

65
Zij



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

" SISTEMA INTERACTIVO DE ANATOMIA
VETERINARIA II PARA LA F.M.V.Z. "

T E S I S
QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A :
YAZMIN LOPEZ ESCOBAR



DIRECTOR DE TESIS:

ING. JUAN MANUEL MARTINEZ VILLALOBOS

MEXICO, D. F.

1996

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN



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.

Señor, gracias por permitirme llegar a esta etapa de mi vida, por estar siempre a mi lado y colmarme de bendiciones.

Queridos Padres, por su ejemplo, apoyo y amor con que me formaron y prepararon para ser alguien útil e independiente. Por su vida llena de esfuerzos y lucha, por darme siempre, todo lo que ha estado a su alcance. Gracias por su más valiosa herencia.

Los amo y admiro.

Juan, sin ti no sería posible todo ésto, por tu paciencia y aliento; gracias por apoyarme, por enseñarme, por ser mi guía y por lo más valioso que me diste, Tu amistad.

Con cariño.

Ernesto, por el impulso y amor que me haz dado, quiero agradecerte el que seas, el mejor motivo de mi vida. Pero agradezco a Dios de forma más profunda, tu existencia. Por tu cariño y comprensión.

Te amo.

Con cariño a mis *Hermanas*:

Arito, ya que no tengo la dicha de poder compartir contigo estos momentos, esperaré el día que podamos hacer una, nuestras alegrías. Te extraño.

Mireya y Liliana, si nos esforzamos podemos lograr lo que queramos, sigan adelante.

Gracias amigos, *Rocio, Dr. Jaime, Rodrigo, Nelly, Guillermo, Elida, José Luis, Dolores y demás amigos* que me han brindado su amistad y cariño. Cada uno en distintas etapas, forman una parte importante de mi vida.

A mi querida *Facultad de Ingeniería*, de la cual me siento muy orgullosa de formar parte, por darme una formación profesional y personal. A los *profesores* que la integran y lo hacen posible.

Y a la *Universidad Nacional Autónoma de México*, por la oportunidad tan importante que me ofreció al convertirme en universitaria y por las oportunidades que he tenido de representarla con gran orgullo y distinción.

Oración.

No me dejes pedir protección ante los peligros,
sino valor para afrontarlos.

No me dejes suplicar que se calme mi dolor, sino
que tenga ánimo para dominarlo.

No me dejes buscar aliados en el campo de
batalla de la vida, como no sea mi propia
fuerza.

No me dejes anhelar la salvación llena de miedo e
inquietud, sino desear la paciencia necesaria
para conquistar mi libertad.

Concédeme no ser cobarde y experimentar tu
misericordia sólo en mi éxito; pero déjame
sentir, que tu mano me sostiene en mi
fracaso.

Fuerza, serenidad y sabiduría.

Que Dios me dé la fuerza
para cambiar lo que pueda cambiar,
la serenidad para aceptar
todo aquello que no pueda cambiar,
y la sabiduría para distinguir la diferencia.

Gracias por todo. . .

Yazmin López Escobar.

Contenido

Introducción iii

Capítulo 1. Antecedentes

- 1.1 La computadora en la educación 1
 - 1.1.1 Creación de aplicaciones 2
 - 1.1.2 Introducción a multimedia 2
 - 1.1.2.1 Plataformas para multimedia 4
 - 1.1.2.2 La PC y la multimedia 4
 - 1.1.2.3 La tecnología multimedia y la educación 9
- 1.2 Introducción a la Anatomía 10
- 1.3 Creación de una aplicación de la computación orientada a la Anatomía Veterinaria 19

Capítulo 2. Análisis del sistema

- 2.1 Identificación de las necesidades del usuario 21
- 2.2 Objetivos del sistema 23
- 2.3 Especificación del sistema 24
 - 2.3.1 Alcances del sistema 24
- 2.4 El ámbito de la Información 27
 - 2.4.1 Modelización 27
 - 2.4.1.1 Diagramas de flujo de datos (DFD) 28
 - 2.4.1.2 Jerarquía más entrada proceso salida 29
- 2.5 Diagramas del "SIIV II" 30

Capítulo 3. Diseño del sistema

- 3.1 Técnicas del diseño 39
- 3.2 Diseño de una interfaz gráfica efectiva para el usuario 40
- 3.3 Diagramas del sistema 40
 - 3.3.1 Diagrama jerárquico o TVC del "SIIV II" 41
- 3.4 Definición de los conjuntos de información 49

Capítulo 4. Desarrollo del sistema

- 4.1 Selección del software 51
 - 4.1.1 El entorno de Windows 53
 - 4.1.2 Lenguaje de programación Visual Basic 56
 - 4.1.2.1 Programación orientada a eventos 56
 - 4.1.2.2 ¿Qué significa orientado a eventos? 56
 - 4.1.2.3 Visual Basic 57
 - 4.1.3 Acceso a bases de datos 69
 - 4.1.3.1 Dbase III Plus 70
- 4.2 Control de una aplicación 71
- 4.3 Codificación 71
- 4.4 Diagramas del "SIAV II" y su codificación 74

Capítulo 5. Pruebas del sistema

- 5.1 Técnicas de prueba del software 92
 - 5.1.1 Objetivos de la prueba 92
 - 5.1.2 Diseño de casos de prueba 93
- 5.2 Estrategias de prueba 96
 - 5.2.1 Verificación y validación 96
- 5.3 Pruebas del "SIAV II" 99
 - 5.3.1 Verificar y validar la unidad e integración del "SIAV II" 100
 - 5.3.2 Validación y verificación de los conjuntos de información empleados en el sistema "SIAV II" 101
 - 5.3.3 Validación del "SIAV II" como producto terminado 101

Capítulo 6. Conclusiones 110

- Apéndice A. Manual de usuario A-1**
 - Botones de comandos A-15

- Apéndice B. Glosario B-1**

- Bibliografía 115**

Introducción

La competencia que actualmente se vive en el mercado de trabajo y la cada vez más palpable aplicación de la computación a todas las actividades de la sociedad, hacen necesario que el estudiante de veterinaria, se familiarice con las tendencias que sigue el mundo del cómputo.

El presente trabajo introduce al lector al desarrollo de una herramienta orientada a la educación, un sistema educativo de Anatomía II, denominado "SIAV II".

El desarrollo del Sistema Interactivo para la materia de Anatomía II supone un importante apoyo para el departamento de Morfología de la Facultad de Medicina Veterinaria, para cualquiera que se encuentre involucrado con el estudio de la medicina veterinaria y en especial para el alumnado de dicha facultad.

Pero en caso de no encontrarse dentro de este grupo de personas, el usuario debe estar seguro de que el sistema desarrollado es un "interfaz gráfico de la computadora con el usuario", el cual permite que las aplicaciones sean fáciles de aprender y de usar, por lo que el manejo del sistema logra parte de su objetivo.

La metodología a seguir en la presente tesis es la *Ingeniería del Software*; la cual se define [FAI88] como "la disciplina tecnológica y administrativa dedicada a la producción sistemática de productos de programación, que son desarrollados y modificados a tiempo y dentro de un presupuesto definido".

El proceso de desarrollo de software en la *Ingeniería del Software* contiene las siguientes fases genéricas: análisis, diseño, implementación, prueba y mantenimiento para resolver una problemática existente.

El contenido de los capítulos se muestra a continuación:

Capítulo 1. Antecedentes.

En este capítulo se expone el importante papel que juega la computadora en la educación, presenta una visión general acerca de la historia de la Anatomía y finalmente se aplica la computación a la Anatomía.

Capítulo 2. Análisis del sistema.

En esta fase se plantea la problemática identificando las necesidades del usuario, se determinan los recursos con los que se cuenta, la forma como se maneja la información; de acuerdo a esto se realiza el análisis de objetivos para la creación de un sistema educativo, sus alcances y especificaciones. Se identifican las entradas de información disponibles y las salidas de las mismas, para posteriormente presentar un diagrama que muestra el flujo de la información.

Capítulo 3. Diseño del sistema.

El diseño del software es realmente un proceso multipaso que se enfoca sobre tres atributos distintos del programa: estructura de datos, arquitectura del software y detalle procedimental. El proceso de diseño traduce los requerimientos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación.

Se estudian las técnicas de programación, se determinan los módulos o procesos que componen el sistema y la forma como interactúan entre sí; y se elaboran los diagramas (tabla visual de contenidos) del sistema.

Capítulo 4. Implementación.

En esta sección del trabajo se establece el software que se utilizará. Tenemos un panorama del lenguaje de programación y la codificación del sistema.

Capítulo 5. Pruebas.

Esta fase se ocupa de que los componentes estructurales sean operacionales, aquí es donde todo el trabajo de diseño y desarrollo llega a un clímax.

Este capítulo expone las técnicas y estrategias de prueba de software, así como las pruebas realizadas al sistema para que sea operacional.

Capítulo 6. Conclusiones.

Este capítulo final presenta las conclusiones del trabajo.

Apéndice A, manual de usuario del sistema "SIIV II".

Apéndice B, glosario de términos empleados en este trabajo.

Capítulo 1

Antecedentes

1.1 La computadora en la educación

El uso de tecnología de vanguardia en el proceso educativo brinda beneficios significativos a estudiantes y maestros. Uno de los elementos más importantes dentro de este nuevo esquema educativo es la computadora. Algunas de las ventajas del uso de la computadora como apoyo a la educación son la capacidad de interacción (acción \Leftrightarrow reacción = interacción) que ésta ofrece a los educandos. La computadora es una poderosa herramienta para la enseñanza por su versatilidad.

Un elemento esencial de la tecnología del cómputo y tal vez la más importante, es la *interactividad*, lo cual significa que el usuario tiene el control, lo que ve y escucha, es el resultado de sus acciones y decisiones. Una experiencia interactiva muy familiar es la de conducir un automóvil; donde el conductor tiene el control sobre el vehículo. La clave está en que la interactividad requiere control. Las personas siempre se sienten más a gusto cuando tienen el control, y un ambiente de esta naturaleza las hace más receptivas a nueva información y a nuevas situaciones. El impacto potencial que tiene es mayor que el de la televisión y de las publicaciones juntas, debido a que la tecnología utilizada no sólo integra el poder audiovisual de la televisión con el de las publicaciones (texto, gráficos e imágenes fijas), sino que además agrega la capacidad de la interacción [LUT92].

El principio básico de la educación o capacitación apoyada en la computación, es que el estudiante capta y comprende los conceptos

presentados de forma más eficiente que cuando sólo lee o escucha acerca de ellos. El aprendizaje es aún más efectivo cuando el estudiante además interactúa con la información. En un ambiente así, aprende al ver, escuchar y hacer, trabajando por sí mismo y a su propia velocidad.

1.1.1 Creación de aplicaciones.

La capacitación interactiva por computadora no es nada nuevo ya que se ha utilizado ampliamente por más de una década bajo el nombre de *entrenamiento basado en computadora* (CBT Computer Based Training). De hecho, la *tecnología multimedia* nació de CBT. Los primeros sistemas de entrenamiento por computadora que eran capaces de reproducir audio y/o video requerían un gran número de equipos analógicos conectados a la computadora digital. Los sistemas multimedia de la actualidad son más integrados, compactos y baratos debido a que ahora el audio y el video se reproducen utilizando dispositivos digitales, y toda la información es almacenada, procesada y controlada por una computadora [LAR93].

1.1.2 Introducción a Multimedia.

Medio: factor a través del cual actúa una fuerza o se produce un efecto.

Multimedia: uso de varios medios de comunicación a la vez a fin de presentar información.

Multimedia se ha definido en el mundo de la computación como "*la integración, en una aplicación, de cuando menos tres de cinco tipos de datos susceptibles de ser procesados en una computadora*" [LAR93] (definición avalada por la Asociación Mexicana de Multimedia y Nuevas Tecnologías, y por el Laboratorio de Inteligencia Artificial y Multimedia del Instituto de Ingeniería de la UNAM). Estos tipos de datos son:

- audio,
- video,
- animación,

- imagen fija y
- texto.
- Hipertexto, actualmente se considera a este como un medio de despliegue de información que se integra al concepto multimedia.

El concepto de multimedia implica el uso de hardware adicional al estándar de cómputo (principalmente en IBM PC's) como: tarjetas digitalizadoras y reproductoras de audio, CD-ROM y bocinas, así como el software necesario para hacer uso de estos dispositivos, principalmente. Sin embargo, la creación de aplicaciones denominadas multimedia, no es producto de la adquisición de este hardware y software, sino de la integración de diversas disciplinas apoyadas en la manipulación de las herramientas de cómputo en diferentes niveles [WOD92].

Una clase especial de capacitación en la que multimedia tiene importantes ventajas, es la ayuda en línea o tutoriales, que se pueden encontrar en la mayoría de las aplicaciones de la actualidad. Estos sistemas están basados en texto casi en un 100%, no obstante, el uso del *hipertexto* se está haciendo cada vez más común. En un sistema de hipertexto, el usuario puede señalar una palabra o una frase y saltar inmediatamente a información más detallada acerca del tema seleccionado. Sin embargo, los sistemas de ayuda se vuelven más agradables y versátiles cuando se usa información en forma de audio, video y/o animaciones, de la misma forma que se usa el hipertexto, esto recibe el nombre de *hipermedia*.

Aunque las bases de datos de los sistemas de ayuda en línea pueden contener toda la información que el usuario necesita, es difícil presentarla de forma amigable utilizando sólo texto, especialmente a usuarios que no están familiarizados con el uso de computadoras. Con multimedia, el texto puede ser reemplazado por voz en su mayor parte y, donde sea de utilidad, la presentación también puede incluir audio (distinto a la voz) y video o animaciones para explicar conceptos confusos. El resultado es que el usuario percibe que la aplicación entera es más amigable, más fácil de entender y de operar al reducirse la carga cognitiva (trabajo necesario para asimilar un tema). Por ejemplo, el Laboratorio de Multimedia del Instituto de Ingeniería de la UNAM en colaboración con el Departamento de Fisiología de la

Facultad de Medicina de la misma institución, desarrolló un sistema educativo multimedia sobre el ciclo cardiaco humano, los estudiantes que usaron este sistema requirieron entre 25% y 30% menos tiempo para aprender el tema que los estudiantes que utilizaron métodos tradicionales [LAR93].

1.1.2.1 Plataformas para multimedia.

Los sistemas multimedia pueden ser presentados prácticamente sobre cualquier plataforma de cómputo. Estas plataformas pueden clasificarse como: IBM PC's y compatibles, Apple Macintosh, Commodore Amiga, estaciones de trabajo (entre las que se incluye Silicon Graphics) y mainframes [WOD92].

La computadora a utilizar depende del tipo de aplicaciones que se deseen crear. Las PC's, a pesar de las capacidades que se les pueden agregar por medio de tarjetas de audio y de procesamiento de video, son los equipos más limitados, mientras que las Macintosh son una plataforma ideal para hacer multimedia. La ventaja de las PC's es, que representan la mayor capacidad instalada en cuanto a equipo de cómputo se refiere.

1.1.2.2 La PC y la multimedia.

La tecnología multimedia demanda gran poder de cómputo. Esta demanda surge principalmente de las necesidades de presentación de video. Las aplicaciones multimedia exigen mucho trabajo al sistema de video de la computadora al desplegar rápidamente imágenes complejas. Esto significa que la computadora debe ser capaz de recuperar rápidamente información de video del almacenamiento masivo, procesarla y pasarla al subsistema de video para su presentación. El video de alta calidad (alta resolución y mayor cantidad de colores) hace esta demanda aún mayor. Los archivos de video digital son tan grandes que son prácticamente imposibles de manejar sin usar técnicas de compresión [MIR92]. La compresión de video requiere procesamiento de la computadora para eliminar redundancias en las imágenes (ver apéndice B).

La resolución VGA (Video Graphics Array) estándar tiene 640x480 píxeles lo que da un total de 307,200 píxeles en una imagen que ocupe toda la pantalla. Para convertir este número en una cantidad de datos, es necesario tomar en cuenta el número de colores que pueden ser desplegados por cada píxel. En un sistema VGA de 16 colores se necesitan 4 bits por cada píxel o un byte por cada dos píxeles. Por lo tanto, una imagen que abarque toda la pantalla en 16 colores requiere 153,600 bytes de datos ($640 \times 480 / 2$). La calidad de las imágenes que se pueden presentar con un sistema VGA de 16 colores no es suficiente para desplegar aplicaciones multimedia realizadas en resoluciones y cantidad de colores superiores.

Las imágenes de video requieren calidad fotográfica para lo cual es necesario manejar más de 16 colores. Existen sistemas que pueden desplegar hasta 16,777,216 colores usando 24 bits (3 bytes) de datos por cada píxel ($640 \times 480 \times 3 = 921,600$ bytes necesarios para guardar una imagen que ocupe toda la pantalla).

Para reproducir video usando imágenes de computadora, es necesario desplegar imágenes a una velocidad de cuando menos 15 cuadros por segundo. Para el sistema VGA de 16 colores, 15 imágenes por segundo significan 4.6 Mb de datos por segundo, para el sistema VGA de 16.7 millones de colores significan 13.824 Mb de datos por segundo. Estos números están fuera de la capacidad de las computadoras actuales. Por lo cual para tener video es necesario hacer uso de varios trucos que han sido desarrollados para reducir los requerimientos de datos de video [MIR92].

El microprocesador.

La arquitectura de todas las PC's está basada en microprocesadores de la familia Intel 80x86. El corazón del sistema es el microprocesador que está conectado a los buses del sistema (líneas de conexión de todos los componentes de la computadora), al que se conectan todos los otros dispositivos que conforman la PC. Como el bus es en paralelo, cualquier dispositivo puede comunicarse con los demás siempre bajo el control del microprocesador. Normalmente el bus se localiza físicamente en la tarjeta principal (mother board).

El microprocesador determina la velocidad y las capacidades de la PC. Además de la frecuencia a la que trabaja el microprocesador, existe otra forma de medir su velocidad que es en *millones de instrucciones por segundo* (MIPS). Una aplicación multimedia se beneficia con un microprocesador muy rápido. Los microprocesadores 80386 y 80486 tienen dos variantes: "SX" y "DX". Las versiones SX manejan buses de 16 bits, mientras que las versiones DX manejan buses de 32 bits [WOD92]. El software multimedia trabaja bien en arquitecturas basadas en microprocesadores "SX", pero es preferible usar versiones DX para mejores resultados, aunque no es el único requisito para mejorar el desempeño de las aplicaciones multimedia.

La RAM.

Otra parte importante de la computadora es la RAM (Random Access Memory). Permite el almacenamiento temporal de los datos y programas que están siendo usados por el microprocesador. Mientras más memoria RAM tenga un sistema, funcionará mejor. La tecnología multimedia requiere de 16 Mb de RAM para tener buenos resultados en una aplicación.

El sistema de video.

El sistema de video en todas las PC's usa un mecanismo de memoria mapeada, en el cual una región de memoria guarda los valores de todos los píxeles a ser desplegados en la pantalla. Para crear una imagen, el adaptador de video lee esta sección de memoria a una gran velocidad y convierte los valores de los píxeles en señales que controlan el dispositivo de video (generalmente un monitor). Este proceso requiere una gran cantidad de actividad en memoria para leer los datos y refrescar la pantalla. La frecuencia típica a la que trabaja este mecanismo es de 60 ciclos por segundo.

Como consecuencia de lo anterior, un sistema VGA de 16 colores accesa 9.22 millones de bytes por segundo. El bus del sistema no podría transmitir a esta velocidad aún cuando fuese lo único que hiciera. Para resolver este problema, los adaptadores de video

incluyen su propia memoria para almacenar los datos que se ven en pantalla. El bus del sistema es utilizado para actualizar la información contenida en esta memoria.

Para mejorar el desempeño de las aplicaciones multimedia es recomendable el uso de aceleradores gráficos.

El sistema de audio.

La capacidad de una PC para producir sonido consiste de una pequeña bocina controlada por una salida de 1 bit de la computadora. Al enviar secuencias de 1's y 0's, variando la frecuencia, se logra una gran cantidad de sonidos. Sin embargo, estos sonidos son artificiales y no son aplicables a ningún sonido cercano a lo real y mucho menos a audio de calidad CD. La misma situación se presenta en lo referente al video.

Para mejorar la calidad del sonido que una PC puede reproducir, es necesario agregarle un tarjeta de audio.

El almacenamiento masivo y el CD-ROM.

La velocidad de los dispositivos de almacenamiento masivo está determinada por el factor al cual pueden leer del medio de almacenamiento. Esta operación también requiere de gran cantidad de trabajo por parte del microprocesador. Las aplicaciones multimedia requieren grandes cantidades de espacio en disco duro debido a la información de audio, video, animaciones e imágenes fijas que manejan. Multimedia requiere que éstos dispositivos además de tener gran capacidad para contener información, sean lo más rápidos posible.

Los sistemas de memoria de sólo lectura basados en CD son importantes medios de almacenamiento y distribución de aplicaciones multimedia. La tecnología CD-ROM permite almacenar 680 Mb (en el PC Semanal del 9 de abril de 1994 se publicó un artículo en el que se dice que en investigaciones recientes se ha logrado almacenar de 1.5 a

2 Gb de información y que se espera llegar hasta los 5 Gb) [PC94] en sólo un disco compacto de 12 cm. Esta tecnología se basa en la de los reproductores de discos de audio y por lo tanto no es costosa.

El software multimedia.

Este software debe realizar las tareas de forma que presente imágenes aceptables, audio y video de calidad. El audio y el video deben ser reproducidos sin retardos ni distorsiones. El software debe responder de forma efectiva a la interacción con el usuario y debe ser lo suficientemente flexible para permitir la realización de las diferentes variantes de presentación que se requieren para cubrir un amplio rango de aplicaciones. Algunas de las demandas de este tipo de software exceden las capacidades de la PC más poderosa, por lo que se hace necesario el uso de hardware extra que auxilie a la arquitectura básica de la PC. Esto se cumple especialmente en el caso de reproducción de audio y video. Además del sistema operativo, el software requerido para hacer multimedia [LUT92] se clasifica en:

- ❑ **Software para creación.** Dentro de esta categoría se encuentran todos aquellos programas cuyo objetivo es permitir al autor crear materiales para aplicaciones multimedia. Como ejemplo se tienen los programas de diseño gráfico y dibujo como Paint Brush, Corel Draw, Aldus Photo Styler; así como los que sirven para crear animaciones: 3D Studio y Animator entre otros.
- ❑ **Software para edición.** Esta categoría incluye aquellos programas que permiten al autor editar y /o modificar los elementos creados previamente con otras herramientas. Como ejemplo se tienen los editores de imágenes como Photo Finixh que vienen con los dispositivos scanner. Después de capturar una imagen con el scanner, se utiliza el programa editor para retocarla. Otro ejemplo de este tipo de software es el que viene con las tarjetas de audio que además de permitir grabar sonidos en la computadora, permite editarlos y agregarles efectos tales como eco o filtros.
- ❑ **Software para integración.** Una vez que se han creado todos los elementos es necesario integrarlos en una aplicación. La

integración se puede hacer utilizando lenguajes de programación de tercera generación como Pascal o C, o bien lenguajes de cuarta generación o sistemas de autoraje.

1.1.2.3 La tecnología multimedia y la educación.

La forma en la que aprende el ser humano está basada en los sentidos, por ello es importante presentar información y desarrollar conceptos utilizando técnicas que permitan el aprendizaje visual, auditivo y táctico. Por otro lado, es importante resaltar el hecho de que, para que un estudiante comprenda una idea nueva, debe primero relacionarla con sus propias experiencias. La compañía IBM ha realizado estudios a fin de determinar la cantidad de información que puede retener el ser humano a través de sus sentidos (Información Científica y Tecnológica, Tecnología y educación, octubre de 1993). Dichos estudios muestran los siguientes resultados: se retiene a corto plazo alrededor del 10% de lo que se ve, 20% de lo que se escucha, 40% de lo que se ve y se escucha, y el 75% de lo que se ve, se escucha y se lleva a la práctica.

Por otro lado, se ha desarrollado una serie de investigaciones [CON93] a fin de cuantificar la efectividad de la tecnología multimedia en el ámbito educativo, obteniendo el siguiente resultado: los estudiantes aprenden en 50% menos tiempo y retienen 25% más información.

De lo anterior se puede concluir que el proceso de aprendizaje se beneficia si la información que se desea consultar se presenta en forma tal que impacte los sentidos y exista a su vez posibilidad de interactuar con ella.

1.2 Introducción a la Anatomía

Anatomía es la rama de la ciencia biológica que trata de la forma y estructura de los organismos. Se halla en íntima correlación con la fisiología, que trata de las funciones del cuerpo [FRA88].

Etimológicamente la palabra "anatomía" significa cortar separando o disociando las partes del cuerpo. En las primitivas fases de su desarrollo la Anatomía fue una ciencia puramente descriptiva, basada sólo en las observaciones que eran posibles a simple vista y con la única ayuda de los instrumentos necesarios para la disección, el escalpelo, las pinzas, etc. En aquella época, por lo tanto, el término empleado expresaba la naturaleza de la materia. Pero, como la extensión de la ciencia se ha ido simplificando y han aumentado los conocimientos anatómicos, se ha hecho necesario establecer subdivisiones e introducir nuevos términos para designar campos y métodos especiales. La introducción del microscopio y sus accesorios permitió el estudio de los más finos detalles de estructura y el conocimiento de organismos diminutos hasta entonces desconocidos, y este campo de investigación se ha desarrollado rápidamente, constituyendo la ciencia de la *anatomía microscópica* o histología, separada convencionalmente de la *anatomía macroscópica*. Al propio tiempo, el estudio de los cambios que el organismo experimenta durante su desarrollo alcanzó pronto importancia suficiente para ser considerado desde el punto de vista práctico como una rama separada, conocida con el nombre de *embriología*. Este término se aplica sólo corrientemente a las primeras fases de desarrollo durante las cuales se forman los tejidos y órganos. El término *ontogenia* se usa para designar el desarrollo completo de lo individual. La filogenia o historia ancestral de las especies está constituida por los cambios evolutivos que han experimentado, según revelan los documentos geológicos.

La *Anatomía veterinaria* es la rama que trata de la forma y estructura de los principales animales domésticos. Su estudio se hace generalmente con miras a las necesidades profesionales, y por lo mismo su carácter es ampliamente descriptivo.

La ciencia de la anatomía se ha extendido tanto que hoy se divide en varias ramas especializadas, hasta 30 subdivisiones, de las cuales estamos particularmente interesados en la *anatomía descriptiva* o *anatomía macroscópica*; es decir, el estudio de las formas y relaciones (posiciones relativas) de las estructuras del cuerpo, visibles sin ayuda de medios ópticos.

La *anatomía comparada* es el estudio de las estructuras de varias especies de animales, con particular atención a los caracteres que ayudan a su clasificación.

La *embriología* es el estudio de la evolución anatómica que abarca desde la concepción (fecundación del óvulo en el interior del cuerpo materno) hasta el nacimiento.

El perfeccionamiento más reciente de los estudios anatómicos es la *citología ultraestructural*, la cual trata de las porciones de las células según se observan al microscopio electrónico. El término "estructura fina" se emplea algunas veces para designar los detalles puestos de relieve en microfotografías electrónicas.

Se emplean dos métodos de estudio especiales, el *sistemático* y el *topográfico*. En el primero se considera el cuerpo como formando sistemas de órganos o aparatos que son similares por su origen y estructura y están asociados en la realización de ciertas funciones. El término *anatomía topográfica* designa los métodos con que se determinan exactamente las posiciones relativas de varias partes del cuerpo. Presupone un conocimiento perfecto de la anatomía sistemática.

Nuestro método de estudio de la anatomía procederá principalmente por sistemas y aparatos, es decir, sistemático. Para designar la ciencia, se añade el morfema "logía", que significa estudio o tratado, a la raíz verbal del sistema en cuestión. En el cuadro 1-1 se señalan los sistemas y aparatos comúnmente aceptados, el nombre de la ciencia que los estudia y sus principales estructuras.

Cuadro 1-1 Nomenclatura de la anatomía sistemática

Sistema o aparato	Rama de conocimiento	Principal tejido de que se ocupa
<i>Sistema esquelético</i>	<i>Osteología</i>	<i>Huesos</i>
<i>Sistema articular</i>	<i>Artrología</i>	<i>Articulaciones</i>
<i>Sistema muscular</i>	<i>Miología</i>	<i>Músculos</i>
<i>Aparato digestivo</i>	<i>Esiacnología</i>	<i>Estómago e intestinos</i>
<i>Aparato respiratorio</i>	"	<i>Pulmones y vías aéreas</i>
<i>Aparato urinario</i>	"	<i>Riñones y vejiga</i>
<i>Aparato reproductor</i>	"	<i>Ovarios y testículos</i>
<i>Aparato endocrino</i>	<i>Endocrinología</i>	<i>Glándulas de secreción interna</i>
<i>Sistema nervioso</i>	<i>Neurología</i>	<i>Encéfalo, médula y nervios</i>
<i>Aparato circulatorio</i>	<i>Angiología</i>	<i>Corazón y vasos</i>
<i>Sistema cutáneo</i>	<i>Dermatología</i>	<i>Piel</i>
<i>Aparato sensitivo</i>	<i>Estesiología</i>	<i>Ojos y oídos</i>

En la nomenclatura geográfica se emplean ciertos planos y puntos arbitrarios de referencia conocidos como meridianos y paralelos de longitud y latitud. Sin embargo, como un animal normalmente no puede orientarse exactamente con una línea de la superficie terrestre, nuestros marcos de referencia deben establecerse respecto al animal mismo, y han de aplicarse cualquiera que sea su posición (fig. 1-1 y 1-2). Para cumplir estas exigencias nos valemos de planos y direcciones escogidos arbitrariamente para situar los diferentes órganos [FRA88].

Craneal y *anterior* son términos de dirección o sentido, que significan "hacia la cabeza". Así, decimos que el tórax está en posición craneal con respecto al abdomen, pues el primero está más cerca de la cabeza que el segundo.

Por oposición, *caudal* o *posterior* significan en dirección de la cola. El anca es caudal con respecto al lomo.

El plano *medio* o *anteroposterior* es el plano imaginario que corta el cuerpo de la cabeza a la cola, y lo divide en dos mitades iguales,

derecha e izquierda. Un buey sacrificado se divide en los rastros por el plano medio.

Se llama *plano sagital* todo plano paralelo al medio; éste se llama a veces *mediosagital*.

El plano *transverso* o *transversal* se dispone en ángulo recto con el anterior, por lo que divide al cuerpo en dos segmentos, craneal y caudal. Toda sección de un organismo suele hacerse en sentido paralelo a este plano. La cincha llevada por un animal define este plano, el cual corta a una vaca por el abdomen.

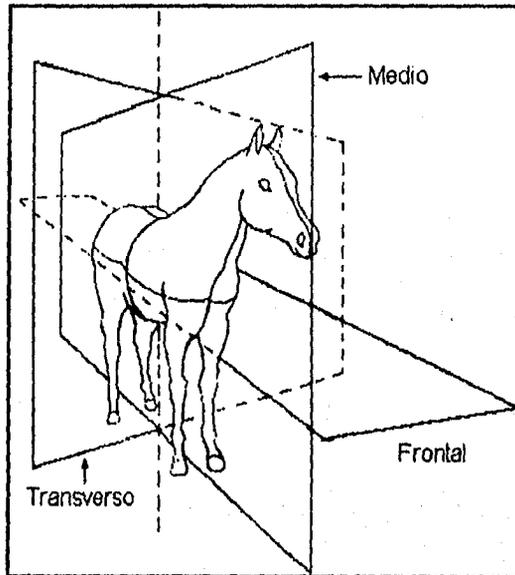


Fig. 1-1 Planos imaginarios de referencia.

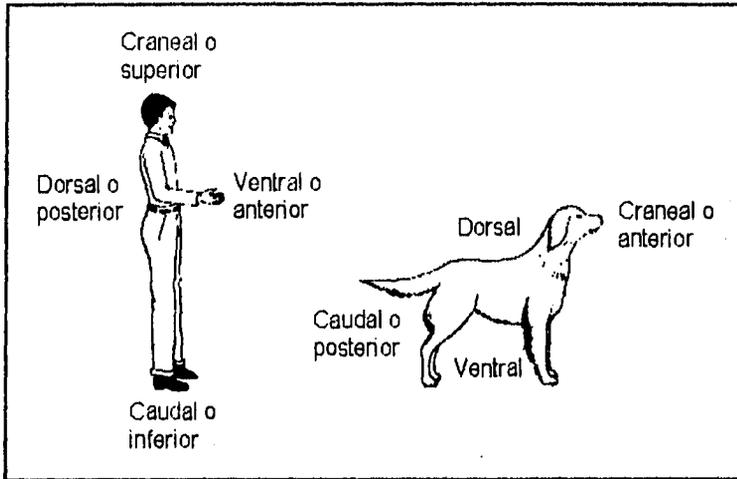


Fig. 1-2 Terminología comparada entre hombre y animal.

El plano *frontal* está también en ángulo recto con los dos anteriores, el medio y el transversal; divide el cuerpo en las porciones dorsal o superior y ventral o inferior. Si una vaca se mete en una laguna hasta que el agua le cubre el costillar, la superficie líquida equivale al plano frontal.

Además de los planos de referencia, hay otros términos valiosos para precisar la zona de estudio.

Medial es un adjetivo que significa próximo o hacia el plano medio, como el corazón, que es medial con respecto a los pulmones, pues está más cercano que ellos al plano medio. La excrecencia córnea de las patas está en situación medial de la pata del caballo, o sea que está dentro, más cerca de la línea media.

Lateral es término antónimo de medial; significa alejado del plano medio. Las costillas son laterales con respecto al pulmón; están fuera, alejadas del plano medio.

Dorsal es un término de dirección que significa cerca del espinazo o columna vertebral, en dirección a ella o más allá. Los riñones son dorsales con respecto a los intestinos, pues están más cerca que estos últimos del espinazo. *Dorso* es el nombre que se refiere a la porción dorsal del cuerpo. La silla se coloca en el dorso del caballo.

Ventral, por oposición, significa alejado de la columna vertebral y en dirección o próximo a la parte inferior de la pared abdominal. La ubre es el órgano más ventral del cuerpo de la vaca.

Profundo o *interno* significan próximo al centro de gravedad del animal o al centro de una extremidad. El húmero (hueso del brazo) es el más profundo de los órganos en la extremidad anterior.

Superficial y *externo* hacen referencia a la proximidad de la piel o a la superficie de un órgano o extremidad. El pelo es superficial con respecto a cualquier otra estructura del organismo.

Proximal significa relativamente cerca de una parte, que suele ser la columna vertebral, el cuerpo, el centro de gravedad o un punto de referencia, a veces una extremidad o miembro. La rodilla es proximal con respecto al casco.

Distal significa relativamente lejano de la columna vertebral. El casco es distal respecto al carpo o rodilla.

Palmar (volar) se refiere a la superficie de flexión o caudal de la extremidad anterior, distal respecto de la articulación del hombro. El término *dorsal*, cuando se aplica a una extremidad, se refiere al lado opuesto (craneal).

Plantar hace referencia a la superficie caudal de las extremidades posteriores, por debajo de la corva; se refiere al lado directamente opuesto (el lado craneal).

Prono se refiere a la posición en la cual la superficie dorsal (dorso) de todo el cuerpo o de una extremidad está en la parte superior. La *pronación* es el acto por el cual un órgano toma la posición prona.

Supino se refiere a la posición en que la superficie ventral del cuerpo, o las superficies volar o plantar de una extremidad están en la parte superior. *Supinación* es la acción de llevar una superficie a posición supina.

Todo ser vivo, animal o vegetal, está formado por unidades diminutas llamadas *células*. Los animales más sencillos, como las amibas, constan de una sola célula, la cual puede cumplir todas las funciones vitales como crecimiento (aumento de tamaño), metabolismo (utilización de los alimentos), reacción a los estímulos (como moverse hacia la luz), contracción (reducir el volumen en una dirección) y reproducir (formación de nuevos individuos de la misma especie).

Una célula típica está formada por tres partes principales: *citoplasma*, *núcleo* y *membrana*).

Cuando el número de células aumenta en un cuerpo vivo, algunas de ellas se especializan en una o más funciones peculiares; estas células diferentes se agrupan y forman *tejidos*.

Diversos tejidos se agrupan en conjuntos funcionales llamados *órganos*. Un grupo de órganos con función común forman un *aparato*.

Los principales tipos de tejido son: 1) *epitelial*, que cubre la superficie del cuerpo, tapiza las cavidades y se encuentra también en la estructura activa de las glándulas; 2) *conectivo*, que sostiene y une entre sí otros tejidos, y del que, en el caso de la médula ósea, se derivan los elementos figurados de la sangre; 3) *muscular*, especializado en la contracción, y 4) *nervioso*, conductor de impulsos de una parte a otra del organismo.

Todos los animales de granja son vertebrados, y como tales tienen columna vertebral. El organismo (con escasas excepciones) presenta simetría bilateral, lo que significa que las mitades derecha e izquierda son casi idénticas. Los órganos de cada lado se llaman pares, del mismo modo que hablamos de un par de guantes, similares pero no intercambiables. Casi todos los órganos impares están situados en la línea media o cerca de ella y, como se comprende, hay uno en cada cuerpo animal: lengua, tráquea, columna vertebral y corazón. Las

costillas, ojos, extremidades y muchos músculos son ejemplos de órganos pares.

El corte medial de un vertebrado muestra dos cavidades: una *dorsal*, que aloja el encéfalo y la médula, y una *ventral*, donde están contenidas las vísceras (órganos blandos). Esta última cavidad está dividida por el diafragma en cavidad torácica en sentido craneal, y cavidad abdominopélvica, caudalmente (que incluye las cavidades abdominal y pélvica).

La *cavidad torácica* contiene el *saco pericardiaco*, que envuelve el corazón, y dos *sacos pleurales*, que rodean a ambos pulmones. Estos sacos formados por *membranas serosas*.

En la *cavidad abdominal* están contenidos los riñones, casi todos los órganos digestivos y algunos órganos reproductores internos en ambos sexos. En la *cavidad pélvica* está alojada la porción terminal del aparato digestivo (el recto) y las porciones internas del aparato urogenital que no pertenecen a la cavidad abdominal; la membrana serosa que envuelve las vísceras abdominales, y también parte de las pélvicas, se llama *peritoneo*.

Las células epiteliales glandulares se especializan en la secreción o la excreción. *Secreción* es la liberación por parte de la célula glandular de una sustancia sintetizada por la célula, la cual normalmente afecta a otras células en otras partes del cuerpo. La *excreción* es la expulsión de sustancias de desecho inútiles para el organismo animal.

Las glándulas pueden clasificarse como endocrinas (sin conducto excretor, que vierten sus productos directamente en la corriente sanguínea) y exocrinas (que vierten sus productos por un conducto que desemboca en una superficie epitelial).

Las *glándulas endocrinas* son parte importante de los mecanismos de regulación del organismo, pues producen compuestos químicos especiales que llevan el nombre de *hormonas*. Estas se difunden por todo el organismo y constituyen el *control humoral* del cuerpo. Este control y el de los nervios son los dos mecanismos que mantienen la homeocinesia, llamada también, *homeostasia* (estado relativamente

estable pero cambiante del cuerpo). La respuesta humoral a los estímulos del medio (externo o interno) es mucho más lenta y prolongada que la respuesta a la excitación del sistema nervioso.

Según la clasificación morfológica, una glándula es simple si su conducto no se divide; compuesta si el conducto se ramifica. Si la porción activa forma una estructura parecida a un tubo, se llama *tubular*; si dicha porción se asemeja a un racimo o cavidad, se llama *alveolar* o *acinar*. La combinación de estructuras secretoras tubulares y alveolares dará lugar a una glándula tubulo-alveolar [FRA88].

1.3 Creación de una aplicación de la computación orientada a la Anatomía Veterinaria

Algunos de los puntos que debe cubrir una aplicación orientada a la enseñanza de la Anatomía veterinaria son:

- lograr la atención del estudiante
- presentar el tema
- realizar una sesión de preguntas y respuestas
- evaluar lo aprendido por el usuario

Lograr la atención del estudiante.

Además de lograr la atención del estudiante es necesario hacerlo receptivo a nueva información. Esto normalmente se logra exponiendo un poco de nuevo material, de preferencia una parte de las más interesantes, para tratar de despertar el espíritu inquisitivo del estudiante y hacerlo desear aprender más.

Presentar el tema.

Los métodos tradicionales de enseñanza en el salón de clases involucran ya sea la presentación donde el profesor expone el tema o bien, el libro de texto de donde el estudiante lee la información. La mayoría de los cursos usan una combinación de ambos métodos. Sus equivalentes son las presentaciones no interactivas (para las exposiciones en clase) y las presentaciones interactivas (para la lectura de libros de texto). Debido a que la interactividad es mucho mejor que la de un libro de texto, este es el método preferido.

Un buen método para presentar el tema de estudio es mostrar al estudiante algo que inicie el camino hacia la información, y dejarle explorar el resto de ese camino por sí mismo. Permitirle al usuario estudiar por medio de dibujos, imágenes, o video es mejor que usando sólo texto, y es aún mejor cuando el tema de estudio puede ser

simulado por la computadora y el estudiante puede trabajar directamente con él. El propósito de esta sección en cualquier aplicación educativa es exponer el tema al estudiante de una forma que le permita entender su estructura y detalles.

Sesión de preguntas y respuestas.

En el salón de clases, normalmente, el profesor conduce una sesión libre de preguntas y respuestas o de discusión. En el ambiente de la computación, una discusión libre es imposible, por lo que la computadora necesita tener algún otro método de saber que el estudiante tiene una pregunta. Parte de la solución a este problema puede ser una sesión de ejercicios en la que el estudiante repase lo más importante de cada módulo estudiado. El estudiante es libre de repetir esta sección cuantas veces quiera. Normalmente, una aplicación educativa debe ofrecer explicaciones alternativas para cada concepto, que puedan ser accesibles por el estudiante cuando tenga un problema.

Evaluación.

Los métodos de evaluación del aprendizaje logrado por el estudiante han sido desarrollados desde que surgió el entrenamiento basado en computadora (CBT). La evaluación típica del entrenamiento basado en computadora es del tipo de elección múltiple, debido a que es muy difícil interpretar respuestas en forma de texto libre. Excepto por el hecho de poder presentar imágenes, animaciones, audio y/o video para plantear preguntas, multimedia no agrega nada nuevo o especial al problema de la evaluación basada en computadora.

Capítulo 2

Análisis del sistema

El análisis del sistema define el papel de cada elemento de un sistema informático, asignando finalmente al software el papel que va a desempeñar. La fase de definición se centra sobre el *qué*. Esto es durante la definición, el que desarrolla el software intenta identificar qué información ha de ser procesada, qué función y rendimiento se desea, qué interfaces han de establecerse, qué restricciones de diseño existen y qué criterios de validación se necesitan para definir un sistema correcto. Por tanto, han de identificarse los requisitos clave del sistema y del software. Primero se plantean los objetivos del sistema, pues una vez que se han definido, se procederá a identificar las entradas, salidas y archivos del mismo; así como el flujo de los datos, los diferentes procesos de transformación y los principales elementos de datos que se van a manejar en el sistema.

Esta fase y la de diseño, se distinguen por la alta interactividad que existe con los usuarios, por lo que es indispensable utilizar algunas técnicas que ayuden a lograr los objetivos deseados.

2.1 Identificación de las necesidades del usuario

El Departamento de Morfología de la Facultad de Medicina Veterinaria y Zootecnia, se ha preocupado por proporcionar a profesores y estudiantes, un medio que facilite el estudio de las materias que pertenecen a dicho departamento, tomando como piloto para la

implantación de un *sistema de cómputo didáctico*, la materia de *Anatomía II*, al cual denominaremos "**SIIV II**" (Sistema Interactivo de Anatomía Veterinaria II). Con la ambición de que dicho sistema, permita la creación de un laboratorio donde tanto alumnos como profesores, encuentren un apoyo para el estudio y la labor académica.

Aspectos generales del Departamento de Morfología.

El Departamento de Morfología consta de 2 áreas:

- *Histología*, donde se ubican las materias de Histología y Biología Celular que se imparten en el primer semestre.
- *Anatomía*, formada por las materias de Anatomía I y Anatomía II que se imparten en el primer y segundo semestre respectivamente.

En el curso de Anatomía I se estudia la anatomía de los mamíferos domésticos desde un punto de vista descriptivo, es decir, se describen macroscópicamente cada una de las estructuras que integran los diferentes sistemas o aparatos de los animales, para esto los profesores se apoyan en esquemas, diapositivas, modelos tridimensionales, etc. Las sesiones prácticas se realizan con cadáveres de perro en el anfiteatro, donde el estudio se hace por medio de disecciones, que realizan los propios alumnos, asesorados por su profesor y por los ayudantes.

A la materia de Anatomía II se le da un enfoque aplicativo, es decir, el conocimiento de la Anatomía adquirido tendrá una aplicación en el desarrollo de sus siguientes materias y en el desempeño de su vida profesional. Para su estudio dividen el cuerpo del animal en pequeñas áreas, estudiando de éstas, todas y cada una de sus estructuras y su utilidad.

Para poder realizar esta división deben de apoyarse en puntos visibles o palpables que sirvan como referencia para delimitar el área de estudio, por lo que la práctica no sólo se realiza con cadáveres sino también con animales vivos.

El proceso de enseñanza-aprendizaje.

El profesor imparte la clase teórica apoyándose en diapositivas, esquemas, o modelos, etc. El alumno observa este material y toma sus notas, realiza sus prácticas con los cadáveres y animales vivos pero no tiene apoyo para el estudio de los conocimientos adquiridos y para la realización de exámenes. Este apoyo no se tiene por falta de espacio o de recursos humanos.

2.2 Objetivos del sistema

Los objetivos del "SIAV II" (Sistema Interactivo de Anatomía Veterinaria II) son:

- Proporcionar al alumno un medio de aprendizaje autodidacta, que por sus características innovadoras y a través de una interface interactiva, sea una motivación para que el usuario dedique tiempo al estudio de la Anatomía.
- Proporcionar al profesor de asignatura un auxiliar en la enseñanza de Anatomía II, así como en la evaluación del aprendizaje alcanzado por el alumno.
- Hacer uso de nuevas formas de interacción hombre-computadora.
- Explotar y aprovechar los recursos que brinda la tecnología para el diseño y construcción de aplicaciones interactivas orientadas a la educación.
- Poner al alumno en contacto con tecnología de punta al interactuar con hardware y software orientado a multimedia.
- Permitir que sus usuarios lo puedan usar en cualquier computadora PC compatible que soporte el ambiente gráfico Windows.

2.3 Especificación del sistema

Para lograr que el contenido informativo de la aplicación educativa sea de máxima calidad, es necesario trabajar en colaboración con un experto en el tema sobre el que versará dicha aplicación. Independientemente de las investigaciones bibliográficas que se realicen deberán definirse los alcances del sistema (en cuanto a contenido teórico y funcionamiento) en las recomendaciones y experiencia del experto. Definimos también en esta etapa los medios de despliegue y reproducción de información (audio, vídeo, animaciones, texto, imágenes fijas o hipertexto) factibles de ser usados en la aplicación, así como la configuración de hardware y software con que deberán contar los usuarios potenciales para poder correr la aplicación en óptimas condiciones. Esto permitirá al diseñador de la aplicación y al experto en el tema proponer el uso de uno u otro medio de despliegue y reproducción para presentar información específica.

2.3.1 Alcances del sistema

Para el caso del "SIIV II", se tuvo el apoyo de un experto en el tema de Anatomía II. Con cuya colaboración, se definieron y delimitaron los alcances del sistema. El sistema "SIIV II" contemplará los siguientes temas de estudio, de la materia de Anatomía II, de la Facultad de Medicina Veterinaria y Zootecnia:

1. Generalidades.

- ☐ Cabeza
- ☐ Cuello
- ☐ Tronco
- ☐ Miembros locomotores

2. Sistema Tegumentario.

- ☐ Características macroscópicas
- ☐ Tegumento
- ☐ Piel
- ☐ Apéndices cutáneos

Dentro del sistema el usuario podrá estudiar un tema en forma teórica o práctica. Esto implica que podrá aprender o reforzar sus conocimientos y evaluar su rendimiento, determinando deficiencias o avances y comparar su desempeño con el requerido para aprobar la materia.

La gran mayoría de las computadoras que se usan actualmente son IBM PC's o compatibles y gran parte de éstas tienen procesador 80386 o superior, la interfaz gráfica de usuario más popular es Windows, así aprovechando esta circunstancia, se decidió que "SIIV II" deberá ser diseñado y construido para trabajar sobre esta interfaz gráfica y, por lo tanto, deberá cumplir con sus estándares.

Una de las limitaciones más importantes es el hecho de que la aplicación debe permitir a sus usuarios hacer uso de ésta en cualquier PC que soporte Windows. Esto significa que el programa debe ser capaz de trabajar en computadoras que muy probablemente no cuenten con una tarjeta de audio, lo cual implica que no puede emplearse este medio para reproducir información descartando esta opción para su uso dentro del sistema.

Algo muy similar sucede con el video, ya que una resolución VGA de 16 colores (que actualmente se requiere como mínimo para trabajar en Windows) no ofrece la calidad necesaria para presentarlo de manera conveniente, además de que si la velocidad de procesamiento del microprocesador de la computadora en la que está corriendo el sistema es baja (caso del 80386), la presentación del video será muy lenta, afectándose como consecuencia el desempeño de la aplicación.

Por la misma razón, el uso de *animaciones* y de *imágenes fijas* se debe limitar a resoluciones de 16 colores y a animaciones cortas que no requieran de mucha capacidad de procesamiento. Se emplearán también el *texto* y el *hipertexto*, al trabajar con el sistema teóricamente el usuario tendrá la facilidad de señalar una palabra, una frase, un objeto o una imagen y saltar inmediatamente a información detallada acerca del tema seleccionado.

Las figuras 2-1 y 2-2, ejemplifican las imágenes que se manejan en el estudio de la materia de Anatomía II [POP81].

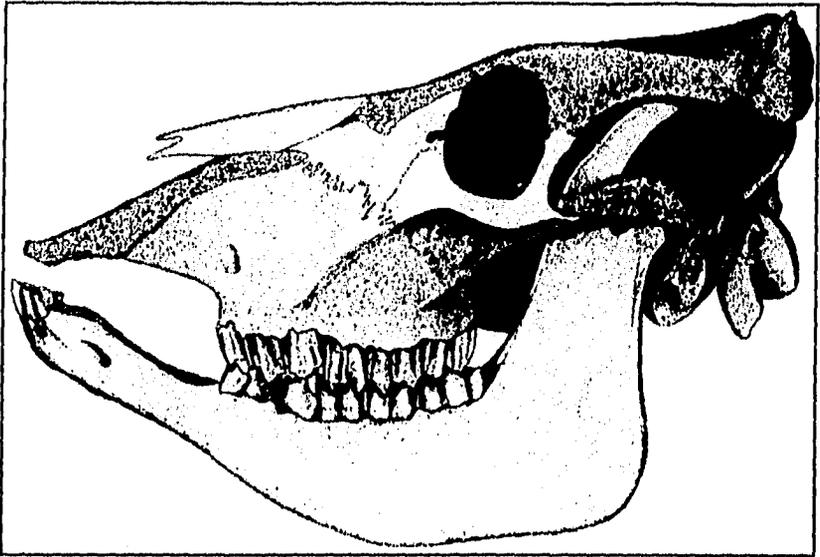


Fig. 2-1 Bovino. Cráneo vista lateral izquierda.

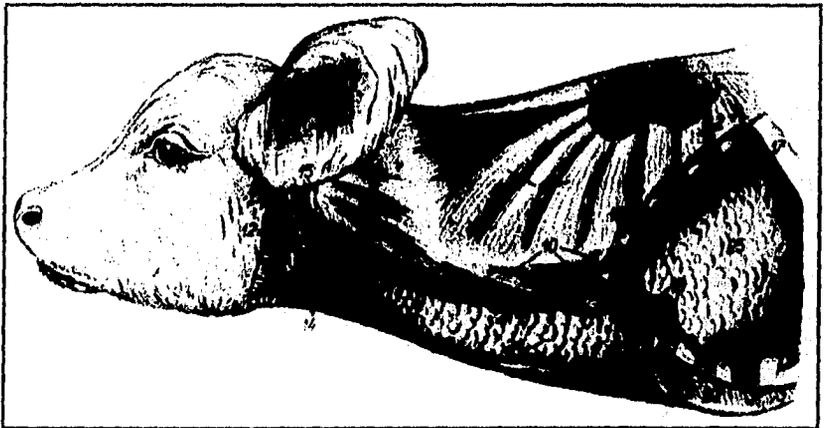


Fig. 2-2 Ternero. Disección cervical del cuello.

2.4. El ámbito de la información

Todas las aplicaciones del software pueden colectivamente llamarse *procesamiento de datos*. Este término contiene la clave para la comprensión de los requisitos del software. El software se construye para procesar *datos*: para transformar datos de una forma a otra, es decir, para aceptar una entrada, manipularla de alguna forma y producir una salida.

El software también procesa *sucesos*. Un suceso representa algún aspecto de control del sistema y realmente no es más que un dato booleano (verdadero o falso, si o no, existencia o no existencia). Por tanto, dentro del ámbito de de información del problema residen tanto los datos (números, caracteres, imágenes, sonidos, etc.) como el control (sucesos).

2.4.1 Modelización.

Durante el análisis de los requisitos del software, creamos modelos del sistema a construir [PRE93]. Los modelos se centran en lo que tiene que hacer el sistema y no en cómo lo tiene que hacer y desempeñan los siguientes papeles importantes:

- El modelo ayuda al analista a entender la información, la función y el comportamiento del sistema, haciendo por ello que la tarea del análisis de requisitos sea más fácil y más sistemática.
- El modelo se convierte en el punto focal para la revisión y por tanto, en la clave para la determinación de la integridad, la consistencia y la eficacia de la especificación.
- El modelo se convierte en la base del diseño.

2.4.1.1 Diagramas de flujo de datos (DFD).

Un diagrama de flujo de datos (DFD) es un modelo que describe los flujos de datos y los procesos que cambian o transforman los datos en un sistema [BUR92]. Dos tipos populares de DFD son los diagramas de burbujas, que utilizan círculos para representar los procesos y líneas curvas para el flujo de datos; y rectángulos con esquinas redondeadas y líneas rectas para el flujo de datos. La primera clase se asocia con De Yourdon y Tom DeMarco. La segunda clase utiliza los símbolos mostrados en la figura 2-3 y se asocia típicamente con Chris Gane y Trish Sarson. Todas estas personas son proponentes del análisis y diseño estructurado.

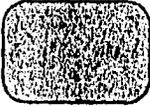
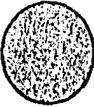
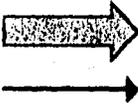
Descripción	Clase 2	Clase 1
Proceso		
Flujo		
Conectores		
Entrada/salida	Mostrada en las líneas de flujo	
Almacenamiento de datos o archivo		
Fuente o destino de los datos		

Fig. 2-3 Símbolos para los diagramas de flujo de datos.

2.4.1.2 Jerarquía más entrada proceso salida (HIPO).

La IBM desarrolló el HIPO, una técnica de diagramación que emplea un conjunto de diagramas para mostrar la entrada, la salida y las funciones de un sistema o de un programa [BUR92]. Sirve como *herramienta de análisis y diseño*, así como para la documentación del sistema. El HIPO facilita un diseño descendente (top-down). No sustituye a los diagramas de flujo, pero los complementa.

El HIPO implica el empleo de dos tipos de diagramas: (1) un diagrama jerárquico, o tabla visual de contenidos (TVC), que muestra cómo encajan los módulos de la aplicación y (2) un diagrama general o diagramas funcionales de detalle que muestran la entrada, el procesamiento y la salida para cada módulo.

Diagrama jerárquico o Tabla visual de contenidos (TVC).

Es representado por rectángulos o cajas. Cada caja puede representar un sistema, un subsistema, un programa o un módulo de programa. Su propósito es el de exhibir los módulos funcionales generales y hacer referencia a los diagramas generales y de detalle HIPO. Los módulos se presentan en un nivel de detalle creciente.

Diagrama general o funcional de detalle.

El diagrama funcional, ya sea a nivel general o a nivel de detalle, consta de una caja de entrada, una caja de proceso y una caja de salida. En el TVC se crea un diagrama funcional para cada módulo. Los símbolos de los diagramas de flujo tradicionales representan los medios, como cinta magnética, disco magnético, y salida impresa.

2.5 Diagramas del "SIIV II"

El diagrama general de flujo de datos de la aplicación se obtuvo del análisis realizado, contemplando el flujo de la información, sus entradas y salidas (Fig. 2-4).

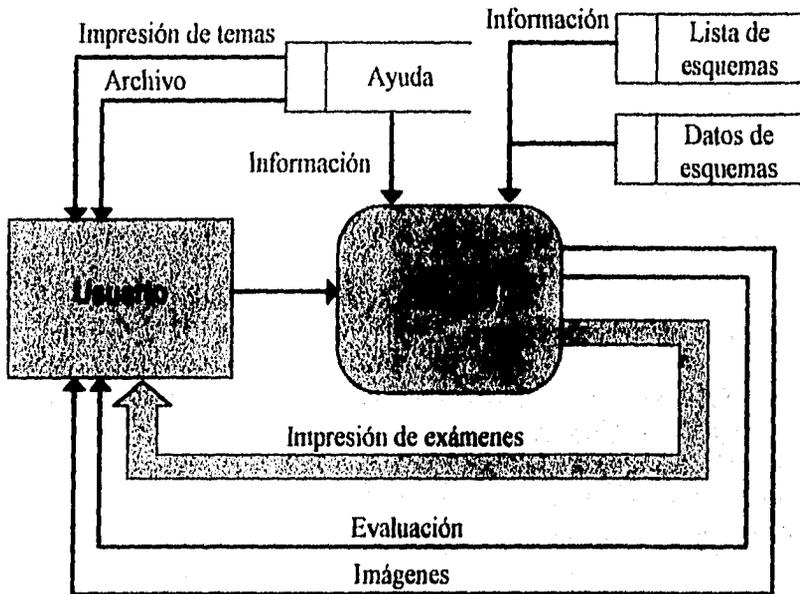
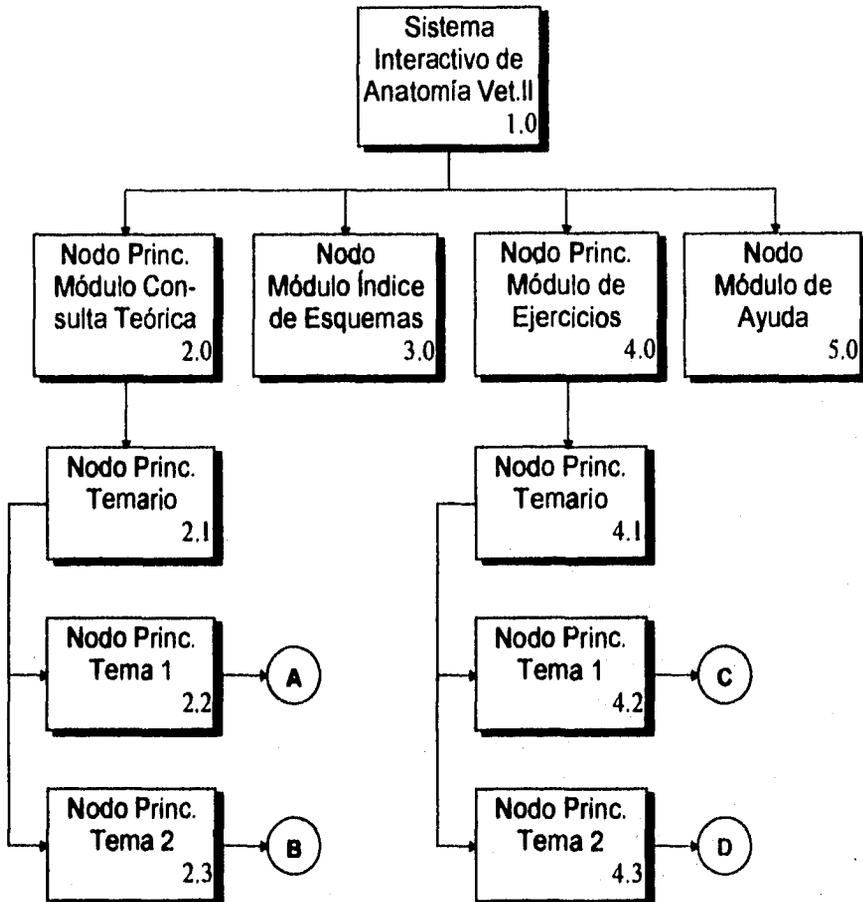
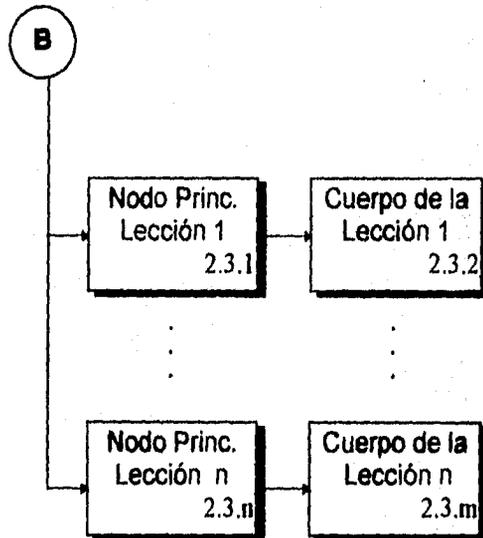
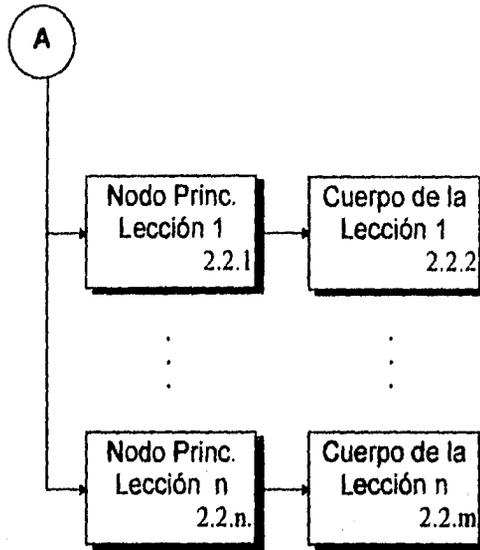


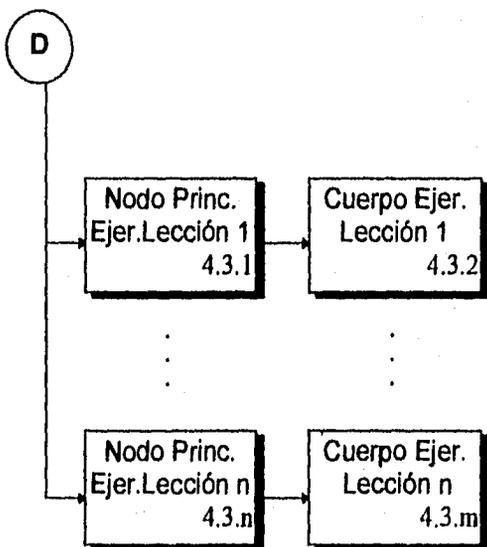
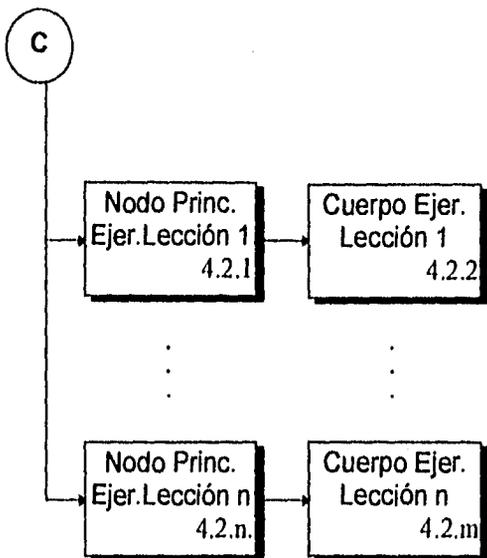
Fig. 2-4 Diagrama de flujo de datos (DFD), del "SIIV II".

Se diseñó la tabla visual de contenidos (TVC), basándose para esto en el flujo del proceso de enseñanza de los temas. El trabajo de las etapas siguientes se basará en este diagrama (Fig. 2-5).

Fig. 2-5 Tabla visual de contenidos HIPO, del "SIAV II".







La figura 2-5 muestra la tabla visual de contenidos HIPO del "SIIV II". Como se puede apreciar, esta estructura tiene como raíz un nodo, o pantalla principal, a partir del cual es posible llegar a cuatro nodos: nodo principal de consulta teórica, nodo principal de ejercicios, nodo de índice de esquemas y nodo de ayuda del tema.

El primer nodo, de consulta teórica, es un apoyo en el aprendizaje reforzando la teoría impartida en el temario de la materia. El nodo principal de ejercicios ofrece al estudiante la opción de realizar dos tipos de exámenes con los que podrá reforzar los conocimientos adquiridos en la teoría. El nodo de índice de esquemas permitirá de forma rápida elegir un esquema y trabajar con él, mientras que en el nodo de ayuda, se ofrecerá información detallada referente a palabras o frases (*hipertexto*), que el estudiante puede señalar para profundizar en algún tema de interés particular.

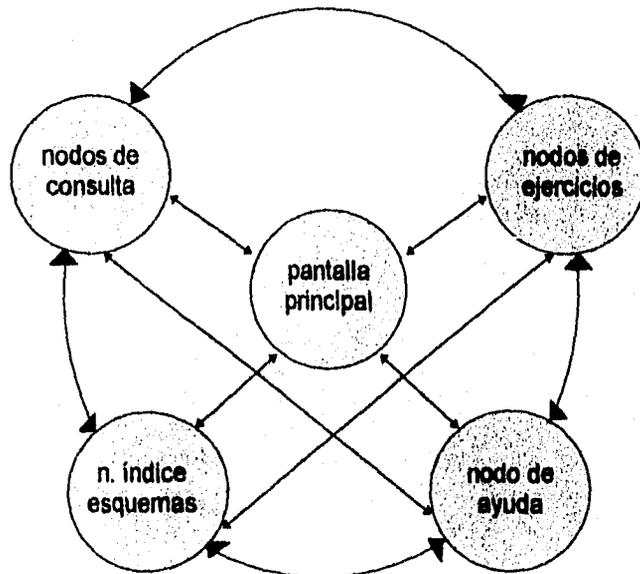


Figura 2-6. Diagrama de navegación del "SIIV II" (DFD "Burbuja").

El cuerpo principal de "SIIV II" se centrará en los nodos de consulta teórica y ejercicios, estos permitirán que el estudiante pueda elegir un tema específico para consultar, o bien, para hacer ejercicios respectivamente y dirigirse directamente a él.

La figura 2-6 muestra la "navegación" del "SIIV II", es decir, las posibilidades que ofrecerá el sistema para ir de una sección a otra del mismo. De esta figura se puede observar que la navegación del sistema permitirá ir a cualquier punto que desee el usuario sin importar donde se encuentre; el usuario podrá "saltar de un área a otra", excepto en el caso de que se esté realizando un examen en el módulo de ejercicios.

Capítulo 3

Diseño del sistema

La fase de desarrollo se centra en el cómo. Esto es, durante esta fase, el que desarrolla el software intenta descubrir cómo han de diseñarse las estructuras de datos y la arquitectura del software, cómo han de implementarse los detalles procedimentales, cómo ha de traducirse el diseño a un lenguaje de programación (o lenguaje no procedimental) y cómo ha de realizarse la prueba.

Los métodos aplicados durante la fase de desarrollo variarán, pero de alguna forma se producirán tres pasos concretos: diseño, implementación y prueba del software. Cada uno de los cuales se aplicarán a partir de este momento para el desarrollo del "SIIV II".

El diseño es técnicamente la parte central de la ingeniería del software. Durante el diseño se desarrollan, se revisan y se documentan los progresivos refinamientos de las estructuras de datos, de la estructura del programa y de los detalles procedimentales. El diseño da como resultado representaciones del software cuya calidad se puede evaluar.

Durante las tres últimas décadas, se han propuesto varios conceptos fundamentales del diseño de software [PRE93].

- Los conceptos de *modularidad* (tanto de programa como de datos) y de *abstracción* permiten simplificar y reutilizar los componentes de software.
- El *refinamiento* es un mecanismo que permite representar sucesivas capas de detalle funcional.

- La *estructura de programa y de datos* contribuyen a la visión general de la arquitectura del software.
- El *procedimiento* proporciona los detalles necesarios para la implementación de los algoritmos.
- El *ocultamiento de la información y la independencia funcional* son heurísticas que permiten conseguir una modularidad efectiva.

El diseño del software puede verse desde las perspectivas técnica y de gestión del proyecto. Desde el punto de vista técnico, el diseño comprende cuatro actividades: diseño de los datos, diseño arquitectónico, diseño procedimental y diseño de interfaces. Desde el punto de vista de gestión del proyecto, el diseño va del diseño preliminar al diseño detallado.

A las entradas y salidas diseñadas por el usuario y a las restricciones de tiempo se les conoce como requerimientos funcionales del sistema. Estos requerimientos funcionales proporcionan la información necesaria para plantear una serie de alternativas de solución que se transforman en especificaciones funcionales del sistema.

Las especificaciones funcionales del sistema corresponden a las entradas, salidas, interfaces y algoritmos de cada módulo, es decir, contienen toda la información de los subsistemas. Un subsistema o módulo es un componente de un programa compilado por separado que contiene datos y estructuras de control.

Después de definir los requerimientos de un sistema, se debe desarrollar en varias etapas el diseño de programación que satisfaga éstos. Dichas etapas se mencionan a continuación:

- Es necesario definir de manera clara los objetivos que se esperan del sistema, ya que éstos determinan el criterio que deberá tomarse para las decisiones individuales, que se harán en el diseño del sistema.
- Deben conocerse los puntos de decisiones que existen, así como reconocer las posibles opciones que pueden presentarse, para así tomar decisiones claras.
- Deben establecerse los subsistemas que componen el sistema de programación.

- ☑ Cada subsistema debe dividirse en componentes individuales y ha de establecerse la especificación de éstos, definiendo la operación de esos componentes.
- ☑ El siguiente paso es refinar cada componente, lo que implica la especificación de cada uno, así como una jerarquía de subcomponentes.
- ☑ Durante el proceso de refinamiento deben especificarse con detalle los algoritmos empleados en cada componente.
- ☑ Finalmente es necesario conocer alguna metodología o técnicas que auxilien en la toma de decisiones para construir un diseño satisfactorio.

Algunos *factores* que afectan la *calidad del software* y hacen del diseño una fase que debe realizarse para la solución de una problemática [PRE93] se exponen a continuación:

Eficiencia. Es la cantidad de recursos de computadora y de código requeridos por un programa para llevar a cabo sus funciones (CPU, memoria primaria, dispositivos de almacenamiento secundarios, tiempo de uso de los periféricos de entrada/salida, tiempo de proceso de la información, etc.).

Corrección. Es el grado en que un programa satisface sus especificaciones y consigue los objetivos de la misión encomendada por el cliente.

Integridad. Es el grado en que puede controlarse el acceso al software o a los datos, por personal no autorizado.

Fiabilidad. Es el grado en que se puede esperar que un programa lleve a cabo sus funciones esperadas con la precisión requerida.

Facilidad de uso. Es el esfuerzo requerido para aprender un programa, trabajar con él, preparar su entrada e interpretar su salida.

Facilidad de mantenimiento. Es el esfuerzo requerido para localizar y arreglar un error en un programa.

Acoplamiento. Es una medida del grado de independencia entre módulos. A menor acoplamiento entre los módulos, mayor independencia.

Cohesión. Es el factor esencial del diseño, con él se puede medir la fuerza de asociación de las instrucciones y llamadas que efectúa un módulo. En un programa con cohesión, los elementos muestran un alto grado de relación funcional.

Existe una relación importante entre acoplamiento y cohesión: a mayor grado de cohesión de un módulo se tiene un menor acoplamiento entre los módulos. Es preferible mantener una alta cohesión aunque haya una pérdida de acoplamiento.

3.1 Técnicas del diseño

El diseño efectivo del software se logra utilizando una metodología consistente de diseño. Hay una gran cantidad de metodologías de diseño desarrolladas y que se utilizan en diferentes aplicaciones. En esencia, la mayoría de éstas pueden clasificarse en una de las tres áreas siguientes [PRE93]:

1. *Diseño funcional descendente.* El sistema se diseña desde un punto de vista funcional, empezando con una visión de alto nivel y refinándola de manera progresiva hasta llegar a un diseño más detallado. Dicha metodología está ejemplificada por el diseño estructurado y el refinamiento de pasos.
2. *Diseño orientado al objeto.* El sistema se ve más como una colección de objetos que como funciones que pasan mensajes de un objeto a otro. Cada objeto tiene su propio conjunto de operaciones asociadas. El diseño orientado a objetos se basa en la idea de ocultamiento de información.
3. *Diseño controlado por los datos.* Plantea que la estructura de un sistema de software debe reflejar la estructura de los datos que éste

procesa. Por tanto, el diseño de software se obtiene de un análisis de los datos del sistema de entrada y salida.

La técnica que se utilizará para el diseño del sistema será, el *diseño funcional descendente*.

3.2 Diseño de una interfaz gráfica efectiva para el usuario

Una *interfaz* es aquel componente de una aplicación que traduce una acción del usuario en una o más peticiones para desarrollar funciones y retroalimentar al usuario con las consecuencias y sus acciones.

La creación de una aplicación efectiva comienza con un buen diseño y una visión amplia del contenido y propósito de ésta; la manera en que se organiza depende de la información que contiene y de cómo se espera que el usuario la utilice. Se deberá tener una aplicación amplia en la cual los usuarios navegarán fácilmente en ella siguiendo una trayectoria lógica, intuitiva y no lineal marcada con indicaciones claras.

Una aplicación bien diseñada debe guiar a los usuarios de un punto a otro, permitiéndole abandonar el proceso en cualquier momento, por lo tanto, los controles que estén dentro de la aplicación se convertirán en las señales que guiarán al usuario dentro de la aplicación.

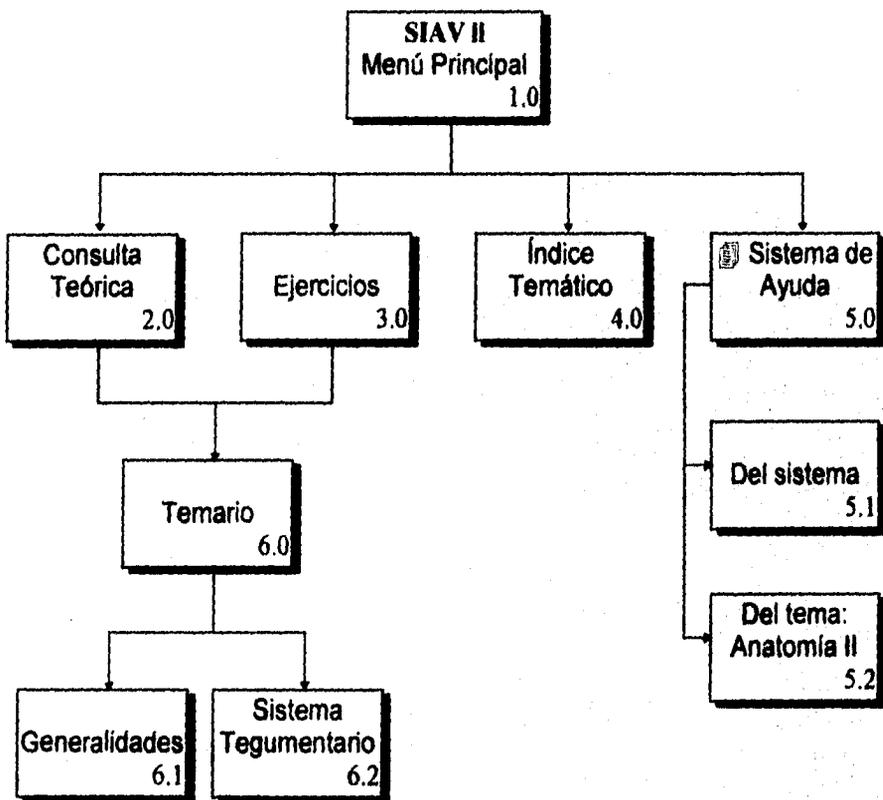
3.3 Diagramas del sistema

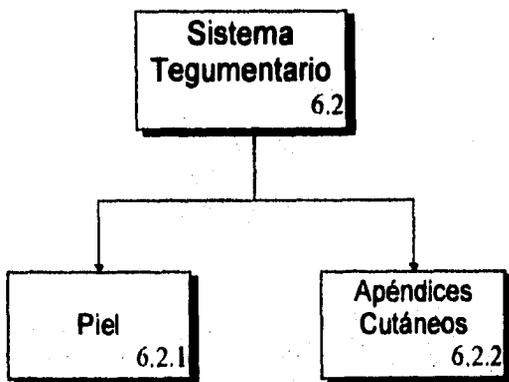
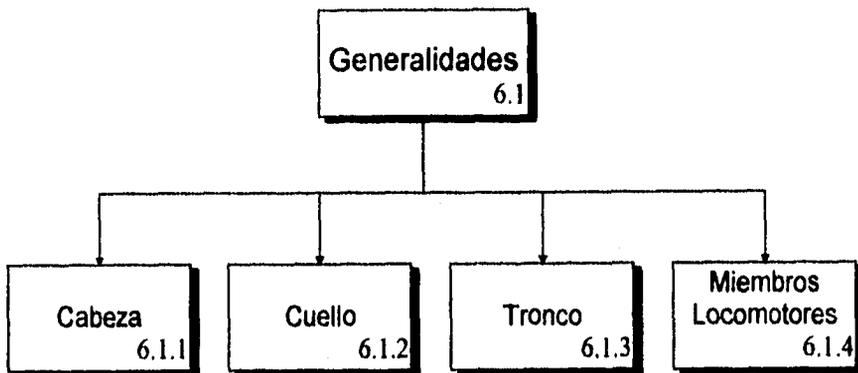
El objetivo fundamental del *diagrama TVC* (HIPO) o *diagrama jerárquico*, es proporcionar la estructura de los módulos del sistema que facilite el entendimiento de sus funciones y una relación jerárquica de sus elementos. En el capítulo anterior se explicó más a detalle.

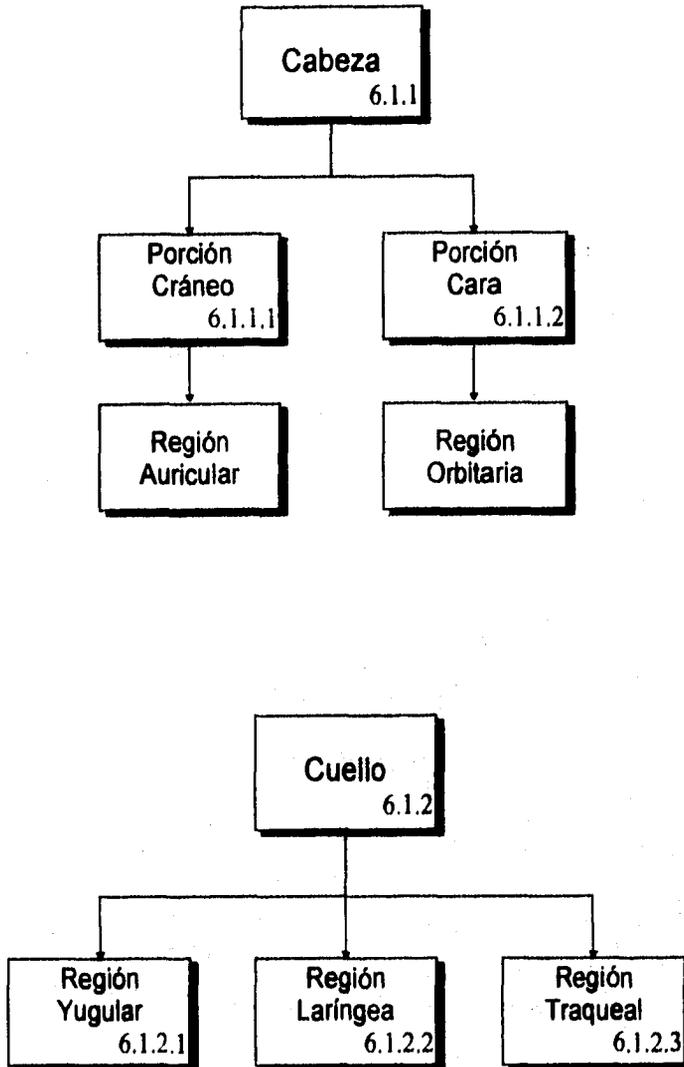
El diagrama debe leerse de izquierda a derecha y de arriba hacia abajo, tratando de identificar la función o funciones que se desean analizar, de esta manera puede ser utilizado para localizar una función específica o entender al sistema a un cierto nivel de la estructura.

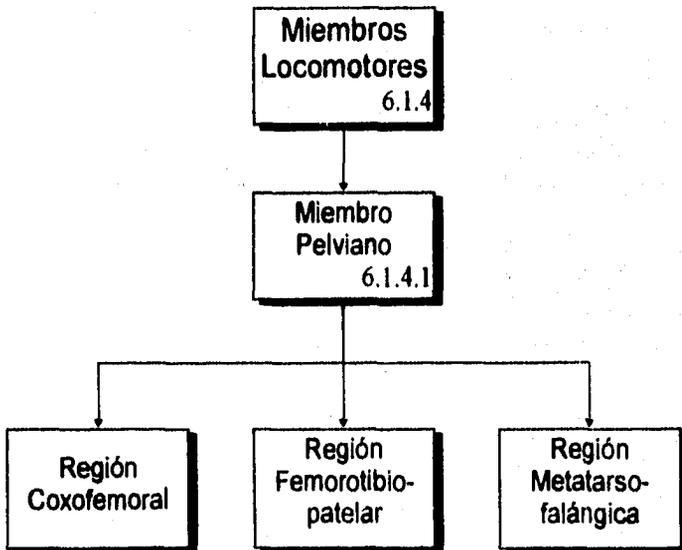
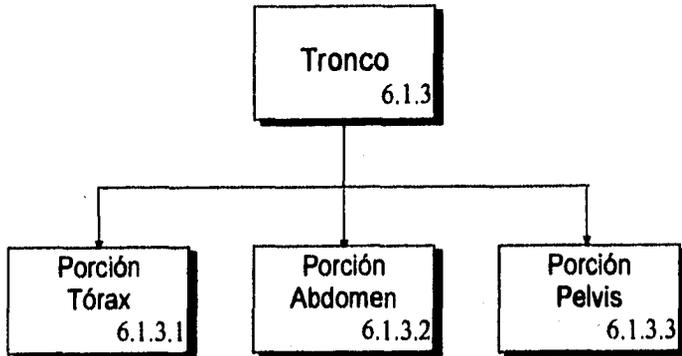
A continuación se presentan los diagramas jerárquicos (TVC) del "SIAV II" (Sistema Interactivo de Anatomía Veterinaria II).

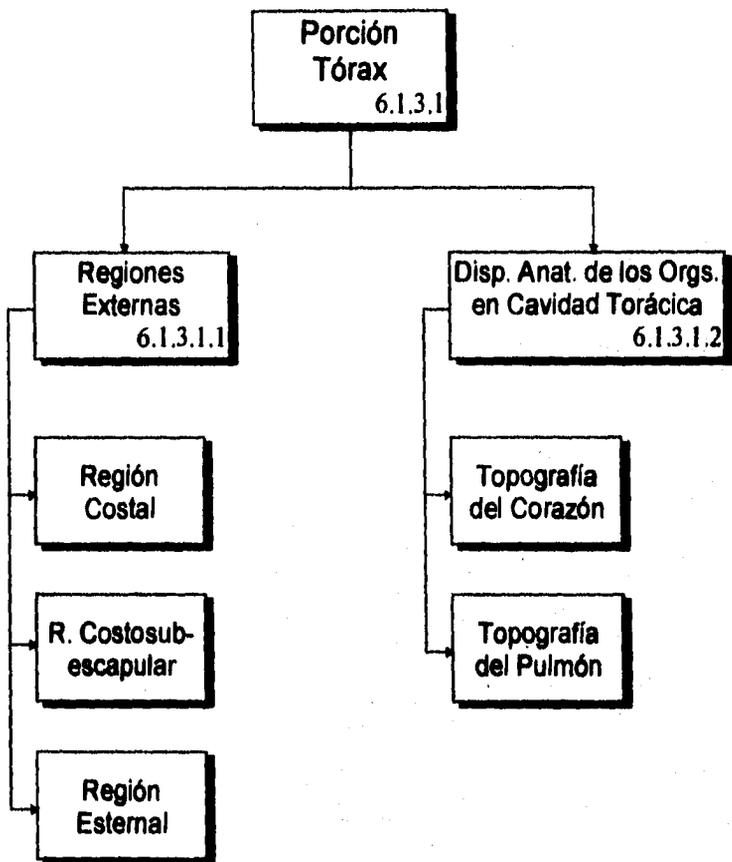
3.3.1 Diagrama jerárquico o TVC del "SIAV II".

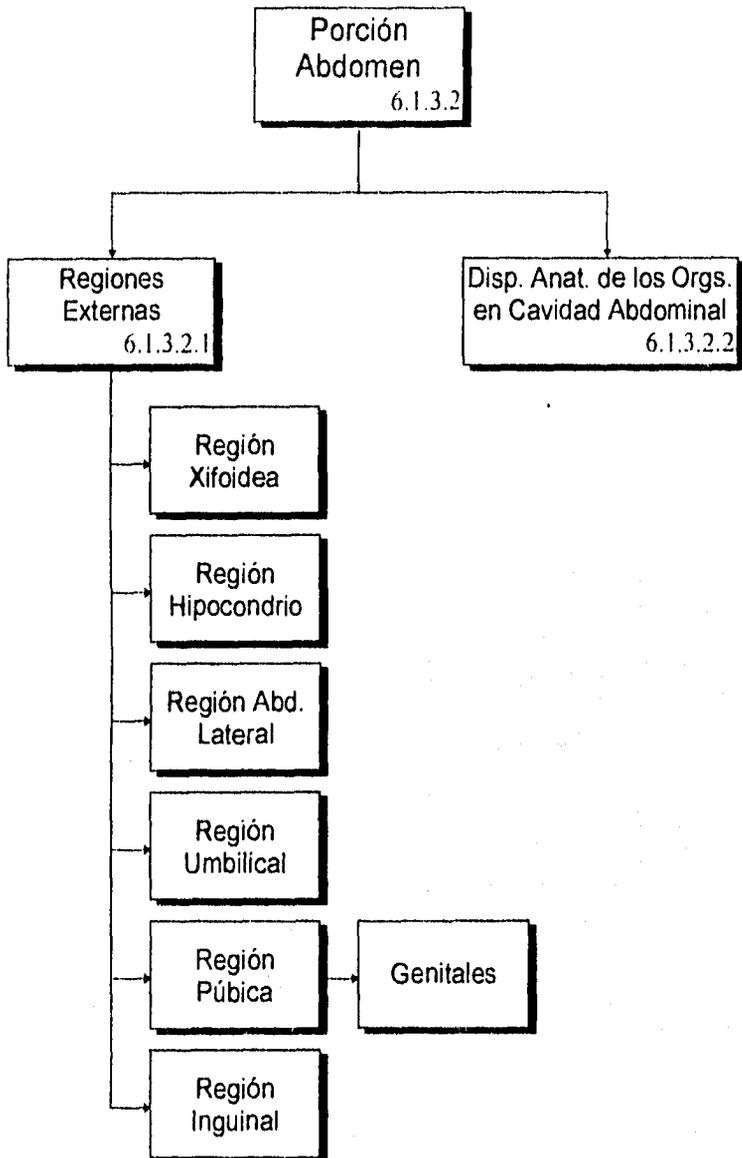


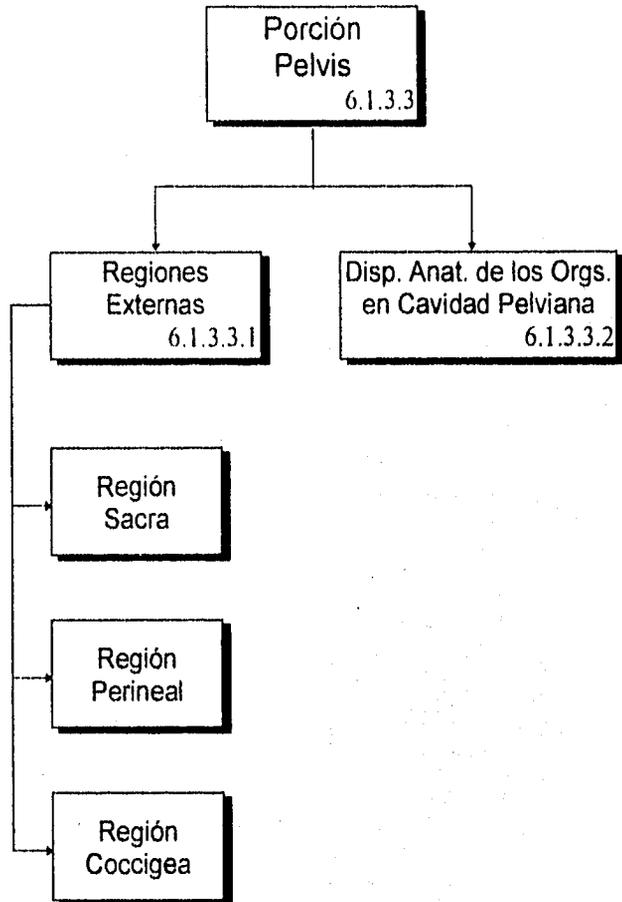


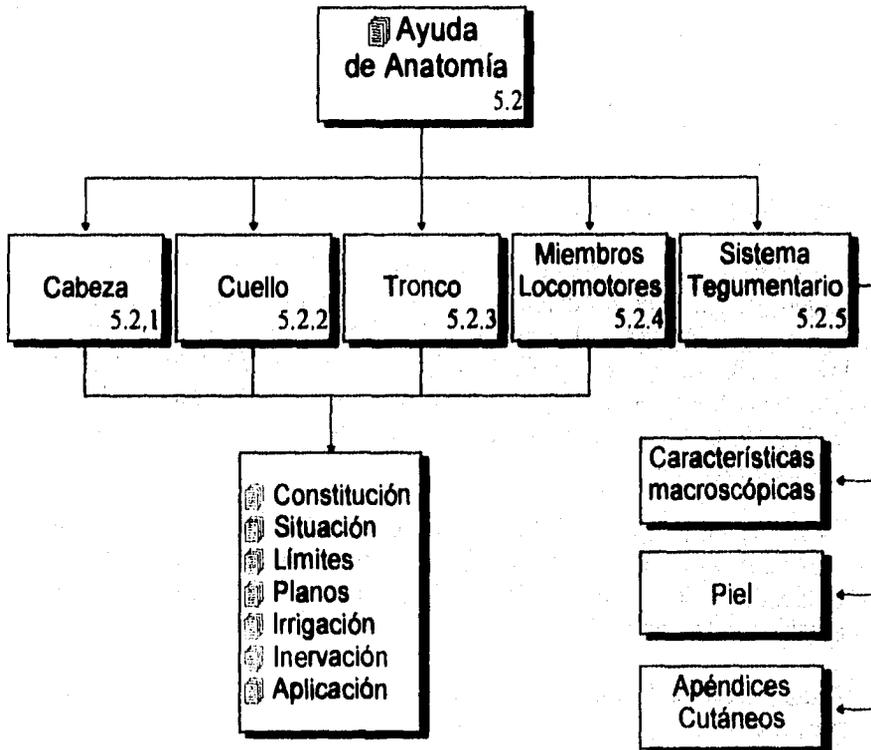
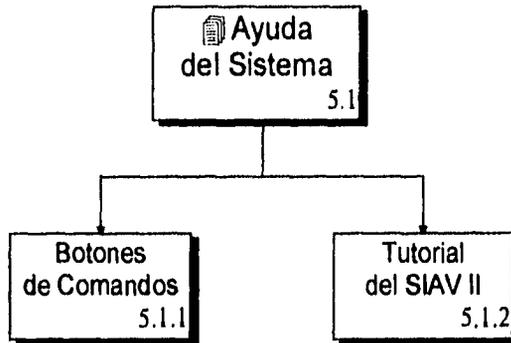












3.4 Definición de los conjuntos de información

Los conjuntos de información se dividen en tres tipos y son:

1. Conjuntos de información de entrada.
2. Conjuntos de información de salida.
3. Conjuntos de información de bases de datos.

Conjuntos de información de entrada del sistema.

El proceso de lectura de la información de entrada del "SIIV II" es transparente para el usuario, lo cual facilitará su operación. Los conjuntos de información de entrada, son todos aquellos elementos de información necesarios para la correcta operación de la programación y que usualmente se capturan a través de los diferentes dispositivos de lectura, a través del teclado respondiendo a preguntas, resolviendo ejercicios y movimientos de navegación en el sistema.

Conjuntos de información de salida del sistema.

El despliegue de imágenes con su información, la impresión de los esquemas (por tipo de examen), la evaluación obtenida al realizar un examen y el archivo de "ayuda del sistema", el cual contendrá la información relativa al manejo del mismo; por medio del éste el usuario se familiarizará con el "SIIV II" o podrá resolver dudas acerca de las funciones de cada control, ventanas, etc..., es decir, del entorno en el que se va a mover.

El archivo de "ayuda de los temas" se relacionará con el uso del *hipertexto* en cada parte del temario de la materia de Anatomía, proporcionando al usuario información opcional y concreta sobre lo que estará viendo y aprendiendo.

Cada módulo del sistema esperará una respuesta a todos los eventos asociados.

Conjuntos de información de bases de datos.

Se crearán las bases de datos que deberán contener la información necesaria para el funcionamiento del sistema. Es decir, los datos que se manejan en el mismo.

Se tendrán dos tipos de bases de datos 1) donde se almacenará la lista de nombres de los esquemas y 2) donde estarán los datos señalados en los esquemas.

Su estructura es la siguiente:

BD1. Lista de esquemas.

<i>Nombre del Campo</i>	<i>Tipo</i>	<i>Longitud</i>
Clave	Caracter	7
N_esquema	Caracter	50

BD2, BD3, ...BD5. Esquemas.

<i>Nombre del Campo</i>	<i>Tipo</i>	<i>Longitud</i>
Clave	Caracter	7
Inciso	Caracter	1
Nombre	Caracter	40

Clave. Dato alfanumérico que identifica a un esquema y sirve de liga con las otras bases de datos.

N_esquema. Es el nombre de un esquema.

Inciso. Respuesta de un examen de "opción múltiple".

Nombre. Respuesta de un examen "completar respuesta".

 **PK.** Llave primaria, es un atributo elegido como identificador de una tabla.

Capítulo 4

Implementación del sistema

Las representaciones del diseño deben ser traducidas a un lenguaje. El paso de la codificación es el que lleva a cabo esa traducción. Se traduce el diseño detallado a un lenguaje de programación que, por último, es (automáticamente) transformado en instrucciones ejecutables por la computadora.

Una vez que el software ha sido implementado en una forma ejecutable por la máquina, debe ser probado para descubrir los defectos que puedan existir en la función, en la lógica y en la implementación.

4.1 Selección del software

Las características psicológicas y técnicas de un lenguaje de programación afectan a la facilidad de la traducción del diseño y al esfuerzo requerido para la prueba y el mantenimiento del software. Estas características se pueden aplicar a los lenguajes de programación que entran en una de las cuatro generaciones de lenguajes [PRE93].

Por lo tanto, el lenguaje debe tener características de control y estructuración de los datos que permitan producir un programa entendible.

La elección de un lenguaje de programación para un proyecto específico debe tener en cuenta tanto las características de ingeniería como las psicológicas. 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, lo cual es el caso [PRE93].

De acuerdo con Ian Sommerville [SOM88], se tomaron en cuenta dos **criterios para la selección** del software. A continuación se describen.

1. *Los requisitos del contratista del sistema.* El contratista de un sistema de software puede especificar que se use un lenguaje de programación en particular y, en general, se debe respetar ese requisito. El contratista también puede proporcionar una lista de lenguajes apropiados y se debe elegir uno de ellos para realizar el proyecto. Corresponde a los diseñadores y realizadores del sistema decidir cuál es el lenguaje más apropiado para cada proyecto en particular.
2. *El conocimiento del personal de programación existente.* Aunque no supone una dificultad especial para los programadores experimentados aprender un nuevo lenguaje, necesitan cierta práctica del lenguaje antes de adquirir una verdadera competencia. Si no hay otros factores en contra, es deseable elegir un lenguaje con el que esté familiarizado el personal de programación.

Considerando lo anterior, la elección se realizó en conjunción con el usuario y el asesor del trabajo de tesis, tomando en cuenta los recursos con que se contaba. Como lenguaje de programación y manejador de bases de datos, **Visual Basic y Dbase III**, respectivamente.

En conclusión se utilizará el siguiente software:

- ☐ Sistema operativo MS-DOS versión 3.0 o superior.
- ☐ Windows versión 3.0 o superior.
- ☐ Visual Basic 3.0 para Windows.
- ☐ Dbase III Plus.

4.1.1 El entorno de Windows.

Windows no es sólo un software que se ejecuta sobre el sistema operativo, sino que en cierta medida lo reemplaza; dando al usuario un entorno gráfico amigable para su interacción con la parte de *hardware*. A nivel interno posee toda una serie de características destinadas tanto a facilitar su uso, como a mejorar las presentaciones del sistema operativo.

Las principales ventajas que nos ofrece este entorno son:

- **Homogeneidad del entorno.** (CUA, Common User Acces). La inmensa mayoría de las aplicaciones Windows presenta al usuario una interface estandarizada, de esta forma su aspecto es siempre similar y su utilización idéntica, lo que facilita enormemente el aprendizaje.
- **Intercambio de datos entre aplicaciones.** En el entorno de Windows es común realizar fácilmente el intercambio de datos entre aplicaciones mediante un método estandarizado, como por ejemplo, el uso del portapapeles.
- **Gestión de toda la memoria.** Con la aparición del procesador 80286, Windows puede acceder a toda la memoria que se tenga disponible.
- **Independencia del hardware.** Windows permite ejecutar una misma aplicación en múltiples configuraciones de hardware, por ejemplo, con distintas tarjetas gráficas, distintas impresoras, etc. Esto es debido a que las aplicaciones no necesitan administrar los distintos recursos disponibles, sino que se limitan a solicitar a Windows la gestión necesaria en cada momento y es éste el que con ayuda de los controladores adecuados (drivers, proporcionados por el fabricante del hardware) se encarga de realizarlo.
- **Multitarea.** Es la ejecución de varias aplicaciones a la vez; como mínimo el propio Windows. Dentro del entorno Windows, se pueden mantener abiertas a la vez varias ventanas y en cada una de ellas puede encontrarse una aplicación distinta.

Cabe añadir que la multitarea Windows, no es, al menos de momento, una multitarea real, en la que el sistema operativo determina el tiempo de CPU que asignará a cada aplicación, sino que es necesaria la colaboración amistosa de todas las aplicaciones, que deben de liberar el uso del CPU a intervalos apropiados para que sea otra aplicación la que se utilice.

Todas estas características también proporcionan una serie de ventajas para el desarrollo de aplicaciones Windows, como son el no tener que realizar una programación específica de los periféricos de entrada y salida, acceso a más memoria, un rápido aprendizaje de la utilización de su aplicación por el usuario y con presentaciones y resultados más elaborados, etc.

La programación en Windows.

La aparición de *Windows* no sólo conmovió la forma de interactuar del usuario con la máquina sino que introdujo una nueva filosofía también para los programadores.

Una de las principales diferencias que presenta la programación en Windows viene impuesta por la multitarea propia del sistema, ya que así, no es una sola aplicación la que se está ejecutando, sino que son varias; como mínimo el propio Windows. Es por eso que no puede hacerse un uso exclusivo de los recursos del sistema. Esto es evidente si pensamos que la pantalla no es exclusiva del programa sino que la salida se limita a una ventana, aún cuando ésta ocupe toda la pantalla. Al existir varias aplicaciones en ejecución y la necesidad de que los recursos del sistema se compartan entre ellas, no puede ser el programa el que administre los recursos, sino tan solo emitir peticiones de adjudicación de los mismos, por ejemplo, supongamos un "click" en el mouse o una pulsación del teclado, la aplicación no puede permanecer esperando la pulsación o el clic, observando continuamente el estado del teclado, como en MS-DOS. En el caso de aplicaciones Windows, aún cuando existan varias, únicamente una es la activa y ella irá dirigida a la pulsación o al clic. El sistema necesita pues *algo* que se encargue de recoger la pulsación o el clic y dirigirlo a la aplicación adecuada, esto es, *Windows*.

El propio entorno de Windows es el que se encarga de vigilar el sistema y, frente a cualquier eventualidad, avisar a la aplicación adecuada. Estos avisos son los que se denominan "sistema de mensajes de Windows", siendo uno de los puntos primordiales de toda la filosofía Windows. En resumen, ante cualquier evento que se produzca en el sistema, Windows genera un mensaje dirigido a la aplicación correspondiente.

Con todo esto se ha introducido un nuevo concepto: la *interactividad* de la aplicación con el entorno.

En una programación tradicional el código es en línea; unas sentencias se ejecutan tras otras sin más posibilidad que esperar entrada de datos en ciertos puntos y continuando su ejecución con posterioridad. En la programación bajo Windows la aplicación consiste en una serie de subrutinas que entran en acción cuando se produce un evento determinado, lo que genera el correspondiente mensaje o la ejecución de nuevos eventos.

Como se ha visto hasta este momento, se ha empleado el término **evento**, pero ¿qué es en realidad un evento dentro de la programación Windows?. Cualquier suceso provocado por el usuario o el propio sistema y que es capaz de generar una acción (por ejemplo el despliegue de un mensaje).

Por ejemplo, un doble clic, una pulsación de teclado, un clic, un movimiento del mouse, el cambio de tamaño de una ventana, etc. Todas estas peculiaridades originan, como es lógico, toda una serie de ventajas e inconvenientes que a continuación se resumen.

Los interfaces gráficos de usuario, o GUI (Graphical User Interface) han revolucionado la industria de las microcomputadoras. Han demostrado que el proverbio "vale más una imagen que mil palabras" no ha perdido validez. En lugar del crítico carácter de aviso de órdenes C:> que los usuarios del DOS utilizan, encuentran un escritorio poblado de iconos. Todo esto da una imagen gráfica de lo que puede ofrecer la computadora.

Quizá más importante a largo plazo que el aspecto de las aplicaciones para Windows es la sensación que proporcionan. Las aplicaciones para Windows tienen generalmente una interfaz de usuario consistente [NEL94]. Esto significa que los usuarios disponen de más tiempo para dominar la aplicación sin tener que preocuparse de qué teclas deben pulsarse, dentro de los menús y cuadros de diálogo.

4.1.2 Lenguaje de programación Visual Basic.

4.1.2.1 Programación orientada a eventos.

Para el logro de nuestro objetivo se utiliza uno de los primeros lenguajes de programación llamada "orientada a eventos", un estilo de programación especialmente adaptado a los interfaces gráficos de usuario.

El sistema de programación Visual Basic para Windows supone un importante avance para cualquiera que esté involucrado en la creación de aplicaciones para Windows. Su corazón de programación orientada a eventos y sus innovadoras herramientas de diseño visual, Visual Basic permite sacar el máximo provecho del entorno gráfico Windows para crear potentes aplicaciones con rapidez.

4.1.2.2 ¿Qué significa orientado a eventos?

Tradicionalmente, la programación ha estado muy orientada al proceso, al paso a paso, de manera muy similar al de una receta; uno de los inconvenientes de este estilo consiste en que la persona que escribió la receta (el programa) es la que determina qué hay que hacer en cada momento, pero en los modernos programas el objetivo es que sea el usuario el que determine en cada momento qué es lo que quiere hacer.

Y eso es exactamente lo que proporciona la programación orientada a eventos, la cual se rige por determinadas secciones de código que se accionan sólo si ocurre una determinada acción. En lugar de escribir un programa que determina cada uno de los pasos en un orden

determinado, el programador escribe un programa que responde a las acciones del usuario: elegir un comando, hacer clic en una ventana, mover el ratón. En vez de escribir un gran programa, el programador crea una aplicación que es realmente una colección de microprogramas que cooperan entre ellos y que se ejecutan a raíz de eventos iniciados por el usuario.

El tiempo utilizado en crear un programa o aplicación por este medio, es substancialmente menor, al requerido para realizar uno de manera tradicional.

4.1.2.3 Visual Basic.

En 1991 Microsoft presentó **Visual Basic**. El sistema de programación Visual Basic sortea la complejidad de Windows de una manera realmente espectacular. Combinando las probadas posibilidades del lenguaje Basic con herramientas de diseño visual proporciona simplicidad y facilidad de uso, sin sacrificar prestaciones o las características gráficas que hacen de Windows un entorno tan apetecible en el cual trabajar.

Los programas en los lenguajes de programación convencionales se ejecutan de arriba abajo. En los antiguos lenguajes de programación la ejecución comienza en la primera línea y se desplaza con el flujo del programa a las distintas partes según se necesite. Un programa en **Visual Basic** funciona de un modo totalmente diferente. El núcleo de un programa en Visual Basic es un conjunto de diferentes partes de código que son *activadas por*, y que solamente *responden a*, los sucesos que se les ha indicado que reconozcan. Esto es un avance fundamental. Ahora, en lugar de diseñar un programa que haga lo que el programador piense que debe hacer, el usuario tiene el control.

Más aún, Visual Basic hace sencilla la creación de grandes programas mediante las modernas técnicas modulares de programación. Esto significa que se puede dividir un programa en módulos, más sencillos de manejar, y por lo tanto, menos sensibles a los errores (un *módulo* es una parte relativamente pequeña y manejable de código de programación).

Visual Basic es una herramienta extraordinariamente poderosa y flexible, que le permite a los programadores crear aplicaciones dinámicas y sobre todo con rapidez.

El enfoque tradicional del desarrollo de una aplicación de Windows ha sido programar en "C" de Microsoft y utilizar el "Windows Software Development Kit". El ciclo de desarrollo transcurre de la siguiente manera: se elabora un programa Windows esbozando los módulos principales y la lógica que los enlaza; después, se escribe el programa compilando, enlazando y depurando el código repetidamente. Los iconos, menús, cajas de diálogo y todos los controles en la interfaz del usuario se deben diseñar y compilar por separado; todos estos componentes deben ser controlados por el programa principal, a través de un laberinto de funciones y mensajes.

La filosofía de programación de Visual Basic es, elaborar controles como ventanas, iconos, menús, etc.; después, escribir los procedimientos que invocan a cada uno de los controles.

En resumen las ventajas que nos proporciona Visual Basic son:

- ✕ Cuenta con múltiples facilidades para la elaboración de interfaces gráficas atractivas y eficientes.
- ✕ Proporciona un lenguaje formal de programación.
- ✕ Cuenta con herramientas para el manejo directo de bases de datos.
- ✕ Facilita la incrustación de objetos provenientes de otras aplicaciones, tales como gráficos, imágenes, textos, tablas, etc...

Requerimientos de hardware.

Visual Basic está diseñado para correr en un procesador 80286 o mayor con 1 MB de memoria y Windows 3.0 o posterior. Un disco duro, una unidad de disco de 5.25" o 3.5" y requiere de mouse instalado.

Programación en Visual Basic.

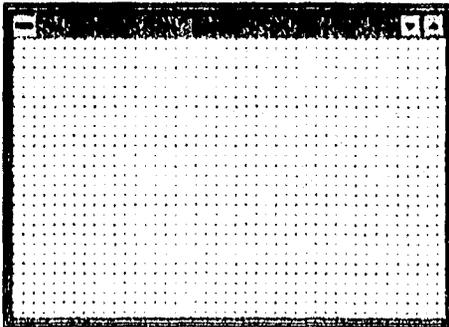
Una *forma* es una ventana que se crea y adapta a sus necesidades para cada aplicación, esto es la base de cualquier aplicación Visual Basic, que eventualmente correrá como un programa independiente en Windows. Un *control* es el nombre preciso para cualquier objeto que se sitúe en una forma. Los botones de comando, menús, cajas de diálogo y barras de desplazamiento, son ejemplos de controles.

Cuando el usuario activa un control al seleccionarlo, el código que está asignado a éste es invocado y realiza una tarea específica. Un *módulo* es una estructura para escribir el código que se asigna a una forma y sus controles. Un *evento* es una acción que es reconocida por un control. Orientado a eventos significa que todos los controles que se utilicen en una forma, especifican cómo se comportará la interfaz; en otras palabras, los controles de Visual Basic esperan que sucedan eventos particulares para llevar a cabo una función o proceso.

Elementos de una aplicación en Visual Basic.

Los objetos en Visual Basic constan de propiedades, eventos, procedimientos y métodos. El objeto de mayor jerarquía es la forma y a continuación se describe a cada uno de los elementos.

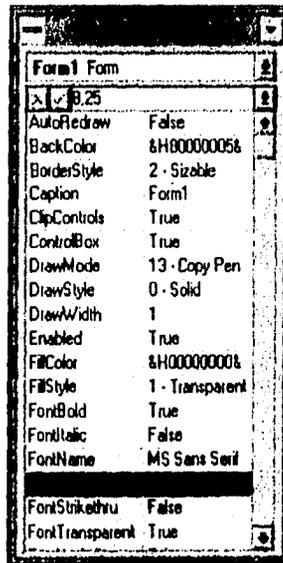
Forma. Ventana que se crea y adapta a las necesidades de la aplicación a ser creada.



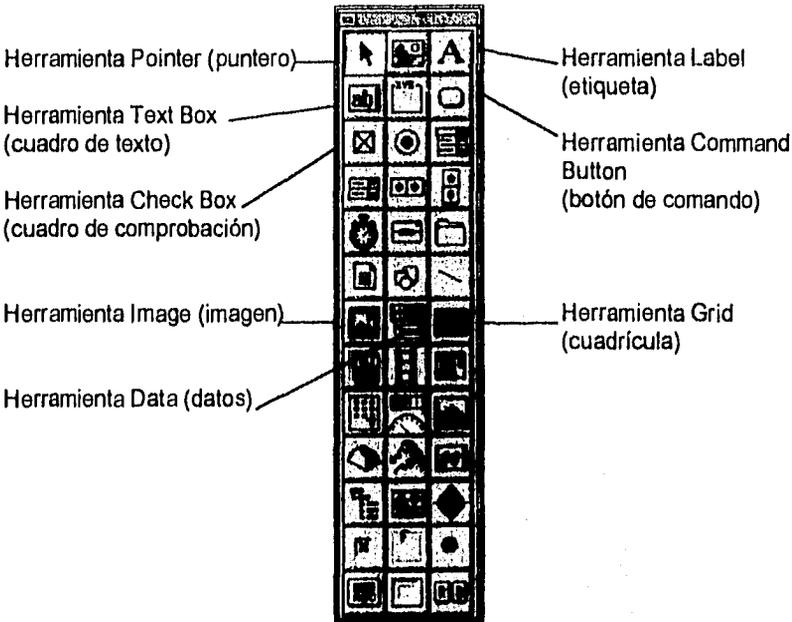
Propiedad. Característica o atributo de un objeto. Visual Basic define una serie de propiedades que se aplican solamente a cada objeto.

Definición. Es el valor de una propiedad. Se puede modificar la definición de la mayoría de las propiedades mientras se está construyendo una aplicación.

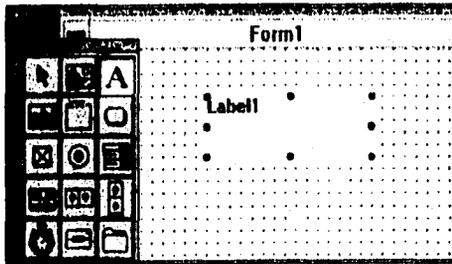
Un ejemplo de la ventana de propiedades de un objeto es la siguiente imagen:



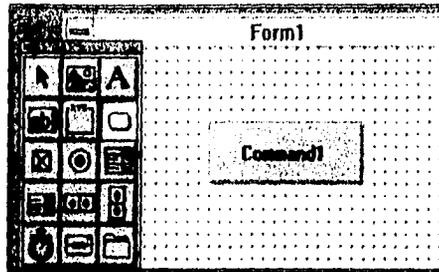
Control. Un término general utilizado para describir cualquier forma o elemento gráfico que se utilice dentro de una forma, incluyendo cajas de texto, cajas de listas, botones de comandos, cajas de imágenes, barras de desplazamiento e iconos, etc. Son los datos acoplados con una serie de rutinas o métodos que se utilizan exclusivamente para acceder y manipular el control.



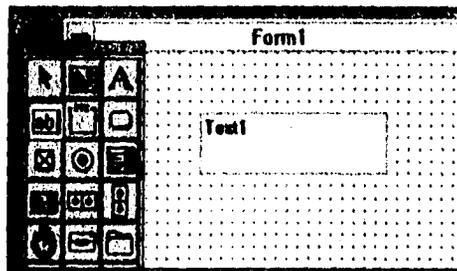
La caja de herramientas de Visual Basic.



La herramienta Label (etiqueta).



La herramienta *Command Button* (botón de comando).



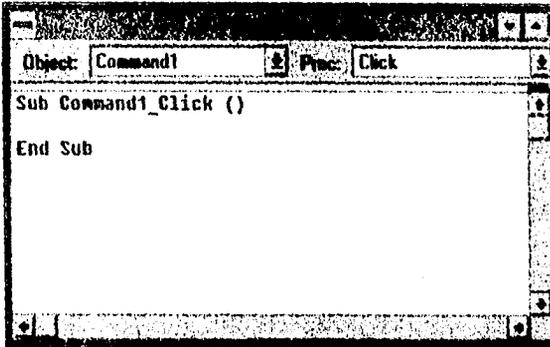
La herramienta *Text Box* (cuadro de texto).

Evento. La acción reconocida por un control de Visual Basic.

Método. Palabra de código de Visual Basic que es similar a una función o instrucción, pero que siempre actúa sobre un control en particular; para cada control Visual Basic predefine una serie de métodos que se pueden utilizar.

Procedimiento. Término que se refiere tanto a procedimientos *Sub* y *Function*. Es sencillamente una secuencia de instrucciones Visual Basic que son ejecutadas en grupo durante toda la corrida.

Existen dos tipos de procedimientos: *procedimientos de evento* que se limitan a las formas y los controles; y *procedimientos generales* que se utilizan durante toda la aplicación y pueden ser requeridos por procedimientos de eventos.



Ventana de procedimientos.

Proyecto. El conjunto de todos los archivos que forman la aplicación desarrollada.

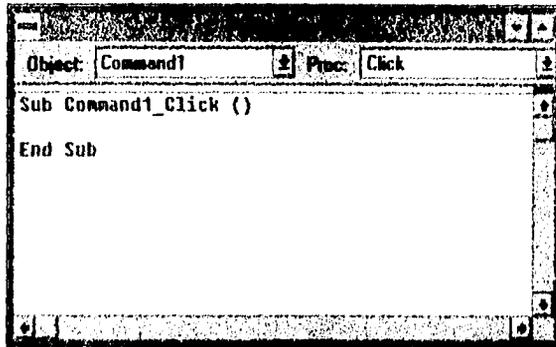
Una forma posee muchas propiedades que pueden afectar su apariencia y comportamiento cuando ésta aparece en pantalla; las propiedades pueden establecerse usando tanto la ventana *Properties* a medida que se realiza el diseño de una aplicación, así como escribiendo el código del programa, para que establezca las propiedades cuando el programa se esté ejecutando; algunas de las más comunes son *AutoRedraw*, *BackColor*, *BorderStyle*, *Caption*, *Enabled*, *FontName*, *Name*, *Picture*, *Visible*, *WindowState*, etc.

BackColor (color de fondo). Determina el color de fondo de la ventana.

Caption (título). Contiene el texto de la barra del título de la ventana.

Enabled (activado). Se puede establecer como *True* para activarla o *False* para desactivarla si se desea que la forma no reciba ningún evento del ratón o del teclado.

Existen dos tipos de procedimientos: *procedimientos de evento* que se limitan a las formas y los controles; y *procedimientos generales* que se utilizan durante toda la aplicación y pueden ser requeridos por procedimientos de eventos.



Ventana de procedimientos.

Proyecto. El conjunto de todos los archivos que forman la aplicación desarrollada.

Una forma posee muchas propiedades que pueden afectar su apariencia y comportamiento cuando ésta aparece en pantalla; las propiedades pueden establecerse usando tanto la ventana *Properties* a medida que se realiza el diseño de una aplicación, así como escribiendo el código del programa, para que establezca las propiedades cuando el programa se esté ejecutando; algunas de las más comunes son *AutoRedraw*, *BackColor*, *BorderStyle*, *Caption*, *Enabled*, *FontName*, *Name*, *Picture*, *Visible*, *WindowState*, etc.

BackColor (color de fondo). Determina el color de fondo de la ventana.

Caption (título). Contiene el texto de la barra del título de la ventana.

Enabled (activado). Se puede establecer como *True* para activarla o *False* para desactivarla si se desea que la forma no reciba ningún evento del ratón o del teclado.

FontName (nombre de la fuente). Esta propiedad establece el nombre de la fuente que se utilizará para escribir en la forma.

Name (nombre). Define el nombre de la forma.

WindowState (estado de la ventana). Determina si la ventana aparecerá en estado normal, maximizada o minimizada a ejecutarse.

Los objetos como las formas constan también de eventos como:

Click (clic). Cuando el usuario hace clic en la forma, Visual Basic llama al procedimiento *Form_Click*.

Dblclick (doble clic). Este evento se recibe si el usuario hace clic dos veces de manera rápida.

Load (cargar). Este evento ocurre automáticamente cuando se carga una forma, por ejemplo cuando se arranca una aplicación y una forma es llamada.

Algunos de los métodos que se aplican a los objetos son:

Cls (borrar). Este método borra de la forma todos los gráficos y texto.

Print (imprimir). Sirve para imprimir texto en la forma.

SetFocus (da el foco). El método *SetFocus* sitúa el cursor en el objeto, le da el foco de atención.

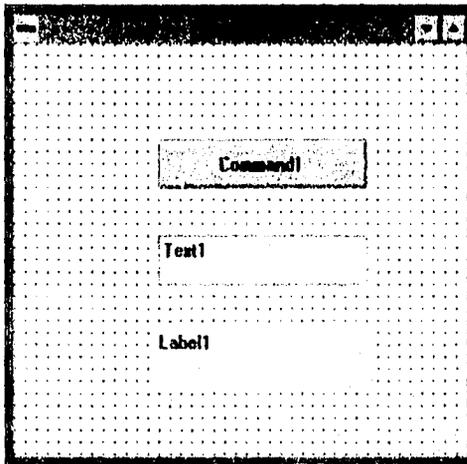
En Visual Basic existen archivos de aplicación que contienen únicamente código y tienen extensión BAS. Estos son llamados módulos de código y en ellos se hacen declaraciones globales de variables y procedimientos que pueden ser compartidos por todo el programa.

Las *formas* junto con su código asociado crean archivos con extensión FRM. Se pueden crear aplicaciones que contengan varias formas que estén contenidas en una mayor denominada FORMA.MDI. El *proyecto* que es el conjunto de todos los archivos crea un archivo cuya extensión es MAK.

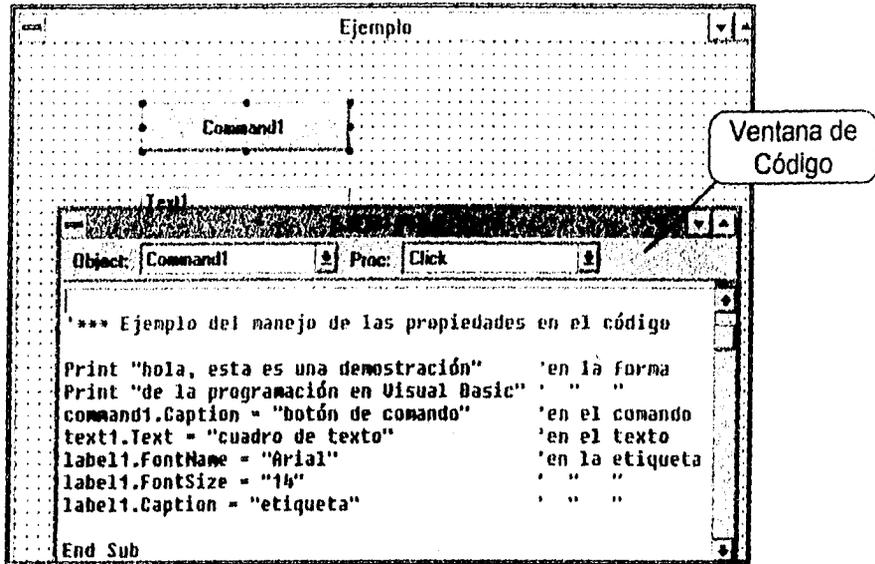
Ejemplo de una aplicación.

El proyecto "Ejemplo", es una aplicación que le permite conocer los objetos básicos de Visual Basic, de manera que pueda familiarizarse con éstas herramientas, así como con su programación.

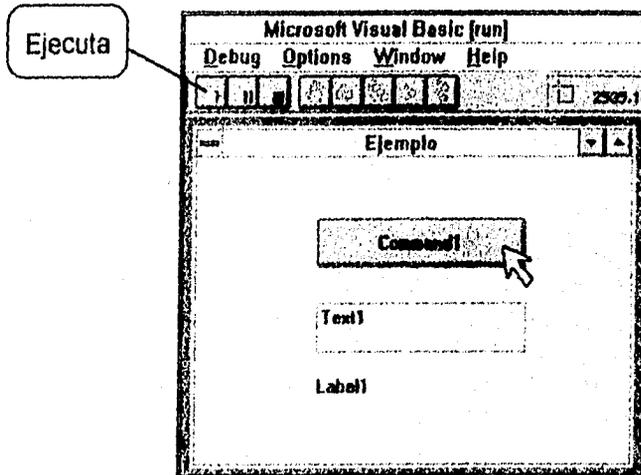
En la *forma* del proyecto "Ejemplo" sólo se modificó la propiedad *Caption* y se hace durante el diseño usando la ventana *Properties*. A la forma se le añadió un botón de comando, un cuadro de texto y una etiqueta.



Para establecer las propiedades de los controles que se tienen y su procedimiento de evento, por medio de programación, se utiliza la ventana de código para escribirlo. Para abrir la ventana de código, es necesario hacer doble clic con el mouse, sobre el control que quiera programar. En este caso, se hace sobre *Command1* y se teclea el código. Observe la siguiente figura.

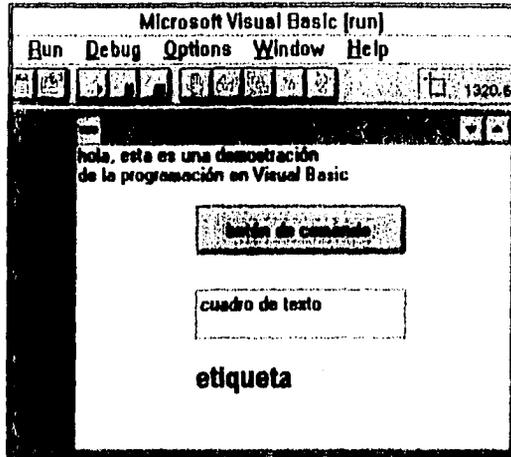


Al ejecutar la aplicación presionando el botón señalado en la siguiente ilustración, la ventana de la aplicación "Ejemplo" se ve así:



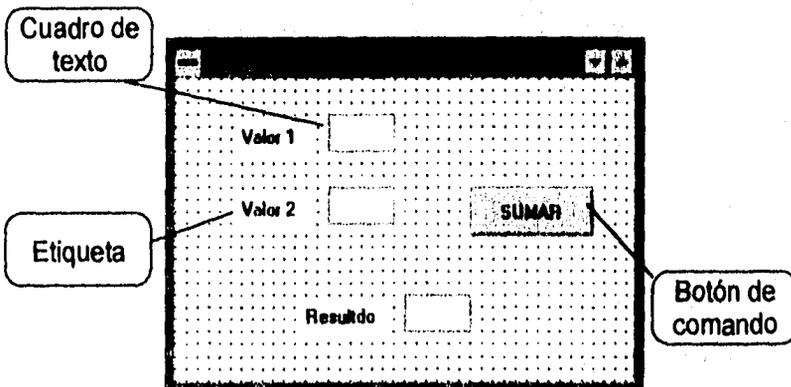
Ejecución de la aplicación "Ejemplo".

Si ahora se hace clic sobre *Command1* el resultado a esta acción es:

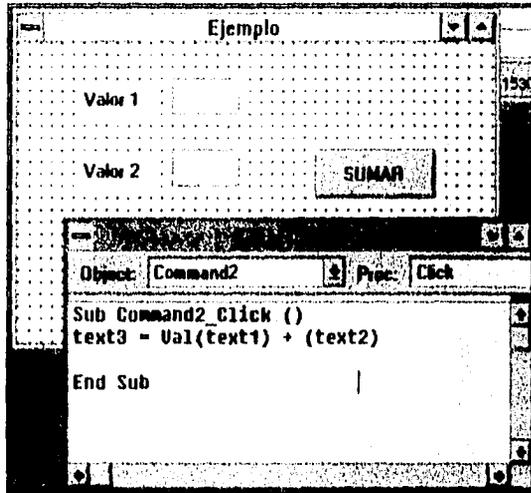


Resultado de la ejecución.

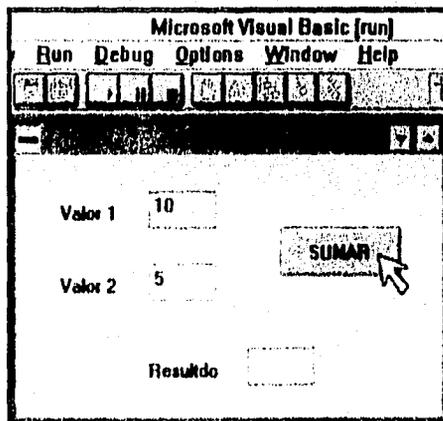
Se añade a la aplicación "Ejemplo" dos cuadros de texto, dos etiquetas y se modifican sus propiedades como puede verlo en la siguiente figura.



En la ventana de código se edita el procedimiento *Sumar_Click* (en el botón de comando *Sumar* y el evento "click"), su código se muestra a continuación.

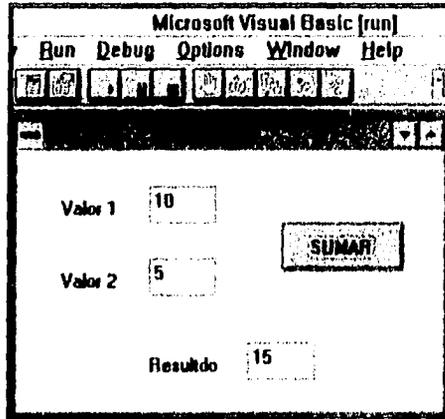


Al ejecutar la aplicación, se teclean los valores que se ven en los cuadros de texto. Para realizar una suma de ambos se presiona el botón *Sumar*.



Ejecución de la aplicación.

Puede darse cualquier número a "valor 1" y "valor 2". El resultado de la ejecución es el siguiente:



Resultado de la ejecución.

4.1.3 Acceso a bases de datos.

Visual Basic provee de una herramienta denominada *Data* (Datos), que permite crear un control de datos y a través de éste se tiene acceso a información específica de una base de datos.

El control *Data*, tiene diversas propiedades que definen la conexión entre la aplicación Visual Basic y la base de datos a la que quiere acceder.

La versión 3.0 de Visual Basic, permite intercambiar datos con los siguientes manejadores de bases de datos (DBMS): Microsoft Access, Microsoft Fox Pro, Borland Dbase y Borland Paradox.

4.1.3.1 Dbase III Plus.

Dbase III Plus, es un manejador de bases de datos relacionales, elaborado por la compañía ASHTON-TATE.

Características.

- Está escrito en lenguaje C.
- Posee características de programación estructurada para el desarrollo de programas
- Se puede instalar en cualquier PC con sistema operativo MS-DOS versión 2.0 o superior.
- Contiene funciones que pueden realizar varios procesos a la vez.

Requerimientos.

Se requieren 384 Kbytes de memoria RAM como mínimo en la microcomputadora, teniéndose las siguientes ventajas:

- El número de registros máximos es 1 billón.
- El número máximo de campos por registro es 128.
- El tamaño de los campos es, hasta de 254 caracteres.
- Para campos tipo fecha se tienen 8 caracteres.
- Se pueden abrir hasta 10 bases de datos al mismo tiempo.
- Se pueden abrir máximo 7 archivos indexados por cada base de datos.

4.2 Control de la aplicación

La manera en la cual se le ofrece a los usuarios la sensación de controlar una aplicación en Windows es hacer que ellos la manipulen directamente, como al abrir y cerrar una aplicación; el transitar dentro de una aplicación; el escoger objetos; el llevar a cabo un evento de una serie de controles, etc. Los usuarios deben esperar que sus acciones físicas obtengan resultados visibles.

Al mismo tiempo, Visual Basic da los recursos para crear aplicaciones que presenten un ambiente visualmente dinámico para que los usuarios completen sus tareas, en donde puedan seguir rutas no lineales proporcionadas por objetos. Los usuarios pueden interactuar de manera directa con la pantalla, escogiendo objetos y tareas que a ellos les interesa explorar con sólo señalarlos.

4.3 Codificación

La fase de implementación del desarrollo del sistema tiene que ver con la traducción de las especificaciones de diseño a código fuente. El objetivo principal de la implementación es el escribir código fuente y la documentación interna de modo que la concordancia del código con sus especificaciones sea fácil de verificar, y que se faciliten la depuración, pruebas y modificaciones. Este objetivo puede alcanzarse haciendo el código fuente tan claro y sencillo como sea posible. Sencillez, claridad y elegancia son los sellos de los buenos programas; oscuridad, ingeniosidad, y complejidad son indicaciones de un diseño inadecuado y un pensamiento mal orientado.

Una característica importante, es la *modularidad*. Se ha establecido que la "modularidad, es un atributo del software que permite a un programa ser independiente". El software monolítico (es decir, un gran programa compuesto de un único módulo) no puede ser fácilmente abarcado por un lector. El número de caminos de control, expansión

de referencias, número de variables y complejidad global, podría hacer imposible el comprenderlo bien. La modularidad se ha convertido en un enfoque acertado. Un diseño modular, reduce la complejidad, facilita los cambios (un aspecto crítico del mantenimiento) y da como resultado una implementación más fácil, posibilitando el desarrollo paralelo de diferentes partes de un sistema [PRE93].

El estilo es un atributo del código que puede determinar la inteligibilidad de un programa. Los elementos de estilo incluyen la documentación interna, los métodos de declaración de datos, los procedimientos de construcción de sentencias y la codificación de la E/S. En todos los casos, la claridad y la sencillez son características clave. Una derivación del estilo es la eficiencia obtenida en tiempo de ejecución y en memoria.

Estructuras de control.

En un nivel fundamental, todos los lenguajes de programación modernos, en este caso Visual Basic, permiten al programador representar secuencias, condiciones y repeticiones -las construcciones lógicas de la programación estructurada-. La *secuencia* implementa los pasos de procesamiento esenciales de la especificación de cualquier algoritmo, la *condición* da la posibilidad de seleccionar un procedimiento basado en alguna concurrencia lógica y la *repetición* proporciona iteración (Fig. 4-1) [PRE93].

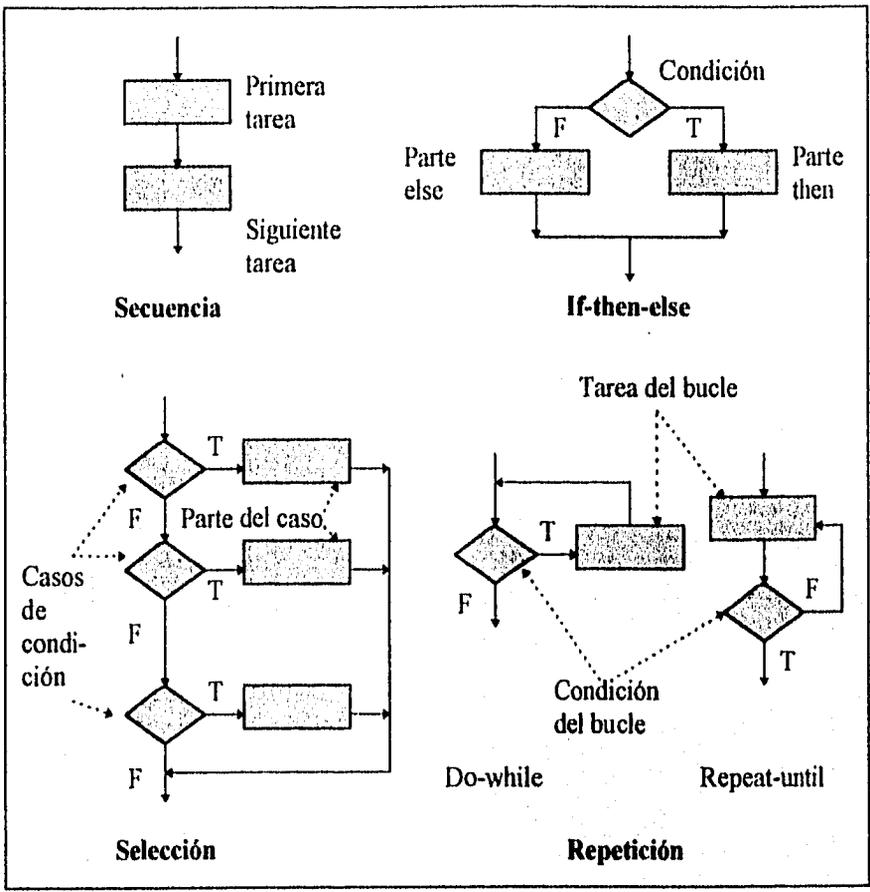


Figura 4-1. Construcciones lógicas estructuradas.

4.4 Diagramas del "SIAV II" y su codificación

Los siguientes diagramas, representan el flujo de la información, a través de las formas del sistema para su posterior implementación.

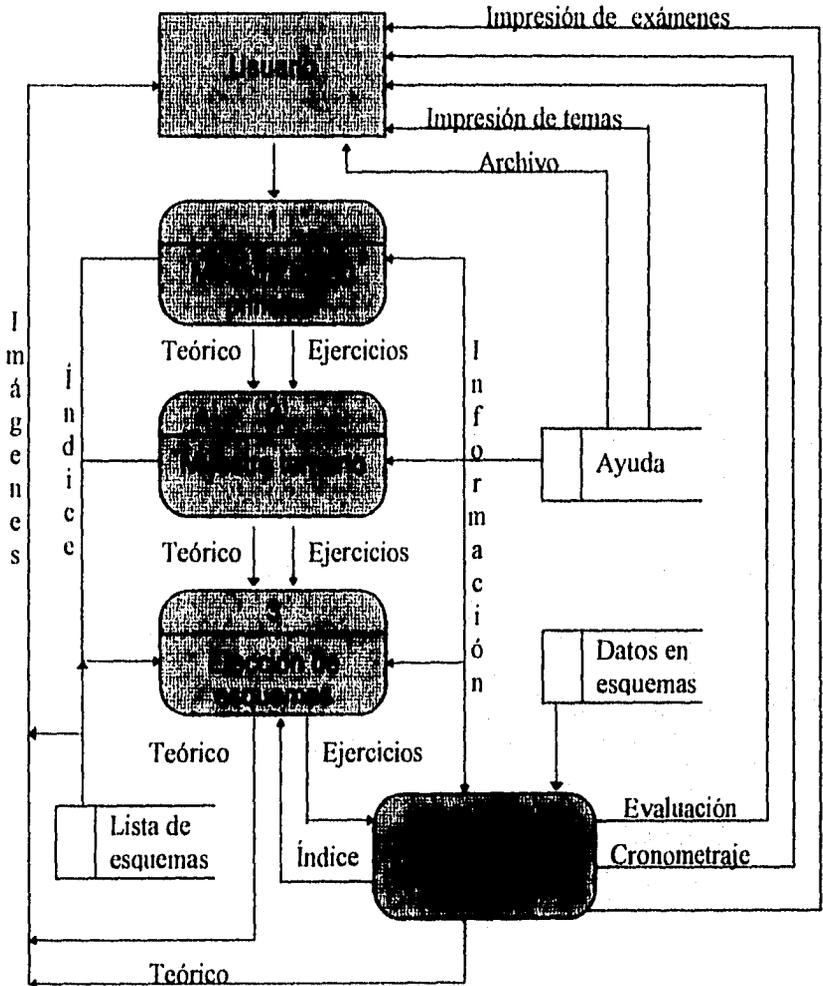


Fig. 4-2. Diagrama de flujo de datos (DFD), del "SIAV II".

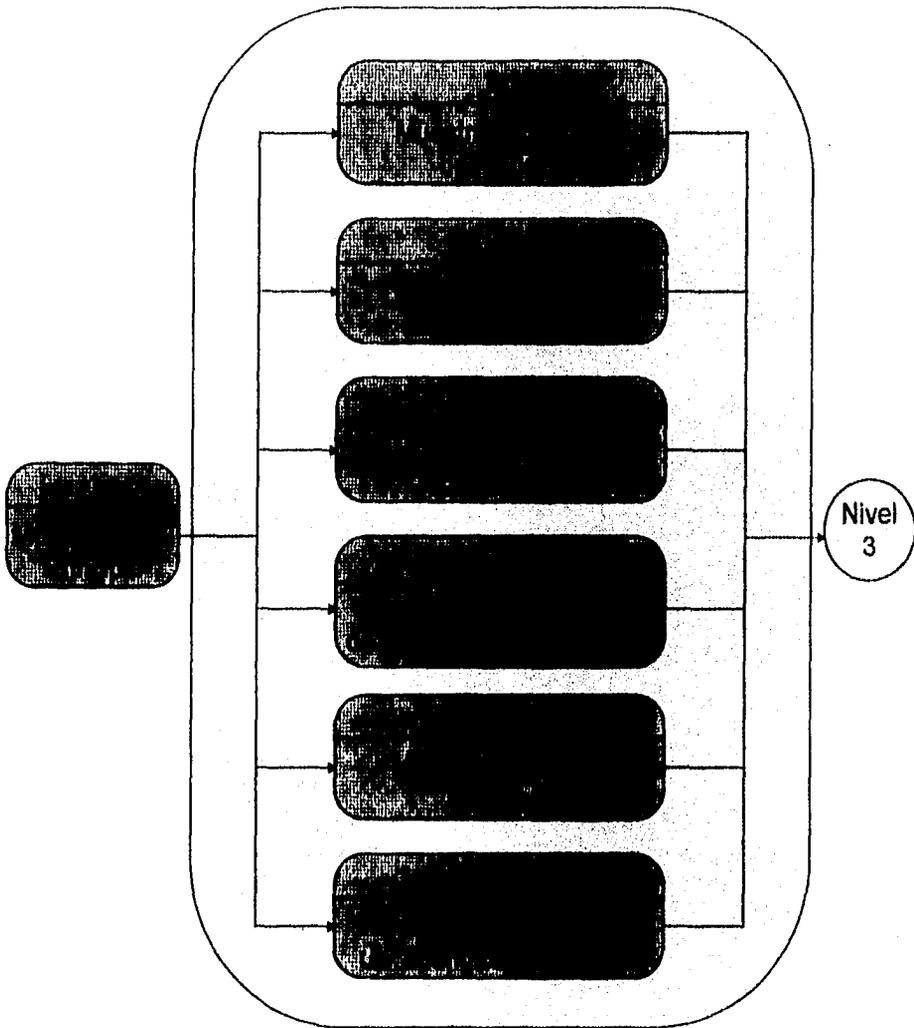


Fig. 4-3. DFD detallado de la fig. 4-2 de "Muestra temarlo" (nivel 2), del "SIIV II".

Con el apoyo en los diagramas siguientes, se dará una visión de la traducción del diseño al código, en el lenguaje Visual Basic, del Sistema Interactivo de Anatomía Veterinaria II.

En la figura 4-4 (DFD de burbuja) se pueden ver los módulos que integran al sistema y para el propósito que se busca, se usará la ruta del módulo de ejercicios, hasta llegar a la forma *Tipos de Examen*.

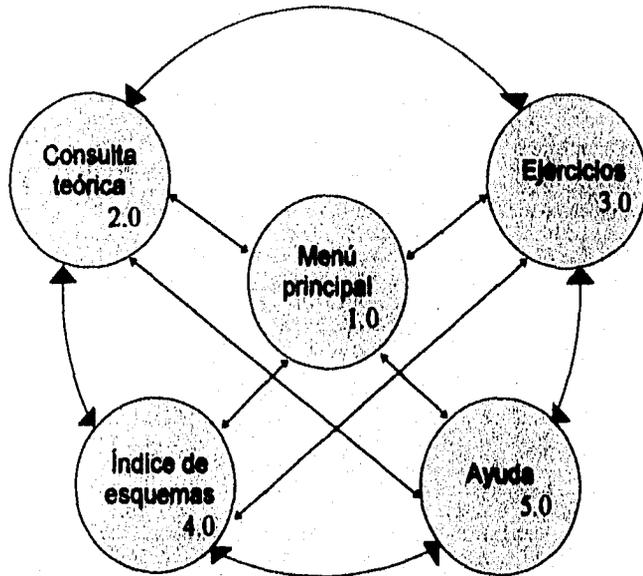


Figura 4-4. Navegación del "SIIV II" (DFD de burbuja).

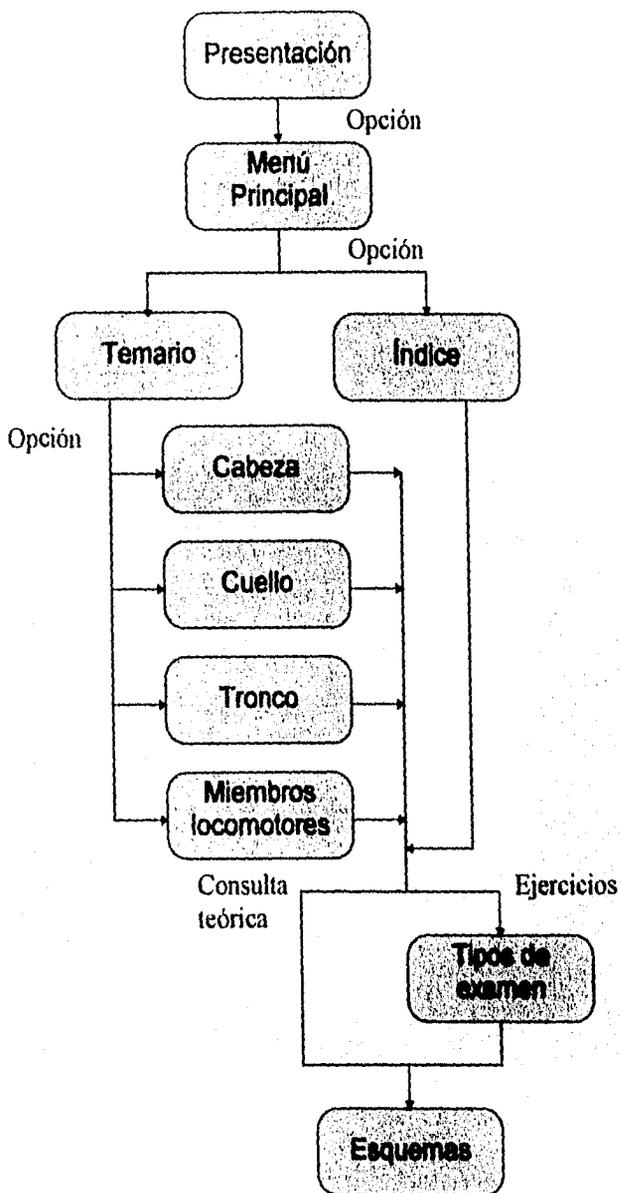


Figura 4-5. Diagrama de formas del "SIIV II" (DFD).

Dentro de los módulos del programa hay objetos que constan de propiedades, eventos, procedimientos y métodos; en el caso de la forma, cuando se hace "click" en ella, se está utilizando el evento *Click* de la forma para llamar al procedimiento *Form_Click*, donde se escribe el código utilizando estructuras de control. Las formas (los objetos de mayor jerarquía) que integran al programa se pueden observar en la figura anterior (4-5 diagrama de formas).

En la figura 4-6 vemos la forma de *Tipos de Examen*, cuyo nombre aparece en la ventana de propiedades.

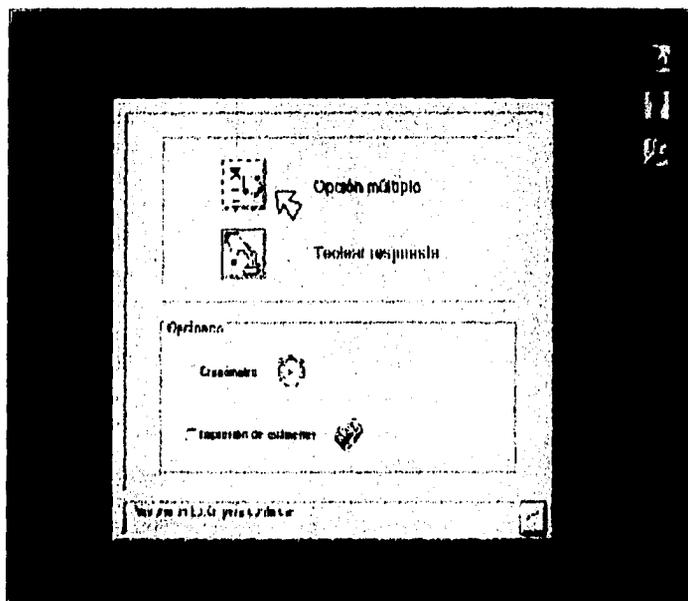


Figura 4-6. Forma de Tipos de Examen (EXAM).

La figura muestra la ventana de propiedades de la forma *Exam*.

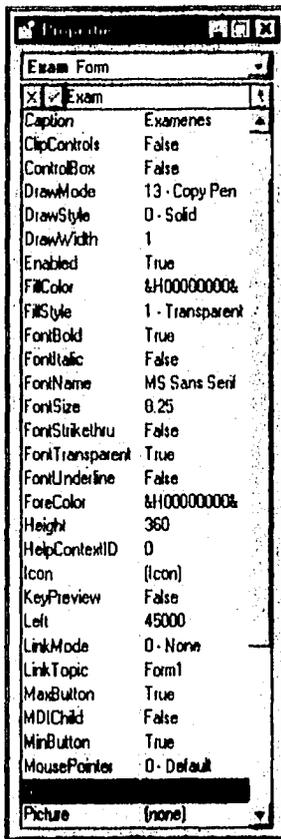


Figura 4-7. Propiedades de la forma EXAM.

En caso de seleccionar el examen de *Opción múltiple*, se utiliza el objeto con su código, que se describe a continuación.

Objeto: OPCMUL**Procedimiento: Click**

```

Sub OPCMUL_Click ()
TIPOEXAMEN = "OPCMUL"           'variable global
If ruta = "INDESQ" Then        'Si la ruta es índice de esquemas
  Select Case Ind               'selecciona el esquema por el indice
  Case 1                        'Ind es una variable global
    T1_1_01.Show               'muestra la forma con el esquema
  Case 3
    T1_1_04.Show
  Case 0
    T1_1_05.Show
  Case 4
    T1_1_06.Show
    .
    .
    .
  Case 18
    T2_1_01.Show
  Case 19
    T2_1_02.Show
  Case 17
    T2_2_01.Show
  Case 11
    T2_2_02.Show
  End Select

Else                             'Si la ruta no es índice de esquemas

Select Case E                   'selecciona el esquema por nombre
  Case "T1_1_01"                'E es una variable global
    T1_1_01.Show               'muestra la forma con el esquema
  Case "T1_1_04"
    T1_1_04.Show
  Case "T1_1_05"
    T1_1_05.Show
    .
    .
    .

```

```
Case "T2_1_02"  
    T2_1_02.Show  
Case "T2_2_01"  
    T2_2_01.Show  
Case "T2_2_02"  
    T2_2_02.Show  
End Select  
End If
```

'rutina para cronometrar el tiempo del examen

```
calificar = "no"  
If Check3D2.Value = -1 Then 'inicia cronometraje si se activó el control  
    horainicio = Now  
    txtinicio.Text = Format(Now, "hh:mm:ss")  
    txtinicio.Visible = True  
    label1.Visible = True  
    txtfin.Text = ""  
    txtlapsus.Text = ""  
End If
```

```
exam.Hide          'oculta la forma
```

```
End Sub
```

Objeto: OPCMUL

Procedimiento: MouseDown

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

```
OPCMUL.SetFocus          'el método SetFocus sitúa el cursor en el  
objeto, le da el foco de atención
```

```
End Sub
```

Objeto: OPCMUL**Procedimiento: MouseMove**

Sub OPCMUL_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)

```
SCREEN.MousePointer = 0 'cambia el cursor a flecha
lblico.Caption = " Haz click sobre la lista, elige una opción y haz click
sobre el cuadro correcto" 'manda el mensaje a una etiqueta
```

End Sub

La siguiente forma "Esquema", varía de acuerdo a la selección que se haga. Suponiendo que se decidió trabajar con el tema "Cuello" (forma *MNUCUELLO*) y que el esquema elegido es "Disección cervical" (nombre: *T1_02_06*), la forma será:

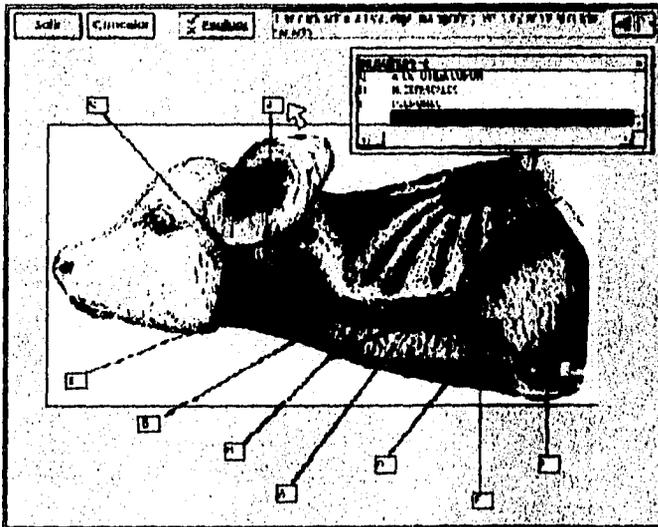


Figura 4-8. Forma T1_02_06 (Esquema de disección cervical).


```

grid1.Row = J           'llena la tabla con la base de datos
grid1.Col = 0          'empieza en el renglón J, columna 0
grid1.Text = DATA1.Recordset("INCISO")
grid1.Col = 1
grid1.Text = DATA1.Recordset("NOMBRE")
DATA1.Recordset.MoveNext 'va al siguiente registro
J = J + 1              'suma 1 a la posición del renglón J
Loop
grid1.FixedRows = 1

```

```

'===== Trae el nombre del esquema de la lista de esquemas LSTESQ
(DATA2)
DATA1.Recordset.FindFirst      "CLAVE="      +      ""      +
DATA2.Recordset("CLAVE") + ""
Iblico.Caption = DATA2.Recordset("N_ESQUEMA")

```

***** Estado Teórico *****

```

If ESTADO = "TEORICO" Then      'Si estado es teórico entonces
    EVALUAR.Visible = False      'oculta botón
    line5.Visible = False        'oculta líneas
    line4.Visible = False        ' " "
    Iblico.Visible = True        'etiqueta mensajes visible
    Cancelar.Visible = False     'oculta botón
    For index = 0 To grid1.Rows - 2 'para el índice 0 hasta el total de
        'renglones del grid, que contiene los nombres
        OM(index).Visible = False 'oculta cuadros de opción múltiple
        CR(index).Visible = False 'oculta cuadros respuesta completa
        XOP(index).Visible = False 'oculta cruz o paloma
        Nom(index).Visible = True  'muestra nombres en los esquemas
    Next index
    index = 0

```

```

'===== Pone los nombres a las partes señaladas en los esquemas
DATA1.Recordset.MoveFirst
Do While Not DATA1.Recordset.EOF
    Nom(index).Caption = DATA1.Recordset("NOMBRE")
    'el nombre es igual al registro "nombre" de la base de datos

```

```

index = index + 1
DATA1.Recordset.MoveNext      'va al siguiente registro
Loop

```

```
End If 'fin estado teórico
```

```
***** Estado Práctico *****
```

```

If ESTADO = "PRACTICO" Then 'Si estado es práctico entonces
    EVALUAR.Visible = True 'aparece el botón evaluar y la etiqueta
    lblico.Visible = True
    Command3D2.Picture = LoadPicture("C:\YAZMIN\ferinac.ico")
                        'carga la imagen de la flecha inactiva
    For index = 0 To grid1.Rows - 2 'para el índice 0 hasta el total de
                                renglones del grid, que contiene los nombres
        Nom(index).Visible = False 'oculta los nombres
    Next index

```

```
'Examen de opción múltiple
```

```

If TIPOEXAMEN = "OPCMUL" Then 'Si el examen es opción múltiple
    VsElastic2.Visible = True 'muestra la tabla de respuestas
    For index = 0 To grid1.Rows - 2
        OM(index).Visible = True 'muestra cuadros de opción m.
        OM(index).Text = "" 'limpia cuadros de texto
        CR(index).Visible = False 'oculta cuadros de resp. completa
        XOP(index).Visible = False
    Next index
    OM(0).SetFocus 'el método set focus pone el cursor en el
                  cuadro de índice 0

```

```
End If 'fin examen OPCMUL
```

```
'Examen de respuesta completa
```

```

If TIPOEXAMEN = "COMRESP" Then 'Si es completa respuesta
    VsElastic2.Visible = False 'oculta tabla de respuestas
    For index = 0 To grid1.Rows - 2
        OM(index).Visible = False
        CR(index).Visible = True 'muestra cuadros de resp. completa
        CR(index).Text = "" 'limpia cuadros de texto

```

```

        XOP(index).Visible = False
    Next index
    CR(0).SetFocus      'el método set focus pone el cursor en el
                        cuadro de índice 0
End If      'fin examen COMRESP

End If 'fin estado práctico

```

***** Impresión *****

```

If IMPR = "IMPRIME" Then      'Si enviamos a impresión
    VsElastic6.Visible = False 'oculta objetos en la forma
    VsElastic2.Visible = False
    image1.BorderStyle = 0    'quita margen y da fondo blanco
    BackColor = &HFFFFFF
    label1.Visible = True    'muestra etiqueta con el nombre del esquema
    label1.Caption = lblico.Caption
    resp = MsgBox("Impresión de esquemas, desea continuar??", 1 + 32
+ 256, "Impresión")          'envía mensaje de confirmación
    If resp = 1 Then          'Si acepta entonces
        PrintForm            'imprime
        Unload T1_2_06       'descarga la forma
    Else                       'de lo contrario
        Unload T1_2_06       'descarga la forma
    End If
Exam.Show                    'muestra tipos de examen
End If
End Sub

```

Objeto: Form Procedimiento: DragDrop

```

Sub Form_DragDrop (Source As Control, X As Single, Y As Single)
Source.Move X, Y            'permite mover el grid sobre la forma
End Sub

```

Objeto: Form **Procedimiento: Load**

```
Sub Form_Load ()
'Selecciona la base de datos
DATA1.RecordSource = "SELECT * FROM CUELLO where
CUELLO.CLAVE='T1_2_06'"
DATA2.RecordSource = "SELECT * FROM LSTESQ where
LSTESQ.CLAVE='T1_2_06'"
End Sub
```

Objeto: Grid1 **Procedimiento: Click**

```
Sub Grid1_Click ()
SCREEN.MousePointer = 5 'cambia el cursor a flechas
End Sub
```

Objeto: Grid1 **Procedimiento: MouseMove**

```
Sub Grid1_MouseMove (Button As Integer, Shift As Integer, X As
Single, Y As Single)
Iblico.FontSize = 9.75
Iblico.Caption = " Haz click sobre la lista, elige una opción y haz click
sobre el cuadro correcto"
'manda mensaje al mover el mouse sobre el grid
End Sub
```

Objeto: Grid1 **Procedimiento: MouseUp**

```
Sub Grid1_MouseUp (Button As Integer, Shift As Integer, X As Single,
Y As Single)
grid1.SetFocus
End Sub
```

Objeto: Evaluar Procedimiento: Click

```

Sub EVALUAR_Click ()
Dim index As Integer
Dim calif As Variant
Dim callot As Double

Iblico.Visible = False           'muestra mensajes
mh3dlabel4.Visible = True       ' " " letrero "calificación
panel3d1.Visible = True        ' " " calificación
If Exam.Check3D2.Value = -1 Then 'Si se activó cronometraje
    horafin = Now                'toma hora de fin
    lapsus = horafin - horainicio 'calcula tiempo del examen
    Exam.txtfin.Text = Format(horafin, "hh:mm:ss")
    Exam.txtlapsus.Text = Format(lapsus, "hh:mm:ss")
    Exam.txtfin.Visible = True   'muestra etiquetas con tiempos
    Exam.Label2.Visible = True
    Exam.txtlapsus.Visible = True
    Exam.Label3.Visible = True
End If

For index = 0 To grid1.Rows - 2
    XOP(index).Visible = False   'oculta palomas y cruces
Next index

'Examen de opción múltiple
If TIPOEXAMEN = "OPCMUL" Then   'Si el examen es opción múltiple
    VsElastic2.Visible = False   'oculta la lista de respuestas
    index = -1
    calif = 0
    DATA1.Recordset.MoveFirst   'se mueve al primer registro
    Do While Not DATA1.Recordset.EOF 'mientras no sea fin de data1
        index = index + 1
    If OM(index).Text = DATA1.Recordset("INCISO") Then
        'compara respuesta con el registro de la base de datos
        calif = calif + 1         'suma 1 punto a la calificación
        XOP(index).Visible = True 'si es correcto, aparece paloma
        XOP(index).Picture = LoadPicture("C:\YAZMIN\paloma.ico")
    Else

```

Objeto: Evaluar Procedimiento: Click

```

Sub EVALUAR_Click ()
Dim index As Integer
Dim calif As Variant
Dim caltot As Double

lblico.Visible = False           'muestra mensajes
mh3dlabel4.Visible = True        ' " " letrero "calificación
panel3d1.Visible = True          ' " " calificación
If Exam.Check3D2.Value = -1 Then 'Si se activó cronometraje
    horafin = Now                 'toma hora de fin
    lapsus = horafin - horainicio 'calcula tiempo del examen
    Exam.txtfin.Text = Format(horafin, "hh:mm:ss")
    Exam.txtlapsus.Text = Format(lapsus, "hh:mm:ss")
    Exam.txtfin.Visible = True    'muestra etiquetas con tiempos
    Exam.Label2.Visible = True
    Exam.txtlapsus.Visible = True
    Exam.Label3.Visible = True
End If

For index = 0 To grid1.Rows - 2
    XOP(index).Visible = False    'oculta palomas y cruces
Next index

'Examen de opción múltiple
If TIPOEXAMEN = "OPCMUL" Then    'Si el examen es opción múltiple
    VsElastic2.Visible = False    'oculta la lista de respuestas
    index = -1
    calif = 0
    DATA1.Recordset.MoveFirst    'se mueve al primer registro
    Do While Not DATA1.Recordset.EOF 'mientras no sea fin de data1
        index = index + 1
    If OM(index).Text = DATA1.Recordset("INCISO") Then
        'compara respuesta con el registro de la base de datos
        calif = calif + 1          'suma 1 punto a la calificación
        XOP(index).Visible = True 'si es correcto, aparece paloma
        XOP(index).Picture = LoadPicture("C:\YAZMIN\paloma.ico")
    Else

```

```

        XOP(index).Visible = True      'si es incorrecto, aparece cruz
        XOP(index).Picture = LoadPicture("C:\YAZMIN\X.ico")
    End If
    DATA1.Recordset.MoveNext          'se mueve al siguiente registro
Loop                                  'ciclo
    caltot = (calif / (grid1.Rows - 1)) * 10      'calcula calificación
    panel3d1.Caption = Format$(caltot, "#,##0.00") 'muestra calificación
End If                                  'fin OPCMUL

'Examen de respuesta completa
If TIPOEXAMEN = "COMRESP" Then
    index = -1
    calif = 0
    DATA1.Recordset.MoveFirst          'se mueve al primer registro
    Do While Not DATA1.Recordset.EOF
        index = index + 1
        If CR(index).Text = DATA1.Recordset("NOMBRE") Then
            'compara respuesta
            calif = calif + 1          'suma 1 punto a la calificación
            XOP(index).Visible = True  'pone cruz o paloma
            XOP(index).Picture = LoadPicture("C:\YAZMIN\paloma.ico")
        Else
            XOP(index).Visible = True
            XOP(index).Picture = LoadPicture("C:\YAZMIN\X.ico")
        End If
        DATA1.Recordset.MoveNext      'se mueve al siguiente registro
    Loop
    caltot = (calif / (grid1.Rows - 1)) * 10
    panel3d1.Caption = Format$(caltot, "#,##0.00") 'muestra calificación
End If                                  'fin COMRESP
End Sub

```

Objeto: Cancelar **Procedimiento: Click**

```

Sub Cancelar_Click ()
SCREEN.MousePointer = 0      'cambia cursor
If tipoexam = OPCMUL Then   'borra la última respuesta
    OM(n).Text = ""
else
    CR(n).Text = ""
End If
End Sub

```

Objeto: OM **Procedimiento: Click**

```

Sub OM_Click (index As Integer)
grid1.Col = 0                'va a la primer columna del grid
OM(index) = grid1.Text      'toma el texto del grid
SCREEN.MousePointer = 0     'cambia el cursor a flecha
n = index                   'variable índice
End Sub

```

Objeto: Salir **Procedimiento: Click**

```

Sub SALIR_Click ()
Unload T1_2_06              'descarga el esquema
mnuPrinc.Show              'muestra menú principal
End Sub

```

Si ejecutamos el programa y contestamos los incisos el resultado será el siguiente.

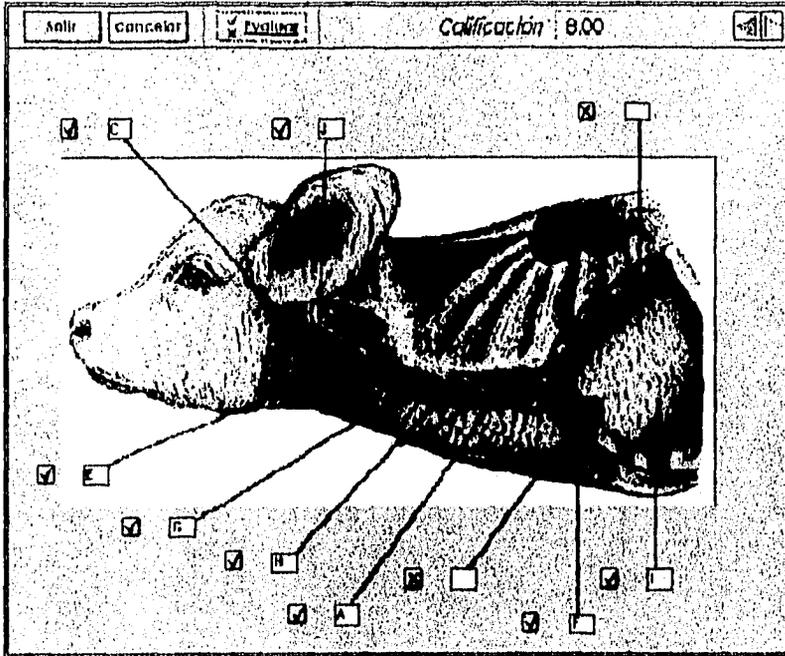


Figura 4-9. Evaluación del examen de opción múltiple, forma T1_2_06.

Capítulo 5

Pruebas del sistema

Durante la última fase del proceso de ingeniería del software, se prueba éste para encontrar el mayor número posible de errores antes de que sea puesto en circulación, se prepara para su lanzamiento y se mantiene a lo largo de toda su vida útil.

5.1 Técnicas de prueba del software.

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

El diseño de casos de prueba se centra en un conjunto de técnicas para la creación de "casos prueba" que satisfagan los objetivos globales de la misma [PRE93].

5.1.1 Objetivos de la prueba.

El objetivo es diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo. Como ventaja secundaria la prueba demuestra hasta qué punto las funciones del software satisfacen las especificaciones y parecen alcanzarse los requerimientos de rendimiento.

Al llevarse a cabo la prueba se evalúan los resultados; es decir, se comparan los resultados de la prueba con los esperados. Cuando se descubren datos erróneos, esto implica que hay un error, comienza la depuración. El proceso de depuración es una consecuencia impredecible de la prueba.

Al ser recopilados y evaluados los resultados de la prueba comienza a vislumbrarse una medida cualitativa de la calidad y la fiabilidad del software.

5.1.2 Diseño de casos de prueba.

Cualquier producto de ingeniería (y muchas otras cosas) puede ser probado 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"; es decir, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.

El primer enfoque de prueba se denomina *prueba de la caja negra* y la segunda *prueba de la caja blanca*.

Los atributos de la prueba de la caja blanca así como de la caja negra se pueden combinar para llegar a una aproximación que valide la interfaz del software y asegure selectivamente que el funcionamiento interno del software es correcto [PRE93].

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 la

caja blanca, el ingeniero del software puede derivar casos de prueba que:

- Garanticen que se ejercitan por lo menos una vez todos los caminos independientes de cada módulo.
- Se ejercitan todas las decisiones lógicas en sus opciones, verdadera y falsa.
- Se ejecutan todos los bucles en sus límites y con sus límites operacionales.
- Se ejercitan las estructuras de datos internas para asegurar su validez.

La *prueba del camino básico*, una técnica de la 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 condiciones y del flujo de datos* ejercitan más aún la lógica del programa, y la *prueba de bucles* complementa a otras técnicas de la caja blanca, proporcionando un procedimiento para ejercitar bucles de distintos grados de complejidad.

Prueba de la Caja Negra.

Los métodos de prueba de la caja negra se centran en los requerimientos funcionales del software, es decir, 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 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 de terminación.

La prueba de la caja negra intencionadamente ignora la estructura de control, concentra su atención en el dominio de la información.

Matriz de casos de prueba.

Los casos de prueba deberán simular fielmente los datos reales que el programa pretende procesar. Deberá crearse un conjunto de casos prueba para probar al programa. La matriz de casos de prueba será una herramienta para ello.

La matriz de casos de prueba contiene cuatro secciones:

1. Objetivo de la prueba.
2. Resultados esperados.
3. Casos de prueba.
4. Resultados reales.

Los casos deberán formalizarse, registrarse y documentarse en una matriz de casos prueba similar a la que se muestra en la siguiente figura.

Objetivo de la prueba	Resultados esperados					Diseño de casos de prueba	Resultados reales
	Muestra ayuda	Cambia ruta	Exhibe mensaje	Cambia pantalla	Selección		

5.2 Estrategias de prueba

Una estrategia de prueba de software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que llevan a una construcción correcta del software [PRE93].

Una estrategia para el software debe acomodar pruebas de bajo nivel que verifiquen que cada pequeño segmento de código fuente ha sido implementado correctamente, así como pruebas de alto nivel que muestren la validez de las principales funciones del sistema frente a los requerimientos del cliente.

5.2.1 Verificación y validación.

La prueba del software a menudo se referencia como verificación y validación (V&V). La *verificación* se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica. La *validación* se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requerimientos del cliente.

Para conseguir el objetivo, descubrir los errores, se planifican y ejecutan una serie de pasos [PRE93]:

- Prueba de Unidad.
- Prueba de Integración.
- Prueba de Validación.
- Prueba del Sistema.

Prueba de Unidad.

La prueba de unidad centra el proceso de verificación en la menor unidad del diseño del software, *el módulo*. Usando la descripción del diseño detallado como guía, 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.

Se ejercitan todos los caminos independientes (caminos básicos) de la estructura de control con el fin de asegurar que todas las sentencias de módulo se ejecutan por lo menos una vez. Y finalmente, se prueban todos los caminos de manejo de errores.

Además de las estructuras de datos locales, durante la prueba de unidad se debe comprobar (en la medida de lo posible) el impacto de los datos globales sobre el módulo.

Prueba de Integración.

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 coger los módulos probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.

Integración descendente. Se integran los módulos moviéndose hacia abajo por la jerarquía de control, comenzando con el módulo de control principal. Los módulos subordinados al módulo de control principal se van incorporando en la estructura.

La estrategia de integración descendente verifica los puntos de decisión o de control principales más pronto en el proceso de prueba. En una estructura de programa bien fabricada, la toma de decisiones se da en los niveles superiores de la jerarquía y por tanto se encuentra antes.

Prueba de Validación.

Tras la culminación de la prueba de integración, debe comenzar una serie final de pruebas del software lo que conocemos como prueba de validación. La validación se logra cuando el software funciona de acuerdo con las expectativas razonables del cliente.

La validación del software se consigue mediante una serie de pruebas de la caja negra, que demuestran la conformidad con los requerimientos.

La **prueba alfa** es conocida por un cliente (*alpha-tester*) en el lugar de desarrollo. Se usa el software de forma natural, con el encargado del desarrollo "mirando por encima del hombro" del usuario y registrando errores y problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado.

La **prueba beta** se lleva a cabo en uno o más lugares del cliente (*beta-testers*) por los usuarios finales del software. A diferencia de la prueba alfa, el encargado del desarrollo normalmente no está presente. Así, la prueba beta es una aplicación "en vivo" del software en un entorno que no puede ser controlado por el equipo de desarrollo. El cliente registra todos los problemas (reales o imaginarios) que encuentra durante la prueba beta e informa a intervalos regulares al equipo de desarrollo. Como resultado de los problemas anotados durante la prueba beta, el equipo de desarrollo del software lleva a cabo modificaciones y así prepara una versión del producto de software para toda la base de clientes.

Prueba del Sistema.

El software una vez validado, se debe combinar con otros elementos del sistema (hardware, gente, bases de datos), la prueba del sistema verifica que cada elemento se acople adecuadamente, alcanzando la funcionalidad y el rendimiento del sistema total.

La prueba del sistema realmente está constituida por una serie de pruebas diferentes, cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Éstas son:

- Pruebas de recuperación
- Pruebas de seguridad
- Pruebas de resistencia
- Pruebas de rendimiento

El proceso de depuración.

La depuración aparece como una consecuencia de una prueba efectiva. Es decir, cuando un caso de prueba descubre un error, la depuración es el proceso que resulta en la eliminación del error.

La depuración comienza con la ejecución de un caso de prueba. Independientemente del enfoque que se utilice, la depuración tiene un objeto primordial: encontrar y corregir la causa de un error en el software. El objetivo se consigue mediante una combinación de una evaluación sistemática y de una gran cantidad de intuición.

5.3 Pruebas del "SIIV II"

Para realizar las pruebas del sistema se tomaron en cuenta las siguientes consideraciones. Conforme a:

- Las características, especificaciones, requerimientos, funciones, alcances y objetivos del sistema.
- Las características propias del lenguaje en el cual se programó el "SIIV II" (Visual Basic) y el ambiente operativo (Windows).

Se llevó a cabo una estrategia de prueba del sistema, involucrando las técnicas de prueba que consideramos pertinentes, para garantizar ante todo la calidad, el buen funcionamiento y eficiencia del "SIIV II".

Dentro de la estrategia de prueba se cubrieron aspectos de gran importancia para poder brindar al usuario un producto atractivo, eficiente y al mismo tiempo cumplir con los objetivos planteados en la etapa de desarrollo.

La estrategia de prueba constó de las siguientes partes:

1. Verificar y validar la unidad e integración del sistema "SIAV II", utilizando las técnicas de prueba en la etapa de desarrollo del sistema.
2. Verificación y validación de los conjuntos de información empleados en el "SIAV II".
3. Validación del sistema "SIAV II" como producto terminado.

5.3.1 Verificar y validar la unidad e integración del "SIAV II".

La programación de Visual Basic es modular y orientada a eventos; donde se programa cada módulo de manera independiente, vinculando por medio de los diferentes eventos los conjuntos de información del sistema y/o módulos correspondientes. De tal manera que al estar elaborando el sistema se utilizó la *prueba de la caja blanca*, para verificar las unidades (módulos) que componen el sistema; al mismo tiempo comprobar los diferentes caminos de control de cada uno de ellos, verificando su integridad.

Esta prueba se realizó durante la etapa que Visual Basic denomina tiempo de diseño; es decir, se lleva a cabo en el proceso de prueba, a diferencia de la prueba de la caja negra que tiende a ser aplicada en tiempo de ejecución.

Se probó la ejecución de cada unidad de forma independiente para verificar la interfaz del módulo, asegurando que el flujo de la información fuera el adecuado; probando las estructuras de datos locales y visualizando la integridad de los mismos, durante todos los pasos de ejecución del algoritmo empleado en la unidad. Además, al estar creando los controles utilizados en cada unidad se verificaron las respuestas esperadas al evento asignado y los conjuntos de información asociados al código de cada forma.

Partiendo de la carta de navegación planteada para la realización del sistema, se llevó a cabo la *prueba del camino básico*, ejecutando cada uno de los eventos asociados a los módulos involucrados en el sistema

para garantizar su operatividad. Se realizó esta prueba para forzar la ejecución de cada una de las líneas de código del programa.

5.3.2 Verificación y validación de los conjuntos de información empleados en el sistema "SIIV II".

Al realizar las pruebas, se resolvieron los ejercicios del sistema y de esta forma se verificó la introducción de datos al sistema, para la validación de las bases de datos se comprobó su estructura, se manejaron los datos de diversas formas para tratar de encontrar fallas y corregirlas. En el caso de los módulos de ayuda, donde se maneja el hipertexto, se realizaron pruebas exhaustivas para verificar las características de navegación en la información. Para comprobar la integridad de la información, se contó con la ayuda de expertos que llevaron a cabo la validación de la misma.

5.3.3 Validación del sistema "SIIV II" como producto terminado.

Se realizaron pruebas de bajo y alto nivel, para así verificar que cada pequeño segmento de código se implementó correctamente. Se comprobó que las funciones y procesos cumplieran con los objetivos planteados para el buen funcionamiento del sistema.

Para validar el software es necesario realizar una serie de *pruebas de la caja negra*, que demuestren la conformidad con los requisitos; para esto se crea una matriz de casos de prueba.

A continuación se da una breve explicación de las pruebas que se realizaron a los módulos y se registraron dentro de la matriz de casos de prueba.

Para verificar los caminos de control de los módulos y su integridad, se utilizó la integración descendente; moviéndose por la jerarquía del control se inician las pruebas por el módulo de control principal y luego con los módulos que se derivan de él.

Se probó la ejecución de cada módulo en forma independiente empezando por la *Presentación*.

En éste módulo se probaron los controles *OK*, *Salir* y *Ayuda*, con los cuales, continúa, termina la sesión y llama la ayuda del sistema, respectivamente.

Al estar en el *Menú Principal* se probaron los módulos principales del funcionamiento del sistema: la unidad de consulta teórica, la de ejercicios, el índice, así como la ayuda de Anatomía II.

Para continuar se usa el módulo de ejercicios y se probó la unidad *Temario*, donde se utilizó el subsistema *Piel* para seguir las pruebas hasta el módulo *Tipos de Examen*, utilizando los controles de *Opción múltiple*, *Respuesta completa* y la opción de *Imprimir*.

Se dio solución a un examen de *Teclear respuesta* para llegar hasta el módulo *Esquemas*, donde se aplicaron pruebas de introducción de datos y la evaluación del examen.

Se regresó al *Temario* y se utilizó el icono que muestra la *Barra de Herramientas*, la cual está integrada por iconos que representan los módulos principales del sistema (mencionados anteriormente). Con ésta se controlan las posibilidades que ofrece el sistema para ir de un módulo a otro del mismo (navegación del sistema).

De ésta forma se siguieron las pruebas a todos los módulos que integran al "SIAV II".

La matriz de casos de prueba es la siguiente.

Objetivo de la prueba	Resultados Esperados								Diseño de casos de prueba	Resultados reales
	Aparece lista	Llama ayuda	Cambia la ruta	Cambia a mayúsculas	Exhibe mensaje	Cambia de pantalla	Selecciona	Permite corregir		
OK						✓			● Hacer clic con el botón izq. ó moverse con las flechas sobre el control y pulsar Enter	Muestra el menú principal
Salir					✓					
Ayuda		✓								
OK		✓					✓		● Clic con el botón der. y F1 ó moverse con las flechas sobre el control y pulsar F1	● Selecciona y muestra la ayuda del sistema
Salir		✓					✓			
Ayuda		✓					✓			
Inf						✓			●	Muestra la presentación
		✓					✓		●	●
Consulta teórica y Ejercicios			✓			✓			●	M. temario y cambia la ruta
		✓					✓		●	●
Índice de esquemas						✓			●	Muestra el índice
		✓					✓		●	●

* Abreviaciones: der. = derecha; izq. = izquierda; M. = muestra

Objetivo de la prueba	Resultados Esperados								Diseño de casos de prueba	Resultados reales	
	Aparece lista	Llama ayuda	Cambia la ruta	Cambia a mayúsculas	Exhibe mensaje	Cambia de pantalla	Selecciona	Permite corregir			
Controles de la forma índice		✓						✓		Clic sobre un elemento de la lista	Aparece el nombre seleccionado en un cuadro de texto
							✓			Clic sobre un elemento y presiona OK	Carga el esquema elegido
							✓			Doble clic sobre un elemento de la lista	
							✓			Presionar cancelar	Regresa a la pantalla anterior
Ayuda de Anatomía		✓								Clic con el botón izq. ó moverse con las flechas sobre el control y Enter	Muestra la ayuda del tema Anatomía II
		✓								Clic con el botón der. ó moverse con las flechas sobre el control y F1	

Objetivo de la prueba	Resultados Esperados								Diseño de casos de prueba	Resultados reales
	Aparece lista	Llama ayuda	Cambia la ruta	Cambia a mayúsculas	Exhibe mensaje	Cambia de pantalla	Selecciona	Permite corregir		
En la forma Temario: control Piel	✓								Clic con el botón izq. ó moverse con las flechas sobre el control y Enter	Selecciona y muestra la lista
	✓	✓						✓	Clic con el botón izq. ó moverse con las flechas sobre el control y F1	Muestra ayuda del tema
								✓	Clic con el botón der. ó moverse con las flechas sobre el control y Enter	Oculto la lista
		✓						✓	Clic con el botón der. ó moverse con las flechas sobre el control y F1	Muestra ayuda del tema
		✓						✓	Clic sobre un elemento de la lista y F1	Selecciona y M. ayuda del sistema
							✓	✓	Doble clic sobre un elemento de la lista	Muestra tipos de examen

Objetivo de la prueba	Resultados Esperados								Diseño de casos de prueba	Resultados reales
	Impresión	Llama ayuda	Cambia la ruta	Cambia a mayúsculas	Exhibe mensaje	Cambia de pantalla	Selecciona	Permite corregir		
Examen de: Opción Múltiple y Teclar Resp.						✓			Clic con el botón izq. ó moverse con las flechas sobre el control y Enter	Muestra esquema
		✓						✓	Clic con el botón der. ó moverse con las flechas sobre el control y F1	Selecciona y M. ayuda del sistema
	✓				✓	✓	✓		Clic con el botón izq. y opción "Impresión de examen", verdadera	Muestra presentación preliminar, exhibe mensaje, permite cancelar o continuar
						✓			Clic con el botón izq. y opción "Impresión de examen", falsa	Muestra esquema

Objetivo de la prueba	Resultados Esperados								Diseño de casos de prueba	Resultados reales
	Aparece lista	Llama ayuda	Cambia la ruta	Cambia a mayúsculas	Exhibe mensaje	Cambia de pantalla	Selecciona	Permite corregir		
Solución de Examen Teclar Res.				✓			✓	✓	Respuesta en letras minúsculas y Enter	Enter o movimiento sobre respuesta: cambia a mayúsculas, permite corregir, se mueve a la sig. respuesta.
		✓					✓		Presionando F1	
				✓	✓		✓		Respuesta en letras minúsculas y Evaluar	Cambia a mayúsculas, permite corregir, da calificación, errores y aciertos
				✓	✓			✓	No hay respuestas y Enter	

Objetivo de la prueba	Resultados Esperados								Diseño de casos de prueba	Resultados reales
	Aparece lista	Llama ayuda	Cambia la ruta	Cambia a mayúsculas	Exhibe mensaje	Cambia de pantalla	Selecciona	Permite corregir		
Icono de herramientas	✓	✓	✓		✓	✓	✓		Clic con el botón izq. ó moverse con las flechas sobre el control y Enter	Manda mensaje, muestra/oculta barra de herramientas. Cambia ruta, llama índice, da la ayuda del tema
		✓			✓		✓		Clic con el botón der. ó moverse con las flechas sobre el control y F1	Manda mensaje y con F1, llama ayuda del sistema

* Abreviaciones: der. = derecha; izq. = izquierda.

Se manejó el sistema en cargas de trabajo reales para realizar ajustes, encontrar errores en estructura de datos, en acceso a las bases de datos, errores de inicialización o terminación del sistema y sobre todo errores de interfaz.

Se llevaron a cabo las pruebas *alfa* y *beta*. La *alfa*, con un usuario en el lugar de desarrollo del software, donde se registraron ideas de mejoras y dudas que causaban problemas en su uso. La *beta*, en el departamento de Morfología y el laboratorio de Genética, de la cual se generaron las últimas mejoras tomando en cuenta las opiniones de los usuarios; de esta forma se tuvo lista la versión final del sistema para todos los usuarios.

Capítulo 6

Conclusiones

Es necesario que todos nos familiaricemos con el mundo del cómputo, debido a que cada día es más palpable la aplicación de la computación a todas las actividades de la sociedad.

La creatividad de los desarrolladores de aplicaciones, apoyada en las nuevas herramientas, está provocando una verdadera revolución en el procesamiento y transmisión de información. Esta revolución alcanza todas las actividades en las que el ser humano moderno se desenvuelve y en particular los procesos de enseñanza-aprendizaje.

El uso de aplicaciones visuales por medio de la computadora está ocasionando cambios drásticos en los procesos y métodos tradicionales de enseñanza, al permitir a los estudiantes aprender a su propio ritmo, al mismo tiempo que estimula sus sentidos, haciéndolos más receptivos a la información.

El "Sistema Interactivo de Anatomía Veterinaria II", es el resultado de esfuerzos encaminados a incorporar la computadora en el proceso educativo, en la Facultad de Medicina Veterinaria y Zootecnia (departamento de Morfología); cumpliendo con los objetivos que se plantearon, para lo cual hubo que sacrificar el uso de audio y video para garantizar la portabilidad del sistema hacia cualquier computadora IBM PC o compatible que soporte Windows 3.1.

El desarrollo del "SIIV II" fue una experiencia excitante y un reto constante. Desde el momento en que se decidió utilizar Visual Basic como lenguaje de programación con el cual no estaba familiarizada.

Es importante mencionar que el sistema "SIIV II" fue creado con la idea de tener un apoyo didáctico para el aprendizaje de la Anatomía, sin poder sustituir con esto las sesiones prácticas que el estudiante de Medicina Veterinaria pueda realizar durante su curso, pero si proveer al estudiante de una herramienta para mejorar su aprovechamiento. Los profesores de asignatura serán los encargados de proporcionar el material que servirá de base teórica a la aplicación desarrollada.

Es uno de los objetivos que este sistema ayude a los estudiantes de Medicina Veterinaria principalmente, a aprender Anatomía de forma más amena que con los métodos tradicionales. Después de realizar una investigación para determinar hasta qué grado el sistema "SIIV II" reduce el tiempo necesario, para que el estudiante promedio aprenda el tema, así como el nivel de conocimientos que puede alcanzar, se obtuvieron los siguientes resultados.

Para realizar la investigación se formaron dos grupos aleatoriamente, integrados por 5 personas cada uno. Ambos recibieron la clase tradicionalmente, pero uno de los grupos tuvo una sesión de una hora de estudio, dos veces por semana en la computadora con el programa (grupo prueba) y el otro utilizó los métodos tradicionales de estudio (grupo control). Finalmente se les aplicó a ambos un examen.

Los cuadros siguientes nos muestran las calificaciones obtenidas y el tiempo de estudio (extraclase) de los alumnos en los dos grupos.

Tabla del grupo control.

<i>Alumno</i> <i>Tema</i>	1	2	3	4	5
Cabeza	7	6	8	7	9
Cuello	6	8	7	7	8
Tronco	7	7	8	6	8
<i>Promedio</i>	6.7	7	7.7	6.7	8.4
<i>Promedio General</i>				7.3	

Tabla del grupo prueba.

<i>Alumno</i> <i>Tema</i>	1	2	3	4	5
Cabeza	8	10	9	8	10
Cuello	9	9	10	10	10
Tronco	10	10	9	10	9
<i>Promedio</i>	9	9.7	9.4	9.4	9.7
<i>Promedio General</i>				9.44	

El tiempo de estudio empleado por ambos grupos se muestra en la tabla siguiente:

Tabla de tiempos de estudio.

<i>Grupos</i> <i>Tema</i>	<i>Control</i> <i>(horas)</i>	<i>Prueba</i> <i>(horas)</i>
Cabeza	4	3
Cuello	5	4
Tronco	5	4

De esto podemos concluir, que los estudiantes que usaron el sistema requirieron entre 20% y 25% menos tiempo para aprender el tema y su nivel de conocimientos es mayor en un 23%, que los estudiantes de utilizaron métodos tradicionales.

Es importante mencionar que se cumplen los objetivos primordiales de la creación del "SIIV II" que son, proporcionar al alumno un medio de aprendizaje autodidacta que lo motive para el estudio de la materia de Anatomía II y al profesor de asignatura un auxiliar en la enseñanza de su materia, así como en la evaluación del aprendizaje alcanzado por los alumnos. Hasta antes de la implementación de éste sistema sólo se contaba con métodos tradicionales de estudio. Al proporcionar ésta herramienta de estudio se pretende mejorar el nivel de conocimientos y el tiempo necesario para asimilar los temas.

El sistema "SIIV II" es amigable, y tiene un manejo fácil y llamativo de la información, estos factores son un punto de referencia para el mundo de la enseñanza, ya que los diferentes elementos empleados en el desarrollo de temas y presentaciones deben atrapar la atención del estudiante involucrando sus sentidos.

Desde el punto de vista educativo, las ventajas ofrecidas por el sistema se pueden englobar en los siguientes puntos:

- "SIIV II" permite a los estudiantes navegar entre imágenes que ilustren una idea clave y nombrar los conceptos relacionados con dicha idea. Esta dinámica da a las lecciones otra dimensión ya que promueve una amplia participación del estudiante, beneficiando su proceso de aprendizaje.
- Con "SIIV II" el estudiante está listo para ejecutar algo, interactuando con la computadora al contestar cuestionarios y navegando en la información, creando su propio patrón de conocimientos. Una vez que los estudiantes han experimentado una lección del sistema, pasan a la fase del cuestionamiento en donde hacen preguntas relacionadas con los conceptos que vieron, desarrollando así su pensamiento crítico y pasando de la comprensión y aplicación de un tema, a su análisis.
- Una clase en donde se utilice como apoyo el sistema ("SIIV II"), será activa y estimulante por naturaleza; los estudiantes y los profesores no pueden adquirir un papel pasivo, ya que están interactuando entre ellos todo el tiempo.
- Cada persona tiene su propia forma de aprender; hay personas cuya capacidad de aprendizaje está orientada a material escrito. A otras les puede ser más fácil aprender utilizando medios visuales. Algunas personas aprenden más rápido que otras. El sistema "SIIV II" soporta estas diferentes necesidades de aprendizaje, pudiendo así elevar la capacidad de aprendizaje de los alumnos al individualizar el proceso de enseñanza-aprendizaje.

Gracias a este sistema los estudiantes tienen la oportunidad de aprender haciendo.

Otro punto importante son las limitaciones del sistema, para ello podemos enfocarnos en dos puntos básicos. Por un lado, el uso de esta herramienta estaría reservada únicamente para quienes tienen acceso a una computadora personal, con las características requeridas por el sistema "SIIV II". Por otra parte, el sistema abarca temas de estudio que se enfocan sólo en animales bovinos, debido a la extensión de la materia y se está planeando con el personal encargado abarcar las áreas de cerdos, caballos, perros y pequeñas especies como gatos y ratones. La actualización del sistema en caso de reestructurar el temario será parte del mantenimiento que deberá darse al sistema.

"SIIV II", es un primer paso hacia la educación apoyada en computadoras en la Facultad de Medicina Veterinaria y Zootecnia de la Universidad Nacional Autónoma de México.

Apéndice A

Manual de Usuario

Introducción.

El surgimiento de la computadora y su aplicación como herramienta de trabajo ha traído como consecuencia la modificación de las formas de trabajar en prácticamente todos los ámbitos. Actualmente se interactúa con las computadoras al realizar actividades cotidianas como las operaciones bancarias que realizamos en un cajero, o al ordenar alimentos en los restaurantes. Esto ha traído como consecuencia el nacimiento de nuevas tecnologías que han permitido la creación de herramientas cada vez más poderosas que faciliten las múltiples actividades que realizamos, enfocadas a diversas áreas como lo son la medicina, administración, finanzas, investigación científica, ingeniería, arte y entretenimiento; y el desarrollo de interfaces cada vez más amigables, las cuales faciliten el uso de las computadoras a los usuarios sin experiencia. Con este propósito y haciendo uso de tecnología de vanguardia, la creación de una aplicación orientada al proceso educativo como el *Sistema Interactivo de Anatomía Veterinaria II (SIAV II)*, brinda beneficios significativos a estudiantes y maestros, ya que uno de los elementos más importantes dentro de este esquema es la computadora.

Con base en lo anterior se definen a continuación los propósitos fundamentales del presente sistema.

- Proporcionar al alumno un medio de aprendizaje autodidacta, que por sus características innovadoras y a través de una interface

interactiva, sea una motivación para que el usuario dedique tiempo al estudio de la Anatomía.

- Proporcionar al profesor de asignatura un auxiliar en la enseñanza de Anatomía II, así como en la evaluación del aprendizaje alcanzado por el alumno.
- Hacer uso de nuevas formas de interacción hombre-computadora.
- Permitir que sus usuarios lo puedan usar en cualquier computadora PC compatible que soporte el ambiente gráfico Windows.

Requerimientos del equipo de cómputo para la operación del sistema.

Las características del hardware que se requiere para poder hacer uso del sistema "SIIV II" son:

Configuración mínima:

Computadora	IBM PC o compatible con procesador 80386.
Velocidad	33 MHz.
Memoria	4 MB de memoria RAM.
Disco duro	7.5 MB de capacidad disponible.
Monitor	Monitor SVGA (800 x 600) de 16 colores (no se recomienda usar monitores monocromáticos).
Unidad de discos	Drive de 3" ½ de doble densidad (720 KB).
Impresora	Matriz de puntos de 300 c.p.s., de carro de 15" capaz de emular IBM Proprinter.
Mouse	Mouse serial compatible con el equipo.
Plataforma	MS-DOS versión 3.3 y Windows 3.1.

Configuración recomendable:

Computadora	IBM PC o compatible con procesador 80486 o superior.
Velocidad	66 MHz.
Memoria	8 MB de memoria RAM.
Disco duro	Capacidad disponible mayor a 7.5 MB.
Monitor	Monitor SVGA (800 x 600) de 256 colores o superior (no se recomienda usar monitores monocromáticos).
Unidad de discos	Drive de 3" ½ de alta densidad (1.44 MB).
Impresora	HP Láser Jet 4L o superior, HP Deskjet 660 o superior.
Mouse	Mouse serial compatible con el equipo.
Plataforma	MS-DOS versión 6.2 o posterior y Windows 3.1 o 95.

Como se puede ver, el sistema "SIIV II" sólo requiere una PC típica y no se necesita tener ningún equipo especial ni muy costoso. Esto permite que el sistema sea 100% transportable entre una PC y otra, gracias a lo cual los usuarios no tienen que depender de una computadora en particular para hacer uso del mismo.

Procedimiento de instalación.

1. En la línea de comandos del DOS escriba **Win** y pulse la tecla **Enter**.
2. Coloque el disco de instalación 1 en la unidad del disquete.
3. Haga clic en **Archivo**, en la barra del menú de la ventana *Administrador de programas*.
4. En el menú de Archivo, escoja **Ejecutar**. Aparecerá el cuadro de diálogo Ejecutar.

5. En la casilla línea de comando, escriba el designador de la unidad del disco flexible (por ejemplo, **A:**), seguida de **INSTALAR**, como sigue: **A:\INSTALAR**
6. Haga clic en **OK**.
7. El programa de instalación verificará los recursos de su equipo, así como el espacio necesario en disco dura para iniciar su instalación.
8. Durante la instalación del Sistema Interactivo de Anatomía Veterinaria II, aparecerán varios cuadros de diálogo. Se le pedirá la unidad del disco duro y el subdirectorio destino para la instalación.
9. Automáticamente se asignará la siguiente ruta: **C:\SIAV2**, si está de acuerdo presione **Enter** para iniciar la instalación.

Nota. En caso de existir un directorio con el mismo nombre se le informará, permitiéndole seleccionar otro directorio para iniciar la instalación, de lo contrario se borrará la información contenida en éste y se instalarán los archivos del "SIAV II".

Cómo ejecutar el sistema.

Una vez instalado el programa, siga estos pasos.

1. Estando en Windows, deberá *seleccionar el grupo* dentro del cual se encuentra el icono que ejecuta el sistema.
2. Haga *doble clic* en el icono del *Sistema Interactivo de Anatomía Veterinaria II*.



3. Una vez dentro del sistema verá la pantalla de presentación y los controles de opción.

Utilizando el manual.

La elaboración de este manual tiene como finalidad proporcionarle una guía práctica que le facilite el conocimiento y manejo del *Sistema Interactivo de Anatomía Veterinaria II*. Se encuentra organizado en forma jerárquica y navegará a través del sistema, partiendo desde el menú principal, moviéndose por cada una de sus opciones y herramientas disponibles. La característica de *interacción* del sistema permite que el manejo de éste sea una tarea fácil de aprender y dominar, lo cual se podrá comprobar durante el estudio de este sencillo manual.

El sistema contiene las siguientes pantallas principales:

1. Presentación.
2. Menú Principal.
3. Temario.
4. Elección de regiones (ésta depende del segmento que se haya elegido en el temario).
 - 4.1 Tipos de examen.
 - 4.2 Esquema elegido (depende de la pantalla anterior).



Para realizar una acción es necesario seleccionar un botón y presionarlo utilizando el botón izquierdo del mouse.



Para llamar la ayuda sobre cada botón o tema, selecciónelo primero, presionando el botón derecho del mouse y enseguida la tecla <F1>.

1 Presentación, proporciona los detalles de la elaboración del sistema.

Puede llamar la ayuda del sistema presionando el botón

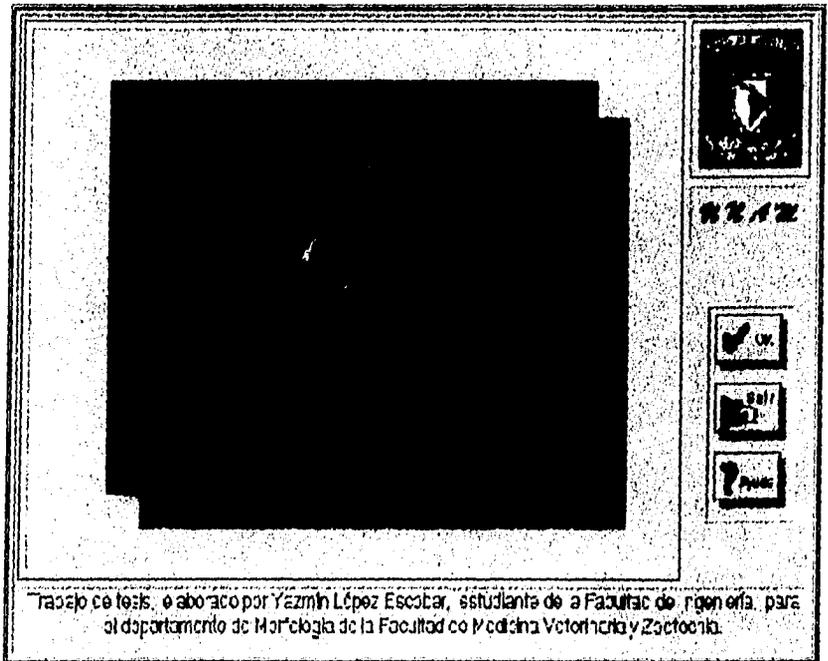


continuar la sesión con



o abandonarla presionando 

Estos tres botones los encontrará en todas las pantallas. La pantalla de presentación que verá es:



2 Menú Principal, presenta las opciones de introducción al sistema.

El "SIIV II", tiene como raíz este menú, a partir del cual es posible manejar cuatro nodos distintos (navegación del sistema):



Nodo de consulta teórica, ofrece la opción de aprender o repasar los conceptos del tema, Anatomía II.



Nodo de ejercicios, ofrece la opción de realizar dos tipos de exámenes con los que podrá reforzar los conocimientos adquiridos en la teoría.



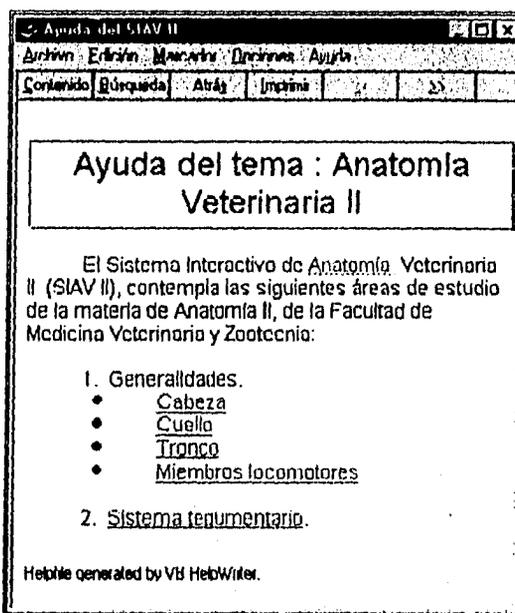
Nodo de índice de esquemas, funcionará como auxiliar para realizar un acceso rápido a esquemas específicos, que se encontrarán ordenados alfabéticamente.



Nodo de ayuda de Anatomía, le presenta un estudio de cada región, hipertexto, conceptos e información rápida.

Es importante aclarar que al navegar en el "SIIV II", en cualquier momento, excepto al llegar a un esquema, puede utilizar los iconos mostrados anteriormente, (que constituyen la **barra de herramientas**) para establecer la ruta de trabajo (teoría o práctica) o moverse en el sistema.

La pantalla de Ayuda de Anatomía se muestra a continuación:



Básicamente el cuerpo del sistema se compone de *dos rutas o módulos* de trabajo, una **teórica** y una **práctica**.

Consulta teórica

En este módulo se despliegan los esquemas contenidos en el sistema con su respectiva información. Es posible obtener información detallada de cada una de las partes señaladas en los esquemas, pulsando la tecla <F1>.

Ejercicios

El módulo de ejercicios permite evaluar sus conocimientos adquiridos mediante la solución de *dos tipos de exámenes*:

- a) Opción múltiple.
- b) Teclear respuesta.

3 Temario, aparece al elegir cualquiera de los dos casos anteriores, en esta pantalla el usuario debe elegir alguno de los temas siguientes:

Generalidades

-  Cabeza
-  Cuello
-  Tronco
-  Miembros locomotores

Sistema tegumentario

-  Piel
-  Apéndices cutáneos

 Nota 1. Estando en esta pantalla presione el botón  para hacer visible la barra de herramientas .

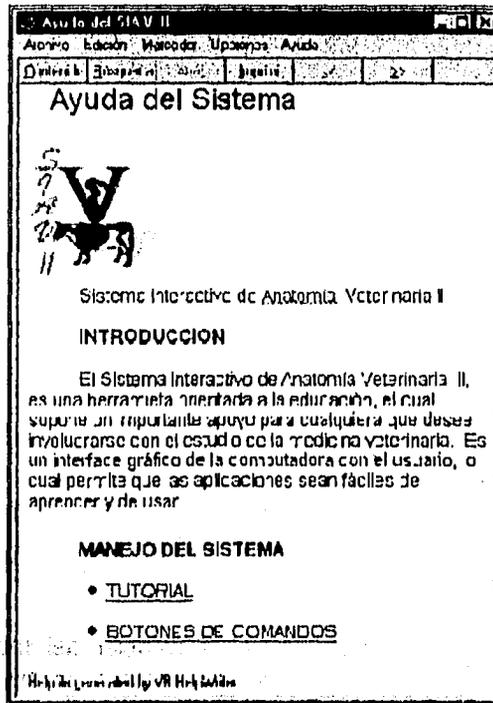
Presione uno de los iconos si desea cambiar del modo teórico a ejercicios o viceversa; ir al índice u obtener la ayuda del tema (Anatomía).

La ayuda del sistema.

Podrá llamar la **ayuda** si hace "click" con el botón derecho del mouse y enseguida presiona la tecla <F1>, sobre:

-  El título del segmento en que se encuentra (Cabeza, cuello, tronco o miembros locomotores).
-  En alguna región iluminada de la imagen.
-  Una lista de esquemas.
-  Cualquier otro botón.

La pantalla de ayuda es la siguiente:



4 Elección de regiones. En caso de haber elegido un subtema del tema "Generalidades", aparecerá una imagen que nos mostrará las regiones del segmento a estudiar. Para trabajar en estas pantallas los pasos básicos son:

- ☞ Mueva el ratón sobre la imagen, podrá observar el nombre de la porción en que se encuentra posicionando el cursor.
- ☞ Si desea un estudio más detallado, haga "click" sobre una región. Los nombres de las regiones disponibles, se encuentran en color azul.
- ☞ En caso de que esta región contenga varios esquemas, aparecerá una lista de ellos, haga *doble "click"* con el ratón sobre el esquema

que elija y la pantalla cambiará dependiendo del modo en que se encuentre trabajando.

- ☞ Existen esquemas generales de la porción que eligió, presione el botón que lo indica; si son varios aparecerá una lista de la cual puede elegir uno.

 Nota 2. Para "cancelar" la operación haga *doble "click"* sobre la misma región, o un "click" fuera de las regiones iluminadas y la lista desaparecerá.

En caso de haber *elegido* un elemento de la lista, el resultado de la siguiente pantalla, depende del modo o ruta en que haya decidido trabajar.

Ver Nota 1 .

 Si está en ejercicios:

4.1 Aparecerá una pantalla que contiene los **Tipos de Examen** en que se pueden realizar ejercicios.

- Examen de opción múltiple.
- Examen teclear respuesta.

Ver Nota 1 .

Tipos de examen.



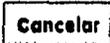
Opción Múltiple. El examen de opción múltiple iniciará en el momento que presione el botón; aparecerá el esquema seleccionado, y una lista de la cual podrá elegir la respuesta, haciendo "click" con el mouse sobre un elemento de ella; después colóquese sobre los recuadros de respuesta y repita la acción.



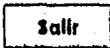
Teclear respuesta. En este ejercicio podrá contestar tecleando las respuestas en los recuadros de respuesta que aparecerán, tal como lo estudió en la teoría. Este tipo de examen requiere de que el texto sea el mismo que aparece en la sección teórica para que la respuesta sea correcta. Después de teclear la respuesta presione <Enter> para continuar o utilice el mouse para moverse al siguiente recuadro.

En un esquema.

Si desea cancelar la última respuesta en cualquier caso, presione



Para regresar a la pantalla anterior "tipos de exámenes" presione éste icono.

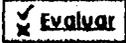


Podrá salir del esquema al hacer "click" en este botón y regresar al menú principal.

Opciones de los exámenes.



Cronómetro. Ambos exámenes tienen la opción de ser cronometrados para conocer el tiempo que tarda en resolver estos ejercicios, para esto es necesario presionar el botón; éste nos indicará que está activo el cronómetro cambiando de color.



Evaluación. El tiempo empieza a correr en el momento en que seleccione el examen y aparezca el esquema. Finalizará en el momento en que presione el botón; obtendrá la calificación y el resultado en cada respuesta.



Impresión. Mientras esté presionado el botón se iniciará el proceso de impresión cada vez que se llame un esquema (presionando uno de los botones de examen). Podrá ver una presentación previa del esquema en impresión y aparecerá un cuadro de diálogo donde podrá confirmar o cancelar la acción.

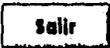


Nota 3. Si desea obtener ayuda sobre algún botón, selecciónelo presionando el botón derecho del mouse  y enseguida la tecla <F1>.



Si está en consulta teórica:

4.2 La pantalla que aparecerá es el **Esquema Elegido** con toda su información. Aquí puede utilizar:



para regresar al "Menú principal",

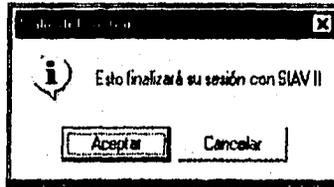


para ir al siguiente esquema.

Abandonar la sesión.

- Presione el control 

- Aparecerá el siguiente cuadro de diálogo para confirmar o cancelar.



Botones de Comandos.



Con este botón la sesión continúa al menú principal.



Presione el botón de ayuda cuando desee obtener información del sistema. Llama el archivo de ayuda del sistema.



El botón salir, cierra la pantalla en la que se encuentra y regresa al menú principal.



El botón de información, se encarga de mostrar detalles acerca de la elaboración del sistema.



Dependiendo del estado de este botón (ver herramientas), si está presionado, se hace visible la barra de herramientas, si no lo está se oculta.



La barra de herramientas que contiene 4 iconos se hace visible, sólo si, se encuentra presionado el botón anterior (llave de herramientas), la encontrará en la mayoría de las pantallas y hace posible la navegación en el sistema.



Este icono indica el modo en que se está trabajando, si se eligió consulta teórica su aspecto será: 

 Igual que el anterior, indicará si el modo de trabajo es ejercicios de la forma siguiente: 

 Proporciona un acceso rápido a los esquemas. Si el icono es presionado, aparecerá la ventana de índice, con una lista de esquemas ordenada alfabéticamente.

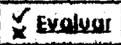
 Presione este icono para ver la ayuda del tema Anatomía II.

 Presione este botón para establecer como tipo de examen, *Opción múltiple*.

 Igual que el anterior, establece como tipo de examen, *Teclear respuesta*.

 Si desea cronometrar el tiempo que le toma resolver el examen, presione este icono.

 Si está presionado este control, se encuentra activa la impresión. Podrá imprimir un esquema en el formato de examen que elija.

 **Evaluar** Para finalizar el examen, presione este botón y obtendrá su calificación. Podrá observar sus aciertos y errores en cada respuesta.



Utilice estos iconos para moverse de una pantalla a otra. La flecha izquierda presentará la pantalla anterior, la derecha, la pantalla siguiente. Cabe mencionar que seguirán un nivel jerárquico para formar su ruta de navegación.



Estas flechas tienen la misma función que las anteriores, sólo que se encuentran en los esquemas y le permitirán moverse dentro de los grupos formados por cada tema.

Cancelar

Éste se encuentra en la pantalla de esquemas y puede utilizarlo para eliminar la última respuesta contestada en los ejercicios.

Salir

Este botón se encuentra en los esquemas y lo conducirá al menú principal.

Apéndice B

Algoritmo: conjunto de pasos ordenados para resolver un problema, tal como una fórmula matemática o las instrucciones de un programa.

Adaptador de video: denominado también, tarjeta, placa de video o controlador gráfico. Es una tarjeta que convierte los comandos de visualización de la computadora en imágenes sobre la pantalla. Funciona como un transductor, convirtiendo las señales de la computadora en señales de voltaje que controlarán el barrido de la pantalla de despliegue.

Animación (gráfico animado): dibujos o diagramas en movimiento. Ocupan mucho menos espacio en disco que las imágenes en video.

Archivo: una colección de registros relacionados. En gráficos por computadora, un conjunto de descriptores de imágenes para una figura, tanto en formato de TV (gráficos de trama) como en formato de líneas o de objetos (gráficos vectoriales).

Base de datos: colección integrada de datos, interrelacionados entre sí, almacenados sin redundancia para servir a una o más aplicaciones.

Base de datos relacional: base de datos formada por un conjunto de archivos (tablas) y relaciones entre ellos. Una tabla es un arreglo bidimensional o una colección de registros, formada por renglones (registros) y columnas (atributos).

Bit (dígito binario): unidad mínima de almacenamiento de datos en un computador. Un dígito simple de un número binario (1 o 0). En el computador, un bit es físicamente una celda de memoria (constituida por transistores o un transistor y un condensador), un punto magnético en un disco o una cinta, o un pulso de alto o bajo voltaje viajando a través de un circuito. Conceptualmente, un bit

se puede pensar como una lamparita eléctrica, encendida o apagada. Grupos de bits forman unidades de almacenamiento en el computador llamadas caracteres, bytes o palabras, que son tratados como un grupo.

Byte (octeto, byte): La unidad más común de almacenamiento en computación desde computadoras personales hasta macrocomputadoras. Se compone de ocho dígitos binarios (bits). Puede agregarse un noveno como bit de paridad, para comprobación de errores. Un byte contiene el equivalente a un solo carácter, tal como la letra A, el signo \$, o el punto decimal. En cuanto a los números, un byte puede contener un solo dígito del 0 al 9 (decimal), dos dígitos numéricos (decimal empaquetado) o un número entre 0 y 255 (números binarios).

Click: seleccionar un objeto presionando un botón del ratón cuando el cursor está apuntando la opción o icono deseados.

Compresión de imágenes: existen varios métodos de compresión, uno de ellos busca *patrones repetitivos* en el archivo fuente y, en vez de almacenar todos los datos, guarda cada patrón que se repite sólo una vez junto con las instrucciones necesarias para reconstruir la imagen original. Usando esta técnica la imagen restaurada es idéntica a la original.

Otro método se llama *compresión lossy*, donde alguno de los datos del archivo original se pierden en el momento de hacer la compresión. En vez de buscar patrones repetitivos, este método realiza un análisis matemático sofisticado sobre los datos originales. La idea de este método es que el ojo humano no es capaz de detectar la información perdida en la imagen reconstruida. El formato más común en las PC's para almacenar los datos de una imagen es el RGB (guarda las proporciones de rojo, verde y azul que contiene la imagen). Las imágenes guardadas así son más fotorealistas, sobre todo en alta resolución. Por lo tanto, mientras mejor sea la imagen, mayor será la cantidad de datos requerida. La técnica de compresión lossy tiene grandes ventajas cuando se le compara con los métodos de compresión tradicionales logrando, en ocasiones, compresiones de 10:1 o 20:1

sin distorsionar seriamente la imagen. Tiene también algunos problemas, por ejemplo, al comprimir nuevamente un archivo restaurado, se pierden más datos que la primera vez lo cual ocasiona mayor degradación en la imagen y algunas aplicaciones no lo soportan.

Existen dos estándares de compresión relativamente recientes, que combinan el uso de hardware y software para lograr la compresión; fueron desarrollados por el *Join Photographic Experts Group* (JPEG) y el *Join Motion Picture Experts Group* (MPEG) trabajando bajo el auspicio de la *International Standards Organization* (ISO). El primer grupo propuso un estándar universal para la compresión y descompresión de imágenes fijas en sistemas de cómputo, mientras el segundo hizo lo propio para video y audio.

Además de convertir las imágenes a formato RGB para hacer más fácil el proceso de compresión, JPEG ordena los colores en la imagen para determinar cuales son los más comúnmente usados. Esto permite controlar, en cierta medida el grado de degradación de la imagen durante el proceso de compresión.

JPEG divide la imagen en pequeños bloques y comprime cada uno de estos antes de pasar al siguiente. Esto permite la compresión instantánea, pero introduce algunos problemas. Los límites de bloque pueden ser remarcados, distorsionando la imagen cuando se manejan altos factores de compresión. JPEG tiene problemas al manejar colores que involucran altas frecuencias como el azul. Esto significa que los colores azulados tienen pocas variantes. Este hecho no es muy importante, ya que el ojo humano tiene el mismo problema para distinguir variaciones mínimas en tonos de azul. Este estándar es soportado en PC's por Windows.

CPU (Control Precessing Unit): unidad central de proceso (procesamiento). La parte de una computadora que realiza la computación.

También llamada el procesador, está constituida por la unidad de control y la ALU (unidad aritmética y lógica).

Cursor: un símbolo móvil en la pantalla de presentación que sirve como punto de contacto entre el usuario y los datos. En los sistemas basados en texto, el cursor es un rectángulo o símbolo de subrayado destellante. En los sistemas gráficos, el cursor puede tomar cualquier forma (flecha, cuadrado, pincel, etc.) y habitualmente cambia de forma se desplaza a diferentes zonas de a pantalla.

Dato: Número o caracter.

DOS (Sistema operativo en disco): la denominación genérica de un sistema operativo. Un sistema operativo monousuario para las series PC, PS/1 y PS/2 de IBM. Desarrollado por Microsoft, el desarrollo del DOS es compartido con IBM de tanto en tanto.

EXE (archivo ejecutable): programa en Dos, OS/2 y VMS. En dos, un programa que cabe en 64K, puede ser un archivo COM.

Gestión de memoria: Llamada también administración de memoria, es la forma en la que una computadora trata con su memoria, lo que incluye la protección de memoria (técnica que prohíbe que un programa afecte accidentalmente a otro programa), memoria virtual (técnica que simula más memoria que la que realmente existe, enlazando el almacenamiento interno con un almacenamiento auxiliar) y técnicas de intercambio de bancos.

Un administrador de memoria es un conjunto de programas que manejan los contenidos de la memoria de un sistema. Incluye la asignación de los programas a ubicaciones de la memoria y el movimiento de segmentos de programas y datos hacia y desde la memoria interna y los dispositivos de almacenamiento externo.

GUI (Graphical User Interface): interfaz gráfica de usuario basada en gráficos que incorpora iconos, menús enrollables y un ratón, tal como se encuentra en los entornos de Macintosh, Windows, OS/2 y Presentation Manager.

Hardware: los dispositivos electrónicos (por ejemplo, CPU, memoria) que proporcionan la capacidad de computación y los dispositivos

electromecánicos (por ejemplo, sensores, motores, bombas) que proporcionan las funciones del mundo exterior.

Icono: una diminuta representación pictórica de un objeto, tal como una aplicación, archivo o unidad de disco, que se utiliza en interfaces gráficas de usuario (GUI). El usuario selecciona un objeto apuntando el icono correspondiente y presionando la tecla de ratón. Un icono puede ser desplazado en la pantalla.

Información: Sentido que le da el manejador a los datos

Ingeniería de software: es la disciplina tecnológica y administrativa dedicada a la producción sistemática de productos de programación, que son desarrollados y modificados a tiempo y dentro de un presupuesto definido.

Interacción: acción recíproca entre dos fenómenos, factores o sistemas.

Interactividad: tecnología que permite a los usuarios controlar, como nunca antes se había hecho, el flujo de las aplicaciones. Lo que ve, escucha y/o lee, es el resultado directo de sus acciones y decisiones.

Interfaz: es aquel componente de una aplicación que traduce una acción del usuario en una o más peticiones para desarrollar funciones y retroalimentar al usuario con las consecuencias y sus acciones.

Mapa de memoria: localización (posición) de instrucciones y de datos en memoria.

Mapeo: Transferencia de un conjunto de objetos de un lugar a otro.

MB: 1,000,000 bytes o caracteres.

Memoria: Almacenamiento de grupos de dígitos (palabras) binarios que pueden representar instrucciones (programa) que la computadora ejecutará y los datos que serán operados por el

programa. Físicamente es una colección de chips. Es un recurso importante de la computadora ya que determina el tamaño y el número de programas que pueden ejecutarse al mismo tiempo, como también la cantidad de datos que pueden ser procesados instantáneamente.

Monitor: pantalla de visualización de alta resolución para la salida desde una computadora, una cámara VCR u otro generador de video. La claridad del monitor se basa en el ancho de banda del video y la densidad de puntos.

Mouse: un objeto semejante a un tejo o pastilla que se usa como un dispositivo puntero y de dibujo. A medida que se hace rodar sobre una superficie, el cursor o puntero se mueve sobre la pantalla.

Multimedia: la comunicación de información en más de una forma; incluye el uso de texto, audio, imágenes fijas, animación, video e hipertexto.

Multiprogramación (multitarea): la ejecución de dos o más programas en una computadora al mismo tiempo. La multitarea se controla por el sistema operativo, que carga los programas y los maneja hasta que terminan.

El número de programas que pueden ser efectivamente ejecutados depende de la cantidad de memoria disponible, la velocidad del CPU, capacidad y velocidad de los recursos periféricos, así como también de la eficiencia del sistema operativo. La multitarea se realiza debido a las diferencias de velocidades de entrada/salida y procesamiento.

Con programas interactivos, los segundos de demora entre entradas de teclado se usan para ejecutar instrucciones de otros programas

Tradicionalmente multitarea significaba ejecutar dos o más tareas dentro del mismo programa al mismo tiempo, y multiprogramación significaba ejecutar dos o más programas en la computadora al

mismo tiempo. Hoy, multitarea significa multiprogramación y multilectura significa, multitarea.

Multilectura: Multitarea dentro de un programa simple. Se utiliza para procesar muchas transacciones o mensajes tratados por partes. También se requiere para crear aplicaciones de audio y video sincronizadas.

Pentium: Es el último microprocesador de Intel, de 64 bits (64 de datos /32 de direcciones), compatible hacia atrás en código con la familia 80x86. Se han hecho esfuerzos por hacerlo un procesador rápido a pesar de la enorme carga que es la compatibilidad. Realiza las operaciones 16 o 32 bits de manera optimizada, con su sistema predictivo de tuberías intenta realizar hasta dos instrucciones simultáneas siempre que el código se lo permita.

Pixel: elemento de imagen. Es el elemento más pequeño en una pantalla de presentación de video. Una pantalla se divide en miles de pequeños puntos, y un pixel es uno o más puntos que se tratan como una unidad.

Procedimientos: los pasos que definen el uso específico de cada elemento del sistema o el contexto procedimental en que reside el sistema.

Punto: figura geométrica sin dimensiones.

Sistema: conjunto u ordenación de elementos organizados para llevar a cabo algún método, procedimiento o control mediante el procesamiento de información.

Software: los programas de computadora, estructura de datos y documentación asociada que sirven para realizar el método lógico, procedimiento o control requerido.

SVGA: presentación de video con 256 colores y modo de resolución de 800 x 600 pixeles, más alta que en VGA. La Video Electronics Standards Association (VESA) ha establecido un estándar para

Super VGA; un paso importante, ya que muchas versiones independientes han sido desarrolladas.

Usuario: individuo que opera el software y el hardware.

VGA: un estándar de presentación de video de IBM, es está incorporado a los modelos más sofisticados de la serie PS/2 de IBM y suministra textos y gráficos de media a alta resolución. VGA se ha convertido en el estándar de video preferido, y comúnmente hay tarjetas disponibles para todas las máquinas de bus PC y AT. Soporta estándares de presentación previos y requiere de un monitor RGB analógico. Tiene 16 colores en su máximo modo gráfico (600x480) pixeles.

80386: microprocesador perteneciente a la familia Intel, presenta dos versiones la 386SX, que es un chip 16/32 con un bus de direcciones de 24 bits; y la 384DX, es un procesador 32/32 con bus de direcciones de 32 bits y puede acceder hasta 4 gigabytes de memoria.

80486: el Chip Intel 80486 es una versión de mayor velocidad del 386 y tiene un coprocesador matemático incorporado. Es aproximadamente el 50% a 300% más rápido que el 386, dependiendo de la aplicación, y es adecuado para estaciones de trabajo gráficas, servidores de archivos de red y computadoras multiusuarios.

Bibliografía

- [BER87] BERG, Rolf; **Anatomía Topográfica y Aplicada de los Animales Domésticos**. AC; 2a. edición; Madrid, España, 1987; 415 pags.
- [BUR92] BURCH, G. John & GRUDNITSKI, Gary; **Diseño de Sistemas de Información**. Grupo Noriega; 1a. edición; México, 1992; 985 pags.
- [CEB94] CEBALLOS, Sierra Francisco Javier; **Enciclopedia de Visual Basic. ra-ma**; Madrid, España, 1994; 792 pags.
- [CON93] CONACYT; *Información científica y tecnológica*. Vol.15, No. 205; Octubre, 1993; Artículo: "Tecnología y Educación".
- [COR94] CORNELL, Gary; **Manual de Visual Basic 3 para Windows**. Osborne / Mc. Graw-Hill; Madrid, España, 1994; 792 pags.
- [FAI88] FAIRLEY, Richard; **Ingeniería de Software**. Mc. Graw-Hill; México, 1988; 290 pags.
- [FEL93] LARA, Lozano Felipe & GUERRERO, Briseño Pedro; *Multimedia del Instituto de Ingeniería en la UNAM*; Laboratorio de Inteligencia Artificial y Multimedia; Mayo, 1993; Artículo: "Metodología básica para el desarrollo de aplicaciones de multimedia en la educación".
- [FRA88] FRANDSON, R.D.; **Anatomía y Fisiología de los Animales Domésticos**. Mc. Graw-Hill; 4a. edición; México, D.F., 1988; 527 pags.
- [FRE94] FREEDMAN, Alan; **Diccionario de computación**. Mc. Graw-Hill; 5a. edición; México, 1994; 394 pags.
- [GRA85] GRAU, Hugo / WALTER, Peter; **Histología y anatomía microscópica comparada de los mamíferos domésticos**. Labor; 1a. edición; Barcelona, España, 1985; 210 pags.

- [LAM95] LAMMERS, Jim & PETERSON, Michael T.; **3D Studio**. New Riders Publishing; Indianapolis, IN. 46290 USA, 1995; 497 pags.
- [LUT92] LUTER, Arrch C.; **Designing Interactive Multimedia**. Bantam Books; E.U.A., 1992.
- [MIR92] MIRCHANDANI, Tatjana P.; **The Video Logic Multimedia Solutions Guide**. Video Logic Inc.; Verano, 1992.
- [NEL94] NELSON, Ross; **Guía Completa de Visual Basic para Windows**. Mc. Graw-Hill; 2a. edición; México, 1994; 396 pags.
- [NUS77] NUSSHAG, Wilhelm; **Compendio de Anatomía y Fisiología de los Animales Domésticos**. ACRIBA; Zaragoza, España, 1977; 385 pags.
- [PC94] PC SEMANAL. 9 de Abril, 1994; Artículo: "Tecnología CD-ROM".
- [POP81] POPESKO, Peter; **Atlas de Anatomía Topográfica de los Animales Domésticos**. Tomo I, II y III, Salvat; Barcelona, España, 1981.
- [PRE93] PRESSMAN, Roger S.; **Ingeniería del Software**. Mc. Graw-Hill; 3a. edición; México, 1993; 824 pags.
- [SIS83] SISSON y GROSSMAN; **Anatomía de los Animales Domésticos**. Salvat; 5a. edición; México, 1983; 1335 pags.
- [SOM88] SOMMERVILLE, Ian; **Ingeniería de Software**. Adison Wesley Iberoamericana; 2a. edición; E.U.A., 1988; 380 pags.
- [WOD94] WODASKI, Ron; **Multimedia Madness**. Sams Publishings; E.U.A., 1992.