

SISTEMA DE GRAFICACION G

GABRIEL MAURICIO ALVAREZ MEDINA

LUIS MIGUEL MORALES MANILLA

1a. EDICION: MARZO 1986

F-DEPFI

D-62

1986

g.2



DEPFI

CONTENIDO

Presentación.....	1
Rutinas de Graficación.....	3
Gonoff.....	4
Gpos.....	6
Gline.....	7
Gpol.....	8
Grec.....	9
Gcurv.....	10
Gvec.....	12
Glines.....	13
Gcol.....	14
Gbp.....	14
Gcv.....	14
Garc.....	15
Gcir.....	16
Gscale.....	17
Gaxis.....	18
Gtext.....	20
Gbar.....	21
Gdig.....	23
Set-up.....	26
Llamadas a las rutinas de G.....	28
FORTTRAN y G.....	28
Pascal y G.....	29
BASIC y G.....	30
Ejemplos.....	32

U.N.A.M

DIVISION DE ESTUDIOS DE POSGRADO

FACULTAD DE INGENIERIA

UNIDAD DE COMPUTO

G

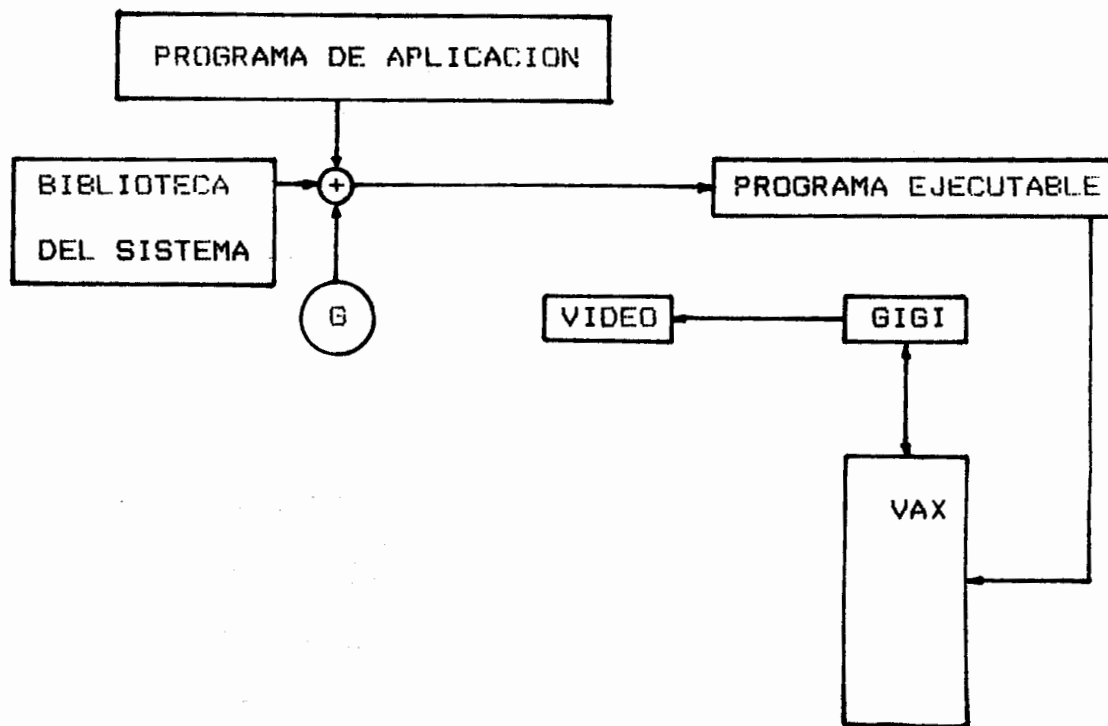
Los actuales equipos de graficación incluyen como elemento básico para su operación un conjunto elemental de comandos. En diversas ocasiones los elementos del conjunto están formados por series de caracteres de control o, en el mejor de los casos, por funciones predefinidas y almacenadas en la biblioteca del sistema.

El conjunto de comandos puede ser utilizado en forma independiente o se le puede emplear utilizando un lenguaje de alto nivel como anfitrión, permitiendo esto último la facilidad de cálculo y graficación.

Los sistemas de graficación aprovechan las facilidades que presenta una etapa denominada ligado para unir diversos módulos objetos, de entre los cuales se encuentra el que posee la definición de los comandos de graficación, siendo ésta la base mediante la cual, contando con una sola definición de comandos de graficación, sea posible emplear diferentes lenguajes de programación.

G es un conjunto de rutinas básicas de graficación que pueden ser llamadas desde cualquier lenguaje de programación soportado por la familia de minicomputadoras VAX. Las rutinas se encuentran escritas en lenguaje FORTRAN 77 y utilizan como funciones primitivas de graficación, a los comandos proporcionados por ReGIS (Remote Graphics Instruction Set). ReGIS se encuentra soportado por la microcomputadora GIGI (General Imaging Generator and Interpreter), la cual se comunica con una VAX. GIGI es la encargada de interpretar los comandos de graficación en los cuales está basado G.

El siguiente es un diagrama general de interrelación de elementos mostrando la acción de G en una aplicación



R U T I N A S D E G R A F I C A C I O N

G proporciona las rutinas básicas para crear cualquier ambiente complejo de graficación. Se proveen rutinas para la graficación de puntos, vectores, vectores concatenados, rectángulos, curvas, ejes, escalamientos, círculos, posicionamiento, selección de color, digitalización y selección de opciones básicas de G. Todas las rutinas de G están identificadas porque su nombre se inicia siempre con G. Las rutinas requieren como parámetros únicamente los elementos básicos que identifican la acción a realizar, así, por ejemplo, para trazar un rectángulo, sólo se requieren dos coordenadas; para el trazo de una curva, sólo las coordenadas que el usuario juzgue suficientes para interpolar la curva; para escalar un conjunto de coordenadas, sólo se requiere conocer la ventana deseada.

Lista de rutinas del sistema "G":

Garc.....	Traza arcos de circunferencia.
Gaxis.....	Dibuja ejes coordenados.
Gbar.....	Elabora barras verticales.
Gbp.....	Limpia el video.
Gcir.....	Traza círculos.
Gcol.....	Selecciona color de la imagen.
Gcurv.....	Dibuja curvas cerradas o abiertas.
Gcv.....	Selecciona color de la pantalla.
Gdig.....	Digitaliza imágenes por pantalla.
Gline.....	Traza una línea recta dadas dos coordenadas.
Glines....	Une una serie de puntos con una línea recta.
Gonoff....	Selecciona modo gráfico y otras opciones.
Gp.....	Dibuja un punto.
Gpol.....	Dibuja un polígono de n coordenadas.
Gpos.....	Posiciona el cursor.
Grec.....	Traza un rectángulo.
Gscale....	Escala coordenadas en un puerto del video.
Gvec.....	Dibuja una línea recta dada una coordenada.

`Gonoff( id, sw )` permite colocar en estado activo o inactivo la opción `id`, donde `sw` es 0 o 1, (0 = estado inactivo; 1 = estado activo). `Id` es el identificador de la opción sobre la que se actúa y puede tomar los siguientes valores :

- a) 'mgraf' modo gráfico
- b) 'sneg' video inverso
- c) 'blink' blink
- d) 'shade' modo llenado de áreas

Mediante esta opción, es posible iniciar la etapa de graficación, seleccionando el modo gráfico; una vez terminado el proceso de graficación, se sale de él mediante la misma opción.

La opción de 'sneg' permite seleccionar video inverso sobre la imagen que trabajamos.

Cuando se selecciona la opción de 'blink', provoca que todo elemento gráfico que se dibuje, presente el atributo de "blink" (parpadeo). Al salir de este modo, los elementos subsecuentes que se dibujen ya no poseerán este atributo, pero los que se dibujaron anteriormente seguirán parpadeando.

La opción de 'shade' permite realizar llenados de áreas con algún color, carácter o forma definida por el usuario. Esta opción requiere una línea de referencia en el eje de las ordenadas, tomándose para ello el valor de la coordenada `Y` donde se encuentra el cursor al momento de elegir la opción.

Los siguientes son ejemplos de llamadas de estas rutinas :

Llamadas desde PASCAL

```

Gonoff('mgraf',1);      { entra a modo gráfico           }
Gonoff('blink',1);      { blink a todo elemento a graficar }
Gpos(100,50);           { posiciona el cursor en 100,50    }
Gonoff('shade',1);      { toma como referencia la posición }
                        { 50 de Y para el llenado de áreas }
Gonoff('mgraf',0);      { abandona modo gráfico           }

```

Llamadas desde FORTRAN 77

```

call Gonoff('mgraf',1)
call Gonoff('blink',1)
call Gpos(100,50)
call Gonoff('shade',1)
call Gonoff('mgraf',0)

```

Llamadas desde Basic

```

CALL Gonoff ('mgraf' BY DESC,1% BY REF)
CALL Gonoff ('blink' BY DESC,1% BY REF)
CALL Gpos    BY REF (100%,50%)
CALL Gonoff ('shade' BY DESC,1% BY REF)
CALL Gonoff ('mgraf' BY DESC,0% BY REF)

```

Gpos(X, Y) coloca el cursor en la posición X, Y del video. Los parámetros son enteros, teniendo como rango para X, desde 0 a 767, y para Y de 0 a 479. El origen está reconocido por la coordenada (0,0) correspondiente a la esquina inferior izquierda del video. El final del video se localiza en la esquina superior derecha y corresponde a la coordenada (767,479). Cualquier referencia fuera de este rango, provoca la acción de un mecanismo de recorte de referencia.

Gp(X, Y) realiza la graficación de un punto en la posición especificada. El rango de X y Y es el mismo especificado en Gpos.

El siguiente ejemplo ilustra el manejo de esta rutina y la posición de los puntos graficados :

GP (0,0)
GP (384,240)
GP (767,479)

767, 479

384, 240

.0,0

Gline(X, Y, X1, Y1), dadas dos coordenadas (X,Y) y (X1,Y1), traza una línea que las une. Los rangos de las X's y Y's son los mismos especificados en Gpos.

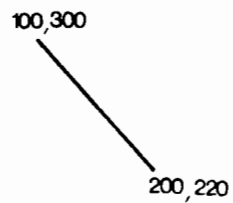
Ejemplo

I = 100

J = 200

CALL Gline(I,I+200,J,J-20)

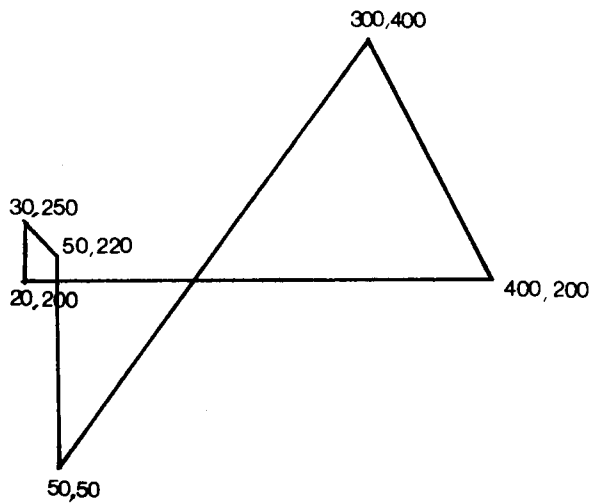
Este código produce la siguiente salida en el video:



Gpol(X, Y, N) dados dos arreglos de enteros X y Y, se realiza la graficación de un polígono de N coordenadas. Véase el siguiente ejemplo :

```
INTEGER   X(6)/50,300,400,20,30,50/
INTEGER   Y(6)/50,400,200,200,250,220/
CALL Gpol(X,Y,6)
```

El código anterior produce la siguiente salida :

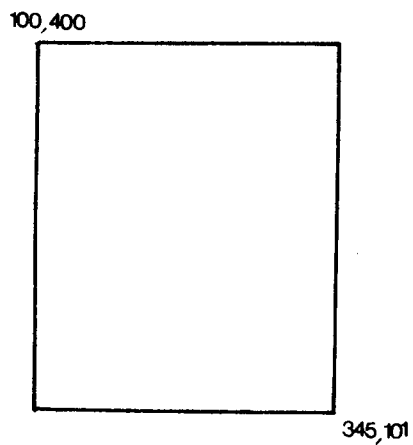


Debe observarse que la graficación del polígono sigue la secuencia especificada por el orden proporcionado de los vértices.

`Grec( X, Y, X1, Y1 )` dadas dos coordenadas (X,Y) y $(X1,Y1)$, se despliega un rectángulo donde (X,Y) y $(X1,Y1)$ corresponden a las coordenadas de los vértices opuestos en el sentido de cualquiera de sus dos diagonales. Veamos el siguiente ejemplo :

```
IX1 = 100  
IX2 = 345  
IY1 = 400  
IY2 = 101  
CALL Grec(IX1,IY1,IX2,IY2)
```

Este código genera la siguiente salida :

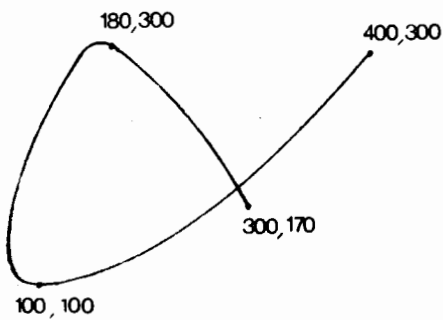


Gcurv(T, N, X, Y) permite, dados los arreglos enteros X y Y, realizar una interpolación de una curva compuesta por N coordenadas. El parámetro T indica si la curva será cerrada ('b') o abierta ('s').

Ejemplos

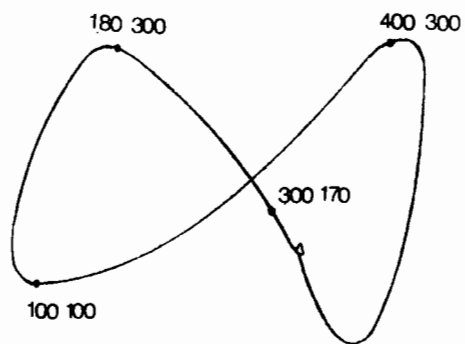
```
INTEGER IX(4)/400,100,180,300/, IY(4)/300,100,300,170/  
CALL Gcurv('s',4,IX,IY)
```

El código anterior produce la siguiente gráfica :



OPCION s

Si en la llamada anterior a Gcurv especificamos 'b' en lugar de 's', tendríamos el despliegue de una curva cerrada como se muestra a continuación :

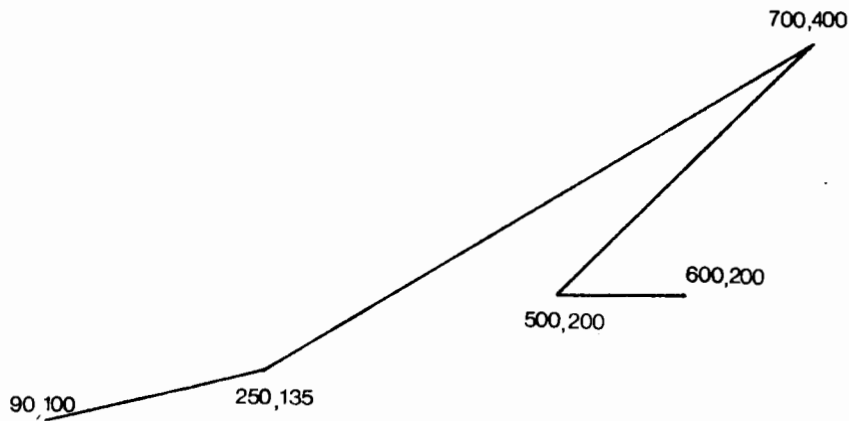


OPCION b

`Gvec( X, Y )` permite trazar un vector tomando como origen la posición actual del cursor hasta la posición especificada por la coordenada (X,Y). Veamos el siguiente ejemplo

```
CALL Gpos(90,100)
CALL Gvec(250,135)
CALL Gvec(700,400)
CALL Gvec(500,200)
CALL Gvec(600,200)
```

Este código produce la siguiente salida en video :

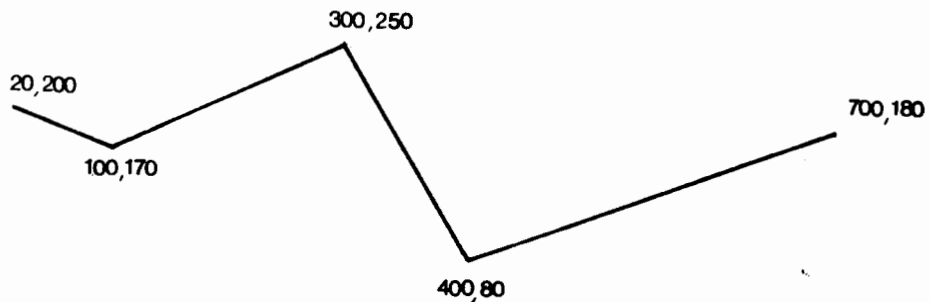


Debe observarse que la coordenada origen de un vector siempre es la posición actual del cursor, lo cual permite realizar la graficación de vectores a velocidades mayores.

Glines(X, Y, N, C) dibuja un conjunto de vectores unidos por sus extremos, pero que no forman un polígono cerrado. X y Y son arreglos enteros con las N coordenadas a unir; C especifica el color de la cadena de vectores (0-7). Ejemplo :

```
INTEGER  X(5)/20, 100, 300, 400, 700/  
INTEGER  Y(5)/200, 170, 250, 80, 180/  
CALL Glines(X,Y,5,1)
```

El código anterior produce la gráfica mostrada en color azul marino.



Gcol(N) permite seleccionar uno de ocho colores que provee el sistema. La tabla de correspondencia es la siguiente :

negro - 0	azul - 1	rojo - 2
magenta - 3	verde - 4	cyan - 5
amarillo - 6	blanco - 7	

Gbp permite borrar el video

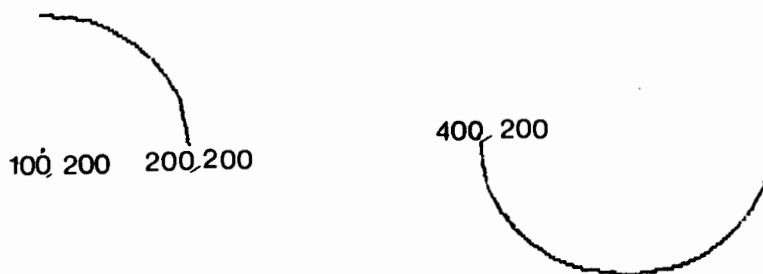
Gcv(n) permite seleccionar el color del fondo del video.
El parámetro n especifica el color seleccionado.

Garc (O, G, X, Y) dibuja un arco de circunferencia con una abertura de G grados; el radio del arco será igual a la distancia entre el punto donde se encuentra el cursor y el punto de coordenadas X, Y; el parámetro O indica dos opciones : la primera (' ') traza un arco con el centro en la posición original del cursor y deja el cursor en esa misma posición; la segunda ('c') traza un arco con el centro en la posición indicada por X,Y a partir de la posición original del cursor y deja el cursor en la última localidad dibujada sobre el arco.

Si el ángulo es positivo, los arcos se trazan en sentido contrario al de las manecillas del reloj y viceversa. Los parámetros G , X y Y son constantes o variables de tipo "entero", O es una constante o variable de tipo "carácter".

Ejemplo

```
CALL Gpos (100,200)
CALL Garc ( ' ',90,200,200 )
CALL Gpos (400,200)
CALL Garc ( 'c',180,500,200 )
```

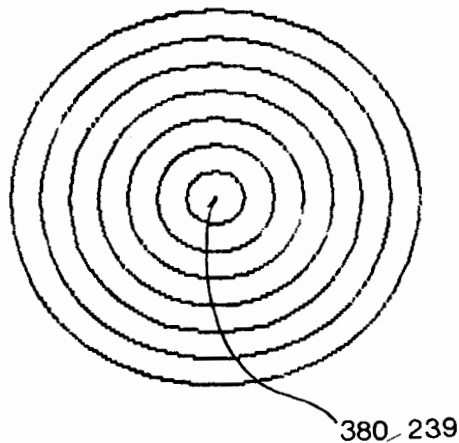


Gcir (O, X, Y) dibuja un círculo cuyo radio será igual a la distancia entre el punto donde se encuentra el cursor y el punto de coordenadas X, Y; el parámetro O indica dos opciones : la primera (' ') traza un círculo con el centro en la posición original del cursor y deja el cursor en esa misma posición, la segunda ('c') traza un círculo con el centro en la posición indicada por X,Y a partir de la posición original del cursor y deja el cursor en la ultima localidad dibujada sobre el círculo.

Los círculos se trazan en sentido contrario al de las manecillas del reloj. Los parámetros X y Y son constantes o variables de tipo "entero", O es una constante o variable de tipo "carácter".

Ejemplo

```
CALL Gpos (380,239)
DO I = 1,7
CALL Gcol (I)
CALL Gcir ( ' ',380+I*20,239)
END DO
```

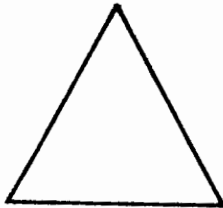


Gscale (X1, Y1, X2, Y2, XI, YI, XF, YF, N, X, Y, XE, YE)
 permite escalar N coordenadas X,Y en un puerto de la pantalla
 definido por los puntos X1,Y1 y X2,Y2, dados los valores mínimos
 XI,YI y máximos XF,YF de la ventana deseada del mundo real. Los
 valores escalados serán devueltos en los parámetros XE y YE.

Los parámetros X y Y son vectores de valores reales que
 serán escalados y transformados en los vectores enteros XS y YS;
 XI, YI, XF, YF, y N son constantes o variables de tipo entero.
 Los valores de las coordenadas del puerto deben estar entre 0 y
 767 en la horizontal, y entre 0 y 479 en la vertical.

Ejemplo

```
INTEGER XE(3),YE(3)
REAL X(3)/10,15,20/,Y(3)/10,20,10/
CALL Gscale (100,100,400,400,10,10,20,20,3,X,Y,XE,YE)
CALL Gpol (XE,YE,3)
```

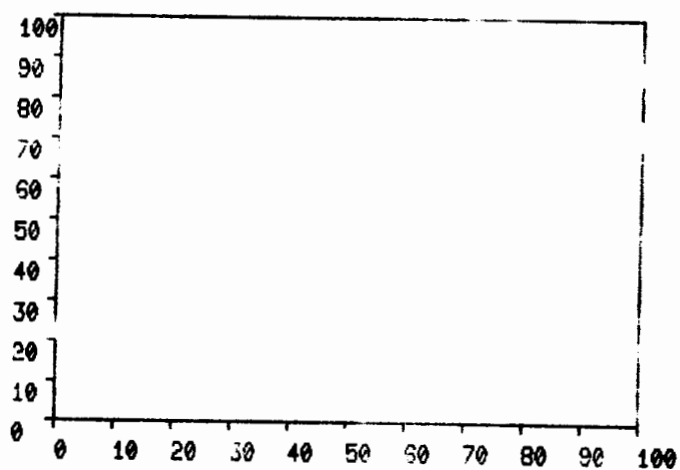


Δ

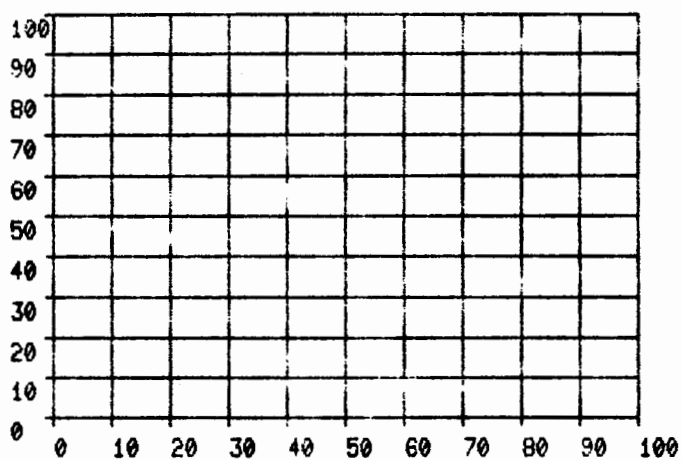
Gaxis (X1, Y1, X2, Y2, XI, YI, XF, YF, NX, NY, SX, SY, RC, RT, C) es una rutina que permite dibujar ejes en un puerto de la pantalla definido por los puntos (X1,Y1) y (X2,Y2). Los ejes son dibujados sobre los bordes del puerto especificado, a partir de las coordenadas iniciales de la ventana XI, YI; tanto estas coordenadas iniciales como las finales de ambos ejes, están dadas en las unidades que se manejarán en la gráfica. Además, es posible definir: a) el color de los ejes (C); b) el intervalo entre las marcas de ambos ejes (SX y SY); c) trazar una retícula (RT) o un recuadro (RC); d) indicar si se desea o no numeración sobre uno o ambos ejes (NX y NY).

Los parámetros X1, Y1, X2, Y2, XI, YI, XF, YF y N, son variables o constantes de tipo "entero"; el parámetro C (color) puede tomar cualquier valor entero entre 0 y 7; los intervalos de marcas SX y SY son también de tipo "entero", al igual que los parámetros NX, NY, RT y RC, los cuales sólo pueden tener dos valores posibles 1 o 0 (1 = activo, 0 = inactivo). En estos últimos cuatro parámetros, debe observarse que si no se desea numeración en uno o en ambos ejes, el valor de los intervalos de marcas correspondientes debe ser de 0; también es conveniente señalar que si se desea retícula, no es necesario que se especifique recuadro. Los valores de las coordenadas del puerto deben ser positivos y deben estar entre 0 y 767 sobre la horizontal, y entre 0 y 479 sobre la vertical. Los valores de las coordenadas de la ventana pueden ser positivos o negativos y pueden tomar cualquier valor. La numeración, si se especifica, sólo despliega números enteros positivos o negativos.

CALL Gaxis (100,100,500,400,0,0,100,100,1,1,10,10,1,0,7)



CALL Gaxis (100,100,500,400,0,0,100,100,1,1,10,10,0,1,5)



`Gtext( opciones, texto )` permite desplegar el texto dada una serie de opciones. Las posibles opciones se muestran a continuación.

Función		Comentario
Tamaño del texto	S <num>	<num> varía de 0 a 16
Dirección del texto	D <ang>	<ang> varía de 0 a +-360
Altura del texto	H <num>	<num> varía de 1 a 16
Angulo de Itálicas	I <ang>	<ang> varía de 0 a + 45
Tamaño explícito del caracter	S [ancho,altura]	donde ancho y altura estan dados en pixels
Espaciamiento	[{+-}X, {+-}Y]	espaciamiento entre caracteres, en pixels, donde los signos son opcionales.

Ejemplos de esta rutina serían

```
Gtext('S1H2','HOLA COMO ESTAS')
Gtext('[20,0]S0','texto espaciado')
```

En el primer ejemplo se desplegaría un texto en mayúsculas con un ancho normal y una altura dos veces más grande que la normal. En el segundo caso, se desplegaría un texto en minúsculas con el tamaño mínimo y un espaciamiento horizontal de 20 pixels entre los caracteres. Nótese que la opción de espaciamiento está situada antes que cualquiera otra opción. Tanto el conjunto de opciones, como el texto, deben ir encerrados entre apóstrofes.

Gbar (DX, DY, N, XI, YI, XF, SX, SH, COL) dibuja N barras verticales cuyas alturas están dadas por el vector DY, la anchura de las barras se especifica en DX, y el espaciamento entre cada barra en SX. Los parámetros XI y XF indican los valores de las abscisas extremas del puerto de video en donde estarán colocadas las barras, el parámetro YI señala la ordenada inicial de dicho puerto. Además, es posible seleccionar un color (COL) para las barras, y especificar si se desean barras llenas o vacías (SH).

Los parámetros DX, SX, y el vector DY son de tipo "real", mientras que SH, COL, N y las coordenadas del puerto XI, YI, XF son de tipo "entero". Normalmente, los valores de la anchura y el espaciamento entre barras oscilan entre 0 y 1. Por ejemplo, si se especifica una anchura de barras de 0.9, el espaciamento debe ser de 0.2, si se especifica una anchura de 1.0, el espaciamento es de 0.0, etc. aunque el usuario puede, en caso de requerirlo, dar valores más grandes que la unidad. El color puede ser uno de los 8 existentes (0-7). La altura de las barras, especificadas en el parámetro DY, deben ser escaladas previamente, ya sea mediante la rutina Gscale o bien por otro algoritmo proporcionado por el usuario.

Ejemplo

```
INTEGER XE(12),DY(12)
REAL ALT(12)/10.0,20.0,30.0,40.0,50.0,60.0,60.0,40.0,30.0,
* 20.0,10.0/,X(12)
CALL Gscale(100,100,400,200,0,0,12,60,12,X,ALT,XE,DY)
CALL Gbar(0.7,DY,12,100,100,400,0.3,1,5)
```

En este ejemplo los vectores XE y X no tienen valores, sin embargo es necesario pasarlos en la rutina Gscale, por lo cual se han dimensionado. La anchura de las barras es de 0.7 y la separación entre c/u es de 0.3.



En este ejemplo, los vectores XE y X no tienen valores; sin embargo es necesario pasarlos en la rutina $Gscale$, por lo cual se han dimensionado. La anchura de las barras es de 0.7 y la separación entre c/u es de 0.3.

Gdig (NOM, STA, SW, N) es una rutina que se utiliza para digitalización de imágenes a través del video. El parámetro NOM se refiere al nombre del archivo en el cual se grabarán las coordenadas digitalizadas; STA indica el status de este archivo, es decir si el archivo es nuevo o bien ya existe. El parámetro SW posibilita la opción de dibujar un punto sobre cada coordenada digitalizada. N es el número de coordenadas digitalizadas, y es devuelto al programa principal desde la rutina.

El nombre del archivo se ajusta a las características que tiene el sistema VAX, archivo de tipo secuencial, con una longitud de registro de 8 bytes. El modo de acceso es 'sequential' cuando el status del archivo es 'new', y es 'append' cuando el status es 'old'. El parámetro STA sólo puede tener dos valores 'new' cuando el archivo se va a usar por vez primera y 'old' cuando ya existe en disco. Estos dos primeros parámetros son de tipo carácter, el primero es de longitud variable, y el segundo es de tres bytes de longitud. El parámetro SW puede tener los valores de 1 o de 0 según si se desea o no activar la opción, en caso de ser una variable, debe ser de tipo "entero". N también debe ser de este tipo.

La digitalización puede realizarse seleccionando arbitrariamente puntos en el video, o bien colocando sobre el video una reproducción, en papel transparente, de la imagen a digitalizar. Cuando se accesa la rutina de digitalización, se despliega sobre el video un recuadro sobre los bordes de la pantalla. Este recuadro indica el area útil de digitalización, más allá de la cual no se reportan coordenadas válidas. También

se despliega un cursor intermitente de brazos largos, en la esquina inferior izquierda, señalando la coordenada origen (0,0) de la pantalla; los límites de la pantalla corresponden al vértice opuesto de coordenadas (767,479). El cursor se posiciona en la localidad deseada mediante las teclas de dirección. Una forma de posicionar más rápido el cursor, es oprimiendo simultáneamente la tecla SHIFT (de mayúsculas) y la tecla de dirección deseada, esto hace avanzar al cursor en incrementos de 10 pixels. Una vez posicionado el cursor, se procede a reportar sus coordenadas hacia el archivo respectivo; esto se hace oprimiendo la tecla de RETURN. Cada vez que se oprime esta tecla, se envía una coordenada (x,y) al archivo, las coordenadas así reportadas son grabadas en el archivo, a partir de la primera columna, con el siguiente formato : ### ## (esto es equivalente a xxx yyy respectivamente). Para abandonar la rutina una vez que se ha terminado de digitalizar, se mueve el cursor hasta los extremos físicos del video, es decir más allá de los los bordes inferior y derecho, en donde el cursor no puede avanzar más ni hacia la derecha, ni hacia abajo, aun cuando se estén oprimiendo las teclas de dirección. En tal posición, se teclea RETURN y se abandona la rutina.

Ejemplo

```
CALL Gonoff('mgraf',1)
CALL Gbp
CALL Gdig('cuadrado','new',1,n)
CALL Gonoff('mgraf',0)
PRINT *,'EL NUMERO DE COORDENADAS DIGITALIZADAS ES DE :',n
```

En este ejemplo se digitaliza un cuadrado, cuyas coordenadas de los vértices se graban en el archivo cuadrado.dat.

P1

P2

P4

P3

S E T - U P

La microcomputadora GIGI puede trabajar en forma local o en línea conectada a la minicomputadora VAX. GIGI permite especificar una serie de parámetros de control llamados SET-UP's, mediante los cuales es posible adaptarla a diversas situaciones de operación. Para trabajar en línea con la VAX o trabajar en forma local es necesario especificar ciertos SET-UP's. Estos pueden ser especificados en forma directa mediante la tecla de SET-UP o mediante alguna rutina de programación. Los más útiles son presentados a continuación :

BA0	deshabilita GIGI BASIC
BA1	habilita GIGI BASIC modo local
BA2	habilita GIGI BASIC controlado via VAX on-line
CKn	on/off de las secuencias de escape enviadas por las teclas de dirección
KCn	on/off del sonido del teclado
SM0	scrolling deshabilitado
SM1	scrolling rápido
SM2	scrolling lento
RVn	on/off de video inverso
VC0	cursor invisible
VC1	cursor de texto
VC2	cursor gráfico
VC3	habilita cursor gráfico y de texto
LEn	on/off del eco de caracteres
GD0	interpretación de comandos gráficos sin texto informativo de cual se interpreta
GD1	despliegue de comandos gráficos mediante texto sin ejecución del comando
GD2	despliegue de comandos de graficación como texto y además se ejecuta el comando
GPn	on/off de la interpretación del caracter ! como gráfico
KRn	on/off de la autorrepetición de caracteres
KP1	transmisión de secuencias de escape de aplicación
KP0	transmisión normal de código del teclado
LLO	estación GIGI trabajando en modo local
LL1	estación GIGI trabajando en línea
PK0	habilita las teclas del teclado auxiliar para transmitir secuencias de escape standard
PK1	habilita las teclas del teclado auxiliar para

transmitir secuencias de escape específicas de cierta aplicación

RSn

velocidad de recepción de GIGI, donde para cada valor de n se tiene una tabla de correspondencia siguiente :

0 - 110	1 - 300	2 - 600	3 - 1200
4 - 2400	5 - 4800	6 - 9600	7 - 19200

TSn

velocidad de transmisión de GIGI donde n es la misma de RSn

INTEGRACION DE G A LOS DIVERSOS LENGUAJES

G permite trabajar con cualquier lenguaje soportado en la minicomputadora VAX, para lo cual se requiere realizar pequeños procedimientos, muchos de los cuales son tan comunes como ligar rutinas, con el fin de generar el ambiente adecuado para la ejecución de un programa.

Presentaremos tales procedimientos para la ejecución de G a través de FORTRAN 77, Pascal y Basic, así como ejemplos de aplicación para cada caso.

Las rutinas de G pueden emplearse utilizando como anfitrión cualquiera de los lenguajes disponibles en VAX. Los mecanismos para su integración, como llamadas desde cada lenguaje, poseen algunas particularidades, y cada una de ellas será tratada.

FORTRAN y G

Este lenguaje constituye el anfitrión con el cual se logra la comunicación más natural. No se requiere de ningún elemento adicional para realizar llamadas desde éste. Las subrutinas sólo requieren una especificación adecuada de los parámetros que las forman por naturaleza propia.

Una vez escrito un programa en este lenguaje, el cual hace referencia a algunas rutinas de G, se sigue el siguiente procedimiento :

- Compilación del programa fuente
- Ligado del programa objeto resultante con el módulo GCOM.LIG



DEPFI

La secuencia anterior produce un código ejecutable listo para representar en forma gráfica las acciones programadas por el usuario.

PASCAL y G

La graficación por computadora es uno de los campos de la computación que justifica la existencia de procesadores ampliamente poderosos, dado que el volumen de cálculo que involucra una aplicación de esta índole es considerable, pero si bien, el hardware especializado es importante, el software ocupa también un lugar importante debido a las estructuras de datos que en particular son manipuladas en cada lenguaje.

Pascal cuenta con estructuras dinámicas que permiten implementar eficientemente una gran variedad de aplicaciones de graficación, por lo cual diversos usuarios lo han elegido como el lenguaje de implementación gráfica.

El mecanismo requerido para manejar las llamadas a G desde PASCAL es el siguiente :

- a) En la sección de constantes, incluir la constante GLONG con el valor máximo que se requerirá para los arreglos de graficación enteros.
- b) En la sección de TYPE, incluir %INCLUDE 'GTYPESG.PAS/NOLIST'
- c) En la sección de VAR, incluir %INCLUDE 'GCALLS.PAS/NOLIST'

Después de realizar estas asignaciones y compilado el programa fuente, realizamos la etapa de ligado con el módulo GCOM.LIG, lo cual producirá el módulo ejecutable deseado.

BASIC y G

Las llamadas a G desde este lenguaje requieren dos elementos importantes; primero, la declaración explícita del tipo de dato de las variables que pasaremos como parámetros a las rutinas de G; y como segundo elemento, la especificación del paso de parámetros, el cual puede ser de tres tipos : por valor, por referencia y por descripción.

El paso de parámetros diferentes a una cadena de caracteres debe realizarse por referencia. Para cadenas de caracteres, el paso debe realizarse por descripción.

Veamos el siguiente ejemplo :

```

10  DECLARE STRING C
20  DECLARE LONG  A
30  A = 1
40  C$ = "mgraf"
50  CALL GONOFF(C BY DESC,A BY REF)
60  CALL GP  BY REF (300%,200%+A)
70  CALL GONOFF("mgraf" BY DESC,0% BY REF)
80  END

```

En el ejemplo anterior, declaramos las variables C y A que utilizamos en las llamadas a GONOFF y GP. Debe notarse que la variable tipo "string" C fue pasada por descripción, mientras que los demás parámetros fueron pasados por referencia.

Además, se observa que las constantes numéricas deben escribirse con el signo de % al final.

Todos los elementos anteriores son necesarios para lograr compatibilidad de parámetros entre BASIC y las rutinas de G.

Después de compilar el programa fuente anterior, al momento de ligarlo, lo hacemos con el módulo GCOM.LIG y generamos el archivo ejecutable correspondiente.

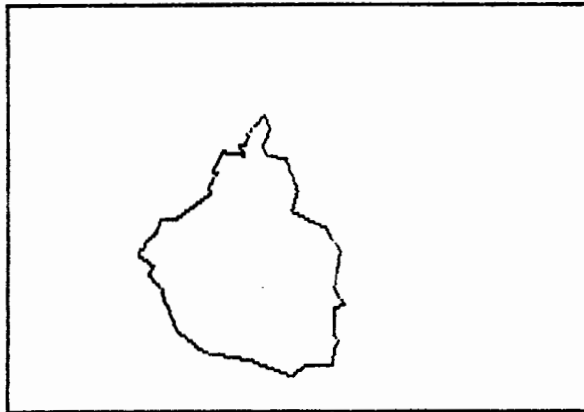
Como elemento de ayuda para los usuarios de G, anexamos una serie de ejemplos realizados en cada uno de los lenguajes anfitriones mencionados.

```

C -----
C      EJEMPLIFICACION DE G TOMANDO FORTRAN COMO ANFITRION
C
C EL PROGRAMA TOMA LAS COORDENADAS GENERADAS POR LA RUTINA DE DI-
C GITALIZACION (GDIG), Y RECONFIGURA LA IMAGEN ORIGINAL A ESCALA
C SOBRE UN PUERTO DEL VIDEO, DEFINIDO POR LOS PUNTOS (100,100) y
C (500, 400).
C
C EN ESTE EJEMPLO SE OBSERVA QUE PARA LLAMAR A LAS RUTINAS NO SE
C NECESITAN DECLARACIONES ESPECIALES, YA QUE LAS RUTINAS DE G ES-
C TAN ESCRITAS EN FORTRAN, UNICAMENTE ES OBLIGATORIO QUE LOS TI-
C POS DE LOS PARAMETROS, Y LOS TIPOS DE LAS VARIABLES EMPLEADAS
C EN EL PROGRAMA PRINCIPAL, SEAN CONCORDANTES.
C -----
C
C DECLARACION DE VARIABLES
C
C      REAL X(100),Y(100)
C      INTEGER XS(100),YS(100)
C
C DIGITALIZACION DE LA IMAGEN. LAS COORDENADAS SON GRABADAS EN EL
C ARCHIVO 'MEXICO.DAT'
C
C      CALL GONOFF ('mgraf',1)
C      CALL GBP
C      CALL GDIG ('MEXICO','NEW',1,N)
C
C LECTURA DE LAS COORDENADAS DE LOS VERTICES DIGITALIZADOS
C
C      OPEN (7,FILE='MEXICO.DAT',STATUS='OLD')
C      READ (7,*)(X(I),Y(I),I=1,N)
C
C ESCALAMIENTO Y DIBUJO DE LA IMAGEN DIGITALIZADA
C
C      CALL GSCALE (100,100,500,400,0,0,767,479,50,X,Y,XS,YS)
C      CALL GBP
C      CALL GREC (100,100,500,400)
C      CALL GLINES (XS,YS,N,5)
C      CALL GPOS (100,50)
C      CALL GTEXT (' ',Distrito Federal y Zona Metropolitana')
C      CALL GONOFF ('mgraf',0)
C      END

```

El programa anterior produce la siguiente salida :



Distrito Federal y Zona Metropolitana

```

{-----}
{ EJEMPLIFICACION DE G UTILIZANDO PASCAL COMO ANFITRION }
{ }
{ EL PROGRAMA TDIN PERMITE DESPLEGAR UNA MALLA INFLUENCIADA POR }
{ VARIOS PUNTOS DE CONTROL. }
{ LOS VALORES GENERADOS PARA LA MALLA SE ENCUENTRAN ALMACENADOS }
{ EN LA MATRIZ MGRAF. }
{ LAS COORDENADAS DEL MUNDO REAL MAPEADAS A UN PUERTO DEL VIDEO }
{ SE ENCUENTRAN ALMACENADAS EN LA MATRIZ MC. }
{ LAS VARIABLES POS REPRESENTAN LAS POSICIONES DE INFLUENCIA }
{ DE LOS PUNTOS DENTRO DE LA MALLA. }
{ LAS VARIABLES AMPLIF INDICAN LA AMPLIFICACION PARA CADA PUNTO }
{ DE INFLUENCIA. }
{ ATEN INDICA EL GRADO DE ATENUACION QUE UN PUNTO PROVOCA SOBRE }
{ LOS DEMAS. }
{-----}

```

```

PROGRAM TDIN(INPUT,OUTPUT);
CONST MALLA = 25;      {LINEAS QUE FORMAN LA MALLA EN UN SENTIDO}

```

```

{ EN LA REGION DE CONSTANTES EL USUARIO DEBE ESPECIFICAR LA
CONSTANTE GLONG. SU VALOR DEPENDE DEL NUMERO DE ELEMENTOS QUE
TENDRAN LOS ARREGLOS DE COORDENADAS QUE SE UTILIZARAN EN LAS
LLAMADAS DE G. }

```

```

    GLONG = 25;

```

```

TYPE  CPUNT = ARRAY [1..4] OF INTEGER;
      CORD  = RECORD CX,CY : INTEGER
              END;

```

```

{ EN LA REGION DE TYPE, DEBE ESPECIFICARSE EL INCLUDE
GTYPESG.PAS}

```

```

    %INCLUDE 'GTYPESG.PAS/NOLIST'

```

```

VAR  MGRAF   : ARRAY [1..MALLA,1..MALLA] OF REAL;
      MC      : ARRAY [1..MALLA,1..MALLA] OF CORD;
      POS22,POS33,MITAD2,MITAD,POS2,POS3,I,J : INTEGER;
      ATEN,AMPLIF,MODIF,MODIFF,RAD,AMP2,AMP3 : REAL;
      DATOS   : TEXT;
      OPCION  : CHAR;
      X,Y     : CPUNT;

```

```

{ EN LA REGION DE VARIABLES, DEBE INCLUIRSE EL INCLUDE
GCALLS.PAS}

```

```

    %INCLUDE 'GCALLS.PAS/NOLIST'

```

```

FUNCTION PE(POSIS,POS2:REAL;I,J:INTEGER):REAL;
{DETERMINA EL VALOR NETO DE LA FUNCION EN UN PUNTO}
BEGIN
```

{LAS SIGUIENTES RUTINAS DETERMINAN EL MAPEO DE COORDENADAS DEL MUNDO REAL A COORDENADAS DE UN PUERTO DEL VIDEO}

FUNCTION ECP(IJ:INTEGER; P1,P2:REAL):REAL;EXTERN;

FUNCTION PENDIENTE(X1,Y1,X2,Y2:REAL;EXTERN;

PROCEDURE MAPEA(I,J:INTEGER;X,Y:CPUNT;VAR XM,YM:INTEGER;);EXTERN;

PROCEDURE GRAFMAT;

{DESPLIEGA LA MALLA GENERADA}

VAR I,J,X1,Y1,X2,Y2: INTEGER;

BEGIN

GONOFF('MGRAF',1); { ENTRA A MODO GRAFICO }

GCOLOR(7); { SELECCIONA COLOR BLANCO }

GBP; { BORRA VIDEO }

FOR I:= 1 TO 3 DO { DESPLEGANDO BASE }

GLINE(X[I],ROUND(Y[I]),X[I+1],ROUND(Y[I+1]));

GLINE(X[4],ROUND(Y[4]),X[1],ROUND(Y[1]));

GLINE(X[1],ROUND(Y[1]),X[1],ROUND(Y[1]+70));

FOR I:= 1 TO MALLA DO { DESPLEGANDO MALLA }

BEGIN

GPOS(MC[I,1].CX,ROUND(MC[I,1].CY+MGRAF[I,1]));

FOR J:= 1 TO MALLA-1 DO

B

```
{DETERMINACION DE COORDENADAS EN VIDEO}
```

```
FOR I:= 1 TO MALLA DO
```

```
FOR J:= 1 TO MALLA DO
```

```
  BEGIN
```

```
    MAPEA(I,J,X,Y,XM,YM);
```

```
    MC[I,J].CX:=XM
```

```
    MC[I,J].CY:=YM
```

```
  END;
```

```
END;
```

```
{PROGRAMA PRINCIPAL}
```

```
BEGIN
```

```
  OPCION:='S';
```

```
  WHILE OPCION <> 'N' DO
```

```
    BEGIN
```

```
{LECTURA DE POSICIONES DE INFLUENCIA, AMPLIFICACION Y ATENUACION}
```

```
  WRITE('POS(123) ,AMP(123) Y ATENUACION :');
```

```
  READ(MITAD,MITAD2,POS2,POS22,POS3,POS33,AMPLIF,AMP2,AMP3,ATEN);
```

```
{GENERACION DE VALORES DE LA MALLA}
```

```
  FOR I:= 1 TO MALLA DO
```

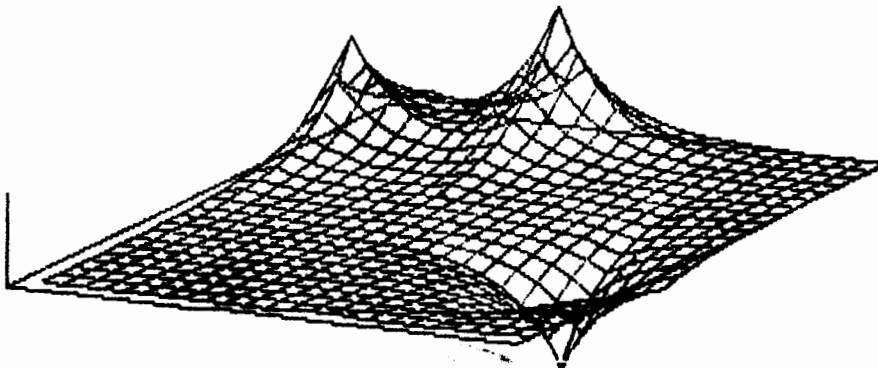
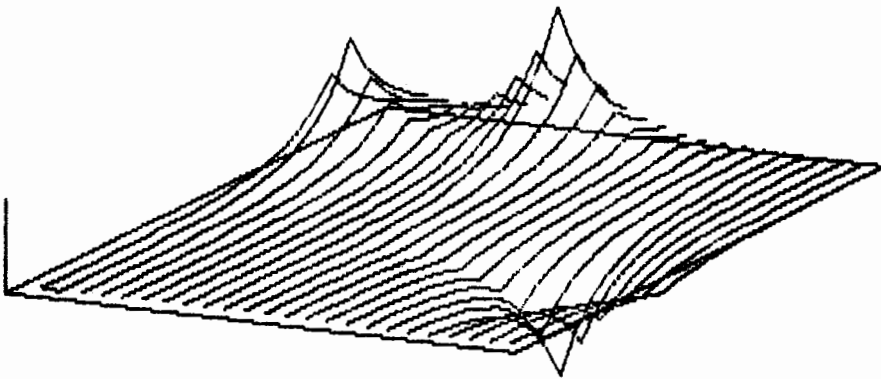
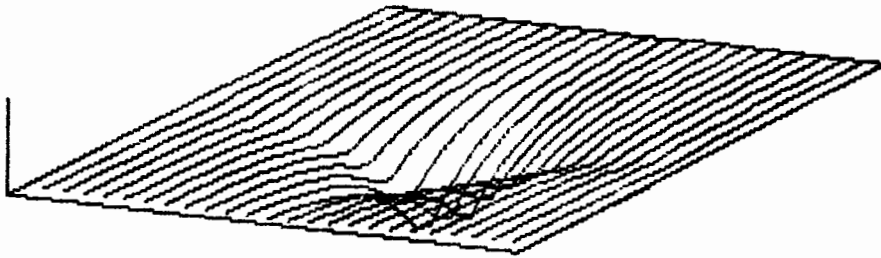
```
    BEGIN
```

```
      FOR J:= 1 TO MALLA DO
```

```
        BEGIN
```

```
          MGRAF[I,J]:=AMPLIF*PE(MITAD,MITAD2,I,J)+AMP2*PE(POS2,POS22,I,  
J)+AMP3*PE(POS3,POS33,I,J);
```

Salida del programa anterior :

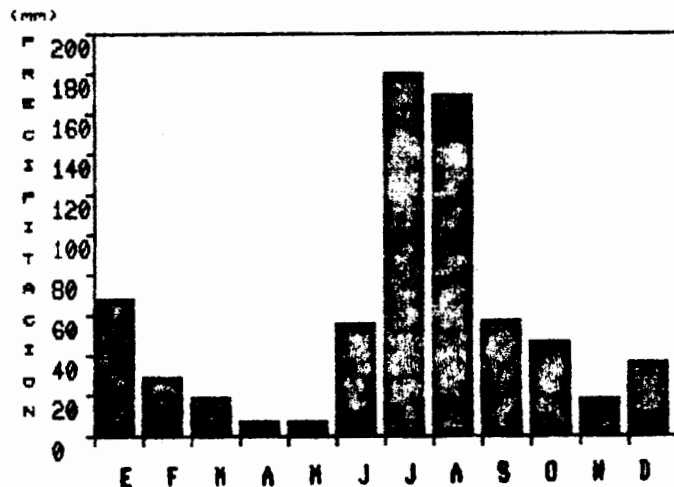


```

1 REM-----
2 REM      EJEMPLIFICACION DE G USANDO BASIC COMO ANFITRION.
3 REM
4 REM EL PROGRAMA CONSTRUYE UNA GRAFICA DE BARRAS PARA REPRESENTAR
5 REM LA MARCHA ANUAL DE LA PRECIPITACION PLUVIAL EN LA ESTACION
6 REM METEOROLOGICA DE CREEL, CHIH.
7 REM
8 REM EN ESTE EJEMPLO ESTAN INCLUIDAS TODAS LAS DECLARACIONES Y
9 REM DIRECTIVAS NECESARIAS PARA LLAMAR A LAS RUTINAS DE G.
10 REM-----
20 DECLARE REAL M(12),P(12)
30 DECLARE LONG ME(12),PE(12)
40 CALL GONOFF('MGRAF' BY DESC,1% BY REF)
50 CALL GBP
60 CALL GAXIS BY REF(100%,100%,500%,400%,0%,0%,12%,200%,0%,1%,1%,
    20%,1%,0%,1%)
70 CALL GPOS BY REF(116%,70%)
80 CALL GCOL BY REF(5%)
90 CALL GTEXT BY DESC("[33]SO","EFMAMJJASOND")
100 CALL GPOS BY REF(50%,400%)
110 CALL GTEXT BY DESC("[,23]SO","PRECIPITACION")
120 CALL GPOS BY REF(40%,420%)
130 CALL GTEXT BY DESC("ES0","(mm)")
140 OPEN "CREEL.DAT" FOR INPUT AS FILE #1
150 FOR I = 1 TO 12
160 INPUT #1,P(I)
170 M(I)=I
180 NEXT
190 CALL GSCALE BY REF(100%,100%,500%,400%,0%,0%,13%,200%,M(1),
    P(1),ME(1),PE(1))
200 PRINT "W(V)"
210 PRINT "W(O)"
220 CALL GBAR BY REF(0.8,PE(1),12%,100%,100%,500%,0.2,1%,5%)
230 CALL GPOS BY REF(100%,450%)
240 CALL GTEXT BY DESC("ES0","MARCHA ANUAL DE LA PRECIPITACION.
    CREEL, CHIH.")
250 CALL GONOFF("MGRAF" BY DESC,0% BY REF)
260 CLOSE #1
270 END

```

MARCHA ANUAL DE LA PRECIPITACION, CREEL, CHIH.



F-DEPFI/D- 62/1986/Ej.2



711485

F-DEPFI
D-62
1986
Ej.2 ✓

G(2)- 11485

NOTA: EL CONTENIDO EXPRESADO EN ESTA OBRA ES RESPONSABILIDAD DE LOS AUTORES.