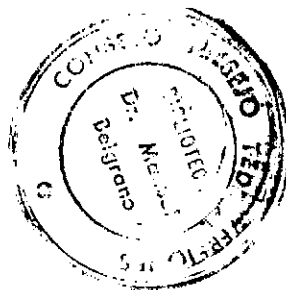


1030

CATALOGADO



PRIMER CURSO SOBRE  
ESPECIALIZACION EN EL SISTEMA  
S. A. S.



\* El presente texto se reproduce para uso exclusivo de los  
participantes de los Cursos dictados por el Consejo Fe- 677  
deral de Inversiones.-

1030

27034

507

MANUAL DE LENGUAJE  
DEL SISTEMA S. A. S.

B. 34

U. 151

AUTORA: LIC. MIRTA E. DE SOUSA  
AÑO: 1982  
REFERENCIA: SAS. USER'S GUIDE  
EDICION: 1979

\*\*\* I N D I C E \*\*\*

	Pág.
<u>CAPITULO I</u> - Introducción	
Introducción	2
Trabajo SAS	3 - 4
Paso DATA	5
Paso PROC.	6
Salida	7
Archivo SAS	7 - 8
<u>CAPITULO II</u> - Lenguaje	
Lenguaje SAS	9
Valores de datos	10
Variables	11
Observaciones y Nombres SAS	12
Sentencias	13 - 14
Expresiones, Constantes y Operadores	14 - 19
Valores faltantes	19 - 20
Errores	20 - 21
Ejemplos	22 - 23
<u>CAPITULO III</u> - Creación de Archivos SAS	
Creación de Archivos SAS y Entrada de Datos	24
Paso DATA	25 - 26
Sentencias:	
- DATA	26 - 27

# CONSEJO FEDERAL DE INVERSIONES

//.

- CARDS	28
- INFILE	28 - 29
- INPUT y Formatos	29 - 39

## CAPITULO IV - Manejo de Archivos SAS

SET	41 - 49
MERGE	50 - 54
UPDATE	54 - 61

## CAPITULO V - Instrucciones de Acción

Asignación	63
Suma	64
Funciones:	65
- SAS	66
- Matemáticas	67
- Trigonométricas	67
- Estadísticas	67 - 68
DELETE	69
OUTPUT	70 - 74
STOP	74
RETAIN	75
LIST	76
IF - THEN	77 - 80

## CAPITULO VI - Instrucciones de Información y Control

DROP	82
KEEP	83

///.

## CONSEJO FEDERAL DE INVERSIONES

///.

RENAME	84
LABEL	84 - 85
IF THEN - ELSE	86
GØ TØ	86 - 88
RETURN	88
LINK Y RETURN	89
DO Y END	90 - 92

### CAPITULO VII - Instrucciones de Impresión y Salida a Archivos Externos.

FILE	94 - 96
PUT y Formatos	96 - 103

### CAPITULO VIII - Instrucciones Complementarias

Comentario	105
Título	106
BY	106 - 107
END	108 - 109
MACRO	110

**CAPITULO I**

**INTRODUCCION AL SISTEMA DE ANALISIS ESTADISTICO**

## CONSEJO FEDERAL DE INVERSIONES

### INTRODUCCION

SAS es un sistema integrado para manejo de datos y análisis estadístico. Por la flexibilidad de la combinación estadística con las extensas capacidades para el manipuleo de datos y escritura, el SAS da un sistema total para ayudar a solucionar los problemas de computación; permite hacer simples tareas de proceso de datos y análisis estadísticos con poco esfuerzo.

SAS acepta datos de cualquier forma y desde cualquier dispositivo de entrada, así como saca resultados en cualquier dispositivo de salida.

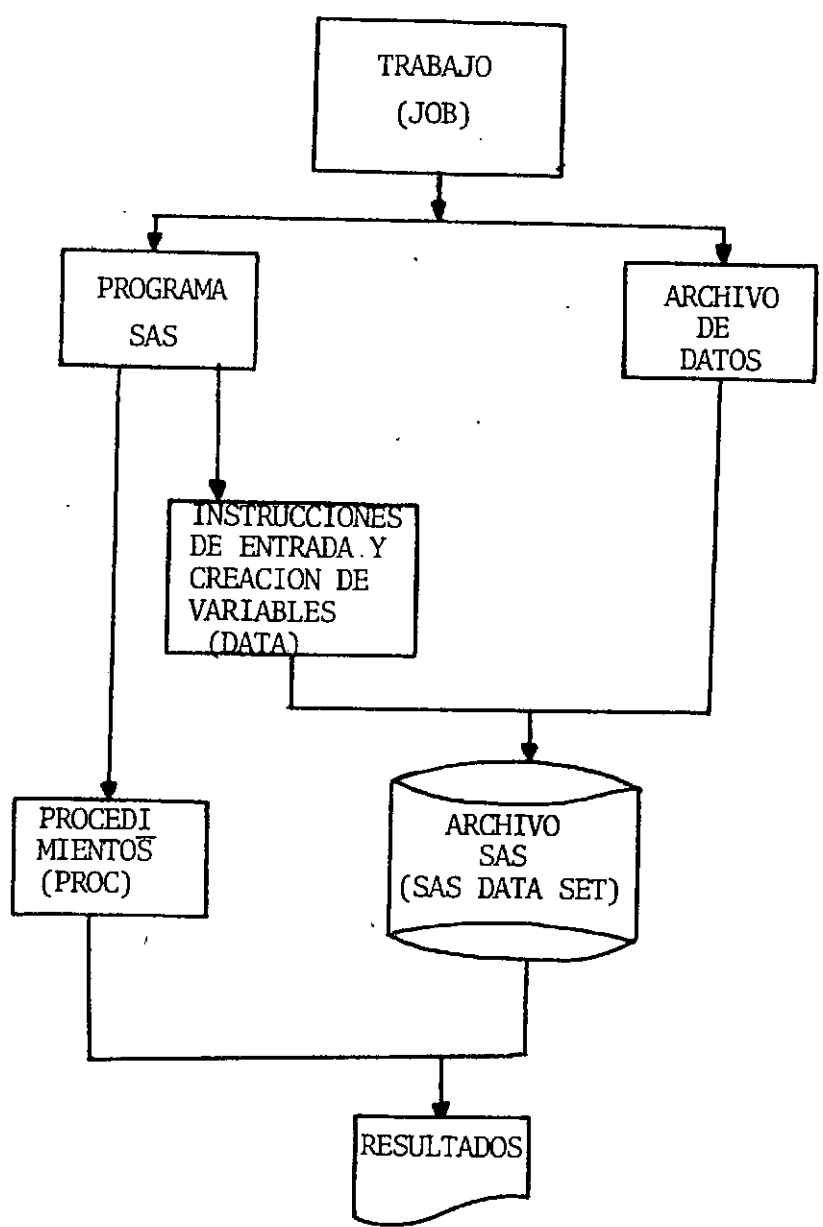
El lenguaje utilizado por SAS es similar al lenguaje PLI; las sentencias de SAS transforman y/o eliminan variables y observaciones, crean nuevas variables y nuevos conjuntos de datos de los conjuntos ya existentes, se puede entrelazar datos de dos o más archivos o subarchivos, encadenarlos, actualizarlos y crear subarchivos. Agregando estas características a los procedimientos de SAS para imprimir, clasificar, hacer ranking y graficar los datos da un manejo distinguido del sistema. Lo más destacado de la capacidad estadística del SAS es su facilidad para los procedimientos de mínimos cuadrados, lo que produce una amplia gama de análisis de regresiones lineales y no lineales, análisis de variancia y covariancia y análisis de variancia multivariada.

SAS provee también coeficientes de correlación parciales y múltiples, los coeficientes de correlación de Spearman y Kendall, tiene muchos procedimientos para analizar datos de series de tiempo, se puede obtener frecuencias simples y cuadros de frecuencia, puede ejecutar análisis discriminante, factorial y cluster; puede construir y evaluar la escala de Guttman, ejecuta test de confianza o análisis de probabilidades, genera diseños randomizados para experimentos, etc.

TRABAJO SAS

Un trabajo SAS consta de una secuencia de programas SAS y archivo de datos.

Gráfico 1





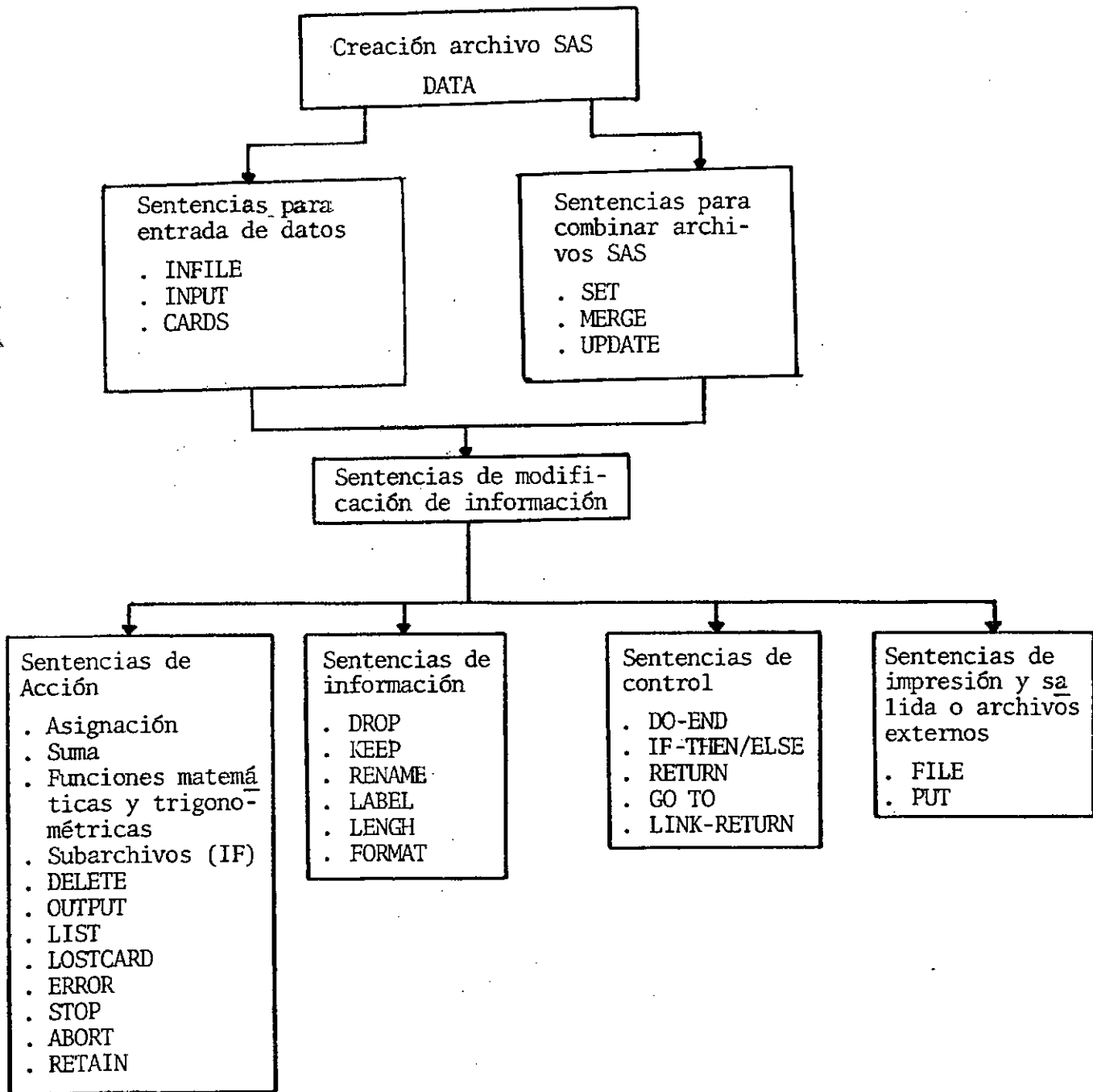
## CONSEJO FEDERAL DE INVERSIONES

Archivo de datos: El archivo tiene la forma estandar de cualquier archivo en cinta, disco, tarjetas, etc.

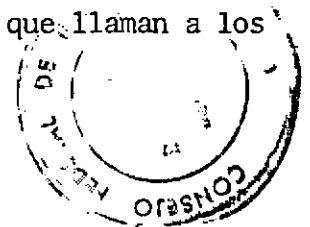
Programa SAS: Un programa SAS está formado por instrucciones SAS y consta de dos pasos bien definidos que son: el paso DATA y el paso PROC.

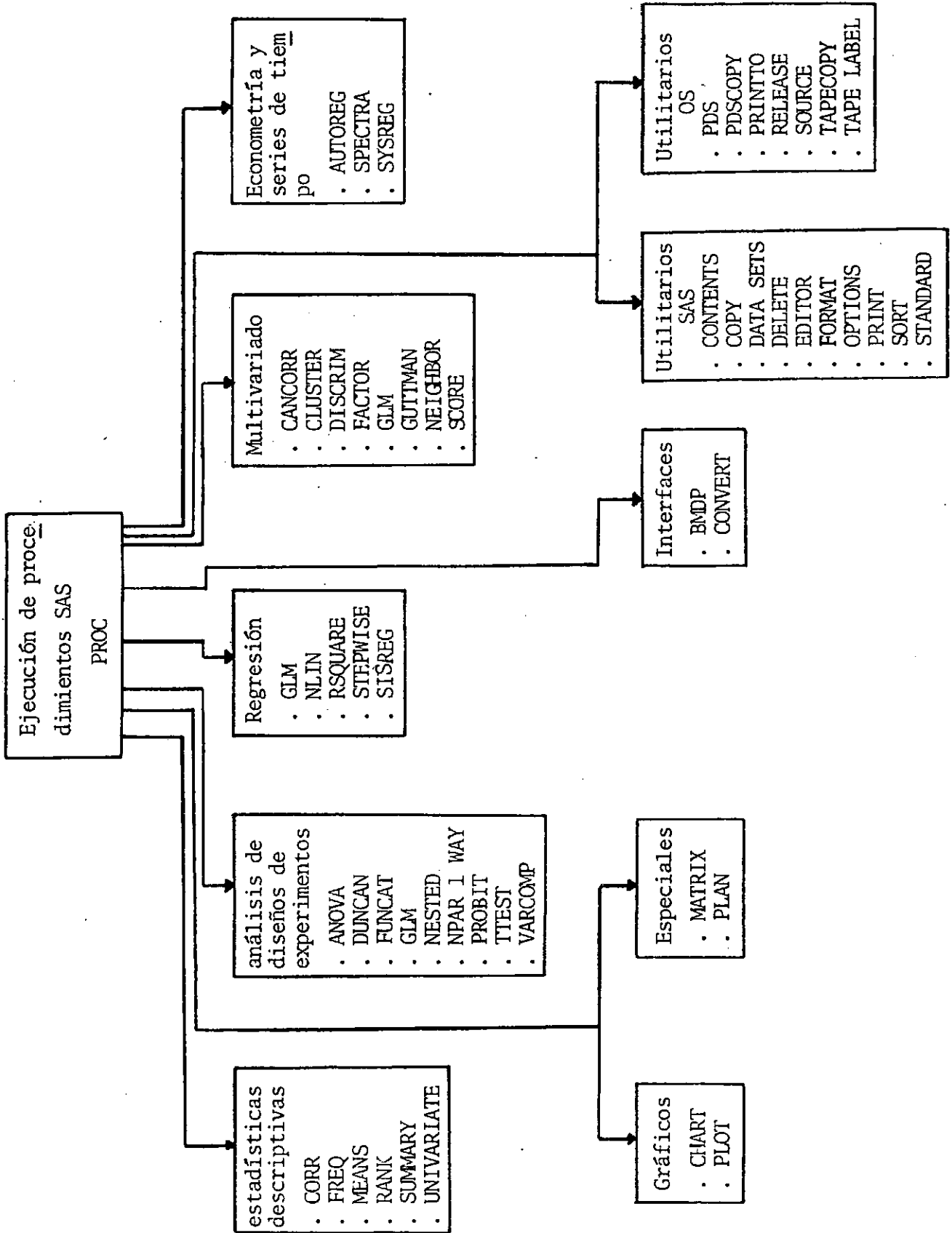
Si se observa el gráfico, vemos que el paso DATA actúa sobre el archivo de datos estandar obteniendo así un archivo SAS el cual tiene un formato particular que se vera mas adelante y sobre el cual únicamente tiene posibilidad de actuar el paso PROC. para obtener los resultados requeridos.

- Paso DATA: Este paso es el de creación del archivo SAS, obteniendo como resultado el archivo necesario para aplicar el análisis que corresponde.



- Paso PROC: En este paso se encuentran todos los procedimientos que llaman a los algoritmos para obtener el resultado requerido.





## CONSEJO FEDERAL DE INVERSIONES

### Salida de un trabajo SAS

La salida de un trabajo SAS es impresa en dos secciones:

- El SAS log: donde aparece la lista de todas las sentencias del programa numeradas; comentarios iniciados con la palabra NOTA o ERROR que contienen información acerca de que archivo se está creando, cantidad de observaciones, memoria requerida para ejecutar la creación del archivo o de los procedimientos, página donde está la salida de ese procedimiento, mensajes de errores, etc. Termina con:

```
NOTE    SAS  INTITUTE INC
        SAS  CIRCLE
        BOX  8.000
        CARY, N.C. 27511
```

- El SAS print (páginas de salida) contiene
  - . Numeración de páginas (borde superior derecho) de acuerdo a lo referido anteriormente en el SAS log.
  - . La fecha y hora en la cual se corrió el trabajo.
  - . Se puede utilizar un largo y ancho de hoja no estandar.
  - . Si fueron colocados títulos estos aparecen en los primeros renglones de cada hoja.
  - . Salidas de todos los procedimientos requeridos.

### Archivo de datos SAS

Un archivo SAS es una colección de observaciones de datos. Tiene forma de matriz donde las columnas son las distintas variables ó atributos y las filas son las distintas observaciones.

CONSEJO FEDERAL DE INVERSIONES

OBS	VAR	Y	Y	-----										Y	
		R	R												R
		1	2												M
1	A														Z
2															
3															
4															
·															
·															
·															
·															
·															
N															

A es el valor del dato leído para la primera observación y la primera variable.

Z es el valor del dato leído para la primera observación de la última variable.

**CONSEJO FEDERAL DE INVERSIONES**

CAPITULO II

LENGUAJE SAS

## CONSEJO FEDERAL DE INVERSIONES

### Lenguaje SAS

SAS tiene su propio vocabulario y sintáxis.

### Valores de los datos

Están clasificados en alfanuméricos y numéricos.

#### - Valores alfanuméricos

Pueden tener hasta 200 caracteres, incluyen letras, números, blancos o cualquier caracter especial.

Ejemplo:

```
HR304  
VARIEDAD 1  
Ø 'NEILL
```

#### - Valores numéricos

Pueden estar precedidos de signo + ó -, no puede haber comas (,), un valor numérico puede ser escrito con o sin punto decimal.

(La coma decimal se escribe con . siempre).

Ejemplo:

```
75  
.0057  
-5.  
8214.7221
```

Los valores numéricos con que trabaja SAS están en el rango de

$10^{-73} < n < 10^{73}$

CONSEJO FEDERAL DE INVERSIONES

Variables (VAR)

Una variable es una colección de datos representando el mismo atributo. Por ejemplo: en la matriz anterior los N valores de la primera columna van a formar la variable VAR1.

Una variable puede ser alfanumérica o numérica según contenga datos alfanuméricos o numéricos.

Variable numérica: Solamente puede contener datos numéricos.

Variable alfanumérica: Pueden conter datos numéricos o alfabéticos.

SAS no puede ejecutar operaciones aritméticas con variables de datos alfanuméricos, pero si con variables de datos numéricos.

El número máximo de variables en un archivo SAS es 4.000.

Lista de variables numerada:

Para definir o utilizar variables como lista numerada se utiliza la forma:

a .... n.... - a..... n.....

donde: a..... es un nombre de 1 a 7 caracteres.

n..... es un número de 1 a 7 dígitos

pero el total de caracteres mas los dígitos no puede exceder 8.

Ej.: N1 N2 N3 N4 N5 son los nombres de 5 variables, usando la lista abreviada se escribe N1-N5 representando las mismas 5 variables.

No debe mantenerse la misma cantidad de dígitos numéricos para la variable inicial y final.

Ej.: Supongamos 10 variables N1 N2 N3 N4 N5 N6 N7 N8 N9 N10



## CONSEJO FEDERAL DE INVERSIONES

Incorrecto	Correcto
NO1 - N10	NL-N10
NO5 - NO8	N5-N8
	N1-N3
	N6-N10

Orden de las variables: El orden de las variables en un archivo SAS responde al orden secuencial en que aparecen en las distintas instrucciones.

### Observaciones (OBS)

Una observación es un grupo de valores de datos que representan distintos atributos de un mismo individuo.

Llamamos individuo a una persona, un animal de experimento, una región geográfica, etc.

En la matriz vista anteriormente los datos de las variables VAR1, VAR2, ..., VARm en cada fila construye una observación.

No hay límites en el número de observaciones que un archivo SAS puede contener.

### Nombres de SAS

Los nombres válidos tienen de 1 a 8 caracteres de longitud, comenzando con una letra o un subguión, luego del primer carácter los demás pueden ser letras, números y subguiones, se utilizan para variables o rótulos de sentencias.

## CONSEJO FEDERAL DE INVERSIONES

Ejemplo de nombres:

<u>VALIDOS</u>	<u>NO VALIDOS</u>
H22A	REG_LØGICØ (+ de 8 caracteres)
REP_NØ	REP-NØ (caracter inválido)
_1972	1CAMP_AN (comienza con número)
BS_SA	BS AS (tiene blanco entremedio)

### Sentencias SAS

Las sentencias SAS dicen la operación a realizarse.

Cada sentencias SAS comienzan siempre con una palabra clave escrita en mayúscula, la cual debe colocarse si se quiere realizar esa acción.

A continuación, lo escrito en letra minúscula nos dice exactamente la información que debe colocarse en ese, sobre la cual se realizará la acción, y en tre paréntesis lo opcional (< >).

Ejemplo:

```
DRØP lista de variables;
```

Palabra clave DRØP.

Lista de variables: en ese lugar se deben colocar todas las variables que se deseen, separadas por un blanco entre variables.

Ejemplo:

```
DRØP X Y Z;
```

### Reglas generales de las sentencias

- La sentencia puede comenzar y terminar en cualquier posición de la tarjeta

## CONSEJO FEDERAL DE INVERSIONES

desde columna 1 a 80 inclusive.

- La sentencia termina siempre con un ; (punto y coma, con la única excepción de la sentencia MACRO que termina con %).
- Más de una sentencia puede aparecer en un solo línea (pues cada ; indicaría el fin de la instrucción).
- La sentencia SAS puede comenzar en una línea y continuar en otras.

Ejemplo:

```
DRØP X
      Y
      Z;
```

- Las variables o números en una sentencia SAS deben estar separadas por uno o más blancos ó caracteres especiales.

Ejemplo:

<u>CORRECTO</u>	<u>INCORRECTO</u>
DRØP X Y Z;	DROPXYZ;
X=Y; ó X= Y; ó X = Y; ó X = Y;	

### Expresiones, Constantes y Operadores

Dentro de las sentencias encontramos expresiones corrientes y operadores.

Expresiones: Una expresión es una secuencia de nombres de variables, constantes y/o nombre de funciones ligadas por un operador.

Ejemplo:

```
A = B+C;
```

## CONSEJO FEDERAL DE INVERSIONES

### Constantes

Una constante es un número o una secuencia de caracteres fija. Las constantes numéricas están escritas como valores de datos numéricos. La constante alfanumérica difiere de un valor de un dato alfanumérico, en que la constante debe estar encerrada entre apóstrofes ('A').

Una constante representando un valor missing es escrita con un simple punto (.) decimal.

Diferencia entre elemento de dato y constante.

<u>Dato</u>	<u>Constante</u>
0.05 (numérico)	0.05
1 (alfanumérico)	'1'
TRIGO "	'TRIGO'
. (missing) (numérico)	. (missing)

### Operadores

Los operadores le dicen a SAS como ejecutar algunos cálculos aritméticos o determinar la verdad o falsedad de algunas operaciones.

Hay 3 clases generales de operadores: aritméticos, de comparación y booleano.

#### - Operadores aritméticos

Los operadores aritméticos indican que un cálculo aritmético va a ser ejecutado.

Son:

\*\* exponenciación

\* multiplicación

CONSEJO FEDERAL DE INVERSIONES

/ división

+ suma

- resta

Ejemplo:

Operación a realizar

Escritura

$$B = A^3$$

$$B = A ** 3$$

$$B = 2X$$

$$B = 2 * X$$

Los operadores aritméticos deben ser únicamente aplicados a variables numéricas.

- Operadores de comparación

Proponen una relación entre dos cantidades y le pregunta a SAS si se cumple la relación, se se cumple, el resultado de llevar a cabo la operación es VERDADERO (1), si no se cumple es FALSO (0).

Operadores ó Operadores

igual	=	EQ
distinto	≠	NE
mayor	>	GT
no mayor	≧	LE
menor	<	LT
no menor	≦	GE
mayor o igual	>= ó =>	GE
menor o igual	<= ó =<	LE

Ejemplo I:

CONSEJO FEDERAL DE INVERSIONES

$A \leq B$   
 variables numéricas

Si  $A = 4$  y  $B = 3 \Rightarrow 4 \not\leq 3$  la expresión es FALSA y da como resultado 0.  
 Si  $A = 5$  y  $B = 9 \Rightarrow 5 \leq 9$  la expresión es verdadera y el resultado 1.

Ejemplo II:

$NOMBRE \neq 'MARIA'$   
 variable alfanumérica      constante alfanumérica

Si la variable NOMBRE tiene como dato el valor MARIA, la expresión es FALSA (0).  
 Si la variable NOMBRE tiene un dato distinto de MARIA, la expresión es VERDADERA (1).

Ejemplo III:;

$N \neq 5$   
 variable numérica      constante numérica

El resultado será verdadero o falso según sea el valor de la variable N.

- Operadores Booleanos

Son utilizados en expresiones que incluyen operadores de comparación. Los operadores Booleanos son:

- AND
- ØR
- NØT

Tablas de verdad de cada operador.

CONSEJO FEDERAL DE INVERSIONES

AND

	V	F
V	V	F
F	F	F

ØR

	V	F
V	V	V
F	V	F

NØT

	V	F
	F	V

Tambin como resultado dar un resultado de verdad o falsedad.

Ejemplo: AND

A<B AND C>0

La expresin tendr valor (1)verdadero cuando A sea menor que B y C positivo.

EDAD>12 AND EDAD<30

Esta expresin ser verdadera solamente cuando la variable edad tenga como dato un valor mayor que 12 y menor que 30; tambin se puede escribir como:

12<EDAD<30 6 30>EDAD>12

Ejemplo: OR

A<B OR C>0

Solamente es FALSO cuando ambas condiciones son falsas, sino siempre es ver

## CONSEJO FEDERAL DE INVERSIONES

dadero.

Ejemplo: NOT

NOT (NOMBRE='MARIA')

Primero compara lo que está entre paréntesis puede dar verdadero (1) ó falso (0).

Luego el NOT con el paréntesis verdadero da como resultado FALSO; el NOT con el valor del paréntesis falso da como resultado VERDADERO.

NOT (NOMBRE='MARIA') se puede escribir también

NOMBRE $\neq$ 'MARIA'

El orden de la ejecución de las operaciones es el mismo que en FORTRAN.

### Valores faltantes (missing)

Los valores de datos de entrada faltantes pueden representarse con punto (.) decimal o dejando en blanco el campo. SAS transforma en cualquiera de los dos casos en valor missing.

Este valor missing puede ser utilizado en operaciones de comparación; cada procedimiento aclara que hace con estos valores pero en general cuando un valor missing en una variable aparece y esta entra a un procedimiento la observación se elimina.

### Propagación de los valores missing

Cuando un argumento de uno de estos operadores + - \* / \*\* es missing ó cuan



## CONSEJO FEDERAL DE INVERSIONES

do un argumento de una función es missing, el resultado de la operación o de la función es también missing. Esto significa la propagación del argumento al resultado.

Ejemplo I:

$$Y = X1 + X2 ; \Rightarrow . = . + 5$$

Si X1 y/o X2 es missing Y resulta missing.

Ejemplo II:

$$SUM = SUM + A ;$$

Supongamos un proceso de iteración, basta que una vez el valor de A sea missing para que el resultado final sea missing.

### Errores

Cuando un error es detectado se debe a:

- a) Valor no numérico en un campo numérico.
- b) Falta de valores por parte de una observación.
- c) Dato con más cantidad de dígitos que los supuestos.
- d) El uso de alguna función con argumento inválido.
- e) División por cero.
- f) Variables en un procedimiento no definidas anteriormente.
- g) Falta del ; .
- h) etc., etc.

**CONSEJO FEDERAL DE INVERSIONES**SAS en presencia de los errores tipo (a-e)

- 1) Da a la variable con problemas, valor missing.
- 2) Escribe una NOTA donde dice que pasó con esa variable, en que línea, e imprime la observación con problema.
- 3) Error de Sintáxis. Da un número debajo del lugar donde se detectó el error, y luego más abajo escribe ese número de error y su significado.



S T A T I S T I C A L   A N A L Y

NOTE THE JOB CAS07 HAS BEEN RUN UNDER RELEASE 79.5 OF SAS AT C.I.IN.-C.F.I.

NOTE SAS OPTIONS SPECIFIED ARE  
SORTM4

1 DATA CAS07;  
2 INPUT XI XF F;  
3 XM=((XF-XI)/2) \* XI;  
4 CARDS;

NOTE DATA SET WORK.CAS07 HAS 5 OBSERVATIONS AND 4 VARIABLES. 232 OBS/TRK.  
NOTE THE DATA STATEMENT USED 0.00 SECONDS AND 172K.

10 OPTIONS LS=12;  
11 FRCC PRINT; VAR XI XF XM F;

NOTE THE PROCEDURE PRINT USED 0.00 SECONDS AND 180K  
AND PRINTED PAGE 1.

12 FRCC MEANS; VAR XM; FREQUENCY F;

NOTE THE PROCEDURE MEANS USED 0.00 SECONDS AND 180K  
AND PRINTED PAGE 2.

NOTE SAS USED 180K MEMORY.

NOTE SAS INSTITUTE INC.  
SAS CIRCLE  
BOX 8000  
CARY, N.C. 27511

**CONSEJO FEDERAL DE INVERSIONES**

**CAPITULO III**

**CREACION DE ARCHIVO SAS Y ENTRADA DE DATOS**

## CONSEJO FEDERAL DE INVERSIONES

### Paso Data

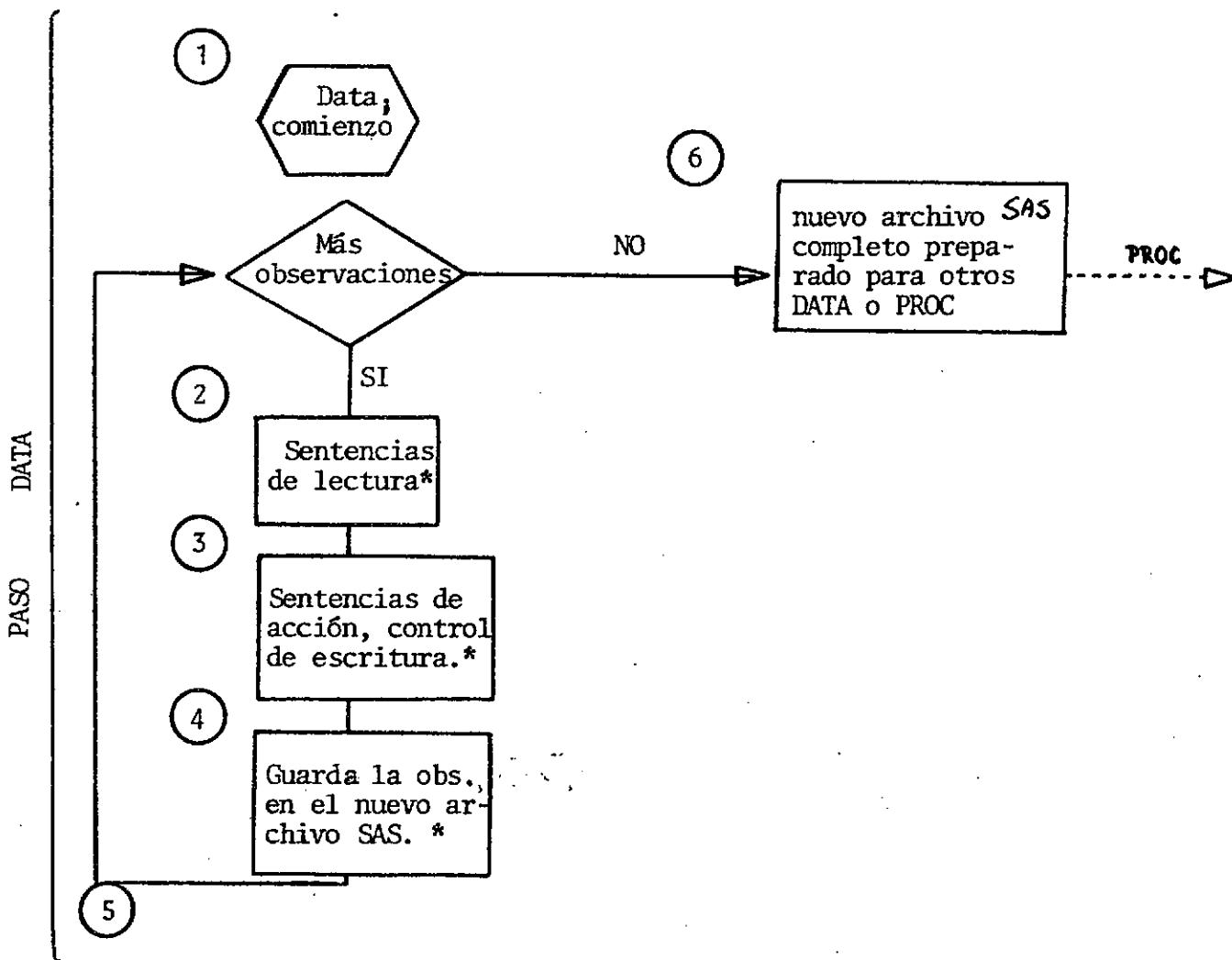
Todos los trabajos SAS consisten de 2 clases de blocks bien definidos, los pasos DATA y los pasos PROC. Los pasos DATA se usan para crear archivos de datos SAS y para hacer otros procesamientos como imprimir reportes, tarjetas perforadas y sacar archivos en cinta o disco. Los pasos PROC se usan para analizar los archivos de datos SAS creados en posteriores pasos DATA.

Este paso se usa para los siguientes propósitos:

- recuperar, tomando datos de entrada dentro de un archivo de datos SAS.
- editar, chequeando errores en sus datos y corrigiéndose datos, calculando nuevas variables.
- imprimiendo reportes acorde a sus especificaciones, escribiendo archivos en disco y cintas y perforando tarjetas.
- produciendo nuevos archivos de datos SAS, desde la existencia de sub-archivos, combinación de archivos y actualización de viejos archivos de datos.

### Sentencias de un paso DATA

Un paso data siempre comienza con una sentencia DATA, para señalar el comienzo a SAS del paso. Las sentencias siguientes SAS las traduce a lenguaje de máquina y son ejecutadas usualmente todas, por cada una de las observaciones de sus datos de entrada.



\* observación corriente.

Los pasos 2 a 5 se repiten tantas veces como observaciones se estén leyendo.

En el paso 6 queda preparado un nuevo archivo SAS con todas las observaciones creadas con los pasos 2 a 5.

Sentencia DATA

Forma general: DATA (nombre del archivo SAS);

## CONSEJO FEDERAL DE INVERSIONES

Esta es la primera instrucción en la mayoría de los programas SAS.

Su función es iniciar el paso DATA y la creación de un archivo SAS asignándole un nombre que nos permitirá referenciarlo en otros puntos del programa.

En los programas simples podemos emitir el nombre, en cuyo caso SAS automáticamente le asigna el nombre DATA1, DATA2, y así sucesivamente para cada sentencia DATA que haya en el programa.

La forma de escribir esta sentencia sería:

```
DATA;
```

El uso de nombres particulares facilita la tarea del programador, ya que le permite identificar el tipo de información que contiene cada archivo.

Los nombres de los archivos, deben cumplir los requisitos de los nombres aceptados por SAS.

En el caso de creación de varios archivos simultáneamente.

```
DATA (lista-nombres de archivos);
```

Esta sentencia deberá ser complementada con otras que nos permitan seleccionar cuáles observaciones deben ser incluidas en cada uno de ellos, en caso contrario todos contendrán la misma información.

Ejemplos:

```
DATA;  
DATA RIEGØ;  
DATA UNØ DØS TRES;
```

A continuación de ésta debe seguir otra sentencia que indique de dónde debe ser tomada la información:

- Si la misma es leída, deberá continuar una sentencia INPUT (y a continuación CARDS)
- Si fuera tomada de archivos ya existentes en el trabajo continuará SET.
- Para combinar las observaciones de dos archivos se usará MERGE.
- Para actualizar los datos de las observaciones de un archivo anterior, colocaremos UPDATE.



## CONSEJO FEDERAL DE INVERSIONES

- Instrucciones opcionales para seleccionar y/o modificar observaciones y variables.

### Sentencias para entrada de datos

#### Sentencias CARDS

Forma general: CARDS;

Esta sentencia se usa en los casos en que los datos son leídos durante la creación de un archivo.

Le indica a SAS que a continuación se encuentran las tarjetas o líneas que contienen los datos. En caso de proceder la información de cintas y/o discos magnéticos, deberán usarse otras instrucciones:

```
Ejemplo: DATA LEØ;
          INPUT (lista);
          CARDS;
          012345
          790600
          =
          =
          =
          } datos de entrada
```

SIEMPRE DEBEN CONTINUAR LOS DATOS A UNA SENTENCIA CARDS.

En general usamos esta instrucción cuando trabajamos con tarjetas, diskettes o desde una terminal. Los datos y el programa son leídos en el mismo periférico.

#### Sentencia INFILE

Forma general:

INFILE (nombre que referencia a un archivo);

Esta sentencia es usada cuando los datos de entrada (archivo), se encuentran en disco o cinta.

Se utiliza el nombre que referencia al archivo para identificar las especificaciones particulares del archivo y el tipo de periférico.

**CONSEJO FEDERAL DE INVERSIONES**

```

Ejemplos: // VIEJO DD DSN = DATOS, UNIT = TAPE, DISP = OLD, LABEL = 1,
          DATA LEO;
          INFILE VIEJO;
          INPUT (lista);
          .
          .
          .

```

Sentencia INPUT

La información del archivo de datos con el cual queremos trabajar puede estar en tarjetas, disco, cinta, etc.

La sentencia INPUT describe al SAS la forma de las líneas de datos, y construye así las observaciones (OBS1....OBSn).

Esta sentencia tiene muchas formas, y todas ellas serán presentadas,

Forma general: INPUT ( lista de variables < formatos > );

Sentencia INPUT sin formato:

Lista INPUT

Los nombres de variables son los que se quiere para cada una de ellas, las cuales deben estar sepradas por uno o más blancos.

```

Ejemplo: INPUT SEXO EDAD ALTURA;

```

Archivos SAS

```

Ejemplo: DATA A;
          INPUT X Y;
          CARDS;
          12 54.7 38
          14 76.9
          2 83.7

```

	OBS	VAR	X	Y
	1		12	54.7
	2		14	76.9
	3		2	83.7

SAS asigna el primer valor de cada línea a la primera variable, el segundo valor a la segunda; el tercer valor de la primera línea la ignora pues no existe variable a quién asignárselo.

## CONSEJO FEDERAL DE INVERSIONES

Para que se pueda utilizar la lista de variables sin formatos en la sentencia INPUT, los datos deben estar separados por lo menos con un blanco; pues SAS toma como dato para la primera variable desde el primer caracter no blanco hasta el primer blanco que aparezca, y así siguiendo.

### Lista INPUT de variables alfanuméricas

Para describir variables alfanuméricas en una lista debe colocarse después del nombre de la variable el signo "\$".

```
Ejemplo:  DATA A;
           INPUT  PROVINC  $  FRAC  RADIO;
           CARDS;
           CORDOBA  1  1
           CORRIENTES 2  1
```

Observaciones	Variables	PROVINC	FRAC	RADIO
1		CORDOBA	1	1
2		CORRIENT	2	1

SAS toma el dato de la variable PROVINC del primer no blanco al primer blanco. Si ese dato tiene más de 8 columnas se debe utilizar otro formato, si el dato tiene un blanco intermedio debe utilizarse el signo "&" ó "E" antecediendo o precediendo al signo \$ y los datos de las distintas variables deben estar separadas por 2 ó más blancos.

```
Ejemplo;  DATA A;
           INPUT  PROVINC  & $  FRAC  RADIO;
           CARDS;
           S LUIS  1  4
           SANTA FE  8  7
           BUENOS AIRES 3  1
```

**CONSEJO FEDERAL DE INVERSIONES**

O B S	V A R	PROVINC	FRAC	RADIO
1		S LUIS	1	4
2		SANTA FE	8	7
3		BUENOS A	3	1

INPUT con formato:

Se puede utilizar formatos en una lista INPUT para leer variables alfanuméricas o para leer variables en otras notaciones, escribiendo ":" y un formato después del nombre de la variable.

Ejemplo: INPUT CIUDAD : \$ 20.;

SAS busca la primera columna no blanca, empezando de esa columna lee un valor de 20 posiciones para la variable CIUDAD ó hasta que encuentre un blanco.

Si un blanco entremedio puede aparecer debe colocarse "\$" y leerá las 20 posiciones o hasta que encuentre 2 o más blancos.

INPUT por columna:

Cuando el valor de una variable ocupa las mismas posiciones en todas las líneas de datos, se puede utilizar la entrada por columna.

Ejemplo: INPUT EDAD 5-7 ALTURA 8-10 CØD 11;

SAS sabe que el valor de edad en cada línea la encuentra en las columnas 5 a 7, altura en las columnas 8 a 10 y código en columna 11, e ignora si hay blancos delante y detrás de estas columnas. Este tipo de formato se usa generalmente cuando los datos de cada variable no están separados por blancos.

Si dentro de las columnas que se indican para leer un dato encuentra un blanco intermedio da un mensaje de error.

## CONSEJO FEDERAL DE INVERSIONES

La forma general de este INPUT es:

INPUT variable <comienzo de col-fin de col>;

INPUT por columna con decimales:

Se puede insertar el punto decimal en el valor de una variable numérica, se éste ya no tiene anteriormente.

Ejemplo: DATA A;  
INPUT N 1 PESO 10-15 3;  
CARDS;  
1 80750  
2 51150  
3 49.80

O B S V A R		N	PESO
1		1	80.750
2		2	51.150
3		3	49.80

INPUT por columna para variables alfanuméricas:

En la sentencia INPUT, debe aparecer "\$" después del nombre de la variable.

Ejemplo: INPUT NOMBRE \$1-10;

SAS tomará el dato de las columnas 1 a 10 para el valor de NOMBRE; eliminando los blancos que están antes y después.

Si blancos entremedio aparece debe colocarse "\$".

Ejemplo: INPUT NOMBRE \$ 1-10;

## CONSEJO FEDERAL DE INVERSIONES

Si SAS está mirando la décima columna de la segunda línea de datos, los indicadores dicen

línea = 2 y columna = 10

Estos indicadores pueden ser modificados de acuerdo a las necesidades de la lectura o impresión.

### Pointer de columna "Q"

Ejemplo:

```
INPUT Q26 FECHA Q45 SIEMBRA;
```

Q26 mueve el indicador de columna a la posición 26 a partir de allí busca el dato de fecha, siguiendo hasta el primer no blanco, luego mueve a columna 45 y saca el valor de siembra hasta el primer no blanco.

El indicador debe estar seguido de un número o variable numérico entera y positiva.

Ejemplo:

```
INPUT Q26 FECHA;  
INPUT NQ N FECHA;  
INPUT 78 ENCUES Q10 ESTRATQ Q1 SUPERF;
```

### Pointer de línea "#"

A menudo 2 o más líneas de datos pueden formar una observación. El indicador de línea se puede mover a cualquier línea de esa multi-lineal observación escribiendo el "#" seguido por el número de línea antes del nombre de la variable.

El número de línea puede ser un número o una variable entera y positiva.

Ejemplo:

```
INPUT #3 FECHA;  
ó  
INPUT #N FECHA; donde N debe ser dada anteriormente (N =3).
```

En estos casos el indicador se posiciona en la primera columna de la línea 3.

## CONSEJO FEDERAL DE INVERSIONES

Cuando # y @ son usados juntos # debe ser usado antes.

Ejemplo: INPUT #8 @26 LIBRO #1 COMPANIA;

Esto indica que la observación está formada por datos de 8 líneas, las cuales quedan todas a disposición del INPUT para leer todos los datos necesarios, en cualquier orden.

También podemos escribir "+" seguido por un número o una variable, esta produce que el indicador se mueva hacia adelante tantas posiciones como se indique.

Ejemplo: INPUT @10 FECHA +20 EDAD;

Se puede utilizar también "/" para mover el indicador a la próxima línea.

Por ejemplo:

INPUT EDAD / SEXO PROFES;

En la primera línea toma el valor de edad salta a la segunda línea y toma el valor de sexo y profesión.

### Muchas líneas por Observación

Cuando 2 o más líneas de datos son leídas se utiliza el "#", una vez que el cardinal apunta a una línea para una variable, las demás variables no necesitan el cardinal (en esa línea).

Por ejemplo si una observación tiene 4 líneas pero solamente los datos de la línea 1 y 2 son necesarios.

```
INPUT #1 X 1-7 #2 Y 23-30 #4;
6 INPUT X 1-7 #2 Y 23-3- #4;
6 INPUT X 1-7 / Y 23-30 #4;
```

Si no se coloca el # no podemos retornar a una línea anterior para tomar información.

## CONSEJO FEDERAL DE INVERSIONES

Ejemplo: INPUT A/B #1 @ 52C;

Esta instrucción no funciona, pues hace que se lea la variable A de la primera línea, salte a la segunda línea y lea el valor de B, solamente la última línea leída es ahora disponible para el indicador, la primera línea no está más disponible para el indicador, por lo tanto el # 1 produce un mensaje de error.

La sentencia puede ser escrita:

```
INPUT A/B #1 @52 C #2;
```

### Otro uso para el @

El "@" en el final de una sentencia INPUT es un indicador especial de dirección: le dice al SAS que deje el indicador en la posición actual (retenga).

La siguiente sentencia INPUT ejecutada podrá usar esa misma línea o conjunto de líneas en vez de leer en una nueva como es costumbre.

```
Ejemplo: DATA A;
          INPUT TIPO @;
          IF TIPO = 1 THEN INPUT @ 26 I P;
          IF TIPO = 2 THEN INPUT @ 32 P I;
```

La observación está formada por una línea que contiene tipo de tarjeta, el valor de identificación y peso, pero estos últimos están en distinto orden según la tarjeta sea de tipo 1 ó 2, con este juego de instrucciones los datos son leídos en la forma correcta.

Para leer líneas con más de una observación de la misma variable se usa

```
Ejemplo: DATA A;
          INPUT X Y @@;
          CARDS;
          14.4. 53 13.7 55 17.8 41 14.5 54
          :
          :
```



**CONSEJO FEDERAL DE INVERSIONES**

O B S	V A R	X	Y
1		14.4	53
2		13.7	55
3		17.8	41
4		14.5	54
5		⋮	⋮
⋮		⋮	⋮
N		⋮	⋮

FORMATOS

Se pueden utilizar formatos para leer datos en casi cualquier notación.

Los formatos le dicen a SAS cuantas columnas son ocupadas y como leer los datos para la variable.

Los formatos más comúnmente usados son aquellos para variables numéricas y alfanuméricas.

Variables numéricas

Escribiendo 'w.' le dice a SAS comenzar con la actual columna y leer los próximos w columnas como el valor de la variable.

EL w debe ser entero positivo

Ejemplo:

```
INPUT @ 25 ANIMAL 3;
```

mueve el pointer a la columna 25, SAS comienza allí y lee las columnas 25,26 y 27 como el valor de ANIMAL.

Escribiendo "w.d " le dice a SAS leer el valor de la variable como anteriormente y luego insertar un punto decimal, antes de los últimos d dígitos.

## CONSEJO FEDERAL DE INVERSIONES

Ejemplo: INPUT @10 PRECIO 6.2;

Le dice a SAS comenzar en la columna 10, leer las próximas 6 columnas como el valor de PRECIO e insertar el punto decimal antes de los 2 últimos dígitos

Si un valor de dato tiene ya un punto decimal SAS lo deja en su lugar, utilizando solamente el valor de w. para saber la cantidad de dígitos que corresponde a esa variable (incluido el punto) y d no lo utiliza.

### Variables alfanuméricas

Escribiendo "\$w" le dice a SAS comenzar en la actual columna y leer las próximas w columnas como un valor alfanumérico.

Ejemplo: INPUT @30 NOMBRE \$ 45;

Llama al SAS para leer de columnas 30-74 para el valor de la variable NOMBRE los blancos delante y detrás son descartados.

Si se desea mantenerlos se debe usar el formato "\$CHARw."

Ejemplo:

INPUT @30 NOMBRE \$ 45;

### LISTA DE FORMATOS

Se pueden escribir muchas variables de una sola vez con una lista de Formatos.

Dicha lista incluye las direcciones de los indicadores y los formatos.

Una sentencia INPUT con lista de formato tiene la siguiente forma.

INPUT (lista de variables) (direcciones de los indicadores y formatos);

Tanto la lista de variables como los formatos deben estar encerrados entre paréntesis "( )"

Ejemplo: INPUT (DIA1) (@20 5.);

Esta forma vemos que está entre paréntesis a diferencia del caso visto anteriormente como:

## CONSEJO FEDERAL DE INVERSIONES

INPUT @20 DIA1 5;

6

INPUT . DIA1 20-24;

Para SAS estos tres casos trabajan exactamente igual.

Ejemplo II:

```
INPUT (NOMBRE RANGØ) ($30 @50 3.);
```

SAS lee las primeras 30 columnas como un alfanumérico para la variable NOMBRE, va a la columna 50 y lee 3 columnas como numérico para la variable RANGØ..

Ejemplo III:

```
INPUT (RANGØ 1 - RANGØ 3) (5.);
```

6

```
INPUT (RANGØ 1 - RANGØ 3) (5. 5. 5.);
```

Estos dos formatos son idénticos.

Ejemplo IV:

```
INPUT (RANGØ 1 - RANGØ 3) (3*5.);
```

Un formato puede ser repetido utilizando una constante entera y un signo "\*".

Los formatos no pueden ser anidados.

Ejemplo:

```
INPUT (R1 -R6 F1 - F2) (3* (5. 1.) 2. 3.);
```

debe colocarse:

```
INPUT (R1 - R6 F1 - F2) (5. 1. 5. 1. 5. 1. 2. 3.);
```

6

```
INPUT (R1 - R6) (5. 1.) (F1 - F2) (2. 3.);
```

**CONSEJO FEDERAL DE INVERSIONES**

Combinación de métodos en INPUT

Muchos tipo de formatos para lectura de datos en SAS, pueden ser combinados en una sentencia INPUT.

Ejemplo:

```
INPUT X 54-56 Y Z;  
INPUT A 10-12 (Q1 - Q20) (@31 20*2.);  
INPUT @10 A 3. @31 (Q1-Q20) (2.);  
INPUT @5 Y A.15-20 @31 (Q1 -Q10) (2.+2);
```

**CONSEJO FEDERAL DE INVERSIONES**

**C A P I T U L O   I V**

**MANEJO DE ARCHIVOS SAS**

## CONSEJO FEDERAL DE INVERSIONES

### Sentencia SET

Se utiliza esta sentencia para crear un nuevo conjunto de datos de uno o más conjuntos de datos SAS ya existentes.

Las observaciones de uno o más archivos viejos pueden ser modificados o concatenados para producir un nuevo conjunto.

Esta sentencia siempre sigue a la sentencia DATA.

La forma general de escritura es:

SET (nombre-viejo-archivo);  
o bien  
SET (lista-nombres de viejos archivos);

Donde la lista puede incluir hasta 50 nombres.

Debemos recordar que los archivos sólo están disponibles durante la ejecución del programa.

Usos de la sentencia SET:

- Modificar o sustituir las observaciones de un simple archivo SAS, creando otro nuevo.
- Concatenar dos o más archivos SAS, en este caso el nuevo archivo tendrá las observaciones del primer archivo más las del segundo archivo y así sucesivamente.
- Intercalando observaciones de 2 o más archivos, formando uno nuevo clasificado.
- Escribir reportes, en este caso especifica el archivo que contiene la información.

#### 1. Modificación y Sustitución del Archivo

La secuencia es:

CONSEJO FEDERAL DE INVERSIONES

DATA NUEVO;  
SET VIEJO;

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_ } sentencias de programa

Cualquier número de sentencias de programa pueden ser usadas, si no se colocan sentencias, el nuevo archivo será una copia fiel del viejo.

Ej:

DATA VIEJOS;  
SET POBLACION;  
IF EDAD > 65;  
=

El archivo POBLACION tiene toda la población existentes, en cambio el archivo VIEJOS, solamente tendrá a las personas de más de 65 años.

Ej:

DATA A;  
SET TEST;  
B = C+D;  
=

El archivo TEST tendrá todas las variables del archivo A mas la nueva variable B creada por programa.

2. Concatenación de Archivos

DATA NUEVO;  
SET VIEJO1 VIEJO2;

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Hasta 50 archivos se pueden concatenar de esta forma.

CONSEJO FEDERAL DE INVERSIONES

El archivo NUEVO contendrá las observaciones del archivo VIEJO1 y a continuación las observaciones del archivo VIEJO2.

Los archivos concatenados no necesariamente necesitan tener las mismas variables, pues si contienen algunas, estas estarán en missing en el archivo que no existan.

Ejemplo I:

DATA A;  
SET B C;

=====  
=====  
=====  
=====

El archivo

B tiene:

VAR \ OBS	Z <sub>1</sub>	Z <sub>2</sub>
1	8	5
2	4	3

El archivo

C tiene:

VAR \ OBS	Z <sub>1</sub>	Z <sub>3</sub>
1	5	4
2	9	1

El archivo A tendrá la siguiente información:

VAR \ OBS	Z <sub>1</sub>	Z <sub>2</sub>	Z <sub>3</sub>
1	8	5	.
2	4	3	.
3	5	.	4
4	9	.	1

} B  
 } A  
 } C



CONSEJO FEDERAL DE INVERSIONES

Ejemplo II: Supongamos tener los siguientes archivos

UNØ			
VAR			
Obs	A	B	C
1	10	7.9	5
2	20	10.6	7
3	30	15.0	9

DØS			
VAR			
Obs	A	B	C
1	40	8.6	9
1	10	7.5	7

TRES			
VAR			
Obs	A	B	C
1	50	6.52	8
2	10	7.3	7
3	20	10.5	5
4	60	8.9	2

Con la siguiente secuencia de instrucciones SAS:

```
DATA NUEVO;
```

```
SET UNØ DØS TRES;
```

```
—  
—  
—  
—
```

procedimientos

obtenemos el siguiente resultado:

CONSEJO FEDERAL DE INVERSIONES

VAR Obs	A	B	C
1	10	7.9	5
2	20	10.6	7
3	30	15.0	9
4	40	8.6	9
5	10	7.5	7
6	50	6.52	8
7	10	7.3	7
8	20	10.5	5
9	60	8.9	2

contenido del archivo  
NUEVO

En estos casos se pueden presentar dos archivos que no contengan las mismas variables, los valores faltantes son considerados "missing".

O sea, SAS crea un nuevo data set que contiene todas las variables de ambos conjuntos de datos y tantas observaciones como la suma de la cantidad de observaciones de ambos archivos, rellenando con "." la información que no figura en ellos.

Ejemplo III:

VAR Obs	A	B	COD
1	10	5	6
2	15	8	6
3	20	11	6
4	25	14	6

VAR Obs	A	X	COD
1	10	6.3	7
2	25	2.7	7
3	40	6.8	8

### CONSEJO FEDERAL DE INVERSIONES

Si se ejecutara el siguiente programa:

```
DATA TERCERO;  
SET PRIM1 PRIM2;
```

—  
—  
—

Se formaría el data TERCERO de la siguiente forma:

VAR Obs	A	B	COO	X
1	10	5	6	.
2	15	8	6	.
3	20	11	6	.
4	25	14	6	.
5	10	.	7	6 3
6	25	.	7	2.7
7	40	.	8	6.8

### 3. Crear un subarchivo de otro ya existente

Por medio de instrucciones adicionales, por ej. IF, podemos seleccionar las observaciones que queremos que sean incluidas en un nuevo archivo.

```
Por ej. DATA PREMIO;  
SET EMPLEADO;  
IF ANTIG > 20;
```

—  
—  
—  
—

CONSEJO FEDERAL DE INVERSIONES

Siendo: EMPLEADØ

VAR Obs	NOMBRE	LEGAJØ	ANTIG
1	PEREZ	0207	22
2	LØPEZ	1211	5
3	SØLER	0130	26
4	SMITH	1156	3
5	JØNES	0135	24

Quedaría:

PREMIØ

VAR Obs	NOMBRE	LEGAJØ	ANTIG
1	PEREZ	0207	22
2	SØLER	0130	26
3	JØNES	0135	24

El valor de la variable ANTIG es comparada, observación a observación, contra 20, quedando incluidas en el nuevo archivo sólo aquellas para las cuales es verdadera la condición, o sea los individuos con mas de 20 años de antigüedad.

- 4. Crear un nuevo archivo agregando variables por medio de sentencias de asignación y/o aritméticas.

Spongamos que en el ejemplo de 1) queremos agregar al archivo VIEJØ una variable  $PRØM = (A+Y)/2$ .

La nueva secuencia de instrucciones sería:

```
DATA  NUEVØ;
SET   VIEJØ;
PRØM = (A+Y)/2;
```

—  
—  
—

**CONSEJO FEDERAL DE INVERSIONES**

Con los valores de A e Y de cada una de las observaciones se evalúa la variable PROM, que es incluida en la línea correspondiente.

El archivo NUEVO estaría formado:

VAR Obs	A	B	Y	PROM
1	0.5	1	6.5	3.5
2	0.8	1	7.9	4.35
3	0.7	2	6.9	3.8
4	1.3	3	5.8	3.55

5. Intercalar las observaciones de dos o más archivos ya existentes, ordenadas de acuerdo a los valores de una determinada variable.

Todos los archivos deben estar ordenados siguiendo la variable elegida. En este caso SAS toma las observaciones en forma ordenada, y en caso de existir dos con el mismo valor colocará primero la del primer archivo de la lista, luego la del segundo, y así sucesivamente.

Sean los data sets PRIM1 y PRIM2 como en el ejemplo 2).

Procesados según la siguiente secuencia:

```
DATA ORDENADO;  
SET PRIM1 PRIM2;  
BY A;
```

—  
—  
—

CONSEJO FEDERAL DE INVERSIONES

Obtendríamos:

VAR Obs	A	B	CØD	X
1	10	5	6	.
2	10	.	7	6.3
3	15	8	6	.
4	20	11	6	.
5	25	14	6	.
6	25	.	7	2.7
7	40	.	8	6.8

Estos distintos usos de la sentencia SET pueden ser combinados y completados con otras instrucciones de modo de adecuar la información para su posterior análisis.

Reportes Escritos

A menudo se quiere imprimir información de un archivo ya creado.

Ejemplo:

```
DATA_NULL ;
SET ESTUD ;
IF CREDITOS>100;
PUT @ 10 NOMBRE @ 45 CREDITOS;
```

El archivo ESTUD tiene la información de todos los estudiantes, solamente se quiere imprimir la información de aquellos que tienen crédito mayor de 100, este procedimiento no crea un nuevo archivo.

**CONSEJO FEDERAL DE INVERSIONES**

Sentencia MERGE

Se puede utilizar para combinar observaciones de dos o más archivos SAS.  
Hasta 50 archivos pueden aparecer en una sentencia MERGE.

Esta sentencia combina observaciones, no archivos.

Se utiliza para:

- Combinar Observaciones una por una.
- Mezclando con una sentencia BY.
- Múltiples Observaciones con los mismos valores BY.

Forma general:

MERGE (archivo 1 .....archivo n);

Combinando Observaciones una por una

Se utiliza para unir observaciones de dos o más archivos SAS conteniendo el mismo número de observaciones y en el mismo orden.

Ejemplo:

DATA A;

MERGE B C;

—

— opcional programa

—

—

Se unirá la primera observación del archivo B y la primera observación del C dando como resultado la primera observación del A y así siguiendo.

### CONSEJO FEDERAL DE INVERSIONES

Si uno de los archivos tiene más observaciones que el otro, el mayor tendrá variables con valor missing.

Ejemplo: Supongamos un archivo A con:

VAR OBS	NOMBRE	EDAD	ALTURA
1	JUAN	15	160
2	LUIS	18	170
3	RAUL	25	165

Un archivo B con:

VAR OBS	NOMBRE	EDAD	PESO
1	JUAN	16	60
2	LUIS	17	65
3	RAUL	25	70
4	ROQUE	15	58

Las sentencias:

DATA NUEVO;

MERGE A B ;

—  
—  
—  
—

Produce como resultado:



**CONSEJO FEDERAL DE INVERSIONES**

VAR OBS	NOMBRE	EDAD	ALTURA	PESO
1	JUAN	16	160	60
2	LUIS	17	170	65
3	RAUL	25	165	70
4	ROQUE	15	.	58

En este caso vemos que la variable EDAD se ha modificado. Con los datos del archivo B, que la variable PESO se agregó y la variable ALTURA queda igual salvo la cuarta observación donde es missing pues en el primer archivo no existe, la variable NOMBRE tiene las cuatro observaciones del segundo archivo.

Mezclando con una sentencia BY

Para igualar y combinar observaciones en dos o más archivos, se puede usar la sentencia BY.

Forma:

```
DATA A;
MERGE B C D E;
BY HZ;
```

```
— } programa opcional
— }
— }
— }
```

Ejemplo:

Supongamos un archivo A con:

VAR OBS	NOMBRE	EDAD
1	JUAN	15
2	LUIS	18
3	RAUL	25

CONSEJO FEDERAL DE INVERSIONES

Otro archivo B con:

VAR Obs	NOMBRE	ALTURA
1	JUAN	150
2	LUIS	180
3	RAUL	170
4	ROQUE	165

Las sentencias:

```
DATA NUEVO ;  
MERGE A B ;  
BY NOMBRE ;  
==
```

Produce un archivo NUEVO:

VAR Obs	NOMBRE	EDAD	ALTURA
1	JUAN	15	150
2	LUIS	18	180
3	RAUL	25	170
4	ROQUE	.	165

- Múltiples observaciones con los mismos valores BY

Ejemplo:

Supongamos un archivo A con:

VAR OBS	NOMBRE	EDAD	ALTURA
1	JUAN	15	160
2	LUIS	16	165

## CONSEJO FEDERAL DE INVERSIONES

Supongamos un archivo B con:

VAR OBS	NOMBRE	PESØ
1	JUAN	60
2	JUAN	65
3	LUIS	58
4	LUIS	64

Las sentencias:

```
DATA INFO;  
MERGE A B ;  
BY NOMBRE;  
=
```

produce un archivo:

VAR OBS	NOMBRE	EDAD	ALTURA	PESØ
1	JUAN	15	160	60
2	JUAN	15	160	65
3	LUIS	16	165	58
4	LUIS	16	165	64

### Sentencia UPDATE

Esta sentencia permite modificar un archivo con nueva información.

Se puede utilizar para:

- Agregar nueva información
- Cambiar los valores de las variables
- Agregar nuevas variables
- Condicionalmente cambiar el valor de las variables

CONSEJO FEDERAL DE INVERSIONES

Para utilizar la sentencia UPDATE, ambos archivos deben contener la misma identificación de las variables (igual nombre y tipo).

Debemos tener:

- . un viejo archivo - No puede contener identificadores repetidos.
- . un nuevo archivo - Es el nuevo archivo creado por el UPDATE.
- . un nuevo UPDATE - Es el que trae las modificaciones - puede tener identificadores duplicados.

Forma general: UPDATE (viejo archivo archivo de modificación);

Ejemplo:

```
(a) * CREACION DE UN VIEJO ARCHIVO;  
DATA VIEJO ;  
INPUT (sentencia);  
CARDS;  
  
—  
— tarjetas de datos clasificados  
—  
  
(b) * CREACION DE ARCHIVO UPDATE;  
DATA MODIF;  
INPUT (sentencia);  
CARDS;  
  
—  
— datos clasificados  
—  
  
(c) * CREACION DEL NUEVO ARCHIVO;  
DATA NUEVO  
UPDATE VIEJO MODIF;  
BY (identificación variables);  
  
—  
— sentencias opcionales de programa  
—
```

**CONSEJO FEDERAL DE INVERSIONES**

Si (a) y (b) ya fueron creadas en pasos anteriores solo se utilizan las sentencias del programa parte (c).

Si se utiliza la sentencia BY los archivos deben estar clasificados.

Agregar nuevas observaciones (ALTAS)

Si se quiere agregar nuevas observaciones al viejo archivo, el archivo de modificación deberá tener todas las variables del viejo archivo.

Ejemplo:

Supongamos un archivo VIEJØ;

<u>LICENCIA</u>	<u>MARCA</u>	<u>LUGAR</u>
AA-11	FØRD	CAPITAL
CC-33	DØDGE	CASTELAR

Supongamos un archivo para agregar llamado MØDIF con la siguiente información:

<u>LICENCIA</u>	<u>MARCA</u>	<u>LUGAR</u>
BB-22	CHEVROLET	LINIERS
DD-44	RENAULT	AVELLANEDA

Supongamos las siguientes sentencias:

DATA NUEVØ ;  
UPDATE VIEJØ MØDIF;  
BY LICENCIA;

—  
—  
—  
—  
—

CONSEJO FEDERAL DE INVERSIONES

PRØDUCE:

<u>LICENCIA</u>	<u>MARCA</u>	<u>LUGAR</u>
AA-11	FØRD	CAPITAL
BB-22	CHEVROLET	LINIERS
CC-33	DØDGE	CASTELAR
DD-44	RENAULT	AVELLANEDA

Cambiando el valor de las variables (MODIFICACIONES)

Para cambiar el valor de ciertas variables supongamos un archivo VIEJØ:

<u>LICENCIA</u>	<u>MARCA</u>	<u>LUGAR</u>	<u>KM</u>
AA-11	FØRD	LANUS	950
BB-22	CHEVY	CASTELAR	6500
CC-33	DOBGE	CAPITAL	23500

Archivo UPDATE UPS;

<u>LICENCIA</u>	<u>LUGAR</u>	<u>KM</u>
AA-11		1108
BB-22	ADROGUE	7654

Como no hay cambios en la variable MARCA no aparece para ser modificada.

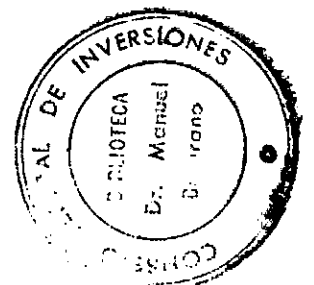
Las sentencias serfan:

DATA NUEVØ

UPDATE VIEJØ UPS;

BY LICENCIA;

—  
—  
—  
—  
—



**CONSEJO FEDERAL DE INVERSIONES**

Produce

<u>LICENCIA</u>	<u>MARCA</u>	<u>LUGAR</u>	<u>KM</u>
AA-11	FORD	LANUS	1108
BB-22	CHEVY	ADROGUE	7654
CC-33	DODGE	CAPITAL	23500

Multiples Observaciones en un update

El update puede contener múltiples observaciones:

Ejemplo: Archivo VIEJØ

<u>LICENCIA</u>	<u>MARCA</u>	<u>LUGAR</u>	<u>KM</u>
AA-11	FØRD	LANUS	950
BB-22	CHEVY	CASTELAR	6500
CC-33	DØDGE	CAPITAL	23500

Archivo UPS;

<u>LICENCIA</u>	<u>LUGAR</u>	<u>KM</u>
BB-22	ADRØGUE	7654
BB-22		8300

Dos observaciones de este archivo corresponde a una del archivo viejo.

Las sentencias:

DATA NUEVØ;

UPDATE VIEJØ UPS;

BY LICENCIA;

—  
—  
—  
—  
—

### CONSEJO FEDERAL DE INVERSIONES

Produce un archivo NUEVO

<u>LICENCIA</u>	<u>MARCA</u>	<u>LUGAR</u>	<u>KM</u>
AA-11	FØRD	LANUS	950
BB-22	CHEVY	ADRØGUE	8300
CC-33	DØDGE	CAPITAL	23500

#### Sumando nuevas variables

Para agregar nuevas variables al viejo archivo, esas nuevas variables aparecen en el UPDATE.

Archivo VIEJO contiene:

<u>LICENCIA</u>	<u>MARCA</u>	<u>LUGAR</u>
AA-11	FØRD	CASTELAR
BB-22	CHEVY	LINIERS
CC-33	DØDGE	CAPITAL

Queremos agregar una nueva variable MODELO entonces el archivo UPS contiene:

<u>LICENCIA</u>	<u>MODELO</u>
AA-11	CØUPE
BB-22	FAMILIAR
CC-33	RURAL

Las sentencias:

```
DATA NUEVO;  
UPDATE VIEJO UPS;  
BY LICENCIA;
```

—  
—  
—  
—  
—



## CONSEJO FEDERAL DE INVERSIONES

Produce estas observaciones en el NUEVO archivo.

<u>LICENCIA</u>	<u>MARCA</u>	<u>LUGAR</u>	<u>MODELO</u>
AA-11	FØRD	CASTELAR	CØUPE
BB-22	CHEVY	LINIERS	FAMILIAR
CC-33	DØDGE	CAPITAL	RURAL

Eliminando observaciones (Bajas)

Utilizando sentencias de programas en un UPDATE para DELETEDAR observaciones.

Supongamos las sentencias:

```
DATA NUEVO ;
```

```
UPDATE VIEJO UPS;
```

```
BY LICENCIA;
```

```
IF MARCA = 'FØRD' THEN DELETE;
```

Elimina todas las observaciones cuya marca es FØRD.

Valores Missing

Valores missing en un archivo UPDATE hace que el archivo viejo en esos valores no se altere.

Si queremos transformar el valor de una variable vieja en missing debe colocarse para variables numéricas el valor missing '\_' subguión produciendo como salida un '.', para variables alfanuméricas el subguión '\_' produciendo como salida blanco.

Supongamos Archivo VIEJO:

<u>LICENCIA</u>	<u>MARCA</u>	<u>KM</u>
AA-11	FØRD	950
BB-22	CHEVROLET	6500
CC-33	DØDGE	23500



**CONSEJO FEDERAL DE INVERSIONES**

**CAPITULO V**

**INSTRUCCIONES DE ACCION**

**CONSEJO FEDERAL DE INVERSIONES**

Cuando estamos creando un archivo SAS podemos necesitar hacer una acción sobre los datos, tanto en variables como en observaciones o individuos, para lo cual encontramos las siguientes instrucciones.

Suma: Para acumular totales o subtotales.

IF: Para la creación de un nuevo archivo seleccionando observaciones.

DELETE: Para eliminar observaciones del nuevo archivo que se está creando.

OUTPUT: Para creación de observaciones.

STOP: Para no agregar más observaciones en el nuevo archivo que se está creando.

RETAIN: Retiene un valor de variable anterior a través de las observaciones.

LIST: Para listar líneas de entrada.

Instrucciones de Asignación: Se utilizan para crear nuevas variables y/o modificar las existentes.

Nuevas variables pueden ser creadas combinando en una expresión una o más variables existentes y asignando el valor de la expresión a la nueva variable.

forma general:

$$\text{variable} = \text{expresión};$$

Ejemplo:

```

DATA UNO;
INPUT A B;
X = A + B;
A = A + 5;
CARDS;
1 3
2 4
3 2
-
-
-

```

OBS \ VAR	A	B	X
	1	6	3
2	7	4	6
3	8	2	5

**CONSEJO FEDERAL DE INVERSIONES**

En este caso X es una nueva variable, cuyo valor para cada observación será la suma del valor de A y B respectivamente.

Variable: Puede ser nueva o una variable ya existente. Será de tipo numérica o alfanumérica de acuerdo al resultado de la expresión.

Expresión: Es una secuencia de nombres de variables, constantes y/o nombres de funciones unidas por operadores y cuando corresponde por paréntesis.

Ejemplos:

A = X + 1;

B = 3;

C = 1 - EXP (N/(N-1)); ← C = 1 - e<sup>N/(N-1)</sup>

D = 'CFI';

Num.

"

"

Alfanum.

Instrucciones de suma automática. Estas instrucciones de suma permiten acumular valores de variables a través de las observaciones.

Se utilizan para totales, subtotales, contadores y acumuladores.

Forma general:

variable + expresión ;

La expresión puede ser lógica o aritmética.

Cada vez que la instrucción es ejecutada el valor de la expresión es acumulado en la variable.

CONSEJO FEDERAL DE INVERSIONES

```

DATA;
INPUT X Y Z ;
N + 1;
A + Y * Y;
M + X * N;
B = Y * Y;
TOT + Z;
CARDS;
1 2 4
. 1 3
5 8 1

```

OBS.	VAR							
	X	Y	Z	N	A	M	B	TOT
1	1	2	4	1	4	1	4	4
2	.	1	3	2	5	1	1	7
3	5	8	1	3	69	2	64	8

Las variables N y M son contadores, A es acumulador y TOT es un totalizador.

Funciones SAS, Matemáticas y Trigonométricas.

Forma general:

<.....> nombre función (argumento/s).

Los argumentos deben ser numéricos en las funciones matemáticas y trigonométricas, y si son variables también ellos deben ser numéricas.

Los argumentos de las funciones SAS pueden ser numéricas o alfanuméricas.

Algunas funciones requieren argumentos dentro de un cierto rango, pues fuera de dichos valores la función no está definida.

Si un valor no admisible o missing es colocado en el argumento SAS imprime un mensaje de error, y al resultado le asigna valor missing.

**CONSEJO FEDERAL DE INVERSIONES**

Funciones SAS:

LAG: Valores atrasados.

Forma: LAGn (argumento)

n puede variar de 1 a 100 o sea se puede utilizar LAG, LAG1, LAG2,....., LAG 100, es decir es factible utilizar hasta el valor atrasado 100 de las variables.

Ejemplo:

```
DATA UNO;
INPUT X A ;
Y = LAG (X); W = LAG (A);
Z = LAG2 (X); CARDS;
2 B
1 C
2 D
```

OBS \ VAR	X	A	Y	W	Z
1	2	B	.	.	.
2	1	C	2	B	.
3	2	D	1	C	2

DIF: Diferencias.

Forma: DIFn(argumento)

El argumento debe ser siempre numérico.

n puede variar de 1 a 100, o sea se puede utilizar de DIF, DIF1, ....., DIF 100.

la función DIFn es definida como:

$$DIFn (X) = X - LAG n(X)$$

Ejemplo:

```
DATA DOS;
INPUT X;
Z = LAG (X) ;
D = DIF (X);
CARDS;
1
2
6
4
7
```

OBS \ VAR	X	Z	D
1	1	.	.
2	2	1	1
3	6	2	4
4	4	6	-2
5	7	4	3

CONSEJO FEDERAL DE INVERSIONES

Funciones Matemáticas:

ABS (argumento)	valor absoluto
CEIL (argumento)	entero $\geq$ argumento
FLOOR (argumento)	entero $\leq$ argumento
INT (argumento)	Valor entero o truncación
MIN (lista de argumentos)	el menor valor
MAX (lista de argumentos)	el mayor valor
SQRT (argumento)	raíz cuadrada. ( $\sqrt{\quad}$ )
EXP (argumento)	e <sup>argumento</sup> e $\approx 2.71828$
LOG (argumento)	Logaritmo natural
LOG 10 (argumento)	Logaritmo decimal
LOG 2 (argumento)	Logaritmo en base 2.
SIGN (X)	signo siendo el resultado
	- 1 si X $\leq$ 0
	0 si X = 0
	1 si X $>$ 0

Funciones trigonométricas e hiperbólicas

ARCOS (argumento)	arcoseno ó cos <sup>-1</sup>	} argumento entre (-1 y + 1)
ARSIN (argumento)	arcoseno ó sen <sup>-1</sup>	
ÁTAN (argumento)	arcotangente	
COS (argumento)	coseno	
SIN (argumento)	seno	
TAN (argumento)	tangente	
COSH (argumento)	coseno hiperbólico	
SINH (argumento)	seno hiperbólico	
TANH (argumento)	tangente hiperbólico	

Todos los resultados están dados en radianes.

Funciones que dan estadísticas simples

Hay 15 funciones que nos permiten obtener estadísticas univariadas.



**CONSEJO FEDERAL DE INVERSIONES**

Forma:

Función (lista de argumentos)

o

Función (OF nombre de variables)

Estas funciones nos permiten hacer las estadísticas por fila entre variables.

Los valores missing no son tenidos en cuenta.

- N.....Nº argumentos ≠ missing
- NMISS.....Nº valores missing
- SUM.....Suma de los valores
- MEAN.....Media
- MIN.....Valor mínimo
- MAX.....Valor máximo
- RANGE.....Rango
- STD.....Desviación standard
- VAR.....Variancia
- CSS.....Suma de cuadrados corregida
- USS.....Suma de cuadrados no corregida
- CV.....Coeficiente de variación
- STDERR.....Error standard de la media
- SKEWNESS.....Skewness
- KURTØSIS.....Kurtosis

Si los argumentos son constantes deben ser separados por comas.

Ejemplos:

SUM (ØF X1- X12) dará la suma de los valores de X1 a X12.

X = RANGE (2,6, 3) ⇒ X = 4.

## CONSEJO FEDERAL DE INVERSIONES

### Instrucción DELETE:

Esta sentencia permite evitar la inclusión de observaciones en el archivo que estamos creando.

Ejemplo:

```
DATA TRES;
```

```
SET LOCAL;
```

```
IF LØC= 10 THEN DELETE;
```

```
-
```

```
-
```

El nuevo DATA TRES contendrá aquellas observaciones del LOCAL que tengan un valor de la variable LØC distinto de 10.

Supongamos el siguiente archivo ya creado

OBS \ VAR	CØDIGØ	AGR	GAN
1	15	905	250
2	17	0	540
3	25	100	0
4	69	70	110
5	108	20	50

creamos un subarchivo con las siguientes instrucciones:

```
DATA SEGUNDØ;
```

```
SET PRIMERØ;
```

```
SUPTØT = AGR + GAN;
```

```
IF SUPTØT < 150 THEN DELETE;
```

```
=
```

Obtendríamos:

OBS \ VAR	CØDIGØ	AGR	GAN	SUPTØT
1	15	905	250	1155
2	17	0	540	540
3	69	70	110	180

Instrucción OUTPUT:

Esta sentencia trabaja a la inversa del DELETE, o sea, nos permite indicar a SAS cuándo y cuáles observaciones deben ser incluidas en el archivo.

Esta instrucción podemos usarla en distintas situaciones.

- 1 - Para crear más de una observación para cada línea de entrada.
- 2 - Para crear más de un archivo con la misma entrada de datos.
- 3 - Para combinar varias líneas de entrada en una sola observación.
- 4 - Para crear nuevas variables con valores de otras ya existentes.

Forma de uso: OUTPUT;

Si estamos creando sólo un archivo bastará con escribirlo de esa forma.

Si estuviéramos creando dos o más archivos será necesario colocar:

OUTPUT lista - nombres;

para indicar en cuál o cuáles de los archivos debe ser incluida la observación. En caso de omitirlo será incluido en todos.

Hasta ahora no hemos colocado en ningún momento esta sentencia, pues al omitirla, se ejecuta automáticamente luego de todas las instrucciones de programa.

Por ejemplo:

DATA OBSERVA;

INPUT LOC TRAT REND1 REND2 REND3;

TOTAL =REND1 + REND2 + REND3;

..... puede omitirse) → OUTPUT;

CARDS;

10	15	105.8	110.6	98.0
16	17	100.0	120.8	106.9
8	25	96.5	97.1	98.4

-

- procedimientos

-

## CONSEJO FEDERAL DE INVERSIONES

El archivo quedaría formando de la siguiente manera:

VAR OBS	LØC	TRAT	REND1	REND2	REND3	TØTAL
1	10	15	105.8	110.6	98.0	314.4
2	16	17	100.0	120.8	106.9	327.7
3	8	25	96.5	97.1	98.4	292.0

Por cada línea que lee forma una observación.

i) Formar varias observaciones de cada una.

Cada variable RENDi corresponde a un bloque del tratamiento, para poder hacer un análisis debemos tener el número de bloque y el rendimiento en cada observación y bajo el mismo nombre de variable.

```
DATA, NUEVØ;  
SET ØBSERV; DRØP REND1 - REND3 TØTAL;  
BL=1; REND = REND1; ØUTPUT;  
BL=2; REND = REND2; ØUTPUT;  
BL=3; REND = REND3; ØUTPUT;  
- } procedimientos  
- }  
- }
```

Por cada observación de ØBSERV se ejecutan las instrucciones anteriores; toma el valor de LØC, el de TRAT, pone BL=1, REND= REND1 y al ejecutar el ØUTPUT quedó formada la primera observación. Ejecuta la segunda línea (los valores de LØC y TRAT no han variado) y forma la segunda observación y así con la tercera línea, recién entonces busca la segunda línea del archivo ØBSERV.

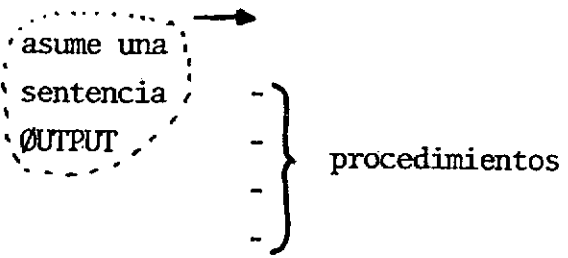
**CONSEJO FEDERAL DE INVERSIONES**

El contenido del data NUEVO:

OBS \ VAR	LØC	TRAT	BL	REND
1	10	15	1	105.8
2	10	15	2	110.6
3	10	15	3	314.4
4	16	17	1	100.0
5	16	17	2	120.8
6	16	17	3	106.9
7	8	25	1	96.5
8	8	25	2	97.1
9	8	25	3	98.4

Qué pasaría si omitiéramos el ØUTPUT:

```
DATA MAL;
SET ØBSERV; DRØP REND1 -REND3 TØTAL;
BL = 1; REND = REND1;
BL = 2; REND = REND2;
BL = 3; REND = REND3;
```



Toma la primera observación del archivo ØBSERV, pone BL= 1 y REND=105.8, pasa a ejecutar la segunda línea y coloca BL=2, REND=110.6 y la tercera, BL=3, REND=98,0 y recién entonces ejecuta un ØUTPUT, de este modo obtendríamos el siguiente resultado (al asignar un valor a una variable se pierde el valor que contenía hasta ese momento);

CONSEJO FEDERAL DE INVERSIONES

OBS	VAR			
	LØC	TRAT	BL	REND
1	10	15	3	98.0
2	16	17	3	106.9
3	8	25	3	98.4

2) Crear más de un archivo con el mismo conjunto de datos de entrada.

Supongamos que tenemos información sobre tres cultivos distintos en un sólo archivo.

Los códigos de los cultivos son: 01 - trigo  
02 - centeno  
03 - cebada

Para el análisis es necesario tener la información agrupada por cultivo

```
DATA TRIGØ CENTENØ CEBADA;  
INPUT CULT LØC MV ALT;  
IF CULT= 1 THEN ØUTPUT TRIGØ;  
IF CULT=2 THEN ØUTPUT CENTENO;  
IF CULT= 3 THEN ØUTPUT CEBADA;  
CARDS;
```

- 
- datos
- 
- procedimientos
- 

3) Combinar varias líneas de entrada en una observación. Sea NUEVO el archivo creado en el punto 1).

```
DATA ORIG;  
SET NUEVO ; RETAIN REND1 - REND3; DROP BL REND;  
IF BL = 1 THEN REND1 = REND;  
IF BL = 2 THEN REND2 = REND;  
IF BL = 3 THEN REND3 = REND;  
IF BL = 3 THEN OUTPUT;
```

procedimientos

El nuevo archivo quedaría formado como el OBSERV salvo por la variable TOTAL.

La sentencia RETAIN mantiene el valor de las variables REND1, REND2 y REND3 hasta que se ejecute el OUTPUT.

La cuarta opción se verá más detallada al explicar la sentencia RETAIN.

Instrucciones STOP:

Se utiliza para parar la entrada de observaciones a un archivo.

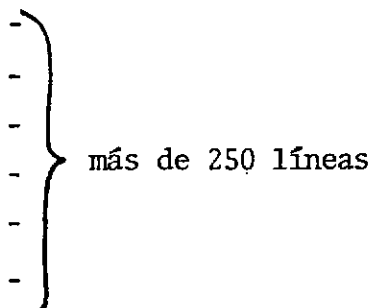
Forma:

```
STOP;
```

Esta instrucción no afecta la ejecución del resto del programa SAS. La ejecución continúa desde el primer DATA o PROC después de la instrucción STOP.

Ej:

```
DATA A;  
INPUT X Y Z;  
IF _N_ = 250 THEN STOP;  
CARDS;
```



```
PROC PRINT;
```

## CONSEJO FEDERAL DE INVERSIONES

El archivo A contendrá 249 observaciones pues en la 250 encontró el STOP, luego se listarán las 249 observaciones.

En este ejemplo se utilizó una variable automática N que es un contador de observaciones, no se define, se usa directamente con los subguiones.

### Instrucción RETAIN:

Esta sentencia nos permite retener valores, de las variables especificadas en la lista, y "arrastrarlos" de una observación a la siguiente, hasta que el mismo sea modificado por una sentencia aritmética o un INPUT.

Forma general: RETAIN lista-var valores-iniciales;

El primer valor que tomará cada una de las variables será el que figura en la sentencia RETAIN.

Ejemplo

	ORIGINAL		TOTAL	
	VAR	A	VAR	Suma
OBS			OBS	
DATA TOTAL;	1	6	1	6
SET ORIGINAL;	2	5	2	11
RETAIN SUMA 0;	3	3	3	14
SUMA = SUMA + A;	4	8	4	22

- procedimientos

O sea, inicializa SUMA en 0 y para cada observación del archivo ORIGINAL que toma acumula el nuevo valor de A.

Resultado: Como no tiene ninguna sentencia OUTPUT después de ejecutar la última sentencia y al encontrarse con las sentencias PROC, la ejecuta automáticamente, almacenando para cada nueva observación el valor actual de SUMA.



**CONSEJO FEDERAL DE INVERSIONES**

Recién en la última observación tendremos el total.

Si conocemos el último valor de A podríamos sólo almacenar el resultado final.

```
DATA TOTAL;
SET ORIGINAL; DROP A;
RETAIN SUMA 0;
SUMA = SUMA + A;
IF A=8 THEN OUTPUT;
PROC ....
```

TOTAL	
VAR	SUMA
OBS	
1	22

Instrucción LIST

Se utiliza para listar toda la observación que se está ejecutando en ese momento. Puede aparecer en cualquier parte del paso DATA y listará la observación leída con las variables construídas hasta la sentencia list.

Forma: LIST;

Ejemplo:

```
DATA A;
INPUT SSN 1-9 # W2 5-12 2;
IF W2 = THEN LIST;
CARDS;
-
- datos
-
-
-
```

Listará toda las observaciones donde el valor de la variable W2 sea missing.

Instrucción IF-THEN:

La sentencia IF se utiliza cuando se quiere saber el valor de una determinada variable para:

- Crear una nueva variable
- Transformar el valor de una variable ya existente
- Hacer un cálculo
- Eliminar una observación
- Leer determinados datos
- Imprimir
- Crear otros archivos
- Saltar instrucciones
- Volver al comienzo del programa
- Ir a hacer un determinado proceso y volver a la siguiente instrucción
- etc.

Forma general

IF expresión < THEN sentencia;

Ejemplo I:

IF AÑO = 1979 THEN COLOR = 'AZUL';

Cuando una observación es procesada, SAS evalúa la expresión siguiente al IF.

Si es verdadero para esa observación, la sentencia que sigue a THEN es ejecutada. Cuando la expresión es FALSA, SAS ignora la sentencia que sigue a THEN y procesa directamente la próxima instrucción del programa.

Las sentencias que pueden aparecer siguiendo a THEN son:

- INPUT
- Asignación
- Suma
- RETURN
- PUT
- DELETE
- OUTPUT
- STOP
- etc

CONSEJO FEDERAL DE INVERSIONES

Ejemplo II:

```
IF VALOR =. THEN DELETE;
IF PRØV = 'BS' ØR PRØV = 'CØ' THEN REGION = '1';
IF EDAD < 0 THEN EDAD = 0;
IF EXPLØT1 = 0 AND EXPLØT2 = 0 THEN SUPERF = SUPERF1 + SUPERF2 ;;
```

Ejemplo III:

```
DATA A;
INPUT X Y;
IF X=2 THEN Z=1;
IF X =3 THEN Z=X+Y;
CARDS;
```

-  
-  
-  
-

Ejemplo IV:

```
DATA A;
INPUT X Y;
IF X = THEN GØ TØ B;
X = 5
Y = 10 ; GØ TØ C;
B: Z= 15;
C; ;
CARDS;
```

-  
-  
-  
-

## CONSEJO FEDERAL DE INVERSIONES

Si el valor de X = . se ejecuta el `GO TO B` por lo tanto va a la instrucción que comienza con el rótulo B seguido de ":" y una expresión, en este caso asigna el valor de 15 a la variable Z.

Si X es distinto de missing ejecutará las instrucciones `X=5, Y=10` y luego salta a C.

La instrucción con rótulo C es una instrucción que no hace nada (continúa).

Ejemplo V:

```
DATA A; INPUT N @;  
IF N=1 THEN INPUT X Y ;  
CARDS;
```

Ejemplo VI:

```
DATA;  
INPUT X Y Z;  
IF X=4 THEN N+1;  
IF N=250 THEN STOP ;  
CARDS;
```

Cada vez que es ejecutada, N se incrementará en 1. Cuando se hayan leído 250 observaciones con X=4 el valor de N será 250.

**CONSEJO FEDERAL DE INVERSIONES**

La sentencia IF N= 250 THEN STØP le dice a SAS que pare de construir el nuevo archivo cuando N = 250.

Ejemplo VII:

```
DATA A B;  
INPUT EDAD X Y;  
IF EDAD < 15 THEN OUTPUT A;  
IF EDAD >= 15 THEN OUTPUT B;
```

En este caso SAS crea dos archivos el A con los datos de los individuos <15 años y el B con los datos de los individuos con más de 15 años.

Ejemplo VII; otra forma

```
DATA A; INPUT EDAD X Y ; IF EDAD < 15;  
DATA B; INPUT EDAD X Y ; IF EDAD >= 15;
```

**CONSEJO FEDERAL DE INVERSIONES**

**CAPITULO VI**

**INSTRUCCIONES DE INFORMACION Y DE CONTROL**

## CONSEJO FEDERAL DE INVERSIONES

### Instrucciones de información

El Sistema SAS tiene instrucciones que dan información extra para la creación del archivo en los pasos DATA, en algunos casos puede aparecer en el paso PROC.

Se desarrollaran los siguientes:

DROP	para excluir ciertas variables del archivo o análisis.
KEEP	para incluir ciertas variables en el archivo o análisis.
RENAME	para cambiar el nombre de las variables en el archivo.
LABEL	para dar rótulos descriptivos como nombre de variables a las distintas variables.

Instrucción: DROP: Permite eliminar las variables que no se quieren en el archivo o análisis.

Forma general:

DROP Lista de variables;

Ejemplo: Supongamos tener un archivo SAS llamado VIEJO con las variables A, B e Y

Aplicando:

```
DATA NUEVO;  
SET VIEJO;  
DROP A Y;  
  
PROM = (A + Y) / 2;  
-  
-  
-  
-  
-
```

CONSEJO FEDERAL DE INVERSIONES

ARCHIVO  
VIEJO

Var Obs	A	B	Y
1	3	1	4
2	2	1	1
3	1	2	5
4	3	3	2

ARCHIVO  
NUEVO

Var Obs	B	PROM.
1	1	3.5
2	1	1.5
3	2	3
4	3	2.5

Instrucción KEEP

Podríamos decir que la sentencia KEEP es la "inversa" de DROP.

La forma general es: KEEP lista-variables;

Las variables que figuran en la lista son las que sí serán incluidas en el archivo que se crea.

La secuencia:

```

DATA  NUEVO;
SET   VIEJO;
KEEP  B    PROM;
PROM = (A + Y )/2;
-
-
-
-

```

resulta equivalente a la que se presenta en el ejemplo del DROP.

El uso de DROP o KEEP se decide en base a la cantidad de variables que haya que colocar en una u otra lista.

Las variables pertenecientes a todos los archivos de la lista en la sentencia SET, pueden ser referenciadas durante la creación del nuevo data set, figuren o no en una instrucción KEEP o en una DROP.



**CONSEJO FEDERAL DE INVERSIONES**

El orden en que son almacenadas las variables en el nuevo archivo no depende del orden con que aparecen en una lista KEEP o DROP, sino en el orden en que fueron creadas.

Instrucción RENAME:

Se utilizar para cambiar el nombre a las distintas variables.

Forma General como instrucción:

RENAME viejo nombre = nuevo nombre .....

Ejemplo:

RENAME VIEJO = NUEVO X = Y;

Las variables que antes figuraban como VIEJO y X, ahora se llamarán NUEVO e Y.

Forma General como opción:

(RENAME = (Viejo nombre = Nuevo nombre .....

Ejemplo:

DATA A (RENAME = (X = X1982));  
SET B;

-  
-  
-  
-

Produce el mismo efecto que anteriormente. Como opción RENAME puede aparecer en una instrucción DATA o una PROC.

Si aparece junto con una DROP o KEEP, son aplicadas antes que el RENAME.

Instrucción LABEL

Se utiliza para rotular variables. Muchos procedimientos SAS usaran esos rotulos de las variables en la impresión de los resultados del análisis.

## CONSEJO FEDERAL DE INVERSIONES

Forma General:

```
LABEL variable      =      rótulo
-
-
-
-
-
-
-
.....;
```

Ejemplo:

```
LABEL ENC = ANALISIS ENCUESTA 1982;
```

Hasta 40 caracteres incluyendo los blancos de separación pueden ser usados como nombre asociado a la variable.

### Instrucciones de control

Dentro de un paso DATA, las sentencias SAS son ejecutadas una por una en forma secuencial para cada observación. Si se desea romper la secuencia para ciertas observaciones, o cambiar el orden de las instrucciones encontradas, se utilizarán este tipo de instrucciones.

Las instrucciones que transfieran el control desde una parte a otra del programa son:

IF - THEN/ELSE	Comparación
GØ TØ	Ruptura de secuencia (ir a un lugar determinado)
RETURN	Retornar a un lugar determinado
LINK - RETURN	Simil sub-routine
DO	Para ciclar
END	Detecta fin del ciclo DO

## CONSEJO FEDERAL DE INVERSIONES

### Instrucción IF - THEN/ELSE

Se utiliza para ejecutar una parte del programa si se cumple la condición especificada.

Forma General:

```
IF expresión THEN sentencia;  
ELSE sentencia;
```

La expresión es la de comparación.

Las sentencias que pueden aparecer después de un THEN o ELSE son:

INPUT, asignacion, suma, GØ - TØ, RETURN, INFILE, FILE,  
LINK, PUT, DELETE, OUTPUT, STOP, LIST, SET, MERGE, UPDATA, IF-- THEN ó DO.

Ejemplo:

```
DATA EN1; INPUT RES X Y W;  
IF RES = . THEN DELETE;  
IF RES = 1 THEN Z = 'PROP';  
ELSE Z = 'ARR';  
IF _ N _ = 50 THEN STOP;  
CARDS;
```

-  
-  
-  
-  
-

### Instrucción GO TO

Forma General GO TO rótulo ;

Esta instrucción tiene como función alterar el orden de ejecución de las instrucciones, transmitiendo el control a la sentencia identificada con ese rótulo o label.

## CONSEJO FEDERAL DE INVERSIONES

El rótulo permite hacer referencia a esa instrucción desde otro punto del programa.

La forma de colocar el rótulo es:

rótulo: sentencia;

donde 'rótulo' es un nombre SAS válido.

En general el uso del GO TO está ligado a la sentencia IF, que condiciona la bifurcación.

Ejemplo:

```
DATA DØS;  
INPUT A X Y;  
P1 = A * 100 / X;  
PS = Y * A / 100;
```

```
CARDS;
```

```
-  
- } datos  
-
```

Con esta secuencia estamos creando un archivo llamado DØS que contendrá cinco variables: A, X, Y, P1 y PS, tres de las cuales son leídas y dos son calculadas en base a las anteriores.

Supongamos que entre los datos puede darse X=0, en cuyo caso el cociente no se podrá efectuar

```
DATA DØS;  
INPUT A X Y;  
IF X=0 THEN GO TO label  
CERO;  
P1 = A * 100 / X;  
  
label  
CERO; PS = Y * A / 100;
```

CONSEJO FEDERAL DE INVERSIONES

CARDS;

- }  
- } datos  
- }

Instrucción RETURN

Se utiliza para evitar la ejecución de las siguientes instrucciones del programa, para la observación corriente.

Ejemplo I:

```
DATA;  
INPUT X Y Z ; ←.....  
IF X=Y THEN RETURN;.....  
X = Y + Z;  
A = X ** 2;  
  
CARDS;  
-  
-  
-  
-
```

Quando X es = a Y la sentencia RETURN es ejecutada. SAS agrega la observación al DATA SET y regresa a buscar una nueva observación. En este caso las dos sentencias siguientes al IF no son ejecutadas.

Si X es ≠ Y la sentencia RETURN no se ejecuta y se ejecutan las 2 sentencias siguientes al IF.

La sentencia RETURN es útil sobre todo utilizada con las sentencias GO TO y LINK - RETURN.

Ejemplo II:

```
DATA;  
INPUT A B C;  
IF A = 0 THEN GO TO YES_A;  
PUT A = ;  
ROOTA = SQRT (ABS (A));  
RETURN;  
----- YES_A : ROOTA = SQRT (A);
```

CONSEJO FEDERAL DE INVERSIONES

CARDS;

-  
-  
-  
-  
-

Si A es negativo RETURN hace que no se ejecute la sentencia siguiente a RETURN y vuelve a tomar otra observación.

Instrucción LINK y RETURN

La sentencia LINK dice a SAS que vaya a ejecutar las instrucciones con el rótulo que dice dicha instrucción. Cuando una sentencia RETURN es encontrada, SAS retorna a la sentencia siguiente a la LINK y continúa ejecutando desde ese punto.

Esto funciona en forma similar a una subrutina de FORTRAN:

Forma: LINK rótulo ;

Ejemplo:

```
DATA;
  INPUT  SSN 1-9 SEX $ HORAS MIN;
  IF SSN = 23866 THEN LINK EJE;
  HORAS = HORAS + (MIN/60);
  RETURN;

  EJE: SEX = 'F' ;
      PUT SSN =;
      RETURN;

CARDS;
-
-
-
-
```

## CONSEJO FEDERAL DE INVERSIONES

Las sentencias entre un rótulo y un RETURN puede ser cualquier sentencia SAS válida.

Pueden aparecer hasta 5 sentencias LINK dentro de un paso DATA y pueden ser anidadas.

### Instrucción DO y END

Esta instrucción se puede utilizar para dos situaciones diferentes.

- 1 - para ejecutar un grupo de instrucciones en un determinado momento.
- 2 - para una ejecución iterativa.

Caso 1 - La sentencia DO especifica que las instrucciones que están a continuación del DO serán ejecutadas hasta la sentencia END.

Forma General:

```
DO;  
=  
END;
```

La sentencia DO generalmente aparece asociado a una sentencia IF o ELSE.

Ejemplo:

```
-  
-  
-  
IF      X = Y      THEN DO;  
I = I + 1;  
PUT X;  
END;  
PUT Y;  
-  
-  
-  
-
```

Especifica que las instrucciones entre el DO y el END (grupo DO) se ejecutaron si  $X = Y$ .

**CONSEJO FEDERAL DE INVERSIONES**

Si X es  $\neq$  de Y, las instrucciones del grupo DO serán salteadas y se ejecutará la instrucción "PUT Y".

Caso 2: Ejecución iterativa de un grupo DO puede ser usada usando una variable índice en la sentencia DO.

Forma General:

DO variable índice = comienzo TO fin BY incremento;  
DO variable índice = comienzo TO fin;

Los valores de comienzo y fin deben ser números o expresiones que produzcan números. Cuando la sentencia DO tiene una de estas formas, las sentencias entre el DO y el END serán ejecutadas repetitivamente.

Ejemplo:

```
DO I = 2 TO 10 BY 2;  
-  
-  
-  
-  
-  
-  
END;  
-  
-  
-  
-  
-
```

I tomará los valores 2-4-6-8-10 o sea que iterará 5 veces desde la sentencia DO hasta la END, con I igual a cada uno de los valores antes mencionados.



CONSEJO FEDERAL DE INVERSIONES

Ejemplo:

```

DATA;
INPUT      X      Y Z;
IF      X =. THEN DO K=1 TO 5 ;
X = UNIFORM (0) * 100;
OUTPUT;
END;
CARDS;
-
-
-
-

```

\* Lee los valores de x, y y z.

- Si x = missing realiza un loop para K = 1a 25 variando de 1 (pues no existe BY), calcula los 25 valores de x como pseudo random y los guarda (o sea guarda 25 observaciones).

- Si x ≠ missing descarta la observación y va a leer otra.

Ejemplo:

```

DO I = 2 TO 10 BY 2, 11, 13, 15 TO 20;

```

Será ejecutado para I = 2, 4, 6, 8, 10, 11, 13, 15, 17, 18, 19 y 20

**CONSEJO FEDERAL DE INVERSIONES**

**CAPITULO VII**

**Instrucciones de impresión y Salida  
a archivos externos**

## CONSEJO FEDERAL DE INVERSIONES

Se puede usar las instrucciones FILE y PUT para producir reportes escritos, archivos O.S. en disco o cinta, o perforar tarjetas.

Estos reportes o archivos pueden tener cualquier formato que se requiera.

La instrucción FILE y PUT similar a INFILE e INPUT, son sentencias de programa SAS que se ejecutan por cada observación procesada por un paso DATA.

Elas aparecen sólo en un paso DATA, en el momento en que se quiere escribir un reparte, o se quiere un nuevo archivo O.S.

El efecto de ejecución de una sentencia FILE es definir el archivo en que en ese momento se va a volcar la información. La instrucción PUT que la sigue inmediatamente tiene la función de escribir los valores y/o constantes descriptas en la misma.

### INSTRUCCION FILE

La función de la sentencia FILE es definir el corriente archivo de salida.

Más de una sentencia FILE puede aparecer en un paso DATA, las sentencias PUT siempre escribe líneas en el corriente archivo de salida, definido por la más reciente sentencia FILE ejecutada.

Forma general.

FILE ← -PRINT  
          - - - - -PUNCH (opciones);  
          -nombre  
          -LOG

1) FILE PRINT (opciones);

Esta instrucción producirá la impresión de las líneas en el SAS PRINT.

**CONSEJO FEDERAL DE INVERSIONES**

II) FILE PUNCH (opciones);

Esta instrucción producirá las líneas de salida en tarjetas perforadas.

III) FILE LOG (opciones);

Esta instrucción imprimirá las líneas de salida en el SAS LOG.

En este caso el LOG puede ser omitido y escribir solamente FILE (opciones);

IV) FILE nombre (opción);

Especifica que el archivo será el definido anteriormente en una J.C.L. con ese mismo nombre, o sea puede referir a disco, etc.

Opciones:

En los cuatro casos antes mencionados se pueden utilizar las siguientes opciones

- HEADER ;cabezal

Forma General:

HEADER = rótulo

Se puede especificar un grupo de sentencias identificadas por el rótulo y que serán ejecutadas cuando salte a una nueva página.

Este grupo de sentencias debe tener:

- Un rótulo .
- Poner el HEADER=rótulo en la sentencia FILE .
- Poner una sentencia RETURN en el fin de este grupo de sentencias para que vuelva a la instrucción siguiente al FILE.

## CONSEJO FEDERAL DE INVERSIONES

Ejemplo:

```
DATA _NULL_ ;
SET MAESTRO; BY DEPT;
FILE PRINT HEADER= NEWPAGE;
PUT @22 DEPT @34 VENTAS;
RETURN;
NEWPAGE:
    PUT @20' VENTAS DE 1981' /
    @25 DEPT= ;
RETURN;
```

- NOTITLES: No coloca títulos  
forma general:

NOTITLES.

Ej: FILE PRINT NOTITLES;

No aparecerán en el SAS PRINT ninguno de los títulos dados por la sentencia TITLE del programa.

### Instrucción PUT:

Las instrucciones PUT e INPUT tienen las mismas componentes. Ellas difieren en que la sentencia INPUT lee valores de variables desde un archivo de entrada, y la sentencia PUT escribe valores de variables en un archivo de salida.

La sentencia PUT es una sentencia de programa, debe aparecer después de una sentencia DATA, usualmente una sentencia INPUT y/o SET seguirán a la sentencia DATA, y luego aparece la sentencia PUT.

Esta sentencia como otras sentencias de programa es ejecutada una vez para cada observación en el DATA SET que ha sido creado.

CONSEJO FEDERAL DE INVERSIONES

Ejemplo I:

```
DATA;  
INPUT X Y;  
PUT X= Y=;  
CARDS;  
22      5  
77      9  
44      7
```

Resultará una salida de 3 líneas

```
X = 22 Y = 5  
X = 77 Y = 9  
X = 44 Y = 7
```

Ejemplo II:

```
DATA A;  
INPUT X Y;  
CARDS;  
22      5  
77      9  
44      7  
DATA B; SET A;  
PUT X = Y = ;
```

Este ejemplo dará el mismo resultado que el anterior, pero podemos ver que en el ejemplo I la sentencia PUT aparece después de un INPUT, en cambio en el ejemplo II la sentencia PUT aparece después de una sentencia SET.

La forma en la cual SAS escribe la variable de un PUT depende del formato de la misma.

La sentencia PUT puede dar su salida por impresora, en tarjetas perforadas, en cinta o disco, etc.

## CONSEJO FEDERAL DE INVERSIONES

### Lista de Salida

Cuando una lista de nombres de variables aparecen después de un PUT, SAS escribe el valor actual de las variables dentro de la línea de salida, separando los ítems con un blanco.

Ejemplo:

```
DATA;  
X = 35.2 ; Y = 15.00;  
PUT X Y;  
Produce una línea:  
35.2 15
```

### Lista de Salida con Formatos

Cuando el nombre de una variable está seguida por ":" y un formato (:w.d), SAS convierte el dato para la cantidad de dígitos pedidos en el formato y elimina los blancos que están delante y detrás del dato.

Ejemplo:

```
DATA:  
X = 35.2; Y = 15.00;  
PUT X: 7.2;           6           PUT X 7.2.;
```

Produce una línea:

35.20

↓  
col1

Produce una línea:

35.20

↓  
col7

Nombres de Salida para variables

Cuando un signo "=" aparece después del nombre de una variable en una sentencia PUT, SAS escribe el nombre, un signo (=) y el valor actual de la línea de salida.





Los Pointers

La sentencia PUT utiliza los mismos indicadores que la sentencia INPUT.

# variable # k	} mueve a columna 1, línea k
@ variable @ n	} mueve a columna n
/	mueve a la próxima línea
+ n	mueve n columnas
_ PAGE _	mueve a la próxima página (este es solamente para PUT).
OVERPRINT	sobre impresión
@	retiene la línea de salida hasta la próxima sentencia PUT

\_ PAGE \_ : Usando esta palabra clave en la sentencia PUT. SAS automáticamente comienza una nueva página.

Ejemplo: PUT \_ PAGE \_ ;

OVERPRINT: Se usa para hacer sobre impresión.

Ejemplo:

PUT X OVERPRINT '\_' ;

Produce los valores de la variable X subrayados.

\_ ALL \_ : Imprime todos los valores de las variables con sus nombres respectivos.

Ejemplos: IF A THEN PUT \_ ALL \_ ;

## CONSEJO FEDERAL DE INVERSIONES

Imprime todos los valores de las distintas variables.

si A es  $\neq$  0 y de missing.

\_\_INFILE\_\_ : Escribe la línea leída más recientemente por una instrucción INPUT.

Ejemplo: DATA;  
INPUT A B;  
PUT \_\_INFILE\_\_;  
CARDS;  
-  
-  
- datos  
-  
-

### Lista de Formatos:

La lista de formatos puede ser usada en la sentencia PUT como en la sentencia INPUT.

Ejemplo: DATA;  
GRADØ1 = 27.5;  
GRADØ2 = 33.25;  
GRADØ3 = -1;  
PUT (GRADØ1 - GRADØ3) (7.2);  
  
produce una línea  
27.50 33.25 -1.00

CONSEJO FEDERAL DE INVERSIONES

```

Ejemplo:  DATA;
          GRADØ1  = 27.5;
          GRADØ2  = 33.25;
          GRADØ3  = -1;
          PESØ    = 88;
          PUT (PESO GRADØ1 - GRADØ3 (2. 5.1. +1.5.2. @ 16 2);

```

Produce una línea:

```

      88      27.5      33.25      -1
      |       |       |       |
      |       |       |       |
      |       |       |       |
col. 12      7      9 13      16 17

```

Constantes alfanuméricas

Las constantes alfanuméricas son caracteres entre apóstrofes, ellas pueden aparecer en una sentencia PUT delante de un nombre de una variable en la lista de formatos.

Cuando SAS encuentra una constante alfanumérica en una sentencia PUT, esa constante es impresa en la línea a la cual apunta el indicador.

Si los espacios en esa línea no alcanzan imprime hasta donde entra y lo restante lo coloca en la próxima línea.

Si la cantidad de campos enteros del dato es menor que el formato utilizado coloca \*\*\*\* en el campo de esa variable.

```

Ejemplo:  DATA;
          MØDELØ = 'DB';
          CØSTØ = 402.50;
          PUT 'EL COSTO DEL MODELO' MØDELØ $ 2. + 1 ' ES' / '$' CØSTØ 6.2;

```

Produce las líneas:

```

EL COSTO DEL MODELO DB ES
$ 402.50

```

## CONSEJO FEDERAL DE INVERSIONES

- Se puede usar "n\*" para repetición de una constante alfanumérica, el n debe ser un número entero positivo .

Ejemplo:      DATA;  
                PUT 2\*ESTADØ';  
                Produce una línea:  
                ESTADØ' ESTADØ'

**CONSEJO FEDERAL DE INVERSIONES**

**CAPITULO VIII**

**INSTRUCCIONES COMPLEMENTARIAS**

CONSEJO FEDERAL DE INVERSIONES

1) Existen instrucciones que son informativas como ser:

°Comentario (COMMENT)

°Título (TITLE)

2) Instrucciones que son de identificación de cortes de variables :

°BY

°Parámetro END.

3) Instrucciones para identificar subprogramas y para llamar al subprograma :

°MACRO

1- Instrucciones de información :

Comentario: Se utiliza para poner comentarios en un programa SAS, no es una instrucción ejecutable, solo aparecerá en el SAS LOG.

Forma general:

COMMENT mensaje;

ó \* mensaje;

El mensaje no tiene limitación en cuanto a su longitud.

Título

Por medio de esta instrucción podemos hacer que se imprima en la salida de cualquier procedimiento hasta 10 líneas de título.

Automáticamente SAS centraliza el título.

La limitación de cantidad de caracteres es de 132 (capacidad de la impre-

## CONSEJO FEDERAL DE INVERSIONES

sora).

La forma general es:

TITLE<sub>n</sub> título ;

donde n puede ser 1,2,3, ..., 10.

Para TITLE<sub>1</sub> se puede colocar TITLE.

Si varios TITLE<sub>n</sub> aparecen, son impresos en una misma hoja, quedando líneas en blanco si algún valor entre 1 y el máximo es omitido.

Cualquier sentencia TITLE<sub>j</sub> que aparece, anula anteriores sentencias TITLE<sub>j</sub> con valor i > j.

El título es repetido a través de todos los procedimientos del programa, salvo que un nuevo título aparezca.

Si un título debe incluir una comilla (') habrá que colocar doble comilla (").

Si debe incluir un punto y coma habrá que colocarlo entre comillas (';').

### 2. Instrucciones de Cortes :

BY : Detecta los cambios en la variable que se define a continuación.

Forma general BY nombre de variables <NOTSORTED> ;

Ejemplo BY SEXO ;

Realizará los cortes para los distintos valores de sexo.

Quando se utilizó una instrucción BY el archivo debe estar clasificado

**CONSEJO FEDERAL DE INVERSIONES**

para las variables asociadas a BY, en caso contrario debe colocarse NOTSORTED.

Ejemplo:

```
DATA ;  
INPUT SEXO $ PESO ;  
CARDS;
```

```
F 123  
F 185  
F 120  
M 219  
M 179  
M 180
```

```
PROC PRINT ;  
BY SEXO;
```

Obtendremos:

SEXO = F

Obs	Peso
1	123
2	185
3	120

SEXO = M

Obs	Peso
4	219
5	179
6	180



**CONSEJO FEDERAL DE INVERSIONES**

Parámetro END

Definimos una variable, por ej. FIN y le asignamos el valor de este parámetro automático que usa SAS.

Los valores que FIN puede tomar son:

- 1: cuando está construyendo LA ULTIMA observación para el nuevo archivo.
- y 0: en otro caso.

De este modo podemos controlar cuándo tenemos el total por medio de la variable FIN.

La sentencia IF FIN = 1 THEN sentencia; } detecta fin de archivo  
es equivalente a: IF FIN THEN sentencia;

La asignación del valor del parámetro END a la variable FIN la hacemos en la sentencia SET:

```
SET ORIGINAL END = FIN;
```

El ej. anterior quedaría:

```
DATA TOTAL; KEEP SUMA ;
SET ORIGINAL END = FIN;
RETAIN SUMA 0;
SUMA = SUMA + A;
IF FIN THEN OUTPUT;
PROC ....
```

TOTAL	
Obs	SUMA
1	22

-  
-  
-  
-

**CONSEJO FEDERAL DE INVERSIONES**

No necesitamos conocer el valor de A en la última observación.

Ejemplo 2:

```
DATA DEXX; SET RESIDCØV END=FIN;
RETAIN EXX 0; KEEP EXX;
EXX = EXX + RESCØV **2;
IF FIN THEN ØOUTPUT;
.....
```

RESIDCØV		
Obs	CØV	RESCØV
1	Ø	Ø
.		
.		
.		
52	Ø	Ø

DEXX

Obs	EXX
1	122.5

Ejemplo 3:

```
DATA CALCULØS; SET AUX END = FIN1;
RETAIN N 0 SUMX 0 SCTRAT 0;
KEEP SCTRAT PRØMCØV;
N = N + NFRUTØS;
SUMX = SUMX + SUMCØV;
SCTRAT = SCTRAT + (SUMFRUTØ * SUMCØV) / NFRUTØS;
IF FIN1 THEN PRØMCØV = SUMX/N;
IF FIN1 THEN OUTPUT;
-
-   proc
-
```

El archivo AUX contenía las variables: MFRUTØS MCØV NFRUTØS NCØV SUMFRUTØ SUMCØV, y 13 observaciones.

El resultado obtenido:

CALCULØS		
Obs	VAR	
	SCTRAT	PRØMCØV
1	123898	13.6154

3. Sub programas.

Instrucción MACRO

Cuando una parte del programa se repite en otras partes del programa, esa parte común puede ser escrita dentro de una instrucción MACRO y luego llamarla desde cualquier punto del programa.

Forma: MACRO nombre del Macro(instrucciones) %

Debe estar un MACRO siempre antes de donde se los llame.

Ejemplo MACRO CURSO IF SSN=N THEN DELETE;

A = B+C ; %

DATA A ; SET Z ; CURSO

DATA D ; SET Y ; CURSO

ABR 1982

La palabra CURSO aparece sin ";". SAS insertará las dos sentencias donde aparece la palabra CURSO.

El nombre del MACRO debe ser un nombre válido de SAS.

No hay límite en la longitud de la información de un MACRO. Cualquier número de sentencias MACRO puede aparecer en un programa SAS.

Si 2 macros tienen el mismo nombre, el segundo reemplaza al primero.

No se aconseja utilizar MACROS anidados.

Para llamar a un macro como se vió en el ejemplo anterior solo se escribe el nombre del macro y no se coloca ; .

El número máximo de MACROS aceptados en un programa SAS es 40.