

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

MANEJO DE ALGUNOS PAQUETES DE LA COMPUTADORA
BURROUGHS B-6700 PARA INVESTIGACION DE
OPERACIONES Y ESTADISTICA

T E S I S

Que para obtener el título de:

A C T U A R I O

Presenta:

JUAN MANUEL PADILLA MIRANDA

México, D.F.

1982



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

TESIS CON FALLA DE ORIGEN

I N D I C E

INTRODUCCION	1
CAPITULO I INTRODUCCION A LA COMPUTADORA	
BURROUGHS B-6700	4
1.0 HISTORIA DE LA COMPUTACION	5
2.0 ESTRUCTURA DE UN COMPUTADOR	10
3.0 FORMAS DE TENER ACCESO A LA COMPUTADORA BURROUGHS B-6700 ...	17
4.0 INTRODUCCION AL EDITOR CANDE ...	24
5.0 INTRODUCCION A WFL	35
CAPITULO II MANEJO DEL PAQUETE TEMPO/MPS/ALL EN LA SOLUCION DE PROBLEMAS DE INVESTIGACION DE OPERACIONES ...	39
1.0 INTRODUCCION	40
2.0 MARCO TEORICO DEL ALGORITMO DEL METODO SIMPLEX	44
3.0 ALGORITMO DEL METODO SIMPLEX REVISADO	48
4.0 ANALISIS DE SENSIBILIDAD	52
5.0 EL DUAL DE UN PROBLEMA DE PROGRAMACION LINEAL	57
6.0 INTRODUCCION A TEMPO/MPS/ALL ...	59
7.0 EJEMPLOS DE APLICACION	77
8.0 LIMITACIONES	125

CAPITULO III	MANEJO DEL PAQUETE TEMPO/NETWORK EN LA SOLUCION DE PROBLEMAS DE REDES	126
1.0	INTRODUCCION A TEMPO/NETWORK ...	127
2.0	EJEMPLO DE APLICACION	137
3.0	LIMITACIONES	149
CAPITULO IV	MANEJO DEL PAQUETE BASIS/BASIS EN LA SOLUCION DE PROBLEMAS DE ESTADISTICA	150
1.0	INTRODUCCION A BASIS/BASIS	151
2.0	EJEMPLOS DE APLICACION	162
3.0	LIMITACIONES	203
CAPITULO V	MANEJO DEL PAQUETE SPSS EN LA SOLUCION DE PROBLEMAS DE ESTADISTICA	204
1.0	INTRODUCCION A SPSS	205
2.0	EJEMPLOS DE APLICACION	213
3.0	LIMITACIONES	236
CONCLUSIONES	237
REFERENCIAS	238

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

Desde el inicio de la historia escrita , los avances científicos y culturales han dependido de el uso de los símbolos La historia de la civilización puede considerarse como la historia del uso cada vez más sofisticado , que el hombre hace de los símbolos . Por ejemplo el hombre primitivo descubrió que las ideas se desarrollan y se comunican mejor por medio del lenguaje hablado y escrito , es decir con el uso de símbolos que representan las imágenes mentales , en cuanto avanza el conocimiento en cualquier área , los símbolos utilizados se hacen cada vez más abstractos .

Cuando los conceptos a que se refieren los símbolos son esencialmente no-cuantitativos , la nomenclatura y su relación puede analizarse en el marco de la lógica , cuando la nomenclatura representa conceptos cuantitativos la matemática es útil y de hecho , indispensable para analizar su relación . En la matemática pura (abstracta) las definiciones , axiomas , teoremas y supuestos se establecen con precisión en forma simbólica y el análisis se hace por deducción para demostrar los resultados obtenidos y determinar conclusiones .

La matemática aplicada difiere de la matemática pura en un aspecto muy importante , en la matemática pura los símbolos representan conceptos abstractos cuyas propiedades se fijan por definición , mientras que en la matemática aplicada muchos símbolos representan variables que se observan en el mundo real; las propiedades de estas variables deben determinarse por observación , no por definición abstracta y posteriormente establecerse en forma matemática . Además en matemáticas aplicadas puede determinarse la precisión empírica de las deducciones por lo tanto el análisis matemático aplicado se basa en definiciones determinadas empíricamente e hipótesis por medio de las cuales se obtienen por deducción conclusiones verificables empíricamente .

Los análisis matemáticos , puro y aplicado difieren -- solamente en cuanto al aspecto empírico de las definiciones y -- supuestos de las conclusiones no en cuanto a los métodos de deducción . Gran parte de los problemas de estadística e investigación de operaciones son indudablemente matemáticos en su naturaleza , por lo que las matemáticas proporcionan una estructura sistemática lógica dentro de la cual pueden estudiarse las relaciones cuantitativas . Cuando las variables de decisión se representan con símbolos y sus propiedades se establecen en forma matemática , la matemática suministra las técnicas para analizar relaciones entre los símbolos y por lo tanto entre las variables que ellos representan .

Gran parte del análisis de los problemas de estadística e investigación de operaciones es entonces análisis aplicado en el análisis de los problemas que se presentan en estadística e investigación de operaciones , como ocurre en general en la -- matemática aplicada , las deducciones obtenidas se interpretan y se evalúan empíricamente , a este respecto debe anotarse que -- si las deducciones que siguen a un conjunto de definiciones y -- supuestos no son correctas con relación a la observación empírica, el análisis matemático (si se realiza correctamente) , no es responsable de ello y la dificultad debe hallarse en las definiciones o hipótesis , las matemáticas capacitan para ser preciso al definir variables , para plantear claramente hipótesis hechas para ser lógico en el desarrollo del análisis y para considerar un número de variables necesario sin embargo no evita omisiones o definiciones empíricamente incorrectas de variables de decisión, el análisis matemático toma definiciones y supuestos tal -- como se dan y obtiene conclusiones que se deducen lógicamente de ellas , así el análisis matemático es lógico no empírico , por lo mismo si se realiza correctamente el análisis matemático pero las conclusiones son empíricamente incorrectas , deberán revisarse definiciones y suposiciones en cuanto a exactitud y suficiencia.

ciencia , por suministrar una estructura sistemática para deducir conclusiones empíricamente verificables , el análisis matemático ayuda a determinar precisión de definiciones e hipótesis , si las conclusiones son insostenibles deberán revisarse las definiciones e hipótesis .

C A P I T U L O I

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

A L A
C O M P U T A D O R A

B U R R O U G H S

B - 6 7 0 0

H I S T O R I A D E L A C O M P U T A C I O N

1.1 LA MAQUINA DIFERENCIAL . En 1812 Charles Babbage con
cibió una maquina , la que llamo el "difference engine" , la -
cual era un calculador que podía tabular el resultado de opera-
ciones matemáticas elementales . Su uso era limitado pero ño -
asi el principio que habia determinado su invención . En 1834
Babbage perfiló más sus ideas , descubriendo la llamada ---
Máquina Analítica , dispositivo capaz de almacenar información
numérica en una pequeña memoria , efectuar una secuencia de --
cálculos complejos siguiendo instrucciones contenidas en ---
tarjetas perforadas , imprimir los resultados y repetir incan-
sablemente ciclos enteros de operaciones preestablecidas . La
Máquina Analítica incluía en su repertorio de funciones dos --
posibilidades revolucionarias . Podía comparar datos y orien-
tarse hacia diferentes secuencias de instrucciones según el --
resultado de la comparación y podía tomar en cuenta el resulta-
do de una instrucción para modificar otras instrucciones , ---
cambiando así su propio programa .

Sin embargo , el magnifico proyecto de Babbage no --
pudo llevarse a efecto por falta del suficiente apoyo financie-
ro y de la tecnología apropiada . Hacia 1880 , Herman Hollerith
encontró el modo de aplicar el principio de la tarjeta perfora-
da a un material carente de conductividad eléctrica , de manera
que, almacenando información numérica en el , por medio de ---
perforaciones , podía recuperarla utilizando una corriente --
eléctrica . El invento de Hollerith probó su utilidad en las -
tabulaciones estadísticas del censo norteamericano de 1880 .
Poco a poco surgieron tabuladoras , clasificadoras e impresoras
mecánicas que podían procesar grandes volúmenes de información
con gran economía de tiempo y esfuerzo . Al primer computador
le correspondería la coordinación y sistematización interna de
estas operaciones , a la manera de un centro orgánico superior.

1.2 EL PRIMER COMPUTADOR .En 1945 , J. Presper Eckerrt y John W. Mauchly mostraron en operación un pesado aparato de 30 toneladas , que ocupaba unos 40 metros cuadrados de superficie diseñado por ellos y bautizado como ENIAC (Electronic Numerical Integrator and Calculator) . Programado por medio de bancos de conmutadores y conexiones alámbricas , el ENIAC no sería considerado hoy como un computador electrónico ; le faltaba memoria para guardar las instrucciones del programa . Sin embargo , mucho más que un simple calculador electrónico, aunque menos que un verdadero computador , ENIAC es considerado como el computador de la generación cero .

1.3 GENERACIONES DE COMPUTADORES . El concepto de generación no tiene un significado del todo claro . Para muchos , representa un índice de evolución de los computadores dado por la relación costo/rendimiento , ("ejecución") .

Cada vez que se produce un cambio apreciable en esa relación , esto es cuando un aparato ofrece funciones claramente más sofisticadas al mismo o más bajo costo , el especialista diría que se ha cruzado el umbral de una nueva generación .

El computador de la primera generación , dotado de un programa almacenable en la memoria , surgió tras el estudio , hoy clasico de Jhon Von Neumann intitulado "Preliminary discussion of the logical design of an electronic computing instrument" . En Abril de 1951 , el UNIVAC 1 de la Remington Rand (Hoy Sperry Rand) empezo a funcionar haciendo millones de cálculos por hora . De grandes dimensiones , el UNIVAC 1 utilizaba bulbos , consumia mucha energía y despedía excesivo calor . Para permitirle enfriarse , era necesario suspender operaciones cada cierto tiempo .

Entre 1959 y 1965 aparecieron los cerebros electrónicos de la segunda generación . Estos mejoraban , unas diez veces , el costo/rendimiento de los computadores de la primera generación, lo que hizo posible el cambio de unos a otros fue un pequeño objeto revolucionario inventado en 1948 en los laboratorios de la Bell Telephone por J. Bardeen , W Brattain y W. Shockley ; llamado el transistor , mientras los bulbos

requerían mucho espacio , cuidado y energía , el transistor -- permitía reducir las dimensiones del computador , multiplicando simultáneamente su capacidad y grado de confiabilidad .

En 1965 salían al mercado los computadores de la -- tercera generación , los cuales se caracterizarón por la ---- miniaturización de componentes , la estandarización de los --- transistores y el montaje de circuitos integrados sobre diagramas impresos en cajas de plástico , lo que representaba adelantos tecnológicos y que permitía otra mejora en el índice ---- costo/rendimiento .

El salto a la cuarta generación se produjo poco --- despues con el semiconductor y los circuitos integrados de -- mayor miniaturización . Donde un semiconductor es un circuito o componente de circuito construido con materiales de conductividad eléctrica inferior a los llamados buenos conductores y - superior a los no conductores . Los semiconductores han encontrado diferentes aplicaciones en el desarrollo tecnológico del computador , particularmente en los dispositivos de memoria. En estos la velocidad de conducción es decisiva pues permite - operar conmutaciones más rápidas ; También es necesario --- disponer de líneas de conexión más cortas y mayor densidad de almacenamiento . El semiconductor reveló ser muy útil en cada una de estas áreas .

1.4 C A R A C T E R I S T I C A S G E N E R A L E S D E U N C O M P U T A D O R D I G I T A L

BONDADDES DE UN COMPUTADOR . Un fenómeno curioso que se ha producido a lo largo de la rápida evolución del computador en sus aspectos tecnológicos , es que el aumento en ---- sofisticación , velocidad y confiabilidad ha sido acompañada - de un abaratamiento general del producto , ya que en la actualidad algunas de las microcomputadoras son capaces de hacer lo mismo que se hacía con computadores de hace unas decadas y el costo de un microcomputador en la actualidad es relativamente "bajo" .

La precisión , eficiencia y economía de recursos que han acompañado sus diversas aplicaciones por su flexibilidad

de programación son beneficios obtenidos en base a cuatro --- atributos del computador .

- 1/ Su gran velocidad de procesamiento .
- 2/ Su importante capacidad de almacenamiento .
- 3/ La accesibilidad de la información .
- 4/ Capacidad de transferencia de dispositivos perifericos a la unidad central . (f)

VELOCIDAD DE PROCESAMIENTO .- Mientras una máquina no electrónica realiza operaciones al cabo de un segundo , un -- computador moderno de alta velocidad opera las mismas operaciones en menos de un microsegundo (millonesimo de segundo) . Esta velocidad podra apreciarse mejor haciendo notar que el -- parpadeo del ojo humano toma una decima de segundo , es decir 100 000 microsegundos . Ahora bien considerando que solo se -- lleva a cabo una operación por segundo en una maquina no ---- electrónica y la misma operación se realiza en un microsegundo en un computador de alta velocidad , entonces en los 100 000 largos microsegundos que dura un abrir y cerrar de ojos un -- computador de alta velocidad puede ejecutar 1 millón de ---- instrucciones.

CAPACIDAD DE ALMACENAMIENTO .- La capacidad de memoria y la flexibilidad de expansión son factores decisivos en la -- estructura del computador . Este puede utilizar varios dispositivos de almacenamiento; Tarjetas perforadas , cintas magneticas discos magneticos y otros . En vista de su costo relativamente elevado la memoria principal suele mantenerse dentro de proporciones muy especificas . Su capacidad puede apreciarse a través de la siguiente comparación . Escrita a máquina , a doble -- espacio , una página puede contener unas 200 palabras , es decir aproximadamente 1000 caracteres . Una archivadora de tamaño -- normal puede almacenar en cada una de sus 5 gavetas , 30 páginas por centimetro , esto es , unos 30000 caracteres . En una gaveta de 60 cm. cabran 1800 000 caracteres y en la archivadora 9000000. Imaginando un banco de 80 archivadoras el total almacenable sería

(f) más tarde se hablara de estos componentes
(Cuando se vea la estructura de un computador)

de unos 720 000 000 de caracteres . En una computadora un aparato sencillo de almacenamiento , consistente en 8 discos que trabajan en condiciones normales de control se puede almacenar unos 800 000 000 de caracteres . Ahora bien , un computador puede disponer de varias unidades periféricas de este tipo .

ACCESIBILIDAD DE LA INFORMACION .- Cuando la organización de la información es apropiada , cualquier segmento de los 800 millones de caracteres es accesible en cuestión de --- microsegundos . Eso quiere decir que cualquier palabra de --- cualquier página de las 80 archivadoras puede ser recuperada de la memoria del computador en menos tiempo de lo que dura un abrir y cerrar de ojos . Tratándose de la memoria principal , el tiempo de extracción es más breve todavía ; el mismo dato de la misma página puede ser localizado en unos cuantos --- nanosegundos . [E]

CAPACIDAD DE TRANSFERENCIA DE DISPOSITIVOS PERIFERICOS A LA UNIDAD CENTRAL .- Un millón de caracteres presentado al computador bajo forma de tarjetas perforadas puede ser --- leído en una media hora . Si se le presenta , en cambio , la información bajo forma de una cinta magnética , la lectura no durara más que unos segundos . Si , en fin , se utilizan discos magneticos ultrarrápidos , el millón de caracteres puede ---- transferirse al area principal de almacenamiento en breves -- segundos . Una vez en la memoria principal la velocidad de - transferencia es tal que el millón de caracteres puede ser -- procesado en casi un segundo . Esta rapidez queda limitada , - sin embargo por las velocidades relativamente bajas que introduce el usuario cuando manipula el dispositivo de entrada o las --- terminales de acceso .

[E] El tiempo necesario para almacenar o traer una palabra desde un registro en memoria principal esta en el rango de .0000001 segundos (100 nanosegundos) hasta .00001 segundos (10 microsegundos = 10000 nanosegundos) , la rapidez y los diversos registros de la maquina, depende del tamaño y precio de la maquina .

(En algunos e casos el rango es de 10 nanosegundos a un microsegundo)

2 ESTRUCTURA DE UN COMPUTADOR

Un computador es una máquina cuya función es ---- procesar datos . En un computador se evita la presencia del -- hombre en el proceso del cálculo (este solo participa cuando - suministra datos y recibe la información) y efectua sus ---- operaciones a una velocidad de impulsos electronicos , anulán- do la presencia del hombre en todp el proceso de computo .

Ahora bien se podria decir que existen dos tipos de computadores , que son los siguientes :

1/ Digitales .- Los cuales representan sus datos mediante una serie finita de digitos .

2/ Analógicos .- En estos la información recibida de los captadores puede utilizarse , en la mayoría de los casos , sin una previa conversión de la misma , de la forma analógica a la digital (numérica) . El término de computador analógico se -- aplica a la clase de maquinas que utilizán señales electronicas niveles mecanicos u otros medios similares para simular el --- comportamiento de otros sistemas .

En lo sucesivo convendremos en que al hablar de un - computador implicara hacerlo de un computador digital .

Un computador esta constituido por los siguientes -- cuatro componentes :

- 1/ Memoria principal (MM) .
- 2/ Unidad central de proceso (CPU) .
- 3/ Unidad de aritmética y lógica (ALU) .
- 4/ Unidad de entrada/salida (I/O) .

En la fig. A , se muestrán las relaciones entre los cuatro componentes .

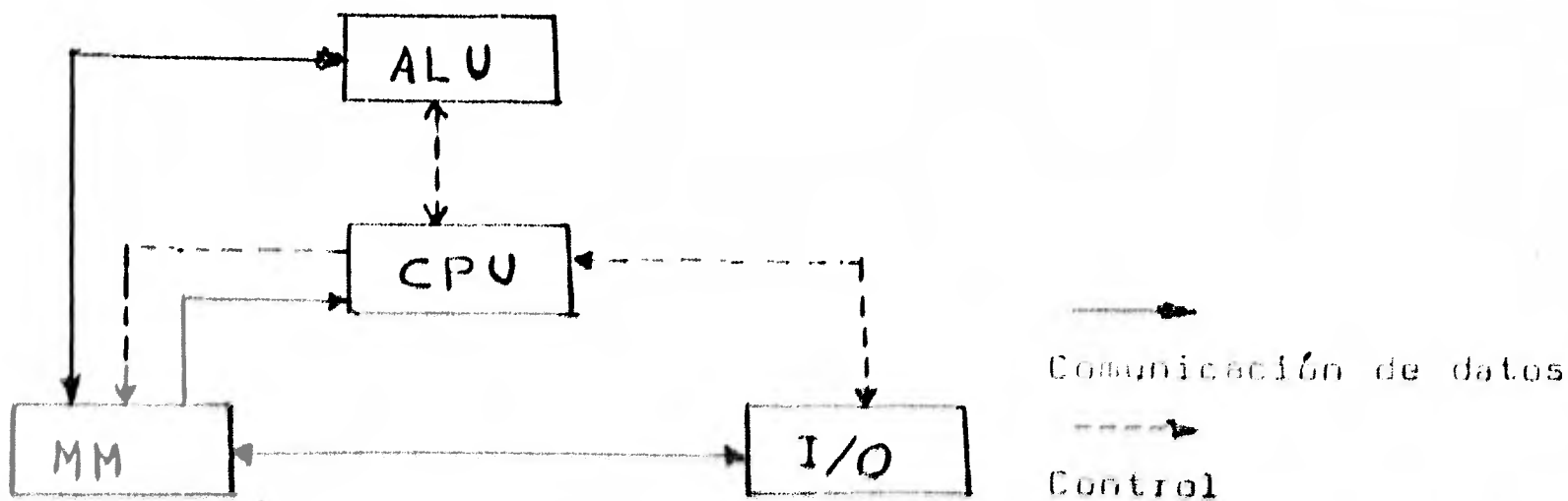


fig. A. Interacción de los componentes de una computadora .

Memoria principal .- Esta parte del computador esta comprendida de un conjunto de localidades , cada una de las -- cuales es capaz de almacenar un dato , los datos que se almacenán pueden representar valores de datos o instrucciones en --- lenguaje de maquina . Ninguna distinción se hace entre las --- localidades que contienen datos de las que contienen instrucciones . Cada localidad en la memoria principal de la computadora esta asignada a una dirección unica , esa dirección se utiliza para referenciar datos y efectuar transferencias de control - entre las instrucciones .

Unidad central de proceso .- La función elemental del CPU es la de secuenciar a través de programas y direcciones la ejecución de cada una de sus direcciones .

Unidad de aritmética y lógica .- La Unidad de aritmética y lógica es la parte de una computadora que realiza las - operaciones aritméticas y lógicas . Esta unidad normalmente -- contiene dispositivos fisicos , para realizar una aritmética - de punto fijo o punto flotante y la comparación de valores . El orden en que lleva al cabo sus cálculos y almacena los ---- resultados obtenidos , el ALU es utilizando una o más localidades de almacenamiento llamados registros . Estos són utilizados para almacenar operandos y resultados de operación .

Unidad de entrada/salida .- La función de la unidad de entrada/salida (I/O) es la de establecer comunicación entre el usuario y la computadora .

Los medios a través de los cuales se ejecután las -- operaciones en una computadora són los siguientes :

1/ Hardware .- Que es lo que podemos entender como el --- circuito físico (El equipo físico) .

2/ Software .- Que es un conjunto de programas manuales y posibilidades de un computador . ■■

■■ más tarde se hablara de lo que es un programa y se le definira .

La computadora para trabajar requiere recabar ordenes que generalmente son dadas a través de archivos de datos , por lo que trabaja con archivos de datos de diferentes tipos , esto es archivos de datos de instrucciones , llamados programas y - que son hechos en distintos lenguajes , ya que asi cómo nosotros nos comunicamos en diferentes idiomas , las computadoras entienden también un número limitado de lenguajes , llamados lenguajes de programación . Ejemplo de ellos són : Algol , Fortrán , Pascal Cobol y otros . El otro tipo de archivos es de tipo datos , esto es , archivos con datos que serán accesados por medio de un -- programa .

Para darle ordenes a la máquina se puede optar por - dos caminos :

- 1/ Programar .
- 2/ Usar programas ya hechos (Paquetes) .

Si se opta por el camino 1/ se tendrán las siguientes necesidades ; conocer la sintaxis del lenguaje a usar , conocer su semántica , tener la suficiente experiencia para elaborar -- programas que "hagan" lo que se pide y que no abusén de la --- computadora ; es decir que optimicen más recursos de espacio y tiempo , y que eviten en lo posible errores de redondeo . Visto desde otro enfoque , esto requiere de un tiempo aproximado de - un año de experiencia en programación .

Si se opta por el camino 2/ se va a necesitar conocer la forma en que los programas aceptán los datos , lo cual puede resultar 20 veces más rápido que aprender un lenguaje .

DISCUSION DE LAS OPCIONES

- Los paquetes son programas de propósito específico , por lo - que no son tan flexibles (No se puede esperar que un paquete de estadística resuelva problemas de contabilidad) .
- Un lenguaje permite resolver problemas no cubiertos en un -- paquete .
- La elaboración de programas es más costosa que la utilización de un paquete .
- Muchos errores en un programa no son detectables facilmente .

- La optimización de recursos por parte de un programa elaborado por nosotros requiere de un mayor tiempo de programación (Evitar errores de redondeo y uso excesivo de memoria y proceso).
- Los paquetes han sido generalmente hechos tratando de optimizar los recursos de la computadora y evitando errores de redondeo .
- Los paquetes normalmente han sido probados para que no fallen ante casos poco comunes .
- Los programas normalmente tienen que pasar por una o más etapas de prueba , que permitén "confiar" en que no fallarán .

Por lo que sería conveniente para aquellas personas interesadas en Estadística e Investigación de operaciones y que no disponen del tiempo necesario para aprender a programar , que conozcan los paquetes que existen y usarlos . De acuerdo a las dificultades a que se enfrenta el usuario de una computadora a surgido el concepto de distancia usuario computador , lo cual se analiza a continuación .

DISTANCIA ENTRE USUARIO Y COMPUTADOR

Esto se puede ver en cuatro fases , de acuerdo a las dificultades a las que se enfrenta el usuario con la computadora para resolver sus problemas .

1/ La primer fase, es en la que el usuario está muy ligado con la maquina , al trabajar con la computadora da las instrucciones en lenguaje de maquina , es decir en binario (las direcciones son numericas y su uso por parte del usuario es muy complicado)

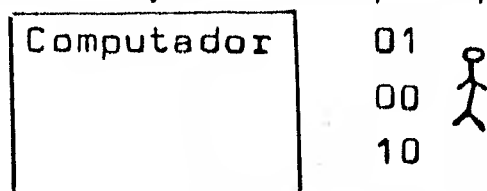


fig. a.1.1 El usuario da instrucciones en lenguaje de maquina .

2/ La segunda fase , es en la que el usuario empieza a trabajar en ensamblador (Las instrucciones son simbolicas). En esta fase el lenguaje ensamblador lo que hace es una conversión de lenguaje simbolico a binario ; el ensamblador definio una cierta sintaxis elemental (Etiquetas , instrucciones , operandos); ahora las instrucciones son codigos simbólicos , las direcciones son también etiquetas simbólicas , se les utiliza cómo un traductor para --

convertir símbolos en códigos numéricos (uno a uno) ; para la conversión de lenguaje simbólico a numérico, utiliza una tabla de código simbólico. Entre los lenguajes ensambladores se --- consideran varios tipos, cómo lo són, ensambladores de paso y medio, de dos pasos y de varios pasos. Un ensamblador de paso y medio, en el primer paso convierte instrucciones y registra las etiquetas, y en el medio paso siguiente si aparece un operando que no ha aparecido cómo etiqueta lo recuerda para convertirlo al valor numérico correspondiente cuando la etiqueta aparezca. Un ensamblador de dos pasos hace lo siguiente ; en el primer -- paso barre etiquetas y les asigna su valor numérico, en el --- segundo paso convierte las instrucciones y direcciones (el --- usuario ahora ya trabaja con instrucciones cómo LOAD, RSTORE, HALT DATA, etc. es menos complicado que el lenguaje de máquina, pero aún es complicado su uso por parte del usuario) .

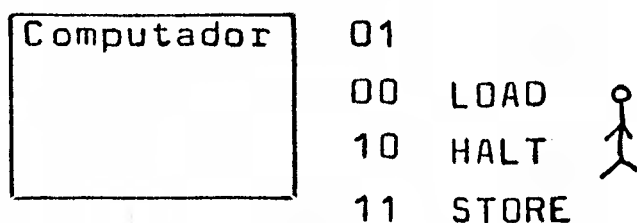


fig. a.1.2 El usuario da instrucciones en lenguaje ensamblador .

- 3/ La tercer fase, es en la que el usuario trabaja con lenguajes de alto nivel, es decir da las instrucciones a la máquina en un lenguaje. En los 50's surgió la idea de compiladores (--- lenguajes de alto nivel). En esta fase el lenguaje es independiente de la máquina y se traduce a ella cada línea de código, (se puede generar mucho código de máquina). En un lenguaje de alto nivel se permite describir fácilmente las indicaciones del usuario. En un compilador existe un programa que traduce el -- código a lenguaje de máquina, y este lenguaje de máquina se - ejecuta después ; por lo que el compilador verifica :
- 1/ Si el programa es válido (Sintaxis) .
 - 2/ Que quiere decir el programa (Semántica) .
 - 3/ Que código se genera para cada proposición del lenguaje (Implementación) .
 - 4/ Traducción del programa hecho en lenguaje de alto -- nivel a lenguaje de máquina (Que se genere código objeto).

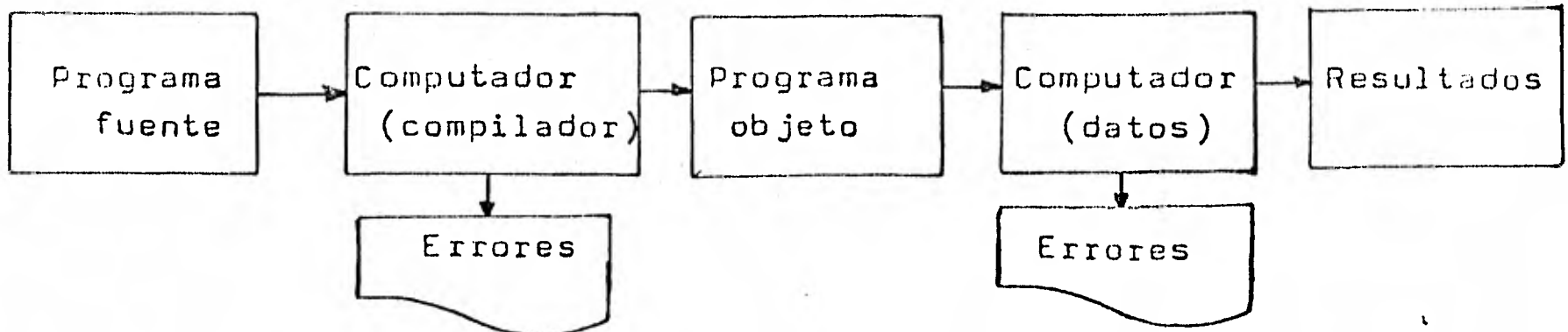


fig a.1.3 Proceso seguido para la ejecución de un programa hecho en un lenguaje de alto nivel .

La computadora Burroughs B-6700 cuenta con los siguientes compiladores : Algol , Fortrán , Pascal , Basic , etc .

En esta fase, el usuario de la computadora tiene más facilidades para resolver sus problemas , ya que evita el uso de lenguaje de maquina y de lenguaje ensamblador que eran muy complicados .

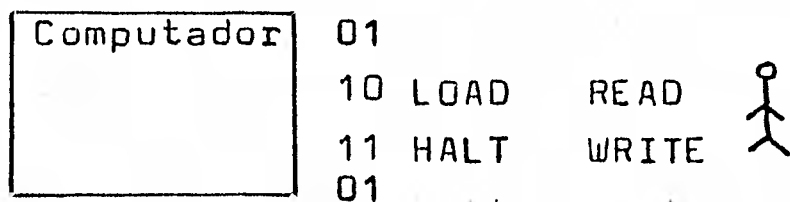


fig. a.1.4 El usuario da instrucciones en lenguaje de alto nivel .

4/ La cuarta fase , es en la que el usuario utiliza paquetes -- para resolver sus problemas ; es decir que ahora se considera - el manejo de programas ya hechos y que resuelven un tipo de problema específico . Lo que hace que el uso de la computadora sea accesible a personas que no tienen mayor conocimiento sobre --- computación .

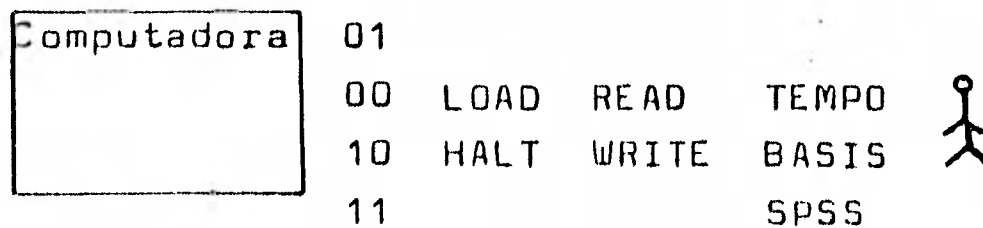


fig. a.1.5 El usuario resuelve su problema a través de paquetes de computadora .

Por lo que cada computadora cuenta con paquetes ; es decir con programas de proposito específico , y muchos de estos programas se encuentran en diferentes computadoras , por lo que el usuario cuando cambie de centro de trabajo y tenga que -----

resolver sus problemas , posiblemente en su nuevo centro de --- trabajo cuente con el mismo paquete o en su defecto contara con otro muy parecido .

La computadora Burroughs B-6700 cuenta con los siguientes paquetes entre otros :

Para problemas de :

Estadística	1/	BASIS
	2/	SPSS
	3/	IMSL
Inv. de operaciones	1/	TEMPO
	2/	TEMPO/NETWORK
	3/	ALPS/1
	4/	IMSL
	5/	GAMA
	6/	PROMIS
Simulación	1/	SIMULA
	2/	DYNAMO
Graficación de mapas	1/	SYMAP
	2/	INGRID
Clasificación de fichas bibliograficas	1/	ONIONS
	2/	LIBRO/UNAM
Ingenieria	1/	STRESS
	2/	CECAFI

3.0 FORMAS DE TENER ACCESO A LA COMPUTADORA BURROUGHS B-6700

Todas las aplicaciones de la computadora consisten de tres fases - Entrada , Procesamiento y Salida . Esto es , la información entra a la computadora , es procesada (y esto quiere decir se realizan los calculos) y los resultados se comunican al usuario . La entrada puede ser a través de tarjetas perforadas , cintas magneticas , discos magneticos , --- terminales de la computadora ; la salida se puede obtener en los medios anteriores o bien en papel impreso .

La tarjeta perforada (figura a.2.1) a sido utilizada desde hace mucho tiempo , su desarrollo fue motivado porque se requería que un censo federal de los E.E.U.U. se llevara a cabo cada 10 años . Cómo la ciudad se expandia , el procesar los datos del censo consumia mucho tiempo y el gobierno de los E.E.U.U. requería la tabulación de los datos . Herman Hollerit introdujo la tarjeta de 80 columnas a fines de 1880. Esta - tarjeta perforada se proceso en dispositivos mecanicos , los cuales no eran una computadora , pero servían para procesar - estas tarjetas en el siglo XIX . La tarjeta perforada tiene - 80 columnas verticales , cada una de las cuales consta de 12 renglones y contiene un caracter particular , cada caracter tiene una única combinación de renglones perforados . La --- letra A por ejemplo tiene una perforación en el renglón 12 y en el renglón 1 (ver figura a.2.1) . La letra B tiene una -- perforación en el renglón 12 y en el renglón 2 . En la figura a.2.1 , A y B aparecen en las columnas uno y dos respectivamente. Los tres renglones de la parte superior de la tarjeta , se -- les conoce cómo zonas (renglones 12,11, y 0) ; los otros -- renglones se les conoce cómo digitos . Cada letra tiene dos - perforaciones , una zona y un digito . Las letras de la A a la I tienen la misma zona , el renglón 12 . Las letras de la J a la R tienen una perforación en el renglón 11 , y de la S a la Z en el renglón 0. Los numeros , del 0 al 9 tienen solo una perforación en el renglón de su digito . Por ejemplo la columna 47 contiene el número 9 .

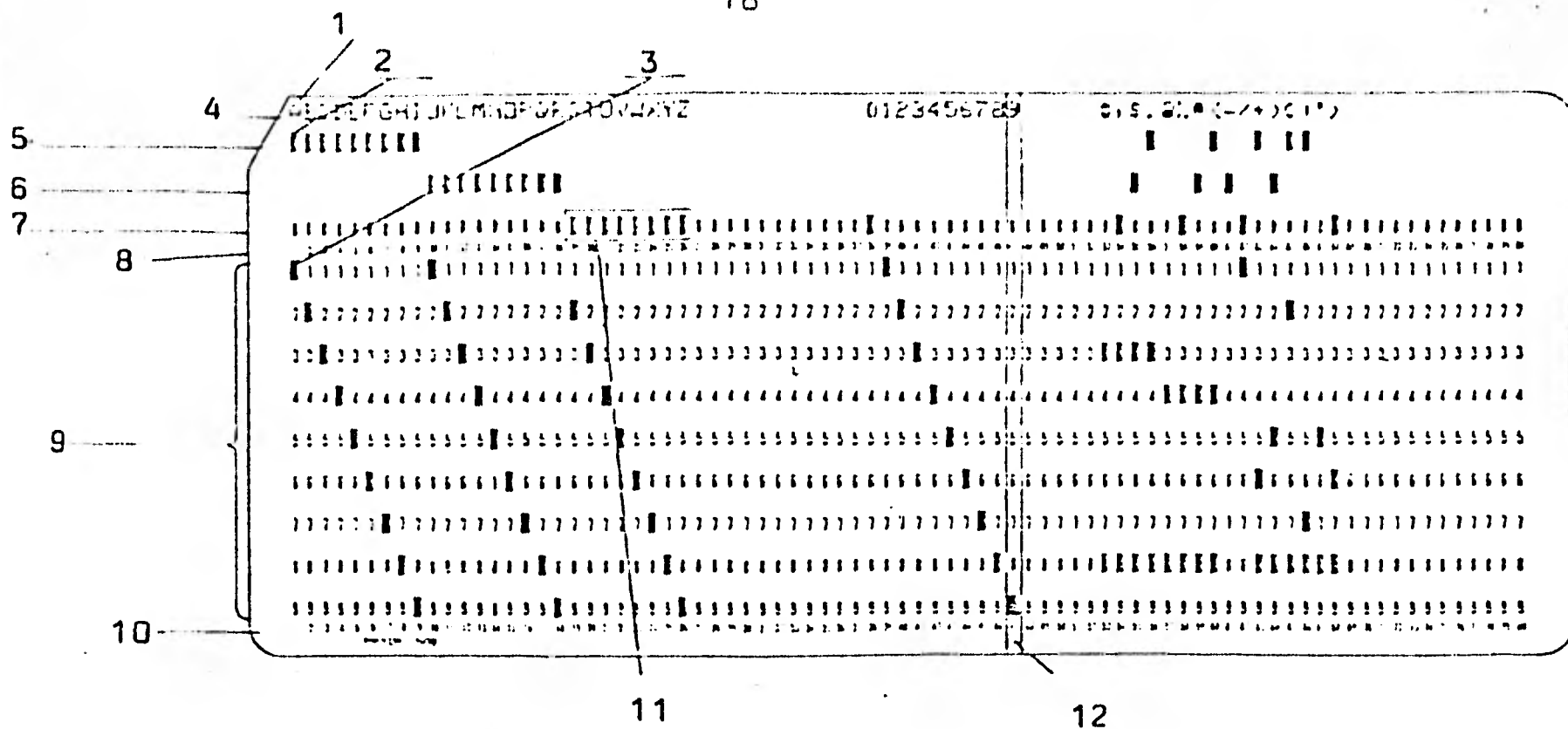


figura a.2.1 Tarjeta perforada de 80 columnas .

Descripción de la tarjeta perforada :

- 1 .- Letra A , con una perforación en el renglón 12 y una perforación en el renglón 1 .
- 2 .- Perforación en el renglón 12 .
- 3 .- Perforación en el renglón 1 .
- 4 .- Interpretación alfabética .
- 5 .- Renglón 12 .
- 6 .- Renglón 11 .
- 7 .- Renglón 0 .
- 8 .- Indicador de las columnas .
- 9 .- Renglonas 1-9 .
- 10 .- Indicador de las columnas .
- 11 .- Las letras S-Z , todas tienen una perforación en el renglón 0 .
- 12 .- El número 9 esta perforado en la columna 47 .

El color de las tarjetas no importa , aunque en -- algunas instalaciones se utilizán tarjetas de un color al -- principio y de otro color al final , para indicar donde empieza un paquete de tarjetas y en donde termina .

Ahora bién , la reproducción física de un paquete - de tarjetas no es lo mejor , pero tampoco es ineficiente ; un metodo superior es desarrollar programas en línea a través de una terminal remota , lo que se lleva a cabo por medio de un editor de textos (En Burroughs B-6700 llamado CANDE (de -- COMAND AND EDIT)). Para accesar de esta forma a la computa- dora , el usuario se situa frente a una terminal remota e -- interactua con la computadora mediante el lenguaje editor -- CANDE ; el cual se utiliza para crear archivos de datos , --- programas y llevar a cabo instrucciones en los mismos por -- medio de la terminal ; tales cómo lo son las siguientes - -- Insertar un texto despues de la i-esima línea ; eliminar un conjunto de líneas ; reemplazar una cuerda en una línea ; - terminar el proceso de edición ; y otros . ■■

Esto es se puede crear un programa en la terminal , modificarlo y almacenarlo en un archivo , y nuevos cambios - se podrían almacenar en el mismo archivo o bién en un nuevo - archivo . Aunque la mayor ventaja es que el programa puede ser ejecutado desde la terminal y su salida examinada en la misma terminal . Para apreciar el significado de este concepto , - considere un ambiente en el que se trabaja con tarjetas - perforadas ; en el cual las correcciones son hechas a los --- programas contenidos en tarjetas , volviendolas a hacer y -- colocandolas en el paquete de tarjetas del programa , y esto se lleva a cabo cada vez que se tengán errores en la compilación de los programas , aun cuando en una situación ideal el tiempo de espera por el listado no es menor de 20 minutos , despues de lo cual uno revisa el error de compilación , que fue --- causado tal vez por un parentesis omitido o por algun otro -- detalle trivial , sin embargo el ciclo de tarjetas se repite y se espera repetidamente la salida en la impresora con --- resultados desconocidos durante otros 20 minutos .

Si los cambios son hechos en una terminal remota (on-line , en línea) y el programa modificado se ejecuta a - través de la terminal , los resultados se pueden conocer más rapidamente en la misma terminal , y cuando el programa compila correctamente sin errores , es decir que el programa es apto

■■ Estas instrucciones se verán posteriormente .

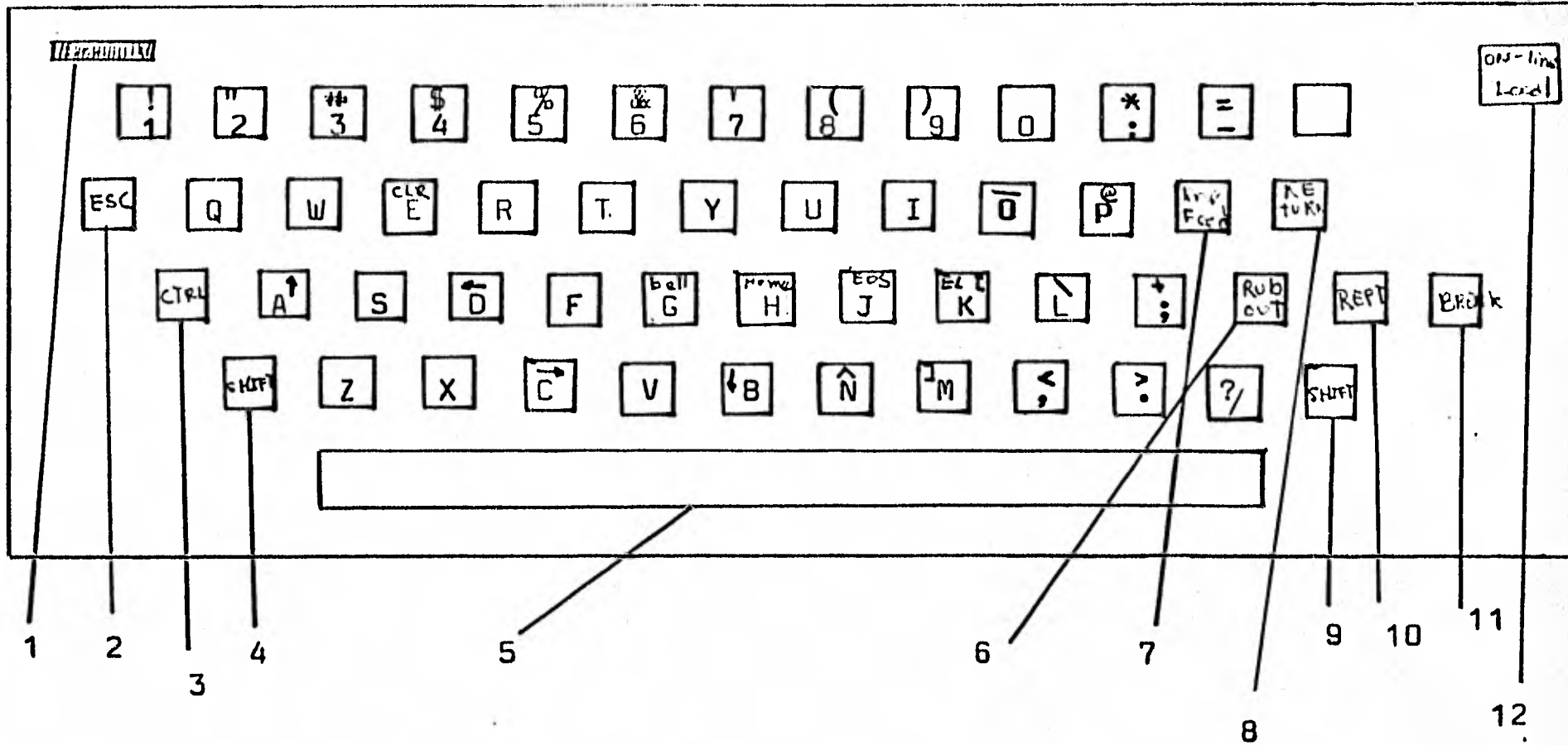


figura a.2.2 Tablero de una terminal remota .

para ejecutarse , el usuario puede pedir que la salida sea -
 desplegada en la terminal o bién que la salida se efectue en
 una impresora de papel , correcciones adicionales que se ---
 requieren pueden efectuarse de inmediato en la terminal . El
 resultado de trabajar con una terminal es de que el conocimien-
 -to individual acopla significativamente más trabajo en una
 terminal remota que en el ambiente convencional batch (-
 tarjetas perforadas) . las instrucciones del editor son --
 diferentes de las instrucciones de algún lenguaje (Algol ,
 Fortrán , Cobol , etc.) , de programación .

El lenguaje editor es utilizado en una fase interac_
 tiva , por tal motivo las instrucciones que suministra el ---
 usuario en la terminal , representán los datos que accesa el
 lenguaje editor . Entre las instrucciones con que se cuenta -
 están las de crear archivos de datos tipo - DATA , ALGOL ,
 FURTRAN , COBOL y otros ; insertar lineas en un texto , ----
 modificaciones en el proceso de edición , terminación del --
 proceso de edición , etc .

En la figura a.2.2 , en la que se muestra el esquema de un tablero de terminal remota , notamos su parecido con el de una maquina convencional , para escribir a maquina , con la consideración de que el tablero de una terminal remota - tiene otras teclas adicionales , especiales , con una función especifica , por lo que a continuación se describen las teclas especiales .

Descripción del tablero de una terminal remota (figura a.2.2):

- 1 .- Control de brillo de la pantalla de la terminal .
- 2 .- ESC , se utiliza para borrar lineas de la pantalla junto con otra tecla ; por ejemplo ESC y CLR , borrarán todas las lineas de la pantalla ; con EL (K) borra la linea que se acaba de teclear ; con EOS (J) borra de la linea que se acaba de teclear hacia abajo .
- 3 .- CTRL , se utiliza para controlar el cursor .
- 4 .- SHIFT , se utiliza para transmitir el caracter de la parte superior de las teclas (si lo hay) .
- 5 .- Esta tecla funciona cómo el espaciador de una maquina de escribir .
- 6 .- RUB-OUT , se utiliza para discontinuar la linea tecleada.
- 7 .- LINE-FEED , se utiliza para bajar el cursor una linea.
- 8 .- RETURN , se utiliza para transmitir la linea tecleada.
- 9 .- Similar a 4 .
- 10 .- REPT , se utiliza al mismo tiempo que una de las - teclas de caracteres , repitiendo el caracter .
- 11 .- BREAK , se utiliza para discontinuar una ejecución .
- 12 .- Esta tecla se utiliza para que la terminal transmita las instrucciones a la computadora ; hacia arriba (on-line) indica que la transmisión de instrucciones se lleve a cabo ; hacia abajo (local) indica que la terminal no transmita a la computadora las instrucciones.

El uso de terminales remotas surgio con la idea de tiempo compartido , motivo por el cual se describen a ----- continuación algunas técnicas comunmente utilizadas para --- tiempo compartido .

TIEMPO COMPARTIDO.- Es un modo de operar del sistema en el -
 cual el sistema atiende a varios usuarios al mismo tiempo , -
 vía terminales remotas , dándole la sensación al usuario de -
 ser atendido en particular . Las características de este modo
 de operar el sistema son : tiempo de respuesta más rapido al
 usuario , uso simultaneo de los recursos para varios procesos,
 en esto se considerán tres razones :

- 1/ Hacer uso de otros programas dentro de los propios .
- 2/ Tener acceso a bases de datos , paquetes .
- 3/ Evitar tener copias del mismo recurso en memoria cada
 vez que un proceso lo requiera

La manera de implementar el tiempo compartido es mediante --
 multiprogramación .

MULTIPROGRAMACION.- En multiprogramación más de un programa se
 almacena en memoria al mismo tiempo .

MULTITASKING.- Es la capacidad de un sistema para soportar más
 de una actividad de proceso ; existe en la multiprogramación
 pero no necesariamente puede existir en multiaccesamiento .

MULTIACCESAMIENTO.- A los usuarios se les permite tener acceso
 a puntos del sistema simultaneamente , vía terminales remotas.

MULTIPROCESAMIENTO.- Este termino se aplica a las computadoras
 que soportán más de un procesador .

CONCURRENCIA.- Se refiere a la existencia o posible existencia
 de varios procesos activados en paralelo ; un proceso se refiere
 a cualquier actividad de cálculo existente en la máquina ; los
 procesos en paralelo los definimos cómo la ejecución de varios
 procesos al mismo tiempo , desde su inicio hasta su termino .
 Los procesos interactúan de dos maneras , indirectamente ---
 compitiendo por los mismos recursos e indirectamente -----
 compartiendolos .

JOB.- Es un conjunto de proposiciones o comandos para la máquina (Sistema operativo) . Mediante el cual el usuario puede entre otras cosas , compilar y correr sus programas , ejecutar programas ya existentes en la máquina (a los que tenga acceso) , etc .

TASK.- Es cada una de las partes que constituyén un job . Esto es , un job puede mandar ejecutar un programa en código objeto, compilar un programa en algún lenguaje de programación , llamar algún paquete , programas de biblioteca ; cada una de estas acciones , constituyén un task .

WFL.- Es el medio por el cual se describén y presentán "jobs" a la computadora Burroughs B-6700 . WFL (WORK FLOW LANGUAGE) Es el lenguaje por medio del cual se describén , cada uno de los jobs cómo un conjunto de tareas (tasks) que se llevarán a cabo . WFL acepta jobs desde una grán variedad de fuentes :

- 1/ Lectora de tarjetas .
- 2/ Terminales remotas .
- 3/ Consola de operador , y otros .

El compilador WFL , produce código para controlar los tasks dentro de un job , tal cómo el usuario lo indica . El compilador WFL realiza las siguientes funciones .

- 1/ Verifica la sintaxis de las instrucciones .
- 2/ Genera código para manipular las tareas que constituyén el trabajo .
- 3/ Genera el archivo de trabajo en disco .

4.0 I N T R O D U C C I O N A L E D I T O R C A N D E

Cuando el usuario se conecta al sistema , de la -
 computadora Burroughs B-6700 , mediante una terminal remota ,
 el usuario estara interactuando con la computadora mediante -
 un lenguaje editor llamado CANDE (COMAND AND EDIT) . CANDE -
 proveé comandos para controlar programas , editar archivos --
 fuente o de datos , para comunicarse con los operadores , etc.

En este capítulo se verán los comandos más utilizados
 del editor CANDE , y se considerara que en las indicaciones
 que se den , los terminos que se encuentren encerrados entre
 parentecís de pico , serán terminos que suplira el usuario .
 Esto es , por ejemplo <FILENAME> indicara que el nombre de un
 archivo sera utilizado en esta posición en el comando .

No todos los comandos de CANDE serán mostrados aquí
 unicamente se mostrarán los más utilizados , y serán descritos.
 Para una mayor información al respecto se sugiere ver la ---
 referencia [1] , o bién la tarjeta de CANDE .

Para laborar con la computadora , mediante una ---
 terminal remota del sistema , lo primero que se requiere es -
 tener una clave de usuario del sistema ; ya que esta sera la
 forma de tener acceso a la máquina , porque es lo primero que
 se tendra que teclear en la terminal , para poderse comunicar
 con la máquina .

SESION DE CANDE .- El tiempo en que el usuario interactua con la computadora , desde que empieza hasta que se despide por medio de un BYE , SPLIT o una falla del sistema , se le conoce como una "SESION" , Este es el equivalente interactivo de un "JOB" por tarjetas . Durante una sesión CANDE respondera a dos clases de comandos , comandos de control y comandos CANDE .

Los comandos de control serán una clase especial de comandos y serán ejecutados inmediatamente , a menos que se tenga una cola de comandos CANDE , porque entonces se irán ejecutando en orden secuencial . Los comandos de control estarán precedidos por el caracter de control "?" , sin las comillas .

ARCHIVO DE TRABAJO CANDE .- Un archivo de trabajo es cómo una especie de "pizarrón" en el cual se trabaja . Este puede ser creado con un nombre temporal (o título) y un tipo de formato utilizando el comando MAKE . Por ejemplo :

```
MAKE XYZ
```

Las líneas de este archivo de trabajo serán secuenciadas , y se podrá referir a ellas por medio de su número de secuencia. La secuencia es controlada por el usuario y se puede cambiar cuando se crea necesario , las correcciones a el archivo se pueden hacer refiriendose a una línea en particular o a un rango de líneas . Las correcciones se pueden hacer en cualquier orden , cualquier línea se puede acceder en cualquier momento . El archivo XYZ de el ejemplo sera de un formato tipo "SEQDATA" , con un número de secuencia de ocho dígitos , empezando en 100 y con incrementos de 100 , esta secuencia estara en las columnas 73-80 , del registro de 80 columnas , en EBCDIC , y sus números de secuencia serán parte del archivo .

SALVANDO UN ARCHIVO DE TRABAJO .- Para esto se utiliza el comando SAVE , el cual origina que una copia del archivo de trabajo se almacene en una biblioteca del usuario bajo este

título , es decir XYZ reemplazara cualquier versión previa del archivo de este título . El comando

SAVE AS <FILENAME>

origina que una copia del archivo de trabajo sea almacenada en la biblioteca del disco del usuario , bajo el FILENAME - especificado , no con su título original .

BORRANDO ARCHIVOS DE TRABAJO .- El comando REMOVE borrara el archivo de trabajo .

REMOVE <ARCH1,ARCH2,...>

Este comando borrara los archivos especificados de la biblioteca en disco del usuario .

LLAMANDO UN ARCHIVO DE TRABAJO .- Para esto se utiliza el -- comando GET o el comando LOAD .

GET <FILENAME>

o

LOAD <FILENAME>

Este comando ocasionara que sea traída una copia al area de - trabajo , de un archivo almacenado previamente , por CANDE -- cómo archivo fuente .

GET <FILENAME><RANGO-DE-SECUENCIA>

Unicamente se trae al area de trabajo la porción especificada del archivo . Si unicamente una porción del archivo se llamo el archivo de trabajo no tendra nombre .

LISTANDO LOS NOMBRES DE LOS ARCHIVOS .- El comando FILES lista los nombres de todos los archivos residentes en la biblioteca del usuario .

LISTANDO EL CONTENIDO DE UN ARCHIVO DE TRABAJO .- El comando LIST ocasiona que el archivo de trabajo sea listado por --- completo .

LIST <S1,S2,S3-S4,S5,...>

Indica que se listén unicamente estas lineas especificadas ; los numeros de secuencia deberán estar en orden creciente .

Cómo por ejemplo : LIST 200,700,1200-1700,2800 .

LIST : UN

Ocasiona que el archivo sea listado sin los numeros de secuencia.

LIST <FILENAME> , es decir LIST XYZ

Listara el archivo nombrado pero no lo traera al area de --
trabajo .

BUSQUEDA EN UN TEXTO .- El comando FIND hace una busqueda en un archivo hasta que aparece una cadena especifica de caracteres o bién termina cuando busco en todo el archivo sin exito .

La forma general del comando es :

FIND LITERAL <DELIMITADOR><TEXTO><DELIMITADOR>: TEXT

En donde

- | | |
|---------------|---|
| LITERAL | - Es opcional y se utiliza para una comparación exacta en el texto , incluyendo blancos , aunque no es ecencial . Si esta opción no se utiliza , los caracteres en blanco no tendrán significado . |
| <DELIMITADOR> | - Un caracter no alfa-numerico tal cómo - una diagonal (/), el cual no aparesca en el texto . |
| <TEXTO> | - Cuerda de caracteres que sera buscada . |
| : TEXT | - Es opcional y se utiliza para que las - lineas en donde el texto se encontro seán desplegadas en la terminal . Si no se - utiliza esta opción , unicamente los -- numeros de las lineas es desplegado . |

Ejemplos de busqueda

1/ FIND /FILEID/

#WORKFILE DOC/CANDEJR

18400, 18420, 18430, 18800, 18900

2/ FIND\$FILEID\$: TEXT

#UPDATING

#WORKFILE DOC/CANDEJR

18400 FIND .FILEID.

18420 SPACES AROUND FILEID ARE DELIMITERS

18430 TESTING THE WORD , FILEID.

18800 "FILEID" . THE SEQUENCE NUMBERS OF
18900 THE 6 CHARACTERS FILEID ARE DELIMITERS .

NUMEROS DE SECUENCIA .- El comando SEQ ocasiona que se numerén las líneas que se ván insertando en un texto, dentro de un -- archivo de trabajo , y esto ocurre a cada control de carro , es decir , cada vez que se utiliza un RETURN . El sistema --- suministra un espacio para que el texto sea legible , pero el primer caracter suministrado por el usuario estara en la ---- columna 1. Un RETURN a continuación de la secuencia suministrada por el sistema ocasionara que se suspenda este modo , y esto no se darán más líneas de secuencia a menos que se indique esto por parte del usuario . Los incrementos en la secuencia por omición son de 100; a menos que el usuario indique otro --- incremento .

SEQ 990+10

Ocasiona que se empiece la secuencia en la línea 990 con --- incrementos de 10. En el caso de solo querer insertar una -- línea , se teclea el número de la línea y su texto .

255ABC

Automaticamente se inserto la línea 255 , empezando en la col. 1.

SEQ END

Empieza a dar líneas de secuencia a partir de la última línea del archivo de trabajo .

RESECUENCIANDO LAS LINEAS DE UN ARCHIVO DE TRABAJO .- Para esto se utiliza el comando RESEQ .

RESEQ +<I>

Este comando asigna nuevos números de secuencia al archivo de trabajo sin intercambiar el orden, empezando con el número de secuencia B , y continuando en intervalos de I . Si B e I se omitén se considera que el inicio es en la línea 100 con ---- incrementos de 100 .

ELIMINACION DE LINEAS .- Esto se lleva a cabo por medio del - comando DELETE .

DELETE <S1>, <S2>, ...

Con esta instrucción se elimina cada una de las líneas, a las que corresponde el número de secuencia, del archivo de trabajo.

DELETE <S1> - <S2>

Elimina todas las líneas de S1 a S2 inclusive.

SUBSTITUCION DE PATRONES - FIJANDO UNA LINEA .- El comando FIX altera el contenido de una línea, en el archivo de trabajo, insertando un nuevo texto en el lugar especificado del texto de la tarjeta. El formato general del comando es:

FIX <NOSEQ> <DELIM><VIETEXT><DELIM><NUETEXT>

En donde

- <NOSEQ> - Número de secuencia de la línea que será modificada.
- <DELIM> - Un carácter no alfa-numérico, tal como un signo de dólar, \$.
- <VIETEXT> - Texto que será reemplazado.
- <NUETEXT> - Texto que sustituirá al reemplazado.

Ejemplo

FIX 398 \$AB\$XYZ

Se localiza la primera vez que aparece la cadena AB en la línea 398 y la reemplaza por XYZ.

SUBSTITUCION DE PATRONES - EN MULTIPLES LINEAS .- Para este propósito utilizaremos el comando REPLACE, el cual funciona en forma de una combinación del comando FIND y del comando FIX. La forma general del comando REPLACE es:

REP <DELIM><VIETEXT><DELIM><DELIM><NUETEXT><DELIM><S1-S2>:TEXT

Este comando ocasiona que un nuevo texto sustituya un texto anterior, en el conjunto de líneas comprendido de la línea S1 a la línea S2; en el caso de omitir S1 y S2 se considera que el cambio se efectuara en todo el archivo de trabajo.

Ejemplos

1/ REPLACE .AB..ABB.

En el archivo de trabajo se busca la cadena de caracteres AB , y cada vez que esta aparece es reemplazada por la cadena ABB , en todo el archivo de trabajo .

2/ REP /I3//I3(INDEX,ARRAY):TEXT

Reemplaza cada vez que aparezca I3 por "I3(INDEX,ARRAY)" .El texto y el número de línea se muestran cada vez que ocurre -- un reemplazo .

INSERCIÓN DE LINEAS DE OTRO ARCHIVO .- Esto se puede llevar a cabo con el comando INSERT ; este comando indica que se copie un conjunto de líneas del archivo de trabajo , o de algún otro archivo compatible y las coloque en el archivo de trabajo con una nueva secuencia . Si como resultado de hacer la inserción se traslapa una línea o más , la inserción no se lleva a cabo. La forma general utilizada para este comando es :

INSERT <FILENAME> <SEQ1-SEQ2> AT + <I>

En donde

- <FILENAME> - Es opcional , y representa el título del archivo , del cual las líneas serán -- copiadas .Su omisión indica usar el presente.
- <SEQ1-SEQ2> - Es opcional , y representa el rango de los números de secuencia que serán copiados del archivo fuente . Si no se especifica el archivo se copia por completo .
- - Especifica el número de secuencia en el archivo de trabajo , en donde la primer línea será colocada. END se puede utilizar en lugar de B , para que la inserción se lleve a cabo al final del archivo de trabajo.
- <I> - Es opcional, y indica el incremento que será agregado a los siguientes números de secuencia de las líneas , por omisión el valor es 100 .

Ejemplos :

1/ INSERT MYFILE 100500-101000 AT 20000

Copia del archivo MYFILE las líneas, 100500 hasta 101000 , y las inserta empezando en la línea 20000 , utilizando el incremento

de omisión de 100.

2/ INSERT 2670-2931 AT END

Copia de la línea 2670 a la 2931 , de el archivo de trabajo , y las coloca al final del mismo archivo de trabajo .

3/ INSERT SUBLIB/GAUSHAUS AT END+20

Copia el archivo llamado SUBLIB/GAUSHAUS y lo inserta al final del archivo de trabajo , utilizando incrementos de 20 para las líneas , en sus numeros de secuencia .

MOVIENDO LINEAS .- El mover líneas de algun área a otra del archivo de trabajo , se lleva a cabo por medio del comando - MOVE . Si en el movimiento una línea se traslapa , entonces - este no se lleva a cabo . El formato general de este comando es:

MOVE <SEQ1-SEQ2> TO + <I>

En donde

- <SEQ1-SEQ2> - Indica el rango de la secuencia que será movida ; es decir todas las líneas desde SEQ1 hasta SEQ2 inclusive .
- - Número de secuencia a donde será movida la primer línea , aunque puede ser --- especificado el final con END .
- <I> - Es opcional , y indica el incremento que utilizara par las subsecuentes líneas .

Ejemplos

1/ MOVE 234-417 TO 595+5

Mueve de la línea 234 a la 417 , a una nueva localidad , en las líneas 595,600,605,....

2/ MOVE 1400 TO 1150

Mueve la línea 1400 , a la línea 1150 .

3/ MOVE 16700-END TO 1710+10

Aquí se indica mover de la línea 16700 hasta la última línea del archivo de trabajo , a la línea 1710 cómo principio y --- con incrementos subsecuentes en la secuencia de 10, esto es 1710,1720,1730,....

EJECUCION DE UN PROGRAMA INTERACTIVO .- El comando CANDE se utiliza para que un programa se ejecute en forma interactiva . Por ejemplo

RUN MYPROGRAM

Este comando ocasiona que la versión compilada del programa MYPROGRAM , se ejecute . Si la versión compilada no se ----- encuentra en la biblioteca , entonces aparece el siguiente -- mensaje "DSED MISSING CODE FILE" . Pero si este programa no es de biblioteca sino del usuario entonces lo compila y lo ----- ejecuta .

TERMINACION DE LA EJECUCION DE UN PROGRAMA DESDE CANDE .- En algunas ocasiones el usuario , requerirá descontinuar un programa el cual se esta ejecutando en forma interactiva , es decir -- se requerirá terminar su ejecución . Esto se lleva a cabo por medio del comando siguiente :

?DS

Este comando termina la ejecución en forma anormal , del --- programa interactivo que se ejecuta en ese momento . Este -- comando es innecesario si el programa tiene una terminación -- normal .

COMUNICACION CON OTROS USUARIOS .- En una terminal remota se pueden enviar o recibir mensajes , por parte de los usuarios . Cómo por ejemplo

?TO AR85 UN MENSAJE

De esta forma se envía un mensaje al usuario con clave AR85.

?SS SPO UN MENSAJE

De esta forma se envía un mensaje al operador.

RECUPERACION DE UN ARCHIVO DE TRABAJO .- Si ocurre una terminación anormal , de la sesión de CANDE , ya sea porque de repente -- se desconecto el sistema , o por otras fallas del sistema , - los registros actualizados del archivo de trabajo sera ---- almacenado en un archivo de recuperación (RECOVERY).

Cuando el sistema se restaure y el usuario se vuelva a conectar con la computadora mediante una terminal remota, -

CANDE le indicara al usuario la existencia de estos archivos mediante un mensaje apropiado .

~~#~~RECOVERY DATA:

NNNN WORKFILENAME (DATE)

En donde NNNNes el indicador para recuperar el archivo de --- trabajo . Y para recuperar este archivo y traerlo al área - de trabajo se hace lo siguiente :

RECOVER <NNNN>

en caso de que este archivo no se quiera recuperar , se puede borrar con la siguiente instrucción :

DISCARD <NNNN>

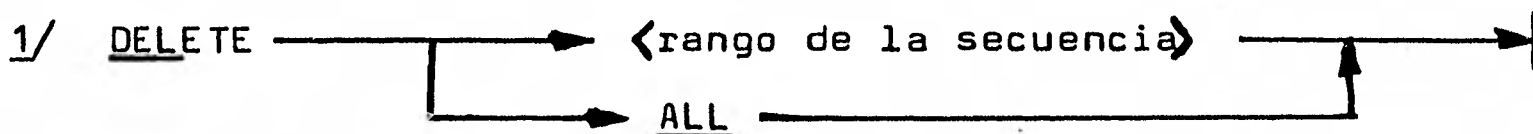
Ahora bién la forma en que se describe la sintaxis de CANDE , en la tarjeta de referencia del editor , es por -- medio de diagramas de rieles . Por lo que se considera conveniente conocerlos , y ver la forma en que se utilizán . Motivo por - el cual se describén a continuación .

DIAGRAMAS DE RIELES

Un diagrama de rieles , es una tecnica utilizada -- para representar graficamente la sintaxis de los elementos de un lenguaje . Recorriendo un diagrama de rieles de izquierda a derecha , o en la direccíon que indiquén las puntas de las flechas , y siguiendolo hasta los limites indicados por los puentes , produciremos una instrucción valida sintacticamente, o bién podremos verificar si una instrucción es valida , con el recorrido en el diagrama de rieles del lenguaje . La continuación de una linea de un diagrama a otro estara representada por una flecha derecha ">" apareciendo al final de la linea - actual y al inicio de la siguiente linea .

El diagrama de sintaxis completo termina con una -- barra vertical "/" o con un signo de porcentaje "%" . Los -- datos encerrados entre parentecis de pico "< >" serán variables que el usuario debe indicar .

Ejemplos

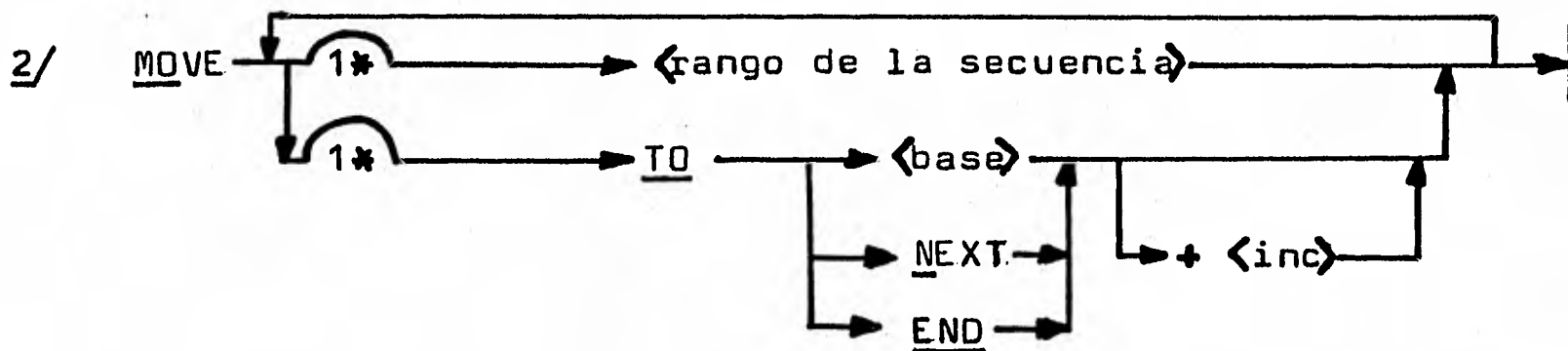


La instrucción DELETE se utiliza para borrar líneas de un archivo de trabajo ; en este caso se tienen dos opciones , lo que se observa del diagrama de rieles . ■E■

Estas dos opciones són :

1.1 DELETE rango de la secuencia

1.2 DELETE ALL



La instrucción MOVE se utiliza para mover líneas de un punto a otro dentro del archivo de trabajo y -- cambia sus numeros de secuencia . ■E1■

Las opciones que se tienen con esta instrucción són:

2.1 MOVE rango de la secuencia TO base

2.2 MOVE rango de la secuencia TO base + inc

2.3 MOVE rango de la secuencia TO NEXT

2.4 MOVE rango de la secuencia TO NEXT + inc

2.5 MOVE rango de la secuencia TO END

2.6 MOVE rango de la secuencia TO END + inc

■E■ La parte subrayada en las instrucciones , indica que con esa parte es suficiente , en la instrucción .

■E1■ El número antes del asterisco indica cuantas veces se debe recorrer forzosamente ese camino , sin el asterisco indica a lo más cuantas veces se puede recorrer ese camino .

Cuando el usuario se conecta al sistema , de la --
computadora Burroughs B-6700 , mediante tarjetas perforadas ,
el usuario estara interactuando con la computadora mediante -
un lenguaje llamado WFL (WORK FLOW LANGUAGE) . WFL provee -
comandos para controlar programas , manipular archivos fuente
o de datos , etc . Mediante tarjetas perforadas .

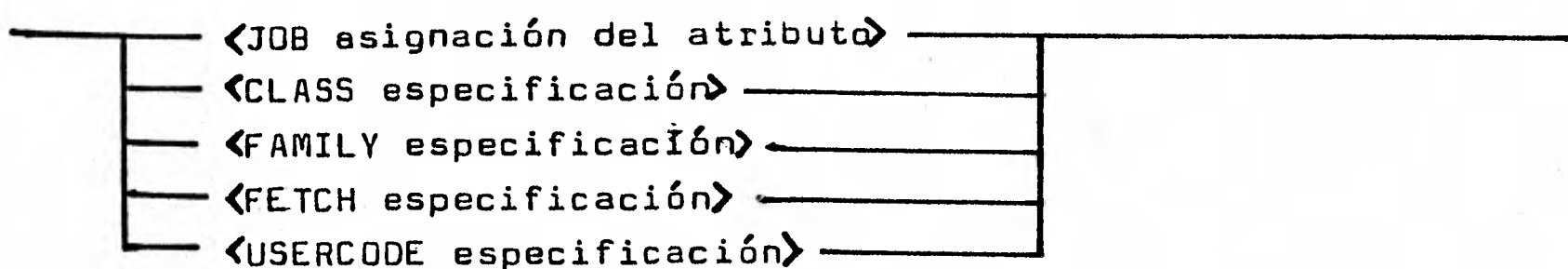
En este capítulo se verán los comandos más utilizados
de WFL , y se considerara que en las indicaciones que se den
los terminos que se encuentren encerrados entre parentesis de
pico , serán terminos que suplira el usuario . Esto es , por
ejemplo <FILENAME> indicara que el nombre de un archivo sera
utilizado en esta posición en el comando .

No todos los comandos de WFL serán mostrados aquí -
unicamente se mostrarán los más utilizados , y serán descritos.
Para una mayor información al respecto se sugiere ver la ---
referencia ■15■ .

Para laborar con la computadora , mediante tarjetas
perforadas , lo primero que se requiere es tener una clave de
usuario del sistema ; ya que por medio de esta se tendra --
acceso a la máquina .

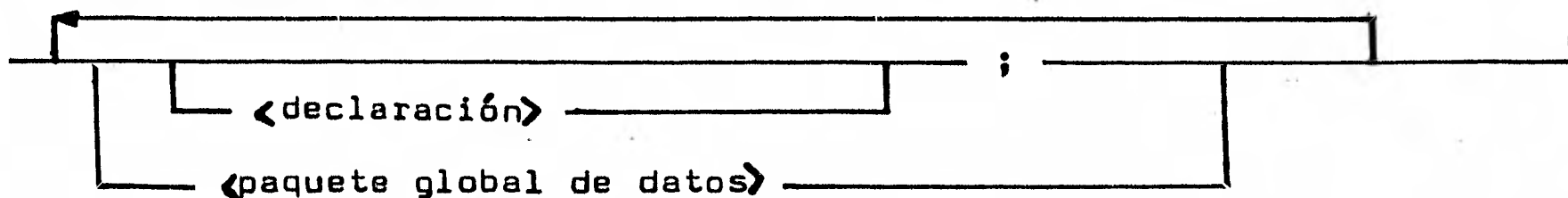
especifican los atributos de la tarea , dentro de la -----
 <especificación de atributos del job> , o se considera por el
 compilador WFL , si no se dan explisitamente los atributos del
 nombre. La <especificación de atributos> termina cuando aparece
 cualquier instrucción que no sea una <especificación de atributos> .

<especificacion de atributos del job>

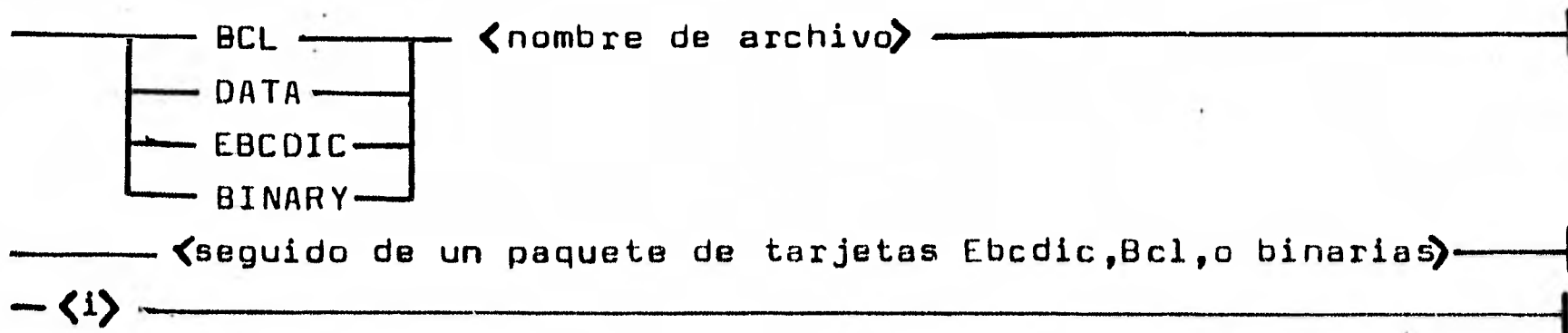


Cada una de estas <especificaciones de atributo del
 job> sera descrita posteriormente . Una declaración de atributos
 de archivo dentro de la <especificación de atributos del job>
 no será considerada. Una instrucción que empiece con FILE será
 interpretada cómo una declaración de archivo (cómo parte de
 la <declaración de instrucciones del job>) y terminara con la
 <especificación de atributos del job> .

<declaración de instrucciones del job>



<paquete global de datos>



Nota.- El paquete global de tarjetas puede no aparecer .

C A P I T U L O I I

M A N E J O D E L P A Q U E T E
T E M P O / M P S / A L L
E N L A S O L U C I O N D E
P R O B L E M A S D E
I N V E S T I G A C I O N D E O P E R A C I O N E S

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

En la actualidad muchos problemas conciernen --- esencialmente con la asignación de recursos limitados - dinero personal, materiales , máquinas , espacio , tiempo , etc . - para maximizar alguna medida de rendimiento o minimizar alguna medida de costo . La técnica para planear el asignación de -- recursos se conoce como programación matemática ; el caso -- especial en el cual la medida de rendimiento o costo es una - función de las variables controlables y las restricciones en la obtención o utilización de recursos se pueden expresar como ecuaciones lineales o desigualdades lineales se llama ----- programación lineal .

Más específicamente , el problema general de progra-
mación lineal trata la maximización o minimización de una ---
función de varias variables , llamada función objetivo , ----
sujeta a un conjunto de ecuaciones lineales o desigualdades -
lineales llamadas restricciones , Ninguna de las variables --
puede ser negativa . (Nótese , sin embargo , que una variable
negativa se puede expresar como la diferencia de dos variables
positivas) .

Formalmente el problema de programación lineal se -
expresa de la siguiente manera :

Maximice una función objetivo $Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$

sujeta a las restricciones

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\leq b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq b_m \\ x_j &\geq 0 \quad j = 1, 2, \dots, n \end{aligned}$$

Esto se puede escribir en forma más compacta como

$$\text{Maximice } Z = \sum_{j=1}^n c_j x_j$$

sujeta a

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \text{ para } i=1,2,\dots,m$$

$$x_j \geq 0 \text{ para } j=1,2,\dots,n$$

y , en notación matricial , como

$$\text{Maximice } Z = CX$$

sujeta a

$$AX \leq B ; X \geq 0$$

en donde

$$C = (c_1, c_2, \dots, c_n)$$

;

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

;

$$B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Análogamente , un problema de programación lineal para minimizar se puede enunciar . Estas formulaciones se pueden llamar las formas estándar para los problemas de maximización y de minimización lineal , respectivamente , ya que las desigualdades de restricción se expresan todas en la forma \leq para el problema de maximización y \geq para el problema de minimización . Se puede demostrar que cuando hay m restricciones en un problema de programación lineal , hay a lo más m variables o asignaciones en su solución . Existen varios métodos para determinar qué combinación de m de las x_j maximiza la función objetivo Z . ■■

■ Para algunos autores la forma estándar es cuando todas las restricciones tienen signo de igualdad .

Problemas de programación lineal que no encierrén -- más de dos variables se pueden resolver geoméricamente ; -- aunque una solución algebraica es usualmente más eficiente -- inclusive para estos problemas . Hay diversos métodos ----- algebraicos para resolver problemas de programación lineal ; para problemas más complicados que encierrán muchas variables y muchas restricciones , el método simplex revisado , el cual se utiliza en la mayoría de los paquetes de programas de --- computadora , parece ser el más eficiente .

Es conveniente por esta razón conocer estos ----- paquetes y utilizarlos , ya que los problemas reales són ---- sumamente complicados y unicamente por computadora se puede - tener la confianza de obtener resultados , para este tipo de problemas , que seán confiables . Por lo que en este capítulo se describira uno de estos paquetes , llamado TEMPO/MPS/ALL.

Una de las ventajas de conocer este paquete , es la de qué la mayoría de paquetes de programación matemática --- utilizán un formato parecido , en la descripción de problemas así como en la descripción de rutinas de optimización .

Para mayores detalles se sugieren las referencias - siguientes - ■2■ , ■6■ , ■7■ , ■8■ , ■9■ , ■10■ - .

En este capítulo se ilustra la solución de problemas de optimización pequeños . Pero la presentación es suficiente para la preparación de datos y solución de problemas grandes y complejos .

El paquete TEMPO esta diseñado para resolver problemas de la forma :

$$\text{Max (o Min) } Z = f(x)$$

sujeto a.

$$h_i(x) \leq b_i \quad ; \quad i = 1, 2, \dots, m$$

donde las b_i son conocidas y $f(x)$, $h_i(x)$ son funciones reales , por lo que un caso particular es cuando $f(x)$ y $h_i(x)$ son lineales y las x 's son mayor o igual a cero , y estos problemas són de la forma siguiente :

$$\text{Max (o Min) } Z = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$$

sujeto a.

$$a_{11}x_1 + \dots + a_{1n}x_n \leq b_1$$

$$\vdots$$

$$a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m$$

$$x_1, \dots, x_n \geq 0$$

y el problema anteriormente mencionado, en forma matricial se puede expresar como :

$$\text{Max (o Min) } Z = C'x$$

sujeto a.

$$Ax \leq b$$

$$x \geq 0$$

TEMPO para resolver este tipo de problemas cuenta con - rutinas de entrada , de optimización , de salida , de análisis de post-optimización , de preservación de bases , de utileria .

Los problemas que resuelve són de programación lineal , programación entera , programación separable , programación mixta, acotamiento superior generalizado (algoritmos GUB) .

A continuación se describén los elementos del marco --- teorico de programación lineal y posteriormente como resolver los problemas de programación lineal , por medio del paquete así como su interpretación de resultados .

2. MARCO TEORICO DEL ALGORITMO DEL METODO SIMPLEX

Un conjunto C contenido en R^n es convexo si para cualesquier dos vectores $X, Y \in C$ entonces $Z = aX + (1-a)Y \in C$, para cualquier $a \in [0, 1]$.

Considerese $C = \{x \mid Ax = b, x \geq 0\} \in R^{n+m}$, $A_{m \times (n+m)}$ generado de las restricciones iniciales despues de introducir variables de holgura y artificiales. Lo que se afirma es que una solución al problema de programación lineal corresponde a un punto extremo del conjunto de restricciones C .

1/ $x \in C$ es un punto extremo de C si y solo si no existe $y \in C$ $z \in C$ ($y \neq z$) y $n \in (0, 1)$ tal que $x = ny + (1-n)z$

2/ Una solución factible es cualquier vector $x \in R^{n+m}$ que satisfice a C , es decir $x \in C$.

3/ Una solución basica factible de C es una solución factible con a lo más m x_j positivas.

4/ Una solución basica factible no degenerada es una solución basica con m x_j positivas. Por lo que de 3 y 4 una solución basica factible tiene m x_j positivas, reescribiendo C .

$$C = \left\{ K \mid \sum_{i=1}^{n+m} K_i P_i = P_0, K_i \geq 0 \right\}$$

Donde P_i representa la i -ésima columna de A y P_0 el vector b , ahora bien si C es expresado cómo una combinación lineal de las variables basicas, entonces una solución optima del problema original se expresa cómo

$$x^0 = \sum_{j=1}^m K_j P_j \quad \dots (a)$$

En el que los P_j se han reordenado de tal manera que las primeras m columnas forman una base optima de A , se ha considerado que las m restricciones en A son linealmente independientes (se asume no degeneración), y de 4 cada K_j en (a) es positiva para $j=1, 2, \dots, m$ y cero para $j=m+1, \dots, n+m$.

5/ K^0 es un punto extremo de C si y solo si los $K_j^0 - 0$ son --
coeficientes de vectores linealmente independientes .

Lo que se exhibe de la siguiente forma. Sean $r - 0$
elementos de K^0 y reordenese el arreglo de tal forma que los --
primeros r elementos seán diferentes de cero , las columnas de
 A reordenense de la misma manera y suponga que los primeros r -
vectores son linealmente dependientes , por lo que existen $z_1,$
 z_2, \dots, z_r e R no todos cero de tal forma que

$$\sum_{j=1}^r z_j P_j = 0 \quad \dots (b)$$

pero con $K^0 \in C$, (b) se tiene entobces que

$$\sum_{j=1}^r (K_j^0 P_j + z_j P_j) = b$$

y se sigue que para cualquier $\epsilon > 0$, $\sum_{j=1}^r (K_j^0 + z_j) P_j = b$ donde

$K_j^0 > 0$, $j=1,2,\dots,r$ se puede elegir $\epsilon > 0$ de tal forma que --
 $K_j^0 - z_j > 0$ y $K_j^0 + z_j > 0$ seán

$$K^1 = \begin{bmatrix} K_1^0 & - & z_1 \\ \vdots & & \vdots \\ K_r^0 & - & z_r \end{bmatrix} , \quad K^2 = \begin{bmatrix} K_1^0 & + & z_1 \\ \vdots & & \vdots \\ K_r^0 & + & z_r \end{bmatrix}$$

entonces $K^0 = (1/2)(K^1 + K^2)$ y K^0 no es un punto extremo de C
por lo que los primeros r elementos de K^0 son distintos de cero
y los primeros r vectores son linealmente independientes .

Por demostrar : si los primeros r vectores de A son
linealmente independientes entonces K^0 es un punto extremo de
 C . Prueba : se supone K^0 no es punto extremo , entonces existen
 $K^1, K^2 \in C$ ($K^1 \neq K^2$) , y $a \in (0, 1)$ de tal forma que

$$K^0 = a K^1 + (1 - a) K^2 \quad ; \quad 0 < a < 1$$

para $j = r+1, \dots, n+m$

$$0 = K_j^0 = a K_j^1 + (1 - a) K_j^2$$

entonces $K_j^1 = K_j^2 = 0$ para $j = r+1, \dots, n+m$, donde $K_j^1 \geq 0$,

$K_j^2 \geq 0$, para $j = 1, 2, \dots, r$

$$\sum_{j=1}^r k_j^1 P_j = P_0 \quad \dots (c)$$

$$\sum_{j=1}^r k_j^2 P_j = P_0 \quad \dots (d)$$

restando (c) de (d) se tiene $0 = \sum_{j=1}^r (k_j^1 - k_j^2) P_j$, pero P_j

$j = 1, 2, \dots, r$ son linealmente independientes; por lo que $k_j^1 = k_j^2$ para todo j , lo que es una contradicción ya que se supuso que $k^1 \neq k^2$, por lo que los r vectores son linealmente independientes y por tanto K^0 es un punto extremo de C .

6/ Si K^0 es un punto extremo de C entonces K^0 tiene a lo más m coeficientes diferentes de cero.

7/ C tiene un número finito de puntos extremos, lo que se infiere porque A tiene un número finito de bases las cuales están acotadas por $\binom{n-m}{m}$ si el rango de A es M .

8/ El punto extremo asociado con una base es único. Lo que se exhibe a continuación, supongase que existen $K^1, K^2 \in C$ ($K^1 \neq K^2$), tal que

$$\sum_{j=1}^m k_j^1 P_j = P_0, \quad \sum_{j=1}^m k_j^2 P_j = P_0, \quad \text{lo que implica que}$$

$$\sum_{j=1}^m (k_j^1 - k_j^2) P_j = 0 \quad \text{ó} \quad k_j^1 = k_j^2, \quad j = 1, 2, \dots, m$$

por lo que existe solo un punto extremo para una base dada.

9/ El conjunto de soluciones factibles C es un conjunto convexo.

Prueba: tomemos $a \in [0, 1]$, $z = ax + (1-a)y \geq 0$ por demostrar que $Az = b$, dado que $Ax = b$, $Ay = b$ y $a \in [0, 1]$ implica $Az = aAx + (1-a)Ay = ab + (1-a)b = b$, por lo tanto $z \in C$.

10/ Si el máximo $c'x$, $x \in C$, es igual a $c'x^0$ para algún $x^0 \in C$ entonces el máximo $c'x$, $x \in C$ es igual a $c'x^*$ donde x^* es un punto extremo de C .

Para probar 10, seán x^1, x^2, \dots, x^k puntos extremos de C entonces cualquier solución x^0 se puede escribir cómo $\sum_{i=1}^k K_i x^i$ con cada $K_i \geq 0$ y $\sum_{i=1}^k K_i = 1$, ahora bien $c'x^0 = c' \sum_{i=1}^k K_i x^i = \sum_{i=1}^k K_i (\max_k c'x^k) = \max_k c'x^k = c'x^*$ y en vista de que x^* e C y $c'x^0 \geq c'x$ para todo $x \in C$, se sigue que $c'x^0 \geq c'x^*$ por lo tanto $c'x^0 = c'x^*$.

Todos estos puntos se pueden resumir en :

- i/ Cada solución factible de C corresponde a un punto -- extremo de un conjunto convexo de soluciones factibles.
- ii/ Cada punto extremo esta asociado con m vectores linealmente independientes de el conjunto $P_j \quad j=1,2,\dots,n+m$.
- iii/ Hay un punto extremo en el cual la función objetivo $z = c'x^0$ toma su valor máximo .

11/ $K \in \mathbb{R}^n$ es un cono si $ky \in K$ para algun $y \in K$, $k \in \mathbb{R}$, $k \geq 0$

12/ Un conjunto K contenido en \mathbb{R}^n es un cono convexo si $ax + (1-a)y \in K$ ($a \in [0, 1]$) $x, y \in K$.

13/ $K \in \mathbb{R}^n$ tal que $K^* = \{ y^* \in \mathbb{R}^n \mid y^* y \geq 0, \text{ para todo } y \in K \}$ se dice que K^* es el cono polar de K .

14/ $V \in \mathbb{R}^n$ es una variedad lineal si $x, y \in V$, tal que $ax + (1-a)y \in V$ para toda $a \in \mathbb{R}$.

15/ $H \in \mathbb{R}^n$ es un hiperplano si es una variedad lineal de dimensión $n-1$.

16/ Un conjunto C se le llama poliedro si C esta definido por la intersección de un número finito de semiespacios cerrados de la forma, $C = \{ Ax \geq (\text{Respectivamente } \leq) b \}$.

3.0 ALGORITMO DEL METODO SIMPLEX REVISADO

Considere el problema .

$$\text{Máximizarse } Z = C' x$$

sujeto a.

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned} \quad \dots(a)$$

el cual es equivalente a .

$$\text{Max } Z$$

sujeto a.

$$\begin{aligned} Z - C'x &= 0 \\ Ax &= b \\ x &\geq 0 \end{aligned} \quad \dots(b)$$

Después de agregar las variables de holgura y artificiales apropiadas se tendrán $n-1$ variables y $m-1$ restricciones -- con una base de rango $m-1$ y el problema se rescribe como .

$$\text{Max } Z$$

sujeto a.

$$\begin{bmatrix} 1 & & -C' \\ 0 & & A \end{bmatrix} \begin{bmatrix} Z \\ x \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix} \quad \text{tal que } x \geq 0 \quad \dots(c)$$

Si reordenamos las columnas de A de tal forma que las primeras m columnas formen una base de rango m , entonces una base de rango $m-1$ será

$$B = \begin{bmatrix} 1 & & -C'_B \\ 0 & & B_m \end{bmatrix} \quad \dots(d)$$

donde C'_B es un vector de m componentes en el cual el j -ésimo elemento es el coeficiente del beneficio original de la variable básica asociada con el j -ésimo renglón, y la inversa

de B es

$$B^{-1} = \left[\begin{array}{c|c} 1 & C_B' B_m^{-1} \\ \hline 0 & B_m^{-1} \end{array} \right] \dots (e)$$

por lo que conociendo B^{-1} se determina la siguiente tabla

$$B^{-1} \left[\begin{array}{c|c|c} 1 & -C' & 0 \\ \hline 0 & A & b \end{array} \right] = \left[\begin{array}{c|c|c|c|c|c} 1 & Z_1 - C_1 & Z_2 - C_2 & \dots & Z_j - C_j & \dots & C_B' X^0 \\ \hline 0 & X_1 & X_2 & & X_j & & X^0 \end{array} \right] \dots (f)$$

en el que

$$B_m^{-1} \bar{P}_j = X_j \dots (g)$$

$$Z_j = C_B' X_j = C_B' B_m^{-1} \bar{P}_j \dots (h)$$

\bar{P}_j representa la j -ésima columna de A y P_j representa la j -ésima columna de (c), así la j -ésima columna de (c) se puede transformar por B^{-1} a .

$$B^{-1} P_j = B^{-1} \left[\begin{array}{c} -C_j \\ \bar{P}_j \end{array} \right] = \left[\begin{array}{c|c} 1 & C_B' \cdot B_m^{-1} \\ \hline 0 & B_m^{-1} \end{array} \right] \left[\begin{array}{c} -C_j \\ \bar{P}_j \end{array} \right] = \left[\begin{array}{c|c} -C_j & + C_B' B_m^{-1} P_j \\ \hline & B_m^{-1} \bar{P}_j \end{array} \right]$$

$$\left[\begin{array}{c} Z_j - C_j \\ X_j \end{array} \right] \dots (i)$$

$$y \quad B^{-1} \left[\begin{array}{c} 0 \\ b \end{array} \right] = \left[\begin{array}{c} C_B' X^0 \\ X^0 \end{array} \right] = \left[\begin{array}{c} Z^0 \\ X^0 \end{array} \right] \dots (j)$$

De esto se infiere que si los elementos originales de A, b, C y algún B^{-1} tendremos toda la información importante necesaria para determinar que vectores entran y salen de la base más aun, y los valores que hayan permanecido de esta tabla no sera necesario que seán calculados.

De (i) el menor en valor negativo de $Z_j - C_j$ estara -
 determinado por multiplicar el primer renglón de B^{-1} por cada -
 $\begin{bmatrix} -C_j \\ \bar{P}_j \end{bmatrix}$ definido en (c), llamemoslo .

$$(B^{-1})_o \begin{bmatrix} -C_j \\ \bar{P}_j \end{bmatrix} = Z_j - C_j \quad \dots(k)$$

Aunque el problema de cómo calcular B^{-1} permanece por lo que nos revocamos al sistema $[B \mid I]$ en el cual la inversa de B puede ser calculada utilizando eliminación gaussiana (*) esto es por medio de las operaciones elementales en los renglones de los elementos de B pueden ser transformadas a la matriz identidad .

$$[B \mid I] \longrightarrow [I \mid D]$$

En el que en el ultimo sistema equivalente D representa a B^{-1} , si ahora consideramos cualquier tabla y suponemos que - pivoteamos sobre X_{rk} . Sea y_{ij} el (ij)-esimo elemento de B^{-1} , la inversa actual, despues el r-esimo renglon es dividido por el elemento pivote el k-esimo vector de A y el j-esimo vector de B^{-1} sera transformado .

$$\begin{bmatrix} \vdots \\ X_{ik} \\ \vdots \\ X_{rk} \\ \vdots \end{bmatrix} \longrightarrow \begin{bmatrix} \vdots \\ X_{ik} \\ \vdots \\ 1 \\ \vdots \end{bmatrix}, \quad \begin{bmatrix} \vdots \\ y_{ij} \\ \vdots \\ y_{rj} \\ \vdots \end{bmatrix} \quad \begin{bmatrix} \vdots \\ y_{ij} \\ \vdots \\ y_{rj}/X_{rk} \\ \vdots \end{bmatrix} \dots(1)$$

Para completar la transformación 'simplex los componentes X_{ik} para $i=1,2,\dots,m$ ($i \neq r$) debere ser eliminado, haciendo los correspondientes componentes de (1) para $i \neq r$, serán reemplazados por.

$$Y_{ij} = -(y_{rj}/X_{rk})X_{ik} + y_{ij} \quad \dots(m)$$

(*) Ver algebra lineal de Howard Anton

Por esto en la nueva inversa la j -ésima columna es actualizada cómo sigue .

$$Y_{ij} = y_{ij} - (X_{ik}/X_{rk})y_{rj} \quad i \neq r \quad \dots(m.1)$$

$$Y_{rj} = (1/X_{rk})y_{rj} \quad \dots(m.2)$$

Esto implica que existe alguna matriz E tal que .

$$E B_y^{-1} = B_y^{-1} \quad \dots(n)$$

Siendo la matriz E de la forma siguiente .

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & \\ \vdots & & \ddots & \\ \vdots & & & 0 \\ \vdots & & & \vdots \\ \vdots & & & 1 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & \vdots \end{bmatrix} \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \dots & 0 \\ \vdots & \dots & \vdots \\ 0 & \dots & \vdots \\ 1 & \dots & \vdots \\ \vdots & \dots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \quad \dots(R)$$

Entonces en vista de (R) , la j -ésima columna de B_y^{-1} puede ser transformada a (m.1) y (m.2) si y solo si .

$$n_t = \frac{1}{X_{rk}} \begin{bmatrix} -X_{0k} \\ -X_{1k} \\ \vdots \\ 1 \\ -X_{r-1,k} \\ \vdots \end{bmatrix} = \frac{1}{X_{rk}} \begin{bmatrix} -(Z_k - C_k) \\ -X_{1k} \\ \vdots \\ 1 \\ -X_{r-1,k} \\ \vdots \end{bmatrix} \quad \dots(o)$$

Ahora bien si B es la matriz identidad inicial en la tabla 0 , entonces .

$$B_0 = I = B_0^{-1} \quad , \text{Ademas}$$

$$B_t^{-1} = E_t (B_t^{-1}) = E_t E_{t-1} (B_{t-2}^{-1})$$

$$\vdots \\ = E_t E_{t-1} E_{t-2} \dots E_1 I \quad \dots(p)$$

Por esa razón una sucesión de E_t matrices es suficiente para calcular B_t^{-1} para alguna t .

Notese que .

$$E_t = [e_0, e_1, \dots, e_{r-1}, n_t, e_{r+1}, \dots, e_m]$$

donde e_i es el vector unitario con un 1 en la $(i+1)$ coordenada, por eso E_t es reconstruido de un conocimiento explícito de n_t y r , y solo se necesita conocer (n_1, n_2, \dots, n_t) y (r_1, r_2, \dots, r_t) para calcular B_t^{-1} .

4.0 ANALISIS DE SENSIBILIDAD

En muchos casos nos preguntaremos en cuanto un -- coeficiente de beneficio c_w puede variar sin afectar la solución, en particular sobre que rango c_w varia sin alterar la presente solución basica, esta pregunta es frecuentemente --- solicitada dado que c_w se puede ver cómo el beneficio por unidad de una comodidad especifica.

Considerese el w -esimo componente de C el cual es --- intercambiado por $c_w + \&$, para algun $\&$, es decir .

$$c_j^* = c_j, \quad c_w^* = c_w + \& \quad j \neq w \quad \dots(a)$$

Para determinar el rango sobre el cual varia c_w , se considerán dos casos .

1/ X_w es no-basica .

$$Z_j - c_j^* \geq 0 \quad \text{para todo } j \neq w \quad \dots(b)$$

Y esto ya que $Z_j = C_B^* B_m^{-1} \bar{P}_j$ y $c_j^* = c_j$ para $j \neq w$ por lo que $Z_j - c_j^*$ en (b), no se alterara al intercambiar por -- c_w . Si $c_w^* = c_w - \&$ es tal que

$$Z_w - c_w^* \geq 0 \quad \dots(c)$$

y entonces tendremos una solución óptima, de (c), & deberá ser de tal forma que

$$Z_w - (c_w + \&) \geq 0$$

por lo que se infiere que $-\& \geq -Z_w + c_w$, o

$$\& \leq Z_w - c_w \quad \dots (d)$$

luego entonces si & es seleccionado de tal forma que

$$\& \in (-\infty, Z_w - c_w]$$

la solución básica previa es aun óptima.

Podría notarse que x_w entrara a la solución cuando $\& \geq Z_w - c_w$.

2/ x_w es básica.

Cambiando c_w por $c_w + \&$ en la tabla 0 y actualizando la tabla al multiplicar por B_{opt}^{-1} tendremos

$$B_{opt}^{-1} \begin{bmatrix} -c_0 & -c_1 & \dots & -c_w^* & \dots & 0 \\ \bar{p}_0 & \bar{p}_1 & \dots & \bar{p}_w & \dots & b \end{bmatrix} \quad B_{opt}^{-1} = \begin{bmatrix} 1 & C_B' B_m^{-1} \\ 0 & B_m^{-1} \end{bmatrix}$$

Otra vez, $Z_j - c_j \geq 0$ para todo $j \neq w$ donde el único cambio de la tabla original es c_w^* . Para $j = w$, $Z_j - c_j^* = Z_j - c_j - \& = -\&$.

La tabla actualizada no está en la forma canónica dado que la w -ésima columna es ahora de la forma

$$\begin{bmatrix} -\& \\ 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Para obtener la forma canónica multiplicaremos el renglón básico por & y sumaremos este renglón a la función objetivo.

Si todos los nuevos coeficientes en el renglón 0 son no-negativos la solución básica es aun óptima y el único cambio es el valor en la función objetivo. Determinemos el rango que los valores c_w^* -- pueden asumir mientras se preserva la optimalidad, para algun $\&$ dado y renglón básico i' , los nuevos coeficientes actualizados -- $Z_j - c_j$ para las variables no-básicas serán

$$(Z_j - c_j) - \& (x_{i',j}) \quad \dots (e)$$

las básicas $Z_j - c_j$ son aun cero, por lo que es suficiente --- considerar solamente las columnas no-básicas actualizadas.

SI $x_{i',j} < 0$, entonces en (e) las restantes son no negativas cuando

$$\& x_{i',j} \geq - (Z_j - c_j) \quad \text{para } j \text{ no-básica}$$

$$\& \leq - (Z_j - c_j) / x_{i',j}$$

Esto implica que el valor máximo para $\&$ puede tenerse sin cambiar la presente solución básica y es el infimo de las -- desigualdades anteriores, es decir. (*)

$$\&_{\max} = \text{infimo} \left\{ -(Z_j - c_j) / x_{i',j} \mid x_{i',j} < 0, j \text{ e no-básicas} \right\} \dots (f)$$

si $x_{i',j} > 0$, entonces en (e) las restantes son no negativas cuando

$$\& x_{i',j} \geq - (Z_j - c_j) \quad \text{para } j \text{ no-básica}$$

o $\&_{\min} \geq - (Z_j - c_j) / x_{i',j}$. Por eso el máximo decremento -- en c_w ocurre cuando /

$$\&_{\min} = \text{supremo} \left\{ -(Z_j - c_j) / x_{i',j} \mid x_{i',j} > 0, j \text{ e no-básicas} \right\} \dots (g)$$

Un cambio en el vector de recursos. Asumimos que el vector de recursos tiene su r -ésima componente cambiada, en particular, considerese $b_r^* = b_r + \&$, para verificar la optimalidad se necesita multiplicar únicamente este vector por B_{opt}^{-1} , en el que opt -- denota óptimo,

(*) donde infimo denota la mayor de las cotas inferiores y supremo la mínima de las cotas superiores.

$$B_{opt}^{-1} b^* = B_{opt}^{-1} \left(b + \begin{bmatrix} 0 \\ \vdots \\ \delta \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right)$$

$$= x^0 + \delta (B_{opt}^{-1})^r$$

Donde r denota la r -ésima columna de B^{-1} .

Para determinar el rango sobre b_r^* , se debe determinar cuando al actualizar el vector del lado derecho resulta mayor - que o igual a cero, con $\delta (B_{opt}^{-1})^r = \delta s$, entonces $x_i^0 - \delta s_i \geq 0$ para $i = 1, 2, \dots, m$, o

$$\delta s_i \geq -x_i^0 \quad \text{para } i = 1, 2, \dots, m \quad \dots (h)$$

1/ Si $s_i < 0$, δ debera ser tal que $\delta \leq -x_i^0 / s_i$ para toda i con $s_i < 0$, es decir el máximo incremento en $b_r^* = b_r + \delta$ esta restringido por

$$\delta_{max} = \infimo \left\{ -x_i^0 / s_i \mid s_i < 0 \right\} \quad \dots (i)$$

si δ_{max} es vacio el nivel puede ser incrementado sin cota sin - sin estar cambiando las variables basicas. Note que el valor - de la solución optima esta cambiando desde que el vector derecho esta cambiando.

2/ Si $s_i > 0$ entonces de (h), $\delta \geq -x_i^0 / s_i$, lo cual implica que el máximo decremento en b_r ocurre cuando

$$\delta_{min} = \supremo \left\{ -x_i^0 / s_i \mid s_i > 0 \right\} \quad \dots (j)$$

Cambios en los coeficientes de las restricciones.

Cuando los coeficientes de la matriz de restricciones son cambiados, más bien interesa su rango de variación ya -- conocida la solución para ver si se tiene la misma solución -- optima, por lo que del última tabla simplex, se considerara -- que los coeficientes a_{iw} ahora tienen un nuevo valor es decir

$$a_{iw}^* = a_{iw} + \delta$$

1/ Si x_w es no-básica, entonces $x^0 = B^{-1} b \geq 0$ y x^0 es todavía una solución factible, para verificar la optimalidad solo se requiere verificar los $Z_j - c_j$. Si $j \neq w$ se tiene que

$$Z_j^* - c_j = C'_B B_m^{-1} \bar{p}_j - c_j = Z_j - c_j$$

es decir para $j \neq w$ el valor de $Z_j - c_j$ permanece, para $j = w$

$$Z_w^* - c_w = C'_B B_m^{-1} \bar{p}_j^* - c_w \quad \dots(k)$$

la única entrada en el renglón 0 se puede verificar que es (k) si (k) es no-negativo la solución es todavía óptima, aunque si (k) es negativo se debe continuar el algoritmo simplex con x_w entrando en la solución.

2/ Si x_w es básica, la matriz básica B^* es diferente y por tanto $(B^*)^{-1}$ es distinta de B_{opt}^{-1} o bien esta no puede existir considerese que $(B^*)^{-1}$ no existe, por lo que un camino para retornar a la forma canónica es descomponer la w -ésima columna.

$$P_w^* = P_w + \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \vdots \end{bmatrix} = P_w + P_w$$

Sumaremos a la tabla original un nuevo vector artificial, P_w , en la tabla final el vector actualizado es $B^{-1} P_w$ el cual corresponde a una variable no-básica, por lo que se puede proceder como en el diseño 1.

Si $(B^*)^{-1}$ existe, se tienen 3 condiciones

- i/ $(x^*)^0 = (B^*)^{-1} b \geq 0$
ii/ $(x^*)^0 \geq 0$, pero $Z_j^* - c_j \not\geq 0$ para todo j
iii/ $(x^*)^0 \geq 0$ y $Z_j^* - c_j \geq 0$ para todo j

En i se usa el dual simplex o se introduce una función pseudo-objetivo para obtener la forma canónica, en ii continuar el algoritmo simplex, iii implica que la solución es óptima.

5. EL DUAL DE UN PROBLEMA DE PROGRAMACIÓN LINEAL

A cada problema de programación lineal le corresponde un segundo problema de programación lineal llamado el dual .

Cuando el problema inicial o primal trata la maximización (minimización) de una función objetivo , el problema dual trata la minimización (maximización) de una función objetivo . El número de variables en el problema dual es igual al número de restricciones en el problema primal . La siguiente tabla sintetiza la correspondencia entre el problema primal y el problema dual.

Problema Primal	Problema dual
Constantes en las restricciones.	Coeficientes en la función objetivo .
coeficientes en la función objetivo .	Constantes en las restricciones .
Signos de las desigualdades en las restricciones .	Signos de desigualdad invertidos .
i-ésima fila de coeficientes en las restricciones .	i-ésima columna de coeficientes .
En notación matricial .	

$$\begin{aligned} \text{Max } Z &= Cx \\ \text{sujeto a } & \end{aligned}$$

$$Ax \leq b$$

$$x \geq 0$$

$$\begin{aligned} \text{Min } W &= by \\ \text{sujeto a } & \end{aligned}$$

$$yA \geq C$$

$$y \geq 0$$

La solución del problema dual también nos proporciona la solución del problema primal ; si $m < n$, la resolución del problema dual requiere menos cálculos y por consiguiente es usualmente preferible , La resolución del problema dual nos provee de precios implícitos que nos llevarían a una asignación eficiente de los recursos . La siguiente tabla sintetiza la correspondencia entre las soluciones del primal y del dual .

Problema Primal

Valor de la función objetivo.
 Criterios para las variables primarias .
 Criterios para las variables sueltas o de holgura .
 Soluciones para las variables primarias .
 Soluciones para las variables sueltas .

Problema Dual

Valor de la función objetivo .
 Soluciones para las variables sueltas .
 Soluciones para las variables primarias .
 - Criterios para las variables sueltas .
 - Criterios para las variables primarias .

Luego el problema dual es completamente simétrico al - problema primal y la solución de un problema da información -- completa en lo que se refiere a la solución del otro problema .

El problema dual nos provee de una rutina mucho más - simple desde el punto de vista de los cálculos que el problema primario cuando la tabla simplex del dual tiene menos filas --- que la tabla simplex del primal .

6.0 INTRODUCCION A TEMPO / MPS / ALL

El paquete TEMPO/MPS/ALL esta constituido de un grupo de algoritmos , de programación matemática . En cuyo diseño se a considerado lo siguiente :

- 1/ Modularidad .- Debido a que cada procedimiento esta asociado con un paso bién definido en la solución de un problema .
- 2/ Procesamiento eficiente .- Se tiene un procesamiento eficiente debido a que se utilizarón las mejores tecnicas en el desarrollo de este paquete .
- 3/ Control .- El paquete provee de un extenso control al usuario , para llevar a cabo operaciones .
- 4/ Diseño flexible .- El paquete provee facilidades al usuario de aumentar su capacidad , en la resolución de problemas de programación matemática .

Los procedimientos del paquete están clasificados de acuerdo a sus funciones en los siguientes :

- 1/ Procedimientos de entrada .- Estos procedimientos se utilizán para definir los problemas de programación lineal y obtener una representación interna de los mismos que se procese eficientemente .
- 2/ Procedimientos de optimización .- Estos procedimientos se utilizán para resolver los problemas de programación matemática .
- 3/ Procedimientos de salida .- Estos procedimientos dán información acerca de los problemas así como su solución .
- 4/ Procedimientos de inicio avanzado .- Estos procesos determinan una solución mejorada par resolver los problemas .
- 5/ Procedimientos de análisis post-óptimo .- Estos procedimientos se utilizán para determinar los cambios en la solución de los problemas , cuando estos són alterados .

Posteriormente se verán los procedimientos más utilizados así como la descripción de resultados que con ellos se obtiene .

- 6/ Procedimientos de preservación de bases .- Estos -- procedimientos se utilizan para almacenar y posteri -
ormente recuperar las bases asociadas con la solución
de problemas .
- 7/ Procedimientos de utileria .- Estos procedimientos se
utilizán para realizar diversas funciones , sobre el
problema de optimización que se resuelve .

El paquete resuelve problemas de programación mate
mática de la siguiente forma : ■■

- 1/ Problemas de programación lineal .
- 2/ Programación entera .
- 3/ Programación binaria .
- 4/ Programación mixta .
- 5/ Programación separable .
- 6/ Problemas de acotamiento superior generalizado .

Primero se especificara el formato utilizado en la
descripción de problemas de optimización que resuelve el paquete,
seguido de la descripción de instrucciones con que se indica
al paquete el problema y posteriormente la descripción de ---
algunos de los procedimientos utilizados por el paquete asi -
como la indicación de los resultados que emiten .

■■ Solo se considerarán aquí los primeros cuatro tipos de
problemas , para los restantes se sugiere ver ■■ .

En esta sección se especifica el formato utilizado - por el paquete de programación matemática TEMPO/MPS/ALL , el - cual para resolver un problema utiliza dos archivos de datos .

1/ CARD .- En este archivo están contenidas las instrucciones de procedimiento para resolver el problema [C] .

2/ CARDIN .- En este archivo están contenidos los datos del - problema , los cuales se dividen en cinco secciones : ROWS , - COLUMNS , RHS , RANGES , BOUNDS ; si rangos o cotas no se --- utilizán , se omiten las secciones de RANGES y BOUNDS . Cada -- tarjeta de datos consta de 80 columnas y se divide en 6 campos un * en la columna uno indica se trata de una tarjeta de -- comentario . [C1]

FORMATO DE LOS DATOS DE ENTRADA
(formato MPS)

CAMPO	1	2	3	4	5	6
	1 2 3	5-12	15-22	25-36	40-47	50-61
	N A M E	DATOS				
	*	Tarjeta de comentario				
	R O W S N	Nombre de la función objetivo (E para restricción de igualdad ; G para restricción de mayor o igual ; L para restricción de menor o igual)				
	C O L U M N S	Nombre de var.	Nombre de rest.	Coef.	Nombre de rest.	Coef.
	R H S	Nom. del vector b	"	"	"	"
	R A N G E S	Nom. de rangos	"	"	"	"
	B O U N D S L O	Nom. de cotas	Nom. de var.	"		
	E N D A T A					

Tabla 2.1

[C] Estas tarjetas de procedimiento se verán posteriormente .

[C1] La descripción de estas cinco secciones , se encuentra en las páginas siguientes .

Organización de los datos de entrada del problema de programación matemática , vease la tabla 2.1 .

1/ Una tarjeta de NAME sera la primer tarjeta en los datos y una de ENDATA al final , ya que con esta se indica fin de -- archivo de datos , la tarjeta de NAME también tiene en el campo numero 3 un nombre especificado por el usuario . Las secciones de ROWS , COLUMNNS , y RHS son necesarias , y las secciones -- RANGES , BOUNDS serán opcionales y no es necesario declararlas o especificarlas .

2/ La sección de ROWS es en la que se define el tipo de res-- tricción de cada renglón , y se utilizan 4 indicadores para - especificar el tipo de restricción en el modelo , seguido por algun nombre especificado por el usuario en el campo dos;

- a/ N Renglón de la función objetivo .
- b/ E Igualdad en la restricción .
- c/ G Mayor o igual en la restricción .
- d/ L Menor o igual en la restricción .

3/ La sección de COLUMNNS , es en la que se especificán los - coeficientes de las restricciones y los nombres de cada varia- ble , el nombre de la columna es dado en el campo dos; , y los nombres de los renglones (restricciones) estarán especificados en los campos 3 y 5 , en los campos 4 y 6 se declarán los --- coeficientes correspondientes , El sistema trata cómo cero a - los coeficientes omitidos en el modelo , es decir solo es --- necesario declarar los coeficientes que seán diferentes de -- cero.

4/ La sección de RHS , es donde se especifica el vector de -- recursos (o vectores) empezando en el campo dos , el formato es el mismo conque fue definido columns , aunque un nombre para el vector de recursos debere ser declarado y esto para cada - vector de recursos . con que se desea resolver el modelo .

5/ La seccion de RANGES , es utilizada para condensar los dat os de entrada .

Esto se hace cuando una restricción es mayor o igual y menor o igual , El renglón original necesita ser especificado en la sección de rows y columns solamente con uno de sus limit es superior o inferior $b(i)$ especificado en la sección de rhs.

EL rango es utilizado cómo se especifica en la siguiente tabla , donde $r(i)$ es el rango sobre el renglón i dado en la sección de rangos , si $r(i)$ es negativo para un renglón G o un renglón L , se utiliza el valor absoluto.

Tipo de renglón	Signo de $b(i)$	Limite superior que resulta en la rest.	Limite inferior q' resulta en la rest.
G	+	$b(i) + r(i)$	$b(i)$
L	+	$b(i)$	$b(i) - r(i)$
E	+	$b(i) + r(i)$	$b(i)$
E	-	$b(i)$	$b(i) - r(i)$

Tabla 2.2

6/ La sección de BOUNDS , es en donde se especifica que varia bles estarán acotadas , si esta sección no se especifica se -- considera que todas las variables son mayor o igual que cero , se utilizarán 6 indicadores para especificar el tipo de cota que sera impuesto sobre la actividad .

- a/ LO Cota inferior.
- b/ UP Cota superior.
- c/ FX Valor fijo.
- d/ MI Cota inferior es - infinito.
- e/ PL Cota superior es + infinito.
- f/ FR Variable libre (- infinito a + infinito)

En los campos 3 y 4 se especifica el nombre de la -- variable que será acotada y su cota finita asociada respectiva mente , dejar el campo 4 en blanco si la cota es infinito. El campo 2 identifica un nombre asociado con un conjunto de cotas esto es , puede haber más de un conjunto de cotas declaradas en la sección de bounds , lo cual es similar a la opción de definir varios vectores de recursos en la sección rhs .

Dentro de las secciones de ROWS y COLUMNS las opciones 'MARKER' son provistas con el siguiente formato.

- i/ El campo 1 permanece en blanco excepto cuando es utilizado para datos GUB .
- ii/ El campo 2 identifica el nombre de el marker , el cual - debe ser distinto de los nombres de rows y columns .
- iii/ El campo 3 contiene la palabra 'MARKER' (incluyendo los apostrofos) .
- iv/ El campo 4 permanece en blanco .
- v/ El campo 5 contiene las palabras claves :
 - a/ 'INTORG' indica que las variables que están en las tarjetas que siguen , son enteras hasta donde se -- encuentre 'INTEND!' .
 - b/ 'BIVORG' indica que las variables que están en las tarjetas que siguen , son binarias (solo toman 'el - valor cero o uno) hasta donde se encuentre 'BIVEND'.
 - c/ 'SEPORG' y 'SEPEND' son reservadas para programación separable.
 - d/ 'GUBORG' y 'GUBEND' se reserván para definir renglones tipo GUB.

las palabras incluyen los apostrofes , los nombres asignados a marker , renglones , variables , terminos del vector de recursos , cotas , y rangos son asignados por el usuario y son a lo más de ocho caracteres .

- vi/ El campo 6 permanece en blanco .

La forma de ejecutar el programa por tarjetas es :

```

?JOB LINEAL;USER=SF82/MP;CLASS=5;BEGIN
?RUN*SERVICIO/MPS/ALL;DATA CARD
      .
      Instrucciones      } Tarjetas de
      .                  } procedimiento
      .
?DATA CARDIN
      .
      Tarjetas de datos
      .
?END JOB
  
```

Las instrucciones de procedimiento o programa de -- control utilizadas por el archivo.CARD utiliza dos tipos de - instrucciones de procedimiento :

- 1/ De asignación de nombres para cada uno de los ----- componentes del problema .
- 2/ De procedimiento , para el proceso a seguir en la --- solución del problema de optimización que se tiene .

Las instrucciones de asignación son las siguientes: **CC**

ZDATA = "DATOS"

Esta instrucción se utiliza para asignar un nombre al conjunto de datos ; debe aparecer con el mismo nombre con el que se identifica el conjunto de datos en el archivo CARDIN (tarjeta de NAME) , en este caso se asignó el nombre DATOS .

ZNAME = "LINEAL"

Esta instrucción se utiliza para asignar un nombre al problema ; en este caso se asigno el nombre LINEAL .

ZOBJ = "FO"

Esta instrucción se utiliza para asignar un nombre a la función objetivo , el cual debe coincidir con el nombre de la restricción etiquetada con N en la sección de ROWS . En este caso se asigno el nombre FO a la funcion objetivo .

ZRHS = "TI"

Esta instrucción se utiliza para asignar un nombre al vector de recursos , el cual debe coincidir con el nombre utilizado en el segundo campo en la sección de RHS . En este caso fue asignado el nombre TI al vector de recursos.

ZRNGST = "RANGO"

Esta instrucción se utiliza para asignar un nombre al conjunto de rangos , de las restricciones el cual debe --- coincidir con el nombre utilizado en el segundo campo en la - sección de RANGES . En este caso se asigno el nombre RANGO a el conjunto de rangos .

ZBNDST = "COTA"

Esta instrucción se utiliza para asignar un nombre al conjunto de cotas , de las variables , el cual debe ----- coincidir con el nombre utilizado en el segundo campo en la

CC Los nombres asignados deberán ser a lo más de ocho caracteres.

sección de BOUNDS . En este caso se asigno el nombre COTA a -
el conjunto de cotas .

Las instrucciones de procedimientos más utilizados
del paquete son las siguientes :

BCDOU

Este procedimiento produce la impresión de los datos
del problema almacenado en ZPROF . Esto es convierte el problema
que se encuentra en un formato empacado en el archivo ZPROF
para una salida externa , en el formato de los datos de entrada.

PRIMAL

Este procedimiento obtiene una solución factible -
optima (si existe) utilizando para ello el metodo simplex --
revisado . Obtiene primero una solución factible y despues --
una solución optima . Si el problema es no factible o no ---
acotado termina su ejecución , indicandolo con un mensaje --
apropiado . Para determinar la solución utiliza una serie de
intercambios de vectores , llamados iteraciones .

PICTURE

Este procedimiento imprime una grafica de la matriz
de coeficientes tecnicos . En esta grafica los coeficientes
diferentes de ± 1 se convierten a un codigo alfabetico el cual
indica su magnitud . Cada rango de magnitud es una potencia -
de 10 . Coeficientes igual a 1 se imprimen como 1 . Un signo
(-) indica un coeficiente negativo . Los espacios en blanco
indican un coeficiente cero .

Se imprime una tabla también indicando el número de
elementos en cada rango de magnitud .

TRANCOL

Este procedimiento imprime el ultima tabla por ---
columnas . Transforma las variables no basicas , pre-multipli
cando cada variable por la inversa de la base actual y -----
imprime los resultados . Los coeficientes en el vector -----
transformado representan el efecto en la solución de utilizar
una unidad del vector .

INPUT

Este procedimiento se utiliza para leer los datos que describen el problema , transformandolos a un formato binario - empacado y almacenandolos en un archivo llamado ZPROF . A la - vez que se genera el archivo ZPROF se verifican errores ; las - estadísticas del problema se recopilan para utilizarlas ----- posteriormente . El conjunto de datos que se leen debe estar -- definido en ZDATA . El problema se etiqueta con el nombre con que se definio en ZNAME . Si un problema residente en ZPROF -- tiene el mismo nombre del nuevo problema , se elimina del ---- directorio de archivos ; aun cuando el area de almacenamiento - ocupada por el conjunto de datos eliminados no sea requerida .

El formato de los datos de entrada se describe en --- otra sección . Las tarjetas de NAME y ENDATA especificados en - la sección de entrada de datos se requieren . Asi como las seccio- nes de ROWS , COLUMNNS y RHS . Las impresiones que lleva a cabo este procedimiento son las siguientes :

- 1/ Mensajes de error se imprimen para cada error detectado; los mensajes son autoexplicativos e indican que acción - se debe seguir .
- 2/ Los nombres del vector de recursos , cotas y rangos se - imprimen .
- 3/ Las columnas , cotas y rangos que no poseen elementos se listan pero no se leen . Los renglones que no tienen -- elementos se listan pero no se eliminan .
- 4/ Si CHECK se especifica , los nombres de columnas duplica- das que no estén dentro de un grupo de 16 se listan pero no se eliminan .
- 5/ El número de renglones y columnas , el número de elementos diferentes de cero y la densidad de la matriz se imprimen .
- 6/ Si SUMMARY se especifica , se generán estadísticas para - cada renglón y columna .
- 7/ Los comentarios de el conjunto de datos se imprimen .

OUTPUT

Este procedimiento lista la solución actual incluyendo niveles de actividad , costos reducidos , costos originales , -- el vector de recursos y las actividades duales . La solución que se lista durante o despues de la optimización o un procedimiento de programación parametrica ; se imprime en tres secciones :

- 1/ Sección de identificación .
- 2/ Sección de ROWS (renglones) .
- 3/ Sección de Columns (columnas) .

La salida opcionalmente se puede escribir en el archivo de soluciones ZSOLF , para posteriormente utilizarla , para la impresión de un reporte o por SOLOUT .

La impresión que se emite es como sigue :

- 1/ En la sección de identificación :
 - 1.1/ Nombre del problema .
 - 1.2/ Nombre del conjunto de rangos , si se utilizán .
 - 1.3/ Nombre del conjunto de cotas , si se utilizán .
 - 1.4/ Nombre de la función objetivo .
 - 1.5/ Nombre del vector de recursos .
 - 1.6/ Estado de la solución . **■**
 - 1.6.1/ INFEASIBLE .- Infactible .
 - 1.6.2/ NON-OPTIMAL .- Factible pero no optimo .
 - 1.6.3/ OPTIMAL .- Optimo .
 - 1.7/ Iteration number .- Número de iteración .
 - 1.8/ Valor de la función objetivo .
- 2/ Las secciones de ROWS y COLUMNS tienen una estructura similar asi como en su contenido . Se lista una linea para cada variable . Los siguientes ocho datos de --- información se imprimén :
 - 2.1/ NUMBER .- Número interno asignado a la variable.
A la primer columna se le asigna el número (n+1) en donde n es igual al número del ultimo renglón.
A la segunda columna se le asigna el número (n+2), y así sucesivamente . Este número se puede --- utilizar para identificar a las variables .

■ El estado puede marcar un error , dependiendo del procedimiento utilizado antes de OUTPUT .

- 2.2/ NAME .- Nombre asignado a la columna o renglón .
- 2.3/ STATUS .- Dos caracteres indicarán el estado del renglón o columna en la solución . El estado de un renglón se refiere a la actividad y no a la holgura del renglón , los indicadores utilizados son los siguientes :
- 2.3.1/ BS .- En la base y factible .
- 2.3.2/ ** .- En la base y infactible .
- 2.3.3/ FR .- No básica , libre .
- 2.3.4/ EQ .- No básica , artificial o fija .
- 2.3.5/ UL .- No básica , con actividad en la cota superior .
- 2.3.6/ LL .- No básica , con actividad en la cota inferior .
- 2.3.7/ IV .- No básica , variable entera .
- 2.4/ ACTIVITY .- Valor de la actividad del renglón o columna en la solución .
- 2.5/ SLACK ACTIVITY/INPUT COST .- En la sección de ROWS este dato representa a la variable de holgura . En la sección de COLUMNS este dato es el costo de entrada.
- 2.6/ LOWER LIMIT .- Mínimo valor factible que la actividad puede tomar .
- 2.7/ UPPER LIMIT .- Máximo valor factible que la actividad puede tomar .
- 2.8/ DUAL ACTIVITY/REDUCED COST .- En la sección de ROWS este dato representa el actividad dual (variables duales o precios sombra) . En la sección de COLUMNS este dato representa el costo reducido de la columna. Los costos reducidos son conocidos como los d_j 's . Las actividades duales son conocidas como los --- multiplicadores simplex . El costo reducido de una variable es la tasa de incremento en la función --- objetivo , por el incremento unitario en la actividad de la variable .

La presencia de variables no básicas con costos reducidos igual a cero en la solución óptima indican que la solución óptima alternativa existe. Estas variables se identifican por medio de una A en la parte de la derecha de la tabla que se imprime.

Para todo renglón o columna compuesta, los valores que se imprimen para el renglón o columna básicos son los valores compuestos. Si la función objetivo es compuesta, las actividades duales, costos, y costos reducidos son compuestos para la función objetivo. Si el vector de recursos es compuesto, las cotas inferior y superior en la sección de ROWS serán para el vector de recursos compuesto.

Cantidades iguales a cero o aproximadas se imprimirán como un punto decimal.

RANGE

Este procedimiento produce un análisis de sensibilidad de la solución óptima . Esto es , determina el rango sobre el cual pueden variar , costos y el vector de recursos sin - que varíe la base óptima ; esto se efectúa por renglones y por columnas .

En la impresión de este procedimiento se indica el nombre de la función objetivo , el vector de recursos , el -- conjunto de rangos , y el conjunto de cotas . La impresión se efectúa en el siguiente orden :

- 1/ Renglones que se encuentran en su nivel mínimo .
- 2/ Columnas que se encuentran en su nivel mínimo .
- 3/ Renglones que se encuentran en un nivel intermedio .
- 4/ Columnas que se encuentran en un nivel intermedio .

La descripción de salidas para un modelo de ----- minimización es como sigue :

- 1/ Rango de variación de los renglones .

1.1/ El primer renglón indica el estado del renglón en la solución :

1.1.1/ Number .- Número interno del renglón .

1.1.2/ Row .- Nombre del renglón .

1.1.3/ Status .- Una clave indica el estado de actividad :

1.1.3.1/ BS .- La actividad se encuentra en un nivel intermedio .

1.1.3.2/ EQ .- La actividad se encuentra en un nivel fijo .

1.1.3.3/ UL .- La actividad se encuentra en su nivel superior .

1.1.3.4/ LL .- La actividad se encuentra en su nivel inferior .

1.1.4/ Activity .- El valor de la actividad del renglón , calculada como la diferencia - entre el vector de recursos y el vector de holgura .

1.1.5/ Slack Activity .- Actividad de la variable de holgura del renglón .

1.2/ A continuación se imprime utilizando dos líneas de ----- impresión . La línea superior indica el costo de la ---- actividad relacionada con decrementos de la actividad -- por el costo unitario de incremento . La línea inferior indica el costo de la actividad relacionada con incrementos de la actividad , por el costo unitario de decremento.

1.2.1/ Lower Limit .- Limite inferior de entrada para el - renglón . Especificado o implícito .

1.2.2/ Upper Limit .- Limite superior de entrada para el renglón . Especificado o implícito .

1.2.3/ Lower Activity .- Actividad del renglón que se --- puede decrementar a este nivel a un costo , por -- costo unitario de decremento . Un incremento en ---- este nivel tiene un costo unitario distinto .

1.2.4/ Unit Cost .- El cambio en la función objetivo por un decremento unitario en la actividad del renglón (línea superior) .

1.2.5/ Unit Cost .- El cambio en la función objetivo por un incremento unitario en la actividad del renglón (línea inferior) .

1.2.6/ Limiting Process .- Nombre del renglón o columna - que podría cambiar su estado , si el nivel de la - actividad de este renglón se decrementara por --- debajo de la actividad inferior . Si el renglón es básico, entra a la base ; en otro caso , sale de la base (línea superior) .

1.2.7/ Limiting Process .- Nombre del renglón o columna - que podría cambiar su estado , si el nivel de la - actividad de este renglón se incrementara por --- encima de la actividad superior . Si el renglón es básico, entra a la base ; en otro caso , sale de la base (línea inferior) .

1.2.8/ Status .- Estado del renglón (líneas superior e inferior) .

1.2.8.1/ LL .- El renglón o columna que sale o entra a la base se encuentra en su cota inferior .

1.2.8.2/ UL .- El renglón o columna que sale o entra a la base se encuentra en su cota superior .

2/ Rango de variación de las columnas .

2.1/ El primer renglón indica el estado de la columna - en la solución .

2.1.1/ Number .- Número interno de la columna .

2.1.2/ Row .- Nombre de la columna .

2.1.3/ Status .- Una clave indica el estado de --- actividad :

2.1.3.1/ BS .- En la base a nivel intermedio.

2.1.3.2/ EQ .- No básica , en un nivel fijo.

2.1.3.3/ UL .- No básica , en el límite --- superior .

2.1.3.4/ LL .- No básica , en el límite ---- inferior .

2.1.3.5/ FR .- No básica , libre (es decir puede tomar cualquier valor en los reales) .

2.1.4/ Activity .- Valor de la actividad de la --- columna .

2.1.5/ Input Cost .- Costo unitario de la variable como se especifico en la entrada de datos .

2.2/ A continuación se imprime utilizando dos líneas de impresión . La línea superior indica el costo de la actividad relacionada con decrementos de la actividad por el costo unitario de incremento . La línea inferior indica el costo de la actividad relacionada con -- incrementos de la actividad , por el costo unitario de decremento .

- 2.2.1/ Lower Limit .- Cota inferior de la columna .
- 2.2.2/ Upper Limit .- Cota superior de la columna .
- 2.2.3/ Lower Activity .- Nivel de actividad que --- se obtendria en caso de cambiar los costos - de entrada , a un costo superior .
- 2.2.4/ Upper Activity .- Nivel de actividad que -- se obtendria en caso de cambiar los costos - de entrada , a un costo inferior .
- 2.2.5/ Unit Cost .- Cambio en la función objetivo - por decremento unitario en la actividad de - la columna . El costo unitario no es valido para un decremento en la actividad, abajo del actividad inferior (línea superior) .
- 2.2.6/ Unit Cost .- Cambio en la función objetivo - por incremento unitario en la actividad de - la columna . El costo unitario no es valido para un incremento en la actividad , encima del actividad superior (línea inferior) .
- 2.2.7/ Upper Cost .- Costo más alto que la columna puede tener sin alterar su actividad . Si el costo se incrementa por debajo del costo --- superior , el nivel de actividad se podria - decrementar a la actividad inferior .
- 2.2.8/ Lower Cost .- Costo inferior que la columna puede tener sin alterar su actividad . Si el costo se decrementa por encima del costo -- inferior , el nivel de actividad se podria - incrementar a la actividad superior .
- 2.2.9/ Limiting Process .- Nombre del renglón o --- columna que podria cambiar su estado , si el nivel de la actividad de este renglón se --- decrementara por debajo de la actividad infe-
rior . Si la columna es basica , entra a la - base ; en otro caso sale de la base (línea - superior) .

2.2.10/ Limiting Process .- Nombre del renglón o columna que podría cambiar su estado , si el nivel de la actividad de esta columna se incrementara por -- encima de la actividad superior . Si la columna es básica , entra a la base ; en otro caso sale de la base (línea inferior) .

2.2.11/ Status .- Estado de la columna (líneas superior e inferior) .

2.2.11.1/ LL .- El renglón o columna que sale o entra a la base se encuentra en su -- cota inferior .

2.2.11.2/ UL .- El renglón o columna que sale o entra a la base se encuentra en su -- cota superior .

Si el problema es el de un modelo en que se requiere - maximizar en vez de minimizar , las siguientes palabras se ----- podrían substituir en la descripción anterior .

Modelo de Minimización	Modelo de Maximización
-----	-----
Costo	Beneficio
Decremento	Incremento
Incremento	Decremento
Inferior	Superior
Superior	Inferior

REVISE

76

Este procedimiento se utiliza para modificar un -- problema residente en el archivo ZPROF . Las modificaciones - se pueden llevar a cabo en cualquier renglón , columna , vector de recursos , conjunto de rangos o cotas ; en estas partes el problema puede ser modificado , borrado o insertado . El nuevo problema se agrega al archivo ZPROF . Si el nombre del nuevo problema se encuentra duplicado en ZPROF , se borra entonces el problema anterior .

SETUP

Este procedimiento utiliza los datos del problema - que están almacenados en el archivo ZPROF para generar la --- matriz de trabajo . Lee un problema de ZPROF , analiza las -- estadísticas del modelo y genera los archivos de trabajo que serán utilizados por procedimientos computacionales . Determina rangos de renglones y cotas de las columnas ; únicamente un - conjunto de rangos y cotas se deberá utilizar . Si rangos o - cotas no se especifican , entonces no se utilizarán en ZPROF . Todos los objetivos y recursos originales residentes en ZPROF están disponibles después del uso de este procedimiento .

Este procedimiento también verifica la validez de - marcas que indican programación entera , binaria , mixta , -- separable y descomposición . SETUP se debe utilizar a continuación de INPUT o REVISE para hacer que los datos del problema - estén disponibles para todos los procedimientos TEMPO .

SETUP determina el tipo interno de cada variable de la matriz de coeficientes técnicos , y el conjunto de rangos y cotas especificados . Imprime estadísticas indicando el -- número de renglones y columnas de cada tipo . El número de -- variables enteras que no son fijas también se especifica .

Indica también la memoria utilizada para la matriz de coeficientes técnicos y de su inversa , un mensaje especifica cuando la matriz está en disco o en memoria . La inversa es - almacenada en memoria hasta que la memoria disponible es --- excedida .

7.0 EJEMPLOS DE APLICACION

En esta sección se considerarán unos problemas de optimización y el proceso seguido para su solución mediante el paquete TEMPO/MPS/ALL .

1/ Considere el siguiente problema de programación lineal:

$$\text{Min } Z = -20x_1 - 10x_2 - x_3$$

Sujeto a .

$$3x_1 + 2x_2 + 10x_3 \leq 10$$

$$2x_1 + 4x_2 + 20x_3 \leq 15$$

$$x_1 \geq 0 ; x_2 \geq 0 ; x_3 \geq 0$$

Se resolvera este problema usando el paquete TEMPO/MPS/ALL a través de tarjetas . En la hoja de codificación anexa se muestra :

- 1/ Tarjetas de control del sistema .- Estas tarjetas son las que en la columna uno tienen un caracter "?" . Y se utilizán para accesar el paquete .
- 2/ Tarjetas de instrucciones sobre el problema .- En estas tarjetas se indica el proceso a seguir en la solución del problema . Siendo el siguiente :
 - 2.1/ Se le asigna el nombre de DATOS a el conjunto de datos del archivo CARDIN .
 - 2.2/ Se le asigna el nombre de MOLINO al problema .
 - 2.3/ Se le asigna el nombre de F OB a la función objetivo.
 - 2.4/ Se le asigna el nombre de TERMI a el vector de recursos.
 - 2.5/ Se llama el procedimiento de lectura , indicando que la lectura se hara de tarjetas .
 - 2.6/ Se llama el procedimiento BCDOUT para que se imprima el conjunto de datos del problema .
 - 2.7/ Se llama el procedimiento SETUP para reservar el area de trabajo del problema e indicar que se trata de un problema de minimización .
 - 2.8/ Se llama el procedimiento PRIMAL para resolver el problema primal .
 - 2.9/ Se llama el procedimiento OUTPUT para que se imprimán los resultados obtenidos por el paquete, de la solución del problema .

3/ Tarjetas de datos .- Estos datos son los coeficientes en el problema propuesto y se encuentran en el formato descrito en la sección anterior .

En la siguiente hoja , despues de la de codificación se muestra el proceso seguido para la solución del mismo --- problema mediante una terminal remota. El proceso seguido en la solución es el mismo, con la ventaja de que para los datos que describen el problema se utiliza un formato libre .

Una variación es el aparición de la instrucción -- ZPRINTER = .TRUE. , que es utilizada para que los resultados no solamente se emitán en la terminal , sino tambien en una impresora aparecerán los resultados .

Otra variación es el uso de un * para indicar que la variable utilizada es la misma que se transmitio en la ---- instrucción anterior .

La ejecución del paquete en una terminal remota se indica por medio de la instrucción EXIT .

Los resultados obtenidos por el paquete como solución al problema fuerón los siguientes :

Min Z = -66.66667

$x_1 = 3.333$ siendo basica factible

$x_2 = 0$ siendo no basica , con actividad
en la cota inferior

$x_3 = 0$ igual que x_2

Lograndose alcanzar el optimo en una iteración .

SF82/MP	(R)
RUN*SERVICIO/MPS/ALL	(R)
ZDATA = "DATOS"	(R)
ZNAME = "LINEAL"	(R)
ZRHS = "TI"	(R)
ZOBJ = "OF"	(R)
INPUT(REMOTE, SUMMARY)	(R)
ROWS	(R)
N OF	(R)
L R1	(R)
L R2	(R)
COLUMNS	(R)
X1 OF -20	(R)
* R1 3	(R)
* R2 2	(R)
X2 OF -10	(R)
* R1 2	(R)
* R2 4	(R)
X3 OF -1	(R)
* R1 10	(R)
* R2 20	(R)
ENDATA	(R)
RHS	(R)
TI R1 10	(R)
* R2 15	(R)
ENDATA	(R)
SETUP(MIN)	(R)
BCDOUT	(R)
PRIMAL	(R)
OUTPUT	(R)
EXIT	(R)

Nota.- (R) indica teclar return .

#B6700 126 CANDE 30.140 YOU ARE SCHED#(01(99)

DEFAULT PRINT DESTINATION=SITE

#SESSION 3028 17:32:25 01/14/81

#UNSERVIC/MPS/ALL

#PLANNING 3029

#?

#7700/B6700 TEMPC(28.600.000 - MARCH 71) 01/14/81 17:33:26

READY

#PRINTER = .TRUE.

READY

#DATA = "DATCS"

READY

#NAME = "MCLIND"

READY

#RHS = "TERMI"

READY

#OBJ = "LF"

READY

INPUT(REMOTE,SUMMARY)

--- INPUT ---

#

#UNS

#

CF

L R1

X

L R2

X

COLUMNS

X

X1 OF -20

X

* R1 3

X

* R2 2

X

X2 OF -10

X

* R1 2

X

* R2 4

X

X3 OF -1

X

* R1 10

X

* R2 20

X

RMS

X

TERM1 R1 10

X

* R2 15

X

OLD ENTRY MCLING DELETED ON ZPROF (OR ZSCLF)
 NEW ENTRY MCLING ENTERED ON ZPROF (OR ZSCLF)
 NUMBER OF ELEMENTS BY COLUMN ORDER

49 X13 X23 X33

NUMBER OF ELEMENTS BY ROW ORDER, EXCLUDING RHS, INCLUDING SLACK ELEMENT

1 N CF4 L R14 L R24

READY
 SETUP(MIN)

--- SETUP ---

READY
 PRIMAL

--- PRIMAL ---

--- INVERT ---

PROBLEM FEASIBLE.

ITER NO	OBJ. VALUE	NEG DJ	CUT	IN
1	-66.66667	3	2	49

EXIT CONDITION: OPTIMAL SOLUTION.

FUNCTION VALUE *

*66.66667

READY

OUTPUT

*** OUTPUT ***

SOLUTION STATUS * OPTIMAL
ROWS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	DUAL ACTIVITY
CF	BS	66.667	NONE	NONE	1.000
R1	UL	10.000	NONE	10.000	6.667
R2	BS	6.667	NONE	15.000	.

COLUMNS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	REDUCED COST
X1	BS	3.333	.	NONE	.
X2	LL	.	.	NONE	3.333
X3	LL	.	.	NONE	65.667

READY

EXIT

07700/06700 TEMPC TERMINATED.

#ET=4124.4 FT=7.4 LC=5.0

2/ Considere el siguiente problema de programación entera :

$$\text{Max } Z = 101x_1 + x_2 + 12x_4 + 7x_5 + 8x_6 + 6x_7 + 3x_8$$

Sujeto a .

$$6x_1 + .8x_2 + x_3 + 10x_4 + 5x_5 + 6x_6 + 5x_7 + 3x_8 \leq 25, 30$$

$$x_1, x_3, x_4, x_5, x_6, x_7, x_8 = 0, 1$$

$$x_2 \text{ e } z^*$$

Se resolvera este problema usando el paquete TEMPO/MPS/ALL a través de una terminal remota en forma interactiva . En la hoja anexa se muestra :

- 1/ Proceso seguido en la solución del problema :
 - 1.1/ Instrucción para acceder el paquete .
 - 1.2/ Se indica que habra de imprimir los resultados no solo en la terminal , sino tambien en la -- impresora .
 - 1.3/ Se le asigna el nombre de ENTERA al problema.
 - 1.4/ Se le asigna el nombre de datos al conjunto de datos del problema .
 - 1.5/ Se le asigna el nombre de OF a la función objetivo.
 - 1.6/ Se le asigna el nombre de TI a el vector de recursos .
 - 1.7/ Se llama el procedimiento INPUT de lectura indicando que los datos serán suministrados por medio de la terminal .
 - 1.8/ Se empiezan a teclear los datos del problema en formato libre (Los que se utilizaban en el archivo CARDIN).
 - 1.9/ Se llama el procedimiento SETUP para reservar el area de trabajo del problema e indicar que se trata de un problema de maximización .

■ Solo un dato por linea se puede indicar .

- 1.10/ Se llama el procedimiento PRIMAL para resolver el problema primal .
- 1.11/ Se llama el procedimiento MXINT para resolver el problema de programación entera (Este procedimiento utiliza el algoritmo de BRANCH & BOUND).
- 1.12/ Se llama el procedimiento OUTPUT para que se impriman los resultados obtenidos por el paquete, de la solución del problema , utilizando para el vector de recursos el coeficiente 25 .
- 1.13/ Se declara el nombre nuevo que tendrá el vector de recursos TI2 y que es en el que se utiliza el coeficiente 30.
- 1.14/ Se vuelve a llamar el procedimiento MXINT para resolver el problema de programación entera con el nuevo vector de recursos .
- 1.15/ Se utiliza la instrucción RETURN para transferir el control a la instrucción siguiente .
- 1.16/ Se vuelve a llamar el procedimiento OUTPUT para que se imprimán los resultados , utilizando el segundo vector de recursos , en la solución del problema .
- 1.17/ Se utiliza la instrucción EXIT para terminar la ejecución del paquete .

Los resultados obtenidos por el paquete como solución al problema fueron los siguientes :

Utilizando en la restricción primero el coeficiente 25 se tuvo la siguiente solución :

Max Z = 126

$$x_1 = x_5 = x_6 = 1 \quad ; \quad x_3 = x_4 = x_7 = x_8 = 0 \quad ; \quad x_2 = 10$$

Utilizando en la restricción después el coeficiente 30.

Max Z = 130

$$x_1 = x_4 = x_5 = x_7 = x_8 = 1 \quad ; \quad x_3 = x_6 = 0 \quad ; \quad x_2 = 1$$

a/ Los resultados obtenidos cuando la restricción está acotada superiormente por 25 son los siguientes :

$x_2 = 10$ y se trata de una variable básica factible ;

$x_1 = x_5 = x_6 = 1$ son variables no-básicas con actividad en el límite superior ;

$x_3 = x_4 = x_7 = x_8 = 0$ son variables no-básicas con actividad en el límite inferior .

La solución es óptima y el valor de la función objetivo es de 126.00 ya que

$$\text{Max } Z = 10(1) + 1(10) + 7(1) + 8(1) = 126$$

y en la restricción

$$6(1) + .8(10) + 5(1) + 6(1) = 25 \leq 25$$

b/ Los resultados obtenidos cuando la restricción está acotada superiormente por 30 son los siguientes :

$x_1 = x_2 = x_4 = x_5 = x_7 = x_8 = 1$ son variables enteras ;

$x_3 = x_6 = 0$ son variables enteras también ;

El problema tiene solución óptima y es cuando la función objetivo es igual a 130

$$\text{Max } Z = 10(1) + 1(1) + 12(1) + 7(1) + 6(1) + 3(1) = 130$$

y en la restricción se obtiene a través del paquete

$$6(1) + .8(1) + 10(1) + 5(1) + 5(1) + 3(1) = 29.8 \leq 30 ;$$

RUN*SERVICIO/MP5/ALL	(R)
ZPRINTER=.TRUE.	(R)
ZNAME="ENTERA"	(R)
ZDATA="DATOS"	(R)
ZOBJ="OF"	(R)
ZRHS="TI"	(R)
INPUT(REMOTE,SUMMARY)	(R)
ROWS	(R)
N OF	(R)
L R	(R)
COLUMNS	(R)
EMPIEZA 'MARKER' 'BIVORG'	(R)
X1 OF 101	(R)
X1 R 6	(R)
X3 R 1	(R)
X4 OF 12	(R)
X4 R 10	(R)
X5 OF 7	(R)
X5 R 5	(R)
X6 OF 8	(R)
X6 R 6	(R)
X7 OF 6	(R)
X7 R 5	(R)
X8 OF 3	(R)
X8 R 3	(R)
TERMINA 'MARKER' 'BIVEND'	(R)
EMP 'MARKER' 'INTORG'	(R)
X2 OF 1	(R)
X2 R .8	(R)
TERM 'MARKER' 'INTEND'	(R)
RHS	(R)
TI R 25	(R)
TI2 R 30	(R)
ENDATA	(R)
SETUP(MAX)	(R)

PRIMAL	(R)
MXINT	(R)
OUTPUT	(R)
ZRHS="TI2"	(R)
MXINT	(R)
RETURN	(R)
OUTPUT	(R)
EXIT	(R)

MB6700:126 CANDE 30.140; YOU ARE SCHEDNO01(102)

N DEFAULT PRINT DESTINATION=SITE

SEMINARIOS PARA MAYO/JUNIO TECLEE NEWS

SESSION 3343 15:13:14 05/06/81

RUN*SERVICIO/MPS/ALL

RUNNING 3344

~?

B7700/B6700 TEMPOC28.600.000 - MARCH 77) 05/06/81 15:13:19

READY

ZPRINTER=.TRUE.

READY

ZNAME="ENTERA"

READY

ZDATA="DATOS"

READY

ZOBJ="OF"

READY

ZRHS="TI"

READY

INPUT(REMOTE, SUMMARY)

--- INPUT ---

X

ROWS

X

N OF

Z
COLUMNS
Z
EMPIEZA 'MARKER' 'BIVORG'

Z
X1 OF 101

Z
X1 R 6

Z
X3 R 1

Z
X4 OF 12

Z
X4 R 10

Z
X5 OF 7

Z
X5 R 5

Z
X6 OF 8

Z
X6 R 6

Z
X7 OF 6

Z
X7 R 5

Z
X8 OF 3

Z
X8 R 3

Z
FORMULA 'MARKER' 'BIVORG'

λ

X2 OF 1

λ

X2 R .8

λ

TERM 'MARKER' 'INTEND'

λ

RHS

λ

T1 R 25

λ

T12 R 30

λ

ENDATA

NEW ENTRY ENTERED ON ZPROF (OR ZSOLF)

NUMBER OF ELEMENTS BY COLUMN ORDER

49	X12	X31	X42
52	X52	X62	X72
55	X82	X22		

NUMBER OF ELEMENTS BY ROW ORDER, EXCLUDING RHS, INCLUDING SLACK ELEMENT

1 N OF8 L R9

READY

SETUP(MAX)

--- SETUP ---

ERROR * INTEGER X2 BOUNDS RESET TO: UPPER= 1022 LOWER= 0

READY
PRIMAL

--- PRIMAL ---

--- INVERT ---

PROBLEM FEASIBLE.

ITER NO	OBJ VALUE	NEG DJ	OUT	IN
1	101.00000	7	490	490
2	113.00000	7	510	510
3	121.00000	7	530	530
4	125.20000	7	2	52
5	125.60000	1	520	510
6	126.00000	1	51	56

93

EXIT CONDITION: OPTIMAL SOLUTION.

FUNCTION VALUE = 126.00000

READY
MXINT

--- MXINT ---

MXINT BRANCH & BOUND
LP SOLUTION IS INTEGRAL

READY
OUTPUT

SOLUTION STATUS = OPTIMAL

ROWS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	DUAL ACTIVITY
UF	BS	126.000	NONE	NONE	1.000
R	UL	25.000	NONE	25.000	-1.250

COLUMNS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	REDUCED COST
X1	UL	1.000	.	1.000	93.500
X3	LL	.	.	1.000	-1.250
X4	LL	.	.	1.000	-0.500
X5	UL	1.000	.	1.000	0.750
X6	UL	1.000	.	1.000	0.500
X7	LL	.	.	1.000	-0.250
X8	LL	.	.	1.000	-0.750
X2	BS	10.000	.	1022.000	.

76

READY
 ZRHS="TJZ"
 READY
 MXINT

--- MXINT ---

MXINT BRANCH & BOUND

INTEGER SOLUTION

NODE ** 12 ** OPTIMIZED INTEGER NODE
 FIRST INTEGER SOLUTION OBJECTIVE = 129.0000
 CUTOFF NOW AT 129.000 POSTPONE AT 129.000

3. X4	51	1.00
4. X5	52	1.00
6. X7	54	1.00
8. X2	56	3.00

EXIT MXINT ON INTEGER SOLUTION DEMAND

*** ZINTSOL DEMAND SET ***

TCL COMMANDS MAY NOW BE ENTERED-ENTER 'RETURN' TO RESUME AN ITERATIVE
PROCEDURE.

READY

RETURN

--- MXINT ---

** RETURN TO MXINT FROM DEMAND EXIT **

BRANCH TREE STATUS: 6 ACTIVE NODES

I N T E G E R S O L U T I O N

NODE ** 13 ** OPTIMIZED INTEGER NODE

SECOND INTEGER SOLUTION OBJECTIVE = 130.0000

CUTOFF NOW AT 130.000 POSTPONE AT 136.000

A C T I V E I N T E G E R V A R I A B L E S

NAME	NUMBER	ACTIVITY
1. X1	49	1.00
3. X4	51	1.00
4. X5	52	1.00
6. X7	54	1.00
7. X8	55	1.00
8. X2	56	1.00

EXIT MXINT ON INTEGER SOLUTION DEMAND

PROCEDURE.

READY
OUTPUT

--- OUTPUT ---

SOLUTION STATUS = OPTIMAL

ROWS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	DUAL ACTIVITY
OF	BS	130.000	NONE	NONE	1.000
R	BS	29.800	NONE	30.000	.

COLUMNS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	REDUCED COST
X1	IV	1.000	.	1.000	101.000
A X3	IV	.	.	1.000	.
X4	IV	1.000	.	1.000	12.000
X5	IV	1.000	.	1.000	7.000
X6	IV	.	.	1.000	8.000
X7	IV	1.000	.	1.000	6.000
X8	IV	1.000	.	1.000	3.000
X2	IV	1.000	.	1022.000	1.000

READY

EXIT

B7700/B6700 TEMPO TERMINATED.

NET=40.7 PT=10.6 IU=5.3

3/ Considere el siguiente problema , el cual es un problema de programación mixta .

$$\text{Min } Z = 3x_1 + 2x_2 - 6x_3$$

Sujeto a.

$$9 \leq 3x_1 + 6x_2 \leq 20$$

$$4x_1 + 2x_2 - 5x_3 = -15$$

$$3 \leq x_2 \leq 5 , \quad x_2 \text{ variable entera}$$

$$x_3 = 0 , 1$$

$$x_1 \geq 0 , \quad x_1 \text{ variable real}$$

Se resolvera este problema usando el paquete TEMPO/MPS/ALL a través de una terminal remota .

Note que en la primer restricción se considera para este problema cómo una restricción con limite inferior 9 y rango 11 , la variable x_1 pertenece a los reales y es no-negativa , la variable x_2 es entera y esta acotada con limite inferior 3 y limite superior 5 , y la variable x_3 es binaria , es decir solo puede tomar el valor cero o uno .

En las hojas a continuación se muestra la forma de resolver este problema a través de la terminal , primero se dan los comandos utilizados en la secuencia seguida y posteriormente la forma en que se interactuo con la computadora a través de la terminal .

Obteniendose una solución infactible para este problema.

- 1/ Proceso seguido en la solución del problema :
 - 1.1/ Instrucción para acceder el paquete .
 - 1.2/ Se indica que habra de imprimir los resultados - no solo en la terminal , sino tambien en la impresora.
 - 1.3/ Se declararán archivos ; El archivo ZPROF se ---- generara con el nombre de PROBLEMA , El archivo ZSOLF se utilizara , es decir se almacenara en - disco la solución , en un archivo llamado SOLUCION.
 - 1.4/ Se le asigna el nombre de N al problema .
 - 1.5/ Se le asigna el nombre de N al conjunto de datos. ■■
 - 1.6/ Se llama el procedimiento INPUT de lectura , - indicando que los datos serán dados en la terminal.
 - 1.7/ Se tecleán los datos del problema , en formato libre .
 - 1.8/ Se llama el procedimiento BCDOUT para que se imprimán los datos en el formato que el paquete utiliza -- --- para su lectura en tarjetas .
 - 1.9/ Se asigna el nombre que se utilizo para la función objetivo , que en este caso fue Z .
 - 1.10/ Se asigna el nombre de SOLUCIO a la solución que se almacenara en el archivo ZSOLF .
 - 1.11/ Se asigna el nombre que se utilizo para el vector de recursos , que en este caso fue B.
 - 1.12/ Se asigna el nombre al conjunto de rangos , en - este caso fue R.
 - 1.13/ Se asigna el nombre al conjunto de cotas , en - este caso fue C.
 - 1.14/ Se llama la rutina SETUP para reservar el area - de trabajo del problema e indicar que se trata - de un problema de minimización .
 - 1.15/ Se llama el procedimiento PRIMAL para resolver el Problema primal .
 - 1.16/ Se llama el procedimiento MXINT para resolver el problema de programación mixta .
 - 1.17/ Se llama el procedimiento OUTPUT indicando que se imprimán los resultados en el archivo ZSOLF.
 - 1.18/ Se da termino a la ejecución del paquete .

■■ Estas asignaciones son requeridas antes de llamar a INPUT .

RUNXSERVICIO/MPS/ALL	(R)
ZPRINTER = .TRUE.	(R)
\$FILE ZPROF = PROBLEMA	(R)
\$FILE ZSOLF = SOLUCION	(R)
ZNAME = "N"	(R)
ZDATA = "N"	(R)
INPUT(REMOTE)	(R)
ROWS	(R)
N Z	(R)
G R1	(R)
E R2	(R)
COLUMNS	(R)
X1 Z 3	(R)
X1 R1 3	(R)
X1 R2 4	(R)
EMP1 'MARKER' 'INTORG'	(R)
X2 Z 2	(R)
X2 R1 6	(R)
X2 R2 2	(R)
TERM1 'MARKER' 'INTEND'	(R)
EMP2 'MARKER' 'BIVORG'	(R)
X3 Z -6	(R)
X3 R2 -5	(R)
TERM2 'MARKER' 'BIVEND'	(R)
RHS	(R)
B R1 9	(R)
B R2 15	(R)
RANGES	(R)
R R1 11	(R)
BOUNDS	(R)
UP C X2 5	(R)
LO C X2 3	(R)
ENDATA	(R)
BCDOUT	(R)
ZOBJ = "Z"	(R)

ZSOLNM = "SOLUCIO"	(R)
ZRHS = "B"	(R)
ZRNGST = "R"	(R)
ZBNDST = "C"	(R)
SETUP(MIN)	(R)
PRIMAL	(R)
MXINT	(R)
OUTPUT(FILE)	(R)
EXIT	(R)

#86(00:126 CANDE 30:140) YOU ARE SCHED#001(102)

DEFAULT PRINT DESTINATION=SITE

SEMINARIOS PARA MAYO/JUNIO TECLEE NEWS

#SESION 3582 15:34:05 05/06/81

RUN*SERVICIO/MPS/ALL

#RUNNING 3583

#?

B7700/B6700 TEMPC(20:000:000 - MARCH 77) 05/06/81 15:34:10

READY

ZPRINTER=.TRUE.

READY

SFILE ZPROF= PROBLEMA

READY

SFILE ZSOLF= SOLUCION

READY

ZNAME = "N"

READY

ZDATA = "N"

READY

INPUT(REMOTE)

-- INPUT -->

X

ROW?

X

N Z

X

G R1

X

E R2

X

COLUMNS

X

X1 4 3

X

X1 N1 3

X

X1 N2 4

X

EMP1 ZMARKER2 ZINTORG2

X

X2 4 2

X

X2 N1 6

X

X2 N2 2

X

TERM1 ZMARKER2 ZINTEND2

X

EMP2 ZMARKER2 ZBIVORG2

X

X3 4 =6

X

X3 N2 =5

X

TERM2 ZMARKER2 ZBIVEND2

X

X
B R1 9
X
B R2 15
X
RANGES
X
R R1 11
X
BOUNDS
X
UP 4 X2 5
X
LO 4 X2 3
X

ENDATA
OLD ENTRY N DELETED ON ZPROF (OR ZSOLF)
NEW ENTRY N ENTERED ON ZPROF (OR ZSOLF)

READY
BCDOUT

--- BCDOUT ---

NAME N
ROWS
N 2
G R1
E R2
COLUMNS

X1	Z	3.00000	R1	3.00000
X1	R2	4.00000		
EMP1	ZMARKERZ		ZINTORGZ	
X2	Z	2.00000	R1	6.00000
X2	R2	2.00000		
TERM1	ZMARKERZ		ZINTENDZ	
EMP2	ZMARKERZ		ZRIVORGZ	
X3	Z	-6.00000	R2	-5.00000
TERM2	ZMARKERZ		ZRIVENDZ	
RHS				
B	R1	9.00000	R2	15.00000
RANGES				
R	R1	11.00000		
BOUNDS				
UP C	X2	5.00000		
LO C	X2	3.00000		

ENDDATA
 READY
 ZOBJ = "Z"
 READY
 ZRHS = "B"
 READY
 ZRNUST = "R"
 READY
 ZBNVST = "C"
 READY
 SETUP(MIN)

--- SETUP ---

READY

--- PRIMAL ---

--- INVERT ---

PROBLEM INFEASIBLE

NUMBER OF INFAS = 1

SUM OF INFAS = -9.00000

ITER NO	SUM OF INF	NO INF	CUT	IN
1	-6.33333	1	20	49

EXIT CONDITION: INFEASIBLE SOLUTION

NUMBER OF INF = 1

SUM OF INF = -6.33333

*** ZDUNFS DEMAND SET ***

TCL COMMANDS MAY NOW BE ENTERED-ENTER >RETURN> TO RESUME AN ITERATIVE PROCEDURE.

READY

MXINT

--- MXINT ---

MXINT BRANCH & HOLD

CONTINUOUS PROBLEM INFEASIBLE -- PRIMAL CALLED

--- PRIMAL ---

SUM OF INF

-6.33333

*** ZDUNFS DEMAND SET ***

TCL COMMANDS MAY NOW BE ENTERED - ENTER ZRETURN2 TO RESUME AN ITERATIVE PROCEDURE.

READY
ZSOLIM = "SOLUCIO"

READY
OUTPUT(FILE)

--- OUTPUT ---

SOLUTION STATUS = INFEASIBLE
ROWS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	DUAL ACTIVITY
Z	BS	8.000	NONE	NONE	1.000
R1	UL	20.000	9.000	20.000	-1.000
R2	**	8.667	15.000	15.000	.

COLUMNS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	REDUCED CCST
X1	BS	0.667	.	NONE	.
X2	LL	3.000	3.000	5.000	-4.000
X3	LL	.	.	1.000	-6.000

NEW ENTRY SOLUCIO ENTERED ON ZPROF (OR ZSOLF)

READY

EXIT
87700/00700 TLMPC TERMINATED.

Los archivos más utilizados por el paquete TEMPO/MPS

ALL son :

- 1/ ZPROF .- Este archivo almacena los problemas generados por INPUT , REVISE , SAVE en un formato binario empa-
cado , por lo que BCDOUT es utilizado para obtener la información contenida en el archivo en imagen de tarjetas .
- 2/ ZSOLF .- Este archivo almacena la solución en un formato empa-
cado , el cual en otra corrida puede ser --
impreso al utilizar SOLOUT .
- 3/ DISKIN .- Este archivo es utilizado para leer de disco los datos de un problema , grabados en el formato --
especificado en la sección II , primero se le --
asigna el nombre de ZRHS , ZOBJ , ZDATA , ZNAME y en caso de que se utilice rangos y cotas , --
ZRNGST , ZBNDST , que deben coincidir con los --
nombres utilizados en el archivo , el cual se --
lee con INPUT(DISK) , despues se utilizán las --
instrucciones de procedimiento.

Cómo ejemplo considerece el problema de programación mixta de la sección anterior , el cual sera modificado usando el procedimiento REVISE , para resolver el problema siguiente:

$$\text{Min } Z = 2x_2 - 6x_3 - x_4$$

Sujeto a.

$$7 \leq 6x_2 - x_4 \leq 18$$

$$-x_2 + 3x_4 + 4x_3 \leq 15$$

$$x_2 \leq 666; 2 \leq x_4 \leq 8 ; x_2 , x_4 \in Z$$

$$x_3 = 0 , 1$$

El procedimiento seguido fue el siguiente :

1/ Para que el problema original se grabara con su solución se declaro el archivo ZPROF y ZSOLF de la forma siguiente :

1.1/ \$FILE ZPROF = PROBLEMA , con lo que los datos se --- grabarón en un archivo llamado problema .

1.2/ \$FILE ZSOLF = SOLUCION , con lo que la solución del problema quedo grabada en un archivo llamado solución y el nombre asignado a la solución dentro del archivo fue solucio , el que fue asignado a través de ZSOLNM.

2/ Para modificar el problema de la sección IIIc , se volvio a correr el paquete declarando ZPROF y ZSOLF con los mismos--- nombres en que se declar la versión anterior y que resulto ser no factible , y lo que hizo fue lo siguiente :

2.1/ Listar los datos del problema original (el de programación mixta) con el procedimiento BCDOUT .

2.2/ Asignar el nombre de la solución requerida que en este caso es SOLUCIO asignado en ZSOLNM y se lista lo que se tenia cómo solución con SOLOUT .

2.3/ Se modifira el problema con el procedimiento REVISE de la siguiente forma :

i/ Se modifica la segunda restricción de igual a menor o igual .

ii/ Se borra la columna x_1 .

iii/ Se inserta el vector x_4 despues del vector x_2 para que x_4 sea una variable entera .

iv/ Se modifican los coeficientes de x_2 y x_3 en la - segunda restricción .

v/ Se modifica la restricción 1 .

vi/ Se modificán las variables acotadas .

vii/ Se lista lo hecho para verificar los cambios , -- posteriormente se lista todo el problema .

2.4/ Se dan las instrucciones de procedimiento ya conocidas en las secciones anteriores de este capítulo .

3/ A continuación se da la secuencia seguida en las instrucciones , y posteriormente el listado con la ejecución en la terminal.

1/ Los resultados obtenidos para este problema son :

$x_2 = 2$ se trata de una variable entera

$x_3 = 1$ se trata de una variable entera

$x_4 = 4$ se trata de una variable entera

La solución es óptima y el valor de la función ---
objetivo es de -6 .

$$\text{Min } Z = 2(2) - 6(1) - 4 = -6$$

en las restricciones

$$7 \leq 6(2) - 4 = 8 \leq 18$$

$$-2 + 3(4) + 4(3) = 14 \leq 15$$

Ademas se cumplen las restricciones de las variables
en sus cotas .

RUN*SERVICIO/MPS/ALL	(R)
ZPRINTER=.TRUE.	(R)
\$FILE ZPROF = PROBLEMA	(R)
\$FILE ZSOLF = SOLUCION	(R)
ZNAME = "N"	(R)
ZDATA = "N"	(R)
BCDOUT	(R)
ZSOLNM = "SOLUCIO"	(R)
SOLOUT	(R)
ZONAME = "N"	(R)
ZNAME = "N1"	(R)
REVISE(REMOTE)	(R)
ROWS	(R)
MODIFY	(R)
L R2	(R)
COLUMNS	(R)
DELETE	(R)
X1	(R)
AFTER X2	(R)
X4 Z -1	(R)
X4 R1 -1	(R)
X4 R2 3	(R)
MODIFY	(R)
X2 R2 -1	(R)
X3 R2 4	(R)
RHS	(R)
MODIFY	(R)
B R1 7	(R)
BOUNDS	(R)
MODIFY	(R)
UP C X4 8	(R)
LD C X4 2	(R)
UP C X2 666	(R)
LIST	(R)
ENDATA	(R)

BCDOUT	(R)
ZBNDST = "C"	(R)
ZRNGST = "R"	(R)
ZOBJ = "Z"	(R)
ZRHS = "B"	(R)
SETUP(MIN)	(R)
PICTURE	(R)
TRANCOL	(R)
PRIMAL	(R)
MXINT	(R)
RANGE	(R)
OUTPUT(BASIS)	(R)
OUTPUT(INFEAS)	(R)
TRANCOL	(R)
TRANCOL(MATRIX)	(R)
TRANCOL(INVERSE)	(R)
OUTPUT	(R)
EXIT	(R)

#B6(00)126 CANDE. 30*140 YOU ARE SCHED#001(102)

DEFAULT PRINT DESTINATION#SITE

SEMINARIOS PARA MATU/JUNIC TECLEE NEWS

#SESSION 3747 15151157 05/06/81

RUN*SERVICIO/MPS/ALL

#RUNNING 3749

#?

B7700/B6700 TEMPC(20*600*000 - MARCH 77) 05/06/81 151521 2

READY

ZPRINTER#*TRUL*

READY

SFILE 4PROF = PROBLEMA

READY

SFILE 4SOLF = SOLUCION

READY

ZNAME = "N"

READY

ZDATA = "N"

READY

BCDUUT

--- BCDUUT ---

NAME N

ROWS

N 2

G R1

E R2
COLUMNS

X1	Z	3.00000	R1	3.00000
X1	R2	4.00000		
EMP1	ZMARKERZ		ZINTORGZ	
X2	Z	2.00000	R1	6.00000
X2	R2	2.00000		
TERM1	ZMARKERZ		ZINTENDZ	
EMP2	ZMARKERZ		ZBIVORGZ	
X3	Z	-6.00000	R2	-5.00000
TERM2	ZMARKERZ		ZBIVENDZ	

RHS

B	R1	9.00000	R2	15.00000
---	----	---------	----	----------

RANGES

R	R1	11.00000
---	----	----------

BOUNDS

UP C	X2	5.00000
LO C	X2	3.00000

ENDATA
 READY
 ZSDINM = "SOLUCIO"
 READY
 SOLUUT

--- SOLUUT ---

S O L U T

PROBLEM IDENTIFICATION
 PROBLEM NAME = N

RANGE SET NAME = R
 BOUND SET NAME = U
 FUNCTIONAL NAME = Z
 RESTRAINT NAME = B

SOLUTION STATUS = INFEASIBLE
 ITERATION NUMBER = 1
 FUNCTIONAL VALUE = 8.00000

ROWS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	DUAL ACTIVITY
Z	BS	8.000	NONE	NONE	1.000
R1	LL	20.000	9.000	120.000	-1.000
R2	**	8.667	15.000	15.000	.

COLUMNS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	REDUCED CGST
X1	BS	0.667	.	NONE	.
X2	LL	3.000	3.000	5.000	-4.000
X3	LL	.	.	1.000	-6.000

READY

ZONAME = "N"

READY

ZNAME = "N1"

READY

REVISE (REMOTE)

=== REVISE ===

X
ROWS
X
MODIFY
X
L R2
X
COLUMNS
X
DELETE
X
X1
X
AFTER X2
X
X4 4 -1
X
X4 R1 -1
X
X4 R2 3
X
MODIFY
X
X2 R2 -1
X
X3 R2 4
X
RHS
X
MODIFY
X

B R 7
 X
 BOUNDS
 X
 MODIFY
 X
 UP 6 X4 8
 X
 LO 6 X4 2
 X
 UP 6 X2 666
 X

LIST
 NAME N
 ROWS

MODIFY			
L R2			
COLUMNS			
DELETE			
X1			
AFTER	X2		
X4	L		-1
X4	R1		-1
X4	R2		3
MODIFY			
X2	R2		-1
X3	R2		4

RHS
 MODIFY
 B R1 7

BOUNDS

MODIFY

UP C X4 8
 LO C X4 2
 UP C X2 666

ENDATA

*

ENDATA

REVISING N IN N1 ACCORDING TO N
 NEW ENTRY N1 ENTERED ON ZPROF (OR ZSOLF)
 READY
 BCDOUT

--- BCDOUT ---

NAME N

ROWS

N 2

G R1

L R2

COLUMNS

EMP1	ZMARKER2		ZINTOR02	
X2	L	2.00000	R1	6.00000
X2	R2	-1.00000		
X4	L	-1.00000	R1	-1.00000
X4	R2	3.00000		
TERM1	ZMARKER2		ZINTEND2	
EMP2	ZMARKER2		ZRIVOR02	
X3	L	-6.00000	R2	4.00000
TERM2	ZMARKER2		ZRIVEND2	
RHS				
b	R1	7.00000	R2	15.00000

UP C X2 666.00000
 UP C X4 8.00000
 LO C X4 2.00000

ENDATA
 READY
 ZBNDST = "C"
 READY
 ZRNGST = "F"
 READY
 ZOBJ = "Z"
 READY
 ZRHS = "B"
 READY
 SETUP(MIN)

--- SETUP ---

PROBLEM STATISTICS

	NUMBER	FREE	FIXED	BOUNDED	NORMAL
ROWS :	3	1	0	1	1
COLUMNS:	3	0	0	3	0

MATRIX IN CORE : MEMORY ALLOCATION = 75 WORDS.
 INVERSE: MEMORY ALLOCATION = 450 WORDS. RECORD LENGTH = 225 WORDS.

NUMBER OF INTEGER VARIABLES = 3
 READY
 PICTURE

--- PICTUR ---

OUTPUT FOR PICTURE WILL BE PRODUCED ON SYSTEM PRINTER ONLY.

118

B6700/7700 TEMPO
 VERSION: 28.600.000

N1

PAGE

PART 1 OF 1
 PAGE 1

X	X	X	B	R	S
2	4	3		R	C
				A	A

UPPER	SCALE	C	A		
LOWER	BOUND				
	BOUND				
	TYPE				
7	1	A	-1	-4	7
R1	G	A	-1	A	R1
R2	L	-1	A	A	R2

SUMMARY OF MATRIX

SYMBOL	RANGE	COUNT	INCL	RHS
Z	LESS THAN	.000001		
Y	.000001 THRU	.000009		
X	.000010 THRU	.000099		
W	.000010 THRU	.000099		
V	.001000 THRU	.009999		
U	.010000 THRU	.099999		
T	.100000 THRU	.999999		
S	1.000000 THRU	1.000000		
R	1.000001 THRU	10.000000		
Q	10.000001 THRU	100.000000		
P	100.000001 THRU	1000.000000		
O	1000.000001 THRU	10000.000000		
N	10000.000001 THRU	100000.000000		
M	100000.000001 THRU	1000000.000000		
L	1000000.000001 THRU	10000000.000000		
K	GREATER THAN	1.000000		

MINIMUM = -6.000000 MAXIMUM = 15.000000

READY
 TRANCO

--- TRANCO ---

OUTPUT FOR TRANCO WILL BE PRODUCED ON SYSTEM PRINTER ONLY.

--- INVERT ---

CURRENT INVERSE : ETA RECORDS = 1. ETA VECTORS = 0. ELEMENTS = 0.
 CURRENT EASTS : EQUALITY = 0. SLACKS = 2. STRUCTURALS = 0. ELEMENTS = 2.
 FUNCTION VALUE = -2.00000

T R A N C O L

T A B L E A U

BOUND	X2 LOWER	X4 LOWER	X3 LOWER	B LOWER
Z	2.00000	-1.00000	-6.00000	
R1	-6.00000	1.00000		-7.00000
R2	-1.00000	3.00000	4.00000	15.00000

READY
 PRIMAL

--- PRIMAL ---

ZOBJ = Z ZRHS = B
 ITERATION NUMBER SUM OF NUMBER REDUCED PIVOT VECTOR VECTOR FUNCTION
 TYPE NUMBER INFEAS NEG DJ COST INDEX OUT IN VALUE
 P 1 1 0 . 1 -6.00000 2 2 49 1.00000

SOLUTION FEASIBLE.

119

CURRENT INVERSE : EQUALITY = 0.0
 CURRENT EASTS : FORWARD TRIANGULAR VECTORS = 1.00000
 NEW INVERSE : FUNCTION VALUE = 1.00000
 ZOBJ = Z ZRHS = B
 ITERATION NUMBER SUM OF NUMBER REDUCED PIVOT VECTOR VECTOR FUNCTION
 TYPE NUMBER INFEAS INFEAS NEG DJ COST INDEX OUT IN VALUE
 PB2 2 0 : 2 0 3 51U 51J
 1 3 0 : 2 3 3 50 50
 -6.00000
 -0.66667

EXIT CONDITION: OPTIMAL SOLUTION.

FUNCTION VALUE = -6.52941

READY
MXINT

--- MXINT ---

M X I N T B R A N C H & B O U N D

I N T E G E R S O L U T I O N

NODE ** 4 ** OPTIMIZED INTEGER NODE
 FIRST INTEGER SOLUTION OBJECTIVE # 6.0000
 CUTOFF NOW AT 0.0000 POSTPONE IT 6.0000
 ACTIVE INTEGER VARIABLES

NAME	NUMBER	ACTIVITY
1. X2	49	2.00
2. X4	50	4.00
3. X3	51	1.00

EXIT MXINT ON INTEGER SOLUTION DEMAND

*** ZINTSOL DEMAND SET ***

TCL COMMANDS MAY NOW BE ENTERED ENTER ZRETURN2 TO RESUME AN ITERATIVE

READY
RANGE

--- RANGE ---

COLUMNS AT LIMIT LEVEL

COLUMN ST	ACTIVITY INPUT COST	LO LIMIT UP LIMIT	LO ACTIV UP ACTIV	UNIT COST	UP COST LO COST	LIMITING PROCESS	ST
X2	0	0	-0.167	2.000	INFINITY	R1	LL
LL	2.000	664.000	1.667	2.000	0	R1	UL
X4	4.000	2.000	-6.000	1.000	0	R1	LL

UL	1.000	4.000	4.333	1.000	INFINITY R2	LL
X3	1.000	.	-INFINITY	6.000	.	NCNE
UL	6.000	1.000	1.250	6.000	INFINITY R2	UL

ROWS AT INTERMEDIATE LEVEL

ROW	ACTIVITY	LO LIMIT	LO ACTIV	UNIT	LIMITING	ST
ST	SLK ACTIV	UP LIMIT	UP ACTIV	COST	PROCESS	
R1	8.000	7.000	8.000	INFINITY	NONE	
BS	1.000	18.000	INFINITY	0.333	X2	LL
R2	14.000	NONE	16.000	0.333	X4	UL
BS	1.000	15.000	14.000	INFINITY	NONE	

READY
OUTPUT(BASIS)

--- OUTPUT ---

SOLUTION STATUS = OPTIMAL
ROWS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	DUAL	ACTIVITY
Z	BS	6.000	NCNE	NONE		1.000
R1	BS	8.000	7.000	18.000		.
R2	BS	14.000	NCNE	15.000		.

COLUMNS SECTION

NAME STAT ACTIVITY LOWER LIM UPPER LIM REDUCED COST

READY
OUTPUT (INFEAS)

--- OUTPUT ---

SOLUTION STATUS : OPTIMAL
ROWS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	DUAL ACTIVITY
Z	BS	-6,000	NONE	NONE	1,000
R1	BS	8,000	7,000	18,000	.
R2	BS	14,000	NONE	15,000	.

COLUMNS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	REDUCED COST
X2	IV	2,000	.	666,000	2,000
X4	IV	4,000	2,000	8,000	-1,000
X3	IV	1,000	.	1,000	-6,000

READY
TRANCO

--- TRANCO ---

OUTPUT FOR TRANCO WILL BE PRODUCED ON SYSTEM PRINTER ONLY.
READY
TRANCO (MATRIX)

--- TRANCO ---

OUTPUT FOR TRANCO WILL BE PRODUCED ON SYSTEM PRINTER ONLY.

READY
TRANCO(INVERSE)

--- TRANCO ---

OUTPUT FOR TRANCO WILL BE PRODUCED ON SYSTEM PRINTER ONLY.

READY
OUTPUT

--- OUTPUT ---

SOLUTION STATUS : OPTIMAL
ROWS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	DUAL ACTIVITY
Z	BS	-6.000	NONE	NONE	1.000
R1	BS	8.000	7.000	18.000	.
R2	BS	14.000	NONE	15.000	.

COLUMNS SECTION

NAME	STAT	ACTIVITY	LOWER LIM	UPPER LIM	REDUCED CCST
X2	IV	2.000	.	666.000	2.000
X4	IV	4.000	2.000	8.000	-1.000
X3	IV	1.000	.	1.000	-6.000

READY
EXIT
B7700/B6700 TEMPC TERMINATED.
#ET=1109.2 PT=15.0 IO=8.5
#END SESSION 2747 ET=1112.7 PT=15.0 IO=8.5
#USER = SF82 1515.10 05/06/81

C A P I T U L O I I I

M A N E J O D E L P A Q U E T E
T E M P O / N E T W O R K
E N L A S O L U C I O N D E
P R O B L E M A S D E
R E D E S

1.0 INTRODUCCION A TEMPO / NETWORK

Los problemas más interesantes de programación --- matemática que se encuentran , son aquellos en los que se -- involucra determinar el valor mínimo , de una función , del - flujo en una red . Desde que George Dantzig desarrollo el -- algoritmo simplex en 1947 , los modelos de redes son estudia- -dos extensamente . Algunos atribuyen que esto se debe a las numerosas aplicaciones de modelos de flujo en redes . Debido a que la estructura de una red se puede explotar por medio de algoritmos desarrollados especializados en redes , que deter- minán la solución optima mucho más rápido que los algoritmos generales de programación matemática . Además la geometría de una red , se puede describir más fácilmente por medio de una gráfica ; simplificando así la comprensión del problema que se analiza . A continuación se da una lista de algunos sistemas en los cuales se utilizarán modelos de redes :

- 1/ Sistemas de producción y distribución .
- 2/ Sistemas de trafico urbano .
- 3/ Sistemas de redes de tubería .
- 4/ Sistemas de comunicación .
- 5/ Sistemas ferroviarios .
- 6/ Redes eléctricas .

El paquete TEMPO/NETWORK que se trata en este ---- capítulo , esta diseñado para resolver en forma eficiente -- problemas de redes . El manejo de este paquete se ilustrara - por medio de la solución de un pequeño problema de traslado .

El proceso seguido en la solución del problema de - traslado será suficiente para comprender el manejo del paquete asi como el proceso a seguir en la preparación de datos , para resolver problemas más complejos . En este capítulo se supone que el usuario sabe programación lineal y conoce algunas --- aplicaciones ; en particular aplicaciones en redes .

El paquete utiliza un grupo de algoritmos los cuales -
resuelven eficientemente los siguientes tipos de problemas :
Problemas de redes , capacidad de flujo a costo mínimo , ----
problemas de traslado .

Estos problemas tienen la siguiente forma :

$$\text{Min } Z = \sum_{(i,j) \in A} c_{ij} \cdot x_{ij}$$

sujeto a :

$$\sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = a_i ; \quad i \in N$$

$$L_{ij} \leq x_{ij} \leq U_{ij} \quad \text{y} \quad (i,j) \in A$$

donde : N denota el conjunto de nodos (i)

A denota el conjunto de arcos (i,j) , i,j ∈ N

donde la pareja ordenada (i,j) estara denotando
el arco del nodo i a el nodo j

x_{ij} denota el flujo a través del arco (i,j)

c_{ij} denota el costo de transporte por unidad de flujo
a través del arco (i,j)

a_i denota la unidad de flujo útil en el nodo i (si
es positivo la oferta en el nodo i , si es negativ
o la demanda en el nodo i)

L_{ij} denota las cotas inferiores sobre el flujo a través
del arco (i,j)

U_{ij} denota las cotas superiores sobre el flujo a través
del arco (i,j) .

El paquete TEMPO/NETWORK en la solución de problemas de redes utiliza dos archivos de datos ; en un archivo de datos llamado CARDIN se indican los datos que describen la red . Y en el otro archivo llamado CARD utiliza un programa de control que se construye por medio de llamadas a procedimientos del paquete. [C]

Los procedimientos que utiliza el paquete se clasifican de acuerdo a sus funciones en los siguientes : [1]

- 1/ Procedimientos de entrada .- Estos procedimientos se utilizarán para definir el problema de redes y obtener una representación interna de el mismo que se procese eficientemente .
- 2/ Procedimiento de optimización .- Este procedimiento - utiliza un metodo primal de redes para obtener la -- solución optima del problema de redes .
- 3/ Procedimientos de salida .- Estos procedimientos dan información acerca del problema , así como su solución.
- 4/ Procedimientos de preservación de bases .- Estos -- procedimientos se utilizarán para almacenar y posteri - ormente recuperar la base asociada con el problema .
- 5/ Procedimientos de utileria .- Estos procedimientos se utilizarán para realizar diversas funciones. , sobre el problema de redes que se resuelve .

[C] Estos archivos serán tratados más adelante .

[C1] Posteriormente se verán los procedimientos más utilizados y la descripción de resultados que con ellos se obtiene .

Para una mayor información se sugiere ver [3]

La forma en que el usuario describe el problema de redes al paquete , es por medio de un archivo de datos , que el paquete reconoce como CARDIN . La manera en que se accesa la información por este archivo , es utilizando tres tipos de tarjetas :

- 1/ Tarjetas de nodos .
- 2/ Tarjetas de restricciones en los nodos .
- 3/ Tarjetas de arcos .

El formato utilizado por estas tarjetas es el que se muestra en la tabla 3.1 .

Campo	1	2	3	4	5
Columnas	5 - 12	15 - 22	25 - 36	40 - 51	55 - 66
Contenido	Nombre	Nombre	Valor	Valor	Valor

Tabla 3.1

1/ Tarjetas de nodos .- En este tipo de tarjeta se indica el nombre de cada centro (nodo) en el campo 1 . La oferta - disponible del centro en el campo 3 . Y la demanda requerida por el centro en el campo 4 . Los otros campos permanecen en blanco , como se indica en la tabla 3.2 .

			C a n t i d a d		
	CENTRO		OFERTA	DEMANDA	
Campo	1	2	3	4	5

Tabla 3.2

2/ Tarjetas de restricciones en los nodos .- En este tipo de tarjeta se indica para cada centro (nodo) , el nombre del centro en el campo 1 . Un símbolo * , + o - dependiendo del -

centro ; un * indica que se trata de un centro de traslado , un + indica que se trata de un centro de demanda , un - indica que se trata de un centro de oferta ; el símbolo ocupara el campo 2 y debera estar justificado a la izquierda . El costo de traslado o producción (oferta) o venta (demanda) ---- ocupara el campo 3 . El flujo máximo permitido en el centro - se indicara en el campo 4 . El flujo mínimo permitido en el centro se indicara en el campo 5 . Esto se muestra en la --- tabla 3.3.

	CENTRO	SIMBOLO	PRECIO POR UNIDAD	FLUJO MAXIMO	FLUJO MINIMO
Campo	1	2	3	4	5

Tabla 3.3

3/ Tarjetas de arcos .- En este tipo de tarjeta se indica el centro de origen en el campo 1 . El centro de destino con el que se forma el arco en el campo 2 . El costo de transporte - por el arco se indica en el campo 3 . El flujo máximo que --- puede pasar por el arco en el campo 4 . El flujo mínimo que - puede pasar por el arco en el campo 5 . Esto se muestra en la tabla 3.4 .

	ARCO		COSTO DE TRANSPORTE	FLUJO MAXIMO	FLUJO MINIMO
	ORIGEN	DESTINO			
Campo	1	2	3	4	5

Tabla 3.4

Las tarjetas que constituyen el archivo de datos -
CARDIN utilizan el siguiente orden :

- 1/ Una tarjeta de NAME , con el nombre que se le asigna al problema de redes en el campo 2 .
- 2/ Tarjetas de nodo en el siguiente orden :
 - 2.1/ Tarjeta de nodo .
 - 2.2/ Todos los arcos originados del nodo .
 - 2.3/ Todas las restricciones de demanda para el nodo .
 - 2.4/ Todas las restricciones de oferta para el nodo .
 - 2.5/ Todas las restricciones de traslado para el nodo .
- 3/ Una tarjeta de ENDATA en el campo 1 .

El programa de control que resuelve el problema de redes es accesado por el archivo CARD . Este archivo utiliza dos tipos de instrucciones de procedimiento , en el proceso seguir en la solución del problema de redes . Estos dos tipos de instrucciones de procedimiento son las siguientes :

- 1/ De asignación de nombres , del conjunto de datos y del problema .
- 2/ De procedimiento , para el proceso a seguir en la --- solución del problema de redes .

Las instrucciones de asignación son las siguientes :

ZDATA = "REDES"

Esta instrucción se utiliza para asignar un nombre al conjunto de datos , debe aparecer con el mismo nombre con el que se identifica el conjunto de datos en el archivo CARDIN (tarjeta de NAME) , en este caso se asigno el nombre REDES .

ZNAME = "REDES"

Esta instrucción se utiliza para asignar un nombre al problema de redes , en este caso se asigno el nombre REDES .

Las instrucciones de procedimientos más utilizados del paquete son las siguientes :

BUILD

Este procedimiento se utiliza para inicializar un archivo ZPROF , en el que se almacenara la descripción del problema , en disco . Este procedimiento no es necesario si el usuario planea almacenar el problema en un archivo ZPROF ya existente .

BCDOUT

Este procedimiento produce la impresión de los ---- datos del problema almacenado en ZPROF . Esto es convierte - el problema que se encuentra en un formato empacado en el --- archivo ZPROF para una salida externa , en el formato de los datos de entrada .

INPUT

Este procedimiento se utiliza para leer los datos - que describen el problema de redes , transformandolos a un -- formato binario empacado y almacenandolos en un archivo ----- llamado ZPROF . A la vez que se genera el archivo ZPROF se -- verificán errores ; las estadísticas del problema se recopilan para utilizarlas posteriormente . El conjunto de datos que se leen debe estar definido en ZDATA . El problema se etiqueta - con el nombre con el que se definio en ZNAME . Si un problema residente en ZPROF tiene el mismo nombre del nuevo problema - entoces se borra ; aun cuando el area de almacenamiento --- ocupada por el conjunto de datos borrados no sea requerida .

Las impresiones que lleva a cabo este procedimiento son las siguientes :

- 1/ Se imprimen los errores detectados con un mensaje ---- autoexplicativo que indica que acción seguir.
- 2/ El número de nodos (incluyendo el número de restricci-
ones) , el número de arcos , el número de errores .

SETUP

Este procedimiento utiliza los datos del problema - que están almacenados en el archivo ZPROF para generar la -- matriz de trabajo . Si una oferta o demanda no balanceada --

existe , se crea un nodo artificial llamado S (sumidero), el cual balancea una oferta excesiva . El nodo fuente balancea una demanda excesiva . Este procedimiento produce la impresión de el número de nodos , número de nodos originales , ----- número de restricciones en nodos , número de nodos artificiales número de arcos , y demanda y oferta neta .

OPTIMIZE

Este procedimiento obtiene una solución optima --- factible (si existe) , utilizando un metodo de redes .

Imprimiendo a cada iteración lo siguiente :

- 1/ ITERATION NUMBER .- Número de iteración .
- 2/ ARC IN AND ARC OUT .- Número interno del arco que entra o sale de la base . (0 - 0) indica que ningun arco - sale de la base . Cuando el arco que sale es igual al arco que entra , y el número de arcos artificiales se decrementa , entonces el arco artificial se reemplaza. Cuando el arco que sale y el que ntra a la base son - iguales y el número de arcos artificiales se preserva indica que el arco se encuentra en una de sus cotas.
- 3/ AMOUNT OF FLOW .- Cambio de flujo .
- 4/ NUMBER OF ART. ARCS .- Número de arcos artificiales.
- 5/ TOTAL COST OF ART. ARCS .- Penalización ocasionada - por arcos artificiales .
- 6/ REDUCED COST .- Costos reducidos por un cambio unitario de flujo .
- 7/ OBJECTIVE VALUE .- Valor de la función objetivo .

OUTPUT

Este procedimiento imprime la solución del problema de redes . La solución que se lista durante o despues de la - optimización , se imprime en dos secciones :

- 1/ Sección de identificación .
- 2/ Sección de resúmenes de los nodos .

En la impresión se emite la siguiente información :

1/ Sección de identificación .

1.1/ Nombre del problema .

1.2/ Estado de la solución .

1.2.1/ INFEASIBLE .- Infactible .

1.2.2/ OPTIMAL .- Optima .

1.2.3/ NON-OPTIMAL .- NO optima .

1.3/ Número de iteración .

1.4/ Valor de la función objetivo .

1.5/ Número de arcos artificiales .

2/ Sección de resúmenes de los nodos .- Para cada nodo se imprime su actividad en la solución . Esta impresión contiene la siguiente información :

2.1/ Nombre del nodo .

2.2/ Número interno que le asigno el paquete .

2.3/ Tipo de nodo .

2.3.1/ SOURCE .- Nodo fuente .

2.3.2/ SINK .- Nodo sumidero .

2.3.3/ TRANSSHIP .- Nodo de traslado .

2.4/ Oferta disponible en el nodo .

2.5/ Demanda requerida en el nodo .

2.6/ Flujo total que ingresa al nodo .

2.7/ Flujo total que sale del nodo .- En el caso de el nodo fuente se lista un resumen de sus arcos principales . El destino de los arcos se lista bajo la columna DESTINATION . Para el nodo --- sumidero se listan sus arcos principales . El origen de estos arcos se lista bajo la columna ORIGINATION . Para los nodos de traslado se --- imprimen dos listas de sus arcos , una de los nodos de origen de arcos bajo la columna ORIGINATION y la otra de nodos de destino bajo la columna DESTINATION. También se imprime para cada arco la siguiente información :

- 2.7.1/ NUMBER .- Número interno del nodo origen o nodo destino del arco .
- 2.7.2/ STATUS .- Dos caracteres indicarán el estado del arco en la solución del problema de redes.
 - 2.7.2.1/ BS .- El arco esta en la base .
 - 2.7.2.2/ UL .- El arco es no basico y su actividad se encuentra en su cota superior .
 - 2.7.2.3/ LL .- El arco es no basico y su actividad se encuentra en su cota inferior .
 - 2.7.2.4/ AB .- Se trata de un arco artificial que se encuentra en la base'.
- 2.7.3/ ACTIVITY .- Flujo que pasa a través del arco.
- 2.7.4/ INPUT COST .- Costo unitario del flujo que pasa por el arco .
- 2.7.5/ UPPER LIMIT .- Flujo máximo que puede pasar a través del arco .
- 2.7.6/ LOWER LIMIT .- Flujo mínimo que puede pasar a través del arco .
- 2.7.7/ REDUCED COST .- Tasa de incremento en la función objetivo por incremento unitario de flujo en el arco .

Si el nodo tiene restricciones , un resumen de las restricciones se imprime . Listando para cada restricción la siguiente información :

- 2.8/ TYPE .- Tipo de restricción .
 - 2.8.1/ DEMAND .- Demanda .
 - 2.8.2/ SUPPLY .- Oferta .
 - 2.8.3/ TRANSSHIP .- traslado .
- 2.9/ NUMBER .- Número interno de la restricción .
- 2.10/ STATUS.- Dos caracteres indicarán el estado de la restricción , en la solución .
 - 2.10.1/ BS .- La restricción se encuentra en la base.

2.10.2/ UL .- No basica , con actividad en la cota superior .

2.10.3/ LL .- No basica , con actividad en la cota inferior .

2.11/ ACTIVITY .- Unidades de flujo .

2.12/ COST PRICE .- Costo unitario de una oferta , demanda o traslado .

2.13/ UPPER LIMIT .- Flujo máximo requerido .

2.14/ LOWER LIMIT .- Flujo mínimo requerido .

2.0 EJEMPLO DE APLICACION

A continuación se resolvera un problema de traslado el cual se describe en las paginas siguientes .

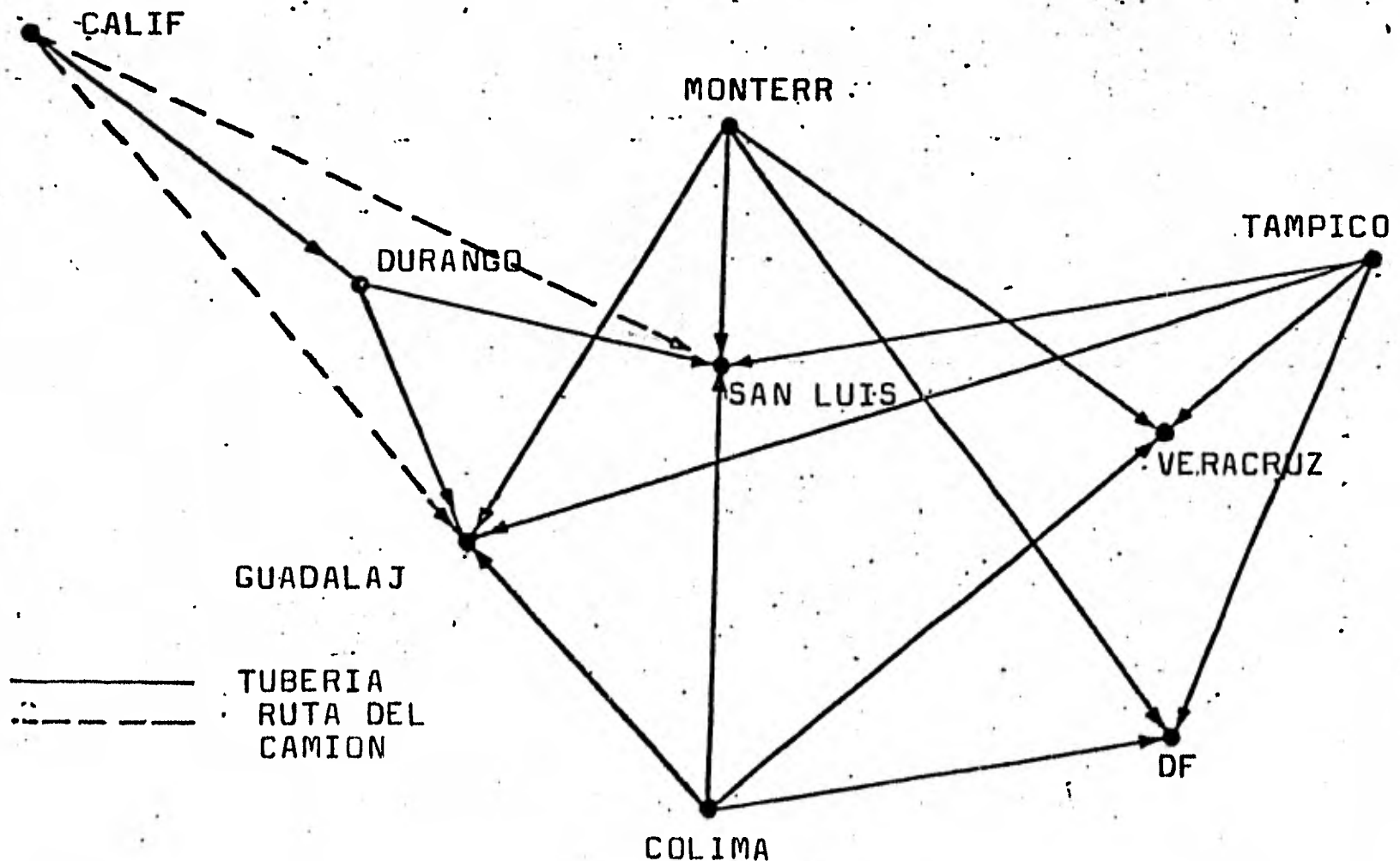


Figura 3.1 Red de ejemplo .

Tabla 3.5

LOCALIDAD			VERACRUZ	DF	SAN LUIS	GUADALAJ	FACILIDAD DE TRASLADO LOCALIDAD DURANGO
MERCADO	DEMANDA (*1)	PRECIO DE VENTA (*2)	280000 \$.54	260000 \$.48	220000 \$.52	240000 \$.46	
REFINERIA			COSTO UNITARIO DE TRANSPORTE				
LOC.	PRODUCCION (*1)	COSTO (*2)					
TAMPICO	400000	\$.40	\$.03	\$.09	\$.11	\$.13	---
COLIMA	250000	.30	.10	.04	.07	.03	---
CALIF	200000	.20	---	---	.22	.24	\$.10
MONTERR	150000	.35	.06	.12	.03	.08	---
FACILIDAD DE TRASLADO							
LOC.	COSTO DE TRASLADU						
DURANGO	\$.02		---	---	.08	.08	

1 TODOS LOS VALORES DE PRODUCCION Y/O DEMANDA ESTAN EN LITROS .

2 TODOS LOS COSTOS Y PRECIOS ESTAN EN PESOS POR LITRO .

El problema de ejemplo que se resolvera consiste en lo siguiente . Una compañía petrolera desea maximizar sus beneficios en la distribución diaria de gasolina , desde cuatro de sus refinarias , localizadas en TAMPICO , COLIMA , CALIF , y MONTEERR a cuatro mercados localizados en VERACRUZ , DF , - SAN LUIS , y GUADALAJ . En la figura 3.1 se muestra la red de traslado , y en la tabla 3.5 se indican los costos , precios de venta , producciones y demandas asociadas con la red .

Agregando a las restricciones de la tabla 3.5 , la facilidad de traslado que se ofrece en DURANGO , y que puede tomar no más de 150,000 litros por día y un convenio con la - tuberia principal de TAMPICO-VERACRUZ , que requiere al menos 100,000 litros diarios que pasán por esta tuberia . La ----- capacidad de la tuberia de TAMPICO-VERACRUZ es de 200,000 litros diarios .

Los tres tipos de datos requeridos , nodo , restricción en nodos , y arcos requeridos por el problema de ejemplo se muestrán en la tabla 3.6 . Tal como se especifico en paginas anteriores . Para el ejemplo se considero en la red lo ----- siguiente :

- 1/ DATOS DE NODOS .- Cada refinaria , mercado , y facilidad de traslado esta representado en la red por medio de un nodo y tiene una entrada correspondiente a el . Esta - entrada incluye el nombre del nodo y la oferta (para una refinaria) o demanda (para un mercado) en el - nodo .
- 2/ DATOS DE RESTRICCION EN LOS NODOS .- Las restricciones de los nodos en el ejmplo están representadas por el flujo que pasa a través del nodo , costos unitarios de ofertas al nodo , precios unitarios de venta o demanda en el nodo , y costos unitarios de traslado de flujo a través del nodo . En el problema de ejemplo el flujo que pasa por DURANGO esta restringido a 150,000 litros diarios ; cada refinaria tiene un costo unitario (por litro) de producción) ; cada mercado tiene un precio unitario de venta ; y la facilidad de traslado que - ofrece DURANGO , tiene un costo unitario de traslado de \$.02 .

NAME	REDES			
TAMPICO		400000.		
TAMPICO	VERACRUZ	.03	200000.	100000.
TAMPICO	SAN LUIS	.11		
TAMPICO	DF	.09		
TAMPICO	GUADALAJ	.13		
TAMPICO	-	.4		
MONTEERR		150000.		
MONTEERR	VERACRUZ	.06		
MONTEERR	SAN LUIS	.03		
MONTEERR	DF	.12		
MONTEERR	GUADALAJ	.08		
MONTEERR	-	.35		
COLIMA		250000.		
COLIMA	VERACRUZ	.1		
COLIMA	SAN LUIS	.07		
COLIMA	DF	.04		
COLIMA	GUADALAJ	.03		
COLIMA	-	.3		
CALIF		200000.		
CALIF	SAN LUIS	.22		
CALIF	GUADALAJ	.24		
CALIF	DURANGO	.1		
CALIF	-	.2		
DURANGO				
DURANGO	SAN LUIS	.08		
DURANGO	GUADALAJ	.08		
DURANGO	*	.02	150000.	
SAN LUIS			220000.	
SAN LUIS	+	.52		
GUADALAJ			240000.	
GUADALAJ	+	.46		
VERACRUZ			280000.	
VERACRUZ	+	.54		
DF			260000.	
DF	+			

ENDATA

Tabla 3.6

El costo de traslado representa el costo adicional en que se incurre porque transite el flujo de la tubería ----- CALIF-DURANGO a la tubería DURANGO-SAN LUIS y/o a la tubería de las ciudades de DURANGO-GUADALAJ .

3/ DATOS DE ARCOS .- Cada tubería o ruta de camión --- representa un arco en la red , cada arco tiene una entrada en los datos de lectura , los cuales incluyen el costo de embarque por unidad de flujo sobre el arco y todas las restricciones sobre la capacidad del arco. Por ejemplo , La entrada al arco que representa a la tubería que va de TAMPICO a VERACRUZ , incluye el costo de transporte de \$.03 por litro y indica un flujo -- máximo de 200,000 litros , y un flujo mínimo de --- 100,000 litros .

El proceso seguido en la solución del problema de - ejemplo se da a continuación :

1/ Tarjetas de control del sistema .- Estas tarjetas son las que en la columna uno tienen un caracter "?" . y se utilizarán para acceder el paquete .

2/ Tarjetas de instrucciones sobre el problema .- En estas tarjetas se indica el proceso a seguir en la solución del problema . Siendo el siguiente :

2.1/ Se llama el procedimiento BUILD para que se --- almacene en disco la definición del problema , en un archivo llamado ZPROF .

2.2/ Se le asigna el nombre de REDES al conjunto de datos.

2.3/ Se le asigna el nombre de REDES al problema de - ejemplo .

2.4/ Se llama el procedimiento de lectura , para leer la descripción de la red , que se tiene en el -- conjunto de datos del archivo CARDIN .

2.5/ Se llama el procedimiento SETUP .

2.6/ Se llama el procedimiento OPTIMIZE , para determinar la solución óptima del problema de ejemplo .

2.7/ Se llama el procedimiento OUTPUT , para que se - listen los resultados del problema .

2.8/ Se utiliza la instrucción EXIT para indicar que el programa de control termina .

3/ Tarjetas de datos .- Se utilizarón los datos que describen la red de ejemplo , que se encuentran en la tabla 3.6 .

El ejemplo completo del paquete de tarjetas que se utilizo en la solución del problema de redes de ejemplo se muestra en la tabla 3.7 .

```
?JOB REDES;USER=SF82/MP;CLASS=5;BEGIN
?RUN*SYSTEM/NTW/ALL;DATA CARD
BUILD
ZDATA="REDES"
ZNAME="REDES"
INPUT
BCDOUT
SETUP(MAX)
OPTIMIZE
OUTPUT
EXIT
?DATA CARDIN

    Tarjetas de datos de la tabla 3.6

?END JOB
```

Tabla 3.7

La solución optima (el beneficio máximo) para el problema de redes de ejemplo es \$93,300.00 .

B 6 7 0 0 T E M P 0 / N E T W O R K

VERSION : 27.101.00

RELEASED : AUG 76

DATE : 2/24/81

TIME : 15143.7

BUILD
ZDATA="REDES"
ZNAME="REDES"
INPUT
BCDOUT
SETUP(MAX)
OPTIMIZE
OUTPUT
EXIT

123456789

TEMPO/NETWORK

BUILD TIME--PROCFSSUR = 0.04 ELAPSED= 0.15

INPUT TIME--PROCFSSUR = 0.04 ELAPSED= 0.17

ZDATA = RECES ZNAME = REDFS

NEW ENTRY REDES ENTERED ON ZPROF

PROBLEM STATISTICS: 9 NODES, 17 ARCS, 9 NODE RESTRICTIONS.
 0 MINOR ERRORS, 0 MAJOR ERRORS.

BCDOUT TIME--PROCFSSUR = 0.04 ELAPSED= 0.44

NAME	REDES			
TAMPICO		400000.		
TAMPICO	VERACRUZ	.03000		
TAMPICO	SAN LUIS	.11000	200000.	100000.
TAMPICO	DF	.09000		
TAMPICO	GUADALAJ	.13000		
TAMPICO	-	.40000		
MUNTERR		150000.		
MUNTERR	VERACRUZ	.06000		
MUNTERR	SAN LUIS	.03000		
MUNTERR	DF	.12000		
MUNTERR	GUADALAJ	.08000		
MUNTERR	-	.35000		
CULIMA		250000.		
CULIMA	VERACRUZ	.10000		
CULIMA	SAN LUIS	.07000		
CULIMA	DF	.04000		
CULIMA	GUADALAJ	.03000		
CULIMA	-	.30000		
CALIF		200000.		
CALIF	SAN LUIS	.22000		
CALIF	GUADALAJ	.24000		
CALIF	DURANGO	.10000		
CALIF	-	.20000		
DURANGC				
DURANGC	SAN LUIS	.08000		
DURANGC	GUADALAJ	.08000		
DURANGC	*	.02000	150000.	
SAN LUIS			220000.	
SAN LUIS	+	.52000		
GUADALAJ			240000.	
GUADALAJ	+	.46000		
VERACHUZ			280000.	
VERACHUZ	+	.54000		

DF
DF
ENDATA

+

.48000

260000.

SETUP TIME--PROCFSSOR = 0.09 ELAPSED= 0.47

ZNAME = REDES

TOTAL NUMBER OF NODES : 10 TOTAL NUMBER OF ARCS: 25
NO. OF NODES INPUT : 9
NO. OF RESTRICTED NODES: 9 NET SUPPLY
NO. OF ARTIFICIAL NODES: 1 AND DEMAND: 0

OPTIMIZE TIME--PROCFSSOR = 0.10 ELAPSED= 0.47

OPTIMIZE TIME--PROCFSSOR = 0.10 ELAPSED= 0.48

OPTIMIZE TIME--PROCFSSOR = 0.10 ELAPSED= 0.48

PASS III - OPTIMIZE THE BASIS

ITERATION NUMBER	ARC	ARC	REDUCED COST	AMOUNT OF FLOW	NUMBER OF ART. ARCS	OBJECTIVE VALUE
40	(13= 5) (17= 7)	0.11000	50000	1	66300.00000
41	(11= 15) (15= 5)	0.17000	300000	1	71400.00000
42	(7= 17) (17= 11)	0.06000	200000	1	83400.00000
43	(7= 20) (19= 20)	0.02000	150000	1	86400.00000
44	(12= 22) (1= 2)	0.07000	0	1	86400.00000
45	(1= 2) (1= 2)	0.07000	0	1	86400.00000
46	(3= 7) (3= 11)	0.06000	10000	1	87000.00000
47	(13= 7) (3= 7)	0.11000	10000	1	88100.00000
48	(9= 15) (15= 7)	0.12000	10000	1	89300.00000
49	(11= 17) (17= 7)	0.06000	50000	1	92300.00000
50	(11= 20) (13= 9)	0.08000	0	1	92300.00000
51	(7= 17) (17= 11)	0.02000	50000	1	93300.00000

145

OUTPUT TIME--PROCFSSOR = 0.10 ELAPSED= 0.49

OUTPUT

PROBLEM NAME REDES
SOLUTION STATUS OPTIMAL
FUNCTIONAL VALUE 93300.00000
NO. ARTIFICIAL ARCS

TEMPO/NETWORK

NODE: TAMPICO
 TYPE: SOURCE
 SUPPLY AT THE NODE : 400000
 FLOW INTO THE NODE : 0
 INTERNAL NUMBER : 3
 DEMAND AT THE NODE : 0
 FLOW FROM THE NODE : 400000
 ARC SUMMARY
 DESTINATION NUMBER STATUS ACTIVITY INPUT COST UPPER LIMIT LOWER LIMIT REDUCED COST
 VERACRUZ 5 UL 200000 -0.0300 200000 100000 0.08000
 DF 9 BS 200000 -0.0900 NONE NONE 0.00000

RESTRICTION SUMMARY
 TYPE NUMBER STATUS ACTIVITY COST/PRICE UPPER LIMIT LOWER LIMIT
 SUPPLY 4 BS 400000 -0.4000 NONE NONE

NODE: MONTERR
 TYPE: SOURCE
 SUPPLY AT THE NODE : 150000
 FLOW INTO THE NODE : 0
 INTERNAL NUMBER : 13
 DEMAND AT THE NODE : 0
 FLOW FROM THE NODE : 150000
 ARC SUMMARY
 DESTINATION NUMBER STATUS ACTIVITY INPUT COST UPPER LIMIT LOWER LIMIT REDUCED COST
 VERACRUZ 5 BS 80000 -0.0600 NONE NONE -0.00000
 SAN LUIS 7 BS 70000 -0.0300 NONE NONE 0.00000

RESTRICTION SUMMARY
 TYPE NUMBER STATUS ACTIVITY COST/PRICE UPPER LIMIT LOWER LIMIT
 SUPPLY 14 BS 150000 -0.3500 NONE NONE

NODE: CULIMA
 TYPE: SOURCE
 SUPPLY AT THE NODE : 250000
 FLOW INTO THE NODE : 0
 INTERNAL NUMBER : 15
 DEMAND AT THE NODE : 0
 FLOW FROM THE NODE : 250000
 ARC SUMMARY
 DESTINATION NUMBER STATUS ACTIVITY INPUT COST UPPER LIMIT LOWER LIMIT REDUCED COST
 DF 9 BS 60000 -0.0400 NONE NONE -0.00000
 GUADALAJ 11 BS 190000 -0.0300 NONE NONE 0.00000

RESTRICTION SUMMARY
 TYPE NUMBER STATUS ACTIVITY COST/PRICE UPPER LIMIT LOWER LIMIT
 SUPPLY 16 BS 250000 -0.3000 NONE NONE

NODE: CALIF
 TYPE: SOURCE
 SUPPLY AT THE NODE : 200000
 FLOW INTO THE NODE : 0
 INTERNAL NUMBER : 17
 DEMAND AT THE NODE : 0
 FLOW FROM THE NODE : 200000
 ARC SUMMARY
 DESTINATION NUMBER STATUS ACTIVITY INPUT COST UPPER LIMIT LOWER LIMIT REDUCED COST
 SAN LUIS 7 BS 50000 -0.2200 NONE NONE 0.00000
 DURANGO 19 BS 150000 -0.1000 NONE NONE 0.00000

RESTRICTION SUMMARY
 TYPE NUMBER STATUS ACTIVITY COST/PRICE UPPER LIMIT LOWER LIMIT
 SUPPLY 18 BS 200000 -0.2000 NONE NONE

TEMPO/NETWORK

NODE: DURANGO
 TYPE: TRANSSHIP
 SUPPLY AT THE NODE : 0
 FLOW INTO THE NODE : 150000
 INTERNAL NUMBER : 19
 DEMAND AT THE NODE : 0
 FLOW FROM THE NODE : 150000
 ARC SUMMARY
 ORIGINATOR NUMBER STATUS ACTIVITY INPUT COST UPPER LIMIT LOWER LIMIT REDUCED COST
 CALIF 17 BS 150000 -0.1000 NONE NONE 0.00000
 DESTINATION
 SAN LUIS 7 BS 100000 -0.0800 NONE NONE 0.00000
 GUADALAJ 11 BS 50000 -0.0800 NONE NONE 0.00000
 RESTRICTION SUMMARY
 TYPE NUMBER STATUS ACTIVITY COST/PRICE UPPER LIMIT LOWER LIMIT
 TRANSSHIP 20 UL 150000 -0.0200 150000 NONE

NODE: SAN LUIS
 TYPE: SINK
 SUPPLY AT THE NODE : 0
 FLOW INTO THE NODE : 220000
 INTERNAL NUMBER : 7
 DEMAND AT THE NODE : 220000
 FLOW FROM THE NODE : 0
 ARC SUMMARY
 ORIGINATOR NUMBER STATUS ACTIVITY INPUT COST UPPER LIMIT LOWER LIMIT REDUCED COST
 DURANGO 19 BS 100000 -0.0800 NONE NONE 0.00000
 CALIF 17 BS 50000 -0.2200 NONE NONE 0.00000
 MONTERR 13 BS 70000 -0.0300 NONE NONE 0.00000
 RESTRICTION SUMMARY
 TYPE NUMBER STATUS ACTIVITY COST/PRICE UPPER LIMIT LOWER LIMIT
 DEMAND 8 BS 220000 0.5200 NONE NONE

NODE: GUADALAJ
 TYPE: SINK
 SUPPLY AT THE NODE : 0
 FLOW INTO THE NODE : 240000
 INTERNAL NUMBER : 11
 DEMAND AT THE NODE : 240000
 FLOW FROM THE NODE : 0
 ARC SUMMARY
 ORIGINATOR NUMBER STATUS ACTIVITY INPUT COST UPPER LIMIT LOWER LIMIT REDUCED COST
 DURANGO 19 BS 50000 -0.0800 NONE NONE 0.00000
 CULIMA 15 BS 190000 -0.0300 NONE NONE 0.00000
 RESTRICTION SUMMARY
 TYPE NUMBER STATUS ACTIVITY COST/PRICE UPPER LIMIT LOWER LIMIT
 DEMAND 12 BS 240000 0.4600 NONE NONE

NODE: VERACRUZ
 TYPE: SINK
 SUPPLY AT THE NODE : 0
 FLOW INTO THE NODE : 280000
 INTERNAL NUMBER : 5
 DEMAND AT THE NODE : 280000
 FLOW FROM THE NODE : 0
 ARC SUMMARY
 ORIGINATOR NUMBER STATUS ACTIVITY INPUT COST UPPER LIMIT LOWER LIMIT REDUCED COST
 MONTERR 13 BS 80000 -0.0600 NONE NONE 0.00000
 TAMPICO 3 UL 200000 -0.0300 200000 100000 0.00000

RESTRICTION SUMMARY
 TYPE DEMAND NUMBER 6 STATUS BS ACTIVITY 280000 COST/PRICE 0.5400 UPPER LIMIT NONE LOWER LIMIT NONE

NODE: DF
 TYPE: SINK
 SUPPLY AT THE NODE : 0
 FLOW INTO THE NODE : 260000
 INTERNAL NUMBER : 9
 DEMAND AT THE NODE : 260000
 FLOW FROM THE NODE : 0

ARC SUMMARY
 ORIGINATOR NUMBER STATUS ACTIVITY INPUT COST UPPER LIMIT LOWER LIMIT REDUCED COST
 CULIMA 15 BS 60000 -0.0400 NONE NONE -0.00000
 TAMPICO 3 BS 200000 -0.0900 NONE NONE 0.00000


RESTRICTION SUMMARY
 TYPE DEMAND NUMBER 10 STATUS BS ACTIVITY 260000 COST/PRICE 0.4800 UPPER LIMIT NONE LOWER LIMIT NONE

NODE: SUPER*N*
 TYPE: TRANSSHIP
 SUPPLY AT THE NODE : 0
 FLOW INTO THE NODE : 0
 INTERNAL NUMBER : 1
 DEMAND AT THE NODE : 0
 FLOW FROM THE NODE : 0

ENDRUN TIME-PROCESSOR = 0.12 ELAPSED= 0.51

8.0

L I M I T A C I O N E S

- 1/ El paquete TEMPO/NETWORK no se puede acceder en forma -
interactiva .
 - 2/ El paquete se puede utilizar solo para resolver modelos
de redes deterministas .
 - 3/ El paquete no resuelve problemas de redes con ganancias .
 - 4/ Continuamente no se encuentra cargado el paquete en el -
sistema .
 - 5/ Requiere que el usuario tenga experiencia en la solución
de problemas de redes .
 - 6/ Requiere que el usuario conozca programación lineal y -
algunas de sus aplicaciones ; especialmente aplicaciones
en redes .
- 

C A P I T U L O I V

M A N E J O D E L P A Q U E T E

B A S I S / B A S I S

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

P R O B L E M A S D E

E S T A D I S T I C A

1.0 INTRODUCCION A BASIS / BASIS

La naturaleza de los problemas en estadística .- La estadística juega una regla indispensable en todas las posibles esferas de la actividad humana en el mundo moderno . Los --- aspectos inferenciales de metodos estadisticos se han hecho - esenciales para acoplar esta herramienta en un marco cientifico.

Análisis de regresión , estimación , pruebas de --- hipotesis , diseño y análisis de experimentos , muestreo , -- clasificación y agrupamiento de datos , y análisis de series de tiempo son de los mayores metodos estadisticos que se han aplicado en diversos campos . Tales aplicaciones han contribuido al engrandecimiento de la teoria y metodos inferenciales ---- basados en datos .

El problema en estimación .- El problema general de la estimación es el de seleccionar una función de densidad -- correspondiente a la familia especificada de funciones de --- densidad de los datos observados . Para este proposito se --- utiliza una función de las observaciones llamada "estimador" la cual define el valor de la estimación para un dato observado dado , es el estimado de la función de densidad desconocida . Cuando unicamente interesa estimar ciertos parametros de la - función de densidad , se podria no estimar la función de ---- densidad por completo . Tales problemas son en verdad problemas de optimización , como lo menciona Rao en Linear Statistical Inference and Its Applications :

Para determinar un estimador requerimos un conjunto de criterios los cuales nos ayuden a formar un juicio

Intuitivamente por medio de un estimador de un -- parametro θ de una función T de las observaciones (x_1, \dots, x_n) las cuales encierrán el valor verdadero en -- algun sentido . Extendiendo el criterio de estimación uno asume que se puede proveer una medida del estimador del valor verdadero que encierra el parametro y que -- impone una restricción aceptable sobre la clase de --- estimadores .

Un estimador optimo en la clase restringida de estimadores esta determinado por la minimización de la medida del -- estimador del valor verdadero que encierra a el parametro.

Pruebas de hipótesis estadísticas.- El problema de hacer pruebas de hipótesis estadísticas fue considerado ----- originalmente por Neyman y Pearson (1936) . La conexión entre el conocido lema de Neyman-Pearson para construir pruebas --- uniformemente más potentes de una hipótesis simple, tienen una alternativa particular y es con los modelos de optimización - lineal . [1]

Análisis de regresión .- Los problemas en análisis de regresión estan relacionados con la predicción de una variable llamada "variable dependiente" , sobre la base de información que proveen otras ciertas variables llamadas "variables --- independientes" . Una función $f(x_1, \dots, x_p)$ de las variables - independientes $X=(x_1, \dots, x_p)$ es llamada una predicción de una variable dependiente Y que se considera . Otra vez estos ---- criterios se pueden considerar por medio de optimización . Por ejemplo el tratamiento clasico de optimización es el de minimizar el error cuadrado medio , es decir minimizar $E(Y-f(X))^2$.

Asi mismo la necesidad de datos dignos de de ----- confianza para entender mejor el mundo en el cual vivimos es basica y los estadisticos han considerado metodos para la --- colección de tales datos . Información sobre una población - puede ser coleccionada mediante la enumeración completa de la población o mediante una muestra de la población . El costo a que conduce utilizar una muestra es menor que el costo de la enumeración de la población completa . Aunque con el uso de - una muestra la precisión disminuye , se tiene la ventaja de - que el costo disminuye , asi como el tiempo en que se lleva a cabo la colección de la información , aunado a esto si se -- recolecto mucha información el procesarla a mano sera casi -- imposible y el desarrollar programas llevara mucho más tiempo aumentando el costo a medida que pasa este . Por lo que es -- conveniente mejor utilizar los paquetes disponibles que hay - para estadística . Para un estudio más profundo se sugieren - las referencias [5], [7] , [11] , [12] , [13] , [14] y [4] .

[1] Para ver este problema de la relación entre Estadística y optimización se puede ver Linear Programming and Extensions (1963) de Dantsig's .

En este capítulo se tratara el manejo del paquete - BASIS/BASIS en la solución de problemas de estadística . El - cual esta diseñado para ser utilizado por personas que no --- necesariamente tengán experiencia en computación , pero si en estadística . Ya que el paquete esta constituido por un --- conjunto de rutinas estadísticas que el usuario puede acceder en la computadora Burroughs , para obtener estadísticas de sus archivos de datos . El usuario del paquete tiene acceso a los siguientes 23 análisis : ■■

- 1/ AID .- DETECCION INTERACTIVA AUTOMATICA .
- 2/ ANOVAN .- ANALISIS DE VARIANZA .
- 3/ BUILD .- CREACION DE ARCHIVOS DE DATOS TRANSFORMADOS
- 4/ CANCOR .- ANALISIS DE CORRELACION CANONICA .
- 5/ CHISQR .- PRUEBA DE BONDAD DE AJUSTE CHI-CUADRADO .
- 6/ CORREL .- ANALISIS DE CORRELACION LINEAL .
- 7/ FACTOR .- ANALISIS FACTORIAL .
- 8/ LOOK .- INSPECCION DE DATOS .
- 9/ MDISC .- ANALISIS DISCRIMINANTE .
- 10/ MISDAT .- TABULACION DE DATOS CONSIDERADOS FALTANTES .
- 11/ MOMENT .- CALCULO DE MOMENTOS DE ORDEN MAYOR ..
- 12/ MULTR .- REGRESION LINEAL MULTIPLE .
- 13/ PLOT .- GRAFICACION DE DATOS .
- 14/ POLY .- REGRESION LINEAL POLINOMIAL .
- 15/ QTEST .- PRUEBA Q-COCHRAN .
- 16/ SRANK .- CORRELACION DE RANGO DE SPEARMAN .
- 17/ STAT .- CALCULO DE ESTADISTICAS BASICAS .
- 18/ STEPR .- REGRESION LINEAL STEPWISE .
- 19/ TSA .- ANALISIS DE SERIES DE TIEMPO .
- 20/ TTEST .- PRUEBA T-STUDENT .
- 21/ UNIVAR .- HISTOGRAMAS Y TABLAS DE FRECUENCIAS .
- 22/ WILCOX .- PRUEBA WILCOX .
- 23/ XCLASS .- ANALISIS DE TABLAS DE CONTINGENCIA .

■ El nombre de BASIS del paquete es referido de las iniciales de BURROUGHS ADVANCED STATISTICAL INQUIRY SYSTEM ,

El paquete en la solución de problemas de estadística considera tres reportes principales ; y son los siguientes :

- 1/ Descripción de datos de entrada .
- 2/ Definición del análisis estadístico .
- 3/ ejecución del trabajo .

La estructura general de estos reportes se muestra a continuación , para posteriormente acoplarlos y ver como se - utilizarán en la solución de problemas de estadística .

1/ Descripción de datos de entrada .- La definición de los datos , describe el archivo de datos . En estas instrucciones se asigna el nombre del archivo de datos y el formato utilizado en el archivo . Se pueden definir como máximo 340 variables , la definición puede contener la siguiente información .

- 1.1/ Título .
- 1.2/ Comentarios .
- 1.3/ Etiquetas para las variables .
- 1.4/ Relaciones de datos faltantes .

La estructura general que se considera en la definición de datos es la siguiente : **■**

```

Definición del nombre:      DATA <nombre> <disponibilidad>;
Especificación de archivo:  FILE <BINARY> <nombre> <parametros>;
Variables :
    Formato fijo             (<definición del campo>) "<etiquet_
                             ta>" = <relación de datos faltantes>;
    Formato libre           <posición> '<etiqueta>' = <relación de
                             datos faltantes>;
    Binario                 (<definición del campo>) "<etiquet_
                             ta>" = <relación de datos faltantes>;
COMMENT (es opcional):     COMMENT <comentario>;
TITLE (es opcional):       TITLE "<título>";
termina la definición:     END;

```

Un ejemplo de definición de datos se mostrara a -- continuación.

■ Los datos que se encuentran encerrados entre paréntesis de pico serán asignados por el usuario .

Ejemplo de definición de datos :

```

Definición del nombre:      DATA DATOS;
Especificación de archivo:  FILE EJE;
Definición de variables:   (1,4) "X" = -99;
    (Formato fijo)         De la columna 1-4 ira un valor entero
                           llamado X , el cual si es -99 se le
                           considerara como valor faltante.
                           (8,3) "Y";
                           De la columna 8-10 ira un valor entero
                           llamado Y.
                           (14,A3) "Z" = BLANK ;
                           De la columna 14-16 ira una variable
                           alfabetica llamada Z ; si el campo -
                           esta en blanco se considera valor --
                           faltante .
                           (2/6,XXX) "W" LE 0;
                           En una segunda tarjeta de la columna
                           6-8 ira una variable entera , la cual
                           si es menor o igual a cero sera tratada
                           como valor faltante .
                           (2/10,6.3) "XX";
                           En la segunda tarjeta de la columna -
                           10-15 ira un valor real llamado XX y
                           cuyo punto decimal estara en la
                           columna 12 .
COMMENT (es opcional):     COMMENT DESCRIPCION ;
TITLE (es opcional):      TITLE " CIA. ACME " ;
Termina la definición:    END;

```

Descripción de datos que asigna el usuario en la --
definición de datos :

1/ En la instrucción DATA , se asigna el nombre de la --
definición de datos .

1.1/ <nombre> .- Nombre con el que el usuario se referira
al conjunto de datos ; debe ser a lo más
de 6 caracteres alfanumericos , empezando
con un caracter alfabetico .

1.2/ <disponibilidad> .- Se utiliza para almacenar la definición de datos en disco .

1.2.1/ Si no se utiliza la disponibilidad , la - definición de datos se destruye al finalizar la ejecución.

1.2.2/ LIBRARY .- La definición de datos se almacena en forma permanente .

1.2.3/ LIBRARY "FECHA" .- La definición de datos se almacena en disco hasta la fecha indicada .

2/ En la instrucción FILE , se asigna el nombre del archivo de datos que sera procesado por el paquete , puede estar en tarjetas .

2.1/ <BINARY> .- Se utiliza unicamente si los datos se --- encontrarán en cinta , definidos en palabras enteras .

2.2/ <nombre> .- Nombre con el que se encuentra grabado el archivo de datos en disco , o tarjetas .

2.3/ <parametros> .- Se utiliza unicamente en el caso de que se requiera especificar las características con que fue grabado el archivo . La esp. de parametros tiene la siguiente -- estructura : (A,B,C) .

2.3.1/ A .- Número de registros por observación , a lo más 15.

2.3.2/ B .- Número de palabras por -- registro , a lo más 1022.

2.3.3/ C .- Número de palabras por -- bloque , a lo más 1022.

3/ La instrucción de definición de variables indica el formato en que se encontrarán los datos .

3.1/ La forma de las instrucciones en formato fijo es :

3.1.1/ <definición del campo> .- La definición del

Un * si los datos están en tarjetas o se daran en formato libre en la terminal .

campo tiene la siguiente forma (A/B,C).

3.1.1.1/ A .- Es un número entero especificando el registro en que se encuentra la observación se especifica solo que la observación no se encuentre en el primer registro.

3.1.1.2/ B .- Es un número entero que indica la -- columna en que empieza el campo que - utiliza la observación .

3.1.1.3/ C .- Especifica el formato de la variable .

3.1.1.3.1/ w.d .- w es el número de -- caracteres incluyendo el punto y el signo , d es el número de decimales .

3.1.1.3.2/ Aw .- Indica w caracteres - alfabéticos .

3.1.2/ <etiqueta> .- Se pueden utilizar hasta 12 caracteres alfanuméricos .

3.1.3/ <relación de datos faltantes> .- Es opcional y es la especificación del valor o cadena alfanumérica que denota los datos faltantes de la observación .

3.1.3.1/ BLANCK .- Si el campo de la observación esta en blanco , se considera valor faltante .

3.1.3.2/ Relaciones .- Se utilizarán las mismas --- de fortrán , sin los puntos.

3.2/ La forma de las instrucciones en formato libre es:

3.2.1/ <posición> .- Un número entero indicando la posición de la observación , en orden secuencial . Los otros datos que asigna el usuario son en forma similar a los de formato fijo.

4/ La tarjeta de comentarios se utiliza para documentar la definición de datos y no tiene efecto en el proceso del paquete .

4.1/ comentario .- Cuerda de caracteres alfanuméricos .

5/ La tarjeta de título puede utilizar hasta la columna 72 para describir el título de la definición de datos .

- 2/ Definición del análisis estadístico .- La definición del análisis estadístico se lleva a cabo por medio de un --- procedimiento . En donde el procedimiento utiliza uno de los 23 análisis con que cuenta BASIS además de otras --- instrucciones , las cuales indican como serán procesadas las observaciones por el análisis . Las instrucciones --- opcionales permitidas son las siguientes :
- 2.1/ Instrucciones para la manipulación de datos .
- 2.1.1/ Filtración de observaciones .
- 2.1.2/ Transformaciones de variables .
- 2.2/ Instrucciones de control sobre el procedimiento estadístico .
- 2.2.1/ Asignación de opciones .
- 2.2.2/ Asignación de valores a los parametros .
- 2.2.3/ Identificación de variables con atributos específicos .
- 2.3/ Instrucciones de salida de reportes .
- 2.3.1/ Especificación de reportes .
- 2.3.2/ Destino de salida de reportes (terminal o impresora).
- 2.4/ Títulos .
- 2.5/ Comentarios .

La estructura general que se considera en la definición de un procedimiento es la siguiente : ■■

Definición del nombre:	PROCEDURE <nombre> <disponibilidad>;
Programa de análisis:	USE <análisis>;
Selección de variables:	{ INPUT LIST <lista de variables>; INPUT PAIR(<pareja de variables>; INPUT GROUP (<grupo de variables>;
Control del análisis: (opcional)	{ OPTION <lista de opciones>; <nombre del parametro asignado> = <valor>; <nombre de atributo> <variable>; { INCLUDE o EXCLUDE } <filtro>;

■■ Los datos que se encuentran encerrados entre paréntesis de pico serán asignados por el usuario .

Manipulación de datos: (opcional)	{ <variable> = <transformación>; <variable temporal> IS "<etiqueta>";
Salidas de reportes: (opcional)	{ PRINT o } <lista de reportes>; DISPLAY
COMMENT (es opcional):	COMMENT <comentario>;
TITLE (es opcional):	TITLE "<título>";
Termina la definición:	END;

Un ejemplo de definición de procedimiento se mostrara a continuación :

Ejemplo de definición de procedimiento :

Definición del nombre:	PROCEDURE REGRE;
Programa de análisis:	USE MULTR;
Selección de variables:	INPUT GROUP ("X","Y");
Control del análisis:	OPTION INVERSE;
Manipulación de datos:	{ TS = "X" - "Y"; TS IS "DIFER";
Salidas de reportes:	PRINT ALL;
Termina la definición:	END;

Descripción de datos que asigna el usuario en la -- definición de un procedimiento :

- 1/ En la instrucción PROCEDURE , se asigna el nombre de la definición del procedimiento .
 - 1.1/ nombre .- Nombre con el que el usuario se referira al procedimiento ; debe ser a lo más de 6 caracteres alfanumericos , empezando con un caracter alfabetico .
 - 1.2/ disponibilidad .- Similar al de DATA .
- 2/ En la instrucción USE se utiliza uno de los 23 análisis.
 - 2.1/ análisis .- Nombre de uno de los 23 análisis con que cuenta el paquete .
- 3/ La instrucción INPUT se utiliza para especificar las --- variables que serán analizadas por el programa estadístico designado en USE .

- 3.1/ <lista de variables> .- La lista de variables -
identifica a las variables que serán procesadas
por el programa estadístico . Si se especifica
más de una variable , los identificadores de las
variables deberán estar separados por comas .
- 3.2/ <pareja de variables>.- La pareja de variables -
identifica a la pareja que sera procesada por el
programa estadístico . Cada pareja consite de dos
identificadores de variables separados por una
coma y encerrados entre parentecis .
- 3.3/ <grupo de variables>.- El grupo de variables --
identifica un grupo de variables que serán ---
procesadas por la rutina estadística . Cada --
grupo consiste de una lista de variables ----
encerradas entre parentecis .
- 4/ <nombre del parametro asignado>.- Esta instrucción --
permite al usuario asignar un valor especifico a un
parametro asociado con el programa estadístico .
- 5/ <valor>.- Es el valor asignado al parametro por el -
usuario y puede ser numerico o alfabetico . Si es --
alfabetico debe ir encerrado entre comillas .
- 6/ <nombre de atributo>.- Esta instrucción identifica una
variable que tiené una carcteristica especifica .
- 7/ <variable>.- Especifica que variable tiene el atributo.
- B/ <filtro>.- Especifica que variables serán incluidas o
excluidas del análisis .
- 9/ <variable>.- Identifica la variable que sera modificada
o creada como resultado de una transformación .
- 10/ <variable temporal>.- Identifica a la variable que fue
creada por medio de una transformación .
- 11/ <etiqueta>.- Nombre asignado a la variable temporal .
- 12/ <lista de reportes>.- Indica que reportes son requeridos.
ALL indica que se impriman todos los reportes .
- 13/ COMMENT y TITLE .- Similares a los de definición de datos.

3/ Ejecución del trabajo .- El trabajo que se define combinando una definición de datos, con una definición de procedimiento con los cuales se realiza el análisis estadístico . El modulo de trabajo se puede ejecutar inmediatamente .

La estructura que se considera en la ejecución del - trabajo es la siguiente :

```
EXECUTE PROCEDURE <nombre de definición de procedimiento> WITH
DATA <nombre de definición de un conjunto de datos>;
```

Como ejemplo considere la ejecución del ejemplo de definición de procedimiento , y el ejemplo de definición de datos.

```
EXECUTE PROCEDURE REGRE WITH DATA DATOS ;
```

Descripción de datos que asigna el usuario para la ejecución de un procedimiento con un conjunto de datos :

- 1/ <nombre de definición de procedimiento>.- El nombre asignado a un procedimiento que se definio .
- 2/ <nombre de definición de un conjunto de datos>.- el nombre asignado a una definición de datos .

El paquete tambien considera un conjunto de instrucciones por medio de las cuales el usuario se puede comunicar al -- sistema , con el siguiente proposito .

- 1/ Borrar una definición que se empieza , o ya definida .
- 2/ Para obtener ayuda .
- 3/ Para acceder un archivo de datos por medio del paquete.

Estas instrucciones son las siguientes :

ABORT

Esta instrucción se utiliza unicamente cuando el paquete se accesa desde una terminal remota , se utiliza antes de la ---- instrucción END , para que no se considere la definición que se - estaba llevando a cabo .

EXPLAIN

Esta instrucción se utiliza para obtener una documentación interna del paquete .

EXPLAIN

ALL .- Explicación completa del paquete .
 ANALYSIS .- Lista de análisis que tiene BASIS .
 BASIS .- Descripción del paquete .
 CHANGES .- Lista de cambios hechos .
 CONTROL .- Información acerca de instrucciones de control . ;
 DATA .- Definición de datos .
 FILE .- Comandos de mantenimiento .
 JOB .- Definición de un trabajo .
 PROCEDURE .- Definición de procedimiento .
 <análisis> .- Documentación de uno de los 23 análisis .

En donde <análisis> , es indicado por medio de uno de los 23 análisis con que cuenta el paquete .

DATAFOLLOWS

Se utiliza unicamente cuando el paquete se accesa por medio de tarjetas de datos , y la instrucción de FILE se define con un - , esta instrucción indica el inicio del archivo de datos del que se obtendra un análisis estadístico .

2.0 EJEMPLOS DE APLICACION

En la tabla 4.1 se muestra un ejemplo de un análisis estadístico . En el ejemplo se utiliza un análisis de regresión sobre un conjunto de datos . El proceso seguido para hacer un análisis de regresión fue el siguiente :

- 1/ Tarjetas de control del sistema .- Estas tarjetas son las que en la columna uno tienen un caracter "?" . Y se utilizarán para acceder el paquete .
- 2/ Tarjetas de instrucciones .- En estas tarjetas se indica el proceso a seguir , para llevar a cabo un análisis estadístico .
 - 2.1/ La instrucción USER se utiliza para reservar area de memoria en el disco , se utiliza un número entero a lo más de tres digitos .
 - 2.2/ Se define un procedimiento de datos , al que se le llamo DATOS , indicando que en los datos se leerán dos variables en formato libre .
 - 2.3/ Se define un procedimiento de análisis estadístico llamado REGRE ; en el que se indica que el análisis

a utilizar es el de regresión lineal múltiple , en el cual la variable 1 ; se considerara como la variable dependiente , y la variable 2 como --- independiente . De opción se pide la inversa de la matriz normal y de la de correlación .

- 3/ Se ejecuta el procedimiento REGRE con la definición del conjunto de datos DATOS .
- 4/ Se dan los datos en formato libre⁴ ; con los cuales se ajustara un modelo de regresión lineal .
- 5/ Se utiliza la instrucción QUIT para indicar que la - ejecución del paquete termine .

En una misma corrida se pueden llevar a cabo varios análisis estadísticos , con varios conjuntos de datos , utilizando la siguiente estructura :

```

USER=<número entero , a lo más de tres dígitos>;
{
  Def. Procedimiento 1
  :
  :
  Def. Procedimiento n
}
{
  Def. De datos 1
  :
  :
  Def. De datos m
}
{
  Ejecución 1
  :
  Ejecución s
}
QUIT;
```

```
?JOB REGRESION;USER=SF82/MP;CLASS=5;BEGIN
?RUNXBASIS/BASIS;EBCDIC USER;
USER=90;
EXPLAIN ALL;
DATA DATOS;
FILE*;
1 "Y";
2 "X";
END;
PROCEDURE REGRE;
USE MULTR;
INPUT GROUP(V2,V1);
OPTION INVERSE;
PRINT ALL;
END;
EXECUTE PROCEDURE REGRE WITH DATA DATOS;
DATAFOLLOWS;
10,11.6
12,13.87
13,13.972
14,14.895
16,17.56
24,25.88
26,27.79
27,28,53
15,16.43
18,18.985
21,22.675
END;
QUIT;
?END JOB
```

Table 4.1

USER=90;
LIBRARY DIRECTORY FOUND

DATA DATCS;

FILE#;

2 "X";

END;

NUMBER OF ERRORS = 0
DEFINITION STORED TEMPORARILY:

PROCEDURE REGRE;

USE MULTIB;

INPUT GROUP(V2,V1);

OPTION INVERSE;

PRINT ALL;

END;

NUMBER OF ERRORS = 0
DEFINITION STORED TEMPORARILY:

EXECUTE PROCEDURE REGRE WITH DATA DATCS;
DATAFOLLOWS;

1335

10525.6

JOB DEF = *****
 DATA DEF = DATCS
 ANALYSIS = MULTP
 DATA SET = 1

OBSERVATIONS: 11 READ, 11 PROCESSED, 0 REJECTED, SLM WEIGHTS = 11

M U L T I P L E L I N E A R R E G R E S S I O N A N A L Y S I S

PROBLEM NUMBER = 1
 SAMPLE SIZE = 11
 INDEPENDENT VARIABLES = 1
 DEPENDENT VARIABLE = Y

VAR	LABEL	MEAN	STD DEVIATION
1	X	15.28973	5.59908
2	Y	17.81818	5.86205

NORMAL MATRIX -

1
 ROW 1
 359.6891

INVERSE OF NORMAL MATRIX -

1
 ROW 1
 0.0028

CORRELATION MATRIX -

1	2	
ROW 1	1.0000	0.9981
ROW 2	0.9984	1.0000

INVERSE OF CORRELATION MATRIX -

1	2	
ROW 1	311.7942	-311.2938
ROW 2	-311.2938	311.7942

JOB DEF = *****
 PROC DEF = REGRES
 DATA DEF = DATA
 ANALYSIS DEF = MCLYR
 DATA SET = 1

OBSERVATIONS: 11 READ 11 PROCESSED 0 REJECTED SUM WEIGHTS= 11

REGRESSION COEFFICIENTS AND STANDARDIZED (BETA) COEFFICIENTS =

VAR	LABEL	REGRESSION COEFFICIENT	BETA COEFFICIENT
1	INTERCEPT	-1.00070	
	X	0.97559	0.99840

STANDARD ERROR OF ESTIMATE = 0.34994
 COEFFICIENT OF DETERMINATION = 0.99674
 COEFFICIENT OF DETERMINATION (ADJ) = 0.99644
 MULTIPLE CORRELATION COEFFICIENT (ADJ) = 0.99840

STANDARD DEVIATIONS AND T VALUES OF COEFFICIENTS =

VAR	LABEL	STD ERROR	STD ERR BETA	T VALUE	CONF
1	X	0.01845	0.01888	52.888	100.000

PARTIAL CORRELATIONS AND R2-DELETE =

VAR	LABEL	PARTIAL CORR	R2-DELETE
1	X	0.99840	1.13430E-09

ANALYSIS OF VARIANCE TABLE

SOURCE OF VARIATION	DEGREES FREEDOM	SUM OF SQUARES	MEAN SQUARE	F RATIO	CONF LEVEL
REGRESSION	1	342.5342	342.5342	2797.1476	100.000
RESIDUAL	9	1.1021	0.1223		
TOTAL	10	343.6364			

TABLE OF RESIDUALS

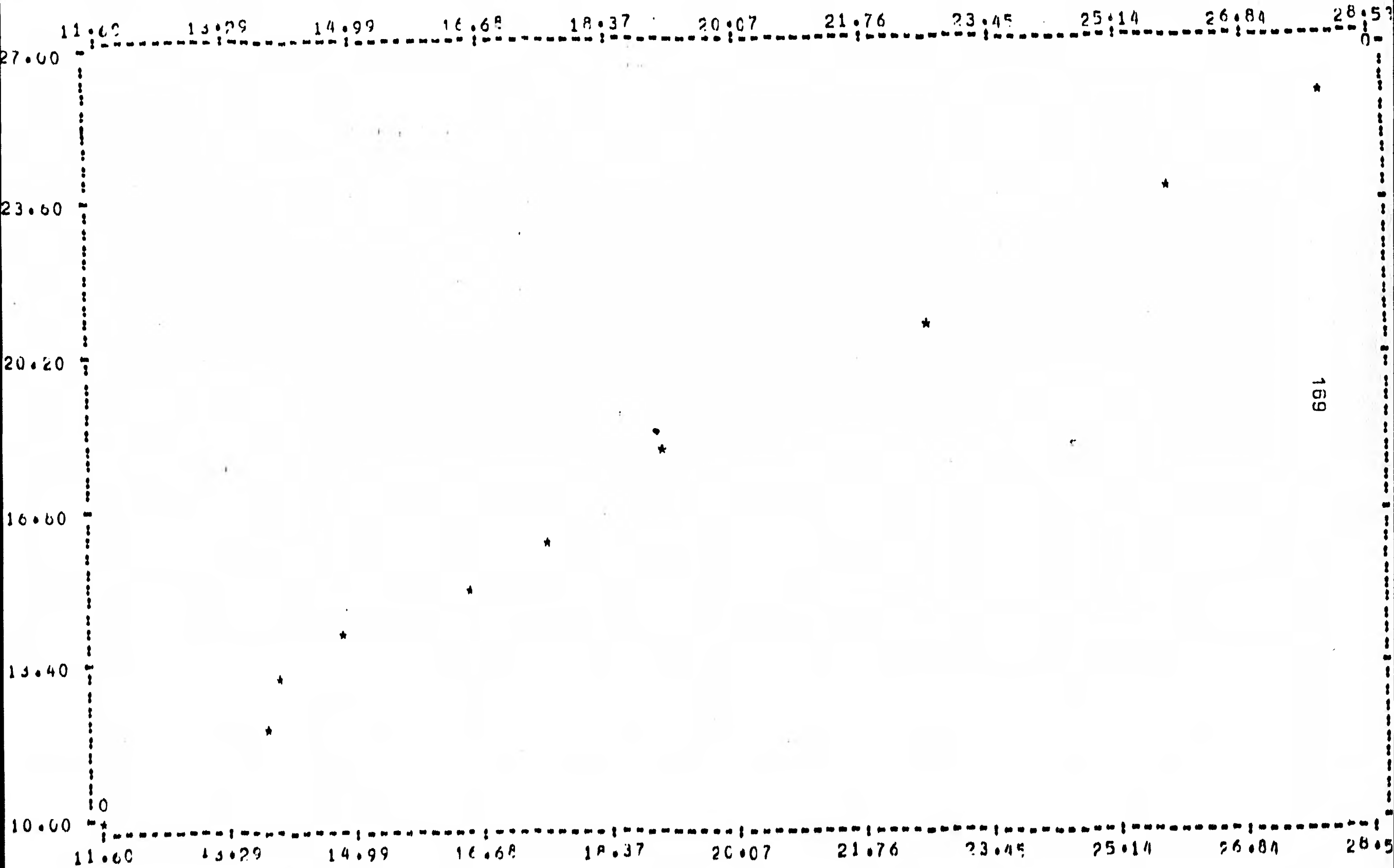
OBS	ACTUAL Y	PREDICTED Y	RESIDUAL
1	10.00000	10.31616	-.31616
2	12.00000	12.53075	-.53075
3	13.00000	13.73028	-.73028
4	14.00000	13.53073	-.46927
5	16.00000	16.13068	-.13068
6	24.00000	24.24759	-.24759
7	26.00000	26.11097	-.11097
8	27.00000	26.83291	-.16709
9	15.00000	15.02826	-.02826
10	16.00000	17.52089	-.47911
11	21.00000	21.12082	-.12082

DURBIN-WATSON D STATISTIC = 1.80735

* REPRESENTS (

X)

Y)



En la tabla 4.2 se muestra un segundo ejemplo de un análisis de regresión lineal múltiple . Llevado a cabo por medio de una terminal remota . El proceso seguido para el análisis estadístico fue el siguiente :

- 1/ La primer instrucción es para accesar el paquete .
- 2/ Se indico que no se requeria una introducción al paquete.
- 3/ Se le dio un codigo de usuario , para el area de trabajo en disco .
- 4/ Se indico que no se requeria el directorio de archivos generados por BASIS , en corridas anteriores .
- 5/ Se declararon dos procedimientos , para llevar a cabo dos análisis estadísticos .
- 6/ Se declaro un solo archivo de datos , el cual estaba residente en disco , con el nombre de LECTURA.
- 7/ Se ejecutaron los dos procedimientos con el mismo --- archivo de datos .
- 8/ Se indico que terminara la ejecución del paquete .

La interpretación de resultados del ejemplo uno y de este se dara posteriormente .

RUN*BASIS/BASIS	(R)
NO	(R)
66	(R)
NO	(R)
PROCEDURE REGRE1;	(R)
USE MULTR;	(R)
INPUT GROUP(V2,V1);	(R)
OPTION ALLSAVE;	(R)
DISPLAY ALL;	(R)
END;	(R)
PROCEDURE REGRE2;	(R)
USE MULTR;	(R)
T2=V1+V2;	(R)
T3= V1*V2;	(R)
INPUT GROUP(T2,T3,V2,V1);	(R)
PRINT ALL;	(R)
END;	(R)
DATA DATO;	(R)
FILE LECTURA;	(R)
1 "Y" = -99;	(R)
2 "X" = 16.43;	(R)
END;	(R)
EXECUTE PROCEDURE REGRE1 WITH DATA DATO;	(R)
EXECUTE PROCEDURE REGRE2 WITH DATA DATO;	(R)
END;	(R)
QUIT;	(R)

Tabla 4.2

RUN *BASIS/BASIS
#RUNNING 7944

*** B6700/B7700 BASIS *** 7.06 DATE 05/21/81 11:54 AM

INTRODUCTION NECESSARY:

??

NO

OK

USER ID=

66

ID= 66 UK

IS DIRECTORY TO BE DISPLAYED:

NO

OK

READY:

PROCEDURE REQUIRE:

USE MULT:

INPUT GROUP(V2,V1):

OPTION ALLSAVE:

DISPLAY ALL:

END:

WAIT=

#7944 (SF82)BAS066/DIRECTY REMOVED ON UFAM2 PR115 .

DEFINITION STORED TEMPORARILY:

PROCEDURE REGRE2;

USE MULT;

T2 = V1 + V2;

T3 = V1 * V2;

INPUT GROUP(T2, T3, V2, V1);

PRINT ALL;

END;

WAIT=

#7944 (SF82)DAS066/DIRECTY REMOVED ON UNAM2 PK115 .

DEFINITION STORED TEMPORARILY!

DATA DATO;

FILE LECTURA;

1 "Y" = -99;

2 "X" = 10.43;

END;

WAIT=

#7944 (SF82)DAS066/DIRECTY REMOVED ON UNAM2 PK115 .

DEFINITION STORED TEMPORARILY!

EXECUTE PROCEDURE REGRE1 WITH DATA DATO;

WAIT=

EXECUTION BEGINS

MULTIPLE LINEAR REGRESSION ANALYSIS

PROBLEM NUMBER = 1
 SAMPLE SIZE = 10
 INDEPENDENT VARIABLES = 1
 DEPENDENT VARIABLE = Y

VAR	LABEL	MEAN	STD DEVIATION
1	X	19.57570	6.24405
2	Y	18.10000	6.10009

COVARIANCE MATRIX -

	1
1	350.8933

CORRELATION MATRIX -

	1	2
1	1.0000	0.9984
2	0.9984	1.0000

VAR	LABEL	REGRESSION COEFFICIENT	BETA COEFFICIENT
	INTERCEPT	-0.99271	
1	X	0.97534	0.99835

STANDARD ERROR OF ESTIMATE	=	0.37102
COEFFICIENT OF DETERMINATION	=	0.99671
COEFFICIENT OF DETERMINATION (ADJ)	=	0.99630
MULTIPLE CORRELATION COEFFICIENT	=	0.99835
MULTIPLE CORRELATION COEFFICIENT (ADJ)	=	0.99815

STANDARD DEVIATIONS AND T VALUES OF COEFFICIENTS -

VAR	LABEL	STD ERROR	STD ERR BETA	T VALUE	CONF
1	X	0.01981	0.02027	49.244	100.000

PARTIAL CORRELATIONS AND R2-DELETE -

VAR	LABEL	PARTIAL CORR	R2-DELETE
1	X	0.99835	5.53184E-10

ANALYSIS OF VARIANCE TABLE

SOURCE OF VARIATION	DEGREES OF FREEDOM	SUM OF SQUARES	MEAN SQUARE	F RATIO	CONF LEVEL

REGRESSION	1	333.7988	333.7988	2424.9275	100.000
RESIDUAL	8	1.1012	0.1377		
TOTAL	9	334.9000			

TABLE OF RESIDUALS

OBS	ACTUAL Y	PREDICTED Y	RESIDUAL
1	10.00000	10.32100	-0.32100
2	12.00000	12.53502	-0.53502
3	13.00000	12.63450	0.36550
4	14.00000	13.53474	0.46526
5	16.00000	16.13401	-0.13401
6	24.00000	24.24882	-0.24882
7	26.00000	26.11171	-0.11171
8	27.00000	26.83346	0.16654
10	18.00000	17.52387	0.47613
11	21.00000	21.12286	-0.12286

DURBIN-WATSON D STATISTIC = 1.62533

DATA DEFINITION FOR OUTPUT FILE IS --

DATA UUC01 LIBRARY

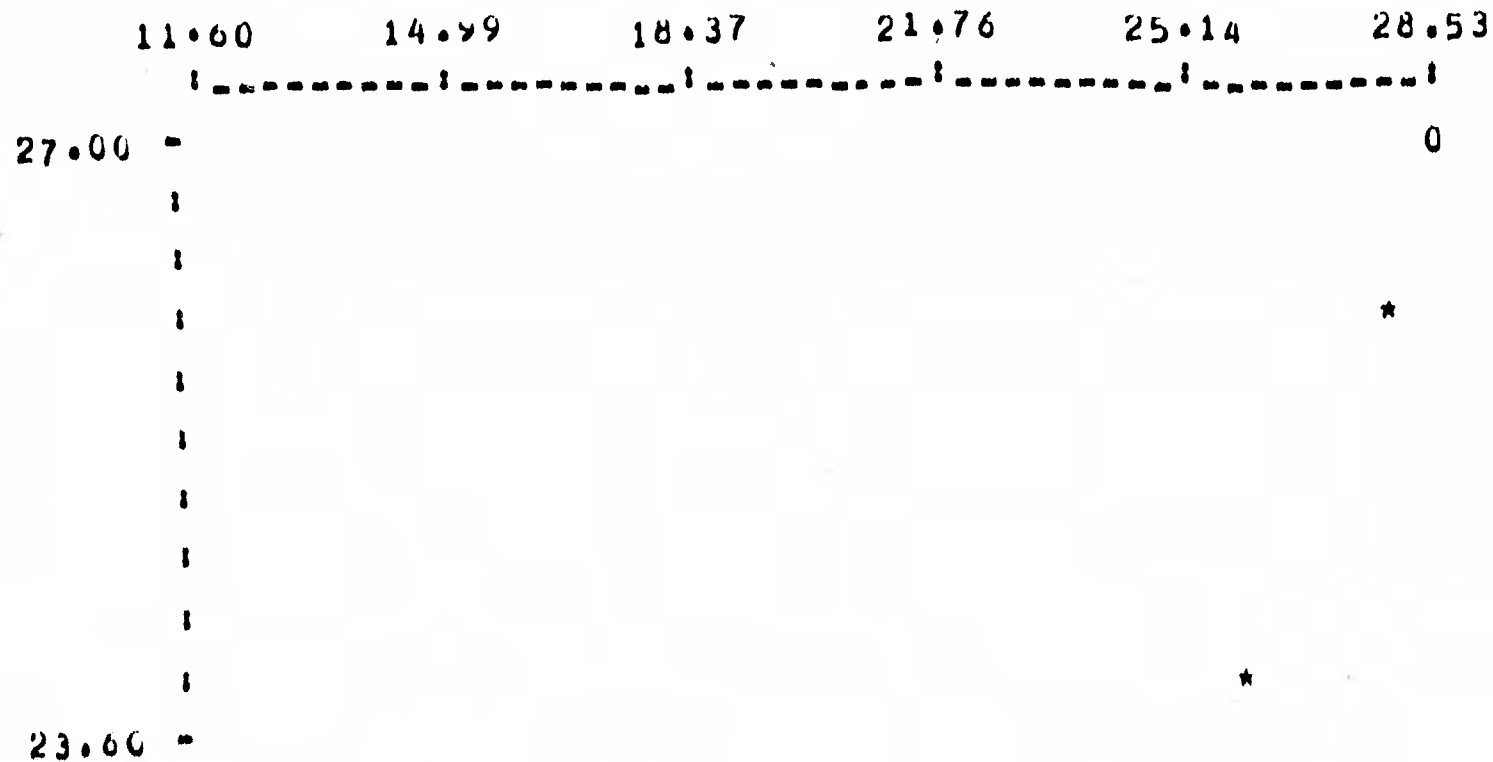
- (1) "X"
- (2) "Y"
- (3) "PREDICTED";
- (4) "RESIDUAL"

#7944 (SF02)DAS066/DIRECTY REMOVED ON UFAM2 PK115 .
 END;

NUMBER OF ERRORS = 0

#7944 (SF02)DAS066/DIRECTY REMOVED ON UFAM2 PK115 .
 DEFINITION STORED IN USERS LIBRARY:

* REPRESENTS (X, Y)



EXECUTION COMPLETED:

END;

QUIT;

NORMAL TERMINATION OF BASIS.

#7944 (SF02)BAS066T/BP02 REMOVED ON UNAM2 PK115 .
#7944 (SF02)BAS066T/BP04 REMOVED ON UNAM2 PK115 .
#7944 (SF02)BAS066T/BD02 REMOVED ON UNAM2 PK115 .
#7944 (SF02)BAS066/DIRECTY REMOVED ON UNAM2 PK115 .
START 11154 AM
STOP 11157 AM
PROCESS 0.27 MIN
I/O 0.21 MIN

#ET=3136.6 F1=17.0 IU=13.1

DATA SET = MCTF
 DATA SET = 1

OBSERVATIONS: 11 READ, 11 PROCESSED, 0 REJECTED, SLM WEIGHTS = 11

MULTIPLE LINEAR REGRESSION ANALYSIS

PROBLEM NUMBER = 1
 SAMPLE SIZE = 10
 INDEPENDENT VARIABLES = 3
 DEPENDENT VARIABLE = Y

VAR	LABEL	MEAN	STD DEVIATION
1	*T2*	37.67570	12.33906
2	*T3*	388.54410	43.05044
3	X	19.57570	6.24405
4	Y	18.10000	6.10009

NORMAL MATRIX

	1	2	3
ROW 1	1370.2719	26836.3051	693.1326
ROW 2	26836.3051	631661.0418	13587.3542
ROW 3	693.1326	13587.3542	350.8933

CORRELATION MATRIX

	1	2	3	4
ROW 1	1			
ROW 2	0.9943	1		
ROW 3	0.9996	0.9948	1	
ROW 4	0.9996	0.9929	0.9984	1

PROC DEF = REGRES2
 DATA DEF = LATEC
 ANALYSIS = MULTR
 DATA SFI = 1

OBSERVATIONS: 11 READ 11 PROCESSED 0 REJECTED SLM WEIGHTS= 11

REGRESSION COEFFICIENTS AND STANDARDIZED (BETA) COEFFICIENTS -

VAR	LABEL	REGRESSION COEFFICIENT	BETA COEFFICIENT
1	INTERCEPT	-2.72412E-08	2.02277
2	*T2*	1.00000	-1.00000E-09
3	*T3*	-4.51905E-11	-1.02360
	X	-1.00000	

STANDARD ERROR OF ESTIMATE = 0.00003
 COEFFICIENT OF DETERMINATION = 1.00000
 COEFFICIENT OF DETERMINATION (ADJ) = 1.00000
 MULTIPLE CORRELATION COEFFICIENT = 1.00000
 MULTIPLE CORRELATION COEFFICIENT (ADJ) = 1.00000

STANDARD DEVIATIONS AND T VALUES OF COEFFICIENTS -

VAR	LABEL	STD ERROR	STD ERR BETA	T VALUE	CONF
1	*T2*	0.00003	0.00006	31807.171	100.000
2	*T3*	4.43677E-07	0.00002	-1.019E-04	0.006
3	X	0.00007	0.00007	-15347.548	100.000

PARTIAL CORRELATIONS AND R2-DELETE -

VAR	LABEL	PARTIAL CORR	R2-DELETE
1	*T2*	1.00000	0.99663
2	*T3*	-0.00004	1.00000
3	X	-1.00000	0.99921

ANALYSIS OF VARIANCE TABLE

SOURCE OF VARIATION	DEGREES OF FREEDOM	SUM OF SQUARES	MEAN SQUARE	F RATIO	LEVEL
REGRESSION	3	334.9000	111.6333	1.0274E+11	100.000
RESIDUAL	6	6.5193E-09	1.0865E-09		
TOTAL	9	334.9000			

DATA DEF = DATE
ANALYSIS = MULT
DATA SET = 1

OBSERVATIONS:

11 READ

11 PROCESSED

0 REJECTED

SUM WEIGHTS =

11

TABLE OF RESIDUALS

UFS	ACTUAL Y	PREDICTED Y	RESIDUAL
1	10.00000	10.00000	-1.51340E-09
2	12.00000	12.00000	-3.23869E-09
3	13.00000	13.00000	-4.42178E-09
4	14.00000	14.00000	-4.88544E-09
5	15.00000	15.00000	-2.56114E-09
6	24.00000	24.00000	-3.25563E-09
7	26.00000	26.00000	-6.98492E-10
8	27.00000	27.00000	-5.02630E-09
10	16.00000	18.00000	3.72529E-09
11	21.00000	21.00000	-2.79397E-09

DURBIN-WATSON D STATISTIC = 1.86584

Interpretación de resultados de los ejemplos de la tabla 4.1 y 4.2 ; de los análisis estadísticos de regresión .

Conceptos del marco teórico utilizados .

1/ Promedio muestral = $\bar{X} = \sum_{i=1}^n x_i / n$

2/ Desviación standard = $S = \sqrt{\sum_{i=1}^n (y_i - \bar{Y})^2 / (n - 1)}$

3/ Varianza = $\hat{V}^2 = S^2$

4/ Coeficiente de correlación muestral = $r_{x,y} =$

$$= \left[\frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{n} \right] / \left[\frac{\sum_{i=1}^n (x_i - \bar{X})^2}{n} \right]^{1/2} \left[\frac{\sum_{i=1}^n (y_i - \bar{Y})^2}{n} \right]^{1/2}$$

i/ $r_{x,y} \rightarrow 1$ entonces $y = a + bx$; $b > 0$

ii/ $r_{x,y} \rightarrow -1$ entonces $y = a + bx$; $b < 0$

iii/ $r_{x,y} \rightarrow 0$ no indica que no están asociadas las variables , quiere decir que no están asociadas linealmente .

5/ Coeficiente de determinación = $r_{x,y}^2$

6/ Covarianza = $Cov(x,y) = \sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y}) / n$

6.1/ $Cov(x,y) > 0$ entonces si x aumenta y aumenta
si x disminuye y disminuye .

6.2/ $Cov(x,y) < 0$ entonces si x aumenta y disminuye
si x disminuye y aumenta .

6.3/ $Cov(x,y) = 0$ inconcluyente .

En estadística uno de los modelos lineales más utilizados es ; El Modelo De Regresión lineal múltiple ,el cual es de la forma siguiente :

$$y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \dots + b_r x_{ri} + e_i$$

Y que en forma matricial se representa por :

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1r} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{nr} \end{bmatrix} \begin{bmatrix} b_0 \\ \vdots \\ b_r \end{bmatrix} + \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}; \quad e_i \sim N(0, \sigma^2)$$

o también $Y = XB + E$; $E \sim N(0, \sigma^2 I)$

En donde x_{ij} representa el j -ésimo valor observado de las r variables , con el i -ésimo valor de Y , y b_j es el parámetro que especifica cómo Y está relacionado con los x_j . La variable Y es conocida como la variable dependiente y las variables x son conocidas como variables independientes , b_j es la derivada parcial $\partial Y / \partial x_j$ por lo que los b_j son conocidos como coeficientes de regresión parcial y indican el cambio en la variable dependiente Y asociado con una unidad de cambio en la correspondiente variable independiente (x_j) , mientras el resto de variables independientes permanecen constantes , b_0 es el valor de intercepción con el eje Y , cuando todos los x_j son cero .

Un caso particular es el modelo de regresión lineal simple que es de la forma:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} + \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}; \quad e_i \sim N(0, \sigma^2)$$

Considere el siguiente modelo el cual se resolvió utilizando el paquete :

$$\begin{bmatrix} 10 \\ 12 \\ 13 \\ \vdots \\ \vdots \\ 18 \\ 21 \end{bmatrix} = \begin{bmatrix} 1 & 11.6 \\ 1 & 13.87 \\ 1 & 13.972 \\ \vdots & \vdots \\ \vdots & \vdots \\ 1 & 18.985 \\ 1 & 22.675 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ \vdots \\ e_{10} \\ e_{11} \end{bmatrix}; \quad e_i \sim N(0, \sigma^2)$$

El modelo en forma matricial es de la forma siguiente:

$Y = XB + E$; $E \sim N(0, \sigma^2 I)$. El modelo de regresión lineal estimado es $\hat{Y} = X\hat{B}$, donde $Y - \hat{Y}$ es el vector de residuales , y la suma de cuadrados de los residuales es $\sum_{i=1}^n E_i^2 = (Y - \hat{Y})'(Y - \hat{Y})$, la cual cuando se minimiza con respecto a B produce el sistema de ecuaciones $(X'X)B = X'Y$, lo que es la forma equivalente de las ecuaciones normales en forma matricial . $X'X$ es simétrica y consiste del producto cruz de las variables x , $X'Y$ es el vector de los productos cruz xy . Al resolver el sistema se obtiene el estimador (\hat{B}) .

$$\hat{B} = (X'X)^{-1}X'Y = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

Ahora bién la información de que nos provee el paquete en la rutina de regresión lineal múltiple (MULTR) es :

$$\bar{X} = 19.28$$

Promedio de las x_i

$$S_x = 5.999$$

Desviación standard de las x_i

$$\bar{Y} = 17.81$$

Promedio de las y_i

$$S_y = 5.862$$

Desviación standard de las y_i

$$\text{Matriz normal } X'X = \begin{bmatrix} 11 & 212.13 \\ 212.13 & 4450.3 \end{bmatrix} \approx \begin{bmatrix} 1 & 19.28 \\ 0 & 359.8891 \end{bmatrix}$$

$$\text{Matriz de correlación } \begin{bmatrix} r_{xx} & r_{xy} \\ r_{yx} & r_{yy} \end{bmatrix} = \begin{bmatrix} 1 & .9984 \\ .9984 & 1 \end{bmatrix}$$

Opcionalmente se dan las inversas de la matriz normal y de la de correlación con OPTION INVERSE ; .

$$\hat{B} = (X'X)^{-1}X'Y = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} -1.0007 \\ 0.9755 \end{bmatrix}$$

$$\text{Error standard estimado} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{(n-2)}} = .34994$$

El modelo obtenido es: $y_i = -1.0007 + .9755x_i + e_i$

$$\text{Coeficiente de determinación} = \frac{\sum_{i=1}^n (y_i - \bar{Y})^2}{\sum_{i=1}^n (y_i - \bar{Y})^2 + \sum_{i=1}^n (y_i - \hat{y}_i)^2} = .9967$$

$$\text{Coeficiente de correlación múltiple} = R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{Y})^2} = .9984$$

La tabla de análisis de varianza es utilizada para la prueba de hipótesis, $H_0 : B = 0$ contra $H_a : B \neq 0$

TABLA DE ANALISIS DE VARIANZA

fuerza de variación	grados de libertad	suma de cuadrados	cuadrado medio	F
Regresión	1	$SSR = \frac{\left[\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y}) \right]^2}{\sum_{i=1}^n (x_i - \bar{X})^2}$ $= 342.5342$	$MSR = \frac{SSR}{1}$ $= 342.53$	$F = \frac{MSR}{MSE}$ $= 2797.14$
Residual	n-2 = 9	$SSE = SST - SSR$ $= 1.1021$	$MSE = \frac{SSE}{n-2}$ $= .1225$	
Total	n-1 = 10	$SST = \sum_{i=1}^n (y_i - \bar{Y})^2$ $= 343.6364$	— — —	

Regla.- Si $F > F_{1, n-2}^{1-\alpha}$ (de tablas), rechazar H_0 , es decir si F es muy grande rechazamos la hipótesis H_0 , esto quiere decir que la explicación al modelo de nuestras variables es muy buena.

$$F_{1,9}^{.01} = 5.12 < F \quad \text{significativa}$$

$$F_{1,9}^{.05} = 10.6 < F \quad \text{muy buena}$$

Lo que quiere decir es que la prueba es altamente significativa.

En la tabla de residuales se tiene, observación i , y_i observación real, $\hat{y}_i = b_0 + b_1 x_i$ observación estimada, $y_i - \hat{y}_i$ residuales.

Por último tenemos la estadística d Durbin-Watson que es la base para probar autocorrelación en análisis de regresión. La prueba se basa sobre la acepción de que los errores constituyen una serie recursiva de primer orden.

$$u_i = \rho u_{i-1} + e_i \quad \dots (a) \quad |\rho| < 1$$

$$e_i \sim N(0, \sigma^2) \quad ; \quad E \sim N(0, \sigma^2 I)$$

En otras palabras el residual para el periodo t es una función del residual del periodo previo ($t-1$) más un error aleatorio, la existencia de un modelo recursivo se puede detectar por un análisis de los residuales del modelo, al utilizar la estadística d Durbin Watson.

$$d = \frac{\sum_{i=2}^n (e_i - e_{i-1})^2}{\sum_{i=1}^n e_i^2} = 1.80735$$

Para probar la hipótesis nula $H_0: \rho = 0$ contra la alternativa $H_a: \rho > 0$; note que cuando $\rho = 0$ en (a) los u 's son no correlacionados, el parámetro se puede estimar por r .

$$\rho = r = \frac{\sum_{i=2}^n e_i e_{i-1}}{\sum_{i=1}^n e_i^2}$$

Donde existe una relación de aproximación entre d y r , $d \approx 2(1-r)$ d toma valores entre cero y cuatro, cuando $d \rightarrow 2$ es cuando $\rho = 0$ y $d \rightarrow 1$ cuando $\rho = 1$, al hacer tender el valor muestral d a 2 se afirma la evidencia de que no se tiene autocorrelación en el error, una evidencia de autocorrelación es indicada por la desviación de d de 2, la prueba formal para autocorrelación opera de la siguiente forma:

- i/ Si $d < d_L$; rechazar $H_0: \rho = 0$
- ii/ Si $d > d_U$; no rechazar $H_0: \rho = 0$
- iii/ Si $d_L < d < d_U$; no decidir

Los valores (d_L, d_U) se encuentran tabulados por Durbin y Watson en una tabla en el apendice del libro *Regressión -- Analysis By Example*, de Samprit Chatterjee y Bertram Price.

En modelos de 2 variables BASIS también provee de una grafica de los puntos (x,y). En modelos de regresión lineal - multiple la tabla de análisis de varianza es de la forma sig.:

TABLA DE ANALISIS DE VARIANZA

fuelle de variación	grados de libertad	suma de cuadrados	cuadrado medio	F
Regresión	m	$SSR = BX'Y = Y'X(X'X)^{-1}X'Y$	$MSR = \frac{SSR}{m}$	$\frac{MSR}{MSE}$
Residual	n-m-1	$SSE = SST - SSR$	$MSE = \frac{SSE}{n-m-1}$	
Total	n-1	$SST = \sum_{i=1}^n (y_i - \bar{Y})^2$		

En el segundo ejemplo los modelos ajustados fueron:

$$1/ \quad y_i = b_0 + b_1 x_i + e_i$$

obteniendo $\hat{y}_i = -.9929 + .9753x_i$

se grabaron en disco las variables x_i , y_i , predicciones y residuales utilizando OPTION ALLSAVE;, en formato binario en el archivo B0061, que fue asignado por BASIS, si se utiliza CODE=23; el archivo creado se llamaría BD231.

$$2/ \quad y_i = b_0 + b_1(T2)_i + b_2(T3)_i + b_3 x_i + e_i$$

donde $T2 = (y_i + x_i)$, $T3 = (y_i x_i)$

obteniendo

$$\hat{y}_i = -2.724(10)^8 + (T2)_i - 4.519(10)^{-11}(T3)_i - x_i$$

en ambos casos se consideraron valores faltantes iguales para la variable V1 o Y -99, y para la variable V2 o X 16.43.

El paquete BASIS tiene la capacidad de actualizar definiciones de datos, de procedimientos, y definiciones de - trabajo almacenadas en disco , que se hayan creado por medio del paquete . Estas actualizaciones pueden ser requeridas por el usuario para definir nuevas variables , modificar variables hacer nuevas definiciones en la definición de datos .

Para que el usuario lleve a cabo una actualización se requiere :

- 1/ Nombre con el que se almaceno la definición .
- 2/ Definiciones de impresión o borrado completo de la definición .
- 3/ Instrucciones de inserción o eliminación de lineas , de definiciones almacenadas .

Esta capacidad del paquete se utiliza unicamente en forma interactiva (por medio de una terminal remota) . Y las instrucciones que utiliza son las siguientes :

JOB UPDATE

Esta instrucción indica que se empezarán a hacer actualizaciones y esta terminara cuando se de la instrucción END; .

PRINT o DISPLAY

Print indica que la impresión de cambios se haga en una impresora . Display indica que las salidas se requieren en la terminal .

REMOVE

Se utiliza para borrar definiciones de datos o procedimientos.

Las modificaciones de definiciones se llevan a cabo unicamente por medio de una terminal , de la siguiente forma :

- 1/ FETCH <definición que se modificara> ; .
 - 1.1/ DATA <nombre>.- Se modificara una definición de datos.
 - 1.2/ PROCEDURE <nombre>.- se modificara un procedimiento.
 - 1.3/ JOB <nombre> .- se modificara un trabajo .

- 2/ DELETE <número> .- Se borrara la linea con el número -
indicado .
- 3/ INSERT <número> .- Se insertara la linea con el número
indicado .
- 4/ DISPLAY .- Se utiliza para listar la definición que se
modifica .
- 5/ RETURN .- Se utiliza si se requiere almacenar la definición
con los nuevos cambios .
- 6/ END; .- Esta instrucción indica que se termina la ---
actualización de una definición .

Para ejemplificar esta capacidad del paquete considere que se ejecuta el problema de la tabla 4.3 . El cual se modificara posteriormente . Las modificaciones que se haran se muestran en la tabla 4.4 . Y son las siguientes :

- 1/ Se llama el paquete .
- 2/ Se indica que no se requiere una introducción .
- 3/ Se utiliza el mismo número de USE , con el que se --
crearón las definiciones .
- 4/ Se indica que se hara una actualización .
 - 4.1/ Se lista el procedimiento que se va a modificar .
 - 4.2/ Se indica el procedimiento a modificar.
 - 4.2.1/ Se indica que no se requiere explicación , ni
que se listen las definiciones que se tienen
almacenadas en disco .
 - 4.2.2/ Se inserta en la linea 55 una ---
instrucción de filtración .
 - 4.2.3/ Se borra la linea 50 .
 - 4.2.4/ Se lista el procedimiento modificado.
 - 4.2.5/ Se inseta en la linea 75 un comentario.
 - 4.2.6/ Se indica que las modificaciones se -
harán permanentes .
- 5/ Se indica que termina la ctualización .
- 6/ Se ejecuta el procedimiento modificado , con un conjunto
de datos que se darán en la terminal .
- 7/ Se da por terminada la ejecución del paquete .

RUN *BASIS/BASIS; FILE LINE(REMOTE)	(R)
NO	(R)
88	(R)
NO	(R)
DATA DATOS LIBRARY;	(R)
FILE *;	(R)
1 "X";	(R)
2 "Y";	(R)
3 "Z";	(R)
TITLE " DATOS A GRAFICAR "	(R)
END;	(R)
PROCEDURE GRAF LIBRARY " MAY 30, 1981";	(R)
USE PLOT;	(R)
COMMENT SE PUEDEN GRAFICAR HASTA 9 FUNCIONES	(R)
INPUT GROUP(V1,V2),(V2,V3);	(R)
PLOTCHAR="&";	(R)
XMIN=0	(R)
TITLE " ///--- G R A F I C A S ---///"	(R)
PRINT ALL;	(R)
END;	(R)
EXECUTE PROCEDURE GRAF WITH DATA DATOS;	(R)
13,43,66	(R)
17,25,235	(R)
55,45,98	(R)
17,34,87	(R)
45,19,24	(R)
87,72,26	(R)
55,43,22	(R)
39,26,11	(R)
18,32,223	(R)
43,75,143	(R)
72,57,129	(R)
19,75,143	(R)
END;	(R)
QUIT;	(R)

06700:126 CALCE 30:140; YOU ARE SCHED#001(102)

DEFAULT PRINT DESTINATION=SITE

SESSION 9111 13:40:49 05/21/81

UN*BASIS/BASIS;FILE LINE(REMOTE)

RUNNING 9115

** 06700/B7700 BASIS *** 7.06 DATE 05/21/81 1:49 PM

INTRODUCTION NECESSARY:

?

NO

OK

USER ID=

BB

ID= BB UK

NO LIBRARY DIRECTORY FOUND ON DISK

***** IF YOU WISH TO HAVE A LINE-PRINTER LISTING

***** OF THE LATEST BASIS DOCUMENTATION, ENTER

***** THE COMMAND:

***** EXPLAIN ALL

READY!

NO

DATA DATOS (LIBRARY)

FILE *;

1 "X";

2 "Y";

3 "Z";

TITLE " DATOS A GRAFICAR "

END;

WAIT=

DEFINITION STORED IN USERS LIBRARY:

PROCEDURE GRAF LIBRARY " MAY 30, 1981";

USE PLOT;

COMMENT SE PUEDEN GRAFICAR HASTA 9 FUNCIONES

INPUT GROUP(V1,V2),(V2,V3);

PLOTCHAR="8";

XMIN=0;

TITLE "///--- G R A F I C A S ---///"

PRINT ALL;

END;

WAIT=

19115 (SF82)BASE08/DIRECTLY REMOVED ON UNAMP2 FK115 .

DEFINITION STORED IN USERS LIBRARY:

EXECUTE PROCEDURE GRAF WITH DATA DATOS;

WAIT=

EXECUTION BEGINS

ENTER FREE FORMAT DATA RECORDS:

17,25,235
 55,45,98
 17,34,87
 45,19,24
 87,72,26
 55,43,22
 39,26,11
 19,32,223
 43,75,143
 72,57,129
 19,75,143
 END)

\$\$\$
 \$\$\$\$
 \$\$
 \$\$\$
 \$\$\$
 \$\$\$
 \$\$\$
 \$\$\$

\$\$\$\$\$	\$\$\$\$\$	\$\$\$\$\$	\$\$\$\$\$	\$\$\$
\$\$\$\$\$	\$\$\$\$\$	\$\$\$\$\$	\$\$\$\$\$	\$\$\$\$\$
\$	\$	\$	\$	\$
\$	\$	\$	\$	\$
\$\$\$\$\$	\$\$\$\$\$	\$\$\$\$\$	\$\$\$	\$\$\$\$\$
\$\$\$\$\$	\$\$\$\$\$	\$\$\$\$\$	\$\$\$	\$\$\$\$\$
\$	\$	\$	\$	\$

#####

JOB DEF= *****

///--- G R A F I C A S ---///

PAGE 1

PROC DEF= GRAF

DATA DEF= DATOS

DATOS A GRAFICAR

BASIS 7.06

ANALYSIS= PLOT

05/21/81

DATA SET= 1

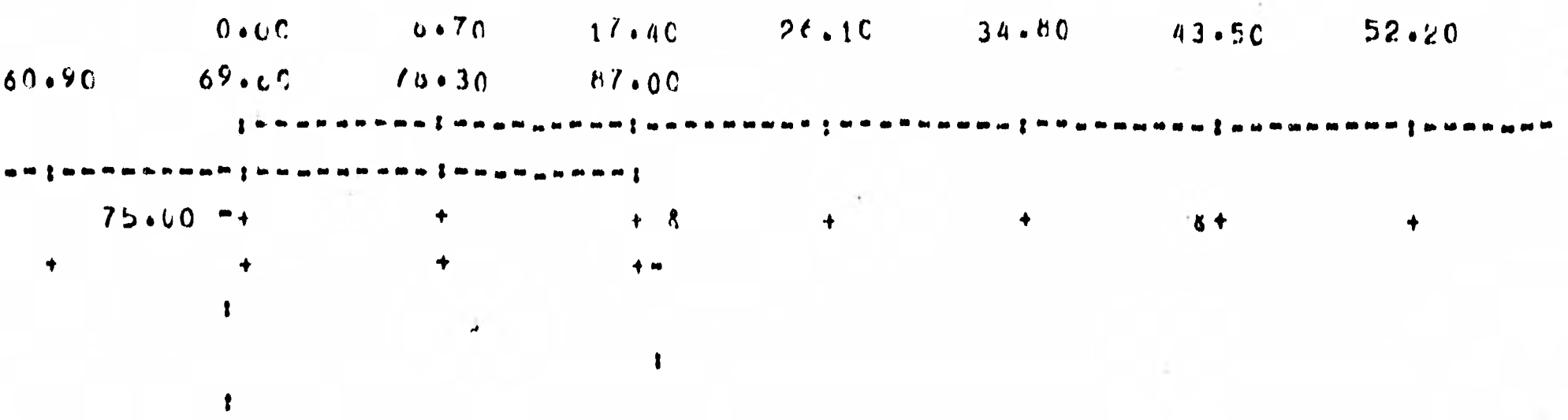
OBSERVATIONS: 12 READ, 12 PROCESSED, 0 REJE

CTED. SUM WEIGHTS=

12

& REPRESENTS (X)

Y)



195

RUN*BASIS/BASIS	(R)
NO	(R)
88	(R)
YES	(R)
JOB UPDATE	(R)
DISPLAY PROCEDURE GRAF	(R)
FETCH PROCEDURE GRAF	(R)
NO	(R)
NO	(R)
INSERT55	(R)
EXCLUDE V1 GT 66	(R)
DELETE 50	(R)
DISPLAY	(R)
INSERT 75	(R)
COMMENT MODIFICADO	(R)
RETURN	(R)
END;	(R)
EXECUTE PROCEDURE GRAF WITH DATA DATOS;	(R)
13,43,66	(R)
17,25,235	(R)
55,45,98	(R)
17,34,87	(R)
45,19,24	(R)
87,72,26	(R)
55,43,22	(R)
39,26,11	(R)
18,32,223	(R)
43,75,143	(R)
72,57,129	(R)
19,75,143	(R)
END;	(R)
QUIT;	(R)

Tabla 4.4

#B6700:126 CALDE 30:1403 YOU ARE SCHED#001(1C2)

* DEFAULT PRINT DESTINATION=SITE

#SESSION 9301 14:00:21 05/21/81

RUN*BASIS/BASIS

#RUNNING 9302

*** E6700/B7700 BASIS *** 7.06 DATE 05/21/81 2:08 PM

INTRODUCTION NECESSARY:

??

NO

OK

USER ID=

BB

ID= BB OK

IS DIRECTORY TO BE DISPLAYED:

YES

LIBRARY DIRECTORY FOLLOWS

STATUS: T=THIS SESSION ONLY, P=PERMANENT, D=DATED

PROCEDURE DEFINITIONS

GRAF 2 0

DATA DEFINITIONS

DATUS 2 P .

NO JOB DEFINITIONS

READY:

JOB UPDATE

DISPLAY PROCEDURE GRAP

```
1  PROCEDURE GRAP LIBRARY " MAY 30, 1981"
10  USE PLOT$
20  COMMENT SE PODEN GRAFICAR HASIA 9 FUNCIONES
30  INPUT GROUP(V1,V2),(V2,V3)
40  PLOTCHAR="8"
50  XMIN=0
60  TITLE "///--- G R A F I C A S ---///"
70  PRINT ALL$
80  END$
```

FETCH PROCEDURE GRAP

ARE COMMAND EXPLANATIONS REQUIRED:

NO

IS DEFINITION TO BE DISPLAYED:
NO
OK

ENTER MODIFICATION COMMANDS:

INSERT 55
ENTER STATEMENT NO 55:
EXCLUDE V1 GT 66
INSERTED
DELETE 50
DELETED
DISPLAY

```
1 PROCEDURE GRAP LIBRARY " MAY 30, 1981"  
10 USE PLOT;  
20 COMMENT SE PUEDEN GRAFICAR HASIA 9 FUNCIONES;  
30 INPUT CROLP(V1,V2),(V2,V3);  
40 PLOTCHAR="R";  
55 EXCLUDE V1 GT 66;  
60 TITLE "///--- G R A F I C A S ---///";  
70 PRINT ALL;  
80 END;
```

INSERT 75
ENTER STATEMENT NO 75:
COMMENT MODIFICADO
INSERTED

RETURN
NOW COMPILING
WAIT-

19302 (SFB2)BAS088/DIRECTY REMOVED ON UNAM2 PK115 .

MODIFIED DEFINITION STOPPED

ALL RECOMPILATION COMPLETED

#7302 (SF82)BAS088/DIRECTY REMOVED ON UNAM2 PK115 .
END)

READY:

EXECUTE PROCEDURE GRAP WITH DATA DATCS)
WAIT-

EXECUTION BEGINS

ENTER FREE FORMAT DATA RECORDS:

13,43,66

17,25,235

55,45,90

17,34,87

45,19,24

87,72,20

55,43,22

39,26,11

18,32,223

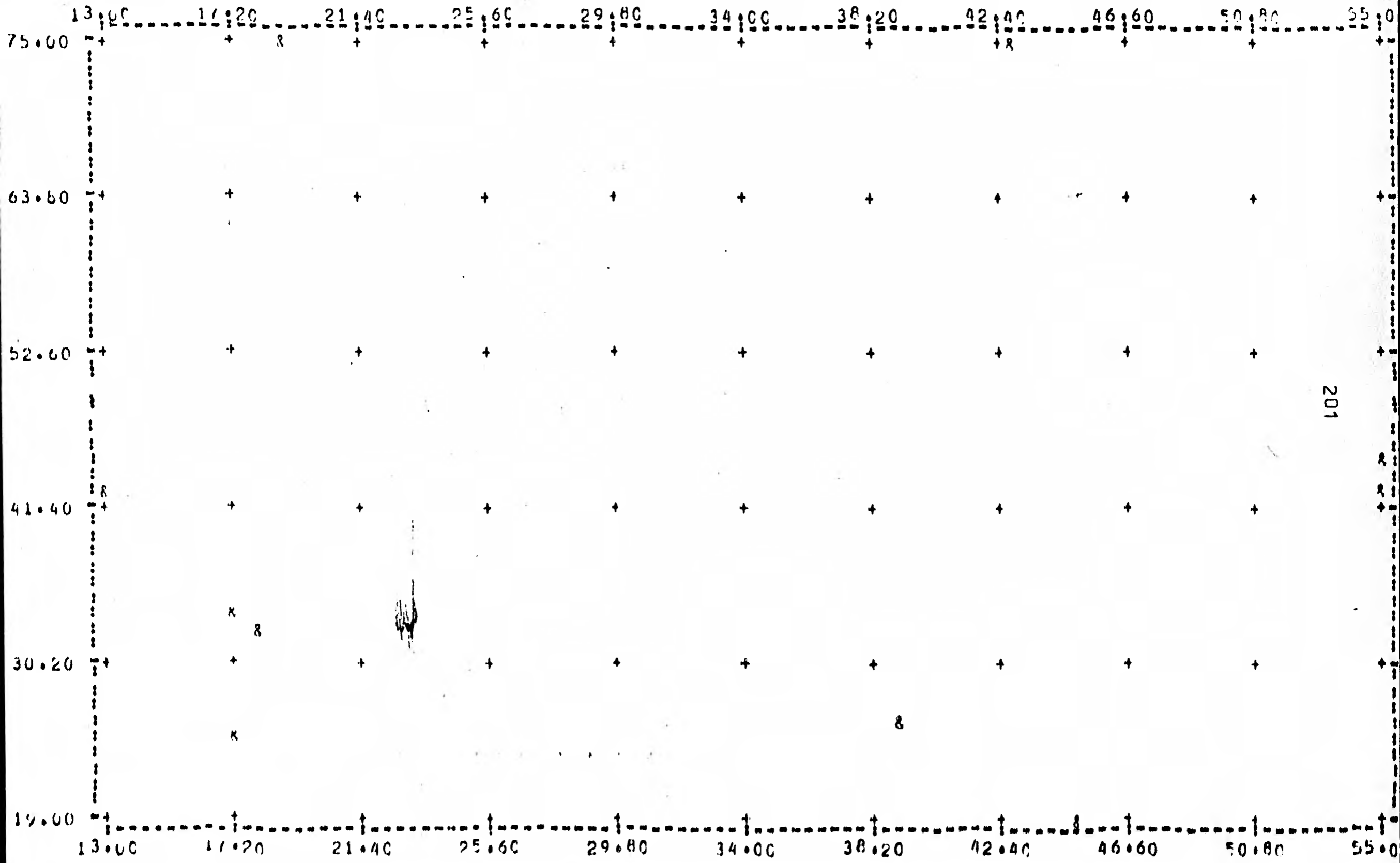
43,75,143

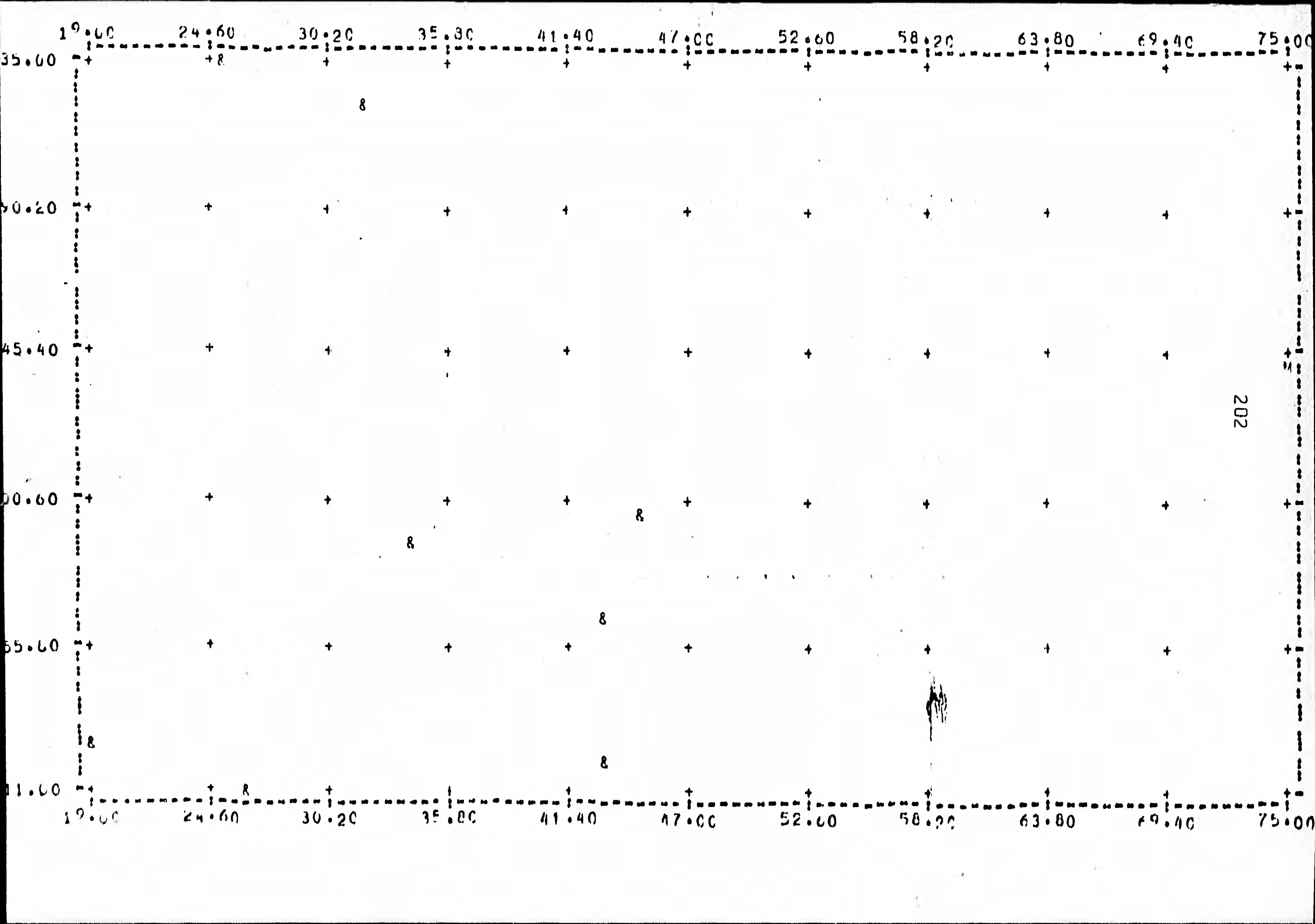
72,57,129

19,75,143

END)

EXECUTION COMPLETED





3.0

L I M I T A C I O N E S

- 1/ Se requiere un mínimo de diez observaciones para los -- análisis .
- 2/ Al llevar a cabo transformaciones en variables no admite muchas operaciones de cálculo para generar una nueva - variable .
- 3/ Al ajustar modelos de regresión lineal múltiple . Falla cuando se ajustan modelos en que se utilizan cinco ----- variables o múltiplos de cinco variables .
- 4/ Cuando se ajustan modelos de regresión ; en los que se - linealiza el modelo por medio de logaritmos . El paquete envía un error de lógica , en el que un arreglo se sale fuera de rango (Invalid Index) , en la mayoría de los casos .
- 5/ Limitaciones sobre cada uno de los análisis se pueden -- obtener por medio de la explicación de que provee el -- paquete para cada uno de los análisis .
- 6/ Requiere que el usuario sepa estadística .

C A P I T U L O V

M A N E J O D E L P A Q U E T E
S P S S
E N L A S O L U C I O N D E
P R O B L E M A S D E
E S T A D I S T I C A

En este capítulo se vera el manejo del paquete SPSS en la solución de problemas de estadística . ■■

Este paquete fue desarrollado por una corporación - que acoplo los conocimientos de tres tipos de especialistas :

- 1/ Investigadores practicos de las ciencias sociales .
- 2/ Investigadores de ciencias de la computación .
- 3/ Estadísticos .

En cada etapa del desarrollo del paquete se considero :

- 1/ Que los procedimientos estadísticos fuerán matemáticamente y estadísticamente correctos .
- 2/ Un diseño del programa eficiente .
- 3/ Que la logica y la sintaxis del paquete fuerán aprovechados en el análisis de datos de las ciencias sociales.
- 4/ Que se tuviera la facilidad de manipular los procedimientos y datos .

Este paquete empezo a desarrollarse en la Universidad de Stanford en 1965 . Ahora bién para el manejo de este paquete serán considerados unos ejemplos , así como las instrucciones más utilizadas del paquete . Estos ejemplos serán pequeños pero - daran la idea a seguir en el manejo del paquete , para la --- solución de problemas grandes y complejos .

Para un mayor detalle sobre el paquete se sugiere - ver la referencia ■■5■ .

■■ El nombre de SPSS es referido de las iniciales de ---
Statistical Package for the Social Sciences .

El paquete SPSS esta constituido de un conjunto de procedimientos de análisis estadísticos . Originalmente fue - diseñado para el análisis de datos en las ciencias sociales .

En la actualidad es utilizado por investigadores de diversas ramas , en las que se requiere de análisis estadísticos.

El paquete consta de los siguientes 18 análisis :

- 1/ AGGREGATE .- AGREGA SUBPOBLACIONES ESTADISTICAS LAS CUALES ALMACENA EN UN ARCHIVO EN DISCO .
- 2/ ANOVA .- ANALISIS DE VARIANZA .
- 3/ BREAKDOWN .- DESCRIBE SUBPOBLACIONES Y PRUEBA - LINEALIDAD .
- 4/ CANCORR .- ANALISIS DE CORRELACION CANONICA .
- 5/ CONDESCRIPTIVE .- CALCULO DE ESTADISTICAS BASICAS .
- 6/ CROSSTABS .- ANALISIS DE TABLAS CRUZADAS .
- 7/ DISCRIMINANT .- ANALISIS DISCRIMINANTE .
- 8/ FACTOR .- ANALISIS FACTORIAL .
- 9/ FREQUENCIES .- ANALISIS DE TABLAS DE FRECUENCIAS .
- 10/ GUTTMAN SCALE .- ANALISIS DE ESCALOGRAMAS .
- 11/ NONPAR CORR .- ANALISIS DE CORRELACION DE SPEARMAN O KENDALL .
- 12/ ONEWAY .- ANALISIS DE VARIANZA ONE-WAY .
- 13/ PARTIAL CORR .- CALCULO DE COEFICIENTES DE ----- CORRELACION PARCIAL .
- 14/ PEARSON CORR .- CALCULO DEL COEFICIENTE DE ----- CORRELACION DE PEARSON .
- 15/ REGRESSION .- ANALISIS DE REGRESION LINEAL ----- MULTIPLE .
- 16/ SCATTERGRAM .- IMPRESION DE GRAFICAS DE PAREJAS DE DATOS .
- 17/ T-TEST .- PRUEBA T-STUDENT .
- 18/ WRITE CASES .- ESTE PROCEDIMIENTO SE UTILIZA PARA ALMACENAR ALGUNAS O TODAS LAS --- VARIABLES EN UN ARCHIVO EN DISCO O EN CINTA , EN FORMATO BCD (EL -- FORMATO UTILIZADO POR FORTRAN) .

El procedimiento seguido para acceder las rutinas estadísticas con que cuenta SPSS ; utiliza registros de 80 -- caracteres en los cuales se considerarán dos campos .

1/ Campo de control .- El cual ocupa de la columna uno a la quince , y contiene una palabra o varias palabras de control .

2/ Campo de especificación .- El campo de especificación ocupa de la columna 16 a la 80 , y contiene los -- parametros y argumentos -- requeridos por la tarjeta de control .

El paquete lleva a cabo los análisis estadísticos - por medio de un programa de control (elaborado con instrucciones de SPSS) , que tiene la siguiente estructura :

```
?JOB <nombre>;USER=<clave>;CLASS=<clase>;BEGIN
?RUN=SERVICIO/SPSS6;DATA IOCR
```

```
{ <Instrucciones SPSS>
```

```
{ <Instrucciones de procedimiento SPSS> (1)
```

```
READ INPUT DATA
```

```
{ <Conjunto de datos que serán analizados> █ █
```

```
{ <Instrucciones de procedimiento SPSS> (2) █ █1 █
```

```
{ <Instrucciones de procedimiento SPSS> (n)
```

```
SAVE FILE █ █
```

```
FINISH
```

```
?END JOB
```

█ █ Esta instrucción es opcional y se utiliza solo en el caso de que el usuario requiera almacenar el prog. SPSS y los datos

█ █ En el caso de que el paquete se ejecute en una terminal remota , el conjunto de datos se almacena en un archivo en disco , y las instrucciones SPSS en otro archivo .

█ █ Los procedimientos 2-n son opcionales .

A continuación se indicara el procedimiento para - elaborar un programa de control SPSS , para acceder los análisis del paquete . En las cuales se considerara lo siguiente :

- 1/ Orden secuencial de las instrucciones ; en el que deben aparecer en un programa de control SPSS .
- 2/ Lo que aparezca subrayado en el campo de control se - considerara requerido ; el resto opcional .

A continuación se muestra un programa de control SPSS, en el cual el 1 en la parte superior indica el inicio del primer campo ; el 16 en la parte superior indica el inicio del segundo campo . Se describen solo <Instrucciones SPSS>.

1	16
RUN NAME	<nombre1>
<u>FILE NAME</u>	<nombre2> <etiqueta1>
<u>VARIABLE LIST</u>	<lista de variables1>
<u>INPUT MEDIUM</u>	<dispositivo>
SUBFILE LIST	<lista de subarchivos>
<u>INPUT FORMAT</u>	<formato>
<u>N OF CASES</u>	<número1>
MISSING VALUES	<lista de variables2>
VAR LABELS	<lista de variables3>
VALUE LABELS	<lista de variables4>
PRINT FORMATS	<lista de variables5>
RUN SUBFILES	<opción1>
TASK NAME	<título>
ASSIGN MISSING	<lista de variables6>
COMMENT	<comentario1>
DOCUMENT	<comentario2>

<Instrucciones de procedimiento SPSS>(1)

OPTIONS	<opción2>
<u>STATISTICS</u>	<número2>
<u>READ INPUT DATA</u>	

<Conjunto de datos que serán analizados>

SAVE FILE
FINISH

En donde :

1/ <nombre1>.- Indica el nombre que se le asigna a la -
corrida .

2/ <nombre2>.- Indica el nombre que se le asigna a el --
conjunto de datos .

2.1/ <etiqueta1>.- Puede ser un comentario
sobre el archivo de datos
o no aparecer .

3/ <lista de variables1>.- Indica la lista de variables
que serán leídas del conjunto
de datos .

3.1/ Puede indicarse una lista
de variables separadas por
comas . El nombre de cada
variable sera alfanumerico
empezando con un caracter
alfabetico.

3.2/ Puede utilizarse la palabra
TQ . Por ejemplo VA TQ VB
indica un conjunto de vars.
en la que la primera se --
llama VA y la ultima VB .

4/ <dispositivo>.- Indica el medio en que serán leídos los
datos.

4.1/ CARD .- Indica tarjetas perforadas.

4.2/ DISK .- Indica de disco .

4.3/ TAPE .- Indica de cinta .

5/ <lista de subarchivos>.- Indica cuantos subarchivos se
considerara que forman el --
conjunto de datos . En caso de
utilizarse , N OF CASES ya no
se utiliza .

6/ <formato>.- Indica el formato en que serán leídos los
datos .

6.1/ FREEFIELD .- Indica serán leídos en
formato libre .

6.2/ FIXED(formato BCD) .- Se indica el formato
formato en fortrán.

- 7/ <número1>.- Un valor entero indicando el número de -
casos que serán analizados .
- 8/ <lista de variables2>.- Se indica una lista de variables
seguida de un número encerrado -
entre parentecis , indicando que
cuando estas variables adquirán
este valor se consideren como --
valore faltantes .
- 9/ <lista de variables3>.- Se indica una lista de variables
con sus etiquetas , separadas por
una diagonal .
- 10/ <lista de variables4>.- Se indica una lista de variables
con sus valores etiquetados ,
separadas por una diagonal .
- 11/ <lista de variables5>.- Se indica una lista de variables
seguidas de un número encerrado
entre parentecis , indicando el
número de decimales requeridos ,
máximo cinco , separadas por una
diagonal .
- 12/ <opción1>.- Indica la ejecución de los subarchivos .
12.1/ ALL .- Indica que todos los subarchivos
se traten como uno solo .
12.2/ EACH .- Indica que cada subarchivo sea
tratado independientemente .
- 13/ <título>.- Indica que este título apareciera al principio
de cada corrida , hasta encontrar otro task
name.
- 14/ <lista de variables6>.- Lista de variables creadas por
medio de una transformación y
a las cuales se les asigna un
valor faltante .
- 15/ <comentario1>.- Indica un comentario ; en caso de que
se almacene el prog. SPSS este no se
almacenara .


```

1          16
CROSSTABS  VARIABLES = <lista de variables> (<valor
              mínimo> , <valor máximo> )
              <lista de variables> (<valor mínimo> , <valor
              máximo> )
              TABLES = <lista de variables> BY <lista de
              variables> BY <lista de variables> .../

```

```

1          16
CROSSTABS  TABLES = <lista de variables> BY <lista de
              variables> BY .../ <lista de variables>
              BY <lista de variables>

```

La primer forma requiere que las variables especificadas tengan unicamente valores enteros .

3/ Instrucciones de procedimiento SPSS .- Para un análisis de regresión -- lineal multiple .

```

1          16
REGRESSION VARIABLES = <lista de variables> /
              REGRESSION = <variable dependiente> ] (<parametros>
              ) ] WITH <lista de variables> (<nivel de inclusión>
              ) <lista de variables> (<nivel de inclusión> ) ...
              RESID = <0 o del modelo> ] / [REGRESSION = .../
              VARIABLES = .../ ]

```

A continuación se muestra un ejemplo , en el que se utiliza el paquete para resolver un problema de análisis de - regresión .

2.0 EJEMPLOS DE APLICACION

1. El procedimiento regression es utilizado para ajustar modelos de regresión lineal simple y multiple , consta de 15 opciones y 7 estadísticas .

OPCIONES :

- 1.- Incluir valores faltantes en el cálculo de coeficientes de correlación .
- 2.- Eliminación de datos faltantes en forma de parejas.
- 3.- Suprimir la impresión de las etiquetas de las variables .
- 4.- La matriz de correlación sera dada por el usuario.
- 5.- Medias y desviaciones estándar serán dadas por el usuario , precediendo a la matriz de correlación , (se usa cuando se use la opción 4)
- 6.- Suprimir la impresión step-by-step y solo el resumen de la tabla de regresión se imprime .
- 7.- Suprimir el resumen y solo el step-by-step se imprime.
- 8.- La matriz de correlación o matrices utilizadas en los calculos serán impresas en una unidad , la que es especificada en RAW OUTPUT UNIT por el usuario.
- 9.- La matriz de correlación es indexada por la lista de variables (no se puede usar sin la opción 4) .
- 10.- Ocasiona se secuencie la información de la columna 1 a la 20 de cada registro sobre el raw-output-data file .
- 11.- y 12.- Los residuales estandarizados se grabarán en una unidad especificada por el usuario sobre el raw-output-data file .
- 13.- Predicciones estandarizadas , las cuales són un producto pesado de los datos existentes .
- 14.- Suprimir la impresión de ejes sobre las graficas de predicciones estandarizadas contra residuales estandarizados .
- 15.- Imprimir medias y desviaciones estándar en el raw-output-data file especificado por el usuario.

ESTADISTICAS :

- 1.- Imprimir la matriz de correlación .
- 2.- Media , desviación estándar y número de casos --
validos.
- 3.- Imprime la matriz de correlación y una nota sobre -
sus elementos .
- 4.- Grafica residuales estandarizados contra los casos
reales ,del archivo de datos .
- 5.- Estadística Durbin-Watson .
- 6.- Grafica de residuales estandarizados contra \hat{Y} ---
estandarizada .
- 7.- Imprime la matriz de correlación y el número de casos.

El problema de regresión que se resolviera con el paquete SPSS es el primer ejemplo que se resolvió con el paquete BASIS , en el capítulo anterior; por lo que se grabó un programa SPSS en disco con el nombre de REGRESION y un archivo de datos con el nombre DATOSR .

El archivo REGRESION contiene la siguiente información :

1	16
RUN NAME	REGRESION LINEAL MULTIPLE
FILE NAME	INFORMACION , AJUSTE DE UN MODELO LINEAL
COMMENT	
COMMENT	A CONTINUACION LA LISIA DE VARIABLES
COMMENT	
VARIABLE LIST	V1,V2
COMMENT	
COMMENT	SE ACCESA POR TARJETAS ES CARD
COMMENT	EN DISCO DISK , EN CINTA TAPE
COMMENT	
COMMENT	
INPUT MEDIUM	DISK
COMMENT	
COMMENT	LOS DATOS ESTARAN EN FORMATO LIBRE
COMMENT	
INPUT FORMAT	FREEFIELD
COMMENT	
COMMENT	EL NUMERU DE CASOS ES ONCE

```

1          16
COMMENT
N OF CASES      11
COMMENT
COMMENT          ETIQUETAMOS LAS VARIABLES V1 COMO Y,V2 COMO X
VAR LABELS      V1 Y/ V2 X/
COMMENT
COMMENT          UTILIZAMOS EL PROCEDIMIENTO DE REGRESION
COMMENT
REGRESSION      VARIABLES = V1,V2/
                REGRESSION = V1 WITH V2(1)/

STATISTICS      ALL
READ INPUT DATA
FINISH

```

La forma de ejecutar este programa es la siguiente :

```

RUN*SERVICIO/SPSS6;FILE FILE6(REMOTE),%          (R)
FILE5(KIND=DISK,TITLE=REGRESION),%              (R)
FILE8(KIND=DISK,TITLE=DATOSR)                    (R)

```

En las paginas siguientes se muestran los resultados obtenidos al ejecutar el PAQUETE SPSS en la computadora .

El procedimiento seguido en la solución de este -- problema fue el siguiente :

1/ Se creo un programa SPSS .- En este programa se llamo el análisis de regresión lineal multiple ; y fue creado en una terminal remota con el nombre de REGRESION.

1.1/ Se le asigno el nombre de REGRESION LINEAL MULTIPLE a la corrida .

1.2/ Se le asigno el nombre INFORMACION al conjunto de datos ; etiquetandolos como AJUSTE DE UN MODELO LINEAL .

- 1.3/ Se declaro una lista de dos variables ; V1 y V2 .
- 1.4/ Se declaro que los datos que se procesarían con el paquete residian en disco .
- 1.5/ Se indico que los datos se leerían en formato -- libre , es decir separados por comas .
- 1.6/ Se declaro que se tenian once observaciones ; para ajustar el modelo lineal .
- 1.7/ Se etiquetarón las variables .- A la variable V1 se le etiqueto como Y ; a la variable V2 se le - etiqueto como X.
- 1.8/ Se llamo el procedimiento de regresión lineal - multiple .
 - 1.8.1/ Se declaro que las variables V1 y V2 se utilizarían en el modelo .
 - 1.8.2/ Se indico que en el modelo de regresión que se ajustaria , la variable dependiente seria V1 .
- 1.9/ No se utilizarón numeros de opción .
- 1.10/ Se solicitarón todas las estadísticas que emite el procedimiento REGRESSION .
- 1.11/ Se indico la lectura de las observaciones .
- 1.12/ Se indica el final del programa SPSS .
- 2/ Se procedio a crear el archivo de datos .- A este -- archivo se le llamo DATOSR .
- 3/ Se procedio a ejecutar el paquete .
 - 3.1/ Se pidieron los resultados en la terminal .
 - 3.2/ Se indico que el programa SPSS se encontraba en disco , grabado con el nombre de REGRESION .
 - 3.3/ Se indico que los datos que accesaría el programa SPSS se encontraban en disco , grabados con el - nombre de DATOSR . ■■

Los resultados obtenidos se pueden verificar en la interpretación de resultados del capítulo anterior , del ejemplo uno .

■■ La forma en que utiliza los archivos se vera más adelante.

DISTRIBUTED FOR THE BURROUGHS B6700 BY THE
SOCIAL SCIENCE DATA SERVICE
UNIVERSITY OF CALIFORNIA, DAVIS
DEFAULT WORKSPACE FOR THIS RUN: 20000 WORDS

RUN NAME
FILE NAME
COMMENT
COMMENT
VARIABLE LIST
COMMENT
COMMENT
INPUT MEDIUM
COMMENT
COMMENT
INPUT FORMAT
COMMENT
COMMENT
N OF CASES
COMMENT
VAR LABELS
COMMENT
COMMENT
REGRESSION
STATISTICS

REGRESION LINEAL MULTIPLE
INFORMACION A AJUSTARLE UN MODELO LINEAL
A CONTINUACION LA LISTA DE VARIABLES
V1, V2
SE ACCESA POR TARJETAS ES CARD
EN DISCO DISK, EN CINTA TAPE
DISK
LOS DATOS ESTARAN EN FORMATO LIBRE
FREEFIELD
EL NUMERO DE CASOS ES 11
ETIQUETAMOS LAS VARIABLES V1 COMO Y, V2 COMO X
V1 Y/V2 X/
UTILIZAMOS EL PROCEDIMIENTO DE REGRESION
VARIABLES = V1, V2/
REGRESION = V1 WITH V2(1)/
ALL

***** REGRESSION PROBLEM REQUIRES 34 WORDS OF WORKSPACE, NOT INCLUDING RESIDUALS *****

HEAD INPUT DATA
END OF DATA INPUT, READ COUNT = 11 DATA ERROR COUNT = 0.

217

FILE INFERMAC (CREATION DATE = 06/02/81)

A AJUSTARLE UN MODELO LINEAL

VARIABLE	MEAN	STANDARD DEVIATION	CASES
V1	17.8182	5.8621	11
V2	19.2897	5.9991	11

REGRESSION LINEAL MULTIPLE

06/02/81

PAGE 3

FILE INFCRMAC (CREATION DATE = 06/02/81) A AJUSTARLE UN MODELO LINEAL

CORRELATION COEFFICIENTS

VALUE OF 99.00000 IS PRINTED
IF A COEFFICIENT CANNOT BE COMPUTED.

	V1	V2
1	1.00000	0.99840
2	0.99840	1.00000

REGRESSION LINEAL MULTIPLE

FILE INFCMAC (CREATION DATE = 06/02/81) A AJUSTARLE LN MODELU LIN

***** MULTIPLE REG
DEPENDENT VARIABLE: V1 Y

VARIABLE(S) ENTERED ON STEP NUMBER 1. V2 X

MULTIPLE R 0.99840 ANALYSIS OF VARIANCE
R SQUARE 0.99679 REGRESSION
ADJUSTED R SQUARE 0.99644 RESIDUAL
STANDARD ERROR 0.34994

----- VARIABLES IN THE EQUATION -----

VARIABLE	B	BETA	STL ERROR B	F
(CONSTANT)	-1.00070	0.99840	0.01845	2797.148

MAXIMUM STEP REACHED

06/02/81

PAGE 4

REGRESSION LIST

EAL
REGRESSION *****

DF	SUM OF SQUARES	MEAN SQUARE	F
1	342.53424	342.53424	2797.14779
9	1.10213	0.12246	

----- VARIABLES NOT IN THE EQUATION -----

VARIABLE BETA IN PARTIAL TOLERANCE F

FILE INFCRMAC (CREATION DATE = 06/02/81) A AJUSTABLE LN MODELU LINEAL

***** MULTIPLE REGRESSION *****

DEPENDENT VARIABLE: V1 Y

SUMMARY TABLE

VARIABLE	MULTIPLE R	R SQUARE	RSC CHANGE	SIMPLE R	B
(CONSTANT) X	0.99840	0.99679	0.99679	0.99840	0.97 -1.00

DATA TRANSFORMATION DONE UP TO THIS POINT.

NU OF TRANSFORMATIONS 0
 NU OF RECODE VALUES 0
 NU OF ARITHM. OR LOG. OPERATIONS 0

221

FINISH

2. Como un segundo ejemplo considere la obtención de estadísticas básicas por medio del paquete .

El procedimiento condescriptive es utilizado para obtener estadísticas básicas , consta de 4 opciones y 9 estadísticas .

OPCIONES :

- 1.- Incluir todos los casos a excepción de los declarados faltantes .
- 2.- Suprimir impresión de las etiquetas de las variables.
- 3.- Imprimir valores estandarizados (debe indicarse la unidad en RAW OUTPUT UNIT , que debe ser de 15 a 20 , y se asigna título , maxrecsize , kind , etc. por omisión es cinta, tape y maxrecsize 22) .
- 4.- Imprime un diccionario en forma alfabética de las variables utilizadas .

ESTADÍSTICAS :

- | | |
|--------------------------|---------------------|
| 1.- Media . | 8.- Simetría . |
| 2.- Error estándar . | 9.- Rango . |
| 5.- Desviación estándar. | 10.- Valor mínimo . |
| 6.- Varianza . | 11.- Valor máximo . |
| 7.- Curtosis . | |

En este caso el problema se ejecutara de la siguiente forma :

```

RUN*SERVICIO/SPSS6;FILE FILE6(REMOTE);%      (R)
FILE FILE5(DISK,TITLE=PROGSPSS);%           (R)
FILE FILE8(DISK,TITLE=DATOSSPSS);%         (R)
FILE FILE4(DISK,TITLE=ESTADISTICA)         (R)

```

Se considero que el programa SPSS se encontraba en disco con el nombre PROGSPSS , y los datos también se tenían en disco con el nombre DATOSSPSS , y se utilizo la instrucción SAVE FILE para almacenar el programa spss y los datos en disco con el nombre ESTADISTICA para posteriormente accederlo sin tener que generarlo de nuevo .

```
1          16
RUN NAME    EJEMPLO DE ESTADISTICAS BASICAS
FILE NAME   ESTADISTICA,ARCHIVO CON LA INFORMACION
COMMENT
COMMENT     ESTE ARCHIVO SERA ALMACENADO EN UN ARCHIVO
COMMENT     QUE SE LLAMA ESTADISTICA PARA DESPUES ACCESARLO
COMMENT
COMMENT     CON OTRO PROCEDIMIENTO DE SPSS
COMMENT
VARIABLE LIST  V1 TO V4,V8,V9
INPUT MEDIUM  DISK
COMMENT
COMMENT     SE UTILIZARAN TRES SUBARCHIVOS EN ESTA CORRIDA
COMMENT
SUBFILE LIST  SUBA(15),SUBB(15),SUBC(13)
COMMENT     EL PRIMER SUBARCHIVO CONSTA DE 15 CASOS
COMMENT
COMMENT     EL SEGUNDO DE 15 CASOS Y EL TERCERO DE 13 CASOS
COMMENT
INPUT MEDIUM  FREEFIELD
COMMENT
VAR LABELS   V1 ZONA ASIGNADA/V2 NIVEL/V3 PESO ASIGNADO/
              V4 CANTIDAD DE COMPRA/V8 VOLUMEN DE PRODUCCION/
              V9 CREDITO/
COMMENT
COMMENT     SE ETIQUETAN VALORES EN LAS VARIABLES
COMMENT
VALUE LABELS V1(0) NO TUVO (1) PENDIENTE/
              V2 (2) PERMANECIO IGUAL/
              V3 (0) NORMAL (3) EN EXCESO/
COMMENT
COMMENT     LAS VARIABLES V8 Y V9 SE IMPRIMIRAN CON 4 DECIMALES
COMMENT
```

```

1          16
PRINT FORMATS  VB,V9(4)
RUN SUBFILES   ALL
TASK NAME      ESTADISTICAS BASICAS
COMMENT
COMMENT        SE UTILIZARA EL PROCEDIMIENTO CONDESCRIPTIVE
COMMENT
CONDESCRIPTIVE V2 TO V8
OPTIONS        1,4
STATISTICS     ALL
READ INPUT DATA
SAVE FILE
FINISH

```

En las paginas siguientes se muestra la ejecución .

El procedimiento seguido en la solución de este -- problema el siguiente :

1/ Se creo un programa SPSS .- En este programa se llamo el procedimiento de cálculo de estadísticas básicas ; y fue creado en una terminal remota con el nombre de PROGSPSS .

1.1/ Se le asigno el nombre de EJEMPLO DE ESTADISTICAS BASICAS a la corrida .

1.2/ Se le asigno el nombre ESTADISTICA al conjunto de datos ; etiquetandolos como ARCHIVO CON LA ---- INFORMACION .

1.3/ Se declaro una lista de variables ; en la que la primer variable se llamaria V1 y las ultimas tres se llamarían V4 , V8 y V9 .

1.4/ Se declaro que los datos que se procesarían con el paquete residian en disco .

1.5/ Se indico que el archivo de datos tendria una estructura de subarchivos . Considerando tres subarchivos el primero y el segundo con 15 casos , y el tercero con trece casos .

- 1.6/ Se indico que los datos se leerían en formato libre.
 - 1.7/ Se etiquetarón las variables .
 - 1.8/ Se etiquetarón valores de variables .
 - 1.9/ Se indico que las variables V8 y V9 se imprimieran con cuatro decimales .
 - 1.10/ Se indica que los subarchivos se traten como uno solo , en el cálculo de estadísticas básicas .
 - 1.11/ Se indico que se imprimiera en la corrida el --
letrero ESTADISTICAS BASICAS .
 - 1.12/ Se llamo el procedimiento que calcula estadísticas básicas .
 - 1.13/ Se utilizarón las opciones 1 y 4 de este procedimiento estadístico .
 - 1.14/ Se solicitarón todas las estadísticas que emite el procedimiento CONDESCRIPTIVE .
 - 1.15/ Se indico la lectura de las observaciones .
 - 1.16/ Se indica que se almacene almacene el programa SPSS y los datos en un archivo en disco .
 - 1.17/ Se indica el final del programa SPSS .
- 2/ Se procedio a crear el archivo de datos .- A este archivo se le llamo DATOSSPSS.
- 3/ Se procedio a ejecutar el paquete .
- 3.1/ Se pidieron los resultados en la terminal .
 - 3.2/ Se indico que el programa SPSS se encontraba en disco , grabado con el nombre de PROGSPSS.
 - 3.3/ Se indico que los datos que accesaría el programa SPSS residián en disco , con el nombre de DATOSSPSS.
 - 3.4/ Se indico que se generaría un nuevo archivo ; el cual contendría el programa y los datos ; este -- nuevo archivo se generaría con el nombre de --- ESTADISTICA , para accesarlo posteriormente con el paquete .

DISTRIBUTED FOR THE BURROUGHS B6700 BY THE
SOCIAL SCIENCE DATA SERVICE
UNIVERSITY OF CALIFORNIA, DAVIS
DEFAULT WORKSPACE FOR THIS RUN •• 20000 WORDS

UN NAME EJEMPLO DE ESTADISTICAS BASICAS
FILE NAME ESTADISTICA ARCHIVO CON LA INFORMACION
COMMENT ESTE ARCHIVO SERA ALMACENADO EN UN ARCHIVO
COMMENT QUE SE LLAMA ESTADISTICA PARA DESPUES ACCESARLO
COMMENT CON OTRO PROCEDIMIENTO DE S P S S

VARIABLE LIST V1 TO V4, V8, V9
INPUT RECORD DISK
COMMENT SE UTILIZARAN TRES SUB/ARCHIVOS EN ESTA CORRIDA
COMMENT SUBA(15), SUBB(15), SUBC(13)
COMMENT EL PRIMER SUBARCHIVO CONSTA DE 15 CASOS
COMMENT EL SEGUNDO DE 15 CASOS Y EL TERCERO DE 13 CASOS
INPUT FORMAT FREEFIELD

VAR LABELS V1 ZONA ASIGNADA / V2 NIVEL / V3 PESO ASIGNADO /
V4 CANTIDAD DE COMPRA / V8 VOLUMEN DE PRODUCCION /
V9 CREDITO /

COMMENT SE ETIQUETAN VALORES EN LAS VARIABLES

VALUE LABELS V1 (0) NO TUVO (1) PENDIENTE /
V2 (2) PERMANECIO IGUAL PESO /
V3 (3) NORMAL (3) EN EL PESO /

COMMENT LAS VARIABLES V8 Y V9 SE IMPRIMIRAN CON 4 DECIMALES

UNIT FORMATS V8, V9(4)
UN SUBFILES ALL
JOB NAME ESTADISTICAS BASICAS
COMMENT SE UTILIZARA EL PROCEDIMIENTO CON DESCRIPTIVE
UNDESCRIPTIVE V2 TO V8

226

EJEMPLO DE ESTADISTICAS BASICAS
ESTADISTICAS BASICAS
OPTIONS 1,4
STATISTICS ALL
READ INPUT DATA
END OF DATA INPUT HEAD COUNT = 13 DATA ERROR COUNT = 0.

VARIABLE	V2	NIVEL	STD ERROR	1.088	STD DEV	7.134
MEAN		7.349	KURTOSIS	10.202	SKEWNESS	3.236
VARIANCE		50.899	MINIMUM	2.000	MAXIMUM	36.000
RANGE		34.000				
VALID OBSERVATIONS	=	43	MISSING OBSERVATIONS	=	0	

VARIABLE	V3	PESU ASIGNADO	STD ERROR	4.749	STD DEV	31.140
MEAN		-3.465	KURTOSIS	5.864	SKEWNESS	-2.806
VARIANCE		969.683	MINIMUM	-99.000	MAXIMUM	19.000
RANGE		118.000				
VALID OBSERVATIONS	=	43	MISSING OBSERVATIONS	=	0	

VARIABLE	V4	CANTIDAD DE COMPRA	STD ERROR	0.470	STD DEV	3.083
MEAN		7.140	KURTOSIS	3.243	SKEWNESS	1.815
VARIANCE		9.504	MINIMUM	2.000	MAXIMUM	18.000
RANGE		16.000				
VALID OBSERVATIONS	=	43	MISSING OBSERVATIONS	=	0	

VARIABLE	V6	VOLUMEN DE PRODUCCION	STD ERROR	0.494	STD DEV	3.237
MEAN		7.372	KURTOSIS	6.569	SKEWNESS	2.035
VARIANCE		10.477	MINIMUM	3.000	MAXIMUM	21.000
RANGE		18.000				
VALID OBSERVATIONS	=	43	MISSING OBSERVATIONS	=	0	

227

VARIABLE INDEX

VARIABLE	PAGE	VARIABLE	PAGE	VARIABLE	PAGE
V2	0	V3	0	V4	0
		V8			

ALPHABETIC INDEX

VARIABLE	PAGE	VARIABLE	PAGE	VARIABLE	PAGE
V2	0	V3	0	V4	0
		V8			

DATA TRANSFORMATION DONE UP TO THIS POINT..

NO OF TRANSFORMATIONS	0
NO OF RECODE VALUES	0
NO OF ARITHM. OR LOG. OPERATIONS	0

SAVE FILE

FILE ESTALIST HAS BEEN SAVED WITH 9 VARIABLES..

LNUM	SUBFILE	CASHGT	V1	V2	V3	V4	V8	V9
------	---------	--------	----	----	----	----	----	----

THE SUBFILES ARE..

NAME	NO OF CASES
SUBA	15
SUBB	13
SUBC	

3. El procedimiento crosstabs es utilizado para obtener tablas de contingencia , por lo que este procedimiento nos da una tabulación cruzada la cual es una distribución de frecuencia conjunta de acuerdo a dos o más variables , operando bajo dos modos :

i/ Entero utilizando solo variables enteras .

ii/ General utilizando variables numéricas y alfanúmericas. consta de 9 opciones y 10 estadísticas .

OPCIONES :

- 1.- Incluir valores perdidos en las tablas y cálculo de estadísticas .
- 2.- Suprimir la impresión de etiquetas de las variables.
- 3.- El porcentaje por renglón se suprime .
- 4.- El porcentaje por columna se suprime .
- 5.- El porcentaje total se suprime .
- 6.- Omite etiquetas de valores pero imprime las etiquetas de las variables .
- 7.- Los valores perdidos se incluyen en las tablas pero no en el cálculo de las estadísticas .
- 8.- Los valores de la variable renglón se imprimen en orden inverso , la variable columna no es afectada por esta opción y las estadísticas no se alteran.
- 9.- Imprime un índice de todas las tablas producidas por el procedimiento crosstabs y también el número de pagina .

ESTADÍSTICAS :

- 1.- Chi-cuadrado .
- 2.- Phi para tablas de 2x2 y V de Cramer para tablas mayores .
- 3.- Coeficiente de contingencia .
- 4.- Lambda simétrica y asimétrica .
- 5.- Coeficiente de incertidumbre simétrico y asimétrico.

- 6.- "Tau" B de Kendall .
- 7.- "Tau" C de Kendall .
- 8.- "Gama" parcial y de orden céro para tablas de 3-n variables .
- 9.- "D" de Somers simetrica y asimetrica .
- 10.- Eta .

El procedimiento crosstabs procesa datos también de caracter nominal proporcionando un medio para comprobar hipótesis de correlación entre 2 ó más variables .

En el ejemplo para el procedimiento crosstabs se -- utilizará el programa del ejemplo anterior que se almaceno en disco cón el nombre ESTADISTICA , y se pedirán las tablas --- cruzadas de la variable V4 cón V8 , por lo que creamos un programa en disco con solo el procedimiento crosstabs y lo llamamos TABLAS , y consta de las siguientes instrucciones .

1	16
RUN NAME	TABLAS CRUZADAS
GET FILE	ESTADISTICA
CROSSTABS	TABLES = V4 BY V8
OPTIONS	1,2,3
STATISTICS	1,7
FINISH	

Y la forma de ejecutar el problema es la siguiente :

RUN*SERVICIO/SPSS6;FILE FILE6(REMOTE),%	(R)
FILE5(KIND=DISK,TITLE=TABLAS),%	(R)
FILE3(KIND=DISK,TITLE=ESTADISTICA)	(R)

a continuación en las paginas siguientes se muestra la ejecución de este problema .

El proceso seguido en la solución de este problema fue el siguiente :

- 1/ Se creo un programa SPSS.- En este programa se llamo el análisis de tablas cruzadas ; y fue creado en una terminal remota con el nombre de TABLAS .
 - 1.1/ Se le asigno el nombre de TABLAS CRUZADAS a la corrida .
 - 1.2/ Se llamo un programa SPSS y datos que se tenian . Esto es , que se crearón en una corrida anterior por medio de un programa SPSS ; el cual se llama ESTADISTICA .
 - 1.3/ Se llamo el procedimiento de tablas cruzadas .
 - 1.3.1/ Se pidieron las tablas cruzadas de las - variables V4 contra la variable V8 .
 - 1.4/ Se pidieron las opciones 1,2 y 3 .
 - 1.5/ Se pidieron las estadísticas 1 y 7 .
 - 1.6/ Se indico el final del programa .
- 2/ Se procedio a ejecutar el paquete .
 - 2.1/ Se pidieron los resultados en la terminal .
 - 2.2/ Se indico que el programa SPSS se encontraba en disco , grabado con el nombre TABLAS .
 - 2.3/ Se indica que el programa SPSS definido con anterioridad y que contiene los datos tambien se llama ESTADISTICA.

DISTRIBUTED FOR THE BURROUGHS B6700 BY THE
SOCIAL SCIENCE DATA SERVICE
UNIVERSITY OF CALIFORNIA, DAVIS
DEFAULT WORKSPACE FOR THIS RUN.. 20000 WORDS

RUN NAME
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
GET FILE

EJEMPLO DE TABLAS CRUZADAS
SE UTILIZARA EL ARCHIVO DE DATOS GENERADO EN
LA SECCION ANTERIOR CON EL PROCEDIMIENTO DE
ESTADISTICAS BASICAS
ESTADISTICA

FILE ESTATIST HAS 9 VARIABLES
THE SUBFILES ARE..

NAME	NO OF CASES
SUBA	15
SUBB	15
SUBC	13

COMMENT
COMMENT
COMMENT

AHORA LLAMAMOS EL PROCEDIMIENTO DE CROSSTABS

CROSSTABS
OPTIONS
STATISTICS

TABLES = V4 BY V8
1,2,3
1,7

232

***** GIVEN WORKSPACE ALLOWS FOR 3996 CELLS AND 2 DIMENSIONS FOR CROSSTAB PROBLEM *****

END OF DATA INPUT, READ COUNT = 13 DATA ERROR COUNT = 0.

ESTADIST (CREATION DATE = 06/02/81) ARCHIVO CON LA INFORMACION

***** C R O S S T A B U L A T I O N O F *****
***** BY V8 ***** PAGE 1 OF 2

COUNT	V8												ROW TOTAL	
	COL	PCT	1	2	3	4	5	6	7	8	9	11		16
1			3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000	11.0000	16.0000	21.0000		
2	25	2.3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	0	0.0	0.0000	0.0000	16.7	0.0000	0.0000	0.0000	9.1	0.0000	0.0000	0.0000	0.0000	0.0000
4	25	2.3	0.0000	0.0000	16.7	42.9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	50	4.7	0.0000	0.0000	16.7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0	0.0	0.0000	0.0000	33.4	14.3	16.7	0.0000	20.3	0.0000	0.0000	0.0000	0.0000	0.0000
7	0	0.0	10.0	0.0000	0.0000	28.6	16.7	60.3	20.3	0.0000	0.0000	0.0000	0.0000	0.0000
8	0	0.0	0.0000	0.0000	0.0000	14.3	66.7	20.3	16.3	0.0000	0.0000	0.0000	0.0000	0.0000
9	0	0.0	0.0000	0.0000	16.7	0.0000	0.0000	20.3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
COLUMN TOTAL	9.3	2.3	14.0	16.3	14.0	11.6	25.6	2.3	2.3	2.3	2.3	2.3	2.3	2.3

233

NUED)

STATIST (CREATION DATE = 06/02/81) ARCHIVO CON LA INFORMACION
SUBA SUBB SUBC

***** C R O S S T A B U L A T I V E I N Q U E R Y *****
***** PAGE 2 OF 2 *****

COUNT	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000	11.0000	16.0000	21.0000	NUM TOTAL
10.	0	0	0	0	0	0	1	1	0	0	4
10.	0.0	0.0	0.0	0.0	0.0	0.0	2.3	2.3	0.0	0.0	4
10.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	4
10.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.3	2.3	4
COLUMN TOTAL	4	1	6	7	6	5	11	1	1	1	100

CHI SQUARE = 132.36026 WITH 81 DEGREES OF FREEDOM SIGNIFICANCE = 0.0002
L.S. TAU C = 0.50958 SIGNIFICANCE = 0.0000

234

DATA TRANSFORMATION DONE UP TO THIS POINT..
NO OF TRANSFORMATIONS 0
NO OF RECODE VALUES 0
NO OF ARITHM. OR LOG. OPERATIONS 0

FINISH

Los archivos más utilizados por SPSS son los siguientes :

tes :

- 1/ FILE5 .- Es el archivo de lectura (conocido por tarjetas - cómo IOCR) .
- 2/ FILE8 .- Es el archivo de lectura de datos , cuando el --- paquete no es accesado por tarjetas .
- 3/ FILE4 .- Este archivo es utilizado para almacenar un archivo SPSS con sus datos .
- 4/ FILE3 .- Es utilizado para leer un archivo y datos que se - hayan grabado utilizando FILE4
- 5/ FILE6 y FILE7 .- Son los archivos de impresión de SPSS .
- 6/ FILE9 .- Se utiliza cuando se accesa la instrucción de SPSS de WRITE CASES .
- 7/ FILE15,FILE16,FILE17,FILE18,FILE19,FILE20.- Son archivos que se utilizan cuando se requiere utilizar en un procedimiento una opción de RAW OUTPUT UNIT .

L I M I T A C I O N E S

- 1/ El paquete SPSS no se puede acceder en forma interactiva.
- 2/ Limitaciones sobre cada uno de los análisis se pueden ver en la referencia [5].
- 3/ Cuando se utiliza el procedimiento de regresión ; este en ocasiones no funciona , cuando en el modelo que se ajusta no se utiliza un nivel de opción .
- 4/ El desconocimiento de archivos que maneja el paquete .
- 5/ La versión que se utiliza es la número seis .
- 6/ Transformaciones se llevan a cabo unicamente por casos .
- 7/ Cuando mucho se pueden utilizar cinco digitos decimales en la impresión de resultados .
- 8/ No se cuenta con un manual de usuario , especifico para la computadora Burroughs B-6700 .
- 9/ No estan muy difundidas las aplicaciones de este paquete utilizando la computadora Burroughs B-6700.

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

En la actualidad el manejo de paquetes de computadora científicos y comerciales empieza a tener gran auge , debido a las facilidades que dan al usuario para resolver sus problemas.

Los paquetes como se menciona en el capítulo uno no requieren que el usuario tenga un gran conocimiento sobre ---- computación . Por otra parte los paquetes de computadora proveen generalmente más información de la que el usuario requiere , por lo que en algunas ocasiones al utilizarlos el usuario puede --- ubicar preguntas que no considero originalmente , así como en - nuevas alternativas en las que no se hubiera detenido "muchas - veces por no considerarlas" o porque en esos momentos no se le ocurrirían .

Por otra parte no conviene que el usuario se vicie en el manejo de paquetes de computadora para resolver sus problemas. Es decir que el usuario espere que todos sus problemas se ----- solucionen por medio de paquetes de computadora . Esto se podría enfatizar mejor si mencionamos el hecho de que el usuario podría complementar los resultados que emiten los paquetes ; en decisiones que el usuario tenga que tomar . Por lo que sería conveniente -- que el usuario se familiarice con alguno de los lenguajes de -- programación , con que cuenta la computadora , para que mediante programas que el usuario diseñe , se acoplen los resultados --- emitidos por los paquetes . Para determinar una decisión confiable que el usuario tenga que tomar en problemas grandes y complejos.

Por lo que se espera que el trabajo desarrollado , --- ayude a usuarios de la computadora Burroughs B-6700 en la solución de sus problemas de Estadística e Investigación de Operaciones . Considerando como una alternativa para el usuario, el manejo de - paquetes de Investigación de Operaciones y Estadística con que cuenta la computadora Burroughs B-6700 .

R E F E R E N C I A S

- 1 B7000/B6700 Series CANDE user's manual .
- 2 BURROUGHS B7700/B6700 systems tempo , mathematical programming system .
- 3 BURROUGHS tempo/network user's manual .
- 4 BURROUGHS advanced statistical inquiry system , basis large systems user's manual .
- 5 SPSS statistical package for the social sciences , second edition , Norman H. Nie , C. Hadlai Hull , Jean G. Jenkins , Karin Steinbrenner , Dale H. Bent , Mc. Graw-Hill Book Company .
- 6 Glover , F. D. , Klingman D. , & Stults J. 1973 . Augmented Threaded Index method for Network Optimization . Research Report CS-144 , Center for Cybernetic Studies . Austin University of Texas .
- 7 Kwak N: K. 1973 . Mathematical programming with business applications . Mc. Graw-Hill , New York .
- 8 Lasdon , Leon S. 1970 . Optimization theory for large systems . Mc. Millan , New York .
- 9 Hirshfeld , David S. 1972 . Very large linear programming models and how to solve them professionally . Paper presentated at 41st Natl. ORSA Meet . New Orleans .
- 10 Gass , S. L. 1969 . Linear programming : Methods and Applications Mc. Graw-Hill New York .
- 11 Samprit Chatterjee , Bertram Price , 1977 . Regression Analysis By Example . Jhon Wiley & Sons .
- 12 Rudolf J. Freund , Paul D. Minton . 1979 . Regression Methods A Tool for data analysis . Marcel Dekker , Inc. New York and Basel .
- 13 Draper , N. R. , and H. Smith . 1966 . Applied Regression Analysis . Wiley , New York .
- 14 Jhonston , J. 1972 . Econometric Methods , 2nd ed. , Mc. Graw-Hill New York .
- 15 B7000/B6000 Series WORK FLOW LANGUAGE , reference manual .