

6  
2ej

# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO



ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

SANTA CRUZ ACATLAN

## SISTEMA DE SIMULACION CON GRAFICACION HASTA EN TRES DIMENSIONES

TESIS CON  
FALLA DE ORIGEN

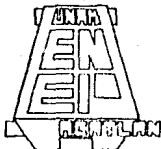
T E S I S

QUE PARA OBTENER EL TITULO DE

LICENCIADO EN MATEMATICAS  
APLICADAS Y COMPUTACION

P R E S E N T A :

VICENTE GONZALEZ JUAREZ





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 está 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.

## INDICE DE CONTENIDO.

	página
I. INTRODUCCION	1
1.1 Alcance que se pretende del sistema	3
1.2 Objetivo General	4
1.3 Objetivos Especificos	4
II. LA SIMULACION POR COMPUTADORA	6
2.1 Fundamentos racionales de la simulación en computadoras	6
2.2 Simulación	8
2.3 Simulación continua	8
2.4 Sistema de simulación	9
2.5 Modelo	9
2.6 Propiedades de los modelos para la simulación	10
2.7 Clasificación de los modelos para simulación	11
2.8 Modelo matemático	12
2.9 Modelo deterministas	13
2.10 Simulación de sistemas	13
2.11 Metodología para desarrollar un modelo de simulación	14
2.11.1 Diseño de modelos matemáticos	14
2.12 Planeación de los experimentos de simulación	16
2.12.1 Formulación de un problema (paso 1)	18
2.12.2 Recolección y procesamiento de datos (paso 2)	19
2.12.3 Formulación de los modelos matemáticos (paso 3)	21
2.12.4 Estimación de los parámetros de las características operacionales a partir de los datos reales (paso 4)	24
2.12.5 Evaluación del modelo y de los parámetros estimados (paso 5)	27
2.12.6 Formulación de un programa computacional (paso 6)	30
2.12.7 Validación (paso 7)	33
2.12.8 Diseño de experimentos de simulación (paso 8)	34
2.12.9 Análisis de los datos simulados (paso 9)	35
2.13 Aspectos para la construcción de un modelo de simulación	37
2.14 Importancia de la graficación en la simulación	38
III. LA GRAFICACION POR COMPUTADORA	39
3.1 Módulos Funcionales de un Sistema de Graficación	41
3.2 Graficación de un objeto	44
3.2.1 Graficación de un punto	44
3.2.2 Conversión de un punto en pantalla por rastreo	45
3.2.3 Graficación de una línea recta por rastreo	46
3.3 Sistema Coordinado	47

3.4	Cambio de Coordenadas	47
3.5	Herramientas de apoyo para Graficación	49
3.5.1	Sistemas Coordenados Bidimensionales	48
3.5.2	Escalaiento	50
3.5.3	Rotación	51
3.5.4	Traslación	54
3.6	Dispositivos de entrada y salida en sistema de graficación	54
3.6.1	Dispositivos de exhibición	56
3.6.1.1	CRT de rastreo por barrido	57
3.6.1.2	Rastreo por barrido	57
3.7	Funciones de graficación	58
3.8	Normas de Software	59
3.9	Técnicas de despliegue tridimensional	62
3.9.1	Proyecciones	62
3.9.1.1	Proyecciones en Paralelo	64
3.9.2	Rotación en tres dimensiones	70
IV.	DESCRIPCION DEL SISTEMA	74
4.1	Requerimientos de equipo	75
4.2	Estructuras del sistema	76
4.2.1	Descripción de los programas que comprenden el sistema	70
4.2.1.1	GRAFICA1	78
4.2.1.2	GRAFICA2	80
4.2.1.3	GRAFICA3.PAS (GRAFICA4)	82
4.2.1.4	FUNCION.PAS (FUNCION2)	84
4.2.1.5	SIDEVIS	84
4.2.1.6	INSTRUYE.SDV	85
4.2.1.7	AYUDA.SDV	85
4.3	Partes que integran al sistema	85
4.3.1.	Editor de datos	86
4.3.2.	Editor de funciones	86
4.3.3.	Sección de simulación	87
4.3.4.	Sección de graficación de vectores de datos	87
4.3.5.	Sección de graficación de funciones	87
4.3.6.	Sección de ayuda al usuario	87
4.3.7.	Sección de interacción con el sistema operativo	88
4.3.8.	Opción de salida del sistema	88
4.4	Alcance y limitaciones del sistema	88
4.4.1	Limitaciones del sistema	88
4.4.2	Alcances del sistema	89
	CONCLUSIONES	91
	BIBLIOGRAFIA	94

## INTRODUCCION.

En el proceso de solución de todos aquellos fenómenos que incluyen al binomio teoría-realidad, el uso de computadoras electrónicas rápidas y precisas, la representación gráfica de los fenómenos cuantificables, que permiten su interpretación y simulación, han jugado un papel importante en la solución de muchos de estos fenómenos. Estos tres elementos básicos, serán el motivo de este trabajo, con especial énfasis en los aspectos de simulación y graficación; por lo anterior, se hará referencia a la importancia y características esenciales de cada uno de ellos, para posteriormente combinarlos y lograr el objetivo fundamental que se pretende desarrollar como trabajo de tesis y que es el desarrollo integral de un sistema de simulación determinista con facilidades de graficación.

En la actualidad, el bajo costo de las computadoras personales, ha permitido que las computadoras sean utilizadas para la enseñanza, no únicamente en las áreas especializadas de computación, sino en otras tales como Química, Física, Medicina, Psicología, Humanidades, Administración, etc. La tendencia actual en utilizar las computadoras cada vez más con este propósito tanto en instituciones públicas como en privadas, de nivel superior y siendo cada vez más utilizadas en los niveles inferiores como lo son las escuelas de enseñanza media superior y aún las escuelas primarias. Dichas escuelas o instituciones ponen a la mano de sus educandos, las ventajas en rapidez y

exactitud proporcionados por una computadora.

Por otra parte, la velocidad y precisión de cálculos matemáticos de las computadoras aumentó la utilización y aprovechamiento de técnicas que antes no eran utilizadas como son los métodos numéricos y la simulación. El uso de esta última, se ha extendido de manera tan rápida, que hoy día, las personas que trabajan en distintas disciplinas requieren con urgencia el comprender de manera más clara y profunda de sus elementos básicos.

Es muy frecuente que en el estudio de un problema, debido a la complejidad del mismo, no se entienda completamente el comportamiento de las variables involucradas y por tanto no se conozca con claridad o no se identifique el modelo matemático que lo representa, en tales casos se puede hacer uso de las técnicas de simulación para ir adecuando un modelo que represente adecuadamente al fenómeno de estudio, para luego experimentar en él.

Finalmente, para visualizar y comprender mejor el comportamiento de un fenómeno, se puede hacer uso de la graficación por computadora, para que, de esta manera, se obtenga una descripción gráfica de dicho fenómeno, lo cual permitirá, entender mejor las relaciones existentes entre las variables involucradas y así identificar los cambios que se producen al variar algún parámetro dentro de una función dada.

La representación gráfica de un fenómeno tiene múltiples ventajas para el entendimiento del mismo, ya que incluso puede

realizarse representaciones con animación, en las cuáles se pueden ir observando, los cambios a través del tiempo; asimismo, utilizando gráficas superpuestas se puede observar el efecto de las variaciones en uno o más parámetros del modelo.

Cabe señalar que la graficación ha venido a revolucionar a la computación en los últimos años, debido a que permite una apreciación visual de muchos fenómenos que pueden ser descritos por medio de computadoras.

Como se ha señalado, los procesos de simulación y de graficación se enlazan para lograr una mayor claridad de la solución de los problemas que representen en la teoría-realidad, todo ello con el auxilio imprescindible de las computadoras electrónicas que aseguren a la vez rapidez y exactitud en los cálculos matemáticos involucrados. Este enlace en un enfoque exclusivamente funcional, será el propósito fundamental de este trabajo de tesis.

### 1.1. Alcance que se pretende del sistema.

El alcance que se pretende con el desarrollo de esta tesis es esencialmente crear un sistema de simulación y graficación para microcomputadora que sea de fácil utilización y que permita construir el modelo de una situación real y realizar experimentos de simulación, bajo distintos valores de estos y condiciones iniciales para modelos simbólicos de tipo exclusivamente matemático y determinista.

Este sistema podrá realizar simulaciones a través del

tiempo, de fenómenos de tipo continuo y determinista, cuya representación o modelo matemático no involucren la solución de ecuaciones diferenciales. El sistema sería un gran apoyo en los niveles de enseñanza media superior, para la experimentación y observación del comportamiento a través del tiempo de fenómenos físicos, químicos, matemáticos, etc. Esta experimentación es muy fácil de realizar con este sistema, los resultados se obtienen rápidamente y es el medio más económico y rápido de experimentar, dicha experimentación redundará en una mejor comprensión por parte del alumno de los fenómenos que se están estudiando.

### 1.2. Objetivo General.

El objetivo general del presente trabajo de tesis es el de crear un sistema computarizado de simulación con facilidades de graficación, con gráficos hasta de tres dimensiones, para representar el comportamiento de un modelo matemático simbólico determinista, el cual servirá de apoyo en instituciones de enseñanza e investigación.

### 1.3. Objetivos Específicos.

Los objetivos específicos que se programa cubrir con este trabajo de tesis, son los siguientes:

- 1.- Crear un subsistema a través del cual se logre la captura de los datos requeridos para establecer la simulación y la graficación, para una, dos o tres variables, según sea el caso, así como del modelo matemático a simular; una vez capturados



estos datos, el subsistema realizará la simulación cargando los datos tanto en la Unidad Central de Proceso como en el disco, dejando todo listo para que el subsistema que ha continuación se describirá sea invocado en el proceso.

2.- Crear un subsistema que asimile los datos y los represente de manera gráfica, esto es, que dibuje en la pantalla los puntos que son representativos de los datos en el espacio de estudio, para después poder usar las distintas faces constituidas por: imágenes fijas, imágenes con escalamiento, imágenes con rotación, e imágenes con traslación; teniendo como premisa el que este subsistema se constituya en la representación gráfica del proceso de tipo determinista que se vaya a cuestionar. A los dos subsistemas descritos se les identificará como subsistema 1 y subsistema 2, respectivamente.

3.- Fusionar los subsistemas 1 y 2 para que den como resultado el enfoque dual que se requiere para la creación del sistema de simulación con graficación hasta de tres dimensiones que constituye al objetivo general de este trabajo de tesis.

## II. LA SIMULACION POR COMPUTADORA.

### 2.1. Fundamentos racionales de la simulación en computadores.

El fundamento racional para usar la simulación en cualquier disciplina, es la búsqueda constante del hombre por adquirir conocimientos relativos a la predicción del futuro.

La simulación puede tener muchas razones para su empleo, entre las cuales podemos mencionar a continuación algunos, sin que con esto signifique que estas sean las únicas razones, y estas que se mencionan pueden ser probablemente, motivo de discusión; dicho lo anterior, las razones en cuestión son las siguientes:

2.1.1. Mediante la simulación es posible estudiar y experimentar con diferentes variables incluyendo las interacciones que ocurren en el interior de un sistema dado, ya sea una empresa, industria, etc. o de una parte de él, es decir, de un subsistema.

2.1.2. A través de la simulación se pueden estudiar los efectos que ciertos cambios informativos, ya sea de organización o ambientales, que se ejercen en la operación de un sistema; dicho estudio se puede realizar al establecer alteraciones en el modelo y observar los efectos de éstas en el comportamiento del sistema.

2.1.3. La observación detallada del sistema que se está simulando, conduce a un mejor entendimiento del mismo y proporciona sugerencias para mejorarlo, mismas que de otro modo no podrían obtenerse. Hay, en realidad, una buena evidencia de que los seres humanos tienen una gran capacidad para entender el funcionamiento de los sistemas complicados y de que pueden

encontrar reglas de decisión casi óptimas, ó procedimientos de operación más eficientes, si tienen suficiente conocimiento experimental del sistema y si éste es lo suficientemente estable (1).

2.1.4. La simulación puede usarse como recurso pedagógico, para enseñarles tanto a estudiantes como a practicantes, como apoyo en el análisis estadístico y en la toma de decisiones. Entre las disciplinas en que la simulación ha sido utilizada con éxito para el mencionado propósito, pueden incluirse la administración de empresas, la economía, la medicina, la biología, la agronomía y el derecho (2).

2.1.5. Los juegos operacionales, han demostrado constituir un medio excelente para estimular el interés y el entendimiento de parte del participante y son particularmente útiles en la orientación de personas con experiencia en la disciplina relativa al juego (1).

2.1.6. La experiencia que se adquiere al diseñar un modelo de simulación en una computadora, puede ser más valiosa que la simulación en si misma. El conocimiento que se obtiene al diseñar un estudio de simulación sugiere, frecuentemente, cambios que pueden probarse, a través de la simulación, antes de implantarlos en el sistema real (1).

2.1.7. La simulación de sistemas complejos puede producir un valioso y profundo conocimiento acerca de cuáles de las variables son más importantes que otras en el sistema y cómo ellas obran

entre si.

2.1.8. La simulación puede emplearse para experimentar con situaciones nuevas acerca de las cuales tenemos muy poca o ninguna información, con el objeto de estar preparados para alguna eventualidad.

2.1.9. La simulación puede servir como una prueba de preservicio para ensayar nuevas políticas y reglas de decisión en la operación de un sistema, antes de tomar el riesgo con el sistema real.

2.1.10. Las simulaciones se menciona, son valiosas algunas veces, ya que proporcionan una forma conveniente de dividir un sistema complicado en subsistemas, cualquiera de los cuales puede ser modelado por un analista o por un equipo de expertos en esa área [3].

## 2.2. Simulación.

La simulación es una técnica ampliamente usada en todas las áreas de la ciencia, es particularmente útil cuando no es posible obtener soluciones inmediatas debido a que al tiempo de proceso para encontrar la solución es muy grande y debido a su alto costo [2].

## 2.3. Simulación continua.

La simulación continua tiene como fin el describir a los fenómenos que tienen un desarrollo continuo, esto es que los cambios que sufren las variables en el sistema son suaves.

Dadas las limitantes impuestas para el sistema con que se va a trabajar, se puede formular la definición de simulación, misma que servirá para poder describir la base de la estructura del sistema que se pretende estructurar con esta tesis.

Simulación, es una técnica numérica para conducir experimentos en una computadora digital, los cuales requieren ciertos tipos de modelos lógicos y matemáticos, que describen el comportamiento de un modelo de tipo determinista, en el cual los cambios que se observan reflejan continuidad en periodos de tiempo real [4].

#### 2.4. Sistema de simulación.

Un sistema es un agregado o conjunto de objetos o entidades entre las cuales existe una o más relaciones. En el caso particular de los sistemas dinámicos los cuales serán el objeto de este trabajo, las interacciones provocan cambios a través del tiempo. Estos cambios que ocurren dentro del sistema lo alteran con frecuencia [3].

#### 2.5. Modelo.

Un modelo es la manera conveniente de representar física, analógica o simbólicamente un sistema construido con el propósito de estudiarlo. Se caracterizan por ser descriptivos y explicativos. En esta tesis se trabajará únicamente con modelos simbólicos. Debido a que el propósito del estudio determina la naturaleza de la información que se requiere, no hay un modelo único para un sistema. El modelo construido se revisa y se va

transformando obteniéndose así diferentes aspectos del sistema, y diferentes niveles de detalle según cambie la comprensión del mismo.

En la simulación primero se debe identificar un modelo que represente el sistema que se desea estudiar. La función del modelo de simulación, más que describir la conducta general del sistema como ocurre en los fenómenos analíticos, describirá la operación del sistema en términos de eventos o sucesos que les ocurren a las componentes del sistema a través del tiempo. Se debe procurar que el modelo sea lo más sencillo posible, y que a la vez, represente lo mejor posible al sistema [3].

### 3.6. Propiedades de los modelos para la simulación.

Es posible definir el modelo como una abstracción de algún sistema real, que tiene la posibilidad de ser empleado para propósitos de predicción y control. El objeto del modelo es permitir al analista la determinación de uno o más cambios en los aspectos del sistema modelado que afectan otros aspectos del sistema. Para que un modelo sea útil, debe necesariamente incorporar elementos de dos atributos en conflicto: realismo y simplicidad. Por un lado, el modelo ha de servir como una aproximación razonable al sistema real e incorporar la mayor parte de los aspectos importantes de éste; por otro, no es conveniente que el modelo resulte tan complejo que resulte imposible entenderlo o manipularlo. Desafortunadamente, los modelos realistas rara vez son simples y en éstos últimos, rara vez son realistas [1].

## 2.7. Clasificación de los modelos para simulación.

Se han realizado muchos tipos de clasificaciones para los modelos de sistemas. Una clasificación es la que se realiza en términos de la naturaleza del flujo del tiempo en el sistema que modelan; así, los modelos se clasifican como modelos continuos y modelos discretos [3].

Un sistema continuo es aquel en que las actividades predominantes dentro del sistema provocan cambios suaves en los atributos de las entidades del mismo. Los modelos que representan estos sistemas serán modelos continuos. Cuando se modela matemáticamente al sistema, las variables del modelo que representan los atributos de las entidades del sistema se controlan mediante funciones continuas. Por lo general, para describir la manera en que están relacionados los atributos del sistema, los modelos se representan en forma de ecuaciones algebraicas. En ocasiones las relaciones describen las tasas de cambio de los atributos, de manera que el modelo consiste de ecuaciones diferenciales, cabe notar que existen sistemas continuos que no son de ecuaciones diferenciales, y que son de una estructura más sencilla; estos últimos son los que servirán de base para los propósitos de esta tesis.

Un sistema discreto es aquel en que las actividades predominantes dentro del sistema provocan cambios predominantemente discontinuos.

Otra clasificación de los modelos se apoya en la aleatoriedad de sus componentes dando origen a la clasificación en modelos deterministas y modelos estocásticos.

Los modelos de un sistema se pueden clasificar también en

modelos físicos y modelos simbólicos, un ejemplo de modelos físicos son las reproducciones a escala de autos, casas, etc., y como ejemplo de modelos simbólicos son las ecuaciones, los números, etc., el objetivo de estos modelos consiste en representar la realidad. Existen también los modelos icónicos los cuales son una representación a escala de la realidad.

Los modelos simbólicos se pueden clasificar a su vez en modelos estáticos y dinámicos, esta clasificación se apoya en los cambios que sufre el sistema con el tiempo. Dichos modelos a su vez se pueden dividir de acuerdo a la técnica que se emplee para resolver el modelo en modelos analógicos y los modelos matemáticos [3].

En la figura 2.1 se ilustran las clasificaciones de los modelos descritos:

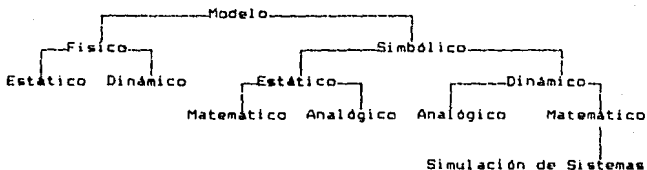


Fig. 2.1. Clasificación de Modelos.

La simulación de sistemas es una técnica numérica que se utiliza con los modelos simbólicos dinámicos.

### 2.9. Modelo matemático.

Dentro de los modelos simbólicos se tiene los modelos matemáticos que son una representación o abstracción de una



situación u objeto real, que muestra las relaciones (directas e indirectas) y las interrelaciones de la acción y la reacción en términos de causa y efecto. Como un modelo es una abstracción de la realidad, puede parecer menos complicado que la misma. Para que sea completo el modelo debe ser representativo de aquellos aspectos de la realidad que están investigándose.

Un modelo matemático usado comunmente es una ecuación. Sus símbolos son mucho más fáciles de manipular que las palabras, se escriben más rápidamente y permiten su manipulación en computadora.

Para clasificar a los modelos de simulación, se ha sugerido un cierto número de sistemas taxonómicos diferentes, sin embargo para el interés de esta tesis solo nos interesan los modelos matemáticos, los cuales se clasifican como deterministas o estocásticos, y como estáticos o dinámicos. De manera particular, haremos referencia a los modelos de tipo determinista [3].

### 2.9. Modelos deterministas.

Un modelo determinista es aquel que es descrito de manera certera y en el cual no intervienen factores de tipo aleatorio, por tanto, ni a las variables exógenas ni a las endógenas, se les permite ser variables al azar; en tanto que se suponen relaciones exactas para las características de operación en lugar de funciones de densidad de probabilidad [3].

### 2.10. Simulación de sistemas.

Aún cuando la definición de simulación dada por Shubik no es una definición suficientemente precisa, consideramos que es la

más apropiada para los propósitos de esta tesis.

Simulación de un sistema (o un organismo) es la operación de un modelo (simulador), el cual es una representación del sistema. Este modelo puede sujetarse a manipulaciones que serían imposibles de realizar, demasiado costosas o imprácticas si se realizarán en el sistema mismo. La operación de un modelo puede estudiarse y con ello, inferirse las propiedades concernientes al comportamiento del sistema o subsistema real [1].

## 2.11. Metodología para desarrollar un modelo de simulación.

La metodología básica utilizada para el desarrollo de un modelo de simulación es descrita en los conceptos fundamentales que a continuación se describen.

### 2.11.1. Diseño de modelos matemáticos.

Los modelos matemáticos constan de cuatro elementos básicos bien definidos que proporcionan los recursos e instrumentos intelectuales con los que se ha de trabajar para construir el sistema teórico o modelo del sistema, a saber: los componentes, las variables, los parámetros y las relaciones funcionales.

Los componentes tienen la tendencia a variar ampliamente, dependiendo de las características del sistema simulado.

Las variables que aparecen en dichos modelos se emplean para relacionar un componente con otro y se clasifican a su vez, como variables exógenas, variables de estado y variables endógenas.

Las variables exógenas son las independientes o de entrada del modelo y las cuales se supone que han sido predeterminadas y proporcionadas independientemente del sistema que se modela.

Puede considerarse que estas variables actúan sobre el sistema, pero no reciben acción alguna por parte de él. Es posible clasificar a las variables exógenas en controlables y no controlables. Las primeras (o instrumentales) son aquellas variables o parámetros susceptibles de manipulación o control por quienes toman decisiones o crean políticas para el sistema.

Las variables de estado, describen el estado de un sistema o uno de sus componentes, ya sea al comienzo, al final o durante un período (de tiempo). Estas variables interaccionan con las variables exógenas del sistema y con las endógenas, de acuerdo a las relaciones funcionales supuestas para el sistema. El valor de una variable de estado, durante un período particular de tiempo, puede depender no solamente de los valores de una o más variables exógenas en algún período precedente, sino también del valor de ciertas variables de salida en períodos anteriores. Cuando un componente toma su entrada de una porción de su propia salida en un período previo, se dice que ocurre una retroalimentación. Al introducir los retardos en los ciclos de retroalimentación, los valores de las variables de salida del sistema dependen de los parámetros de las ecuaciones de funcionamiento y es posible que exhiban una tendencia hacia las oscilaciones, ya sean amortiguadas o explosivas.

Las variables endógenas son las dependientes o de salida del sistema y son generadas por la interacción de las variables exógenas con las de estado, de acuerdo con las características de operación del último.

Finalmente el hecho de que una variable en particular esté clasificada como exógena, de estado o endógena, depende del

propósito de la investigación [1].

### 2.12. Planeación de los experimentos de simulación.

En esta sección se engloba el ciclo o las etapas de la investigación que comienza con el planteamiento de un problema, la formulación de una hipótesis y, finalmente, la interpretación de resultados y las conclusiones. La experiencia sugiere que la planeación de experimentos de simulación requiera de un procedimiento que conste de los pasos siguientes [1]:

Paso 1. La formulación de un problema.

Paso 2. La recolección y procesamiento de datos formados de la realidad.

Paso 3. La formulación de un modelo matemático.

Paso 4. La estimación de los parámetros de las características operacionales a partir de los datos reales.

Paso 5. La evaluación del modelo y de los parámetros estimados.

Paso 6. La formulación de un programa para la computadora.

Paso 7. La validación.

Paso 8. El diseño de los experimentos de simulación.

Paso 9. El análisis de los datos simulados.

Los pasos anteriores señalados se ilustran a continuación en la figura 2.2:

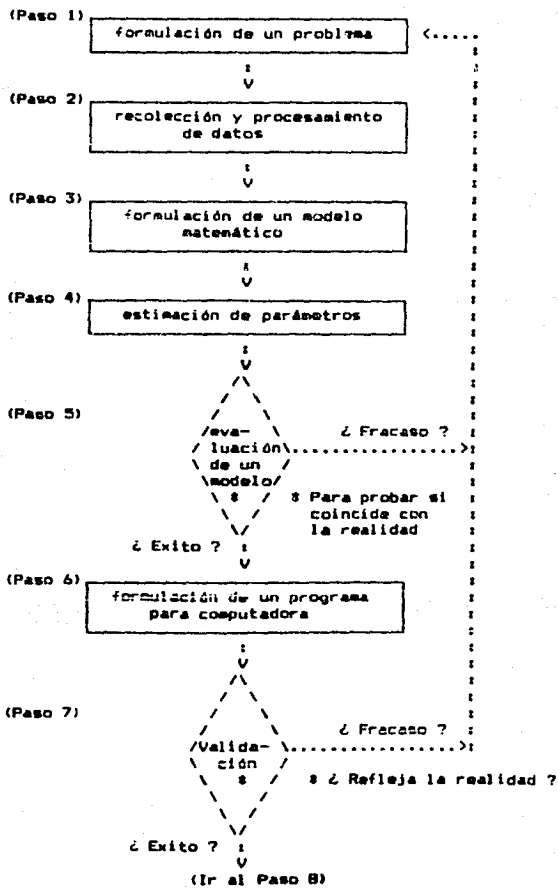


Fig.2.2 Diagrama de flujo para planear experimentos de simulación

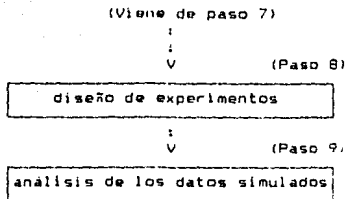


Fig.2.2 (Continuación)

2.12.1. Formulación de un problema (Paso 1).

En la realización de un estudio de simulación, es necesario, en primer lugar, definir claramente los objetivos de la investigación, antes de hacer cualquier intento encaminado a planear la realización de un experimento en simulación. Con toda seguridad, se encontrará que la exposición original del problema varía considerablemente de su versión final, ya que la formulación del problema es un proceso secuencial que generalmente requiere una reformulación continua y progresiva y un refinamiento de los objetivos del experimento durante su realización.

Si el objetivo del estudio de simulación es el de dar respuesta a una o más preguntas, es necesario que se intente plantear éstas detalladamente desde el comienzo del experimento, aún cuando sea posible refinar las preguntas en el curso del experimento. A menos que se especifique lo que se entiende por una respuesta adecuada a una pregunta, no deberá esperarse resultados significativos de la simulación en computadora.

Por otro lado, el objetivo de la investigación podría consistir en probar una o más hipótesis relativas al

comportamiento del sistema bajo estudio.

Por consiguiente, deben tomarse dos decisiones importantes antes de comenzar a trabajar con cualquier experimento de simulación. En primer término, hay que decidir los objetivos de la investigación y en segundo lugar, es necesario decidir el conjunto de criterios para evaluar el grado de satisfacción al que deba sujetarse el experimento a fin de que cumpla los objetivos. De hecho, existe la posibilidad de que después de tomar estas dos decisiones se tenga que rechazar completamente la simulación en computadora, ya sea por el costo excesivo, la complejidad o la inhabilidad de satisfacer los objetivos planteados.

#### 2.12.2. Recolección y procesamiento de datos (Paso 2).

Se cree que es imposible formular un problema o conjunto de objetivos para un experimento, sin tener acceso adecuado a la información (cuantitativa en este caso) acerca del sistema que se investiga. Para nuestros propósitos, resulta completamente irrelevante que los requerimientos para el procesamiento de datos precedan a la formulación del problema o viceversa; si se han de dirigir experimentos de simulación, es importante que ambas funciones se lleven a cabo; en todos los casos se partirá de los datos generales aceptados como válidos, de los cuales, por medio del razonamiento lógico, podrán deducirse varias suposiciones.

En primera instancia los datos referentes al sistema que se va a investigar, constituye un requisito previo a la formulación del problema. Ya que si no se cuenta con datos del sistema que se debiera simular, es imposible probar la validez del modelo

construido. Esta información se puede utilizar en diferentes etapas de la construcción del modelo. Los datos reducidos a una forma significativa, pueden sugerir hipótesis que se usarán en la formulación de los modelos matemáticos que describan el comportamiento de un sistema dado. Los datos también pueden sugerir mejoras o refinamientos en los modelos matemáticos que existen en el sistema por simularse. Los datos para estimar los parámetros de las características de operación relativas a las variables endógenas, exógenas y de estado del sistema.

Es posible identificar seis fases importantes para el procesamiento de datos, y que es necesario realizar al ejecutar experimentos de simulación en computadoras, y son: recolección, almacenamiento, conversión, transmisión, manipulación y salida.

La recolección de datos es el proceso de captación de los hechos disponibles, con lo cual éstos pueden ser procesados posteriormente, cuando sea necesario. En realidad, el proceso de recolección y el de almacenamiento de datos ocurren simultáneamente, pues el primero implica que los datos sean o hayan sido almacenados. A menudo se ha demostrado que la tarea de recolectar y almacenar los datos es sumamente costosa y laboriosa, pues comprende la identificación, revisión, edición, codificación (asignación de claves), transcripción y verificación de ellos.

La manera en la cual los datos se almacenan durante la primera etapa del procesamiento, no constituye, por lo general la forma más eficiente que se debe emplear en las etapas



posteriores; por esta razón, la conversión de los datos de una forma a otra tiene una función crucial en la determinación de la eficiencia del procesamiento. Bajo ciertas circunstancias, existen problemas adicionales en la conversión de los datos de una forma a la otra, que implican una transmisión de ellos, esto es, el transporte de la información desde una localidad hasta el lugar en donde será procesada.

Una vez que los datos han sido recolectados, almacenados, convertidos a una forma eficaz, y transmitidos al lugar de procesamiento final, se puede comenzar con las operaciones de manipulación de datos y la preparación de estos para su salida final. Las etapas de manipulación requieren de la realización de operaciones como son las de clasificar, cotejar, intercalar, recuperar información y otras, como las operaciones aritméticas y lógicas. Estas operaciones se realizan con una computadora o sin ella; el uso o no de computadoras, depende de hecho, tanto de la cantidad de datos por manipular, como de la utilización que finalmente tengan.

### 2.12.3. Formulación de los modelos matemáticos (Paso 3).

La formulación de modelos matemáticos consta de tres pasos:

- i. Especificación de los componentes.
- ii. Especificación de las variables y de los parámetros, y
- iii. Especificación de las relaciones funcionales.

Una de las primeras consideraciones que se toman en cuenta en la formulación de un modelo matemático reside en saber cuántas variables se deben incluir en el modelo. En lo referente a las variables endógenas o de salida del modelo, se tiene poca

dificultad en la elección de cuales son, debido por lo general, a que estas variables se determinan al comenzar el experimento, cuando formulamos los objetivos del estudio. Obviamente, habrá un límite superior en el número de las variables endógenas posibles de investigar en un solo experimento de simulación ya que el tamaño de la computadora para el investigador, impondrá necesariamente ciertas limitaciones relativas a este aspecto. Sin embargo, la dificultad real surge en la elección de las variables exógenas que afectan a las variables endógenas. La existencia de muy pocas variables exógenas pueden llevar a modelos inválidos, en tanto que una abundancia de ellas hacen imposible la simulación en la computadora debido a la insuficiencia en la capacidad de su memoria o bien complicar de modo innecesario, los programas de computación.

La segunda consideración importante en la formulación de modelos matemáticos se refiere a la complejidad de los mismos. Por lo general estamos interesados en la formulación de modelos matemáticos que produzcan descripciones o predicciones, razonablemente exactas, referentes al comportamiento de un sistema dado y reduzcan a la vez, el tiempo de computación y programación. No es posible acentuar la completa interdependencia de estas características en los modelos matemáticos, ya que tanto el número de variables en un modelo, como su complejidad, se encuentran directamente relacionadas con los tiempos de programación, cómputo y validez. Si alteramos cualquiera de las citadas características, alteraremos a su vez el resto de ellas.

Una tercera consideración en la formulación de modelos matemáticos para simulación en computadora estriba en el área de

la eficiencia de computación. Entendemos por ello, la cantidad de tiempo de cómputo requerida para lograr algún objetivo experimental específico. Como regla general, estamos comúnmente interesados en uno de los dos objetivos siguientes relacionados con la eficiencia de los experimentos de simulación: en el primer caso, es posible que deseemos reducir el tiempo de cómputo requerido para generar los valores de nuestras variables endógenas sobre un periodo específico. En el segundo caso, podríamos interesarnos en la reducción de tiempo de computación requerido para lograr los valores de ciertos parámetros estadísticos generados por nuestro modelo de simulación.

El tiempo consumido en la programación de la computadora, constituye una cuarta consideración al formular modelos matemáticos para simulación. El tiempo requerido para escribir un programa que genere los tiempos planificados para las variables endógenas de un conjunto particular de modelos matemáticos, depende en parte de el número de variables utilizadas en los modelos y de su complejidad.

La quinta área de interés en la construcción de modelos es la validez o la cantidad de realismo incorporado en ellos.

La sexta y última consideración, al formular modelos para simulación en computadora consiste en su compatibilidad con el tipo de experimentos que se van a realizar con ellos. Ya que nuestro objetivo principal al formular un modelo matemático, es el de permitirnos dirigir experimentos de simulación, deberá pensarse qué forma particular y que características tendrán los diseños de experimentos que se utilizaron en el modelo.

Existen ciertas dificultades potenciales que afrontará el constructor de modelos para describir el comportamiento de los modelos reales mediante la técnica de modelos matemáticos. Primero, quizá sea imposible cuantificar o medir ciertos tipos de variables que afectan el comportamiento del sistema. Segundo, el número de variables por considerarse al describir un sistema dado, posiblemente exceda la capacidad de la computadora disponible. Tercero, podemos desconocer algunas de las variables exógenas significativas que afectan la salida del sistema. Cuarto, es factible que además de desconocer, sea imposible obtener alguna de las relaciones entre las variables exógenas y las endógenas de nuestro sistema. Quinto, las relaciones entre las variables que afectan el comportamiento del sistema, son en muchos casos tan complejas que no pueden expresarse como una o más ecuaciones matemáticas.

#### 2.12.4. Estimación de los parámetros de las características operacionales a partir de los datos reales (Paso 4).

Una vez que se han formulado un número de modelos matemáticos describiendo el comportamiento del sistema y se han colectado los datos necesarios para la validación del mismo y para la estimación de parámetros, se puede proceder a estimar los valores de éstos para los diferentes modelos involucrados y probar la significancia estadística de estas estimaciones.

A manera de ejemplo particular, y para mostrar la problemática que puede involucrar la estimación de los valores de los parámetros, haremos referencia a los modelos económicos.

Como La estimación de los parámetros de modelos económicos

involucra el conocimiento de modelos lineales y en particular de la Econometría en el párrafo siguiente se definen ciertas herramientas estadísticas.

La Econometría puede ser definida como una ciencia social en la cual las herramientas de la teoría económica, matemáticas, e inferencia estadística son aplicadas al análisis de fenómenos económicos. Su principal objetivo es dar contenido empírico a la teoría económica, da una formulación matemática de la teoría económica, desarrolla técnicas apropiadas de inferencia estadística, y por último hay inferencia estadística actual de datos económicos.

Con muy poca preparación estadística se podrá estimar los parámetros de operabilidad de las características de un sistema económico, pero en algunos casos se deberá poseer más dominio de la metodología econométrica según aumente el grado de dificultad en el sistema económico.

J. Jhonston considera que la teoría económica involucra a tres pasos básicos:

El primer paso en la medida de las relaciones económicas es la especificación de las variables que intervienen en dichas relaciones. Para presentarlo de la manera más simple considérese el caso más elemental en el cual se considera una relación simple, la cual contiene solamente dos variables. Denotando las variables por Y y X, se puede postular que

$$Y = f(X)$$

Este primer paso identifica la variable X, la cual se considera que influencia a la variable Y. El segundo paso consiste en especificar la forma de la relación que hay entre Y y X. El

tercer paso consiste en la introducción de un término estocástico en las relaciones económicas, el cual produce "perturbaciones" aleatorias. Jhonston considera al comportamiento humano en sus hábitos de consumo como un ejemplo de esto. Agrega, por otra parte, que otras fuentes de error están constituidas por los errores de observación o de medida.

Tomando en cuenta estos pasos, se busca definir un modelo para la generación de observaciones de tipo económico, y con ellas definir una población estadística, así, una cierta cantidad de observaciones de tipo económico pueden ser estudiadas como una muestra proveniente de una población.

Una vez que se tiene una especificación de una población básica, se puede utilizar en los principios y en los criterios de la inferencia estadística, para desarrollar un método racional para medir las relaciones de la teoría económica, a través de una muestra de observaciones. En muchos casos se puede utilizar también algún conocimiento teórico o empírico previo que se tenga acerca del valor de los parámetros de la población. Tal información a priori, es un rasgo distintivo o característico de la teoría econométrica [5].

Entre los más importantes métodos econométricos de estimación descritos por Goldberger y Johnston y en base a sus propiedades estadísticas y computacionales se tienen los siguientes:

1. Métodos de una sola ecuación

- a. Mínimos cuadrados ordinarios
- b. Mínimos cuadrados indirectos

- c. Información limitada y ecuación única
  - d. Mínimos cuadrados en dos etapas.
2. Métodos de ecuaciones simultaneas
- a. Información completa, máxima verosimilitud
  - b. Mínimos cuadrados en tres etapas.

#### 2.12.5. Evaluación del modelo y de los parámetros estimados (Paso 5).

Una vez que se han formulado un grupo de modelos matemáticos que describen el comportamiento del sistema, y se han estimado los parámetros operacionales de las características apoyados en las observaciones tomadas de la población del mundo real, se debe entonces hacer una valoración inicial de la adecuación del modelo con la realidad. Esto es, se debe probar el modelo.

Es claro que poco se ganaría si se usa un modelo inadecuado para llevar a cabo los experimentos de simulación en una computadora, ya que en este caso se estaría realmente "simulando la propia ignorancia".

Esta etapa en el procedimiento de simulación, solamente representa la primer etapa de la prueba de un modelo de simulación, previa a las corridas efectivas de simulación. Y se busca básicamente probar la validez de los supuestos que fueron realizados en el modelo. En el paso 7 de la pagina 16 del procedimiento de simulación mencionado previamente se tratará lo concerniente a la validación de los datos de salida generados por el modelo computarizado de simulación y entonces se estará interesado en comparar los datos simulados contra los datos históricos, para determinar la capacidad predictiva del modelo.

En el caso de características operacionales que tienen la forma de distribuciones probabilísticas, se desea aplicar pruebas de "bondad de propiedad o ajuste" para determinar que tan bien representa una distribución probabilística hipotética al mundo real del cual fueron generados los datos. Se deseará también probar la significación estadística de las estimaciones de los valores esperados medias, varianzas, y otros parámetros de estas distribuciones probabilísticas.

Las siguientes pruebas pueden ser incluidas:

1. Pruebas concernientes con promedios
  - a. Prueba de media para una muestra
  - b. Diferencia entre muestras.
2. Pruebas concernientes a varianzas
  - a. Prueba de Chi cuadrada
  - b. Prueba de F.
3. Pruebas basadas en recuento de datos
  - a. Pruebas concernientes a proporciones
  - b. Diferencias entre K proporciones
  - c. Tablas de contingencia
  - d. Pruebas de bondad de ajuste.
4. Pruebas no paramétricas
  - a. Prueba del signo
  - b. Prueba basada en la suma de rangos
  - c. Prueba de la mediana
  - d. Prueba U
  - e. Prueba de clases
  - f. prueba de correlaciones seriadas.



En el caso de características operacionales que en el caso de modelos económicos tendrán la forma de modelos econométricos se deseará probar la significancia estadística de cada uno de los parámetros estimados de ese modelo, a través de pruebas estandar de  $t$  y  $F$ . También se deseará aplicar pruebas que nos permitieran detectar alguna violación en las asunciones bajo las cuales fue planteado el modelo econométrico, estas pueden incluir pruebas para:

1. Errores en las variables
2. Multicolinealidad
3. Heterosedasticidad
4. Autocorrelación
5. Identificación

Entre las cosas que se deberá cuestionar a este nivel de desarrollo del modelo de simulación, están las siguientes:

1. ¿ Se han incluido alguna(s) variable(s) que no sea(n) necesarias(s) en el sentido de que contribuyen muy poco a la capacidad para predecir el comportamiento de variables endógenas en el sistema ?
2. ¿ Se han dejado de incluir una o más variables exógenas que están probablemente afectando el comportamiento de variables endógenas a en el sistema ?
3. ¿ Se han formulado de manera imprecisa una o más de las relaciones entre las variables endógenas y exógenas del sistema ?
4. ¿ Han sido estimados con propiedad los parámetros de operabilidad de las características del sistema ?
5. ¿ Son las estimaciones de los parámetros en el modelo estadísticamente significativos ?

6. ¿ Con base en los cálculos manuales (puesto que aún no se ha formulado un programa computacional), ¿ como son los valores teóricos de las variables endógenas del sistema comparadas con los valores históricos o los valores actuales de las variables endógenas reales ?

Si y solo si, se puede dar respuestas satisfactoria a las seis preguntas formuladas, se deberá proceder al paso 6 (pag. 16) para formular el programa computacional. De otra forma se deberá repetir los pasos (pag. 16) del 1 al 5 tanto como sea necesario, hasta que se logre tener respuestas satisfactorias a las seis preguntas antes formuladas.

Como se ha podido notar, es necesario comprender y manejar herramientas científicas y metodológicas para poder desarrollar un modelo en particular. En el caso de la simple representación de una función determinista, es necesario comprender el comportamiento de ésta, es precisamente aquí, donde el sistema servirá de herramienta auxiliar al interesado en la comprensión de dicho proceso.

#### 2.12.6. Formulación de un programa computacional (Paso 6).

La formulación de un programa computacional para los propósitos de la conducción de experimentos de simulación por computadora con el modelo del sistema bajo estudio, requieren la realización de las seis actividades básicas siguientes:

1. Diagramas de flujo
2. Códigos de computadora ya sea utilizando lenguajes de propósitos generales o lenguajes especiales para propósitos de simulación

3. Verificación de errores
4. Información de entrada y condiciones de arranque
5. Generación de información
6. Reportes de salida

El primer paso en la escritura o realización de un programa de simulación por computadora incluye la formulación de un diagrama de flujo que indique esquemáticamente la secuencia lógica de los eventos que van a ser realizados por la computadora a través del tiempo y el flujo y transformación de las variables endógenas del modelo. La importancia del diagrama de flujo en la realización o escritura de un programa de computadora no debe ser sobreenfatizado.

En la realización de un diagrama de flujo de la lógica de un experimento de simulación, se puede utilizar código de computación el cual depende del lenguaje que se use. En general existen dos alternativas disponibles: 1) se puede utilizar en un lenguaje de propósito general como sería: FORTRAN, ALGOL, COBOL, PL/I ó PASCAL entre otros; 2) se puede usar uno de los lenguajes especiales para propósitos de simulación tales como el GPSS, CSMP, SIMSCRIPT, GASP, SIMPAC, DYNAMO, PROGRAM SIMULATE, SISI, etc. La ventaja principal de usar un lenguaje especialmente hecho para propósitos de simulación en lugar de un compilador de propósitos generales, es que éste requiere de menos tiempo de programación. Estos lenguajes han sido escritos para facilitar la programación de ciertos tipos de sistemas. Así para simular sistemas económicos de larga escala que han sido diseñados como modelos econométricos integrados por gran cantidad de ecuaciones (hasta 200) se ha utilizado CSMP, GPSS. SIMSCRIPT, y GASP son

particularmente adecuados para encuestas y líneas o colas de espera. Aún cuando se puede podamos reducir el tiempo de programación al usar un lenguaje de simulación, usualmente se tiene que pagar un precio por este beneficio, en términos de reducción de flexibilidad en los modelos e incremento en tiempo de corridas de computadora. Otra ventaja especial de los lenguajes de simulación es que usualmente proveen técnicas de verificación de errores que son superiores a aquellas que están disponibles en FORTRAN, ALGOL, etc.

Otro aspecto dentro de la fase de la programación del modelo es el correspondiente a la entrada de información y a las condiciones de arranque para los experimentos de simulación. Dado que los experimentos de simulación son por su propia naturaleza dinámicos, una cuestión sale a cuentas: que valores se deberá asignar a las variables del modelo en el tiempo en el que se inicia la simulación. ¿ Se debe hacer una pausa dentro del sistema en algún punto particular en el tiempo ? ¿ Cuando así se haga, que suposiciones se deberán establecer con respecto de las condiciones de equilibrio ó de las condiciones de arranque del sistema ? Inútil es decir, que estas cuestiones no son fácilmente resueltas para muchos sistemas, y que el investigador generalmente debe recurrir al método de ensayo y error para determinar una serie de valores iniciales para el sistema que no lleguen a distorsionar los resultados de los periodos de tiempo posteriores.

Un problema directamente relacionado con la escritura de programas de simulación por computadora es el desarrollo de

técnicas numéricas para la generación de datos. Los datos usados en los experimentos de simulación por computadora pueden ser, leídos por la computadora a través de fuentes externas como discos y cintas magnéticas, o bien pueden ser generadas internamente en la computadora por medio de rutinas especiales.

Una consideración final en el desarrollo del programa de computación para un experimento de simulación, es que clase de reportes de salidas se necesitan para dar la información requerida acerca del comportamiento del sistema de simulación. Si se utiliza un lenguaje de computación de propósitos generales como es el PASCAL, entonces habrá un mínimo de restricciones impuestas en el formato de los reportes de salida. Sin embargo, si se usa un lenguaje especial para propósitos de simulación tal como SIMSCRIPT, entonces se tendrá que limitar las salidas a los formatos proporcionados por el lenguaje.

#### 2.12.7. Validación (Paso 7).

El problema de validar modelos de simulación por computadora, es en realidad difícil debido a que éste involucra a un sinnúmero de complejidades prácticas, teóricas, estadísticas y aún filosóficas. La validación de experimentos de simulación es simplemente parte de un problema más general conocido como validación de modelos ó hipótesis. La pregunta básica es "¿ Qué significa validar una hipótesis ?" y "¿ Qué criterio debería ser usado para establecer la validez de una hipótesis ?".

En general, dos pruebas parecen ser apropiadas para la validación de modelos de simulación. Primero ¿qué tan buenos son los valores simulados de las variables endógenas, comparados con

los datos históricos conocidos?, esto si los datos históricos estan disponibles. Segundo ¿qué tan exactos son los modelos de simulación en la predicción del comportamiento del sistema real en los periodos futuros de tiempo? Asociado a cada una de estas dos pruebas, existen una variedad de clásicas y aún más recientes y ventajosas pruebas estadísticas, las cuales no se señalaran en este trabajo debido a que no son necesarios estos conocimientos para un estudio del tipo y de la magnitud que pretende abarcar esta tesis .

#### 2.12.8. Diseño de experimentos de simulación (Paso B).

Una vez que se está razonablemente satisfechos con la validación del modelo computacional al modelo de simulación se podrá iniciar el diseño de los experimentos de simulación, definiendo los factores (variables exógenas y parámetros), que se controlarán y las variables endógenas que serán respuestas.

En esta fase se debe primero seleccionar los niveles y combinaciones de niveles de los factores que se usarán y el orden o arreglo de la experimentación. Segundo, habiendo seleccionado la combinación de factores se debe establecer un mecanismo que permitan que los resultados se obtengan libres de errores de cualquier tipo.

Mucho ha sido escrito en la literatura de los diseños experimentales acerca del primer objetivo, una gran parte de lo cual es relevante al diseño de experimentos de simulación. Mucho ha sido también escrito en la literatura acerca del segundo objetivo, pero debido a que algunos aspectos característicos de los experimentos de simulación por computadora se diferencian de

los del mundo real de la experimentación, del material existente es mucho menos lo que es aplicable o relevante al segundo objetivo. Algún material relevante al segundo objetivo o meta esta contenido en las técnicas de Montecarlo.

Deberá enfatizarse, sin embargo, que la aplicación de la técnica de los diseños experimentales al diseño de experimentos de simulación por computadora permanece dentro de un área de la investigación donde hay en realidad una gran cantidad de trabajo por realizar.

#### 2.12.9. Análisis de los datos simulados (Paso 9).

El paso final es el procedimiento conocido como análisis de datos generados a través de un modelo computacional de un sistema que ha sido simulado. El análisis de los datos simulados consiste de los siguientes pasos:

- i. Colección y procesamiento de datos simulados
- ii. Computación de pruebas estadísticas
- iii. Interpretación de resultados

Aunque el análisis de datos simulados es en verdad similar al análisis de datos del mundo real, hay algunas diferencias importantes. Comparativamente, la simulación computarizada es similar a la técnica estadística conocida como de distribuciones de muestreo, las cuales han sido usadas en el campo de la estadística por muchos años. Telchroew sin embargo, ha señalado, acerca del análisis de datos de la simulación por computadora, que éste es mucho más difícil que el análisis de datos del mundo real.

En las distribuciones de muestreo, la manera en la cual la

aleatoriedad interviene es bien entendida y pueden ser explícitamente establecida. En experimentos de simulación dicha intervención es de forma sumamente complicada y usualmente las relaciones no pueden ser establecidas explícitamente, sino a través del algoritmo por el cual los valores numéricos son computados.

Otra complicación mayor es el hecho de que, en tanto la distribución de muestreo usualmente considera modelos estáticos, la simulación por su propia naturaleza, tiene que ver con modelos dinámicos. Las salidas son un grupo de series relacionadas con el tiempo. El análisis de estos, por lo tanto, es mucho más difícil que el análisis de un grupo de números que sean considerados como una muestra proveniente de una distribución dada.

Otro factor es el gran número de parámetros que intervienen en el estudio de simulación. Las distribuciones de muestreo usualmente tratan cuando más con unos cuantos parámetros; los experimentos de simulación frecuentemente tratan con miles de ellos. En cualquier caso particular, es casi imposible determinar un conjunto mínimo suficiente de parámetros importantes en base a los cuales los experimentos diseñados pueden ser corridos en la computadora.

Continuando con el tema del análisis de datos simulados, una vez construido el modelo de simulación, este es activado utilizando datos conocidos para así simular la operación del sistema que representa y verificar si realmente se ha creado el modelo correcto, si no es así, se deberá revisar éste, incluir nuevas variables, quitar otras, definir nuevas relaciones o



modificar las existentes para crear un nuevo modelo que nuevamente será activado y verificado, de esta manera se irá depurando el modelo hasta llegar a una representación satisfactoria. Posteriormente se utilizarán datos generados, repitiendo la operación del sistema bajo varias alternativas y comparando su ejecución, e identificando las estrategias más productivas. Debe indicarse que, a este respecto, la computación ha permitido que la simulación explique una gran variedad de problemas, debido a que en su mayoría se requiere de una gran cantidad de cálculos que sin una computadora no podrían realizarse.

### 2.13. Aspectos para la construcción de un modelo de simulación.

Los aspectos que se deben considerar para construir un modelo de simulación de un sistema, son los siguientes:

- a) La formación en bloques. La descripción del sistema se debe organizar en una serie de bloques, o subsistemas. El propósito de formar los bloques es simplificar la especificación de las interacciones dentro del sistema. Cada bloque describe parte del sistema, depende de pocas variables de entrada y produce unas pocas variables de salida. Posteriormente, puede describirse al sistema como un todo en términos de las interconexiones entre los bloques. En forma correspondiente, se puede representar gráficamente al sistema como un diagrama simple de bloques.
- b) La relevancia. El modelo sólo debe de incluir los aspectos del sistema relevantes a los objetivos del estudio. Aunque la información irrelevante en el modelo no lo perjudica, se debe

excluir debido a que aumenta la complejidad del modelo y genera más trabajo en la solución del modelo.

c) La exactitud. Debe de tenerse en cuenta la exactitud de la información que se recabe.

d) La agregación es un factor adicional que debe considerarse, esto es, el grado con que pueden agruparse las distintas entidades individuales en entidades más grandes.

#### 2.14 Importancia de la graficación en la simulación.

Cabe resaltar que para poder analizar un conjunto de variables, en ocasiones se torna complicado debido a que es difícil apreciar el comportamiento de dichas variables a través de su representación numérica. Es entonces cuando la técnica de la graficación entra a resolver el problema de la visualización clara del comportamiento de fenómenos a través de sus variables, la graficación permite al investigador observar de manera más clara el comportamiento, de los fenómenos presentados mediante la simulación. En el siguiente capítulo se analizarán algunas de las técnicas que serán de utilidad para la implementación de la parte gráfica del sistema de simulación.

### III. LA GRAFICACION POR COMPUTADORA.

La graficación por computadora se refiere a todos los procesos que involucren la entrada, construcción, almacenamiento, recuperación, manipulación, alteración y análisis de información que conduzcan a un despliegue en forma gráfica o pictórica. Los usos de estas formas de manejo de información han tenido aplicaciones muy diversas desde el inicio, siendo quizá las más conocidas las de los juegos y las de generación de gráficas estadísticas [6]. Una aplicación específica es la orientada hacia la simulación, ayudando al responsable del análisis de la información que se genere a percibir de manera más clara el comportamiento de dicha información.

El manejo de información para despliegue gráfico puede realizarse fuera de línea donde la computadora principal (UCP) no interviene al momento de hacerse el proceso gráfico, o en línea.

Otra forma más significativa para clasificar el proceso gráfico, es considerando las diferencias entre operación pasiva y activa (o interactiva) de un sistema de programas. En los sistemas de graficación pasivos (o graficación pasiva) el usuario u observador no ejerce ningún control sobre el objeto que se genera; una vez que éste se encuentra ya listo para su despliegue, el usuario genera una gráfica y la envía al dispositivo que la construirá, ya sea una impresora, un graficador o una pantalla de video [6].

En la graficación interactiva existe comunicación en ambos sentidos entre la computadora y el usuario. La computadora recibe, a través del software de graficación, información sobre cómo debe modificarse el desplegado. En este caso se establece un

diálogo computadora humano en tiempo real auxiliado por la entrada manual de información. El caso de las interacciones manejadas al gusto del usuario como es el caso de rotación y traslación de una gráfica visualizada en pantalla es un caso de graficación interactiva [6].

Los sistemas de graficación pasivos se utilizan principalmente para salidas gráficas a papel, microfilm, etc. Entre otras aplicaciones de la graficación pasiva se cuentan el análisis de escenas, el reconocimiento de patrones y la animación computarizada [6].

La graficación interactiva se ha usado en sistemas avanzados como son los de entrenamiento de pilotos, diseño ingenieril combinados con sistemas de análisis de componentes estructurales de edificios, autos, barcos, aviones y otros, así como en los sistemas de enseñanza de tópicos específicos por computadora, manipulación de diagramas de fórmulas químicas de moléculas, superficies, sólidos y otros [8].

Por razones históricas principalmente del equipo, han sido más populares los despliegues de "esqueletos" bidimensionales y tridimensionales que las representaciones de sólidos a través de superficies.

La graficación interactiva es efectiva en muchas aplicaciones, sobre todo cuando pueden construirse sistemas de software que respondan rápida y efectivamente a las solicitudes del usuario. Entre las aplicaciones de mayor interés se cuentan la generación de mapas climáticos, de contornos para exploración y explotación de yacimientos de petróleo, minería y otros, la

cartografía en general, las redes de PERT, telefonía, y la animación computarizada en ciencia y educación.

Los programas de aplicaciones que hacen uso de graficación suelen llamarse programas de aplicaciones o sistemas de graficación. es necesario aclarar que se trata de sistemas de software con un uso específico que incluyen:

- programas que generan salida gráfica,
- manejadores de interrupciones gráficas.
- rutinas de análisis de la aplicación específica,
- rutinas de acceso y manejo de la base de datos de la aplicación,
- rutinas de acceso y manejo de la base de datos de información gráfica ( programas manejadores de las estructuras de datos, y estructuras de datos gráficos ) [7].

### 3.1 Módulos Funcionales de un Sistema de Graficación.

- A - Módulo de enlace/comunicación con el usuario. Presenta al sistema y sus facilidades, así como los resultados de las facilidades solicitadas y/o mensajes de error con posibles sugerencias de recuperación.
- B - Módulo de análisis de las entradas del usuario. Análisis sintáctico del lenguaje de comunicación o enlace con el usuario.
- C - Módulo de interpretación de entrada del usuario. Análisis semántico de las solicitudes que se ajustan a las reglas del lenguaje.
- D - Módulo de ejecución de entrada del usuario. Activa los submódulos del sistema apropiados para la realización de la instrucción del usuario y les pasa los parámetros

pertinentes.

- E - Módulo de creación y manejo de estructuras de datos. Construye y mantiene las estructuras de datos necesarias y administra la memoria.
- F - Módulo generador de la salida gráfica. Este puede contener un submódulo encargado del manejo exclusivo del archivo de despliegue a través de las rutinas básicas existentes o de otras creadas como parte del sistema.
- G - Módulo manejador de errores.
- H - Módulo programa Monitor o Maestro. Se encargará del control de las acciones entre todos los módulos si se elige una política de control centralizado.

La interacción entre los módulos puede visualizarse gráficamente mediante el diagrama mostrado en la figura 3.1.

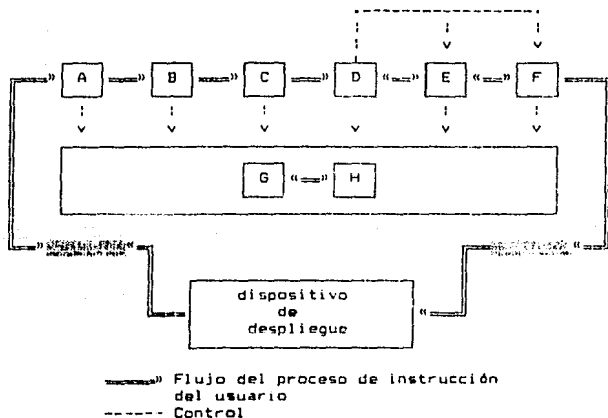


Fig. 3.1. Diagrama de interacción entre módulos.

El diseño general involucra varios problemas; entre los más importantes se cuentan la selección del lenguaje de enlace, la selección de las estructuras de datos, la selección de una política de control general y de manejo de errores. Estas dos últimas pueden ser: a) control centralizado, b) control descentralizado o distribuido, c) control parcialmente descentralizado.

Es necesario también seleccionar el nivel del sistema. El nivel en el que conviene situar al sistema es un nivel de aplicaciones, esto es, el más alto y cercano al usuario no programador, el usuario sólo seleccionará opciones y no requiere programar en un lenguaje convencional. La selección de opciones, (en este caso B y C son un solo módulo) puede ser por menú (simple o extenso) o a través de un lenguaje de uso sencillo especialmente construido para el sistema.

La especificación de los módulos funcionales, especialmente de los módulos de ejecución y generador de la salida gráfica y la descripción de la interacción entre estos, definen aunque informalmente las aplicaciones que el sistema de graficación puede tener. De hecho es un modelo conceptual para el programador del sistema de aplicaciones gráficas interactivas, en el que se describen objetos que en el se podrán manejar y otros datos para las facilidades que se implementen [9].

A continuación se describen las facilidades mínimas con que se debería contar un sistema de graficación computarizada y que se incluirán en el sistema que describe en el presente trabajo y que se aplican a un objeto que es desplegado.

### 3.2. Graficación de un objeto.

Un objeto a graficar en la computadora, es un conjunto de puntos que tienen asociados color, intensidad y posición. Para poder graficar el objeto en una pantalla, las posiciones de los puntos que constituyen al objeto deberán expresarse en las coordenadas que utilice el sistema de despliegue [7].

A continuación se describen los elementos básicos que se deben conocer para poder graficar un objeto en una pantalla de computadora y posteriormente se describen las formas como se puede transformar el objeto.

#### 3.2.1. Graficación de un punto.

El CRT (Tubo de Rayos Catódicos) es un plano bidimensional en el cual lo normal es que las localizaciones de los puntos se describan por medio de un sistema coordenado cartesiano [7]. Al utilizar este sistema, existe la tendencia a considerar que el extremo inferior del CRT es el eje X, la orilla izquierda es el eje Y, y la esquina inferior izquierda es el origen, como en un sistema coordenado cartesiano derecho (véase la figura 3.2). Sin embargo, debido a la organización de la memoria, la mayor parte de los sistemas de computadora se basan en un sistema coordenado cartesiano derecho y utilizan como origen la esquina superior izquierda (fig. 3.3).



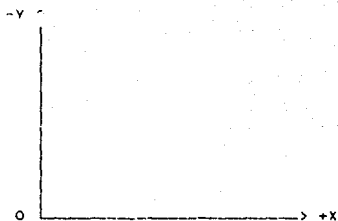


Fig. 3.2. Descripción de un sistema coordenado cartesiano derecho con el origen en la esquina inferior izquierda.

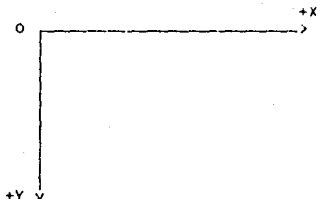


Fig. 3.3. Descripción de un sistema cartesiano derecho con el origen en la esquina superior izquierda.

### 3.2.2. Conversión de un punto en pantalla por rastreo.

Cada pixel en el exhibidor de gráficas representa una región que teóricamente puede contener un número infinito de puntos. Convertir por rastreo un punto implica iluminar el pixel que lo contiene. Por ejemplo, los puntos coordenados de exhibición  $P_1(2 \frac{1}{3}, 1 \frac{3}{4})$  y  $P_2(2 \frac{2}{3}, 1 \frac{1}{4})$  de la figura 3.4) estarían expresados ambos por el pixel (2,1). En general, un punto  $P(x,y)$  está representado por la parte entera de  $x$  y la parte entera de  $y$ , esto es, pixel  $(\text{int}(x), \text{int}(y))$  [6].

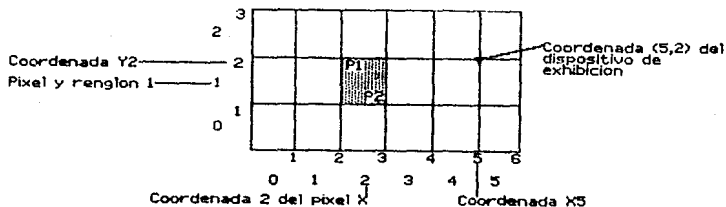


Fig. 3.4. Puntos coordenados de exhibición en un CRT.

### 3.2.3. Graficación de una línea recta por rastreo.

Una línea recta puede definirse por medio de dos puntos extremos y una ecuación (véase la fig. 12)]. En la figura 3.5 los dos puntos extremos están descritos por  $(X_1, Y_1)$  y  $(X_2, Y_2)$ . La ecuación de la recta se utiliza para describir las coordenadas  $X, Y$  de todos los puntos que se encuentran entre estos dos puntos extremos. Utilizando la ecuación de la línea recta,  $Y=MX+B$ , en donde  $M=\Delta Y/\Delta X$  ( $\Delta$  = incremento) y  $B$ =intersección con el eje  $Y$ , es posible determinar los valores de  $Y$  al incrementar  $X$  de  $X=X_1$  a  $X=X_2$ . Al convertir por rastreo estos valores  $X, Y$  calculados, se representa a la recta como una secuencia de píxeles [7].

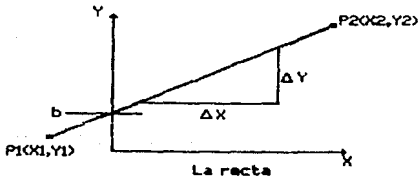


Fig. 3.5. Puntos extremos descritos en un CRT.

### 3.3. Sistema Coordinado.

Por convención, la esquina superior izquierda de la pantalla de gráficos es (0,0) [6]. Los valores de x, o columnas, se incrementan a la derecha. Los valores de y, o renglones, se incrementan hacia abajo. Así, en el modo CGA (Adaptador de Gráficos en Color) en 320x200, las coordenadas de la pantalla para cada una de las cuatro esquinas la pantalla se vería serán como aparece en la figura 3.6.

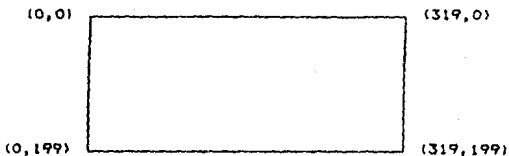


Fig. 3.6. Sistema coordinado de pantalla en el modo CGA.

### 3.4. Cambio de Coordenadas.

Cuando las coordenadas (0,0), (319,0), (0,199), (319,199) no son las adecuadas para el despliegue gráfico que se desea realizar, puede trabajarse con un nuevo sistema coordinado, en el que no se modifican el origen ni los ejes coordinados, pero se introduce distintas unidades de medición a lo largo de los ejes X e Y permitiendo abscisas y ordenadas de una magnitud mayor para desplegar. En el eje X se puede desplegar hasta un valor  $S_x$  y en el eje Y hasta un valor  $S_y$ . Un objeto es desplegado en este nuevo sistema coordinado a través de la transformación de escala  $\bar{S}_{S_x, S_y}$ :

$$(X', Y') = \bar{S}_{S_x, S_y} (X, Y)$$

en donde  $X' = (1/S_x)$  y  $Y' = (1/S_y)$ . En la figura 3.7 se muestra la transformación de escala coordinada utilizando los factores de

escalamiento  $S_x=2$  y  $S_y=1/2$  [6].

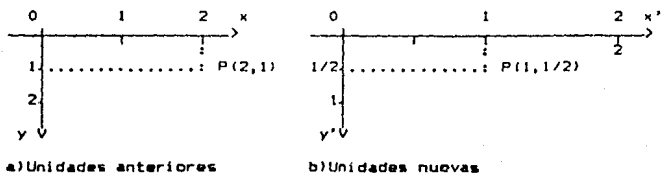


Fig. 3.7. Transformación de escala coordenada.

### 3.5. Herramientas de apoyo para Graficación.

#### 3.5.1. Sistemas Coordenados Bidimensionales.

Muchos paquetes de gráficas están diseñados para utilizar sistemas de Coordenadas Cartesianas, algunos permiten usar más de un sistema cartesiano ya que diferentes dispositivos de salida pueden requerir diferentes sistemas de coordenadas, y por lo general se permite que se definan imágenes u objetos en el sistema de referencia cartesiano que convenga a la aplicación que se desea.

Para la representación de un objeto en coordenadas se debe considerar un patrón real de medida (kilometros, metros, etc.) para poder establecer las dimensiones de cada eje de referencia, en un sistema de graficación, a este tipo de coordenadas se les conoce como coordenadas mundiales.

Las coordenadas mundiales y las coordenadas que utiliza un dispositivo de salida particular reciben el nombre de coordenadas

de dispositivo o bien coordenadas de pantalla en caso de que se trate de un monitor de video. Las definiciones de las coordenadas mundiales permiten a un usuario fijar cualquier dimensión adecuada sin verse obstaculizado por las restricciones de un dispositivo de salida determinado. Los proyectos arquitectónicos podrian especificarse en fracciones de un pie, mientras que otras aplicaciones podrian definir escalas de coordenadas en términos de milímetros, kilómetros o años-luz. Una vez dadas las definiciones de las coordenadas mundiales, el sistema de gráficas las convierte en las coordenadas de dispositivo adecuadas para su despliegue [10].

Un procedimiento común que se usa en los paquetes de gráficas consiste primero en convertir las definiciones de las coordenadas mundiales en coordenadas normalizadas antes de la conversión final a coordenadas de dispositivos específicas, esto hace que el sistema sea lo suficientemente flexible para dar cabida a varios dispositivos de salida ( fig. 3.8 ). Las coordenadas X y Y, normalizadas, reciben cada una valores en el ámbito de 0 a 1. Estas coordenadas normalizadas se transforman después en coordenadas de dispositivo (enteras) dentro del intervalo (0,0) a  $(X_{max}, Y_{max})$  para un dispositivo particular. Para tener en cuenta diferencias de escalas y razones de aspecto, las coordenadas normalizadas pueden trazarse en un área cuadrada del dispositivo de salida de manera que se conserven las proporciones adecuadas. En un monitor de video, el área restante de la pantalla a menudo sirve para desplegar mensajes o listar opciones de programa interactivos [6].

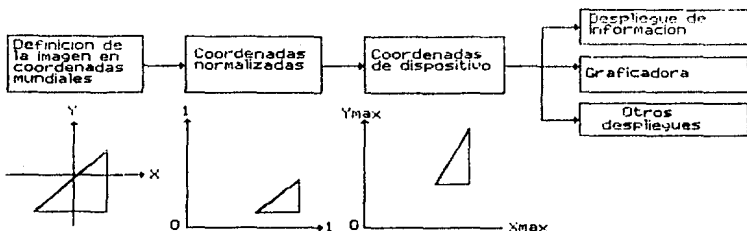


fig. 3.8. La transformación de la definición de la imagen de un usuario (coordenadas mundiales) en coordenadas normalizadas a menudo es efectuada por un sistema de gráficas a fin de ofrecer una interface con diferentes tipos de dispositivos de salida.

### 3.5.2. Escalamiento.

Una transformación que altera el tamaño de un objeto se denomina escalamiento. Esta operación cuando se trata de polígonos se realiza multiplicando los valores coordenados  $(x, y)$  de cada vértice de la frontera del polígono por los factores de escalamiento  $S_x$  y  $S_y$  para producir las coordenadas transformadas  $(x', y')$ .

$$x' = x \cdot S_x \quad , \quad y' = y \cdot S_y$$

El factor de escalamiento  $S_x$  escala el objeto en la dirección  $x$ , mientras que  $S_y$  lo hace en la dirección  $y$ .

Cualquier valor numérico positivo puede asignarse a los factores de escalamiento  $S_x$  y  $S_y$ . Los valores menores que 1 reducen el tamaño de los objetos, los valores mayores producen un agrandamiento. Si se especifica un valor de 1 para  $S_x$  y  $S_y$  se mantiene inalterado el objeto. Cuando a  $S_x$  y  $S_y$  se les asigna el mismo valor, se produce una escalación uniforme, que mantiene las

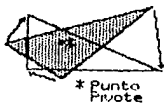
proposiciones relativas del objeto a escala [6].

### 3.5.3. Rotación.

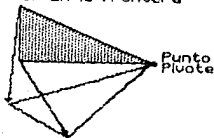
Cuando los puntos que constituyen un objeto son desplazados en una trayectoria circular se obtiene una rotación del objeto. Este tipo de transformación es especificada a través de un ángulo de rotación, que define cuanto se rotará cada vértice y/o punto del objeto [6].

Se puede hacer que los objetos giren alrededor de un punto arbitrario denominado punto de rotación seleccionado o punto pivote. El punto pivote de la transformación de rotación puede colocarse en cualquier parte en el interior o en el exterior del objeto. Cuando el punto pivote se especifica en la frontera o límite del objeto, el efecto de la relación consiste en oscilar el objeto con respecto a este punto interno (ver figura 3.9). Con un punto pivote externo, todos los puntos del objeto se despliegan en trayectorias circulares alrededor del punto pivote (ver figura 3.10).

a) En el interior



b) En la frontera



c) En el exterior

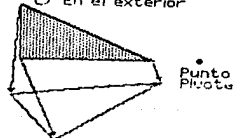


Fig. 3.9. Rotación de un objeto con el punto pivote colocado en la parte interior del objeto, la frontera del objeto y en la parte exterior del objeto.

Este tipo de transformación se especifica con un ángulo de rotación, el cual determina la cantidad de rotación de cada vértice de un polígono.

La figura 3.10 ilustra el desplazamiento de un punto de la posición  $(x, y)$  a la posición  $(x', y')$ , como lo determina un ángulo de rotación especificado  $\theta$  relativo al origen coordenado.

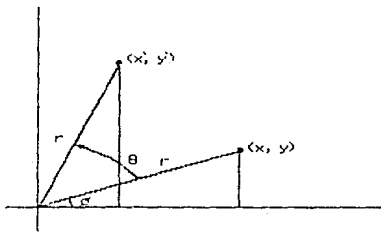


Fig. 3.10. Rotación de un punto desde la posición  $(x, y)$  hasta la posición  $(x', y')$  a través de un ángulo de rotación  $\theta$ , que se especifica relativo al origen coordenado. La posición angular original del punto desde  $x$  es  $\phi$ .

En esta figura, el ángulo  $\theta$  es la posición angular original del punto desde la horizontal. Se pueden determinar las ecuaciones de transformación de la rotación del punto de las relaciones existentes entre los lados de los triángulos rectángulos que se muestran y los ángulos asociados mediante el uso de estos triángulos e identidades trigonométricas normales, se puede escribir:

$$x' = r \cos(\theta + \phi) = r \cos\theta \cos\phi - r \sin\theta \sin\phi$$

$$y' = r \sin(\theta + \phi) = r \sin\theta \cos\phi + r \cos\theta \sin\phi \quad (A)$$

donde  $r$  es la distancia del punto desde el origen. También se tiene



$$x = r \cos\theta, \quad y = r \sin\theta \quad (B)$$

de modo que las ecuaciones (A) pueden volver a expresarse en términos de  $x$  y  $y$  como

$$\begin{aligned} x' &= x \cos\theta - y \sin\theta \\ y' &= y \cos\theta + x \sin\theta \end{aligned} \quad (C)$$

Los valores positivos de  $\theta$  en estas ecuaciones indican una rotación contraria a las manecillas del reloj y los valores negativos de  $\theta$  hacen girar los objetos en el sentido de las manecillas del reloj.

Se puede hacer que los objetos giren alrededor de un punto arbitrario modificando las ecuaciones (C) para incluir las coordenadas  $(x_R, y_R)$  para el punto de rotación seleccionado (o punto pivote). La rotación con respecto a un punto de rotación arbitrario se muestra en la figura 3.11.

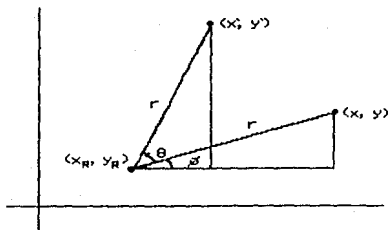


Figura 3.11. Rotación de un punto desde  $(x, y)$  hasta  $(x', y')$  a través de un ángulo  $\theta$ , que se especifica relativo a un punto pivote situado en  $(x_R, y_R)$ .

Las ecuaciones de transformación de las coordenadas rotadas pueden obtenerse a partir de las relaciones trigonométricas de esta figura como

$$x' = x_R + (x - x_R)\cos\theta - (y - y_R)\sin\theta$$

$$y' = y_R + (y - y_R)\cos\theta + (x - x_R)\sin\theta$$

Como los cálculos de rotación implican funciones trigonométricas y varias operaciones aritméticas para cada punto, el tiempo de cálculo por computadora puede volverse excesivo.

#### 3.5.4. Traslación.

Una traslación es el movimiento en línea recta de un objeto de una posición a otra. Se traslada un punto de la posición coordenada  $(x,y)$  a una nueva posición  $(x',y')$  agregando distancias de traslación,  $T_x$  y  $T_y$ , a las coordenadas originales:

$$x' = x + T_x \quad ; \quad y' = y + T_y$$

El par de distancia de traslación  $(T_x, T_y)$  se denomina también vector de traslación o bien vector de cambio.

Los polígonos se trasladan agregando las distancias de traslación especificadas a las coordenadas de cada punto extremo de la línea en el objeto.

Las distancias de traslación pueden especificarse como cualquier número real (positivo, negativo o cero). Si un objeto se traslada más allá de los límites del despliegue en coordenadas del dispositivo, el sistema podría retornar un mensaje de error, suprimir partes del objeto que sobrepasan los límites del despliegue o presentar una imagen distorsionada [6].

#### 3.6. Dispositivos de entrada y salida en sistema de graficación.

Las estaciones de trabajo de gráficas pueden contar con varios dispositivos de entrada y salida. El dispositivo de

entrada que se usará en el sistema para realizar la captura de cadenas de caracteres y valores de datos será el teclado de la microcomputadora.

En un teclado se pueden incluir varios tipos de llaves, discos e interruptores para manejar diversas aplicaciones.

El teclado es un dispositivo eficaz para introducir datos no gráficos como son rótulos de imágenes asociados con un despliegue de gráficas. También tiene la característica de facilitar la entrada de valores numéricos, coordenadas de la pantalla, selecciones de menús o funciones de gráficas.

Las llaves de control del cursor y las llaves de funciones son características comunes en los teclados de uso general. Las llaves de funciones permiten a los usuarios introducir operaciones de uso común con un solo golpe de la llave; las llaves conocidas como de control del cursor, seleccionan posiciones coordinadas ubicando el cursor de la pantalla en un sitio dado del monitor de video.

Los sistemas de gráficas interactivos requieren además de la unidad central de procesamiento (UCP), y de los dispositivos de E/S, un PROCESADOR DE DESPLIEGUES de uso general que interactúa con la UCP y controla la operación del dispositivo de despliegue (ver figura 3.12).

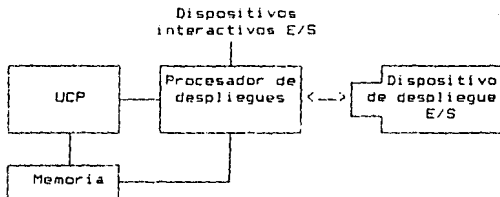


Fig. 3.12. Diagrama de Hardware simplificado de un sistema interactivo de gráficas.

El procesador de despliegue se utiliza para convertir información digital de la UCP en valores de una tensión adecuada al dispositivo de despliegue. La forma en la cual se realiza esta conversión de digital en analógico depende del tipo de despliegue que se usa (pantalla, graficador de pluma, etc.) y de las funciones de gráficas particulares que se han instrumentado en el hardware, algunos los dan en términos de niveles de intensidad de luz de puntos coordinados de la pantalla [6].

### 3.6.1. Dispositivos de exhibición.

Una computadora es capaz de enviar su salida a una amplia variedad de dispositivos, sin embargo, los que aquí interesan son sólo los dispositivos de aplicación especial y general capaces de producir salida gráfica, y los principios que ayudan a convertir un modelo almacenado en la memoria de una computadora en una imagen en el exhibidor de esa computadora, algunos ejemplos son: la pantalla de la computadora, la impresora, otra computadora ya sea del mismo tipo o de alguna aplicación específica, etc. [6].

Los dispositivos de computadora empleados por el sistema objeto de esta tesis para producir la salida gráfica son:

- Tubo de rayos catódicos de rastreo por barrido (CRT),
- Impresoras.

#### 3.6.1.1. CRT de rastreo por barrido.

El tipo más común de dispositivo de salida por computadora, de uso actual, capaz de exhibir salida gráfica, es el CRT de rastreo por barrido. Los CRT de rastreo por barrido se utilizan en los aparatos de televisión.

Un dispositivo digital es aquel que representa todas las cantidades como valores discretos. Uno analógico es el que representa todas las cantidades como valores continuos. Ya que la computadora es un dispositivo digital y el CRT es un dispositivo analógico, un procesador de despliegue interno, se encarga de tomar los valores discretos como continuos y así poderlos representar en el CRT. Esta interfase no será tratada en esta tesis, pero si explicaremos algunos detalles que aclaren la razón del comportamiento gráfico que existe dentro del sistema [6].

#### 3.6.1.2. Rastreo por barrido.

El término "rastreo" es sinónimo del término "matriz". Por tanto un CRT de rastreo por barrido barre una matriz con un haz de electrones.

El barrido es un recorrido en la pantalla o en el CRT el cual se comporta como en la figura 3.13.

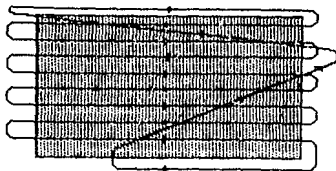


Fig. 3.13. Secuencia del barrido en pantalla o CRT (Tubo de rayos catódicos).

este recorrido exita a los puntos ó pixeles de la pantalla ya sea borrando o escribiendo un carácter o pixel, el barrido es continuo y en el caso de no haber modificación alguna se repetirá la representación en pantalla, este proceso ocurre de manera muy rápida por lo que un ser humano no detecta los cambios. Estos cambios al borrar una pantalla completa y crear una nueva si se notan debido a la velocidad posible del proceso, puesto que se tiene que recorrer dos veces la pantalla y por tanto se nota que se borra la pantalla y despues que se crea otra en el caso de una modificación total [6].

### 3.7. Funciones de graficación.

Un paquete de gráficas de uso general ofrece a los usuarios una variedad de funciones para crear y manipular imágenes. Estas rutinas pueden ser de: salida, entrada, definición de atributos, transformaciones de segmentos, visualización o control general.

Las estructuras básicas de despliegue se conocen como

primitivas de salida. Los procedimientos que despliegan primitivas de salida, dirigen a un dispositivo de salida para producir estructuras geométricas específicas en una posición coordinada proporcionada y en un dispositivo de salida seleccionado.

Las rutinas que ejecutan las primitivas de salida ofrecen las herramientas básicas para la construcción de figuras, generalmente son: puntos, líneas, rectángulos y círculos.

Los atributos son las propiedades de las primitivas de salida. Incluyen especificaciones de color e intensidad, estilos de líneas, estilos de texto y modelos de llenado de áreas. Las funciones dentro de esta categoría pueden usarse para fijar los atributos de grupos de primitivas de salida.

Dada la definición de primitivas y atributos de una figura en coordenadas mundiales, un paquete de gráficas proyecta una vista seleccionada de la imagen en un dispositivo de salida. Las transformaciones de vistas se utilizan para especificar la vista que se presentará y la porción del área en el despliegue de salida que se usará.

Por último, un paquete de gráficas contiene comúnmente varias tareas de mantenimiento, como el borrado de la pantalla de un despliegue o la inicialización de parámetros. Podemos agrupar las funciones para realizar estos trabajos rutinarios en el encabezado operaciones de control [7].

### 3.8. Normas de Software.

El objetivo principal del software de gráficas estandarizado es la portabilidad. Cuando los paquetes se diseñan con funciones

de gráficas estándar, el software puede moverse fácilmente hacia diferentes tipos de sistemas de hardware y usarse en diferentes instrumentaciones y aplicaciones, en otros casos, los programas diseñados para un sistema de hardware a menudo no pueden transferirse a otro sin reescribir el software [11].

Las organizaciones internacionales y nacionales de planeación de normas de muchos países han cooperado en un esfuerzo por crear un estándar que se acepte en general para las gráficas de computadora. Después de realizar un esfuerzo considerable, este trabajo sobre normas condujo a la creación del Sistema Kernel de Gráficas (GKS). Este sistema ha sido adoptado como norma de software de gráficas por la International Standards Organization (ISO) y por varias organizaciones nacionales de normas, como el American National Standards Institute (ANSI). Aunque GKS se diseñó originalmente como paquete de gráficas bidimensional, después se creó una extensión tridimensional del GKS.

Las funciones finales del GKS, adoptadas como normas, se vieron influenciadas por varias normas de gráficas propuestas con anterioridad. de particular importancia entre estas proposiciones iniciales es el Sistema de Gráficas de Nucleo (o SIGGRAPH, el Grupo de Interés Especial sobre Gráficas de Computadora de la Association for Computing Machinery (ACM) [6].

Las funciones de gráficas estándar se definen como un conjunto de especificaciones abstractas, independientes de cualquier lenguaje de programación. Para instrumentar una norma de gráficas en un lenguaje de programación determinado, debe



definiere una vinculación de lenguaje. Esta vinculación define la sintaxis para acceder las diferentes funciones de gráficas que se especifican dentro de la norma. Por ejemplo, GKS especifica una función para generar una secuencia de segmentos rectilíneos conectados con el título descriptivo [6].

```
polyline(n,x,y)
```

En FORTRAN 77, este procedimiento se instrumenta como una subrutina con el nombre GPL. Un programador de gráficas, utilizando FORTRAN, invocaría este procedimiento con la proposición para llamar a la subrutina

```
CALL GPL (N, X, Y)
```

Se han definido vinculaciones de lenguaje GKS para el FORTRAN, Pascal, Ada, C, PL/1 y COBOL. Cada vinculación del lenguaje se define para hacer aprovechar al máximo las capacidades del lenguaje correspondiente y para manejar varios aspectos de sintaxis, como tipos de datos, paso de parámetros y errores.

Aunque GKS presenta una especificación de funciones básicas de gráficas, no ofrece una metodología estándar para una interfase de gráficas con dispositivos de salida. Tampoco especifica métodos para modelado en tiempo real ni para almacenar y transmitir imágenes. Se han creado normas especiales para cada una de estas tres áreas. La estandarización de los métodos de interfase de dispositivos se da en el sistema de Interfase de las Gráficas de Computadora (CGI). El sistema Metaarchivo de Gráficas de Computadora (CGM) especifica normas para archivar y

transportar imágenes. Y la Norma de Gráficas Interactiva Jerárquica del Programador (PHIGS) define métodos estándar para modelado en tiempo real y otras capacidades de programación de nivel superior no consideradas por GKS.

Para otras explicaciones de las normas de software véase Graphics Standards Planning Committee, Comité de Planeación de Normas de Gráficas, (1977 y 1979), Hatfield y Herzog (1982) y Warner (1981) etc.

### 3.9. Técnicas de despliegue tridimensional.

Las representaciones de un objeto sólido en una superficie de visión por lo general contienen información de profundidad, para que se pueda identificar con facilidad en una imagen seleccionada dada, cuáles son las partes anterior y posterior del objeto.

#### 3.9.1. Proyecciones.

Las operaciones básicas definidas en dos dimensiones (rotación, traslación, etc.) se aplican también en el caso de tres dimensiones, agregando una coordenada de más a cada objeto. Sin embargo, falta encontrar la forma para representar el objeto en la pantalla que es un dispositivo bidimensional. Para lograr lo anterior se hace uso de las proyecciones.

Existen dos métodos básicos para proyectar objetos tridimensionales sobre una superficie de visión bidimensional: 1) Todos los puntos del objeto pueden proyectarse sobre una superficie a lo largo de líneas paralelas; 2) los puntos pueden proyectarse a lo largo de líneas que convergen hacia una posición

denominada centro de proyección. Los dos métodos, llamados proyección en paralelo y proyección en perspectiva, respectivamente, se ilustran en la figura 3.14:

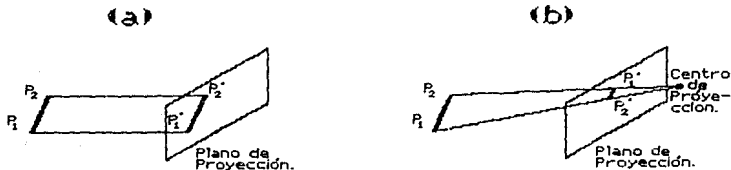


Fig. 3.14. Dos métodos para proyectar una línea sobre una superficie de visión (plano de proyección).

Una proyección en paralelo preserva dimensiones relativas de los objetos, ésta es la técnica que se utiliza en el dibujo mecánico para producir trazos a escala de objetos en tres dimensiones. Este método sirve para obtener vistas exactas de los diferentes lados de un objeto pero, sin embargo, una proyección en paralelo no ofrece una representación realista del aspecto de un objeto tridimensional. En cambio, una proyección en perspectiva produce vistas realistas pero no preserva las dimensiones relativas. Las líneas distantes se proyectan como menores que aquellas que están más próximas al plano de proyección, como se puede apreciar en la figura 3.15.

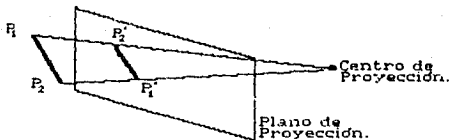


Fig. 3.15. Una línea distante se despliega de menor tamaño que una línea más cercana de igual longitud cuando se utiliza una proyección en perspectiva.

Para los fines que pretende cubrir este trabajo de tesis, lo que se requiere es el poder establecer una comparación exacta y clara de los datos que se encuentren involucrados en una gráfica es absurdo el representar en perspectiva la gráfica, ya que se tendrá error al comparar los puntos más profundos con los que están más próximos al plano de proyección, por esta razón se usará el método de proyección en paralelo [6].

#### 3.9.1.1. Proyecciones en Paralelo.

Un objeto sólido puede representarse en dos dimensiones proyectando puntos sobre la superficie del objeto a lo largo de líneas paralelas sobre una superficie de visión plana. Al seleccionar diferente posición de observación, se pueden proyectar puntos visibles sobre la superficie de visión con el fin de obtener diferentes vistas bidimensionales del objeto, como en la figura 3.16. En una proyección en paralelo, las líneas paralelas sobre la superficie del objeto se proyectan en líneas paralelas sobre el plano de visión bidimensional. Esta técnica se

utiliza en ingeniería y trazos arquitectónicos para representar el objeto con un conjunto de vistas que mantienen proporciones relativas del objeto. El aspecto del objeto sólido puede reconstruirse después a partir de las vistas principales.

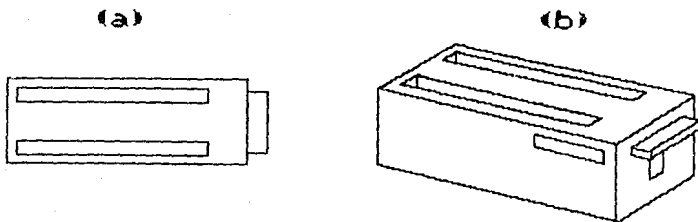


Fig. 3.16. Dos vistas con proyección en paralelo de un objeto, que se usan para mostrar proporciones relativas. Las líneas sobre el objeto que son paralelas siguen siéndolo en las vistas de proyección.

Las vistas formadas con proyecciones en paralelo pueden caracterizarse de acuerdo con el ángulo que la dirección de la proyección forma con el plano de proyección. Cuando la dirección de proyección es perpendicular al plano de proyección, se tiene una Proyección Ortogonal. Una proyección que no es perpendicular al plano, se denomina Proyección Oblicua.

Las proyecciones ortogonales son utilizadas con mayor frecuencia para producir las vistas del frente, lado y parte superior de un objeto.

También se pueden formar proyecciones ortogonales que muestren más de una cara de un objeto. Tales vistas se denominan proyecciones ortogonales axonométricas. La proyección

axonométrica que se usa con más frecuencia es la proyección isométrica. Una proyección isométrica se obtiene alineando el plano de proyección de modo que se corte con cada eje coordenado (denominados ejes principales) en el cual se defina el objeto a la misma distancia del origen, o dicho de otra manera, el plano de proyección paralela en dos dimensiones a la imagen de tres dimensiones (ver figura 3.17).

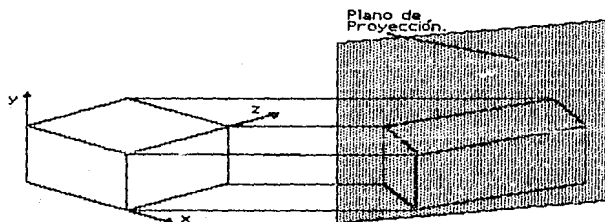


Fig. 3.17. Proyección isométrica de un objeto sobre una superficie de visión.

Existen ocho posiciones, una en cada octante, para obtener una vista isométrica. Los tres ejes principales se reducen equitativamente en una proyección isométrica de manera que se conserven las proporciones relativas. Este no es el caso en una proyección axonométrica general, donde los factores de escalación pueden ser diferentes en las tres direcciones principales.

Las ecuaciones de transformación para efectuar una proyección paralela ortogonal se obtienen fácilmente. Para cualquier punto  $(X, Y, Z)$ , el punto de proyección  $(X_p, Y_p, Z_p)$  sobre la superficie de visión  $(X, Y)$  se obtiene como

$$X_p = X,$$

$$Y_p = Y,$$

$$Z_p = 0$$

Una proyección oblicua se obtiene proyectando puntos a lo largo de líneas paralelas que no son perpendiculares al plano de proyección. La figura 3.18 muestra una proyección oblicua de un punto  $(X, Y, Z)$  por una línea de proyección a la posición  $(X_p, Y_p)$ . Las coordenadas de proyección ortogonal en el plano son  $(X, Y)$ . Esta línea, de longitud  $L$ , está en un ángulo  $\theta$  con la dirección horizontal en el plano de proyección. Podemos expresar las coordenadas de proyección en términos de  $X$ ,  $Y$ ,  $L$  y  $\theta$  :

$$X_p = X + L \cos \theta$$

$$Y_p = Y + L \sin \theta$$

(ecs. 1)

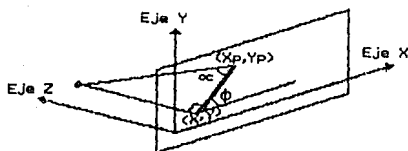


Fig. 3.18. Proyección oblicua del punto  $(X, Y, Z)$  a la posición  $(X_p, Y_p)$  sobre el plano de proyección.

Una dirección de proyección puede definirse seleccionando valores para los ángulos  $\alpha$  y  $\theta$ . Alternativas comunes del ángulo  $\theta$  son  $30^\circ$  y  $45^\circ$ , los cuales despliegan una vista combinada del frente, el lado y la parte superior (o bien del frente, lado y parte inferior) de un objeto. La longitud  $L$  es función de la coordenada  $Z$  y podemos evaluar este parámetro a partir de las relaciones

$$\tan \alpha = Z / L = l / L_1$$

(ecs. 2)

donde  $L_1$  es la longitud de la línea de proyección de  $(X, Y)$  a  $(X_p, Y_p)$  cuando  $Z=1$ . De la ecuación (ecs. 2), se tiene

$$L = Z L_1 \quad (\text{ecs. 3})$$

y las ecuaciones de proyección oblicua (ecs. 1) puede escribirse como

$$X_p = X + Z (L_1 \cos \theta)$$

$$Y_p = Y + Z (L_1 \sin \theta)$$

Una proyección ortogonal se obtiene cuando  $L_1=0$  (que ocurre en un ángulo de proyección  $\alpha$  de  $90^\circ$ ). Las proyecciones oblicuas se generan con valores distintos de cero para  $L_1$ . Los valores de las coordenadas  $X$  e  $Y$  dentro de cada plano de  $Z$  constante se cambian en una cantidad proporcional al valor de  $Z$  del plano de manera que los ángulos, distancias y líneas paralelas del plano se proyecten con exactitud. Este efecto se muestra en la figura 3.19, donde el plano anterior de la caja se corta y se cubre con el plano frontal de la proyección a la superficie de visión. Una arista de la caja que conecta los planos anterior y frontal se proyecta en una línea de longitud  $L_1$  que forma un ángulo  $\theta$  con una línea horizontal en el plano de proyección.

Dos ángulos que se usan comúnmente en las proyecciones oblicuas son aquellas para las cuales  $\tan \alpha = 1$  y  $\tan \alpha = 2$ . En el primer caso,  $\alpha = 45^\circ$  y las vistas que se obtienen se denominan proyecciones CABALLERA. Todas las líneas perpendiculares al plano de proyección se proyectan sin cambio de longitud. Algunos ejemplos de las proyecciones caballera de un cubo se dan en la figura 3.20.



Cuando el ángulo de proyección se escoge tal que  $\tan \alpha = 2$ , a la vista resultante se le llama proyección DE GABINETE. Este ángulo de proyección de aproximadamente  $63.4^\circ$  ocasiona que las líneas perpendiculares a la superficie de visión se proyecten en una mitad de su longitud. Las proyecciones de gabinete parecen más realistas que las de montura debido a esta reducción en la longitud de las perpendiculares. La figura 3.21 muestra una proyección de gabinete de un cubo (6).

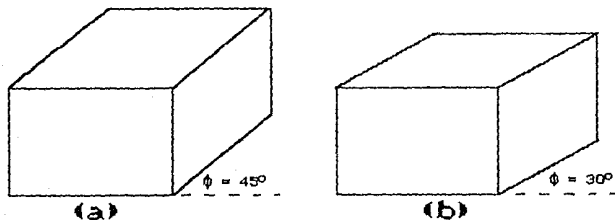


Fig. 3.19. Proyección oblicua de una caja sobre una superficie de visión en el plano  $Z=0$ .

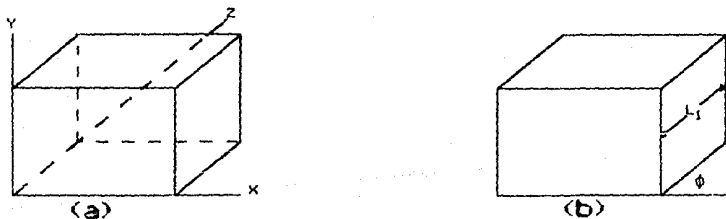


Fig. 3.20. Proyecciones caballeras de un cubo sobre un plano de proyección con dos valores del ángulo  $\phi$ . La profundidad del cubo se proyecta igual a la anchura y altura.

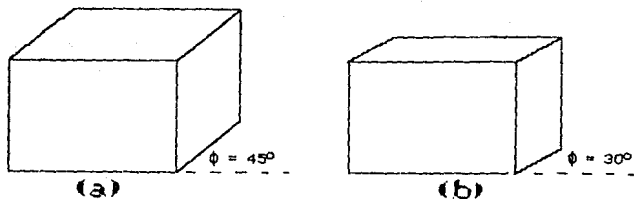


Fig. 3.21. Proyecciones de gabinete de un cubo sobre un plano de proyección con dos valores del ángulo  $\phi$ . la profundidad del cubo se proyecta igual a la anchura y altura.

### 3.9.2. Rotación en tres dimensiones.

Para especificar una transformación de rotación de un objeto, se debe designar un eje de rotación (en torno al cual se hará girar el objeto) y la cantidad de rotación angular.

En tres dimensiones, un eje de rotación puede tener cualquier orientación espacial. Los ejes de rotación más fáciles de manejar son aquellos que son paralelos a los tres ejes coordenados con el fin de producir una rotación en torno a cualquier eje de rotación especificado en forma arbitraria.

Adoptamos el convencionalismo de que las rotaciones en sentido contrario a las manecillas del reloj, en torno a un eje coordenado, se producen con ángulos de rotación positivos, si se observa la mitad positiva del eje en dirección del origen coordenado (figura 3.22). Este convencionalismo concuerda con nuestra explicación previa de las ecuaciones de rotación en dos dimensiones, que especifican rotaciones en torno al eje Z. Las ecuaciones de rotación del eje Z bidimensional se desarrollan

fácilmente a tres dimensiones (6):

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta \quad (\text{ecu. 4})$$

$$z' = z$$

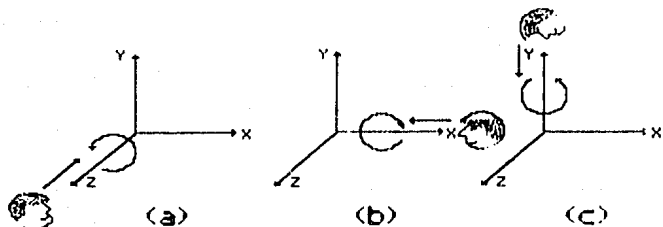


Fig. 3.22. Las direcciones de rotación positivas en torno a los ejes coordenados son en sentido contrario a las manecillas del reloj como se observa a lo largo de la posición positiva de cada eje en dirección del origen.

La figura 3.23 ilustra la rotación de un objeto en torno al eje Z.

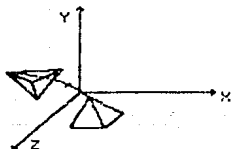


Fig. 3.23. Rotación de un objeto en torno al eje Z.

Las ecuaciones de transformación para rotaciones en torno a los ejes coordenados pueden obtenerse con una permutación cíclica de los parámetros coordenados de las ecuaciones (ecs. 4). Sustituimos  $x$  por  $y$ ,  $y$  por  $z$ , y  $z$  por  $x$ , utilizando esta permutación en las ecuaciones para una rotación del eje  $x$ :

$$\begin{aligned} y' &= y \cos \theta - z \sin \theta \\ z' &= y \sin \theta + z \cos \theta \\ x' &= x \end{aligned} \quad (\text{ecs. 5})$$

La rotación en torno al eje  $z$  se demuestra en la figura 3.24.

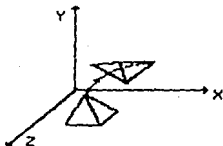


Fig. 3.24. Rotación de un objeto alrededor del eje  $X$ .

Las coordenadas cíclicamente permutantes de las ecuaciones (ecs. 5) dan las ecuaciones de transformación para una rotación del eje  $y$ :

$$\begin{aligned} z' &= z \cos \theta - x \sin \theta \\ x' &= z \sin \theta + x \cos \theta \\ y' &= y \end{aligned}$$

En la figura 3.25 se muestra un ejemplo de la rotación del eje  $y$ .

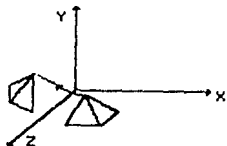


Fig. 3.25. Rotación de un objeto en torno al eje  $Y$ .

#### IV. DESCRIPCIÓN DEL SISTEMA

SIDEVIS es un sistema interactivo para microcomputadoras IBM PC compatibles desarrollado en el lenguaje de programación Turbo Pascal Versión 5.0 y compilado en la Versión 5.5.

Las funciones que cumple el sistema son básicamente dos:

- 1) Permite analizar, tanto numérica, como gráficamente, el comportamiento de funciones de una dos o tres variables,
- 2) Simular fenómenos dinámicos a lo largo del tiempo y graficar su comportamiento.

Para la realización de la primera, el sistema cuenta con facilidades para generar los valores de esta(s) de acuerdo a especificaciones dadas por el usuario para describir el comportamiento de éstas a petición del usuario.

La segunda función permite describir el comportamiento de fenómenos a través del tiempo. En este tipo de situaciones es necesario efectuar una serie de cálculos repetitivos y a veces tediosos, amén de que requieren de mucho tiempo para su realización. El sistema viene a simplificar esta tarea debido a que puede elaborar estos cálculos a través del tiempo, y trazar gráficas de los valores obtenidos con gran velocidad.

Las razones de utilizar el lenguaje Pascal en la implementación del presente sistema son:

- a) Su disponibilidad en microcomputadoras IBM PC y compatibles.
- b) La transportabilidad, ya que se puede ejecutar un programa escrito en Pascal en una computadora con un procesador, un sistema operativo o ambos, diferentes al que se usó en el

desarrollo del programa [12].

Turbo Pascal se ha diseñado para permitir la transportabilidad de código entre todas sus versiones posteriores, esto es, las versiones nuevas aceptan a las versiones anteriores a éstas.

c) Es un lenguaje estructurado, y por esto se puede conseguir una expresión clara y elegante de los algoritmos, que participaron en el sistema, logrando con esto un mayor control en la programación del sistema.

d) Sus facilidades de graficación, incluyendo 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. Estas facilidades de programación para desplegar y manipular salidas de gráficas las brinda Turbo Pascal 5.0 [13].

#### 4.1 Requerimientos de equipo.

Requerimientos mínimos de software y Hardware. Para hacer uso del sistema SIDEVIS:

- 1) Una microcomputadora IBM-PC o compatible con al menos 256 Kb de memoria RAM.
- 2) Monitor a color o monócromático.
- 3) Tarjeta para poder desplegar gráficos en pantalla, y en caso de tener monitor monócromático, es necesario además un emulador de color e intensidad como es el caso de los "Programas para tarjeta de video. Printaform".
- 4) Dos discos flexibles alojando el sistema SIDEVIS, o un disco

de 3 1/2 pulgadas.

5) Sistema operativo Ms-Dos.

#### 4.2 Estructuras del Sistema.

SIDEVIS cuenta con un menú principal el cual está inmerso en el programa principal del sistema (SIDEVIS.PAS), dicho menú da al usuario la capacidad de efectuar el análisis inmediato de una o varias funciones, o de efectuar dicho análisis de manera particionada, esto es, primero preparar el archivo en el que se alojarán los datos, segundo definir la función o funciones que darán los valores a los datos, tercero el procesamiento de las funciones y la generación de los datos y, cuarto y último, la graficación de los datos. Cabe notar que si el usuario lo desea, puede crear un archivo de datos y él mismo insertar los valores a éstos sin necesidad de alguna función.

Además tiene en su menú procesos secundarios como son la consulta de ayuda para los casos en que exista duda en el manejo de alguna sección del sistema; procesos para el mantenimiento de discos y el uso del sistema operativo estando dentro del sistema.

A continuación se describe a cada uno de los archivos que contienen las unidades principales y demás programas auxiliares usados por el sistema, así como el diagrama de flujo del mismo, en la figura 4.1.

SIDEVIS se compone de 5 unidades principales de las cuales las unidades GRAFICA3.PAS y FUNCION.PAS generan nuevas unidades versiones de programas fuente en Pascal y ejecutables cuando se requiere. El controlador del sistema es el programa SIDEVIS.



NUM	PROGRAMA O UNIDAD	No. DE LINEAS PROGRAMA FUENTE	BYTES FUENTE	EJECUTABLE
1	SIDEVIS	1089	41389	32432
2	GRAFICA1	985	36847	58976
3	GRAFICA2	824	30618	51760
4	GRAFICA3	398	12789	-----
5	FUNCION	69	2564	-----
TOTAL	5	3365	124207	141168

Los Programas más importantes que componen el sistema son:

- 1.-SIDEVIS.EXE
- 2.-AYUDA.SDV
- 3.-INSTRUYE.SDV
- 4.-FUNCION.PAS
- 5.-GRAFICA1.EXE
- 6.-GRAFICA2.EXE
- 7.-GRAFICA3.PAS
- 8.-TPC.EXE
- 9.-TPC.TPL
- 10.-TURBO.EXE
- 11.-TURBO.TPL
- 14.-GRAPH.TPU (La unidad de gráficos)
- 15.-\*.BGI (Las unidades para las distintas tarjetas gráficas)

De esto, los programas contemplados del inciso 10 al inciso 15 son parte del lenguaje de programación de Turbo Pascal 5.0.

La estructura del sistema se describe en la figura 4.1.

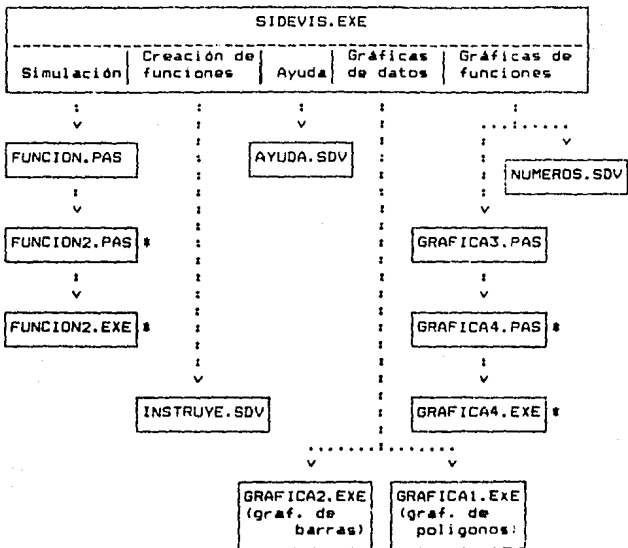


Fig. 4.1. Estructura de los programas fundamentales que integran al sistema, las unidades nuevas que genera el sistema están señaladas por (\*).

#### 4.2.1. Descripción de los programas que comprenden el sistema.

##### 4.2.1.1. GRAFICA1.

El programa contiene varios procedimientos, los cuales se describen a continuación:

- 1.- Procedimiento LEEDATOS. Este procedimiento es el primero en ejecutarse. solicitará el nombre del archivo de datos que se desea graficar, y de existir, guardará los valores de las

dimensiones del archivo para usarlos posteriormente. En caso de que no existá el archivo, enviará un mensaje y se saldrá de GRAFICAL regresando al menú principal del sistema.

2.- Procedimiento DIMENSIONES. De acuerdo a las características del archivo que se leyó en LEEDATOS, activará el procedimiento que sea el adecuado para la graficación de los datos. En el caso de que los datos puedan tratarse con dos de los procedimientos, se enviará un mensaje al usuario, dándole las opciones de los diferentes procedimientos a usar para que elija el indicado a sus propósitos y que son:

- A. Un\_Vector\_VS\_Tiempo
- B. Vector\_VS\_Vector
- C. Dos\_Vectores\_vs\_Tiempo
- D. Vector\_Vector\_Vector
- E. Tres\_Vectores\_VS\_Tiempo

A. Un\_Vector\_VS\_Tiempo .- Este procedimiento se encarga de graficar tomando como una de las coordenadas al valor del dato y como la otra coordenada la posición del dato dentro del vector de datos. Por ejemplo, en el vector siguiente:

```
      25  
-->  31  
      13  
      10  
      .  
      .  
      .
```

el dato es 31 y su posición es la 2. La graficación se hará al gusto del usuario, esto es, dando la libertad de escoger el eje para que el dato quede en las Y's (2,31) o en las X's (31,2) según el gusto o necesidad del usuario.

B. Vector\_VS\_Vector .- Este procedimiento va a utilizar como coordenadas los valores de los vectores 1 y 2, permitiendo que el usuario elija el eje que desea asignar a cada vector de datos.

C. Dos\_Vectores\_vs\_Tiempo .- Este procedimiento asignará un eje coordenado a los dos vectores de datos y el otro eje coordenado será tiempo, el cual quedará definido por el lugar en el que está posicionado el dato en el vector. El resultado serán dos gráficas superpuestas en el plano.

D. Vector\_Vector\_Vector .- Este proceso asigna un eje a cada vector, esto es, se obtendrá una representación gráfica de tres dimensiones en dos dimensiones, pudiéndose rotar la figura resultante en torno a alguno de los ejes ya sea X, Y ó Z.

E. Tres\_Vectores\_VS\_Tiempo .- Este procedimiento trabaja igual que el C pero con tres vectores en vez de dos, y su representación gráfica será de tres curvas en el plano.

3.- Procedimiento INIT. Es activado desde los procedimientos de graficación A,B,C,D y E descritos anteriormente. Su función es buscar el modo gráfico que vaya de acuerdo al sistema de computo que se esté usando, de no haber alguno en existencia, el sistema lo indicará y se saldrá del programa GRAFICAL retornando al menú principal del sistema. En caso de que todo este bien procederá a la graficación.

#### 4.2.1.2. GRAFICA2.

El programa tiene un funcionamiento estructural semejante al de GRAFICAL con algunas variantes mínimas. Sus procedimientos y el uso de los mismos se describen a continuación.

1.- El primer procedimiento que se usa es LEEDATOS, el cual

pedirá se tecleé el nombre del archivo de datos y de existir guardará los valores de las dimensiones del archivo para usarlos posteriormente en el inciso 3. En caso de que no exista el archivo enviará un mensaje y posteriormente saldrá de GRAFICAZ regresando al menú principal del sistema.

2.- A continuación se usa el procedimiento DIMENSIONES el cual de acuerdo a las características del archivo envía al procedimiento que sea más conveniente para los datos, en los casos en que los datos puedan tratarse con dos de los procedimientos, se pondrá un mensaje al usuario dándole las opciones de los diferentes procedimientos a usar para que él elija el indicado a sus propósitos y que son:

- A. Líneas
- B. Barras
- C. Barras\_3D

A. Líneas .- Este procedimiento se encarga de graficar líneas en un plano cartesiano de dos dimensiones, donde uno de los ejes representará al tiempo y el otro eje a la variable a graficar. La gráfica consistirá de líneas las cuales tomarán la longitud del dato, por ejemplo, en el vector siguiente:

```
    25  
--> 31  
    13  
    10  
    .
```

el dato es 31 y su posición es la 2. La graficación se hará al gusto del usuario, esto es, dando la libertad de escoger el eje para que el dato quede en las Y's (2, 31) o en las X's (31, 2) según el gusto o necesidad del usuario.

B. Barras .- Este procedimiento va a utilizar los valores de dos vectores de datos en disco, permitiendo que el usuario elija el factor de barra o ancho de esta. El otro vector define la altura de la barra.

C. Barras\_3D .- Este procedimiento a diferencia del anterior utilizará los valores de los datos en disco (tres vectores), permitiendo también al usuario elegir la relación que guarda cada barra, en este caso barras tridimensionales o mejor dicho prismas rectangulares. Cada prisma representa a un trio de datos mediante el uso de anchura, profundidad y altura del prisma.

3.- Procedimiento INIT. Es activado desde los procedimientos de graficación A,B y C descritos anteriormente. Su función es buscar el modo gráfico que vaya de acuerdo al sistema de cómputo que se esté usando, de no haber alguno en existencia, el sistema lo indicará y se saldrá del programa GRAFICA2 retornando al menú principal del sistema. En caso de que todo este bien procederá a la graficación.

#### 4.2.1.3. GRAFICA3.PAS (GRAFICA4).

Este programa genera el programa GRAFICA4.PAS para posteriormente hacerlo ejecutable y efectuar el cálculo requerido por el usuario.

Este programa, realiza los cálculos de cada función que el usuario desee experimentar y/o graficar, por lo que requiere los parámetros de variación que requerirá cada función y para así poder crear un sistema universal coordinado para los gráficos representativos de dichas funciones.

GRAFICA3 utiliza los datos contenidos en el archivo

NUMEROS.SDV, el cual contiene el número de funciones y el número de variables que se van a trabajar, posteriormente usando el número de variables GRAFICA3.PAS escogerá uno de los siguientes procedimientos:

A. DosVariables

B. TresVariables

A. DosVariables .- El procedimiento pide al usuario los valores que irá tomando la variable "x" desde el valor mínimo al máximo y también el incremento en que irá aumentando el valor mínimo dado hasta llegar al valor máximo. Después se analizarán los datos generados mediante las funciones dadas por el usuario e insertadas en GRAFICA4.PAS, una vez compilado GRAFICA4 se usan los valores dados anteriormente para con esto establecer el escalamiento apropiado para que todos los puntos a graficar queden contemplados en la pantalla. Por último graficará a cada función.

B. TresVariables .- El procedimiento trabaja igual que el A pero además pide los valores que la variable "y" irá tomando. Como se trata de tres variables, se dará una representación de una gráfica de tres dimensiones en dos dimensiones.

Los procedimientos A y B usan el procedimiento INIT para aceptar o rechazar la continuación del programa. El desarrollo de este procedimiento se indica a continuación:

2.- El procedimiento INIT busca el modo gráfico que vaya de acuerdo al sistema de cómputo que se este usando, de no haber alguno en existencia, el sistema lo indicará y se saldrá del programa GRAFICA3 y retornará al menú principal del sistema.

Nota: De exceder las limitaciones en el número de variables

el programa lo indicará con un mensaje y regresará al menú principal del sistema principal.

#### 4.2.1.4. FUNCION.PAS (FUNCION2.PAS).

El desarrollo de este programa es muy sencillo, es un programa fuente que acepta hasta 15 funciones desde una hasta tres variables, como no las toma como si fueran variables simples es necesario entrar en un proceso de compilación el cual se efectuará de la manera siguiente. El programa se irá duplicando en otro programa fuente llamado FUNCION2.PAS hasta llegar a la zona de aceptación de las funciones anteriormente mencionadas, es aquí donde insertará a las funciones en el mismo código fuente en Turbo Pascal, por lo que es necesario que al insertar las funciones se siga un formato parecido al Pascal, esto se describirá posteriormente, y una vez terminado el proceso de inserción de las funciones se continuará insertando el resto del código de FUNCION.PAS. Al generar a FUNCION2.PAS se compilará y se ejecutará mediante la ayuda del programa TPC.EXE de las utilerías de Turbo Pascal para que se realicen las operaciones indicadas por las funciones anteriormente insertadas y depositar el resultado de estas en un archivo el cual viene indicado dentro del programa. Tanto las operaciones a efectuar como el archivo al que se envían se indican por medio del programa principal.

#### 4.2.1.5. SIDEVIS

Este programa es el programa principal del sistema debido a que desde él se manejan a lo demás programas y éste los llama según lo requiera el usuario mediante un menú principal de



selecciones, el cual está incluido en este programa. En los casos de los programas fuente que requieren compilación y ejecución, SIDEVIS.PAS es el encargado de llevar a cabo el desarrollo de estas tareas.

#### 4.2.1.6 INSTRUYE.SDV.

Es un archivo de texto que contiene ayudas rápidas para el usuario en su interacción con el editor del sistema, de tal manera que al requerir la presencia de este texto en pantalla, el sistema lo desplegará mediante una orden.

#### 4.2.1.7. AYUDA.SDV.

De la misma manera que el archivo anterior, este archivo sirve de auxilio al usuario, a diferencia que este archivo ayuda al usuario de manera general pues ayuda a comprender el manejo del sistema de manera global.

### 4.3. Partes que integran al sistema.

El sistema cuenta con las siguientes secciones:

- 4.3.1. Editor de datos.
- 4.3.2. Editor de funciones.
- 4.3.3. Sección de simulación.
- 4.3.4. Sección de graficación de vectores de datos.
- 4.3.5. Sección de graficación de funciones.
- 4.3.6. Sección de ayuda al usuario.
- 4.3.7. Sección de interacción con el sistema operativo.
- 4.3.8. Opción de salida del sistema.

4.3.1. Editor de Datos. Da al usuario la capacidad de crear un archivo de datos, los cuales están divididos en tres vectores, esto es, para poder recibir hasta tres variables (X,Y,Z) para poder posteriormente cambiarlos o capturar los diferentes valores que tomen en una simulación de tipo determinista.

En la formación del archivo de datos el primer número que va a recibir va a ser la longitud que van a tener los vectores, el segundo va a ser la cantidad de vectores que manejará el archivo, los cuales varían de 1 a 3 vectores, y por último los valores que tomarán los vectores.

Se pueden ir insertando los valores de manera directa al archivo a trabajar en este editor o, en su caso, modificar alguno de los valores insertados, cabe señalar que la longitud de los vectores y la cantidad de éstos no se podrán modificar.

También por medio de este editor se puede mandar llamar para su consulta o para su modificación, se puede salvar al archivo con el mismo nombre que tenga en el disco, o crear al mismo con o sin modificaciones en otro archivo en disco.

Para crear un archivo de datos de manera externa al sistema es necesario crear un archivo de datos de tipo real donde los primeros dos valores sean el número de vectores (de uno a tres) y la capacidad general de los vectores, los valores posteriores serán los datos que contendrán los vectores de datos.

4.3.2. Editor de Funciones. Este editor sirve para poner las funciones que se van a emplear en alguna simulación, también se pueden poner comentarios de estas funciones, la capacidad de este editor es de 10 renglones por 72 columnas.

Para generar un archivo de funciones mediante el uso de otro

editor ajeno al sistema siempre y cuando su salida sea en código ASCII y que se respete el requerimiento de 10 renglones por 72 columnas, de otra forma se tendrán errores en la compilación generada por el TPC.EXE.

4.3.3. Sección de simulación. Esta sección se encarga de combinar los archivos de datos con los de funciones para generar nuevos valores dando, la posibilidad de crear con éstos un nuevo archivo de datos para consulta, modificación o visualización. Además genera un archivo temporal para consulta inmediata o visualización llamado SIMULA.DSD, que puede ser modificado por el usuario si así lo desea.

4.3.4. Sección de graficación de datos. Esta opción da la posibilidad de representar de manera gráfica a algún archivo de datos dando, la opción de presentarlos ya sea en gráficas de barras o en gráficas de puntos y poligonales, según la manera en que mejor se aprecie el comportamiento de los datos a criterio del usuario.

4.3.5. Sección de graficación de funciones. Esta opción sirve para consulta rápida de funciones de dos o tres variables. Este archivo no guarda ningún dato para visualización posterior al uso de esta opción. Tiene una capacidad de hasta 15 funciones en la misma pantalla reflejando con ésto el comportamiento sobrepuesto de estas funciones.

4.3.6. Sección de ayuda al usuario. Esta sección esta compuesta por un pequeño archivo, da ayuda para consulta del usuario al momento de que este lo solicite ya sea dentro de la sección de menú principal o en alguno de los dos editores que

comprende el sistema.

4.3.7. Sección de interacción con el sistema operativo. Esta sección comprende el manejo del MS-DOS dentro del sistema para depurar archivos dentro de algún disco, o consultar si hay en existencia algún archivo de datos o funciones. Se puede hacer cualquier cosa que contenga el sistema operativo que se este trabajando al momento de cargar o encender a la computadora; de la misma manera, se puede mandar a ejecutar a algún archivo que este en versión ejecutable ya sea .EXE, .COM o .BAT si es que se cuenta con el suficiente espacio de memoria disponible en RAM.

4.3.8. Opción de salida del sistema. Esta opción permite al usuario concluir la sesión en el sistema y regresar al sistema operativo (MS-DOS).

#### 4.4. Alcance y limitaciones del sistema.

##### 4.4.1. Limitaciones del sistema.

Debido a que el sistema requiere de todo el archivo de datos para efectuar sus modificaciones y cálculos, la capacidad de retención de la RAM restringe el tamaño de los archivos de datos. Actualmente una computadora de 640KB de RAM soporta archivos hasta de 3 vectores de 800 datos por vector.

En cuanto a la sección de graficación de funciones la velocidad del proceso dependerá de la cantidad de operaciones que se le ordene efectuar a la computadora. Así, si tenemos varias funciones y los rangos de variación de las mismas son amplios y además con incrementos en las variables, pequeños, el proceso será entonces muy tardado, sin embargo debe recordarse que la ganancia en detalle de la gráfica pudiera elevar al usuario a

decidirse por esta opción, tomando en cuenta que en otros tiempos y con los sistemas de cálculo manual tradicionales esto era una labor pocas veces alcanzada. Por ejemplo al elaborar una gráfica de dos dimensiones de 24 puntos de la manera tradicional (papel y lápiz) lleva aproximadamente de 15 a 45 minutos según la habilidad de la persona, en cambio usando el sistema SIDEVIS la creación de la misma gráfica toma de 5 a 10 minutos según la habilidad del usuario.

Por otro lado, dentro de la pantalla se tiene un espacio de graficación de 300x180 píxeles para la representación de los datos de alguna función o archivo de datos. Esto limita al usuario en cuanto a lo que es una definición super detallada. Sin embargo aún cuando la capacidad que tiene el sistema es suficiente para apreciar el comportamiento de algún conjunto de datos razonablemente grande, si se requiere de más detalle se puede fraccionar la función de manera que se vea en partes.

La última limitante a señalar, es que nos tenemos que restringir a tres colores o tonalidades (según sea el monitor usado) dentro de la pantalla, puesto que es la capacidad máxima permitida por el sistema.

#### 4.4.2. Alcances del sistema.

Con el sistema aquí presentado el usuario puede hacer simulaciones de modelos deterministas cuyos procesos se expresan en funciones matemáticas y pueda representar los resultados de la simulación gráficamente. Finalmente también da la opción de poder ver el comportamiento de una o más funciones dadas de manera gráfica, esta función es de gran utilidad para

la enseñanza de áreas como son las matemáticas, la física y la química, ayudando al estudiante a comprender los fenómenos y sus alteraciones causado por cambios a los parametros al ver su comportamiento gráfico.

## CONCLUSIONES.

Como resultado del proceso de integración de esta tesis, se obtuvo un sistema que permite aplicar una técnica que permite construir el modelo de una situación real acuada a la realización de experimentos de simulación con modelos simbólicos de tipo exclusivamente deterministas.

Se cumplió con el objetivo general planteado de crear un sistema computarizado de simulación con facilidades de graficación, con gráficos hasta de tres dimensiones, para representar el comportamiento de un modelo matemático simbólico determinista, el cual serviría de apoyo en instituciones de enseñanza e investigación.

Así mismo se cumplió con los objetivos específicos de:

1.- Crear un subsistema a través del cual se logre la captura de los datos requeridos para establecer la simulación y la graficación, para una, dos o tres variables, según sea el caso, así como del modelo matemático a simular. Una vez capturados estos datos, el subsistema realizará la simulación cargando los datos tanto en el UCP como en el disco y por último, llamará al subsistema de simulación y representación que se describe en el punto 2.

2.- Crear un subsistema que asimile los datos y los represente de manera gráfica, esto es, que dibuje los puntos que son representativos de los datos asimilados en un espacio definido en la pantalla, para después poder usar las distintas facetas constituidas por: imágenes fijas, imágenes con escalamiento,

imagenes con rotación, e imagenes con translación; teniendo como premisa el que este subsistema se constituya en la representación gráfica del proceso de tipo determinista que se vaya a cuestionar.

A los dos subsistemas descritos se les identificará como subsistema 1 y subsistema 2, respectivamente.

3.- Fusionar los subsistemas 1 y 2 para que den como resultado el enfoque dual que se requiere para la creación del sistema de simulación con graficación hasta de tres dimensiones que constituye al objetivo general de este trabajo de tesis.

Se creó por otra parte, un sistema que permite obtener gráficas de puntos, poligonales, líneas y barras de dos y tres dimensiones. Así mismo, un sistema que permite presentar a las gráficas de tres dimensiones en forma rotativa en ángulos de 30°, 40° u otro que el usuario considerará conveniente para sus necesidades de visualización. Un sistema que para el caso de tres dimensiones efectúa gráficas interactivas. Por otra parte se creó un sistema que presenta la opción de poder representar gráficamente el comportamiento de hasta 15 funciones en un campo visual de hasta tres dimensiones.

Asimismo el sistema por su estructura puede ser usado en equipos de alta difusión y de fácil acceso como son las microcomputadoras IBM-PC y compatibles.

Se logró disponer asimismo de un sistema sencillo, creado en un lenguaje estructurado de alta difusión y de gran versatilidad que permitiría, de así deseárselo complementarlo y ampliarlo cuanto



fuese necesario.

Finalmente estamos convencidos de que el sistema por su sencillez es susceptible de ser usado en los laboratorios de cómputo de las instituciones de enseñanza e investigación, así como en los negocios y creemos que aún en problemas relacionados con hogares. Es posible por otra parte su aplicación en los campos de las matemáticas, la física, la ingeniería y otras.

Sin embargo no se debe dejar de señalar que debido a la capacidad que presentan los equipos para los que fue diseñado el sistema, se tienen limitantes en cuanto a la resolución en pantalla y en el tiempo de ejecución de algún proceso, además de que sólo puede crear archivos de datos hasta de 800 valores por vector con un máximo de tres vectores, debido a que fue la máxima capacidad aceptada por una computadora de 640 Kb en un arreglo matricial, sin embargo, para los fines que pretende servir el sistema, dichas limitantes no son de consideración.

Por último, también es importante señalar que debido a la programación estructurada en que fue diseñado se le pueden agregar procedimientos o modificar los que comprenden al sistema, ya sea para establecer mejoras o particularizar el sistema hacia un problema o modelo en particular.

## BIBLIOGRAFIA.

- 1).- Autor: Thomas H. Naylor  
Título de obra: Técnicas de Simulación por Computadora.  
Español.  
Traducción:  
Número de Edición:  
País:  
Editorial:  
Año:  
Volumen:  
Páginas:
- 2).- Autor: Graciela Bueno Aguilar  
Título de obra: Apuntes de clase. Simulación de Sistemas. CEC. CP.  
Original. (español)  
Traducción:  
Número de Edición:  
País: México  
Editorial: Colegio de Postgraduados  
Año: 1989  
Volumen: Único  
Páginas:
- 3).- Autor: Geoffrey Gordon  
Título de obra: Simulación de Sistemas.  
Original. (español)  
Traducción:  
Número de Edición: Primera  
País: México  
Editorial: Diana  
Año: 1986  
Volumen: Único  
Páginas: 344
- 4).- Autor: Graciela Bueno  
Título de obra: Simulación Continua.  
Original. (español)  
Traducción:  
Número de Edición: Primera  
País: México  
Editorial: Colegio de Postgraduados  
Año: 1981  
Volumen: Único  
Páginas: 51
- 5).- Autor: J. Johnston  
Título de obra: Econometric Methods  
Traducción:  
Número de Edición: Original. (ingles)  
País: USA  
Editorial: McGraw-Hill  
Año: 1960  
Volumen: Único  
Páginas: 300

- 6).- Autor: Donald Hearn, M. Pauline Baker  
 Título de obra: Gráficas por Computadora.  
 Traducción: (español), Juan Carlos Vega F.  
 Número de Edición: Primera  
 País: México  
 Editorial: Prentice-Hall Hispanoamericana, S.A.  
 Año: 1988  
 Volumen: Único  
 Páginas: 380
- 7).- Autor: Marc Berger  
 Título de obra: Computer Graphics with Pascal.  
 Traducción: Original (ingles)  
 Número de Edición: Primera  
 País: USA  
 Editorial: The Benjamin / Cummings  
 Publishing Company, Inc.  
 Año: 1986  
 Volumen: Único  
 Páginas: 347
- 8).- Autor: David F. Rogers, J. Alan Adams  
 Título de obra: Mathematical Elements for  
 Computer Graphics.  
 Traducción: Original (ingles)  
 Número de Edición: Primera  
 País: USA  
 Editorial: McGraw-Hill Book Company  
 Año: 1976  
 Volumen: Único  
 Páginas: 239
- 9).- Autor: Yolanda Fernández Ordoñez  
 Título de obra: Graficación por Computadora.  
 Técnicas de Modelación.  
 Traducción: Original (español)  
 Número de Edición: Primera  
 País: México  
 Editorial: CEC. CP.  
 Año: 1988  
 Volumen: 7  
 Páginas: 110
- 10).- Autor: Owner's Handbook  
 Título de obra: Turbo Graphix Toolbox  
 Traducción: Original (ingles)  
 Número de Edición: Segunda  
 País: USA  
 Editorial: Borland International, Inc.  
 Año: 1985  
 Volumen: Único  
 Páginas: 257

- 11).- Autor: Elisa Viro Gurovich  
 Título de obra: Análisis de Lenguajes  
 Traducción: Original (español)  
 Número de Edición: Primera  
 País: México  
 Editorial: UNAM  
 Año: 1987  
 Volumen: Único  
 Páginas: 230
- 12).- Autor: Herbert Schildt  
 Título de obra: Programación y técnicas  
 Turbo Pascal Avanzado  
 Traducción: Original (español)  
 Número de Edición: Primera  
 País: España  
 Editorial: Osborne / McGraw-Hill  
 Año: 1987  
 Volumen: Único  
 Páginas: 258
- 13).- Autor: Borland International  
 Título de obra: Turbo Pascal Ver.5.0  
 User's Guide  
 Traducción: Original (inglés)  
 Número de Edición: Primera  
 País: USA  
 Editorial: Borland International, Inc.  
 Año: 1988  
 Volumen: Único  
 Páginas: 350

#### BIBLIOGRAFIA COMPLEMENTARIA

- 14).- Autor: Dimitris N. Chorafas  
 Título de obra: System and Simulation  
 Traducción: Original (inglés)  
 Número de Edición: Primera  
 País: USA  
 Volumen: Único
- 15).- Autor: Roy A. Plastock  
 Gordon Kalley  
 Título de obra: Gráficas por Computadora  
 Traducción: (español), María de Lourdes  
 Fournier  
 Número de Edición: Primera  
 País: USA  
 Editorial: (Serie SCHAUM) McGraw-Hill  
 Año: 1987  
 Volumen: Único  
 Páginas: 349

- 16).- Autor: Steven Harrington  
 Título de obra: Computer Graphics. A Programming Approach.  
 Traducción: Original (Inglés)  
 Número de Edición: Segunda  
 País: USA  
 Editorial: McGraw-Hill International Book Company  
 Año: 1983  
 Volumen: Único
- 17).- Autor: Robert J. Traister  
 Título de obra: Graphics Program for the IBM PC  
 Traducción: Original (Inglés)  
 Número de Edición: Primera  
 País: USA  
 Editorial: Tab Books Inc.  
 Año: 1983  
 Volumen: Único
- 18).- Autor: Victor Serrano Altamirano  
 Título de obra: VIC-SITEM Un sistema para calcular tamaños de muestra y estimadores en estudios por muestreo  
 Traducción: Original (español)  
 Número de Edición: Primera  
 País: México  
 Editorial: (TESIS) Colegio de Postgraduados  
 Año: 1989  
 Volumen: Único  
 Páginas: 73
- 19).- Autor: Concepción Cardoso Martínez  
 Título de obra: Angelica del R. Lozano Cuevas Proyecto de un laboratorio de cómputo  
 Traducción: Original (español)  
 Número de Edición: Primera  
 País: México  
 Editorial: UNAM, ENEP Acatlán (TESIS)  
 Año: 1989  
 Volumen: Único  
 Páginas: 247
- 20).- Autor: Gerardo Ayala San Martín  
 Título de obra: Introducción a la Computación  
 Traducción: Original (español)  
 Número de Edición: Primera  
 País: México  
 Editorial: Ed. Porrúa. UNAM.  
 Año: 1987  
 Volumen: Único  
 Páginas: 90

- 20).- Autor: Herbert Schildt  
 Título de obra: Lenguaje C  
 Programación Avanzada  
 Traducción: Original (español)  
 Número de Edición: Primera  
 País: España  
 Editorial: Osborne / McGraw-Hill  
 Año: 1987  
 Volumen: Único  
 Páginas: 299
- 21).- Autor: Peter Grogono  
 Título de obra: Programación en Pascal.  
 Traducción: (español), Gabriel Guerrero  
 Número de Edición: Primera  
 País: México  
 Editorial: Fondo Educativo Interamericano  
 Año: 1984  
 Volumen: Único  
 Páginas: 371
- 22).- Autor: Elliot B. Koffman  
 Título de obra: Pascal. Introducción al  
 lenguaje y resolución de  
 problemas con programación  
 estructurada.  
 Traducción: (español), Norma Gisela G.  
 Número de Edición: Primera  
 País: México  
 Editorial: Fondo Educativo Interamericano  
 Año: 1984  
 Volumen: Único  
 Páginas: 537
- 23).- Autor: Byron S. Gottfried  
 Título de obra: Programación en Pascal  
 Traducción: (español), Alfredo Bautista P.  
 Número de Edición: Primera  
 País: España  
 Editorial: McGraw-Hill de México  
 Año: 1986  
 Volumen: Único  
 Páginas: 398
- 24).- Autor: Kris Jansa, Steven Nameroff  
 Título de obra: Turbo Pascal Biblioteca de  
 Programas.  
 Traducción: (español), J.R.García B.  
 Número de Edición: Primera  
 País: España  
 Editorial: McGraw-Hill/Interamericana,S.A.  
 Año: 1988  
 Volumen: Único  
 Páginas: 448

- 25).- Autor: Borland International  
 Título de obra: Turbo Pascal Ver.5.0  
 Traducción: Original (ingles)  
 Número de Edición: Primera  
 País: USA  
 Editorial: Borland International, Inc.  
 Año: 1988  
 Volumen: Unico  
 Páginas: 493
- 26).- Autor: Owner's Handbook  
 Título de obra: Turbo Pascal Ver.4.0  
 Traducción: Original (ingles)  
 Número de Edición: Primera  
 País: USA  
 Editorial: Borland International, Inc.  
 Año: 1987  
 Volumen: Unico  
 Páginas: 655

SOFTWARE

- 27).- Autor: Borland International, Inc.  
 Título de obra: Turbo Pascal Ver.5.0  
 Traducción: Original (ingles)  
 Número de Edición: Primera  
 País: USA  
 Año: 1988
- 28).- Autor: Spring Board  
 Título de obra: News Room Pro Ver. 2.1  
 Traducción: Original (ingles)  
 Número de Edición: Primera  
 País: USA
- 29).- Autor: Word Star  
 Título de obra: Word Star Ver 4.0  
 Traducción: Original (ingles)  
 Número de Edición: Primera  
 País: USA