instrucciones aplicables a todos los catálogos del SIIMCELE).

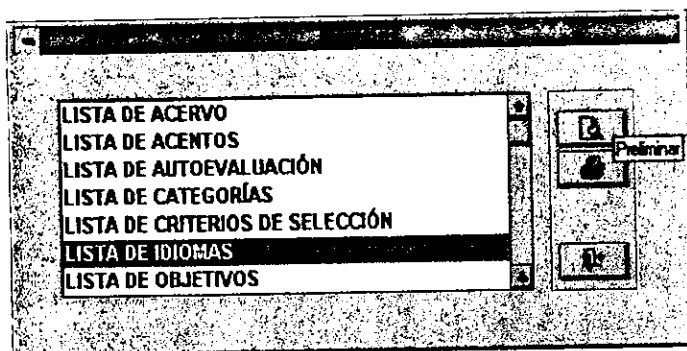
Para hacer altas:

1. En el menú **Administración** elegir la opción Catálogos y posteriormente, elegir uno de los ahí listados.
2. Haga click en el icono "Agregar registro".
3. En el cuadro de texto, capture el nombre correspondiente. La clave se establecerá automáticamente. Podrá deshacer equivocaciones empleando el icono "Revertir registro".
4. Haga click en el icono "Guardar".
5. Podrá seguir capturando registros siguiendo la secuencia anterior, o si lo desea, es posible salir del catálogo con el icono "Salir del formulario".

Para hacer modificaciones:

1. En el menú **Administración** elegir la opción Catálogos y posteriormente, elegir uno de los ahí listados.
2. Utilice los iconos: "Primer registro", "Registro anterior", "Registro siguiente", "Último registro" o "Buscar registro" como sea necesario, para visualizar el registro a modificar.
3. Haga click en el icono "Modificar registro".
4. Capture la modificación correspondiente.
5. Podrá seguir haciendo modificaciones siguiendo la secuencia anterior, o si lo desea, es posible salir del catálogo con el icono "Salir del formulario".

Informes. Este formulario permite emitir reportes de la información contenida en cada uno de los catálogos de la Base de Datos del *SIIMCELE* (Los nombres de los catálogos están listados en nuestro capítulo de Base de Datos).



Cómo imprimir Informes.

1. En el menú **Administración**, elija Informes.
2. Seleccione el informe requerido con SHIFT + las flechas de dirección o con un click.
3. Haga click en el icono "Imprimir", o si desea observar previamente la manera en que será impreso el informe, haga click en el icono "Preliminar".
4. Podrá seguir imprimiendo informes siguiendo la secuencia anterior, o si lo desea, es posible salir del formulario con el icono "Salir del formulario".

Estadísticas. Muestra una lista en la cual es posible seleccionar las estadísticas generadas por el *SIIMCELE* en cuanto a:

- Bitácora de Usuarios.
- Bitácora de Acervo.
- Asistencia de Usuarios.
- Materiales Ocupados.

Estadísticas de Bitácora de Usuarios. Este formulario muestra un resumen general de los datos ingresados en la Bitácora de Usuarios e informe de comentarios de usuarios, considerando un rango de fechas a elegir.

Cómo ver Estadísticas de Bitácora de Usuarios.

1. En el menú **Administración**, elija **Estadísticas**.
2. Haga click en la opción "Bitácora de Usuarios".
3. Seleccione un rango de fechas.
4. Haga click en una de las opciones: "Resumen Bitácora de Usuarios" o "Comentarios de Usuario".
5. Podrá ver los resultados haciendo click en el botón "Preliminar" u obtenerlos, haciendo click en el Icono "Imprimir".
6. Para salir del formulario haga click en el Icono "Salir".

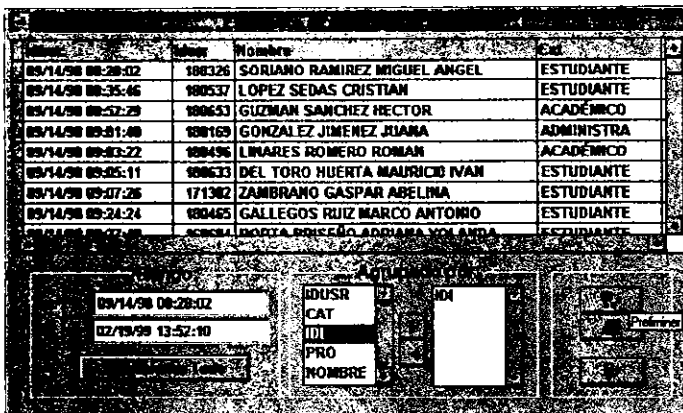
Estadísticas de Bitácora de Acervo. Este formulario muestra un resumen general de los datos ingresados en la Bitácora de Acervo, considerando un rango de fechas a elegir.

Cómo ver Estadísticas de Bitácora de Acervo.

1. En el menú **Administración**, elija **Estadísticas**.
2. Haga click en la opción "Bitácora de Acervo".
3. Seleccione un rango de fechas.

4. Podrá ver los resultados haciendo click en el botón "Preliminar" u obtenerlos, haciendo click en el botón "Imprimir".
5. Para salir del formulario haga click en el botón "Salir".

Estadísticas de Asistencia de Usuarios. Este formulario muestra un informe que puede ser agrupado y totalizado por Identificador de Usuario, Categoría, Idioma, Procedencia y Nombre de los Usuarios. Esta información es recabada en el Módulo de Acceso, es decir, considera sólo aquellos usuarios que han asistido a la *MEDIATECA*. Es necesario seleccionar un rango de fechas.

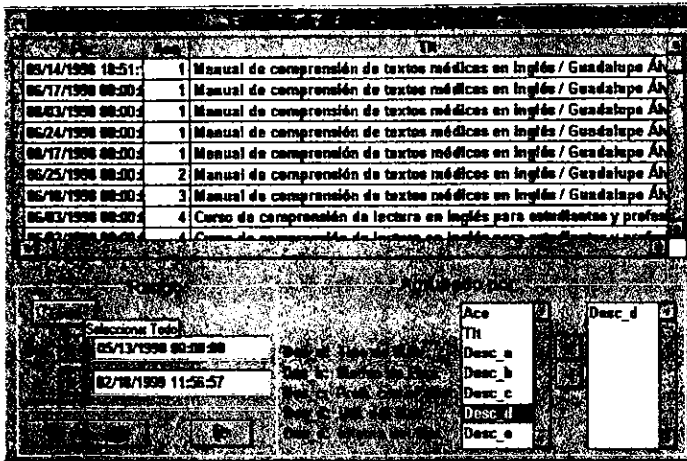


Cómo ver Estadísticas de Asistencia de Usuarios.

1. En el menú **Administración**, elija **Estadísticas**.
2. Haga click en la opción "Asistencia de Usuarios".
3. Seleccione un rango de fechas.
4. Seleccione los campos a agrupar, mostrados en la lista "Agrupado por".
5. Podrá ver los resultados haciendo click en el botón "Preliminar" u obtenerlos, haciendo click en el botón "Imprimir".
6. Para salir del formulario haga click en el botón "Salir".

Estadísticas de Materiales Ocupados. Este formulario muestra un informe que puede ser agrupado y totalizado por Identificador de Acervo, Título, Tipo de Material, Motivo

de Elección, Problemas con el Material, Utilidad del Material e Idioma del Material. Esta información es obtenida de la Bitácora de Materiales. Es necesario seleccionar un rango de fechas.

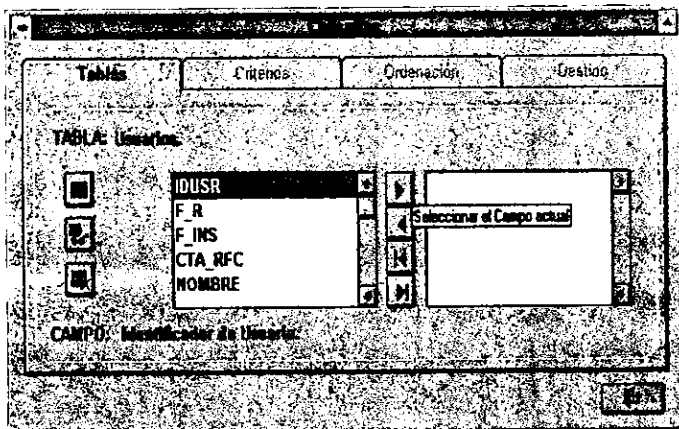


Cómo ver Estadísticas de Materiales Ocupados.

1. En el menú **Administración**, elija **Estadísticas**.
2. Haga click en la opción "Materiales Ocupados".
3. Seleccione un rango de fechas.
4. Seleccione los campos a agrupar, mostrados en la lista "Agrupado por".
5. Podrá ver los resultados haciendo click en el botón "Preliminar" u obtenerlos, haciendo click en el botón "Imprimir".
6. Para salir del formulario haga click en el botón "Salir".

MENÚ	FUNCIONES
Utillerías.	Consultas. Registrar Administrativos. Reindexar/Validar BD.

Consultas. Este formulario permite realizar las consultas no planeadas a la Base de Datos del SIIMCELE. Los resultados de las consultas pueden obtenerse en pantalla, impresora o almacenamiento en disco bajo diversos formatos. Adicionalmente, es posible efectuar bajas a los registros seleccionados.



Cómo realizar Consultas.

1. En el menú **Utilerías**, elija **Consultas**.
2. En el cuadro combinado "Tabla", elija una opción, es decir, una tabla.
3. Seleccionar al menos un campo de la tabla elegida (si no se selecciona ningún campo, podrán observarse todos los campos de la tabla elegida si se hace click en el icono "Muestra el contenido de la Tabla").
4. Haga click en la pestaña "Criterios".
5. Haga click en un valor de la lista desplegable "Campo".
6. Haga click en un valor de la lista desplegable "Operador".
7. Introduzca un dato en el cuadro de texto "Valor".
8. Es posible crear un criterio conjuntivo o disyuntivo haciendo click en "Y" u "O", si es así, repetir los pasos 5 al 7.
9. Opcionalmente establezca un criterio de ordenación haciendo click en la pestaña "Ordenación" y posteriormente seleccione un campo de la lista desplegable que ahí se mostrará.

10. Haga click en la pestaña "Destino".

11. Elija con un click el dispositivo de salida para transferir la información de su búsqueda.

12. Finalmente, haga click en el botón "Ejecutar".

13. El icono "Salir" estará disponible en cualquier momento.

Nota: Si usted ha elegido como dispositivo de salida, el almacenamiento en disco, podrá además seleccionar un nombre y formato de archivo como destino.

Registro de Administrativos. En este formulario, la coordinación de la *MEDIATECA* realizará altas y modificaciones a los registros de los administrativos, así como también asignará sus contraseñas y privilegios, condicionando de esta manera el acceso a los menús del *SIIMCELE*.

Nivel	01
Nombre	LIC. MARINA CHÁVEZ
Cargo	COORDINADORA
Teléfono	6989998
Hora	0:00
Fecha de Caducidad	02/20/99 08:33:17 PM
Contraseña	
Nivel	EGRESADA DE LA CARRERA DE REL. INTERNACIONALES DE LA FAC. DE CIENCIAS POLÍTICAS DE LA UNAM.

Cómo Registrar Administrativos.

Para dar de alta un registro:

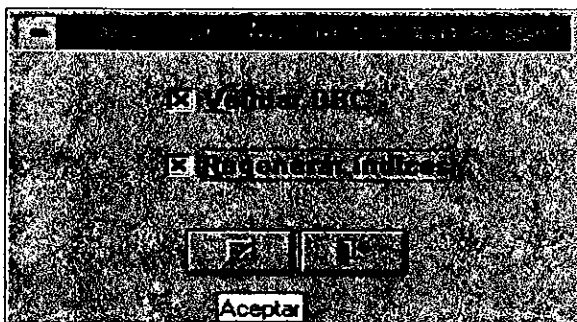
1. En el menú **Utillerías**, elija Registrar administrativos.
2. Haga click en el icono "Agregar registro".
3. Capture los datos correspondientes en los cuadros de texto o haga las selecciones necesarias en las listas desplegables. Es necesario recordar que el Nivel 1 corresponde al usuario de los servicios de la *MEDIATECA*, el Nivel 2 a los administrativos de la *MEDIATECA* y el Nivel 3 a la coordinación de la *MEDIATECA*.

4. Haga click en el icono "Guardar". La clave del administrativo se establecerá automáticamente.
5. Podrá seguir capturando registros siguiendo la secuencia anterior, o si lo desea, es posible salir del formulario con el icono "Salir del formulario".

Para hacer modificaciones a un registro:

1. Utilice los iconos: "Primer registro", "Registro anterior", "Registro siguiente", "Último registro" o "Buscar registro" como sea necesario, para visualizar el registro a modificar.
2. Haga click en el icono "Modificar registro".
3. Tecleé la modificación correspondiente.
4. Podrá seguir haciendo modificaciones siguiendo la secuencia anterior, o si lo desea, es posible salir del catálogo con el icono "Salir del formulario".

Reindexar/Validar DB. Con este formulario es posible validar la consistencia y/o regenerar los índices de la Base de Datos del *SIMCELE*.



MENÚ	FUNCIONES
Ayuda.	Contenido. Buscar ayuda sobre. Acerca de ...

Contenido. Muestra el índice de la Ayuda del *SIIMCELE*:

Buscar ayuda sobre. Muestra el cuadro de diálogo "Buscar" que permitirá localizar temas en la ayuda del *SIIMCELE*.

Acerca de. Muestra información de desarrollo del *SIIMCELE*.

8. PRUEBAS Y CONFIABILIDAD DEL SIIMCELE.

8.1 PRUEBA DE UNIDAD DEL SIIMCELE.

La Prueba de Unidad se centra en los módulos del sistema, siendo posible solucionar problemas tales como comparaciones entre tipos de datos distintos, terminación de secuencias inapropiadas y fallos en la presentación de la información de cada módulo. Con estas pruebas, se asegura que la información empleada por cada módulo del *SIIMCELE* conserva su integridad durante la ejecución.

La Prueba de Unidad también nos permite asegurar que los módulos del sistema funcionan correctamente en los límites de procesamiento establecidos por el equipo de cómputo con el que se cuenta en la *MEDIATECA*. Las pruebas se realizaron considerando a cada módulo de manera independiente y fueron llevadas a cabo tanto por los usuarios como por nosotros; el objeto era detectar fallos en los siguientes aspectos:

1. Tipificación impropia.
2. Inicialización o valores erróneos.
3. Nombres de variables incorrectos.
4. Tipos de datos inconsistentes.
5. Desbordamientos en los campos.
6. Comparaciones entre tipos de datos distintos.
7. Operadores lógicos o de precedencia incorrectos.
8. Variables o comparadores incorrectos.
9. Terminación de bucles inapropiada o inexistente.

Para la Prueba de Unidad, desarrollamos datos tanto válidos como no válidos que examinarían valores mínimos y máximos posibles, así como todas las variaciones que

sean posibles en formato para cada módulo. Verificamos con cuidado la salida de los datos de prueba tanto en pantalla, almacenamiento en archivos y en informes. A continuación mostramos unas tablas con los tipos de datos válidos de prueba para cada proceso del *SIIMCELE* indicados desde el DFD. Cada proceso tiene asociados campos de captura mediante el teclado o mediante lectores de códigos de barras; las pruebas se centraron en los campos que permiten la captura sin hacer una selección de las opciones disponibles en pantalla, es decir, en los campos que pudieran considerarse susceptibles a permitir errores de teclado. Obviamente, todos los campos ya han sido descritos en el Diccionario de Datos y los que no se incluyen en las siguientes tablas es porque o bien son establecidos automáticamente por el *SIIMCELE* o se eligen de valores previamente establecidos y que se muestran por medio de cajas de selección de opciones o por listas desplegables.

Proceso: *Inscripción/Reinscripción.*

Campos correspondientes al proceso.	Tipo de datos válidos.
No. de cuenta/RFC	Numérico de 8 a 11 dígitos. Ejemplos: 87256936 SOQM-720922
Nombre	Carácter hasta 50 símbolos. Ejemplos: Lee Je Wook. Monter Espinoza Ma. Del Rosario Cristina.
Fecha de nacimiento	Fecha con los siguientes formatos: dd/mm/aa y dd/mm/aaaa
Calle	Alfanumérico de cantidad indefinida de símbolos.
No. de calle	Alfanumérico de cantidad indefinida de símbolos.
Colonia	Alfanumérico de cantidad indefinida de símbolos.
Delegación/Municipio	Carácter hasta 30 símbolos. Ejemplos: Tlalpán. Magdalena Contreras.
Código Postal	Numérico de 5 dígitos. Ejemplo: 09770
Teléfono	Numérico hasta 10 dígitos. Ejemplos: 6930590 9159383557 28128965
Semestre de la lengua	Numérico de 1 dígito.
Observaciones	Alfanumérico de cantidad indefinida de símbolos.

Número de inscripciones	Numérico de 2 dígitos.
-------------------------	------------------------

Proceso: *Control de acceso.*

Campos correspondientes al proceso.	Tipo de datos válidos.
Identificador del usuario	Numérico de 6 dígitos. Ejemplo: 180228

Proceso: *Llenado de bitácora de usuarios.*

Campos correspondientes al proceso.	Tipo de datos válidos.
Comentarios	Alfanumérico hasta 200 símbolos.
Identificador del usuario	Numérico de 6 dígitos. Ejemplo: 180228

Proceso: *Llenado de bitácora de materiales.*

Campos correspondientes al proceso.	Tipo de datos válidos.
Identificador de acervo	Numérico de 6 dígitos. Ejemplo: 000056

Proceso: *Registro de acervo.*

Campos correspondientes al proceso.	Tipo de datos válidos.
ISBN	Alfanumérico hasta 20 símbolos. Ejemplos: 968-36-4152-0 333-17431-3 84-7553-001-X
ISSN	Alfanumérico hasta 20 símbolos. Ejemplos: 0036-981-X 0-551-31242-6
Clasificación	Alfanumérico hasta 25 símbolos. Ejemplos: CL00057 CL00136-147 Ej. 1 EO00010C
Título	Alfanumérico hasta 200 símbolos. Ejemplo: New York by John Scott. Enterprise one: English for the

	commercial world (workbook) / C. J. Moore, Judy West; il by Terry McKivragan and Oxford Illustrators.
Autor personal	Alfanumérico hasta 100 símbolos. Ejemplo: Black, Val and Alexander, L. G. (Louis George), 1932-?
Edición	Alfanumérico hasta 20 símbolos. Ejemplos: 2a. ed. Ed. rev.
Pie de imprenta	Alfanumérico de cantidad indefinida de símbolos.
Descriptor	Alfanumérico de cantidad indefinida de símbolos.
Serie	Alfanumérico hasta 80 símbolos. Ejemplos: Nelson English Language Teaching. Les Usuels. Longman Structural Readers: background; stage 1.
Notas	Alfanumérico de cantidad indefinida de símbolos.
Descripción	Alfanumérico de cantidad indefinida de símbolos.
Fondo	Alfanumérico hasta 40 símbolos. Ejemplo: UNAM, CELE MEDIATECA.
Resumen	Alfanumérico de cantidad indefinida de símbolos.
Referencias	Alfanumérico de cantidad indefinida de símbolos.
Dificultad	Alfanumérico hasta 30 símbolos. Ejemplos: Principiante. Principiante/Intermedio. Indefinido.
Catalogador	Alfanumérico hasta 4 símbolos. Ejemplo: ELS.

Proceso: *Consulta a la base de datos.* Esta consulta se realiza seleccionando las opciones disponibles.

Proceso: *Registro de administrativos.*

Campos correspondientes al proceso.	Tipo de datos válidos.
Nombre	Carácter hasta 50 símbolos.
Cargo	Carácter hasta 20 símbolos. Ejemplos: Coordinador. Administrativo.
Teléfono	Número hasta 10 dígitos.

Contraseña	Alfanumérico hasta 8 caracteres.
Notas	Alfanumérico de cantidad indefinida de símbolos.

Proceso: *Consulta al acervo*. Esta consulta se realiza seleccionando las opciones disponibles.

Los errores detectados fueron corregidos hasta que los usuarios mostraron su total satisfacción en el empleo de cada módulo.

8.2 PRUEBA DE INTEGRACIÓN DEL SIIMCELE.

La Prueba de Integración se establece con el objetivo de detectar errores en la interacción de todos los módulos del SIIMCELE. Con estas pruebas, se trabaja un módulo, a continuación se agrega otro, y así sucesivamente hasta tener el sistema completo ejecutándose.

Esta prueba fue realizada por nosotros en primer lugar; conforme la integración de módulos era satisfactoria, permitimos que los usuarios emplearan el sistema para la detección de otros errores hasta que el funcionamiento del SIIMCELE en cuanto a la integración de sus módulos fue la adecuada.

8.3 PRUEBA DE VOLUMEN DEL SIIMCELE.

Con la Prueba de Volumen, se asegura el funcionamiento adecuado del sistema en condiciones de gran demanda de recursos en cantidad y frecuencia.

Para este tipo de pruebas, tuvimos que trabajar con volúmenes de información cercanos a la realidad de trabajo de la MEDiateca.

El módulo de *inscripción*, fue probado con aproximadamente 2000 registros, siendo posible, en esta etapa, la impresión de 700 credenciales para los usuarios inscritos en ese momento.

Con la información almacenada por el módulo anterior y con esas credenciales impresas, el *control de acceso* fue también examinado con un promedio de 60 accesos de usuarios al día, durante un período de un mes.

Esta información nos permitió también comprobar el funcionamiento del módulo de *reinscripción*, en cuanto a volumen se refiere, con un promedio de 50 reinscripciones mensuales.

Los módulos del *llenado de bitácoras de usuarios y de acervo* se pusieron a prueba con un promedio de 40 llenados diarios cuyos usuarios empleaban unos 80 materiales del acervo.

Para el módulo del *registro del acervo* se contó con aproximadamente 2000 registros capturados, esta cantidad permitió unas pruebas confiables para el módulo de *consulta al acervo*, llevada a cabo por los usuarios de los servicios de la *MEDIATECA* a un promedio de 20 consultas diarias.

El módulo de *registro de administrativos* emplea un máximo 10 registros, la prueba en este caso, resultó sin complicaciones.

Todos los problemas de manejo de volumen detectados en ese periodo mensual, afortunadamente pudieron ser corregidos.

8.4 PRUEBA DE VALIDACIÓN DEL SIIMCELE.

Una vez concluidos el *SIIMCELE* y las pruebas correspondientes, el sistema fue puesto a disposición de la coordinación de *MEDIATECA* para que iniciaran la Prueba de Validación, que en este caso, corresponde a la comparación de los requisitos que ellos establecieron con el producto final para lograr la aceptación del sistema.

La Prueba de Validación se realiza una vez que se ha instalado el sistema en el equipo destinado a los usuarios y que se haya ofrecido la capacitación correspondiente en el manejo del *SIIMCELE*.

El *SIIMCELE* fue aceptado por la coordinación de la *MEDIATECA* del CELE.

9. CALIDAD Y MANTENIMIENTO DEL SIIMCELE.

9.1 CALIDAD DEL SIIMCELE.

Siguiendo las directrices para las revisiones técnicas formales, garantizamos la calidad del sistema mediante:

1. Realizar revisiones del sistema dentro de un ambiente de diálogo, con la intención de colaborar en la eliminación de los defectos y no de culpar a los posibles responsables.
2. Mantener el orden y el tiempo establecido para llevar a cabo las revisiones programadas.
3. Registrar las diferencias de puntos de vista sobre las revisiones para considerarlas con más cuidado en el momento programado para ello.
4. Tener en cuenta que las revisiones al sistema no implican soluciones a los problemas, éstos se resuelven posteriormente.
5. Registrar en papel, las declaraciones que tienen lugar durante las revisiones.
6. Preparar anticipadamente las revisiones e involucrar únicamente a los participantes estrictamente necesarios.
7. Estructurar una lista de las revisiones que se efectúan.
8. Establecer el plan de actividades que surgen como resultado de las revisiones y que tienen el objeto de hacer las modificaciones requeridas.
9. Conocer adecuadamente los elementos que son revisados y comprender las posibles implicaciones de esas partes en conjunto.
10. Tener presentes en cada revisión, los registros de las revisiones anteriores.

La relación entre los factores de calidad y las métricas correspondientes, se aplicaron al *SIIMCELE* como se muestra a continuación, es preciso señalar que tales métricas no son absolutas sin embargo, proporcionan una manera sistemática de valorar la calidad del sistema. Durante el proceso del sistema, cada factor de calidad indicado, era

verificado por las métricas correspondientes, con el objeto de garantizar la calidad del producto final.

Factor de Calidad	Métrica aplicada
Corrección	Complitud Consistencia Trazabilidad
Fiabilidad	Exactitud Complejidad Consistencia Tolerancia a errores Modularidad Simplicidad
Eficiencia	Concisión Eficiencia de ejecución Operatividad
Integridad	Facilidad de auditoría Instrumentación Seguridad
Mantenimiento	Concisión Consistencia Instrumentación Modularidad Autodocumentación Simplicidad
Flexibilidad	Complejidad Concisión Consistencia Capacidad de expansión

	<p>Generalidad</p> <p>Modularidad</p> <p>Autodocumentación</p> <p>Simplicidad</p>
Capacidad de pruebas	<p>Facilidad de auditoría</p> <p>Complejidad</p> <p>Instrumentación</p> <p>Modularidad</p> <p>Autodocumentación</p> <p>Simplicidad</p>
Portabilidad	<p>Generalidad</p> <p>Independencia del hardware</p> <p>Modularidad</p> <p>Autodocumentación</p> <p>Independencia del sistema</p>
Reusabilidad	<p>Generalidad</p> <p>Independencia del hardware</p> <p>Modularidad</p> <p>Autodocumentación</p> <p>Independencia del sistema</p>
Interoperatividad	<p>Estandarización de comunicaciones</p> <p>Estandarización de datos</p> <p>Generalidad</p> <p>Modularidad</p>
Usabilidad	<p>Operatividad</p> <p>Facilidad de formación</p>

9.2 MANTENIMIENTO DEL *SIIMCELE*.

Debido a que se detectaron y corrigieron los errores surgidos en el proceso de verificación de calidad durante el desarrollo del *SIIMCELE*, el mantenimiento que se realizará consistiría en la mejora del sistema, más que responder a una falla del sistema. A medida que cambien los requerimientos de los usuarios, el sistema y su documentación también cambiarán.

Este mantenimiento también contempla que los módulos del sistema podrían volverse a codificar para mejorar su eficiencia adaptándose a los cambios de la coordinación de la *MEDIATECA*.

Como parte del mantenimiento, establecimos canales de comunicación con los administrativos de la *MEDIATECA* que nos permitan una retroalimentación acerca de las necesidades cambiantes respecto al *SIIMCELE* y/o a sus posibles desperfectos.

10. CONCLUSIONES.

El análisis, desarrollo, prueba y mantenimiento de un sistema, requieren un enfoque sistemático que permita asegurar su aceptación por parte de los usuarios. Las necesidades y limitaciones del usuario deben estar determinadas y claramente establecidas para lograr un análisis confiable de la situación a resolver; en el diseño del producto es necesario tener presentes a los usuarios ya que ellos trabajarán directamente con el sistema y a quienes irá dirigido el manual de uso, las pruebas que se realicen a lo largo del desarrollo garantizan la aceptación y calidad del sistema. Y finalmente, las tareas de mantenimiento incluirán solicitudes de análisis de cambio, rediseño y modificación, así como actualización de la documentación.

Para lograr el desarrollo y mantenimiento de un producto de software, es necesario utilizar técnicas de resolución de problemas junto con las notaciones adecuadas. Para el caso de nuestro proyecto, realizamos en primer lugar, una descripción de la situación por resolver considerando las restricciones del problema para justificar una estrategia de solución computarizada que realice las funciones requeridas. La estrategia adoptada se fundamenta en el análisis y diseño estructurado, todo esto enmarcado en el ciclo de vida clásico de desarrollo de sistemas.

Cuando ya se tienen claros los requisitos y las funciones que realizará el sistema, el siguiente paso consiste en especificar la manera en que esto será estructurado en un producto de software; en este punto es necesario emplear herramientas de abstracción que nos permitan trabajar con los requisitos, independientemente del lenguaje de programación que se empleará en una etapa posterior a este proceso. Tales herramientas incluyen a los diagramas de flujos de datos que permiten especificar las actividades de los procesos y las transferencias de datos entre ellos de una manera gráfica, y precisamente, esta característica gráfica, facilita al mismo tiempo, la comunicación con los clientes en esta etapa de desarrollo del sistema. El uso del lenguaje en forma estructurada proporciona especificaciones paso a paso de un

algoritmo y; cuando se requiere la especificación de una lógica de decisión más compleja, se hace uso de tablas de decisión; como complemento al flujo de datos, se emplea el diccionario de datos, que contiene un listado organizado de todos los elementos de datos involucrados en el sistema permitiendo una comprensión precisa de las entradas, salidas, almacenamientos y pasos intermedios desarrollados por los procesos del producto de software.

Mientras se diseña el sistema empleando las herramientas ya descritas, es inevitable el refinamiento por pasos para lograr en cada ocasión, una mayor abstracción desde los asuntos generales a los detalles específicos. El diseño estructurado implica la conversión de diagramas de flujo en cartas de estructura, estas últimas, son elementos que documentan la estructura jerárquica y las interconexiones de los procesos de un sistema. Después, las cartas de estructura se redefinen mediante criterios de modularización tales como la cohesión de funcionalidad dentro de los módulos y el acoplamiento entre ellos. Por lo tanto, se puede resaltar que los aspectos más útiles del diseño estructurado son la transición sistemática de los diagramas de flujo de datos a las cartas de estructura y la comprensión del acoplamiento y la cohesión de los módulos.

El desarrollo de un sistema de programación incluye la determinación del tipo de sistema a desarrollar, en nuestro caso, se trata de un sistema de procesamiento de datos, que se apoya de una base de datos para el almacenamiento de la información, dentro de este ámbito existen terminologías y notaciones particulares que permiten manejar las relaciones entre los datos y entidades que requiere el sistema para su funcionamiento, tales relaciones se expresan gráficamente mediante los diagramas entidad-relación, y se apoyan en tablas bidimensionales en las cuales los renglones representan los registros y las columnas los atributos de la entidad. En general, se cuenta con un modelo para mostrar cómo se relacionan lógicamente los datos de un registro. El orden de los datos no es significativo, puesto que las tablas relacionales muestran las relaciones lógicas, no físicas. Y al solicitar cierta información, el sistema

produce resultados de la consulta que contiene específicamente lo solicitado gracias al empleo de otro concepto importante, el de normalización con el cual se simplifica la relación entre los campos de un registro.

Después del análisis y diseño, es momento de trabajar con un lenguaje de programación para traducir lo desarrollado en instrucciones para la computadora; las características disponibles en el lenguaje de programación ejercen influencia sobre la estructura del sistema, para nuestro proyecto, Visual FoxPro constituyó un instrumento útil en el manejo de los conceptos del análisis y diseño estructurado y de bases de datos.

Antes de la entrega final del sistema es necesario tener preparado el manual del usuario, documento de apoyo para el uso del producto tanto en las pruebas del sistema como en el uso del mismo después de su aceptación final.

Los objetivos de las pruebas fueron valorar y mejorar la calidad del sistema, siguiendo la idea de que la mejor manera de minimizar el número de errores en un programa es encontrarlos y eliminarlos durante el análisis y el diseño. La confiabilidad del sistema implica un conjunto de actividades que incluyeron diversas pruebas que garantizan el apego a las especificaciones de requisitos, principios generales de facilidad en el manejo del sistema, estándares en el lenguaje de programación, estándares organizacionales de la *MEDIATECA*, documentación y expectativas del usuario.

Las actividades que sucederán después de la entrega del producto caen dentro del mantenimiento del sistema, tales actividades incluirán, cuando así sea requerido, mejoras en el producto, adaptaciones a nuevos ambientes y corrección de problemas, todo esto con el objeto de proporcionar nuevas funciones e interacciones o trasladar el software a nuevos sistemas operativos y/o nuevas máquinas. Claro es que, todas estas posibles tareas se deben efectuar con un enfoque sistemático ordenado que analice los requisitos de las modificaciones cuidando el rediseño, reimplantación, revalidación y redocumentación de los cambios.

Finalmente, podemos establecer que el objetivo fundamental de este proyecto fue proporcionar un sistema de computadora que facilitará a los usuarios llevar a cabo su trabajo, atendiendo a la calidad del producto final y empleando los métodos, herramientas y técnicas que juzgamos convenientes para la situación por resolver que se presentó en la *MEDIATECA* del CELE. La enseñanza que obtuvimos, además de la relacionada con la metodología empleada, es que la ingeniería en computación es una actividad que involucra de manera intensiva a la gente, nosotros desarrollamos software que será usado por otras personas quienes nos establecieron sus necesidades y expectativas del producto final.

APÉNDICE.

CÓDIGOS DE BARRAS.

Un *código de barras* es en realidad, sólo una clave para un registro de una base de datos externa que contiene información útil.

El tipo de código de barras que debe elegirse para identificar cualquier objeto depende de la clase de información que se desea codificar sobre él, del espacio de impresión disponible, del grado de seguridad requerido y del equipo de escaneo con el que se cuenta.

Cuando se elige un tipo de código de barras, se debe asegurar que el lector de códigos sea capaz de leerlo. El tipo de lectores de códigos de barras es el principal factor cuando existen problemas y errores de lectura, aunque también pueden existir errores debidos al tamaño de los códigos: o muy pequeños o muy grandes. También debe asegurarse que el lector sea configurado para leer el tipo de código que se ha elegido junto con su suma de comprobación (un tipo de comprobación matemática para verificar que se está usando el código correcto).

La siguiente, es una lista de códigos de barras de uso generalizado:

- UPC-A.
- UPC-E.
- EAN/JAN.
- CODE 39 (3 OF 9).
- EXTENDED CODE 39.
- CODE 93.
- EXTENDED CODE 93.
- HIBC.
- INTERLEAVED 2 OF 5.

- CODE 128.
- UCC-128.
- MSI PLESSEY.
- CODABAR.
- PDF417.
- DATA MATRIX.
- CODE 49.
- CODE 16K.
- ZIP+4 POSTNET.

CÓDIGOS DE BARRAS EN EL *SIIMCELE*.

En el *SIIMCELE*, se propuso el empleo de códigos de barras con sus correspondientes lectores, para facilitar y agilizar la lectura de los identificadores de usuarios y de acervo en los procesos de *Control de acceso* y de *Llenado de bitácora*.

El código de barras seleccionado es el CODABAR ya que el uso principal de éste consiste en codificar los datos frecuentemente empleados por las bibliotecas; en la *MEDIATECA*, se utilizan datos numéricos similares para establecer los identificadores de los usuarios y del acervo. El código CODABAR puede imprimir valores numéricos y seis caracteres de puntuación. Es un código de longitud variable que puede codificar los siguientes 16 caracteres:

- Números: 0 a 9
- Símbolos: - \$: / . +

El CODABAR debe terminar y finalizar con uno de los cuatro caracteres especiales de inicio/alto, A, B, C o D.

El CODABAR es tan versátil como para permitir impresiones de códigos sobre una superficie de 4 cm. X 1 cm. que es el espacio disponible en las credenciales de los usuarios y en los materiales del acervo.

Los lectores seleccionados, permiten ser configurados para leer los códigos CODABAR y garantizan la lectura correcta de los datos. Son dos lectores tipo "lápiz" para la lectura de identificadores de usuario y de materiales en el proceso de *Llenado de bitácora* y otro lector más, tipo "ranura" para la lectura del identificador de usuario en el proceso de *Control de acceso*.

BIBLIOGRAFÍA.

1. Pressman, Roger. S., Ingeniería del software, Editorial Mc Graw-Hill, México 1993.
2. Fairley, Richard, Ingeniería de Software, Editorial Mc Graw-Hill, México, 1988.
3. Kendall K. E., and Kendall, J. E., Análisis y diseño de sistemas, Editorial Prentice-Hall Hispanoamericana, México, 1991.
4. Martin, J., Fourth Generation Languages, volume 1, Editorial Prentice-Hall, USA, 1985.
5. Tsai, Alice Y. H., Sistemas de bases de datos (administración y uso), Editorial Prentice-Hall Hispanoamericana, México 1990.
6. Yourdon, E. Techniques of program structure and design, Editorial Prentice-Hall, USA, 1979.
7. Yourdon, E. and Constantine, L., Structured design, Editorial Prentice-Hall. USA, 1979.
8. Blanchard, B. S. and Fabrycky W. J., Systems Engineering and Analysis, Editorial Prentice-Hall. USA. 1981.
9. Stevens, W., Myers, G. and Constantine, L., Structured Design, Editado por el MIT. USA. 1959.
10. Myers, G., The art of software tests, Editorial Wiley. USA. 1979.

11. McCall, J., Richards, P. and Walters, G., Factors in software quality, Editado por Ntis. USA. 1977.
12. Swanson, E.B., The dimensions of maintenance, Editado por el IEEE, USA, 1976.
13. Guía de características profesionales. Microsoft Visual FoxPro, Editado por Microsoft Corporation. USA. 1995.