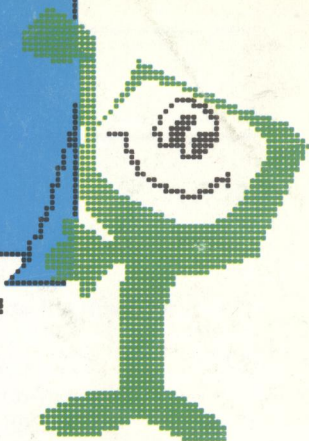


# VIDEO BASIC

20 LECCIONES DE BASIC  
PARA APRENDER CON EL C-64



**INGELEK**



**JACKSON**

*Unidad de disco  
Cómo funciona un lector de disco  
Uso y mantenimiento  
de los disquetes  
Archivos secuenciales  
y aleatorios  
Comandos DOS  
Uso de los archivos  
Videojuego N.º 14*

**14**

**COMMODORE  
C-64**

## VIDEO BASIC

Una publicación de  
INGELEK JACKSON

### Director editor por INGELEK:

Antonio M. Ferrer

### Director editor por JACKSON HISPANIA:

Lorenzo Bertagnolio

### Director de producción:

Vicente Robles

**Autor:** Softidea

### Redacción software italiano:

Francesco Franceschini,

Stefano Cremonesi

### Redacción software castellano:

Fernando López, Antonio Carvajal,

Alberto Caffarato, Pilar Manzanera

### Diseño gráfico:

Studio Nuovaidea

### Ilustraciones:

Cinzia Ferrari, Silvano Scolari,

Equipo Galata

### Ediciones INGELEK, S. A.

Dirección, redacción y administración,

números atrasados y suscripciones:

Avda. Alfonso XIII, 141

28016 Madrid. Tel. 2505820

**Fotocomposición:** Espacio y Punto, S. A.

**Imprime:** Gráficas Reunidas, S. A.

Reservados todos los derechos de reproducción y  
publicación de diseño, fotografía y textos.

©Grupo Editorial Jackson 1985.

©Ediciones Ingelek 1985.

ISBN del tomo 4: 84-85831-23-3

ISBN del fascículo: 84-85831-14-4

ISBN de la obra completa: 84-85831-13-6

Depósito Legal: M-15075-1985

Plan general de la obra:

20 fascículos y 20 casetes. de aparición quincenal,  
coleccionables en 5 estuches.

Distribución en España:

COEDIS, S. A.

Valencia, 245. 08007 Barcelona.

INGELEK JACKSON garantiza la publicación de todos  
los fascículos y casetes que componen esta obra y el  
suministro de cualquier número atrasado o estuche  
mientras dure la publicación y hasta un año después de  
terminada.

El editor se reserva el derecho de modificar  
el precio de venta del fascículo,

en el transcurso de la obra, si las circunstancias del  
mercado así lo exigen.

Octubre, 1985.

Impreso en España.

## INGELEK



## JACKSON

## SUMARIO

### HARDWARE ..... 2

Unidad de disco.

Cómo funciona una unidad de  
disco.

Mantenimiento de los disquetes

### EL LENGUAJE ..... 10

El DOS. Los archivos.

Instrucciones DOS.

SAVE, VERIFY, LOAD, NEW,

SCRATCH, RENAME, VALIDATE,

INITIALIZE.

### LA PROGRAMACION ..... 28

Uso de los archivos

secuenciales.

Movimiento controlado.

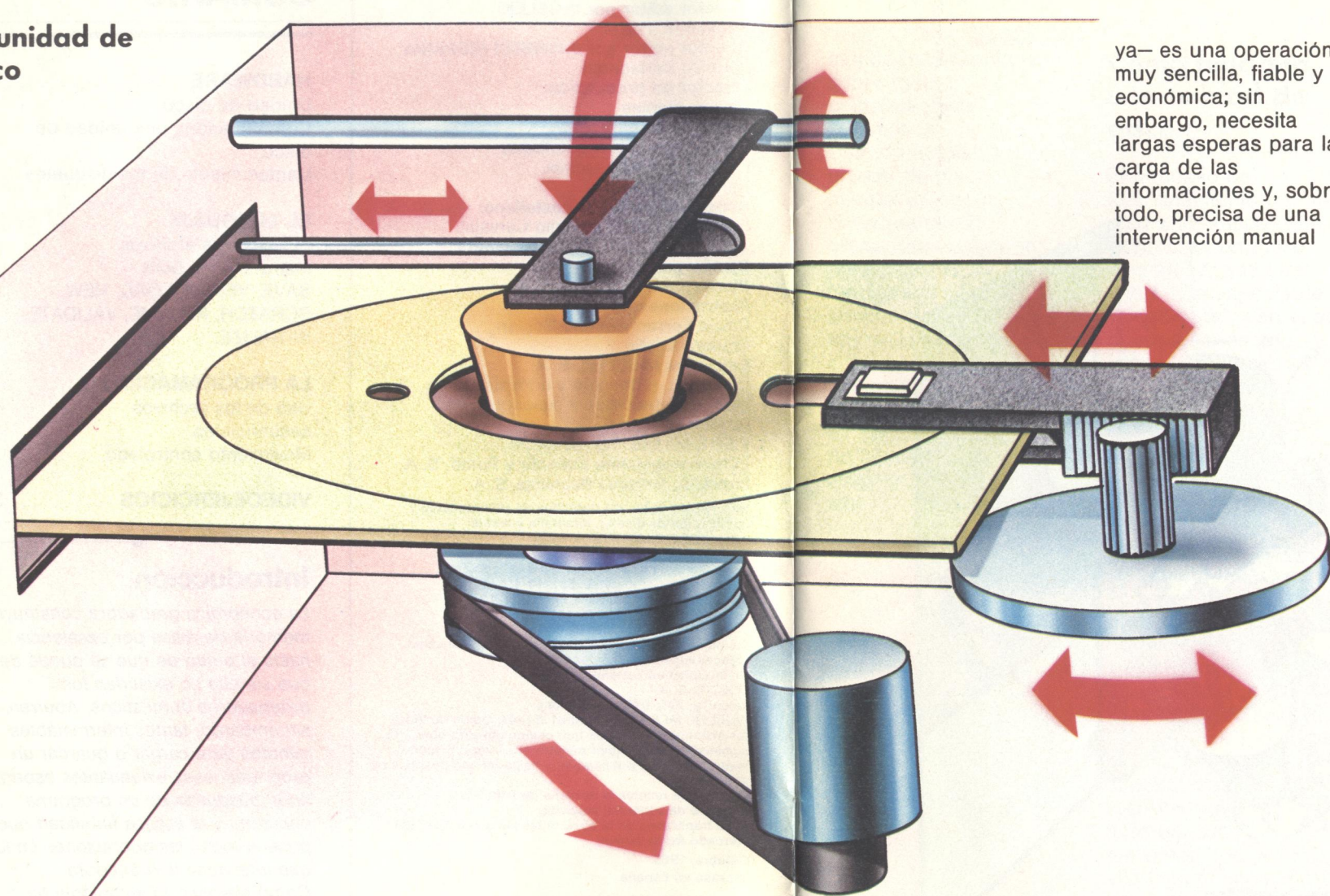
### VIDEOEJERCICIOS ..... 32

## Introducción

*La económica grabadora constituye la memoria de masa por excelencia, hasta el punto de que se puede decir que sin ella no existirían los ordenadores domésticos. Aburren, sin embargo, tantos interminables minutos para cargar o guardar un programa, esas extenuantes esperas en la búsqueda de un programa concreto, y la escasa fiabilidad que poseen todas las operaciones en las que interviene la grabadora. Como siempre, la tecnología ha sabido dar respuesta a todos estos problemas: la unidad de disco (floppy disk drive).*

*Tratar ahora enormes masas de datos, en poco tiempo y con una fiabilidad casi total, resulta fácil; el único truco necesario es conocer bien las instrucciones.*

## La unidad de disco



En lecciones anteriores hemos visto ya como es posible —a través de una grabadora de casete— memorizar en cinta magnética todos los programas y los datos que de otra

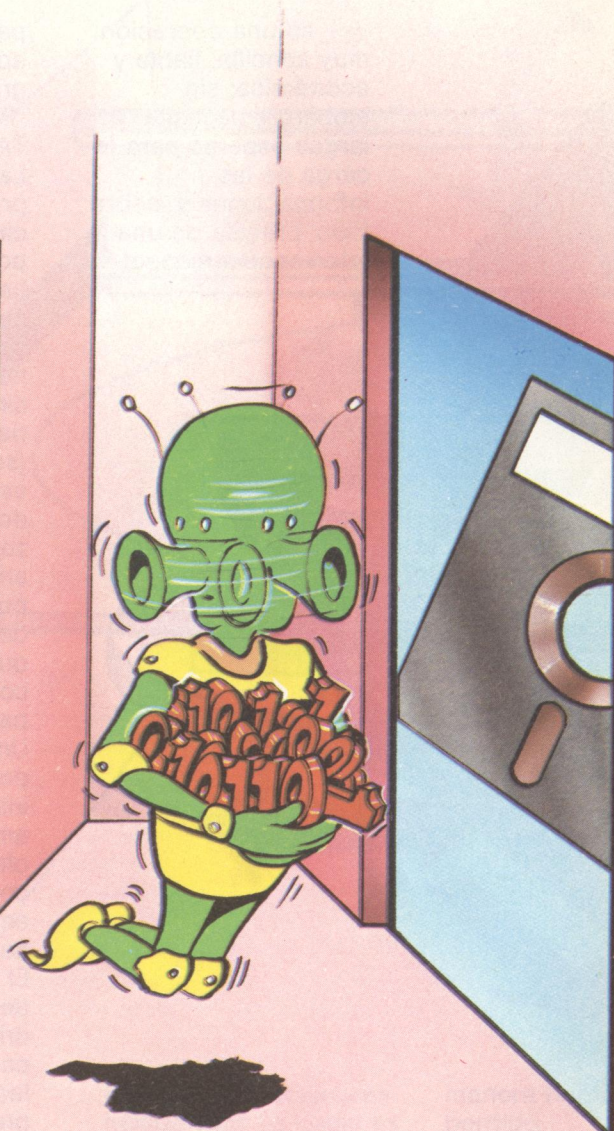
manera hubieramos perdido irremisiblemente al apagar el ordenador. La grabación y la lectura de la cinta —seguramente habrás podido darte cuenta

ya— es una operación muy sencilla, fiable y económica; sin embargo, necesita largas esperas para la carga de las informaciones y, sobre todo, precisa de una intervención manual

para poner en marcha y apagar el motor de la grabadora (el famoso "PRESS PLAY ON TAPE").

La búsqueda de los programas sobre la cinta está garantizada por una técnica bastante empírica, que consiste en contar los segundos, o apuntar el número del contador de cinta del aparato, con el riesgo permanente (sobre todo cuando uno es bastante distraído) de grabar sobre una zona ya grabada anteriormente provocando la pérdida de un programa que quizás nos haya costado largas horas de paciente teclear.

Una de las posibilidades más interesantes para ampliar las prestaciones ofrecidas por tu C64 consiste seguramente en la unidad de disco flexible (floppy disk). El objeto de la unidad de disco (o "disk drive"), de la que nos ocuparemos en esta lección, es precisamente la eliminación de todos los "inconvenientes" antes citados, permitiendo una grabación tan segura como la ofrecida por la grabadora, pero



mucho más rápida, y, sobre todo, completamente automática. Cuando se utiliza una unidad de disco no es necesario, como ocurriría con la grabadora, ocuparse para nada de su funcionamiento; la gestión de las diversas operaciones —como por ejemplo, la grabación o la carga de los programas— tiene lugar en este control directo de la unidad central, sin ninguna intervención exterior de búsqueda o verificación.

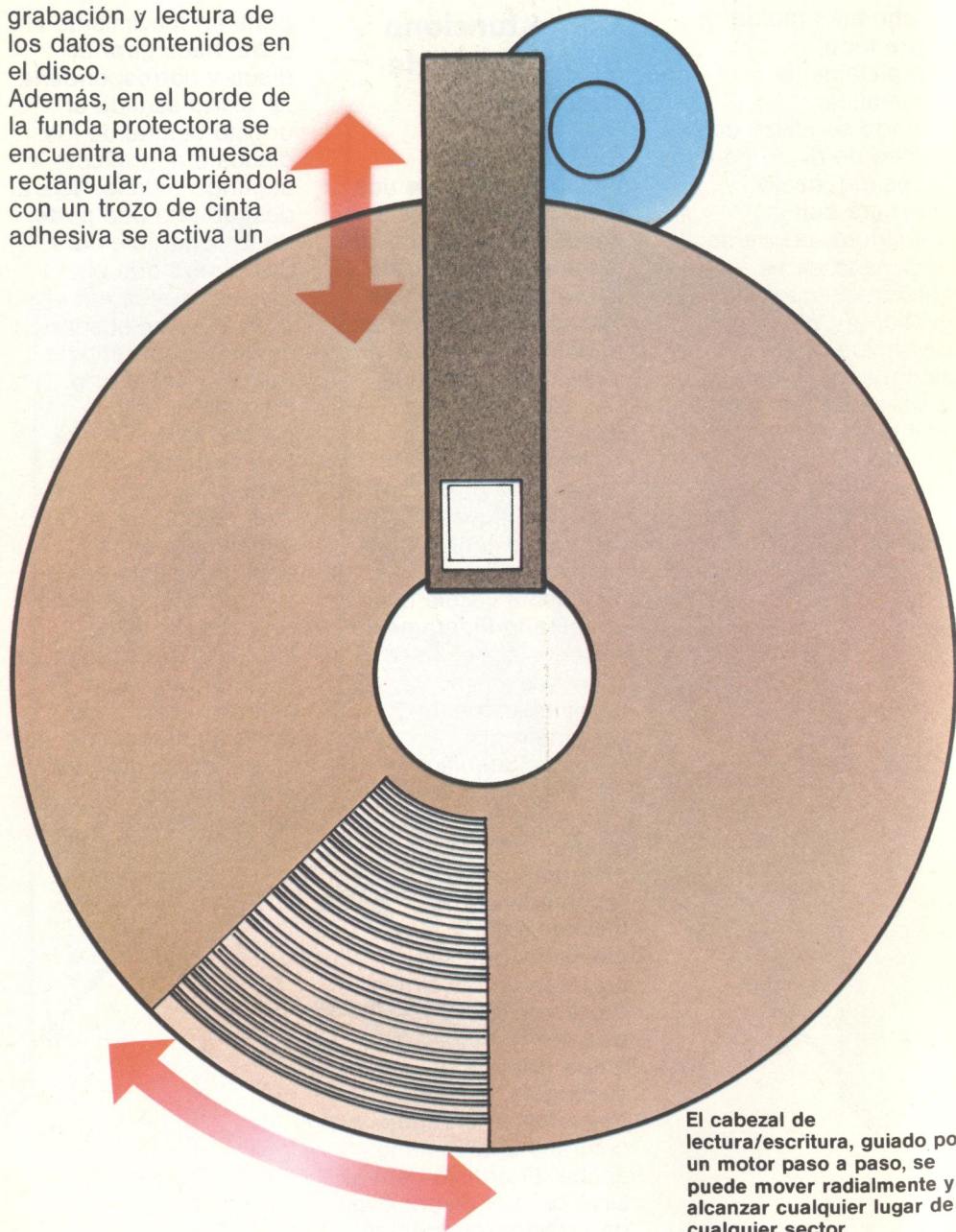
## Cómo funciona una unidad de disco

El principio de funcionamiento de una unidad de disco está basado en un disco magnético, que gira sobre una especie de "tocadiscos" y en el que pueden ser escritas o leídas informaciones por un cabezal de lectura/escritura. La introducción del disco en la unidad se efectúa a través de una ranura existente en su parte delantera, fácilmente visible dado su tamaño (ligeramente superior al del disco; 5 pulgadas y 1/4, aproximadamente 13 centímetros). El disco, también llamado disco flexible ("floppy disk"), consiste en una fina lámina de material plástico, recubierta por los mismos óxidos metálicos que se emplean en las cintas de casete normales, y está contenido en una funda protectora semirígida, en la que hay además un agujero central y una ventana ancha. El orificio central sirve para el dispositivo de arrastre (cuando se

pone en movimiento el drive, hace girar el disco y correspondería en cierto sentido al orificio central de los discos fonográficos); el segundo orificio, deja al descubierto una parte de la superficie del disco para que el cabezal pueda entrar en contacto con el soporte magnético durante la rotación del mismo. Próximo al orificio central se encuentra otro pequeño orificio circular, a través del cual es posible ver uno o más pequeños agujeros existentes en la superficie del disco. Una cédula fotoeléctrica, situada en el lector, es capaz de detectar el paso de estos agujeros, una vez que el disco gire. Estos orificios sirven de referencia para la

grabación y lectura de los datos contenidos en el disco.

Además, en el borde de la funda protectora se encuentra una muesca rectangular, cubriéndola con un trozo de cinta adhesiva se activa un



El cabezal de lectura/escritura, guiado por un motor paso a paso, se puede mover radialmente y alcanzar cualquier lugar de cualquier sector.

interruptor que impide la escritura de datos sobre el disco. Esta precaución debe emplearse cuando no se desee correr ningún riesgo de borrado o de alteraciones accidentales de los datos existentes en el disco.

En el interior del drive están presentes los motores para la rotación del disco y el desplazamiento del cabezal, además de los circuitos necesarios para manejarlos y el interface para comunicar a través del cable de conexión con la unidad central. Además, el sistema de control de los discos está constituido de forma que sea capaz de definir sobre cada disquete virgen (es decir, aún no grabado), una estructura compleja

pero perfectamente definida; de hecho, el disquete cuando está nuevo no es utilizable de forma inmediata, es preciso grabar sobre su superficie una serie de informaciones que son indispensables para su sucesivo buen funcionamiento.

Así, para poder leer un disco, el drive necesita que dicho disco haya pasado por la operación llamada "formateado", es decir, una operación que grabe en el disco en cuestión una especie de "mapa" que posteriormente usará la cabeza de lectura escritura como sistema de referencia para orientarse durante la grabación o la lectura de las informaciones. Sin entrar demasiado en detalles técnicos será suficiente con saber que durante el formateado el disco es subdividido en una serie de invisibles huellas concéntricas, que a su vez son subdivididas en los llamados sectores, que toman este nombre porque son representables exactamente como sectores de circunferencia.

En total, al final del formateado, el disco queda dividido en 683 sectores, cada uno de los cuales es capaz de contener 256 bytes. De estos 683 sectores son de uso efectivo 664 puesto que los otros 19 le sirven al drive para memorizar algunas informaciones para su uso exclusivo. Por lo tanto, la capacidad real de memoria de un disquete es de aproximadamente 170000 bytes (664 \* 256).

## Mantenimiento de los disquetes

Ya hemos visto que a pesar de sus reducidas dimensiones, un disquete es capaz de grabar con precisión una enorme cantidad de informaciones; esta es la razón de que sea muy importante tomar escrupulosamente algunas sencillas, pero indispensables precauciones, parecidas a las que ya hemos visto al tratar de la cinta magnética, pero aún más teniendo en cuenta la relativa fragilidad de los disquetes:

- no tocar jamás la

superficie magnética del disquete (a través del orificio oval). Sobre los dedos hay siempre una fina capa de grasa que puede impedir el contacto correcto entre el cabezal y la cinta;

- evitar que el polvo se deposite sobre el disquete: incluso un microscópico grano de

polvo o de ceniza puede estropear irremisiblemente la capa magnética. La mejor solución es devolver cada disquete a su funda protectora inmediatamente después de usarlo;

- no exponer los disquetes al sol, al calor o a las bajas temperaturas, puesto que podrían producir peligrosas deformaciones del soporte plástico;

- aún cuando el término disco flexible parezca dejar el camino abierto a interpretaciones de lo más variado, se debe evitar absolutamente somerterlos a esfuerzos mecánicos, tratando de no manejarlos con seguridad excesiva, no superponerlos y no



aplastarlos con pesos de ningún tipo;

- alejar todo lo posible los disquetes de imanes, televisores, altavoces, y, en general, cualquier aparato eléctrico;
- hacer una copia de seguridad de todos los disquetes que contengan informaciones importantes, conservándolos además en un lugar distinto de

aquél en que estén los originales;

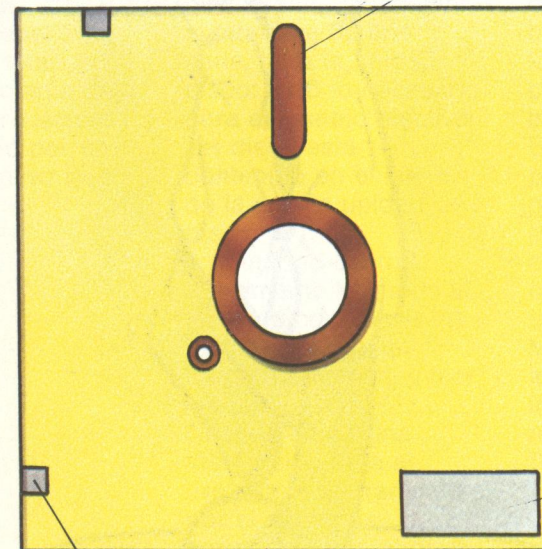
- no dejar nunca los disquetes en el lector cuando el ordenador está apagado: extraerlos siempre antes de apagar la máquina y guardarlos inmediatamente.

Antes de concluir nuestro capítulo es necesario recordar que conviene siempre comprar disquetes de una marca acreditada, de doble densidad de

grabación (doble densidad significa que la calidad del soporte magnético es capaz de responder en forma óptima a las más pequeñas exigencias provenientes del cabezal de lectura, con la casi absoluta seguridad y garantía de no perder —a causa de posibles defectos debidos al grosor del grano magnético— ningún dato o información).

INSERTAR  
ASI

VENTANA  
DE LECTURA/ESCRITURA



PROTECCION  
CONTRA ESCRITURA

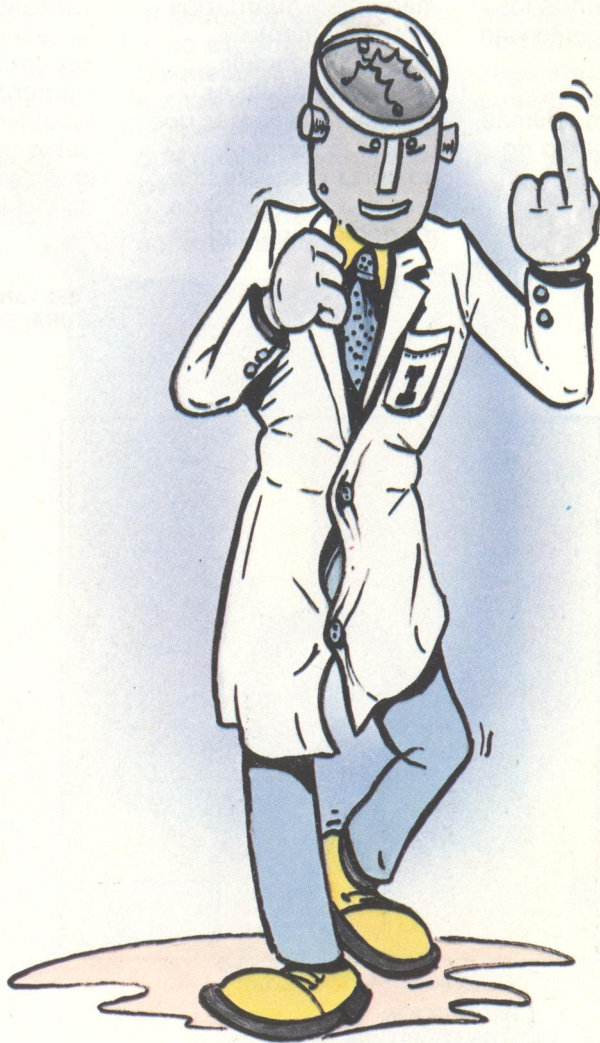
ETIQUETA

## EL DOS

Ahora que ya te has hecho una idea del funcionamiento de la unidad de disco es necesario que nos paremos un momento para hablar de cómo

acontece el intercambio de informaciones entre el ordenador y el disquete.

El manejo de la unidad de disco está confiado a un programa especial



llamado DOS (abreviatura de Disk Operating System, sistema operativo del disco), que se ocupa de coordinar las múltiples actividades y operaciones que realiza el lector.

La función del DOS es, bajo muchos puntos de vista, muy similar a la del intérprete BASIC, dado que permite impartir órdenes y controlar —mediante pocas y breves instrucciones de fácil comprensión— operaciones que no son nada sencillas, haciéndose cargo de las partes más aburridas y repetitivas. De hecho, la totalidad del funcionamiento de un ordenador siempre está manejado por un sistema operativo que

comprueba y coordina en todo momento (sin que el usuario tenga tiempo ni para darse cuenta) el funcionamiento de todo el sistema.

En nuestro caso concreto, el DOS es la parte del sistema operativo que se ocupa de la unidad de disco. Por lo que respecta al C64, el DOS reside en una zona de la memoria ROM situada en el interior del lector; la parte de sistema operativo residente en la memoria central, que se ocupa de manejar el disco, queda reducida por lo tanto, únicamente al interface de tipo serie, para la unidad disco: todo lo necesario para el manejo efectivo del disco está contenido en el interior de la propia unidad de disco.

A la unidad central únicamente le queda la tarea de comunicar las instrucciones a la unidad periférica, y leer las respuestas. Cuando se da una instrucción para el drive, ésta se transmite por el cable de conexión serie, el drive la reconoce y empieza a ejecutarla. En el caso de que la instrucción no

implicara ulteriores intercambios entre unidad central y periférico, la unidad central, una vez comprobada la recepción de la instrucción, continúa con la ejecución del programa en curso; en el caso contrario, esperaría las informaciones de respuesta del drive. Si se intenta enviar otra instrucción al disco, ésta será ejecutada únicamente al final de la instrucción anterior, puesto que es la recepción de la instrucción misma la que permanece bloqueada mientras que la unidad periférica siga con la ejecución de una instrucción anterior. Este sistema de procedimiento, que permite mejorar las prestaciones de todo el conjunto, está completamente controlado por el DOS. Más adelante aprenderemos las instrucciones reconocidas por el DOS.

## Los archivos

Los programas son bastante útiles pero, en verdad, aquello que realmente se desea de un ordenador es el tratamiento de informaciones, datos, números, nombres, direcciones y cantidades.

Cualquier instrucción que pueda ser escrita puede también ser introducida en la memoria del ordenador, y, en consecuencia, memorizada en un disquete.

Un archivo (en inglés, "file"), es un conjunto de informaciones que, por un motivo cualquiera, se desea mantener agrupadas. Ejemplos de archivos pueden ser los nombres de los alumnos de una clase, los números extraídos durante una partida de bingo, los nombres y las direcciones de aquéllos nacidos en un día determinado, o un programa. El DOS pone a tu disposición dos tipos principales de archivo:

— archivos secuenciales: son archivos en los que las informaciones son grabadas unas a continuación de otras, en forma secuencial.

Son los más fáciles de emplear pero no permiten una gran flexibilidad de uso; — archivos aleatorios, (de acceso directo): en este caso, las informaciones se escriben (o se leen) en un punto cualquiera del archivo. Respecto a los archivos secuenciales, el archivo de acceso directo es más práctico y eficiente, pero tiene una estructura ligeramente más rígida, y, por tanto, más complicada de manejar.

Examinemos, por tanto, qué debemos hacer para guardar en los disquetes, reuniéndolos, todos los datos e informaciones que deseáramos que no se perdieran para siempre al apagar el ordenador. Para hacer esto es necesario desde el principio introducir dos términos nuevos, muy útiles cuando se habla de archivos: Registros y Campos.

Hagámoslo (para ser lo más claros posible) a través de un ejemplo.

La guía telefónica de tu ciudad, considerada en conjunto, es un archivo: contiene informaciones ordenadas y clasificadas según un orden bien definido.

Este archivo está constituido por millares de registros, cada uno de los cuales está compuesto por un apellido, un nombre, una dirección y un número de teléfono. Así, un registro puede

ser definido como un grupo de datos relacionados entre sí; cada uno de los datos concretos que introducimos en el registro se llaman, en cambio, campos. En nuestro caso,

existen 4 campos para cada registro: un campo Apellido, un campo Nombre, un campo Dirección y un campo Número de teléfono. La elección del formato

**Sectores y huellas sobre un disquete.**





del registro (es decir, del número de campos) es, por supuesto, arbitraria: habríamos podido considerar, junto con los campos ya citados, también un campo Número de prefijo, que indicara el prefijo telefónico. Hemos dicho ya que en los archivos secuenciales, como indica su propio nombre, las informaciones son archivadas una detrás de otra. Esto significa que para leer (o escribir) la última información del archivo en uso es necesario haber leído (o escrito) antes todas las informaciones anteriores. Esta limitación puede parecer una notable

desventaja (también sería más cómodo para nosotros encontrarnos, cuando lo estuviéramos buscando, con "López, Juan" nada más abrir la guía telefónica), pero lo cierto es que —gracias a la notable velocidad de lectura de los datos sobre el disquete— encontrar un registro aislado contenido en un archivo de dimensiones medias requiere un tiempo de búsqueda bastante razonable. Verificaremos y profundizaremos en este tema en la parte de la lección dedicada a la programación.

## Instrucciones DOS

Cuando tu C64 está conectado a una o más unidades de disco, muchas instrucciones que ya estaban disponibles para el control de la grabadora se hacen válidas también para la unidad de disco. Todas las instrucciones que permiten operar con discos, al igual que ocurría con la grabadora, se pueden emplear tanto en modo inmediato como desde dentro de un programa.

## SAVE

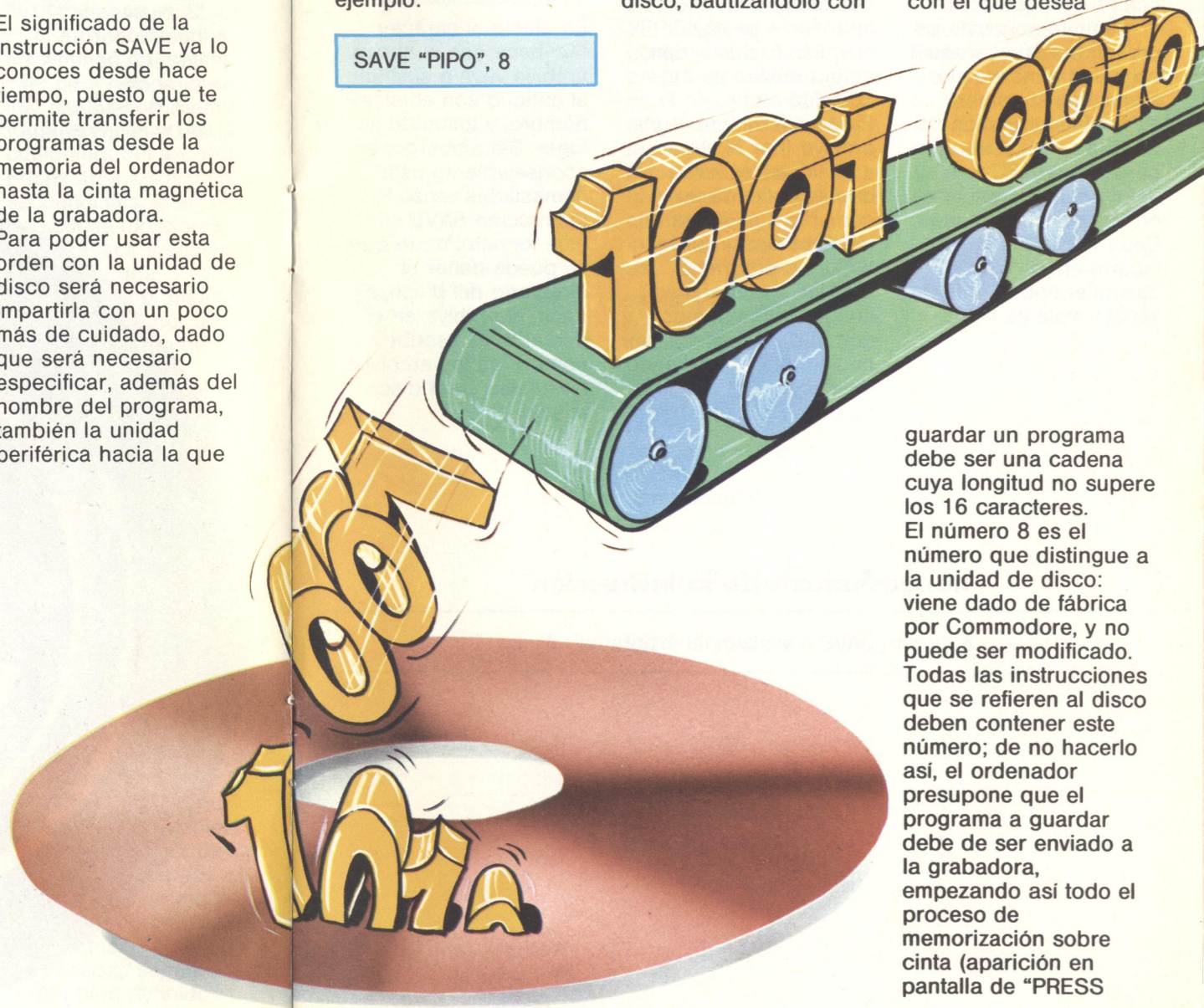
El significado de la instrucción SAVE ya lo conoces desde hace tiempo, puesto que te permite transferir los programas desde la memoria del ordenador hasta la cinta magnética de la grabadora. Para poder usar esta orden con la unidad de disco será necesario impartirla con un poco más de cuidado, dado que será necesario especificar, además del nombre del programa, también la unidad periférica hacia la que

el programa ha de ser enviado. Veamos en seguida un ejemplo:

SAVE "PIPO", 8

Memoriza el programa actualmente en memoria sobre la unidad de disco, bautizándolo con

el nombre de PIPO, y al igual que para la grabadora, el nombre con el que desea



guardar un programa debe ser una cadena cuya longitud no supere los 16 caracteres. El número 8 es el número que distingue a la unidad de disco: viene dado de fábrica por Commodore, y no puede ser modificado. Todas las instrucciones que se refieren al disco deben contener este número; de no hacerlo así, el ordenador presupone que el programa a guardar debe de ser enviado a la grabadora, empezando así todo el proceso de memorización sobre cinta (aparición en pantalla de "PRESS

PLAY ON TAPE", etc.). El drive, antes de grabar el programa, comprueba que sobre el disquete no existe ya un nombre igual a aquél asignado al nuevo archivo: si el nombre existe, la instrucción SAVE no se ejecuta, y parpadea el LED rojo del drive para indicar el error. Cada vez que el drive incurre en un error de cualquier tipo la única señal visible es la del

LED: por lo tanto, es necesario prestarle mucha atención, dado que aunque en apariencia se haya ejecutando una orden, en la realidad no ha ocurrido nada. Para poder escribir un archivo (por el término archivo, te recuerdo que se entiende un conjunto cualquiera de informaciones, por lo tanto, un programa también es un archivo) con el mismo nombre que uno ya existente en el disquete, y perdiendo lógicamente el antiguo, se puede dar entonces

esta instrucción:

```
SAVE "@0 : PIPO", 8
```

En efecto el carácter "@" hace que el nuevo archivo vaya a sustituir al antiguo con el mismo nombre, y tomando su lugar. Sin embargo, es aconsejable no usar demasiadas veces la instrucción SAVE con este formato, dado que se puede dañar el directorio del disco, es decir, el archivo en el que el drive escribe el nombre de los archivos existentes en el disco mismo.

## Sintaxis de la instrucción

```
SAVE "nombre del programa", 8
```

## VERIFY

Después de la instrucción SAVE es aconsejable usar siempre la instrucción VERIFY, para comprobar que el programa memorizado en el disco sea idéntico al que está presente en la memoria. Suponiendo que hayamos memorizado el archivo PIPO, la instrucción se escribirá así:

```
VERIFY "PIPO", 8
```

lo que nos permitirá comprobar que la memorización del programa se ha efectuado correctamente.

Si algo no hubiera funcionado correctamente, siempre podrás repetir la operación de memorización, evitando así la pérdida de todo lo que se encontraba en memoria y querías conservar, por causa de un error o de un mal funcionamiento de algún dispositivo.

## Sintaxis de la instrucción

```
VERIFY "nombre del programa", 8
```

## LOAD

Ahora que hemos guardado un programa en un disquete queremos aprender qué es lo que hay que hacer para cargarlo de nuevo en la memoria. La instrucción encargada de este asunto es una vez más idéntica a la del grabador de casetes. Por lo tanto, si deseáramos recuperar el programa que hemos escrito en el disquete con el nombre de PIPO, deberíamos indicar:

LOAD "PIPO", 8

después de unos pocos segundos el programa se encontrará en la memoria, listo para su uso.

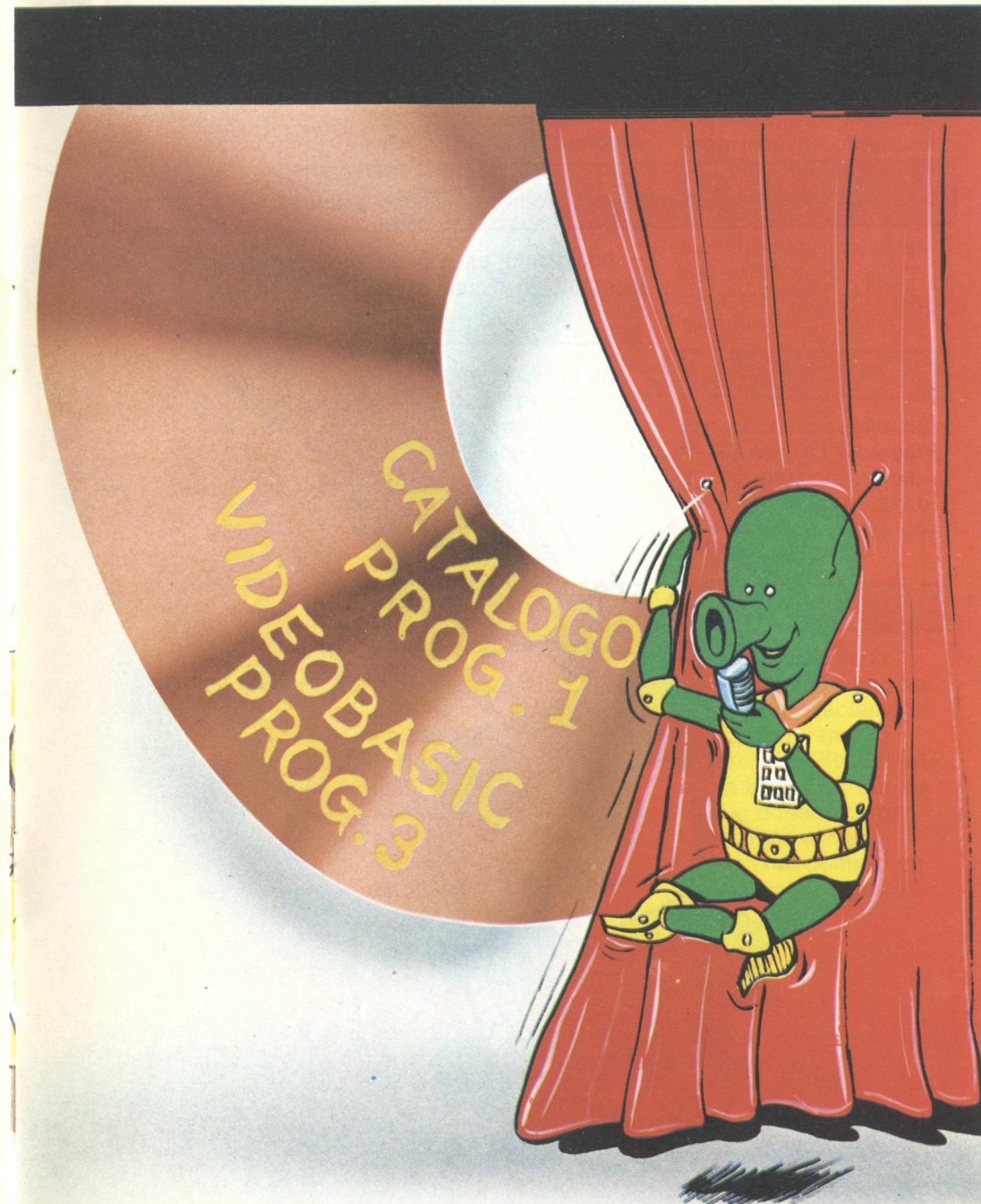
Sin embargo, el ejemplo explicado es bastante especial: estábamos seguros de que en el disquete se encontraba el programa PIPO, y por lo tanto, hemos podido cargarlo sin ninguna duda.

En cambio, en otras ocasiones, el listado de los archivos existentes en el disco nos resulta completamente desconocido: sería cómodo poder disponer de una especie de catálogo en el que leer los archivos que se hayan grabado sobre

ese disquete. Ya hemos explicado que el disco se puede suponer que esté dividido en anillos concéntricos, llamados pistas: pues en la pista número 18, el ordenador memoriza precisamente el índice de los programas residentes en el disco, llamado DIRECTORY (directorío). Este índice es un archivo como los demás y por lo tanto, puede ser cargado con la instrucción LOAD. La instrucción completa a impartir es la siguiente:

LOAD "\$", 8

La luz roja se encenderá durante un instante y el directorio será cargado en la memoria. Una instrucción LIST te permitirá observar su contenido:  
— la primera línea contiene en reverse el nombre asignado al disquete con la instrucción de formateado (ver más adelante).

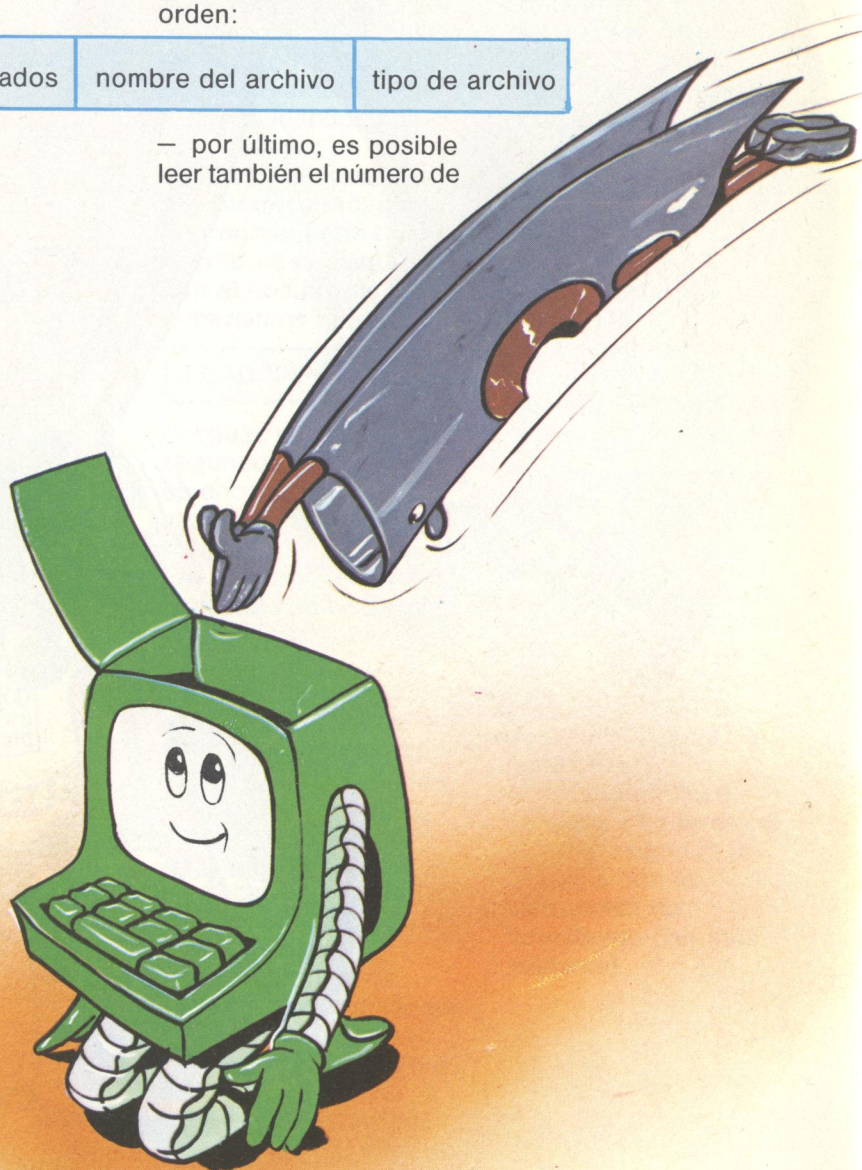


— en las líneas sucesivas están indicados todos los archivos existentes en el disquete según este orden:

los bloques restantes, aún no memorizados, y por lo tanto, disponibles para otros datos.

bloques ocupados	nombre del archivo	tipo de archivo
------------------	--------------------	-----------------

— por último, es posible leer también el número de



Además de permitirse cargar los programas en la forma que ya hemos visto, es posible

usar LOAD sin especificar del todo el nombre del programa, puesto que, y gracias a la opción llamada "PATTERN MATCHING" la parte final del nombre se puede sustituir por el carácter "\*".

Así, por ejemplo,

```
LOAD "PRO*", 8
```

provocará la carga en memoria del primer programa que tenga PRO por iniciales. Pero si en el disco existieran dos programas que empiezan así:

```
PROYECTO  
PROGRAMA 1
```

y desearas cargar el segundo, la instrucción correcta será:

```
LOAD "PROG*", 8
```

La instrucción LOAD "\*", 8 causará, en cambio, la carga del último

programa escrito o cargado en el disco; si antes de la instrucción no se hubiera cargado o memorizado ningún programa, se obtendría entonces el primero de la lista del directorio. También es posible sustituir algunas letras del nombre por el carácter "?". Por ejemplo:

```
LOAD "P??G*", 8
```

provocará la sustitución de ?? por RO y cargará el segundo programa. Si existiera en el disco un tercer programa con el nombre:

```
PREGON
```

la instrucción sería ambigua, dado que no determina exactamente a cuál programa se refiere, si a PROGRAMA 1 o a PREGON, que es el que deseas cargar. En este caso se cargaría el primero en la lista del directorio.

## Sintaxis de la instrucción

LOAD "nombre del programa", 8

## NEW

En todos los ejemplos anteriores siempre hemos supuesto que teníamos a nuestra disposición un disquete ya formateado. Ahora veremos cuáles son las operaciones necesarias para preparar un disquete

virgen para sucesivas grabaciones. En primer lugar es necesario asegurarse de que el disquete a formatear no contenga otros programas de nuestro interés (dado que esta operación los destruirá) y de que no tenga cubierta con cinta adhesiva la muesca de protección contra escritura. Ahora será necesario indicar las siguientes instrucciones:

OPEN 15, 8, 15  
PRINT # 15, "NEW0: NOMBRE. 12"

o bien

PRINT # 15, "N0: NOMBRE. 12"

— la primera instrucción abre el canal (¿te acuerdas de lo que son los canales?) número 15 hacia el periférico número 8 (la unidad de disco), que tiene como dirección secundaria el número 15 (que el drive utilizará únicamente para transmitir o recibir informaciones);  
— la segunda ejecuta el verdadero formateado del disquete, es decir, la subdivisión en pistas y sectores. Es una operación que requiere aproximadamente un

minuto de tiempo. NEW sirve para indicar que el disquete debe de ser tratado como si fuera nuevo, y, en consecuencia, formateado;  
— el número 0 indica que el formateado debe realizarse en el primer disco. En un sistema dotado de dos drives este número, para obtener el formateado del segundo drive, debería haber sido el 1;  
— el nombre es una cadena alfanumérica que puedes elegir a tu gusto, con una longitud máxima de 16 caracteres, mientras que el 12 es un número cualquiera de dos cifras (es decir, que habríamos podido indicar cualquier número, por ejemplo 36 o 89). Estos le serán útiles al drive para la clasificación en el disquete. Como de costumbre, y como ya hemos visto al hablar de la impresora, una vez terminada la operación de formateado es necesario cerrar el canal que habíamos abierto.

CLOSE 15

## SCRATCH

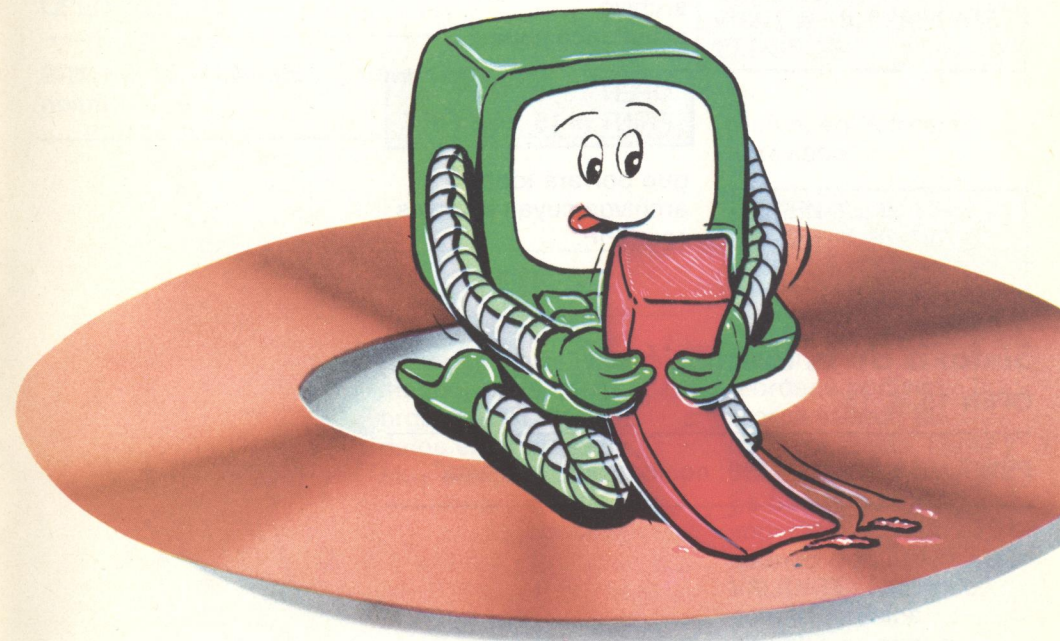
Esta instrucción sirve para borrar archivos del disquete, dejando nuevamente disponible para otras operaciones el espacio que ocupaban anteriormente. También en este caso es necesario abrir el canal hacia la unidad de disco antes de indicar la instrucción. Suponiendo que deseáramos borrar nuestro programa PIPO,

A partir de ahora el disco está listo para grabar las informaciones que deseas.

## Sintaxis de la instrucción

OPEN 15, 8, 15

PRINT # 15, "NEW0: nombre, 12"



la instrucción habría de ser:

```
OPEN 15, 8, 15
PRINT # 15, "SCRATCH0 : PIPO"
```

o bien, en la forma abreviada

```
OPEN 15, 8, 15
PRINT # 15, "S0 : PIPO"
```

Y como de costumbre, una vez terminada la operación, se debe cerrar el canal: CLOSE 15. También en este caso es posible usar el "PATTER MATCHING" para eliminar más de un archivo simultáneamente:

```
OPEN 15, 8, 15
PRINT # 15, "S0:PIP*"
```

que borrará todos los archivos cuyas iniciales sean PIP.

## Sintaxis de la instrucción

OPEN 15, 8, 15

PRINT # 15, "SCRATCH0: nombre del programa"

## RENAME

La instrucción RENAME permite cambiarle el nombre a un archivo; en la práctica, modifica únicamente el nombre en el directorio, sin ir para nada al punto en el que encuentra el programa. Por esta razón es una operación que necesita poquísimo tiempo para ser ejecutada. Por ejemplo, si sobre el disquete tuviéramos un

archivo llamado PRUEBA y quisiéramos llamarlo DEFINITIVO, podríamos indicar:

```
OPEN 15, 8, 15
PRINT # 15, "RENAME0 : DEFINITIVO = PRUEBA"
CLOSE 15
```

o bien, en la forma abreviada

```
OPEN 15, 8, 15
PRINT # 15, "R0 : DEFINITIVO = PRUEBA"
CLOSE 15
```

## Sintaxis de la instrucción

OPEN 15, 8, 15

PRINT # 15, "RENAME0: nuevo nombre=antiguo nombre"

## VALIDATE

A diferencia de la cinta magnética, la organización de los datos en el disquete es muy complicada, con

pedazos de programas situados a lo largo de las diversas pistas y sectores.

A veces puede ocurrir que a base de borrados, grabaciones y puestas al día, la suma de los sectores libres y ocupados señalada en la lista del directorio no dé como resultado 664, como debiera ser. Para poner orden en un disco donde hayan tenido lugar estas situaciones irregulares se utiliza la instrucción VALIDATE:

```
OPEN 15, 8, 15
PRINT # 15, "VALIDATE"
CLOSE 15
```

o bien, en la forma abreviada

```
OPEN 15, 8, 15
PRINT # 15, "V"
CLOSE 15
```

Esta instrucción reordena el disco, reorganizando las distintas zonas, libres y ocupadas, del disquete. Es buena norma impartir de vez en cuando una orden VALIDATE: haciendo esto evitaremos posibles errores o problemas.

## Sintaxis de la instrucción

OPEN 15, 8, 15

---

PRINT # 15, "VALIDATE"

---

### INITIALIZE

---

Esta instrucción sirve, como su propio nombre indica, para inicializar el disquete en el drive. Normalmente la operación de inicialización la efectúa el drive en el momento del encendido: si observas la unidad, verás como la luz roja se enciende durante un tiempo brevísimo y oirás girar el disquete. Pero, sin embargo, puede ocurrir que después de una condición de error por parte de la unidad de disco, algunas operaciones no se puedan realizar. INITIALIZE devuelve la unidad de disco a las mismas condiciones en las que se encontraba en el momento del encendido. Teóricamente, como alternativa a esta instrucción, es posible apagar y volver a encender la unidad de disco (la operación no resulta muy elegante

pero es absolutamente eficaz):

```
OPEN 15, 8, 15  
PRINT # 15, "INITIALIZE"
```

o bien, en la forma abreviada

```
OPEN 15, 8, 15  
PRINT # 15, "I"  
CLOSE 15
```

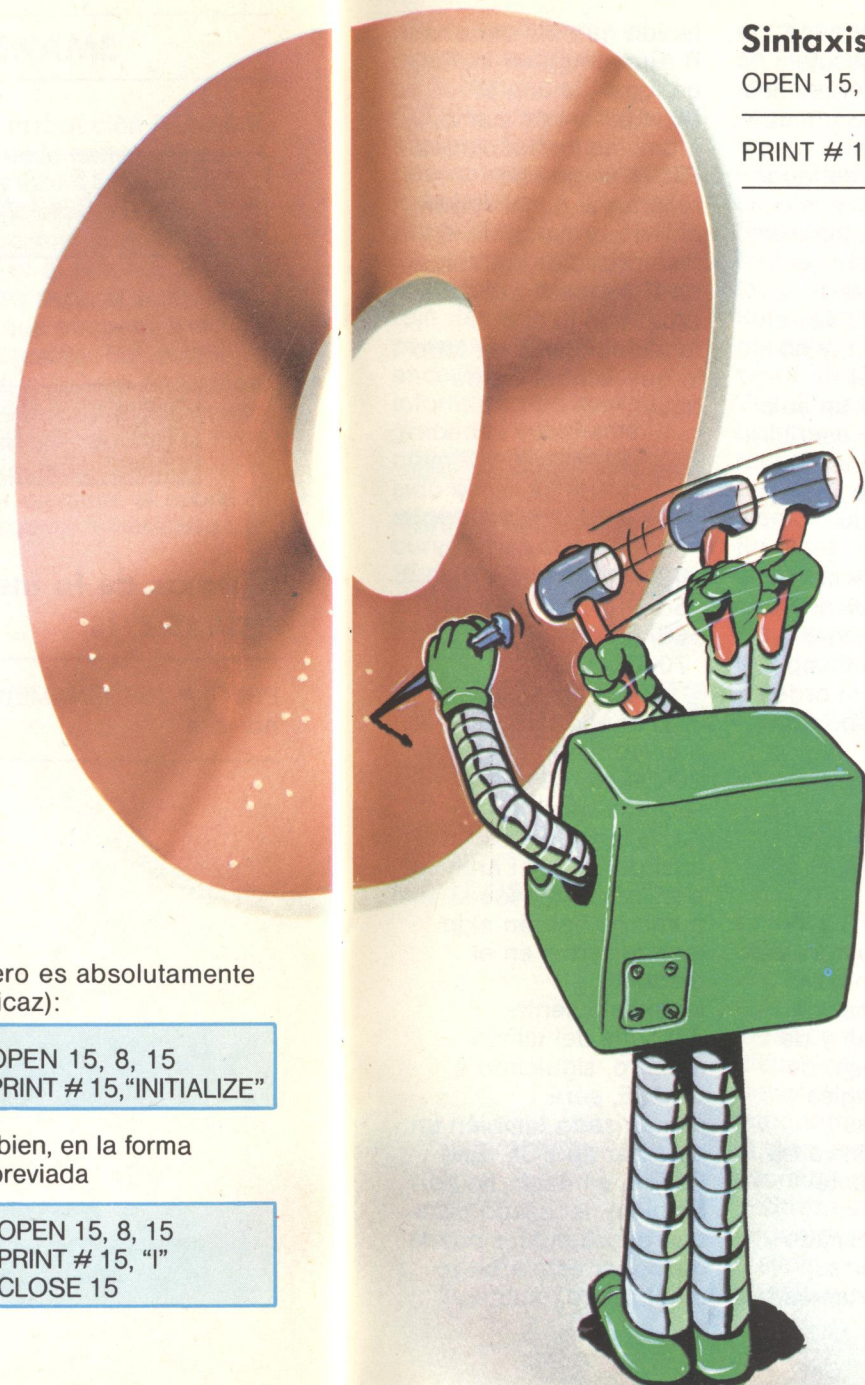
## Sintaxis de la instrucción

OPEN 15, 8, 15

---

PRINT # 15, "INITIALIZE"

---



## Uso de los archivos secuenciales

En esta parte de la lección profundizaremos en el tema de los archivos que hemos comenzado antes.

En primer lugar, antes de poder acceder a un archivo es necesario abrirlo.

Abrir un archivo significa ordenar al DOS que busque las informaciones referentes a dicho archivo: si está en el disco, y, si es así, dónde se encuentra exactamente.

La orden OPEN hace que el disco ejecute todas estas operaciones reservando además una cierta área en la memoria del ordenador que será usada como buffer (llamada también memoria tampón). El buffer permite al drive acceder sólo de vez en cuando al disco, y no cada vez que el programa tiene un solo dato que leer y escribir.

En otras palabras, el acceso al disco tiene lugar a través de bloques de informaciones y no por datos independientes, con el consiguiente gran ahorro de tiempo. Un ejemplo de la orden OPEN para la apertura de un archivo secuencial es el siguiente:

```
OPEN 2, 8, 2, "PRUEBA, S, W"
```

Los caracteres S y W después del nombre del archivo, indican que este habrá de ser de tipo S(ecuencial) y de modo W, es decir, de escritura (del inglés W(rite)). En otras palabras, el archivo ha de ser abierto para escritura, si hubieramos deseado abrirlo para leer su contenido, habríamos

tenido que indicar una R (R(eading), en inglés) en lugar de una W. Al llegar a este punto, todas las instrucciones PRINT # que se refieran al canal recién abierto provocarán la memorización de datos en el disquete. Este breve programa ilustra el funcionamiento de todo lo que acabamos de decir:

```
10 OPEN 2, 8, 2, "PRUEBA, S, W"
20 FOR I = 1 TO 200
30 LET A = RND (0)
40 PRINT A
50 PRINT # 2, A
60 NEXT I
70 CLOSE 2
```

Tras la apertura del archivo serán enviados 200 números al azar respectivamente a la pantalla y al disco; al final del ciclo el archivo será cerrado y los números habrán sido memorizados en el disquete. Inmediatamente después del último número, siguiendo a CLOSE, será memorizado también un carácter de EOF (End Of File, es decir, fin del archivo) necesario para que el ordenador pueda gestionar este archivo en el futuro, cuando

quieras recuperar lo que has escrito. La importancia de CLOSE, y no sólo por el carácter EOF, es relevante incluso viendo trabajar a este programa. En efecto, podrás comprobar que la luz roja del drive no permanecerá siempre encendida: las informaciones serán grabadas solamente cuando el buffer haya sido llenado sucesivamente en el curso de la ejecución. Si no hubiéramos enviado la orden CLOSE, los datos que

```
10 OPEN 2, 8, 2, "PRUEBA, S, R" : REM ARCHIVO DE LECTURA
20 FOR I = 1 TO 200
30 INPUT # 2, A
40 PRINT A
50 NEXT I
60 CLOSE 2
```

permanecen contenidos en el buffer al terminar el ciclo no habrían llegado nunca al disquete, ya que el ordenador estaría esperando nuevas "llegadas" y el archivo habría quedado incompleto. Por lo tanto, siempre es necesario acordarse de cerrar los archivos. Veamos ahora qué es lo que hay que hacer para recuperar los números que acabamos de memorizar en el disquete. En lugar de enviar instrucciones PRINT, ahora deberemos pedir los datos con instrucciones INPUT:

El funcionamiento del programa será idéntico al del anterior, excepto que el ordenador, en lugar de escribir en el disco, leerá. Revisten una importancia especial, en el momento de la escritura de los archivos los caracteres separadores entre los distintos campos, puesto que para poder

leer datos mediante la instrucción INPUT es necesario que los campos hayan sido grabados situando detrás de cada uno de ellos un carácter separador válido, es decir, reconocidos como tal por el disco. Los caracteres separadores reconocidos son el RETURN (CHR\$ (13)) y la coma. El primero se puede insertar con la función CHR\$ (13), o haciendo terminar la instrucción PRINT sin ninguna puntuación (tal y como hemos hecho nosotros en el ejemplo recién visto).

En cambio, el segundo tiene unas limitaciones que provocan a veces la pérdida de datos o la existencia de errores. Ya que requiere una cierta experiencia en su uso, te recomendamos que no lo emplees. Lo mejor, y sin entrar demasiado en detalles, es no realizar nunca PRINT o INPUT múltiples (es decir, del tipo PRINT # 4, A, B, C o INPUT # 4, A, B, C), sino emplear siempre instrucciones más sencillas. Aunque así los programas se hagan algo más largos, la facilidad de uso de esta solución permite



mantener siempre la situación bajo control, por lo menos hasta el momento en que se adquiriera la suficiente familiaridad con los secretos de los archivos.

Otro aspecto a tener en cuenta es que la puesta al día de un archivo secuencial sólo se puede conseguir reescribiendo todo el archivo por completo. La presencia del carácter EOF inmediatamente después del último

```

10 OPEN 15, 8, 15
20 OPEN 2, 8, 2, "PRUEBA, S, R"
30 OPEN 10, 8, 10, "PRUEBA2, S, W"
40 FOR I = 1 TO 200
50 INPUT # 2, A
60 PRINT # 10, A
70 NEXT I
80 FOR I = 1 TO 100
90 PRINT # 10, RND (0)
100 NEXT I
110 CLOSE 10 : CLOSE 2
120 PRINT # 15, "SCRATCH : PRUEBA"
130 PRINT # 15, "RENAME:PRUEBA = PRUEBA 2"
140 CLOSE 15
150 END
    
```

registro impide la inserción de nuevos caracteres al final de la lista. Lo único que entonces se puede hacer es:

- abrir para lectura el archivo a poner al día;
- abrir para escritura

un nuevo archivo; - copiar ordenadamente los registros del viejo archivo en el nuevo, aportando en cada caso las modificaciones que se necesiten y cambiando de nombre el nuevo archivo, atribuyéndole el del antiguo.

Veamos pues qué es lo que deberíamos hacer para añadirle otros 100 números al azar al archivo que habíamos creado anteriormente:

El ejemplo que acabamos de ver era bastante sencillo, dado que sabíamos perfectamente cuantos números teníamos que transferir del viejo al nuevo archivo antes de pasar a la extracción de

nuevos números al azar. La mayoría de las veces el número de lecturas que hay que realizar antes de llegar al EOF es completamente desconocido y ten en cuenta además que intentar leer más allá del último registro provoca un mensaje de error. Por lo tanto, no es posible leer los registros "al azar". Un método casi infalible para solventar el problema es el siguiente: será necesario modificar el programa para escribir en el archivo, como último registro uno o más caracteres especiales, o fácilmente reconocibles, que tú establezcas como tu propio fin de archivo (por ejemplo, en nuestro archivo de números habríamos podido escribir un número negativo; con un archivo de cadenas de caracteres podríamos en cambio escribir una cadena que tuviera para nosotros un significado inequívoco, como "Z\*Z\*Z").

De esta manera, en la fase de lectura será suficiente con insertar una instrucción que compruebe el fin de archivo, expresión que

se verificará en el momento en que el carácter especial sea detectado por el programa.

Veamos un ejemplo al respecto: el programa siguiente escribirá en el disco un archivo con tantas cadenas como las que introduzcamos por el teclado.

Cuando decidas acabar te será suficiente con pulsar RETURN. Antes de cerrar el archivo el ordenador escribirá automáticamente como último registro nuestro código especial "Z\*Z\*Z".

```

10 OPEN 2, 8, 2,
   "PRUEBA3, S, W"
20 INPUT "ESCRIBE LA
CADENA (O SOLO *
PARA TERMINAR)";
A$
30 IF A$ = "*" THEN
GOTO 60
40 PRINT # 2, A$
50 GOTO 20
60 PRINT # 2, "Z*Z*Z"
70 CLOSE 2
    
```

Ahora podrás recuperar y leer las cadenas que acabas de teclear con este segundo programa:

```

10 OPEN 2, 8, 2,
   "PRUEBA3, S, R"
20 INPUT # 2, A$
30 IF A$="Z*Z*Z" THEN
CLOSE 2: END
40 PRINT A$
50 GOTO 20
    
```

Ahora ya no es necesario conocer el número de elementos existentes en el archivo, nuestro "Z\*Z\*Z" advierte a la unidad central de que no debe intentar

```

10 DATA 60, 126, 219, 255, 195, 126, 90, 195
15 POKE 56334, PEEK (56334) AND 254
20 POKE 1, PEEK(1) AND 251
25 FOR I=0 TO 511
30 POKE I+12288, PEEK (I+53248):NEXT
35 POKE 1, PEEK (1) OR 4
40 POKE 56334, PEEK (56334) OR 1
45 FOR C=0 TO 7:READ A:POKE 12288+C,A:NEXT
50 X=0:Y=0:PRINT " @ ":POKE 53272,29
55 GET T$:IFT$=" " THEN GOTO 55
60 T=ASC(T$):C=32:GOSUB 100
65 IFT=17 AND Y<21 THEN Y=Y+1
70 IFT=29 AND X<25 THEN X=X+1
75 IFT=145 AND Y>0 THEN Y=Y-1
80 IFT=157 AND X>0 THEN X=X-1
85 IFT=83 THEN POKE 53272,21:END
90 C=0:GOSUB 100:GOTO 55
100 A=X+Y*40:POKE 1024+A, C
105 POKE 55296+A,3:RETURN
    
```

leer más allá puesto que ya no existen más registros. De esta manera nos resguarda de un posible error.

## Movimiento controlado

Con este programa introducimos una de las técnicas para controlar un carácter o una animación por medio del teclado (con las teclas de cursor). De la línea 35 a la línea 60 se efectúan las operaciones de comprobación de la tecla pulsada y se determinan los desplazamientos relativos.

# EJERCICIOS

Después de haber observado el listado con toda atención contesta a las preguntas, comprobando personalmente tus respuestas mediante el teclado.

```
10 GOSUB 50
20 GOSUB 10
30 RETURN
40 END
50 S=32400:S=S+1
60 PRINT "CUANTAS VARIABLES..."
70 GOTO 30
80 IF S<56 THEN RETURN
90 RETURN
100 K=K+1:GOSUB 100:REM PARA CONTAR CUANTOS GOSUB SE
    NECESITAN PARA LLENAR EL STACK
```

¿Qué error causará este programa?

¿El "IF" en la línea 80 podrá ser alguna vez verdadero?

¿Cuántos GOSUB puede soportar el "GOSUB STACK"?

¿El programa se parará con "END"... o no?



# Su Commodore 64 tiene mucho que decirle. Unidad de Disco.

El Commodore 64 es el resultado de la experiencia internacional de Commodore como líder indiscutible en el mercado de los microordenadores.

El Commodore 64 es el ordenador más completo y potente de su categoría,.... pero todavía tiene mucho que decirle.

Por ejemplo su Unidad de Disco.

**commodore 64**

Sienta como aumenta notablemente la capacidad de memoria de su C-64, como agiliza la carga y descarga de programas y facilita la localización, casi instantánea, de cualquier dato.

Amplie las posibilidades de su C-64, descubriendo su extensa gama de periféricos.

Ahora que ya sabe que su Commodore 64 tiene todavía mucho que decirle, prepárese a conocerle mejor.

**PRINCIPALES CARACTERISTICAS**  
- 170 K de capacidad - Ficheros secuenciales y relativos y de acceso directo - Unidad inteligente, con sistema operativo incorporada.

  
**commodore**



**Microelectrónica y Control**

c/ Valencia, 49-53 08015 Barcelona - c/ Princesa, 47 3.º G 28008 Madrid  
Unico representante de Commodore en España.