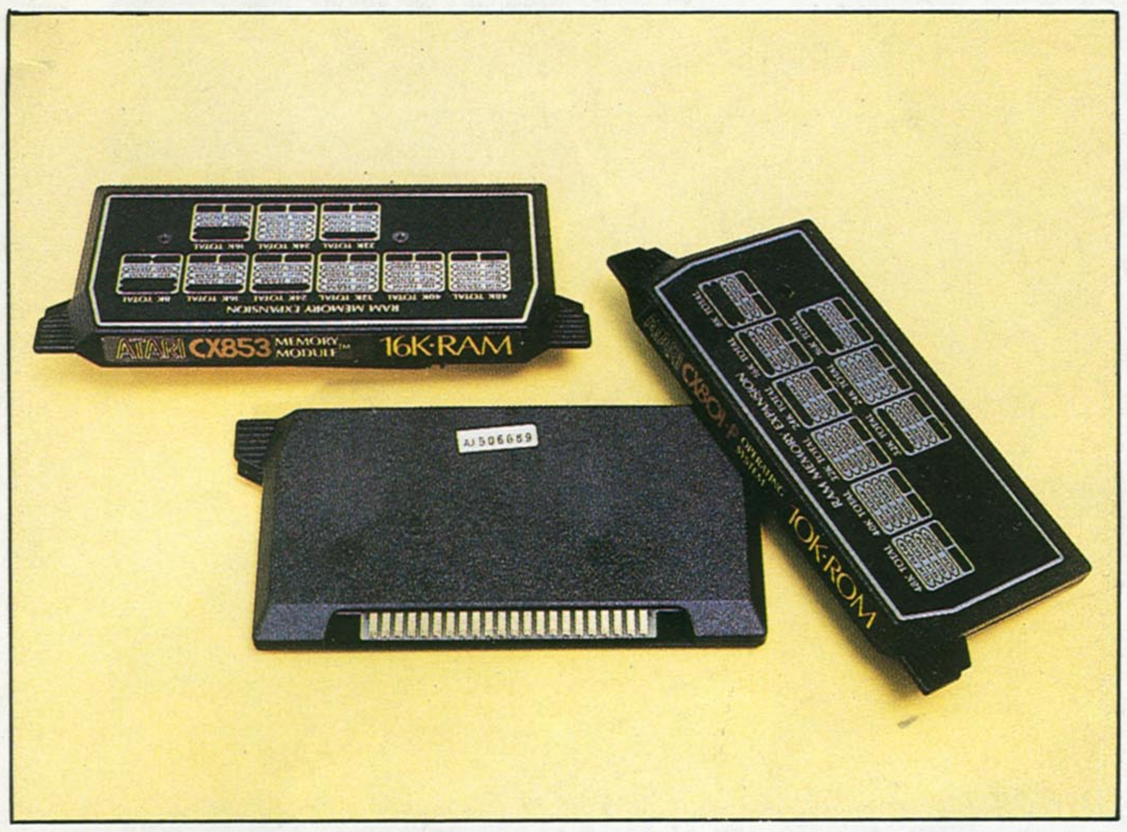
Enciclopedia Práctica de la

INFORMATICA

LA MEMORIA PRINCIPAL DE LOS MICROORDENADORES / HARDWARE : HP-86 / EL LENGUAJE BASIC (y 3) BASE DE DATOS CONTROL







Nueva Lente / Ingelek

125 ptas.

LA MEMORIA PRINCIPAL DE LOS MICROORDENADORES

LA memoria principal de los microordenadores está dividida en dos unidades de almacenamiento de información: *la memoria RAM* y *la memoria ROM.*

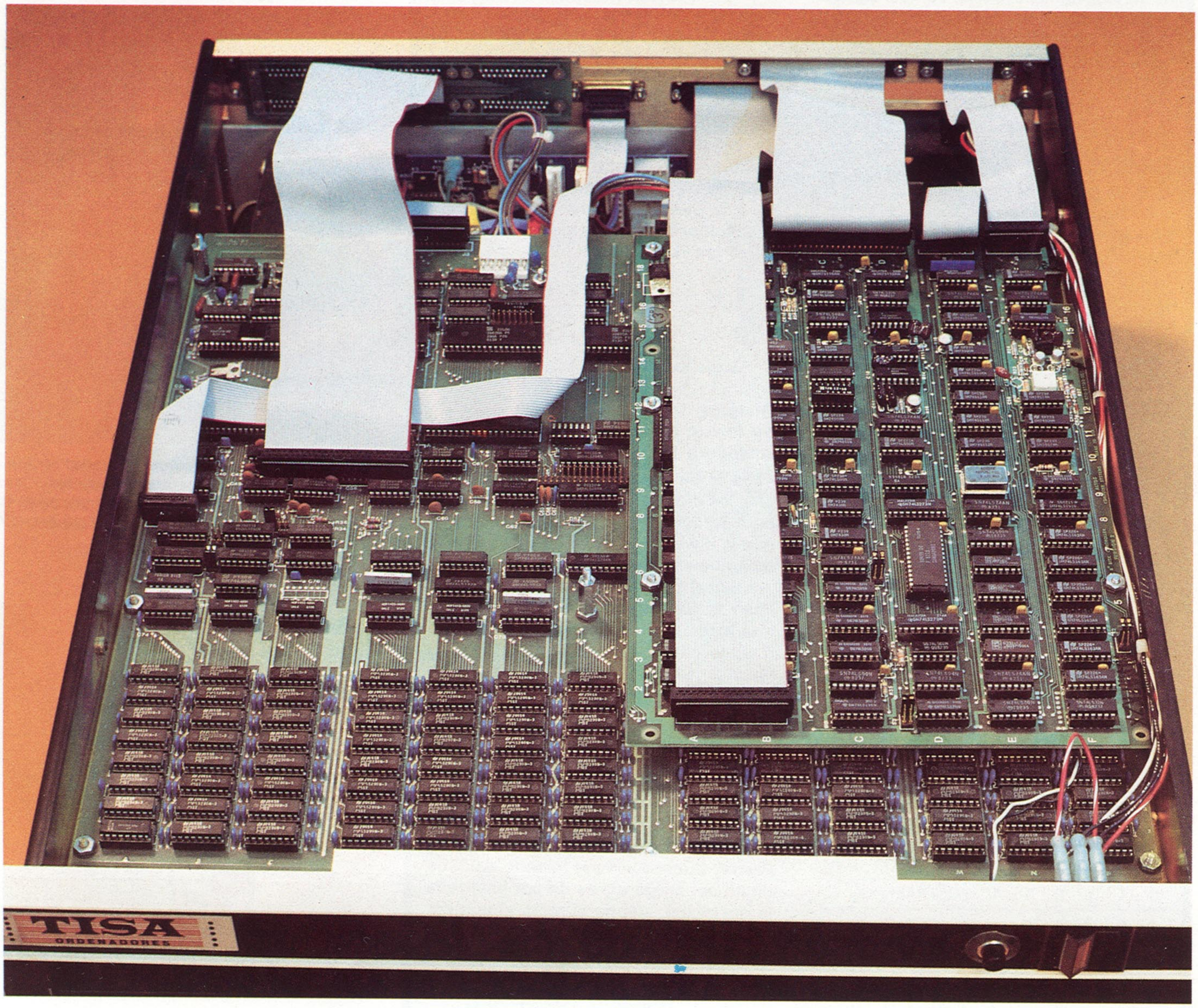
El concepto de memoria se aplica a todo dispositivo electrónico que pueda almacenar información. De esta forma, consideramos como memoria de un ordenador tanto a la memoria central utilizada por la CPU para la ejecución de programas, como a la auxiliar que servirá para almacenar información de forma masiva. La diferencia principal entre estas dos clases de memorias es que la auxiliar no puede ser procesada directamente por la CPU; por ejemplo, si disponemos de un programa almacenado en un diskette (memoria auxiliar), y queremos ejecutarlo, es imprescindible cargarlo previamente en la memoria principal.

En este capítulo nos ocuparemos del estudio de la memoria principal, tanto de la zona RAM como ROM, concretando los ejemplos al caso de los sistemas microordenadores.

Memoria RAM

La memoria RAM (Random Acces Memory) se suele denominar también memoria de lectura/escritura (R/W, Read/Write), ya que en ella se puede leer o escribir información, indistintamente.

Los medios de comunicación de la memoria con la CPU (por ejemplo, microprocesador) son el *bus de direcciones,* mediante el cual se apunta a la dirección de memoria que ocupa o va a ocupar la información, y el bus de da-



La memoria principal de los sistemas microordenadores consta de dos zonas de almacenamiento: memoria ROM y memoria RAM.

LA MEMORIA PRINCIPAL DE LOS MICROORDENADORES

Conceptos básicos

Aritmética binaria

En este apartado vamos a describir las cuatro operaciones aritméticas básicas: adición, sustracción, multiplicación y división, con el sistema de numeración binario. De esta forma obtendremos una idea de cómo funciona la unidad aritmético-lógica de la CPU en su componente aritmética.

Adición binaria

La suma binaria se realiza dígito a dígito, según la siguiente tabla:

PRIMER SUMANDO	SEGUNDO SUMANDO	RESULTADO	ACARREO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Al igual que en la suma decimal «7 + 8 = 5 más un dígito de acarreo para la cifra situada a la izquierda», en la operación binaria equivalente, se verifica que «1 + 1 = 0 y acarreamos 1 para la cifra siguiente»; el fundamento de este resultado es que 1 + 1 = 2 que en binario es 10.

1	1	0	1	1	0	ACARREO
	1	1	0	1	1	0 SUMANDO-1
	1	1	0	1	1	1 SUMANDO-2
1	1	0	1	1	0	1 RESULTADO

Luego, 110110 + 110111 = 1101101.

Sustracción binaria

El procedimiento es semejante al utilizado en aritmética decimal, si bien la resta binaria se ajusta a la siguiente tabla:

MINUENDO	SUSTRAENDO	RESULTADO	PENDIENTE
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

La cantidades pendientes se suman a la siguiente cifra del sustraendo. Por ejemplo:

1	0	1	1	0	1	MINUENDO
0	1	1	0	1	1	SUSTRAENDO
1	0	0	1	0		PENDIENTE
0	1	0	0	1	0	RESULTADO

Con lo que: 101101 – 011011 = 10010.

Multiplicación binaria

La tabla de multiplicar para un único bit es la siguiente:

MULTIPLICANDO	MULTIPLICADOR	RESULTADO
0	0	0
0	1	0
1	0	0
1	1	1

El procedimiento para multiplicar números binarios de más de un bit es similar al utilizado comúnmente para la multiplicación de números decimales:

Paso 1: Efectuar el producto de la cifra menos significativa del multiplicador por todas las del multiplicando.

Paso 2: Repetir el Paso 1 para todas las restantes cifras del multiplicador, escribiendo el resultado parcial, desplazado un lugar hacia la izquierda, debajo del resultado anterior. Si el bit tratado del multiplicador es 0, omitir la operación y desplazar una posición adicional el próximo resultado parcial.

Paso 3: Cuando no queden más dígitos del multiplicador, efectuar la suma de todos los resultados.

```
                1 0 1 1 0 1
          ×     1 0 1 1
        ---------------------
                1 0 1 1 0 1
              1 0 1 1 0 1
          1 0 1 1 0 1
        ---------------------
          1 1 1 1 0 1 1 1 1
```

División binaria

Se realiza aplicando un procedimiento análogo a la división decimal.

Paso 1: Se toman, de izquierda a derecha, los dígitos necesarios del dividendo para formar un número mayor o igual que el divisor, a este número le llamaremos número intermedio, y se coloca un + 1 en el resultado.

Paso 2: Se resta el divisor del número intermedio y se añade al resultado la siguiente cifra del dividendo.

Paso 3: Si el número obtenido en el Paso 2 es mayor que el divisor, se coloca un «1» en el resultado y se repite el Paso 2, en caso contrario se coloca «0» en el resultado, se añade una nueva cifra del dividendo y se repite el Paso 2. En ambos casos, el nuevo número intermedio es el resultado de la última resta y las cifras añadidas.

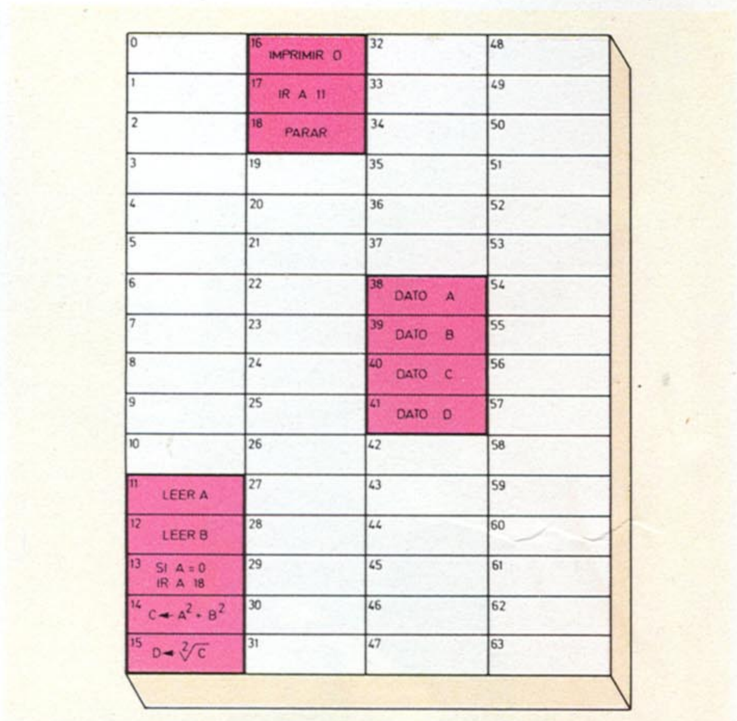
Paso 4: Cuando se haya tomado el último dígito del dividendo se da por finalizada la operación, siendo el resto el número intermedio si éste es inferior al divisor, si no lo es se repite una vez más el proceso.

```
1 0 1 0 0 0 | 1 1 0
0 1 1 0     ---------
  1 0 0 0     1 1 0
  0 1 1 0
    0 1 0 0
```

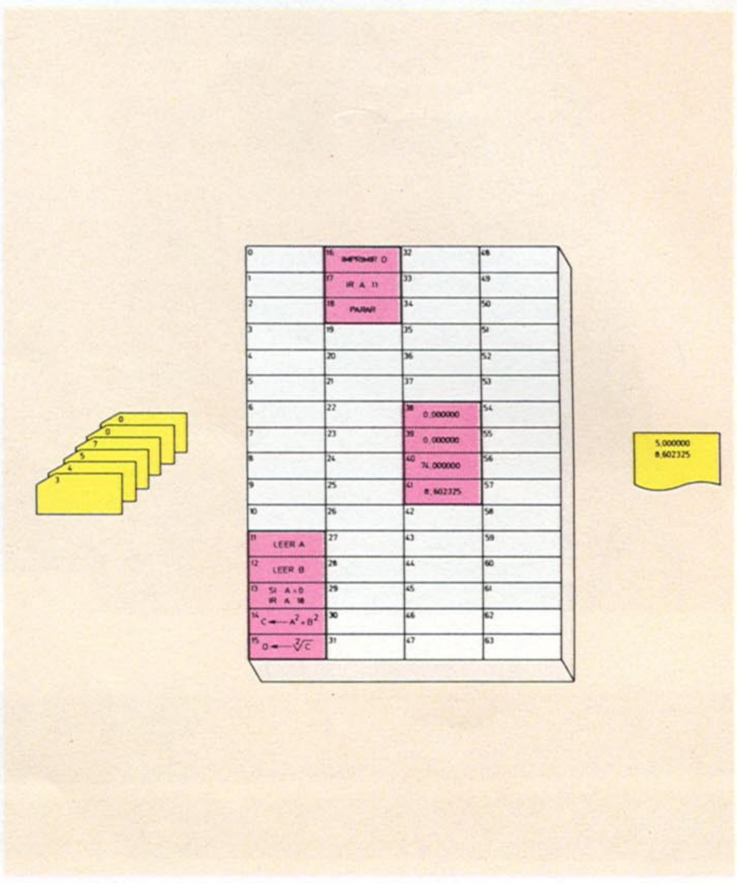
tos, a través del que se transmite la instrucción o dato apuntado por el bus de direcciones. Esta transmisión puede efectuarse en los dos sentidos, es decir, desde el exterior hacia la memoria o desde la memoria hacia el exterior.

La forma de determinar si la información va a ser leída o escrita en memoria es a través de una señal de control con dos estados posibles: uno implica lectura y otro escritura. En ambos casos la operación se realizará a través del bus de datos. Otra señal de control autorizará o no la utilización de la memoria.

Todas las operaciones con la memoria



Representación de la memoria principal de un ordenador en la que está almacenado un programa para el cálculo de la hipotenusa de un triángulo rectángulo en función de los catetos.

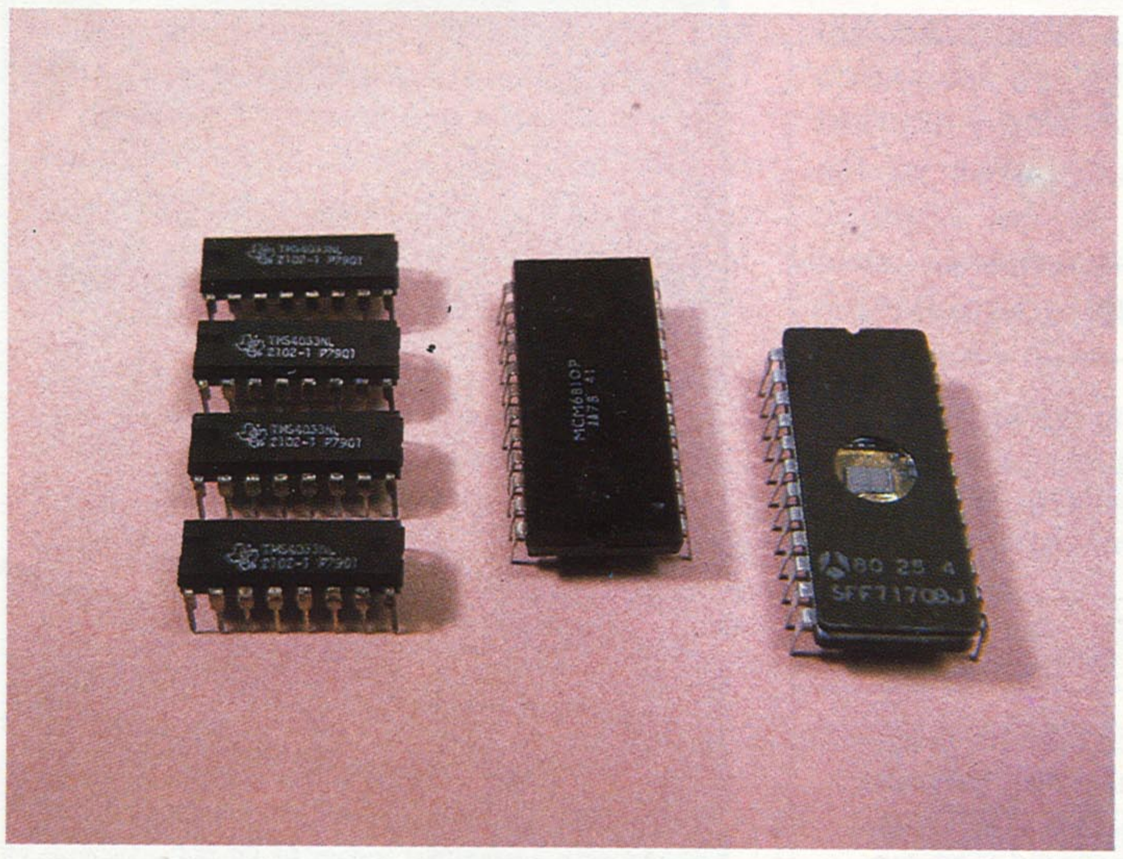


Después de ejecutar el programa anterior con los datos reflejados en las fichas (a), la memoria queda tal como se indica en el gráfico (b). La zona (c) muestra el correspondiente listado final.

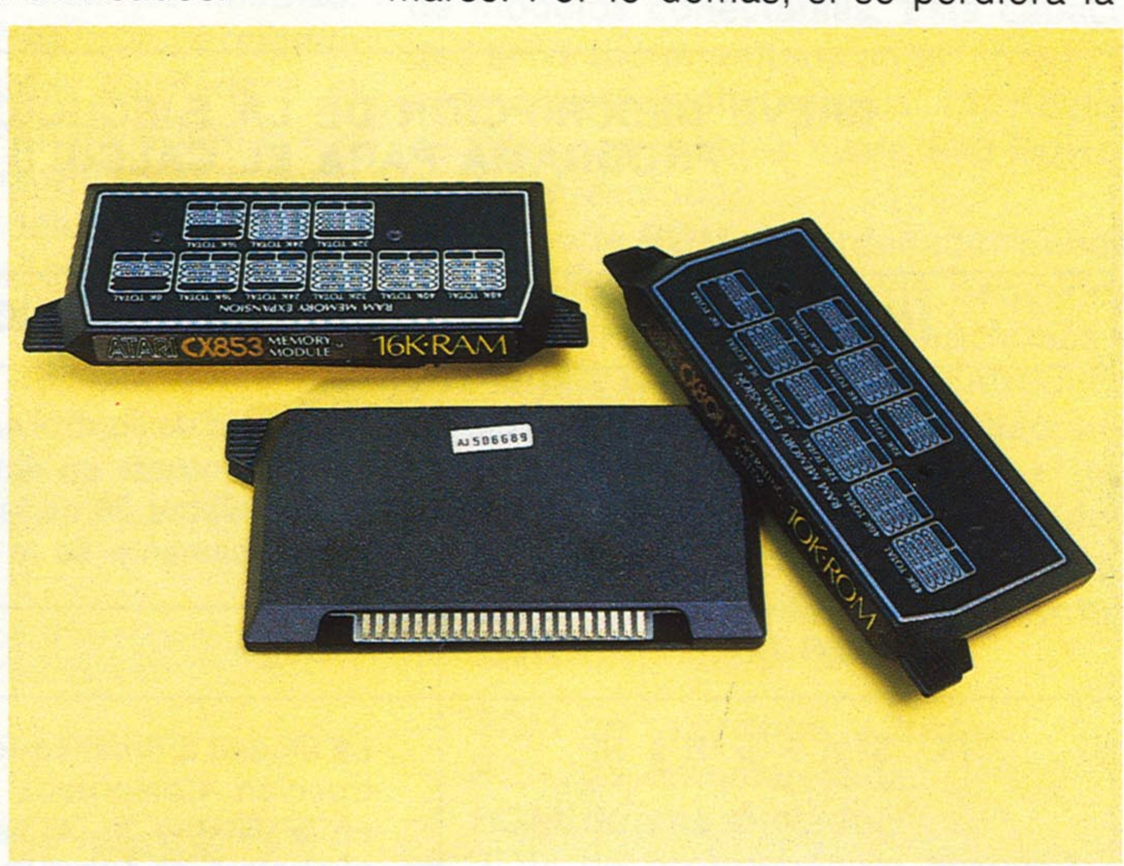
están controladas por la unidad de control integrada en la CPU (microprocesador).
Dentro del ordenador o microordenador, la memoria RAM se utilizará tanto para almacenar programas y datos como para guardar resultados intermedios.
Otra característica de la memoria RAM es su volatilidad; la falta de alimentación eléctrica hace desaparecer toda la información que estuviera almacenada en ella. Esto no debe suponer un grave problema, puesto que el usuario debe tener almacenados sus programas y datos en una memoria auxiliar no volátil (cinta magnética, diskette, etc.), de forma que el único riesgo ante una falta de energía se reduzca a la pérdida de las modificaciones efectuadas durante la sesión en curso.

Memoria ROM

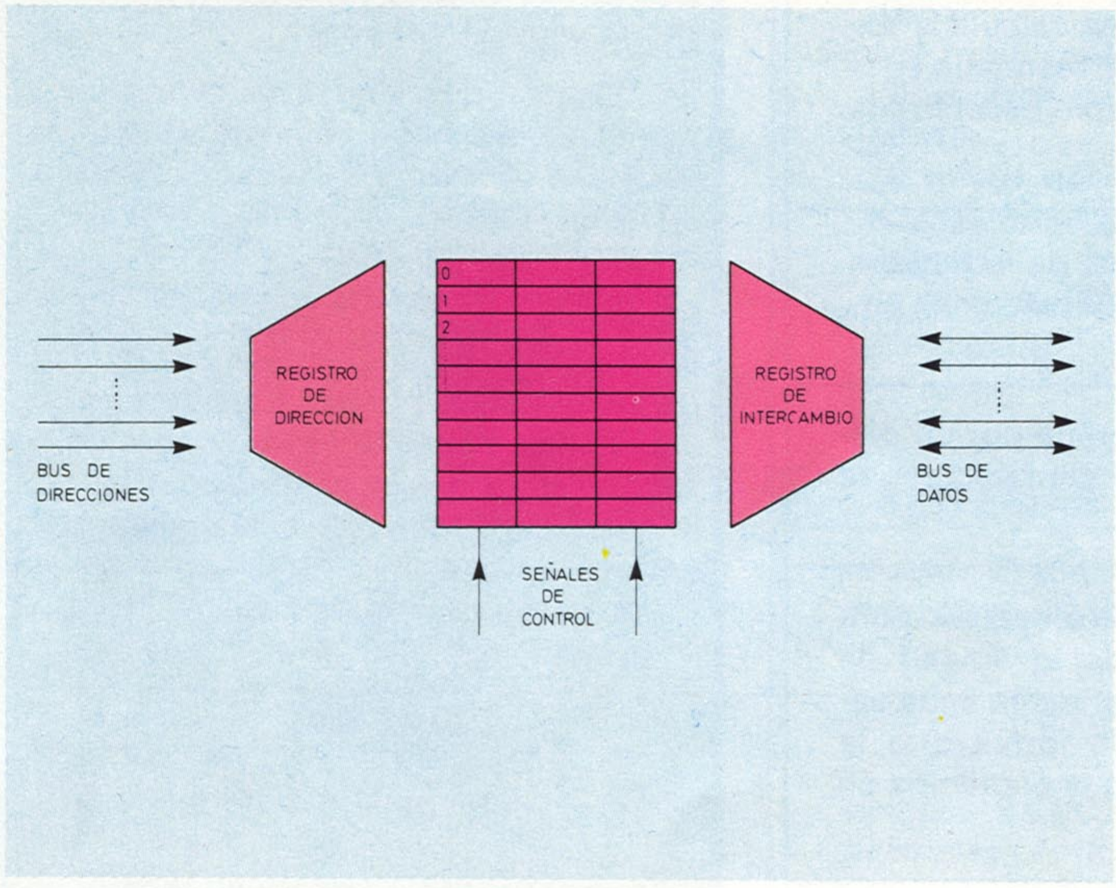
La memoria ROM (Read Only Memory) sólo permite la operación de lectura, de forma que los programas grabados en ella por el fabricante pueden ser utilizados, pero nunca modificados.
La comunicación con el microprocesador se efectúa, al igual que en las memorias de tipo RAM, a través de los buses de direcciones y datos. No obstante, en este caso, el bus de datos sólo permite la salida de información desde la memoria hacia el exterior y no al revés.
La señal de control sólo interviene para autorizar la utilización de la memoria ROM.
Evidentemente, las memorias de este tipo no son volátiles dado que su contenido es fijo y no puede reprogramarse. Por lo demás, si se perdiera la



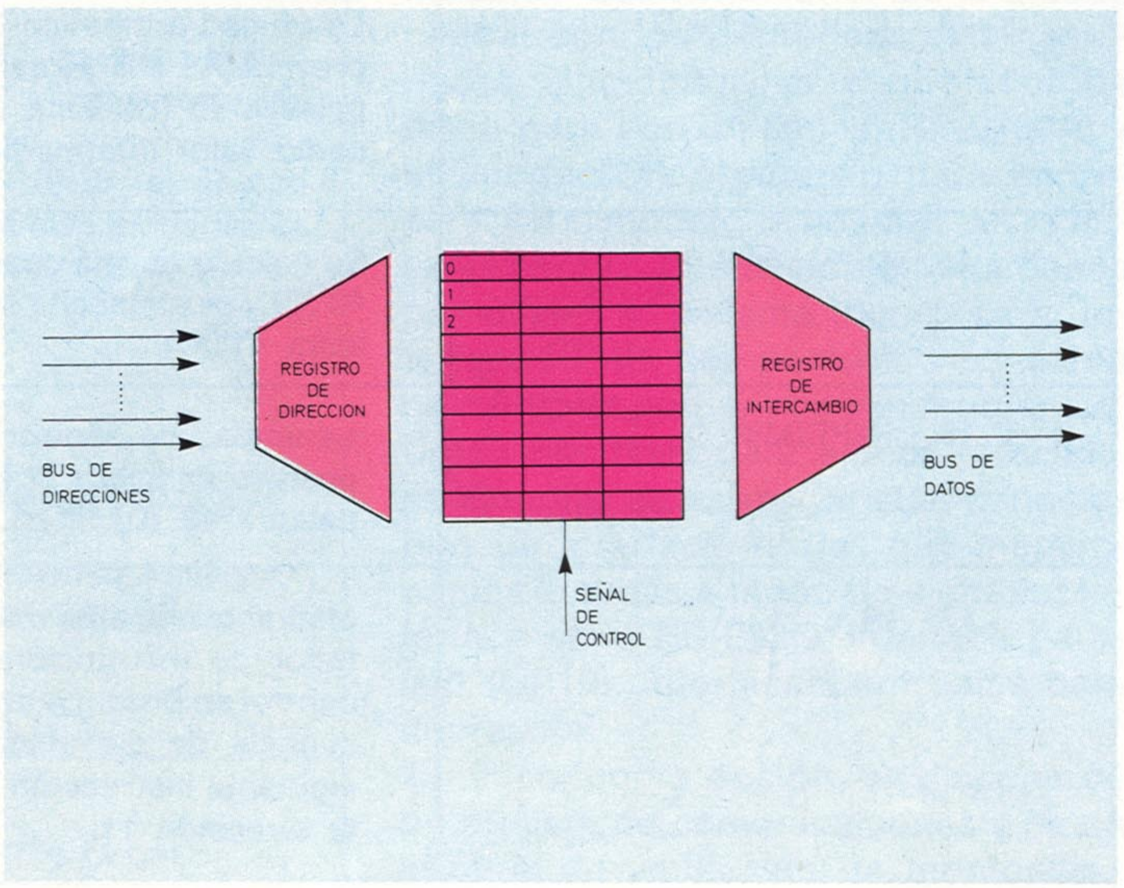
La zona de memoria de los modernos sistemas microordenadores está constituida a base de la asociación de circuitos integrados de memoria cuya capacidad por chip puede variar desde algunos bytes hasta varias decenas de Kbytes.



Algunos microordenadores permiten la ampliación de la memoria principal por medio de la incorporación de módulos exteriores de memoria RAM o ROM.



Estructura típica de una unidad de memoria RAM. Obsérvese que en este tipo de memorias (de lectura y escritura) los datos pueden entrar o salir de la unidad a través del correspondiente bus de datos.



Organización de una unidad de memoria ROM. Dada su característica de memoria de solo lectura, los datos se canalizan sólo en un sentido: del interior de la memoria hacia el exterior a través del bus de datos.

LA MEMORIA PRINCIPAL DE LOS MICROORDENADORES

información contenida en ellas, quedarían inutilizables.

La memoria principal de un microordenador puede ampliarse, incrementando el número de unidades conectadas. La única limitación consiste en la capacidad de direccionamiento del bus de direcciones. Normalmente, los microprocesadores de 8 bits (CPU de los microordenadores de 8 bits) suelen disponer de 16 líneas en dicho bus, con lo que pueden llegar hasta $2^{16} = 65.536$ direcciones distintas. Como cada dirección está ocupada por una palabra de 8 bits (byte), el tope máximo de ampliación de este tipo de microordenadores es de 64 Kbytes ($64 \times 1.024 = 65.536$).

Otro indicador importante para caracterizar una memoria es el *caudal*, así se denomina al número máximo de informaciones leídas o escritas en la memoria por unidad de tiempo. La unidad más usual para medir el caudal es el Kilobyte por segundo o el Megabyte por segundo. Por ejemplo, podemos hablar de un caudal de 15 Kilobytes por segundo, si el bus de datos puede transmitir $15 \times 1.024 = 15.360$ bytes por segundo.

BREVE DESCRIPCION DE LA EJECUCION DEL PROGRAMA PARA EL CALCULO DE HIPOTENUSAS

INSTRUCCION	EJECUCION
11. LEER A	La unidad de control acepta por el dispositivo de entrada un dato que se almacena en la palabra 38 de la memoria (a la que simbólicamente se la llama A).
12. LEER B	Análogo.
13. SI A = 0 IR a 18	La unidad aritmético-lógica ejecuta la instrucción. Comprueba si el valor cargado en la palabra 38 (A) es igual a 0. En caso afirmativo, se altera la secuencia y se ejecuta la instrucción 18; en caso negativo, se continúa en secuencia.
14. C ← $A^2 + B^2$	La unidad aritmético-lógica calcula la expresión ($A^2 + B^2$) y carga el resultado en la palabra 40 (C). Esta variable es utilizada como valor intermedio.
15. D ← $\sqrt{C}$	Se calcula la raíz cuadrada de la palabra 40 (C) y se almacena el resultado en la palabra 41 (D).
16. IMPRIMIR D	La unidad de control imprime por un dispositivo de salida el valor contenido en la palabra 42 (D).
17. IR a 11	Mediante una alteración en el registro contador de instrucciones de la unidad de control se produce una variación en la secuencia de ejecución, de forma que la siguiente instrucción será la contenida en la dirección 11.
18. PARAR	La unidad de control da por terminada la ejecución del programa, con lo que la memoria utilizada queda libre.

Glosario

¿Se puede ejecutar directamente un programa almacenado en una cinta magnética?

No. La intervención de la memoria principal es esencial para la ejecución de un programa, por tanto, habrá que pasarlo previamente de la memoria auxiliar (cinta magnética) a la memoria principal para que sea posible su ejecución.

¿Qué tipo de información se almacena en la memoria RAM?

Tantos datos como instrucciones, ya que este tipo de memoria está reservada para el usuario que puede escribir y leer en la misma a través de los programas que construya.

¿Se puede utilizar la memoria ROM para almacenar datos?

No. Unicamente puede contener los programas cargados por el fabricante (sistema operativo, programas de utilidad, etc.) y algunos datos fijos. El usuario sólo puede utilizar dichos programas (leerlos), pero nunca modificarlos.

¿En qué consiste la volatilidad de la memoria?

En la pérdida de su contenido ante la falta de fluido eléctrico. Las memorias RAM son volátiles, pero la ROM y la mayoría de las memorias auxiliares no lo son.

¿Se puede ampliar la memoria principal de un microordenador?

Sí. Siempre que el bus de direcciones pueda apuntar hasta la última palabra de la unidad de memoria utilizada en la ampliación.

¿A qué se denomina caudal?

A la cantidad de información por unidad de tiempo que se puede transmitir entre la memoria y la CPU. Las unidades utilizadas para representar el caudal son el Kilobyte por segundo y el Megabyte por segundo.

HP-86

DENTRO de la gama de ordenadores personales de la Serie 80 de Hewlett Packard, el HP-86 se presenta como un modelo monousuario, orientado principalmente hacia la resolución de problemas de cálculo e ingeniería, en los que el fabricante ha demostrado una buena reputación. No obstante, debido a que es capaz de trabajar con el sistema operativo CP/M (que se presenta como opción), las posibilidades de aplicación se ven notablemente incrementadas, al existir una importante biblioteca de programas compatibles con dicho sistema y que abarcan campos tales como: ventas, marketing, contabilidad, etc.

Hewlett Packard ha desarrollado el equipo HP-86, acogiéndose al criterio básico de *modularidad,* permitiendo al usuario la ampliación del mismo, conforme a sus necesidades y posibilitando la conexión de un gran número de tipos de periféricos.

Una característica fundamental que se ha de destacar en este equipo es la absoluta compatibilidad de software con el resto de los modelos de la serie: el HP-85 y HP-87.

El HP-86 se comercializa en dos versiones que reciben la denominación de HP-86A y HP-86B. En líneas generales, el HP-86B es de mayor potencia que el modelo A, al disponer de mayor capacidad de memoria, posibilidad de elección del teclado en distintos idiomas, bus de interface HP-IB, que permite la conexión de diferentes periféricos al sistema y que es compatible con el estándar de la industria IEEE-488 (siendo opcional en el modelo A), amén de otras características que veremos inmediatamente al compararlo con el modelo A. De no mencionarse específicamente, los datos ofrecidos corresponden a este modelo.

Ordenador: **HP-86.**
Fabricante: **Hewlett-Packard.**
Nacionalidad: **Estados Unidos.**
Distribuidor: **Hewlett Packard.**

CARACTERISTICAS BASICAS

UNIDAD CENTRAL	MEMORIAS DE MASA
CPU: Procesador de 8 bits (propio de HP). *RAM versión básica:* 64 Kbytes (HP-86A), 128 Kbytes (HP-86B). *ROM versión básica:* 48 Kbytes (HP-86A), 56 Kbytes (HP-86B). *Máxima RAM (con ampliación):* 576 Kbytes (HP-86A), 640 Kbytes (HP-86B). *Accesos periféricos básicos:* Interface Centronics para impresora y 2 HP Disc Drive. *Interfaces especializadas (opcionales):* Interface BCD, GPIO, RS-232C, HP-IB y HP-IL.	*Discos flexibles:* Dos unidades para discos de 5 y 1/4" o para microfloppies de 3 y 1/4 pulgadas, de 270 Kbytes por disco. *Discos rígidos:* Unidad de disco rígido Winchester de 4,6 Mbytes o de 9,6 Mbytes. (A través del bus HP-IB, el equipo puede conectarse a más unidades de disco de los tipos indicados o a otros de la gama HP.)
TECLADO	**SISTEMAS OPERATIVOS**
Versión estándar: Teclado QWERTY de 91 teclas, con zona de teclas funcionales y de control del cursor.	*Estándar:* Propio de Hewlett Packard. *Opcionales:* CP/M y USCD p-System.
PANTALLA	**LENGUAJES**
Versión estándar: Monocroma de fósforo verde. *Formato de presentación:* 16 ó 24 líneas de 80 caracteres. *Resolución gráfica:* 400 ó 544 × 240 puntos.	*Estándar:* BASIC. *Opcionales:* Fortran-77 y Cobol.

Unidad central de proceso

La unidad central de proceso está constituida por un procesador de 8 bits específico de Hewlett Packard. Dispone además de 48 Kbytes de memoria ROM estándar (donde reside el S. O. y el lenguaje BASIC ampliado) y de 64 Kbytes de RAM destinados al usuario y que son ampliables hasta 576 Kbytes por medio de módulos conectables. Esta capacidad se ve incrementada en el modelo B hasta los siguientes valores: 128 Kbytes de RAM, ampliables hasta 640 Kbytes de ROM estándar.

La memoria ROM del ordenador puede ser ampliada por medio de pequeños módulos independientes conectables a una tarjeta que es la que se introduce en una de las «ranuras de ampliación» situadas en la parte posterior del equipo. Cada uno de estos módulos tiene asignada una función operativa concreta. Por ejemplo, algunos de los módulos disponibles son:

— ROM de impresión y trazado.
— ROM de entrada/salida.
— ROM de ensamblador.
— ROM de programación avanzada.
— ROM de matrices.

Cada tarjeta admite un máximo de 6 módulos. Las conexiones de la unidad central con los periféricos están asignadas al monitor, al controlador de disco para dos unidades (HP-9130A) y al interface tipo paralelo para impresora. En el modelo B, además del monitor —para comunicación directa con la pantalla—, el sistema incorpora el interface HP-IB, que permite la conexión de múltiples periféricos o instrumentos. Como opciones para ambos modelos, se dispone de varias tarjetas conectables de interface, de las que mencionamos algunas a modo de ejemplo: Interface serie RS-232C, Interface paralelo (GP10), Interface Centronics para impresora.

También, como opción, se dispone de un modem de comunicaciones y de un sistema de emulación de terminales.

Teclado

El teclado, que es solidario con la uni-

HP-86

dad central, dispone de 91 teclas entre las que se encuentran: 4 para el movimiento del cursor, borrado e inserción de caracteres, teclado numérico independiente y teclado alfanumérico.
Otra característica destacable es que dispone de teclas de «comandos del sistema», que permiten la introducción de comandos comúnmente utilizados como son PAUSE, RUN o LIST, con la pulsación de una sola tecla.
Además de las teclas de comandos, existen en la zona superior una red de teclas «programables» por el usuario que permiten definir las instrucciones que el operador desee e introducirlas posteriormente con la acción sobre una sola tecla.
En el modelo HP-86B se ofrece la posibilidad de elegir el teclado en distintos idiomas, entre ellos el español.

Pantalla

El fabricante ofrece dos monitores diferentes para conectar a la unidad central, que se diferencian en el tamaño de la pantalla: el HP-82912, de 9 pulgadas (23 cm.) y el HP-82913, de 12 pulgadas (30 cm.).
La capacidad de representación en modo alfanumérico es de 16 ó 24 líneas × 80 columnas, con caracteres definidos por una matriz de 5 × 7 puntos. En modo gráfico, la definición es de 400 ó 544 × 240 puntos. La capacidad de movimiento vertical de la imagen (*scrolling*) es de 54 ó 204 líneas y dispone de un conjunto de 256 caracteres: un juego de 128 más el mismo número para modo de video inverso.
El monitor dispone de una memoria RAM interna de 16 Kbytes y la representación se realiza en color verde sobre fondo negro.

Memorias de masa

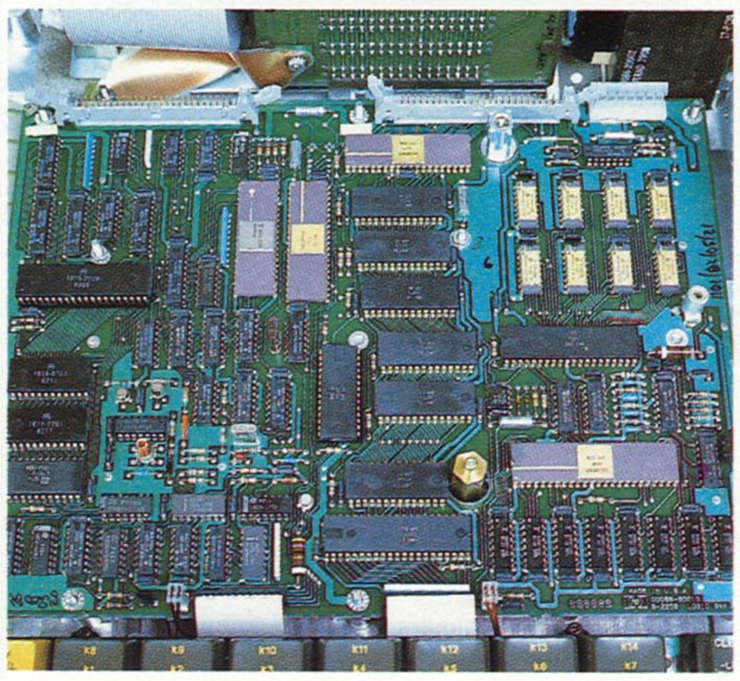
El HP-86 puede soportar tanto discos flexibles como discos rígidos de tipo Winchester.
Incorpora el controlador para la unidad de disco flexible HP-9130A de 5 1/4 pulgadas y 270 Kbytes de capacidad, que le permite la conexión de dos unidades. La velocidad de transferencia es de 620 bytes/seg. y el tiempo de acceso varía entre 187 ms. y 435 ms. (dependiendo de que el disco se encuentre girando o no).
Mediante el interface HP-82937A (HP-IB), el sistema puede controlar además otros dos tipos de unidades de disco flexible: la HP-9121D y la HP-9121S, que se diferencian exclusivamente en la capacidad de almacenamiento (540 Kbytes para la primera y 270 Kbytes para la segunda); la velocidad de transferencia es de 980 bytes por seg. y el tiempo de acceso varía entre 415 y 1.415 ms.
Con este mismo interface, se puede conectar la unidad de disco Winchester HP-9133B, de 4,6 Mbytes de capacidad y con una velocidad de transferencia de 1.400 bytes por segundo; el tiempo de acceso está comprendido entre 60 y 173 ms. Se ofrece también la posibilidad de conectar una unidad de disco Winchester de 9,6 Mbytes, si se dispone de los módulos: «Electronic Disc ROM» (incluido en el HP-86B) y del «ROM Drawer» (que proporciona todos los comandos necesarios para el control de la unidad).

Periféricos

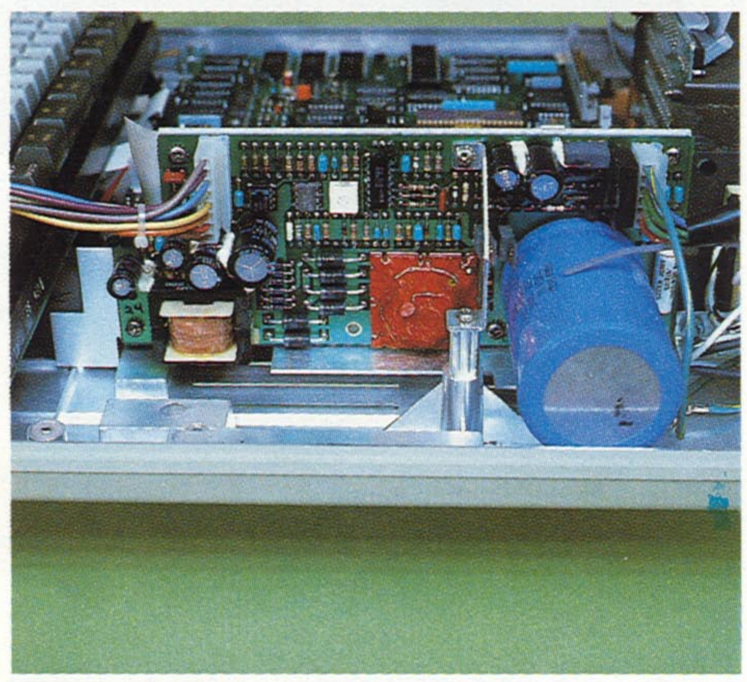
El HP-86A está preparado para controlar una impresora de tipo paralelo de aplicación general, aunque con el interface opcional HP-IB puede controlar un gran número de dispositivos periféricos tales como: *plotters,* digitalizadores o terminales de comunicaciones. Concretamente, a través del interface HP-IB, el HP-86A puede conectarse a un máximo de 12 × 4 dispositivos periféricos y el modelo B a un total de 12 × 5 unidades periféricas.
Entre las características propias de al-



El HP-86 es uno de los ordenadores personales de la serie 80 de Hewlett Packard. Se trata de un sistema monousuario orientado básicamente a aplicaciones de cálculo e ingeniería.



La unidad central del HP-86 está organizada en torno a un procesador de diseño exclusivo de la firma Hewlett Packard.



Este procesador permite el acceso a un máximo de 576 Kbytes, de RAM en la versión A y de 640 Kbytes de RAM en el modelo B.

gunos periféricos podemos citar, para la impresora de aplicación general HP-82905B, una velocidad de impresión de 80 c/s sobre 80 columnas, método de impresión por impacto y capacidad de plasmar representaciones gráficas. Para el plotter HP-7470A: dos puntos de impresión y capacidad para 16 colores. Y para el modem de comunicaciones HP-82950A: 3 modos de transmisión (serie, asíncrona o duplex), a una velocidad de 300 bits por seg., protocolo XON/XOFF y método de modulación binaria de fase coherente (FSK).

Software

La familia HP-86 emplea un sistema operativo propio) almacenado en los 48 Kbytes de ROM) con lenguaje BASIC incorporado; si bien opcionalmente puede trabajar con CP/M (Control Program for Microcomputers) o UCSD p-System.

El UCSD p-System es un completo sistema de desarrollo software, con el que se pueden emplear como lenguajes de programación el UCSD Pascal y el FORTRAN 77.

Cabe mencionar que el empleo de los sistemas operativos opcionales es excluyente con el original y, por tanto, la implantación de uno de ellos implica que no se puede utilizar el sistema operativo propio. Esto trae como consecuencia que algunos de los interfaces no pueden ser conectados al emplear cualquiera de los sistemas operativos opcionales. Como ejemplo, diremos que empleando CP/M, ninguno de los módulos ROM conectables puede ser utilizado, puesto que son incompatibles. Si se desea, por tanto, un sistema operativo opcional, deberemos contemplar estas limitaciones.

Otra característica asociada al empleo del CP/M es la necesidad de conectar un módulo exterior que contiene todo el hardware necesario para operar con este sistema. El módulo en cuestión es controlado por un microprocesador Z-80A, asociado a 64 Kbytes de RAM, que funciona como microprocesador independiente.

Software de aplicación

El software propio desarrollado para el HP-86 por Hewlett Packard se ve notablemente incrementado al ser el sistema compatible con toda la serie 80. A este repertorio cabe añadir todo el software escrito para CP/M, reuniéndose, por tanto, una extensa biblioteca



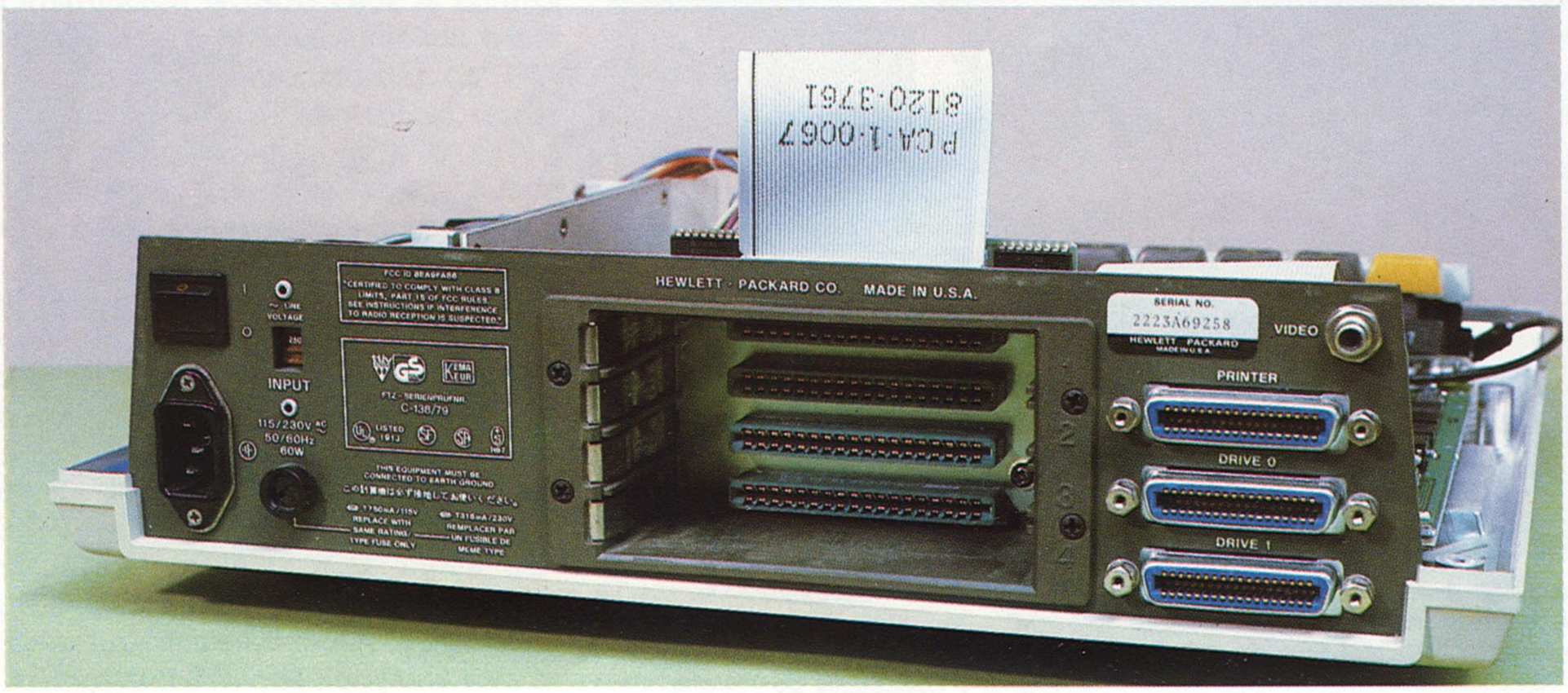
El sistema se presenta en dos versiones: HP-86A y HP-86B, siendo este último modelo de mayor capacidad de memoria y con mayor potencia operativa.



El teclado dispone de un total de 91 teclas, distribuidas en teclado alfanumérico, teclado numérico decimal, teclas para el control del cursor y zona de teclas funcionales.



La versión básica del equipo se presenta con una estructura muy compacta, si bien, dispone al tiempo de muy amplias posibilidades de expansión.



Las posibilidades del equipo pueden ampliarse conectando diversos módulos de ROM con software especializado a una tarjeta de expansión «enchufable» a los conectores dispuestos en la zona posterior del mueble de la unidad central.

HP-86

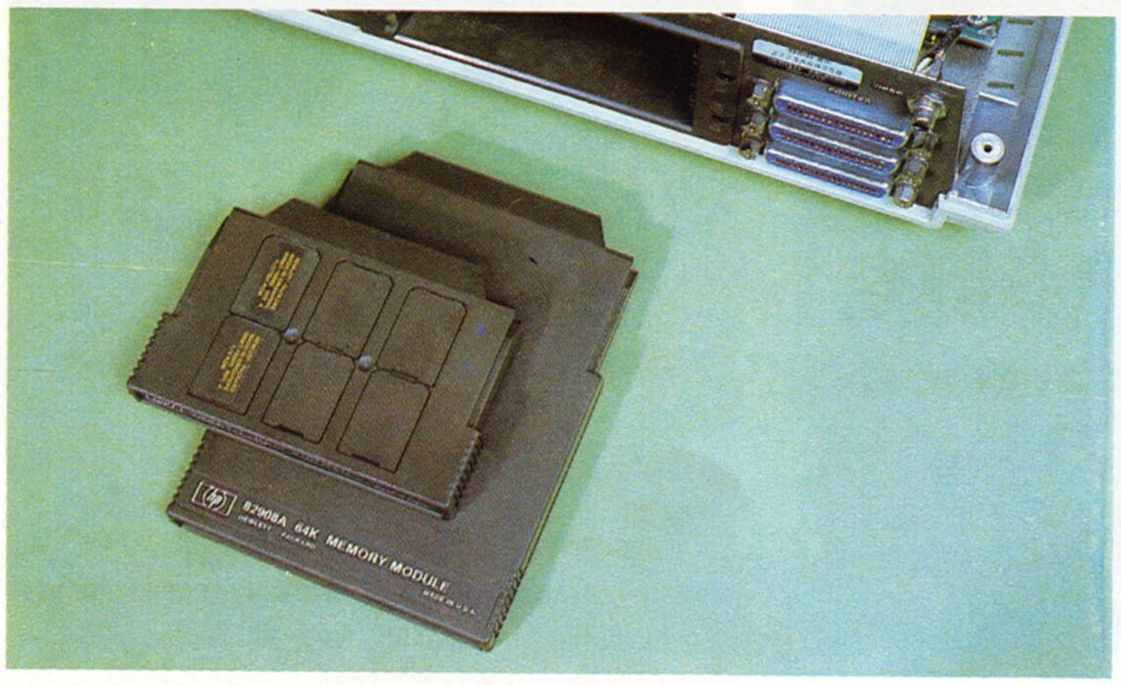
de programas de utilidad. De todos ellos, mencionaremos algunos a título de ejemplo:

— Graphics Presentations: creación de gráficos profesionales.
— WORD 80: edición, formateado, impresión y almacenamiento de textos y documentos.
— Base II: control de bases de datos.
— Comercial Finance: análisis comercial y finanzas.
— CMS STEP 1: control de inventarios.
— Linear Programming: optimización de modelos de programación.
— Topography: creación de mapas topográficos en tres dimensiones.
— VisiCalc: análisis estadísticos y de problemas de cálculo y financieros.
— Math: cálculo matemático.

Soporte y distribución

La información suministrada junto con el equipo consiste en: un manual de introducción, un manual de operación y programación, un manual de bolsillo, un disco de demostración y una hoja de información y mantenimiento, además de otros accesorios de menor importancia. Se dispone también de un servicio de información de sistemas (SIS), que proporciona acceso telefónico con un ingeniero de sistemas al, al que se pueden realizar las consultas oportunas. También se publican periódicamente hojas de actualización de la información. Todos los accesorios opcionales se suministran con la adecuada información técnica.
Está abierta la posibilidad de contratar un servicio de reparación y mantenimiento. El período de garantía inicial del equipo es de noventa días.

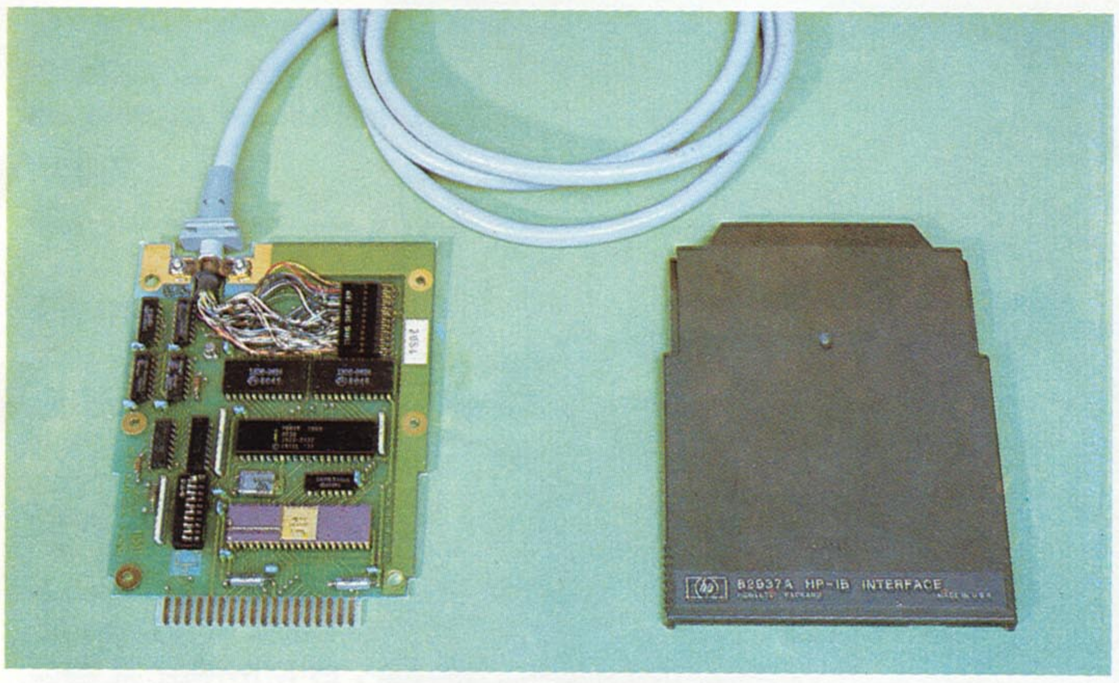
Las personas interesadas en este equipo deben dirigirse a los distribuidores oficiales de Hewlett Packard, donde podrán solicitar las pruebas y demostraciones necesarias.
Configuración básica: Unidad central con 48 Kbytes de ROM y 64 Kbytes de RAM. Teclado y monitor de 9 pulgadas con 16 Kbytes de RAM.
Configuración máxima: Unidad central con 128 Kbytes de ROM y 576 Kbytes de RAM. Teclado (en español para el HP-86B). Monitor, con 16 Kbytes de RAM, de 12 pulgadas monocromático. Impresora (HP-2601A, word processing), de 40 c/s (132 columnas), con interface serie RS-232C. Dos unidades de disco flexible de 270 Kbytes por disco y una unidad de disco Winchester de 4,6 Mbytes con módulo «Electronic disc ROM». Interfaces GP-10, BCD y Paralelo, para otros periféricos e interface general HP-IB.



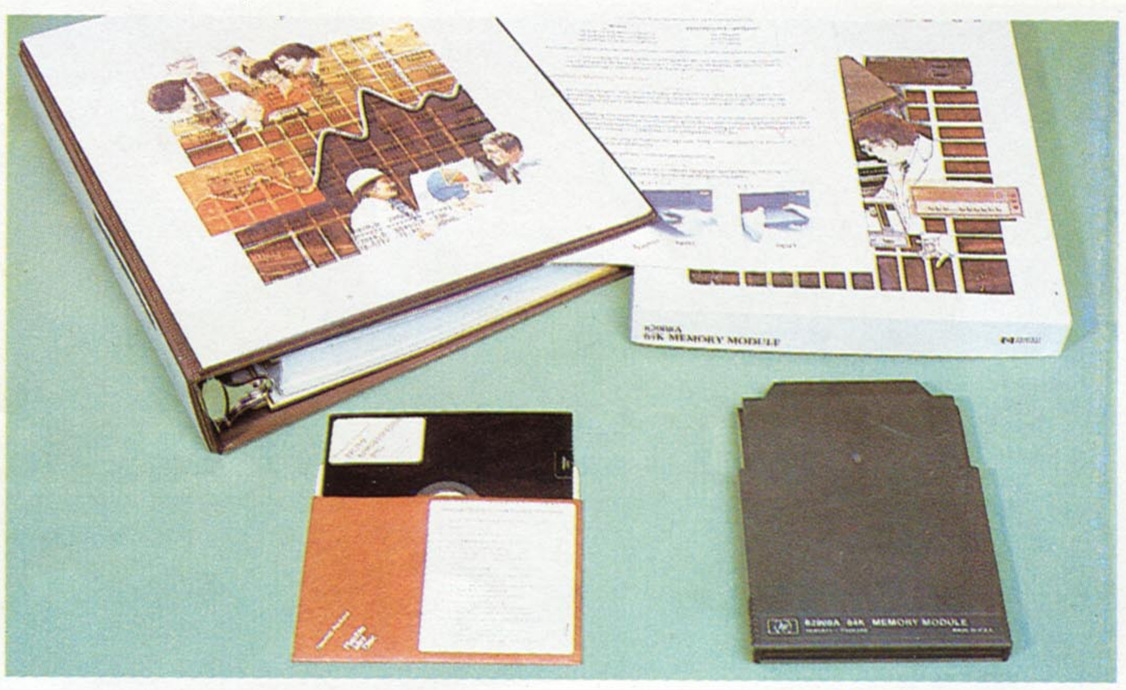
La expansión de la memoria principal por medio de módulos enchufables al equipo básico es una de las particularidades de los miembros de la serie 80 de HP. En la fotografía aparece un módulo de 64 Kbytes de RAM.



En la versión básica, el HP-86 puede controlar directamente a dos unidades de disco flexible de 5 y 1/4 pulgadas con capacidad de 270 Kbytes por disco.



Las posibilidades de adaptación de equipos periféricos se disparan con la conexión al HP-86 del módulo de interface HP-IB. Este da acceso a un total 12 × 4 dispositivos periféricos en el modelo HP-86A y a 12 × 5 en la versión HP-86B.



La documentación que se entrega con el equipo se ve complementada por las notas periódicas de actualización editadas por Hewlett Packard.

EL LENGUAJE BASIC (y 3)

COMO ya sabemos, un bucle permite la repetición de un grupo de instrucciones. Si el número de repeticiones viene controlado por el valor de un índice, que se incrementa a partir de un valor dado (X1) hasta otro valor máximo (X2), podemos simplificar la programación empleando las instrucciones «FOR» y «NEXT».

La instrucción FOR repite la ejecución de las instrucciones que hay entre ella y la instrucción NEXT, cada vez con un valor del índice incrementado en un determinado valor de salto (STEP). El bucle concluye cuando el índice variable supera el valor máximo establecido (expresión aritmética que sigue al comando TO y establece el valor máximo del índice). Si se omite el incremento (valor de STEP), el intérprete asume que es una unidad (contador).

La instrucción NEXT indica el fin del bucle. La variable a usar es la misma que la empleada en la FOR correspondiente. Si no hay bucles anidados puede omitirse la variable.

El uso de las instrucciones FOR...NEXT evita programar directamente la creación del índice, su incremento y la comprobación del cumplimiento de la condición para abandonar el bucle.

Subprogramas

En BASIC el programador puede crear sus propios subprogramas, rutinas o subrutinas, mediante el uso de las instrucciones GOSUB y RETURN.

La instrucción GOSUB cede el control del programa a la línea indicada hasta que encuentra una instrucción RETURN. Entonces, el control se transfiere a la línea que sigue a la última GOSUB ejecutada. Por tanto, puede haber varios niveles de subrutinas, es decir, las subrutinas a su vez pueden llamar a otras subrutinas. El límite de anidamiento viene dado por la capacidad de memoria.

De todas formas hay que tener mucho cuidado con los solapamientos de subrutinas, ya que la ejecución de RETURN devuelve siempre el control a la instrucción siguiente a la última GOSUB ejecutada. También se puede programar una decisión múltiple mediante la instrucción ON, que tiene dos versiones: una permite ir a diversas subrutinas (ON...GOSUB) y la otra continuar en diversas líneas del programa principal (ON...GOTO). En ambos casos, el control se transfiere a la subrutina o línea indicada en el lugar que corresponde en ese momento a la expresión aritmética; así, si la expresión vale 1, continúa en la instrucción que está en la línea cuyo número es el primero de la lista, si 2, con la segunda, y así sucesivamente.

Este tipo de instrucción es útil para todos aquellos casos en los que parte del cálculo depende de un parámetro, como puede ser el cálculo del salario en función de la categoría, etc.

Entradas y salidas

Para que un programa sea útil es necesario que podamos introducir los datos al ordenador y obtener los resultados tanto por pantalla como por impresora.

La instrucción INPUT permite introducir datos desde el terminal, tanto de constantes numéricas como de cadenas, asignando dichas constantes a las variables indicadas en la propia instrucción INPUT.

Cuando el programa llega a una instrucción INPUT, aparece en la pantalla un mensaje (si lo hay), comenzando la entrada de datos en la posición en la



Tal como ya hemos indicado en los dos anteriores capítulos dedicados a este lenguaje, el BASIC es el lenguaje de alto nivel más extendido en el campo de los microordenadores, debido a su fácil aprendizaje, la existencia de intérpretes y la gran diversidad de programas disponibles.

EL LENGUAJE BASIC (y 3)

Conceptos básicos

Microprogramación

La unidad de control de la CPU de un ordenador tiene un componente, denominado *secuenciador,* que después de analizar el código de operación de la instrucción que se está procesando, genera una serie de órdenes, llamadas *microórdenes.* Las microórdenes se envían al conjunto de las unidades del ordenador, unidad aritmético-lógica, memoria, etc., para que ejecuten las distintas fases de la instrucción. Este secuenciador, denominado *secuenciador cableado,* está constituido por una serie de circuitos electrónicos.

La microprogramación consiste en reemplazar el secuenciador cableado por un secuenciador programado, que va a contener microprogramas, de tal forma que, normalmente, a cada instrucción del ordenador le va a corresponder un *microprograma.* La ejecución de este microprograma genera las microórdenes que gobiernan las instrucciones. El microprograma también posee sus instrucciones que reciben el nombre de microinstrucciones. Los ordenadores microprogramados reciben el nombre de ordenadores de lógica almacenada o de *lógica programada.*

Los microprogramas se almacenan en una memoria especial, llamada *memoria de control.* Esta debe poseer una velocidad de lectura muy superior a la de la memoria central, con objeto de que el microprograma se ejecute en el intervalo de tiempo correspondiente a los pocos accesos a la memoria central de la instrucción y sus operandos.

La *microprogramación* ofrece la posibilidad de modificar fácilmente el conjunto de instrucciones de un ordenador microprogramado, puesto que cambiar o añadir un microprograma es mucho más sencillo que cambiar los circuitos electrónicos del secuenciador cableado.

que está el cursor. Los elementos a introducir se separan con comas. En el diagrama que aparece en la tabla adjunta, la cadena implica los posibles mensajes al operador. Las instrucciones PRINT y LPRINT hacen que, respectivamente, se muestren datos en la pantalla o se impriman los resultados en la impresora. Ambas instrucciones operan de la misma forma.

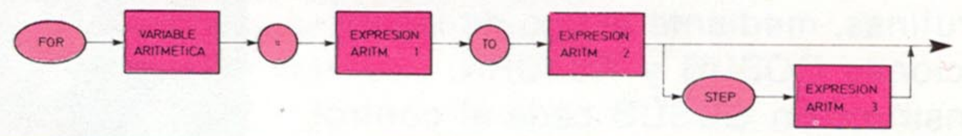
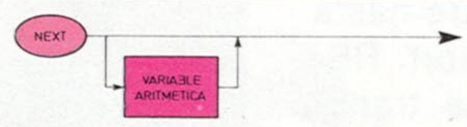
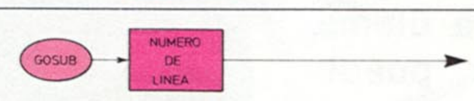
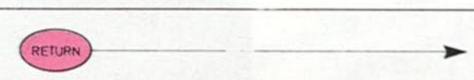
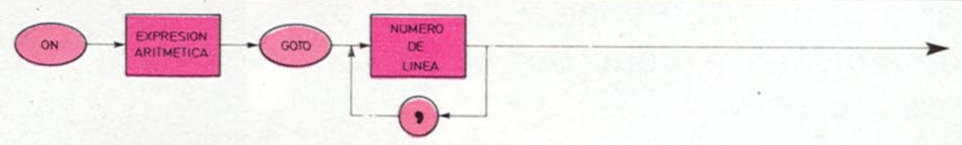
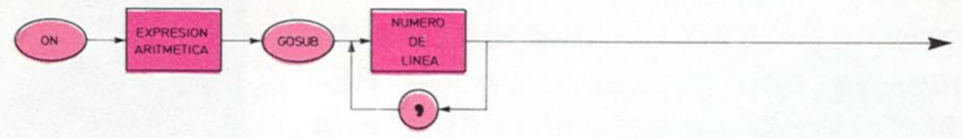
Cuando se ejecuta una instrucción PRINT se evalúa, si es necesario, el valor de la expresión. La presentación empieza a partir de la posición en que se encuentra el cursor, si antes se ha utilizado una función TAB de tabulación o LOCATE. Si la instrucción PRINT anterior acabó en punto y coma, el nuevo valor se mostrará a partir de la columna inmediatamente libre. Si no acabó en punto y coma se mostrará a partir de la columna 1 de la primera línea vacía.

Si se desea formatear la presentación de resultados es necesario usar la sentencia PRINT USING.

También se pueden crear tablas de valores mediante el programa. Así la instrucción DATA crea una tabla de constantes, mientras que la instrucción READ asigna las constantes a las variables especificadas. A la primera variable de la siguiente instrucción READ le asigna la primera constante no



El aprendizaje del lenguaje BASIC se ve facilitado por la existencia de una amplia variedad de programas didácticos orientados a las diversas variantes y dialectos.

INSTRUCCION	ENUNCIADO DE SINTAXIS	DIAGRAMA
FOR	FOR VARIABLE = EXPRESION ARITM. 1 TO EXPRESION ARITM. 2 [STEP EXPRESION ARITM. 3]	
NEXT	NEXT [VARIABLE ARITMETICA]	
GOSUB	GOSUB NUMERO DE LINEA	
RETURN	RETURN	
ON GOTO	ON EXPRESION ARITMETICA GOTO NUMERO DE LINEA [, NUMERO DE LINEA]	
ON GOSUB	ON EXPRESION ARITMETICA GOSUB NUMERO DE LINEA [, NUMERO DE LINEA]	

Instrucciones BASIC para el control de bucles y subrutinas.

asignada de la tabla y se sigue en este orden. Si no quedan constantes por asignar se genera mensaje de un error.
La instrucción RESTORE permite utilizar más de una vez la lista de las constantes. Después de esta instrucción, la primera constante de la tabla creada por DATA se vuelve a asignar a la primera variable de la siguiente instrucción READ que aparezca.

Definición de tipos de datos

El tener que poner detrás de cada variable el símbolo del tipo de datos complica bastante la escritura del programa.

Al cargar el intérprete cualquier nombre de variable que no tenga símbolo de tipo de datos, se representará automáticamente como una variable de precisión sencilla.
Se puede alterar esta regla mediante el uso de instrucciones de definición del tipo de datos. Hay cuatro:

DEFINT.—Define una serie de variables como enteros.
DEFSNG.—Define una serie de variables como de precisión sencilla.
DEFDBL.—Define una serie de variables como de precisión doble.
DEFSTR.—Define una serie de variables como de cadena de caracteres.

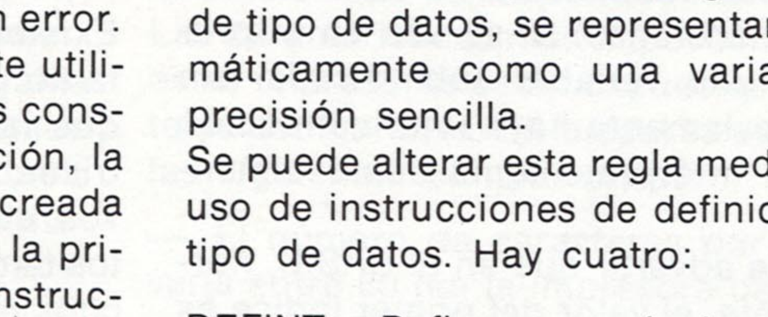
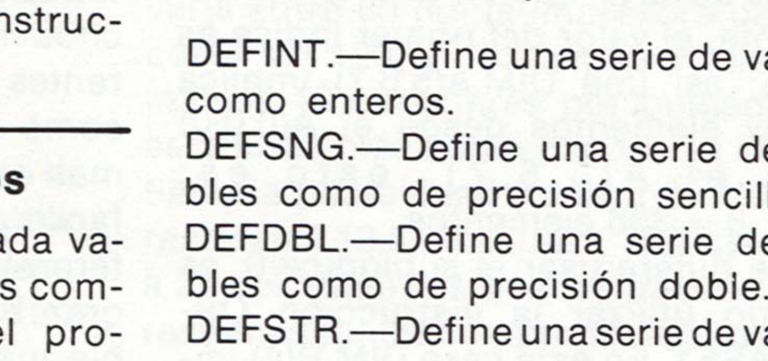
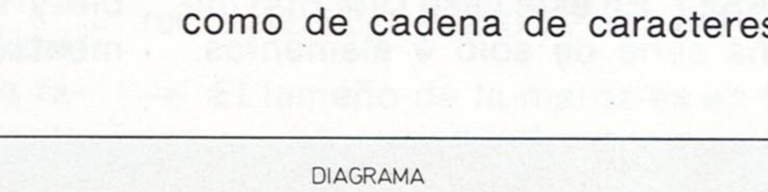
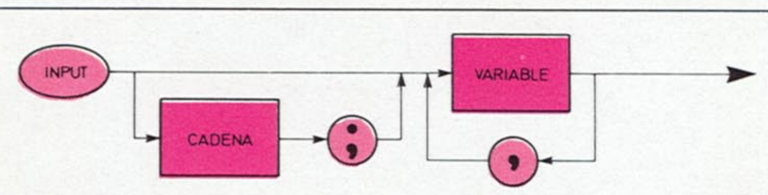
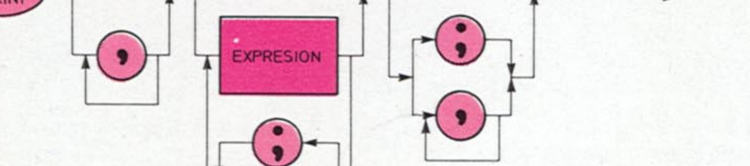
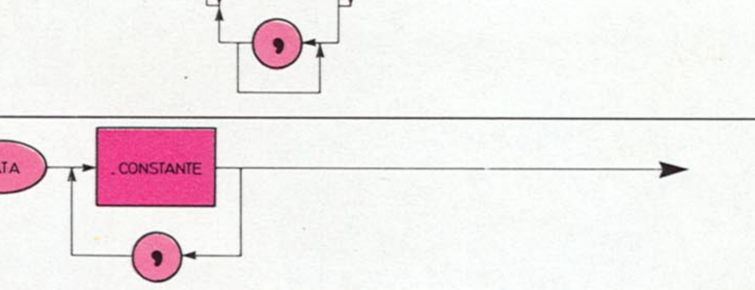
INSTRUCCION	ENUNCIADO DE SINTAXIS	DIAGRAMA
INPUT	INPUT [CADENA ;] VARIABLE [, VARIABLE....]	
PRINT/LPRINT	PRINT/LPRINT [,...] [EXPRESION [{;,} ....] [{;,} ....]	
DATA	DATA CONSTANTE [, CONSTANTE ....]	
READ	READ VARIABLE [, VARIABLE ....]	

Tabla de las instrucciones BASIC más frecuentemente utilizadas para programar operaciones de entrada y salida.

INSTRUCCION	ENUNCIADO DE SINTAXIS	DIAGRAMA
DEFXXX	DEFXXX LETRA INICIAL [- LETRA FINAL]	
DIM		

Instrucciones BASIC para la definición de datos.

Glosario

¿Qué pasa si se ejecuta una instrucción RETURN antes de una GOSUB?

Que se genera un error. No obstante, si hay una RETURN que no corresponde a la GOSUB ejecutada no se detectará el error. Por ello hay que tener mucho cuidado con la lógica de las subrutinas y evitar entrar incorrectamente en ellas.

¿Cómo se puede evitar entrar en una subrutina incorrectamente?

Es una buena práctica preceder a la subrutina con una de las siguientes instrucciones: STOP, END o GOTO hacia una zona situada fuera de la subrutina.

¿Se pueden usar entrecruzadas las instrucciones FOR/NEXT y GOSUB/RETURN?

Una instrucción RETURN anula todas las instrucciones FOR ejecutadas desde la última GOSUB a la que se ha tenido acceso. Por consiguiente, si se accede entonces a una NEXT hay error. Igual ocurre al revés. Lo correcto es ejecutar completamente el bucle dentro de la subrutina o ejecutar la subrutina dentro del bucle. Bucles y subrutinas no deben de cruzarse.

¿Qué sucede si en una instrucción ON no hay número de línea que corresponda al valor de la expresión aritmética?

Si es 0 o mayor que el número de ítems de la lista, se continúa en la siguiente instrucción ejecutable. Si es negativo da un error. Por ello conviene tomar la precaución de asegurarse que el valor de la expresión queda dentro de los límites correctos.

¿Se suele utilizar alguna letra inicial específica para algún tipo de variable?

Sí, por razones históricas, se suelen atribuir a las variables enteras nombres que empiezan por I, J, K, L, M y N, ya que se suelen utilizar como índices, subíndices, contadores, etc.

EL LENGUAJE BASIC (y 3)

Las cuatro tienen el mismo formato: DEFXXX letra inicial-letra final.
Siendo:
XXX: INT, DBL, SNG o STR, según el tipo de datos deseado.
Letra inicial: la única letra especificada o la letra inicial del grupo afectado.
Letra final: (opcional) la letra final del grupo de letras afectada. Así, DEFINT I-N implica que todas las variables cuyo nombre comience con las letras I, J, K, L, M y N serán enteras.
Estas instrucciones no afectan a las variables definidas directamente con el símbolo de tipos de datos.
Para definir la dimensión de una matriz se emplea la instrucción DIM, que junto al nombre de la matriz incluyen entre paréntesis los valores de cada una de sus dimensiones. Si se usa en el programa una variable subindicada, sin que previamente haya una instrucción DIM, el intérprete toma como dimensión 10.
Hay que advertir que en el BASIC, normalmente, el valor del primer índice es el cero; así una DIM A(5,6,7) implica que hay elementos desde el A(0,0,0) hasta el A(5,6,7), esto es: $6 \times 7 \times 8 = 336$ elementos.
Si no se quiere usar el subíndice 0, es necesario utilizar la instrucción OPTION BASE1. En este caso DIM B(4) implica una serie de sólo 4 elementos.

Resumen

Existen otras muchas instrucciones, tanto del programa como del sistema, que facilitan las operaciones con matrices, el manejo de periféricos y archivos, etc., si bien no tienen cabida en estos temas. De todas formas existen muchos manuales de BASIC para los diferentes equipos del mercado, ya que, como hemos indicado, es el lenguaje más extendido en la actualidad, por su facilidad de aprendizaje, el uso de intérprete, su simplicidad de manejo, la gran biblioteca de programas disponible y sus continuas mejoras e implementaciones.



El impulso definitivo del lenguaje BASIC ha llegado de la mano de los microordenadores, una de cuyas familias es la de los microordenadores familiares. Estos equipos están difundiendo el uso de este lenguaje incluso a nivel doméstico.

PERIFERICOS

IMPRESORAS MANNESMANN TALLY

LA firma MANNESMANN TALLY es un fabricante de impresoras para micro y miniordenadores. De entre la gama de impresoras, ofertadas por el citado fabricante, vamos a analizar cuatro modelos, todos ellos de impresión mediante matriz de agujas y sobre papel normal. Estos modelos son: MT 120, MT 140, MT 420 y MT 440.

Cada uno de estos modelos dispone de tres variantes que se referencian con las letras I, D y L. La variante I es una impresora para aplicaciones convencionales de tratamiento de información. La variante D ofrece la posibilidad de impresión de caracteres ópticos tales como códigos de barras. La variante L tiene la posibilidad seleccionable de impresión de tratamiento de datos o impresión de tratamiento de textos con una calidad de escritura adecuada para la edición de correspondencia.

Las características más importantes de estas impresoras están expuestas en la tabla comparativa, de la cual se extraen las siguientes conclusiones:

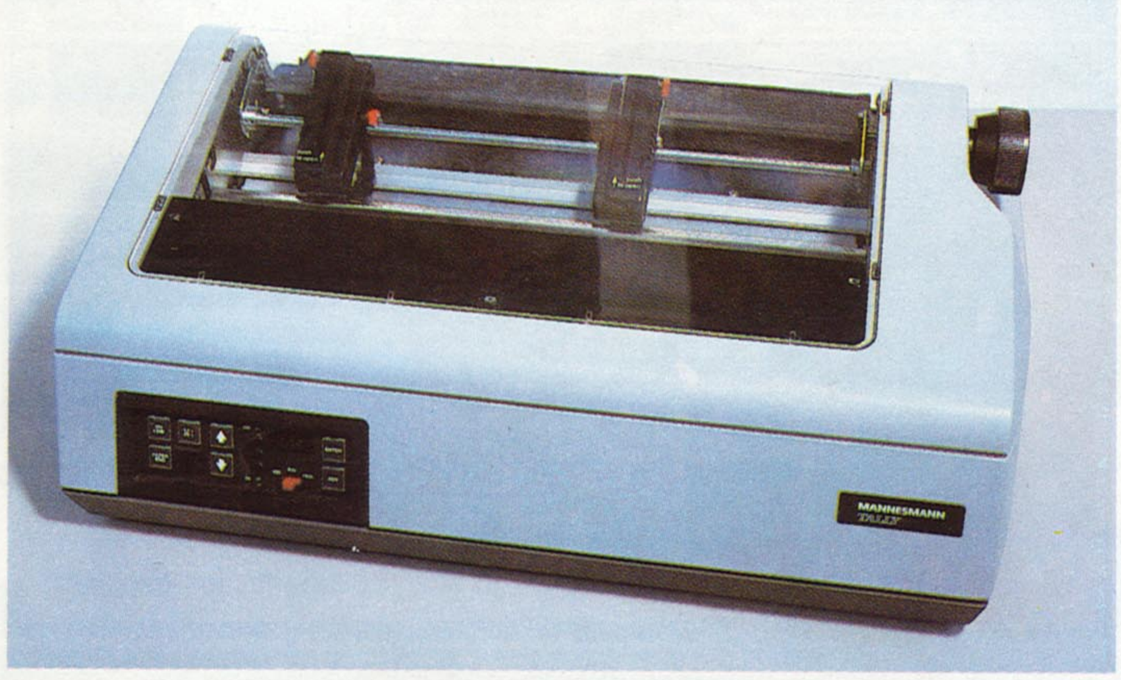
— El número de caracteres por línea varía entre 80 (en la impresora de bajo coste MT 120) y 132, en escritura normal de 10 caracteres por pulgada. En escrituras más condensadas, la MT 120 tiene la posibilidad de escribir 100 caracteres a 12,5 cpi, 132 a 16,7 cpi y 160 a 20 cpi; mientras que los tres modelos restantes pueden imprimir 165 caracteres a 12,5 cpi y 220 a 16,7 cpi.

— El tamaño de la matriz es en todos los modelos de 9 × 7 puntos en la variante I y de 9 × 9 en la variante D. En la variante L se puede seleccionar el tamaño de la matriz a 9 × 7, para una impresión de tratamiento de datos, o a 18 × 40 para obtener la calidad de una escritura de correspondencia asociada a un tratamiento de textos.

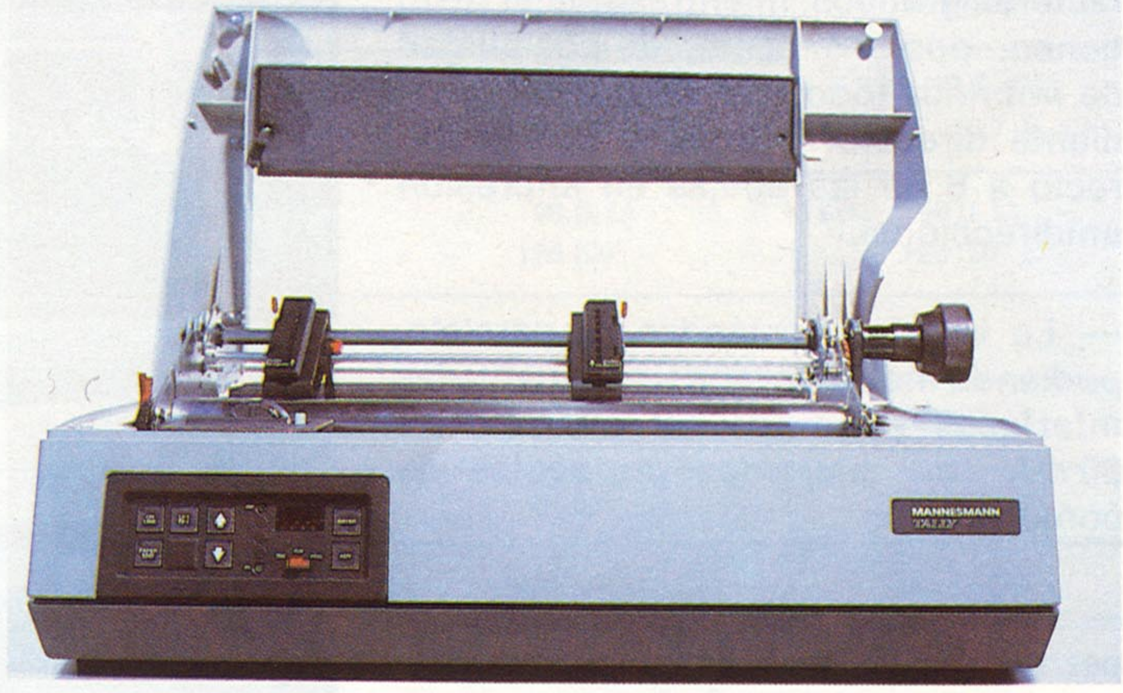
— Son impresoras rápidas, desde los 160 caracteres por segundo de la MT 120 a los 400 cps de la MT 440. La velocidad de impresión disminuye a la cuarta parte en el modelo L cuando se trata de obtener calidad de correspondencia con la matriz de 18 × 40 puntos.

— El tamaño del buffer de memoria es de una línea, lo cual supone en las impresoras de 132 caracteres por línea una memoria de 1 Kbit.

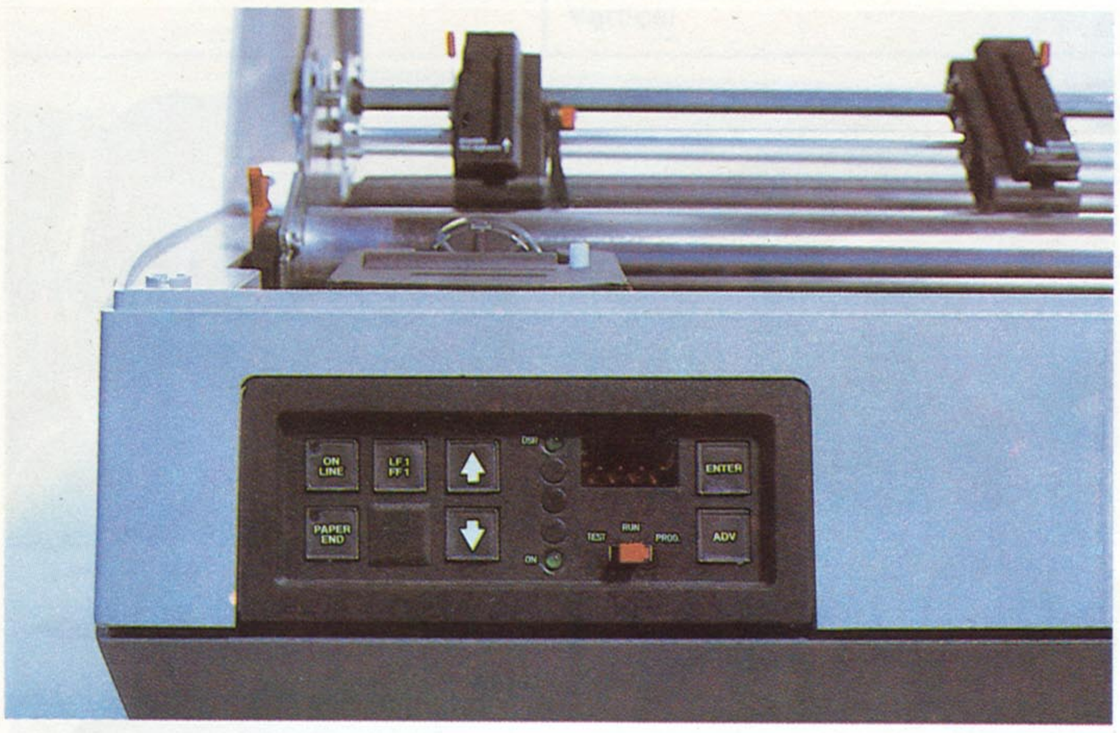
— Disponen de distintos tamaños de ancho de letra para lograr una escritura más condensada. En la variante L, cuando se imprime con matriz de



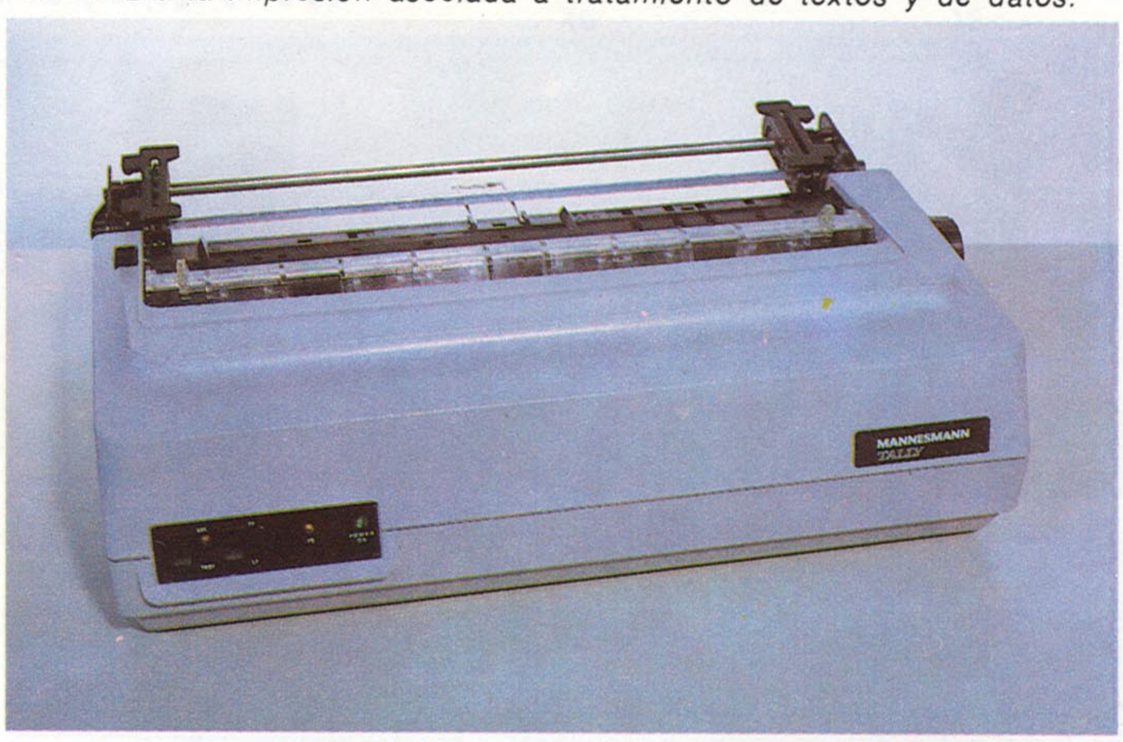
La MT 440 de Mannesmann Tally es una impresora de alta velocidad (400 caracteres por segundo) que se fabrica en las tres variantes I, D y L.



El modelo MT 440 I está orientado a la impresión de datos a alta velocidad, el MT 440 D a la impresión de código de barras OCR y el MT 440 L a la impresión asociada a tratamiento de textos y de datos.



Los modelos de impresora MT 440 disponen de las opciones de arrastre de papel por doble tractor, fricción, inserción frontal e introductor de hojas.



La MT 120 es una impresora de bajo costo disponible en las tres versiones I, D y L. El modelo D es adecuado para la impresión de caracteres ópticos.

IMPRESORAS MANNESMANN TALLY

18 × 40, sólo se puede imprimir con una densidad de 10 caracteres por pulgada (escritura normal). Por lo demás, admiten la posibilidad de escritura ensanchada o elongada; las MT 120 y MT 140 pueden imprimir con escritura ensanchada al doble, mientras que las MT 420 y MT 440 pueden ampliar el ancho de los caracteres a 2, 3, 5 ó 7 veces su ancho normal.

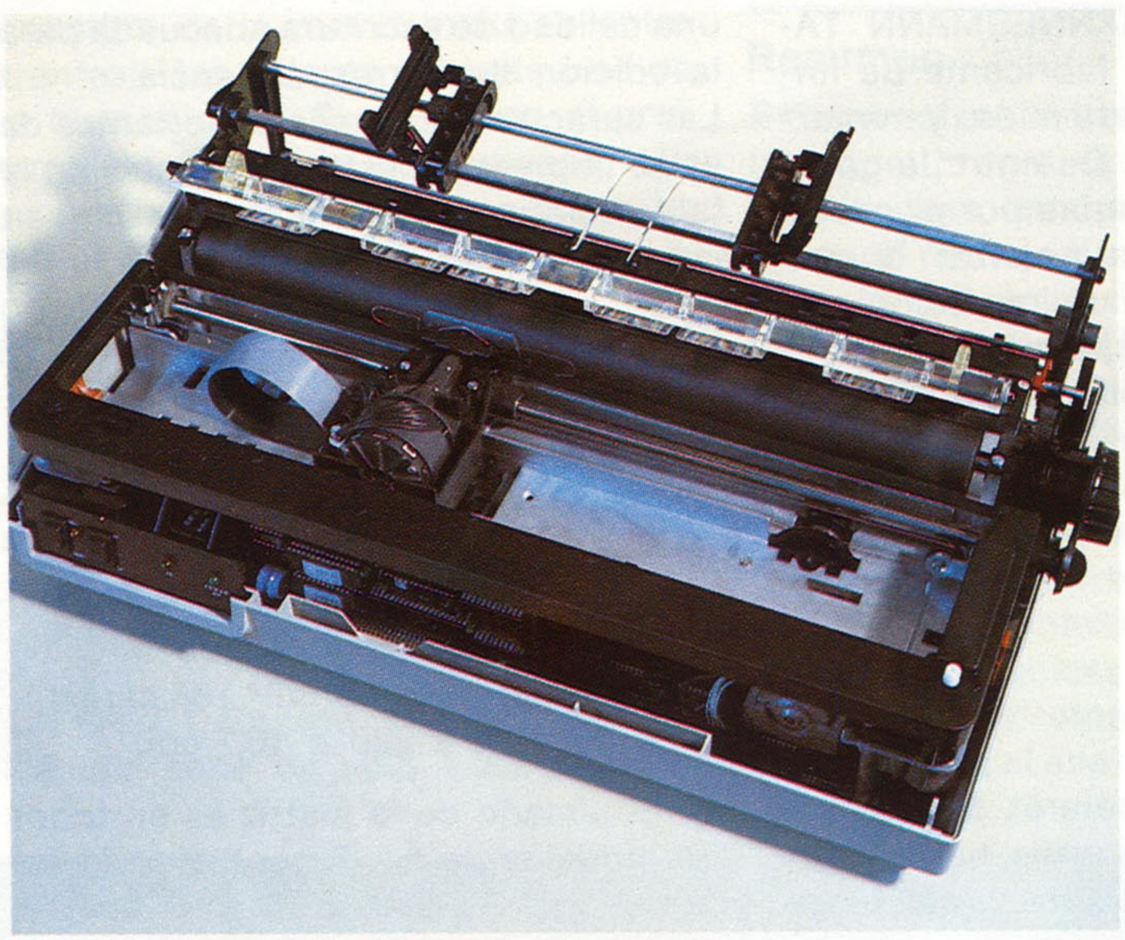
— Disponen de juego de caracteres multinacional, aunque no disponen de la posibilidad de espaciado proporcional.

— La impresión es bidireccional, con circuito lógico de optimización, dependiendo de la posición de la cabeza impresora en cada momento.

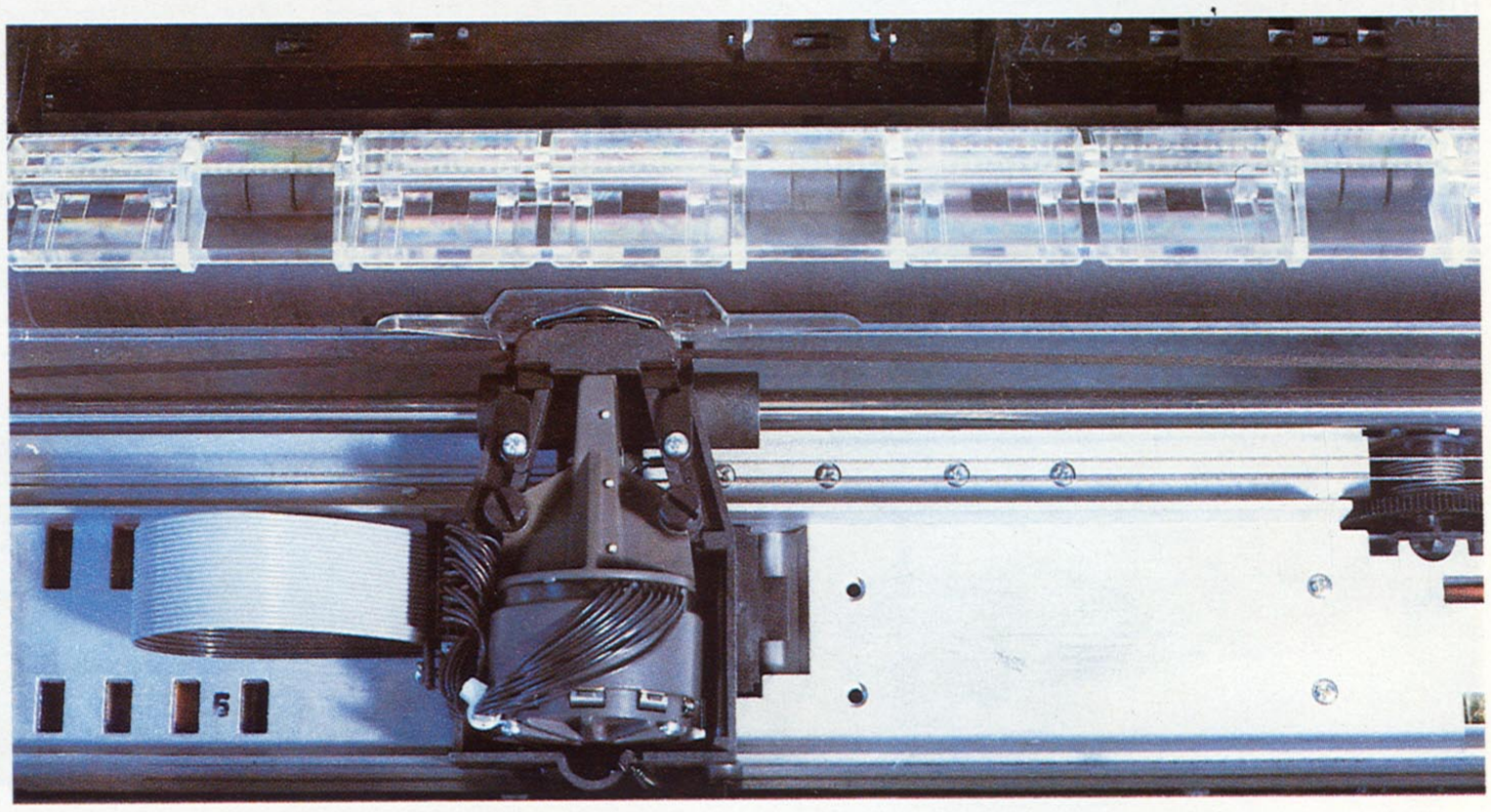
— Las impresoras MT 120 y MT 140 en sus versiones I y L pueden imprimir caracteres gráficos, mientras que el resto tienen, opcionalmente, la posibilidad de imprimir todo tipo de gráficos mediante direccionamiento y acceso directo a 8 de las agujas en impresión unidireccional.

— La interface estándar es paralelo, pudiendo incorporar opcionalmente las interfaces serie RS 232 o bucle de 20 mA con diferentes protocolos de comunicación.

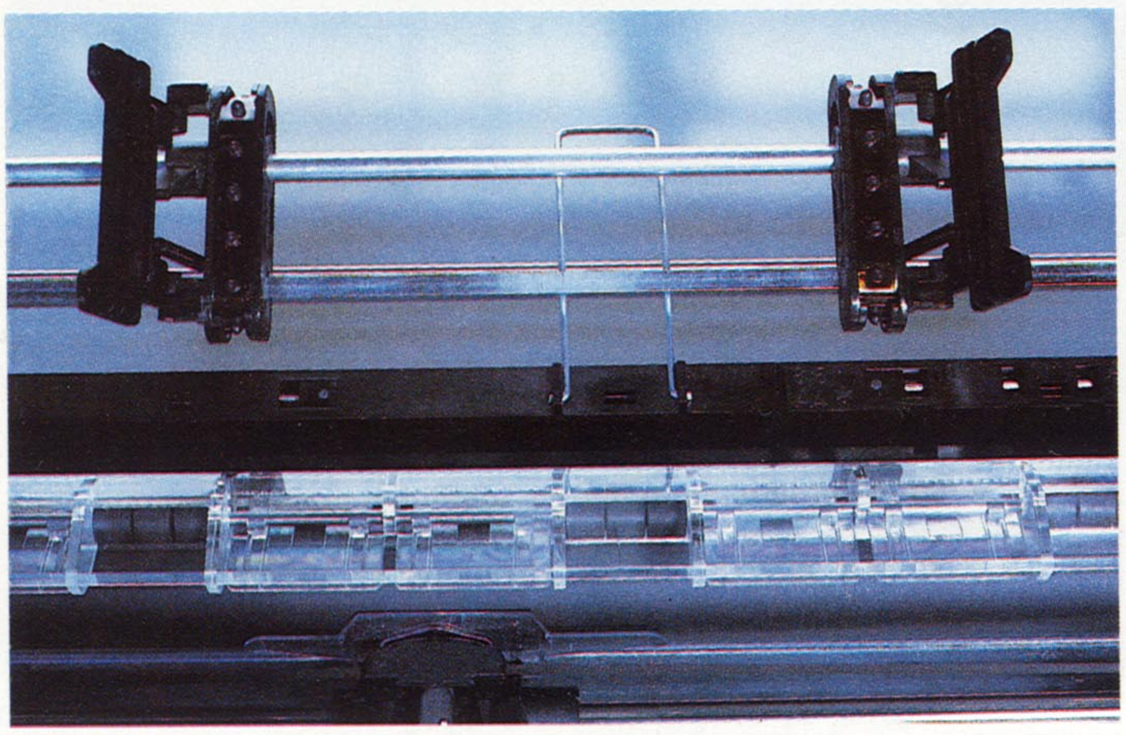
–– Disponen de sensor de final de papel que activa una señal acústica de alarma. Esta alarma acústica es también activada por otras condiciones de error.



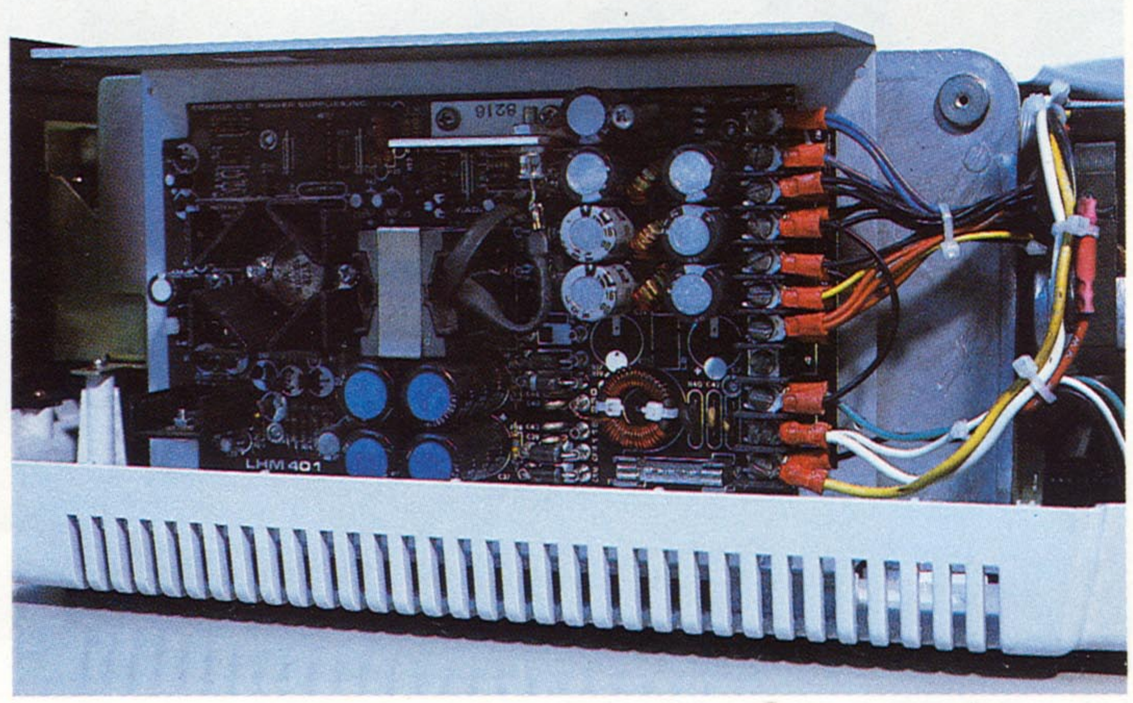
La velocidad de impresión de la MT 120 es de 160 c.p.s., imprimiendo de forma bidireccional optimizada a 80 columnas de ancho.



Las tres versiones de la MT 120 difieren en las características de la matriz de agujas. En la versión MT 120 I la matriz es de 9 × 7, en la versión D de 9 × 9 agujas y en la versión L la matriz es seleccionable a 9 × 7 ó 18 × 40.



Para el arrastre del papel, las tres versiones de la impresora MT 120 disponen de las siguientes opciones: tractor, fricción o introducción manual.



La robustez y la necesidad de un escaso mantenimiento son dos de las características compartidas por la gama de impresoras de matriz de agujas Mannesmann Tally analizadas.

TABLA COMPARATIVA IMPRESORAS MANNESMAN TALLY

		MT-120	MT-140	MT-420	MT-440
Máximo ancho de papel		10″	16″	16″	16″
Caracteres/línea (escritura normal)		80	132	132	132
Densidad de líneas (líneas por pulgada)		6,8	6,8	3, 4, 6, 8, 12	3, 4, 6, 8, 12
Número de agujas		9	9	9	9
Tamaño matriz		9 × 7 (I) 9 × 9 (D) 9 × 7/18 × 40 (L)	9 × 7 (I) 9 × 9 (D) 9 × 7/18 × 40 (L)	9 × 7 (I) 9 × 9 (D) 9 × 7/18 × 40 (L)	9 × 7 (I) 9 × 9 (D) 9 × 7/18 × 40 (L)
Forma de alimentación del papel	Fricción	★	★		
	Tracción	★ (op.)	★ (op.)	★	★
Velocidad de escritura (c.p.s.)		160 (I) 120 (D) 160/40 (L)	160 (I) 120 (D) 160/40 (L)	200 (I) 140 (D) 200/50 (L)	400 (I). 270 (D) 400/100 (L)
Tiempo de avance de 1 línea (mseg.)		150	70	20	20
Tamaño del buffer		1 línea	1 línea	1 línea	1 línea
Distintos tipos de letra (carácter. por pulgada)		10, 12,5 (D) 10, 12,5, 16,7 (I, L)	10, 12,5 (D) 10, 12,5, 16,7 (I, L)	10, 12,5 (D) 10, 12,5, 16,7 (I, L)	10, 12,5 (D) 10, 12,5, 16,7 (I, L)
Juego de caracteres		96 (I, L) 128 (D)	96 (I, L) 128 (D)	96 (I, L) 128 (D)	96 (I, L) 128 (D)
Caracteres españoles		SI	SI	SI	SI
Espaciado proporcional		NO	NO	NO	NO
Número de copias		3	3	5	5
Dirección de impresión		BIDIRECC.	BIDIRECC.	BIDIRECC.	BIDIRECC.
Salto de página, programado hasta (líneas)		127	127	127	127
Posiciones de tabulación	Horizontal			64	64
	Vertical			16	16
Posibilidad de gráficos		Caract. gráficos (I, L) Direccion. agujas (op.)	Caract. gráficos (I, L) Direccion. agujas (op.)	Direccion. agujas (Opcional)	Direccion. agujas (Opcional)
Tipo de interface	Paralelo	★	★	★	★
	RS 232	★ (op.)	A (op.)	★ (op.)	★ (op.)
	Bucle 20 mA	★ (op.)	★ (op.)	★ (op.)	★ (op.)
Nivel de ruido acústico (db)		60	60	60	60
Temperatura de funcionamiento		10 ÷ 40° C	10 ÷ 40° C	10 ÷ 40° C	10 ÷ 40° C
Humedad de funcionamiento		20 ÷ 90 %	20 ÷ 90 %	20 ÷ 90 %	0 ÷ 90 %
Consumo (V.A.)		100	100	150	150

APLICACIONES

BASE DE DATOS CONTROL

UNA gran parte de los microordenadores que existen en el mercado no soportan directamente la opción de ficheros secuenciales indexados, a pesar de que este tipo de ficheros resultan idóneos para constituir los archivos maestros de casi cualquier tipo de aplicaciones de gestión. La solución a este inconveniente se logra a través de las Bases de Datos, unas más potentes que otras, pero que tienen como característica común la de soportar la creación, mantenimiento, visualización y listado de este tipo de ficheros.

Uno de estos paquetes de aplicación es el denominado CONTROL. Concebido para su implementación en microordenadores que incorporan el sistema operativo OASIS, la aplicación CONTROL es un generador de programas para el acceso a bases de datos.

Aplicación: **Base de Datos CONTROL**
Ordenador: **ALTOS Serie 8000 / Serie 5**
Configuración: **Unidad central, unidad de disco, terminal e impresora.**
Sistema operativo: **OASIS**
Soporte: **Disco flexible de 5 y 1/4 u 8″**
Documentación: **Manual de 60 páginas en inglés.**
Copyright: **QUAYDATA Inc.**
Distribuidor: **PHASE ONE SYSTEMS, Inc.**

** La aplicación se ha evaluado en los ordenadores que se indican, si bien puede implementarse en otros equipos dotados de sistema operativo OASIS.*

El programa generador

El paquete generador para el mantenimiento de archivos y pantallas de entrada emplea una serie de programas que permiten al usuario definir una pantalla de datos, así como el fichero que contiene éstos y, a su vez, generar un programa BASIC para su tratamiento. Para generar un programa de tratamiento, por simple que éste sea, es necesaria la intervención de varios de los programas incluidos en el paquete CONTROL, tales como: SCREEN que contiene la secuencia de campos dentro del registro y el formato de la pantalla, SETUP que especifica el fichero objeto de mantenimiento y EXPAND que genera el programa de mantenimiento. Otros programas como INVOKE, DOCUMENT Y SETUP (parte II) se emplean según necesidades.

Las funciones de algunos de los programas auxiliares, cuya actuación se lanza a través del comando oportuno son:

- SCREEN: permite añadir un nuevo campo a la pantalla, modificar uno ya existente, almacenar la pantalla definida, mostrar el estado actual de la pantalla, enmarcar campos, inserción y borrado.
- INVOKE: permite comprobar el diseño y operatividad de una pantalla sin la generación del programa de mantenimiento.
- DOCUMENT: permite la opción de HARD COPY (obtener una copia por impresora) del diseño de la pantalla.
- SETUP: permite definir la relación entre el programa de mantenimiento de ficheros, el fichero que mantiene, la salida de pantalla que utiliza, los campos a emplear y la secuencia en que serán presentados al operador.
- EXPAND: permite, a partir de la in-

DICCIONARIO PROPIO DE LA BASE DE DATOS CONTROL	
Verbos	LISTA, CLASIFICA, SELECCIONA, CLSELECCIONA
Condiciones de selección	=, > , < , <> , > = , < =, CONTENIENDO
Conectivos	Y, O
Especificaciones de clasificación	POR, POR.DESCEN
Especificaciones de salida	MUESTRA, IMPRIME, TOTAL, SUBTOTAL, CABECERA, PIE.DCHA y PIE.IZDA, NO.ID NO.WRAP
Otras palabras (No afectan a la estructura de la frase.)	UN, UNA, SEAN, ALGUN, TODOS, ARCHIVO, PARA, EN, REGISTROS, DE, DEL, SEA, QUE, HASTA, LOS, DONDE, TAMBIEN, TENGAN, ASI, COMO, DAME

RELACION DE SINONIMOS	
CLASIFICA	ORDENA, ORGANIZA
SELECCIONA	—
CLSELECCIONA	—
DESDE	—
TALES	CON
IMPRIME	—
MUESTRA	—
NO.WRAP	—
TOTAL	—
POR	—
POR.DESCEN	—
SUBTOTAL	CORTA, CORTA.EN
PIE.DCHA	—
PIE.IZDA	—
NO.ID	—
TAMPOCO	—
=	EQ, IGUAL, IGUALE
>	GT, DESPUES, MAYOR, MAS, POSTERIORES
<	LT, ANTES, MENOR, ANTERIORES
<>	NE, DISTINTO
< =	LE
> =	GE
CONTENIDO	CONTIENE, CONTENGA, INCLUYE
Y	—
O	—

formación suministrada por SCREEN y SETUP, generar un programa BASIC para el mantenimiento de ficheros.

Sistema de acceso

El procesador de acceso es una herramienta importante. Está diseñado para permitir el fácil acceso del usuario a la información almacenada, mediante la utilización de un «diccionario», por supuesto limitado, que permite construir frases inteligibles para el ordenador. De esta forma, pueden definirse cabeceras, pies de página, subtotales, totales, tabulaciones y cálculos complejos.

El comando VOC se emplea para construir y mantener el vocabulario de atributos utilizado, y el ACCESS para el acceso a los datos a través de este vocabulario. La estructura de estas frases especifica cuatro informaciones básicas: *la operación a realizar, sobre qué archivo, con qué condiciones y en qué formato.* Esto es: verbo, archivo, condición selección, condición clasificación y especificación de salida.

Los verbos utilizables son:

LISTA, para obtención de informes por pantalla o impresora;

CLASIFICA, para obtener un informe clasificado según se especifique;

SELECCIONA, igual que CLASIFICA, pero teniendo salida únicamente a un fichero en disco;

CLSELECCIONA, mezcla de estas dos opciones con salida a disco con nombre «SORTED. usuario».

Las condiciones de selección pueden adoptar las siguientes estructuras: literal, desde-literal, tales-expresión relacional-Y u O-expresión relacional, etcétera.

Dentro de las especificaciones de salida se dan indicaciones acerca de si va a ser por pantalla o impresora, se establece la cabecera, subtotales y totales a calcular y pies de página.

Otras características

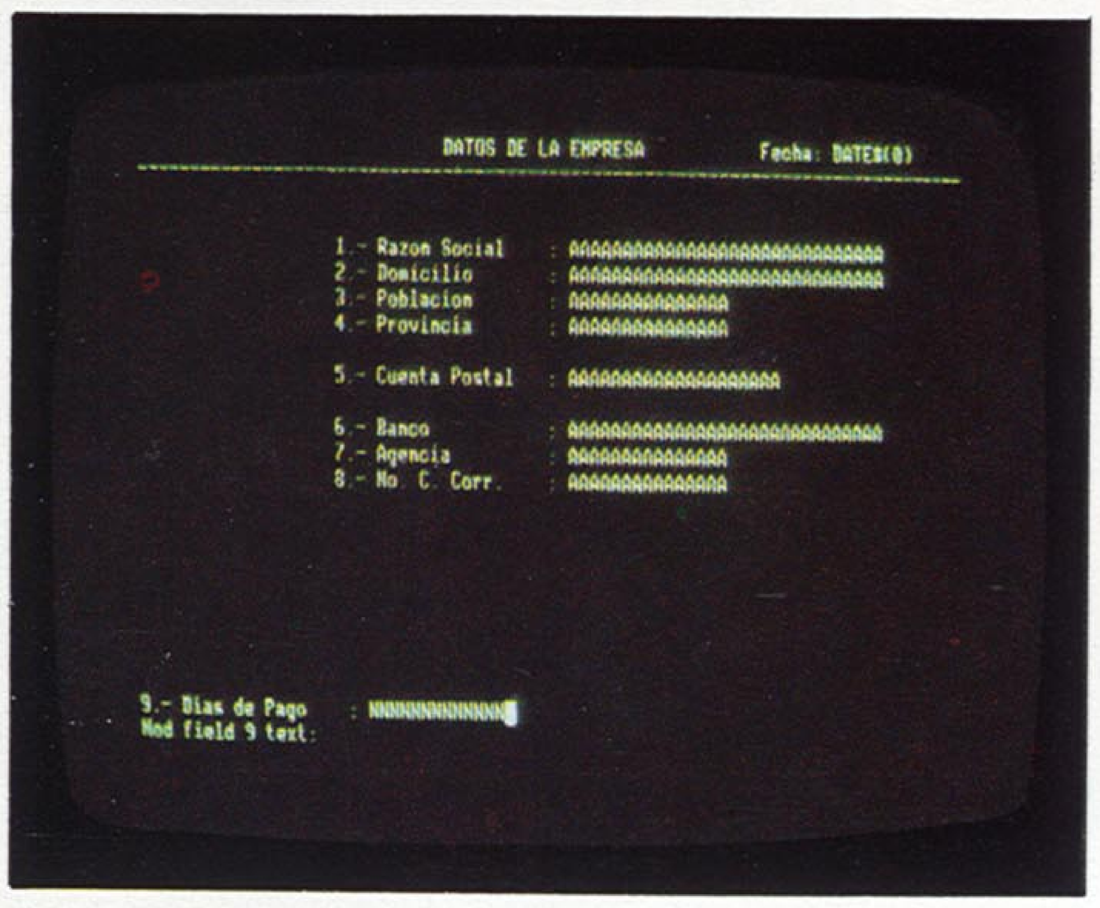
La Base de Datos CONTROL permite también la utilización de palabras clave del sistema operativo OASIS como @ DATE, que contiene la fecha del Sistema; @ TIME, que indica la hora; @ USER, con el nombre de usuario; @ PRIV, con el nivel del usuario y @ ID, que contiene el identificador del registro en curso.

Se pueden establecer entre los campos relaciones aritméticas o condicionales IF, THEN Y ELSE.

La utilización de palabras castellanas, e incluso sinónimos, facilita el trabajo del operador a la hora de informar al sistema de las características y formato en que desea que aparezcan los datos.



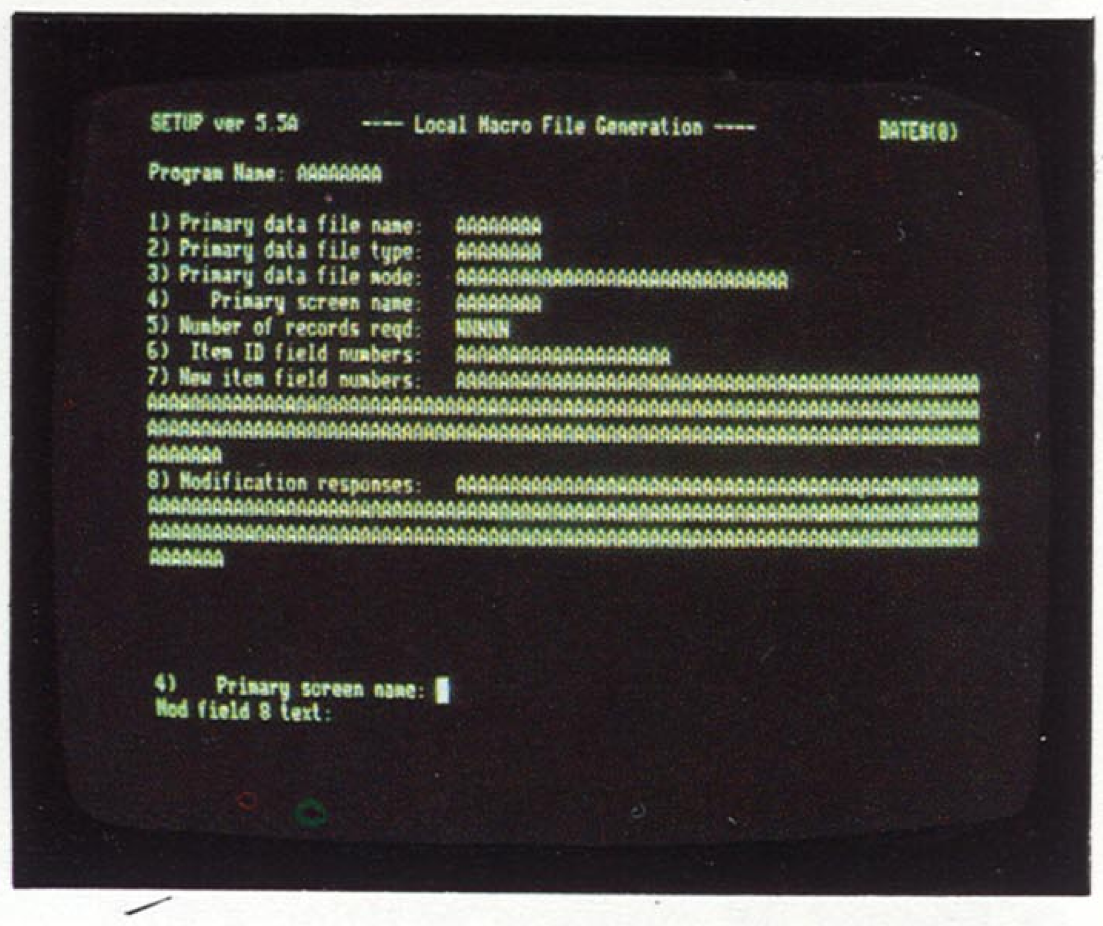
La aplicación CONTROL es un paquete integrado por un conjunto de programas orientados a la creación y mantenimiento de bases de datos. El paquete está diseñado para su implantación en microordenadores dotados del sistema operativo OASIS.



Formato de una pantalla para introducción de datos de empresas. Las letras que aparecen en la zona derecha de cada campo definen la naturaleza del dato (A = carácter alfanumérico).



En este caso, la pantalla diseñada por medio del programa SCREEN está orientada a la creación de un archivo de pedidos. En la línea inferior aparecen los comandos disponibles.



El comando SETUP da entrada al programa que permite definir las características del archivo de datos y las condiciones para la generación del programa de mantenimiento.

PROGRAMA

Título: ***Gus***
Ordenador: ***Sinclair ZX-Spectrum***
Memoria requerida: ***48 Kbytes***
Lenguaje: ***BASIC***

GUS es un reptil muy especial, le encanta la fruta. El fin de este juego es dirigir a GUS para comer el mayor número de frutas posible, si bien hay que evitar, al tiempo, que ingiera alguna de esas sospechosas calabazas de color azul que le supondrían la muerte.

Las paredes del recinto en el que GUS vive están electrificadas, para evitar la huida de tan valioso animal y, por tanto, son intocables. Por otra parte, la mordedura del reptil es tan peligrosa que si llegara a dar un ligero mordisco en su propio cuerpo también fenecería.

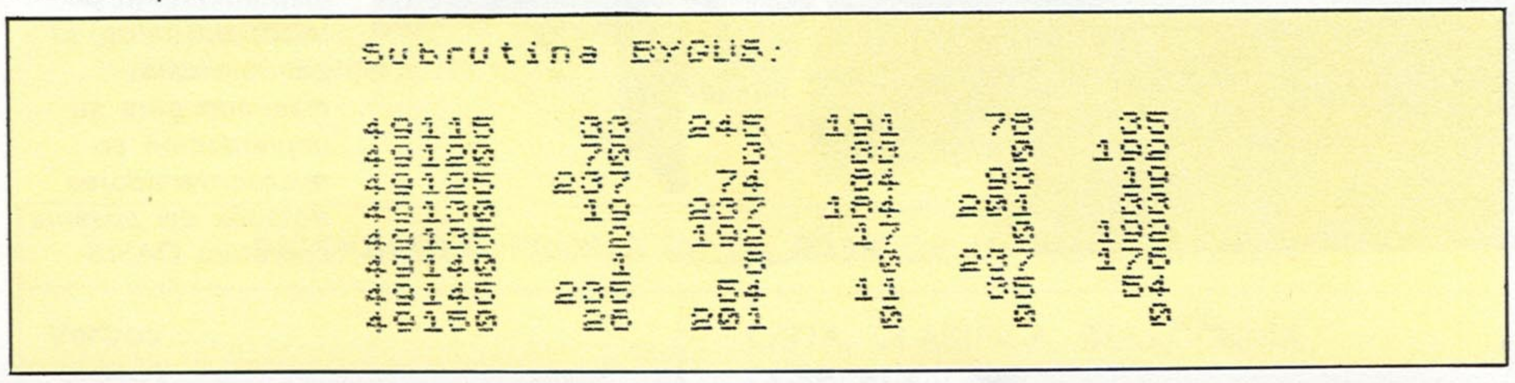
Cabe observar que GUS sólo permanece quieto cuando acaba de comerse una fruta y que, una vez que haya echado a andar, la única forma de detenerlo es volviendo a comer. Por cada vez que ocurra esta situación, el cuerpo de la serpiente se alarga, haciendo más difíciles las maniobras. Cada diez frutas se pasa a una nueva pantalla, algo más difícil que la anterior, debido a la presencia de un mayor número de «calabazas asesinas».

GUS se dirige con las teclas del cursor y el orden de puntuación es: Fresa > Guindas > Pera. Por ello, es más frecuente que las fresas resulten ser «alucinaciones», mientras que las peras nunca lo son. En la parte inferior de la pantalla figuran un marcador de puntos y otro con el número de pantalla que se está jugando.

A efectos de grabación del programa, hay que advertir que las letras mayúsculas que figuran entre comillas en las líneas 210, 590, 600, 610, 680, 720, 790 y 800, corresponden a los gráficos definidos de las mismas. Una vez entrado el programa principal se efectúa un CLEAR 49114 y se introduce el código máquina de la pequeña subrutina BYGUS a partir de la dirección 49115; una vez almacenada esta subrutina, ya se puede proceder a la carga de la subrutina CARACTERES GIGANTES (publicada en el fascículo 6 de la Enciclopedia). A continuación, se almacena el programa GUS con AUTORUN en 10, la subrutina BYGUS y, por último, la subrutina CARACTERES GIGANTES.

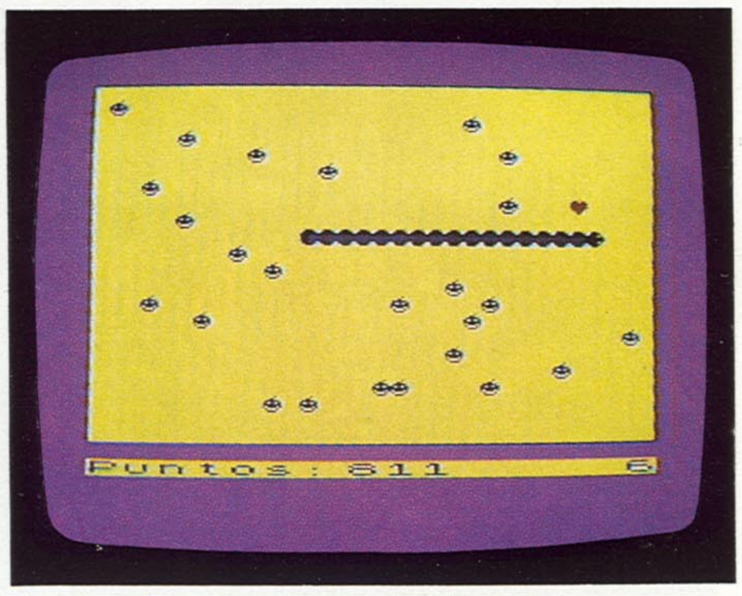
Hay que destacar que, gracias al empleo de la variable F$ como una mezcla de caracteres definidos y caracteres de control del color, se consigue que al presentar cualquiera de las frutas, el sólo hecho de tomar la parte de la cadena en la que ésta se encuentra, implica que su representación aparezca en el color correspondiente, sin necesidad de someter el carácter a una evaluación lógica, previa a la impresión, para encontrar el atributo adecuado.

Subrutina BYGUS:





La finalidad del programa es dirigir a GUS (un inquieto y hambriento reptil) para que coma el mayor número posible de frutas y, a su vez, eluda la ingestión de las venenosas calabazas azules.



Cada vez que GUS engulle diez frutas, se pasa a una nueva pantalla. En la parte inferior de la presentación en pantalla aparece el marcador de puntuación y el número de pantalla en la que se está jugando.

```
  10 REM L.Martinez
  20 CLEAR 49114: LOAD "bygus"CO
DE : LOAD "literal"CODE
  30 DATA 10,17,10,16,10,15,10,1
4,10,13
  40 DATA 60,126,255,255,255,255
,126,60
  50 DATA 28,62,95,255,255,95,62
,28
  60 DATA 60,126,255,255,255,90,
60,24
  70 DATA 24,60,90,255,255,255,1
26,60
  80 DATA 56,124,250,255,255,250
,124,56
  90 DATA 6,2,12,28,62,126,126,6
0
 100 DATA 29,63,127,223,255,71,3
0,28
 110 DATA 4,24,126,219,255,189,6
6,60
 120 DATA 3,2,2,2,2,2,2,2
 130 DATA 192,48,12,3,0,0,0,0
 140 DATA 0,0,0,0,192,48,12,3
 150 DATA 3,4,12,16,34,239,239,2
31
 160 DATA 8,118,255,255,255,126,
60,24
 170 DATA 0,16,0,16,32,66,60,0
 180 RANDOMIZE
 190 GO SUB 510
 200 GO SUB 760
 210 BORDER 3: LET f$=CHR$ 16+CH
R$ 4+"F"+CHR$ 16+CHR$ 2+"M"+CHR$
 16+CHR$ 3+"L"
 220 LET s=0: GO SUB 640
 230 LET p=0: LET c=1
 240 IF p<0 THEN LET p=0
 250 LET z$=STR$ p: GO SUB 550
 260 LET mx=INT (RND*30)+1: LET
my=INT (RND*20)+1: IF SCREEN$ (m
y,mx)<>" " THEN GO TO 260
 270 LET f=INT (RND*3)+1
 280 PRINT AT my,mx;f$(f*3-2 TO
f*3); INK 1
 290 IF j<>0 THEN LET j=0: GO TO
 310
 300 LET m$=INKEY$: IF m$<"5" OR
m$>"6" THEN GO TO 300
 310 LET z$=INKEY$: IF z$>"4" AN
D z$<"9" THEN LET m$=z$
 320 LET j=j+1: LET dx1=dx+(1 AN
D m$="8")-(1 AND m$="5")
 330 LET dy1=dy+(1 AND m$="6")-(
1 AND m$="7")
 340 IF dx1>31 OR dx1<0 THEN GO
TO 310
 350 IF dy1>21 OR dy1<0 THEN GO
TO 310
 360 IF j>10*f AND f<>1 THEN PRI
NT AT my,mx;" ": GO TO 240
 370 IF SCREEN$ (dy1,dx1)=" " TH
EN GO SUB 580: BEEP 0.005,0: GO
TO 310
 380 IF ATTR (dy1,dx1)<>49 THEN
GO TO 440
 390 GO SUB 580
 400 BEEP .15,0: BEEP .15,0: BEE
P .15,0: BEEP 1.3,-5: PAUSE 2
 410 BEEP .15,-2: BEEP .15,-2: B
EEP .15,-2: BEEP 1.3,-7
 420 GO SUB 850
 430 STOP
 440 GO SUB 580: BEEP .5,5: BEEP
 .25,10: BEEP .1,20
 450 LET p1=p+c+s+((f-1)*5)-INT
(j/25): IF p1<p THEN LET p1=p
 460 LET p=p1: LET j=0: LET c=c+
1: LET cc=PEEK 49141+256*PEEK 49
142
 470 FOR i=0 TO 3: LET w=USR 491
15
 480 LET cc=cc+2: POKE 49142,cc/
256: POKE 49141,cc-256*PEEK 4914
2: NEXT i
 490 IF c=11+s*10 THEN LET s=s+1
: GO SUB 640
 500 GO TO 240
 510 RESTORE 40
 520 FOR i=144 TO 157: FOR j=0 T
O 7
 530 READ w: POKE USR CHR$ i+j,w
 540 NEXT j: NEXT i: RETURN
 550 LET i=23306: POKE i,xx: POK
E i+1,yy: POKE i+2,xs: POKE i+3,
ys: POKE i+4,8
 560 LET i=i+4: LET w=LEN z$: FO
R n=1 TO w: POKE i+n,CODE z$(n):
 NEXT n
 570 POKE i+w+1,255: LET w=USR 6
4869: RETURN
 580 IF dx1=dx THEN GO TO 600
 590 LET d$=("B" AND dx1<dx)+("E
" AND dx1>dx): GO TO 610
 600 LET d$=("D" AND dy1<dy)+("C
" AND dy1>dy)
 610 PRINT AT dy,dx;"A": LET dx=
dx1: LET dy=dy1: PRINT AT dy,dx;
 INK 0;d$; INK 1
 620 PRINT AT PEEK 49152,PEEK 49
153;" "
 630 POKE 49147,dy: POKE 49150,d
x: LET w=USR 49134: RETURN
 640 RESTORE : FOR i=49152 TO 49
161: READ j: POKE i,j: NEXT i: P
OKE 49141,6: POKE 49142,0
 650 INK 1: PAPER 6: CLS
 660 LET j=0: LET xx=0: LET yy=1
84: LET xs=2: LET ys=1: LET z$="
Puntos: ": GO SUB 550
 670 LET xx=240: LET yy=184: LET
 xs=2: LET ys=1: LET z$=STR$ (s+
1): GO SUB 550: LET xx=116
 680 LET dx=13: LET dy=10:: PRIN
T AT dy,dx; INK 0;"B"; INK 1;"AA
AA"
 690 DRAW 0,175: DRAW 255,0: DRA
W 0,-175: DRAW -255,0
 700 FOR i=1 TO 5*s
 710 LET mx=INT (RND*30)+1: LET
my=INT (RND*20)+1: IF SCREEN$ (m
y,mx)<>" " THEN GO TO 710
 720 PRINT AT my,mx; INK 1;"H":
NEXT i
 730 BEEP .25,20: BEEP .25,10: B
EEP .25,18: BEEP .25,13: BEEP .5
,20: BEEP .25,5
 740 PAUSE 15: BEEP .10,-5: BEEP
 .10,-10: BEEP .10,-5: BEEP .10,
-10: BEEP .25,-5: PAUSE 15: BEEP
 .75,-20
 750 RETURN
 760 BORDER 3: INK 7: PAPER 3: C
LS
 770 LET z$="GUS": LET xs=8: LET
 ys=4: LET yy=32: LET xx=(256-8*
xs*LEN z$)/2: GO SUB 550
 780 LET z$="© L.M. 1983": LET x
s=2: LET ys=2: LET yy=128: LET x
x=(256-8*xs*LEN z$)/2: GO SUB 55
0
 790 PAUSE 50: LET z$="
                      IUK
                          "
 800 LET y$="
       F GAAAA
                "
 810 FOR i=1 TO 39
 820 PRINT AT 11,1;z$(i TO i+30)
;y$(i TO i+30): BEEP 0.015,0
 830 NEXT i
 840 BEEP .5,5: BEEP .25,10: BEE
P .1,20: PAUSE 100: RETURN
 850 CLS : LET xx=1: LET yy=1: L
ET xs=2: LET ys=4: LET z$="Fruta
s: "+STR$ (c-1): GO SUB 550
 860 LET xx=1: LET yy=41: LET xs
=2: LET ys=4: LET z$="Pantallas:
 "+STR$ s: GO SUB 550
 870 LET xx=1: LET yy=81: LET xs
=2: LET ys=4: LET z$="Puntos: "+
STR$ p: GO SUB 550
 880 PRINT AT 16,0;"N Desearia";C
HR$ 8; OVER 1;"¿"; OVER 0;"e int
entarlo de nuevo ?"
 890 LET z$=INKEY$: IF z$="" THE
N GO TO 890
 900 IF z$="s" THEN RUN 210
 910 IF z$="n" THEN BORDER 2: CL
S : LET xs=1: LET ys=4: LET z$="
Gracias por jugar": LET xx=60: L
ET yy=72: INK 2: GO SUB 550: RET
URN
 920 PRINT AT 20,3;"En castellan
o, por favor.": PAUSE 100: PRINT
 AT 20,3;"
                ": GO TO 890
```

LOS ORDENADORES Y LA GUERRA

EN el futuro, la guerra será cada vez más asunto de los ordenadores. Pero, que no cunda el pánico, su misión consistirá más en solucionarla que en producirla.

Desde la aparición de los ordenadores, los ejércitos mundiales comenzaron a preocuparse por obtener el mayor provecho de esta nueva tecnología. En la actualidad su aplicación militar es muy amplia y variada: desde la existencia de bases de datos, que contienen la máxima información posible sobre países y ejércitos amigos y enemigos, hasta el cálculo de las órbitas de los misiles. De hecho, son ya muchos los sistemas de armas de ataque que incorporan un microordenador interno, encargado de controlar y casi garantizar la total precisión del mismo.

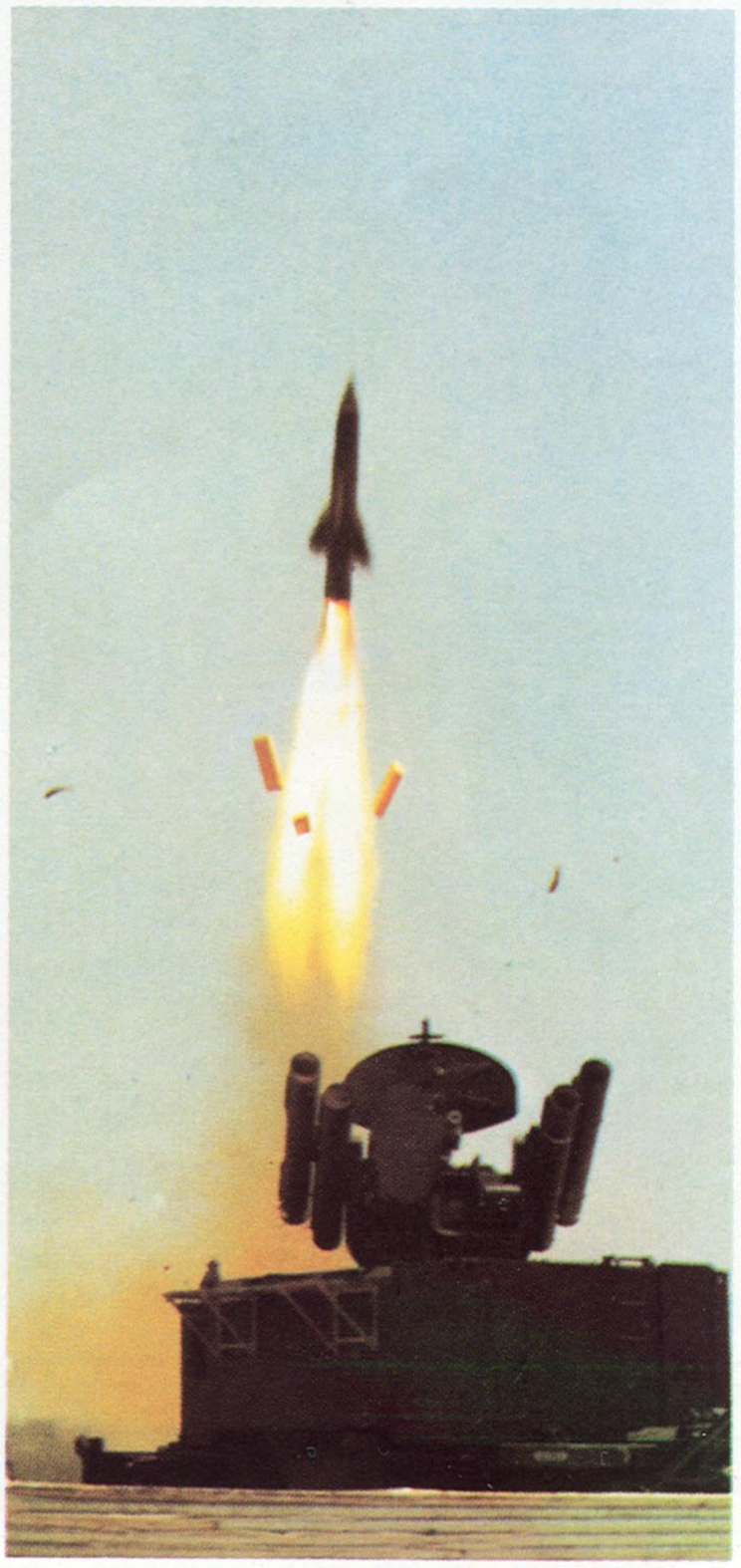
Tampoco es difícil encontrar en las escuelas de guerra militar sistemas ordenadores dedicados a tareas de simulación. Algo parecido a jugar a los «marcianitos», pero con acciones bastante más parecidas a la realidad. Se pueden simular batallas aéreas, navales y marítimas, así como analizar estrategias, recursos disponibles y recursos necesarios, etc.

El papel jugado por los ordenadores puede ser, en algunos casos, incluso trágico. En la mente de muchos estará el desafortunado incidente que ocurrió hace pocos años, cuando una equivocación al introducir los datos en un ordenador del sistema defensivo norteamericano, estuvo a punto de causar un fatal desenlace... el sistema interpretó que se estaba ante una situación de ataque inminente. Afortunadamente, el error se subsanó y los misiles siguieron durmiendo en sus silos.

En los cines de gran parte del mundo está causando furor, en nuestros días, una película cuyo título «War Games» (Juegos de guerra) es ya de por sí lo suficientemente explícito. En ella, un aficionado a los ordenadores personales, David Lightman de diecisiete años, logra acceder a un nuevo juego de video a través de su micro. Lo que realmente ha conseguido es tener acceso a un ordenador del Departamento de Defensa norteamericano. El muchacho, que no parece tonto, también conecta desde su microordenador con el ordenador de la escuela, para cambiar sus notas por otras más brillantes. En cualquier caso, la situación escalofriante es que el precoz protagonista accede al ordenador que controla la alerta previa del Comando de Defensa norteamericano (NORAD). Así, consigue simular un ataque nuclear soviético, que ambas



Es posible que en el futuro la guerra se convierta en un asunto de los ordenadores. Algo semejante a jugar a los «marcianitos», pero ejecutando acciones análogas a las que se plantean en la guerra convencional.



Los equipos microordenadores han entrado ya de lleno en el terreno de los sistemas de armas. En la fotografía se observa el lanzamiento de un misil tierra-aire Roland.

LOS ORDENADORES Y LA GUERRA

potencias creen real. En resumidas cuentas, la película mezcla hechos basados en la realidad, como la posibilidad de acceder a grandes ordenadores desde un micro a través de la red telefónica, con la ficción.

Al parecer, en el mes de enero de 1979, hubo una confusión en el NORAD; una cinta que contenía un juego bélico se tomó como una situación real. Aunque es una situación que probablemente nunca vuelva a ocurrir, sí pone de relieve la importancia de la informática en la defensa.

Un experto americano, Gerard von Diver, desarrolló una idea destinada a evitar la situación real de guerra física. Von Diver puso a punto una simulación basada en la tristemente famosa guerra de las Malvinas. Simulación que posteriormente presentó a los mandatarios de ambas potencias en conflicto.

La simulación consistía en observar una representación gráfica de las islas. A continuación se enviaban tropas a invadir las islas, y, según se fueran ganando batallas, se iban ganando trozos de terreno. Una vez tomada una determinada cantidad de superficie, las islas desaparecían por completo de la pantalla, indicando que se había ganado.

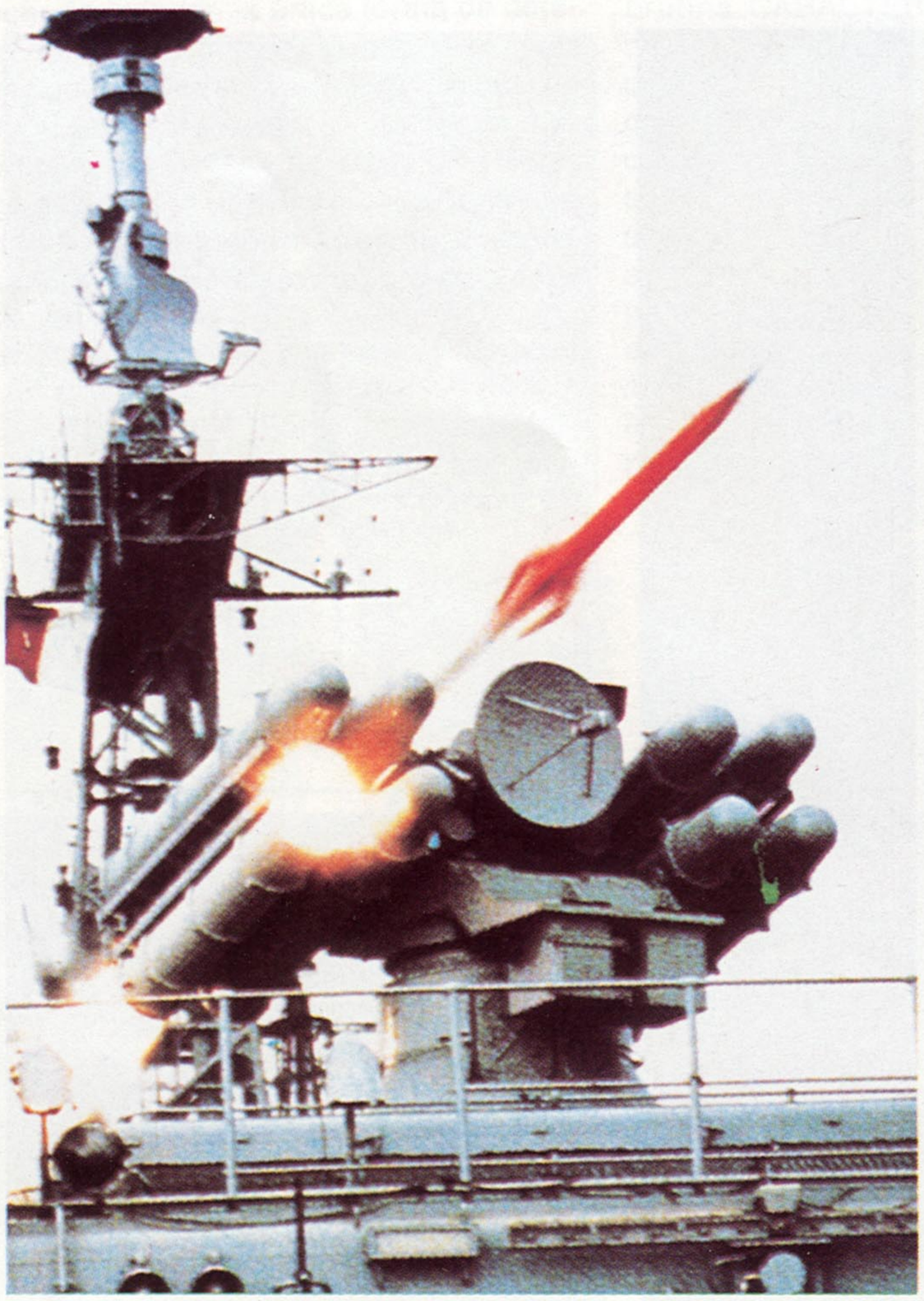
Van Diver perseguía que la guerra hubiera podido dilucidarse sin disparar una sola bala. La simulación que él desarrolló podía ejecutarse en distintos ordenadores, el Apple II, el O. P. de IBM o cualquier ordenador dotado del sistema operativo CP/M de Digital Research.

El objetivo de la simulación era lograr la puntuación máxima. Como punto de partida, se establecía un determinado sistema defensivo en torno a las islas. La puntuación se obtenía en función de lo bien que se defendiera el archipiélago. La victoria final se decantaba por el bando que realizaba una mejor defensa.

Van Diver proponía un ejemplo basado en el hundimiento del destructor «HMS Sheffield». Se habrían introducido en el ordenador los mismos elementos estadísticos utilizados en la guerra real y se habría simulado el resto. El ejército argentino se habría preguntado qué estaba dispuesto a arriesgar para hundir el barco en cuestión.

Finalmente, el ejército victorioso sería aquel dispuesto a utilizar sus tropas, armas y dinero para obtener la misma ventaja estratégica que hubiese obtenido en una guerra real.

Confiemos que en lo sucesivo todos los conflictos bélicos puedan dilucidarse delante de un terminal, relegando al olvido los trágicos campos de batalla.



La electrónica y la informática convierten a los modernos misiles en artefactos destructores de precisión casi infalible. En la fotografía aparece una batería de misiles Crotale rodeando al sistema electrónico de telecontrol.



Aun cuando parezca una posibilidad casi ilusoria, cabe aún confiar en que la guerra se traslade de los campos de batalla al terreno de la simple simulación por ordenador.