

2.3



Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

IMPLEMENTACION DEL PAQUETE DE GRAFICACION
PLOT-10 PARA LAS TERMINALES TEKTRONIX 4006
EN LAS COMPUTADORAS BURROUGHS DE LA UNAM.

T E S I S

Que para obtener el Título de
A C T U A R I O

presenta

MARIA LUISA MELANIA ALVAREZ VILLAFUERTE

México, D. F.

1983



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INDICE.

	Pag
Introducción	1
Capítulo I. Introducción a los principios de graficación por computadora.	3
Capítulo II. Principios del funcionamiento en pantalla y operación básica de la terminal TEKTRONIX 4006-1	12
Capítulo III. Descripción y uso del paquete de graficación PLOT 10.	21
Capítulo IV. Ejemplos	46
Conclusiones	70
Bibliografía	71

Introducción.

Al realizarse cierto tipo de experimentos se obtienen datos cuya graficación facilita su estudio y análisis. Esto ocurre en varias dependencias de la UNAM, principalmente en las del área científica. Por ejemplo, en el Instituto de Astronomía se grafican datos de estrellas para llevar a cabo una interpretación más adecuada de ellos. Sin embargo, hemos visto que en la mayoría de los casos la graficación se hace con métodos rudimentarios.

El objetivo de esta tesis es el de contribuir a facilitar este procedimiento, poniendo a disposición de personas con conocimientos elementales de computación una serie de subrutinas que permiten una graficación más rápida, precisa y variada.

Se escogió este paquete entre otros ya que sobresale por su eficiencia y rapidez. Además, este paquete está implementado en minicomputadoras existentes en algunas instituciones de la UNAM, como la PDP 11 y las NOVAS (Institutos de Astronomía y Geofísica).

Para la elaboración de este paquete se contó con la ayuda de algunas subrutinas de Astronomía que ya se tenían adaptadas para la BURROUGHS y las NOVAS, además de las subrutinas originales del paquete.

El trabajo desarrollado consistió en la implementación de un paquete conocido. Se realizó lo siguiente:

1. Adaptación de las subrutinas a las necesidades de la máquina con la que se trabajó.

2. Compilación.
3. Corrección de errores de subrutinas.
4. Reprogramación de algunas subrutinas.

La labor anterior se llevó a cabo en aproximadamente 80% del paquete original.

I. INTRODUCCION A LOS PRINCIPIOS DE GRAFICACION POR COMPUTADORA.

1.1 Principios de Graficación por Computadora

La graficación mediante computadoras envuelve todo aquello que sea la manipulación, generación, procesamiento y evaluación de objetos gráficos mediante una computadora, así como la asociación de la información no gráfica que se relaciona con él, residente en archivos de la computadora.

Los objetos gráficos pueden ser imágenes fotográficas o pueden ser creadas con la ayuda de la computadora en forma de caracteres alfanuméricos (A/N), signos especiales, dibujo de líneas o sombreado de áreas.

El graficado mediante computadora se divide principalmente en tres partes:

1. Graficas generativas
2. Análisis de imagen
3. Gráficas cognoscitivas.

1.1.2 Gráficas generativas:

Tratan de la creación de gráficas artificialmente.

Las tareas principales de graficas generativas (GG) son:

1. Construcción del modelo (objeto) y generación de su dibujo.
2. Modelo y transformación del dibujo.
3. Identificación del objeto, recuperación y cambio

en su información.

Por modelo se entiende la descripción abstracta de un objeto gráfico la cual puede ser entendida por la computadora y transformada en la correspondiente imagen en la pantalla.

La imagen es la visualización concreta de un modelo.

Las herramientas usadas en las gráficas generativas son graficadores y sistemas iterativos de salida de imagen.

Graficadores:

Son medios de salida de gráficas exclusivamente i.e. no existe la interacción del usuario con el sistema.

Sistemas iterativos de salida de imagen:

Permite al usuario obtener una representación de la información en forma gráfica e interactuar directamente con el sistema creando, manipulando y diseñando objetos gráficos mediante la consola.

La consola posee una superficie donde se muestra la imagen que, en la mayoría de los casos, es un tubo de rayos catódicos, a esta superficie se la llama pantalla.

1.1.2 Procesamiento de gráficas y su análisis:

Trata de las representaciones discretas de imágenes en forma de un arreglo de números los cuales son representaciones de puntos, sombreados, etc. dependiendo de lo que deseemos graficar.

Las tareas principales del procesamiento y análisis de imágenes son:

1. Filtración de la imagen.
2. Evaluación de una imagen.
3. Reconocimiento de patrones.

1.1.3 Gráficas cognoscitivas.

Trata de modelos abstractos de gráficas y las relaciones entre ellos.

Un modelo abstracto es la idea de un objeto sin tomar en cuenta su apariencia instantánea.

El área de gráficas cognoscitivas es el área de gráficas mixtas donde los campos de generación y proceso de imágenes están unidos.

La distinción entre graficadores y sistemas iterativos de salida de imagen estriba en el modo de operación. Vemos que los aspectos iterativos son el tópico crucial en la generación de gráficas del cual el aspecto de graficador forma sólo una parte.

1.2 Dispositivo de salida de imágenes.

El medio más usado para mostrar las gráficas mediante las computadoras es el tubo de rayos catódicos (CRT) que es además el medio utilizado por el paquete de graficación implementado y descrito en esta tesis.

Otro medio también utilizado es el del panel de plasma.

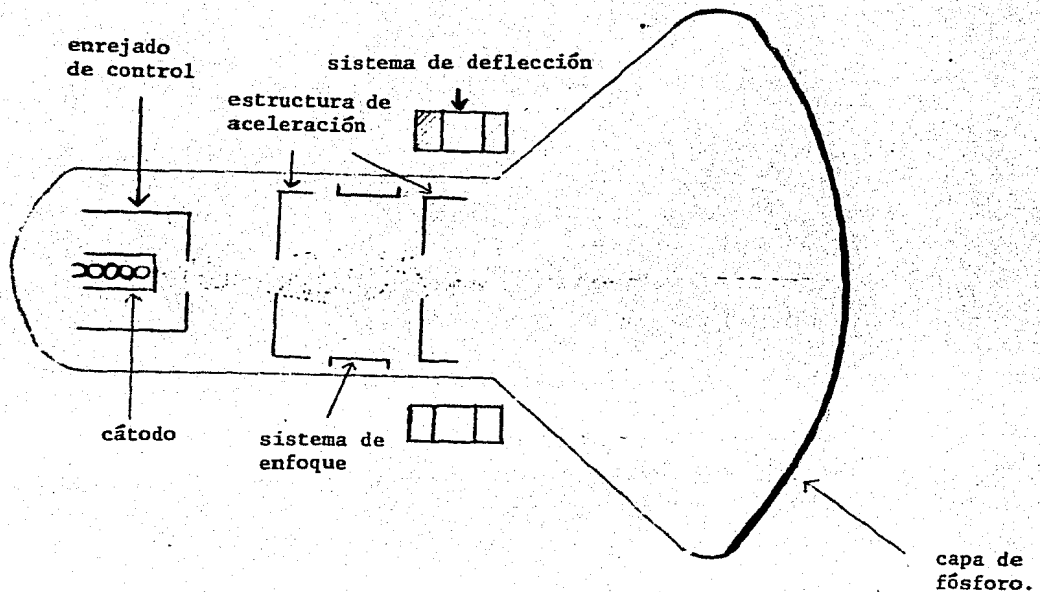
1.2.1 Tubo de Rayos Catódicos:

El tubo de rayos catódicos depende para su operación de dos fenómenos distintos:

1. La fosforescencia de ciertas sustancias para que al ser bombardeada por electrones emita una luz durante un cierto tiempo después del cual deja de emitirse esta luz.

2. El efecto del campo eléctrico en el movimiento de electrones, el cual acelera los electrones a una alta velocidad y los hace converger en una fina partícula que es llevada a la sustancia fosforescente y que al chocar con ésta emite una luz o fosforescencia.

Fig. Tubo de Rayos Catódicos.



1. Cátodo: al ser calentado emite electrones.
2. Rejilla de control: controla la dirección y la cantidad de electrones emitidos.
3. Estructura de aceleramiento: produce una alta velocidad en la partícula de electrones.
4. Sistema de convergencia: asegura que la partícula de electrones converja en una fina mancha cuando se estrelle en la pantalla.
5. Sistema de deflexión: es el sistema que direcciona la partícula compuesta de electrones.
6. Capa de fósforo: la cual brilla cuando la partícula se estrella en ella

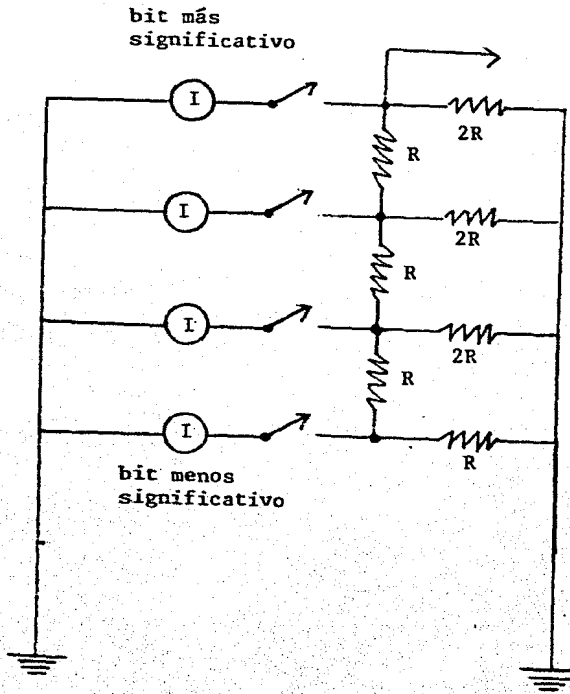
Todos estos componentes están encerrados en una botella cónica al vacío.

1.3 Graficado de puntos:

Graficar puntos es la forma de salida más simple que puede ser generada en respuesta a una computadora.

El usuario puede especificar el punto que desea graficar dando las coordenadas en que desea que éste sea graficado.

Al querer graficar un punto las dos coordenadas pasan a través de un control de graficado y estas coordenadas son convertidas en voltajes analógicos mediante un convertidor analógico digital para poder ser entendidos y usados por el sistema de deflexión del tubo de rayos catódicos.



Un convertidor analógico digital.

En uno de los registros del control del sistema de deflección se especifica el control de la brillantez de los puntos.

Las dos etapas en el proceso de graficar un punto son:

- Cargando el registro "x" y "y", y el paso por el convertidor analógico digital.

- Intensificación del punto.

1.4 Generación de vectores.

Al graficar vemos la gran necesidad de generar líneas en pantalla.

Existen dos caminos básicos en la generación de vectores:

- Los puntos de la línea son computarizados digitalmente y la línea es dibujada como una serie de punteos.
- Un circuito analógico es usado para mover la partícula o haz de electrones desde el punto inicial, hasta el punto final.

Ambos métodos de generación de vectores tienen como fin:

- que la línea deba ser derecha.
- la línea debe comenzar y terminar donde el programador lo requiere.
- la intensidad de la línea debe ser independiente de la longitud y la inclinación de la línea y la brillantez deben ser uniformes a lo largo de la línea.

La generación de una línea debe ser ejecutada muy rápidamente para que muchas líneas puedan ser graficadas sin ningún parpadeo objetable.

1.4.1 Ecuación de una línea

Para la generación de una recta se usan diferentes formulaciones.

En primer lugar los datos que poseemos son el punto inicial (X_I, Y_I) y el punto final (X_F, Y_F) .

$$\text{Tenemos además que } \Delta X = X_F - X_I \quad \Delta Y = Y_F - Y_I$$

Las formulaciones en sí son:

1.-Fórmula diferencial

$$\frac{dY}{dX} = \frac{\Delta Y}{\Delta X}$$

2.-Fórmula integral

$$X = X_I + \int \frac{\Delta X}{T} d\epsilon \quad Y = Y_I + \int \frac{\Delta Y}{T} d\epsilon$$

donde el rango de $0 \leq t \leq 1$ graficará la línea

3.Fórmula exponencial paramétrica.

$$X = X_I + \Delta X(1 - e^{-t/T})$$

$$Y = Y_I + \Delta Y(1 - e^{-t/T})$$

donde el rango de $0 \leq t \leq +\infty$ graficará la línea.

4. Fórmula lineal paramétrica

$$X = (1 - \alpha)X_I + \alpha X_F$$

$$Y = (1 - \alpha)Y_I + \alpha Y_F$$

$$0 \leq \alpha \leq 1$$

1.4.2

Estimación de la longitud de las líneas,

Los algoritmos de generación de vectores estiman la longitud del vector que grafican.

Y uno de sus usos en el método de generación digital es para ver que la línea termine después de un número apropiado de punteos.

Sabemos que la fórmula de la longitud de línea matemáticamente está dada por:

$$\sqrt{(X_F - X_I)^2 + (Y_F - Y_I)^2}$$

Pero para computación existen métodos más simples como los siguientes:

- 1.-Largo de las deltas $\text{MAX}(/ΔX/, /ΔY/)$
2. Longitud de la delta máxima más la mitad de la longitud de la delta mínima

$$\text{MAX}(/ΔX/, /ΔY/) + (\text{Min}(/ΔX/, /ΔY/))/2$$

3. Suma de las deltas.

$$/ΔX/ + /ΔY/$$

4. La potencia más pequeña del excedente de la delta más grande

$$2^i \text{ tal que } 2^{i-1} \leq \text{MAX}(/ΔX/, /ΔY/) < 2^i$$

II PRINCIPIOS DEL FUNCIONAMIENTO DE UNA SALIDA GRAFICA EN PANTALLA Y OPERACION BASICA DE LA TERMINAL TEKTRONIX - 4006-1

2.1 Salida gráfica:

La salida gráfica es un dispositivo en el cual se pueden acceder un número finito de puntos referenciados en forma absoluta con respecto a un origen.

En las pantallas se tienen dos posibilidades para el movimiento del cursor.

1.- Mover el cursor a un punto del dispositivo sin pintar la trayectoria del movimiento.

2.- Mover el cursor a un punto del dispositivo pintando la trayectoria del movimiento.

La orden para cualquiera de estos dos movimientos se indica mandando un caracter al dispositivo.

Las pantallas de graficación pueden clasificarse en pantallas de persistencia y en pantallas de refresco.

Las pantallas de persistencia son pantallas de fósforo - en las que la acción de pintar se consigue mediante la excitación de una zona de la pantalla a través de un haz de electrones.

En las pantallas de refresco se sigue el funcionamiento de una pantalla de televisión. Se tiene un barrido de la pantalla siguiendo un conjunto de líneas horizontales en determinado tiempo y en forma continua.

La diferencia principal entre las dos pantallas de graficación es que en las pantallas de persistencia no pueden cambiarse una parte de la gráfica, debe borrarse completamente la pantalla y graficar nuevamente; en cambio las pantallas de refresco sí puede cambiarse una parte de la gráfica, alterando el dispositivo de memoria de donde se toma la información para el barrido continuo de la pantalla.

La terminal en la que se pondrá en funcionamiento este paquete de subrutinas de graficación será la terminal TEKTRONIX 4006-1 que consta de una pantalla de persistencia y un teclado con el que puede manejarse en forma manual. Además, tiene una interfase RSX23 para manejarla en línea mediante una computadora en caracteres ASCII.

2.2 Operación Básica de las Terminales TEKTRONIX 4006-1.

La terminal 4006-1 es un enlace de comunicación y mecanismo de exhibición en su uso con un amplio rango de sistemas computacionales.

La 4006-1 tiene tres modos básicos de operación:

Alfanumérico, esquematización de gráficas y hard copy.

2.2.1. Modo Alfanumérico.

Cuando se trabaja en modo alfanumérico la pantalla funciona como cualquier terminal convencional. Pero debido a que no es una terminal de refresco, no se pueden mover las líneas escritas en ella, por lo tanto es necesario limpiar la pantalla totalmente cuando esté llena para volver a escribir de

nuevo en ella.

En el modo alfanumérico la terminal es usada para transmitir y/o mostrar cualquiera de los caracteres del teclado im preso.

Existe un cursor que nos muestra en la pantalla nuestra siguiente posición de escritura.

La primera línea que se escribe en la pantalla aparece en la parte mas alta de esta y justificada a la izquierda. La se gunda línea aparece abajo de la primera y justificada también a la izquierda. Así hasta que se escriban 35 líneas. La si guiente línea que se escribe en la pantalla aparecerá en la parte más alta de esta pero justificada a la derecha en un nuevo margen que se encuentra a la mitad de la pantalla.

Cada línea tiene una capacidad para 74 caracteres.

Al llenarse la segunda mitad de la pantalla es necesario limpiarla totalmente para evitar que las nuevas líneas se encimen con las ya escritas. (ver figura 1)

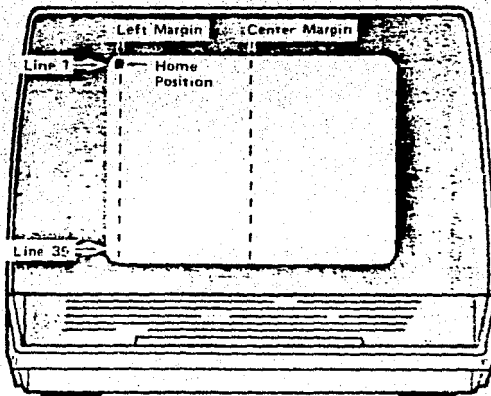


Figura 1.

En el modo alfanumérico la pantalla recibe caracteres de control para varios objetivos utilizando la tecla CTRL junto con otra tecla.

2.2.2 Modo Gráfico.

El modo gráfico provee al usuario con 1048576 puntos que pueden ser direccionados mediante la computadora. De estos - puntos 798720 pueden ser mostrados en la pantalla; los demás pueden ser direccionados pero no mostrados.

Los 1048576 puntos son obtenidos usando 1024 direcciones para el eje vertical (de aquí solo se muestran los primeros - 780) y 1024 direcciones para el eje horizontal. La región visible de la pantalla en este modo de operación es como se muestra en la figura 2.

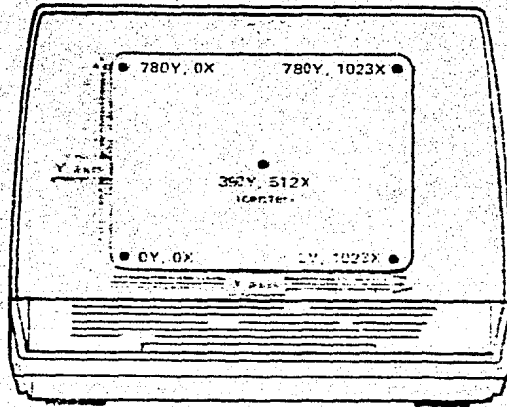
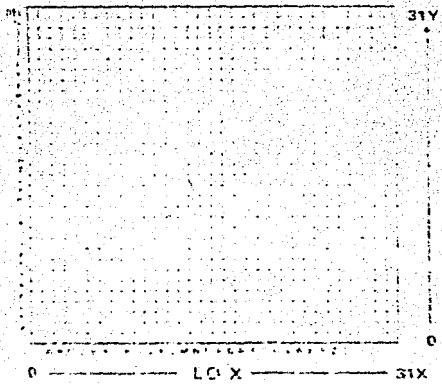
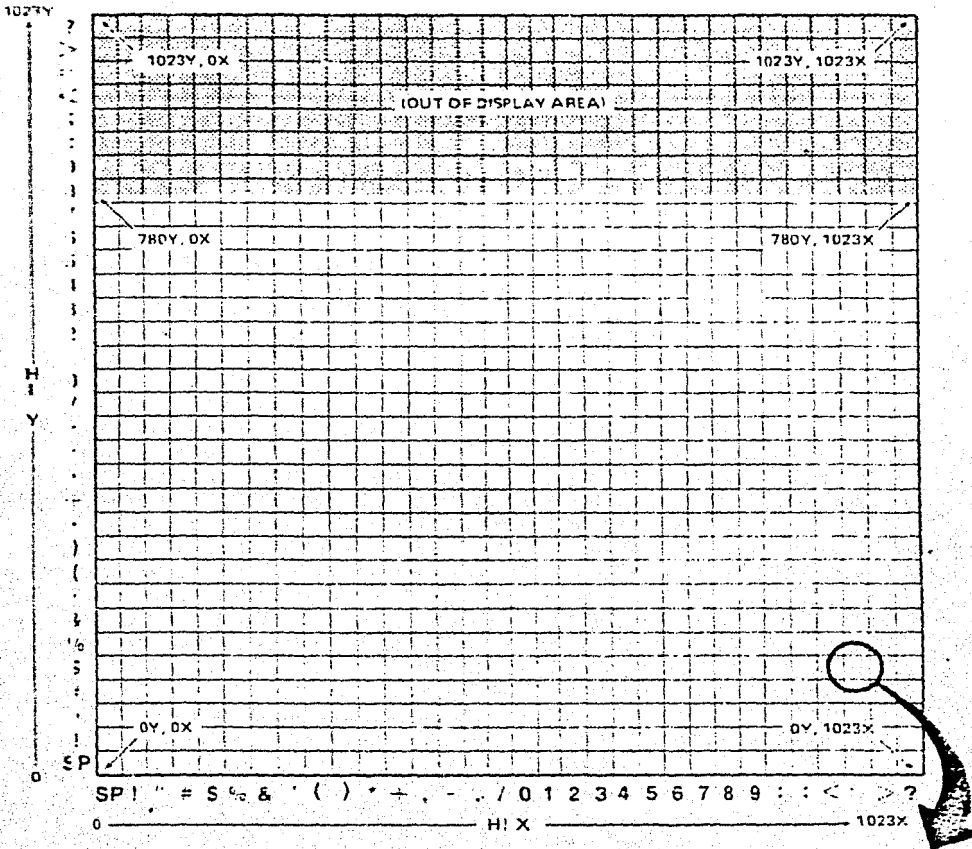


Figura 2.

Cuando la pantalla está en modo gráfico recibe únicamente caracteres de control, que son caracteres ASCII, - que pueden mandarse desde el teclado de la terminal (cuando esta se encuentra en local o echo) o desde una computadora (si está en línea). Dentro de estos caracteres de control se encuentra: un carácter para mandarla a modo gráfico (cuando se enciende queda directamente en modo alfanumérico), otro para regresarlo a modo alfanumérico justificando el cursor al margen de la izquierda o al margen del centro según esté en la mitad izquierda o en la derecha, otro carácter para regresar la pantalla en modo alfanumérico sin regresar el cursor, secuencia de dos caracteres para limpiar la pantalla, etc.

En el modo gráfico se pueden realizar dos tipos de movimientos con el cursor: mover el cursor a un punto de la pantalla sin pintar sobre ella y el otro pintar una línea del punto de la pantalla donde se encuentra el cursor al punto de la pantalla que se le mande.

Los diferentes puntos de la pantalla, se agrupan en una matriz de 32 X 32 elementos y cada elemento de dicha matriz se subdivide en otra matriz de 32 X 32 como se muestra en la figura 3.



Con esto cada punto de la pantalla tendrá asociadas cuatro coordenadas: dos para localizar el elemento de la matriz grande dentro del que se encuentra, coordenadas altas (X,Y) y dos para localizar el elemento de la matriz mas pequeña, que se forma dentro de cada elemento de la matriz grande que le corresponden coordenadas bajas (x,y).

La secuencia para mandar las coordenadas de un punto a la terminal es: Y,y,X,x. Desde el teclado se pueden mandar estas coordenadas mediante sus teclas correspondientes. Desde la computadora, debe mandarse el número ASCII correspondiente a cada tecla.

En algunos casos especiales, si una de las coordenadas de un punto es igual a la coordenada correspondiente del punto anterior, esta coordenada puede omitirse. En la tabla I se muestra cuales coordenadas deben mandarse cuando varían alguna o algunas de las coordenadas de las del punto anterior. Por ejemplo: si la única coordenada que varía es "y", entonces pueden mandarse unicamente las coordenadas "y", "x". Si varían "Y" y "y" pueden mandarse unicamente "Y", "y", "x". Si no varía ninguna coordenada basta con mandar "x".

TABLA 1

Y	y	X	x
*	*	*	*
*	*		*
*			*
	*	*	*
	*		*
			*

* `coordenadas que varían

Por protección a la pantalla, ésta baja su brillantez después de transcurridos 8 ó 9 segundos desde que recibió su última orden. Se puede regresar a su brillantez total oprimiendo cualquier tecla, pero para no afectar lo expuesto en la pantalla o el programa que la esté manejando debe oprimirse la tecla SHIFT.

La máxima velocidad de recepción de datos a la cual se puede desplegar en forma total una línea en estas pantallas es de 2400 baudios. A velocidades mayores no se alcanza a desplegar toda la recta.

Como datos adicionales:

- Un caracter tiene aproximadamente 0.087 pulgadas de ancho por 0.106 pulgadas de alto aproximadamente 11 x 14 puntos.

El espacio entre lineas es de 22 puntos (0.17 pulgadas).

2.2.3 Modo Hard Copy.

El modo hard copy es utilizado para obtener una copia permanente de la información mostrada en la pantalla. En el modo Hard Copy la 4006-1 provee de las señales necesarias para generar una copia de la información. El Hard Copy es inicializado por cualquiera de estas secuencias:

- Presionando la tecla copy en la terminal
- Dar un CTRL SHIFT K seguido por un CTRL W
- Mandando el comando ESC ETB

III DESCRIPCION Y USO DEL PAQUETE DE GRAFICACION PLOT 10

3.1 Inicialización: INITT

El primer paso para poder graficar en la terminal mediante este paquete de graficación, es llamar a la rutina INITT, que es la rutina de inicialización y tiene como finalidad preparar archivos y variables de uso interno para las demás rutinas de graficación.

Esta rutina hace lo siguiente:

- Inicializar variables y banderas comunes de todas las subrutinas.
- La pantalla es borrada y el cursor es puesto en la esquina superior izquierda de la pantalla.
- La terminal se pone en modo alfanumérico.
- Los valores marginales son puestos a la derecha e izquierda de la pantalla.
- La ventana se define, así es que la porción de espacio virtual que será mostrado será equivalente en coordenadas de pantalla.

Ya que el borrado de la pantalla requiere de cierto tiempo, es necesario indicar a la rutina la velocidad con la que se transmiten los datos, la cual será dada por el parámetro IBAUD.

Ejemplo de uso

```
CALL INITT(IBAUD)
```


3.2 Terminación FINITT

Cuando se terminan de usar las rutinas de graficación de este paquete, se llama a la rutina FINITT para que ponga la terminal en modo alfanumérico.

Esta rutina hace lo siguiente

- Pone la terminal en modo alfanumérico.
- Mueve el cursor a las coordenadas de pantalla que se desee.

Ejemplo de uso:

```
CALL FINITT (IX,IY)
```

Parámetros:

IX Coordenada horizontal de pantalla en que se va a dejar el cursor 0 IX 1023

IY Coordenada vertical de pantalla en que se va a dejar el cursor 0 IY 780

3.3 Subrutinas de dibujo en coordenadas de pantalla.

Las tres funciones de dibujo con referencia en coordenadas absolutas de pantalla son MOVABS, DRWABS, PNTABS. El dibujo es absoluto porque las graficaciones en sus movimientos se miden desde un punto fijo que es el origen. Los argumentos de estas rutinas son números enteros donde las coordenadas horizontales pueden ser direccionadas y graficadas del cero hasta el 1023 e igualmente las coordenadas verticales aunque estas últimas sólo son graficadas hasta la coordenada 780

3.3.1 MOVABS:

Esta rutina es solamente de movimiento ya que mueve el cursor a cualquier parte de la pantalla que queramos sin dibujar nada.

Ejemplo de uso:

```
CALL MOVABS (IX, IY)
```

Parámetros:

IX Coordenada horizontal de pantalla a la que se desea mover el cursor

IY Coordenada vertical de pantalla a la que se desea - mover el cursor

Ejemplo:

CALL MOVABS (110,200)

Esta llamada coloca el cursor en la posición (110,200) de la pantalla.

3.3.2 DRWABS:

Esta es la rutina de dibujo la cual pinta una línea - desde la posición en que se encuentre el cursor hasta la posición que se le indica en los parámetros:

Ejemplo de uso

CALL DRWABS (IX,IY)

Parámetros:

IX Coordenada horizontal del punto terminal de la recta.

IY Coordenada vertical del punto terminal de la recta.

Ejemplo:

CALL MOVABS (100,300)

CALL DRWABS (200,500)

Estas llamadas causan en primer lugar un movimiento a la coordenada de pantalla (100,300) y después dibuja una línea desde este punto hasta la coordenada (200,500).

3.3.3 PNTABS:

Esta rutina se mueve a la coordenada dada en el argumento y dibuja un punto ahí.

Ejemplo de uso:

```
CALL PNTABS (IX,IY)
```

Parámetros:

IX Coordenada horizontal del punto que se va a graficar.

IY Coordenada vertical del punto que se va a graficar.

Ejemplo:

```
PROGRAMA QUE DIBUJA UN CUADRADO CON UN PUNTO EN EL -  
CENTRO.
```

```
CALL INITT(4800)  
CALL MOVABS(100,100)  
CALL DRWABS(100,400)  
CALL DRWABS(400,400)  
CALL DRWABS(400,100)  
CALL DRWABS(100,100)  
CALL PNTABS(250,250)  
CALL FINITT(0,0)
```

3.4 Graficado relativo en unidades de pantalla.

En el graficado relativo se da las unidades hacia arriba, abajo, izquierda o derecha que debemos movernos de la posición en que nos encontramos. Los movimientos positivos son hacia la derecha o arriba, los movimientos negativos son hacia la izquierda o abajo dependiendo del parámetro que se trate.

Las rutinas que realizan estas funciones son:

- DRWREL rutina que traza líneas homóloga a DRWABS
- MOVREL rutina de movimiento homóloga a MOVABS
- PNTREL rutina que dibuja puntos homóloga a PNTABS

Ejemplo:

CUADRADO CON UN PUNTO EN EL CENTRO.

Comparemos:

En coordenadas absolutas

```
CALL INITT(2600)
CALL MOVABS(300,200)
CALL DRWABS(500,200)
CALL DRWABS(500,400)
CALL DRWABS(300,400)
CALL DRWABS(300,200)
CALL PNTABS(400,300)
CALL FINITT( 0,757)
```

En coordenadas relativas

```
CALL INITT(2600)
CALL MOVABS(300,200)
CALL DRWREL(200, 0)
CALL DRWREL( 0,200)
CALL DRWREL(-200, 0)
CALL DRWREL( 0,-200)
CALL PNTREL(100,100)
CALL FINITT( 0,757)
```

3.5 Graficaciones en ventana:

En este paquete se puede definir la graficación en ventana la cual consiste en que si el usuario tiene datos de cualquier tipo los puede convertir o traducir a coordenadas de pantalla.

Supongamos que de datos horizontales tenemos números que van del 0 al 10 y de datos verticales números que van del 0 al 100 y se define la ventana que la esquina inferior izquierda es (0,0) y la esquina superior derecha es (10,100) los datos mandados a la pantalla estarán en escala en proporción a esta ventana.

Supongamos que tenemos el dato (2,40) tendremos que será mostrado $(1/10) \times 1024 \times 2 = 204$ unidades de izquierda a derecha

de la pantalla y $(1/100) \times 1024 \times 40 = 409$ unidades de pantalla de abajo a arriba.

Ventana virtual:

Es un espacio rectangular definido en unidades utilizadas por el usuario en el cual se desarrolla el proceso gráfico en pantalla.

Ventana en pantalla:

Es un espacio rectangular colocado en la pantalla físicamente en el cual se ubica la ventana virtual.

Definición de la ventana virtual: VWINDO, DWINDO.

3.5.1 VWINDO:

Con esta rutina se define la ventana virtual dando coordenadas mínimas verticales y horizontales y su ancho.

Ejemplo de uso:

CALL VWINDO(XMIN, XRANGE, YMIN, YRANGE)

Parámetros:

XMIN Coordenada del usuario mínima horizontal.

XRANGE Ancho de la ventana en sentido horizontal en unidades del usuario.

YMIN Coordenadas del usuario mínima vertical.

YRANGE Alto de la ventana en unidades del usuario.

3.5.2 DWINDO:

En esta rutina se define la ventana dando los mínimos y máximos verticales y horizontales.

Ejemplo de uso:

CALL DWINDO(XMIN, XMAX, YMIN, YMAX)

Parámetros:

XMIN Coordenada del usuario mínima horizontal
XMAX Coordenada del usuario máxima horizontal
YMIN Coordenada del usuario mínima vertical
YMAX Coordenada del usuario máxima vertical

Ejemplo:

Las siguientes ventanas son equivalentes y me dan una ventana virtual con coordenada mínima horizontal de 3968 y máxima horizontal de 3972 contra una coordenada mínima vertical de cero y máxima vertical de 100, se ve que la horizontal tiene 4 unidades de ancho de usuario y 100 unidades de usuario de alto.

CALL VWINDO(3068.0,4.0, 0.0, 100.0)

CALL DWINDO(3968.0,3972.0,0.0,100.0)

3.5.3 TWINDO

En esta rutina se define la ventana de pantalla dando los mínimos y máximos de pantalla verticales y horizontales.

Ejemplo de uso:

CALL TWINDO(MINX,MAXX,MINY,MAXY)

Parámetros:

MINX Coordenada mínima de pantalla horizontal
MAXX Coordenada máxima de pantalla horizontal
MINY Coordenada mínima de pantalla vertical
MAXY Coordenada máxima de pantalla vertical.

Ejemplo:

```
CALL TWINDO(20,700,50,400)
```

La ventana dentro de la cual se definirá la ventana vertical tendrá coordenada horizontal mínima de 20 y máxima de 700 y coordenada vertical mínima de 50 y 400 máximas.

3.5.4 SWINDO

En la rutina SWINDO la ventana se define dando las coordenadas mínimas de X y Y y los rangos de altura y anchura.

Ejemplo de uso:

```
CALL SWINDO (MINX,MXRAN, MINY, MXRAN)
```

MINX Coordenada de pantalla mínima horizontal

MXRAN Rango de pantalla horizontal

MINY Coordenada de pantalla mínima vertical

MYRAN Rango de pantalla vertical

Ejemplo:

```
CALL SWINDO (100,600,100,500)
```

La coordenada mínima horizontal es de 100 con un rango de 600 i.e llega hasta la coordenada de pantalla X de 700 y la vertical como mínima es de 100 altura de 500 i.e. 600 de coordenada de pantalla Y.

3.5.5.

Escalamiento y deformación de la ventana en la pantalla.

Llamando a SWINDO o TWINDO se puede modificar el tamaño de la imagen y su posición en la pantalla. Con estas rutinas uno puede hacer que la imagen sea más ancha o más alta depen-

diendo de la ventana horizontal o vertical que demos, de cuanto de la pantalla física queremos abarcar.

3.5.6 Supresión total o parcial de la gráfica en la pantalla.

Con las rutinas VWINDO Y DWINDO se puede lograr la supresión total o parcial de una gráfica, al darle a éstas los parámetros que únicamente incluyan la parte de la gráfica deseada de los límites marcados por el usuario para definir la ventana virtual.

3.6 Graficado absoluto en coordenadas de la ventana virtual.

Para poder graficar en las coordenadas del usuario dentro de la ventana virtual se utilizan las siguientes rutinas:

- MOVEA análoga a MOVABS pero las unidades son las unidades del usuario.
- DRAWA análoga a DRWABS pero las unidades son las unidades del usuario.
- POINTA análoga a PNTABS pero las unidades son reales y son las unidades del usuario.

Ejemplo de uso:

CALL MOVEA (X,Y)

CALL DRAWA (X,Y)

CALL POINTA (X,Y)

Parámetros:

X Coordenada horizontal real (unidades del usuario)

Y Coordenada vertical real (unidades del usuario)

3.7 Graficadõ relativo en unidades del usuario.

Para poder hacer un graficadõ relativo en unidades de usuario tenemos las siguientes rutinas:

- MOVER análoga a MOVREL
- DRAWR análoga a DRWREL
- POINTR análoga a PNTREL

Ejemplo de uso:

CALL MOVER (X,Y)

CALL DRAWR (X,Y)

CALL POINTR (X,Y)

Parámetros:

X Desplazamiento horizontal en unidades del usuario, positivo a la derecha y negativo a la izquierda.

Y Desplazamiento vertical en unidades del usuario, positivo arriba y negativo hacia abajo.

3.8 Dibujo de líneas punteadas.

Si deseamos trazar líneas punteadas de una coordenada a otra existen cuatro rutinas en este paquete que pueden hacerlo, estas son:

- DSHABS análoga a DRWABS
- DSHREL análoga a DRWREL
- DASHA análoga a DRAWA
- DASHR análoga a DRAWR

Ejemplo de uso:

CALL DSHABS(IX, IY, L)

CALL DSHREL(IX, IY, L)

CALL DASHA(X,Y,L)

CALL DASHR(X, Y, L)

Parámetros:

- IX Coordenadas enteras (de pantalla) hacia donde se dirige la línea punteada.
- IY Coordenada vertical entera (de pantalla) hacia donde se dirige la línea punteada.
- X Coordenada horizontal en valores reales para la rutina DSHREL y de usuario para las otras dos rutinas.
- Y Coordenada vertical en valores enteros positivos o negativos dependiendo del desplazamiento para la rutina DSHREL o en unidades de usuario para las otras dos rutinas.
- L Este argumento controla el tipo de punteado que se desea y puede tomar valores desde -1 hasta el número máximo que acepte la computadora.
- 1 Punteo de 5 en 5 unidades
- 2 Punteo de 10 en 10 unidades

3. Punteo de 15 en 15 unidades
4. Punteo de 20 en 20 unidades
5. Punteo de 25 en 25 unidades
6. Punteo de 30 en 30 unidades
7. Punteo de 35 en 35 unidades
8. Punteo de 40 en 40 unidades

3.9CURSOR

Algunas terminales TEKTRONIX tienen un cursor en la pantalla, en el caso de la terminal 4006 el cursor no existe físicamente, por lo cual se desarrolló un par de subrutinas que simulan la existencia del cursor.

El cursor simulado consiste en una pequeña cruz, que aparece en el ángulo inferior izquierdo de la pantalla. A partir de esta posición habrá que "mover" el cursor hasta el punto deseado. Esto se lleva a cabo mediante comandos tecleados en la terminal con el siguiente formato:

IK, IDES

en que:

IK	ACCION
1	Subir el cursor
2	Bajar el cursor
3	mover a la izquierda
4	Mover a la derecha
5	Aceptar como definitiva la posición del cursor

IDES es el número de unidades de pantalla que se desea desplazar el cursor.

Cuando se de IK = 5 habra que dar IDES = 0.

Una vez que ha sido aceptada como definitiva la posición del cursor, cualquier caracter que se oprima será recibido por el programa como respuesta de la subrutina de simulación de cursor.

3.9.1 SCURSR

Ejemplo de uso:

CALL SCURSR (ICHAR, IX, IY)

Parámetros:

- ICHAR - Caracter alfanumérico teclado al final del uso de la subrutina.
- IX - Coordenada horizontal última del cursor (unidades de pantalla)
- IY - Coordenada vertical última del cursor (unidades de pantalla)

3.9.2 VCURSR

Ejemplo de uso:

CALL VCURSR (ICHAR, X, Y)

Parámetros:

- ICHAR - Caracter teclado al final del uso del cursor.
- X - Coordenada horizontal final del cursor en

unidades de usuario (ventana virtual)

Y - coordenada vertical final del cursor en unidades de usuario (ventana virtual)

Nótese que aún cuando el resultado final de la llamada a esta subrutina esté en coordenadas de usuario y no de pantalla, el movimiento del cursor se hace empleando las unidades de esta última.

Si el cursor se ubica fuera de la ventana las coordenadas X,Y regresarán extrapoladas correctamente, por lo que su uso en esta forma es perfectamente aceptable.

Subrutinas de utilería.

3.10 Rutinas de salida alfanumérica.

En lugar de utilizar las expresiones Write y Read del fortran, existen rutinas en este paquete que permiten salidas alfanuméricas cuando estamos graficando. Sabemos que las rutinas de salida alfanumérica se almacenan los caracteres en el amortiguador o buffer y que al llamarse la rutina adecuada salen estos.

Salida de caracteres alfanuméricos: ANCHO Y ANSTR.

3.10.1 ANCHO:

Sin tener que poner la terminal en modo alfanumérico si queremos dar salida en pantalla a un caracter utilizamos la rutina ANCHO, siempre y cuando el caracter no sea de control.

Ejemplo de uso:

CALL ANCHO(ICHAR)

Parámetros:

ICHAR Caracter alfanumérico que saldrá en la pantalla

3.10.2 ANSTR:

Es análoga a ANCHO pero esta rutina dá salida a una cuerda de caracteres en lugar de un solo caracter como ANCHO.

Los caracteres no deben ser de control

Ejemplo de uso:

CALL ANSTR(NCHAR,IADE)

Parámetros:

NCHAR Número de caracteres que saldrán.

IADE Nombre de arreglo que contiene los caracteres alfanuméricos que saldrán en la pantalla

Con estas subrutinas después de posicionar en coordenadas de pantalla o de usuario se pueden escribir los caracteres alfanuméricos en esta posición.

Manejo de caracteres de control

3.10.3 NEWLIN:

Esta rutina salta una línea y mueve el cursor al margen izquierdo

Ejemplo de uso:

CALL NEWLIN

3.10.4 LINEF:

Esta rutina salta una línea

Ejemplo de uso:

CALL LINEF

3.10.5 CARTN:

Mueve el cursor al margen izquierdo.

Ejemplo de uso:

CALL CARTN

3.10.6 HOME:

Mueve el cursor alfanumérico a la esquina superior izquierda de la pantalla

Ejemplo de uso:

CALL HOME

3.10.7 BAKSP:

Genera el regreso de un caracter en la línea (un backspace)

Ejemplo de uso

CALL BAKSP

3.10.8. NEWPAG

Borra lo escrito en la pantalla terminal y retorna el cursor a la esquina superior izquierda de la pantalla.

Ejemplo de uso

CALL NEWPAG

3.10.9 Rotación de Gráficas: RROTAT

La rutina RROTAT puede rotar una gráfica en cualquier ángulo con respecto a su posición horizontal, lo que hace esta subrutina es rotar todo el plano en el que se encuentra la

figura.

Ejemplo de uso:

CALL RROTAT (DEG)

Parámetros:

DEG ángulo de rotación relativo a su posición original

3.10.10 RSCALE

A una figura gráfica dibujada en coordenadas relativas se le puede dar una escala diferente mediante esta subrutina.

Ejemplo de uso:

CALL RSCALE(FACTOR)

Parámetro

FACTOR Factor de la escala

3.10.11 RESET:

Ejecuta las mismas funciones de INITT pero no borra la pantalla ni pone el cursor en la esquina izquierda superior de la pantalla, deja el cursor donde estaba.

Ejemplo de uso:

CALL RESET

3.10.12 HDCOPY:

Si el usuario tiene unidad de hard-copy al utilizar esta rutina obtiene una copia permanente de lo contenido en la pantalla.

Ejemplo de uso:

CALL HDCOPY

3.10.13 ERASE:

Borra el contenido de la pantalla sin cambio en la posición del cursor.

Ejemplo de uso

CALL ERASE

3.10.14 BELL:

Como alarma o indicador de un paso podemos poner un tono audible de salida en la terminal mediante esta rutina.

Ejemplo de uso:

CALL BELL

3.10.15 SEETW:

Me da los valores de pantalla que estoy utilizando

Ejemplo de uso:

CALL SEETW:

Parámetros:

MINX Coordenada de pantalla mínima horizontal

MAXX Coordenada de pantalla máxima horizontal.

MINY Coordenada de pantalla mínima vertical

YMAX Coordenada de pantalla máxima vertical.

3.10.16 SEEDW

Me da los valores de la ventana virtual que estoy utilizando.

Uso:

CALL SEEDW (XMIN, XMAX, YMIN, YMAX)

Parámetros

XMIN Mínimo horizontal en coordenadas de usuario.
 XMAX Máximo horizontal en coordenadas del usuario.
 YMIN Mínimo vertical en coordenadas del usuario.
 YMAX Máximo vertical en coordenadas del usuario.

3.10.17 SEEREL:

Me da los valores de las variables utilizadas por las rutinas virtuales relativas a vectores a escala y rotados.

Uso:

CALL SEEREL (RCOS, RSIN, SCALE)

Parámetros

RCOS Coseno del ángulo de rotación
 RSIN Seno del ángulo de rotación
 SCALE Multiplicador utilizado para la escala

3.10.18 SEETRN:

Me da el valor del conjunto de variables comunes por la ventana y rutinas de transformación.

Uso:

CALL SEETRN (XFAC, YFAC, KEY)

Parámetros

XFAC Escala del factor X
 YFAC Escala del factor Y

KEY Clave de la transformación.

1 lineal

2 logarítmica

3.10.19 KIN:

Esta es una función que transforma las pulgadas en unidades de pantalla.

Uso:

variable=KIN(RI)

Parámetros:

KIN Número de unidades de (RI) pulgadas

RI Pulgadas

3.10.20 KCM:

Esta función transforma centímetros en unidades de pantalla

Uso:

Variable=KCM(RC)

Parámetros

RC Número de centímetros

KCM Número de unidades de (RC) centímetros

3.10.21 LINWDT:

Esta es una función que nos da el ancho horizontal de un cierto número (NUMCHR) de caracteres adyacentes.

Uso:

Variable=LINWDT(NUMCHR)

Parámetros:

NUMCHR Número de caracteres adyacentes.
 LINWDT Ancho en unidades de pantalla de NUMCHR
 caracteres

3.10.22 LINGHT:

Nos da el alto de un número (NUMLIN) de líneas

Uso:

Variable=LINGHT(NUMLIN)

Parámetros:

NUMLIN Número de líneas
 LINGHT Altura de NUMLIN líneas.

3.11 Tabuladores y márgenes.

Este paquete permite al usuario poner o quitar tabuladores para facilitar el formato de salida de los caracteres alfanuméricos.

Se crea primero una tabla de tabuladores cuyo tamaño máximo es de diez tipos de tabuladores donde los arreglos verticales y horizontales tienen el mismo tamaño.

Los tabuladores verticales y horizontales son guardados en arreglos de diez palabras y sus valores están en unidades de pantalla.

3.11.1 TBLSZ.

En esta rutina damos de que tamaño va a ser la tabla de tabuladores la cual no puede ser más de 10. El primer cero nos indica el final de la tabla de tabuladores.

Ejemplo de uso:

```
CALL TPLSZ(ITBLSZ)
```

Parámetro:

ITBLSZ Tamaño de la tabla de tabuladores.

3.11.2 SETTAB:

Esta rutina establece los tabuladores en la tabla de tabuladores, establece el tabulador en coordenadas horizontales o verticales ya que deben ser iguales y da el nombre del arreglo en el que se encuentra.

Si la tabla de tabuladores está llena se perderá el máximo tabulador y la implementación de dos tabuladores iguales no será insertado

Ejemplo de uso:

```
CALL SETTAB(ITAB,ITBTBL)
```

Parámetros

ITAB Establece el tabulador en coordenadas X o Y

ITBTBL Nombre del arreglo donde se encuentra el tabulador.

3.11.3 RSTTAB:

Si queremos remover un tabulador usamos esta rutina.

Ejemplo de uso:

CALL RSTTAB(ITAB,ITBTBL)

Parámetros:

ITAB Coordenadas X o Y del del tabulador que se va a remover.

ITBTBL Nombre del arreglo del tabulador.

3.11.4 TABHOR:

Si queremos tener movimiento en el tabulador horizontal con un tabulador vertical fijo, utilizamos la rutina TABHOR y el valor de las Y estará por el primer tabulador no cero en la tabla de tabuladores.

Ejemplo de uso

CALL TABHOR(ITBTBL)

Parámetros:

ITBTBL Nombre de la tabla horizontal del tabulador que deseamos usar.

3.11.5 TABVER:

Este es el tabulador vertical y es análogo al horizontal

Ejemplo de uso

CALL TABVER(ITBTBL)

Parámetros:

ITBTBL Nombre del tabulador vertical.

3.11.6 SETMRG:

Esta rutina establece márgenes derechos e izquierdos

que serán utilizados por la rutina CARTN, HOME y NEWPAG.

Ejemplo de uso:

```
CALL SETMRG(MLEFT, MRIGHT)
```

Parámetros:

MLEFT Margen izquierdo en coordenadas de pantalla en donde las líneas alfanuméricas pueden comenzar.

MRIGHT Margen derecho en coordenadas de pantalla en donde la línea alfanumérica termina.

3.12. Transformaciones.

Estas rutinas contenidas en el paquete permiten al usuario transformar sus datos a datos logarítmicos y lineales, definiendo cualesquiera de los dos sistemas.

El sistema de default es el lineal.

3.12.1 LINTRN:

Retorna de la ventana logarítmica a la lineal.

Ejemplo de uso:

```
CALL LINTRN:
```

3.12.2 LOGTRN:

Define el eje de las X o de las Y o ambas logarítmicamente con la ventana del usuario.

Ejemplo de uso**CALL LOGTRN(KEY)****Parámetros:****KEY:**

1 eje de las X logarítmicas y de las Y lineales

2 eje de las X lineales y de las Y logarítmicas.

3 eje de las X y de las Y logarítmicas.

Para poder usar esta rutina hay que declarar la ventana del usuario logarítmica antes que la ventana de pantalla.

IV EJEMPLOS

A continuación se ven algunos ejemplos de uso del paquete. Como se utilizan las subrutinas y los resultados.

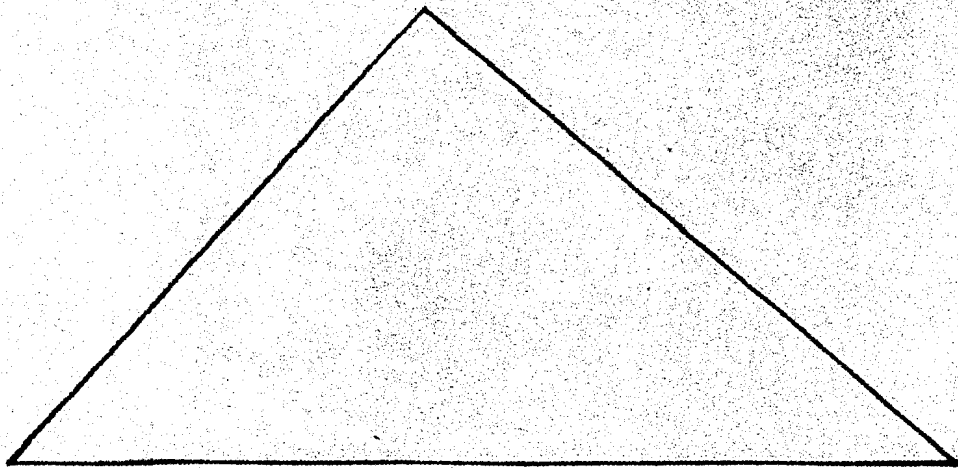
Para hacer uso del paquete se necesita que en las primeras 2 líneas del programa vayan las siguientes instrucciones.

```
$ SET AUTOBIND
```

```
$ BIND = FROM OBJECT/PL010/=
```

```
  . PROGRAMA  
  .  
  .  
  .
```

```
1000 $SET AUTOBIND
1010 $BIND = FROM OBJECT/PLD10/=
1021 C*   ESTA RUTINA DIBUJA UN TRIANGULO EN COORDENADAS
1022 C*   ABSOLUTAS
1030     DIMENSION IX(3),IY(3)
1040     DATA IX/200,1000,500/,IY/100,200,500/
1050     N=3
1060     CALL INITT(1200)
1070     CALL MOVABS(IX(N),IY(N))
1080     DO 10 I=1,N
1090     CALL DRWABS(IX(I),IY(I))
1100 10   CONTINUE
1106     CALL HOCOPY
1110     CALL FINITT(0.780)
1120     STOP
1130     END
#
```



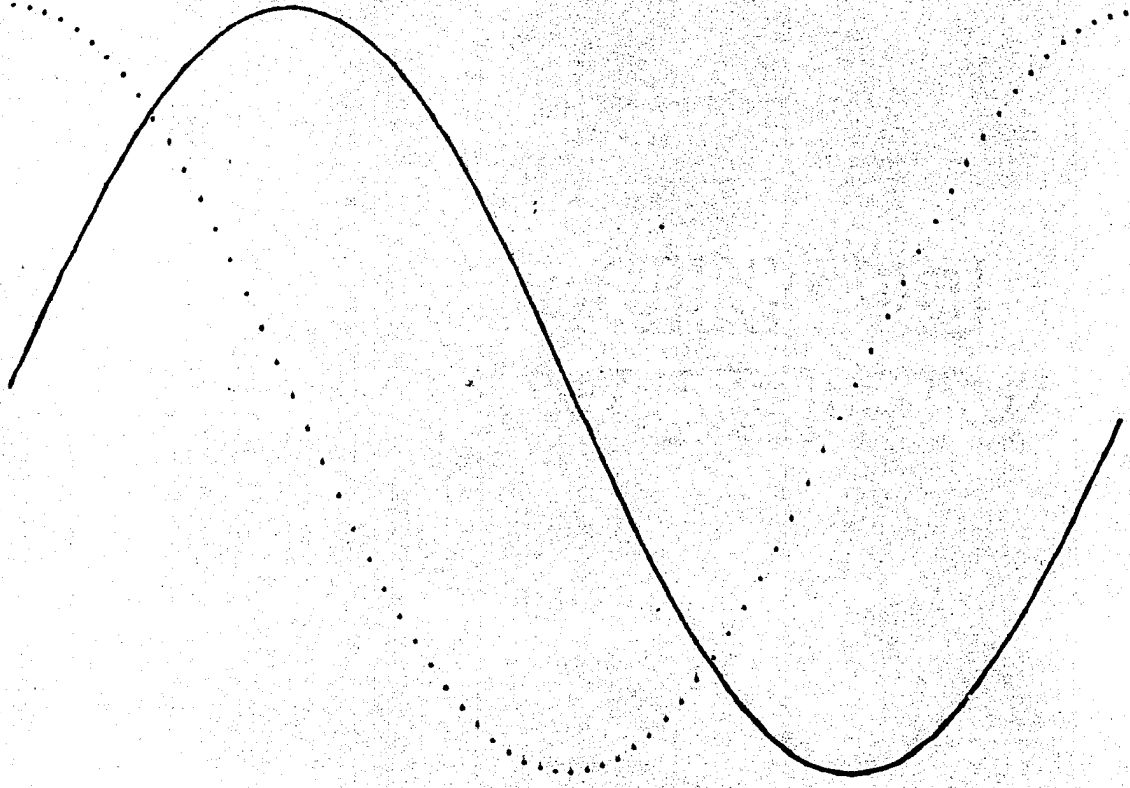
```

31 C*   GRAFICA LAS FUNCIONES SENO Y COSENO
40     CALL INITI(1200)
50     X=(0*(3.14161824)))/90
60     Y=(0/57.29578)
81 C*   DAMOS LAS VENTANAS
70     CALL OMINDO(X,Y,-1,2,1,2)
80     CALL TWINDO(11,1000,10,770)
81 C*   GRAFICA EL SENO CONTINUO
90     X=(0*(3.14161824/2)))/90
100    Y=(0/57.29578)
101    Y=SEN(Y)
110    CALL MOVEA(X,Y)
120    DO 10 I=1,360,5
130    X=(I*(3.14161824/2)))/90
140    Y=(I/57.29578)
150    Y=SEN(Y)
160    CALL DRAWA(X,Y)
170 10  CONTINUE
171 C*   GRAFICA EL COSENO PUNTEADO
180    X=(0*(3.14161824/2)))/90
190    Y=(0/57.29578)
191    Y=COS(Y)
200    CALL MOVEA(X,Y)

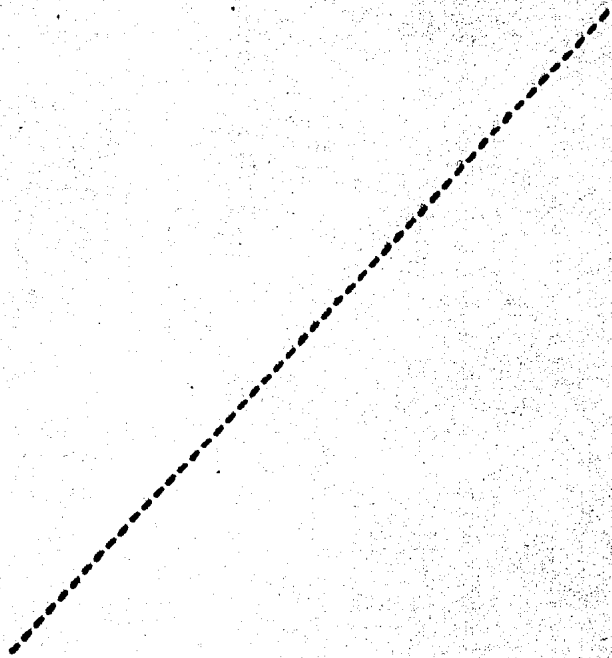
210    DO 20 I=1,360,5
220    X=(I*(3.14161824/2)))/90
230    Y=(I/57.29578)
240    Y=COS(Y)
250    CALL POINTA(X,Y)
260 20  CONTINUE
270    CALL FINITT(0,780)
280    STOP
290    END
#

```

50.



```
10 #SET AUTOBINO
20 #BINO = FROM OBJECT/PL010/=
30 C*   ESTA RUTINA DIBUJA UN PUNTEADO EN COORDENADAS ABSOLUTAS
75 3   CONTINUE
80     CALL INITT(1200)
90     CALL MOVABS(20,30)
100    CALL DSHABS(500,470,2)
103    CALL HRCOPY
110    CALL FINITT(0,780)
120    STOP
130    END
#
```

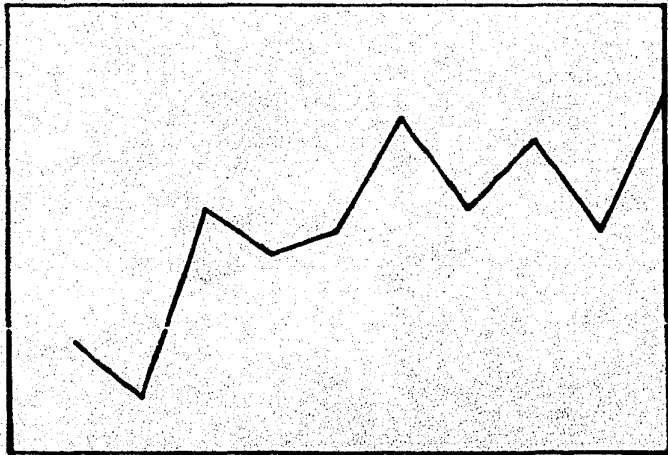
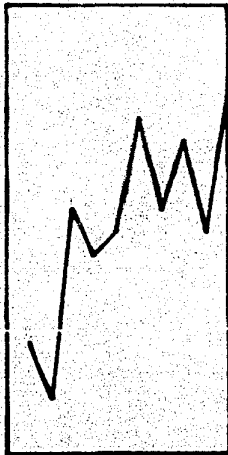


```
1 #SET AUTOBIND
2 #BIND = FROM OBJECT/FL010/*
3 C*   ESTA RUTINA NOS MUESTRA EL GRAFICADO EN LAS VENTANAS
4 C*   Y COMO CAMBIA EL TAMAÑO DE LA GRAFICAS DE ACUERDO
5 C*   AL ESPACIO DE VENTANA DE PANTALLA QUE LE DAMOS CON
6 C*   LA RUTINA TWINDO EN EL CUAL COLOCAMOS LA VENTANA
7 C*   VIRTUAL CON LA RUTINA DWINDO.
8
9 CALL INITT(2300)
10 CALL DWINDO(0,10,20,60)
11 CALL TWINDO(0,200,550,700)
12 CALL GRAFIT
13 CALL TWINDO(330,500,550,700)
14 CALL GRAFIT
15 CALL TWINDO(0,200,0,450)
16 CALL GRAFIT
17 CALL TWINDO(300,900,0,450)
18 CALL GRAFIT
19 CALL FINITT(0,700)
20 END
```



```
0 #SET SEPARATE
2 C*   ESTA SUBROUTINA NOS DA LA GRAFICA Y EL MARCO DE LA
3 C*   GRAFICA EN LA VENTANA VIRTUAL COLOCADA EN LA
4 C*   VENTANA DE PANTALLA ASIGNADA
10     SUBROUTINE GRAFIT
20     DIMENSION X(10),Y(10)
30     DATA X/1,2,3,4,5,6,7,8,9,10/
40     DATA Y/30,25,42,38,40,50,42,48,40,52/
50     CALL MOVEA(X(1),Y(1))
60     DO 100 I=1,10
70 100 CALL DRAWA(X(I),Y(I))
80     CALL MOVEA(0,20)
90     CALL DRAWA(10,20)
100    CALL DRAWA(10,60)
110    CALL DRAWA(0,60)
120    CALL DRAWA(0,20)
130    RETURN
140    END
#
```

55.



```

1 #SET AUTOBIND
2 #BIND = FROM OBJECT PLO10 =
3 C#   ESTA RUTINA DIBUJA LA GRAFICA DE UNA SERIE DE PUNTOS
4 C#   A ESTA RUTINA SE LE HA DADO UNA VENTANA VIRTUAL EN LA
5 C#   QUE NO CABEN TODOS LOS PUNTOS Y VEMOS QUE SE PUEDE GRAFICAR
6 C#   AQUELLA PARTE QUE SE ENCUENTRA EN LA VENTANA VIRTUAL
7 C#   EL ESPACIO DE VENTANA DE PANTALLA ES DADO POR SWINDO
10    DIMENSION X(10),Y(10)
20    DATA X/1,2,3,4,5,6,7,8,9,10/
30    DATA Y/30,25,42,38,40,50,54,48,40,52/
40    CALL INITT(2300)
50    CALL SWINDO(400,300,200,400)
60    CALL UMINDO(3,5,40,15)
70    CALL MOVEA(X(1),Y(1))
80    DO 100 I=1,10
90 100 CALL DRAWA(X(I),Y(I))
100   CALL MOVABS(400,200)
110   CALL DRWABS(700,200)
120   CALL DRWABS(700,600)
130   CALL DRWABS(400,600)
140   CALL DRWABS(400,200)
145   CALL HOCOPY
150   CALL FINITT(0,780)

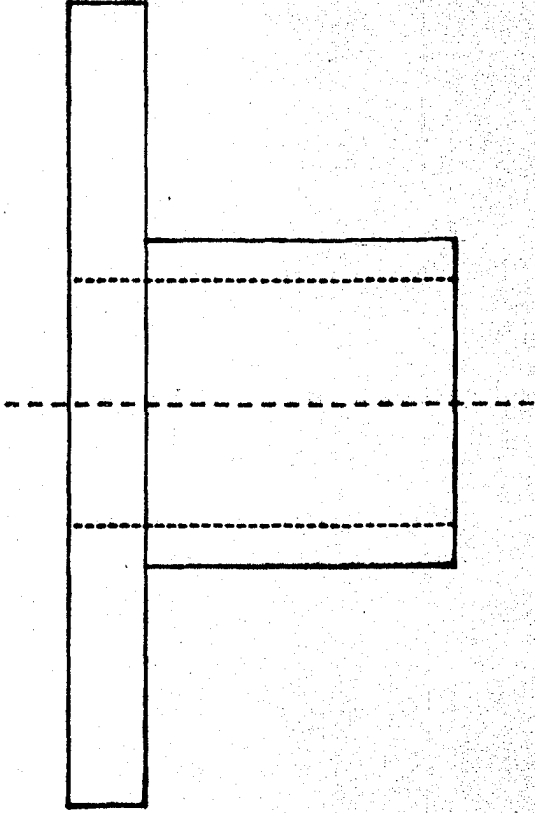
160   END
#

```

57.



```
91 C#      ESTE PROGRAMA NOS DIBUJA LINEAS RECTAS Y PUNTEADAS
92 C#      GUIADAS POR LA Rutina KIN LA CUAL CONVIERTE
93 C#      PULGADAS A UNIDADES DE PANTALLA
100 CALL  INITT(1200)
110 CALL  MOVABS(100,50)
120 CALL  DRWREL(0,KIN(5.0))
130 CALL  DRWREL(KIN(0.5),0)
140 CALL  DRWREL(0,-KIN(5.0))
150 CALL  DRWREL(-KIN(0.5),0)
160 CALL  MOVREL(KIN(0.5),KIN(1.5))
170 CALL  DRWREL(KIN(2.0),0)
180 CALL  DRWREL(0,KIN(2.0))
190 CALL  DRWREL(-KIN(2.0),0)
200 CALL  MOVREL(KIN(2.0),-KIN(0.25))
210 CALL  DSHREL(-KIN(2.5),0,1)
220 CALL  MOVREL(KIN(2.5),-KIN(1.5))
230 CALL  DSHREL(-KIN(2.5),0,1)
240 CALL  MOVREL(KIN(3.0),KIN(0.75))
250 CALL  DSHREL(-KIN(3.5),0,2)
260 CALL  FINIT(0,0)
270 END
#
```

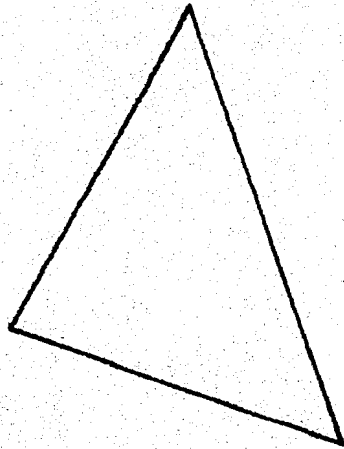
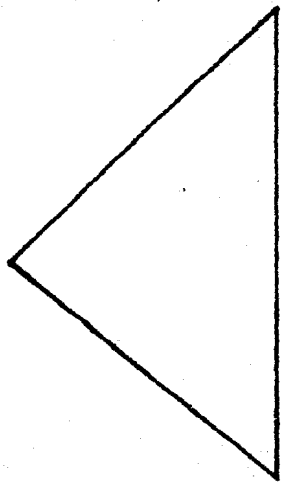


```

140      D=1.0/200.400.700.10.200.400.100.
150      N=3
160      CALL INITT(1205)
170      C+      ESTA RUTINA DIBUJA UN TRIANGULO Y SU ROTACION
180      C+      DE 30 GRADOS
190      DO 15 J=1,2
200      CALL MOVER(X(N),Y(N))
210      DO 10 I=1,N
220      CALL DRAW(X(I),Y(I))
230      10      CONTINUE
231      C*      RUTINA DE ROTACION
240      CALL ROTAT(30)
250      15      CONTINUE
261      CALL ROTAT(0)
262      C*      DIBUJA EL ORIGEN COMO UN PUNTO
270      DO 20 JI=1,100
280      20      CALL POINTA(0,0)
290      CALL HOCOPY
300      CALL FINITT(0,780)
310      STOP
320      END

```

61.



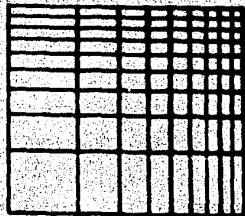
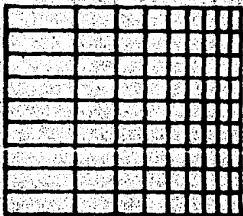
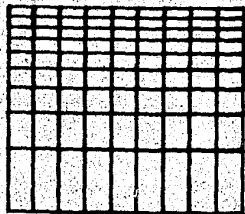
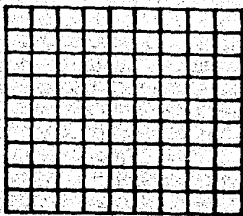

```

31 C*      ESTA RUTINA DIBUJA EN COORDENADAS LINEALES
32 C*      Y LOGARITMICAS
40        CALL INITT(1200)
41 C*      AMBOS EJES LOS DIBUJA LINEALES Y PARA ESTO
42 C*      SE DEFINE LA VENTANA VIRTUAL PARA ESTAS
43 C*      COORDENADAS
50        CALL DWINDO(10,100,10,100)
60        CALL TWINDO(0,250,500,750)
70        CALL GROD
71 C*      EJE X LOGARITMICO Y EJE Y LINEAL
72 C*      SE DEFINE VENTANA VIRTUAL
73        CALL DWINDO(ALOG(10),ALOG(100),10,100)
80        CALL TWINDO(0,250,0,250)
90        CALL LOGTRN(1)
100       CALL GROD
101 C*     EJE X LINEAL Y EJE Y LOGARITMICO
102 C*     SE DEFINE VENTANA VIRTUAL
103       CALL DWINDO(10,100,ALOG(10),ALOG(100))
110       CALL TWINDO(750,1000,500,750)
120       CALL LOGTRN(2)
130       CALL GROD
131 C*     EJE X LOGARITMICO Y EJE Y LOGARITMICO
132 C*     SE DEFINE VENTANA VIRTUAL

133       CALL DWINDO(ALOG(10),ALOG(100),ALOG(10),ALOG(100))
140       CALL TWINDO(750,1000,0,250)
150       CALL LOGTRN(3)
160       CALL GROD
210       CALL FINITT(0,0)
220       END

```

63.



64.

```

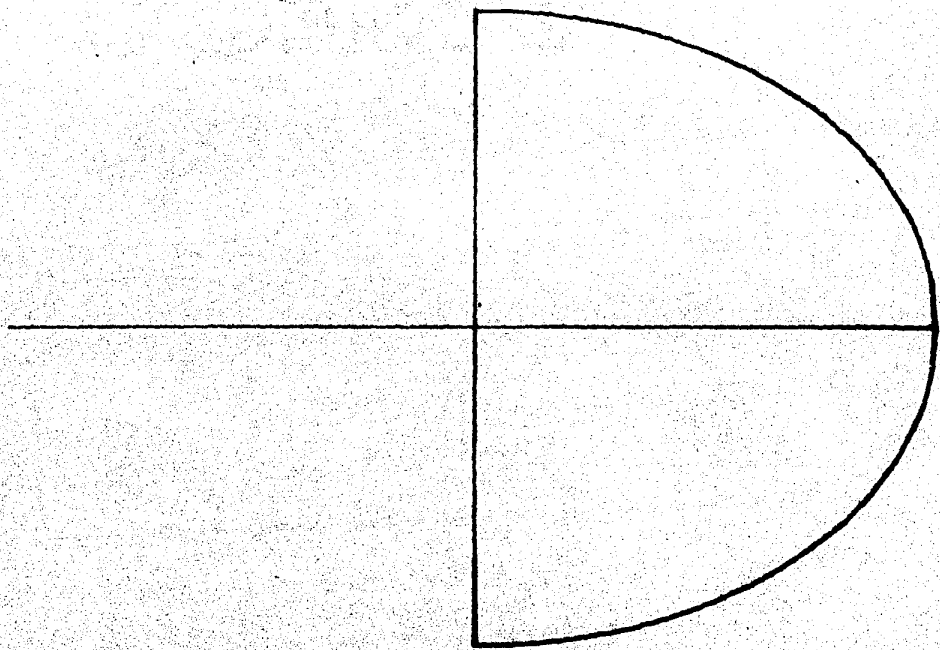
10 #SET AUTOSIND
20 #BIND = FROM OBJECT/PLO10 =
30 C*     ESTE PROGRAMA DIBUJA UN SEMICIRCULO DE RADIO UNO
31 C*     COMO SE VE TAMBIEN SE USA LA SUBROUTINA DE ESCRITURA
40     DIMENSION XX(11),YY(11)
50     DATA XX/"S","E","M","I","C","I","R","C","U","L","O"/
60     DATA YY/" ","O","E"," "," ","R","A","D","I","O"," ","1"/
70     CALL INITT(1200)
80     CALL DWINDO(-1.3,1.3,-1.3,1.3)
90     CALL TWINDO(0,1023,0,760)
100     X=1.
110     CALL MOVEA(1,0)
120     DO 10 I=1,100
130     Y=SQRT((1-X*X))
140     CALL DRAWA(X,Y)
150 10    X=X-.01
160     DO 20 J=1,100
170     Y=SQRT((1-X*X))
180     Y=-Y
190     CALL DRAWA(X,Y)
200 20    X=X+.01
210     CALL MOVEA(0,0)
220     CALL MOVER(-1,0)

230     CALL DRAWR(2,0)
240     CALL MOVER(-1,-1)
250     CALL DRAWR(0,2)
260     CALL MOVABS(2,750)
270     CALL ANSTR(11,XX)
271     CALL MOUREL(4,0)
272     CALL ANSTR(11,YY)
340     CALL FINITT(0,0)
350     END

```

65.

SEMICIRCULO DE RADIO 1



```
10 #SET AUTOBIND
20 #BIND = FROM OBJECT PLO/=
30 C# ESTE ES UN EJEMPLO DE LAS RUTINAS DE TABULADORES
40 C# EL ARPEGLIO DEL TABULADOR HORIZONTAL EL IHOR Y EL
50 C# DEL VERTICAL SERA IVERT.
60 C# IHOR TENDRA LOS VALORES DE 20,50,250,420
70 C# IVERT TENDRA LOS VALORES DE 700,500,400,300
80 C# SE TENDRA EL TABULADOR VERTICAL FIJO Y EL HORIZONTAL
90 C# SE MOVERA Y MARCARA UNA X EN TODOS SUS VALORES
100 C# Y SE VOLVERA A MOVER EL VERTICAL PARA QUE EL HORIZONTAL
110 C# VUELVA A REPETIR LO ANTERIOR.
121 C# TAMBIEN SE VERA QUE CON LA SUBROUTINA RSTTAB Y SETTAB SE CAM-
122 C# BIARA EL VALOR 250 DEL TABULADOR HORIZONTAL POR EL DE 150
130 DIMENSION IHOR(4), IVERT(4)
140 CALL INITT(1200)
150 CALL TTBLSS(4)
160 CALL SETTAB(20, IHOR)
170 CALL SETTAB(50, IHOR)
180 CALL SETTAB(250, IHOR)
190 CALL SETTAB(420, IHOR)
200 CALL SETTAB(700, IVERT)
210 CALL SETTAB(500, IVERT)
220 CALL SETTAB(400, IVERT)
```

66, A

```
230 CALL SETTAB(300,IVERT)
240 CALL MOVABS(0,0)
250 DO 100 ITAB=1,4
251 C* SI ITAB ES 3 SE CAMBIARA 250 POR 150 EN EL TABULADOR
252 C*
260 HORIZONTAL
270 IF(ITAB.EQ.3) CALL RSTTAB(250,IHOR)
280 IF(ITAB.EQ.3) CALL SETTAB(150,IHOR)
290 CALL TABUER(IVERT)
300 DO 50 IHTAB=1,4
310 CALL TABHOR(IHOR)
320 X="X"
330 C* ESCRIBE UNA X
340 CALL ANCHO(X)
350 50 CONTINUE
360 100 CALL LINEF
370 CONTINUE
380 CALL FINITT(0,0)
390 END
```

67.

x x

x

x

x x

x

x

x x

x

x

x x

x

x

62.

```

10 #SET AUTOBIND
20 #BIND = FROM OBJECT.FLO10#
30 C#   EN ESTA RUTINA VEMOS COMO TRABAJA EL CURSOR
40 C#   OOY UNA FUNCION Y VOY A TRATAR DE MARCAR CON EL CURSOR
50 C#   EL ULTIMO PUNTO DE LA GRAFICA
60 C#   TODAS LAS MARCAS DEL CURSOR QUE HAY ME VAN MARCANDO
70 C#   POR DONDE VOY HASTA QUE FINALMENTE MARCO EL ULTIMO
80 C#   PUNTO DE LA FUNCION
90     CALL INITTC(1200)
100    XY= 100*ALOG(100)
110    CALL OWINDO(0,0,101,0,0,0,XY)
120    CALL TWINDO(10,1020,10,770)
130    CALL MOVEAC(0,0)
140    X=1
150    DO 10 I=1,100
160      Y=X*ALOG(X)
170      CALL DRAWACK,Y)
180      X=X+1
190      CALL SCURS(A,10,10)
200      CALL FINITT(0,0)
210      STOP
220      END
#

```


Conclusiones.

El trabajo realizado fue la implementación de un paquete que sirviera como base para proporcionar un servicio adecuado de graficación en una máquina de gran escala.

En un principio pareció una labor muy sencilla, pero lo que se observó después de varios intentos fue el que la implementación consistía en conservar la estructura del paquete y de las funciones que se desarrollaban en cada una de las subrutinas e implementar y transportar la programación, así como la precisión en el proceso de los cálculos efectuados por algunas subrutinas.

Estoy consciente de que la labor que se desarrolló es un intento porque realmente la gente dedicada a la computación vea que antes de lanzarse a desarrollar algo novedoso, pueda observar a su alrededor lo desarrollado y encaminar su esfuerzo a implementar paquetes que tienen buenos antecedentes y futuro promisorio.

BIBLIOGRAFIA

1. Plot-10. Terminal Control System;
User's manual.
Tektronix, Inc. 1974.
2. 4006-1 Computers Display
Terminal users; instruction manual.
Tektronix, Inc. 1975.
3. Newman M. William
Sproull F. Robert
Principles of interactive computers graphics.
Mc.Grow-Hill 1973.
4. Wolfgang K. Giloi
Interactive Computer Graphics
Prentice-Hall, Inc. 1978.
5. Fis Ramón Carmona
Software Basico para manejo de
equipo de Graficación Tektronix.
U.S.C., UNAM 1980.