



61
2 ej

**UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO**

FACULTAD DE INGENIERIA

**ADECUACION DEL GPSS PARA
TEMAS DE PLANEACION Y
CONTROL DE LA PRODUCCION**

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA
(AREA DE INGENIERIA INDUSTRIAL)

P R E S E N T A N

**MIGUEL ANGEL GALVAN SANCHEZ
RICARDO ANGEL MAZA PACHECO**

México, D. F. 1991

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.

INDICE

	Pág.
INTRODUCCION	1
CAPITULO I FUNDAMENTOS TEORICOS	6
1.1 Variables Estocásticas	6
1.2 Procesos Determinísticos y Estocásticos	6
1.3 Funciones de Probabilidad	10
1.3.1 Funciones de Probabilidad Continua	12
CAPITULO II SIMULACION	16
2.1 Modelo de Simulación	16
2.2 Modelo Físico Dinámico	17
2.3 Modelo Matemático Dinámico	17
2.3.1 Atributo	17
2.4 Naturaleza Experimental de la Simulación	19
2.4.1 Análisis del Sistema	19
2.4.2 Estudio del Diseño del Sistema	19
2.4.3 Postulación del Sistema	20
2.5 Pasos Involucrados en los Estudios de Simulación	20
2.6 Lenguajes de Simulación	22
2.6.1 Algol, Fortran, Cobol y Pli	23
2.6.2 GASP (General Activity Simulation Program)	23
2.6.3 GPSS	23
2.6.4 Simscript	23
2.6.5 Dynamo	24
CAPITULO III INTRODUCCION AL GPSS	25
3.1 Descripción General	25
3.2 Tiempos de Acción	27
3.3 Sucesión de Eventos	29
3.4 Elección de Caminos	30
3.5 Facilidades y Almacenajes	31
3.6 Recolección de Estadísticas	33
CAPITULO IV PROGRAMACION DEL GPSS	36
4.1 Simulación de un Supermercado	36
4.2 Simulación de un Sistema Telefónico Simple	43
CAPITULO V APLICACIONES DEL SISTEMA GPSS.....	49
5.1 Simulación de un Taller de Manufactura	49
5.2 El Sistema	52
5.3 Requerimientos del Modelo	53
5.4 Características Numéricas	53
5.5 Modelo en el GPSS	56
5.5.1 Correspondencia entre los Elementos del Sistema y las Entidades del GPSS	56
5.6 Comprensión del Modelo	57
5.6.1 El Modelo de Programación del Trabajo	58
5.6.2 Submodelo de Proceso del Trabajo	59
5.6.3 Submodelo de la Creación del Trabajo	60
5.6.4 Descripción del Programa	61
5.6.4.1 Almacenajes	61
5.6.4.2 Parámetros	61
5.6.4.3 Funciones	63

	Pág.
5.6.4.4 Variable	63
5.6.4.4.1 Variables Bulianas	65
5.6.4.4.2 Palabras de Salvamento	66
5.6.4.4.3 Tablas	66
5.6.5 Comentarios al Listado del Programa	66
5.6.6 Experimentos y Resultados	68
5.6.7 Conclusiones	75
CAPITULO VI ANEXOS	77
ANEXO 1	78
ANEXO 2	82
ANEXO 3	84
ANEXO 4	86
ANEXO 5	95
CAPITULO VII CONCLUSIONES	104
BIBLIOGRAFIA	107

INTRODUCCION

El sistema general de simulación propuesto, GPSS (siglas en idioma Inglés), fué desarrollado por Geoffrey Gordon y presentado en dos partes, en 1961 y en 1962. La primera realización del lenguaje GPSS, frecuentemente llamada "Simulación de Gordon", fué implementada en una máquina IBM 704, 709 y 7090.

El lenguaje fué diseñado para ser usado por personas que no fueran especialistas en sistemas. El uso de diagramas de flujo para describir un sistema es bien conocido, así que el GPSS es estructurado como un lenguaje en el cual se utilizan los diagramas de bloque. Esta filosofía permite a los analistas el someter un modelo de computadora en forma de red de bloques, conectados en un mismo orden y secuencia de eventos.

Una disposición de 25 bloques específicos está definida, cada representación es un sistema básico de acción y se asocia con un tiempo de ejecución y de acción. Los bloques se pueden usar repetidamente en varias ocasiones; la interpretación de una acción representada por una operación en bloque es responsabilidad del analista, y el tiempo de ejecución y de acción también se especifica por la misma persona.

Unidades generales de tráfico en el sistema, como pueden ser las personas, los vehículos, los mensajes o aviones son representadas por transacciones, las cuales estan definidas en un modelo dinámico componente de un sistema. Las transacciones que se mueven a través de un diagrama de bloques bajo el control de los mismos bloques son creados y destruidos a requerimiento del analista.

El concepto fundamental de la estructura del diagrama de bloque y las transacciones han sido retenidas por todas partes durante la evolución del GPSS.

Una segunda versión, GPSS II, es un lenguaje más potente que sigue la misma filosofía que la versión original. Para el GPSS III, sin embargo, se realizó un mayor cambio en su diseño. Los tiempos de demoras fueron eliminados de los bloques y fueron puestos en bloque en particular, llamado ADVANCE, el cual fue usado en el modelo para todos los tiempos de paso. Consecuentemente, toda la estructura del bloque fue definida y la acción del sistema fue representada por tres bloques, los cuales especifican la acción de comienzo, la de retardo y la finalización de la acción, respectivamente.

Estos nuevos bloques son menos complicados que sus equivalentes en la versión original. Esta versión tiene las características de ser más simple y más potente que la

original. Muchas mejoras más fueron hechas, incluyendo un mejor control sobre las ramificaciones, un diagnóstico más completo de los mensajes y la introducción de la cadena de usuarios. La cadena de usuarios da a los analistas más control sobre el comportamiento de las transacciones individuales y puede ser usada también para incrementar la velocidad de la simulación. La estructura básica del lenguaje permanece sin alteración desde la versión del GPSS III y esta realización es más compatible.

En 1967, el GPSS/160 se anunció con el nombre de GPSS. Esta versión tuvo muchas características, entre las que sobresalen: matrices, almacenjes de uno o dos arreglos dimensionales, la selección por tener prioridad y grupos, los que permitieron al usuario direccionar y manipular todas las transacciones con las características dadas. La capacidad del número aleatorio fue extendida para dar la generación de 6 eventos independientes de números aleatorios, y las variables bullanas se introdujeron.

Para simplificar el código ciertos componentes del modelo fueron identificados con un nombre, el número de parámetros que se asocian con una transacción fueron incrementados, los macros pudieron ser definidos para representar líneas de un código común o una simple declaración y las rutinas HELP se usaron como interfase con el lenguaje ensamblador si las instrucciones de la operación

que fueran más poderosas así lo requieran. Un editor de salida fué dispuesto para que el usuario pudiera presentar resultados en una forma más conveniente, incluyendo, por ejemplo, las gráficas o los histogramas.

Dos realizaciones adicionales han sido hechas, una segunda versión del GPSS/360 y el GPSS V. Dentro de las mejoras del GPSS V podemos mencionar, que los cambios directamente simplifican las tareas de la rutina de modelar y flexibilizan las restricciones en el tamaño de los modelos. Un tiempo de corrida puede ser adecuado para limitar el total del tiempo usado de ejecución para la operación. Cuando el tiempo de corrida termina antes de la simulación, se produce una salida y la corrida de la simulación es terminada.

El bloque HELP ha sido extendido para que los analistas tengan una simple interacción con lenguajes como el Fortran o el PL/1. Las formas libres del código reducen la labor de la preparación del programa. Nuevos tipos de parámetros y arreglos dan mayor flexibilidad al el manejo de datos en el modelo y el número de parámetros puede estar asociado con cada transacción que ha sido incrementada significativamente. Desde su comienzo, el GPSS ha mantenido el objetivo al gusto del analista.

La única mejora en la estructura del lenguaje es incrementar su poder y hacerlo accesible pero reteniendo

fundamentalmente su filosofía de manejarse bajo el concepto de diagrama de bloques. Otras características han sido incorporadas al lenguaje, como el reducir el esfuerzo del modelaje sin complicar la etapa del código o incrementar el alcance y el detalle de los modelos. En suma, el GPSS es particularmente una versión muy bien situada para el modelaje de tráfico y sistemas en línea, situaciones en las cuales es necesario estudiar las interacciones de una situación dinámica al detalle.

El objetivo de este trabajo es presentar un panorama del paquete "GPSS" adecuado a temas de Planeación y Control de la Producción, sin pretender exponer conceptos e ideas concluyentes ya que en este campo existirán paquetes más adecuados a ciertos tipos de situaciones. Consecuentemente, los conceptos e ideas aquí expresados serán susceptibles de mejora. El trabajo presenta los siguientes apartados:

Fundamentos teóricos en que se exponen de manera breve las bases matemáticas necesarias para el desarrollo del paquete; **Simulación de sistemas** el cual nos muestra aspectos generales de la simulación de sistemas; **Introducción al GPSS** con el cual se tocan conceptos generales de la estructura del programa; **Programación del GPSS** con el que indica los pasos a seguir para la realización del sistema a simular y una breve **aplicación del sistema GPSS** a temas de Planeación y Control de la Producción.

CAPITULO I

FUNDAMENTOS TEORICOS

1.1 Variables Estocásticas

Se define como una función real que asocia un elemento numérico con cada uno de los eventos elementales del espacio muestral. En la práctica, muchos sistemas incluyen actividades que se comportan aleatoriamente, en cuyo caso no se conoce la secuencia exacta de los eventos. A veces la actividad no es intrínsecamente aleatoria por su naturaleza, aunque no se conoce la información completa relativa a su comportamiento y se debe describir como una actividad aleatoria. Posiblemente se disponga o pueda obtenerse la información completa, pero puede ser tan compleja que sea más conveniente describir la actividad como aleatoria. Aunque no se conoce la secuencia exacta de valores que toma una variable estocástica se puede conocer o suponer, conocido el rango de valores en que puede variar, así como la probabilidad con que tomará los valores. En consecuencia las variables estocásticas se estudian en términos de funciones que describen la probabilidad de la variable que toma diversos valores.

1.2 Procesos Determinísticos y Estocásticos

Un proceso es definido como "determinístico", si por cada valor de entrada la salida puede ser determinada.

Por ejemplo: Si la ecuación $y=x+4$, si la consideramos como un proceso, entonces supongamos: $x=3$, por lo tanto $y=7$; la fórmula verdadera para todos los valores de x .

En contraste, un proceso "estocástico es aleatorio, cuando observaciones suficientes se han realizado, será posible encontrar la distribución estadística que gobierna al sistema. Por ejemplo: ¿Cuánta gente se contaría en los autos que has visto?

El número probablemente caiga en un rango de 0 a 6; no es posible ser más exactos.

Los sistemas pueden ser estudiados ya sea "Estocásticamente o Determinísticamente". Sus modelos pueden ser ambos.

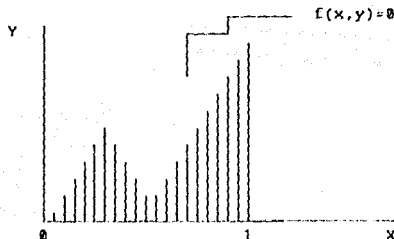
Se tienen las siguientes combinaciones:

SISTEMA		MODELO
	1	
Determinístico		Determinístico
	2	3
Estocástico	4	Estocástico (modelo de Monte Carlo.)

caso 1.- Las ecuaciones matemáticas de movimiento de misiles. El ajuste de movimientos de misiles. El ajuste de ecuaciones que se construyen del modelo, puede ser resuelto analíticamente para producir el sistema.

caso 2.- Se considera la evaluación de una integral definida por medio de un proceso "Estocástico".

EJEMPLO: Estimar el área bajo la curva $f(x,y)=0$ para el rango $0 \leq x \leq 1$ de la siguiente figura:



El método de Monte Carlo establece que si existe una probabilidad igual de selección de cualquier punto en el cuadro unitario mostrado en la figura, y si un número de puntos suficientes son relacionados de una manera aleatoria, entonces el área requerida esta dada por: $A = \lim_{N \rightarrow \infty} \frac{n}{N}$

donde N = Número total de puntos

n = Numero de puntos que caen en el área requerida.

Al efectuar este experimento es necesario generar pares de números aleatorios en el rango de 0-1 tal que todos los puntos en el cuadro unitario ocurren con igual probabilidad. Aunque el método no se aplica tan primitivamente, el ejemplo ilustra el concepto básico del Método de Monte Carlo.

caso 3.- El problema es estocástico, pero el modelo es determinístico, por ejemplo, la generación de números aleatorios por computadora. La probabilidad de este proceso fué analizada en el caso 2.

caso 4.- Tanto el sistema como el modelo son estocásticos. La simulación es utilizada normalmente.

Algunos ejemplos con variables aleatorias son dados en la siguiente tabla; muchos problemas de la vida real son naturalmente estocásticos y no pueden ser completamente descritos con un modelo analítico. Muchos estudios en Investigación de Operaciones están dirigidos para encontrar una estrategia óptima, tal como minimizar costos de operación o maximizar rendimiento. La simulación usualmente no optimiza, pero puede ser usada para seleccionar estrategias disponibles de varias alternativas.

PROBLEMAS (SISTEMAS)	VARIABLE ALEATORIA
• Inventarios	• Demanda
• Taller	• Tiempo de proceso, tiempo interfallas.
• Problemas de reemplazo	• Tiempo de vida, paros súbitos de máquina.

PROBLEMAS (SISTEMAS)	VARIABLE ALEATORIA
• Sistemas tiempo real	• Tráfico.
• Tráfico	• Llegadas y velocidad de vehículos, tiempo de reacción.
• Negocios y Juegos de guerra	• Decisiones de participantes, parámetros que describen factores económicos.

1.3 Funciones de Probabilidad

En algunos casos, es posible generalizar las descripciones probabilísticas de los fenómenos y, por tanto definir una ley del comportamiento probabilístico. Como en el caso de cualquier ley física, la formulación presupone

algo acerca del comportamiento del fenómeno que trata de explicar. Diversas leyes probabilísticas han resultado lo bastante útiles en la práctica como para recibir nombres especiales. A continuación se mencionan algunas de ellas:

Si una variable estocástica puede tomar n distintos valores x_i (donde $i=1,2,3,\dots,n$) y la probabilidad de que se tome el valor x_i es P_i , se dice que el conjunto de números P_i (donde $i=1,2,3,\dots,n$) es una función de probabilidad discreta. Ya que la variable siempre debe de tomar uno de los valores P_i , se sigue que la $\sum_i P_i = 1$. (1)

Sin embargo, es igualmente posible que en ciertos experimentos la probabilidad de ocurrencia no tenga una distribución uniforme, sino ésta sea una función del valor de la variable aleatoria. Esta función que se designa con $f(x)$ se llama de densidad de probabilidad y permite calcular la probabilidad de que una variable aleatoria tenga un valor $a < x < b$, a saber: $p(a < x < b) = \int_a^b f(x) dx$. (2)

Si se tiene presente que todos los valores posibles de x de la variable aleatoria X son: $-\infty < x < \infty$, resulta que la $p(-\infty < x < \infty) = 1$, sustituyendo en (2), se tiene:

$$p(-\infty < x < \infty) = \int_{-\infty}^{\infty} f(x) dx = 1.$$

Esta es una importante propiedad de la función de distribución y el equivalente para distribuciones continuas de (1).

Para funciones continuas se define la función de distribución acumulada como: $F(t) = \int_{-\infty}^t f(t) dt$.

De acuerdo con esta definición la función de distribución de probabilidad $F(t)$, representa la probabilidad de que el valor x de la variable aleatoria X se encuentre entre $-\infty$ y t . Por lo tanto: $F(t) = p(x < t)$.

La media o valor esperado de una distribución esta dada por: $m = \int_{-\infty}^{\infty} x f(x) dx$.

En el caso de una distribución discreta, la media se evalúa de la fórmula: $m = \sum_j x_j p(x_j)$, en que se toma la suma en todos los valores de x en que se define la distribución.

1.3.1 Funciones de Probabilidad Continua

Una variable aleatoria es continua si su dominio de definición es el conjunto de los números reales; es decir, cuando la variable puede tomar cualquier valor dentro de cierto intervalo.

Este tipo de variables se presenta con más frecuencia en los sistemas físicos que las variables discretas, debido a que la masa, el calor, el volumen, el tiempo y otras

características, pueden tomar cualquier nivel dentro de una escala continua.

La manera acostumbrada de organizar los datos deducidos a partir de observaciones es desplegarlos como una distribución de frecuencias, que muestra el número de veces que una variable cae en distintos intervalos. Por ejemplo, suponga que se hacen 1000 observaciones de longitudes de llamadas telefónicas y que se tabula en intervalos de 10 segundos. La distribución de frecuencias podría aparecer como se muestra en la tabla (A) (ver anexo 1). La columna más a la izquierda define una cantidad de intervalos y la siguiente registra el número de llamadas cuyas longitudes cayeron dentro de ese intervalo. También es posible desplegar gráficamente la distribución en una gráfica de barras, como se muestra en la figura (A) (ver anexo 1). Una forma más útil de describir la misma información es una distribución de frecuencias relativas en que se divide el número de observaciones para cada intervalo entre el número total de observaciones.

Sin embargo, es importante notar que la distribución de frecuencias relativas no necesariamente es la función de densidad de probabilidad para la variable. Si los n intervalos para los que se ha generado la distribución de frecuencias son $x_i < x <= x_{i+1}$ ($i=0,1,2,3,\dots,n-1$), y p_i es

la cuenta de la frecuencia relativa para el i -ésimo intervalo, se debe de interpretar a p_i como: $p_i = \int_{x_i}^{x_{i+1}} f(x) dx$.

La selección del tamaño del intervalo por lo general se hace suponiendo que es suficientemente pequeño para ignorar las variaciones de la función de densidad de probabilidad en el intervalo. De ser así, el valor de la función de densidad en cada intervalo es $p_i / (x_{i+1} - x_i)$.

En la práctica hay más interés en deducir la función de distribución acumulativa. Si se acumulan los valores de p_i ,

los valores sucesivos $F_r = \sum_{i=1}^r p_i$ ($r=1,2,3,\dots,n$), representan el valor de la función de distribución acumulativa en los puntos x_i ($r=1,2,3,\dots,n$), como se muestra en la quinta

columna de la tabla (A) (Ver anexo 1). Se puede tomar a una serie de segmentos de líneas rectas trazados a través de estos puntos, como se ilustra en la figura (B) (Ver anexo 1), como una aproximación a la función de distribución acumulativa. La aproximación de la línea recta se conforma con la suposición anterior de que la función de densidad de probabilidad es constante en cada intervalo. Se puede intentar una mejor aproximación ajustando una curva suave a los puntos; en tal caso se puede deducir la función de

densidad de probabilidad graficando la pendiente de la curva de distribución acumulativa.

CAPITULO II

SIMULACION

Cuando no es posible expresar la interrelación en una forma matemática conveniente porque el sistema es complejo o porque algunas respuestas están sujetas a una variación aleatoria, es necesario formular reglas de operación y estudio por medio de Simulación.

La simulación se puede definir como una técnica para resolver problemas, consiste en efectuar experimentos con un modelo del sistema que se desea estudiar.

2.1 Modelo de Simulación

Un modelo de simulación es una descripción de un sistema que puede "manipularse" para medir el efecto producido por la variación de una o mas variables dentro del sistema.

La descripción del problema puede ser simplemente una exposición por escrito de las operaciones o procesos involucrados y cómo varían y se relacionan éstos entre sí. Sin embargo, con el fin de hacer manipulable el modelo, la exposición por escrito se transforma generalmente en un "organigrama de flujo" que muestre la secuencia lógica de las operaciones.

La simulación es un medio de experimentar ideas o conceptos en condiciones que escapen a la experimentación en la vida real por causa del costo, tiempo o riesgo implicados.

2.2 Modelo Físico Dinámico

Es aquel que se apoya en una analogía entre el sistema que se estudia y algún otro sistema de alguna naturaleza distinta, en lo que por lo general la analogía depende de una similitud entre las fuerzas que gobiernan el comportamiento de los sistemas.

2.3 Modelo Matemático Dinámico

Es aquel que permite deducir los cambios de los atributos del sistema en función del tiempo. Dependiendo de la complejidad del modelo, la deducción puede hacerse con una solución analítica o con cómputo numérico.

2.3.1 Atributo

Denota la propiedad de un objeto de interés en un sistema.

La técnica de simulación no pretende resolver analíticamente las ecuaciones de un modelo; por lo general un modelo matemático construido para fines de simulación es

de naturaleza distinta a uno formado para técnicas analíticas.

Se puede construir un modelo de simulación basándose en el método de diagrama de bloques; para lo cual es necesario describir matemáticamente a cada sección en forma directa y natural sin dar demasiada consideración a la complejidad que se introduce por tener muchas de esas secciones. Sin embargo, es necesario formar y organizar las ecuaciones de tal manera que se pueda utilizar un procedimiento rutinario para resolverlas simultáneamente.

En los sistemas continuos, en que el interés primordial es en cambios suaves, generalmente se utilizan conjuntos de ecuaciones diferenciales para describirlos. Se dice que las simulaciones basadas en esos modelos son simulaciones continuas.

Para los sistemas discretos en que el interés primario está en los eventos, las ecuaciones son esencialmente ecuaciones lógicas que expresan las condiciones para que ocurra un evento. La simulación consiste en seguir cambios en el estado del sistema resultados de la sucesión de eventos. Se dice que esas simulaciones son discretas.

2.4 Naturaleza Experimental de la Simulación

La manera en que se desarrollan los experimentos de simulación depende de la naturaleza del estudio. Por lo general, los estudios de sistemas son de tres tipos principales: análisis de sistemas, diseño de sistemas y lo que se llama postulación de sistemas.

2.4.1 Análisis de Sistemas

Este pretende comprender la manera en que opera un sistema existente o propuesto. La situación ideal sería que el investigador pudiera experimentar con el propio sistema, pero lo que realmente se hace es construir un modelo del sistema, y mediante simulación se investiga el comportamiento del modelo. Los resultados obtenidos se interpretan en términos del comportamiento del sistema.

2.4.2 Estudio del Diseño de Sistemas

El propósito es producir un sistema que satisfaga algunas especificaciones. El diseñador puede elegir o planear determinados sistemas de componentes, y conceptualmente elige una combinación determinada de componentes para construir un sistema. El sistema propuesto se modela y se predice su comportamiento a partir del conocimiento del comportamiento del modelo. Si el

comportamiento predicho se compara favorablemente con el comportamiento deseado, se acepta el diseño; de lo contrario, se rediseña el sistema y se repite el proceso.

2.4.3 Postulación del Sistema

Es característica de la manera en que se emplea la simulación en estudios ingenieriles, sociales, económicos, políticos y médicos en que se conoce el comportamiento del sistema pero no así con los procesos que producen dicho comportamiento. Se establecen hipótesis de un conjunto probable de entidades y actividades que pueden explicar el comportamiento.

2.5 Pasos involucrados en los Estudios de Simulación

Básicamente se pueden identificar determinados pasos en el proceso, como son:

1. Definición del problema
2. Plan del estudio
3. Formulación de un modelo matemático
4. Construcción de un programa de computador para el modelo
5. Validación del modelo
6. Diseño de experimentos

7. Ejecución de la corrida de simulación y análisis de resultados

Los dos primeros pasos son definir el problema y plantear el estudio. Aunque estos pasos pueden ser obvios, no dejan de ser importantes. No debe desarrollarse ningún estudio, simulación o cosa parecida sino hasta que se enuncien claramente el problema y los objetivos del estudio. Luego se pueden hacer las estimaciones del trabajo por realizar y del tiempo requerido. La utilidad del plan tampoco concluye cuando se inicia el estudio; el plan puede controlar el desarrollo del trabajo e impedir que el estudio se desequilibre concentrándose en un aspecto del problema a costa de otro.

El tercer paso consiste en construir un modelo, tarea que se puede considerar como que cae dentro de las dos subtareas. Es necesario establecer la estructura del modelo decidiendo los aspectos del comportamiento del sistema que son significativos para el problema de que se trata, y es necesario reunir los datos para proporcionar parámetros correctos para el modelo.

Dado un modelo matemático, la construcción de un programa de computador para el modelo, el cuarto paso, es una tarea relativamente bien definida. No es de necesidad una

tarea fácil y puede ser sumamente dilatada, pero el modelo establece las especificaciones de lo que debe programarse.

El quinto paso, la validación del modelo, es una área que requiere buena cantidad de juicio. En gran medida, el problema es el complemento de la formulación del modelo. Idealmente, los errores del modelo y los de programación se separan validando el modelo matemático antes de iniciar la programación.

El sexto paso es el diseño de un conjunto de experimentos que satisfagan los objetivos del estudio. Un factor que debe considerarse es el costo de correr el modelo del computador, ya que ello puede limitar el número de corridas que puedan hacerse. Y aunque no exista esta limitación, se debe ponderar cuidadosamente el número de corridas que se necesitan.

El último paso en el estudio de un sistema es ejecutar las corridas de simulación e interpretar los resultados.

2.6 Lenguajes de Simulación

Existen diferentes lenguajes especiales que permiten resolver ciertos problemas de simulación. A continuación se dará una breve descripción de cada uno de ellos.

2.6.1 Algol, Fortran, Cobol y Pl1

Estos lenguajes no fueron diseñados especialmente para la simulación de sistemas; sin embargo, su flexibilidad permite que se realice en ellos cualquier clase de programas. Les permite adaptarse a situaciones poco frecuentes y desarrollar los formatos de salida en la forma que se desee.

2.6.2 GASP (General Activity Simulation Program)

GASP consiste en una serie de subrutinas controladas por un programa principal llamado GASP EXECUTIVE; puede adaptarse para resolver un gran espectro de problemas de simulación; por estar desarrollado en FORTRAN y estructurado en una serie de subrutinas.

2.6.3 GPSS

Está especialmente desarrollado para ser utilizado en computadoras IBM. ES un lenguaje de simulación concebido para resolver problemas formulados en forma de diagramas de bloque.

2.6.4 SIMSCRIPT

El proceso que sigue es iterativo. se inicia estableciendo un modelo preliminar y se analizan los

resultados, de acuerdo con estos resultados se modifican las características del modelo para mejorar los resultados.

2.6.5 DYNAMO

Es capaz de manejar fácilmente problemas que presentan situaciones de retroalimentación. La aplicación actual del DYNAMO ha sido principalmente problemas de macro y microeconomía. (Ver cuadro comparativo figura A Anexo 2).

CAPITULO III

INTRODUCCION AL GPSS

3.1 Descripción General

Es un lenguaje de programación científica de propósito general para simular sistemas discretos. El GPSS es llamado El Sistema de Simulación de Propósito General.

El sistema a simular en GPSS se describe como un diagrama de bloques en que éstos representan las actividades, y las líneas que unen los bloques indican la secuencia en que pueden ejecutarse las actividades. Cuando hay selección de actividades, más de una línea sale de un bloque y la condición de la selección se expresa en el bloque.

El enfoque que se sigue en el GPSS es definir un conjunto de 43 tipos de bloques específicos, cada uno de los cuales representa una acción característica de los sistemas.

A cada tipo de bloque se le da un nombre que describe la acción y que se representa mediante un símbolo determinado. Cada tipo de bloque tiene una cantidad de campos de datos. Conforme se describen los bloques, se referirá a los campos como campo A,B,C y así sucesivamente.

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. (Ver Símbolos de los Diagramas de Bloque figura A Anexo 3).

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 un cambio en la condición del sistema.

Un diagrama de bloques del GPSS puede consistir en muchos bloques hasta cierto límite prescrito por el programa. 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 tiene la siguiente 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 la programación de problemas se les da un nombre simbólico. El programa ensamblador asocia el nombre con la localización apropiada. Los nombres simbólicos de nombres y otras entidades del programa deben de ir desde 3 hasta 5 caracteres no blancos, de los cuales los tres primeros deben ser letras.

3.2 Tiempos de Acción

La hora de reloj se representa mediante un número entero y el intervalo del tiempo real corresponde a una unidad del tiempo que escoge el usuario del programa. La unidad de tiempo no se expresa específicamente sino 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 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 puede utilizar distintos tipos de entrada para la función, lo que permite que estas 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 distribuido aleatoriamente, la función puede proporcionar una variable estocástica con una distribución determinada no uniforme.

3.3 Sucesión de Eventos

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 con base en primero entrado, primero servido dentro de la prioridad.

Normalmente, una transacción no pasa tiempos 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 bloque 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 otras 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.

3.4 Elección de Caminos

El bloque de **TRANSFER** permite elegir alguna localización distinta a la siguiente 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, y 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 caso de simultaneidad.

3.5 Facilidades y Almacenes

Asociadas 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 para representar el equipo del sistema.

Se define una facilidad como una entidad que puede utilizar una sola transacción a la vez, hasta cierto límite predeterminado. Pueden haber muchas 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 cero 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	Conmutador	Troncal
Transportación	Automóvil	Caseta de peaje.	Carretera
Procesamiento de datos.	Registro	Terminal	Memoria de computador.

Un troncal significa un cable que consiste en 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 una facilidad.

La figura de Símbolos de los Diagramas de Bloque (figura A Anexo 3), muestra cuatro tipos de 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. 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 cambian 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 mientras una transacción tiene el control de una facilidad. Se aplican arreglos semejantes para la utilización de los almacenajes.

3.6 Recolección de Estadísticas

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** de la figura de Símbolos de los Diagramas de Bloque (figura A Anexo 3), que

sirven para este propósito. Introducen otras dos entidades del programa GPSS, las colas y tablas. Como con 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 pueden 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 relativa a la cola de transacciones a menos que hayan entrado a una entidad de cola. El bloque **QUEUE** (cola) aumenta y el bloque de **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.

También es deseable medir el tiempo que se toman las transacciones para atravesar el sistema o partes del mismo, lo que puede hacerse con los bloques **MARK** y **TABULATE**. Cada uno de éstos tipos de bloque toma nota de la hora a la que llega una transacción al bloque. El bloque **MARK**

sencillamente indica la hora de llegada en la transacción (Si el campo A está en blanco, se utiliza una palabra especial. Con n en el campo A, se utiliza el n-ésimo parámetro). El bloque **TABULATE** resta la hora anotada por el bloque **MARK** de la hora de llegada en el bloque **TABULATE**. El tiempo, al que se conoce como "tiempo de tránsito", se registra 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 de una transacción como el tiempo que ha estado la misma en el sistema y la acción del bloque **MARK** es restaurar el tiempo de tránsito a cero.

Las prioridades del bloque **GENERATE** son tales que si una transacción puede salir del bloque cuando se crea, no se realicen creaciones adicionales hasta despejar el bloque.

El uso de modos de transferencia tanto condicional como incondicional es por medio del bloque **TRANSFER**.

CAPITULO IV

PROGRAMACION DEL GPSS

4.1 Simulación de un Supermercado

Para ilustrar el uso de funciones, parámetros y SNAs, se escribirá un modelo de simulación para un supermercado que opera como sigue:

Los clientes necesitan tomar una cesta antes de comenzar su compra; como hay un número limitado de cestas, salen sin hacer compra si no hay cestas disponibles. 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. En la figura A del Anexo 4, se muestra un diagrama de bloques del modelo y en la figura B del mismo Anexo se presenta el programa de GPSS correspondiente.

Hay cuatro secciones que se refieren a :

1. Tomar una cesta
2. Comprar
3. Pagar
4. Salir

Cada cliente está representado por una transacción y la unidad de tiempo es un 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 $y = \log_e(1-x)$. La regla para utilizar las funciones de los bloques **GENERATE** es que la media multiplica a la función. En este caso se supone un tiempo medio entre llegadas de 36.

Para representar las cestas se utiliza un almacenaje denotado por **BSKT** con capacidad igual al número de cestas; en este caso habrán 50 cestas. La decisión de que un cliente pueda 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 el **BSKT**. Si se dispone de una cesta, el bloque **ENTER** acepta la transacción y aumenta en uno 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 para 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 No. 1, para representar el número de artículos en compra. Este número se determina en un bloque **ASSING** que utiliza la distribución discreta. Siguiendo la técnica, entrada de números aleatorios a la función, hace que el número de artículos sean 5, 15 o 20 con probabilidades de 0.2, 0.3 y 0.4 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 que la relación funcional de la figura C Anexo 4, codificada como la función No 3. Como entrada, la función tiene el parámetro No. 1. El programa evalúa la función con el parámetro No. 1 de cada transacción que entra al bloque **ADVANCE**, la media del 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 es como sigue :

ADVANCE	1.FN3
ASSING	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 una, por lo que se les considera como una unidad de servicio con cinco dependientes, representados por un almacenaje de nombre **CKT** con capacidad de 5.

Habr a cierta congesti n en las cajas, y uno de los objetivos ser a medir el volumen de congesti n. En consecuencia, las transacciones van a un bloque de colas que las forma en una cola llamada **WAIT**. cuando se desocupa una caja, la transacci n sale del bloque **QUEUE** hacia el bloque **ENTER** e inmediatamente va un bloque **DEPART** para salir de la cola. De no haber congesti n, la transacci n se mueve directamente a trav s de los bloques **QUEUE**, **ENTER** y **DEPART**. Se supondr a que la funci n de pago requerir a de 10 segundos por art culo m s 25 segundos al empaclado. Ya que el par metro No. 1 es la cantidad de art culo, la siguiente declaraci n variable calcula el tiempo de pago:

1

VARIABLE

$P1+10+25$

Como mencionamos antes, es posible colocar una **SMA** en la mayoría de los campos de un bloque. En este caso el campo **A** de un bloque **ADVANCE** se hace igual a **V1**. Conforme cada transacción entra al bloque, el programa calcula el tiempo de acción para la declaración variable No. 1. Al complementar la función de pago, las transacciones 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	V1
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 **TRT**. Suponga que se desea llevar un registro del número de artículos que compra cada cliente, lo que realiza al pasar a otro bloque **TABULATE** que tabula **P1** en una tabla denominada **ITM**. Desde luego, la tabulación reproducirá la distribución original de la función 2. Se inserta este paso con el fin de definir el uso del 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 para esta sección es:

	TABULATE	TRT
	TABULATE	ITM
	LEAVE	BSKT
	TERMINATE	1
TRT	TABLE	M1,500,500.10
ITM	TABLE	P1,5,5,5

Notese que en el campo A de este bloque **TERMINATE** aparece un 1 en tanto que el campo A del *ew*, bloque **TERMINATE** está en blanco. En consecuencia, la corrida de simulación sólo cuenta con los clientes atendidos.

Debido a que se hace referencia a una función en el bloque **GENERATE** aquella debe definirse antes de este bloque **GENERATE**.

Suponemos que en el problema del supermercado a los clientes que compran 5 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.

Ahora se reemplaza a la sección de la solución original que se refiere al pago con el diagrama de bloques que se muestra en la figura D del Anexo 4.

Las transacciones van a un bloque **TEST** que prueba si el parámetro No 1, la cantidad de artículos de compra, es mayor que 5. En caso afirmativo, la transacción va a un bloque **ASSING** que pone al No. de parámetro 2 igual a 1. En caso contrario, la transacción va a un bloque **ASSING** que pone el No. de parámetro 2 en 2. Los otros bloques de la sección son iguales que antes, excepto porque el campo A se hace igual a *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 unen a la cola número 1 y utilizan el almacenaje número 1, en tanto que los compradores rápidos utilizan el número 2 en ambos casos. Suponiendo que deban utilizarse distintos tiempos de pago para los dos casos, se define una segunda declaración 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 o 2 de acuerdo con el valor del parámetro número 2.

4.2 Simulación de un Sistema Telefónico Simple

El sistema consta de una serie de llamadas de una cantidad de líneas telefónicas y éste debe conectarlas utilizando uno de un número limitado de enlaces. Sólo puede hacerse una llamada a cualquier línea en todo momento y se supone que pierden las llamadas si el llamado está ocupado o no se dispone de un enlace. Cada línea se representa mediante un interruptor lógico cuyo número es el de la línea. Se considera ocupada la línea si el interruptor está prendido. Cada llamada queda representada por una transacción; la unidad de tiempo elegida es de un segundo. Se supondrá que la distribución de llamadas es de Poisson con un tiempo medio entre llegadas de 12 segundos. También se supondrá que la longitud de las llamadas es exponencial. Se supondrá que cada nueva llamada puede venir de cualquiera de las líneas no ocupadas con igual probabilidad, y que su destino tiene igual probabilidad de ser cualquier línea distinta a sí misma.

En la figura E del Anexo 4 se muestra un diagrama de bloques de GPSS y su listado de compilación (ver Figura F del Anexo 4). Se utiliza un bloque **GENERATE** para crear una serie de transacciones que representan las llamadas. El modificador en el bloque es la misma función. No.1, que se utilizó en el ejemplo del supermercado. Se hace que la media del bloque **GENERATE** sea igual a 12. Se utilizarán los parámetros 1 y 2 para llevar el origen y destino de la

llamada. Se pasa cada transacción a dos bloques **ASSIGN** para elegir y registrar los valores. La fuente de la información es una declaración **VARIABLE**, la No. 1, que elige una línea al azar usando los métodos descritos en el Capítulo I.

El número de líneas N se multiplica por un número aleatorio entre 0 y 1, y se considera que la parte entera representa una selección de 1 entre N . No se permite el valor cero para ninguna entidad de GPSS, de manera que se suma uno al resultado para que la selección varíe entre uno y N . No es necesario emprender ninguna 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 truncando la parte fraccionaria. El número de líneas se varía en distintas corridas de manera que el número deseado de líneas se reserva en el campo de valor No. 1. La instrucción **INITIAL** carga el valor deseado al principio del programa.

La siguiente codificación coloca los números de línea elegidos al azar de 50 líneas en los parámetros Nos. 1 y 2:

	ASSIGN	1.V1
	ASSIGN	2.V1
1	VARIABLE	X1+RN1/1000+1
	INITIAL	X1.50

Se puede utilizar la misma declaración variables 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 es posible que el origen de la llamada ya esté ocupado. Utilizando el direccionamiento indirecto, un bloque de **GATE** verifica si el origen 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 compara el parámetro 2 contra el 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 prendiendo 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; en este

caso, 10. Si no puede entrar la transacción, la llamada se envía a un bloque **TERMINATE** llamado 3LKD y 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 devolver 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 No.1 con una media de 120.

Cuando la transacción sale del bloque **ADVANCE**, se termina la llamada y la transacción pasa a desconectarla apagando ambos interruptores lógicos, liberando el enlace y terminando.

Como ejemplo de utilizar los bloques de operaciones de conjuntos, suponemos que 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** (Ver figura G Anexo 4) que las coloca en una cadena que se conoce con el nombre de **WAIT** en orden ascendente del origen de llamadas (parámetro 1). Cuando una llamada termina y libera un enlace, prueba si hay llamada en espera llenando a un bloque **TEST** que compara un **SNA** de nombre **CHN** contra cero. 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** (Ver figura G Anexo 4), que busca las transacciones de la cadena con destino (parámetro 2) igual a uno. Si existe una de esas llamadas en espera, se desenlaza y envía a **GETL** para conectarse.

Si hay más de una llamada para la línea uno, 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 uno y no encuentra una, va al segundo bloque **UNLINK** para desenlazar la llamada con el origen más bajo. La figura H del Anexo 4 muestra la codificación del problema en su nueva forma. 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 debe dar la oportunidad de dar el enlace. Notamos 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 la cuenta de las llamadas en espera.

CAPITULO V

APLICACIONES DEL SISTEMA GPSS

5.1 Simulación de un taller de Manufactura

Un taller de Manufactura es una planta consistente en un establecimiento de máquinas operando independientemente. Este establecimiento de máquinas debe ser dividido en subgrupos de máquinas, las cuales tienen las mismas características. Un trabajo es una unidad de rendimiento producida por una serie de operaciones ejecutadas en las máquinas con una secuencia específica. Esta secuencia es llamada la rutina o la secuencia tecnológica del trabajo.

Cuando se programan trabajos através del taller el objetivo es asegurar que las operaciones esten hechas en la secuencia apropiada, mientras que el criterio de reunión a determinada hora sea mínimo en la demora de entrega, máxima utilización del equipo, o mínimo inventario en el proceso.

En años recientes han existido investigaciones extensivas en el area de la programación de los talleres de manufactura, y el problema continúa intrigando a analistas, quienes no pueden ver porque un sistema con una simple estructura deba ser tan laborioso de tratar, solo hasta que ellos tengan que realizar sus propios estudios.

Existen dos clases de problemas en un taller de manufactura:

1. El caso Estático, el cual todos los trabajos comienzan en un tiempo cero.
2. El caso Dinámico, en el cual el tiempo de arribo de los trabajos varía con el tiempo.

Particularmente en el último caso la complejidad de las interacciones entre las diversas partes del sistema hacen de la técnica de simulación más apropiada que analítica.

Los aspectos de las operaciones de un taller de manufactura más extensivamente examinadas son, los envíos o las reglas del programa. Muchos trabajos pueden competir por una máquina disponible, y las reglas resuelven los conflictos que ya deben de estar definidos. Muchas reglas de programación deben de ser sugeridas, incluyendo:

1. Primeras entradas primeras salidas (PEPS): El trabajo con el tiempo de arribo más temprano en la línea es seleccionado.
2. Prioridad más alta: A cada trabajo se asigna una prioridad de llegada. El trabajo con la más alta prioridad es asignado primeramente a la máquina.

3. El tiempo de proceso más corto: Es el trabajo que requiere el mínimo tiempo de máquina es seleccionado de la línea de espera de trabajos.

4. El trabajo de menor permanencia de retardo: El trabajo que tiene el mínimo tiempo de atraso se selecciona.

El trabajo atrasado o que tiene el tiempo de permanencia libre, esta definido como fecha vencida, menos el tiempo que corre menos la suma del tiempo que permanece en el proceso para ese trabajo. Así es que este trabajo, lo más probable siempre se encuentre atrasado.

5. Fecha de retraso de un trabajo más reciente: El trabajo con la fecha de retraso más reciente es seleccionado. Esto es similar al punto anterior pero no incluye el tiempo de proceso dentro de la regla de programación.

Las ventajas del uso de la simulación para evaluar estas u otras reglas o combinación de las mismas, descansa en la flexibilidad experimental. La simulación permite al usuario evaluar políticas de operación alternativas sobre una condición razonablemente realista sin necesidad de

simplificaciones usualmente requeridas para obtener un modelo analítico fácil de trabajar.

5.2 El Sistema

El sistema que concideramos tiene las siguientes características:

■ diferentes grupos de maquinas; cada grupo es tratado como un centro de trabajo independiente.

Una operación es el procesamiento de un trabajo en una máquina en particular. Una vez que la operación empieza tiene que continuar hasta que sea completada.

El problema dinámico se considera como: La entrada de un trabajo se considera estocástica. Se asume que M diferentes trabajos son entregados al taller de acuerdo a una distribución de probabilidades conocida.

El tiempo de tránsito requiere de mover trabajos de un grupo de máquinas a otro grupo de máquinas, está incluido el tiempo de procesamiento.

5.3 Requerimientos del Modelo

La siguiente simulación fué diseñada para demostrar la naturaleza de los trabajos de un taller, sus problemas y para mostrar la factibilidad de un sistema computalizado basado en el GPSS. El modelo está basado parte en experiencia práctica y parte en los trabajos teóricos.

Un mayor requerimiento es el que sea más flexibilidad en permitir varios experimentos sobre un sistema predeterminado y así permitir modificaciones más fáciles al propio sistema. Numerosas reglas de programación deben ser evaluadas independientemente o combinadas, y las condiciones del sistema, así como el modelo de las llegadas del trabajo o el tiempo de proceso del mismo debe de ser variado para estudiar el efecto de la variación del tráfico en diferentes estrategias.

Aunque muy flexible, el modelo debe ser suficientemente simple para usarse y extender su estructura rápidamente y así tener la disposición de usarlo y modificarlo sin demora.

5.4 Características Numéricas

Describimos dos configuraciones particulares del taller de manufactura de numerosos casos que fueron modelados. El propósito de esta descripción es mostrar primero cómo el

modelo fué implementado en el GPSS, y segundo cómo pueden hacerse los cambios. El primer taller consiste de cuatro grupos de máquinas, procesando 10 tipos de trabajos. Una descripción de las rutinas de trabajo es mostrada en la matriz de la tabla 1. Cada renglón contiene información acerca de los diferentes tipos de trabajo. La columna 11 especifica el número de operaciones requeridas para un trabajo, y las columnas de la 1 a la 5 muestran el número de grupo de máquinas en la secuencia en la cual son utilizadas para el trabajo. Las columnas de la 6 a la 10 no son usadas, permiten que la secuencia sea extendida 10 operaciones sin la modificación del tamaño de la matriz.

TABLA 1
RUTINA PARA EL PRIMER TALLER

	COLUMNA										
	1	2	3	4	5	6	7	8	9	10	11
R	1	1	4	3	3	4					8
E	2	4	2	1	2	-					4
N	3	4	1	2	-						3
G	4	3	4	2	1	-					4
L	5	2	1	2	4	1					8
O	6	2	2	4	1	-					4
N	7	1	2	4	2	-					4
	8	1	4	1	2	2					8
	9	4	1	2	2	-					4
	10	1	2	1	4	2					4

Además, cada trabajo pertenece a una de las dos clases de procesos. La matriz de la tabla (identificación) contiene información de los tiempos del proceso en minutos; el primer renglón para trabajos de la clase número 1 y el segundo renglón para trabajos de la clase número 2. Las columnas

representan el grupo de máquinas. Por ejemplo en la línea 2 columna 3 da el tiempo de proceso de las máquinas del grupo 3 para trabajos pertenecientes a la clase 2.

TABLA 2
TIEMPOS DE PROCESOS

CLASES	GRUPOS			
	1	2	3	4
1	18	28	38	48
2	15	38	48	45

Diferentes números de máquinas para cada grupo, es llamada capacidad, son mencionadas en nuestra discusión de la implementación del GPSS. Una de las configuraciones es totalizada en la tabla siguiente.

TABLA 3
CAPACIDAD DEL PRIMER TALLER

GRUPO	1	2	3	4
NUMERO	2	2	2	2

Las características numéricas del segundo taller que consiste en 5 grupos de máquinas y 11 tipos de trabajos, están consideradas y totalizadas en las siguientes tablas:

TABLA 4
RUTINA PARA EL SEGUNDO TALLER

		COLUMNA										
		1	2	3	4	5	6	.	.	.	11	
R E N G L O N	1	1	4	3	3	4	-					5
	2	4	3	1	2	-						4
	3	4	1	2	-	-						3
	4	3	4	2	1	5	-					5
	5	3	1	3	4	1	5	-				6
	6	2	3	4	1	-	-					4
	7	1	3	4	2	-	-					4
	8	3	4	1	2	3	5	-				6
	9	4	1	2	3	-	-					4
	10	2	1	4	3	-	-					4
	11	1	3	2	-	-	-					3

TABLA 5
TIEMPOS DE PROCESOS DEL SEGUNDO TALLER

		GRUPO				
CLASE		1	2	3	4	5
1		10	20	30	40	20
2		15	30	40	45	25

TABLA 6
CAPACIDAD DEL SEGUNDO TALLER

GRUPO	1	2	3	4	5
NUMERO	2	2	3	3	1

5.5 Modelo en el GPSS

5.5.1 Correspondencia entre los elementos del sistema y las entidades del GPSS.

Un grupo de máquinas está representado por un almacenaje, el cual tiene una capacidad definida por el número de máquinas en ese grupo.

Una línea de espera antes de un número de máquinas ocupadas esta simulada por tres entidades del GPSS: Una línea, un grupo, y una cadena de usuarios, con el número de referencia correspondiente al número de grupo de máquinas.

En la línea se reúnen las estadísticas. La entidad del grupo GPSS permite al programador usar la instrucción **SCAN** para encontrar características particulares de las transacciones que esperan como el menor trabajo que permanece retrasado o el más temprano trabajo con fecha vencida. Estas características son usadas para programar un trabajo en la primer máquina disponible. La cadena de usuarios contiene transacciones las cuales esperan por el servicio, así que solamente las transacciones correspondientes de trabajos para ser procesados o no de una cadena de usuarios.

5.6 Comprensión del Modelo

El modelo del GPSS consiste de tres submodelos llamados programa de trabajo, proceso del trabajo, y creación del trabajo. Don tipos de transacciones son usadas. La primera representa el trabajo y la segunda el control de las transacciones las cuales programan el trabajo en competencia para una máquina disponible.

5.6.1 El modelo de programación del trabajo. (Ver tabla 7)

Al comienzo de la simulación un control de la transacción es generada para cada número de máquinas. Cada control de la transacción es usado para encontrar que transacción debe ser programada cuando una máquina de su grupo se encuentra disponible. Dos o más de las reglas de programación mencionadas anteriormente debe de ser usadas simultaneamente, es así que el control de la transacción examina todo el trabajo, las transacciones de espera en la cadena de usuarios correspondientes. Esta selección requiere información, así como la más alta prioridad, el tiempo más corto de proceso, el menor trabajo que permanece retrasado, el más largo tiempo en la línea o el trabajo con menos fecha de retraso y almacenaje de contingencia. Cada valor es multiplicado por su correspondiente factor de efectividad (este factor se mostrará en el listado del programa como las variables de la 5 a 9 del GPSS) para proteger las reglas del programa, las cuales no son usadas en la corrida posterior.

Después, de encontrar una transacción sin ninguna, que satisfaga las condiciones activas o las reglas de programación del primer arreglo de condiciones (definidas por variables bulianas) de la BV6 del listado de la corrida. Si ninguno, el segundo arreglo es probado (BV7), y siendo así, totalizado en la figura A Anexo 5, hasta que una transacción del programa pueda ser programada. Esta transacción del

trabajo es diferencial de su cadena de usuarios y enviada al submodelo del proceso del trabajo.

Después el control de la transacción prueba si al menos alguna otro transacción del trabajo está esperando una máquina disponible, así como, el procedimiento es repetido. Tan pronto como una de las condiciones (al menos una transacción de trabajo esperando y una máquina disponible) esta no es satisfecha, el control de la transacción es eslabonado a la cadena de usuarios y de este modo desactivada.

TABLA 7

ARREGLO DE COMBINACIONES	PRIMER ARREGLO	SEGUNDO ARREGLO	TERCER ARREGLO	CUARTO ARREGLO	QUINTO ARREGLO
REGLAS DE PROG.					
MAS ALTA PRIORI.	█	█	█	█	█
MAS CORTO TIEMPO DE PROCESO INMI-NENTE.				█	
MINIMO ATRASO			█		
MAS LARGO TIEMPO EN LA LINEA	█	█			
FECHA DE VENCI-MIENTO INMEDIATA	█				

5.6.2 Submodelo de proceso del trabajo (Ver figuras B y C Anexo 5).

Quando la transacción del trabajo es enviada al

submodelo del proceso del trabajo, este es removido del grupo y de la línea de donde ésta estuvo esperando a una máquina disponible. Después de ser procesada, notifica al control de la transacción que una máquina ha quedado disponible. El número de operaciones permanentes debe ser ejecutada y probada: Si el número es cero, las estadísticas son tabuladas y la transacción del trabajo es destruida. Si al menos una operación aparece, su inminente número de máquina, el tiempo de proceso, el retraso del trabajo son actualizados. La transacción del trabajo entonces entra a la siguiente línea, junto con el siguiente número de trabajos esperando, notifica al control de la transacción correspondiente su llegada, y está encaminada a la cadena de usuarios esperando ser programada.

5.6.3 Submodelo de la creación del trabajo (Ver figuras D y E Anexo 5)

El submodelo de la creación del trabajo genera grupos de transacciones de trabajo. Cada grupo contiene una transacción de los diferentes tipos especificados. Cada transacción es asignada a la secuencia de operaciones correspondiente para su tipo, y toda la información necesaria como si fueran con fecha vencida y prioritaria. Cada transacción de trabajo entra a la línea asignada junto con su grupo, así, las señales de llegada al control de las transacciones antes de ser encaminadas a la cadena de usuarios.

5.6.4 Descripción del programa

El listado del GPSS aparece en las figuras F1, F2 y F3 del Anexo 5. El modelo simulado es el segundo taller de manufactura descrito anteriormente. La siguiente instrucción SNAS (atributo numérico estándar) ha sido usada.

Byte msavevalue 1 y palabra msavevalue 1. Byte msavevalue 1 contiene la información de la rutina definida en la tabla 4. La palabra msavevalue 1 contiene el tiempo de proceso dado en la tabla 5. Ambas msavevalue son inicializadas por la declaración **INITIUM** (declaraciones de la 67 a la 85) y son retenidos sin alteración y son sólo de referencia.

5.6.4.1 Almacenes.- (Ver tabla 5)

La capacidad de un taller de manufactura esta definida por la declaración **STORAGE** (44).

5.6.4.2 Parámetros.- Transacciones del trabajo.

Parámetro Byte

PB1:	Número de clase
PB2:	Número total de operaciones
PB3:	Número de operaciones completas

PB4: Índice de la operaciones inminentes
PB5: Grupo de máquinas al corriente
PB6-PB8: Sin uso
PB9 y PB10: Usado para varias operaciones

Parámetro de media palabra

PH1: Tipo de trabajo; usado como indice de
renglón para un Byte msavevalue 1

Parámetro de palabra

PF1: Contiene la suma de todos los tipos de
proceso para el trabajo
PF2: Generación del tiempo
PF3: Fecha vencida
PF4: Suma de los tiempos del proceso para
operaciones ya ejecutadas
PF5: Tiempo de permanencia en el proceso.
PF6: Tiempo de entrada a la línea.
PF7: Sin uso
PF8: Trabajo atrasado
PF9: Tiempo corriente del proceso

Parámetros: control de las transacciones

Parámetro Byte

PB5: Número de grupo de máquina a ser controlada
PB10: Usada para las operaciones

Parámetro de palabra

PF7: Usada para las operaciones de las variables
de la 15 a la 19

Los otros parámetros no son usados.

5.6.4.3 Funciones

PRIOR.- Función discreta usada para dar prioridad a una transacción de trabajo.

CLASS.- Función discreta para asignar un número de clase (ya sea uno o dos) a una transacción de trabajo.

5.6.4.4 Variables

V1: Define el número de diferentes tipos de trabajo

V2: Define el número de grupo de máquinas

- V3: El número máximo de operaciones en una secuencia
- V4: El factor proporcional para computar una fecha de retraso
- V5 al V9: Especificación de las reglas del programa o la combinación de las reglas del programa si serán usadas. Una valuación de unidad indica si la regla correspondiente es usada. Una valuación de 20 indica si la regla no va a ser efectiva, excepto para V5 cuando el cero es requerido.
- V5: Para la más alta prioridad. El cero indica que la regla de programación es no efectiva
- V6: La más corta operación inminente
- V7: El tiempo más corto que permanece un trabajo atrasado
- V8: El más largo tiempo en la línea
- V9: La fecha de vencimiento más inmediata
- V10: Operaciones de fecha atrasada
- V11: Operaciones que permanecen en el tiempo de proceso
- V12: Sin uso
- V13: Operación del trabajo atrasado
- V14: Da a la cadena de usuarios el número para desactivar el control de la transacción, cada grupo de máquina tiene una cadena de

usuarios correspondiente para el control de la transacción del grupo.

- V15 a V19: Usadas para las operaciones de pantalla. (Ver más detalles en la siguiente sección) Las operaciones valuadas están almacenadas en XF1 a XF5 respectivamente.
- V20: Usado en SPLIT (declaración 112) para generar un grupo de V1 transacciones de trabajo
- V21: Usado en SPLIT (declaración 175)
- VETERMI: Evalua la duración del trabajo (es usado en la tabla (COMPL)

5.6.4.4.1 Variables Bulianas.

BV1 a la BV9 Usada para representar la disposición de las condiciones de la figura A Anexo 5, la cual debe de ser satisfecha por una transacción del programa.

Por ejemplo, BV9 corresponde al cuarto arreglo de la figura A Anexo 5. Una transacción está diferenciada si su prioridad es más grande o igual a la más alta prioridad en un grupo de transacciones esperando y si su tiempo de proceso inminente es menor o igual que el más corto tiempo de proceso inminente.

5.6.4.4.2 Palabras de salvamento

- XF1: Almacenar la más alta prioridad de un grupo común de espera de las transacciones del trabajo
- XF2: El más corto tiempo del proceso inminente
- XF3: El mínimo retraso del trabajo
- XF4: El más largo tiempo en la línea
- XF5: La fecha de vencimiento más inminente.

5.6.4.4.3 Tablas

COMPL.- Tabla del tiempo de frecuencias delante o detrás de un trabajo con fecha de vencimiento. Una valuación negativa significa que el trabajo fué completado en destiempo (registrado por VSTERMI)

TRANS.- Es la tabla de frecuencia del tiempo de tránsito; la duración del trabajo incluyendo los tiempos de espera.

5.6.5 Comentarios al Listado del Programa

El submodelo de la creación del trabajo comienza en la declaración 110 y finaliza en la declaración 137; el submodelo del proceso del trabajo comienza en la declaración 142 y finaliza en la declaración 164; y el submodelo de la

programación del trabajo comienza en la declaración 174 y finaliza en la declaración 214.

Combinar las reglas del programa mencionadas en las secciones anteriores, muchos pasos de computación son requeridos (de la declaración 182 a la 211). Primero, un control de transacción busca todas las transacciones del trabajo que esperan en la cadena de usuarios, así que aquel control solicita el siguiente valor: La más alta prioridad, el mínimo tiempo de proceso inminente, el más corto atraso de trabajo, el más temprano tiempo de entrada a la línea, y la fecha de vencimiento más inmediata. Estas valoraciones son multiplicadas por su correspondiente factor de efectividad (variables de la 5 a la 9), y el resultado es almacenado en la palabra de contingencia XF1 a XF5.

Por ejemplo, declaración 182. (la prioridad de valor más alto de la transacción que espera es encontrada y almacenada en el parámetro de palabra 7 del control de la transacción; así, el valor multiplicado por la variable V5, cuyo valor es 0 ó 1; si es cero, éste indica que el usuario no desea programar el trabajo de acuerdo con el valor de la prioridad.

El resultado de la multiplicación es almacenado en XF1, que es la declaración 183. El contenido de XF1 es en este caso usado en la variable buliana 1, la cuál es usada en BV6 a BV9, las cuales definen el criterio del listado.

Una transacción del trabajo que satisface la regla del programa especificada que es, para una de las cuales correspondientes de las variables bulianas de la BV6 al BV9 da el valor de la unidad, es diferenciada de la cadena de usuarios y enviada al submodelo del proceso del trabajo (declaración 142).

Después de seleccionar la transacción del trabajo, el control de la transacción espera hasta que la transacción del trabajo sea efectivamente diferenciada por el GPSS y enviada al submodelo del proceso del trabajo. Esto es archivado por la instrucción PRIORITI 0, o el bloque BUFFER. (declaración 209), el cual suspende el proceso del control de la transacción e inicializa la búsqueda en el GPSS y eslabona los eventos corrientes.

5.6.6 Experimentos y Resultados

En esta sección daremos una discusión concisa de algunos resultados significados obtenidos de varios experimentos. La tabla 8 totaliza las salidas del programa de las figuras F1, F2 y F3 del Anexo 5, después de 18,000 unidades de tiempo.

Para las características numéricas de la corrida de las tablas 2, 3, 4 y 5 han sido usadas, con la generación del trabajo que proporciona un grupo en cada 170+-20 unidades de

tiempo, uniformemente distribuidos. " El menor tiempo que permanece retrasado un trabajo " aplicamos una regla de programación que ha sido probada, para V7=1, V6-V9=20 y V5=0.

TABLA 8
TABLA DE RESULTADOS SELECTOS: ALMACENAJES

ALMACENAJES	CAPACIDAD	PROM CON- TENIDO	ENTRADAS	PROM TIEMPO POR UNIDAD	PROM TIEMPO TOTAL
11	2	0.883	1278	12.478	0.441
12	2	1.617	1169	24.911	0.888
13	3	2.267	1168	24.943	0.758
14	2	2.784	1169	42.417	0.918
15	1	0.399	317	22.681	0.399

CONTENIDO CORRIENTE	MAXIMO CONTENIDO
-	2
2	2
3	3
2	3
-	1

TABLA 9
TABLA DE RESULTADOS SELECTOS: LINEAS

LINEA	CONTENIDO MAY MAX	CONTENIDO PROMEDIO	TOTAL DE ENTRADAS	ENTRADA CERO	% DE CEROS
11	4	0.177	1278	926	72.6
12	4	0.453	1169	592	50.6
13	5	0.441	1171	664	56.7
14	8	1.194	1178	259	22.1
15	2	0.062	317	233	73.5

TIEMPO PROM DE LA TRANS.	% PROM POR TRANS.	CONTENIDO CORRIENTE
2.512	9.188	-
6.976	14.135	-
6.784	15.678	3
18.227	23.602	1
3.052	13.486	-

* Tiempo promedio por transacción, excluye entradas cero

TABLA 10
RESULTADOS SELECTOS TABLA TRANS

ENTRADAS EN LA TABLA ARGUMENTO PRINCIPAL DESVIACION STD.
1168 187,987 53,888

LIMITE SUP	FREC. OBSERVADA	% DEL TOTAL	% ACUMU- LATIVO	ACUMULATIVO REMANENTE
130	355	30.47	30.4	69.5
150	194	16.65	47.1	52.8
170	176	15.18	62.2	37.7
190	166	14.67	76.3	23.6
210	99	8.49	84.8	15.1
230	67	5.75	90.6	9.4
250	44	3.77	94.3	5.6
270	26	2.23	96.5	3.4
290	19	1.64	98.1	1.8
310	10	0.86	99.0	0.3
330	5	0.34	100	0.0

La permanencia de las frecuencias son todas cero

Cambiaremos ahora la estructura del sistema, de acuerdo con las tablas 1, 2 y 3 para obtener 10 diferentes tipos de trabajos en lugar de 11, cuatro grupos en lugar de 5, y un número máximo de operaciones de 5 en lugar de 6.

En primera instancia se cambian los valores de la variable 1 (ahora 10), variable 2 (ahora 4), y variable 3 (ahora 5).

Segundo, la instrucción **STORAGE** (declaración 44) es alterada para los valores de la tabla 2 y 3. Así como, la declaración **INITIUM** el Byte de la matriz de contingencia contenido 1 es cambiado de acuerdo con la tabla 1.

El arribo del trabajo proporciona (declaración 110) el siguiente cambio: Ahora uniformemente distribuidos en 360 y 200. Si el trabajo esta programado de acuerdo con "El tiempo de operación más corto inminente" la regla, define a $V5=0$, $V6=1$, $V7-V9=20$, las estadísticas estan dadas en las tablas 11, 12 y 13, y los resultados estan reunidos despues de 18000 unidades de tiempo. Si los trabajos estan programados de acuerdo a la programación de reglas "El menor trabajo que permanece retrasado" y "La fecha de vencimiento más inmediata", $V7$ y $V9 = 1$, $V5=0$, $V6$ y $V8 = 20$, los resultados, son mostrados en las tablas 14, 15 y 16, donde se muestran algunas mejoras; comparando ambas tablas **TRANS** por ejemplo.

TABLA 11
PROGRAMA DE UN TALLER DE MANUFACTURA
TABLA DE RESULTADOS SELECTOS: ALMACENAJES

ALMACENAJES	CAPACIDAD	PROM CON-TENIDO	ENTRADAS	PROM TIEMPO POR UNIDAD	PROM TIEMPO TOTAL
11	1	0.762	1102	12.487	0.762
12	2	1.388	1002	24.988	0.692
13	3	1.934	1002	34.747	0.667
14	3	2.097	1102	42.431	0.668

CONTENIDO CORRIENTE	MAXIMO CONTENIDO
1	1
1	2
2	2
3	3

TABLA 12
TABLA DE RESULTADOS SELECTOS: LINEAS

LINEA	CONTENIDO MAX	CONTENIDO PROMEDIO	TOTAL DE ENTRADAS	ENTRADA CERO	% DE CEROS
11	4	0.862	1102	398	34.4
12	5	0.382	1002	598	59.2
13	7	1.498	1002	122	12.1
14	5	0.842	1102	378	33.9

TIEMPO PROM DE LA TRANS.	% PROM TIEMPO POR TRANS.	CONTENIDO CORRIENTE
0.669	12.531	1
0.328	15.578	-
26.099	30.975	1
13.720	20.703	2

* Tiempo promedio por transacción, excluye entradas cero

TABLA 13

RESULTADOS SELECTOS TABLA TRANS

ENTRADAS EN LA TABLA ARGUMENTO PRINCIPAL DEVIACION STD.
598 178.843 58.437

LIMITES	FREQ.	% DEL	% ACUMU-	ACUMULATIVO
SUP	OBSERVADA	TOTAL	LATIVO	REMANENTE
130	211	21.14	21.1	78.8
160	162	16.23	37.3	62.6
170	134	13.42	50.8	49.1
190	109	10.92	61.7	38.2
210	111	11.12	72.8	27.1
220	63	6.21	79.1	18.8
230	45	4.50	83.7	14.2
270	24	2.40	86.1	9.6
300	26	2.60	88.7	6.0
310	23	2.28	91.0	3.4
330	9	0.90	91.9	2.5
350	12	1.20	93.1	1.2
370	5	0.50	93.6	0.7
390	4	0.40	94.0	0.3
410	1	0.10	94.1	0.2
420	0	0.00	94.2	0.2
450	1	0.10	94.3	0.1
470	0	0.00	94.3	0.1
490	1	0.10	100	0.0

La permanencia de las frecuencias son todas cero

TABLA 14
PROGRAMA DE UN TALLER DE MANUFACTURA
TABLA DE RESULTADOS SELECTOS: ALMACENAJES

ALMACENAJES	CAPACIDAD	PROM CON- TENIDO	ENTRADAS	PROM TIEMPO POR UNIDAD	PROM TIEMPO TOTAL
11	1	0.762	1102	12.487	0.762
12	2	1.266	1002	24.091	0.692
13	2	1.924	1002	34.784	0.767
14	3	2.597	1102	42.440	0.866

CONTENIDO CORRIENTE	MAXIMO CONTENIDO
1	1
2	2
2	2
3	2

TABLA 15
TABLA DE RESULTADOS SELECTOS: LINEAS

LINEA	CONTENIDO DO MAX	CONTENIDO PROMEDIO	TOTAL DE ENTRADAS	ENTRADA CERO	% DE CEROS
11	5	0.699	1102	482	36.5
12	5	0.379	1002	598	59.2
13	4	1.400	1002	131	13.0
14	4	0.744	1102	414	37.4

TIEMPO PROM DE LA TRANS.	% PROM TIEMPO POR TRANS. *	CONTENIDO CORRIENTE
0.766	15.421	1
0.694	18.230	-
25.733	30.784	-
12.122	19.307	3

* Tiempo promedio por transacción. excluye entradas cero

TABLA 16
RESULTADOS SELECTOS TABLA TRANS

ENTRADAS EN LA TABLA ARGUMENTO PRINCIPAL DESVIACION STD.
999 177.424 86.988

LIMITE SUP	FREC. OBSERVADA	% DEL TOTAL	% ADUMU- LATIVO	ACUMULATIVO ESPARETE
120	106	10.63	10.6	01.3
130	103	10.33	20.9	02.1
140	139	13.93	34.7	03.2
150	147	14.72	49.5	07.4
160	138	13.83	63.3	11.6
170	72	7.31	70.6	16.3
180	59	5.91	76.5	19.4
190	42	4.20	80.7	23.2
200	32	3.20	83.9	26.9
210	19	1.90	85.8	28.8
220	6	0.60	86.4	29.4
230	2	0.20	86.6	29.6
240	0	0.00	86.6	29.6
250	1	0.10	86.7	29.7

La permanencia de las frecuencias son todas cero

5.6.7 Conclusiones

La simulación de un taller de manufactura estipula una definición diferente de la evaluación de la regla de programación sobre diferentes condiciones experimentales. Los problemas descritos anteriormente no representan a los actuales talleres. Por lo tanto, el resultado del modelo ha sido un poco estilizado para hacerlo más claro. Sin embargo, por su flexibilidad, creemos que el modelo es valorado como una herramienta de simulación avanzada y cursos de Ingeniería Industrial, porque ayuda a las personas a entender los problemas complejos sobre una investigación experimental y

evaluar comparativamente la prioridad de las disciplinas en la programación de la producción.

Otras modificaciones que pueden ser tratadas incluyen:

1. La prioridad de la composición de la regla como se describió.
2. La simulación del corte de las máquinas.
3. Diferentes modelos de llegadas; por ejemplo, una diferente probabilidad de distribución para cada tipo de trabajo.
4. Una búsqueda óptima del programa diseñado para variar los valores de los parámetros automáticamente sin especificar los límites.

CAPITULO VI

ANEXOS

АНЕКС 1

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

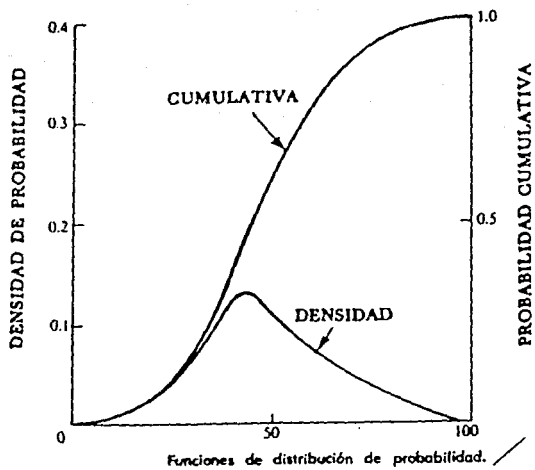
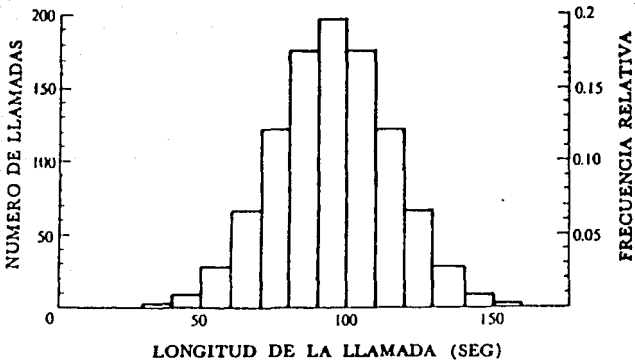


FIGURA A

Longitud de llamada (seg)	Número de llamadas	Frecuencia relativa	Densidad de probabilidad	Distribución acumulativa
0	0	0.000	0.0000	0.000
10	1	0.001	0.0001	0.001
20	1	0.001	0.0001	0.002
30	1	0.001	0.0001	0.003
40	2	0.002	0.0002	0.005
50	8	0.008	0.0008	0.013
60	28	0.028	0.0028	0.041
70	65	0.065	0.0065	0.106
80	121	0.121	0.0121	0.227
90	175	0.175	0.0175	0.402
100	197	0.197	0.0197	0.599
110	175	0.175	0.0175	0.774
120	121	0.121	0.0121	0.895
130	65	0.065	0.0065	0.960
140	28	0.028	0.0028	0.988
150	9	0.009	0.0009	0.997
160	2	0.002	0.0002	0.999
170	1	0.001	0.0001	1.000

TABLA A



Distribución de frecuencias de longitudes de llamadas telefónicas.

FIGURA B

ANEXO 2

FIGURA A

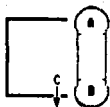
DESCRIPCION	DEFINICIONES	PROPOSITO	PERIODICIDAD - DEL EJERCICIO	CLASIFICACION DE INDICADORES DE RESULTADO
RECURSOS	BASE DUDA LAS COMPUTADORAS	GENERAL	ANUAL	VARIAVEL
2-71	BASE DUDA LAS COMPUTADORAS	GENERAL	REGULAR	BUENOS
6-71	SEM	INDICADOR DE BUDGETO	REGULAR	BUENOS
8-71	SEM	GENERAL	DUDA	BUENOS
9-71	PRINCIPALMENTE SEM	SISTEMAS COMPUTACIONAL	REGULAR	EXCELENTE

CUADRO COMPARATIVO.

ANEXO 3



ARMAZA



ENLAZA



AGARRA



ASIGNA



LOGICA



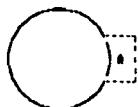
TABULA



SALE



PARCA



TEMINA



ENTRA



PRIORIDAD



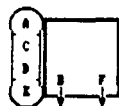
PIENSA



COMPLETA



COLA



DESPLAZA



CONIENE



LIBERA



NUMEROA



RESERVA VALOR

FIGURA A

ANEXO 4

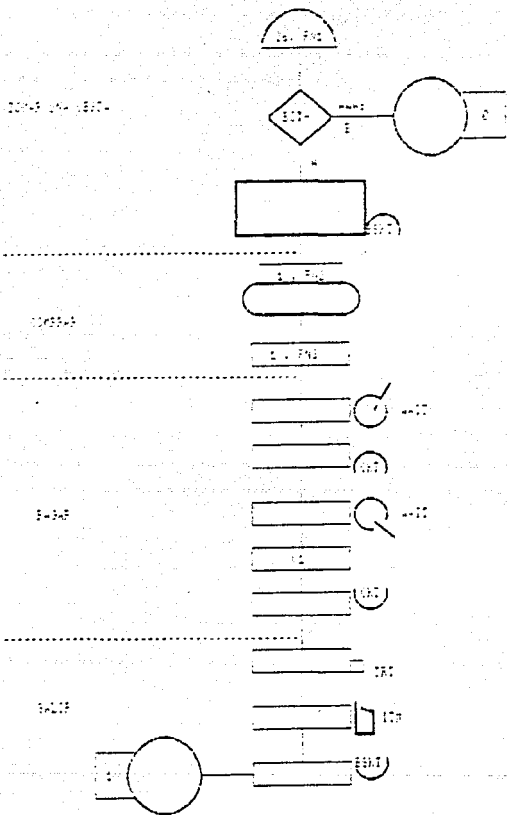
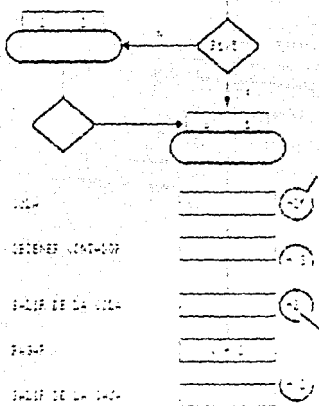


FIGURA A

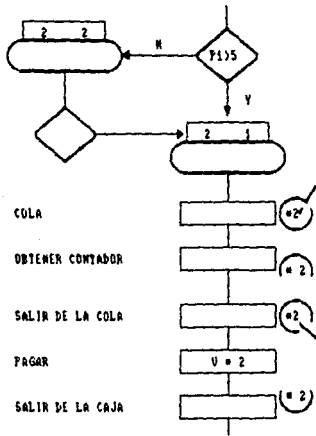
BLOCK NUMBER	LOC	OPERATION	A, M, C, D, F, G	COMMENTS
		STIPULATION OF A SUPERMARKET		
	1	FUNCTION	MN1, C24	FUNCTION FOR I/A INTERVAL
				0.3, 0.0 / 0.1, 0.1 C4 / G. 2, 0.222 / U. 3, 0.355 / 0.4, 0.509 / 0.5, 0.69
				0.6, 0.915 / 0.7, 1.2 / C. 75, 1.38 / U. 8, 1.6 / 0.84, 1.83 / 0. AA, 2.12
				U. 9, 2.3 / U. 92, 2.52 / C. 94, 2.81 / 0. 95, 2.99 / C. 96, 3.27 / 97, 7.5
				U. 98, 3.9 / C. 99, 4.6 / G. 995, 5.3 / G. 998, 6.2 / U. 999, 7.0, 9997.8
1		GENERATE	36, FN1	CREATE SHOPPERS
2		TRANSFER	POTH, .AHAY	CHECK FOR AVAILABLE BASKET
3		ENTER	BSPT	GET A BASKET
4		ASSIGN	1, FN2	DETERMINE NO. OF ITEMS
5		ADVANCE	1, FN3	SHOP
6		QUEUE	WAIT	WAIT FOR COUNTER SPACE
7		ENTER	CR1	GET COUNTER SPACE
8		DEPART	WAIT	LEAVE QUEUE
9		ADVANCE	V1	CHECK-OUT
10		LEAVE	CR1	FREE COUNTER SPACE
11		TABULATE	TRT	TABULATE TRANSIT TIME
12		TABULATE	ITM	TABULATE NO. OF ITEMS
13		LEAVE	BSPT	RETURN BASKET
14		TERMINATE	1	
15		AHAY	TERMINATE	LOST CUSTOMERS
		TIME	TABLE	TRANSIT TIME TABLE
		ITEM	TABLE	ITEM COUNT TABLE
				M1, 500, 500, 1, 0
				P1, 5, 5, 4
		CAT	STORAGE	NUMBER OF COUNTERS
		BASKT	STORAGE	NUMBER OF BASKETS
				5
				50
		2	FUNCTION	DISTR. OF NO. OF ITEMS
				MN1, D4
				.2, .5 / .5, 10 / .9, 15 / 1. C, 20
		3	FUNCTION	SHOPPING TIME DISTR.
				P1, C5
				C, 0, 75, 400 / 1 C, 900 / 15, 1500 / 20, 2250
		1	VARIABLE	CHECK-OUT TIME
				P1, 10, 0, 25
		START		INITIALIZE, SUPPRESS PRINT
		RESET	50, NP	WIPE OUT STATISTICS
		START	1000	MAIN RUN

FIGURA B

FIGURA C



VERSION DE DIRECCION-MIENTO INDICADOS DE SECCION DE 1-1-1



VERSION DE DIRECCIONAMIENTO INDIRECTO DE SECCION DE CAJAS

FIGURA D

BLOCK NUMBER	ALOC	OPERATION	A,B,C,D,E,F,G	COMMENTS
*				
*		SIMULATION OF TELEPHONE SYSTEM - MODEL 1		
*				
*		FUNCTION	PN1,C24	FUNCTION FOR 1/A INTERVAL
		G.O.D/D/O	1.0-104/G.2/C.227/C.3.D.355/D.4.D.500/D.5.0.69	
		J.O.D.	915/D.7.1.7/C.75.1.38/C.8.1.6/D.84.1.R3/D.88.2.17	
		U.9.2.3/D.97.2.57/C.94.2.41/C.95.2.99/D.96.3.21.97.3.5		
		W.98.3.9/D.94.4.6/C.995.5.3/L.958.6.2/D.999.7/D.9997.8		
*				
1		GENERATE	12,PN1	CREATE CALLS
2		TEST G	V2,Z,ABND	TEST IF SYSTEM IS FULL
3	ASN1	ASSIGN	1,V1	PICK ORIGIN
4		GATE LK	P1,ASN1	TEST FOR BUSY
5	ASN2	ASSIGN	2,V1	PICK DESTINATION
6		TEST NE	P1,P2,ASN2	RETRY IF DEST = ORIGIN
7		LOGIC S	*1	MAKE ORIGIN BUSY
8		TRANSFER	H01H,,BLKD	TRY FOR LINK
9	GETL	ENTER	LNKS	GET LINK
10		GATE LK	*2,BUSY	TEST FOR BUSY
11		LOGIC S	*2	MAKE DEST. BUSY
12		ADVANCE	120,PA1	TALK
13		LOGIC H	*1	ORIGIN HANGS UP
14		LOGIC H	*2	DEST. HANGS UP
15		LEAVE	LNKS	FREE LINK
16	TERM	TERMINATE	1	
*				
17	ABND	TERMINATE		ARABNDY CALL
*				
18	BLKD	LOGIC H	*1	ORIGIN HANGS UP
19		TERMINATE	1	BLOCKED CALLS
*				
20	BUSY	LEAVE	LNKS	FREE LINK
21		LOGIC H	*1	ORIGIN HANGS UP
22		TERMINATE	1	BUSY CALLS
*				
LNKS	STORAGE		10	NO. OF LINKS
*				
1	VARIABLE		F1=PN1/10001	PICK A LINE
2	VARIABLE		K1=2*5*LNKS-2	COUNT NO. OF FREE LINES
*				
INITIAL			F1=SC	SET NO. OF LINES
*				
START			10,NP	INITIALIZE, SUPPRESS PRINT
RESET				WIPE OUT STATISTICS
START			1000	MAIN RUN

FIGURA F

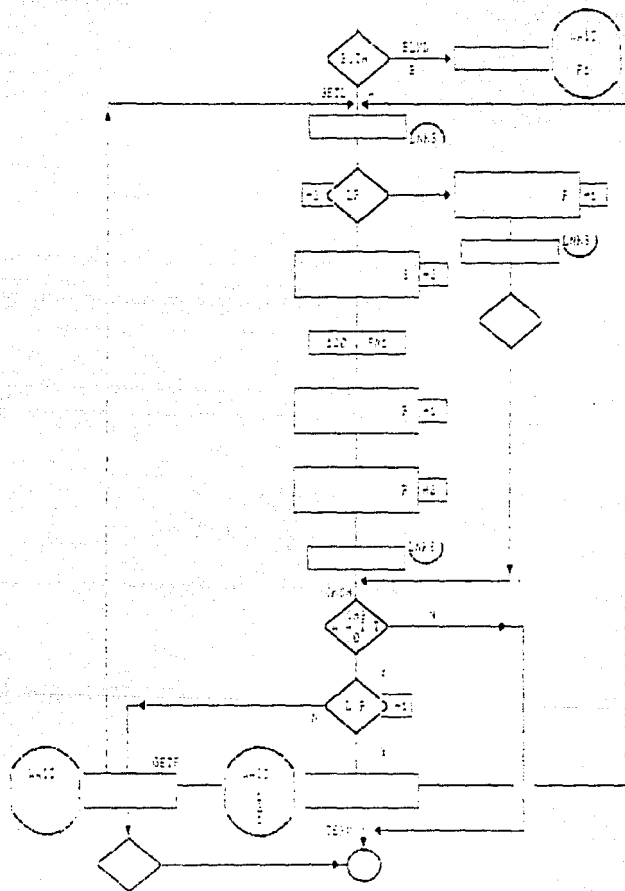


FIGURA G

BLOCK NUMBER	OPUC	OPERATION	A,B,C,D,I,F,G	COMMENTS
		SIMULATION OF TELEPHONE SYSTEM - MODEL 2		
		1	FUNCTION WNI,C24	FUNCTION FOR 1/4 INTERVAL
			C.U.C.0/0.1/G.104/L.2/G.227/G.3/G.355/D.4.3.509/O.5.0.89	
			L.D.C.915/U.7.1.2/L.75.1.30/U.8.1.6/O.84.1.83/O.88.2.17	
			U.4.2.3/C.92.2.52/L.94.2.81/U.95.2.99/O.96.3.27.97.3.5	
			0.98.3.57/C.59.4.6/O.995.5.1/O.998.6.7/O.999.7/O.9997.8	
1		GENERATE	12,FNI	CREATE CALLS
2		TEST G	V2.2,AMND	TEST IF SYSTEM IS FULL
3	ASN1	ASSIGN	1,V1	PICK ORIGIN
4		GATE LH	*1,ASN1	TEST FOR BUSY
5	ASN2	ASSIGN	2,V1	PICK DESTINATION
6		TEST NE	P1,P2,ASN2	RETRY IF DEST = ORIGIN
7		LOGIC S	*1	MAKE ORIGIN BUSY
8		TRANSFER	BOTH,,BLKD	TRY FOR LINK
9	GETL	ENTER	LNKS	GET LINK
10		GATE LP	*2,BUSY	TEST FOR BUSY
11		LOGIC S	*2	MAKE DEST. BUSY
12		ADVANCE	120,FNI	TALK
13		LOGIC H	*1	ORIGIN HANGS UP
14		LOGIC H	*2	DEST. HANGS UP
15		LEAVE	LNKS	FREE LINK
16	CRCH	TEST G	CHWAIT,0,TERM	TEST IF CALLS ARE WAITING
17		GATE LR	1,GETF	SEE IF LINE 1 IS FREE
18		UNLINK	WAIT,GETL,1,2,1,GETF	CONNECT CALL TO 1
19	TERM	TERMINATE	1	
20	GETF	UNLINK	WAIT,GETL,1	CONNECT FIRST WAITING CALL
21		TRANSFER	*TERM	
22	ABND	TERMINATE		ABANDON CALL
23	BLKD	LINK	WAIT,P1	LINK IN ORDER OF CALL ORIGIN
24	BUSY	LOGIC R	*1	CALLER HANGS UP
25		LEAVE	LNKS	FREE LINK
26		TRANSFER	*CRCH	GO TO TEST FOR WAITING CALLS
		LNKS	STORAGE 10	NO. OF LINKS
		1	VARIABLE R10PNI/100001	PICK A LINE
		2	VARIABLE R1-205BLNKS-CHWAIT-2	
			INITIAL R1,00	
		START	10,MP	INITIALIZE, SUPPRESS PRINT
		RESET		WIPF JUI STATISTICS
		START	1000	MAIN RUN

FIGURA II

ANEXO 5

ARREGLO DE COMBINACIONES	PRIMER ARREGLO	SEGUNDO ARREGLO	TERCER ARREGLO	CUARTO ARREGLO	QUINTO ARREGLO
REGLAS DE PROG.					
MAS ALTA PRIORI.					
MAS CORTO TIEMPO DE PROCESO INMINENTE.					
MINIMO ATRASO					
MAS LARGO TIEMPO EN LA LINEA					
FECHA DE VENCIMIENTO INMEDIATA					

FIGURA A

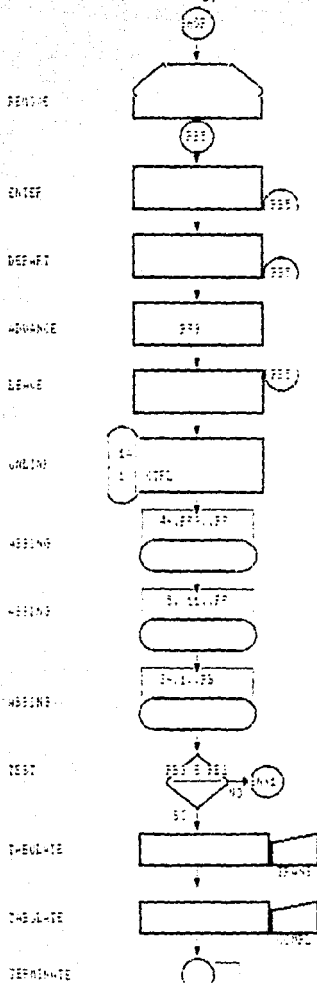


FIGURA B

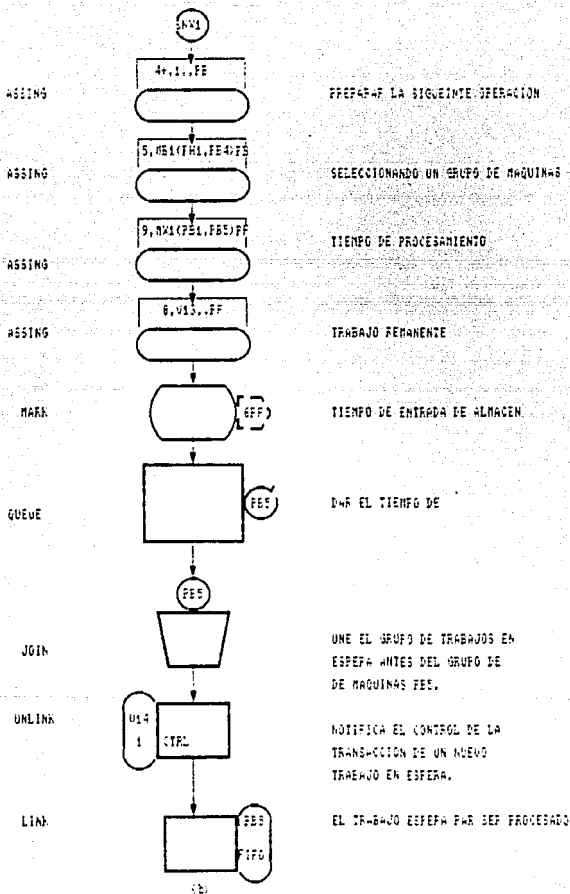
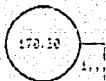


FIG. C CONTINUACION.

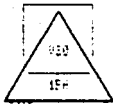
GENE-TE



UN PEE-EDG

COEF. OF 4. LOFE

SELYT



DE TR-BAJO VE

0000

1. FN-CL-EE, FE

ASSING

NUMERO DE CLASIFICACION

2. REL-FM1.11/FE

ASSING

TOTAL DE OPERACIONES

4. 1. FE

ASSING

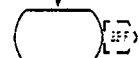
FIJAR LA PRIMER OPERACION

5. REL-FM1. FE + FE

ASSING

GRUPO DE MAQUINAS PENDIENTES

MAFI



TIEMPO DE ALMACENAJE

INICIO

10. REL. FE

ASSING

STOPF

9. REL-FM1. FE1/FE

ASSING

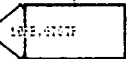
SIGUIENTE MAQUINA

1. N-1/FEL. FE1/FE

ASSING

HACE TIEMPO DE PROCESAMIENTO PARA CADA MAQUINA

LOOP



REPITE TODAS LAS OPERACIONES REQUERIDAS



OPERACION DEL IF-EM-10
FIGURA D

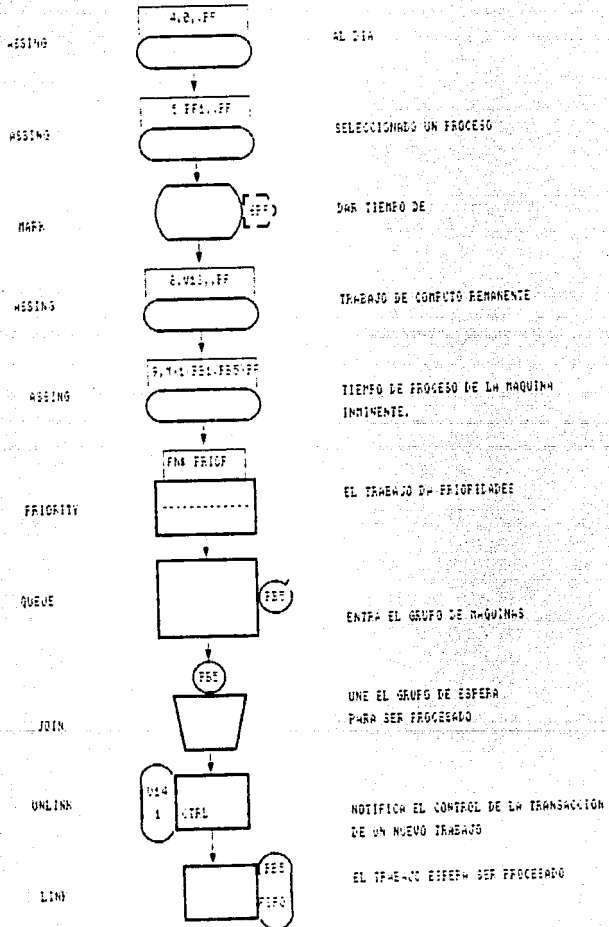


FIG. 5. PROGRAMACION


```

INITIAL W011,11,5/P0112,11,4/P0113,11,3/P0114,11,5 79
INITIAL W015,11,6/P0111,11,4/P0110,11,4/P0119-10,11,4 80
INITIAL W015,11,5/P0110,11,5 81
INITIAL W012,11,1/P0111,11,3/P0111,11,2/P0111,11,3 82
PROCESSING TIME/CLASS/OPERATION
INITIAL W011,11,10/P0111,11,20/P0111,11,30/P0111,41,40 83
INITIAL W011,11,20/P0111,11,25 84
INITIAL W012,11,14/P0112,11,10/P0112,11,40/P0112,41,45 85
86
* 87
* 88
* JCB PARAMETERS 89
* P0111 CLASS 90
* P0112 B CF OPERATIONS (P0111,11) 91
* P0113 CF OPERATIONS (P0111,11) 92
* P0114 IMPURGENT OPERATIONS 93
* P0115 MACHINE GROUP 94
* P0116 LISC 95
* P0117 FOR OPERATIONS 96
*
* P0118 SEQUENTIAL JCB B 13 TC 13 97
*
* P0119. PROCESSING TIME 98
* P0120 TIME OF GENERATION 100
* P0121 CPU TIME 101
* P0122 SUM OF PROC. TIMES OF OPERATIONS DONE 102
* P0123 REMAINING PROC. TIME (13) 103
* P0124 STORE ENTRY TIME IN CURRENT G 104
* P0125 LISC 105
* P0126 JCB SLACK (13) 106
* P0127 PC. TIME 107
*
* JCB CREATION 108
* 130,1,1...10PF,4PH,1CP0 110
*
* 131
* 132
* 133
* 134
* 135
* 136
* 137
* 138
* 139
* 140
* 141
* 142
* 143
* 144
* 145
* 146
* 147
* 148
* 149
* 150
* 151
* 152
* 153
* 154
* 155
* 156
* 157
* 158
* 159
* 160
* 161
* 162
* 163
* 164

```

FIGURA F2

0					105
0					106
0		CONTROL TRANSACTION			107
0					108
0		PARAMETERS			109
0		PTTIBASIC IN V15 TC V16			110
0		PTTIBASIC IN CHUP 0			111
0		PTTIBASIC COMPUTATIONS			112
0					113
40	GENERATE	1,000,100,1000,1000			114
41	SPLIT	2,21,100,500	0 IN PDS		115
42					116
43	CTM	CALL SWP	PPS,LLSD	PAGEING PAGE 3	117
44	TEST	CT	CP,PPS,1,CLIN	JOB IN CT	118
			TCAN WOLLS CT JCP POC WADW (P14/P21)		119
			PUT IN SWP CTM SAC		120
					121
50	SCALE	WAS	PPS,PP,PP,TPP	PRIORITY	122
51	SAVEVALUE		1,015,00		123
52	SCALE	WEN	PPS,00P,00P,TPP	PRUC. TIME	124
53	SAVEVALUE		2,010,00		125
54	SCALE	WEN	PPS,00P,00P,TPP	JOB SLACK	126
55	SAVEVALUE		2,010,00		127
56	SCALE	WEN	PPS,00P,00P,TPP	PIA ENTRY TIME IN Q	128
57	SAVEVALUE		4,010,00		129
58	SCALE	WEN	PPS,00P,00P,TPP	PIA TIME IN C (PEN ENTRY TIME)	130
59	SAVEVALUE		3,010,00	EARLIEST DUE DATE	131
					132
					133
					134
					135
					136
					137
					138
					139
					140
					141
					142
					143
					144
					145
					146
					147
					148
					149
					150
					151
					152
					153
					154
					155
					156
					157
					158
					159
					160
					161
					162
					163
					164
					165
					166
					167
					168
					169
					170
					171
					172
					173
					174
					175
					176
					177
					178
					179
					180
					181
					182
					183
					184
					185
					186
					187
					188
					189
					190
					191
					192
					193
					194
					195
					196
					197
					198
					199
					200
					201
					202
					203
					204
					205
					206
					207
					208
					209
					210
					211
					212
					213
					214
					215
					216
					217
					218
					219
					220

FIGURA F3

CAPITULO VII

CONCLUSIONES

Este trabajo que ha sido de grandes inquietudes, adolece de muchos defectos, propios de la inexperiencia del principiante; pero con él hemos tratado de cumplir la misión encomendada por nuestra facultad. Cabe mencionar que la Ingeniería Industrial atraviesa por un periodo crítico de desarrollo, de ahí que debe ser orientada hacia una función social sin circunscribirla a un campo técnico.

Los futuros Ingenieros debemos adquirir la conciencia de que somos individuos y que como tales debemos de estar en contacto permanente con el elemento más valioso de la sociedad, el hombre. De esta manera, nuestra labor alcanzará una verdadera trascendencia, no exclusivamente utilitaria, y la encaminaremos a la obtención de satisfactores para grupos humanos.

Es importante que los futuros Ingenieros tengamos presente que bajo nuestra capacidad, dinamismo y personalidad habrán de integrarse los recursos humanos, materiales y económicos, que permitan un aumento en la productividad, generando así un bienestar compartido para forjar un México más productivo, más justo y más humano.

No fue posible suministrar todas las reglas matemáticas según las cuales podríamos basarnos para el sustento del GPSS aunque sí expresamos una diversidad de principios que pueden servir como guía. No es factible describir las muchas aplicaciones que se han hecho de la simulación. Se han utilizado en virtualmente todas las ramas de la ciencia e ingeniería, al grado que en toda la literatura técnica se encuentran descripciones de estudios de simulación.

En la introducción al GPSS el enfoque que se siguió fué definir un conjunto de 43 tipos de bloques específicos, cada uno de los cuales representa una acción característica de los sistemas. El usuario del programa debe dibujar un diagrama de bloques del sistema utilizando solo estos tipos de bloques. En la programación del GPSS se dió una explicación más completa del programa, descripciones de las instrucciones de programación así como, los principios de aplicación a casos prácticos.

Se pudo comprobar la fiabilidad del programa GPSS en casos de la vida cotidiana industrial pudiendo observar el comportamiento del programa paso a paso en la solución de los problemas de simulación, los cuales son manejados efectivamente por el GPSS.

Este trabajo tiene la finalidad de presentar conceptos básicos y la teoría en general del paquete GPSS, quedando abierta la opción a la gente que así lo disponga para una mayor profundización, familiarización e investigación enfocada a la rama que se dedique.

BIBLIOGRAFIA

Introducción a la Investigación de Operaciones
HILLIER & LIEBERMAN
Editorial Mc. Graw Hill

Simulación de Sistemas
GORDON Geoffrey
Editorial Diana

Técnicas de Simulación de Computadoras
WAYLOR Thomas
Editorial Limusa

Simulation with GPSS and GPSS V
BOHILLIER Kahan
Editorial Prentice Hall

El Enfoque de Sistemas
JEREZ Victor y GRIJALBA Manuel
Editorial Limusa

Apuntes de Probabilidad y Estadística
BORRAZ IRIARTE Y FONTANA
Facultad de Ingeniería
U.N.A.M.

Apuntes de Ingeniería de Sistemas
FLORES ZAVALA Victor
Facultad de Ingeniería
U.N.A.M.