



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

L3
29^o

FACULTAD DE CONTADURIA Y ADMINISTRACION

MODELOS DE SIMULACION PARA
SISTEMAS DE INVENTARIOS Y LINEAS
DE ESPERA

SEMINARIO DE INVESTIGACION INFORMATICA

QUE EN OPCION AL GRADO DE
LICENCIADO EN INFORMATICA
P R E S E N T A :
RAUL OJEDA VILLAGOMEZ

ASESOR: M.C. MANUEL PIÑUELA DEL RIO

1993

TESIS CON
FALLA DE ORIGEN



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.

| | |
|---|----|
| INTRODUCCION | 1 |
| <u>I. CONCEPTOS BASICOS</u> | 4 |
| 1.1 DEFINICION DE SIMULACION | 4 |
| 1.2 DEFINICION DE MODELO | 6 |
| 1.3 CLASIFICACION DE LOS MODELOS DE SIMULACION | 7 |
| 1.4 OBJETIVOS QUE SE DEBEN TENER EN CUENTA | 12 |
| 1.5 EL PROCESO DE LA SIMULACION | 12 |
| <u>II. GENERACION DE NUMEROS ALEATORIOS</u> | 15 |
| 2.1 PROPIEDADES DE LOS NUMEROS ALEATORIOS DISTRIBUIDOS UNIFORMEMENTE | 16 |
| 2.2 GENERACION DE NUMEROS ALEATORIOS | 18 |
| 2.3 METODO DE LOS CUADRADOS MEDIOS | 19 |
| 2.4 METODO DE LEHMER | 20 |
| 2.5 METODOS CONGRUENCIALES | 21 |
| 2.6 COMPROBACION DE UN GENERADOR DE NUMEROS ALEATORIOS | 23 |
| 2.6.1 Pruebas sobre la uniformidad de la distribución | 24 |
| 2.6.2 Pruebas de buen ajuste | 24 |
| 2.6.3 Test de Kolmogorov-Smirnov | 26 |
| 2.6.4 Test de la J_i cuadrada | 28 |
| 2.6.5 Pruebas de Corridas | 31 |
| 2.6.6 Pruebas de Autocorrelación | 36 |
| 2.6.7 Nivel de Significación | 41 |

| | |
|---|-----|
| <u>III. GENERACION DE PROCESOS</u> | 42 |
| 3.1 METODO MONTECARLO | 42 |
| <u>IV. SIMULACION DE SISTEMAS DE COLAS</u> | 46 |
| 4.1 COMPORTAMIENTO DE LAS LINEAS DE ESPERA | 46 |
| 4.1.1 SISTEMA DE CANAL SIMPLE | 47 |
| 4.1.2 CANALES MULTIPLES EN PARALELO | 50 |
| 4.1.3 CANALES MULTIPLES EN SERIE | 51 |
| 4.2 CONSTRUCCION DEL MODELO DE LINEA DE ESPERA | 52 |
| 4.3 RECOLECCION DE DATOS | 53 |
| 4.4 PROGRAMA DE SIMULACION | 57 |
| <u>V. SIMULACION DE SISTEMAS DE INVENTARIOS</u> | 63 |
| 5.1 MODELO DE COMPRA SIN ESCASEZ | 64 |
| 5.2 PROBLEMA DE UN CONTROL DE INVENTARIOS | 68 |
| 5.3 PROGRAMA DE SIMULACION | 75 |
| <u>VI. CONCLUSIONES Y RECOMENDACIONES</u> | 85 |
| ANEXO 1: Referencia rápida del GPSS/H | 88 |
| ANEXO 2: Temario Matemáticas IV | 90 |
| APENDICE "A" TABLAS ESTADISTICAS | 96 |
| APENDICE "B" | |
| GENERAL PURPOSE SIMULATION SYSTEM / HIGH (GPSS/H) | 99 |
| APENDICE "C" | |
| SALIDA DE UN PROGRAMA GENERAL DE INVENTARIOS | 151 |
| BIBLIOGRAFIA | 176 |

INTRODUCCION

El objetivo de este trabajo es apoyar como material didáctico en la asignatura de Investigación de Operaciones plan 1993, en el capítulo IV y V ver anexo.

En los modelos de control de inventarios, es necesario determinar la demanda la cual en la vida real nunca se sabe, qué solicitud de pedidos vamos a tener o en qué tiempo exactamente nos van a surtir el inventario; por lo tanto se requiere el uso de simulación, para utilizar dicho modelo más efectivamente. En el caso de las filas de espera la frecuencia con la que llegan los clientes a los servidores, también la desconocemos, y es por lo que tenemos que incluir la incertidumbre en cada uno de nuestros modelos para que se ajuste a la realidad; es por esto que en este trabajo presento para apoyar a la impartición de dichos temas un programa interactivo el cual fue realizado utilizando el lenguaje de computadora GPSS/H (sistema de simulación de propósito general para PC); con dicho programa los alumnos podran hacer su casos prácticos y simular el comportamiento de las variables que intervengan en los modelos de inventarios y líneas de espera.

Se platicó con algunos maestros que imparten la materia de Investigación de Operaciones y han sugerido que se elabore un material didáctico para abordar el tema de simulación de sistemas.

Esta obra trata de cubrir cada uno de los aspectos que integran la simulación, sin profundizar, pero dejando una idea clara de tal manera que pueda utilizarse en cualquiera de las tres carreras (Administración, Contaduría e Informática), que se imparten en la Facultad. Aportando con esto una herramienta para que el alumno aborde las partes de la simulación de una manera interactiva, en donde la teoría se pueda consultar en esta obra y cada uno de los conocimientos los pueda ejercitar, poniéndolos en práctica en la computadora, como es la generación de números aleatorios, los test para verificar la confiabilidad de los números aleatorios, la distribución de probabilidades, etc.

Cuando en la obra entremos a simular, será en ese momento cuando el lector esté consciente del por qué se utilizó la χ^2 cuadrada para bondad de ajuste o como probabilidad, el por qué de utilizar cierto generador de números aleatorios, etc. Entonces exploraremos un ejercicio en donde utilizaremos una distribución empírica para partir de lo sencillo, y después abordaremos el comportamiento que se le puede dar a ésta. Con esto tendremos que el alumno estará capacitado para realizar un modelo de simulación y predecir el comportamiento. Al finalizar se darán las pautas para obtener la optimización de un modelo de simulación.

El énfasis que se le da a los primeros capítulos que son la generación de números aleatorios, su comprobación, las pruebas y otros aspectos que son transparentes al utilizar

un lenguaje de simulación, es para, que se conozca cómo hacerlo manualmente, pero haciendo hincapié de que se obtienen mejores resultados y en mucho menor tiempo hacerlo computacionalmente, pero el proceso de aprendizaje se debe realizar desde el inicio para que se comprenda bien la idea, tratando de utilizar y explotar todas las ventajas que nos trae el diseñar una buena simulación.

La simulación es un método para analizar sistemas, y el modelo de simulación es un medio para reunir información sobre el rendimiento de un sistema, basándose en las condiciones establecidas por el analista.

El alcance de este trabajo es que el lector vía el paquete GPSS/H, pueda analizar el modelo de lote económico clásico (en inventarios) y los modelos de líneas de espera, y poder incorporarlos a un modelo real de una empresa con todas las características y variantes que puede traer consigo mismo el poder adaptarla.

I. CONCEPTOS BASICOS

A medida que pasan los años observamos la imperiosa necesidad de recurrir a la simulación, para no incurrir en costos que pueden ocasionar pérdidas graves. Cuando estuve en Japón observé que en las piscinas sacaban a los usuarios cada hora, entonces una persona se iba a la mitad de la piscina ahí "fingía" estar ahogado y entre 5 salvavidas nadaban hasta ese lugar y lo sacaban del agua, 4 lo iban cargando y otro le daba respiración artificial con un aparato. A todo esto lo podríamos llamar ¿Engaño? o bien ¿Planeación?, ¿Simulacro? ó bien ¿Simulación?. Bueno esto se determina dependiendo del objetivo al que se quiera llegar.

1.1 DEFINICION DE SIMULACION

Según el diccionario de la real academia española:

- ° Simular es "fingir, llegar a la esencia de algo prescindiendo de la realidad".
- ° Simular v.t. (lat. simulare). Dar la apariencia de algo que no es; simular una enfermedad, un combate. II Fingir una cosa.
- ° Simulación f. (lat. simulatio) Acción de simular o fingir. (Sinón. V. Disimulo)
- ° Simulacro m. (lat. simulacrum). Imagen, estatua. II Fantasma aparición visión: ver en visión vanos simulacros. II Apariencia sin realidad: En tiempos de Julio César no había en Roma más que un simulacro de República. II Representación acción simulada: un simulacro de combate.
- ° Planear v.t. Trazar el plan de una obra, imaginar. (Sinón. V. proyectar.) II Organizar. II v. i. Cernerse en el aire como las aves. II Vuelo planeado o cernido, el que ejecuta un aeroplano sin utilizar el motor.

- ° Engañar v. t. Hacer caer en un error. (Sinón. Mentir, embaucar, engatusar, chasquear, frustrar, defraudar, traicionar, trampear, burlar. Fam. Abusar de, timar, enredar, liar.) II Producir ilusión: la perspectiva suele engañar nuestros sentidos. II V. r. Equivocarse. (Sinón Equivocar.) II. Cerrar los ojos a la verdad.
- ° Fingir v.t. (lat. fingere). Dar a entender lo que no es: fingir una enfermedad. II Aparentar, simular. II - Sinón. Encubrir, pretextar, hacer como que.

Según la enciclopedia de términos de microcomputación de Linda Gail Christie, Ed. Pretince Hall:

- ° Simular: Capacidad de un sistema para demostrar o imitar parte o todo el comportamiento de otro sistema.
- ° Simulación: Técnica para demostrar o imitar situaciones físicas empleando computadoras. La simulación permite a los usuarios estudiar el comportamiento del sistema sin el riesgo de errores reales. La simulación se utiliza cuando la experimentación directa resulta imposible, no es práctica, no es económica, es inmoral o demasiado lenta.

En el año 1949 en donde inicia el uso de la palabra simulación en su concepción actual con el método de simulación Monte Carlo, dado por John Von Neumann y Stanislaw Ulam, se encuentra ligado con la de Robert Shannon en 1975, que recoge esencialmente el enfoque que a lo largo de las décadas se ha dado: "Simulación es el proceso de diseñar un modelo de un sistema real y realizar experimentos con él para entender el comportamiento del sistema o evaluar varias estrategias para la operación del sistema". En la que estoy de acuerdo con él pero no sólo de un sistema real si no también de sistemas que se desean construir realizando prototipos.

La simulación puede o no implicar un modelo estocástico y una experimentación de tipo Monte Carlo. Esto significa, las entradas al modelo y/o las relaciones funcionales entre sus diversos componentes pueden o no implicar un elemento aleatorio sujeto a las reglas de probabilidad. El modelo de la simulación es una metodología aplicada y experimental que intenta:

- a) Describir el comportamiento de sistemas.
- b) Postular teorías o hipótesis que expliquen el comportamiento observado.
- c) Usar estas teorías para predecir un comportamiento de los efectos que se producirán mediante cambios en el sistema o en su método de operación.

1.2 DEFINICION DE MODELO

Un modelo es la representación de un objeto, sistema, o idea, de forma diferente a la identidad misma (Shanon, 1988).

Se considera que las funciones de un modelo son la predicción y la comparación, para proporcionar una manera lógica de predecir los resultados que siguen las acciones alternativas, indicando una preferencia entre ellas.

El concepto de la representación de un objeto nos sirve como:

- a) Una ayuda para el pensamiento.

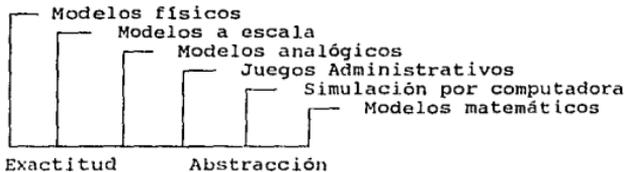
- b) Una ayuda para la comunicación.
- c) Para entrenamiento e instrucción.
- d) Una herramienta de predicción.
- e) Una ayuda para la experimentación.

1.3 CLASIFICACION DE LOS MODELOS DE SIMULACION

Shanon manifiesta dos clasificaciones para los modelos de manera general y los modelos de simulación en particular, con lo cual tenemos una serie de combinaciones para representar nuestros sistemas. Los esquemas para simulación son:

- a) Estático (de corte seccional) vs. dinámico (de series de tiempo)
- b) Determinístico vs. estocástico.
- c) Discreto vs. continuo.
- d) Icónico o físico vs. analógico vs. simbólico.

Los modelos de simulación son como un espectro continuo, empezando con los modelos exactos o modelos reales a escala y siguiendo con los modelos matemáticos completamente abstractos.



Los modelos al principio del espectro continuo, se les llama comúnmente físicos o icónicos, debido a que se asemejan al sistema de estudio.

Los modelos físicos son los que de alguna manera semejan a la entidad que se está modelando, pueden ser estáticos o dinámicos. Los modelos físicos estáticos ayudan a visualizar las relaciones espaciales, tales como los modelos arquitectónicos y los modelos de diseño de planta. Los modelos físicos dinámicos sirven para ver el comportamiento del sistema, un ejemplo sería el de una planta piloto para estudiar un nuevo proceso químico antes de proceder a la producción en gran escala.

Los modelos físicos por su escala pueden ser de tamaño natural (tales como los aparatos de entrenamiento), de menor escala (un sistema solar), o de mayor escala (modelo de un átomo). Estos también pueden ser bi o tridimensionales. Las maquetas a escala son usadas en los estudios de diseño de plantas son un ejemplo de un modelo físico tridimensional, a menor escala, utilizada para la experimentación.

Los modelos analógicos son aquellos en los que una propiedad del sistema real esta representada por una propiedad sustituida, que por lo general se comporta de manera similar. Algunas veces, el problema se resuelve en el estado análogo y la respuesta se traslada a las propiedades originales. Una gráfica es un modelo analógico, en el cual

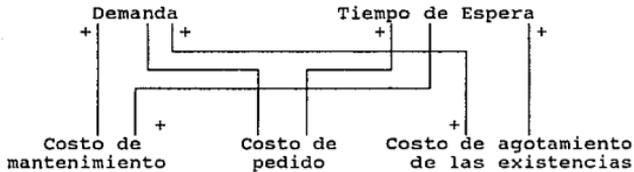
la distancia representa las propiedades tales como tiempo, edad, número, y otras más.

En los juegos administrativos, el hombre interactúa con la salida de computadora, la cual simula todos los otros aspectos del sistema y toma decisiones acerca de la información recibida. Después sus decisiones se retroalimentan a la computadora como entradas al sistema. Esto nos puede llevar a las simulaciones totalmente computarizadas que la gente visualiza por lo general cuando se usa el término simulación.

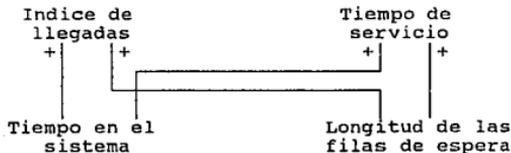
Los modelos matemáticos o simbólicos son aquellos en los que se usa un símbolo, en vez de un dispositivo físico, para representar una entidad. En consecuencia, en un modelo matemático, podríamos usar símbolos tales como "x" y "y" para representar el volumen de producción y el costo, en vez de una escala medida o alguna entidad física. Debido a que estos modelos son los más abstractos y, por lo tanto, los más generales, son los de uso más generalizados en estudios de sistemas.

Por lo general, al tratar de modelar un sistema complejo, el investigador utilizará una combinación o más de uno de los tipos individuales que se acaban de presentar. Usualmente, los modelos simples llevan a modelos más complejos, conforme el investigador analiza y comprende mejor el problema.

Se verá que cuando se emplean valores óptimos para los componentes Q , la cantidad del pedido, y R , el punto de repedido, la demanda que pasa sobre el sistema de inventarios y el tiempo de espera del reabastecimiento (el tiempo necesario para completar las existencias) se relacionan con el costo de mantenimiento, el costo de hacer los pedidos y el costo del agotamiento de las existencias, como se muestra en la siguiente figura (tomada de McMillan, 1986).



En el capítulo 4 se examinará un sistema de líneas de espera, cuyos componentes y cuya interacción se representan en la siguiente figura:



Al modelar éstos sistemas, se empleara el análisis formal y un lenguaje de simulación GPSS/H, cubriendo los

componentes que interactúan entre sí y explicando cada una de ellas.

1.4 OBJETIVOS DE LA SIMULACION QUE SE DEBEN TENER EN CUENTA

Se ha dicho que la simulación es el acto de realizar experimentos con un modelo de un sistema. Para que ésta pueda ser una técnica eficiente de análisis para la determinación de normas eficaces y racionales, hay diferentes objetivos que es preciso tener en consideración. Esos objetivos, en el proceso de simulación, son de tres tipos:

a) *Objetivo del sistema que se está estudiando.* El objetivo consiste en utilizar los recursos asignados de tal modo que se optimicen una o varias cantidades que se reconocen como meta del sistema.

b) *Objetivo del modelo de simulación.* El objetivo del modelo de simulación en computadora consiste en generar eficientemente estadísticas de salida, estas son los parámetros que se deben conocer para ajustar el modelo de la aplicación.

c) *Objetivo del analista de simulación.* Este objetivo consiste en distribuir los recursos del presupuesto del proyecto de simulación de tal modo que se maximicen los beneficios esperados. Esto quiere decir que el objetivo del

analista de simulación consiste en determinar normas eficaces con tanta eficiencia como sea posible.

Existen prioridades entre esos objetivos: el del sistema que se estudia debe preceder necesariamente a todos los demás; la realización o el alcance de ese objetivo es primordial para la organización, el segundo en importancia es el objetivo del analista.

La simulación es un método para analizar sistemas. Es preciso saber que el modelo de simulación es un medio para reunir información sobre el rendimiento de un sistema, basándose en las condiciones establecidas por el analista.

1.5 EL PROCESO DE LA SIMULACION

Si la simulación se usa para investigar las propiedades y el comportamiento de un sistema real, se deben mencionar las siguientes etapas:

1. Definición de sistema: Determinación de los límites o fronteras restricciones y medidas de efectividad que se usarán para definir el sistema que se estudiará.
2. Formulación del modelo: Reducción o abstracción del sistema real a un diagrama de flujo lógico.

3. Preparación de datos: Identificación de los datos que el modelo requiere y reducción de éstos a una forma adecuada.
4. Translación del modelo: Descripción del modelo en un lenguaje aceptable para la computadora que se usará.
5. Validación: Incremento a un nivel aceptable de confianza de modo que la inferencia obtenida del modelo respecto al sistema real sea correcta.
6. Planeación estratégica: Diseño de un experimento que producirá la información deseada.
7. Planeación táctica: Determinación de cómo se realizará cada una de las corridas de prueba especificadas en el diseño experimental.
8. Experimentación: Corrida de la simulación para generar los datos deseados y efectuar el análisis de sensibilidad.
9. Interpretación, y obtención de inferencias con base en datos generados por la simulación.
10. Implantación, y uso del modelo y/o resultados.
11. Documentación, registro de las actividades del proyecto y los resultados así como de la documentación del modelo y su uso.

Con los pasos anteriores se supone que el problema puede resolverse de una mejor manera mediante la simulación. A menudo se ha dicho que la simulación es un planteamiento aproximado de un último recurso para resolver problemas. En realidad, es cierto que cuando un problema puede reducirse a un modelo simple y resolverse analíticamente, la simulación no es necesaria. Debería investigarse todas las herramientas disponibles para manejar cada problema y optimizar entre los resultados y el costo. Se debe tener en cuenta que un modelo analítico simple es inadecuado, y es mejor considerar la posibilidad de la simulación.

II. GENERACION DE NUMEROS ALEATORIOS

En los experimentos de simulación, así como también en la mayoría de los experimentos de muestreo, existe la necesidad de contar con una fuente de números aleatorios.

La importancia de los números aleatorios es actuar como herramienta para incluir la incertidumbre en los modelos de simulación, ya que deben representar los insumos del modelo real que se intenta simular para que pueda predecir adecuadamente su comportamiento, lo que se logrará vía un experimento aleatorio, que es cualquier situación en la cual se puede enumerar de antemano los resultados posibles de los números aleatorios sin que se conozca cuál de ellos se va a presentar, aunque sí sus probabilidades, y este se puede realizar en las mismas condiciones un número indefinido de veces.

El conjunto de los posibles resultados asociados al experimento aleatorio recibe el nombre de **espacio muestral**, y las **variables aleatorias** son funciones definidas sobre el espacio muestral para las que es posible realizar afirmaciones probabilísticas.

2.1 PROPIEDADES DE LOS NUMEROS ALEATORIOS DISTRIBUIDOS UNIFORMEMENTE

Estos se pueden manifestar en las variables aleatorias discretas y continuas.

VARIABLES ALEATORIAS DISCRETAS

Las variables aleatorias discretas se caracterizan porque la diferencia entre dos valores finitos cualesquiera que pueda tomar la variable, es finita y distinta a cero. Ejemplo el número de personas que esperan en una parada de autobús.

La función de distribución acumulativa, $F_X(k)$, de la variable aleatoria X se define como:

$$\begin{aligned} F_X(k) &= \text{Probabilidad de que el valor de } X \text{ sea } \leq \text{ que } k. \\ &= P(X \leq k) \end{aligned}$$

La probabilidad de que X se encuentre entre a y b , ($a < b$) se calcula restando $F_X(a)$ de $F_X(b)$. O sea:

$$P(a < X \leq b) = F_X(b) - F_X(a)$$

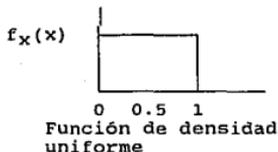
En la mayoría de los casos sólo necesitaremos las variables aleatorias discretas que pueden tomar valores enteros. En tal caso se pueden relacionar mediante:

$$F_X(k) = \sum_{X \leq k} P_X(X)$$

Variabes aleatorias continuas

Una variable aleatoria continua posee la propiedad de que en cualquier intervalo abierto finito (un intervalo abierto es el que no incluye los puntos extremos) en que se define, puede tomar un número infinito de valores. Las variables aleatorias continuas que interesan a los analistas de sistemas incluyen por lo común el tiempo, en alguna forma. La distribución acumulativa esta dada por:

$$F_X(k) = \int_{-\infty}^k f_X(x) dx.$$



La media de una distribución uniforme continua definida sobre el intervalo $(0,1)$, debe ser $1/2$.

Si se divide el intervalo $(0,1)$ entre n clases o subintervalos, el i ésimo subintervalo debería contener N/n observaciones, siendo N el total de observaciones. De esta manera, si hacemos 1000 observaciones de las variables a partir de la distribución uniforme, poniéndolas en una distribución de frecuencias que contuviera 10 intervalos espaciados por igual, podríamos esperar tener una frecuencia de $1000 / 10 = 100$ observaciones en cada intervalo. Esta propiedad debe ser válida, sea cual sea el tamaño de los intervalos.

La probabilidad de observar un valor en un intervalo permanece constante y es independiente del valor obtenido anteriormente.

La propiedad de los eventos equiprobables independientes debe ser válida, de manera ideal, para cualquier nivel de significación que se emplee, es decir deben tener la misma posibilidad de ocurrir.

El tamaño de la muestra, sea cual sea el tamaño deberá reflejar las características de la distribución teórica.

2.2 GENERACION DE NUMEROS ALEATORIOS

Las propiedades que debe tener un generador para obtener variables aleatorias uniformemente distribuidas son:

1) La serie de números que produce ese generador debe seguir la distribución uniforme ideal tan cercana como sea posible.

2) Debe tener un periodo largo. El periodo de un generador de números aleatorios es una medida de la cantidad de números. Cuando comienza a reaparecer la misma secuencia de números, se dice que el generador comienza a reciclarse.

3) El generador debe poder reproducir las mismas series de números que se desee (esta propiedad lo manejan otros autores como pseudoaleatorios, ya que al reproducirse

pierden la característica de aleatoriedad). Por otra parte, debe tener capacidad para producir, a voluntad, un conjunto de números claramente distintos.

4) El generador debe de ser de naturaleza no degenerativa. La degeneración significa que el generador produce continuamente el mismo número.

El autor Leandro Pardo en su obra señala 6 propiedades de los números aleatorios, que el denomina "números pseudoaleatorios" que son:

- a) Tener una distribución uniforme.
- b) Ser estadísticamente independientes.
- c) Ser reproducibles.
- d) Tener un ciclo no repetitivos tan largo como se desee
- e) Generar números pseudoaleatorios a alta velocidad.
- f) Ocupar poca capacidad de memoria de ordenador.

2.3 METODO DE LOS CUADRADOS MEDIOS

Este método de generación de números aleatorios, fundamentado en unos resultados de Von Neuman, consiste en tomar un número al azar, X_0 , de $2n$ cifras, en elevarlo al cuadrado, después tomar de este resultado las $2n$ cifras centrales, y repetir indefinidamente la operación.

Ejemplo:

$$\begin{aligned} X_0 &= 3121 \\ X_1 &= 7406 \end{aligned}$$

$$\begin{aligned} (X_0)^2 &= 9|7406|41 \\ (X_1)^2 &= 54|8488|36 \end{aligned}$$

$$\begin{array}{rcl}
 X_2 = 8488 & (X_2)^2 = 72 & | 0461 | 44 \\
 X_3 = 0461 & (X_3)^2 = 2 & | 1252 | 1 \\
 X_4 = 1252 & (X_4)^2 = 1 & | 5675 | 04
 \end{array}$$

Un inconveniente que se presenta es que los números generados pueden repetirse cíclicamente después de una serie corta. Por ejemplo supondrás que deseamos generar números aleatorios de cuatro dígitos y que el enésimo número generado sea 3 500, tendríamos:

$$\begin{array}{rcl}
 X_n = 3500 & (X_n)^2 = & 12\ 2500\ 00 \\
 X_{n+1} = 2500 & (X_{n+1})^2 = & 6\ 2500\ 00 \\
 X_{n+2} = 2500 & (X_{n+2})^2 = & 6\ 2500\ 00\dots
 \end{array}$$

Esto indica que esta secuencia no se podría tomar como una secuencia de números pseudoaleatorios.

2.4 METODO DE LEHMER

El método desarrollado por Lehmer (quién definió el número pseudoaleatorio como: una noción vaga que encierra la idea de una sucesión en la cual cada término es impredecible para la persona ajena al problema, cuyos dígitos se someten a cierto número de pruebas, comunes a los estadísticos, y que depende en algún sentido del uso que se dará a la sucesión) consiste en lo siguiente: Se parte de un número al azar X_0 , de n cifras, se le multiplica por un número al azar de K cifras, dando lugar a un número de n cifras del cual se resta el de K cifras de la izquierda, obteniéndose un número de n cifras del cual se resta el de K cifras que se había separad.

Ejemplo:

$$\begin{array}{r}
 X0 = 3121 \quad K = 57 \\
 3121 \times 57 = 17|7897 \\
 7897 - 17 = 7880
 \end{array}$$

$$\begin{array}{r}
 X1 = 7880 \\
 7880 \times 57 = 44|9160 \\
 9160 - 44 = 9116
 \end{array}$$

$$\begin{array}{r}
 X2 = 9116 \\
 9116 \times 57 = 51|9616 \\
 9616 - 51 = 9561
 \end{array}$$

$$\begin{array}{r}
 X3 = 9561 \\
 9561 \times 57 = 54|4977 \\
 4977 - 54 = 4923
 \end{array}$$

$$\begin{array}{r}
 X4 = 4923 \\
 4923 \times 57 = 28|0611 \\
 611 - 28 = 583
 \end{array}$$

Esta puede ser a partir de un cierto rango considerada como una secuencia de números pseudoaleatorios.

2.5 METODOS CONGRUENCIALES

Los métodos de generación de números pseudoaleatorios llamados congruenciales se basan en el concepto matemático de números congruentes. El número entero «x» es congruente con «y» módulo m si $x - y$ es divisible por m, es decir, «x» e «y» dan el mismo residuo cuando se dividen por m. Se suele escribir $x \equiv y$.

Ejemplo: El número 140,016 es congruente con el número 16 módulo 10^3 , ya que $140,016 - 16 = 140,000$ es divisible por 10^3 (el residuo de dividir tanto 140,016 como 16 por 10^3 es 16).

La expresión más usual en la generación de números pseudoaleatorios, utilizando números congruentes, es dada por:

$$X_{n+1} = ax_n + b \text{ (módulo } m)$$

donde a, b y m son números enteros convenientemente elegidos y x_0 recibe el nombre de semilla. Dada la semilla x_0 , la fórmula anterior permite construir una sucesión de números enteros, variando cada uno de ellos entre 0 y $m - 1$, mediante:

$$\begin{aligned} X_1 &= ax_0 + b \text{ (módulo } m) \\ X_2 &= ax_1 + b = a^2x_0 + (a + 1)b \text{ (módulo } m) \\ X_3 &= a^3x_0 + (a^2 + a + 1)b = a^3x_0 + \frac{b(a^3 - 1)}{(a - 1)} \text{ (módulo } m) \\ &\dots \\ &\dots \\ X_n &= a^n x_0 + \frac{b(a^n - 1)}{(a - 1)} \text{ (módulo } m) \quad (n \leq m - 1) \end{aligned}$$

Para obtener valores generados de una variable aleatoria uniforme en el intervalo $(0, 1)$, no hay más que considerar la sucesión $u_n = x_n/m$.

Se puede hacer que los números pseudoaleatorios generados se repitan después de un cierto periodo que depende de m .

$$x_{n+1} \equiv ax_n + b \pmod{m}$$

Esta relación implica que la suma $ax_n + b$ se debe dividir entre m y x_{n+1} es igual al residuo. La relación indica " x_{n+1} es congruente con $ax_n + b$ módulo m ".

Ejemplo: sea $m = 25$, $a = 6$ y $b = 1$

$$\begin{array}{ll} x_1 \equiv 6*1+1 \pmod{25} & x_1 \equiv 7 \\ x_2 \equiv 6*7+1 \pmod{25} & x_2 \equiv 18 \\ x_3 \equiv 6*18+1 \pmod{25} & x_3 \equiv 9 \end{array}$$

El método lo propuso inicialmente Lehmer en 1949. Con $b=0$, se denomina método multiplicativo de congruencia. Cuando adopta $b \neq 0$ se denomina método de congruencia mixta.

2.6 COMPROBACION DE UN GENERADOR DE NUMEROS ALEATORIOS

Los requisitos que debe tener una generación de números aleatorios ha de basarse primordialmente en la aplicación esperada de esos números para un experimento de simulación dado. Esto se hace determinando lo siguiente:

- 1.- Números necesarios para una corrida de simulación.
- 2.- Cuántos dígitos debe contener el número aleatorio.
- 3.- Efectos que ejercerá en los resultados una secuencia inadecuada de números aleatorios.

Otras consideraciones se destacan al analizar las diversas pruebas estadísticas.

2.6.1 Pruebas sobre la uniformidad de la distribución

Uno de los análisis básicos que se deben hacer siempre es el validar la uniformidad de la distribución. Para ello se pueden aplicar dos pruebas básicas: la prueba χ^2 cuadrada y la de Kolmogorov-Smirnov. Ambas pruebas se interesan por el grado de acuerdo que existe entre la distribución uniforme teórica. Además, las dos pruebas están basadas en una hipótesis nula de que hay una diferencia no detectable entre una distribución muestral y la teórica. Las dos pruebas se basan en el agrupamiento de datos muestrales en clases, dentro del intervalo $(0,1)$.

2.6.2 Pruebas de bondad de ajuste

Se requiere comprobar la hipótesis de que no existe diferencia entre la distribución de frecuencias de la muestra y la distribución uniforme teórica. Esta hipótesis nula especifica la proporción de observaciones que dentro de cada una de las clases de nuestra población supuesta. Esto quiere decir que, a partir de la hipótesis nula, podemos deducir las frecuencias esperadas. La prueba de χ^2 cuadrada de buen ajuste nos permite determinar si las frecuencias observadas son suficientemente cercanas a las esperadas según nuestra hipótesis nula.

Esencialmente, la hipótesis nula se puede comprobar por medio de:

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

En donde O_i = número observado en la clase i . E_i = número esperado en la clase i , y n = número de clases.

Para la distribución uniforme:

$$E_i = \frac{N}{n}$$

para clases igualmente espaciadas, donde N denota el número total de observaciones registradas.

Puede demostrarse que la distribución muestral de χ^2 , tal como se calcula a partir de la ecuación, sigue la distribución ji cuadrada con $df = n - 1$.

Consideraciones generales sobre la aplicación de esta técnica.

1) La elección del tamaño del intervalo de clase se puede escoger de diversas maneras. Mann y Wald (1942) presentaron un método para escoger el intervalo de clase basado en el nivel de significación de la prueba. Sin embargo, el pensar constantemente en la aplicación le ayudará al analista a escoger ese intervalo. Por ejemplo si el modelo de simulación utiliza los números aleatorios varias veces para hacer una entre varias elecciones

posibles, por ejemplo, K , el intervalo se deberá establecer de modo que refleje esto haciendo $n = k$.

2) El intervalo se debe escoger de manera que refleje el nivel de significación que se le exigirá al número aleatorio en el modelo de simulación.

3) El número de observaciones que se deben hacer deberá basarse en la aplicación propiamente dicha.

2.6.3 Test de Kolmogorov-Smirnov

Una vez que se ha obtenido una sucesión de números aleatorios, el test de Kolmogorov-Smirnov permite constatar si efectivamente esos números provienen de una distribución uniforme. Los pasos a seguir son los siguientes:

- Ordenar los números aleatorios, x_i , de menor a mayor y definir

$$F_n(y_i) = i/n \quad i = 1, 2, \dots, n.$$

(F_n es la distribución empírica de la muestra de los n números pseudoaleatorios simulados.) Por y_1, \dots, y_n se representan los valores x_1, \dots, x_n , una vez ordenados de menor a mayor.

- Evaluar $D_n = |F_n(y_i) - y_i| \quad 0 \leq y_i \leq 1$

(D_n es el estadístico muestral y recibe el nombre de estadístico de Kolmogorov-Smirnov).

- Fijar un valor $\alpha \in (0,1)$, llamado nivel de significación y que representa la probabilidad de rechazar que los números pseudoaleatorios provienen de una distribución uniforme cuando en realidad si provienen.
- Si $D_n \leq D_{n,\alpha}$, se concluye que efectivamente los números aleatorios provienen de una variable aleatoria con distribución uniforme en $(0,1)$. Si $D_n > D_{n,\alpha}$, se concluye que aquellos no provienen.

Ejemplo:

Supondrás que mediante un generador de números aleatorios hemos obtenido los siguientes valores: 0.36, 0.16, 0.52, 0.17, 0.88, 0.90, 0.66, 0.04, 0.93, 0.37, 0.21, 0.10, 0.28, 0.62, 0.68, 0.78, 0.94, 0.53, 0.46

Se fija un nivel de significación de $\alpha = 0.05$, y se va a comprobar si efectivamente esos 20 números provienen de una variable aleatoria con distribución uniforme en $(0,1)$. Construimos la sig. tabla:

| Valores simulados ordenados de menor a mayor | $F_n(y_i)=i/n$ | $ y_i-i/n $ |
|--|----------------|-------------|
| 0.04 | 0.05 | 0.01 |
| 0.10 | 0.10 | 0 |
| 0.16 | 0.15 | 0.01 |
| 0.17 | 0.20 | 0.03 |
| 0.21 | 0.25 | 0.04 |
| 0.28 | 0.30 | 0.02 |
| 0.36 | 0.35 | 0.01 |
| 0.37 | 0.40 | 0.03 |
| 0.46 | 0.45 | 0.01 |
| 0.52 | 0.50 | 0.02 |
| 0.53 | 0.55 | 0.02 |
| 0.61 | 0.60 | 0.01 |
| 0.62 | 0.65 | 0.03 |
| 0.66 | 0.70 | 0.04 |
| 0.68 | 0.75 | 0.07 |
| 0.78 | 0.80 | 0.02 |
| 0.88 | 0.85 | 0.03 |
| 0.90 | 0.90 | 0 |
| 0.93 | 0.95 | 0.02 |
| 0.94 | 1 | 0.06 |

Por lo tanto $D_{20}=\max|F_{20}(y_i)-y_i|=0.07$ y $D_{20,0.05}=0.294$, se confirma que efectivamente los 20 valores efectivamente provienen de una variable aleatoria con distribución uniforme en el intervalo $(0,1)$.

2.6.4 Test de la Ji cuadrada

El siguiente test permitirá comprobar la sucesión de números aleatorios que provienen de una variable aleatoria uniforme en $(0,1)$. Este test se basa en la distribución del mismo nombre con n grados de libertad.

Los pasos a seguir en este test son los siguientes:

- Agrupar los n números en K clases disjuntas I_1, \dots, I_K , e igual amplitud y contar el número de x_i que hay en la clase I_i . A este número se le designa f_i .
- Considerar la variable aleatoria

$$U = \sum_{i=1}^k \frac{(f_i - nz_i)^2}{nz_i}$$

donde z_i es la amplitud del intervalo I_i

Se prueba para n grande que U es una variable aleatoria con distribución χ^2_{k-1} .

- Fijar el nivel de significación α y consultar la distribución Ji cuadrada de $\chi^2_{k-1, \alpha}$ que representa

$$P(\chi^2_{k-1} \geq \chi^2_{k-1, \alpha}) = \alpha$$

- Si el valor de U es mayor que el valor de $\chi^2_{k-1, \alpha}$, se rechaza la hipótesis de que la secuencia de números proviene de una variable aleatoria con distribución uniforme en el intervalo $(0, 1)$.

Ejemplo: Tomando la misma sucesión de números que a la prueba anterior, considerando $K = 4$ y $\alpha = 0.05$.

Al considerar cuatro clases, éstas son:

$$I_1 = (0, 0.025), I_2 = (0.025, 0.050), I_3 = (0.50, 0.075), I_4 = (0.75, 1)$$

con lo cual se tiene

$$z_i = 0.025, i=1, 2, 3, 4. \quad f_1=5, f_2=4, f_3=6, f_4=5$$

Considerando que

$$U = \sum_{i=1}^k \frac{(f_i - nz_i)^2}{nz_i}$$

sustituyendo:

$$U = \left[\frac{5 - 20 \times 0.025}{20 \times 0.025} \right]^2 + \left[\frac{4 - 20 \times 0.025}{20 \times 0.025} \right]^2 +$$

$$+ \left[\frac{6 - 20 \times 0.025}{20 \times 0.025} \right]^2 + \left[\frac{5 - 20 \times 0.025}{20 \times 0.025} \right]^2 = 0.08$$

Como $\chi^2_{3,0.05} = 7.815$, y es mayor que $U = 0.08$, se concluye que la sucesión de números dada proviene de una variable aleatoria con distribución uniforme en el intervalo $(0,1)$.

2.6.5 Pruebas de Corridas ¹

Las pruebas analizadas con anterioridad pueden suponer que una secuencia de números se ha generado a partir de una población uniforme y, no obstante, el orden de los números dentro de la secuencia puede ser de índole tal que haya dudas sobre la "aleatoriedad" de los números. Por ejemplo, tomemos en consideración la secuencia que sigue de 40 números de un solo dígito: 0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9, 0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9.

Ya sea la prueba ji cuadrada o la de Kolmogorov-Smirnov, comprueban la uniformidad de la distribución de esos números, pero no se les puede dar el nombre de números aleatorios. No obstante si reordenamos esos mismos números como sigue: 2,4,7,8,4,2,4,7,3,6,3,6,7,2,9,7,1,9,9,8,5,0,3,5,9,0,3,5,9,0,3,8,5,2,6,6,0,1,1,4,5,8,1,0,6, tendremos menos razones para dudar de su aleatoriedad. De esta manera nos interesa el orden particular de los números dentro de la secuencia, para determinar su aleatoriedad. La finalidad de las pruebas de corrida es evaluar el carácter de aleatoriedad de la secuencia de números.

¹ Esta prueba también es conocida como test de las rachas, véase Leandro Pardo "simulación aplicaciones practicas en la empresa" pag. 43

Tenemos que dividir los casos posibles en un conjunto de tres resultados o eventos mutuamente exclusivos. El primero es nunca un evento es absoluto. Por ejemplo, supondrás que lanzamos al aire una moneda 10 veces, y obtenemos el siguiente resultado: cruz,cara,cruz,cruz,cara,cruz,cara,cara,cara,cruz.

La primera cara va precedida por "no hay evento" y la última cara va sucedida por un "no hay evento". Por lo tanto, podemos decir que cada secuencia de eventos comienza y termina sin evento. Las otras dos categorías mutuamente exclusivas son caras y cruces. Así una corrida se define como una sucesión de eventos similares, precedidos y seguidos por un evento diferente. La longitud de una corrida constituye del número de eventos que ocurren en la corrida. En el ejercicio de lanzamiento de la moneda al aire tenemos siete corridas: la primera y la segunda de longitud uno, la tercera de longitud 2, la cuarta y la quinta de longitud uno, la sexta de longitud tres y la séptima de longitud uno.

Al evaluar la aleatoriedad de una secuencia de números, nos interesa el número o la longitud de las corridas que se producen. En este caso estamos interesados en dos tipos de corridas ascendentes y descendentes. En consideración tomemos la siguiente secuencia de 20 números de un solo dígito:

3,1,2,3,6,4,5,4,1,2,6,8,9,7,5,2,3,1,5,1.

Primeramente, le damos un signo a cada número de la secuencia (+,-). Si a un número le sigue otro mayor, se le asigna +. Si el número siguiente es menor, se le da un signo "-". Puesto que el último número va seguido por un "evento nulo", no se le da + ni -. Por lo tanto, tenemos: -++++-
++++---+--.

Cada sucesión de "+" y "-" constituye una corrida, cuya longitud la determina el número de signos iguales que contiene. Por tanto, la primera corrida es descendente de longitud 1, la segunda es ascendente de longitud tres y así sucesivamente.

Una secuencia de números puede ser no aleatoria si se tienen demasiadas o muy pocas corridas. Si tenemos una secuencia de N números, el número máximo de corridas posibles es N-1 y el número mínimo posible es siempre de uno. Sea a_1 el número total de corridas en una secuencia. La media y la varianza de a_1 , m_{a_1} y $\sigma^2_{a_1}$, están dados por:

$$m_{a_1} = \frac{2N - 1}{3}$$

$$\sigma^2_{a_1} = \frac{16N - 29}{90}$$

Como ya se señaló, se puede rechazar la hipótesis de que una secuencia de números es aleatoria, por un número excesivo o demasiado bajo de corridas. Por ende, se requiere de una prueba de dos líneas de espera para determinar si se ha presentado alguno de esos extremos. La prueba estadística que utilizaremos:

$$Z = \frac{a_1 - m_{a1}}{\sigma_{a1}}$$

$$= \frac{a_1 - \frac{2N - 1}{3}}{\sqrt{\frac{16N - 29}{90}}}$$

Donde Z tiene una distribución normal, con una media de cero y una varianza de la unidad. Si se define el nivel de significación por medio de α y $Z_{1-\alpha/2}$ el valor de Z que hace que:

$$P(Z \geq Z_{1-\alpha/2}) = \frac{\alpha}{2}$$

entonces si

$$|Z| \geq Z_{1-\alpha/2}$$

Se rechaza la hipótesis de aleatoriedad.

Ejemplo: Tenemos la hipótesis de que los números aleatorios son aleatorios, con $\alpha = 0.05$.

La secuencia es:

59,12,19,05,58,83,18,36,00,61,47,24,41,42,98,23,67,84,43,29,
71,88,74,60,10,46,23,15,11,78,31,11,91,99,57,28,18,32,21,12,
95,38,76,07,96,33,63,10,05

La secuencia de números ascendentes y descendentes es como sigue:

```

-----
12345678911 1 11 1 1 1 11 22 2 2 22 22223333
01 2 34 5 6 7 89 01 2 3 45 67890123

```

Por consiguiente, $a_1 = 33$.

Puesto que $N = 50$, para la media y varianza a_1 , se tiene:

$$m_{a_1} = 33$$

$$\sigma^2_{a_1} = 8.57$$

$$Z = \frac{33 - 33}{\sqrt{8.57}}$$

$$Z = 0.00$$

$$Z_{0.975} = 1.96$$

Puesto que $|Z| < Z_{0.975}$, no se puede rechazar la aleatoriedad de los números sobre la base de prueba.

Esta prueba satisface algunas necesidades pero no contempla la media para lo que se tendría que diseñar más corridas en donde se pasaran por encima y por abajo de la media. Como esta tesis no intenta profundizar en este concepto, el lector que este interesado deberá consultar la obra de Schmidt "Análisis y Simulación de Sistemas industriales", pag. 248.

2.6.6 Pruebas de Autocorrelación

Las pruebas de autocorrelación examinan la tendencia de los números a ir seguidos por otros números. Para ilustrar esto observemos la siguiente serie de números aleatorios:

.20,.96,.78,.18,.92,.90,.80,.02,.53,.05,.30,.70,.59,.98,
.90,.03,.37,.86,.73,.06,.53,.25,.67,.78,.33,.97,.63,.25,.33,
.72,.91,.00,.24,.64,.90,.08,.33,.94,.33,.16,.45,.70,.18,.07

Estos números parecen aleatorios y pasar todas las pruebas presentadas hasta ahora, pero, existe una relación clara entre cada sexto número, a partir del segundo. Cada uno de esos números varía en magnitud sucesivamente, de muy grande a muy pequeño. Aunque sería difícil rechazar el generador de números aleatorios que produjo esos números por esta razón, sobre la base de un análisis de sólo 44 número.

Supondrás que deseamos determinar si hay alguna relación entre los números aleatorios $r_i, r_{i+m}, r_{i+2m}, \dots, r_{i+(m+1)m}$, o sea, la amplitud de la autocorrelación entre cada m ésimo

número aleatorio, partiendo del i ésimo. En la suposición de que los números aleatorios tienen una distribución uniforme e independiente sobre el intervalo $(0,1)$, tenemos para la función de densidad de probabilidad de r_{i+k_m}

$$f_{k_i+k_m}(r_{i+k_m})=1 \quad 0 < r_{i+k_m} < 1$$

La prueba se basa en un análisis de pares sucesivos de números aleatorios. Si consideramos el par de números r_{i+k_m} y $r_{i+(k+1)_m}$, deduciremos la distribución del producto de esas dos variables aleatorias. Suponiendo que la distribución sea uniforme e independiente, tenemos:

$$f_{x,y}(x,y)=1, \quad 0 < x < 1, \quad 0 < y < 1.$$

En donde $r_{i+k_m} = X$ y $r_{i+(k+1)_m} = Y$. sea $Z = XY$ y $W=X$,
 $X=W$ Y $Y=Z/W$.

La jacobiana de la transformación está dada por:

$$J = \begin{vmatrix} 1 & 0 \\ -\frac{Z}{W^2} & \frac{1}{W} \end{vmatrix}$$

$$f_{Z,W}(Z,W) = \frac{1}{W}, \quad Z < W < 1, \quad 0 < Z < 1$$

Y

$$f_Z(z) = \int_x^1 \frac{dw}{w}$$

$$= -\ln(z), \quad 0 < z < 1$$

La media y la varianza del producto de dos números aleatorios se determinan por medio de:

$$E(r_{i+km}r_{i+(k+1)m}) = \frac{1}{4}$$

$$\text{Var}(r_{i+km}r_{i+(k+1)m}) = \frac{7}{144}$$

Para analizar la correlación general para todos los pares sucesivos de números aleatorios, utilizaremos la estadística:

$$P_{im} = \frac{1}{M+1} \sum_{k=0}^m [r_{i+km}r_{i+(k+1)m}]$$

En donde N es el número total de números aleatorios en toda la secuencia, y M es el entero mayor tal que $i+(M+1)m < N$.

Ejemplo: Determinese si el 2o., el 7o., el 12o. y el 22o. de los números aleatorios de la secuencia que sigue están autocorrelacionados. Sea $\alpha = 0.10$.

.13, .91, .11, .02, .65, .33, .86, .63, .05, .25, .28, .80, .82, .10, .78, .88, .76, .29, .20, .66, .17, .71, .45, .40, .35

Puesto que nos interesa el grado de autocorrelación de cada quinto número a partir del segundo, $i = 2$, $m = 5$, $N = 25$ y $M = 3$:

$$P_{25} = \frac{1}{4} \sum_{k=0}^3 r_{2+5k} r_{2+5(k+1)}$$

$$= \frac{1}{4} [(0.91)(0.86) + (0.86)(0.80) + (0.80)(0.76) + (0.76)(0.71)]$$

$$= 0.6546$$

$$\sigma_{P_{25}} = \frac{\sqrt{13(3)+7}}{12(4)}$$

$$= 0.141$$

$$Z = \frac{26. - 0.25}{0.141}$$

$$Z = 2.87$$

$$Z_{0.95} = 1.65$$

Puesto que $|Z| > Z_{0.95}$, debemos rechazar la hipótesis de que los números analizados no están autocorrelacionados significativamente.

En las siguientes pruebas solamente se mencionará cual es su función, pero no se explicará su funcionamiento para no desviar el objetivo principal.

Prueba de huecos. Esta prueba se utiliza para determinar el significado de los intervalos entre la repetición de cierto dígito. Si el dígito k va seguido por x dígitos distintos de k , antes de que vuelva a aparecer k , se dice que existe un hueco de tamaño x .

Prueba de Yule. Esta prueba determina si el número de veces que se produce la suma de cada dígito compuesto de los números aleatorios es significativamente el que debería de tener.

Prueba del Póquer. Esta prueba se utiliza para analizar la frecuencia con la que se repiten los dígitos en número aleatorios de cinco dígitos. O sea:

1. Los cinco dígitos son diferentes.
2. Hay exactamente un par.
3. Dos pares diferentes.
4. Tres dígitos iguales (Tercia).
5. Tres dígitos iguales más un par (full).
6. Cuatro dígitos iguales (Póquer).
7. Cinco dígitos iguales (Quintilla).

2.6.7 NIVEL COMPUESTO DE SIGNIFICACION

Si tuviéramos que aplicar n pruebas diferentes a una serie de números y el nivel de significación para cada prueba tuviera que ser α ; entonces, el nivel compuesto de significación, α_T , está dado por:

$$\alpha_T = 1 - (1 - \alpha)^{1/n}$$

O sea, si la hipótesis nula para cada prueba es realmente verdadera, la probabilidad de que se rechace una o más de esas hipótesis está dada por α_T . Por lo tanto, para un α_T dado, el error α para cada una de las pruebas individuales está dada por:

$$\alpha = 1 - (1 - \alpha_T)^{1/n}$$

Por ejemplo, si se tienen que aplicar cinco pruebas diferentes a una serie de números aleatorios de modo que la probabilidad de rechazo incorrecto en una a más de las pruebas sea de 0.05, el error α para cada prueba será

$$\alpha = 1 - (0.95)^{0.2} \approx 0.01$$

III. GENERACION DE PROCESOS

Aquí se estudiará el comportamiento que tienen los valores observados dentro del sistema.

3.1 METODO MONTECARLO

El Método Monte Carlo es fundamental para el concepto de los sistemas de simulación que contienen elementos estocásticos o probabilísticos. Su origen y nombre se debe al trabajo de Von Neumann y Ulan a finales de los años cuarenta, cuando acuñaron el término y aplicaron la técnica para resolver ciertos problemas de protección nuclear. La técnica matemática se ha conocido durante muchos años, pero fue revivida para el trabajo secreto en Los Alamos y se le asignó el nombre clave de Monte Carlo.

En la Técnica de Monte Carlo, la experiencia o datos artificiales se generan mediante el uso de algún generador de números aleatorios y de la distribución de probabilidad acumulada de interés. La distribución de probabilidad por muestrear puede basarse en datos empíricos que se obtienen de registros anteriores, puede ser el resultado de un experimento reciente, o puede ser una distribución teórica conocida. Los números aleatorios se usan para producir una secuencia aleatorizada de valores que duplicarán la experiencia esperada, la cual puede producirse mediante la distribución descrita por alguna función de probabilidad:

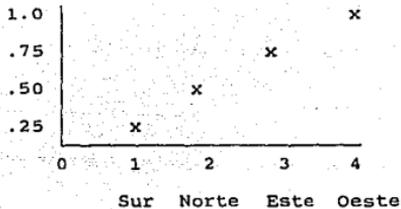
1. Grafique o tabule los datos de interés como una función de distribución de probabilidad acumulada, con los valores de la variable sobre el eje "x" o abscisa y las probabilidades de 0 a 1 que se trazaron sobre el eje "y" de la ordenada.
2. Seleccione un número decimal aleatorio (NA) entre 0 y 1 por medio de un generador de números aleatorios.
3. Proyecte horizontalmente el punto sobre el eje "y" (ordenada) que corresponda a este número decimal aleatorio, hasta que la línea de proyección interseque la curva acumulativa.
4. Proyecte hacia abajo de este punto de intersección sobre la curva al eje x (abscisa).
5. Escriba el valor de x correspondiente a este punto de intersección. Después, este valor de x se toma como el valor de la muestra.
6. Repita los pasos del 2 al 6 hasta que se hayan generado tantas variables aleatorias como se deseen, siguiendo la secuencia en la cual se obtuvieron.

Como ejercicio para exponer estos puntos supongase el ejemplo "el ebrio aleatorio", en el que este personaje tiene que llegar a su casa después de salir de la cantina, el tiene que caminar 10 cuadras en forma aleatoria cual sería la probabilidad de que llegara a su casa.

Como primer paso se acordó que hay que graficar los datos en los ejes cartesianos, en donde tenemos en el eje x

los puntos cardinales y en el eje "y" la probabilidad de que camine hacia ese lugar, y como son 4 puntos, se tiene una probabilidad del 25% en cada punto, la siguiente tabla nos muestra esto:

| Puntos Cardinales | Probabilidad | Probabilidad Acumulada |
|-------------------|--------------|------------------------|
| Sur | 0.25 | 0.25 |
| Norte | 0.25 | 0.50 |
| Este | 0.25 | 0.75 |
| Oeste | 0.25 | 1.00 |



- Como segundo paso tomamos un número aleatorio entre 0 y 1, supongamos 0.456.
- Tercer paso se proyecta horizontalmente el número aleatorio y le corresponde el valor acumulado de .50.
- Cuarto paso se proyecta al eje de las abscisas nos da un valor de 2.
- Quinto paso el valor de x, o sea 2, nos representa que el ebrio camino hacia el Norte.

- El último paso es repetir los pasos del 2 hasta el 5, vamos a realizarlo para las 10 cuadras.

| Número de cuadra | Número aleatorio | Punto Cardinal |
|------------------|------------------|----------------|
| 1 | 0.456 | Norte |
| 2 | 0.118 | Sur |
| 3 | 0.728 | Este |
| 4 | 0.754 | Oeste |
| 5 | 0.346 | Norte |
| 6 | 0.964 | Oeste |
| 7 | 0.178 | Sur |
| 8 | 0.105 | Sur |
| 9 | 0.433 | Norte |
| 10 | 0.961 | Oeste |

Con esto tenemos que el ebrio recorrió tres cuadras al norte, tres al sur, dos al este y uno al oeste; por lo tanto al avanzar 3 al norte menos 3 al sur quedo en el mismo lugar, dos al este y uno al oeste; quedo al finalizar las diez cuadras caminadas en la primer cuadra al este. Para determinar que probabilidad tiene de llegar a su casa hay que determinar a cuantas cuadras esta su casa de la cantina si queda a un número impar de cuadras tiene una probabilidad "0" de llegar, ya que con un numero par de cuadras caminadas, jamas llegara a un numero impar. Si se desea ahondar en esta simulación consultar la obra de Shanon, 1980 p. 83.

IV. SIMULACION DE SISTEMAS DE LINEAS DE ESPERA

4.1 COMPORTAMIENTO DE LAS LINEAS DE ESPERA

En estos sistemas las unidades llegan en busca de servicio, en donde si está desocupado será atendido de inmediato de lo contrario tendrá que esperar su turno si es que se están atendiendo otras unidades en el sistema.

Los fenómenos básicos que se necesitan para diseñar un modelo del fenómeno de espera son:

1. Forma de como los clientes llegan al punto de servicio. Su descripción podrá ser determinística o aleatoria, en donde la forma determinística es a un intervalo constante de tiempo, la forma aleatoria esta determinada por el azar.
2. Forma como se realiza el servicio. Su descripción, al igual que en el caso anterior, es determinística ó aleatoria.
3. Forma de elegir los clientes de la fila de espera en el servicio. Esto podrá ser por la norma FIFO, primero que entra primero que sirve (First Input Last Input) o la norma LIFO, último que entra primero que se sirve (First Input Last Output), o dar prioridad a determinados clientes.

Las características principales cuando se simule un fenómeno de espera son:

- La longitud de la cola en diversos tiempos.
- El tiempo que el cliente pasa esperando en el sistema, esto representa el tiempo que se pasa haciendo cola más el tiempo que se pasa recibiendo el servicio.
- El tiempo que el sistema está inactivo.

4.1.1 SISTEMA DE CANAL SIMPLE

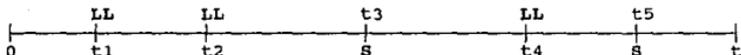
De forma esquemática, cuando únicamente hay un servicio, el esquema es el siguiente:



El sistema tiene alteraciones únicamente cuando un cliente llega al sistema o cuando un cliente sale del sistema una vez que ha recibido el servicio. Al llegar un cliente puede ocurrir que no hay línea de espera (cola) y pase directamente al servicio, pero sí el servicio está ocupado, tendrá que esperar a que el cliente anterior termine de ser servido pasando a la cola, y finalmente, hay una línea de espera, con lo cual tendrá que aguardar a que

se sirvan a todos los clientes que están en la línea de espera.

Por otra parte, una vez que se ha servido a un cliente, bien pasa a ser servido el siguiente de la fila de espera, bien el sistema se queda desocupado. Esto pone de manifiesto que el sistema quedará determinado especificando las llegadas y salidas del sistema. En el siguiente esquema se representa por LL las llegadas y por S las salidas en un intervalo $(0, t)$, con lo cual se tiene que hasta el instante t_4 se han producido 3 llegadas al sistema, en los instantes t_1 , t_2 y t_4 y una salida del sistema en el instante t_3 , por lo que en el instante t_4 hay dos clientes en el sistema.



La forma de diseñar el fenómeno de espera sería el siguiente: se comienza simulando un valor de t_1 que representa la primera llegada al sistema de un cliente, por eso se representa por LL. Se supone que la instalación no está ocupada inicialmente, y el cliente pasa directamente a recibir servicio. Seguidamente, se simulan dos valores t_2 y t_3 , el t_2 representa la llegada de un nuevo cliente al sistema

Proceso de llegada. Cada llegada se llama "ocupación". Dado que se conoce con certeza el tiempo entre dos llegadas,

necesitamos especificar para él una distribución de probabilidad. En el modelo básico se usa una distribución particular llamada distribución exponencial, y queda especificada mediante un parámetro, que es L , es el tiempo promedio de llegada, o sea cuántas ocupaciones en promedio se presentan durante un periodo de tiempo. Ej: un parámetro

$$L = 0.05 \text{ ocupaciones por minuto}$$

Esto significa un promedio de cinco centésimos de ocupación llegan cada minutos. Para la distribución exponencial,

$$\text{tiempo entre llegadas} = \text{tiempo medio de llegada} = 1/L$$

Proceso de servicio. En el modelo básico, el tiempo que dura la conclusión de una ocupación, o sea el tiempo de servicio, se maneja también mediante la distribución exponencial, el parámetro es μ . Representa la tasa media de servicio en ocupaciones por minuto. En otras palabras, μT es el número de ocupaciones que se atenderán durante un periodo de T minutos si la máquina se encuentra ocupada durante ese tiempo.

Magnitud de la línea de espera. No hay límite para el número de ocupaciones que pueden esperar en la línea. Se dice que la línea de espera es infinita.

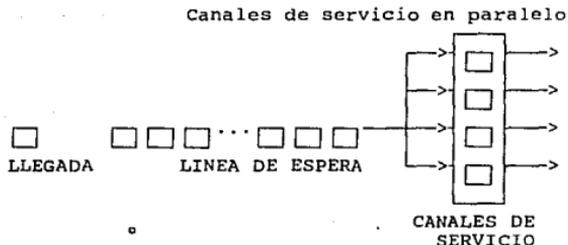
Comportamiento. Las ocupaciones son atendidas en el orden en que se formen en la línea.

Características del modelo de canal simple

| CARACTERISTICA | SIMBOLO | FORMULA |
|--|-----------------|----------------------------|
| Número esperado en el sistema | LL | $\frac{L}{\mu - L}$ |
| Número previsto en la línea de espera | LL _q | $\frac{L^2}{\mu(\mu - L)}$ |
| Tiempo previsto de espera (incluye el tiempo de servicio) | W | $\frac{1}{\mu - L}$ |
| Tiempo previsto de espera en la línea de espera | W _q | $\frac{L}{\mu(\mu - L)}$ |
| Probabilidad de que el sistema esté vacío | P ₀ | $\frac{1 - L}{\mu}$ |

4.1.2 CANALES MULTIPLES EN PARALELO

En este sistema a medida que llegan las unidades, van tomando su lugar en la línea de espera, aquí hay más de un canal disponible para el servicio. Este sistema se observa en la sig. figura:



La primera unidad de la línea de espera entra al primer canal de servicio que queda disponible.

Como ejemplos de estos sistemas tenemos a varios bancos de México, entre ellos esta **BANCOMER**, este banco incluso tiene integrado un sistema de simulación que despliega en pantalla el tiempo aproximado que se tardaría en esperar el cliente para realizar su transacción, en donde se ven regularmente los tiempos que esperan y en los que se realiza su transacción, lo cual es excelente para la toma de decisiones, para aumentar o disminuir: las cajas en ciertas transacciones, el tiempo de servicio (en algunos centros trabajan hasta las 5 p.m. y los sábados medio día), sus sucursales, sus cajeros automáticos; promocionar sus servicios, y en fin para la planeación de sus estrategias.

4.1.3 CANALES MULTIPLES EN SERIE

En este sistema los canales de servicio y sus colas asociadas se ordenan en serie. Para que una unidad que llega reciba el servicio completo, debe pasar por todos y cada uno de los canales de servicio. Este sistema se observa en la siguiente figura:

Canales de servicio en serie

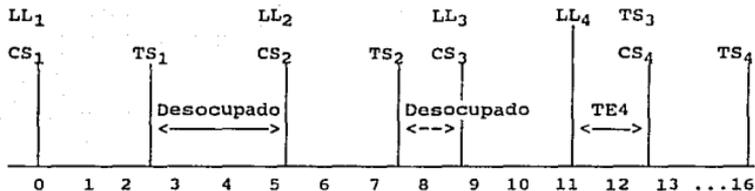


Para quienes quieran abundar en el tema ver Schmidh 1979.

4.2 CONSTRUCCION DEL MODELO DE LINEA DE ESPERA

Para referenciar nuestro modelo de línea de espera tomaremos la siguiente situación: el gerente de una oficina bancaria está preocupado por las quejas de sus clientes con respecto al tiempo que deben esperar para realizar sus transacciones. Un consultor le ha presentado al banco una propuesta, con la cual espera acelerar los servicios. Propuesta en la que se analizará, por unos honorarios de \$5.000, el contenido del trabajo de un cajero y se le reentrenará. El consultor se compromete a reducir el tiempo de prestación de servicio por cliente por lo menos en un 10%. El gerente de la oficina está convencido de que este gasto no se justificaría si el tiempo promedio de espera del cliente no se redujera siquiera en un 15% y asignará a un analista la responsabilidad de determinar si vale la pena o no contratar al consultor.

Los modelos empiezan a operar en un punto de referencia llamado tiempo cero, es en este caso el momento de la llegada de la primera transacción, o en este caso del primer cliente imaginario. El simulador registra a ese cliente debido a que no han atendido más, y el servicio se le dá de manera inmediata. Cuando es atendido el primer cliente, y es probable que llegue un segundo cliente antes de que salga el primero, entonces el segundo deberá esperar hasta que se termine el servicio del primero. La siguiente figura 4.2.1 muestra una simulación de este proceso:



LL_i Momento de llegada del cliente i
 CS_i Momento en el cual se comienza el servicio al cliente i
 TS_i Momento en el cual se termina el servicio al cliente i
 TE_i Tiempo de espera antes de que el servicio comience para el cliente i

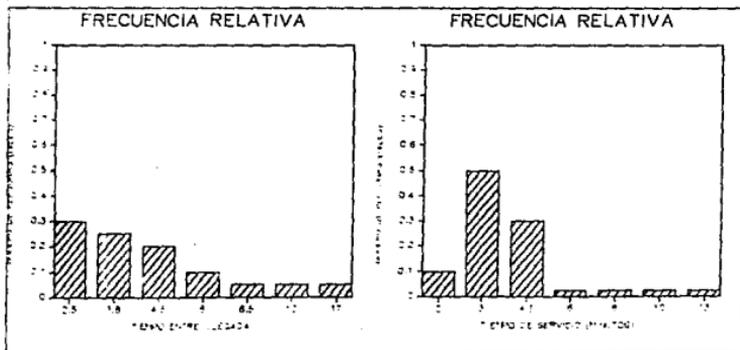
Figura 4.2.1 Flujo de la simulación de los eventos en escala de tiempo

En la figura podemos observar que los tres primeros clientes no deben esperar por que no hay clientes en espera es decir "la cola esta vacía", pero en el momento que llega del cliente número cuatro están atendiendo al cliente anterior y tiene un tiempo de espera para poder ser atendido.

4.3 RECOLECCION DE DATOS

El analista debe registrar el tiempo transcurrido entre las llegadas sucesivas de los clientes, la longitud del tiempo requerido para servir a cada cliente y los tiempos que los clientes deben esperar. Para esto se debe de considerar un tamaño de muestra lo suficientemente adecuado para reflejar la realidad en el modelo. En el ejemplo que estamos siguiendo será de 1000 eventos, es decir se registrarán los datos de 1000 clientes del tiempo transcurrido entre la llegada de los clientes, la longitud del tiempo requerido para servir al cliente y los tiempos que deben esperar.

Esta gráfica nos muestra la llegada de los clientes con una escala de distribución de 0 a 1 y se identifican de esta manera a los clientes, del lado izquierdo tenemos las llegadas tomadas del mundo real, y del lado derecho el

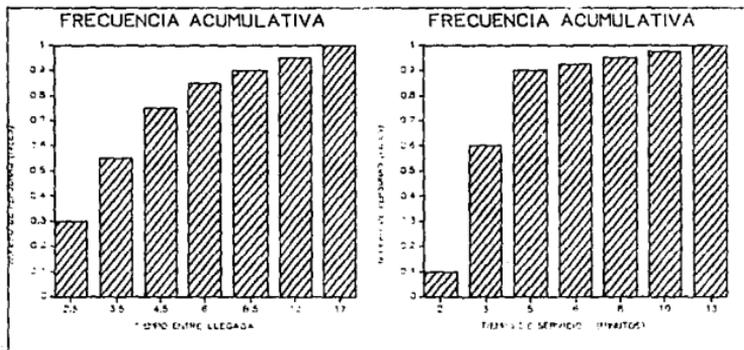


tiempo de servicio que tardan los clientes en ser atendidos.

En la gráfica inferior tenemos la misma información que en las anteriores pero en éstas de una manera distribuidas acumulativamente.

La simulación de estos resultados para ver si es consistente con la realidad se observa en el listado de la página 57.

Este primer listado de la pagina siguiente se efectuó con el paquete micromanager, el cual es muy fácil de utilizar pero su principal desventaja son sus restricciones, las cuales son que únicamente se puede simular 1000 corridas, y contiene un modelo predeterminado el cual no se puede alterar para aumentar variables en la simulación, o incorporarle una distribución teórica.



4.4 PROGRAMA DE SIMULACION

- PAGE 1 -

***** INPUT DATA ENTERED *****

Queuing simulation

Arrival Distribution

| Arrival Interval | Probability |
|------------------|-------------|
| 2.50 | 0.30 |
| 3.50 | 0.25 |
| 4.50 | 0.20 |
| 6.00 | 0.10 |
| 8.50 | 0.05 |
| 12.00 | 0.05 |
| 17.00 | 0.05 |

Service Time Distribution

PROGRAM: Simulation

PAGE 2 -

| Service Time | Probability |
|--------------|-------------|
| 2.00 | 0.10 |
| 3.00 | 0.50 |
| 4.50 | 0.30 |
| 6.00 | 0.03 |
| 8.00 | 0.03 |
| 10.00 | 0.03 |
| 13.00 | 0.03 |

Number of simulation runs : 1000

***** PROGRAM OUTPUT *****

*** Simulation Results for 1000 Arrivals ***

| Arvl | r1 | Arvl Intvl | Arvl Time | Enter Facil'y Time | Wait'g Time | Length of Queue at Entry | r2 | Service Time | Dpture Time | Time in System |
|------|------|---------------|--------------|--------------------------|----------------|--------------------------------|--------|-----------------|----------------|-------------------|
| 1 | 0.00 | 0.0 | 0.0 | 0.0 | 0.0 | 0 | 0.43 | 3.0 | 3.0 | 3.0 |
| 2 | 0.58 | 4.5 | 4.5 | 4.5 | 0.0 | 0 | 0.72 | 4.5 | 7.5 | 4.5 |
| 3 | 0.76 | 6.0 | 10.5 | 10.5 | 0.0 | 0 | 0.76 | 4.5 | 12.0 | 4.5 |
| 4 | 0.04 | 2.5 | 13.0 | 13.0 | 0.0 | 0 | 0.28 | 3.0 | 15.0 | 3.0 |
| 5 | 0.54 | 3.5 | 16.5 | 16.5 | 0.0 | 0 | 0.23 | 3.0 | 18.0 | 3.0 |
| 6 | 0.40 | 3.5 | 20.0 | 20.0 | 0.0 | 0 | 0.12 | 3.0 | 21.0 | 3.0 |
| 7 | 0.49 | 3.5 | 23.5 | 23.5 | 0.0 | 0 | 0.26 | 3.0 | 24.0 | 3.0 |
| 8 | 0.00 | 2.5 | 26.0 | 26.0 | 0.0 | 0 | 0.56 | 3.0 | 27.0 | 3.0 |
| 9 | 0.67 | 4.5 | 30.5 | 30.5 | 0.0 | 0 | 0.43 | 3.0 | 30.0 | 3.0 |
| 10 | 0.61 | 4.5 | 35.0 | 35.0 | 0.0 | 0 | 0.60 | 4.5 | 34.5 | 4.5 |
| 11 | 0.17 | 2.5 | 37.5 | 37.5 | 0.0 | 0 | 0.29 | 3.0 | 37.5 | 3.0 |
| 12 | 0.16 | 2.5 | 40.0 | 40.0 | 0.0 | 0 | 0.79 | 4.5 | 42.0 | 4.5 |
| 13 | 0.42 | 3.5 | 43.5 | 43.5 | 0.0 | 0 | 0.37 | 3.0 | 45.0 | 3.0 |
| 14 | 0.66 | 4.5 | 48.0 | 48.0 | 0.0 | 0 | 0.35 | 3.0 | 48.0 | 3.0 |
| 15 | 0.79 | 6.0 | 54.0 | 54.0 | 0.0 | 0 | 0.39 | 3.0 | 51.0 | 3.0 |
| 16 | 0.46 | 3.5 | 57.5 | 57.5 | 0.0 | 0 | 0.60 | 3.0 | 54.0 | 3.0 |
| 17 | 0.38 | 3.5 | 61.0 | 61.0 | 0.0 | 0 | 0.86 | 4.5 | 58.5 | 4.5 |
| 18 | 0.63 | 4.5 | 65.5 | 65.5 | 0.0 | 0 | 0.59 | 3.0 | 61.5 | 3.0 |
| 19 | 0.12 | 2.5 | 68.0 | 68.0 | 0.0 | 0 | 0.77 | 4.5 | 66.0 | 4.5 |
| 20 | 0.74 | 4.5 | 72.5 | 72.5 | 0.0 | 0 | 0.97 | 10.0 | 76.0 | 10.0 |
| 21 | 0.11 | 2.5 | 75.0 | 76.0 | 1.0 | 0 | 0.66 | 4.5 | 80.5 | 5.5 |
| 22 | 0.75 | 4.5 | 79.5 | 80.5 | 1.0 | 0 | 0.15 | 3.0 | 83.5 | 4.0 |
| 23 | 0.17 | 2.5 | 82.0 | 83.5 | 1.5 | 0 | 0.86 | 4.5 | 88.0 | 6.0 |
| 24 | 0.24 | 2.5 | 84.5 | 88.0 | 3.5 | 0 | 0.70 | 4.5 | 92.5 | 8.0 |
| 25 | 0.71 | 4.5 | 89.0 | 92.5 | 3.5 | 0 | 0.99 | 13.0 | 105.5 | 16.5 |
| 26 | 0.88 | 8.5 | 97.5 | 105.5 | 8.0 | 0 | 0.94 | 8.0 | 113.5 | 16.0 |
| 27 | 0.75 | 4.5 | 102.0 | 113.5 | 11.5 | 1 | 0.77 | 4.5 | 118.0 | 16.0 |
| 28 | 0.09 | 2.5 | 104.5 | 118.0 | 13.5 | 2 | 0.91 | 6.0 | 124.0 | 19.5 |
| 29 | 0.29 | 2.5 | 107.0 | 124.0 | 17.0 | 2 | 0.00 | 2.0 | 126.0 | 19.0 |
| 30 | 0.03 | 2.5 | 109.5 | 126.0 | 16.5 | 3 | 0.45 | 3.0 | 129.0 | 19.5 |
| | | | | | 102.5 | 12 | 4097.0 | | | |

Average waiting time : 0.1
 Average length of the queue at entry: 0.0
 Average total time in the system : 4.1

Este otro listado es con el paquete GPSS/H que permite varias alternativas al usuario pero se requiere conocer las instrucciones para poder manejarlo.

STUDENT GPSS/H RELEASE 2.0 (AY130) 10 Jan 1992 17:50:30 FILE: RE1.gps 1/4

| LINE# | SIM# | IF DO | BLOCK# | *LOC | OPERATION | A,B,C,D,E,F,G | COMMENTS |
|-------|------|-------|--------|------|--------------------------------------|---------------|--|
| 1 | 1 | | | | SIMULATE | | Case Study |
| 2 | 2 | | | | | | Base Time Unit: 1 Minute |
| 3 | 3 | | | | | | |
| 4 | 4 | | | | | | Control Statements (FUNCTION; STORAGE) |
| 5 | 5 | | | | | | |
| 6 | 6 | | | | | | * |
| 7 | 7 | | | | LLEGADA FUNCTION | RN1,07 | job |
| 8 | 8 | | | | .3,2.5/.55,3.5/.75,4.5/.85,6/.90,8.5 | | |
| 9 | 9 | | | | .95,12/1,17 | | |
| 10 | 10 | | | | | | * |
| 11 | 11 | | | | SERVICIO FUNCTION | RN1,07 | job interarrival-time |
| 12 | 12 | | | | .1,2/.6,3/.9,4.5/.925,6/.950,8 | | |
| 13 | 13 | | | | .975,10/1,13 | | |
| 14 | 14 | | | | | | |
| 15 | 15 | | | | | | * |
| 16 | 16 | | 1 | | GENERATE | FN(LLEGADA),0 | objects arrive, one |
| 17 | 17 | | | | | | and are placed on conveyor |
| 18 | 18 | | 2 | | QUEUE | MOVING | start MOVING Queue membership |
| 19 | 19 | | 3 | | SEIZE | BANCO | |
| 20 | 20 | | 4 | | ADVANCE | FN(SERVICIO) | move to the take-off point |
| 21 | 21 | | 5 | | RELEASE | BANCO | |
| 22 | 22 | | 6 | | DEPART | MOVING | end MOVING Queue member |
| 23 | 23 | | | | | | * |
| 24 | 24 | | 7 | | TERMINATE | 1 | leave the system |
| 25 | 25 | | | | | | * |
| 26 | 26 | | | | START | 2000 | convey 500 objects intot |
| 27 | 27 | | | | | | * |
| 28 | 28 | | | | END | | end of Model-File exec. |

STUDENT GPSS/H RELEASE 2.0 (AY130) 10 Jan 1992 17:50:30 FILE: RET.gps 2/4

ENTITY DICTIONARY (IN ASCENDING ORDER BY ENTITY NUMBER; "*" => VALUE CONFLICT.)

Facilities: 1=BANCO

Queues: 1=MOVING

Functions: 1=LLEGADA 2=SERVICIO

Random Numbers: 1

| SYMBOL NUMBER | VALUE | EQU DEFNS | CONTEXT | REFERENCES BY STATEMENT |
|---------------|-------|-----------|------------|-------------------------|
| BANCO | 1 | | Facility | 19 21 |
| MOVING | 1 | | Queue | 18 22 |
| LLEGADA | 1 | 7 | Function | 16 |
| SERVICIO | 2 | 11 | Function | 20 |
| 1 | 1 | | Random Nbr | 7 11 |

STORAGE REQUIREMENTS (BYTES)

| | |
|----------------|-------|
| COMPILED CODE: | 438 |
| COMPILED DATA: | 264 |
| MISCELLANEOUS: | 0 |
| ENTITIES: | 368 |
| COMMON: | 10000 |
| ----- | |
| TOTAL: | 11070 |

STUDENT GPSS/H RELEASE 2.0 (AY130) 10 Jan 1992 17:50:30
 FILE: RET.gps 3/4

Simulation begins.

RELATIVE CLOCK: 10194.0000 ABSOLUTE CLOCK: 10194.0000

| BLOCK | CURRENT | TOTAL |
|-------|---------|-------|
| 1 | | 2000 |
| 2 | | 2000 |
| 3 | | 2000 |
| 4 | | 2000 |
| 5 | | 2000 |
| 6 | | 2000 |
| 7 | | 2000 |

--AVG-UTIL-DURING--

| FACILITY | TOTAL TIME | AVAIL TIME | UNAVL TIME | ENTRIES | AVERAGE TIME/XACT | CURRENT STATUS | PERCENT AVAIL | SEIZING XACT | PREEMPTING XACT |
|----------|---------------|---------------|---------------|---------|----------------------|-------------------|------------------|-----------------|--------------------|
| BARCO | 0.790 | | | 2000 | 4.028 | AVAIL | | | |

| QUEUE | MAXIMUM CONTENTS | AVERAGE CONTENTS | TOTAL ENTRIES | ZERO ENTRIES | PERCENT ZEROS | AVERAGE TIME/UNIT |
|--------|---------------------|---------------------|------------------|-----------------|------------------|----------------------|
| MOVING | 9 | 1.736 | 2000 | 0 | | 8.848 |

| SAVERAGE TIME/UNIT | QTABLE NUMBER | CURRENT CONTENTS |
|-----------------------|------------------|---------------------|
| 8.848 | | 0 |

| RANDOM SQUARE STREAM UNIFORMITY | ANTITHETIC VARIATES | INITIAL POSITION | CURRENT POSITION | SAMPLE COUNT | CHI- |
|--|------------------------|---------------------|---------------------|-----------------|------|
| 1 | OFF | 100000 | 104001 | 4001 | 0.85 |

STATUS OF COMMON STORAGE

9744 BYTES AVAILABLE
 256 IN USE
 1408 USED (MAX)

STUDENT GPSS/H RELEASE 2.0 (AY130) 10 Jan 1992 17:50:30
FILE: RET.gps 4/4

Simulation terminated. Absolute Clock: 10194.0000

Total Block Executions: 14000
Blocks / second: 551

Microseconds / Block: 1816.43

Elapsed Time Used (SEC)

| | |
|------------|-------|
| PASS1: | 0.61 |
| SYM/XREF | 0.05 |
| PASS2: | 0.66 |
| LOAD/CTRL: | 0.60 |
| EXECUTION: | 25.43 |
| OUTPUT: | 0.22 |
| ----- | |
| TOTAL: | 27.57 |

V. SIMULACION DE SISTEMAS DE INVENTARIOS

En este apartado se aborda la aplicación de los métodos de simulación a los modelos de inventarios (STOCKS). En general, se admite que existe un problema de inventario cuando es necesario tener almacenado un conjunto de recursos útiles, en algún periodo de tiempo, con el propósito de satisfacer unas determinadas necesidades. Las empresas almacenan diversas existencias, tales como materias primas, piezas esperando un montaje final, material de oficina, así como artículos elaborados y preparados para la venta. Estas necesidades de almacenar conduce a tener inmovilizado un gran capital. Esta acumulación desmesurada de existencias fue lo que motivó el inicio, después de la segunda guerra mundial, de lo que hoy en día constituye la moderna Teoría de Inventarios. Lógicamente los factores a controlar (según Pardo, 1987) son,

a) Momento en el que se decide realizar un nuevo pedido (punto de pedido).

b) Cantidad a pedir en cada reabastecimiento.

y el objetivo que se pretende es hacer mínimo el costo total a lo largo de un periodo de tiempo determinado. El costo total se entiende como la suma de los siguientes costos:

- 1) Costo de compra de existencias.
- 2) Costo de pedido y recepción (formularios de pedidos; sobres, papel y sellos o llamadas telefónicas; personal).
- 3) Costo de mantenimiento de inventario (impuestos referidos al inventario, seguros del inventario; alquiler del almacén; obsolescencia y depreciación de las existencias, robo, deterioro, capital invertido, salarios del personal relacionados con el inventario).
- 4) Costo de falta de existencias (retraso en atender los pedidos o la imposibilidad de atenderlos).

5.1 MODELO DE COMPRA SIN ESCASEZ ¹

Este modelo se basa en las siguientes suposiciones:

- 1 La demanda se efectúa a tasa constante.
- 2 El remplazo es instantáneo (la tasa de remplazo es infinita).
- 3 Todos los coeficientes de costo (C_1 , C_2 , C_3) son constantes.

¹ Otros autores lo manejan como tamaño económico del lote.

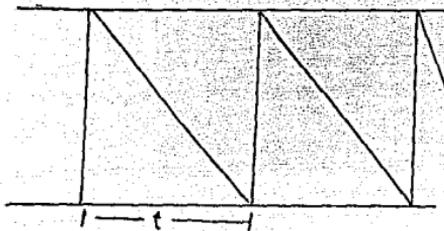


FIGURA 5-1 Modelo de compra sin escasez.

En la figura 5-1 se representan como iguales el inventario máximo I_m y la cantidad económica pedida Q . El tiempo t es el tiempo entre pedidos o el tiempo de un período. El período planeado T se toma como 1 año en la deducción de este y de los demás modelos de inventario del capítulo.

El costo del modelo anterior está formado por 3 componentes del costo:

$$\text{Costo total/año} = \text{costo unitario/año} + \text{costo de ordenar una compra/año} + \text{costo total / período} \times \text{número de período / año} \quad (5-1)$$

El costo unitario por período simplemente es costo de Q unidades por período por año:

$$\text{Costo tot./año} = \text{costo tot./período} \times \text{núm. períodos/año} \quad (5-2)$$

$$C_1Q \quad (5-3)$$

donde C_1 es el costo por unidad.

Puesto que sólo se efectúa una compra por período, el costo de ordenar la compra es el costo de hacer un pedido, y se denomina C_2 .

El inventario promedio por período es $Q/2$. Por consiguiente, el costo de mantenimiento del inventario por período es:

$$C_3tQ/2 \quad (5-4)$$

donde C_3 = costo de mantenimiento o costo de mantener una unidad en inventario durante 1 año, \$/unidad-años.

t = tiempo de un período en años.

El costo total C' por período es:

$$C' = C_1Q + C_2 + C_3tQ/2 \quad (5-5)$$

El tiempo de un período, expresado en años, es

$$t = Q/D \quad (5-6)$$

donde D = demanda de un artículo particular en unidades/año. El número N de períodos o de pedidos por año es el recíproco de la ecuación anterior:

$$N = D/Q \quad (5-7)$$

Sustituyendo la ecuación (5-6) en la ecuación (5-5) y multiplicando luego el resultado por la ecuación (5-7) se obtiene:

$$C=C_1D+C_2D/Q+C_3Q/2 \quad (6-8)$$

Cuando los componentes del costo de la ecuación (6-8) se representan gráficamente como se indica en la figura (6-2) se obtiene un punto óptimo (de costo mínimo).

Una forma de determinar Q óptimo es suponer valores de Q y sustituir en la ecuación (5-8) hasta encontrar el punto de costo mínimo. Un mejor procedimiento consiste en derivar la ecuación (5-8) con respecto a Q e igualar la derivada a cero. Puesto que el término C_1D es constante, la derivada de la ecuación (5-8) es

$$\frac{dC}{dQ} = 0 = -\frac{C_2D}{Q^2} + \frac{C_3}{2} \quad (6-9)$$

Despejando Q se obtiene

$$Q = \sqrt{\frac{2C_2D}{C_3}} \quad (5-10)$$

La ecuación de la cantidad pedida que ocasiona un costo mínimo y tiene como base un balance entre los dos costos variable (costos de almacenamiento y costos de compra) incluidos en el modelo. Cualquier otra cantidad pedida ocasiona un costo mayor.

5.2 PROBLEMA DE UN CONTROL DE INVENTARIO

1) Enunciado del Problema

En una tienda de venta al menudeo, la demanda para su artículo se distribuye normalmente con una media y una desviación estándar de 10 y 2 unidades respectivamente. No obstante el comerciante lleva un inventario manual que esta en o por debajo del punto predeterminado llamado punto de reorden; él ordena un pedido de reabastecimiento con su proveedor. El promedio de reabastecimiento es llamado la cantidad de reorden, es siempre de 100 unidades. El orden de reabastecimiento llega a la tienda de 6 a 10 días desde cualquier lugar, después de establecer la orden. Este lapso de tiempo entre establecer el punto de reabastecimiento y la llegada a la tienda, es llamado tiempo de entrega (lead time). La distribución del tiempo de entrega es mostrado en la siguiente tabla:

TABLA 5.1 La distribución del tiempo de entrega

| Teiempo de Entrega (en días) | Frecuencia relativa | Frecuencia Acumulada |
|---------------------------------|------------------------|-------------------------|
| 6 | .05 | .05 |
| 7 | .25 | .30 |
| 8 | .30 | .60 |
| 9 | .22 | .82 |
| 10 | .18 | 1.00 |

La demanda que surge cuando el negocio no tiene artículos en el inventario esta perdida, esto es, clientes cuya demanda no pueden ser satisfechas inmediatamente se van a cualquier otra parte a tramitar su compra.

El comerciante quiere conocer como en base a su experiencia los artículos variaran. Desde los registros sumariados en la figura 5.1, el conoce que el tiempo de entrega es alrededor de 8 días en promedio. Por que el promedio de demanda de los artículos es de 10 unidades por día, el razona que el punto de reorden no tiene que ser menor a 80; de otra manera, el no tendria suficientes de los artículos a la mano para satisfacer la demanda de tiempo de entrega, esto es, la demanda esperada ocurrira durante el periodo de tiempo de entrega. El razona después que por poner el punto de reorden a un nivel alto, tanto como 90 ó 100, la posibilidad que el experimentaría por perdidas de ventas mientras espera que la llegada del orden de reabastecimiento, decresera. De otra manera, el más alto punto medio de reorden, que el esta llevando más el almacenamiento en promedio, incrementan el costo que envuelve al inventario.

Construir un modelo de GPSS para la situación del comerciante. Diseñar el modelo para medir la distribución de dos variables aleatorias: "ventas perdidas por día" y "numero de unidades llevadas en el inventario". Correr el modelo para estimar esas 2 distribuciones cuando la cantidad de reorden es de 100, y el punto de reorden es de 80 ó 90 ó 100. Para cada alternativa de configuración aislada del modelo despues una simulación de 1,000 días.

Para facilitar, asumimos que el comerciante checa el nivel de inventario cada fin de día de negocios, y entonces hace o no hace lugar a una orden de reabastecimiento, también asume que la orden de reabastecimiento, siempre llega después de cerrar el día, esto quiere decir que ninguno de los reabastecimientos pueden ser usados para reunir la demanda del día en que ocurre la llegada del pedido. También ignoramos cualquier "problema de fin de semana". Cuyos problemas llegan en la realidad por que el comerciante no pudo abrir por un negocio en sábado y/o domingo una orden de reabastecimiento continua en tránsito procediendo hacia su destino. Ignorando el problema de fin de semana, es equivalente a asumir que el comerciante hace negocios 7 días en la semana.

2) Haciendo la construcción de un modelo

El modelo es compuesto de 2 segmentos separados. **La Simulación Del Segmento De Demanda**, la demanda diaria por artículo, arreglada tanto de que la demanda sea introducida, como sea posible y las entradas de los valores de ventas perdidas y el almacenamiento manual dentro de las tablas de GPSS al final de cada día. **El Segmento De Mantenimiento Del Control De Inventario** una observación a la situación del almacenamiento manual, y causas para ordenar un reabastecimiento, serán puestos cuando el almacenamiento manual, esta en o abajo del punto de reorden, y ahí ya se hace la orden de reabastecimiento al proveedor. Estos dos

segmentos separados, se comunican entre sí a través de un simple savevalue (valor salvado) STOCK, cuyo valor es el actual inventario manual.

En el Segmento de Demanda, más bien que permitiendo que ocurra la demanda unidad por unidad como sucede en los días, la demanda total del día es determinada por una simple transacción cuyas entradas al modelo de cada día y muestra desde la distribución de la demanda. Esta demanda total, llevada en la transacción del parámetro 1, es entonces examinadas en un modo de transferencia condicional contra el Savevalue STOCK para determinar si todas las demandas diarias pueden ser reunidas. Si se puede hacer esto, el STOCK es actualizado, las entradas son hechas en 2 tablas, y la transacción deja el modelo. Cuando la demanda diaria excede el STOCK, la transacción toma la salida no secuencial desde el bloque TEST (probar), y procede a computar ventas perdidas y poner el STOCK igual a 0. Esto hace entrada a dos tablas, y deja el modelo.

En el segmento del control del inventario, un simple oficio de un trabajador es detenido delante de un modo negativo en un bloque TEST, esperando para que el STOCK llegue o baje del punto de reorden. El punto de reorden es detenido por si mismo en savevalue ROP (punto de reorden del inglés: reorder point). Cuando el bloque TEST permite pasar indicando que una orden de reabastecimiento debe ser hecha, la transacción mueve a través de esto hasta un bloque de

ADVANCE, donde el tiempo de entrega es simulado. Moviéndose desde el bloque de ADVANCE después del tiempo de entrega que ha transcurrido, el oficio del trabajador actualiza el STOCK adicionando la cantidad de reorden, esto hace que el valor de la cantidad de reorden sea salvada en ROQ (reorder quantity). El oficio del trabajador entonces transfiere de regreso a reestablecer la observación de la situación manual del inventario.

La Transacción en el segmento del control de Inventario tiene más baja prioridad que las transacciones que se mueven a través del segmento de la demanda. Esta distinción de prioridades es hecha en honor a la condición del problema que el reestablecimiento de las ordenes llegan después de cerrar el día de negocios. Las transacciones del trabajador lleva un orden sobre el evento futuro de encadenación no posterior que al fin de los cuarenta y cinco días simulados (el mínimo día de entrega es 6 días). Con certeza el segmento de la transacción de la Demanda no estaba puesto en los eventos futuros encadenados hasta que empiece el quincuagésimo día (planeado moverse dentro del modelo sobre el cuarenta y unavo día. La secuencia del lugar sobre el futuro encadenamiento trata de decir que después de transferir al corriente encadenamiento, el oficio del trabajador debiera estar adelante del segmento de transacción de la demanda si ambas tuvieran que tener el mismo nivel de prioridad.

Esto querá decir que la orden de reabastecimiento llegara antes que el día de la demanda fué terminada lo cual sera una violación de condiciones del problema, haciendo la distinción del nivel de prioridad invertimos la situación.

Discusión

Modelo lógico. En la figura 5.1 el Diagrama de bloques, el modelo usa un bloque TEST en modo negativo. También se observa como los dos modelos de segmento se comunican através del valor salvado en STOCK.

El modelo es diseñado para duplicación de condiciones experimentales. Las dos fuentes de aleatoriedad dan la demanda diaria y tiempo de entrega. Cada uno de los correspondientes distribuciones es representada por una función. (cards del 9 al 16 fig. 5.2) y cada función tiene su propia fuente de numero aleatorio como un argumento. Con la instrucción RMULT (card 5,59,63 Fig. 5.2) esas fuentes de numeros aleatorios son instaladas o reinstaladas para llevar puntos de empiezo antes que la simulación empiese para cada punto de reorden diferente.

Salida del programa la figura 5.3 muestra las tablas de ventas diarias perdidas y el almacenamiento manual producida por la simulación correspondiente a un punto de reorden 80. El número de entradas en cada tabal es de 1000, una por cada día de simulación. En promedio, 0.516 ventas fueron perdidas cada día. Durante el curso de los 1000 días

simulados este promedio de 516 ventas perdidas. Es cerca del 92% de los días, había no perdidas de ventas. En esa ocasión había 14 ventas perdidas en un síple día.

La tabla STOCK (Fig. 5.3) indica que el inventario manual, fue aproximadamente uniformemente distribuido entre 0 y 100 unidades. En promedio el inventario manual fue 47.8 unidades, cerca del 8% del tiempo fué cero al finalizar el día de negocios, se excluye la posibilidad de que una orden de reabastecimiento tiene que ser abonada al fin de ese día. Sobre los 23 días de simulación del inventario al finalizar el día exceda las 100 unidades.

La tabla 5.3 muestra un sumario de la tabla de salida de los 3 puntos de reorden.

Las modas mostradas en la tabla son consistentes con las expectativas. En promedio la perdida diaria decrementa con el punto de reorden es incrementando al mismo tiempo, el promedio del inventario incrementa con un incremento en el punto de reorden.

STUDENT GPSS/H RELEASE 2.0 (AY130) 28 Mar 1993 01:55:22 FILE: Inventor.gps

| LINE# | SMTR# | IF DO | BLOCK# | %LOC | OPERATION | A,B,C,D,E,F,G | COMMENTS |
|-------|-------|-------|--------|------|--|---------------|-----------------------|
| 1 | 1 | | | | SIMULATE | | Case Study Inventario |
| 2 | 2 | | | * | | | Model of a Hospital's |
| | | | | | | | Emergency Room |
| 3 | 3 | | | * | | | Base Time Unit: 1 |
| | | | | | | | Minute |
| 4 | 4 | | | | | | |
| ***** | | | | | | | |
| 5 | 5 | | | * | Control Statements (STORAGE) | | |
| 6 | 6 | | | | | | |
| ***** | | | | | | | |
| 7 | 7 | | | * | | | |
| 8 | 8 | | | | RMULT 11,33 | | |
| 9 | 9 | | | | LTIME FUNCTION RW2,D5 | | |
| 10 | 10 | | | | .05,6/,3,7/.6,8/.82,9/1,10 | | |
| 11 | 11 | | | | SNORM FUNCTION RW1,C25 | | |
| 12 | 12 | | | | 0,-5/.0003,-4/.00135,-3/.00621,-2.5/.02275,-2 | | |
| 13 | 13 | | | | .06681,-1.5/.11507,-1.2/.15866,-1/.21186,-0.8/.27425,-.6 | | |
| 14 | 14 | | | | .34458,-.4/.42074,-.2/.5,0/.57926,.2/.65542,.4 | | |
| 15 | 15 | | | | .72575,.6/.78814,.8/.84134,1/.88493,1.2/.93319,1.5 | | |
| 16 | 16 | | | | .97725,2/.99379,2.5/.99865,3/.99997,4/1,5 | | |
| 17 | 17 | | | | INITIAL XSROP,80 | | |
| 18 | 18 | | | | INITIAL XSROO,100 | | |
| 19 | 19 | | | | INITIAL XSSTOCK,100 | | |
| 20 | 20 | | | | LOSES TABLE P2,0,1,17 | | |
| 21 | 21 | | | | STOCK TABLE XSSTOCK,0,10,12 | | |
| 22 | 22 | | | | DMND FVARIABLE 2*FMSSNORM*10 | | |
| 23 | 23 | | | | LOST VARIABLE P1-XSSTOCK | | |
| 24 | 24 | | | 1 | GENERATE 1,,,,1 | | |
| 25 | 25 | | | 2 | ASSIGN 1,VSDMND | | |
| 26 | 26 | | | 3 | TEST GE XSSTOCK,P1,TRUBL | | |
| 27 | 27 | | | 4 | SAVEVALUE STOCK,P1 | | |
| 28 | 28 | | | 5 | TAB TABULATE STOCK | | |
| 29 | 29 | | | 6 | TABULATE LOSES | | |
| 30 | 30 | | | 7 | TERMINATE 1 | | |
| 31 | 31 | | | 8 | TRUBL ASSIGN 2,V%LOST | | |
| 32 | 32 | | | 9 | SAVEVALUE STOCK,0 | | |
| 33 | 33 | | | 10 | TRANSFER ,TAB | | |
| 34 | 34 | | | 11 | GENERATE ,,,1 | | |
| 35 | 35 | | | 12 | WATCH TEST LE XSSTOCK,XSROP | | |
| 36 | 36 | | | 13 | ADVANCE FM\$TIME | | |
| 37 | 37 | | | 14 | SAVEVALUE STOCK*,XSROO | | |
| 38 | 38 | | | 15 | TRANSFER ,WATCH | | |
| 39 | 39 | | | | START 1000 | | |
| 40 | 40 | | | | RMULT 11,33 | | |
| 41 | 41 | | | | CLEAR XSROO | | |
| 42 | 42 | | | | INITIAL XSROP,90/XSSTOCK,100 | | |
| 43 | 43 | | | | START 1000 | | |
| 44 | 44 | | | | RMULT 11,33 | | |
| 45 | 45 | | | | CLEAR XSROO | | |

| | | | |
|----|----|---------|-------------------------|
| 46 | 46 | INITIAL | X\$ROP,100/X\$STOCK,100 |
| 47 | 47 | START | 1000 |
| 48 | 48 | END | |

ENTITY DICTIONARY (IN ASCENDING ORDER BY ENTITY NUMBER; "*" => VALUE CONFLICT.)

| | | |
|-------------------------|---------|---------|
| Tables: 1=LOSES | 3=STOCK | |
| Functions: 1=LTIME | 2=SNORM | |
| (F)variables: 1=DMND | 2=LOST | |
| Fullword Savexes: 1=ROP | 2=ROO | 3=STOCK |
| Parameters: 1 | 2 | |
| Random Numbers: 1 | 2 | |

SYMBOL VALUE EQU DEFNS CONTEXT REFERENCES BY STATEMENT NUMBER

| | | | | | | | | | | | | | | | | | |
|-------|----|----|--------------|----|----|----|----|----|----|----|----|----|----|--|--|--|--|
| TAB | 5 | 28 | Block | 33 | | | | | | | | | | | | | |
| TRUBL | 8 | 31 | Block | 36 | | | | | | | | | | | | | |
| WATCH | 12 | 35 | Block | 38 | | | | | | | | | | | | | |
| LOSES | 1 | 20 | Table | 29 | | | | | | | | | | | | | |
| STOCK | 3 | 21 | Table | 28 | | | | | | | | | | | | | |
| LTIME | 1 | 9 | Function | 36 | | | | | | | | | | | | | |
| SNORM | 2 | 11 | Function | 22 | | | | | | | | | | | | | |
| DMND | 1 | 22 | (F)variable | 25 | | | | | | | | | | | | | |
| LOST | 2 | 23 | (F)variable | 31 | | | | | | | | | | | | | |
| ROP | 1 | | Fullword Svx | 17 | 35 | 42 | 46 | | | | | | | | | | |
| ROO | 2 | | Fullword Svx | 18 | 37 | 41 | 45 | | | | | | | | | | |
| STOCK | 3 | | Fullword Svx | 19 | 21 | 23 | 26 | 27 | 32 | 35 | 37 | 42 | 46 | | | | |
| 1 | 1 | | Parameter | 23 | 25 | 26 | 27 | | | | | | | | | | |
| 2 | 2 | | Parameter | 20 | 31 | | | | | | | | | | | | |
| 1 | 1 | | Random Nbr | 8 | 11 | 40 | 44 | | | | | | | | | | |
| 2 | 2 | | Random Nbr | 8 | 9 | 40 | 44 | | | | | | | | | | |

STORAGE REQUIREMENTS (BYTES)

| | |
|----------------|-------|
| COMPILED CODE: | 1426 |
| COMPILED DATA: | 792 |
| MISCELLANEOUS: | 0 |
| ENTITIES: | 827 |
| COMMON: | 10000 |
| ----- | |
| TOTAL: | 13045 |

Simulation begins.

RELATIVE CLOCK: 1000.0000 ABSOLUTE CLOCK: 1000.0000

| BLOCK CURRENT | TOTAL | BLOCK CURRENT | TOTAL |
|---------------|-------|---------------|-------|
| 1 | 1000 | 11 | 1 |
| 2 | 1000 | WATCH | 89 |
| 3 | 1000 | 13 | 89 |
| 4 | 901 | 14 | 88 |
| TAB | 1000 | 15 | 88 |
| 6 | 1000 | | |
| 7 | 1000 | | |
| TRUBL | 99 | | |
| 9 | 99 | | |
| 10 | 99 | | |

TABLE LOSSES

| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | NON-WEIGHTED | | |
|------------------|--------------------|--------------------|-----------------------|----------------------|------------------|---------------------|
| 1000.0000 | 0.6950 | 2.3887 | 695.0000 | | | |
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM MEAN |
| 0. | 901.0000 | 90.10 | 90.10 | 9.90 | 0. | -0.2910 |
| 1.0000 | 10.0000 | 1.00 | 91.10 | 8.90 | 1.4388 | 0.1277 |
| 2.0000 | 5.0000 | 0.50 | 91.60 | 8.40 | 2.8777 | 0.5463 |
| 3.0000 | 5.0000 | 0.50 | 92.10 | 7.90 | 4.3165 | 0.9650 |
| 4.0000 | 6.0000 | 0.60 | 92.70 | 7.30 | 5.7554 | 1.3836 |
| 5.0000 | 12.0000 | 1.20 | 93.90 | 6.10 | 7.1942 | 1.8023 |
| 6.0000 | 6.0000 | 0.60 | 94.50 | 5.50 | 8.6331 | 2.2209 |
| 7.0000 | 9.0000 | 0.90 | 95.40 | 4.60 | 10.0719 | 2.6396 |
| 8.0000 | 6.0000 | 0.60 | 96.00 | 4.00 | 11.5108 | 3.0582 |
| 9.0000 | 9.0000 | 0.90 | 96.90 | 3.10 | 12.9496 | 3.4768 |
| 10.0000 | 11.0000 | 1.10 | 98.00 | 2.00 | 14.3885 | 3.8955 |
| 11.0000 | 6.0000 | 0.60 | 98.60 | 1.40 | 15.8273 | 4.3141 |
| 12.0000 | 10.0000 | 1.00 | 99.60 | 0.40 | 17.2662 | 4.7328 |
| 13.0000 | 4.0000 | 0.40 | 100.00 | 0.00 | 18.7050 | 5.1514 |

TABLE STOCK

| ENTRIES IN TABLE | | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | | | NON-WEIGHTED |
|------------------|--------------------|------------------|-----------------------|----------------------|------------------|---------------------|--------------|
| 1000.0000 | | 45.4810 | 31.1088 | 45481.0000 | | | |
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM MEAN | |
| 0. | 105.0000 | 10.50 | 10.50 | 89.50 | 0. | -1.4620 | |
| 10.0000 | 76.0000 | 7.60 | 18.10 | 81.90 | 0.2199 | -1.1405 | |
| 20.0000 | 89.0000 | 8.90 | 27.00 | 73.00 | 0.4397 | -0.8191 | |
| 30.0000 | 95.0000 | 9.50 | 36.50 | 63.50 | 0.6596 | -0.4976 | |
| 40.0000 | 93.0000 | 9.30 | 45.80 | 54.20 | 0.8795 | -0.1762 | |
| 50.0000 | 99.0000 | 9.90 | 55.70 | 44.30 | 1.0994 | 0.1453 | |
| 60.0000 | 88.0000 | 8.80 | 64.50 | 35.50 | 1.3192 | 0.4667 | |
| 70.0000 | 89.0000 | 8.90 | 73.40 | 26.60 | 1.5391 | 0.7882 | |
| 80.0000 | 93.0000 | 9.30 | 82.70 | 17.30 | 1.7590 | 1.1096 | |
| 90.0000 | 88.0000 | 8.80 | 91.50 | 8.50 | 1.9788 | 1.4311 | |
| 100.0000 | 67.0000 | 6.70 | 98.20 | 1.80 | 2.1987 | 1.7525 | |
| OVERFLOW | 18.0000 | 1.80 | 100.00 | 0.00 | | | |

AVERAGE VALUE OF OVERFLOW IS 106.7222

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

ROP: 80, ROQ: 100, STOCK: 40

| RANDOM STREAM | ANTITHETIC VARIATES | INITIAL POSITION | CURRENT POSITION | SAMPLE COUNT | CHI-SQUARE UNIFORMITY |
|---------------|---------------------|------------------|------------------|--------------|-----------------------|
| 1 | OFF | 11 | 1011 | 1000 | 0.86 |
| 2 | OFF | 33 | 122 | 89 | 0.40 |

STATUS OF COMMON STORAGE

9552 BYTES AVAILABLE
448 IN USE
560 USED (MAX)

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

79

RELATIVE CLOCK: 1000.0000 ABSOLUTE CLOCK: 1000.0000

| BLOCK CURRENT | TOTAL | BLOCK CURRENT | TOTAL |
|---------------|-------|---------------|-------|
| 1 | 1000 | 11 | 1 |
| 2 | 1000 | WATCH | 94 |
| 3 | 1000 | 13 | 94 |
| 4 | 962 | 14 | 93 |
| TAB | 1000 | 15 | 93 |
| 6 | 1000 | | |
| 7 | 1000 | | |
| TRUBL | 38 | | |
| 9 | 38 | | |
| 10 | 38 | | |

TABLE LOSES

| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | | | |
|------------------|--------------------|--------------------|-----------------------|----------------------|------------------|---------------------|
| 1000.0000 | 0.2270 | 1.3097 | 227.0000 | NON-WEIGHTED | | |
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM MEAN |
| 0. | 962.0000 | 96.20 | 96.20 | 3.80 | 0. | -0.1733 |
| 1.0000 | 3.0000 | 0.30 | 96.50 | 3.50 | 4.4053 | 0.5902 |
| 2.0000 | 4.0000 | 0.40 | 96.90 | 3.10 | 8.8106 | 1.3538 |
| 3.0000 | 6.0000 | 0.60 | 97.50 | 2.50 | 13.2159 | 2.1174 |
| 4.0000 | 1.0000 | 0.10 | 97.60 | 2.40 | 17.6211 | 2.8809 |
| 5.0000 | 2.0000 | 0.20 | 97.80 | 2.20 | 22.0264 | 3.6445 |
| 6.0000 | 6.0000 | 0.60 | 98.40 | 1.60 | 26.4317 | 4.4080 |
| 7.0000 | 4.0000 | 0.40 | 98.80 | 1.20 | 30.8370 | 5.1716 |
| ... | | | | | | |
| 9.0000 | 5.0000 | 0.50 | 99.30 | 0.70 | 39.6476 | 6.6987 |
| 10.0000 | 4.0000 | 0.40 | 99.70 | 0.30 | 44.0529 | 7.4623 |
| 11.0000 | 1.0000 | 0.10 | 99.80 | 0.20 | 48.4581 | 8.2259 |
| 12.0000 | 2.0000 | 0.20 | 100.00 | -0.00 | 52.8634 | 8.9894 |

TABLE STOCK

| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | | | |
|------------------|--------------------|--------------------|-----------------------|----------------------|------------------|---------------------|
| 1000.0000 | 53.1280 | 30.9325 | 53128.0000 | NON-WEIGHTED | | |
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM MEAN |
| 0. | 45.0000 | 4.50 | 4.50 | 95.50 | 0. | -1.7175 |
| 10.0000 | 53.0000 | 5.30 | 9.80 | 90.20 | 0.1882 | -1.3943 |
| 20.0000 | 84.0000 | 8.40 | 18.20 | 81.80 | 0.3764 | -1.0710 |
| 30.0000 | 95.0000 | 9.50 | 27.70 | 72.30 | 0.5647 | -0.7477 |
| 40.0000 | 105.0000 | 10.50 | 38.20 | 61.80 | 0.7529 | -0.4244 |
| 50.0000 | 93.0000 | 9.30 | 47.50 | 52.50 | 0.9411 | -0.1011 |
| 60.0000 | 98.0000 | 9.80 | 57.30 | 42.70 | 1.1293 | 0.2222 |
| 70.0000 | 96.0000 | 9.60 | 66.90 | 33.10 | 1.3176 | 0.5454 |
| 80.0000 | 99.0000 | 9.90 | 76.80 | 23.20 | 1.5058 | 0.8687 |
| 90.0000 | 102.0000 | 10.20 | 87.00 | 13.00 | 1.6940 | 1.1920 |
| 100.0000 | 74.0000 | 7.40 | 94.40 | 5.60 | 1.8822 | 1.5153 |
| OVERFLOW | 56.0000 | 5.60 | 100.00 | 0.00 | | |

AVERAGE VALUE OF OVERFLOW IS 108.4286

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

RDP: 90, ROD: 100, STOCK: 72

| RANDOM STREAM | ANTITHETIC VARIATES | INITIAL POSITION | CURRENT POSITION | SAMPLE COUNT | CHI-SQUARE UNIFORMITY |
|------------------|------------------------|---------------------|---------------------|-----------------|--------------------------|
| 1 | OFF | 11 | 1011 | 1000 | 0.86 |
| 2 | OFF | 33 | 127 | 94 | 0.32 |

STATUS OF COMMON STORAGE

9552 BYTES AVAILABLE

448 IN USE

560 USED (MAX)

RELATIVE CLOCK: 1000.0000 ABSOLUTE CLOCK: 1000.0000

| BLOCK CURRENT | TOTAL | BLOCK CURRENT | TOTAL |
|---------------|-------|---------------|-------|
| 1 | 1000 | 11 | 1 |
| 2 | 1000 | WATCH | 95 |
| 3 | 1000 | 13 | 95 |
| 4 | 988 | 14 | 95 |
| TAB | 1000 | 15 | 1 |
| 6 | 1000 | | |
| 7 | 1000 | | |
| TRUBL | 12 | | |
| 9 | 12 | | |
| 10 | 12 | | |

TABLE LOSSES

| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | | | |
|------------------|---------------|--------------------|------------------|--------------|----------|-----------|
| 1000.0000 | 0.0730 | 0.8101 | 73.0000 | NON-WEIGHTED | | |
| UPPER | OBSERVED | PERCENT | CUMULATIVE | CUMULATIVE | MULTIPLE | DEVIATION |
| LIMIT | FREQUENCY | OF TOTAL | PERCENTAGE | REMAINDER | OF MEAN | FROM MEAN |
| 0. | 988.0000 | 98.80 | 98.80 | 1.20 | 0. | -0.0901 |
| 1.0000 | 1.0000 | 0.10 | 98.90 | 1.10 | 13.6986 | 1.1442 |
| 2.0000 | 1.0000 | 0.10 | 99.00 | 1.00 | 27.3973 | 2.3786 |
| 3.0000 | 3.0000 | 0.30 | 99.30 | 0.70 | 41.0959 | 3.6130 |
| 4.0000 | 2.0000 | 0.20 | 99.50 | 0.50 | 54.7945 | 4.8473 |
| ... | | | | | | |
| 8.0000 | 2.0000 | 0.20 | 99.70 | 0.30 | 109.5890 | 9.7847 |
| ... | | | | | | |
| 10.0000 | 1.0000 | 0.10 | 99.80 | 0.20 | 136.9863 | 12.2534 |
| ... | | | | | | |
| 12.0000 | 1.0000 | 0.10 | 99.90 | 0.10 | 164.3836 | 14.7221 |
| ... | | | | | | |
| 15.0000 | 1.0000 | 0.10 | 100.00 | 0.00 | 205.4795 | 18.4252 |

TABLE STOCK

| ENTRIES IN TABLE | | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | | | |
|------------------|-----------------------|---------------------|--------------------------|-------------------------|---------------------|------------------------|--|
| 1000,0000 | | 62.2090 | 31.6500 | 62209.0000 | NON-WEIGHTED | | |
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM MEAN | |
| 0. | 14,0000 | 1.40 | 1.40 | 98.60 | 0. | -1.9655 | |
| 10,0000 | 31,0000 | 3.10 | 4.50 | 95.50 | 0.1607 | -1.6496 | |
| 20,0000 | 63,0000 | 6.30 | 10.80 | 89.20 | 0.3215 | -1.3336 | |
| 30,0000 | 80,0000 | 8.00 | 18.80 | 81.20 | 0.4822 | -1.0177 | |
| 40,0000 | 94,0000 | 9.40 | 28.20 | 71.80 | 0.6430 | -0.7017 | |
| 50,0000 | 106,0000 | 10.60 | 38.80 | 61.20 | 0.8037 | -0.3858 | |
| 60,0000 | 97,0000 | 9.70 | 48.50 | 51.50 | 0.9645 | -0.0698 | |
| 70,0000 | 93,0000 | 9.30 | 57.80 | 42.20 | 1.1252 | 0.2462 | |
| 80,0000 | 98,0000 | 9.80 | 67.60 | 32.40 | 1.2860 | 0.5621 | |
| 90,0000 | 100,0000 | 10.00 | 77.60 | 22.40 | 1.4467 | 0.8781 | |
| 100,0000 | 95,0000 | 9.50 | 87.10 | 12.90 | 1.6075 | 1.1940 | |
| OVERFLOW | 129,0000 | 12.90 | 100.00 | 0.00 | | | |

AVERAGE VALUE OF OVERFLOW IS 111.6822

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

ROP: 100, ROQ: 100, STOCK: 118

| RANDOM STREAM | ANTITHETIC VARIATES | INITIAL POSITION | CURRENT POSITION | SAMPLE COUNT | CHI-SQUARE UNIFORMITY |
|------------------|------------------------|---------------------|---------------------|-----------------|--------------------------|
| 1 | OFF | 11 | 1011 | 1000 | 0.86 |
| 2 | OFF | 33 | 128 | 95 | 0.43 |

STATUS OF COMMON STORAGE

9552 BYTES AVAILABLE
448 IN USE
560 USED (MAX)

Simulation terminated. Absolute Clock: 1000.0000

Total Block Executions: 22409

Blocks / second: 553

Microseconds / Block: 1806.86

Elapsed Time Used (SEC)

PASS1: 1.76
SYN/REF 0.17
PASS2: 1.65
LOAD/CTRL: 1.97
EXECUTION: 40.49
OUTPUT: 3.34

TOTAL: 49.38

VI. CONCLUSIONES

La aportación de esta obra en la FCA, es combinar una amalgama de conocimientos matemáticos, probabilísticos, lógicos, estadísticos, e Informáticos, en donde se interrelacionan todos estos elementos para formular una simulación consistente, que pueda representar un sistema real, con todas las características necesarios para predecir su comportamiento de un modo adecuado.

Lo importante al realizar una simulación es realizar un buen modelo, para que al introducir las variables controlables y las incontrolables, se puedan obtener medidas de efectividad.

- La planeación de una buena simulación es observar el comportamiento del sistema real y transportarlo a un modelo, el cual debe ser, lo más sencillo posible para que se pueda acercar con exactitud, y después se le puedan incrementar más interrelaciones.
- Se debe realizar un análisis de lo que se va a simular y si realmente requiere de simulación.
- Realizar un modelo acorde con la realidad que contemple todos los insumos necesarios para poder predecir su comportamiento.

- Comparar el modelo con datos reales para observar su confiabilidad de esta manera.
- Toda simulación tiene como insumo variables controlables y variables incontrolables, que tienen cierta probabilidad de ocurrir, con esto al simular obtenemos variables de efectividad que debemos interpretar para la toma de decisiones.

RECOMENDACIONES

- No realizar simulación si hay procesos analíticos conocidos para la solución del problema.
- Determinar que riesgos se tendría al efectuar una simulación inadecuada.
- Correr el modelo de simulación con datos reales para comprobar su validez.
- Realizar varias pruebas al modelo de simulación.

Consideramos que con lo anterior, se cubre el objetivo de la presente obra, en donde los profesionales podrán emplearla para los casos que ameriten de emplear la simulación.

A 50 años de darse a conocer las mejores técnicas de simulación a nivel mundial, se observa que México, no está aislado de estos cambios y es por lo que espero sea de gran

utilidad a nuestra Facultad, que se preocupa por mantener una gran calidad en sus profesionales.

Deseo agradecer, finalmente, la dedicación incondicional para la elaboración de este trabajo a la M. en A. y C.P. Nadima Simón Domínguez.

ANEXO 1: REFERENCIA RAPIDA DEL GPSSH

Para instalar en discos flexibles:

```
hinstall b: gpssh.exe
hinstall b: gpsshcorr.msg
hisntall b: profile.hdb
2o. disco
hinstall b: mod*.gps
hinstall b: fig*.gps
hisntall b: s*.gps
```

sintaxis GPSSH <FILE.GPS> [OPCIONES]
 OPCIONES (las mayusculas son default) (separadas con espacios)

| | |
|-----------------------------|---|
| DICT/nodict | Da salida o excluye el diccionario. |
| SOURCE | El programa fuente es dirigido por default al disco |
| /nosource | a cualquier otro dispositivo el default es NOSOURCE. |
| TEST | Esta opción es para depurar (al igual que TV, TVTNW y TTNW) La pantalla entera es usada como dialogo. |
| TTNW | Abreviación de las tres opciones test type nowarn. |
| TV | Puede ser usada para especificar que un test-mode toma lugar, la pantalla es dividida en tres ventanas: Source, Status y Dialog. Para alternar entre pantalla completa y ventaneo, se pueden utilizar los comandos: set tv on y set tv off. |
| TYPE | Esta opción da salida en la pantalla al tiempo de |
| Compilación | los mensajes preventivos. El listado fuente, Diccionario, y la Referencia cruzada no seran desplegados, porque el default es NOS NODICT y NOXREF. |
| WARN/ nowarn default. | Esta opción manifiesta los mensajes de prevención. En el Test-mode iniciado por TEST o TV, WARM es el En el Test-mode por TVTNW O TTNW, NOWARN es el default. En las simulaciones por lote el modo WARN es el default. |

Tabla 10.1-----

Tipos de selección que pueden ser desplegados en el Modo Test.

| | |
|---------------|---|
| Modificador | |
| de despliegue | |
| no | Current y Total de Bloque contados. |
| breakpoints | una lista de bloque puntos de interrupción y condiciones Trap. |
| cec | Current Event Chains (Cadena de eventos actuales) |
| clocks | Valores del reloj absoluto y relativo. |
| fac | el reporte de facilidad (facility) |
| output | el caompleta reporte de postsimulación. |
| que | el reporte de cola (que) |
| traps | una lista de condiciones traps |
| xact=xact_id | las propiedades del Xact cuyo id es xact_id (ej.,xact=25) |

tabla 4.2-----

| | |
|-----|---|
| F1 | Zoom a la linea más alta del dialogo todavía en la memoria. |
| F2 | Pantalla completa (set tv off) |
| F3 | Zoom a la limea más baja (la reciente) linea del dialogo. |
| F4 | Ventanas Dialogo, Fuente y Estatus (set tv on) |
| F5 | Scroll 20 columnas a la izquierda en las ventanas del dialogo y fuente. |
| F6 | Scroll 10 columnas a la derecha en las ventanas del dialogo y fuente. |
| F7 | Scroll up (una linea) la ventanas del fuente. |
| F9 | Scroll down (una linea) la ventanas del fuente. |
| F10 | es usado para incrementar 1 (step 1 command). |

Commando trap -----

```
trap scan
untrap scan
      clock
      next
      system
      xact
```



MATEMÁTICAS IV



CLAVE: 93

PLAN: 93

LICENCIATURA (SEMESTRE): ADMINISTRACION, CONTADURIA E INFORMATICA (5^o, 4^o, 5^o)

DEPTO. ACADÉMICO: MATEMÁTICAS

AREA: MATEMÁTICAS

REQUISITOS: NINGUNA ASIGNATURA

CREDITOS: 8

HORAS POR CLASE: 2

CLASES POR SEMANA: 2

HORAS POR SEMESTRE: 68

OBJETIVO GENERAL: AL FINALIZAR EL CURSO EL ALUMNO FORMULARA Y RESOLVERA MODELOS DETERMINISTICOS Y PROBABILISTICOS UTILIZANDO LA METODOLOGIA DE LA INVESTIGACION DE OPERACIONES.

| HORAS | TEMATICA | OBJETIVOS EDUCACIONALES | SUGERENCIAS DIDACTICAS | REFERENCIAS BIBLIOGRAFICAS |
|-------|--|---|--|--|
| 4 | I. INTRODUCCION. 1. Origen y naturaleza de la investigación de operaciones. 2. Concepto de optimización. 3. Modelos de la investigación de operaciones. 4. Metodología de la investigación de operaciones. | 1. PARTICULAR DE LA UNIDAD. Al finalizar esta unidad, el alumno describirá la historia, la naturaleza y la aplicación de la investigación de operaciones así como su metodología. 2. ESPECIFICOS. El alumno será capaz de: Explicar que es el enfoque de sistemas y su relación con la investigación de operaciones. Relatar la historia de la investigación de operaciones. Señalar ejemplos de modelos que se aplican en esta disciplina. Describir la metodología de la investigación de operaciones. | Revisión individual de las referencias bibliográficas sobre el origen, naturaleza, metodología y modelos de la investigación de operaciones. Lecturas comentadas sobre el origen y naturaleza de la investigación de operaciones. Exposición oral acerca de los modelos más importantes de la investigación de operaciones. Audiovisual sobre la historia de la investigación de operaciones. | 2. 1,2 3 a 26 4. 1 18.6 1. 1,2,3 1a 112 5. 1 19 a 37 |
| 18 | II. PROGRAMACION LINEAL 1. Concepto de Programación Lineal. | 1. PARTICULAR DE LA UNIDAD. Al finalizar esta unidad el alumno formulará modelos de programación lineal y encontrará | Ejercicios de los diferentes algoritmos de programación lineal. | 2. 3 a 9 29 a 330 4. 10,11 256 a 392 1. 5,6 141 a 196; 3. 2,3 57 a 326; |

| | | | |
|--|--|--|---|
| <p>2. Métodos de solución.</p> <p>A) Método gráfico B) Método dual-simplex C) Método de uso de computadores.</p> <p>3. El modelo de transporte.</p> <p>4. El modelo de asignación.</p> | <p>la solución óptima.</p> <p>2. ESPECÍFICOS.</p> <p>El alumno será capaz de: Resolver modelos de programación lineal por los métodos gráfico, dual-simplex.</p> <p>Resolver modelos de asignación y de transporte.</p> <p>Resolver ejercicios de los modelos anteriores con software de aplicación.</p> | <p>Prácticas con software de aplicación para la resolución de ejercicios de programación lineal.</p> <p>Ejemplos de aplicación de programación lineal.</p> | <p>5. 2, 3, 4, 21, 24, 25</p> |
| <p>12</p> <p>III. REDES</p> <p>1. Conceptos.</p> <p>2. Problema del árbol de peso mínimo.</p> <p>3. Problema de la ruta más corta.</p> <p>4. Problema de flujo máximo.</p> <p>5. CPN</p> <p>6. PERT/CPM y PERT/Tiempo.</p> | <p>1. PARTICULAR DE LA UNIDAD.</p> <p>A) finalizar esta unidad el alumno analizará modelos de redes, a fin de plantear y encontrar la solución óptima.</p> <p>2. ESPECÍFICOS.</p> <p>El alumno será capaz de: Formular y resolver problemas de ruta más corta.</p> <p>Formular y resolver problemas de flujo máximo en una red.</p> <p>Formular y resolver problemas de la ruta crítica en una red.</p> <p>Formular y resolver problemas de PERT/ Costo.</p> <p>Formular y resolver problemas de PERT/</p> | <p>Resolución de ejercicios de ruta más corta.</p> <p>Resolución de ejercicios de flujo máximo.</p> <p>Resolución de ejercicios de ruta crítica.</p> <p>Resolución de ejercicios PERT.</p> <p>Práctica con software de aplicación.</p> <p>Ejemplos de aplicación de redes.</p> | <p>2. 10 313 a 392</p> <p>1. 11, 12 307 a 358</p> <p>3. 4 329 a 436</p> <p>5. 6 234 a 259</p> <p>12 515 a 551</p> |

| HORAS | TEMATICA | OBJETIVOS EDUCACIONALES | SUGERENCIAS DIDACTICAS | REFERENCIAS BIBLIOGRAFICAS No. Cap. o lib. Págs. |
|-------|---|--|---|---|
| 12 | <p>IV. SISTEMAS DE CONTROL DE INVENTARIOS.</p> <ol style="list-style-type: none"> 1. Problema general de un modelo de inventario. 2. Modelo de lote económico clásico. <ol style="list-style-type: none"> A) Propiedades del modelo B) Caso con fallantes. C) Caso con ventas perdidas. D) Casos con tasa de producción finita. E) Caso con descuentos por cantidad. | <p>Tiempo. Resolver ejercicios de los modelos propuestos con software de aplicación.</p> <p>1. PARTICULAR DE LA UNIDAD. Al finalizar esta unidad el alumno: Categorizará los principales casos de los sistemas de control de inventario</p> <p>2. ESPECIFICOS. El alumno será capaz de: Analizar el problema general de un modelo de inventario. Analizar el modelo de lote económico clásico Categorizar las propiedades del modelo del lote económico clásico. Resolver ejemplos de los sistemas de control de inventarios con software de aplicación.</p> | <p>Ejemplos de aplicación de sistemas de control de inventarios</p> <p>Resolución de problemas reales acerca de los diferentes casos del modelo económico clásico.</p> <p>Práctica con software de aplicación acerca de Sistemas de Control de Inventario</p> | <ol style="list-style-type: none"> 2. IS (687 a 714) 4. 6 (126 a 163) 1. 7 (197 a 238) 3. 2 (93 a 212) 5. 13 (552 a 625) |
| 8 | <p>V. LINEAS DE ESPERA</p> <ol style="list-style-type: none"> 1. Terminología. 2. Estructura básica de una línea de espera 3. Modelos de una cola con un servidor. 4. Modelos de una cola con servidores múltiples en paralelo. 5. Modelos de una cola con servidores múltiples en serie. 6. Comportamiento prioritario de una línea de espera. | <p>1. PARTICULAR DE LA UNIDAD. Al finalizar la unidad el alumno categorizará modelos de líneas de espera.</p> <p>2. ESPECIFICOS. El alumno será capaz de: Emplear la terminología del modelo de la línea de espera.</p> | <p>Ejemplos de aplicación.</p> <p>Ejercicios.</p> <p>Práctica de aplicaciones.</p> | <ol style="list-style-type: none"> 2. 16,17 (594 a 686) 4. 8 (198 a 237) 1. 10 (279 a 306) 3. 3 (243 a 313) 5. 13,16 (653 a 761) |

| HORAS | TEMATICA | OBJETIVOS EDUCACIONALES | SUGERENCIAS DIDACTICAS | REFERENCIAS BIBLIOGRAFICAS N.º. Cap. e Ed. Pág. |
|-------|---|---|---|--|
| 8 | VI. TEORIA DE JUEGOS. 1. Definición de juego. A) Estrategias puras y mixtas. 2. Solución óptima de juegos bipersonales y de suma cero. A) Solución gráfica para juegos (2x2) B) (Mz). B) Teoremas de minimax. C) Resolución por programación lineal. | Examinar la estructura básica de la línea de espera. Analizar los modelos de líneas de espera. Probar el comportamiento prioritario de una línea de espera. Resolver ejemplos de líneas de espera con software de aplicación. 1. PARTICULAR DE LA UNIDAD. Al finalizar esta unidad el alumno: Aplicará la teoría de juegos para resolver problemas administrativos o contables. 2. ESPECIFICOS. El alumno será capaz de: Diferenciar las estrategias puras y mixtas. Resolver problemas de juegos bipersonales y de suma cero dando la solución óptima. Resolver problemas de juegos por el método de gráficas. Utilizar la programación lineal para resolver juegos bipersonales y de suma cero. | Revisión individual de bibliografía sobre las estrategias puras y mixtas. Ejercicios de los diferentes métodos de resolución de problemas. Ejemplos de aplicación de la teoría de juegos. | 2. 12 3. 11 5. 11 435 a 456 725 a 750 492 a 514 |
| 6 | EVALUACION DEL APRENDIZAJE | | | |

BIBLIOGRAFÍA:

BÁSICA.

1. ACKOFF R. y L. SASIEMI, Fundamentos de Investigaciones de Operaciones, México: Limusa, 1977, 502 pp.
2. HILLER F. y G. LIEBERMAN, Introducción a la Investigación de Operaciones, (5a. ed.), México: McGraw-Hill, 1991, 956 pp.
3. PRAWDA J., Métodos y Modelos de Investigación de Operaciones, Vol. I, Vol. II, México: Limusa, 1981, 1036 pp.
4. SHAMBLIN, J. y G. STEVENS, Investigaciones de Operaciones, México: McGraw-Hill, 1975, 423 pp.
5. TAHA H., Investigación de Operaciones, (4a. ed.), México: Alfa-Omega, 1991, 989 pp.

COMPLEMENTARIA.

6. BAZARAA M., y J. JARVIS, Programación lineal y flujo en red, México: Limusa, 1980, 539 pp.
7. BRONSON R., Investigación de Operaciones, México: McGraw-Hill, 1984, 352 pp.
8. BUENO de A. G., Introducción a la programación lineal y al análisis de sensibilidad, México: Trilha, 1990, 189 pp.
9. DAELLENBACH H., GEORGE J., y D. MCNICKLE, Introducción a Técnicas de Investigación de Operaciones, México: CECSA, 1986, 711 pp.
10. GALLAGHER C. y H. WATSON, Métodos cuantitativos para la toma de decisiones, México: McGraw-Hill, 1982, 729 pp.
11. MOSKOWITZ H. y G. WRIGHT, Investigación de Operaciones, México: Prentice Hall, 1982, 790 pp.

APENDICE "A": TABLAS ESTADISTICAS

VALORES CRITICOS PARA EL TEST DE KOLMOGOROV-SMIRNOV

En esta tabla aparecen tabulados los valores de D_n a tales que

$$\alpha \leq P(D_n > D_n, \alpha)$$

para distintos valores de α y n .

| $\alpha =$ | .10 | .05 | .025 | .01 | .005 | $\alpha =$ | .10 | .05 | .025 | .01 | .005 |
|------------|------|------|------|------|------|---------------|-------------|-------------|-------------|-------------|-------------|
| $n=1$ | .900 | .950 | .975 | .990 | .995 | $n = 21$ | .226 | .159 | .187 | .321 | .344 |
| 2 | .684 | .776 | .841 | .900 | .919 | 21 | .221 | .153 | .181 | .314 | .337 |
| 3 | .565 | .636 | .708 | .785 | .829 | 23 | .216 | .147 | .175 | .307 | .330 |
| 4 | .493 | .565 | .614 | .689 | .734 | 24 | .212 | .141 | .169 | .301 | .323 |
| 5 | .447 | .509 | .563 | .617 | .669 | 25 | .208 | .138 | .164 | .295 | .317 |
| 6 | .410 | .468 | .519 | .577 | .617 | 26 | .204 | .133 | .159 | .290 | .311 |
| 7 | .381 | .436 | .483 | .538 | .576 | 27 | .100 | .119 | .154 | .284 | .305 |
| 8 | .358 | .410 | .454 | .507 | .542 | 28 | .197 | .125 | .150 | .279 | .300 |
| 9 | .339 | .387 | .430 | .480 | .513 | 29 | .193 | .121 | .146 | .275 | .295 |
| 10 | .323 | .369 | .409 | .457 | .489 | 30 | .190 | .118 | .142 | .270 | .290 |
| 11 | .308 | .352 | .391 | .437 | .468 | 31 | .187 | .114 | .138 | .266 | .285 |
| 12 | .296 | .338 | .375 | .419 | .449 | 31 | .184 | .111 | .134 | .262 | .281 |
| 13 | .285 | .325 | .361 | .404 | .432 | 33 | .182 | .108 | .131 | .258 | .277 |
| 14 | .275 | .314 | .349 | .390 | .418 | 34 | .179 | .105 | .127 | .254 | .273 |
| 15 | .266 | .304 | .338 | .377 | .404 | 35 | .177 | .102 | .124 | .251 | .269 |
| 16 | .258 | .295 | .327 | .366 | .392 | 36 | .174 | .100 | .121 | .247 | .265 |
| 17 | .250 | .286 | .318 | .355 | .381 | 37 | .172 | .106 | .118 | .244 | .262 |
| 18 | .244 | .279 | .309 | .346 | .371 | 38 | .170 | .104 | .115 | .241 | .258 |
| 19 | .237 | .271 | .301 | .337 | .361 | 39 | .168 | .101 | .113 | .238 | .255 |
| 20 | .232 | .265 | .294 | .329 | .352 | 40 | .165 | .109 | .110 | .235 | .252 |
| | | | | | | Aproximación | <u>1.07</u> | <u>1.22</u> | <u>1.35</u> | <u>1.52</u> | <u>1.63</u> |
| | | | | | | para $n > 40$ | \sqrt{n} | \sqrt{n} | \sqrt{n} | \sqrt{n} | \sqrt{n} |

VALORES CRITICOS EN LA DISTRIBUCION JI-CUADRADA

En esta tabla aparecen tabulados los valores $\chi^2_{n,p}$ tales que

$$P(\chi^2_n \geq \chi^2_{n,p}) = p$$

Con $p = 0.99, 0.98, \dots$ y $n = 1, 2, \dots, 30$.

| Grados de libertad | $p = 0.99$ | 0.98 | 0.95 | 0.90 | 0.80 | 0.70 | 0.50 | 0.30 | 0.20 | 0.10 | 0.05 | 0.02 |
|--------------------------|------------|----------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 1 | 0.000157 | 0.000628 | 0.00393 | 0.0158 | 0.0612 | 0.118 | 0.455 | 1.074 | 1.642 | 2.706 | 3.841 | 5.412 |
| 2 | 0.0201 | 0.0404 | 0.103 | 0.211 | 0.116 | 0.713 | 1.386 | 2.408 | 3.219 | 4.605 | 5.991 | 7.824 |
| 3 | 0.115 | 0.185 | 0.352 | 0.581 | 1.005 | 1.424 | 2.366 | 3.665 | 4.642 | 6.251 | 7.815 | 9.837 |
| 4 | 0.267 | 0.429 | 0.711 | 1.064 | 1.619 | 2.195 | 3.357 | 4.787 | 5.989 | 7.779 | 9.488 | 11.668 |
| 5 | 0.554 | 0.752 | 1.115 | 1.610 | 2.343 | 3.000 | 4.351 | 6.064 | 7.289 | 9.236 | 11.070 | 13.388 |
| 6 | 0.872 | 1.134 | 1.635 | 2.204 | 3.070 | 3.828 | 5.348 | 7.231 | 8.558 | 10.645 | 12.592 | 15.033 |
| 7 | 1.239 | 1.564 | 2.167 | 2.833 | 3.821 | 4.671 | 6.346 | 8.383 | 9.803 | 12.017 | 14.067 | 16.611 |
| 8 | 1.646 | 2.032 | 2.733 | 3.490 | 4.594 | 5.527 | 7.344 | 9.524 | 11.030 | 13.362 | 15.507 | 18.168 |
| 9 | 2.088 | 2.532 | 3.329 | 4.168 | 5.380 | 6.393 | 8.343 | 10.656 | 12.242 | 14.684 | 16.919 | 19.679 |
| 10 | 2.558 | 3.059 | 3.940 | 4.865 | 6.179 | 7.167 | 9.342 | 11.781 | 13.442 | 15.987 | 18.307 | 21.161 |
| 11 | 3.053 | 3.609 | 4.575 | 5.578 | 6.989 | 8.118 | 10.341 | 12.899 | 14.631 | 17.275 | 19.675 | 22.618 |
| 12 | 3.571 | 4.178 | 5.226 | 6.304 | 7.807 | 9.034 | 11.340 | 14.011 | 15.812 | 18.549 | 21.016 | 24.054 |
| 13 | 4.107 | 4.765 | 5.892 | 7.042 | 8.631 | 9.926 | 12.340 | 15.119 | 16.985 | 19.812 | 22.361 | 25.471 |
| 14 | 4.660 | 5.368 | 6.571 | 7.790 | 9.167 | 10.821 | 13.339 | 16.222 | 18.151 | 21.064 | 23.685 | 26.873 |
| 15 | 5.229 | 5.985 | 7.261 | 8.547 | 10.307 | 11.721 | 14.339 | 17.311 | 19.311 | 22.307 | 24.966 | 28.159 |
| 16 | 5.812 | 6.614 | 7.962 | 9.312 | 11.152 | 12.624 | 15.338 | 18.418 | 20.465 | 23.542 | 26.276 | 29.631 |
| 17 | 6.408 | 7.255 | 8.672 | 10.085 | 12.002 | 13.531 | 16.338 | 19.511 | 21.615 | 24.669 | 27.587 | 30.995 |
| 18 | 7.015 | 7.906 | 9.390 | 10.865 | 12.857 | 14.440 | 17.338 | 20.601 | 22.760 | 25.989 | 28.669 | 31.346 |
| 19 | 7.633 | 8.567 | 10.117 | 11.651 | 13.716 | 15.352 | 18.338 | 21.689 | 23.900 | 27.204 | 30.144 | 33.687 |
| 20 | 8.260 | 9.237 | 10.851 | 12.443 | 14.578 | 16.266 | 19.337 | 22.775 | 25.038 | 28.412 | 31.410 | 35.120 |
| 21 | 8.897 | 9.915 | 11.591 | 13.240 | 15.445 | 17.182 | 20.337 | 23.858 | 26.171 | 29.615 | 32.671 | 36.343 |
| 22 | 9.542 | 10.600 | 12.338 | 14.041 | 16.314 | 18.101 | 21.337 | 24.939 | 27.301 | 30.813 | 33.914 | 37.659 |
| 23 | 10.196 | 11.293 | 13.091 | 14.848 | 17.187 | 19.021 | 22.337 | 26.018 | 28.429 | 32.007 | 35.172 | 38.963 |
| 24 | 10.856 | 11.992 | 13.848 | 15.659 | 18.062 | 19.943 | 23.337 | 27.076 | 29.553 | 33.196 | 36.415 | 40.270 |
| 25 | 11.524 | 12.697 | 14.611 | 16.473 | 18.940 | 20.867 | 24.337 | 28.172 | 30.675 | 34.382 | 37.652 | 41.566 |
| 26 | 12.198 | 13.409 | 15.379 | 17.292 | 19.820 | 21.792 | 25.336 | 29.246 | 31.795 | 35.563 | 38.885 | 42.856 |
| 27 | 12.879 | 14.125 | 16.151 | 18.114 | 20.703 | 22.719 | 26.336 | 30.319 | 32.912 | 36.741 | 40.113 | 44.140 |
| 28 | 13.565 | 14.847 | 16.928 | 18.939 | 21.588 | 23.647 | 27.336 | 31.391 | 34.027 | 37.916 | 41.337 | 45.419 |
| 29 | 14.256 | 15.574 | 17.708 | 19.768 | 22.475 | 24.577 | 28.336 | 32.461 | 35.139 | 39.087 | 42.557 | 46.693 |
| 30 | 14.953 | 16.306 | 18.493 | 20.599 | 23.364 | 25.508 | 29.336 | 33.530 | 36.250 | 40.256 | 43.773 | 47.962 |

**APENDICE "B": GENERAL PURPOSE SIMULATION SYSTEM / HIGH
(GPSS/H)**

El GPSS (Sistema de Simulación de Propósito General) es básicamente un sistema que se basa en el manejo de transacciones cuyas características las determina el usuario. Este paquete fué elegido para desarrollar este trabajo por tener la característica de que puede correr bajo las mismas instrucciones en diferentes ambientes, es decir, en equipos de computo grandes, medianos y chicos (mainframes, minicomputadoras y microcomputadoras) en donde vamos a trabajar ya que cualquier persona puede tener acceso a este equipo.

Los programas se pueden generar bajo cualquier editor de textos (ejemplo: SideKick, Word, etc). una vez que se tiene creado el archivo se teclea "gpssh" seguido del archivo creado para que este sea ejecutado.

Acontinuación vamos a ver una tabla de como son manejados los comandos en este paquete y después el uso de las instrucciones que posee.

TIPOS DE BLOQUES DE GPSS

| INSTRUCCION | A | B | F |
|-------------|-------------------------------------|---------------------------------------|-----------|
| C | D | E | F |
| ADVANCE | Media | Modificador | |
| ASSIGN | Parám Núm. (±) | Fuente | |
| DEPART | Co'a Núm. | (Unidades) | |
| ENTER | Almacenaje Núm. | (Unidades) | |
| GATE | Artículo Núm. | (Sigue bloque B) | |
| GENERATE | Media (Desplazamiento) | Modificador (Cuenta) (Prioridad) | (params.) |
| LEAVE | Almacenaje Núm. | (Unidades) | |
| LINK | Cadena Núm. | | |
| LOGIC < | R S I Switch Núm. | | |
| MARK | (Parám. Núm) | | |
| PRIORITY | Prioridad | | |
| QUEUE | Cola Núm. | (Unidades) | |
| RELEASE | Facilidad Núm. | | |
| SAVEVALUE | Reserva Valor Núm. | SNA | |
| TABULATE | Núm. Tabla | (Unidades) | |
| TERMINATE | (Unidades) | | |
| TEST | Arg.1 | Arg. 2 | |
| TRANSFER | Factor selec. (Sig. Bloque B) | Sig. Bloque A | |
| UNLINK | Núm. Cadena (Núm. param.) (arg.) | Siguiente Bloque A (Sig. Bloque B) | Cuenta |

() indica campo opcional.

COMANDOS DE CONTROL DE GPSS

| LOCALIZACION | OPERACION | A | B | E | F |
|----------------|------------|------------------------|-------------|--------------------|---|
| C | D | E | F | | |
| | CLEAR | | | | |
| | END | | | | |
| Núm. Func. | FUNCTION | Argumento < | C D L | Núm. de puntos | |
| | INITIAL Xn | Valor | | | |
| | JOB | | | | |
| | RESET | | | | |
| | SIMULATE | | | | |
| | START | Cuanta de Corrida (NP) | | | |
| Núm almacenaje | STORAGE | Capacidad | | | |
| Núm tabla | TABLE | Argumento | | Lim. inferior | |
| | | Intervalo | | Núm. de intervalos | |

GENERATE Sirve para crear transacciones.

TERMINATE Saca de la simulación las transacciones.

A través del sistema simulado se mueven las entidades que dependen de la naturaleza del sistema. Por ejemplo, un sistema de comunicaciones se refiere al movimiento de mensajes, un sistema de transportación en carreteras se refiere a vehículos motorizados y un sistema de procesamiento de datos a registros, etc. En la simulación se llama transacciones a estas entidades. La secuencia de eventos en el tiempo real se refleja en el movimiento de las transacciones de bloque a bloque en el tiempo simulado.

Las transacciones se crean en uno o más bloques de **GENERATE** y se sacan de la simulación en bloques de **TERMINATE**. Pueden haber muchas transacciones moviéndose simultáneamente a través del diagrama de bloques. Cada transacción siempre se posiciona en un bloque y la mayoría de los bloques pueden mantener simultáneamente a muchas transacciones. La transferencia de una transacción de un bloque, a otro ocurre instantáneamente en un tiempo específico o cuando ocurre algún cambio en la condición del sistema.

Un diagrama de bloques de GPSS puede consistir en muchos bloques hasta cierto límite prescrito por el programa (que generalmente se fija en 1000). A cada bloque se da un

número de identificación llamado localización, y por lo general el movimiento de las transacciones va de un bloque al que tienen la localización más alta. Las localizaciones se asignan automáticamente mediante un programa ensamblador dentro del GPSS, de manera que cuando se codifica un problema, los bloques se listan en orden secuencial. A los bloques que se necesita identificar en la programación de problemas (por ejemplo, en puntos a donde se hace una transferencia) se les dá un nombre simbólico. El programa ensamblador asocia el nombre con la localización apropiada. Los nombres simbólicos de bloques y otras entidades del programa deben de ir desde tres hasta cinco caracteres no blancos de los cuales los tres primeros deben de ser letras.

Tiempos de acción

La hora del reloj se representa mediante un número entero y el intervalo del tiempo real corresponde a una unidad de tiempo que escoge el usuario del programa. La unidad de tiempo no se expresa específicamente si no que se implica dando todos los tiempos en términos de la misma unidad. Un tipo de bloque conocido como **ADVANCE** se refiere a representar el gasto de tiempo. EL programa calcula un intervalo conocido como tiempo de acción para cada transacción conforme entra a un bloque de **ADVANCE**, y la transacción permanece en el bloque durante este intervalo

simulado antes de intentar proseguir. El único otro tipo de bloque que emplea tiempo de acción es el bloque GENERATE, que crea transacciones. EL tiempo de acción en el bloque GENERATE controla el intervalo entre llegadas sucesivas de transacciones.

EL tiempo de acción puede ser un intervalo fijo (incluyendo cero) o una variable aleatoria, y puede hacerse que dependa de condiciones en el sistema en diversas maneras. Se define un tiempo de acción dando una media y modificador para el bloque. Si el modificador es un número positivo (\leq media), el tiempo de acción es una variable aleatoria entera que se elige de la media del rango \pm modificador, con iguales probabilidades dadas a cada número en el rango. A veces esta distribución uniforme es una representación exacta de un proceso aleatorio en el sistema, aunque el propósito principal de dar esta manera de representar un tiempo aleatorio es permitir casos en que se sabe que existe aleatoriedad pero que no se dispone de información detallada acerca de la distribución de probabilidad.

Es posible introducir una cantidad de funciones, o sea tablas de números que relacionan una variable de entrada con una variable de salida. Al especificar que el modificador de un bloque ADVANCE o GENERATE es una función, el valor de la función controla el tiempo de acción, el cual se obtiene multiplicando la media por el

valor de la función. Se pueden utilizar distintos tipos de entrada para las funciones, lo que permite que éstas introduzcan una diversidad de relaciones entre las variables de un sistema.

En especial, al hacer de la función una distribución de probabilidad acumulativa inversa, y al utilizar como entrada a un número aleatorio distribuido aleatoriamente, la función puede proporcionar una variable estocástica con una distribución determinada no uniforme en una manera descrita en el capítulo tres.

EL problema lleva registros de cuándo debe moverse cada transacción dentro del sistema. Avanza completando todos los movimientos programados para ejecutarse en un instante determinado y pueden realizarse lógicamente. Cuando hay más de una operación por moverse, el programa procesa las transacciones en el orden de su clase de prioridad, y en base de primero entrado, primero servido dentro de la prioridad.

Normalmente, una transacción no pasa tiempo en bloques distintos a los de ADVANCE. En consecuencia, una vez que el programa comienza a mover una transacción, continúa moviéndola a través del diagrama de bloques hasta que se plantea una de varias circunstancias.

La transacción puede entrar a un bloque ADVANCE con tiempo de acción no cero, en cuyo caso el programa vuelve su atención a otras transacciones del sistema y regresa a la anterior cuando haya transcurrido el tiempo de acción. En segundo lugar, las condiciones dentro del sistema pueden ser tales que no pueda realizarse en el momento actual la transacción que se intenta ejecutar al entrar a un bloque. Se dice que la transacción está bloqueada y permanece en el último bloque al que entró. EL programa detecta automáticamente cuando se ha eliminado la condición de bloqueo e inicia el movimiento de la transacción de nuevo en ese momento. Una tercera posibilidad es que la transacción entre a un bloque TERMINATE, en cuyo caso se saca de la simulación. La cuarta posibilidad es de que se ponga a la transacción en una cadena.

Cuando el programa ha avanzado una transacción lo más posible, vuelve su atención a cualesquier otra transacciones que deban de moverse en el mismo instante. Si están completos todos esos movimientos, el programa avanza el reloj a la hora del siguiente evento más inminente y repite el proceso de ejecutar los eventos.

Elección de caminos

EL bloque de **TRANSFER** permite elegir alguna localización distinta a la siguiente instrucción secuencial. Normalmente la selección se hace entre dos bloques conocidos como siguientes bloques A y B (también se utilizan los términos salida 1 y 2). El método que se utiliza para la Elección se indica mediante un factor de selección en el campo A del bloque de **TRANSFER**. Se puede prender para que indique una de nueve selecciones. Los siguientes bloques A y B se colocan en los campos B y C respectivamente.

Si no desea hacerse selección, se deja en blanco el factor de selección. Entonces se hace una transferencia incondicional al siguiente bloque A.

Se puede hacer una selección al azar haciendo que el factor de selección S sea una fracción decimal de tres dígitos. Entonces, la probabilidad de ir al siguiente bloque A es $1-S$, al siguiente bloque B es S. Un modo condicional, que se indica haciendo que el campo A sea **BOTH**, permite que una transacción elija otro camino dependiendo de condiciones existentes. La transacción va al siguiente bloque A si el movimiento es posible, y al siguiente bloque B si no lo es. Si ambos movimientos son imposibles, la transacción espera

al primero que sea factible, dando preferencia a A en caso de simultaneidad.

SIMULACION DE UN TALLER DE FABRICACION

Una máquina herramienta de un taller de fabricación produce piezas a razón de una por cada 5 minutos. Conforme se terminan, las piezas van a un inspector que necesita de 1 a 7 minutos o sea 4 ± 3 minutos para examinar cada una y rechaza aproximadamente 10% de las piezas. Cada pieza queda representada por una operación y la unidad de tiempo elegida para el problema será de 1 minuto.

Se utiliza un bloque **GENERATE** para representar la salida de la máquina creando una transacción cada cinco unidades de tiempo. Se utiliza un bloque **ADVANCE** con una media de 4 y modificador de 3 para representar la inspección. En consecuencia, el tiempo que transcurre en la inspección será cualquiera de los valores 1,2,3,4,5,6,7, dando igual probabilidad a cada valor. Al completar la inspección, las transacciones van a un bloque **TRANSFER** con un factor de selección de 0.1, de manera que 90% de las piezas van a la siguiente localización (salida 1) llamada **ACC** para representar las piezas aceptadas y 10 % van a otra localización (salida 2) llamada **REJ** para representar los rechazos. Ya que no se tiene interés adicional en seguir la

historia de las piezas en esta simulación ambas localizaciones a las que se llega desde el bloque de transfer son bloques de **TERMINATE**.

El problema se codifica, un * en la columna 1 se usa para establecer comentarios, . Un campo en las columnas 2 a 6 contienen la localización del bloque cuando es necesario especificarla. El programa de GPSS asigna automáticamente números secuenciales de localización al cargar el programa, de manera que por lo general no es necesario que el usuario asigne las localizaciones. Sin embargo el bloque **TRANSFER** necesita hacer referencia a los bloques de **TERMINATE** a los que envía las transacciones, de manera que a éstos se les dan los nombres simbólicos de localización **ACC** y **REJ**.

El segundo campo de la hoja de codificación, de las columnas 8 a 18, contiene el nombre del tipo de bloque que debe de comenzar en la columna 8. Partiendo de la columna 19 hay una serie de campos que pueden estar presentes, cada uno separado por comas, y sin espacios en blancos se trata como un comentario. El significado de los campos depende del tipo de bloque.

En el caso de los bloques **GENERATE** y **ADVANCE**, los campos A y B tienen respectivamente la media y modificador que controlan el tiempo de acción. El campo C del bloque **GENERATE** es un tiempo de corrimiento, (es decir, si no se

quiere que la primer transacción empiece en cero se debiera anotar el tiempo en que deberá empezar) que determina el momento en que se generará la hora de llegada de la primera transacción. Si es cero, la primera transacción se genera del tiempo igual a 1. La media de un bloque GENERATE puede ser cero, en cuyo caso el usuario debe de disponer que el flujo de las transacciones quede controlado por las condiciones del sistema. El campo D de un bloque GENERATE es un contador que determina el número total de transacciones que crea el bloque. Si es cero no hay límite. Cualquier campo no definido en el momento de encontrar el primer espacio en blanco se hace automáticamente igual a cero.

Para el Bloque TRANSFER, el primer campo es el factor de selección y los campos B y C son salidas 1 y 2 respectivamente. En este caso la salida 1 es el siguiente bloque secuencial, y sería permisible omitir el nombre ACC tanto en el campo B de TRANSFER como en el campo de localización del primer bloque TERMINATE. Sin embargo, es necesario poner ambas comas para demostrar que falta el campo B, de manera que la codificación del bloque de transferencia sería: TRANSFER 1, REJ. También debe de notarse que cuando se utiliza un bloque TRANSFER en un modo incondicional de manera que no se codificara el campo A, sigue necesitándose una coma para indicar el campo. En consecuencia una transferencia incondicional a REJ se codificaría como sigue, TRANSFER, REJ.

El programa corre hasta alcanzar determinada cuenta como resultado de que terminen las transacciones. El campo A del bloque **TERMINATE** tiene un número que indica en cuanto debe de incrementarse la cuenta de terminación cuando una transacción termina en ese bloque **TERMINATE** que tenga un campo A diferente de cero. En este caso, ambos bloques **TERMINATE** tienen 1, de manera que el contador de terminación suma el número total de transacciones que terminan; o sea, el número total de piezas buenas y malas inspeccionadas.

La instrucción **START**, en el campo A contiene el valor que debe de alcanzar el contador de terminación para hacer que termine la corrida de simulación en este caso 1000.

Primero se imprime el lote original de entrada, en que las localizaciones de las operaciones son asignadas por el programa se listan a la izquierda y un número secuencial de líneas a la derecha. Entonces se da una tabla de localizaciones simbólicas que muestra la localización asignada a cada símbolo. Luego sigue un listado del problema ensamblado con todas las referencias simbólicas reemplazadas por sus localizaciones.

FACILIDADES Y ALMACENAJES

Asociados con el sistema que se simula hay muchas entidades permanentes tales como artículos de equipo, que operan sobre las transacciones. En GPSS se definen dos tipos de entidades permanentes para representar el equipo del sistema.

Se define una facilidad como una entidad que puede utilizar una sola transacción a la vez. Se define un almacenaje como una entidad que pueden ocupar muchas transacciones a la vez hasta cierto límite predeterminado. Pueden ocupar muchas transacciones a la vez, hasta cierto límite predeterminado. Pueden haber mucha instancias de cada tipo de entidad hasta un límite impuesto por el programa (generalmente 300). Las entidades individuales se identifican mediante número; se utiliza una secuencia numeral distinta para cada tipo. El número 0 es ilegal para éstas y las otras entidades del GPSS. El usuario puede asignar los números en cualquier orden, o puede utilizar nombres simbólicos y dejar que el programa ensamblador asigne los números. Algunos ejemplos de cómo podrían interpretarse las entidades del sistema en distintos sistemas son:

| Tipo de sistema | Transacción | Facilidad | Almacenaje |
|-----------------|-------------|--------------|------------|
| Comunicaciones | Mensaje | Interruptor | Troncal |
| Transportación | Automóvil | Caseta Peaje | Carretera |
| Proc. de Datos | Registro | Lectura | Memoria |

Una troncal significa un cable que consiste de muchos alambres cada uno de los cuales puede transportar varios mensajes simultáneamente y por tanto se representa como un almacenaje. En este caso se supone que un interruptor sólo pasa un mensaje a la vez, por lo que se representa como facilidad.

Los bloques **SEIZE**, **RELEASE**, **ENTER** y **LEAVE**, que se refieren a la utilización de facilidades y almacenajes. El campo A de cada caso indica la facilidad o almacenaje de referencia, y generalmente la selección se marca en la bandera que se anexa a los símbolos de los bloques. El bloque **SEIZE** permite que una transacción utilice una facilidad si está disponible. El bloque **RELEASE** permite que la transacción libere la facilidad. El bloque **RELEASE** permite que la transacción libere la facilidad. En forma análoga, un bloque **ENTER** permite que una transacción ocupe espacio en un almacenaje, de estar disponible, y el bloque **LEAVE** permite que ceda el espacio. Si están en blanco los campos B de los bloques **ENTER** y **LEAVE**, el contenido del almacenaje se cambia en 1. Si hay un número (≤ 1), se cambia el contenido en ese valor. Se puede colocar cualquier cantidad de bloques entre los puntos en que se toma y libera

una facilidad para simular las acciones que se seguirán mientras una transacción tiene el control de una facilidad. Se aplican arreglos semejantes para la utilización de los almacenajes.

Utilizaremos estos bloques considerando el taller de fabricación. Ya que el tiempo promedio de inspección es de cuatro minutos y el ritmo promedio de generación de piezas es de una por cada 5 minutos, normalmente sólo se inspeccionará una pieza a la vez. Sin embargo, a veces puede llegar una nueva pieza antes de que se haya terminado la inspección de la anterior. Usando un bloque de ADVANCE para representar la inspección, esta condición hará que haya más de una transacción en el bloque de ADVANCE en un momento dado, por lo que de interpretarse la simulación con que se dispone de más de un inspector, con Esto utilizaremos varios inspectores.

Si hay solamente un inspector, lo representaremos, mediante una facilidad para simular el hecho de que sólo se puede inspeccionar una pieza a la vez. Se habrán agregado un bloque SEIZE y un bloque RELEASE para simular la ocupación y desocupación del inspector. Se da una línea de salida para cada facilidad, para mostrar cuántas veces tomó. la utilización promedio de la misma y el tiempo promedio que le retuvieron las transacciones. En este caso, los resultados

muestran que el inspector se mantuvo ocupado 74.9% de su tiempo.

Si hay más de un inspector disponible, se les puede representar como un grupo mediante un almacenaje con capacidad igual al número de inspectores. Entonces se sustituye a los bloques SEIZE y RELEASE mediante bloques ENTER Y LEAVE respectivamente, por ejemplo, suponga que el tiempo de inspección fuera tres veces mayor que antes; se justificarían tres inspectores. Note que se ha agregado una instrucción de control llamada STORAGE en el campo de localización, este identifica el almacenaje y el campo A contiene la capacidad del mismo ($\leq 2,147'483,647$). Para cada almacenaje aparece una línea de salida, que da la información indicada en los encabezados. Por ejemplo muestra que en un promedio 2.275 de los inspectores estuvieron ocupados en todo momento.

Se podría haber modelado el caso de un solo inspector, utilizando un almacenaje con capacidad de 1 en vez de facilidad. Una diferencia lógica entre estas dos representaciones posibles es que sólo la transacción que tomó una facilidad la puede liberar, en tanto que para un almacenaje, la entrada salida pueden ser acciones separadas que realicen independientemente distintas transacciones. No se utilizó esta propiedad en estos modelos, aunque la diferencia es de importancia en muchos.

RECOLECCION DE ESTADISTICAS

Determinados tipos de bloques del GPSS se construyen para recabar estadísticas relativas al comportamiento del sistema, más que para representar las acciones del mismo. Los bloques **QUEUE**, **DEPART**, **MARK** y **TABULATE**, sirven para este propósito. Introducen otras dos entidades del programa GPSS las colas y tablas como las facilidades y almacenajes, pueden haber muchas de esas entidades hasta determinado límite (por lo general, 300 para las colas y 100 para las tablas) y se identifican individualmente mediante un número o nombre simbólico.

Cuando no se satisfacen las condiciones para avanzar una transacción, se puede mantener a varias transacciones esperando en un bloque; el programa las mantiene en orden y cuando las condiciones son favorables se les permite avanzar de acuerdo con la prioridad y generalmente según una regla de primero entrado primero salido. Sin embargo, no se recaba información a la cola de transacciones a menos que hayan entrado a una entidad de cola. El bloque **QUEUE** (cola) aumenta y el bloque **DEPART** (salida) disminuye la cola número A. Si el campo B está en blanco, el cambio es unitario; en caso contrario se utiliza el valor del campo B (≤ 1). El programa mide las longitudes promedio y máxima de las colas, y de requerirse, la distribución del tiempo transcurrido en la cola.

Los bloques **MARK** y **TABULATE** miden el tiempo que toman las transacciones para atravesar el sistema o partes del mismo. Cada uno de estos tipos de bloque toma nota de la hora al que lleva una transacción al bloque. El bloque **MARK** sencillamente indica la hora de llegada en la transacción. (Si el campo A esta en blanco se utiliza una palabra especial. Con n en el campo A, se utiliza el nésimo parámetro.) El bloque **TABULATE** resta la hora anotada por un bloque **MARK** de la hora de llegada en el bloque **TABULATE**. El tiempo, al que se conoce como tiempo de tránsito, se entra en una tabla cuyo número o nombre se indica en el campo A del bloque **TABULATE**. Si la transacción que entra a un bloque **TABULATE** no ha pasado a través de un bloque **MARK**, se deduce su tiempo de tránsito utilizando como base la hora en que se creó la transacción. En efecto, se puede considerar al tiempo de tránsito a cero.

Para ilustrar el uso de estos bloques, volvemos al modelo de un taller de fabricación con tres inspectores. Las propiedades del bloque **GENERATE** son tales que si una transmisión puede salir del bloque cuando se crea, no se realizan creaciones adicionales hasta despejar el bloque. Si se ha terminado una pieza mientras se está inspeccionando la anterior, se detiene el maquinado de piezas adicionales hasta despejar la máquina.

Un caso más realista es que las piezas se acumulen en el banco de trabajo del inspector si la operación no termina con suficiente rapidez. En ese caso interesa medir el tamaño de la cola del trabajo que se forma. Se coloca un bloque de QUEUE con el número 1 de cola inmediatamente antes del bloque ENTER y se coloca un bloque DEPART inmediatamente después del bloque ENTER para quitar la pieza de la cola cuando se inicia la inspección. Toda operación que no tenga que esperar a que se libere un inspector se mueve a través de la cola sin retraso. Las que deben de esperar lo harán en el bloque de QUEUE, y el programa mide automáticamente la duración de estancia en la cola.

La instrucción TABLE, en el campo de localización identifica a una tabla; y en el campo A indica la cantidad que debe tabularse. En los campos C y D de la instrucción TABLE están el límite inferior de la tabla, el tamaño del intervalo de tabulación y el número de intervalos respectivamente. El símbolo MI en el campo A de la instrucción TABLE indica Tabulación del tiempo de transito.

En la figura se verá que ocurre una línea de salida para cada cola, y que se imprime una tabla de acuerdo con las especificaciones de la instrucción TABLE. Se pueden utilizar los bloque TABULATE para recabar diversidad de otros datos estadísticos.

TRANSFERENCIAS CONDICIONALES

El bloque **TRANSFER** permite la transferencia tanto condicional como incondicional. De nuevo se consideran tres inspectores, pero suponga que las piezas fabricadas se transportan en una banda transportadora, la que lleva las piezas al frente a los inspectores colocados a intervalos a lo largo de la misma banda. Una pieza tarda dos minutos en llegar al primer inspector, si está libre cuando llega esa pieza, la toma para inspeccionarla; si está ocupado, la pieza necesita dos minutos adicionales para llegar al segundo inspector, que la toma si esta desocupado, Las piezas que pasan al segundo inspector puede tomarlas el tercero, que está a dos minutos adicionales sobre la banda transportadora; en caso contrario se pierden. Para mantener pequeño el modelo, se registrará todo el tiempo de tránsito de las piezas y se ignorará la posibilidad de que los inspectores rechacen las mismas.

En la figura se muestra un diagrama del bloque del sistema. El movimiento de las piezas a lo largo de la banda se representa mediante bloque **ADVANCE**, cada uno con tiempo de acción de dos minutos. Cuando una transacción sale de un bloque **ADVANCE**, determina si está disponible algún inspector entrando a un bloque **TRANSFER** con un factor de selección prendido en **BOTH**. La salida 1 de cada uno de estos bloques

TRANSFER lleva a un bloque **SEIZE** que representa a cada uno de los inspectores. Si el inspector está desocupado cuando entra la transacción pasa a la siguiente etapa del procesamiento a través de la salida 2.

Cuando las piezas salen de inspección, las transacciones van a un solo bloque **TABULATE** donde se registra el tiempo de tránsito. Debido a la regla mediante la cual las transacciones pasan normalmente de un bloque al bloque con el siguiente número superior, sólo uno de los tres bloques **RELEASE** que completan la fase de inspección puede pasar las transacciones directamente al bloque **TABULATE**. Los otros pasan las transacciones al bloque **TRANSFER** que las envían incondicionalmente al bloque **TABULATE**. Sería posible tener tres bloques **TABULATE** por separado en la salida de cada bloque **RELEASE**, lo que haría innecesarios los bloques de **TRANSFER INCONDICIONAL**. Cada bloque **TABULATE** se puede referir al mismo número de tabla, de manera que las dos maneras alternativas de codificar el diagrama de bloques conducen a las mismas estadísticas. Este principio que permite referirse a la misma entidad de **GPST** en más de un lugar se aplica a todas las entidades.

INSTRUCCIONES DE CONTROL DEL PROGRAMA

La primera instrucción de un programa **GPSS** es **SIMULATE** en el campo de operación. Sin esta instrucción el programa se compila pero no se ejecuta.

Es deseable poder detener y arrancar de nuevo una corrida de simulación y también repetir una corrida de simulación realizando cambios a algunos valores del modelo. En consecuencia, cuando termina una corrida de simulación en GPSS, el programa no destruye de inmediato el modelo utilizado, sino que espera más instrucciones después de la instrucción **START**, manteniendo el modelo exactamente como estaba a la terminación de la corrida. Las instrucciones que a **START** pueden cambiar el modelo. Por ejemplo se redefiniría una capacidad insertando la instrucción **STORAGE** que se refiere al número de un almacenaje definido anteriormente y que da el nuevo valor. También podría modificarse el modelo cambiando los bloques existentes y agregando otros nuevos. Luego que se han leído todas las modificaciones y el modelo está listo para correr otra instrucción éstas indican al programa que comience de nuevo la simulación.

Es posible incluir determinadas instrucciones en **START** para fijar las condiciones bajo las que se hace la nueva corrida. La instrucción **RESET** en el campo de operación borra todas las estadísticas reunidas, pero deja al sistema cargado con las transacciones. El propósito es recabar estadísticas en la segunda corrida excluyendo el periodo inicial de acumulación. Entonces no interesa la salida de la primera corrida y por lo general se suprime escribiendo las letras NP en el campo B de la instrucción **START**.

La instrucción **RESET** también pone en cero el reloj relativo, cuando se completa la corrida, la salida marcada **ABSOLUTE CLOCK** da el tiempo desde que inició la corrida la salida marcada **RELATIVE CLOCK** da el tiempo desde la última instrucción **RESET**. Si no se emplea una instrucción **RESET** los tiempos son iguales.

La instrucción **CLEAR**, borra las estadísticas de la corrida anterior y también las transacciones del sistema; en ese caso la nueva corrida comienza la simulación desde el principio. Se sigue este procedimiento cuando el objetivo es repetir el problema haciendo un cambio a algún valor. Aunque la instrucción **CLEAR** regresa el modelo a su estado inicial, no restaura la semilla generador de número aleatorios. La siguiente secuencia de instrucciones corre el mismo problema dos veces:

```
START  
CLEAR  
S.ART
```

Pero la segunda corrida utilizaría un conjunto distinto de número aleatorios. Cuando hay un proceso estocástico en el modelo, la segunda corrida será una segunda muestra de las salidas posibles que pueden plantearse.

Además del control de distintas corridas del mismo modelo, el GPSS permite que corra cualquier cantidad de distintos problemas sucesivamente con una sola carga del problema. Para ello hay una instrucción JOB en el campo de operaciones que indica al programa que borre todo el modelo anterior a la instrucción y pasar al siguiente problema, lo que puede repetirse tantas veces como se desea. Esta instrucción sí restaura los números aleatorios.

Ya que el programa anticipa entradas adicionales al terminar una corrida, es necesario colocar una tarjeta de control con la instrucción END en el campo de operaciones al final de todos los problemas, aunque sea sólo uno, para terminar toda la simulación. La instrucción SIMULATE sólo aparece una vez, al principio del lote de entrada, aunque haya trabajos múltiples.

PRIORIDADES Y PARAMETROS

Las transacciones no tienen identificación específica; los bloques consideran a cada una en la misma manera como a cualquier otra. En realidad, las transacciones tienen dos tipos de atributos que influyen en la manera en como se procesan. Cada una tiene uno de 128 niveles de prioridad, que se indican mediante los números 0 a 127, en que el 0 tiene la menor prioridad. El bloque se codifica escribiendo la prioridad en el campo A del bloque. También es posible designar la prioridad al tiempo en que se crea una

transacción escribiéndola en el campo E del bloque **GENERATE** que genera la transacción. Si se deja el campo en blanco se hace que la prioridad sea 0.

Cuando se compite entre las transacciones para ocupar un bloque, la regla de servicio es avanzar las transacciones en orden de prioridad de primero entrado primero salido dentro de una clase de prioridad. El programa tiene la capacidad de implementar disciplinas más complejas de colas mediante la característica de cadena.

Una transacción también puede tener parámetros que son números enteros con signo. El número de parámetros que se le asigna a una transacción puede ir desde 0 a 100, y se especifica en el campo F del bloque **GENERATE** que crea la transacción. Si el campo queda en blanco automáticamente se asignarán 12 parámetros.

Todos los valores son cero cuando se crea una transacción. Se da un valor a un parámetro cuando una transacción entra al bloque **ASSIGN**. Este puede utilizar como fuente del valor por asignar a cualquiera de los atributos numéricos estándar (SNA). Se codifica indicando en el campo A y en el campo B el número de parámetro y el SNA que se utilizará. Un bloque **ASSIGN** puede sumar a, restar de o sustituir el valor de un parámetro en el campo A indica que el valor asignado se debe de sumar o restar. Para hacer la sustitución, sólo se da el número de parámetro.

Generalmente, los parámetros registran las características de la entidad representada por la transacción. Por ejemplo, una transacción que representa un mensaje en un sistema de comunicaciones puede utilizar parámetros para representar la longitud del mensaje, su tipo o destino. Una transacción que represente un barco puede utilizar parámetros para representar la velocidad de crucero, número de pasajeros o tipo de carga. Otro uso de los parámetros es el control lógico del modelo. Por ejemplo, se puede controlar una transacción por el número de veces que debe realizarse una transacción, en donde los parámetros transmitan el número de veces que se haya realizado la operación hasta el momento y el número total requerido.

Atributos numéricos estándar

Los primeros son artículos de datos que representan atributos de la transacción. Adicionalmente, hay atributos de las otras entidades del sistema, tal como el número de transacciones en un almacenaje o la longitud de una cola, que quedan disponibles al usuario del programa. En forma colectiva se les llama atributos numérico estándar (SNA). Cada tipo de SNA se identifica mediante una clave de una o dos letras y un número. Por ejemplo, el contenido del almacenaje número 5 se denota mediante S5 y la longitud de la cola número 5 se denota mediante Q5 y la longitud de la cola número 15 es Q15. Por ser completos, los parámetros se

incluyen dentro de la categoría de los SNA, y se identifican mediante el símbolo P_n en que n es el número del parámetro.

Es posible desarrollar cómputos definiendo declaraciones variables en que se cambian matemáticamente SNAs simples con los operadores $+$, $-$, $*$, $/$ y δ para la adición, sustracción, multiplicación, división y división de módulo. Se pueden incluir hasta cinco niveles en la definición de una declaración variable. Una declaración variable se número y define mediante una operación que tiene el número o nombre en el campo de la localización y la palabra VARIABLE en el campo de operaciones. La declaración comienza en la palabra 19 y continúa sin blancos hasta la columna 71 de ser necesario. Por ejemplo la siguiente declaración:

```
5 VARIABLE S6+5*(Q12+Q17)
```

Define a la variable 5 como la suma del contenido actual del almacenaje número 6 más 5 multiplicado por la suma de las longitudes de las colas 12 y 17. Las declaraciones variables se incluyen en la categoría de los SNAs y se denotan por el símbolo V_n de manera que una declaración variable puede ser componente de otra.

Se puede dar muchos usos a los SNAs; proporcionan las entradas a las funciones descritas en la siguiente sección, permitiendo con ello introducir al modelo una gran diversidad de relaciones funcionales. Se recordará que la instrucción TABLE utilizada en combinación con el bloque

TABULATE para medir el tiempo de tránsito lleva M1 en el campo A. El tiempo de tránsito es uno de los **SNAs** y M1 es su símbolo; se puede utilizar cualquier otro **SNA** en una instrucción **TABLE** lo que permite al programa recabar y tabular una diversidad extensa de estadísticas. También se puede utilizar el valor actual de cualquier **SNA** como el valor de casi cualquier campo de un bloque. Las excepciones son los campos de factor de selección del bloque **TRANSFER**.

En general, los valores de los **SNAs** se deben emplear cuando se necesitan. Es conveniente poder reservar los valores calculados en algún momento para utilizarse en otro momento posterior, lo que puede hacerse utilizando la instrucción **SAVEVALUE**. El campo indica en el campo A el número de muchas localizaciones de reservas de valores, y en el campo B da el **SNA** que debe reservarse. Como con una instrucción **ASSIGN**, un signo + o - inmediatamente después del número del **SAVEVALUE** hace que el **SNA** se sume o reste del contenido del valor reservado; en caso contrario, se sustituye el valor. El contenido actual de una localización n de reserva de valor queda disponible como una **SNA** indicando mediante Xn. Una instrucción de control llamada **INITIAL** puede fijar el valor de una reserva de valor al principio de una simulación de manera que las reservas de valores pueden introducir condiciones iniciales. La notación Xn se reserva en el campo A y el valor imicial se escribe en el campo B. Los valores de reserva, de no cero se imprimen

al final de la corrida de simulación de manera que éstos también proporcionan una manera de extraer resultados de la simulación aparte del informe estándar.

Tabla de los Atributos numéricos estándar del GPSS

| | |
|------------|--|
| C1 | Valor actual del tiempo de reloj. |
| CHn | Número de transacciones en la cadena n. |
| Fn | Estado actual de la facilidad número n. Esta variable es 1 si la facilidad está ocupada y 0 si no. |
| Kn | El entero n. (también puede utilizarse el entero n.) |
| M1 | El tiempo de tránsito de una transacción. |
| Nn | El número total de transacciones que han entrado al bloque n. |
| Pn | Número n de parámetro de una transacción. |
| Qn | La longitud de la cola n. |
| Rn | El espacio que queda en el almacenaje n. |
| RNn | Un número aleatorio computado que tiene uno de los valores 1 al 999 con igual probabilidad. (Cuando se hace referencia para dar la entrada a una función, se reduce automáticamente la escala del valor al rango 0 a 1.) Se puede referenciar ocho generadores distintos mediante $n = 1, 2, \dots, 8$. |
| Sn | La ocupación actual del almacenaje n. |
| Vn | El valor de un número de transacciones que están actualmente en el bloque n. |
| Xn | El valor de la localización n de reserva de valor. |

FUNCIONES

Para introducir las relaciones funcionales en un modelo, GPSS puede incluir una diversidad de funciones. cada función se define dando dos o más parejas de números que relacionan una entrada x con una salida y . La función puede estar en un modo continuo o discreto. En el primer caso, el programa interpola linealmente entre los puntos definidos, permitiendo al usuario aproximar una función continua mediante una serie de segmentos de líneas rectas. En un modo discontinuo, se considera a la función como una "función escalera". Si x_i y x_{i+1} son dos puntos sucesivos en que se ha definido la función, un valor $y + i$.

Se puede utilizar cualquier cantidad de puntos (>1) para definir una función; también puede variar con libertad el número de intervalos entre puntos sucesivos excepto porque una función definida en un número aleatorio como entrada debe de tener valores de x en el rango de 0 a 1. Los valores de " x " y " y " pueden ser fraccionarios y negativos. Una referencia a un valor de x por debajo del valor definido más bajo, x_1 , produce el valor de " y " en ese valor más alto.

Al utilizar una función, se puede definir cualquiera de los SNAs como la entrada. Ejemplos de algunas de las selecciones utilizadas más comúnmente de entrada y sus usos son:

a) Utilizando como una entrada un número aleatorio R_N distribuido uniformemente como entrada, se puede obtener cualquier otra distribución.

b) Utilizando el tiempo del reloj C₁, se puede hacer que los tiempos de acción dependan del tiempo, simulando con ello los efectos de las cargas pico.

c) Utilizando el contenido actual de un almacenaje, Un tiempo de acción puede depender de la carga actual en alguna parte del sistema.

d) Utilizando un parámetro P_n, un tiempo de acción puede depender de cada transacción específica.

La selección de los datos de entrada se hace al definir la función, no se eligen cuando se referencia la función. Se necesita definir la función y le siguen los datos de la función. La instrucción es con el siguiente formato:

Campo

Localización Número de función
 Operación FUNCTION
 SNA a utilizar como entrada

Cn para modos continuo

Dn para modo discontinuo ó discreto

en que n es el núm. de puntos a definir.

Los valores se digitan partiendo de la columna 1 insertando comas entre los valores "x" y "y", y diagonales entre párejas sucesivas: $x_1, y_1 / x_2, y_2 / \dots$. Puede haber cualquier dígito en la instrucción en tanto no pase de la columna 71. Se puede utilizar cualquier cantidad de líneas adicionales y no es posible dividir una pareja de valores "x,y" entre dos líneas. La siguiente función muestra la codificación para la aproximación de la función continua que se dio anteriormente.

$$y = \text{Log}_e(1 - x)$$

La compilación para una función discontinua vía la siguiente especificación:

| X | Y |
|------------------|---|
| $0 < x \leq 0.2$ | 5 |

| | |
|-------------|----|
| 0.2 <x≤ 0.5 | 10 |
| 0.5 <x≤ 0.9 | 15 |
| 0.9 <x≤ 1.0 | 20 |

MODOS DE TRANSFERENCIA

El bloque TRANSFER permite hacer uso de parámetros en combinación con las funciones. Más que sólo hacer una selección aleatoria o una selección condicional entre dos bloques, un bloque TRANSFER puede utilizar el valor de un parámetro o una función como la localización a la que envía una transacción. Para utilizar el modo de parámetro, se escribe una P en el campo A del bloque TRANSFER y en el campo B se escribe el número del parámetro. Para el modo de función se escribe FN en el campo A y el número de función en el campo B. Si se escribe un número de parámetro en el campo C, se agrega el valor de parámetro al de la función. Ambos modos de operación son especialmente útiles cuando deben de manejarse distintas categorías de transacciones en la misma manera en una parte del diagrama de bloques, pero que a su debido tiempo deban de separarse.

Pueden emplearse funciones continuas o discretas que utilizan cualquier SNA como entrada, aunque se ha definido un modo de terminado de función para simplificar el uso de funciones para las transferencias. Una función de modo de lista supone que la entrada es un entero n y devuelve el n-ésimo valor listado de la función. Ya que no siempre se

conocen los números de localización antes de la compilación, es permisible utilizar nombres de localización como valores de función; el programa ensamblador suministra el valor numérico correcto. Por ejemplo, suponga que se desea enviar transacciones a una de cuatro localizaciones llamadas LOCA, LOCB, LOCC, y LOCD, y suponga que se ha colocado uno de los números 1,2,3 ó 4 en un parámetro, por ejemplo el número 3. La siguiente codificación realiza la transferencia:

```
TRANSFER FN,1
```

```
.....
```

```
1 FUNCTION P3,L4
```

```
1,LOCA/2,LOCB/3,LOCC/4,LOCD
```

Los caracteres L4 indican que la función está en el modo de lista y tiene cuatro valores listados.

De otra manera, un bloque ASSIGN podría utilizar esta función para hacer que un parámetro, por ejemplo el número 1, sea la localización. Más adelante podría lograr la transferencia el bloque.

```
TRANSFER P,1
```

La siguiente tabla indica los modos de transferencia.

Modos de transferencia del GPSS.

| Modo | Campo A | Campo B | Campo C |
|---------------|---------|----------------|---------------|
| Incondicional | | Sig. bloque A | |
| Aleatorio | | Sig. bloque A | Sig. bloque B |
| Condicional | | Sig. bloque A | Sig. bloque B |
| Parámetro | | Núm. Parámetro | |
| Función | | Núm. Función. | |

SIMULACION DE UN SUPERMERCADO

Opera la simulación como sigue: Los clientes necesitan tomar una cesta, como hay un número limitado de cestas, salen sin hacer compra si no hay. Si obtienen una cesta, hacen su compra y luego pagan pasando por una de las cinco cajas. Después de pagar, devuelven las cestas y salen del supermercado. El problema se puede dividir en cuatro partes:

- a) Tomar una cesta
- b) Comprar
- c) Pagar
- d) Salir

Cada cliente está representado por una transacción y la unidad de tiempo es 1 segundo.

Un bloque GENERATE crea las transacciones que representan a los clientes. Se supone que se puede representar el patrón de llegadas mediante una distribución de Poisson. El proceso para generar dichos números requiere la función $y = \text{Log}_e(1-x)$ que se codifica como la función número 1. La regla para utilizar las funciones del bloque GENERATE es que la media multiplica a la función. En este caso se supone un tiempo medio entre llegadas de 36 segundos.

Para representar las cestas se utiliza un almacenaje denotado por BSKT con capacidad igual al número de cestas; en este caso habrán 50. La decisión de si un cliente puede obtener una cesta se toma en un bloque TRANSFER inmediatamente después del bloque GENERATE. El bloque TRANSFER tiene un factor de selección BOTH e intenta pasar las transacciones a un bloque ENTER utilizando el almacenaje BSKT. Si se dispone de una cesta, el bloque ENTER acepta la transacción y aumenta en 1 el contador de cestas en uso. Sin embargo, si el almacenaje está lleno, no se dispone de cestas y el bloque TRANSFER pasa la transacción a un bloque TERMINATE denominado AWAY que cuenta a los clientes rechazados por falta de cestas.

La codificación de esta sección es:

| | |
|----------|------------|
| GENERATE | 36, FN1 |
| TRANSFER | BOTH, AWAY |
| | |
| ENTER | BSKT |
| AWAY | TERMINATE |

Se prepara la simulación para que el tiempo de compra dependa del número de artículos comprados. Se asigna un parámetro, el número 1, para representar el número de artículos en compra. Este número se determina en un bloque ASSIGN que utiliza la distribución discreta del número 2 de función. Siguiendo la técnica descrita en la capítulo 3, una entrada de números aleatorios a la función hace que el número de artículos sean 5, 15 ó 20 con probabilidades de 0.2, 0.03, 0.4 y 0.1 respectivamente. Las transacciones pasan entonces a un bloque ADVANCE para representar las compras. Entre el número de artículos y el tiempo de compra. Se supone la relación funcional de la sig. figura. y codificada como la función número 3. Como entrada, la función tiene el parámetro número 1. El programa evalúa la función con el parámetro 1 de cada transacción que entra al bloque ADVANCE. La media en el bloque ADVANCE se hace igual a 1 de manera que el valor de la función se aplique directamente como el tiempo de acción. La codificación para la sección de compras como sigue:

| | |
|---------|--------|
| ADVANCE | 1, FN3 |
| ASSIGN | 1, FN2 |

Cuando las transacciones salen del bloque ADVANCE, se ha completado la compra y se pasan a la sección que se refiere al pago. Hay cinco cajas pero en este estudio no es necesario distinguir el funcionamiento de cada uno, por lo que se les considera como unidad de servicio de 5 dependientes, representados por un almacenaje de nombre CKT con capacidad de cinco.

Habrá cierta congestión en las cajas, y uno de los objetivos será medir el volumen de congestión. En consecuencia, las transacciones van a un bloque de colas que las forma en una cola llamada WAIT. Cuando se desocupa una caja, la transacción sale del bloque QUEUE, hacia el bloque ENTER e inmediatamente va a un bloque DEPART para salir de la cola. De no haber congestión, la transacción se mueve directamente a través de los bloques de QUEUE, ENTER y DEPART.

Se supondrá que la función de pago requiere de diez segundos por artículo más 25 segundos para el empaclado. Ya que el parámetro 1 es la cantidad de artículos, la siguiente declaración variable calcula el tiempo de pago:

| | | |
|---|----------|----------|
| 1 | VARIABLE | P1*10+25 |
|---|----------|----------|

Como mencionamos antes, es posible colocar una SNA en la mayoría de los campos de un bloque. En ese caso el campo A de un bloque ADVANCE se hace igual a VI. Conforme cada transacción entra al bloque, el programa calcula el tiempo de acción para la declaración van a un bloque LEAVE, para desocupar el espacio ante la caja. La codificación para la sección de pago es:

| | |
|---------|------|
| QUEUE | WAIT |
| ENTER | CKT |
| DEPART | WAIT |
| ADVANCE | VI |
| LEAVE | CKT |

Al completar el pago, las transacciones pasan a la sección que se refiere a salir del supermercado. Primero van a un bloque TABULATE donde se tabula el tiempo de tránsito en una tabla llamada TRT. Suponga que se desca llevar un registro del número de artículos que compra cada cliente, lo que realiza yendo a otro bloque TABULATE que tabula PI en una tabla denominada ITM. Desde luego, la tabulación reproducirá la distribución original de la función 2. Se insertó este paso para ilustrar el uso de un bloque TABULATE para fines estadísticos distintos al tiempo de tránsito.

Cuando se completa la tabulación, las transacciones van a un bloque LEAVE, que nombra el almacenaje BSKT, para

representar la devolución de la cesta. Finalmente van a un bloque TERMINATE. La codificación es la siguiente:

| | | |
|-----|-----------|---------------|
| | TABULATE | TRT |
| | TABULATE | ITM |
| | LEAVE | BSKT |
| | TERMINATE | 1 |
| TRT | TABLE | M1,500,500,10 |
| ITM | TABLE | P1,5,5,5 |

Note que en el campo A de este bloque TERMINATE aparece un 1 en tanto que en el campo A del bloque TERMINATE está en blanco. En consecuencia, la corrida de simulación solo cuenta los clientes atendidos.

Interruptores Lógicos

Se han introducido dos tipos de entidades, facilidades y almacenajes, para representar equipo. Adicionalmente, se pone a disposición un tercer tipo de entidad conocida como interruptores (switchs) lógicos para representar condiciones de dos estados en un sistema. Cada interruptor está encendido o apagado, y se utiliza un tipo de bloque llamado LOGIC para cambiar su estado. Una transacción que entra al bloque puede prender el interruptor ya está en el estado deseado, no se sigue ninguna acción. Al codificar el bloque LOGIC, en la columna 14 se escribe la letra S,R o I que indica prender, apagar o invertir en el campo A donde se da

el número o nombre del interruptor. El programa no lleva estadística de los interruptores lógicos. Sin embargo, imprime los números de los que estaban prendidos al terminar la corrida.

Algunos ejemplos de cómo se utilizan los interruptores lógicos son: un modelo que representa una fábrica podría utilizar un interruptor lógico para indicar si una máquina está en condición de trabajo o no; en un sistema de tráfico automotriz, un interruptor lógico puede representar un semáforo; y un banco, un interruptor lógico puede representar si una ventanilla está abierta.

Prueba de Condiciones

Con frecuencia es deseable controlar el flujo de las transacciones de acuerdo con condiciones que prevalecen en el sistema. Para este propósito se puede utilizar un tipo de bloque denominado GATE, que puede probar la condición de cualquier facilidad, almacenaje o interruptor lógico en el diagrama de bloques. Las condiciones que pueden probarse y los símbolos utilizados para indicar las condiciones seleccionadas son:

| | | |
|-----|---|-------------------------------|
| LS | n | Interruptor lógico n prendido |
| LR | n | Interruptor lógico n apagado |
| U | n | Facilidad n en uso |
| NU | n | Facilidad n no en uso |
| SF | n | Almacenaje n lleno |
| SNF | n | Almacenaje n no lleno |
| SE | n | Almacenaje n vacío |
| SNE | n | Almacenaje n no vacío |

Una transacción entra al bloque de GATE si la condición probada es verdadera. Cuando no es verdadera, hay una selección de acción. Si se especifica un bloque alternativo, hay una selección de acción. Si no hay alternativa, la transacción espera hasta que la condición probada sea verdadera y luego entra al bloque GATE. No es necesario que el usuario disponga la prueba de nuevo; el programa reconoce automáticamente cuando cambia la condición y mueve la transacción en ese momento. Al codificar el bloque GATE, el código de condición comienza en la columna 13, en tanto que el número o nombre de la entidad a probar se coloca en el campo A. Si se especifica un bloque alterno, se escribe en el campo B.

Otro tipo de bloque, llamado TEST, puede probar una diversidad de relaciones entre dos SNAs cualesquiera. Ya que éstos incluyen declaraciones variables, el bloque TEST puede realizar pruebas complejas de condiciones. En seguida se muestran las realizaciones que pueden probarse y los símbolos que se utilizan para representarlas:

| | |
|----|-------------------|
| G | Mayor que |
| GE | Mayor que o igual |
| E | Igual |
| NE | No igual |
| L | Menor que |
| LE | Menor que o igual |

El símbolo de relación comienza en la columna 13 y los dos SNAs por relacionar se colocan en los campos A y B. En consecuencia, el siguiente bloque prueba si el contenido del

almacenaje 6 es menor que el número registrado en la localización 12 de reserva de valor.

TEST L S6,X12

La acción que se sigue para probar la condición es la misma que para un bloque GATE. Una transacción entra al bloque si la condición es verdadera y va a un bloque alternativo si se especifica (en el campo C) o espera a entrar cuando cambia la condición.

Direccionamiento Indirecto

Cuando se hace referencia a un campo o SNA, se debe de dar un número específico al programa. Generalmente se proporciona el número cuando se codifican los bloques. También se le puede dar el Direccionamiento indirecto que es: sin especificar el bloque en ese momento y preparar el programa para que tome el valor de un parámetro en la transacción que entra a un bloque o que invoca el uso del SNA. La notación *n en un campo de bloque indica que se debe de utilizar un parámetro de número n en ese campo; ya que se puede utilizar un SNA en un campo de bloque, la notación Pn lograría exactamente el mismo efecto pero la notación *n se ejecuta con mayor rapidez. De esa manera en el ejemplo del supermercado, la declaración variable que controla el tiempo de pago se escribió en la forma:

1 VARIABLE P1*10+25

pero también se pudo escribir como

```
1          VARIABLE          *1*10+25
```

Al evaluar esta última forma de la declaración de variable, el programa sustituye *1 por el valor del parámetro número 1.

El direccionamiento indirecto puede reducir considerablemente el tamaño de un diagrama de bloques, en los casos donde hay secciones repetitivas que sólo difieren en el número de las entidades que se utilizan en las secciones. Se puede utilizar una sección común, haciendo que los parámetros de las transacciones suministren los valores específicos. Como ejemplo, suponga que en el problema del supermercado a los clientes que compran cinco o menos artículos se les permite utilizar una sección especial de dos cajas rápidas. Se define una nueva cola y almacenaje para representar esta área rápida. Los números de la cola y almacenaje para los compradores normales serán 1 y la nueva cola y almacenaje serán el número 2.

Las transacciones van a un bloque TEST prueba parámetro número 1, la cantidad de artículos de compra, es mayor que 5. En caso afirmativo, la transacción va a un bloque ASSIGN que pone el número de parámetro 2 igual a 1. De lo contrario, pone el parámetro 2 igual a 2. Los otros bloques de la sección se ponen en el campo A el dato *2. En cada caso las transacciones de entrada hacen que los bloques

operen con el número de entidades indicado por el valor del número de parámetro 2, de manera que los compradores normales se van a la cola 1 y almacenaje 1 y los otros a la cola 2 y almacenaje 2.

Suponiendo que deben de utilizarse distintos tiempos de pago para los dos casos, se define una segunda declaración de variable, la número 2, para el tiempo de pago rápido y se hace que el bloque ADVANCE que representa el tiempo de pago tenga $V*2$ en el campo A, lo que hace que el bloque elija la declaración variable 1 ó 2 de acuerdo con el valor del parámetro 2.

Modelo de GPSS de un sistema telefónico simple

Cada llamada queda representada por una transacción; la unidad de tiempo elegida es 1 segundo. Se supondrá que la distribución de llamadas es de Poisson con un tiempo medio entre llegadas de una longitud exponencial. Cada nueva llamada puede venir de cualesquiera de las líneas no ocupadas con igual probabilidad, y que su destino tiene igual probabilidad de ser cualquier línea distinta a sí misma.

Se utiliza un bloque GENEATE para crear una serie de transacciones que representen las llamadas, se pone de modificador una función. Se hace que la media sea 12 y se utilizan los parámetros para llevar el origen y destino de la llamada, se pasa cada transacción a dos bloques ASSIGN

par elegir y registrar los valores. La fuente de la información es una declaración VARIABLE, la número 1 utilizando un número al azar. El número de líneas N se multiplica por un número aleatorio entre 0 y 1, y se considera que la parte entera representa un entidad de GPSS, de manera que se suma 1 al resultado para que la acción para extraer la parte entera debido a que (con determinadas excepciones) el GPSS trabaja con números enteros. Toda evaluación de una declaración VARIABLE o función que produzca un número fraccionario se convierte a un entero truncado la parte fraccionaria. El número de líneas varía en distintas corridas de manera que el número deseado de líneas se reserva en el campo de reserva de valor número 1. Una instrucción INITIAL carga el valor deseado al principio del programa.

La siguiente codificación coloca los números de línea elegidos al azar de cincuenta líneas en los parámetro número 1 y 2:

| | | |
|---|----------|---------------|
| | ASSIGN | 1,V1 |
| | ASSIGN | 2,V1 |
| 1 | VARIABLE | X1*RN1/1000+1 |
| | INITIAL | X1,50 |

Se puede utilizar la misma declaración de variable para ambas asignaciones debido a que cada referencia a la variable produce un número aleatorio distinto.

Con este método de generar el origen y destino de las llamadas posibles que el origen de la llamada ya esté ocupado. Utilizando el direccionamiento indirecto, un bloque de GATE si el origen del elegido está ocupado. En caso afirmativo, se devuelve la llamada para volver a asignarla. Si todas las líneas están ocupadas, se entra en un ciclo interminable de manera que antes de asignar el origen se hace una prueba en un bloque TEST para asegurarse de que al menos dos líneas no estén ocupadas. El bloque TEST utiliza la VARIABLE 2 para hacer la prueba, y si encuentra insatisfactorias las condiciones abandona la llamada. También es posible que el segundo bloque ASSIGN escoja que el destino sea igual al origen, lo que se prueba en otro bloque TEST que compra el parámetro 2 contra el parámetro 1. Si son iguales, se devuelve la transacción al segundo bloque ASSIGN para reasignar el destino.

Cuando se genera una llamada válida, el modelo hace que la línea de llamada quede ocupada preñdiendo un interruptor, usando direccionamiento indirecto. Luego determina si hay un enlace disponible tratando de entrar al almacenaje llamado LNKS cuya capacidad es igual al número de enlaces (10). Si no puede entrar la transacción, la llamada se envía a un bloque TERMINATE llamado 3LKD y se pierde la llamada después de que se restaura el interruptor de la línea que hizo la llamada. Las transacciones que obtiene un enlace prueba si la parte llamada está ocupada utilizando un bloque GATE,

usando de nuevo el direccionamiento indirecto. Si la línea está ocupada, se pierde la llamada y va a una localización BUSY donde termina la transacción después de apagar el interruptor de la línea que llamó y de volver el enlace. En caso contrario, se establece la llamada prendiendo el interruptor lógico que corresponde al destino. Un bloque ADVANCE representa el gasto de tiempo durante la llamada, utilizando la función número 1 con una media de 120. Cuando la transacción pasa a desconectarla apagando ambos interruptores lógicos, liberando el enlace terminado.

Operaciones de Conjuntos

Un requerimiento importante en un lenguaje de simulación es que pueda manejar conjuntos de entidades temporales con alguna propiedad en común. En el GPSS, las transacciones bloqueadas se entran y sacan automáticamente de los conjuntos con una disciplina PEPS. El programa tiene la manera de permitir al usuario controlar los conjuntos de manera que se puedan simular disciplinas más complejas de colas. Se cuenta con una cantidad de cadenas de usuarios y se coloca a una transacción en una cadena cuando entra a un bloque LINK. El campo A lleva el número (o nombre) de la cadena y el campo B indica la disciplina de colas. Las palabras FIFO o LIFO (First in first out y Last in last out) producen las disciplinas que llaman. Si se utiliza Pn, se ordenan las transacciones según los valores ascendentes del número de parámetro n, en que una regla FIFO (PEPS) de

transacciones tiene el mismo valor. Mientras están en la cadena, las transacciones permanecen en el bloque LINK.

Para corresponder al bloque Link, existe un bloque UNLINK que permite que otra transacción (fuera de la cadena) saque transacciones de la cadena. El bloque llama a la cadena en el campo A y en el campo B de la localización a donde deben de ir las transacciones desenlazadas. El campo C indica cuántas transacciones deben de sacarse. La cuenta puede ser un entero, el valor de un SMA o se puede utilizar la palabra ALL para quitar todas las transacciones. Si se especifican solamente estos tres primeros campos, el programa saca las transacciones del principio de la cadena. Sin embargo, se puede hacer que la extracción dependa del valor de cualquier parámetro de las transacciones en la cadena. El campo D lleva el número del parámetro a examinar y el campo E lleva el valor que debe de tener el parámetro para que se saque la transacción. Si no se utiliza el campo F, la transacción de desenlace va al siguiente bloque, como lo hace siempre si saca una transacción.

En el sistema telefónico las llamadas bloqueadas esperan a que se desocupe un enlace con las siguientes reglas de servicio. La línea 1 pertenece al presidente de la empresa. Si hay una llamada de entrada para la línea 1 y ésta está desocupada, el siguiente enlace desocupado va a esa llamada. En caso contrario, el enlace va a la llamada que tenga el número más bajo de origen.

Cuando se bloquean las transacciones, se envían a un bloque LINK que las coloca en una cadena que se conoce como WAIT en orden ascendente del origen de llamadas (parámetro 1). Cuando una llamada termina y libera un enlace, prueba si hay llamada en espera yendo a un bloque TEST que compara un SNA de nombre CHn contra 0. Este SNA es igual a la cantidad de transacciones en la cadena n. Si hay una llamada en espera, la transacción va a un bloque GATE para determinar si la línea 1 está desocupada. De ser así, va al primer bloque UNLINK que busca las transacciones de la cadena con destino (parámetro 2) igual a 1. Si existe una de esas llamadas de espera, se desenlaza y envía a GETL para conectarse. Si hay más de una llamada para la línea 1, se escoge la que ha esperado el máximo tiempo. Cuando la línea 1, se escoge la que ha esperado el máximo tiempo. Cuando la línea está desocupada, la transacción que desenlaza va al segundo bloque UNLINK que toma la primera transacción de la cadena; es decir, la que tiene el origen más bajo. Si la transacción va a desenlazar una llamada de la línea 1 y no encuentra una va al segundo bloque UNLINK para desenlazar la llamada con el origen más bajo.

Las llamadas que encuentren una condición de ocupado pasan ahora en el bloque TEST en CKCH debido a que pueden haber salido de la cadena. De ser así a otra llamada en espera se le debe de dar la oportunidad de usar el enlace. Note también que se ha cambiado la variable 2, de manera que

el bloque TEST, que determina si es factible generar una nueva llamada, ahora lleva cuenta de las llamadas en espera.

Otra manera de utilizar el bloque UNLINK permite que el desenlazado dependa de las condiciones del sistema. Este método utiliza declaraciones de variables booleanas, que son semejantes a las declaraciones de variables, pero en vez de combinar SNAs simples, utilizan las frases de prueba condicional de los bloques GATE y TEST. Cada término de una variable booleana es solo prueba que se puede hacer en un bloque GATE y TEST. Los términos se combinan con los operandos * y + para las pruebas y (AND) y (OR) . Los términos individuales toman los valores 1 ó 0 de acuerdo con que la prueba sea verdadera o falsa. Los valores 1 ó 0 de acuerdo con que la prueba sea verdadera o falsa. Los valores se combinan de acuerdo con las reglas de los operadores. Por ejemplo, considere la siguiente codificación:

```
I VARIABLE P2'E'1*LR1 + P2'E'2*LR2
```

La variable lógica es igual a 1 (o verdadero) si el parámetro número 2 es igual a 1. Note que las condiciones del bloque TEST se escriben entre apóstrofes.

Si el primer bloque UNLINK tiene BV1 en el campo D (y nada en el campo E), desenlaza las transacciones que satisfacen la condición expresada. Con esta forma del bloque UNLINK no es necesario incluir el bloque GATE que prueba si está desocupada la línea número 1.

Se describirán brevemente otras dos maneras de organizar los conjuntos en el GPSS, aunque no se describirán con detalle los tipos de bloques que se emplean. Las transacciones que están en una cadena se mantienen estáticas hasta desenlazarse. A veces es necesario identificar los números de un conjunto que continúan moviéndose en el sistema. Por ejemplo, puede necesitarse identificar todos los trabajos de una fábrica para un cliente, o todos los autos de una marca dada. El programa de GPSS define una cantidad de grupos para formar esos conjuntos. Un bloque de tipo JOIN permite que una operación se constituya en miembro de un grupo y que luego pase al siguiente bloque. Otro bloque del tipo REMOVE permite que una transacción se saque a sí misma del grupo; se puede utilizar para permitir que una transacción saque a otras en forma semejante a como las transacciones se desenlazan de una cadena. Es posible escudriñar (SCAN) el grupo en busca de miembros específicos, alterar (ALTER) los parámetros del miembro del grupo o examinar (EXAMINE) los constituyentes del grupo de una transacción.

Una razón común de querer formar conjuntos móviles de transacciones es que representan tareas interrelacionadas que deben de coordinarse. Por ejemplo, la formación de un producto puede involucrar muchas operaciones, algunas de las cuales pueden desarrollarse en forma independiente; pero hay otras, tales como un armado, que requiere que se terminen

determinadas operaciones antes. Un bloque SPLIT (división) permite que una transacción produzca muchas copias que se enlazan automáticamente como un conjunto pero pueden proseguir en forma independiente. Un tipo de bloque llamado ASSEMBLE (ensamblar) reúne un número dado de copias y las funde en una. También es posible sincronizar el movimiento de las copias con el uso de un bloque MATCH (correspondencia), que normalmente se utiliza por parte. Las copias que llegan a un bloque MATCH deben de esperar hasta que haya llegado un número especificado de copias a algún otro bloque MATCH antes de que puedan proseguir todas las copias que esperan.

```

BLOCK CURRENT      TOTAL      INVENTAR.gps SOURCE CODE
-----
10                 0                TRANSFER      ,TAB
11                 1                GENERATE      ...1
12                 0                WATCH        TEST LE      XSSTOCK,XSROP
13                 0                ADVANCE      FNSLTIME
14                 0                SAVEVALUE    STOCK+,XSROQ
-----
S/C: OFF ABS CLOCK: 0.      REL CLOCK: 0.      TTG: 1000
-----
XACT: 2      CURBLK: 11      NEXTBLK: WATCH      CHAINS: CEC      PC:
-----
MARK-TIME: ----      MOVE-TIME: 0.      PRIORITY: 0
-----
                AN INTRODUCTION TO SIMULATION USING GPSS/H
                by Thomas J. Schriber (Wiley, 1990)

                Copyright (c) 1990, WOLVERINE SOFTWARE CORPORATION

                ANNHANDALE, VIRGINIA 22003-2500, USA

Ready!
: STEP 1

XACT 2 POISED AT BLOCK 12 (WATCH).  RELATIVE CLOCK: 0.
:
    
```

Pantalla 2.

En pantalla 2, podemos observar tres divisiones: la primera muestra el código fuente la segunda el bloque o instrucción que se está ejecutando y el tercero es la línea de comandos donde podemos ejecutar el programa paso a paso o iniciarle que pare en una transacción deseada.

```

RELATIVE CLOCK: 0.      ABSOLUTE CLOCK: 0.
BLOCK CURRENT      TOTAL      BLOCK CURRENT      TOTAL
1                 0      11                 1
2                 0      WATCH              0
3                 0      13                 0
4                 0      14                 0
TAB               0      15                 0
6                 0
7                 0
TRUBL             0
9                 0
10                0
NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)
      HOP:          80,      ROQ:          100,      STOCK:        100
STATUS OF COMMON STORAGE
      9552 BYTES AVAILABLE
      448 IN USE
      448 USED (MAX)
: D OUT
    
```

Pantalla 3.

En pantalla 3, se aprecia una pantalla completa del diálogo (ZOOM), esto se hace optimizando la tecla "3", en la que se observa el reloj relativo y absoluto, y como estamos en el ejemplo de inventarios y le pedimos que desplegara la salida

APENDICE C:

Salida de un programa general de inventarios

Para entrar al lenguaje gpssh de manera interactiva, se realiza con el siguiente comando:

```
C:\GPSSH>GPSSH INVENTAR -TV
```

Lo cual nos indica que va a ejecutar de manera interactiva el programa inventar.gps, via el parametro TV. La presentación del paquete se muestra en la pantalla 1.

En el caso de que no se hubiera tecleado el parametro TV nos dará un archivo con la salida de ejecución con el nombre del mismo nombre pero con extensión LIS.

Para iniciar la ejecución del programa en el modo de depuración (TV test mode), tenemos que teclear "step 1" para ejecutar una línea u oprimir "F10"; cuando se quiere ejecutar todo el programa se teclaea "run".

| BLOCK CURRENT | TOTAL | INVENTAR.gps | SOURCE CODE |
|---|------------|--------------|----------------------|
| ===== | | | |
| S/C: OFF | ABS CLOCK: | 0. | REL CLOCK: 0. TIG: 0 |
| XACT: | CURBLK: | NEXTBLK: | CHAINS: PC: |
| MARK-TIME: | MOVE-TIME: | PRIORITY: | |
| ===== | | | |
| Simulation begins. | | | |
| This version of Student GPSS/H is distributed with AN INTRODUCTION TO SIMULATION USING GPSS/H by Thomas J. Schriber (Wiley, 1990) | | | |
| Copyright (c) 1990, WOLVERINE SOFTWARE CORPORATION ANNANDALE, VIRGINIA 22003-2500, USA | | | |
| Ready! : | | | |

Pantalla 1.

(D OUT), nos desplaza los bloques ejecutados al momento además de los valores iniciales del punto de reorden (ROP = 80), de la cantidad de reorden (ROQ=100) y el inventario (STOCK = 100).

Apéndice c 153

| BLOCK CURRENT | TOTAL | INVENTAR.gps | SOURCE CODE |
|--|-----------|-------------------|-------------------|
| 1 | 1 | GENERATE | 1,.,.,1 |
| 2 | 0 | ASSIGN | 1,VSDMND |
| 3 | 0 | TEST GE | X\$STOCK,P1,TRUBL |
| 4 | 0 | SAVEVALUE | STOCK-,P1 |
| ----- | | | |
| S/C: OFF ARS CLOCK: 1.0000 | | REL CLOCK: 1.0000 | TTG: 1000 |
| ----- | | | |
| XACT: 1 | CURBLK: 1 | NEXTBLK: 2 | CHAINS: CEC PC: |
| ----- | | | |
| MARK-TIME: 1.0000 | | MOVE-TIME: 1.0000 | PRIORITY: 1 |
| ----- | | | |
| STATUS OF COMMON STORAGE | | | |
| 9552 BYTES AVAILABLE | | | |
| 448 IN USE | | | |
| 448 USED (MAX) | | | |
| : STEP 1 | | | |
| XACT 1 POISED AT BLOCK 2. RELATIVE CLOCK: 1.0000 | | | |
| : | | | |

Pantalla 4

En la pantalla 4, inicia la primer transacción, indicándonos que se encuentra en el primer bloque y el próximo será el bloque 2.

| BLOCK CURRENT | TOTAL | BLOCK CURRENT | TOTAL |
|--|---------|---------------|-----------------|
| 1 | 1 11 | 1 | 1 |
| 2 | 0 WATCH | | 0 |
| 3 | 0 13 | | 0 |
| 4 | 0 14 | | 0 |
| TAB | 0 15 | | 0 |
| 6 | 0 | | |
| 7 | 0 | | |
| TRUBL | 0 | | |
| 9 | 0 | | |
| 10 | 0 | | |
| NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE) | | | |
| ROP: | 80, | ROQ: | 100, STOCK: 100 |
| STATUS OF COMMON STORAGE | | | |
| 9440 BYTES AVAILABLE | | | |
| 560 IN USE | | | |
| 560 USED (MAX) | | | |
| : | | | |

Pantalla 5

En la pantalla 5, se oprimió "FJ" para que nos ampliara la zona del diálogo.

```

2          1 WATCH          0
3          0 13             0
4          0 14             0
TAB        0 15             0
6          0
7          0
TRUBL      0
9          0
10         0
NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)
      ROP:          80,      ROQ:          100,      STOCK:          100
RANDOM    ANTIHETIC    INITIAL    CURRENT    SAMPLE    CHI-SQUARE
STREAM   VARIATIVES   POSITION    POSITION    COUNT    UNIFORMITY
      1          OFF      11      12      1      N/A
STATUS OF COMMON STORAGE
      9440 BYTES AVAILABLE
      560 IN USE
      560 USED (MAX)
:

```

Pantalla 6

En la pantalla 6, esta información nos la muestra al darle "D OUT" en la línea de comandos, en donde observamos en la parte de abajo que se está utilizando el generador de números aleatorios número 1, nos despliega en la Ji cuadrada que no es aplicable ya(N/A) que requiere mínimo de 80 números aleatorios para realizar esta prueba.

```

BLOCK CURRENT    TOTAL    INVENTAR.gps SOURCE CODE
-----
2          1          ASSIGN    1,VSDMND
3          1          TEST GE   X$STOCK,P1,TRUBL
4          0          SAVEVALUE STOCK-,P1
5          0          TAB      STOCK
6          0          TABULATE LOSSES
=====
S/C: OFF ABS CLOCK: 1.0000    REL CLOCK: 1.0000    TTG: 1000
XACT: 1          CURBLK: 3    NEXTBLK: 4          CHAINS: CEC    PC:
MARK-TIME: 1.0000    MOVE-TIME: 1.0000    PRIORITY: 1
=====
STATUS OF COMMON STORAGE
      9440 BYTES AVAILABLE
      560 IN USE
      560 USED (MAX)
: STEP 1
XACT 1 POISED AT BLOCK 4.    RELATIVE CLOCK: 1.0000
:

```

Pantalla 7

En la pantalla 7, regresaremos a las ventanas de diálogo, código fuente y estatus (set tv on), pulsando la tecla "F4".

```

2          1 WATCH          0
3          1 13            0
4          1 14            0
TAB        0 15            0
5          0
6          0
7          0
TRUBL      0
9          0
10         0
    
```

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

ROP: 80, ROQ: 100, STOCK: 91

| | | | | | |
|--------|-----------|----------|----------|--------|------------|
| RANDOM | ANTIETHIC | INITIAL | CURRENT | SAMPLE | CHI-SQUARE |
| STREAM | VARIATES | POSITION | POSITION | COUNT | UNIFORMITY |
| 1 | OFF | 11 | 12 | 1 | N/A |

STATUS OF COMMON STORAGE

```

9440 BYTES AVAILABLE
560 IN USE
560 USED (MAX)
    
```

TABLE STOCK

| | | | | | | |
|------------------|---------------|--------------------|------------------|------------|----------|-------|
| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | NON-W | | |
| 1.0000 | 91.0000 | 0. | 91.0000 | | | |
| UPPER | OBSERVED | PERCENT | CUMULATIVE | CUMULATIVE | MULTIPLE | DEVIA |
| LIMIT | FREQUENCY | OF TOTAL | PERCENTAGE | REMAINDER | OF MEAN | FROM |
| ... | | | | | | |
| 100.0000 | 1.0000 | 100.00 | 100.00 | 0.00 | 1.0989 | 0. |

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

ROP: 80, ROQ: 100, STOCK: 91

| | | | | | |
|--------|-----------|----------|----------|--------|------------|
| RANDOM | ANTIETHIC | INITIAL | CURRENT | SAMPLE | CHI-SQUARE |
| STREAM | VARIATES | POSITION | POSITION | COUNT | UNIFORMITY |
| 1 | OFF | 11 | 12 | 1 | N/A |

STATUS OF COMMON STORAGE

```

9440 BYTES AVAILABLE
560 IN USE
560 USED (MAX)
    
```

Pantalla 8

En la pantalla 8, que es una combinación de dos pantallas para desplegar la salida de información al momento con el comando "D OUT", observamos que aquí ya aparece TABLE STOCK, que va haciendo un registro de las entradas y salidas del almacén (STOCK).

| BLOCK CURRENT | TOTAL | INVENTAR.gps SOURCE CODE | | |
|--|-----------|--------------------------|-------------|-------------|
| 5 | 1 | TAB | TABULATE | STOCK |
| 6 | 1 | | TABULATE | LOSES |
| 7 | 0 | | TERMINATE | 1 |
| 8 | 0 | TRUBL | ASSIGN | 2,V\$LOST |
| 9 | 0 | | SAVEVALUE | STOCK,0 |
| ----- | | | | |
| S/C: OFF ABS CLOCK: 1.0000 | | REL CLOCK: 1.0000 | | TIG: 1000 |
| ----- | | | | |
| XACT: 1 | CURBLK: 6 | NEXTBLK: 7 | CHAINS: CEC | PC: |
| ----- | | | | |
| MARK-TIME: 1.0000 | | MOVE-TIME: 1.0000 | | PRIORITY: 1 |
| ----- | | | | |
| STATUS OF COMMON STORAGE | | | | |
| 9440 BYTES AVAILABLE | | | | |
| 560 IN USE | | | | |
| 560 USED (MAX) | | | | |
| : STEP 1 | | | | |
| XACT 1 POISED AT BLOCK 7 (TAB+2). RELATIVE CLOCK: 1.0000 | | | | |
| : | | | | |

Pantalla 9

En la pantalla 9, se observa que pasamos al bloque 8 el cual es la segunda de las tablas, o sea, las perdidas.

| | | | | | | | |
|--|--------------------|--------------------|-----------------------|----------------------|------------------|----------------|--|
| 10 | | | | | | 0 | |
| TABLE LOSES | | | | | | | |
| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | | SUM OF ARGUMENTS | | NON-W | |
| 1.0000 | 0. | 0. | | 0. | | | |
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM | |
| 0. | 1.0000 | 100.00 | 100.00 | 0.00 | 0.0 | 0. | |
| TABLE STOCK | | | | | | | |
| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | | SUM OF ARGUMENTS | | NON-W | |
| 1.0000 | 91.0000 | 0. | | 91.0000 | | | |
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM | |
| ... | 100.0000 | 1.0000 | 100.00 | 100.00 | 0.00 | 1.0989 | |
| NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE) | | | | | | | |
| : | | | | | | | |

Pantalla 10

En la pantalla 10, se observa ya las dos tablas la del almacén (STOCK) y la de perdidas (LOSES), las cuales indican estadísticas.

| BLOCK CURRENT | TOTAL | INVENTAR.gpa SOURCE CODE | | |
|---|-------------------|--------------------------|-----------|-----------|
| 5 | 1 | TAB | TABULATE | STOCK |
| 6 | 1 | | TABULATE | LOSES |
| 7 | 1 | | TERMINATE | 1 |
| 8 | 0 | TRUBL | ASSIGN | 2,V\$LOST |
| 9 | 0 | | SAVEVALUE | STOCK,0 |
| ----- | | | | |
| S/C: OFF | ABS CLOCK: 1.0000 | REL CLOCK: 1.0000 | TTG: 999 | |
| ----- | | | | |
| XACT: 1 | CURBLK: 6 | NEXTBLK: 7 | CHAINS: | PC: |
| ----- | | | | |
| MARK-TIME: 1.0000 | MOVE-TIME: 1.0000 | PRIORITY: 1 | | |
| ----- | | | | |
| STATUS OF COMMON STORAGE | | | | |
| 9440 BYTES AVAILABLE | | | | |
| 560 IN USE | | | | |
| 560 USED (MAX) | | | | |
| : STEP 1 | | | | |
| XACT 1 DESTROYED AT BLOCK 7 (TAB+2). RELATIVE CLOCK: 1.0000 | | | | |
| : | | | | |

Pantalla 11

En la pantalla 11, se destruye la primer transacción, indicándonos que se encuentra en el primer bloque y el próximo será el bloque 2.

| BLOCK CURRENT | TOTAL | BLOCK CURRENT | TOTAL | | | |
|------------------|--------------------|--------------------|-----------------------|----------------------|------------------|----------------|
| 1 | 1 | 1 | 1 | | | |
| 2 | 1 | WATCH | 0 | | | |
| 3 | 1 | 12 | 0 | | | |
| 4 | 1 | 14 | 0 | | | |
| TAB | 1 | 15 | 0 | | | |
| 6 | 1 | | | | | |
| 7 | 1 | | | | | |
| TRUBL | 0 | | | | | |
| 9 | 0 | | | | | |
| 10 | 0 | | | | | |
| TABLE LOSSES | | | | | | |
| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | NON-W | | |
| 1.0000 | 0. | 0. | 0. | | | |
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
| 0. | 1.0000 | 100.00 | 100.00 | 0.00 | 0.0 | 0. |
| TABLE STOCK | | | | | | |
| : | | | | | | |

Pantalla 12

| BLOCK CURRENT | TOTAL | INVENTAR.gps | SOURCE CODE |
|--|-----------------------|-------------------|----------------------------|
| 1 | 2 | GENERATE | 1,,,1 |
| 2 | 1 | ASSIGN | 1,VSDMND |
| 3 | 1 | TEST GE | X\$STOCK,P1,TRUBL |
| 4 | 1 | SAVEVALUE | STOCK-,P1 |
| ----- | | | |
| S/C: | OFF ABS CLOCK: 2.0000 | REL CLOCK: 2.0000 | TTG: 999 |
| ----- | | | |
| XACT: | 3 | CURBLK: 1 | NEXTBLK: 2 CHAINS: CEC PC: |
| ----- | | | |
| MARK-TIME: | 2.0000 | MOVE-TIME: 2.0000 | PRIORITY: 1 |
| ----- | | | |
| STATUS OF COMMON STORAGE | | | |
| 9552 BYTES AVAILABLE | | | |
| 448 IN USE | | | |
| 560 USED (MAX) | | | |
| : STEP 1 | | | |
| XACT 3 POISED AT BLOCK 2. RELATIVE CLOCK: 2.0000 | | | |
| : | | | |

Pantalla 13

En la pantalla 13, se inicia la tercer transacción, indicándonos que se encuentra en el primer bloque y el próximo sera el bloque 2.

| BLOCK CURRENT | TOTAL | INVENTAR.gps | SOURCE CODE |
|--|-----------------------|-------------------|----------------------------|
| 5 | 2 | TAB | TABULATE STOCK |
| 6 | 2 | | TABULATE LOSES |
| 7 | 1 | | TERMINATE 1 |
| 8 | 0 | TRUBL | ASSIGN 2,V\$LOST |
| 9 | 0 | | SAVEVALUE STOCK,0 |
| ----- | | | |
| S/C: | OFF ABS CLOCK: 2.0000 | REL CLOCK: 2.0000 | TTG: 999 |
| ----- | | | |
| XACT: | 3 | CURBLK: 6 | NEXTBLK: 7 CHAINS: CEC PC: |
| ----- | | | |
| MARK-TIME: | 2.0000 | MOVE-TIME: 2.0000 | PRIORITY: 1 |
| ----- | | | |
| : STEP 1 | | | |
| XACT 3 POISED AT BLOCK 5 (TAB). RELATIVE CLOCK: 2.0000 | | | |
| : STEP 1 | | | |
| XACT 3 POISED AT BLOCK 6 (TAB+1). RELATIVE CLOCK: 2.0000 | | | |
| : STEP 1 | | | |
| XACT 3 POISED AT BLOCK 7 (TAB+2). RELATIVE CLOCK: 2.0000 | | | |
| : | | | |

Pantalla 14

En la pantalla 14, se encuentra en la transacción J con un reloj relativo y absoluto de 2.

```

XACT 3 POISED AT BLOCK 5 (TAB). RELATIVE CLOCK: 2.0000
: STEP 1
XACT 3 POISED AT BLOCK 6 (TAB+1). RELATIVE CLOCK: 2.0000
: STEP 1
XACT 3 POISED AT BLOCK 7 (TAB+2). RELATIVE CLOCK: 2.0000
: D OUT
RELATIVE CLOCK: 2.0000 ABSOLUTE CLOCK: 2.0000
BLOCK CURRENT      TOTAL  BLOCK CURRENT      TOTAL
1                   2      11                   1
2                   2      WATCH                0
3                   2      13                   0
4                   2      14                   0
TAB                 2      15                   0
6                   2
7                   1
TRUBL               0
9                   0
10                  0
:
    
```

Pantalla 15

```

TABLE      LOSER
ENTRIES IN TABLE  MEAN ARGUMENT  STANDARD DEVIATION  SUM OF ARGUMENTS  NON-W
2.0000          0.              0.              0.
UPPER OBSERVED PERCENT CUMULATIVE CUMULATIVE MULTIPLE DEVI
LIMIT FREQUENCY OF TOTAL PERCENTAGE REMAINDER OF MEAN FROM
0.      2.0000    100.00    100.00    0.00    0.0    0.
TABLE      STOCK
ENTRIES IN TABLE  MEAN ARGUMENT  STANDARD DEVIATION  SUM OF ARGUMENTS  NON-W
2.0000          87.0000    5.6569    174.0000    NON-W
UPPER OBSERVED PERCENT CUMULATIVE CUMULATIVE MULTIPLE DEVI
LIMIT FREQUENCY OF TOTAL PERCENTAGE REMAINDER OF MEAN FROM
...
90.0000    1.0000    50.00    50.00    50.00    1.0345    0.
100.0000   1.0000    50.00    100.00    0.00    1.1494    2.
NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)
:
    
```

Pantalla 16

En la pantalla 16, se observa que la tabla de almacén ya tiene dos valores y se puede apreciar mejor sus estadísticas continuas en la sig. página el comando "D OUT".

| ENTRIES IN TABLE 2.0000 | MEAN ARGUMENT 87.0000 | STANDARD DEVIATION 5.6569 | SUM OF ARGUMENTS 174.0000 | NON-W | | |
|----------------------------|--------------------------|------------------------------|------------------------------|----------------------|------------------|----------------|
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
| 90.0000 | 1.0000 | 50.00 | 50.00 | 50.00 | 1.0345 | 0. |
| 100.0000 | 1.0000 | 50.00 | 100.00 | 0.00 | 1.1494 | 2. |

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

ROP: 80, ROQ: 100, STOCK: 83

| RANDOM STREAM | ANTITHETIC VARIATES | INITIAL POSITION | CURRENT POSITION | SAMPLE COUNT | CHI-SQUARE UNIFORMITY |
|---------------|---------------------|------------------|------------------|--------------|-----------------------|
| 1 | OFF | 11 | 13 | 2 | N/A |

STATUS OF COMMON STORAGE

9440 BYTES AVAILABLE
560 IN USE
560 USED (MAX)

Pantalla 17

En la pantalla 17, se observa que ya se pidieron 17 artículos quedando 83 en el almacén.

| BLOCK CURRENT | TOTAL | INVENTAR.gps | SOURCE CODE |
|---------------|-------|--------------|-------------------------|
| 11 | 1 | GENERATE | ,,,1 |
| 12 | 1 | WATCH | TEST LE X\$STOCK,X\$ROF |
| 13 | 0 | ADVANCE | F\$SLTIME |
| 14 | 0 | SAVEVALUE | STOCK+,X\$ROQ |
| 15 | 0 | TRANSFER | ,WATCH |

S/C: OFF ABS CLOCK: 3.0000 REL CLOCK: 3.0000 TTG: 997

XACT: 2 CURBLK: WATCH NEXTBLK: 13 CHAINS: CEC PC:

MARK-TIME: 3.0000 MOVE-TIME: 0. PRIORITY: 0

: STEP 1

XACT 4 POISED AT BLOCK 7 (TAB*2). RELATIVE CLOCK: 3.0000
: STEP 1

XACT 4 DESTROYED AT BLOCK 7 (TAB*2). RELATIVE CLOCK: 3.0000
: STEP 1

XACT 2 POISED AT BLOCK 13 (WATCH*1). RELATIVE CLOCK: 3.0000
:

Pantalla 18

En la pantalla 18, se observa como pasa por la etiqueta WATCH, la cual simula un inspector que inspecciona físicamente el inventario en almacén, y cuando esté llegue a ser el mismo o menor al punto de reorden (ROP=80), se pedira una cantidad fijada en ROQ.

```

: STEP 1
XACT 4 DESTROYED AT BLOCK 7 (TAB+2).  RELATIVE CLOCK: 3.0000
: STEP 1
XACT 2 POISED AT BLOCK 13 (WATCH+1).  RELATIVE CLOCK: 3.0000
: D OUT

RELATIVE CLOCK: 3.0000  ABSOLUTE CLOCK: 3.0000

BLOCK CURRENT  TOTAL  BLOCK CURRENT  TOTAL
1              3      11              1
2              3      WATCH              1
3              3      13              0
4              3      14              0
TAB            3      15              0
6              3
7              3
TRUBL          0
9              0
10             0

TABLE  LOSSES
:

```

Pantalla 19

```

TABLE  LOSSES

ENTRIES IN TABLE  MEAN ARGUMENT  STANDARD DEVIATION  SUM OF ARGUMENTS  NON-W
3.0000             0.              0.              0.              0.

UPPER OBSERVED  PERCENT CUMULATIVE CUMULATIVE  MULTIPLE DEVIAT
LIMIT FREQUENCY OF TOTAL PERCENTAGE REMAINDER OF MEAN FROM
G.      3.0000   100.00   100.00   0.00   0.0   0.

TABLE  STOCK

ENTRIES IN TABLE  MEAN ARGUMENT  STANDARD DEVIATION  SUM OF ARGUMENTS  NON-W
3.0000             82.0000         9.5394             246.0000         0.

UPPER OBSERVED  PERCENT CUMULATIVE CUMULATIVE  MULTIPLE DEVIAT
LIMIT FREQUENCY OF TOTAL PERCENTAGE REMAINDER OF MEAN FROM
...
80.0000  1.0000   33.33   33.33   66.67   0.9756  -0.
90.0000  1.0000   33.33   66.67   33.33   1.0976  0.
100.0000 1.0000   33.33   100.00  -0.00   1.2195  1.

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)
:

```

Pantalla 20

| ENTRIES IN TABLE 3.0000 | MEAN ARGUMENT 82.0000 | STANDARD DEVIATION 9.5394 | SUM OF ARGUMENTS 246.0000 | NON-W | | |
|----------------------------|--------------------------|------------------------------|------------------------------|----------------------|------------------|----------------|
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
| 80.0000 | 1.0000 | 33.33 | 33.33 | 66.67 | 0.9756 | -0. |
| 90.0000 | 1.0000 | 33.33 | 66.67 | 33.33 | 1.0976 | 0. |
| 100.0000 | 1.0000 | 33.33 | 100.00 | -0.00 | 1.2195 | 1. |

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

ROP: 80, ROQ: 100, STOCK: 72

| RANDOM STREAM | ANTITHETIC VARIATES | INITIAL POSITION | CURRENT POSITION | SAMPLE COUNT | CHI-SQUARE UNIFORMITY |
|---------------|---------------------|------------------|------------------|--------------|-----------------------|
| 1 | OFF | 11 | 14 | 3 | N/A |

STATUS OF COMMON STORAGE

9552 BYTES AVAILABLE
448 IN USE
560 USED (MAX)

Pantalla 21

En la pantalla 21, se observa que ya es menor los artículos del almacén que el punto de reorden (72<80).

| BLOCK CURRENT | TOTAL | INVENTAR.gro | SOURCE CODE |
|---------------|-------|--------------|-------------------------|
| 11 | 1 | | GENERATE ,,,1 |
| 12 | 1 | WATCH | TEST LE XSSTOCK,X\$ROP |
| 13 | 1 | | ADVANCE FHLTIME |
| 14 | 0 | | SAVEVALUE STOCK*,X\$ROQ |
| 15 | 0 | | TRANSFER ,WATCH |

=====

S/C: OFF ABS CLOCK: 3.0000 REL CLOCK: 3.0000 TTT: 997

=====

XACT: 2 CURBLK: 13 NEXTBLK: 14 CHAINS: FEC PC:

MARK-TIME: 3.0000 MOVE-TIME: 13.0000 PRIORITY: 0

=====

STATUS OF COMMON STORAGE

9552 BYTES AVAILABLE
448 IN USE
560 USED (MAX)

: STEP 1

XACT 2 PLACED ON FEC AT BLOCK 13 (WATCH+1). RELATIVE CLOCK: 3.0000

Pantalla 22

En la pantalla 22, se observa que ya es menor el STOCK que ROP, y por lo tanto se ejecuta el siguiente bloque que es realizar el pedido por lo que se simula el tiempo de entrega (LTIME, lead time), que por supuesto genera un número aleatorio lo cual lo podemos comprobar con el comando display out (abreviado D OUT). Lo interesante es observar que mueve un tiempo de 13 unidades en el tiempo de espera, lo cual quiere decir que en ese tiempo va a llegar el pedido.

```

XACT 2 PLACED ON FEC AT BLOCK 13 (WATCH+1).  RELATIVE CLOCK: 3.0000
: D OUT

      UPPER      OBSERVED      PERCENT      CUMULATIVE      CUMULATIVE      MULTIPLE      DEVI
      LIMIT      FREQUENCY      OF TOTAL      PERCENTAGE      REMAINDER      OF MEAN      FROM
      ...
      80.0000      1.0000      33.33      33.33      66.67      0.9756      -0.
      90.0000      1.0000      33.33      66.67      33.33      1.0976      0.
      100.0000      1.0000      33.33      100.00      -0.00      1.2195      1.

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

ROP:          80,          ROQ:          100,          STOCK:          72

RANDOM        ANTIHETIC        INITIAL        CURRENT        SAMPLE        CHI-SQUARE
STREAM       VARIATES         POSITION        POSITION        COUNT        UNIFORMITY
  1          OFF           11           14           3           N/A
  2          OFF           33           34           1           N/A

STATUS OF COMMON STORAGE

  9552 BYTES AVAILABLE
  448 IN USE
  560 USED (MAX)
:
    
```

Pantalla 23

En la pantalla 23, se observa que ya aparece un segundo generador de números aleatorios, el cual, es para simular el tiempo de entrega, lo cual ya observamos que se va a tardar 10 días por que estaba en el tiempo 3 y le indico a la transacción moverse al 13.

```

BLOCK CURRENT      TOTAL      INVENTAR.gps SOURCE CODE
-----
1              4              GENERATE 1,,,1
2              4              ASSIGN 1,VSDMND
3              3              TEST GE X$STOCK,P1,TRUBL
4              3              SAVEVALUE STOCK-/P1
5              3              TAB TABULATE STOCK
-----
S/C: OFF ABS CLOCK: 4.0000  REL CLOCK: 4.0000  TTG: 997
-----
XACT: 5          CURBLK: 2          NEXTBLK: 3          CHAINS: CEC          PC:

MARK-TIME: 4.0000  MOVE-TIME: 4.0000  PRIORITY: 1
-----

STATUS OF COMMON STORAGE

  9440 BYTES AVAILABLE
  560 IN USE
  560 USED (MAX)
: STEP 1

XACT 5 POISED AT BLOCK 3.  RELATIVE CLOCK: 4.0000
:
    
```

Pantalla 24

En la pantalla 24, se observa que en la pantalla anterior estábamos en la transacción 2 y al ejecutar la siguiente línea nos mando a la transacción 5, ya teniendo la demanda para esta transacción.

| BLOCK CURRENT | TOTAL | INVENTAR.gps SOURCE CODE | | | | |
|---|------------|--------------------------|-----------|-----------|------------|----|
| 4 | 5 | | SAVEVALUE | STOCK- | P1 | |
| 5 | 5 | TAB | TABULATE | STOCK | | |
| 6 | 4 | | TABULATE | LOSES | | |
| 7 | 4 | | TERMINATE | 1 | | |
| 8 | 0 | TRUBL | ASSIGN | 2,V\$LOST | | |
| S/C: OFF ABS CLOCK: 5.0000 REL CLOCK: 5.0000 TIG: 996 | | | | | | |
| XACT: 6 CURBLK: TAB NEXTBLK: 6 CHAINS: CEC PC: | | | | | | |
| MARK-TIME: 5.0000 MOVE-TIME: 5.0000 PRIORITY: 1 | | | | | | |
| 90.0000 | 1.0000 | 20.00 | 80.00 | 20.00 | 1.2397 | 1. |
| 100.0000 | 1.0000 | 20.00 | 100.00 | 0.00 | 1.3774 | 1. |
| NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE) | | | | | | |
| ROP: | 80, | ROQ: | 100, | STOCK: | 53 | |
| RANDOM | ANTITHETIC | INITIAL | CURRENT | SAMPLE | CHI-SQUARE | |
| STREAM | VARIATES | POSITION | POSITION | COUNT | UNIFORMITY | |

Pantalla 25

En la pantalla 25, se tabulan los valores del inventario y las perdidas.

| | | | | | |
|---|-------|---------------|-------|---|-----|
| 2 | OFF | 33 | 34 | 1 | N/A |
| STATUS OF COMMON STORAGE | | | | | |
| 9552 BYTES AVAILABLE | | | | | |
| 448 IN USE | | | | | |
| 560 USED (MAX) | | | | | |
| : D OUT | | | | | |
| RELATIVE CLOCK: 9.0000 ABSOLUTE CLOCK: 9.0000 | | | | | |
| BLOCK CURRENT | TOTAL | BLOCK CURRENT | TOTAL | | |
| 1 | 9 | 11 | 1 | | |
| 2 | 9 | WATCH | 1 | | |
| 3 | 9 | 13 | 1 | | |
| 4 | 9 | 14 | 0 | | |
| TAB | 9 | 15 | 0 | | |
| 6 | 9 | | | | |
| 7 | 9 | | | | |
| TRUBL | 0 | | | | |
| 9 | 0 | | | | |
| 10 | 0 | | | | |

Pantalla 26

En la pantalla 26, se teclea "D OUT" para observar la información que se ha generado.

| BLOCK CURRENT | TOTAL | BLOCK CURRENT | TOTAL |
|---------------|-------|---------------|-------|
| 1 | 9 | 11 | 1 |
| 2 | 9 | WATCH | 1 |
| 3 | 9 | 13 | 1 |
| 4 | 9 | 14 | 0 |
| TAB | 9 | 15 | 0 |
| 6 | 9 | | |
| 7 | 9 | | |
| TRURL | 0 | | |
| 9 | 0 | | |
| 10 | 0 | | |

TABLE LOSES

| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | NON-W |
|------------------|---------------|--------------------|------------------|-------|
| 9.0000 | 0. | 0. | 0. | |

| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
|-------------|--------------------|------------------|-----------------------|----------------------|------------------|----------------|
| 0. | 9.0000 | 100.00 | 100.00 | 0.00 | 0.0 | 0. |

TABLE STOCK

Pantalla 27

| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | NON-W |
|------------------|---------------|--------------------|------------------|-------|
| 9.0000 | 55.4444 | 23.6913 | 499.0000 | |

| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
|-------------|--------------------|------------------|-----------------------|----------------------|------------------|----------------|
| ... | | | | | | |
| 30.0000 | 2.0000 | 22.22 | 22.22 | 77.78 | 0.5411 | -1. |
| 40.0000 | 1.0000 | 11.11 | 33.33 | 66.67 | 0.7214 | -0. |
| 50.0000 | 1.0000 | 11.11 | 44.44 | 55.56 | 0.9018 | -0. |
| 60.0000 | 1.0000 | 11.11 | 55.56 | 44.44 | 1.0822 | 0. |
| 70.0000 | 1.0000 | 11.11 | 66.67 | 33.33 | 1.2625 | 0. |
| 80.0000 | 1.0000 | 11.11 | 77.78 | 22.22 | 1.4429 | 1. |
| 90.0000 | 1.0000 | 11.11 | 88.89 | 11.11 | 1.6232 | 1. |
| 100.0000 | 1.0000 | 11.11 | 100.00 | -0.00 | 1.8036 | 1. |

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

| | | | | | |
|---------------|-------------------------|------------------|------------------|--------------|-----------------------|
| ROP: | 80, | ROQ: | 100, | STOCK: | 23 |
| RANDOM STREAM | ANTITHETIC VARIATES OFF | INITIAL POSITION | CURRENT POSITION | SAMPLE COUNT | CHI-SQUARE UNIFORMITY |
| 1 | | 11 | 20 | 9 | N/A |

Pantalla 28

| | | | | | | |
|----------|--------|-------|--------|-------|--------|-----|
| 30.0000 | 2.0000 | 22.22 | 22.22 | 77.78 | 0.5411 | -1. |
| 40.0000 | 1.0000 | 11.11 | 33.33 | 66.67 | 0.7214 | -0. |
| 50.0000 | 1.0000 | 11.11 | 44.44 | 55.56 | 0.9018 | -0. |
| 60.0000 | 1.0000 | 11.11 | 55.56 | 44.44 | 1.0822 | 0. |
| 70.0000 | 1.0000 | 11.11 | 66.67 | 33.33 | 1.2625 | 0. |
| 80.0000 | 1.0000 | 11.11 | 77.78 | 22.22 | 1.4429 | 1. |
| 90.0000 | 1.0000 | 11.11 | 88.89 | 11.11 | 1.6232 | 1. |
| 100.0000 | 1.0000 | 11.11 | 100.00 | -0.00 | 1.8036 | 1. |

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

| ROP: | 80, | ROQ: | 100, | STOCK: | 23 |
|--------|------------|----------|----------|--------|------------|
| RANDOM | ANTITHETIC | INITIAL | CURRENT | SAMPLE | CHI-SQUARE |
| STREAM | VARIATES | POSITION | POSITION | COUNT | UNIFORMITY |
| 1 | OFF | 11 | 20 | 9 | N/A |
| 2 | OFF | 33 | 34 | 1 | N/A |

STATUS OF COMMON STORAGE

9552 BYTES AVAILABLE
448 IN USE
560 USED (MAX)

Pantalla 29

En las 4 pantallas anteriores ya se puede observar más información, pero hasta el momento han podido atender todas las demandas de pedido, ya que no hay nada en la tabla de pérdidas.

| BLOCK CURRENT | TOTAL | INVENTAR.gpb | SOURCE CODE |
|---------------|-------|--------------|-------------------|
| 5 | 10 | TAB | TABULATE STOCK |
| 6 | 10 | | TABULATE LOSES |
| 7 | 10 | | TERMINATE 1 |
| 8 | 0 | TRUBL | ASSIGN 2,V\$LOST |
| 9 | 0 | | SAVEVALUE STOCK,0 |

=====

S/C: OFF ABS CLOCK: 10.0000 REL CLOCK: 10.0000 TTG: 990

XACT: 11 CURBLK: 6 NEXTBLK: 7 CHAINS: PC:

MARK-TIME: 10.0000 MOVE-TIME: 10.0000 PRIORITY: 1

=====

: STEP 1

XACT 11 POISED AT BLOCK 6 (TAB+1). RELATIVE CLOCK: 10.0000

: STEP 1

XACT 11 POISED AT BLOCK 7 (TAB+2). RELATIVE CLOCK: 10.0000

: STEP 1

XACT 11 DESTROYED AT BLOCK 7 (TAB+2). RELATIVE CLOCK: 10.0000

: STEP 1

Pantalla 30

| BLOCK CURRENT | TOTAL | INVENTAR.gps SOURCE CODE | | |
|---|--------------------|--------------------------|-------------|-----------------|
| 8 | 1 | TRUBL | ASSIGN | 2,V\$LOST |
| 9 | 1 | | SAVEVALUE | STOCK,0 |
| 10 | 0 | | TRANSFER | ,TAB |
| 11 | 1 | | GENERATE | ,,1 |
| 12 | 1 | WATCH | TEST LE | X\$STOCK,X\$ROP |
| ----- | | | | |
| S/C: OFF | AHS CLOCK: 12.0000 | REL CLOCK: 12.0000 | TTG: 989 | |
| XACT: 13 | CURBLK: 9 | NEXTBLK: 10 | CHAINS: CEC | PC: |
| ----- | | | | |
| MARK-TIME: 12.0000 | MOVE-TIME: 12.0000 | PRIORITY: 1 | | |
| ----- | | | | |
| : STEP 1 | | | | |
| XACT 13 POISED AT BLOCK 8 (TRUBL). RELATIVE CLOCK: 12.0000 | | | | |
| : STEP 1 | | | | |
| XACT 13 POISED AT BLOCK 9 (TRUBL+1). RELATIVE CLOCK: 12.0000 | | | | |
| : STEP 1 | | | | |
| XACT 13 POISED AT BLOCK 10 (TRUBL+2). RELATIVE CLOCK: 12.0000 | | | | |
| : | | | | |

Pantalla 31

En la pantalla 31, se observa que todavía no ha llegado el pedido por que apenas estan en el tiempo 12, pero también ya estan en problemas por que ya no pueden satisfacer la demanda y esto origina que no puedan atender ese pedido.

| | | | | |
|-------------------------|-------|-------------------------|-------|---|
| 9440 BYTES AVAILABLE | | | | |
| 560 IN USE | | | | |
| 560 USED (MAX) | | | | |
| : D OUT | | | | |
| RELATIVE CLOCK: 12.0000 | | ABSOLUTE CLOCK: 12.0000 | | |
| BLOCK CURRENT | TOTAL | BLOCK CURRENT | TOTAL | |
| 1 | 12 | 11 | 1 | |
| 2 | 12 | WATCH | 1 | |
| 3 | 12 | 13 | 1 | 1 |
| 4 | 11 | 14 | 0 | |
| TAB | 11 | 15 | 0 | |
| 6 | 11 | | | |
| 7 | 11 | | | |
| TRUBL | 1 | | | |
| 9 | 1 | | | |
| 10 | 0 | | | |
| TABLE LOSSES | | | | |
| : | | | | |

Pantalla 32

| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | NON-W | | |
|------------------|--------------------|--------------------|-----------------------|----------------------|------------------|----------------|
| 11.0000 | 0. | 0. | 0. | | | |
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
| 0. | 11.0000 | 100.00 | 100.00 | 0.00 | 0.0 | 0. |

TABLE STOCK

| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | NON-W | | |
|------------------|--------------------|--------------------|-----------------------|----------------------|------------------|----------------|
| 11.0000 | 47.4545 | 27.7322 | 522.0000 | | | |
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
| 10.0000 | 1.0000 | 9.09 | 9.09 | 90.91 | 0.2107 | -1. |
| 20.0000 | 1.0000 | 9.09 | 18.18 | 81.82 | 0.4215 | -0. |
| 30.0000 | 2.0000 | 18.18 | 36.36 | 63.64 | 0.8422 | -0. |
| 40.0000 | 1.0000 | 9.09 | 45.45 | 54.55 | 0.8429 | -0. |
| 50.0000 | 1.0000 | 9.09 | 54.55 | 45.45 | 1.0536 | 0. |
| 60.0000 | 1.0000 | 9.09 | 63.64 | 36.36 | 1.2644 | 0. |
| 70.0000 | 1.0000 | 9.09 | 72.73 | 27.27 | 1.4751 | 0. |

Pantalla 33

| | | | | | | |
|----------|--------|-------|--------|-------|--------|-----|
| 30.0000 | 2.0000 | 18.18 | 36.36 | 63.64 | 0.8422 | -0. |
| 40.0000 | 1.0000 | 9.09 | 45.45 | 54.55 | 0.8429 | -0. |
| 50.0000 | 1.0000 | 9.09 | 54.55 | 45.45 | 1.0536 | 0. |
| 60.0000 | 1.0000 | 9.09 | 63.64 | 36.36 | 1.2644 | 0. |
| 70.0000 | 1.0000 | 9.09 | 72.73 | 27.27 | 1.4751 | 0. |
| 80.0000 | 1.0000 | 9.09 | 81.82 | 18.18 | 1.6858 | 1. |
| 90.0000 | 1.0000 | 9.09 | 90.91 | 9.09 | 1.8966 | 1. |
| 100.0000 | 1.0000 | 9.09 | 100.00 | -0.00 | 2.1073 | 1. |

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

ROP: 80, ROQ: 100

| RANDOM STREAM | ANTITHETIC VARIATES | INITIAL POSITION | CURRENT POSITION | SAMPLE COUNT | CHI-SQUARE UNIFORMITY |
|---------------|---------------------|------------------|------------------|--------------|-----------------------|
| 1 | OFF | 11 | 23 | 12 | N/A |
| 2 | OFF | 33 | 34 | 1 | N/A |

STATUS OF COMMON STORAGE

9440 BYTES AVAILABLE
560 IN USE
560 USED (MAX)

Pantalla 34

En la pantalla 34, se observa que ya no se observa la variable STOCK, por que ya se agoto este.

| BLOCK CURRENT | TOTAL | INVENTAR.gps | SOURCE CODE |
|---------------|-------|--------------|-----------------------|
| 13 | 1 | | ADVANCE FRS:TIME |
| 14 | 1 | | SAVEVALUE STOCK,XSROQ |
| 15 | 0 | | TRANSFER WATCH |

S/C: OFF ANS CLOCK: 13.0000 REL CLOCK: 13.0000 TTG: 987

XACT: 2 CURBLK: 14 NEXTBLK: 15 CHAINS: CFC PC:

MANY-TIME: 3.0000 MOVE-TIME: 13.0000 PRIORITY: 0

: STEP 1

XACT 14 POISED AT BLOCK 7 (TAB*2). RELATIVE CLOCK: 13.0000
: STEP 1

XACT 14 DESTROYED AT BLOCK 7 (TAB*2). RELATIVE CLOCK: 13.0000
: STEP 1

XACT 2 POISED AT BLOCK 15 (WATCH*3). RELATIVE CLOCK: 13.0000
:

Pantalla 35

En la pantalla 35, se observa que ya se llegó al tiempo 13, esto nos indica que ya llega en este día nuestro pedido pero como llega hasta la tarde, sólo se pueden utilizar estos artículos hasta el día siguiente.

XACT 2 POISED AT BLOCK 15 (WATCH*3). RELATIVE CLOCK: 13.0000
: D OUT

RELATIVE CLOCK: 13.0000 ABSOLUTE CLOCK: 13.0000

| BLOCK CURRENT | TOTAL | BLOCK CURRENT | TOTAL |
|---------------|-------|---------------|-------|
| 1 | 13 | 11 | 1 |
| 2 | 13 | WATCH | 1 |
| 3 | 13 | 13 | 1 |
| 4 | 11 | 14 | 1 |
| TAB | 13 | 15 | 0 |
| 6 | 13 | | |
| 7 | 13 | | |
| TRUBL | 2 | | |
| 9 | 2 | | |
| 10 | 2 | | |

TABLE LOSES

| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | NON-W |
|------------------|---------------|--------------------|------------------|-------|
| 13.0000 | 1.0769 | 3.5931 | 14.0000 | NON-W |

:

Pantalla 36

En la pantalla 36, se observa que en el bloque 8, o sea, en la etiqueta de problemas (TRUBL), tiene un total de 2 esto quiere decir que a pasado la ejecución 2 veces por este lugar lo que nos indica que en este momento ya hay dos demandas que no se han podido atender.

| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
|----------------|-----------------------|---------------------|--------------------------|-------------------------|---------------------|-------------------|
| 0. | 11.0000 | 84.62 | 84.62 | 15.38 | 0. | -0. |
| 1.0000 | 1.0000 | 7.69 | 92.31 | 7.69 | 0.9596 | -0. |
| ... | | | | | | |
| 11.0000 | 1.0000 | 7.69 | 100.00 | 0.00 | 12.0714 | 0. |

TABLE STOCK

| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | NON-ZERO |
|------------------|---------------|--------------------|------------------|----------|
| 13.0000 | 40.1538 | 30.9592 | 522.0000 | NON-ZERO |

| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
|----------------|-----------------------|---------------------|--------------------------|-------------------------|---------------------|-------------------|
| 0. | 2.0000 | 15.38 | 15.38 | 84.62 | 0. | -0. |
| 10.0000 | 1.0000 | 7.69 | 23.08 | 76.92 | 0.2490 | -0. |
| 20.0000 | 1.0000 | 7.69 | 30.77 | 69.23 | 0.4981 | -0. |
| 30.0000 | 2.0000 | 15.38 | 46.15 | 53.85 | 0.7472 | -0. |
| 40.0000 | 1.0000 | 7.69 | 53.85 | 46.15 | 0.5963 | -0. |
| 50.0000 | 1.0000 | 7.69 | 61.54 | 38.46 | 1.2452 | 0. |
| 60.0000 | 1.0000 | 7.69 | 69.23 | 30.77 | 1.4943 | 0. |
| 70.0000 | 1.0000 | 7.69 | 76.92 | 23.08 | 1.7433 | 0. |

Pantalla 37

En la pantalla 37, se observa que ya los valores de los pedidos perdidos son de 1 y 13, lo cual representa un 15.38 de pedidos insatisfechos y un 13% de ventas que no se efectúan.

| | | | | | | |
|----------|--------|-------|--------|-------|--------|-----|
| 30.0000 | 2.0000 | 15.38 | 46.15 | 53.85 | 0.7472 | -0. |
| 40.0000 | 1.0000 | 7.69 | 53.85 | 46.15 | 0.5963 | -0. |
| 50.0000 | 1.0000 | 7.69 | 61.54 | 38.46 | 1.2452 | 0. |
| 60.0000 | 1.0000 | 7.69 | 69.23 | 30.77 | 1.4943 | 0. |
| 70.0000 | 1.0000 | 7.69 | 76.92 | 23.08 | 1.7433 | 0. |
| 80.0000 | 1.0000 | 7.69 | 84.62 | 15.38 | 1.9923 | 0. |
| 90.0000 | 1.0000 | 7.69 | 92.31 | 7.69 | 2.2414 | 0. |
| 100.0000 | 1.0000 | 7.69 | 100.00 | -0.00 | 2.4904 | 0. |

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

| ROP: | 80, | ROQ: | 100, | STOCK: | 100 |
|--------|------------|----------|----------|--------|------------|
| RANDOM | ANTITHETIC | INITIAL | CURRENT | SAMPLE | CHI-SQUARE |
| STREAM | VARIATES | POSITION | POSITION | COUNT | UNIFORMITY |
| 1 | OFF | 11 | 24 | 12 | N/A |
| 2 | OFF | 33 | 34 | 1 | N/A |

STATUS OF COMMON STORAGE

9552 BYTES AVAILABLE
448 IN USE
560 USED (MAX)

Pantalla 38

En la pantalla 38, se observa que ya llegaron los artículos que se pidieron el día 3. Con esto finaliza la transacción número 2.

| BLOCK | CURRENT | TOTAL | INVENTAR.gps | SOURCE | CODE |
|--|-----------|--------------------|--------------------|-------------|-------------------|
| 3 | | 17 | | TEST GE | X\$STOCK,P1,TRUBL |
| 4 | | 15 | | SAVEVALUE | STOCK-,P1 |
| 5 | | 16 | TAB | TABULATE | STOCK |
| 6 | | 16 | | TABULATE | LOSES |
| 7 | | 16 | | TERMINATE | 1 |
| ----- | | | | | |
| S/C: | OFF | ABS CLOCK: 17.0000 | REL CLOCK: 17.0000 | TTG: | 984 |
| ----- | | | | | |
| XACT: | 18 | CURBLK: 4 | NEXTBLK: TAB | CHAINS: CEC | PC: |
| ----- | | | | | |
| MARK-TIME: | 17.0000 | MOVE-TIME: 17.0000 | PRIORITY: | 1 | |
| ----- | | | | | |
| NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE) | | | | | |
| ROP: | 80, | ROQ: | 100, | STOCK: | 77 |
| RANDOM | ANTIETHIC | INITIAL | CURRENT | SAMPLE | CHI-SQUARE |
| STREAM | VARIATES | POSITION | POSITION | COUNT | UNIFORMITY |
| 1 | OFF | 11 | 28 | 17 | N/A |
| 2 | OFF | 33 | 34 | 1 | N/A |

Pantalla 39

En la pantalla 39, se observa en la línea del estatus que hay un reloj absoluto y relativo de 17, mientras nos hace falta 984 días de un total de 1000.

| BLOCK | CURRENT | TOTAL | INVENTAR.gps | SOURCE | CODE |
|---|---------|--------------------|--------------------|-------------|-----------------|
| 11 | | 1 | | GENERATE | ,,,1 |
| 12 | | 2 | WATCH | TEST LE | X\$STOCK,X\$ROP |
| 13 | 1 | 2 | | ADVANCE | PN\$LTIME |
| 14 | | 1 | | SAVEVALUE | STOCK*,X\$ROQ |
| 15 | | 1 | | TRANSFER | ,WATCH |
| ----- | | | | | |
| S/C: | OFF | ABS CLOCK: 17.0000 | REL CLOCK: 17.0000 | TTG: | 983 |
| ----- | | | | | |
| XACT: | 2 | CURBLK: 13 | NEXTBLK: 14 | CHAINS: FEC | PC: |
| ----- | | | | | |
| MARK-TIME: | 3.0000 | MOVE-TIME: 24.0000 | PRIORITY: | 0 | |
| ----- | | | | | |
| : STEP 1 | | | | | |
| XACT 18 DESTROYED AT BLOCK 7 (TAB+2). RELATIVE CLOCK: 17.0000 | | | | | |
| : STEP 1 | | | | | |
| XACT 2 POISED AT BLOCK 13 (WATCH+1). RELATIVE CLOCK: 17.0000 | | | | | |
| : STEP 1 | | | | | |
| XACT 2 PLACED ON FEC AT BLOCK 13 (WATCH+1). RELATIVE CLOCK: 17.0000 | | | | | |
| : | | | | | |

Pantalla 40

En la pantalla 40, se observa que el reloj relativo esta en la unidad 17 estando la transacción 2.

Una manera de pasar a una transacción definida es pulsando en la línea de comando una trampa, esto se hace con el comando "TRAP" y nombre "XACT" de transacción, el signo igual y el número de transacción al que queremos llegar, después ejecutamos el comando "run", esto se observa en la pantalla 41.

| BLOCK CURRENT | TOTAL | INVENTAR.gps | SOURCE CODE |
|---|--------------------|--------------------|-----------------------------|
| 1 | 22 | GENERATE | 1,,,1 |
| 2 | 22 | ASSIGN | 1,VSDMND |
| 3 | 22 | TEST GE | XSSTOCK,P1,TRUBL |
| ----- | | | |
| S/C: OFF | | ABS CLOCK: 23.0000 | REL CLOCK: 23.0000 TTG: 978 |
| ----- | | | |
| XACT: 24 | CURBLK: "BIRTH" | NEXTBLK: 1 | CHAINS: CEC PC: |
| ----- | | | |
| MARK-TIME: ---- | MOVE-TIME: 23.0000 | PRIORITY: 1 | |
| ----- | | | |
| : STEP 1 | | | |
| XACT 2 PLACED ON FEC AT BLOCK 13 (WATCH+1). RELATIVE CLOCK: 17.0000 | | | |
| : T XACT=24 | | | |
| XACT 24 HAS NOT YET BEEN CREATED. | | | |
| REQUEST QUEUED. | | | |
| : R | | | |
| XACT 24 POISED AT BLOCK 1. RELATIVE CLOCK: 23.0000 | | | |
| : | | | |

Pantalla 41

| | | | |
|--|---------------|-------------------------|------------------|
| XACT 2 POISED AT BLOCK 15 (WATCH+3). RELATIVE CLOCK: 24.0000 | | | |
| : D OUT | | | |
| RELATIVE CLOCK: 24.0000 | | ABSOLUTE CLOCK: 24.0000 | |
| BLOCK CURRENT | TOTAL | BLOCK CURRENT | TOTAL |
| 1 | 24 | 11 | 1 |
| 2 | 24 | WATCH | 2 |
| 3 | 24 | 13 | 2 |
| 4 | 22 | 14 | 2 |
| TAB | 24 | 15 | 1 |
| 6 | 24 | | |
| 7 | 24 | | |
| TRUBL | 2 | | |
| 9 | 2 | | |
| 10 | 2 | | |
| TABLE LOSSES | | | |
| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS |
| 24.0000 | 0.5833 | 2.6526 | 14.0000 NON-W |
| : | | | |

Pantalla 42

En la pantalla 24, se observa que ya se han pasado 24 ejecuciones por los bloques del 1 al 7, quedando los demás como estaban.

| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
|-------------|--------------------|------------------|-----------------------|----------------------|------------------|----------------|
| 0. | 22.0000 | 91.67 | 91.67 | 8.33 | 0. | -0. |
| 1.0000 | 1.0000 | 4.17 | 95.83 | 4.17 | 1.7143 | 0. |
| 13.0000 | 1.0000 | 4.17 | 100.00 | -0.00 | 22.2857 | 4. |

TABLE STOCK

| ENTRIES IN TABLE | MEAN ARGUMENT | STANDARD DEVIATION | SUM OF ARGUMENTS | NON-W |
|------------------|---------------|--------------------|------------------|-------|
| 24.0000 | 47.5000 | 29.9304 | 1140.0000 | |

| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
|-------------|--------------------|------------------|-----------------------|----------------------|------------------|----------------|
| 0. | 2.0000 | 8.33 | 8.33 | 91.67 | 0. | -1. |
| 10.0000 | 1.0000 | 4.17 | 12.50 | 87.50 | 0.2105 | -1. |
| 20.0000 | 2.0000 | 8.33 | 20.83 | 79.17 | 0.4211 | -0. |
| 30.0000 | 3.0000 | 12.50 | 33.33 | 66.67 | 0.6316 | -0. |
| 40.0000 | 3.0000 | 12.50 | 45.83 | 54.17 | 0.8421 | -0. |
| 50.0000 | 2.0000 | 8.33 | 54.17 | 45.83 | 1.0526 | 0. |
| 60.0000 | 2.0000 | 8.33 | 62.50 | 37.50 | 1.2632 | 0. |
| 70.0000 | 2.0000 | 8.33 | 70.83 | 29.17 | 1.4737 | 0. |

Pantalla 43

| | | | | | | |
|----------|--------|-------|--------|-------|--------|-----|
| 30.0000 | 3.0000 | 12.50 | 33.33 | 66.67 | 0.6316 | -0. |
| 40.0000 | 3.0000 | 12.50 | 45.83 | 54.17 | 0.8421 | -0. |
| 50.0000 | 2.0000 | 8.33 | 54.17 | 45.83 | 1.0526 | 0. |
| 60.0000 | 2.0000 | 8.33 | 62.50 | 37.50 | 1.2632 | 0. |
| 70.0000 | 2.0000 | 8.33 | 70.83 | 29.17 | 1.4737 | 0. |
| 80.0000 | 2.0000 | 8.33 | 79.17 | 20.83 | 1.6842 | 1. |
| 90.0000 | 3.0000 | 12.50 | 91.67 | 8.33 | 1.8947 | 1. |
| 100.0000 | 2.0000 | 8.33 | 100.00 | -0.00 | 2.1053 | 1. |

NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE)

| ROP: | BC, | ROQ: | 100, | STOCK: | 113 |
|---------------|---------------------|------------------|------------------|--------------|-----------------------|
| RANDOM STREAM | ANTITHETIC VARIATES | INITIAL POSITION | CURRENT POSITION | SAMPLE COUNT | CHI-SQUARE UNIFORMITY |
| 1 | OFF | 11 | 35 | 24 | N/A |
| 2 | OFF | 33 | 35 | 2 | N/A |

STATUS OF COMMON STORAGE

9552 BYTES AVAILABLE
 448 IN USE
 560 USED (MAX)

Pantalla 44

Con esta últimas 3 pantallas nos percatamos que realizo al momento de llegar hasta la instrucción que le dimos a ejecutar, o sea la transacción 24.

```

BLOCK CURRENT      TOTAL      INVENTAR.gps SOURCE CODE
-----
5                1000      TAB      TABULATE  STOCK
6                1000      TABULATE  LOSSES
7                1000      TERMINATE 1
8                99       TRUBL     ASSIGN   2,V$LOST
9                99       SAVEVALUE STOCK,0
-----
S/C: OFF ABS CLOCK: 1000.0000 REL CLOCK: 1000.0000 TTG: 0
-----
XACT: 1001  CURBLK: 6  NEXTBLK: 7  CHAINS:  PC:

MARK-TIME: 1000.0000 MOVE-TIME: 1000.0000 PRIORITY: 1
-----

STATUS OF COMMON STORAGE

  9552 BYTES AVAILABLE
   448 IN USE
   560 USED (MAX)

: RUN

REQUESTING OUTPUT IN (CONTROL) STATEMENT NUMBER 39. RELATIVE CLOCK: 1000.0000
:
    
```

Pantalla 45

Quando queremos ya finalizar la ejecución del programa, lo podemos hacer ejecutando en la línea de comandos "RUN", también se puede abreviar con la inicial, este llega a su fin siempre y cuando no tenga algún "trap" es decir una trampa para detener su ejecución antes de tiempo. Las siguientes pantallas nos muestran como finaliza este programa el cual también lo podemos observar en el archivo "inventar.lis" que se genera al ejecutar el lenguaje gpash.

```

RANDOM      ANTIHETIC      INITIAL      CURRENT      SAMPLE      CHI-SQUARE
STREAM     VARIATES      POSITION      POSITION      COUNT      UNIFORMITY
  1         OFF          11          1011         1000        0.86
  2         OFF          33          122          89         0.40

STATUS OF COMMON STORAGE

  9552 BYTES AVAILABLE
   448 IN USE
   560 USED (MAX)

: D BLO

BLOCK CURRENT      TOTAL      BLOCK CURRENT      TOTAL
1                1000      11                  1
2                1000      WATCH              89
3                1000      13                  89
4                901       14                  88
TAB              1000      15                  88
5                1000
6                1000
7                1000
TRUBL            99
9                99
10              99
:
    
```

Pantalla 46

Con esta pantalla observamos que se efectuaron 89 pedidos en 1000 días de los cuales han llegado 88.

TABLE LOSSES

| ENTRIES IN TABLE 1000.0000 | MEAN ARGUMENT 0.6950 | STANDARD DEVIATION 2.3887 | SUM OF ARGUMENTS 695.0000 | NON-W | | |
|-------------------------------|-------------------------|------------------------------|------------------------------|----------------------|------------------|----------------|
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
| 0. | 901.0000 | 90.10 | 90.10 | 9.90 | 0. | -0. |
| 1.0000 | 10.0000 | 1.00 | 91.10 | 8.90 | 1.4388 | 0. |
| 2.0000 | 5.0000 | 0.50 | 91.60 | 8.40 | 2.8777 | 0. |
| 3.0000 | 5.0000 | 0.50 | 92.10 | 7.90 | 4.3165 | 0. |
| 4.0000 | 6.0000 | 0.60 | 92.70 | 7.30 | 5.7554 | 1. |
| 5.0000 | 12.0000 | 1.20 | 93.90 | 6.10 | 7.1942 | 1. |
| 6.0000 | 6.0000 | 0.60 | 94.50 | 5.50 | 8.6331 | 2. |
| 7.0000 | 9.0000 | 0.90 | 95.40 | 4.60 | 10.0719 | 2. |
| 8.0000 | 6.0000 | 0.60 | 96.00 | 4.00 | 11.5108 | 3. |
| 9.0000 | 9.0000 | 0.90 | 96.90 | 3.10 | 12.9496 | 3. |
| 10.0000 | 11.0000 | 1.10 | 98.00 | 2.00 | 14.3885 | 3. |
| 11.0000 | 6.0000 | 0.60 | 98.60 | 1.40 | 15.8273 | 4. |
| 12.0000 | 10.0000 | 1.00 | 99.60 | 0.40 | 17.2662 | 4. |
| 13.0000 | 4.0000 | 0.40 | 100.00 | 0.00 | 18.7050 | 5. |

Pantalla 47

La tabla de perdidas nos refleja que del total de pedidos no se pudieron atender un 9.90% del total.

| | | | | | | |
|---------|--------|------|--------|------|---------|----|
| 13.0000 | 4.0000 | 0.40 | 100.00 | 0.00 | 18.7050 | 5. |
|---------|--------|------|--------|------|---------|----|

TABLE STOCK

| ENTRIES IN TABLE 1000.0000 | MEAN ARGUMENT 45.4810 | STANDARD DEVIATION 31.1088 | SUM OF ARGUMENTS 45481.0000 | NON-W | | |
|-------------------------------|--------------------------|-------------------------------|--------------------------------|----------------------|------------------|----------------|
| UPPER LIMIT | OBSERVED FREQUENCY | PERCENT OF TOTAL | CUMULATIVE PERCENTAGE | CUMULATIVE REMAINDER | MULTIPLE OF MEAN | DEVIATION FROM |
| 0. | 105.0000 | 10.50 | 10.50 | 89.50 | 0.2199 | -1. |
| 10.0000 | 76.0000 | 7.60 | 18.10 | 81.90 | 0.4397 | -0. |
| 20.0000 | 89.0000 | 8.90 | 27.00 | 73.00 | 0.6596 | -0. |
| 30.0000 | 95.0000 | 9.50 | 36.50 | 63.50 | 0.8795 | -0. |
| 40.0000 | 93.0000 | 9.30 | 45.80 | 54.20 | 1.0994 | 0. |
| 50.0000 | 99.0000 | 9.90 | 55.70 | 44.30 | 1.3192 | 0. |
| 60.0000 | 88.0000 | 8.80 | 64.50 | 35.50 | 1.5391 | 0. |
| 70.0000 | 89.0000 | 8.90 | 73.40 | 26.60 | 1.7590 | 1. |
| 80.0000 | 93.0000 | 9.30 | 82.70 | 17.30 | 1.9788 | 1. |
| 90.0000 | 88.0000 | 8.80 | 91.50 | 8.50 | 2.1987 | 1. |
| 100.0000 | 67.0000 | 6.70 | 98.20 | 1.80 | | 1. |
| OVERFLOW | 18.0000 | 1.80 | 100.00 | 0.00 | | |

AVERAGE VALUE OF OVERFLOW IS 106.7222

Pantalla 48

En esta tabla observamos que hubieron 45,481 pedidos que se pudieron atender, y de la tabla anterior tenemos que 695 no se pudieron atender, o sea que se tuvo un total de 46,176.

| | | | | | | |
|--|------------------------|---------------------|---------------------|-----------------|--------------------------|----|
| 60.0000 | 88.0000 | 8.80 | 64.50 | 35.50 | 1.3192 | 0. |
| 70.0000 | 89.0000 | 8.90 | 73.40 | 26.60 | 1.5391 | 0. |
| 80.0000 | 93.0000 | 9.30 | 82.70 | 17.30 | 1.7590 | 1. |
| 90.0000 | 88.0000 | 8.80 | 91.50 | 8.50 | 1.9788 | 1. |
| 100.0000 | 67.0000 | 6.70 | 98.20 | 1.80 | 2.1987 | 1. |
| OVERFLOW | 18.0000 | 1.80 | 100.00 | 0.00 | | |
| AVERAGE VALUE OF OVERFLOW IS | | 106.7222 | | | | |
| NON-ZERO FULLWORD SAVEVALUES: (NAME : VALUE) | | | | | | |
| ROP: | 80, | ROQ: | 100, | STOCK: | 40 | |
| RANDOM STREAM | ANTITHETIC VARIATES | INITIAL POSITION | CURRENT POSITION | SAMPLE COUNT | CHI-SQUARE UNIFORMITY | |
| 1 | OFF | 11 | 1011 | 1000 | 0.86 | |
| 2 | OFF | 33 | 122 | 89 | 0.40 | |
| STATUS OF COMMON STORAGE | | | | | | |
| 9552 BYTES AVAILABLE | | | | | | |
| 448 IN USE | | | | | | |
| 560 USED (MAX) | | | | | | |
| : | | | | | | |

Pantalla 49

Con esta última pantalla nos damos cuenta de que se finalizó con un inventario de cuarenta unidades. También se observa la uniformidad que tuvieron los dos generadores de números aleatorios, la cual refleja que para la demanda fué de 0.86 y el tiempo de entrega de los artículos de 0.40.

Con esto finalizamos un programa interactivo en el lenguaje GPSSH, se expusieron los comandos, que pueden ser de más ayuda al usuario de la simulación para que se familiarice con la terminología y con los comandos del lenguaje, para ahondar en los comandos del GPSSH, consulte la obra de Schriber, 1990.

BIBLIOGRAFIA

- 1 ARIMA, J.
"Analogy By Simulation -A Weak Justification Method"
 ICOT RESEARCH CENTER TOKYO, JAPON
 TECHNICAL MEMORANDUM EDICION 1990, 5 p.
 INSTITUTE FOR NEW GENERATION COMPUTER T.

 - 2 DAELLENBACH, Hans G. & GEORGE, Jhon A.
"Introducción A Técnicas De Investigación De Operaciones"
 CECSA MEXICO
 PRIMERA EDICION 1990, 711 p.

 - 3 FREY, Jeffrey (University of Maryland)
"Fed Journal "Device Simulatjon For New Century"
 JOURNAL TOKYO, JAPON
 EDICION 1992, 8 p.
- 2
- 4 FUDENBERG, Drew & TIROLE, Jean
"Game Theory"
 LIBRARY OF CONGRESS MASSACHUSETTS
 SEGUNDA EDICION 1992, 579 p.
 MASSACHUSSETTS INSTITUTE OF TECHNOLOGY

 - 5 GARCIA, Ricardo V.
"La Simulación Mediante Computadora En La Administración De Empresas"
 TESIS U.N.A.M. MEXICO, D.F.
 EDICION 1990, 90 p.

 - 6 GARZA, Tomas
"Elementos De Calculo"
 TEXTOS UNIVER. UNAM MEXICO
 PRIMERA EDICION 1990, 191 p.

 - 7 GORDON, Geoffrey
"Simulación de Sistemas"
 DIANA MEXICO
 PRIMERA EDICION 1989, 344 p.

 - 8 GORDON, Goeffrey
"System Simulation"
 PRETINCE HALL ENGLEWOOD, NEW JERSEY
 SEGUNDA EDICION 1978, 324 p.

- 9 GOULD F. J. & EPPEN G. D.
"Investigación De Operaciones En La Ciencia Administrativa"
 PRETINCE HALL MEXICO
 PRIMERA EDICION 1989, 783 p.
- 10 IMAI, A. & TICK E.
"A Shared-memory Multiprocess Or Garbage Collector And Its Evaluation For Logic"
 ICOT RESECH CENTER TOKYO, JAPON
 TECHNICAL REPORT EDICION 1991, 17 p.
 INSTITUTE FOR NEW GENERATION COMPUTER T.
- 11 KONAGAYA A. & YAMANISHI K. (NEC)
"Stochastic Decision Predicates: A Schema To Represent Motifs"
 ICOT RESEARCH CENTER TOKYO, JAPON
 TECHNICAL REPORT EDICION 1991, 7 p.
 INSTITUTE FOR NEW GENERATION COMPUTER T.
- 12 KUME, Hitoshi
"Statistical Methods For Quality Improvement"
 AOTS TOKYO, JAPON
 OCTAVA EDICION 1991, 231 p.
 THE ASSOCIATION FOR OVERSEAS TECHNICAL S
- 13 MCMILLAN C., GONZALEZ R. & SCHRIBER T.J.
"Análisis de Sistemas"
 TRILLAS MEXICO
 TERCERA EDICION 1986, 654 p.
- 14 MENDENHALL, William & REINMUTH, James E.
"Estadística Para Administración Y Economía"
 GPO E. IBEROAMERICA MEXICO
 PRIMERA EDICION 1988, 707 p.
- 15 MENJU S., K. SAKAI, Y. SATO & A. AIBA
"A Study On Boolean Constraint Solvers"
 ICOT RESEARCH CENTER TOKYO, JAPON
 TECHNICAL MEMORANDUM EDICION 1991, 9 p.
 INSTITUTE FOR NEW GENERATION COMPUTER T.
- 16 MORA, Jose Luis
"Investigación de Operaciones e Informática"
 TRILLAS MEXICO
 SEGUNDA EDICION 1986, 367 p.

- 17 MORENO, Antonio G.
"Presentación Gráfica de Resultados de La Simulación De
 Redes Neuronales"
 TESIS U.N.A.M. MEXICO, D.F.
 EDICION 1982, 0 p.
- 18 PARDO, Leandro & VALDES, Teofilo
"Simulación Aplicaciones Prácticas En La Empresa"
 GESTION INFORMATIZA. ESPAÑA
 PRIMERA EDICION 1987, 293 p.
 PROGRAMAS EN BASIC PARA SIMULAR
- 19 PRAWDA, Juan
"Métodos y Modelos de Investigación de Operaciones"
 LIMUSA NORIEGA MEXICO
 NOVENA EDICION 1990, 935 p. 1
- 20 PRAWDA, Juan
"Métodos y Modelos de Investigación de Operaciones"
 LIMUSA NORIEGA MEXICO
 SEXTA EDICION 1991, 1026 p. 2
- 21 RASCON, Octavio A.
"Introducción A La Teoría De Probabilidades"
 TEXTOS PROG. UNAM MEXICO D.F.
 CUARTA EDICION 1988, 433 p.
- 22 S. BUFFA, Elwood S.
"Administración Y Dirección Técnica De La Producción"
 LIMUSA NORIEGA MEXICO
 SEXTA EDICION 1989, 671 p.
- 23 SCHMIDT, J. W. & TAYLOR, R. E.
"Análisis Y Simulación De Sistemas Industriales"
 TRILLAS MEXICO
 PRIMERA EDICION 1979, 651 p. 1
- 24 SCHRIBER, Thomas J.
"An Introduction To Simulation Using Gpss/h"
 JOHN WILEY & SONS NEW YORK, U.S.A.
 CUARTA EDICION 1990, 423 p. 1
- 25 SHAMBLIN, James E. & STEVENS, G. T. JR.
"Investigación de Operaciones"
 MCGRAW-HILL/INTERAM. MEXICO
 PRIMERA EDICION 1988, 423 p.

- 26 SHANON, Robert E.
"Simulación de Sistemas"
TRILLAS
PRIMERA EDICION 1988, 428 p.
DISEÑO DESARROLLO E IMPLANTACION
- 27 SIMON, Nadima D.
"Diseños de Muestreo, Un Enfoque Administrativo"
FONDO EDITORIAL FCA MEXICO, D.F.
PRIMERA EDICION 1987, 165 p.