

54
2 eg



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

**HERRAMIENTA BASADA EN LA TECNOLOGIA MULTIMEDIA
PARA EL AUTOAPRENDIZAJE DE LA ANATOMIA Y FISILOGIA
DEL SISTEMA NERVIOSO**

TESIS PROFESIONAL
QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A :
VICTOR ENRIQUE TAPIA TEC

DIRECTOR DE TESIS:
ING. ALBERTO TEMPLOS CARBAJAL



CIUDAD UNIVERSITARIA

1999

**TESIS CON
FALLA DE ORIGEN**

27/3/10



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.

PAGINACION

DISCONTINUA

Agradecimientos.

A mi DIOS:

Por dejarme soñar y dejarme realizar mis sueños.

Por dejarme vivir.

Por dejarme llorar.

Por dejarme sufrir.

Por darme la oportunidad haber conocido a tanta gente.

Por darme a la más grande persona en el mundo MI MADRE.

Por recordarme que solo tengo una sola vida y que debo vivirla al máximo

A mi Madre:

Por dejarme tener en ti a una amiga cuando quería platicar, a un consuelo cuando quería llorar, a una persona cuando quería reír, y alguien que siempre ha creído en mi cuando he querido soñar. Pero lo más importante por ser mi madre y estar ahí cuando te he necesitado. Recibe este logro como una meta que los dos empezamos. Gracias

A mi Padre:

Por haberme enseñado que en la vida solo tu decides hasta donde quieres llegar, y que hay que aprender a ser como tu quieres. Gracias por tu apoyo cuando lo necesite. Este fue uno de mis grandes retos y aquí esta mi promesa.

A mi hermana:

Por ser la persona que ha compartido 25 años de risa, llanto, alegrías, tristezas, y muchas otras cosas que nos hicieron crecer. Por creer en mi, por darme un consejo cuando lo necesite, por abrazarme cuando me hacia falta, por darme una gran razón para seguir soñando TU HIJO. Esto es también tu logro. Gracias.

A mi tío Martín:

Por tantos consejos que de ti nacieron cuando me veías crecer, por todas esas veces que me regalaste un abrazo y cuando me regañabas. Por ser una de las personas que a pesar de todo creíste siempre en mi. Y por siempre recordarme que solo se vive una vez, y que debo soñar a lo grande. Por esto y otras tantas cosas. Gracias.

Gracias a mi Abuelita Emidia y a mi abuelito por esa niñez que me regalaron y por estar conmigo cuando lo necesitaba.

A mis grandes amigos de toda la vida.

Jesús, Miguel y Ruben

Por todos las veces que no peleamos, por todas las travesuras que hemos hecho, por ese árbol viejo donde muchas veces jugamos, por ser mis amigos cuando los he necesitado, y por apoyarme a realizar éste objetivo. Gracias.

A Luis Vázquez.

Con quien he compartido más de 10 años de vida, por esos días de desvelo en los que soñábamos llegar a ser ingenieros, por esos días en los que no teníamos ni para un refresco, por ser un hermano para mí, y por preocuparte por no dejarme vencer y estirarme tu mano para levantarme.

A mis tías, a mi abuelita María, y a mi tío Juan:

Yola, Maru, y Elena por haberme abierto las puertas de su casa, por apoyarme, y enseñarme que lo que más importa es tener alguien con quien compartir todo lo que vives. Gracias por esos consejos que me regalaron.

A mis primos:

Alonso, Omar, Osvaldo, Joel, Edgar, Erica, Lidia, Yose, y Claudia.

Por darme la oportunidad de ser su amigo, tomarme en cuenta, y por apoyarme a llegar hasta aquí. Gracias.

A mis grandes amigos Luis, Felipe y Gabriel.

Con quienes inicie una gran etapa de mi vida, y que tengo la suerte de tener como amigos, por todas esas veces que hemos estado juntos, por compartir conmigo sus sueños, por estar ahí cuando los he necesitado y por apoyarme en todas mis loqueras. Gracias.

A Maribel.

Con quien viví una de la mejor época de mi vida, con la que aprendí muchísimo, con quien empecé muchos sueños, y quien me ayudo muchísimo en el inicio de éste objetivo. Gracias.

A la gran Banda Guevoncia:

Pilar (Moustrito), Carlos (Stomatchman), Eduardo (Churritoshead) y Víctor (Viguevas). Con quienes tuve la suerte de empezar éste difícil camino, con quienes compartí muchísimas cosas y con quienes espero contar por mucho tiempo.

Al Lic. Manuel González Oscoy.

Por cederme la idea de realizar este proyecto, por asesorarme y tener la paciencia de trabajar conmigo.

A mi director Ing. Alberto Templos Carbajal.

Por su experiencia, paciencia, su valioso tiempo para realizar este proyecto y llevarlo hasta el final, y por sobre todo por contribuir a que yo de el último paso para lograr mi titulación.

Agradecimiento especial.

A toda la gente que en éste momento no me acuerdo, y que más de una ocasión me ayudo sin ningún compromiso a lograr mi objetivo. Por todas las personas que me regalaron un poco de su valioso tiempo.

Índice

INTRODUCCIÓN	1
CAPÍTULO 1. BREVE HISTORIA DE LA INFLUENCIA DE LAS COMPUTADORAS EN LA EDUCACIÓN.	3
<i>1.1 Antecedentes en instituciones educativas.</i>	3
<i>1.2 El advenimiento de las computadoras en la educación.</i>	4
<i>1.3 Nuevas tecnologías de información.</i>	5
CAPÍTULO 2. EDUCACIÓN ASISTIDA POR COMPUTADORA.	8
<i>2.1 Aplicaciones educativas.</i>	8
2.1.1 Procesadores de texto	8
2.1.2 Tratamiento de gráficos	9
2.1.3 Hojas de cálculo	9
2.1.4 Automatización	9
2.1.5 Juegos	10
<i>2.2 Enseñanza asistida por computadora.</i>	10
2.2.1 Programas de prácticas.	12
2.2.2 Programas de evaluación.	12
2.2.3 Tutoriales.	12
2.2.4 Programas de simulación.	12
2.2.5 Programas de construcción de modelos.	12
2.2.6 Utilidades.	13
<i>2.3 Estrategias para la enseñanza.</i>	13
2.3.1 Sinéctica	14
2.3.2 Enseñanza no directiva.	14
2.3.3 Enseñanza de procesos grupales.	15
2.3.4 Procesamiento de información.	15
<i>2.4 Enseñanza programada.</i>	16
CAPÍTULO 3. INCORPORACIÓN DE LAS COMPUTADORAS A LA EDUCACIÓN.	17
<i>3.1 El ciclo educativo.</i>	17
3.1.1 Reglas de Pedagogía general.	20
3.1.2 Reglas de aprendizaje.	21
<i>3.2 Evaluación de costos y beneficios.</i>	22

CAPÍTULO 4. MULTIMEDIA UNA NUEVA HERRAMIENTA.	25
4.1 Breve historia de la tecnología Multimedia.	25
4.2 <i>¿Qué es la tecnología multimedia?</i>	25
4.3 <i>¿Qué ofrece la tecnología multimedia?</i>	26
4.4 Ventajas que ofrece la tecnología multimedia	27
4.5 <i>Desventajas de la tecnología multimedia.</i>	29
4.6 <i>¿Por qué la tecnología multimedia?</i>	29
CAPÍTULO 5. INGENIERÍA Y DISEÑO DEL SOFTWARE.	31
5.1 <i>La importancia del software.</i>	31
5.2 <i>El software.</i>	32
5.2.1 El software se desarrolla.	32
5.2.2 <i>El software no se "estropea".</i>	33
5.2.3. La mayoría del software se construye a la medida, en vez de ensamblar componentes existentes.	33
5.3 <i>¿Qué es ingeniería del Software?</i>	33
5.3.1 Ciclo de vida	34
5.3.2 Construcción de prototipos.	36
5.4 <i>Técnicas de cuarta generación (T4G)</i>	37
5.5 <i>Programación Orientada a Objetos y a Eventos</i>	39
5.5.1 Objetos	40
5.5.2 Mensajes.	41
5.5.3 Clases.	42
5.5.4 Métodos de organización.	43
5.5.5 Herencia.	44
5.5.6 Programación Orientada a Eventos.	45
CAPÍTULO 6. BREVE DESCRIPCIÓN DEL SISTEMA DE UNIVERSIDAD ABIERTA (SUA).	47
6.1 <i>¿Qué es el SUA?</i>	47
6.2 <i>Objetivos del SUA.</i>	47
6.3 <i>SUA en Psicología.</i>	47
6.3.1 Diferencia con el sistema escolarizado.	48
6.3.2 Modelo de estudio en el sistema abierto.	48
6.3.3 <i>¿Qué implica la modalidad grupal?</i>	50
6.4 <i>Problemática general de los estudiantes del Sistema Abierto.</i>	51

6.5	<i>¿Por qué Anatomía y Fisiología del Sistema Nervioso?</i>	52
6.6	<i>Proyecto actual.</i>	53
	CAPÍTULO 7. UN BUEN SOFTWARE.	54
7.1	<i>Sistemas basados en computadora.</i>	54
7.2	<i>Ingeniería de Sistemas de Computadora.</i>	55
7.3	<i>Análisis del Sistema.</i>	58
7.3.1	<i>Identificación de necesidades.</i>	58
7.3.2	<i>Estudio de Factibilidad</i>	58
7.3.4	<i>Análisis económico</i>	59
7.3.5	<i>Análisis técnico</i>	59
7.4	<i>Análisis del Sistema Propuesto</i>	60
7.5	<i>Definición de Requerimientos.</i>	60
7.6	<i>Descripción general del sistema propuesto.</i>	61
7.6.1.-	<i>Configuración del sistema</i>	61
7.6.2 -	<i>Presentación libre</i>	61
7.6.3 -	<i>Presentación por módulos</i>	61
7.6.4-	<i>Consulta de términos específicos de medicina o anatomía.</i>	62
7.6.5 –	<i>Consulta de imágenes.</i>	62
7.6.6-	<i>Consulta de Videos.</i>	62
7.6.7-	<i>Glosario.</i>	62
7.6.8	<i>Requerimientos Generales</i>	63
7.6.9	<i>Objetivos principales de los elementos del sistema propuesto.</i>	65
7.7	<i>Análisis estructurado</i>	66
7.7.1	<i>Diagramas de flujo de datos (DFD)</i>	66
7.7.2	<i>Diccionario de Datos</i>	67
7.7.3	<i>Diagramas de flujo de datos para el sistema propuesto y su diccionario de datos.</i>	69
7.8	<i>Modelación de datos.</i>	86
7.8.1	<i>Datos objetos, Atributos, y relaciones.</i>	86
7.8.2	<i>Diagramas de entidad-relación(E-R).</i>	86
7.8.3	<i>Diagrama de Entidad-Relación</i>	87
	CAPÍTULO 8. DISEÑO DEL SISTEMA	89
8.1	<i>¿Qué es diseño?</i>	89
8.2	<i>Ingeniería y diseño</i>	89
8.3	<i>El proceso de diseño.</i>	90
8.4	<i>Diseño y calidad de software.</i>	91
8.5	<i>Diseño de Datos</i>	92

8.5.1	Diccionario de Datos	92
8.5.2	Diccionario de Datos para el sistema propuesto.	93
8.5.3	Estructuras de datos	94
8.5.4	El proceso de normalización de datos del sistema propuesto.	97
8.6	<i>Arquitectura del software.</i>	101
8.6.1	Diseño Arquitectónico y de interfaz para el sistema propuesto.	102
8.7	<i>Diseño Procedimental</i>	105
 CAPÍTULO 9. DESARROLLO DEL SISTEMA		 115
9.1	<i>Implementación de la base de datos.</i>	115
9.1.1	¿Qué es un sistema de bases de datos?	115
9.1.2	Sistema de Manejo de Bases de Datos (DBMS)	117
9.1.3	Arquitectura de un sistema relacional	118
9.1.4	Almacenamiento de los datos y métodos de acceso.	121
9.1.5	Creación de la base de datos para el sistema propuesto.	123
9.2	<i>Programación del Sistema</i>	124
9.2.1	El Proceso de traducción	125
9.2.2	Elección de un lenguaje	126
9.2.3	Visual Basic Lenguaje de 4ª Generación	127
9.2.4	Uso de Visual Basic en el sistema Propuesto	130
 CAPÍTULO 10. PRUEBAS Y PUESTA EN MARCHA		 133
10.1	<i>Pruebas del Sistema</i>	133
10.1.1	Flujo de Información de la Prueba	133
10.1.2	Técnicas de Prueba	134
10.1.2.1	Prueba de la Caja Blanca	135
10.1.2.2	Prueba de la Caja Negra	136
10.1.3	Estrategia de Prueba del Software	137
10.2	<i>Puesta en Marcha del Sistema</i>	141
10.2.1	Capacitación a Usuarios	141
 CONCLUSIONES.		 143
 APÉNDICE 1		 147
 APÉNDICE 2		 152
 BIBLIOGRAFÍA.		 153

INTRODUCCIÓN

**Experiencia es el
Nombre que todos
Damos a nuestros
Errores.**

Oscar Wilde.

INTRODUCCIÓN

El avance tecnológico en el campo de la informática y la computación ha dado como resultado el desarrollo de programas dentro de las diferentes actividades que rodean al ser humano, tales como: investigación, simulación, optimización, entretenimiento, educación, comunicación, etc. El beneficio obtenido en cada una de estas actividades, ha sido en mayor o menor grado, dependiendo de los recursos que se asignen, así como de los conocimientos de las personas involucradas.

El campo de la educación, es un de los que han recibido muy pocos beneficios del desarrollo tecnológico, ya que no sólo éste se involucra en la actividad educativa, si no que son una conjunción de varias especialidades tales como: Psicología, Pedagogía y las diversas especialidades de la Computación.

En México el sistema educativo, cuenta con un escaso apoyo en este sentido, nos referimos a infraestructura computacional y a profesionistas que desarrollen los sistemas requeridos. Es el nivel superior el que cuenta con todos los recursos para satisfacer esta nueva necesidad educativa. Desgraciadamente la mayoría de los recursos, sólo son utilizados como sustitutos de las máquinas de escribir, y otros equipos de oficina. Ignorando que las computadoras pueden convertirse en una gran herramienta de apoyo para la educación a todos los niveles.

Sacando provecho de lo último en tecnología, así como de las diferentes teorías existentes entorno al proceso enseñanza-aprendizaje, se pueden obtener sistemas para todo tipo de educación tales como: especial, abierta, escolarizada, etc. Es el sistema abierto tal vez el que menos apoyo tiene en este rubro, ya que el proceso de enseñanza-aprendizaje recae casi en su totalidad en los alumnos. Es por ello que se deberían desarrollar sistemas enfocados a este tipo de educación, la cual se ha ido convirtiendo en una alternativa importante dentro de la preparación de futuros profesionistas.

El presente trabajo, tiene la finalidad de comenzar un proyecto que deberá abrirse camino dentro del sistema de educación abierta. Además de que aprovechando las diversas posibilidades que nos ofrecen la tecnología multimedia y lo último en programación, deberemos obtener un sistema capaz de satisfacer las necesidades de los estudiantes del sistema abierto.

Cabe destacar que, la realización de un sistema de estas características, es un trabajo que implica cuestiones psicológicas y pedagógicas, las cuales deben ser contempladas en el transcurso del análisis, diseño y implementación, por lo que el presente trabajo involucrará temas que no son propiamente utilizados en la ingeniería. Por lo anterior se tratarán de exponer de una forma clara y precisa.

Este trabajo está dividido en tres partes, la primera plantea los aspectos educativos que deben contemplarse en el diseño de sistemas educativos, la segunda parte está dedicada a la metodología que se debe aplicar en el diseño de cualquier sistema, y la última al desarrollo del sistema de autoaprendizaje.

CAPÍTULO 1

Breve historia de la influencia de las computadoras en la educación.

**Mira hacia atrás y riéte de los
Peligros pasados.**

Sir Walter Scott.

CAPÍTULO 1

Breve historia de la influencia de las computadoras en la educación.

1.1 Antecedentes en instituciones educativas.

Desde que las computadoras fueron una realidad, las diferentes instituciones educativas de todo el mundo, se han preocupado por implementar una gran variedad de programas y sistemas que apoyaran la educación. Algunas instituciones han tomado muy en serio este campo de aplicación, por lo que se han conjuntado equipos de investigación, para encontrar nuevas alternativas en la producción de este tipo de sistemas.

En los años 60's y 70's diversos centros de investigación en las universidades y la industria ofrecieron algunos modelos de la forma en que los ordenadores podrían ayudar en el proceso de aprendizaje.

Un ejemplo de cómo los centros educativos han estructurado la investigación es el Reino Unido, el cual a través de sus diferentes universidades ha asignado una área específica (tabla 1).

Tema	Universidad
Estudios de literatura y lingüística	Oxford
Lenguas Modernas	Hull
Historia	Glasgow
Humanidades	Bath
Servicios Humanos	Southampton
Música y Geografía	Lancaster
Sociología y política social	Stirling
Psicología	York
Medicina	Bristol
Biología Y Química	Liverpool
Física	Surrey
Ingeniería	Londres
Estadística y Matemáticas	Birmingham y Glasgow
Computación	Ulster
Contaduría	East Anglia
Biblioteconomía y Estudios de información	Loughborough
Leyes	Warwick
Economía	Bristol
Agricultura	Aberdeen

Tabla 1. Area de investigación en las Universidades

Con respecto a México, existen valiosos intentos de desarrollo, por ejemplo la fundación Rusembluth, el Instituto de Comunicación Educativa, y otras instituciones de educación superior. Pero no obstante el gran esfuerzo, estos tienen limitaciones en la estructuración psicopedagógica del guión instruccional, parte muy descuidada, sobre todo la enfocada a utilizar los últimos avances en el área de cómputo.

1.2 El advenimiento de las computadoras en la educación.

La aparición de computadoras de tiempo compartido en los años 60 despertó grandes esperanzas sobre la utilización de las computadoras en la educación. El tiempo compartido consistía en el hecho de compartir la computadora con el mayor número posible de usuarios.

Esta técnica vino a ser el primer paso hacia la aparición de la computadora personal. Antes, la comunicación entre la computadora y el usuario se supervisaba a través del <<procesamiento Batch>>, por medio de los cuales los programas elaborados se introducían en la computadora y, después de bastante tiempo, aparecía impresa la respuesta de éste.

En los años setenta todo el trabajo se orientó hacia equipar la computadora --en condiciones de explotación según el tiempo compartido-- con el fin de poder ser útil al mayor número de personas con la mayor prontitud posible. En esta década, la exigencia de unos resultados inmediatos y sus consecuencias a gran escala relegaron la utilización de las computadoras, predominantemente, a tareas de enseñanza asistida por computadora (EAC) y de programación elemental en lenguaje BASIC. Las sesiones de EAC podían ser breves y con el tiempo controlado y no recargaban la memoria de la computadora. Por otro lado, la programación era la que más tiempo exigía y la que más ocupaba memoria de la computadora. No obstante, BASIC, en comparación con otros lenguajes, como FORTRAN, no ocupaba excesiva memoria.

En todos los niveles de la enseñanza, BASIC y la EAC fueron acogidos muy favorablemente. Pero esta aceptación no duro demasiado, ya que los programas universitarios de informática eligieron lenguajes estructurados y procedimentales, como PASCAL o LISP como vehículos de comunicación con la computadora.

En los años 80, la utilización de los ordenadores en las escuelas ya no se basaba en el tiempo compartido, sino en manejo de pequeñas computadoras personales. La utilización principal de éstas fue como libros de texto, es decir una

gran secuencia de texto, donde se daban explicaciones, y procedimientos para ciertos problemas. Estos sistemas ocupan parte de la memoria de la computadora, no sólo para operar y almacenar los programas y sus respuestas grabadas si no también para fijar los datos relativos al rendimiento de los estudiantes, y hacerlos en otro momento disponibles. Es a finales de esta década, cuando surgen nuevas tecnologías que involucran a las computadoras personales, por lo que los programas, en si mismos no son ni complicados ni muy extensos.

También es a finales de esta década cuando surgen nuevos lenguajes de propósito general tales como Smalltalk, Logo y Prolog, los cuales están basados en modelos cognoscitivos de como aprende la gente.

1.3 Nuevas tecnologías de información.

Según los expertos y considerando el poder y la interfaz entre el usuario y la computadora, el desarrollo de esta última puede ser definido en tres generaciones. La primera generación (IBM p.c. Apple II) se caracteriza por tener limitaciones gráficas en el proceso de comunicación entre el usuario y la computadora; por tanto la interfaz esta basada fundamentalmente en texto, y tiende a caer en desuso. La segunda generación recurre al uso de gráficos para establecer la relación con el usuario, facilitándole el aprovechamiento del sistema. EL caso típico de esta generación es el sistema Windows. En lo que se refiere a la tercera generación, además de contar con las dos características anteriores, se incorporan tres nuevos elementos: sonido, animación y vídeo, llamando a estas nuevas características tecnología Multimedia, la cual será descrita en el capítulo V.

En cuanto al manejo de información, existen actualmente varias tecnologías, en las cuales la constante es compartir la información utilizando los menores recursos físicos posibles, con gran rapidez y sin importar la distancia entre los usuarios. Ejemplos de éstos son: las diferentes plataformas de Redes que se tienen(LAN, WAN, MAN, etc.). Arquitecturas Cliente/Servidor, en esta nueva tecnología de tratamiento de información se sigue conservando la idea de compartir la información, sólo que ahora surgen dos nuevos elementos, el servidor y el cliente. El primero tendrá la tarea de almacenar la mayor cantidad de información con respecto a un tema, el segundo tiene por meta pedir información a los primeros.

Los servidores se pueden clasificar en los siguientes grupos:

- **Servidores de Archivos.** Permite compartir archivos entre un grupo de trabajo, es decir cuando un usuario solicita un archivo el servidor envía una copia completa al cliente.
- **Servidores de datos y de cómputo.** Éstos se emplean conjuntamente, el servidor de datos se encarga de proporcionar los datos que el servidor de cómputo le solicita, para ser enviados al respectivo cliente.
- **El servidor de aplicaciones es aquel que efectúa tanto el manejo de los datos, como el procesamiento de la aplicación, o sea, es un servidor de datos y un servidor de cómputo en la misma máquina.**

El cliente es una combinación de software y hardware que invoca los servicios de uno o varios servidores, e incluso de otro cliente.

Las ventajas más importantes son:

- Permite un mejor aprovechamiento de la potencia de cómputo de los equipos al poder descargar en el cliente parte de la carga de trabajo del servidor.
- Permite el acceso de un cliente a varios servidores en forma simultánea, lo que resulta cada vez más importante en los ambientes heterogéneos actuales
- Reduce el tráfico en la red, ya que a ella sólo viajan los requerimientos y las atenciones a ellos.
- La arquitectura Cliente/Servidor es particularmente adecuada para los sistemas de Procesamiento de Transacciones en Línea.
- Puede, y de hecho, tiene que operar bajo sistemas abiertos, lo que significa capacidad de cambio de plataformas con mínimo de problemas y riesgos.
- Permite el uso de interfazs gráficas de usuario muy versátiles y amigables en los clientes

Como desventajas tenemos que las aplicaciones desarrolladas para Cliente/Servidor suelen ser más complejas que las tradicionales en la forma de anfitrión/terminales, y que exigen más de la red. En ambos casos, las desventajas pueden ser superadas mediante las ventajas anotadas.

Los expertos en seguridad informática mantienen, que bajo Cliente/Servidor la seguridad es más sencilla de romper.

El potencial que ofrece esta nueva tecnología en el campo de la educación, es bastante amplio, ya que permite que un solo sistema sea aprovechado por un número considerable de alumnos al mismo tiempo, además de poder almacenar una gran gama de datos acerca de ellos.

CAPÍTULO 2

Educación asistida por computadora.

**No se puede superar un riesgo
Sin vencer otro mayor.**

Fernando G. Tola.

CAPÍTULO 2

Educación asistida por computadora.

2.1 Aplicaciones educativas.

Las computadoras han entrado en los centros escolares y sin embargo muchos maestros y profesores no saben muy bien que pueden hacer con éstas. Muchos afirman estar interesados, intrigados incluso, pero se sienten bloqueados por su propia falta de formación en el tema. Esto también se ve reflejado, en los profesionales de la educación, los cuales tienen una gran carencia de conocimientos en cuanto a informática y computación se refiere.

Un inventario de los usos y aplicaciones de la informática y de las computadoras en el medio escolar debe coordinar dos puntos de vista. De entrada, el que concierne a la lista de actividades que practican los niños y adolescentes, considerando en que medida la computadora puede ser asociada a cada una de ellas. Porque si bien es cierto que la computadora augura un magnífico futuro como instrumento pedagógico, es posible que su uso aporte también cambios notables en los métodos educativos. Es así pues que se presenta de forma breve algunas de las actividades en las que las computadoras y la informática pueden jugar un papel muy importante¹.

2.1.1 Procesadores de texto

La escritura, en sus múltiples usos. Ocupa un lugar destacado del trabajo escolar. En este terreno la informática puede aportar distintos tipos de ayudas: es lo que corrientemente se conoce como tratamiento de textos, una actividad que comprende no sólo la edición (en el sentido informático) de los escritos, sino también la organización de las ideas y los argumentos.

Estos programas proporcionan siempre una herramienta correcta, incluso muy calificada, para la edición de todo tipo de documentos individuales, como redacciones o ejercicios, y también colectivos, como periódicos escolares, resultados de encuestas, etc. En la enseñanza, por último, se extraerán otras ventajas suplementarias del uso de las aplicaciones de tratamiento de textos en circunstancias comparables a las de la creación, la edición o la ofimática. Cada

¹ Informática, Educación y Psicología del niño Jean-Pierre Dufcyer
Biblioteca de Psicología. PP 106-121

vez que sea necesario generar, modificar, <<triturar>> un texto, se hallará siempre un editor que permita hacerlo con poco esfuerzo.

2.1.2 Tratamiento de gráficos

Las herramientas de tratamiento gráfico son tan potentes e interesantes como las de textos. Los editores gráficos disponibles en la actualidad en el mercado ofrecen numerosas posibilidades². Algunos de ellos están destinados a realizar sólo dibujos, otros a procesarlos, otros a crear animaciones, etc. El producto final puede, a su vez, insertarse en un texto.

Otras aplicaciones permiten crear esquemas o planos. El usuario determina la forma, el tamaño, las proporciones y los colores de los elementos gráficos que desee crear.

2.1.3 Hojas de cálculo

Desde hace mucho tiempo se ha estado utilizando el término "Computadora Electrónica". Esto proviene del hecho de que las primeras aplicaciones y, en consecuencia, los primeros lenguajes de programación, como el Fortran, fueron esencialmente concebidos para el cálculo científico.

Desde este punto de vista la computadora aparece como una supermáquina de calcular programable. Es posible captar de inmediato qué tipo de ventajas pueden extraerse de tales aparatos, en particular cuando se alcanza un cierto nivel de estudios, como en las clases de bachillerato o en la universidad. Esto nos describe una gran gama de aplicaciones orientadas hacia operaciones de gran complejidad, así como simulaciones de fenómenos que difícilmente podrían imaginarse sin el uso de las computadoras.

2.1.4 Automatización

Bajo el nombre de robótica, la informática y la computación han entrado al mismo nivel de la industria. El control de las máquinas por computadora, o por una persona asistida por una computadora, se ha convertido en un fenómeno

² No obstante, ya no es posible solamente hablar de tratamiento de gráficos, ya que existen también programas dedicados a las imágenes digitales.

ordinario. La actual enseñanza técnica y profesional no puede dejar de lado esta realidad.

2.1.5 Juegos

Los <<Juegos electrónicos>> no tienen muy buena reputación en el mundo de la educación. Esta ha sido ganada, debido a que la mayoría de ellos son mediocres, nulos e incluso estúpidos. La mayoría sólo se enfocan a la sensomotricidad, otros inducen comportamientos de agresividad, etc.

Sin embargo se han desarrollado categorías de juegos que cultivan otro tipo de habilidades en los niños. Algunos acarician lo imaginario, algunos otros refuerzan ciertos conocimientos de diferentes tipos. El interés de tales juegos es innegable y no debemos quitarles la clasificación de educativos. Con todo, la mayor parte de estos programas son todavía demasiado pobres o estereotipados. Pero el principio que los rige es prometedor.

Esto sólo es una pequeña muestra de lo que nos pueden brindar las computadoras y la informática en torno a la educación³

2.2 Enseñanza asistida por computadora.

Bajo la etiqueta enseñanza asistida por computadora(EAC) es posible encontrar prácticas muy diversas. Todo depende, en efecto, de lo que es asistido y del grado de asistencia. Los usos de la computadora e informática en la enseñanza son muy variados, Hebenstreit⁴ lo subraya: el ordenador puede ser usado de mil maneras: como una pizarra, como un aparato donde simular fenómenos o como una máquina de proponer ejercicios.

Un docente puede decidir instalar una computadora en su clase para usarla para labores de servicio: gestión de ficheros, creación de textos de impresión, etc. Su enseñanza estará asistida (técnicamente) por la computadora, utilizando como (y únicamente como) una herramienta de gestión y edición.

³ En lo que respecta a la instrucción y el aprendizaje se le tratará en el siguiente tema.

⁴ Hebenstreit, J., EAC, *mode démploi*, << Éducation et informatique>>, no. 11 (1982) 34-35.

Otro puede plantearse que los alumnos hagan prácticas de programación, en la creencia de que ello conducirá necesariamente a ejercicios algorítmicos que, a su vez, producirán efectos positivos en la educación del razonamiento. Su enseñanza estará también, en ese caso, asistida por la computadora, que creara las oportunidades para realizar trabajos propios de ciertas actividades de resolución de problemas.

Algunas personas, se dedicarán a escribir sus propios programas de evaluación automática, cosa que aligerará las labores de corrección de los ejercicios, suministrando de paso y con mayor regularidad balances más rigurosos de los alumnos y las clases.

En todos estos ejemplos, los conocimientos que constituyen la materia prima de la enseñanza no son impartidos por la computadora, pero hay otras circunstancias en las que puede recurrirse a estas máquinas para impartir auténticos aprendizajes: con el uso de programas didácticos o tutoriales. Son programas diseñados con el objetivo de hacer adquirir cierto tipo de conocimientos a los alumnos, y pueden estar más o menos desarrollados. Los que hay son capaces de adaptarse a la personalidad o al ritmo de aprendizaje individual, de suministrar refuerzos, incluso de elaborar evaluaciones.

Pueden distinguirse al menos tres grandes categorías de posibles usos. En primer lugar, las utilidades (gestión, edición, etc.), de las que ya se mencionaron algunos ejemplos. Luego, los que tienen como objetivo la adquisición de aptitudes generales, cuyo ejemplo más típico son las enseñanzas destinadas al aprendizaje de la programación, y concretamente todas las experiencias que se han llevado a cabo alrededor del entorno logo. Tradicionalmente, no se ha considerado que estas dos categorías de usos formen parte de la EAC. Este último término, que constituye una categoría a parte abarca fundamentalmente aquellas aplicaciones de la informática que contemplan la enseñanza de conocimientos determinados y precisos, a través de un software formado por programas didácticos y tutoriales.

En principio, los instrumentos de la EAC son los programas didácticos, es decir, aquellos programas que están destinados a hacer aprender algo a quien lo utiliza. Este software puede tener funciones⁵, muy diversas, complementarias y más o menos amplias.

⁵ Informática, educación y psicología del niño Jean-Pierre Dufoyer
PP 130-150

2.2.1 Programas de Prácticas.

En este caso, la computadora no realiza más que la función de un repetidor. No hay aporte de nuevos conocimientos. El sujeto recibe sanciones en forma de refuerzos. Las prácticas pueden versar sobre conocimientos determinados, pero implicar también ciertos aspectos del desarrollo sensorial, motor o perceptivo.

2.2.2 Programas de evaluación.

La computadora hace el papel de quien pregunta y corrige. Una vez concluida la ejecución puede emitir una nota, un balance global o incluso sugerencias para continuar el aprendizaje. Estos programas ejecutan algo parecido a un ejercicio escolar o a un examen.

2.2.3 Tutoriales.

Este software toma, de algún modo el lugar del docente. Propone un aprendizaje de determinados conocimientos, generalmente, a través de una estructura de presentación/evaluación. Las secuencias pueden ser más o menos largas y estar más o menos fragmentadas en distintos aspectos de modo que es posible encontrar largas presentaciones -parecidas a una lección- seguidas de un control de los conocimientos adquiridos, o bien, y es el caso más frecuente, unidades sucesivas de presentación/evaluación.

2.2.4 Programas de simulación.

Destinados a promover, fuera de su entorno verdadero, aprendizajes que pueden ser utilizados en circunstancias reales. El ejemplo más ilustrativo es el de los simuladores de vuelo que permiten aprender sin poner en peligro a nadie.

2.2.5 Programas de construcción de modelos.

En este tipo de programas el alumno se enfrenta a un caso particular o a un conjunto de datos determinados, pero tienen la oportunidad de variar a su conveniencia algunos parámetros y constatar los efectos de dichas variaciones.

Todo ello permite construir, poco a poco, un modelo (o una ley general), examinarlo y verificarlo.

2.2.6 Utilidades.

Este software no aporta conocimientos, sino asistencia a las labores de *concepción y creación(textos, dibujos o gráficos) o de gestión de documentos o conocimientos.*

2.3 Estrategias para la enseñanza.

Las formulaciones de las teorías del aprendizaje son descriptivas y tratan de pormenorizar las condiciones que influyen en el proceso adquisitivo. Los enunciados referentes a la instrucción o enseñanza son prescriptivos, puesto que indican la necesidad de crear las condiciones óptimas del aprendizaje. Todavía no contamos con una definición común de la enseñanza, es por eso que en este tema cuando utilicemos el vocablo enseñanza lo tomaremos en la acepción de crear, conservar y ajustar debidamente las condiciones del aprendizaje para satisfacer las necesidades del individuo y las de todo el grupo. Por enseñanza eficaz entendemos la aplicación inteligente de conocimientos; por ejemplo la de los principios del desarrollo, aprendizaje y motivación a un contexto específico de instrucción.

Se describirán en forma breve algunos de los modelos de enseñanza⁶ orientados hacia el desarrollo personal y social

Weil, Joyce y Kluwin⁷ ofrecen dos grandes modelos de instrucción en el desarrollo personal: sinéctica y enseñanza no directiva. Ambos modelos se centran en la realización del potencial humano.

⁶ t.l. good/j.e. brophy Psicología educacional 2ª. Edición PP 371-389

⁷ Weil, M., Joyce, B., & Kluwin, B. Personal Models of teaching Englewood Cliffs, N.J.: Prentice Hall 1978.

2.3.1 Sinéctica

Es un modelo para ayudar a los alumnos a ser más creativos. Esta basada en dos estrategias básicas: primero ayudar a que los estudiantes perciban lo "familiar" desde ángulos más creativos; segundo, darles mayor sentido a las ideas desconocidas. Aquí la función esencial del maestro consiste en ayudar a los estudiantes a postergar la realización de una tarea y que empiecen a reflexionar sobre varios enfoques de un mismo problema. En ese momento pueden trabajar por su cuenta e integrar las posibilidades según su propio punto de vista.

2.3.2 Enseñanza no directiva.

Tiene por objeto lograr que los individuos adquieran mayor conciencia de sus emociones. Esos modelos se tomaron de la obra de Carl Rogers y generalmente son de mayor utilidad en las relaciones personales, aunque son aplicables igualmente a las discusiones de grupo que se centran en el desarrollo personal. Este modelo subraya que la misión de los maestros es facilitar el aprendizaje de sus alumnos, no controlarlo activamente. Una suposición fundamental es que los alumnos desean asumir la responsabilidad de su aprendizaje. En consecuencia, el maestro tiene la obligación de ayudarles a entender las percepciones de sí mismos y a conseguir una mejor integración personal. Otra suposición es que desean aprender.

Por lo que respecta a los modelos sociales, destacan la necesidad de mejorar la disposición del individuo para entender a los demás y trabajar con ellos dentro de un contexto social. Se dispone de muchísimos modelos sociales de enseñanza, algunos son los siguientes:

Representación de papeles: Los que estudian los modelos sociales subrayan la importancia de la representaciones de papeles sin preparación alguna; esto dará como resultado que los estudiantes den un mejor rendimiento y saquen más provecho si reflexionan sobre sus opiniones y actitudes antes de encarar los papeles. La organización por parte del maestro indica que esta técnica persigue una finalidad bien definida y que producirá un aprendizaje determinado a pesar de que se realice en forma agradable para todos los participantes.

2.3.3 Enseñanza de procesos grupales.

Además de la representación de papeles, hay otras experiencias de grupo que sirven para favorecer el desarrollo social, como serían: la pertenencia a un grupo constructivo y los hábitos de cooperación. El trabajo en grupo en la escuela aporta grandes ventajas en cuanto al aspecto académico y social, con tal que los maestros estén dispuestos a dedicarle parte de su tiempo de clases a enseñar estos procesos. La capacidad de participar con buenos resultados en un grupo no es una cualidad innata: hay que aprenderla.

Si se quiere que el aprendizaje en grupo cumpla su cometido, el maestro debe organizar las cosas bien y enseñar la destreza poco a poco, como cualquier otra.

2.3.4 Procesamiento de información.

El tercer modelo de enseñanza propuesto por Weil y Joyce es del tipo de procesamiento de información. En él se pone de relieve el dominio de los datos y conceptos de la signatura con objeto de favorecer el desarrollo intelectual general y una comprensión global, algunos de estos modelos son los siguientes:

- Organizador anticipado. Este modelo sugiere que el maestro exponga el concepto o conceptos fundamentales primero y que después ayude a los estudiantes a organizar la información en torno al organizador por anticipado. Esta modalidad de aprendizaje a veces es objeto de críticas por considerársele aburrida y mecanista.

- Instrucción por medio de la investigación personal. Aquellos para quienes el desarrollo intelectual constituye la esencia de la educación escolar, sostienen un tipo de enseñanza que difiere del modelo de los organizadores por anticipado. Suchman⁸ destaca la necesidad de que los alumnos procesen la información de modo independiente y activo. Se da la prioridad al proceso ideacional; la meta de la instrucción es entonces ayudar al sujeto a pensar con mayor claridad. En vez de iniciar la enseñanza con el concepto, se procura lograr que los estudiantes adquieran conceptos afines.

- Modelo de instrucción directa. Este modelo ha ido adquiriendo gran difusión. Ha nacido de varios estudios correlacionales realizados últimamente, cuya finalidad es averiguar en que se distingue la conducta escolar de los maestros más y menos eficientes. Se caracteriza por centrarse en las metas

⁸ Suchman, J. The elementary school training program in scientific inquiry. Report to the U.S.

académicas, abarcar contenidos muy amplios, vigilar el adelanto de la clase, tener retroalimentación y orientar hacia las tareas todo el ambiente.

2.4 Enseñanza programada.

Este tipo de enseñanza fue creado por B.F. Skinner³, se basa en la siguiente hipótesis: con cada refuerzo de la secuencia estímulo-respuesta aumenta la probabilidad de que esa secuencia se repita después. Y a la inversa, cada vez que se repite sin recibir refuerzo disminuiría la probabilidad de que reaparezca. En consecuencia, la fuerza de un hábito puede expresarse matemáticamente con fórmulas que toman en cuenta el número de veces que se realizó y se reforzó y el número de veces que se repitió sin recibir refuerzo.

Skinner concibe el refuerzo contingente como mecanismo básico que explica la conducta. Define el refuerzo como todos aquellos que incrementan la probabilidad de una reacción.

El principio de este tipo de enseñanza consiste en proponer a los alumnos ciertas secuencias, en las que los conocimientos eran suministrados según un plan minuciosamente establecido. Se trata de <<obtener de los sujetos un cierto número de comportamientos, muy concretamente descritos como comportamientos directamente observables; los procedimientos que permiten alcanzar dicho resultado presuponen la fragmentación del conocimiento a adquirir en elementos simples, paso a paso; la actividad del sujeto es continuamente estimulada por la necesidad de dar una respuesta reforzada cada vez>>.

Para alcanzar este objetivo, es necesario, ante todo, proceder a un fino análisis de los contenidos propuestos, definiendo una estrategia precisa, una jerarquía de nociones y la naturaleza de las respuestas correctas. En muchas aplicaciones de este tipo, los aprendices deben trabajar solos, con ayuda de un cuaderno especial; según las respuestas dadas, buenas o malas.

³ Skinner, B.F. The technology of teaching. New York: Appleton Century-Crofts, 1968

CAPÍTULO 3

Incorporación de las computadoras a la educación.

**Cuando el hombre abandona la envidia
Empieza a prepararse para entrar en el
Camino de la dicha.**

E. Wallace Stevens.

CAPÍTULO 3

Incorporación de las computadoras a la educación.

3.1 El ciclo educativo.

Todo usuario lleva a cabo una evaluación subjetiva del software que emplea. Declara que los programas son buenos o malos, interesantes o molestos, útiles o superfluos, según criterios que habitualmente no se toma la molestia de explicar.

Pero los profesores y los padres que deben escoger un software para los niños piden informaciones cada vez más precisas y observaciones rigurosas. Un programa didáctico es un objeto que se compra.

Teóricamente un programa didáctico debería ser el resultado de un estudio y de un cierto trabajo de concepción que tome en cuenta obligatoriamente distintos aspectos. Algunos conciernen a la pedagogía y las ciencias de la educación, otros, a la psicología, otros más, a la ergonomía, y otras, por último, a la computación e informática misma. La tarea está lejos de ser estrictamente computacional e informática.

La primera serie de problemas afecta a los objetivos del programa. Como toda actividad educativa, es importante que estén formulados y definidos con precisión. ¿Quiénes son los usuarios previstos (edad, nivel escolar)? ¿Está destinado a la enseñanza o a la evaluación? ¿Que conocimientos debe aportar? ¿Que evaluación debe suministrar?

La segunda serie de cuestiones tiene que ver con los medios pedagógicos de que es necesario disponer para alcanzar el resultado previsto. Se trata, esencialmente, de definir la estrategia pedagógica. Ésta se basa en dos elementos, la experiencia y los conocimientos de los creadores del programa, que determinarán en la práctica cada elección: jerarquía, formato y desarrollo espacio-temporal de los contenidos a presentar, tipos de preguntas, naturaleza de los refuerzos y de las eventuales recompensas, medios de evaluación y autoevaluación que suministren al profesor, al alumno, etc.

Una construcción de este tipo no puede si no reposar en buena parte sobre un conocimiento real de las reglas del aprendizaje, en el sentido psicológico del

término. Tal conocimiento implica por igual a los mecanismos de adquisición y a los factores del entorno que pudieran hacerlos variar.

Por último, esta estrategia pedagógica no podrá ser desarrollada más que en relación con los objetivos implicados, pero teniendo en cuenta también las limitaciones materiales. EL conjunto debe quedar constituido, por tanto, como una forma muy particular de lección, curso, ejercicio o control, en donde la computadora sustituirá al profesor y suministrará el soporte material para la emisión y recepción de información. Habrá que tener entonces muy en cuenta las limitaciones. Porque, en la actualidad ninguna computadora habla ni entiende, ni se comporta realmente como un profesor.

Los estudios sobre desarrollo cognoscitivo y la psicología del aprendizaje han permitido establecer un cierto número de principios¹⁰. No obstante, el hecho de que tales principios pueden ser traducidos en reglas aplicables a la concepción de los programas didácticos.

Un aprendizaje supone un cambio duradero en las estructuras cognoscitivas del sujeto. Este cambio es una construcción activa e individual, que consiste en una integración de elementos nuevos en la estructura o estructuras previamente existentes. Podría hablarse, en términos Piagetanos, de acomodación de los esquemas o de equilibración de las estructuras cognoscitivas. En términos más modernos, hablaremos de modelos preexistentes en el sujeto, *modificados por los procesos de aprendizaje y reconstituidos, bajo la forma de modelos nuevos, al término de la sesión.*

En la medida que substituye al docente, el programa didáctico debe poseer u obtener, de un modo u otro, cierta información sobre los conocimientos, estructuras y modelos del usuario una vez que haya finalizado su aprendizaje.

Se puede contemplar la posibilidad de conformarse con dar al usuario algunas indicaciones generales sobre el nivel escolar, como <<programa destinado a los alumnos de CM1>>, pero este tipo de instrucciones globales se muestran insuficientes o poco adaptadas en ciertos casos o con determinadas personas. Sería entonces francamente útil un pequeño examen, que ante la

¹⁰ El objetivo de este tema no es tratar la muy amplia cuestión de educación, pero si se desea se pueden consultar algunas contribuciones como las de Le Ny(1967), Oléron(1964), Piaget e Inhelder(1963) o Gréco(1963)

eventualidad de un fracaso podría redondearse con una recomendación de estudio.

Un tutorial bien concebido debería empezar por establecer un balance mínimo. Sería deseable que mediante un cuestionario o un sondeo adecuado, pudiera certificar con seguridad si el alumno dispone de los conocimientos previos indispensables para sacar debido provecho de los conocimientos que se le van a proponer.

Técnicamente, el software para la enseñanza debe asimismo controlar en entorno educativo, al objeto de crear y mantener -en la medida en que le afecten- las condiciones favorables para el aprendizaje. Se conocen algunos puntos de capital importancia. EL mantenimiento de la motivación es un rasgo dominante. El feed-back es necesario: una información sobre la calidad de las respuestas que se devuelve al sujeto para asegurar que todo va bien. En caso de error resulta indispensable proponer correcciones. No basta, evidentemente, con indicar que la respuesta es incorrecta: hay que precisar por qué. Eventualmente será necesario <<auapar>> al sujeto a una etapa anterior, para que reciba ahí ciertos elementos o demostraciones.

Los refuerzos y las recompensas deben ser controlados de forma circunstancial. En buena parte de los casos será suficiente indicar que la respuesta es correcta.

Es necesario, por último, prever estructuras de control del desarrollo temporal de las sesiones. Parece conveniente informar al usuario del punto en que se halla, en relación con la dimensión del módulo en curso de ejecución, y prever, si el aprendizaje es un tanto largo, algunos puntos de reestructuración de los conocimientos supuestamente adquiridos, incluyendo un examen.

Dicho programa se presentará entonces como el reflejo del análisis psicopedagógico efectuado con anterioridad y de la estrategia que se haya determinado. No se tratará más que de la traducción informática de un organigrama que podría haber sido escrito en términos corrientes. Por último mencionaremos las reglas que debe seguir un programa didáctico para que cumpla su objetivo.

3.1.1 Reglas de Pedagogía general.

Dichas reglas se aplican a cualquier forma de transmisión de conocimiento de tipo escolar.

Contenidos y conocimientos.

- **Pertinencia de la información:** no dar mas informaciones que aquellas que son pertinentes y útiles en relación con los objetivos anunciados.
- **Exactitud:** ausencia de errores, de faltas (incluyendo las faltas de ortografía).
- **Adaptación:** ausencia de cualquier otro requisito previo que no haya sido explícitamente declarado; nivel de dificultad de las nociones en relación con el nivel escolar anunciado.
- **Jerarquía** respeto de la jerarquía de los conocimientos en el ámbito considerado.

Presentación de contenidos (textos)

- **Vocabulario** seleccionado en función del nivel escolar anunciado o de la edad de los sujetos. En particular, se recomienda utilizar para los niños palabras vivas, precisas, concretas y cortas.
- **Control de la longitud de las frases:** es bien sabido que las frases cortas son, en general, mejor descifradas y, en consecuencia, mejor asimiladas que las largas.
- **Control de la estructura de las frases:** se preferirán las estructuras facilitadoras. Si la frase no es muy larga, el uso de subordinadas e insertos favorece una memorización lógica. Las estructuras enumerativas parecen ser más difíciles de retener.
- **Selección de ejemplos.**

Presentación de contenidos (gráficas).

- **Pertinencia** del uso de la imagen para el argumento o tema de que se trate.
- **Concordancia** texto-imagen, de manera ordenada y conforme con el objetivo.
- **Facilitación de la comprensión.** Los gráficos y esquemas tienen generalmente la misión de aportar un complemento con relación al texto.

3.1.2 Reglas de aprendizaje.

Técnicas de aprendizaje.

- Orden de presentación: progresión adaptada, empezando por contenidos o ejercicios fáciles que conduzcan a éxitos iniciales que, a su vez, contribuyan a incrementar la motivación. Es una estrategia muy conocida de los juegos de arcade, de la que los programas didácticos no tienen por que prescindir.
- Momentos de actualización y de rememoración: de conocimientos antiguos y requisitos previos que deben ser revitalizados; de conocimientos adquiridos en fecha reciente.
- Momentos de reestructuración o de síntesis integradora de conocimientos antiguos y nuevos.
- Control de la duración de las lecciones o sesiones. Se acostumbra considerar que la duración de las clases o lecciones en las que el maestro o profesor se halla presente puede oscilar entre 20 y 40 minutos, en función de la edad, el nivel escolar, el tema, etc.

Formulación de preguntas.

- Oportunidad de las preguntas: No se formularán más que cuando existan garantías de que el alumno dispone de medios para contestar.
- Calidad de las preguntas: Las preguntas pueden ser más o menos elaboradas: desde simples solicitudes mnemónicas, de tipo <<elección múltiple>>, hasta auténticas situaciones de resolución de problemas que fomenten las operaciones cognoscitivas en vez de la simple memorización.
- Variedad de cuestionarios y ejercicios, con objeto de no tener que volver a ellos en caso de utilización repetida.
- Redacción de las preguntas. Un único problema por pregunta: evitar las informaciones innecesarias y los giros negativos; respuestas independientes unas de otras.

Feed Back y refuerzo de las respuestas correctas.

- Naturaleza y adecuación a la edad, nivel y simulación. La indicación de que la respuesta es exacta debería ser suficiente en la mayor parte de los casos. Cualquier exceso - basándose en parpadeos luminosos de la pantalla y música triunfal, por ejemplo - corre el riesgo de distraer la atención del sujeto.
- Variación de los refuerzos. La monótona repetición de una rutina que muestre <<exacto>> <<O.K.>> puede llegar a hacerse molesta, perdiendo su eficacia como refuerzo.
- Verificar que el sistema de control de las respuestas no deje pasar errores.

Feed Back y refuerzo de las respuestas incorrectas.

- Naturaleza y adecuación a la edad, nivel y situación. La simple indicación de que la respuesta es incorrecta no puede ser suficiente. Cualquier exceso - basándose en parpadeos luminosos de la pantalla y música siniestra, por ejemplo - puede resultar de mal gusto.
- Posición de los refuerzos: inmediatamente después de cada respuesta.
- Verificar que el sistema de control de las respuestas no clasifique como errores ciertas respuestas atípicas, pero correctas.
- Calidad de feed back: no hay que conformarse con la simple ejecución de una rutina que anuncie sencillamente <<error>>.
- Naturaleza de los procedimientos propuestos tras un error:
 - Indicación de la respuesta correcta;
 - Proposición de otra pregunta equivalente;
 - Aportación de ayuda parcial o indicios facilitadores;
 - Reenvío a la parte del programa didáctico en que se encuentran los conocimientos necesarios;
 - Presentación de los conocimientos bajo otra forma antes de proceder a formular de nuevo la pregunta.

Control de derivaciones y ramificaciones (unidad de evaluación y decisión).

- Naturaleza de las informaciones consideradas (relativas a los resultados obtenidos por el alumno).
- Cantidad de información tratada.
- Diversidad de las posibilidades de invertir la acción.
- Flexibilidad (y, por tanto, personalización) de las decisiones.
- Estructura general del programa didáctico:
 - La peor: secuencial jerárquica absoluta sin saltos ni retornos;
 - La mejor, probablemente: reticular (con más o menos arborescencias o jerarquías según los casos o los contenidos).

3.2 Evaluación de costos y beneficios.

A medida que se incrementa nuestra experiencia en el desarrollo de metodologías educativas sustentadas en la computación, más clara es para nosotros la relación entre libros y programas que operan en computadora, hasta decir que los programas de cómputo están dando lugar a un nuevo tipo de libro del futuro: el libro electrónico.

Esto nos hace pensar que la computadora, como el libro es su tiempo, producirá una profunda revolución educativa. Existen razones para pensar así, ya que con esta nueva metodología educativa es posible no sólo presentar imágenes en movimiento en lugar de los diagramas estáticos de los libros, sino también permitir al estudiante interactuar activamente con los programas, en lugar de desempeñar el papel pasivo del lector o espectador en que los han colocado los libros y, más recientemente, el cine y la televisión.

Este es pues, el más significativo beneficio que se está obteniendo al incorporar las computadoras en las diversas formas de educación. El camino aun es largo, ya que en México, no es explotado esta nueva tecnología educativa.

Existen varios ejemplos de grandes sociedades que con gran empuje han impulsado el uso de las computadoras en los procesos educativos. Conocemos los esfuerzos en Francia e Inglaterra y de los proyectos canadienses. Sabemos también que existen otros proyectos en Sudamérica (un caso el de Colombia) y en México.

Reconocemos que todos estos proyectos se enfrentan a críticas y aun a problemas serios. No pueden ser de otro modo, si como hemos dicho, la asimilación de la computación en los sistemas educativos fue es y será un proceso de años.

¿Cuales son estos problemas? ¿Cuales sus causas y manifestaciones? La respuesta es sencilla: si no existe ni experiencia ni metodologías debidamente probadas ni profesores capaces de aplicarlas ni forma de evaluar resultados, ¿Cómo es posible enfrascarse en programas de grandes dimensiones? No cabe duda que las razones son en algunos casos económicas, motivadas por los fabricantes de equipos que buscan en el sistema educativo salidas para sus excedentes; en otros más son de origen político, con tintes demagógicos.

Igual que en la educación básica y en la educación media, la computadora personal debe tener un papel muy importante en los años siguientes.

En este caso, su utilización en la conformación de laboratorios de experimentación en física, química, biología, ecología, economía, administración y ciencias sociales, representa una posibilidad con consecuencias previsibles.

Dentro de todo lo anterior reflejar de una manera somera pero explicativa los costos y beneficios que se observan al llevar a cabo este proyecto educativo.

- Los costos de adquisición, operación y mantenimiento de laboratorios reales son cada día más altos, y el beneficio que se obtiene es tan bajo que las instituciones de educación superior difícilmente los usan, no obstante se inclusión formal en los planes de estudio.
- Existen limitaciones de otros tipos que hacen difícil realizar experimentos a nivel que los programas de estudio lo requieran. Por ejemplo, la observación del movimiento de un planeta en su órbita puede requerir meses o años en un experimento de astrofísica; el estudio del crecimiento de una planta es también un fenómeno de meses; por el contrario, los fenómenos ondulatorios y de vibraciones se realizan en centésimas o milésimas de segundo, mientras que otros fenómenos son demasiado pequeños.

En todos estos casos, el empleo de computadoras plantea la posibilidad de un nuevo tipo de laboratorios de gran versatilidad y bajo costo, que si bien estarán limitados en sus primeras versiones, sufrirá un acelerado proceso de perfeccionamiento en los años siguientes.

Otro campo digno de mencionarse es la educación abierta, en la que podemos prever sistemas de instrucción personalizada en la que sea factible que los sistemas monitoreen y evalúen el avance del estudiante.

El uso de la computadora como herramienta de cálculo o , en forma más general, como instrumento de trabajo es algo que ya ha empezado a suceder. En los últimos años, hemos observado la aparición de herramientas de trabajo profesional cada vez más ambiciosas y fáciles de usar. Desde programas de diseño ingenieril, sistemas expertos para determinar riesgos y programas que facilitan la organización del conocimiento. Considero vital para las instituciones de educación superior la asimilación de estas tecnologías y la participación activa en su desarrollo.

CAPÍTULO 4

Multimedia una nueva herramienta.

**Si en la vida no hacemos locuras,
No pasa nada interesante..**

Jean-Paul Belmondo.

CAPÍTULO 4

Multimedia una nueva herramienta.

4.1 Breve historia de la tecnología Multimedia.

El término multimedia apareció ya en los años 60 y 70 en el área de pedagogía. Bajo el mismo se agrupaban los nuevos medios de apoyo al proceso de aprendizaje en clases.

La revista norteamericana MPC-World encontró raíces de la multimedia en el año 1500 A.C.: La primera presentación multimedia fue, por tanto, la entrega de los diez mandamientos a Moisés. Voces humanas y celestiales, trompetas, truenos, relámpagos constituían los componentes multimedia de esa época.

El término multimedia fue utilizado para descubrir un paquete de información presentada como una combinación de formas tales como el texto, los audiocasetes y los videocasetes. En los años 90 con la llegada de las computadoras personales, las cuales poseían capacidades de audio y vídeo, la palabra multimedia lleva a una nuevo y más completo significado, debido a que las computadoras hacen posible el acceso y manejo interactivo de una amplia variedad de medios.

De pronto, la tecnología ofrece el poder de integrar y guardar elementos sin relación tomados de diferentes fuentes como son: vídeo, movimientos de imágenes, animación, audio, efectos de sonido, música, narración, discursos, texto y gráficos, para significar algo, en un medio único de almacenamiento.

4.2 ¿Qué es la tecnología multimedia?

Multimedia(Más de un medio): Literalmente su significado sería muchos medios. Cuando aplicamos esta tecnología sobre las computadoras, esto quiere decir alguna combinación de texto, sonido, imágenes estáticas, imágenes en movimiento, todas ellas deliberadamente puestas bajo el control de una computadora.

Multimedia es un recurso poderoso que ofrece nuevas formas de explotar la información para hacer más factible y animado el manejo de la misma. Así es como la mezcla de los elementos anteriores, crean un ambiente multisensorial que

permite a los usuarios de esta tecnología participar de la información almacenada en la computadora en forma clara, intuitiva y espontánea debido a que llega a la mente a través de los sentidos.

Debido a que esta tecnología es de reciente creación y uso, se ha creado un estándar para el uso de la misma, este fue llamado MPC (Multimedia Personal Computer), el cual especifica los requerimientos de software y hardware para la aplicación Multimedia en ambiente MS-DOS. Este estándar incluye requerimientos de memoria, adaptador de memoria, sistema operativo, características de CD-ROM, etc.

La computadora que se use como base para convertirla en una central de control debe contar como mínimo con un procesador 80386SX con una velocidad de 25 MHz, con un mínimo de 2 MB de memoria RAM y 30 MB de disco duro. Además de los requisitos de Hardware, se necesita un software multimedia para poder combinar entre sí e integrar los distintos componentes individuales como el video, la música y la animación, así como gráficos y textos.

Como la PC se convierte en la central de dirección y control, pueden manejarse a través del mismo diferentes dispositivos electrónicos del área de la comunicación y del consumo. Aquí también es el tipo de aplicación lo que define si se utilizara un equipo doméstico de audio, un reproductor y una cámara de video o simplemente las posibilidades de PC multimedia.

Más problemático (y más caro) resulta cuando se quieren integrar todas las posibilidades. Guardar secuencias de video en la computadora o mezclarlas con otras fuentes requieren de un equipamiento complejo y caro. Es por eso que a mi punto de vista, el equipo base que sugiero que se utilice es un procesador 80486DX4 a 100 MHz, con 8 MB de memoria RAM y un disco duro de 512 MB.

4.3 ¿Qué ofrece la tecnología multimedia?

Como ya se mencionó multimedia es más que una simple combinación de texto, sonido e imágenes. La unión de estos elementos combinadas con rutinas de animación auxiliados con interfazs amigables como el Mouse y los monitores sensibles al tacto, da la pauta para la creación de un nuevo tipo de software, un software que vaya más allá de lo tradicional.

En contraste con los modelos tradicionales basados en ejercicio, práctica, preguntas de opción múltiple o textos para llenar espacio, entre otros, multimedia

permite crear modelos más imaginativos como: juegos de aventura interactivo, viajes animados, libros electrónicos, sistemas de referencia entrelazados, por citar algunos los más importantes, que permiten al estudiante tener preguntas que el mismo se plantea más que presentar un diagrama de un proceso, proporciona tutoriales, presenta simulaciones animadas que retan a los estudiantes a descubrir los principios detrás de proceso y hacer sus propias conclusiones, así mismo, permite hacer conexiones entre conceptos que se conviertan en el fundamento de su creatividad y su juicio crítico, en vez de base de datos de hechos.

La tecnología brinda la oportunidad de comunicarnos de formas nuevas. En el centro de sistemas de multimedia están las imágenes interactivas particularmente de vídeo que traen una visión real del mundo en el dominio de la computadora. Dichas imágenes se encuentran llenas de color, pueden tener grandes cantidades de movimiento y pueden ser reproducidas en cualquier orden, a cualquier velocidad y en cualquier dirección.

Los sistemas multimedia pueden proporcionar a los estudiantes y maestros cualquier tipo de interacción; estos sistemas ayudarán a los estudiantes a ir al centro de la información, de las ideas de los conceptos más rápidamente; las imágenes visuales proporcionan ejemplos concretos que los llevara a un mejor entendimiento. Todo esto ayudara a los maestros y estudiantes a mostrar sus ideas más claramente, de esta forma menos tiempo será tomado para tratar de entender el significado en el concepto y mayor tiempo será tomado en la exploración de nuevas y muchas situaciones.

4.4 Ventajas que ofrece la tecnología multimedia

La tecnología multimedia se está convirtiendo en una herramienta necesaria en los negocios, la milicia y la educación. Esta parte únicamente se enfocará en los beneficios de su uso.

Numerosos estudios de investigación en multimedia instruccional, han estado publicándose durante los últimos 10 años. Sin embargo el campo es bastante complejo y diverso, lo cual no nos permite hacer una generalización. Por lo que intentaremos describir los principales beneficios que esta tecnología ha aportado en el campo de la instrucción.

- *Auto confianza.* Si existe o no una diferencia en la eficiencia lograda a través de la instrucción multimedia y la instrucción convencional, si hay una substancial evidencia que los estudiantes en los cuales se ha utilizado este tipo de instrucción, han aumentado el dominio en dicha materia. Esto nos dice que

la percepción de los alumnos, puede ser fuertemente motivada y así disminuir su falta de interés.

- *Reducción del tiempo de instrucción.* Estudios del tiempo requerido para la instrucción de una persona con multimedia y programas basados en esta, han encontrado una reducción del 30% o más sobre la instrucción convencional.
- *Disminución del costo.* Aunque adoptar la instrucción a través de multimedia no es barato, el beneficio es a largo plazo, ya que se necesitan canalizar menos recursos, y se trabaja con mas personas, que en la instrucción convencional.
- *Aprendizaje activo.* Hablar de Multimedia es hablar de estudiantes más activos que pasivos, esto es gracias a que se esta aprendiendo a través de una interacción física y un enfoque cognoscitivo.
- *Libertad multisensorial.* Investigadores en estilos de aprendizaje indican que algunos estudiantes aprenden mejor a través de modalidades específicas, tales como visual, auditiva o kinestesica. Algunos estudiantes son aprendices visuales, algunos aprenden por medio de sonidos, algunos otros aprenden tocando. Multimedia provee instrucciones a través de múltiples canales sensoriales.
- *Libertad de Multilenguaje.* La instrucción multimedia puede incluir múltiples pistas de audio con una gran cantidad de lenguajes. Un mayor beneficio, es que con esto se puede introducir esta tecnología a una gran población.
- *Animación de la exploración.* Otra ventaja de la instrucción basada en programas multimedia, es que éstos pueden ser no lineales, esto quiere decir que los estudiantes pueden explorar el contenido desde múltiples perspectivas. Los programas multimedia que emplean esta metodología pueden ayudar a los estudiantes a desarrollar ricos modelos mentales como bases del aprendizaje futuro; crear ambientes que permitan sostener exploraciones por maestros y alumnos; ayudar a los alumnos a dominar ciertos temas desde diversas perspectivas; y desarrollar estructuras integradas de conocimiento que ayuden a los alumnos a transferir este a tareas mucho más complejas.
- *Motivación.* La tecnología puede inspirar a los alumnos, para que éstos realicen un aprendizaje interactivo y relevante. Estudios de investigación generalmente concuerdan en que los estudiantes son positivamente motivados a través de la

integración de la tecnología. El hecho de que los estudiantes disfruten trabajar con tecnología y, aun así permanezcan por largos periodos de tiempo, puede ser un elemento benéfico de gran valor.

- *Incremento de la Seguridad.* Muchos ambientes son demasiados peligrosos para programas de instrucción. En estos casos, éstos son frecuentemente muy posibles de diseñar en un programa multimedia, que proporciona a los alumnos la interacción con simulaciones del medio ambiente sin correr el riesgo que implicaría un entrenamiento real.

4.5 Desventajas de la tecnología multimedia.

Las desventajas que esta nueva tecnología, son realmente mínimas en comparación con los beneficios que se están obteniendo, sin embargo existen y son prácticamente de índole costo de infraestructura. Esto quiere decir que el equipo de cómputo que se requiere para poder implementar esta nueva forma de enseñanza es en ocasiones demasiado caro, este dependerá directamente del poder de la plataforma que deseemos utilizar.

Aun así, con el continuo desarrollo de la tecnología en cómputo los costos continuamente van en decremento, tal razón nos permite establecer que esta cuestión no representa una desventaja significativa.

4.6 ¿Por qué la tecnología multimedia?

La instrucción multimedia nos presenta nuevas oportunidades y retos para la instrucción. El uso apropiado de esta puede proveernos de una excelente avenida para la motivación, exploración y la instrucción en un multisensorial y diverso mundo.

Multimedia presenta un gran número de oportunidades para los profesores y desarrolladores profesionales, los cuales pueden navegar alrededor de los sistemas multimedia utilizando enlaces de varios tipos entre una pieza de información y otra. Lo importante es encontrar que enlaces, en qué contexto y qué método serán más apropiados para una situación particular de aprendizaje.

Gracias a esta tecnología los estudiantes tienen la oportunidad de aprender practicando, desarrollando así su capacidad creativa y de investigación. Como multimedia permite interactuar y experimentar, el aprendizaje es resultado de un proceso intuitivo basado en el uso de los sentidos, por lo que incrementan la efectividad de la enseñanza sobre los métodos tradicionales.

A través de la tecnología multimedia se puede controlar el espacio y el tiempo; el espacio, separado en dos elementos controlables la localización y el tamaño. De esta forma, el manejo de la localización con las imágenes interactivas significa el cambio de la posición en el espacio; las imágenes pueden situar instantáneamente a una persona en diversas partes del mundo si controla su localización, lo que quiere ver, el ángulo, etc.

CAPÍTULO 5

Ingeniería y diseño del software.

La imaginación es la loca de la casa.

Santa Teresa.

CAPÍTULO 5 Ingeniería y diseño del software.

5.1 La importancia del software.

Han pasado ya más de tres décadas de la aparición de la informática, y se ha enfrentado a muchos desafíos en cuanto al desarrollo de hardware para las computadoras se refiere, principalmente era el costo del procesamiento y almacenamiento de los datos. Esto fue superado en la década de los ochenta, gracias a la microelectrónica. Pero como consecuencia de este resultado, surgió un nuevo problemas, este es el de mejorar la calidad (y por ende el costo) del software utilizado.

En la actualidad la potencia de las grandes computadoras de los años ochenta, se tiene en una computadora personal, y en vísperas del siglo XXI, las perspectivas que se tienen en cuanto al poder de procesamiento, son tal vez inimaginables. Es debido a esto, que el software es el mecanismo que nos facilita utilizar y explotar todo este gran poder de procesamiento, en beneficio del ser humano.

Décadas atrás, era imposible imaginarse que las computadoras lograran realizar una gran diversidad de actividades relacionadas con el ser humano, tales como generar sonidos vocales, reconocer la vox humana, crear realidades que no existen, etc.

Es por eso que el software ha adquirido un lugar dentro de la actividad diaria del ser humano, y dentro de su avance técnico y tecnológico.

El software se encuentra actualmente en lo que los expertos han llamado la cuarta era, y en la que se identifican las siguientes características:

- Técnicas orientadas a objetos, las cuales ya casi han desplazado las formas tradicionales del desarrollo del software.

- Técnicas de cuarta generación para el desarrollo de software(que serán explicadas más adelante)

- Sistemas expertos y software de inteligencia artificial, las cuales ya no son un tema de ciencia-ficción, son una realidad que poco a poco va ir adquiriendo fuerza.

•El software de redes neuronales artificiales, el cual ha abierto excitantes posibilidades para el reconocimiento de formas y habilidades de procesamiento de información, de una forma muy parecida al ser humano.

Pero no son estas las únicas características de la cuarta generación, aunada a esta se siguen enfrentando una serie de problemas relativos al software:

•El gran avance del hardware, ha traído un desequilibrio, en cuanto a la capacidad de construir software que capaz de explotar al máximo el hardware.

•Inherente a esto, no somos capaces de producir el software suficiente que cubra las necesidades que existen.

•En el software que existe, no es posible darle mantenimiento, debido al mal diseño, y al mal uso de los recursos.

5.2 El software.

Definir software actualmente no es una tarea fácil, ya que actualmente, debemos entender que es lo que implica este. Podemos dar las siguientes definiciones, pero no cubrirían por completo el significado de software.

•“Software instrucciones(programas de computadora) que cuando se ejecutan proporcionan la función y el comportamiento deseado”

•“Estructuras de datos que facilitan a los programas manipular adecuadamente la información”

•“Documentos que describen la operación u el uso de los programas” Como podemos leer, las definiciones anteriores, no nos ofrecen un verdadera descripción de lo que significa software, es por eso que debemos enunciar las características que este debe tener para que se le considere como tal.

5.2.1 El software se desarrolla.

Aquí cabría la confusión de que el desarrollo del software y el hardware tienen una gran similitud, pero ambas actividades difieren en gran medida. Aunque una base primordial para ambas es el buen diseño, en el software se tiene la facilidad de corrección en cuanto a la calidad se refiere.

Las personas que se dedican al desarrollo del software, utilizar diferentes métodos para la construcción de un software de calidad. Esto nos da como

resultado que el costo del software se encuentra en la ingeniería, es decir no podemos ver los proyectos de software como un proyecto de fabricación.

5.2.2 El software no se "estropea".

Al hablar de software, hablamos de una sola etapa en donde se presentan cualquier cantidad de fallos los cuales producirán un mal funcionamiento del programa creado. Sin embargo al detectar y corregir los errores el software funcionara óptimamente hasta llegar a la obsolescencia.

Sin embargo la idea anterior se queda muy lejos de la realidad, ya que cada vez que se corrigen los errores detectados, estos en ocasiones generan nuevos errores, los cuales nos llevan a caer en el mismo ciclo una y otra vez. La consecuencia final de este círculo vicioso es un deterioro que va a depender de la calidad del diseño que se halla realizado.

5.2.3. La mayoría del software se construye a la medida, en vez de ensamblar componentes existentes.

Si hablamos de diseño en torno a hardware, contamos con ciertos elementos los cuales podemos utilizar como un componente que realiza cierta función, y así ir diseñando nuestro elemento final.

En este sentido los diseñadores de software no tienen una lista que asemeje los componentes que se tienen en hardware. Esto debido a que el software se adquiere ya desarrollado, pero no puede ser utilizado como un componente auxiliar de otro software. Aunque se esta empezando a utilizar un concepto que se ha denominado "reusabilidad del software".

5.3 ¿Qué es ingeniería del Software?

No existe una definición que sea aceptada por la mayoría de los expertos, por ejemplo la siguiente definición fue propuesta por Fritz Bauder: *"El establecimiento y uso de principios sólidos de Ingeniería para poder obtener software económico que sea rentable y trabaje eficientemente en máquinas reales."*

Pero en este sentido han sido demasiadas las definiciones que entorno a la *ingeniería del software* se han propuesto, pero todas tienen en común el uso de elementos propios de la ingeniería.

Debemos decir que la ingeniería del Software es el resultado de la integración del Hardware y de los sistemas de ingeniería. Este se compone de tres elementos que son:

- Métodos. Este nos proporciona las técnicas a través de las cuales podemos construir software. Aquí se contemplan los siguientes puntos: planeación y estimación del proyecto, análisis de requerimientos del sistema y del software, diseño de las estructuras de datos, arquitectura del programa y algoritmos de procedimiento, codificación, pruebas y mantenimiento. Además de que incluyen lenguaje especial orientado o notación gráfica, así como criterio para la obtención de software de calidad.

- Herramientas. Actualmente podemos encontrar una gran diversidad de herramientas para el diseño de software, los cuales nos pueden proporcionar métodos automatizados o semiautomatizados. Cuando las herramientas pueden integrar información que ha sido creada por alguna otra herramienta, podemos llamarla *Ingeniería de software asistida por computadora* o CASE (*Computer Aided Software Engineering*). Este tipo de herramienta combina hardware, software y bases de datos de ingeniería de software, para crear en ambiente de la ingeniería de software.

- Procedimientos. Son los que se encargan de agrupar los métodos y las herramientas, los procedimientos definen la secuencia en la cual los métodos serán aplicados, la información (documentos, reportes, formas, etc.) que es requerida, el control que ayudara a obtener una calidad.

La ingeniería de Software comprende un conjunto de pasos que abarca métodos, herramientas y procedimientos, de los cuales sólo aplicamos de dos en particular. Estos pasos son frecuentemente llamados Paradigmas de la ingeniería de Software. Cada paradigma es elegido basándose en la naturaleza del proyecto y la aplicación.

5.3.1 Ciclo de vida

EL ciclo de vida es un paradigma clásico dentro de la ingeniería del software. Es también conocido como el modelo en cascada. El cual se muestra en la siguiente figura. Como podemos observar se va obteniendo una aproximación secuencial y progresiva del desarrollo del software.

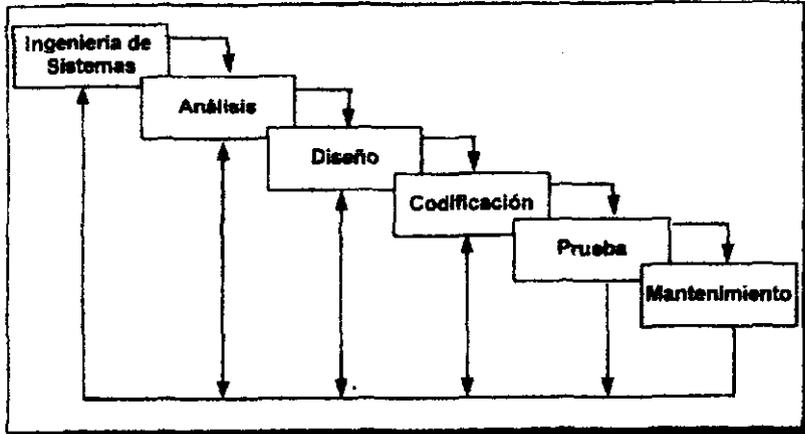


Figura 1.6 Ciclo de Vida Clásico

Aquí podemos ver las etapas que generalmente se consideran en la mayoría de los paradigmas de la ingeniería del software. Los cuales explicaremos de una manera breve y concisa.

- Ingeniería y análisis de sistema: El software siempre será una parte de todo un sistema, es por eso que aquí hay que establecer todos los requerimientos para todos los elementos del sistema, ya que ésta estará interactuando con hardware, usuarios, bases de datos, etc.

- Análisis de requerimiento de Software: Aquí prácticamente se enfoca a los requerimientos de procesos y se enfoca a la especificidad del software, es decir aquí se debe entender cuál será la función real del software, que información manejará, delimita los requerimientos del sistema y de las funciones que se tendrán. Es aquí donde se documentan todos los elementos que se mencionaron, y se revisan las veces que sean necesarios.

- Diseño Es aquí donde entra la imaginación del Diseñador, ya que se hacen realidad todos los requerimientos del paso anterior en una representación que será el software. Este paso consta a su vez de cuatro actividades: *estructuración de los datos, la arquitectura del software, detalles de los procesos y la caracterización de la interfaz*. Al igual que los requerimientos, este paso se debe documentar ya que este pasará a ser una parte de la configuración del sistema.

- Codificación Una vez que se tiene el diseño, éste debe convertirse en toda una secuencia de instrucciones de máquina, y es precisamente en esta fase donde se realiza esto. Aquí hay una situación que depende del grado de detalle que se haga en el diseño, ya que entre mayor sea éste, la codificación será más sencilla y rápida.

•**Pruebas** Una vez que el código ha sido generado, se da paso a la etapa de pruebas. Estas se enfocan principalmente a la lógica interna del software. Para poder asegurar que no habrá falla en el funcionamiento externo, diremos que estamos de acuerdo con los resultados generados, y el software habrá cumplido su objetivo.

•**Mantenimiento** El mantenimiento se enfoca a cambios debido a que se han encontrado errores, debido a que el software debe adaptarse para acomodar sus cambios en un ambiente externo, o por que los compradores requieren cambios de funcionalidad o de apariencia. EL mantenimiento aplica cada uno de los procedimientos del ciclo de vida, a un programa existente, en vez de crear uno nuevo.

5.3.2 Construcción de prototipos.

La construcción de prototipos, tiene la característica, que da al programador una gran facilidad para construir el modelo del software que va a construir, dicho modelo, tomará alguna de las siguientes formas:

- Un prototipo en papel o un modelo basado en PC que describa la interacción hombre-máquina, de forma que facilite al usuario la comprensión de cómo se producirá tal interacción.
- 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.

La siguiente gráfica muestra cuál es la secuencia de eventos que se dan cuando se utiliza este paradigma. Al igual que en los demás paradigmas, se comienza por la recopilación de los requisitos, pero en este caso la participación del cliente es más directa con el desarrollador, ya que entre ambos definirán todas las relaciones que deberán existir, posteriormente, ya definidos, se produce un diseño rápido, el cual contempla principalmente, entradas, y salidas, en cuestión de formatos. Es precisamente este diseño, el cual nos lleva a la construcción del prototipos, el cual será evaluado por el cliente/usuario, y es a través de la repetición de este proceso, que se va refinando el software que se desea.

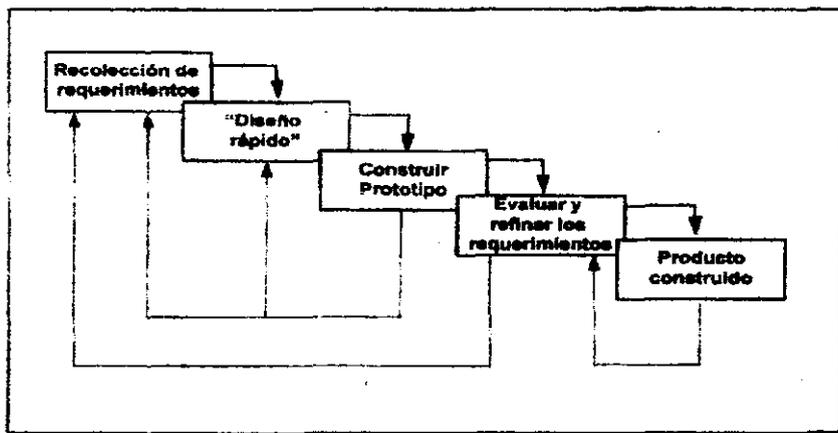


Figura 1.6 Construcción de Prototipos

Pero también aquí como los paradigmas que se están describiendo, existen dificultades, las cuales se resumen de la siguiente manera:

- Aun cuando el cliente ve funcionando el software, este fue hecho sin una estructura sólida, la cual no nos permite establecer la calidad o el mantenimiento a largo plazo.
- En ocasiones, el desarrollador no elige el sistema operativo, o el lenguaje de programación adecuados, debido a que establece tiempos de espera para que el cliente obtenga su producto deseado. Esto da como resultado un algoritmo ineficiente, pero que en primera instancia demuestra la capacidad que se tiene para desarrollar software.

Aun cuando los problemas que se describieron anteriormente, parecerían importantes, no lo son, ya que este paradigma, da muy buenos resultados en el desarrollo de software, lo único que hay que definir bien las reglas de la interacción entre el cliente y el desarrollador.

5.4 Técnicas de cuarta generación (T4G)

Esta técnica, denominada de "Cuarta generación", abarca una gran cantidad de herramientas de software, las cuales nos proporcionan una gran facilidad en el desarrollo de software, ya que la generación de código ya no consume demasiado tiempo, y no es tan compleja, ya que las herramientas

generan este automáticamente. Esto nos da una mayor rapidez en el desarrollo de software.

El paradigma T4G para la ingeniería del Software se orienta más hacia la posibilidad de especificar el software a un nivel más próximo al lenguaje natural o en una notación que proporcione funciones significativas. Pero esto, aun esta lejos de cumplirse, ya que existen pocos elementos que cumplan esto. Algunos de esos elementos son: lenguajes no procedimentales para consulta a bases de datos, generación de información, manipulación de datos, interacción y definición de pantallas, generación de código, etc.

EL diagrama de este paradigma es el siguiente.

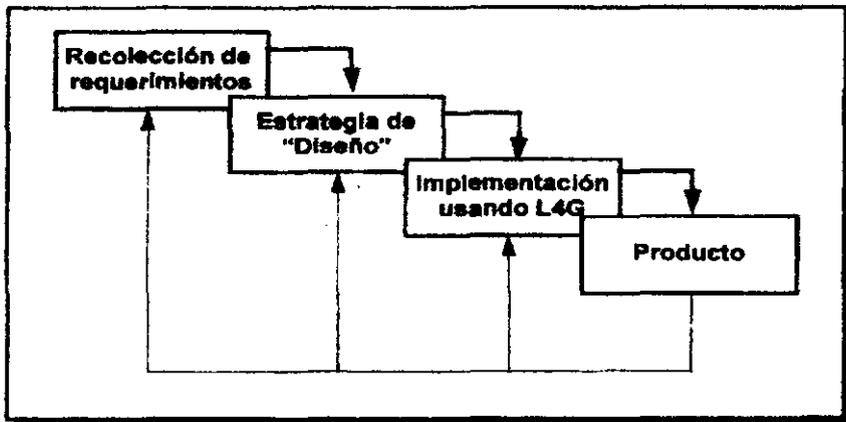


Figura 1.6 Técnicas de Cuarta Generación

Como podemos observar en número de procedimientos para la aplicación de este paradigma, ha disminuido considerablemente, ya que el cliente describe los requisitos, que son, a continuación, traducidos directamente a un prototipo operativo. Sin embargo, en la práctica no se puede hacer eso. Ya que en ocasiones el cliente no esta completamente seguro de lo que quiere, y hay que realizar modificaciones, las cuales son fáciles de realizar debido a las características que tienen los Lenguajes de 4ª Generación (L4G).

Esta técnica es muy recomendable para desarrollar sistemas pequeños, ya que podemos pasar directamente de la especificación de datos, a la implementación, usando un L4G no procedimental. En el caso de sistemas grandes, no se recomienda pasar a la implementación, ya que algunos L4G, necesitan basarse en un buen diseño.

Estas técnicas han ido adquiriendo gran importancia para el desarrollo de software en los campos de sistemas de información, y poco a poco han ganado terreno en la ingeniería y en la programación de tiempo real

Después de haber descrito las metodologías que se han usado en la construcción de sistemas computacionales, se ha decidido utilizar las ventajas que nos ofrecen las técnicas de cuarta generación y la construcción de prototipos.

5.5 Programación Orientada a Objetos y a Eventos

Dadas las características que debe reunir el sistema propuesto, debemos tener una visión clara del tipo de programación que debemos utilizar, ya que necesitamos una programación, que sea lo más flexible posible y que nos permita construir sistemas que sean aparentemente pequeños, pero que den gran confiabilidad, además de que permitan un mantenimiento fácil, y generen la capacidad de evolucionar, conforme a los requerimientos que se generen en un futuro. La Programación Orientada a Objetos(POO), es la que nos permite obtener más fácilmente, la mayoría de los requerimientos planteados, y por ende de lograr un éxito en la construcción del sistema que se quiere construir.

En la POO se manejan varios conceptos tales como, clase, objeto, mensaje, abstracción, encapsulado, polimorfismo, herencia, etc., Pero estos conceptos no pueden ser descritos como elementos simples, si no que deben ser agrupados para poder comprender su objetivo, el agrupamiento es el siguiente:

1. Objetos.
 - Abstracción
 - Encapsulado
2. Mensajes.
 - Reutilización del nombre
 - Polimorfismo
3. Clases.
 - Modularidad
 - Objetos Compuestos
 - Herencia

5.5.1 Objetos

Debemos visualizar los sistemas orientados a objetos, como una colección de dispositivos, los cuales mantienen una independencia entre sí, y tienen una comunicación entre ellos, sin olvidar que cada uno tiene su propia estructura interna. Cada elemento de la colección es denominado un objeto.

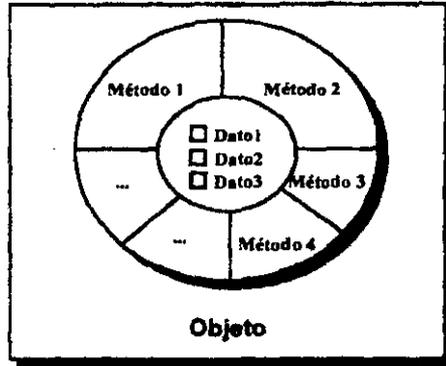
Así, un objeto, lo podemos definir como una entidad tangible del mundo en que vivimos, o en su caso de un concepto teórico. Estos objetos tienen un estado, un comportamiento y una identidad. De aquí podemos decir que un objeto puede ser: un auto, una moto, una bicicleta, una casa, etc.

Un Objeto se caracteriza por tener un estado (peso, posición, altura) y un comportamiento(aumentar velocidad, moverse de lugar), Realizar alguna modificación ya sea del estado o del comportamiento, da como resultado la ejecución de alguna acción (para el comportamiento), o el cambio de estado del objeto. Un objeto tiene identidad única por que todos los objetos son diferentes.

El estado de un objeto se modela a través de un conjunto de valores, los cuales son variables, representan la abstracción del objeto, así como también sus características en el tiempo, a las cuales denominaremos datos. Con respecto al comportamiento, este es modelado basándose en las funciones que realizan las tareas de modificar el estado del objeto, y se les conoce como métodos. Los datos y los métodos se manejan como un todo inseparable.

También existe un concepto llamado encapsulado, el cual maneja, que los datos internos son inaccesibles a cualquier otro objeto o programa.

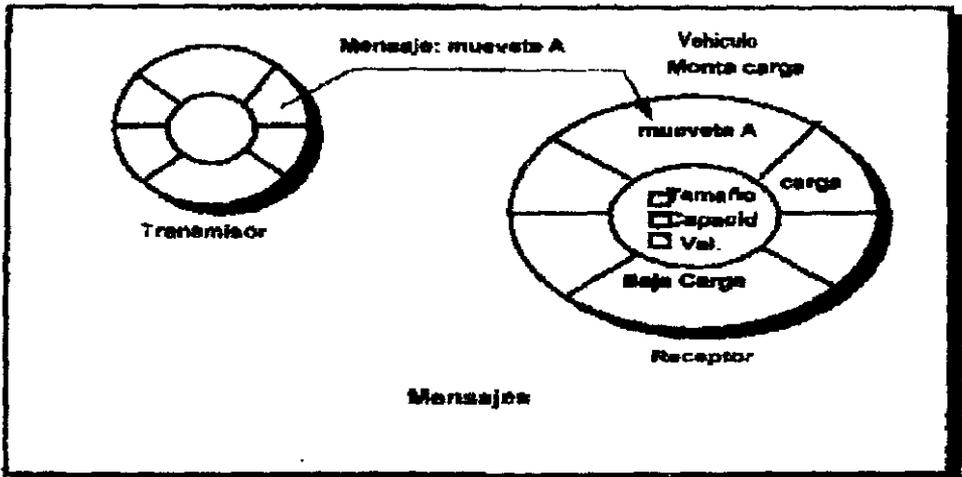
En la gráfica de la figura de la siguiente página podemos observar la protección de los métodos por parte del objeto de cualquier elemento externo. La única manera de poder modificar el estado del objeto es por medio de un método, es así que podemos mantener la integridad de los datos.



5.1 Representación de un objeto

5.5.2 Mensajes.

En la figura siguiente representamos la comunicación entre los objetos, los mensajes que utilizan los objetos para comunicarse, permiten modificar el comportamiento del objeto que recibe el mensaje, generando que este objeto realice cierta tarea o que cambie de estado. Por ejemplo, crear un nuevo objeto, preguntar por el valor de un dato, almacenar un valor, etc.



5.2 Representación de un mensaje

Aquí debemos mencionar que existe un concepto esencial en la POO, este es el polimorfismo, es cual indica que cada objeto ejerce una reacción muy particular a un mensaje recibido.

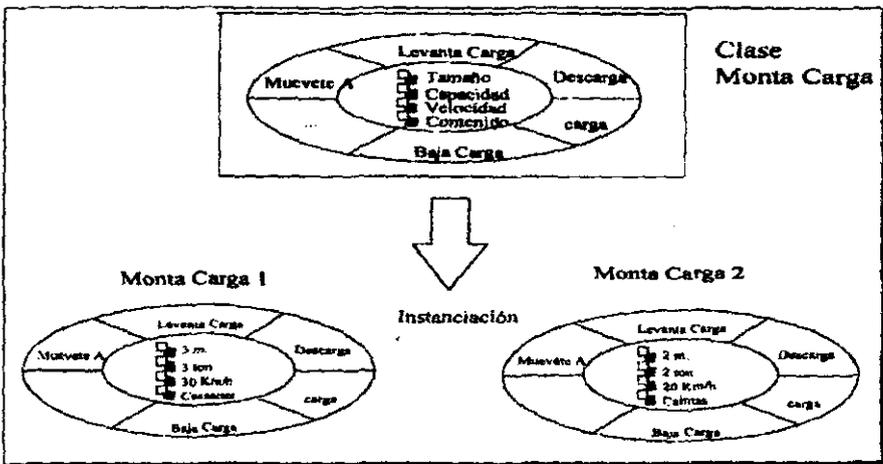
Por ejemplo: todos los animales, comen, pero todos tienen una manera muy particular de hacerlo. Esto nos permite decir que si reemplazamos el animal, este comerá al recibir el mismo mensaje que el animal anterior, es así que el flujo del programa no se verá alterado, esto nos permite tener algo más sencillo y un mantenimiento más fácil.

5.5.3 Clases.

Clase, es la definición abstracta de un grupo de objetos con datos y métodos comunes.

Los objetos son creados mediante las instancias de una clase. Mientras que un solo objeto es la identidad concreta que existe en el tiempo y el espacio, una clase sólo representa una abstracción, la escénica del objeto.

Una clase permite definir las propiedades y métodos que son parte de todas sus instancias, es decir son métodos que comparten dichas instancias, pero a su vez cada objeto posee sus propios valores, los cuales son únicos. esto lo podemos apreciar en la gráfica siguiente.



5.3 Ejemplo de la clase del monta carga

Como podemos observar existe una clase llamada montacargas, donde existen todos los datos con respecto a los montacargas, es decir todos los datos están siendo compartidos, pero cada montacargas tiene su propio tamaño, capacidad, velocidad máxima y contenido específico.

Esto no permite plantear que un objeto siempre es la instancia de una clase, pero la clase no siempre producirá objetos, en ocasiones sólo permitirán expresar características comunes a un grupo de clases, este tipo de clases se les denomina clase abstracta y adquiere importancia en la jerarquía de clases.

Las clases traen orden a los objetos organizándolos en jerarquías de clases –herencia- y en objetos compuestos –objetos que contienen otros objetos- estos conceptos se explicaran posteriormente.

5.5.4 Métodos de organización.

De acuerdo con Yourdon y Coad, los métodos de organización que existen son tres:

- 1.- **Identidad.** Identificar los objetos por medio de sus atributos (estado) y comportamiento. Por ejemplo un tigre.
- 2.- **Organización Parte De (Objetos Compuestos).** Descomponer al objeto en las partes que lo forman. Por ejemplo un tigre tiene, ojos, patas y cabeza.
- 3.- **Organización Tipo De (Jerarquía de Clases).** Identificar la relación que tiene con otros objetos de características semejantes. Por ejemplo el tigre pertenece a los felinos.

Los métodos anteriores, tienen gran aplicación en la POO, estos se aplican de la siguiente forma:

La identidad, esta descrita en el tema de objetos, donde la parte esencial, es la identificación y la definición de los objetos.

La organización en sus dos modalidades, están totalmente involucradas con el concepto de clases, es decir, los usamos para simplificar la apreciación del mundo físico.

Organización Parte de. Es usada cuando se pretende explicar a alguien menor, como esta compuesto algún objeto como una moto o un automóvil, *tratamos de describir sus componentes más esenciales, motor, llantas, volante, etc.* Tratando de explicar cada uno de éstos. Este tipo de objetos son llamados objetos compuestos.

Organización Tipo de. *Basándose en el ejemplo anterior, cuando se ejecuta la descripción, tomamos la referencia de algo para describir mejor el nuevo objeto, es decir, la descripción no empieza de la nada.* En el caso de la moto y el carro podemos utilizar una bicicleta, esto es el carro y la moto, son como la bicicleta, pero más grandes y utilizan un motor.

Las clases actúan como una plantilla agrupando objetos que guardan una relación entre si con respecto a las propiedades y comportamientos.

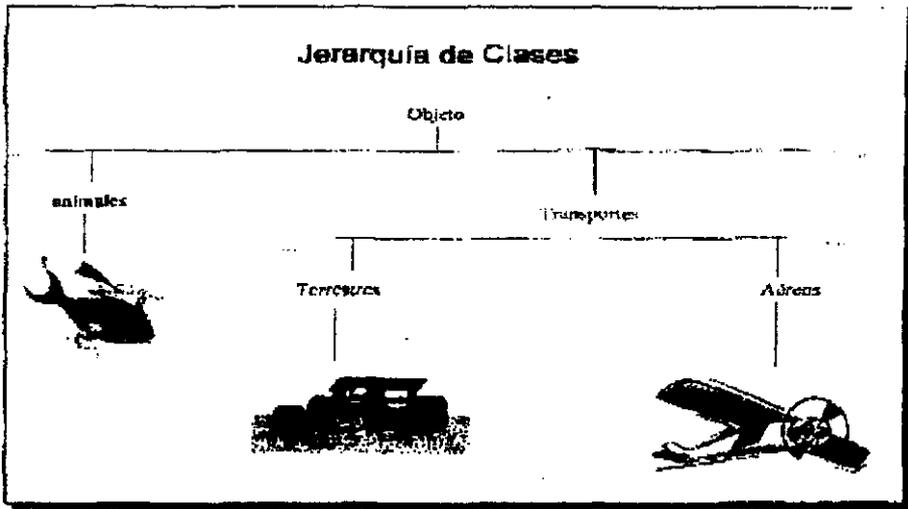
Dentro de la organización Tipo De, las clases son ordenadas a través de una forma jerárquica, es decir en una forma taxonómica de superclases(padres) y subclases (hijos). Es de aquí que se define la jerarquía que existe entre las clases.

Las subclases permiten la definición de objetos, con un grado de especialización mayor, es decir, conforme desciende la jerarquía, las subclases son más específicas. Por el contrario, conforme más nos encontremos en la raíz, las superclases son más generales.

5.5.5 Herencia.

En términos generales una subclase hereda todos los datos y métodos de la superclase a la que pertenece. La subclase puede adicionar datos y métodos heredados. *La figura de la siguiente hoja, muestra el concepto de herencia de clases y de herencia.* Todos los transportes tienen cierto tipo de combustible, una capacidad de carga y transportan bienes o personas de un lugar a otro. Todas las subclases de transporte heredan estas características.

Además de la rama de transportes, todos los transportes terrestres comparten algunas características comunes, de la misma manera que los hacen los transportes aéreos. Los transportes terrestres se mueven en tierra firme, mientras que los aéreos por aire.



5.4 Jerarquía de Clases

La herencia nos brinda una gran ventaja, para este caso, si quisiéramos agregar los transportes marítimos, no tendríamos que crear todas características de los transportes marítimos, sólo tenemos que tomar las características de los transportes, y agregar las características de los transportes marítimos, es decir que se desplazan por el agua, etc.

La herencia nos da la principal característica de la POO, la reutilización del código. A final de cuentas la POO nos permite ver la programación como un manejo natural de las cosas que nos rodean (como objetos con características, relacionados entre si).

5.5.6 Programación Orientada a Eventos.

La programación orientada a eventos, tiene sus bases en la POO. Los objetos son lo que comúnmente se usan en la interfaz Windows, casi todos los elementos gráficos, son objetos, las ventanas, los botones, los botones radiales, las cajas de lista, las líneas de edición, los textos estáticos, etc. Pero si bien sabemos, estos elementos realizan algún procedimiento, o cambian su estado dependiendo de que se le haga (un click con el Mouse, presionar algún botón del Mouse y arrastrarlo, modificar algún texto, etc), es precisamente este método asociado al que se le denomina evento, y aquí podemos encontrar, la destrucción de un objeto, la modificación de algún texto, etc.

Como podemos percatarnos, la programación orientada a eventos es de fácil comprensión, ya que todos los usuarios de Windows están haciendo uso de ésta, además de que podemos realizar una analogía entre la programación orientada a eventos, y la POO para su mejor explicación y funcionamiento.

Debemos señalar que la programación orientada a eventos, es muy importante en el diseño y producción del sistema propuesto. Así el sistema estará basado grandemente en este tipo de programación.

CAPÍTULO 6

Breve descripción del Sistema de Universidad Abierta (SUA).

**La locura, no es otra cosa
que la razón presentada
bajo diferente forma**

Johann Wolfgang von Goethe

CAPÍTULO 6

Breve descripción del Sistema de Universidad Abierta (SUA) .

6.1 ¿Qué es el SUA?

El Sistema de Universidad Abierta es una nueva modalidad de estudio en la cual el alumno tiene la responsabilidad de investigar e implementar todos los elementos para su aprendizaje, en esta modalidad el alumno tiene asesores, a través de los cuales se moldea la forma de adquirir el conocimiento necesario para la aprobación de las materias.

6.2 Objetivos del SUA.

Los estudios humanísticos se hacen día a día más necesarios en una sociedad orientada hacia la tecnología. La complejidad de las organizaciones empresariales, educativas y estatales requieren de humanistas que sean capaces de integrar en unidad la multiciplidad de factores. Las humanidades son un punto neurálgico de toda actividad humana.

Ante la creciente demanda de una mayor preparación en un entorno de competencia, el sistema de universidad abierta ofrece la vía para adquirir un título universitario sin tener que dejar las actividades cotidianas.

6.3 SUA en Psicología.

Al ingresar al Sistema de Universidad Abierta de la Facultad de Psicología, se está ingresando a una modalidad de enseñanza que posee sus propias y muy peculiares características en lo que se refiere a su funcionamiento.

El Sistema de Universidad Abierta de la Facultad de Psicología es una alternativa en la formación de profesionales de esta disciplina.

Desde su fundación en 1974, el SUAP se ha transformado, ha evolucionado en lo que respecta a su funcionamiento, sin embargo en lo referente a sus principios, estos *continuarán vigentes: la formación profesional de alto nivel*, dirigida principalmente a personas que por limitación de tiempo o distancia no pueden acudir a un sistema presencial o escolarizado.

Debe quedar muy claro que la Universidad Abierta no es un sistema de enseñanza por correspondencia, ni una tele-Universidad, sino un sistema que combina la utilización de guías didácticas

6.3.1 Diferencia con el sistema escolarizado.

Existe una gran diferencia entre ambos sistemas de enseñanza, la más importante es la de la presencia en los salones de clase.

Mientras en el sistema escolarizado, se establece al principio de cada ciclo escolar, un horario rígido, en donde se establecen días y horas en las cuales se presentarán los maestros para dar cátedra, en el sistema abierto, esto no existe como tal, ya que aquí no se tienen horarios ni días establecidos para adquirir los conocimientos de la materia en cuestión. Es decir aquí el alumno es el único responsable de establecer sus días y tiempos para estudiar, de igual forma el alumno es el que establece su ritmo de estudio y de presentación de exámenes. Debemos señalar que a diferencia del sistema escolarizado existen asesores, los cuales tienen la función que tiene el maestro en cuestión de dudas y orientación de estudio.

Así la diferencia de ambos sistemas radica única y exclusivamente en los alumnos, ya que el plan de estudios, los derechos, y obligaciones son las mismas.

6.3.2 Modelo de estudio en el sistema abierto.

La mayor parte de los alumnos que ingresan al sistema abierto no han tenido experiencia en esta modalidad educativa, de allí que requieran orientación y apoyo para desempeñarse no sólo sin problemas, si no de manera óptima y exitosa. Uno de los factores que pueden ser determinantes en esta situación, es el conocimiento de los elementos que constituyen el sistema.

Consideramos que es fundamental que el alumno del SUA pueda establecer una buena interacción con el sistema en su conjunto (asesores, materiales, alumnos, servicios, etc.). Se cree que al igual que cualquier estudiante,

tiene como meta, o debe tener, terminar (con éxito) el ciclo escolar que esta iniciando. Aunque lograr esta meta depende de múltiples factores, sabemos que hay una serie de elementos que pueden ser determinantes en el rendimiento escolar de un alumno, tales como la disciplina, la capacidad para organizar el tiempo, los hábitos de estudio y las estrategias de aprendizaje con las que se cuentan.

En el Sistema Abierto, los aspectos antes mencionados, tienen una enorme importancia y son fundamentales para un adecuado desempeño académico. Así como el Sistema Abierto tiene sus propias y particulares características, el estudiante SUA debe tenerlas también, debe desarrollar un perfil de alumno autodidacto, comprometido y responsable. Aunque esto no se logra en un momento.

Existen tres elementos que hacen la dinámica del SUA, los tres son interactuantes a un mismo nivel de importancia; por un lado tenemos los materiales de estudio, por otro a los alumnos y por último a los asesores. Los asesores son los profesores del SUA, es el personal académico que diseña, elabora y produce los materiales de estudio, las guías didácticas que el alumno necesita para cursar las materias, los asesores también realizan una serie de actividades académicas que redundan finalmente en el beneficio de los alumnos; asisten a cursos de perfeccionamiento, de actualización, cursan Posgrado, etc., es decir tienen actividades de formación académica, asimismo, participan en Congresos, Seminarios, Conferencias, Mesas Redondas, etc., exponiendo a través de trabajos y ponencias, sus experiencias docentes y de investigación, es decir, participan en actividades de Difusión, Extensión e Investigación, sin embargo el asesor tiene una tarea fundamental en el SUA y es atender y brindar la asesoría que se requiere.

Cuando el alumno se inscribe a una determinada materia, se esta inscribiendo también con un Asesor que imparte esa materia y será quien en lo subsecuente, brinde asesoría, que consiste en orientar e informar lo relativo a los materiales que conforman la asignatura, será quien resuelva sus dudas respecto al contenido de la misma; si los objetivos no están claros, si el criterio de evaluación no esta explícito, si la bibliografía es contradictoria, etc., y este mismo asesor es también quien evalúa los exámenes y trabajos.

Uno de los elementos básicos del SUA, lo es sin duda, la GUIA DE ESTUDIO, ya que no se tiene clases a la manera tradicional, el principal punto de apoyo, lo constituyen las guías de estudio, la guía es un instrumento esencial para poder cursar una materia.

Cada materia consta de un "paquete" de unidades que deben ser cursadas de manera secuenciada. El número de unidades de que consta cada materia varía de 6 a 9.

La guía de estudio, o guía didáctica consta de varios elementos, cada uno de los cuales posee importancia y tienen una función específica dentro de la guía, consideramos preponderante el conocimiento y manejo de cada uno de estos aspectos.

6.3.3 ¿ Qué Implica la modalidad grupal.?

La mayoría de los alumnos están acostumbrados a un tipo de relación didáctica dependiente y escasamente autónoma.

El sistema abierto supone un cambio radical, de allí que muchos estudiantes requieren de un periodo o etapa de transición entre el sistema escolarizado y el abierto.

La alternativa grupal pretende de alguna manera, ser el enlace entre dichos sistemas. Las asesorías grupales se organizan por materias y se ofrecen en horarios diversos para que el alumno pueda elegir el más conveniente. La inscripción no es obligatoria pero si la asistencia, es decir podrá inscribirse a la asesoría grupal siempre y cuando este inscrito a la materia correspondiente, pero una vez inscrito a la grupal, se adquiere la responsabilidad de asistir.

El alumno se integra a un grupo, que recibe una asesoría una vez a la semana en día y hora predefinida, siendo obligado asistir por lo menos al 80% de las asesorías.

Las asesorías se imparten con base en un programa que permite al alumno cursar las 5 asignaturas correspondientes en un semestre.

Las sesiones de asesoría tienen como propósito aclarar dudas, resolver cuestiones y en todo caso, profundizar en los temas tratados, por lo que es absolutamente necesario que el alumno haya estudiado previamente el material correspondiente.

En esta modalidad, el alumno debe presentar los exámenes respectivos, siguiendo el ritmo establecido en las propias sesiones semanales.

Debemos destacar, que el ritmo de trabajo que requiere la modalidad grupal, es mucho más fuerte que la modalidad individual, ya que en esta se programan hasta dos exámenes por semana y esto implica que el alumno debe estudiar un promedio de dos a tres unidades por semana.

6.4 Problemática general de los estudiantes del Sistema Abierto.

La comunidad de estudiantes del SUA, tiene una gran variedad de situaciones, las cuales los ponen en condiciones diferentes a las de los del sistema escolarizado.

Para efectos de ese punto, sólo nos centraremos en el aspecto de estudio¹¹, y se a tratado de englobarlos en los siguientes grupos¹²:

- **Administrativos.** Los problemas administrativos que más resiente el estudiante de SUA son los que se refieren a que su inscripción esta supeditada a los calendarios del sistema escolarizado, los planes de estudio que cursan son los mismos que en el sistema escolarizado y que no se toma en cuenta las características del SUA (con excepción del SUA de filosofía y letras) la estrategia de calificación es similar a la del sistema escolarizado, se tienen limitaciones temporales para completar una carrera y, por último, no se cuenta con los mecanismos adecuados para que el alumno mantenga una motivación alta y pueda participar en la evaluación de su aprovechamiento.
- **Informativo.** Los problemas de carácter informativo que tiene que resolver el estudiante se refieren a la carencia de información sobre el funcionamiento de este sistema (los folletos informativos que edita cada división, son demasiados genéricos y poco útiles), no se le proporciona al estudiante la información acerca del material didáctico ni de su utilización (algunas veces ni siquiera todo el material didáctico). Y, por último, muchas veces el alumno no sabe cuáles son los mecanismos de evaluación ni sus obligaciones y derechos.
- **Hábitos de estudio.** Los problemas que enfrenta el alumno con respecto a los hábitos de estudio se refieren fundamentalmente a que éste, en la mayoría de las veces, no posee las características de un autodidacta, no se le proporciona

¹¹ Para analizar más a fondo la diferencia entre ambos, consultar tesis "Un estudio comparativo sobre el aprovechamiento escolar en alumnos del SUA y el sistema escolarizado de la Facultad de Psicología", Enedina Fematt Zamarripa, 1987.

¹² "Diagnóstico del papel del alumno en el SUA UNAM (Proposición de un modelo de trabajo) María Mercedes Gamez Calderón, 1984

material ni ayuda que lo orienten para formarse en este sentido (las instancias que deben resolver este problema no tienen una coordinación central y sus esfuerzos se ven limitados por esta razón).

- **Relaciones interpersonales.** Los problemas que tiene el estudiante respecto a sus relaciones interpersonales se remiten a que, muchas veces, los asesores y los alumnos mismos, continúan con una concepción tradicional de la enseñanza y el aprendizaje, los alumnos llegan a desarrollar una dependencia académica hacia los asesores y, por último, existe poca integración de los estudiantes para desarrollar un trabajo académico conjunto.

6.5 ¿Por qué Anatomía y Fisiología del Sistema Nervioso?

Las materias del área de Psicofisiología (Bases Biológicas de la conducta, Anatomía y Fisiología del sistema nervioso, Neuroanatomía y Psicofisiología), presentan ciertas dificultades en la comprensión de procesos y memorización de los términos que implica.

Además de estos problemas, se presentan otros como el cambio del Sistema Escolarizado, al Sistema Abierto; aunado a que el material bibliográfico que se utiliza, está enfocado principalmente para estudiantes de medicina.

Las materias presentan un alto índice de reprobación; es por eso que se pretende usar a la computadora como una herramienta para el apoyo en el aprendizaje siguiendo el modelo de enseñanza programada, se podrá presentar simultáneamente un texto explicativo del contenido teórico, así como imágenes fijas y animadas que ayuden a su comprensión.

Los elementos que involucran Anatomía y Fisiología del Sistema nervioso, permiten la creación y el manejo de animaciones de los diferentes mecanismos que se llevan a cabo en el ámbito orgánico (Este manejo es mucho muy sencillo, pero muy descriptivo).

También es posible mostrar imágenes, para establecer diferencias, similitudes y problemas, los cuales en ocasiones son difíciles de imaginar o las cuales no podemos entender de una manera clara.

Es por eso que el contenido de esta materia, proporciona elementos para generar un programa de aprendizaje auxiliado por computadora, el cual genere grandes beneficios a los alumnos en general, y en este caso a los pertenecientes al Sistema Abierto

6.6 Proyecto actual.

Actualmente no se tiene contemplado ningún tipo de proyecto en torno a estos temas, debido a que en los departamentos dentro de la facultad de Psicología no existe alguno que se dedique específicamente a desarrollar material nuevo que apoye a los estudiantes.

El proyecto que se generó como tesis tiene la finalidad de establecer un parámetro para mostrar que es viable generar este tipo de tecnologías de la educación como un recurso más, no sólo para el sistema abierto, sino una herramienta más para toda la comunidad de la facultad de Psicología, que en su momento necesite tener una consulta más rápida y específica.

Las bases que se tomaron fueron algunos programas que se han generado en la UCII (Unidad de Cómputo Informática e instrumentación) de la Facultad de Psicología. Estos programas fueron diseñados por el Lic. Manuel González Oscoy, quien es profesor de la asignatura de Psicofisiología.

Los programas que se generaron fueron diseñados y creados en autorware, estos programas sólo abarcaban algunas unidades de la primera materia del bloque de Psicofisiología (Bases biológicas de la conducta). Los programas consistían en mostrar algunos temas del temario con sus gráficas (dibujos y esquemas), en cada uno de ellos se mostraban los aspectos más importantes.

Cabe señalar que los programas eran demasiado grandes en cuanto a su tamaño y si se quería realizar alguna corrección en cuanto a su contenido, ya sea gráfico o definición, se tenía que reabrir el código fuente y volver a generar el archivo ejecutable.

CAPÍTULO 7

Un buen software.

**La razón es la inteligencia en ejercicio;
La imaginación es la inteligencia en erección.**

Víctor Hugo.

CAPÍTULO 7 Un buen software.

7.1 Sistemas basados en computadora.

Comenzaremos analizando que significa, y que uso se le a dado a la palabra sistema. Siempre se habla de diferentes tipos de sistemas, sistemas políticos, sistemas educacionales, sistemas bancarios, etc. Pero que nos dice realmente la palabra sistema, ya que la utilizamos para entender que se trata de algo complejo.

Tomemos la definición formal de sistema para comprender mejor su significado.

Sistema: Conjunto de reglas o principios sobre una materia enlazados entre sí. Conjunto ordenado de cosas que contribuyen a un fin.

De igual manera definiremos a continuación el significado de un sistema basado en computadora.

Sistema basado en computadora: Un conjunto de elementos que están organizados de acuerdo a algún método o proceso de control con el fin de procesar información.

Es así como a continuación describiremos los elementos que constituyen un sistema basado en computadora.

•**Software:** Programas de computadora, estructuras de datos, y la documentación relativa que servirá de efecto al método lógico y por último los procedimientos de control que sean necesarios.

•**Hardware:** Elementos electrónicos (CPU, memoria) que provee la capacidad de procesamiento a la computadora, y los elementos electromecánicos que le permiten realizar una interacción con las funciones exteriores.

•**Gente:** Usuarios y operadores de hardware y software.

•**Base de datos:** Una larga, y organizada colección de información que es accesada a través del software y es una parte integral del funcionamiento del sistema.

•**Documentación:** Manuales, formas y cualquier otro tipo de información descriptiva que proporcione la forma de uso y operación del sistema.

•**Procesos:** Los pasos que definen el uso específico de cada elemento del sistema, así como también el contexto procedimental en el cual el sistema recibe.

Los elementos antes descritos a través de una gran variedad de combinaciones permiten obtener diferentes formas para la transformación de la información. Como lo podemos observar en la siguiente figura, en donde se muestra como se dan las diferentes interacciones.

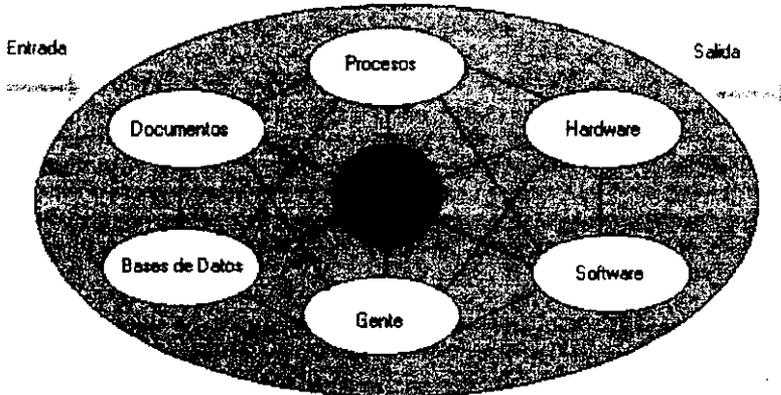


Figura 7.1 Elementos de un sistema

7.2 Ingeniería de Sistemas de Computadora.

La actividad principal de la Ingeniería de Sistemas de Computadora es la solución de problemas a través del diseño de sistemas funcionales compuestos por lo elementos más idóneos. Esta actividad comienza cuando el cliente define las metas, y restricciones, a continuación se procede a con la representación de las funciones, la forma en que se ejecutarán, el diseño de las interfazs y la estructuración de la información que será asignada de los elementos genéricos del sistema descritos en la sección de procedimientos.

El principal problema que se enfrenta al crear un nuevo sistema, es el concepto confuso del diseño de las funciones. Sin embargo, la Ingeniería de Sistemas debe poner los límites al sistema que se está generando, a través de la

identificación real del alcance de las funciones, así como la forma en que queremos que éstas se ejecuten.

Es importante hacer notar que la Ingeniería de Sistemas debe también considerar todos los posibles problemas que puedan presentársele al cliente, y generar así mismo las soluciones. Es por eso que podemos dividir la Ingeniería de Sistemas en las siguientes partes.

Hardware e Ingeniería del Hardware. La Ingeniería de Sistemas de Computadora selecciona algunas combinaciones de componentes de hardware que comprende uno de los elementos de los sistemas basados en computadora. La selección no es tan simple como pareciera, ya que debe comprender las siguientes características.

1. Los componentes son empaquetados como bloques individuales.
2. Las interfazs entre los componentes deben estar estandarizados.
3. Deben proporcionarnos diferentes alternativas.
4. La ejecución, costo y disponibilidad deben ser relativamente fáciles de determinar.

Una configuración de hardware envuelve una jerarquía de componentes discretos (circuitos integrados y componentes electrónicos tales como: resistencias, capacitores, etc.) los cuales son ensamblados en una tarjeta impresa de circuitos, la cual ejecuta un conjunto de operaciones específicas.

La Ingeniería de hardware para computadoras digitales ha crecido y crecerá a través de las décadas. Los procesos de la Ingeniería del hardware pueden ser resumidos en tres fases: planificación y especificación; diseño e implementación del prototipo; y manufacturación, distribución, y campo de servicio.

Software e Ingeniería del Software. Las funciones y formas de ejecución son asignadas al software durante la Ingeniería de Sistemas. En algunos casos, la función es estructurada a través de la implementación de una secuencia de procesos para la manipulación de los datos. En otros casos la función es la coordinación interna y el control concurrente de otros programas, en el primer caso la ejecución no está explícitamente definida, mientras que en el segundo es definida explícitamente en términos de respuesta y tiempos de espera.

El elemento de software de un sistema basado en computadora está comprendido por programas, datos, y documentación que es categorizada como software de aplicación y software de sistema; el primero implementa los procesos que son requeridos para acomodar las funciones del procesamiento de la información, el segundo implementa las funciones de control con las cuales el software será capaz de interactuar con otros elementos del sistema.

Factores Humanos e Ingeniería Humana. Casi todos los sistemas basados en computadora tienen un elemento humano. Una persona que pueda directamente interactuar con el hardware y el software, a través de un diálogo que permita manejar las funciones del sistema; en todos los casos la gente es responsable del desarrollo y mantenimiento del sistema.

En la Ingeniería Humana además de tomar en cuenta la interacción hombre máquina, debemos introducir la palabra "uso amistoso", la cual es de vital importancia en la actualidad en el desarrollo sistemas, ya que el uso amistoso permite un mejor aprovechamiento del sistema que se utiliza.

La Ingeniería Humana es considerada como una actividad multidisciplinaria que aplica el conocimiento derivado de la psicología y tecnología para especificar y diseñar alta calidad en la interacción hombre máquina. El proceso de Ingeniería Humana debe contemplar los siguiente pasos:

1. **Análisis de actividad.** Cada actividad que ha sido asignada a elementos humanos es evaluada en el contexto de requerimientos de interacción con otros elementos.
2. **Análisis y diseño semántico.** El preciso significado de cada acción requerida por el usuario y producida por la máquina debe ser bien definida.
3. **Diseño del ambiente del usuario.** Hardware y software y otros elementos del sistema deben ser combinados para formar el ambiente de trabajo del usuario.
4. **Prototipos.** Este es muy difícil pero no imposible para formalmente especificar una interacción hombre máquina sin el uso de un prototipo.

7.3 Análisis del Sistema.

Esta actividad encierra muchas de las tareas que hemos denominado Ingeniería de sistemas de computadoras. El análisis está orientado teniendo en mente los siguientes objetivos:

1. Identificar las necesidades del cliente.
2. Evaluar el concepto de sistema para la factibilidad.
3. Ejecución económica y técnicas de análisis.
4. Asignación de funciones para el hardware, software, personas, bases de datos, y otros elementos del sistema.
5. Establecer costos y esquemas de necesidades.
6. Crear una definición de sistema que forme las bases para todo el trabajo de ingeniería subsecuente.

A continuación daremos una pequeña descripción de las áreas que debe de abarcar un buen análisis.

7.3.1 Identificación de necesidades.

Este representa el primer paso en el proceso de análisis, aquí se identifican cuáles serán las necesidades a cubrir del futuro software. Este representa el punto de partida para la evolución de cualquier sistema basado en computadoras, aquí es válido realizar todo tipo de preguntas acerca del software que se quiere crear. Algunas de las preguntas que se pueden hacer son las siguientes: ¿Qué información se producirá?, ¿Qué información necesitará?, ¿Qué funciones y acciones son requeridas?, ¿Existe la tecnología necesaria?, etc.

Toda la información que es generada durante la identificación de necesidades sirve para crear el documento de concepto del sistema, éste permitirá establecer los objetivos a los cuales se desea llegar.

7.3.2 Estudio de Factibilidad

En ocasiones no todos los proyectos son factibles para su realización ya que desgraciadamente no se cuenta con los recursos necesarios, ni con el tiempo indispensable. Esto da como resultado que se tenga que realizar una evaluación prudente de la factibilidad del proyecto en el menor tiempo posible.

Durante la Ingeniería de Sistemas, sin embargo, debemos concentrarnos en cuatro aspectos esenciales que son:

- **Factibilidad económica.** Es una evaluación del costo estimado de desarrollo contra los ingresos fundamentales o los beneficios derivados del desarrollo del sistema.
- **Factibilidad técnica.** Es el estudio de las funciones, ejecuciones y restricciones que pueden afectar el funcionamiento adecuado del sistema.
- **Factibilidad legal.** Es la determinación de cualquier tipo de infracción, violación o linealidad que pueda resultar del desarrollo del sistema.
- **Alternativas.** Es una evaluación de todas alternativas posibles que se puedan tener en el desarrollo del sistema.

Sin embargo, un estudio de factibilidad no garantiza una justificación económica, un bajo riesgo técnico, pocos problemas legales y la falta de existencia de alternativas razonables.

7.3.4 Análisis económico

Este es sin duda un elemento esencial en la factibilidad de estudio, nos referimos al análisis de costo beneficio, es decir una valoración de la justificación económica para un proyecto de sistema basado en computadora. El estudio del costo beneficio delinea los costos para el desarrollo del producto y su estimación, además de establecer los beneficios intangibles de un sistema.

La complejidad del análisis dependerá de cuan variadas sean las características del sistema a desarrollar y el relativo tamaño del proyecto.

7.3.5 Análisis técnico

Durante el análisis técnico, el analista evalúa los métodos técnicos del concepto de sistema mientras que al mismo tiempo recolecta información adicional acerca de la forma de ejecución, confiabilidad, mantenimiento y productividad. En algunas ocasiones el proceso de análisis de sistema, también incluye una cantidad limitada de investigación y diseño.

El análisis técnico comienza con una valoración de la viabilidad técnica del sistema propósito, esto se puede hacer a través de las siguientes preguntas: ¿Qué tecnologías son requeridas para producir el funcionamiento del sistema y su ejecución?, ¿Qué nuevo material, métodos, algoritmos, o procesos son requeridos y cuál es su riesgo de desarrollo?, etc.

7.4 Análisis del Sistema Propuesto

- Como se menciona en el capítulo 7, el bloque de materias que conforman el área de Psicofisiología, tienen un alto grado de reprobación como consecuencia del poco manejo de la información entorno a los temas fisiológicos. El sistema que se debe de implementar, debe contemplar las siguientes características:
 - Manejo de toda la información en torno al tema de Anatomía y Fisiología del sistema Nervioso, así como también la posibilidad de modificar y agregar información, o en su caso la eliminación de ésta.
 - Relacionar en lo más posible la información anterior con imágenes gráficas (Fotografías, dibujos, animaciones y videos), y de igual manera, tener la opción de agregar, modificar o eliminar cualquier elemento gráfico. Los gráficos no necesariamente tienen que estar relacionados con la información que se este manejando, ya que en ocasiones se estará tratando algún tipo de información que será más explícita con más de una imagen gráfica.
 - Tener la posibilidad de generar presentaciones relativas al tema o de algún otro tema que pueda ayudar a comprender el tema principal.

Como podemos ver necesitamos un sistema flexible que nos permita hacer un manejo transparente de dos elementos primordiales de información en general(Definiciones, información en general, términos específicos del área medica, etc.) e imágenes gráficas.

7.5 Definición de Requerimientos.

La definición de los requerimientos es un estudio de las necesidades que genera el problema o sistema que se esta tratando de implementar y/o la forma de mejorar los aspectos donde se este fallando. Un requerimiento es una característica que debe ser incluida en el sistema y puede ser una forma de captar o procesar datos, producir información, controlar alguna actividad o dar apoyo para la toma de decisiones.

El primer paso para la identificación de requerimientos es realizar una investigación, y para este paso existe una gran variedad de métodos los cuales

nos permiten recopilar datos sobre la situación existente: observación, cuestionarios, entrevistas, etc. Para nuestro sistema realizamos una descripción general del funcionamiento del SUAP, así como de la interacción de los alumnos que se encuentran estudiando dentro de este entorno.

7.6 Descripción general del sistema propuesto.

7.6.1.- Configuración del sistema

La configuración del sistema abarcará todo el mantenimiento a las bases de datos en donde se almacenarán, este abarca la modificación, eliminación y adición de cualquiera de los elementos multimedia con los cuales se estará trabajando.

Las bases de datos son el elemento medular del sistema, ya que a través de éstas, se parte para los módulos anteriores, siendo las bases de datos los *elementos alimentadores para las presentaciones y los catálogos generales.*

7.6.2 - Presentación libre

La información se recopila de acuerdo a la guía de estudio de la materia de Anatomía y Fisiología del Sistema Nervioso, debe ser presentada al alumno de acuerdo a la relación que existe entre las unidades que conforman la materia.

Esto quiere decir que el alumno deberá tener la posibilidad de escoger directamente cuál es la información que le interesa conocer.

En este módulo se debe tener la conexión de información e imagen, ya que esta combinación dará un mayor soporte al aprendizaje o comprensión de la información que se está consultando.

7.6.3 - Presentación por módulos

La información utilizada en el módulo anterior, puede ser generada en forma de libro electrónico, es decir conforme a una secuencia que puede ser manipulada libremente en cuanto a la secuencia se refiere, se tendrá la opción de tener presentaciones de diferentes temas, los cuales pueden o no estar relacionados con el tema de Anatomía y fisiología del Sistema Nervioso.

7.6.4- Consulta de términos específicos de medicina o anatomía.

Se debe de tener una fuente de términos(Glosario), la cual nos permita entender algún elemento relacionado con el tema en cuestión, o en su caso, nos permita identificar los elementos que son nuevos para nosotros o que jamás los habíamos visto. Este glosario tendrá la facilidad de ser consultado en cualquier parte del programa.

7.6.5 – Consulta de imágenes.

Este módulo permitirá consultar todas las imágenes disponibles, éstas no necesariamente deben ser sólo de la materia, ya que se podrá incorporar imágenes relativas al tema principal, o imágenes que permitan ejemplificar alguna definición u elemento de información.

7.6.6- Consulta de Videos.

De la misma forma que las imágenes, los videos proporcionan un gran apoyo, ya que en ocasiones existen explicaciones de ciertos procesos biológicos que presentan dificultad en cuanto a su comprensión. Es por eso que la incorporación de videos puede ser un punto vital del sistema propuesto.

7.6.7- Glosario.

El glosario permitirá incorporar palabras específicas que se utilizan en medicina y fisiología, estas palabras permitirán dar una explicación más detallada de un tema, o despejar una duda en cuanto a la terminología utilizada.

7.6.8 Requerimientos Generales

Ya establecidos los elementos que deben conformar el sistema propuesto y analizando lo expuesto en el capítulo 7, podemos establecer los siguientes requerimientos:

1.- Hardware.

El hardware necesario dadas las características del programa, será una Computadora personal capaz de soportar multimedia (la cual es especificada en el capítulo 4)

2.- Bases de datos.

Las bases de datos necesarias para poder operar de una forma eficiente y ordenada toda la información planteada en el análisis del sistema.

3.- Documentos.

La fuente de información es básica, ya que es a partir de ésta que se generarán las todas las consultas establecidas en el análisis de la información. Las fuentes de información, serán variadas, ya que los temas sobre fisiología y psicología, abarcan desde libros, revistas, artículos especializados, etc.

Pero aun así se debe considerar que la información sustancial tendrá que girar alrededor de las guías de estudio de la materia de Anatomía y Fisiología del sistema nervioso, y por ende de la bibliografía que se recomiende. No quitando la posibilidad de consultar otro tipo de información.

En la gráfica de la siguiente hoja se muestra la descripción del proceso que sigue la obtención de la información, la cual alimentará las bases de datos del sistema, para su uso y manejo.

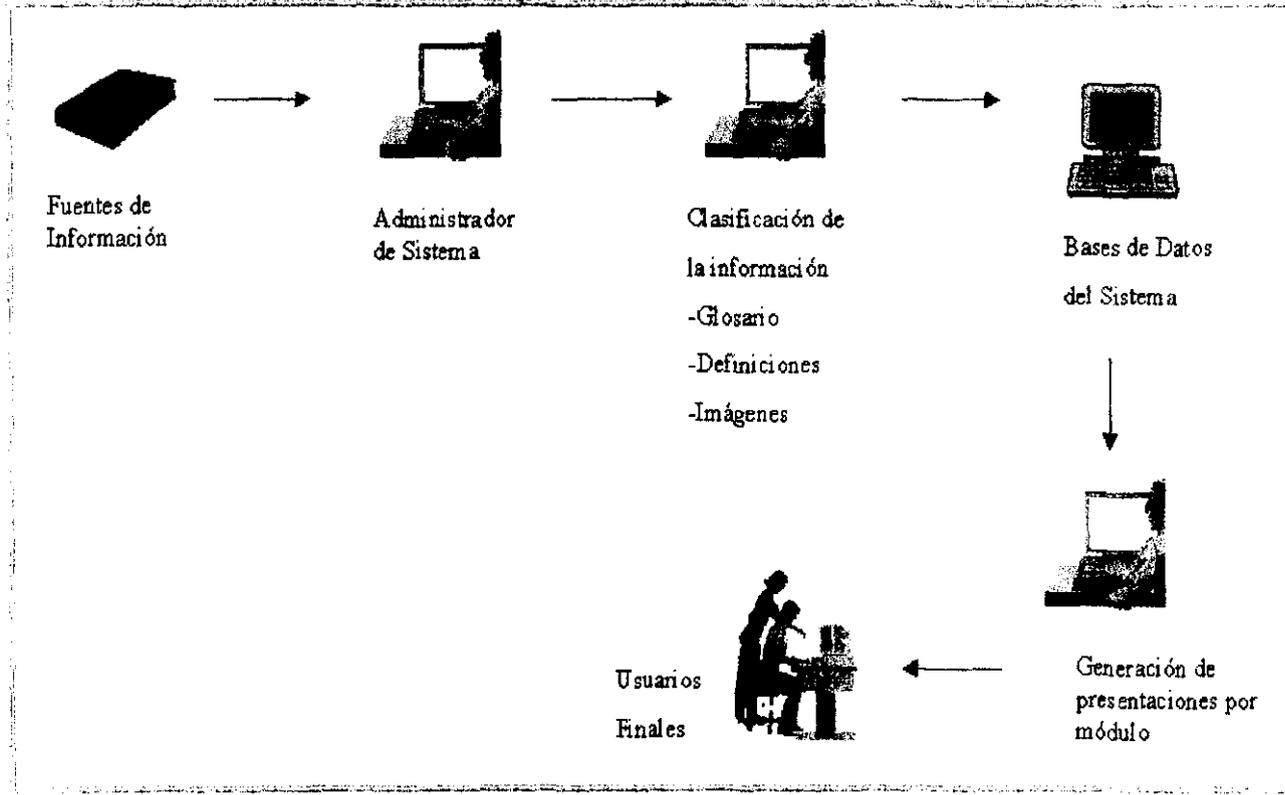


Figura 8.1 Secuencia que sigue la obtención de la información para su uso y aplicación.

7.6.9 Objetivos principales de los elementos del sistema propuesto.

1.- Configuración del sistema.

- Registro y control del catálogo de información.
- Registro y control del catálogo multimedia.
- Registro y control del catálogo de aplicaciones.
- Creación de las presentaciones por módulo.

2.- Presentaciones libres.

- Presentación de la información para su elección.
- Presentación de la imagen correspondiente a la información.

3.- Presentaciones por módulo.

- Presentación de la información en forma secuenciada.
- Presentación de la imagen correspondiente.
- Control de la elección del módulo a utilizar (Aplicación)

4.- Catálogo de imágenes.

- Controlar la selección de la imagen a visualizar.

5.- Catálogo de Videos.

- Control de la selección del vídeo a observar.

6.- Glosario.

- Mostrar los diversos términos que se tienen con referencia al tema principal.
- Controlar el manejo de los términos.

7.7 Análisis estructurado

El análisis estructurado como todos los requerimientos del software dentro de los métodos de análisis es un modelo de construcción. Usa una notación que es única en el método de análisis estructural, se crean modelos que representan la información (datos y control), el flujo de ésta, y su contenido. A través de estas estructuras, subdividimos al sistema en torno a su funcionamiento y a la forma en que éste debe comportarse.

7.7.1 Diagramas de flujo de datos (DFD)

Este es un elemento que forma parte del análisis estructural, ya que en éste se muestran las transformaciones a las cuales serán sometidos los datos utilizados en el sistema. Es una técnica gráfica que muestra el flujo y transformación que es aplicada desde la entrada hasta la salida a la información.

Puede ser usado para representar un sistema o software en cualquier nivel de abstracción. De hecho, los DFD pueden ser particionados en niveles que representen el flujo de incremento de información y los detalles funcionales.

Debemos aclarar que un DFD es sólo una herramienta más de modelado y que únicamente proporciona un punto de vista del sistema, pero con un enfoque orientado a las funciones.

Esta herramienta nos proporciona las siguientes ventajas:

- No se necesita dar una explicación, ya que el diagrama por sí sólo es entendible, debido a la facilidad de su notación y sencillez.
- El tamaño del diagrama no abarca más de una página, esto nos permite revisarlo sin ningún problema, y modelar el sistema sin ningún tipo de complejidad.

El diagrama nos proporciona una gran facilidad de representación para cualquier tipo de sistema a cualquier nivel de abstracción, ya que los DFD pueden ser creados a un gran nivel de detalle funcional.

La notación que se usa en los DFD, no presenta ningún tipo de dificultad, ya que sólo se utilizan los siguientes elementos:

- **Elipse:** Indica una entidad externa, es decir un elemento del sistema, u otro que produce información que alimentara el sistema, u otro sistema que genera información a transformar.
- **Bloque:** Representa un proceso de transformación que se aplica a los datos y a los cambios de alguna forma.
- **Flechas:** Muestran el flujo de datos, desde su origen y hasta su destino.
- **Rectángulo parcial:** Representa un almacenamiento de información, la cual es utilizada por el software.

Debemos aclarar que el DFD no nos indica ningún tipo de procesamiento, ni tampoco alguna secuencia en especial, ya que la representación de los procesos se explica generalmente en la fase de diseño.

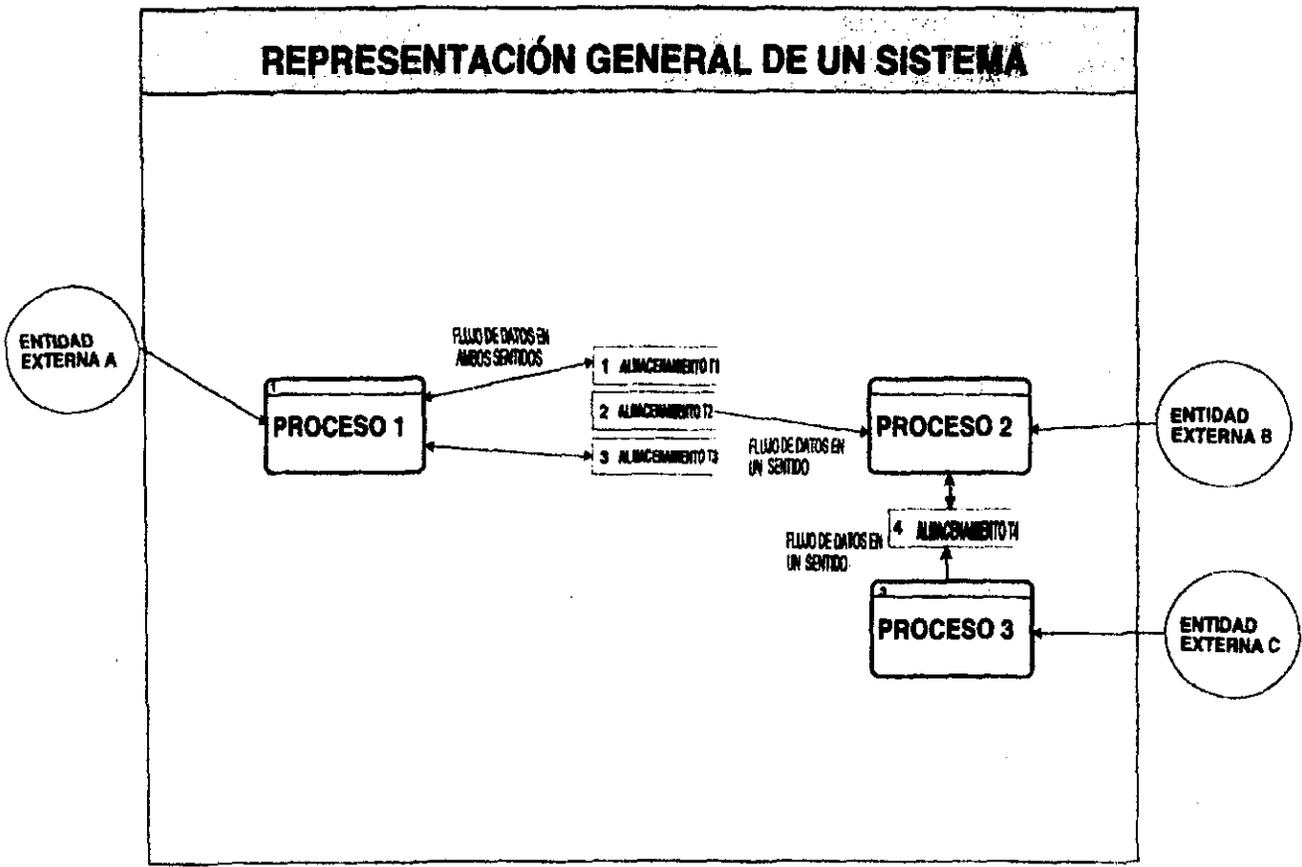
7.7.2 Diccionario de Datos

El análisis del sistema se completa definiendo cada uno de los elementos que conforman el diagrama de flujo de datos, así como también las relaciones que existen entre los elementos definidos.

Dicha definición es el diccionario de requisitos, el cual es utilizado como una gramática casi formal, para describir el contenido de los objetos que se han definido durante el análisis estructurado.

Este diccionario de requisitos tiene como principal objetivo definir en una forma detallada, cuáles serán las relaciones entre todos los elementos que forman el diagrama de flujo de datos. Los elementos que conforman el diccionario de requisitos son: la definición precisa de los almacenamientos de información o estructuras de datos, la descripción detallada de los procesos y la explicación de los elementos externos.

REPRESENTACIÓN GENERAL DE UN SISTEMA



7.7.3 Diagramas de flujo de datos para el sistema propuesto y su diccionario de datos.

Realizando un análisis de los datos de entrada y de salida del sistema, tenemos como resultado dos entidades externas, que se relacionan con él.

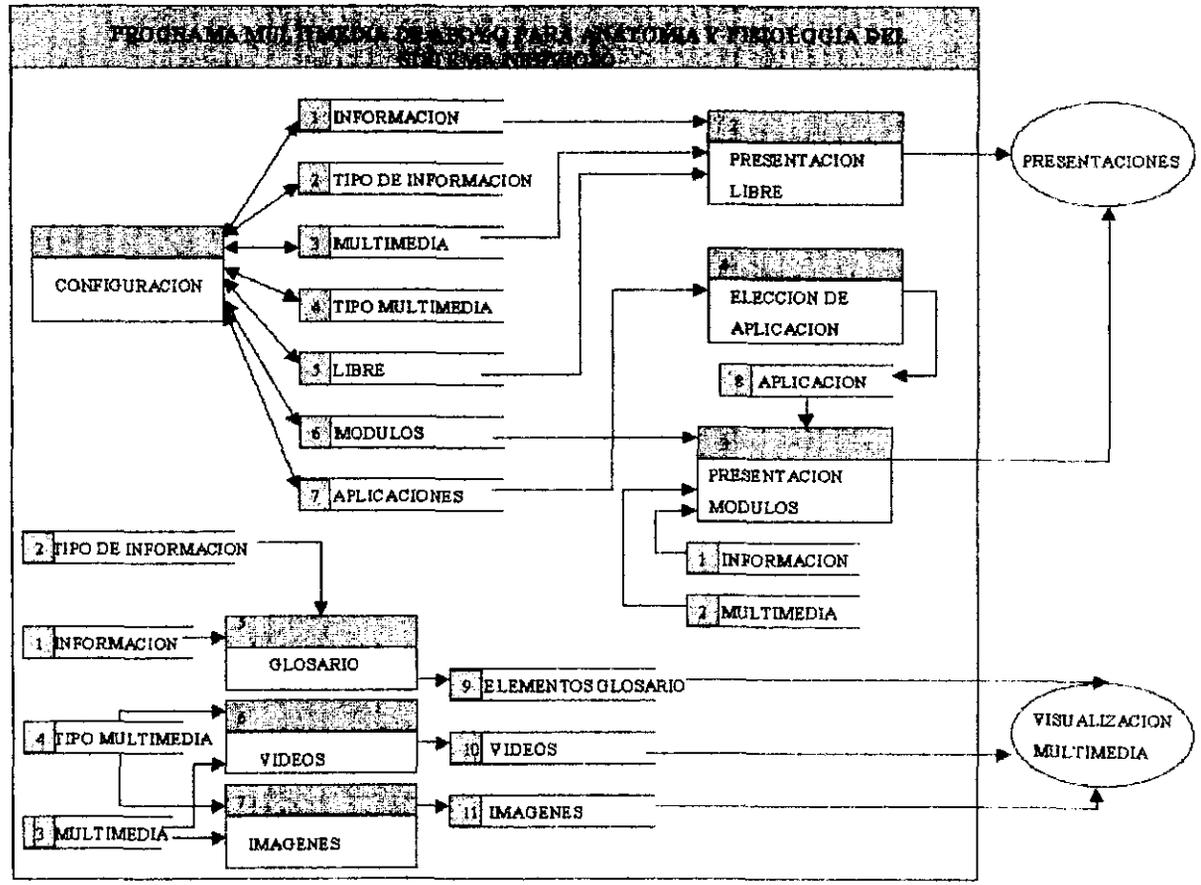
La figura de la siguiente hoja, muestra el DFD el cual se inicia indicando un solo proceso, y a continuación se describen las dos entidades externas que se relacionan con el sistema.

- **Presentaciones:** Esta entidad muestra el resultado de las presentaciones seleccionadas, ya sea la libre o por módulos, esta entidad muestra definiciones e imágenes.
- **Visualización multimedia:** Representa el resultado final de la selección de algún elemento multimedia que se tenga almacenado en los catálogos multimedia, así como de la selección del tipo de multimedia, ya sea videos o imágenes.

Un análisis más detallado de la operación del sistema propuesto, arroja 7 procesos principales: (1) Configuración, (2) Presentación libre, (3) Presentación módulos, (4) Elección de aplicación, (5) Glosario, (6) Videos, (7) Imágenes.

Una característica de los DFDs, es que nos permite realizar un análisis por niveles, los cuales nos proporcionan detalles más específicos, y nos permite definir los alcances y limitaciones de nuestro sistema. Esto nos da la pauta para crear una estructura funcional con un alto grado de detallamiento entorno al sistema. Por último al ir modelando el sistema, la complejidad de los flujos de datos, procesos y almacenamiento se incrementan, es por eso que se deben crear DFDs de primer nivel, segundo nivel, etc., de acuerdo a las necesidades del modelamiento del sistema.

PROGRAMA MULTIMEDIA DE DESARROLLO PARA ANATOMIA Y FISIOLOGIA DEL



Proceso principal: 1.- Configuración

Descripción: Proceso principal de registro y control de catálogos que utiliza el sistema propuesto.

Subprocesos que integran el proceso principal 1:

Proceso 1.1.- Información: Proceso de mantenimiento al catálogo de información en general que manejará el sistema propuesto, se encargará de altas, bajas y cambios.

Proceso 1.2.- Tipo de información: Proceso de mantenimiento al catálogo del tipo de información (esta relacionada con la información del proceso anterior) que se estarán manejando, altas, bajas, y cambios.

Proceso 1.3.- Multimedia: Proceso de mantenimiento al catálogo de los elementos multimedia que se manejarán en el sistema propuesto, altas, bajas, y cambios.

Proceso 1.4.- Tipo multimedia: Proceso de mantenimiento al catálogo de tipos de elementos multimedia (esta relacionado con la información del proceso anterior) que se estarán manejando, altas, bajas y cambios.

Proceso 1.5.- Libre: Proceso de mantenimiento a la relación que existe entre los elementos de información y los elementos multimedia (imágenes) utilizados en la presentación libre, el único proceso que se define es cambio de elementos relacionados.

Proceso 1.6.- Módulos: Proceso de mantenimiento al contenido de los archivos donde se almacena la secuencia del presentación de los elementos de información para un módulo en especial, alta, baja, y cambios.

Proceso 1.7.- Aplicaciones: Proceso de mantenimiento al catálogo de aplicaciones disponibles para presentaciones por módulo, esta relacionado con el proceso anterior, altas, bajas y cambios.

Estructuras de datos asociados al proceso principal 1

T1.- Información: Catálogo de la información que será utilizada por el sistema propuesto.

T2.- Tipo de información: Catálogo del tipo de información que se está utilizando en el catálogo de información.

T3.- Multimedia: Catálogo de los elementos multimedia que serán utilizados por el sistema propuesto.

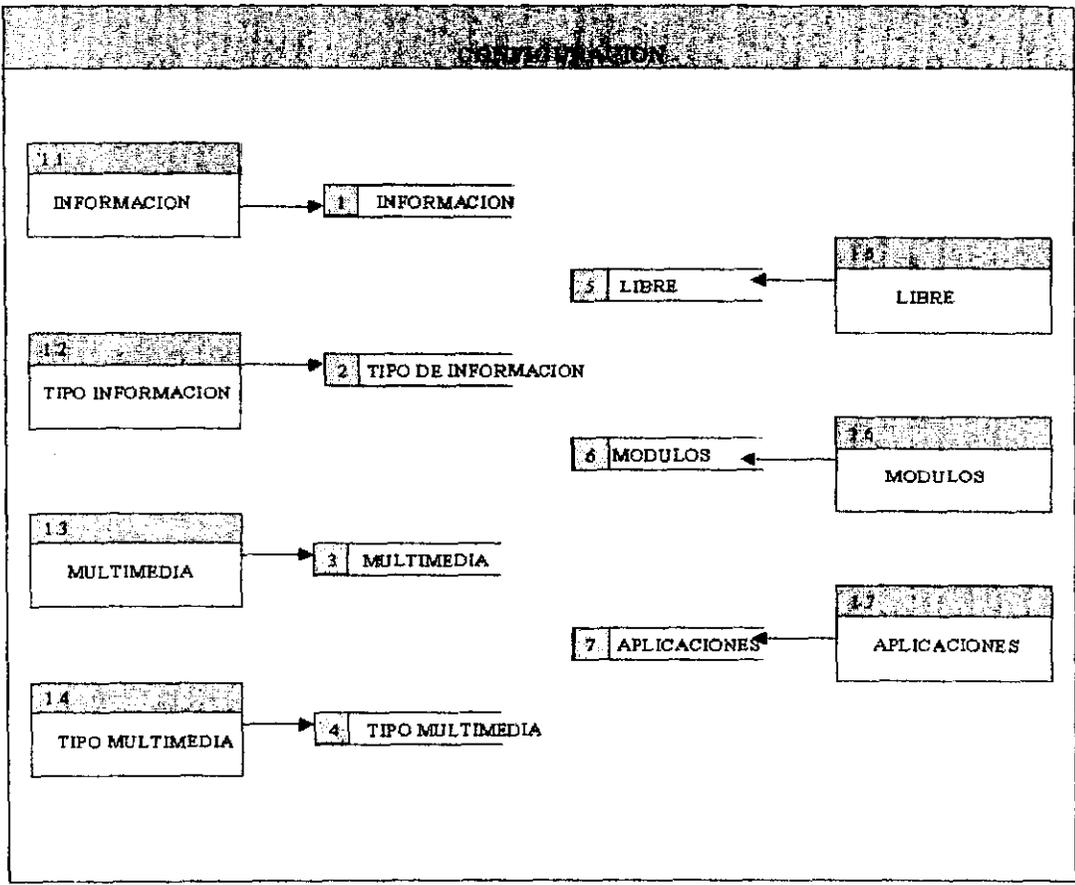
T4.- Tipo Multimedia: Catálogo del tipo de elemento multimedia que se está utilizando en el catálogo multimedia.

T5.- Libre: Catálogo de asociación entre la información y multimedia.

T6.- Módulos: Almacena la secuencia de presentación del módulo.

T7.- Aplicaciones: Catálogo de las aplicaciones para la presentación por módulos.

El Diagrama de Flujo de Datos de configuración se muestra en la siguiente hoja.



Proceso principal 2.- Presentación libre

Descripción. Proceso principal de presentación libre del sistema propuesto.

Subprocesos que integran el proceso principal 2:

Proceso 2.1 Conexión de elementos: Proceso que se encarga de analizar, cuál es la elección del usuario, y mostrar la información y la imagen correspondiente.

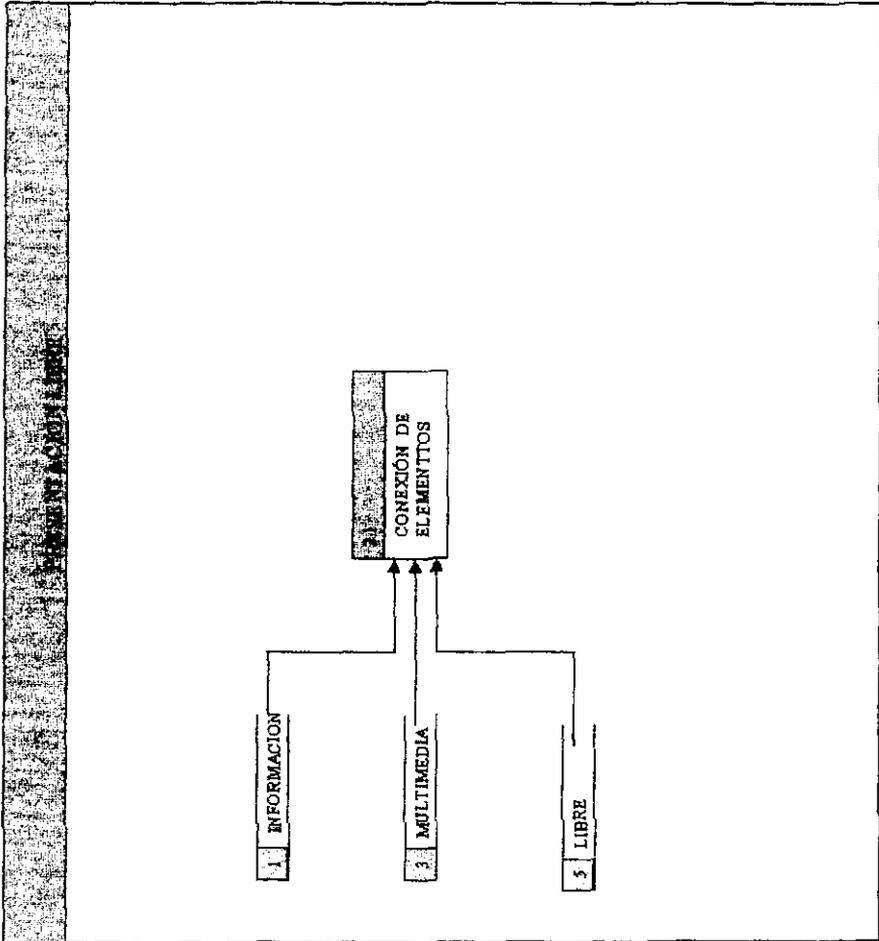
Estructuras de datos asociados al proceso principal 2

T1.- Información: Catálogo de la información que será utilizada por el sistema propuesto.

T3.- Multimedia: Catálogo de los elementos multimedia que serán utilizados por el sistema propuesto.

T5- Libre: Almacena la relación entre la información y la imagen correspondiente.

El Diagrama de Flujo de Datos de la presentación por módulos se muestra en la siguiente hoja.



Proceso principal 3.- Presentación por módulos

Descripción. Proceso principal de presentaciones por módulos del sistema propuesto.

Subprocesos que integran el proceso principal 3:

Proceso 3.1 Lectura de la aplicación: Proceso que se encarga de leer la aplicación correspondiente, para leer los módulos correspondientes donde se encuentran las secuencias de presentación.

Proceso 3.2 Conexión de elementos: Proceso que se encarga de leer la secuencia de los módulos, y leer la información e imagen correspondientes.

Estructuras de datos asociados al proceso principal 3

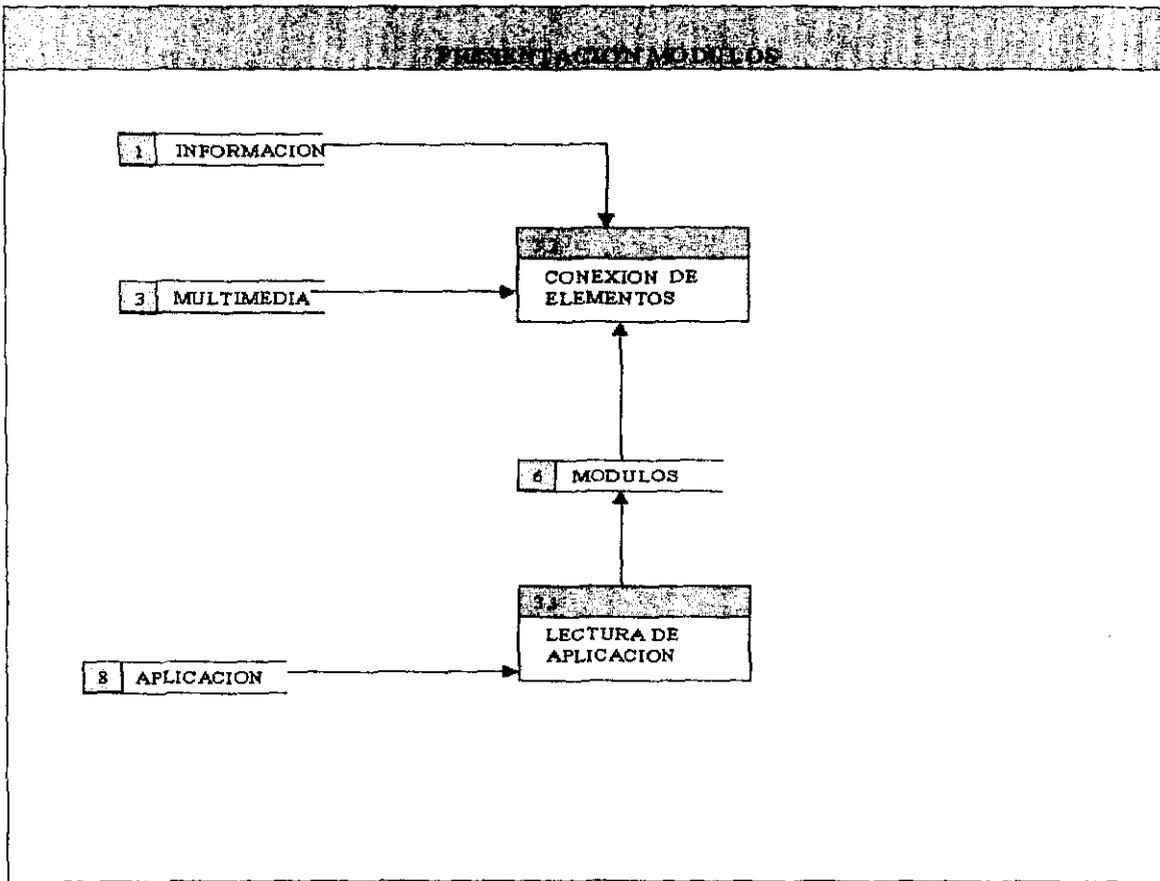
T1.- Información: Catálogo de la información que será utilizada por el sistema propuesto.

T3.- Multimedia: Catálogo de los elementos multimedia que serán utilizados por el sistema propuesto.

T6.- Módulos: Almacena la secuencia de presentación del módulo.

T8.- Aplicación: Almacena las diferentes aplicaciones que se tienen en las presentaciones por módulos.

El Diagrama de Flujo de Datos de la presentación por módulos se muestra en la siguiente hoja.



Proceso principal 4.- Elección de Aplicación

Descripción. Proceso principal de la elección de la aplicación del sistema propuesto.

Subprocesos que integran el proceso principal 4:

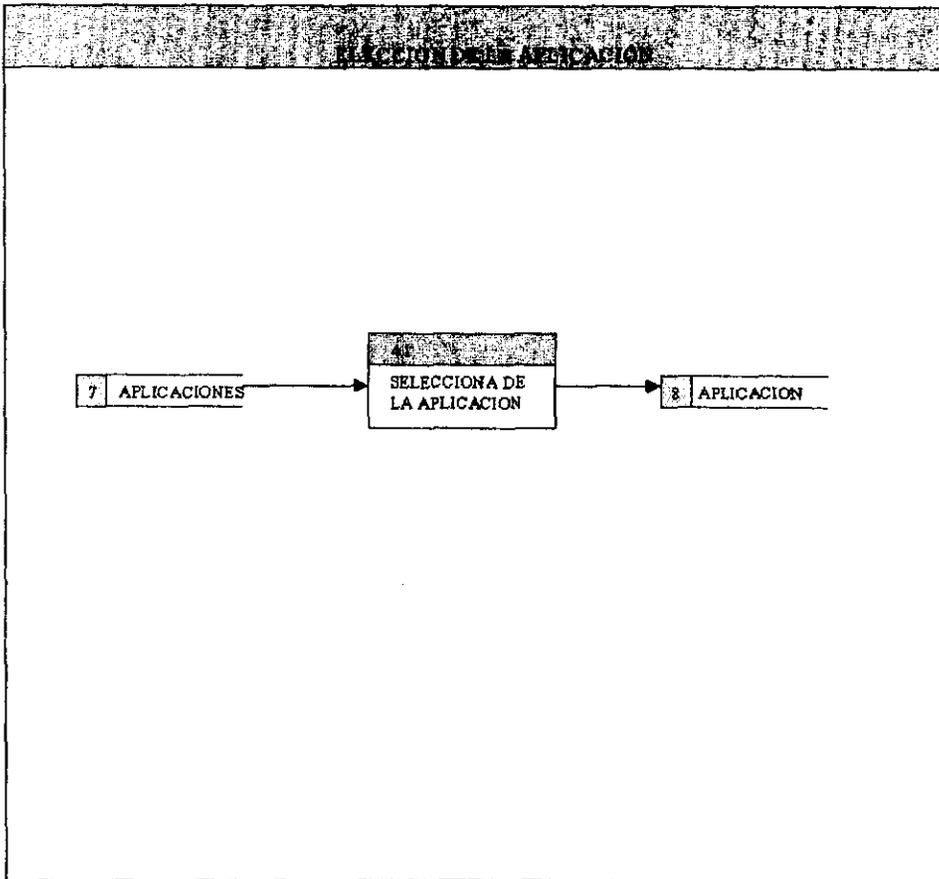
Proceso 4.1 Selección de la aplicación: Proceso que se encarga de escoger la aplicación para leer los módulos correspondientes donde se encuentran las secuencias de presentación.

Estructuras de datos asociados al proceso principal 4

T7.- Aplicaciones: Catálogo de las aplicaciones disponibles para la presentación por módulos.

T8.- Aplicación: Catálogo de los módulos que conforman, cada aplicación.

El Diagrama de Flujo de Datos de la presentación por módulos se muestra en la siguiente hoja.



**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

Proceso principal 5.- Glosario

Descripción. Proceso principal Glosario del sistema propuesto.

Subprocesos que integran el proceso principal 5:

Proceso 5.1 Selección de la información: Proceso que se encarga de leer los elementos de la información, que pertenecen al glosario (este proceso es prácticamente una consulta.)

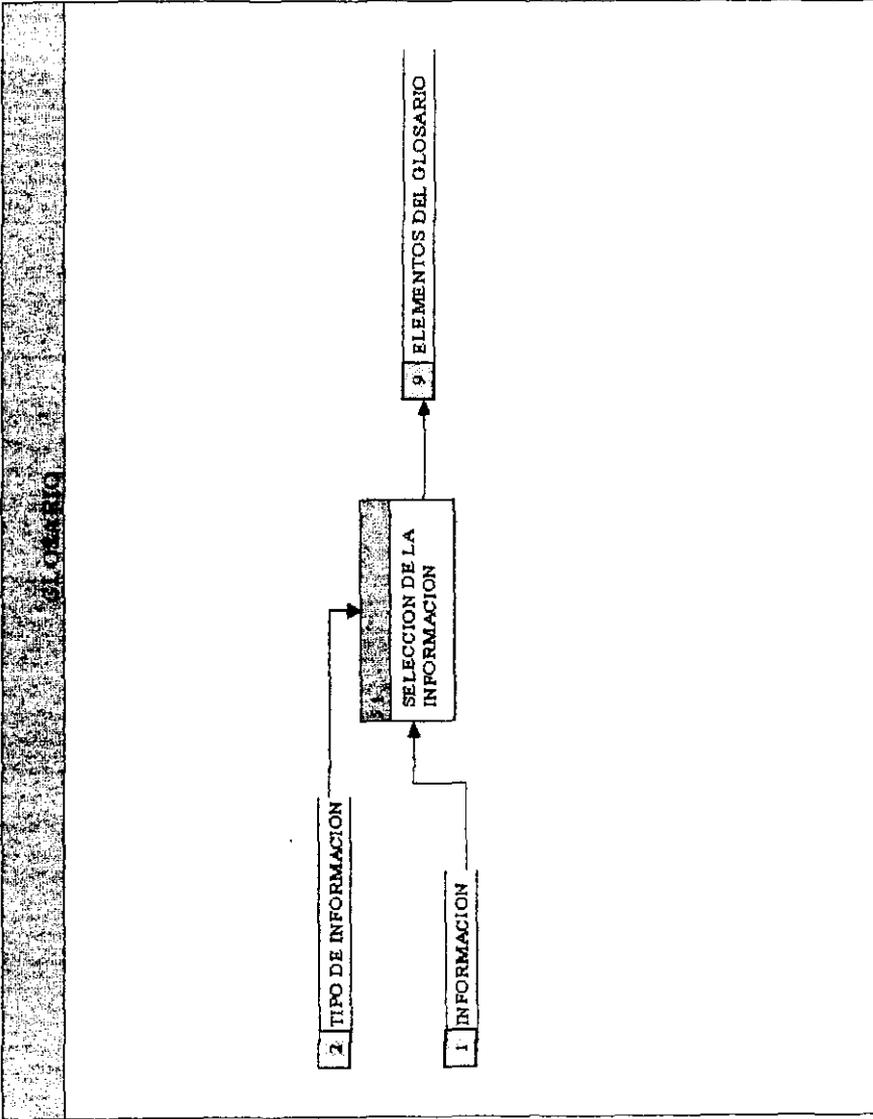
Estructuras de datos asociados al proceso principal 5

T1.- Información: Catálogo de la información que será utilizada por el sistema propuesto.

T2.- Tipo de información: Catálogo del tipo de información que se está utilizando en el catálogo de información.

T9.- Catálogo de los elementos que pertenecen al glosario (es el resultado de la consulta al catálogo de información)

El Diagrama de Flujo de Datos de la presentación por módulos se muestra en la siguiente hoja.



Proceso principal 6.- Videos

Descripción. Proceso principal Videos del sistema propuesto.

Subprocesos que integran el proceso principal 6:

Proceso 6.1 Selección multimedia: Proceso que se encarga de leer los elementos multimedia, que son videos(esteste proceso es practicamente una consulta.)

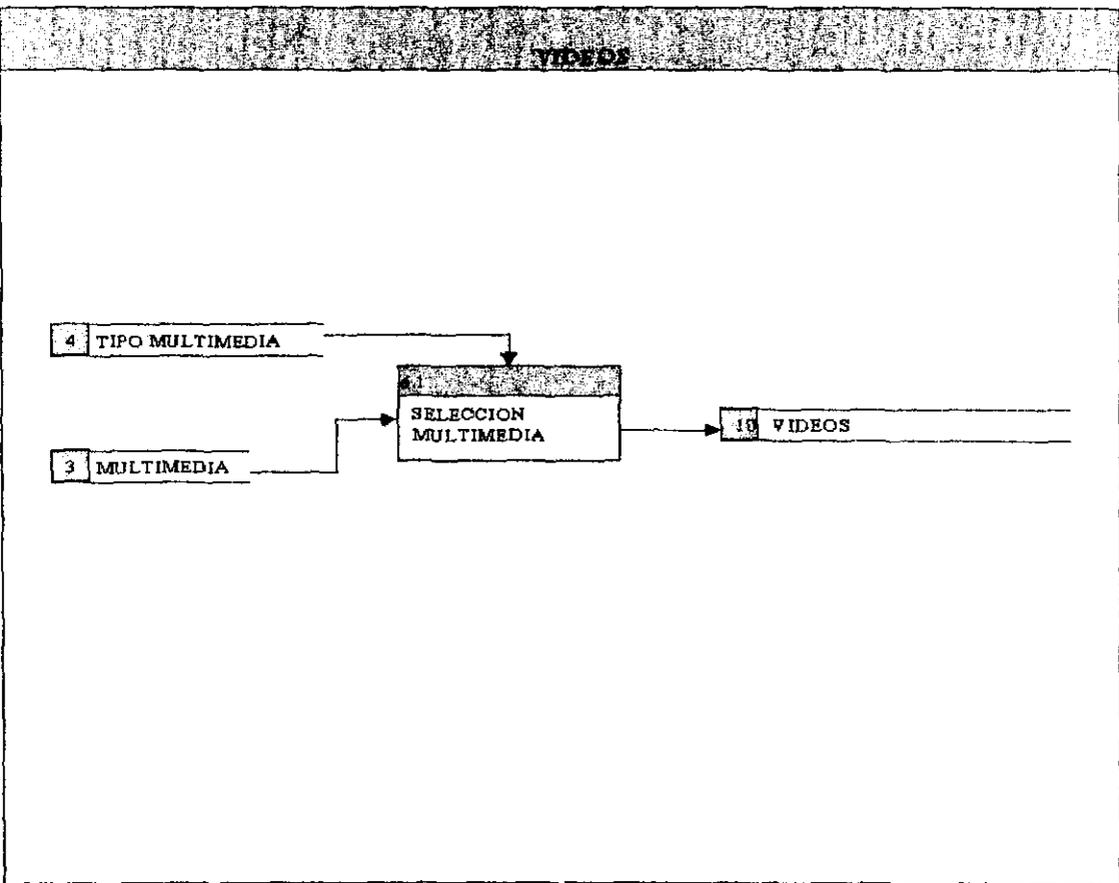
Estructuras de datos asociados al proceso principal 6

T3.- Multimedia: Catálogo de los elementos multimedia que serán utilizados por el sistema propuesto.

T4.- Tipo Multimedia: Catálogo del tipo de elemento multimedia que se esta utilizando en el catálogo multimedia.

T10.- Catálogo de los videos que pertenecen a los elementos multimedia. *(es el resultado de la consulta al catálogo multimedia)*

El Diagrama de Flujo de Datos de la presentación por módulos se muestra en la siguiente hoja.



Proceso principal 7.- Imágenes

Descripción. Proceso principal Imágenes del sistema propuesto.

Subprocesos que integran el proceso principal 7:

Proceso 6.1 Selección imágenes: Proceso que se encarga de leer los elementos multimedia, que son imágenes (este proceso es prácticamente una consulta.)

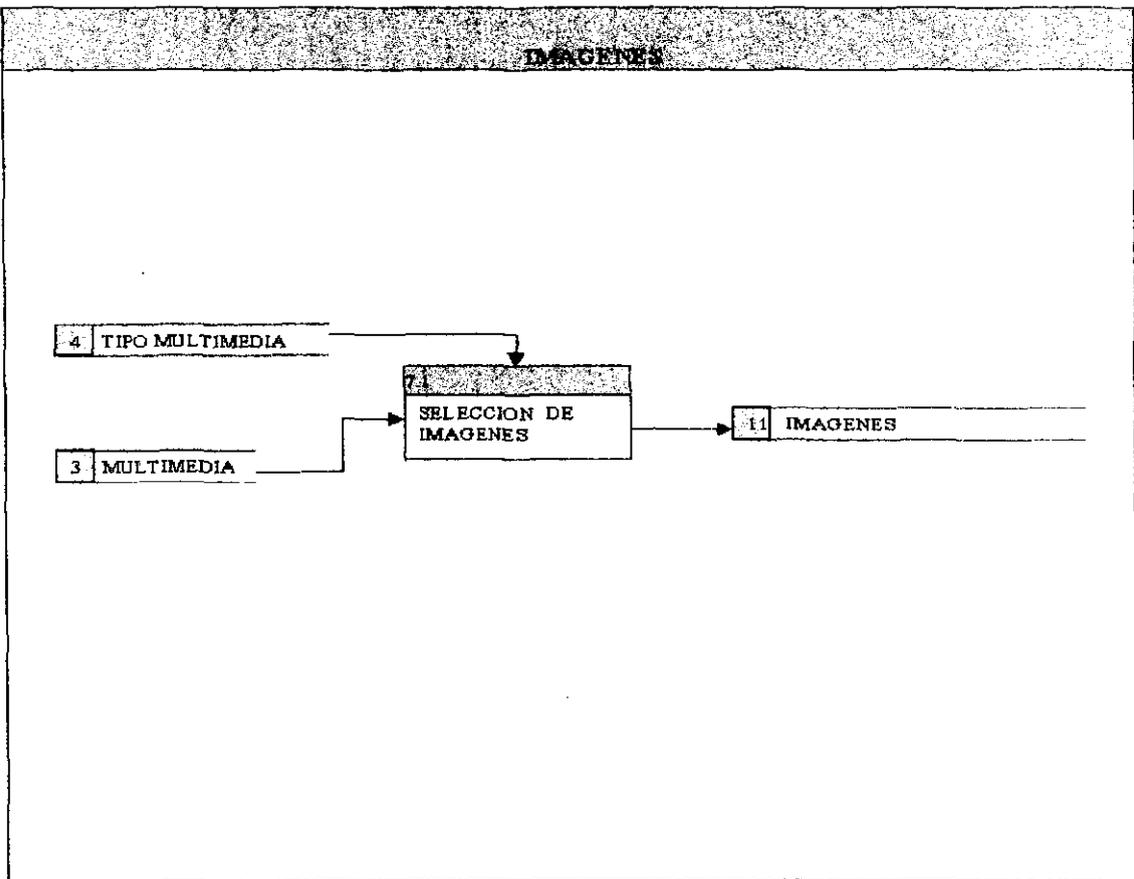
Estructuras de datos asociados al proceso principal 7

T3.- Multimedia: Catálogo de los elementos multimedia que serán utilizados por el sistema propuesto.

T4.- Tipo Multimedia: Catálogo del tipo de elemento multimedia que se esta utilizando en el catálogo multimedia.

T11.- Catálogo de los imágenes que pertenecen a los elementos multimedia. (es el resultado de la consulta al catálogo multimedia)

El Diagrama de Flujo de Datos de la presentación por módulos se muestra en la siguiente hoja.



7.8 Modelación de datos.

La modelación de datos o también llamada modelación de información, se enfoca exclusivamente en los datos. La modelación de datos, es muy usada para aplicaciones en las cuales los datos y las relaciones que gobiernan los datos son complejos. A diferencia de la aproximación hecha por el análisis estructurado, la modelación de los datos, considera datos independientemente del procesamiento que transforma los datos.

7.8.1 Datos objetos, Atributos, y relaciones.

Cuando la modelación es utilizada como una técnica de requerimientos del sistema, el análisis comienza con la creación de modelos para los datos objetos. Un dato objeto es definido en mucho de la misma forma que un objeto para un análisis orientado a objetos. Un objeto encapsula solamente datos, no existe referencia junto a un dato objeto, para operaciones que actúan sobre el dato. Sin embargo el dato objeto puede ser representado como una tabla, la figura siguiente nos muestra la forma de representación.

Las tablas de datos objetos, pueden ser formalizadas a través de la aplicación de reglas de normalización, las cuales resultan en unos modelos relacionales para datos. Las reglas de normalización, cuando son aplicados a las tablas de datos objetos, generan una mínima redundancia, esta es, la cantidad de información que se necesita para mantener satisfecho un particular problema minimizado.

7.8.2 Diagramas de entidad-relación(E-R).

La piedra angular dentro de la notación para la modelación de datos, es el diagrama de entidad-relación (E-R). EL principal propósito de los diagramas E-R es representar los datos objetos y sus relaciones entre sí.

Estos diagramas son relativamente simples. Los datos objetos son representados por rectángulos etiquetados. Las relaciones son representadas por rombos. Las conexiones entre los datos objetos y las relaciones son establecidas usando una variedad de líneas especiales de conexión. También son generadas unas tablas, las cuales contendrán los datos objetos contenidos en el diagrama entidad-relación.

Los diagramas E-R proveen un mecanismo que representa la asociatividad entre los objetos. La modelación de datos y los diagramas E-R generan una herramienta que permite realizar un análisis, con una concisa notación para la examinación de datos junto con el contexto de una aplicación de procesamiento de datos. En la mayoría de los casos, la aproximación que da la modelación de datos, es usada junto con el análisis estructurado, pero puede ser utilizado para el diseño de bases de datos y como soporte de cualquier otro método de análisis de requerimientos.

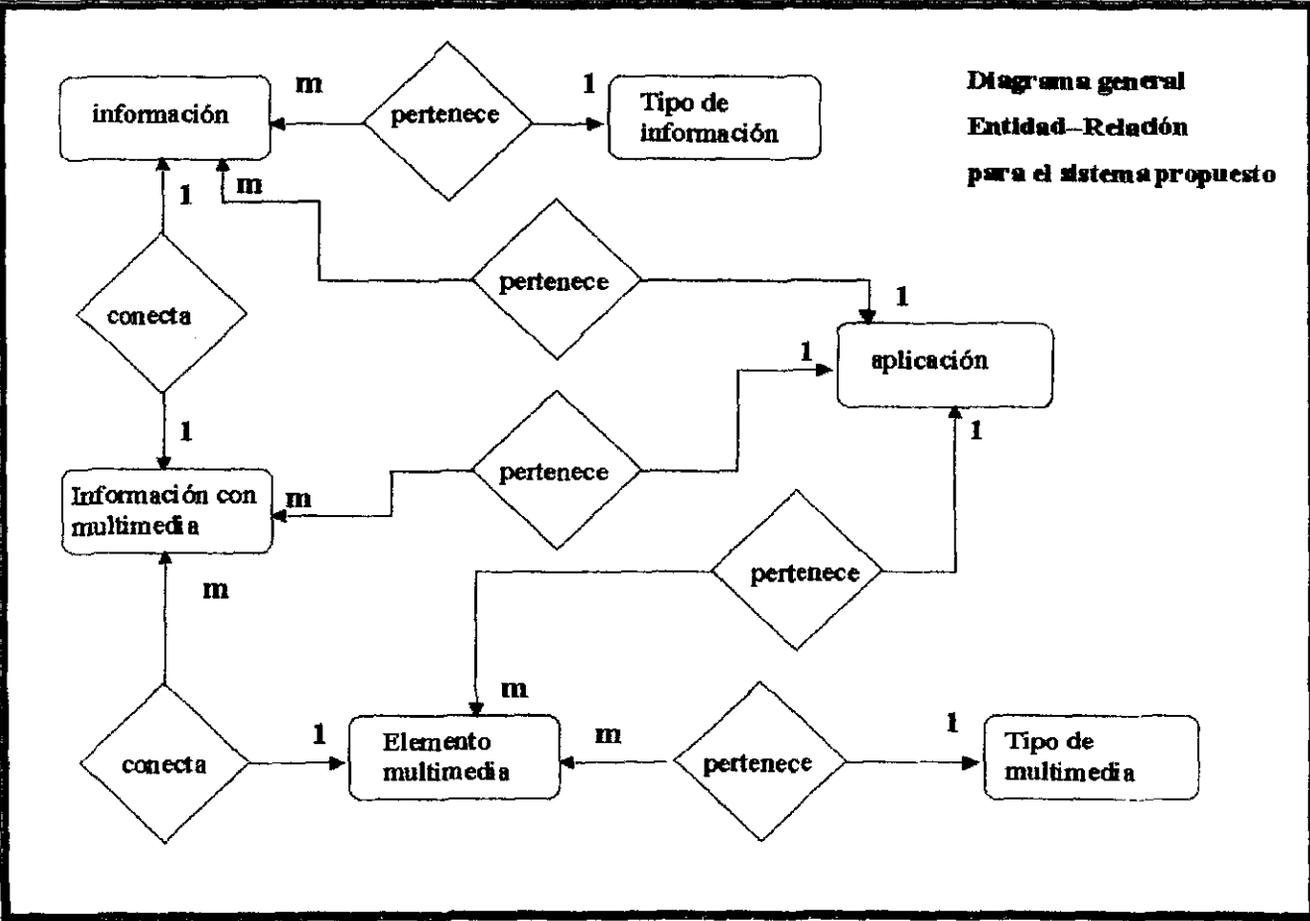
7.8.3 Diagrama de Entidad-Relación

En el diagrama de flujo de datos del sistema y en el diccionario de requisitos vistos anteriormente se reconocieron las entidades que integran al sistema en cuestión. Por la manera en que se usan y la información que contienen se definen las relaciones existentes entre ellas el tipo de éstas (el contenido de la información, se especificará en el siguiente capítulo).

Para la creación del diagrama general de entidad-relación para la base de datos del sistema propuesto, se consideraron todas las entidades (tablas) que arrojó el diccionario de requisitos.

Se realizó un análisis minucioso de la información requerida y de los datos que representan cada una de las entidades, definiendo las relaciones que existen entre ellas. Las relaciones que se pueden dar son uno a uno, uno a muchos y muchos a muchos, se definen a partir de la forma en que se relacionan los grupos de datos contenidos en las entidades.

Como resultado del análisis anterior, se diseñó el siguiente diagrama general de entidad-relación que a continuación se muestra.



CAPÍTULO 8

Diseño del sistema.

**No hemos nacido para trabajar,
Sino para disfrutar de la vida.
El trabajo es sólo una parte de ella.**

Lord Joseph Gormley.

CAPÍTULO 8

Diseño del Sistema

8.1 ¿Qué es diseño?

El diseño es el primer paso en la fase de desarrollo para cualquier sistema que se desee generar. Este puede ser definido de la siguiente manera: "es el proceso de aplicación de varias técnicas y principios para el propósito de definir un dispositivo, un proceso o un sistema con suficiente detalle para permitir la realización física de éste".

La meta del diseñador es producir un modelo o representación de la entidad para posteriormente poder ser construida. El proceso que se sigue en el desarrollo del modelo es principalmente una combinación de intuición y juicio del diseñador, que está basado en la experiencia obtenida con similares entidades.

8.2 Ingeniería y diseño

Una vez que los requerimientos del software fueron analizados y especificados toma lugar la primera de las tres actividades técnicas siguientes, el diseño, codificación y pruebas que son requeridas para construir y verificar cualquier tipo de software. Cada actividad tiene la tarea de transformar la información de tal manera que los resultados que se obtienen den validez al software para computadora.

El flujo de información durante la fase de diseño se visualiza mejor en la figura 8.1.

En ésta podemos apreciar los requerimientos del software manifestados a través de la información, funcionalidad y modelos de comportamiento, los cuales nutren la fase diseño. Se puede usar cualquier método de diseño, de cualquier forma se generarán dos procesos subsecuentes, el primero denominado diseño de datos el cual transforma el dominio de información del modelo creado durante el análisis en estructuras de datos que serán requeridas para la implementación del software; el segundo denominado diseño de procedimientos el cual transforma los componentes estructurales en una descripción procedural del software.

Esta etapa es la más importante durante la creación de cualquier software, ya que aquí podemos establecer la calidad del mismo. El diseño es la única forma que nosotros tenemos para transformar los requerimientos del cliente en un producto final de software. Sin el diseño se corre el riesgo de crear un sistema inestable, esto quiere decir un sistema que fallará cuando se le realicen los más pequeños cambios.

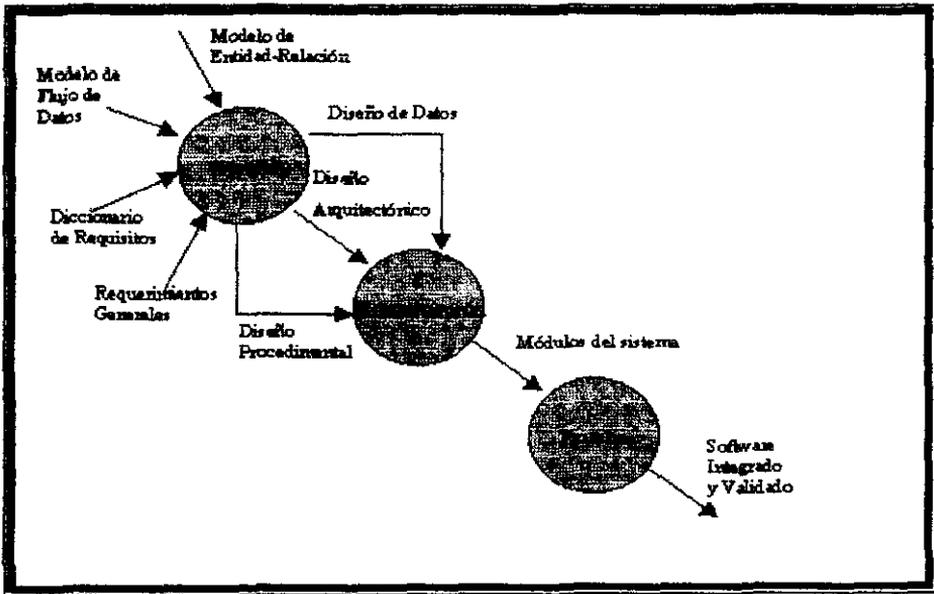


Figura 8.1 Flujo de Información de las etapas de la creación de Software

8.3 El proceso de diseño.

El diseño de software es un proceso a través del cual los requerimientos obtenidos son convertidos en una representación de software.

Desde el punto de vista de la administración del proyecto, el diseño del software se realiza en dos pasos; el primero denominado diseño preliminar el cual se centra en la transformación de requerimientos en datos y arquitectura del software; el segundo llamado diseño detallado el cual se encarga de los

refinamientos de la representación arquitectural que conduce a la refinación de las estructuras de datos y representaciones algorítmicas para el software.

Un elemento adicional es la actividad del diseño de la interfaz, la cual establece las reglas y los mecanismos de interacción entre el hombre y la máquina.

8.4 Diseño y calidad de software.

A través de todo el proceso de diseño, la calidad que envuelve al diseño es evaluada a través de una serie de revisiones técnicas formales¹³. La ingeniería del software designa procesos encaminados a obtener buenos diseños a través de la aplicación de principios fundamentales de diseño, metodología sistemática, y revisiones enteras.

El modelo propuesto para el desarrollo del sistema propuesto para la materia de Anatomía y Fisiología del Sistema Nervioso, consta de tres etapas:

1.- Diseño de Datos, empezando con la creación del diccionario de datos y aplicando las reglas de normalización para la definición de las estructuras de datos.

2.- Diseño Arquitectónico, en donde se explotan los diagramas de flujo de datos para crear la estructura del sistema orientado al flujo de datos. Es en esta etapa donde se incluye el diseño de la interfaz con el usuario.

3.- Diseño Procedimental, el que se definen los detalles algoritmos de los programas que integran el sistema.

De la misma manera que se realizó en la parte del análisis, el resultado de cada proceso que integran esta etapa es parte de la documentación de la misma.

¹³ Para ver estas técnicas se puede consultar el capítulo 17 de Ingeniería del Software de Roger S. Pressman.

Es así, que el presente capítulo describe todos los fundamentos teóricos de los procesos que estarán involucrados en el diseño y desarrollo del sistema propuesto.

8.5 Diseño de Datos

Es la primera actividad dentro del diseño de un software en la ingeniería del software. El diseño de datos tiene una gran influencia en la calidad del software, la estructura de datos, tiene un gran impacto hacia la estructura del programa y hacia la complejidad de los procesos utilizados en el software.

El trabajo principal en el diseño de datos, es elegir las representaciones lógicas de los objetos de datos, las cuales darán como resultado la estructura de datos, estos objetos de datos son identificados durante las fases de definición y especificación de requerimientos. La elección de los objetos puede involucrar un análisis algorítmico de estructuras alternativas, en orden de obtener el diseño más óptimo.

Otra actividad que tiene importancia relativa en la fase de diseño, es la identificación de los modelos de programa que debe operar directamente sobre las estructuras de datos lógicos. Esto permite restringir el ámbito del efecto de las decisiones de diseño de datos individuales.

La actividad de diseño nos aporta dos elementos más, el diccionario de datos y las estructuras de datos.

8.5.1 Diccionario de Datos

El diccionario de datos es una lista de todos los elementos incluidos en el conjunto de los diagramas de flujo de datos a través de los cuales el sistema es descrito, estos elementos son el flujo de datos, el almacenamiento de datos y los procesos. Así el diccionario de datos almacena detalles y descripciones de estos elementos.

El proceso de diseño de datos, también implica la creación del diccionario de datos, y en éste se debe especificar la nomenclatura de cada uno de los campos utilizados, así como el tipo de dato que almacena, y la mínima descripción para poder comprenderlo.

El diccionario adquiere una gran importancia cuando se quieren crear sistemas utilizando modelos de cuarta generación, esto se debe a que el diccionario se vuelve la base de esta herramienta de programación.

En la creación del diccionario de datos, los diagramas de flujo de datos y el análisis de cada flujo de información son bases importantes. El análisis agrega nuevos elementos que se convertirán en datos del diccionario. Para cada elemento se deben definir sus características como dato, es decir, el tipo de dato (entero, decimal, carácter, fecha, etc.), su formato y su longitud basándose en la información que debe contener.

8.5.2 Diccionario de Datos para el sistema propuesto.

Si deseamos construir un diccionario de datos, lo más fácil es empezar con lo más general y terminar con lo más particular, apoyándose de los diagramas de flujo de datos de sistema, del diccionario de requisitos. Este último proporciona la descripción de cada uno de los elementos que integran los diagramas de flujo de datos.

Para ejemplificar la creación del diccionario de datos, sólo tomaremos el proceso principal presentación por módulos, obteniendo los elementos que aporta este proceso al diccionario de datos.

- **Aplicaciones:** Antes de poder generar una presentación por módulos, se debe especificar el nombre la aplicación a la cual pertenece, o en su caso, generar una nueva aplicación con los diferentes nombres de los módulos contendrá la nueva aplicación, cada aplicación, sólo puede contener hasta 6 módulos.

- **Información:** Una vez generada la aplicación a la cual se le debe de asignar una clave, se debe almacenar la información (texto), que va a desplegar el módulo (no importa si es para todos los módulos, este catálogo almacenará toda la información),

- **Multimedia:** De igual manera, que se realiza con la información, se debe generar un elemento multimedia (imágenes), las cuales estarán asociadas con la información, la cual a su vez esta asociada con alguna aplicación.

- **Módulos:** Una vez que se ha generado la aplicación, se ha introducido la información correspondiente, y las imágenes adecuadas, se procede a llenar la secuencia del módulo, tomando los elementos, de la base de datos de la información. Así se genera la secuencia de las presentaciones por módulo.

El análisis que se planteo anteriormente, nos arroja una relación entre la información que se va a manejar en el sistema. Para cada campo de la información debe definirse un nombre, sus características y su descripción, lo anterior nos dará como resultado el diccionario de datos. La tabla 8.1 de la siguiente hoja muestra el diccionario de datos para el proceso principal de la presentación por módulos (Las tablas pertenecientes a los procesos restantes del sistema se presentan en un apéndice al final).

8.5.3 Estructuras de datos

Una estructura de datos es un conjunto de datos que están relacionados entre sí y que describen en forma colectiva un componente de sistema. Tanto el flujo de datos como el almacenamiento del mismo son estructuras de datos

En el capítulo anterior, se generó una modelización entidad-relación, en la cual se obtuvieron las relaciones existentes en el entorno de base de datos del sistema propuesto. Y dado que se manejara un DBMS (manejador de base de datos relacional) la estructura fundamental de los datos será precisamente bajo el modelo relacional.

La idea de usar un método relacional como estructura fundamental de un sistema de manejo de base de datos fue introducida por el DR. E.F. Codd en 1970. El proceso de cristalización de entidades y sus relaciones en formas de tablas usando los conceptos relacionales se llama proceso de Normalización; la teoría de la normalización esta basada en la observación de que un conjunto de relaciones presenta mejores propiedades en un medio de actualización que las que presentan otros conjuntos de relaciones que contienen los mismos datos.

Una de las principales ventajas del planteamiento relacional es su facilidad de comprensión por parte del usuario final; los usuarios finales no se tienen que preocupar por la estructura del almacenamiento físico, se les puede orientar hacia el contenido de la información de sus datos.

Campo	Definición	Descripción
Clave_app	numérico(2)	Clave de la aplicación
Nombre_app	carácter(30)	Nombre de la aplicación
m1	carácter(30)	Nombre del módulo 1
m2	carácter(30)	Nombre del módulo 2
m3	carácter(30)	Nombre del módulo 3
m4	carácter(30)	Nombre del módulo 4
m5	carácter(30)	Nombre del módulo 5
m6	carácter(30)	Nombre del módulo 6
Nombre_inf	carácter(30)	Nombre con el que se identificara la información
Contenido	memo	Contenido o significado de la información
Clave_inf	numérico(2)	Clave del tipo de información que se almacena
Clave_app	numérico(2)	Clave del tipo de información a la que pertenece
Nombre_mult	carácter(30)	Nombre del elemento multimedia
Path	carácter(100)	Localización del elemento multimedia
Clave_mult	numérico(2)	Clave del tipo de elemento multimedia
Clave_app	numérico(2)	Clave de aplicación a la que pertenece

8.1 Diccionario de datos para el proceso principal de presentación por módulos

Existe una manera efectiva de diseñar un modelo, llamado modelo lógico y consiste en aplicar los conceptos del modelo relacional de datos. Estos conceptos se aplican al diccionario de datos basándose en los resultados de la modelización por flujos de datos y por entidad-relación obtenidos en la etapa de análisis.

La razón de usar el proceso de normalización es asegurar que el modelo lógico de la base de datos funcionará. El proceso de normalización descompone una relación inicial universal, contenido en todos los datos del diseño de la base (tomados del diccionario de datos) en varias relaciones más pequeñas. Cada una de las relaciones resultantes se examinan para ver si es necesaria una normalización posterior. De cualquier modo de las relaciones derivadas finales, se puede reconstruir cualquier información de las relaciones originales o intermedias.

El primer paso de la normalización es transformar los campos de datos a una tabla de dos dimensiones. Lo que se requiere usualmente en este paso es la eliminación de ocurrencias repetidas de campos de datos, de manera que se obtenga un archivo fijo.

El proceso de normalización identifica los datos redundantes que pueden existir en la estructura lógica, determina claves únicas necesarias para el acceso a los elementos de datos. Pueden aplicarse tres niveles de normalización llamadas formas normales.

- **Primera forma normal:** una relación está en la primera forma normal (1FN) si todos los campos en cada registro contienen un solo valor tomado de sus dominios respectivos.

El segundo paso de la normalización es establecer las llaves y relacionarlas con los campos de datos. En la primera forma normalizada, el acceso a la tabla depende de todos los campos de datos que están relacionados con alguna parte de la llave completa. Si los campos de datos sólo dependen de un parte de la llave, la llave y los campos conectados a la llave parcial son susceptibles de separarse en registros independientes. La división de la primera tabla normalizada en una serie de tablas en las que cada campo sólo depende de la llave completa, se llama la segunda forma normalizada.

- **Segunda Forma Normal (2FN):** una relación pertenece a la segunda forma normal si esta en la primera forma y cada atributo no llave de la relación es total y funcionalmente dependiente de su llave principal.

El tercer paso en el proceso de normalización consiste en separar los campos de las segundas relaciones normales que, aunque sólo dependen de una llave, deben tener una existencia independiente en la base de datos. Esto se hace de forma tal que la información sobre estos campos pueda introducirse separadamente a partir de las relaciones en las que se encuentra implicada.

- Tercera Forma Normal: Una relación se encuentra en tercera forma normal si se encuentra en segunda forma normal y ningún atributo no llave en la relación es funcionalmente dependiente de algún otro atributo no llave.

A pesar de que algunos campos se pueden repetir en las relaciones normalizadas (aquellos usados como campos de conexión para enlazar relaciones), la normalización generalmente implica la reducción del almacenamiento de la base de datos. Sin embargo, la división de la relación original en varias tablas más pequeñas puede producir efectos negativos en el desempeño de la base por que se necesita el acceso a varias tablas antes de que se pueda contestar las peticiones realizadas por el usuario.

El objetivo del proceso de normalización es minimizar las anomalías encontradas en las operaciones de inserción, eliminación y actualización. Sin embargo para minimizar las anomalías de almacenamiento hay que pagar a cambio desempeño del sistema.

Para que el diseño de datos se pueda considerar completo es necesario realizar un proceso último a las tablas resultado del proceso de normalización; este último proceso se le conoce como desnormalización o ajustes a la normalización enfocados a obtener un mejor desempeño de las consultas realizadas a la base de datos. Es muy importante, para este proceso, considerar los tipos de acceso a la base y la información que será solicitada por los usuarios. Este proceso de desnormalización se conjuga con la definición de métodos de acceso para obtener el desempeño de la base de datos esperado y estipulado como requisito de sistema propuesto.

8.5.4 El proceso de normalización de datos del sistema propuesto.

El proceso de normalización de datos debe abarcar todo el diccionario de datos creado para el sistema propuesto. En este caso, se mostrara toda la tabla que comprende la base de datos. La relación universal definida de campos dato no es una relación normalizada de los datos utilizados en el sistema

Tabla 8.1 de datos no normalizada
Clave_app
Nombre_app
M1
M2
M3
M4
M5
M6
Nombre_inf
Tipo_inf
Contenido
Clave_inf
Nombre_mult
Tipo_mult
Path

Primera forma normal (1FN): Para obtener la primera forma normal se necesita que no existan grupos repetitivos de datos, además se necesita definir la llave primaria de la tabla. Para identificar cualquier elemento, necesitamos saber la clave de la aplicación, nombre de la información y el nombre del elemento multimedia.

Como podemos observar, ya identificamos la llave primaria, de la relación principal, pero algunos de los elementos, no se pueden identificar de manera única, ya que existen elementos nulos, es por eso que la relación se transforma en INFORMACION Y MULTIMEDIA

Tabla 8.2 Relación INFORMACION 1FN
Clave_app
Nombre_app
M1
M2
M3
M4
M5
M6
Nombre_inf
Tipo_inf
Contenido
Clave_inf

Tabla 8.3 Relación MULTIMEDIA 2FN
Clave_app
Nombre_app
M1
M2
M3
M4
M5
M6
Nombre_mult
Tipo_mult
Path

Como podemos ver se eliminaron los elementos nulos, pero todavía existen elementos redundantes y ambas tablas tienen que ser normalizadas por la 2FN.

Segunda forma normal (2FN): Una relación se encuentra en 2FN si esta en 1FN y todo atributo que no sea llave es completamente dependiente de manera funcional de la llave primaria de la relación. Esa regla separa en nuevas relaciones a los atributos que dependen de elementos que conforman la llave principal de la relación.

Como podemos observar en la relación información, existen llaves primarias que dependen de otros elementos, es por eso que al aplicar la 2FN se transforma en dos relaciones INFORMACION TIPOINFORMACION y APLICACIÓN.

De igual manera en la relación Multimedia, existen llaves primarias que dependen de otros valores, por lo que se transformara en MULTIMEDIA, TIPOMULTIMEDIA y APLICACIÓN.

Tabla 8.4 Relación TIPOINFO 2FN
Clave_inf
Tipo_inf

Tabla 8.5 Relación INFORMACION 2FN
Nombre_inf
Contenido
Clave_inf
Clave_app

Tabla 8.6 Relación APLICACIÓN 2FN
Clave_app
Nombre_app
M1
M2
M3
M4
M5
M6

Tabla 8.7 Relación MULTIMEDIA 2FN
Nombre_mult
Tipo_mult
Clave_mult
Path

Tabla 8.8 Relación TIPOMULT 2FN
Clave_mult
Tipo_mult

Tercera forma normal(3FN): Se dice que una relación esta en la tercera forma normal si no existe una dependencia funcional transitiva entre los atributos que no son llave. Esta regla define nuevas relaciones a partir de atributos que dependen de otros que no conforman la llave principal.

En este caso y basándose en libros, necesitaba realizar una conexión entre los elementos principales, los cuales pudieran manipular para efecto del programa, así es que a partir de las relaciones anteriores, se crea una relación, la cual será una estructura de datos física, la cual cumple con las características de las tablas normalizadas, las cuales son de independencia, en la inserción, borrado y modificación de datos. La tabla de la 3FN es la siguiente:

Tabla 8.9 Relación INFOMULT 3FN
Nombre_inf
Nombre_mult
Clave_app

8.6 Arquitectura del software.

El objetivo principal del diseño arquitectónico es el de desarrollar una estructura modular de fácil operación al usuario y definir la interfaz que faciliten la interacción hombre-máquina.

Una forma de definir la estructura del sistema es tomar como base los diagramas de flujo de datos explotándolos hasta llegar en cada caso, al proceso mínimo por realizar. Si los diagramas de flujo de datos llegaron al detalle de cada proceso, aquí solamente se transforman los diagramas a la estructura del sistema. La explotación genera opciones agrupadas lógicamente de acuerdo al flujo de datos al que pertenecen, por lo que el usuario (por su conocimiento de la empresa y el manejo de ésta) puede saber fácilmente en que módulo del sistema se encuentra la opción que se requiere.

La interfaz del usuario se une a la arquitectura del sistema como dos de los elementos más importantes para que el usuario entienda, confíe y utilice el sistema como parte de su trabajo diario. Si las opciones del sistema se agrupan de una manera que no comprenda el usuario, que no siga la operación lógica de la empresa o si la interfaz del sistema es muy compleja para el tipo de usuario, es predecible el rotundo fracaso del sistema.

Existen diferentes modelos de interfazs o estilos de interacción hombre-máquina, entre los que destacan: la interfaz de preguntas y órdenes, la interfaz de menú simple, la interfaz de ventanas y la interfaz de íconos. Una vez determinada la estructura o arquitectura del sistema basada en los flujos.

Esta es aludida a dos importantes características de programas de computadora: la estructura jerárquica de los componentes procedurales (módulos); la estructura de datos. La arquitectura de software es derivada a través de un particionamiento de procesos que refiere los elementos que intervienen en la selección del software.

8.6.1 Diseño Arquitectónico y de interfaz para el sistema propuesto.

La interacción que se definió para el sistema propuesto es totalmente gráfico, es decir no existen menús a través de los cuales se haga uso del programa. Ya que el programa pretende apoyar al usuario con conceptos, y elementos multimedia, se debe tener una interfaz que invite a la exploración del sistema propuesto, además de que la herramienta de cuarta generación que se esta utilizando trabaja por medio de ventana ya que corre bajo la plataforma Windows, por lo que la creación de la estructura del sistema es rápida y simple.

La interfaz esta apoyada principalmente con íconos, los cuales muestran actividades o manejo de información, los íconos permiten manejar imágenes que sean atractivas para el usuario, y motiven al uso del sistema. Los elementos multimedia son una pieza clave en el uso del sistema, es por eso que para estos elementos se utilizan botones que indican el uso de vídeo, o el cargar una imagen.

En la página siguiente , se muestran las interfazs que se utilizan en el sistema propuesto. Como podemos observar, a partir de la estructura funcional de la interfaz del software se puede construir el prototipo del sistema, este tipo de construcción, permite presentar al cliente el sistema prototipo, lo cual permitirá generar una retroalimentación a la(s) persona(s) encargada(s) del desarrollo y evaluación del sistema.

La herramienta (lenguaje) de cuarta generación que se maneja, es Visual Basic 4.0, la cual permite y apoya la utilización de prototipos debido a la rapidez y facilidad que ofrece en la creación de aplicaciones simples que reflejan lo que será el producto final. El prototipo nos permite mostrar un esquema funcional del sistema, la interfaz y las aplicaciones consideradas para cubrir los requisitos del software.

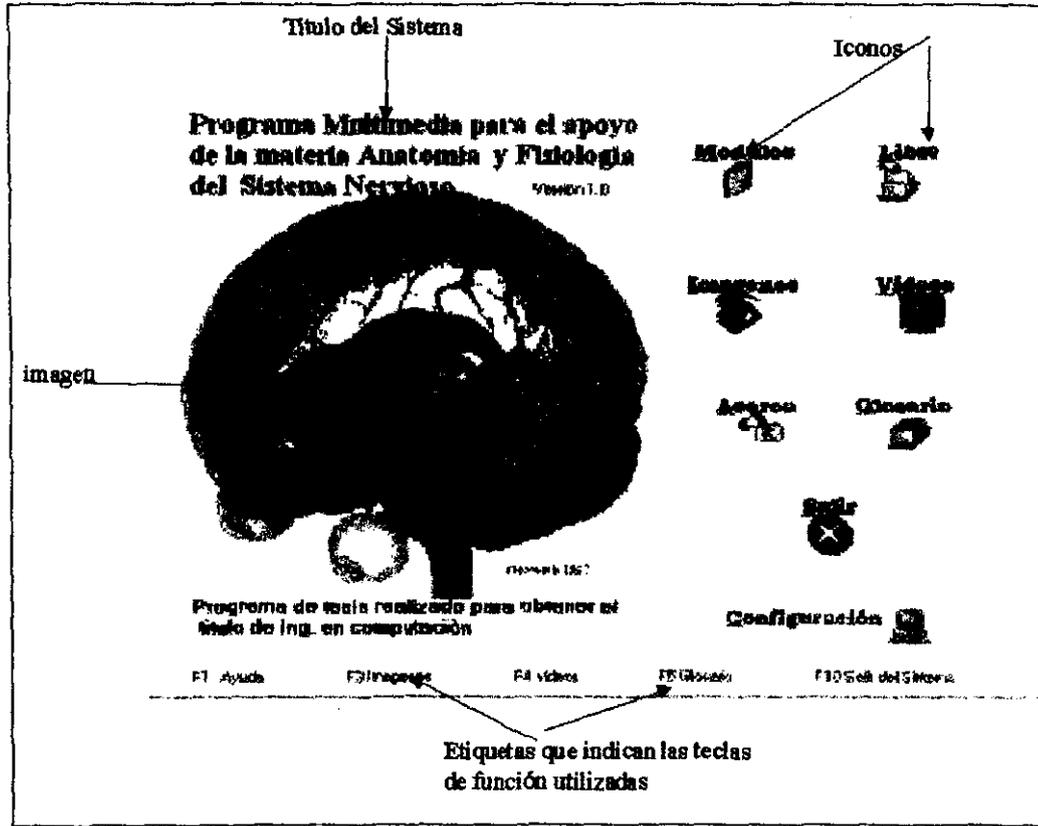


Figura 8.2 Interface del Sistema Propuesto

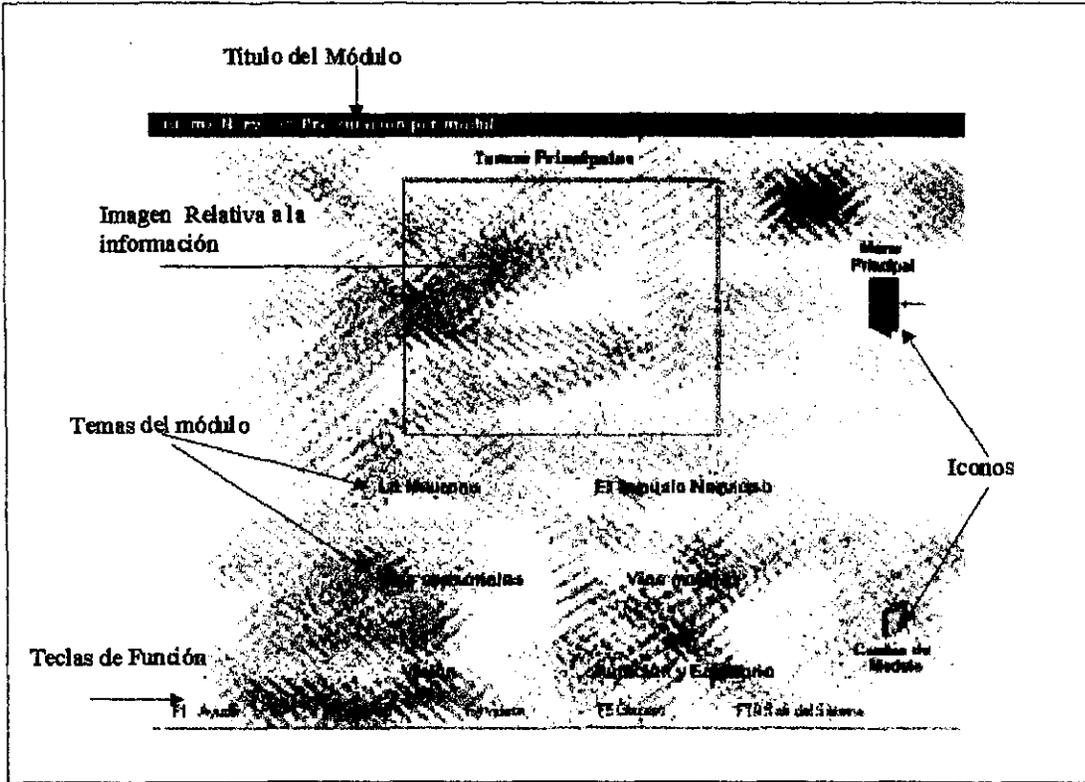


Figura 8.3 Interface de Módulos

8.7 Diseño Procedimental

El diseño procedimental se realiza después de que se ha definido la arquitectura del sistema y las estructuras que conforman la base de datos. Lo que se realiza en este proceso de diseño, es transformar los elementos estructurales en una descripción procedimental del software.

El lenguaje natural es el que utilizamos a diario para comunicarnos con las personas y esta es su principal ventaja, aunque el propio lenguaje tiene la característica de prestarse a ambigüedades, lo cual no es deseable ya que la descripción procedimental debe ser precisa y concisa.

La utilización del pseudocódigo hace que la descripción procedimental sea mucho más concisa para cada parte o procedimiento, pero al revisar el pseudocódigo de un sistema para entender la estructura y funcionamiento del mismo resulta un proceso tedioso y complicado.

La última herramienta que revisaremos es la utilización de diagramas. Los diagramas representan un gran número de ventajas con respecto a las herramientas anteriores; primero porque permite una visualización rápida y clara de la relación entre procedimientos y su forma de operación; no se presta a ambigüedades debido a que tiene que ser muy específico y cada procedimiento no involucra más de una función; existe una gran gama de simbologías y distintos métodos de diagramación, según la necesidad de los diseñadores que las utilizan.

Como se vio en los párrafos anteriores, la mejor opción para realizar el diseño procedimental es utilizar la herramienta de diagramación, por lo que en las siguientes páginas encontramos los diagramas de diseño para el sistema.

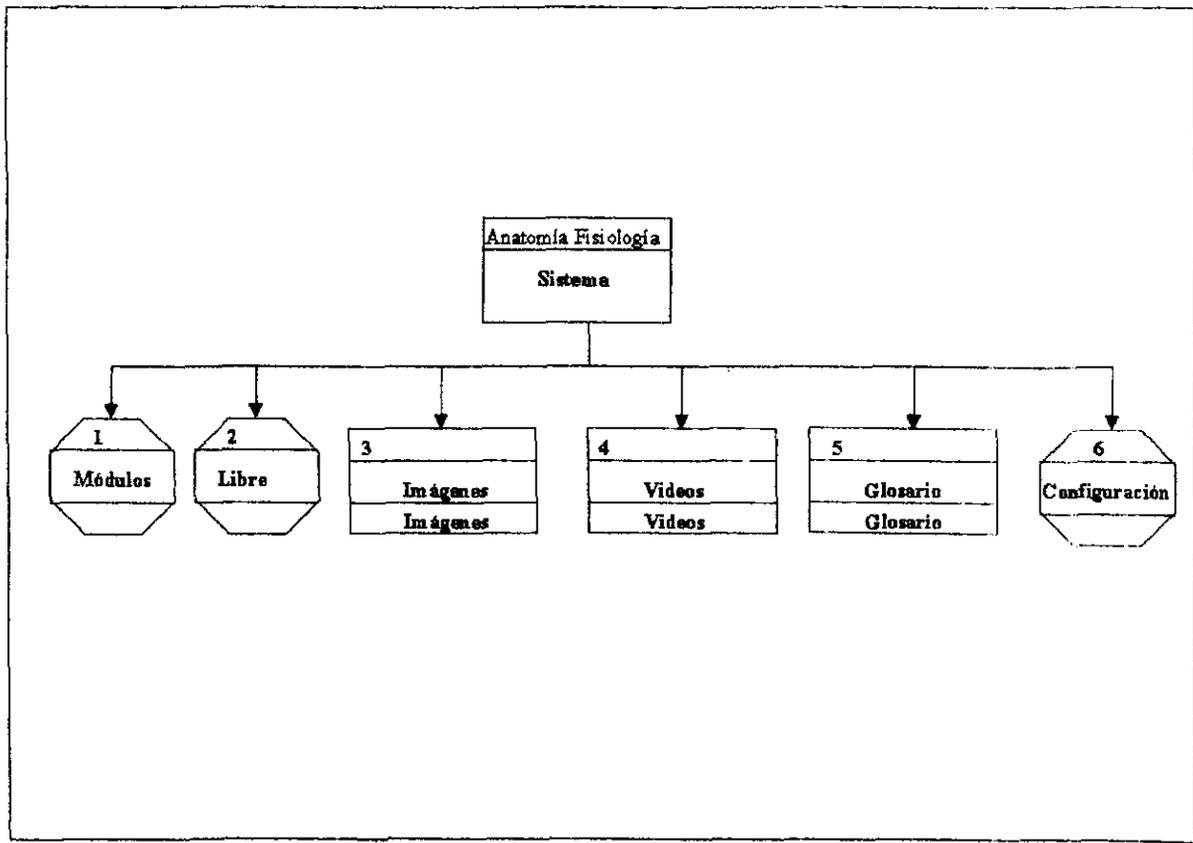


Diagrama de diseño del procedimiento principal

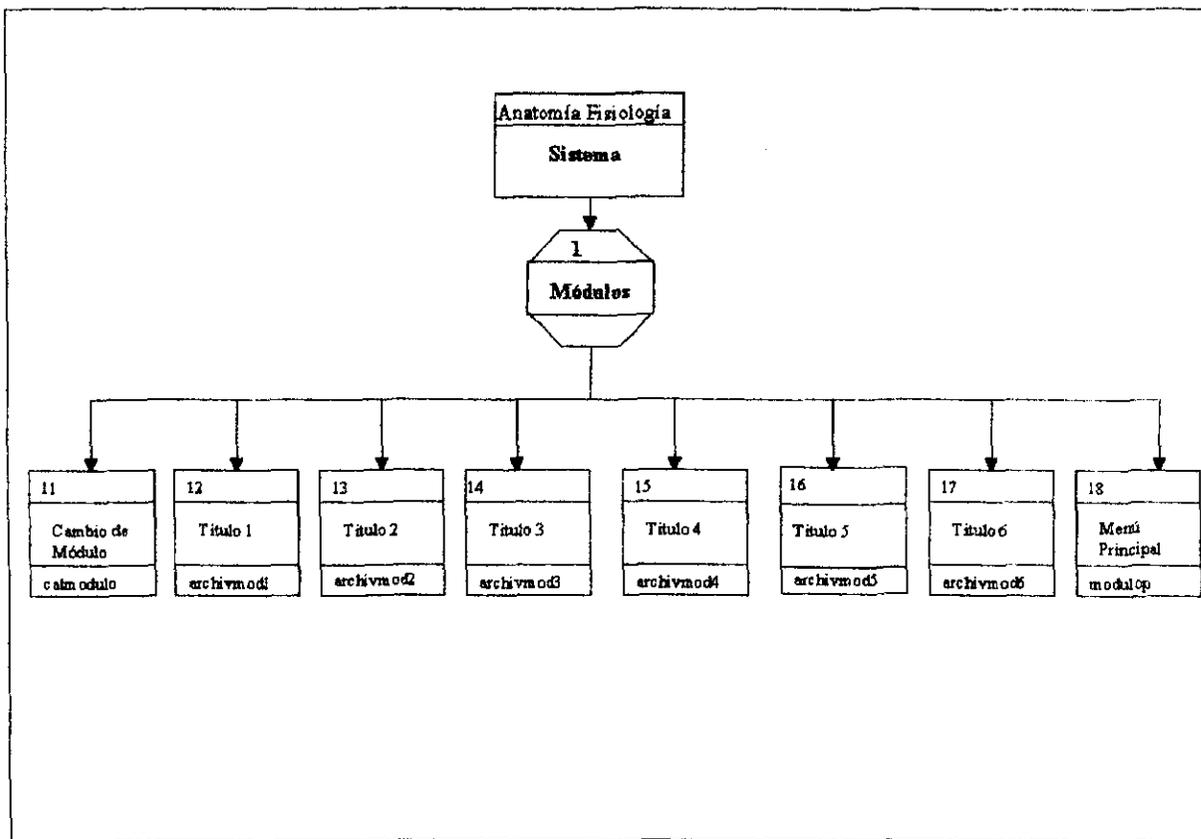


Diagrama de diseño para el procedimiento de módulo

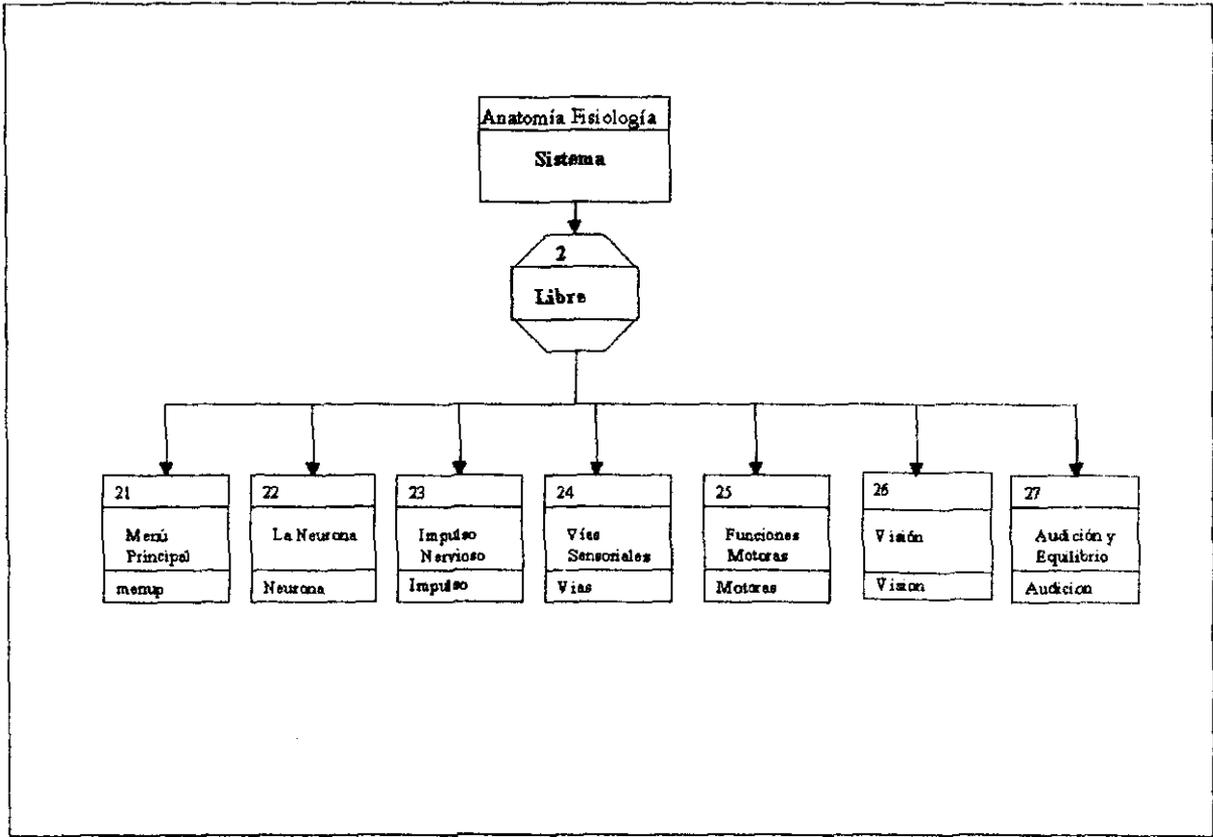


Diagrama de diseño para el procedimiento libre

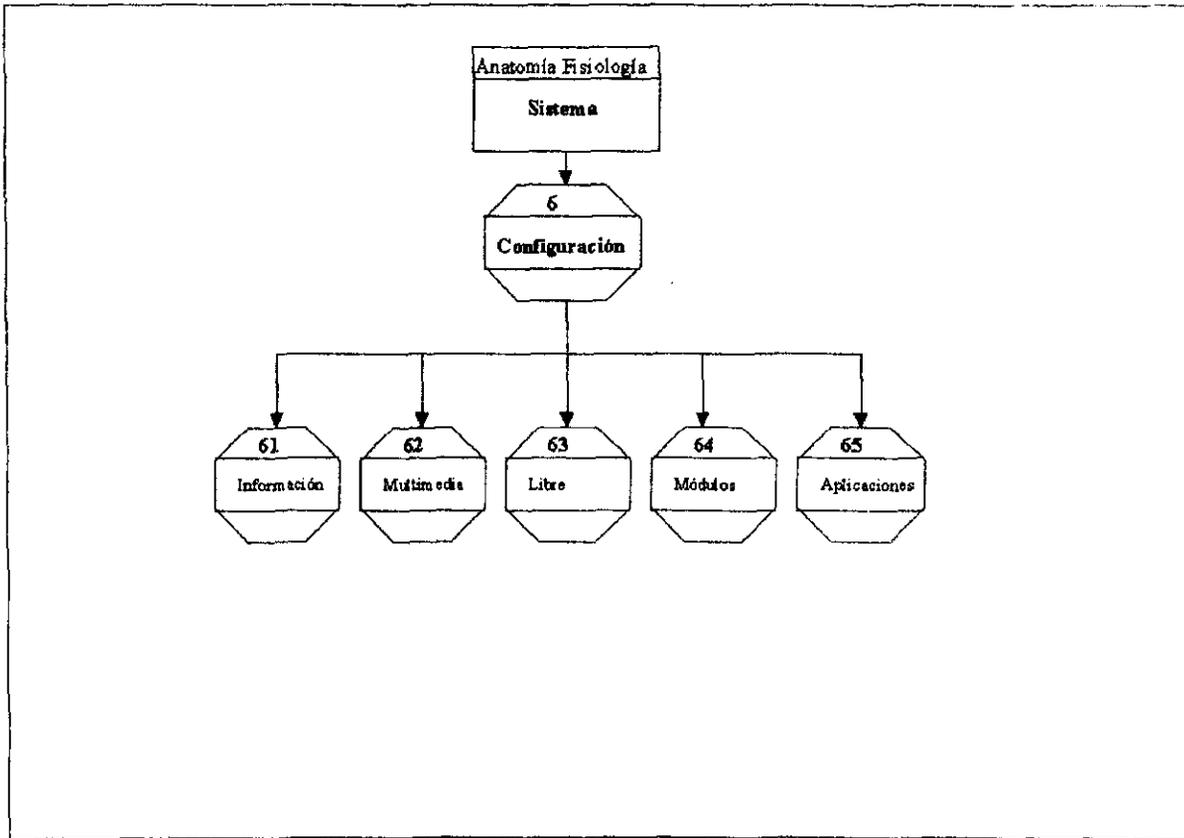


Diagrama de diseño para el procedimiento configuración.

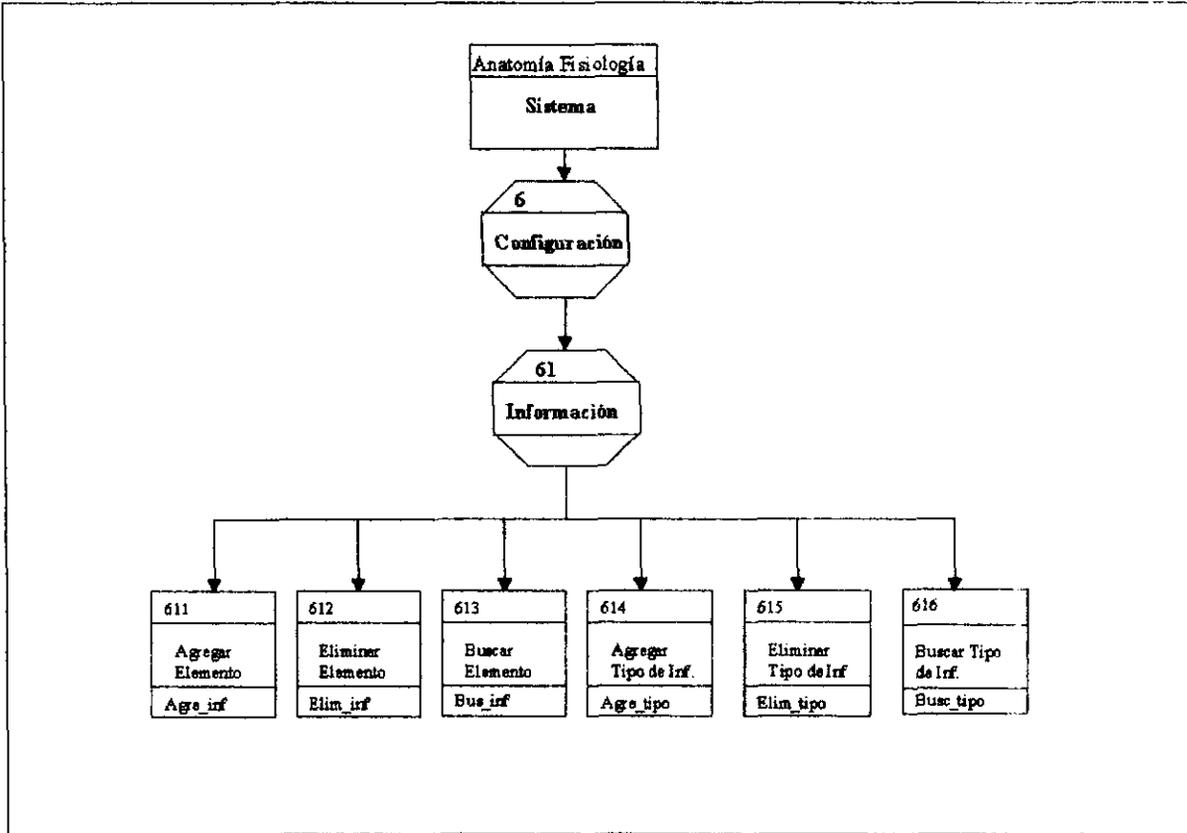


Diagrama de diseño de procedimiento configuración de información.

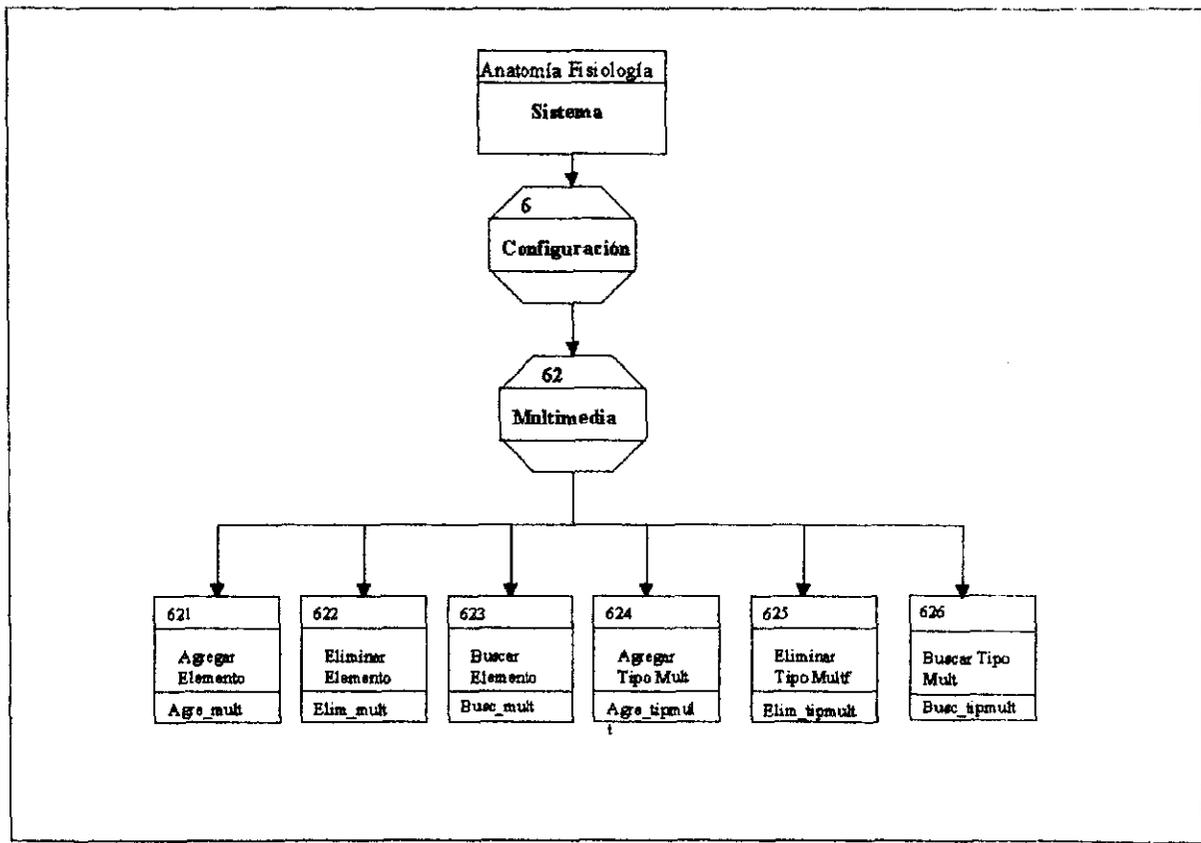


Diagrama de diseño de procedimiento configuración de multimedia

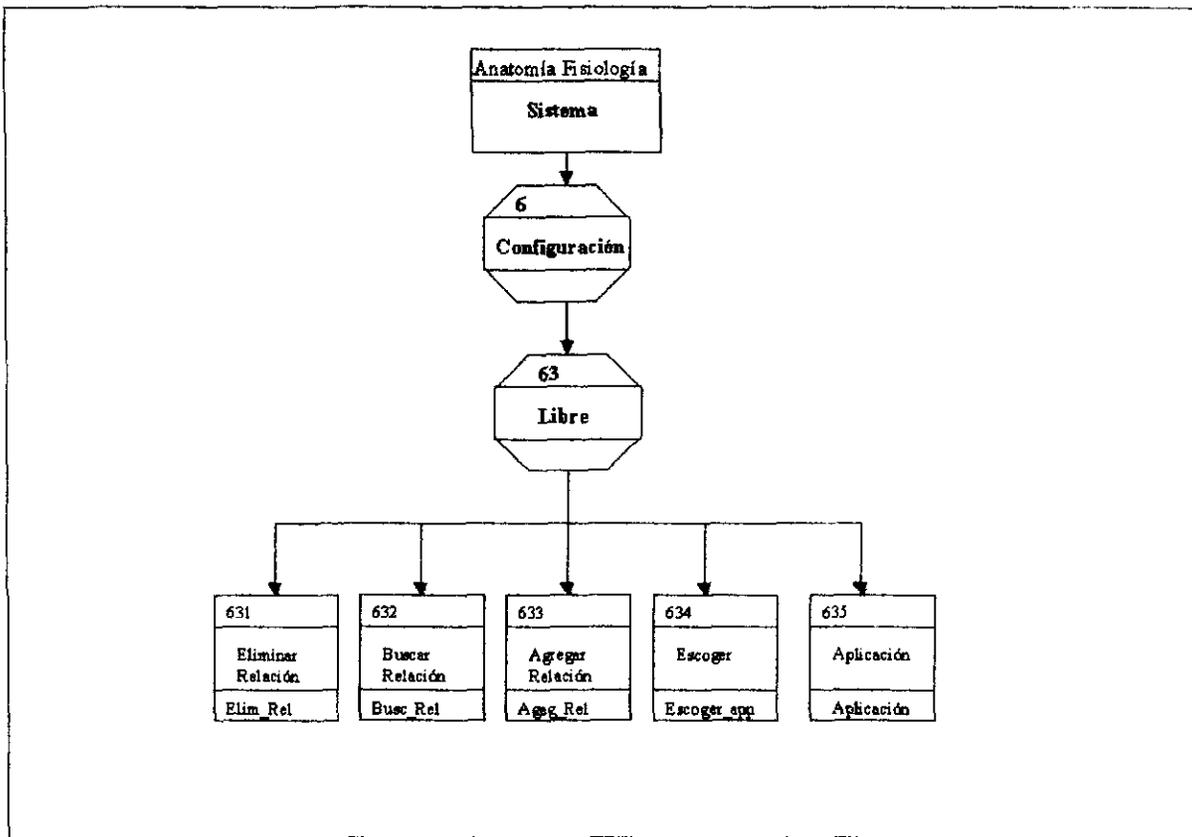


Diagrama de diseño de procedimiento configuración libre

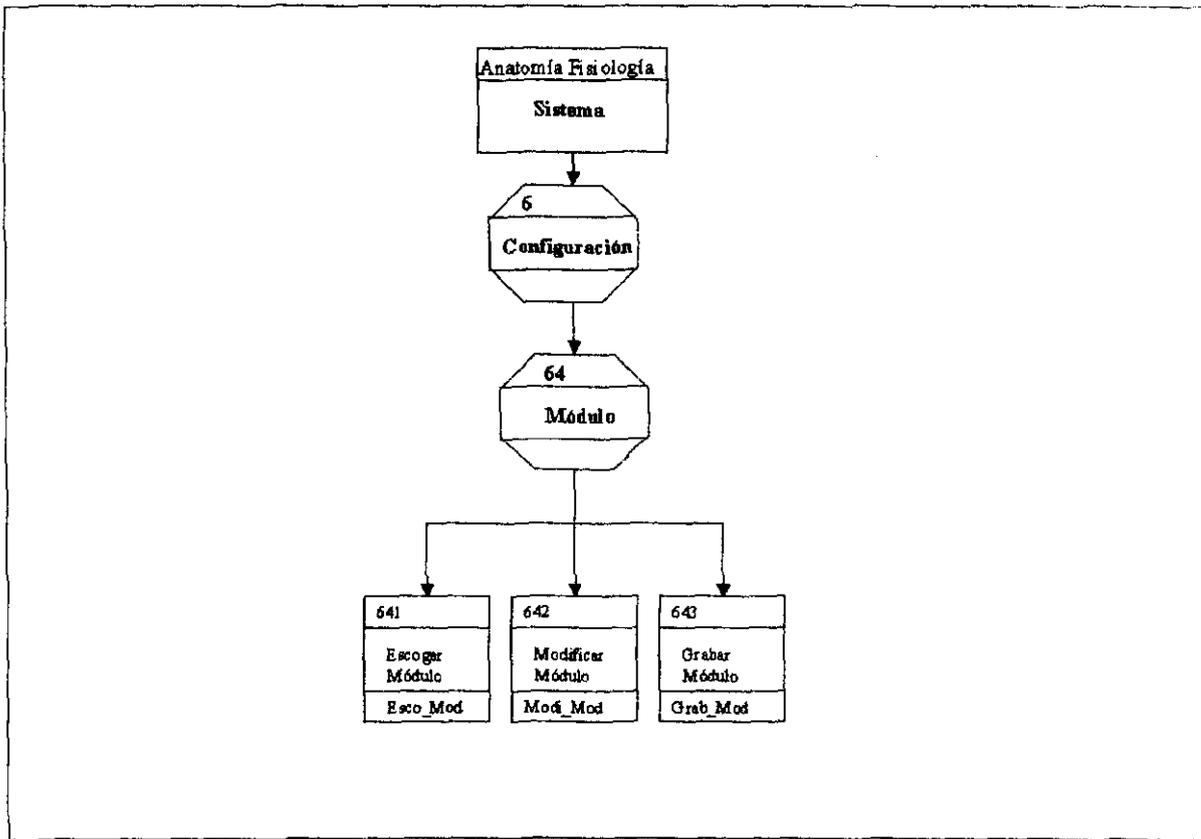


Diagrama de diseño de procedimiento configuración módulo

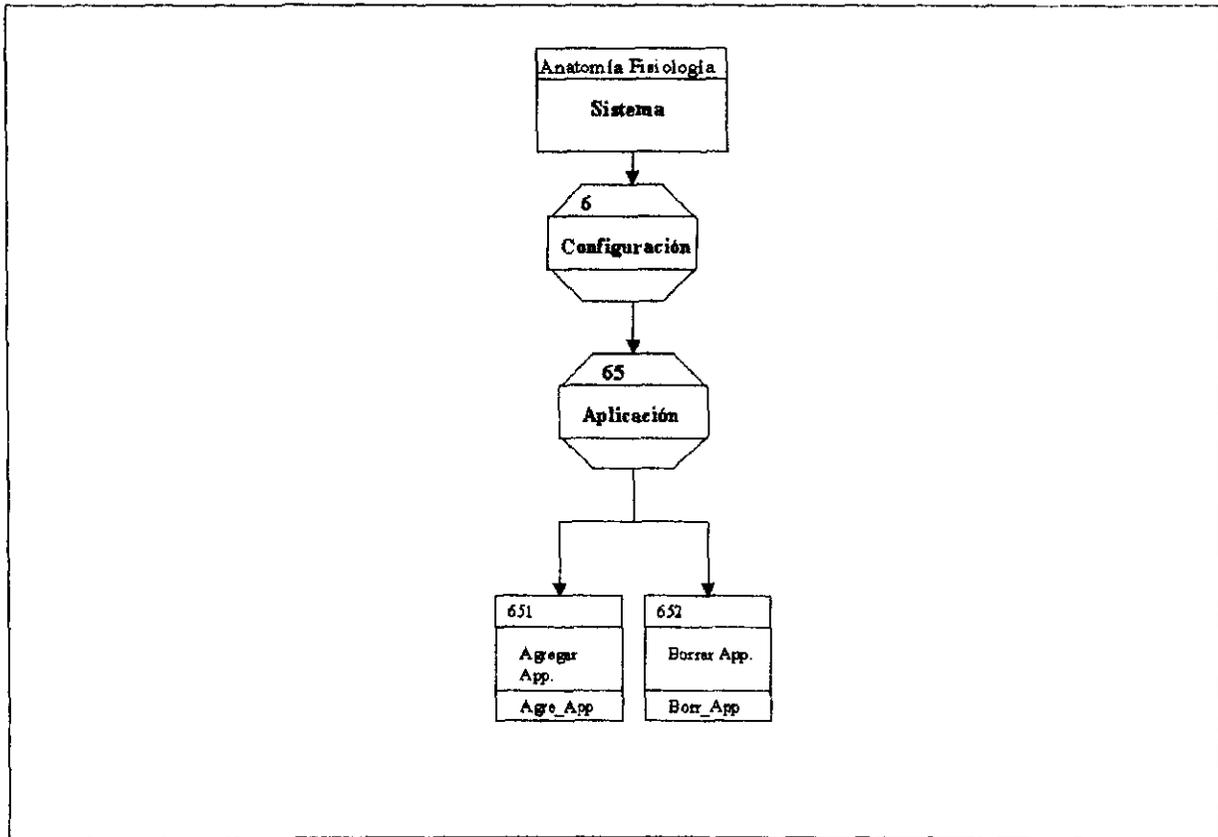


Diagrama de diseño de procedimiento configuración aplicación

CAPÍTULO 9

Desarrollo del sistema.

**Un experto es una persona que ha cometido
Todos los errores que se pueden cometer
en un determinado campo**

Niels Bohr.

CAPÍTULO 9

Desarrollo del Sistema

Los pasos anteriormente realizados de la ingeniería de software van dirigidos hacia un objetivo final: traducir las representaciones del software a una forma que pueda ser comprendida por la computadora. Hemos llegado al paso de la creación de la base de datos y la codificación de los programas. Un proceso que transforma el diseño en un lenguaje de programación y que da la pauta para la división lógica que integran la construcción de la base de datos y la programación del sistema

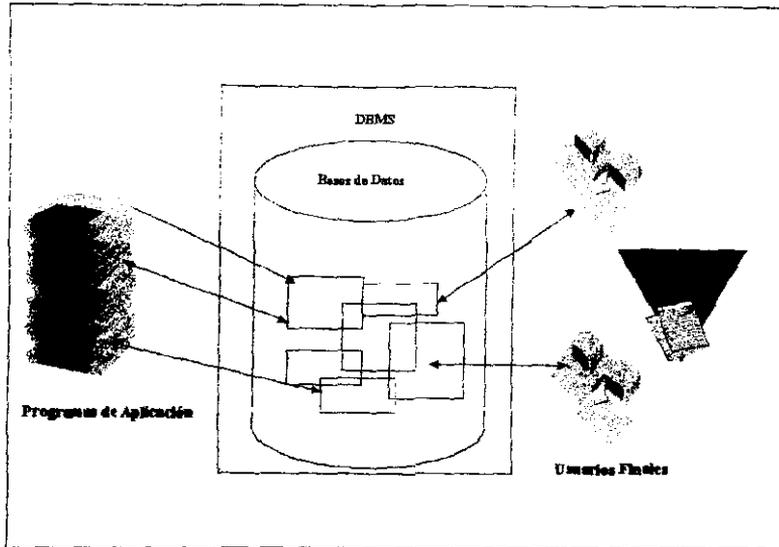
9.1 Implementación de la base de datos.

Antes de construir la base de datos del sistema, es necesario conocer algunos conceptos básicos para tener una mejor idea de que es lo que se realizó y cómo se llevó a cabo, a continuación se mencionan los conceptos necesarios para entender la construcción de la base de datos.

9.1.1 ¿Qué es un sistema de bases de datos?

En esencia un sistema de base de datos es un sistema de mantenimiento de registros basados en computadora, es decir, un sistema cuyo propósito general es registrar y mantener información. Tal información puede estar relacionado con cualquier cosa que sea significativa para la organización donde el sistema opera. En la siguiente figura 9.1 se muestra una representación simplificada de un sistema de base de datos. Los cuatro componentes principales que integran el sistema son: datos, hardware, software y usuarios.

Datos: Los datos almacenados en el sistema se dividen en una o más bases de datos. Aunque se puede pensar desde el punto de vista didáctico que sólo hay una base de datos, la cual contiene todos los datos almacenados en el sistema. Una base de datos es un repositorio de datos almacenados y en general es íntegra y compartida. Es íntegra por que la base de datos puede considerarse como una unificación de varios archivos independientes donde se elimina parcial o totalmente cualquier redundancia entre los mismos. Y es compartida porque, partes individuales de la base de datos pueden compartirse entre varios usuarios distintos, en el sentido de que cada uno de ellos puede tener acceso a la misma parte de la base de datos y utilizarla con propósitos diferentes.



9.1 Representación de un sistema de base de Datos

Hardware: El hardware se compone de los volúmenes de almacenamiento secundario (discos duros, tambores magnéticos), donde reside la base de datos, junto con dispositivos asociados como las unidades de control.

Software: Entre la base de datos física en si (es decir, el almacenamiento real de los datos) y los usuarios del sistema existe un nivel del software, que a menudo recibe el nombre de sistema de administración de base de datos o DBMS.

Usuarios: Se considera aquí tres clases de usuarios: 1) programador de aplicación, es el encargado de escribir los programas de aplicación que utilice la base de datos; 2) usuario final, es el que recurre a un programa de aplicación escrito por un usuario programador que acepte órdenes desde la terminal y a su vez formule solicitudes al DBMS en nombre del usuario final; 3) administrador de la base de datos, es el encargado de crear, mantener y administrar la base de datos.

Resumiendo un sistema de base de datos es un sistema computarizado de información para el manejo de datos por medio de paquetes de software llamados sistemas de manejo de bases de datos (Data Base Management System (DBMS)). Los componentes principales de un sistema de bases de datos son el software DBMS y los datos por manejar.

9.1.2 Sistema de Manejo de Bases de Datos (DBMS)

Un sistema administrador de bases de datos es un paquete de programación que corre en el sistema operativo central como un largo y sofisticado programa de aplicación. Contiene rutinas relacionadas que efectúan funciones especializadas para el procesamiento de las base de datos. En la mayoría de los casos el DBMS depende del sistema operativo central para el manejo de los datos.

Los DBMS se pueden clasificar dentro de cualquiera de una de las tres siguientes categorías: red, jerárquica o relacional. La clasificación se basa en la estructura lógica a nivel externo. Estas estructuras lógicas constituyen la manera en que el usuario percibe la estructura de la base que será mapeada por el DBMS hacia su almacenamiento físico.

El sistema DBMS proporciona al usuario un lenguaje especial de descripción de datos (Data Description Language (DDL)) para describir esquemas conceptuales y externos, así como distintos lenguajes de manejo de datos (Data Manipulation Language (DML)) para manejar los datos de la base. El tipo de estructura lógica de la base que usa un DBMS no sólo dicta el diseño de su esquema y subesquema DDL sino también el modo de sus operaciones de entrada-salida.

Dado que el DBMS que maneja Visual Basic es del tipo relacional, a continuación se analiza a más detalle esta categoría.

En el sistema DBMS relacional, una estructura lógica se representa por medio de tablas bidimensionales llamadas relaciones. Una entidad se representa por un renglón de la tabla.

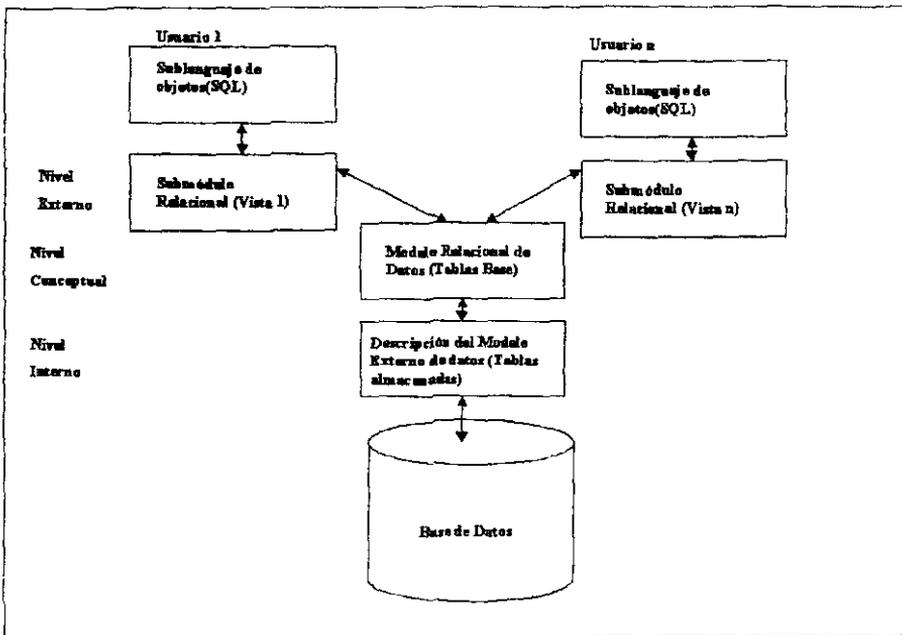
La ventaja principal que se tiene con el enfoque relacional sobre los de red y jerárquico está en la simplicidad de su representación en la estructura lógica de la base de datos y la flexibilidad para establecer relaciones de datos por medio de campos de conexión. Todas las entidades en una base de datos relacional están representadas como tablas separadas y no están colocadas en ninguna jerarquía fija, como es el caso de los árboles o estructuras plex. La mayoría de los usuarios finales probablemente entiendan las tablas mucho mejor de lo que entenderían una estructura compleja de red.

El enfoque relacional hace posible el alcanzar mayor independencia de los datos usando campos de conexión en lugar de apuntadores para enlazar registros relacionados en diferentes archivos (o relaciones).

Una característica única del sistema relacional es su independencia de trayectorias de entrada-salida. Ya que una base de datos relacional consta de una colección de tablas separadas, cualquier tabla (o relación) se puede acceder directamente sin necesidad de acceder otras relaciones basadas en una estructura de datos fija, tales como un árbol o una red. El sistema relacional permite recuperar una tabla (o un conjunto de registros en vez de un solo registro) con una sola proposición DML.

9.1.3 Arquitectura de un sistema relacional

Para explicar la arquitectura relacional utilizamos la figura 9.2, en la cual observamos tres niveles de abstracción y los componentes que integran la arquitectura relacional.



9.2 Arquitectura de un sistema relacional de base de datos

1) Esquema de almacenamiento: en el nivel interno, cada tabla base se implanta como un archivo lógico almacenado (que puede estar conformado por varios archivos físicos). Para las recuperaciones sobre las llave principal o secundaria se puede establecer uno o más índices para acceder al archivo.

2) Modelo relacional de datos: En el nivel conceptual, el modelo relacional de datos esta representado por una colección de relaciones almacenadas (también llamadas tablas base). Cada registro de tipo conceptual en un modelo relacional de datos se implanta como un archivo almacenado distinto.

3) Submodelo de datos: Los esquemas externos de un sistema relacional se llaman submodelos relacionales de datos; cada uno consta de uno o más escenarios o vistas (views) para describir los datos requeridos por una aplicación dada. Una vista puede incluir datos de una o más tablas.

Cada programa de aplicación está provisto de un buffer (también llamado área de trabajo del usuario) donde el DBMS puede depositar los datos recuperados de la base para su procesamiento, o puede guardar temporalmente sus salidas antes de que el DBMS las escriba en la base de datos.

4) Sublenguaje de datos: El Sublenguaje de datos es un lenguaje de manejo de datos para el sistema relacional. Originalmente, se propusieron dos sublenguajes de datos a) álgebra relacional, b) cálculo relacional. Estos sublenguajes fueron propuesto por Codd que además demostró que ambos lenguajes son relacionamente completos y equivalentes, esto es, cualquier relación que puede derivarse de una o más tablas de datos también se puede derivar con un sólo comando del sublenguaje. Por tanto el modelo de operación de entrada/salida en un sistema relacional se puede procesar en la forma de una tabla a la vez en lugar del procesamiento tradicional de un registro a la vez. En otras palabras, se puede recuperar una tabla en vez de un sólo registro con la ejecución de un comando del sublenguaje de datos.

El álgebra relacional es un lenguaje de procedimientos de alto nivel que permite el uso de operadores para derivar la tabla deseada en diversos pasos desde las tablas base originales en el modelo relacional, o también otras tablas derivadas intermedias. Por otro lado, el cálculo relacional es un potente lenguaje de consulta que permite al usuario la recuperación de datos mediante el establecimiento de condiciones de consulta sin necesidad de codificar en forma detallada los pasos de la recuperación.

Aunque el álgebra relacional no es tan accesible para el usuario como lo es el cálculo relacional, proporciona los conceptos básicos para el desarrollo de lenguajes relacionales de manejo de datos. Los dos principales sublenguajes relacionales, SQL (Structured Query Language) y QBE (Query By Example), están implementados con el mismo espíritu que el cálculo relacional.

Una de las principales características del SQL es que se puede usar no sólo como DML para manejar los datos de la base sino también como lenguaje de definición de datos (DDL) para definir tablas base, escenarios o vistas y los archivos almacenados en los niveles conceptual, externo e interno. El SQL es un lenguaje flexible cuyos comandos se pueden emitir ya sea interactivamente desde una terminal o incluirse en el lenguaje del programa principal. Con estas características, el SQL es el lenguaje que se ha adoptado en los principales DBMS relacionales.

5) Usuarios: Éstos realizan accesos a la base de datos mediante proposiciones del sublenguaje de datos o en el caso de los usuarios programadores, algunas veces pueden hacer uso del SQL inmerso el cual representa un acoplamiento del SQL y el lenguaje huésped. Casi cualquier proposición del lenguaje SQL puede estar inmersa en el lenguaje huésped.

Siguiendo con el tema de los DBMS relacionales, la gran mayoría de éstos constan de dos subsistemas principales:

- a) Sistema de indagación de almacenamiento (Research Storage System (RSS))
- b) Sistema relacional de datos (Relational Data System (RDS))

En esencia el RDS proporciona la interfaz del usuario externo, que soporta las estructuras de datos tabulares y los operadores sobre esas estructuras (SQL generalmente), y el RSS proporciona al RDS una interfaz de registros almacenados, a continuación consideramos con más detalle cada subsistema.

La siguiente página presenta la figura 9.3, la cual muestra como el sistema de indagación de almacenamiento (RSS) es una interfaz interna que maneja los datos almacenados, índices, buffers, distribución del almacenamiento, etc. Sin embargo, el RSS confía en los métodos de acceso subyacentes del sistema operativo para el manejo de los archivos de bajo nivel.

El RDS es una interfaz similar al sistema de control de la base de datos. Sus funciones principales son el análisis sintáctico y la optimización. Cuando el sistema recibe una proposición SQL, se le analiza gramaticalmente con base a la

gramática del lenguaje para determinar si pertenece al lenguaje y cual es su estructura. El optimizador de RDS escogerá una trayectoria eficiente de acceso y el comando SQL se traducirá a los comandos adecuados para llamar rutinas de control de ejecución de RDS.

9.1.4 Almacenamiento de los datos y métodos de acceso.

El buen funcionamiento de los programas que interactúan con la base de datos depende en gran medida de la forma en que los datos son almacenados y accedidos. Un sistema de manejo de base de datos generalizado utiliza algunos métodos de acceso del modelo interno junto con los métodos de acceso especializados disponibles por el modelo externo. El modelo interno es el modelo físico, es la forma en la cual se guarda la información y el modelo externo es el punto de vista del usuario, aunque existe el modelo conceptual, el cual tiene relación con el modelo interno la mayoría de las veces. Casi en todos los DBMS el diseñador de la base de datos tiene la flexibilidad de escoger y combinar varios métodos de acceso que estén disponibles por el DBMS.

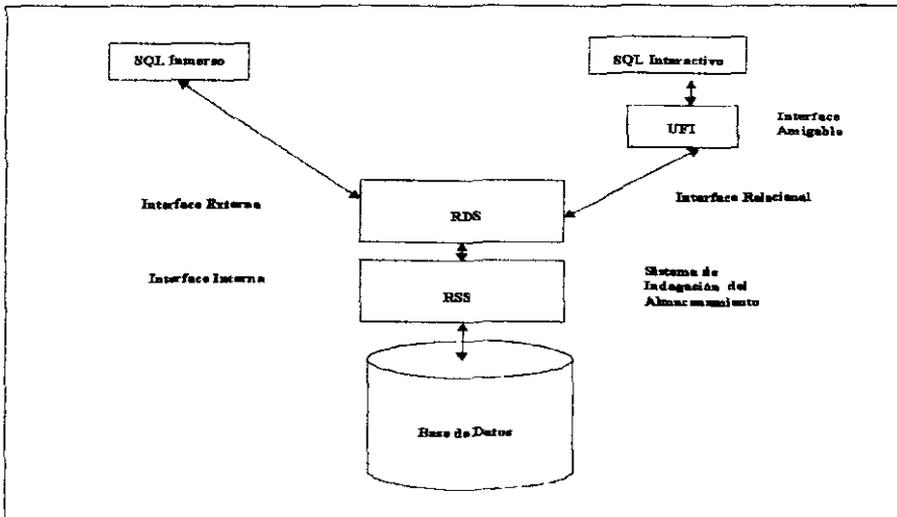


Figura 9.3 Subsistema de un sistema relacional

Al ir en aumento el número de entradas y salidas físicas que se necesitan para recuperar los datos que satisfagan la demanda del usuario, el desempeño del funcionamiento del sistema puede decrecer. El ciclo comienza con la demanda del usuario y termina con la satisfacción de ésta, pero antes tiene que pasar por varias interfazs que se muestra en la siguiente figura 9.4

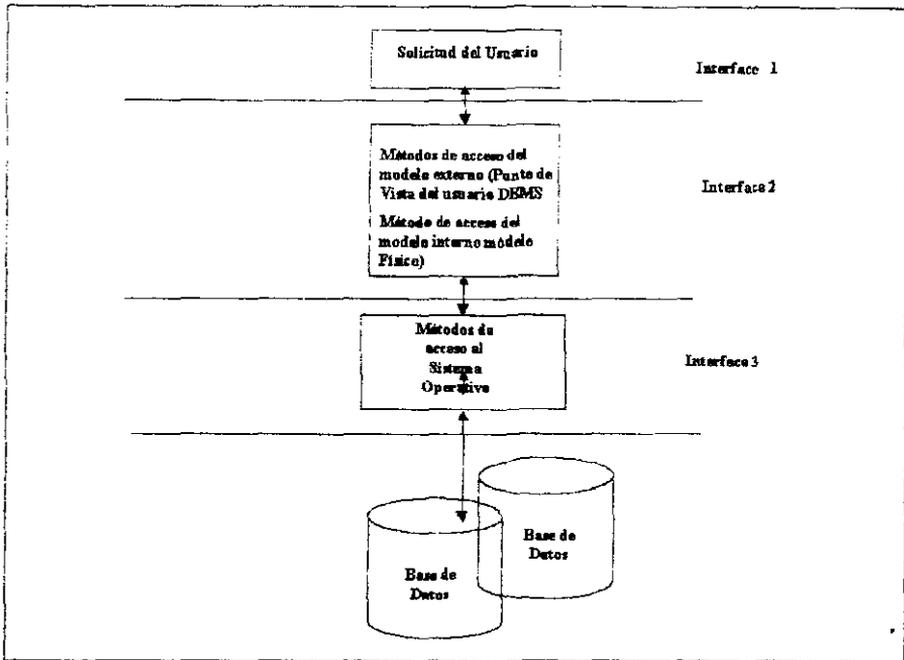


Figura 9.4 Proceso de atención a una solicitud de un usuario

Interfaz 1: Una vez realizada la demanda por parte del usuario, el DBMS reconoce la descripción del punto de vista del usuario, del programa de aplicación y de las condiciones de seguridad. Con base a las características anteriores sabe que base de datos acceder y de ésta toma la descripción de la base de datos para saber que métodos de acceso al modelo interno se pueden utilizar.

Interfaz 2: El DBMS utiliza un método de acceso al modelo interno. Los métodos de acceso son métodos especializados proporcionados por el DBMS y otros están compuestos de métodos de acceso generalizado proporcionados por el sistema operativo.

Interfaz 3: Los métodos de acceso del modelo interno y los métodos de acceso del sistema operativo accesan los registros de la base de datos física; una vez que estos registros se han accedido, se encuentran los registros requeridos utilizando los métodos de acceso que describen la relación lógica entre las distintas partes del registro de la base de datos.

Los métodos de acceso del sistema operativo y los de los modelos internos y externos nos permiten recuperar datos de la base de datos física para

presentarlos al DBMS. El DBMS tiene como tarea decir que parte de los datos se presentan al usuario, en que formato, etc.

Por lo que el diseñador de la base de datos debe tomar en cuenta todos los métodos de acceso que el DBMS ofrece y determinar el óptimo para cada caso que aparezca en los programas de aplicación. En la mayoría de los DBMS el diseñador define los índices como método principal de acceso a la información para que el DBMS, al precompilar las sentencias SQL del programa, utilice los índices para acceder los datos. Otro aspecto importante a considerar son las consultas no planeadas para incluir índices y hacer más eficiente los accesos.

9.1.5 Creación de la base de datos para el sistema propuesto.

Dado que el sistema se realizará utilizando Visual Basic y además su DMBS es el motor de base de datos Microsoft Jet, el control de datos y la interfaz de programación de objetos de acceso a datos (DAO). Debemos señalar que sus bases nativas son las creadas por Access, es por eso que podemos decir que Access será el DBMS a través del cual se creará las Bases de Datos.

Es por eso que mencionaremos características del motor de base de datos, y de Access.

Utilizando objetos de acceso de datos, puede crear bases de datos y construir aplicaciones complejas para acceder a muchos formatos de bases de datos, incluyendo Microsoft Access, Btrieve, Dbase, Microsoft Foxpro y Paradox, así como bases de datos cliente-servidor ODBC (Open Database Connectivity) como Microsoft SQL Server.

Las Tablas que maneje la base de datos, sus especificaciones, sus índices y sus llaves son también almacenadas en el mismo archivo creado anteriormente y el DBMS de Visual Basic se encarga del manejo de las tablas y de las relaciones existentes entre ellas.

Las tablas son definidas desde el DBMS (en este caso se definen con ACCESS 95), al crear la tabla, el DMBS nos permite definir los campos, así como sus características, y el tipo de datos que vamos a almacenar en cada uno de los campos también se definen las llaves primarias y foráneas que va a utilizar la tabla en diseño.

Una vez creadas las tablas, y antes de asignarle un nombre, se debe asignar los índices que se requieran, para poder acceder la información, Los índices que maneja el DBMS de ACCESS es Primario, Unico, con Repetición, y sin Repetición. Debemos aclarar que sólo podemos tener un índice primario, el cual representa la llave principal de la tabla en cuestión, con respecto a los demás índices, se pueden tener tantos índices como uno desee.

El proceso de creación de las bases de datos fue el siguiente:

1.- Creación de las bases de datos.

1.1 Seleccionar el directorio

1.2 Asignar el nombre físico y lógico de la base de datos

2.- Creación de las tablas.

2.1 Asignar el nombre de los campos

2.2 Asignar el tipo de datos que manejarán los campos

2.3 Asignar la llave primaria de la tabla

2.4 Grabar la tabla y asignar nombre

3. Creación de los índices.

3.1 Seleccionar la Tabla

3.2 Seleccionar los campos índice

3.3 Seleccionar el tipo de índice

3.4 Seleccionar si es único o no

Para la base de datos del sistema propuesto, se creo la base de datos llamada INFORMACION.MDB y las tablas que fueron creadas están definidas en el anexo 2 de este trabajo.

9.2 Programación del Sistema

La petición de servicios de procesamiento por computadora se puede codificar o expresar en un lenguaje natural, tal como el inglés. Hoy en día existe un conjunto de las llamadas técnicas de cuarta generación que han cambiado la idea del término "Lenguaje de programación". Más que codificar, los que desarrollan algún tipo de sistema de gestión pueden hoy en día escribir los resultados deseados, en lugar del procedimiento deseado. Así se genera automáticamente el código fuente en un lenguaje de programación convencional.

Cuando se considera como un paso del proceso de ingeniería del software, la codificación es una consecuencia general del diseño. Sin embargo, las características del lenguaje de programación y el estilo de programación pueden afectar profundamente a la calidad y al mantenimiento del software.

9.2.1 El Proceso de traducción

El proceso de codificación traduce una representación del software dada por un diseño detallado a una realización en un lenguaje de programación. El proceso de traducción continúa cuando un compilador acepta el código fuente como entrada y produce como salida un código objeto dependiente de la máquina. Más tarde la salida del compilador es traducida a código máquina, es decir las instrucciones reales que dirigen la lógica cableada o microprogramada de la unidad central de proceso.

El proceso inicial de traducción, del diseño detallado al lenguaje de programación, es un punto fundamental dentro del contexto de la ingeniería del software. En el proceso de traducción puede aparecer "ruido" de muchas formas distintas. La interpretación equivocada de las especificaciones del diseño o las restricciones de un lenguaje de programación pueden conducir a un código fuente muy complicado que resulte difícil de probar y mantener. Más sutilmente, las características de un lenguaje de programación pueden influir en la forma de pensar, propagando diseños de software y estructuras de datos innecesariamente limitados.

Las características del lenguaje tienen un impacto directo sobre la calidad y la eficiencia de la traducción.

Desde un punto de vista ingenieril las características de los lenguajes de programación se centran en las necesidades que puede tener un proyecto específico de desarrollo de software. Aunque se podrían derivar esotéricos requerimientos para el código fuente, se pueden establecer un conjunto general de características de ingeniería: 1) facilidad de traducción del diseño al código; 2) eficiencia del compilador; 3) portabilidad del código fuente; 4) disponibilidad de herramientas de desarrollo y 5) facilidad de mantenimiento.

El paso de codificación comienza tras haber definido, revisado y modificado en caso necesario el diseño. En teoría, la generación de código fuente a partir de las especificaciones del diseño detallado deberá ser algo directo.

El grado de facilidad de la traducción del diseño al código proporciona una indicación de cómo se aproxima un lenguaje de programación a la representación del diseño.

Por otro lado los rápidos avances en velocidad del procesador y densidad de memoria ha comenzado a disminuir la necesidad de "código super eficiente", aunque muchas aplicaciones requieren programas rápidos y "ajustados". Las críticas actuales a los compiladores de lenguajes de alto orden van dirigidas a la incapacidad de producir código ejecutable rápido y ajustado.

La portabilidad del código fuente es una característica de los lenguajes de programación se puede interpretar de la siguiente forma: El código fuente puede ser transportado de un procesador a otro y de un compilador a otro sin ninguna o muy pocas modificaciones.

La disponibilidad de herramientas de desarrollo puede acortar el tiempo requerido para la generación del código fuente y puede mejorar la calidad del código.

La facilidad de mantenimiento del código fuente es críticamente importante para cualquier esfuerzo no trivial de desarrollo de software, El mantenimiento no se puede llevar a cabo hasta que no se entienda el software. La documentación del diseño proporciona un fundamento para la facilidad de comprensión, ya que el código fuente final debe ser leído y modificado de acuerdo con los cambios en el diseño.

9.2.2 Elección de un lenguaje

La elección de un lenguaje de programación para un proyecto específico debe tener en cuenta las características anteriormente vistas. Sin embargo, el problema asociado con la elección puede desaparecer si sólo se dispone de un lenguaje o si el cliente demanda uno en particular.

Entre los criterios que se aplican durante la evaluación de los lenguajes disponibles están: 1) área de aplicación general; 2) complejidad algorítmica y computacional; 3) entorno en el que se ejecuta el software; 4) consideraciones de rendimiento 5) complejidad de las estructuras de datos, y 6) disponibilidad de un buen compilador o compilador cruzado. El área de aplicación de un proyecto es el criterio que más se aplica durante el proceso de selección del lenguaje.

En nuestro caso sólo se dispone de un lenguaje en particular y el cual como se ha mencionado anteriormente es Visual Basic

9.2.3 Visual Basic Lenguaje de 4ª Generación

Visual Basic es una combinación de herramientas visuales dentro del entorno Windows, con códigos escritos en Basic. Una serie de Herramientas nos permiten, con algunos pocos "Clicks", crear ventanas de Windows, colocarle botones de comando, menús, cuadros de texto, listas, listas descendentes, etc. Todo ello sin escribir una línea de programación. Cuando deseamos programar lo que cada uno de esos elementos visuales debe ejecutar lo hacemos en Basic, dentro de una ventana. Finalmente el producto es un programa ejecutable que actúa con todas las características de un programa que corre bajo Windows.

El hecho de que Visual Basic sea una combinación entre elementos de programación Visual del entorno Windows (ver figura 9.6 de la página siguiente) y Lenguaje Basic, no nos, debe hacer perder de vista algo mucho más esencial. En Basic, en especial en sus versiones más antiguas, todas las instrucciones se realizaban de manera totalmente secuencial. Una orden seguía a otra, a lo sumo, un comando direccionaba el flujo del programa a otro punto del mismo (GOTO, GOSUB, etc.). A medida que las versiones de Basic fueron cada vez más estructuradas, esta "secuencialidad" propia del lenguaje se fue moderando y con una programación muy sutil se podían lograr resultados mucho más flexibles.

Visual Basic de inicio, es un lenguaje orientado a determinados eventos. Si el usuario hace "click" sobre este botón se ejecutará este otro sector del programa. En resumen. Mucha libertad y procesos que respetan mucho más naturalmente la voluntad del usuario.

La interfaz de Visual Basic consta de los siguientes elementos.

Barra de Herramientas. Proporciona un acceso rápido a los comandos usados normalmente en el entorno de programación.

Caja de Herramientas. Proporciona un conjunto de herramientas que pueden usar durante el diseño para colocar controles en un formulario.

Barra de menús. Presenta los comandos que se usan para construir la aplicación.

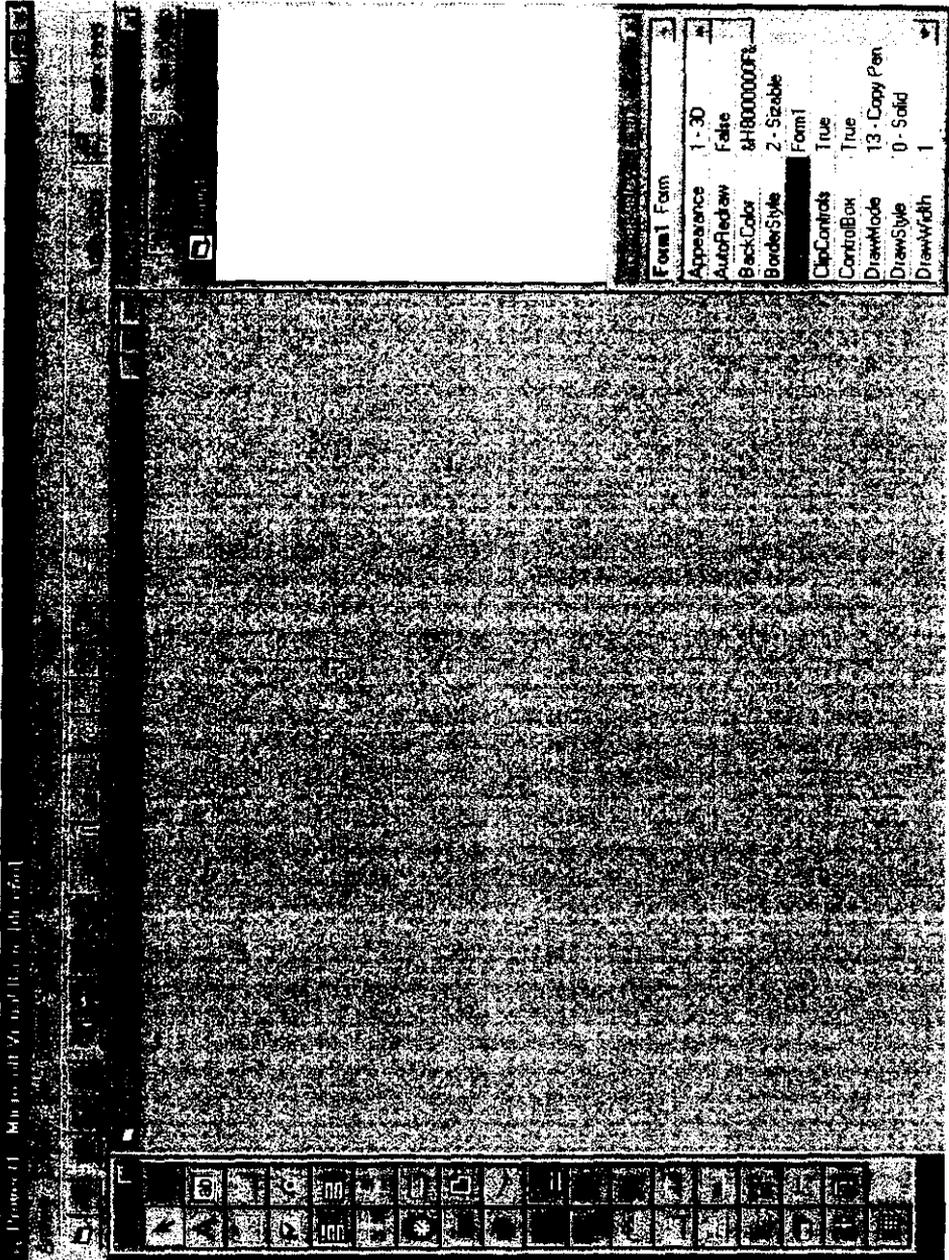


Figura 9.6 interface de la programación en Visual Basic

Ventana Proyecto. Enumera los formularios y módulos de su proyecto actual.

Por otro lado tenemos que en la barra de herramientas, existen una serie de opciones, con las cuales se cuenta para desarrollar los programas

Menú Archivo. Esta opción permite crear una aplicación en Visual Basic o seleccionar una aplicación ya existente y poder trabajar con ella.

Menú Edición. Esta opción permite realizar cualquier manipulación de edición en los formularios y en los códigos. En cuanto a los formularios, se pueden copiar y cortar todos los objetos que se encuentren en el formulario, de igual forma se puede manipular el código escrito.

Menú Ver. En esta opción, podemos manipular los elementos que comprenden la interfaz de Visual Basic, es decir podemos habilitar cualquiera de los elementos que se mencionaron anteriormente (Barra de menús, ventana de proyecto, etc.)

Menú Inserta. Esta opción nos permite agregar elementos como formas del tipo normal o MID, o módulos normales o de clase, estos elementos van formando el proyecto final.

Menú Ejecutar. Ejecuta el proyecto que se esta creando.

Menú Herramientas. Aquí se puede agregar menús personalizados dentro del proyecto, agregar controles OCX, los cuales permiten tener una mayor facilidad en cuanto a la creación de proyectos. Cabe señalar que los controles OCX es un objeto insertable que, cuando se añade a un proyecto mediante el cuadro de diálogo Controles personalizados, extiende la Caja de herramientas. Además en esta opción, se pueden establecer las Características de las Formas que se tienen en el proyecto, así como también algunas herramientas para la depuración del programa, entorno a su funcionamiento.

Menú Complemento. Esta opción permite generar reportes con la herramienta Crystal Reports para Visual Basic, la cual es compatible con Access. Además de poder agregar un administrador de Base de datos.

9.2.4 Uso de Visual Basic en el sistema Propuesto

EL uso de un L4G como lo es Visual Basic, reduce de manera muy significativa, el tiempo que se dedica a la creación de sistemas de información complejos, siempre y cuando se haga un buen uso del lenguaje y se saque el mayor provecho de las herramientas y elementos que tiene el lenguaje.

En este ambiente, la creación de prototipos, reportes y actualizaciones a bases de datos (que no sean demasiado complejas) se realizan de una forma bastante rápida y es aquí donde más se nota la facilidad que el lenguaje tiene para realizarlos.

Debemos añadir que Visual Basic es un lenguaje que permite manipular todos los elementos multimedia de una manera muy fácil y sencilla, y con pocas líneas de código, se puede crear un prototipo totalmente enfocado al entorno multimedia. Es por eso que este lenguaje es una muy buena opción para generar aplicaciones multimedia y bases de datos.

Después de la creación de las bases de datos, los pasos que se siguieron en la programación del sistema fueron los siguientes:

1. Creación de un nuevo Proyecto.
- 2.. Creación de las formas que se utilizarían en todo el proyecto.
 - 2.1 Agregar los elementos que estarían en las formas iconos manejadores
 - 2.2 Creación de los ControlData(Controles, que permiten el control de las bases de datos creadas anteriormente).
 - 2.3 Creación de los procesos con las sentencias de lenguaje en Visual Basic para los eventos requeridos de los objetos de las formas.

El lenguaje de Visual Basic, es una evolución del Basic tradicional, que surgió como un lenguaje de propósito general, esto es, las instrucciones que utiliza, caso son palabras comunes del idioma inglés, por los que su aprendizaje, programación y mantenimiento no presentan gran problema y complejidad.

Como se mencionó anteriormente, este lenguaje fue escogido, debido a sus características en cuanto a la programación y mantenimiento, y por su gran compatibilidad con el entorno de Windows, así como por su facilidad de manejo de multimedia. Además de estas características, se puede realizar una programación orientada a objetos, y crear elementos OLE.

Para obtener una idea más clara del lenguaje Visual Basic, presentamos el código que utiliza la primera interfaz del sistema propuesto.

Como podemos ver en las primeras hojas (el formulario y elementos) se muestran las características tanto del formulario como de cada uno de los elementos que conforman esta primera interfaz, debemos aclarar que todas las propiedades de los elementos, están dadas por default cuando se añade un elemento a la forma, por lo que sólo se modifican las que más interesan al diseño del prototipo, como color, tamaño, posición, etc.

En el apéndice 2 podemos apreciar que la programación, no es nada complicada, ya que para este caso estamos utilizando funciones las cuales utilizan el evento KEY UP, el cual se ejecuta cada vez que se presiona y se libera una tecla, esto es dependiendo del código de la tecla se realizara un llamado a la función que controla las opciones de la interfaz.

Con respecto al manejo de los elementos multimedia en la segunda sección casi al final de la página, podemos apreciar los elementos Mmcontrol, los cuales permiten manipular sonido, y vídeo de una forma muy sencilla, en este caso estamos manejando un archivo de audio, para poder hacer uso de estos elementos, debemos indicar el tipo de elemento multimedia y el lugar donde se encuentra almacenado, así como la instrucción de PLAY para que se comience a ejecutar el archivo de sonido.

En la página 3 de la segunda sección, podemos apreciar la forma en que se manejan las bases de datos así como la forma en que se establecen los índices principales para el manejo de la información, debemos destacar que los control Data fueron creados en otras formas y sólo se están referenciando hacia donde se encuentran.

En la página 4 podemos apreciar más claramente lo que implica la programación orientada a eventos, ya que se programo el evento de click con el Mouse, estos click pertenecen a los iconos que muestran las diferentes opciones que se tienen para poder consultar la información del sistema, además se programo el evento de movimiento del Mouse, el cual cambia de color a las etiquetas que indican la opción de la interfaz.

Pues bien como podemos ver realmente la programación con Visual Basic es muy sencilla y clara, además de que permite la manipulación de igual forma de los elementos principales para el sistema propuesto, que son las bases de datos, a través de las cuales se lleva un control de toda la información respecto a la materia de Anatomía y Fisiología de Sistema Nervioso, así como de los videos e imágenes que complementan la información de la materia.

CAPÍTULO 10

Pruebas y puesta en marcha.

**Vale más errar creyendo,
Que errar dudando.**

Amado Nervo.

CAPÍTULO 10

Pruebas y puesta en marcha

10.1 Pruebas del Sistema

La prueba del software es un elemento crítico para la garantía de la calidad del software y además representa un último repaso a las especificaciones del diseño y la codificación.

La prueba presenta una interesante anomalía para los ingenieros de software ya que en las fases anteriores de definición y de desarrollo, lo que intenta el ingeniero es construir el software partiendo de un control abstracto y llegando a una implementación tangible. Posteriormente al llegar la etapa de prueba el ingeniero crea una serie de casos de prueba que intentan demoler y acabar con el software que ha sido construido. De hecho, la prueba es uno de los pasos de la ingeniería de software que se puede ver como destructivo en lugar de constructivo.

10.1.1 Flujo de Información de la Prueba

El flujo de información para la prueba sigue el esquema que se describe en la figura 10.1. Se proporcionan dos clases de entradas al proceso de prueba: 1) una configuración del software que incluye la especificación de requerimientos del software, la especificación de diseño y el código fuente; y 2) una configuración de prueba que incluye un plan y procedimiento de prueba, casos de prueba y resultados esperados.

Se lleva a cabo la prueba y se evalúan los resultados, o sea, se comparan los resultados de la prueba con los esperados. Cuando se descubren datos erróneos esto implica que hay un error y empieza la depuración.

A medida que se va recopilando y evaluando los resultados de la prueba, comienza a vislumbrarse una medida cualitativa de calidad y fiabilidad del software. Si se encuentran serios errores que requieren modificaciones en el diseño, la calidad y la fiabilidad del software quedan en entredicho, siendo necesarias más pruebas. Si, por otro lado, el funcionamiento del software parece correcto y los errores que se encuentran son fácilmente corregibles, se puede decir que la calidad y fiabilidad del software son aceptables.

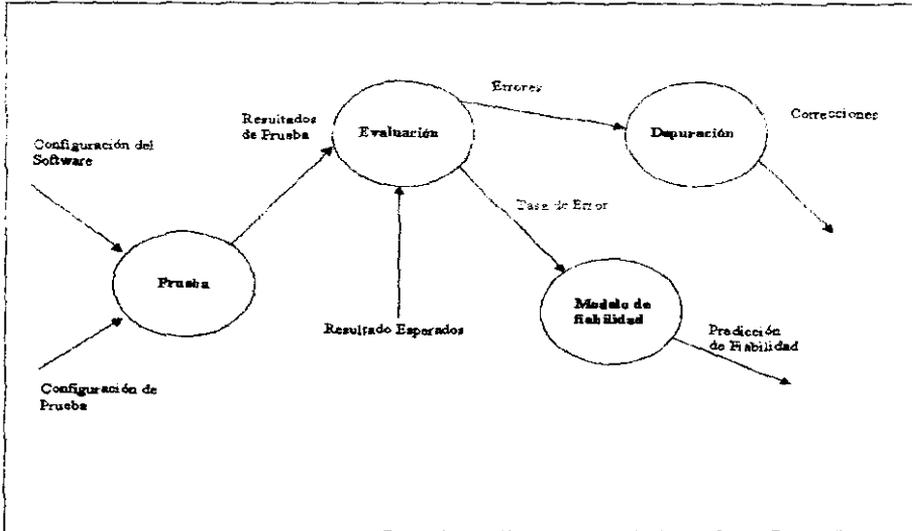


Figura 10.1 Flujo de la información de la prueba

Cualquier producto de ingeniería puede ser probado de una de dos formas: 1) conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa; 2) conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que "todas las piezas encajan"; o sea que, la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comportado de forma adecuada. La primera aproximación se denomina prueba de la caja negra y la segunda prueba de la caja blanca.

10.1.2 Técnicas de Prueba

Cuando se considera el software de computadora, la prueba de la caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Los casos pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Una prueba de caja blanca examina algunos aspectos del modelo fundamental del sistema sin tener en cuenta la estructura lógica interna del software.

La prueba de caja blanca del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejecuten casos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o afirmativo.

10.1.2.1 Prueba de la Caja Blanca

La prueba de la caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de prueba. Mediante los métodos de prueba de caja blanca el ingeniero puede derivar casos de prueba que: 1) garanticen que se ejercitan por lo menos una vez todos los caminos independientes de cada módulo; 2) se ejercitan todas las decisiones lógicas en sus caras verdadera y falsa; 3) se ejecutan todos los bucles en sus límites y con sus límites operacionales, y 4) se ejercitan todas las estructuras de datos internas para asegurar su validez.

Quizá en este punto sea válido preguntarse ¿por qué gastar tanto tiempo en minuciosidadés en lugar de gastarlo en el aseguramiento de haber alcanzado los requerimientos del programa?. Pero las justificaciones se dan a continuación.

- Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de que se ejecute un camino del programa. Los errores tienden a reproducirse en nuestro trabajo cuando diseñamos e implementamos funciones, condiciones o controles que se encuentran fuera de lo normal.

- A menudo creemos que un camino lógico tiene pocas posibilidades de ejecutarse cuando, de hecho, se puede ejecutar de forma regular.

- Los errores tipográficos son aleatorios. Cuando se traduce un programa a código fuente de un lenguaje de programación, es muy probable que se den algunos errores de escritura. Muchos serán descubiertos por los mecanismos de comprobación de sintaxis, pero otros permanecerán indetectados hasta que comience la prueba.

Las prueba de caja blanca se centran en la estructura de control del programa. Se derivan casos de prueba que aseguren que durante la prueba se han ejecutado por lo menos una vez todas las sentencias del programa y que se ejercitan todas las condiciones lógicas.

La prueba del camino básico, es una técnica de las denominadas de caja blanca, hace uso de grafos de programa (o matrices de grafo) para derivar el conjunto de caminos linealmente independientes que aseguren la cobertura. La prueba de bucles complementa a otras técnicas de la caja blanca proporcionando un procedimiento para ejercitar bucles de distintos grados de complejidad.

10.1.2.2 Prueba de la Caja Negra

Los métodos de prueba de la caja negra se centran en los requerimientos funcionales del software. La prueba de la caja negra permite al ingeniero del software derivar conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales de un programa. La prueba de la caja negra no es una alternativa a las técnicas de prueba de la caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de la caja blanca.

La prueba de la caja negra intenta encontrar errores de las siguientes categorías: 1) funciones incorrectas o ausentes; 2) errores de interfaz, 3) errores en estructuras de datos, o en accesos a bases de datos externas; 4) errores de rendimiento; 5) errores de inicialización y determinación.

A diferencia de las pruebas de caja blanca, la prueba de caja negra tiende a ser aplicada durante posteriores fases de prueba. Ya que la prueba de la caja negra intencionadamente ignora la estructura de control, concentra su atención en el dominio de la información.

Las pruebas de caja negra son diseñadas para validar los requerimientos funcionales sin fijarse en el funcionamiento interno de un programa. Las técnicas de prueba de la caja negra se centran en el dominio de información de un programa de forma que se proporcione una cobertura completa de prueba. La partición equivalente divide el dominio de entrada en clases de datos que tienden a ejercitar determinadas funciones del software. El análisis de valores límite, prueba la habilidad del programa para manejar datos que se encuentren en los límites aceptables. Los grafos de causa efecto se incluyen en una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones. Finalmente, la prueba de validación de datos asegura que los datos interactivos (conducidos por órdenes) sean procesados correctamente.

10.1.3 Estrategia de Prueba del Software

Una estrategia de prueba de software integra las técnicas de diseño de casos de prueba en una serie planificada que lleva a una construcción correcta del producto de software. Y lo que es más importante, una estrategia de prueba de software proporciona un plano para el programador, para la organización de control de calidad y para el cliente, un plano que describe: los pasos a llevar a cabo como parte de la prueba; cuando se deben de planificar y realizar estos pasos, y cuanto esfuerzo, tiempo y recursos serán requeridos.

Una estrategia de prueba de software debe ser suficientemente flexible para promover la creatividad y la adaptabilidad necesarias para adecuar la prueba a todos los grandes sistemas basados en software. Al mismo tiempo, la estrategia debe ser suficientemente rígida para promover un razonable seguimiento de la planificación y la gestión a medida que progresa el proyecto.

Existen y se han propuesto varias estrategias de prueba de software en distintos libros y trabajos, pero todas estas pruebas proporcionan al ingeniero de software una plantilla para la prueba y todas tienen las siguientes características generales:

- La prueba comienza en el nivel de módulo y trabajan "hacia afuera", hacia la integración del sistema completo.
- En diferentes puntos son adecuadas a la vez distintas técnicas de pruebas
- La prueba la lleva a cabo el que desarrolla el software y un grupo de prueba independiente.
- La prueba y la depuración son actividades diferentes, pero la depuración puede entrar en cualquier estrategia de prueba.

La estrategia de prueba que se sigue para el software producido se puede explicar al observar la figura 10.2 (Pasos de prueba del software). La prueba, en el contexto de la ingeniería de software, realmente es una serie de cuatro pasos que se llevan a cabo secuencialmente.

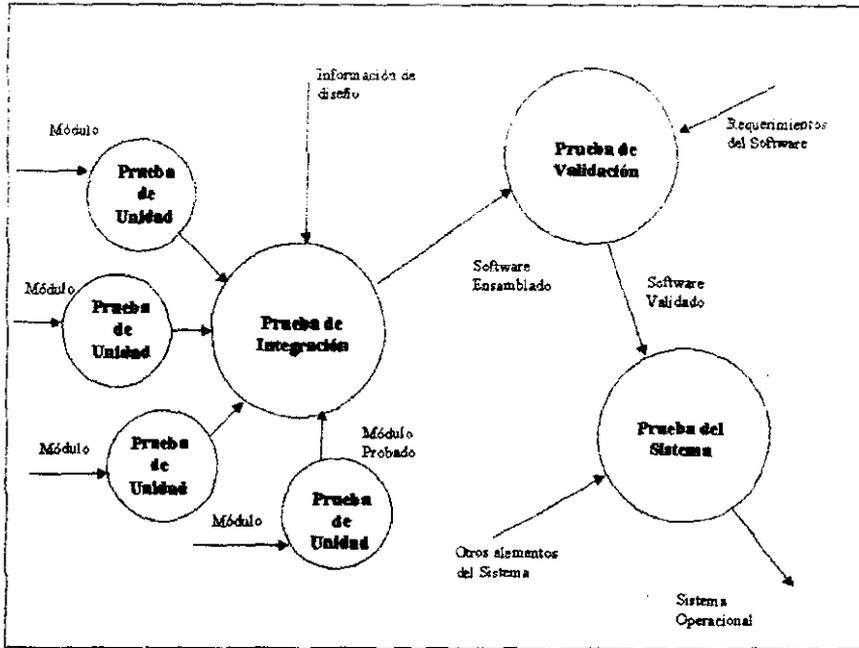


Figura 10.2 Pasos de Prueba del Software

Inicialmente la prueba se centra en cada módulo individual, asegurando que funcionen adecuadamente como una unidad. De ahí el nombre de prueba de unidad. La prueba de unidad hace un uso intensivo de las técnicas de prueba de la caja negra, ejercitando caminos específicos de la estructura de control del módulo para asegurar un alcance completo y una detección máxima de errores. A continuación se deben ensamblar o integrar los módulos para formar el paquete de software completo. La prueba de integración se dirige a todos los aspectos asociados con el doble problema de verificación y de construcción del programa. Durante la integración, las técnicas que más prevalecen son las de diseño de casos de prueba de la caja negra, aunque en algunos casos se pueden llevar a cabo unas pocas pruebas de caja blanca para asegurarse de que se cubren los principales caminos de control. Por último se deben comprobar los criterios de validación. La prueba de validación proporciona una seguridad final que el software satisface todos los requerimientos funcionales y de rendimiento. Durante la validación se usan exclusivamente técnicas de prueba de caja negra. El fin de la prueba es la prueba del sistema y aquí se verifica que cada elemento encaja de forma adecuada y que se alcanza la funcionalidad y el rendimiento del sistema total.

Prueba de Unidad

La prueba de unidad centra el proceso de verificación en lo que es la menor unidad del software "El Módulo". Usando la descripción del diseño, se prueban los caminos de control importantes con el fin de descubrir errores dentro del ámbito del módulo. La prueba de unidad siempre está orientada a la caja blanca, y este paso se puede llevar a cabo en paralelo para múltiples módulos.

En la prueba de unidad se prueba la interfaz del módulo para asegurar que la información fluye de forma adecuada hacia y desde la unidad del programa que esta siendo probada. Se examinan también las estructuras de datos locales para asegurar que los datos que se mantienen temporalmente y conservan su integridad durante todos los pasos de ejecución del algoritmo. Se prueban además las condiciones límite para asegurar que el módulo funciona correctamente en los límites establecidos como restricciones de procesamiento. Se ejercitan todos los caminos independientes de la estructura de control con el fin de asegurar que todas las sentencias del módulo se ejecutan por lo menos una vez. Y finalmente se prueban todos los caminos de manejo de errores.

Prueba de Integración

El problema que aquí se analiza es cuando los módulos independientes interaccionan en conjunto, ya que los datos se pueden perder en una interfaz, un módulo puede tener un efecto adverso sobre otro, las subfunciones cuando se combinan pueden no producir la función principal deseada, la imprecisión aceptada individualmente puede crecer hasta niveles inaceptables; y las estructuras de datos globales pueden presentar problemas.

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que al mismo tiempo se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es escoger los módulos probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.

Para llevar a cabo la prueba se utiliza el proceso de integración incremental en donde el programa se construye y se prueba en pequeños segmentos en los que los errores son más fáciles de aislar y de corregir, es más probable que se puedan probar completamente las interfazs y se pueda aplicar una aproximación de prueba sistemática.

La prueba para el software producido se llevó a acabo mediante la integración descendente; la cual como su nombre implica, empieza la construcción

y la prueba con los módulos atómicos (o sea, los módulos de niveles más bajos de la estructura del programa). Dado que los módulos son integrados de abajo hacia arriba, el procesamiento requerido de los módulos subordinados siempre está disponible y se elimina la necesidad de resguardos.

Pruebas de Validación

Una vez que el software se encuentra ensamblado y se han corregido los errores de interfazs, la prueba que sigue es la de validación. La validación se logra cuando el software funciona de acuerdo con las expectativas razonables del cliente, las expectativas razonables están definidas en la especificación de requerimientos.

La validación de software se consigue mediante una serie de pruebas de caja negra que demuestran la conformidad con los requerimientos. El plan de prueba traza las pruebas que se deben llevar a cabo y un procedimiento de prueba para definir los casos de prueba específicos que serán usados para demostrar la conformidad con los requerimientos. Tanto el plan como el procedimiento están diseñados para asegurar que se satisfacen todos los requerimientos funcionales y que se alcanzan todos los requerimientos de rendimiento.

Prueba del Sistema

Cuando el software es incorporado a otros elementos del sistema como son un nuevo hardware o información, se realiza una serie de pruebas de integración del sistema y de validación.

La prueba del sistema realmente está conformada por una serie de pruebas diferentes cuyo propósito primordial es la ejercitar profundamente el sistema. Aunque cada prueba tiene un propósito distinto, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas.

Las pruebas mencionadas anteriormente son: 1) la prueba de recuperación, para forzar a un fallo del software de muchas formas y verificar que la recuperación se lleva a cabo apropiadamente; 2) prueba de seguridad, verifica que los mecanismos de protección incorporados al sistema lo protegerán, de hecho, de la penetración impropia; 3) prueba de resistencia, para enfrentar a los programas con situaciones anormales; 4) pruebas de rendimiento, la cual esta

diseñada para probar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado.

10.2 Puesta en Marcha del Sistema

La puesta en marcha del sistema es el proceso de colocar el sistema de información desarrollado para ponerlo en operación normal en las actividades de la empresa. Esto se realiza una vez que el producto de software se encuentra totalmente desarrollado y aprobado por las especificaciones con el cual fue creado, entonces lo único que falta es insertarlo en el ambiente operativo de la empresa e iniciar las operaciones.

La puesta en marcha del sistema es de suma importancia para el éxito del mismo, ya que un inicio de operaciones deficiente o mal implementado puede causar el fracaso inicial del producto desarrollado. Para el proceso de puesta en marcha se deben considerar todos los factores que influyen en la operación de un sistema como los siguientes: factor humano (capacitación para los usuarios), factor procedimental (implementación de procedimientos para administrar la operación del sistema), factor hardware (comunicaciones, cableado, etc.), etc.

Los factores que consideramos para la puesta en marcha del sistema son los factores de capacitación a los usuarios y la carga inicial de datos.

10.2.1 Capacitación a Usuarios

Dado que la capacitación es un factor para el éxito del producto de software se pone un especial énfasis en dicho factor, ya que incluso el mejor sistema puede tener éxito o falla debido a la forma en que se opera y se utiliza. Por lo anterior la calidad con la que se da la capacitación a los usuarios que manejarán al sistema ayuda o dificulta el éxito de la puesta en marcha de un sistema de información. Las personas que se verán involucradas con el sistema deben conocer en máximo detalle las funciones que realiza el sistema, así como la forma de utilizarlo y lo que hace y deja de hacer.

La mayor parte de la capacitación de los usuarios radica en la operación del sistema mismo aunque en ocasiones la capacitación abarca el empleo del equipo y en estos casos los usuarios deberán recibir primero la capacitación respecto a la operación del equipo.

Dentro de la capacitación, se destacan las actividades de manejo de datos que reciben la máxima atención durante el proceso de capacitación de usuarios y los cuales son los siguientes, entrada de nuevos datos (es la inserción de nuevos datos a los ya existente); edición de datos (como cambiar datos ya existentes) y la consulta de datos (búsqueda de datos almacenados previamente). La parte más importante del uso del sistema incluye el conjunto de actividades descritas anteriormente, por lo que la capacitación ocupa la mayor parte de su tiempo en estas actividades.

Los dos aspectos más importantes que se manejarán en la capacitación a los usuarios son: 1) la familiaridad con el sistema de procesamiento, es decir, el equipo que se usa; 2) la capacitación en el empleo de la aplicación, es decir, en el software que recibe los datos, los procesa y genera un resultado. Es importante resaltar que si se tienen carencias en cualquier aspecto de la capacitación probablemente se terminará por hacer fracasar la utilización del sistema.

CONCLUSIONES.

Cada vez que cometo un error me parece descubrir una verdad que aún no conocía.

Maurice Maeterlinck.

CONCLUSIONES .

Después de haber implementado y desarrollado el sistema de apoyo para el sistema de universidad abierta que se propuso desde el principio de este proyecto de tesis, así como la metodología sugerida, podemos mencionar y evaluar los resultados que se obtuvieron, así como los aspectos involucrados a través del desarrollo del proyecto.

El enfoque de la programación orientada a objetos y eventos, permitió dar un enfoque más claro del objetivo de proyecto, esto en contraste con el enfoque que ofrecía la programación estructurada, este tipo de programación no permite dar sentido a los eventos ya que su metodología es lineal y se tiene que tener bien definido cada paso.

La programación orientada a objetos y eventos, nos da la pauta para diseñar y construir sistemas con múltiples elementos (objetos), y es sobre éstos donde recae el principal soporte del programa, ya que éstos tendrán una gran variedad de eventos, los cuáles nos podrán dar una gran versatilidad en los sistemas finales. Esto nos arroja obtener un conocimiento exacto de la cantidad de eventos que puede tener un objeto, así como la relación que pueda tener con otros objetos y eventos.

Al hablar de programación orientada a objetos debemos señalar que debemos tener claro cuáles son las bases de la programación orientada a objetos, debido a que se tiene una gran similitud con la programación orientada a eventos, y esto nos da una gran ventaja en este nuevo tipo de programación.

Puedo decir que la programación orientada a eventos, es una herramienta que nos permite generar sistemas que puedan ser usados en forma más natural, es decir que no sean tan metódicos (como los que se desarrollaban con la programación estructurada), además de que se pueden diseñar e implementar en menos tiempo.

Otro punto que merece ser resaltado, es la interfaz gráfica que adoptan casi todos los entornos de desarrollo de programación orientada a eventos, este tipo de interfaz tiene un impacto muy positivo en los usuarios finales, ya que invita al uso de los sistemas, dando un manejo más eficaz y sencillo. Además debido a que la

interfaz esta basada en el ambiente Windows, podemos tener varias aplicaciones (conocidas como ventanas) trabajando al mismo tiempo, lo cual da una mayor comodidad al usuario.

Referente al punto anterior, quiero comentar que fue un punto de apoyo en la propuesta del proyecto, ya que se quería obtener un ambiente, que a través de imágenes, íconos, sonido, etc. proporcionara un uso sencillo y agradable para el usuario final. Esto pensando en que la mayoría de los usuarios finales han trabajado con Windows o aplicaciones que corren bajo este esquema, para el sistema que se desarrolló, se estableció que el sistema sería totalmente gráfico, y con poco uso del teclado.

Pasando a la funcionalidad que proporciona el sistema, debo comentar que la finalidad era agrupar toda la información en torno a la materia de Anatomía y Fisiología del Sistema Nervioso, y con esto reducir el tiempo que necesita el alumno al consultar y compilar la información. Otro aspecto que proporciona el sistema, es que podemos agregar información que tenga que ver o no con la materia, es decir podemos generar módulos que ayuden a comprender mejor la materia, o algún punto en particular.

La información de anatomía no es muy fácil de comprender y manejar, y es en este sentido que la tecnología multimedia, aunada a la programación orientada a eventos nos proporcionan una poderosa herramienta, ya que estamos brindando a través de sonido, imágenes, y vídeo, una forma más tangible para que el usuario (alumno) comprenda el funcionamiento del cuerpo y sus funciones, así como también una forma menos técnica del uso de conceptos que en su gran mayoría son enfocados a los médicos.

Es aquí donde los lenguajes de cuarta generación, como Visual Basic, son una gran herramienta en la construcción de prototipos, ya que reducen el tiempo de desarrollo y construcción del sistema. La ventaja que nos proporciona entorno al análisis y diseño es grandísima, ya que sabemos que podemos brindar mayor tiempo a estas dos etapas que son vitales para el producto final.

Para el desarrollo del sistema, fue importante definir cuál era el lenguaje de cuarta generación que nos ofrecía las mayores ventajas con respecto al manejo de elementos multimedia, el manejo de bases de datos, y la interrelación de ambos elementos. Esto nos permite afirmar que el conocimiento del L4G fue vital para

obtener los resultados deseados. Además de los puntos anteriores, el uso de un L4G nos permite realizar una codificación más rápida y sencilla, siempre y cuando se tenga conocimiento y experiencia del nuevo enfoque que proporciona este nuevo tipo de herramientas.

El problema principal al que se enfrentó, fue idear cuál era la mejor manera, de conjuntar toda la información que se pretendía usar, así como cuál sería el elemento multimedia a utilizar. Es aquí donde se optó por utilizar las bases de datos, a través de las cuales podíamos controlar la información, así como también el tipo de elemento multimedia que le asignaríamos.

Pero lo anterior no fue tan sencillo, ya que al principio las posibles relaciones entre información y elementos multimedia, arrojaba bases de datos demasiado complejas, es decir que tenían mucha información duplicada, o en algunos casos información que no era necesario asignarle elemento multimedia. Para este problema, fue necesario y esencial aplicar la normalización de la base de datos, y así se obtuvieron bases más simples, pero que contemplaban toda la información sin duplicidad, no espacios vacíos, es por eso que debo señalar que la normalización en cualquier sistema que pretenda utilizar bases de datos es fundamental para obtener un funcionamiento óptimo y eficaz.

La combinación de las metodologías usadas para el desarrollo del sistema fue de gran importancia ya que la combinación de los modelos tradicionales del software con la utilización de herramientas de cuarta generación nos permitió generar el software que se propuso desde el principio.

Al rescatar las principales ventajas que nos ofrece la metodología de generación de prototipos y combinarla con la metodología L4G y también sus principales ventajas, nos permitió elaborar un producto con mayores ventajas, que si hubiéramos seguido forzosamente una metodología individual.

La generación de un sistema multimedia, que está enfocado a proporcionar un elemento más en la enseñanza, no es sólo trabajo de una sola persona, ya que como ingeniero, puedo sugerir y proporcionar un análisis de cuál es la mejor herramienta y metodología para desarrollar el sistema multimedia, pero también son importantes el contenido, así como a quién va dirigido, que método de

enseñanza se va a utilizar, cuales son las imágenes a utilizar, etc. Es por eso que se debe de trabajar en forma conjunta con profesionistas que manejen los aspectos anteriores.

Es por eso que el éxito del sistema en general depende de varios factores, tanto humanos, como de recursos en general. Pero hablando en el sentido del desarrollo del sistema, el éxito del sistema se debe a la calidad del análisis y diseño, ya que estos sientan las bases para que el desarrollo y la integración de los elementos sean por ende de una muy buena calidad, y proporcionen un gran beneficio como una herramienta más en el apoyo de la enseñanza.

Por último podemos decir que los objetivos que se fijaron en un principio y durante el desarrollo del sistema, se cumplieron y se realizaron satisfactoriamente, aplicando todos los pasos y la metodología que se explican en el presente trabajo.

Apéndice 1

Campo	Definición	Descripción
Nombre_inf	carácter(80)	Nombre con el que se identificara la información
Contenido	memo	Contenido o significado de la información
Clave_inf	numérico(2)	Clave del tipo de información que se almacena
Clave_app	numérico(2)	Clave del tipo de información a la que pertenece
Nombre_mult	carácter(50)	Nombre del elemento multimedia
País	carácter(100)	Localización del elemento multimedia
Clave_mult	numérico(2)	Clave del tipo de elemento multimedia
Clave_app	numérico(2)	Clave de aplicación a la que pertenece

Al Diccionario de datos para el proceso principal de presentación libre

Campo	Definición	Descripción
Clave_app	numérico(2)	Clave de la aplicación
Nombre_app	carácter(30)	Nombre de la aplicación
m1	carácter(30)	Nombre del módulo 1
m2	carácter(30)	Nombre del módulo 2
m3	carácter(30)	Nombre del módulo 3
m4	carácter(30)	Nombre del módulo 4
m5	carácter(30)	Nombre del módulo 5
m6	carácter(30)	Nombre del módulo 6

A2 Diccionario de datos para el proceso principal de selección de aplicación.

Campo	Definición	Descripción
Nombre_inf	carácter(80)	Nombre con el que se identificara la información
Contenido	memo	Contenido o significado de la información
Clave_inf	numérico(2)	Clave del tipo de información que se almacena
Clave_app	numérico(2)	Clave del tipo de información a la que pertenece
tipo_inf	carácter(50)	Identificación para el tipo de información

A3 Diccionario de datos para el proceso principal de Glosario

Campo	Definición	Descripción
Nombre_mult	carácter(50)	Nombre del elemento multimedia
Path	carácter(100)	Localización del elemento multimedia
Clave_mult	numérico(2)	Clave del tipo de elemento multimedia
Clave_app	numérico(2)	Clave de aplicación a la que pertenece
tipo_mult	carácter(50)	Identificación del tipo de multimedia

A4 Diccionario de datos para el proceso principal de Videos

Campo	Definición	Descripción
Nombre_mult	carácter(50)	Nombre del elemento multimedia
Path	carácter(100)	Localización del elemento multimedia
Clave_mult	numérica(2)	Clave del tipo de elemento multimedia
Clave_app	numérica(2)	Clave de aplicación a la que pertenece
tipo_mult	carácter(50)	Identificación del tipo de multimedia

A5 Diccionario de datos para el proceso principal de imágenes

Apéndice 2

Windows 95 desktop environment. Desktop icons include: Libro, Modulos, Imágenes, Videos, Acceso, Glosario, Salir, and Configuración. A window titled "Programa Multimedia para el apoyo de la materia Anatomía y Fisiología del Sistema Nervioso - Versión 1.0" is open, displaying the following text:

Programa de tesis realizado para obtener el título de Ing. en Computación

F1 Ayuda F2 Imágenes F3 Videos F4 Glosario F5 Salir del Sistema

MENUP - 1

```
Private Sub Command1_Click()
```

```
End Sub
```

```
Private Sub Command10_KeyUp(KeyCode As Integer, Shift As Integer)  
teclas KeyCode, 1  
End Sub
```

```
Private Sub command10_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
End Sub
```

```
Private Sub command2_KeyUp(KeyCode As Integer, Shift As Integer)  
teclas KeyCode, 1  
End Sub
```

```
Private Sub command2_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
End Sub
```

```
Private Sub Command3_KeyUp(KeyCode As Integer, Shift As Integer)  
teclas KeyCode, 1  
End Sub
```

```
Private Sub command3_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
End Sub
```

```
Private Sub Command4_KeyUp(KeyCode As Integer, Shift As Integer)  
teclas KeyCode, 1  
End Sub
```

```
Private Sub command4_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
End Sub
```

```
Private Sub Command5_Click()
```

```
End Sub
```

```
Private Sub Command5_KeyUp(KeyCode As Integer, Shift As Integer)  
teclas KeyCode, 1  
End Sub
```

```
Private Sub command5_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
End Sub  
Private Sub Command6_Click()
```

MENUP - 2

End Sub

```
Private Sub Command6_KeyUp(KeyCode As Integer, Shift As Integer)
    teclas KeyCode, 1
End Sub
```

```
Private Sub Command6_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
```

End Sub

```
Private Sub Command7_KeyUp(KeyCode As Integer, Shift As Integer)
    teclas KeyCode, 1
End Sub
```

```
Private Sub Command7_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
```

End Sub

```
Private Sub Command8_KeyUp(KeyCode As Integer, Shift As Integer)
    teclas KeyCode, 1
End Sub
```

```
Private Sub Command8_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
```

End Sub

```
Private Sub Command9_KeyUp(KeyCode As Integer, Shift As Integer)
    teclas KeyCode, 1
End Sub
```

```
Private Sub Command9_MouseMove(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
```

End Sub

```
Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
    'LLAMA A LA FUNCION PASANDO LOS VALORES DE LA TECLA PULSADA
    'EL 1 INDICA QUE FUE LLAMADA DESDE EL MENU PRINCIPAL
    teclas KeyCode, 1
End Sub
```

```
Private Sub Form_Load()
    App.HelpFile = App.Path & "\ayuda\SISNERV.hlp"
    Imagen1.Picture = LoadPicture(App.Path & "\imagenes\inicio\certras.bmp")
    REPEICIONES = 0
    '-----Corresponde al control que manejava el audio
    MENUP.MMControl2.Notify = False
    MENUP.MMControl2.Wait = True
    MENUP.MMControl2.Shareable = False
    MENUP.MMControl2.DeviceType = "WaveAudio"
```

MENUP - 3

```
MENUP.MMControl1.filename = App.Path & "\sonidos1.wav"  
MENUP.MMControl2.Command = "OPEN"  
unidad modulo = 1 'La unidad default es Anatomia  
End Sub
```

```
Private Sub Form MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)  
limpia etiquetas  
End Sub
```

```
Private Sub Image1 Click()  
If IZQUIERDO = 1 Then  
    IZQUIERDO = 0  
    teclas 10, 1  
End If  
End Sub
```

```
Private Sub Image1 MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)  
Label13.ForeColor = RGB(255, 0, 0)  
End Sub
```

```
Private Sub Image1 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)  
If Button = 1 Then IZQUIERDO = 1  
End Sub
```

```
Private Sub Image10 Click()  
If IZQUIERDO = 1 Then  
    IZQUIERDO = 0  
    Screen.MousePointer = 11  
    Load Menu  
    Load Terminos  
    Load IMAGENES  
    IMAGENES.Data1.Recordset.Index = "PrimaryKey"  
    Terminos.Data1.Recordset.Index = "primarykey"  
    Menu.Caption = "PRESENTACION LIBRE"  
    Menu.Show  
    Menu.RichTextBox1.Text = contenido termino("UNIDADES")  
    OPCION = 2  
    MENUP.Hide  
    Screen.MousePointer = 1  
End If  
End Sub
```

```
Private Sub Image10 MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)  
label6.ForeColor = RGB(255, 0, 0)  
End Sub
```

```
Private Sub Image10 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)  
If Button = 1 Then IZQUIERDO = 1  
End Sub
```

```
Private Sub Image11 MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

MENUP - 4

limpia etiquetas
End Sub

```
Private Sub Image12 Click()  
If IZQUIERDO = 1 Then  
IZQUIERDO = 0  
Screen.MousePointer = 11  
config.Show  
MENUP.Hide  
Screen.MousePointer = 1  
End If
```

End Sub

```
Private Sub Image12 MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)  
Label15.ForeColor = RGB(255, 0, 0)
```

End Sub

```
Private Sub Image12 MouseUp(Button As Integer, Shift As Integer, X As Single, Y  
As Single)  
If Button = 1 Then IZQUIERDO = 1
```

End Sub

```
Private Sub Image2 Click()  
If IZQUIERDO = 1 Then  
IZQUIERDO = 0  
teclas 1, 1  
End If  
End Sub
```

```
Private Sub Image2 MouseMove(Button As Integer, Shift As Integer, X As Single, Y  
As Single)  
Label14.ForeColor = RGB(255, 0, 0)  
End Sub
```

```
Private Sub Image2 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As  
Single)  
If Button = 1 Then IZQUIERDO = 1  
End Sub
```

```
Private Sub Image3 Click()  
If IZQUIERDO = 1 Then  
IZQUIERDO = 0  
MENUP.Enabled = False  
ACERCA.Show  
End If
```

End Sub

```
Private Sub Image3 MouseMove(Button As Integer, Shift As Integer, X As Single, Y  
As Single)  
Label12.ForeColor = RGB(255, 0, 0)  
End Sub
```

MENUP = 5

```
Private Sub Image3 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub
```

```
Private Sub Image4 Click()
If IZQUIERDO = 1 Then
```

```
End If
```

```
End Sub
```

```
Private Sub Image4 MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
'referencias uno 8
Label11.ForeColor = RGB(255, 0, 0)
End Sub
```

```
Private Sub Image4 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub
```

```
Private Sub Image5 Click()
If IZQUIERDO = 1 Then
IZQUIERDO = 0
Load Menu
```

```
Menu.Label1.Caption = "Actividades"
Menu.Show
OPCION = 3
MENUP.Hide
```

```
End If
```

```
End Sub
```

```
Private Sub Image5 MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
'referencias uno 6
Label9.ForeColor = RGB(255, 0, 0)
End Sub
```

```
Private Sub Image5 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub
```

```
Private Sub Image6 Click()
If IZQUIERDO = 1 Then
IZQUIERDO = 0
Load Menu
Menu.Caption = "EVALUACIONES"
'INICIO BOTONES
Menu.Show
OPCION = 4
MENUP.Hide
```

MENUP - 6

End If

End Sub

```
Private Sub Image6 MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
'referencias uno 7
label10.ForeColor = RGB(255, 0, 0)
```

End Sub

```
Private Sub Image6 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub
```

```
Private Sub Image7 Click()
If IZQUIERDO = 1 Then
    IZQUIERDO = 0
    teclas 3, 1
End If
```

End Sub

```
Private Sub Image7 MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
label7.ForeColor = RGB(255, 0, 0)
End Sub
```

```
Private Sub Image7 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub
```

```
Private Sub Image8 Click()
If IZQUIERDO = 1 Then
    IZQUIERDO = 0
    teclas 2, 1
End If
```

End Sub

```
Private Sub Image8 MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
label8.ForeColor = RGB(255, 0, 0)
End Sub
```

```
Private Sub Image8 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub
```

```
Private Sub Image9 Click()
If IZQUIERDO = 1 Then
    IZQUIERDO = 0
    Screen.MousePointer = 11
    Load Terminos
    Load IMAGENES
    IMAGENES.Data1.Recordset.Index = "primarykev"
```

MENUP - 7

```
Terminos.Data1.Recordset.Index = "primary/keu"  
Modulos.Show  
MENUP.Hide  
Screen.MousePointer = 1
```

```
End If  
End Sub
```

```
Private Sub Image9 MouseMove(Button As Integer, Shift As Integer, X As Single, Y  
As Single)  
'referencias uno 1  
Label5.ForeColor = RGB(255, 0, 0)  
End Sub
```

```
Private Sub Image9 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As  
Single)  
If Button = 1 Then IZQUIERDO = 1  
End Sub
```

```
Private Sub Label10 Click()  
If IZQUIERDO = 1 Then  
IZQUIERDO = 0  
Load Menu  
Menu.Caption = "EVALUACIONES"  
'INICIO BOTONES  
Menu.Show  
OPCION = 4  
MENUP.Hide
```

```
End If
```

```
End Sub
```

```
Private Sub Label10 MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)  
'referencias uno 7  
label10.ForeColor = RGB(255, 0, 0)
```

```
End Sub
```

```
Private Sub Label10 MouseUp(Button As Integer, Shift As Integer, X As Single, Y  
As Single)  
If Button = 1 Then IZQUIERDO = 1  
End Sub
```

```
Private Sub Label11 Click()  
If IZQUIERDO = 1 Then
```

```
End If
```

```
End Sub
```

```
Private Sub Label11 MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)  
'referencias uno 8  
label11.ForeColor = RGB(255, 0, 0)
```

```
End Sub
```

```
Private Sub Label11 MouseUp(Button As Integer, Shift As Integer, X As Single, Y  
As Single)
```

MENUP = 4

```
If Button = 1 Then IZQUIERDO = 1
End Sub
```

```
Private Sub Label12 Click()
If IZQUIERDO = 1 Then
    IZQUIERDO = 0
    MENUP.Enabled = False
    ACERCA.Show
End If
```

End Sub

```
Private Sub Label12 MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
Label12.ForeColor = RGB(255, 0, 0)
End Sub
```

```
Private Sub Label12 MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub
```

```
Private Sub Label13 Click()
If IZQUIERDO = 1 Then
    IZQUIERDO = 0
    teclas 10, 1
End If
```

End Sub

```
Private Sub Label13 MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
Label13.ForeColor = RGB(255, 0, 0)
End Sub
```

```
Private Sub Label13 MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub
```

```
Private Sub Label14 Click()
If IZQUIERDO = 1 Then
    IZQUIERDO = 0
    teclas 1, 1
End If
```

End Sub

```
Private Sub Label14 MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
Label14.ForeColor = RGB(255, 0, 0)
End Sub
```

```
Private Sub Label14 MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub
```

MENUP - 9

```
Private Sub Label15_Click()
IF IZQUIERDO = 1 Then
  IZQUIERDO = 0
  config.Show
  MENUP.Hide
  config.Data1.Recordset.Index = "primarykey"
  config.Data2.Recordset.Index = "primarykey"
  config.Data3.Recordset.Index = "primarykey"
  config.Data4.Recordset.Index = "primarykey"
  config.Data5.Recordset.Index = "nombre inf"
  config.Data6.Recordset.Index = "primarykey"

  Set DATA = DBEngine.Workspaces(0).OpenDatabase(App.Path & "\bases\informacio
n.mdb")
  SQLQ = "SELECT * FROM informacion WHERE [clave inf]=1 ORDER BY [nomb
re inf] "
  Set Dato2 = DATA.OpenRecordset(SQLQ, dbOpenDynaset)
  Set config.Data6.Recordset = Dato2

  Set DATA = DBEngine.Workspaces(0).OpenDatabase(App.Path & "\bases\informacio
n.mdb")
  SQLQ = "SELECT * FROM multimedia WHERE [clave mult]=1 ORDER BY [nomb
re mult] "
  Set Dato2 = DATA.OpenRecordset(SQLQ, dbOpenDynaset)
  Set config.Data7.Recordset = Dato2
End If
End Sub

Private Sub Label15_MouseMove(Button As Integer, Shift As Integer, X As Single,
Y As Single)
Label15.ForeColor = RGB(255, 0, 0)
End Sub

Private Sub Label15_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub

Private Sub Label5_Click()
If IZQUIERDO = 1 Then
  IZQUIERDO = 0
  Screen.MousePointer = 11
  Load Terminos
  Load IMAGENES
  IMAGENES.Data1.Recordset.Index = "primarykey"
  Terminos.Data1.Recordset.Index = "primarykey"
  Modulos.Show
  MENUP.Hide
  Screen.MousePointer = 1
End If
End Sub

Private Sub Label5_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
Label5.ForeColor = RGB(255, 0, 0)
End Sub
```

MENUP - 10

```
Private Sub Label5 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub
```

```
Private Sub Label6 Click()
If IZQUIERDO = 1 Then
    IZQUIERDO = 0
    Screen.MousePointer = 11
    Load Menu
    Load Terminos
    Load IMAGENES
    IMAGENES.Data1.Recordset.Index = "primarykey"
    Terminos.Data1.Recordset.Index = "primarykey"
    Menu.Caption = "PRESENTACION LIBRE"
    Menu.Show
    OPCION = 2
    MENUP.Hide
    Screen.MousePointer = 1
End If
```

End Sub

```
Private Sub Label6 MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Label6.ForeColor = RGB(255, 0, 0)
End Sub
```

```
Private Sub Label6 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub
```

```
Private Sub Label7 Click()
If IZQUIERDO = 1 Then
    IZQUIERDO = 0
    teclas 3, 1
End If
```

End Sub

```
Private Sub Label7 MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Label7.ForeColor = RGB(255, 0, 0)
End Sub
```

```
Private Sub Label7 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 1 Then IZQUIERDO = 1
End Sub
```

```
Private Sub Label8 Click()
If IZQUIERDO = 1 Then
    IZQUIERDO = 0
    teclas 2, 1
End If
```

MENUP - 11

End Sub

```
Private Sub Label8 MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
label8.ForeColor = RGB(255, 0, 0)
```

End Sub

```
Private Sub Label8 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
If Button = 1 Then IZQUIERDO = 1
```

End Sub

```
Private Sub Label9 Click()
```

```
If IZQUIERDO = 1 Then
```

```
IZQUIERDO = 0
```

```
Load Menu
```

```
Menu.Label11.Caption = "Actividades"
```

```
Menu.Snow
```

```
OPCION = 3
```

```
MENUP.Hide
```

```
End If
```

End Sub

```
Private Sub Label9 MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
'referencias uno 6
```

```
label9.ForeColor = RGB(255, 0, 0)
```

End Sub

```
Private Sub Label9 MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
If Button = 1 Then IZQUIERDO = 1
```

End Sub

```
Private Sub Picture1 KeyUp(KeyCode As Integer, Shift As Integer)
```

```
teclas KeyCode, 1
```

End Sub

```
Private Sub Timer1 Timer()
```

```
'CIERRA EL CONTROL MULTIMEDIA DE LA PANTALLA DE ENTRADA
```

```
'EJECUTA EL SONIDO QUE SE ESCUCHARA DURANTE EL USC DEL PROGRAMA
```

```
Bienvenida.MMControl1.Command = "close"
```

```
Unload Bienvenida
```

```
MENUP.MMControl2.Command = "play"
```

```
Timer1.Enabled = False
```

```
Timer2.Enabled = True
```

End Sub

```
Private Sub Timer2 Timer()
```

```
'ESTE ARCHIVO TIENE UNA DURACION DE 1.60 MINUTOS
```

```
'EL TIMER2 EJECUTA ESTE MODULO CADA 40 SEGUNDOS
```

```
'CUANDO REPETICIONES=4 VUELVE A REPETIR EL ARCHIVO
```

```
REPETICIONES = REPETICIONES + 1
```

```
If REPETICIONES = 4 Then
```

```
    MENUP.MMControl2.Command = "PREV"
```