



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA EDITOR DE FIGURAS GEOMETRICAS  
BASICAS EN TERCERA DIMENSION PARA PC.

**T E S I S**

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A N :

GONZALEZ GALINDO GRACIELA  
HIDALGO CORONA ALEJANDRO  
VELAZQUEZ AVILA FABRICIO

Director de Tesis: Ing. Juan Manuel Martínez Villalobos



MEXICO, D. F.

JUNIO, 1995

FALLA DE ORIGEN

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

*A mi madre:*

*Por todo el amor y comprensión  
que me ha brindado siempre.*

*Gracias.*

*A mi hermana:*

*Por todo su cariño y apoyo gracias.*

*A Alejandro:*

*Porque desde que te conozco me has dado  
tú amor, comprensión y apoyo para concluir  
esta carrera.*

*Gracias.*

*Graciela.*

*A la memoria de mi papá que siempre  
quizo que lograra esta meta.*

*A ti mamá, que siempre me has apoyado  
y tolerado tantas cosas. Por todo lo que  
me has dado, gracias.*

*A mis hermanos, que siempre me han  
brindado su apoyo, cariño y ayuda.*

*Gracias*

*A Graciela que me ha dado tanto.*

*Gracias*

*Gracias a toda mi familia.*

*Alejandro.*

*A ti Yamina, que sin tu apoyo, paciencia  
y comprensión no lo hubiese logrado.*

*A mi madre y padre, pues me dieron todo,  
me impulsaron y sobre todo me dieron  
su confianza.*

*A mis hermanos, Lizúli, Deyanira y Bricio.*

*Fabricio.*

*A todos nuestros maestros por sus  
enseñanzas y amistad recibida.*

*Gracias.*

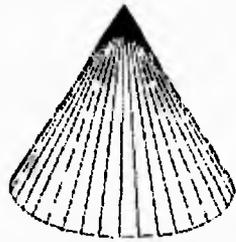
*Con afecto a todos nuestros amigos, compañeros  
y todas aquellas personas que nos han  
apoyado en algún momento de nuestra vida.*

*A nuestro director de tesis, maestro y amigo  
Ing. Juan Manuel Martínez Villalobos  
quien sin su colaboración no habría sido  
posible este trabajo.*

*Gracias.*

*Agradecemos a la Universidad y a  
la Facultad de Ingeniería que  
nos permitió comenzar el camino.*

**INDICE**

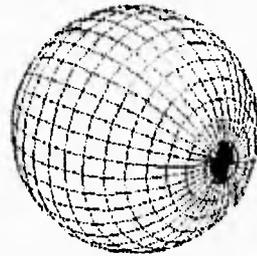


## INDICE

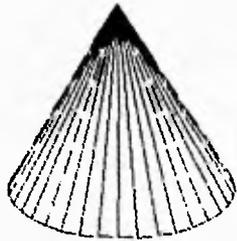
PROLOGO..	. i
INTRODUCCION.	. v
CAPITULO 1. ANTECEDENTES.	. 1
1.1.- Diferentes Modos Gráficos..	. 2
1.2.- Interpretación de los Gráficos por el Cerebro Humano..	. 2
1.3.- Funciones Gráficas.	. 3
1.4.- Funciones Gráficas Avanzadas.	. 4
1.5.- Métodos del Software para dibujar sobre la pantalla..	. 5
1.6.- Cómo Seleccionar el Color..	. 5
1.7.- Historia de la Graficación.	. 7
CAPITULO 2. ANALISIS.	. 9
2.1.- Técnicas de Recopilación de Información.	. 10
2.2.- Análisis de Hardware.	. 11
2.3.- Análisis de Software..	. 16
2.4.- Principios de Geometría Plana y Sólida.	. 20
2.5.- Sistema Propuesto.	. 26
2.6.- Presentación de los Diagramas del Sistema Propuesto..	. 27
CAPITULO 3. DISEÑO DEL SISTEMA.	. 53
3.1.- Técnicas del Diseño.	. 57
3.2.- Diagramas de Estructuras del Sistema Propuesto..	. 59

## **INDICE**

CAPITULO 4. IMPLEMENTACION.	. 87
4.1.- Software y Hardware Seleccionado.	. 88
4.2.- Conceptos de Gráficos 3D.	. 92
4.3.- Descripción Estructurada de Procesos del Sistema.	. 101
4.4.- Diagramas Generales de Acciones y Procesos.	. 108
4.5.- Diagramas de Formas del Sistema.	. 110
CAPITULO 5. CONCLUSIONES	. 119
APENDICE A. MANUAL DE OPERACION DEL USUARIO.	. A-1
APENDICE B. METODOS DE: ORDENAMIENTO Y DE SUPRESION DE CARAS OCULTAS.	. B-1
APENDICE C. DICCIONARIOS.	. C-1
APENDICE D. BIBLIOGRAFIA.	. D-1



**PROLOGO**



**Figura:**  
Espacio cerrado por  
líneas o superficies

---

## PROLOGO

**H**ace tiempo se reconoció en México la importancia estratégica de la tecnología de la información, y se comenzó a utilizar para propiciar el desarrollo de la industria nacional. Así se iniciaron acciones de diferente índole para estimular su evolución, de acuerdo con las condiciones generales del país y el desarrollo de la tecnología.

El impacto de la tecnología computacional ha sido tan amplio que ahora todos los sectores económicos han transformado sus estructuras de producción y comercialización, así como la prestación de servicios públicos y privados, con base en ella. Aunado a esto, la informática se ha introducido en el ámbito social y en todas las actividades cotidianas de las personas.

La revolución de este sector se ha dado en un contexto dinámico y mundial, tanto en lo político como en lo económico y en lo social, otorgando un nuevo papel a la informática, al constituirse como una herramienta de cambio y modernización imprescindible, para enfrentar los retos que implica la economía internacional, para así promover mayor nivel de bienestar en todos los sectores del país.

La actual situación económica del país no es muy buena y únicamente podrá ser superada si se trabaja con eficiencia y asumiendo nuestras responsabilidades individuales. Es por ello que ante la enorme cantidad de información que se genera día con día en todos los sectores del país, resulta de vital importancia que todo aquel que desempeñe un papel en la vida productiva del país esté en contacto con las computadoras.

Una de las principales ventajas que se han dado al utilizar computadoras es que éstas, al ir evolucionando, han permitido el manejo de información en cantidades cada vez mayores y a velocidades difícilmente imaginables hasta hace pocas décadas.

Para poder tomar decisiones inmediatas que puedan resolver una situación en particular, es necesario hacer una valoración de la información que una computadora arroja. Buscando un mecanismo capaz de mostrar toda la información requerida de una manera fácilmente apreciable y sin necesidad de hacer un análisis matemático complejo es como surge la graficación, la cual ha permitido agrupar, resumir, analizar y mostrar gran cantidad de información de una manera simple para ser analizada y poder determinar como se desarrolla

## **PROLOGO**

dicho proceso. Así la graficación ha llegado ha ser una área importante dentro de la ciencia de la computación al grado de tratar de incorporar todos los sistemas actuales en ambientes gráficos y que al procesar la información se obtengan automáticamente resultados gráficos para su fácil comprensión.

Al mismo tiempo la graficación, al ir evolucionando, ha permitido contar con la animación por computadora. Esto es, no sólo es posible observar el resultado de procesos sino también mostrar como se desarrollan en cualquier punto y de igual manera, si se cuenta con la información suficiente simularlos y observar gráficamente su desarrollo. Esto es útil para pronosticar el comportamiento de algunos eventos, siendo más fácil tomar decisiones acertadas sin necesidad de sacrificar recursos ni exponerse a riesgos innecesarios.

Gracias a la graficación por computadora es posible implementar cualquier tipo de proyecto, siendo la única limitante la capacidad del equipo con el que se cuente, pudiendo así desarrollar toda la capacidad creativa al máximo y crear sistemas útiles para casi todas las actividades del hombre como la medicina , el arte, la educación, la diversión, etc. Estos sistemas, por su facilidad de manejo al utilizar gráficos, hacen posible que cualquier persona con conocimientos mínimos de computación pueda aprovechar al máximo la capacidad de éstos sin necesidad de entrenamientos largos ni costosos.

Dada la importancia que en esta época ha tenido la graficación por computadora y por la gran cantidad de paquetes que se encuentran en el mercado para graficar y manejar gráficos, se vió la conveniencia de desarrollar una herramienta con la cual sea posible la manipulación de figuras geométricas básicas en tercera dimensión, y de igual forma poder manipularlas variando sus propiedades principales como son la forma, el tamaño, el color, su posición dentro de un espacio y poderlas conjuntar con otras figuras para crear figuras más complejas y de esta manera apoyar el aprendizaje de la geometría sólida a un nivel básico.

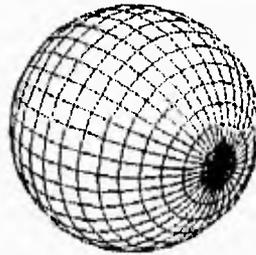
Debido a la gran diversidad de figuras geométricas que existen, el editor de figuras geométricas en tres dimensiones será capaz de mostrar figuras básicas como son : líneas, triángulos, cubos, esferas, piránides, etc., con los cuales será posible formar casi todas las formas geométricas que existen.

El objetivo de la creación de este editor de figuras geométricas es: aportar una herramienta sencilla de apoyo en el diseño asistido por computadora,

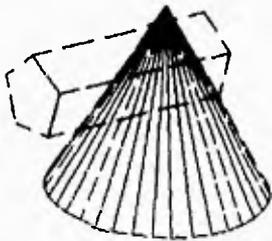
## **PROLOGO**

---

utilizando algoritmos sencillos de graficación y a su vez mostrar la versatilidad, eficiencia y alcances de la graficación por computadora con pocos recursos y facilitar el aprendizaje de la geometría básica; también se pretende con este sistema obtener una herramienta útil en el procesamiento de textos, pues se podrá generar una serie ilimitada de diseños para posteriormente ser utilizados en varios de los procesadores de textos existentes en el mercado.



## INTRODUCCION



### **Geometría:**

**Disciplina matemática que tiene por objeto el estudio riguroso del espacio y de las formas que en él se pueden imaginar.**

---

## INTRODUCCION

**L**a graficación por computadora es un campo de reciente desarrollo y el cual hoy en día está presente en muchas de las aplicaciones de los usuarios de computadoras personales, en la automatización de oficinas, la medicina, la publicidad electrónica, los procesos de control, el arte, las simulaciones y animación para proyectos científicos y en el entretenimiento, por mencionar algunos de los múltiples campos de la que es parte fundamental. De este modo nace el interés por desarrollar una herramienta de utilidad para esta área de la computación.

En este trabajo se presentan las fases del desarrollo de sistemas, que lleva a realizar un análisis, diseño, e implementación; necesarios para el Sistema de Edición de Figuras Geométricas Tridimensionales.

En el capítulo 1 se da una breve reseña de lo que hasta hoy en día ha sido la graficación por computadora, se da un panorama general de algunos antecedentes y conceptos previos necesarios para tener una visión más clara en el momento de hacer un análisis de la problemática a resolver.

En el capítulo 2 se presenta la fase del Análisis, que es la encargada de estudiar dentro de la problemática, la organización, la identificación del flujo de información o en otras palabras los datos.

En esta fase se identifica el problema que nos ocupa, se determinan los recursos con los que se cuenta, la forma como se maneja la información, se identifican las entradas de información disponibles y las salidas de las mismas, para posteriormente presentar un diagrama que muestra el flujo que tiene la información a lo largo del sistema.

También se realiza un análisis de los principios básicos de la geometría plana y sólida.

En el capítulo 3 se determinan los módulos o procesos que componen al sistema y la forma como interactúan entre sí. Los diagramas que pueden mostrar estos procedimientos son los diagramas de estructura, que son representaciones jerárquicas de todos los procedimientos interconectados entre sí, de un sistema.

El capítulo 4 está relacionado con la implementación, se establece el Software y Hardware a utilizar y se realiza la implementación de los

## **INTRODUCCION**

módulos que componen al sistema. Se presentan diagramas que muestran la manera como se interrelacionan los eventos y procedimientos de cada forma.

También se presentan y analizan algunas de las herramientas para la generación de figuras tridimensionales y un pseudocódigo de los algoritmos de mayor utilidad para la graficación por computadora.

En el capítulo 5 se presentan las conclusiones obtenidas del sistema, se discute si los objetivos fijados al principio de este trabajo fueron alcanzados, si todos los requerimientos fueron satisfechos, si el sistema es lo suficientemente accesible a cualquier tipo de usuario, las deficiencias que pudiera presentar y se analizan los resultados obtenidos.

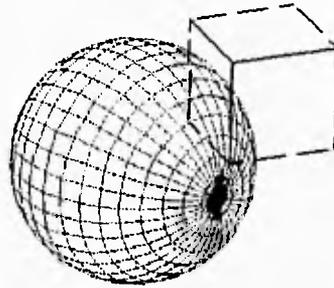
Después de estos capítulos se presenta los siguientes apéndices:

A) **MANUAL DE OPERACION DEL USUARIO.** El manual del usuario sirve para la instalación, operación y utilización del sistema, para que sea fácil su manejo, también se dan a conocer las principales pantallas que el sistema tiene, con la finalidad de que no se tenga ninguna duda.

B) **METODOS DE: ORDENAMIENTO Y SUPRESION DE CARAS OCULTAS:** En él se presentan algunos métodos de ordenamiento de datos y los métodos más usuales en la graficación por computadora para la supresión de caras ocultas.

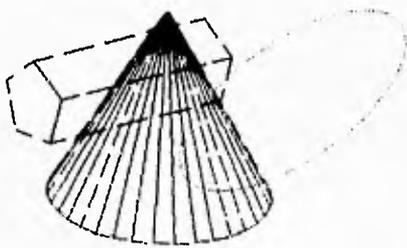
C) **DICCIONARIOS:** En este apéndice se presentan algunos términos técnicos que aparecen con frecuencia a lo largo de este trabajo y pudieran crear alguna confusión, por lo que consideramos conveniente explicar. También se presenta un diccionario de las variables utilizadas en los diagramas de estructura.

D) **BIBLIOGRAFIA:** Finalmente se presentará la bibliografía que se utilizó como material de apoyo a lo largo del desarrollo del sistema.



## CAPITULO 1

### ANTECEDENTES



Punto:  
Figura geométrica  
sin dimensiones.

---

## ANTECEDENTES

**E**s un hecho que la mayor parte de la información captada por los animales más evolucionados entra en el cerebro a través de los ojos. El cerebro humano, de hecho, dedica una gran porción de sus células simplemente a interpretar esa información.

Quizá por ello el ser humano siempre ha tenido la necesidad de expresarse gráficamente: basta ver a un niño pequeño que encuentra un lápiz y cualquier superficie, o a nuestros antepasados en Altamira o Lascaux.

Entre el ocre usado para pintar bisontes en las cavernas y la tecnología digital se encuentran cientos de medios gráficos que incluyen (aparte de las innumerables técnicas de dibujo y pintura) la fotografía, el cine y la televisión.

### 1.1.- DIFERENTES MODOS GRAFICOS

Actualmente los gráficos son un componente vital en la programación de computadoras. Ya que la mayoría de los programas, los incluyen para una mejor comprensión.

La gran mayoría de las computadoras personales poseen un adaptador gráfico. Por otro lado, un gran número de usuarios que ya están familiarizados con los gráficos esperan los programas donde se utilicen interfaces gráficas; estas personas encontraron que al usar cualquier sistema o paquete que se maneje a través de gráficos tienen un gran ahorro de tiempo para aprender a usarlo y para ejecutar diversas tareas. Sin embargo, se toma conciencia de que los gráficos no son un sustituto de los textos, el uso de los gráficos sirve junto con un texto para mejorar la ejecución y presentación de cualquier sistema.

### 1.2.- INTERPRETACION DE LOS GRAFICOS POR EL CEREBRO HUMANO

Usando la combinación de gráficos y texto en la mayoría de los sistemas, se puede mejorar en gran medida la calidad de comunicación entre el sistema y el usuario. Esta mejora se debe a la forma en que trabaja el cerebro humano.

El hemisferio izquierdo del cerebro tiene la función de interpretar material en forma de texto mientras que el hemisferio derecho tiene la habilidad de interpretar objetos, es decir, posee una interpretación visual. Investigaciones médicas concluyeron que el hemisferio izquierdo trabaja con partes de

## **ANTECEDENTES**

cualquier cantidad y secuencias de partes, por otro lado, la parte derecha trabaja con cantidades completas, es decir, con objetos enteros.

Una porción significativa de la mente humana no es capaz de retener solo texto con gran facilidad, por lo que si se le ayuda introduciendo gráficos se incrementa el canal de comunicación del sistema con el entendimiento del cerebro humano.

Recientes estudios científicos demuestran que la comprensión de las ideas se incrementa cuando alguna secuencia de imágenes es adherida a un sistema. La conclusión es que si se requiere de buena comunicación del usuario con el sistema es necesario usar gráficos.

### **1.3.- FUNCIONES GRAFICAS**

En general, las funciones que se requieren en una PC para generar rutinas gráficas esencialmente son cinco y se listan a continuación:

A) Poder establecer un modo gráfico en la pantalla: Los adaptadores gráficos VGA, EGA y CGA pueden soportar una gran variedad de modos gráficos.

B) Limpiar la pantalla.

C) Dibujar líneas: La habilidad para dibujar líneas simples es muy importante ya que de aquí se pueden hacer polilíneas, curvas, círculos, pequeños segmento, etc. Algo que es muy importante antes de dibujar la línea es que se debe de definir su color y establecer el punto de comienzo de la misma.

D) Poder rellenar áreas con color: Estas funciones gráficas operan definiendo un punto para empezar a rellenar el área con cierto color hasta que otro es encontrado definiendo así el límite de la figura.

E) La habilidad para pasar material alfanumérico a gráfico.

F) La habilidad para regresar al modo texto: Un programa creado para desplegar gráficos debe tener la habilidad de regresar a modo texto una vez que se halla completado la tarea gráfica.

## CAPITULO 1

### **1.4.- FUNCIONES GRAFICAS AVANZADAS.**

En una PC se tienen las siguientes funciones gráficas avanzadas:

A) Medio tono (Halftoning): En la dibujo por computadora se utiliza la técnica de medio tono para conseguir un efecto de sombra gradual en las figuras. Si sólo se rellenan la mitad de los pixeles, con una función de relleno de áreas se creará una sombra del 50%. De la misma manera si se rellena una cuarta parte se creará una sombra del 25%. De esta forma se puede buscar la manera de crear nuevos colores en base a los colores estándar que se tienen dependiendo del adaptador gráfico. En la figura 1.1 se ilustran diferentes niveles de sombra que pueden ser creados en este proceso.

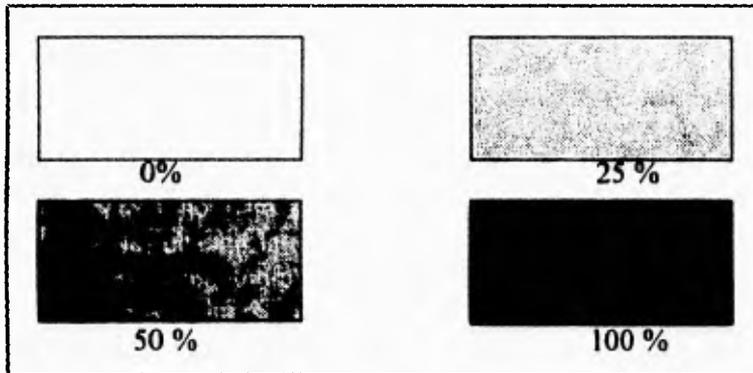


Fig. 1.1

B) Línea vibrando (line dithering): Esta técnica, también es llamada line style (tipo de línea) y se refiere al estilo de los puntos usados para dibujar una línea (punteada, continua, etc.).

C) Copiando páginas (page copying): En esta técnica, cierta imagen colocada en una página pasa de una forma muy rápida a otra página limpia o en blanco, generalmente es así como funcionan la mayoría de los programas de animación. Al movimiento de la imagen de una página a otra se le llama page copying.

Este método se puede utilizar cuando se tienen adaptadores gráficos EGA y VGA. En caso de que se cuente con un CGA se podría hacer una simulación de páginas gráficas con ayuda de la memoria RAM.

D) Habilidad para dibujar gráficas en páginas escondidas: Como los adaptadores VGA y EGA ofrecen numerosas páginas gráficas no es necesario

## **ANTECEDENTES**

hacer una gráfica en la misma página que está siendo desplegada sobre el monitor.

E) Gráficas de bloque (block graphics): La animación Bitblt (transferencia de bloques de bits) utiliza la técnica de gráficas de bloque para poder mover entidades gráficas a través de la pantalla. En algunos lenguajes de programación ésto es conocido como graphics array animation.

La animación Bitblt utiliza la característica del hardware que mueve un bloque rectangular de bits desde la memoria principal a la memoria del monitor a alta velocidad. La tecnología bitblt mejora el rendimiento de los computadores utilizados en animación en tiempo real y modelado sobre pantalla. Este método es muy rápido y se puede llegar a una velocidad de 30 fps (páginas por segundo).

### **1.5.- METODOS DEL SOFTWARE PARA DIBUJAR SOBRE LA PANTALLA**

Básicamente existen dos métodos para dibujar sobre la pantalla, el método de las coordenadas dinámicas y el método de las coordenadas de la base de datos. Ambos métodos utilizan las coordenadas XY.

El método de las coordenadas dinámicas permite ahorro de memoria ya que las coordenadas xy son calculadas durante la ejecución del programa, lo que origina consumo de tiempo.

Cuando se utiliza el método de las coordenadas de bases de datos las coordenadas xy son almacenadas manualmente en algún arreglo o base de datos, lo que implica un consumo de memoria.

### **1.6.- COMO SELECCIONAR EL COLOR**

Los adaptadores gráficos VGA, EGA, CGA, etc. encontrados en las computadoras personales IBM y compatibles son capaces de generar una amplia gama de colores, y esta característica se puede incrementar al conocer y aplicar la terminología de los colores que se presenta a continuación.

#### **1.6.1.- TERMINOLOGIA DE COLORES**

Los colores cromáticos son aquellos que tienen su propio matiz (o tinte), por ejemplo los que se muestran en la figura 1.2.

## CAPITULO 1

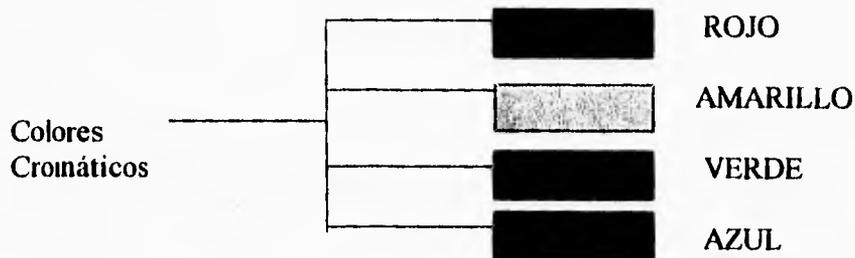


Fig. 1.2

Por otro lado los colores acromáticos son aquellos neutros, por ejemplo blanco, gris y negro. Estos no tienen su propio matiz (fig. 1.3).

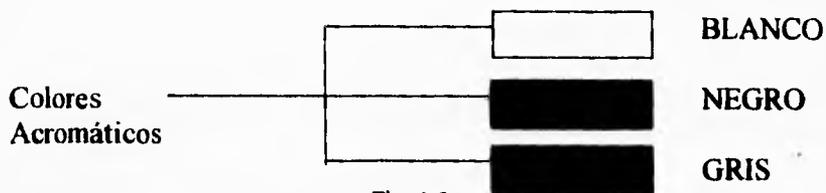


Fig. 1.3

Un punto importante es conocer la diferencia entre matiz, valor y cromo. A continuación se mencionan las diferencias de cada una.

Por medio de un matiz podemos distinguir un color cromático de uno que no lo es.

Los matices se encuentran divididos en dos categorías:

- a) Matiz cálido.- Que son el rojo, amarillo y anaranjado.
- b) Matiz fresco.- Que son el verde y el azul.

Por otro lado, el valor se refiere al brillo o claridad que presentan los colores.

El término cromo, se refiere a la relativa pureza del color. Por ejemplo, el anaranjado tiene un cromo muy fuerte. El cromo también es llamado intensidad o saturación.

### 1.6.2.- DIFERENCIAS ENTRE: SOMBRA, TONOS Y TINTES.

La sombra, es formada por la combinación de cualquier color puro con el negro. Por ejemplo el café, marrón y olivo se consideran sombras.

## **ANTECEDENTES**

En lo que se refiere a los tonos, éstos se definen como la mezcla de cualquier color puro con el gris. El beige y el bronce son tonos.

Por último, los tintes son formados de la combinación de cualquier color puro con el blanco. Es un ejemplo de tintes el durazno.

Las sombras, tonos y tintes creados de un mismo matiz cálido siempre resultan ser diferentes entre sí. Por otro lado sombras, tonos y tintes de un matiz fresco esencialmente tienen el mismo tipo de colores.

### **1.6.3.- ESCALAS DE COLORES.**

Usando la física de la luz, la cual es utilizada por el monitor, los matices básicos son combinados en círculos de colores para crear un amplio rango de éstos, los cuales pueden ser reconocidos por el ojo humano.

Un simple matiz al ser combinado con diferentes cantidades de blanco y negro produce un ilimitado rango de sombras, tonos y tintes. La mezcla entre el matiz puro y el blanco forman la escala del tinte. La mezcla entre el matiz y el negro forman la escala de la sombra. La mezcla entre el blanco puro y el negro producen el gris. Las combinaciones formadas por el matiz puro y diferentes tonalidades de gris son llamadas tonos.

El ojo humano es capaz de distinguir nueve sombras distintas entre blanco y negro puros.

### **1.7.- HISTORIA DE LA GRAFICACION**

La graficación por computadora ha tenido un rápido crecimiento debido a la creatividad del hombre, a los actuales medios de almacenamiento y manipulación de objetos e imágenes. En la actualidad el control de todo lo tienen los usuarios, ellos deciden el contenido, la estructura y apariencia de los objetos y las entradas de información que se necesitan para el despliegue de imágenes.

Muchos años de estudio y de desarrollo fueron necesarios para todos los avances que ahora tenemos en la graficación por computadora. Antes era un campo pequeño, especializado para algunos áreas científicas, debido a que el hardware era caro y los programas basados en gráficos fáciles de utilizar y de bajo precio eran pocos. A principios de 1950 se desarrolló el primer dispositivo de despliegue que únicamente servía para generar y desplegar gráficas muy simples. Este dispositivo fue hecho de un tubo de rayos catódicos (CRT) muy

## **CAPITULO 1**

similar a los utilizados en los televisores. Hasta finales de la década se dio un incremento muy rápido de la graficación por computadora.

Los avances en la graficación por computadora siguieron y un hecho muy importante fue un artículo publicado en 1962 de Ivan F. Sutherland, que provee de muchos elementos importantes que permiten hacer de la graficación por computadora un recurso viable, muy usual y excitante. A mediados de los 60's hubo grandes proyectos de desarrollo en la graficación por computadora realizados por General Motors y Bell Telephone Laboratories; la edad de oro de la graficación por computadora había empezado.

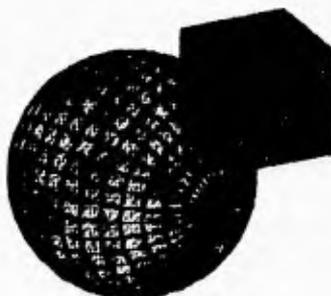
Si se considera a los 60's como la década de los grandes desarrollos de la graficación por computadora, los 70's es la década en la que se empieza a recoger los frutos. Los dispositivos de despliegue gráfico son ahora usados para los propósitos de educación en la enseñanza elemental en algunos países.

Hasta principios de los 80's con el uso de las PC's (computadoras personales) se popularizó el uso de bitmap gráfico para la interacción entre el usuario y la computadora. Un bitmap es una zona en la memoria de la computadora con la forma de un arreglo rectangular de unos y ceros que representa una figura en la pantalla. Las interfaces gráficas atrajeron a millones de usuarios pues era posible tener el fácil control de sus aplicaciones, programas, así como procesadores de palabras, bases de datos y programas de dibujo a su alcance.

El concepto de desktop empezó a ser un método popular para la organización del espacio en la pantalla. Y con conceptos de administrador de ventanas o "window manager", el usuario puede crear, cambiar la posición y el tamaño de áreas de pantalla llamadas windows. Algunas veces en este método del escritorio se representan algunas funciones por iconos que no solamente representan un archivo de datos y programas de aplicación, sino también objetos de oficina que realizan la operación equivalente de la vida real al lenguaje de la computadora. La manipulación directa de objetos vía "señalar y seleccionar" permite a los usuarios la activación de los programas u objetos, del menú de la pantalla para hacer sus cambios necesarios.

Aún cuando la gente no utiliza las computadoras en su trabajo diario se topan con la computación gráfica en todas partes y es muy común en la vida diaria de cualquier persona la posibilidad de la visualización en dos dimensiones (2D) y tres dimensiones (3D).

No es de extrañarse que desde que las computadoras tuvieron capacidad de mostrar gráficos, los primeros y más exitosos programas fuesen los de diseño e ilustración ya que hasta aquéllos que tienen una regular o nula capacidad con un lápiz y un papel pueden realizar en ellos trabajos decorosos.



## CAPITULO 2

### ANALISIS



Línea:

Extensión considerada en una sola de sus tres dimensiones: la longitud

---

## ANALISIS

**E**l objetivo de esta fase es obtener una definición clara del sistema que será desarrollado e implantado. En este capítulo se deben analizar con detalle todos y cada uno de los aspectos que han de contemplarse con el sistema, buscando su justificación de acuerdo con los objetivos.

La primer actividad que se planea es determinar los objetivos específicos del sistema, pues una vez definidos éstos, se procederá a identificar las entradas, salidas y archivos del mismo, así como su flujo de datos, los diferentes procesos de transformación y los principales elementos de datos que se van a manejar con el sistema.

Podemos definir al análisis como el proceso para transformar una cadena de información acerca de las operaciones actuales y de nuevos requerimientos para obtener una descripción ordenada y rigurosa de un sistema que será construido. En el contexto del ciclo de vida de los sistemas, la fase de análisis se presenta al inicio, donde una vez establecidos los objetivos y requerimientos del sistema, se procede a la generación de especificaciones del sistema que será creado.

La fase de análisis, junto con la fase de diseño, se distingue también por la alta interactividad que debe existir con los usuarios, por lo que es indispensable utilizar técnicas que ayuden a lograr los objetivos deseados.

### 2.1.- TECNICAS DE RECOPIACION DE INFORMACION

Para diseñar adecuadamente un sistema se requiere efectuar un trabajo previo de recopilación de información. Esta actividad es necesaria para definir los objetivos del sistema, identificar sus restricciones y limitaciones, analizar y evaluar al sistema en operación y definir los nuevos requerimientos de información. Las técnicas que se utilicen para obtener esta información deben ser lo suficientemente objetivas para lograr las metas establecidas y una de las principales técnicas es el uso de diagramas de flujo de datos.

#### 2.1.1.- Diagrama de Flujo de Datos.

El diagrama de flujo de datos es una técnica que tiene como finalidad interpretar gráficamente las entradas, salidas y archivos de un sistema, pasando por todos sus procesos de transformación.

Este diagrama tiene tres atributos principales:

- 1.- Puede representar el flujo de información de cualquier sistema
- 2.- Todos los procesos de transformación son representados visualmente con todas sus entradas y salidas.
- 3.- Enfatizar principalmente en flujo de datos más que en el control del sistema.

Un diagrama de flujo de datos (DFD) es una herramienta para modelar un sistema sin importar su grado de automatización. Los aspectos primordiales que deben tomarse en cuenta al elaborar estos diagramas son los siguientes:

- 1.- Identificar las entradas externas implicadas
- 2.- Identificar la entradas y salidas.
- 3.- Identificar los procesos
- 4.- Depuración del diagrama.

Conforme la información se mueve a través del software, se modifica mediante una serie de transformaciones. Un diagrama de flujos de datos, es una técnica gráfica que describe el flujo de información y las transformaciones que se aplican a los datos conforme se mueven de la entrada a la salida. Son diagramas que se utilizan para describir un diseño de sistemas de alto nivel; muestran cómo se transforman los datos al pasar de un componente del sistema a otro. Estos diagramas documentan cómo los datos de entrada se transforman en datos de salida, donde cada etapa del diagrama representa una transformación diferente.

A continuación se procederá a realizar un análisis de los requerimientos de Software y Hardware, así como de las bases teóricas de la geometría que se utilizará en el sistema, y los diagramas de flujo de datos correspondientes se muestran al final del capítulo.

## **2.2.- ANALISIS DE HARDWARE**

### **2.2.1 - Procesadores.**

A principios de los 70 se desarrolló el primer procesador de 8 bits llamado 8008. En 1974 se introduce el 8080 que fue el primer procesador comercial de

## **CAPITULO 2**

8 bits. Para 1978 hicieron su aparición el 8086 y el 8088 y este último fue el procesador utilizado en las computadoras denominadas XT.

El 8088 es un procesador 8/16, es decir tiene un bus de datos externo de 8 bits y un bus de datos interno de 16 bits. El 8086 es un procesador 16/16. Respecto al bus de direcciones, ambos son de 20 bits; esto les permite acceder hasta un megabyte ( $2^{20}$ ) de memoria RAM. El 80186 no ofreció grandes ventajas sobre el 8086 y casi no fue comercializado.

El 80286 o simplemente 286 es un procesador 16/16 con un bus de direcciones de 24 bits con los cuales puede direccionar hasta 16 megabytes ( $2^{24}$ ) de RAM. A las computadoras que utilizan este procesador se les denomina AT.

El 386 es un procesador 32/32 con bus de direcciones de 32 bits y puede acceder hasta 4 gigabytes de memoria. A este procesador también se le conoce como 386DX.

Existe un hermano menor del 386DX llamado 386SX, es un chip 16/32 con un bus de direcciones de 24 bits.

El penúltimo procesador que apareció es el 486. Existen las versiones 486SX, 486DX, 486DX2 y 486DX4. Los procesadores 486 son chips 32/32 con un bus de direcciones de 32 bits. La diferencia entre el SX y el DX es que este último tiene un coprocesador matemático incorporado. Cabe decir que esta diferencia no es aplicable para los procesadores 386.

DX2 significa que el procesador corre internamente con el doble de velocidad interna. DX4 indica que el procesador tiene el triple de velocidad interna.

El último procesador que apareció es el Pentium; se le dio este nombre, y no 586, por cuestiones de derecho de uso.

El Pentium es un procesador 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.

A la fecha prácticamente no existen aplicaciones que realmente utilicen al procesador en modo de 64 bits, pero éste 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 lo permita.

### **2.2.2.- Tarjetas de Video.**

La pantalla de la computadora es la interfaz de comunicación entre la máquina y el hombre; es importante recordar que, para que se despliegue una

imagen en la pantalla, es necesario un elemento que convierta los datos proporcionados por el procesador en resultados legibles por el hombre.

Los dispositivos de despliegue gráficos que se utilizan actualmente son extremadamente sencillos en su construcción. Estos constan de tres componentes:

- ① Una memoria digital o un buffer de memoria en el cual se almacena la imagen a desplegarse, ésta es almacenada como una matriz de valores de intensidad
- ② Un monitor con un cinescopio similar al de una televisión, y
- ③ Una simple interfaz llamada controlador de despliegue.

La imagen puede ser desplegada por el monitor, más de 30 veces por segundo.

A esta interfaz se le denomina tarjeta, adaptador, placa de video o controlador gráfico. El adaptador de video es una tarjeta que convierte los comandos de visualización de la computadora en imágenes sobre la pantalla. El adaptador de video funciona como un transductor, convirtiendo las señales de la computadora en señales de voltaje que controlarán el barrido de la pantalla de despliegue. Toda tarjeta de video tiene tres partes fundamentales:

A) Un circuito integrado conocido como CONTROLADOR GRAFICO que contiene los comandos que regulan la operación de la pantalla de despliegue.

B) Una memoria RAM DE VIDEO ESTATICA especialmente diseñada que se usa para contener y transferir una imagen a la pantalla de video; del tamaño de esta memoria depende la resolución del monitor y el número de colores máximos representables.

C) Una memoria ROM DE VIDEO que es donde están contenidos los juegos de caracteres, tanto alfanuméricos, como gráficos. Sirve para traducir a pixeles la información de la memoria RAM.

Algunos adaptadores de video tienen un procesador gráfico o coprocesador de video; que libera al procesador central de la computadora del control de la representación gráfica aumentando la velocidad .

Se puede acelerar el video de dos maneras: en software o en hardware.

La solución en software significa escribir un manejador optimizado para el entorno. Para acelerar el video por hardware es muy importante la clasificación

## **CAPITULO 2**

en la que puede caer la tarjeta de video. Existen tres principales categorías: buffers de imagen, aceleradores gráficos y coprocesadores.

A partir de la década de los 80 se empezaron a utilizar dos tipos de monitores: Los monitores monocromáticos y los de color.

Los adaptadores de video iniciales fueron MDA (Monochrome Display Adapter "Adaptador de Despliegue Monocromático") que solo puede visualizar caracteres en un sólo color y tiene una resolución limitada a 80 x 25, y el CGA (Color Graphics Adapter "Adaptador Gráfico de Color") que muestra textos y gráficos como una secuencia finita de puntos y hasta en 16 colores; posteriormente se diseñaron otros, entre los que destaca el adaptador gráfico Hércules por su bajo costo y presenta la ventaja de su modo de alta resolución (720 x 348 pixeles), solo puede utilizarse en combinación con otro adaptador de video y únicamente se puede conectar a los monitores monocromáticos.

El CGA fue diseñado para permitir compatibilidad con televisores y monitores compuestos, permiten trabajar tanto en modo texto como en modo gráfico, posee 3 modos gráficos (alta, media y baja resolución), su posibilidad de utilización de colores es sumamente limitada, este adaptador sí puede trabajar en combinación con otro como es el caso del MDA o el EGA. El modo texto tiene disponibles 2 anchos: 40 y 80 columnas. Todos los modos texto tienen 25 líneas de caracteres en la pantalla.

Las TARJETAS DUALES son una mezcla de la tarjeta MDA y CGA, se trata de interfaces de video compuestos por la integración en una única tarjeta de ambos adaptadores. Se puede seleccionar un modo u otro por medio de un mecanismo de control.

El EGA (Enhanced Graphics Adapter "Adaptador Gráfico Mejorado") combina las características del MDA, el CGA y el adaptador HERCULES, adicionando nuevos modos de video que pueden ser utilizados con monitores de color de alta calidad. Su principal ventaja es la de poder utilizarse directamente para controlar cualquier monitor monocromático o de color.

El VGA (Video Graphics Array "Arreglo Gráfico de Video) apareció en 1987 con la introducción de las computadoras PS/2 de IBM. Es una versión ampliada y mejorada del CGA. La diferencia entre EGA y VGA es que los VGA son analógicos mientras que los EGA son digitales y por el hecho de ser

digitales, se necesita una memoria super-dotada. El VGA puede manejar varias páginas gráficas, posee la paleta de colores más variada.

La tarjeta VGA mejorada (SUPER VGA) tiene una resolución de 800 x 600 pixeles y 256 colores, por lo que se necesita una RAM de 512 Kbytes, manejando también una resolución de 1024 x 768 pixeles en 16 colores. La administración de esta memoria adicional aumenta la carga en el CPU.

La tarjeta Extra VGA es capaz de proporcionar una resolución de 1024 x 768 pixeles en 256 colores, por lo que tiene una memoria de video de 1 MByte.

Los coprocesadores son programables mientras que los aceleradores tienen una funcionalidad fija.

En lugar de usar el CPU de la computadora para calcular la información para cada pixel de la imagen en la pantalla, el CPU le envía a la tarjeta de video un juego de comandos para funciones gráficas; el coprocesador interpreta las funciones y las convierte en una imagen de bits. La principal ventaja es que el procesador se libera mucho antes de las labores de representación de imagen. Otra estrategia es usar aceleradores gráficos que tienen funciones fijas lo que hace que aumente la velocidad. Los distintos coprocesadores de gráficos tienen sus propios juegos de comandos, que varían en complejidad y sofisticación. Existen adaptadores programables que son más flexibles que los aceleradores gráficos.

Las tarjetas inteligentes son más rápidas especialmente cuando se trata de más resolución y color. Pero en los modos estándar VGA y SVGA de 16 colores, las tarjetas SVGA son más veloces. Mientras el acelerador puede destacar en una aplicación, puede retrasarse en otra.

### **2.2.3.- Monitores.**

Los sistemas de computación pueden adaptarse a aplicaciones de los gráficos en varias formas, conforme a los recursos del hardware y software que se dispongan. En la sección anterior se dio una breve explicación de lo que son tarjetas de video pero ahora corresponde hablar un poco de los dispositivos de despliegue principales de salida que son utilizados. Estos dispositivos son los monitores.

La operación de muchos monitores de video se basa en el diseño estándar del tubo de rayos catódicos (CRT), pero existen otras tecnologías.

Se entiende por resolución el número de puntos y líneas que una tarjeta gráfica es capaz de mostrar en la pantalla. Uno de los inconvenientes que puede presentar la resolución trabajando con programas de gráficos es que cuando no

## **CAPITULO 2**

alcanza el tamaño adecuado no se puede representar todo el dibujo al mismo tiempo en la pantalla, y es necesario utilizar al recurso de desplazar el dibujo por la pantalla, hasta situarse en la zona deseada. La resolución está relacionada tanto con el tipo de adaptador gráfico que esté instalado en la computadora como con el monitor. Así, la resolución final del sistema de video será la más baja de las dos resoluciones.

Existen diversos tipos de monitores dependiendo de la tecnología utilizada entre los más importantes están los siguientes:

- ★ Monitores de rastreo al azar y de rastreo con rastreador
- ★ Monitores CRT de color
- ★ Tubo de almacenamiento con vista directa
- ★ Monitores de LED y LCD

### **2.3.- ANALISIS DE SOFTWARE**

#### **2.3.1.- Sistemas Operativos.**

Un Sistema Operativo es la base de todo sistema, pero en ocasiones no se le presta la suficiente atención que se merece por ello es necesario hablar un poco de él. Un Sistema Operativo (SO) se puede ver como una colección organizada de software que extiende al hardware y que consta de rutinas de control para operar una computadora y proporcionar un entorno para la ejecución de programas. Algunos programas confían en las facilidades que proporciona el Sistema Operativo para conseguir acceder a los recursos del sistema de la computadora, tales como archivos y dispositivos de entrada/salida (E/S). Los programas invocan los servicios del Sistema Operativo mediante las llamadas a él, además, los usuarios pueden interactuar con el Sistema Operativo directamente mediante las ordenes de éste. En ambos casos, el Sistema Operativo actúa como una interfaz entre los usuarios y el hardware de un sistema de computadoras.

El rango y las extensiones de los servicios proporcionados por un Sistema Operativo dependen de algunos factores. Entre otras cosas, las necesidades y características del entorno que el Sistema Operativo pretende soportar determinan en gran medida las funciones visibles al usuario. Internamente, un Sistema Operativo actúa como un gestor de los recursos del sistema de computadora, tales como el procesador, la memoria y los dispositivos de E/S.

Un Sistema Operativo puede procesar sus cargas de trabajo en serie o concurrentemente. Esto es, se pueden dedicar los recursos de un sistema de computadora a un programa único hasta completarlo, o se puede reasignar dinámicamente entre una colección de programas activos en diferentes estados de ejecución. Tales Sistemas Operativos son conocidos comúnmente como Sistemas Multiprogramados por su capacidad de ejecutar múltiples programas de manera intercalada. Existen diversas variaciones de Sistemas Operativos en serie y multiprogramados.

### **2.3.2.- Lenguajes de Programación**

En la actualidad el desarrollo de los sistemas de información está orientado a la creación y utilización de gráficos.

Los comandos de programación para desplegar y manipular salidas de gráficas están diseñadas como extensiones de los lenguajes existentes. Un ejemplo de dicho paquete de gráficas es el sistema Plot 10 creado por Tektronics, Inc., para usarse con FORTRAN en sus terminales gráficas. Las funciones básicas de que se dispone en un paquete diseñado para el programador de gráficas incluyen aquellas para la generación de componentes de una figura (líneas rectas, polígonos, circunferencias y otras figuras), fijación de valores de color e intensidad, selección de vistas y aplicación de transformaciones. En cambio, los paquetes de gráficas de aplicación diseñados para no programadores se forman de manera que los usuarios puedan producir gráficas sin preocuparse por la forma en que lo hagan. La interfaz con las rutinas gráficas en estos paquetes les permite establecer comunicación con los programas en términos propios.

A continuación se analizarán los lenguajes de programación.

Pascal.

C++.

Visual Basic.

#### **PASCAL.**

Uno de los rasgos principales de la computadora son las posibilidades gráficas. Mediante los gráficos se pueden crear dibujos, diagramas, textos multifuente o cualquier cosa que se pueda dibujar. Pero el uso de los gráficos requiere bastante más trabajo que el uso del modo texto de la PC. Se tienen que desarrollar métodos para dibujar líneas y caracteres y ser capaz de escalarlos

## **CAPITULO 2**

---

en la perspectiva adecuada. Pascal es un lenguaje de propósitos generales, Por lo que Pascal proporciona un conjunto excepcionalmente rico de rutinas de gráficos que pueden hacer que la programación de gráficos sea mucho más fácil.

Comparado con el modo gráfico, el modo de texto del PC es fácil de usar. Mostrar esta información en la pantalla es tan sencillo como colocar caracteres ASCII en posiciones de memoria específicas. La pantalla de texto está ingeniosamente dividida en 80 columnas y 25 filas, y la computadora ya sabe cómo trazar los caracteres ASCII.

El modo gráfico requiere una orientación completamente diferente: en lugar de bytes de caracteres y atributos, se tienen pixeles que son elementos de dibujo más pequeños del monitor de la computadora. Un carácter en la pantalla de la computadora está formado por muchos pixeles organizados en un patrón que conforma el carácter. En el modo gráfico, se pueden activar pixeles en cualquier parte del monitor. La pantalla de gráficos está formada por pixeles ordenados en líneas horizontales y verticales. Esto es válido para todos los modos gráficos de la computadora; la diferencia principal está en el tamaño del pixel: en el modo de baja resolución de CGA, los pixeles son bastante grandes, por lo que solo caben 320 horizontalmente y 200 verticalmente. En el adaptador VGA, un controlador de video relativamente nuevo con alta resolución, los pixeles son tan pequeños que caben 640 horizontalmente y 480 verticalmente. Cuanto más pequeño es el pixel hay más pixeles por imagen y mayor es la calidad de la imagen gráfica.

Las coordenadas gráficas comienzan en la posición (0,0), que está situada en la esquina superior izquierda de la pantalla. El pixel más a la derecha es el número 319 y el de más abajo es el número 199. Al referirse a los pixeles mediante coordenadas, se puedan activar los pixeles que se deseen en una imagen gráfica, recordando que los sistemas de coordenadas de la pantalla varían de adaptador en adaptador y de modo en modo.

Además en las versiones superiores a 6.0 es posible la programación orientada a objetos.

### **Desventajas**

- Cuando se pretende utilizar un entorno gráfico para Windows, es necesario crear y programar todos los objetos que componen dicho ambiente.
- Casi todos los procedimientos y funciones de gráficos pueden producir errores importantes cuando no se usan adecuadamente.

- Es necesario especificar el adaptador de video en el programa cuando se conoce de antemano el adaptador que se va a usar. Sin embargo, si éste es incorrecto no se ejecutará.
- Si se especifica en la computadora un adaptador de gráficos con un sistema de coordenadas y éste es diferente, el programa no funcionará correctamente. Para hacer que un programa de gráfico sea verdaderamente útil, debe ser capaz de ejecutarse en cualquier adaptador de gráficos e idealmente debería usar, en la medida de lo posible, las características avanzadas de ese adaptador.
- En Pascal se tienen muchas funciones gráficas que facilitan la programación en un sistema, pero para poder realizar una aplicación de Windows requiere mucho código de programación que desvian la atención del objetivo fundamental, que es la graficación por computadora.

### **C++.**

El C++ es un lenguaje orientado a objetos que en la actualidad tiene una gran importancia debido a la capacidad de poder realizar casi todas las tareas requeridas en una computadora. Por ello ofrece una serie de funciones que simplifican la tarea de la graficación por computadora,

Existen funciones relativas a la configuración de la computadora, relativas a la declaración de coordenadas, relativas al uso de las paletas de colores que permiten cambiar los colores asignados a las paletas si el modo gráfico y el hardware lo soportan, rutinas para obtener o poner atributos (como por ejemplo el color de fondo, el color del primer plano, el tipo de línea a dibujar, recubrimiento de áreas), creación de una mascara, visualizar imagenes, visualizar texto, y hasta se puede realizar animación de objetos utilizando C++.

En su versión para Windows, facilita la tarea de crear entornos semejantes a cualquier aplicación de Windows.

### **Desventajas**

- De igual forma que en Pascal se tienen las mismas desventajas que se presentan al utilizar C++.
- En algunas sentencias gráficas se deben de aplicar los factores de corrección, para obtener resultados desde el punto de vista visual correctos.
- Se puede apreciar que ofrece una gran cantidad de ventajas, pero para realizar un programa que sea aplicación de Windows y que cumpla con todas

## **CAPITULO 2**

---

las características que se desean obtener llevaría mucho código de programación extra que haría muy extenso el programa.

- Los requerimientos de hardware que se requieren para instalar el software son muchos y costosos.

### **VISUAL BASIC**

Visual Basic es un lenguaje de programación para desarrollar aplicaciones para ambiente Windows, el cual contiene un conjunto de herramientas gráficas y un intérprete de lenguaje de alto nivel que facilita la realización de cualquier aplicación, con dichas herramientas es posible crear un entorno gráfico en el que intervienen objetos propios del lenguaje, de los cuales el programador sólo utiliza los eventos de dichos objetos, ahorrando con esto la dificultad que presenta crear un objeto.

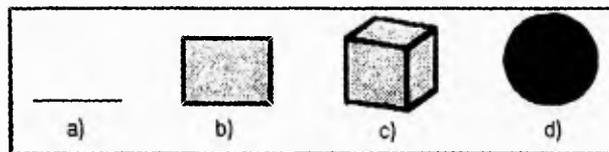
Para utilizar Visual Basic se debe estar familiarizado de manera general con el ambiente de Windows, así como tener una visión general de cómo un lenguaje de programación de alto nivel interactúa con Windows, nociones de programación estructurada y BASIC.

#### **Desventajas**

- Tiene un número limitado de objetos disponibles a los cuales el programador tiene acceso.
- Algunas funciones básicas que son de fácil acceso en otros lenguajes, en éste es necesario acceder a Windows o hacer uso de las funciones de C++.

### **2.4.- PRINCIPIOS DE GEOMETRIA PLANA Y SOLIDA**

En la geometría plana las figuras están constituidas por puntos y líneas. Estas son llamadas figuras planas dado que todos los puntos y líneas de cada figura se encuentran en el mismo plano. En la geometría sólida todos los cuerpos están constituidos también por puntos y líneas, pero en éstos los puntos y líneas que los forman no están necesariamente en el mismo plano. Por ejemplo no todos los puntos y líneas de un cubo están en el mismo plano.



## ANALISIS

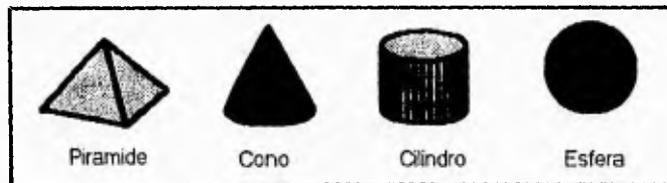
Las figuras en la geometría plana tienen una o dos dimensiones. En la figura anterior se muestran algunos cuerpos geométricos. En la figura a) se tiene una línea que tiene una sola dimensión (longitud), en la figura b) se muestra un paralelogramo que tiene dos dimensiones (longitud y anchura) en las figuras c) y d) se presentan cuerpos de geometría sólida las cuales tienen tres dimensiones (longitud, anchura y altura).

Una superficie curva es aquella que no tiene parte plana. Una superficie puede constituirse de dos o más planos, o de dos o más superficies curvas, o de una combinación de superficies planas y curvas.

La geometría sólida es una rama de las matemáticas que se encarga del estudio de superficies completamente cerradas que incluyen una porción del espacio.

A los cuerpos geométricos los podemos dividir en varias familias y éstas son:

- ★ Poliedros
  - \* Prismas
  - \* Paralelepípedos
  - \* Cilindros
- ★ Superficies Piramidales
  - \* Pirámides
  - \* Conos
- ★ Esferas



Existen algunos conceptos de la geometría plana que son de gran utilidad para la geometría sólida y que se considera conveniente explicarlos:

**a) PUNTO:** Elemento de la recta, del plano o el espacio ordinarios al que es posible asignar una posición, pero que no posee extensión en ninguna de las dimensiones posibles.

**b) LINEA:** Extensión considerada en una sola de sus tres dimensiones: la longitud. En el plano o en el espacio, una recta o una curva, indistintamente.

## **CAPITULO 2**

---

c) **POLIGONO:** Un polígono es una área que está cerrada por un conjunto de líneas en un mismo plano.

d) **POLIGONO REGULAR:** Un polígono regular es un polígono que es equilátero y sus ángulos internos son iguales.

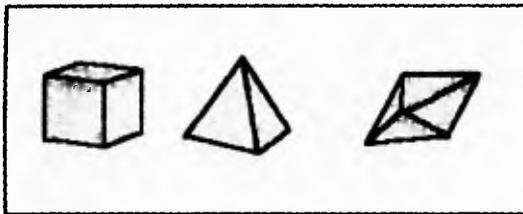
e) **CUADRILATERO:** Un cuadrilátero es un polígono que tiene cuatro lados.

f) **PARALELOGRAMO:** Un paralelogramo es un cuadrilátero que tiene dos pares de lados paralelos.

g) **RECTANGULO:** Un rectángulo es un paralelogramo que tiene sus ángulos internos de  $90^\circ$  y un par de sus lados paralelos es mayor al otro par.

h) **CIRCULO:** Un círculo es un plano curvo cerrado en donde todos los puntos están equidistantes de un punto en el interior llamado centro.

### **2.4.1.- Poliedros**



El poliedro es un sólido formado por porciones de superficies planas, éstas son llamadas las caras de el poliedro. Las líneas de intersección de las caras son orillas del poliedro, la

intersección de las orillas son vértices.

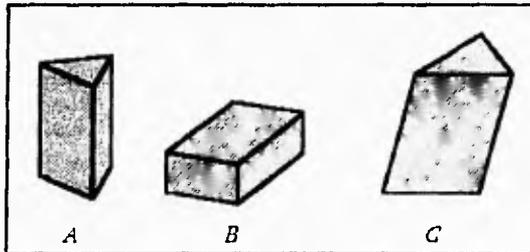
Una diagonal de un poliedro es una línea trazada de dos vértices cualquiera que no están en la misma cara. La figura plana es formada por la intersección de un plano y es llamada una sección de el sólido. La sección de un poliedro es un polígono. Un poliedro es convexo si cada sección de éste es un polígono convexo. Un poliedro se puede clasificar de acuerdo a sus caras. Los poliedros de cuatro caras se llaman tetraedros, uno de seis caras es un hexaedro, uno de ocho caras es un octaedro, uno de doce caras es un dodecaedro y uno de veinte caras es un icosaedro. Los poliedros regulares son aquellos cuyas caras son polígonos regulares congruentes y cuyos ángulos son igualmente congruentes, ésto hace posible que no puedan ser mas de 5 poliedros regulares.

**PRISMAS:** Un prisma es un poliedro formado por una superficie prismática cerrada y dos planos paralelos que cortan todos los elementos. Las secciones hechas por los dos planos son las bases de los prismas. Los demás lados del prismas son llamados caras laterales y las intersecciones de las caras

## ANALISIS

laterales son llamadas orillas laterales. Una sección recta de un prisma es aquella que es hecha por un plano perpendicular con la base. Una sección oblicua es hecha por un plano oblicuo de una de las orillas laterales.

Un Prisma recto es aquel cuyas bases son secciones derechas, como los



son los prismas A y B. Un prisma Oblicuo es un prisma cuyas bases no son secciones derechas como lo es por ejemplo el prisma C. Un prisma regular es un prisma recto cuyas bases son polígonos regulares.

### Propiedades de los prismas:

- ❖ Las caras laterales de un prisma son paralelogramos.
- ❖ Los bordes laterales de un prisma son paralelos e iguales
- ❖ Los bordes laterales de un prisma son perpendiculares a el plano de la sección derecha.

Los prismas los podemos clasificar de acuerdo a la forma que tienen sus bases; existen prismas triangulares, cuadrangulares, pentagonales, etc. de acuerdo a sus bases que son triángulos, cuadriláteros, pentágonos, etc.

**PARALELEPIPEDOS:** Un paralelepípedo es un prisma cuyas bases son paralelogramos. Un paralelepípedo recto es aquel que es un prisma recto. Por lo que todas sus orillas laterales son perpendiculares a los planos de las bases. Un paralelepípedo oblicuo es un paralelepípedo que es un prisma oblicuo. Las orillas laterales son oblicuas a los planos de la bases. Un paralelepípedo rectangular es un paralelepípedo recto cuyas bases son rectángulos y son frecuentemente llamados rectángulos sólidos.

Un cubo es un paralelepípedo rectangular cuyas orillas son todas iguales.

### Propiedades de los Paralelepipedos

- ❖ Todas las caras de un paralelepípedo son paralelogramos.
- ❖ Todas las caras de un rectángulo sólido son rectángulos.
- ❖ Las caras opuestas de un paralelepípedo son congruentes.
- ❖ Todas las caras de un cubo con cuadrados congruentes.

## **CAPITULO 2**

**CILINDRO:** Un cilindro es un sólido formado por una superficie cilíndrica cerrada y dos planos paralelos que cortan a todos los elementos. Las secciones de los planos paralelos son las bases del cilindro y la superficie lateral en la superficie cilíndrica. La altura de el cilindro es la perpendicular que se forma entre los planos de las bases.

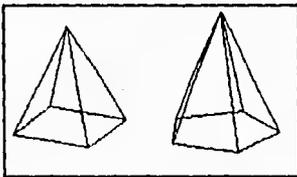
Una sección recta de un cilindro es aquella que se hace con un plano perpendicular de uno de los elementos. Un cilindro recto es aquel en donde los planos de las bases son perpendiculares a uno de los elementos. Mientras que un cilindro oblicuo es aquel en donde uno de los planos de la base es oblicuo a todos los elementos de la superficie.

Un cilindro circular es aquel que tiene por bases círculos.

### **Propiedades de los Cilindros.**

- ❖ Los elementos de un cilindro son paralelos e iguales.

### **2.4.2.- Pirámides**



Una línea que siempre se intercepta y atraviesa completamente a un polígono fijo, y pasa a través de un punto fijo que no está en el plano del polígono genera una superficie piramidal. El movimiento de la línea se le llama generatriz y el polígono fijo es la directriz.

Una pirámide es un sólido formado por una parte de una superficie piramidal y un plano, el cual corta a todos los elementos. La base de la pirámide es la sección de la superficie piramidal hecha por el plano. Las caras triangulares son las caras laterales y el lugar donde se unen las caras laterales es el borde lateral. El vértice de la superficie piramidal es el vértice de la pirámide y es donde se unen todas las caras laterales de la pirámide. La altura de la pirámide es la perpendicular de el vértice y el plano de la base de la pirámide.

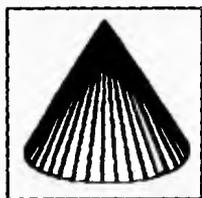
Las pirámides se clasifican de acuerdo a sus bases por ejemplo: una pirámide es triangular, cuadrangular, pentagonal, etc., de acuerdo a si su base es un triángulo, cuadrilátero, un pentágono, etc. Una pirámide triangular tiene cuatro caras, por ello se les llama tetraedro. Cualquiera de las caras del tetraedro puede ser utilizada como base, lo cual no sucede en las demás pirámides.

Una pirámide regular es una pirámide cuya base es un polígono regular, y cuya altura es perpendicular al centro de la base.

Propiedades de las Pirámides.

- ❖ Los bordes laterales de una pirámide regular son iguales.
- ❖ Las caras laterales de una pirámide regular son triángulos isósceles congruentes.
- ❖ Las alturas de las caras triangulares de una pirámide regular son iguales.

**2.4.3.- Conos.**



Una línea que siempre intercepta y atraviesa un plano convexo cerrado fijo y pasa a través de un punto fijo que no está en el plano de la curva genera una superficie convexa cónica. El movimiento de la línea es llamado la generatriz y la curva fija la directriz. Una superficie cónica al igual que una superficie piramidal

está compuesta de dos partes separadas por el vértice de la superficie.

Un sólido formado por una de las partes de la superficie cónica y cortando a todos los elementos por un plano es llamada cono. La base del cono es la sección de la superficie cónica hecha por el plano, y la superficie lateral es la parte curva de la superficie. El radio de un cono es el radio de su base. Un cono circular es un cono cuya base es un círculo. El eje de un cono circular es el segmento de línea del vértice al centro de la base. Un cono circular recto es una cono cuyo eje es perpendicular al plano de la base. Un cono cuyo eje es oblicuo al plano de la base es llamado oblicuo. De esta manera un cono circular recto tiene una superficie que puede ser generada por la revolución de un triángulo rectángulo por uno de sus bordes como eje, por lo cual en algunas ocasiones se les llama conos de revolución.

Propiedades de los conos circulares rectos.

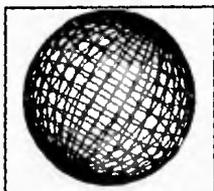
- ❖ El eje de un cono circular recto es la altura del cono
- ❖ Los elementos de un cono circular recto son iguales.

La sección de un cono circular recto hecha por un plano es llamada sección cónica. Si el plano que corta es paralelo a la base de el cono, la sección es un círculo. Si el plano que corta es oblicuo a la base y corta a todos los elementos del cono la sección resultante es una elipse. Si el plano es paralelo solamente a uno de los elementos de el cono, la sección resultante es una parábola. Si el plano es paralelo al eje de el cono y se cortan a las dos partes de la superficie cónica, resulta una sección con dos elementos llamada hipérbola.

## **CAPITULO 2**

---

### **2.4.4.- Esfera**



Una esfera es una superficie cerrada por todos los puntos y que está equidistante de un punto fijo interior llamado centro. Una esfera también puede ser definida como un conjunto de puntos en el espacio a una distancia dada desde un punto dado.

El radio de una esfera es un segmento de línea del centro a cualquier punto de la esfera. El diámetro de una esfera es el segmento de línea que pasa por el centro y va de cualquier punto de la esfera a otro punto dado.

#### Propiedades de la esfera.

- ❖ Todo radio de una esfera es igual al radio de otra esfera si dichas esferas son iguales.
- ❖ Una esfera puede ser generada por la revolución de semicírculos tomando como eje de revolución su diámetro.

### **2.5.- SISTEMA PROPUESTO**

El sistema que se está presentando es un editor de figuras geométricas en tres dimensiones, que cuenta con una serie de opciones para la manipulación de archivos tales opciones son: la creación de nuevos archivos de tipo gráfico, abrir archivos ya existentes para hacer modificaciones en ellos o incluso para imprimirlos. El formato con el cual se salvan los diseños realizados es BMP, lo que permite tener una compatibilidad con algunos paquetes, éste ofrece la posibilidad de insertar los diseños elaborados en el editor en algún procesador de texto, compatible con el sistema, en donde se requiera de un diseño en tercera dimensión.

También se cuenta con un menú para seleccionar el cuerpo geométrico que se desea elaborar, el cual tiene una ventana para la edición de la figura, donde, se le dan las dimensiones y el ángulo de inclinación que se desea, una vez requeridas las condiciones del usuario se toma la figura y se pone en un espacio tridimensional donde se podrá combinar con otras figuras diseñadas hasta lograr el diseño que el usuario desea.

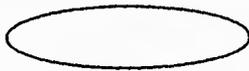
Se tiene una serie de opciones para escoger la apariencia o características propias de la figura como son su color, su consistencia, (es decir si es transparente o sólido), una imagen presenta toda una serie de vistas desde

distintos ángulos, permitiendo que el usuario sea el que rote en torno al objeto y no el objeto el que tenga que girar alrededor del usuario.

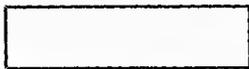
## **2.6.- PRESENTACION DE LOS DIAGRAMAS DEL SISTEMA PROPUESTO**

### **2.6.1.- Entradas y Salidas del Sistema Propuesto**

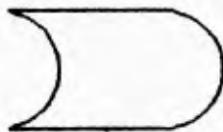
El diagrama que a continuación se muestra se obtiene del análisis realizado y también contempla el flujo de la información; pero se utilizó una cierta representación gráfica de algunas entidades en el sistema que se explican a continuación:



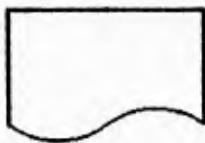
Este símbolo fue utilizado para representar una entidad física que va a alterar de alguna medida la entrada de información en el sistema.



Con este rectángulo se está representando lo que corresponde a un proceso de transformación de información para nuestro caso es el Editor 3D.



Este símbolo representa un archivo de salida producto del proceso que se realizó en el bloque de transformación.



Finalmente también se obtiene una salida impresa en el sistema para lo que se utiliza el dibujo de la izquierda para su representación dentro del diagrama.

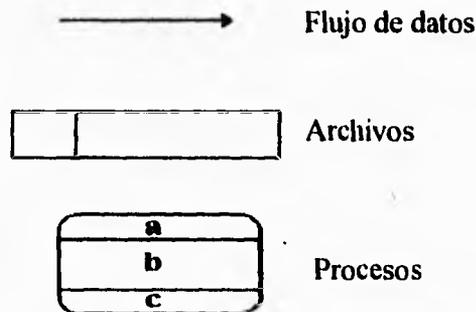
## DIAGRAMA DE ENTRADAS Y SALIDAS DEL SISTEMA



**2.6.2.- Representación Lógica del Sistema Propuesto.**

Los diagramas Lógicos de flujos de datos (DLFD) pueden particionar al sistema en diferentes niveles de expansión para poder representar los procesos o funciones que corresponden al sistema al flujo de datos.

Los elementos que componen los diagramas lógicos de flujo de datos son los siguientes:



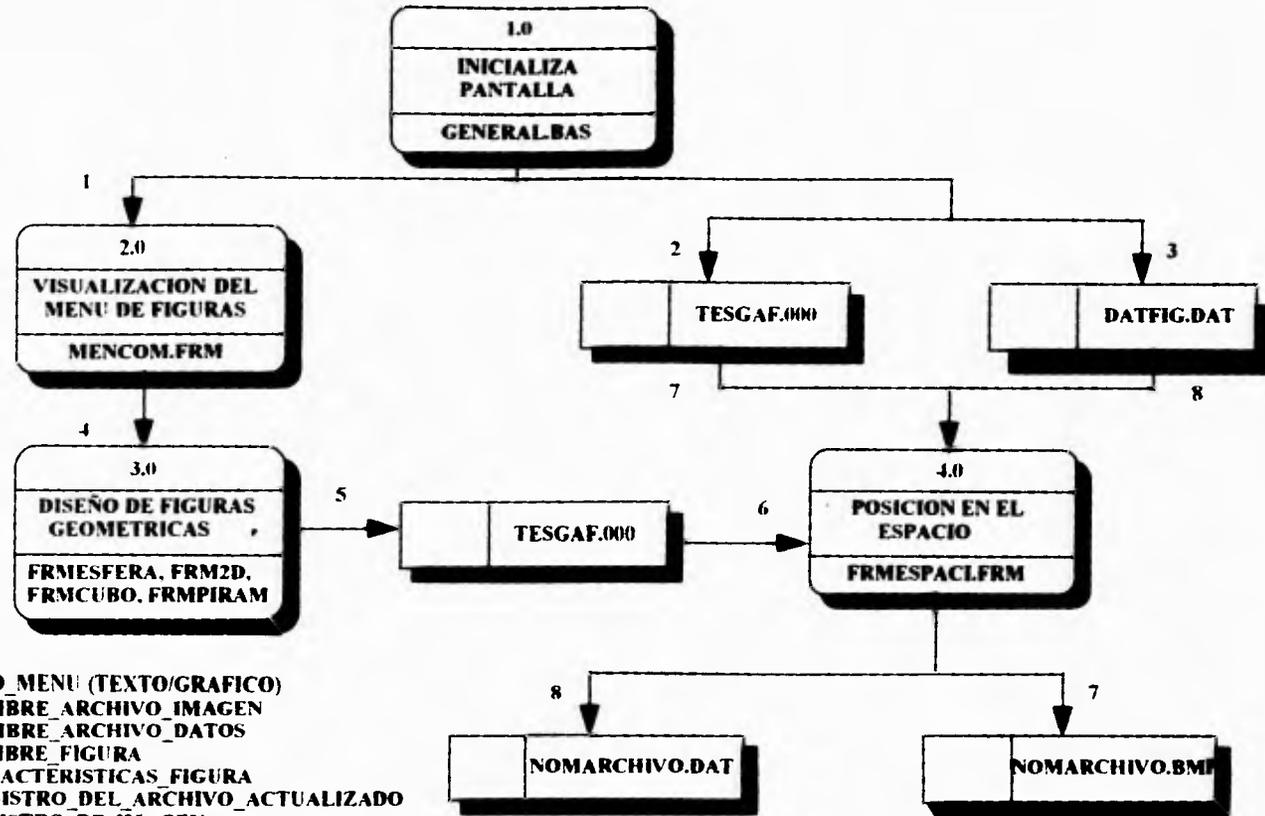
Los procesos o funciones que cambian los flujos de datos de entrada en flujos de datos de salida, se dividen en tres partes:

- a) Identificación o número dentro de un determinado nivel.
- b) Descripción de los procesos o funciones.
- c) entidad donde se ejecuta el proceso o función.

Cada nivel de los DLFD puede expandirse en niveles inferiores, dando a las cajas de proceso de nivel inferior un número de identificación que contenga el decimal de las cajas de proceso de superior nivel.

En las siguientes paginas se muestran los DLFD del sistema propuesto. El primer diagrama se muestra al sistema relacionado con los elementos terminales, en el segundo diagrama se le enfoca de manera general, y los diagramas de nivel inferior son expansiones del segundo diagrama en niveles más particulares.

## SISTEMA PROPUESTO EXPANSION NIVEL 0



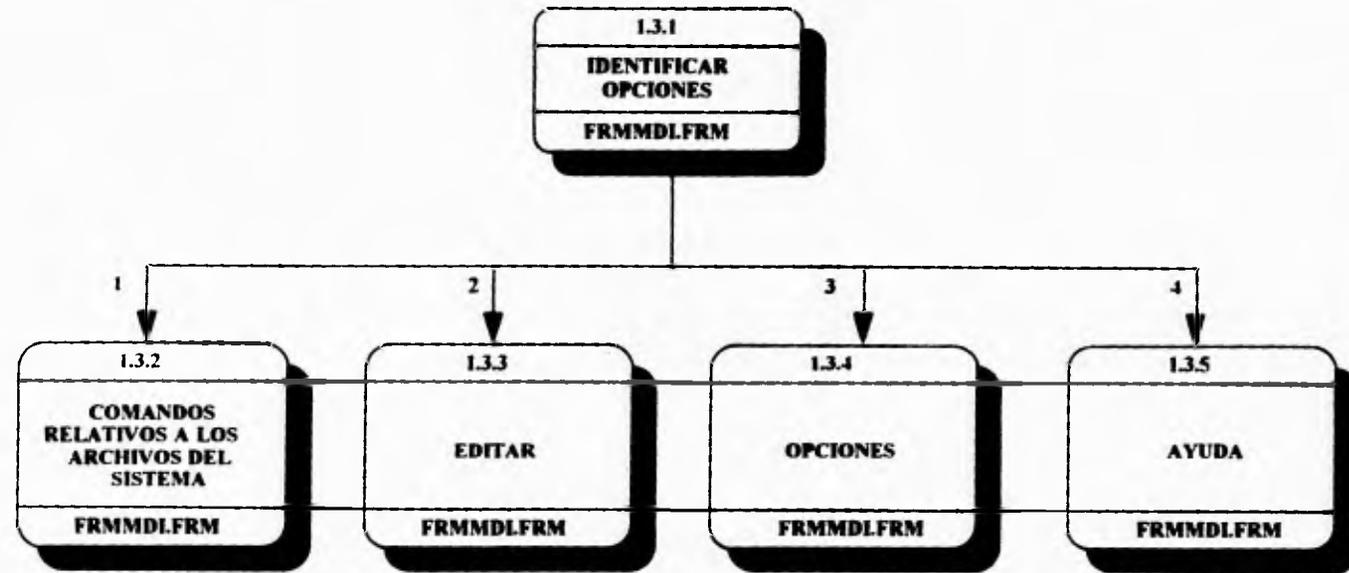
- 1.- TIPO\_MENU (TEXTO/GRAFICO)
- 2.- NOMBRE\_ARCHIVO\_IMAGEN
- 3.- NOMBRE\_ARCHIVO\_DATOS
- 4.- NOMBRE\_FIGURA
- 5.- CARACTERISTICAS\_FIGURA
- 6.- REGISTRO\_DEL\_ARCHIVO\_ACTUALIZADO
- 7.- REGISTRO\_DE\_IMAGEN
- 8.- REGISTRO\_DEL\_ARCHIVO

**SISTEMA PROPUESTO  
EXPANSION NIVEL 1.0**



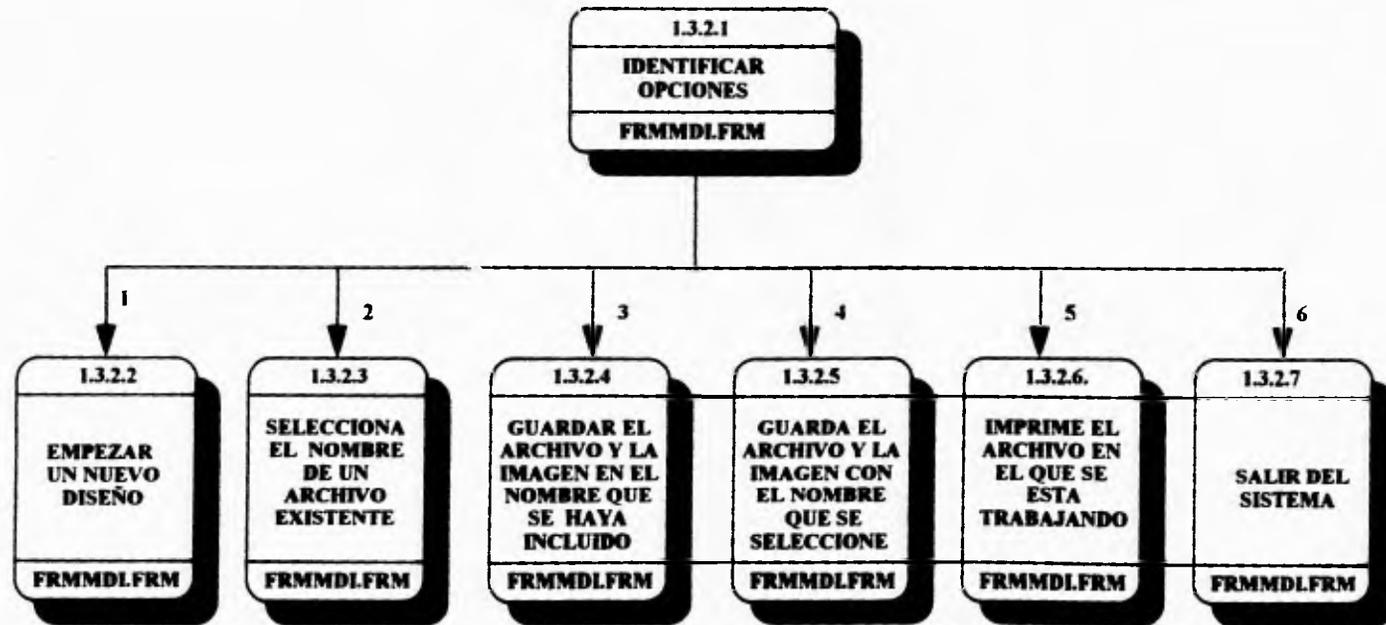
**1.- DIMENSIONES\_DE\_PANTALLA**

## SISTEMA PROPUESTO EXPANSION NIVEL 1.3



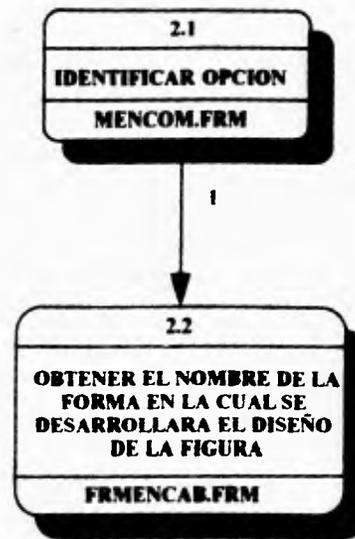
- 1.- SELECCION\_ARCHIVO
- 2.- SELECCION\_EDITAR
- 3.- SELECCION OPCIONES
- 4.- SELECCION AYUDA

## SISTEMA PROPUESTO EXPANSION NIVEL 1.3.2



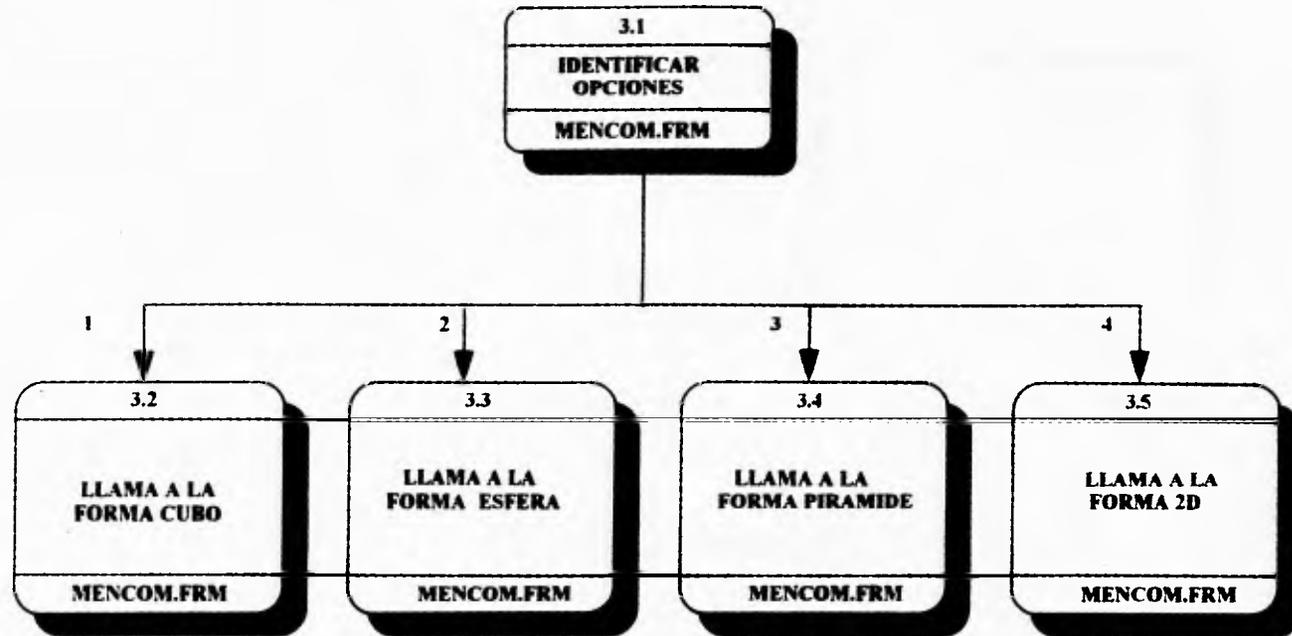
- 1.- SELECCION\_NUEVO
- 2.- SELECCION\_ABRIR
- 3.- SELECCION\_SALVAR
- 4.- SELECCION\_SALVAR\_COMO
- 5.- SELECCION\_IMPRIMIR
- 6.- SELECCION\_SALIR

**SISTEMA PROPUESTO  
EXPANSION NIVEL 2.0**



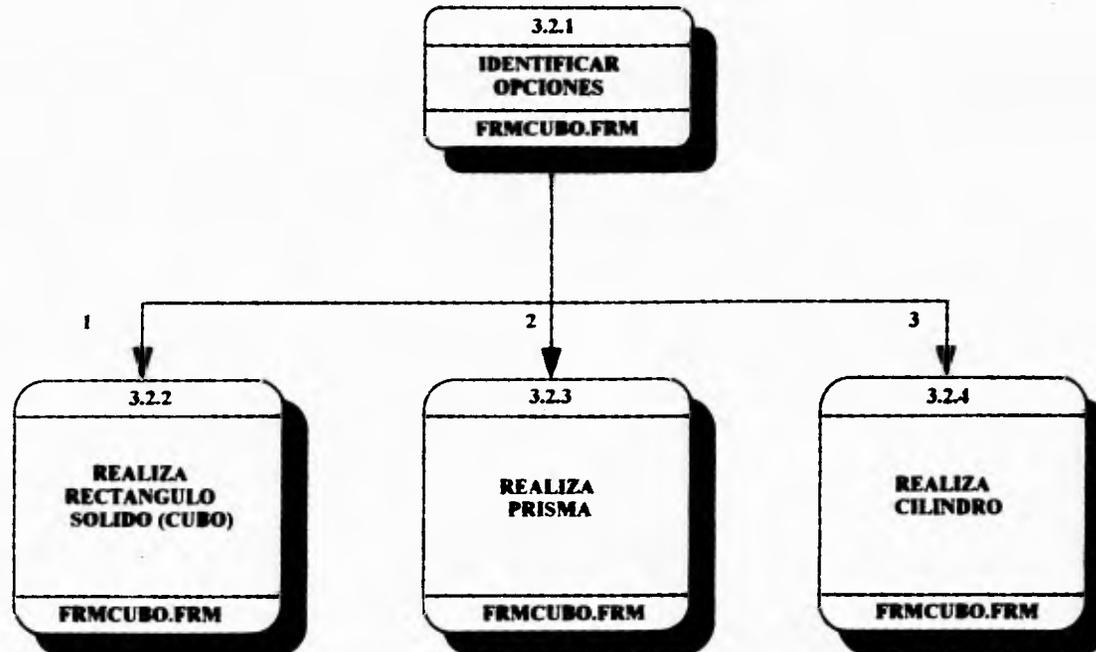
**1.- NOMBRE\_FIGURA**

## SISTEMA PROPUESTO EXPANSION NIVEL 3.0



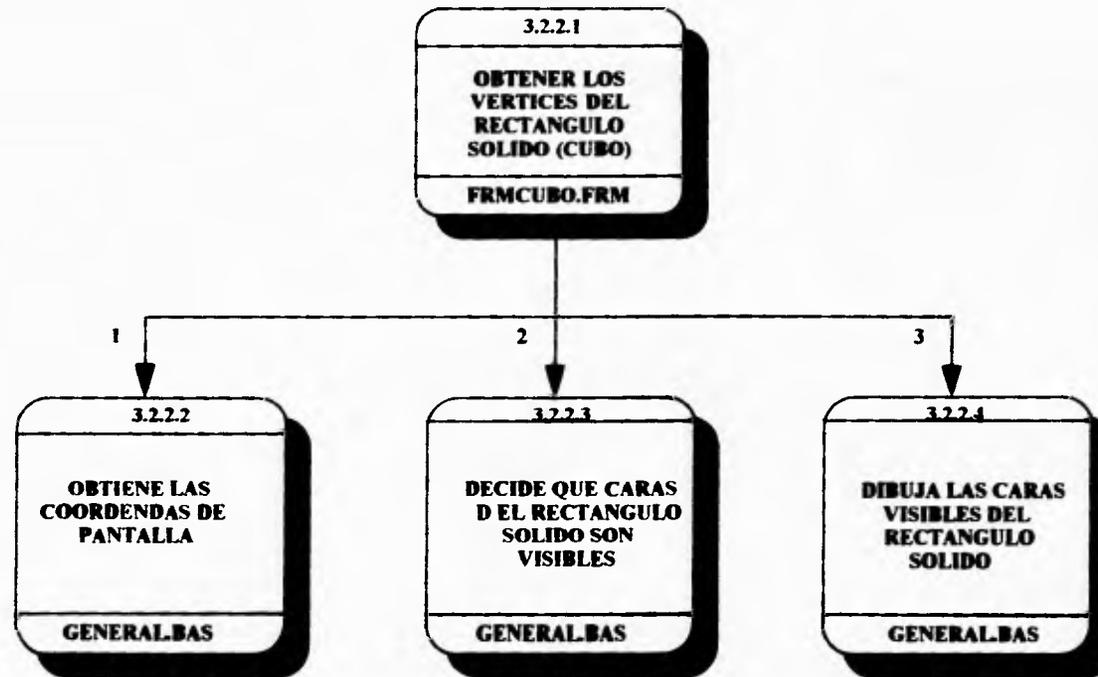
- 1.- CLASE POLIEDROS\_NO\_PIRAMIDALES
- 2.- CLASE\_ESFERA
- 3.- CLASE\_PIRAMIDES
- 4.- CLASE\_2D

## SISTEMA PROPUESTO EXPANSION NIVEL 3.2



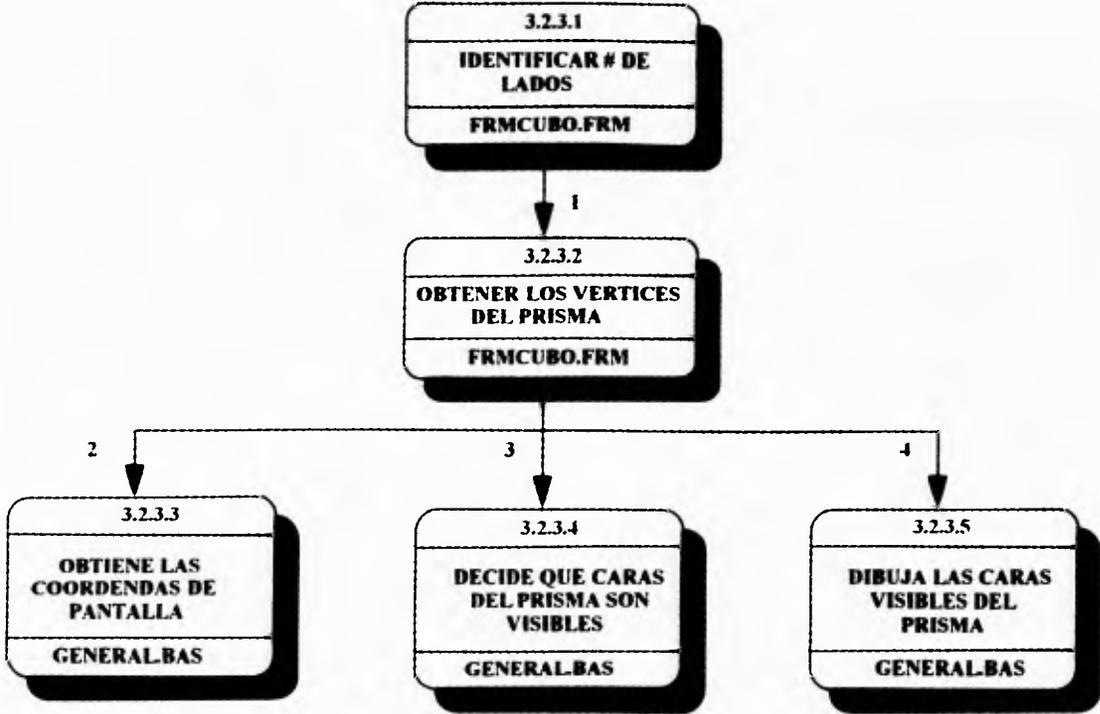
- 1.- TIPO\_CUBO
- 2.- TIPO\_PRISMA
- 3.- TIPO\_CILINDRO

**SISTEMA PROPUESTO  
EXPANSION NIVEL 3.2.2**



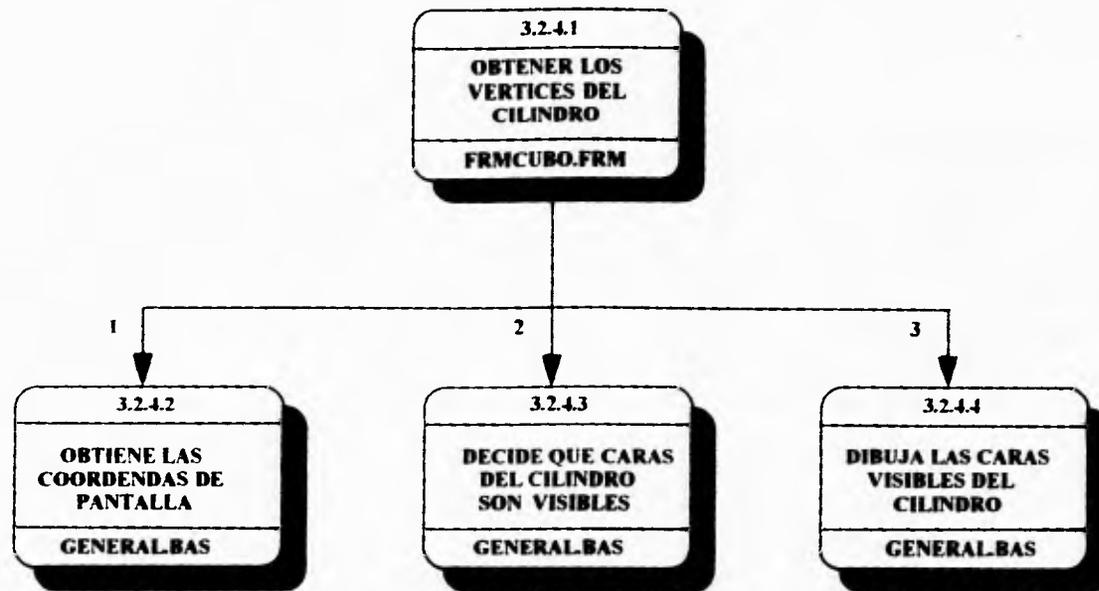
- 1.- COORDENADAS UNIVERSALES
- 2.- ECUACIONES DEL PLANO X,Y,Z
- 3.- COORDENADAS\_PANTALLA

**SISTEMA PROPUESTO  
EXPANSION NIVEL 3.2.3**



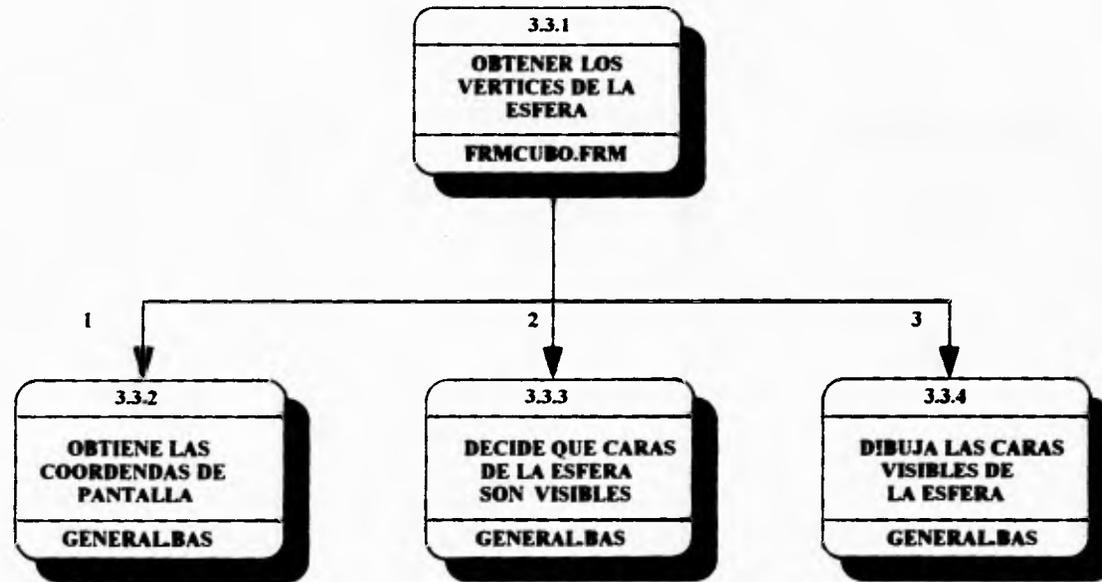
- 1.- NUMERO\_LADOS
- 2.- COORDENADAS\_UNIVERSALES
- 3.- ECUACIONES\_DEL\_PLANO\_X,Y,Z
- 4.- COORDENADAS\_PANTALLA

**SISTEMA PROPUESTO  
EXPANSION NIVEL 3.2.4**



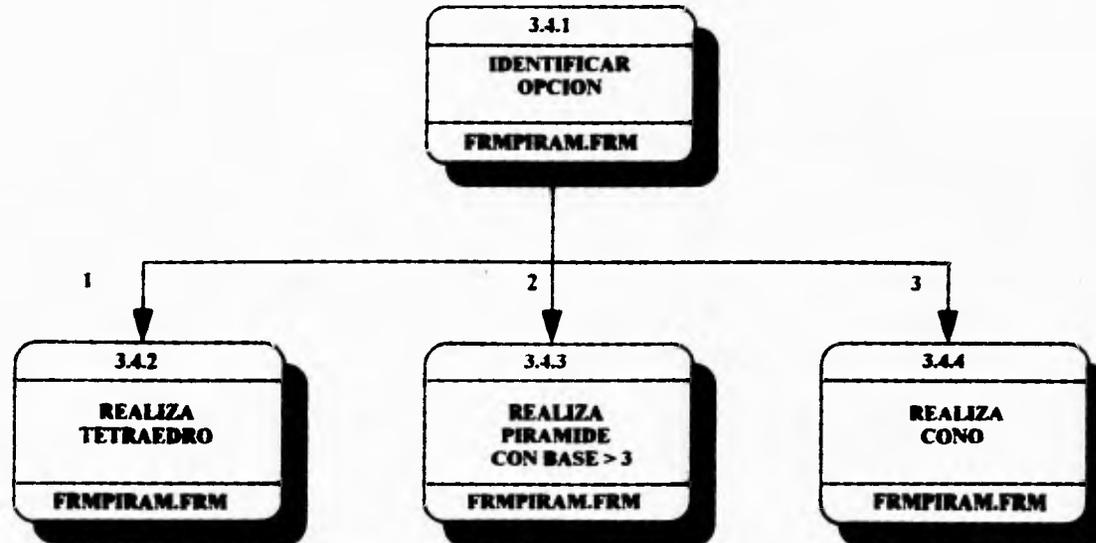
- 1.- COORDENADAS\_UNIVERSALES
- 2.- ECUACIONES\_DEL\_PLANO\_X,Y,Z
- 3.- COORDENADAS\_PANTALLA

**SISTEMA PROPUESTO  
EXPANSION NIVEL 3.3**



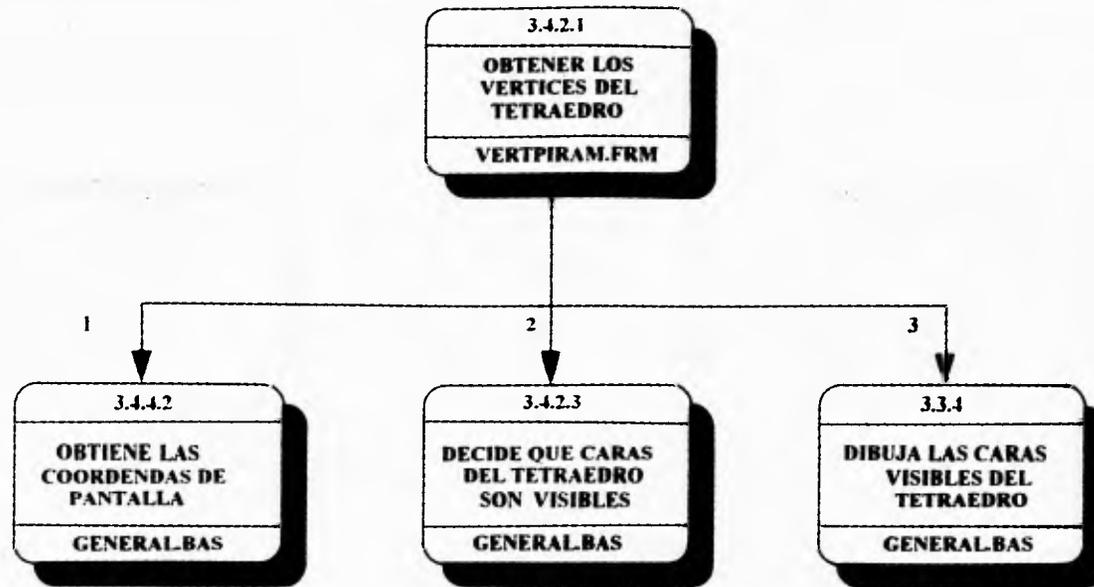
- 1.- COORDENADAS UNIVERSALES
- 2.- ECUACIONES DEL PLANO X,Y,Z
- 3.- COORDENADAS PANTALLA

## SISTEMA PROPUESTO EXPANSION NIVEL 3.4



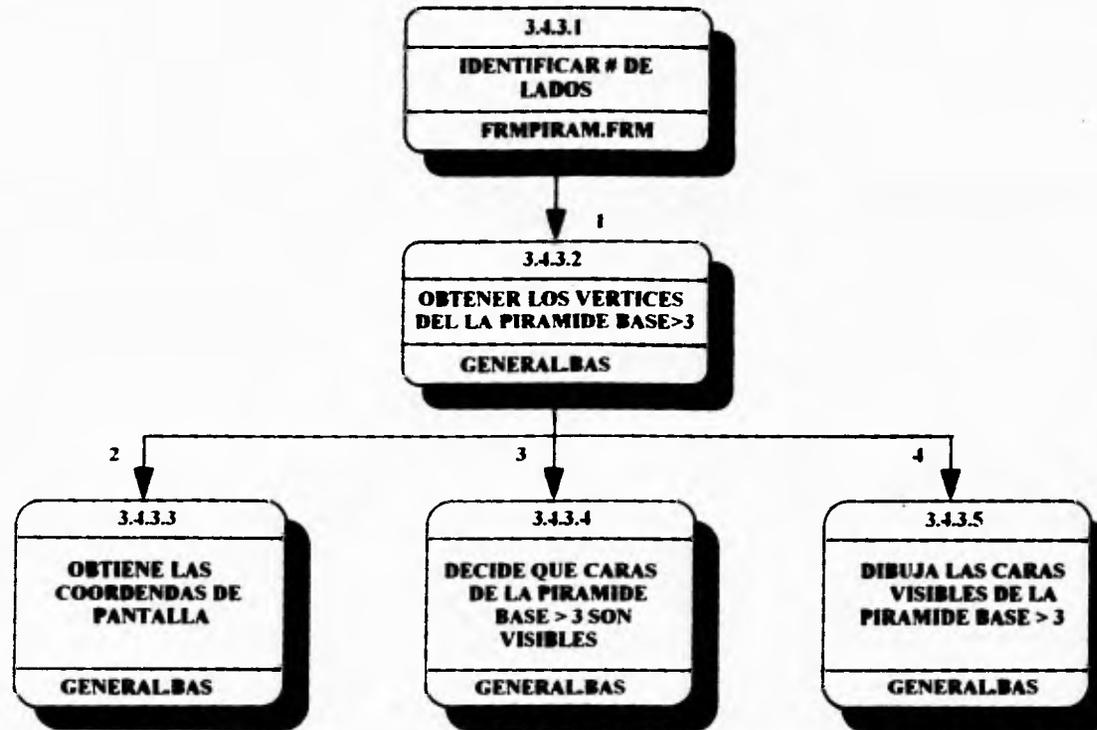
- 1.- TIPO\_TETRAEDRO
- 2.- TIPO\_BASE>3
- 3.- TIPO\_CONO

**SISTEMA PROPUESTO  
EXPANSION NIVEL 3.4.2**



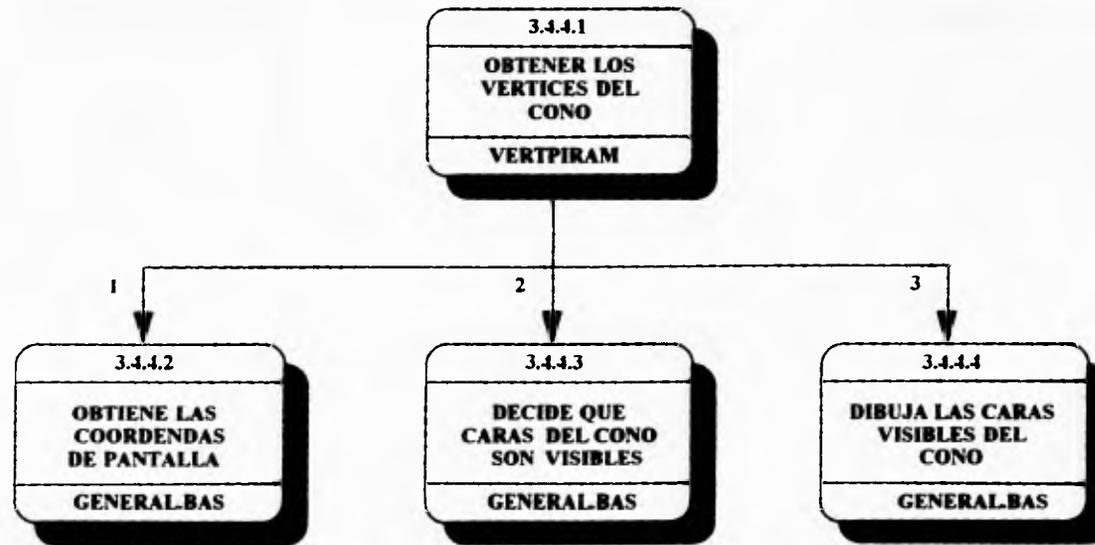
- 1.- COORDENADAS UNIVERSALES
- 2.- ECUACIONES DEL PLANO X,Y,Z
- 3.- COORDENADAS PANTALLA

## SISTEMA PROPUESTO EXPANSION NIVEL 3.4.3



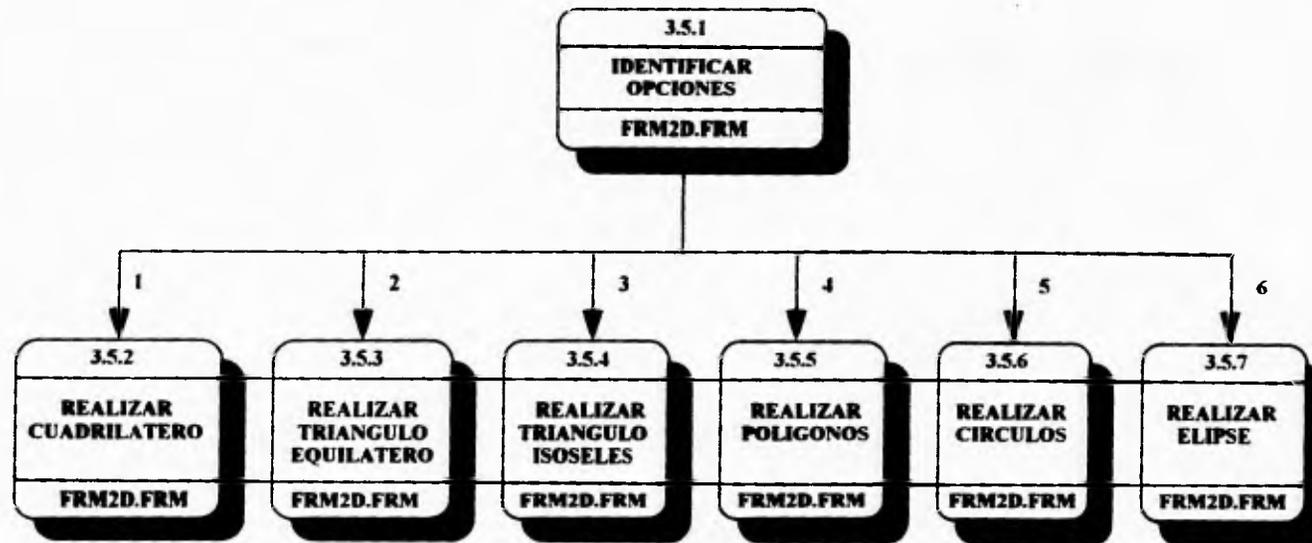
- 1.- NUMERO LADOS
- 2.- COORDENADAS UNIVERSALES
- 3.- ECUACIONES DEL PLANO X,Y,Z
- 4.- COORDENADAS PANTALLA

**SISTEMA PROPUESTO**  
**EXPANSION NIVEL 3.4.4**



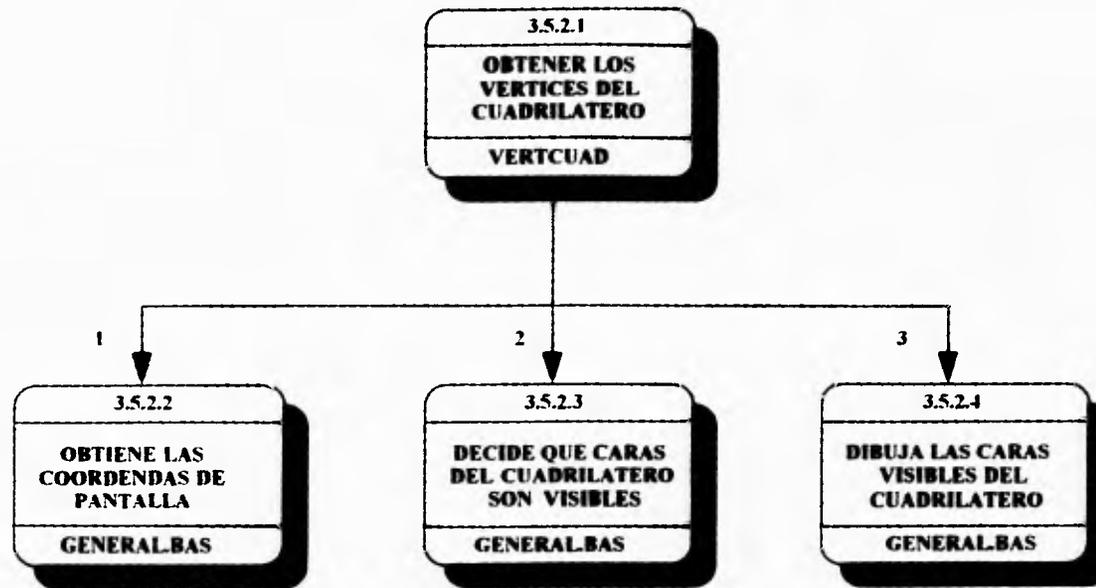
- 1.- COORDENADAS UNIVERSALES
- 2.- ECUACIONES DEL PLANO X,Y,Z
- 3.- COORDENADAS PANTALLA

## SISTEMA PROPUESTO EXPANSION NIVEL 3.5



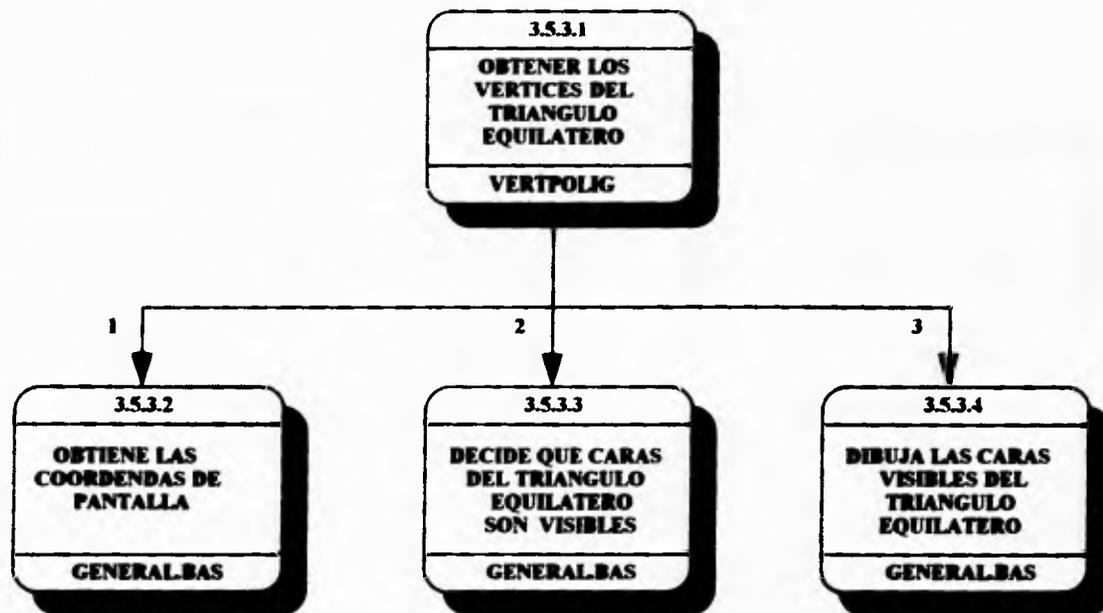
- 1.- TIPO\_CUADRILATERO
- 2.- TIPO\_TRIANGULO\_EQUILATERO
- 3.- TIPO\_TRIANGULO\_ISOSELES
- 4.- TIPO\_POLIGONO
- 5.- TIPO\_CIRCULO
- 6.- TIPO\_ELIPSE

## SISTEMA PROPUESTO EXPANSION NIVEL 3.5.2



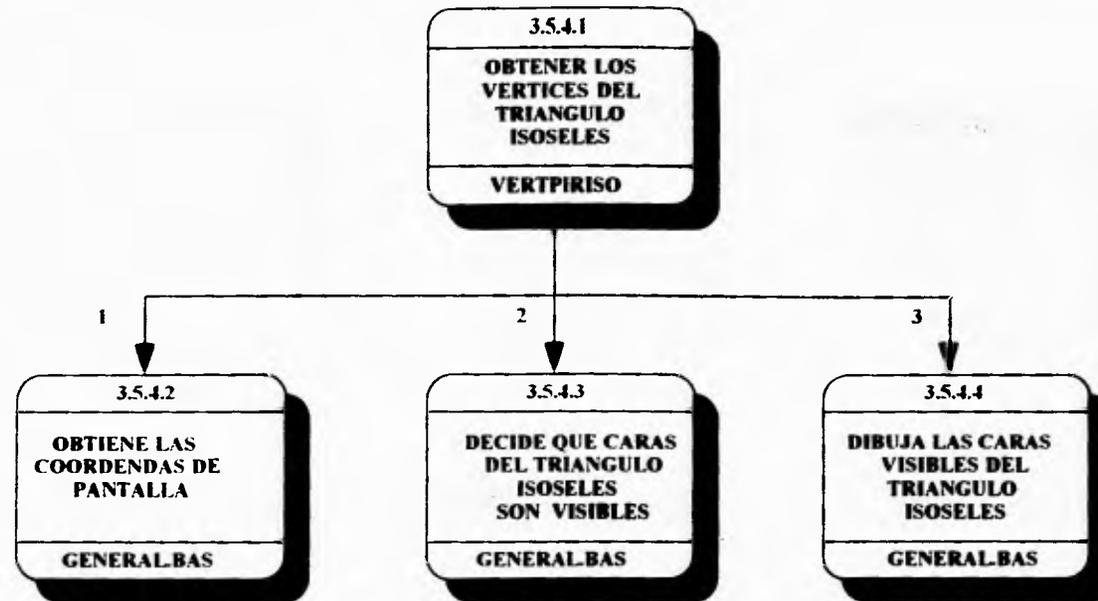
- 1.- COORDENADAS\_UNIVERSALES
- 2.- ECUACIONES\_DEL\_PLANO\_X,Y,Z
- 3.- COORDENADAS\_PANTALLA

## SISTEMA PROPUESTO EXPANSION NIVEL 3.5.3



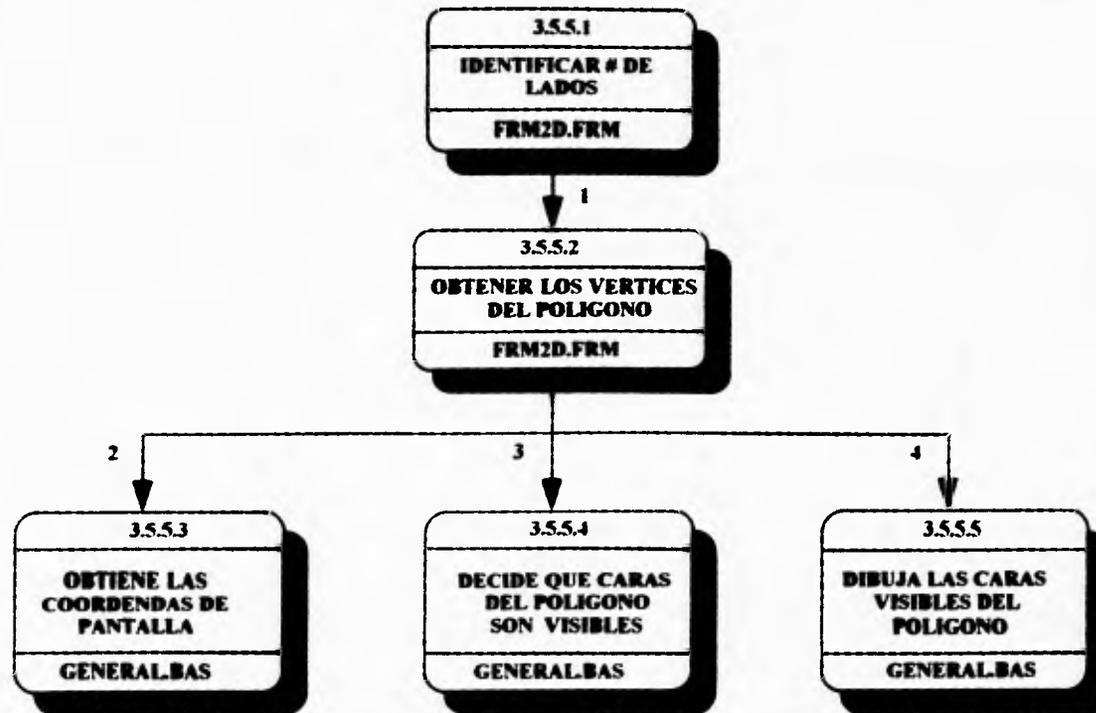
- 1.- COORDENADAS UNIVERSALES
- 2.- ECUACIONES DEL PLANO X,Y,Z
- 3.- COORDENADAS PANTALLA

## SISTEMA PROPUESTO EXPANSION NIVEL 3.5.4



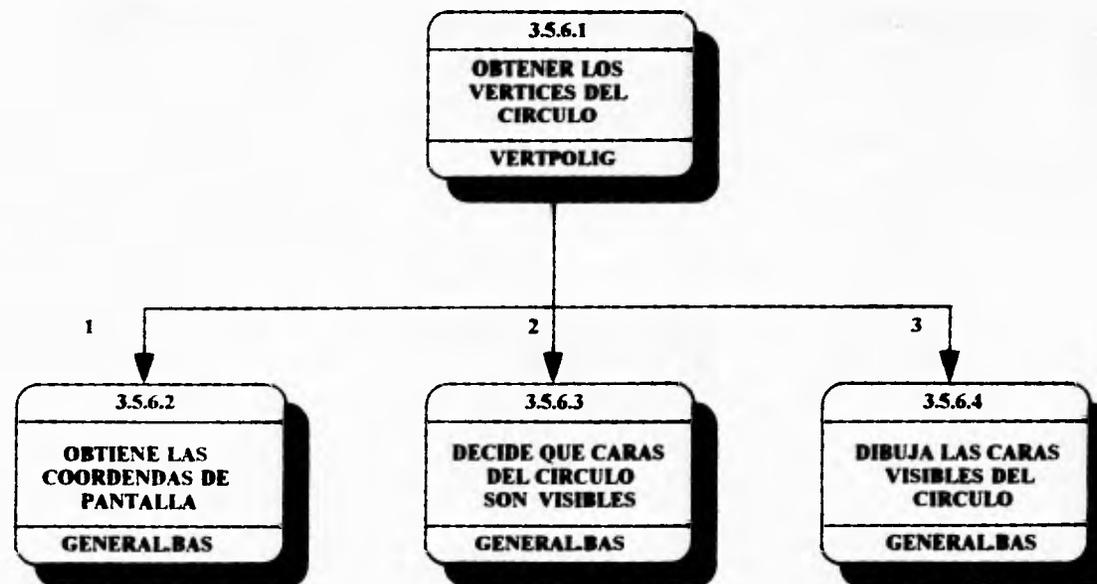
- 1.- COORDENADAS UNIVERSALES
- 2.- ECUACIONES DEL PLANO X,Y,Z
- 3.- COORDENADAS PANTALLA

## SISTEMA PROPUESTO EXPANSION NIVEL 3.5.5



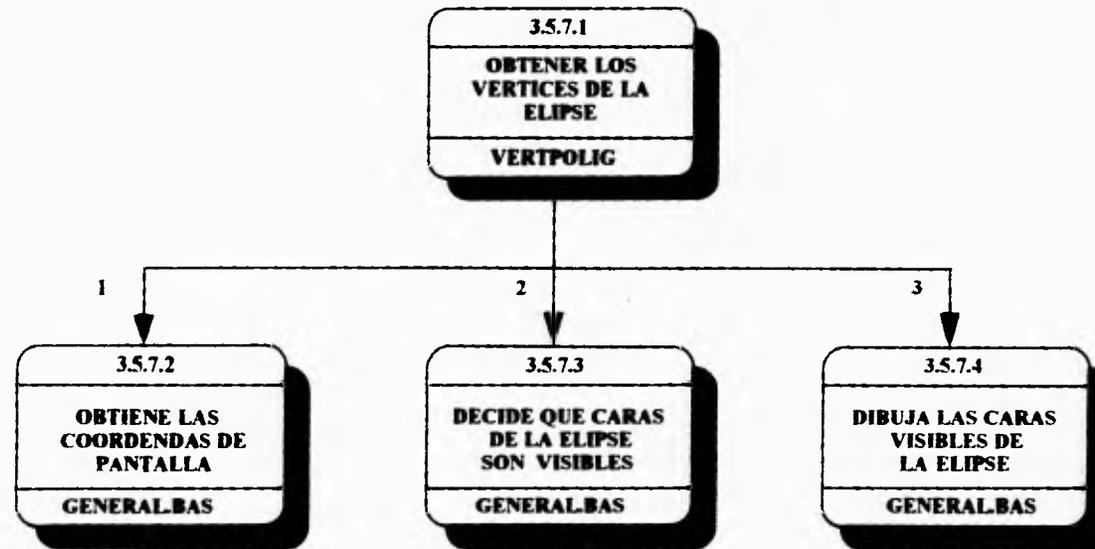
- 1.- NUMERO\_LADOS
- 2.- COORDENADAS\_UNIVERSALES
- 3.- ECUACIONES\_DEL\_PLANO\_X,Y,Z
- 4.- COORDENADAS\_PANTALLA

**SISTEMA PROPUESTO**  
**EXPANSION NIVEL 3.5.6**



- 1.- COORDENADAS\_UNIVERSALES
- 2.- ECUACIONES\_DEL\_PLANO\_X.Y.Z
- 3.- COORDENADAS\_PANTALLA

**SISTEMA PROPUESTO  
EXPANSION NIVEL 3.5.7**



- 1.- COORDENADAS UNIVERSALES
- 2.- ECUACIONES DEL PLANO X,Y,Z
- 3.- COORDENADAS PANTALLA



### CAPITULO 3

## DISEÑO DEL SISTEMA



Superficie:

Figura geométrica definida por el conjunto de puntos del espacio cuyas coordenadas verifican una función.

**PAGINACION VARIA**

**COMPLETA LA INFORMACION**

---

## DISEÑO DEL SISTEMA

**M**uchos de los sistemas de computación que se desarrollan no tienen un avance metódico en su estructura. Todas las funciones que lo componen y sus interfaces generalmente no han sido planeadas de una manera sistemática. El diseño, trata de responder a muchas preguntas que no han podido resolverse en muchos procesos de organización de datos.

Diseño significa la planeación de las tareas que se han de realizar para la solución de un problema determinado y es este proceso el que determina las características principales del sistema, establece los principales pasos a realizar y la mejor implementación que podemos realizar y que determinan las características finales de un sistema. La ingeniería busca la mejor solución en términos establecidos dentro de unas limitaciones previamente conocidas trabajando dentro de las posibilidades del mundo real.

El diseño no crea una nueva actividad en el proceso de desarrollo de los sistemas, sino solamente consolida, formaliza y hace visibles actividades y decisiones que ocurren inevitablemente durante la creación del sistema y las aborda como alternativas evaluables.

Resumiendo, al diseño se le puede ver como el arte de diseñar los componentes de un sistema y la interrelación entre éstos para lograr un resultado óptimo en un sistema decidiendo qué componentes interconectan con cuáles y de qué forma se puede resolver de una mejor manera un problema en específico.

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 una secuencia contigua de instrucciones, delimitadas por elementos de frontera o por límites específicos, teniendo un identificador asociado a él.

Dada una definición de requisitos se debe desarrollar el diseño de un sistema de programación que satisfaga éstos, y se realizan en varias etapas:

- 1.- Para hacer el diseño de un sistema de computación la ingeniería dice que se necesita primeramente definir de manera clara los objetivos que

## DISEÑO DEL SISTEMA

se esperan de el sistema, ya que éstos determinan el criterio que se deberá tomar para las decisiones individuales que se harán en el diseño del sistema.

- 2.- Para poder tomar decisiones claras, se deben de conocer los puntos de decisiones que existen así como reconocer las posibles opciones que se pueden presentar .
- 3.- Deben establecerse los subsistemas que componen el sistema de programación.
- 4.- Cada subsistema debe dividirse en componentes individuales y ha de establecer la especificación de los subsistemas definiendo la operación de esos componentes.
- 5.- Después, cada programa se puede diseñar a base de subcomponentes que actúen recíprocamente.
- 6.- Después hay que refinar cada componente. Esto suele implicar la especificación de cada componente como una jerarquía de subcomponentes.
- 7.- En algún momento de este proceso de refinamiento hay que especificar con detalle los algoritmos utilizados en cada componente .
- 8.- Finalmente se necesita conocer alguna metodología o técnicas que auxilien en la toma de decisiones para construir un diseño satisfactorio.

Existe una serie de razones por las que es conveniente utilizar el diseño en la elaboración de un sistema, dichas razones se exponen a continuación:

- ★ **Eficiencia:** En general se quiere, que los sistemas y programas sean eficientes y que hagan un uso eficiente de los recursos del sistema, estos recursos incluyen no únicamente a la unidad central de proceso (CPU) y a la memoria primaria del sistema sino también a los dispositivos de almacenamiento secundarios, el tiempo de uso de los periféricos de entrada/salida, el tiempo de proceso en la información, y muchas otras cosas más.
- ★ **Mantenimiento :** El termino "mantenimiento" se ha aplicado al proceso de modificar un programa cuando ya se ha entregado y está en uso. Estas modificaciones pueden implicar cambios sencillos para corregir errores de codificación, cambios mayores para corregir errores de diseño o reescrituras drásticas para corregir esos errores de especificación o introducir nuevos requisitos. Tiene poco sentido producir un sistema si no se puede comprender , si sólo lo pueden utilizar sus realizadores y si es difícil o

### **CAPITULO 3**

---

imposible modificarlo. Por lo que es muy necesario invertir tiempo y esfuerzo cuando se diseña y aplica un sistema para reducir el mantenimiento y con ello los costos totales .

- ❶ **Confiabilidad:** La confiabilidad de cualquier sistema depende de la corrección de su diseño, lo correcto de la correspondencia entre éste y la aplicación; se puede decir que un sistema es confiable si cumple con sus especificaciones y se comporta según lo estipulado, nunca debe permitir su corrupción, debe de reaccionar de una manera significativa y útil en situaciones inesperadas, y sólo debe de fallar por completo cuando sea imposible seguir avanzando.
- ❷ **Utilidad:** La utilidad o fácil uso es una medida que se utiliza con frecuencia para medir el buen funcionamiento de un sistema y recibe una atención mayor en el momento en que se plantea el análisis de los sistemas. Sabemos que parte de estos problemas se basan en el conocimiento de factores humanos y otra parte en el diseño de caminos apropiados para la solución de los problemas.
- ❸ **Acoplamiento:** Un módulo es independiente si puede funcionar sin la presencia de algún otro módulo, pero ésto es prácticamente imposible, pues lo normal es que haya una interface por la cual se le pase el control y los datos. Para hacer módulos independientes es necesario reducir el acoplamiento entre los diferentes módulos. El acoplamiento es una medida del grado de interdependencia entre módulos.
- ❹ **Cohesión:** Se puede hablar de un programa con cohesión si los elementos de éste muestran un alto grado de relación funcional. Esto significa que cada elemento de la unidad de programa debe ser esencial para que este programa alcance su propósito, como ordenar un archivo, buscar en un diccionario, etc. La cohesión es el factor esencial del diseño , ya que con ella, se puede medir la fuerza de asociación de las instrucciones y llamadas que efectúa un módulo. Existen entre el acoplamiento y la cohesión una relación muy importante: a mayor grado de cohesión de un módulo se tiene un menor acoplamiento entre los módulos. Cuando se tiene que escoger alguna de las dos opciones, es preferible mantener una alta cohesión aunque haya una pérdida en el acoplamiento.

### **3.1.- TECNICAS DEL DISEÑO**

El diseño efectivo del software se logra mejor 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 estas metodologías se pueden clasificar en una de las tres áreas siguientes:

- 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.** Esta metodología 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.

Unas notaciones consistentes y completas son muy apreciadas en la creación de objetos abstractos como los sistemas. Sin tales notaciones, los diseños no se pueden evaluar, comparar, probar o comunicar. Aunque el programa de computación es en sí mismo la especificación absoluta del diseño, el grado de detalle que se presenta en el programa es tal que resulta inadecuado para transmitir el diseño a los lectores humanos.

Se han elegido dos notaciones complementarias que pueden utilizarse juntas para describir un diseño de software.

Las notaciones elegidas son:

- 1) Diagramas de flujo de datos. Que fue explicado en el capítulo anterior.
- 2) Diagramas de estructura. Que a continuación se explica

## **CAPITULO 3**

---

### **3.1.1- Diagrama Jerárquico Funcional.**

El objetivo fundamental de este diagrama es proporcionar una estructura del sistema que facilite el entendimiento de sus funciones. Su construcción se realiza con base en el principio de diseño de arriba hacia abajo (top-down design). El diagrama jerárquico funcional proporciona una representación esquemática del sistema y una relación jerárquica de sus elementos (subsistemas, submódulos, procesos, etc. )

Cada bloque en el diagrama jerárquico tiene un número de identificación que servirá de referencia para los diagramas de flujo de datos, así como para los diagramas de procedimientos. Los bloques del diagrama pueden ser numerados secuencialmente, definiendo su estructura jerárquica funcional.

El diagrama jerárquico se utiliza para estructurar funcionalmente un sistema. Cada función se divide en subfunciones y éstas, a su vez, en otras subfunciones.

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:

- a) Localizar una función específica.
- b) Entender al sistema a un cierto nivel de la estructura.

Un diagrama jerárquico puede utilizarse para iniciar el diseño conceptual de un nuevo sistema, la importancia de un diagrama jerárquico es que identifica todas las funciones de un sistema y establece la estructura jerárquica de dichas funciones.

Son gráficas de jerarquía que muestran la relación estructural de los componentes de un sistema de software. Esta notación es más útil para describir el diseño de sistemas de alto nivel. Estos describen el sistema de programación como una jerarquía de partes y lo muestran gráficamente como un árbol. Estos diagramas documentan como se pueden aplicar los elementos de un diagrama de flujo de datos como una jerarquía de unidades de programa. Un diagrama de estructura muestra las relaciones entre las unidades de programa sin incluir ninguna información acerca del orden de activación de esas unidades. Se traza mediante tres símbolos:

- A)** Un rectángulo con el nombre de la unidad
- B)** Una flecha que contenga los rectángulos

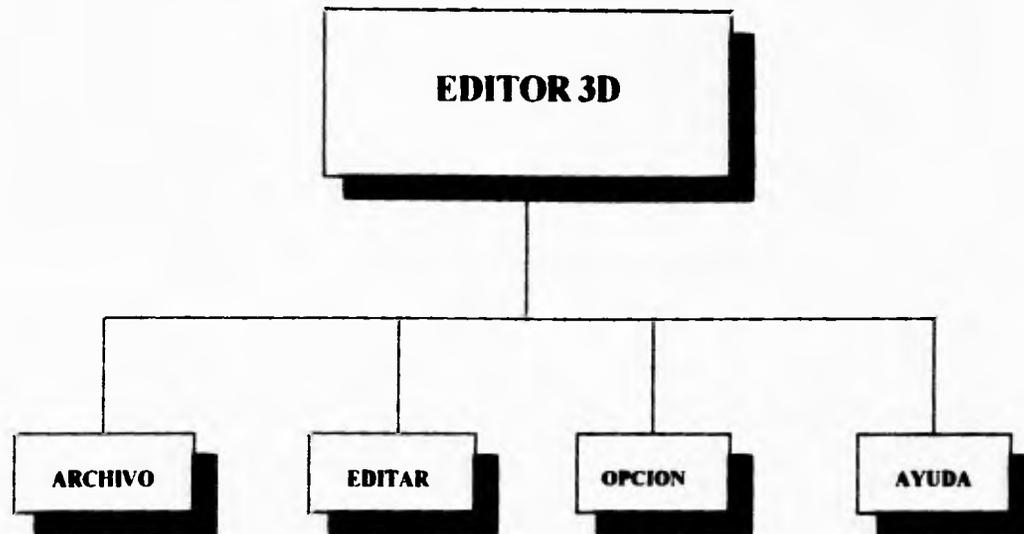
## **DISEÑO DEL SISTEMA**

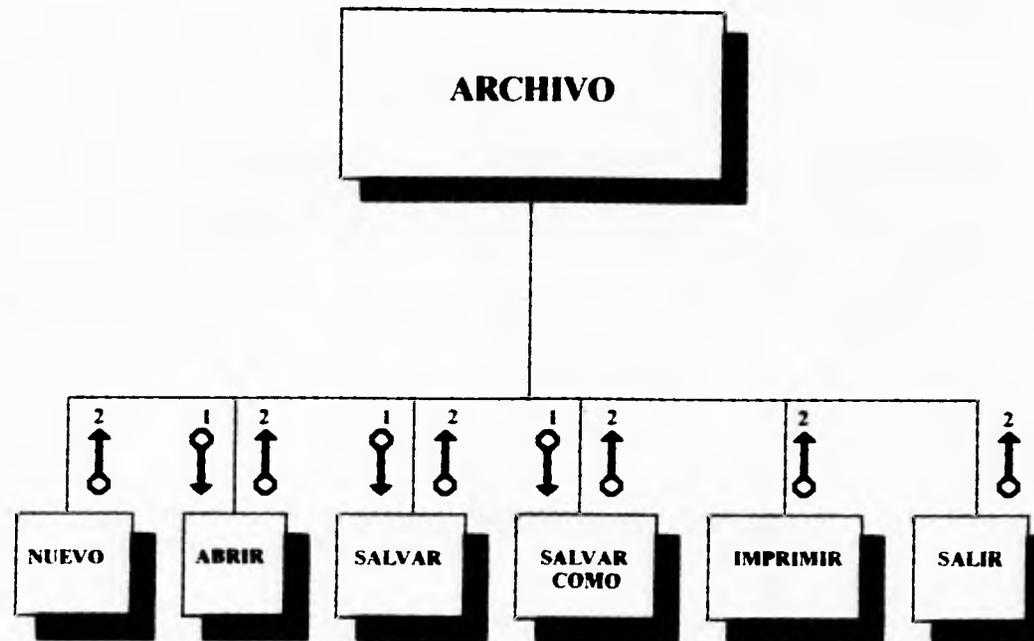
- C) Una flecha con un círculo, con el nombre de los datos que se pasan entre los elementos del diagrama de estructura. Las flechas con círculos suelen dibujarse paralelas a las flechas que contengan los rectángulos del diagrama.

### **3.2.- DIAGRAMAS DE ESTRUCTURA DEL SISTEMA PROPUESTO.**

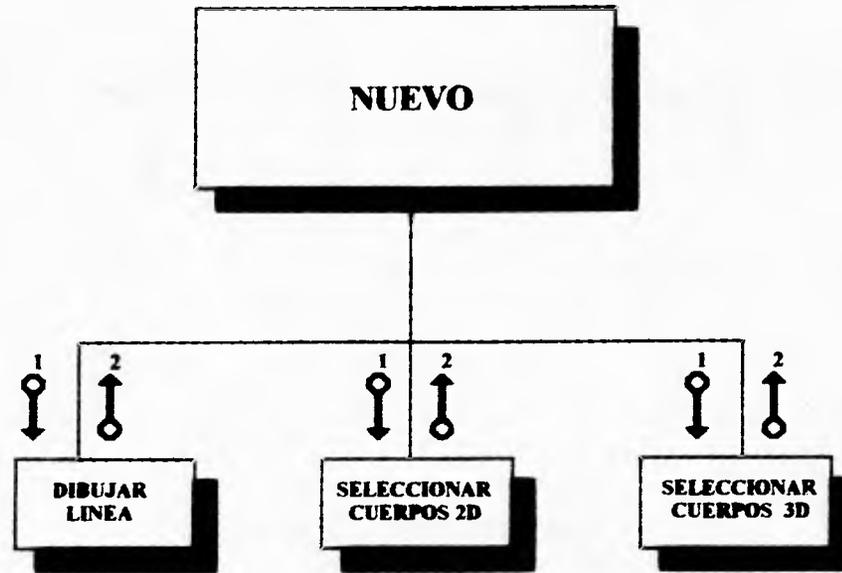
En las siguientes páginas se muestran algunos Diagramas de Estructura del sistema. En los diagramas se mencionan los nombres físicos de los procesos que se pueden verificar en el capítulo de implementación, así como los parámetros que se utilizan en los procesos.

El primer Diagrama de Estructura muestra en forma general al sistema, y se formó a partir de un análisis de transacciones. Los demás Diagramas de Estructura se encuentran refinados para obtener una alta cohesión.



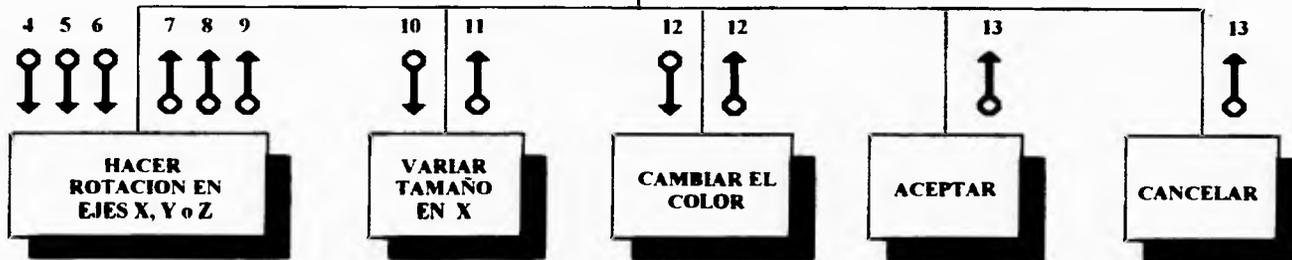
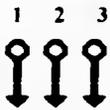


1.- ABRIRARCHIVO  
2.- OK



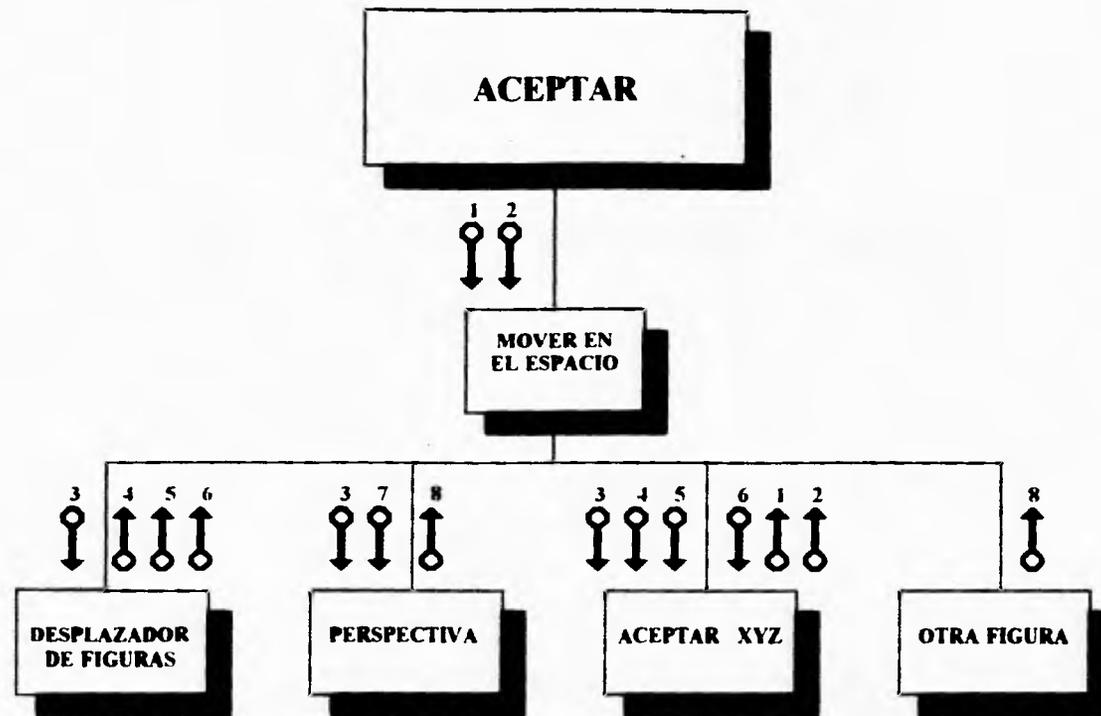
1.- FIGURA  
2.- OK

# DIBUJAR LINEA



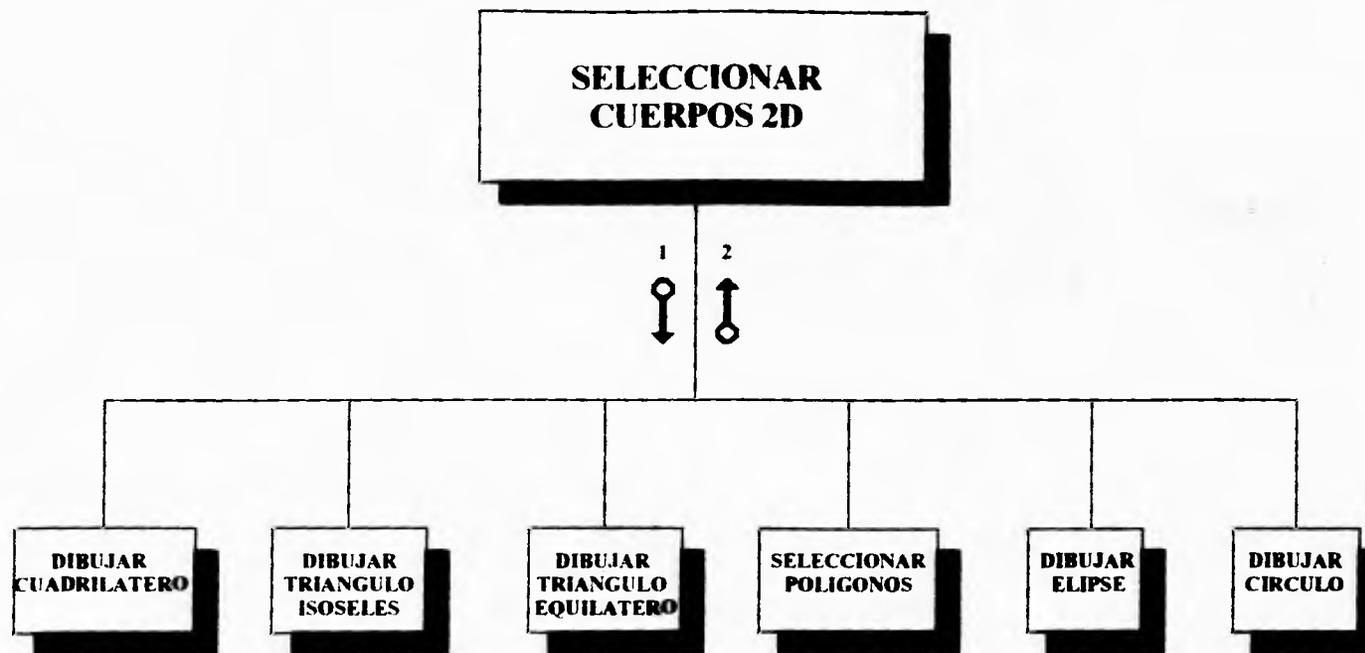
- 1.- OTRALISTA
- 2.- TITULO
- 3.- TIPO
- 4.- VALR1
- 5.- VALR2
- 6.- VALR3
- 7.- R1

- 8.- R2
- 9.- R3
- 10.- VALX
- 11.- XV
- 12.- COLORLINEA
- 13.- OK



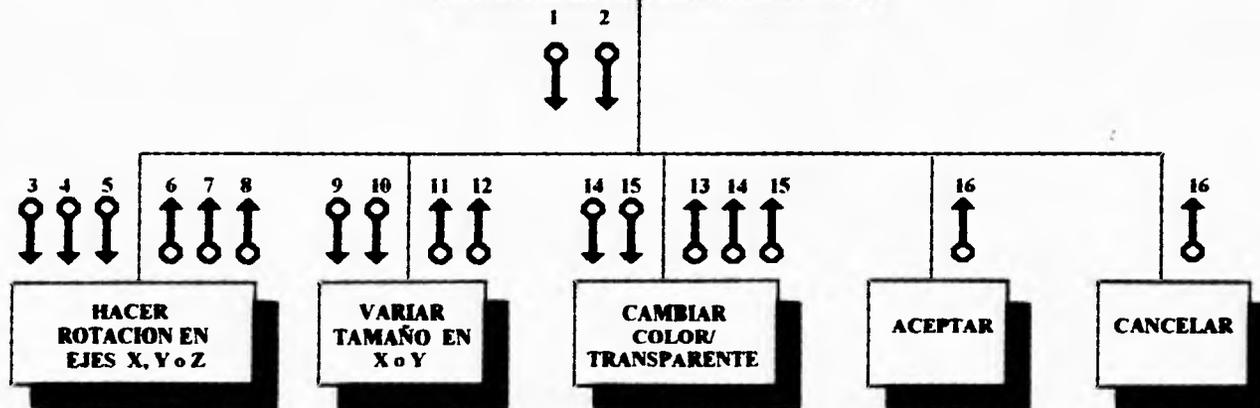
1.- INFORMACION  
 2.- NI MFIGURA  
 3.- LEIDO  
 4.- MX

5.- MY  
 6.- MZ  
 7.- ROTVISX  
 8.- OK



1.- OTRALISTA  
2.- OK

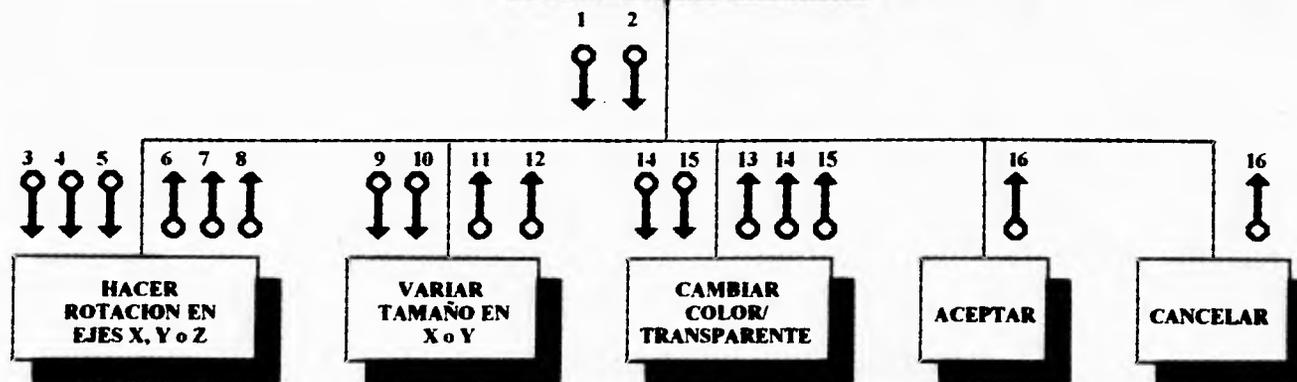
# DIBUJAR CUADRILATERO



- 1.- TIPO
- 2.- TITULO
- 3.- VALR1
- 4.- VALR2
- 5.- VALR3
- 6.- R1
- 7.- R2
- 8.- R3

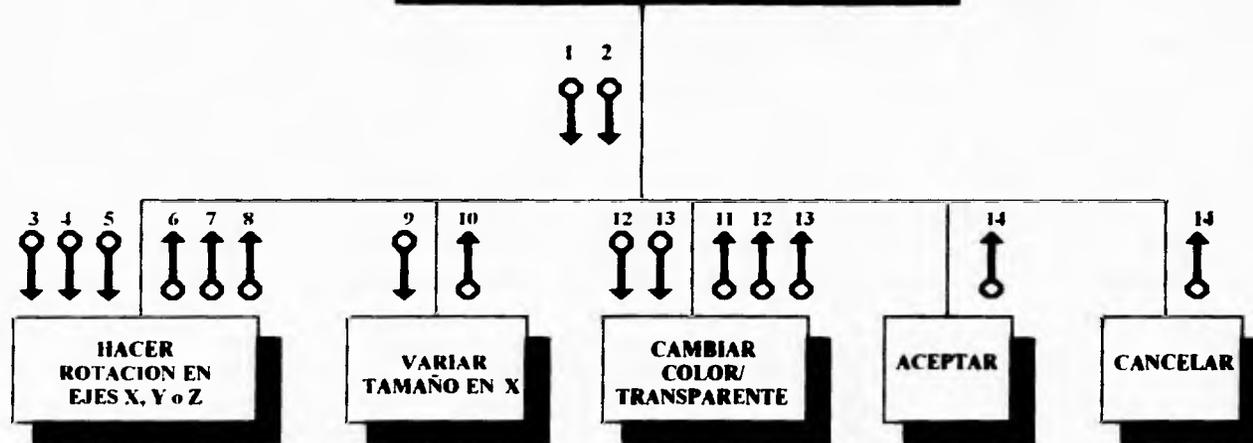
- 9.- VALX
- 10.- VALY
- 11.- XV
- 12.- YV
- 13.- SOL
- 14.- COLORLINEA
- 15.- COLORCARA
- 16.- OK

# DIBUJAR TRIANGULO ISOSELES



- |            |                 |
|------------|-----------------|
| 1.- TIPO   | 9.- VALX        |
| 2.- TITULO | 10.- VALY       |
| 3.- VALR1  | 11.- XV         |
| 4.- VALR2  | 12.- YV         |
| 5.- VALR3  | 13.- SOL        |
| 6.- R1     | 14.- COLORLINEA |
| 7.- R2     | 15.- COLORCARA  |
| 8.- R3     | 16.- OK         |

# DIBUJAR TRIANGULO EQUILATERO



- 1.- TIPO
- 2.- TITULO
- 3.- VALR1
- 4.- VALR2
- 5.- VALR3
- 6.- R1
- 7.- R2

- 8.- R3
- 9.- VALX
- 10.- XV
- 11.- SOL
- 12.- COLORLINEA
- 13.- COLORCARA
- 14.- OK

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

**SELECCIONAR  
POLIGONOS**

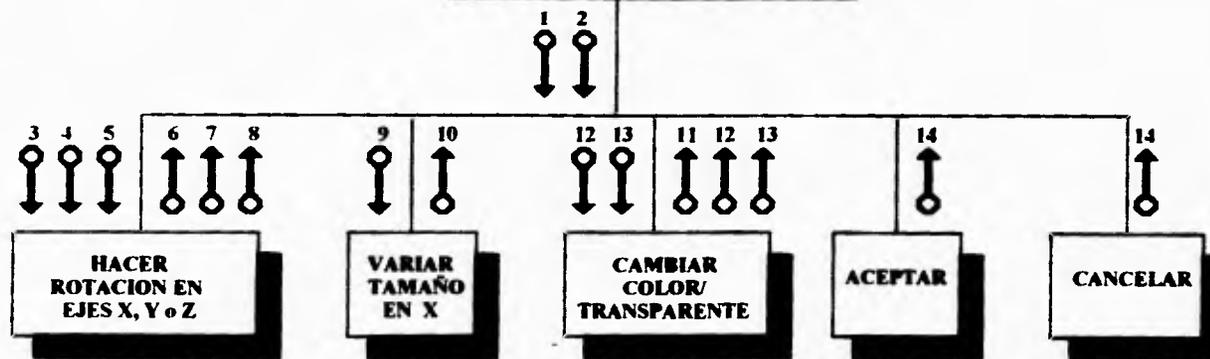
**SELECCIONAR  
EL NUMERO  
DE LADOS**



**DIBUJAR  
POLIGONOS**

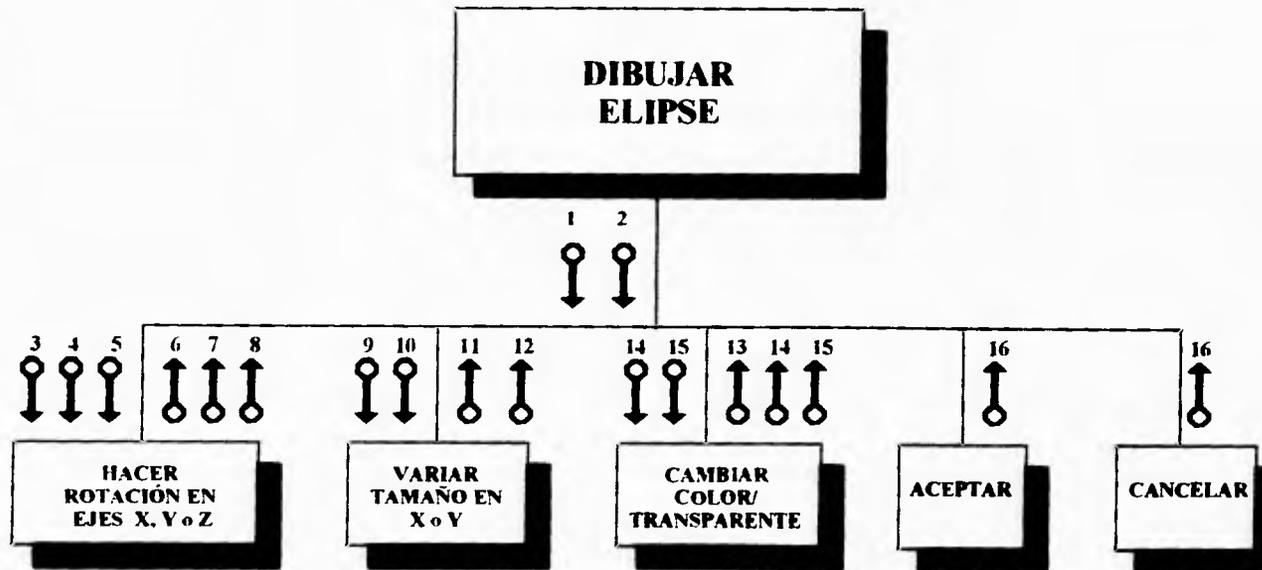
- 1.- TIPO
- 2.- TITULO
- 3.- OK

# DIBUJAR POLIGONOS



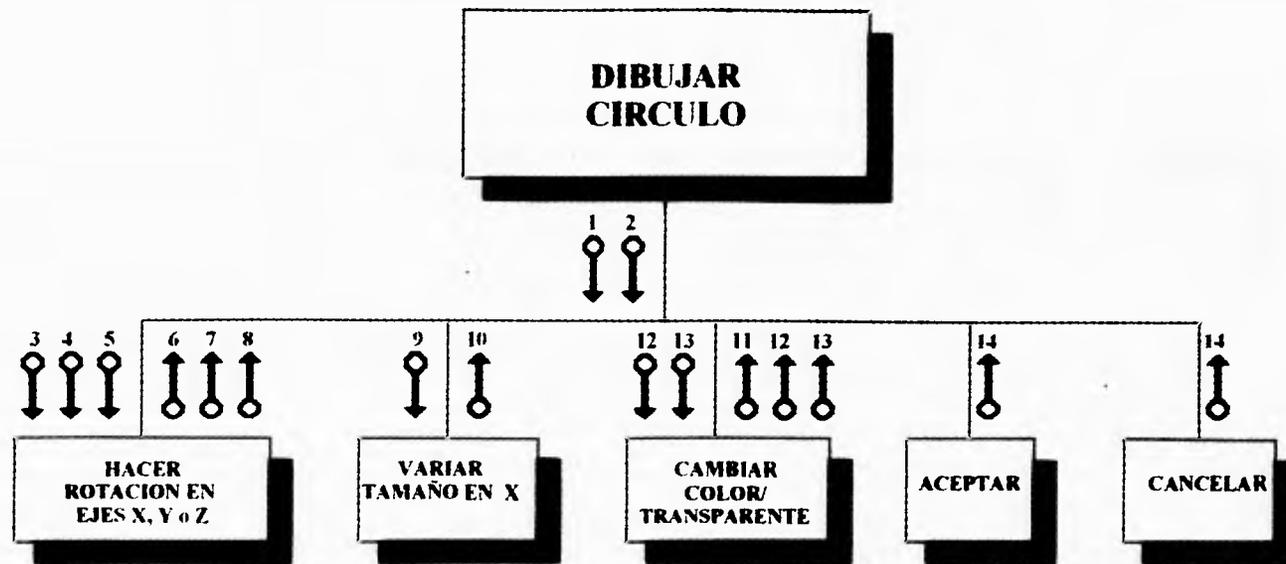
- 1.- TIPO
- 2.- TITULO
- 3.- VALR1
- 4.- VALR2
- 5.- VALR3
- 6.- R1
- 7.- R2

- 8.- R3
- 9.- VALX
- 10.- XV
- 11.- SOL
- 12.- COLORLINEA
- 13.- COLORCARA
- 14.- OK



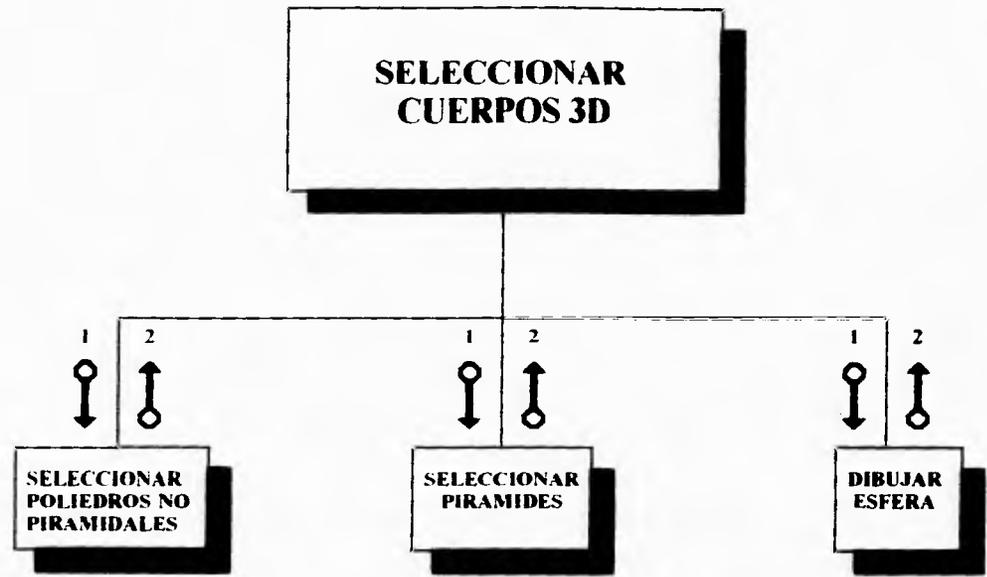
- 1.- TIPO
- 2.- TITULO
- 3.- VALR1
- 4.- VALR2
- 5.- VALR3
- 6.- R1
- 7.- R2
- 8.- R3

- 9.- VALX
- 10.-VALY
- 11.- XV
- 12.- YV
- 13.- SOL
- 14.- COLORLINEA
- 15.- COLORCARA
- 16.- OK



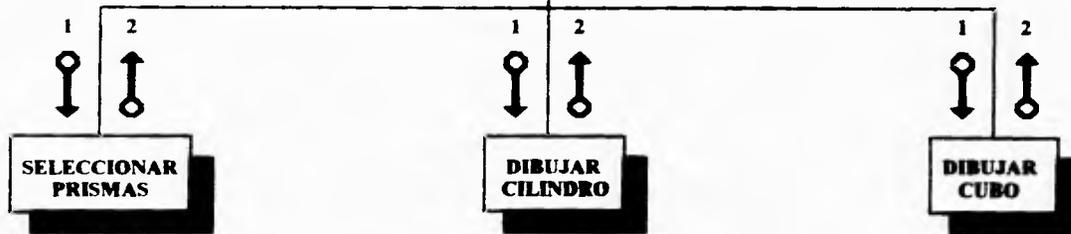
- 1.- TIPO
- 2.- TITULO
- 3.- VALR1
- 4.- VALR2
- 5.- VALR3
- 6.- R1
- 7.- R2

- 8.- R3
- 9.- VALX
- 10.- XV
- 11.- SOL
- 12.- COLORLINEA
- 13.- COLORCARA
- 14.- OK

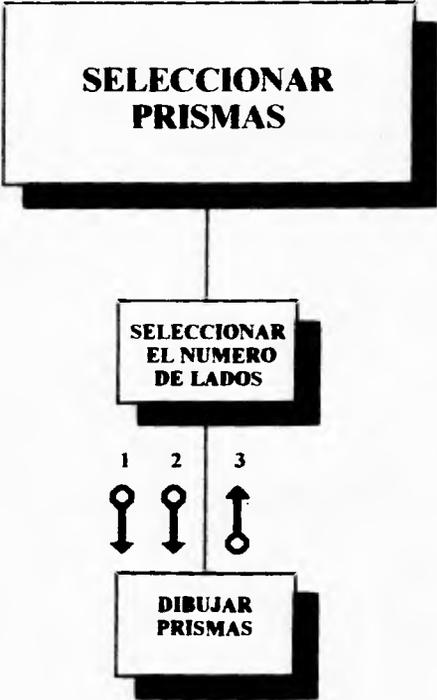


1.- OTRALISTA  
2.- OK

**SELECCIONAR  
POLIEDROS NO  
PIRAMIDALES**

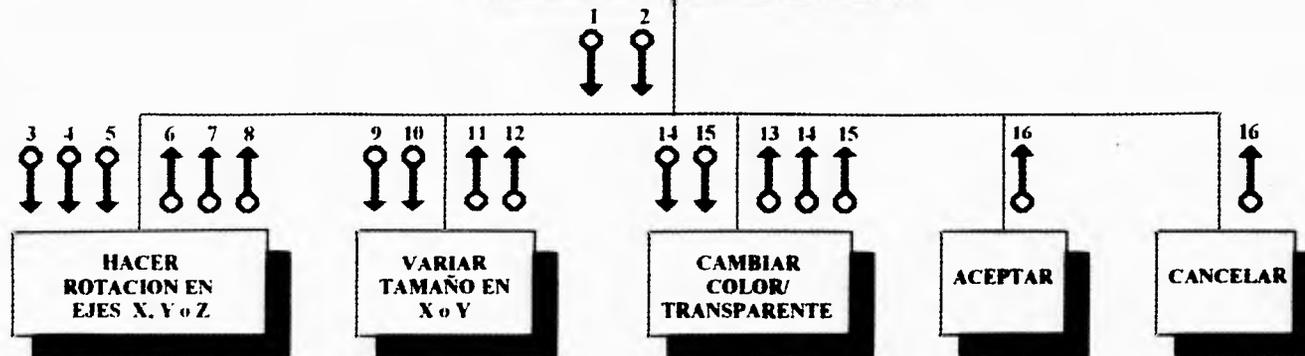


1.- OTRALISTA  
2.- OK



- 1.- TIPO
- 2.- TITULO
- 3.- OK

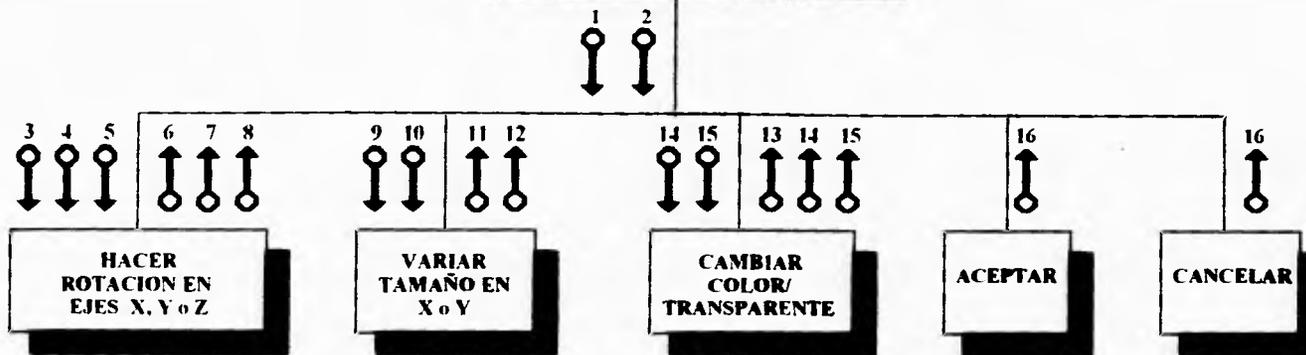
# DIBUJAR PRISMAS



1.- TIPO  
2.- TITULO  
3.- VALR1  
4.- VALR2  
5.- VALR3  
6.- R1  
7.- R2  
8.- R3

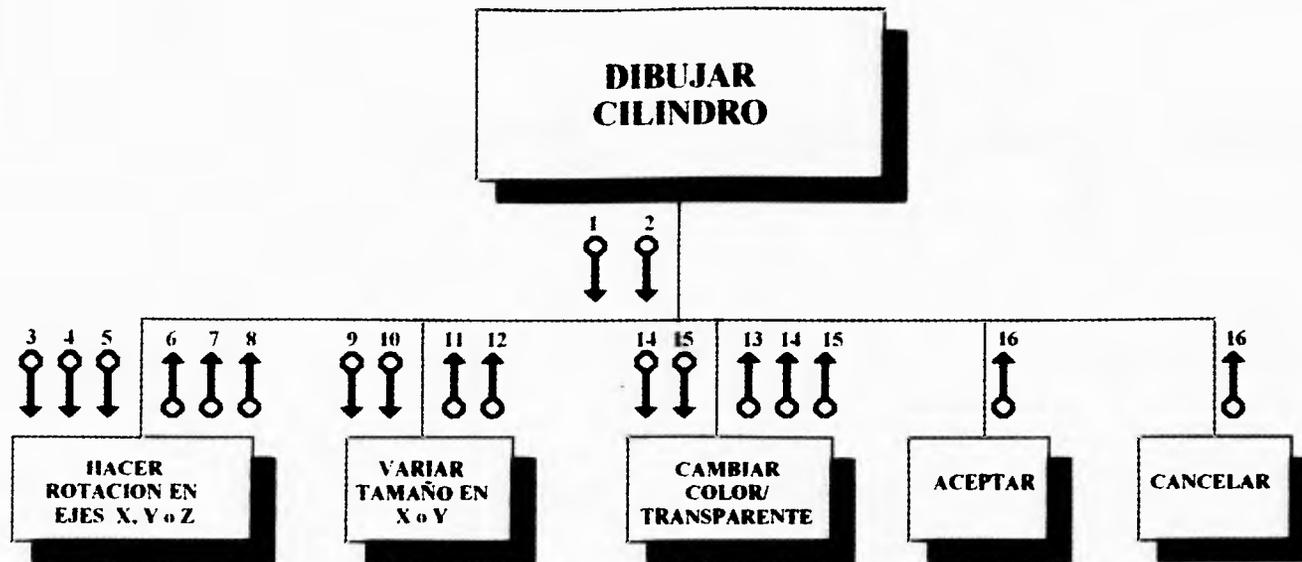
9.- VALX  
10.- VALY  
11.- XV  
12.- YV  
13.- SOL  
14.- COLORLINEA  
15.- COLORCARA  
16.- OK

# DIBUJAR PRISMAS

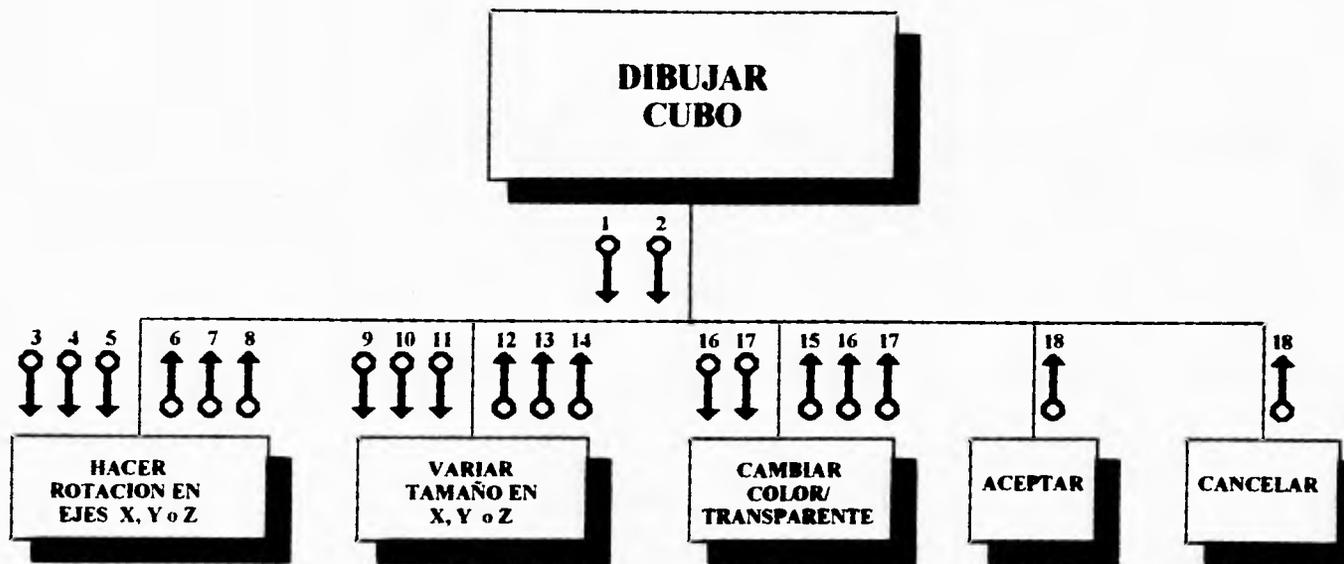


1.- TIPO  
2.- TITULO  
3.- VALR1  
4.- VALR2  
5.- VALR3  
6.- R1  
7.- R2  
8.- R3

9.- VALX  
10.- VALY  
11.- XV  
12.- YV  
13.- SOL  
14.- COLORLINEA  
15.- COLORCARA  
16.- OK



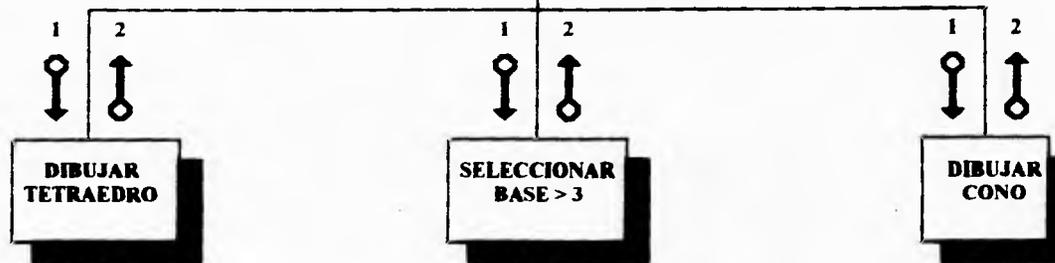
- |            |                 |
|------------|-----------------|
| 1.- TIPO   | 9.- VALX        |
| 2.- TITULO | 10.- VALY       |
| 3.- VALR1  | 11.- XV         |
| 4.- VALR2  | 12.- YV         |
| 5.- VALR3  | 13.- SOL        |
| 6.- R1     | 14.- COLORLINEA |
| 7.- R2     | 15.- COLORCARA  |
| 8.- R3     | 16.- OK         |



- 1.- TIPO
- 2.- TITULO
- 3.- VALR1
- 4.- VALR2
- 5.- VALR3
- 6.- R1
- 7.- R2
- 8.- R3
- 9.- VALX

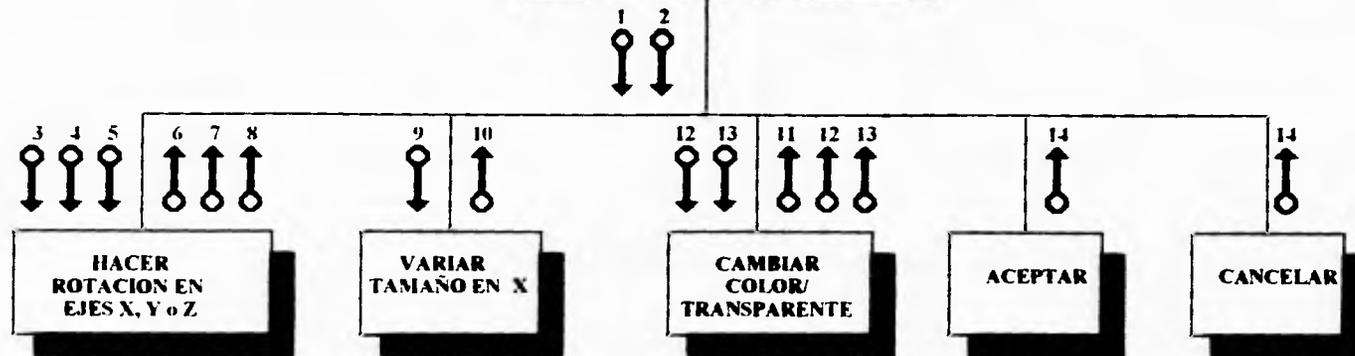
- 10.- VALY
- 11.- VALZ
- 12.- XV
- 13.- YV
- 14.- ZV
- 15.- SOL
- 16.- COLORLINEA
- 17.- COLORCARA
- 18.- OK

# SELECCIONAR PIRAMIDES



1.- OTRALISTA  
2.- OK

# DIBUJAR TETRAEDRO

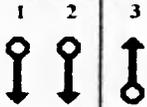


- 1.- TIPO
- 2.- TITULO
- 3.- VALR1
- 4.- VALR2
- 5.- VALR3
- 6.- R1
- 7.- R2

- 8.- R3
- 9.- VALX
- 10.- XV
- 11.- SOL
- 12.- COLORLINEA
- 13.- COLORCARA
- 14.- OK

**SELECCIONAR  
BASE > 3**

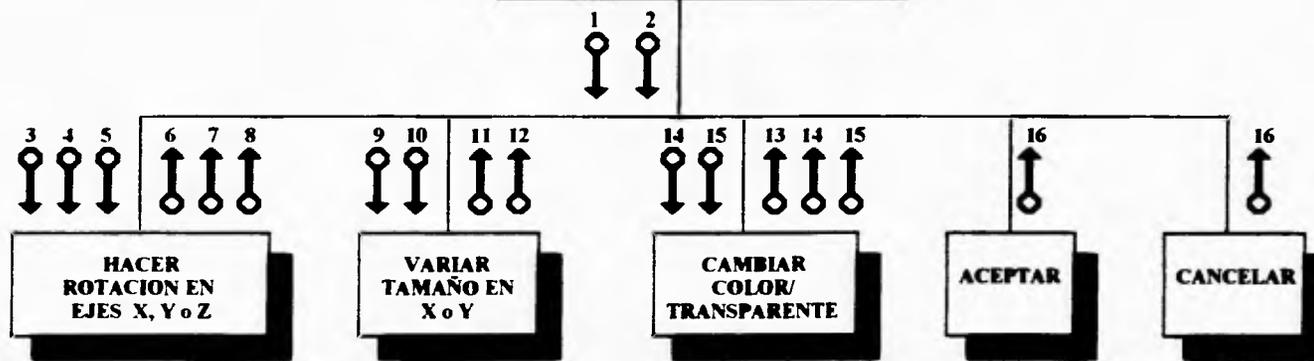
**SELECCIONAR  
EL NUMERO  
DE LADOS**



**DIBUJAR  
PIRAMIDES**

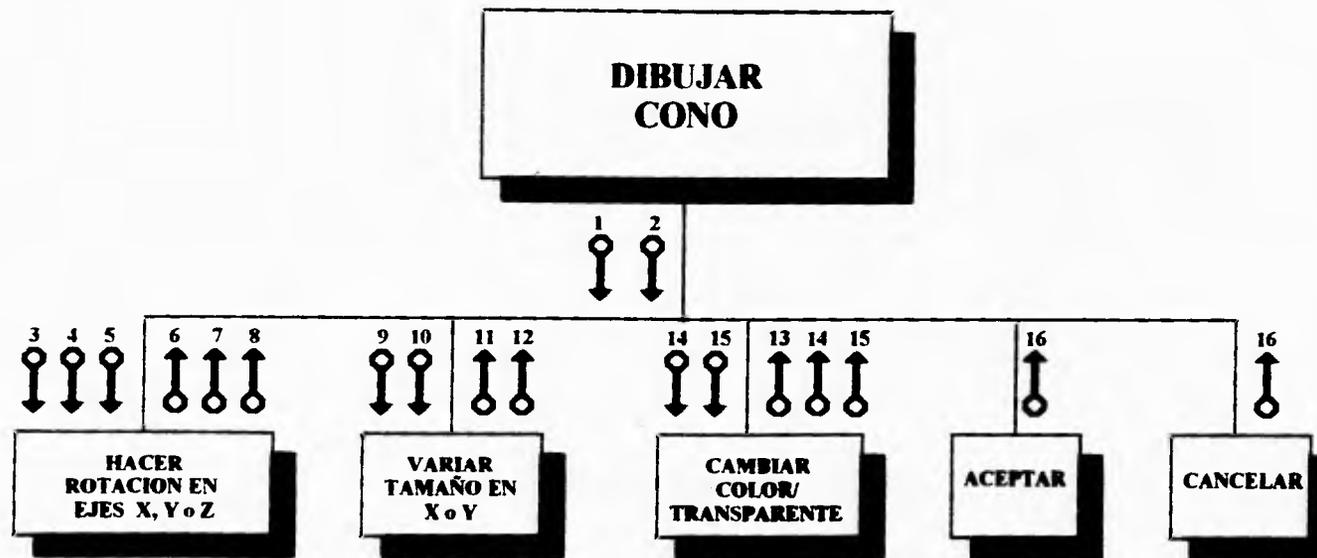
- 1.- TIPO
- 2.- TITULO
- 3.- OK

# DIBUJAR PIRAMIDES



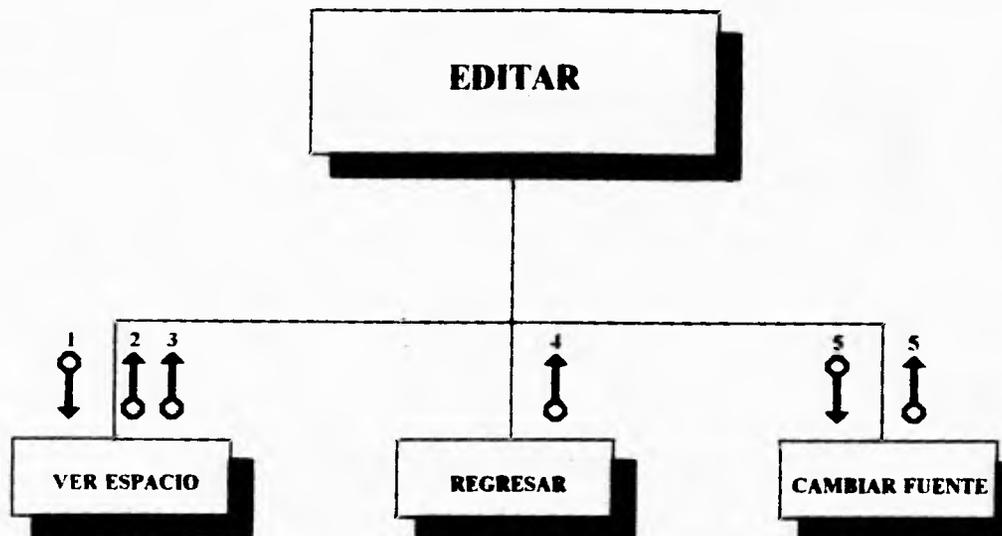
1.- TIPO  
2.- TITULO  
3.- VALR1  
4.- VALR2  
5.- VALR3  
6.- R1  
7.- R2  
8.- R3

9.- VALX  
10.- VALY  
11.- XV  
12.- YV  
13.- SOL  
14.- COLORLINEA  
15.- COLORCARA  
16.- OK

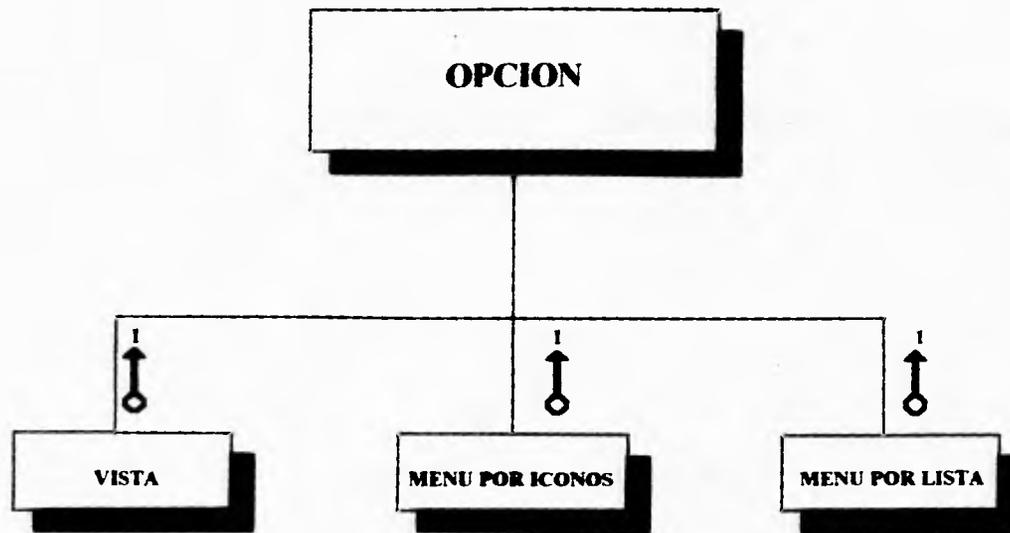


- 1.- TIPO
- 2.- TITULO
- 3.- VALR1
- 4.- VALR2
- 5.- VALR3
- 6.- R1
- 7.- R2
- 8.- R3

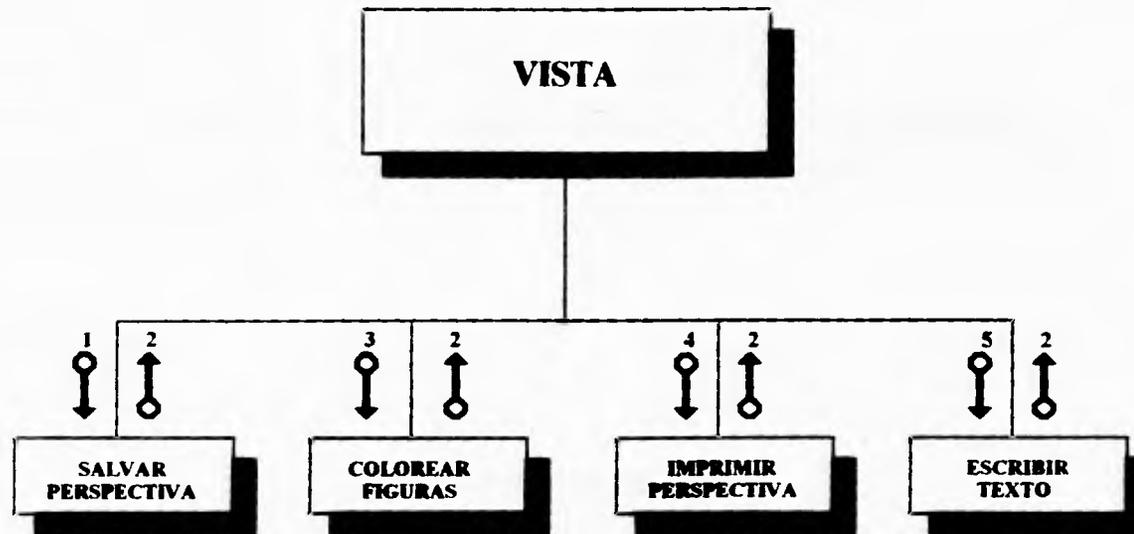
- 9.- VALX
- 10.- VALY
- 11.- XV
- 12.- YV
- 13.- SOL
- 14.- COLORLINEA
- 15.- COLORCARA
- 16.- OK



- 1.- ABRIRARCHIVO
- 2.- FLECHAS
- 3.- ABIERTO
- 4.- OK
- 5.- CARACTS.\_ FUENTE



1-OK



- 1.- SALVAPER
- 2.- OK
- 3.- LEIDO
- 4.- ABRIRARCHIVO
- 5.- LETRERO



## CAPITULO 4

### IMPLEMENTACION



Plano:

Superficie ilimitada que contiene la totalidad de la recta que une dos de sus puntos.

---

## IMPLEMENTACION

**E**l objetivo de esta fase es hablar del lenguaje de programación seleccionado para realizar el sistema, así como los requerimientos de software y hardware que se requieren; mostrar las técnicas de graficación que se utilizaron para elaborar el programa fuente, un pseudocódigo (español estructurado) de algunos procesos que se realizan en el sistema y algunos diagramas del contenido de las FORMAS y como se relacionan éstas entre sí.

### 4.1.- SOFTWARE Y HARDWARE SELECCIONADO

#### 4.1.1.- VISUALBASIC

¿Porqué seleccionar Visual Basic como una herramienta de programación? Se sabe que las interfaces gráficas con el usuario son muy importante hoy en día en cualquier sistema que se esté desarrollando y una de las interfaces de mayor utilidad es Windows dado a su popularidad y facilidad de manejo que ofrece, por lo que buscando entre las múltiples herramientas que se presentan en el mercado Visual BASIC resulto ser la que mas ventajas ofrece, dado que es un interesante y exitoso lenguaje de programación que permite escribir sofisticadas aplicaciones de Windows en un tiempo realmente sorprendente, ya que el sistema de programación de Visual BASIC sortea la complejidad de Window de una manera sencilla.

Visual BASIC desde su lanzamiento al mercado en 1991 fue una brillante solución para todos aquellos que desean ofrecer un producto de calidad para cualquier aplicación. Posteriormente el lanzamiento de la versión 3 introdujo nuevas características y herramientas de trabajo que permiten obtener presentaciones mejoradas, una serie completa de herramientas de creación y manipulación de bases de datos inclusive cuenta con medios de acceso visual a datos mediante un control de datos, de manera que se pueden crear aplicaciones que examinen los datos sin tener que escribir código, pero una de las características mas importantes es que cuenta con un conjunto de cuadros de dialogo estándar que se ocupa de las tareas más habituales de interfaces con el usuario y la posibilidad de crear menús estilo popup en cualquier posición de la aplicación.

Otro factor muy importante es el hecho de que Visual BASIC es también uno de los primeros lenguajes de programación que admite la programación

## **IMPLEMENTACION**

llamada "Orientada a eventos", un estilo de programación especialmente adaptado a los interfaces gráficos de usuario (GUI).

Con Visual Basic pueden crearse poderosas y completas aplicaciones que exploten al máximo todos los recursos que brinda Windows, incluyendo interface de múltiples documentos (MDI), enlace e incrustación de objetos (OLE), intercambio dinámico de datos (DDE), gráficos y más. Y Visual Basic puede extenderse agregando controles externos y por llamada a procedimientos con librerías de enlace dinámico (DLL). También Visual Basic Tiene la opción de crear discos de instalación que incluyan todos los componentes necesarios para una aplicación.

Es por todo esto que se considera que las herramientas, potencia y posibilidades de Visual BASIC hacen de éste una de las herramientas más accesibles para la programación dentro del complejo mundos de las interfaces gráficas.

### **Programación Orientada a Eventos.**

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

Para su programación no se necesita otra cosa que crear una zona de trabajo y asignarle código correspondiente a cada uno de los elementos dibujados en el área de trabajo ( que en Visual Basic se denominan FORMs o formnas). Con solo señalar y hacer doble click en el elemento en el que se quiera introducir código, Visual Basic pone a disposición una ventana en la que se permite al programador especificar qué ocurrirá al accionarse tal elemento.

### **¿Qué significa Orientado a Eventos?**

Es probable que el término "Orientado a Eventos" resulte un tanto confuso para algunos; sin embargo no es complicado. En la denominada programación tradicional" el programador deberá sentarse frente a la computadora y elaborar un código continuo que ésta ejecutará de principio a fin. Tradicionalmente, la programación ha estado muy orientada al proceso, al paso a paso, Uno de los inconvenientes de este estilo consiste en que la persona que escribió el

## CAPITULO 4

programa es la que determina qué hay que hacer en cada momento. Eso puede ser aceptable en algunos casos (muy escasos), 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 programación orientada a eventos se rige por determinadas secciones de código que se acciona **sólo si ocurre una determinada acción**. En lugar de escribir un programa que determine cada uno de los pasos en un orden determinado, el programador escribe un programa que responda a las acciones del usuario: elegir un comando, escribir un texto, hacer click en la 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 una de modo tradicional.

### **Los objetos de Visual Basic**

Una de las ventajas de una interface gráfico de usuario estriba en que el usuario puede interactuar con un conjunto estándar de objetos, como ventanas, botones y barras desplazadoras. Las aplicaciones que utilizan estos objetos se comportan de una manera estándar, haciendo que las aplicaciones sean fáciles de aprender. El trabajo del programador consiste en seleccionar el estilo de interacción que sea más apropiado para la situación y en escribir los programas de una manera intuitiva para el usuario.

Los objetos en Visual Basic constan de *Propiedades*, *Eventos*, *Procedimientos* y *métodos*.

El objeto de mayor jerarquía es la FORMA que corresponde a una ventana cuando se ejecuta una aplicación y algunos ejemplos de sus propiedades son:

**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 es que se desea que la forma no reciba ningún evento del ratón o del teclado.

**FontName** (nombre de la fuente). Esta propiedad se ajusta al nombre de la fuente que se utilizará para escribir en la forma.

**Name** (nombre). Define el nombre de la forma en el código del programa.

## **IMPLEMENTACION**

---

**WindowState** (estado de la ventana). Determina si la ventana aparecera en estado normal, maximizada o minimizada.

Algunos de los eventos con que cuenta son:

**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 una manera muy rápida.

**load** (cargar). Este evento ocurre automáticamente cuando se carga una forma, por ejemplo cuando se arranca una aplicación.

Algunos métodos de las formas son:

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

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

En Visual Basic también existen archivos de aplicación que contienen únicamente código y tiene extensión BAS. Estos son llamados módulos de código y sirven para las 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*.

### **Requerimientos de Hardware.**

Visual Basic está diseñado para correr en un procesador 80286 o mayor con 1MB 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.

Puesto que el lenguaje a utilizar es Visual Basic y éste requiere de Windows y a su vez del sistema operativo DOS se considera conveniente hablar un poco de ellos.

#### **4.1.2.- DOS (Disk Operating System "Sistema Operativo en Disco")**

El DOS es un sistema operativo monousuario para las series PC, PS/1 y PS/2 de IBM, desarrollado por Microsoft. Es a veces denominado PC-DOS, para diferenciarlo del MS-DOS, la versión de Microsoft para las PC no IBM. Ambos productos son prácticamente idénticos y a ambos se les llama DOS.

## **CAPITULO 4**

---

El DOS corresponde al sistema operativo de disco. Es un sistema maestro de control que se ejecuta automáticamente cuando se arranca la computadora. El DOS permanece en la computadora todo el tiempo, permitiendo ejecutar un programa y gestionar archivos.

Para utilizar el DOS, se debe conocer donde se encuentra almacenado el programa y como hablar con el sistema operativo.

### **4.1.3.- Windows**

Entorno operativo para gráficos de Microsoft que se integra con DOS. Proporciona un entorno de sobremesa similar al Macintosh, en el cual cada aplicación activa se visualiza en una pantalla movable y redimensionable sobre la pantalla.

Con objeto de usar todas las funciones del Windows, las aplicaciones deben escribirse específicamente para él. Sin embargo, Windows también ejecuta aplicaciones de DOS y se puede usar como el entorno operativo desde el que se lanzan todos los programas.

## **4.2.- CONCEPTOS DE GRAFICOS 3D**

Los objetos del mundo real tienen tres dimensiones: altura, anchura y profundidad. A las imágenes que aparecen en la pantalla de una computadora simulando estas tres dimensiones se les llama tridimensionales o 3D.

### **4.2.1.- Coordenadas Universales.**

La forma básica del modelo que se desea dibujar viene expresada en coordenadas X,Y,Z. Las tres dimensiones de altura, anchura y profundidad pueden definirse mediante la utilización de un sistema de ejes coordenados. Por convención X representa dimensiones de izquierda-derecha, el eje Y mide lo que está arriba-abajo y el eje Z describe la lejanía-cercanía.

Sin importar cuál sea la forma del modelo, cada posición significativa de un modelo 3D puede describirse como un conjunto de coordenadas X,Y,Z. Cada tríada X,Y,Z describe una sola y única posición dentro del espacio 3D del sistema de coordenadas.

El primer paso en el modelo de un objeto es definir sus coordenadas universales. Estas delimitan la forma básica o el diseño del modelo. Las coordenadas universales reciben también el nombre de coordenadas absolutas, coordenadas cartesianas y coordenadas espaciales del modelo.

## IMPLEMENTACION

El sistema de ejes X,Y,Z se llama espacio de coordenadas universales. Representa el mundo real, independiente del programa e independientemente de la computadora que se utilice.

### 4.2.2.- Generación de Modelos 3D.

Existen tres fases conceptuales en la generación de modelos 3D en una computadora personal (ver fig. 4.1). Primero, la estructura fundamental del modelo se define en **COORDENADAS UNIVERSALES**. Segundo, este modelo fundamental es girado y después trasladado a una nueva posición dentro del sistema de ejes 3D para producir la vista deseada (el punto de vista siempre está situado en las coordenadas 0,0,0 del sistema de ejes 3D). A este giro se le llama rotación. El movimiento recibe el nombre de traslación. La rotación y la traslación producen las **COORDENADAS VISUALES**. Tercero, se aplican fórmulas de proyección para dibujar las imágenes en el monitor, creando así las **COORDENADAS DE PANTALLA**.

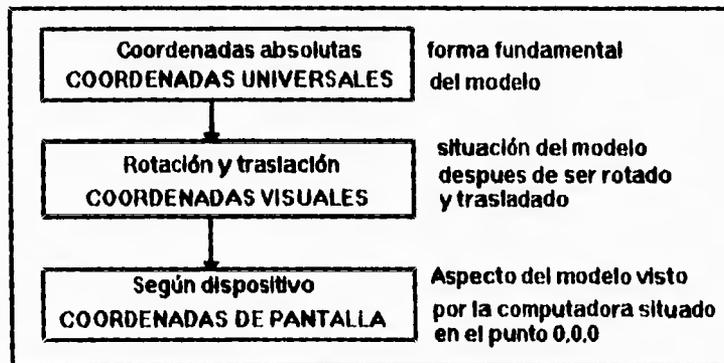


Fig. 4.1

Las **COORDENADAS UNIVERSALES** y las **COORDENADAS VISUALES** son coordenadas independientes del dispositivo. En cambio las **COORDENADAS DE PANTALLA** si son dependientes del dispositivo.

Las **COORDENADAS UNIVERSALES** consisten en tríadas de coordenadas X,Y,Z. Las **COORDENADAS VISUALES** son tríadas de coordenadas X,Y,Z giradas y trasladadas. Las **COORDENADAS DE PANTALLA** 2D finales son pares de coordenadas X,Y ya que la pantalla admite solamente una coordenada X y otra Y para trazar puntos.

## **CAPITULO 4**

### **4.2.3.- Componentes del Modelo.**

Todo modelo 3D contiene ciertos elementos como puede verse en la figura 4.2. Un vértice es una esquina: la intersección de dos o más líneas. Los vértices son grupos de intersecciones. a un área del modelo se le llama cara. Las caras pueden ser planas o curvas, rectangulares o poligonales (regulares o irregulares), visibles u ocultas. Una cara invisible recibe el nombre de cara oculta. Al algoritmo que calcula si una cara dada puede o no ser observada (desde un determinado punto de vista) se le llama rutina de eliminación de caras ocultas. Del mismo modo, una línea que no es visible recibe el nombre de línea oculta. Cualquier línea en un modelo 3D puede llamarse arista.

Las partes más importantes de un modelo 3D son los vértices y las caras. Si se conocen las coordenadas de los vértices se puede dibujar el modelo; y se puede determinar la orientación de las diversas caras del mismo.

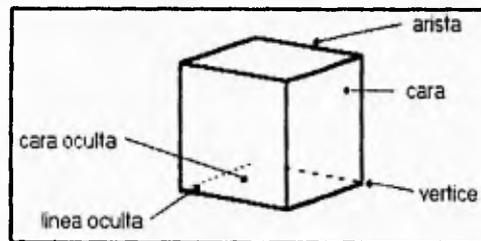


Fig. 4.2

### **4.2.4.- Las matemáticas de las 3D.**

En términos generales las matemáticas necesarias para crear y manipular modelos tridimensionales en computadora son muy simples. Obviamente, no se requieren fórmulas matemáticas para introducir las coordenadas universales.

Para rotar el objeto hay que realizar operaciones con matrices y utilizar las funciones trigonométricas de seno y coseno. Para la traslación solo son necesarias sumas y restas. Se utiliza la geometría para transformar la imagen 3D en una pantalla 2D.

### **4.2.5.- Manipulación de los Modelos.**

Las **COORDENADAS UNIVERSALES** de un modelo describen la forma o el diseño fundamental del objeto, El modelo debe ser manipulado a fin de conseguir la vista deseada.

## IMPLEMENTACION

El siguiente algoritmo proporciona una flexibilidad completa para la manipulación de los modelos y es el utilizado en el programa.

$$\begin{array}{ll}
 XA=CR1*X-SR1*Z; ZA=SR1*X+CR1*Z & \longrightarrow \text{rotación de deriva} \\
 X=CR2*XA+SR2*Y; YA=CR2*Y-SR2*XA & \longrightarrow \text{rotación de balanza} \\
 Z=CR3*ZA-SR3*YA; Y=SR3*ZA+CR3*YA & \longrightarrow \text{rotación de caída} \\
 X=X+MX; Y=Y+MY; Z=Z+MZ & \longrightarrow \text{traslación} \\
 SX=D*X/Z; SY=D*Y/Z & \longrightarrow \text{proyección}
 \end{array}$$

Las variables X,Y,Z son las **COORDENADAS UNIVERSALES**. Las variables XA, YA, ZA son variables temporales que acumulan los valores cambiantes de las **COORDENADAS UNIVERSALES** según van pasando de una fórmula a otra para convertirse en **COORDENADAS VISUALES**.

### **Fórmulas de Rotación.**

La rotación de un punto en el espacio 3D se implementa a través de multiplicación de matrices. Las anteriores fórmulas de rotación se obtienen de la siguiente manera:

Deriva: Rotación alrededor del eje Y

$$[X,Y,Z,1] \begin{bmatrix} \text{COS} & 0 & \text{SEN} & 0 \\ 0 & 1 & 0 & 0 \\ -\text{SEN} & 0 & \text{COS} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Balanza: Rotación alrededor del eje Z

$$[X,Y,Z,1] \begin{bmatrix} \text{COS} & \text{SEN} & 0 & 0 \\ -\text{SEN} & \text{COS} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Caída: Rotación alrededor del eje X

$$[X,Y,Z,1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{COS} & \text{SEN} & 0 \\ 0 & -\text{SEN} & \text{COS} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## **CAPITULO 4**

La primera manipulación efectuada por el algoritmo es la rotación. Al modelo se le somete a un giro de deriva, que es un giro a la derecha o a la izquierda (rotación alrededor del eje Y). La variable SR1 de la fórmula almacena el seno del ángulo de deriva R1 que ha sido declarado (o calculado) previamente en otra parte del programa. La variable CR1 toma el valor del coseno del ángulo de deriva R1.

La siguiente rotación que se realiza es el balanceo, que es un giro equivalente al de las agujas de un reloj (rotación alrededor del eje Z). Las variables SR2 y CR2 adquieren el valor del seno y del coseno, respectivamente, del ángulo de balanza R2.

La última rotación que se realiza es la caída, que es un giro hacia atrás o hacia adelante (rotación alrededor del eje X). Las variables SR3 y CR3 son el seno y el coseno, respectivamente, del ángulo de caída R3.

### **Fórmulas de Traslación.**

La segunda manipulación que hace el algoritmo es la traslación. El modelo puede ser trasladado, o movido, a lo largo de los tres ejes en el sistema de coordenadas **3D**. Como el punto de vista siempre está situado en el origen 0,0,0 del sistema de coordenadas, el modelo debe ser movido a una situación que dé el punto de vista apropiado.

La variable MX define la distancia recorrida a través del eje X. Esta variable ha sido definida con anterioridad. La variable MY se refiere a la distancia que debe moverse el objeto a través del eje Y, arriba o abajo. La variable MZ designa la distancia del movimiento por el eje Z, que es un movimiento de aproximación o alejamiento.

### **Fórmulas de Proyección.**

La tercera gran manipulación realizada por el algoritmo es la proyección. Las **COORDENADAS VISUALES** que han sido creadas por las fórmulas de giro y traslación deben ser modificadas para que puedan ser trazadas en la pantalla.

La geometría implicada en la conversión de las **COORDENADAS VISUALES 3D** a las **COORDENADAS DE PANTALLA** en el espacio **2D** para su trazado en el monitor aparece en las figuras 4.3 y 4.4. Como puede observarse, la figura 4.3 concierne a la coordenada y de la pantalla, que recibe el nombre de SY. La figura 4.4 se refiere a la coordenada x de la pantalla, y recibe el nombre de SX.

## IMPLEMENTACION

El algoritmo geométrico en el que se basan las fórmulas de proyección asume que una pantalla plana y transparente de **2D** se sitúa entre el observador y el modelo **3D** en cuestión. Esta pantalla transparente recibe el nombre de plano del dibujo. La distancia entre el observador y el objeto viene expresada por **Z**, que es la coordenada que describe la cercanía o lejanía. La distancia entre el observador y la pantalla se denota como **D**.

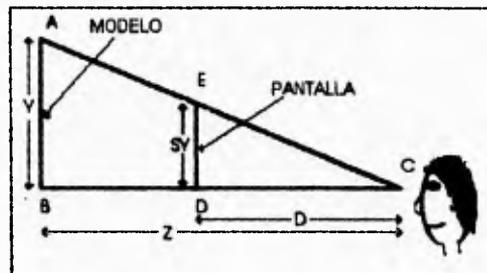


Fig. 4.3

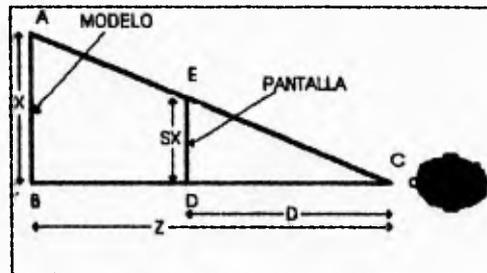


Fig. 4.4

Como todo este escenario puede ser interpretado por dos triángulos rectángulos, las relaciones entre los diversos lados y ángulos son proporcionales. En la figura 4.3, por ejemplo, los valores **Y**, **Z** y **D** son conocidos. Resulta por tanto, relativamente sencillo utilizar la geometría para resolver el valor buscado de **SY** (de la **COORDENADA DE PANTALLA**).

$SY/D=Y/Z$  expresa la igualdad entre el cociente de los lados de los dos triángulos rectángulos de la figura 4.3. Despejando, llegamos a  $SY=D*Y/Z$ . Este mismo criterio se aplica para los triángulos de la figura 4.4 para obtener la coordenada **SX**.

### 4.2.6.- Radianes Frente a Grados.

Los ángulos de deriva, balanza y caída siempre se miden en radianes cuando se utilizan fórmulas 3D. Así como hay 360 grados en un círculo

## CAPITULO 4

completo, hay  $2\pi$ , o aproximadamente 6.28 radianes en un círculo. Los radianes se basan en la relación existente entre los diversos elementos que se encuentran en el círculo unidad. Este es simplemente aquél de radio uno.

La fórmula de conversión de grados a radianes es:

$$RADIANES = \#GRADOS * \pi / 180$$

### 4.2.7.- La Secuencia Rotación-Traslación.

La secuencia de las fórmulas de rotación que se muestran en el algoritmo dan una determinada orientación del modelo 3D que se está manipulando; si estas operaciones son intercambiadas en el algoritmo producirán diferentes resultados en la pantalla (ver figura 4.5), pero esto no quiere decir que ese resultado sería incorrecto. Lo único que pasa es que cambia el orden de las rotaciones. Por lo tanto se puede decir que estas operaciones no son conmutativas.

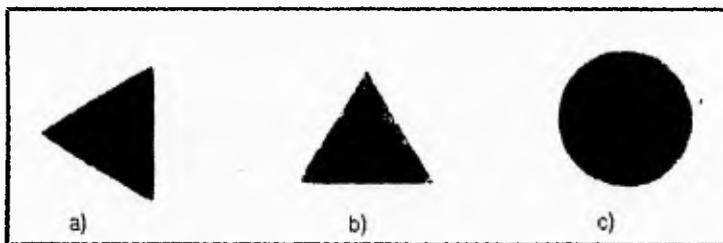


figura 4.5. Diferentes secuencias del cálculo de las rotaciones y sus respectivas salidas mostradas en pantalla, al manipular un cono si son implementadas en ese orden en el algoritmo.

a) Deriva, Balanza y Caída; b) Caída, Deriva y Balanza; c) Balanza, Caída y Deriva

Si el modelo es trasladado antes que rotado, también habrá un resultado diferente en la pantalla, ya que da la apariencia de que cambió el punto de vista.

### ELIMINACIÓN DE CARAS OCULTAS

Existen principalmente tres métodos para suprimir las caras ocultas: *preselección radial*, *selección radial* y *ecuación del plano*. Los tres métodos se basan en las superficies del cubo y emplean el principio de eliminación de las caras de atrás. Estos métodos son explicados en el apéndice B.

## IMPLEMENTACION

De los tres métodos, el más conveniente es el de la ecuación del plano, y es el utilizado para la solidificación de las figuras 3D del sistema.

Cuando en un diseño se necesita que una figura esté adelante de otra, el sistema utiliza el método de selección radial, dibujando primero las figuras más lejanas para ser tapadas por las más cercanas, si es que así debe ser. Para saber el orden de las figuras se utiliza el método del embudo. Este y algunos otros métodos son explicados en el apéndice B.

El siguiente algoritmo es el encargado de determinar la visibilidad de una cara y es el utilizado en el programa.

$$SP1 = X1*(Y2*Z3 - Y3*Z2); SP1 = -SP1$$

$$SP2 = X2*(Y3*Z1 - Y1*Z3)$$

$$SP3 = X3*(Y1*Z2 - Y2*Z1)$$

$$SP = SP1 - SP2 - SP3$$

Si  $SP < 0$  entonces la cara es visible

Las fórmulas utilizadas por el anterior algoritmo se basan en la aritmética matricial, como se explica a continuación en la rutina de supresión de caras ocultas.

### **Rutina de supresión de caras ocultas**

La rutina de supresión de caras ocultas emplea la ecuación del plano para comprobar la visibilidad (ver fig. 4.6). Como el punto de vista del usuario está siempre situado en el origen 0,0,0 la posición de este punto relativo a la orientación de la superficie puede utilizarse para determinar si una cara esta oculta o es visible.

Se usa el calculo matricial para calcular las constantes A, B, C, D de la ecuación del plano al que pertenece la cara como se indica a continuación:

$$A = Y1(Z2 - Z3) + Y2(Z3 - Z1) + Y3(Z1 - Z2)$$

$$B = Z1(X2 - X3) + Z2(X3 - X1) + Z3(X1 - X2)$$

$$C = X1(Y2 - Y3) + X2(Y3 - Y1) + X3(Y1 - Y2)$$

$$D = -X1(Y2Z3 - Y3Z2) - X2(Y3Z1 - Y1Z3) - X3(Y1Z2 - Y2Z1)$$

## CAPITULO 4

$$AX + BY + CZ + D = 0$$

ECUACION DEL PLANO

con X,Y,Z un punto en la superficie del plano

A,B,C,D, son constantes derivadas de los tres puntos del plano

Si  $AX + BY + CZ + D > 0$  entonces X,Y,Z está en la superficie interior del poliedro convexo

Si  $AX + BY + CZ + D < 0$  entonces X,Y,Z está en la superficie exterior del poliedro convexo

Fig. 4.6

Las coordenadas X,Y,Z necesarias para calcular estas constantes deben recibirse en sentido contrario a las manecillas del reloj según se avance por el perímetro del plano.

Entonces, si la ecuación da 0 cuando se usa un conjunto de coordenadas X,Y,Z para valorarla, el punto definido por dichas coordenadas pertenece al plano. Si la ecuación produce un resultado mayor que cero, el punto está en el interior del poliedro convexo (la cara está oculta). Si es menor que cero, el punto está visible en la parte exterior a la superficie de un poliedro convexo (la cara es visible).

De lo anterior se concluye que  $SP = D$ , y las coordenadas  $X_1, Y_1, Z_1$  junto con las coordenadas  $X_2, Y_2, Z_2$  son los extremos del vector1 y las coordenadas  $X_3, Y_3, Z_3$  junto con las coordenadas  $X_2, Y_2, Z_2$  son los extremos del vector2, estos dos vectores son las aristas de la cara que se está analizando como se ilustra en la figura 4.7

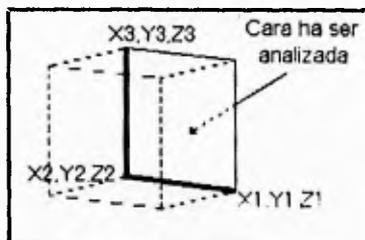


Fig. 4.7

## IMPLEMENTACION

### **4.3.- DESCRIPCION ESTRUCTURADA DE PROCESOS (ESPAÑOL ESTRUCTURADO) DEL SISTEMA PROPUESTO**

La sintaxis del español estructurado está fundamentada en las cuatro estructuras básicas que definen al flujo de datos que son:

- 1) Secuencia . La secuencia está dada por una serie de frases simples o instrucciones .
- 2) Repetición. La repetición está dada mediante las palabras clave:

**REPITE**

---

--- Proceso a repetir

---

**HASTA** Condición de fin de repetición

o

**MIENTRAS** Condición de fin de repetición

---

--- Proceso a repetir

---

**FIN DE MIENTRAS** Marca el fin del proceso a repetir

- 3) Selección. La selección ocurre:

- a) Cuando es una alternativa.

**SI CONDICION**

---

--- Proceso de condición verdadera

---

**EN CASO CONTRARIO**

---

--- Proceso de condición falsa

---

**FIN DE CONDICION**

- b) Cuando son varias alternativas para un caso específico.

**CASO** Variable

**CASO 1**

**Condición 1**

---

## **CAPITULO 4**

---

```

          --- Proceso si condición 1
          ---
CASO 2    Condicion 2
          ---
          --- Proceso si condición 2
          ---
EN OTRO CASO
          ---
          --- Proceso si otra condición
          ---
FIN DE DECISION
```

4) Ciclo. Un ciclo se representa por medio de:

```

CICLO DE variable de control DESDE inicio HASTA fin
  --- Bloque de instrucciones a ejecutar
FIN DE CICLO variable de control
```

El español estructurado de los procesos que controlan el dibujo de un cubo, en las expansiones del nivel 2.0 de los DLFD del sistema propuesto, se muestran a continuación.

### **Diseño general de las figuras**

Ejecuta el Procedimiento form\_load

Ejecuta el Procedimiento form\_activate

Ejecuta el procedimiento DECIDE

Repite

Identifica cambios en algún objeto de selección

Caso Objeto con cambio

```

TAMAÑO      : Ejecuta Procedimiento DECIDE
ROTACION    : Ejecuta Procedimiento DECIDE
TRANSPARENTE : Ejecuta Procedimiento bt3dtrans
COLOR       : Ejecuta Procedimiento bt3dsolido
ACEPTAR     : Ejecuta Procedimiento
CANCELAR    : Ejecuta Procedimiento
```

Hasta opción CANCELAR o ACEPTAR

## **IMPLEMENTACION**

### **Procedimiento DECIDE**

Si la figura a dibujar es un polígono

    Fija los valores máximos de las Barras Desplazadoras para el tamaño

    Ejecuta el procedimiento VERTPOLI

En caso contrario

    Fija los valores máximos de las Barras Desplazadoras para el tamaño

    Ejecuta el procedimiento VERTCUB

Fin Si

Fin de Procedimiento

### **Procedimiento DIBCUBO**

Fija el color y el tipo de línea

Traza las cuatro líneas de una cara

Si se seleccione figura sólida

    Rellena la cara

Fin Si

Fin de Procedimiento

### **Procedimiento VERTCUB**

Limpia Pantalla

Define los vectores B1, B2 y B3

Inicializa los valores de los vectores B1 y B2

Ciclo de i desde 0 hasta 7

    Ejecuta Procedimiento PAR3D

    Actualiza los valores de los vectores B1 y B2

Fin de ciclo i

Ciclo de i desde 0 hasta 5

    Ejecuta procedimiento PAR3D

    Actualiza los valores del vector B3

Fin de ciclo i

'CARA 1

Asigna a las variables X, Y y Z los valores del vector B1

Si se eligió figura sólida

    Ejecuta Procedimiento ECPLANO

Fin Si

## **CAPITULO 4**

---

**Si la cara es visible**

**Asigna a las variables X y Y los valores de los vectores B2 y B3**

**Ejecuta Procedimiento DIBCUB**

**Fin Si**

**'CARA 2**

**Asigna a las variables X, Y y Z los valores de los vectores B1**

**Si se eligió figura sólida**

**Ejecuta Procedimiento ECPLANO**

**Fin Si**

**Si la cara es visible**

**Asigna a las variables X y Y los valores de los vectores B2 y B3**

**Ejecuta Procedimiento DIBCUB**

**Fin Si**

**'CARA 3**

**Asigna a las variables X, Y y Z los valores de los vectores B1**

**Si se eligió figura sólida**

**Ejecuta Procedimiento ECPLANO**

**Fin Si**

**Si la cara es visible**

**Asigna a las variables X y Y los valores de los vectores B2 y B3**

**Ejecuta Procedimiento DIBCUB**

**Fin Si**

**'CARA 4**

**Asigna a las variables X, Y y Z los valores de los vectores B1**

**Si se eligió figura sólida**

**Ejecuta Procedimiento ECPLANO**

**Fin Si**

**Si la cara es visible**

**Asigna a las variables X y Y los valores de los vectores B2 y B3**

**Ejecuta Procedimiento DIBCUB**

**Fin Si**

**'CARA 5**

**Asigna a las variables X, Y y Z los valores de los vectores B1**

**Si se eligió figura sólida**

**Ejecuta Procedimiento ECPLANO**

**Fin Si**

**Si la cara es visible**

**Asigna a las variables X y Y los valores de los vectores B2 y B3**

**Ejecuta Procedimiento DIBCUB**

## **IMPLEMENTACION**

Fin Si  
'CARA 6  
Asigna a las variables X, Y y Z los valores de los vectores B1  
Si se eligió figura sólida  
Ejecuta Procedimiento ECPLANO  
Fin Si  
Si la cara es visible  
Asigna a las variables X y Y los valores de los vectores B2 y B3  
Ejecuta Procedimiento DIBCUB  
Fin Si

Fin de Procedimiento

### **Procedimiento VERTPOLI**

Limpia Pantalla  
Calcula el número de caras  
Calcula el número de vertices  
Define los vectores B1, B2 y B3  
Calcula el incremento del ángulo  
Ciclo de i desde 1 hasta número de lados  
Inicializa los valores del vector B1  
Fin de ciclo i  
Ciclo de i desde 1 hasta número de lados  
Inicializa los valores del vector B3  
Fin de ciclo i  
Ciclo i desde 1 hasta número de vértices  
Ejecuta el Procedimiento PAR3D  
Actualiza los valores de los vectores B1 y B2  
Fin de ciclo i  
Ciclo de i desde 1 hasta número de caras  
Ejecura el procedimiento PAR3D  
Actualiza los valores del vector B3  
Fin de ciclo i  
' Para todas las caras menos la sup. e inf.  
Ciclo de i desde 1 hasta número de caras - 2  
si i = número de lados  
Asigna a las variables X, Y y Z los valores de los vectores B1  
Si se eligió figura sólida

## **CAPITULO 4**

---

Ejecuta Procedimiento ECPLANO

Fin Si

Si la cara es visible

Asigna a X, Y y Z los valores de los vectores B2 y B3

Ejecuta el Procedimiento DIBCUB

Fin Si

En caso contrario

Asigna a las variables X, Y y Z los valores de los vectores B1

Si se eligió figura sólida

Ejecuta Procedimiento ECPLANO

Fin Si

Si la cara es visible

Asigna a X, Y y Z los valores de los vectores B2 y B3

Ejecuta el Procedimiento DIBCUB

Fin Si

Fin Si

Fin de ciclo i

' Para la cara superior

Asigna a las variables X, Y y Z los valores del vector B1

Si se eligió figura sólida

Ejecuta Procedimiento ECPLANO

Fin Si

Si la cara es visible

Ciclo de i desde 1 hasta número de lados - 1

Dibuja línea

Fin de ciclo i

Dibuja línea

Si se eligió figura sólida

Rellena el área al rededor del punto

B3(numlad + 1, 1), B3(numlad + 1, 2)

Fin Si

Fin Si

' Para la cara inferior

Asigna a las variables X, Y y Z los valores del vector B1

Si se eligió figura sólida

Ejecuta Procedimiento ECPLANO

Fin Si

Si la cara es visible

Ciclo de i desde número de lados + 1 hasta número de vértices - 1

## **IMPLEMENTACION**

---

Dibuja línea  
Fin de ciclo i  
Dibuja línea  
Si se eligió figura solida  
Rellena el area al rededor del punto  
B3(numlad + 1, 1), B3(numlad + 1, 2)  
Fin Si  
Fin Si  
Fin de Procedimiento

### **Procedimiento bt3dsolido\_click**

Ejecuta Procedimiento COLORES  
Ejecuta Procedimiento DECIDE  
Fin de Procedimiento

### **Procedimiento bt3dtrans\_click**

Fija tipo y color de línea  
Ejecuta Procedimiento DECIDE  
Fin de Procedimiento

### **Procedimiento form\_activate**

Fija condiciones iniciales  
Ejecuta Procedimiento DECIDE  
Fin de Procedimiento

### **Procedimiento form\_load**

Fija tamaño y posición de los objetos que se presentan en la forma  
Fin de Procedimiento

### **Procedimiento COLORES**

Activa Dialog1

## CAPITULO 4

---

Obtiene el Color de la cara

Si Cambia Color de línea

REPITE

Activa Dialog1

Obtiene el color de la línea

Si color de línea es blanco

Color no disponible para el contorno

Fin si

HASTA color de línea  $\neq$  blanco

Fin si

Fin de Procedimiento

### 4.4.- DIAGRAMAS GENERALES DE ACCIONES Y PROCESOS

Diagrama de Acciones Generales del Sitema

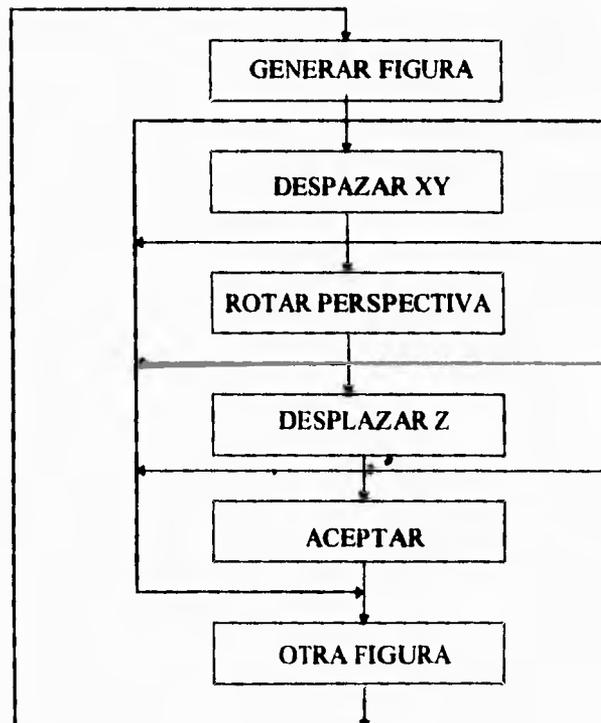
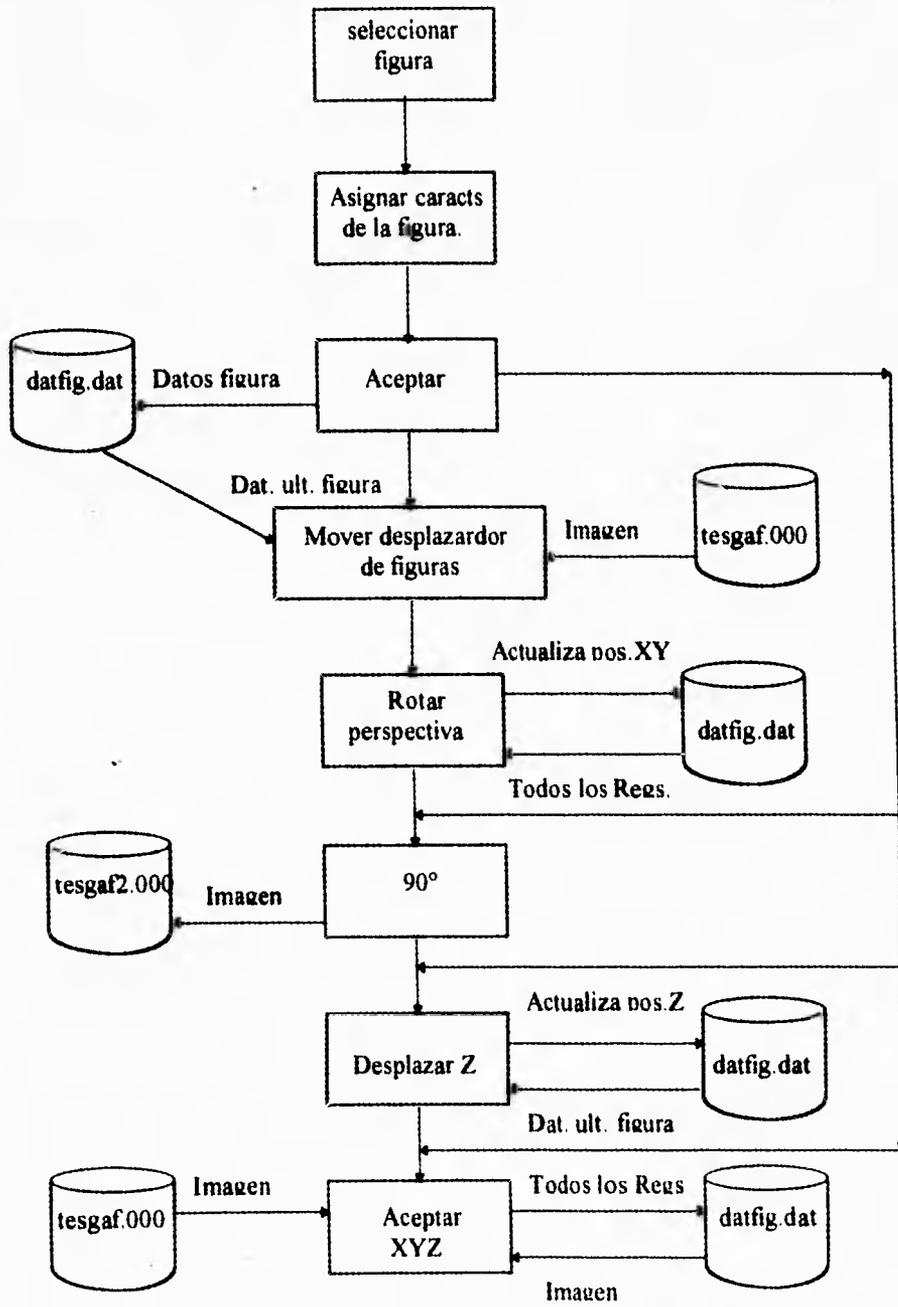


Diagrama General de los Procesos del Sistema



## **CAPITULO 4**

---

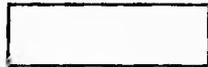
### **4.5.- DIAGRAMAS DE FORMAS DEL SISTEMA**

En las siguientes páginas se presentan algunos diagramas de las FORMAS pertenecientes al sistema.

Para los cuatro últimos diagramas los siguientes símbolos usados significan:

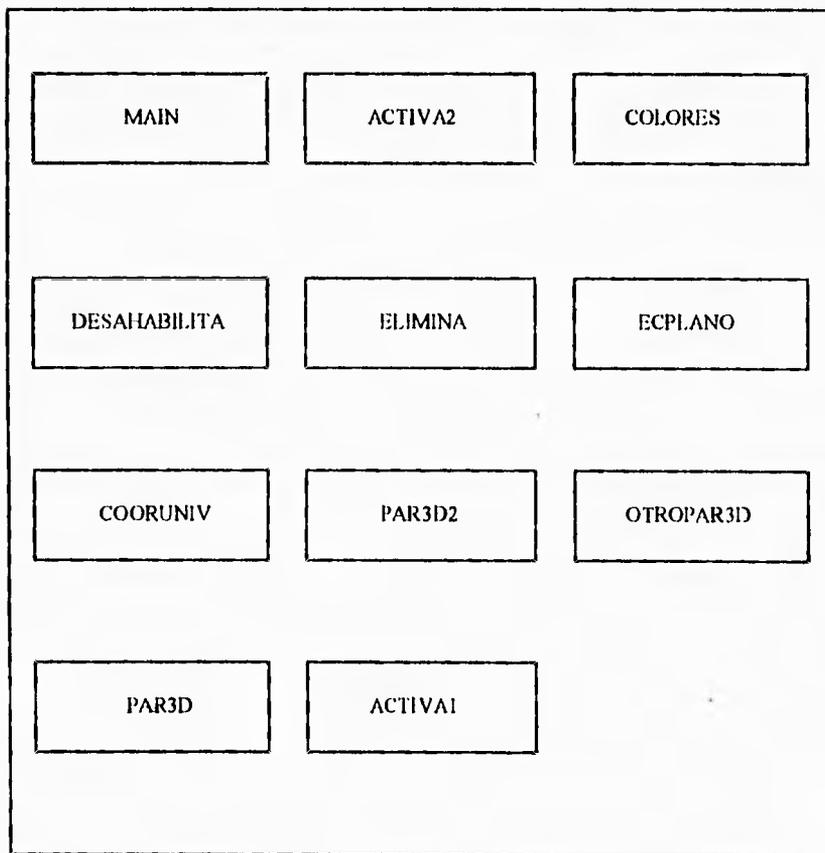


**Bóton que colorea la figura**

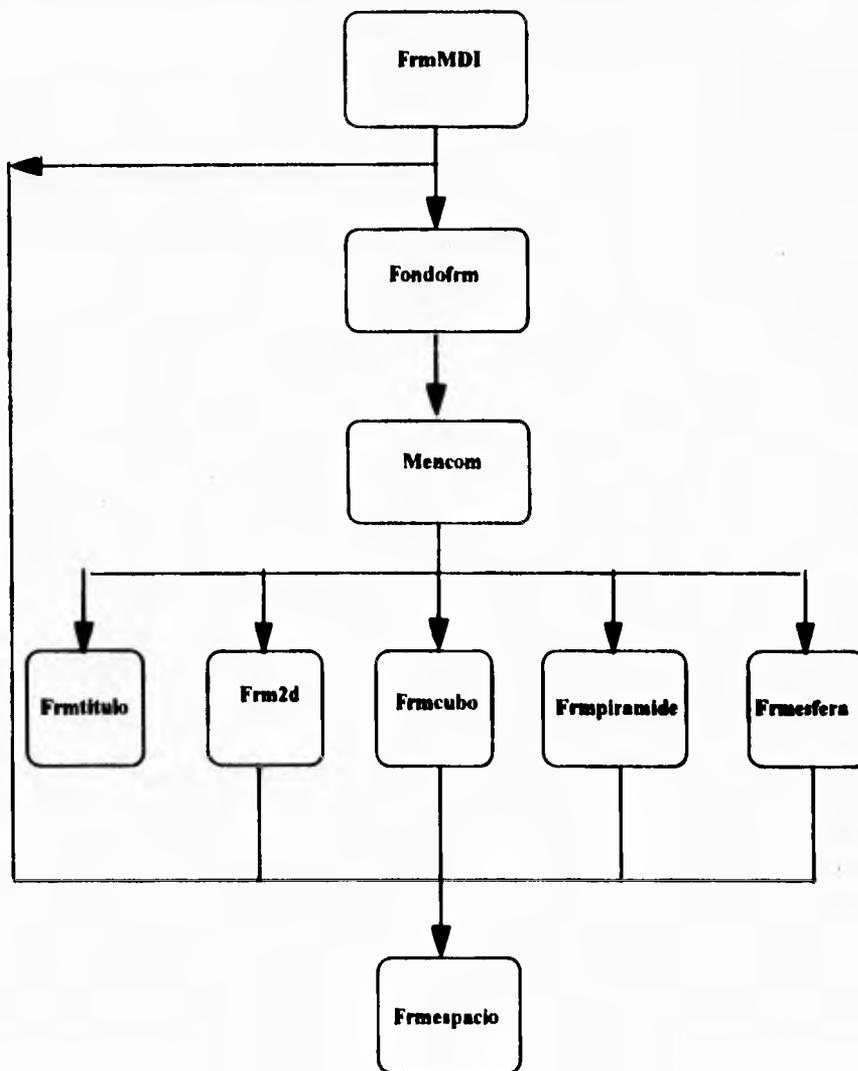


**Procedimiento**

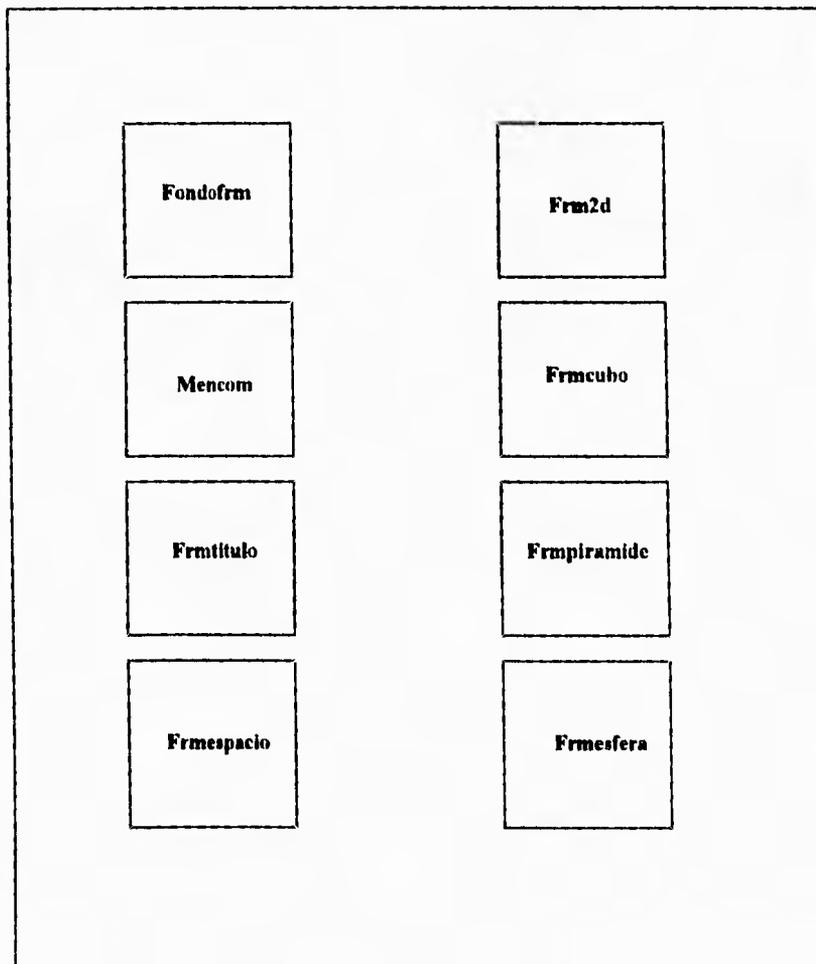
**PROCEDIMIENTOS PERTENECIENTES  
A GENERAL.BAS**



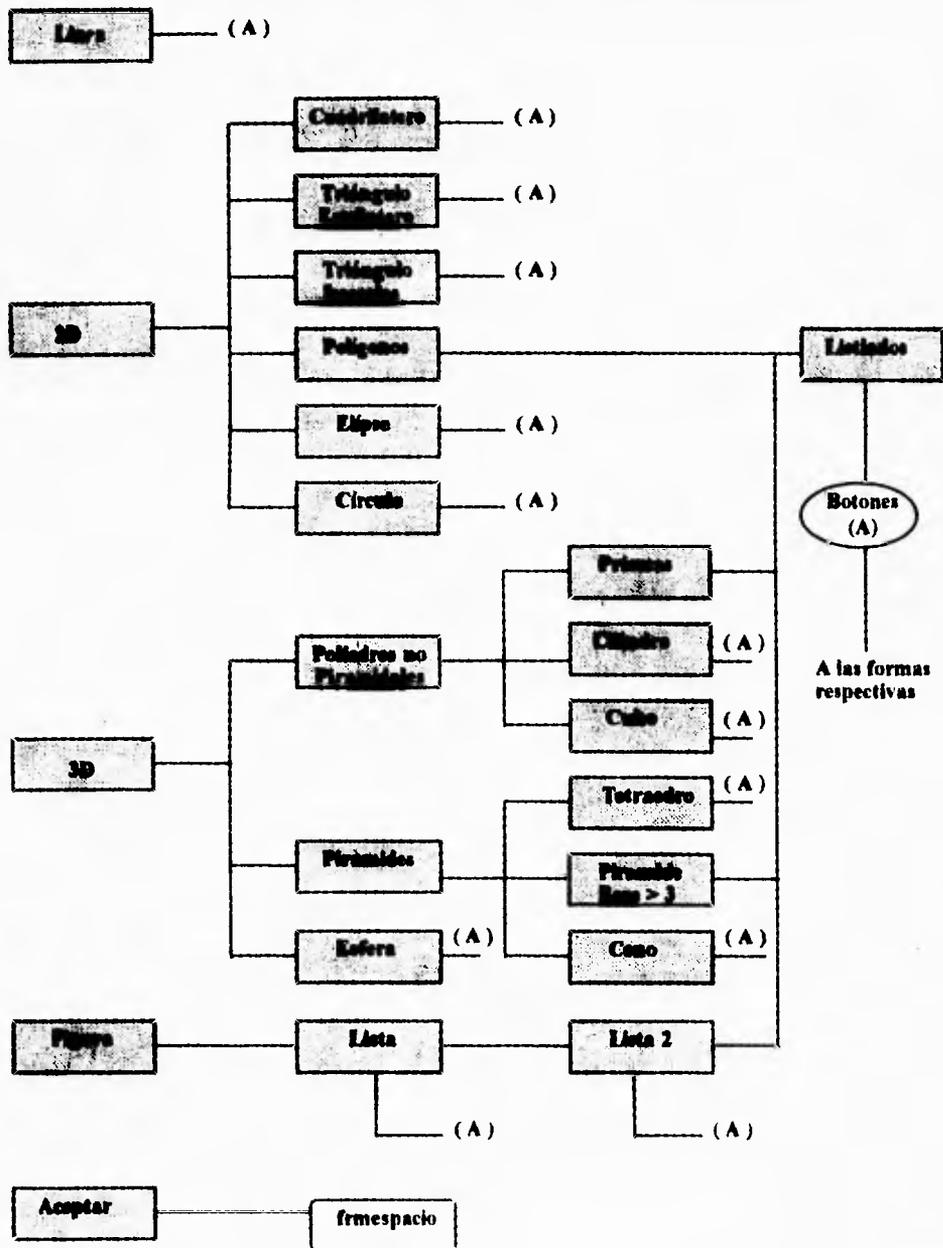
## DIAGRAMA DE FORMAS



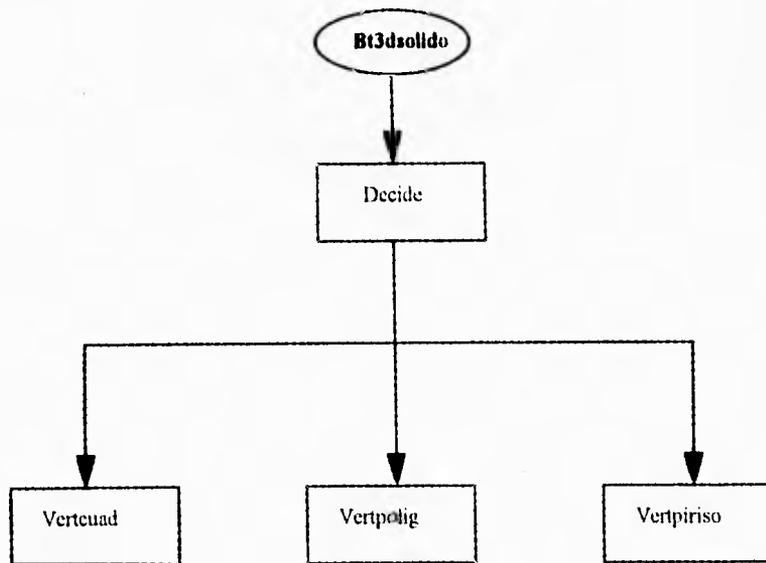
## DIAGRAMA DE FORMAS PERTENECIENTES A FRMMDI



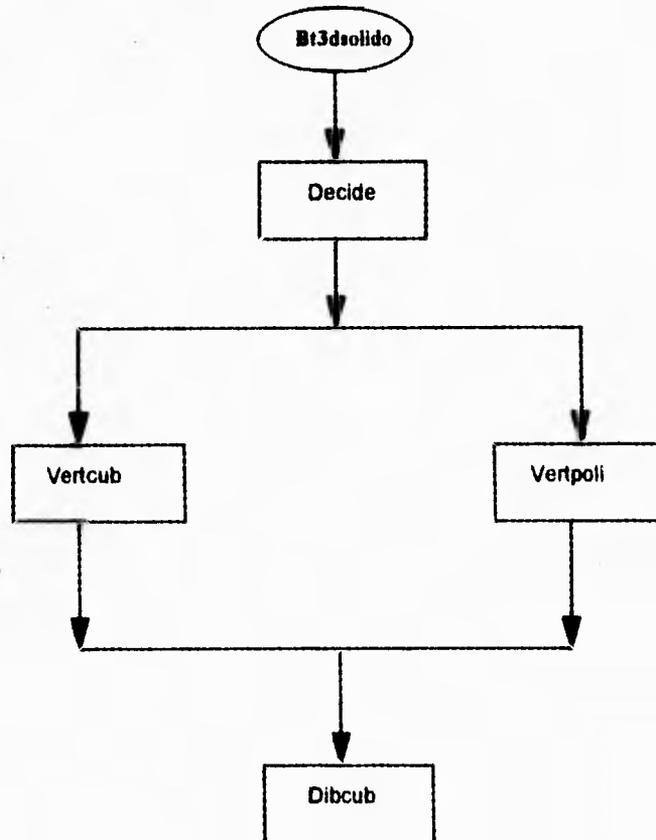
## DIAGRAMA DE MENCOM



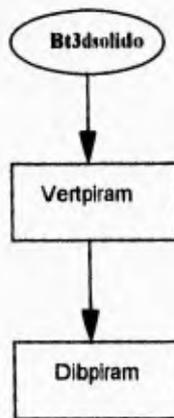
## DIAGRAMA DE FRM2D



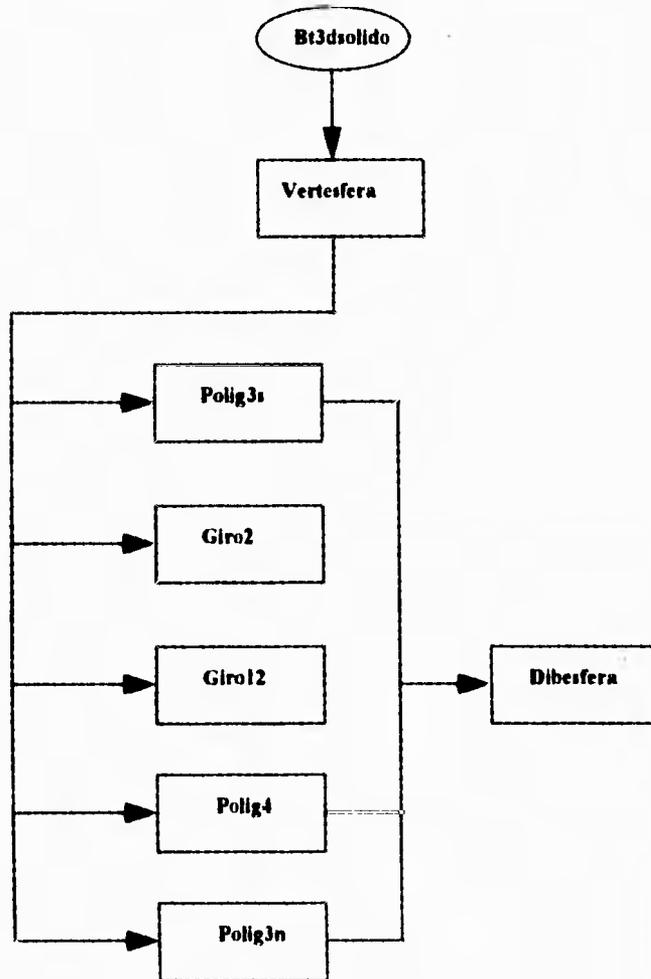
## DIAGRAMA DE FRMCUBO

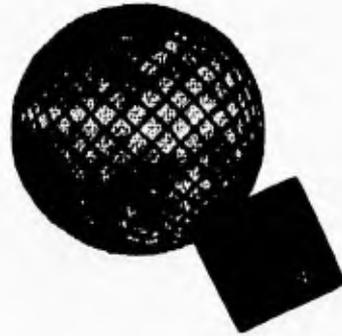


## DIAGRAMA DE FRMPIRAMIDE



## DIAGRAMA DE FRMESFERA





## CAPITULO 5

### CONCLUSIONES



**Espacio:**

Extensión indefinida  
de tres dimensiones  
que constituye el  
objeto de la geometría  
del espacio.

---

## CONCLUSIONES

**D**espués de la implementación del sistema se debe de llevar a cabo un análisis y ver si se cumplieron los objetivos que fueron planteados al principio de este trabajo.

El principal objetivo que se planteo fue el de crear un sistema que nos pudiera permitir la creación y manipulación de figuras geométricas en tercera dimensión en una PC, y pudiéndose obtener compatibilidad con algunos paquetes existentes en el mercado para poder ser utilizado, al mismo tiempo que fuera una herramienta útil para la elaboración de trabajos que requirieran de este tipo de esquemas, por lo que podemos concluir que fue cumplido y de manera satisfactoria el objetivo ya que este sistema maneja las principales figuras geométricas, pudiendo manipular sus características dimensionales y de posición en el espacio y genera archivos de tipo BMP lo cual nos permite una gran compatibilidad con la mayoría de los paquetes comerciales que se encuentran en el mercado.

Si se hace un análisis más profundo para ver como se comporta el sistema respecto a su funcionalidad, bajo costo de operación, modularidad y compatibilidad se puede apreciar que cumple con todas estas características de una manera optima, ya que se ha diseñado para que cualquier persona, aún contando con muy pocos conocimientos de computación pueda usarlo.

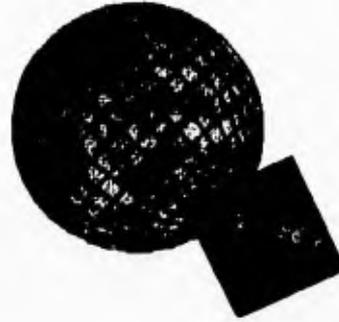
El sistema puede ser accesible a todo tipo de usuario. Es posible que estudiantes de nivel básico lo utilicen como una herramienta en el aprendizaje de la geometría; otro aspecto a considerar es la capacidad de funcionar en casi la mayoría de las PC requiriendo solamente de un Mouse y una impresora como equipo adicional a la propia computadora, además el sistema cumple con ser modular, ya que están bien definidos los procedimientos que se utilizan y si en un determinado momento se desea hacer un cambio o agregar nuevas figuras u opciones, es fácil de identificar el lugar o módulo dónde ésto debe de realizarse.

Podemos concluir que los resultados son satisfactorios pues se han cumplido con los objetivos primordiales que dieron origen a esta tesis.

También podemos concluir que haber utilizado Visual Basic para el desarrollo del sistema fue buena decisión, pues nos facilitó la creación del ambiente tipo Windows y pudimos aprovechar los recursos gráficos que brinda.

## **CONCLUSIONES**

De igual manera se pudieron aprovechar algunas utilerías, como aquellas que sirven para realizar discos de instalación y para generar la ayuda del sistema. Con ésto creamos un sistema con una mejor presentación y de buena calidad con ambiente Widows que llama más la atención a los usuarios de computadoras.



**APENDICE A**

**MANUAL DE OPERACION  
DEL USUARIO**



## **APENDICE A**

---

### **CONTENIDO**

I INTRODUCCION . . . . .	A-4
II REQUERIMIENTOS DEL EQUIPO DE COMPUTO PARA LA OPERACION DEL SISTEMA . . . . .	A-6
III PROCEDIMIENTO DE INSTALACION . . . . .	A-7
IV CONSIDERACIONES PARA EL FUNCIONAMIENTO DEL SISTEMA . . . . .	A-8
V COMO ACCESAR EL SISTEMA . . . . .	A-10
VI BARRA DE MENUS	
• Archivo . . . . .	A-11
• Editar . . . . .	A-11
• Opcion . . . . .	A-12
• Ayuda . . . . .	A-12

## MANUAL DE OPERACION DEL USUARIO

### 1. COMANDOS DEL MENU DE ARCHIVO

1.1. Nuevo . . . . .	A-13
1.2. Abrir . . . . .	A-14
1.3. Salvar . . . . .	A-15
1.4. Salvar Como . . . . .	A-15
1.5. Imprimir . . . . .	A-16
1.6. Salir . . . . .	A-16

### 2. COMANDOS DEL MENU DE EDITAR

2.1. Ver espacio . . . . .	A-16
2.2. Regresar . . . . .	A-17
2.3. Cambiar Fuente. . . . .	A-17

### 3. COMANDOS DEL MENU DE OPCION

3.1. Vista . . . . .	A-17
3.2. Menú por Iconos . . . . .	A-18
3.3. Menú por Lista . . . . .	A-18

### 4. COMANDOS DEL MENU DE AYUDA

4.1. Contenido . . . . .	A-19
--------------------------	------

### 5. MANEJO DEL SISTEMA

5.1. Diseño de la línea, el triángulo equilátero, el polígono, el círculo, la esfera y el tetraedro . . . . .	A-22
5.2. Diseño de el cuadrilátero, el triángulo isósceles, la elipse, el cilindro, el cono, el prisma y la pirámide base mayor a tres . . . . .	A-22
5.3. Diseño del cubo . . . . .	A-23
5.4. Posición en el espacio . . . . .	A-23

## **APENDICE A**

---

### **I INTRODUCCION**

En la actualidad la computadora es una herramienta indispensable para todos los sectores del país; la industria productiva la necesita para tener al día su información referente a su producción, a sus finanzas, y muchos otros factores indispensables para ella; en los centros de investigación también son un elemento fundamental que nos ha permitido el desarrollo en todas las áreas de las ciencias; también en los hogares son útiles como una herramienta de apoyo capaz de ayudar al aprendizaje de todos los miembros de la familia, sin importar cual sea su ocupación. En base a esto consideramos de gran importancia el desarrollo de este Sistema Editor de Figuras Geométricas en Tres Dimensiones para servir como un apoyo en el aprendizaje de la geometría sólida haciéndolo de una manera gráfica y divertida, sirviendo a la vez como una herramienta en el diseño gráfico asistido por computadora.

Con base en lo expuesto anteriormente se definen a continuación los propósitos del presente sistema:

- Modernizar los métodos tradicionales de enseñanza-aprendizaje en cuanto a lo que es la geometría sólida facilitando de esta manera algunos conceptos que necesitan ser visualizados para poder ser comprendidos y asimilados correctamente y no existe una mejor manera de aprender que cuando se está observando un ejemplo gráfico.
- Ser un apoyo para los diferentes programas educativos que realizan la función de enseñanza de figuras geométricas.
- Atender de manera oportuna a todos los que necesiten de un paquete de diseño gráfico por computadora que requieran la visualización de figuras geométricas en sus diseños

Este sistema se basó en la mayor cantidad de información sobre gráficos que se pudo obtener y con la ventaja de que corre bajo un ambiente que la mayoría de los usuarios tiene, como es el ambiente Windows, para obtener todas las ventajas que este ambiente nos permite.

Como resultado colateral nos permite obtener una opción que consideramos muy importante, que es el hecho de una vez obtenido nuestro diseño final podemos "girar" con respecto al diseño y no el diseño girar que es lo que usualmente se observa en otro tipo de diseñadores gráficos; esto se considera que es una ventaja importante pues nos sirve para tener diversas perspectivas del diseño final.

## **MANUAL DE OPERACION DEL USUARIO**

Además, el sistema genera un archivo de mapa de bits (BMP) que puede ser utilizado por cualquier paquete de Windows que exporte archivos de este tipo.

En este manual encontrará la información necesaria para el manejo y administración del Sistema Editor de Figuras Geométricas en Tercera Dimensión.

## **APENDICE A**

---

### **II REQUERIMIENTOS DEL EQUIPO DE COMPUTO PARA LA OPERACION DEL SISTEMA**

El equipo en que se instale el sistema debera tener las siguientes características para su adecuada operación:

#### **CONFIGURACION MINIMA:**

Computadora:	Procesador 80386 de 33 MHZ o superior.
Memoria:	4 MB de Memoria RAM
Disco Duro:	1 MB de Capacidad disponible
Monitor:	VGA de Color o monocromático
Unidad de discos:	Una unidad de lectura de discos de doble densidad. Si es de 3 1/2" con capacidad para 720 KB
Impresora:	Impresora de Matriz de Puntos de 300 c.p.s., de carro de 15" que sea capaz de emular IBM Proprinter.
Plataforma:	MS-DOS versión 5.0 o posterior. Windows 3.1.
Mouse:	Mouse Serial compatible con el equipo.

#### **CONFIGURACION RECOMENDABLE:**

Computadora:	Procesador 80486 de 66 MHZ o superior.
Memoria:	8 MB de Memoria RAM o superior
Disco Duro:	1.5 MB de Capacidad disponible
Monitor:	Super VGA de Color
Unidad de discos:	Una unidad de lectura de discos de alta densidad. Si es de 3 1/2" con capacidad para 1.44 MB o si es de 5 1/4" con capacidad para 1.2 MB.
Impresora:	Impresora de Matriz de Puntos de 300 c.p.s. capaz de emular IBM Proprinter e Impresora Láser de 8 PPM capaz de emular HP Láser Jet 4L.
Plataforma:	MS-DOS versión 6.20 o posterior. Windows 3.1
Mouse:	Mouse Serial compatible con el equipo.

## MANUAL DE OPERACION DEL USUARIO

### III PROCEDIMIENTO DE INSTALACION

Al recibir el discos para la instalación del sistema en su equipo de cómputo debe realizar el siguiente proceso:

- a. Inserte el disco en la unidad de lectura de discos correspondiente.
- b. En la línea de comandos del D.O.S. escriba WIN y pulse la tecla ENTER.
- c. Una vez dentro de WINDOWS deberá seleccionar el comando de Archivo, y seleccionar el comando de Ejecutar.
- d. Cuando esté en la ventana correspondiente al comando Ejecutar, escriba A:\INSTALAR ó B:\INSTALAR de acuerdo a la unidad de lectura de disco en donde se encuentre el disco del sistema y presione ENTER.
- e. El programa de instalación verificará los recursos de su equipo, así como el espacio necesario en disco duro para iniciar la instalación.
- f. Se le preguntará la unidad del disco duro y el subdirectorío destino en el cual desee instalar el sistema; de forma automática se presentará la siguiente ruta destino C:>\EDITOR3D, si acepta el directorío propuesto pulse ENTER, de lo contrario escriba el de su preferencia.

En caso de existir anteriormente un directorío con el mismo nombre se le informara de su existencia, permitiéndole seleccionar otro directorío para iniciar la instalación. Si no selecciona directorío alguno, se borrara la información de éste para continuar con el proceso. Una vez seleccionados la unidad y el directorío destino se procedera a realizar la instalación de archivos ejecutables y sus librerías necesarias.

- g. Para acceder el sistema consulte el punto V de este manual.

## **APENDICE A**

---

### **IV CONSIDERACIONES PARA EL FUNCIONAMIENTO DEL SISTEMA**

Pasos elementales para operar:

- A. Para la correcta operación del sistema se deberán tener unos cuantos conceptos básicos de geometría sólida, para saber con claridad a que se refieren algunos términos que podrían ser utilizados a lo largo del sistema, y ocasionalmente podrían parecer ambiguos, los conocimientos necesarios son realmente mínimos pues este sistema se ha diseñado como una ayuda en el aprendizaje de geometría sólida.
- B. Si el sistema se va a utilizar como una herramienta gráfica para generar figuras que posteriormente se utilizarán en algún procesador de texto, el conocimiento de la manipulación e inserción de imágenes en el texto queda en el entendido que el usuario conoce todo lo necesario para su buen funcionamiento.
- C. El sistema corre bajo un ambiente completamente WINDOWS por lo que se necesita tener una cierta familiarización con este ambiente, lo cual es realmente fácil.
- D. El sistema genera dos archivos uno que es el BMP en donde se encuentran las imágenes hechas, pero también se tiene un archivo con extensión DAT que contiene los datos referentes a las imágenes es muy importante que se encuentren los dos archivos cuando deseamos abrir un archivo previamente elaborado pues si no el sistema dará por hecho que no existe el archivo.
- E. Es importante recalcar que el orden de las rotaciones siempre es el mismo, la primer rotación que se realiza es la rotación con respecto al eje Y, la siguiente rotación es con respecto al eje Z y la última rotación es con respecto al eje X.

Por ejemplo :

- 1o. Se realiza una rotación de  $90^\circ$  con respecto al eje Y.
- 2o. Se rota  $90^\circ$  con respecto al eje X.
- 3o. Se rota ahora  $90^\circ$  con respecto al eje Z.

## MANUAL DE OPERACION DEL USUARIO

Se esperaba que el sistema al realizar la última rotación respetara el orden dado (Fig. A-1), pero al incluir la rotación en Z, éste realizará otro movimiento (Fig. A-2).

De esta manera parecería que la última rotación se realizó con respecto al eje Y, pero no fué así, simplemente se realizaron las rotaciones en el siguiente orden:

- 1o. Rotó  $90^\circ$  con respecto al eje Y.
- 2o. Rotó  $90^\circ$  con respecto al eje Z.
- 4o. Rotó  $90^\circ$  con respecto al eje X.

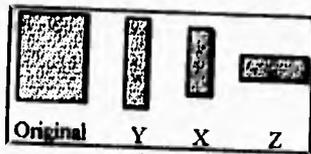


Fig. A-1

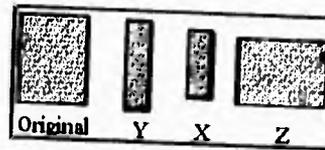


Fig. A-2

## **APENDICE A**

---

### **V COMO ACCESAR EL SISTEMA**

Una vez que el sistema haya sido instalado en el equipo de cómputo, ejecute las siguientes acciones:

- A. Una vez dentro de Windows, deberá seleccionar el grupo dentro del cual se encuentra el icono que activara al Sistema.
- B. Para acceder el sistema deberá hacer un doble clic en el icono del Sistema Editor de Figuras Geométricas en tres dimensiones (Fig. A-3)



Fig. A-3

- C. Una vez dentro del Sistema se mostrara la pantalla de presentación y también se mostrará la barra de menús de comandos. (Fig.A-4)

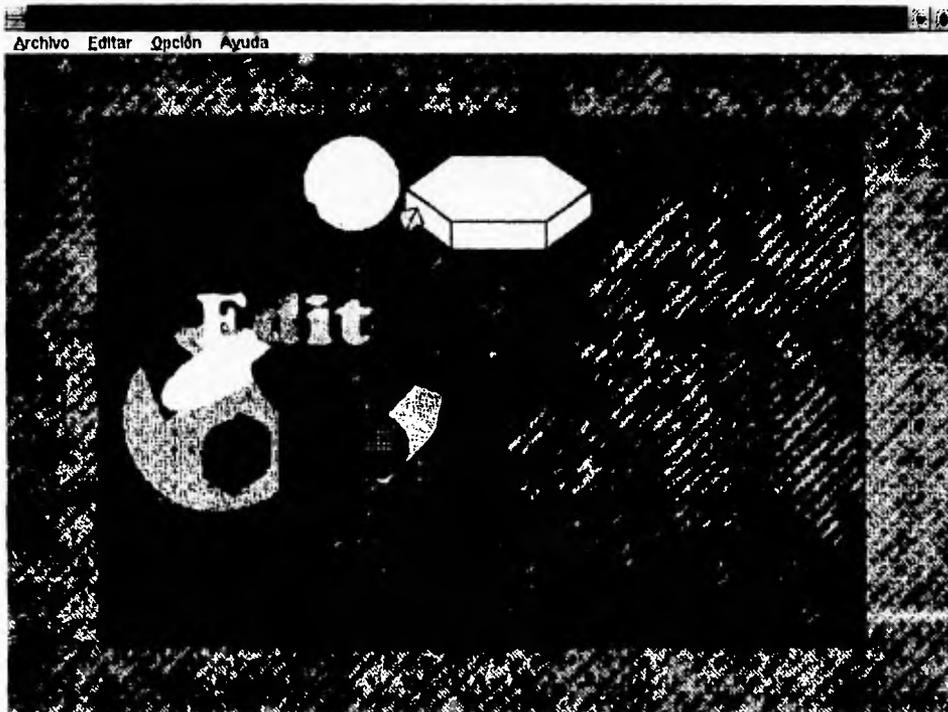


Fig A-4

## MANUAL DE OPERACION DEL USUARIO

### **VI BARRA DE MENUS**

La barra de menús contiene las siguientes opciones:

#### **ARCHIVO**

Para abrir este menú se deben presionar simultáneamente las teclas [Alt]-[A] o seleccionar con el ratón el comando de Archivo. En este menú se encuentran los comandos relacionados con la manipulación de los archivos, éstos son:

- **Nuevo:** Este comando es el principal de nuestro sistema pues con el se inicia el proceso de creación de las figuras geométricas, aparecerá un menú para seleccionar la figura que se va a realizar. Se accesa a esta opción seleccionandola o presionando la tecla [N] cuando está abierto el menú Archivo.
- **Abrir:** Este comando es de gran utilidad, pues nos sirve para abrir archivos que anteriormente fueron creado, por si es necesario realizarles un cambio o simplemente visualizarlos. A este comando se puede accesar presionando la tecla [B].
- **Salvar:** Esta opción permite guardar un diseño en un archivo y la tecla para este comando es la [S].
- **Salvar Como:** Con este comando se permite guardar el diseño actual en una archivo distinto al que se tenía o en otra unidad de disco. Esta opción se activa con la tecla [C].
- **Imprimir:** Uno de los objetos primordiales del sistema es obtener una copia impresa de lo realizado y este comando da la oportunidad de realizarlo. Para habilitar este comando se utiliza la tecla [I].
- **Salir:** Este comando nos permite salir del sistema y regresar al ambiente WINDOWS. Este comando funciona con la tecla [L] cuando está abierto el menú ARCHIVO, y si no lo está se puede accesar a él oprimiendo simultaneamente las teclas [Ctrl]-[L].

#### **EDITAR**

Para abrir este menú deben presionarse simultáneamente las teclas [Alt]-[E] o seleccionar con el ratón el comando de Editar.

En esta opción se encuentran los siguientes comandos:

## **APENDICE A**

---

- ★ **Ver espacio:** Esta opción tiene como finalidad fundamental la de permitir ver las figuras que se encuentran ya posicionadas en el espacio cuando se está generando una nueva figura. Se accesa a este comando presionando la tecla [V] si está abierto el menú Editar.
- ★ **Regresar:** Una vez que se vio el estado actual del espacio, con esta opción se puede regresar a la ventana de diseño de la figura y su tecla es la [R].
- ★ **Cambiar Fuente:** Esta opción permite cambiar el tipo, tamaño y color de la letra cuando se desea escribir texto en el diseño. A este comando se accesa con la tecla [F].

### **OPCION**

Para abrir este menú deben presionarse simultáneamente las teclas [Alt]-[O] o seleccionar con el ratón el comando Opcion.

En esta opción se encuentran los comandos relacionados con las opciones que presenta el sistema en su funcionamiento para hacerlo flexible:

- ★ **Vista:** Esta opción sólo estará activada cuando se encuentran todas las figuras posicionadas dentro del espacio, y se desea obtener una vista específica con una determinada cantidad de grados de perspectiva. Para acceder a este comando debe presionarse la tecla [V] cuando está abierto el menú Opción.
- ★ **Menú por Iconos:** En el sistema existe la opción de trabajar con dos tipos de menús, este comando permite seleccionar el modo de iconos para trabajar a lo largo del sistema. Este comando funciona con la tecla [I].
- ★ **Menú por Lista:** Este comando permite seleccionar el otro tipo de menú que funciona con una serie de listas donde se encuentran las opciones de figuras a realizar. Este comando funciona con la tecla [L].

### **AYUDA**

Para abrir este menú deben presionarse simultáneamente las teclas [Alt]-[Y] o seleccionar con el ratón el comando de aYUDA.

En esta opción se encuentra El siguiente comando:

- ★ **Contenido:** En esta opción se encuentra la ayuda general del sistema. Para acceder a esta opción debe presionarse la tecla [C] cuando el menú Ayuda está abierto o presionando la tecla [F1] aunque esté cerrado.

## **1.- COMANDOS DEL MENÚ DE ARCHIVO**

Los comando de este menú son (Fig. A-5):

### **1.1.- Nuevo**

Este comando nos sirve para la creación de las figuras geométricas que el sistema es capaz de realizar, ésta es la parte principal del sistema pues en esta opción se empezaran a generar las figuras que se desean incluir en algún diseño. Al seleccionar este comando se desplegará un menú gráfico (Menú por Iconos), el cual puede ser cambiado por un menú tipo texto (Menú por Lista) dependiendo de la opción de tipo de menú que se haya seleccionado (para mayor claridad de estos temas se abordaran ampliamente en los apartados 3.2 y 3.3 de este manual.), para tener un mejor entendimiento del funcionamiento de ambos menús se explicara primero el menú por Iconos y posteriormente el menú por Lista.

### **MENU POR ICONOS**

Aquí aparecerá el menú de iconos (Fig. A-6) que nos indicara que tipo de figura deseamos realizar: una línea, una figura de dos dimensiones o una de tres.

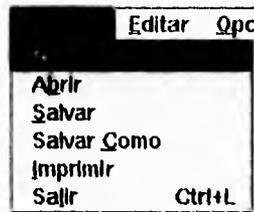


Fig. A-5



Fig. A-6

Una vez seleccionado el tipo de figura que se desea realizar, si se trata de una figura de dos dimensiones se presentará una serie de iconos (Fig. A-7) entre los que se deberá indicar que figura se hará y que en los proximos apartados se explicará la forma en que se realizan dichas figuras. Cuando se desea realizar un polígono se deberá de indicar el número de lados que se desea tenga el polígono, seleccionando el número de la lista que aparece para tal caso (Fig. A-8). Si el tipo de figura que fue seleccionado es de tres dimensiones se presenta otra serie de iconos (Fig. A-9) dentro de los que se deberá de indicar la figura a hacer. Cuando se selecciona el icono de Poliedro no piramidal, saldrá otro conjunto de iconos (Fig. A-10) para indicar el tipo de poliedro a

## APENDICE A

realizar; si se selecciona el icono de Prismas se debera indicar el número de lados que se desea tenga la base del prisma de igual manera que con los polígonos. Si se selecciona el icono de Pirámides saldrá otro conjunto de iconos (Fig. A-11) que pedira indicar que tipo de pirámide se va a realizar y si selecciona el icono de base mayor a 3 se deberá de indicar cuantos lados tendrá la base de dicha pirámide, de igual forma que con los prismas.



fig. A-7

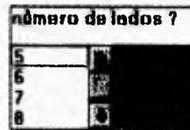


fig. A-8



Fig. A-9



Fig. A-10



Fig. A-11

### MENU POR LISTA

Aquí aparecerá el menú que indicará que tipo de figura se desea realizar: una línea, una figura de dos dimensiones o una de tres. Si se trata de una figura de dos dimensiones se presentará un submenú (Fig. A-12) en el que se deberá indicar que figura se hará. En los próximos apartados se explica más detalladamente la forma en que se realizan dichas figuras. Cuando se desea realizar un polígono se deberá de indicar el número de lados que se desea tenga el polígono. Si el tipo de figura que fue seleccionado es de tres dimensiones se presenta un submenú en el que deberá de indicar la figura a realizar, (Figs. A-13 y A-14), cuando se selecciona el submenú de Poliedros no piramidales, saldrá otro menú para indicar el tipo de poliedro a realizar (Fig. A-13), si selecciona la opción de prismas se deberá indicar cuantos lados tendrá la base del prismas; si se selecciona el submenú de Pirámides saldrá otro menú (Fig. A-14) para indicar que tipo de pirámide se va a realizar y si selecciona la opción de base mayor a 3 se deberá de indicar cuantos lados tendrá la base de dicha pirámide.

Una vez seleccionado el tipo de figura a realizar se pasara a la etapa de diseño de la figura que se explicara detalladamente en los siguientes apartados.

#### 1.2.- Abrir

Este comando se utiliza para abrir un archivo que ya fue creado con anterioridad y que se desea visualizar y/o modificar en este momento. Al

## MANUAL DE OPERACION DEL USUARIO

seleccionar este comando saldrá una ventana en la que se le solicitará que se señale el nombre del archivo a abrir. Después de esto aparecerá el diseño que acaba de abrir, esto se hace con la finalidad de que se vea el estado actual del diseño y de ahí surjan las modificaciones a realizar. Para poder pasar al menú principal de figuras basta con seleccionar el botón "otra figura" o se puede empezar a rotar la perspectiva del diseño ya sea para imprimir o grabar la vista (posteriormente se verá con mayor detalle este comando).

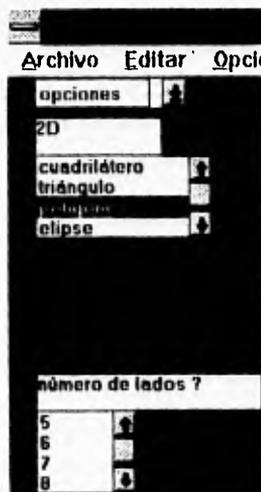


Fig. A-12

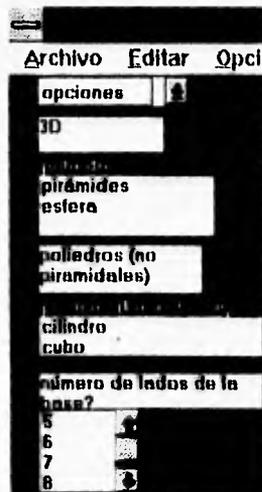


Fig. A-13

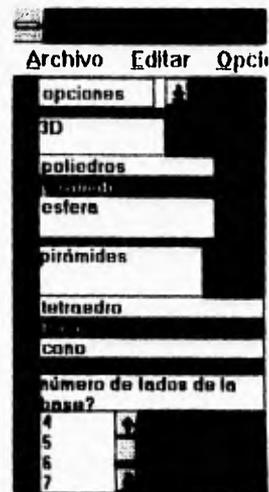


Fig. A-14

### 1.3.- Salvar

El comando de salvar sirve para guardar en un archivo específico el diseño que se realizó. Cuando sea la primera vez que se desee guardar el archivo, (que por default trae el nombre de "noname.bmp") se presentará una ventana para preguntar el nombre que se le quiere dar al archivo. Si este archivo ya tiene nombre, ya sea, porque se salvo anteriormente o bien, porque era un archivo ya existente, entonces ya no preguntará el nombre, simplemente salvará el archivo con el nombre que se tenía anteriormente. Cuando es la primera vez que se ejecuta este comando en una sesión se mostrará un mensaje avisando que el diseño se salvará con perspectiva de  $0^\circ$  y sin texto alguno (si es que existe), además de su correspondiente archivo de datos (Fig. A-15).

### 1.4.- Salvar Como

Este comando tiene una función muy similar al anterior, de respaldar toda la información que se desee, pero éste siempre preguntará el nombre con el que

## **APENDICE A**

---

se desea guardar la información, es muy útil en las situaciones en las que se quieren guardar los cambios hechos a un archivo pero sin perder el archivo original, de entrada mostrará el nombre que tiene actualmente pero se le puede cambiar simplemente con teclear el nuevo nombre. Cuando es la primera vez que se ejecuta este comando en una sesión se mostrará un mensaje avisando que el diseño se salvara con perspectiva de 0° y sin texto alguno (si es que existe), además de su correspondiente archivo de datos (Fig. A-15).

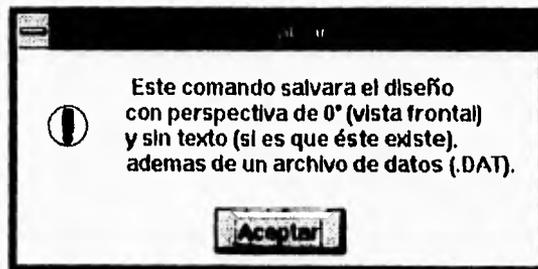


Fig. A-15

### **1.5.- Imprimir**

Uno de los requerimientos que se le solicita a todos los sistemas es el hecho de poder obtener una copia en papel del trabajo final y este comando es lo que va a permitir realizar. Para poder ejecutar este comando es necesario abrir el archivo deseado previamente y una vez que se está visualizando en el espacio lo podemos imprimir en ese mismo momento.

### **1.6.- Salir**

Este comando sirve para terminar la operación del Sistema Editor de Figuras Geométricas y regresar a Windows.

## **2.- COMANDOS DEL MENU DE EDITAR**

Los comando de este menú son (Fig. A-16):

### **2.1.- Ver espacio**

Una vez que se ha creado una figura geométrica y se coloca en el espacio, tal vez sea necesario volver a él para ver en que posición quedo la figura y en base a ésta hacer el diseño de las siguientes. Pues bien este comando es

## MANUAL DE OPERACION DEL USUARIO

justamente lo que permite hacer. Cuando se está en la ventana de diseño de una figura se puede ir a hechar un "vistazo" para ver como va el diseño en el espacio.

### **2.2.- Regresar**

Este comando se podría decir que es el complemento del anterior pues una vez que se ha visto el espacio y se ha analizado la posición que debe tener la siguiente figura es necesario regresar para continuar o bien empezar el diseño de una figura, y es este comando el que da esa oportunidad.

### **2.3.- Cambiar Fuente**

Este comando se encuentra deshabilitado hasta el momento en que se habilita la opción "vista" del menu "Opción". Con él se puede cambiar el tipo de letra y sus características (color, tamaño, tipo, etc.) con la que se escribirá el texto en el diseño.

## **3.- COMANDOS DEL MENU DE OPCION**

Los comando de este menú son (Fig. A-17):

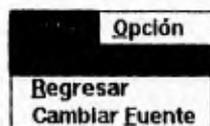


fig. A-16

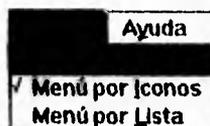


fig. A-17

### **3.1.- Vista**

Como ya se mencionó, el sistema cuenta con la opción de simular la rotación del usuario sobre los objetos, y justamente este comando es el encargado de hacer una parte de este trabajo, en la ventana del espacio se encuentra el control para hacer que la perspectiva del usuario empiece a cambiar, y cuando se tiene la vista deseada con este comando se inicia el proceso de coloración de las figuras; al elegir esta opción aparecerán cuatro botones (Fig. A-18) en la ventana del espacio. El primero de los botones tiene un disquete, y va a permitir guardar la vista del diseño en un archivo, pero este archivo no podra ser mandado llamar por el editor ya que sólo se salvara el archivo de imagen y no se salvara el archivo de datos; el segundo botón es el que tiene una brocha y es el que sirve para colorear las figuras, conservando la

## APENDICE A

---

inclinación que se desea; el tercer botón es el que tiene una impresora, sirve para obtener una impresión de lo que en ese momento se está viendo en la pantalla; y el último botón es el que tiene una letra **B** y permite escribir texto, aquí aparece una ventana pidiendo que se introduzca el texto deseado (Fig. A-19), una vez aceptado, el texto se imprimira en el diseño donde se haga clic con el ratón. Es importante mencionar que esta opción está activa únicamente cuando todas las figuras ya están colocadas en su posición final y que si existieran figuras pendientes de acomodar se encuentra deshabilitada.



Fig. A-18

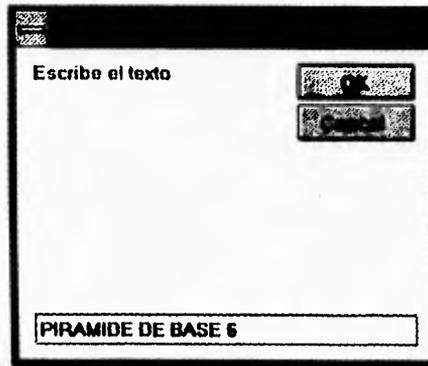


Fig. A-19

### 3.2.- Menú por Iconos

Para hacer más amigable el sistema se decidió tener la opción de decidir por parte del usuario el modo en el que quiera trabajar, con esta opción se activaran todos los iconos necesarios para poder hacer la selección de las figuras a editar por medio de iconos. Esta opción únicamente estará activada cuando se está en la selección de la figura a realizar posteriormente se desactivara. Por default este es el menú que está activado al inicio del sistema.

### 3.3.- Menú por Lista

Con esta opción se activa el menú por Lista la selección de la figura, aparecerán una lista y de ahí se va seleccionando la opción deseada.

## MANUAL DE OPERACION DEL USUARIO

### 4.- COMANDOS DEL MENU DE AYUDA

#### 4.1.- Contenido

Al seleccionar esta opción se muestra la ayuda del Sistema Editor de Figuras Geométricas en Tercera Dimensión (Fig. A-20).

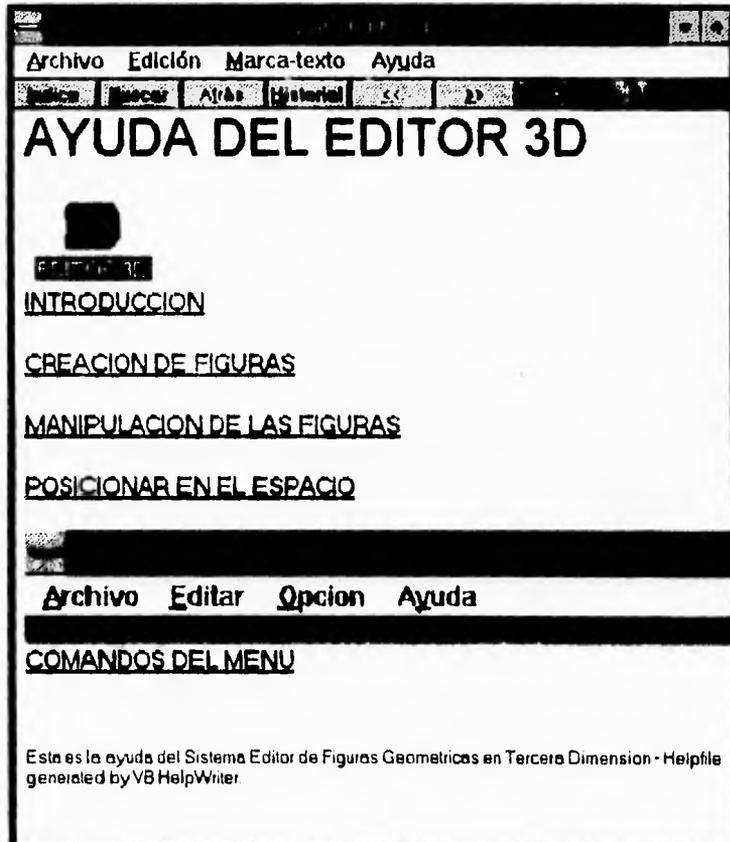


Fig. A-20

### 5.- MANEJO DEL SISTEMA

Una vez que se ha seleccionado en el menú que figura se va a realizar se presentará una ventana en la que se podrá empezar el diseño (Fig. A-21), en esta ventana aparecerán varios botones que servirán para poder manipular libremente la figura.

## APENDICE A

A continuación se explican los botones que son comunes para todas las figuras y que tienen la misma función, más adelante se explicaran los controles del tamaño de la figura para cada caso.

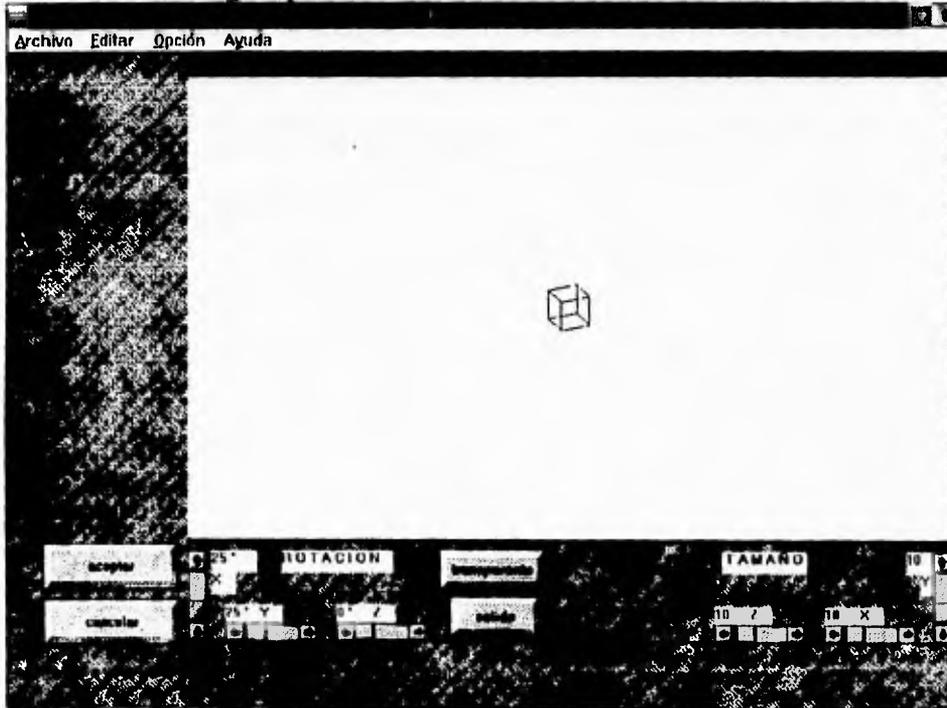


Fig. A-21

❶ **TRANSPARENTE:** Este botón sirve para volver la figura a la estructura alámbrica y se vuelve transparente a la vista, quedando dibujado únicamente el contorno de la figura y pudiéndose observar que es lo que hay atrás de ella (si es que hubiera algo.).

**NOTA:** Este botón no está disponible cuando se está diseñando una línea, ya que ésta no puede ser transparente.

❷ **SOLIDO:** Este botón vuelve a una figura 3D sólida y pintará del color que se le indique en la ventana que se muestra para este fin (Fig. A-22); posteriormente aparecerá un mensaje preguntando si se desea también cambiar el contorno de las caras de la figura (Fig.A-23), si se contesta afirmativamente aparecerá nuevamente la ventana de colores para seleccionar el que se desee, en caso contrario quedara el color del borde que se tenía antes de ejecutar este comando.

## MANUAL DE OPERACION DEL USUARIO



Fig. A-22

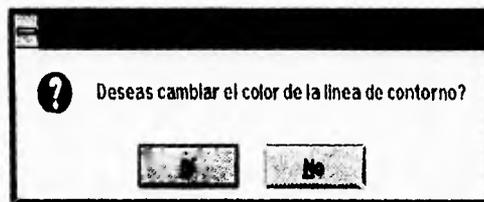


Fig. A-23

- **COLOR:** Este botón, aunque no se muestra en la fig. A-21, existe cuando se diseña cualquier figura de dos dimensiones o una línea. Este botón ocupa el lugar del botón sólido y funciona de igual manera.
- **ROTACION EN X:** Con esta opción se logra que la figura gire con respecto al eje X, se observa una barra desplazadora con la que se puede controlar el giro ya sea para adelante o para atrás. Se puede lograr que la figura dé una vuelta completa.
- **ROTACION EN Y:** Con esta opción se lograra un giro de la figura sobre el eje Y observándose un movimiento similar al que se observa al dar vuelta a la hoja de un libro, para ello se utiliza la barra desplazadora que viene para este efecto. Se puede lograr una vuelta de 360 grados.

A pesar de que se esté diseñando un cuerpo de dos dimensiones es posible hacer que éste gire sobre los ejes X y Y.

- **ROTACION EN Z:** Con esta opción se observará que la figura se mueve a la derecha o a la izquierda (rotación respecto al eje Z) dependiendo de los grados que se desee darle, lo cual se logra con la barra desplazadora que aparece para tal efecto, se puede lograr un giro de 360 grados para la figura.

Es posible realizar una serie de combinaciones con los tres tipos de rotaciones para obtener la vista deseada, pero siempre considerando lo mencionado en el inciso E del capítulo IV "Consideraciones para el funcionamiento del sistema".

## **APENDICE A**

- ⊕ **ACEPTAR:** Una vez que se a hecho el diseño final de la figura debe presionarse este botón para pasar a colocarlo en el espacio donde es posible combinarlo con otras figuras que se han hecho anteriormente, lo referente al espacio se explicar en un apartado más adelante dentro del mismo manual.
- ⊕ **CANCELAR:** Cuando se ha hecho una figura y después no se desea pasar al espacio, más bien se desea realizar otra figura, este botón va a permitir realizar esa acción con sólo presionarlo.

Estos son los comandos que son iguales para todas las figuras geométricas que se realizan con el sistema, pero en lo referente al tamaño de las figuras existen algunas diferencias en el manejo de estas dimensiones por lo que se agruparon en tres grupos dependiendo de la manera como se manipularan los controles.

### **5.1.- Diseño de la línea, el triángulo equilátero, los polígonos, el círculo, la esfera y el tetraedro.**

En este primer grupo todas las figuras tienen un sólo control para manipular su tamaño en X. Su uso es similar al de los anteriores; se cuenta con una barra desplazadora que lograra que aumente o disminuya el tamaño de la figura.

En la línea este control sirve para variar la longitud de ella; en el triángulo equilátero y los poigonos se podra variar la longitud de sus lados, que como se sabe son iguales; para el círculo y la esfera servirá para variar su radio; y finalmente en el tetraedro se podrá variar el tamaño de los lados de los triángulos que forman este cuerpo. El tamaño mínimo para las figuras de dos dimensiones es de cero unidades para que se pueda hacer un punto con ellas (si se desea), mientras que para las de tres dimensiones es de cinco unidades.

### **5.2.- Diseño de el cuadrilátero, el triángulo isósceles, la elipse, el cilindro, el cono, el prisma y la pirámide base mayor a tres.**

Estas figuras se caracterizan dentro del sistema en tener dos controles para manipular su tamaño. En el cuadrilátero, el triángulo isósceles y la elipse se controla su altura y su ancho, mientras que en el cilindro, y el cono lo que se controla es el radio de su base y la altura del cuerpo; finalmente en los prismas y en las pirámides se puede manipular el tamaño de la base a partir del radio que se mide del centro de la base a uno de los vertices que pertenecen a la base poligonal y la altura.

### **5.3.- Diseño del cubo**

Finalmente se tiene el diseño del cubo que difiere a los anteriores en que tienen tres controles con lo que se puede variar lo ancho del cuerpo (tamaño en X), la altura del mismo (tamaño en Y) y finalmente la profundidad del cuerpo (tamaño en Z) que se realiza con una barra desplazadora. El funcionamiento de los controles es el mismo que para los anteriores.

- **TAMAÑO EN X:** Esta opción sirve para variar el ancho de la figuras, se pueden apreciar una barra desplazadora que aumenta y disminuye las unidades.
- **TAMAÑO EN Y:** Esta opción sirve para variar la altura de la figura, se puede apreciar una barra desplazadora que indica que aumentan o disminuyen las unidades de altura.
- **TAMAÑO EN Z:** Esta opción sirve para variar la profundidad del cubo, se puede apreciar una barra desplazadora que indica que aumentan o disminuyen las dimensiones de profundidad del cubo.

### **5.4. Posición en el espacio**

Una vez que se acabó el diseño de la figura deseada es necesario pasar al espacio para colocarla junto con otras figuras, o para poder darle diversas vistas, para ello se presiona el botón aceptar y se presentará una pantalla como la que se muestra en la Fig. A-24; dentro de ella se podrá mover la figura, considerando que lo que se está observando en ese momento es el plano XY.

Para poder mover la figura a través del espacio y colocarla en la posición deseada se utilizara el desplazador de figuras (Fig. A-25). Para realizar un cambio de posición únicamente es necesita seleccionar el desplazador de figuras con el botón izquierdo del ratón y sin soltarlo arrastrarlo hasta el lugar deseado y al soltar el botón del ratón se verá que aparece el contorno de la figura en línea punteada. Una vez seleccionada la posición de la figura en el plano XY podemos pasar a acomodarla en el plano XZ, para ello se presionara la barra desplazadora que tiene el titulo de "perspectiva" (Fig. A-26), se podrá observar que la vista empieza a ir girando poco a poco, en el momento en que llegue a los 90° de rotación, aparecerá nuevamente el desplazador de figuras para permitir mover la figura en el plano -ZX; para comprender mejor, este movimiento nos dirá quién está más al fondo en el diseño final; una vez que se colocóla figura en la posición deseada, para aceptar la posición final de ésta debe presionarse el botón "aceptar xyz" (Fig. A-27), lo que hará que se de por

## **APENDICE A**

aceptada la posición de la figura, en este momento se vera la perspectiva del plano XY con la ultima figura pero ya como se vera finalmente.

Existe otro botón que es el denominado "otra figura" (Fig. A-28), este botón permite regresar al menú para seleccionar la siguiente figura que deseamos realizar, si este botón es oprimido sin haber dado "Aceptar XYZ" la última figura hecha se perdera (pues no fue aceptada).

Si al estar realizando el diseño deseamos modificar la figura esto es posible seleccionando en el menú superior la opción "Regresar " y que ya fue explicada anteriormente en el punto 2.2. siempre y cuando no se haya oprimido el botón "Aceptar XYZ".

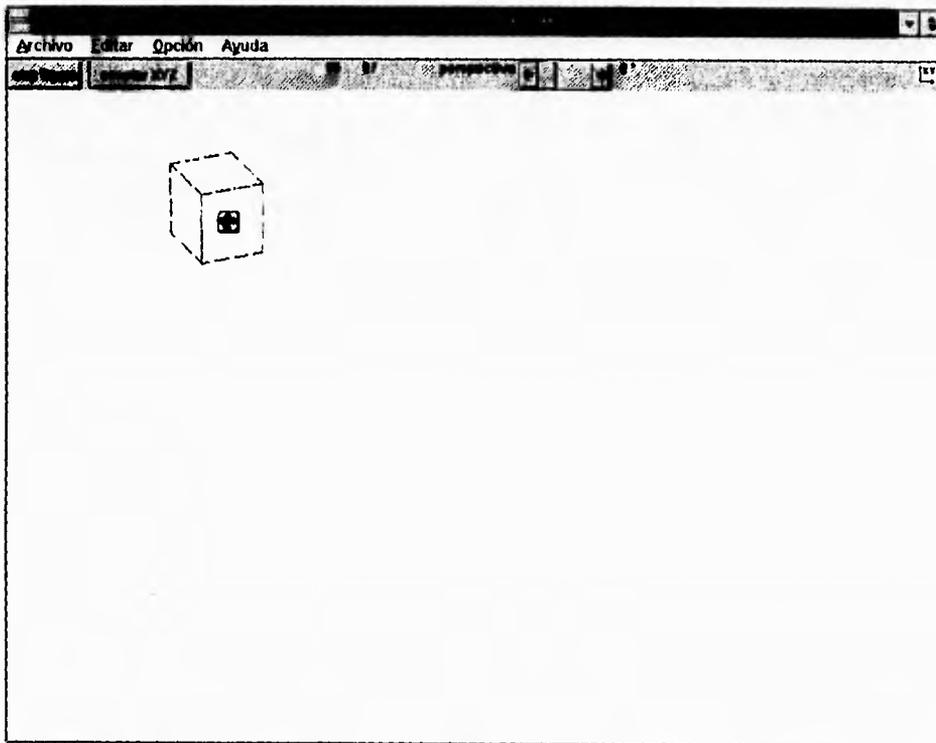


Fig. A-24



Fig. A-25



Fig. A-26

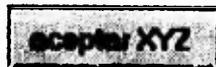


Fig. A-27



Fig. A-28

## MANUAL DE OPERACION DEL USUARIO

Un ejemplo de cómo se ve un diseño después de haber aceptado algunas figuras es el que se muestra en la fig. A-29. En la ilustración se aprecia una pirámide con base de seis lados que se encuentra atrás de unos cubos. Uno de estos cubos es sólido y oculta una parte de la pirámide el otro cubo es transparente y por lo tanto no tapa al pirámide. El tercer cubo que se puede apreciar es sólido pero está punteado ya que aún no es aceptada su posición en el espacio.

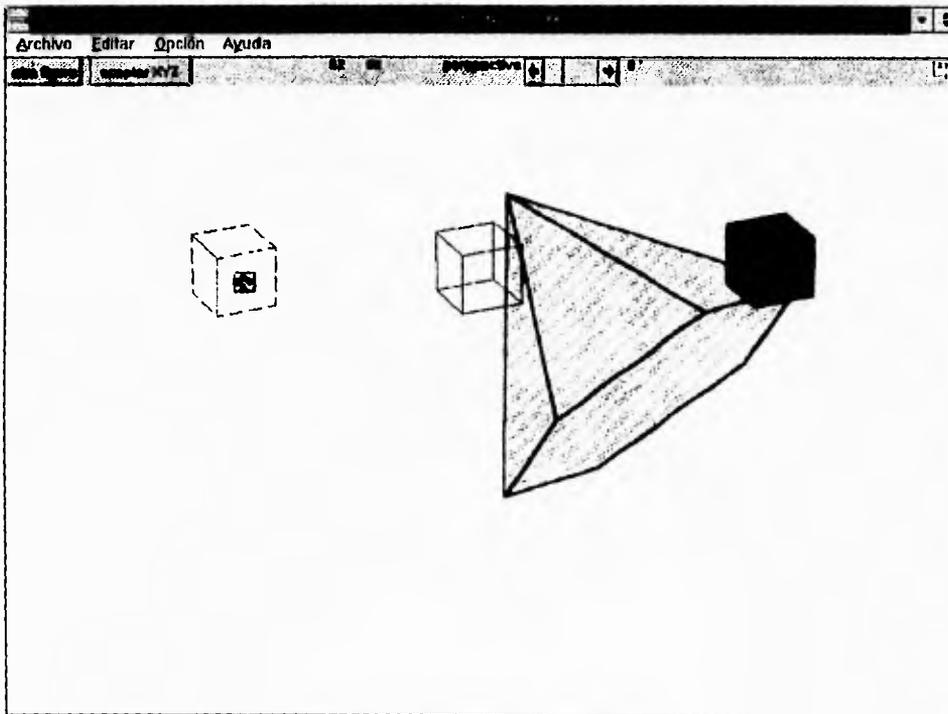
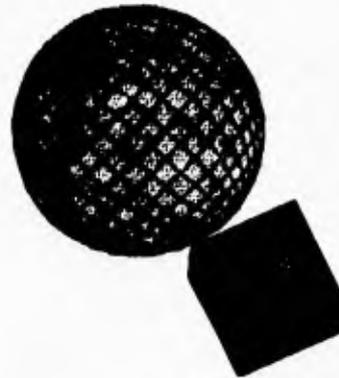
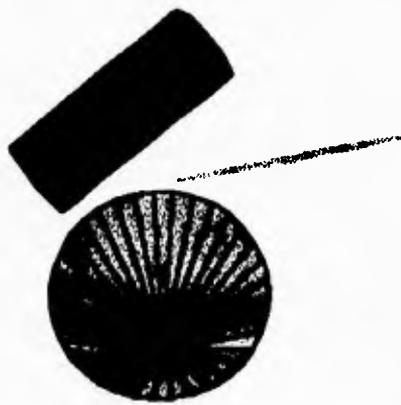


Fig. A-29



**APENDICE B**

**METODOS DE:  
ORDENAMIENTO Y  
SUPRESION DE CARAS OCULTAS**



## **APENDICE B**

### **1.- METODOS DE ORDENAMIENTO**

El ordenamiento es una de las operaciones más importantes que se practica sobre una estructura de datos. Ordenar una estructura de datos es establecer un orden de precedencia entre los elementos de la estructura, de acuerdo a uno o más campos (llaves) que se seleccionan para tal fin.

Para lograr esto, se han desarrollado una gran cantidad de algoritmos. A continuación se dan elementos para el análisis y la selección de los métodos.

#### **1.1.- Generalidades.**

Es frecuente encontrar operaciones de ordenamiento como parte de los procesos para el manejo electrónico de datos, Es conveniente mantener ordenado un conjunto de datos para facilitar la actualización o recuperación de ellos. Ordenar un conjunto de datos puede parecer una operación trivial, pero en realidad es una operación costosa que deberá realizarse solamente cuando sea estrictamente necesario y en este caso seleccionar el método apropiado.

#### **1.2.- Consideraciones para Selección del Método.**

Para seleccionar un método de ordenamiento en una cierta situación, es aconsejable considerar lo siguiente:

- 1.- El tipo de memoria en la que se encuentran los datos. Esta puede ser de acceso directo y de alta velocidad, de acceso directo y de mediana velocidad, y de acceso secuencia.
- 2.- Las características del sistema operativo para el manejo de archivos y de la memoria.
- 3.- Los tiempos de acceso a los dispositivos.
- 4.- La cantidad, el tipo y la distribución inicial de los datos.
- 5.- La eficiencia del método. Para establecer la eficiencia de un método de ordenamiento, basta con determinar el número promedio de comparaciones y de intercambios, así como la cantidad de memoria que requiere.

Para tener una idea del comportamiento de los algoritmos de ordenamiento, los mejores métodos realizan un número de comparaciones proporcional a  $n \log_2 n$ . ( $n$  = numero de elementos a ordenar).

## MÉTODOS DE ORD. Y DE SUPRESION DE CARAS

### **1.3 Ordenamientos Internos.**

Los métodos de ordenamiento que se utilizan para un conjunto de datos almacenados en una memoria de acceso directo de alta velocidad, son llamados métodos de ordenamiento interno. Estos se encuentran clasificados en métodos de selección, intercambio, inserción distribución e intercalación de acuerdo al principio en el que se basan.

La siguiente tabla muestra los algoritmos más importantes en cada caso.

Método por selección

- DIRECTA
- REPETITIVA
- TORNEO
- HEAP

Método por intercambio:

- BURBUJA
- TRANSPOSICIÓN DE PARES Y NONES
- EMBUDO
- QUICK

Método por inserción:

- DIRECTA
- BINARIA DE DOBLE ENTRADA
- SHELL

Método por distribución:

- CUBETAS

Método por intercalación:

- MEZCLA

En muchas de las aplicaciones de los ordenamientos, la información contenida en los registros es grande y, generalmente, el o los campos que la identifican son pequeños. En los siguientes métodos se toma de cada registro solamente los campos considerados como llaves.

## **APENDICE B**

---

### **1.3.1.- Métodos por Selección**

Los métodos de selección, como su nombre lo indica, seleccionan del conjunto de datos, según el criterio que se siga, al mayor o al menor de los datos del conjunto y lo excluye para proceder sobre los restantes de manera similar.

#### **1.3.1.1.- Selección Directa**

Este método consiste en encontrar el elemento menor, excluirlo del conjunto y continuar con el mismo criterio con los restantes.

#### **1.3.1.2.- Selección Repetitiva**

Este método es una variante del anterior y consiste en dividir el conjunto de elementos en grupos de raíz  $n$  elementos cada uno. De cada uno de estos grupos se extrae el menor y se copia (o almacena) en un grupo extra. De este grupo extra se obtiene el menor, y se excluye de este grupo y del grupo del que provino. Su lugar en el grupo extra lo ocupará el elemento menor del grupo del que provino el dato seleccionado anteriormente.

#### **1.3.1.3.- Torneo**

El conjunto de datos es estructurado como un árbol binario en el que las hojas del árbol son los datos originales. En el nivel  $m-1$  se tienen los datos que resultan ser los menores de cada par. Esta idea se extiende a todos los niveles hasta tener en la raíz al menor de todo el grupo. Para continuar con el procedimiento, se sustituyen los nodos que contienen al menor por un número muy grande y se realizan nuevamente las comparaciones entre los pares de nodos afectados por la sustitución.

#### **1.3.1.4.- Heap**

Un heap es un árbol estrictamente binario completo al que pueden faltarle algunas de las hojas situadas más a la derecha del último nivel. Solamente existe un nodo de grado uno cuya hoja es la última de izquierda a derecha.

En este árbol debe seguirse el criterio de que el valor de cualquier nodo es mayor o igual que el valor de cualquiera de sus hojas.

## MÉTODOS DE ORD. Y DE SUPRESION DE CARAS

### **1.3.2.- Métodos por Intercambio**

Los métodos por intercambio transponen o intercambian sistemáticamente pares de datos que se encuentran fuera de orden hasta que dejen de existir.

#### **1.3.2.1.- Burbuja**

El nombre burbuja se debe a la manera en que se mueven los datos dentro del conjunto, aparentando ser burbujas en el agua subiendo la superficie. El algoritmo se inicia comparando las llaves  $n$  y  $n-1$ , y las intercambia si  $n < n-1$ , compara después  $n-1$  y  $n-2$  y las intercambia si  $n-1 < n-2$ . Este procedimiento se practica hasta comparar los datos 1 y 2 y, cuando esto sucede, el menor dato a alcanzado su posición final.

#### **1.3.2.2.- Transposición de Pares y Nones**

El algoritmo consiste en dos etapas. En la primera, se comparan los datos nones con su sucesor y se les intercambia si es necesario. En la segunda, se comparan los datos pares con su sucesor y se les intercambia si es necesario. Este proceso se practica hasta que en ambas pasadas no se encuentra un intercambio.

#### **1.3.2.3.- Embudo**

Este algoritmo es muy similar al de la burbuja, excepto que las pasadas se alternan, una para colocar la llave menor, otra para colocar la llave mayor y así sucesivamente.

#### **1.3.2.4.-..Quick**

El algoritmo de quick consiste en intercambiar los datos para colocarlos en orden con respecto a uno de ellos ( $D$ ), de tal manera que a la derecha de  $D$  queden los datos mayores a  $D$  y a la izquierda los datos menores a  $D$ . Este procedimiento se repite con las sublistas, a la izquierda y derecha de  $D$  hasta que todas las sublistas tengan únicamente un solo dato.

## **APENDICE B**

---

### **1.3.3.- Métodos por Inserción**

En estos métodos se suponen que el conjunto de llaves se encuentra ordenado. Para una llave K que se desea agregar al conjunto, se determina el lugar que debe ocupar y se mueven los datos una posición para insertarlo en su posición correcta.

#### **1.3.3.1.- Inserción Directa**

El algoritmo considera que el conjunto ordenado sólo tiene un elemento y a partir de éste se irán insertando los demás.

#### **1.3.3.2.- Inserción Binaria de Doble Entrada**

Este método consiste en comparar el dato con el elemento central del conjunto de datos y determinar si las comparaciones continúan sobre los datos a la derecha del elemento central o a la izquierda. Este proceso se repite hasta determinar entre que elementos debe estar el dato.

#### **1.3.3.3.- Shell**

El algoritmo Shell clasifica un conjunto de llaves ordenando sucesivamente subconjuntos de llaves que están entremezclados en todo el conjunto. Estos subconjuntos están definidos por la secuencia  $h(1)$ ,  $h(2)$ ,  $h(3)$ , ...,  $h(t-1)$ ,  $h(t)$ , llamados incrementos, que determinan qué elementos pertenecen a un subconjunto. El incremento final  $h(t)$ , es siempre 1.

### **1.3.4.- Método por Distribución**

Los métodos de distribución son análogos al proceso de ordenamiento que emplea un clasificador de tarjetas, método que consiste en una colección de casilleros, donde se depositan las tarjetas que coinciden con la marca del casillero.

#### **1.3.4.1.- Cubetas**

El algoritmo de radix sort o bucket sort es de los más representativos entre los métodos de distribución. En estos algoritmos podemos diferenciar dos fases: una de distribución de los datos sobre los buckets o cubetas, y la otra, de recolección de los datos.

## **METODOS DE ORD. Y DE SUPRESION DE CARAS**

### **1.3.5.- Método por Intercalación**

Los métodos de ordenamiento por intercalación entremezclan dos o más conjuntos de datos para formar un nuevo conjunto ordenado.

#### **1.3.5.1.- Mezcla**

El caso típico en los procesos de mezcla consiste en entremezclar dos archivos, A y B, ordenados para formar un nuevo archivo C ordenado. Para realizar esta operación, los dos primeros elementos de A y B, son comparados, y el menor se mueve a C. Después se compara el dato siguiente del conjunto ganador, con el otro, y el menor se mueve a C; este proceso continúa hasta que uno de los conjuntos se agote y simplemente el otro sea copiado a C.

## **2.- METODOS DE SUPRESION DE CARAS OCULTAS**

### **2.1.- Método de preselección radial**

Si se decide inicialmente la dirección desde la cual va a ser visto el modelo, entonces sólo se tiene que dibujar las superficies visibles desde ese punto de vista. Esta solución al problema produce un modelo aparentemente sólido, pero de hecho no es más que una cascara. A esta técnica se le llama método de preselección radial, porque la dirección (ángulo radial) ha sido predefinido.

### **2.2.- Método de selección radial**

Mientras que el programador debe hacer todo el trabajo en el anterior método, en este segundo método es la computadora quien se encarga de dicho trabajo. El programa determina qué superficies son visibles u ocultas, dependiendo del ángulo de visión. En este método se dibuja primero las caras que se encuentran atrás, es decir, se determina que caras se encuentran más lejos y los dibuja primero, así las caras delanteras tapan a las de atrás.

### **2.3.- Método de la ecuación del plano**

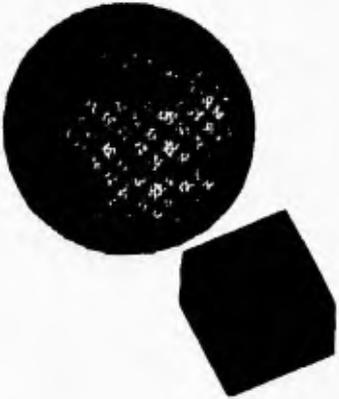
Utilizando la multiplicación de vectores es posible determinar si un punto está a un lado u otro de un plano dado. Y ya que las caras del cubo son planos, puede usarse este método para determinar si el punto de vista está a un lado u

## **APENDICE B**

---

otro de la cara. Esta relación entre cara y punto de vista determina si la superficie puede o no ser visible.

De estos tres métodos, el que más conviene utilizar para poder generar los modelos 3D es el de la ecuación del plano, ya que solamente dibuja las caras necesarias y no todas.



**APENDICE C**

**DICCIONARIOS**



## APENDICE C

### DICCIONARIO DE VARIABLES DEL SISTEMA PROPUESTO

Los elementos utilizados en los Diagramas Lógicos de Flujo de Datos del sistema propuesto se definen a continuación.

**ABIERTO.** Variable que determina si se abrió un archivo existente y habilita las opciones para REGRESAR .

**ABRIRARCHIVO.** Variable que contiene el nombre del archivo que se utiliza.

**CARACTS\_FUENTE.** Variable que contiene el tipo, tamaño y otras características de la letra fijada.

**COLORCARA.** Variable que indica el color de la cara de la figura.

**COLORLINEA.** Variable que indica el color del borde de la figura.

**FIGURA.** Variable que contiene el nombre de tipo de figura a realizar ('LINEA', '2D' y '3D').

**FLECHAS.** Bandera que determina si es visible el desplazador de figuras.

**INFORMACION.** Registro que contiene las características de la figura en el archivo de datos del diseño y se utiliza cuando se escribe en el archivo.

**LEIDO.** Registro que contiene las características de la figura en el archivo de datos del diseño y se utiliza cuando se lee el archivo.

**LETRERO.** Variable que contiene el texto que se va a escribir en el diseño.

**MX.** Variable que contiene la coordenada X del centro de la figura (desplazamiento en X).

**MY.** Variable que contiene la coordenada Y del centro de la figura (desplazamiento en Y).

**MZ.** Variable que contiene la coordenada Z del centro de la figura (desplazamiento en Z).

**NOMFIG.** Variable que contiene el complemento del nombre de las figuras PRISMAS Y PIRAMIDES (indica el número de lados de la base).

**NUMFIGURA.** Variable que contiene el número de figuras aceptadas.

**OTRALISTA.** Variable que indica el tipo de figura ('2D', 'PIRAMIDES' o 'POLIEDROS'), determina la FORMA que se abrirá y los valores que se asignan a NOMFIGURA, TITULO y TIPO.

**OK.** Indica que el proceso se ha realizado.

**R1.** Variable que indica el número de radianes de rotación con respecto al eje Y.

**R2.** Variable que indica el número de radianes de rotación con respecto al eje Z.

**R3.** Variable que indica el número de radianes de rotación con respecto al eje X.

**ROTVISX.** Variable que indica el número ángulo de perspectiva.

## DICCIONARIOS

- SALVAPER.** Variable que contiene el nombre del archivo con el que será salvada la perspectiva.
- SOL.** Variable que determina si la figura es sólida o transparente.
- TIPO.** Variable que indica el tipo de procedimiento que se va a utilizar.
- TITULO.** Variable que almacena el número que corresponde a la figura.
- VALR1.** Variable que indica el número de grados de rotación con respecto al eje Y.
- VALR2.** Variable que indica el número de grados de rotación con respecto al eje Z.
- VALR3.** Variable que indica el número de grados de rotación con respecto al eje X.
- VALX.** Variable que indica el tamaño de la figura (ancho).
- VALY.** Variable que indica el tamaño de la figura (alto).
- VALZ.** Variable que indica el tamaño de la figura (profundidad).
- XV.** Variable que almacena el tamaño de la figura (ancho).
- YV.** Variable que almacena el tamaño de la figura (alto).
- ZV.** Variable que almacena el tamaño de la figura (profundidad).

### DICCIONARIO DE TERMINOS

Los siguientes términos son mencionados a lo largo de los capítulos dar su significado descripción.

- 8086/88:** En 1981, se lanzó el chip 8088 de Intel en las PC originales. Tanto él como el 8086 (versión más rápida) son buenos para procesamiento de texto y aplicaciones de bajo volumen. Para incrementar la memoria más allá de un megabyte, se usan plaquetas de memoria EMS.
- 286:** El 80286 de Intel, usado por primera vez en la IBM AT en 1984, provee a los usuarios una mayor respuesta de máquina que los modelos basados en 8088. Memoria adicional, más allá de un megabyte, puede ser agregada y configurada ya sea como memoria extendida o memoria EMS, dependiendo de los programas y del entorno operativo usado en ella.
- 2 Dimensiones:** Número de dimensiones que tienen las figuras planas y que son anchura y altura.
- 386 Y 386SX:** El Intel 80386, usado por primera vez en el Compaq 386, es una computadora de alto rendimiento con una arquitectura mucho más avanzada y flexible que el 286. Tiene un modo 86 virtual especial,

## **APENDICE C**

---

que puede ejecutar múltiples aplicaciones de DOS simultáneamente y puede facilitar la transición desde DOS a OS/2 cuando sea necesario. El 386SX es una versión de velocidad más lenta del 386 con la misma flexibilidad.

**3 Dimensiones:** Numero de dimensiones que se manejan con los cuerpos sólidos, estas dimensiones son: anchura, altura y profundida.

**486:** Con los modelos iniciales presentados a fines de 1989, el Intel 80486 es una versión de mayor velocidad del 386 y tiene un coprocesador matemático incorporado. El 486 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.

**Algoritmo:** Proceso de cálculo que permite llegar a un resultado final.

**ALU (Arithmetic Logic Unit):** unidad aritmética lógica. Circuito de alta velocidad en la CPU que realiza las comparaciones y cálculos.

**American Bell:** El nombre usado por AT&T inmediatamente después de su disociación legal, que tuvo lugar el 16 de enero de 1984. Más tarde se les ordenó suprimir el nombre Bell.

**ASCII (American Standard Code for Information Interchange):** Código Americano Estandard para Intercambio de Información Código binario de datos que se usa en comunicaciones, en la mayor parte de las minicomputadoras y en todas las computadoras personales.

**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).

**API (Interfaz de programa de aplicación):** Lenguaje y formato utilizados por un programa para comunicarse con otro programa. También puede incluir los comandos utilizados para interrumpir a la computadora con el fin de llamar la atención a otro programa. Un API utilizado en comunicaciones se llama protocolo.

**Bit (dígito binario):** 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. La unidad de almacenamiento más común es byte, constituido por ocho bits y equivalente a un carácter alfanumérico.

**Bit map (mapa de bits):** En gráficos por computadora, un área de memoria que representa la imagen de video. Para pantallas monocromáticas, un bit en el mapa de bits representa un pixel de la pantalla. Para la escala del gris o pantallas de color, varios bits en el mapa de bits representan un pixel o grupo de pixeles en la pantalla. También se puede definir como cualquier representación binaria en donde cada bit o conjunto de bits corresponde a algún objeto o condición.

**Bitblt (transferencia de bloques de bits):** En gráficos por computadora, una facilidad de hardware que mueve un bloque rectangular de bits desde la memoria principal a la memoria del monitor a alta velocidad. La tecnología bitblt mejora el rendimiento de los computadores utilizados en animación en tiempo real y modelado sobre pantalla.

**Bits en color:** Número de bits asociado con cada elemento en pantalla que representa su color. Para 16 colores, se utilizan cuatro bits; y para 256 colores ocho bits.

**BMP (Mapa de Bits):** Formato de gráficos Windows que se puede desarrollar de forma independiente o dependiente. Los archivos de AIB independientes para dispositivos, se codifican para su conversión en una gran variedad de visualizaciones y de impresoras.

**Byte (Octeto, byte):** La unidad 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 de 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 de 0 a 9 (decimal), dos dígitos numéricos (decimal empaquetado) o un número entre 0 y 255 (números binarios).

**C:** Un lenguaje de programación de alto nivel desarrollado en Bell Labs, que es capaz de manipular la computadora a bajo nivel, tal como lo haría un lenguaje ensamblador. Durante la segunda mitad de la década del 80, el C se convirtió en el lenguaje elegido para el desarrollo de software comercial.

**CGA de exploración doble:** Un método de hardware que duplica la resolución efectiva del estándar de presentación CGA, sin requerir un cambio en el software.

**CGI (Interfaz de Computadoras para Gráficos):** Lenguaje independiente de gráficos para dispositivos y en particular para pantallas de

## APENDICE C

visualización, impresoras y trazadores gráficos que surgieron de GKS.

**Click:** Seleccionar un objeto presionando el botón del ratón cuando el cursor está apuntado la opción de menú o icono deseados.

**Color de alta densidad:** Capacidad de generar 32,768 colores (15 bits) o 65.536 colores (16 bits). El color de 15 bits utiliza cinco bits para cada punto en pantalla rojo, verde y azul. El bit 16 puede ser un color, como XGA con cinco rojos, seis verdes y cinco azules, o ser un bit de solapamiento que selecciona puntos en pantalla para visualizar entrada de video.

**Convexo:** Dícese de la línea o superficie curvas que, respecto del que las mira, tiene su parte más prominente en el centro.

**Coordenada:** Cada una de las líneas que sirven para determinar la posición de un punto.

**Coprocador:** Un procesador secundario empleado para acelerar las operaciones, llevando a cabo parte de la carga de trabajo de la CPU principal.

**CPU (Control Processing 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.

**CRT (Cathode Ray Tube):** tubo de rayos catódicos El tubo de vacío utilizado como pantalla de presentación en una terminal de video o en TV. El término se utiliza con frecuencia para referirse a la terminal en sí misma.

**Cursor:** Un símbolo móvil en una 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 cuando se desplaza a diferentes zonas de la pantalla.

**DDE (Dynamic Data Exchange):** Intercambio Dinámico de Datos El protocolo de mensajes de Microsoft Windows que permite que los programas de aplicación pidan e intercambien los datos automáticamente. Una programa en una ventana puede interrogar a un programa en otra ventana utilizando el protocolo DDE.

**Dimensión:** Cada una de las magnitudes necesarias para la evaluación de las figuras planas y de los sólidos (longitud, anchura, altura o profundidad).

- DLL (dynamic link library):** librerías de Enlace Dinámico. Un conjunto de rutinas de programa que están disponibles para las aplicaciones en tiempo de ejecución.
- 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. El DOS es a veces denominado PC-DOS, para diferenciarlo del MS-DOS, la versión de Microsoft para las PC no IBM. Ambos productos son prácticamente idénticos y a ambos se los llama DOS.
- EGA (Adaptador de gráficos mejorado):** Estándar de exhibición de video de IBM que provee textos y gráficos de resolución media. EGA soporta los modos de presentación previos y requiere un monitor digital RGB Enhanced Color Display o equivalente. EGA ha reemplazado por VGA.
- EMS (Expanded Memory Specification):** Especificación de Memoria Expandida. Una técnica para expandir la memoria más allá de un megabyte en PCs corriendo bajo DOS. La versión 4.0 de EMS incrementa el total de memoria con la que las aplicaciones DOS pueden trabajar desde uno a 32MB, suministrando una capacidad de conmutación de bloques que permite a segmentos de la memoria convencional apuntar a la memoria EMS
- Entrada/salida (E/S):** Una transferencia de datos entre la CPU y un dispositivo periférico. Toda transferencia es una salida de un dispositivo y una entrada en otro.
- Espacio:** Extensión indefinida de tres dimensiones que constituye el objeto de la geometría del espacio.
- EXE (Archivo ejecutable):** Programa ejecutable en DOS, OS/2 y VMS. En DOS, un programa que cabe en 64K, puede ser un archivo COM.
- Figura:** Espacio cerrado por líneas o superficies.
- Geometría:** Disciplina matemática que tiene por objeto el estudio riguroso del espacio y de las formas que en él se pueden imaginar.
- GKS (Graphical Kernel System):** Un sistema y lenguaje de gráficos para crear imágenes gráficas en 2-D, 3-D y trama. Es un sistema independiente del dispositivo que permite a los programas de aplicación crear y manipular gráficos en muchas variedades de dispositivos de presentación. GKS permite que las aplicaciones sean desarrolladas en un sistema y fácilmente llevadas a otro con mínimo cambio o sin él. GKS es el primer verdadero estándar para

## **APENDICE C**

---

programadores de aplicaciones gráficas y ha sido adoptado por ANSI e ISO.

**Gráficos:** Con respecto a la computadora, la creación y administración de figuras. Las figuras pueden ser introducidas en la computadora usando dispositivos de entrada como tarjetas gráficas, ratones o lápices de luz, y las figuras existentes en papel pueden ser pasadas a la computadora utilizando exploradores o cámaras.

Una vez almacenadas, las figuras pueden ser manipuladas y copiadas de muchas maneras. Los colores pueden ser cambiados, los objetos pueden ser aumentados o disminuidos en tamaño, torcidos, comprimidos y aplastados. Cuadros de grabaciones de video pueden ser combinados con objetos dibujados.

Pueden agregarse descripciones de texto para producir diagramas, informes, folletos y otros tipos de materiales de presentación.

**Gráficos Animados:** Dibujos o diagramas en movimiento. Se encuentran frecuentemente en programas didácticos basados en computadoras; los gráficos animados ocupan mucho menos espacio en disco que las imágenes en video.

**Gráficos Hercules:** Un estándar de exhibición por video común para PCs, de Hercules Computer Technology Inc., que provee gráficos monocromáticos y texto con una resolución de 720x 348 pixels. El primer video monocromático para PC de IBM no proveía gráficos, y Hercules presentó su adaptador de exhibición de cubrir el vacío en 1982. Rápidamente se convirtió en un estándar de hecho.

**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:** Son todos los componentes físicos de un sistema de computo, como lo son la CPU, el monitor y periféricos.

**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 del ratón. Los iconos pueden ser desplazados en la pantalla.

**Lenguaje de alto nivel:** Un lenguaje de programación independiente de la máquina, como VISUAL BASIC, Pascal y C. Los lenguajes de alto nivel permiten a los programadores concentrarse en la lógica de los problemas a ser resueltos en vez de la intrincada arquitectura de la

máquina, como requiere en los lenguajes ensambladores de bajo nivel.

**Lenguaje de propósitos generales:** Un lenguaje de programación que puede ser utilizado para resolver una amplia variedad de problemas. FORTRAN, COBOL y BASIC son ejemplos de lenguajes para propósitos generales.

**Línea:** Extensión considerada en una sola de sus tres dimensiones: la longitud.

**Matriz de puntos:** Una patrón de pequeños puntos que forman los caracteres y las imágenes gráficas en las pantallas de video y las impresoras. Las pantallas de presentación usan una matriz (Conjunto de filas y columnas) de puntos al igual que los aparatos de televisión. A mayor densidad de puntos, mayor será la resolución de los caracteres y gráficos.

**Memoria:** Almacenamiento de trabajo de la computadora, que físicamente es una colección de chips RAM. 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.

**Monitor analógico:** Monitor de video que acepta una señal analógica de la computadora (la conversión digital a analógico se realiza en el controlador del video). Los monitores analógicos se pueden diseñar para que acepten una pequeña gama de resoluciones de visualización por ejemplo (solamente VGA, o VGA y Super VGA), o monitores analógicos multisync que pueden aceptar una amplia gama de resoluciones incluyendo TV (NTSC). Los monitores analógicos en color aceptan señales separadas, y en rojo, verde y azul para mostrar un contraste más definido.

**Monitor Digital:** Monitor de video que acepta una señal digital de la computadora y la convierte en analógica para iluminar la pantalla. Los monitores digitales aceptan resoluciones de visualización específicas de computadoras, como CGA y EGA.

**Mouse (Ratón):** 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 el escritorio en cualquier dirección, el cursor o puntero se mueve correspondientemente sobre la pantalla.

## APENDICE C

- Oblicuo:** Que no es perpendicular ni paralelo a un plano o línea dados. Que está inclinado, que no es vertical ni horizontal.
- OLE (Object Linking and Embedding):** Enlace e Incrustación de Objetos. Protocolo de documentos compuestos de Microsoft Windows. La aplicación "cliente" crea el documento; la aplicación "servidor" crea un objeto dentro del documento. Cuando un usuario tecléa dos veces en un objeto empotrado en una aplicación "cliente", se carga la aplicación "servidor" y se recupera el archivo de datos adecuado.
- Paralelo:** Dícese de dos o más rectas que, dos a dos, se encuentran en un mismo plano y no se cortan.
- PC (Personal Computer):** computadora personal Todas las máquinas que se ajustan a los estándares de IBM PC y PS/2.
- Perpendicular:** Que forma un ángulo recto ( $90^\circ$ ).
- Plano:** Superficie ilimitada que contiene la totalidad de la recta que une dos de sus puntos.
- Programa de dibujo:** Software gráfico que permite al usuario diseñar e ilustrar productos y objetos. Los programas de dibujo mantienen una imagen en formato de gráficos vectoriales, lo que permite que todos los elementos del objeto gráfico puedan ser aislados y manipulados individualmente.
- PROM (Programmable Read Only Memory):** memoria programable de sólo lectura Un chip de memoria permanente que es programado, o llenado por el cliente, en lugar del fabricante de chips. Nótese la diferencia con ROM, el cual es programado al momento de su fabricación.
- Pixel (PIX [picture] Element):** Elemento de imagen, pixel. 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.
- Punto:** Figura geométrica sin dimensiones.
- Software:** Instrucciones para una computadora. Una serie de instrucciones que realizan una tarea en particular se llama programa o programa de software. Las dos categorías principales son software de sistemas y de aplicaciones.
- Superficie:** Figura geométrica definida por el conjunto de puntos del espacio cuyas coordenadas verifican una función.
- RAM (Random Access Memory):** memoria de acceso aleatorio Lo mismo que memoria.

## DICCIONARIOS

**ROM (Read Only Memory):** Memoria de Sólo Lectura Chip de memoria que almacena permanentemente instrucciones y datos. Sus contenidos se crean en el momento de la fabricación y no se pueden alterar. Se utiliza ampliamente para almacenar rutinas de control en computadoras personales (ROM BIOS) y en controladores de periféricos, también se utiliza en cartuchos conectables para impresoras, video juegos y otros sistemas. Cuando el software se almacena en ROM la actualización a la versión siguiente requiere volver a colocar el chip de la ROM.

**Vector:** Segmento de recta orientado en el que se distingue un origen y un extremo.

**Vértice:** Punto en el que concurren los dos lados de un ángulo. Punto donde concurren tres o más planos.

**VGA (Matriz array gráfica de video):** Un estándar de presentación de video de IBM, que 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 comunmente hay tarjetas disponibles para todas las máquinas de bus PC y AT.

La VGA soporta estándares de presentación previos y requiere de un monitor RGB analógico. Tiene 16 colores en su máximo modo gráfico (640 x 480), pero los proveedores han aumentado los colores a 256 y han agregado un modo de resolución más alta, "Super VGA", de 800 x 600.

La Video Electronics Standards Association (VESA) ha establecido un estándar para Super VGA; un paso importante, ya que muchas versiones independientes han sido desarrolladas.



**APENDICE D**

**BIBLIOGRAFIA**



## **APENDICE D**

---

♣ **Sistemas Operativos**  
Conceptos y Diseño  
MILENKOVIC, Milan.  
España 1988  
Edit. Mc Graw Hill

♣ **Ingeniería de Software**  
FAIRLEY, Richard.  
México 1988  
Edit. Mc Graw Hill

♣ **Ingeniería de Software**  
SOMMERVILLE, Ian.  
México 1988  
Edit. Adisson-Wesley  
Iberoamericana

♣ **Ingeniería del Software**  
Un enfoque práctico  
PRESSMAN, Roger S.  
México 1988  
Edit. Mc Graw Hill

♣ **Curso de programación con C**  
Microsoft c  
CEBALLOS, Francisco Javier.  
México 1990  
Edit. Macrobit

♣ **Tarjetas de Video**  
Informática Windows  
SHELDON, Tom.  
México 1989  
Edit. Mc Graw Hill

♣ **Master Visual Basic 3**  
GUREWICH, Nathan.y  
GUREWICH, Ori.  
Estados Unidos de Norteamérica  
1994  
Edit. SAMS Publishing

♣ **Organización de Computadoras**  
Un enfoque estructurado  
TANENBAUM, Andrew S.  
México 1988  
Edit. Prentice Hall Iberoamericana

♣ **Guía completa de Visual Basic**  
para Windows  
NELSON, Ross.  
México 1993  
Edit. Mc Graw Hill

♣ **Gráficos por computadoras**  
HEARN, Donald. y BAKER, M.  
Pauline  
México 1983  
Edit. Prentice Hall

♣ **Programación avanzada de**  
graficos interactivos  
Modelado, Acabado y Animación  
en 2D y 3D  
LEE, Adams  
España 1980  
Edit Anaya Multimedia

♣ **Solid Geometry**  
WELCHONS, A.M. y  
KRICKENBERG, W.R.  
Estados Unidos de Norteamérica  
Edit. Ginn

## **BIBLIOGRAFIA**

### **REVISTAS**

♥ Tarjetas SVGA: lo  
suficientemente rápidas para  
Windows, lo suficientemente  
baratas para usted  
PC Magazine en español  
Noviembre de 1991  
Mary Kathleen Flynn y Robert W.  
Kane

♥ Alta resolución y alta velocidad  
PC Magazine en español  
Octubre de 1991  
Alfred Poor

♥ Mejorando la ejecución de videos  
bajo Windows 3.1  
PC Magazine en español  
Junio de 1991  
Manuel López M.

♥ Procesadores 486 DX ¿Queé...?  
Personal Computing Mexico  
Septiembre de 1994  
Jorge Arturo Piñon Pavón