



9 24  
Universidad Nacional  
Autónoma de México

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
ARAGON

Diseño e Implantación de un Programa de  
Monitoreo Mediante Despliegues Gráficos,  
para una Planta Termoeléctrica

T E S I S  
QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA  
P R E S E N T A  
*Carlos Hernández López*

TESIS CON  
FALLA DE ORIGEN



MEXICO, D. F.

1991



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

<b>Capítulo uno</b>	<b>Requerimientos para la construcción de desplegados en la Interfaz hombre-máquina</b>	<b>1</b>
1.1	Interfaz hombre-máquina	3
1.2	Norma ISA-S5.5	4
1.3	Construcción de los desplegados de la Interfaz hombre-máquina	7
<b>Capítulo dos</b>	<b>Generador de despliegues</b>	<b>8</b>
2.1	Elementos de un sistema de despliegue "raster scan"	10
	Sistema de despliegue "raster scan"	10
	Monitores RGB	14
	Tubo de rayos catódicos	14
	Disposición de los cañones electrónicos	17
	Tipos de pantallas de color	18
2.2	Arquitectura de hardware de la Interfaz hombre-máquina	21
2.3	El equipo generador de despliegues	24
	Módulos de hardware del generador de despliegues	27
	Filosofía de operación del generador de despliegues	31

<b>Capítulo tres El manejador del equipo de despliegues</b>	<b>33</b>
3.1 Arquitectura de software de la Interfaz hombre-máquina	35
Manejadores	35
Programas de adquisición de datos	36
Programas de infraestructura	36
Funciones de presentación	37
3.2 Requerimientos y arquitectura del manejador del equipo de despliegues	39
3.3 Estructura de los mensajes entre el manejador del generador de despliegues y las funciones de presentación de la IHM	42
<b>Capítulo cuatro La norma gráfica GKS y la biblioteca de funciones gráficas residentes en el generador de despliegues</b>	<b>44</b>
4.1 La norma gráfica GKS	46
Primitivas	51
Sistemas de coordenadas y transformaciones	56
Funciones de entrada	61
Modos de operación de los dispositivos de entrada	62
Estaciones de trabajo	63
Segmentos	65
Niveles del GKS	65
Estados del GKS	67
4.2 El Graphical Kernel Firmware	69

<b>Capítulo Cinco</b>	<b>Diseño, programación e implantación del programa de supervisión mediante despliegues gráficos</b>	<b>72</b>
5.1	Especificaciones y requerimientos funcionales de DIAFLU	74
	Especificaciones	74
	Requerimientos funcionales	75
5.2	Biblioteca de dispositivos	78
	Rutinas de la Biblioteca de dispositivos	79
	Operación de la Biblioteca de dispositivos	80
5.3	Diseño y programación de Diagramas de flujo	85
	Los archivos de DIAFLU	85
	Generación de los archivos	85
	Arquitectura de DIAFLU	88
	Pseudocódigo de DIAFLU	90
	Operación de DIAFLU	101
	<b>Conclusiones</b>	<b>103</b>
	<b>Apéndice uno</b>	<b>105</b>
	<b>Apéndice dos</b>	<b>116</b>
	<b>Apéndice tres</b>	<b>125</b>
	<b>Bibliografía</b>	<b>130</b>

**I - Requerimientos para la construcción de desplegados en la  
interfaz hombre-máquina**

**Introducción**

**1.1 Interfaz hombre-máquina**

**1.2 Norma ISA-85.5**

**1.3 Construcción de los desplegados de la interfaz  
hombre-máquina**

## Introducción

El empleo de símbolos gráficos para representar elementos de un proceso industrial ha evolucionado rápidamente desde la década de 1970 y mediante el uso de computadoras se ha logrado desarrollar sistemas que supervisan procesos físicos mediante desplegados gráficos.

En este tipo de sistemas se usan distintos dispositivos de visualización para mostrar al usuario una representación gráfica del proceso que se está controlando. La simbología empleada para representar a los elementos que intervienen en el proceso es sumamente importante ya que es a través de ella como el usuario supervisa y controla una determinada operación.

Por otra parte, los símbolos gráficos que se utilizan para presentar información de un proceso dependen principalmente del diseñador y del usuario del producto, ya que generalmente están adecuados a la aplicación de que se trate y a los requisitos del usuario.

La necesidad de uniformizar los símbolos gráficos originó la creación de diversas normas para controlar y supervisar los procesos. En este capítulo se describe, en particular, la norma

ISA-S5.5, Graphic Symbols for Process Displays ("Símbolos gráficos para procesos de despliegue"), y los lineamientos que se siguieron para construir los diagramas que integran la interfaz hombre-máquina a través de la cual se supervisa la planta.

### 1.1 Interfaz hombre-máquina

El término interfaz hombre-máquina es el nombre colectivo que se otorga a todos los componentes perceptibles de una máquina que el ser humano utiliza para comunicarse con ella; éstos pueden ser medidores de aguja, pantallas, lámparas, etc.

La forma y estructura de la interfaz hombre-máquina depende básicamente del tipo de proceso que se lleva a cabo y del usuario del producto; en este trabajo, el proceso es la generación de energía en una planta termoeléctrica, y los usuarios son los operadores e ingenieros de la planta. Por lo anterior, el término interfaz hombre-máquina se emplea aquí para referirse únicamente a los desplegados gráficos que el operador de la planta termoeléctrica utiliza para observar su estado e interactuar con ella.



## 1.2 Norma ISA-85.5

La función principal de los símbolos gráficos es proveer al usuario de una representación fácilmente comprensible de los elementos que intervienen en su proceso a través de un dispositivo de despliegue. Existen diversas normas creadas para este fin; una de las cuales es la ISA-S5.5, Graphic Symbols for Process Displays, que establece los símbolos gráficos que deben utilizarse en dispositivos de despliegue, monocromáticos y de color, a fin de que puedan representar caracteres y figuras. Esta norma hace las siguientes recomendaciones para el uso de colores en la construcción de desplegados :

1- Se deben emplear combinaciones de color compatibles, es decir aquellas con alto radio de cromaticidad. Se deben evitar, por ejemplo, las siguientes combinaciones : rojo-verde, azul-amarillo y rojo-azul.

2- El número de colores en un desplegado se debe limitar al mínimo necesario. En la mayoría de los casos, no se necesitan más de 4 colores para transmitir información dinámica.

3- Las áreas grandes de fondo deben ser negras

4- Se debe utilizar el color de acuerdo con las convenciones usuales, por ejemplo, rojo para alto o peligro y verde para adelante o libre.

5- A fin de asegurar una respuesta rápida del operador, se deben escoger colores altamente saturados como rojo o amarillo.

6- Los colores no deben emplearse cualitativamente para indicar valor.

7- El diseñador de los desplegados debe establecer el significado de cada color antes de desarrollar una lista de los elementos asociados con cada color.

En la tabla 1-1 se muestra una lista de colores, su significado genérico y los elementos asociados con cada color según propone esta norma.

Tabla 1-1 Lista de colores y significados genéricos propuestos por la norma ISA-85.5

Color	Significado genérico	Elementos asociados
Negro	Fondo del desplegado	
Rojo	Emergencia	a) Alto b) Alarma de la más alta prioridad c) Cerrado
Amarillo	Precaución	a) Condición anormal b) Alarma de segunda prioridad
Verde	Seguro	a) Operación normal b) Inicio c) Abierto
Violeta	Estático e irrelevante	a) Equipo de proceso en servicio b) Mayoría de los letreros
Azul	No esencial	a) Equipo de proceso en "standby" b) Letreros , etiquetas
Magenta	Radiación	a) Alarma de radiación
Blanco	Datos dinámicos	a) Mediciones e información de estado b) Mensajes del sistema c) Tendencia d) Activación de un paso secuencial

### 1.3 Construcción de los desplegados de la interfaz hombre-máquina

Los elementos que intervienen en los desplegados que muestran la información de la planta, se diseñaron tratando de seguir la norma ISA-S5.5, aun cuando en general se dibujaron de acuerdo con su apariencia física real para que los operadores que deben interactuar con la interfaz hombre-máquina puedan identificarlos y asociarlos inmediatamente con el equipo al que representan.

El empleo de colores se apega casi totalmente a la norma ISA-S5.5; en aquellas variables cuyo valor puede cuantificarse, se utiliza el color rojo para los estados de alarma crítica, amarillo para alarma precrítica y azul para estado normal. En el caso de los iconos que representan a equipos se emplea rojo para equipo en operación y verde para equipo apagado o fuera de servicio.

Los diagramas se construyeron sobre un fondo negro y muestran una gran densidad de información aún cuando algunos autores sugieren utilizar como máximo el 20 % de la superficie total de despliegue disponible.

En el Apéndice uno se muestran algunos de los desplegados a través de los cuales se realiza la supervisión de la planta.

## **II Generador de despliegues**

### **Introducción**

- 2.1 Elementos de un sistema de despliegue "Raster scan"**
- 2.2 Arquitectura del hardware de la interfaz hombre-máquina**
- 2.3 El equipo generador de despliegues**

## **Introducción**

El equipo generador de despliegues permite la construcción y el despliegue de gráficas y diagramas que pueden modificarse dinámicamente para efectuar la supervisión de procesos industriales en tiempo real.

Los diagramas que se utilizan en la supervisión de la planta se dibujaron con el generador de despliegues, usando un editor gráfico instalado en la computadora host.

En este capítulo se describen brevemente los módulos de hardware que integran al generador de despliegues, su filosofía de operación y el sistema de despliegue que se usó para presentar en los monitores los diagramas generados.

## 2.1 Elementos del sistema de despliegue "raster scan"

Para desplegar una imagen primero se debe generar una lista de comandos gráficos que represente sus elementos. Estos serán creados a través de subrutinas que al ejecutarse generarán la información de cada pixel a desplegar.

Una vez ejecutados los comandos gráficos de la lista de despliegue, la información que genera cada comando debe convertirse a pixeles individuales. Esta conversión se realiza a través de un proceso conocido como **rasterizing** o **scan conversion**, tras el cual la información se envía a los monitores para desplegarla. A continuación se describe el sistema de despliegue **raster scan** y posteriormente los elementos principales de un monitor RGB.

### Sistema de despliegue raster scan

Los monitores que muestran la información de la planta utilizan el sistema de despliegue **raster scan**. El elemento principal en este sistema es el tubo de imagen, que es un tubo de rayos catódicos (TRC) con tres cañones electrónicos y una pantalla de fósforo dentro de una envoltura de vidrio vaciada de aire.

El cuello del TRC contiene los cañones electrónicos los cuales producen haces de electrones. La tensión anódica positiva acelera estos electrones hasta la pantalla . El ánodo es un recubrimiento conductor de la superficie interior de la envoltente de vidrio.

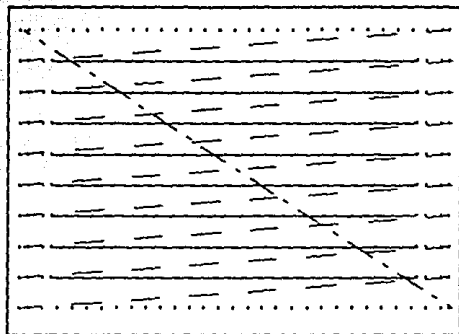
Para formar la pantalla, la superficie interior de la cara frontal de la envoltente de vidrio está recubierta con un material luminescente que produce luz cuando los electrones de los haces lo excitan.

En el sistema **raster scan** los haces recorren la pantalla de izquierda a derecha y de arriba hacia abajo. El período durante el cual regresan hacia la izquierda se conoce como "**retorno horizontal**". Durante la mayor parte de éste los cañones electrónicos deben estar deshabilitados para evitar que se escriba en un área de despliegue activa.

El período del retorno horizontal durante el cual los cañones están deshabilitados se conoce como **horizontal blanking**. La zona que circunda al área de despliegue, en la cual los haces se deben habilitar durante el retorno horizontal, se conoce como **borde u over scan**.

Los elementos que intervienen en el sistema **raster scan** se muestran en la figura 2-1.





- AREA DE DESPLIEGUE ACTIVA
- ..... OVER SCAN
- - - - - RETORNO HORIZONTAL
- - - - - RETORNO VERTICAL

FIG. 2-1 DIAGRAMA SIMPLIFICADO DE SCAN

Después de completar un barrido horizontal, los haces se mueven a la siguiente línea durante el retorno horizontal. Esta secuencia continúa hasta la última línea, al final de la cual se inicia el retorno vertical.

El retorno vertical es similar al horizontal; durante este período los cañones electrónicos están habilitados en un área de barrido pequeña y después se deshabilitan y regresan a la esquina superior izquierda de la pantalla. El período durante el cual los cañones están deshabilitados en el retorno vertical se conoce como vertical blanking.

La resolución vertical en el desplegado depende de las frecuencias del barrido horizontal y vertical. Una mayor frecuencia de barrido horizontal permite desplegar más líneas durante cada ciclo vertical; del mismo modo, un mayor intervalo vertical (menor frecuencia de barrido vertical) permite desplegar más líneas horizontales.

## Monitores RGB

Los monitores de despliegue RGB (red green blue) reciben las señales para los colores rojo, verde y azul en líneas separadas y la intensidad del color de cada señal está en función del voltaje en cada línea. A continuación se describen brevemente los componentes principales de un monitor RGB.

### Tubo de rayos catódicos (TRC)

El TRC de este tipo de monitor tiene tres cañones electrónicos, una pantalla con puntos o tiras verticales de fósforo rojo, verde y azul que se excitan con los haces generados por los cañones y una máscara de sombra o reguladora. La máscara es en realidad una placa metálica con ranuras verticales u orificios muy pequeños, los cuales permiten el paso de los electrones que excitan el fósforo de la pantalla. Cuando los tres cañones convergen debido a que los ángulos de sus trayectorias son correctos, cada haz excita su color respectivo sin incidir en los otros dos colores.

En el cuello del TRC se tienen montados el yugo de deflexión, el yugo de convergencia y el imán de pureza. El yugo de deflexión

desvía los tres haces de electrones, vertical y horizontalmente, para producir la trama de exploración, mientras que el yugo de convergencia obliga a que los haces converjan para producir colores precisos sin franjas coloreadas en los bordes de la imagen. El imán de pureza posiciona los tres haces para producir colores puros, sin que cada uno de los colores afecte a los otros.

La figura 2-2 muestra el esquema de un TRC con tres cañones electrónicos. Hay tres cátodos separados y tres rejillas de control individuales; la separación es necesaria para que se las señales que corresponden a cada color se puedan acoplar al circuito rejilla-cátodo a fin de modular la intensidad del haz de electrones.

Cada cañón tiene una rejilla de pantalla y una de enfoque separadas, aunque conectadas interiormente en paralelo para que la tensión de enfoque sea común.

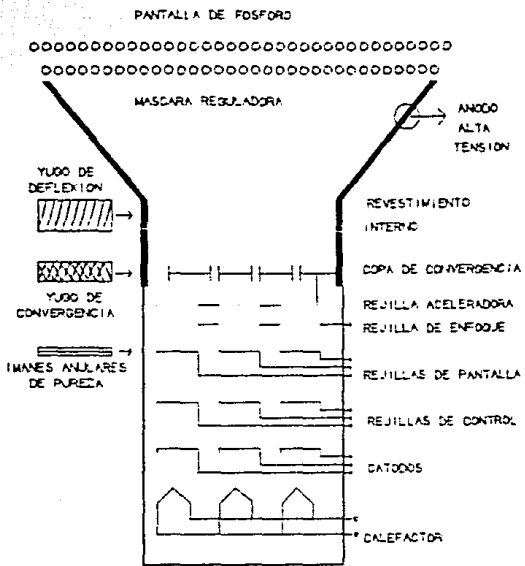
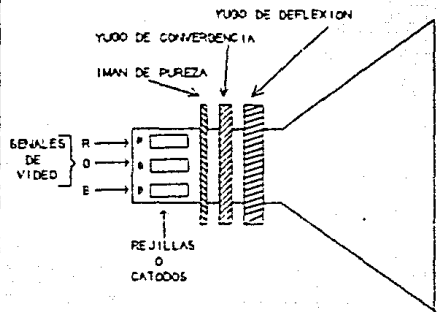


FIG. 2-2 DIAGRAMA DE TUBO DE IMAGEN EN COLOR CON TRES CANALES ELECTRONICOS. EL YUGO DE DEFLEXION, EL YUGO DE CONVERGENCIA Y LOS IMANES DE PUREZA RODEAN AL CUELLO DEL TUBO.

## Disposición de los cañones electrónicos

En la figura 2-3 se muestran las disposiciones más comunes de los cañones electrónicos en un monitor RGB. En la primera, los tres cañones están separados 120 grados en un círculo. A esta disposición se le conoce como "delta". En virtud de la separación uniforme, cada cañón puede tener el máximo diámetro posible en el cuello del TRC; generalmente los cañones están inclinados hacia el centro para favorecer la convergencia de los haces.

En la segunda disposición, los tres cañones se encuentran en un plano horizontal también orientados hacia el centro; en este arreglo "en línea", la convergencia de los haces es más fácil aunque cada cañón debe ser más pequeño comparado con el arreglo en "en delta".

En la tercera se muestra un arreglo poco frecuente en el cual se tienen tres cátodos para producir tres haces electrónicos, los cuales se enfocan y aceleran como un cañón electrónico común.

## **Tipos de pantallas de color**

El fósforo en los monitores de color puede estar arreglado en líneas o tríadas.

Una tríada está formada por puntos de fósforo rojo, verde y azul arreglados en forma de triángulo. Este tipo de pantalla se suele usar con cañones "en delta" y una máscara de sombra con orificios.

En los sistemas de líneas se usan tiras verticales de fósforo de color verde, rojo y azul arregladas horizontalmente ; los cañones presentan un arreglo "en línea" y la máscara de sombra muestra ranuras verticales. En ambos casos, tríadas y líneas, la máscara de sombra bloquea una gran cantidad de la corriente del haz electrónico ya que de otro modo serían imposibles la convergencia y la pureza del color.

En la figura 2-4 se muestran los arreglos "en delta" y "en línea".

La tríada es el arreglo más común, en este tipo de arreglo una máscara entre el fósforo y los cañones electrónicos permite que cada cañón ilumine sólo un color del fósforo cuando éstos se encuentran alineados correctamente.

El tamaño del hueco en la máscara se conoce como dot pitch y a menor tamaño de éste mayor calidad en el desplegado. Debe notarse que el tamaño real de los puntos de fósforo no está relacionado directamente con el tamaño del pixel, ya que un pixel puede estar formado por varias tríadas de fósforo.



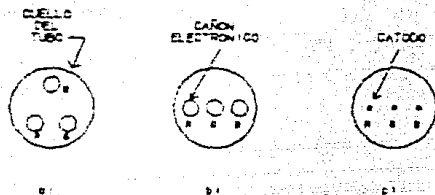


FIG. 2-3 METODOS PARA OBTENER TRES HACES ELECTRONICOS EN UN TUBO DE IMAGEN:  
 a) CAÑONES EN DELTA. b) CAÑONES EN LINEA. c) UN CAÑÓN CON TRES CÁTODOS EN LINEA.

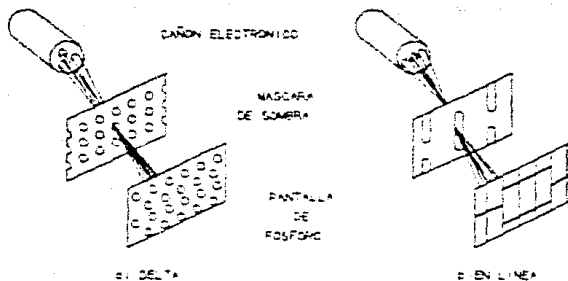


FIG. 2-4 TIPOS BASICOS DE TUBOS DE IMAGEN EN COLOR:  
 a) CAÑONES EN DELTA CON TRIADAS DE PUNTOS DE FOSFORO  
 b) CAÑONES EN LINEA CON MASCARA RANURADA PARA TIRAS DE FOSFORO

## 2.2 Arquitectura de hardware de la Interfaz hombre-máquina

El hardware de la interfaz hombre-máquina está constituido básicamente por : una computadora host o anfitrión con consola de operador y teclado, el equipo generador de despliegues, dos impresoras, tres graficadores , cuatro monitores de alta resolución y un teclado de uso rudo.

Los manejadores residentes en la computadora host se encargan de controlar al equipo generador de despliegues, a ambas impresoras y a los graficadores. El generador de despliegues controla los cuatro monitores.

Los módulos de adquisición de datos en campo se encuentran enlazados a la computadora host a través de un puerto serie, el cual opera a una velocidad de 19.2 Kbytes/s.

El generador de despliegues se encuentra enlazado a la computadora host a través de un puerto paralelo que opera a 500 Kbytes/s y se encarga de proveer los desplegados gráficos de salida a través de los monitores que se emplean para este fin.

Como se indicó anteriormente, el sistema cuenta con dos impresoras: una para la generación de reportes de eventos a

petición del usuario y otra para el registro de alarmas en tiempo real.

El sistema cuenta con tres graficadores para registrar el comportamiento de algunas variables importantes del proceso en tiempo real y para obtener gráficas del comportamiento histórico de estas variables en determinados lapsos de tiempo establecidos por el operador.

La interacción con los operadores e ingenieros de la planta se realiza a través del teclado de uso rudo, el cual en adelante se denominará "teclado del sistema" y se encuentra enlazado a la computadora host.

En la figura 2-5 se muestra la arquitectura del hardware de la IHM en la Central Termoeléctrica de Dos Bocas, Veracruz.

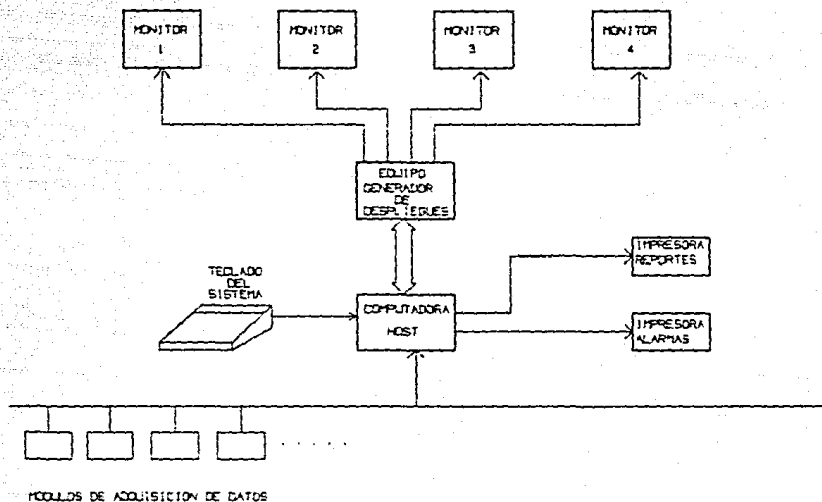


FIG. 2-5 ARQUITECTURA DE LA IHM DE DOS BOCAS. VER.

### 2.3 El equipo generador de despliegues

Como se mencionó, el generador de despliegues se encarga de proporcionar los desplegados que muestran el estado de planta, entre sus principales características se cuentan las siguientes :

- Posee dos puertos paralelos de E/S con una velocidad de hasta 500 Kbytes por segundo.
- Puede manejar hasta cuatro canales gráficos de salida.
- Puede manejar hasta 256 colores.
- Permite el manejo de cuatro monitores RGB con una resolución de 640 x 480 pixeles.
- Cuenta con un procesador bit-slice de alta velocidad para ejecutar los comandos de generación y almacenamiento de imágenes y el cálculo de los datos a desplegarse en cada pixel de los monitores.
- Puede manejar hasta ocho planos de memoria por cada canal gráfico.

En la figura 2-6 se muestra un diagrama a bloques del generador de despliegues y en la figura 2-7 un esquema de su arquitectura.

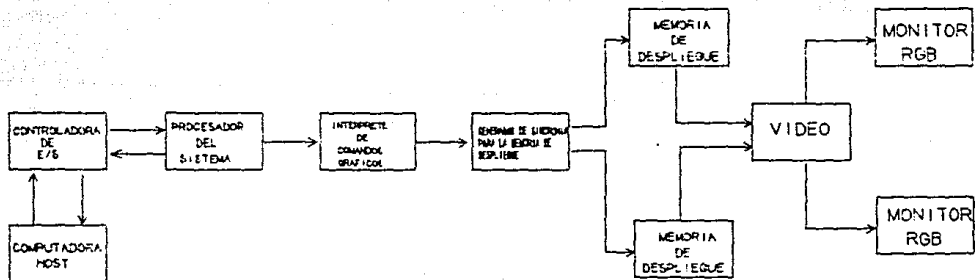


FIG.2-6 DIAGRAMA A BLOQUES DEL GENERADOR DE DESPLIEGUES

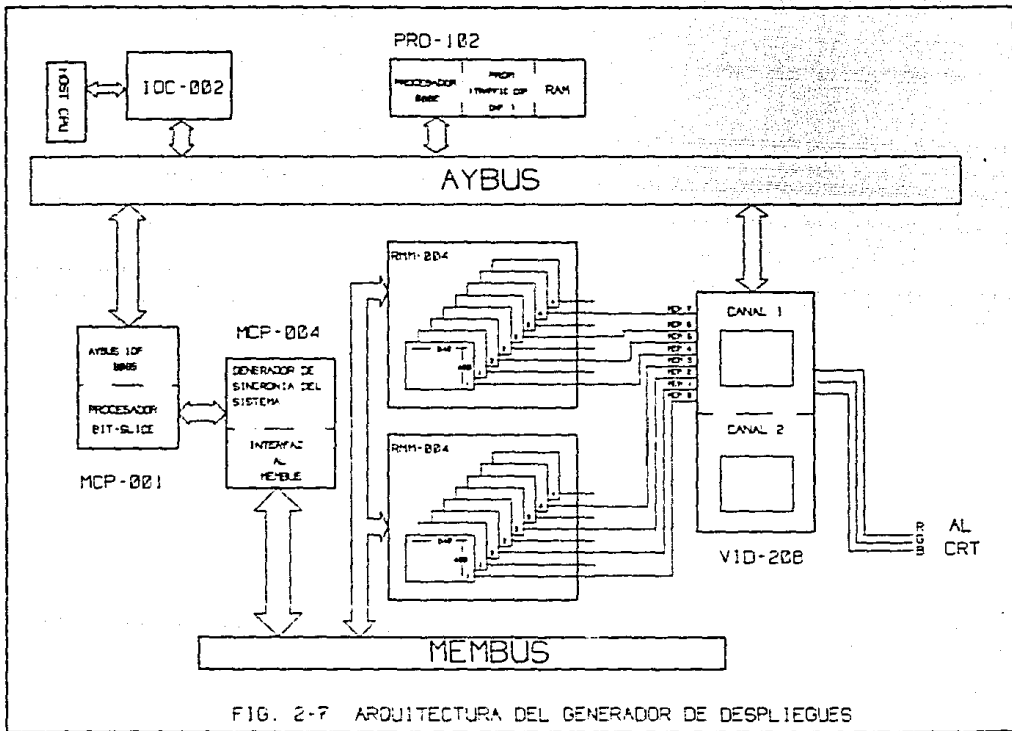


FIG. 2-7 ARQUITECTURA DEL GENERADOR DE DESPLIEGUES

## Módulos de hardware del generador de despliegues

El generador de despliegues consta de los siguientes módulos de hardware :

- Módulo procesador (PRO-102)
- Módulo controlador de E/S (IOC-002)
- Procesador de control de memoria (MCP-001)
- Procesador de control de memoria (MCP-004)
- Módulo de refresco de memoria (RMM-004)
- Módulo de video (VID-208)

A continuación se da una breve descripción de cada uno de los módulos de hardware que integran al generador de despliegues.

### MODULO PRO-102

El módulo PRO-102 es el sistema central de procesamiento de las señales de E/S del generador de despliegues, además de controlar los recursos de éste. Su función básica es interpretar los comandos y controlar las comunicaciones entre la computadora host y el generador de despliegues.



#### MODULO IOC-002

Este módulo controla la transferencia de datos entre la computadora host y el generador de despliegues. Es importante mencionar que cuando éstos dos se encuentran enlazados, el generador de despliegues es visto como un periférico por la computadora. El módulo IOC-002 está físicamente enlazado a la tarjeta DRV11WA de la computadora host.

#### MODULO MCP-001

Este módulo tiene dos funciones :

- Controlar el bus principal (AYBUS) del generador de despliegues.
- Generar los caracteres y procesar los comandos gráficos.

El módulo MCP-001 contiene dos microprocesadores que ejecutan tareas específicas. El primero de ellos se dedica a controlar el AYBUS, es decir, la transferencia de datos entre los módulos del generador de despliegues.

El segundo es un procesador bit-slice destinado a la ejecución gráfica mediante la generación de caracteres e imágenes para el

módulo de refresco de memoria RMM-004. Este procesador convierte los comandos gráficos, que envía la computadora host en patrones de bits que posteriormente el módulo MCP-004 transfiere a la tarjeta RMM-004 adicionándoles algunos parámetros de control.

El módulo MCP-001 puede operar independientemente o bajo el control directo del PRO-102. Como ya se mencionó, el MCP-001 ejecuta los comandos gráficos localizados en la memoria del bus principal del generador de despliegues. Estas instrucciones las puede haber generado el procesador del módulo PRO-102, o bien, la computadora host las puede haber cargado en memoria.

En la aplicación que trata este trabajo, la computadora host carga en memoria todas las instrucciones o comandos gráficos.

#### MODULO MCP-004

Este módulo desarrolla las siguientes funciones :

- Transfiere datos entre el módulo MCP-001 y el bus de refresco de memoria (MEMBUS).
- Provee todas las señales de control y sincronización para las operaciones del MEMBUS.
- Genera todas las señales de sincronización necesarias para

manejar los monitores conectados a los canales gráficos de salida del generador de despliegues.

El módulo MCP-004 recibe datos del MCP-001 a través de un bus interno y los transfiere al MEMBUS, el cual proporciona la conexión de los módulos de refresco de memoria (RMM-004) con sus dispositivos de control asociados.

#### MODULO RMM-004

El módulo de refresco de memoria RMM-004 provee la memoria de despliegue física para el generador de despliegues. Cada módulo RMM-004 tiene dos puertos de acceso de datos, uno de sólo lectura y el otro de lectura/escritura.

La salida para los módulos VID-208 se genera mediante accesos de lectura secuenciales al puerto de sólo lectura, para una posterior conversión paralelo-serie de los datos leídos en este puerto.

El puerto de lectura/escritura proporciona el acceso a fin de que el módulo MCP-004, a través del MEMBUS, transfiera los datos con el objeto de que el RMM-004 los escriba o lea.

## MODULO VID-208

Este módulo contiene los circuitos necesarios para manejar dos canales gráficos de video, cada uno de los cuales proporciona las señales de salida para un monitor RGB. Los módulos VID-208 reciben una trama de 8 señales TTL en serie, de un canal gráfico proveniente del módulo RMM-004 a fin de que a partir e estos datos se generen las señales analógicas de video para los monitores conectados a los canales gráficos.

Se recibe una trama de 8 señales debido a que el generador de despliegues tiene 8 planos de memoria y, por lo tanto, cada pixel que se va a desplegar tiene un valor determinado por aquellos planos de memoria en los que dicho pixel se ha utilizado. El valor de un pixel corresponde a una dirección en la tabla de colores (lookup-table) del generador de despliegues; el pixel se desplegará en el monitor con el color almacenado en esa dirección.

### Filosofía de operación del generador de despliegues

Una vez que el generador de despliegues se enlaza con la computadora host recibe una serie de comandos gráficos que ésta le envía a través de su módulo IOC-002.

Todas la tareas de interfaz y preprocesamiento entre los diversos módulos que integran al generador de despliegues se realizan sobre el bus principal o AYBUS. El procesador que controla el AYBUS genera listas de instrucciones de despliegue con los comandos que la host cargó en la memoria y las envía al módulo MCP-001 para ejecutarlas.

En el módulo MCP-001 se lleva a cabo el procesamiento de los comandos gráficos tanto para la generación de imágenes y su almacenamiento, como para la generación de señales de video.

Los datos que se obtienen de las transformaciones ejecutadas en el MCP-001 se transfieren al RMM-004 por medio del módulo MCP-004 a través del MEMBUS.

Los módulos RMM-004 convierten los datos recibidos del MCP-004 en una trama serie de 8 bits en niveles TTL y realizan la transferencia de estos datos hacia los módulos de video VID-208, los cuales, después de una conversión digital/analógica, transfieren las señales de video a los monitores RGB conectados al generador de despliegues en donde se muestra la información de los comandos gráficos que envía la computadora host.

### **III El manejador del equipo de despliegues**

#### **Introducción**

- 3.1 Arquitectura de software de la Interfaz Hombre-Máquina**
- 3.2 Requerimientos y arquitectura del manejador del equipo de despliegues**
- 3.3 Estructura de los mensajes entre el manejador del generador de despliegues y las funciones de presentación de la IHM**

## Introducción

En el capítulo anterior se ofreció un panorama general de la arquitectura de hardware de la Interfaz Hombre-Máquina (IHM), en este capítulo se da una semblanza de su arquitectura de software haciendo énfasis en el papel que desempeña el manejador del equipo generador de despliegues dentro de la IHM, su filosofía de operación y el formato de los mensajes que el resto de los módulos de la IHM envía a dicho manejador para que éste los envíe al generador de despliegues.

### **3.1 Arquitectura de software de la Interfaz Hombre-Máquina**

Todos los módulos de software que integran a la interfaz hombre-máquina operan en un ambiente de multitareas en tiempo real bajo el sistema operativo VAXELN, instalado en la computadora host.

Estos módulos se pueden agrupar de acuerdo con la función que desempeñan dentro del sistema en : manejadores, programas de adquisición de datos, programas de infraestructura y funciones de presentación.

#### **Manejadores**

Son los programas encargados de controlar a los puertos de entrada, salida y equipo periférico a la CPU de la computadora host.

Aunque el sistema operativo proporciona los manejadores tanto del disco y la consola del operador, como de algunas tarjetas de comunicación, a fin de desarrollar este proyecto se requirió modificar los manejadores originales de las tarjetas DRV11WA y DHV11 de modo que se pudiera lograr no sólo el control del



generador de despliegues y de las impresoras, sino también recibir información de los módulos de adquisición de datos en campo.

#### **Programas de adquisición de datos**

La IHM cuenta con un solo programa de adquisición de datos que se ejecuta asincrónicamente recibiendo información de los módulos de adquisición en campo, a través del software de comunicaciones del sistema, y transformándola a un formato que esté al alcance del resto de los módulos del software de la IHM.

Este módulo de adquisición también se encarga de reflejar el cambio de los datos adquiridos y procesados en la base de datos del sistema.

#### **Programas de infraestructura**

Los módulos de infraestructura son aquellos que, aun cuando no se encuentran en un nivel de software similar al de los manejadores y programas de adquisición de datos, sirven como interfaz entre éstos y las funciones de presentación. Dentro de estos módulos se puede mencionar a la base de datos y su

manejador, el administrador del teclado del sistema, el administrador de las impresoras, el módulo de alarmas y el encargado de llevar el registro histórico de las variables del sistema.

### **Funciones de presentación**

Las funciones de presentación son aquellos módulos de software encargados de mostrar al operador la información del proceso que se está llevando a cabo en la planta por medio de desplegados gráficos en los monitores del sistema, despliegues en consola, listados en las impresoras y gráficas en papel. El tema de este trabajo es el desarrollo de la función de presentación conocida como Diagramas de flujo (DIAFLU) de la cual se hablará con mayor detalle en el Capítulo V.

En la figura 3-1 se muestra el diagrama de flujo de datos de la interfaz hombre-máquina.

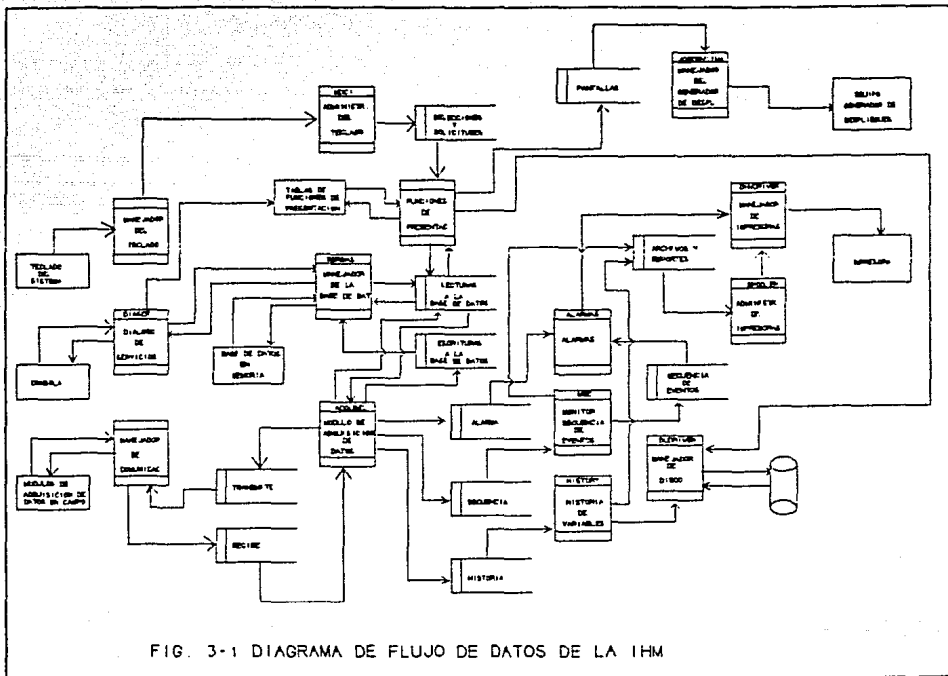


FIG. 3-1 DIAGRAMA DE FLUJO DE DATOS DE LA IHM

### 3.2 Requerimientos y arquitectura del manejador del equipo de despliegues

El manejador del generador de despliegues tiene como función básica controlar la tarjeta de comunicaciones DRV11WA de la computadora host para administrar los recursos de dicho equipo, por lo que debe desempeñar las siguientes tareas :

- Inicializar la tarjeta DRV11WA en la computadora host
- Transmitir los mensajes de las funciones de presentación hacia el generador de despliegues
- Recibir datos desde el teclado del sistema y hacer eco de ellos en el monitor que tenga el control del teclado.
- Desplegar el tiempo del sistema en los monitores activos
- Detectar y recuperarse de errores.

El manejador está formado por tres procesos, los cuales se ejecutan concurrentemente. El proceso principal está dedicado a administrar el envío de información proveniente de las funciones de presentación hacia el generador de despliegues. Existe un segundo proceso cuya función es desplegar la fecha y hora actuales del sistema en los monitores que estén activos; y el tercer

proceso está dedicado a atender al teclado del sistema, decodificando los códigos recibidos, haciendo eco de ellos en el monitor que tenga el control de éste y transfiriéndolos al módulo manejador del teclado del sistema, el cual se encarga de administrar y validar las selecciones y solicitudes.

En la figura 3-2 se muestra el diagrama de flujo de datos del manejador del equipo generador de despliegues y en el Apéndice 2 se proporciona su pseudocódigo.

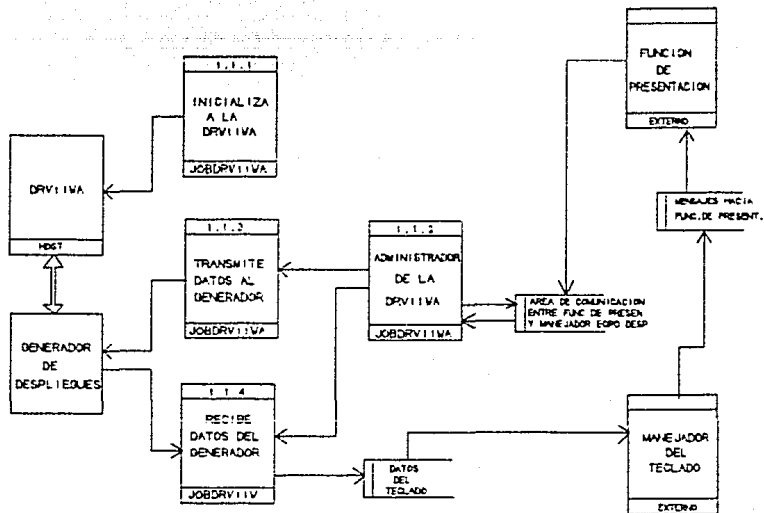


FIG.3-2 DIAGRAMA DE FLUJO DE DATOS DEL MANEJADOR DEL EQUIPO DE DESPLIEGUES

### 3.3 Estructura de los mensajes entre el manejador del generador de despliegues y las funciones de presentación de la IHM

Todos los mensajes de las funciones de presentación hacia el manejador del generador de despliegues se realizan empleando un arreglo de palabras de 16 bits con el siguiente formato:

Palabra de conteo  
Palabra de encabezado y direccionamiento  
Mensaje  
Palabra de fin de mensaje

Los dos primeros bytes del mensaje forman la palabra de conteo. En ella se envía el número total de bytes que se desea transmitir al generador de despliegues; este número debe incluir a los dos bytes de la palabra de conteo, los dos de la de encabezado y direccionamiento, el número de bytes que ocupe el cuerpo del mensaje que se va transmitir y los dos bytes de la palabra de fin de mensaje.

La palabra de encabezado y direccionamiento está formada por el tercer y cuarto byte del mensaje. El tercer byte del mensaje es el de encabezado tiene asignado un valor de 01h. El cuarto byte es el de direccionamiento y determina si el mensaje se va a enviar al procesador gráfico GKF o bien al coordinador de mensajes del generador de despliegues.

En la aplicación que se trata en este trabajo todos los mensajes que recibe por el manejador del generador de despliegues están direccionados hacia el procesador gráfico GKF asignándole a este byte un valor de 10h. Por lo tanto la palabra de encabezado y direccionamiento tiene un valor de 1001h.

En el cuerpo del mensaje se envían todos los comandos GKF que se requieran para dibujar o modificar una determinada imagen. La longitud mínima del mensaje es de 5 bytes y la máxima de 64 Kbytes.

Todas las transmisiones hacia el manejador del generador de despliegues deben terminar con un código de fin de mensaje que es una palabra con valor de 0003h.

En cualquier transferencia del manejador del generador de despliegues hacia este equipo, una rutina de servicio de interrupción lee automáticamente las dos primeras palabras del mensaje; después que éstas se han recibido, al coordinador de mensajes del generador de despliegues se le avisa que la computadora host está enviando datos y con la palabra de conteo y la dirección del canal de destino se abre el canal apropiado y se conduce el mensaje.



## **IV La norma gráfica GKS y la biblioteca de funciones gráficas residente en el equipo generador de despliegues**

### **Introducción**

**4.1 La norma gráfica GKS**

**4.2 El Graphical Kernel Firmware (GKF)**

## Introducción

La estandarización de las gráficas generadas por computadora es una actividad que el American National Standards Institute (ANSI) y la International Standards Organization (ISO) empezaron a desarrollar desde 1970 con el fin de alcanzar los siguientes objetivos :

- a) Permitir a los programas de aplicación relacionados con gráficas ser fácilmente transportables entre diferentes instalaciones.
- b) Ayudar a quienes generan programas de aplicación en la comprensión y el uso de los métodos gráficos .
- c) Servir a los fabricantes de equipo gráfico como guía en la selección de las capacidades gráficas más usuales de un dispositivo.

Así, en la actualidad existen diversas normas gráficas: Graphical Kernel System (Sistema gráfico de núcleo o GKS), Color Graphics Virtual Device Interface (Gráficas en color con interfaz virtual del dispositivo o CGVDI), Programmer's Hierarchical Interactive Graphics Standard (Norma de gráficas interactivas jerárquicas del programador o PHIGS), X Windows, etcétera.

En este capítulo se ofrece un panorama general de la norma gráfica GKS y la implantación en firmware de esta norma que reside en el equipo generador de despliegues.

#### 4.1 La norma gráfica GKS

El Sistema gráfico de núcleo (Graphical Kernel System) o GKS es una norma aceptada por la ANSI y la ISO para generar gráficos en dos dimensiones el cual establece un modelo para la conceptualización y especificación de operaciones gráficas por computadora.

El GKS se define independientemente de un lenguaje de programación en particular, de algún tipo de hardware específico y de un sistema operativo determinado. Estas características hacen que las aplicaciones basadas en GKS sean fácilmente transportables a cualquier sistema compatible con esta norma.

El GKS provee una interfaz funcional entre un programa de aplicación y la configuración de los dispositivos de entrada y salida en un sistema gráfico. Esta interfaz está formada por una biblioteca de funciones que desarrollan tareas gráficas que pueden utilizar los programadores de aplicaciones, de modo que sus

programas puedan producir y manipular imágenes. Las aplicaciones van desde la generación de imágenes pasivas hasta aplicaciones en tiempo real.

El GKS es el núcleo de un sistema gráfico, el cual es independiente del lenguaje de programación en el que se desarrollen las aplicaciones; por lo tanto, para que la aplicación tenga acceso a las funciones del GKS debe existir una capa de software que sirva de enlace entre la aplicación y el GKS; esta capa se conoce comúnmente como Lenguaje "Binding".

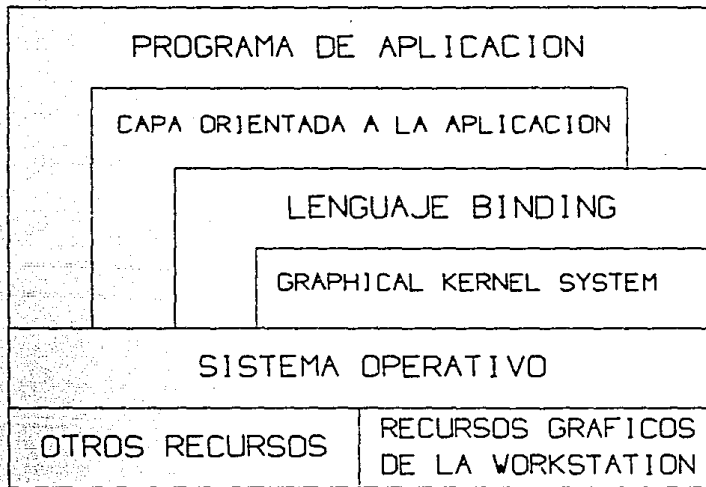


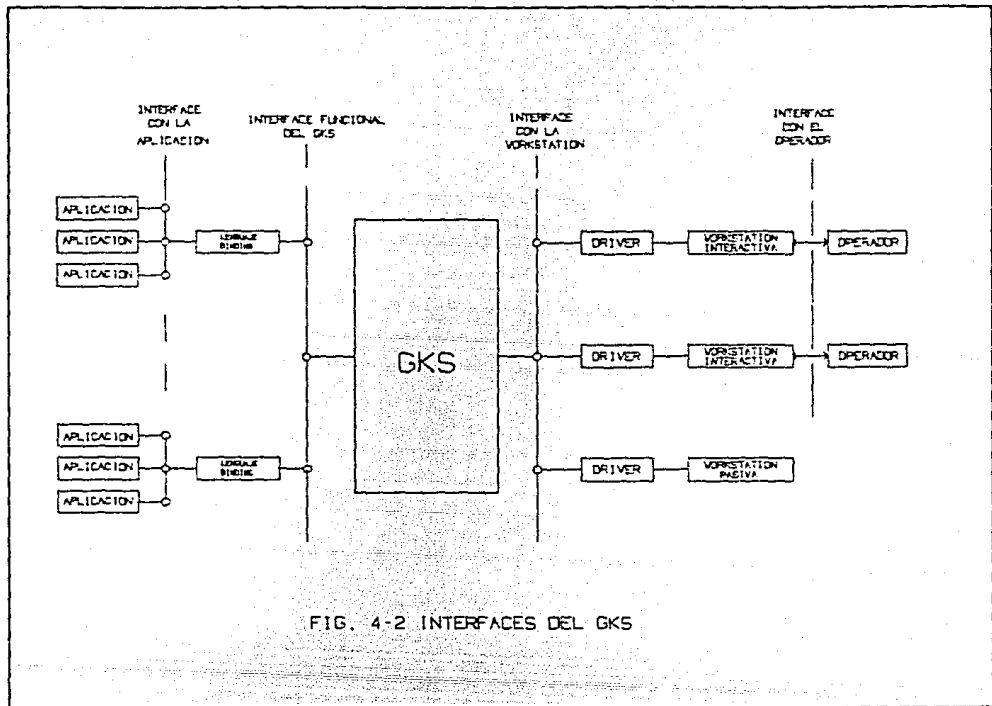
FIG. 4-1 MODELO DE CAPAS DEL GKS

El modelo representado en la figura 4-1 muestra el papel del GKS en un sistema gráfico. De acuerdo con la figura cada capa puede llamar a las funciones de sus capas inferiores. En general el programa de aplicación usa tanto la capa orientada a la aplicación, como la dependiente del lenguaje o Lenguaje "Binding" y los recursos del sistema operativo.

El GKS es el núcleo de un sistema gráfico no sólo en el sentido de que proporciona funciones gráficas para diferentes aplicaciones independientemente de un lenguaje de programación específico, sino también porque estas funciones son independientes de dispositivos gráficos determinados. Por ello existe una importante interfaz entre el núcleo del sistema y los dispositivos gráficos unidos a él.

En el GKS, el dispositivo de salida gráfica y los dispositivos de entrada conectados a éste se conocen como Estación de trabajo (Workstation), concepto que se explica más adelante.

La figura 4-2 muestra al GKS como un núcleo entre la interfaz con la aplicación y la interfaz con la estación de trabajo.



A continuación se explican brevemente los principales conceptos que maneja el GKS.

### **Primitivas**

La salida gráfica que genera el GKS está constituida por dos grupos de elementos básicos : primitivas de salida y sus atributos.

Las primeras son elementos básicos de dibujo que pueden utilizarse para construir una imagen. El GKS proporciona seis primitivas de salida : polilínea, polimarcador, texto, area llena, arreglo de celdas y primitivas de uso generalizado.

**Polilínea (Polyline)** : genera un conjunto de líneas rectas que conectan a una secuencia de puntos dada.

**Polimarcador (Polymarker)** : da origen a cierto tipo de símbolos centrados en una posición dada, a los cuales generalmente se les llama marcadores.

**Texto (Text)** : produce una cadena de caracteres en una posición determinada.

**Area llena (Fill Area)** : dibuja un polígono el cual puede estar vacío, o lleno con un color uniforme o un patrón determinado.

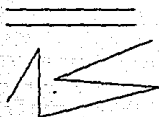


**Arreglo de Celdas (Cell Array) :** genera un arreglo de celdas rectangulares con colores individuales.

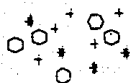
**Primitiva de uso generalizado (Generalized Drawing Primitive o GDP) :** es una primitiva con la cual el GKS direcciona características geométricas especiales de salida de una estación de trabajo, tales como dibujar círculos, rectángulos, líneas curvas y arcos circulares o elípticos.

En la figura 4-3 se muestran ejemplos de las diferentes primitivas de salida.

POLILINEAS



POLIMARCADORES



TEXTO

este es un TEXTO  
otro TEXTO

*un texto más*

AREA LLENA



ARREGLO DE CELDAS



GDP

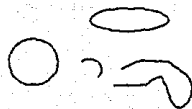


FIG. 4-3 EJEMPLOS DE PRIMITIVAS PROPORCIONADAS  
POR EL GKS

Cada primitiva tiene 3 tipos básicos de atributos : geométricos, no geométricos y de identificación. Los dos primeros determinan la apariencia exacta de la primitiva de salida, mientras que el tercero se utiliza para relacionarla con una entrada.

Los atributos geométricos de la primitiva controlan los aspectos que afectan su forma y tamaño. Los no geométricos se refieren a los aspectos que tienen que ver únicamente con su apariencia, por ejemplo el tipo de línea y el color para una polilínea.

Un caso especial es la GDP la cual no tiene atributos geométricos explícitos y su aspecto está controlado por cero o más conjuntos de atributos de polilíneas, polimarcadores, textos y áreas llenas.

Uno de los atributos de identificación es el "pick identifier", éste se usa para reconocer a una primitiva o a un grupo de primitivas en un segmento seleccionado. Un segmento es un conjunto de primitivas de salida (y sus atributos asociados) dispuestas en un grupo, este concepto se explica con mayor profundidad posteriormente.

En la tabla 4-1 se muestra el conjunto de atributos que controlan la apariencia de cada una de las primitivas de salida.

PRIMITIVA	ATRIBUTOS	
POLILINEA	ANCHO DE LINEA TIPO DE LINEA	PICK IDENTIFIER COLOR
POLIMARCADOR	TAMAÑO DEL MARCADOR TIPO DEL MARCADOR	PICK IDENTIFIER COLOR
TEXTO	FONT PRECISION ALINEAMIENTO ESPACIAMIENTO TRAYECTORIA	PICK IDENTIFIER ALTURA DE CARACTER EXPANSION DEL CARACTER DIRECCION DEL CARACTER COLOR
AREA LLENA	TAMAÑO DEL PATRON PUNTO DE REFERENCIA DEL PATRON ARREGLO DEL PATRON	PICK IDENTIFIER ESTILO INTERIOR ESTILO DE SOMBREADO COLOR
ARREGLO DE CELDAS	COLOR	PICK IDENTIFIER
GDP	PICK IDENTIFIER DEPENDE DEL TIPO DE GDP	

TABLA 4-1 PRIMITIVAS DE SALIDA PROPORCIONADAS  
POR EL GKS Y ATRIBUTOS DE ESTAS

## Sistemas de coordenadas y transformaciones

En el GKS se encuentran definidos los tres sistemas de coordenadas siguientes :

- **Coordenadas Mundiales (World Coordinates o WC)**
- **Coordenadas Normalizadas de Dispositivo (Normalized Device Coordinates o NCD)**
- **Coordenadas de Dispositivo (Device Coordinates o DC)**

El programador de aplicaciones usa el sistema de **Coordenadas Mundiales**, en él se definen los elementos de una imagen independientemente de los dispositivos gráficos de entrada y salida de la estación de trabajo, en un espacio de dibujo virtualmente infinito.

Cada dispositivo de salida posee un sistema de coordenadas particular por lo que GKS, a fin de mantener su independencia de los dispositivos, define o normaliza un espacio para el sistema de coordenadas que utilizan éstos.

Para obtener la posición relativa de una primitiva de salida, se mapea una porción del sistema de **Coordenadas Mundiales** en un

sistema de **Coordenadas Normalizadas de Dispositivo** independiente del dispositivo gráfico de salida de la estación de trabajo pero normalizado a un cierto rango.

Conceptualmente, el sistema de **Coordenadas Normalizadas de Dispositivo** es un espacio que se extiende hasta el infinito, pero su porción visible en la estación de trabajo se localiza en el rango  $[0, 1] \times [0, 1]$ .

El sistema de **Coordenadas de Dispositivo** es totalmente dependiente del dispositivo de salida de la estación de trabajo, el sistema **Coordenadas Normalizadas de Dispositivo** se mapea en este espacio

El mapeo del sistema de **Coordenadas Mundiales** hasta el sistema de **Coordenadas de Dispositivo** se realiza mediante una transformación de normalización y una transformación de estación de trabajo. En la primera se define una **Ventana (Window)**, o sea los límites de un área en el sistema de **Coordenadas Mundiales**, que se mapea en un **Puerto de Visión (Viewport)**, es decir en un área específica del sistema de **Coordenadas Normalizadas de Dispositivo**; en la segunda el **Puerto de Visión** se mapea a la superficie de despliegue de la estación de trabajo.

La figura 4-4 esquematiza los sistemas de coordenadas del GKS y las transformaciones necesarias para pasar de un sistema a otro. La figura 4-5 muestra la relación entre los sistemas de coordenadas del GKS.

PROGRAMA DE APLICACION

IMAGEN NORMALIZADA

SUPERFICIE DE DESPLIEGUE

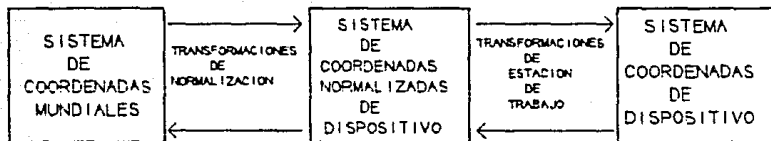
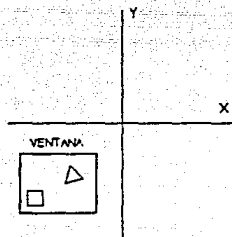


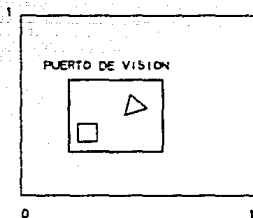
FIG. 4-4 SISTEMAS DE COORDENADAS EN EL GKS Y TRANSFORMACIONES EFECTUADAS PARA PASAR DE UN SISTEMA A OTRO.



COORDENADAS MUNDIALES



COORDENADAS NORMALIZADAS



SUPERFICIE DE DESPLIEGUE

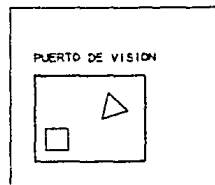


FIG. 4-5 RELACION ENTRE LOS SISTEMAS DE COORDENADAS DEL GKS

## Funciones de entrada

El GKS contiene funciones que permiten introducir valores desde diversas clases de dispositivos de entrada configurados en diferentes modos de operación. Las funciones de entrada que GKS proporciona son :

**Localización (Locator)** : provee una posición en coordenadas mundiales. El operador proporciona la posición al desplazar un dispositivo de localización (por ejemplo un "mouse" ).

**Trazo (Stroke)** : suministra una secuencia de posiciones en coordenadas mundiales. El operador genera las posiciones al colocar un dispositivo de localización en diferentes ubicaciones.

**Valuación (Valuator)** : manda el valor de un número real, generalmente desde el teclado.

**Opción (Choice)** : proporciona un valor entero no negativo, el cual representa una selección entre varias opciones (por ejemplo al presionar uno de los botones de un "mouse").

**Selección (Pick)** : provee el nombre de un segmento y su "pick identifier". El operador selecciona un segmento al reconocer una parte de la imagen que el sistema le está desplegando.

**Cadena (String)** : envía una cadena de caracteres, generalmente desde el teclado.

#### **Modos de operación de los dispositivos de entrada**

Cada dispositivo de entrada se puede usar en uno de tres distintos modos de operación, el cual se selecciona basandose en la aplicación. Se puede utilizar sólo un modo a la vez y , dependiendo de cuál se seleccione, el operador introduce los valores de entrada los cuales pasan a la aplicación de diferentes maneras. Los modos de operación son los siguientes :

**Solicitud (Request)** : La invocación de una función de entrada para un dispositivo que opera en este modo origina que el GKS intente leer un valor de este dispositivo esperando hasta que el operador introduzca la entrada o bien desarrolle una acción de cancelación.

**Muestreo (Sample)** : La invocación de una función de entrada de un dispositivo que se encuentre en el modo de muestreo origina que el GKS, sin esperar una acción del operador, regrese el valor lógico de la entrada actual del dispositivo especificado.

**Evento (Event)** : GKS mantiene una cola que contiene reportes de eventos, los cuales a su vez contienen la identificación del dispositivo de entrada y el valor de la entrada lógica de ese dispositivo. Los reportes de eventos se generan asincrónicamente, sólo por la acción del operador.

## Estaciones de Trabajo (Workstations)

Un concepto fundamental del GKS es la estación de trabajo, la cual se define como uno o más dispositivos de entrada enlazados a un solo dispositivo de salida.

El GKS se basa en el concepto de estaciones de trabajo gráficas abstractas, con ello se proporciona una interfaz lógica a través de la cual la aplicación controla a los dispositivos físicos.

Una estación de trabajo gráfica abstracta con capacidades máximas :

- Tiene una superficie de despliegue direccionable
- Permite sólo espacios de despliegue rectangulares
- Soporta diversos tipos tanto de líneas, como de caracteres y tamaños de texto y otras primitivas a fin de permitir que las primitivas se dibujen con diferente aspecto.
- Tiene uno o más dispositivos de entrada para cada clase de entrada en distintos modos de operación.
- Almacena segmentos y provee facilidades para su manipulación y cambio.

Cualquier estación de trabajo cae dentro de las seis categorías siguientes :

- Salida (Output)
- Entrada (Input)
- Salida y entrada (Outin)
- Estación de trabajo con almacenamiento independiente de segmentos (Workstation independent Segment Storage o Wiss)
- Metaarchivo GKS de salida (GKS Metafile output o MO)
- Metaarchivo GKS de entrada (GKS Metafile input o MI)

**Salida :** Posee sólo capacidades de salida. Puede desplegar todas las primitivas, a excepción de GDP la cual es opcional.

**Entrada :** Tiene al menos un dispositivo de entrada, pero no cuenta con capacidad de salida.

**Entrada y salida :** Posee las características de las estaciones de trabajo de entrada y salida.

**Wiss, Mo y Mi :** Son facilidades del GKS que proporcionan los medios para el almacenamiento permanente o temporal de información gráfica.

## **Segmentos**

Las primitivas de salida y sus atributos pueden agruparse en un segmento. Los segmentos son unidades para la manipulación y el cambio. La primera abarca la creación, borrado y cambio de nombre del segmento. El cambio incluye la transformación de un segmento, esto es, hacerlo visible o invisible, o bien dejarlo intermitente.

Los segmentos se distinguen por un nombre único : "identificador del segmento". Todas las primitivas de salida se juntan en un segmento después de que éste se ha creado y antes de que se cierre. Una vez cerrado, ninguna primitiva puede adicionarse o borrarse del segmento.

## **Niveles del GKS**

Las funciones del GKS están organizadas en nueve niveles. Una implantación de GKS provee precisamente las funciones de uno de esos niveles. Mientras el nivel más bajo contiene sólo un conjunto mínimo de funciones, en los niveles más altos se permite extenderse más allá de las necesidades básicas. En la tabla 4-2 se muestran los niveles del GKS y las aplicaciones usuales de cada nivel.

NIVEL	APLICACIONES
0a	GRAFICAS COMERCIALES: DRIVERS DE BAJO NIVEL DE SALIDA PARA SISTEMAS GRAFICOS MAYORES
0b	GRAFICAS COMERCIALES INTERACTIVAS: CONTROL DE PROCESOS: SISTEMAS SENCILLOS DE GENERACION DE IMAGENES: PROVEER DE ENTRADAS A SISTEMAS GRAFICOS MAYORES
0c	SISTEMAS BASADOS EN ENTRADAS CON FUERTES REQUERIMIENTOS DE INTERACCION. COMO APLICACIONES DIGITALIZADAS
1a 1b 1c	SISTEMAS DE MULTIPLES WORKSTATIONS. DESPLIEGUE DE VARIAS IMAGENES SIMULTANEAMENTE. INTERACCION CON PARTE DE LAS IMAGENES DESPLEGADAS Y MANIPULACION DE ESTAS. DISEÑO DE APLICACIONES Y SISTEMAS DE SIMULACION
2a 2b 2c	DISEÑO Y BOSQUEJO DE SISTEMAS CON USO DE LAS MAXIMAS FACILIDADES PROVISTAS POR EL GKS. USO DE SEGMENTOS YA CREADOS PARA LA CONSTRUCCION DE IMAGENES. COPIADO DEL CONTENIDO DE UNA WORKSTATION A OTRA. EDITORES GRAFICOS. SISTEMAS CAD

TABLA 4-2 NIVELES DEL GKS-Y APLICACIONES USUALES DE CADA NIVEL

## Estados del GKS

Durante la ejecución de un programa que emplee GKS, éste se puede encontrar en uno de cinco diferentes estados operativos :

GKCL	GKS cerrado (GKS closed)
GKOP	GKS abierto (GKS open)
WSOP	Al menos una estación de trabajo abierta (At least one workstation open)
WSAC	Al menos una estación de trabajo activa (At least one workstation active)
BGOP	Segmento abierto (Segment open)

Las funciones de control llamadas por la aplicación causan la transición de un estado a otro, los accesos a funciones específicas del GKS se permiten o no dependiendo del estado operativo. La figura 4-6 muestra las transiciones de estado del GKS, y las funciones permitidas en cada estado.



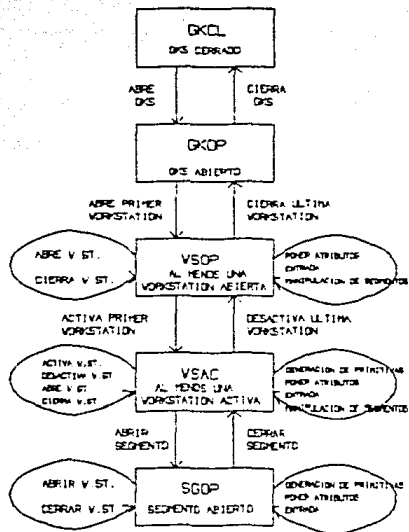


FIG. 4-6 TRANSICIONES ENTRE LOS ESTADOS OPERATIVOS DEL GKS Y FUNCIONES PERMITIDAS EN CADA ESTADO

#### 4.2 El Graphical Kernel Firmware (GKF)

El Graphical Kernel Firmware (GKF) es una implantación en firmware del GKS que reside en el módulo PRO-102 del equipo generador de despliegues. Cabe hacer notar que esta implantación proporciona la mayoría de las funciones permitidas en el nivel 10 del GKS.

El GKF está formado por un conjunto de comandos de alto nivel que direccionan y ejecutan todas las funciones gráficas del sistema. A través de este grupo de comandos se dibujan las primitivas que forman las imágenes manipuladas en tiempo real por la aplicación que trata este trabajo.

Los comandos provistos por el GKF pueden agruparse en :

- Comandos para estación de trabajo
- Comandos para primitivas de salida
- Comandos para atributos de primitiva
- Comandos para segmentos
- Comandos para funciones de E/S
- Comandos para operaciones de rastreo

Cada comando es una palabra de 16 bits.

Estos grupos de comandos se diferencian entre sí por los valores asignados a los bits 15, 14 y 13 de cada comando. El bit 12 no se utiliza para diferenciar a un grupo de comandos de otro sino para establecer la posición en que se ejecutará un comando gráfico específico. Los 12 bits restantes son únicos para cada comando del GKF.

El grupo de comandos para primitivas de salida se caracteriza por tener los bits 15, 14 y 13 puestos en "0". El bit 12 se pone en "0" para obtener la posición actual (current position o CP) por omisión y puesto a "1" para definir una CP específica. En este grupo se encuentran los comandos para dibujar las siguientes primitivas :

Polilínea

Polimarcador

Texto

Area Llana

Círculo

Arco de círculo

Elipse

Arco de Elipse

Polígono regular

Polígono irregular

Triángulo

Rectángulo

El grupo de comandos para atributos de primitiva se caracteriza por tener los bits 15 y 14 puestos en "0", y el bit 13 puesto en "1".

El grupo de comandos de segmentos se caracteriza por tener el bit 15 puesto en "0", y los bits 14 y 13 puestos en "1".

En el Apéndice tres se muestran los comandos GKF de las principales funciones gráficas empleadas para programar las rutinas que generan las imágenes que requiere este trabajo.

V Diseño, programación e implantación del programa de supervisión mediante despliegues gráficos.

#### Introducción

- 5.1 Especificaciones y requerimientos funcionales de DIAFLU
- 5.2 Biblioteca de dispositivos
- 5.3 Diseño y programación de DIAFLU

## Introducción

En este capítulo se hace referencia a los requisitos funcionales, la arquitectura y el criterio de diseño empleado para desarrollar la función de presentación Diagramas de flujo (DIAPLU) perteneciente a la IHM de la Central Termoeléctrica de Dos Bocas, Veracruz.

La finalidad de este programa es supervisar el comportamiento de los sistemas y equipos principales de la planta, mediante desplegados gráficos actualizados en tiempo real, con el objeto de proporcionar a los operadores una herramienta durante la supervisión de ésta.

## 5.1 Especificaciones y requerimientos funcionales de DIAFLU

Este módulo de software forma parte de un proyecto bajo contrato del Instituto de Investigaciones Eléctricas (IIE) con Comisión Federal de Electricidad (CFE), por lo cual los ingenieros de la planta termoeléctrica e investigadores del IIE determinaron las características de los programas que componen a la IHM, dando lugar a especificaciones y requisitos funcionales que deberían satisfacer.

### Especificaciones

Las especificaciones dadas por CFE para DIAFLU son las siguientes:

- 1 - La función podrá desplegarse en cualquier monitor.
- 2 - Se presentarán tres niveles de información: el primero abarcará la vista general, que será el diagrama de flujo de un paquete de la planta; el segundo la vista del sistema, y el tercero la vista del subsistema.
- 3 - Todas las respuestas o solicitudes del usuario se validarán con el fin de desplegar los mensajes de error correspondientes.

4 - En cualquier momento y nivel se podrá dar por terminada la función.

5 - La función podrá presentar variables digitales que representen el estado de los equipos.

6 - El alta o baja de las variables, así como el formato de presentación, se podrá modificar desde la terminal de la computadora host por medio de un programa especial de mantenimiento.

#### **Requerimientos funcionales**

Al igual que en el caso de las especificaciones, se tienen los siguientes requisitos funcionales para DIAFLU :

1 - El despliegue de esta función podrá hacerse en cualquier monitor que se encuentre libre y que el usuario lo haya elegido previamente mediante las teclas de selección de pantalla, en el teclado del sistema. La información que deberá presentar cada diagrama puede dividirse en :



### **Información estática**

Es aquella información que no sufre cambio durante la presentación del diagrama, como por ejemplo los elementos materiales (tuberías, líneas de energía, etc.), títulos en el diagrama, unidades de las variables, etc.

### **Información dinámica**

Es aquella información que se debe actualizar en tiempo real, como el valor de las variables analógicas y las variables asociadas a los elementos funcionales del diagrama (bombas, válvulas, cuchillas, ventiladores, motores, interruptores, etc).

2 - El primer desplegado será la vista general, en la cual se presentará como pantalla estática el diagrama de flujo de un paquete agrupando en sistemas a sus principales equipos.

A cada sistema representado se le asignará un número que también se desplegará y cuyo color dependerá del estado funcional de las variables que lo formen. Si alguna de estas variables se encuentra en estado de alarma crítica, el color será rojo; si el estado es de alarma precrítica, el color será amarillo y si todas se encuentran en estado normal, el color será azul.

3 - Una vez seleccionado el sistema que se desea, la siguiente vista será el diagrama de ese sistema, en el cual aparecerán representados los subsistemas que lo conforman. A cada subsistema representado en el diagrama se le asignará un número que se desplegará y cuyo color se definirá de acuerdo con el criterio establecido anteriormente.

Este desplegado mostrará en la información dinámica el estado de las variables analógicas y digitales del sistema. Así, si alguna variable analógica se encuentra en estado de alarma crítica estará de color rojo; si está en alarma precrítica, será de color amarillo y si está normal, de color azul. Las variables digitales, que muestran el estado de los equipos, estarán de color rojo si la variable indica el equipo arrancado o abierto y verde para equipo apagado o cerrado.

4 - Si se deseara observar los siguientes sistemas se podrá hacer de dos formas : mediante las teclas de "avanza página" y "regresa página" o mediante la tecla de "regresa nivel", volviendo a desplegar la vista general y seleccionando un nuevo sistema.

5 - Si se quisiera observar el nivel de los subsistemas, se debe escoger uno de los que presenta cualquier vista del sistema y mediante el uso de las teclas de "avanza página" y "regresa

página" se podrán ver los subsistemas restantes. La información dinámica que contendrá el diagrama de flujo del subsistema se mostrará con los colores definidos anteriormente.

6 - A fin de regresar al nivel de los sistemas se deberá presionar la tecla de "regresa nivel", volviendo a desplegar el sistema seleccionado previamente.

7 - Si existiera algún error al momento de seleccionar un sistema o subsistema se presentará un mensaje de error en color rojo por un período de 5 segundos.

## 5.2 Biblioteca de dispositivos

Para llevar a cabo la supervisión de la planta en tiempo real, se requiere la actualización dinámica de los iconos que aparecen en los diagramas. Debido a ello se detectó la necesidad de generar un módulo de software que funcionara como biblioteca, el cual incluyera a todas las rutinas necesarias para dibujar los iconos, de modo que DIAFLU pudiera hacer uso de ellas en el momento en que las requiriera.

A continuación se explica brevemente la función de la Biblioteca de dispositivos y la forma de operación de las rutinas que la forman.

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

## Rutinas de la Biblioteca de dispositivos

Las rutinas de la Biblioteca de dispositivos se encargan de generar el código GKF necesario para dibujar los elementos que se actualizan dinámicamente durante la supervisión de la planta. Estos elementos son : textos que indican el estado de un equipo, textos que indican el valor de una variable, válvulas, bombas, motores, ventiladores, cuchillas, interruptores, etcétera.

Los cambios que sufren estos elementos durante la supervisión del proceso son básicamente de color : al pasar de un estado activo (en operación) a uno inactivo (fuera de operación), o viceversa; en el caso de equipos y en el caso de los textos al pasar un equipo de un estado normal a uno anormal, al detectarse un disparo en dicho equipo, o bien al alarmarse alguna variable.

La Biblioteca de dispositivos está compuesta por más de 100 rutinas, las cuales contienen los comandos GKF, interpretados por el firmware del generador de despliegues para dibujar las primitivas necesarias con el objeto de formar un determinado icono con las características que requiere la aplicación, así como para modificar los atributos de estas primitivas.

A fin de programar las rutinas que integran la Biblioteca de

dispositivos, primero se identificó a los iconos y textos que se modifican en tiempo real por **DIAPLU** resultando un total de 127 y se procedió después a programar las rutinas en código GKF.

#### **Operación de la Biblioteca de dispositivos**

La Biblioteca de dispositivos (**DEVICES\_GKF.OLB**) se obtiene a partir del programa **DEVICES\_GKF.PAS** el cual contiene dos procedimientos : **Dibuja\_Icono** y **Dibuja\_Texto**, los cuales hacen llamados a todas las rutinas que originen el código GKF ejecutable para que lo interprete el firmware del generador de despliegues y genere en tiempo real los iconos y textos que requiere **DIAPLU**.

**Dibuja\_Icono** hace llamados a todas las rutinas que originan el código GKF para dibujar algún icono, mientras que **Dibuja\_Texto** se utiliza para desplegar el valor de las variables analógicas en distintos formatos.

La operación de estas rutinas es muy sencilla : Si **DIAPLU**, o cualquier otro módulo quiere dibujar un icono, debe llamar al procedimiento **DIBUJA\_ICONO** de la biblioteca y pasarle como parámetros el número de dispositivo que desea dibujar, ya que éstos se identifican con un número, así como el color y las

coordenadas X y Y de la posición que haya escogido, e internamente este procedimiento llama a la rutina que genera el código GKF del ícono deseado.

En el caso de las variables que deben desplegar su valor a través de un texto, su actualización en tiempo real se realiza mediante el procedimiento DIBUJA\_TEXTO de la biblioteca, al cual se le pasan los siguientes parámetros : número de caracteres a desplegar, color, coordenadas X y Y, y el arreglo de palabras que represente el valor a desplegar.

En la tabla 5-1 se muestran algunos de los íconos generados por la Biblioteca de dispositivos, el elemento del proceso al que representan y sus variables asociadas, es decir las variables que son supervisadas y desplegadas en tiempo real por DIAFLU.








ICONOS GENERADOS POR DIAFLU	ELEMENTO DEL PROCESO AL QUE REPRESENTA	VARIABLES ASOCIADAS
	EYECTOR	EYECTOR NORMAL TURBINA DE VAPOR
	MOTOR ACOPLADO A LA FLECHA DEL GENERADOR	DISPOSITIVO DE ARRANQUE GENERADOR ELECTRICO TG
	INTERRUPTOR DE ALTA PRESION	PRESION AIRE SELLO COMPRESOR TG PRESION DIFERENCIAL FILTROS LUBRICACION TG PRESION DIFERENCIAL BBA AGUA ALIMENT.RECUPERADOR CALOR PRESION ACEITE SISTEMA ELECTROHIDRAULICO TV
	VENTILADOR	VENTILADOR INFERIOR AIRE/AIRE TURBINA DE GAS VENTILADOR SUPERIOR AIRE/AIRE TURBINA DE GAS
	INTERRUPTOR	INTERRUPTOR BAJA TENSION EXCITADOR GENERADOR TG INTERRUPTOR CAMPO EXCITADOR GENERADOR TG INTERRUPTOR CAMPO EXCITADOR GENERADOR TV INTERRUPTOR ALTA TENSION EXCITADOR GENERADOR TV
	CUCHILLA PARA TIERRA	CUCHILLA SUBESTACION
	INTERRUPTOR DE ALTA TEMPERATURA	TEMPERATURA AGUA SELLOS BBA.ALIMENTACION REC.CALOR TEMPERATURA FLUIDO ELECTROHIDRAULICO TV TEMPERATURA ACEITE CONVERTIDOR GENERADOR TG

TABLA 5-1 ICONOS GENERADOS POR LA BIBLIOTECA DE DISPOSITIVOS








ICONOS GENERADOS POR DIAFLU	ELEMENTO DEL PROCESO AL QUE REPRESENTA	VARIABLES ASOCIADAS
	NIVEL	NIVEL DE ACEITE DE LUBRICACION CARTER TG NIVEL TANQUE ANTIESPUMA GENERADOR ELECTRICO TG NIVEL CONDENSADOR VAPOR SELLOS TV NIVEL TANQUE DE ACEITE LUBRICANTE TV NIVEL CARTER ACEITE ELECTRO-HIDRAULICO TV
	BOMBA	BOMBA AGUA ALIMENTACION RECUPERADOR DE CALOR BOMBA LUBRICACION BOMBA AGUA ALIMENTAC REC. CAL. BOMBA ACEITE DE LUBRICACION TV BOMBA AGUA DE REPUESTO CONDENSADOR BOMBA AGUA DE CIRCULACION CONDENSADOR
	CUCHILLA	CUCHILLA ALTA TENSION GENERADOR ELECTRICO TG CUCHILLA BAJA TENSION GENERADOR ELECTRICO TG CUCHILLA ALTA TENSION GENERADOR ELECTRICO TV CUCHILLA BAJA TENSION GENERADOR ELECTRICO TV
	VALVULA	VALVULA VENTED DE GAS TG VALVULA DRENAJE COMBUSTIBLE TG
	VALVULA	VALVULA PRINCIPAL VAPOR DE RECUPERADOR DE CALOR VALVULA DE CORTE SUPERIOR TV VALVULA DE CORTE INFERIOR TV VALVULA ROMPEDORA DE VACIO EYECTORES TV VALVULA DESCARGA BOMBA AGUA DE CIRCULACION C.C.
	VALVULA	VALVULA SOBREVOLOCIDAD TG
	VALVULA	VALV. SUCCION BBA. AGUA DE ALIMENTACION REC. CALOR

TABLA 5-1 ICONOS GENERADOS POR LA BIBLIOTECA DE DISPOSITIVOS










ICONOS GENERADOS POR DIAFLU	ELEMENTO DEL PROCESO AL QUE REPRESENTA	VARIABLES ASOCIADAS
	VALVULA	VALVULA DE SANGRADO TG
	VALVULA	VALVULA DE CORTE INFERIOR TV VALVULA DE CORTE SUPERIOR TV
	VALVULA	VALVULA DESCARGA BBA. AGUA ALIMENT. RECUPERADOR DE CALOR
	TORNAFLECHA DEL GENERADOR	TURBINA EN TORNAFLECHA TG MOTOR TORNAFLECHA GIRANDO TV
	REJILLA	REJILLA MOVIL SIST AGUA DE CIRCULACION
	FLAMA	FLAMA TG PRESENTE FLAMA QUEMADOR POSTERIOR RECUPERADOR DE CALOR
	BOMBA VERTICAL	BBA. PRINCIPAL ACEITE DE LUBRICACION TV BBA. RESPALDO ACEITE DE LUBRICACION TV

TABLA 5-1 ICONOS GENERADOS POR LA BIBLIOTECA DE DISPOSITIVOS

### 5.3 Diseño y programación de Diagramas de flujo

Al ejecutarse DIAFLU necesita hacer acceso a varios archivos almacenados en disco, los cuales abarcan la información de las variables que se desplegarán y el archivo que contiene a todas las pantallas de los diagramas que son desplegados. A continuación se explica brevemente el proceso desarrollado para generar estos archivos y la arquitectura empleada para diseñar a DIAFLU .

#### Los archivos de Diaflu

A fin de operar, DIAFLU requiere tres archivos : DIAFLU\_SIS.DOS, DIAFLU\_SUB.DOS y PICTURE\_FILE.COV; en los dos primeros se tiene la información necesaria para acceder a la base de datos y desplegar la información de las variables de los diagramas que presenta la función. El tercer archivo contiene el código GKF para dibujar los diagramas que se presentan de manera estática.

#### Generación de los archivos

Como se mencionó anteriormente, todos los desplegados que aparecen en DIAFLU se realizaron con el editor gráfico AYPIX el

cual genera archivos con las extensiones : .DIS, .CNF, .PIC, .GKF, .DIR, .DEV y .TAG para cada uno de los dibujos creados.

Los archivos que requiere DIAFLU se obtienen a partir de los archivos de extensión .PIC y .TAG generados por AYPX. En los .PIC no sólo se almacenan los códigos GKF necesarios para desplegar un determinado dibujo, sino que a partir de ellos también se genera el archivo PICTURE\_FILE.COV en el cual se tienen los códigos GKF para desplegar todas las pantallas estáticas que se requieren para la operación de DIAFLU.

En los archivos de extensión .TAG se almacenan los elementos o dispositivos que se intitularon en el dibujo, es decir, el nombre con que se rotuló cada uno de los dispositivos, su número y sus coordenadas X y Y en el dibujo.

En este caso se rotularon aquellos elementos que representan a una variable del proceso dentro del dibujo y el nombre que se les asignó fue la llave de cada una de estas variables para acceder a la base de datos del sistema.

Los archivos DIAFLU\_SIS.DOS y DIAFLU\_SUB.DOS se originaron a partir de los archivos de extensión .TAG generados por AYPX.

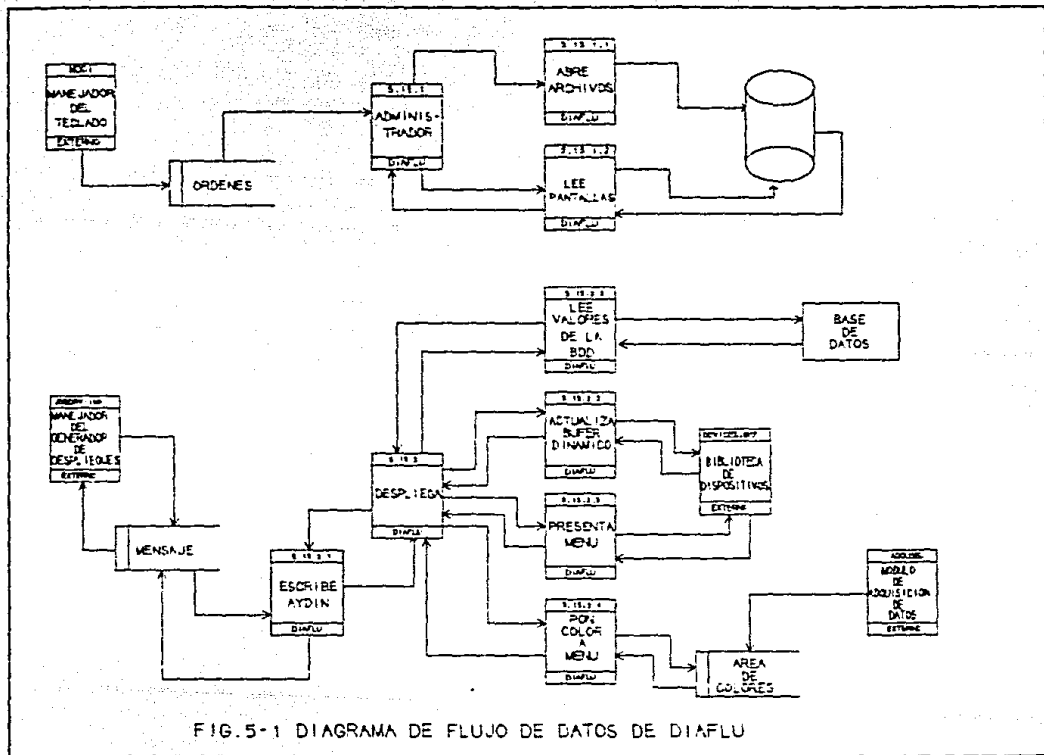


FIG. 5-1 DIAGRAMA DE FLUJO DE DATOS DE DIAFLU

## Arquitectura de DIAFLU

Diaflu está compuesto por dos procesos : uno de administración y uno de despliegue. En la figura 5-1 se muestra el diagrama de flujo de datos de DIAFLU y a continuación se describen las tareas que ejecuta cada uno de los procesos.

El proceso de administración se encarga de :

- Recibir las selecciones que hace el operador a través del teclado del sistema, validarlas y decidir las acciones a ejecutar de acuerdo con la selección recibida.
- Calcular las direcciones y leer los códigos GKF de la pantalla estática a desplegar de acuerdo con el sistema y subsistema en los que se encuentre la función.
- Hacer las lecturas de las tablas archivo de la función

El proceso de despliegue se encarga de :

- Hacer lecturas de la base de datos del sistema
- Hacer llamadas a las rutinas de la Biblioteca de dispositivos para dibujar los iconos y textos que la pantalla requiere.

- Evaluar el color en que se deben desplegar los números de menú que aparecen en la vista general y en las vistas de los sistemas de la función.
- Enviar los códigos GKF de las pantallas a desplegar al manejador del generador de despliegues.

A continuación se muestra el pseudocódigo de DIAFLU :

## Pseudocódigo de Diagramas de flujo

### Módulo Diaflu

---

#### Procedimiento Lee\_pantalla

---

Entradas: Número de sistema y subsistema de la pantalla a leer

Salidas : Códigos GKF de la pantalla a desplegar

Rutinas que llama :

Inicio

Calcula la dirección de registro de la pantalla a leer de acuerdo con su sistema y subsistema

Posiciónate en el archivo y lee la pantalla

Fin

---

#### Procedimiento Abre\_archivos

---

Entradas: Trayectorias en disco de los archivos de DIAFLU

Salidas : Archivo de pantallas estáticas abierto y tablas archivo en RAM.

Rutinas que llama :

Inicio

Abre archivo de tabla\_archivo para sistemas y lee su contenido

Abre archivo de tabla\_archivo para subsistemas y lee su contenido

Abre archivo de pantallas

Fin

---

=====

PROGRAMA PRINCIPAL DE DIAGRAMAS DE FLUJO

=====

Inicio

Crea puertos y enlazate con el manejador del teclado del sistema

Crea las áreas para acceder a la base de datos

Abre\_archivos

Crea semáforos de sincronización

Crea proceso despliega

Por siempre :

    Espera mensaje del manejador del teclado

    Decodifica el mensaje (tecla y monitor)

    En caso de tecla:

        Dflu : Si monitor libre entonces

            Lee pantalla de vista general

            Espera por semáforo

        c.c.

            Envía mensaje de error al monitor

    Rpag : En caso de nivel

        1 : manda mensaje de error al monitor

        2 : si sistema=1 entonces

            sistema=11

        c.c.

            sistema=sistema-1



Lee\_pantalla

Espera por semaforo

3 : Si subsistema=1 entonces

subsistema=número de subsistemas  
del sistema actual

c.c.

subsistema=subsistema-1

lee\_pantalla

espera por semaforo

Apag : En caso de nivel

1 : manda mensaje de error al monitor

2 : si sistema=11 entonces  
sistema=1

c.c.

sistema=sistema+1

lee\_pantalla

espera por semáforo

3 : Si subsistema=número de subsistemas del  
sistema actual entonces

subsistema=1

c.c.

subsistema=subsistema+1

lee\_pantalla

espera por semáforo

Rniv : En caso de nivel

1 : Manda mensaje de error al monitor

2 : Nivel=1

lee\_pantalla

espera por semáforo

3 : nivel=2

lee\_pantalla

espera por semáforo

Ffun : Apodérate del área del manejador del teclado

Libera estructura ocupada por la función

Libera área del manejador del teclado

Nivel=4

Espera por semáforo

Txmt : Decodifica el mensaje

en caso de nivel

1 : Si selección no esta en [1...11]  
entonces

manda mensaje de error al monitor

c.c.

nivel=2

sistema=selección

lee\_pantalla

espera por semáforo

2 : Si seleccion no esta en [1..maximo de subsistemas del sistema actual] entonces

manda mensaje de error al monitor

C.C.

nivel=3

Subsistema=selección

Lee\_pantalla

Espera por semáforo

3 : Manda mensaje de error al monitor

Fin (del programa principal de Diaflu)

-----  
=====

PROCESS\_BLOCK DESPLIEGA

=====

-----

Procedimiento Escribe\_aydin

-----

Entradas : Pantalla a desplegar

Monitor en el que se desea desplegar la pantalla

Puerto en el que la función recibirá la respuesta del manejador del equipo de despliegues.

Salidas : Mensaje para el manejador del equipo de despliegues

Rutinas que llama :

Inicio

Crea el mensaje a enviar al manejador del equipo de despliegues

Deposita los datos a enviar en el mensaje

Envía el mensaje al manejador del generador de despliegues  
Espera la respuesta del manejador del Aydin

Fin

-----  
-----  
Procedimiento Evalúa\_color  
-----  
-----

Entradas : Estructura donde Adquise1 registra el color de todas las variables de la base de datos

Cantidad de números de menú a evaluar

Salidas : Arreglo con el código de color de cada uno de los números de menú evaluados

Rutinas que llama :

Inicio

Para cada número de menú :  
Evalúa el color de acuerdo con la lógica establecida

Pon color en el arreglo de códigos

Fin

-----  
-----  
Procedimiento Pon\_color\_a\_menú  
-----  
-----

Entradas : Nivel en que se encuentra la función

Sistema actual

Número de subsistemas del sistema actual

Salidas : Arreglo con el código de color correspondiente a cada número de menú existente en la pantalla

Rutinas que llama : Evalúa\_color

Inicio

En caso de nivel

1: Apoderate del área compartida con Adquasel

Lee el color de las variables de los 11 sistemas de la vista general

Libera el area

Evalúa color

2: Apoderate del área compartida con Adquasel

Lee el color de las variables de los subsistemas que contenga el sistema que estas presentando

Libera el área

Evalúa color

Fin

-----  
-----  
Procedimiento Draw\_aydin\_full  
-----  
-----

Entradas : Número de dispositivo  
          Número de decimales con que se presentará el valor  
          Valor real de la variable leído de la base de datos

Salidas : Arreglo de palabras

Rutinas que llama :

Inicio

Convierte el valor real leído de la bdd a un arreglo de 8 bytes con el número de decimales especificado

Reacomoda el arreglo de 8 bytes en uno de 6

Convierte el arreglo de 6 bytes en un arreglo de 3 palabras

Fin

---

Procedimiento Lee\_valores\_de\_la\_bd

---

Entradas : Número de variables a desplegar en el diagrama actual

Arreglo con las variables que se van a leer de la bdd

Salidas : Color y valor actual leído de la Base de Datos

Rutinas que llama : Lectura\_diaflu (Externa)

Draw\_aydin\_full .

Inicio

Para el número de variables de la pantalla actual

Lectura\_diaflu (color, valor)

Si se trata de una variable analogica entonces

Draw\_aydin\_full

Fin

---

---

Procedimiento Actualiza\_bufér\_dinámico

---

Entradas : Número de variables a desplegar en el diagrama actual

Arreglo con las variables que se van a desplegar

Salidas : Buffer con códigos GKF para enviar al manejador del equipo de despliegue

Número de palabras que contiene el buffer

Rutinas que llama : Dibuja\_texto (Externa)

Dibuja\_ícono (Externa)

Inicio

Para el número de variables

Si la variable es analógica entonces

Dibuja\_texto(color,  
                  coordenada x,  
                  coordenada y,  
                  número de caracteres,  
                  códigos GKF del valor a desplegar)

c.c.

Dibuja\_icono(número de dispositivo,  
                  color,  
                  coordenada x,  
                  coordenada y)

Fin

-----  
-----  
Procedimiento Pantalla\_aydin  
-----  
-----

Entradas : Nivel

Número de monitor en el que se despliega

Puerto de comunicación entre la función y el manejador  
del equipo de despliegues

Semáforo de sincronización

Bandera para controlar los envíos al manejador del  
equipo de despliegues

Salidas : Pantalla a enviar al manejador del equipo de  
despliegues

Rutinas que llama : Escribe\_aydin  
                  Atr\_menü  
                  Pon\_color\_a\_menü  
                  Presenta\_menü  
                  Lee\_valores\_de\_la\_bdd  
                  Actüaliza\_büfer\_dinámico

## Inicio

Si nivel no es 4 entonces

Pon en el buffer a enviar al aydin la pantalla estática  
c.c.

Pon en el buffer a enviar al aydin los códigos para  
limpiar la pantalla del monitor

Escribe\_aydin(bufer, monitor, puerto de enlace)

En caso de nivel

1: Atr\_menu

Repite

Pon\_color\_a\_menú

Presenta\_menú

Escribe\_aydin

Hasta que cambie bandera de envío al aydin

2: Repite

Lee\_valores\_de\_la\_bd

Actualiza\_bufer\_dinámico

Pon\_color\_a\_menú

Atr\_menú

Presenta\_menú

Escribe\_aydin

Hasta que cambie bandera de envío al aydin

3: Repite



```
    Lee_valores_de_la_bd
    Actualiza_bufer_dinámico
    Escribe_aydin
Hasta que cambie bandera de envío al aydin
Fin
```

---

---

Programa principal del Process\_Block Despliega

---

---

Inicio

```
    Crea los puertos y enlázate con el manejador del Aydin
    Crea procesos para atención de monitores
    Abre archivo con la configuracion GKF de default
    Mientras no sea fin de función
        Pantalla_aydin
        Libera semáforo

    Cierravos      en disco
```

Fin

---

Fin {del modulo Diaflu}

## Operación de DIAFLU

Los operadores de la planta termoeléctrica interactúan con los módulos de la interfaz hombre-máquina a través del teclado del sistema, mediante el cual pueden solicitar la activación de algún módulo en especial y, una vez activo éste, el diálogo se realiza a través de la selección de alguna opción presentada por los diversos menús con que cuentan los módulos.

El desplegado inicial del módulo Diagramas de flujo es un diagrama que muestra todos los equipos que intervienen en el ciclo de generación de energía de un paquete de la planta agrupados en sistemas.

Una vez que el usuario ha seleccionado un sistema en particular, se le presenta el diagrama de los subsistemas que lo conforman, y se le solicita seleccione uno de estos subsistemas.

Al seleccionar un subsistema se muestra el diagrama correspondiente, con lo cual el módulo llega a su nivel de mayor profundidad en cuanto a la presentación de información en detalle.

De manera similar, el usuario de la interfaz hombre-máquina puede obtener información de cualquier sistema o subsistema de la planta que sea de su interés, a través de las teclas de "regresa nivel", "avanza página" y "regresa página".



## Conclusiones

Al inicio de este trabajo se mencionó que el objetivo principal de esta tesis era diseñar e implantar un programa de computadora que proporcionara una herramienta para la supervisión de una planta termoeléctrica, y se puede afirmar que dicho objetivo se cumplió satisfactoriamente, pues DIAFLU forma parte de la interfaz hombre-máquina que actualmente se encuentra instalada en la planta termoeléctrica de Dos Bocas, en la Ciudad de Veracruz.

Desde que se llevó a cabo esta instalación, los operadores han empleado a DIAFLU para supervisar sus sistemas y equipos principales sin que tengan dificultad alguna para interpretar los diagramas desplegados en los monitores. Esto comprueba la utilidad de haber realizado los diagramas siguiendo una norma para la representación de dispositivos y además haber dibujado los equipos de acuerdo con su forma física real.

Como se mencionó con anterioridad, DIAFLU forma parte de un sistema que opera en tiempo real. En este tipo de sistemas el tiempo de respuesta de los programas que lo integran es de suma importancia para su desempeño global; los tiempos de respuesta de DIAFLU en activación, refresco de información y avances/retrocesos de página son congruentes con los requerimientos del sistema.

El haber creado una biblioteca con las rutinas para dibujo de iconos, además de dotar de modularidad y hacer totalmente configurable a DIAFLU, contribuyó a que el código fuente de este programa sea pequeño. Por otra parte, la arquitectura de DIAFLU permite que todos los cambios y actualizaciones en la información dinámica que muestran los diagramas sean sumamente fáciles de incorporar.

Considero que el estudio de la norma gráfica GKS para el diseño y la posterior programación de las rutinas de dibujo de iconos fue fundamental, pues proporcionó las bases para la comprensión de otras normas gráficas.

Finalmente se debe mencionar que las experiencias que se obtuvieron al desarrollar este trabajo fueron invaluable, tanto en el plano personal como en el profesional; dichas experiencias se vieron enriquecidas enormemente debido a que el diseño y programación de la Biblioteca de dispositivos y de DIAFLU se realizó en un 90 % en la planta de Dos Bocas, lo cual hizo posible establecer contacto con los operadores e ingenieros de la planta y sus comentarios acerca de los desplegados y la información que presentan serán de utilidad en futuros proyectos.

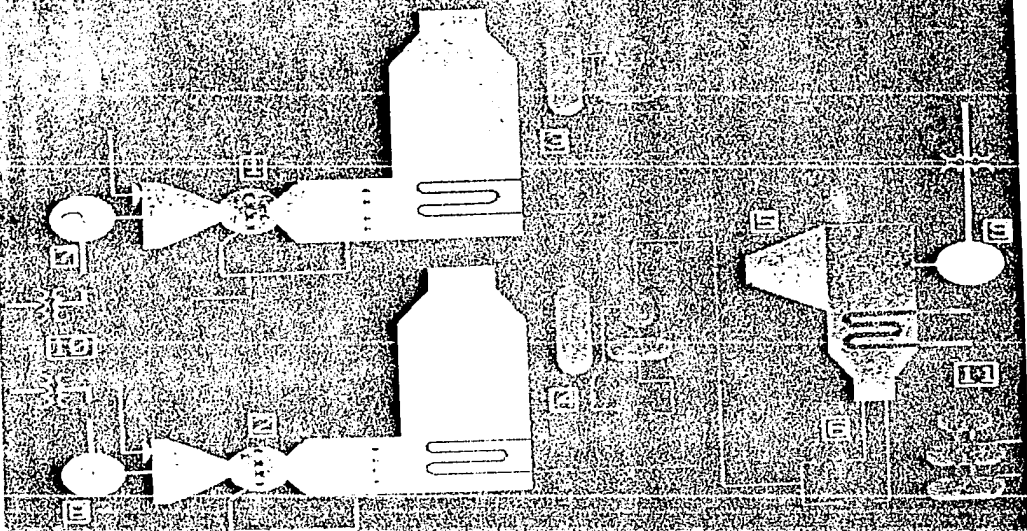
## **Apéndice uno**

### **Algunos desplegados de Diagramas de flujo**

02/2/89

11/25/16

# CICLO COMBINADO DOS BOCAS



ALUE - 0011

SELECCION 1  
1200-115-10 (GON 513) 2049

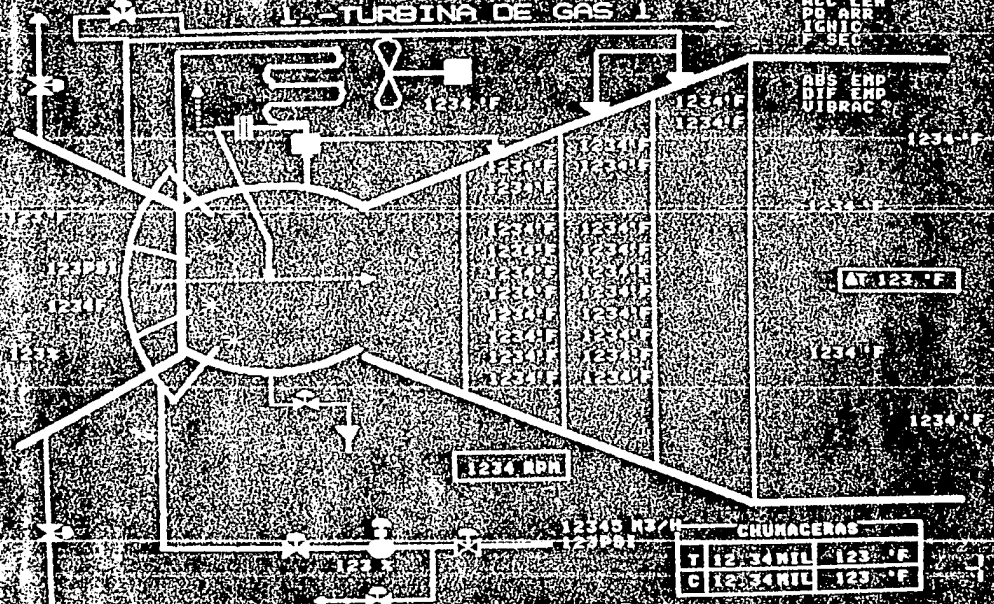




02/03/83

10 09 49  
U SANG  
ACC LEN  
PO ARR  
IGNIC  
F SEC

# L. - TURBINA DE GAS 1



ABS EMP  
DIF EMP  
VIBRAC

AT 123 °F

1234 RPN

CHUMACERAS	
T 12.34 MIL	123 °F
C 12.34 MIL	123 °F

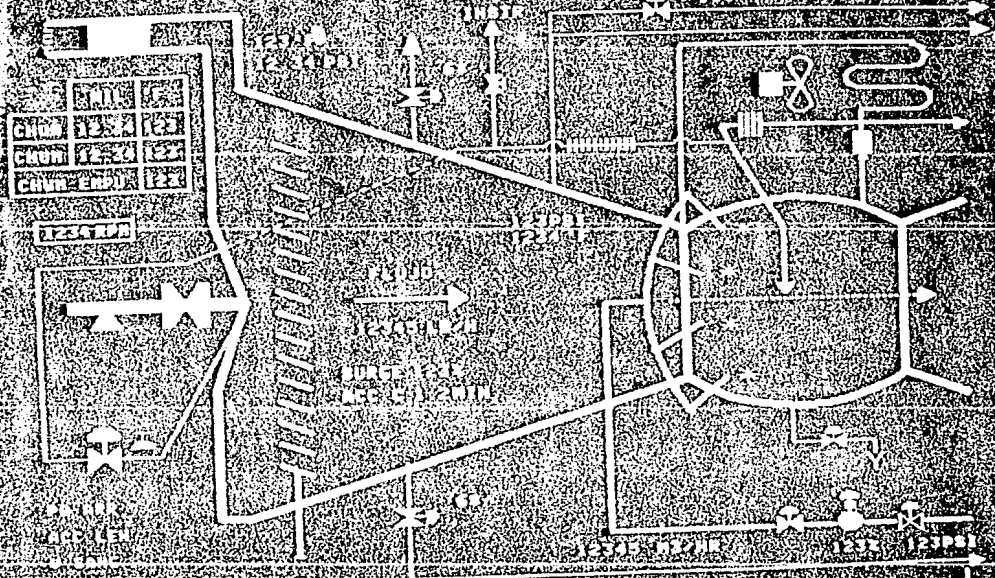
BLUP = BOLT

GRID SIZE 256 ICON SIZE 2048

01/03/80

11 29 58

### COMPRESSOR TO



12345	12346	12347
12348	12349	12350
12351	12352	12353
12354	12355	12356
12357	12358	12359
12360	12361	12362
12363	12364	12365
12366	12367	12368
12369	12370	12371
12372	12373	12374
12375	12376	12377
12378	12379	12380
12381	12382	12383
12384	12385	12386
12387	12388	12389
12390	12391	12392
12393	12394	12395
12396	12397	12398
12399	12400	12401

12345

12346  
12347  
12348  
12349  
12350

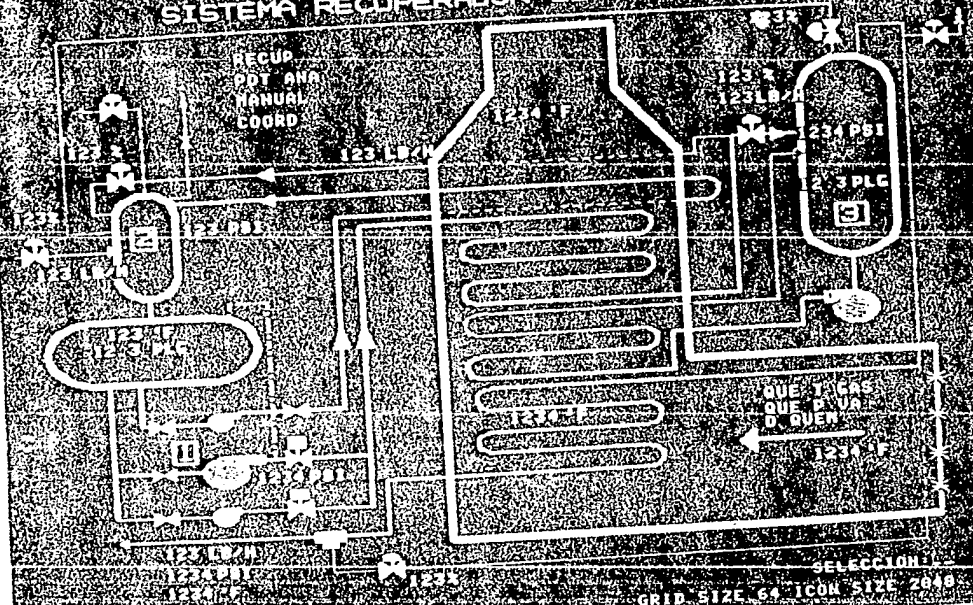
12351  
12352  
12353  
12354  
12355  
12356  
12357  
12358  
12359  
12360

GRID SIZE 120 ICON SIZE 2048

03/07/85

12 47 50

# SISTEMA RECUPERADOR DE CALOR 1



BLUE = OUIY

03/07/09

17 50 00

### 3.- DOMO Y SOBREALENTADO REC 1



QUE T CAS  
QUE P UA  
D. QUEN

1234 F

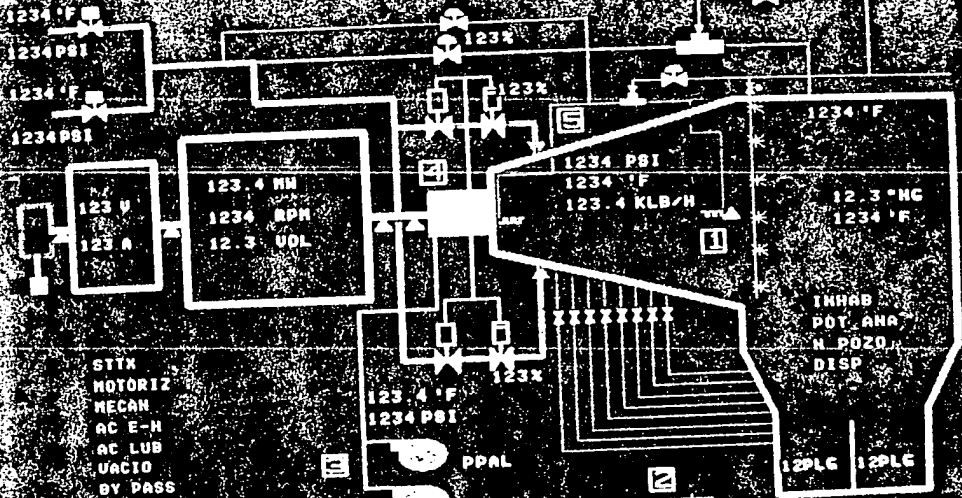
BLUE = QUIY

GRID SIZE 64 ICON SIZE 2848

14/07/85

12 06 30

# SISTEMA TURBINA VAPOR



STTX  
 MOTORIZ  
 MECAN  
 AC E-M  
 AC LUB  
 VACIO  
 BY PASS  
 POT ANA  
 VIBRA

MILB  
 CHU1 12.34  
 CHU2 12.34

SELECCION

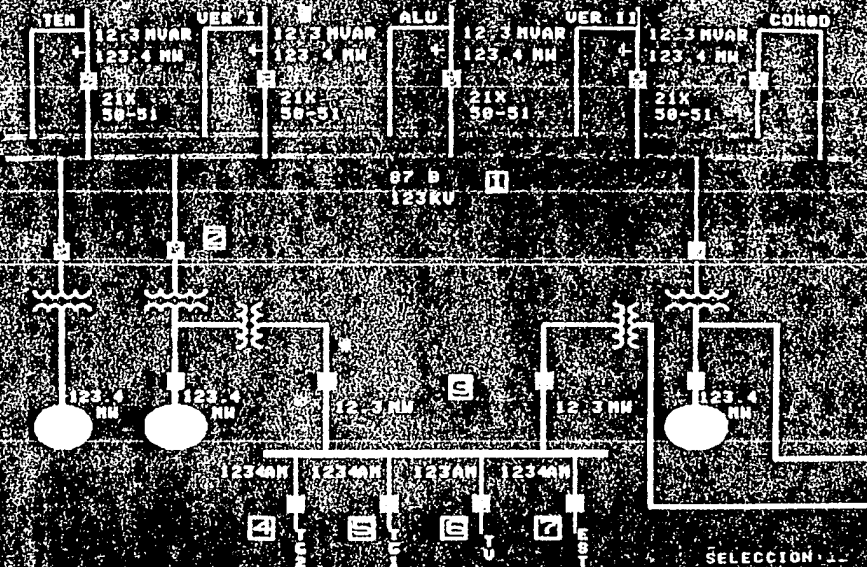
GRID SIZE 128 ICON SIZE 2048

BLUE = QUIET

30/06/89

13 27 42

# SISTEMA SUBESTACION ELECTRICA



BLUC = QUIT

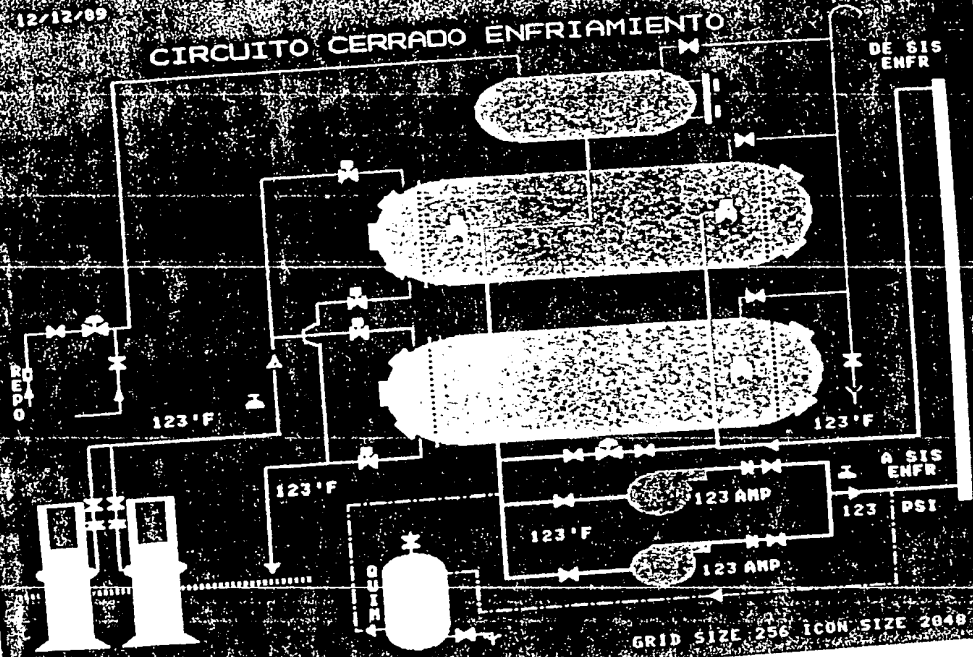
SELECCION 1  
GRID SIZE 64 ICON SIZE 2048



12/12/89

88-45-55

# CIRCUITO CERRADO ENFRIAMIENTO



BLUE = GUIT

GRID SIZE 256 ICON SIZE 2048



**Apéndice dos**

**Pseudocódigo del manejador del equipo de despliegues**

```
{-----}
Procedimiento Initdrv11wa
{-----}
```

Procedimiento Configura\_monitor (# monitor)

Inicio

Pon en modo Full Graphics (# monitor)

Carga la tabla de colores

Carga los planos de visión

carga la ventana

Fin

Procedimiento Carga\_las\_tablas\_de\_color (# monitor)

Inicio

Configura\_monitor (# monitor)

Envía al generador de despliegues

Fin

```
{-----}
Programa principal de Initdrv11wa
{-----}
```

Inicio

Espera\_cursor = falso

Abre como archivo lógico el puerto TXA3 de la host

Monta el disco DUA0 de la host

Crea los mensajes necesarios para manejar el buffer a transmitir al generador de despliegues

Crea los mensajes de comunicación con las funciones de presentación

Ubica el nombre del puerto de comunicación con el manejador del teclado del sistema

Inicializa al generador de despliegues

Crea el area de datos compartida con el manejador del teclado del sistema

Crea el proceso Lee\_kbd

Crea el proceso Tiempo

Desde i=0 hasta 3

Carga\_las\_tablas\_de\_color (monitor #i)

Inicializa la estructura de datos donde se lleva la posición XY del cursor para cada monitor

Fin

(-----)

(=====)

Procedimiento Administrador

(=====)

Inicio

Espera por un mensaje

Recibe el mensaje

Coloca en el encabezado del buffer el número de la workstation a la que va el mensaje

Si el mensaje no es nulo entonces

transmite al generador

c.c.

no hacer nada

Actualiza las variables globales Coord\_x y Coord\_y

Actualiza la estructura de datos donde se lleva el control de la posición XY del cursor para cada workstation

Si (espera\_cursor = verdadero) AND  
(monitor = ws\_posee\_kbd) entonces

Libera el semáforo del proceso Lee\_kbd

Fin

{-----}

{=====  
Programa principal del manejador del generador de despliegues  
=====}

Inicio

Initdrvllwa

Por siempre ejecuta :

administrador

Fin

{-----}

{=====  
Process\_Block Lee\_kbd  
=====}

Inicio

Carga las coordenadas X y Y del cursor por omisión

No\_pruebas = verdadero

Dialar\_act = falso

Por siempre :

Ubica la workstation que tiene asignada el teclado

Si (Dialar se activa en esa workstation) AND  
(No\_pruebas = verdadero) entonces

Dialar\_act = verdadero

No\_pruebas = falso

Ubica cursor

C.C.

no hacer nada

Si (Dialar se desactiva en esa workstation) entonces

No\_pruebas = verdadero

Dialar\_act = falso

C.C.

no hacer nada

Espera por un código

En caso de código :

CR : Si (dialar\_act = falso) entonces

Llena buffer de eco con blancos

Envía buffer de eco al generador de despliegues

Envía la cadena recibida al manejador del teclado del sistema

Espera\_cursor = verdadero

Bloquea (semáforo)

Espera\_cursor = falso

Reinicializa algunas variables

c.c.

no hacer nada

BS : Si (dialar\_act = falso) entonces

Pon un blanco al último caracter que ingresó al buffer de eco

Envía buffer de eco al generador de despliegues

Actualiza algunas variables

c.c.

no hacer nada

CONTROL : Espera por el segundo código de control

Si (segundo código de control es cambio de monitor) entonces

Carga en ese monitor la posición XY del cursor que se tiene en ese momento

Actualiza la variable ws\_posee\_kbd

Llena buffer de eco con blancos

c.c.

Si (segundo código de control es "flecha  
arriba" o "flecha abajo") AND  
(dialar\_act = verdadero) entonces

Actualiza la nueva posición XY  
del cursor

c.c.

no hacer nada

DE OTRO MODO : Si (dialar\_act = falso) AND  
(código\_ esta en ['A'...'Z', '0'...'9'])  
entonces

Actualiza contador 'cont\_char'

Cadena\_temp = 'cadena\_temp' + código

Si 'cont\_char' <= 8 entonces

Actualiza buffer de eco

Envía buffer de eco al generador  
de despliegues

c.c.

Pon blancos al buffer de eco

Inicializa 'cont\_char' = 0

Envía buffer de eco al generador  
de despliegues

c.c.

no hacer nada

Fin

```
(=====)
                          Process_Block Tiempo
(=====)
```

Inicio

Abrir el archivo donde estan los códigos GKF de la fecha

Crear el buffer donde se harán las transmisiones de la fecha a los monitores donde haya una función de presentación activa

Leer los códigos GKF de la fecha

Por siempre ejecuta :

Obtén el tiempo del sistema

Cronología = fecha en string

Fecha = substring de cronología

Hora = substring de cronología

Convertir fecha a words

Convertir hora a words

Depositara fecha en buffer

Depositara hora en buffer

Depositara en inf\_fps la información de las funciones de presentación que tiene el manejador del teclado del sistema

Desde i=0 hasta 3 ejecuta

Si inf\_fps^[i] no está terminada entonces

En caso de inf\_fps^[i].monitor

0 : carga en el buffer el código para



monitor 0

1 : carga en el buffer el código para  
monitor 1

2 : carga en el buffer el código para  
monitor 2

3 : carga en el buffer el código para  
monitor 3

Enviar buffer al generador de despliegues

C.C.

no hacer nada

Espera un segundo

Fin

{-----}

### **Apéndice tres**

**Comandos gráficos proporcionados por el GKP**

Grupo de Comandos	Descripción	Código en Hexadecimal
-------------------	-------------	-----------------------

Estación de trabajo (WS)	WS (selecciona)	A008
	WS superficie de visión (selecciona)	A000
	WS ventana (selecciona)	A002
	WS puerto de visión (selecciona)	A003
	WS ventana/puerto de visión (selecciona ambos)	A001
	WS rotación, escala, transl., (selecciona todo)	A005
	WS rotación (selecciona)	A006
	WS escale (selecciona)	A007
	WS translación (selecciona)	A004
	WS angulo de visión (selecciona)	A009
	WS color (habilita/deshabilita)	A00C

Primitiva	Flecha (dibuja)	000F
	Limpia (pantalla)	000C
	Limpia (puerto de visión)	000D
	Circulo (dibuja)	0003
	Arco circular (dibuja)	0005
	Posición actual (obten)	0100
	Posición actual (pon)	1000
	Elipse (dibuja)	0006
	Arco elíptico (dibuja)	0007
	Area llena (dibuja)	000B
	No operación	0000
	Polígono irregular cerrado (dibuja)	0008
	Poligono regular (dibuja)	0009
	Polilínea (dibuja)	0001
	Polimarcador (dibuja)	0002
	Rectangulo (dibuja)	0011
	Texto (dibuja)	0012
Triángulo (dibuja)	0010	

Grupo de Comandos	Descripción	Código en Hexadecimal
-------------------	-------------	-----------------------

Atributos de Primitiva	Area llena (pon todos los atributos)	201C
	Area llena (pon color)	2018
	Area llena (pon patrón)	2019
	Area llena (pon separación de patrón)	201B
	Area llena (pon estilo)	201A
	Tabla de colores (pon una dirección)	201F
	Tabla de colores (pon varias direcciones)	2025
	Selección (pon identificador)	201E
	Polilínea (pon todos los atributos)	2004
	Polilínea (pon color)	2001
	Polilínea (pon estilo)	2002
	Polilínea (pon ancho)	2003
	Polimarcador (pon todos los atributos)	2009
	Polimarcador (pon color)	2006
	Polimarcador (pon tipo)	2007
	Polimarcador (pon tamaño)	2008
	Atributos de primitiva (pon todos)	201D
	Texto (pon todos los atributos)	2016
	Texto (pon color fondo)	200C
	Texto (pon color frente)	200B
	Texto (pon "font")	2013
	Texto (pon altura)	2011
	Texto (pon justificación)	2015
	Texto (pon trayectoria)	200E
	Texto (pon precisión)	2014
	Texto (pon rotación)	200D
	Texto (pon escala)	2010
	Texto (pon espaciamiento)	200F
	Texto (pon ancho)	2012

Grupo de Comandos	Descripción	Código en Hexadecimal
Segmento	Segmento (cierra)	6001
	Segmento (crea)	6000
	Segmento (borra)	6002
	Segmento (borra todos)	600E
	Segmento (ejecuta)	6006
	Segmento (ejecuta todos)	600F
	Segmento (carga)	6015
	Segmento (recupera de memoria)	6013
	Segmento (lee)	6012
	Segmento (lee encabezado)	6019
	Segmento (renombrar)	6004
	Segmento (pon atributos)	600D
	Segmento (pon puntos de error)	600C
	Segmento (pon banderas)	6008
	Segmento (pon prioridad)	6007
	Segmento (pon rotación)	600A
	Segmento (pon escala)	600B
	Segmento (pon translación)	6009
	Segmento (pon puerto de visión)	6017
	Segmento (pon angulo de visión)	6018
	Segmento (pon ventana)	6016

Grupo de Comandos	Descripción	Código en Hexadecimal
Funciones de E/S	Botón (inicializa)	800C
	Botón (pon decodificador)	8011
	Botón (pon en modo)	8014
	Botón (muestrea)	8019
	Botón (solicita)	801E
	Canal (cancela)	8024
	Punto (inicializa)	800E
	Punto (pon modo)	8016
	Punto (muestrea)	801B
	Punto (solicita)	8020
	Dispositivo (ejecuta)	8008
	Dispositivo (lee bytes)	8004
	Dispositivo (lee estado)	8000
	Dispositivo (escribe un byte)	8005
	Dispositivo (escribe bytes)	8006
	Localización (inicializa)	800A
	Localización (pon decodificador)	800F
	Localización (pon en modo)	8012
	Localización (muestrea)	8017
	Localización (solicita)	801C
	Localización (obten)	8021
	Localización (pon)	8022
	Selección (inicializa)	800D
	Selección (pon en modo)	8015
	Segunda selección (pon en modo)	A00D
	Selección (muestrea)	801A
	Selección (solicita)	801F
	Selección eco (habilita/deshabilita)	A00A
	Cola errores (lee)	8025
	Cola reportes (lee)	8023
	Cadena (inicializa)	800B
	Cadena (pon decodificador)	8010
	Cadena (pon en modo)	8013
Cadena (muestrea)	8018	
Cadena (solicita)	801D	

## Bibliografía

Enderle, G, K. Kinsky y G. Pfaff, **Computer Graphics Programming: GKS- The graphics standar**, 1a. ed., Berlin, Springer-Verlag, 1984, 594 p.

Mc Kay, Lucia, **GKS Primer**, 1a. ed., Austin Texas, Nova Graphics International Corporation, 1986.

Kliwer, Bradley, **EGA/VGA a Programmer's Reference Guide**, 1a. ed., New York, Mc Graw Hill Book Company, 1988, 269 p.

Graphic Software Systems Inc., **GSS-GKS Kernel System Programmer's Guide**, Versión 1.0, Wilsonville Oregon, 1985.

Aydin Controls, **Graphical Kernel System (GKS) Level 1C -Software Interface Manual for the Aycon 2320-**, (Rev. A), Washington, 1987.

Aydin Controls, **Technical Information - Firmware Manual**, Doc. No. 150-2320-004 (Rev. A), Washington, 1987.

Aydin Controls, Aycon 2320 Engineering Drawings, Doc. No. 150-2320-005 (Rev. A), Washington, 1897.

International Organization for Standardization (ISO), Information Processing Systems -Computer Graphics- Graphical Kernel System (GKS) Functional Description, (Ref. No. ISO 7942-1985 (E) ), 1a. ed., Suiza, 1985, 245 p.

Hearn, D., P. Baker, Gráficas por Computadora, 1a. ed., México, Prentice Hall, 1988, 380 p.

Hall, Douglas, Microprocessors and Interfacing, Programming and Hardware, 1a. ed., New York, Mc Graw Hill Book Company, 1986.

Lomov, B., V. Venda, La Interrelación Hombre-Máquina en los Sistemas de Información, 1a. ed., México, Progreso, 1983.