9

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

CAMPUS ACATLÁN

SIMULAT: UN LENGUAJE DE SIMULACIÓN PARA LÍNEAS DE ESPERA Y REDES

T E S I S
QUE PARA OBTENER EL TÍTULO DE :
LICENCIADO EN MATEMATICAS APLICADAS Y COMPUTACIÓN

P R E S E N T A :

FERNANDO GARCÍA MINQUINI

ASESORA:
ACT. MARÍA DEL CARMEN GONZÁLEZ VIDEGARAY

UNAM
CAMPUS ACATLÁN

OCTUBRE 2000

A mis padres: Fernando y Silvia;
A mis hermanos: Erika y Rogelio;
A mi esposa: Claudia;
A mi asesora y gran amiga MariCarmen;
A mis grandes amigas Bety, Coco, Lupita, Mayra, Paty, Ruth y Yola;
A mis mejores amigos Héctor, José Antonio, Pedro, Rubén, Wilfrido, Yola;
A todos aquellos que me han apoyado a lo largo de mis proyectos;
A todos ellos ... GRACIAS

INDICE GENERAL

INTRODUCCIÓN

En los países desarrollados día a día se buscan nuevas aplicaciones de las matemáticas a los diversos sectores de la industria, el comercio y medicina, debido a los constantes y crecientes requerimientos de la sociedad. En los últimos años, la investigación de operaciones ha resuelto una gran cantidad de los requerimientos que ya se mencionaron, pero existen otros donde la IO[1] es inútil o poco práctica, esto es cuando se manejan problemas donde los modelos son demasiado complejos y que no tienen solución analítica, para estos casos se ha desarrollado un método[2] llamado simulación.

Según Frederick S. Hillier y Gerald J. Lieberman, en su texto Introducción a la Investigación de Operaciones *"la simulación es una técnica de muestreo estadístico controlado para estimar el desempeño de sistemas estocásticos complejos cuando los modelos analíticos no son suficientes"*, de acuerdo con esto, la simulación debe de emplearse como un último recurso, debido a que existe un error aleatorio y el sistema no se comportará exactamente como el modelo simulado, pero sí la metodología de la simulación se realizó correctamente, el resultado se comportará como el sistema real.

Existen algunas desventajas de utilizar la simulación, entre ellas podemos contar las siguientes:

- ◊ pocas herramientas computacionales
- ◊ existencia de un error aleatorio
- ◊ cuando el modelo no está implementado, no se pueden validar los resultados

Pero, si existen tantos problemas con la simulación, ¿por qué hablar de esta?, ¿por qué realizar una tesis acerca de la simulación?, la respuesta es sencilla, existen una gran cantidad de problemas que se resuelven por medio de la simulación de modelos contándose entre estos: líneas de espera, procesos de Poisson, procesos de Markov, series de tiempo, procesos de producción, control de calidad, pronósticos, juegos de azar; entre otros, entonces como podemos ver, es necesario hablar de la simulación, que es un área de investigación tan amplia como interesante.

Entrando en materia, el presente trabajo tiene por objeto desarrollar un lenguaje de simulación, pero antes es necesario tener los conocimientos matemáticos mínimos necesarios para entender cómo se realiza el proceso de simulación y desarrollar adecuadamente los algoritmos, es por eso que este trabajo se desarrolló de la siguiente forma:

- ◊ En el primer capítulo se habla sobre terminología empleada en la simulación de sistemas, la metodología propuesta, en la cual describe los pasos a seguir, y la forma de validar los resultados de una simulación.

[1] Investigación de Operaciones.
[2] Los diversos autores difieren al nombrar a la simulación como técnica, método o proceso.

◊ El segundo capítulo trata sobre las diferentes formas que hay de generar series de números aleatorios[3], así como los algoritmos más frecuentemente utilizados para generar variables aleatorias.

◊ En el capítulo tres se muestra la forma en que se realizan las simulaciones de las diferentes funciones de probabilidad, aquí se muestran los procedimientos utilizados por SIMULAT.

◊ Para el capítulo cuatro, nos encontramos con el lenguaje resultante, se dan las características generales de SIMULAT, tales como requisitos de hardware, requisitos de software y las instrucciones que se emplean es este lenguaje.

◊ En el quinto capítulo habla sobre los alcances y límites que tiene SIMULAT, se mencionan también algunas características importantes acerca del lenguaje.

◊ Por último, en el capítulo seis, tenemos las aplicaciones de SIMULAT, solamente se mencionan algunas ya que es un tema bastante extenso.

---

[3] Conocidos como pseudoaleatorios.

CAPÍTULO 1
TERMINOLOGÍA Y METODOLOGÍA DE LA SIMULACIÓN, VALIDACIÓN DEL MODELO.

El ser humano constantemente toma decisiones para las actividades que realiza. Esto no es sencillo, pues se debe tomar una decisión acertada en la cual interviene un análisis, la evaluación de los datos obtenidos y con base en esto, se dará un pronóstico más aproximado de lo que puede ocurrir. Esta técnica es llamada *simulación* y es la que nos permite realizar un análisis matemático de la forma en que ocurre un fenómeno[1], esto nos proporcionará un modelo que nos ayude a entender sucesos presentes, prever[2] sucesos futuros y nos permita reproducir fenómenos del pasado.

El realizar una simulación implica ciertas ventajas como la reproducción de los eventos, con la posibilidad de cambiar las condiciones en que éstos se desarrollan, dando como resultado una visión más general de lo que puede ocurrir en diferentes momentos y tomar las decisiones en forma acertada con respecto al fenómeno.

1.1 Terminología.

Para entender lo que es la simulación, es necesario familiarizarse con algunos términos que frecuentemente son encontrados al utilizar esta técnica. A continuación se decriben los términos más importantes dentro de la simulación.

Sistema: es un conjunto de elementos que interactuan entre sí, para alcanzar un objetivo o bien común y es el objeto de estudio de la simulación.

Los sistemas se pueden clasificar en diversos tipos:

- ◊ *Determinísticos*: son sistemas que se pueden predecir con toda certeza.
- ◊ *Estocásticos*: son los sistemas que contienen eventos aleatorios o probabilísticos.
- ◊ *Estáticos:* son sistemas que no sufren cambios a través del tiempo.
- ◊ *Dinámicos:* con el tiempo, estos sistemas sufren variaciones.

La mayor parte de los sistemas que se encuentran en la vida real son dinámicos-estocásticos.

[1] Entiéndase por fenómeno cualquier actividad que realice el hombre o que ocurra en la naturaleza.
[2] La simulación reproduce eventos en forma semejante ó aproximada a la realidad.

Otro término muy utilizado es *estado de sistema* y lo podemos definir como la totalidad de las características relevantes del sistema. Por lo general, un sistema puede caracterizarse por un conjunto de atributos de interés, y el estado del sistema en un momento será el valor particular de cada uno de ellos.

Ahora bien, si cada uno de los atributos del sistema puede ser medido o cuantificado, entonces puede usarse una variable única que lo represente, nombrándose *variable de estado.* Si un sistema tiene *m* variables de estado, estas pueden ser expresadas por (s_1, s_2,...,s_m) o en forma vectorial ***S***.

En ocasiones, existen variables cuyos valores pueden ser definidos por el analista al inicio del problema, independientemente de otras consideraciones. . Estas son llamadas *variables de decisión.* Pueden representarse como (x_1, x_2, ... , x_n) o en su forma vectorial ***X***. Normalmente estas variables afectan el estado del sistema y son independientes de él. Un conjunto de variables de decisión asociado a un conjunto de variables de estado, forma una *política de operación.*

Las relaciones que describen la interacción de las variables de estado, variables de decisión y los parámetros del sistema son llamadas *relaciones de causa-efecto* y estas son leyes físicas, económicas, estadísticas que generalmente son representadas por funciones matemáticas.

La forma de relacionar todos los elementos anteriores es a través de un *modelo*, que es una representación de la realidad.

Los modelos pueden ser clasificados en los siguientes tipos:

◊ *Modelos Abstractos*: Son los que contienen relaciones y ecuaciones matemáticas.

◊ *Modelos Iconográficos*: Son los que representan objetos físicos a escala.

◊ *Modelos Visuales*: Son representaciones gráficas

A su vez los modelos abstractos, que son nuestro objeto de estudio, se pueden subdividir según su uso:

◊ Por su objetivo:
- *Descriptivos*: Muestran el comportamiento del fenómeno.
- *Explicativos*: Relacionan el comportamiento causa-efecto.
- *De pronóstico*: Sirven para predecir el comportamiento del fenómeno a futuro en base a su historia[3] o en circunstancias diferentes a las actuales.
- *De optimización*: Cuando el objetivo es encontrar el mejor valor de una función.

[3] Llamada series de tiempo.

- *De control*: Sirve para mantener el fenómeno dentro de ciertos límites.

◊ Por el tipo de análisis:
- *Discretos/Continuos:* Los modelos discretos utilizan ciertos valores y con límites determinados, los modelos continuos utilizan valores reales.
- *Estáticos/Dinámicos:* Se les llama así porque dependen de la variable tiempo.
- *Estocásticos/Determinísticos:* Son dependiendo del tipo de variables que manejen, las variables aleatorias se usan en los modelos estocásticos, mientras los modelos determinísticos se predicen con toda certeza.

◊ Por la generalidad de su aplicación:
- *Pre-costruidos*
- *A la medida*

El analista[4] es quien toma la decisión de qué tan simple o complejo será el modelo a utilizar, de acuerdo a sus necesidades, tomando en cuenta que el modelo resultante tiene que ser más simple que el sistema real.

1.2 Metodología.

Para realizar un experimento de simulación se tiene que seguir toda una metodología, así los resultados obtenidos serán más confiables y se tomarán mejores decisiones.
La metodología que en el presente trabajo se sugiere para preparar un experimento de simulación es la siguiente:

1. *Formulación del problema.* Es la parte donde se establecen los objetivos de la investigación. A partir de observar un sistema se determinan su comportamiento y sus necesidades, esto no es sencillo en la práctica, ya que la problemática debe ser detallada correctamente para el siguiente paso lo cual implica excluir muchas cosas y dejar sólo algunas.

2. *Conceptualización del modelo.* Es donde se asocia una descripción matemática al sistema, se debe especificar el objetivo del modelo y las restricciones del mismo.

3. *Obtención y procesamiento de los datos.* Antes de poder formular un modelo matemático, es necesario conocer un poco más del comportamiento del sistema, la mejor manera de hacer ésto es tomando muestras para realizar un estudio. Para la recolección de los datos, generalmente, es necesario el trabajo de un equipo interdisciplinario.

[4] Analista es la persona que está realizando el análisis del sistema a simular.

4. *Formulación del modelo matemático.* Es la realización del modelo matemático, aquí se decide cuántas variables se utilizarán, de qué tipo van a ser, la complejidad del modelo para su programación. Para formular el modelo es necesario diseñarlo. Existen dos tipos de diseños para modelos:

   ◊ *Diseños generalizados*: Que describen el funcionamiento de un sistema completo.
   ◊ *Diseños modulares o en bloque*: Intentan simplificar un modelo generalizado por varios modelos, los cuales describen partes importantes del sistema.

5. *Estimación de los parámetros.* Ya con muestras de datos y con un modelo, se tienen que estimar los parámetros necesarios y ver qué tan significativos son. Esto se logra por medio de pruebas estadísticas[5] y estimación por mínimos cuadrados. Cuando estos métodos no pueden ser utilizados se usan métodos de tipo iterativo para encontrar los estimadores apropiados.

6. *Implementación del modelo.* Es la etapa donde se programa el modelo en la computadora, se decide qué tipo de lenguaje, de propósito específico o general, según las ventajas de cada uno de ellos[6] y las características del modelo a simular.

7. *Evaluación del modelo.* Aquí se recolectan los datos producidos por el modelo y se obtienen los parámetros del mismo tales como medias y varianzas, que al utilizar pruebas estadísticas se evalúa la significancia de cada uno de ellos. Es importante mencionar que es necesario realizar pruebas que permitan detectar violaciones a suposiciones fundamentales del modelo.

8. *Validación de los resultados.* En esta etapa se hace un estudio de los datos obtenidos, con alguno de los métodos de validación, del modelo de simulación para saber que tan confiable es este.

9. *Diseño del experimento de simulación.* Ya teniendo un modelo de simulación válido se diseñan los experimentos a realizar, tomando en cuenta su importancia, orden, alternativas y combinaciones de los mismos.

10. *Análisis de los resultados.* Aquí se realiza la interpretación y el análisis de los resultados obtenidos del modelo experimental, esto se puede reducir a tres pasos:
   ◊ Recolección y proceso de los datos simulados.
   ◊ Análisis estadístico.
   ◊ Interpretación de resultados.

---

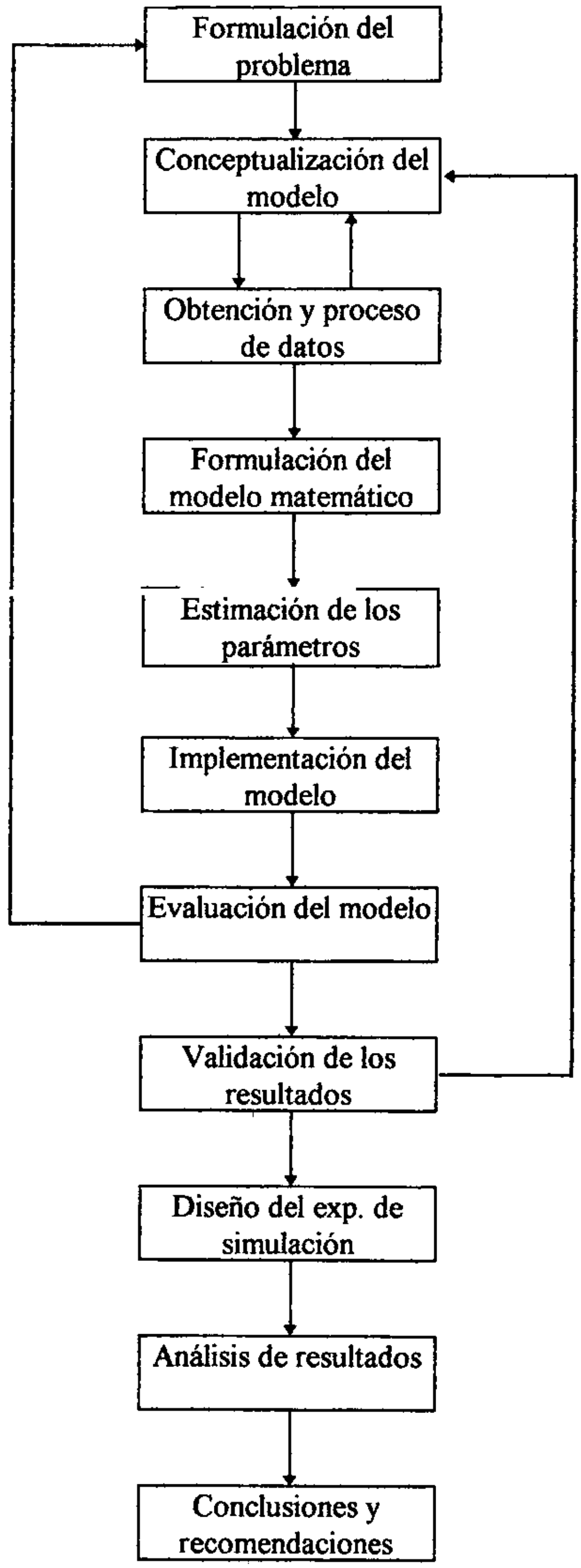
[5] Generalmente se utilizan los métodos de Máxima verosimilitud y de Momentos.

[6] Los lenguajes de uso específico tienen algunas ventajas que es conveniente mencionar: facilidad para la descripción del modelo, movimiento automático del modelo a través del tiempo, obtención automática de reportes, generación de procesos probabilísticos. Algunas de sus desventajas son: costo elevado, puede ser desconocido para el programador, se puede elegir mal el leguaje y no ser el apropiado para el tipo de modelo, altos costos en tiempo para ejecutar un programa en un lenguaje de uso específico, dificultad para validar el modelo por el nivel de abstracción de los lenguajes.

11. *Conclusiones y recomendaciones.* Después de interpretar los resultados, se hacen las recomendaciones pertinentes y se toman las decisiones para obtener la mejor política de operación del sistema.

Estos pasos no son necesariamente secuenciales, pueden ser iterativos, ya que algunos pasos se repetirán continuamente hasta obtener el modelo buscado, esto se presente en el siguiente diagrama de flujo:

DIAGRAMA DE FLUJO PARA UN MODELO DE SIMULACIÓN



1.3 Validación.

El analista debe decidir qué aspectos del mundo real se incluirán en el modelo y cuáles deben de ser ignorados, por esta razón todo modelo debe de ser validado para revisar que se hayan contemplado los aspectos con mayor importancia del sistema, para que la información obtenida de los experimentos sea confiable y estar seguros de tomar las decisiones adecuadas.

Para realizar la validación de un modelo de simulación existen varios métodos, pero son cuatro los más comunes:

- ◊ Comparación de los resultados con el sistema real.
- ◊ El método de Delphi.
- ◊ La prueba de Turing.
- ◊ La conducta en casos extremos.

Comparación de los resultados con el sistema real.
Consiste en comparar las medidas equivalentes del modelo de simulación con el sistema real, para esto se utilizan pruebas de igualdad de medias, varianzas y algunas pruebas no paramétricas. El mayor inconveniente de este método es el muestreo[7], ya que no siempre se pueden obtener datos del sistema real[8].

Método Delphi.
Esta prueba fue desarrollada para obtener una aproximación a la solución de problemas con poca información cuantitativa. Para realizar esta prueba se selecciona a un grupo de *expertos*[9] para realizar un consenso sobre los datos analizada. Lo importante de este método es que los expertos no discuten el problema en grupo, se les pide que contesten un cuestionario que contiene las preguntas necesarias para la validación, utilizando la información obtenida del cuestionario, se formulan preguntas más específicas.

Prueba de Turing.
Para realizar esta prueba se reúne a un grupo de expertos en el sistema real y se les entrega un reporte con la descripción del sistema real y el sistema simulado, sin aclarar cuál es cuál, si este grupo de expertos no distingue cuál es el reporte del sistema real del obtenido en la simulación, entonces se puede tomar el modelo de simulación como válido.

[7] Hay que recordar que para que un muestreo sea correcto debe ser lo suficientemente grande y representativo de la población que se estudia.
[8] Muchas veces el sistema real no existe, las condiciones del experimento no se han dado en la realidad, porque no se cuenta con la información suficiente o no ha transcurrido el tiempo suficiente.
[9] Entiéndase por *expertos* a los usuarios y administradores del sistema que, por el tiempo que han estado involucrados con él, lo conocen a profundidad.

La conducta en los casos extremos.
En esta prueba se simulan los casos extremos[10] del sistema en el modelo, generalmente las situaciones reales son fácilmente predecibles, si su comportamiento es parecido al comportamiento del sistema real el modelo puede ser válido. En este caso se recomienda utilizar otra prueba para verificar la validez del modelo.

[10] Son los casos en que las condiciones del sistema se exageran al máximo o al mínimo.

CAPÍTULO 2
GENERADORES DE NÚMEROS Y VARIABLES ALEATORIAS.

Para que todo experimento de simulación sea considerado como tal debe tener dos aspectos muy importantes: ser estocástico[1] y ser reproducible, para modificar las condiciones del sistema.

Para que un experimento se considere estocástico deben emplearse números aleatorios (o pseudoaleatorios) y para reproducir un experimento es necesario reproducir[2] los números aleatorios que fueron utilizados para llevarlo a cabo la primera vez que se realizó, esto sólo es posible utilizando un método que genere los números aleatorios en el mismo orden para otro experimento. Esto es con el fin de poder determinar qué ocurriría si se presentan diferentes fenómenos en el experimento . El hecho de producir los números aleatorios con un algoritmo[3] ha llevado a diversos autores a manejar el término pseudoaleatorios.

2.1 Generadores de números aleatorios.

Los algoritmos que producen los números pseudoaleatorios son llamados generadores y para recibir ese nombre, deben de cumplir con algunas características que permitan que los números obtenidos se puedan considerar como aleatorios:

- ◊ Ser reproducibles.
- ◊ Los números pseudoaleatorios deben ser independientes uno de otro y provenir de una distribución de probabilidad teórica o empírica.
- ◊ Ser rápidos.
- ◊ Que su período[4] de repetición sea grande para que los números producidos sean suficientes antes de que se repita el ciclo.
- ◊ No ser degenerativo, es decir, no debe de repetir un mismo número en forma indefinida.

Existen diversos métodos para generar números pseudoaleatorios, estos métodos se enumeran de la siguiente forma:

- ◊ Manuales.
- ◊ Tablas.

---

[1] Aleatorio.
[2] Volver a generar.
[3] Al seguir un algoritmo para generar los números aleatorios, se pierde la aleatoriedad y los números son completamente determinísticos. Estos algoritmos siempre inician con un valor $X_0 = r$, donde r es llamada semilla.
[4] Ciclo.

◊ Métodos para computadora digital.

Los métodos manuales son utilizados de manera didáctica, pero en la práctica no son útiles ya que son lentos, y no son reproducibles.

Las tablas son útiles en simulaciones pequeñas, pero son lentas y no son utilizadas en las computadoras ya que ocuparían demasiado espacio en su almacenamiento.

Los métodos para las computadoras digitales son más prácticos ya que aparte de las características ya mencionadas hay que agregarles el hecho de que deben de utilizar la menor cantidad de tiempo y memoria de almacenamiento. Se han desarrollado distintos generadores:

◊ Cuadrados medios.
◊ Congruencial multiplicativo.
◊ Congruencial mixto.

2.1.1 Método de cuadrados medios.

Este método ya no se utiliza mas que como referencia histórica y consta de los siguientes pasos:

1. Se selecciona un número de *n* dígitos, donde *n* es par.
2. Se eleva al cuadrado el número anterior y se agregan ceros a la izquierda hasta obtener un número de *2n* dígitos.
3. Se toman los *n* dígitos centrales del número obtenido en el paso 2, que es el número aleatorio generado.
4. Repetir el paso 2 con el número obtenido en 3.

Como se ve, es muy sencillo, aunque más adelante se verán sus desventajas. A continuación se presenta un ejemplo:

1. n=2, y X_0=37:

0	0	3	7

2.

1	3	6	9

3. X_1= 36

Si se continúa obtenemos los siguientes valores:

$X_2 = 29$
$X_3 = 84$
$X_4 = 05$
$X_5 = 02$
$X_6 = 00$
$X_7 = 00$

Como se aprecia el método es degenerativo[5], lento, y tiene un período muy corto por lo que es un mal generador de números aleatorios.

2.1.2 Método congruencial multiplicativo.

El método congruencial se basa en la relación de congruencia (≡), definida como:

$$r_i \equiv (ar_{i-1}) \bmod m$$

donde:

r_0 es la semilla
a es el multiplicador
m es el módulo

Es importante mencionar que todos los métodos congruenciales caen finalmente en un ciclo, pero ese ciclo dependerá de los valores que se asignen a los elementos anteriores.

La mejor manera de asignar los valores es la siguiente:

1. Se hace el modulo m lo más grande posible, generalmente se asigna el valor $m = 2^b$, donde b es el número de bits por palabra[6], esto provoca la optimización de los recursos cómputacionales.

2. El multiplicador a se elige de manera que se minimicen las correlaciones entre los valores sucesivos de r, y se obtenga un período lo más largo posible, por lo que se hace:

$$a \approx 2^{\left(\frac{b-1}{2}\right)}$$
$$a \equiv \pm 3 \bmod 8$$

5 Se dice que un método es degenerativo si en algún momento genera un número indefinidamente.
[6] Sin incluir el bit de signo.

3. El período obtenido es: $P = 2^{b-2} = \frac{m}{4}$.

4. Se asigna una semilla arbitraria.

5. Para obtener una distribución uniforme continua entre $(0,1)$, sólo se divide el valor r entre el módulo:

$$u = \frac{r}{m}$$

2.1.3 Método congruencial mixto.

El método congruencial mixto[7] se basa en la relación de congruencia:

$$r_i = \left(ar_{i-1} + c\right) \bmod m$$

donde:

r_0	es la semilla
a	es el multiplicador
c	es el aditivo o sumando
m	es el modulo

El período de este método es de m por lo que se le conoce también como el *método de período completo.*

Los pasos para que el ciclo sea completo son;

1. $m = 2^b$
2. $a \approx 2^{\frac{b+1}{2}}$
   $a \equiv 1 \bmod 4$

4. Se asigna una semilla y la constante aditiva arbitrariamente[8].

5. Para obtener una distribución uniforme continua entre (0,1), sólo se divide el valor r entre el modulo:

[7] Se llama mixto porque usa suma y producto.

[8] Generalmente son valores impares y menores que m.

$$u = \frac{r}{m}$$

2.1.4 Pruebas estadísticas de uniformidad.

En el área de las matemáticas no se puede dejar ningún cabo suelto[9], por tal motivo, se realizan diversas pruebas estadísticas para verificar que los datos obtenidos de un generador de números aleatorios sean realmente aleatorios y uniformes.

La prueba que más frecuentemente se utiliza es la prueba de bondad de ajuste Ji-Cuadrada; para llevar a cabo esta prueba se requieren de al menos 50 observaciones y al menos 5 de ellas en cada clase.

La hipótesis nula que se desea probar en este caso es:

> *Ho*: Los resultados del generador provienen de una población con distribución uniforme, entre cero y uno.

Contra la hipótesis alternativa:

> Ha: Los resultados pertenecen a cualquier otra distribución.

Se divide el intervalo $(0,1)$ en n subintervalos de misma longitud[10] y se comparan las frecuencias obtenidas u observadas con las frecuencias esperadas[11]. Para esto se utiliza el *estadístico de prueba*[12]:

$$\chi^2 = \sum_{i=1}^{n} \frac{(FO_i - FE_i)^2}{FE_i}$$

donde:

FO_i = Frecuencia observada en la clase *i*.
FE_i = Frecuencia esperada en la clase *i*.
n = Número de clases.

El resultado obtenido del estadístico se compara con el valor en tablas de una Ji-Cuadrada con n-1 g.l.[13] y un nivel de significancia α.

---

[9] Todo se debe de demostrar, para que exista la seguridad de lo que se está trabajando con los datos correctos.
[10] Las clases que se van a analizar deben ser mutuamente exclusivos y colectivamente exhaustivos.
[11] La frecuencia esperada es el número de observaciones que pertenecen teóricamente a cada clase, en la distribución uniforme es: datos/n.
[12] Es el valor obtenido de la muestra que permite establecer la comparación de las frecuencias observadas y las frecuencias esperadas.
[13] Grados de libertad.

La regla de decisión es: Si el estadístico de prueba es menor que el valor en tablas entonces se acepta la hipótesis nula.

Otra prueba estadística para el ajuste de los datos es la prueba de Kolmogorov-Smirnov, que permite evaluar la hipótesis de que una muestra de datos fue tomada de una distribución teórica F continua. Esta prueba es no-paramétrica y exacta para todos los tamaños de la muestra.

El procedimiento para realizar la prueba de Kolmogorov-Smirnov a la distribución uniforme continua es como sigue:

1. Generar los n números aleatorios.
2. Ordenarlos en forma ascendente.
3. Calcular la frecuencia relativa acumulada con la expresión:

$$F_n(x) = \frac{i}{n}$$

donde i es la posición de el número x_i del vector obtenido en 2.

4. Calcular el estadístico de Kolmogorov-Smirnov utilizando las ecuaciones:

$$D^+ = max\left\{\frac{i}{n} - F(x_i)\right\}$$

$$D^- = min\left\{F(x_i) - \frac{(1-i)}{n}\right\}$$

$$D = max\left\{D^+, D^-\right\}$$

5. Si $D < d_{\alpha,n}$ (que es el valor en tablas) no se rechaza la hipótesis nula de que los números generados siguen una distribución uniforme.

2.2 Generadores de variables aleatorias.

Así como existen métodos para generar números aleatorios uniformes entre cero y uno también existen métodos para generar *variables aleatorias*[14] con cualquier tipo de distribución, ya que los problemas comúnmente encontrados no utilizan únicamente la distribución uniforme. Los siguientes son los métodos más comunes:

[14] Una variable aleatoria es una función que está definida en S y toma valores en $\Re$.

◊ Método de la transformada inversa.
◊ Método de composición.
◊ Método de convolución
◊ Método de rechazo.
◊ Método de simulación directa.
◊ Método para distribuciones empíricas.

Cada uno de los métodos anteriores tiene sus propias características, ventajas y limitaciones, por lo que cada variable aleatoria se genera con un método específico.

2.2.1 Método de la transformada inversa.

En este método se utiliza la *distribución de probabilidad acumulativa*[15] F(x). Esta función es continua y está contenida en el intervalo $(0,1)$[16].

Este método consta de los siguientes pasos:

1. Se construye la función de distribución acumulativa:

$$F(x) = \sum_{-\infty}^{x} p(X) \text{ }^{17}$$

$$F(x) = \int_{-\infty}^{x} f(X)dX \text{ }^{18}$$

2. Se obtiene un número aleatorio r con distribución uniforme en el intervalo $(0,1)$, utilizando alguno de los métodos descritos anteriormente (se sugiere el método congruencial multiplicativo).
3. Se iguala el número aleatorio a la función: $r = F(x)$
4. Se despeja en valor de la variable: $x = F^{-1}(r)$

Algunas de las distribuciones que se pueden simular por este método son[19]:

◊ Distribución Uniforme

[15] Es la función $F:\Re \rightarrow \Re$ definida por: $F(a) = P(X \leq a)$.
[16] $0 \leq F(x) \leq 1$.
[17] $P(X \leq x)$, para X discreta.
[18] $P(X \leq x)$, para X continua.
[19] En el Capítulo 3 se explicará cómo se obtienen las simulaciones.

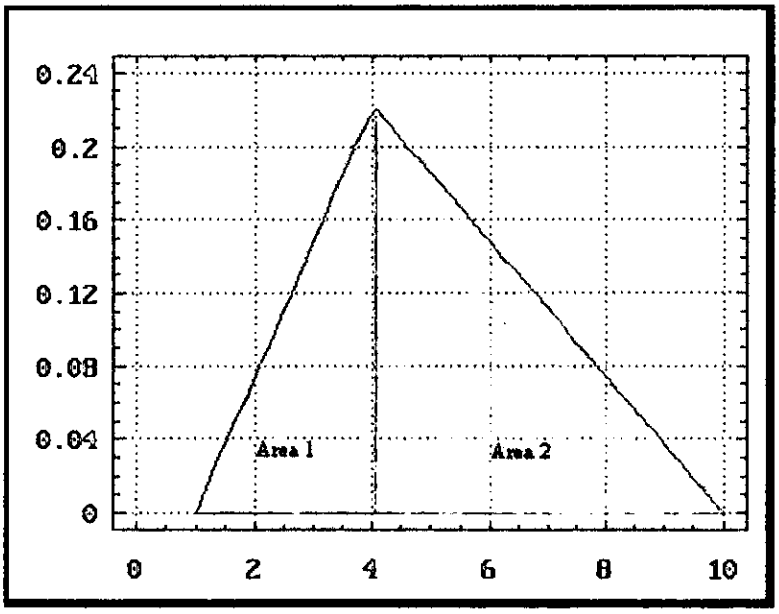
◊ Distribución Exponencial Negativa
◊ Distribución Triangular.

2.2.2 Método de composición.

Este método expresa la distribución de probabilidad *f(x)* como una composición de varias distribuciones de probabilidad $f_i(x)$, las cuales son seleccionadas con el objeto de minimizar el tiempo de generación de variables aleatorias.

Este método consta de los siguientes pasos:

1. Dividir la distribución de probabilidad original en *i* sub-áreas.



2. Definir una distribución de probabilidad $f_i(x)$ para cada sub-área[20].
3. Expresar la distribución de probabilidad con la ecuación:

$$f(x) = A_1 f_1(x) + A_2 f_2(x) + \cdots + A_i f_i(x)$$

4. Obtener la distribución acumulada de áreas.

---

[20] En el caso de rectas se utiliza la fórmula: $Y - y_0 = \dfrac{y_0 - y_1}{x_0 - x_1}(X - x_0)$, donde (x_0, y_0) es el punto inicial y (x_1, y_1) es el punto final.

$$\sum_{i=1}^{n} A_i f_i(x) = 1$$

5. Generar r_1 con distribución uniforme.
6. Con el método de la transformada inversa, obtener F_i, y de acuerdo a las A_i[21] elegir la F_i adecuada.
7. Sustituir r_1 en F_i.

Este método es muy útil en las siguientes distribuciones:

- ◊ Distribución Triangular.
- ◊ Distribución Trapecial.
- ◊ Distribución Doble exponencial[22].

2.2.3 Método de convolución.

Este método genera variables aleatorias (*X*), que se obtienen de la suma de *m* variables aleatorias (*Y*), que siguen una distribución específica. Esto se representa por:

$$X = Y_1 + Y_2 + \cdots + Y_m$$

El algoritmo para este método es intuitivo:

1. Generar las Y_m variables aleatorias con distribución G[23].
2. Se realiza la suma para obtener X.

Las distribuciones que se pueden simular por este método son:

- ◊ Distribución Gamma.
- ◊ Distribución Erlang.
- ◊ Distribución Normal Estandar.

2.2.4 Método de aceptación-rechazo.

Este método es también llamado *directo* y se puede utilizar con cualquier distribución, se requiere que la función de densidad *f(x)* sea continua y que se pueda acotar en un intervalo

[21] Esto se obtiene de la regla de decisión: $A_{i-1} \leq r < A_i$

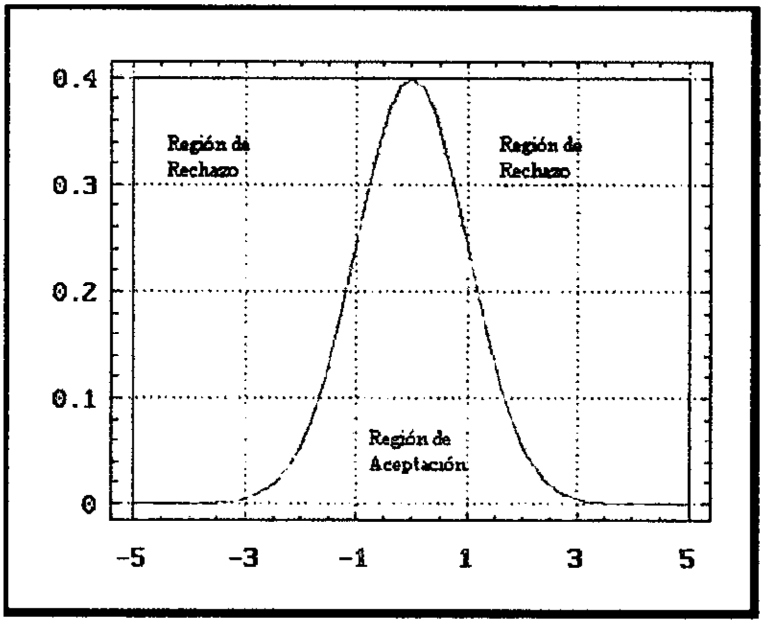
[22] También conocida como distribución de Laplace: $ce^{x} + ce^{-x}$; donde *c* es una constante.

[23] *G* es cualquier distribución.

$[a,b]$[24]. Generalmente se usa cuando otros métodos fallan o son ineficientes, no se usa como primera opción debido a el tiempo excesivo que se requiere para procesar datos.

Los pasos a seguir en este algoritmo son:

1. Encerrar en un rectángulo la distribución de probabilidad. En el caso de que sea asintótica en el eje de las abscisas[25], se trunca en un punto adecuado, generalmente se trunca en $\mu \pm 2\sigma$.



2. Se determina la altura máxima del rectángulo[26], *M*, que es conocida como probabilidad de la moda.
3. Se generan dos números aleatorios con distribución uniforme, entre 0 y 1, denominados r_1 y r_2.
4. Para asegurar que el valor x se encuentra dentro de el intervalo $[a,b]$ se hace la siguiente relación:

$$x = a + (b - a)r_1$$

5. Evaluar $f(x)$ con el valor de x obtenido en el paso anterior.
6. Se acepta x como variable aleatoria con distribución G si se cumple la siguiente relación[27]:

---

[24] $0 \leq f(x) \leq f_{max}$ y que $a \leq x \leq b$.

[25] Eje X.

[26] El rectángulo tiene la altura máxima de la curva.

[27] G es cualquier distribución.

$$r_2 = \frac{f(x)}{M}$$

Debido a que la probabilidad de rechazar a x es alta, este método es lento en la generación de variables.

2.2.5 Método de simulación directa.

En este método se obtiene una expresión analítica para la función de distribución acumulativa y se resuelve la expresión en forma explícita para x.
También se pueden utilizar las características propias de cada distribución para generar las variables aleatorias.

Las distribuciones que comúnmente se simulan con este método son:

- ◊ Distribución Uniforme Discreta
- ◊ Distribución Bernoulli
- ◊ Distribución Binomial
- ◊ Distribución Geométrica
- ◊ Distribución Binomial Negativa
- ◊ Distribución Poisson
- ◊ Distribución Hipergeométrica
- ◊ Distribución Normal

2.2.6 Distribuciones empíricas.

Existen dos tipos de distribuciones empíricas:

- ◊ Distribución empírica discreta.
- ◊ Distribución empírica continua.

Para simular la distribución empírica discreta, se asocian valores aleatorios enteros con las posibles variables, de modo que cada número aleatorio asignado a cada variable sea proporcional a la probabilidad de ocurrencia.

Ahora, para simular una distribución empírica continua es necesario aproximar la distribución con un número finito de puntos discretos[28] y realizar una interpolación lineal inversa.

Los pasos para realizar esta simulación son:

1. Obtener la función acumulativa de la distribución, suavizando con puntos discretos.
2. Obtener un r_1 aleatorio uniforme entre 0 y 1.

[28] Entre mayor sea el número de puntos discretos, mejor será la aproximación.

3. De acuerdo a r_1 se obtienen x_1 y x_2: $x_1 \leq r_1 \leq x_2$
4. Sustituir r_1 en la fórmula[29]:

$$x = \frac{(x_2 - x_1)(r_1 - F(x_1))}{F(x_2) - F(x_1)} + x_1$$

5. El valor x es la variable aleatoria buscada.

Este método de simulación es efectivo cuando la distribución de los datos no pertenece a alguna distribución teórica de probabilidad. Para comprobar que una muestra pertenece a una distribución de probabilidad se realiza la prueba de hipótesis que se mencionó anteriormente.

Es recomendable que se utilicen las distribuciones teóricas de probabilidad siempre que sea posible, ya que con las distribuciones empíricas no se contemplarán algunos valores que sean posibles, por ejemplo:

Si se tira un dado 8 veces y se obtiene la siguiente secuencia de números: 1, 6, 4, 3, 6, 2, 6, 1. Con los datos anteriores obtenemos las siguientes probabilidades:

Número	Prob.
1	2/8
2	1/8
3	1/8
4	1/8
5	0
6	3/8

[29] Esta ecuación se obtiene de la ecuación de la recta, haciendo las sustituciones y despejes necesarios.

CAPÍTULO 3
SIMULACIÓN DISCRETA Y SIMULACIÓN CONTINUA.

Ya que se conocen los métodos de generación de números y variables aleatorias, es necesario saber como se aplican esos métodos para realizar la simulación de las distribuciones teóricas de probabilidad. Cada distribución tiene un método adecuado para su simulación, que depende de las características propias de cada distribución[1].

Dado que el presente trabajo no pretende dar una gran explicación acerca de cada distribución, solamente se mencionan algunas características de las distribuciones y con que método se simulan. Es necesario hacer mención de que algunos de los generadores cambian según los valores que tengan los parámetros que contenga cada distribución y solo se menciona el método de simulación.

3.1 Simulación discreta.

Antes de describir los métodos de simulación de distribuciones discretas, se tiene que conocer lo que es una función discreta.

Una función discreta es aquella que contiene en su dominio un número finito de puntos acotados dentro de un intervalo. La representación de estos valores es:

$$S = \{x_1, x_2, \ldots, x_n\}$$

Donde S es el espacio muestral de la distribución.

Las distribuciones de probabilidad discretas sirven para procesos de conteo sobre muestras finitas o infinitas, donde la presencia o ausencia de un atributo es aleatoria.

Las distribuciones discretas más utilizadas son:

- ◊ D. Uniforme
- ◊ D. Bernoulli
- ◊ D. Binomial
- ◊ D. Geométrica
- ◊ D. Binomial Negativa
- ◊ D. Poisson

[1] El rango, los parámetros y los estimadores.

3.1.1 Distribución Uniforme.

Se utiliza esta distribución cuando una variable aleatoria puede tomar valores dentro de un intervalo $[a,b]$ con idénticas probabilidades decir si existen n puntos dentro del intervalo, la probabilidad de cada uno de ellos tiene la siguiente definición:

$$p(x) = \frac{1}{n}$$

Los parámetros de esta distribución son a y b, donde a es el parámetro inicial de la distribución también conocido como de localización y b es el parámetro final, un parámetro adicional a esta distribución es el parámetro de escala que se obtiene de $b-a$

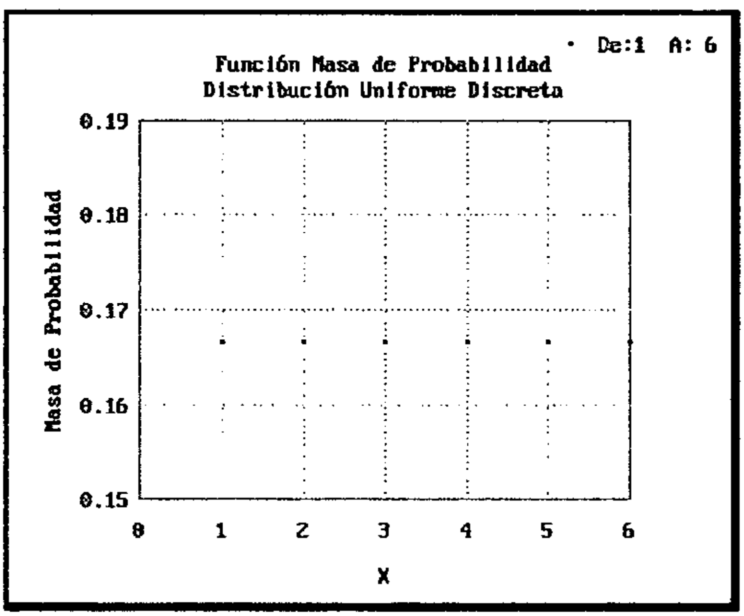
El rango de la distribución uniforme es: $S = \{a, a+1, a+2, \ldots, b\}$.

En ocasiones es necesario conocer la media[2] y la varianza[3] de las distribuciones, las fórmulas para obtener estas medidas son:

$$\mu = \frac{a+b}{2}$$

$$\sigma^2 = \frac{\left[(b-a+1)^2 - 1\right]}{12}$$

La gráfica de esta distribución es:



[2] La media es una medida de tendencia central.

[3] La varianza es una medida de dispersión.

El método de simular esta distribución es por el método de simulación directa, el algoritmo para simularla es el siguiente:

1. Se genera un número aleatorio *r*.
2. Se busca el intervalo al que pertenece *r*, es decir, $f_{i-1}(x) < r \le f_i(x)$.
3. El valor buscado es x_i.

3.1.2 Distribución Bernoulli.

Esta distribución se utiliza cuando existen dos posibles resultados, que se denotaran 0 y 1, y tienen diferente probabilidad de ocurrir, también es utilizada para generar otras variables aleatorias discretas:

◊ Distribución Binomial
◊ Distribución Geométrica
◊ Distribución Binomial negativa

Su función de probabilidad esta definida por:

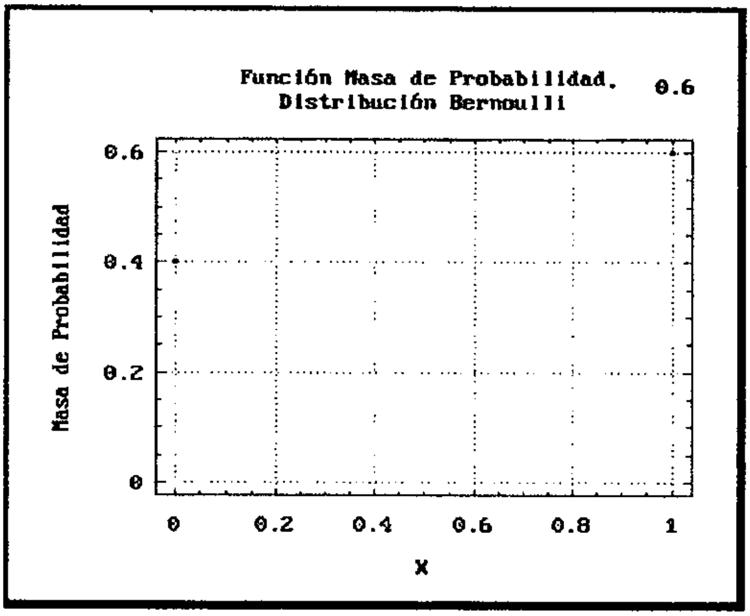
$$p(x) = \begin{cases} p \\ p-1 \end{cases}$$

Donde p es el único parámetro y representa la probabilidad de que ocurra 1, y p-1 es la probabilidad de que ocurra 0. Cabe mencionar que p está contenido dentro del intervalo $(0,1)$.

El rango de esta distribución sólo cuenta con 2 elementos: $S = \{0,1\}$.

Su media está definida por: $\mu = p$. Mientras que su varianza es definida por $\sigma^2 = p(1-p)$.

La gráfica de esta distribución es:



La simulación de esta distribución también es por medio de simulación directa:

1. Generar un r con distribución uniforme continuo.
2. Evaluar la regla de decisión: $r \leq f_1(x)$.
3. Si se cumple entonces es 1. En caso contrario entonces es 0.

3.1.3 Distribución Binomial.

Esta distribución describe el resultado de una colección de ensayos Bernoulli independientes entre sí.

Su función de probabilidad es definida por:

$$p(x) = \frac{n!\, p^x (1-p)^{n-x}}{x!(n-x)!}$$

Esta función tiene dos parámetros: p que es la probabilidad de que ocurra 1 y está contenido en el intervalo $(0,1)$, el otro parámetro es n que es el número de ensayos Bernoulli a realizar.

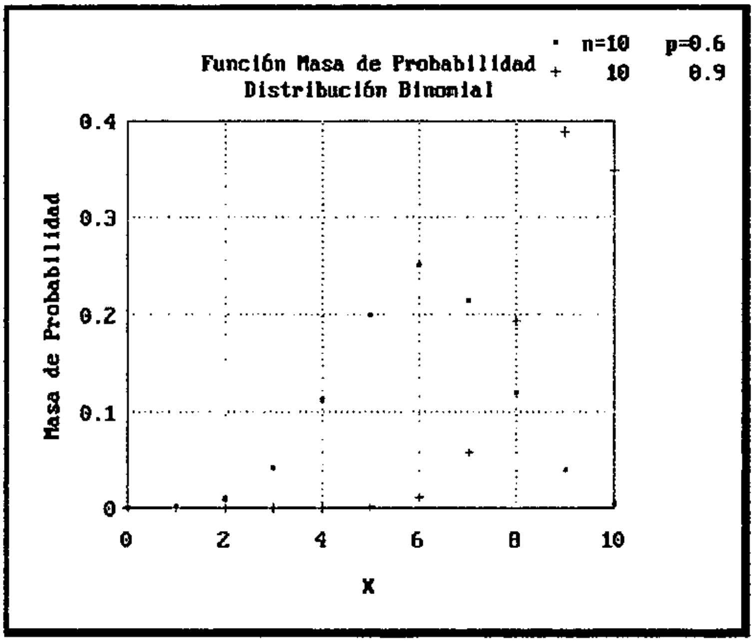
Su rango es: $S = \{0,1,2,\ldots,n\}$.

La media y la varianza están representadas por:

$$\mu = np$$

$$\sigma^2 = np(1-p)$$

La gráfica de esta distribución es como sigue



Esta distribución también se simula directamente:

1. Se simulan n eventos Bernoulli.
2. Se cuentan los eventos con resultado 1.

3.1.4 Distribución Geométrica.

Esta distribución también utiliza la distribución Bernoulli como base, ya que calcula la probabilidad de que al k-ésimo experimento se obtenga el primer resultado 1, por esta razón no tiene un número determinado de eventos.
Esta función se utiliza comúnmente en el área de control estadístico de la calidad y para distribuciones de intervalos econométricos.

Su función de probabilidad es definida por:

$$p(x) = (1-p)^{x-1} p$$

El rango de la distribución no está definido, por lo que el espacio muestral será: $S = \{0,1,2,\ldots\}$.

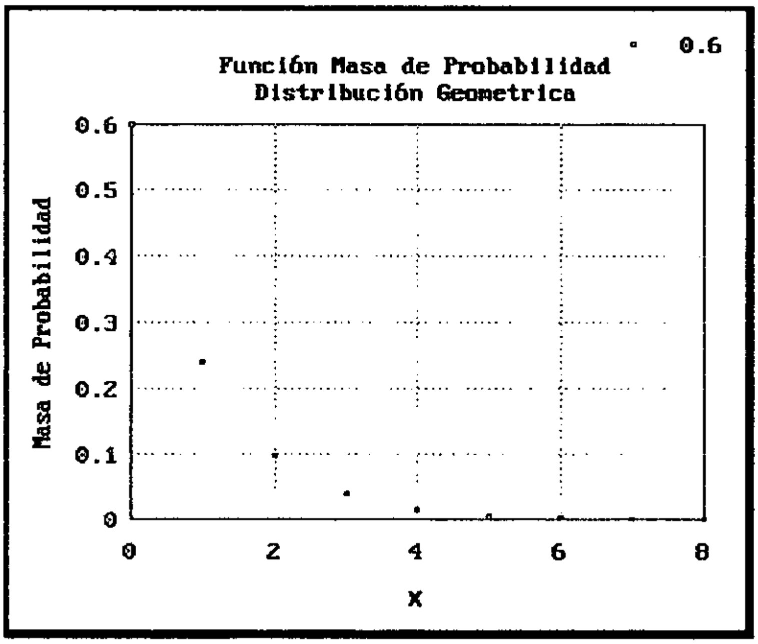
El único parámetro es p, que representa la probabilidad de que ocurra 1.

Su media y varianza están representadas por:

$$\mu = \frac{1-p}{p}$$

$$\sigma^2 = \frac{1-p}{p^2}$$

Su gráfica tiene la siguiente forma:



La simulación de esta distribución es por medio de simulación directa y se tiene que seguir el siguiente algoritmo:

1. Hacer $k = 0$
2. Se genera un evento Bernoulli.
3. Si el evento es 1 pasar a 5.
4. Si el evento es 0 hacer $k = k + 1$, regresar a 2.
5. Hacer $k = k + 1$.
6. Terminar.

Donde k es el número de eventos realizados

3.1.5 Distribución Binomial Negativa.

También conocida como Distribución de Pascal. Como las distribuciones anteriores la distribución binomial negativa tiene su base en la distribución Bernoulli, pero en este caso la variable aleatoria representa el ensayo en el cual ocurre el r-ésimo resultado 1.

La función de probabilidad es:

$$p(x) = \binom{x-1}{r-1} p^r (1-p)^{x-r}$$

El valor de x es a partir de r.

Los parámetros utilizados por esta distribución son p, que representa la probabilidad de que ocurra 1, y el número de veces que se repite 1 que se representa por r. Algo muy importante es que r, obviamente, es un número entero positivo.

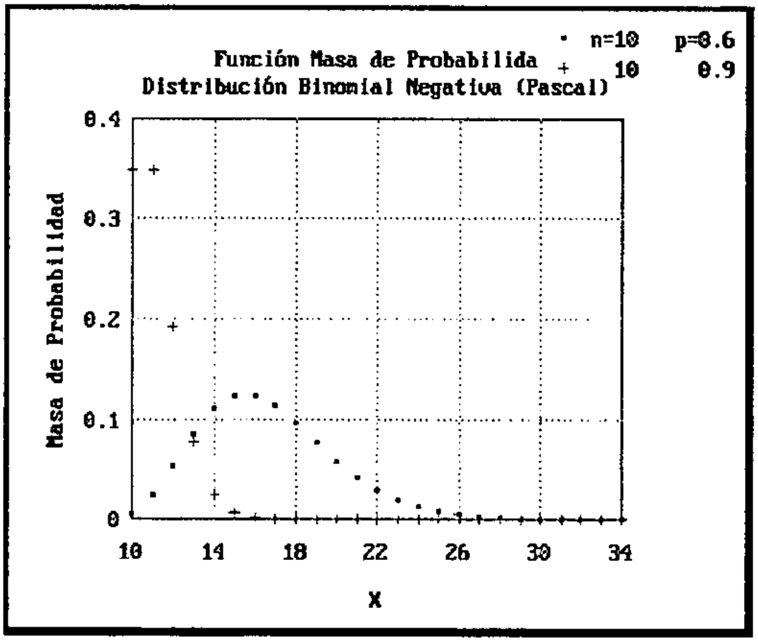
El rango de la distribución esta contenido en $S = \{r, r+1, r+2, ...\}$.

La media y la varianza de esta distribución están dadas por:

$$\mu = \frac{r(1-p)}{p}$$

$$\sigma^2 = \frac{r(1-p)}{p^2}$$

La gráfica de esta función es:



La simulación de esta distribución también se realiza por simulación directa, el algoritmo es el siguiente:

1. Hacer $k = 0$
2. Se genera un evento Bernoulli.
3. Si el evento es 1 pasar a 5.
4. Si el evento es 0 hacer $k = k + 1$, regresar a 2.
5. Hacer $k = k + 1$, $z = z + 1$.
6. Si $z < r$ regresar a 2.
7. Terminar

Donde k es el número de experimentos realizados y z es el número de experimentos con resultado 1.

3.1.6 Distribución Poisson.

Esta distribución se refiere a las probabilidades de ocurrencia de cierta cantidad de eventos dentro de un lapso de tiempo o espacio.

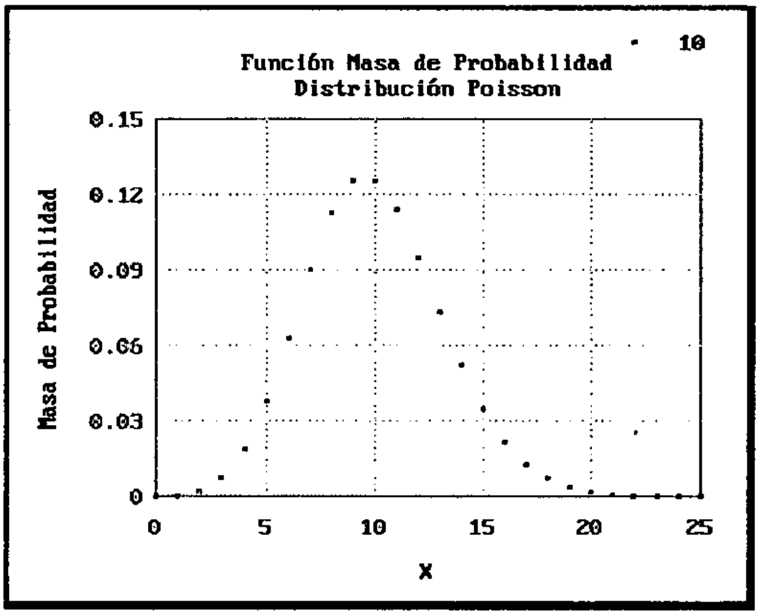
La función de la probabilidad es definida por:

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

El único parámetro necesario para manejar esta distribución es λ, que representa a la media y la varianza[4].

El rango de esta distribución es denotado por: $S = \{0,1,...\}$.

La gráfica de esta distribución se muestra a continuación:



El método de simulación de esta distribución es también por simulación directa, pero se explicará más adelante.

[4] Cabe mencionar que λ tiene que ser mayor que cero, ya que es el promedio del número de ocurrencias del evento en una unidad de tiempo o espacio.

3.2 Simulación continua.

Una función continua es aquella que contiene en su dominio un número infinito de puntos acotados dentro de un intervalo. La representación de estos valores es:

$$S = (a,b)$$

Donde S es el espacio muestral de la distribución.

Las distribuciones de probabilidad continuas más utilizadas son:

- ◊ D. Uniforme
- ◊ D. Normal
- ◊ D. Lognormal
- ◊ D. Exponencial
- ◊ D. Gamma
- ◊ D. Erlang
- ◊ D. Triangular
- ◊ D. Beta
- ◊ D. Weibull

3.2.1 Distribución Uniforme.

También conocida como distribución rectangular, su uso es muy importante dentro de la simulación, ya que con ella se generan los números aleatorios necesarios para obtener variables aleatorias con otras distribuciones de probabilidad.

Su función de densidad es:

$$f(x) = \frac{1}{b-a}$$

Los parámetros de esta distribución son a y b que son reales y $a<b$.

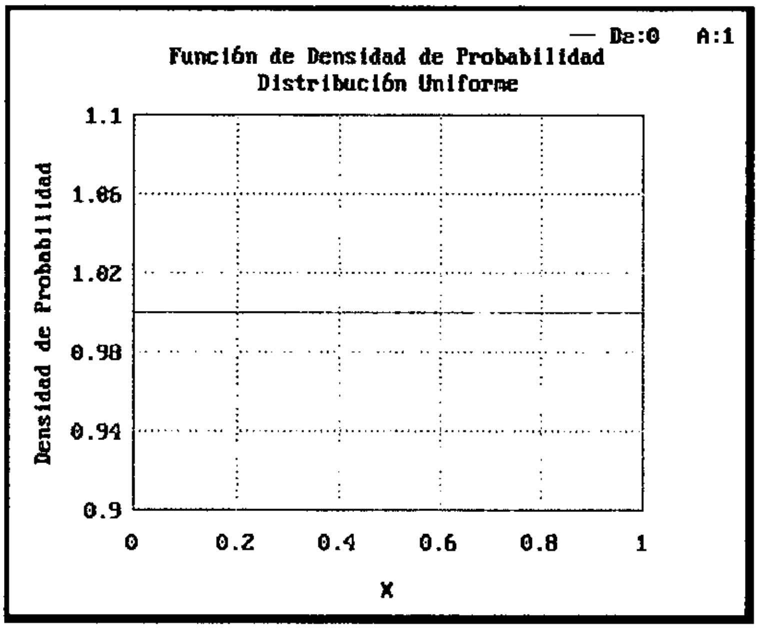
El rango de esta distribución esta dado por el intervalo cerrado $[a,b]$.

La media y la varianza están dadas por:

$$\mu = \frac{a+b}{2}$$

$$\sigma^2 = \frac{(b-a)^2}{12}$$

Su gráfica es la siguiente:



El método de simulación de esta distribución es el método de la transformada inversa y el algoritmo es el siguiente:

1. Hacemos: $F(x) = \int_a^x \frac{1}{b-a} dx$.

2. Resolviendo la integral obtenemos: $r = \frac{x-a}{b-a}$.

3. Despejando r se llega al generador buscado: $x = (b-a)r + a$.

Siendo este un generador sencillo para variables aleatorias uniformes en el intervalo $[a,b]$.

3.2.2 Distribución Normal.

Es una de las distribuciones más usadas en estadística, debido a que representa muchos fenómenos reales y por el teorema de límite central[5].

---

[5] Teorema del Limite Central: Sean $X_1, X_2, \ldots$ variables aleatorias independientes que están distribuidas idénticamente y tienen media μ y varianza σ^2 finitas. Entonces si $S_n = X_1, X_2, \ldots, X_n$,

$$limP\left(a \le \frac{S_n - n\mu}{\sigma\sqrt{n}} \right) \le b = \frac{1}{\sqrt{2}} \int_a^b e^{-\frac{u^2}{2}} du$$

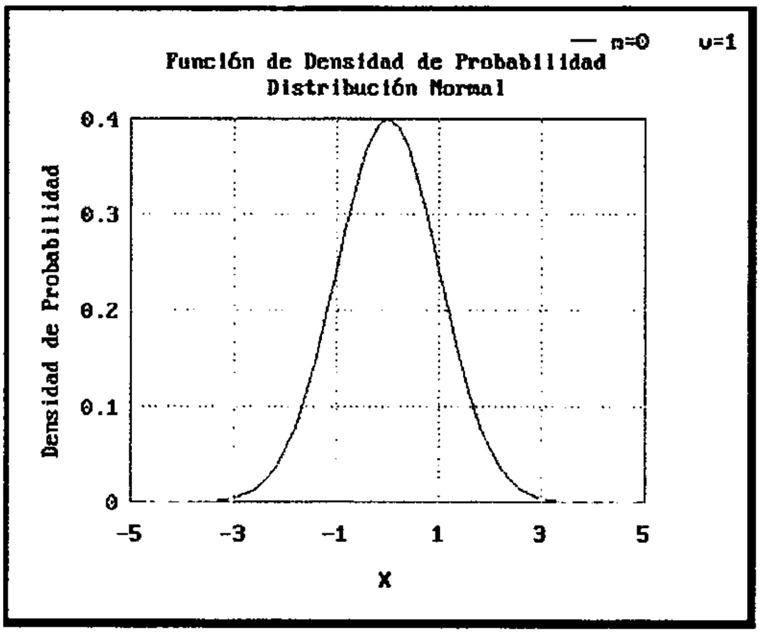
La función de densidad de la distribución normal es:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

Los parámetros necesarios para poder manejar esta distribución de probabilidad son la media μ y la varianza σ^2, siendo esta última mayor que cero.

El rango está denotado por todos los números reales, es decir $x \in \Re$, ya que esta distribución es asintótica en el eje de las abscisas.

Los datos de esta distribución, tienden agruparse alrededor de la media, es por eso la forma de campana simétrica que tiene.



La forma de simular la distribución normal es con el método de rechazo o bien por simulación directa, siguiendo uno de los dos algoritmos:

Algoritmo 1.

1. Se generan 12 números aleatorios con distribución uniforme entre 0 y 1.
2. Realizar la suma y restar 6. $Z = \sum_{i=1}^{12} r_i - 6$.

---

es decir la variable aleatoria $\left(\frac{S_n - n\mu}{\sigma\sqrt{n}}\right)$ que es la variable tipificada a S_n, es normal asintóticamente.

3. Al resultado obtenido multiplicar la desviación estándar y sumar la media: $x = \mu + Z\sigma$.

Este algoritmo tiene algunas desventajas como son:

- ◊ Utiliza muchos números uniformes.
- ◊ Requiere mucho tiempo operacional.
- ◊ La distribución de una suma de 12 elementos no es exactamente una normal, por lo que los números generados podrían desviarse.

Algoritmo 2.

1. Se generan dos números aleatorios con distribución uniforme.

2. Se aplica una de las siguientes fórmulas:

$$Z = \left(-2r_1\right)^{\frac{1}{2}} \operatorname{sen}\left(2\pi r_2\right)$$

$$Z = \left(-2r_1\right)^{\frac{1}{2}} \cos\left(2\pi r_2\right)$$

3. Multiplicar por la desviación estándar y sumar la media: $x = \mu + Z\sigma$.

El segundo algoritmo fue desarrollado por Box y Muller, es un método fácil de utilizar y rápido en un proceso computacional, por lo que es el método utilizado por SIMULAT.

3.2.3 Distribución Lognormal.

Esta distribución es la forma más sencilla de una función de densidad cuyo logaritmo sigue una distribución normal, esta distribución surge de la combinación de varias variables aleatorias en un proceso multiplicativo.

La función de densidad de esta distribución es:

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}$$

Para toda x>0.

Los parámetros de esta distribución son los mismos que para la normal: media μ y varianza σ^2.

El rango está comprendido por el intervalo $[0,\infty)$.

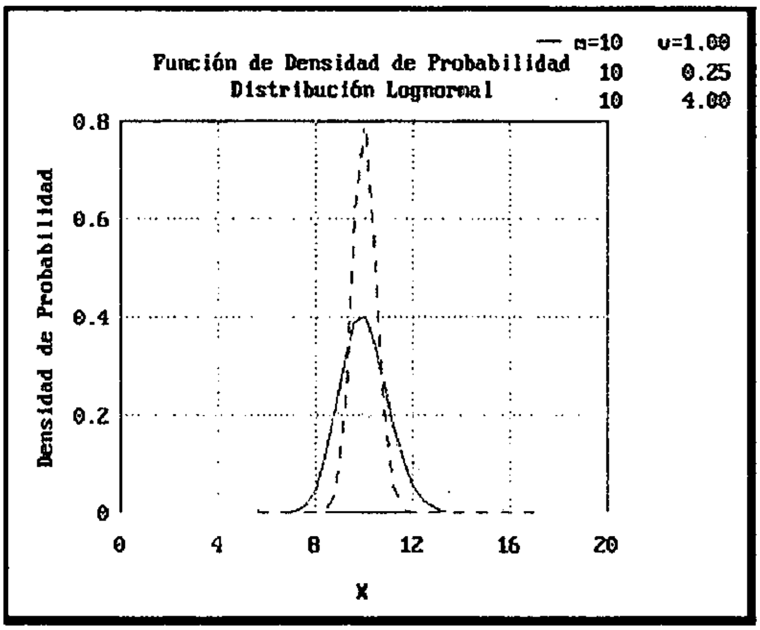
La media se obtiene mediante la fórmula:

$$e^{\frac{\mu+\sigma^2}{2}}$$

La varianza esta dada por la ecuación:

$$e^{2\mu+\sigma^2}\left(e^{\sigma^2}-1\right)$$

La gráfica es la siguiente:



El forma de simular esta distribución es por simulación directa:

1. Generar un número Z con distribución normal.
2. Hacer $x = e^Z$.

3.2.4 Distribución Exponencial.

También es llamada Exponencial Negativa. Esta distribución representa la división del tiempo entre las ocurrencias de los eventos de conteo motivo por el cual está muy relacionada con la distribución de Poisson. De hecho, es la distribución del tiempo que transcurre entre eventos Poisson.

La función de densidad está definida por:

$$f(x) = \lambda e^{-\lambda x}$$

El único parámetro de esta distribución es λ.

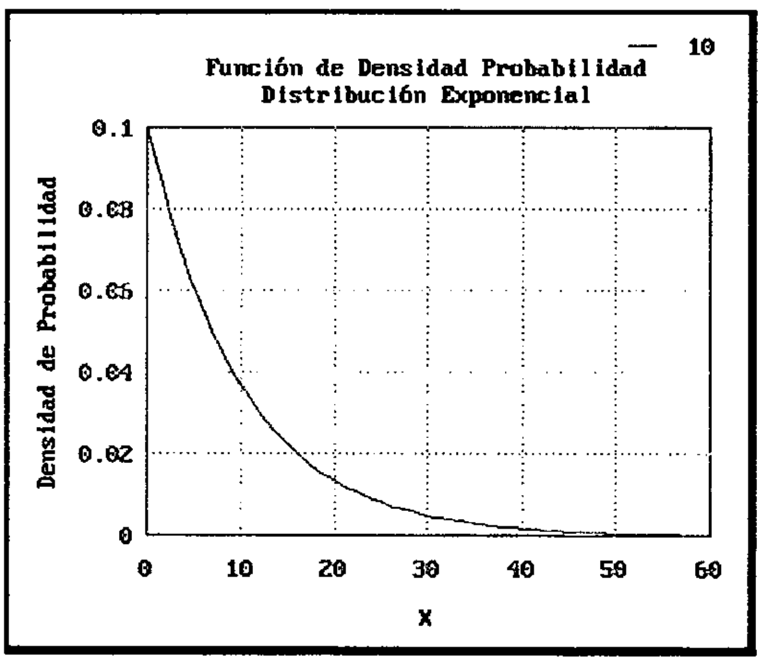
El rango está comprendido en el intervalo $[0,\infty)$.

La media y varianza están definidas por:

$$\mu = \frac{1}{\lambda}$$

$$\sigma^2 = \frac{1}{\lambda^2}$$

La gráfica es la siguiente:



La distribución exponencial se simula por el método de la transformada inversa, el algoritmo es el siguiente:

1. Se resuelve la integral

$$\int_0^x \lambda e^{-\lambda x} dx$$

2. Igualando a r y despejándola obtenemos el generador

$$x = \frac{-\ln(1-r)}{\lambda}$$

Esta distribución es importante ya que de ella se pueden obtener por convolución la distribución Gamma.

La distribución de Poisson se simula mediante la distribución exponencial y para esto se sigue el siguiente algoritmo:

1. Se hace $T = 0$ y $x = 0$.
2. Se simula un r que siga una distribución exponencial.
3. Se hace $T = T + r$.
4. Si $T > 1$ entonces terminar simulación.
5. Hacer $x = x + 1$.
6. Repetir 2.

3.2.5 Distribución Gamma.

También es conocida como distribución Erlang. Esta distribución se obtiene como resultado de una convolución[6] de k variables exponenciales con idéntica λ y además k sigue una distribución binomial negativa o geométrica.

Antes de hablar de la distribución Gamma, se explicará lo que es la función gamma la cual esta esta definida por:

$$\Gamma(n) = \int_0^\infty x^{n-1} e^{-x} dx$$

Al realizar la integración por partes obtenemos que:

$$\Gamma(n) = (n-1)\Gamma(n-1)$$

Lo cual define a la función como:

$$\Gamma(n) = (n-1)!$$

[6] Suma.

Por otro lado la distribución Gamma define su función de densidad de probabilidad como:

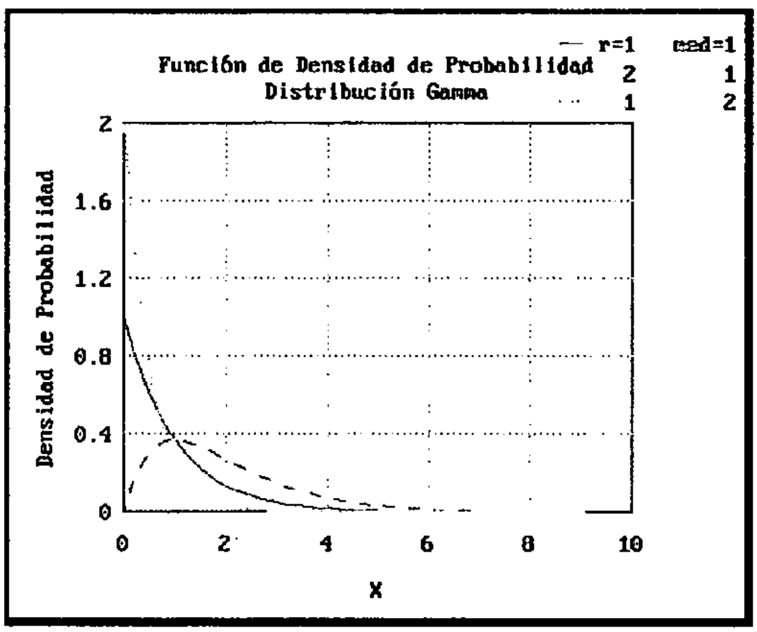
$$f(x) = \frac{\lambda}{\Gamma(r)}(\lambda x)^{r-1} e^{-\lambda x}$$

Esta distribución cuenta con dos parámetros, el parámetro de forma r y el parámetro de escala λ, ambos tienen que ser mayores que cero.

El rango de esta distribución se encuentra en el intervalo $[0,\infty)$.

Su media esta dada por $\frac{r}{\lambda}$, mientras que su varianza es $\frac{r}{\lambda^2}$.

La gráfica de esta distribución tiene la siguiente forma:



El método para simular la distribución Gamma es por convolución:

1. Se generan k exponenciales con media λ.
2. Se suman los valores simulados.

3.2.6 Distribución Triangular.

Esta distribución es asimétrica según convenga al analista y es fácil de utilizar.

Su función de densidad de probabilidad es la siguiente:

$$f(x)=\begin{cases}\dfrac{2(x-O)}{(L-O)(P-O)}, O\le x\le L\\[2ex]\dfrac{2(P-x)}{(P-L)(P-O)}, L<x\le P\end{cases}$$

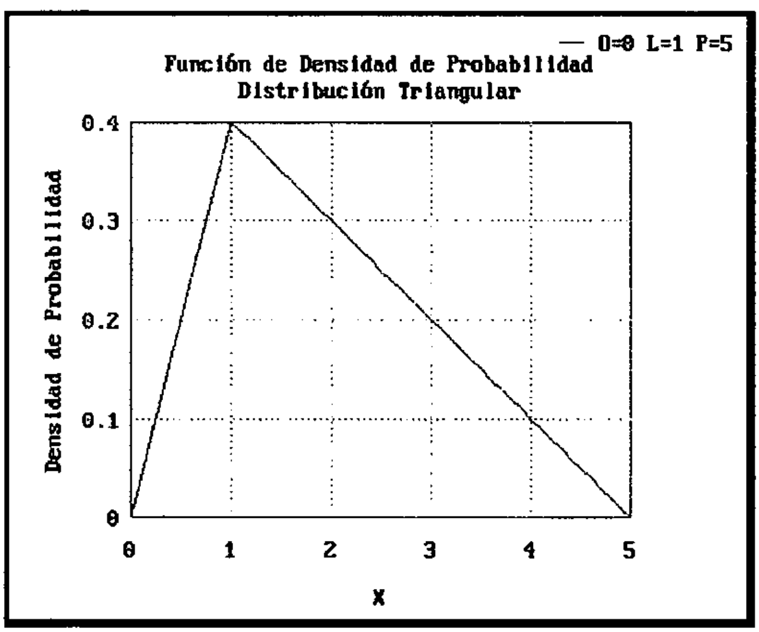
Esta distribución maneja tres parámetros: O es el menor, L es el más probable (moda) y P que es el más lejano al origen.

El rango está dado por: $[O,P]$.

La media está dada por la ecuación: $\dfrac{O+L+P}{3}$.

La varianza se define por: $\dfrac{O^2+L^2+P^2-OL-OP-PL}{18}$.

La gráfica de esta distribución como se puede ver es un triángulo.



La forma de simular esta distribución es por transformada inversa:

1. Se resuelven las integrales de $f(x)$, se despeja x y se se obtienen los generadores:

$$x=O+\sqrt{r(L-O)(P-O)}$$

$$x = P - \sqrt{(1-r)(P-L)(P-O)}$$

2. Se genera un r aleatorio uniforme entre 0 y 1.
3. Se evalúa la regla de decisión:
   - ◊ Se usa el primer generador si: $r < \dfrac{L-O}{P-O}$.
   - ◊ Se usa el segundo generador si $r \geq \dfrac{L-O}{P-O}$.

CAPÍTULO 4
GENERALIDADES DE SIMULAT.

Para poder realizar la simulación de muchos eventos es necesario un gran número de operaciones, por lo que es necesario buscar la manera más rápida de obtener los resultados, por lo que la computadora es, generalmente, indispensable.

Existen diversas herramientas que se pueden ocupar para la simulación contándose las siguientes:

- ◊ Paquetes estadísticos
- ◊ Lenguajes de uso general
- ◊ Lenguajes de uso específico

Dentro de los lenguajes de uso específico, se contempla SIMULAT, que es empleado para la simulación de variables aleatorias, líneas de espera, procesos markovianos y series de tiempo.

A lo largo de los siguientes capítulos se darán las especificaciones, instrucciones y aplicaciones de SIMULAT.

4.1 Requerimientos de SIMULAT.

Los vertiginosos avances en cuanto a computación se refiere, implican, que para poder ejecutar un software adecuadamente se requiere de un equipo con determinadas características, las cuales darán un mejor funcionamiento, mayor rapidez e incluso, permitan una mejor presentación, o simplemente la ejecución del software.

Por lo expuesto anteriormente se darán los requerimientos mínimos necesarios, en cuanto a hardware y software se refiere, para poder ejecutar SIMULAT: Un lenguaje de simulación para líneas de espera y redes.

El lenguaje de simulación SIMULAT fue diseñado en DELPHI 1.0, que es un lenguaje orientado a objetos gráficos, por lo que se maneja bajo ambiente Windows® de Microsoft® utilizando la versión 3.11 o superior. Ya que conocemos los requerimientos de software, se presentan en la siguiente tabla los requerimientos de hardware:

Hardware	Requerido mínimo
Procesador	386
Co-procesador matemático	No requerido
Memoria RAM	1 MB
Espacio en disco duro	1 MB
Impresora	Esté dada de alta en Windows®

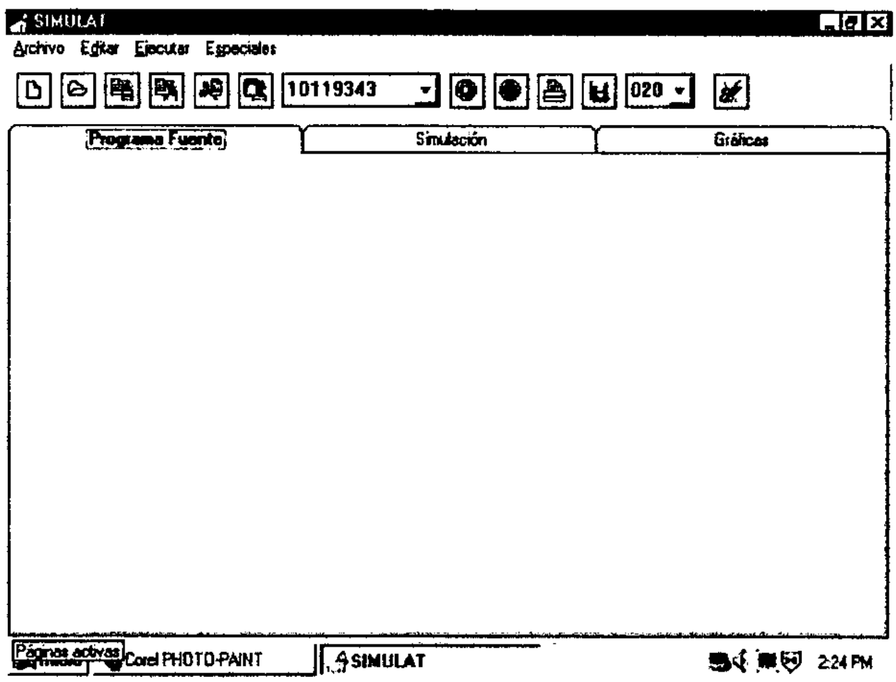
Como se puede ver, con lo que ofrece el mercado en la actualidad, es suficiente para satisfacer los requerimientos de SIMULAT.
Por las características de la computadora donde se elaboró el presente trabajo se presentaran las pantallas que muestra Windows® 95, siendo las pantallas muy similares en Windows® 3.11.

4.2 Iniciando con SIMULAT.

Para iniciar una sesión con SIMULAT basta con seguir los siguientes pasos:

1. Inserte el disco donde se encuentra almacenado SIMULAT.
2. Desde el Explorador de Windows® ejecute SIMULAT.EXE. Puede crear un acceso directo para la próxima vez que utilice el Lenguaje.

Al iniciar con SIMULAT se mostrará la siguiente pantalla [1]:



Existen 4 opciones de menú las cuales se presentan como sigue:

Menú	Opciones	Función
ARCHIVO	• Abrir	Abre un archivo de SIMULAT. Guarda el programa y los resultados[2].

[1] Es la pantalla de inicio cuando se utiliza Windows® 95.

	• Guardar • Guardar como • Nuevo • Salir	Modifica el nombre de un archivo permitiendo hacer modificaciones. Limpia la pantalla para un nuevo programa. Termina la ejecución de SIMULAT[3].
EDITAR	• Copiar • Cortar • Pegar	Copia la selección al portapapeles. Corta la selección y la almacena en el portapapeles. Pega el contenido del portapapeles.
EJECUTAR	• Analizar • Ejecutar • Corridas	Revisa la sintaxis del programa fuente. Realiza la ejecución del programa y muestra los resultados. Modifica el número de veces que se ejecuta el programa.
ESPECIALES	• Imprimir • Histográma	Envía los resultados de la simulación a la impresora. Muestra un histograma de los resultados.

Se manejan también botones de acceso directo los cuales son:

Botón	**Función**
	Nuevo programa.
	Abre un programa.
	Guarda un archivo.
	Copia la selección al portapapeles.
	Corta la selección al portapapeles.
	Pega el contenido del portapapeles.
	Analiza el código del programa[4].
	Ejecuta el programa[5].
	Envía a la impresora los resultados.
	Muestra una gráficas de los resultados.

[2]El código fuente del programa se guarda en un archivo *.SIM, y los resultados se almacenan en un archivo *.DAT.

[3] Esta opción no almacena las modificaciones hechas al programa fuente o a los resultados de la simulación, por lo que es necesario usar las opciones **Guardar** o **Guardar como** antes de salir.

[4] Botón amarillo.

[5] Botón verde.

	Borra los resultados para una nueva simulación.

Existen dos cajas de texto con posibles valores a elegir estas son:

Nombre	Caja	Función
Número de corridas	020	Es el lugar donde el usuario puede seleccionar el número de veces que se ejecutará el programa. El valor por defecto es 20, pero se puede cambiar dentro del rango de 1 a 999.
Semilla	10119343	Es el lugar donde el usuario puede seleccionar la semilla del generador de números pseudoaleatorios. El valor por defecto es 10119343, y el rango es desde 1 hasta 2^{32}-1.

En las pestañas superiores se indican las páginas disponibles y cuál es la que está en uso:

Pestaña	Función
Programa Fuente	Muestra la pantalla donde se encuentra el código fuente del programa.
Simulación	Muestra la pantalla donde se encuentran los resultados de la simulación.
Gráficas	Despliega el histograma de la simulación (sólo en variables aleatorias y procesos de Markov).

4.3 Las instrucciones de SIMULAT.

El total de instrucciones que componen a SIMULAT son 16 y se pueden combinar de diferentes formas, pero sus usos están limitados para el fin con que fueron creadas. A continuación se presenta la sintaxis de las instrucciones de SIMULAT, así como la combinación correcta de las instrucciones.

4.3.1 Variable aleatoria.

SIMULAT permite realizar la simulación de una variable aleatoria, mediante el uso de la instrucción:

$$VARALE\ (n, f(x))$$

donde:

Parámetro	Función
n	Es la cantidad de números que se generan, debe ser menor a 2500.
f(x)	La distribución de probabilidad.

Las distribuciones de probabilidad que puede tomar *f(x)* son:

Distribución	Parámetros	*f(x)*
Bernoulli	p: probabilidad de que ocurra el evento 1.	BR (p)
Binomial	N: es el número de ensayos Bernoulli. p: probabilidad de que ocurra el evento 1.	BN (N.p)
Geométrica	p: probabilidad de que ocurra el evento 1.	GE (p)
Binomial Negativa	p: probabilidad de que ocurra el evento 1. R: el r-ésimo resultado 1.	PL (p,R)
Poisson	λ: es el promedio de ocurrencias de un evento dentro de un intervalo de tiempo.	PO (λ)
Uniforme Discreta	N: es el número de eventos equiprobables.	UD (N)
Exponencial	λ: es el promedio de ocurrencias de un evento dentro de un intervalo de tiempo[6].	EX (λ)
Gamma	λ: es el promedio de ocurrencias de un evento dentro de un intervalo de tiempo. N: número de exponenciales a sumar.	GM (λ,N)
Normal	μ: la media de la distribución. σ^2: la desviación estándar de la distribución.	NM (μ,σ^2)
Lognormal	μ: la media de la distribución. σ^2: la desviación estándar de la distribución.	LN (μ,σ^2)
Triangular	O: el límite inferior de la	TR (o,m,p)

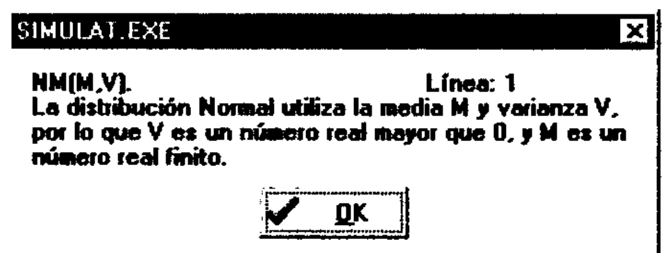
[6] Hay que recordar que la distribución exponencial representa el tiempo (espacio) entre dos eventos de Poisson.

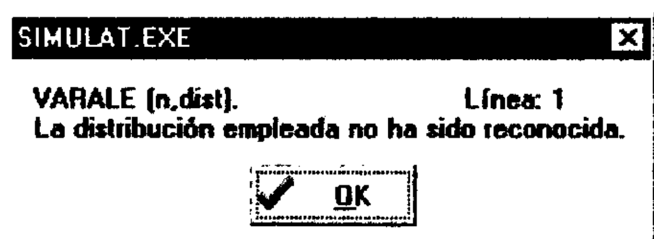
	distribución. M: la moda de la distribución. P: el límite superior de la distribución.	
Uniforme continua	a: límite inferior de la distribución. b: límite superior de la distribución.	UC (a,b)

Por ejemplo, si se quieren generar 10 números aleatorios que se distribuyan como una normal con media 0 y varianza 1 se escribe:

VARALE (10,NM(0,1))

Al realizar la revisión de sintaxis, si SIMULAT encuentra algún error lo mostrará en pantalla mediante alguna ventana como las siguientes:





Como se puede ver la ventana muestra la línea de programa donde se presenta el error.

Los resultados de la simulación se presentan mostrando primero los valores generados e inmediatamente su media y su varianza.

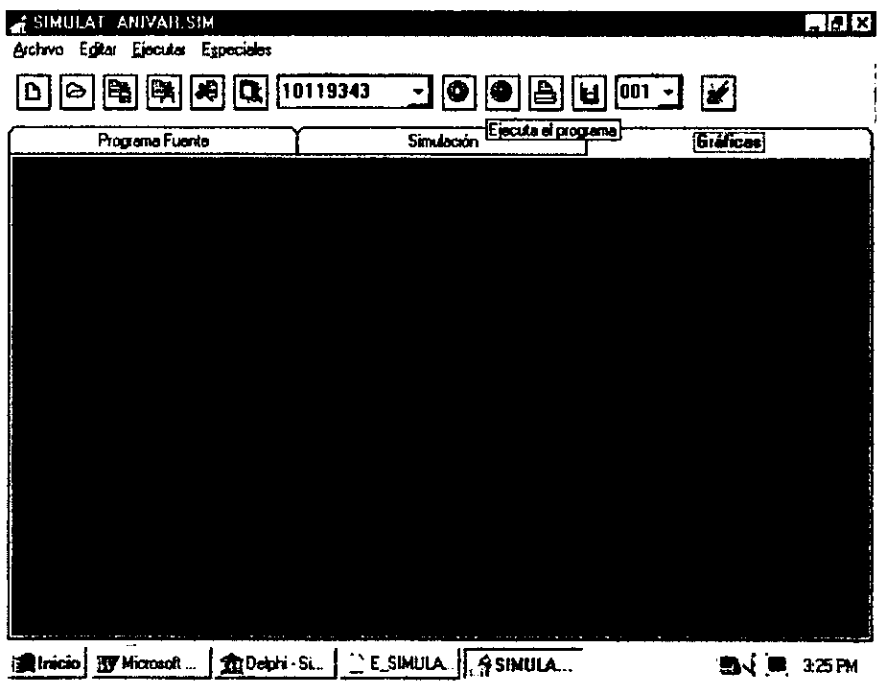
```
     * * * * *        Inicia corrida:   1         * * * * *

0.57621
-0.45655
0.80481
0.47062
```

```
0.48995
-0.70911
0.15499
0.99069
0.64086
-0.48426
 La media de esta distribución es: 0.247819738843418
 La varianza de la distribución es: 0.318909406573766

     * * * * *       Termina la corrida:  1       * * * * *
```

Cuando se elige la opción Gráfica muestra el histograma de los datos obtenidos en la simulación.



Los datos que se muestran pueden ser utilizados por paquetes estadísticos[7] eliminando las líneas que presentan el número de corridas.

[7] Existen diversos paquetes estadísticos como el STATGRAPHICS, SPSS, SYSTAT, entre otros, pero el que se probó fue el STATGRAPHICS.

4.3.2 Modelos autorregresivos y de medias moviles.

Con este lenguaje de simulación también se pueden crear series de tiempo, utilizando modelos autorregresivos y de medias móviles, que se obtienen de aplicar la metodología de Box-Jenkis, usando la instrucción:

ARMA (AR, MA, n, Parám. AR, Parám. MA)

donde:

Parámetro	Función
AR	Es número de parámetros autorregresivos que serán considerados.
MA	Es número de parámetros de medias móviles que serán considerados.
n	Es el número de datos a simular.
Parám. AR	Son los valores que tomará cada ϕ del modelo[8].
Parám. MA	Son los valores que tomará cada θ.

SIMULAT realiza la simulación de la serie de tiempo iniciando desde cero (0), para que esto no ocurra se maneja una instrucción que permite dar un valor inicial a la serie, esta instrucción es:

INICIAL (Z_0)

donde:

Parámetro	Función
Z_0	Es el valor con el cual se iniciará la serie de tiempo.

Por ejemplo, para realizar la simulación de una serie de tiempo que se comporta como un modelo ARMA(1,1) con parámetros $\phi= 0.1$ y $\theta= 0.5$ se utilizará el siguiente código:

ARMA (1,1,10,0.1,0.5)

Si se trata de simular una serie con las características anteriores, pero, inicia en 100 es necesario establecer el siguiente código:

INICIAL (100)
ARMA (1,1,10,0.1,0.5)

[8] Cuando son más de un dato, se separarán por comas (,).

Como sabemos existe un error aleatorio, pero no siempre con varianza 1, esto lo podemos modificar con la instrucción:

VARIANZA (varianza)

Al realizar la revisión de la sintaxis también se mostrarán ventanas que indicarán el posible error del programa y la línea donde se localiza.

Los resultados sólo se presentan en la ventana SIMULACIÓN en forma de listado, tal y como se muestra abajo:

```
     * * * * *        Inicia corrida:   1        * * * * *

100.57621
159.88918
166.56545
167.52957
167.47822
166.28369
166.42880
167.71106
167.90731
166.62690

     * * * * *        Termina la corrida:  1        * * * * *
```

4.3.3 Procesos markovianos.

SIMULAT da la facilidad de realizar simulaciones de procesos markovianos de forma rápida, sencilla y eficiente, mediante el uso de la instrucción:

PM (Estados, Nombres)

donde:

Parámetro	Función
Estados	Es el número de estados[9] del proceso markoviano.
Nombres	Asigna el nombre a cada estado.

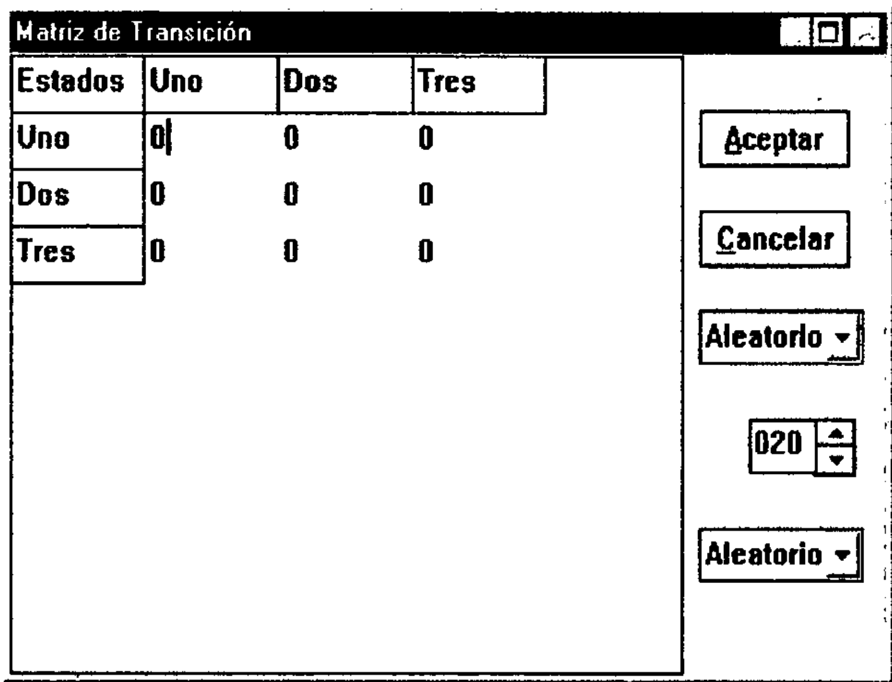
Al ejecutar el código se abrirá una ventana para llenar la matriz de transición, se puede modificar el número de unidades de tiempo a simular, y dar los estados inicial y final.

[9] El número máximo de estados es 10, y el mínimo es 2.

Por ejemplo: si se quiere un proceso markoviano con los estados Uno, Dos y Tres se realiza como:

PM (3, Uno, Dos, Tres)

Al ejecutar el código aparece la siguiente ventana:



Aquí aparece la matriz de transición, el estado donde se inicia, el número de unidades de tiempo para simular y el estado donde se detendrá el proceso[10].
Al elegir el botón "Aceptar" se revisará que en cada renglón, de la matriz de transición, la suma de probabilidades sea igual a 1, en caso de que esto no se cumpla se mostrará una ventana con el renglón y el porque no se cumple en este requisito.

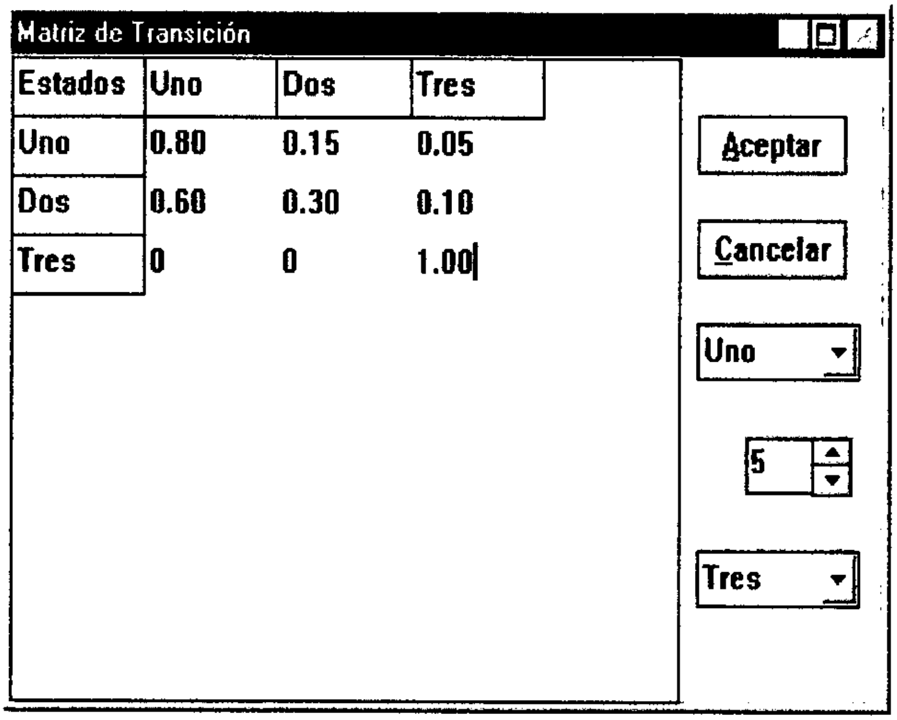
Al elegir el estado final del proceso hay que tener cuidado de manejar sólo estados absorbentes ya que la simulación se detendrá en ese momento y dará inicio un nuevo proceso.

Continuando con el ejemplo, si manejamos la siguiente matriz de transición:

Estado	Uno	Dos	Tres
Uno	0.80	0.15	0.05
Dos	0.60	0.30	0.10
Tres	0.00	0.00	1.00

[10] Aparecerán etiquetas con lo que realiza cada opción.

Tomando el estado Uno como inicial, el estado Tres como el estado final y se realiza la simulación de 10 unidades de tiempo, tenemos:



Los resultados se muestran indicando los estados y su nombre, el estado inicial, se indican los movimientos a través del tiempo, el número de veces que se visita cada estado y el total de visitas en toda la simulación[11]:

```
* * * * *        Inicia corrida:    1        * * * * *


Estado   1: Uno
Estado   2: Dos
Estado   3: Tres

Estado inicial:   Uno

Del estado Uno pasa al estado Uno
Del estado Uno pasa al estado Uno
Del estado Uno pasa al estado Dos
Del estado Dos pasa al estado Uno
Del estado Uno pasa al estado Uno
```

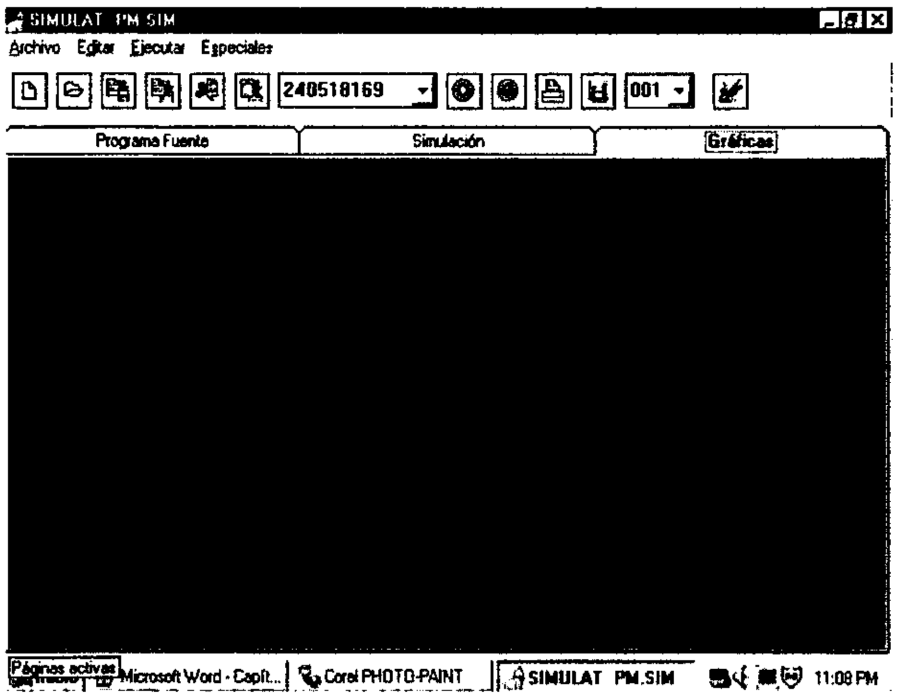
[11] Se muestra esta información cuando se realizan 10 o menos corridas, si se ejecutan más de 10 corridas no se muestra el movimiento a través del tiempo.

```
El estado Uno fue visitado 4 veces. Total 4
El estado Dos fue visitado 1 veces. Total 1
El estado Tres fue visitado 0 veces. Total 0

* * * * *        Termina la corrida:  1        * * * * *
```

Como se puede ver, se presenta cada uno de los cambios en el proceso y al final se indica cuantas veces fue visitado cada estado.

Si se elige la opción Gráfica se crea un histograma que muestra el comportamiento global de la simulación :



4.3.4 Líneas de espera.

SIMULAT permite también la simulación de líneas de espera mediante el uso de las instrucciones:

Instrucción	Función
COLA	Genera las llegadas a una línea de espera.
SERVIDOR	Genera el tiempo de servicio de cada transacción dentro del servidor.
ESPERAC	Detiene las llegadas a la cola de espera por un tiempo determinado.

ESPERAS	Detiene el servicio de los servidores por un tiempo determinado.

La sintaxis de las instrucciones es la siguiente:

COLA (NombreC, f(x)c, Máximo, Inicial, Tipo, Tiempo)

donde:

Parámetro	Función
NombreC	Es el nombre de la cola.
f(x)c	Es la distribución de tiempo entre llegadas a la cola[12].
Máximo	Es el tamaño máximo de la cola
Inicial	El número de elementos que están esperando servicio en la cola al inicio de la simulación.
Tipo	Es la forma en que se maneja la cola FIFO (First In, First Out) o LIFO (Last In, First Out).
Tiempo	Es el tiempo que se simularán llegadas a la cola.

SERVIDOR (NombreS, Servidores, ServOcu, f(x)s)

donde

Parámetro	Función
NombreS	Es el nombre del servidor.
Servidores	Es el número total de servidores disponibles.
ServOcu	Es el número de servidores ocupados al iniciar la simulación.
f(x)s	Es el tiempo de servicio (se recomienda utilizar la distribución exponencial).

ESPERAC (TiempoI, TiempoF)

donde:

Parámetro	Función
TiempoI	Es el minuto donde inicia la interrupción las llegadas a la cola.
TiempoF	Es el minuto donde termina la interrupción de llegadas a la cola.

ESPERAS (TiempoI, TiempoF)

[12] Se manejan las mismas distribuciones que para VARALE, generalmente, una línea de espera se comporta como un proceso Poisson, por lo que es común utilizar la distribución exponencial.

donde:

Parámetro	Función
TiempoI	Es el minuto donde inicia la interrupción de servicio en los servidores.
TiempoF	Es el minuto donde termina la interrupción de servicio en los servidores.

Por ejemplo si se quiere simular 240 unidades de tiempo de una línea de espera, en la cual las transacciones llegan en forma exponencial con media 5, se tienen 3 servidores los cuales realizan un servicio el cual se comporta como una distribución exponencial con media 1.5.

COLA (Transacción, Ex(5), 0, 0, FIFO, 240)
SERVIDOR (Servicio, 3, 0,EX(1.5))

Con este mismo ejemplo, si suponemos que los servidores realizan una pausa desde la unidad de tiempo 100 hasta la 150 horas y las transacciones dejan de llegar en el intervalo de 170 a 200 .

COLA (Transacción, Ex(5), 0, 0, FIFO, 240)
ESPERAS (100,150)
ESPERAC (170,200)
SERVIDOR (Servicio, 3, 0,EX(1.5))

Los resultados se muestran únicamente en forma de texto en la siguiente forma:

```
   COLA:  TRANSACCIÓN                    TIPO: FIFO
     Transacciones                    Tiempo promedio
Totales:            1233
En la cola:         1233          De espera:       9.01669
Destruidas:            0
Que esperaron:      1102          De espera:      10.08855
En espera:           122
Tamaño máximo de la cola:     123
Tiempo máximo de espera:       22.46921

   SERVIDOR:  SERVICIO                TOTAL DE SERVIDORES:  3
          Tiempo ocupado                    Tiempo en reposo
Servidor  1:      239.38685         Servidor  1:       0.61315
Servidor  2:      240.59331         Servidor  2:      -0.59331
Servidor  3:      238.93059         Servidor  3:       1.06941
```

A continuación se explica cada uno de los datos mostrados en a pantalla:

- Cola: Nombre de la cola.
- Tipo: Tipo de cola, Fifo o Lifo.
- Totales: El número total de transacciones creadas.
- En la cola: Número de transacciones que entraron a la cola.
- De espera (Tiempo promedio): Da el promedio del tiempo de espera de todas las transacciones.
- Destruidas: Son las transacciones que no entraron a la cola (se usa cuando la cola tiene límite).
- Que esperaron: Son las transacciones que realmente esperaron.
- De espera: Da el promedio del tiempo de espera de las transacciones que esperaron.
- En espera: Son las transacciones que quedaron en la cola.
- Tamaño máximo de la cola: Da el tamaño máximo de la cola
- Tiempo máximo de espera: Da el tiempo máximo de espera
- Servidor: Da el nombre del servidor
- Total de servidores: Da el número de servidores.
- Tiempo ocupado: Da el tiempo que está ocupado el servidor.
- Tiempo en reposo: Da el tiempo que el servidor esta inactivo (Cuando es negativo, indica que el servidor brindó servicio más tiempo del que dura la simulación).

4.3.5 Operaciones básicas.

Se tienen seis operaciones básicas, estas son: suma, resta, multiplicación, división, potenciación e igualación.
La forma de utilizar cada una se describe en el siguiente cuadro:

Operación	Instrucción
A+B	Suma (A,B)
A-B	Resta (A,B)
A*B	Multi (A,B)
A/B	Div (A,B)
A^B	Pot (A,B)
A=B	X (B) Y(B)

El parámetro A puede tomar diversos valores: una variable aleatoria, la cual irá con un asterisco al final, un valor numérico, el resultado anterior[13] o un valor almacenado, mientras el parámetro B solamente puede tomar valores numéricos o el resultado anterior. El resultado anterior se representa por la palabra "RESULTADO" quedando de la siguiente forma:

[13] Este resultado se refiere a las operaciones básicas.

- suma (resultado,5)
- multi (*f(x)**,resultado)
- div (resultado,resultado)
- resta (10,resultado)
- pot (*X*,2)
- X (resultado)
- Y (14)

Para mostrar el resultado de cualquiera de estas operaciones, es necesario escribir la instrucción:

Instrucción	Muestra en pantalla
VER (x)	El valor almacenado en X.
VER (y)	El valor almacenado en Y.
VER (resultado)	El valor de la última operación realizada.

Por ejemplo, si se quiere obtener el resultado de la multiplicación de una distribución normal estándar por cinco se representaría el código de la siguiente forma:

```
MULTI (NM(0,1)*,5)
VER (resultado)
```

El resultado obtenido sería:

```
* * * * *        Inicia corrida:   1        * * * * *

2.88107059358663


* * * * *        Termina la corrida:  1        * * * * *
```

Es importante remarcar que las igualdades X(B) y Y(B) no aceptan variables aleatorias, pero para que X o Y sean una variable aleatoria se puede hacer:

```
suma (f(x)*,0)
x (resultado)
```

De igual manera es necesario que las otras operaciones tengan en su formato primero la igualdad y después el otro parámetro:

```
suma (X, resultado)
resta (Y, 12)
```

CAPÍTULO 5
ALCANCES Y LIMITES DE SIMULAT.

Existen diversos aspectos que influyen en el desarrollo de un software, pero los más importantes son:

- ◊ Flexibilidad de la herramienta de desarrollo (lenguaje).
- ◊ Conocimiento de los requerimientos del software.
- ◊ Características del equipo de cómputo (PC).
- ◊ Creatividad del desarrollador (programador).

El conocer estos puntos ayudarán a comprender los límites, errores y fallas de cualquier software, en este caso del lenguaje SIMULAT, el presente trabajo no tiene por objetivo dar una amplia explicación de estos puntos, pero dará una breve descripción de cada uno.

5.1 Flexibilidad de la herramienta de desarrollo (lenguaje).

Este punto consiste en las "facilidades" de programación que brinda el lenguaje elegido para el desarrollo de la herramienta (SIMULAT) :

- ◊ Uso de memoria.
- ◊ Manejo de un ambiente gráfico.
- ◊ Facilidad de comunicación con los diferentes periféricos.
- ◊ Manipulación de datos.

Ahora bien, los lenguajes pueden manipular entidades que tienen un uso especifico, propiedades bien definidas y sucesos que se pueden presentar. A las entidades anteriores reciben el nombre de objetos[1]. La programación orientada a objetos (POO) se basa en esto y la mayoría de los lenguajes actuales son en ambiente visual y tienen objetos de texto, menúes, gráficos, manipuladores de bases de datos y herramientas del sistema, entre los más importantes.

Delphi es un lenguaje orientado a objetos, el cual fue utilizado para crear SIMULAT, se emplearon los objetos propios del lenguaje, por lo que si alguno tiene restricciones, estas se heredaron a SIMULAT.

[1] Existen lenguajes que ya tienen objetos creados listos para usarse, aunque un objeto se puede crear en casi todos los lenguajes.

5.2 Conocimiento de los requerimientos del software.

El conocimiento de los requerimientos del software es lo que hará que durante el desarrollo sea el correcto, se prevean posibles errores y así no se obtengan resultados equivocados. En este punto se pueden presentar errores continuamente, ya que al realizar el análisis de los requerimientos no siempre se toman todas las variables para realizarlo.

Aquí los requerimientos son fórmulas matemáticas, distribuciones de probabilidad, modelos ARMA, procesos en las líneas de espera, y si no se conoce la fórmula correcta o se implementa mal, el sistema tendrá errores en el resultado y se tomará la decisión equivocada.

5.3 Características del equipo de cómputo (PC).

Este es un punto importante, debido a que las características bajo las que se programa en una computadora frecuentemente no son las mismas que en las computadoras de los usuarios, por lo que ocasionalmente se presentan errores al utilizar algún software, es curioso pero en ocasiones un programa se ejecuta correctamente en una computadora y al cambiar de máquina no funciona adecuadamente.

Las características a las que hago mención pueden ser entre otras: sistema operativo, procesador, velocidad, co-procesador matemático, memoria RAM, tarjeta de video, tarjeta de audio, unidades de disco flexible (almacenamiento).

Los requerimientos que maneja SIMULAT no son muchos, pero no es difícil que al utilizarlo en alguna máquina con determinadas características, no se pueda ejecutar correctamente, o simplemente, no funcione, como sería en el caso de una computadora que no tuviera disco duro, o bien, no tenga Windows®.

5.4 Creatividad del desarrollador (programador).

Aquí nos enfrentamos a la forma en que el programador resuelve los problemas que se le presentan durante el desarrollo del software, al estar programando, revisar el código, por lo que se pueden cometer mil errores o mil genialidades al implementar fórmulas o procedimientos, mejorar el código ya realizado o simplemente al corregir un error.

5.5 Comentarios acerca de SIMULAT.

Se han encontrado algunos errores en SIMULAT y se han resuelto de la mejor forma posible, esta sección explicará cada uno. Existen errores que no se pudieron resolver debido a cuestiones del software o por no saber cómo resolver determinado problema de programación.

5.5.1 Desglose de los resultados.

Al mostrar en pantalla los resultados, se presentaba un error debido a que la capacidad de la caja de texto es limitada y al desplegar los resultados de la simulación se podía sobrepasar esa capacidad, por lo que se truncaba la ejecución. La solución que se le dio a este problema fue que al exceder la capacidad se almacenará en archivos adicionales numerados, por ejemplo ALEA01.DAT. Al realizar esto no se pueden imprimir los primeros resultados, solo se imprimirá la última pantalla de texto, esto se puede corregir, pero la forma en que SIMULAT presenta los resultados, implicaría que se utilizaran demasiadas hojas lo cual hace muy difícil el análisis de los resultados.

5.5.2 Tamaño de las colas.

La idea original acerca del tamaño de las cola fue que tuvieran tamaño infinito, pero al hacer esto se desbordaría la capacidad de memoria de la stack, lo cual provocaría que la ejecución se truncara, o bien la computadora se congelara. Esto es lógico, ya que se utilizaba toda la memoria, la solución fue limitar el tamaño de la cola lineal a 2500 transacciones, esto por que Delphi no permite apuntadores mayores

La cola tipo "FIFO" (First in, first out) no presenta problemas con su tamaño, ya que al ser una cola circular es de tamaño infinito.

La cola tipo "LIFO" (Last in, first out) es de tipo lineal, pero por lo comentado anteriormente se generan 2500 transacciones, pero debido a que con cada transacción que sale de la cola se libera espacio en memoria, pueden crearse un poco más.

5.5.3 Muestra de los resultados de las colas.

No es precisamente un error al presentar los resultados de las líneas de espera, sino que es una omisión, no se puede ver lo que pasa con la cola en determinados momentos del tiempo, como lo permiten otros lenguajes de simulación.

5.5.4 Ayuda.

SIMULAT es un lenguaje que no cuenta con ayuda como tal, en lugar de eso, se presentan ventanas con información acerca de las instrucciones y de los datos que se deben de ingresar.

5.5.5 Animación.

Muchos lenguajes de programación orientados a la simulación tienen entre sus características la opción de mostrar una animación de lo que va ocurriendo durante la simulación. En el proyecto original se pretendía añadir animación a los diferentes procedimientos, pero dado la complejidad de esto, y que no es el objetivo del presente trabajo, se omitió.

5.5.6 Tipo de lenguaje.

Existen dos tipos de lenguajes:

- Interpretes (reconocedor de sintáxis): que al momento de ejecutar el código fuente se revisa línea por línea y muestran los errores existentes.
- Compiladores: que revisan el código fuente y pueden crear un programa ejecutable, es decir, traduce un lenguaje de alto nivel a lenguaje máquina[2].

SIMULAT es un interprete, debido a que revisa el código fuente línea por línea y después realiza las funciones programadas. El no crear un programa ejecutable no es ningún problema para un lenguaje de simulación, al contrario, sería una desventaja, ya que al momento de ejecutarse, se simularían exactamente los mismos procesos una y otra vez.

5.5.7 Comentarios.

Generalmente, los lenguajes de programación permiten incluir comentarios dentro del código fuente, pero, SIMULAT no permite esto.

5.5.8 Funciones Polinomiales.

Con SIMULAT se puede simular cualquier función polinomial (con exponentes enteros positivos), al combinarse las operaciones básicas en forma adecuada, por ejemplo:

$$x^3 + 2x^2 - 9x + 3$$

El código para simular esta función siendo x una variable aleatoria con distribución normal estándar:

```
SUMA (NM(0,1)*,0)
X (RESULTADO)
POT (X,3)
Y (RESULTADO)
POT (X,2)
MULT (RESULTADO,2)
```


[2] El lenguaje máquina es el usado directamente por la computadora, en el cual las instrucciones se encuentran en código binario.

```
SUMA (Y,RESULTADO)
Y (RESULTADO)
MULT (X,-9)
SUMA (Y,RESULTADO)
SUMA (RESULTADO,3)
```

5.6 Pruebas.

Las fórmulas en SIMULAT se desarrollaron de acuerdo con las distribuciones expuestas y al realizar pruebas de hipótesis con el software STATGRAPHICS 7.0 a los resultados de las simulaciones de 500, 1000 y 2500[3] variables aleatorias[4], dando los resultados que la distribución simulada era la adecuada.

Al hacer un análisis de series de tiempo con STATGRAPHICS 7.0 a los resultados obtenidos de las instrucción ARMA de SIMULAT, se obtiene el modelo ARMA(p.q) simulado, pero los valores de los parámetros θ de la serie no son exactos, esto es debido a que existe un error en STATGRAPHICS 7.0 al calcular estos parámetros.

[3] No en todas las distribuciones se pudieron probar con 2500 valores.

[4] Se probaron todas las distribuciones aceptadas por SIMULAT.

CAPÍTULO 6
APLICACIONES DE SIMULAT

Existen diversas aplicaciones para el lenguaje SIMULAT, en este capítulo sólo se presentan algunas de las posibles aplicaciones, no son las únicas, ni tampoco es regla utilizar SIMULAT en las que aquí se presentan, será el usuario quien determine si el uso de este lenguaje de simulación es el adecuado para sus fines o no.

6.1 Variables aleatorias.

El uso de las variables aleatorias permite la simulación de eventos aleatorios independientes entre sí, como:

◊ Procesos de producción: Para establecer políticas de producción de una empresa, se puede simular el comportamiento de la producción, cuantos productos pueden salir de los límites de calidad, el tiempo esperado para lograr una cantidad determinada de productos, la cantidad de productos defectuosos o con una característica específica.

◊ Procesos de Poisson: Muchos eventos de la vida diaria se comportan como un proceso de Poisson, las líneas de espera, la número de eventos en un tiempo específico (o espacio) como el número de votantes que asisten a una casilla a determinada hora del día.

◊ Características de una población: Para realizar estudios sociológicos, políticos y económicos se simularían los resultados de encuestas, las tendencias políticas de un sector de la población, los ingresos poblacionales, los miembros de la población con determinada característica.

◊ Problemas de vialidad: Para evitar los problemas de transito se puede simular el tiempo entre la llegada de cada vehículo, su posible destino y el tiempo de recorrido de las posibles vías de transito.

◊ Valores esperados: Se utilizan para saber como se pueden comportar las ventas a lo largo de un año, las ganancias esperadas de la venta de un artículo.

Como se puede ver el uso de las variables aleatorias es amplio y sólo esta limitado por la imaginación del analista .

6.2 Modelos autorregresivos y de medias móviles.

Las series de tiempo sirven para simular problemas que implican el tiempo, que es un factor para que ocurra un cambio, algunos posibles ejemplos son:

◊ En Ecología se pueden simular los índices de contaminación, para preparar políticas de salubridad.

◊ En salubridad, para tener personal necesario en hospitales, se puede simular el número de nacimientos, al igual que número de muertes (por tipo: accidente, natural, violencia).

◊ En Economía se pueden simular el tiempo entre las devaluaciones para evitar grandes perdidas en la economía de un país, también se puede simular el valor de las acciones de las empresas.

◊ Para brindar mayor seguridad a la sociedad se puede simular el número de asaltos por zona.

◊ Estudios de crecimiento poblacional.

6.3 Procesos markovianos.

Se utiliza para simular el cambio de estado (característica) de una entidad en especial, los posibles ejemplo son:

◊ Para tomar políticas de adquisición de una máquina se puede simular su vida util, y los gastos que se deriven de las reparaciones.

◊ Movimientos de un partícula suelta en el aire.

◊ El estado de convalecencia de una persona enferma.

◊ El ciclo de los diversos componentes químicos que existen en el ambiente.

◊ Procesos de nacimiento-muerte.

6.4 Líneas de espera.

Las diferentes aplicaciones que se tienen para las líneas de espera son relacionadas con Teoría de Colas, líneas de servicio y producción, control de inventarios.

◊ Número de cajeros necesarios en un banco a determinada hora.

◊ Acceso a un lugar multitudinario.

◊ Cantidad de servicios sanitarios en lugares multitudinarios.

◊ Comportamiento en la espera de un servicio.

6.5 Funciones polinomiales.

Las funciones polinomiales se pueden utilizar para simular procesos físicos ó económicos, por lo que son de mucha utilidad en Física, Economía, Administración, Econometría y Contabilidad.

CONCLUSIONES

Las conclusiones a las que se ha llegado a los largo del desarrollo del presente trabajo, son las siguientes:

◊ Las aplicaciones de la simulación son muy amplias y cada día se dan a conocer nuevas aplicaciones, por lo que las instituciones de educación superior deben de formar profesionistas con la capacidad de realizar e interpretar los experimentos de simulación.

◊ Las empresas (según su giro) deben de capacitar a su personal, para que sean capaces de realizar e interpretar experimentos de simulación.

◊ Existen pocas herramientas computacionales de simulación disponibles en el mercado, por lo que hace falta desarrollarlas en diversas áreas de aplicación que tiene la simulación, debido a que una sola herramienta puede abarcarlo todo, pero siempre se quedarán pequeños detalles pendientes.

◊ Existen en el mercado diferentes tipos de software estadístico, pero cada uno tiene diferentes deficiencias (como se mencionó acerca del STATGRAPHICS), sin embargo ayudan a la realización de experimentos de simulación.

◊ Las diferentes herramientas para el desarrollo de software (lenguajes) ofrecen grandes ventajas para el desarrollo de software estadístico y de simulación; pero ofrecen desventajas a los usuarios finales, que se ven obligados a tener cierto sistema operativo, un software determinado o un equipo específico, para ejecutar la aplicación obtenida.

BIBLIOGRAFÍA

1) DALE, Nell & WEEMS, Chip
PASCAL. SEGUNDA EDICIÓN
McGraw-Hill
México, 1991.

2) FREUND, John & WALPOLE, Ronald
ESTADÍSTICA MATEMÁTICA CON APLICACIONES
Prentice Hall
México, 1990.

3) GONZÁLEZ Videgaray, MariCarmen
MODELOS DE DECISIÓN CON PROCESOS ESTOCÁSTICOS II (METODOLOGÍA DE BOX-JENKINS)
Universidad Nacional Autónoma de México
México, 1990.

4) GONZÁLEZ Videgaray, MariCarmen
MODELOS Y SIMULACIÓN
Universidad Nacional Autónoma de México
México, 1996.

5) GOTTFRIED, Byron
ELEMENTS OF STOCHASTIC PROCESS SIMULATION
Prentice Hall
Estados Unidos, 1984.

6) HILLIER, Frederick S. & LIEBERMAN, Gerald J.
INTRODUCCIÓN A LA INVESTIGACIÓN DE OPERACIONES
McGraw-Hill Inc.
México, 1997

7) LAW, Averill M, & KELTON, W. David
SIMULATION MODELING & ANALYSIS
McGraw-Hill Inc.
Estados Unidos, 1991.

8) MANNING, Michelle M.
DELPHI 2. GUÍA OFICIAL DE BORLAND
Prentice Hall
México, 1996.

9) MATCHO, Jonathan; SALMANOWITZ, Brian & STROOL, Scott
EDICIÓN ESPECIAL: DELPHI 2
Prentice Hall
México, 1996.

10) MENDENHALL, William; WACKERLY, Dennis & SCHEAFFER, Richard
ESTADÍSTICA MATEMÁTICA CON APLICACIONES
Grupo Editorial Iberoamericana
México, 1994.

11) PRAWDA Witenberg, Juan
MÉTODOS Y MODELOS DE INVESTIGACIÓN DE OPERACIONES
Vol. 2 Modelos estocásticos.
LIMUSA
México, 1996.

12) PRITSKER, A. Alan & PEGDEN, Claude Dennis
INTRODUCTION TO SIMULATION AND SLAM
A HALSTED PRESS BOOK; JOHN WILEY & SONS
Estados Unidos, 1979.

13) RODARTE Cordova, Pedro Alberto
MANUAL DE DELPHI PARA DESARROLLADORES
Petróleos Mexicanos
México, 1997.

14) RODRÍGUEZ Moreno, Guadalupe del Carmen
SIMULACIÓN PARA LA SOLUCIÓN DE PROBLEMAS DINÁMICOS ESTOCÁSTICOS Y DE OPTIMIZACIÓN Y CONTROL
Tesis para obtener el grado de licenciatura
México, 1996.