



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

**METODOLOGIA PARA AUMENTAR LA EFICIENCIA
DEL PROCESO DE TRABAJOS BATCH EN COMPUTA-
DORAS IBM CON SISTEMA OPERATIVO OS / MVS**

T E S I S

**QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION**

P R E S E N T A:

JOSE ANTONIO SILVA FERNANDEZ



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

METODOLOGIA PARA AUMENTAR LA EFICIENCIA DEL PROCESO DE TRABAJOS
BATCH EN COMPUTADORAS IBM CON SISTEMA OPERATIVO OS/MVS

INDICE GENERAL

INTRODUCCION	1
CAPITULO I	
"INTRODUCCION AL SISTEMA OPERATIVO OS/MVS"	5
CAPITULO II	
"DESCRIPCION DEL ESTUDIO DE CAPACIDAD DEL SISTEMA Y DEFINICION DEL PROBLEMA"	15
2.1 DESCRIPCION DEL SYSTEM CONTROL PROGRAM	18
2.1.1 JOB ENTRY SUBSYSTEM	18
2.1.2 INICIADORES	19
2.2 DESCRIPCION DEL PROBLEMA	25
CAPITULO III	
"PROBABILIDAD Y ESTADISTICA"	31
3.1 PROBABILIDAD Y DISTRIBUCIONES DE PROBABILIDAD	37
3.2 DISTRIBUCIONES DISCRETAS DE PROBABILIDAD	35
3.3 ESTADISTICA	36
3.3.1 DISTRIBUCIONES ESTADISTICAS	37
3.3.2 DISTRIBUCION DE FRECUENCIAS	38
3.4 GENERACION DE NUMEROS ALEATORIOS	39
CAPITULO IV	
"PROPOSICION DE METODOLOGIA"	41

CAPITULO V

MEDICION DEL RENDIMIENTO ACTUAL 45

5.1 HERRAMIENTAS DE MEDICION 46

5.2 RECOLECCION DE INFORMACION 54

CAPITULO VI

DESCRIPCION DEL MODELO DE SIMULACION 115

6.1 INTRODUCCION 116

6.2 FACTORES A CONSIDERAR EN EL DESARROLLO
DEL MODELO DE SIMULACION 120

6.3 DESCRIPCION DEL MODELO DE SIMULACION 121

6.4 DIAGRAMA DE BLOQUES EN EL LENGUAJE
DE SIMULACION ESCOGIDO 125

CAPITULO VII

*APLICACION DEL MODELO DE SIMULACION
Y ANALISIS DE RESULTADOS* 137

7.1 FACTORES A CONSIDERAR PARA LA APLICACION DEL
MODELO DE SIMULACION 138

7.2 ANALISIS DE RESULTADOS 143

CAPITULO VIII

CONCLUSIONES 175

APENDICE A

INTRODUCCION A SAS 180

APENDICE B

INTRODUCCION A GPSS	203
GLOSARIO	223
BIBLIOGRAFIA	227

I N T R O D U C T I O N :

INTRODUCCION

INTRODUCCION:

Un estudio de capacidad, como su nombre lo indica, es el total de información que todo o parte de un sistema de cómputo puede manejar, tal como el total de bits que se pueden almacenar (capacidad de memoria) o la cantidad de procesos que el sistema puede ejecutar en un intervalo de tiempo.

Los estudios de capacidad se llevan a cabo por lo general, solo en sistemas de cómputo grandes, esto es, sistemas de cómputo que manejan mucha información, así como una gran cantidad de equipo de cómputo y que tiene como finalidad tener conocimiento del aprovechamiento actual de los recursos del sistema (CPU, MEMORIA, dispositivos de E/S, etc) y ver la posibilidad de hacerlos más eficientes.

El presente estudio esta orientado a mejorar la capacidad en la cantidad de procesos batch que se pueden ejecutar en un intervalo de tiempo (throughput), esto se logra aumentando la eficiencia en la distribución de clases asignadas a los iniciadores. Los Iniciadores, son unas rutinas que asignan 'Regiones' (porciones de memoria

INTRODUCCION

de tamaño variable) a los Jobs (trabajos) que llegan al sistema, dependiendo del tipo de clase, de prioridad y los requerimientos de recursos que tenga dicho trabajo, con el fin de ser procesados.

Cada trabajo (job) que entra al sistema tiene asignada una clase, dependiendo del tipo de job que sea, esto es, por ejemplo si se trata de una compilación en fortran tiene un tipo de clase, si se trata de producción cuyo tiempo de duración es menor de 5 minutos tiene otro tipo de clase y así las demás clases se pueden asignar por duración, por cantidad de recursos que consume, etc.

Así entonces una buena distribución de clases en los iniciadores nos va a permitir tener una buena distribución de carga en el sistema y por lo tanto un mejor aprovechamiento de él.

En el capítulo I, se da una introducción general al sistema operativo OS/MVS de IBM.

En el Capítulo II, tenemos una descripción del estudio de capacidad, así como una descripción del problema y las partes del sistema que están involucrados en dicho estudio.

El Capítulo III, contiene antecedentes acerca de Probabilidad y Estadística.

INTRODUCCION

En el Capítulo IV, se define la metodología que se va a aplicar.

En el capítulo V, se lleva a cabo la medición del rendimiento actual del sistema así como una descripción de las herramientas necesarias para realizar dicha medición.

Esto nos va a permitir tener la definición del sistema así como recolectar la información que se utilizará en el modelo de simulación.

En el Capítulo VI, contiene una introducción a los estudios de simulación, así como una descripción del modelo de simulación que se llevará a cabo.

En el Capítulo VII, tenemos la aplicación del modelo de simulación, así como una corrida del programa y los resultados que arroja.

En el Capítulo VIII, se muestran las conclusiones.

El Apéndice A, contiene una introducción al lenguaje SAS (Statistical Analysis System) que fue utilizado para obtener el rendimiento del sistema.

El Apéndice B, contiene una introducción al paquete GPSS, que fue utilizado para llevar a cabo la simulación del sistema.

C A P I T U L O I

INTRODUCCION AL SISTEMA

OPERATIVO OS/MVS.

CAPITULO I

El presente estudio se realizó en un sistema 4341 de IBM, con OS/MVS.

Un Sistema Operativo (OS), es un conjunto de programas que están almacenados parte en memoria real y parte en memoria secundaria (discos).

El propósito de un Sistema Operativo (SO), es el de optimizar la administración de los recursos de una instalación de Proceso de Datos con el fin de aumentar la cantidad de trabajos que se puedan realizar en un periodo de tiempo (Throughput).

Los recursos disponibles en una instalación de proceso de datos incluye a los siguientes :

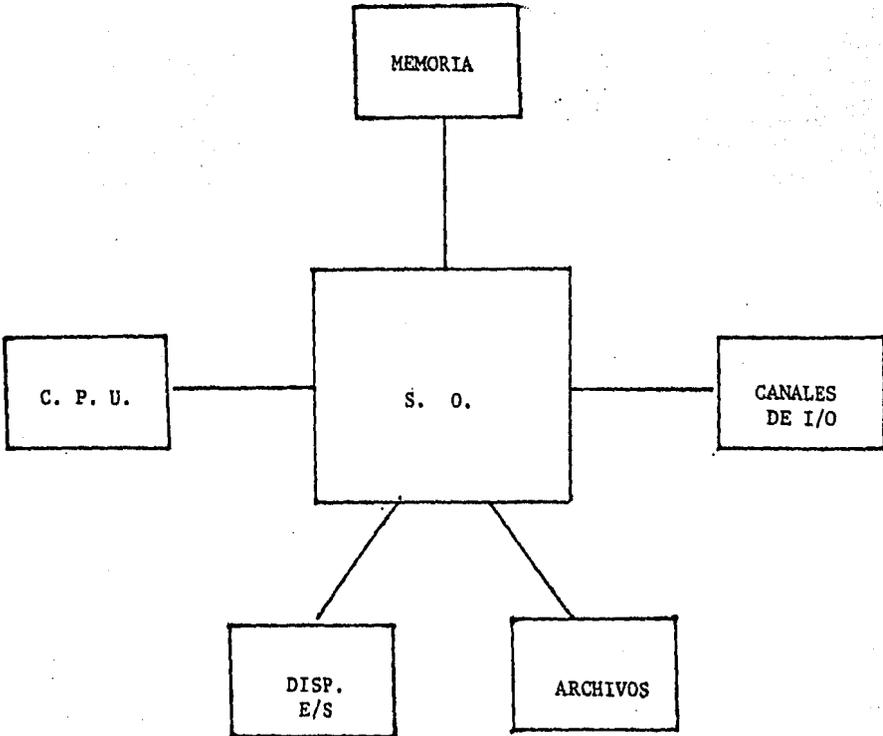
- a).-Personal.
- b).-Programas.
- c).-Datos que requieren ser procesados (archivos).
- d).-Hardware del sistema de cómputo.

-MEMORIA

-CPU

-DISPOSITIVOS de E/S

EL S.O. COMO ADMINISTRADOR DE RECURSOS



CAPITULO I

El personal pueden ser los usuarios, programadores, operadores, etc.

El personal puede comunicarse con el SO por medio de una facilidad del sistema llamada Job Control Languaje (JCL), esto permite al personal :

- Decirle al sistema que programa(s) queremos ejecutar.
- Identificar los datos requeridos por el programa.
- Especificar los recursos del sistema que requerimos.
- Especificar el tipo de prioridad para ejecutar el programa.
- Especificar que hacer con los resultados obtenidos, etc.

Otras facilidades del SO que permite reducir tiempo y espacio de almacenamiento al hacer un programa, son las llamadas "Utilerías". Estos son programas que ya existen dentro del sistema y que son soportados por el SO y que pueden ser usados por cualquier usuario.

Algunas utilerías del sistema son por ejemplo :

Sort, Copiar o imprimir información, etc. Estas facilidades

CAPITULO I

permiten que no sea necesario que cada usuario tenga una copia de dichos programas dentro del suyo evitando así tiempo y trabajo.

El SO por medio de diferentes facilidades puede atender a varios requerimientos (Jobs) para que entren al sistema seleccionando para ejecución a aquellos de mayor prioridad en forma automática. También el SO examina el Job (analiza su JCL) antes de comenzar su ejecución para determinar los recursos del sistema que éste necesita (como una cinta o un disco) y avisa al operador para asegurar que los recursos requeridos por el Job estarán disponibles cuando se necesiten.

El SO también cuenta con programas para detectar mal funcionamiento de hardware. Existen unos procedimientos que permiten que el sistema se recupere sin necesidad de parar el sistema o de intervención manual.

Los programas, son un conjunto de instrucciones que dicen al sistema de cómputo como procesar datos.

Existen también unas facilidades del sistema llamadas 'Bibliotecas' que evitan que se mezclen los programas y además permiten llevar un control de todos los programas que entran al sistema. Esto también permite que se tenga una sola copia de un programa que puede ser utilizado por mul-

CAPITULO I

tiples usuarios, ahorrando así espacio de memoria al evitar que cada usuario tenga una copia de ese mismo programa.

En el Sistema operativo OS, vamos a tener definidos dos tipos de bibliotecas:

'Bibliotecas del sistema' y 'Bibliotecas de usuarios'.

Las bibliotecas del sistema, contienen un conjunto de programas que pertenecen al sistema operativo.

Las bibliotecas de usuarios se dividen en dos tipos:

Bibliotecas Fuente y Bibliotecas objeto.

Las bibliotecas Fuente contienen todos los programas fuente que son creados por los usuarios.

Las bibliotecas objeto, contienen todos los programas en forma ejecutable (prog. objeto) de los programas de la bib. Fuente que se deseen que estén en producción.

Los Datos, son la información con que proveemos a un programa para que éste sea procesado.

Los datos se pueden organizar para su fácil y rápido acceso o almacenamiento, con otra facilidad del sistema llamada 'Data Set' (Conjunto de Datos). Data Set es un archivo de registros lógicamente organizados para ser procesados por un programa. Todos los Data Sets están localiza-

CAPITULO I

dos en un Catálogo, el cual puede ser accesado por cualquier programa.

El hardware del sistema de cómputo incluye los siguientes componentes:

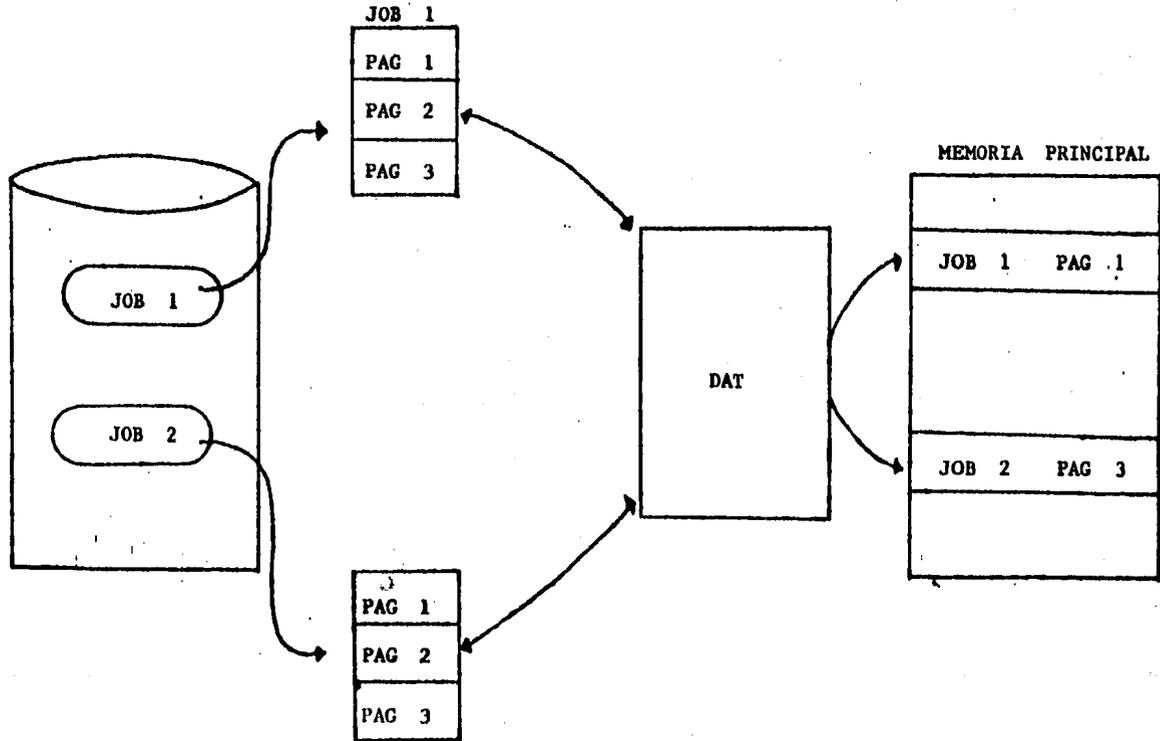
- Dispositivos de Entrada/Salida (I/O).
- Proceso de datos (CPU).
- Almacena programas y sets de datos (Memoria).
- Canales de I/O (Conectan I/O con Mem. Principal).

Funciones del SO como administrador de recursos:

- a).-Mantener un control sobre el estado de los recursos (libre ,corriendo,ocupado,etc).
- b).-determinar políticas acerca de quien obtiene el recurso, cuando y por cuanto tiempo.
- c).-Asignar el recurso.
- d).-Recuperar el recurso.

El almacenamiento virtual puede exceder la capacidad de almacenamiento real. Esto es posible haciendo que el almacenamiento virtual se haga entre dispositivos de acceso directo (DASD) y memoria real, manejado esto por un elemento de hardware llamado Dynamic Address Translation (DAT).

MEMORIA VIRTUAL



• Cada trabajo tiene la "ilusión" de que toda su información está en memoria principal.

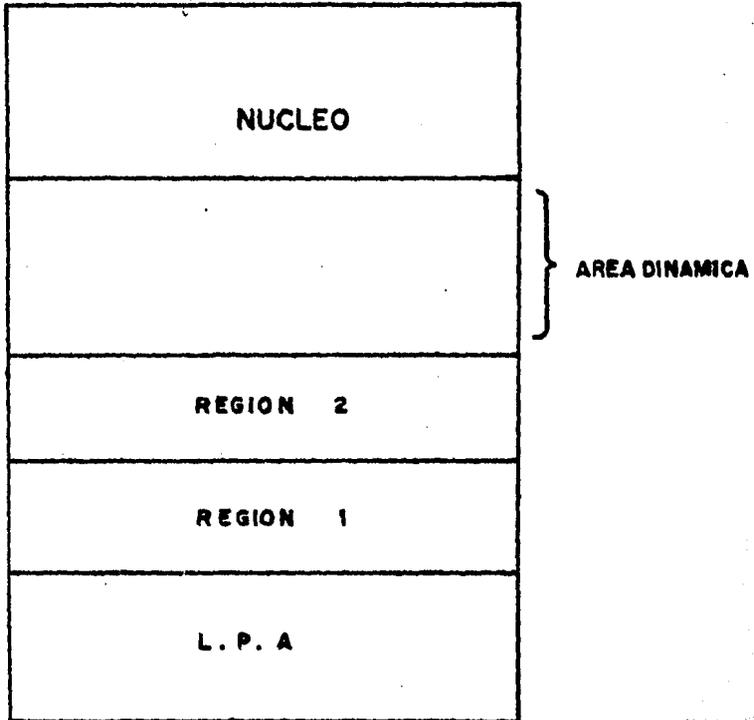
CAPITULO I

El concepto de almacenamiento virtual se basa en que solo una pequeña porción de todo el programa está activa en un determinado momento, por lo que, un programa no procesa todos los datos asociados con él, al mismo tiempo permitiendo así tener la parte inactiva en DASD y estar intercambiando porciones por medio del DAT entre memoria real y DASD. Estas porciones en que se divide el programa son llamadas páginas y tienen una capacidad de 4KBYTES.

Como se mencionó anteriormente, parte del sistema operativo se carga en memoria principal y parte en memoria secundaria (DASD). A la parte que se carga en memoria principal se le denomina "NUCLEO", y está en la parte mas alta de la memoria.

Existe otra area reservada de memoria principal llamada Link Pack Area (LPA), la cual contiene aquellos programas que son muy utilizados ya sea por el sistema o por usuarios, permitiendo reducir el numero de I/O (evita que dichos programas esten siendo cargados y descargados muchas veces).

MEMORIA PRINCIPAL



C A P I T U L O I I .

DESCRIPCION DEL ESTUDIO DE

CAPACIDAD DEL SISTEMA

DESCRIPCION:

Los estudios de Planeación de Capacidad en Centros de Cómputo se llevan a cabo, con el fin de tener una visión ordenada del sistema. Esto se debe a que en sistemas de cómputo muy grandes en los que se maneja una gran cantidad de información, el proceso para mantener a dicho sistema en niveles óptimos de eficiencia es muy complejo, ya que, necesita tomar en cuenta varios factores tales como: el personal y el equipo de cómputo (Software/Hardware), por lo que es difícil tener una idea clara y completa del sistema y de su crecimiento.

Forma parte fundamental en los estudios de Planeación de Capacidad, el análisis de la carga actual.

Para esto, debe llevarse a cabo una medición del rendimiento actual del sistema, lo que nos va a permitir tener conocimiento de la utilización del equipo como:

CAPITULO II

Conocer el consumo de los recursos (CPU, Memoria, Dispositivos de E/S, etc), así como conocer los tiempos de ejecución, de espera (colas), los tipos de proceso que se ejecutan en el sistema, las horas en las que se tiene el mayor número de procesos, etc.

La medición del rendimiento nos va a permitir tener conocimiento de la carga actual de trabajo, así como el comportamiento descrito en el crecimiento de dicha carga de trabajo.

Un factor que juega un papel muy importante en el rendimiento y por tanto en el análisis de Planeación de Capacidad, es el de tener un número apropiado de iniciadores y una buena distribución de clases dentro de cada iniciador.

Como vimos en el Capítulo I, podemos tener varios trabajos en el sistema al mismo tiempo (Multiprogramación).

Para poder llevar a cabo esto, tenemos que asignar áreas de trabajo a los procesos que se están llevando a cabo, esto es, que se divide el área de memoria principal en varias partes para asignar una parte a cada proceso (Regiones).

La manera en que esta organizado esto es por medio

CAPITULO II

de unas facilidades del S.O. llamadas Iniciadores cuyo papel es el de asignar esa area de trabajo, así como de proveer los recursos necesarios para que se lleve a cabo dicho proceso.

En este Capítulo veremos en forma mas detallada el funcionamiento de estas facilidades.

2.1 Breve descripción del System Control Program (SCP)

El SCP es un conjunto de rutinas del sistema cuyo propósito es el de optimizar los recursos del sistema (Memoria, ciclos de proceso, dispositivos de E/S, etc) .

Se listan a continuación los mas importantes.

2.1.1

Job Entry Subsystem (JES2).

El JES2 es una rutina del SCP que ayuda en el manejo del proceso de los Job's.

La secuencia de pasos en el ciclo de proceso de trabajo es la siguiente:

- 1).-Job Submission.-Se submite el trabajo que se desea procesar.

CAPITULO II

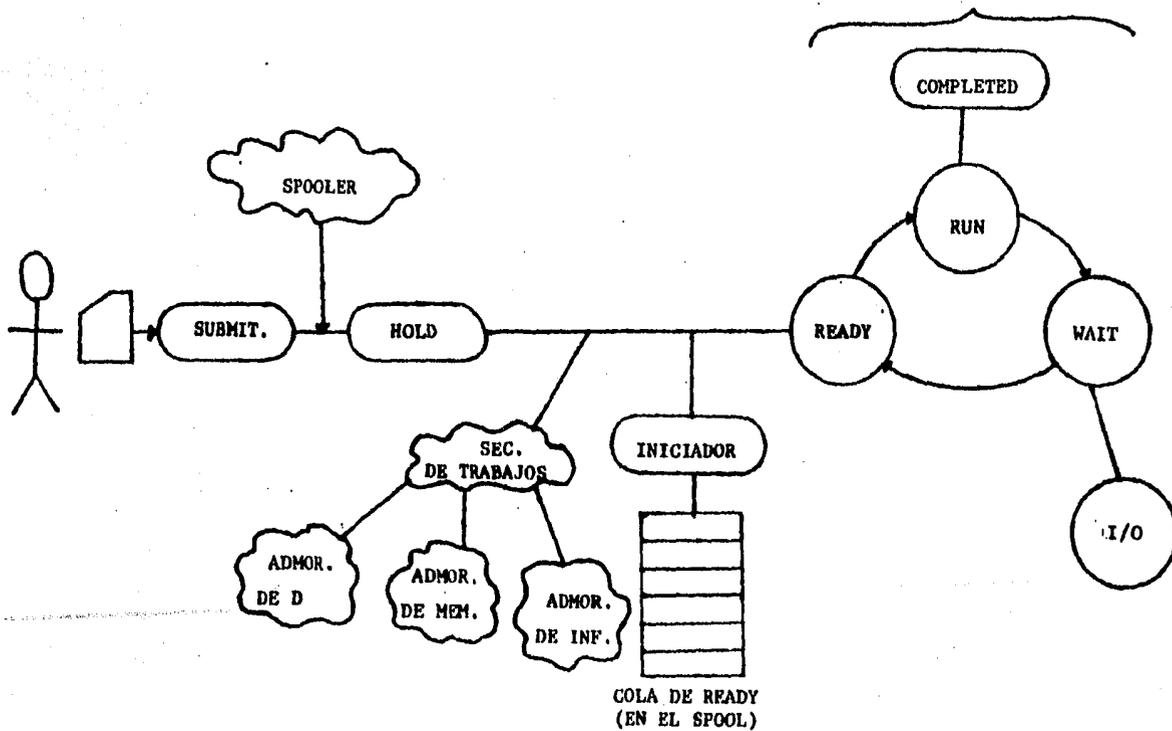
- 2).-Job Reading.- El JES2 lee la corriente de entrada (Job Stream) guardando su JCL y su Card Image Input (datos de entrada al programa).
- 3).-Job Spooling.- El JES2 almacena el JCL y el Card image input en un DASD intermedio llamado Spool Data Set.
El job espera en el Spool hasta ser seleccionado para ejecución.
- 4).-Job Selection.- El trabajo es seleccionado basado en su clase y prioridad.
- 5).-Job Execution and Print/output Spooling.- Los datos procesados se almacenan en el Spool para su posterior transcripción.
- 6).-Output transcription.- Después de que se complete el Job, los datos son recuperados del spool y son impresos.
Se borra el job y sus resultados del spool.

2.1.2

INICIADORES.

Son los responsables de preparar a un job para su ejecución.

"SECUENCIA DE PASOS EN EL CICLO DE PROCESO DE JOB'S"



CAPITULO II

El Iniciador analiza los requerimientos de recursos (Set de Datos, I/O, etc) establecidos en el JCL al principio de cada programa.

Para cada paso, el Iniciador funciona a través de un proceso llamado 'Allocation Function', el cual se encarga de obtener requerimientos de I/O y de volúmenes (cintas o disk packs) para la ejecución de cada Job Step.

Entonces el Iniciador es responsable de asignar recursos y debe estar disponible antes de que un job pueda ser seleccionado para su ejecución.

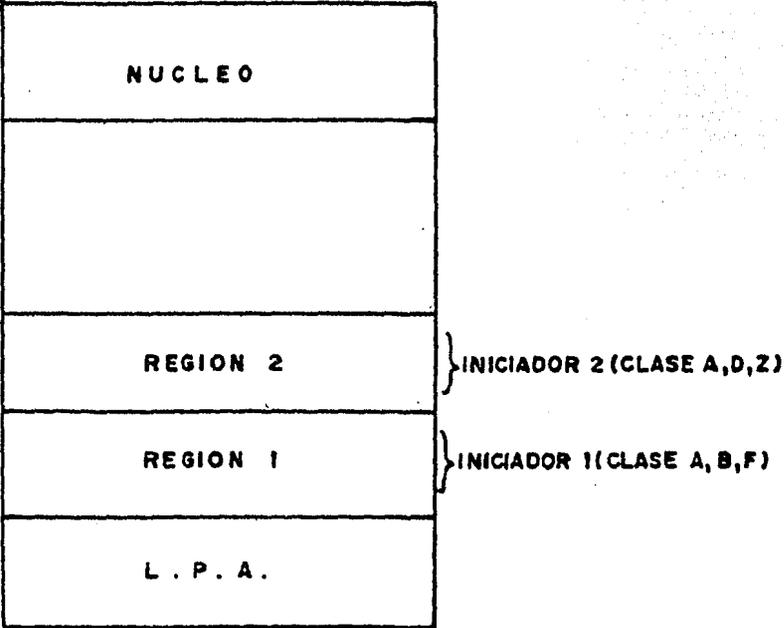
El sistema contiene cierto número de iniciadores activos en cualquier tiempo. El número de iniciadores activos es una decisión operacional.

Quando un Iniciador en particular termina de prepararse para la ejecución de un programa, el iniciador espera hasta que el programa termine. Esto es que el iniciador se libera hasta que el job concluya su ejecución. Entonces un nuevo job puede ser seleccionado para su ejecución, solo hasta que un iniciador este activo y no tenga ningún job trabajando con él, hasta ese momento otro job podrá ser tomado del spool para su ejecución.

Los Jobs son almacenados en el set de datos del spool en la forma (orden) que fueron leídos por la computadora.

ASIGNACION DE UN INICIADOR A UNA REGION

MEMORIA PRINCIPAL



CAPITULO II

Existen dos factores en adición a su secuencia de entrada para ser seleccionados para su ejecución:

- Job Class
- Job Selection Priority.

Job Class.- Cada job contiene una descripción del tipo de trabajo que realiza, o de los recursos que necesita especificados en el JCL como Job Class. Cada instalación establece los estándares usados para cada Job Class, dependiendo del número de cinta que utiliza, los ciclos de CPU o el tamaño de partición utilizada.

Cada iniciador al ser arrancado, o durante el día de ejecución, es preparado para seleccionar Jobs de determinadas clases.

El iniciador debe ser preparado para tratar con Jobs de clase A, y se le puede decir que si no hay un clase A en el pool, que prepare un clase X.

La asignación de clase y su orden de importancia son establecidos por cada iniciador.

Si le damos el Submit a un job clase C, comenzará su ejecución solo cuando un iniciador que trata con clase C esté buscando a un job de ésta clase.

CAPITULO II

Job Selection Priority.- Este valor determina la importancia relativa de un job comparado con otro job de la misma clase.

Los jobs son traídos para ejecución (del spool) cuando tienen la mayor prioridad y el iniciador de este tipo de clase esté disponible.

Entre mayor sea el número mayor será la prioridad.

Una vez que un job es seleccionado para ejecutarse su Job Class y su Job Selection Priority no son usadas más por el sistema.

Terminator.- Después de que cada programa termina, un programa llamado Terminator borra cualquier trazo del programa en el sistema y libera todos los recursos utilizados.

En resumen el JES pone al Job Stream en el sistema y lo localiza en el spool. El iniciador entonces, cuando esté disponible para comenzar un job, selecciona el de mayor prioridad en su clase. El iniciador aparta los recursos (I/O, etc) y permite a un programa llamado Fetch que traiga unos programas de usuario específicos. Al finalizar la ejecución del programa, el Terminator

CAPITULO II

libera los recursos y el iniciador puede empezar a trabajar otra vez.

2.2 Descripción del problema.

Una forma de representar la forma en que se almacenan los Jobs al llegar al sistema, es la siguiente:

Partiremos de que vamos a tener una serie de llegadas de varios trabajos (Jobs) al sistema.

Su tarjeta de control de JCL es la siguiente:

```
//X JOB CLASS=E,PRTY=(1..15)
```

en donde x sería el nombre del job.

Ejemplo de las llegadas de varios Jobs.

```
//Y JOB CLASS=E,PRTY=5
```

```
//M JOB CLASS=A,PRTY=1
```

```
//N JOB CLASS=A,PRTY=10
```

Estas tarjetas son analizadas por el 'JOB MANAGEMENT' y las almacena en el SPOOL (almacen temporal), considerando su clase y prioridad por medio del 'SCHEDULING PRIORITY' de manera semejante al de la tabla 2.2.1

En ese ejemplo podemos apreciar, que en la clase A tenemos 2 llegadas con diferente prioridad. En este

ORGANIZACION DE TRABAJOS ASIGNADOS EN EL SPOOL

PRIORIDAD DE EJECUCION	3					
	2	M				
	1	N				Y
		A	B	C	D	E
		CLASE				

Tabla 2.2.1

CAPITULO II

caso tendremos que se ejecutará primero el Job M ya que tiene mayor prioridad que el Job N.

Como se explicó anteriormente, el papel que juegan los Iniciadores en la capacidad del sistema es muy importante.

En la actualidad la forma en que se elige el número de iniciadores, así como la distribución de clases, se realiza sin ninguna secuencia y se confía en la decisión de una o varias personas que confían en su intuición.

En base a experiencia se a llegado a la conclusión de que el mejor número de iniciadores (en este sistema) es de seis. Esto se hace con el fin de proteger el nivel de multiprogramación (número de trabajos que se están procesando en forma concurrente) del TRASHING, osea que cuando se tienen muchos procesos, el tiempo de CPU que tiene cada proceso (tarea) es mas pequeño, y el sistema pierde mas tiempo moviendo páginas entre memoria virtual y memoria secundaria (SWAPPING) que procesando trabajos.

Partiremos de que estos iniciadores se asemejan a una matriz en donde cada renglón es un iniciador y cada columna es una clase (ver figura 2.1).

**MATRIZ DE ASIGNACIONES
DE CLASES A INICIADORES**

CLASES

INICIADORES

	1	2	3		b
1	A	b	D		
2	E	A	F		
3	G	A	D		
h					

Fig. 2.1

CAPITULO II

Se tienen que considerar dos tipos de prioridades en cada iniciador:

La primera prioridad es con respecto a la columna. Esto nos dice que la clase que tiene mayor prioridad en ese iniciador es la que está en la primera columna, la que le sigue es la que esta en la segunda columna, y así sucesivamente.

La segunda prioridad es la que tiene cada Job dentro de una misma clase. Esto es que pueden llegar dos Jobs al sistema. En el momento que se libere un Iniciador que acepte esa clase, este se asignará al Job que tenga la mayor prioridad. En caso de que los dos Jobs que estan esperando tengan la misma prioridad, se asignará el iniciador al Job que haya llegado antes al sistema.

El propósito de ésta tesis es el de proponer un modelo que simule esta parte del sistema, y asignar de diferentes maneras las clases en cada iniciador (primero las mas frecuentes, primero las mas cortas, etc.) analizando como se comporta el sistema en cada una de las diferentes asignaciones.

Al final se elige la distribución que nos de los menores tiempos de espera, así como la mayor ocupación

CAPITULO II

de los recursos del sistema.

El obtener la mejor distribución de clases en el sistema, nos va a dar como resultado una mejora en la producción, así como un mejor aprovechamiento en la capacidad del sistema.

C A P I T U L O I I I .

PROBABILIDAD Y ESTADISTICA

CAPITULO III.

PROBABILIDAD Y ESTADISTICA :

Quando deseamos llevar a cabo la simulación de un sistema real, a través de un modelo matemático, es importante tomar en cuenta unos conceptos básicos de probabilidad y estadística, ya que ellos nos van a permitir definir de forma adecuada las variables que servirán de entrada en el modelo de simulación, así como analizar las variables de salida de dicho modelo de simulación.

En los problemas de toma de decisiones, a menudo se tiene la necesidad de tomar dicha decisión en base a fenómenos asociados con la incertidumbre. Esta incertidumbre resulta de la inconsistencia o la variación en los fenómenos naturales. Esta variabilidad puede ser incorporada al modelo matemático y manejarse cuantitativamente. En general, esto puede llevarse a cabo si los fenómenos exhiben cierto grado de regularidad, de modo que su variación puede ser descrita mediante un modelo de probabilidad.

Las siguientes notaciones son las utilizadas en los

CAPITULO III.

modelos probabilísticos :

Ω - Espacio muestra.- El conjunto de todos los resultados posibles;

ω - Punto.- Cada resultado en el espacio muestra;

E - Evento.- Es el conjunto de los resultados de un experimento;

X - Es una variable aleatoria.- Es una función de valor numérica definida sobre el espacio muestra (es una regla número a cada valor del dominio o sea del espacio muestra);

x - Las mayúsculas es notación de las variables aleatorias, y las minúsculas denotan los valores que toma la variable aleatoria.

3.1 Probabilidad y distribuciones de probabilidad.

La probabilidad de un evento es una medida del grado de confianza que se tiene de que ese evento ocurra al realizar un experimento.

Generalmente es difícil obtener estos valores en forma numérica, pero sin embargo, puede postularse su existencia.

CAPITULO III.

Una forma de determinar una probabilidad es en base a la frecuencia :

$P\{E\} = \lim M/N$ cuando N tiende a infinito y en donde N es el número de veces que se efectuó el experimento y M es el número de éxitos del evento.

Asociada a cada variable aleatoria existe una Función de Distribución Acumulada (FDA).

La FDA se denota por :

$$F(b) = F_X(b) = P\{w/X(w) \leq b\} = P\{X \leq b\}.$$

que es el conjunto de los resultados w en el espacio muestra, tales que la variable aleatoria X toma todos los valores menores o iguales a b .

En la mayor parte de los problemas que se encuentran en la práctica, no se tiene interés en los eventos del espacio muestra y sus probabilidades asociadas. Por el contrario, el interés se enfoca en las variables aleatorias y sus Funciones de Distribución Acumuladas asociadas.

Generalmente se elige una variable aleatoria y se establece una hipótesis acerca de la forma de la FDA, o acerca de la variable aleatoria. Estas suposiciones no se hacen en forma arbitraria, en realidad implican suposicio-

CAPITULO III.

nes acerca de las probabilidades asociadas con los eventos del espacio muestra. Estas suposiciones pueden ser verificadas ya sea por evidencia empírica, o bien, por consideraciones teóricas.

3.2 Distribuciones discretas de probabilidad.

Las variables aleatorias discretas, son aquellas que toman un conjunto de valores finito o infinito, pero contable.

Tenemos entonces que la FDA para una variable aleatoria discreta se puede definir como :

$$F_X(b) = \text{Sum } P\{X=K\}, \text{ para toda } K \leq b$$

en donde K es un índice que varía para todos los valores posibles que puede tomar la variable aleatoria X .

CAPITULO III.

3.3 Estadística.

Un método estadístico es cualquier técnica utilizada para obtener, analizar y presentar datos estadísticos.

Los elementos de la técnica estadística incluyen :

- 1.- La recolección y resumen de datos;
- 2.- La clasificación y resumen de los datos;
- 3.- La presentación de los datos en :
 - a) Forma directa;
 - b) Forma tabular;
 - c) Mediante gráficas;
- 4.-El análisis de los datos.

Dentro de las características y limitaciones de los métodos estadísticos tenemos los siguientes :

- 1.- Los métodos estadísticos, constituyen el único medio para manejar grandes masas de datos numéricos.
- 2.- Las técnicas estadísticas solo se pueden aplicar a datos que sean reducibles a una forma cuantitativa.
- 3.- Las técnicas estadísticas son objetivas. Sin em-

CAPITULO III.

bargo los resultados estan afectados por la interpretación necesariamente subjetiva.

4.- Las técnicas estadísticas son idénticas tanto para las ciencias sociales como para las ciencias físicas.

3.3.1 Distribuciones estadísticas (series).

Antes de analizar los datos numéricos, es necesario disponerlos sistemáticamente. Los datos pueden disponerse en cierto número de formas diferentes. Tal disposición se denomina técnicamente, "DISTRIBUCION" o "SERIE".

tenemos diferentes tipos de distribuciones segun la forma en que se agrupan los datos :

- 1.- Magnitud -Distribución de frecuencia.
- 2.-Tiempo de realización -Serie Cronológica.
- 3.-Localización geográfica -Distribución espacial.

CAPITULO III.

3.3.2 La distribución de frecuencias.

La distribución de frecuencias es una disposición de los datos numéricos según su magnitud.

Se crea una tabla en la que se agrupan los datos informando cuantos de ellos están comprendidos entre determinados límites.

Estos límites se pueden fijar de dos formas :

Una es tomando el rango (la diferencia entre el mayor y el menor valor de los datos) y dividiendo entre el número de intervalos de clases (límites) deseados (empíricamente se calcula entre diez y veinte).

La otra forma de calcular estos límites, es mediante fórmula de Sturges:

$C = \text{rango} / (1 + 3.22 \log n)$. $n = \text{número de observaciones.}$

CAPITULO III.

3.4 Generación de números rectangulares.

En todos los experimentos de simulación existe la necesidad de generar valores de variables aleatorias que representan a una cierta distribución de probabilidad. Durante un experimento de simulación el proceso de generar una variable aleatoria de una distribución particular, puede repetirse tantas veces como se desee y tantas veces como distribuciones de probabilidad existan en el experimento de simulación. Sin embargo es conveniente señalar, que el proceso de generación de variables aleatorias no uniformes se hace a partir de la generación de números rectangulares.

La importancia de los números rectangulares (distribución uniforme) radica en su uso para la generación de variables aleatorias más complicadas que son requeridas en los experimentos de simulación. Algunos autores como Tocher han sugerido tres formas para obtener números rectangulares: La provisión externa, La generación interna a partir de un proceso físico al azar, y la generación interna de sucesiones de dígitos por medio de una relación de recurrencia.

El primer método implica tener los números aleatorios.

CAPITULO III.

como por ejemplo las tablas de Rand, en una cinta magnética o en un disco y tratar a estos números como datos de entrada para el problema que se está simulando. El segundo método implica utilizar un aditamento especial de la computadora digital capaz de registrar los resultados de un proceso aleatorio, y además que reduzca esos resultados a sucesiones de dígitos.

El tercer método y uno de los más utilizados, implica la generación de estos números rectangulares a través de una relación de recurrencia.

Independientemente del proceso que utilice para la generación de números rectangulares, estos deben poseer ciertas características deseables que aseguren o aumenten la confiabilidad de los resultados obtenidos en la simulación.

- 1.-Uniformemente distribuidos.
- 2.-Estadísticamente independientes.
- 3.-Reproducibles.
- 4.-Periodo largo (sin repetición dentro de una longitud determinada de la sucesión).
- 5.-Generados a través de un método rápido.
- 6.-Generados a través d un método que no requiera mucha capacidad de almacenamiento de computadora.

C A P I T U L O I V :

PROPOSICION DE METODOLOGIA

CAPITULO IV.

METODOLOGIA .

Descripción de la Metodología para la asignación de clases a iniciadores.

Como se explicó en capítulos anteriores, la forma en que el sistema procesa los Job's (Trabajos) Batch que llegan a él, es en base a unos programas llamados iniciadores los cuales se encargan de reservar los recursos del sistema (memoria, archivos, etc), que el Job requiere para ser procesado.

Estos Iniciadores, van a asignar los recursos dependiendo del tipo de clase que puede procesar cada iniciador.

1.-Definición del sistema.

Se hace un análisis preliminar del sistema, con el fin de determinar su estado actual, sus restricciones y las variables que interactúan dentro del sistema. permitiendonos definir que tipo de resultados deseamos obtener.

Este punto esta especificado en el Capítulo V.

CAPITULO IV.

2.-Formulación del modelo.

Una vez que están definidos los resultados que se esperan obtener del estudio, el siguiente paso es definir y construir el modelo con el cual se obtendrán los resultados deseados. En la formulación del modelo es necesario definir todas las variables que forman parte de él, sus relaciones lógicas y los diagramas de flujo que describan en forma completa al modelo.

Este punto corresponde al Capítulo VI.

3.-Recolección de datos.

Es posible que la facilidad de obtención de algunos datos o la dificultad de conseguir otros, pueda influenciar en el desarrollo y formulación del modelo.

Por consiguiente es muy importante que se definan con claridad y exactitud los datos que el modelo va a requerir para obtener los resultados deseados.

Normalmente la información requerida para un modelo se pueda obtener de registros contables, de órdenes de trabajo, de órdenes de compra, de opiniones de expertos o si no hay otro remedio por experimentación.

Este punto corresponde al Capítulo V.

CAPITULO IV.

4.-Implementación y experimentación del modelo de simulación.

Una vez que se a definido el modelo de simulación, el siguiente paso es elegir el lenguaje de simulación a utilizar, implementarlo y después pasar a la experimentación.

Esto corresponde al Capítulo VII.

7.-Interpretación.

En esta etapa del estudio, se interpretan los resultados que arroja la simulación y en base a ésto se toma una decisión. Es obvio que los resultados que se obtienen de un estudio de simulación ayudan a soportar decisiones del tipo semiestructurado, es decir, la computadora en si no toma la decisión, sino que la información que proporciona ayuda a tomar mejores decisiones y por consiguiente a sistematicamente obtener mejores resultados.

C A P I T U L O V

MEDICION DEL RENDIMIENTO ACTUAL

MEDICION DEL RENDIMIENTO

La medición del rendimiento nos va a permitir definir el sistema. Esto significa que podremos conocer el estado actual del sistema, las restricciones que existen, así como las variables que interactúan dentro del sistema, permitiéndonos establecer los resultados que deseamos obtener.

La medición del Rendimiento del Sistema, es una de las herramientas principales en el estudio de Planeación de Capacidad, ya que nos permite conocer la carga actual de trabajo así como los requerimientos actuales del sistema, analizando así el crecimiento en dicha carga de trabajo.

5.1 Herramientas de medición.

Para llevar a cabo la Medición del Rendimiento en un sistema I.B.M. existen dos programas monitores : 'SMF' y 'RMF'.

El SMF (System Management Facilities) forma parte

CAPITULO V.

del Sistema Operativo, y aparte de ser un monitor bastante completo tiene un consumo muy bajo de Registros Físicos.

El SMF es un monitor manejador de eventos que tiene su propio Set de Datos en los cuales escribe varios registros cuando ciertos eventos ocurren.

Los registros del SMF son de longitud variable y cada uno contiene un registro que indica: Cuando es movido dicho registro al buffer del SMF. Cuando el buffer se llena, se escribe en un set de datos de una unidad de disco llamada "SYS1.MANX", y a su vez cuando este set de datos se llena, aparece un aviso al operador y comienza a llenarse otro set de datos llamado "SYS1.MANY", en este momento se copia el contenido de los dos sets de datos (MANX y MANY) a una cinta y se da un reset al SMF para comenzar a escribir de nuevo en el set de datos "SYS1.MANX".

El tamaño de cada uno de los dos buffers es de 4096Bytes, lo que hace un total de 8192Bytes.

El RMF (Resource Measurement Facilities), es también una herramienta poderosa además de barata.

Esta tiene la particularidad de que registra la

CAPITULO V.

utilización de los recursos del sistema.

Para efectuar un análisis sobre los eventos registrados por SMF y RMF, utilizaremos un lenguaje llamado SAS (Statistical Analysis System), debido a sus características de fácil comprensión y poderosa aplicación. (Para mayor información sobre el lenguaje SAS ver Apéndice A).

La estructura básica para analizar los registros SMF y RMF, es la creación de un Set de Datos SAS.

La información relevante de la utilización del sistema está contenida en 46 tipos de registro SMF y 10 tipos de registro RMF.

Podemos definir tres diferentes miembros para la creación de Sets de Datos SAS :

1) Miembros cuyos nombres empiezan con TYPE, procesan datos de SMF/RMF y crean uno o dos set de datos SAS. Por ejemplo :

TYPE434 lee un archivo SMF y selecciona los registros 4 y 34 (escritos cuando termina un Job Step en Batch o un Job Session en TSO), y construye un set de

CAPITULO V.

datos SAS llamado TYPE434.

2) Miembros cuyos nombres empiecen con MAC son SAS MACROS, que permiten la creación de varios sets de datos SAS en una pasada de los registros SMF/RMF.

3) Los demás miembros realizan análisis usando sets de datos contruidos por miembros TYPE o MAC.

Los datos de SMF/RMF se pueden catalogar por medio de un registro de identificación 'ID'. Lo primero que hay que hacer es asociar cada ID (Registro de Identificación), del set de datos que va a ser construido con su miembro correspondiente. Algunos registros pueden ser asociados por un tipo de recurso dado como : CPU, Memoria, Dispositivos Periféricos, etc., en cambio otros se clasifican en términos de el uso de datos contenidos o por hacer consumo excesivo de algun recurso.

Algunas variables de tiempo utilizadas para medir duraciones, inicio o terminación de eventos, se describen a continuación:

Variabes terminadas en TIME son TIME STAMPS o tiempo marca en el cual se registra inicio o terminación de algún suceso reelevante en el transcurso de un

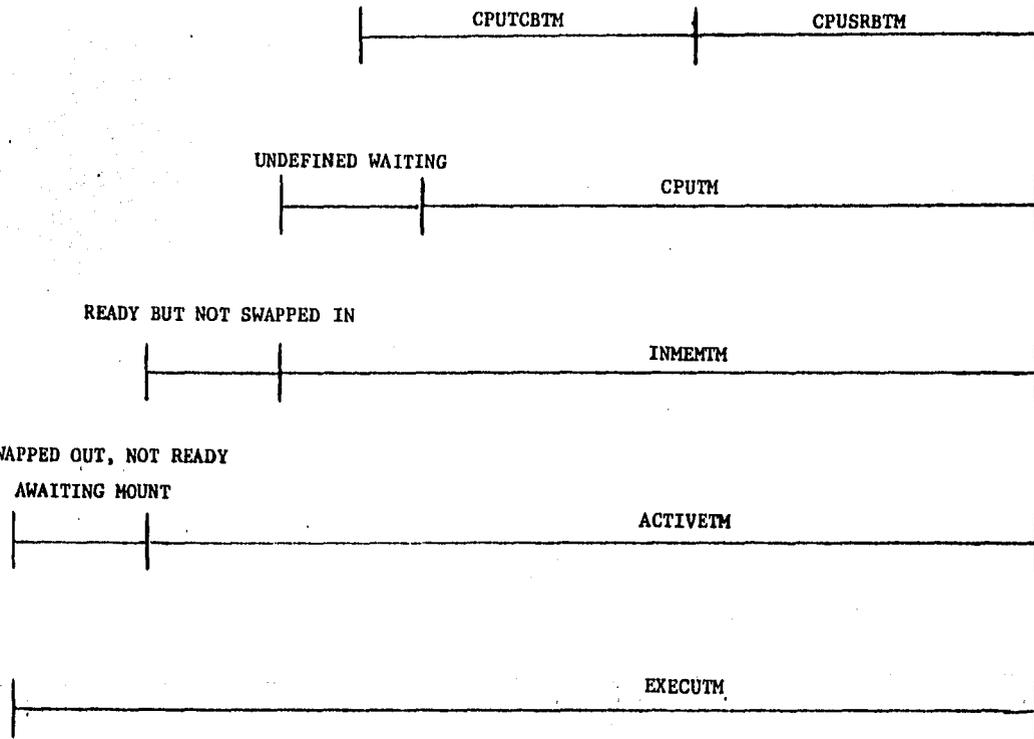
CAPITULO V.

evento.

VARIABLES terminadas en TM define 'duraciones' de eventos, esto es por ejemplo, el tiempo transcurrido entre el inicio y la terminación de un evento. La figura 5.1 muestra las variables de tiempo utilizadas.

Breve descripción de las variables de tiempo típicas de un Job Step :

- INITTIME.- Tiempo de iniciación para el Step;
- ALOCTIME.- Marca el inicio de la asignación de recursos;
- DSENQTM .- Todos los Set de Datos en cola son dictados o emitidos durante este intervalo en el primer Step para todos los Set de Datos que serán utilizados por el Job. Además en éste intervalo los bloques de control construidos por el convertidor son examinados por lo que puede terminar la ejecución aquí debida por ejemplo a un error de JCL tal como 'Set de Datos no encontrado';
- ALOCTM' .- Duración de asignación de recursos;
- LOADTIME.-Es el tiempo cuando fueron satisfechas todas las asignaciones (no necesariamente montadas) y el Job Step comienza su ejecución;



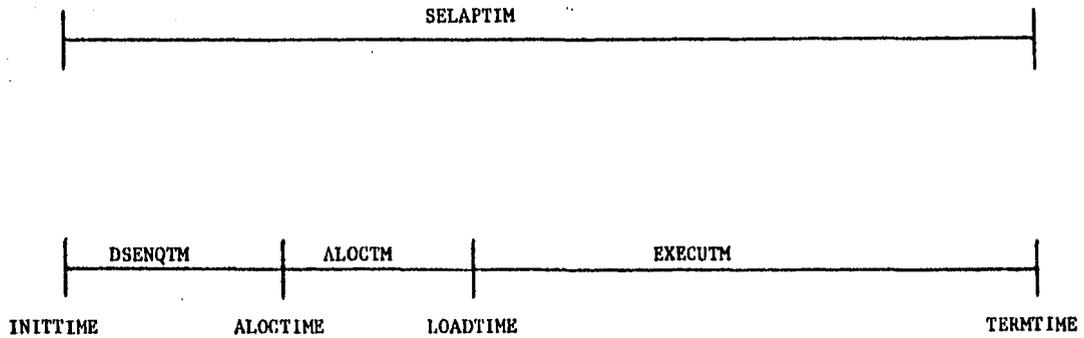


FIG. 5.2

CAPITULO V.

- TERMTIME.- 'Time Stamp' de terminación para esta ejecución;

-EXECUTM.- Es el máximo tiempo de ejecución posible para este Step, o sea que el programa tuvo acceso al procesador a lo más la duración de EXECUTM;

-SELAPTM .- Es la duración en que el Step estuvo en iniciación (TERMTIME-INITTIME);

La figura 5.2 compara varias duraciones acumuladas con la duración EXECUTM.

El contador de tiempo acumulado del ACTIVETM (fig 5.2) es mantenido por el 'System Resource Manager' (SRM) y siempre será igual o menor a EXECUTM.

ACTIVETM es el tiempo que el SRM reconoce a este Job Step como activo (elegible para ejecutarse).

ACTIVETM puede subdividirse en el tiempo en que la tarea estuvo activa en memoria, o que fue residente de memoria, y el tiempo en que la tarea se consideró como activa, aunque no fue residente de memoria (SWAPPEDIN).

La porción de ACTIVETM en la cual la tarea fue residente, se acumula en una variable llamada INMEMTM, o sea la duración en la que este Job Step estuvo en

CAPITULO V.

memoria real.

Para algunas porciones de este tiempo de residencia, el Job Step estuvo ejecutando instrucciones.

El tiempo de CPU acumulado bajo el Task Control Block (TCB) es acumulado por el SMF en la variable CPUTCBTM, mientras que el tiempo de procesador acumulado bajo el Task Service Request Block (SRB) es acumulado en CPUSRBTM.

La suma de CPUTCBTM y CPUSRBTM es entonces CPUTM, la cual representa el total de tiempo de CPU usado por este Job Step.

5.2 Recolección de información.

Como hemos visto, tanto el RMF como el SMF coleccionan muchos eventos ocurridos en el sistema, y los almacenan en unos Buffers los cuales son vaciados en unas cintas.

Nosotros podemos acceder la información que consideremos reelevante por medio de unos programas en SAS.

Con la información obtenida se pueden generar unas gráficas las cuales nos van a permitir analizar la carga

CAPITULO V.

actual del sistema.

La asignación de clases a procesos es la de la tabla 5.2.1.

Se debe considerar la asignación de clases que actualmente rige en el sistema para posteriormente poder hacer una comparación.

La asignación de clases actual en el sistema es la siguiente:

INIC1	R C J L V G D
INIC2	C J G V L R D
INIC3	C J V G R L D
INIC4	G I N F I P M S
INIC5	G I S P M F I
INIC6	H

Para llevar a cabo la recolección de eventos se deben tomar en cuenta las siguientes consideraciones:

Se debe tomar como mínimo una muestra de 30 días y se deben separar las muestras obtenidas por hora y por día.

Esto se logra con el programa SAS el cual muestra algunas de las gráficas obtenidas.

T A B L A 5. 2. 1.

Asignación de clases a los tipos de procesos existentes en la instalación existente.

<u>CLASE</u>	<u>TIPO</u>	<u>TIPO DE PROCESOS</u>
		<u>D E S C R I P C I O N</u>
R	Reproceso	Proceso de producción que se tiene que volver a correr.
L	Prod. Largo	Proceso de producción cuyo tiempo de CPU de duración excede de 5 minutos.
C	Prod. Corto	Proceso de producción cuyo tiempo de CPU de duración máxima es de 5 minutos.
V	Utilería	Utilities en general.
G	Compilación Cobol	Compilación Cobol. Compilación Cobol.
F	Compilación Fortran	Compilación Fortran. Compilación PL 1
I	Compilación PL 1	
P	Prueba	Procesos que se encuentran en desarrollo y que serán liberados una vez - concluida la etapa de pruebas.
S	SAS/PL1	Procesos que utilizan programas SAS o PL1
N	BMP	Procesos batch que corre en la región de control de IMS, éste puede ser de pruebas o de producción.
D	Prod. Largo	Proceso de producción largo que utiliza más de una cinta.
J	Prod. Corto	Proceso de producción corto que utiliza más de una cinta.
H	IMS	Pruebas de IMS (DB)
M	Prueba	Pruebas que utilizan más de una cinta.
l	Soporte Técnico	

CAPITULO V.

El primer tipo de gráficas obtenidas nos permiten hacer un análisis diario por hora, acerca del promedio de llegadas de jobs al sistema. Esta variable nos permite obtener las horas pico (de más trabajo), así como los días con mayor carga.

El segundo tipo de gráficas, nos permiten conocer el promedio de llegadas al sistema de cada clase por hora en una muestra de varios días.

El tercer tipo de gráfica RUNTIME nos va a dar el promedio de duración de cada clase de job, esto es, el tiempo promedio que cada tipo de clase tuvo asignada una Región.

El cuarto tipo de gráfica nos da un promedio del tiempo de espera por clase. Este es el promedio de tiempo que tuvo que esperar cada clase para que se le asignara un iniciador.

CAPITULO V.

JCL NECESARIO PARA EJECUTAR EL PROGRAMA
SAS QUE SE UTILIZA PARA LA RECOLECCION
DE INFORMACION EN EL SISTEMA.


```

1 //BABL014H JOB (0007,VI,PCACHE, JCB 4711
// CLASS=V,MSGCLASS=C 000024P
// MSGLEVEL=(1,1),NOTIFY=PARLO14,DDPLY=0 00000351
2 //MSG EXEC ATC 00000451
3 XXWTO PPRC CLASS=***, 00000010
KA MSG=***, 00000010
*** 00000010
*** LIB PPRC,PROCLIB(WTO) 00000070
*** GOI MVS/SP SYSTEM IPL OPERATIONS GUIDE 00000070
*** DOC THIS PROCEDURE EXPLAINS THE PROGRAM WTD. IT SHOWS HOW 00000070
*** OR HOW PLACES TO BE GOAT TO THE OPERATOR. THE MESSAGE 00000070
*** MAY BE PLACED DIRECTLY AFTER THE EXEC STATEMENT. IT ONLY 00000070
*** THE MESSAGE TO BE RE SENT. IT MAY BE OMITTED IN THE PARA 00000070
*** FIELD OF THE EXEC STATEMENT. 00000070
*** 00000070
*** MSG= DISPLAYS THE LINE TO OPERATOR NOTE * - *** 00000070
*** CLASS= PRINT CLASS 00000070
*** 00000070
4 XXIFPRRC EXEC PGM=ICL, 00000070
PARM=CP550, 00000070
5 XXSYSPRINT DD SYSOUT=CLASS,CCO=(RECF=FB,LF=1,PKSIZ=1600) 00000070
6 //SYSA DD * GENERATED STATEMENT 00000070
7 XXSYSA DD DDNAME=ICLPRC 00000070
XXIFPRC DD DDNAME,DD=PKSIZ=80 00000070
*** 00000070
*** PUT MESSAGES TO OPERATOR HERE IF YOU WANT ONE... 00000070
8 //SAS1 EXEC SAS 00000070
9 XXSAS PPRC CP1PRC=4,CP2PRC=4,CP3PRC=4,CP4PRC=4,CP5PRC=4, 00000070
XX LGFC=PRINT,PGM=CP550,LP=STOR 00000070
***** 00000070
*** STATISTICAL ANALYSIS SYSTEM, VERSION 04.0 00000070
*** DOCUMENTATION SAS LIBRARY GUIDE, 1997 11 11 00000070
*** PRO1 SAS INSTITUTE INC., 3600, CARY, NC 27513 00000070
***** 00000070
10 XXSAS EXEC PGM=ENTRY,PARM=OPTIONS,SORT=CP550,MSG=CP550 00000070
11 XSLIBRARY DD DSN=SLIBRARY,UNIT=TEMP,SPACE=(CYL,(1,1),101), 00000070
XX DSN=CP550,DISP= 00000070
12 XXSLIBPR DD DSN=CP550,DISP=,CP550,DISP= 00000070
13 XX DD DSN=CP550,DISP=,CP550,DISP= 00000070
14 XX DD DSN=CP550,DISP=,CP550,DISP= 00000070
15 XX DD DSN=CP550,DISP=,CP550,DISP= 00000070
16 XX DD DSN=CP550,DISP=,CP550,DISP= 00000070
17 XX DD DSN=CP550,DISP=,CP550,DISP= 00000070
18 XX DD DSN=CP550,DISP=,CP550,DISP= 00000070
19 XXSASDD DD DSN=CP550,DISP=,CP550,DISP= 00000070
20 XSLIBPR DD DSN=CP550,DISP=,CP550,DISP= 00000070
21 //SASSEK 00000070
22 XCLIBPR DD SYSOUT=CLASS,DD=CP550,DISP=,CP550,DISP= 00000070
XCLIBPR DD SYSOUT=CLASS,DD=CP550,DISP=,CP550,DISP= 00000070
XX CP550 00000070
23 XCLIBPR DD SYSOUT=CLASS,DD=CP550,DISP=,CP550,DISP= 00000070
24 XCLIBPR DD DSN=CP550,DISP=,CP550,DISP= 00000070
25 XCLIBPR DD DSN=CP550,DISP=,CP550,DISP= 00000070
26 XCLIBPR DD DSN=CP550,DISP=,CP550,DISP= 00000070
27 XCLIBPR DD DSN=CP550,DISP=,CP550,DISP= 00000070
28 XCLIBPR DD DSN=CP550,DISP=,CP550,DISP= 00000070
29 XCLIBPR DD DSN=CP550,DISP=,CP550,DISP= 00000070
30 XCLIBPR DD DSN=CP550,DISP=,CP550,DISP= 00000070
31 // 00000070
// 00000070
// DD DSN=CP550,DISP=,CP550,DISP= 00000070
// DSN=CP550,DISP=,CP550,DISP= 00000070
// 00000070

```

```
32  // DD DSN=SYS2.SMPF001, MAY, 015P=500, UNIT=1AFL,  
    // VCL=SCR=PRIVAT, LABEL=(1,PLP),  
    // DCB=SYS2.SMPF001  
33  // DD DSN=SYS2.SMPF001, MAY, 015P=500, UNIT=1AFL,  
    // VCL=SCR=5MAY, LABEL=(1,51),  
    // DCB=SYS2.SMPF001  
34  //SYSIN DD *
```

```
0C0C9554  
0C0C9554  
0C0C9554  
0C0C9554  
0C0C9554  
0C0C9554
```


CAPITULO V.

**PROGRAMA QUE CONTIENE LAS
INSTRUCCIONES NECESARIAS
PARA LA RECOLECCION DE
EVENTOS EN EL SISTEMA.**

EF289	VOL SER NOS = RES001.				
EF289	SYS05177.T124466.RA000.DARL014H.LIBRARY			PASSED	
EF289	VOL SER NOS = CAT001.				
EF289	CATALOG.VISCUSTR			KEPT	
EF289	VOL SER NOS = SC140.				
EF289	SYS251.W022.SASHLF.			KEPT	
EF289	VOL SER NOS = PRSRES.				
EF289	SYS05177.T124466.RA000.DARL014H.HK			DELETED	
EF289	VOL SER NOS = CAT001.				
EF289	JES2.JCR04711.SCO106			SYSOUT	
EF289	JES2.JCR04711.S00105			SYSOUT	
EF289	JES2.JCR04711.SCO106			SYSOUT	
EF289	SYS05177.T124466.RA000.DARL014H.R000001			DELETED	
EF289	VOL SER NOS = CAT001.				
EF289	JES2.JCR04711.SCO106			SYSOUT	
EF289	SHORTMS.SCHTLB			KEPT	
EF289	VOL SER NOS = RES001.				
EF289	SYS05177.T124466.RA000.DARL014H.R000002			DELETED	
EF289	VOL SER NOS = AMSA29.				
EF289	SYS05177.T124466.RA000.DARL014H.R000003			DELETED	
EF289	VOL SER NOS = AMSA29.				
EF289	SYS05177.T124466.RA000.DARL014H.R000004			DELETED	
EF289	VOL SER NOS = CAT001.				
EF289	SYS2.SMF001.MAY			KEPT	
EF289	VOL SER NOS = SMFMAY.				
EF289	SYS2.SMF001.MAY			KEPT	
EF289	VOL SER NOS = SMFMAY.				
EF289	SYS2.SMF001.MAY			KEPT	
EF289	VOL SER NOS = SMFMAY.				
EF289	SYS2.SMF001.MAY			KEPT	
EF289	VOL SER NOS = SMFMAY.				
EF289	JES2.JCR04711.S10102			SYS IN	
EF289	STEP 2SAS / START 05177.1245				
EF289	STEP 2SAS / STOP 05177.1312 CPU 4MIN 02.56SEC SBU			MIN 37.48SEC VIRT 2048K SYS 276K	
EF289	117 ALLOCATED TO SYS0001				
EF289	SYS05177.T112006.RA000.DARL014H.R000001			KEPT	
EF289	VOL SER NOS = CAT001.				
EF289	SYS05177.T124466.RA000.DARL014H.LIBRARY			DELETED	
EF289	VOL SER NOS = CAT001.				
EF289	SYS05177.T124466.RA000.DARL014H.LIBRARY			KEPT	
EF289	VOL SER NOS = CAT001.				
EF289	JOB /DARL014H/ START 05177.1244				
EF289	JOB /DARL014H/ STOP 05177.1312 CPU 4MIN 02.77SEC SBU			MIN 37.49SEC	

3 SAS LOG DS SAS 02-4 VS2/MV3 JOB PABU134P STEP SAS PROC SAS1 12 45

39

00510049

NOTE DATA SET WORK.TYPES HAS 11335 OBSERVATIONS AND 16 VARIABLES. 342 OBS/TRY.
NOTE THE PROCEDURE SORT USED 6.36 SECONDS AND 1136K.

39

PROC CHART DATA=TYPES

00510049

40

NOAR JOBCLASS / 50-1770=RUNTIME TYPE=NOAR

00511049

41

TITLE GRAFICA DE CAPACIDADES

00512049

NOTE THE PROCEDURE CHART USED 2.75 SECONDS AND 320K AND PRINTED PAGE 46.
NOTE SAS USED 1136K MEMORY.

NOTE

SAS INSTITUTE INC.
SAS CIRCLE
PO BOX 8010
CARY, N.C. 27511-0000

CAPITULO V.

G R A F I C A S T I P O I.

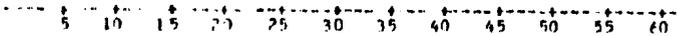
NOS PERMITEN HACER UN ANALISIS DIARIO Y POR HORA ACERCA DEL PROMEDIO DE LLEGADAS DE JOB'S AL SIST. CADA GRAFICA ES UN DIA(HORA/FRECC).

EN LA PARTE DERECHA DE CADA GRAFICA, SE MUESTRAN UNAS ESTADISTICAS LAS CUALES NOS DAN LA FRECC DE LLEGADAS POR HORA, LA FRECC ACUMULADA EN EL DIA, EL PORCENTAJE QUE REPRESENTA CADA HORA EN ESE DIA, ASI COMO EL PORCENTAJE ACUMULADO.

EN LA PARTE INFERIOR, SE MUESTRA EL SIMBOLO QUE ESTA PRESENTADO EN LA GRAFICA Y LA CLASE A LA CUAL REPRESENTA

FREQUENCY BAR CHART

HORA		FREQ	CUM. FREQ	PERCENT	CUM. PERCENT
8	13	13	4.26	4.26
10	H	1	14	0.22	4.59
11	GGTPPPVVVVV	12	26	3.02	8.52
12CGGGGGGGGHHHHHHPPPPPPVVVVVVV	41	67	13.44	21.97
13	.GGGGPPPPVVVVV	18	85	5.50	27.47
14	GGGGGGGGGGPPPPSVV	20	105	6.56	34.03
15CGGGGGGGHHHHPPPPPPVVVVVVVVVVVV	39	144	12.79	47.21
16GGGGGGGGGHHJJPPPPPPPPVV	29	173	9.51	56.72
17GGGGGGGGGTPPPPPPPPPPSL	29	202	9.51	66.23
18	.CCCCCCCCCCCCCCCCCCCCCCCCDDGGGGGLPPPPPPVVVVVV	61	263	20.00	86.23
19	CGGLPPVVVVVVVV	15	278	4.52	91.15
20	.CCCCGGGLLLLLLLLL	18	296	5.50	97.05
21	.CCCCGG	7	303	2.20	99.24
22	.C	2	305	0.66	100.00

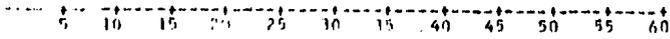


JORCLASS	SYMBOL								
J	C	C	P	P	G	G	V	V	I

FREQUENCY BAR CHART

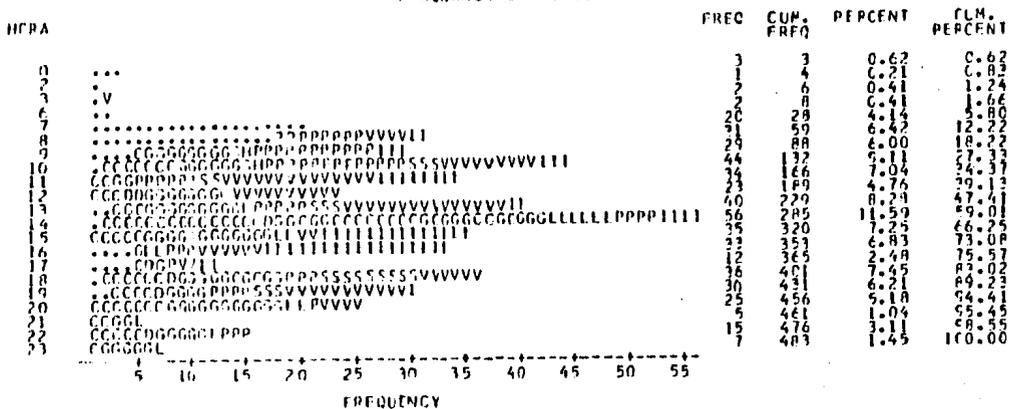
HORA		FREQ	CUM. FREQ	PERCENT	CUM. PERCENT
0VVVVVVVVVVVVVVVI	24	24	4.44	4.44
7	...V	4	28	0.74	5.19
3	.	1	29	0.19	5.37
7	..GGVV	6	35	1.11	6.48
8	..GGGGGGHPPPPPPPPPPPPPI	27	62	5.00	11.48
9	.CCCCGGGGHSSSSVVVVVVVVVIIIIIIII	31	93	5.74	17.22
10	VIIIIII	6	99	1.11	18.33
11	8	107	1.48	19.81
12CCGGGPPPVVVVIIIIIIIIIIIIIIII	37	144	6.65	26.47
13CCCCCCCCCCCCGGGGHPPVVVVVVVVVIIIIIIIIIIIIIIIIIIIIII	60	204	11.71	37.70
14	..CCCCCCCCCCCCGGGGGGHSSSSVVVVVVVVVIIIIIIIIIIIIIIIIII	51	255	9.44	47.22
15	..CCCCGGGGIPSSSSSSSSVVVVVVVVVIIIIIIII	39	294	7.72	54.44
16CCCCGGGGGGGGHIIIIIMPPPPPPPPSSVVVVVVVII	53	347	9.81	64.26
17CCCCGGGGGGGGIIPPPPPPPPPPPVVVVVVVVVVI	46	393	8.92	72.78
18	.CCCCCCCCCCCCCCCCCCCCGGGGHIIIIIIIIIIIIIIIIIIIIII	54	447	10.00	82.78
19	CCCCCCCCCCCCCCCCCCCCGGGGHIIIIIIIIIIIIIIIIIIIIII	48	495	8.95	91.67
20	CCCCCCCCGGG	12	507	2.22	93.89
22	..SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS	12	519	5.43	99.81
23	.	1	540	0.19	100.00

-75-



L. JORCLASS	SYMBOL JORCLASS									
	C	H	D	P	G	S	H	V	I	I

FREQUENCY BAR CHART



SYMBOL	JOBCLASS								
I	I	C	C	D	D	G	G	H	H

CAPITULO V.

GRAFICAS TIPO II.

ESTAS GRAFICAS, NOS PERMITEN OBTENER EL PROMEDIO DE LLEGADAS AL SISTEMA DE CADA CLASE POR HORA, EN UNA MUESTRA DE VARIOS DIAS.

SE RECOMIENDA QUE ESTAS MUESTRAS SE TOMEN PARA LAS 24hrs, CON INTERVALOS SEMANALES, YA QUE ESTO NOS PERMITE ANALIZAR LAS VARIANCIAS ENTRE SEMANAS.

EN LA PARTE SUPERIOR SE MUESTRA LA HORA EN QUE OCURREN LOS EVENTOS.

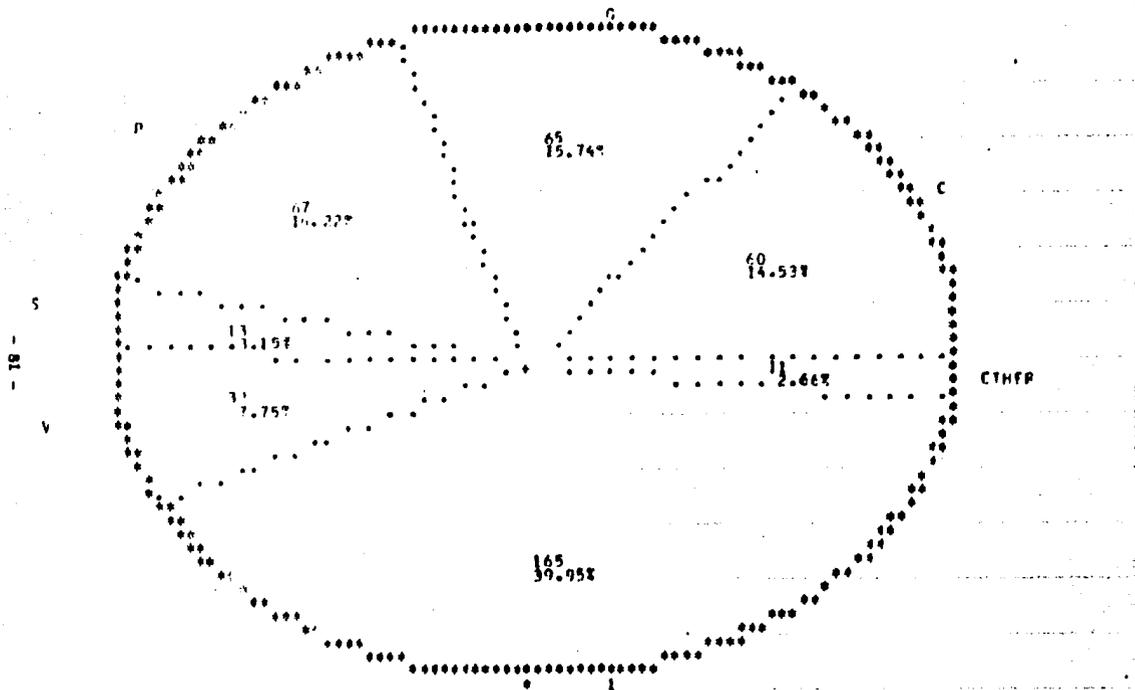
EN EL EXTERIOR DE LA GRAFICA SE ESPECIFICA EL TIPO DE CLASE.

EN EL INTERIOR DE LA GRAFICA SE PUEDE OBSERVAR TANTO LA FRECC DE LLEGADAS DE DICHA CLASE EN ESA HORA, COMO TAMBIEN EL PORCENTAJE QUE ESTO REPRESENTA EN DICHA HORA.

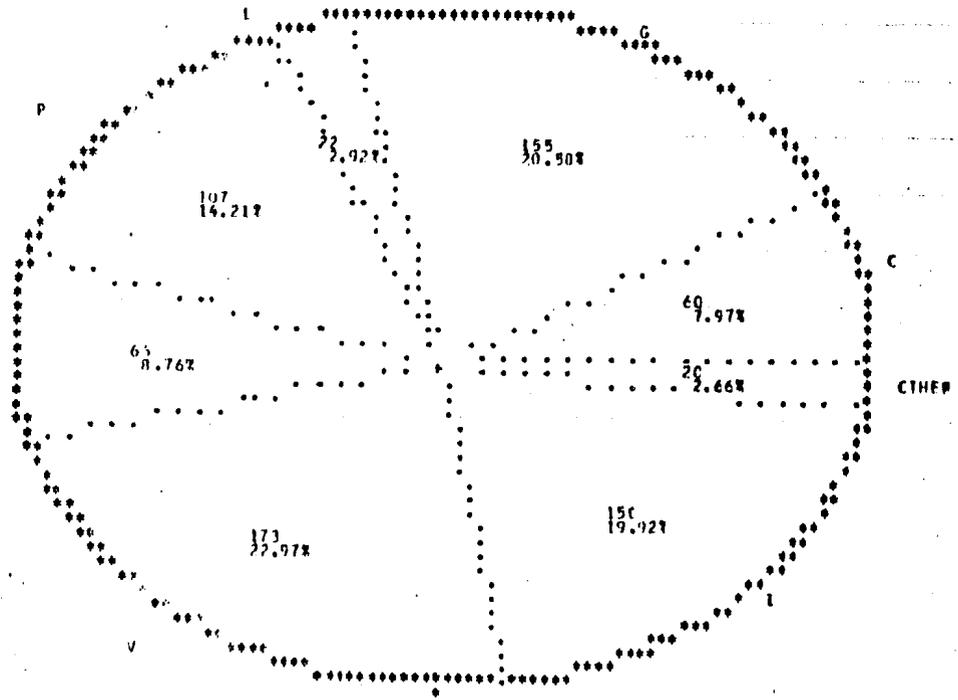
--- COLLECTION OF ENTRADAS ---
MORANO

14 46 WEDNESDAY, JULY 3,

FREQ PIE CHART OF JORCLASS



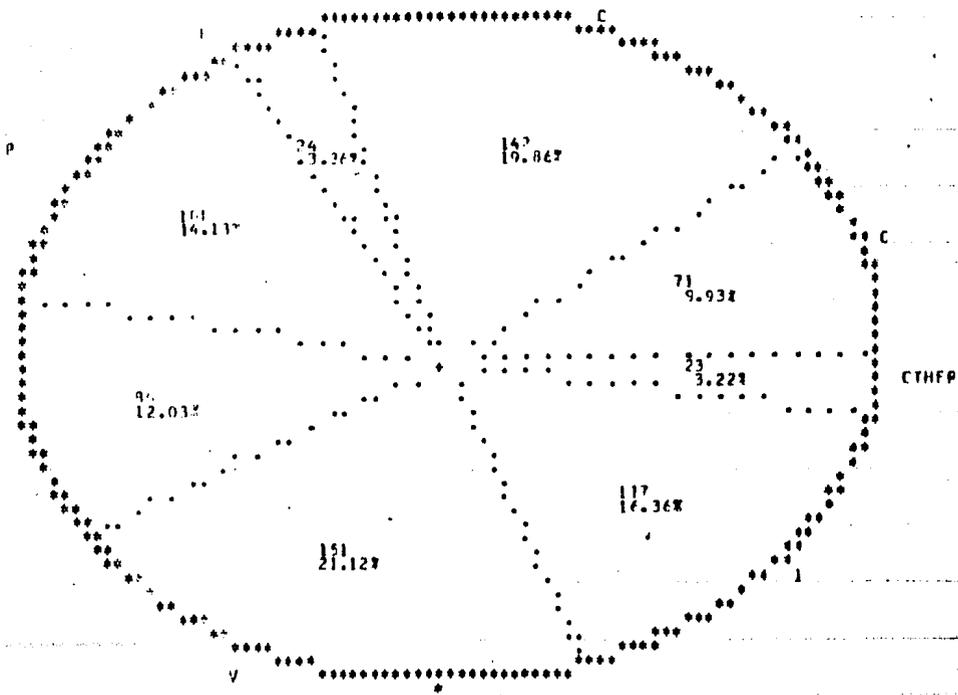
FREC PIE CHART OF JOBCLASS



---COLLECTION OF ENTRADAS---
HORACII

14 46 WEDNESDAY, JULY 3,

ERIC PIE CHART OF JONCLASS

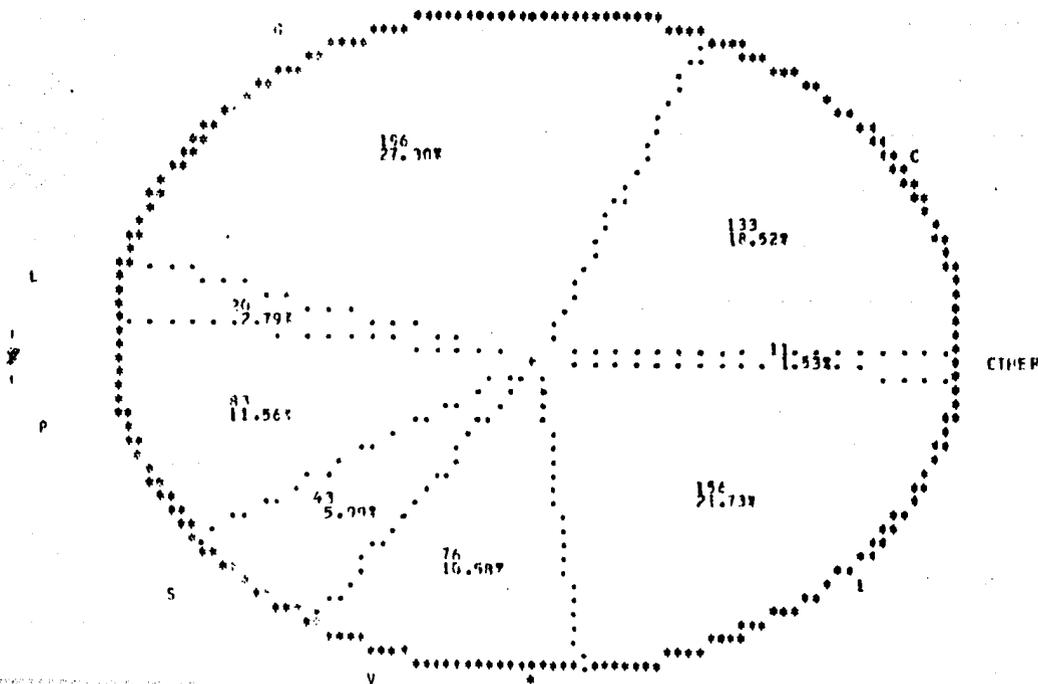


CTHFP

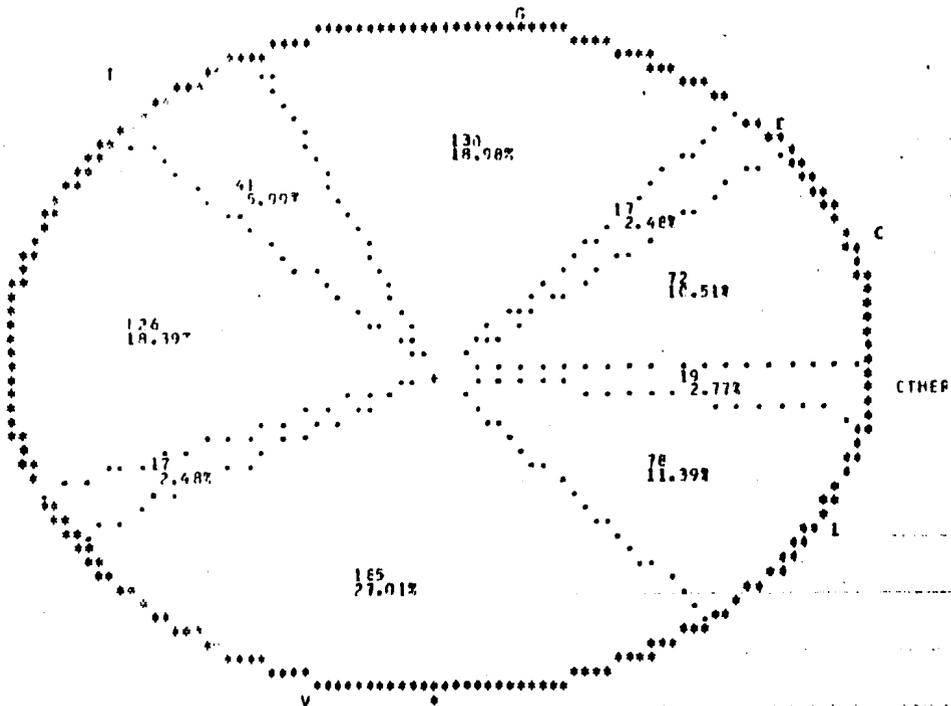
--- CORRECCION DE ENTRADAS ---
MONA 12

14 46 WEDNESDAY, JULY 3,

FREQ PIE CHART OF JONCLASS



FREC PIE CHART OF JOBCLASS



P
- 85 -

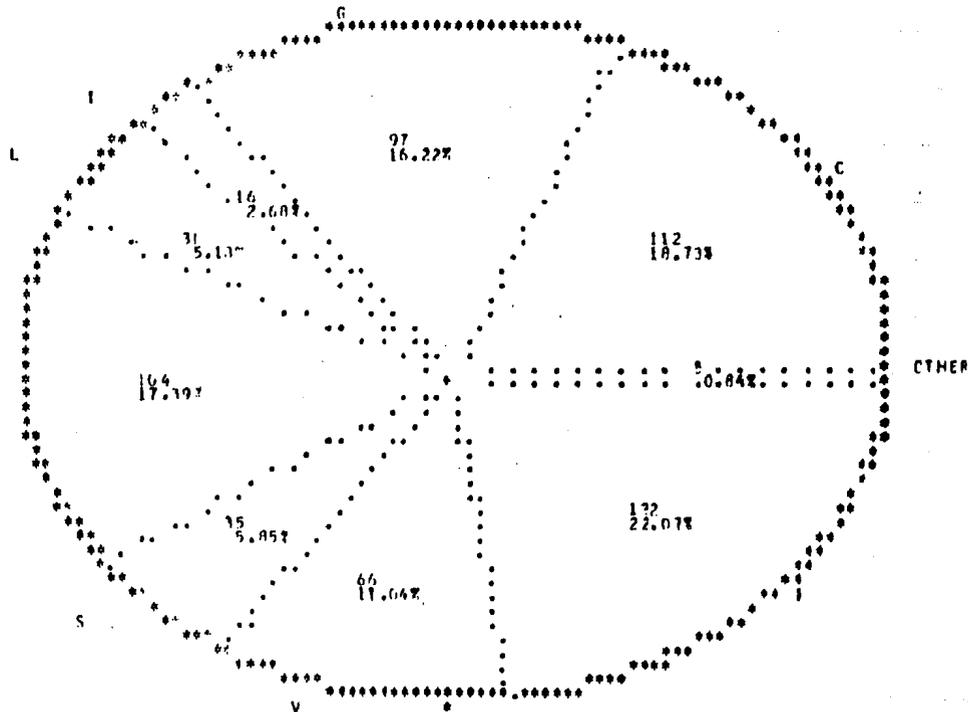
S

OTHER

---COLECCION DE ENTRADAS---
HORAS 14

14 46 WEDNESDAY, JULY 3,

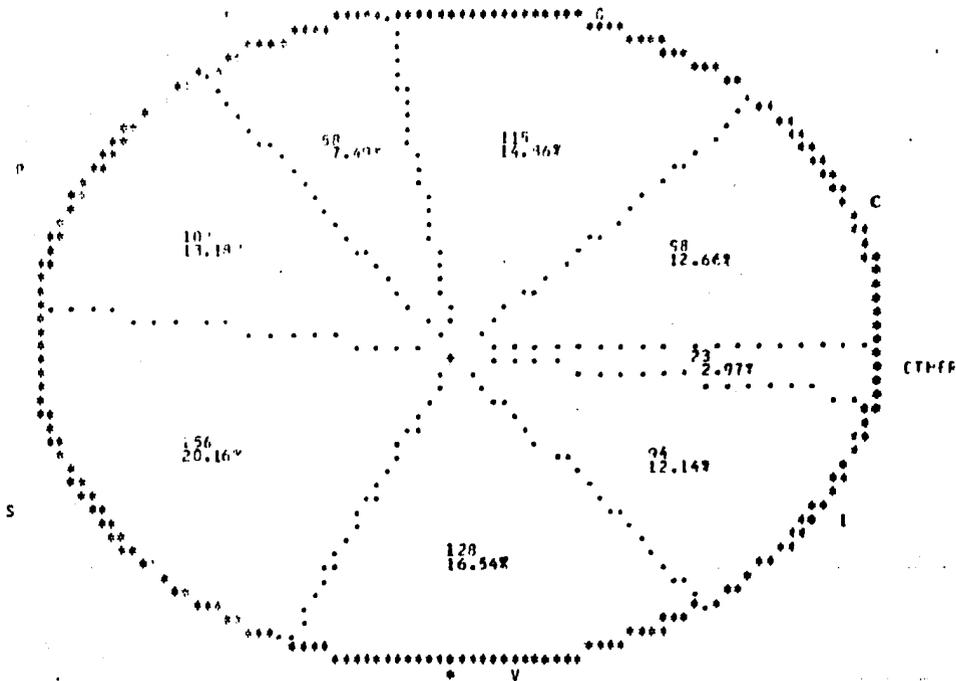
FREC PIE CHART OF JORCLASS



---ORIGIN OF ENTRADAS---
URBANA

14 46 WEDNESDAY, JULY 3,

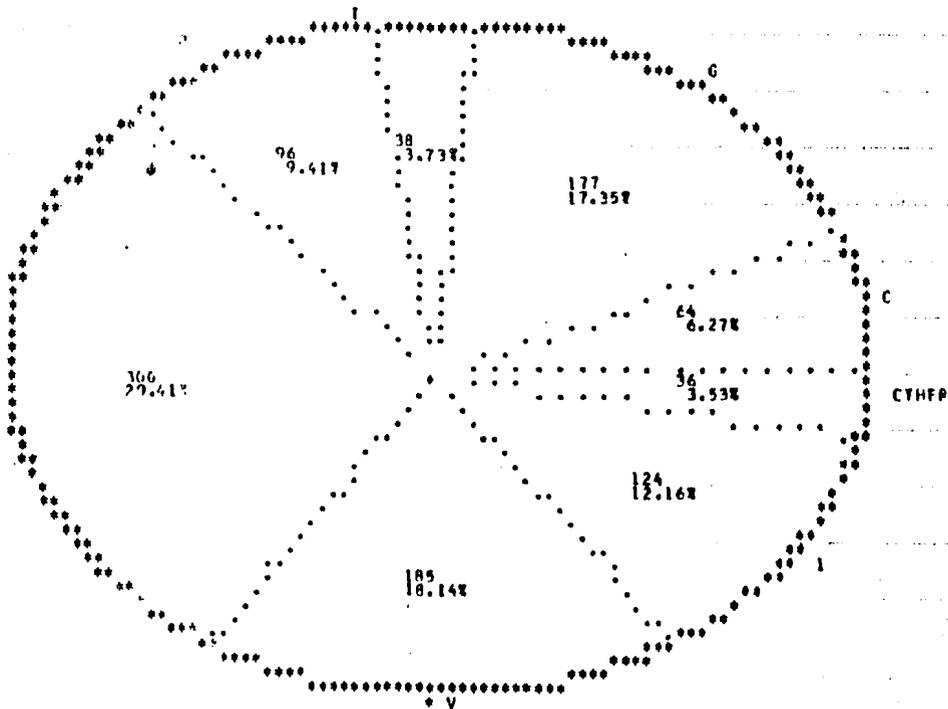
FREQ PIC CHART OF JODCLASS



--- COLECCION DE ENTRADAS ---
HORAF16

14 46 WEDNESDAY, JULY 3,

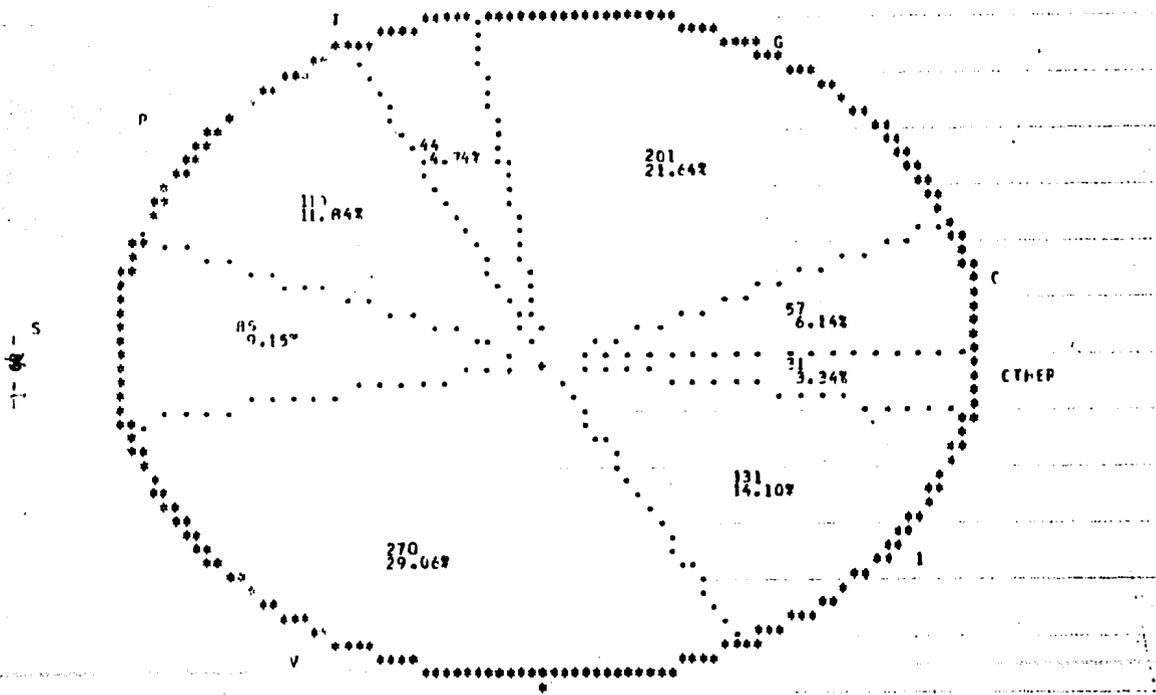
FREC PIE CHART OF JORCLASS



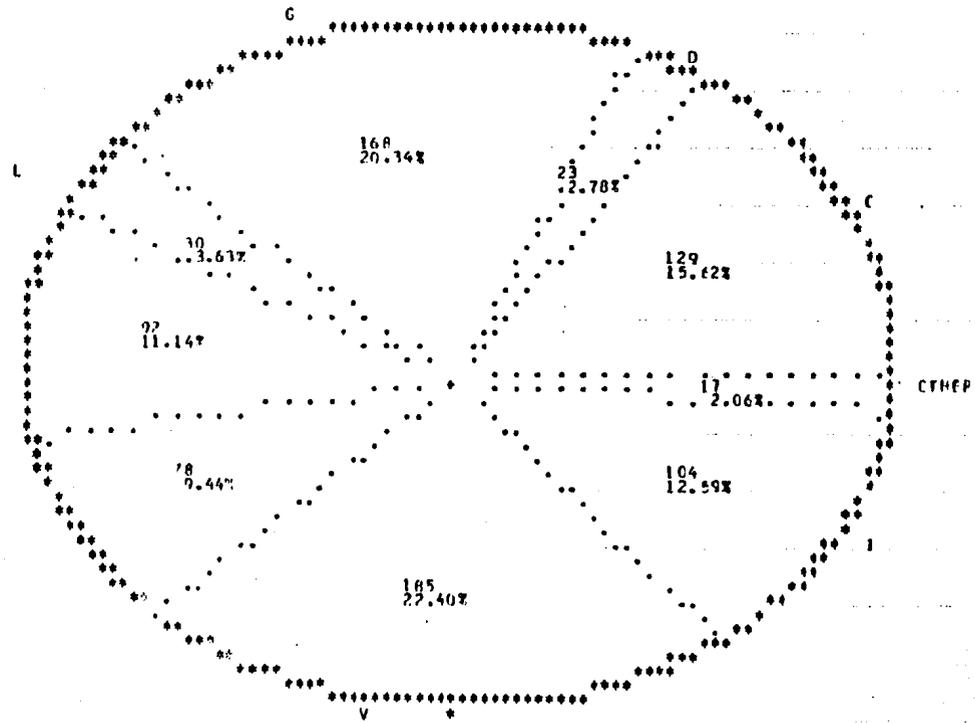
--- COLECCION DE ENTRADAS ---
HCRAC17

14 46 WEDNESDAY, JULY 3,

FREC PIE CHART OF JOBCLASS



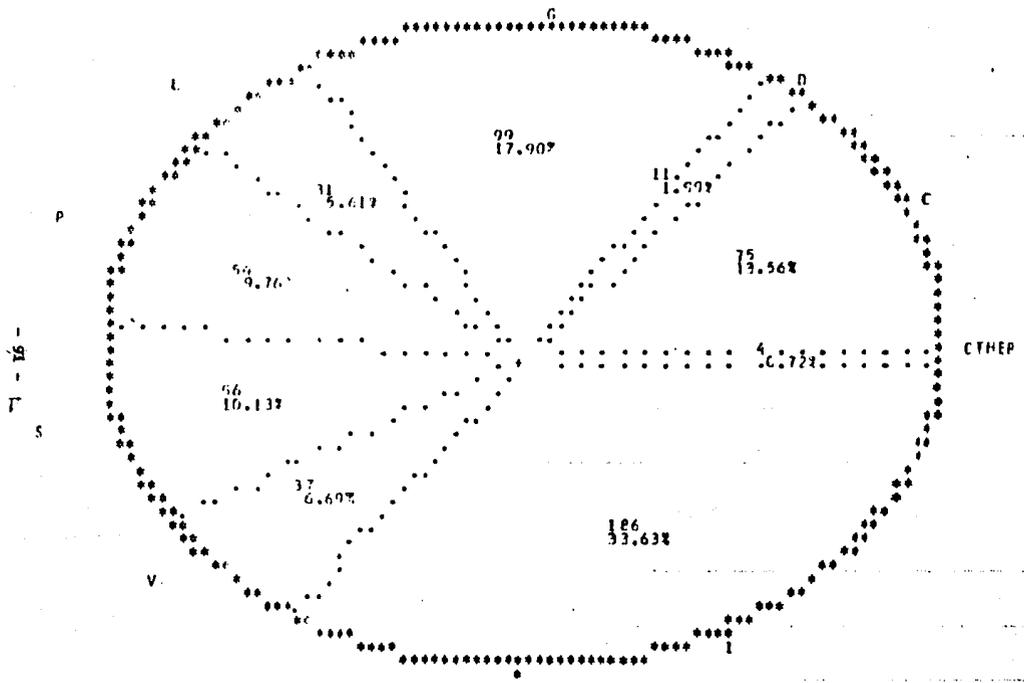
FREC PIE CHART OF JOBCLASS



--- CORRECTION OF ENTRADAS ---
MORAP 10

14 46 WEDNESDAY, JULY 3,

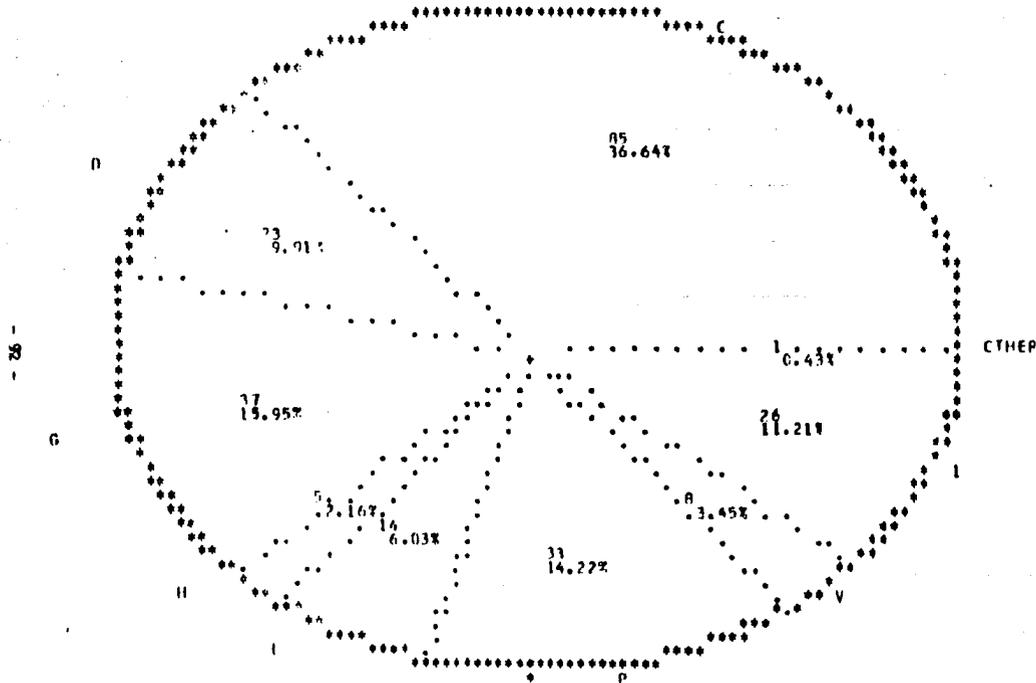
FREQ P/F CHART OF JODCLASS



---(SECCION DE ENTRADAS)---
HORAS 20

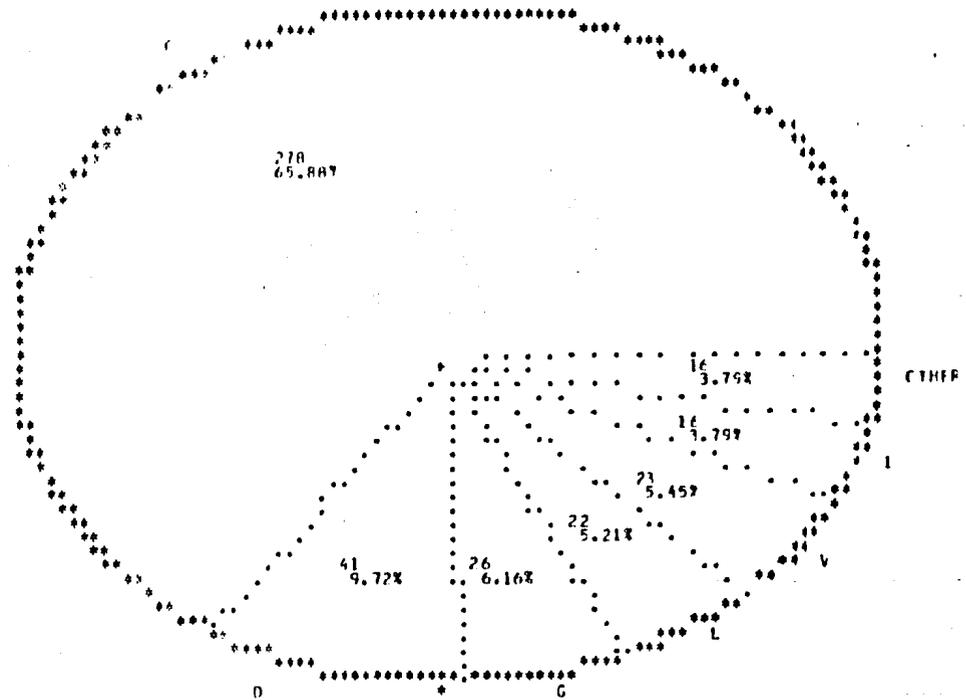
14 46 WEDNESDAY, JULY 3,

FREC PIE CHART OF JONCLASS



FREQ PIP CHART OF JOBCLASS

96



D

*

G

L

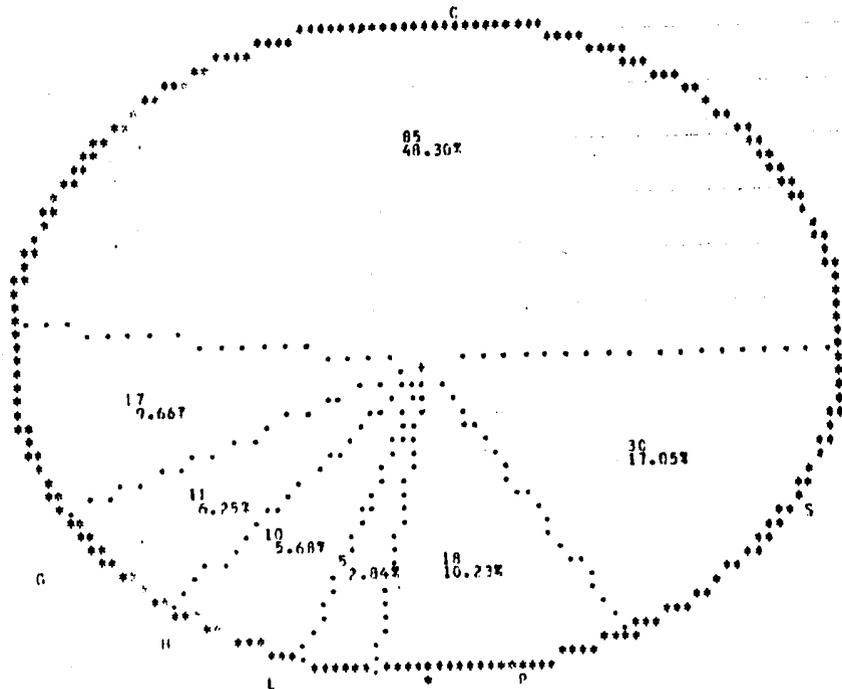
V

I

---COLLECCION DE ENTRADAS---
HORAS 22

14 46 WEDNESDAY, JULY 3,

FREC PIE CHART OF JOBCLASS



CAPITULO V.

GRAFICAS TIPO III

ESTAS GRAFICAS, NOS PERMITEN OBTENER EL PROMEDIO DE DURACION DE CADA JOB, ESTO ES, EL TIEMPO PROMEDIO QUE CADA TIPO DE CLASE TUVO ASIGNADA UNA REGION:

ESTAS GRAFICAS TAMBIEN SE TOMAN DE UNA MUESTRA DE VARIOS DIAS. SE RECOMIENDA TOMAR LAS MUESTRAS POR SEMANA.

EN LA PARTE DERECHA DE CADA GRAFICA, SE OBSERVA LA FRECC DE LLEGADAS POR CADA CLASE EN ESA MUESTRA (FREQ), Y EL TIEMPO PROMEDIO DE DURACION (RUNTIME MEAN) EN HH:MM:SS.

GRAFICA DE DURACIONES
BAR CHART OF MEANS

14 18 WEDNESDAY, JUL

JOBCLASS	FREQ	RUNTIME MEAN
C *****	159	0 07 41
D *****	26	0 17 32
G **	285	0 02 17
H *****	21	1 35 43
I *	4	0 00 51
J **	1	0 02 15
L *****	37	0 21 47
P *****	191	0 06 23
S *****	29	0 09 32
V **	239	0 02 04
W *****	124	0 26 02

0 10.00 0.20.00 0.30.00 0.40.00 0.50.00 1.00.00 1.10.00 1.20.00 1.30.00

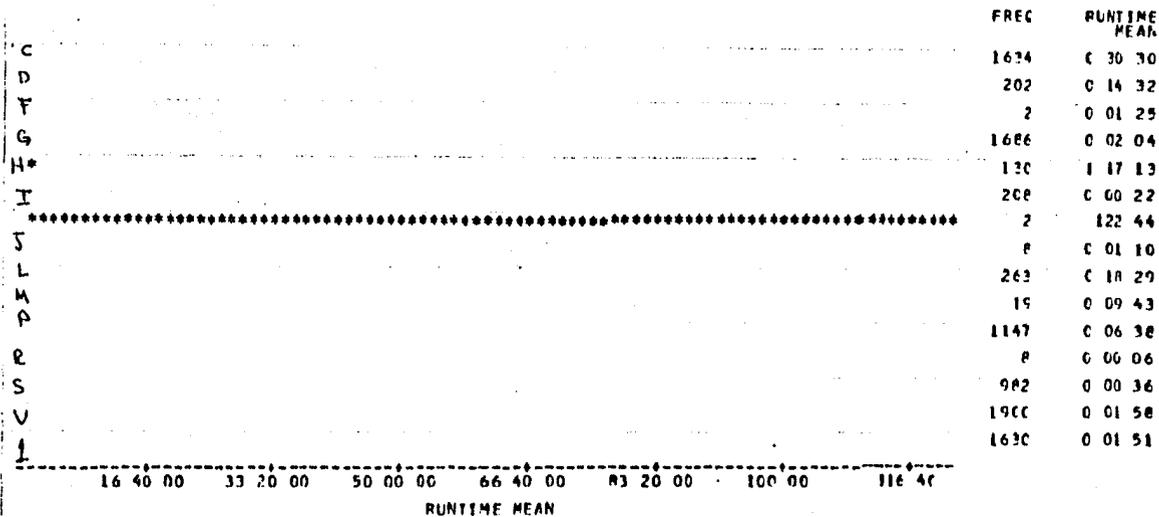
RUNTIME MEAN

1
8
1

GRAFICA DE CURACIONES

14 46 WEDNESDAY, JULY 3, 19...

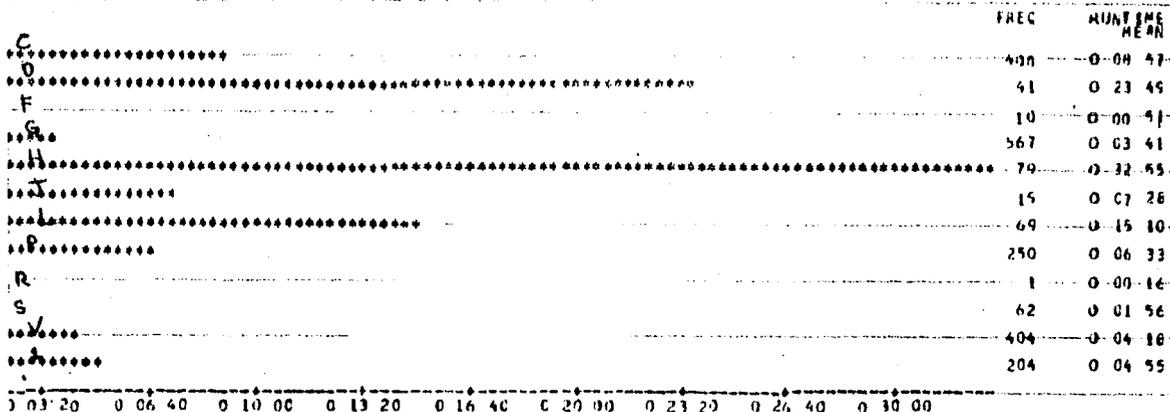
BAR CHART OF MEANS



GRAFICA DE DURACIONES

14 17 WEDNESDAY, JUNE 26, 1985

BAR CHART OF PLAYS



RUNTIME MEAN

1
8
1

CAPITULO V.

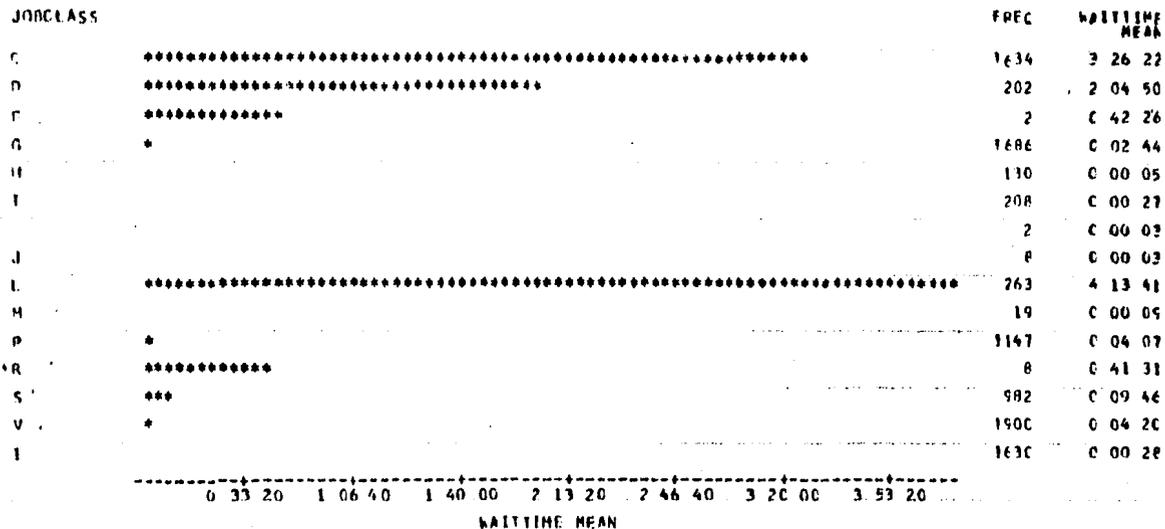
GRAFICAS DEL TIPO IV

ESTAS GRAFICAS, NOS DAN EL PROMEDIO DE TIEMPO DE ESPERA POR CLASE. ESTE ES EL PROMEDIO DE TIEMPO QUE TUVO QUE ESPERAR CADA CLASE PARA QUE SE LE ASIGNARA UN INICIADOR (UNA REGION).

ESTAS GRAFICAS, AL IGUAL QUE LAS ANTERIORES, SE OBTIENEN DE UNA MUESTRA DE VARIOS DIAS (SEMANA). LA VARIABLE 'WAITTIME MEAN' NOS DA EL TIEMPO DE ESPERA POR CLASE EN H:MM:SS.

GRAFICA DE TIEMPO DE ESPERA
BAR CHART OF MEANS

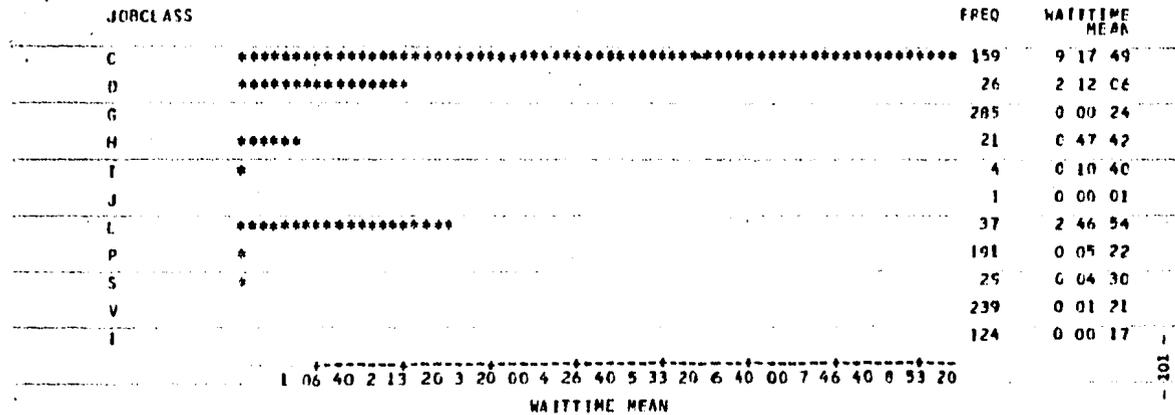
14 46 WEDNESDAY, JULY



GRAFICA DE TIEMPO DE ESPERA

14 18 WEDNESDAY, JUL

BAR CHART OF MEANS



CAPITULO V.

Haciendo un análisis preeliminar de las Gráficas Tipo 1 y 2, se observa que el período comprendido entre las 19 hrs. y las 9 hrs. tiene una carga de trabajo mínima, por lo que se considera que cualquier asignación de clases que se haga en el sistema no afectará a este sistema. Por tal, este período no será considerado en el modelo de simulación.

El siguiente paso, es obtener el promedio diario de llegadas de Job's al sistema por hora.

Esto se obtiene a partir de las gráficas del tipo 2, dividiendo el número de clases entre el número de días que se consideraron en cada muestra.

El resultado obtenido se muestra en la Tabla 5.3.1

En dicha Tabla 5.3.1 en el margen izquierdo se muestra el tipo de clase y el número que lo representará en el momento de la simulación.

En la parte superior de la Tabla tenemos la hora en que llegaron los Job's. La Tabla contiene el promedio diario de llegadas por hora, de cada clase.

En el último renglón, se tiene el promedio diario de llegadas por hora.

Esta Tabla se puede resumir en la Gráfica de la

CAPITULO V.

fig. 5.3 en donde se puede apreciar que tenemos dos periodos críticos bien definidos:

PERIODO 1.....de las 10 a las 14 hrs.

PERIODO 2.....de las 15 a las 18 hrs.

Como se puede apreciar, el Período 2 es más crítico ya que presenta mayor número de llegadas al sistema.

Estos dos periodos se resumen en las Tablas 5.3.2 y 5.3.3 respectivamente. En estas tablas se muestra la siguiente información:

- Clase, el tipo de clase y el número que representa a dicha clase en la simulación.
- Hora, la hora del día.
- Prom, promedio de clases diario por hora.
- Prom1, promedio de clases diario por hora por clase real.
- Prom2, promedio de clases diario por hora por clase que se tomarán en cuenta para la simulación (no real).

En este caso se les asigna un tiempo de llegadas muy pequeño a las clases menos frecuentes (con menos llegadas) con el fin de que se tomen en cuenta en el momento de la simulación.

FIGURA 5.3

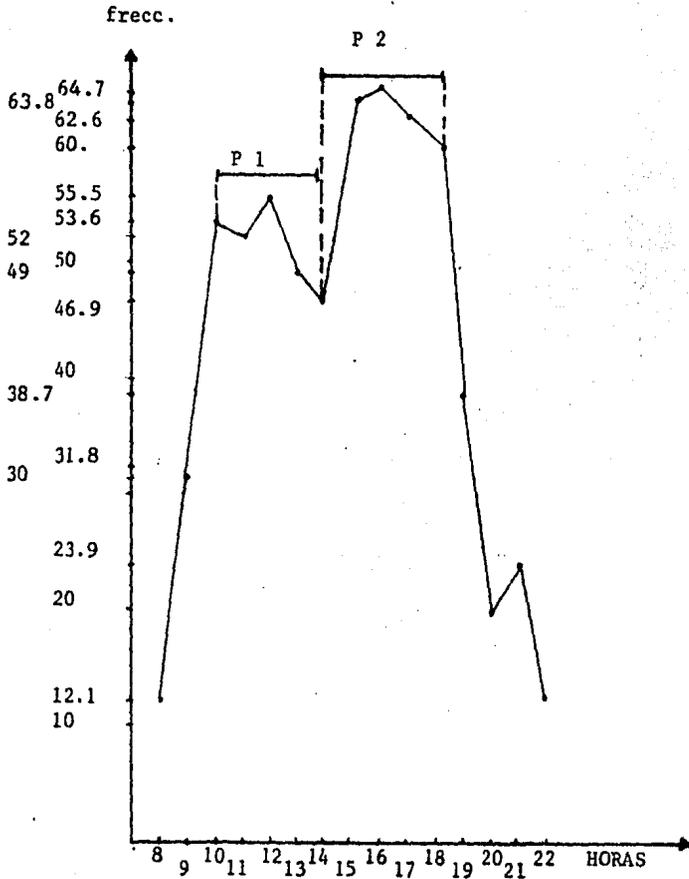


TABLA 5. 3. 1.

CLASE/HORA	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
C 1	.55	4.05	5.3	4.9	9	5.5	7.6	6.9	5.2	5.6	13.8	6.8	7.7	16.6	6.5
D 2			.22			1.1	.27				1.4	1.1	1.5	2.5	1.3
F 3															
G 4	2.2	6.3	12.5	12.2	16	10.6	8.7	11.4	11.9	15.3	12.6	7.3	3.9	2.3	1.2
H 5	2.3	.55	.5	.16	.44	.16	.55	.83	.77	.9	.22	.16	.27		.55
I 6						2.2	.8	3.2	2.1	2.4					
J 7						.2								.1	.1
L 8			1.2	1.7	1.5	.3	.2		.1	.5	2.3	1.8	1.8	2.2	.6
M 9															
N 10															
P 11	2.8	6.3	8.6	8.5	6.7	9.3	7.6	7.7	7.1	8.2	5.7	3.8	2.2	.22	1.2
R 12															
S 13		1.5	3.8	5	2.7	1.6	2.5	8.6	16.8	4.8	4.7	3.6	.1		1.6
V 14	.5	3.1	11.7	11	8.8	12.2	7.1	11.3	12.7	16.9	12.3	3.1	1	1.2	
I 15	3.8	10													
PROM.	12.1	31.8	53.6	51.7	55.5	49.6	46.9	63.8	64.71	62.6	60.5	38.7	20.1	23.9	13.2

* Tabla, resumen del promedio diario de llegadas de Job's al sistema por hora.

PERIODO 1

Número de clase por hora, por día.

Se toman las muestras de las horas 10, 11, 12, 13, 14.

CLASE/HORA	10	11	12	13	14	PROM. 1	PROM. 2	%	F.D.A.
C 1	5.3	4.9	9	5.5	7.6	5.3	5.3	.103	.103
D 2				1.16	.27	.28	.3	.005	.108
F 3							.2	.003	.111
G 4	12.5	12.1	16	10.6	8.7	12	12	.234	.345
H 5	.5	.16	.44	.16	.55	.362	.36	.007	.352
I 6				2.2	.8	.6	.6	.011	.363
J 7				.2		.04	.2	.003	.366
L 8	1.2	1.7	1.5	.3	2	1.34	1.34	.026	.392
M 9							.2	.003	.395
N 10							.2	.003	.398
P 11	8.6	8.5	6.7	9.3	7.6	8.14	8.14	.158	.556
R 12							.2	.003	.559
S 13	3.8	5	2.7	1.6	2.5	3.12	3.12	.060	.619
V 14	11.7	11	8.8	12.2	7.1	10.16	10.16	.198	.817
1 15	10	8.1	10.4	6.4	9.8	8.94	8.9	.173	1.000
PROM.	53.6	51.7	55.5	49.6	46.9				

TABLA 5.3.2.

T A B L A 5.3.2 (Continuación)

Tiempo entre llegadas

#JOBS - 3600

l - x

JOBS Tiempo entre llegadas

Tiempo entre llegadas

53.6 - 67

51.7 - 70

46.9 - 77

55.5 - 65

49.6 - 72.5

70 ± 6

PERIODO 2

Número de clase por hora, por día.

Se toman las muestras de horas pico 15, 16, 17, 18.

CLASE	15	16	17	18	PROM. 1	PROM. 2	F.D.A.
C 1	6.9	5.2	5.6	13.8	7.85	7.85	.1222
D 2				1.4	.35	1.4	.144
F 3							.1471
G 4	11.4	11.9	15.3	12.6	12.8	12.8	.3464
H 5	.83	.77	.9	.22	.68	.68	.3569
I 6	3.2	2.1	2.4		1.92	2.5	.3958
J 7							.3989
L 8		.1	.5	2.3	.72	.96	.4138
M 9							.4169
N 10							.42
P 11		7.7	7.1	8.2	5.9	7.66	.5383
R 12							.5414
S 13	8.6	16.8	4.8	4.7	8.72	8.72	.6772
V 14	11.3	12.7	16.9	12.3	13.3	13.3	.8843
1 15	6.2	8.1	8	7.5	7.4	7.4	1.000
PROM.	63.83	64.71	62.6	60.5			

TABLA 5.3.3.

T A B L A 5. 3. 3. (Cont.)

TIEMPO ENTRE LLEGADAS

JOBS --- 3600

1 --- X

JOBS

63.83 -- 57

64.71 -- 56

62.6 -- 58

60.5 -- 60

Tiempo entre llegadas:

[55 ± 5]

CAPITULO V.

También de estas tablas se puede obtener una aproximación, al promedio de tiempo entre llegadas.

Este valor va a ser utilizado en el modelo para simular el tiempo entre llegadas al sistema.

La forma en que se calcula el tiempo entre llegadas es la siguiente:

$$\text{TIEMPO ENTRE LLEG.} = (1 \text{ JOB} * 3600 \text{ seg}) / \# \text{ Job's}$$

Para finalizar, se obtienen las Tablas 5.3.4 y 5.3.5, las cuales contienen los tiempos de duración y tiempos de espera.

Estas Tablas nos dan la siguiente información:

-La clase y el número que se le asignó para llevar a cabo la simulación.

-Límite superior e inferior que son los valores máximos y mínimos de tiempo de ejecución de cada una de las clases. Viene dado en H:MM:SS.

Se observa en estas tablas que algunos valores no se ajustan a lo programado. Por ejemplo en la clase 'C' (Producción Corto). Se supone que debería contener aquellos Job's cuyo tiempo de ejecución no excedan de 5 minutos, y sin embargo están comprendidas en un rango

CAPITULO V.

de 6 a 28 min. en las muestras obtenidas. Esto se debe a que los programadores o usuarios, no respetan la condición de límite de tiempo, y al procesar sus trabajos, estos le asignan una clave indebida. Por lo tanto, para efectos de la simulación, consideraremos los tiempos obtenidos en la presente recolección de eventos.

Del análisis efectuado anteriormente, se observa que las variables del sistema, que son de interés para el presente estudio son:

1) Tiempo entre llegadas. Esta variable se utiliza en el modelo de simulación, y permite simular el flujo de llegadas de trabajos al sistema.

2) Períodos críticos. Estos períodos de tiempo, se clasifican según la carga de trabajo que recibe el sistema (# Job's, clases, duración, etc.), para después hacer simulaciones en dichos períodos y ver como afecta a cada uno las diferentes asignaciones de clases.

3) Tipos de clases. Estas variables, nos van a permitir conocer la frecuencia de llegadas de cada una de las clases, para posteriormente poder hacer una asignación adecuada en el modelo de simulación.

4) Duración. La duración, al igual que el tipo de

TIEMPOS DE DURACION

DURACION (hrs., min., seg.,)

C L A S E	LIM. INF.	LIM. SUP.
C 1	6:40	28:0
D 2	11:40	21:0
F 3	:45	1:25
G 4	2:0	3:20
H 5	31:40	1:27:37
I 6	:20	:45
J 7	1:10	2:05
L 8	15:0	20:27
M 9	6:40	10:0
N 10	1:20	2:0
P 11	5:0	6:40
R 12	:05	:17
S 13	:30	1:50
V 14	1:20	3:20
l 15	1:40	5:40

T A B L A 5. 3. 4.

TIEMPO DE ESPERA POR CLASE CON LA DISTRIBUCION
DE CLASES ACTUAL, OBTENIDA DE LAS GRAFICAS
TIPO 4

T A B L A 5. 3. 5.

C L A S E	T. ESPERA (SEG.)
1 C	21,600
2 D	2,490
3 F	2,546
4 G	164
5 H	2,844
6 I	640
7 J	3
8 L	12,600
9 M	9
10 N	-
11 P	322
12 R	2,491
13 S	586
14 U	260
15 I	23

CAPITULO V.

clase, nos va a permitir conocer el promedio de tiempo de ejecución que tiene cada trabajo, para poder asignarlo al modelo de simulación.

5) Tiempo de espera. Esta variable nos indica el tiempo promedio que tiene que esperar un trabajo de determinada clase, antes de poder ser procesado.

Podemos apreciar que todas estas variables son muy importantes para poder llevar a cabo la simulación.

Los resultados que deseamos obtener son:

- Obtener un tiempo de espera bajo (el tiempo que transcurre desde que se submitió el job hasta que se empezó a procesar).

- Tratar de tener ocupados la mayor parte del tiempo los recursos del sistema (CPU, Memoria, etc).

C A P I T U L O VI :

DESCRIPCION DEL MODELO DE SIMULACION

CAPITULO VI.

SIMULACION .

6.1 INTRODUCCION.

Con el advenimiento de la computadora, una de las más importantes herramientas para analizar el diseño y operación de sistemas o procesos complejos, es la SIMULACION.

Una de las razones principales del enfoque de la simulación, es que sustrae la esencia del problema y revela la estructura fundamental proporcionando en consecuencia el conocimiento de las relaciones causa-efecto dentro del sistema.

La simulación, típicamente comprende la construcción de un modelo que, en naturaleza, es matemático en gran parte.

El modelo de simulación describe la operación del sistema en términos de eventos individuales de los componentes del sistema por separado. En particular el sistema se divide en elementos cuyo comportamiento se

CAPITULO VI.

puede predecir, al menos en términos de distribuciones de probabilidad, para cada uno de los diversos estados posibles del sistema y de sus entradas. También se introducen al modelo las interrelaciones entre los elementos.

Por tanto, la simulación suministra un medio para dividir la tarea de construcción del modelo en partes componentes más pequeñas, y a continuación combinar estas partes en su orden natural y dejar que la computadora presente el efecto de sus interacciones mutuas.

Después de construir el modelo, se activa entonces para simular la operación real del sistema en el tiempo y registrar su comportamiento conjunto.

Repitiendo este procedimiento para las diversas configuraciones alternativas correspondientes al diseño y a las políticas de operación del sistema, y comparando sus rendimientos se pueden identificar las configuraciones más prometedoras.

Debido al error estadístico es imposible garantizar que la configuración que presenta el mejor rendimiento simulado, en efecto es la óptima, pero al menos, debe estar muy cercana a la óptima.

Se puede definir a la simulación como :

CAPITULO VI.

Simulación, es el proceso de diseñar y desarrollar un modelo computarizado del sistema o proceso, y conducir experimentos con este modelo con el propósito de entender el comportamiento del sistema.

Para los modelos de simulación es necesario describir y sincronizar las llegadas o entradas de clientes al sistema.

Existen dos métodos para manejar tal sincronización:

El incremento de tiempo fijo, y el incremento según el intervalo siguiente.

Con el incremento de tiempo fijo, se utiliza el procedimiento siguiente en dos pasos, empezando con el sistema de su estado inicial, en un punto dado del tiempo.

Primero se avanza el tiempo en una pequeña cantidad fija, se agrega i a un registrador que sirve como 'Reloj Maestro' para el sistema, a fin de registrar el paso de este tiempo.

Segundo, se actualiza el sistema, determinando cuales eventos ocurrieron durante esta unidad de tiempo transcurrida y cual es el estado resultante del sistema.

Se repiten estos dos pasos tantas unidades de tiempo como se desee.

CAPITULO VI.

El incremento según el evento siguiente, difiere del incremento de tiempo fijo, en que el 'Reloj Maestro' se incrementa en una cantidad variable, en lugar de una cantidad fija cada ocasión.

Desde el punto de vista conceptual, el incremento según el evento siguiente, es hacer que el sistema simulado siga desarrollándose sin interrupción hasta que ocurra un evento, en cuyo punto la computadora hace una pausa momentáneamente para registrar el cambio en el sistema.

Para poner en práctica esta idea conceptual, en realidad la computadora procede observando cuando está programada la ocurrencia de unos cuantos eventos simulados siguientes, saltando en el tiempo simulado al primero de estos eventos y actualizando el sistema.

Este ciclo se repite tantas veces como se desee.

CAPITULO VI.

6.2 Factores a considerar en el desarrollo del modelo de simulación.

Lenguaje de programación.

Las primeras etapas de un estudio de simulación se refieren a la definición del sistema en términos de relaciones lógicas de sus variables y diagramas de flujo.

Sin embargo llega el momento de describir el modelo en un lenguaje que sea aceptado por la computadora que se va a usar.

En este caso utilizaremos un paquete de simulación llamado GPSS (General Purpose Simulating System) el cual fue facilitado por la Gerencia de Orientación e Informática de la compañía ICA.

Condiciones iniciales.

La mayoría de los problemas de simulación se corren con la idea de estudiar al sistema en una situación de estado estable. Sin embargo la mayoría de los modelos presentan en su etapa inicial estados transientes los cuales no son típicos del estado estable. Por lo tanto es necesario establecer claramente las alternativas o cursos de acción que existen para resolver este problema. Algunos

CAPITULO VI.

autores piensan que la forma de atacar este problema sería a través de :

- Usar un tiempo de corrida lo suficientemente grande de modo que los estados transientes sean relativamente insignificantes con respecto a la condición de estado estable.

- Excluir una parte apropiada de la parte inicial de la corrida.

En nuestro caso utilizaremos el de excluir una parte apropiada de la parte inicial de la corrida.

6.3 Descripción del modelo de simulación.

El modelo que se llevará a cabo, debe considerar todos los aspectos que se contemplan en el sistema.

Como se menciona en capítulos anteriores, se debe de considerar el estado transiente (cuando se inicializa el sistema y aún no es estable), también se deben de considerar dos tipos de prioridades: Una interna que se especifica en cada Job (como PRTY=), y otra dependiendo del orden en que se le asignen las clases a cada uno de los iniciadores (la primera clase asignada, tendrá mayor

CAPITULO VI.

prioridad que la segunda en posición y así sucesivamente).

El modelo además debe de guardar estadísticas de todos los eventos que ocurren en el sistema.

El modelo de simulación propuesto es el de la figura 6.1 y consta de los siguientes pasos:

I.- Inicializar las variables del sistema.

- Asignar las clases a los iniciadores.

- Asignar el número de iniciadores.

- Inicializar el área de almacenamiento (SPOOL).

II.- Eliminar los estados transientes.

1).- Verificar si hay algún iniciador disponible y si existe algún Trabajo en el SPOOL esperando que se libere algún iniciador. Si no cumple la condición pasa al punto 4.

2).- Verifica si el trabajo que está en el SPOOL, tiene una clase tal que corresponde a las clases asignadas al iniciador disponible.

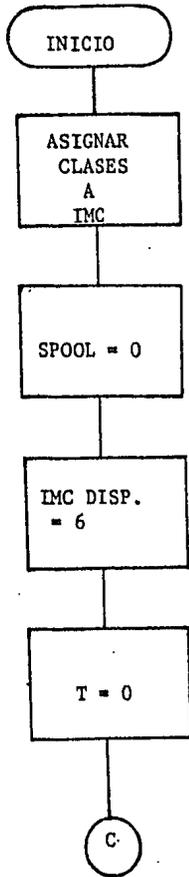
3).- Asigna el trabajo al iniciador y decrementa los contadores correspondientes tanto al iniciador como al SPOOL.

4).- Se incrementa el tiempo.

5).- Verifica si se genera alguna llegada al sistema.

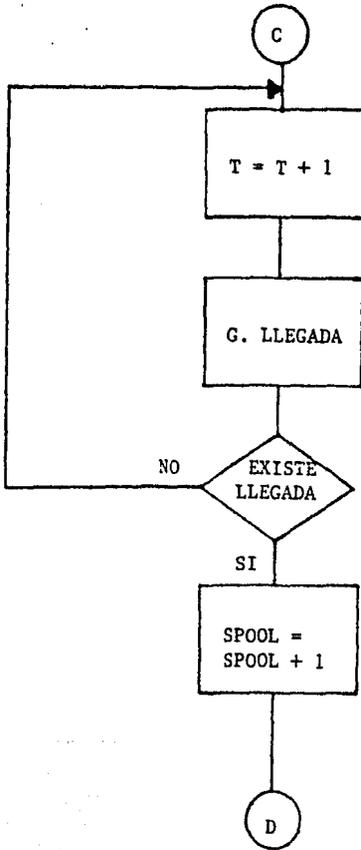
FIG. 6.1

DIAGRAMAS DE FLUJO DEL MODELO DE
SIMULACION



INICIALIZA EL SISTEMA

FIG. 6.1
(CONTINUACION)



GENERA LA PRIMERA
LLEGADA

FIG. 6.1

(CONTINUACION)

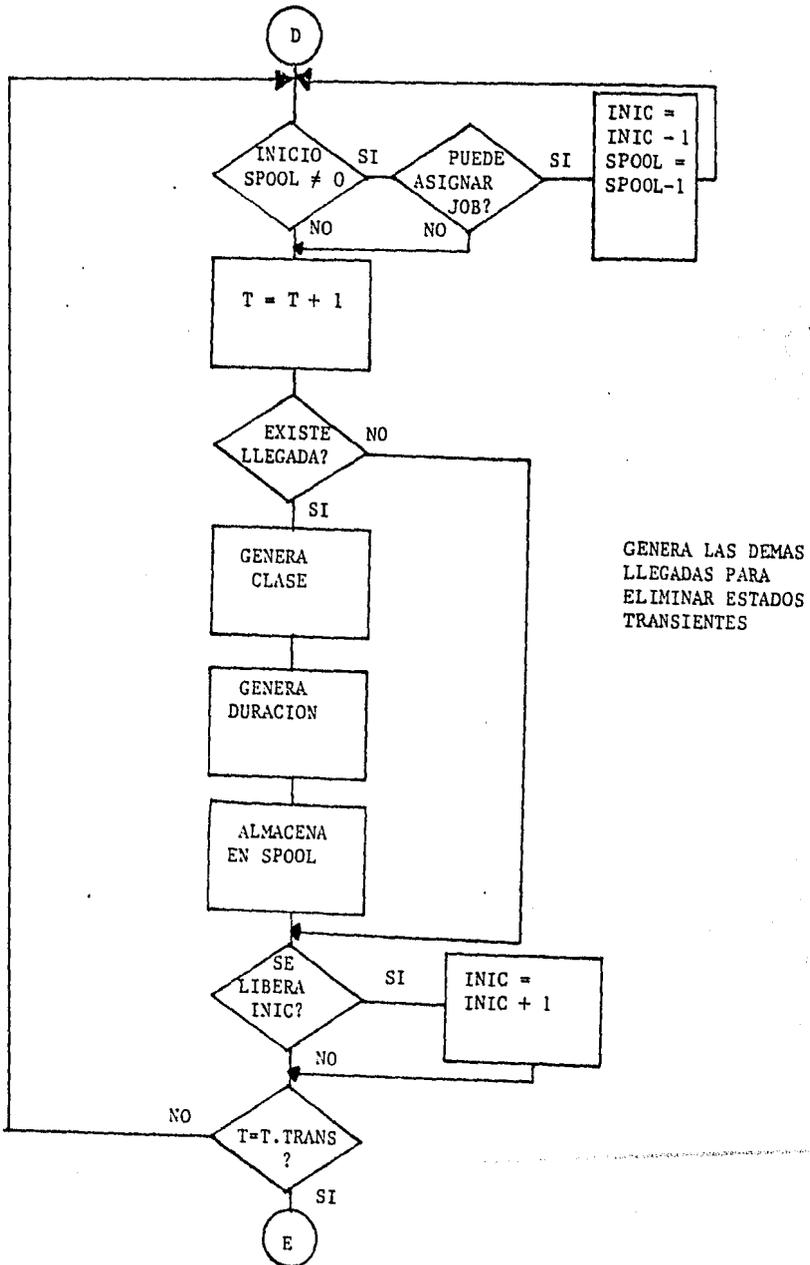
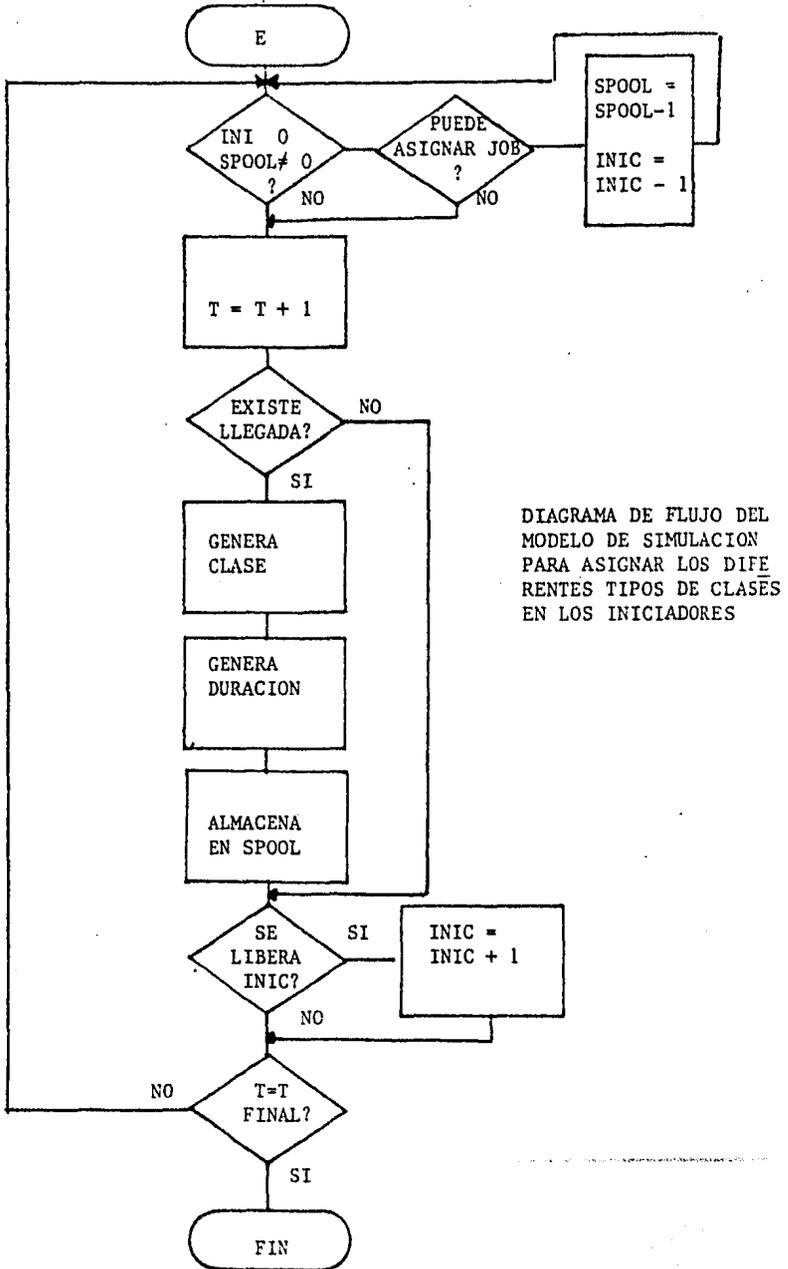


FIG. 6.1
(CONTINUACION)



CAPITULO VI.

Si no pasa al punto 9.

6).- Le genera una clase al trabajo que acaba de llegar al sistema dependiendo de una función que se obtuvo en la recolección de eventos del capítulo 5.

7).- Le genera una duración que también depende de una función generada a partir de la recolección de eventos.

8).- Lo almacena en el SPOOL e incrementa el contador del SPOOL.

9).- Verifica si se libera algún iniciador. Sino, pasa al punto 11.

10).- Incrementa el contador de iniciadores disponibles.

11).- Verifica si el tiempo 'T' es igual al tiempo asignado como estado transiente. Sino, regresa al punto 1.

12).- Se vuelven a repetir todos los pasos (1 a 11) pero asignando ahora a la variable de tiempo 'T' el valor final de tiempo que deseemos. En el paso 11 preguntaríamos si ya transcurrió el tiempo que especificamos para la simulación (sin considerar ya el estado transiente).

CAPITULO VI.

6.4.- Diagrama de bloques en el lenguaje de simulación escogido (GPSS).

Se eligió el lenguaje de simulación GPSS por dos motivos:

- Era el único paquete de simulación disponible. (Facilitado por el departamento de Gerencia de informática de la compañía ICA).
- Cumple con los requisitos necesarios para llevar a cabo el modelo de simulación requerido.

Se recomienda la simulación por medio de este paquete, ya que cumple con los requerimientos necesarios en el modelo, además de que genera unas estadísticas que nos permiten ver claramente el comportamiento del sistema.

El diagrama de bloques en GPSS es el que se muestra en la figura 6.2.

La eliminación de estados transientes, se logra haciendo una primera corrida con el tiempo deseado.

El sistema guarda el estado actual del modelo (iniciadores ocupados, trabajos en el SPOOL, y la estadística correspondiente), y sólo se tiene que vol-

CAPITULO VI.

ver a correr el modelo fijando ahora el tiempo de simulación deseado.

FIG. 6.2

DIAGRAMA DE BLOQUES EN GPSS

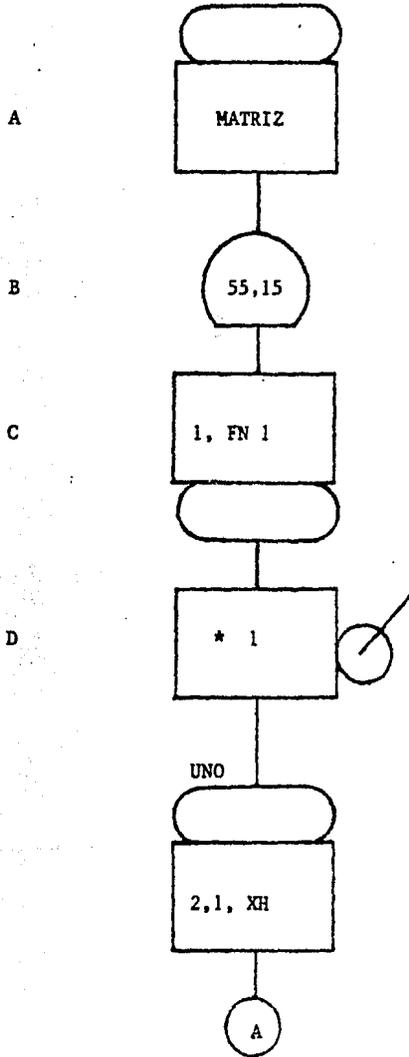


FIG. 6.2. (CONT.)

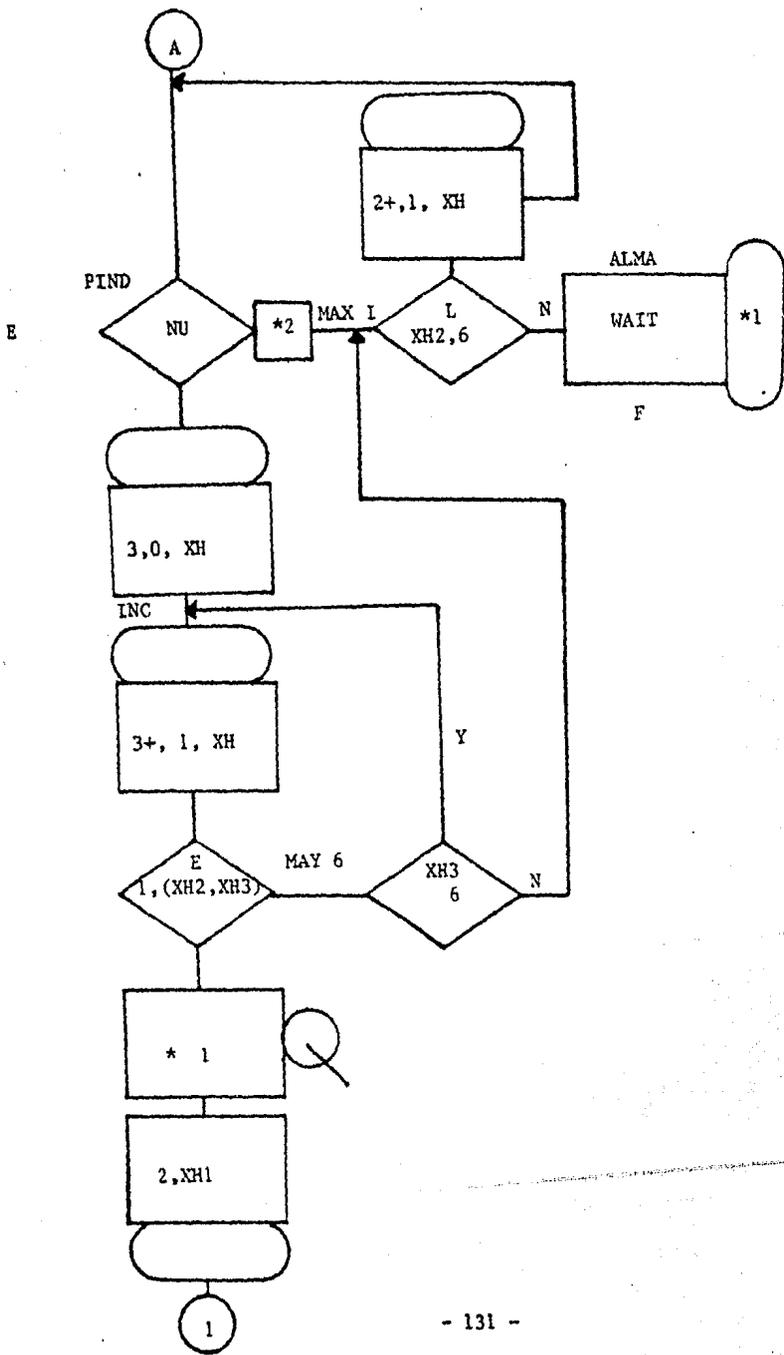


FIG. 6.2.. (CONT.)

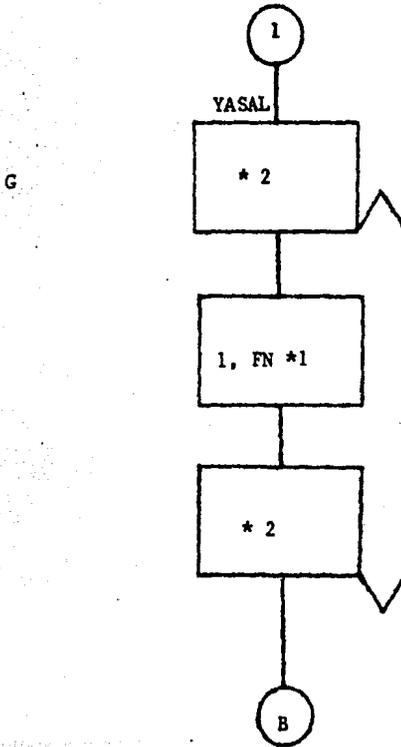


FIG. 6.2. (CONT.)

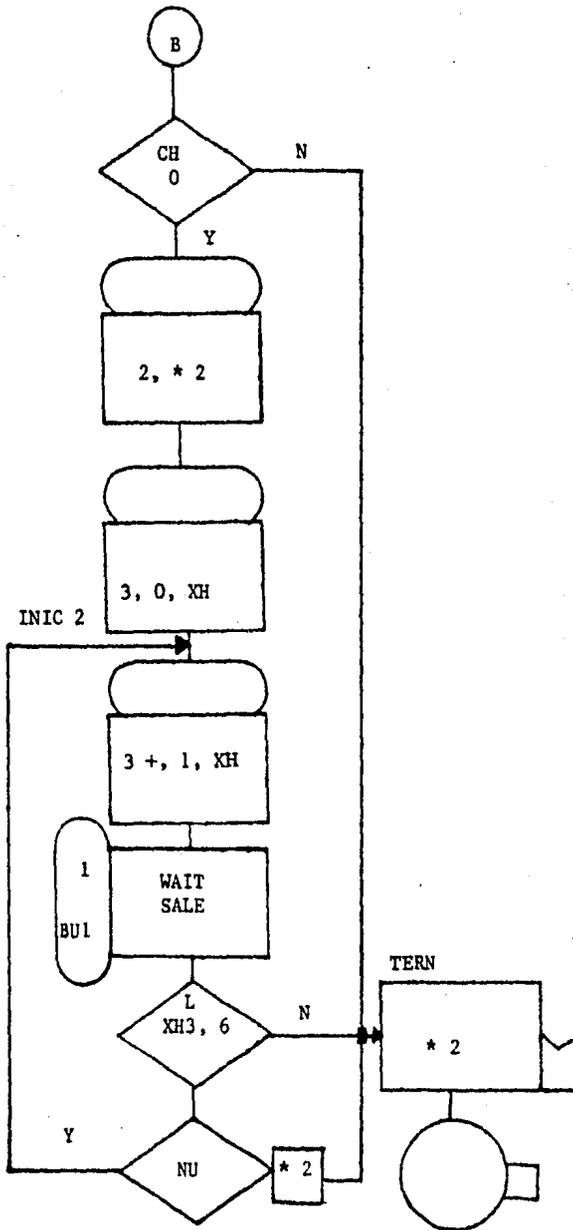
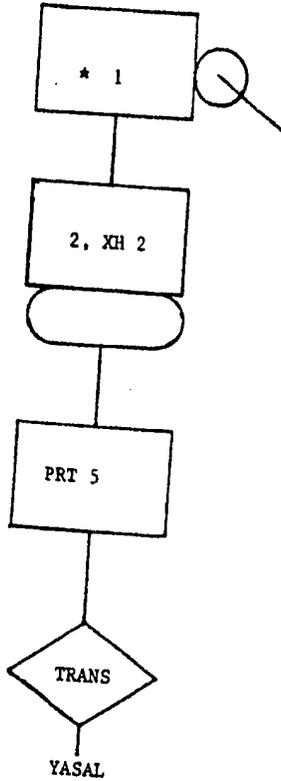


FIG. 6.2. (CONT.)

RUTINA SALE

SALE

H



CAPITULO VI.

MODELO DE SIMULACION EN GPSS.

-A, Inicializa la matriz (asigna las clases a los iniciadores).

-B, Genera llegadas. Esto lo hace generando números aleatorios con una medida y una distribución que nosotros le asignaremos en base a la información obtenida en la recolección de eventos en el capítulo V.

-C, Le asigna una clase en un parámetro del Job (trabajo) que acaba de llegar (en caso de que exista una llegada) por medio de la generación de números aleatorios aplicados a una función de distribución que nosotros definiremos en base a la recolección de eventos del capítulo V.

-D, Lo pone en una cola que se utiliza para las estadísticas del sistema.

-E, Trata de asignar el trabajo a un iniciador. Si no puede, lo almacena.

-F, Almacenamiento en el SPOOL.

-G, Asigna iniciador.

CAPITULO VI.

-H, Rutina que saca un Job del SPOOL, en caso de que pueda ser asignado a un iniciador.

Cada uno de estos bloques es una instrucción en GPSS, y están explicados en la corrida (listado) del programa.

Para mayor información, consulte el Apendice B.

C A P I T U L O V I I

APLICACION DEL MODELO DE SIMULACION.

Y ANALISIS DE RESULTADOS

CAPITULO VII

7.1 Factores a considerar para la aplicación del modelo de simulación.

Para la aplicación del modelo de simulación, se propone la siguiente metodología:

Para este modelo de simulación se va a considerar una corrida inicial de 5000 unidades de tiempo (seg) que son aproximadamente 1.4 hrs. (1 h 24 min), con el fin de eliminar los estados transientes.

En lo que respecta al tiempo de simulación, se llevará a cabo una corrida de 50,000 unidades de tiempo, esto es aprox. 14 hrs.. Se considera suficiente una corrida de este tiempo, ya que esta representa más de tres veces el período 1 (4 hrs) y más de cuatro veces el período 2 (3 hrs), que son los períodos críticos en nuestro sistema.

1.- Tomar el período más crítico (en este caso el período 2)

2.-Aplicar el modelo de simulación suponiendo que la carga de trabajo del sistema es la del período 2.

3.-Una vez que se ha obtenido la distribución adecuada (la que mejoró la producción), se simula el compor-

CAPITULO VII.

tamiento del sistema considerando ahora el otro período (período 1) como la carga actual del sistema.

Esto nos va a permitir, comprobar que la asignación de clases propuesta tiene un buen funcionamiento en el transcurso del día.

Debe notarse, que los factores que se están considerando como los que mejoran la producción son:

- Disminuir el tiempo de espera.

- Tener asignados la mayor parte del tiempo los recursos del sistema (que no esté ocioso el sistema).

En el lenguaje de simulación, a cada tarea (job) que llega al sistema se le llama 'Transacción'.

Como dijimos anteriormente, vamos a considerar que tenemos 6 iniciadores en el sistema y que podemos asignar 6 clases por cada iniciador.

Esto nos va a dar un total de 36 posiciones en las cuales se pueden asignar las 15 clases existentes (debe tenerse al menos una asignación por cada clase) pudiéndose repetir las que se consideren convenientes.

Del análisis obtenido en el Capítulo V (período 1 y período 2), se pueden observar las clases que tienen la mayor y la menor frecuencia de llegadas al sistema.

CAPITULO VII

Se va a llevar a cabo un proceso de simulación, en el cual se van a hacer diferentes asignaciones de clases a los iniciadores, para ver cual asignación tiende a ser la óptima con los mejores tiempos de espera.

Los diferentes criterios que se van a considerar para la asignación de clases son los siguientes:

- 1.- Asignar primero los más frecuentes. (las clases más frecuentes),
- 2.- Asignar primero los menos frecuentes.
- 3.- Asignar los más cortos primero.
- 4.- Asignar los más cortos con menor frecuencia primero.
- 5.- Combinar los más frecuentes con los menos frecuentes y viceversa.
- 6.- Repetir más veces los más frecuentes al final para asegurar que todos los iniciadores trabajen.

De estos criterios se seleccionarán los más eficientes, y se harán modificaciones hasta encontrar la mejor asignación.

El programa de simulación nos da los siguientes resultados:

CAPITULO VII

- Absolute Clock, el cual nos da el tiempo total de corrida. (tiempo de simulación + estado transiente).

- Relative Clock, el cual nos da el tiempo de corrida, sin considerar las condiciones iniciales. (T. de simulación).

En la misma hoja nos da el número de transacciones (tareas) que pasaron por cada uno de los bloques, así como los que contiene actualmente (al finalizar la corrida).

-User Chains, nos da el uso que tuvo nuestro almacenamiento (Spool) dándonos el número total de entradas, el contenido actual, el máximo contenido que tuvo en un tiempo, así como el promedio de tiempo de estancia de cada Job en el Spool.

- Facilities, ésta nos va a dar el promedio de uso de los iniciadores. Nos dice cuantas entradas tuvo cada iniciador, el promedio del tiempo de estancia de cada transacción (job) en el iniciador, y el promedio de utilización de cada iniciador (% de tiempo que estuvo ocupado cada iniciador).

Queues, nos da el tiempo promedio de espera de cada clase que llega al sistema (\$AVERAGE TIME/TRANS) así como el total de llegadas de cada clase al sistema.

- Tables, después vienen 6 tablas (una por cada inicia-

CAPITULO VII

dor) las cuales nos dan el total de llegadas a cada uno de los iniciadores, así como el número de llegadas de cada clase y el porcentaje que ésta representó en dicho iniciador.

- Half Word Matrices, la cual nos imprime la asignación de clases en esa corrida (en forma numérica).

Para poder aplicar el modelo de simulación se tiene que hacer una asignación numérica a cada una de las clases, ya que el sistema no trabaja con caracteres.

La asignación propuesta es la siguiente:

NUMERO ASIGNADO-CLASE

1-C	2-D	3-F	4-G	5-H
6-I	7-J	8-L	9-M	10-N
11-P	12-R	13-S	14-V	15-1.

CAPITULO VII

7.2 ANALISIS DE RESULTADOS.

La figura 7.1 nos muestra los diferentes criterios propuestos para la asignación de clases a los iniciadores (en forma numérica).

En la tabla 7.1, se muestran los resultados obtenidos al aplicar al modelo de simulación la distribución de clases actual.

El resultado obtenido al aplicar el modelo de simulación a los 6 diferentes criterios propuestos (Fig 7.1), están resumidos en la Tabla RESUMEN.

En la tabla resumen vamos a tener dos secciones: La primera, denominada FACILITIES, nos muestra el porcentaje de uso de cada uno de los iniciadores, ésto es, el tiempo que estuvieron asignados dichos iniciadores (renglones) con respecto a cada uno de los criterios asignados (columnas).

En el último renglón de esa tabla, viene el promedio de uso de todos los iniciadores por cada uno de los diferentes criterios de asignación.

La segunda sección denominada QUEUE, nos muestra la clase y el tiempo promedio de espera por clase en cada uno de los diferentes criterios de asignación (columnas).

FIG. 7.1

CRITERIOS DE ASIGNACION DE CLASES A INICIADORES

1	1	2	3	4	5	6
1	14	9	15	8	4	1
2	4	7	11	2	13	15
3	13	5	1	6	14	11
4	1	8	10	13	11	4
5	11	2	3	4	15	14
6	15	6	12	14	1	13

2	1	2	3	4	5	6
1	10	1	13	4	14	15
2	3	11	4	14	13	1
3	12	15	14	6	1	4
4	9	6	13	1	2	14
5	7	2	4	11	8	13
6	5	8	14	15	11	4

- 1 LOS MAS FRECUENTES PRIMERO
- 2 LOS MENOS FRECUENTES PRIMERO

FIG. 7.1
(CONTINUACION)

3	1	2	3	4	5	6
1	12	4	5	13	1	11
2	6	11	8	1	14	13
3	13	7	6	11	4	14
4	3	10	14	15	13	3
5	15	9	4	2	1	13
6	14	2	8	6	4	15

4	1	2	3	4	5	6
1	12	9	2	14	4	1
2	6	14	11	1	15	4
3	3	4	5	14	11	1
4	13	11	15	6	4	5
5	7	15	14	8	13	15
6	10	8	4	2	13	14

- 3 LOS MAS CORTOS PRIMERO
- 4 LOS MAS CORTOS CON MENOR FRECUENCIA PRIMERO

FIG. 7.1
(CONTINUACION)

5	1	2	3	4	5	6
1	14	15	8	9	10	13
2	4	11	2	7	3	14
3	3	1	6	5	12	4
4	10	7	6	1	4	13
5	3	9	2	11	14	1
6	12	5	8	15	13	14

6	1	2	3	4	5	6
1	6	10	4	14	15	13
2	2	3	13	4	11	15
3	8	12	14	13	1	4
4	5	9	15	1	13	14
5	7	2	14	11	4	1
6	1	6	4	15	14	11

- 5 MAS FRECUENCIA CON LOS MENOS FRECUENTES
 6 REPETIR MAS VECES LAS MAS FRECUENTES AL FINAL PARA ASEGURAR QUE TRABAJEN TODOS LOS INICIADORES.

TABLA RESUMEN
CRITERIOS DE ASIGNACION

F A C I L I T I E S

INIC.	1	2	3	4	5	6
1	.941	.902	.926	.948	.765	.812
2	.973	.914	.918	.935	.963	.854
3	.885	.820	.860	.908	.889	.938
4	.850	.832	.876	.860	.896	.836
5	.868	.819	.916	.671	.904	.829
6	.776	.828	.861	.779	.657	.826
PROM.	.882	.852	.892	.850	.845	.849

Q U E U E

C L A S E	1	2	3	4	5	6
C 1	7490	4841	6536	11419	15650	6479
D 2	1248	1599	1911	1318	3156	1053
F 3	490	694	52	604	470	317
G 4	18425	7045	5398	6544	10661	4658
H 5	23	3395	299	1496	2352	749
I 6	2551	4353	1370	3130	6295	1411
J 7	912	1582	492	0	481	704
L 8	1992	1572	885	465	971	6730
M 9	331	459	966	1047	27	3222
P 11	4532	7642	8276	9915	17707	9266
R 12	3209	1186	2869	2371	384	2760
S 13	6239	3615	4805	4039	7003	3812
V 14	9767	9104	7528	5586	6166	4817
l 15	1645	4191	6165	2844	5750	1935
TOT.	58834	51258	47552	50779	77073	47913
PROM.	4202	3661	3397	3627	5505	3422

- EN ESTA TABLA SE RESUME EL % DE USO DE LOS INIC. Y EL PROMEDIO
TIEMPO DE ESPERA POR CLASE AL APLICAR EL MODELO DE SIMULACION
CON LOS PRIMEROS 6 CRITERIOS DE ASIGNACION (DE LA FIG. 7.1)

T A B L A 7. 1.

Tabla que nos da el tiempo total de espera por clase del sistema con la asignación de clases actual.

FACILITIES

INIC.	PERIODO 1	PERIODO 2
1	.849	.995
2	.867	.998
3	.810	1.000
4	.827	.851
5	.758	.806
6	.410	.625
PROM.	.735	.879

Q U E U E

C L A S E	TOT. ENT.	AV. TIM/TRAN	T. ESP. TOT.	TOT. ENT.	AV. TIM/TRAN	T. ESP. TOT.
C 1	121	218.1	26390	77	98.4	7576
D 2	22	110.1	2422	00	00	00
F 3	1	131	131	1	12	12
G 4	158	30.5	3819	153	13.1	2004
H 5	11	1118.1	12299	6	668.1	4009
I 6	47	50.1	2355	8	96.3	770
J 7	5	190.7	954	3	91.6	275
L 8	16	140.5	2248	24	139.1	3338
M 9	3	00	00	4	65	260
N 10	2	34	68	4	159	636
P 11	108	40	4320	116	56.3	6531
R 12	4	179.7	719	2	495.5	991
S 13	109	53.8	5864	41	55.7	2283
V 14	195	262.8	51246	144	100	14400
1 15	104	42.1	4378	131	48.4	6340
PROM.			118208			49425

CAPITULO VII

Se muestra también el total y el promedio del tiempo de espera de cada uno de los criterios de asignación.

Analizando esta tabla, se observa que las mejores opciones son la 3 y la 4, ya que éstas son las que tienen los mejores tiempos de espera, y además tienen un buen porcentaje de uso.

Una vez que hemos seleccionado el criterio para la asignación de clases (en este caso asignando los trabajos más cortos primero), el siguiente paso es el de crear una nueva serie de asignaciones basándose en dichos criterios, con el fin de optimizar estas asignaciones.

Estos criterios se hacen cambiando sólo algunas asignaciones, aplicando el modelo de simulación y viendo como se comporta el sistema.

En este punto, también vamos a considerar otro factor muy importante para poder tomar la decisión final, y es que también se va a realizar la simulación pero ahora considerando la carga de trabajo del Período 1.

La asignación final, dependerá del criterio con el que obtengamos los mejores tiempos de espera tanto en el Período 1 como en el Período 2.

La tabla 7.2, nos muestra los resultados obtenidos al aplicar dichos criterios al modelo de simulación.

T A B L A 7.2

Tabla que se obtiene al aplicar el grupo de criterios al
MODELO DE SIMULACION.

FACILITIES PERIODO 2

INIC.	1	2	3	4	5	6
1	.974	.952	.969	.944	.940	.926
2	.967	.959	.981	.987	.994	.986
3	.956	.952	.957	.931	.914	.839
4	.924	.945	.970	.982	.980	.949
5	.944	.919	.950	.962	.984	.971
6	1.000	.986	.929	.970	.969	.957
PROM.	.96	.952	.959	.962	.963	.938

Q U E U E

C L A S E	1	2	3	4	5	6
C 1	129	104	90	93	151	126
D 2	136	285	618	216	227	112
F 3	112	979	87	179	298	111
G 4	110	78	69	77	74	57
H 5	1811	865	1097	2473	1770	1849
I 6	67	103	89	124	97	70
J 7	146	67	201	468	427	461
L 8	396	651	326	371	431	452
M 9	308	513	743	652	804	360
N 10	538	152	344	205	105	308
P 11	80	76	70	76	75	60
R 12	880	367	154	430	317	586
S 13	64	59	96	87	89	87
V 14	87	68	61	61	59	52
l 15	160	164	79	74	90	97
PROM. T.	5024	4531	4124	5407	5015	4788
PROM.	334.9	302	274.9	360.4	334.3	319.2

TABLA 7.2 (Cont.)

P E R I O D O 2

TABLA QUE NOS DA EL TIEMPO TOTAL DE ESPERA POR CLASE, EN EL PERIODO 2.

QUEUE. Frecuencia por tiempo promedio de espera/JOB.
 Frace de llegadas X tiemp. prom. de espera por clase = T.
 esp. tot. por clase.

C L A S E	1	2	3	4	5	6
C 1	11446	9160	8213	8885	15799	11844
D 2	3128	5707	12367	4761	4542	2247
F 3	112	979	174	358	298	222
G 4	3344	10064	8629	9638	8999	5809
H 5	18110	9522	9873	29676	21242	16649
I 6	2880	3730	3901	4592	3494	2875
J 7	880	404	1207	2809	2994	2309
L 8	6341	9124	4565	5564	6907	7686
M 9	617	1026	2229	1958	2413	1080
N 10	538	152	689	410	210	617
P 11	7696	7300	7077	8155	7644	4925
R 12	3520	1836	770	2150	1585	1760
S 13	5424	5264	9085	9062	9141	9067
V 14	13431	10811	8767	8499	9081	6025
I 15	13431	14147	7660	7170	9345	8704
TOT.	90949	75079	85206	103687	103691	81819

F A C I L I T I E S

INIC.	1	2	3	4	5	6
1	.974	.952	.969	.944	.940	.926
2	.967	.959	.981	.987	.994	.986
3	.956	.952	.957	.931	.914	.839
4	.924	.945	.970	.982	.980	.949
5	.944	.919	.950	.962	.984	.971
6	1.000	.986	.929	.970	.969	.957
PROM.	.960	.952	.959	.962	.963	.938

T A B L A 7. 3.

CAPITULO VII

La tabla 7.3 se obtiene multiplicando la frecuencia de llegadas de cada una de las clases por el tiempo promedio de espera por clase.

Esto nos permite tener como resultado, el tiempo de espera total de cada una de las clases en el Período 2.

Las tablas 7.4 y 7.5 nos da la misma información que las tablas 7.2 y 7.3 pero para el Período 1.

Haciendo un analisis a estas últimas tablas (7.4 y 7.5), y tomando en cuenta cuales clases consideramos que deban tener el menor tiempo de espera (depende de las prioridades de cada sistema) podemos elegir la mejor asignacion de clases.

Se puede observar que las asignaciones que tienen los mejores tiempos de espera son los criterios 2 y 6. Ahora bien, si le damos mayor prioridad a las clases que tienen mayor frecuencia de llegadas, vemos que la mejor opción es la del criterio número 6 que se muestra en la tabla 7.6.

PERIODO I
FACILITIES

INIC.	1	2	3	4	5	6
1	.894	.835	.696	.8	.8	.785
2	.929	.886	.857	.862	.862	.868
3	.744	.725	.838	.743	.743	.769
4	.784	.735	.705	.685	.685	.649
5	.649	.716	.768	.822	.822	.816
6	.632	.602	.527	.494	.494	.526
PROM.	.772	.749	.731	.734	.734	.735

QUEUES

CLASE	1	2	3	4	5	6
C 1	89	92	86	94	94	78
D 2	0	0	0	0	0	0
F 3	171	597	107	0	0	207
G 4	69	66	57	44	44	42
H 5	1441	539	4495	1277	1277	1689
I 6	84	94	34	61	61	155
J 7	10	32	396	315	315	357
L 8	874	733	731	773	773	704
M 9	136	197	679	136	136	483
N 10	396	683	393	143	143	700
P 11	70	87	73	91	91	75
R 12	802	477	167	517	517	772
S 13	63	59	50	59	59	74
V 14	91	87	54	53	53	44
l 15	95	116	55	70	70	91
PROM. TOT.	4341	3859	7377	3633	3633	5471
PROM.	313	275	526	259	259	390

T A B L A 7. 4.

- TABLA QUE SE OBTIENE AL ASIGNAR EL SEGUNDO CRUPO DE
CRITERIOS AL PERIODO I.

T A B L A 7. 5.

PERIODO 1

TABLA QUE NOS DA EL TIEMPO TOTAL DE ESPERA POR CLASE, EN EL PERIODO 2.

C L A S E	1	2	3	4	5	6
C 1	1684	1932	1214	2580	2460	2115
D 2	0	0	0	0	0	0
F 3	342	597	107	0	0	207
G 4	2702	2794	1930	1491	1491	1186
H 5	7206	2158	4495	5110	5110	5068
I 6	337	378	33	366	366	465
J 7	21	32	1188	947	947	1073
L 8	18362	12470	12432	12374	12374	11275
M 9	136	395	679	272	272	966
N 10	1190	1366	1181	573	573	2100
P 11	3254	2262	2198	3006	3006	2038
R 12	1604	955	334	1035	1035	1545
S 13	634	653	506	538	538	644
V 14	3566	3665	1131	1223	1223	610
1 15	3733	5580	3118	3822	3822	5227
TOT.	44771	35232	30546	33337	33317	34539

F A C I L I T I E S

INIC.	1	2	3	4	5	6
1	.894	.835	.696	.8	.8	.785
2	.929	.886	.857	.862	.862	.868
3	.744	.725	.838	.743	.743	.769
4	.784	.735	.705	.685	.685	.649
5	.649	.718	.768	.822	.822	.816
6	.632	.602	.527	.494	.494	.526
PROM.	.772	.749	.731	.734	.734	.735

T A B L A 7.6

REPRESENTACION NUMERICA

INI	1	2	3	4	5	6
1	12	1	9	2	14	4
2	2	14	11	1	15	14
3	10	3	4	5	14	11
4	13	11	15	6	4	14
5	13	15	14	7	8	1
6	6	13	14	4	11	1

REPRESENTACION REAL

INI	1	2	3	4	5	6
1	R	C	M	D	V	G
2	D	V	P	C	I	G
3	N	F	G	H	V	P
4	S	P	I	I	G	V
5	S	I	V	J	L	C
6	I	S	V	G	P	C

BLOCK NUMBER	LOC #	OPERATION	A, B, C, D, E, F, G, H, I	COMMENTS	STATEMENT NUMBER
1		SIMULATE			1
		GENERATE	...1		2
	1	MATRIX	MN.6.6	DECLARACTN DE LA MATRIZ	3
2		MSAVEVALUE	1.1,1,12,MH		4
3		MSAVEVALUE	1.1,2,1,MH		5
4		MSAVEVALUE	1.1,3,9,MH		6
5		MSAVEVALUE	1.1,4,2,MH		7
6		MSAVEVALUE	1.1,5,14,MH		8
7		MSAVEVALUE	1.1,6,4,MH		9
8		MSAVEVALUE	1.2,1,2,MH		10
9		MSAVEVALUE	1.2,2,14,MH	INICIALIZA LA MATRIZ	11
10		MSAVEVALUE	1.2,3,11,MH		12
11		MSAVEVALUE	1.2,3,11,MH	ASIGNA LAS CLASES A CADA	13
12		MSAVEVALUE	1.2,4,1,MH	UNO DE LOS INICIADORES	14
13		MSAVEVALUE	1.2,5,15,MH		15
14		MSAVEVALUE	1.2,6,4,MH		16
15		MSAVEVALUE	1.3,1,10,MH		17
16		MSAVEVALUE	1.3,2,3,MH		18
17		MSAVEVALUE	1.3,3,4,MH		19
18		MSAVEVALUE	1.3,4,5,MH		20
19		MSAVEVALUE	1.3,5,14,MH		21
20		MSAVEVALUE	1.3,6,11,MH		22
21		MSAVEVALUE	1.4,1,13,MH		23
22		MSAVEVALUE	1.4,2,11,MH		24
23		MSAVEVALUE	1.4,3,15,MH		25
24		MSAVEVALUE	1.4,4,6,MH		26
25		MSAVEVALUE	1.4,5,4,MH		27
26		MSAVEVALUE	1.4,6,14,MH		28
27		MSAVEVALUE	1.5,1,13,MH		29
28		MSAVEVALUE	1.5,2,15,MH		30
29		MSAVEVALUE	1.5,3,14,MH		31
30		MSAVEVALUE	1.5,4,7,MH		32
31		MSAVEVALUE	1.5,5,8,MH		33
32		MSAVEVALUE	1.5,6,1,MH		34
33		MSAVEVALUE	1.6,1,6,MH		35
34		MSAVEVALUE	1.6,2,13,MH		36
35		MSAVEVALUE	1.6,3,14,MH		37
36		MSAVEVALUE	1.6,4,4,MH		38
37		MSAVEVALUE	1.6,5,11,MH		39
38		MSAVEVALUE	1.6,6,1,MH		40
		TERMINATE			41
39		PRINT	1.1,MH		42
40		GENERATE	55,15		43
41		ASSIGN	1,PHCLASE	GENERA LAS LLEGADAS	44
42		QUEUE	*1	ASIGNA CLASE AL JOB	45
43	UNO	SAVEVALUE	2.1,XH	METE EN COLA DE ESPERA DE I.	46
44	PIND	GATE NO	XH2,MAXI	VA AL PRIMER INICIADOR	47
45		SAVEVALUE	3,0,XH	PRUEBA DISPON. DE INIC. N	48
46	IHC	SAVEVALUE	3+1,XH	INICIAL. CONT. DE COLUMNAS	49
47		TEST E	M1,MH1(XH2,XH3),MAY5	INCREMENTA CONTADOR DE COL.	50
48		DEPART	*1	SACA DE LA COLA ESA CLASE SI NO SE CA A MAY6.	51
49		ASSIGN	2,XH2	ASIGNA AL PAR. 2 el VALOR DE LA VAR XH2	52
50	YASAL	SLIZE	*2	ASIGNA INICIAZOR	53
					54
					55

51	ADVANCE	1,FH*1	ASIGNA DURACION Y DETIENE	56
52	RELEASE	*2	LIBERA INICIADOR H	57
53	TEST L	CH\$WAIT,0,TERM	SI ESTA VACIA LA CADENA TERM	58
54	SAVEVALUE	2,*2,XH	Asigna a la var. XH2, el parametro 2 del job actual	59
55	SAVEVALUE	3,0,XH	Asigna la var. XH3 y la incrementa.	60
56	SAVEVALUE	4,1,XH		61
57	INCLINK	WAIT,SALE,1,DV1	SI NO TRATA DE ASIGNAR UN JOB	62
58	TEST L	XH3,6,TERM	PRUEBA SI HAY MAS COL.	63
59	GATE HU	XH2,TERM	PREGUNTA SI YA ASIGNO INIC.	64
60	TRANSFER	,INC2	SINO INCREMENTA LA COL.	65
61	TERM	TABULATE	TABLA DE FRECC DEL INI. N	66
62	TERMINATE	*2	REGISTRA SALIDA DEL JOB	67
				68
63	* ALMA LINK	WAIT,P1	CADENA DE ALMACENAMIENTO	69
				70
64	* MAY6 TEST L	XH3,6,MAXI	Prueba si ya checo todas las clases asignadas a	71
65	TRANSFER	,INC	dicho iniciador. Si no incrementa la columna.	72
			Si ya los revizo todos va a MAXI	73
66	* SALE DEPART	*1	Saca al job de la cola, le asigna prioridad y lo	74
67	ASSIGN	2,XH2	manda a asignarle iniciador.	75
68	PRIORITY	5		76
69	TRANSFER	,YASAL		77
				78
	1	TABLE P1,1,1,20	Estas Tablas contienen la estadistica de cada uno	79
	2	TABLE P1,3,1,20	de los iniciadores (cuantos llegaron, que clases	80
	3	TABLE P1,1,1,20	se le asignaron a cada uno de ellos).	81
	4	TABLE P1,1,1,20		82
	5	TABLE P1,1,1,20		83
	6	TABLE P1,1,1,20		84
				85
				86
				87
	CLASE FUNCTION	RM3,D15		88
	.1222,1/,144,2/,1471,3/,3464,4/,3549,5/,3958,6/,3989,7		Function para la asignacion de clases.	89
	.4138,8/,4169,9/,42,10/,5383,11/,5414,12/,6772,13			90
	.8843,14/1.0,15			91
		FUNC ASIGNA DURACION		92
				93
	1	DVARIABLE P1'E'PHI(XH2,XH3)	Variable Booleana para tratar de asignar un Job.	94
				95
70	* MAXI TEST L	XH2,6,ALMA	Prueba si ya checo todos los iniciadores. Si ya termino	96
71	SAVEVALUE	2*,1,XH	va a ver si ya puede asignar otro (PIND). Si no	97
72	TRANSFER	,PIND	incrementa el renglon.	98
				99
	1	FUNCTION RH1,C5		100
	.2,400/,4,720/,6,1040/,8,1360/1.0,1680		Function para la asignacion de clases.	101
	2	FUNCTION RH2,C5		102
	.2,700/,4,840/,6,980/,8,1120/1.0,1260			103
	3	FUNCTION RH3,C5		104
	.2,450/,4,550/,6,650/,8,750/1.0,85			105
	4	FUNCTION RH4,C5		106
	.2,120/,4,140/,6,160/,8,180/1.0,200			107
	5	FUNCTION RH5,C10		108
	.1,1900/,2,2273/,3,2646/,4,3019/,5,3392/,6,3765/,7,4138			109
	.8,4511/,9,4884/1.0,5257			110
	6	FUNCTION RH6,C5		111
	.2,200/,4,260/,6,320/,8,380/1.0,45			112
	7	FUNCTION RH7,C5		113
	.2,700/,4,830/,6,960/,8,1090/1.0,125			

8	FUNCTION	RH2,C5		114
.2,900/	.4,982/	.6,1063/	.8,1145/1.0,1227	115
9	FUNCTION	RH3,C5		116
.2,400/	.4,450/	.6,500/	.8,550/1.0,600	117
10	FUNCTION	RH4,C5		118
.2,80/	.4,90/	.6,100/	.8,110/1.0,120	119
11	FUNCTION	RH5,C5		120
.2,300/	.4,335/	.6,350/	.8,375/1.0,400	121
12	FUNCTION	RH6,C5		122
.2,5/	.4,8/	.6,11/	.8,16/1.0,17	123
13	FUNCTION	RH7,C5		124
.2,30/	.4,50/	.6,70/	.8,90/1.0,110	125
14	FUNCTION	RH4,C5		126
.2,80/	.4,110/	.6,140/	.8,170/1.0,200	127
15	FUNCTION	RH3,C5		128
.2,100/	.4,160/	.6,220/	.8,280/1.0,340	129
N				130
73	GENERATE	1000		131
74	TERMINATE	1		132
	START	5,MP	Elimina estados transientes. (5000 seg.)	133
	RESET			134
	START	50	Corrida de simulacion. (50,000 seg.)	135
	END			136

CROSS-REFERENCE
BLOCKS

SYMBOL	NUMBER	REFERENCES
ALMA	63	95
IHC	46	72
IHC2	56	65
MAXI	70	49 71
MAY6	64	52
PIND	44	97
SALE	66	62
TERM	61	58 63 64
UHO	43	
YASAL	50	77

^ SIMULATE SIMULA AREA DE TRABAJO DE CPU

1 GENERATE ,,,1
1 MATRIX MH,6,6

2 MSAVEVALUE 1,1,1,12,MH
3 MSAVEVALUE 1,1,2,1,MH
4 MSAVEVALUE 1,1,3,9,MH
5 MSAVEVALUE 1,1,4,2,MH
6 MSAVEVALUE 1,1,5,14,MH
7 MSAVEVALUE 1,1,6,4,MH
8 MSAVEVALUE 1,2,1,2,MH
9 MSAVEVALUE 1,2,2,14,MH
10 MSAVEVALUE 1,2,3,11,MH
11 MSAVEVALUE 1,2,4,1,MH
12 MSAVEVALUE 1,2,5,15,MH
13 MSAVEVALUE 1,2,6,4,MH
14 MSAVEVALUE 1,3,1,10,MH
15 MSAVEVALUE 1,3,2,3,MH
16 MSAVEVALUE 1,3,3,4,MH
17 MSAVEVALUE 1,3,4,5,MH
18 MSAVEVALUE 1,3,5,14,MH
19 MSAVEVALUE 1,3,6,11,MH
20 MSAVEVALUE 1,4,1,13,MH
21 MSAVEVALUE 1,4,2,11,MH
22 MSAVEVALUE 1,4,3,15,MH
23 MSAVEVALUE 1,4,4,6,MH
24 MSAVEVALUE 1,4,5,4,MH
25 MSAVEVALUE 1,4,6,14,MH
26 MSAVEVALUE 1,5,1,13,MH
27 MSAVEVALUE 1,5,2,15,MH
28 MSAVEVALUE 1,5,3,14,MH
29 MSAVEVALUE 1,5,4,7,MH
30 MSAVEVALUE 1,5,5,8,MH
31 MSAVEVALUE 1,5,6,1,MH
32 MSAVEVALUE 1,6,1,6,MH
33 MSAVEVALUE 1,6,2,13,MH
34 MSAVEVALUE 1,6,3,14,MH
35 MSAVEVALUE 1,6,4,4,MH
36 MSAVEVALUE 1,6,5,11,MH
37 MSAVEVALUE 1,6,6,1,MH
38 TERMINATE

PROGRAMA ENSAMBLADO.

39 PRINT 1,1,MH
40 GENERATE 55,15
41 ASSIGN 1,FH16
42 QUEUE *1
43 SAVEVALUE 2,1,XH
44 GATE MH XH2,70
45 SAVEVALUE 3,0,XH
46 SAVEVALUE 3,1,XH
47 TEST E *1,MH1(XH2,XH3),64
48 DEPART *1
49 ASSIGN 2,XH2
50 SETE *2
51 ADVANCE 1,FH1
52 RELEASE *2
53 TEST G CH1,0,61
54 SAVEVALUE 2,*2,XH

55 SAVEVALUE 3,0,XH
 56 SAVEVALUE 3,1,XH
 57 UNLINK 1,66,1,8V1
 58 TEST L XH3,6,61
 59 GATE HU XH2,61
 60 TRANSFER ,56
 61 TRAILATE *2
 62 TERMINATE

63 LINK 1,P1

64 TEST L XH3,6,70
 65 TRANSFER ,46

66 DEPART *1
 67 ASSIGN 2,XH2
 68 PRIORITY 5
 69 TRANSFER ,50

1 TABLE P1,1,1,20
 2 TABLE P1,1,1,20
 3 TABLE P1,1,1,20
 4 TABLE P1,1,1,20
 5 TABLE P1,1,1,20
 6 TABLE P1,1,1,20

FUNCTION	RH3, D15				
.1222	1	.146	2	.1471	3
.3464	4	.3569	5	.3958	6
.3989	7	.4138	8	.4169	9
.42	10	.5383	11	.5414	12
.6772	13	.8843	14	1.0	15

FUNC ASTONA DURACION

1 BVARIABLE P1*E*MH(XH2,XH3)

70 TEST L XH2,6,63
 71 SAVEVALUE 2,1,XH
 72 TRANSFER ,49

FUNCTION	RH1, C5				
.1	400	.4	720	.6	1040
.8	1360	1.0	1680		
.2	FUNCTION	RH2, C5			
.2	700	.4	840	.6	980
.3	1120	1.0	1260		
.2	FUNCTION	RH3, C5			
.2	45	.4	55	.6	55
.4	75	1.0	85		
.4	FUNCTION	RH4, C5			
.2	120	.4	140	.6	160
.8	180	1.0	200		
.5	FUNCTION	RH5, C10			
.1	1900	.2	2273	.3	2646
.4	3019	.5	3392	.6	3765
.7	4138	.8	4511	.9	4884
1.0	5257				
.6	FUNCTION	RH6, C5			
.2	20	.4	26	.6	32

.8	38	1.0	45		
.7	FUNCTION	RN7,C5			
.2	70	.4	83	.6	96
.8	109	1.0	125		
.8	FUNCTION	RN2,C5			
.2	900	.4	982	.6	1063
.8	1145	1.0	1227		
.9	FUNCTION	RN3,C5			
.2	400	.4	450	.6	500
.8	550	1.0	600		
10	FUNCTION	RN4,C5			
.2	80	.4	90	.6	100
.8	110	1.0	120		
11	FUNCTION	RN5,C5			
.2	300	.4	325	.6	350
.8	375	1.0	400		
12	FUNCTION	RN6,C5			
.2	5	.4	8	.6	11
.8	14	1.0	17		
13	FUNCTION	RN7,C5			
.2	30	.4	50	.6	70
.8	90	1.0	110		
14	FUNCTION	RN4,C5			
.2	80	.4	110	.6	140
.8	170	1.0	200		
15	FUNCTION	RN3,C5			
.2	100	.4	160	.6	220
.8	280	1.0	340		
73	GENERATE	1000			
74	TERMINATE	1			
	START	5,NP			
	RESET				
	START	50			

RELATIVE CLOCK		50000 ABSOLUTE CLOCK			55000									
BLOCK	COUNTS	TOTAL	BLOCK	CURRENT	TOTAL	BLOCK	CURRENT	TOTAL	BLOCK	CURRENT	TOTAL	BLOCK	CURRENT	TOTAL
1	0	0	11	0	0	21	0	0	31	0	0	41	0	907
2	0	0	12	0	0	22	0	0	32	0	0	42	0	907
3	0	0	13	0	0	23	0	0	33	0	0	43	0	907
4	0	0	14	0	0	24	0	0	34	0	0	44	0	4813
5	0	0	15	0	0	25	0	0	35	0	0	45	0	363
6	0	0	16	0	0	26	0	0	36	0	0	46	0	1864
7	0	0	17	0	0	27	0	0	37	0	0	47	0	1864
8	0	0	18	0	0	28	0	0	38	0	0	48	0	220
9	0	0	19	0	0	29	0	0	39	0	0	49	0	220
10	0	0	20	0	0	30	0	0	40	0	907	50	0	901
BLOCK	CURRENT	TOTAL	BLOCK	CURRENT	TOTAL	BLOCK	CURRENT	TOTAL	BLOCK	CURRENT	TOTAL	BLOCK	CURRENT	TOTAL
51	6	901	61	0	901	71	0	3906						
52	0	901	62	0	901	72	0	3906						
53	0	901	63	0	687	73	0	50						
54	0	732	64	0	1664	74	0	50						
55	0	732	65	0	1501									
56	0	4392	66	0	685									
57	0	4392	67	0	685									
58	0	4392	68	0	685									
59	0	3660	69	4	6A5									
60	0	3660	70	0	4593									

- 164 -

EN ESTA HOJA SE ENCUENTRAN:

-RELATIVE CLOCK

-ABSOLUTE CLOCK

-NUMERO DE TRANSACCIONES QUE PASARON POR CADA BLOQUE

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
M                                M
M          USER CHAINS          M
M                                M
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

USER CHAIN	TOTAL ENTRIES	AVERAGE TIME/TRANS	CURRENT CONTENTS	AVERAGE CONTENTS	MAXIMUM CONTENTS
WAIT	687	119.164	2	1.637	9

- 153 -

EN ESTA HOJA SE MUESTRA EL USO DEL ALMACENAMIENTO (DEL SPOOL).

XX
 M M M
 M QUEUES M
 M M M
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

QUEUE	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	PERCENT ZEROS	AVERAGE TIME/TRANS	AVERAGE TIME/TRANS	TABLE NUMBER	CURRENT CONTENTS
1	3	.237	118	24	20.3	100.618	126.308		
2	2	.044	21	1	4.7	107.142	112.500		
3	1	.004	2		.0	111.000	111.000		
4	2	.116	157	56	35.6	36.955	57.445		
5	3	.332	10	1	9.9	1664.899	1849.888		
6	1	.057	46	5	10.8	62.565	70.195		2
7	1	.046	5		.0	461.799	461.799		
8	2	.153	18	1	5.5	427.111	452.235		
9	1	.021	3		.0	360.333	360.333		
10	1	.012	2		.0	308.500	308.500		
11	2	.098	114	32	28.0	43.201	60.060		
12	1	.035	4	1	25.0	440.000	586.866		
13	3	.181	117	13	11.1	77.546	87.240		
14	2	.120	185	70	37.8	32.567	52.391		
15	2	.174	105	16	15.2	62.990	97.910		

AVERAGE TIME/TRANS = AVERAGE TIME/TRANS EXCLUDING ZERO ENTRIES

1
167

-TIEMPO PROMEDIO DE ESPERA POR CLASE.

-QUEUE, REPRESENTA LA CLASE.

-TOTAL ENTRIES, EL TOTAL DE ENTRADAS DE CADA CLASE AL SISTEMA.

-AVERAGE TIME/TRANS. EL PROMEDIO DE TIEMPO DE ESPERA POR CLASE.

TABLE 2
ENTRIES IN TABLE
110

	MEAN ARGUMENT 7.772	STANDARD DEVIATION 5.933	SUM OF ARGUMENTS 855.000	NON-WEIGHTED		
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
1	29	26.36	26.3	73.6	.128	-1.151
2	10	9.09	35.4	64.5	.257	-.972
3	0	.00	35.4	64.5	.385	-.804
4	16	14.54	49.9	50.0	.514	-.635
5	0	.00	49.9	50.0	.643	-.467
6	0	.00	49.9	50.0	.771	-.290
7	0	.00	49.9	50.0	.900	-.113
8	0	.00	49.9	50.0	1.029	.038
9	0	.00	49.9	50.0	1.157	.206
10	0	.00	49.9	50.0	1.286	.375
11	16	14.54	64.5	35.4	1.415	.543
12	0	.00	64.5	35.4	1.543	.712
13	0	.00	64.5	35.4	1.672	.880
14	19	17.27	81.8	18.1	1.801	1.049
15	20	18.18	100.0	.0	1.929	1.218

REMAINING FREQUENCIES ARE ALL ZERO

TABLE 3
ENTRIES IN TABLE
112

	MEAN ARGUMENT 8.991		STANDARD DEVIATION 4.359		SUM OF ARGUMENTS 1007.000		NON-WEIGHTED
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN	
1	0	.00	.0	100.0	.111	-1.853	
2	0	.00	.0	100.0	.222	-1.603	
3	2	1.78	1.7	98.2	.333	-1.374	
4	39	34.82	36.6	63.3	.444	-1.144	
5	6	5.35	41.9	58.0	.556	-.915	
6	0	.00	41.9	58.0	.667	-.686	
7	0	.00	41.9	58.0	.778	-.456	
8	0	.00	41.9	58.0	.889	-.227	
9	0	.00	41.9	58.0	1.000	.002	
10	2	1.78	43.7	56.2	1.112	.231	
11	29	25.89	69.6	30.3	1.223	.460	
12	0	.00	69.6	30.3	1.334	.690	
13	3	.00	69.6	30.3	1.445	.919	
14	34	30.35	100.0	.0	1.557	1.149	

REMAINING FREQUENCIES ARE ALL ZERO

TABLE 4
ENTRIES IN TABLE
302

	MEAN ARGUMENT 11.261	STANDARD DEVIATION 3.917	SUM OF ARGUMENTS 3407.000	NON-WEIGHTED		
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
1	1	.33	.3	99.6	.088	-2.624
2	1	.33	.6	99.3	.177	-2.368
3	0	.00	.6	99.3	.265	-2.113
4	39	12.91	13.5	86.4	.394	-1.858
5	0	.00	13.5	86.4	.443	-1.603
6	33	10.92	24.5	75.4	.531	-1.348
7	0	.00	24.5	75.4	.620	-1.092
8	0	.00	24.5	75.4	.709	-.837
9	0	.00	24.5	75.4	.797	-.582
10	0	.00	24.5	75.4	.886	-.327
11	45	14.90	39.4	60.5	.975	-.071
12	1	.33	39.7	60.2	1.063	.183
13	65	21.52	61.2	38.7	1.152	.438
14	57	18.87	80.1	19.8	1.240	.693
15	60	19.86	100.0	.0	1.329	.949

REMAINING FREQUENCIES ARE ALL ZERO

TABLE 5
ENTRIES IN TABLE
116

	MEAN ARGUMENT 9.551	STANDARD DEVIATION 5.550	SUM OF ARGUMENTS 1108.000	NON-WEIGHTED		
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
1	29	25.00	25.0	75.0	.104	-1.560
2	0	.00	25.0	75.0	.209	-1.360
3	0	.00	25.0	75.0	.314	-1.160
4	0	.00	25.0	75.0	.418	-1.000
5	0	.00	25.0	75.0	.523	-.820
6	0	.00	25.0	75.0	.628	-.639
7	5	4.31	29.1	70.6	.732	-.459
8	18	15.51	44.2	55.1	.837	-.279
9	0	.00	44.2	55.1	.942	-.099
10	0	.00	44.2	55.1	1.046	.080
11	0	.00	44.2	55.1	1.151	.260
12	0	.00	44.2	55.1	1.256	.441
13	22	18.96	63.7	36.2	1.361	.621
14	16	13.79	77.5	22.4	1.465	.801
15	26	22.41	100.0	.0	1.570	.981

REMAINING FREQUENCIES ARE ALL ZERO

TABLE 6
ENTRIES IN TABLE
149

		MEAN ARGUMENT 7.872	STANDARD DEVIATION 5.160	SUM OF ARGUMENTS 1173.000		NON-WEIGHTED
UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
1	34	22.81	22.8	77.1	.127	-1.331
2	0	.00	22.8	77.1	.254	-1.138
3	0	.00	22.8	77.1	.381	-.964
4	28	18.79	41.6	58.3	.508	-.750
5	0	.00	41.6	58.3	.635	-.556
6	12	8.05	49.6	50.3	.762	-.362
7	0	.00	49.6	50.3	.889	-.169
8	0	.00	49.6	50.3	1.016	.024
9	0	.00	49.6	50.3	1.143	.218
10	0	.00	49.6	50.3	1.270	.412
11	22	14.76	64.4	35.5	1.397	.606
12	0	.00	64.4	35.5	1.524	.799
13	29	19.46	83.8	16.1	1.651	.993
14	24	16.10	100.0	.0	1.778	1.187

REMAINING FREQUENCIES ARE ALL ZERO

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X                                     X
X           HALFORD MATRICES         X
X                                     X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

```

HALFORD MATRIX 1
ROW/COLUMN 1
1 12
2 2
3 10
4 13
5 13
6 6
END

```

	1	2	3	4	5	6
1	12	1	9	2	14	4
2	2	14	11	1	15	4
3	10	3	4	5	14	11
4	13	11	15	6	4	14
5	13	15	14	7	8	1
6	6	13	14	4	11	1

XXXX TOTAL RUN TIME (INCLUDING ASSEMBLY) = .04 MINUTES XXXXX

174

-MUESTRA LA ASIGNACION DE CLASE QUE SE HIZO EN LA PRESENTE SIMULACION.
 LOS CERQUEJONES SON LOS INICIADORES (6).
 LAS COLUMNAS PRESENTAN LA PRIORIDAD RELATIVA QUE TIENE CADA CLASE
 DENTRO DE CADA INICIADOR.
 POR EJ. EN EL RENGLON 2 (INIC. 2), LA CLASE 11 TIENE MAYOR PRIORIDAD
 QUE LA CLASE 1.

C A P I T U L O V I I I :

C O N C L U S I O N E S

CAPITULO VIII

CONCLUSIONES:

1.- Al llevar a cabo el estudio de la carga de trabajo actual en el sistema, se observa que dicha carga no es igual durante todo el día, ni durante toda la semana. Por lo tanto, se recomienda utilizar para el modelo de simulación, los días que está más cargado el sistema (lunes a viernes), y separar en dos o más periodos cada día dependiendo de la carga del sistema, esto es, por ejemplo que la mayoría de las compañías trabajan con un horario de 9 a 18 hrs. por lo que la carga de trabajo en este horario es más pesada que de las 19 a las 7 hrs. en donde por lo general sólo se corren algunos trabajos de producción.

En la aplicación de este método y en base a la recolección de eventos del Capítulo V, propongo tres periodos (definidos en dicho Capítulo).

Para hacer más eficiente la asignación, desecho el Periodo 3, ya que tiene una carga de trabajo mínima y cualquier asignación podría satisfacerla fácilmente. Después, busco la asignación que optimice a mi sistema en el periodo que tenga la carga de trabajo más pesada

CAPITULO VIII

(Período 2), y finalmente busco la asignación que haga más eficiente a mi sistema tanto en el Período 1 como en el Período 2, ya que estos representan realmente la carga de trabajo en el sistema.

2.- Al hacer la formulación del modelo, hay varios factores importantes que deben de tomarse en cuenta, tales como considerar los estados iniciales (eliminando los estados transientes), considerar un tiempo de simulación bastante grande (por lo menos tres veces el tiempo real), así como considerar todas las variables que ayudan en la descripción de la operación del sistema (tiempo entre llegadas, duración, etc).

3.- Se recomienda la utilización del paquete estadístico SAS para analizar la información recolectada por los sistemas SMF y RMF, ya que este nos entrega una serie de gráficas que facilitan el análisis de la carga actual en el sistema.

Se recomienda la utilización de una rutina de exit (como el RACF de IBM) o de algún procedimiento (como el RPF de ROSCOE) que le permita al sistema automatizar los Submit's de los Trabajos (Jobs), validándolos y generándoles la tarjeta de control antes de mandarlos a las colas de ejecución (en el JES). Esto es con el fin de

CAPITULO VIII

asegurar que los Job's Submitidos tengan la clase adecuada, ya que pudimos apreciar variaciones por ejemplo en la duracion presupuesta y la duraci3n real en la clase 'C'. Esto se debe a que los usuarios le asignan esa clase a Job's que no cumplen con los requisitos para dicha clase (duraci3n, uso de recursos, etc).

4.- Es recomendable utilizar el paquete de simulaci3n GPSS al momento de implementar el modelo del sistema, ya que dicho paquete cuenta con los requerimientos necesarios para facilitar la simulaci3n del sistema.

5.- Al obtener los resultados de las 3ltimas asignaciones, se pudo observar que ya no varían mucho los tiempos de espera. Haciendo un análisis de los tiempo de espera por clase obtenidos de este 3ltimo grupo de asignaciones, se puede considerar, el aplicar un 3ltimo criterio para la asignacion final. En nuestro caso, la asignaci3n final fue considerando que tuvieran menor tiempo de espera las clases con mayor frecuencia de llegadas.

Con los resultados obtenidos en la asignaci3n final de clases que se muestra en el Capítulo VII, se tiene un aumento notable en la producci3n, ya que al disminuir los tiempos de espera por clase, agiliza la ejecuci3n de

CAPITULO VIII

los jobs que llegan a nuestro sistema, permitiendonos además aumentar la carga de trabajo en el sistema.

6.- Haciendo una comparación de la asignación propuesta con la que esta actualmente asignada en el sistema, se observa que los tiempos de espera disminuyen notablemente, lo que nos conduce a un aumento significativo en la producción.

	ESTADO ACTUAL	PROPUESTO	DISMINUCION
P1	49,425	34,538	30 %
P2	118,208	81,819	30.7 %

lo que nos da una disminución en tiempo de espera de aproximadamente un 30%.

7.- Al aplicar esta metodología además de permitirnos mejorar la distribución de clases en los iniciadores, nos permite evaluar las cargas de trabajo en el sistema, predecir su estado a futuro y sugerir soluciones.

A P E N D I C E A :

INTRODUCCION A SAS .

APENDICE A

INTRODUCCION A SAS:

SATATISTICAL ANALISYS SYSTEM (SAS).

Es un lenguaje que se utiliza como herramienta para describir y analizar registros tanto del SMF (SYSTEM MANAGEMENT FACILITIES) como del RMF (RESOURCE MESAUREMENT FACILITIES).

Siempre que se está ejecutando SAS puede estar en uno de dos estados. Puede estar construyendo un set de datos SAS, o puede estar analizando un set de datos ya construído.

Si el programa comienza con un 'DATA', significa que empezará a construir un nuevo set de datos. Si comienza con 'PROC', significa que comenzará a analizar un set de datos.

Un módulo lógico de trabajo en SAS, consiste en el contenido desde el primer DATA o PROC hasta el siguiente DATA o PROC.

SAS interpreta todas las instrucciones sólo hasta encontrar el segundo DATA o PROC.

APENDICE A

Instrucciones SAS.

Todas las instrucciones SAS son de libre formato. Todas las columnas son revisadas (desde la 1 hasta la 80) hasta encontrar un delimitador de instrucción que en este caso es un punto y coma (;).

En caso de no existir un punto y coma se considera que la instrucción es continua.

SAS permite varias instrucciones en un sólo renglón y además se pueden insertar líneas en blanco sin ningún efecto.

Nombres y palabras reservadas, deben ser separadas por blancos, pero no se requiere con caracteres especiales y palabras reservadas o nombres.

SAS NAMES. Nombres de set de datos, archivos, variables etc. están limitados a 8 caracteres y deben comenzar con un caracter alfabético o con un guión.

Al igual que en lenguajes como el PL/1, el mismo nombre puede ser utilizado repetidamente en un programa SAS para diferentes propósitos.

El almacenamiento del set de datos SAS puede ser en

APENDICE A

BDAM en disco o en BSAM en cinta.

Cada set de datos SAS es una colección de observaciones que puede ser considerada como filas de una matriz. Cada observación es una colección de variables cuyo nombre es asignado por el usuario, y cada variable puede ser considerada como una columna.

Un data step, toma datos de entrada (los cuales pueden ser de un set de datos del OS en cinta o en disco, un set de datos previamente construido o una combinación de ambas) y contruir un nuevo set de datos SAS.

-PASO DATA.

Para introducir los datos en un set de datos SAS :

- a).- Se crea un set de datos SAS;
- b).- Se describen los datos para su lectura;
- c).- Se indica en donde se encuentran los datos.

La estructura de una instrucción en SAS es :

Instrucción SAS=Que hacer + Como hacerlo + ;

Entonces un PASO DATA esta compuesto por las instrucciones que crean un set de datos.

APENDICE A

-PASO PROC.

Para procesar o analizar un set de datos SAS se siguen los siguientes pasos :

- a).- Definir el nombre del procedimiento a ejecutar;
- b).- Especificar el nombre del set de datos a procesar;
- c).- Especificar si el set de datos va a ser procesado en subconjuntos;
- d).- Determinar las variables a procesar.

Un PASO PROC está compuesto por las instrucciones que procesan o analizan los set de datos SAS.

Los Procedimientos SAS son programas que leen el conjunto de datos, realizan varios cálculos, e imprimen los resultados.

Las instrucciones SAS para introducir datos en un set de datos (PASO DATA) son las siguientes :

- 1.- DATA
- 2.- INPUT

APENDICE A

3.- CARDS o INFILE.

El formato para DATA es entonces :

DATA + nombre del set de datos + ;

ejemplo :

DATA EJEMPLO;

El formato para INPUT es :

INPUT + nombre de la variable + formato + posición + ...;

ejemplo :

INPUT NOMBRE \$ I-I00;

INPUT VENTAS DEPTO \$ DIA;

El formato para CARDS es :

CARDS + líneas de datos;

APENDICE A

ejemplo

CARDS;

1980 30000 palas

1980 35000 martillos

;

El formato para INFILE es :

INFILE + nombre de definición de datos (DD);

ejemplo :

INFILE SMF; en donde SMF es un set de datos

INFILE TRANS;

Tenemos entonces el siguiente resumen para introducir datos en un set de datos SAS :

- 1.- Comenzar con una instrucción DATA;
- 2.- Describir el orden de los valores de los datos con una instrucción INPUT;
- 3.- Si los datos van a ser introducidos por una

APENDICE A

terminal, usar una instrucción 'CARDS' a continuación, seguida de las líneas de datos:

4.- Si los datos están en cinta o disco, colocar la instrucción 'INFILE' antes de la instrucción 'INPUT'.

Las instrucciones SAS para analizar un set de datos SAS (PASO PROC) son las siguientes :

- 1.- PROC
- 2.- DATA=
- 3.- BY
- 4.- VAR

El formato para la instrucción PROC es :

PROC + nombre del procedimiento + instrucciones adicionales

ejemplo :

PROC PRINT ; es un procedimiento para imprimir

PROC PLOT; es un procedimiento para graficar

APENDICE A

El formato para la instrucción 'DATA=' es :

DATA= nombre del set de datos SAS;

ejemplo :

```
PROC PRINT DATA=SALIDA;
```

```
PROC SORT DATA=MAESTRO;
```

El formato para la instrucción 'BY' es :

BY + variables que definen los subconjuntos;

ejemplo :

```
PROC PRINT;
```

```
BY EDAD SEXO;
```

```
PROC MEANS;
```

```
BY VUELOS;
```

APENDICE A

NOTA : Antes de usar una instrucción BY, se deben ordenar o clasificar los datos por los valores de las variables en la instrucción BY. Esto es ;

```
PROC SORT;  
.BY EDAD SEXO;
```

```
PROC SORT;  
BY VUELOS;
```

El formato para la instrucción 'VAR' es :

VAR + variables a procesar ;

ejemplo :

```
PROC PRINT;  
VAR EDAD ESTATURA;
```

```
PROC MEANS;  
VAR TARIFAS;
```

APENDICE A

Para adaptar los datos :

- 1.- Se crean nuevas variables;
- 2.- Se listan líneas de datos;
- 3.- Se pueden suprimir observaciones;
- 4.- Se puede cambiar el orden de ejecución.

Sus instrucciones en SAS serian :

- 1.- '=' (asignacion)
- 2.- LIST
- 3.- DELETE
- 4.- IF

El formato para la instrucción de asignación es :

nombre de la nueva variable = expresión

ejemplo :

```
PESOKG=PESOLB*.45;
```

```
Y=2;
```

APENDICE A

Los símbolos que se pueden usar en una instrucción de asignación son los siguientes :

SIMBOLO	OPERACION	EJEMPLO	SAS
**	EXPON.	$Y=X^2$	$Y=X**2$
*	MULT.	$A=B*C$	$A=B*C$
/	DIV	$G=H/I$	$G=H/I$
+	SUMA	$R=S+T$	$R=S+T$
-	RESTA	$U=U-X$	$U=U-X$

Los formatos para las instrucciones LIST y DELETE son:

LIST

DELETE

ejemplo :

```
IF SEXO='M' THEN DELETE;
```

El formato para la instrucción IF es :

AFENDICE A

IF condición THEN instrucción;

ejemplo :

IF ESTATURA GT 2 THEN DELETE;

Los operadores que se pueden usar en condiciones de IF son :

SIMB ABREV RELACION

<	LT	MENOR QUE
<=	LE	MENOR O IGUAL QUE
>	GT	MAYOR QUE
>=	GE	MAYOR O IGUAL QUE
=	EQ	IGUAL QUE

APENDICE A

Ejemplo de SAS

```
DATA TYPE434 (KEEP=JOB READTIME CPUTM);
LENGTH DEFAULT=4 READTIME @;
FORMAT READTIME DATETIME19.2 CONCODE HEX3.;
INFILE SMF STOPOVER LENGTH=LENGTH COL=C RECFM=VBS
LRCL=32756;
INPUT @2 ID PIB1.@3 SMFTIME SMFSTAMP8.
@23 READDATE PD4.@16 JOB $8. @;
IF ID=4 THEN GO TO TYPE4;
IF ID=4;
BEGIN=105;
LINK EXCPGET;
OUTPUT TYPE434;
X=.;
EXCPTOT=SUM(EXCP2314,EXCP3330,EXCP3350);
RIGHTBIT=MOD(BYTE,2);
LEFTBIT=FLOOR(BYTE/128);
...

DATA TYPE434 (KEEP=JOB REDATIME CPUTM);
```

-Data invoca el compilador de SAS y nombra el set de

APENDICE A

datos que será construido (TYPE434). Se pueden establecer unas opciones por el paréntesis: 'KEEP=' indica que sólo las variables JOB, RESDIME y CPUTM serán incluidas en este set de datos (TYPE434).

```
LENGTH DEFAULT=4 READTIME8;
```

-Las variables numéricas están almacenadas de punto flotante. La longitud de default de esas variables es de 8 (doble precisión). Se puede ahorrar espacio en disco cambiando este default a 4, ya que no es tan importante la exactitud excepto en el READTIME que tiene hasta décimas de segundo.

```
FORMAT READTIME DATETIME19.2 CONCODE HEX3.;
```

-Se pueden asociar formatos de salida para las variables al momento de construir un set de datos. SAS posee varios formatos y funciones para construir fechas, tiempos, etc. DATETIME19.2 convierte el valor de READTIME, el cual es el número de segundos y se imprime de la siguiente forma:

```
04JUL79:12:23:02:34
```

```
INFILE SMF STOPOVER LENGTH=LENGTH COL=C
```

APENDICE A

```
RECFM=VBS LRECL=32756;
```

-INFILE nombra el DD (archivo de entrada o DATA DEFINITION) que contiene los datos que serán leídos.

-STOPOVER. Permite leer más de un registro lógico con un sólo INPUT.

-LENGTH=LEN. Permite especificar al usuario un nombre a la variable (en este caso LEN), que contiene la longitud actual del registro lógico que está siendo procesado. LENGTH se necesita cuando se procesan algunos registros de SMF, ya que, conociendo la longitud el usuario puede identificar que parte del SD escribió ese registro.

-COL=C. Permite especificar un nombre variable (C), el cual contiene la posición actual del apuntador al registro de entrada.

-RECFM=VBS LRECL=32756 Los atributos del DCB del archivo de entrada pueden ser especificados por INFILE, DD o una etiqueta en cinta o disco.

```
INPUT @2 ID PIB1.@3 SMFTIME SMFSTAMP8.
```

```
@23 READDATE PD4.@16 JOB #8. @;
```

La instrucción INPUT extrae datos de los registros del

APENDICE A

DS basados en el formato de entrada especificados para cada variable, y se almacenan en variables SAS las cuales fueron creadas por las instrucción INPUT.

El símbolo '@', localiza al apuntador en la columna deseada.

La variable 'ID' describe el tipo de registro (PIB1, entero binario positivo de longitud uno).

En forma similar la variable SMFTIME se localiza en la tercera columna y con un formato especial de SMFTIME8 que convierte la fecha de 4 BYTES y la hora de 4 BYTES en un sólo valor de fecha tiempo en SAS. En la columna 23 con un formato empacado de longitud 4, se almacena en una variable de READDATE. La variable JOB es una cadena de caracteres de longitud 8. El último signo @ causa que la última instrucción INPUT recuerde esta entrada.

```
IF ID=4 THEN GO TO TYPE4;
```

Esta instrucción controla el flujo lógico del set de datos SAS. Las instrucciones lógicas pueden ser sencillas o altamente complejas con ayuda de operadores Booleanos (AND, OR, NOT).

En este caso en caso de cumplirse la instrucción

APENDICE A

(verdadera), se irá a la instrucción TYPE4.

Esta instrucción acepta expresiones tales como :

```
IF 70 <= ID <= 79 THEN LABEL 'RMF';
```

```
IF ID=4;
```

Esta es una forma corta de manejar el IF-THEN; si la expresión se cumple la siguiente instrucción se ejecuta, si no se cumple, este registro no se une al set de datos y el programa regresa a la siguiente instrucción de INPUT.

```
BEGIN=105;
```

```
LINK EXCPGET;
```

La instrucción LINK es una llamada a subrutina. Los argumentos para esta subrutina son asignados antes de LINK (BEGIN=105).

```
OUTPUT TYPE434;
```

Permite al usuario decirle a SAS explícitamente cuando deseamos agregar las observaciones actuales al set de datos que está siendo construido. En caso de que se estén creando

APENDICE A

varios set de datos, la instrucción OUTPUT nombra al set de datos al que se desea que se agreguen las observaciones obtenidas. En caso de no existir la instrucción OUTPUT, cada observación es añadida en forma automática al set de datos cuando la última instrucción es ejecutada, o al encontrarse la instrucción RETURN.

RETURN;

Tiene dos usos en SAS :

Cuando una instrucción de LINK es ejecutada, y se encuentra la instrucción RETURN, SAS regresa a la siguiente instrucción después de LINK.

En caso de no existir la instrucción LINK, la instrucción RETURN, causa que SAS regrese a tomar la siguiente instrucción.

GOTO;

Origina que SAS comience la ejecución en la etiqueta nombrada.

X=.;

APENDICE A

Valores perdidos. El concepto de valores perdidos de una variable en análisis de registros de SMF.

Este concepto diferencia el valor de cero con un valor no existente de una variable. Internamente el valor perdido (missing value) es el más pequeño valor negativo existente.

Para prevenir errores tanto estadísticos como aritméticos se pueden probar los valores de las variables de la siguiente forma :

```
IF X=. THEN MISS='YES';
```

```
RETAIN;
```

Al comenzar un programa SAS, todas las variables toman el valor de 'Missing values', y después comienza a asignar los valores a dichas variables. Por esto una variable no va a retener su valor de una observación a otra. Si se desea que una variable conserve su valor se nombra a dicha variable con una instrucción de RETAIN.

```
EXCPTOT=SUM(EXCP2314,EXCP3330,EXCP3350);
```

Una función de SUM ayuda a eliminar el problema de

APENDICE A

'valores perdidos' sumando los valores de las variables sin tomar en cuenta los que tienen 'valores perdidos'.

```
RIGHTBITE=MOD(BYTE,2);
```

La función MOD nos da el módulo del primer argumento entre el segundo, y asigna ese valor a la variable RIGHTBITE.

```
LEFTBITE=FLOOR(BYTE/128);
```

La función FLOOR nos da el entero más largo que sea menor o igual que el argumento. En este ejemplo LEFTBITE toma el valor de uno si BYTE vale de 127 a 256 y vale cero si vale menos.

Valores de Tiempo, Fecha y de Fecha-Tiempo.

Existen en SAS varias facilidades para leer y manejar fechas y tiempos.

Las Fechas se almacenan como el número de días transcurridos desde 1 enero de 1960.

APENDICE A

Para poder probar y comparar fechas y tiempos, estas se pueden convertir en valores SAS a través de unas funciones. En adición a esto en SMF, la mayoría de las fechas se tienen como días JULIANOS, en un campo decimal empacado.

Tenemos dos alternativas para una fecha :

```
INPUT DATE PD4.;
IF DATE=79034;
o
INPUT DATE PD4.;
SASDATE=DATEJUL (DATE);
IF SASDATE=MDY(02,03,1979);
```

La segunda opción es más fácil de trabajar.

Los valores de tiempo en SAS se almacenan internamente en segundos. Existen también funciones para convertirlos.

```
INPUT TIME HHMMSS.;
IF TIME=HMS(10,15,0) REPRESENTA LAS 10:15
```

Los valores de tiempo en SAS, son también almacenados como el número de segundos transcurridos desde enero 1 de 1960.

APENDICE A

Para manejar sólo una porción de un valor de fecha-tiempo :

```
DATE=DATEPART(DATETIME);
```

```
TIME=TIMEPART(DATETIME);
```

```
SET
```

La instrucción 'SET' es como la instrucción 'INPUT', excepto que la entrada proviene de un set de datos previamente construido.

Ejemplo :

```
DATA ALL;
```

```
SET DAILY WEEKLY MONTHLY;
```

El set de datos ALL contiene todas las observaciones de los set de datos : 'DAILY', seguida por todas las observaciones del set de datos 'WEEKLY', y después del set de datos 'MONTHLY'. Sin la instrucción BY, la instrucción SET concatena los sets de datos de izquierda a derecha.

A P E N D I C E B

INTRODUCCION A GPSS

APÉNDICE B

INTRODUCCION AL GPSS.

El GPSS es un lenguaje de simulación de propósito general para simular sistemas discretos.

Cuando deseamos simular un sistema, éste se describe como un diagrama de bloques en el 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 un bloque.

En general la forma que toma la descripción de un diagrama de bloques depende de la persona que dibuja el diagrama.

Para basar un lenguaje de programación en este método descriptivo, se debe dar un significado preciso a cada bloque.

El enfoque que se sigue en el GPSS es definir un conjunto de 43 tipos de bloques específicos, cada uno de

APENDICE B

los cuales representa una acción característica de los sistemas. El usuario del programa debe de dibujar un diagrama de bloques del sistema utilizando sólo estos tipos de bloques.

A cada tipo de bloque se le da un nombre que describe el tipo de acción y que se representa mediante un símbolo determinado. La figura B.1 muestra los tipos de símbolos que se utilizan para los tipos de bloques.

Las instrucciones de codificación para todos los tipos de bloques descritos, se han reunido en forma semejante en la tabla B.1. Cada tipo de bloque tiene una cantidad de campos de datos.

A través del sistema simulado se mueven las entidades que dependen de la naturaleza del sistema.

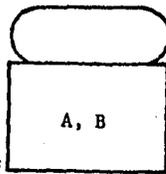
Las transacciones se crean en uno o mas bloques 'GENERATE' y se sacan de la simulación en bloques TERMINATE. Pueden haber muchas transacciones moviéndose simultáneamente a través del diagrama de bloques. Cada transacción se posiciona siempre en un bloque y la mayoría de los bloques pueden mantener simultáneamente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



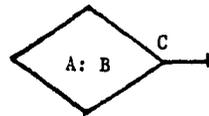
LIBERA



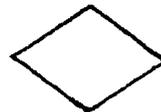
TERMINA



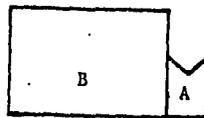
RESERVA VALOR



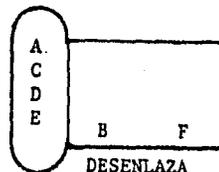
AGARRA



TRANSFIERE

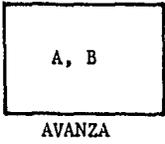


TABULA

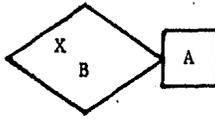


DESENLAZA

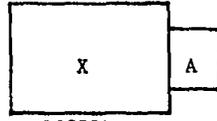
TABLA B 1



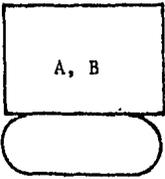
AVANZA



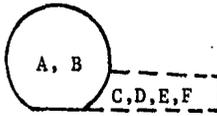
COMPUERTA



LOGICA



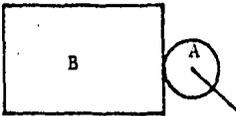
ASIGNA



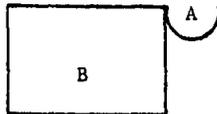
GENERA



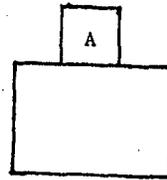
MARCA



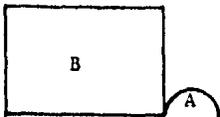
SALE



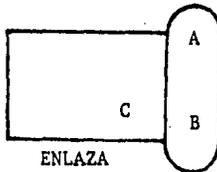
ABANDONA



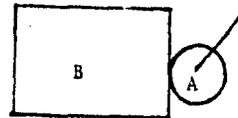
PRIORIDAD



ENTRA



ENLAZA



COLA

T A B L A B 2

TIPOS DE BLOQUES

OPERACION	A	B	C	D	E	F
ADVANCE	Media	Modificador				
ASSIGN	Param Núm. (+ -)	Fuente				
DEPART	Cola Núm.	(Unidades)				
ENTER	Almacenaje Núm.	(Unidades)				
GATE	Artículo Núm.	(Sigue Blo- que B)				
GENERATE	Media	Modificador	(Desplaza- miento)	(Cuenta)	(Prioridad)	(Params.)
LEAVE	Almacenaje Núm.	(Unidades)				
LINK	Cadena Núm.	Orden	(Sigiente bloque B)			
LOGIC R, S, I Mark	Switch Núm. (Param. Núm.)					
PRIORITY	Prioridad					
QUEUE	Cola Núm.	(Unidades)				
RELEASE	Facilidad Núm.					
SEVAVALUE	Reserva (+-) valor Núm.	SNA				
SEIZE	Núm. Facili- dad.					
TABULATE	Núm. Tabla	(Unidades)				
TERMINATE	(Unidades)					
TEST	Arg. 1	Arg. 2	(Sigiente bloque B)			
TRANSFER	Núm. Cadena	Siguiente bloque A	Siguiente bloque B			
UNLINK	Núm. Cadena	Siguiente bloque A	Cuenta	(Núm. Param)	(Arg.)	(Sig. Bloque B)

APENDICE B

del sistema.

Un diagrama de bloques en GPSS puede consistir en muchos bloques hasta un límite prescrito por el programa (generalmente se fija en 1000). Acada bloque se le da un número de identificación llamado 'localización', y por lo general el movimiento de las transacciones va de un bloque al que tiene la localización siguiente 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 de manera secuencial. A los bloques que se necesita identificar dentro de la programación de problemas (puntos a donde se hace una transferencia) se les da un nombre simbólico. El programa ensamblador asocia el nombre con la localización apropiada. Los nombres simbólicos de bloques y otras entidades del programa deben de ir desde tres hasta cinco caracteres no blancos de los cuales los tres primeros deben de ser letras.

TIEMPOS DE ACCION.

La hora del reloj se representa mediante un número entero y el intervalo el tiempo real corresponde a una

APENDICE B

unidad de tiempo que escoge el usuario del programa (seg, hora, día, etc).

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 de tiempo conocido como "tiempo de acción" para cada transacción conforme entra a un bloque de 'ADVANCE', y la transacción permanece en el bloque durante este intervalo simulado antes de intentar proseguir. El único otro 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 0) 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 un modificador para el bloque. Si el modificador es un

APENDICE B

unidad de tiempo que escoge el usuario del programa (seg, hora, día, etc).

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 de tiempo conocido como 'tiempo de acción' para cada transacción conforme entra a un bloque de 'ADVANCE', y la transacción permanece en el bloque durante este intervalo simulado antes de intentar proseguir. El único otro 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 0) 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 un modificador para el bloque. Si el modificador es un número positivo, el tiempo de acción es una variable aleatoria entera que se elige de la media del rango +/- modificador, con iguales probabilidades dadas a cada número en el rango. A veces esta distribución uniforme es

APENDICE B

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 en un bloque de '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.

SUCESION DE EVENTOS.

El problema lleva registros de cuando debe de moverse una transacción dentro del sistema. Avanza completando todos los movimientos programados a 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

APENDICE B

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

Normalmente, una transacción no pasa tiempo en bloques distintos a los de 'ADVANCE'. En consecuencia una vez que el programa empieza 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 de 'ADVANCE' con un tiempo de acción no cero.

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.

La tercera posibilidad es que la transacción entra 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 mas posible, vuelve su atención a cualesquiera otras transacciones que deban de moverse en el mismo instante.

Si están completos todos los movimientos, el programa avanza el reloj a la hora del evento mas inminente y repi-

APENDICE B

te el proceso de ejecutar los eventos.

ELECCION 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 los siguientes bloques A y B. El método que se utiliza para la elección se indica mediante un factor de selección en el campo A del bloque TRANSFER. Se pueden 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 se desea hacer 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 de $1-S$, y al siguiente bloque B es de 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

APENDICE B

movimiento es posible, y al siguiente bloque B si no lo es.

FACILIDADES Y ALMACENAJES.

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 permanentes para representar el equipo del sistema.

Se define una facilidad (SEIZE) como una entidad que puede utilizar una sola transacción a la vez. Se define un almacenaje (ENTER) como una entidad que pueden ocupar muchas transacciones a la vez, hasta cierto límite predeterminado (generalmente 300). Las entidades individuales se identifican mediante números; se utiliza una secuencia numeral distinta para cada tipo. El número 0 es ilegal para esta y otras entidades del GPSS. El usuario puede asignar los números en cualquier orden.

Algunos ejemplos de como podrían interpretarse las entidades del sistema en distintos sistemas son:

APENDICE B

TIPO DE SIST.	TRANSACC.	FACILITY	ALMACENAJE.
COMUNICACION	MENSAJE	CONMUTADOR	TRONCAL
TRANSPORTE	AUTO	CASETA	CARRETERA
PROC. DE DAT.	REGISTRO	PERFORADORA	MEMORIA DEL C.

Troncal es un cable que consiste en muchos alambres, cada uno de los cuales puede transportar varios mensajes simultaneamente.

RECOLECCION DE ESTADISTICAS.

Determinados tipos de bloques del GPSS se construyen para recabar estadísticas relativas al comportamiento del sistema, mas que para representar las acciones del mismo. Los bloques QUEUE, DEPART, MARK y TABULATE sirven para este propósito. Introducen otras dos entidades del programa GPSS, las colas y las tablas. Como las facilidades y los 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.

Quando no se satisfacen las condiciones para avanzar una transacción, se puede mantener a varias transacciones esperando en un bloque; el programa las mantiene en orden

APENDICE B

y cuando las condiciones son favorables se les permite avanzar de acuerdo con la prioridad y generalmente según la regla de primeros entrados, primeros atendidos. 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 esta en blanco, el cambio es unitario; en caso contrario se utiliza el valor del campo B. El programa mide las longitudes promedio y máxima de las colas, y de requerirse, la distribución del tiempo transcurrido en la cola.

TARJETAS DE CONTROL DEL PROGRAMA.

La primera tarjeta del lote de entrada de GPSS es una tarjeta de control con la palabra SIMULATE en el campo de operación. Sin esta tarjeta, el problema se ensambla pero no se ejecuta.

Es deseable poder detener y arrancar de nuevo una corrida de simulación y también repetir una corrida de simulación realizando cambios a algunos de los valores en el modelo. En consecuencia cuando termina una corrida de simulación en GPSS, el programa no destruye de inmediato

APENDICE B

el modelo utilizado, sino que busca mas tarjetas de entrada después de la tarjeta START, manteniendo el modelo exactamente como estaba a la terminación de la corrida.

Una tarjeta con la palabra aislada RESET en el campo de operación borra todas las estadísticas reunidas hasta el momento pero deja al sistema cargado con las transacciones. El propósito es inicializar el sistema para eliminar estados transientes.

Si a la tarjeta START, se le pone en el campo B las letras NP, el modelo de simulación corre el número de transacciones indicadas en el campo A y no imprime salida alguna. Se pone entonces a continuación una tarjeta con RESET y después otra con START con el campo A conteniendo algún valor (el número de transacciones que se desean correr) y así queda inicializado nuestro programa.

Otra tarjeta de control, CLEAR no solo borraría las estadísticas de la corrida anterior, sino también las transacciones del sistema, en cuyo caso la nueva corrida comienza la simulación desde el principio.

PRIORIDADES Y PARAMETROS.

Las transacciones tienen dos tipos de atributos que

APENDICE B

influyen la manera en que se procesan. Cada una tiene uno de 128 niveles de prioridad, que se indican mediante los números 0 a 127, en que el 0 tiene la menor prioridad. En cualquier punto del diagrama de bloques se puede cambiar la prioridad hacia arriba o hacia abajo mediante un bloque de prioridad. El bloque se codifica escribiendo la prioridad en el campo A del bloque. También es posible designar la prioridad al tiempo que se crea una transacción escribiéndola en el campo E del bloque 'GENERATE' que genera la transacción. Si se deja el campo en blanco se hace que la prioridad sea 0.

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

Todos los valores de parámetros son zeros al crearse una transacción. Se da un valor a un parámetro cuando una transacción entra a un bloque ASSIGN. Este puede utilizar como fuente del valor por asignar cualquiera de los atributos numéricos estandar (S N A). Se codifica especificando en el campo A y B el número de parámetro y

APENDICE B

el SNA que utilizará. Un bloque 'ASSIGN' puede sumar a, restar de o sustituir el valor de un parámetro.

Generalmente los parámetros representan las características de la entidad representada por la transacción. Así por ejemplo una transacción que representa un mensaje en un sistema de comunicaciones puede utilizar parámetros para representar su tipo, su longitud, o su destino.

ATRIBUTOS NUMERICOS ESTANDAR.

Los parámetros son artículos de datos que representan atributos de la transacción. Adicionalmente hay atributos de las otras entidades del sistema, tal como el número de transacciones en un almacenaje o la longitud de una cola, que quedan disponibles al usuario de un programa. A estos se les llama SNAs. Cada tipo de SNA se identifica mediante una clave de una o dos letras y un número. Por ejemplo, el contenido del almacenaje número 5 se denota mediante 'SS' y la longitud de la cola número 15 se denota por Q15. Por ser completos, los parámetros se incluyen dentro de los SNA, y se identifican mediante el símbolo Pn en el que n es el número de parámetro.

Se pueden hacer declaraciones variables de la siguien-

APENDICE E

te manera:

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

Se puede acceder al valor de esa variable para asignarse a algun parámetro o cualquier otra cosa así:

```
ASSIGN  P2,V5
```

Lo que hace es asignar al parámetro 2 de la transacción, el valor de la variable 5.

De la misma manera se pueden hacer asignaciones BOOLEANAS.

OPERACIONES DE CONJUNTOS.

Un requerimiento importante en un lenguaje de simulación es que pueda manejar conjuntos de entidades temporales con una propiedad común. Se cuenta con una cantidad de cadenas de usuarios y se coloca a una transacción en una cadena cuando entra a un bloque de LINK. El campo A lleva el número o nombre de la cadena y el campo B indica la disciplina de colas (PEPS o Fb).

APENDICE B

El bloque UNLINK permite que otro transacción fuera de la cadena saque transacciones de la cadena. El bloque llama a la cadena en el campo A, y en el campo B da la localización a donde deben ir las transacciones desenlazadas. El campo C indica cuantas transacciones deben de sacarse. Se puede hacer que la extracción dependa del valor de cualquier parámetro de las transacciones en la cadena. El campo D lleva el número de parámetro a examinar y el campo E lleva el valor que debe tener el parámetro para que se saque la transacción.

G L O S A R I O .

- Address Space.- Area que ocupa un Job en memoria principal. Area de programas y datos que pueden ser accedidos por un proceso.
- Batch Processing.- Proceso de un Job individual que no permite que interactue con un usuario una vez que el programa a sido leido por la computadora.
- Catalog.- Es una tabla que contiene informacion descriptiva para cada set de datos almacenado en un sistema. Meter informacion en dicha tabla.
- CPU.- Procesador que manipula y realiza operaciones sobre datos en memoria. Tambien ejecuta instrucciones para controlar a otros procesadores.
- Data Set.- Es un archivo de registros o datos organizados de manera logica para ser procesados por un programa de computadora (informacion que fue o sera procesada).
- DASD.- Direct Acces Storage Device. Dispositivo de almacenamiento de acceso directo (discos, etc).

- JCL.- Job Control Language. Identifica al programa que va a ejecutarse, su prioridad, y los recursos del sistema que va a necesitar para su ejecucion (dispositivos E/S, volúmenes, sets de datos, etc).

- JOB.- Es la coleccion de trabajo que queremos que realice el sistema de computo. Consiste en uno o mas programas que seran ejecutados en secuencia logica y los sets de datos que utiliza para entrada y salida (Job Step).

- Job Class.- Es un programa del sistema operativo que selecciona el Job que se va a ejecutar en base a normas de prioridad y capacidad establecidas.

- Job Step.- Es cada programa que sera ejecutado durante el proceso.

- Memoria.- Almacen de instrucciones y datos.

- Multiproceso.- Atiende a varios usuarios simultaneamente (tiene varios procesadores).

Multiprogramacion.- Sistema que puede tener a varios procesos en estado de ejecucion (es cuando los calculos han comenzado pero no han terminado)

- On-Line Operation.- Una operacion que se realiza

bajo el control del CPU. Designacion de un sistema o equipo que esta bajo el control del CPU (terminal, disk drive, printer). Es una operacion en la cual los datos que entran por la terminal son inmediatamente procesados por el CPU para alterar el contenido de los registros contenidos en disco.

- Procesador.- Dispositivo capaz de interpretar instrucciones y de ejecutarlas.

- Procesador de E/S.- Procesador disenado para controlar los dispositivos de E/S y de movimiento de datos entre la memoria y los dispositivos.

- Programa.- Secuencia de instrucciones.

- Software.- Coleccion de programas y datos usados para realizar un Job.

- Spooling.- Procedimiento mediante el cual se guarda en un almacen temporal los programas y sus caracteristicas (dispositivos, prioridad, etc), hasta que pueden ser ejecutados.

- SWA.- Scheduler Work Area. Secuenciador de area de trabajo.

- Swapping.- Movimientos de pages o slots entre memoria virtual y memoria auxiliar.

- Time Sharing.- Es un sistema en el cual la rapida alternacion en la ejecucion de programas, permite que dos o mas jobs sean procesados de manera que parecen simultaneos.

- Throughput.- El total de proceso ejecutado en un lapso de tiempo dado por el sistema.

- Unidad de control.- Circuitaria de control comun a varios dispositivos de E/S.

B I B L I O G R A F I A

- METODOS ESTADISTICOS
Herbert Arkin Raymond R. Colton C.E.C.S.A.

- SIMULACION UN ENFOQUE PRACTICO
Raúl Coss Bv. LIMUSA

- INTRODUCCIÓN A LA INVESTIGACION
DE OPERACIONES
Hillier, Lieberman MC GRAW HILL

- OPERATING SYSTEMS
John J. Donovan MC GRAW HILL

- MANUALES DE I.B.M.
OS/MVS PLANNING GUIDE
OS/MVS ARCHITECTURE DATA
FACILITY PRODUCT
OS/MVS ARCHITECTURE SYSTEM
DATA ADMINISTRATION

OS/MVS ARCHITECTURE DATA
ADMINISTRATION GUIDE
OS/MVS ARCHITECTURE TSO
TERMINAL USER'S GUIDE

- PROBABILITY, STATISTICS, AND
QUEUEING THEORY
Allen ACADEMIC PRESS

- TEORIA DE LA PROBABILIDAD
Iván Obregón LIMUSA

- MERRILL'S GUIDE TO C.P.E.
Merril, Ph. SAS CIRCLE

- SIMULACION DE SISTEMAS
Geothy Gordon DIANA