
SISINF

INSTRUCCIONES GENERALES

Versión 7.3

Kratos, S.A. de C.V.

La Tecnología en Software.

Derechos Reservados ©. Prohibida la reproducción total o parcial sin permiso escrito de **KRATOS, S.A. de C.V.** El uso de programas que integran **SISINF** se vende y renta bajo contrato con **KRATOS, S.A. de C.V.**

CONTENIDO

CONTENIDO.....	2
PREFACIO.....	4
1) INSTRUCCIONES DE INICIO, FIN, ARITMETICA Y MOVER.....	5
1.1) Instrucción de Programa.....	5
1.2) Instrucción de Fin de Ejecución.....	10
1.3) Expresiones Aritméticas.....	12
1.4) Instrucción de Mover Información.....	14
1.5) Instrucción de Convertir.....	17
1.6) Instrucción para sacar Parámetros.....	18
1.7) Buscar en una tabla y mover un renglón.....	26
2) INSTRUCCIONES DE CONTROL.....	29
2.1) Etiquetas.....	29
2.2) Instrucción de comparación total.....	29
2.3) Instrucción de comparación parcial.....	32
2.4) Instrucción para ejecutar otra Instrucción.....	34
3) INSTRUCCIONES DE FECHAS.....	35
3.1) Sumar días a una fecha.....	35
3.2) Instrucción de restar dos fechas.....	36
4) EJECUTA PROGRAMA Y RUTINA.....	38
4.1) Instrucción para pedir la ejecución de otro PROGRAMA.....	38
4.2) Instrucción para pedir la ejecución de una RUTINA.....	39
4.3) Declaración de inicio de una RUTINA.....	40
4.4) Instrucción para regresar de una RUTINA.....	40
4.5) Ejemplo de uso de RUTINAS.....	41
5) TEXTOS.....	43
5.1) Formar Texto.....	43
5.1.1) Opciones de /INICIO y /AGREGAR.....	44
5.1.2) Cambiar las identaciones de DBD. Opciones DIP y DII.....	45
5.1.3) Información de Control de indentación y renglones en blanco.....	45
5.1.4) Uso de textos base.....	50
5.1.5) Pasar una variable texto a otra variable texto.....	52

SISINF (IG)	CONTENIDO
5.2) Ejemplo.....	52
6) DEFINICION DE IGUAL.....	55
6.1) Definición de IGUAL en DBD.....	55
6.2) Definición de IGUAL en CLS.....	56
7) VECTORES.....	59
7.1) Declaración de VECTOR.....	59
7.2) Subíndices.....	61
8) DOCUMENTACION SISINF.....	62
8.1) Registros con *D.....	62
8.2) DOCUMENTACION y Estadísticas de Uso.....	62
9) EJECUCION DE CLS.....	64
9.1) Pasos para la ejecución.....	64
9.2) Datos para la ejecución.....	65
9.3) Uso de Memoria.....	68
9.4) Reglas de Sintaxis.....	71
10) INSTRUCCIONES A CLS.....	73
10.1) Para que sirven?.....	73
10.2) Ejemplo.....	73
11) FUNCIONAMIENTO DE CLS.....	75
12) VALORES MAXIMOS DE CLS.....	77
13) FORMAR IMAGEN.....	78
13.1) Gráficas.....	78
13.2) Gráfica de Pastel.....	82
13.3) Gráfica de Barras.....	85
13.4) Gráfica de Barras varias series.....	88
13.5) Gráficas Marcadas.....	90
13.6) Imágenes marcadas.....	96
13.7) Figuras y Texto.....	101
13.8) Resumen de Figuras.....	115
14) ERRORES DE EJECUCION.....	117

PREFACIO

Las instrucciones del Lenguaje **SISINF** se pueden agrupar en instrucciones para el manejo del banco de datos, de archivos secuenciales, de terminal y de impresora. Además de estos grupos se tiene otro que incluye las instrucciones de inicio, fin, mover, control, para manejo de fechas, para manejo de textos, ... A este grupo se le ha llamado Instrucciones Generales y Declaraciones que son las que se describen en este manual.

También se describe el funcionamiento general de CLS.

Algunas de estas instrucciones son parecidas a las de otros lenguajes de programación por lo que se dan ejemplos aislados de su uso. Otras son particulares de **SISINF** por lo que su uso se describe con ejemplos en que se usan otras instrucciones, pero dando la explicación de las involucradas.

1) INSTRUCCIONES DE INICIO, FIN, ARITMETICA Y MOVER.

A continuación se describen las instrucciones para iniciar un programa (PROGRAMA), para finalizarlo (FIN), la instrucción aritmética y las de mover (MOVER, CONVERTIR, SACAR PARAMETROS y BUSCAR TABLA).

1.1) Instrucción de Programa.

Esta instrucción es la primera de un programa y define el letrero que aparece como nombre del programa en la terminal además de algunas opciones que se describen a continuación:

El formato general de la instrucción es:

```
PROGRAMA LET (/AREAS-COMUNES NOM-A1 NOM-A2 ...)  
(/PROCESO-DE-ERROR XYYY)  
(/EDICION-EN-INGLES)  
(/COLOR NOM-F NOM-L)  
(/USAR-MENOS-MEMORIA)  
(/RCM NUM-REN NUM-COL)
```

Las siguientes restricciones y comentarios son aplicables:

- ◆ Debe ser la primera instrucción de un programa.
- ◆ Debe no tener etiqueta.
- ◆ En un programa solo puede haber una sola instrucción de PROGRAMA.
- ◆ El orden en que aparecen las opciones puede ser cualquiera.
- ◆ Como el letrero del nombre del programa aparece en el primer renglón de la terminal junto con otra información, si este letrero es muy grande se trunca.
- ◆ Con la opción /AREAS-COMUNES se especifica en NOM-A1, NOM-A2, ... el nombre de las diferentes áreas comunes que tendrá el programa.
- ◆ Si sólo se tiene la opción de /AREAS-COMUNES esta se puede omitir

especificando sólo los nombres de las áreas.

- ◆ Con la opción **/PROCESO-DE-ERROR** se especifica en la literal XXYY el nombre del programa que procesará los errores y se activa la opción por el resto de la ejecución de este programa y los siguientes. En XX se especifican las iniciales de DBD y en YY las iniciales del programa. Al final de esta sección se explica con más detalles el uso de esta opción.
- ◆ La opción de **/EDICION-EN-INGLES** ocasiona que en la ejecución del programa, las máscaras M13 a M15 y M16 a M18 se trabajen desplegando o imprimiendo la información en inglés. Las fechas se leen de terminal o temporal externo como mes, día y año.
- ◆ Con la opción de **/COLOR** se puede especificar el color del fondo de toda la terminal en NOM-F y el color de las letras en NOM-L. Si no se especifica la opción se asume AZUL y BLANCO.
- ◆ En DBD se definieron las áreas con nombre NOM-A1, NOM-A2, ... así como las variables que las forman.
- ◆ A las variables que pertenecen a un área con nombre que ya se usó dentro de una misma ejecución de ELS, se le asignan el último valor que tuvieron dichas variables.
- ◆ En un mismo programa NO se permite que el mismo nombre de variable se use en dos áreas diferentes. Se diagnostica en compilación.
- ◆ Todas las variables de un área con nombre usarán memoria del programa.
- ◆ El orden en el que se especifiquen NOM-A1, NOM-A2, ... puede ser cualquiera.
- ◆ Aunque en V7 se dispone de más memoria para la ejecución de los programas puede suceder que al compilar un programa se dé error de exceder el límite máximo de memoria. Para que dichos programas se puedan compilar se puede poner **/USAR-MENOS-MEMORIA**.
- ◆ Cuando se usa la opción de **/USAR-MENOS-MEMORIA** no se podrá tener conectividad, OPCION DISCO DISCO usada más de una vez y las nuevas instrucciones de impresión.
- ◆ Con la opción **/RCM** se puede especificar el número de renglones y columnas máximos de las áreas de terminal.
- ◆ En NUN-REN se especifica el número de renglones máximos que tendrá el programa. Los valores permitidos son de 23 a 50.

- ◆ En NUN-COL se especifica el número de columnas máximos que tendrá el programa. Los valores permitidos son 80 a 200.
- ◆ El área de /ERROR si no se define o no tiene /POS. usará los 2 últimos renglones.
- ◆ Esta opción solo trabaja con el Emulador Sisinf, con ELS diagnostica error.
- ◆ Al inicio del Emulador Sisinf con el tamaño especificado (640-480, 800-600, 1024-768, Auto) se busca un tipo de letra aplicable a la resolución escogida y a una pantalla de Sisinf de 23 renglones por 80 columnas, con esto se fija el tamaño de la ventana que usará el emulador. Dicho tamaño permanecerá igual durante toda la ejecución de los diferentes programas.
- ◆ La opción de AUTO en el tamaño del Emulador Sisinf sirve para usar otras resoluciones diferentes a las listadas, lo anterior se debe a que dependiendo del hardware (tarjeta de video y monitor) es posible escoger de entre bastantes resoluciones.
- ◆ Cuando se ejecute un programa con la opción de /RCM, el Emulador Sisinf buscará un nuevo tipo de letra que con el tamaño actual de la ventana. Este tipo de letra será igual o mas pequeño que el tipo de letra inicial, es decir las letras serán mas pequeñas que en los programas de 23 por 80.

Con la estrategia seguida por el Emulador Sisinf puede suceder que el tipo de letra seleccionado sea muy pequeño y que por el hardware que se tenga sea ilegible. Se recomienda primero poner la opción y si las áreas actuales del programa con el tipo de letra seleccionado son legibles en los diferentes hardware que usen la aplicación, entonces hacer los cambios.

- ◆ Las variables tipo TEXTO se pueden ahora definir en áreas de mayor tamaño pero un renglón no puede ser mayor a 120, lo mismo si se usa la opción /LONG. en variables numéricas, alfanuméricas y fecha. En el caso de IMAGENES se permite cualquier valor para las opciones de /LONG. y /RENG. siempre que queden dentro del área de terminal.

A continuación se tiene un ejemplo del uso de esta instrucción cuando se definen y usan áreas con nombre. En DBD se definió:

```
AREA CIAS
NUM-CIA ** R0 S N1 0
NOM-CIA ** R0 S X 50
AREA CONS
COEF-1 ** R0 S N3 2
```

COEF-2 ** R0 S N3 2

En el programa PSCOME se puso:

PROGRAMA 'MENU' CIAS CONS

Se define NUM-CIA, NOM-CIA, COEF-1, COEF-2

En el programa PSCOA1 se puso:

PROGRAMA '...' CIAS

En programa PSCOB1 se puso:

PROGRAMA '...' CONS

En el programa PSCOD1 se pudo:

PROGRAMA '...' CONS CIAS

Si en la misma ejecución de ELS el orden en el que se ejecutan es:

- 1) PSCOME
- 2) PSCOA1
- 3) PSCOB1

4) PSCOD1

Entonces al ejecutarse PSCOME se definen las variables de las dos áreas, al ejecutarse PSCOA1 se usan los valores de CIAS definidos en PSCOME, al ejecutarse PSCOB1 los de CONS y por último al ejecutarse PSCOD1 se usarán los valores de CIAS de PSCOA1 y los valores de CONS de PSCOB1.

Los siguientes comentarios son aplicables a la ejecución de esta instrucción:

- ◆ En el módulo ELS primeramente la pantalla se borra y aparecerá en el primer renglón de la terminal, el nombre de la empresa, el nombre del programa, la hora y la palabra EJECUCION.
- ◆ En el módulo ELSI-Emulador el nombre de la ventana de ejecución sera el nombre del programa y el nombre de la empresa. El estado se desplegará en la parte inferior entre las teclas de función.
- ◆ A continuación y si no es el primer programa que se ejecuta, se cargan los valores de las diferentes variables que pertenecen a áreas con nombre que use este programa del archivo TEMxxyy.
- ◆ Si en algún área de terminal de tipo MENU se puso la opción /CALCULAR-NIVEL se ejecutará la rutina indicada para calcular el nivel de cada una de las opciones de menú.

A continuación se describe el uso de la opción de /PROCESO-DE-ERROR.

Esta opción sirve para que los errores de ejecución sean desplegados al usuario final en una forma más clara y que le permitan dadas las explicaciones tomar la acción más adecuada, para ello se incluye un código del tipo de acción a tomar por el usuario final y con esto de ser posible resuelva el problema.

La forma de trabajo de la opción se describe a continuación:

- ◆ Primero es necesario que se especifique el programa que procesará los errores y que se active la opción por el resto de la ejecución de éste y los siguientes programas. Para ello se deberá poner en la instrucción de PROGRAMA la opción /PROCESO-DE-ERROR y el nombre del programa.
- ◆ Si hay un error informativo, se almacena la información del mismo en tablas internas y al final de la ejecución del programa donde ocurre se pasa el control al programa para el proceso de errores. Si hay varios errores informativos solo se conocerá el último.
- ◆ Si hay error de cancelación, se almacena la información en tablas internas e inmediatamente se ejecuta el programa para procesar los errores.

- ◆ El programa para procesar los errores podrá mediante la instrucción de SACAR PARAMETROS conocer la información del error y desplegarla, imprimirla, ...
- ◆ Si ocurre un error del tipo informativo o de cancelación en el programa para procesar los errores se procederá en forma normal.
- ◆ Al final el programa que procesa los errores podrá continuar la ejecución de otro programa o terminar. Si el error fue de tipo informativo con SACAR PARAMETROS PROGRAMA-SIGUIENTE se conocerá cual es y se puede continuar la ejecución.

Para que el programador pueda usar esta opción en forma rápida, se tiene en el sistema de documentación el archivo ISERRD, mismo que se deberá de incluir en la aplicación y un programa prototipo para procesar los errores en PSDCER.

Así el programador deberá:

- ◆ Incluir en su DBD la instrucción &I ERRD.
- ◆ Copiar el programa PSDCER a PSxxyy donde se encuentren los programas de su aplicación.
- ◆ Incluir en el programa menú principal la opción /PROCESO-DE-ERROR 'xxyy' en la instrucción de PROGRAMA.

1.2) Instrucción de Fin de Ejecución.

Esta instrucción sirve para indicar que se termine la ejecución del programa y en forma opcional se ejecute otro.

El formato general de la instrucción es:

FIN ((VAR-A-P/LET-P) VAR-A-D/LET-D)

Los siguientes comentarios y restricciones son aplicables:

- ◆ Puede haber más de una instrucción de FIN en un programa, pudiendo estar en cualquier parte.
- ◆ Esta instrucción NO indica que ya no hay más instrucciones en el archivo PSxxyy.

- ◆ Si la última instrucción dentro del archivo PSxxyy NO es FIN o EJECUTA se genera internamente una instrucción de FIN.
- ◆ La siguiente instrucción después de ésta, deberá tener etiqueta, ya que de otra forma nunca se ejecutaría y se deberá hacer referencia a ella.
- ◆ En VAR-A-P o LET-P se puede poner las iniciales (xxyy) de otro programa en el Lenguaje **SISINF**, dicho programa se ejecutará a continuación.
- ◆ En VAR-A-D o LET-D se puede poner un nombre lógico o el directorio del dispositivo donde esta el programa. Si se omite será el mismo que del programa que se está ejecutando.
- ◆ Si VAR-A-D o LET-D son blancos dependerá del Sistema Operativo el dispositivo que tomará.
- ◆ La variable VAR-A-P que se especifique debe ser alfanumérica de 4 caracteres (definida como X 4), o bien un letrero (LET) que se especifique de 4 caracteres también.
- ◆ VAR-A-D o LET-A no deben ser mayores a 40 caracteres.

Para aclarar más el uso de esta instrucción se tiene el siguiente ejemplo:

```

_____
_____
_____
10  FIN
_____
_____
_____
20  FIN  'CBA1'
_____
_____
_____
30  FIN  'PRA3'
_____
_____
_____
40  FIN  PROG

```

En el primer FIN (etiqueta 10) se termina el proceso, en el segundo (etiqueta 20) se ejecuta el programa PSCBA1, en el tercero (etiqueta 30) se ejecuta el programa PSPRA3. Note que puede llamarse programas de diferentes DBD. En el último se llama al programa cuyas iniciales tenga en ese momento la variable PROG. Si PROG vale CBA4 se ejecuta PSCBA4.

Los siguientes comentarios son aplicables a la ejecución de esta instrucción.

- ◆ Primeramente se cierran todos los archivos que se usaron durante la ejecución del programa, esto incluye impresora, temporales y permanentes.
- ◆ Si el programa tiene la opción de Estadísticas de Uso (EST-USO) estas se actualizan en el archivo Esxxyy. Si el programa tiene áreas con nombre y se pide la ejecución de otro, el contenido de las diferentes variables de dichas áreas con nombre se guarda en el archivo TEMxxy para su posible uso en otro programa.
- ◆ Se actualiza el sistema de contabilidad de uso de **SISINF** para registrar la ejecución de este programa.
- ◆ Si se pidió la ejecución de otro programa, se inicia ésta.

1.3) Expresiones Aritméticas.

Esta instrucción sirve para calcular expresiones y almacenar su resultado en una variable.

El formato general de la instrucción es:

VAR-N = VAR-N/NUM OP VAR-N/NUM OP VAR-N/NUM ...

VAR-N =R VAR-N/NUM OP VAR-N/NUM OP VAR-N/NUM ...

En donde OP es cualquiera de + - * / **

Los siguientes ejemplos ilustran el uso de esta instrucción:

A = B * C / D - 8.63

AI = 6

Los siguientes comentarios y restricciones deben ser observados:

- ◆ La expresión se calcula de izquierda a derecha. Note que no hay

paréntesis.

Así, el ejemplo de:

$$A = B * C / D - 8.63$$

se efectuará de la siguiente forma:

- a) Se calcula $B * C$,
 - b) El resultado se divide entre D ,
 - c) Al resultado se le resta 8.63 ,
 - d) El resultado se almacena en A .
- ◆ El resultado de la expresión si se puso $=$ se almacenará en la variable de la izquierda de acuerdo a los decimales especificados, truncando los no usados. Si se puso $=R$ se redondea de acuerdo a los decimales de la variable y luego se almacena. Se redondea correctamente si el resultado es negativo.
 - ◆ Si se pierde precisión porque el resultado de la expresión no cabe en la variable que se almacena se diagnostica, salvo en los siguientes casos:

$VAR-N = VAR-N$

$VAR-N = NUM$

en que se trunca y **no** se diagnostica. Así, por ejemplo, en $A = B + 1$ se dará diagnóstico si se pierde precisión, en cambio en $A = B$ no se dará diagnóstico si se pierde precisión.

- ◆ Cuando se usa $*$ (multiplicación) o $/$ (división) la precisión de los decimales puede ser diferente en las diferentes marcas y modelos de computadoras en donde está implantado **SISINF**. Cuando se usa $**$ (exponenciación) los algoritmos y precisión de cada máquina son diferentes por lo que se pueden obtener diferencias en los dígitos menos significativos entre máquinas. Así $100 ** 0.5$ en algunas máquinas puede ser 10 , en otras 9.999 y en otras 10.001 .
- ◆ También por los algoritmos usados, $A * A$ y $A ** 2$ pueden dar resultados diferentes.
- ◆ En constantes numéricas el punto decimal es opcional si no hay decimales. Así los siguientes números generan el mismo resultado: 18 , $18.$, 18.00 .

- ◆ En la expresión se permiten variables numéricas del tipo N1, N2, N3 o N4 mezcladas, con cualquier número de decimales.
- ◆ La variable de la izquierda, donde se almacena el resultado puede ser N1, N2, N3 o N4 con cualquier número de decimales.
- ◆ En las expresiones aritméticas se permiten variables Fechas, las cuales se asumen iguales a N2 con cero decimales. Es responsabilidad del programador conservar la integridad de la variable fecha. Así por ejemplo, es válido poner:

FECHA = 20020101

1.4) Instrucción de Mover Información.

Se mueve información de un número, literal o variable a una variable alfanumérica. O bien se mueve información de una variable texto, imagen o audio a otra variable texto, imagen o audio.

El formato general de la instrucción es:

Formato 1:

MOVER NUM/LET/VAR **A** VAR-X
 ((/INICIO) o
 (/AGREGAR NUM1/VAR1-N) o
 (/POS. NUM2/VAR2-N))

Formato 2:

MOVER VAR-T/VAR-I/VAR-A **A** VAR-T/VAR-I/VAR-A

Formato 3:

MOVER VAR-TA VAR-X (VAR-N)
 (/CC VAR-XC (VAR-N))

Los siguientes comentarios son aplicables al Formato 1:

- ◆ La información a mover puede ser un número (NUM), una literal (LIT) o una variable (VAR) numérica, alfanumérica fecha o SI-NO.
- ◆ La variable receptora VAR-X tiene que ser alfanumérica.

- ◆ Cuando se especifica NUM o una variable (VAR) numérica o fecha, el campo se **edita** antes de mover, de forma que el resultado editado es la **información alfanumérica** a mover. Cuando se usa la opción de /INICIO se mueve la información alfanumérica a VAR-X desde el inicio. Si dicha información alfanumérica tiene una longitud menor que la variable receptora (VAR-X) se rellena a blancos, si es mayor se trunca y NO se diagnostica.
- ◆ Cuando se usa la opción de /AGREGAR NUM1/VAR1-N se busca en la variable receptora de derecha a izquierda por el primer caracter NO blanco y a partir de este caracter y dejando NUM1/VAR1-N blancos se mueve la información. En este caso si la longitud para almacenar la información es menor se trunca y NO se diagnostica.
- ◆ Si en la opción de /AGREGAR se omite NUM1/VAR1-N se asume un blanco.
- ◆ Si en la opción de /AGREGAR, la variable receptora solo tiene blancos, se mueve a partir de la primera posición.
- ◆ Cuando se usa la opción de /POS. NUM2/VAR2-N se indica que a partir del caracter NUM2/VAR2-N de la variable receptora, se mueve la información. El primer caracter de una variable es el 1.
- ◆ Si NUM2/VAR2-N no está en rango se dará error en ejecución.
- ◆ NUM1, VAR1-N, NUM2, VAR2-N deben ser N1 de cero decimales. Si no, se diagnostica en compilación.

Los siguientes comentarios son aplicables al formato 2:

- ◆ Ambas variables tiene que ser del tipo texto (T), del tipo imagen (I) o del tipo (A).
- ◆ Cuando se mueve información de una variable texto a otra variable texto se copia toda la información. La variable receptora debe tener área (máxima) para poder almacenarla.
- ◆ En el caso de mover una información de texto/imagen/audio se dará diagnóstico si se excede el AREA máxima separada para la variable receptora.
- ◆ En este formato NO se deben especificar opciones y se asume siempre /INICIO.

Los siguientes comentarios son aplicables al formato 3:

- ◆ Se mueve la información de **la variable texto VAR-T, al vector alfanumérico VAR-X** con subíndice VAR-N.
- ◆ VAR-N se incrementa en el número de renglones que se pasaron, se incrementa ANTES de pasar cada renglón del texto.
- ◆ Si se desea saber los caracteres de control que se usaron para formar el texto se debe poner la opción /CC seguida de un vector alfanumérico de 2 caracteres con subíndice VAR-N. Note que el subíndice tiene que ser el mismo del vector con los renglones.

El siguiente ejemplo ilustra el uso de esta instrucción:

a) En DBD se definió:

```
FOTO-1    ** R0 S I AREA 20
FOTO-2    ** R0 S I AREA 200
TEXTO     ** R0 S T
DIA       ** R0 S F M16
X50       ** R0 S X 50
SUB       ** R0 S N1 0
RENG.     ** R0 S X 30
```

b) En CLS se ponen las siguientes instrucciones:

```
MOVER 'A LA FECHA' A X50
MOVER DIA A X50 /AGREGAR
MOVER FOTO-1 A FOTO-2
VECTOR RENG RANGO 1 100
SUB = 0
MOVER          TEXTO          A          RENG( SUB )
```

c) Los siguientes comentarios son aplicables a las instrucciones:

- ◆ Con la primera instrucción de mover se almacena en X50 la literal 'A LA FECHA' rellenando a blancos. Como NO especificó ninguna opción, se asumió la de /INICIO.
- ◆ En la segunda instrucción se edita la variable DIA con su máscara (M16) y el resultado se mueve dejando un blanco entre la palabra FECHA y el resultado editado.

- ◆ En la tercer instrucción, la FOTO-1 se mueve a la variable FOTO-2 como el área es mayor no es posible un diagnostico.
- ◆ Por último se pasan a RENG los diferentes renglones del texto.
- ◆ Si TEXTO NO tiene información, SUB vale 0. Si se pasan por ejemplo 3 renglones vale 3.

1.5) Instrucción de Convertir.

Esta instrucción pasa uno o varios caracteres de una variable a otra y en el caso de números se realiza la conversión que sea necesaria.

El formato general de la instrucción es:

```
CONVERTIR    NUM/LET/VAR1  A    VAR2
POS. NUM-3/VAR3-N  NUM-4/VAR4-N
(/LONG. NUM5/VAR5-N)
(/ERROR #)
```

Los siguientes comentarios son aplicables:

- ◆ Se pasa uno o varios caracteres de NUM/LET/VAR1 a VAR2.
- ◆ Con /POS. se indica la posición NUM3/VAR3-N de donde se saca el caracter y la posición NUM4/VAR4-N en donde se almacena.
- ◆ Con /LONG. (Opcional) se especifica en NUM-5/VAR5-N cuantos caracteres se va a mover. Si no se especifica se asume 1.
- ◆ Las variables VAR1 y VAR2 pueden ser N1, N2, N3, N4, X, F o S. En el caso de F se asume igual a N2 con cero decimales y en el caso de S se asume igual a X de longitud 2.
- ◆ Si VAR2 es numérica o fecha y VAR1 alfanumérica y se pasa un caracter que no es numérico, ejecuta la instrucción con etiqueta # de la opción /ERROR #. Si NUM3/VAR3-N o NUM4/VAR4-N contiene una posición fuera de la variable, se ejecuta la instrucción con etiqueta # de la opción /ERROR #.
- ◆ Si NUM3-N, VAR3-N, NUM4, VAR4-N tienen decimales, son ignorados por esta instrucción.

- ◆ Para el cálculo de la posición, se asume que el primer caracter a la izquierda es el 1, el segundo a la izquierda es el 2, etc.
- ◆ Para efectos de esta instrucción el punto decimal no ocupa una posición.
- ◆ Si no se especifica la opción /ERROR y hay un error se cancela el programa.
- ◆ La instrucción básica pasa un caracter, cuando se usa la opción /LONG. con NUM5 mayor a 1 o con VAR5 se genera el ciclo para hacer la instrucción.

El siguiente ejemplo aclara el uso de esta instrucción. Suponga que se desea pasar la fecha que se encuentra a FECHA-A a FECHA-N:

a) En DBD se definió:

AREA

FECHA-A * * R0 S X 8

FECHA-N * * R0 S F

b) En CLS se ponen las siguientes instrucciones de Lenguaje **SISINF**:

FECHA-N = 19000000

CONVERTIR FECHA-A A FECHA-N /POS. 1 7 /LONG.2

CONVERTIR FECHA-A A FECHA-N /POS. 4 5 /LONG.2

CONVERTIR FECHA-A A FECHA-N /POS. 7 3 /LONG.2

c) Si se supone que el contenido de FECHA-A es:

23/08/86

12345678 (posición)

d) El contenido de FECHA-N es:

19860823

12345678 (posición)

1.6) Instrucción para sacar Parámetros.

Esta instrucción sirve para sacar parámetros tales como la fecha, la hora, el

número de la terminal y moverlos a una variable. También se mueve el contenido de variables internas usadas por diferentes instrucciones a la variable indicada.

El formato general de la instrucción es:

SACAR PARAMETROS	(FECHA	VAR-F)
	(HORA	VAR-X)
	(TERM.	VAR-N)
	(NUM-CA.	VAR-N)
	(PAG.	VAR-N)
	(RENG.	VAR-N)
	(BATCH	VAR-N)
	(PROGRAMA-ANTERIOR	VAR-X)
	(PROGRAMA	VAR-X)
	(CONDICION-EJECUTA	VAR-N)
	(CONDICION-TERMINAR	VAR-N)
	(NUMERO-DE-LECTURAS	VAR-N)
	(NUMERO-DE-VARIABLES	VAR-N)
	(NOMBRE-DE-VARIABLE	VAR-X)
	(NUMERO-DE-AREA	VAR-N)
	(RENGLON-COLUMNA	VAR-NR-N VAR-NC-N)
	(SELECCIONADO	VAR-N)
	(OPCION	VAR-X)
	(CARACTERES-DE-TEXTO	VAR-N)
	(CARACTERES-DE-IMAGEN	VAR-N)
	(TIPO-DE-ERROR	VAR-N)
	(CODIGO-DE-ERROR	VAR-N)
	(SUBCODIGO-DE-ERROR	VAR-N)
	(DESCRIPCION-DEL CODIGO	VAR-X)
	(DESCRIPCION-POR-SO	VAR-X)
	(PROGRAMA-SIGUIENTE	VAR-X)
	(ARCHIVO-DEL-ERROR	VAR-X)
	(LINEA-DE-ERROR	VAR-N)
	(PROGRAMA-DEL-ERROR	VAR-X)

(APLICACION	VAR-X)
(VERSION	VAR-X)
(INSTALACION	VAR-N)
(MODO	VAR-X)
(INICIALES-DEL-ERROR	VAR-X)
(SISTEMA-OPERATIVO	VAR-N)
(AVANCE-REGLONES	VAR-N)
(AVANCE-PUNTOS	VAR-N)

Los siguientes comentarios y restricciones son aplicables:

- ◆ Se puede pedir una o varias de estas opciones en cualquier orden.
- ◆ Si se pide opción **FECHA**, se almacena en VAR-F la fecha del día. VAR-F debe ser variable fecha.
- ◆ Si se pide la opción **HORA**, se almacena en VAR-X la hora del día en el formato de HH:MM:SS. La variable VAR-X debe ser alfanumérica y si es de menos de 8 caracteres se trunca; si es de más se rellena a blancos.
- ◆ Si se pide **TERM.** se almacena el número de la terminal en VAR-N. La variable VAR-N debe ser numérica. En algunas versiones el número puede ser también asignado por una variable de ambiente o un número de usuario de **SISINF** consecutivo (Ver opción de Número de Terminal Fijo en la opción de Parámetros de Instalación en el manual de CPS).
- ◆ Si se pide **NUM-CA.** Se almacena en VAR-N el número asociado a la clave de acceso que se teclee al ejecutar el módulo de ELS.
- ◆ La opción **PAG.** almacena en la variable numérica VAR-N la página en donde se imprimió información con IMPRIME DETALLE o IMPRIMIR ENCABEZADO.
- ◆ La opción **RENG.** almacena en la variable numérica VAR-N el renglón en donde se imprimió información con IMPRIMIR DETALLE o IMPRIMIR ENCABEZADO. Si estas instrucciones se usaron con la opción "/" el renglón almacenado será el último.
- ◆ Si se usa la opción **/RENG.** en un programa en que se usen las nuevas instrucciones de impresión, se cambia a AVANCE-REGLONES.
- ◆ Si se pide **BATCH** se almacena en VAR-N un 0 si el programa se ejecuta en forma interactiva o un 1 si se esta ejecutando en batch.

- ◆ Si se pide **PROGRAMA-ANTERIOR** se almacena en VAR-X las iniciales de DBD y las del programa que pidió la ejecución de este programa. Si es el primero que se ejecuta se almacenarán blancos.
- ◆ Si se pide **PROGRAMA** se almacena en VAR-X las iniciales de DBD y las iniciales de este programa. Esta opción es útil cuando se hacen rutinas generales que se usarán en varios programas.
- ◆ Si VAR-X de la opción **PROGRAMA** ó **PROGRAMA-ANTERIOR** es de menos de 4 caracteres se trunca, si es de mas se rellena a blancos.
- ◆ Si se pide **CONDICION-DE-EJECUTA** se almacenará en VAR-N la causa por la que se terminó la instrucción de LEER TERMINAL. Un valor del 1 al 8 indica que se tecló una tecla de función, un -1 tecló \, un -2 tecló ?, un -3 terminó por el tiempo, un -4 por la primera condición de ejecuta, ...
- ◆ Si se pide **CONDICION-TERMINAR** se almacena en VAR-N la tecla de función de OPCION TERMINAL AREA ... /TERMINAR. Cero indica que no se tecló o no hay.
- ◆ Las opciones **CONDICION-EJECUTA** y **CONDICION-TERMINAR** se pueden intercambiar ya que se tiene los dos nombres solo por mayor claridad en los programas.
- ◆ Si se pide **NUMERO-DE-LECTURAS** se almacena en VAR-N el número de instrucciones de LEER TERMINAL que se han ejecutado en un área específica entre el desplegar y la condición de ejecuta.
- ◆ Si se pide **NUMERO-DE-VARIABLE** se almacena en VAR-N el número de la variable dentro de un LEER TERMINAL y en la cual se detectó la condición de ejecuta.
- ◆ Si se pide **NOMBRE-DE-VARIABLE** se almacena en VAR-X el nombre de la variable de un LEER TERMINAL mismo en el que se detectó la condición de ejecuta. Esta opción puede consumir mucha memoria ya que se deberá tener los nombres de todas las variables que se lean o modifiquen en las diferentes áreas.
- ◆ Si VAR-X de la opción **NOMBRE-DE-VARIABLE** es de menos de 30 caracteres se trunca, si es de más se rellena con blancos.
- ◆ Es importante señalar que si la variable es igual, el nombre que se almacenará será el definido en DBD.
- ◆ Si se pide **NUMERO-DE-AREA** se almacena en VAR-N el número del área que se esta procesando en este momento.

- ◆ Si se pide **REGLON-COLUMNA** se almacena en VAR-NR-N el renglón y en VAR-NC-N la columna en donde está la variable en el que se detectó la condición de ejecuta. Tanto el renglón como la columna son referentes a la terminal y NO relativos al área en donde está la variable.
- ◆ En el formato de MENU con marcas se saca el renglón y la columna (aproximado) donde se seleccionó o donde se posicionó en una marca. Con esto se puede saber donde desplegar información relativa a la marca.
- ◆ Si se pide **SELECCIONADO** se almacena en VAR-N un 0 (Posicionado) o un 1 (Seleccionado) dependiendo si se seleccionó la marca. Lo anterior es aplicable solo al formato de MENU de marcas.
- ◆ Si se pide **OPCION** se almacena en VAR-X el contenido de la opción de MENU para que se calcule su NIVEL en la rutina especificada en /CALCULAR-NIVEL.
- ◆ Si se pide **OPCION** se almacena en VAR-X el contenido de la opción de MENU cuando se tecléa una tecla de función para que se pueda dar la ayuda adecuada.
- ◆ Las opciones **CONDICION-EJECUTA, CONDICION-TERMINAR, NUMERO-DE-LECTURAS, NUMERO-DE-VARIABLE, NOMBRE-DE-VARIABLE, NUMERO-DE-AREA, REGLON-COLUMNA, SELECCIONADO y OPCION** se discute su uso junto con las instrucciones de terminal.
- ◆ Si se pide **CARACTERES-DE-TEXTO** se almacena en la variable numérica el número de caracteres de la última variable texto que se leyó (LEER DISCO) o se formó (FORMAR TEXTO). Este número de caracteres incluye también los de control.
- ◆ Si se pide **CARACTERES-DE-IMAGEN** se almacena el total de caracteres que ocupa una variable imagen después de un LEER DISCO.
- ◆ Si VAR-N de caracteres de texto ó caracteres de imagen no es N2 se puede truncar información.
- ◆ Si se pide la opción de **TIPO-DE-ERROR** en el programa que procesa los errores se almacenará por una sola vez en la variable VAR-N un número que identifica las diferentes causas de los errores, los códigos negativos indican error informativo, los positivos son errores de cancelación. Los códigos son:
 - 1 error en la programación o datos, -2 error en la programación, -3 error en datos, -4 error de actualización, 1 error interno, 2 error de

programación, 3 error de seguridad, 4 error al abrir un archivo, 5 error al hacer un pedido al sistema operativo, 6 error de falta de área en disco, 7 error al compartir un archivo, 8 archivo dañado, 9 archivo lleno, 10 error en RECA interno, 11 error en comunicación con PEVX, 12 error de máximos, 13 error en datos y 14 error de archivo mal formado.

- ◆ Si se pide la opción de **CODIGO-DE-ERROR** en el programa que procesó los errores se almacenará por una sola vez en la variable VAR-N el número del error. Este número es el que aparece con la descripción de errores que se despliegan en la ejecución de las instrucciones. Al final de cada sección como es Instrucciones Generales, Archivos Temporales, Instrucciones de Terminal, ... se listan los diferentes errores.
- ◆ Con la opción **SUBCODIGO-DE-ERROR** en el programa que procesa los errores se almacenará por una sola vez en la variable VAR-N el subcódigo asociado al error. Dependerá de cada error el significado del subcódigo.
- ◆ Con la opción **DESCRIPCION-DEL-CODIGO** en el programa que procesa los errores se almacenará por una sola vez en la variable VAR-X la descripción del error. Al final de cada sección como es Instrucciones Generales, Archivos Temporales, ... se listan los diferentes errores, esta descripción es la que se almacena.
- ◆ Con la opción **DESCRIPCION-POR-SO** en el programa que procesa los errores se almacenará por una sola vez en la variable VAR-X la posible causa del error reportada por el Sistema Operativo de la máquina.
- ◆ Con la opción **PROGRAMA-SIGUIENTE** en el programa que procesa los errores se almacenará por una sola vez en la variable VAR-X el nombre del siguiente programa a ejecutar en caso de error informativo. Si el error es de cancelación se almacenarán blancos. Cuando se está procesando un programa y ocurre un error informativo, al pedir la ejecución del siguiente programa se procederá a ejecutar el programa; una vez desplegada la información del error la ejecución deberá continuar en el programa siguiente.
- ◆ Con la opción **ARCHIVO-DEL-ERROR** en el programa que procesa los errores se almacenará por una sola vez en la variable VAR-X el nombre del archivo en que ocurrió el error. Si no hay archivo se almacenaran blancos.
- ◆ Con la opción **LINEA-DEL-ERROR** en el programa que procesa los errores se almacenará por una sola vez en la variable VAR-N la línea del programa en donde ocurrió el error.
- ◆ Con la opción **PROGRAMA-DEL-ERROR** en el programa que procesa los errores se almacenará por una sola vez en la variable VAR-X el nombre del

programa de la instrucción PROGRAMA en donde ocurrió el error.

- ◆ Con la opción **APLICACION** en el programa que procesa los errores se almacenará en la variable VAR-X el nombre de la aplicación, la cual es dato en el ELS de paquetes. En el ELS normal se almacenan blancos.
- ◆ Con la opción **VERSION** en el programa que procesa los errores se almacena en la variable VAR-X la versión de la aplicación, la cual es dato en el ELS de paquetes. En el ELS normal se almacenan blancos.
- ◆ Con la opción de **INSTALACION** en el programa que procesa los errores se almacena en VAR-N el número de la instalación la cual es un dato en el ELS de paquetes. En el ELS normal se almacenan blancos.
- ◆ Con la opción **MODO** se almacena en la variable VAR-X dos letras para indicar el tipo de la última instrucción de lectura y así poder dar una mejor ayuda. Las claves son:

RR Renglón, lectura con /R

CR Columna, lectura con /R

RL Renglón y lectura

CL Columna y lectura

RM Renglón y modificación

CM Columna y modificación

ML Menú de Letrero

MV Menú de variables

TL Texto en lectura

TM Texto en modificación

IL Imagen en lectura

IM Imagen en modificación

AL Audio en lectura

AM Audio en modificación

MI Menú de imagen

MM Menú de marcas

- ◆ Con la opción **INICIALES-DE-ERROR** se almacena las iniciales de la DBD y las iniciales de programa en donde se dio el último error en VAR-X.

Nota: Si el error ocurrió en un programa que se ejecutó con EJECUTA-PROGRAMA no se sacaría el programa del error con la opción

PROGRAMA-ANTERIOR.

- ◆ Con la opción de **SISTEMA-OPERATIVO** se almacena en VAR-N un número para identificarlo. Los números son:

12	PC	WINDOWS XP, Vista ó 7
20	PC	XENIX, UNIX-SCO
21	HP	HP-UX
24	PC	LINUX

- ◆ Con la opción **AVANCE-RENGLONES** se almacena en VAR-N el total de renglones impresos. Los renglones van a font principal ó font 0. La variable VAR-N puede ser N1 con 0 decimales o N2 con 0 a 4 DECIMALES.
- ◆ Con la opción **AVANCE-PUNTOS** almacena en VAR-N el total de puntos impresos. Los renglones van a font 0. La variable VAR-N puede ser N1 con 0 decimales ó N2 con 0 a 4 decimales.

La opción **TERM.** es muy útil cuando se desea que un programa sólo ejecute en una sola terminal o en ciertas terminales, entonces por programa se saca el número de la terminal y se revisa que sea el adecuado.

La opción **NUM-CA.** es muy útil cuando se desea llevar registro de las personas que dan de alta cierta información. Así, por ejemplo, en un sistema de cheques se puede grabar en el archivo el número asociado a la clave de acceso de la persona que pidió se generara un cheque. Con esto se puede responsabilizar a cada quién de la información.

Otro uso de la opción **NUM-CA.** es restringir el uso de ciertas opciones en un sistema. Así, desde el programa de menú de un sistema se pueden desplegar solamente las opciones a las que tiene acceso la persona que en ese momento usa dicho sistema.

La opción **PAG.** se puede usar para formar índices de reportes. Así por ejemplo en un catálogo de clientes que se imprimirá ordenado por apellido, se puede detectar el cambio de la primera letra del mismo y sacar la página donde ocurrió.

Cuando se desea que cierta información que se imprimirá en varios renglones aparezca en la misma página se puede usar la opción de las instrucciones de impresión. Cuando por la naturaleza del programa NO se pueda hacer la impresión en una sola instrucción se podrá sacar el renglón actual con la opción **RENG.** y si ya no va a ser posible que se imprima en la misma página CAMBIAR de hoja.

La opción **CARACTERES-DE-TEXTO** se puede usar para calcular el tamaño de un área en el formato de TEXTO, es importante señalar que incluye los caracteres de control pero al formarse el área, las palabras que ya no caben en un renglón se pasan al siguiente.

1.7) Buscar en una tabla y mover un renglón.

Se busca en una tabla definida en DBD por rango, por igual, o renglón por renglón. Una vez encontrado el renglón de la tabla que cumple con la búsqueda, se mueve dicho renglón en las variables que se indiquen. El formato general es:

```
BUSCAR TABLA NOMT EJECUTA # IGUAL VAR-B/LET/NUM    $
                               VAR-C2 VAR-C3 VAR-C4 ...
```

```
BUSCAR TABLA NOMT EJECUTA # RANGO VAR-B/LET/NUM    $
                               VAR-C1 VAR-C2 VAR-C3 ...
```

```
BUSCAR TABLA NOMT EJECUTA # INICIO                $
                               VAR-C1 VAR-C2 ...
```

```
BUSCAR TABLA NOMT EJECUTA # SIG.                  $
                               VAR-C1 VAR-C2 ...
```

Los siguientes comentarios y restricciones son aplicables:

- ◆ **NOMT** es el nombre de la tabla definida en DBD.
- ◆ **VAR-B/LET/NUM** es la variable, literal o número que se usa como variable de búsqueda.
- ◆ Si el renglón que se busca no se encuentra en la tabla se ejecuta la instrucción con etiqueta igual a #.
- ◆ Si se especifica la opción de **IGUAL** se busca en la primera columna de la tabla, el renglón que sea igual a la variable de búsqueda. Si hay varios renglones cuya primera columna es igual, se encontrará siempre el primero.
- ◆ Si se especifica la opción de **RANGO** se busca en la primera columna de la tabla, el renglón que sea mayor que la variable de búsqueda y la solución es el renglón inferior. **NO** se revisa que la tabla esté ordenada, así que el primer renglón que cumpla, es la solución.
- ◆ Si se especifica la opción de **INICIO** se buscará el primer renglón de la Tabla.

- ◆ Si se especifica la opción de SIG. Se buscará el siguiente renglón al que está posicionado. Al final cuando ya no se tenga información se posicionará al inicio.
- ◆ Las columnas de las tablas se almacenan en las variables VAR-C1, VAR-C2, ... dependiendo de la opción. Note que al usar la opción de IGUAL no se almacena la primer columna.
- ◆ Los tipos de la variable de búsqueda, así como VAR-C1, VAR-C2, ... deben corresponder a la columna de la tabla, es decir, si la columna es numérica (N1, N2, N3 o N4), la variable debe ser numérica (N1, N2, N3 o N4), aunque las decimales sean diferentes. Si la columna es X, la variable debe ser X aunque puede ser de diferente longitud, si la columna es F, la variable debe ser F, y si es S debe ser S.
- ◆ El número de variables en la instrucción de BUSCAR TABLA puede ser menor al número de columnas de la tabla.

El siguiente ejemplo ilustra el uso de esta instrucción:

a) En DBD se definió:

AREA

NUM-EST * * R0 S X N1 0

NOM-EST * * R0 S X 30

*

TABLA EST 2 N1 0 X 20

T 1 'NUEVO LEON'

T 2 'EDO. DE MEXICO'

T 3 'PUEBLA'

*

COEF-X * * R0 S N2 1

COEF-1 * * R0 S N1 0

COEF-2 * * R0 S N3 8

*

TABLA COEF 2 N1 0 N 2 4

T 1 0.0002

T 12 0.0005

T 20 0.0008

b) Un ejemplo de buscar por igual sería:

Se define NUM-EST

BUSCAR TABLA EST EJECUTA 10 IGUAL NUM-EST NOM-EST

En este ejemplo NUM-EST puede tomar cualquier valor entre 1 y 3. Note que la segunda columna en la tabla es de longitud 20 y NOM-EST es de longitud 30, en este caso se rellena a blancos. En el caso de ser menor se trunca.

c) Una búsqueda por rango:

Se define COEF-X

BUSCAR TABLA COEF EJECUTA 10 RANGO COEF-X COEF-1 COEF-2

En éste ejemplo si COEF-X es igual a 15 se almacenará en COEF-1 un 12 y en COEF-2 un 0.00050000. Note que hay correspondencia entre campos dado que todos son numéricos aunque uno es N2 y el otro N1 y el número de decimales no sea igual.

Si el valor de COEF-X fuera de 0 se sale por el ejecuta, si fuera de 50 daría por resultado el último renglón.

d) Un ejemplo de buscar con SIG.:

10 BUSCAR TABLA EST EJECUTA 11 SIG. NUM-EST NOM-EST

 EJECUTA 10
 11 _____

En éste ejemplo se recorren todos los renglones de la tabla y al final estará nuevamente posicionado al inicio.

2) INSTRUCCIONES DE CONTROL.

En **SISINF** la ejecución de las instrucciones es secuencial a menos que se use la instrucción de SI, EJECUTA, EJECUTA RUTINA o ciertas instrucciones del banco de datos, de archivos secuenciales y de Terminal que tienen incluida la pregunta por ciertas condiciones especiales. Dichas instrucciones además de realizar funciones específicas también realizan funciones de **control** de la ejecución del programa.

Aquí solo se describen las instrucciones de SI y EJECUTA.

2.1) Etiquetas.

A las instrucciones del lenguaje **SISINF** se les puede asignar un número llamado etiqueta para que posteriormente se haga referencia a las instrucciones que tienen dichas etiquetas. Las siguientes son reglas para asignarlas:

- ◆ Se especifican por un número de 1 a 9999.
- ◆ Se especifican a la izquierda de la instrucción.
- ◆ En un programa o una rutina no puede haber etiquetas repetidas.
- ◆ Sí se puede tener el mismo número de etiquetas en un programa y en una rutina.
- ◆ Se pueden asignar en cualquier orden.
- ◆ Las declaraciones (IGUAL, VECTOR, PROGRAMA,...) no pueden tener etiqueta.

2.2) Instrucción de comparación total.

Esta instrucción sirve para hacer una o varias comparaciones y si se cumple se ejecuta otra instrucción.

El formato general es:

SI condición **OP** condición **OP** ... otra instrucción.

En donde OP es:

- Y Indica que se debe cumplir la condición anterior y la siguiente.
- O Indica que se debe cumplir la condición anterior o la siguiente.

El formato general de **condición** es:

```
NUM/VAR-N/VAR-F OPC NUM/VAR-N/VAR-F  
LET/VAR-X/VAR-S OPC LET/VAR-X/VAR-S
```

En donde OPC es cualquiera de los siguientes:

- = igual
- <> no igual
- < menor
- > mayor
- <= menor igual
- >= mayor igual
- =P igual parcial
- <>P no igual parcial

Los dos últimos operadores son descritos en la sección 2.3.

Los siguientes son ejemplos válidos de esta instrucción:

```
SI A = 3 o B = 4 EJECUTA 10  
SI A = B y C = D I = I + 1  
SI FECHA > 20050101 y FECHA <= 20060101 EJECUTA 50  
SI CIUDAD <> 'MTY' EJECUTA 100  
SI IMPORTE < 20000.00 EJECUTA 60
```

Las siguientes restricciones y comentarios deben ser observados para la condición:

- ◆ En una condición no se pueden mezclar variables alfanuméricas ni letreros con variables numéricas o números.
- ◆ En una instrucción de SI, una condición puede ser alfanumérica y otra condición numérica.
- ◆ Las variables tipo Imagen (I) o Audio (A) no se pueden comparar.
- ◆ Las variables tipo Texto (T) no se pueden comparar en esta forma (ver

sección 2.3).

- ◆ Se pueden comparar dos variables numéricas que sean de diferente precisión o número de decimales, es decir se puede comparar una variable N1 con 2 decimales, con una N3 con 0 decimales.
- ◆ La comparación numérica se hace con una cierta precisión, en lo referente a los decimales la cual varía para los diferentes modelos y marcas de computadora.
- ◆ En la comparación alfanumérica, el orden de las letras y números dependerá del modelo y marca de la computadora.
- ◆ La comparación alfanumérica se hace de izquierda a derecha, si una variable tiene una longitud mayor que la otra, la de menor longitud se completa a blancos a la derecha del campo.
- ◆ En constantes numéricas el punto decimal es opcional si no hay decimales. Así. Los siguientes números generan el mismo resultado: 50, 50., 50.000.
- ◆ Las variables FECHA se asumen numéricas, y dado que el formato es año, mes y día, se pueden comparar sin mayor problema.
- ◆ Las variables FECHA se pueden comparar usando una especie de constante fecha; especificándola en 8 dígitos, los primeros 4 para el año (0 es el año cero), los siguientes dos para el mes (01 a 12), y los siguientes dos para el día (01 a 31). NO se revisa si la fecha especificada en esta constante es una fecha válida.
- ◆ Las variables S se asumen igual a una variable X de longitud 2 para propósito de esta instrucción.
- ◆ No se diagnostica si se comparan dos constantes.

Los siguientes comentarios son aplicables a la instrucción en general:

- ◆ El orden en el que se realizan las comparaciones es de izquierda a derecha.
- ◆ Al encontrar una 'O' implica que si ya se cumplieron las condiciones anteriores se ejecuta la otra instrucción.
- ◆ La otra instrucción no puede ser una instrucción de PROGRAMA, otro SI o una declaración (IGUAL, VECTOR, ...)

2.3) Instrucción de comparación parcial.

El formato general de la instrucción es el que se describió en la sección para la instrucción SI, en esta sección se describe el funcionamiento de los operadores =P y <>P. El formato de condición es:

LET / VAR-F-X =P VAR-X / VAR-T

LET / VAR-F-X <>P VAR-X / VAR-T

Estos operadores permiten revisar si el contenido de una variable alfanumérica o texto cumple con un patrón de letras en alguna posición.

Los siguientes comentarios son aplicables:

- ◆ En LET o VAR-F-X está una o varias letras, o una o varias palabras que son las que se buscarán. VAR-F-X debe ser tipo X.
- ◆ Se busca en la variable tipo X (VAR-X) o en la variable tipo texto (VAR-T).
- ◆ Si se especifica =P la condición se cumple si se encuentra, con <>P si no se encuentra.
- ◆ Existen caracteres especiales al hacer la comparación, dichos caracteres son /\$, /?, /# y /@.
- ◆ Como al leer la información se quitan blancos a la izquierda y blancos a la derecha e internamente antes de hacer la búsqueda también se hace, si se desea comparar con blanco se deberá especificar /\$.
- ◆ A la variable VAR-X se le agrega un blanco al inicio y un blanco al final para hacer la comparación.
- ◆ A la variable VAR-T se le agrega un blanco al inicio y un blanco al final de cada conjunto de palabras especificado. Note que cuando se lea de Terminal cada conjunto será un renglón y cuando se forme con FORMAR TEXTO será la literal o variable de la instrucción.
- ◆ Cuando se especifica /? se busca por cualquier caracter diferente a blanco.
- ◆ Cuando se especifica /# se busca por cualquier caracter numérico (0 a 9).
- ◆ Cuando se especifica /@ se busca por cualquier caracter no numérico.
- ◆ Para efectos de ésta instrucción Mayúsculas y Minúsculas son iguales.

Los siguientes ejemplos ilustran el uso de esta condición en la instrucción de SI.

SI 'ARCH' =P AYUDA EJECUTA 10

Se busca en la variable tipo Texto con nombre AYUDA por las 4 letras que forman ARCH, si se encuentra se ejecuta la instrucción con etiqueta 10.

Note que si la literal se hubiera especificado como ' ARCH ', el resultado sería el mismo ya que los blancos a la izquierda y a la derecha son eliminados.

SI '/\$ARCHIVO/\$' =P AYUDA EJECUTA 10

Se busca por blanco (/\$), la palabra ARCHIVO y blanco (/\$) en la variable texto AYUDA. Si se encuentra se ejecuta instrucción con etiqueta 10. Note que si en el texto se tiene la palabra ARCHIVOS no se encontrará. También es importante señalar que si se puso:

FORMAR TEXTO AYUDA CON 'Archivo ...' /P

Es decir se inicia un párrafo con la palabra 'Archivo' al hacer la búsqueda internamente se agrega un blanco a la izquierda y como Mayúscula y Minúscula se toman igual, entonces se dará por encontrada.

SI '/\$MTY/\$?' =P DIRECCION EJECUTA 10

Se busca por blanco (/\$), la palabra MTY y blanco (/\$) en la variable tipo X DIRECCION. Si se puso:

MOVER 'MTY'	A DIRECCION	Si se encuentra.
MOVER 'MTY.'	A DIRECCION	No se encuentra.
MOVER 'CIUDAD MTY'	A DIRECCION	Si se encuentra.
SI 'T###/##/\$' =P	AYUDA	EJECUTA 10

Se busca por la letra T, 3 números y un blanco en la variable tipo texto AYUDA. Las siguientes literales:

'-T801'	Si se encuentra.
'T0801'	No se encuentra.
'T801.'	No se encuentra.

2.4) Instrucción para ejecutar otra Instrucción.

Esta instrucción sirve para pedir que se ejecute la instrucción con la etiqueta que se indica y de esta forma cambiar la secuencia de ejecución de instrucciones.

El formato general de la instrucción es:

EJECUTA (ETIQUETA) #

Los siguientes comentarios y restricciones son aplicables:

- ◆ La ejecución de las instrucciones es secuencial a menos que se use ésta instrucción u otra con la posibilidad de alterar el orden de ejecución.
- ◆ La palabra ETIQUETA es opcional.
- ◆ La siguiente instrucción después de ésta, deberá tener etiqueta, ya que de otra forma nunca se ejecutaría.

3) INSTRUCCIONES DE FECHAS.

En el lenguaje **SISINF** se tienen variables fecha que tiene la finalidad de manejar este tipo de información sin que el programador tenga que hacer algoritmos para hacer las operaciones más comunes.

Las variables fecha se pueden usar en instrucciones aritméticas, de pregunta y de convertir.

El formato con que se almacena es AAAAMMDD en donde AAAA es el año (cero el año cero), MM es el mes (01 a 12) y DD el día (1 a 28, 29, 30, 31, según el caso).

Así, por ejemplo, si se desea sacar el año de una variable fecha se puede poner la siguiente expresión aritmética.

$$\text{NUM} = \text{FECHA} / 10000 - 2000$$

Si en fecha se tiene 20020325 se almacenará en NUM un valor de 2.

Para otro tipo de operaciones se recomienda usar las dos instrucciones que tiene **SISINF** para el manejo de fechas, es decir, la de SUMAR o la de RESTAR.

Las instrucciones para leer de la terminal, escribir en la terminal o imprimir variables fecha son las mismas que para los demás tipos de variables, por lo que su uso se describe en la sección correspondiente.

3.1) Sumar días a una fecha.

Esta instrucción sirve para sumar o restar una cierta cantidad de días, meses o años a una fecha.

Su formato general es:

$$\text{SUMAR} \quad \text{VAR-N/NUM OP} \quad \text{A} \quad \text{VAR-F}$$

en donde OP es DIAS, MESES, AÑOS.

Los siguientes comentarios son aplicables:

- ◆ Si VAR-N o NUM es número negativo, se restarán días, meses o años.

- ◆ Si el número de días que se especifica sobrepasa varios años, la suma se hace correctamente, ya que se toma en cuenta que se pase de un año a otro y si es bisiesto.
- ◆ Si al sumar meses o años la fecha que resulta no es válida, se forzará el último día del mes para que sea válida, así, sumar 1 mes a 200950131 resultará 20050228.
- ◆ Si se suman MESES y la fecha es último día del mes, el resultado será último día del mes. Así, sumar al 20050228 2 meses se obtendrá 20050430.
- ◆ Si VAR-N o NUM tiene decimales, son ignoradas por esta instrucción.
- ◆ La variable VAR-N puede ser N1, N2, N3, N4 pero el número de días, meses o años que se suman deberán estar entre -9999 a 9999. Si es mayor se diagnostica.

3.2) Instrucción de restar dos fechas.

Esta instrucción sirve para restar dos fechas y obtener los días, meses y años.

Su formato general es:

RESTAR VAR-F1 A VAR-F2 VAR-D-N VAR-M-N VAR-A-N

Los siguientes comentarios y restricciones son aplicables:

- ◆ VAR-F1 y VAR-F2 deben ser definidos en DBD como variables fecha.
- ◆ VAR-D-M, VAR-M-N, VAR-A-N deben ser definidos en DBD como N1, N2, N3 o N4 con cualquier número de decimales.
- ◆ VAR-F1 debe ser menor o igual a VAR-F2, si no lo es se diagnostica.
- ◆ Se restan dos fechas y en VAR-D-N se almacenan los días de diferencia, en VAR-M-N los meses y en VAR-A-N los años.
- ◆ Si por ejemplo, se resta 1/1/05 a 31/12/05, el resultado sería 364 días, 11 meses, 0 años.
- ◆ Si se resta 31/1/05 a 28/2/05 el resultado sería de 28 días, 1 mes, 0 años.

- ◆ No se podrán restar dos fechas cuyo resultado sea mayor de 30000 días.
- ◆ Si al almacenar en VAR-D-N, VAR-M-N o VAR-A-N no cabe la información, se diagnostica.

4) EJECUTA PROGRAMA Y RUTINA.

Durante la ejecución de un programa, es posible que se desee ejecutar otro programa sin perder la posición ni los datos proporcionados durante la sesión; puede deberse a que al estar capturando información se detecte que es necesario dar de alta un registro de otro archivo. Para ello en **SISINF**, se tiene la instrucción EJECUTA PROGRAMA que permite transferir el control a un programa totalmente independiente y al terminar, continuar con la ejecución en la siguiente instrucción donde se le llamó.

También puede suceder que una serie de instrucciones se repitan en varias partes de un programa. Para ello el lenguaje **SISINF** tiene el concepto de RUTINAS. En una rutina se pone una sola vez ésta serie de instrucciones y luego se pide su ejecución varias veces en el programa. Bajo este concepto un programa de **SISINF** quedará integrado por el programa principal y al final del mismo una serie de rutinas. Es muy importante señalar que las variables que se usen serán comunes para todos. NO así las etiquetas en que el programa principal tiene las suyas y cada rutina también.

Este capítulo discute la instrucción para ejecutar otro programa y dos instrucciones y una declaración para el manejo de rutinas; Estas son una para pedir la ejecución de una rutina, otra para retornar, indicando que se continúe donde se pidió la ejecución de la rutina, y la que define su inicio.

Mediante opciones de ciertas instrucciones también se puede pedir la ejecución de una rutina, ésto solo se describe con las instrucciones de dichas opciones.

4.1) Instrucción para pedir la ejecución de otro PROGRAMA.

La instrucción para pedir la ejecución de otro programa es:

EJECUTA PROGRAMA VAR-A/LET

Los siguientes comentarios son aplicables:

- ◆ En VAR-A o LET se debe especificar xxyy del programa PSxxyy de **SISINF** que se desea ejecutar.
- ◆ Se asume que el programa que se especifica en VAR-A o LET está en el mismo dispositivo que el programa que se está ejecutando.

- ◆ La variable VAR-A que se especifica debe ser alfanumérica de 4 caracteres o bien un letrero (LET) que se especifique en 4 caracteres también.
- ◆ El programa llamado es un programa totalmente independiente y compilado por separado en Versión 5.1 o superior.
- ◆ El programa llamado regresará el control al primer programa cuando ejecute una instrucción de FIN, con o sin encadenamiento; Esto permite llamar los mismos programas que se accesan con menús en el sistema.
- ◆ El programa llamador deberá liberar los registros que pueda utilizar el programa llamado para evitar posibles ESPERA-R; No podrá tener archivos abiertos en forma exclusiva que use el programa llamado (permanentes o temporales), provocaría ESPERA-E o ESPERA-T,
- ◆ Esta instrucción no puede llamarse desde un programa que inició su ejecución a través de otro EJECUTA PROGRAMA.
- ◆ El programa que haga un EJECUTA PROGRAMA no podrá estar generando un listado, si se hace se diagnostica en ejecución.
- ◆ El programa que haga un EJECUTA PROGRAMA, si usa instrucciones de terminal, deberá ser con OPCION TERMINAL AREA.

4.2) Instrucción para pedir la ejecución de una RUTINA.

La instrucción para pedir la ejecución de una rutina es:

EJECUTA RUTINA NOM-1 (NOM-2 ...)

Los siguientes comentarios son aplicables:

- ◆ NOM-1, NOM-2, ... son los nombres de las rutinas que se desean ejecutar. Se debe especificar al menos una y se puede poner la misma varias veces.
- ◆ Se ejecutará primero la rutina con nombre NOM-1, cuando se retorne de ésta se ejecutará NOM-2, ...
- ◆ NOM-1, NOM-2, ... son rutinas que se deben definir al final del programa.
- ◆ Dentro de una rutina se puede pedir la ejecución de otra rutina, y dentro de esta otra, ... hasta 10 niveles.
- ◆ NO se revisa el caso cuando dentro de una rutina se pide la ejecución de

otra, dentro de esta a otra, etc. y una de ellas ya se había pedido su ejecución; así un ejemplo sería R1 pide ejecutar R2, R2 a R3, R3 a R4 y R4 a R2. El programador debe recordar que las variables son las mismas.

4.3) Declaración de inicio de una RUTINA.

La declaración para especificar el inicio de una rutina es:

RUTINA NOM

Los siguientes comentarios son aplicables:

- ◆ NOM es el nombre de la rutina cuyo inicio se define.
- ◆ El nombre de una rutina puede ser de 1 a 30 caracteres y no puede tener blancos intermedios ni iniciarse con + - \$ ' & números. Esta declaración NO puede tener etiqueta.
- ◆ Esta instrucción termina el programa principal o la rutina que se definía donde fue incluida.
- ◆ Si antes de esta declaración está el programa principal se genera un FIN.
- ◆ Si antes de esta declaración está otra rutina se genera un RETORNO.
- ◆ Las variables de esta rutina son comunes al programa principal y a otras rutinas.
- ◆ Los archivos temporales, permanentes, tablas y áreas de terminal que se usen también son comunes, es decir, se puede poner OPCION TEMPORAL 10... en el programa principal y hacer ESCRIBIR TEMPORAL 10... en la rutina y se hace referencia al mismo temporal.
- ◆ Las etiquetas son propias de cada rutina. Así es posible tener la etiqueta 10 en el programa principal y la etiqueta 10 también en una o varias rutinas. Cada una sólo servirá en donde se define y no tendrá relación.

4.4) Instrucción para regresar de una RUTINA.

El formato general de la instrucción para regresar a continuar la ejecución a donde se pidió la rutina es:

RETORNO

4.5) Ejemplo de uso de RUTINAS.

Un ejemplo del uso de las rutinas puede ser:

```

PROGRAMA '...'
_____
_____
B = 8
10  EJECUTAR RUTINA CALCULO
_____
_____
FIN
RUTINA CALCULO
_____
_____
10  SI B = 8 EJECUTA RUTINA MOVER
_____
_____
RETORNO
RUTINA MOVER
_____
_____
SI PAG = 0 RETORNO
_____
_____

```

Los siguientes comentarios son aplicables:

- ◆ En el programa se pide la ejecución de la rutina CALCULO en la instrucción con etiqueta 10.
- ◆ Se define las rutinas CALCULO y MOVER, al final del programa principal. Note que en la primera se usa la etiqueta 10 también, pero esta NO tiene relación con la 10 del programa principal.

- ◆ Cuando se usa la rutina CALCULO el valor de B que se usará será el mismo que se definió en el programa principal o sea 8.
- ◆ Una rutina puede pedir la ejecución de otra rutina. Así CALCULO pide la ejecución de MOVER.

5) TEXTOS.

En **SISINF** se tienen variables tipo texto (T) que son usadas para imprimir o desplegar información alfanumérica en uno o varios renglones.

Las características principales de este tipo de variables son:

- ◆ Puede ocupar varios renglones al desplegar o imprimirla.
- ◆ Se pueden formar con el contenido de una o más variables, cuyos valores de algunas de ellas se definen en ejecución.
- ◆ La longitud de un renglón se define en cada caso.

Este tipo de variables están pensadas para que se lea información de los textos por la terminal y luego mediante la instrucción de FORMAR TEXTO, formar el texto definitivo con uno o varios textos y con una o varias variables.

Aquí se describe la instrucción de FORMAR TEXTO, luego en las instrucciones de terminal e impresión se describe como desplegar e imprimir.

5.1) Formar Texto.

Se tiene una instrucción para formar un texto con base en otros textos, literales, variables numéricas, variables alfanuméricas, ...

Su formato general es:

```

FORMAR TEXTO VAR-T CON NUM/LIT/VAR
(/INICIO o /AGREGAR)
(/DIP (NUM-B/VAR-B NUM-C1B/VAR-C1B))
(/DII (NUM-B/VAR-B NUM-C1B/VAR-C1B NUM-C2B/VAR-C2B))
(/P o /I o /S (/RBA NUM-B/VAR-B))
(/F (/RBD NUM-B/VAR-B))
(/C o /JD o /CONTINUACION LIT-TE/VAR-TE LET-IN/VAR-IN)
(/HASTA-MARCA LIT-I/VAR-I LIT-D/VAR-D VAR-M)
(/ENCONTRADA # o /NO-ENCONTRADA #)
(/ERROR #) (/SACAR-DESDE-INICIO)

```

Los siguientes son comentarios generales a la instrucción:

- ◆ Se forma el texto en la variable VAR-T con un número (NUM) o una literal (LIT) o una variable (VAR).
- ◆ La variable VAR-T sólo puede ser del tipo texto.
- ◆ La variable VAR puede ser del tipo N1, N2, N3, N4 F, X, S o T dependiendo de las opciones.
- ◆ Las opciones se pueden especificar en cualquier orden.
- ◆ Como se describe en la explicación de las opciones algunas de ellas son aplicables a VAR-T, otras a VAR cuando es una variable texto y otras a NUM/LIT/VAR ésta última como numérica, alfanumérica, fecha o SI-NO.
- ◆ Cuando se especifica NUM, o VAR es una variable numérica o fecha, después de haberse editado se quitan los blancos a la izquierda y a la derecha y así se pasa al texto.
- ◆ Cuando se especifica VAR y ésta es una variable tipo X, los blancos a la izquierda sí se pasan al texto, pero los blancos a la derecha NO.
- ◆ Cuando se despliega o imprime LIT/VAR, la indentación es /P o /I y la literal o variable es la primera de la línea, se quitan blancos a la izquierda.
- ◆ Cuando se especifica VAR y ésta es una variable tipo Texto (T) el contenido se pasa tal como está VAR-T, salvo lo que se describe con /HASTA-MARCA.

5.1.1) Opciones de /INICIO y /AGREGAR.

Un texto se forma usando una o más veces ésta instrucción. Los siguientes comentarios son aplicables a la opción /INICIO o /AGREGAR.

- ◆ La primera vez que se use la instrucción se debe poner la opción de /INICIO para indicar que se inicialice VAR-T y que por lo tanto ésta sea la primera información del texto.
- ◆ Para formar el resto del texto se deberá usar ésta instrucción una o más veces con la opción de /AGREGAR con lo que se indica que se agregue al final del texto ya existente la nueva información.
- ◆ Si no se especifica /INICIO o /AGREGAR se asume /AGREGAR.

5.1.2) Cambiar las identaciones de DBD. Opciones DIP y DII.

Cuando un campo texto se define en el módulo DBD se especifica la indentación que tendrán los párrafos e incisos. Si éstas indentaciones se desean cambiar se necesita usar las opciones de /DIP o /DII.

Los siguientes comentarios son aplicables:

- ◆ Con la opción **/DIP** se define la indentación de párrafo que se usará en el texto de aquí en adelante. Con /DII se define la de inciso.
- ◆ En el caso de /DIP se dejarán NUM-B/VAR-B renglones en blanco antes de iniciar el párrafo, NUM-CB/VAR-CB columnas en blanco.
- ◆ En el caso de /DII se dejarán NUM-B/VAR-B renglones en blanco antes de iniciar el inciso, NUM-C1B/VAR-C1B columnas en blanco en el primer renglón del inciso y NUM-C2B/VAR-C2B columnas en blanco en el segundo, tercer, ... renglón del inciso.
- ◆ Con la opción /DIP sin NUM-B ... o la opción /DII sin NUM-B ... se define que de aquí en adelante se use nuevamente la indentación definida en DBD de VAR-T.
- ◆ NUM-B, VAR-B, ... deben ser numéricas del tipo N1 con cero decimales.
- ◆ Las opciones /DIP y /DII son aplicables a esta instrucción y las subsecuentes hasta que se cambie.
- ◆ Si un texto que contenga internamente las opciones de /DIP o /DII se modifica en la terminal, éstas opciones se ignoran para la modificación y se pierden.
- ◆ Si un texto que contenga internamente las opciones de /DIP o /DII se pasa con esta instrucción a otra variable texto, estas opciones se tendrán en la nueva variable.

5.1.3) Información de Control de indentación y renglones en blanco.

Cuando un texto se despliega o se imprime además de la información alfanumérica tiene información de control. Dicha información es por ejemplo fin de párrafo o fin de inciso, si una redacción es un párrafo o si es un inciso, etc.

Cuando se forma un texto se debe especificar donde se inicia un párrafo

donde se inicia un inciso y en donde terminan. Para ello se tienen las opciones de /P, /I, /S y /F. Estas opciones son información de control que estarán con la información alfanumérica.

Cuando se lee un texto de la terminal también se deberá indicar lo anterior.

Al desplegar o imprimir el texto con ésta información de control se tendrán las identaciones adecuadas y los renglones en blanco necesarios.

Los siguientes comentarios son aplicables:

- ◆ Con la opción /P (Párrafo) se indica que NUM/LIT/VAR será inicio de un párrafo y que se aplique la indentación de DIP de DBD o la de /DIP del programa.
- ◆ Con la opción /I (Inciso) se indica que NUM/LIT/VAR será el inicio de un inciso y que se aplique la indentación de DII de DBD o la de /DII del programa.
- ◆ Con la opción /S (Sin indentación) se indica que NUM/LIT/VAR se desea desplegar o imprimir sin indentación, es decir a partir de la columna 1.
- ◆ Con la opción /F (Fin) se indica que NUM/LIT/VAR serán el fin de párrafo o del inciso.

Dado que un texto se puede formar con el uso de ésta instrucción varias veces, no es necesario iniciar siempre un párrafo o un inciso con las opciones de /P, /I o /S y no es necesario que se termine siempre con /F. Los siguientes comentarios son aplicables:

- ◆ Cuando se especifica /P, /I o /S se asume que se termina NUM/LIT/VAR de la instrucción anterior, como si se hubiera especificado en ella la opción /F.
- ◆ Cuando se especifica /F se asume que en la siguiente instrucción de este tipo que se ejecute NUM/LIT/VAR será el inicio de un párrafo, de un inciso o bien será sin indentación. Lo anterior dependiendo de la última /P, /I o /S especificada.
- ◆ Cuando se usa la opción de /INICIO y no se especifica /P, /I o /S se asume /S es decir NUM/LIT/VAR se imprimirá o desplegará sin indentación.
- ◆ Note que un párrafo o un inciso se puede formar con varias instrucciones de este tipo. La primera debe tener la opción /P o /I O /S, la última opción /F y las intermedias no tener opción.
- ◆ Dado que VAR puede ser una variable texto puede suceder que como

información de control queden varias opciones del tipo /P, /I o /S seguidas. La última será la que se use como información de control, las opciones de /P, /I o /S se ponen antes que NUM/LIT/VAR y /F se pone después.

A manera de ejemplo y para ilustrar el uso de las diferentes opciones suponga que se desea desplegar una ayuda con:

El uso de las teclas de función es:

- ◆ F1 Se debe teclear para Terminar.
- ◆ F2 Se debe teclear para regresar al menú.

La mejor manera de leer éste texto y guardarlo (junto con otras ayudas) es un archivo. Pero si se quiere formar con ésta instrucción se tendría en DBD:

```
AYUDA      * * R0 S T DIP 0 3 DII 0 0 5
```

Se define un campo tipo texto con la indentación de párrafo de 0 renglones en blanco y 3 columnas en blanco en el primer renglón. La indentación de inciso será de 0 renglones en blanco, el primer renglón con 0 columnas en blanco y las restantes con 5.

En CLS se pondría:

```
FORMAR TEXTO AYUDA CON 'El Uso de las teclas de $
                        $función es: ' /P /F /INICIO
```

```
FORMAR TEXTO AYUDA CON '-F1 Se debe teclear para Terminar' $
                        /I /F /AGREGAR
```

```
FORMAR TEXTO AYUDA CON '-F2 Se debe teclear para $
                        $ regresar al menú' /F
```

Internamente el contenido de la variable texto AYUDA será:

```
/P  'El uso de las teclas de función es:' /F
```

```
/I  '-F1 se debe teclear para Terminar' /F
```

```
'-F2 se debe teclear para regresar a menú' /F
```

Los siguientes comentarios son aplicables:

- ◆ La primera instrucción de FORMAR TEXTO tiene la opción de /INICIO con esto se indica que será la primer información del texto.

- ◆ La segunda instrucción tiene la opción de /AGREGAR, la tercera NO la tiene pero al no especificarse, se asume. Con lo anterior esta información se agrega a lo ya existente.
- ◆ En la primer instrucción se tiene la opción de /P con la que se indica que la literal será un párrafo.
- ◆ La segunda instrucción tiene /I con lo que será un inciso. Note que la opción /F se podría haber omitido en la primer instrucción.
- ◆ En la tercer instrucción no se puso la opción /I ya que es la que está vigente.
- ◆ El contenido de la variable AYUDA tendrá las literales o sea la información alfanumérica del texto y los caracteres de control para indicar el uso de la indentación de párrafo o de inciso y las terminaciones de cada uno. Note que las opciones /P y /I están antes y las /F después.
- ◆ Si la variable AYUDA se hubiera leído se tendría la misma información en ella.

Se tienen algunas opciones que sirven para hacer alineaciones especiales (/C y /JD) y una de continuación cuando la información se va a imprimir o desplegar en dos o más renglones (/CONTINUACION): Estas opciones sólo se aplican a NUM/LIT/VAR de esta instrucción.

Los siguientes comentarios son aplicables:

- ◆ La opción /C indica que NUM/LIT/VAR se debe Centrar en el renglón.
- ◆ La opción /JD indica que NUM/LI/VAR se debe justificar a la derecha en el renglón.
- ◆ Si por el número de columnas en el que se imprime o despliega la información, el contenido de NUM/LIT/VAR no cabe en uno solo, se truncará la información.
- ◆ Para dejar renglones en blanco antes de NUM/LIT/VAR con opción /C o /JD se deberá usar la opción /RBA. Otra forma es especificando en la instrucción anterior /F y /RBD.
- ◆ Después de /C o /JC se asume una /F de forma que lo especificado en la siguiente instrucción será un párrafo o un inciso.
- ◆ Con la opción de /CONTINUACION LIT-TE/VAR-TE LIT-IN/VAR-IN se indica que si NUM/LIT/VAR no cabe en el renglón actual, se termine

poniendo LIT-TE/VAR-TE y en el siguiente renglón se inicie con LIT-IN/VAR-IN y se continúa con el resto de NUM/LIT/VAR.

- ◆ VAR-TE y VAR-IN deben ser variables tipo X de 1 a 10 caracteres.
- ◆ Por razones de implantación de **SISINF** en algunas máquinas LIT-TE, VAR-TE, LIT-IN y VAR-IN se redondea a un número de caracteres par.
- ◆ Si la longitud en caracteres de LIT-TE/VAR-TE más LIT-IN/VAR-IN más la indentación excede el máximo de columnas con las que se imprime o despliega la información, se ignora la opción en el renglón.
- ◆ Si un texto que tenga /CONTINUACION se modifica en la terminal, esta opción se ignora para el desplegado y se pierde, es decir no estará en el texto modificado.
- ◆ Si se usa /C, /JD, /CONTINUACION junto con VAR ésta no puede ser del tipo texto.

Una aplicación de la opción de /CONTINUACION es para imprimir o desplegar cantidades numéricas con máscara M13 a M15, las cuales pueden necesitar de más de un renglón y entonces es necesario terminar el primer renglón con una cierta literal e iniciar el siguiente renglón con otra literal y continuar con la cantidad.

Se tienen opciones para dejar renglones en blanco antes y después de NUM/LIT/VAR, dichas opciones son /RBA y /RBD. Los siguientes comentarios son aplicables:

- ◆ Con la opción /RBA NUM-B/VAR-B (Renglones en blanco antes) se indica el número de renglones en blanco antes de NUM/LIT/VAR.
- ◆ Con la opción /RBD NUM-B/VAR-B (Renglones en blanco después) se indica el número de renglones en blanco después de /NUM/LIT/VAR.
- ◆ La opción /RBA se debe usar junto con /P o /I o /S o /C o /JD.
- ◆ La opción /RBD se debe usar junto con /F o /C o /JD.
- ◆ Cuando /RBD o /RBA se usa con /P o /I o /F se cancela por una sola vez lo definido en DIP o DII.
- ◆ Cuando se usa /C o /JD y se quieren renglones en blanco se deberá usar /RBA o /RBD.
- ◆ Al inicio de desplegar o imprimir información NO se tendrán renglones en

blanco a menos que se use /RBA.

- ◆ La información de /RBA y /RBD es información de control. Si al formar el texto hay varias seguidas se usa la última.
- ◆ Cuando se usa /S se pueden dejar renglones en blanco con estas opciones.

En las opciones /DIP, /DII, /RBA y /RBD se pueden especificar los renglones en blanco y las identaciones con constantes y con variables. En el caso de constantes el compilador NO revisa que estén en rango. En ambos casos en ejecución se asume lo siguiente:

- ◆ En el caso de renglones en blanco y de identaciones, si tienen un valor negativo se asume 0.
- ◆ En el caso de indentación, si exceden al máximo de columnas especificadas, éstas se asumen. Es decir se despliega o imprime un caracter por línea.
- ◆ En el caso de renglones se revisa que estos no excedan de una página para el caso de impresión o de ½ página para el caso de que se despliegue por terminal.

5.1.4) Uso de textos base.

Se tiene la opción /HASTA-MARCA que sirve para formar un texto usando un texto base el cual contendrá marcas que indicarán el lugar y la información que se debe insertar en lugar de ellas.

Así por ejemplo suponga que se tiene la siguiente carta que se usará como texto base:

Estimado Señor [NOMBRE]

A la fecha nos adeuda la cantidad de [CANTIDAD] por

lo que Señor [NOMBRE] le sugerimos pase a pagar.

La opción de /HASTA-MARCA sirve para formar un nuevo texto en el que se reemplaza [NOMBRE] y [CANTIDAD]. NOMBRE y CANTIDAD son las palabras clave que se sustituyen.

La forma de trabajo de la opción:

/HASTA-MARCA LIT-I/VAR-I LIT-D/VAR-D VAR-M es:

- 1) Se pasa la información de VAR a VAR-T hasta encontrar la marca que se define con LIT-I/VAR-I. (Marca izquierda).
- 2) Se busca la marca definida con LIT-D/VAR-D. (Marca derecha).
- 3) La información entre ambas marcas se almacena en VAR-M.
- 4) Se guarda la posición siguiente a la segunda marca para que al ejecutar nuevamente ésta instrucción en VAR se continúe de aquí en adelante.

Los siguientes comentarios son aplicables a ésta opción y otras relacionadas:

- ◆ VAR debe ser de tipo texto.
- ◆ LIT-I, LIT-D, VAR-I y VAR-D deben ser de un caracter.
- ◆ VAR-I, VAR-D y VAR-M deben ser del tipo X.
- ◆ La opción de /ENCONTRADA # indica que se ejecute la instrucción con etiqueta # si se encuentra la marca.
- ◆ La opción de /NO-ENCONTRADA # indica que se ejecute la instrucción con etiqueta # si no se encuentra la marca.
- ◆ La opción de /ERROR # indica que se ejecute la instrucción con etiqueta # si se encuentra la primera marca pero no la segunda o bien la información NO cabe en VAR-M. Si no se especifica /ERROR y ocurren los problemas mencionados se cancela la ejecución del programa por error.
- ◆ Cuando se usa /HASTA-MARCA y ya no se encuentran marcas en VAR, se pone el apuntador al inicio, de forma que la próxima ejecución de esta instrucción en VAR sacará la información desde el inicio. La opción /SACAR-DESDE-INICIO realiza la misma operación.
- ◆ Note que la opción /SACAR-DESDE-INICIO es aplicable a VAR y no a VAR-T.
- ◆ Cuando se lee VAR de disco se está posicionado al inicio.
- ◆ Las marcas deben estar en un NUM/LIT/VAR, si está parte en uno y parte en otro no se encontrara. Cuando se lea de terminal deben estar en un renglón.
- ◆ No se puede tener LIT/VAR con opción /C, /JC o /CONTINUACION y

marcas.

5.1.5) Pasar una variable texto a otra variable texto.

Por último los siguientes comentarios son aplicables cuando se pasa una variable texto a otra variable texto:

- ◆ Las opciones /DIP, /DII, /P, /I, /S, /F, /RBA, /RBD, /C, /JD y /CONTINUACION es información de control que se almacena en la variable texto junto con la información alfanumérica; por lo cual en éste caso se pasa toda la información de VAR a VAR-T.
- ◆ Cuando se usa la opción /HASTA-MARCA toda la información de control de VAR se pasa también a VAR-T.
- ◆ Cuando se usa esta instrucción sin opciones, se pasa toda la información de VAR a VAR-T incluyendo la de control.
- ◆ Cuando se usa esta instrucción con opciones, se pasan éstas así como las que contenga VAR. Es importante señalar que varias opciones /P, /I o /S seguidas ocasionan que sólo se use la última al igual si se tiene varias /RBD, /RBA.
- ◆ Las opciones de /DIP y /DII sin parámetros ocasionan que se usen los valores definidos en DBD de la variable receptora.

5.2) Ejemplo.

El siguiente ejemplo de un sistema de clientes ilustra el uso de éstas opciones, para ello suponga que se tiene la variable tipo texto CARTA-TIPO, la cual se leyó y se almacenó en disco. Dicha CARTA-TIPO tiene dentro de la información las marcas NOMBRE y CANTIDAD las cuales deben ser reemplazadas por el contenido de la variable NOM-CLT y por SALDO, ésta última editada con números y letras.

Las definiciones en DBD serían:

ARCHIVO CLTS ...

NOM-CLT 'NOMBRE DEL CLIENTE' * R0 S X 40

 SALDO * * R0 S N3 2 MX '\$ZZ,ZZZ,ZZ9.99'

 ARCHIVO CARTAS ...

 CARTA-TIPO * * R0 S T

 AREA

M13 * * R0 S N3 2 M13

DIA * * R0 S F M18

CLAVE * * R0 S X 20

La parte del programa sería:

*

* Leer la información de disco

*

LEER DISCO CARTAS CARTA-TIPO

LEER DISCO CLTS NOM-CLT SALDO

SACAR PARAMETROS FECHA DIA

*

* Poner la fecha del día en la CARTA

*

IGUAL CARTAS CARTA-TIPO CARTA

FORMAR TEXTO CARTA CON DIA /JD /RBD 3 /INICIO /RBA 1

*

* Pasar texto hasta marca

*

10 FORMAR TEXTO CARTA CON CARTA-TIPO \$

/HASTA-MARCA '[' ']' CLAVE \$
/NO-ENCONTRADA 50 /ERROR 20

*

* Se encontró NOMBRE poner el del cliente

*

SI CLAVE <> 'NOMBRE' EJECUTA 11
FORMAR TEXTO CARTA CON NOM-CLT
EJECUTA 10

*

* Se encontró CANTIDAD editar el saldo

*

11 SI CLAVE <> 'CANTIDAD' EJECUTA 20
FORMAR TEXTO CARTA CON SALDO
M13 = SALDO
FORMAR TEXTO CARTA CON M13 \$
/CONTINUACION '###' '###'
EJECUTA 10

*

* Error en CARTA-TIPO

*

20 ERROR 'CARTA TIPO INCORRECTA'
FIN

*

* Imprimirla

*

50 _____

Se asume que otro programa lee, valida y guarda las diferentes CARTA-TIPO en el archivo CARTAS y luego este programa selecciona el tipo de carta y lo imprime. Note que la misma marca se puede repetir varias veces y cada vez se hará el remplazo adecuado.

6) DEFINICION DE IGUAL.

En el módulo DBD se describió la definición de IGUAL, el módulo CLS también tiene una definición de IGUAL para propósitos similares a los del IGUAL de DBD.

En ésta parte se describe y ejemplifica el uso de ambas definiciones de IGUAL.

6.1) Definición de IGUAL en DBD.

Las variables que se especificaron en DBD con la definición IGUAL ocupan la misma posición de memoria y por lo tanto tienen el mismo contenido aunque pertenezcan a diferentes archivos. Lo anterior nos evita el tener que mover la información cuando todas sus características son iguales.

A continuación se presenta un ejemplo del uso de variables definidas con IGUAL en DBD.

a) En DBD se define:

```

ARCHIVO ART 0 1
RECURRENTE 0 C 100 M 0
NUM-ART 'NUMERO DEL ARTICULO' * R0 LL1 N1 0
NOM-ART 'NOMBRE DEL ARTICULO' * R0 S X 30
*
ARCHIVO PROV 1 1
RECURRENTE 0 C 100 M 1 1
RECURRENTE 1 R 5 P 0
NUM-PROV 'NUMERO DEL PROVEEDOR' * R0 LL1 N1 0
NOM-PROV 'NOMBRE DEL PROVEEDOR' * R0 S X 30
IGUAL ART NUM-ART R1 S

```

En el archivo de proveedores en el recurrente 1 se tendrán los números de los artículos que ese proveedor puede surtir.

b) Se desea hacer un programa para consultar los artículos que surte un proveedor.

```

PROGRAMA 'CONSULTA ARTICULOS DE PROV.'
5  LEER TERMINAL EJECUTA 20 NUM-PROV
   BUSCAR DISCO PROV EJECUTA 11 PRIN-G1 NUM-PROV
   LEER DISCO PROV NUM-ART
8  BUSCAR RECURRENTE PROV 1 EJECUTA 5 SIG.
   LEER DISCO PROV NUM-ART
   BUSCAR DISCO ART EJECUTA 12 PRIN-G1 NUM-ART
   LEER DISCO ART NOM-ART
   ESCRIBIR TERMINAL NUM-ART NOM-ART
   EJECUTA 8
11 ERROR 'NO ENCONTRO PROVEEDOR'
    EJECUTA 5
12 ERROR 'NO ENCONTRO ARTICULO'
    EJECUTA 8
20  FIN

```

6.2) Definición de IGUAL en CLS.

Se define una variable del programa con las mismas características de otra, de un archivo específico. El formato general de la instrucción es:

```

IGUAL      NOMA    NOMV      NOMN-1 NOMN-2 ...

```

Los siguientes comentarios son aplicables:

- ◆ NOMA es el nombre del archivo permanente al cual pertenece la variable cuyo nombre es NOMV. Si la variable es de programa se deberá poner en lugar de NOMA la palabra AREA o en el caso de áreas con nombre, dicho nombre.
- ◆ NOMN-1, NOMN-2, etc., son los nombres que se asignan a las variables con características iguales a NOMV.
- ◆ Las variables NOMN debe no estar definidas en DBD.
- ◆ Esta instrucción se puede poner en cualquier parte del programa pero antes de usar las nuevas variables.
- ◆ Esta instrucción es una declaración, es decir, no se ejecuta.

- ◆ Esta instrucción no puede tener etiqueta.
- ◆ Las variables NOMV y las variables NOMN ocupan posiciones diferentes en memoria, y por lo tanto su contenido puede ser diferente.
- ◆ Se puede igualar varias veces a la misma variable definida.
- ◆ Esta instrucción se puede usar para definir acumuladores, áreas temporales en un programa, etc. Se recomienda definir por ejemplo, dentro de AREA de DBD una variable N1 de 0, N2 de 2, etc. y luego por esta instrucción igualar a N1 – 0 y N2 – 2 los acumuladores necesarios para un programa.

A continuación se presenta un ejemplo del uso de esta instrucción.

a) En DBD se define:

```
ARCHIVO CLTES 7 3
```

```
-----
```

```
NUM-CLI 'NUMERO DE CLIENTE' * R0 LL1 N1 0
```

```
NOM-CLI 'NOMBRE DEL CLIENTE' * R0 S X 30
```

```
-----
```

b) Se hace un programa para listar los clientes entre un cierto rango:

```
PROGRAMA 'LISTA CLIENTES'
```

```
- - - instrucciones para definir los encabezados
```

```
IGUAL CLTES NUM-CLI NUM-CLI-I NUM-CLI-F
```

```
* LEER RANGO
```

```
LEER TERMINAL EJECUTA 100 $
```

```
'CLIENTE INICIAL' NUM-CLI-I $
```

```
'CLIENTE FINAL' NUM-CLI-F
```

```
BUSCAR DISCO CLTES EJECUTA 100 APROX-G1 NUM-CLI-I
```

```
* REVISAR RANGO
```

```
10 LEER DISCO CLTES NUM-CLI
```

```
SI NUM-CLI > NUM-CLI-F EJECUTA 100
```

```
* LEER E IMPRIMIR LOS DATOS DEL CLIENTE
```

```
* TOMAR OTRO
```

```
BUSCAR DISCO CLTES EJECUTA 100 SEC-G1
```

EJECUTA 10

100 FIN

Los siguientes comentarios son aplicables al ejemplo:

- ◆ Al leer NUM-CLI-I y NUM-CLI-F se hará con las mismas características de validación que la variable original NUM-CLI.
- ◆ Note que al hacer el BUSCAR DISCO opción APROX.Gx se usó NUM-CLI-I.
- ◆ Como las variables definidas con IGUAL en CLS ocupan lugar en memoria diferente, se puede hacer la comparación de NUM-CLI con NUM-CLI-F.

Si no se hubiera usado ésta instrucción sería necesario definir ésta variable en la DBD con los problemas de mantenimiento al hacer cambios en la definición de NUM-CLI, hacerlos también en las demás variables.

7) VECTORES.

En **SISINF** es posible tener vector o variables con subíndice, tal como se tiene en otros lenguajes, pero éstos son de programa. Para explicar éste concepto suponga que al definir un archivo de clientes, en el registro maestro se tendrá entre otras variables el número del cliente NUM-CLI. Este variable se podrá usar en algunos programas sin subíndice y en otros con subíndice, pero en el archivo siempre será sin subíndice.

Es decir, los vectores en **SISINF** son de programa ya que todas las variables de los archivos son sin subíndice y en los programas dichas variables se podrán usar con o sin él.

Es muy importante usar variables con subíndices sólo cuando sea necesario ya que un uso indiscriminado hará que se tengan programas más complicados y que consuman más memoria y tiempo de proceso.

Se tienen dos tipos de vectores, uno que se aloja totalmente en la memoria del programa (RANGO) y por consecuencia es más rápido y otro (RANGO-EXTENDIDO) que aloja las primeras 512 palabras en memoria y lo demás en disco, que tiene la ventaja de poder ser más grande al no tener el límite de memoria.

7.1) Declaración de VECTOR.

Esta declaración sirve para definir que una variable del programa se usará en el mismo con subíndice. El formato general de la instrucción es:

```

VECTOR NOMV RANGO                NUM-I NUM-S
VECTOR NOMV RANGO-EXTENDIDO      NUM-I NUM-S

```

Los siguientes comentarios son aplicables:

- ◆ NOMV es el nombre de la variable que se desea usar con subíndice.
- ◆ NUM-I y NUM-S son dos números que indican el subíndice inferior y el superior que podrá tener la variable.
- ◆ Ambos deben estar entre - 9999 y + 9999 para RANGO, y entre - 99,999,999 y +99,999,999 en RANGO-EXTENDIDO, no tener decimales y NUM-I ser menor a NUM-S.

- ◆ Esta instrucción se deberá poner antes de usar la variable NOMV.
- ◆ La variable NOMV puede haber sido generado por el IGUAL de CLS. La declaración de IGUAL debe estar antes de la de VECTOR.
- ◆ La variable NOMV se deberá usar con subíndice en todo el programa, salvo cuando se usa como declarativa.
- ◆ En el caso de RANGO, se usa la memoria de ELS para alojar el vector.
- ◆ Para RANGO-EXTENDIDO, se tiene un área de memoria de 512 palabras para almacenar una parte del vector, el resto está en disco. Note que el uso de memoria es independiente del subíndice inferior y superior. Se deberá tener el área en disco para almacenar el o los vectores que se usen con ésta opción en forma simultánea. El archivo que se usa es TEMxyy. Si se accesa el vector en forma aleatoria la diferencia en tiempos de ejecución puede ser significativa.

Un ejemplo puede ser en un sistema de nóminas el caso de llevar un acumulado por departamento. Primero se deberá declarar la variable como vector con:

```
IGUAL AREA N22 ACUM
VECTOR ACUM RANGO 100 199
```

luego se puede usar así:

```
ACUM (DEPT) = ACUM (DEPT) + IMPORTE
```

Es decir, primero se declaró ACUM igualándola a una variable de AREA llamada N22. Luego se definió que se usaría ACUM con subíndice y que los valores que tomaría éste serían 100, 101, ... 199.

Un caso que se puede tener es que una variable en una parte del programa, por lógica del mismo, se necesita sin subíndice y en otra parte con subíndice. Para ello suponga la variable número de cliente (NUM-CLT) de un archivo CLTS. Entonces en aquellas partes del programa en donde se necesite sin subíndice se deberá usar NUM-CLT y para poder usarla con subíndice es necesario declarar:

```
IGUAL CLTS NUM-CLT NUM-CLT-V
VECTOR NUM-CLT-V RANGO 1 500
```

y el nombre de la variable usado como vector será NUM-CLT-V.

7.2) Subíndices.

El formato general de los subíndices es:

(VAR-N/NUM)

Los siguientes comentarios son aplicables a los subíndices:

- ◆ Se debe especificar la variable VAR-N o el número NUM entre paréntesis. Se deberá dejar al menos un espacio en blanco entre el paréntesis izquierdo y la variable o número, y entre la variable o número y el paréntesis derecho.
- ◆ Para propósitos de descripción de errores en CLS el subíndice cuenta como 1 campo.
- ◆ La variable VAR-N deberá ser N1, N2, N3 o N4.
- ◆ La variable VAR-N no puede haber sido declarada en VECTOR.
- ◆ No se revisa en compilación que el número (NUM) usado como subíndice esté en el rango especificado en la declaración de VECTOR.
- ◆ Si en ejecución la variable VAR-N o el número NUM usados como subíndice están fuera de rango, se cancelará el trabajo.
- ◆ En ejecución todos los elementos del vector están inicializados a ceros (N1, N2, N3, N4, F) o a blancos (X o S).
- ◆ Las variables de opciones de la instrucción OPCION TERMINAL AREA no aceptan variables con subíndices. Se dará error de compilación.
- ◆ Ciertas instrucciones NO aceptan la misma variable con subíndices en las diferentes variables de la instrucción. Se dará error de compilación.

8) DOCUMENTACION SISINF.

En **SISINF** se tiene un módulo de Documentación que sirve para hacer la documentación de programación que se puede usar en la etapa de desarrollo del sistema y que sirve mucho en la etapa de implantación del mismo, así como en la de mantenimiento. Básicamente se discute aquí los registros de *D y la declaración DOCUMENTACION para lograr lo anterior.

También se discute la opción EST-USO Estadística de Uso que forma parte del sistema de Contabilidad y uso de **SISINF**, y consiste en registrar cierta información de la ejecución de un programa con el propósito de revisar los consumos de CPU y accesos a disco del mismo para optimizar dicho programa.

Se sugiere revisar el módulo de documentación **SISINF**, así como el de contabilidad y uso de **SISINF**.

8.1) Registros con *D.

Se propone que la documentación de lo que hace cada programa en forma general así como cada una de las partes o rutinas del mismo se incluya en el programa fuente mediante el uso de registros de comentarios que se inicien con *D.

Así una buena política es que todos los programas se inicien con una descripción general de lo que hace el programa.

También que mediante *D se documenten las partes importantes de la lógica describiendo en forma general la forma de trabajo.

El módulo de documentación **SISINF** saca esta información de todos los programas que forman el sistema y junto con otra información de la DBD se forma el reporte de documentación de programación.

También es posible con este sistema consultar para un programa específico los registros de *D que se incluyeron.

8.2) DOCUMENTACION y Estadísticas de Uso.

El formato general de la declaración de documentación es:

DOCUMENTACION

NUM-S

NUM-G

Los siguientes comentarios son aplicables.

- ◆ Con la declaración de DOCUMENTACION se asigna el número del sistema (NUM-S) y el número del grupo (NUM-G) al que pertenece el programa.
- ◆ Se asume que en DBD se definió un sistema con número NUM-S y un grupo con número NUM-G.
- ◆ No se revisa que NUM-S y NUM-G existan en la DBD pero se dará diagnóstico al realizar el proceso de actualización del módulo de documentación y uso de **SISINF**.
- ◆ Si un programa no tiene esta declaración el programa se documentará en PROGRAMAS NO DOCUMENTADOS.

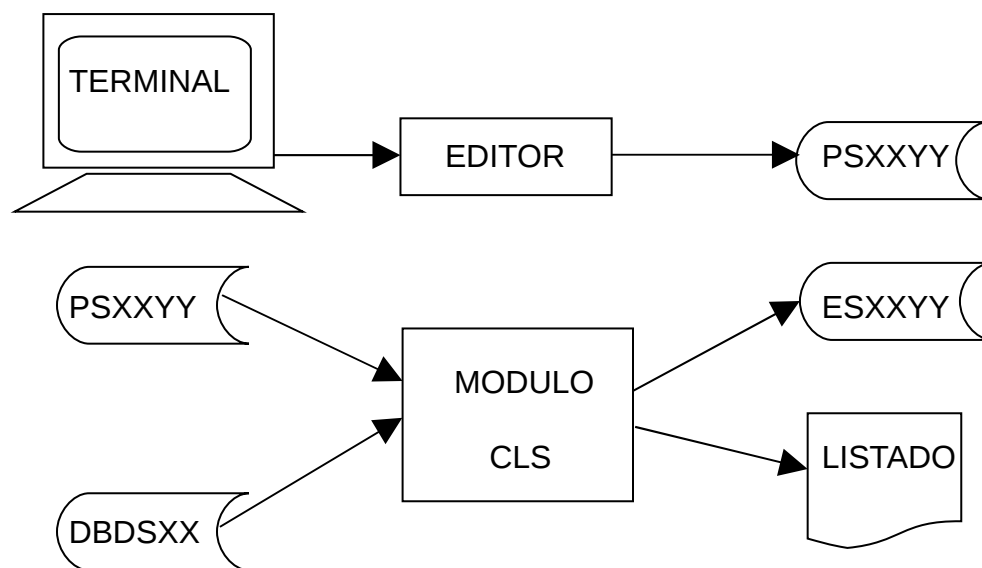
9) EJECUCION DE CLS.

9.1) Pasos para la ejecución.

Una vez que el usuario tenga su programa en el Lenguaje **SISINF** deberá efectuar los siguientes pasos:

- ♦ **CREAR** por instrucciones del sistema operativo el archivo PSxxyy en donde 'xx' son las iniciales de DBDSxx y 'yy' son las iniciales asignadas al programa.
- ♦ Con el **EDITOR** que se provee con la computadora, teclear los programas en el archivo PSxxyy.
- ♦ Ejecutar el módulo **CLS**, el cual revisa las instrucciones y si no hay errores genera el archivo Esxxyy, el cual es usado por el módulo **ELS** para poder ejecutar el programa.
- ♦ Si hay errores se deberán corregir a través del **EDITOR** y volver a ejecutar el módulo **CLS** hasta que no se tengan más errores.

Gráficamente la ejecución queda:



9.2) Datos para la ejecución.

La ejecución del módulo CLS es a través de la instrucción del sistema operativo en la que se pide la ejecución de un programa, ordinariamente RU, CLS o CLS. Al hacerlo, la pantalla aparece como:

SISINF	CLS	VX.X	MAQUINA	EMPRESA
CLAVE DE ACCESO			XXXXXXXXXX	
INICIALES DE DBD			XX	
DISPOSITIVOS DE DBD			XXXXXXXXXXXXXXXXXX	
INICIALES DE PROGRAMA			XX	
DISPOSITIVO DE PROGRAMA			XXXXXXXXXXXXXXXXXX	

El usuario deberá teclear a CLAVE DE ACCESO un blanco si es que no se tiene el sistema de claves o con la clave de acceso a los diferentes módulos de **SISINF** que le fue asignada por el departamento de sistemas.

A continuación deberá contestar a INICIALES DE DBD con las iniciales xx de su archivo de DBDSxx; a DISPOSITIVO (en algunas máquinas se llama DIRECTORIO, en otras GRUPO y CUENTA, etc.) deberá contestar con el dispositivo en donde se creó el archivo DBDSxx, si sólo oprime RETURN se le asigna un valor de acuerdo a cada marca y modelo de computadora; a INICIALES DE PROGRAMA con las iniciales yy que serán usadas para localizar el archivo PSxxyy. Por último, deberá responder a DISPOSITIVO DEL PROGRAMA con el DISPOSITIVO en donde se creó el archivo, PSxxyy. Si sólo se oprime la tecla de RETURN se asignará un valor de acuerdo a cada marca y modelo de computadora.

La clave de acceso puede tener valores prefijados para:

- ◆ Iniciales de DBD.
- ◆ Dispositivo de DBD.
- ◆ Dispositivo del programa.

evitando que se tecleen cada vez y sólo se tecleen las iniciales del programa que se compila.

A continuación aparecerá la hora de inicio de los 8 programas que integran el módulo CLS, el último borra la pantalla y ésta aparecerá:

SISINF	CLS	VX.X	XXXX	YYYY
INICIALES DE DBD			XX	
NUMERO DE ERRORES			XXXX	
MEMORIA			XXXX	xx% de uso de memoria
UNIDAD DE SALIDA (?/T/I)			X	
SE LISTA (?/P/E/V/A/M/X/C)			X	
PARA CANCELAR EL DESPLEGADO POR TERMINAL DAR / O \				
ESTA COMPILACION NO SE REGISTRA EN DOCUMENTACION				
SISINF POR ESTAR INACTIVA.				

La memoria usada indica cuántas palabras usa el programa para datos, instrucciones y archivos. Vea 9.3 para mas comentarios sobre la memoria.

El usuario debe revisar el NUMERO DE ERRORES que se tuvieron y con esto decidir las siguientes acciones a tomar.

Deberá contestar la UNIDAD DE SALIDA donde desea ver los listados, una T para terminal y una I para la Impresora. Si se tecléa ? se dará ayuda.

Se deberá contestar a SE LISTA con una o varias de las siguientes letras, si no se pone nada se asume P.

P Se lista programa PSxxyy y los ISzzzz y errores.

E Se listan los errores solamente.

- V Se listan las variables usadas.
- A Se listan los archivos, tablas usadas y rutinas.
- M Para listar la memoria usada por datos, instrucciones, archivos, ...
- X Se da una referencia cruzada de variables y etiquetas.
- C Para listar las instrucciones al compilador y las líneas que si quitan.

Para que el programador no necesita recordarlas puede dar ?.

Cuando se lista por la pantalla y ésta se llena, se deberá oprimir la tecla de espacio y luego RETURN para que se despliegue la siguiente. Si se da \ no se sigue listando más información y se termina el módulo.

Si se tiene las declaraciones de DOCUMENTACION pero ésta no está activada se da el mensaje que NO se está registrando esta compilación.

Cuando una línea tiene muchos errores y se llega a fin de pantalla o fin de hoja, la línea se repite en la nueva pantalla o en la nueva hoja.

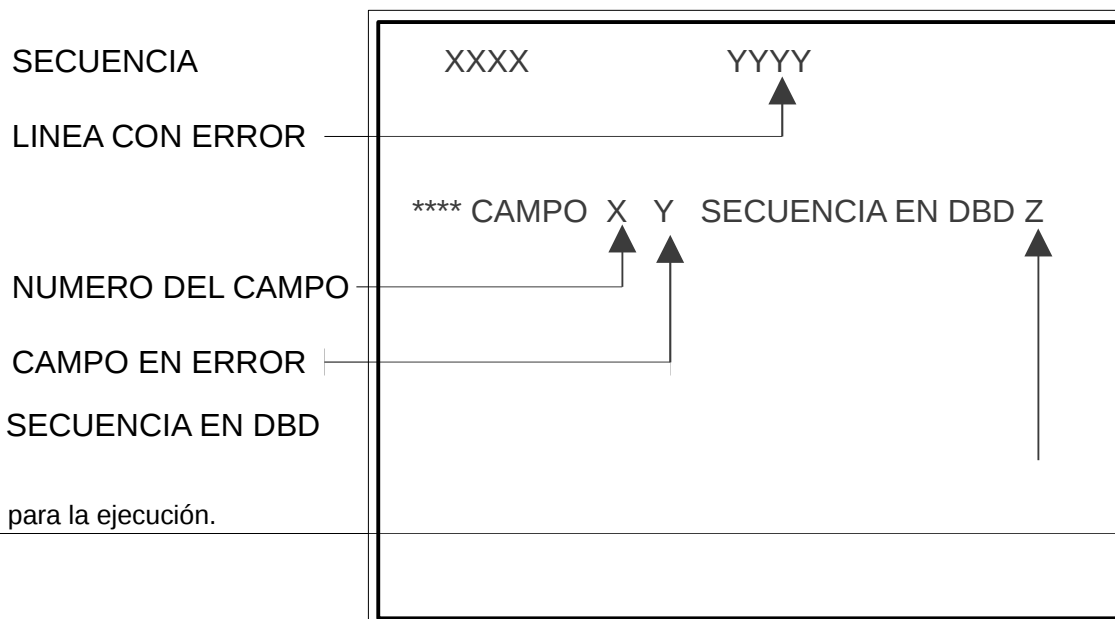
El siguiente es un ejemplo del formato para dar los diagnósticos:

```

0010   OPCION   TERMINAL LE.  NOM-CLI
****   CAMPO   03          LE.

        OPCION EN 'OPCION TERMINAL' NO ES VALIDA
    
```

El formato general es:



DE VARIABLE	_____	
DESCRIPCION DEL	→	XXXXXXXXXXXXXXXXXXXXXX
ERROR		XXXXXXXXXXXXXXXXXXXXXX

Para el número del campo se deberá contar de izquierda a derecha en la línea con error, y cada palabra, número o letrero entre apóstrofes se considera un campo. En algunos diagnósticos el número de campo no es exacto, sino aproximado. En otros, el error no pertenece a ninguna línea, como es el caso de exceder tablas internas. Cuando esto sucede el error aparecerá en la línea 1 del programa y con número de campo igual a cero.

La opción "E" es muy útil para quitar errores en sesiones de terminal en que se compila y se edita, se vuelve a compilar y editar y así sucesivamente hasta que ya no hay más errores.

Al final aparecerá el tiempo total usado en la ejecución de CLS desde el inicio hasta este letrero.

El módulo CLS despliega los diagnósticos de errores en archivos en el formato normal de **SISINF**, esto es:

OP = XX ERROR EN YYYYYY (ZZZZ)

Al efectuar la operación 'xx' en el archivo 'yyyyyy' ocurrió el error 'zzzz'.

Se deberá consultar el manual del sistema de archivos para revisar el tipo de error, que pueda ser falta de disco, muchos archivos abiertos, etc. Si no es un error normal se deberá consultar a personal de su Distribuidor.

9.3) Uso de Memoria.

En todas las versiones de **SISINF** se ha venido incrementando la memoria máxima de un programa, a partir de Versión 7.3 la memoria máxima es de 89,600 palabras. Anteriormente se tenía limitaciones de hardware o software en las diferentes máquinas, también se tenía la barrera de los 32K como límite. Por ello se cambió la forma de manejo de la memoria en los programas de **SISINF**.

A partir de la V5.1 las tablas de control y las tablas con la información de las áreas de terminal (ambas relacionadas con el proceso de Opción Terminal Area), son asignadas a segmentos independientes de memoria cuyos límites son de 32K palabras también, con la condición de que los buffers de CLAVE y TERMINAL (para instrucción con el mismo nombre) deben quedar contenidos dentro de los primeros 16K, por lo cual se cargan primero; Además el programa completo no puede exceder de 89,600 palabras; Si al compilar un programa, se rebasa alguno de estos límites, se diagnostica como "El programa se excede de Memoria".

En esta forma de trabajo el total de memoria usada por un programa, no indica que en algún segmento ya se va a agotar. Por lo anterior ahora además de este total de memoria usada se tiene un % de la memoria usada que refleja el segmento más usado. Así un programa de 20K palabras puede tener un 90% de memoria usada, indicando que en algún segmento solo queda un 10%; Ahora se necesita revisar ambos parámetros para saber el consumo de memoria.

Por razones de implantación de **SISINF** en las diferentes marcas y modelos de computadora un mismo programa puede usar diferente cantidad de memoria en diferentes máquinas. Por lo anterior, se recomienda no usar más de 80K palabras, con lo cual se puede garantizar (con pocas posibilidades de error) que se puede ejecutar en cualquier máquina).

En algunos casos un sistema sólo se usará en la computadora en que se está desarrollando, por lo que podrá usar más memoria. Cuando un programa exceda al límite de memoria permitido se podrá sacar con la opción "M", a la hora de compilarlo, el uso de memoria por diferentes conceptos.

Dichos conceptos son:

FIJOS. Estas palabras se usan por SISINF para los letreros de EJECUTANDO, LECTURA, ... e indicadores internos.

DATOS. Estas palabras se usan por las variables, letreros y constantes que se usen en el programa. Así por ejemplo. Cuando se usa la instrucción de LEER TERMINAL EJECUTA 10 NUM-CLT se usará memoria por el letrero que aparecerá en la pantalla, por la variable y si se especificó rango, por las constantes para validar dicho rango.

INSTRUCCIONES. Estas palabras se usan por el código interno que se genera para la ejecución de las instrucciones del programa.

M13-15. Estas palabras se usan para las tablas internas que se cargan si el programa tiene una o varias variables con máscaras M13 a M15.

M17-M18. Estas palabras se usan para las tablas internas que se cargan si el

programa tiene una o varias variables con máscaras M17 a M18.

IMPRESORA. Estas palabras se separan para el buffer de impresora si se tienen instrucciones de IMPRESION.

DETALLE. Estas palabras se separan como buffer para poder procesar la instrucción de DETALLE.

CLAVE. Estas palabras se separan como buffer para poder procesar la instrucción de OPCION TERMINAL CLAVE.

ORDEN. Estas palabras se separan como buffer para poder procesar la instrucción de OPCION TEMPORAL # ORDEN.

SUBPROGRAMAS. Estas palabras se separan como buffer de los parámetros en SUBPROGRAMAS.

LLAMAR. Estas palabras se separan para el área de comunicación y parámetros necesarios de la instrucción de LLAMAR.

TERMINAL. Esta área se separa como buffer para leer y escribir en la pantalla con la instrucción TERMINAL.

(Nombre de Tabla). Estas palabras se separan para poder procesar las instrucciones de BUSCAR TABLA o la validación por VALOR. Básicamente, se tiene una tabla interna y los valores de la tabla. Si en un programa sólo se usa la validación por valor, sólo se carga la primera columna de dicha tabla. Si se usa la instrucción de BUSCAR TABLA se carga todo.

(Nombre de Temporal). Estas palabras se separan para poder procesar las instrucciones del temporal cuyo nombre se indica. Se tienen tablas internas, áreas para el manejo de archivos del sistema operativo.

REG TEMP. Estas palabras se separan para cierto buffer común a todos los archivos temporales (sólo se tiene en algunas máquinas).

(Nombre del Permanente). Estas palabras se separan para poder procesar las instrucciones de dicho archivo permanente. Se tienen tablas internas para registros, para directorios, para el archivo, así como los buffers para cada recurrente que se use.

VECTOR xx. Estas palabras se separan para el vector xx, previamente se iguala xx a la variable.

TEXTO,... Estas palabras se una siempre que se tenga un Texto, Imagen o Audio.

AREA-XX YYYY. Estas palabras se separan para procesar el área con número YYYY.

TABLA DEL AREA. Estas palabras se usan por OPCION TERMINAL AREA para el proceso interno.

En general si un programa excede la memoria se deberá:

- 1) Si se exceden las 89,600 palabras tratar de revisar los conceptos anteriores para disminuir algunos de ellos.
- 2) Si se tiene instrucciones de TERMINAL, recordar que estos no pueden pasar de 16K.
- 3) El tamaño máximo de segmento es de 32K por lo que si se tienen muchas áreas, o muchos archivos, ... se puede exceder en algún segmento.

Para procesar la instrucción de BUSCAR RECURRENTE con opción ORD y OPCION DISCO ORD-Rx se separan unas tablas cuya memoria ya se incluyó en la del archivo permanente a que pertenece, pero el módulo ELS cuando se ejecuta esta instrucción, trata de usar el resto de la memoria disponible para hacer el ordenamiento en memoria. Como es posible que se tengan varias de estas instrucciones en diferentes recurrentes y archivos, el módulo CLS asigna el resto de la memoria disponible para el proceso de esta instrucción a cada uno de ellos y los reporta como:

(nombre del permanente) x mmmm nnnn

En donde 'x' es el número del recurrente, 'mmmm' es la memoria usada para procesar la opción, y 'nnnn' es la cantidad de recurrentes que se pueden ordenar en memoria.

Se recomienda cuando se use esta opción revisar 'nnnn' ya que si es menor que la cantidad de recurrentes que tiene la mayoría de registros, los ordenamientos se tendrán que hacer a través de archivos secuenciales y esto es ineficiente.

9.4) Reglas de Sintaxis.

Para especificar las instrucciones en el archivo Psxxyy se deberá seguir las siguientes reglas:

- ◆ Entre cada palabra, letrero o número se deberá dejar al menos un espacio en blanco.

- ◆ Los letreros se teclean entre apóstrofes y pueden tener blancos intermedios, los cuales se incluyen en el letrero.
- ◆ Si un letrero no cabe en una línea se puede terminar con \$ e iniciar con la siguiente línea con \$.
- ◆ Es posible definir letreros en la forma nnn”C en donde “nnn” es el número de veces que se repite el caracter C.
- ◆ Las palabras, letreros o números que se especifican para cada instrucción se pueden dar en cualquier columna, pero respetando el orden especificado en la explicación de dicha instrucción.
- ◆ En números, el signo + o – puede ir junto o separado del número. El signo + puede omitirse.
- ◆ Las columnas donde se especifica la información es de la 1 a la 70. Se diagnostica si se usan más columnas.
- ◆ Si una instrucción no cabe en una línea, se deberá poner un \$ después de la última palabra, letrero o número y continuar en la siguiente línea.
- ◆ Un * en la columna 1 significa comentario.
- ◆ Dentro de una instrucción lo que se especifique entre : y ; se asume comentario.

10) INSTRUCCIONES A CLS.

10.1) Para que sirven?

Cuando los programas de una aplicación se van a usar en varias instalaciones, es necesario hacer cambios para acoplarlos a las necesidades propias de cada instalación, esto ocasiona que se tengan varias copias de los programas. Para que sólo sea necesario tener una copia se tienen las instrucciones a DBD y a CLS.

Otro caso es cuando por propósitos de usar la misma rutina en varios programas se desea tener un solo archivo fuente.

Básicamente las instrucciones a la DBD y a CLS son las mismas, en este capítulo sólo se dan ejemplos de su aplicación, ya que en el capítulo 10 del Manual de DBD se proporciona una explicación de las mismas.

10.2) Ejemplo.

Suponga que un sistema esta diseñado para trabajar con varias empresas en los mismos archivos o bien con una sola empresa.

Entonces para el caso de varias empresas se tendrá un archivo de empresas con el número, nombre y otra información relevante al sistema. También en otros archivos se tendrá la llave de número de empresa.

En DBD se define una variable para el compilador que es la que decide la inclusión de varias empresas.

```
& VALOR  VARIAS-EMPRESAS  SI
```

Para quitar o poner en DBD el archivo de empresas se pondría:

```
& QUITAR ARC-EMP  SI  VARIAS-EMPRESAS = NO
ARCHIVO EMP
```

```
....
```

```
NOM-EMP 'NOMBRE DE EMPRESA' * R0 S X 50
```

```
...
```

```
& VALOR LET-VAR NOM-EMP
& FIN-QUITAR ARC-EMP
```

Con la lógica anterior se define la variable de compilador LET-VAR a NOM-EMP, la cual será necesario definirla al nombre que se usará para el caso de una sola empresa.

```
& QUITAR DEF SI VARIAS-EMPRESAS = SI
& VALOR LET-VAR 'La Comercial, S.A.'
& F DEF
```

En los programas con & QUITAR se deberá también quitar las referencias al archivo de empresas y en los encabezados se pondría:

```
OPCION IMPRIMIR ... $
&R LET. &LET-VAR ... $
```

con lo anterior se reemplazará la variable de compilador LET-VAR a NOM-EMP o al letrero 'La Comercial, S.A.' dependiendo del caso.

Un ejemplo de cómo quitar los programas del MENU que sirve para varias empresas sería:

```
OPCION TERMINAL AREA 100 /MENU $
...
...
'CLA3' 'LISTAR CATALOGO' $
&Q PROG-VE SI VARIAS-EMPRESAS = NO
    'CLIB1' 'ABM EMPRESAS' $
    'CLIB2' 'LISTAR EMPRESAS' $
&f PROG-VE
```

Es decir si VARIAS-EMPRESAS = SI, se incluyen en el MENU las opciones de varias empresas si vale NO se quitan.

El potencial de las instrucciones al compilador tanto DBD como CLS es muy grande pero se deberá planear para que quede sencillo.

11) FUNCIONAMIENTO DE CLS.

El módulo CLS es el que compila las instrucciones en el Lenguaje **SISINF**; consta de ocho partes y forma el archivo ejecutable de **SISINF**.

El archivo ejecutable que forma CLS, nombrado ESxxyy, contiene las instrucciones, letreros de impresión y terminal e información sobre archivos permanentes y temporales que usa el programa PSxxyy de forma que la ejecución de ELS sea más ágil.

Los archivos formados por DBD (ARCSxx, CAMSxx, DIRSxx, TIPSxx, VYTSxx, COMSxx y VCCSxx) son usados por el módulo de CLS, ya que ellos tienen la información de la DBD usada.

Adicional a estos archivos se usan unos archivos de trabajo, llamados S1xxyy a S7xxyy para el paso de información entre las diferentes etapas de ejecución de CLS. También se usa el archivo ERxxyy para los errores.

En algunas máquinas los archivos S1xxyy a S7xxyy y el archivo ERxxyy se integran en un solo llamado S1xxyy. Estos archivos se crean en donde está el archivo PSxxyy.

La ejecución de CLS está formada por ocho partes las cuales corresponden a cada tiempo que aparece en la ejecución de dicho módulo. Las funciones de cada una de esas ocho partes se describe a continuación:

La parte I pasa la información del archivo PSxxyy que está en formato libre a otro archivo en formato fijo. Revisa que las palabras clave del lenguaje sean las correctas, por ejemplo poner OPCION TERMINAL BOR. Dará diagnóstico por BOR.

Es importante señalar que las variables a DBD y CLS se leen aquí del archivo VCCSxx para cambiar o quitar las instrucciones.

La parte II revisa que las variable, archivos y tablas se hayan definido en la DBD; Si no están, se diagnostica en esta parte. Se procesan las áreas con nombre.

La parte III obtiene del CAMSxx la información referente a letreros rangos, etc., dependiendo de la instrucción y la pasa a otro archivo interno. También revisa que el formato de las instrucciones sea el correcto, por ejemplo, poner LEER TERMINAL NUM-CLT dará diagnóstico por faltar el "EJECUTA #".

La parte IV genera las instrucciones para la ejecución de ELS en un archivo interno y revisa errores de inconsistencia. Así poner MOVER NOM-CLT a "xx" dará diagnóstico por mover a una literal.

La parte V procesa las etiquetas y genera las tablas para OPCION TERMINAL AREA. Se generan los errores relacionados con las mismas.

La parte VI forma el archivo Esxxyy con la información de los archivos internos generados y con los de DBD. El orden es el siguiente:

- ◆ Cargar letreros fijos como por ejemplo EJECUTANDO, LECTURA.
- ◆ Cargar literales, constantes, iniciar las variables.
- ◆ Cargar las instrucciones que ejecutará ELS para hacer lo indicado en el PSxxyy.
- ◆ Cargar las tablas que usa el programa.
- ◆ Separar área para procesar impresión, terminal, ...
- ◆ Cargar tablas para procesar máscaras especiales.
- ◆ Formar tablas de control para vectores, temporales y permanentes.
- ◆ Carga las tablas de OPCION TERMINAL AREA.

La parte VII forma la información necesaria para la documentación **SISINF**.

La parte VIII es un listador ya sea en la impresora o en la terminal. Ciertas opciones como V o X necesitan de un "SORT" el cual se hace en memoria.

La parte I a V y la parte VII y VIII de CLS llevan a cabo las funciones de un **compilador** de cualquier lenguaje. La parte VI, las de un **cargador**, de forma que el archivo Esxxyy es una **imagen** del área de memoria de ELS en donde se cargan los programas del usuario.

Dada la forma de trabajo de CLS, con cualquier cambio de PSxxyy será necesario volver a compilarlo.

Dada la forma de trabajo de DBD y CLS en cualquier cambio a DBDSxx será necesario ejecutar el módulo DBD y luego ejecutar el módulo CLS para los programas afectados por el cambio.

12) VALORES MAXIMOS DE CLS.

1) Número de archivos permanentes por programa.	40
2) Número de tablas por programa.	40
3) Número de variables por programa.	600
4) Número de temporales por programa.	40
5) Longitud máxima de un temporal.	512 caracteres
6) Número de opciones SEG., máximo por programa.	20
7) Número máximo de variables en la instrucción de SUBPROGRAMA.	50
8) Número máximo de vectores.	50
9) Número máximo de rutinas.	50
10) Número máximo de áreas.	64
11) Número máximo de letreros y variables en un opción terminal Area.	100
12) Número máximo de variables en áreas de Terminal.	150
13) Número máximo de letreros y variables en un opción terminal Area tipo /COLUMNA contando su repetición en los grupos	400
14) Número máximo de etiquetas en un programa o rutina. Para las etiquetas se usa un lugar para definir y otro por cada referencia, también ciertas instrucciones generan "etiquetas" internas	2000

13) FORMAR IMAGEN.

La instrucción de FORMAR IMAGEN existe desde la versión 4.0 para computadoras personales y desde versión 7.1 se modificó para que pudiera ser usada en el módulo ELSI-EMulador en todas las máquinas donde se encuentra. Aunque la instrucción de la versión 4 es casi compatible con la actual, se recomienda cambiar a lo nuevo. Esta instrucción sirve para los siguientes propósitos:

- ◆ Construir imágenes con gráficas estadísticas (pastel, barras, tendencias, etc.)
- ◆ Manipular marcas de imágenes.
- ◆ Construir imágenes por medio de formas geométricas básicas (líneas, círculos, cuadros, etc.) y/o por medio de otras imágenes o partes de ellas.

Dado que la instrucción tiene muchas opciones pero su uso es para los propósitos mencionados, se describe la instrucción de acuerdo a estos casos.

Muchos datos que se pasan a la instrucción son valores numéricos, por ejemplo los colores en donde 0 es negro, 1 es azul, ... Para que el programador NO tenga que recordar estos valores y que el programa sea mas entendible el compilador acepta poner en la instrucción el nombre, así se puede poner 'NEGRO'. El compilador pone el número correspondiente.

En los ejemplos se verán también como desplegar la información con la instrucción de OPCION TERMINAL AREA, se asume que ya se leyó el manual respectivo.

13.1) Gráficas.

El Formato general de la instrucción de FORMAR IMAGEN para hacer gráficas es:

```
FORMAR IMAGEN VAR-IMA  
/GRAFICA-PASTEL /GRAFICA-BARRAS /GRAFICA-TENDENCIAS  
/GRAFICA-HISTOGRAMA  
(título (Subtítulo (LetreroX (LetreroY)))  
/COLOR Color-fondo (Color-letras Color-separador Color-título
```

Color-subtítulo Color-letreros)

/2D /3D /3DM /LETREROS /NO-LETREROS /VALORES /NO-VALORES

/LINEA-S /LINEA-N /HORAS

/SERIE Etiqueta Color (Efecto)

/DATO Valor Etiqueta Color (Efecto Marca)

/DEFINIR-COLOR NUM 'CRRRGGBBB-XXX'

/VIARIAS-INSTRUCCIONES /FIN-DE-INSTRUCCIONES /CONSTRUIR

Los siguientes comentarios son aplicables:

- ◆ Los tipos de gráfica que se manejan se indican con: /GRAFICA-PASTEL /GRAFICA-BARRAS /GRAFICA-TENDENCIAS /GRAFICA-HISTOGRAMA.
- ◆ Después del tipo de gráfica se debe de especificar el Título de la misma y en forma opcional el Subtítulo, el letrero que indique que son los valores en el eje X en LetreroX y lo mismo para el eje Y en LetreroY.
- ◆ Con la opción de /COLOR se definen algunos colores de la gráfica. Si se omite la opción se asume el color del fondo del área o en blanco para la impresora, y los demás se asumen dependiendo del fondo que tenga. Se puede especificar solo el fondo y los demás se asumen dependiendo de este. O bien especificar el fondo, que puede ser TRANSPARENTE y algunos de los otros colores, los que se omitan se asume 'NEGRO'.
- ◆ La gráfica se despliega en forma plana (/2D) o bien se le da un efecto de tres dimensiones (/3D) o se le da este efecto y se agregan unos marcos (/3DM).
- ◆ Se pone una leyenda con el significado de cada dato y su color (/LETREROS). O bien no se pone (/NO-LETREROS).
- ◆ Se pone el valor de cada dato (/VALORES) o no se pone (/NO-VALORES).
- ◆ Las líneas para delimitar las barras y tendencias son sencillas (/LINEA-S) o en negrita (/LINEA-N) mas ancha.
- ◆ La opción /HORAS solo aplicable a histograma y significa que el dato de 4 dígitos es una hora siendo los dos primeros las horas y los dos siguientes los minutos. Así 1830 es 18:30.
- ◆ Para dar cada dato se usa la opción de "/DATO Valor Etiqueta Color (Efecto Marca)" en donde Valor es una constante o variable N1, N2 N3 o N4 con o sin decimales, Etiqueta es el letrero para poner un nombre al valor que se esta graficando y con qué Color y Efecto. Si se quiere marcar

poner su número en marca. Esto último se explica en la sección 13.5.

- ◆ Cuando la gráfica tiene varias series de datos se define cada una con "/SERIE Etiqueta Color (Efecto)" en donde Etiqueta es el letrero del significado de la serie con su Color y Efecto.
- ◆ La opción de /SERIE solo se usa en gráficas de barras múltiple, tendencias e histograma.
- ◆ Cuando se especifica la primer /SERIE los /DATOS solo tienen Valor y Etiqueta, para la segunda, tercera, ... solo el Valor. Si se especifican los demás son ignorados.
- ◆ Los colores son números del 0 al 16 aunque se puede poner su nombre, con /DEFINIR-COLOR se define en NUM un número de color de 20 a 29 y a continuación se especifican 3 números de 0 a 255 con el rojo, verde y azul. Lo que se especifique en '-xxx' es solo para documentación.
- ◆ En muchos casos por razones de programación la gráfica se genera en varias instrucciones de FORMAR IMAGEN, entonces se debe poner en la primera /VARIAS-INSTRUCCIONES. En la última se debe poner /FIN-DE-INSTRUCCIONES
- ◆ La opción de /CONSTRUIR indica que ya se dio toda la información de la gráfica y que esta se genere.

La tabla de números y nombres de color es:

0	'NEGRO'
1	'AZUL'
2	'VERDE'
3	'CELESTE'
4	'ROJO'
5	'MAGENTA'
6	'CAFÉ'
7	'BLANCO'
8	'GRIS'
9	'AZUL-B'
10	'VERDE-B'
11	'CELESTE-B'
12	'ROJO-B'

- 13 'MAGENTA-B'
- 14 'AMARILLO'
- 15 'BLANCO-B'
- 16 'TRANSPARENTE'

La tabla de números y nombres de efectos es:

- 0 'LLENO'
- 1 'PUNTOS'
- 2 'LINEAS-HORIZONTALES'
- 3 'LINEAS-VERTICALES'
- 4 'DIAGONALES-DESCENDENTES'
- 5 'DIAGONALES-ASCENDENTES'
- 6 'ROMBOS'
- 7 'CUADRADOS'
- 8 'CIRCULOS'
- 9 'ZIG-ZAG'
- 51 'PUNTOS-GRANDES'
- 52 'HORIZONTALES-SEPARADAS'
- 53 'VERICALES-SEPARADAS'
- 54 'DESCENDENTES-SEPARADAS'
- 55 'ASCENDENTES-SEPARADAS'
- 56 'ROMBOS-GRANDES'
- 57 'CUADRADOS-GRANDES'
- 58 'CIRCULOS-GRANDES'
- 59 'ZIG-ZAG-GRANDE'

Ciertas opciones se fijan por omisión de acuerdo al tipo de gráfica de acuerdo a la siguiente tabla:

Pastel	Barras	Tendencias	Histograma
/LETRERO	/LETRERO	/LETRERO	
/2D	/3D	/2D	
/LINEA-S	/LINEA-S	/LINEA-N	/LINEA-S

En general los valores por omisión dan resultados adecuados pero si se desean cambiar se puede hacer.

A continuación se tienen ejemplos de algunos tipos de gráfica, así como los programas para hacerlas.

13.2) Gráfica de Pastel.

Suponga que en su empresa se dan cursos y Usted desea graficar la evaluación que se hace de los mismos. En dicha evaluación se pregunta "Como se impartió el curso" y los encuestados seleccionan entre deficiente, aceptable, bueno y excelente. Usted captura cada evaluación con lo cual se acumula para cada curso el resultado. Si se acumulan los de todos los cursos se tendrá el resultado global, asuma que el nombre de las variables es SUM-PA, SUM-PB, ...

La instrucción para formar la gráfica sería:

```
FORMAR IMAGEN GRAF-EST /GRAFICA-PASTEL $
  'Resultados de la evaluacion' /3D $
/DATO SUM-PA 'Deficiente' 'ROJO' $
/DATO SUM-PB 'Aceptable' 'AMARILLO' $
/DATO SUM-PC 'Bueno' 'GRIS' $
/DATO SUM-PD 'Excelente' 'VERDE' $
/CONSTRUIR
```

En donde GRAF-EST se definio en la DBD en AREA como:

```
GRAF-EST ** R0 S I
```

Como se puede concluir la generación de la imagen es muy sencilla, ahora se tiene el programa completo como PSSIA1:

```
PROGRAMA 'Ejemplo I-1 gráfica pastel'
*
* Este programa despliega una gráfica de pastel, se asume en
* este ejemplo que los datos de la gráfica son sobre los
* resultados de la evaluacion de un curso.
*
* Definir el area para desplegarla
*
OPCION TERMINAL AREA 10 /IMAGEN $
  /LONG. 18 76 /POS. 3 3 /MARCO /CONFIRMAR $
```

```

/TECLAS-DE-FUNCION 3 'Imprimir' $
/BN VI /COLOR BLANCO-B BLANCO-B NEGRO $
/CAMPOS 2 2 GRAF-EST /VAR. /ESC. /LONG. 72 /RENG. 15

```

*

* Se calculan los valores de la gráfica
 * (Aqui Usted pone su lógica para hacerlo)

*

```

IGUAL AREA N10 SUM-PA SUM-PB SUM-PC SUM-PD

```

```

SUM-PA = 1

```

```

SUM-PB = 5

```

```

SUM-PC = 3

```

```

SUM-PD = 6

```

*

* Formar la gráfica

*

```

FORMAR IMAGEN GRAF-EST /GRAFICA-PASTEL          $
  'Resultados de la evaluación'   /3D           $
/DATO SUM-PA 'Deficiente' 'ROJO'                $
/DATO SUM-PB 'Aceptable' 'AMARILLO'            $
/DATO SUM-PC 'Bueno' 'GRIS'                    $
/DATO SUM-PD 'Excelente' 'VERDE'              $
/CONSTRUIR

```

*

* Desplegarla

*

```

OPCION TERMINAL AREA 10 /DESPLEGAR
ESCRIBIR TERMINAL GRAF-EST
OPCION TERMINAL AREA 10 /TERMINAR

```

*

* Se imprime

*

```

IGUAL AREA N10 COND

```

```

SACAR PARAMETROS CONDICION-TERMINAR COND

```

```

SI COND <> 3 FIN 'SIA2'
OPCION IMPRIMIR /NO-REGLON-INICIAL $
/DF 'SUIZO' 16 -5 $
/ENC. 30 'gráfica de evaluaciones' /AR 1 /CEL $
/ENC. 1 70 /LHS
IMPRIMIR DETALLE 5 GRAF-EST /L 400 /R 200
FIN 'SIA2'

```

En su programa se tiene un AREA de IMAGEN, dicha área es muy parecida a las de REGLON con la diferencia que solo aceptan variables tipo imagen. El código sería:

```

OPCION TERMINAL AREA 10 /IMAGEN $
/LONG. 18 76 /POS. 3 3 /MARCO /CONFIRMAR $
/TECLAS-DE-FUNCION 3 'Imprimir' $
/BN VI /COLOR BLANCO-B BLANCO-B NEGRO $
/CAMPOS 2 2 GRAF-EST /VAR. /ESC. /LONG. 72 /RENG. 15

```

Para las variables imagen se necesita definir la longitud en columnas que ocupará la imagen y el número de renglones que ocupará dentro del área. Note que se usa la tecla de función 3 para imprimirla.

Como opción adicional se puede mandar a imprimir la gráfica para ello las instrucciones serían:

```

IGUAL AREA N10 COND
SACAR PARAMETROS CONDICION-TERMINAR COND
SI COND <> 3 EJECUTA 100
OPCION IMPRIMIR /NO-REGLON-INICIAL $
/DF 'SUIZO' 16 -5 $
/ENC. 30 'Gráfica de evaluaciones' /AR 1 /CEL $
/ENC. 1 70 /LHS
IMPRIMIR DETALLE 10 GRAF-EST /L 400 /R 200
OPCION IMPRIMIR
100 ....

```

En el programa se revisa si se dio la tecla de función 3 para pedir la impresión, se ejecuta OPCION IMPRIMIR, definiendo los tipo de fuentes y encabezados y luego en el IMPRIMIR DETALLE se pone la variable imagen con la gráfica. Se especifica la longitud de la misma en puntos en este caso 400 y el

total de puntos en los renglones 200. Por definición 72 puntos ocupan una pulgada. Estos puntos son los mismos que Usted usa al definir los tipo de fuentes a usar en los editores de texto. Internamente SISINF construye la gráfica de 400 por 200 puntos y la pasa a la impresora.

13.3) Gráfica de Barras.

Suponga el mismo caso de la evaluación pero ahora con gráfica de barras. La instrucción de FORMAR IMAGEN permite formar la gráfica con la ejecución de varias instrucciones y NO una sola como en el ejemplo anterior (PSSIA1), lo anterior permite que la lógica del programa primero calcule un valor a graficar y lo pase con FORMAR IMAGEN, luego otro y lo pase, ...

Paso 1 iniciar la gráfica. La instrucción es:

```
FORMAR IMAGEN GRAF-EST /GRAFICA-BARRAS          $
'Resultados de la evaluacion'                    $
/VARIAS-INSTRUCCIONES /VALORES /2D              $
/COLOR 'TRANSPARENTE' 'AZUL' 'NEGRO' 'ROJO-B'
```

Ahora es una gráfica de BARRAS, si Usted desea que aparezcan los valores que se grafican debe poner /VALORES.

También SISINF asume ciertos colores para hacer la gráfica, el fondo usa el mismo del área en que se despliega o el color BLANCO-B si se imprime. Otros colores de la gráfica los escoge también.

Si Usted usa la opción de /COLOR puede definir el color del fondo, el de las letras, el de las líneas de valores y el del título. Para que se use el color del área o el blanco brillante en impresión puede poner 'TRANSPARENTE' en color del fondo.

Aparece también la opción de /VARIAS-INSTRUCCIONES con lo cual se le indica que la gráfica se formará con varias instrucciones de FORMAR IMAGEN.

Paso 2, pasar primer dato después de calcularlo:

```
FORMAR IMAGEN GRAF-EST $
/DATO SUM-PA 'Deficiente' 'ROJO' 'CIRCULOS-GRANDES'
```

Note que también se agregó un efecto en la gráfica que es de 'CIRCULOS-GRANDES'.

Paso 3, pasar segundo dato después de calcularlo:

FORMAR IMAGEN GRAF-EST \$

/DATO SUM-PB 'Aceptable' 'AMARILLO' 'LLENO'

Por último se genera con:

FORMAR IMAGEN GRAF-EST /CONSTRUIR /FIN-DE-INSTRUCCIONES

El programa completo quedaria comp PSSIA2:

PROGRAMA 'Ejemplo I-2 gráfica barras simples'

*

* Este programa es igual al PSSIA1 pero se asume que los datos

* para la gráfica se van calculando de uno en uno

*

IGUAL AREA N10 SUM-PA SUM-PB SUM-PC SUM-PD

*

* Definir el area para desplegarla

*

OPCION TERMINAL AREA 10 /IMAGEN \$

/LONG. 18 76 /POS. 3 3 /MARCO /CONFIRMAR \$

/BN VI /COLOR BLANCO BLANCO NEGRO \$

/TECLAS-DE-FUNCION 3 'Imprimir' \$

/CAMPOS 2 2 GRAF-EST /VAR. /ESC. /LONG. 72 /RENG. 15

*

* Formar la gráfica

*

FORMAR IMAGEN GRAF-EST /GRAFICA-BARRAS \$

'Resultados de la evaluacion' \$

/VARIAS-INSTRUCCIONES /VALORES /2D \$

/COLOR 'TRANSPARENTE' 'AZUL' 'NEGRO' 'ROJO-B'

*

* Se calculan el valor del primer dato y se pasa

* (Aqui Usted pone su lógica para hacerlo)

*

SUM-PA = 1

FORMAR IMAGEN GRAF-EST \$

/DATO SUM-PA 'Deficiente' 'ROJO' 'CIRCULOS-GRANDES'

*

* Se calculan el valor del segundo y se pasa, ...

* (Aqui Usted pone su lógica para hacerlo)

*

SUM-PB = 5

FORMAR IMAGEN GRAF-EST \$

/DATO SUM-PB 'Aceptable' 'AMARILLO' 'LLENO'

SUM-PC = 3

FORMAR IMAGEN GRAF-EST \$

/DATO SUM-PC 'Bueno' 'GRIS' 'HORIZONTALES-SEPARADAS'

SUM-PD = 6

FORMAR IMAGEN GRAF-EST \$

/DATO SUM-PD 'Excelente' 'VERDE' 'DIAGONALES-ASCENDENTES'

*

* Se construye y se despliega

*

FORMAR IMAGEN GRAF-EST /CONSTRUIR /FIN-DE-INSTRUCCIONES

OPCION TERMINAL AREA 10 /DESPLEGAR

ESCRIBIR TERMINAL GRAF-EST

OPCION TERMINAL AREA 10 /TERMINAR

*

* Se imprime

*

IGUAL AREA N10 COND

SACAR PARAMETROS CONDICION-TERMINAR COND

SI COND <> 3 FIN 'SIA3'

OPCION IMPRIMIR /NO-REGLON-INICIAL \$

/DF 'SUIZO' 16 -5 \$

/ENC. 30 'gráfica de evaluaciones' /AR 1 /CEL \$

/ENC. 1 70 /LHS

IMPRIMIR DETALLE 5 GRAF-EST /L 400 /R 200

FIN 'SIA3'

13.4) Gráfica de Barras varias series.

Si se continúa con el ejemplo de las evaluaciones de los cursos, suponga que ahora Usted desea combinar los resultados de 2 trimestres en una solo gráfica. Es decir tiene varias SERIES de datos una para cada trimestre. Los pasos importantes son:

Se inicia la gráfica:

```
FORMAR IMAGEN GRAF-EST /GRAFICA-BARRAS $
'Resultados de la evaluacion' $
/VARIAS-INSTRUCCIONES
```

Se calculan los datos de la primera SERIE (en este caso primer trimestre) y se pasan:

```
FORMAR IMAGEN GRAF-EST $
/SERIE '1 Trimestre' 'GRIS' 'ROMBOS' $
/DATO SUM-PA 'Deficiente' $
/DATO SUM-PB 'Aceptable' $
/DATO SUM-PC 'Bueno' $
/DATO SUM-PD 'Excelente'
```

Note que se tiene la opción de /SERIE en donde se da el nombre de la misma, el color y el efecto (opcional). Ahora con /DATO solo se especifica el total y el letrero de lo que significa.

De la misma forma se pasan los datos de la segunda serie pero ahora con /DATO, solo se da el valor ya que el nombre es igual a la primera serie.

El programa completo PSSIA3 sería:

```
PROGRAMA 'Ejemplo I-3 gráfica varias barras'
*
* Este programa es igual al PSSIA2 pero se asume que los datos
* son de 2 períodos diferentes por lo que se tienen dos SERIES
*
IGUAL AREA N10 SUM-PA SUM-PB SUM-PC SUM-PD
*
```

* Definir el area para desplegarla

*

```
OPCION TERMINAL AREA 10 /IMAGEN          $
  /LONG. 18 76 /POS. 3 3 /MARCO /CONFIRMAR $
  /BN VI /COLOR BLANCO BLANCO NEGRO      $
  /TECLAS-DE-FUNCION 3 'Imprimir'        $
  /CAMPOS 2 2 GRAF-EST /VAR. /ESC. /LONG. 72 /RENG. 15
```

*

* Formar la gráfica

*

```
FORMAR IMAGEN GRAF-EST /GRAFICA-BARRAS  $
  'Resultados de la evaluacion'         $
  /VARIAS-INSTRUCCIONES
```

*

* Se calculan los valores de la primera serie

* (Aqui Usted pone su lógica para hacerlo)

*

SUM-PA = 2

SUM-PB = 4

SUM-PC = 4

SUM-PD = 3

```
FORMAR IMAGEN GRAF-EST          $
  /SERIE '1 Trimestre' 'GRIS' 'ROMBOS' $
  /DATO SUM-PA 'Deficiente'         $
  /DATO SUM-PB 'Aceptable'          $
  /DATO SUM-PC 'Bueno'              $
  /DATO SUM-PD 'Excelente'
```

*

* Se calculan los valores de la segunda serie

* (Aqui Usted pone su lógica para hacerlo)

*

SUM-PA = 1

SUM-PB = 5

SUM-PC = 3

SUM-PD = 6

FORMAR IMAGEN GRAF-EST \$

/SERIE '2 Trimestre' 'AZUL' \$

/DATO SUM-PA /DATO SUM-PB /DATO SUM-PC /DATO SUM-PD

*

* Se construye y se despliega

*

FORMAR IMAGEN GRAF-EST /CONSTRUIR /FIN-DE-INSTRUCCIONES

OPCION TERMINAL AREA 10 /DESPLEGAR

ESCRIBIR TERMINAL GRAF-EST

OPCION TERMINAL AREA 10 /TERMINAR

*

* Se imprime

*

IGUAL AREA N10 COND

SACAR PARAMETROS CONDICION-TERMINAR COND

SI COND <> 3 FIN 'SIA4'

OPCION IMPRIMIR /NO-REGLON-INICIAL \$

/DF 'SUIZO' 16 -5 \$

/ENC. 30 'gráfica de evaluaciones' /AR 1 /CEL \$

/ENC. 1 70 /LHS

IMPRIMIR DETALLE 5 GRAF-EST /L 400 /R 200

FIN 'SIA4'

Este programa se puede cambiar y poner /GRAFICA-TENDENCIAS en lugar de /GRAFICA-BARRAS y así formar el PSSIA4.

13.5) Gráficas Marcadas.

En SISINF se tiene el concepto de imágenes marcadas, que permite que el usuario profundice en la información. Para ejemplificar estos conceptos se tiene el programa PSSIA6 en el que se asume que una empresa tiene 3 plantas y se desea graficar en un sistema de mantenimiento de las plantas, el total de averías que ocurren en cada una de ellas, así mismo se pueda seleccionar una

de las plantas y desplegar las averías por trimestre. El programa sería:

```
PROGRAMA 'Ejemplo I-6 gráfica marcada'
```

```
*
```

```
* Este programa es para ilustrar gráficas marcadas, para ello
* se asume que una empresa tiene 3 plantas y se desea graficar
* en un sistema de mantenimiento las averías que ocurren en cada
* una de ellas, así mismo se pueda seleccionar una de las plantas
* y desplegar las averías por trimestre
```

```
*
```

```
IGUAL AREA GRAF-EST GRAF-MES
```

```
IGUAL AREA N10 NUM-AVE-T1 NUM-AVE-T2 NUM-AVE-T3 $
      NUM-AVE-T4 NUM-AVE COL-PLA SUB
```

```
IGUAL AREA X20 NOM-PLA
```

```
VECTOR NUM-AVE-T1 RANGO 1 3
```

```
VECTOR NUM-AVE-T2 RANGO 1 3
```

```
VECTOR NUM-AVE-T3 RANGO 1 3
```

```
VECTOR NUM-AVE-T4 RANGO 1 3
```

```
VECTOR NOM-PLA RANGO 1 3
```

```
VECTOR COL-PLA RANGO 1 3
```

```
*
```

```
* Se definen los nombres de las plantas
```

```
*
```

```
MOVER 'Monterrey' A NOM-PLA ( 1 )
```

```
MOVER 'Puebla' A NOM-PLA ( 2 )
```

```
MOVER 'Guadalajara' A NOM-PLA ( 3 )
```

```
*
```

```
* Se definen los colores
```

```
*
```

```
COL-PLA ( 1 ) = 1
```

```
COL-PLA ( 2 ) = 4
```

```
COL-PLA ( 3 ) = 8
```

```
*
```

```
* Definir las áreas para desplegar las gráficas
```

*

OPCION TERMINAL AREA 10 /MENU /SELECCIONAR-MARCA \$
 /LONG. 18 34 /POS. 3 2 /MARCO /VARIAS-SELECCIONES \$
 /BN VI /COLOR BLANCO-B BLANCO-B NEGRO \$
 /TECLAS-DE-FUNCION 2 'Terminar' /TIEMPO 0 0 5 \$
 /CAMPOS 'Seleccione la planta' /ENC. \$
 GRAF-EST /VAR-OP. SUB /VAR-SEL.

*

OPCION TERMINAL AREA 12 /IMAGEN \$
 /LONG. 18 34 /POS. 3 42 /MARCO /CONFIRMAR \$
 /BN VI /COLOR BLANCO-B BLANCO-B NEGRO \$
 /CAMPOS 2 2 GRAF-MES /VAR. /ESC. /LONG. 24 /RENG. 15

*

* Formar la gráfica

*

FORMAR IMAGEN GRAF-EST /GRAFICA-BARRAS \$
 ' ' 'Planta' 'Avería' \$
 /VARIAS-INSTRUCCIONES /VALORES /2D

*

* Se calculan los valores a graficar o sea el número de averías en
 * cada planta para cada trimestre

* (Aqui Usted pone su lógica para hacerlo)

*

NUM-AVE-T1 (1) = 4

NUM-AVE-T1 (2) = 3

NUM-AVE-T1 (3) = 6

NUM-AVE-T2 (1) = 2

NUM-AVE-T2 (2) = 3

NUM-AVE-T2 (3) = 1

NUM-AVE-T3 (1) = 5

NUM-AVE-T3 (2) = 7

NUM-AVE-T3 (3) = 9

NUM-AVE-T4 (1) = 2

NUM-AVE-T4 (2) = 3

NUM-AVE-T4 (3) = 1

*

* Pasar los valores a la gráfica

*

SUB = 1

10 NUM-AVE = NUM-AVE-T1 (SUB) + NUM-AVE-T2 (SUB) \$
+ NUM-AVE-T3 (SUB) + NUM-AVE-T4 (SUB)

FORMAR IMAGEN GRAF-EST /DATO NUM-AVE NOM-PLA (SUB) \$
COL-PLA (SUB) 'LLENO' SUB

SUB = SUB + 1

SI SUB <= 3 EJECUTA 10

*

* Se construye, se despliega y se lee cual se desea.

*

FORMAR IMAGEN GRAF-EST /CONSTRUIR /FIN-DE-INSTRUCCIONES
OPCION TERMINAL AREA 10 /DESPLEGAR
ESCRIBIR TERMINAL GRAF-EST

20 LEER TERMINAL EJECUTA /F2 30 SUB

*

* Formar la gráfica por trimestre de la planta seleccionada

*

FORMAR IMAGEN GRAF-MES /GRAFICA-BARRAS \$
'Averias por mes' NOM-PLA (SUB) 'Trimestre' 'Averias' \$
/3D /VALORES \$
/DATO NUM-AVE-T1 (SUB) '1' 'NEGRO' \$
/DATO NUM-AVE-T2 (SUB) '2' 'GRIS' \$
/DATO NUM-AVE-T3 (SUB) '3' 'VERDE' \$
/DATO NUM-AVE-T4 (SUB) '4' 'AZUL' \$
/CONSTRUIR

*

* Desplegar la gráfica trimestral, regresar a leer otra

*

OPCION TERMINAL AREA 12 /DESPLEGAR /SALVAR-ANTERIORES \$
/EMPALMAR

ESCRIBIR TERMINAL GRAF-MES

OPCION TERMINAL AREA 12 /TERMINAR

EJECUTA 20

30 FIN

A continuación se explican algunos conceptos usados en el programa. Se definen dos áreas de terminal, una tal como ya se había hecho en los programas anteriores y otra de menú que queda como:

OPCION TERMINAL AREA 10 /MENU /SELECCIONAR-MARCA \$
/LONG. 18 34 /POS. 3 2 /MARCO /VARIAS-SELECCIONES \$
/BN VI /COLOR BLANCO-B BLANCO-B NEGRO \$
/TECLAS-DE-FUNCION 2 'Terminar' /TIEMPO 0 0 5 \$
/CAMPOS 'Seleccione la planta' /ENC. \$
GRAF-EST /VAR-OP. SUB /VAR-SEL.

SISINF al detectar que es área de /MENU pero que tiene /SELECCIONAR-MARCA asume que se va a leer de una imagen marcada aquella que seleccione el usuario. Note como esta área es como la de MENU con variables pero sólo acepta una variable del tipo imagen. En SUB se leerá el número de marca seleccionado por el usuario.

También es importante señalar que se puso /TIEMPO 0 0 5 con lo cual se indica que si el ratón se queda posicionado en una marca por más de 5 segundos, ésta se asume que es la seleccionada. Con SACAR PARAMETROS SELECCIONADO var-n1-0 se puede saber si dio botón izquierdo o fue por tiempo. Si no se pone esta opción siempre tendrá que dar botón izquierdo del ratón para seleccionar.

En el programa se tienen los vectores NUM-AVE-T1, NUM-AVE-T2, ... que son el número de averías por cada trimestre, el subíndice es el número de la planta. Estos valores se asignan, pero en un caso real se deberán de calcular de archivos.

La gráfica se forma en /VARIAS-INSTRUCCIONES y la lógica para cada dato es:

SUB = 1

10 NUM-AVE = NUM-AVE-T1 (SUB) + NUM-AVE-T2 (SUB) \$
+ NUM-AVE-T3 (SUB) + NUM-AVE-T4 (SUB)

```

FORMAR IMAGEN GRAF-EST /DATO NUM-AVE NOM-PLA ( SUB ) $
          COL-PLA ( SUB ) 'LLENO' SUB
SUB = SUB + 1
SI SUB <= 3 EJECUTA 10

```

Con la variable SUB se controla el número de cada planta, primero se calcula el total de averías sumando las de cada trimestre previamente calculadas dejándolo en NUM-AVE y luego se pasan a la gráfica.

La opción de /DATO tiene el valor, el nombre de la planta, el color, el efecto y un último parámetro que es el número asignado a esta barra o sea su número de marca que en este caso es el número de la planta.

La gráfica se genera como las anteriores pero ahora cada DATO se asocia a un número de marca.

Para la construcción, el desplegado y leer selección, las instrucciones para hacerlo son:

```

FORMAR IMAGEN GRAF-EST /CONSTRUIR          $
      /FIN-DE-INSTRUCCIONES
OPCION TERMINAL AREA 10 /DESPLEGAR
ESCRIBIR TERMINAL GRAF-EST
20 LEER TERMINAL EJECUTA /F2 30 SUB

```

Como se ve son parecidas al proceso de menú con variables. En este caso SISINF construye la gráfica con las dimensiones del área 10 y cuando el usuario selecciona con el ratón una de las barras regresa en SUB el número de la marca, que para este ejemplo es el número de la planta.

Con esto se genera otra gráfica con las instrucciones ya vistas con las averías de cada planta.

Como una extensión a este programa, esta última gráfica podría ser también marcada, bastaría con cambiar el área 12 a MENU y agregar en la de /DATO el número de marca que puede ser el número del trimestre. Con esto se puede proceder a desplegar datos de las averías de este trimestre.

13.6) Imágenes marcadas.

En SISINF también una imagen leída como puede ser un mapa se puede marcar, en este caso estas marcas son visibles y tienen diferentes formas, dimensiones y colores. La instrucción de FORMAR IMAGEN sirve para manipular dichas marcas, las opciones de marcas son:

/SACAR-MARCA	Número
/LEER-MARCA	Número Tipo Dimensión Color-fondo Color-marco
/CAMBIAR-MARCA	Número Tipo Dimensión Color-fondo Color-marco
/BORRAR-MARCA	Número
/BORRAR-MARCAS	
/INCLUIR-MARCA	Número
/INCLUIR-MARCAS	
/ERROR	Etiqueta
/OPCIONES-MARCA	Número Letrero

Los siguientes comentarios son aplicables:

- ◆ Todas las variables usadas para número de marca, tipo, ... deben ser N1 de cero decimales.
- ◆ Con /SACAR-MARCA se pueden obtener los números de marca que tiene la imagen. Primero se pone en el número un valor de 0 para indicar que se busque desde el inicio, al ejecutarse almacena en la variable el número de la primera marca. A volver a ejecutar, busca desde la primera marca la segunda y regresa su número en la variable, ... Poner la opción de /ERROR para que al final salga por esta opción al no haber más marcas.
- ◆ Con la opción de /LEER-MARCA especificando el número de la misma se almacena sus características en Tipo, Dimensión, Color-fondo y Color-marco.
- ◆ Con la opción de /CAMBIAR-MARCA especificando el número de la misma se cambian sus características con los valores de Tipo, Dimensión, Color-fondo y Color-marco.
- ◆ Se puede borrar una marca con la opción de /BORRAR-MARCA y el número de la misma.
- ◆ Se pueden borrar todas las marcas que tiene la imagen con la opción /BORRAR-MARCAS.

- ◆ Para desplegar o imprimir una imagen marcada es necesario pedir que se incluyan todas las marcas en la imagen que se despliega o imprime con la opción /INCLUIR-MARCAS, o bien se puede incluir una o varias con /INCLUIR-MARCA y su número.
- ◆ En las opciones de marcas si la marca no existe se diagnosticará con cancelación del programa. Para evitar esto, se puede poner la opción de /ERROR y la etiqueta de la instrucción a ejecutar en este caso.
- ◆ Con la opción /OPCIONES-MARCA se puede asignar a una marca un Letrero mismo que será desplegado cuando el "mouse" pase sobre la marca.
- ◆ Si una imagen se va a modificar usando OPCION TERMINAL AREA, NO se acepta que se ejecuten las opciones de /INCLUIR-MARCA, /INCLUIR-MARCAS, /OPCIONES-MARCA, las opciones de gráficas y figuras. Se da error de cancelación.

Los número de marca son de 1 a 9999

Los diferentes tipos de marca son:

- 0 'CUADRO '
- 1 'TRIANGULO'
- 2 'FLECHA-DERECHA'
- 3 'FLECHA-ARRIBA'
- 4 'FLECHA-ABAJO'
- 5 'FLECHA-IZQUIERDA'
- 6 'ROMBO'
- 7 'ASTERISCO'
- 8 'CIRCULO'
- 9 'CRUZ'
- 10 'ESTRELLA'

Los valores permitidos de la dimensión son de 3 a 50.

El siguiente programa ilustra el manejo de marcas:

PROGRAMA 'Ejemplo I-7 Imágenes con Marcas'

*

* Este programa hace lo siguiente:

* -Lee una imagen

* -Agrega marcas

* -Selecciona una de ellas y la modifica

* -Despliega la imagen

*

IGUAL AREA N10 MARCA-NUM MARCA-TIPO MARCA-DIM \$
 MARCA-CF MARCA-CM SUB

*

OPCION TERMINAL AREA 10 /IMAGEN \$
 /LONG. 18 40 /POS. 3 20 /MARCO \$
 /BN VI /COLOR BLANCO-B BLANCO-B NEGRO /CONFIRMAR \$
 /AGREGAR-MARCA MARCA-NUM MARCA-TIPO MARCA-DIM \$
 MARCA-CF MARCA-CM \$
 /CAMPOS 2 2 IMA-MAPA /VAR. /LEC. /MOD-1 \$
 /LONG. 30 /RENG. 15

*

OPCION TERMINAL AREA 12 /MENU /SELECCIONAR-MARCA \$
 /LONG. 18 40 /POS. 3 20 /MARCO /VARIAS-SELECCIONES \$
 /BN VI /COLOR BLANCO-B BLANCO-B NEGRO \$
 /TECLAS-DE-FUNCION 2 'Terminar' \$
 /CAMPOS 'Seleccione la marca' /ENC. \$
 IMA-MAPA /VAR-OP. SUB /VAR-SEL.

*

* Leer el mapa

*

OPCION TERMINAL AREA 10 /DESPLEGAR /USO-PRIMARIO
 LEER TERMINAL EJECUTA 100 IMA-MAPA
 OPCION TERMINAL AREA 10 /PROCESADA /CANCELAR-CONFIRMAR
 OPCION TERMINAL AREA 10 /TERMINAR

*

* Leer las marcas

*

MARCA-NUM = 0

```
MARCA-TIPO = 0
MARCA-DIM = 10
MARCA-CF = 8
MARCA-CM = 0
10  MARCA-NUM = MARCA-NUM + 1
    OPCION TERMINAL AREA 10 /DESPLEGAR /USO-1
    ESCRIBIR TERMINAL IMA-MAPA
    LEER TERMINAL EJECUTA 20 IMA-MAPA
    OPCION TERMINAL AREA 10 /PROCESADA /CANCELAR-CONFIRMAR
    OPCION TERMINAL AREA 10 /TERMINAR
    EJECUTA 10
*
*  Poner letrero a marcas
*
20  MARCA-NUM = 0
22  FORMAR IMAGEN IMA-MAPA /SACAR-MARCA MARCA-NUM /ERROR 24
    MOVER 'Planta #' A X20
    MOVER MARCA-NUM A X20 /AGREGAR
    FORMAR IMAGEN IMA-MAPA /OPCIONES-MARCA MARCA-NUM X20
    EJECUTA 22
*
*  seleccionar una marca
*
24  OPCION TERMINAL AREA 12 /DESPLEGAR
    ESCRIBIR TERMINAL IMA-MAPA
    LEER TERMINAL EJECUTA 30 SUB
    OPCION TERMINAL AREA 12 /TERMINAR
*
*  Cambiar la marca
*
    FORMAR IMAGEN IMA-MAPA /LEER-MARCA SUB MARCA-TIPO  $
        MARCA-DIM MARCA-CF MARCA-CM
    MARCA-CF = 14
```

FORMAR IMAGEN IMA-MAPA /CAMBIAR-MARCA SUB MARCA-TIPO \$
 MARCA-DIM MARCA-CF MARCA-CM

EJECUTA 20

*

* Desplegar el mapa

*

IGUAL MAPAS IMA-MAPA IMA-MAPA-MARCADO

30 MOVER IMA-MAPA A IMA-MAPA-MARCADO

FORMAR IMAGEN IMA-MAPA-MARCADO /INCLUIR-MARCAS

OPCION IMPRIMIR /NO-REGLON-INICIAL /ARCHIVO /BORRAR \$
 /DF 'SUIZO' 16 -5 \$

/ENC. 30 'Mapa con marcas' /AR 1 /CEL \$

/ENC. 1 70 /LHS

IMPRIMIR DETALLE 5 IMA-MAPA-MARCADO /L 400 /R 200

OPCION IMPRIMIR /DESPLEGAR-ARCHIVO ' '

100 FIN

Los siguientes comentarios son aplicables a este programa:

- ◆ El AREA 10 se despliega con el /USO-PRIMARIO para leer la imagen del mapa.

- ◆ En el AREA 10 tiene la opción

/AGREGAR-MARCA MARCA-NUM MARCA-TIPO MARCA-DIM \$
 MARCA-CF MARCA-CM \$

Al desplegar el AREA 10 con el /USO-1 se usa esta opción y sirve para agregar marca por marca a la imagen al modificarla cada vez. El usuario solo da "click" con el "mouse" en el punto donde se desea. Previo se movió a las variables de la opción los datos de la marca. Otra opción es poner /MARCAR en el AREA tanto en la lectura como en la modificación, con esto se abre un diálogo para hacerlo.

- ◆ Después de leer las marcas, se pone un letrero a cada marca, para ello se usa la opción SACAR-MARCA para conocer los números de marca y la de /OPCIONES-MARCA para poner los letreros.
- ◆ Una vez marcada la imagen se procede a usar el AREA 12 que es un /MENU pero de marcas, la lógica es que cada vez que se selecciona una

marca se le cambia su color por '14=rojo' y se pide volver a seleccionar una marca.

- ◆ Al dar F2 se procede a desplegar la imagen marcada, para ello se mueve a otra variable imagen, se pide que se incluyan las marcas en la misma y se despliega.

Una aplicación real puede ser marcar el mapa de un país con las plantas productivas que tiene una empresa o bien en el mapa de una ciudad marcar la localización de diferentes tiendas de una compañía. Este programa se dividiría en tres, el primero para leer el mapa y guardarlo en un archivo permanente, el segundo para agregar o borrar marcas y un tercero para que seleccionando una marca de información, despliegue información sobre el significado de la misma. También cambiando el tipo, las dimensiones o los colores se puede usar para indicar cierta información. Así el color se puede usar para saber que tienda vende más y cual menos. En esta forma de trabajo con solo desplegar para hacer la selección ya se dio más información como sería que tienda vende más.

13.7) Figuras y Texto.

A continuación se tiene un programa que contiene una rutina para desplegar un mes del calendario y que el usuario pueda seleccionar un día de dicho mes. Sus usos pueden ser varios, un ejemplo sería en el caso de un "Sistema de reservaciones" en que mediante colores se pueda desplegar la disponibilidad que se tiene. El programa es el siguiente:

```

PROGRAMA 'Calendario'
IGUAL AREA N10 DIA-SEL
IGUAL AREA FECHA FECHA-DESP
SACAR PARAMETROS FECHA FECHA-DESP
10  EJECUTA RUTINA CALENDARIO
SI DIA-SEL = 0 FIN
OPCION TERMINAL AREA 1 /DESPLEGAR
ESCRIBIR TERMINAL FECHA-DESP
OPCION TERMINAL AREA 1 /TERMINAR
EJECUTA 10
*
OPCION TERMINAL AREA 1 /REGLON /POS. 10 20 /LONG. 3 40 $
/MARCO /CONFIRMAR /COLOR BLANCO BLANCO NEGRO $

```

/CAMPOS 'Fecha seleccionada' FECHA-DESP /VAR. /ESC.

*

RUTINA CALENDARIO

*

* Esta rutina sirve para desplegar un calendario y que el
* usuario pueda seleccionar una fecha si así lo desea.

*

IGUAL AREA GRAF-EST FIGURA-BASE FIGURA-MES

IGUAL AREA N10 DIM-X DIM-Y SEPX SEPY SEPX2 SEPY2 SEPY4 \$
X Y YL1 YL2 YL3 XT1 XT2 XT3 YT1 YT2 YT3 XR0 YR0 TRX TRY \$
LA LX I J XRM

IGUAL AREA X2 XDIAS

IGUAL AREA X3 XMES

IGUAL AREA X4 XAN

IGUAL AREA FECHA FECHA-BASE FECHA-1DM FECHA-1DPM FECHA-PRO

IGUAL AREA N10 DIF-D DIF-M DIF-A SEMANA NUM-DIA NUM-MES \$
DIA-PRO

VECTOR X RANGO 1 8

VECTOR Y RANGO 1 6

*

* Área para desplegar el calendario

*

OPCION TERMINAL AREA 8000 /MENU /POS. 2 10 /LONG. 20 60 \$

/COLOR GRIS GRIS NEGRO \$

/TECLAS-DE-FUNCION 2 'Terminar' \$

/SELECCIONAR-MARCA /CAMPOS \$

FIGURA-MES DIA-SEL /VAR-SEL.

*

* Formar figura base

*

FORMAR IMAGEN FIGURA-BASE /X-Y-NORMAL \$

/INICIO 0 0 'TRANSPARENTE' /VARIAS-INSTRUCCIONES \$

/MEDIDA-AREA 8000 DIM-X DIM-Y

*

* Dibujar un rectángulo en el área de imagen para que sirva
* de marco al área.

*

FORMAR IMAGEN FIGURA-BASE \$

/EFECTO-LINEAS 'LLENO' 50 \$

/RECTANGULO 0 0 DIM-X DIM-Y 'AZUL' 'LLENO' 'TRANSPARENTE' \$

/NO-OPCIONES-FIGURAS

*

* Dividir el eje Y en 9 renglones y calcular la Yp de cada uno.

* Se usará 2 para el título, 1 para el letreros de "Do Lu ..."

* y 6 para poner los días del mes Se dejan un espaciado de

* 5 arriba y abajo, 4 mas para las 2 líneas

*

SEPY = DIM-Y - 14 / 9

SEPY2 = SEPY / 2

SEPY4 = SEPY2 / 2

YL1 = DIM-Y - SEPY - 5

YL2 = YL1 - SEPY

YL3 = 6 * SEPY + 7

Y (1) = 5 * SEPY + 5

Y (2) = 4 * SEPY + 5

Y (3) = 3 * SEPY + 5

Y (4) = 2 * SEPY + 5

Y (5) = SEPY + 5

Y (6) = 5

*

* Dividir el eje X en 7 partes una para cada día de la semana

* menos un espaciado de 10 a ambos lados

*

SEPX = DIM-X - 20 / 7

SEPX2 = SEPX / 2

X (1) = 10

$$X(2) = X(1) + \text{SEPX}$$

$$X(3) = X(2) + \text{SEPX}$$

$$X(4) = X(3) + \text{SEPX}$$

$$X(5) = X(4) + \text{SEPX}$$

$$X(6) = X(5) + \text{SEPX}$$

$$X(7) = X(6) + \text{SEPX}$$

$$X(8) = X(7) + \text{SEPX}$$

*

* Poner las figuras para seleccionar el mes

*

* Triángulo mes izquierdo con línea de 3 pixeles y efecto
* de rombos en su interior.

*

$$XT1 = X(1)$$

$$YT1 = YL1$$

$$XT2 = XT1 + \text{SEPX2}$$

$$YT2 = YT1 + \text{SEPY2}$$

$$XT3 = XT1 + \text{SEPX2}$$

$$YT3 = YT1 - \text{SEPY2}$$

FORMAR IMAGEN FIGURA-BASE \$

/EFECTO-LINEAS 'LLENO' 30 \$

/TRIANGULO XT1 YT1 XT2 YT2 XT3 YT3 \$

'ROJO' 'ROMBOS' 'CELESTE-B' 101

*

* Triángulo mes derecho, poner el triángulo igual y cancelar
* las opciones

*

$$XT1 = X(3)$$

$$YT1 = YL1$$

$$XT2 = XT1 - \text{SEPX2}$$

$$YT2 = YT1 + \text{SEPY2}$$

$$XT3 = XT1 - \text{SEPX2}$$

$$YT3 = YT1 - \text{SEPY2}$$

```

FORMAR IMAGEN FIGURA-BASE $
  /TRIANGULO XT1 YT1 XT2 YT2 XT3 YT3 $
    'ROJO' 'ROMBOS' 'CELESTE-B' 102 $
  /NO-OPCIONES-FIGURAS

```

*

* Rectángulo entre ellos, sólido de color rojo

*

```
XR0 = X ( 2 ) - SEPX2
```

```
YR0 = YL1 - SEPY4
```

```
TRX = SEPX
```

```
TRY = SEPY2
```

```
FORMAR IMAGEN FIGURA-BASE $
```

```
  /RECTANGULO XR0 YR0 TRX TRY 'ROJO'
```

*

* Texto mes

*

```
FORMAR IMAGEN FIGURA-BASE $
```

```
  /TEXTO X ( 2 ) YL1 'C-M' 'Mes' 'SUIZO' SEPY2 0 $
```

```
    'TRANSPARENTE' 'CELESTE-B'
```

*

* Poner las figuras para seleccionar el año

* Triángulo año izquierdo

*

```
XT1 = X ( 6 )
```

```
YT1 = YL1
```

```
XT2 = XT1 + SEPX2
```

```
YT2 = YT1 + SEPY2
```

```
XT3 = XT1 + SEPX2
```

```
YT3 = YT1 - SEPY2
```

```
FORMAR IMAGEN FIGURA-BASE $
```

```
  /EFECTO-LINEAS 'LLENO' 30 $
```

```
  /TRIANGULO XT1 YT1 XT2 YT2 XT3 YT3 $
```

```
    'ROJO' 'ROMBOS' 'CELESTE-B' 103
```

*

* Triangulo año derecho

*

$$XT1 = X (8)$$

$$YT1 = YL1$$

$$XT2 = XT1 - SEPX2$$

$$YT2 = YT1 + SEPY2$$

$$XT3 = XT1 - SEPX2$$

$$YT3 = YT1 - SEPY2$$

FORMAR IMAGEN FIGURA-BASE \$

/TRIANGULO XT1 YT1 XT2 YT2 XT3 YT3 \$

'ROJO' 'ROMBOS' 'CELESTE-B' 104 \$

/NO-OPCIONES-FIGURAS

*

* Rectángulo entre ellos

*

$$XR0 = X (7) - SEPX2$$

$$YR0 = YL1 - SEPY4$$

$$TRX = SEPX$$

$$TRY = SEPY2$$

FORMAR IMAGEN FIGURA-BASE \$

/RECTANGULO XR0 YR0 TRX TRY 'ROJO'

*

* Texto año

*

FORMAR IMAGEN FIGURA-BASE \$

/TEXTO X (7) YL1 'C-M' 'Año' 'SUIZO' SEPY2 0 \$

'TRANSPARENTE' 'CELESTE-B'

*

* Poner letrero de dias, es decir Do Lu ...

*

$$I = 1$$

$$LA = SEPY - 4$$

```

5   LX = X ( I ) + SEPX2
    CONVERTIR 'DLMMJVS' A XDIAS /POS. I 1
    CONVERTIR 'ouaiuia' A XDIAS /POS. I 2
    FORMAR IMAGEN FIGURA-BASE $
    /TEXTO LX YL3 'C' XDIAS 'SU-N' LA 0 'TRANSPARENTE' 'NEGRO'
    I = I + 1
    SI I < 8 EJECUTA 5
    XR0 = DIM-X / 2
    TRX = SEPX + SEPX2 - 4
    TRY = SEPY - 3
*
*   Poner una línea entre los días y las flechas de selección del
*   mes y año, en forma opcional poner un elipse donde se pondrá
*   el año y el mes, preparar para dibujar los días.
*
    FORMAR IMAGEN FIGURA-BASE          $
    /LINEA 0 YL2 DIM-X YL2 20 'AZUL'     $
    /EFECTO-LINEAS 'LLENO' 1            $
    /USAR-COLOR-MARCO                    $
*   /ELIPCE XR0 YL1 TRX TRY 0 360 'BLANCO' $
*           'LINEAS-VERTICALES' 'GRIS'   $
    /REDONDEADO                           $
    /FIN-DE-INSTRUCCIONES
*
*   Sacar de fecha a desplegar, cual es el primer día de la
*   semana 1=do 2=lu, ...
*
10  FECHA-1DM = FECHA-DESP / 100
    FECHA-1DM = FECHA-1DM * 100 + 1
    FECHA-BASE = 19960101
    SI FECHA-DESP < FECHA-BASE FECHA-DESP = FECHA-BASE
    RESTAR FECHA-BASE A FECHA-1DM DIF-D DIF-M DIF-A
    SEMANA = DIF-D + 1 / 7

```

NUM-DIA = SEMANA * -7 + DIF-D + 2

*

* Sacar dia primero de próximo mes

*

FECHA-1DPM = FECHA-1DM

SUMAR 1 MES A FECHA-1DPM

*

* Sacar el mes y desplegar mes y año

*

NUM-MES = DIF-A * -12 + DIF-M + 1

CONVERTIR 'EFMAMJJASOND' A XMES /POS. NUM-MES 1

CONVERTIR 'neabauugecoi' A XMES /POS. NUM-MES 2

CONVERTIR 'ebrrynloptvc' A XMES /POS. NUM-MES 3

CONVERTIR FECHA-DESP A XAN /POS. 1 1 /LONG. 4

LA = SEPY - 2

LX = X (4) + SEPX2

FORMAR IMAGEN FIGURA-MES CON FIGURA-BASE \$

 /VARIAS-INSTRUCCIONES \$

/TEXTO LX YL1 'C' XAN 'SU-N' LA 0 'TRANSPARENTE' 'NEGRO' \$

/TEXTO LX YL2 'C' XMES 'SU-N' LA 0 'TRANSPARENTE' 'NEGRO'

*

* Formar el mes

*

FECHA-PRO = FECHA-1DM

DIA-PRO = 1

I = NUM-DIA

J = 1

MOVER '' A XDIAS

TRX = SEPX - 6

TRY = SEPY - 6

LA = SEPY - 8

*

* Rectángulo del día

```

*
20  XR0 = X ( I ) + 3
    YR0 = Y ( J ) + 3
    XRM = X ( I ) + SEPX2
    SI DIA-PRO < 10  CONVERTIR DIA-PRO A XDIAS /POS. 4 1
    SI DIA-PRO >= 10 CONVERTIR DIA-PRO A XDIAS /POS. 3 1 /LONG. 2
*
    SI FECHA-DESP <> FECHA-PRO  FORMAR IMAGEN FIGURA-MES $
    /RECTANGULO XR0 YR0 TRX TRY 'ROJO' 'ROMBOS' 'GRIS' DIA-PRO $
    /TEXTO XRM YR0 'C' XDIAS 'SU-N' LA 0 'TRANSPARENTE' 'CELESTE-B'
*
    SI FECHA-DESP = FECHA-PRO  FORMAR IMAGEN FIGURA-MES $
    /RECTANGULO XR0 YR0 TRX TRY 'ROJO' 'LLENO' 'ROJO' DIA-PRO $
    /TEXTO XRM YR0 'C' XDIAS 'SU-N' LA 0 'TRANSPARENTE' 'CELESTE-B'
    I = I + 1
    SI I < 8 EJECUTA 22
    I = 1
    J = J + 1
22  DIA-PRO = DIA-PRO + 1
    SUMAR 1 DIA A FECHA-PRO
    SI FECHA-PRO < FECHA-1DPM EJECUTA 20
    FORMAR IMAGEN FIGURA-MES /FIN-DE-INSTRUCCIONES
*
    OPCION TERMINAL AREA 8000 /DESPLEGAR
    ESCRIBIR TERMINAL FIGURA-MES
    LEER TERMINAL EJECUTA 100 DIA-SEL
    OPCION TERMINAL AREA 8000 /TERMINAR
*
    SI DIA-SEL > 100 EJECUTA 30
    CONVERTIR DIA-SEL A FECHA-DESP /POS. 3 7 /LONG. 2
    RETORNO
30  SI DIA-SEL = 101 SUMAR -1 MES A FECHA-DESP
    SI DIA-SEL = 102 SUMAR 1 MES A FECHA-DESP

```

SI DIA-SEL = 103 SUMAR -1 ANOS A FECHA-DESP

SI DIA-SEL = 104 SUMAR 1 ANOS A FECHA-DESP

EJECUTA 10

100 DIA-SEL = 0

RETORNO

A continuación se explica el funcionamiento de la RUTINA CALENDARIO.

Se define el área 8000 como de /MENU /SELECCIONAR-MARCAS para desplegar la imagen. Las marcas serán los días del mes. Una característica de la rutina es que se puede cambiar el tamaño del área y automáticamente el calendario se ajusta. Aquí se definió como /LONG. 20 60 pero se puede usar por ejemplo 10 20. Nota importante: si se ponen valores pequeños, pueden NO ser adecuados por el uso de fonts muy pequeños.

Una variable imagen puede contener una gráfica, una imagen leída o bien FIGURAS o combinación de gráficas e imágenes leídas bajo ciertas reglas. Para el caso de figuras se debe poner /INICIO para indicar que se va a proceder a formar la imagen y además dar cierta información. La instrucción sería:

FORMAR IMAGEN FIGURA-BASE	\$
/INICIO 0 0 'TRANSPARENTE'	\$
/MEDIDA-AREA 8000 DIM-X DIM-Y	\$
/X-Y-NORMAL	\$
/VARIAS-INSTRUCCIONES	

La explicación es la siguiente:

- ◆ Con /INICIO se especifica que se va a iniciar la imagen con las medidas en pixeles que se indican a continuación en este caso 0 0 y con el color de fondo indicado. Al poner 'TRANSPARENTE' se toma el del área de terminal en donde se despliega.
- ◆ Con /MEDIDA-AREA se especifica que las medidas de /INICIO sean remplazadas por las del área de terminal 8000, razón por la cual se puso 0 0 en dicha opción, además en forma opcional se especifican 2 variables N1-0 para que SISINF deje el valor en PÍXELES de las dimensiones del área. En este caso DIM-X DIM-Y , NOTA: Con esto la lógica se hace más general ya que se ajusta al tamaño del área en que se despliega.
- ◆ Con /X-Y-NORMAL que es el valor preestablecido se especifica que los ejes X-Y donde se darán los datos son los normales es decir el eje Y de abajo hacia arriba estando el punto 0-0 en la parte inferior izquierda, con la opción /X-Y-GRAFICO se especifica que el eje Y es de arriba hacia abajo,

estando el punto 0-0 en la parte superior izquierda. El valor preestablecido es normal por lo que se pudo haber omitido.

- ◆ Se especifica que se va a formar con /VARIAS-INSTRUCCIONES Note que el orden en que se especifiquen estas opciones puede ser cualquiera.

En este caso el área 8000 no tiene marco y para darle uno al calendario se dibujará un rectángulo con un cierto ancho de línea. La instrucción sería:

```

FORMAR IMAGEN FIGURA-BASE          $
      /EFECTO-LINEAS 'LLENO' 50      $
      /RECTANGULO 0 0 DIM-X DIM-Y     $
      'AZUL' 'LLENO' 'TRANSPARENTE'  $
      /NO-OPCIONES-FIGURAS

```

La explicación es la siguiente:

- ◆ Cuando se dibuja una figura geométrica (rectángulo, triángulo, círculo, ...) Se dibujan una o varias líneas para definir los contornos de dicha figura formando un marco y luego se rellena el interior. Con /EFECTO-LINEAS se especifican los efectos y el ancho de dicha línea en este caso 'LLENO' o continuo y de un ancho de 5.0 pixeles (expresado sin decimales y multiplicado por 10 o sea 50)
- ◆ Luego se pide dibujar un /RECTANGULO en la posición en X y en Y de 0 0 y con dimensión en X de DIM-X y en Y de DIM-Y que son las del área de la terminal. La línea que define el marco del rectángulo será 'AZUL' y el efecto del relleno es 'LLENO' y de un color 'TRANSPARENTE' es decir, lo que esté en el fondo en este caso 'GRIS'. Note que también se puede poner 'GRIS' en lugar de 'TRANSPARENTE' pero de esta forma queda mas flexible ya que con solo cambiar el fondo del área 8000 se cambia también el del rectángulo.
- ◆ Se cancela el efecto de las líneas con /NO-OPCIONES-FIGURAS
- ◆ Note que en el caso de figuras, el orden en que se especifican en la instrucción de FORMAR IMAGEN es el orden de ejecución, así una figura puede estar sobrepuesta a otra. Si se hubiera puesto primero /RECTANGULO y luego /EFECTO-LINEAS no habría el efecto de la línea para ese rectángulo.

Como se necesita calcular las coordenadas de todas las figuras que se van a desplegar, se divide el eje Y en 9 renglones y calcular la Y de cada uno. Se usará 2 para el título, 1 para el letreros de "Do Lu ..." y 6 para poner los días del mes, se dejan un espaciado de 5 arriba y abajo, 4 más para la línea. Dividir el

eje X en 7 columnas una para cada día de la semana menos un espaciado de 10 a ambos lados.

Para que el usuario pueda cambiar el mes se dibujan dos triángulos a manera de flechas unidos por un rectángulo y con la palabra 'Mes'. Las instrucciones para el triángulo izquierdo del mes y para el texto son:

```

FORMAR IMAGEN FIGURA-BASE          $
  /EFECTO-LINEAS 'LLENO' 30         $
  /TRIANGULO XT1 YT1 XT2 YT2 XT3 YT3 $
    'ROJO' 'ROMBOS' 'CELESTE-B' 101
FORMAR IMAGEN FIGURA-BASE          $
  /TEXTO X ( 2 ) YL1 'C-M' 'Mes'     $
    'SUIZO' SEPY2 0 $
    'TRANSPARENTE' 'CELESTE-B'

```

Como comentarios a la primera instrucción para dibujar el triángulo izquierdo se tiene:

- ◆ Se dibuja el triángulo con un marco 'ROJO' de 3.0 pixeles, la parte interior tiene un efecto de 'ROMBOS' de color 'CELESTE-B'
- ◆ Al final se especifica un 101 para indicar que este triángulo está marcado con la marca 101, de forma que si el usuario oprima botón izquierdo con el ratón en dicho triángulo, el número de marca que se leerá será el 101, con lo que la lógica del programa regresa un mes.

Como comentarios a la segunda instrucción para poner el texto:

- ◆ Se define la posición X y Y de un rectángulo en donde se escribirá el texto, estos valores están en pixeles como todas las opciones de figuras.
- ◆ Luego aparece la opción de como interpretar la posición. Si es la posición inferior izquierda del rectángulo se debe poner 'JI' por justificado a la izquierda, si la posición X es el centro del rectángulo entonces se pone 'C' por centrado y si es la derecha 'JF' por justificado a la derecha 'JD'. También si la posición en Y es la mitad del rectángulo se agrega un guión y una M.
- ◆ En este caso la posición X es centrado y la Y es a la mitad por lo que se puso 'C-M'
- ◆ Se tiene la literal con 'Mes' que también se podría poner una variable alfanumérica.

- ◆ Luego se define la familia del font al igual que en las instrucciones de impresión. Aquí permite poner negrita (N), inclinado (I) o relieve (R). Así poner 'SU-NIR' sería suizo con las tres opciones.
- ◆ Luego se especifica el alto del rectángulo en pixeles que es parecido a tamaño en puntos.
- ◆ Viene el ancho máximo del rectángulo ocasionando que se trunque el texto si se excede o bien un 0 que equivale a lo que necesite.
- ◆ Por último el color del fondo y de las letras, en este caso como 'TRANSPARENTE' y 'CELESTE-B'

Con las mismas instrucciones se dibuja algo parecido al inciso anterior pero para el 'Año', luego las iniciales de cada día de la semana separados de lo anterior por una línea y al final se prepara para el dibujo de los rectángulos que contendrán cada día del mes. La instrucción para hacerlo es:

```

FORMAR IMAGEN FIGURA-BASE          $
  /LINEA 0 YL2 DIM-X YL2 20 'AZUL'   $
  /EFECTO-LINEAS 'LLENO' 1          $
  /USAR-COLOR-MARCO                 $
  /REDONDEADO                        $
  /ELIPCE XR0 YL1 TRX TRY 0 360 'BLANCO' $
      'LINEAS-VERTICALES' 'GRIS'     $
  /FIN-DE-INSTRUCCIONES

```

Los siguientes comentarios son aplicables a las diferentes opciones, recuerde algunas opciones son aplicables a los rectángulos que se dibujarán para cada día del mes. Estos son:

- ◆ Con /LINEA se dibuja la línea de separación entre las flechas y las iniciales de los días de la semana, los datos son posición inicial en X y Y, posición final en X y Y, ancho y color.
- ◆ Si el ancho de la línea para el marco de una figura se hace muy pequeño de todas formas se dibuja lo mas pequeño posible, la única excepción es poner 1 que equivale a cancelarlo. Por eso se puso /EFECTO-LINEAS 'LLENO' 1 porque en este caso NO se desea marco en las figuras.
- ◆ Los efectos en figuras se generan con el color del efecto y el color inverso a menos que se especifique /USAR-COLOR-MARCO en donde se usará el del marco.

- ◆ Los rectángulos se pueden dibujar con las esquinas redondeadas para ello especificar /REDONDEADO.
- ◆ Como un ejemplo de otra figura se puso un elipse en donde se despliega el mes y año del calendario los datos son posición X y Y del centro del elipse, radio en X y Y, ángulo inicial y final, color del marco, efecto y color de efecto. En este caso se puso 'BLANCO' 'LINEAS-VERTICALES' 'GRIS' que por las opciones anteriores se dibuja un elipse sin marco, con un efecto de líneas verticales blanco con gris.
- ◆ Por ultimo se pone /FIN-DE-INSTRUCCIONES con lo cual se almacenan TODAS las instrucciones en FIGURA-BASE.

Una vez construída la imagen base se procede a completarla con la información propia de cada mes como son el texto del año y del mes. Para ello la instrucción es:

```

FORMAR IMAGEN FIGURA-MES          $
    CON FIGURA-BASE                $
    /VARIAS-INSTRUCCIONES          $
    /TEXTO LX YL1 'C' XAN 'SU-N' LA 0 $
        'TRANSPARENTE' 'NEGRO'     $
    /TEXTO LX YL2 'C' XMES 'SU-N' LA 0 $
        'TRANSPARENTE' 'NEGRO'

```

Aquí se copia la FIGURA-BASE a FIGURA-MES, iniciando esta última, ya que esta parte de la lógica se usará cada vez que cambia el mes o el año. Se pone el texto para hacerlo. Luego se ponen los días diferentes a la fecha del día con la instrucción:

```

FORMAR IMAGEN FIGURA-MES          $
    /RECTANGULO XR0 YR0 TRX TRY     $
        'ROJO' 'ROMBOS' 'GRIS' DIA-PRO $
    /TEXTO XRM YR0 'C' XDIAS 'SU-N' LA 0 $
        'TRANSPARENTE' 'CELESTE-B'

```

Se dibuja el rectángulo que por las últimas instrucciones puestas en FIGURA-BASE será sin marco y con un efecto de 'ROMBOS' en rojo con gris. Note que el rectángulo tiene una marca en la variable DIA-PRO que es el número del día que se podrá seleccionar, Luego se pone el texto del número del día.

Al final se pone:

FORMAR IMAGEN FIGURA-MES /FIN-DE-INSTRUCCIONES

13.8) Resumen de Figuras.

Las opciones de figuras se pueden resumir en:

/INICIO Dim-X Dim-Y Color-fondo	Iniciar la imagen a unas dimensiones.
/MEDIDA-AREA Numero-Area (Dim-X Dim-Y)	Sacar las medidas de un área de terminal y remplazarlas en /INICIO.
/EMPALMAR Pos-X Pos-Y Dim-X Dim-Y Imagen-Leída Número-Marca	Agrandar o reducir una imagen leída y ponerla en otra imagen asociándola a una marca.
/CORTAR Pos-X Pos-Y Dim-X Dim-Y	Cortar una imagen y dejar la nueva en la original.
/X-Y-NORMAL o /X-Y-GRAFICO	Ejes X-Y normales o Gráficos.
/EFECTO-LINEAL Efecto Ancho*10	Definir efectos de las líneas de figuras.
/REDONDEADO	Redondear rectángulos.
/USAR-COLOR-MARCO	Usar color marco para el efecto en vez de color inverso.
/NO-OPCIONES-FIGURAS	Cancelar opciones de figuras.
/LINEA Pos-X-i Pos-Y-i Pos-X-f Pos-Y-f Ancho*10 (Efecto)	Dibujar línea.
/TRIANGULO Pos-X1 Pos-Y1 Pos-X2 Pos-Y2 Pos-X3 Pos-Y3 Color-marco (Efecto Color-efecto Número-Marca)	Dibujar triángulo.
/RECTANGULO Pos-X Pos-Y Dim-X Dim-Y Color-marco (Efecto Color-efecto Número-Marca)	Dibujar rectángulo.
/ELIPSE Pos-X Pos-Y Radio-1 Radio-2 Angulo-1 Angulo-2 Color-marco (Efecto Color-efecto Número-Marca)	Dibujar elipse.
/CIRCULO Pos-X Pos-Y Radio Angulo-1 Angulo-2 Color-marco	Dibujar círculo.

(Efecto Color-efecto Número-Marca)

/TEXTO Pos-X Pos-Y Tipo-pos Texto Agregar un Texto.
 Font Alto Ancho Color-fondo
 Color-letras (Efecto Color-efecto)

Los tipos de posición en texto son:

- 0 'JI' Justificado a la izquierda
- 1 'C' Centrado horizontalmente
- 2 'JD' Justificado a la derecha
- 3 '+90' Girado 90 grados (de abajo hacia arriba)
- 4 '-90' Girado -90 grados (a arriba para abajo)
- 10 'JI-M' Justificado a la izquierda, centrado verticalmente
- 11 'C-M' Centrado horizontalmente, centrado verticalmente
- 12 'JD-M' Justificado a la derecha, centrado verticalmente

Los valores de font son:

- 1 'MODERNO'
- 2 'DECORATIVO'
- 3 'ROMANO'
- 4 'SUIZO'
- 5 'MANUSCRITO'

Para generar las opciones de Negrita, Inclinado o Relieve se deberá de tomar las dos primeras letras del font, poner el guión y una, dos o las tres letras de N I R (En ese orden).

Efectos de líneas:

- 0 'NORMAL'
- 1 'PUNTOS-JUNTOS'
- 2 'PUNTOS-SEPARADOS'
- 3 'GUION-GIGANTE'
- 4 'GUION-GRANDE'
- 5 'GUION-MEDIANO'
- 6 'GUION-CHICO'
- 7 'GUION-MEDIANO-CHICO'
- 8 'GUION-PUNTO'

14) ERRORES DE EJECUCION.

En la ejecución de la instrucción aritmética puede haber varios tipos de errores, todos son informativos. Se debe revisar el programa para solucionarlos:

Error	Causa/Solución
001 El número calculado excede la capacidad de la variable, se trunca..	♦ Revisar Programa y/o información.
003 Número calculado en la 'SUMA'. Mayor a 16 dígitos, se trunca.	♦ Revisar Programa y/o información.
004 Número calculado en la 'MULTIPLICACION' mayor a 16 dígitos, se trunca.	♦ Revisar Programa y/o información.
005 División entre cero, no se efectúa.	♦ Revisar Programa y/o información.
006 Número calculado en la 'DIVISION' mayor a 16 dígitos, se trunca.	♦ Revisar Programa y/o información.
017 Número fuera de rango en exponenciación	♦ Revisar Programa y/o información.

En la ejecución de la instrucción de SUMAR o RESTAR puede haber varios tipos de errores, todos ellos informativos. Se debe revisar el programa para solucionarlos:

Error	Causa/Solución
007 Fecha incorrecta, mes y día fuera de rango. Se ignora la instrucción.	♦ Por una igualación (FECHA = constante) o por LEER TEMPORAL se tiene una fecha inválida.
008 Fecha inicial mayor a fecha final. Se ignora la instrucción.	♦ Cambiar el orden de la lógica.

Error	Causa/Solución
009 Número de días en 'RESTAR' mayor a 30000, se ignora la instrucción.	<ul style="list-style-type: none"> ◆ Revisar programa y/o información.
010 Incremento en 'SUMAR' no esta entre -9999 y +9999, se ignora.	<ul style="list-style-type: none"> ◆ Revisar programa y/o información.

Los posibles errores de cancelación y su posible corrección son:

Error	Causa/Solución
021 Versión de 'CLS' y 'ELS' incompatible, ejecute 'CLS' y luego 'ELS'.	<ul style="list-style-type: none"> ◆ Se debe compilar nuevamente el programa. ◆ Revisar que los módulos de CLS y ELS sean compatibles.
023 Nivel de Seguridad en escritura de la 'CLAVE DE ACCESO' no es el adecuado.	<ul style="list-style-type: none"> ◆ El usuario no puede ejecutar este programa. ◆ Revisar con CPS la clave de acceso y ponerle los valores adecuados.
024 Nivel de Seguridad en lectura de la 'CLAVE DE ACCESO' no es el adecuado.	<ul style="list-style-type: none"> ◆ El usuario no puede ejecutar este programa. ◆ Revisar con CPS la clave de acceso y ponerle los valores adecuados.
025 Esta 'CLAVE DE ACCE-SO' no puede ejecutar programas de esta DBD..	<ul style="list-style-type: none"> ◆ El usuario no puede ejecutar este programa. Revisar con CPS la clave de acceso y ponerle los valores adecuados.

Error	Causa/Solución
<p>026 La 'CLAVE DE ACCESO' no tiene un número de seguridad en este programa</p>	<ul style="list-style-type: none"> ◆ El usuario no puede ejecutar este programa. ◆ Revisar con CPS la clave de acceso y ponerle los valores adecuados.
<p>032 Error en MOVER, posición fuera de rango.</p>	<ul style="list-style-type: none"> ◆ Revisar el programa.
<p>033 No se especifica /ERROR en Convertir o Formar Texto. Hay error al ejecutarla.</p>	<ul style="list-style-type: none"> ◆ Poner /ERROR # para que no se cancela la ejecución. ◆ Revisar el programa.
<p>034 Se excede el máximo separado para Texto/Imagen/Audio en mover o formar texto.</p>	<ul style="list-style-type: none"> ◆ Cambiar la especificación de AREA para el campo en DBD y programas que la usen. ◆ Revisar se tenga en texto la opción /INICIO para no acumular Textos.
<p>156 Información inconsisten-te en registro de Texto/Imagen/audio.</p>	<ul style="list-style-type: none"> ◆ Si el campo causa error en todos los programas, el campo dañado se deberá borrar e insertar de un respaldo. ◆ Si solo causa error en un programa volver a compilarlo. Si no se corrige reportarlo a personal de su Distribuidor.
<p>157 Error interno en /Texto/Imagen/Audio.</p>	<ul style="list-style-type: none"> ◆ Se debe recompilar el programa. Si el error persiste reportarlo a personal de su Distribuidor

Error	Causa/Solución
165 Error interno en proceso de área con nombre.	<ul style="list-style-type: none"> ◆ Se debe recompilar el programa. Si el error persiste reportarlo a personal de su Distribuidor
166 Se cambió la definición del área con nombre en DBD compilar nuevamente.	<ul style="list-style-type: none"> ◆ Se debe compilar el programa.
168 Más de 42 áreas comunes en una ejecución de ELS.	<ul style="list-style-type: none"> ◆ Juntar varias áreas en una sola y ejecutar DBD y CLS.
171 Subíndice fuera de Rango.	<ul style="list-style-type: none"> ◆ Revisar el programa.
176 Más de 10 Ejecutar Rutina.	<ul style="list-style-type: none"> ◆ Revisar el programa.
177 Error en tablas internas de Rutina.	<ul style="list-style-type: none"> ◆ Se deberá recompilar el programa y si el error persiste, reportarlo a personal de su Distribuidor.
218 Error en Archivo TEMxyy.	<ul style="list-style-type: none"> ◆ Es necesario mediante el uso del módulo CPS agrandar el tamaño de TEMxyy ya que no tiene capacidad para todas las áreas con nombre, variables tipo texto, imagen o audio y para OPCION TERMINAL AREA. ◆ El archivo TEMxyy no se puede crear. El error del Sistema Operativo se despliega en la pantalla para poder resolver el problema. ◆ Error al acceder TEMxyy. Reportar a personal de su Distribuidor si al reintentar no se corrige.
226 Se despliega un área de menú en la rutina de calcula nivel.	<ul style="list-style-type: none"> ◆ Revisar el programa.

Error	Causa/Solución
227 En Texto o Imagen tamaño de área insuficiente	<ul style="list-style-type: none">◆ Agrandar en DBD la opción de AREA y compilar DBD Y programas que la usen.
266 Gráfica o imagen creada en forma incorrecta	<ul style="list-style-type: none">◆ No se siguieron todas las reglas para formar la imagen. Revise Programa.◆ Faltan datos, marcas repetidas.