

2ej. 10-A



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

“ANALISIS Y DISEÑO DE UN NUEVO LENGUAJE INTERACTIVO PARA REALIZAR GRAFICAS POR COMPUTADORA”

TESIS

Que para obtener el Título de
INGENIERO EN COMPUTACION

P r e s e n t a

Héctor Cuauhtémoc Espejel Cortina

Director del Trabajo:

Ing. Carlos Ruiz de Velasco G.



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 está 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

Hoja

ANTECEDENTES	1
OBJETIVO	4
1.- FUNDAMENTACION TEORICA	
1.1. Requerimientos del lenguaje.....	7
1.2. Sistemas desarrollados anteriormente.....	12
2.- CARACTERISTICAS DE LA INTERFAZ HOMBRE-MAQUINA	
2.1. Problemática y consideraciones.....	23
2.2. Metodología y componentes.....	29
3.- DESCRIPCION Y DISEÑO DEL LENGUAJE	
3.1. Atributos, características y detalles de las gráficas.....	43
3.2. Gramática en B.N.F. y Diagramas de sintaxis.....	49
4.- EJEMPLOS Y CONCLUSIONES.....	77
5.- BIBLIOGRAFIA.....	117

ANTECEDENTES

Una correcta definición del concepto "Graficación por Computadora" es: EL ARTE O LA CIENCIA DE PRODUCIR IMAGENES GRAFICAS CON LA AYUDA DE UN COMPUTADOR . Sin embargo , ninguna - definición por extensa que ésta sea, podrá dar una idea completa del campo tan amplio en el que se puede aplicar dicho concepto en las diferentes ciencias (Medicina, Biología, Matemáticas, Física, etc).

Ciertamente, las gráficas por computadora tienen que ver con la creación de dibujos, ya sea que estos sean desplegados en una terminal de computadora, grabados en una película o dibujados en papel haciendo uso del graficador respectivo, normalmente llamado "plotter".

El primer uso que se le dió a la graficación por computadora fué el de simplificar la salida de los datos de los computadores de alta velocidad. La mayoría de estos tenían como equipo estándar, tubos de rayos catódicos sobre los cuales, los datos (puntos), podían ser desplegados en forma de gráfica de una manera mucho más fácil y rápida de lo que se podía hacer en cualquier otro dispositivo de salida disponible.

Actualmente, la graficación por computadora es mucho más que un simple vaciado de datos. Las gráficas son usadas como una liga esencial entre el ser humano y el computador, ya que a través de ellas se pueden transferir una alta CANTIDAD de -- información con alta CALIDAD. Los números, no importa que -- tan bien presentados estén, son a menudo difíciles de entender. Los dibujos o diagramas, por el contrario, pueden ex---presar la información que se desea transmitir de una manera más clara y concisa.

La revolución de los microprocesadores, ha cambiado drásticamente la graficación mediante la ayuda de un computador.- Mientras que hace algunos años los sistemas de graficación -- eran restringidos a dependencias de Gobierno y a algunas U--niversidades debido al costo tan alto, hoy en día estos sistemas pueden adquirirse con mucha mayor facilidad y a un --- costo menor.

Sin embargo, todavía existe mucho por realizar dentro de este campo ya que además de la tecnología del micropocesor,- el proceso distribuido, los lenguajes de alto nivel, el análisis de algoritmos, sistemas operativos, y muchas otras áreas de la ciencia de la computación están de alguna manera propiciando que se busquen nuevas alternativas para mejorar la actividad de graficar por medio de un computador.

Todo lo anterior, trae como consecuencia la necesidad de ---
diseñar sistemas de "software" que permitan manejar la des--
cripción geométrica y matemática de objetos, con el fin de -
obtener de una manera clara y concisa la información que se-
desea analizar.

OBJETIVO

El objetivo del trabajo es diseñar un lenguaje con sintaxis en Español que permita dibujar gráficas haciendo uso de un computador, el cual tenga conectado un dispositivo de graficación.

Este lenguaje, al cual se le dió el nombre de 'INTERGRAF', le permite al usuario dibujar sus gráficas a través de comandos sencillos y de una manera altamente interactiva.

El alcance del trabajo contempla la definición del lenguaje hasta la fase de diseño y está estructurado de la siguiente manera:

En el capítulo 1 se muestra la fundamentación teórica del diseño, las premisas requeridas para que pueda funcionar un lenguaje de esta naturaleza, además de exponer algunos sistemas de software para realizar gráficas desarrollados anteriormente, con sus respectivas características e inconvenientes.

En el capítulo 2 se describen primeramente los problemas más comunes que existen en el diálogo hombre-máquina durante el desarrollo de sistemas automatizados interactivos. En segundo lugar se hace mención de una metodología específica para

el diseño de la interfaz hombre-máquina y la forma en como se pensó desarrollarla considerando algunos aspectos de la metodología propuesta por Mark Green y por Newman y Sproull.

Posteriormente en el capítulo 3, se definen los atributos, características y detalles de las gráficas, se indican algunos conceptos relacionados con gramáticas y se muestra la gramática del lenguaje 'INTERGRAF' expresada en la forma BNF (Backus Naur Form) con los diagramas de sintaxis respectivos.

En el capítulo 4 se presentan algunos ejemplos de los comandos válidos del lenguaje con su explicación respectiva, con el objeto de que se observe claramente el potencial y la facilidad en dibujar gráficas que representaría hacer uso de este lenguaje, además de presentar las conclusiones de este trabajo.

Finalmente en el capítulo 5 se presenta la bibliografía.

1.- FUNDAMENTACION TEORICA

1.1. Requerimientos del lenguaje.

El lenguaje 'INTERGRAF' fué diseñado pensando en que muchas veces los usuarios de un computador tienen problemas con la definición de sus requerimientos de graficación y en la manera en como deben alimentar sus datos e instrucciones a la máquina. Lo anterior se debe en gran medida a la complejidad de los comandos que deben aprenderse para realizar una función de graficación específica ; sin embargo, en este momento no se tocará ese punto ya que en el capítulo dos se profundiza en esta problemática referente a la interfaz hombre-máquina.

Debido a que el alcance de este trabajo abarca sólo hasta el diseño del lenguaje, es decir, la definición, estructura, -- presentación de la gramática y sintáxis del lenguaje, es --- necesario establecer ciertos requerimientos que deben existir cuando se quiera utilizar 'INTERGRAF' en cualquier computador que tenga conectado un graficador ("plotter").

Primeramente es necesario recordar la secuencia de ejecución de una instrucción por un computador desde que se emite un comando o una serie de ellos por un usuario, hasta la ejecución de las mismas.

Como es sabido, la computadora únicamente reconoce lo que es llamado universalmente lenguaje de máquina, es decir, maneja toda la información a través del código binario(1's y 0's).

Debido a lo anterior es necesario pasar por una serie de fases previas a la ejecución de un programa. En forma esquemática lo podemos apreciar en la figura 1.

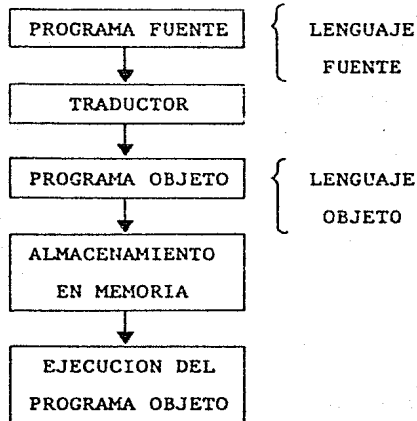


FIG. 1

Con base en esto se puede definir el concepto de 'TRADUCTOR' como un programa que toma como entrada un conjunto de instrucciones escritas en un lenguaje (LENGUAJE FUENTE) y produce como salida un conjunto de instrucciones en otro lenguaje (LENGUAJE OBJETO)

Si el Lenguaje Fuente es un lenguaje de alto nivel, por ejemplo FORTRAN o COBOL y el Lenguaje Objeto es un lenguaje de máquina, al TRADUCTOR se le llama COMPILADOR.

Para ejecutar un programa escrito en un lenguaje de alto nivel es necesario considerar dos pasos:

El programa fuente debe ser primeramente compilado, esto es, traducido al programa objeto y entonces éste es cargado a memoria y ejecutado.

Tomando como base lo explicado anteriormente, el requisito fundamental para que el lenguaje de graficación 'INTERGRAF' pueda operar en un computador, consiste en que exista el SOFTWARE (programas) necesario que sirva como interfaz entre 'INTERGRAF' y el lenguaje propio del graficador particular, es decir que exista implementado un TRADUCTOR.

Este traductor no debe ser necesariamente un COMPILADOR completo. Basta el haber desarrollado el ANALIZADOR LEXICO Y EL ANALIZADOR SINTACTICO para que 'INTERGRAF' pueda funcionar.

A continuación se explicará brevemente las funciones que deben realizar:

ANALIZADOR LEXICO (A.L.): EL analizador léxico deberá ser capaz de leer el programa fuente, que para este caso consis-

tirá de los comandos del lenguaje 'INTERGRAF', caracter por caracter, dividiendo el programa fuente o conjunto de comandos en una secuencia de unidades atómicas llamadas 'tokens'. Cada 'token' representará una secuencia de caracteres que -- pueden tratarse como una entidad lógica simple. Los identificadores, palabras reservadas, constantes, operadores y símbolos de puntuación (puntos, comas, paréntesis, etc), son 'tokens' típicos.

El A.L. tomará como base la gramática del lenguaje 'INTER---GRAF' para determinar si los caracteres que recibe pertenecen al lenguaje o no.

ANALIZADOR SINTACTICO [A.S.] : Como su nombre lo dice, su -- función consiste en analizar la sintaxis del lenguaje. El A. S. toma como entrada los códigos de los 'tokens' que encontró anteriormente el A.L.

Con base en la información que proporciona el A.L. al A.S. -- se determina si un comando o instrucción es correcta o no.

En caso de ser correcta, el A.S. deberá ser capaz de traducir la función indicada a través de un comando de 'INTER ---GRAF', a su correspondiente función del lenguaje que maneja el graficador particular. En caso de ser incorrecta la sintaxis de una instrucción deberá enviar al usuario un código que indique la existencia de un error de sintaxis en la instrucción correspondiente.

Básicamente el contar con un A.L. y un A.S. para el lenguaje

'INTERGRAF' son los requisitos o requerimientos del mismo. En la figura 2 se presenta en forma esquemática la secuencia de ejecución que debe seguir un comando emitido en 'INTER---GRAF' para que se ejecute en un computador específico donde se quiera dibujar las gráficas.

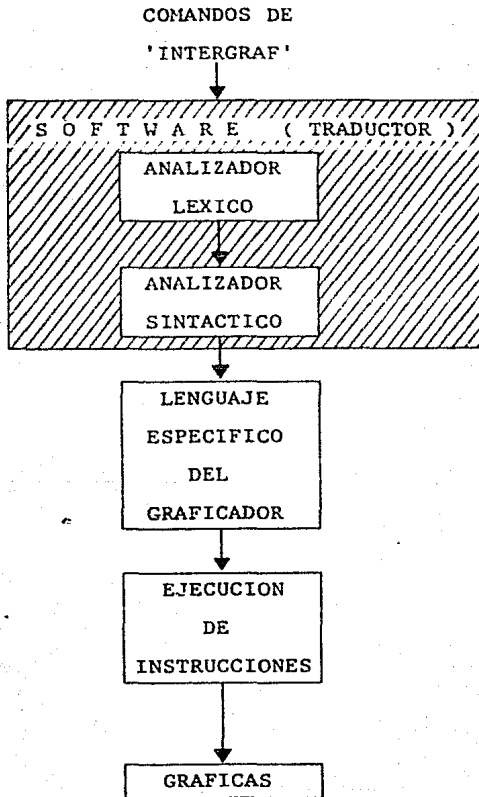


FIG. 2

1.2. Sistemas desarrollados anteriormente.

Existen varias alternativas para diseñar sistemas que permitan dibujar gráficas utilizando la ayuda de una computadora; básicamente se pueden considerar las siguientes:

A) Por medio de programas especializados.

Para explicar esta manera de dibujar gráficas se tomará como ejemplo el sistema "GRAFO" desarrollado para utilizarse por el graficador Hewlett Packard modelo HP7203A, el cual escala y grafica automáticamente cualquier serie de datos. Sus principales características e inconvenientes son los siguientes:

Características:

- I. El sistema puede procesar hasta 100 gráficas proporcionadas por el usuario.
- II. Las gráficas por dibujar pueden ser de 2 tipos:
 - i) Se dan los valores de una función $f(x)$ cuando se ha discretizado "x" en forma equidistante.
 - ii) Se dan parejas de valores $(x, f(x))$ en donde "x" puede tomar valores arbitrarios.

III. Cada una de las gráficas puede ser procesada de acuerdo con 5 distintos tipos de unión entre los valores numéricos proporcionados por el usuario; por ejemplo, el programa pregunta:

GRAFICA # 2, QUE TIPO DE GRAFICA (1,2,3,4,5) ?

a lo cual el usuario debe contestar alguno de los valores válidos.

Los tipos son los siguientes:

<u>TIPO</u>	<u>S I G N I F I C A D O</u>
1	Gráfica continua.
2	Gráfica de barras.
3	Histograma (empezando horizontalmente).
4	Histograma (empezando verticalmente).
5	Puntos.

IV. Del número total de gráficas que el usuario requiera dibujar, se pueden agrupar en conjuntos de gráficas sucesivas que serán dibujadas utilizando la misma escala para todas y cada una de ellas.

Por ejemplo el programa "GRAFO" pregunta:

GRUPO # 3

GRAFICA INICIAL, GRAFICA FINAL DE GRUPO. ?

a lo que el usuario contestaría:

1,20

Sin embargo, en caso de que el usuario no deseara observar las magnitudes relativas en un grupo de gráficas, -- sería necesario declarar a cada una de ellas como un grupo conteniendo un elemento cada uno.

- V. El tipo de gráfica ó gráficas por dibujar, puede ser seleccionado independientemente del grupo ó grupos a los que pertenezcan.

VI. Entrada de datos.

Previamente a la ejecución de "GRAFO", el usuario debe haber creado un archivo en disco bajo un nombre cualquiera --- (compatible con el sistema operativo de la BURROUGH'S B6700, B6800 o B7800) que él mismo elija.

El formato del archivo dependerá de una de las 2 formas en que el usuario quiera ver graficados sus datos:

- i) Gráficas equidistantes respecto a las abscisas o
- ii) Gráficas NO equidistantes respecto a las abscisas.

Al momento en que se ejecuta "GRAFO", una de las 2 formas -- anteriores debió haber sido seleccionada por el usuario, ya que el sistema no puede procesar una mezcla de ambas contenidas en el mismo archivo.

Inconvenientes:

I. Existe un bajo indice de interactividad en el software, - ya que cuando se requiere indicar un atributo de una gráfica (por ejemplo: QUE TIPO DE GRAFICA (1,2,3,4,5)?), el sistema pregunta cada vez, el tipo de la gráfica. Esto no tendría mayor importancia si únicamente se pudieran dibujar un número pequeño de gráficas, pero como se permite - un máximo de 100, resulta bastante incómodo estar contestando al sistema cuando se requiere dibujar un número --- considerable de ellas.

El mismo problema resulta al momento de definir las coordenadas máximas y mínimas de cada gráfica, así como cuando se requiere definir grupos.

II. El sistema no distingue entre usuarios que ya han tenido contacto con "GRAFO" anteriormente y los que lo van a usar por primera vez, lo cual repercute en desesperación por parte de los usuarios ante el sistema.

III. Otro inconveniente consiste en que el usuario tiene que contar anticipadamente el número de puntos que tendrá - una gráfica, lo cual también repercute en la actitud -- del usuario ante el sistema.

B) Por medio de un conjunto de rutinas.

Esta es la segunda forma por medio de la cual se pueden realizar gráficas por computadora; para explicarla se tomará -- como referencia el sistema "HPGRAF" desarrollado también para usarse conjuntamente con el graficador HP7203A. Sus principales características e inconvenientes son:

Características:

I. Este sistema fue desarrollado con el objetivo de que los usuarios del graficador no se preocuparan de la forma en que deben ser alimentados los datos. "HPGRAF" está formado por - un conjunto de instrucciones básicas que pueden ser llamadas como subrutinas desde el programa del usuario escrito en un lenguaje de alto nivel como FORTRAN, ALGOL o PASCAL, proporcionándole una manera sencilla, rápida y clara para generar cualquier tipo de gráfica.

II. "HPGRAF" contiene una sola rutina nombrada GRAFIK, misma que es llamada con 5 parámetros, usándose el primero para seleccionar la "acción" que se va a realizar .

III. Estas "acciones" permiten al usuario, por ejemplo, controlar la posición de la pluma (subirla o bajarla), controlar el estado del graficador (prenderlo o apagarlo), trazar líneas continuas o de guiones, marcar puntos, etc.

IV. Cuando se hace referencia a un punto $p(x,y)$, las coordenadas pueden ser:

a) Absolutas:

Referidas al sistema del graficador

$$0 \leq x \leq 9999$$

$$0 \leq y \leq 9999$$

b) Reales:

De acuerdo con un marco de referencia especificado por el usuario con anterioridad.

c) Referidas a una ventana sobre el graficador

$$0 \leq x \leq 9999$$

$$0 \leq y \leq 9999$$

En este punto es conveniente definir dos conceptos:

VENTANA: Es el marco de referencia del mundo real que permite tener o definir por medio de cuatro coordenadas, es decir dos puntos $X1, Y1$ y $X2, Y2$ la magnitud de la imagen por dibujar.

PUERTO: Es el marco de referencia que permite definir el espacio en el graficador que será utilizado para dibujar la gráfica o gráficas que requiera el usuario.

En la figura 3 se presenta un ejemplo de ventana y de puerto:

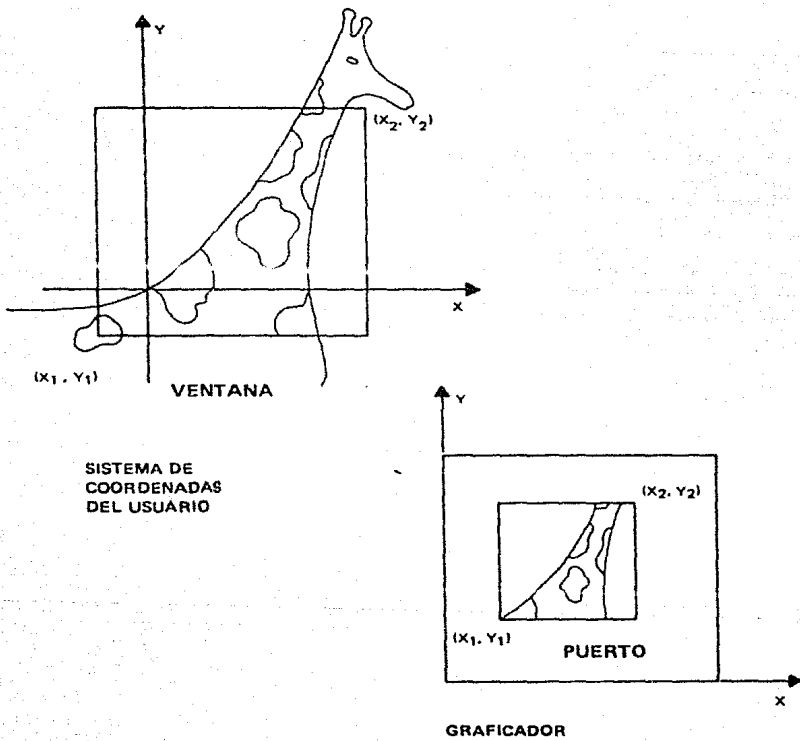


Fig. 3

V. La rutina GRAFIK también cuenta con los comandos necesarios que permiten definir la ventana del mundo real y la ventana en el graficador.

Inconvenientes:

I. Debido a que el número de parámetros en una llamada a una subrutina es fijo, siempre se deberá llamar a GRAFIK con los 5 parámetros aunque en determinado momento no todos ellos -- tengan algún significado para realizar la función que se desea efectuar. Por ejemplo, para mover la pluma al punto de coordenadas absolutas (xa,ya) la forma de llamar a GRAFIK será:

GRAFIK ("MOVA",xa,ya,z,z)

donde "z" representa cualquier valor o variable que será ignorado para realizar esta acción, pero que debe ser incluido para completar el número de parámetros.

II. Para efectos de graficación, GRAFIK hace uso de un archivo llamado FILE 50, que el usuario debió declarar con anterioridad en su programa. Este archivo deberá ser para uso -- exclusivo de GRAFIK.

III. Durante el tiempo en el que el graficador esté prendido el usuario no podrá leer ni escribir en la terminal a la --- cual esté conectado el graficador, por lo que al terminar de dibujar una gráfica, el usuario deberá 'apagarlo' con el comando "APAG".

C) Por medio de una extensión a un lenguaje de programación.

Esta manera de generar gráficas consiste básicamente en alterar la estructura de un lenguaje de programación de alto nivel como FORTRAN o ALGOL, incluyendo variables que le permitan al usuario usar dicho lenguaje para dibujar gráficas, además de utilizarlo como un instrumento normal de programación.

Estas modificaciones mencionadas a los lenguajes, necesariamente implica la alteración del compilador correspondiente, lo cual es una tarea bastante compleja.

**2.- CARACTERISTICAS DE LA INTERFAZ
HOMBRE-MAQUINA**

2.1. Problemática y consideraciones.

Ningún componente de un sistema interactivo es más impredecible en su realización que la interfaz con el usuario; es decir, la parte del sistema que determina la manera en la cual el usuario y la máquina se comunican durante una sesión normal de trabajo.

Las malas interfaces no solo son difíciles de aprender sino que hacen a los programas ineficientes en su operación. Para ilustrar lo anterior, se tomarán los ejemplos siguientes:

Ejemplo A:

LO QUE PREGUNTA LA MAQUINA.	LO QUE EL USUARIO CONTESTA.
# ALTO?	NO
# ALTO?	SI
# ALTO?	N
# ALTO?	S
# ALTO?	YES

(En este momento el usuario cae en la desesperación)

En el ejemplo anterior, el programador asume que el usuario ha leído con anterioridad el manual donde se explica que las respuestas correctas son "P" para suspender y "C" para contnuar.

Ejemplo B:

DESEA MONITOREAR TEMPERATURAS?

C

CUANTOS PUNTOS DESEA MONITOREAR?

3

POR FAVOR INDIQUE EL PUNTO NO. 1

123

POR FAVOR INDIQUE EL PUNTO NO. 2

32

POR FAVOR INDIQUE EL PUNTO NO. 3

12

OPRIMA 'C' PARA CONTINUAR O 'P' PARA SUSPENDER

C

POR FAVOR INDIQUE EL INTERVALO DE MONITOREO EN SEGUNDOS

55

OPRIMA 'C' PARA CONTINUAR O 'P' PARA SUSPENDER

V

***** ERROR 26 CARACTER ILEGAL *****

DESEA MONITOREAR TEMPERATURAS?

En este ejemplo B, las respuestas que debe dar el usuario -- son obvias, sin embargo, cuando comete un error, el programa transfiere el control a la primera pregunta, lo cual repercute nuevamente en desesperación y frustración por parte del usuario, ya que la información que se había alimentado al -- sistema deberá ser proporcionada nuevamente.

En los 2 ejemplos anteriores, el aspecto importante que se - debe observar es que ninguna de las dificultades presentadas hubieran sido evitadas por una mayor sofisticación en el --- software o hardware; éstos son problemas humanos.

En el primer caso, un manual era necesario, sin embargo, el usuario o no sabía que se requería o no lo tenía a la mano; - en el segundo caso, el programador asumió que los errores iban a ser tan pocos que no tenía que pensar en las consecuencias que pudieran surgir.

Como punto de partida es importante considerar que el diálogo hombre-máquina se comporta de una manera semejante a la - comunicación natural entre las personas por medio de un lenguaje.

Una de las principales metas a las cuales hay que tender en el desarrollo de sistemas interactivos de graficación o de cualquier otra índole, es la simbiosis que debe existir entre el hombre y la computadora. Cuando un usuario es capaz de interactuar por terminal con una computadora de tal manera que no se de cuenta del medio de comunicación o de la computadora misma, a esta interacción se le puede llamar conversacional; de esta forma, el conjunto de elementos aportados por el hombre y por la máquina llegan a convertirse en uno solo, trabajando para satisfacer un requerimiento específico.

Aunque las ideas anteriores se podrían considerar como utópicas, representan la meta a la cual se debe tender en el diseño e implantación de un sistema interactivo.

Se pueden expresar 2 principios fundamentales en el buen diseño de un sistema conversacional que permita generar y dibujar gráficas, mismos que se tratan de cumplir en el diseño de 'INTERGRAF':

a) Principios en lenguaje.

El lenguaje y contenido de la conversación debe ser inherente y natural al lenguaje del hombre. Es esencial que el lenguaje sea eficiente, completo y tenga una gramática natural; se puede decir que un lenguaje es eficiente cuando expresa -

las ideas de una manera clara y concisa. Una gramática natural es la que ofrece pocos impedimentos al usuario en la expresión de sus ideas usando símbolos fundamentales que estén disponibles, y además, debe permitir que el sistema sea útil con un mínimo de entrenamiento por parte del usuario.

La interacción en el lenguaje debe permitir al usuario concentrarse en la semántica de lo que desea expresar. En este sentido, las distracciones y discontinuidades surgidas en el pensamiento del usuario, provocadas por la sintaxis o el vocabulario, deben ser mínimas o de ser posible no existir.

b) Principios psicológicos.

El segundo principio que se pretende cumplir, consiste en evitar bloques psicológicos que a menudo previenen el comportamiento del usuario en una interacción con la computadora.

Los bloques más típicos que pueden existir en este sentido son: Aburrimiento, Pánico y Frustración.

1. Aburrimiento.

Un aspecto muy importante es tener tiempos de respuesta adecuados; éstos dependen de las características del hard

ware disponible y de los algoritmos desarrollados para responder al usuario. Si lo anterior se logra, es dar un gran paso para evitar el aburrimiento en el usuario.

2. Pánico.

Es el resultado de retrasos demasiado largos no esperados o cuando no existe respuesta del sistema en un período de tiempo muy grande. Una manera de resolver este problema es indicar de alguna forma al usuario el avance parcial del proceso solicitado.

3. Frustración.

Es causada por sistemas inflexibles; por ejemplo, este problema surge cuando el usuario proporciona una cadena de instrucciones mismas que son aceptadas inmediatamente por el sistema impidiendo con ésto, modificar un error durante la indicación del comando y por tanto, el usuario debe esperar a que se hayan ejecutado las instrucciones iniciales para cancelarlas y posteriormente proporcionar las correctas.

De lo anterior, se puede concluir que las metas por alcanzar en el diseño e implantación de un lenguaje interactivo para generar y dibujar gráficas deben ser el proveer un formato del lenguaje natural donde no existan los 3 bloques citados.

2.2.- Metodología y componentes.

Existen tres metas a las que cualquier metodología para el diseño de interfaces interactivas con el usuario debe tener.

a) La metodología debe proporcionar una notación formal para describir la interfaz con el usuario. Esta es la mayor contribución de cualquier metodología en el diseño de software. Una notación formal además de poder ser usada como una comunicación entre diseñadores, también puede usarse como una referencia por los programadores que lleven a cabo la implantación de la interfaz.

Los programadores no deben tomar decisiones en el diseño de la interfaz hombre-máquina; por tanto, una descripción completa de ella debe estar disponible antes de iniciar la implantación.

El hecho de que exista una notación formal es importante debido a lo siguiente:

. Asegura que todos los usuarios de la notación estén de acuerdo sobre la interpretación de la descripción de la interfaz y

. Permite la construcción adecuada de programas que procesen las descripciones de la interfaz.

b) La metodología debe proporcionar un mecanismo que permita determinar la veracidad del diseño antes de que se implante. En la práctica es muy difícil de lograr, si no imposible, --

mostrar y comprobar que un diseño es totalmente correcto.

c) La metodología debe proporcionar un mecanismo que permita evaluar la efectividad de la interfaz al ser usada y probada por diferentes tipos de usuarios. Esta meta trata de que la calidad de la interfaz sea la mejor posible.

El punto de partida en cualquier proceso para diseñar la interfaz hombre-máquina o usuario-máquina, consiste en estudiar el problema que el usuario está tratando de resolver; - en este sentido, el primer componente de la interfaz lo constituye el MODELO DEL USUARIO.

MODELO DEL USUARIO.- Es un modelo conceptual formado por el usuario, de la información a manejar y de los procesos que - aplicará a esta información i.e., qué problema se quiere resolver, por ejemplo el tipo de gráficas que se requiera dibujar. Este modelo le debe permitir al usuario desarrollar con poco o ningún conocimiento de computación, un buen entendimiento de lo que hace el sistema de graficación.

Con la ayuda del Modelo, el usuario puede anticipar el efecto de sus acciones y puede planear sus estrategias propias - para la operación del programa o sistema.

En realidad, cada usuario tendrá un modelo conceptual diferente de las tareas a ser realizadas y de la forma en cómo - opera la interfaz.

En el diseño del MODELO DEL USUARIO es importante considerar dos aspectos:

. Que es un modelo mental y que actúa como un marco para el desarrollo de estrategias en la operación del sistema. Es

análogo a la gramática que conforma un lenguaje extranjero - en el sentido de que se confía en él en función de que exista éxito en la comunicación.

. El modelo del usuario debe emplear conceptos familiares. El uso de estos conceptos hace al modelo más intuitivo y más fácil de aprender; sin embargo, es importante también considerar las restricciones impuestas por el 'hardware' y 'software' existente.

Para comprender mejor los componentes del modelo del usuario es importante, primeramente, definir el concepto de ANALISIS DE LA TAREA.

ANALISIS DE LA TAREA. - Es el proceso de descomponer un problema en un número finito de tareas. La mayoría de los problemas, si no es que todos, pueden descomponerse en una serie de tareas las cuales cuando se combinan en el orden correcto solucionan el problema original. Estas tareas deben ser atómicas, es decir, que ya no se puedan descomponer en otras. Desde el punto de vista del usuario existen varias posibles descomposiciones de tareas para resolver un problema dado, por tanto, de cada descomposición correcta surgirá un modelo del usuario diferente.

Con base en lo anterior se puede explicar el primer componente del MODELO DEL USUARIO que consiste en una descripción formal de las tareas atómicas de un problema dado; a esta --

descripción se le llama MODELO DE LA TAREA.

Este Modelo está en términos de objetos y operadores. Los objetos son entidades en el problema a ser solucionado, son elementos de información sobre los cuales el usuario tiene -- control; por ejemplo, los puede mover, borrar, modificar, -- puede crear otros objetos en su lugar.

A menudo existe una relación jerárquica entre objetos; en un editor de texto, el objeto más simple es un carácter, conjuntos de caracteres forman palabras, las cuales forman líneas; éstas forman párrafos que a su vez forman páginas y finalmente un conjunto de páginas forman un documento.

Los operadores son las acciones u operaciones que el usuario puede aplicar a los objetos; así, el conjunto completo de -- acciones define la capacidad funcional del sistema. Normalmente existe una correspondencia biunívoca entre operadores y tareas.

El desarrollo del MODELO DE LA TAREA tiene como propósitos:

- . Proporcionar una notación que el diseñador puede utilizar cuando realice el análisis de la tarea así como describir el problema que el usuario está tratando de resolver y

- . Establecer el punto de partida para el resto de la notación formal.

La notación usada para el MODELO DE LA TAREA tiene tres componentes: Definición de objetos, definición de operadores e invariantes o restricciones.

La definición de objetos establece las propiedades de ellos en el análisis de la tarea; son similares a las definiciones de las estructuras de datos. La definición de un objeto consiste en asignar una serie de atributos cada uno de los cuales puede ser del tipo de dato primitivo de la notación u otro objeto del modelo. En la notación, los datos primitivos pueden ser números enteros, reales, conjuntos, puntos, etc.

La definición de operadores establece las condiciones bajo las cuales se puede aplicar un operador y el efecto que causa sobre los objetos existentes en el modelo.

Cada definición de un operador consiste de una cabeza y un cuerpo. La cabeza contiene el nombre del operador y las declaraciones de sus operandos. Por otro lado, el cuerpo de la definición se divide en dos secciones:

- La primera contiene una lista de proposiciones lógicas que dan las condiciones bajo las cuales se puede aplicar un operador; estas condiciones son llamadas pre-condiciones.
- La segunda sección contiene una lista de expresiones lógicas que establecen los efectos del operador.

Las invariantes son relaciones que existen entre los operadores y los objetos en el modelo. Estas relaciones expresan el punto de vista del usuario de cómo opera el dominio del problema.

A continuación se muestra un ejemplo de ANALISIS DE LA TAREA con el objeto de reafirmar los conceptos explicados anteriormente:

Después de analizar un 'x' problema por resolver se llegó a la conclusión de que se tienen tres tareas atómicas:

1. Crear un objeto geométrico nuevo y colocarlo en el área de trabajo.
2. Mover un objeto geométrico ya existente a una nueva posición.
3. Suprimir un objeto geométrico de un área de trabajo.

El ANALISIS DE LA TAREA anterior sugiere que habrá dos tipos de objetos y tres operadores en el correspondiente MODELO DE LA TAREA.

Los tipos de objetos son objetos geométricos y el área de -- trabajo. A su vez cada objeto geométrico tiene tres atributos que son su forma, su posición en el área de trabajo y el área que cubre dentro del área de trabajo.

Los atributos del objeto área de trabajo son su tamaño y una lista de objetos geométricos que se han puesto sobre ella.

Los tres operadores en el modelo son: 'coloca' , 'mueve' y - 'suprime' , los cuales corresponden a las tres tareas identificadas en el ANALISIS DE LA TAREA.

La pre-condición para ejecutar la tarea de 'colocar' establece que el área donde se desee ubicar el objeto debe contener a éste.

La post-condición para esta tarea establece que el nuevo objeto sea incluido en el área de trabajo en la posición indicada por el usuario.

La invariante o restricción en este MODELO DE LA TAREA establece que todos los objetos geométricos en el área de trabajo deben estar completos, i.e., ninguna parte del objeto puede estar fuera de esa área.

El segundo componente del MODELO DEL USUARIO, llamado MODELO DE CONTROL, es una descripción de cómo ve el usuario la operación de la interfaz. El principal propósito de este paso es diseñar un modelo consistente de la forma en que opera la interfaz.

Este modelo será usado por el usuario para predecir el efecto de un comando o de una secuencia de ellos. El usuario no verá este modelo sino que lo inferirá de su experiencia con la interfaz. El MODELO DE CONTROL debe describir todas las acciones que el usuario puede realizar, mismas que son descritas en términos de los objetos en los MODELOS DE LA TAREA Y DE CONTROL.

La estructura del MODELO DE CONTROL es similar a la del MODELO DE LA TAREA y se usa la misma notación para determinar ambos modelos.

Diferencia entre ambos modelos:

El MODELO DE LA TAREA describe las tareas a ser realizadas, mientras que el MODELO DE CONTROL describe cómo ve el usuario los comandos que están disponibles para realizar dichas tareas. Al igual que el modelo de la tarea, El MODELO DE CON

TROL es realizado a partir de definiciones de objetos, definiciones de operadores e invariantes. Todos los objetos en el MODELO DE LA TAREA están implícitamente incluidos en el MODELO DE CONTROL.

Otras características de este último son:

- . Los objetos definidos en él son los introducidos por la interfaz; por ejemplo menús, selección de operandos, etc.
- . Los operadores definen comandos y acciones proporcionados por la interfaz.
- . Las invariantes establecen relaciones entre los objetos y los operadores en el modelo.

El MODELO DE CONTROL es una descripción de alto nivel de la interfaz con el usuario; después de haber desarrollado este modelo es necesario verificar su consistencia con el MODELO DE LA TAREA. Lo anterior se logra construyendo una secuencia de operadores del MODELO DE CONTROL que realicen las mismas funciones especificadas en el MODELO DE LA TAREA.

Si el mapeo anterior puede lograrse para todos los operadores en ambos modelos, entonces se puede decir que el MODELO DE CONTROL es consistente con el MODELO DE LA TAREA. Un parámetro que puede usarse para determinar la calidad de una interfaz puede ser precisamente este tipo de mapeo, ya que en la medida que ello exista en todos los operadores existentes en uno y otro modelo se reducirán los problemas de interacción del usuario con la máquina.

El segundo componente de la interfaz hombre máquina lo constituye el LENGUAJE DE COMANDOS.

Una vez que el usuario ha entendido los modelos explicados anteriormente, requiere de comandos con los que pueda manejarlos.

Los lenguajes de comandos los podemos encontrar tanto en sistemas interactivos como en no-interactivos. Probablemente el más usado sea el lenguaje de control de trabajos usado en -- las máquinas IBM llamado JCL.

La clave para diseñar lenguajes en sistemas gráficos, radica en reconocer que un LENGUAJE DE COMANDOS no sólo es una lista de instrucciones o comandos, sino un lenguaje verdadero, en el sentido de que los comandos que lo constituyen se relacionan entre sí de una manera sistemática y que en conjunto definen una sintaxis.

El primer problema en el diseño de un conjunto de comandos es el diseño del lenguaje mismo. Se debe definir una sintaxis y asegurarse de que ésta sea consistente; se deben determinar los errores de sintaxis que el usuario pueda cometer y además estar seguro que la sintaxis sea suficientemente clara para su comprensión y amplia y versátil para satisfacer las necesidades del usuario.

En el diseño del lenguaje no sólo debe considerarse la sintaxis sino también la semántica, i.e., el significado que se refiere a cada construcción sintáctica. Esta semántica se relaciona bastante con el MODELO DEL USUARIO, ya que los comandos operan sobre los objetos definidos en él y en muchas

ocasiones realizan las acciones del modelo. Esta relación - simplifica la tarea de diseñar el LENGUAJE DE COMANDOS y un MODELO DEL USUARIO adecuado; cada comando corresponde a una acción y los operandos de cada comando son los objetos que - afecta la acción. EL MODELO DEL USUARIO y el LENGUAJE DE COMANDOS tienden a diseñarse en forma paralela; es importante - tratar de que sean lo más sencillos posible.

Principios en el diseño de un LENGUAJE DE COMANDOS.

Los objetos y las acciones del MODELO DEL USUARIO proporcionan un buen punto de partida para el diseño del LENGUAJE DE COMANDOS al cual se le puede ver como una representación concreta del primero; en el lenguaje las acciones se reemplazan por comandos y los objetos representan los operandos del -- comando.

Una vez que se defina completamente el MODELO DEL USUARIO, - EL LENGUAJE DE COMANDOS debe poderse derivar de él usando un proceso de sustitución.

Es importante considerar que existen cuatro principios básicos que deben tomarse en cuenta para un buen diseño del LENGUAJE DE COMANDOS:

1. Modos de comando.- Algunos programas interactivos interpretan acciones de usuario en dos o más formas dependiendo - del estado del programa; por ejemplo, un editor de texto puede interpretar la letra 's' como una operación de 'suprimir'

durante un estado y como alguna parte de algun otro texto en otro estado.

Cada estado en el cual una operación proporcionada por el -- usuario es interpretada en forma diferente por el programa, es llamado un MODO DE COMANDO. Mientras más modos de comando existan en un lenguaje, mayor es la posibilidad de que existan errores por el usuario al olvidar en que 'modo' se encuentra en algún momento. Se recomienda que los LENGUAJES DE COMANDOS tengan un solo modo para evitar que surja este problema.

2. Secuencia en la selección.- La mayoría de las aplicaciones requieren que se seleccionen operandos en un comando; esto puede ser antes o después de que se ha especificado la operación.

Si la selección se realiza primero resultan menos modos, lo cual es lo más recomendable. Como ejemplo podemos considerar el comando ' SUPRIME LA GRAFICA G1 '. En el siguiente capítulo se explica el lenguaje de comandos utilizado con mayor detalle.

3. Mecanismo de interrupción de comando.- Muchos comandos involucran una secuencia de dos o más pasos; en cierto momento, por ejemplo a la mitad del comando, se puede dar el caso de que surja la necesidad de interrumpirlo, por tanto es necesario proporcionarle al usuario la facilidad de que lo pueda hacer. A menudo se utiliza la tecla de 'escape' por ejem-

plo, sin embargo, en algunos casos deben utilizarse otro tipo de herramientas.

4. Manejo de errores.- El diseñador del lenguaje de comandos debe considerar la manera en la cual el programa responderá a entradas erróneas. Es muy importante que el usuario se dé cuenta de su error, de otra manera él podría emitir una continua secuencia de comandos inválidos sin saber que son erróneos.

Además es importante considerar el estilo del LENGUAJE DE -- COMANDOS, i.e., debe existir una consistencia en él. No se debe cambiar constantemente al usuario de un dispositivo de entrada a otro. Si se decide adoptar una sintaxis de comandos de modo-simple, se debe estar seguro de que todos los -- comandos se apeguen a él, debido a que las excepciones invariablemente confundirán al usuario. La consistencia debe ser lo más obvia posible para que el usuario pueda anticipar la sintaxis y semántica de comandos no familiares.

Esta consistencia puede realizarse siguiendo un estilo. La opción que se elija, dependerá de varios factores incluyendo la configuración del hardware, el rango de objetos y acciones en el MODELO DEL USUARIO y los requerimientos específicos del mismo.

El tercer componente de la interfaz es la RETROALIMENTACION. Tiene como función auxiliar al usuario en la operación del sistema. La RETROALIMENTACION puede presentarse de varias formas:

- . A través de agradecimientos,
- . Por medio de mensajes explicativos,
- . Por la indicación de objetos seleccionados,
- . Solicitando digitar algún caracter especial, etc.

La RETROALIMENTACION ayuda al usuario a estar seguro de que los comandos emitidos son recibidos correctamente y entendidos por el sistema.

El cuarto y último componente de la interfaz lo constituye el DESPLIEGUE DE INFORMACION.

La imagen es una confirmación al usuario que su modelo es el correcto. Debe realizarse un buen esfuerzo a fin de que exista un realismo en la imagen desplegada.

3.- DESCRIPCION Y DISEÑO DEL LENGUAJE

3.1. Atributos, características y detalles de las gráficas.

Los objetivos que pretende cumplir el diseño del Lenguaje de graficación 'INTERGRAF' son los siguientes:

. Ser un Lenguaje con una notación y sintaxis consistente -- muy cercana a la del mismo Lenguaje Español; estar estructurado de tal manera que el significado y la secuencia de sus instrucciones sean fáciles de entender.

. Ser un Lenguaje flexible que pueda ser usado tanto por una persona que lo haya utilizado frecuentemente, en cuyo caso el usuario pudiera omitir algunas partes de un comando o bien, puede ser utilizado por una persona que nunca antes ha tenido contacto con 'INTERGRAF'. Si es este el caso, entonces el Lenguaje brinda la facilidad de servir de ayuda en forma interactiva, sobre todo en el uso de la sintaxis correcta que se requiera para emitir un comando específico, es decir, 'INTERGRAF' puede asumir opciones y parámetros a menos que el usuario indique otras instrucciones.

Realmente 'INTERGRAF' es un Lenguaje de Graficación verdadero, ya que los comandos que lo constituyen se relacionan entre sí de una manera sistemática y en conjunto definen una sintaxis, además de expresar una semántica adecuada.

Cada comando corresponde a una acción por realizar y los operandos de cada comando son los objetos que afecta la acción.

Ahora bien, con base en lo explicado anteriormente y en lo - detallado en el capítulo anterior, se mostrarán las principa les características, atributos y detalles de las gráficas -- que pueden dibujarse haciendo uso de 'INTERGRAF'.

MODELO DE LA TAREA.

Las tareas atómicas son:

1. DEFINIR algún objeto.
2. CAMBIAR algún objeto.
 - 2.1. SUPRIMIR una gráfica.
 - 2.2. MODIFICAR las características de una gráfica.
 - 2.3. AGREGAR un objeto a un grupo de gráficas.
3. DIBUJAR un objeto o conjunto de ellos.
4. LISTAR un objeto para ver sus características.
5. AYUDAR al usuario en la sintaxis del lenguaje.
6. ASIGNAR un objeto a una gráfica.

Ahora bien, con base en las tareas atómicas, se pueden des - cribir los OBJETOS que 'INTERGRAF' maneja, así como los atri butos que se pueden asignar a cada uno de ellos.

Definición de los OBJETOS.

1. PUERTO. Sus características consisten en tener asociado - un identificador y cuatro coordenadas (la sintaxis se mues-

tra en la gramática) que representan dos puntos en el graficador los cuales sirven como fronteras.

2. VENTANA. Sus características consisten, al igual que en el puerto, en tener asociado un identificador y cuatro coordenadas que permiten obtener el marco de referencia del mundo real que se pretende dibujar.

3. GRAFICA. Cada gráfica debe tener asociado primeramente un identificador, posteriormente una ventana, un puerto y los atributos propios de la misma; éstos son:

3.1. COLOR. Los colores que se pueden usar son ROJO, NEGRO, AZUL o VERDE.

3.2. ESCALA. Puede ser de tipo LINEAL, LOGARITMICA o SEMI LOGARITMICA.

3.3. TIPO DE LINEA. Puede ser CONTINUA, de RAYA o de PUNTO.

3.4. TIPO DE DIBUJO. Puede ser un HISTOGRAMA o CONTINUO (una función continua).

4. GRUPO. Se refiere a la facilidad de formar grupos de gráficas; cada grupo debe tener asociado, primeramente, un identificador; cada gráfica que lo forme deberá tener sus propios atributos y se le puede asignar algún puerto o ventana específico.

De los atributos de las gráficas mencionados, al momento de definir o dibujar un GRUPO, el usuario tiene la facilidad de asignar a una gráfica los atributos específicos que se requieran; en caso de que alguno de los atributos sea omitido, el sistema tomará el 'default' que tenga asignado.

En cuanto a la descripción de los OPERADORES, i.e., las ACCIONES u OPERACIONES que se pueden aplicar a los objetos se tienen los siguientes:

1. DEFINIR:

- 1.1. PUERTOS con su identificador y coordenadas respectivas.
- 1.2. VENTANAS con su identificador y coordenadas respectivas.
- 1.3. GRAFICAS con su identificador y atributos propios, además de tener la opción de asignarles un puerto o una ventana específica.
- 1.4. GRUPOS de gráficas con su identificador asociado. Además, cada gráfica que se defina al grupo, se le puede asignar una ventana y un grupo particular.

2. CAMBIAR:

- 2.1. Las definiciones de los PUERTOS.
- 2.2. Las definiciones de las VENTANAS.
- 2.3. Los atributos de las GRAFICAS.

2.4. A los GRUPOS de gráficas se les puede:

- 2.4.1. AGREGAR gráficas con sus atributos correspondientes.
- 2.4.2. SUPRIMIR gráficas del GRUPO.
- 2.4.3. ASIGNAR puertos o ventanas específicos para una gráfica en particular.
- 2.4.4. MODIFICAR los atributos de las gráficas en el GRUPO.

3. DIBUJAR:

- 3.1. Una GRAFICA o varias GRAFICAS con atributos especificados anteriormente o asignarlos al momento de dibujarlas.
- 3.2. SUMA o SUMAS DE GRAFICAS, con la opción de asignarles en ese momento atributos específicos y puertos y ventanas diferentes.
- 3.3. DIFERENCIA o DIFERENCIAS DE GRAFICAS, con las mismas opciones de la SUMA o SUMAS.
- 3.4. Un GRUPO o varios GRUPOS de gráficas, con la opción de asignarles un puerto y una ventana específicos.

4. LISTAR las características asociadas con:

- 4.1. PUERTOS.
- 4.2. VENTANAS.
- 4.3. GRUPOS DE GRAFICAS.
- 4.4. GRAFICAS específicas.

5. AYUDAR al usuario en la forma correcta de emitir algún comando específico; esto se puede lograr mostrando en pantalla el diagrama de sintaxis correspondiente a la acción que se quiera ejecutar, ya sea DEFINIR, CAMBIAR, DIBUJAR o LISTAR.

RESTRICCIONES DEL LENGUAJE.

A continuación se presentan las restricciones de 'INTERGRAF' dando a su vez el concepto que se debe tomar por 'default' - cuando se omite indicar algún atributo asociado con una gráfica o la ventana o puerto que se debe considerar:

Para dibujar una gráfica, su identificador se debió definir previamente; en caso de no haber definido sus atributos, ni ventana ni puerto asociado, se tomarán los siguientes conceptos por omisión:

Para la ventana, se tomarán los puntos mínimos y máximos de toda la gráfica a dibujar.

Para el puerto, se tomará el área completa del graficador.

Para los atributos de las gráficas, el color será negro, el tipo de línea continua y el tipo de dibujo también continuo.

Para cambiar ya sea un puerto, una ventana, una gráfica o un grupo de gráficas, éstos debieron definirse previamente.

3.2. Gramática en BNF y Diagramas de Sintaxis.

Antes de mostrar los diagramas de sintaxis del Lenguaje 'INTERGRAF', se mencionarán algunos conceptos relacionados con gramáticas y lenguajes.

En términos generales se pueden tener las siguientes definiciones:

Gramática.- Es la descripción sintáctica de un lenguaje de programación.

Lenguaje.- Es un subconjunto del conjunto de todas las secuencias de palabras ó símbolos de algún vocabulario básico.

Alfabeto.- Es un conjunto finito no vacío de elementos a los cuales se les llama símbolos.

Cadena.- Cualquier secuencia finita de símbolos de un alfabeto.

Cadena vacía.- No tiene símbolos; en este trabajo se representará por el carácter '"'."

Longitud de una cadena.- Es el número de símbolos que existen en una cadena.

Definición formal de Gramática:

Para definir una Gramática $G(Z)$ se deben definir cuatro elementos los cuales son:

1. V_T al cual se le llama el alfabeto de la gramática y consiste del conjunto de símbolos TERMINALES con los que se construirán las SENTENCIAS del Lenguaje $L(G)$.

2. V_N el cual se define como el conjunto de símbolos NO-TERMINALES o categorías gramaticales.

3. Z se define como el símbolo inicial. A partir de este símbolo se derivarán todas las sentencias en $L(G)$.

4. P se define como el conjunto de todas las producciones. - Una producción es una regla que indica cómo se puede generar una FORMA SENTENCIAL a partir de otras.

Por otro lado tenemos:

Definición.- 'V' produce 'W' se indica $V \xRightarrow{+} W$ si existe una secuencia de derivaciones directas.

$$V=U_0 \Rightarrow U_1 \Rightarrow U_2 \Rightarrow \dots \Rightarrow U[n]=W$$

donde n debe ser mayor a 0. La secuencia es llamada una derivación de longitud ' n '.

Definición.- Sea $G(Z)$ una gramática. Una cadena X es llamada FORMA SENTENCIAL, si X se deriva del símbolo -- inicial Z , es decir, si $Z \xRightarrow{+} X$.

Una SENTENCIA es una FORMA SENTENCIAL consistente solamente de símbolos terminales. De esta manera se tiene que el Lenguaje $L(G[Z])$ es el conjunto de SENTENCIAS:

$$L(G) = \left\{ X \mid Z \xRightarrow{+} X \text{ y } X \text{ está en } V_T^+ \right\}$$

Con base en lo anterior tenemos que el Lenguaje es justamente un subconjunto del conjunto de todas las cadenas terminales (cadenas sobre V_T). La estructura de una sentencia está dada por la gramática. Sin embargo, algunas gramáticas diferentes pueden generar el mismo Lenguaje. Normalmente se dice "una sentencia de la gramática" cuando en realidad debe decirse "una sentencia del Lenguaje definido por la gramática".

Al imponer ciertas restricciones a las Gramáticas se pueden considerar gramáticas del tipo 1, del tipo 2 y del tipo 3.

Gramática del tipo 1.

Esta gramática se define como aquella en la cual el consecuente de cada producción contiene al menos igual número de símbolos que el antecedente.

Formalmente, las reglas cuya forma sea:

$$X \Rightarrow Y$$

deben cumplir con la restricción $|X| \leq |Y|$

Una Gramática es del tipo 1 si cada producción de ella es de la forma: $X A Y \Rightarrow X Z Y$

donde: $X, Y \in (V_N \cup V_T)^*$

y $A \in V_N$

$Z \in (V_N \cup V_T)^+$

A estas gramáticas se les llama dependientes del contexto.

Gramática del tipo 2.

También llamadas "Libres del contexto", se definen como aquellas en las cuales todas sus producciones son de la forma:

$$A \Rightarrow X$$

donde:

$$A \in V_N \text{ y } X \in (V_N \cup V_T)^+$$

Las restricciones anteriores significan que el antecedente de cada producción es un solo símbolo del alfabeto de no-terminales y que el consecuente es cualquier cadena formada con los símbolos de ambos alfabetos (El de no-terminales V_N y el de terminales V_T).

Gramática del tipo 3.

A estas gramáticas también se les conoce como 'regulares', 'de estados finitos' o 'lineales unilaterales'. Serán de este tipo sólo si todas sus producciones son de la forma:

$$\begin{aligned} A \Rightarrow X & \quad \text{o} \quad A \Rightarrow XB \text{ (Regular derecha)} \\ & \quad \text{o} \quad A \Rightarrow BX \text{ (Regular izquierda)} \end{aligned}$$

donde:

$$A, B \in V_N \text{ y } X \in V_T^+$$

Toda gramática 'Regular izquierda' tiene una gramática 'Regular derecha' equivalente.

Una vez explicado lo anterior, a continuación se muestra la gramática del lenguaje 'INTERGRAF' expresada en la forma --- "BACKUS NAUR"; el caracter '/' se utilizará para indicar el 'O' en cada una de las SENTENCIAS.

Primeramente se indican los elementos de los conjuntos tanto de V_T (símbolos TERMINALES) como de V_N (símbolos NO-TERMINALES).

CONJUNTO V_T (PALABRAS RESERVADAS):

DEFINE	PUERTOS	GRAFICAS
CAMBIA	LOG	DIFERENCIA
DIBUJA	COLOR	DIFERENCIAS
LISTA	ESCALA	SUMA
AYUDA	TIPO	SUMAS
EXPLICA	ROJO	VENTANAS
FIN	AZUL	CONTINUO
VENTANA	NEGRO	[
PUERTO	VERDE	*
GRAFICA	LINEAL	.
GRUPO	UNA	-
SEMILOG	PUERTOS	:
LA	LINEA	;
LAS	DIBUJO	E
UN	CONTINUA]
EL	PUNTO	0 1 ... 9
LOS	RAYA	A B ... Z
HISTOGRAMA	ASIGNA	+
AGREGA	SUPRIME	-
MODIFICA	GRUPOS	

CONJUNTO V_N :

<COMPLE 1>	<COMPLE 2>	<COMPLE 3>
<COMPLE 4>	<COMPLE 5>	<ARPRE 1>
<ARPRE 2>	<PRED 1>	<PRED 2>
<PRED 3>	<IDENT>	<PUNTOS>
<LISTA>	<REP 1>	
<NUM>	<COR1>	<CORD>
<ATRIBUTOS>	<C1>	<E1>
<T1>	<TONO>	<REP 2>
<TIPE>	<TIESC>	<LIGRA>
<TILIN>	<TIGRAF>	<REP 3>
<VARIOS>	<REP 4>	<ALGO>
<MODASIG>	<ARS 1>	<AVEPU>
<LISID>	<REP 5>	<PRED 4>
<COSAS>	<MODAG>	<GRAFS>
<MODMOD>	<ATGRAS>	<ATRIE>
<ARFEMS>	<GRAFS 2>	<AGRAF>
<DIGRAF>	<SIGNO>	<REP 6>
<ARMAS>	<GRUPS>	<PUGRUP>
<GRAVENT>	<PUER>	<VENTS>
<VERBO>	<LET>	<DIG>
<SIG>	<REAL>	<EXP>
<ENTERO>		

GRAMATICA EXPRESADA EN "BACKUS NAUR FORM"

(B N F)

POSIBLES COMANDOS

COMANDO	→	DEFINE	<COMPLE 1>	/
		CAMBIA	<COMPLE 2>	/
		DIBUJA	<COMPLE 3>	/
		LISTA	<COMPLE 4>	/
		AYUDA	<COMPLE 5>	/
		EXPLICA	<COMPLE 5>	/
		FIN		

PRODUCCIONES DE <COMPLE 1> :

<COMPLE 1> → <ARPRE 1> VENTANA <PRED 1> /
 <ARPRE 2> PUERTO <PRED 1> /
 <ARPRE 1> GRAFICA <PRED 2> /
 <ARPRE 2> GRUPO <PRED 3>

<ARPRE 1> → UNA / LA / LAS / &
<ARPRE 2> → UN / EL / LOS / &

PRODUCCIONES DE <PRED 1> :

<PRED 1> → <IDENT><PUNTOS><LISTA><REP1>

<PUNTOS> → : / &
<LISTA> → <NUM>b<NUM>b<NUM>b<NUM>
<REP 1> → , <IDENT><PUNTOS><LISTA><REP 1>/&

DONDE: 'b' significa un espacio en blanco.

PRODUCCIONES DE <PRED 2>:

<PRED 2> → <IDENT> /

<IDENT> <CORI> <ATRIBUTOS> <CORD> /

<IDENT> <CORI> <ATRIBUTOS> <CORD> <ALGO>

<CORI> → [

<CORD> →]

<ATRIBUTOS> → COLOR <C1> /

ESCALA <E1> /

TIPO <T1>

<C1> → <TONO> <REP2>

<TONO> → ROJO / AZUL / NEGRO / VERDE

<E1> → <TIPE> <TIESC> <REP2>

<TIPE> → TIPO / &

<TIESC> → LINEAL / LOG / SEMILOG

<T1> → <LIGRA> <REP2>

<LIGRA> → LINEA <TILIN> /

DIBUJO <TIGRAF>

<TILIN> → CONTINUA / PUNTO / RAYA

<TIGRAF> → HISTOGRAMA / CONTINUO

<REP2> → , <ATRIBUTOS> <REP2> / &

PRODUCCIONES DE <PRED 3>:

<PRED 3> → <IDENT><PUNTOS><IDENT><REP 3><VARIOS>

<VARIOS> → <CORI> <ATRIBUTOS> <CORD> <REP. 4> <ALGO> /
* <MODASIG> / &

<ALGO> → * <MODASIG> / &

<REP 3> → , <IDENT> <REP 3> / &

<REP 4> → <REP 3><CORI><ATRIBUTOS><CORD><REP 4> / &

MODULO ASIGNA:

<MODASIG> → ASIGNA <ARS 1> <AVEPU> <LISID> <REP 5>

<AVEPU> → VENTANA / PUERTO

<LISID> → <IDENT> / <LISTA>

<ARS 1> → EL / LA / &

<REP 5> → , <ARS 1> <AVEPU> <LISID> <REP 5> / &

PRODUCCIONES DE <COMPLE 2>:

<COMPLE 2> → <ARPRE 1> VENTANA <PRED 1> /

<ARPRE 2> PUERTO <PRED 1> /

<ARPRE 1> GRAFICA <PRED 2> /

<ARPRE 2> GRUPO <PRED 4>

PRODUCCIONES DE <PRED 4>:

<PRED 4> → <IDENT> <PUNTOS> <COSAS>

<COSAS> → AGREGA <MODAG> /
SUPRIME <ARPRE 1><GRAFS><IDENT><REP 3> /
* <MODASIG> /
MODIFICA <MODMOD>

<GRAFS> → GRAFICA / GRAFICAS / &

MODULO AGREGA:

<MODAG> → <ARPRE 1><GRAFS><IDENT><REP 3><ATGRAS>

<ATGRAS> → <COR1> <ATRIBUTOS> <CORD> <REP 4> / &

MODULO MODIFICA:

<MODMOD> → <ARPRE 1><GRAFS><IDENT><REP 3><ATRIE>

<ATRIE> → <COR1> <ATRIBUTOS> <CORD> / &

PRODUCCIONES DE <COMPLE 3>:

<COMPLE 3> → <ARFEMS><GRAFS 2><IDENT><REP 3><AGRAF> /

<ARFEMS><DIGRAF><IDENT><SIGNO><IDENT><REP 6><VARIOS> /

<ARMAS><GRUPS><IDENT><REP 3><VARIOS>

<ARFEMS> → LA / LAS / &

<ARMAS> → EL / LOS / &

<GRAFS 2> → GRAFICA / GRAFICAS

<AGRAF> → <CORI><ATRIBUTOS><CORD><REP 4><ALGO> / &

<DIGRAF> → DIFERENCIA / DIFERENCIAS /
SUMA / SUMAS

<SIGNO> → + / -

<REP 6> → , <IDENT><SIGNO><IDENT><REP 6> / &

<GRUPS> → GRUPO / GRUPOS

PRODUCCIONES DE <COMPLE 4>:

<COMPLE 4> → <ARMAS><PUGRUP><IDENT><REP 3> /
<ARFEMS><GRAVENT><IDENT><REP 3>

<PUGRUP> → <PUER> / <GRUPS>

<PUER> → PUERTO / PUERTOS

<GRAVENT> → <GRAFS 2> / <VENTS>

<VENTS> → VENTANA / VENTANAS

PRODUCCIONES DE <COMPLE 5>:

<COMPLE 5> → <VERBO>.

<VERBO> → DEFINE /
CAMBIA /
DIBUJA /
LISTA /
&

PRODUCCIONES ESPECIALES:

$\langle \text{IDENT} \rangle \longrightarrow \langle \text{LET} \rangle (\langle \text{LET} \rangle / \langle \text{DIG} \rangle)^*$

$\langle \text{LET} \rangle \longrightarrow A / B / C / D / E / \dots / Z$

$\langle \text{DIG} \rangle \longrightarrow 0 / 1 / 2 / 3 / 4 / \dots / 9$

$\langle \text{NUM} \rangle \longrightarrow \langle \text{SIG} \rangle \langle \text{REAL} \rangle \langle \text{EXP} \rangle$

$\langle \text{EXP} \rangle \longrightarrow E \langle \text{SIG} \rangle \langle \text{DIG} \rangle \langle \text{DIG} \rangle / \&$

$\langle \text{REAL} \rangle \longrightarrow \langle \text{ENTERO} \rangle /$
 $\langle \text{ENTERO} \rangle . /$
 $. \langle \text{ENTERO} \rangle /$
 $\langle \text{ENTERO} \rangle . \langle \text{ENTERO} \rangle$

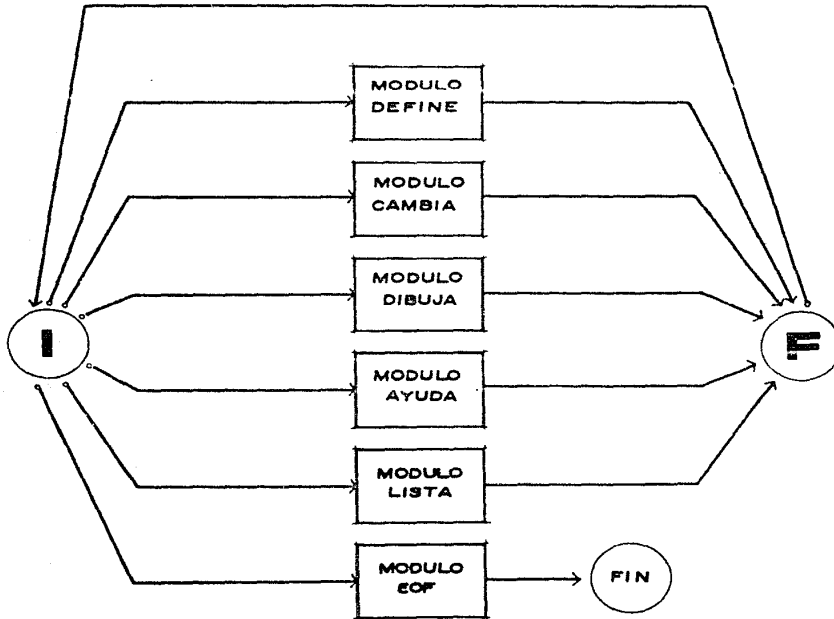
$\langle \text{ENTERO} \rangle \longrightarrow \langle \text{DIG} \rangle / \langle \text{DIG} \rangle \langle \text{ENTERO} \rangle$

$\langle \text{SIG} \rangle \longrightarrow - / + / \&$

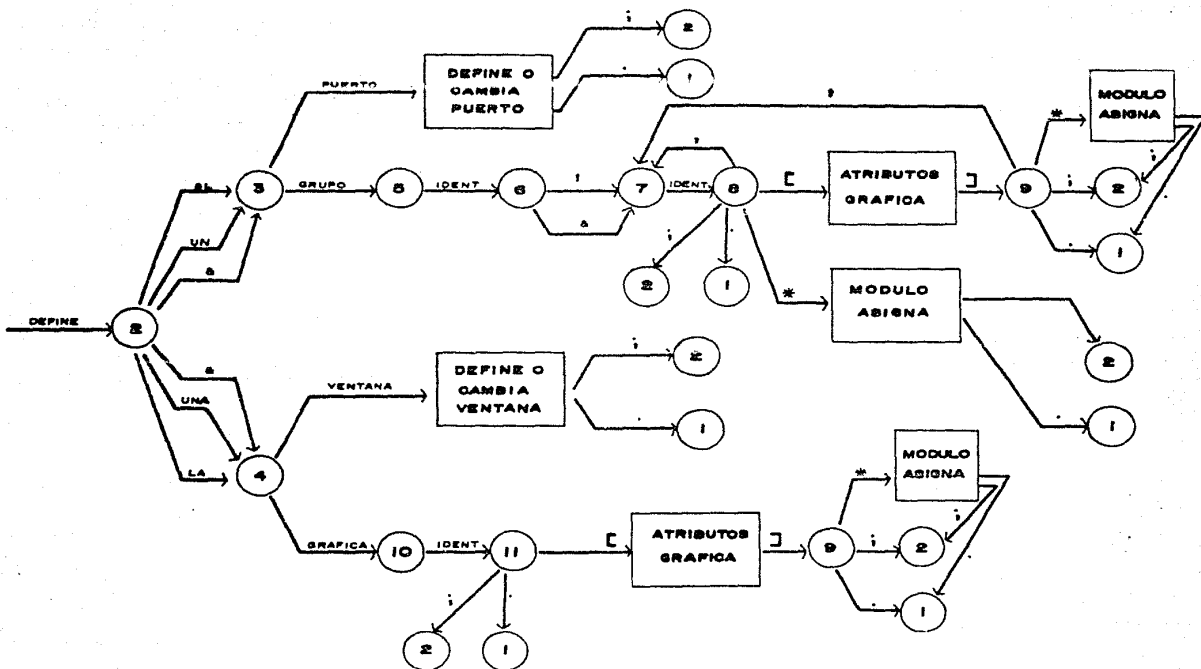
$\langle \text{DEL} \rangle \longrightarrow ; / .$

DIAGRAMAS DE SINTAXIS

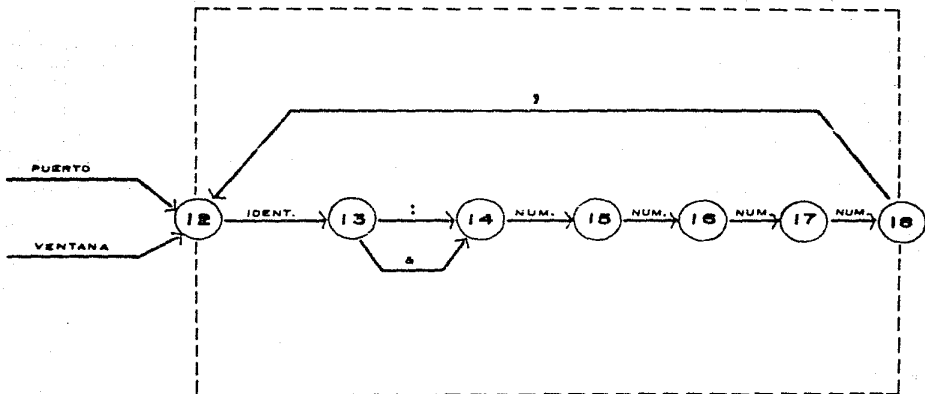
MODULO GENERAL



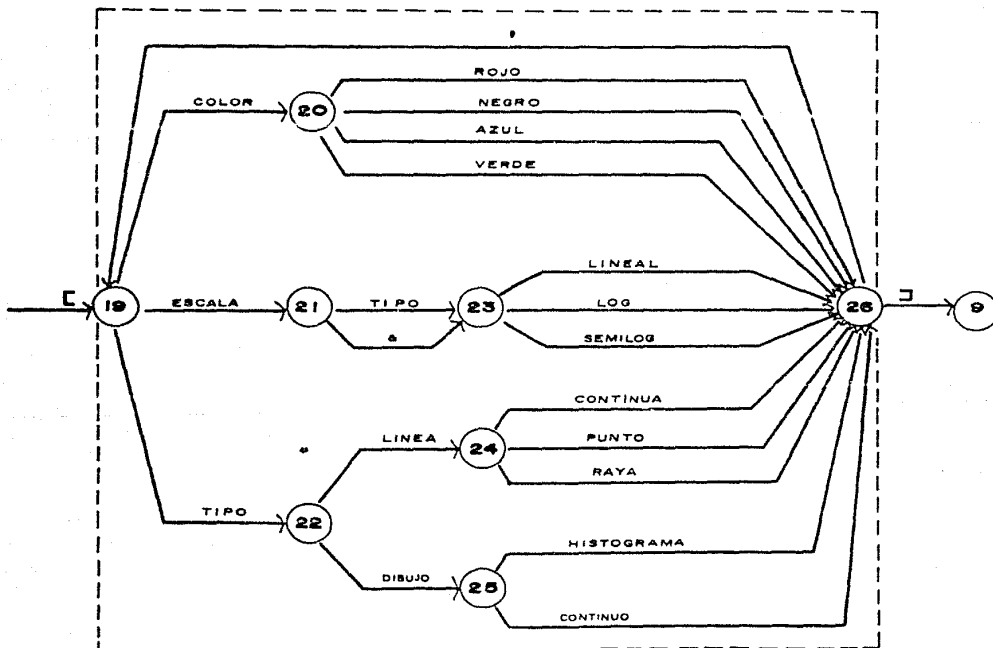
MODULO DEFINE



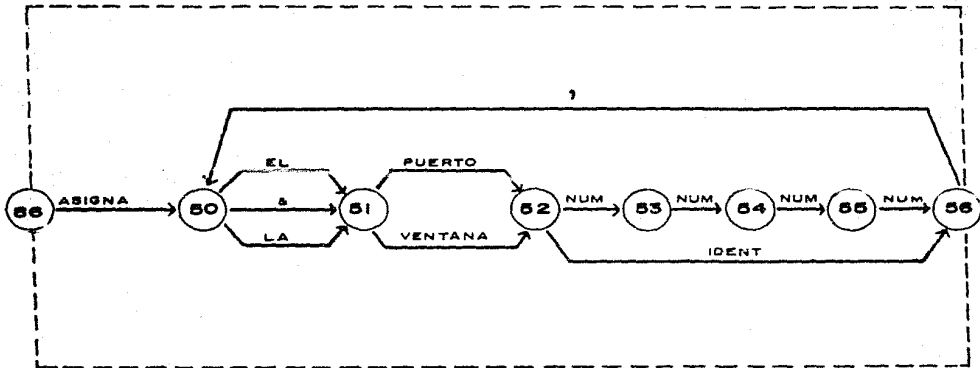
MODULO DEFINE O CAMBIA PUERTO y/O VENTANA



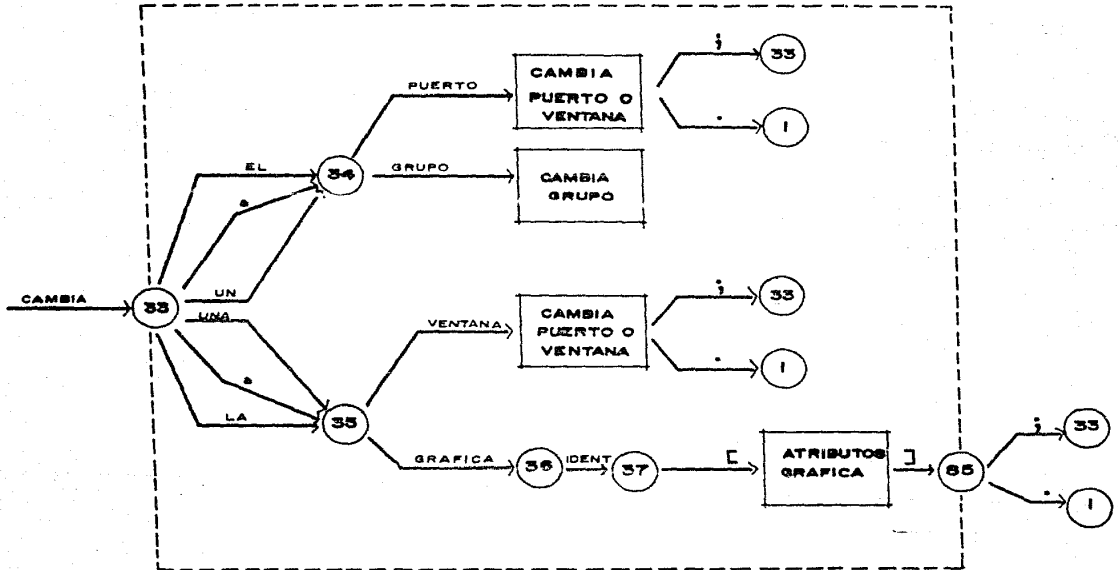
DEFINE O CAMBIA ATRIBUTOS DE GRAFICA



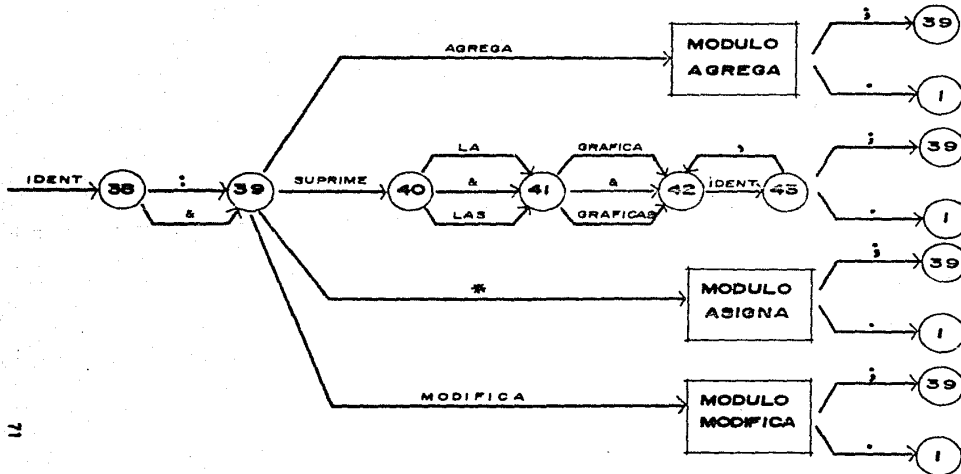
MODULO ASIGNA



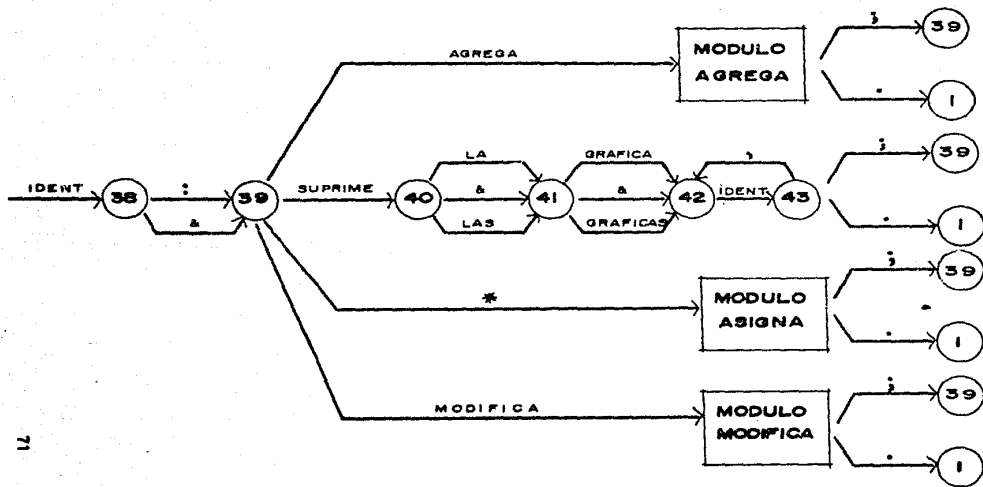
MODULO CAMBIA



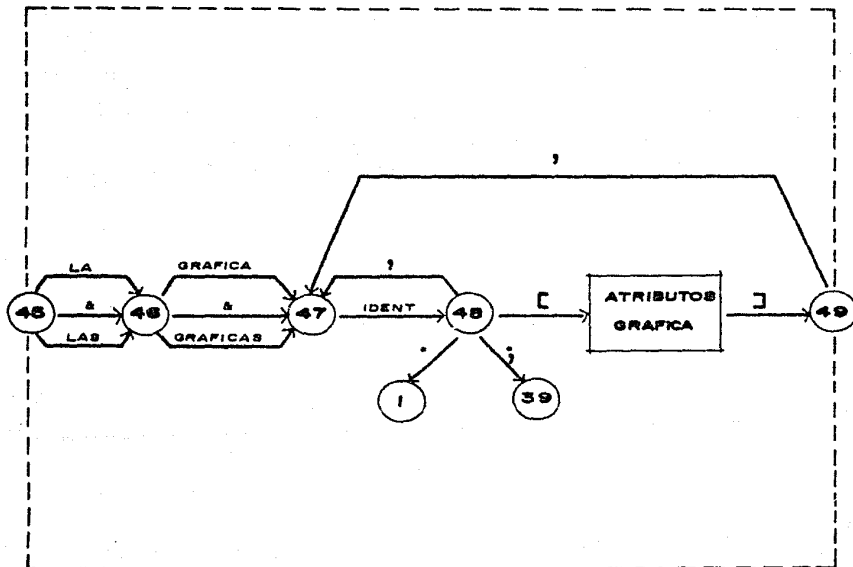
MODULO CAMBIA GRUPO



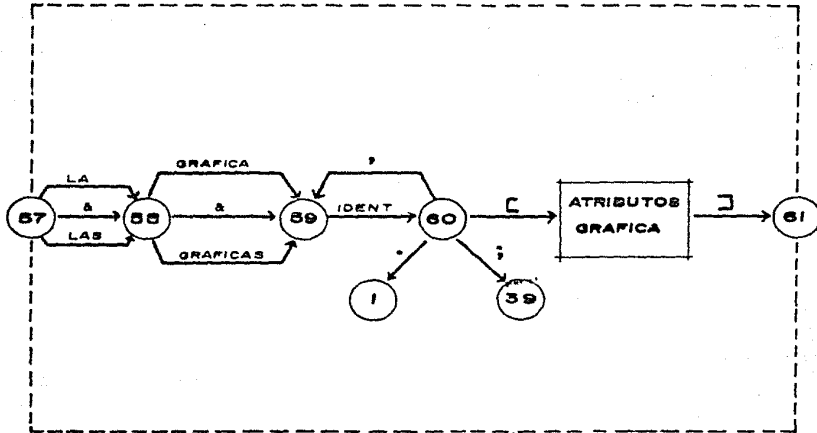
MODULO CAMBIA GRUPO



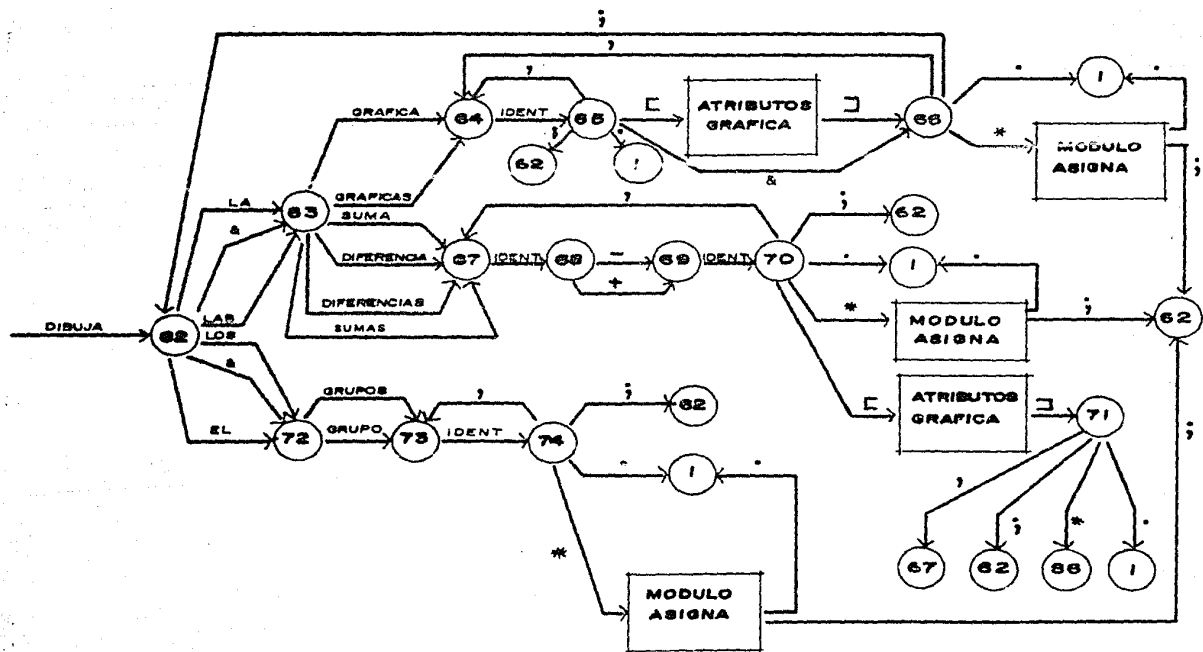
MODULO AGREGA



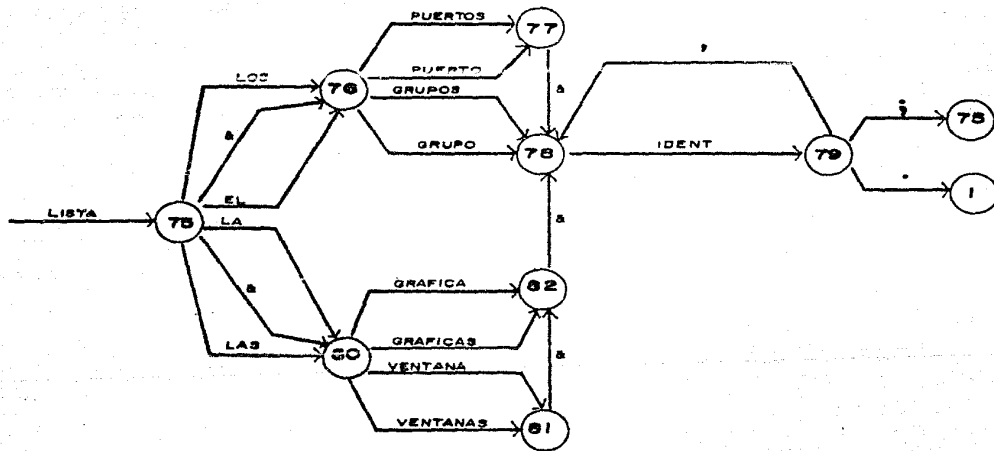
MODULO MODIFICA



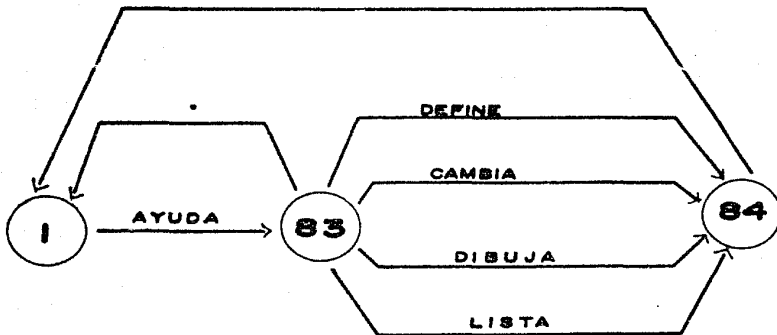
MODULO DIBUJA



MODULO LISTA



MODULO AYUDA Y EXPLICA



4.- EJEMPLOS Y CONCLUSIONES

Para mostrar lo flexible y poderoso que puede ser el Lenguaje 'INTERGRAF' al implantarse en un equipo de cómputo, a continuación se presentan algunos ejemplos . La manera en como se muestran es la siguiente:

1. Primeramente se indican los requerimientos específicos para realizar una gráfica usando 'INTERGRAF'.
2. Posteriormente se muestra el comando o cadena de ellos -- del Lenguaje que le permitirá al usuario satisfacer sus requerimientos.
3. Se muestran las producciones en su secuencia lógica que es necesario aplicar para lograr que el comando emitido sea válido, i.e., que está formado por símbolos existentes en la gramática de 'INTERGRAF' además de estar en la secuencia correcta.
4. Finalmente se muestran las partes específicas de los -- diagramas de sintaxis que permiten emitir el comando, haciendo resaltar la trayectoria que corresponda.

EJEMPLO 1.

Se requiere definir una gráfica con el nombre de GRI de color rojo, con escala lineal, que el tipo de línea sea continua y que el tipo del dibujo sea un histograma; además, a GRI se le desea asignar el puerto en el graficador de coordenadas $X1=2.2$, $Y1=4.5$, $X2=23.5$, $Y2=16.8$ y la ventana VTA1 que fue definida previamente. Por otro lado, también se desea definir un grupo de gráficas de nombre GRUP11, el cual esté formado por las gráficas GR2, GR3 y GR4. Los atributos de las dos primeras serán los de 'default', es decir, de color azul, escala lineal, tipo de línea continua y tipo de dibujo continuo; sin embargo, la gráfica GR4 debe definirse de color negro, escala logarítmica, el tipo de línea debe ser de puntos y el tipo de gráfica ó dibujo debe ser continuo. Para lograr el requerimiento anterior sería necesario emitir el siguiente comando:

```
DEFINE LA GRAFICA GRI[COLOR ROJO,ESCALA TIPO LINEAL,TIPO LINEA CONTINUA,TIPO DIBUJO HISTOGRAMA]*ASIGNA EL PUERTO 2.2 4.4 20.2 15.9,LA VENTANA VTA1;EL GRUPO GRUP11:GR2,GR3,GR4[COLOR NEGRO,ESCALA TIPO LOG,TIPO LINEA PUNTO,TIPO DIBUJO CONTINUO].
```

Secuencia de producciones:

COMANDO → DEFINE <COMPLE1>

→ DEFINE <ARPRE> GRAFICA <PRE2>

→ DEFINE LA GRAFICA <IDENT> <CORI>
<ATRIBUTOS> <CORD> <ALGO>

→ DEFINE LA GRAFICA <LET> (<LET>/<DIG>)*
<CORI> <ATRIBUTOS> <CORD> <ALGO>

→ DEFINE LA GRAFICA GRI [COLOR <C1> <CORD>
<ALGO>

→ DEFINE LA GRAFICA GRI [COLOR <TONO> <REP2>
<CORD> <ALGO>

→ DEFINE LA GRAFICA GRI [COLOR ROJO <REP2>
<CORD> <ALGO>

→ DEFINE LA GRAFICA GRI [COLOR ROJO ,
<ATRIBUTOS> <REP2> <CORD> <ALGO>

→ DEFINE LA GRAFICA GRI [COLOR ROJO ,
<ESCALA> <E1> <REP2> <CORD> <ALGO>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
<TIPE> <TIESC> <REP2> <REP2> <CORD> <ALGO>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO <TIESC> <REP2> <REP2> <CORD> <ALGO>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL <REP2> <REP2> <CORD> <ALGO>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , <ATRIBUTOS> <REP2> <REP2> <CORD> <ALGO>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO <T1> <REP2> <REP2> <CORD> <ALGO>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO <LIGRA> <REP2> <REP2> <CORD> <ALGO>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA <TILIN> <REP2> <REP2> <CORD>
<ALGO>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA <REP2> <REP2> <CORD>
<ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , <ATRIBUTOS> <REP2> <REP2>
<CORD> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO <T1> <REP2> <REP2>
<CORD> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO <LIGRA><REP2> <REP2>
<CORD> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO <TIGRAF>
<REP2> <REP2> <CORD> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
<ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* <MODASIG>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA <ARSL> <AVEPU> <LISID> <REP5>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL <AVEPU> <LISID> <REP5>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO <LISID> <REP5>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO <LISTA> <REP5>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO <NUM> <NUM> <NUM> <NUM> <REP5>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO <SIG> <REAL> <EXP> <NUM> <NUM> <NUM>
<REP5>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO <ENTERO> . <ENTERO> <NUM> <NUM> <NUM>
<REP5>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO <DIGITO> . <DIGITO> <NUM> <NUM> <NUM>
<REP5>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2 . 2 <NUM> <NUM> <NUM> <REP5>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2 . 2 <SIG> <REAL> <EXP> <NUM> <NUM>
<REP5>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2 . 2 <ENTERO> . <ENTERO><NUM> <NUM>
<REP5>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2 . 2 <DIGITO> . <DIGITO> <NUM> <NUM>
<REP5>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 <NUM> <NUM> <REP5>

—————▶ DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 <SIG> <REAL> <EXP> <NUM>
<REPS>

—————▶ DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 <ENTERO> . <ENTERO> <NUM>
<REPS>

—————▶ DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 <DIGITO> <ENTERO> . <DIGITO>
<NUM> <REPS>

—————▶ DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 <DIGITO> <DIGITO> . <DIGITO>
<NUM> <REPS>

—————▶ DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 <NUM> <REPS>

—————▶ DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 <SIG> <REAL> <EXP>
<REP5>

—————▶ DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 <ENTERO>. <ENTERO>
<REP5>

—————▶ DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 <DIGITO> <ENTERO>.
<DIGITO> <REP5>

—————▶ DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 <DIGITO> <DIGITO>.
<DIGITO> <REP5>

—————▶ DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 <REP5>

—————▶ DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , <ARS 1>
<AVEPU> <LISID> <REP5>

—————→ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA <AVEPU>
<LISID> <REP5>

—————→ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
<LISID> <REP5>

—————→ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
<LET> (<LET>/ <DIG>) * <REP5>

—————→ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
<VTAL>

—————→ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VTAL ;

NOTA: Debido a lo potente y flexible que puede ser este Len-
guaje, al momento que se aplica la producción DEL → ; / .

y en caso de sustituir por ; el control se transfiere al nodo inmediatamente después del verbo inicial que inició el comando, en este caso es DEFINE . En este sentido y observando la gramática en B.N.F. se verá que el siguiente símbolo NO-TERMINAL después de DEFINE es <COMPLE 1>

Con base en lo anterior y para terminar el requerimiento del ejemplo 1 se tiene lo siguiente:

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VTAL ; <COMPLE 1>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VTAL ; <ARPRE 2> GRUPO <PRED 3>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VTAL ; EL GRUPO <PRED 3>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT11 ; EL GRUPO <IDENT> <PUNTOS> <IDENT> <REP 3> <VARIOS>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT11 ; EL GRUPO <LET> (<LET> / <DIG>) * <PUNTOS> <IDENT>
<REP 3> <VARIOS>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT11 ; EL GRUPO GRUP11 <PUNTOS> <IDENT> <REP 3> <VARIOS>

—————> DEFINE LA GRAFICA GRI [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT11 ; EL GRUPO GRUP11 : <IDENT> <REP 3> <VARIOS>

—————→ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : <LET> (<LET> / <DIG>) * <REP 3>
<VARIOS>

—————→ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 <REP 3> <VARIOS>

—————→ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , <IDENT> <REP 3> <VARIOS>

—————→ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , <LET> (<LET> / <DIG>) *
<REP 3> <VARIOS>

—————→ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 <REP 3> <VARIOS>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , <IDENT> <REP 3> <VARIOS>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , <LET> (<LET> / <DIG>)
<REP 3> <VARIOS>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 <VARIOS>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 <CORI> <ATRIBUTOS>
<CORD> <REP4> <ALGO>

—————▶ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [<ATRIBUTOS> <CORD>
<REP4> <ALGO>

—————▶ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR <Cl> <CORD>
<REP4> <ALGO>

—————▶ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR <TONO><REP2>
<CORD> <REP4> <ALGO>

—————▶ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO <REP2>
<CORD> <REP4> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO ,
<ATRIBUTOS> <REP2> <CORD> <REP4> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO ,
ESCALA <E1> <REP2> <CORD> <REP4> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO ,
ESCALA <TIPE> <TIESC> <REP2> <CORD> <REP4> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO ,
ESCALA TIPO <TIESC> <REP2> <CORD> <REP4> <ALGO>

—————▶ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO , ESCALA TIPO LOG <REP2> <CORD> <REP4> <ALGO>

—————▶ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO , ESCALA TIPO LOG , <ATRIBUTOS> <REP2> <CORD> <REP4> <ALGO>

—————▶ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO , ESCALA TIPO LOG , TIPO <T1><REP2> <CORD> <REP4> <ALGO>

—————▶ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO , ESCALA TIPO LOG , TIPO <LIGRA> <REP2> <CORD> <REP4> <ALGO>

—————▶ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO , ESCALA TIPO LOG <REP2> <CORD> <REP4> <ALGO>

—————▶ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO , ESCALA TIPO LOG , <ATRIBUTOS> <REP2> <CORD> <REP4> <ALGO>

—————▶ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO , ESCALA TIPO LOG , TIPO <T1><REP2> <CORD> <REP4> <ALGO>

—————▶ DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO , ESCALA TIPO LOG , TIPO <LIGRA> <REP2> <CORD> <REP4> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO ,
ESCALA TIPO LOG , TIPO LINEA <TILIN> <REP2> <CORD> <REP4>
<ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO ,
ESCALA TIPO LOG , TIPO LINEA PUNTO <REP2> <CORD> <REP4>
<ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO ,
ESCALA TIPO LOG , TIPO LINEA PUNTO , <ATRIBUTOS><REP2>
<CORD> <REP4> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO ,
ESCALA TIPO LOG , TIPO LINEA PUNTO , TIPO <T1> <REP2>
<CORD> <REP4> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO , ESCALA TIPO LOG , TIPO LINEA PUNTO , TIPO <LIGRA><REP2><REP2> <CORD> <REP4> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA ;
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO , ESCALA TIPO LOG , TIPO LINEA PUNTO , TIPO DIBUJO<TIGRAF><REP2> <REP2> <CORD> <REP4> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO , ESCALA TIPO LOG , TIPO LINEA PUNTO , TIPO DIBUJO CONTINUO <REP2> <REP2> <CORD> <REP4> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA VTA1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO , ESCALA TIPO LOG , TIPO LINEA PUNTO , TIPO DIBUJO CONTINUO <CORD> <REP4> <ALGO>

—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO ,
ESCALA TIPO LOG , TIPO LINEA PUNTO , TIPO DIBUJO
CONTINUO] <REP4> <ALGO>

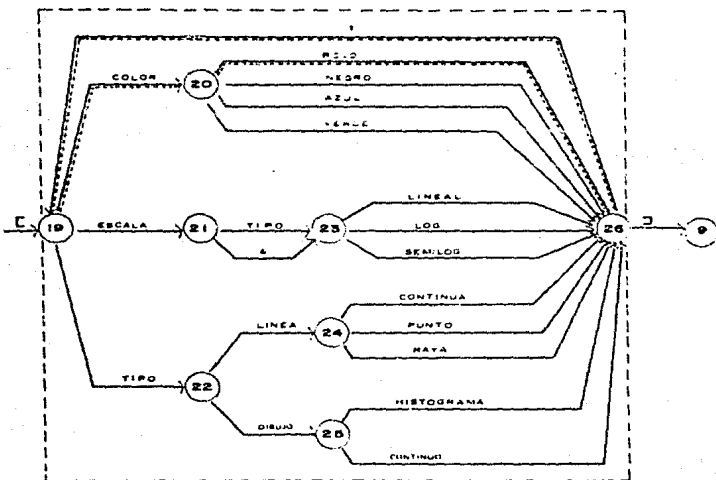
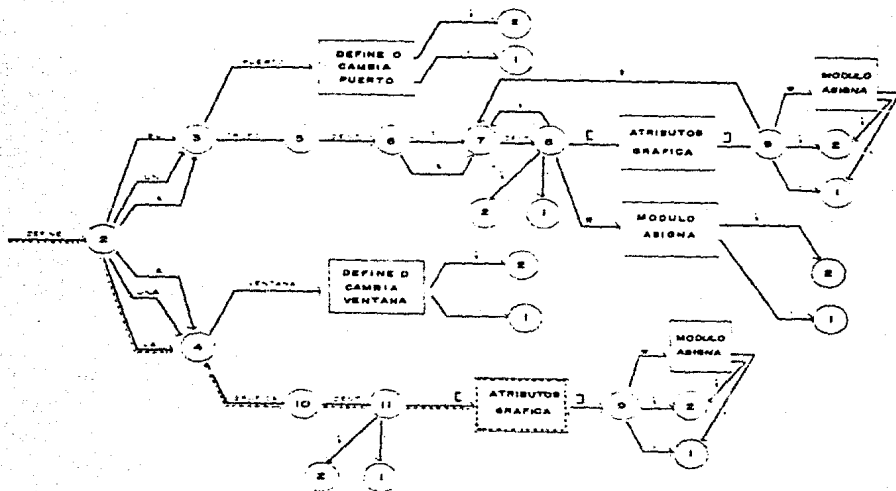
—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO ,
ESCALA TIPO LOG , TIPO LINEA PUNTO , TIPO DIBUJO
CONTINUO] <ALGO>

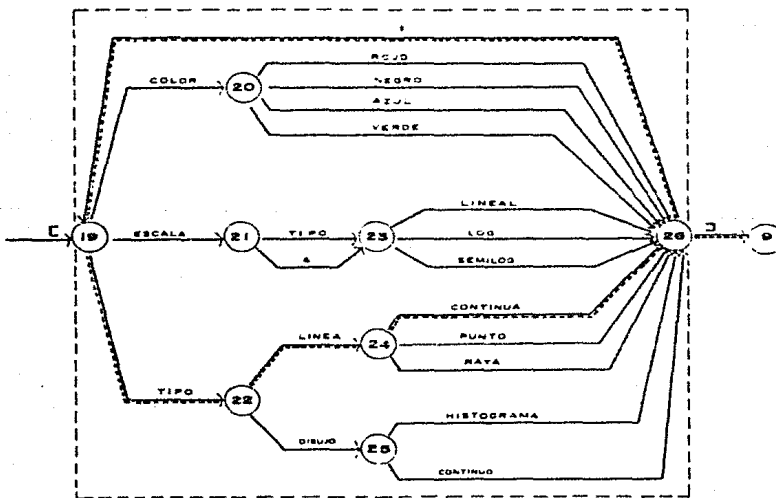
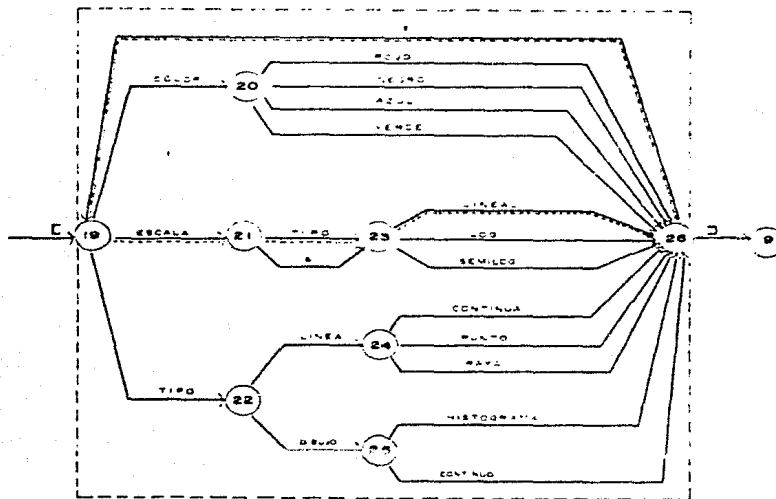
—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO ,
ESCALA TIPO LOG , TIPO LINEA PUNTO , TIPO DIBUJO
CONTINUO]

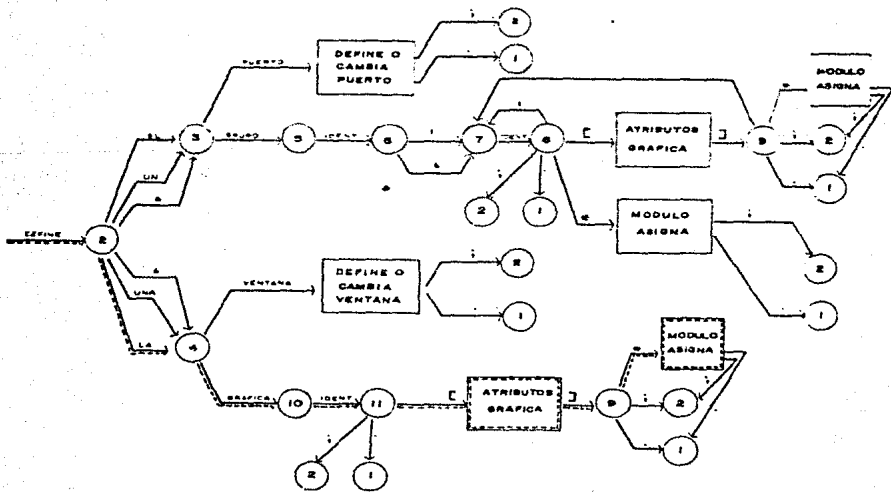
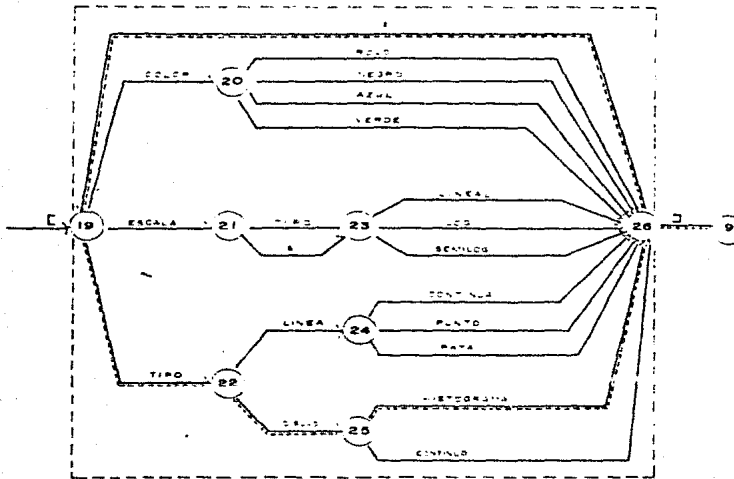
—————> DEFINE LA GRAFICA GR1 [COLOR ROJO , ESCALA
TIPO LINEAL , TIPO LINEA CONTINUA , TIPO DIBUJO HISTOGRAMA]
* ASIGNA EL PUERTO 2.2 4.4 20.2 15.9 , LA VENTANA
VT1 ; EL GRUPO GRUP11 : GR2 , GR3 , GR4 [COLOR NEGRO ,
ESCALA TIPO LOG , TIPO LINEA PUNTO , TIPO DIBUJO
CONTINUO] .

DIAGRAMAS DE SINTAXIS DEL EJEMPLO 1

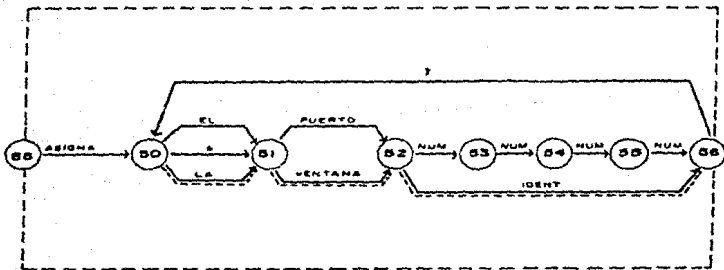
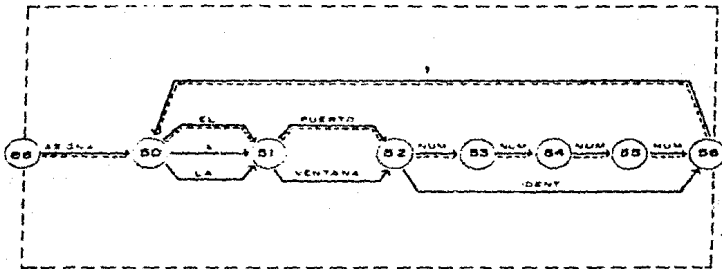
MODULO DEFINE

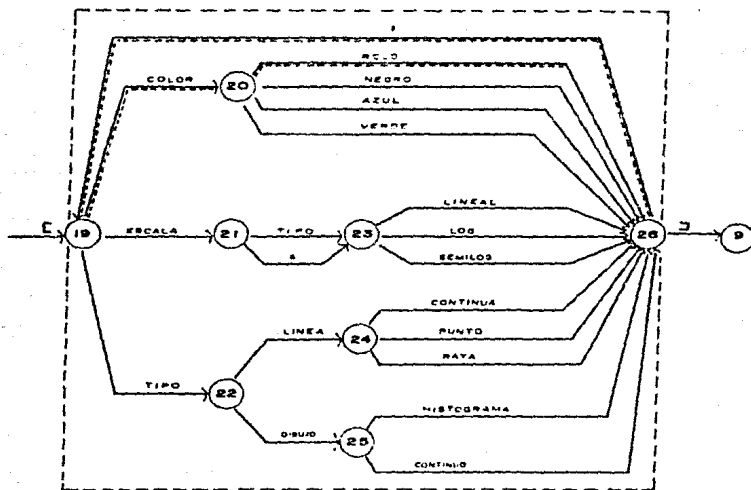
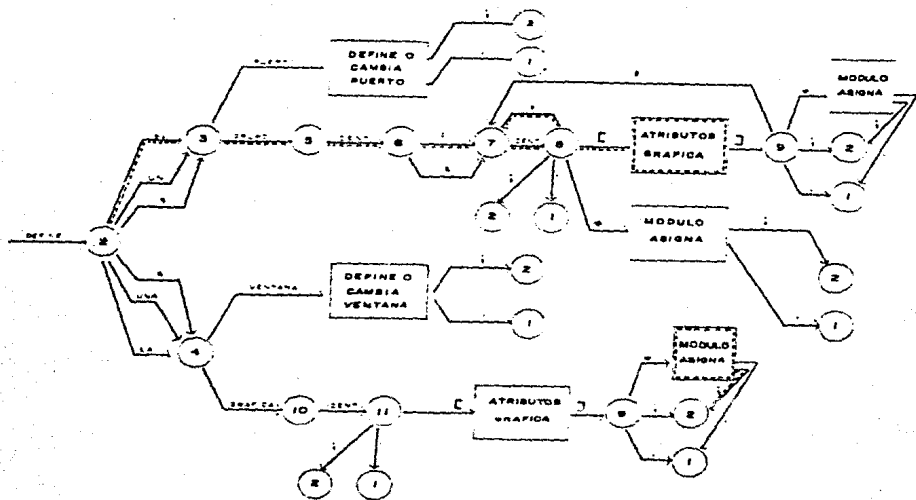


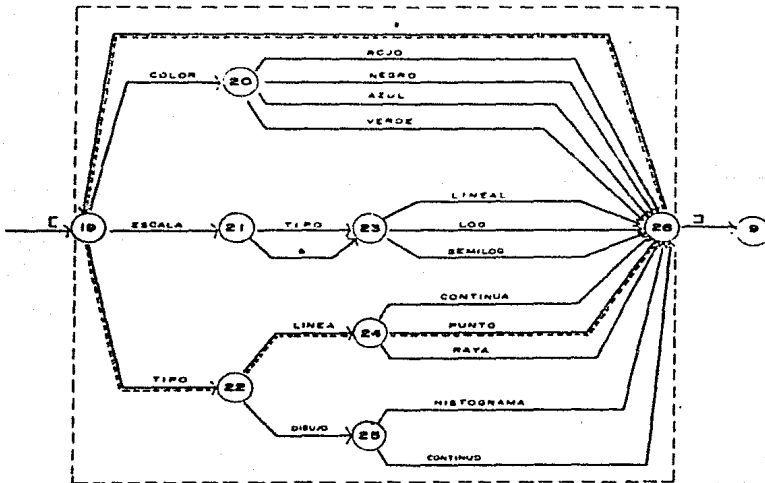
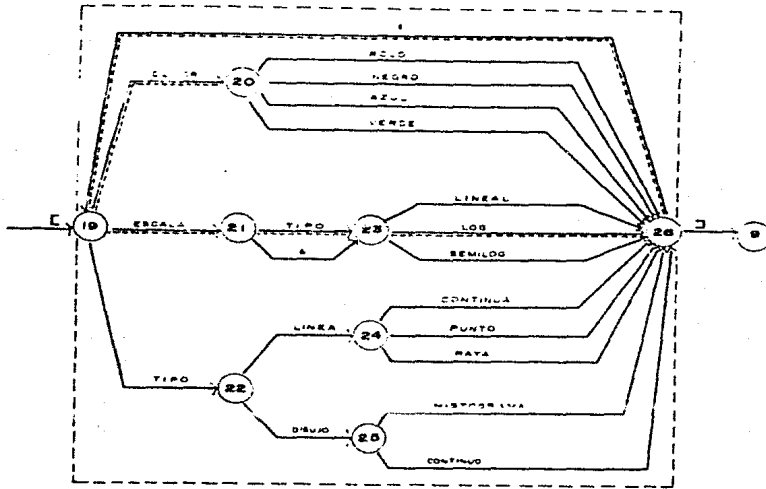




MODULO ASIGNA







→ DIBUJA LA GRAFICA <LET> (<LET> / <DIG>)*
<REP 3> <AGRAF>

→ DIBUJA LA GRAFICA GR1 <REP 3> <AGRAF>

→ DIBUJA LA GRAFICA GR1 <AGRAF>

→ DIBUJA LA GRAFICA GR1

→ DIBUJA LA GRAFICA GR1 ;

Por la misma razón que en el ejemplo 1 , al momento de sus -
tituir el símbolo NO-TERMINAL <COMPLE 3> por ; el control
se transfiere nuevamente a <COMPLE 3> , debido a ello, se
tiene:

→ DIBUJA LA GRAFICA GR1 ; <COMPLE 3>

→ DIBUJA LA GRAFICA GR1; <ARFEMS><DIGRAF>
<IDENT> <SIGNO> <IDENT> <REP 6> <VARIOS>

→ DIBUJA LA GRAFICA GR1; LA <DIGRAF>
<IDENT> <SIGNO> <IDENT> <REP 6> <VARIOS>

→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
<IDENT> <SIGNO> <IDENT> <REP 6> <VARIOS>

—————► DIBUJA LA GRAFICA GR1; LA DIFERENCIA
<LET> (<LET> / <DIG>) * <SIGNO> <IDENT> <REP 6> <VARIOS>

—————► DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 <SIGNO> <IDENT> <REP 6> <VARIOS>

—————► DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - <IDENT> <REP 6> <VARIOS>

—————► DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - <LET> (<LET> / <DIG>) * <REP 6> <VARIOS>

—————► DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 <REP 6> <VARIOS>

—————► DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 <VARIOS>

—————► DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3

—————► DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ;

—————► DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; <COMPLE 3>

—————> DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; <ARFENS> <DIGRAF> <IDENT> <SIGNO> <IDENT>
<REP 6> <VARIOS>

—————> DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA <DIGRAF> <IDENT> <SIGNO> <IDENT> <REP 6>
<VARIOS>

—————> DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA <IDENT> <SIGNO> <IDENT> <REP 6>
<VARIOS>

—————> DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA <LET> (<LET> / <DIG>)^{*} <SIGNO>
<IDENT> <REP 6> <VARIOS>

—————> DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 <SIGNO> <IDENT> <REP 6>
<VARIOS>

—————> DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + <IDENT> <REP 6> <VARIOS>

—————> DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + <LET> (<LET> / <DIG>)^{*}
<REP 6> <VARIOS>

—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4 <REP 6> <VARIOS>

—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4 <VARIOS>

—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4

—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4 ;

—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4 ; <COMPLE 3>

—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4 ; <ARMAS> <GRUPS>
<IDENT> <REP 3> <VARIOS>

—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4 ; EL <GRUPS> <IDENT>
<REP 3> <VARIOS>

—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4 ; EL GRUPO <IDENT>
<REP 3> <VARIOS>

—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4 ; EL GRUPO
<LET> (<LET> / <DIG>) * <REP 3> <VARIOS>

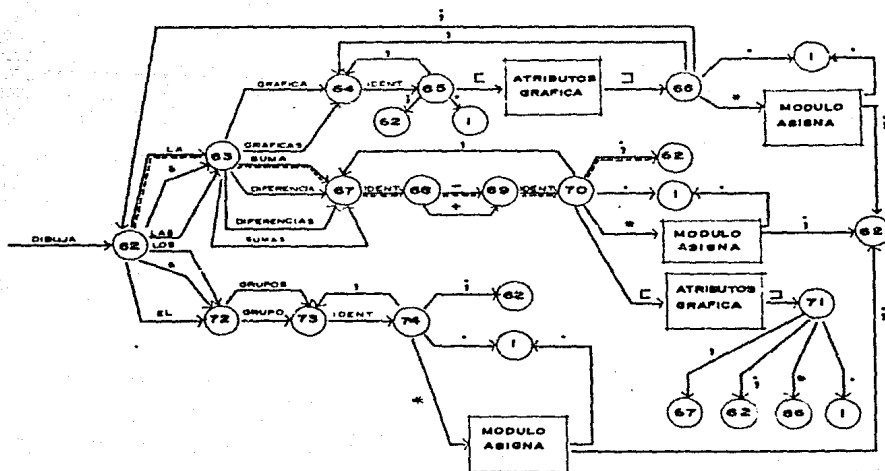
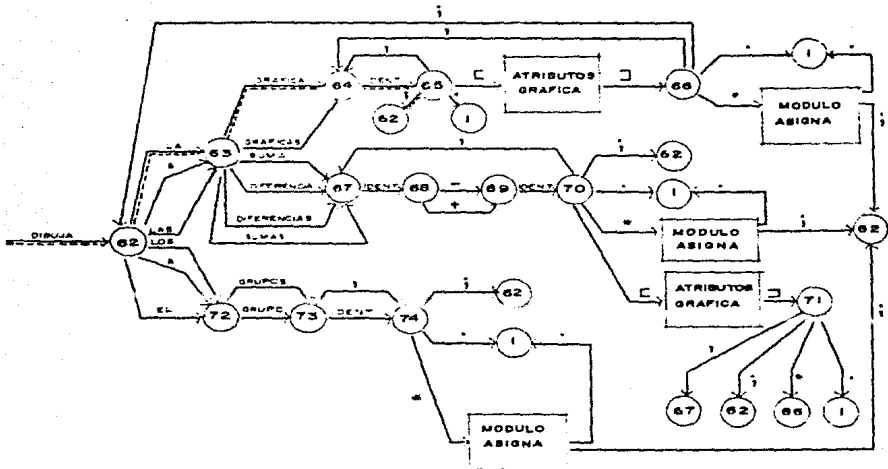
—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4 ; EL GRUPO
GRUP11 <REP 3> <VARIOS>

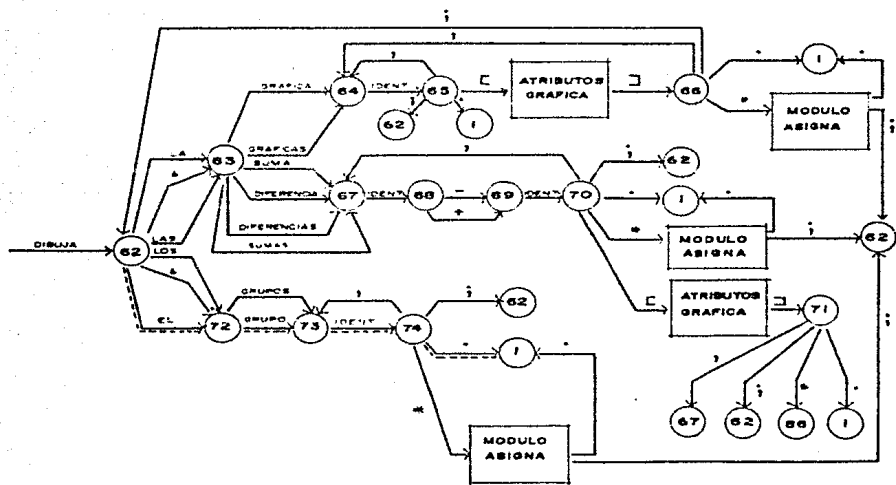
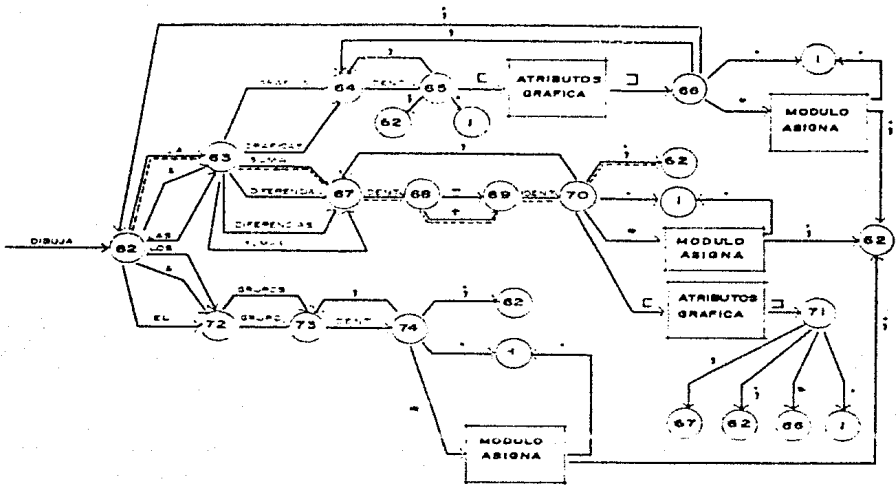
—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4 ; EL GRUPO
GRUP11 <VARIOS>

—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4 ; EL GRUPO
GRUP11

—————→ DIBUJA LA GRAFICA GR1; LA DIFERENCIA
GR2 - GR3 ; LA SUMA GR3 + GR4 ; EL GRUPO
GRUP11 .

MODULO DIBUJA





CONCLUSIONES

El haber realizado este trabajo, sin duda trae consigo el re cuerdo de las personas que deben utilizar lenguajes de diferente naturaleza para cumplir con su trabajo en un ambiente de computación; estos lenguajes pueden ser de programación , de graficación, de control de trabajos o procesos, etc.

La primer problemática con la que estas personas se enfrentan es con que dicho lenguaje está escrito en un idioma extranjero, por lo que es necesario primeramente aprender ese idioma (Inglés, Francés, etc.). El siguiente problema a resolver es que muchas veces el significado de las instrucciones o sea la semántica es oscura y difícil de entender en cuanto a la relación COMANDO-INTERPRETACION (¿ Qué quiere decir o qué dice ?). Este problema da origen al siguiente , el cual consiste en que al tener duda del significado de alguna instrucción o mensaje, es necesario buscar en varios ma nuales para entender realmente el significado.

El Lenguaje 'INTERGRAF' como se le bautizó al lenguaje desarrollado en este trabajo, trata de contribuir en el ambiente de la computación, ofreciendo a las personas que requieren - dibujar gráficas a través de un graficador, comúnmente llamado 'plotter', un Lenguaje claro, con sintaxis sencilla y sobre todo muy cercana a la sintaxis de nuestro Lenguaje el Es pañol.

El mensaje que se pretende proporcionar con este trabajo con siste en hacer conciencia de que dentro del ambiente de com-

putación en México, es posible diseñar e implantar Lenguajes interactivos que permitan al personal que utiliza recursos de cómputo llevar a cabo su trabajo de una manera más eficiente y eficaz al tener disponibles este tipo de Lenguajes - n sintaxis en Español e interactivos; no sólo para realizar gráficas, sin también para manejar procesos con cintas magnéticas, copiado de archivos entre dispositivos magnéticos, protección de archivos, controlar procesos de producción en una instalación de cómputo, etc.

Por otro lado, considero que para alguna persona que tenga la intención de diseñar e implantar algún tipo de Lenguaje como los expresados anteriormente, puede tomar como base la mayoría de los conceptos expuestos en este trabajo, principalmente :

- . La metodología a seguir en el análisis y el diseño de la interfaz hombre-máquina.
- . Los principios que es necesario cuidar en cuanto al desarrollo del lenguaje que pretenda ser conversacional.
- . El tipo de gramática que se pretenda diseñar y la forma de elaborar los diagramas de sintaxis, así como
- . La definición de los conjuntos de símbolos terminales y no terminales.

5.- BIBLIOGRAFIA

1. KELLOGG S. BOOTH
INTRODUCTION TO COMPUTER GRAPHICS
TUTORIAL: COMPUTER GRAPHICS
IEEE COMPUTER SOCIETY, INC. 1979.

2. ALFRED V. AHO / JEFFREY D. ULLMAN
PRINCIPLES OF COMPILER DESIGN
ED. ADDISON-WESLEY. 1979

3. DAVID GRIES
COMPILER CONSTRUCTION FOR DIGITAL COMPUTERS
ED. JOHN WILEY AND SONS, INC. 1971

4. NEWMAN / SPROULL
PRINCIPLES OF INTERACTIVE COMPUTER GRAPHICS
ED. MC. GRAW HILL. 1979

5. IAN HIRSCHSOHN; UNIVERSITY OF SAN DIEGO AT LA JOLLA
AMESPLOT-A HIGHER LEVEL DATA PLOTTING SOFTWARE SYSTEM
COMMUNICATIONS OF THE ACM. 1970

6. MARK GREEN, CSRG. UNIVERSITY OF TORONTO
A METHODOLOGY FOR THE SPECIFICATION OF
GRAPHICAL USER INTERFACE
TUTORIAL: COMPUTER GRAPHICS
IEEE COMPUTER SOCIETY, INC. 1981

7. JAMES D, FOLEY AND VICTOR L WALLACE
THE ART OF NATURAL GRAPHIC MAN-MACHINE CONVERSATION
TUTORIAL:COMPUTER GRAPHICS
IEEE COMPUTER SOCIETY INC. 1979

8. P.F. JONES
FOUR PRINCIPLES OF MAN-COMPUTER DIALOGUE
TUTORIAL:COMPUTER GRAPHICS
IEEE COMPUTER SOCIETY INC. 1979

9. HERBERT FREEMAN
INTERACTIVE COMPUTER GRAPHICS
ED, JOHN WILEY AND SONS INC. 1980

10. CARLOS RUIZ DE VELASCO G.
"HP/GRAF" PAQUETE DE RUTINAS BASICAS PARA
EL GRAFICADOR HP7203-A
COMUNICACIONES INTERNAS DEL I.I.M.A.S. UNAM. 1980

11. JOSE LUIS FARAH / CARLOS RUIZ DE VELASCO G.
"GRAFO" RUTINA DE GRAFICACION PARA EL
GRAFICADOR HP7203-A
COMUNICACIONES INTERNAS DEL I.I.M.A.S. UNAM. 1980