

**MANUAL DEL  
USUARIO DDI-1  
UNIDAD DE DISCO  
CP/M 2.2  
LOGO**

**AMSTRAD  
CPC-464**

**NOTICIA IMPORTANTE**

**NUNCA ENCIENDAS O APAGUES  
EL SISTEMA DE DISCOS CON  
UN DISKETTE EN LA UNIDAD  
DE DISCO**

**SIEMPRE ENCIENDE LA(S) UNIDAD(ES)  
DE DISKETTE ANTES DE ENCENDER  
EL ORDENADOR**

Editado por **INDESCOMP, S.A.**  
Avda. Mediterráneo, 9 - 28007 MADRID (ESPAÑA)

Derechos reservados en lengua española: **INDESCOMP, S.A.**

Traduce, compone e imprime: **CONORG, S.A.**

**I.S.B.N.: 84-86176-16-6**

**Depósito Legal: M-12932-1985**

# AMSTRAD

## UNIDAD DE DISCO & INTERFACE

### DDI-1

**Enhorabuena. Te has convertido en el orgulloso propietario del Amstrad DDI-1. Pronto descubrirás lo acertado de tu elección, y que tu unidad es digna de la inversión efectuada.**

Su velocidad te permitirá cargar y guardar ficheros en segundos, lo que tardaría varios minutos usando cinta en cassette. Además, cada diskette te permite almacenar hasta 180K Bytes de información sobre una sola cara del disco (360K en ambas caras). En otras palabras, todos tus programas utilitarios en un disco, 10 o más juegos en un disco, todos tus ficheros con cartas patrón en un disco, la base de datos correspondiente a tu sistema de clasificación en un disco, y así sucesivamente.

Además, puedes efectuar accesos a los ficheros a tu propia **discreción** -'refrencialmente'- lo que significa que no tienes que pasar laboriosamente a través de todos los ficheros que no quieres hasta alcanzar el que deseas emplear.

La operación con diskettes se convierte en imprescindible cuando quieres escribir cualquier clase de programa. Cuando estás desarrollando cualquier programa es muy conveniente de vez en cuando, sacar una copia de él, por ejemplo para el caso de que al intentar ejecutarlo descubras que te has olvidado un 'ON BREAK GOSUB', que al estar acechando, te pide siempre escapar de él y regresar al modo directo. Y si en hacer la copia tardas 5 segundos en lugar de 5 minutos, obviamente ¡habrás adelantado mucho más!

Al desarrollar programas en **ensamblador**, la conveniencia de poder ensamblar desde el disco y luego comprobar en unos segundos, en lugar de minutos, será un favor tanto para el programador avanzado, como para el principiante que está intentando abrirse camino a través del **código máquina** por primera vez.

La inclusión de CP/M abre un vasto panorama de experiencia en programación al usuario. El modo de 80 columnas le asegura que los programas requerirán modificaciones mínimas para ser ejecutados en el CPC464 -y el número de libros escritos sobre el tema de CP/M y cómo se usa, sobrepasa fácilmente los números de tres cifras.

Diciéndolo brevemente, has hecho exactamente la decisión correcta al subir de grado y basar las operaciones en diskette. Te ayudará y acelerará tu comprensión y apreciación de la **computación** en una manera que simplemente no es comparable con sistemas basados sólo en cintas de cassette.

Además, el **formato compacto** de los discos flexibles de 3 pulgadas, con su carcasa plástica robusta (con su "cierre" protector que cubre el área de los cabezales) impide el daño accidental y provee una colocación favorable y firme del mecanismo de arrastre.

Se suministran dos sistemas operativos:

AMSDOS es una ampliación del sistema de archivo en cassettes del CPC464 BASIC, y permite acceso al interpretador BASIC.

CP/M 2.2 es el sistema operativo en disco standard del Z80 con acceso **referencial** con más de 5000 programas comerciales -que van desde las aplicaciones contables e industriales hasta los segundos lenguajes y análisis científicos y de ingeniería. Gracias a la inteligente implementación del CP/M en el CPC464, los ficheros del CP/M pueden registrarse libremente en un disco junto con fichero AMSDOS, que son automáticamente **etiquetados** con la clase de fichero apropiada.

El famoso Dr. LOGO de Digital Research ha emergido como el medio más universal para la educación y la enseñanza -combinando la **cordialidad** con el usuario que es específica de los **gráficos con tortuga**, con una elevada potencia de procesamiento. Dr. LOGO está reconocido como la implementación más completa del LOGO disponible en el mercado, y se suministra libre como parte del paquete Amstrad DDI-1.

# AMSOFT

A division of

# AMSTRAD

CONSUMER ELECTRONICS PLC

© Copyright 1984 AMSOFT, AMSTRAD Consumer Electronics plc

Neither the whole or any part of the information contained herein, or the product described in this manual may be adapted or reproduced in any material form except with the prior written approval of AMSTRAD Consumer Electronics plc ('AMSTRAD').

The product described in this manual and products for use with it are subject to continuous development and improvement. All information of a technical nature and particulars of the product and its use (including the information and particulars in this manual are given by AMSTRAD in good faith. However, it is acknowledged that there may be errors or omissions in this manual. A list of details of any amendments or revisions to this manual can be obtained by sending a stamped, self addressed envelope to AMSOFT Technical Enquiries. We ask that all users take care to submit their reply paid user registration and guarantee cards.

AMSOFT welcome comments and suggestions relating to the product or this manual.

All correspondence should be addressed to:

**AMSOFT**  
169 Kings Road  
Brentwood  
Essex CM14 4EF

All maintenance and service on the product must be carried out by AMSOFT authorised dealers. Neither AMSOFT nor AMSTRAD can accept any liability whatsoever for any loss or damage caused by service or maintenance by unauthorised personnel. This guide is intended only to assist the reader in the use of the product, and therefore AMSOFT and AMSTRAD shall not be liable for any loss or damage whatsoever arising from the use of any information or particulars in, or any error or omission in, this guide or any incorrect use of the product.

Dr LOGO and CP/M are trade marks of Digital Research Inc

Z80 is the trademark of Zilog Inc.

AMSDOS and CPC464 are trademarks of AMSTRAD

First Published 1984

Compiled by Roland Perry and Ivor Spital with acknowledgements to Locomotive Software Ltd

Published by AMSTRAD

Typeset by AMSOFT Computer Graphics

AMSTRAD is a registered trademark of AMSTRAD Consumer Electronics plc. Unauthorised use of the trademark or word AMSTRAD is strictly forbidden

**Manual AMSTRAD Unidad de Disco & Interface DDI-1**

# Contenido

## Fundamentos

---

- F1 Instalación
- F2 Los diskettes
- F3 La carga de programas/juegos
- F4 Presentación de AMSDOS y CP/M

## Capítulo 1 Preparación de Discos de Trabajo

---

- Disco maestro de 'respaldo'
- Un disco de trabajo SISTEMA/UTENSILIO
- Un disco con sólo BASIC
- Discos AMSTRAD BASIC 'llave en mano'
- Discos y programas en CP/M 'llave en mano'
- Configuración de discos
- Arranque y auto-arranque de un paquete CP/M 'llave en mano'

## Capítulo 2 Lo primordial del AMSDOS

---

- Directorio del disco
- Nombres de ficheros y clases de ficheros en AMSDOS
- Construcción del nombre del fichero, cabeceras y comodínes
- Ejemplos de uso de comandos AMSDOS en un programa
- Guardar variables y efectuando un 'volcado' de pantalla
- Guía de referencia de los comandos AMSDOS
- Copia de ficheros
- Guía de referencia de los mensajes de error AMSDOS

## Capítulo 3 Lo primordial del CP/M

---

- Operando con CP/M
- Pistas del sistema CP/M
- Sector para configuración
- Códigos de control en consola
- Inscripción de un disco en el sistema
- Comandos directos de consola
- Comandos 'transeúntes'
- Copia de ficheros y discos
- Gestión del sistema
- Generación de discos

## **Capítulo 4 Introducción al LOGO**

---

Qué es LOGO

Los 'procedimientos' con Dr. LOGO

Revisión de programas y procedimientos

Sugerencias en la operación

Resumen de 'primitivos' con Dr. LOGO

Tratamiento de palabras y listas

Operaciones aritméticas

Operaciones lógicas

Variables

Procedimientos

Revisión

La pantalla de texto

La pantalla de gráficos

Los gráficos con la 'tortuga'

Teclado, joystick

Sonido

'Primitivos' del sistema

Variables del sistema

Propiedades del sistema

## **Capítulo 5 Información técnica para el usuario**

---

Cabeceras

Requisitos de almacenamiento

Mensajes de error

Mensajes del AMSDOS

Mensajes del BIOS

Organización del diskette

Interceptación de los 'saltos' -por el AMSDOS

Re-interceptación de los 'saltos' -por el usuario

Parámetros devueltos

Apelaciones interceptadas

## **Apéndices**

---

Apéndice 1 Glosario de términos empleados

Apéndice 2 Acuerdo para la Licencia del Programa al Usuario Final

Apéndice 3 Índice

# AMSTRAD DDI-1

## LOS FUNDAMENTOS

### Fundamentos 1: Instalación

#### Conectando el Cable de Red

La unidad de diskette Amstrad opera con un suministro de energía de 220-240V y 50Hz.

El Cable de Red está situado en la parte posterior de la unidad. Coloca una Clavija de Red apropiada al extremo del Cable de Red. Si se usa una clavija de 13 Amp (BS363), deberá emplearse un fusible de 3 Amp. El fusible de 13 Amp suministrado con una nueva clavija NO debe usarse. Si se emplea cualquier otra clase de Clavija, debe instalarse un fusible de 5 Amp, bien sea en la clavija o Adaptador, o en la Placa de Distribución.

#### IMPORTANTE

Los hilos en este Cable de Red son del color que corresponde al código siguiente:

**Azul : Neutro**  
**Marrón : Activo**

Ya que los colores de los hilos en el Cable de Red de este aparato puede que no corresponda con las marcas de color que identifican las patillas en tu Clavija, procede de la manera siguiente:

El hilo de color Azul debe conectarse al terminal que está marcado con la letra 'N' o está coloreado en Negro.

El hilo de color Marrón debe conectarse al terminal que está marcado con la letra 'L' o está coloreado en Rojo.

Desconecta la Clavija de Red del Zócalo de Suministro de Energía, cuando no lo estás usando.

Nunca intentes quitar ningún tornillo, ni abrir la carcasa de la unidad de diskette. Obedece siempre el aviso en la Etiqueta de Características que está pegada por debajo de la carcasa de la unidad de diskette.

**AVISO DENTRO HAY PARTES CON TENSION. NO QUITES NINGUN TORNILLO**

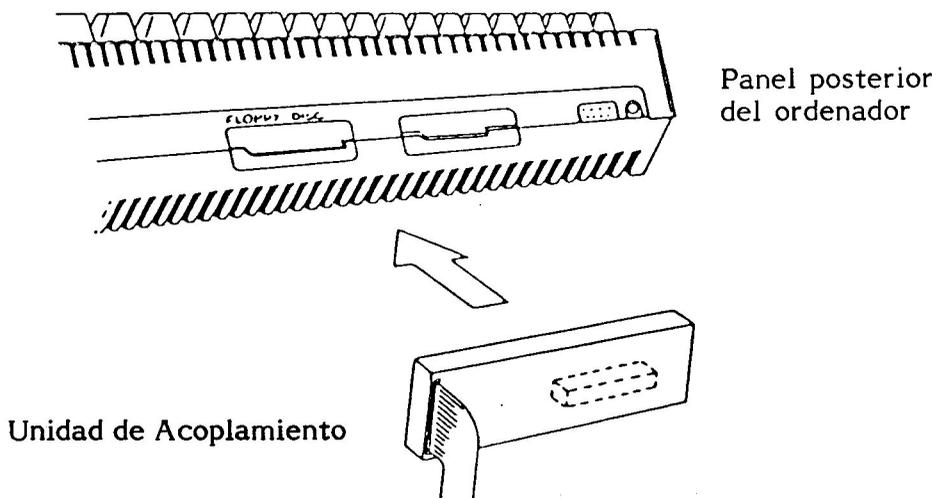
El conmutador POWER ON/OFF, que enciende o apaga la unidad o unidades de diskettes está situado en el panel posterior de dicha unidad.

Asegúrate siempre que has quitado cualquier diskette que pudiera haber en la(s) unidad(es) antes de apretar o quitar el conmutador de energía (antes de POWER ON/OFF).

Aprieta siempre el conmutador POWER (ON) de las unidades de diskette, antes de apretar el conmutador POWER (ON) del ordenador.

### Conexión del DDI-1 al Ordenador

1. Comprueba que el ordenador y la(s) unidad(es) de disco están apagadas (interruptor quitado).
2. Enchufa la Unidad de Interface firmemente en el conector de borde marcado FLOPPY DISK en el panel posterior del ordenador (véase Figura 1).



3. En el extremo del cable plano de la Unidad de Acoplamiento, encontrarás una clavija que se emplea para conectarlo a la unidad de disco principal (Drive A; véase Figura 2).

La clavija que encuentres ligeramente más atrás en el cable (véase Figura 2) es para usarse únicamente si se conecta una segunda unidad de disco (Drive B); i.e. si has adquirido un Amstrad FD-1 adicional.

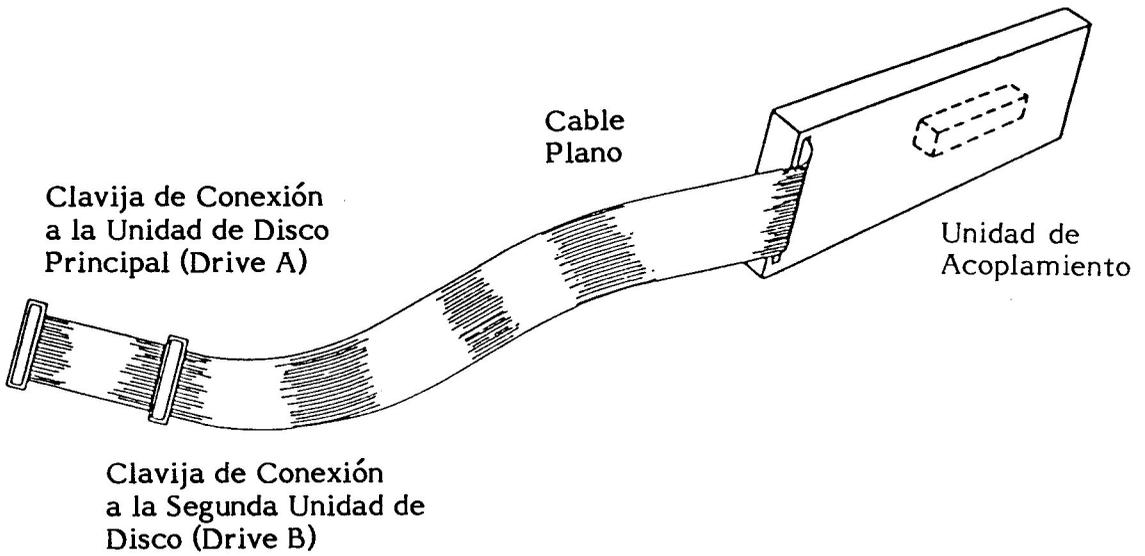
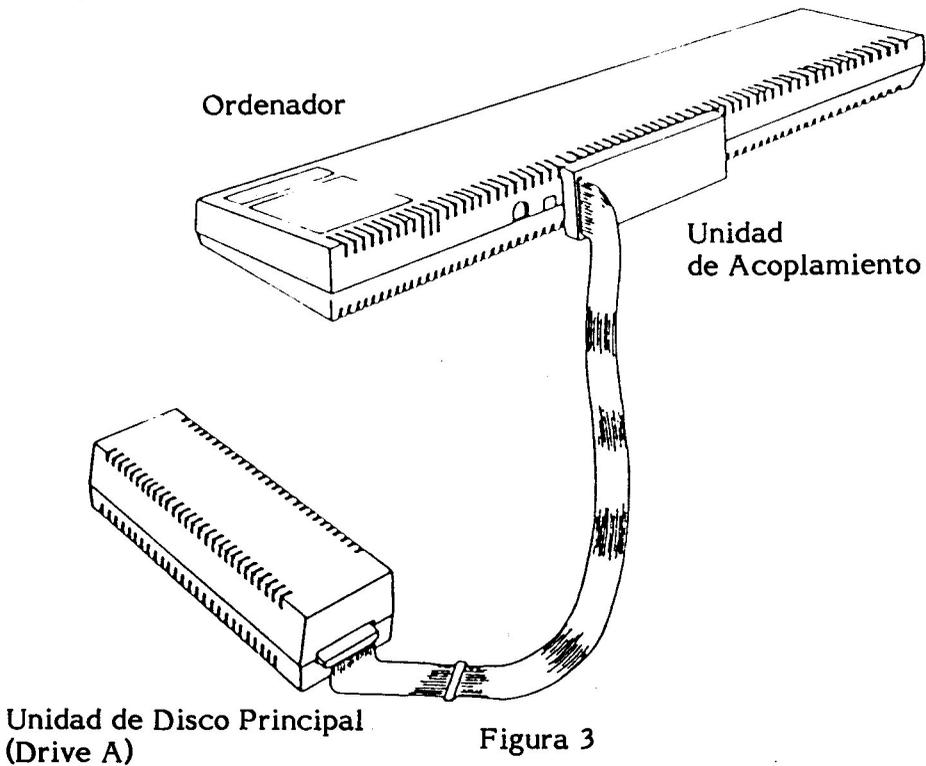
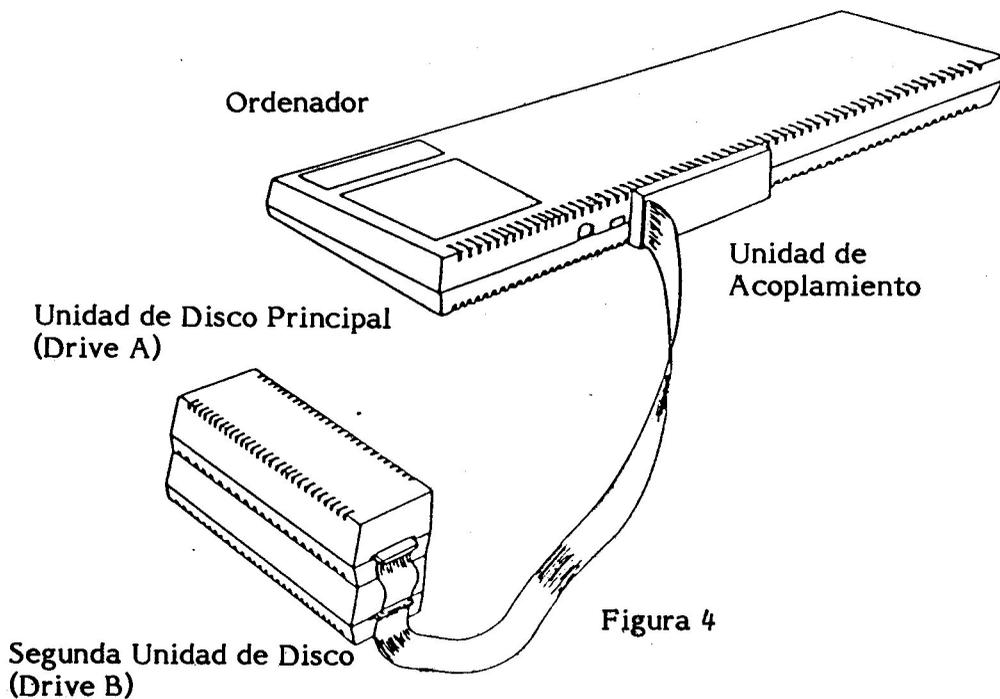


Figura 2

4. Si estás operando con una sola unidad de diskette, conecta el ordenador y la unidad tal y como se muestra en la Figura 3.



5. Si estás operando con 2 unidades de diskette, conecta el ordenador y las unidades tal y como se muestra en la Figura 4.

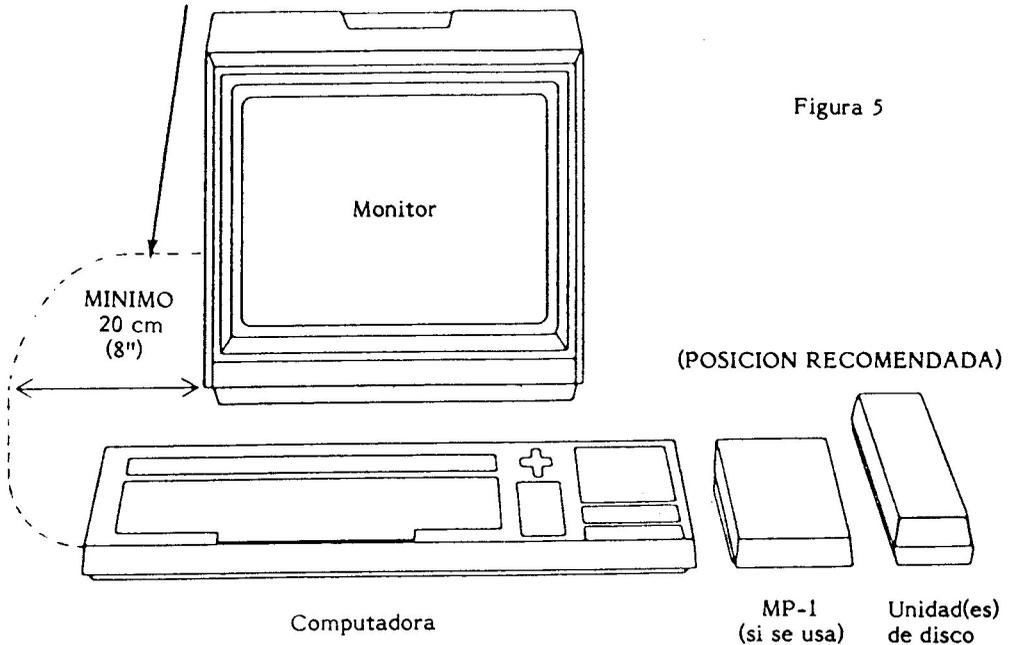


Aprieta el interruptor de red de la(s) unidad(es) de disco, y luego el del ordenador. Tu sistema está ahora preparado para trabajar con diskettes.

## IMPORTANTE

- 1) Para una fiabilidad máxima en la información, no coloques tu(s) unidad(es) más cerca de 20 cm (8") con respecto al lateral izquierdo del monitor (Figura 5). Se recomienda que coloques tu(s) unidad(es) de disco a la derecha de la computadora (o del alimentador/modulador MP-1, si se usa).

NO COLOQUES TU(S) UNIDAD(ES)  
DE DISCO DENTRO DE ESTE AREA



Asegúrate siempre que tu(s) unidad(es) de disco no estén situadas cerca de una fuente de interferencias/ruido eléctrico que pudiera provocar errores en la información.

- 2) Asegúrate siempre que tu(s) unidad(es) y tus discos no están colocados cerca de intensos campos magnéticos tales como los producidos por televisores, altavoces de alta fidelidad, etc.
- 3) El cable plano desde la unidad de acoplamiento hasta tu(s) unidad(es) no debe situarse paralelamente a los Cables de Alimentación de la(s) unidad(es) de disco ni al del monitor, ni tampoco cerca del lateral izquierdo del monitor.
- 4) Nunca bloques ni cubras las ranuras de ventilación de las partes superior, inferior y laterales de la(s) unidad(es) de disco.
- 5) No uses ni guardes la(s) unidad(es) de disco bajo la luz directa del sol, o en ambientes excesivamente calientes, húmedos o polvorientos, ni en sitios sujetos a fuertes vibraciones.

## Fundamentos 2: Los Discos

La unidad de disco Amstrad usa discos 'flexibles' compactos. Recomendamos que para una transferencia fiable de datos hasta y desde el disco, uses sólo discos flexibles compactos Amsoft CF-2. Sin embargo, los diskettes hechos por fabricantes de calidad, también pueden emplearse.

Cada cara de un diskette puede usarse separadamente. Un disco debiera insertarse con su 'etiqueta' mirando hacia la parte externa de la unidad, y con la cara que desees usar hacia arriba (véase Figura 6).

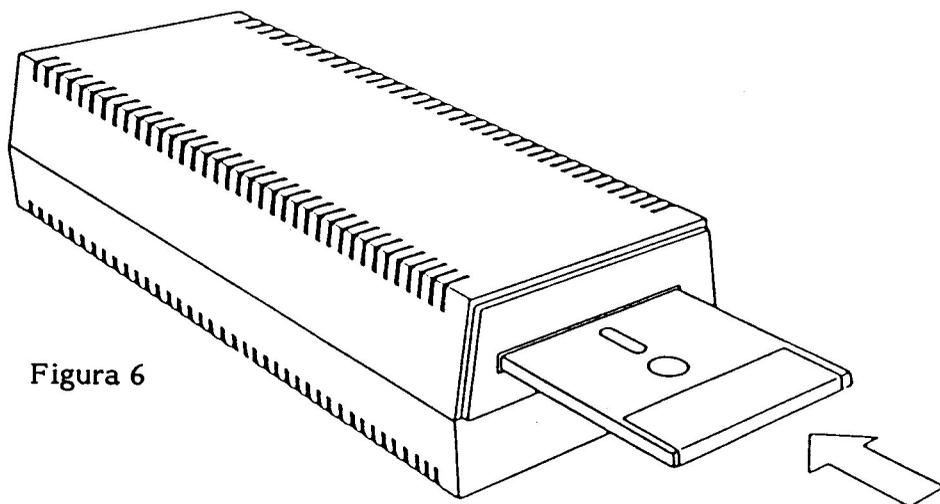


Figura 6

### Protección ante Escritura

En la esquina de la izquierda de cada cara de un diskette en blanco, verás una flecha señalando hacia un pequeño hueco con su cierre. Es el llamado hueco de Protección ante Escritura y facilita la protección frente al borrado y a la **sobre-escritura**. Véase Figura 7.

Hueco de Protección  
ante Escritura

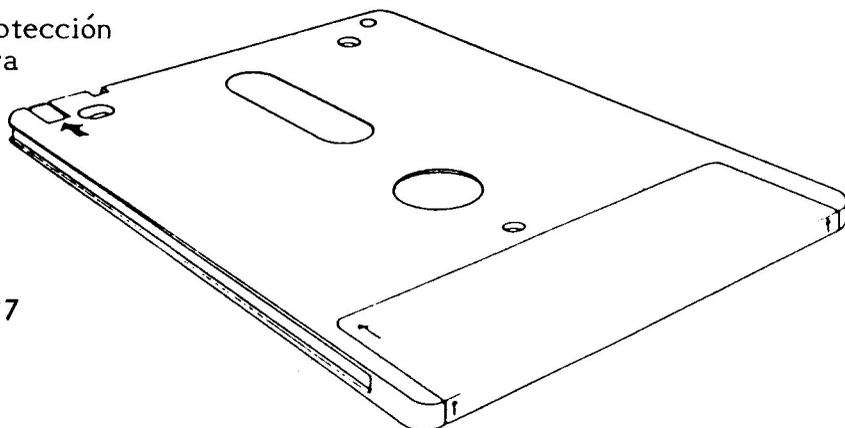


Figura 7

Cuando el hueco está obturado, el ordenador puede **escribir** información en el diskette. Sin embargo, cuando el hueco está abierto, no estará permitido que se escriba información sobre el disco, permitiéndote así evitar el borrado accidental de programas valiosos. Diversos fabricantes de discos **flexibles** compactos, emplean diferentes mecanismos para cerrar y abrir el hueco de protección ante escritura. La operación con los discos **flexibles** compactos Amsoft CF-2 puede realizarse en la forma siguiente:

Para abrir el hueco de protección ante escritura, desliza el pequeño cierre '**palpebral**' localizado en la esquina izquierda del disco, y el hueco quedará abierto, (véase Figura 8 (a)).

Hueco de Protección ante Escritura (ABIERTO)

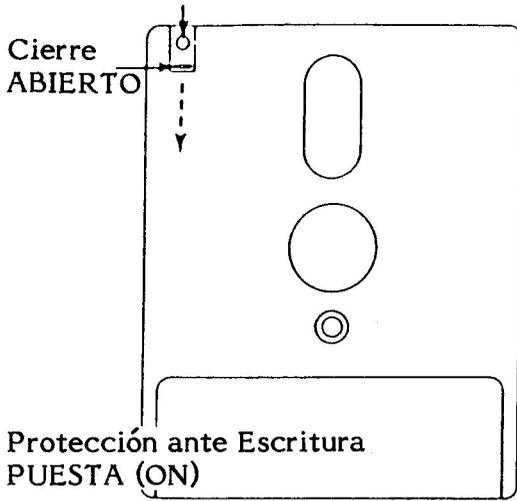


Figura 8(a)

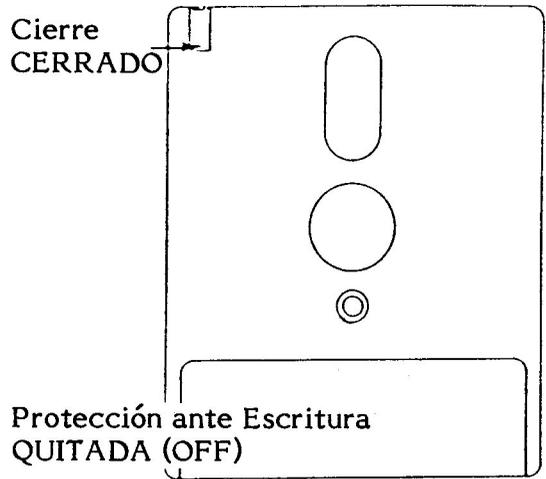


Figura 8(b)

Para cerrar el hueco de protección ante escritura, simplemente desliza el cierre 'palpebral' a su posición de obturación, véase Figura 8(b).

Algunos otros discos **flexibles** compactos emplean una pequeña palanca de material plástico situada en una ranura en la esquina izquierda, véase Figura 9.

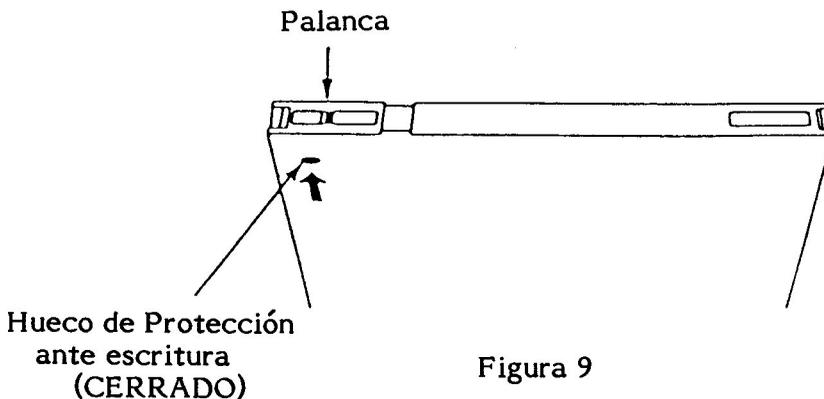


Figura 9

Para abrir el hueco de protección ante escritura en esta clase de discos, desliza la palanca hacia el punto medio del disco, usando la punta de un bolígrafo u objeto similar (véase Figura 10).

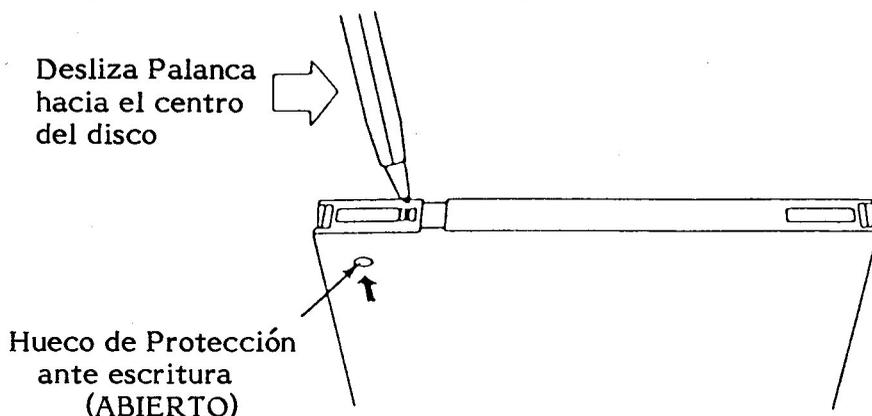


Figura 10

Observa que sin considerar el método empleado para abrir y cerrar el hueco de protección ante escritura, en todos los casos abriendo el hueco se pone la protección contra la **sobre-escritura**.

### IMPORTANTE

Asegúrate siempre que los huecos de protección ante escritura en tu disco maestro CP/M están abiertos.

### Cuando tu Disco está Inserto

En la parte frontal de la unidad, verás un piloto indicador rojo, y un pulsador para **eycción** del disco (véase Figura 11).

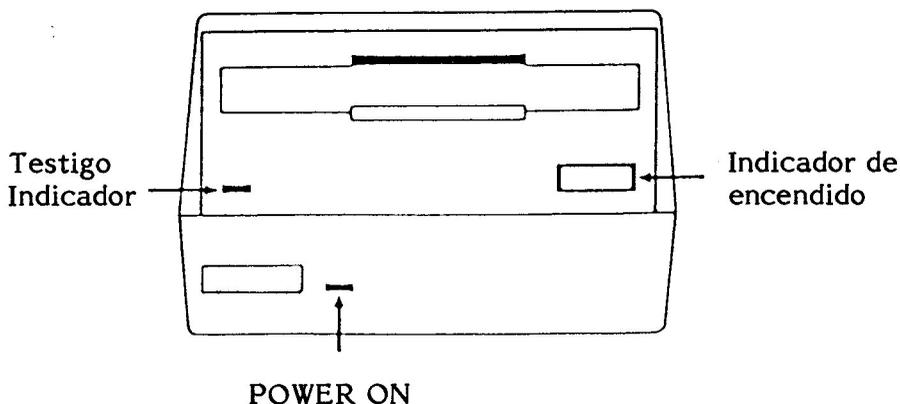


Figura 11

## Testigo indicador

El piloto indicador rojo tiene dos funciones de comprobación:

Si sólo está conectada una unidad de disco (Drive A), la lámpara iluminada indica que se está efectuando una operación de lectura o de escritura -un acceso- al disco.

Si están conectadas dos unidades de disco, el indicador rojo de la unidad B se iluminará constantemente, dándote así una indicación rápida de qué unidad es la considerada como unidad B.

Sin embargo, cuando se efectúa una lectura o una escritura del disco, el indicador vuelve a su función primaria de indicar que se está efectuando una operación de acceso al disco.

## Pulsador para Eyección

Presionando el botón de eyección te permite quitar el disco de la unidad.

### IMPORTANTE

1. SIEMPRE DEBES QUITAR EL DISCO DE LA UNIDAD ANTES DE ENCENDER O APAGAR EL SISTEMA.
2. NO PULSES EL BOTON DE EYECCION MIENTRAS SE ESTA EFECTUANDO UN ACCESO AL DISCO, PARA HACER LA LECTURA O LA ESCRITURA DE INFORMACION. SACANDO EL DISCO EN ESE MOMENTO DARA COMO RESULTADO ERRORES DE LECTURA/ESCRITURA.
3. EL DISCO FLEXIBLE MAGNETICO ESTA PROTEGIDO POR SU CARCASA EXTERNA DE PLASTICO RIGIDO. NO INTENTES NUNCA CONSEGUIR ACCESO FISICO AL DISCO, NI TOCARLO DIRECTAMENTE.

A partir de ahora, supondremos en este manual que estás familiarizado con el ordenador CPC464 y con su lenguaje BASIC, y que has leído cuidadosamente y comprendido el libro Guía del Usuario CPC464.

## Fundamentos 3:

# Carga de Programas/Juegos

Los programas/juegos disponibles en disco para el sistema Amstrad se **cargan** en memoria -desde el diskette- mucho más rápidamente que sus equivalentes suministrados en cinta cassette.

Sin embargo, es necesario recordar algunas reglas importantes que conciernen a los nombres de los programas que van a ser cargados o ejecutados.

Un comando tal como **RUN**" normalmente empleado al cargar programas en cinta, para que sean directamente ejecutados, no es válido durante una operación en disco, ya que el nombre del programa que va a ejecutarse es obligatorio cuando se trabaja con discos.

Los nombres de los programas (o **nombrefichero**) en disco constan de dos partes (o **campos**) separados por un punto (.).

El primer campo puede contener hasta 8 caracteres, y es habitualmente el **título** del programa o del juego. El segundo campo es especificable optativamente y por lo general, guarda relación con la clase de programa, e.g. BAS o BIN (en BASIC o en **binario**).

Observa que los nombrefichero no deben contener espacios en blanco o signos de puntuación, ni en el primero ni en el segundo de los campos.

Observa que está permitido el uso de MAYUSCULAS o minúsculas cuando se teclea el nombrefichero en los comandos AMSDOS y CP/M.

Posteriormente, en este manual encontrarás más información sobre cómo **denominar** a los ficheros en disco, bajo las secciones concernientes a AMSDOS y CP/M.

Por ahora, para ejecutar un programa grabado en disco tal como "Roldán de las Cavernas", tecla:

```
run "rolcave"
```

Si has insertado correctamente el disco de programas, el programa mencionado será **traído** por el ordenador a su memoria desde el disco, en unos pocos segundos, y comenzará a ejecutarlo inmediatamente.

Si no lo has hecho, estudia cualquier mensaje de error que aparezca en la pantalla, para ver cuál has cometido:

```
Drive A: disc missing  
Retry, Ignore or Cancel?
```

indicando que **falta el disco en la unidad A**; y preguntándote si **reintenta, lo ignora o cancela** la operación.

ROLCAVE . not found

indica que **no ha encontrado** el programa mencionado, por lo que debes analizar si has introducido un disco que no era, o no has tecleado exactamente el nombre correspondiente a ese programa.

(Sigue siempre las instrucciones de carga que acompañan a cada programa/-juego).

Bad command

significa que ese **comando es malo**, y puede ser porque hayas mencionado incorrectamente el programa, bien sea porque has sobrepasado el número de caracteres permitido en un campo o porque has introducido un espacio en blanco o un signo de puntuación dentro del nombre.

Type mismatch

significa que hay **discordancia de clase** porque has omitido el signo comillas ("").

Syntax error

Significa **error sintáctico** porque no has deletreado bien la palabra **RUN**.

Drive A: read fail  
Retry, Ignore or Cancel?

significa que el ordenador ha tenido un **fallo de lectura en la unidad A**, y te sugiere si **reintenta, ignora o cancela** la operación. Comprueba que has metido el disco correcto y pulsa **R** para Reintentarlo. Si continúa apareciendo el mensaje anterior cuando intentas **accesar** ese disco en concreto, es probable que el disco esté deteriorado, o que no lo hayas **formateado** correctamente usando el Amstrad CP/M.

Finalmente, si aparece el mensaje:

Press PLAY then any key:

que te dice **aprieta PLAY y luego cualquier tecla**, es porque la(s) unidad(es) de disco o la unidad de Interface no ha sido correctamente conectada al ordenador, o que la unidad de disco no está enchufada al Suministro de Red, o que no has apretado el interruptor de encendido.

— Observa que si deseas **ejecutar** un programa del disco, después de haber operado previamente con el Datacorder -la lecto-grabadora de cassette- será necesario dar el comando **disc** descrito en la siguiente sección.

#### — AVISO SOBRE LA UTILIZACION DE MEMORIA —

— El sistema DDI-1 reduce la cantidad disponible de memoria en el CPC464, en 1280 bytes. Eso puede dar como resultado que algunos programas basados en cassette dejen de funcionar correctamente mientras esté conectado al ordenador la unidad de almacenamiento en discos.

— La demostración 'Home Runner' de la cassette de Bienvenida al CPC464, es un ejemplo de este aspecto de la operación.

— Si encuentras cualquiera de esos problemas con programas basados en cassette, saca primero cualquier disco presente en la(s) unidad(es), y luego apaga el ordenador y la(s) unidad(es) de disco(s).

— Enchufa la unidad de acoplamiento del panel posterior del ordenador, y luego enciende de nuevo el ordenador. Ahora ya podrás trabajar con el cassette en la forma normal.

## Fundamentos 4:

### Presentación de AMSDOS y CP/M

Quando pones en marcha tu sistema, automáticamente el ordenador efectúa comprobaciones internas para ver los periféricos que están conectados a los zócalos del panel posterior. Si el ordenador detecta que está conectada una unidad de disco, interpretará cualquier comando dirigido hacia la lectora-grabadora de cinta, y la ejecutará como enviada hacia la unidad de disco. Por lo tanto, si tecleas cualquiera de los comandos siguientes:

```
load "filename"  
run "filename"  
save "filename"  
chain "filename"  
merge "filename"  
chain merge "filename"  
openin "filename"  
openout "filename"  
closein  
closeout  
cat  
eof  
input #9  
line input #9  
write #9  
list #9
```

en la que 'filename' equivale a 'nombrefichero', actuará sobre el disco en lugar de sobre la cinta en cassette.

Sin embargo, el comando **speed write** fija la 'rapidez de escritura' en la cinta cassette únicamente, ya que no existe tal facilidad como podía ser 'disc speed write'.

Habiendo detectado por tanto, que hay conectada una unidad de disco, el ordenador revierte a la operación regida por el sistema AMSDOS. AMSDOS son las siglas de AMstrad Disc Operating System, y te permite programar en el BASIC Amstrad normal, ampliado con comandos extra para gestión de ficheros en disco.

Estos comandos se denominan comandos **externos**, y no están disponibles en el ordenador standard cuando no está conectada la unidad de disco. El programa que regula y ejecuta estos comandos está contenido en un circuito ROM (Memoria de Únicamente Lectura) dentro de la unidad DDI.

Los comandos externos están identificados porque van precedidos del símbolo `.` (Provocarás el envío desde el teclado hacia el ordenador si pulsas simultáneamente [**SHIFT**] y la tecla marcada con la `@`).

Algunos de los comandos **externos** más comunes que emplearás son:

```
|a
|b
| tape (que puede subdividirse en |tape.in y |tape.out)
| disc (que puede subdividirse en |disc.in y |disc.out).
```

Los comandos `|a` y `|b` indican al ordenador la **unidad** hacia la que tiene que dirigir cualquier comando subsiguiente referido a disco.

Teclea por ejemplo:

```
|a
load "filename"
```

que obliga al ordenador a que **tráiga** el programa cuyo nombre es "filename" desde el disco situado en la unidad A, y lo **cargue** en su memoria.

Si no se ha introducido primeramente el comando `|a` ni el `|b`, o el ordenador ha sido **restaurado** a sus condiciones iniciales, el sistema tiene prescrito para omisiones la utilización de la unidad A, que se considera la principal.

Si sólo estás usando una unidad de disco, puede considerarse como la unidad A; y no necesitas lanzar los comandos `|a` ni `|b`. Dando el comando `|b` cuando sólo está conectada una unidad de disco, dará como resultado el mensaje en pantalla:

```
Drive B: disc missing
Retry, Ignore or Cancel
```

a lo que deberás responder C para cancelar la operación.

El comando `|tape` le dice al ordenador que efectúe todas las operaciones de carga, escritura, etc., con la **cinta** en vez de con el disco. A menos que se mande `|tape`, el ordenador tiene prescrito para omisiones que debe efectuar las operaciones con disco desde que se enciende, o se **restauran** las condiciones originales.

Para regresar a la operación con disco después de haber mandado `|tape`, simplemente teclea:

```
|disc
```

Alternativamente, puedes por ejemplo desear traer de cassette un programa, cargarlo en memoria, y guardarlo después en disco. Puedes entonces usar el comando:

```
|tape.in
```

para indicarle al ordenador que es la **cinta de entrada** (para efectuar todas las operaciones de **ingreso** de datos en el sistema con la cinta), pero continúa escribiendo los datos sobre el disco (la unidad prescrita para omisiones).

Igualmente, para traer información del disco, cargarla en memoria y luego guardarla en cassette, necesitas primeramente teclear: **|disc.in** para contrarrestar el comando anteriormente dado **|tape.in**; luego: **|tape.out** para obligar al ordenador a considerar la **cinta de salida** como el periférico hacia el que tiene que **expedir** la información que saque a partir de ese momento.

Puede verse por tanto que los comandos **|tape.in** y **|tape.out** contrarrestan respectivamente a los comandos **|disc.in** y **|disc.out**, y viceversa.

Posteriormente, en este manual encontrarás más información sobre cómo dirigir el flujo de información hasta y desde discos y cintas, bajo las secciones concernientes a AMSDOS y CP/M.

### **Depositar Información sobre el Disco**

Antes de escribir cualquier información sobre un disco nuevo en blanco (**virgen**); el propio disco debe primero **formatearse** según las reglas establecidas por el sistema. Dicho formateo puede asemejarse a la construcción de una serie de 'estantes' claramente separados e identificados -cada uno con su **referencia**- sobre el disco, antes de que podamos depositar información en dichos estantes. En otras palabras, distribuyendo el espacio según una determinada organización, identificando claramente cada parte, para que el ordenador pueda dejar y coger información de **cualquiera** de esos sitios, simplemente sabiendo la **referencia** que tiene asignada.

Realmente, el formateo divide una cara del disco en 360 áreas claramente separadas e identificadas, que denominamos **sectores** (véase Figura 12).

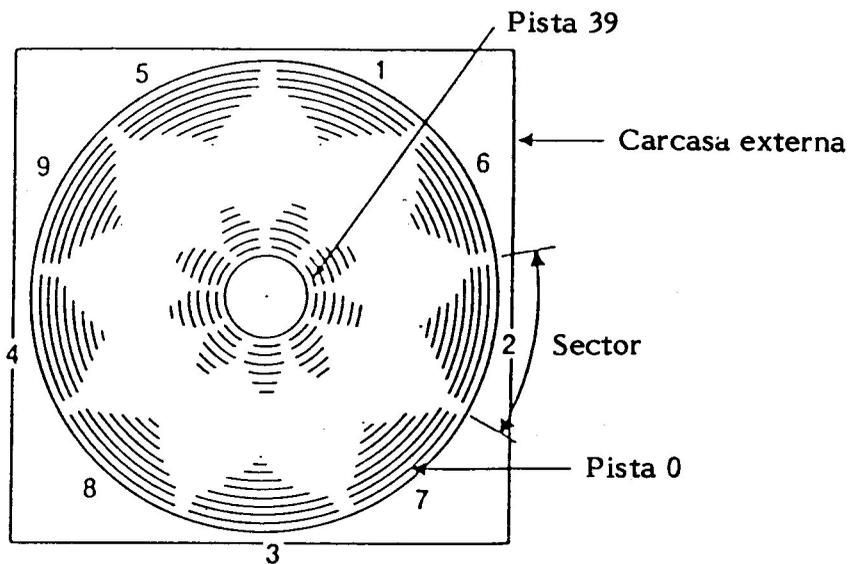


Figura 12

Hay **40 pistas** en la superficie externa de cada cara, numeradas desde la **corona** circular más externa (pista 0), hacia la más interna (pista 39), y toda la superficie circular del disco está dividida en **9 sectores**, formando así el 'entramado' de estantes.

Como cada **tramo** de cualquier pista dentro de un sector puede almacenar hasta 512 bytes de información, la capacidad total disponible de almacenamiento en **cada cara** de un disco es de 180K bytes ( $40 \cdot 9 \cdot 512 / 1024$ ).

### Primeros Pasos en el Uso del Disco CP/M

Para preparar un nuevo disco donde efectuar escritura y lectura de tus propios programas, necesitarás formatearlo usando el disco del sistema CP/M.

Enciende el sistema, y mete en la unidad A el disco CP/M que se te suministra.

(Si estás operando con 2 unidades de disco, introduce siempre el disco CP/M en la unidad A).

Teclea ahora el comando que reclama al CP/M:

```
l cpm
```

Este comando entrega el control de la operación del sistema que tenía el AMSDOS para que lo asuma el CP/M (CP/M corresponde a Control Program para Microcomputers).

Después de unos pocos segundos verás en pantalla el mensaje siguiente:

```
CP/M 2.2 - Amstrad Consumer Electronics plc.  
A>
```

Este es el mensaje de **saludo** con el que se te indica que ahora la operación del sistema está bajo control directo del **sistema operativo CP/M**.

Los dos signos A> que aparecen en la pantalla, es al mismo tiempo un aviso y una **divisa** que identifica al CP/M (similar al mensaje **Ready** que identifica al BASIC Amstrad), e indica que el ordenador está preparado para recibir tus órdenes.

CP/M usa también otras **divisas**, que explicaremos posteriormente en este manual para indicar que estás tratando con algún otro programa distinto del CP/M principal.

Una vez que estés operando bajo control de CP/M, no puedes darle comandos de BASIC, ya que el CP/M no las comprenderá.

Si por ejemplo, tecleas el comando para el BASIC:

```
cls
```

el CP/M que gobierna al ordenador rechazará el comando que le has dado, junto con un símbolo de interrogación:

```
CLS?
```

indicando que no comprende ese comando!

Para examinar brevemente algunos de los comandos que sí reconoce el CP/M, teclea:

```
dir
```

En pantalla verás el **DIR**ectorio del disco CP/M, con todos los programas que componen el sistema; y entre ellos observarás algunos con el sufijo **COM**, que corresponde a comandos directamente utilizables. Uno de estos en **format**, así que teclea:

```
format
```

En la pantalla verás el mensaje:

```
Please insert disc to be formatted into drive A  
then press any key
```

Con el que te está diciendo que **por favor, introduzcas el disco a formatear en la unidad A y luego pulses cualquier tecla.**

Quita pues el disco CP/M e inserta en la unidad un disco nuevo; luego pulsa cualquier tecla negra o azul. La operación de formateo comenzará a partir de la pista 0 y terminando en la pista 39, después de la cual el ordenador te preguntará a través de la pantalla:

Do you want to format another disc (Y/N):

Pregunta con la que te quiere decir **si quieres formatear otro disco (SI/NO)**. Si realmente lo deseas, teclea Y (porque YES=SI) y recibirás otra vez el mensaje inicial de la operación.

El proceso de formateo de discos puede repetirse cualquier número de veces, hasta que contestes a la pregunta con N (por NO=NO), en cuyo momento el sistema te pedirá que:

Please insert a CP/M system disc into drive A  
then press any key:

Con lo que te indica que **por favor, introduzcas el disco del sistema CP/M en la unidad A y luego pulses cualquier tecla.** Después de hacerlo, el ordenador volverá al modo directo CP/M ('modo consola'), y esperará a recibir un nuevo mandato. Comentaremos más adelante en este manual otros comandos CP/M, pero por ahora, habiendo aprendido a **formatear** un disco nuevo (es como 'estrenarlo') mediante el CP/M, volveremos a estar bajo el control de AMSDOS, tecleando simplemente:

amsdos

Siempre mantén una copia **maestra** de tu disco CP/M en un lugar seguro, ya que realmente es la **llave** de tu sistema. Posteriormente en este manual, te mostraremos cómo hacer una **copia de trabajo** de tu disco CP/M, de manera que puedas guardar tu copia maestra alejada en lugar seguro.

### SE CONSCIENTE

EL FORMATEO DE UN DISCO EN EL QUE PREVIAMENTE SE HA REGISTRADO INFORMACION, HACE QUE SE BORRE SU CONTENIDO.

No podrás formatear un disco que tenga su hueco de protección ante escritura abierto (ya que el formateo es una escritura seguida de lectura posterior de comprobación). Si intentas hacerlo, te aparecerá en pantalla el mensaje:

Drive A: disc is write protected  
Retry, Ignore or Cancel?

En el que se te indica que el **disco de la unidad A está protegido contra escritura**, y luego se te pide que decidas si debe **reintentar, ignorar o cancelar** la operación. Pulsa C para cancelarla, y luego sigue las instrucciones que aparezcan en pantalla.

La operación de formateo quedará entonces abandonada.

### **...Regreso al AMSDOS**

Ahora que ya tienes uno o dos discos formateados según el sistema CP/M, podemos comenzar a manipular programas en BASIC que manejen información desde y hasta el disco (y la cinta) usando el AMSDOS.

### **Guardar en el Disco el Programa que hay en Memoria**

Habiendo tecleado un programa en la memoria del ordenador, lo puedes **guardar** en el disco si tecleas:

```
save "filename"
```

También aquí es obligatorio mencionar el **nombrefichero** con que va a quedar registrado el programa en el disco, y que no debe contener más de 8 caracteres en el primero de los campos (filename), y hasta 3 caracteres en el segundo de los campos, que aquí no hemos reflejado. Si no especificamos el segundo de los campos, el AMSDOS colocará como **sufijo** la palabra de tres letras **BAS** (abreviatura de BASIC) para identificar la clase de fichero (el 'apellido').

Observa también que los nombrefichero no deben contener espacios en blanco ni signos de puntuación, ni en el primero ni en el segundo de los campos. El signo punto (.) es el que está preceptuado para separar los campos.

Posteriormente en este manual, encontrarás más información sobre la **denominación** de los ficheros en disco, bajo las secciones concernientes a AMSDOS y CP/M.

Como ejemplo de la operación de guardar un programa en disco, escribe en la memoria un breve programa, mete un disco formateado en la unidad, y luego teclea:

```
save "example"
```

Después de algunos segundos, volverá a aparecer en pantalla el aviso **Ready** y el programa habrá sido guardado en el disco. (Si no ha sido así, comprueba cualquier mensaje de error que aparezca en la pantalla para observar si bien has olvidado meter el disco en la unidad correcta, has olvidado cerrar el hueco de protección ante escritura, has deletreado mal el comando, o incluso enchufar las unidades y apretar el interruptor).

Los ficheros con textos binarios, protegidos y ASCII, también pueden guardarse en disco usando los sufijos normales que empleas con la cinta cassette ,**b** ,**p** y ,**a** respectivamente.

Los nombrefichero de los programas cuando están registrados en el disco no requieren el uso del signo de admiración (!) que les precede al trabajar con cinta, para suprimir los avisos de lectura/escritura que aparecen en la pantalla. Cualquier signo ! que preceda al nombre, será quitado del nombrefichero e ignorado por el AMSDOS.

### **Catálogo de ficheros de un disco**

Al igual que los cassettes, también puede obtenerse el **catálogo** de los ficheros que hay en el disco, pero a una velocidad muchísimo mayor. Después de guardar el programa anterior, teclea el comando de BASIC:

```
cat
```

En la pantalla verás:

```
Drive A: user 0  
EXAMPLE.BAS 1K
```

Donde te indica que es la **unidad A: usuario 0**, y luego te muestra el nombre fichero, incluyendo cualquier segundo campo, lo hayas especificado o lo haya añadido el sistema automáticamente, junto con la **longitud del fichero** (redondeada al kbyte superior más próximo).

### **Carga desde el Disco en la Memoria**

Los programas pueden cargarse desde el disco en la memoria y luego ejecutarse usando consecutivamente los comandos:

```
load "filename"  
run
```

o pueden directamente hacer que el sistema los ejecute, si usas el comando:

```
run "filename"
```

Observa que los programas protegidos únicamente pueden ejecutarse directamente.

Si hay en operación 2 unidades de disco, debes especificar a cuál de las unidades (A o B) te estás refiriendo al mandar una operación; y para ello puedes lanzar el comando |a o |b antes de los comandos **save**, **cat**, **load** o **run**.

### **Copia de Programas desde un Disco a Otro**

Usando los comandos ya aprendidos en esta sección, puedes ver que la copia de programas de disco a disco puede efectuarse simplemente trayendo el programa desde el disco **origen** a la memoria, quitando el disco origen y guardando el programa desde la memoria en el disco **destino**.

Para traspasar un programa de un disco a otro cuando dispones de 2 unidades de disco, es preferible meter el disco **origen** en, por ejemplo, la unidad B, y el disco **destino** en la unidad A. Para copiar un programa de esta forma, teclea sucesivamente:

```
|b  
load "filename"  
|a  
save "filename"
```

Para copiar todo el contenido de un disco, en otro disco, es mucho más fácil y más rápido usar las facilidades **DISCCOPY** y **COPYDISC** disponibles en el CP/M, en lugar de repetida e individualmente hacer que el BASIC **cargue** y **guarde** cada programa.

En esta sección encontrarás más adelante, una breve descripción de cómo usar **DISCCOPY** y **COPYDISC**, y una explicación más detallada se te dará bajo la sección concerniente al CP/M.

Las facilidades **DISCCOPY** y **COPYDISC** pueden usarse para producir una **copia de trabajo** del propio disco con el sistema CP/M.

### **Copia de Programas desde Cassette hasta Diskette**

Inserta el cassette con el programa en la lecto-grabadora de cinta, y luego teclea:

```
|tape.in  
load "program name" or load""
```

Para que tráiga desde la cinta y lo cargue en memoria el programa con dicho nombre, o el primero que encuentre en la cinta respectivamente.

El ordenador previamente te responderá con el mensaje:

Press PLAY then any key:

Para que **aprietes PLAY y luego cualquier tecla**. Después de que lo hayas hecho y que el programa esté cargado en memoria, y el motor del cassette detenido, teclea:

```
save "filename"
```

Para guardarlo en el disco (unidad prescrita para omisiones) con el **nombre fichero** mencionado. Observa que los programas protegidos y los de juegos/sistema no pueden ser copiados ni sobre disco ni sobre otra cinta en cassette.

### Copia de Programas desde Disco hasta Cassette

Inserta el disco con el programa en la unidad. Si previamente has cargado programas desde cassette, teclea el comando:

```
ldisc.in
```

para que considere como equipo de **ingreso de datos** la unidad de disco; y luego teclea:

```
load "filename"
```

Después de que haya traído del disco y cargado en memoria el programa mencionado, aparecerá la divisa **Ready** del BASIC, y podrás guardar el programa en cassette tecleando sucesivamente:

```
ltape.out  
save "program name" or save ""
```

Para indicarle que tome como medio por donde **expedir** la información de salida la cassette, y grabe en ella el programa con el nombre mencionado en el comando, o sin ningún nombre. El ordenador responderá con:

Press REC and PLAY then any key:

Para que **aprietes REC y PLAY y luego cualquier tecla**. Después de eso y de que el programa haya quedado grabado en la cinta, se detendrá el cassette. Con eso has traspasado un programa desde el disco hasta la cinta en cassette.

Observa que el comando **|tape** indica que la **cinta** ha de usarse tanto para 'ingresar' como 'expedir' información desde y hasta la cinta (es equivalente a lanzar sucesivamente los comandos **|tape.in** y **|tape.out**. Igualmente, **|disc** hace que se 'ingresen' y se 'expidan' los datos desde y hasta el disco.

### Copia Empleando el Disco con el sistema CP/M

Todo el contenido de un diskette puede copiarse desde uno a otro, usando el sistema CP/M. Si sólo tienes conectada una unidad de disco al ordenador, usa el programa **utilitario** o utensilio de programación **DISCCOPY**.

Si estás operando con 2 unidades de disco, encontrarás que es más fácil y más rápido usar el programa utilitario o utensilio para programación del CP/M, denominado **COPYDISC**.

### Copia Empleando el DISCCOPY del CP/M

Inserta el disco con el sistema CP/M y teclea:

```
l cpm
```

Después que la **divisa A** >aparece en pantalla, teclea:

```
disccopy
```

El ordenador te pedirá que **por favor, inserta el disco origen en la unidad A, luego pulsa cualquier tecla.**

Quita el disco con el sistema CP/M de la unidad, e inserta el disco que consideras es el **original**, del que vas a efectuar la copia. Si deseas copiar el propio disco CP/M, simplemente déjalo metido dentro de la unidad.

Cuando el disco **origen** ya está metido, y has pulsado la tecla, el ordenador indicará el mensaje:

```
Copying started  
Reading track 0 to 7
```

Con el que te indica que **comenzó a copiar**, y está **leyendo las pistas 0 a 7**. Después de lo cual, te pedirá que **por favor, inserta el disco destino en la unidad A y luego pulsa cualquier tecla.**

Para ello, habrás de quitar tu disco **origen** y meter el disco en el que quieres obtener la **copia**, que normalmente llamamos **disco destino**. Ten en cuenta que cualquier información previa que haya en tu disco de destino, quedará **sobreescrita** (se escribirá encima de ella) al efectuar la operación de copia.

Si insertas un disco que está sin formatear, o formateado según otro sistema, automáticamente también será formateado según las reglas del sistema CP/M, durante la operación de copia.

Después de haber insertado tu disco **destino** y haber pulsado una tecla, el ordenador te mostrará el mensaje:

```
Writing track 0 to 7
```

para indicarte que está **escribiendo las pistas 0 a 7**. Cuando acabe, te invitará a que metas otra vez el disco **destino** para hacer la lectura de las pistas 8 a 15, y más adelante el disco **destino** para hacer la escritura de dichas pistas. Este proceso lectura/escritura continuará, con tacadas de 8 pistas cada vez, hasta que haya concluido la última pista (la 39). En ese momento, te preguntará si **deseas copiar otro disco**, a lo que debes contestar **Y = Sí**, cuando desees otra copia, o **N = No** cuando no la desees.

Si has terminado de hacer tus copias, sigue las instrucciones que aparecerán en pantalla para que puedas **salir** del modo **DISCCOPY** en el que estabas trabajando.

### **Copia Usando el COPYDISC del CP/M**

**COPYDISC** sólo puede usarse si tienes conectadas 2 unidades de disco. Este 'utensilio para programar' te permite copiar todo el contenido de un disco en otro disco, de manera más rápida pero similar a la operación **DISCCOPY** previamente descrita. La ventaja de **COPYDISC** es que no tienes que estar continuamente metiendo y sacando los discos origen y destino.

Habiendo leído y comprendido la operación de **DISCCOPY**, la operación **COPYDISC** puedes efectuarla fácilmente en la forma siguiente:

Insertas el disco con el sistema CP/M y tecleas el comando:

```
lcpm
```

Cuando aparece la divisa típica del CP/M **A>**, simplemente teclea:

```
copydisc
```

Luego sigue las instrucciones que aparecen en pantalla, y el contenido de tu disco **origen** será transferido -copiado- hasta tu disco **destino**, con 8 pistas en cada pasada, hasta que haya concluido con la última pista (la 39). Al igual que **DISCCOPY**, también **COPYDISC** incorpora el formateado automático del disco, si no ha sido efectuado según las reglas del CP/M. **COPYDISC** puede usarse para producir una **copia de trabajo** del propio sistema CP/M.

## Comprobación de Discos

El disco con el sistema CP/M permite la facilidad de comparar un disco con otro para comprobar si la copia de información está completamente libre de errores.

Si tienes conectada únicamente una unidad de disco al ordenador, pasa el control al CP/M y luego teclea:

```
discchk
```

Sigue las instrucciones que aparecen en pantalla, y el CP/M comprobará la información en el disco **destino** con relación a la del disco **origen**. Si el ordenador detecta alguna diferencia entre ellas, mostrará el mensaje:

```
Failed to verify destination disc correctly :  
track x sector y
```

Con el que te indica que ha **fallado en verificar correctamente el disco destino** y te señala además la **pista x sector** y donde ha detectado la diferencia entre la información de ambos discos.

Si la verificación tiene éxito, el ordenador continuará con su comprobación, 8 pistas de cada vez, hasta que esté completa. Si se detectó un error, se mostrará un **WARNING** que constituye un **aviso**, antes de preguntarte si deseas comprobar otro disco.

Si estás operando con 2 unidades de disco, la comprobación de discos puede realizarse más rápida y fácilmente, usando ahora el otro utensilio de programación denominado **CHKDISC**. Opera de forma similar al **DISCCHK** previamente descrito. La ventaja de **CHKDISC** es que no tienes que estar continuamente metiendo y sacando los discos origen y destino.

Para usar **CHKDISC** elige CP/M y luego teclea:

```
chkdisc
```

y sigue las instrucciones que aparecen en pantalla.

## Abandono de una operación a medias

Observa que las funciones y operaciones llevadas a cabo empleando el disco con el sistema CP/M pueden ser **abandonadas** a medias, si das el comando de 'corte', pulsando simultáneamente **CTRL** y la tecla marcada **C**. Hacer eso te permite regresar al modo directo del CP/M, llamado normalmente **modo consola**.

Como ejemplo, pasa al control del CP/M y teclea:

disccopy

Cuando el ordenador te invita a que insertes un disco en la unidad, pulsa simultáneamente [ CTRL ] y C. La operación que estaba en curso quedará 'cortada' y será abandonada.

En este manual y bajo la sección concerniente al CP/M, hallarás más información sobre DISCCOPY, COPYDISC, DISCHK y CHKDISC, así como sobre otros 'programas utilitarios' disponibles en el sistema CP/M, tales como FORMAT.

Para concluir estos FUNDAMENTOS, con sus cuatro partes, te presentamos un breve resumen de los puntos primordiales tratados hasta ahora:

## INSTALACION

---

1. Conecta siempre el Cable de Red a una clavija de 3 patillas, siguiendo las instrucciones contenidas en la primera sección titulada 'Instalación'.
  2. Nunca conectes la(s) unidad(es) de disco a ningún suministro de energía distinto de 220-240V y 50Hz.
  3. No hay partes y componentes manipulables por el usuario dentro de las unidades. No intentes conseguir acceso al interior del equipo. Ponte en contacto con el personal cualificado de servicio de tu distribuidor.
  4. No coloques tus unidades de disco más cerca de 20 cm en la parte izquierda de tu monitor, ni tampoco cerca de ninguna fuente de interferencias eléctricas.
  5. Mantén las unidades de disco y los discos alejados de los campos magnéticos.
  6. Mantén el cable plano de interconexión alejado del cable de red.
  7. No bloques ni recubras ninguna de las ranuras de ventilación.
  8. No uses ni almacenes el equipo en áreas excesivamente calientes, frías, húmedas o polvorientas.
-

## OPERACION

---

1. Siempre quita cualquier disco de la(s) unidad(es) antes de encender o apagar el sistema.
  2. Siempre aprieta primero el interruptor de la(s) unidad(es) de disco (POWER ON) antes de apretar el interruptor del ordenador (POWER ON).
  3. No oprimas el botón de Eyección del disco, mientras que en la unidad se esté efectuando un acceso para lectura o escritura.
  4. Asegúrate que no pierdes ni sobre-escribes (escribes encima de) accidentalmente tu disco con el sistema CP/M. Se te aconseja sacar una copia de **trabajo** de dicho disco, y asegúrate que están abiertos siempre sus huecos de Protección ante Escritura.
  5. Nunca toques la propia superficie del disco flexible, dentro de su carcasa protectora rígida.
  6. Si estás usando programas basados en cinta de cassette, mientras el sistema DDI-1 está conectado al ordenador, observa el aviso sobre memoria utilizable de la página F3.2 de los capítulos de Fundamentos en este manual.
-

# CAPITULO 1

## Preparando Discos de Trabajo

Este capítulo comenta cómo preparar los discos que se usan diariamente, e introduce algunas de las facultades del CP/M y sus programas de Utilidad.

Los temas tratados son:

- \* Preparación de una copia de respaldo del disco maestro.
- \* Preparación de un disco de trabajo con el sistema/utensilio.
- \* Operación de un disco sólo para trabajar con BASIC.
- \* Instalación de una aplicación 'llave en mano' en AMSTRAD BASIC.
- \* Instalación de una aplicación 'llave en mano' en CP/M.

Los capítulos de Fundamentos han descrito cómo formatear un disco del sistema recién adquirido, y que puedes usar con BASIC y para juegos así como en CP/M; y cómo hacer copias exactas de diskette mediante el programa utilitario DISCCOPY (una unidad), o el COPYDISC (dos unidades). Este capítulo explica cómo preparar diskettes que tengan registrados los programas que desees.

### 1.1 Copias de Respaldo del Disco Maestro

Es de importancia extrema sacar una copia del disco maestro con el Sistema/Utensilio que se suministra con tu DDI-1 y mantener el original en lugar seguro -sería muy costoso reemplazarlo si está dañado. Recuerda que el disco suministrado tiene dos caras, la cara del Sistema/Utensilios y la cara del Dr. LOGO.

Cada diskette, de hecho, tiene dos caras y eres libre de usar una u otra cara para cualquier propósito.

Recuerda que si estás usando un diskette virgen sobre el que quieres copiar algo, los programas utilitarios DISCCOPY y COPYDISC efectuarán el formateado del disco al mismo tiempo que están haciendo el copiado.

## 1.2 Un disco de trabajo con el SISTEMA/UTENSILIOS

Encontrarás que preparar una copia para el trabajo diario con tu disco Maestro con el Sistema/Utensilios y el Dr. LOGO, también es muy conveniente preparar un **disco de trabajo** que contenga aquellos programas utilitarios sacados del disco Maestro con el Sistema/Utensilios, que uses más a menudo. Con eso te quedará todavía mucho espacio para incorporar tus programas. Si en algún momento requieres la ayuda de cualquier otro programa utilitario, siempre lo puedes conseguir de la copia que tienes del disco Maestro con el Sistema/Utensilios completo. Sin embargo, todos aquellos **utensilios para programar** que uses más habitualmente, los tendrás más a mano.

Primero usa el CP/M y el programa utilitario **FORMAT** para formatear un nuevo disco virgen. Luego usa otro utensilio del CP/M, el programa **FILECOPY** para copiar sobre el nuevo disco cada uno de los programas que desees. La experiencia te dirá cuáles son los programas que quieres tener más a mano. En este ejemplo hemos escogido **AMSDOS.COM**, **FILECOPY.COM**, y **DICCOPY.COM**.

Usando tu copia del disco maestro con el Sistema/Utensilios como disco del sistema CP/M, elige CP/M y la divisa **A >** aparecerá; y ante ella teclea **FILECOPY FILECOPY.COM** y sigue las instrucciones mostradas. (El disco **SOURCE** -el **origen**- es el que estaba inicialmente en la unidad, y el disco **DESTINATION** -el de **destino**- es el nuevo disco que estás preparando). Cuando hayas acabado, copia los otros dos programas que desees mediante los comandos **FILECOPY AMSDOS.COM** y **FILECOPY DISCCOPY.COM**.

Una vez que hayas preparado tu disco de trabajo con la parte del Sistema/-Utensilios que has elegido, puedes preparar también la otra cara de ese disco, o ambas caras de otros discos, simplemente usando los programas **DISCCOPY** o **COPYDISC** que ya tienes en tu disco de trabajo. Observa sin embargo, las **restricciones sobre copias** establecidas en el Acuerdo para la Licencia al Usuario Final, en el Apéndice 2 de este manual.

## 1.3 Un disco sólo para el BASIC

Las pistas 0 y 1 de un disco del sistema están reservadas para el CP/M y no las puedes utilizar. Si quisieras usar todo el espacio que hay en disco para programas o para datos en BASIC, y NUNCA vas a intentar usar CP/M ni NINGUNO de los programas utensilios del CP/M en ese disco, puedes conformarlo sin que posea las pistas del sistema CP/M. Se denomina entonces un disco BASIC o de datos.

El disco debe formatearse usando una de las opciones del programa **FORMAT**. Teclea **FORMAT D** en lugar de **FORMAT**. Para copiar programas sobre este disco, puedes usar **FILECOPY** (cogiéndolo de un disco de sistema) o **LOAD** y **SAVE** trabajando en BASIC.

En un sistema con dos unidades es posible usar el utensilio del CP/M denominado PIP. Los programas COPYDISC y DISCCOPY formatearán el disco de **destino** de acuerdo con el disco **origen**.

#### 1.4 Libros 'llave en mano' en AMSTRAD BASIC

Si adquieres un programa de aplicación escrito en el AMSTRAD BASIC para el CPC464, debiera estar preparado para funcionar inmediatamente que lo pones en marcha. (La expresión 'llave en mano', o mejor la original 'vuelta de llave' -**turnkey**- proviene de los días en que todos los ordenadores pequeños tenían un conmutador de puesta en marcha con forma de **llave rotatoria**). Todo lo que tienes que hacer es instalarlo sobre un disco de trabajo adecuado.

##### 1.4.1 Llave en mano con BASIC usando el disco suministrado

Simplemente copia el disco maestro mediante COPYDISC o DISCCOPY, preserva el disco maestro y usa la copia. Sigue las instrucciones dadas para ejecutar el programa. Si requieres cualquier programa adicional de tu disco Maestro con el Sistema/Utensilios, usa FILECOPYY para transferirlo.

##### 1.4.2 Llave en mano con BASIC sobre tu disco de Trabajo

En este caso, copia los programas suministrados sobre el disco que ya tienes. Teclea FILECOPYY \*.\* y sigue las instrucciones. En concreto, contesta N a la pregunta:

```
Ambiguous file name: Confirm individual files (Y/N)?
```

Que te indica que el nombre del fichero es ambigüo y te pregunta si quieres **confirmar los ficheros individualmente**. Luego al actuar el programa FILECOPYY te irá informando de cada nombre de fichero a medida que lo va copiando. Al acabar, ya puedes pasar el nuevo programa de aplicación a partir de tu disco de trabajo así obtenido.

#### 1.5 Discos llave en mano con CP/M

El sistema operativo CP/M te permite trabajar con una inmensa colección de programas que ya han sido escritos para computadoras personales que admiten el CP/M. La **lógica** fundamental de esos programas ya está concebida; todo lo que se requiere para usarlos en tu CPC464 es depositarlos sobre un diskette adecuado y puede que tengas que adaptarlos respecto al método particular que el CPC464 emplea para trabajar con la pantalla.

Un **conjunto de programas** en un mismo disco, diseñado para efectuar una aplicación específica es lo que se denomina un **paquete**.

Estos paquetes están normalmente diseñados para trabajar sobre una amplia gama de ordenadores diferentes, cada uno de los cuales tiene su propio tamaño de pantalla y su propia manera de mover el cursor. Algunas veces, el paquete que compres habrá sido previamente **instalado** para el CPC464, o dispone para ello de un menú con instalaciones alternativas. En esos casos, sigue simplemente las instrucciones dadas con el paquete. Si el paquete no tiene incorporado la variante del CPC464, entonces la sección 1.5.2 indica los comandos que pueden enviarse hasta la pantalla del CPC464 para producir la clase de efectos que requiere el paquete. Normalmente, la instalación o el **procedimiento de acomodación** ("Customisation") involucrará el tecleo de los códigos pertinentes cuando se te soliciten. También aquí, has de seguir las instrucciones suministradas con el paquete.

El programa que has comprado debe estar en un disco adecuado para usar en el CPC464. Cada fabricante de computadoras usa un **formato** diferente en el disco. Aunque muchos tengan el mismo tamaño de disco, no significa necesariamente que también haya compatibilidad entre uno y otro con respecto a la información contenida. Pide siempre a tu proveedor la versión para diskettes de 3" del AMSTRAD CPC464.

### 1.5.1 Preparando un disco llave en mano con el Sistema CP/M

Es sensato preservar el disco maestro original que contiene el nuevo **paquete de programas** y 'transportar' los programas a otro diskette.

Aunque las instrucciones que aquí te damos son para un sistema con una sola unidad de disco, es en general más fácil de seguirlas si tienes un sistema con unidades gemelas (simplemente ignorando la segunda unidad).

Primero **formatea** un nuevo disco de sistema. Luego copia todos los programas desde tu disco maestro del paquete, usando el programa utensilio **FILECOPY** obtenido de tu disco con el Sistema/Utensilio.

Teclea **FILECOPY \*.\*** y sigue las instrucciones en pantalla. En concreto, contesta N a la pregunta:

```
Ambiguous filename: Confirm individual files (Y/N)?
```

Que te indica que el dado en el comando es un **nombre de fichero ambigüo** y te solicita si quieres **confirmar cada fichero individualmente**. Durante su trabajo el programa **FILECOPY** te informará de cada nombre de fichero que va copiando.

Cuando hayas acabado de usar el utensilio **FILECOPY** tendrás una copia de trabajo del disco 'llave en mano'. Si requieres cualquier programa utensilio adicional, cópialo de tu disco con el Sistema/Utensilios mediante el programa **FILECOPY**.

## 1.5.2 Configuración de un Programa bajo CP/M

El CPC464 admite una amplia gama de **códigos de control** adecuados para acomodar un paquete de programación y hacerlo ejecutable bajo CP/M. La mayoría del tratamiento de datos y muchas otras aplicaciones exigen exponer mensajes en alguna parte de la pantalla y recoger información introducida mediante el teclado, y generalmente comprender los controles de movimiento del cursor.

Si tu paquete ya ha sido **acomodado** al CPC464, no necesitas preocuparte más sobre el tema.

### 1.5.2.1 Configurando la Salida de información desde el programa

El procedimiento de instalación de un programa o paquete de programas, consistirá normalmente en pasar un programa especial (a menudo denominado INSTALL=instalar) que te hará una serie de preguntas sobre los **parámetros** de la pantalla CPC464. Las respuestas deberás deducirlas de la tabla que te damos, que está extraída del manual de referencia Amstrad BASIC SOFT157:

Valor Hexadecimal	Valor Decimal	Operación
&07	7	Pitido del zumbador
&08	8	Retrocede el cursor una posición horizontal.
&09	9	Avanza el cursor una posición horizontal.
&0A	10	Baja el cursor una línea
&0B	11	Sube el cursor una línea.
&0C	12	Sitúa el cursor en la posición base y limpia la pantalla.
&0D	13	Mueve el cursor hasta el borde izquierdo de la pantalla en la línea actual.
&10	16	Suprime el carácter situado en la posición del cursor.
&11	17	Limpia desde el borde izquierdo de la ventana, hasta, e incluyendo, la posición actual del cursor.
&12	18	Limpia desde, e incluyendo, la posición actual del cursor hasta el borde derecho de la ventana.
&13	19	Limpia desde el comienzo de la ventana hasta, e incluyendo, la posición actual del cursor.
&14	20	Limpia desde, e incluyendo la posición actual del cursor hasta el final de la ventana.

&18	24	Bascula (pone/quita) la inversión de video.
&1E	30	Sitúa el cursor en la posición base (home = casa)
&1F <c> <r>	31 <c> <r>	Mueve el cursor dentro de la ventana actual hasta la posición mencionada. <c> es la columna, normalmente de 1 a 80; <r> es la fila (row = renglón) normalmente de 1 a 25.

### 1.5.2.2 Configuración de la Entrada al programa

Los programas han de ser capaces de **interrogar** el teclado. La mayoría de las teclas en el CPC464 devuelven valores standard excepto las correspondientes al movimiento del cursor. Es posible usar el utensilio **SETUP** (véase lo primordial del CP/M) para re-definir los códigos producidos en el teclado; aunque si es posible, es preferible que cada programa diferente sea configurado para aceptar los valores standards prescritos.

La columna marcada 'Valor WP' en la tabla siguiente, indica los valores típicos que pueden establecerse para el teclado, vía el utensilio **SETUP**, cuando se está en un ambiente de **tratamiento de textos** si por ejemplo, se requieren códigos de cursor tanto procedentes del tablero separado como de alguna parte del teclado principal, y el paquete de Tratamiento de Texto sólo es capaz de reconocer un código único para cada operación.

El procedimiento de instalación de un programa de aplicación, consistirá normalmente en preguntar lo relativo a los parámetros del teclado CPC464. Las respuestas deben obtenerse de la tabla siguiente, que está extraída del manual de referencia Amstrad BASIC SOFT157:

Nombre de Tecla	Valor Hex.	Valor Decimal	[Nº de tecla si se requiere el uso del utensilio SETUP]	Valor WP (Decimal)
Cursor arriba	&F0	240	0	5
Cursor derecha	&F3	243	1	4
Cursor abajo	&F1	241	2	24
Cursor izquierda	&F2	242	8	19
Clr	&10	16	16	7
Retorno carro	&0D	13	18	13
Espacio en blanco	&20	32	47	32
Escape	&FC	252	66	27
Tabulación	&09	9	68	9
Del (suprime)	&7F	127	79	127

El valor WP (Word Processing = Tratamiento de Textos) es típico.

### 1.5.3 Arranque de un Programa llave en mano CP/M

Normalmente, todo lo que se requiere es teclear el **nombre del programa** principal de dicho paquete, cuando está en pantalla la divisa **A >** del CP/M. Por ejemplo, para trabajar con un programa de salarios denominado **PAYROLL.COM** simplemente teclea **PAYROLL**.

### 1.5.4 Autoarranque de un paquete llave en mano bajo CP/M

Es posible arreglar el sistema operativo CP/M para que automáticamente ejecute un programa concreto cada vez que se entra en CP/M usando un diskette particular. Eso se efectúa mediante una de las opciones del programa utensilio **SETUP**. (Véase lo primordial del CP/M para más detalle).

# CAPITULO 2

## Lo Primordial del AMSDOS

Este capítulo describe el sistema operativo en disco AMSDOS, y comenta todos los comandos disponibles y su utilización. Detalla la forma de los nombres de ficheros y proporciona una guía de referencia para los comandos. Cada paso se ilustra mediante ejemplos.

Temas tratados en este capítulo:

- \* Introducción al AMSDOS
- \* Directorio del Disco
- \* El formato de los nombres de ficheros
- \* Cabeceras de fichero en AMSDOS
- \* Símbolos 'comodín'
- \* Un ejemplo de programa usando Comandos de AMSDOS
- \* Guía de referencia para los Comandos del AMSDOS
- \* Manipulación de ficheros
- \* Guía de referencia para los Mensajes de Error

### 2.1 Introducción

El AMSDOS amplía al AMSTRAD BASIC suministrado con tu CPC464, añadiéndole varios comandos **externos** y alterando algunos de los comandos BASIC existentes. Los comandos externos adicionales están identificados por el símbolo 'barra vertical' (|) y son automáticamente tramitados por el programa en la ROM del DDI-1, cuando dicho sistema está conectado al CPC464.

OBSERVA que cuando un comando externo exige una expresión literal, debe traspasársele empleando la dirección de la variable literal. Analiza el comando **ERA** como un ejemplo. Observa también que no está permitido incluir el símbolo barra (|) dentro de las instrucciones **REM**.

El AMSDOS permite al usuario cambiar libremente los discos, en tanto y en cuanto no está empleando ningún fichero -en cuyo caso mostraría un mensaje de error y pudiera provocarse una pérdida de información si estaba escribiendo en el fichero que quedaba abierto.

## 2.2 Directorio del Disco

Cada disco tiene dos secciones: el **directorio** y el **área de datos**. El directorio contiene una lista con todos los nombres de los ficheros registrados en el disco, y un **mapa** de dónde se puede encontrar cada fichero registrado en el disco. El AMSDOS o el CP/M pueden calcular el tamaño de un fichero concreto examinando la **reseña** que de él hay en el directorio. El cálculo de la cantidad de espacio disponible en un disco, se efectúa sumando los tamaños de los ficheros tal y como están reseñados en el directorio y viendo luego cuánto espacio queda sin usar.

Siempre que se accesa un fichero del disco, se examina primeramente su **reseña** en el directorio, y se obtiene el sitio dentro del disco donde está almacenada la información. Cuando se crea un nuevo fichero, se le reserva espacio libre, y cuando se borra un fichero dicho espacio se libera. El directorio trabaja en unidades de 1K y puede haber hasta 64 reseñas diferentes. Los ficheros grandes tendrán una reseña por cada 16K de datos, aunque normalmente este hecho queda oculto para el usuario.

## 2.3 Nombres y clases de ficheros en AMSDOS

Cuando se usa la lecto-grabadora en cinta CPC464, se permite que los nombres de los ficheros tengan hasta 16 caracteres, y normalmente no contienen ninguna información sobre la clase de fichero que es (eg. BASIC, Binario, etc.). Dicha información está contenida en un pequeño **registro** al comienzo del fichero que denominamos la **cabecera**, y que puede mostrarse mediante el comando **CAT**. Cuando se usa almacenamiento en disco es práctica habitual dar un **nombre de fichero** y registrarlo en el disco de manera que constituya una indicación de la clase de fichero que es. Este convenio de nombres NO FUERZA al ordenador a usar el fichero de ninguna manera particular, pero algunos programas sólo aceptarán un fichero que tenga la clase de nombre correcta. BASIC aceptará cualquier tipo de nombre, pero buscará con preferencia ciertas clases de ficheros si no se le especifica una concreta. (Véase sección 2.3.2).

### 2.3.1 Construcción de los Nombres de Ficheros

El **nombre de fichero** consta de dos partes separadas por un punto (.). La primera parte puede tener hasta 8 caracteres y es propiamente el **nombre**, y la segunda puede tener hasta 3 caracteres. Así por ejemplo, "**SCREEN.BIN**", "**WELCOMED.BAS**" y "**FORMAT.COM**" son todos nombres de fichero **legales**. La segunda parte del nombre se denomina la **clase de fichero**, y vulgarmente el **apellido**. Los nombres y las clases pueden estar compuestos de cualquier combinación de letras y números, pero no pueden tener espacios en blanco incluidos.

Algunos nombres convencionales para las clases de ficheros son:

- . espacio** Tipo no específico. Puede ser un fichero de datos creados mediante **OPENOUT "RESULTS"** o un programa en BASIC guardado por el AMSDOS mediante **SAVE "PROGRAM",A**.
- .BAS** Programa en BASIC guardado por el AMSDOS mediante **SAVE "PROGRAM" O SAVE "PROGRAM",P** o bien **SAVE "PROGRAM.BAS",A**.
- .BIN** Programa o zona de memoria guardada por el AMSDOS usando **SAVE "MEMORY",B**, **parámetros binarios** .
- .BAK** Vieja versión -de respaldo- de un fichero, cuando el AMSDOS o un utensilio de programación ha guardado una versión más reciente del fichero usando un nombre que ya existe. Eso permite que el usuario vuelva a consultar la versión previa si lo precisa.
- .COM** Fichero de comandos. Todos los **utensilios de programación** disponibles en el CP/M son programas que pertenecen a esta clase de fichero, directamente ejecutable por la computadora.

### 2.3.2 Registros de cabecera en el AMSDOS

El AMSDOS automáticamente **GUARDA** ficheros con un **identificador de clase** apropiado, de manera que normalmente no se necesita especificar ninguno, a no ser que desees revocar las clases prescritas como normales y descritas anteriormente.

Los ficheros con programas en BASIC AMSDOS, los ficheros con programas protegidos en BASIC y los ficheros con programas en Binario, se guardan en el disco con un registro de cabecera (similiar al grabado en los ficheros en cinta), de manera que el comando **LOAD** del AMSDOS pueda reconocerlos y adoptar la acción apropiada. (Al igual que con la cinta, el comando **LOAD** no necesita que se le diga si el fichero que va a **traer y cargar** en memoria es de la clase BASIC protegido, Binario o ASCII -lo determina él mismo a partir de la cabecera). Si el AMSDOS no puede encontrar una cabecera cuando va a cargar un fichero, supone que el fichero es un programa en ASCII, ie. simple texto seguido.

No obstante, el contenido del registro de cabecera, cuando al **cargador** del AMSDOS se le pide que traiga del disco un fichero y no se le especifica la clase de fichero, busca primero uno de la clase **. espacio** ; luego si ese no existe busca uno de la clase **.BAS** y luego uno de la clase **.BIN**. Eso permite al usuario no escribir el nombre del fichero completo, ie. no necesita especificar en la mayoría de los casos la clase de fichero.

Un fichero de datos en disco accesible con el comando **OPENOUT** y sobre el que se ha escrito a continuación, no tendrá ningún registro de cabecera y el contenido estará en ASCII, ie. simple texto seguido, depositado mediante los comandos **WRITE**, **PRINT** o **LIST** del BASIC. El comando **OPENIN** buscará los ficheros según el orden descrito anteriormente para el comando **LOAD** cuando no se menciona explícitamente la clase de fichero.

### 2.3.3 Nombres de ficheros con dos unidades de disco

En un sistema con dos unidades, los ficheros pueden estar depositados en uno u otro disco. La computadora no busca automáticamente un fichero en ambas unidades, por lo que el usuario debe especificar cuál es la unidad que ha de examinarse. Puedes emplear cualquiera de los comandos **A**, **B** o bien **DRIVE** (descripción completa en la sección 2.5.2) para elegir una u otra unidad, y luego usar un nombre de fichero normal; o alternatively puedes revocar la unidad prescrita para omisiones, especificando como prefijo del propio nombre el identificativo **A:** o bien **B:** de cada unidad. Así por ejemplo:

```
| B
SAVE "PROG.BAS"
| A
```

y también

```
| A
SAVE "B:PROG.BAS"
```

ambas tomarán el programa en memoria y lo guardarán en el disco metido en la segunda unidad, la B

### 2.3.3 Símbolos comodín

A menudo se requiere efectuar operaciones en disco (copiado, borrado, etc.) sobre unos cuantos ficheros del disco. Cuando se menciona un nombre de fichero para una operación concreta, el sistema escruta el directorio del disco en busca de un nombre que **concuerde** exactamente con el mencionado. Es posible, cuando el comando lo permite, efectuar la operación sobre un **conjunto de ficheros** donde algunos de los caracteres del nombre tienen el significado de 'no te preocupes'. Eso se le indica usando el símbolo **?** en la posición que no queremos sea tenida en cuenta. Si toda la parte del nombre del fichero o la parte de la clase de fichero no ha de tenerse en cuenta, toda la serie de símbolos comodín **?** puede sustituirse y usar un solo símbolo comodín, el asterisco (**\***). Así por ejemplo, **FRED.\*** es abreviatura de **FRED.???** y se refiere a todos los ficheros que tengan por nombre **FRED**, **independientemente** de la clase. (Es lo que erróneamente se denomina 'nombre ambigüo').

Igualmente **F\*.BAS** es una forma abreviada de referirse a **F?????????.BAS** que son todos los ficheros de la clase **BAS** cuyo nombre empieza por la letra **F**. Finalmente, la expresión **\*.\*** significa **todos los ficheros** que haya en el disco.

DIRECTORY	Match *.BAS	Match FRED?.BAS	Match F*.BA?
BERT.BAS	BERT.BAS		
FRED1.BAS	FRED1.BAS	FRED1.BAS	FRED1.BAS
FRED2.BAS	FRED2.BAS	FRED2.BAS	FRED2.BAS
FRED3.BAK			FRED3.BAK
FRED3.BAS	FRED3.BAS	FRED3.BAS	FRED3.BAS
FINISH.BAS	FINISH.BAS		FINISH.BAS

## 2.4 Ejemplos de Uso de los Comandos AMSDOS en un programa

Para que consigas una buena comprensión de los comandos del AMSDOS, te recomendamos que practiques con los ejemplos, consultando las secciones pertinentes del resto de este capítulo a medida que prosigues. **NO** trabajes con estos programas con tu disco Maestro del Sistema/Utensilio instalado en el equipo -los programas se escriben en el disco y **NUNCA** debes arriesgarte a estropear el disco **patrón**. Usa el disco de trabajo o una copia.

### 2.4.1 Conservar datos y efectuar un Vuelco de Pantalla

OBSERVA el uso de las clases de ficheros **.DAT** y **.SRN**. Estas clases de ficheros las usamos para recordarnos lo que hay en el fichero, y no porque tengan ningún significado intrínseco. El fichero **PARAM.DAT** será un fichero de datos en ASCII sin registro de cabecera, mientras que **FLAGDUMP.SRN** es un fichero en Binario AMSDOS con cabecera. Los programas se suministran dentro del disco Maestro con el Sistema/Utensilio en forma ASCII sin proteger.

El primer ejemplo (**EX1.BAS**) dibuja una bandera de la "Union Jack" y luego guarda en el disco toda la imagen en pantalla. Para ejecutarlo teclea **RUN"EX1"**. Como ya comentamos, el AMSDOS escrutará automáticamente el directorio en busca del fichero, añadiéndole automáticamente el sufijo **.BAS** como clase de ficheros. Los detalles concretos del **vuelco de pantalla**, o sea, el modo de pantalla, los colores de la 'paleta' y el nombre de fichero que contiene la información real también se guarda en un fichero de parámetros. Así ilustramos el uso de un fichero de datos para escribir los valores de las variables de un programa (**dumpfile\$**) y de datos constantes (**modo=1**), conservándolos para ser usados por otro programa.

```

10 DIM colour(2)
20 MODE 1:ORIGIN 0,0,0,640,0,400 : REM restaura pantalla
30 dumpfile$="flagdump.srn"
40 FOR i=0 TO 2
50 READ colour(i): REM apunta colores de la lista DATA
60 INK i,colour(i)
70 NEXT
80 OPENOUT "param.dat"
90 WRITE #9,dumpfile$,1: REM guarda nombre de fichero y modo
100 FOR i=0 TO 2
110 WRITE #9,colour(i): REM guarda colores
120 NEXT i
130 CLOSEOUT
140 BORDER 0
150 FOR x=-65 TO 60 STEP 2
160 MOVE x,400:DRAWR 240,-150,1
170 MOVE x,0:DRAWR 240,150
180 NEXT x
190 FOR x=575 TO 700 STEP 2
200 MOVE x,400:DRAWR -240,-150
210 MOVE x,0:DRAWR -240,150
220 NEXT x
230 FOR x=-40 TO 0 STEP 2
240 MOVE x,400:DRAWR 240,-150,2
250 NEXT x
260 FOR x=0 TO 40 STEP 2
270 MOVE x,0:DRAWR 240,150
280 NEXT x
290 FOR x=640 TO 680 STEP 2
300 MOVE x,0:DRAWR -240,150
310 NEXT x
320 FOR x=600 TO 640 STEP 2
330 MOVE x,400:DRAWR -240,-150
340 NEXT x
350 ORIGIN 0,0,260,372,0,400:CLG 1
360 ORIGIN 0,0,0,640,150,250:CLG 1
370 ORIGIN 0,0,284,348,0,400:CLG 2
380 ORIGIN 0,0,0,640,169,231:CLG 2
390 SAVE dumpfile$,b,&C000,&4000
400 DATA 2,26,6

```

El segundo ejemplo (EX2.BAS) es un programa de propósito general para mostrar un **vertido de pantalla** sobre el disco, usando un fichero de parámetros para controlar su acción. Observa cómo se imponen los valores de las variables mediante **INPUT** procedente del fichero de datos, usando la función **EOF** para permitir la variación automática en el tamaño del fichero. Es importante que el **volcado de pantalla** mostrado por este programa haya sido realizado con la pantalla en una determinada posición dentro de la memoria; en caso contrario, el resultado estaría "distorsionado". Eso se asegura en el programa **guardante** mediante la ejecución de un comando **MODE** y a partir de ahí, teniendo cuidado de no provocar que la imagen **se deslice** con respecto a la pantalla.

```
10 DIM colour(15): REM provision para 16 colores
20 OPENIN "param.dat"
30 INPUT #9, filename$, screenmode
40 i=0
50 WHILE NOT EOF
60 INPUT #9, colour(i)
70 INK i, colour(i)
80 i=i+1
90 WEND
100 CLOSEIN
110 MODE screenmode:BORDER 0
120 LOAD filename$
```

#### 2.4.2 Previendo la 'recogida de basura'

Si el programa que efectúa transferencias de datos al fichero maneja variables literales ('string'), y particularmente tablas de literales ('string arrays'), se acelerará la **apertura** de ficheros si se impide que el sistema efectúe una **recogida de basura** en ese momento. La **recogida de basura** es el nombre con el que se conoce el proceso que efectúa el BASIC para limpiar y volver a colocar el espacio de memoria usado para almacenar los literales, cuando se le requiere que reserve dinámicamente una zona intermedia de memoria -un 'buffer'-para depositar temporalmente los datos que transfiere desde o hasta un fichero.

Si un programa opera realmente con unos cuantos datos literales, sean variables simples o múltiples, puedes incorporar las líneas del programa que viene a continuación, en que se abre un fichero **postizo** para que se reserve permanentemente un **buffer** para el fichero, lo que impide la 'recogida de basura' cuando subsiguientemente se abren ficheros. Observa que esas líneas de programa deben incorporarse después de cualquier comando **SYMBOL AFTER** que haya en tu programa.

```
OPENOUT "DUMMY"
MEMORY HIMEM-1
CLOSEOUT
```

## 2.5 Guía de referencia a los Comandos del AMSDOS

Consulta el Capítulo 8 de tu Guía del Usuario CPC464 si tienes alguna duda sobre la notación empleada, o el efecto de un comando concreto cuando se usa con el cassette.

### 2.5.1 Resumen de los comandos del BASIC

Los siguientes comandos del BASIC AMSTRAD son **interceptados** para que operen sobre el sistema de diskettes en lugar de sobre la cinta en cassette de la lecto-grabadora. A parte del comando **CATálogo**, cuya acción es sustancialmente diferente, el efecto de estos comandos es muy similar a las versiones para cinta en cassette descritas en la Guía del Usuario CPC464.

#### Comandos para **lectura de ficheros**:

LOAD, RUN, CHAIN, MERGE, CHAIN MERGE,  
OPENIN, EOF, INPUT #9, LINE INPUT #9, CLOSEIN

#### Comandos para **escritura de ficheros**:

SAVE  
OPENOUT, PRINT #9, WRITE #9, LIST #9, CLOSEOUT

#### Comando para examinar el **catálogo**:

CAT

El directorio del disco está ordenado alfabéticamente según los nombres de los ficheros y se muestra en columnas junto con el tamaño de cada fichero. También se muestra la cantidad de espacio libre.

### 2.5.2 Resumen de los comandos externos del AMSDOS

Los siguientes comandos son cumplimentados por el programa residente en una ROM dentro de la interface para la unidad de disco. Están disponibles en cuanto se instala y se enciende el DDI-1.

IA

IA

COMANDO: Establece que la unidad prescrita para omisiones sea la unidad A. Es equivalente a **DRIVE** con A como parámetro.

IB

IB

COMANDO: Establece que la unidad prescrita para omisiones sea la unidad B. Es equivalente a |**DRIVE** con **B** como parámetro.

| CPM

| CPM

COMANDO: Conmuta al **entorno operativo** alternativo, trayendo y cargando el sistema operativo CP/M de un disco de sistema. El sistema operativo suministrado con el DDI-1 es el CP/M 2.2. Este comando no se cumplimentará si en la unidad A no hay un disco de sistema con CP/M.

| DIR

| DIR { , ·string expression · }

f \$ = " \* . BAS "

| DIR , @ f \$

COMANDO: Muestra el directorio del disco (en el estilo CP/M) y el espacio libre. Si se omite la <expresión literal> , se supone los símbolos comodines, \*.\* para mostrar todos los ficheros.

| DISC

| DISC

COMANDO: Equivalente a los dos comandos |**DISC.IN** y |**DISC.OUT**

| DISC.IN

| DISC.IN

COMANDO: Usa el disco como medio de **entrada** para los ficheros (ingresar datos).

| DISC.OUT

| DISC.OUT

COMANDO: Usa el disco como medio de **salida** para los ficheros (expedir datos).

| DRIVE

| DRIVE , ·string expression ·

a \$ = " A "

| DRIVE , @ a \$

COMANDO: Estipula la unidad **prescrita** para omisiones, y será la **activa** a partir de ese comando. No se cumplimentará si el AMSDOS es incapaz de efectuar la lectura del disco metido dentro de la unidad solicitada.

## I ERA

I ERA , <string expression>

```
a$="FRED.BAK"  
I ERA , @a$
```

COMANDO: Todos los ficheros cuyo nombre **concuerde** con el nombre de fichero mencionado, y no sean **únicamente de lectura** son borrados del disco. El comando permite usar los símbolos comodines.

## I REN

I REN , <string expression> , <string expression>

```
o$="OLDNAME.BAS"  
n$="NEWNAME.BAS"  
I REN , @n$ , @o$
```

COMANDO: Da al fichero de **nombre viejo "oldname"** el **nombre nuevo "newname"**. No debe existir ya en el disco un fichero con el mismo nombre que el nuevo nombre que se da.

## I TAPE

I TAPE

COMANDO: Equivalente a los dos comandos |TAPE.IN y |TAPE.OUT.

## I TAPE.IN

I TAPE . IN

COMANDO: Usa la cinta como medio de **entrada** para los ficheros (ingresar datos).

## I TAPE.OUT

I TAPE . OUT

COMANDO: Usa la cinta como medio de **salida** para los ficheros (expedir datos).

## I USER

I USER , <integer expression>

```
I USER , 3
```

COMANDO: Uso solamente de especialistas. Consulta el manual de referencia del CP/M.

## **2.6 Copia de Ficheros**

Ya hemos descrito el uso del programa utensilio **FILECOPY** del CP/M para preparar discos de trabajo y discos con paquetes de aplicaciones. Esta sección describe cómo copiar todas las clases de fichero, desde disco hasta disco, y entre disco y cinta.

### 2.6.1 Ficheros en AMSDOS con cabeceras

Es posible copiar estos ficheros bajo el entorno CP/M usando los utensilios **PIP** o **FILECOPY** (véase Lo Primordial del CP/M). Cualquier fichero creado por AMSDOS que tenga un **registro de cabecera** (véase 2.3.2) será copiable como una sola 'pieza de información' desde disco hasta disco, desde disco hasta cinta, desde cinta hasta disco, pero en general el contenido de dicho fichero no será comprensible por cualquier programa bajo CP/M.

### 2.6.2 Ficheros en ASCII

Los ficheros creados por AMSDOS sin cabeceras, están generalmente en ASCII y son a la vez copiables y comprensibles por programas bajo CP/M. En particular, será posible intercambiar ficheros con programas en ASCII, ficheros con datos en ASCII y ficheros con textos en ASCII, libremente entre programas bajo AMSDOS y bajo CP/M.

### 2.6.3 Ficheros de únicamente lectura

Es posible, usando el CP/M, estipular que un fichero concreto sea **únicamente de lectura** y/o invisible en las operaciones que muestran el directorio del disco. Tales **atributos** sólo pueden instituirse o restituirse bajo el entorno del CP/M, pero son respetados por el AMSDOS. Para mayores detalles, véase en "Lo primordial del CP/M" el programa utensilio **STAT**.

### 2.6.4 Métodos para la Copia de Ficheros

Las tablas que te presentamos ahora cubren el copiado de ficheros de todas las clases entre cinta y disco. En ellas se supone un sistema de discos con una sola unidad. No es posible copiar un programa en BASIC protegido en **absoluto**, ni copiar un fichero en binario (tal como un programa de juego escrito en código máquina) a no ser que las direcciones de carga sean conocidas. En "Lo Primordial del CP/M" se detallan más los programas utilitarios **FILECOPY**, **CLOAD** y **CSAVE**.

El copiado de ficheros de un disco a otro, con un sistema de dos unidades de disco, es normalmente más fácil usando el utensilio PIP disponible en el CP/M (véase Lo Primordial del CP/M).

COPIA DESDE:

COPIA HASTA:	AMSTRAD BASIC en cinta *	Datos ASCII en cinta *	Binario en cinta *
AMSTRAD BASIC en cinta *	<pre> ITAPE LOAD "FILE" .change tapes- SAVE "FILE" IDISC                     </pre>		
Binario en cinta *			<pre> H=HIMEM ITAPE MEMORY .s.-1 LOAD "FILE" .change tapes- SAVE "FILE" ,B,s.,l ,r  IDISC MEMORY H .note 2.                     </pre>
ASCII en cinta *	<pre> ITAPE LOAD "FILE" .change tapes- SAVE "FILE",A IDISC                     </pre>	<pre> ICPM CLOAD "FILE", TEMP .change tapes- CSAVE TEMP ,"FILE" ERA TEMP AMSDOS .note 1.                     </pre>	
AMSTRAD BASIC en disco *	<pre> ITAPE LOAD "FILE" IDISC SAVE "FILE"                     </pre>		
ASCII en disco	<pre> ITAPE LOAD "FILE" IDISC SAVE "FILE",A                     </pre>	<pre> ICPM CLOAD "FILE" AMSDOS                     </pre>	
Binario en disco *			<pre> H=HIMEM ITAPE MEMORY .s.-1 LOAD "FILE" IDISC SAVE "FILE" ,B,s.,l ,r  MEMORY H .note 2.                     </pre>

\* El fichero tiene un registro de cabecera

<nota 1> Requiere espacio libre en el disco para el fichero temporal "TEMP"

<nota 2> <s> es la dirección de comienzo del fichero, <l> es su longitud, <r> es opcional y la dirección de ejecución (run=ejecutar).

COPIA DESDE:

COPIA HASTA:

AMSTRAD BASIC  
en disco \*

Datos ASCII  
en disco

Binario AMSDOS  
en disco \*

Todos los otros  
en disco

AMSTRAD BASIC  
en cinta \*

LOAD "FILE"  
ITAPE  
SAVE "FILE"  
IDISC

Binario  
en cinta \*

H=HIMEM  
MEMORY .s.-1  
LOAD "FILE"  
ITAPE  
SAVE "FILE",  
B, .s, .l, .r.  
IDISC  
MEMORY H  
.note 2.

ASCII  
en cinta \*

LOAD "FILE"  
ITAPE  
SAVE "FILE", A  
IDISC  
ICPM  
CSAVE FILE  
AMSDOS

ICPM  
CSAVE FILE  
AMSDOS

ICPM  
CSAVE FILE  
AMSDOS  
.note 3.

AMSTRAD BASIC  
en disco \*

LOAD "FILE"  
.change discs.  
SAVE "FILE"  
-or-  
ICPM  
FILECOPY FILE  
.follow instructions.  
AMSDOS

ASCII  
en disco

LOAD "FILE"  
SAVE "FILE", A

ICPM  
FILECOPY FILE  
.follow instructions.  
AMSDOS

Binario AMSDOS  
en disco \*

ICPM  
FILECOPY FILE  
.follow instructions.  
AMSDOS

Todos los otros  
en disco

ICPM  
FILECOPY FILE  
.follow instructions.  
AMSDOS

\* El fichero tiene un registro de cabecera

<nota 2> <s> es la dirección de comienzo del fichero, <l> es su longitud, <r> es opcional y la dirección de ejecución (run=ejecutar).

<nota 3> El fichero de destino no puede usarse directamente por BASIC. Sin embargo, esta opción es útil como almacenamiento de bajo coste o medio de copias de respaldo. El fichero puede ser de nuevo copiado en disco mediante CLOAD "FILE".

## 2.7 Guía de referencia a los Mensajes de Error del AMSDOS

Cuando el AMSDOS no puede llevar a cabo un comando por alguna razón, mostrará un **mensaje de error**. Si hay un problema con los circuitos del equipo, el mensaje de error irá seguido de la pregunta **Retry, Ignore or Cancel?**

**R** por **Retry=Reintentar**, hace que se repita la operación, posiblemente después de que el usuario haya adoptado alguna acción correctora de la anomalía.

**I** de **Ignore=Ignorar**, hace que el ordenador dé por no recibido el comando y continúe como si no hubiera ocurrido el problema, lo que a menudo lleva a resultados inesperados y posiblemente inconvenientes.

**C** de **Cancel=Cancelar**, hace que la operación sea anulada, lo que a menudo comporta un mensaje de error adicional.

Para más información, véase el Capítulo 5.

### **Unknown command**

Comando desconocido. El comando no está correctamente escrito o el acoplamiento a la unidad de disco no está instalado correctamente o no está puesto el interruptor de encendido.

### **Bad command**

Comando 'malo'. El comando no puede cumplimentarse por alguna razón. Es un error sintáctico o un fallo de los circuitos.

### **<FILENAME> already exists**

<NOMBREFICHERO> ya existe. El usuario está intentando dar a un fichero un nombre que ya está dado a otro fichero del disco.

### **<FILENAME> not found**

<NOMBREFICHERO> no encontrado. No existe en el disco un fichero que posea ese nombre.

### **Drive <DRIVE>: directory full**

Unidad <UNIDAD>: directorio lleno. No hay más espacio en el directorio del disco para reseñar un nuevo fichero.

### **Drive <DRIVE>: disc full**

Unidad <UNIDAD>: disco lleno. No hay más espacio en el disco para archivar nueva información.

**Drive <DRIVE>: disc changed, closing <FILENAME>**

Unidad <UNIDAD>: disco cambiado, cerrando <NOMBREFICHERO>. El disco de esa unidad se ha cambiado con ficheros todavía abiertos en el programa.

**<FILENAME> is read only**

<NOMBREFICHERO> es únicamente de lectura. No se puede tratar ese fichero porque está estipulado como únicamente de lectura. Los ficheros se pueden estipular como únicamente de lectura o como de escritura-lectura cuando se trabaja bajo el 'ambiente' CP/M.

**Drive <DRIVE>: disc missing**

Unidad <UNIDAD>: disco falta. No hay un disco en esa unidad, o el disco no está asentado y girando apropiadamente. La acción recomendada es expulsar el disco y volver a meterlo, y luego teclear R.

**Drive <DRIVE>: disc is write protected**

Unidad <UNIDAD>: disco está protegido ante escritura. Se ha intentado acceder un disco para escribir en él, con el hueco de Protección ante Escritura abierto, para usar el disco sácalo, cierra el hueco de protección ante escritura, y teclaea R.

**Drive <DRIVE>: read fail**

Unidad <UNIDAD>: fallo en lectura. Un error de los circuitos de lectura del disco. La acción recomendada es sacar y volver a meter el disco y teclear R.

**Drive <DRIVE>: write fail**

Unidad <UNIDAD>: fallo de escritura. Error de los circuitos de escritura en el disco. La acción recomendada es sacar y volver a meter el disco y teclear R.

**Failed to load CP/M**

Fracaso en cargar CP/M. Un error de lectura al intentar traer el CP/M a memoria como consecuencia de un comando CPM, o no estás usando un disco de sistema que sea válido y contenga el CP/M.

# CAPITULO 3

## Lo Primordial del CP/M

### Operación bajo CP/M

Temas tratados en este capítulo:

- \* Implantación del CP/M
- \* Sector de configuración
- \* Modo de consola directo
- \* Programas **transeúntes**
- \* Gestión de periféricos

El CP/M es un sistema operativo basado en disco. Es un programa especial que te da acceso a toda la potencia completa de tu sistema en discos CPC464. Dado que el CP/M está disponible para tantos ordenadores diferentes, significa que hay millares de paquetes de aplicación entre los que puedas escoger y un amplio bagaje de conocimiento y experiencia del que puedas aprovecharte.

Los detalles completos del CP/M, incluyendo información de cómo escribir tus propios programas y la información para la implementación del CPC464 están contenidos en el manual SOFT159 Una Guía al CP/M.

### 3.1 Introducción

El sistema operativo CP/M **-Programa de Control para Microcomputadoras-** provee el acoplamiento del usuario al equipo de almacenamiento en disco - una manera de que te comuniques con el ordenador y manipules los ficheros y los periféricos.

El **lazo de unión** fundamental de que se dispone es lo que se denomina Modo de Consola Directo y está identificado por las dividas **A >** o **B >**. Ciertos comandos incorporados **-residentes-** están disponibles, pero la mayoría de la operación se consigue cargando y ejecutando **programas transeúntes** del sistema. Se denominan "transeúntes" porque sólo están en la memoria de la máquina (a la que son traídos desde el disco) mientras estás usándolos como **utensilios** de trabajo, y por oposición a los que están incorporados de manera permanente en la memoria, que se llaman "residentes".

Además de los mensajes de error del CP/M standard, el sistema de discos AMSTRAD también genera algunos mensajes de error referentes a los circuitos específicos del ordenador. Consulta la sección 2.7.

### 3.2 Pistas para el sistema CP/M

La mayor parte del CP/M está grabado en las dos pistas más externas del disco. El DDI-1 toma el CP/M de esas pistas y lo **carga** en la memoria del CPC464 usando un proceso de dos etapas:

Primeramente el comando del AMSDOS de clave **CPM** carga el primer sector de la pista 0 en la memoria del CPC464. En un disco de sistema, este sector ha sido preparado para ser un programa que a su vez carga el resto de las pistas del sistema en la memoria del ordenador. Se efectúan diversas comprobaciones para determinar si las pistas del sistema contienen programas válidos de CP/M y para calcular dónde cargarlos en memoria.

Parte del CP/M está permanentemente en la ROM interna del DDI-1, el resto se carga en la memoria de escritura/lectura. En la configuración standard se dejan 39.75K de memoria de escritura/lectura para los **programas transeúntes**, con 2K de ellas usadas para el Procesador de Comando de Consola, que se carga tomándolo de las pistas del sistema siempre que se vuelve a re-entrar en el Modo de Consola Directo.

### 3.3 Sector de Configuración

Durante el proceso de carga en memoria, cuando se activa por primera vez el CP/M, se cargan diversos **parámetros del sistema** trayéndolos a memoria desde un sector especial dentro de las pistas de sistema. Dichos parámetros incluyen el mensaje de **saludo**, los códigos especiales del teclado que se requiere, etc. El utensilio **SETUP** está previsto para **fixar completamente** - "acomodar"- dicho sector de configuración a tus necesidades.

### 3.4 Códigos de control de consola

Bajo la ambientación CP/M, se usan diversas operaciones con teclas especiales para controlar la secuencia de operaciones de un programa. Estas pulsaciones de tecla sustituyen la acción de la tecla [ESC]ape, usada en el AMSTRAD BASIC, aunque algunos paquetes de aplicación pueden volver a restituir a la tecla [ESC]ape algo de su primitivo poder.

[CTRL]S detiene la salida en pantalla procedente del CP/M. Pulsa cualquier tecla para que continúe.

[CTRL]C tecleadas simultáneamente al principio de una línea de comandos, hace que se regrese al Modo Directo de Consola. Muchos programas de aplicación y utensilios de programación también reconocerán esta pulsación de "corte" como una petición de abandonar la ejecución del programa.

[CTRL]P basculador de impresora. Pone/Quita la facultad de sacar toda la imagen en pantalla a través de la impresora anexa.

[CTRL]Z final de texto.

### 3.5 Inscribiendo un disco en el CP/M

A menos que se tomen acciones especiales en el programa bajo CP/M (tal y como por ejemplo hace el programa de utilidad **FILECOPY**), el CP/M no te permitirá que escribas nada en un disco, a no ser que previamente lo hayas **inscrito** en el sistema (como la **matrícula** inicial). Además, el formato del disco (SYSTEM, DATA o IBM) sólo vuelve a ser determinado por el CP/M cuando se le **inscribe** un disco. Para la unidad A esta inscripción se efectúa siempre que el CP/M regresa al Modo Directo de Consola, o cuando se tecléa **CTRL** y **C** estando en pantalla la divisa **A >** o **B >**. Para la unidad B esta inscripción de disco sólo tiene lugar la primera vez que se efectúa un acceso a la unidad **B** después de que haya sido registrado el disco contenido en la unidad **A**.

Si intentaras grabar en un disco que previamente no ha sido **inscrito**, te aparecería el mensaje de error **Bdos Err on <DRIVE> : R/O**, con el que te indica que hay un error en la unidad, porque el disco al no haber sido inscrito, está considerado automáticamente como **únicamente de lectura**. Pulsa cualquier tecla para continuar. Si el disco cambiado era también de formato diferente, entonces ocurrirá un error de lectura o escritura. Tecléa pues **CTRL** **C** para continuar, siempre que cambies un disco.

### 3.6 Comandos Directos de Consola

Hay cinco comandos directos de consola que pueden teclearse estando en pantalla la divisa **A >** o la divisa **B >**. El primero de ellos **-SAVE-** es sólo para uso de especialistas.

Los mensajes de error del CP/M tienden a ser concisos y normalmente consisten en repetir el comando que no comprende o el nombre del fichero seguido de un signo de interrogación **?**. Repite el comando corrigiendo la equivocación cometida.

#### 3.6.1 Nombres de Ficheros

La mayoría de los comandos admiten nombres de ficheros como parámetros, y donde está especificado el nombre del fichero puede contener **símbolos comodín** (véase secciones 2.3.1 y 2.3.3 de "Lo Primordial del AMSDOS"). Todos los nombres de ficheros son forzados a mayúsculas.

Los Comandos Directos de Consola y la mayoría de los utensilios de programación disponibles en el sistema no requieren que los nombres de ficheros estén **entrecomillados** (" "). Los utensilios **CLOAD** y **CSAVE** (véase sección 3.7.2.3) sí requieren sin embargo, que se coloquen las comillas antes y después del nombre de fichero que corresponde a la cinta en cassette sóloamente.

Recuerda que los nombres de ficheros pueden tener como prefijo el identificativo de la unidad **A:** o **B:** para forzar al CP/M a que use la unidad mencionada en lugar de la que esté prescrita en ese momento.

### 3.6.2 Prescribir la unidad a usar en omisiones

Si tienes dos unidades de disco es posible que cambies la selección de la unidad prescrita entre la unidad A y la unidad B, simplemente tecleando **A:** o **B:** siempre que está presente la divisa **B>** o **A>** respectivamente. Esa **divisa** te dice desde luego cuál es la unidad **prescrita** en ese momento. Añadiendo el prefijo **A:** o **B:** a un nombre de fichero hace que la unidad mencionada **predomine** -pero no altera- la unidad prescrita para omisiones.

### 3.6.3 Comando DIR

**DIR** lista el **DIR**ectorio del disco. Los nombres de fichero no están ordenados de ninguna manera particular, sino que la posición del nombre del fichero en el listado del **DIR**ectorio, indica únicamente la posición que la reseña de ese fichero tiene en el directorio. Se permiten los símbolos comodín.

<b>DIR</b>	listará todos los ficheros reseñados en el directorio.
<b>DIR B:</b>	listará todos los ficheros del disco en la Unidad B.
<b>DIR *.BAS</b>	listará todos los ficheros de la clase <b>.BAS</b> en la unidad prescrita.
<b>DIR B:*.BAS</b>	listará todos los ficheros de la clase <b>.BAS</b> , en la unidad B.
<b>DIR PIP.COM</b>	listará sólo el fichero de nombre <b>PIP.COM</b> , si es que existe en la unidad prescrita.

### 3.6.4 Comando ERA

**ERA** de **ERASE=Borrar**, eliminar, se usa para quitar ficheros del directorio. Observa que sólo se suprime la Reseña del directorio, de manera que la información todavía está en la sección de datos del disco hasta que dicho espacio vuelva a ser utilizado por otro fichero; pero a pesar de eso, la información ahí depositada no es recuperable. Se permiten símbolos comodín en los nombres de fichero. Si se especifica **.\*** como nombre de fichero, el comando **ERA** pedirá la confirmación, ya que el comando obliga a eliminar todos los ficheros. **ERA** no lista los nombres de los ficheros que va borrando. Si encuentra que cualquiera de los ficheros que va a ser borrado es de únicamente lectura (véase el utensilio **STAT**) entonces la ejecución del comando de borrado será abandonada inmediatamente.

<b>ERA PIP.COM</b>	borrará el fichero <b>PIP.COM</b> en la unidad prescrita
<b>ERA B:PIP.COM</b>	borrará el fichero <b>PIP.COM</b> en la unidad B
<b>ERA *.BAS</b>	borrará todos los ficheros de la clase <b>.BAS</b> en la unidad prescrita.

### 3.6.5 Comando REN

**REN** -de RENAME=Re-denominar- te permite cambiar el nombre de un fichero ya existente. Se especifica en el comando primeramente el nuevo nombre del fichero, seguido del signo igual, y luego el viejo nombre del fichero. Si el nuevo nombre ya existe, aparecerá un mensaje de error. No están permitidos símbolos comodines con este comando.

**REN HELLO.BAS=HALLO.BAS**      Cambia el nombre de un fichero de  
**REN B:HELLO.BAS=HALLO.BAS**      **HALLO.BAS** a **HELLO.BAS**

### 3.6.6 Comando TYPE

**TYPE** -de Type=Tipografiar- pide que el fichero mencionado en el comando sea escrito en la pantalla, mostrando "tipo a tipo" su contenido (carácter a carácter). Si el fichero no es un fichero con texto en ASCII, obviamente aparecerán efectos colaterales impredecibles y posiblemente indeseables.

**TYPE EX1.BAS**      Muestra el fichero cuyo contenido es el programa  
                         **EX1.BAS.**

## **3.7 Comandos Transeúntes**

Para efectuar una **gestión de ficheros** más complicada que la permitida en el Modo Directo de Consola, debes emplear uno de los varios **programas de utilidad** -'utensilios'- provistos en el disco de sistema. Estos se ponen en funcionamiento simplemente tecleando el nombre del programa; y seguido a veces de algunos parámetros. Probablemente ya habrás usado **FORMAT** y **DISCCOPY**.

Los comandos caen dentro de ciertas categorías como indicamos a continuación. La documentación completa de estos programas es muy extensa y está contenida en Una Guía al CP/M (SOFT159).

Los programas que cumplimentan los comandos **SYSGEN**, **BOOTGEN**, **FILECOPY**, **COPYDISC**, **DISCCOPY**, **CHKDISC**, **DISCCHK**, **FORMAT**, **SETUP**, **CSAVE**, **CLOAD** y **AMSDOS** han sido diseñados por AMSTRAD y funcionan exclusivamente en el CPC464. No tienen ninguna función en cualquier otro sistema CP/M, aunque otros fabricantes pueden llevar a cabo tareas similares utilizando otros **utensilios** (a menudo con el mismo nombre) acomodados a sus equipos.

### 3.7.1 Gestión de Periféricos

La **utilidad PIP** (Peripheral Interchange Program=Programa para el Intercambio de Documentación entre Periféricos) te permite transferir información entre la computadora central y sus equipos periféricos.

En general, la forma de este comando es **PIP <destino>=<origen>**.

El origen (source) y el destino (destination) puede ser un bien un nombre de fichero, con símbolos comodín admitidos en el fichero origen, o un **periférico logical** (que no tiene existencia concreta física, y es simplemente una 'entidad' a usar en un programa de forma abstracta). Los periféricos lógicos que pueden usarse son:

Como Origen

Como Destino

**CON:** teclado

**CON:** pantalla ("consola")

**RDR:** interface serie

**PUN:** interface serie

**LST:** impresora ("listadora")

ejemplos:

**PIP B:=A:\*.COM**

copia todos los ficheros **.COM** desde la unidad **A:** hasta la unidad **B:**

**PIP SAV.BAS=EX1.BAS**

hace una copia del **EX1.BAS** y la llama **SAV.BAS**

**PIP CON:=EX1.BAS**

envía el fichero **EX1.BAS** hacia **CON** (que concretamente suele ser la pantalla, lográndose un efecto similar a **TYPE EX1.BAS**

**PIP LST:=EX1.BAS**

envía el fichero **EX1.BAS** hacia **LST** (que suele ser en concreto la impresora)

**PIP TYPEIN.TXT=CON**

recoge lo ingresado mediante **CON:** (habitualmente el teclado) y lo **pone dentro** de un fichero llamado **TYPEIN.TXT**. Observa que esta operación se termina mediante el código de control **[CTRL] Z**, y que para conseguir un avance de línea debes teclear **[CTRL] J** después de **[ENTER]** cada vez. **[CTRL] J** es el código ASCII para el avance de línea.

Observa que **PIP** no puede usarse como **utensilio** para copiar ficheros de un disco a otro cuando se dispone de una sola unidad. Para esa tarea usa **FILECOPY**.

### 3.7.2 Copiado de ficheros y discos

#### 3.7.2.1 Copia de un solo fichero

El programa de utilidad **FILECOPY** te permite copiar ficheros de un disco a otro cuando el sistema sólo tiene una unidad. Se encarga del cambio de discos y te da instrucciones completas a través de la pantalla. Si se especifica en el nombre de fichero un símbolo comodín, entonces **FILECOPY** te pide que le confirmes que de verdad deseas copiar cada fichero, individualmente. El programa te va informando de cada nombre de fichero que vaya copiando.

**FILECOPY \*.BAS** Copia todos los ficheros de la clase **.BAS**  
**FILECOPY EX1.BAS** Copia el fichero **EX1.BAS**

### 3.7.2.2 Copiado y comprobado de todo el disco

**DISCCOPY** (para sistemas de una sola unidad) y **COPYDISC** (para sistemas de dos unidades) te permiten sacar una **copia de respaldo** de todo el disco. Te dan instrucciones completas a través de la pantalla. Si el disco **destino** no está formateado, o no es del mismo formato como el disco **origen**, estos programas de utilidad automáticamente formatearán el disco destino correctamente al efectuar la copia. Los utensilios complementarios **DISCCHK** y **CHKDISC** te permitirán **cotejar** dos discos y comprobar que tienen exactamente el mismo contenido.

### 3.7.2.3 Ficheros en cassette de cinta magnética

Se dispone de dos **utensilios** para transferir ficheros entre cintas y discos. Excepto para un uso especializado, es poco probable que transfieras provechosamente ficheros distintos a los ASCII, ie. simple texto.

**CLOAD** puede aceptar dos parámetros, el primero es el nombre del fichero (cassette) destino, encerrado entre comillas, y el segundo el nombre del fichero (diskette) destino. Si se omite el nombre de fichero destino, el fichero quedará registrado en el disco con el mismo nombre que tiene en la cinta de cassette. Si se omite el nombre de fichero destino, **CLOAD** tomará el primer fichero que encuentre en la cinta. Si el primer carácter del nombre del fichero es la admiración (!), suprimirá los mensajes normales de trabajo con el cassette. Un ejemplo de este comando transeúnte es:

```
CLOAD "MYLETTER" MYLETTER.TXT
```

**CSAVE** puede aceptar tres parámetros. El primero es el nombre de fichero (diskette) origen, y el segundo el nombre de fichero (cassette) destino, encerrado entre comillas. Si se omite el nombre de fichero destino, el fichero en cassette tendrá el mismo nombre que el fichero en disco. Si el primer carácter del nombre del fichero es la admiración (!), entonces se suprimirán los mensajes normales al trabajar con cassette. Si se especifican ambos nombres de fichero, puede usarse entonces un tercer parámetro para especificar la velocidad de grabación en la cinta: 0 para los 1000 baudios nominales, 1 para los 2000 baudios nominales. Un ejemplo de este comando transeúnte es:

```
CSAVE OUTPUT.TXT "OUTPUT TEXT" 1  
CSAVE DATAFILE
```

### 3.7.3 Gestión del Sistema

#### 3.7.3.1 STAT

STAT -de Statistical- proporciona un directorio más completo y permite llevar la gestión de los periféricos. Todas las reglas normales se aplican a los nombres de los ficheros, incluyendo el uso de símbolos comodín. Las facilidades provistas son:

Mostrar el estado del disco y el espacio libre que hay en él:

```
STAT
STAT A:
STAT B:
```

Mostrar información ampliada del directorio sobre un fichero particular:

```
STAT *.COM
STAT EX1.BAS
```

Fijar como atributo de un fichero el estado de **únicamente lectura** de manera que no pueda borrarse ni escribirse encima accidentalmente:

```
STAT *.COM $R/O
STAT EX1.BAS $R/O
```

Estipular como atributo de un fichero el estado **lectura/escritura**, cambiando la asignación de únicamente lectura:

```
STAT *.COM $R/W
STAT EX1.BAS $R/W
```

Estipular como atributo de un fichero el estado **sistema**, de manera que es invisible ese nombre en los listados de directorio y en los programas que copian ficheros. El fichero todavía estará disponible para otros propósitos:

```
STAT *.COM $SYS
STAT SECRET.BAS $SYS
```

Estipular como atributo de un fichero el estado **directorio**, invirtiendo el atributo **sistema**:

```
STAT *.COM $DIR
STAT SECRET.BAS $DIR
```

#### 3.7.3.2 SETUP

Este utensilio de programación (de SETUP="Establecer completamente", fijar) te permite re-definir las características del teclado CPC464, de la unidad de disco y del acoplamiento serie, y reclamar diversas acciones cuando se implanta, se carga por primera vez el CP/M. Cuando acaba su función, actualiza el **sector de configuración**.

El programa está gobernado por un menú, y cuando un determinado parámetro es correcto, o no requiere ninguna modificación, pasa al siguiente contestando **Y** -por Yes=SÍ- a la pregunta

Is this correct (Y/N):\_

Que te dice **¿Está correcto?**. El programa puede cortarse mediante las teclas [CTRL] **C**, simultáneamente. Cuando haya hecho todos los cambios que le solicites, te preguntará:

Do you want to update your system disc (Y/N):\_

Con el que te pregunta **si quieres actualizar tu disco de sistema** con lo que tienes la oportunidad de retener el sector de configuración existente en el disco (si tecleas **N** por No), o de cambiarlo. Y luego después:

Do you want to restart CP/M (Y/N):\_

en el que te pregunta, **si deseas volver a 'arrancar' el CP/M**, permitiéndote que cargues y ensayes -al teclear **Y** por Yes=SÍ- la nueva configuración que has estipulado.

Para copiar un sector de configuración desde un disco a otro, bien usa **BOOTGEN** (véase más adelante) o carga **SETUP** desde el disco origen, contesta **Y** a cada pregunta, inserta el disco destino antes de contestar la penúltima pregunta que te hemos descrito.

Los caracteres con código ASCII menor del valor decimal 32, pueden incluirse en los literales, tecleando la flecha ascendente ( **↑** ) seguida del carácter correspondiente tomado del repertorio **@ , A-Z, [, \ , ] , > , \_**

Las siguientes opciones son las que normalmente requieren más atención:

**\*\* Initial command buffer:** Cualquier carácter que introduzcas en este "buffer para el comando inicial" aparecerá como si lo hubieras tecleado en el Modo de Consola Directo cuando se carga por primera vez el CP/M. Tiene como efecto la **autoejecución** de un programa particular en ese momento. Recuerda incluir el equivalente de la tecla [ENTER] que está representado por los dos caracteres **↑M**. Por ejemplo, para que automáticamente se ponga en marcha el programa **STAT**, debes incluir para el **buffer del comando inicial** la serie de caracteres **STAT↑M**.

**Sign-on String:** Ese es el **literal de saludo** que aparece en la parte superior de la pantalla cuando se carga por primera vez el CP/M. Observa el uso de **↑J↑M** para conseguir el efecto de retorno de carro -avance de línea. La parte delantera del mensaje standard fija los colores adecuados de pantalla y reborde para trabajar en el modo de 80 columnas (véase el Capítulo 9 de la Guía del Usuario CPC464) y debieras copiarlo exactamente si vas a conser-varlos.

**Keyboard translations:** Eso permite **traducciones de teclado**, ya que pueden estipularse nuevos códigos ASCII a cada tecla, simulando de una forma eficaz el comando **KEY DEF** del **AMSTRAD BASIC**. Los parámetros requeridos son los **números de tecla** -o códigos de entrada- y los valores ASCII asignados a ellos. Consulta la **Guía del Usuario CPC464** para ver el mapa detallado de los números de tecla.

**Keyboard expansions:** Simula eficazmente el comando **'KEY'** del **BASIC AMSTRAD**, para **ampliación de teclado**, asignando todo un comando a la pulsación de una tecla.

### 3.7.3.3 AMSDOS

Este comando transeúnte quita el control del **CP/M** y lo entrega al **BASIC AMSTRAD** incorporado en el ordenador, con el que se dispondrá también de los comandos del **AMSDOS**.

## 3.7.4 Generación de discos

### 3.7.4.1 FORMAT

El sistema de almacenamiento en discos **AMSTRAD DDI-1** admite tres **formatos** de disco, uno de los cuales tiene dos variantes.

El formato usual es el **Sistema**, obtenido usando el comando **FORMAT standard**. Las pistas del sistema se toman del disco que contiene el programa **FORMAT.COM** y automáticamente se escriben en el disco destino.

Los otros **formatos** se obtienen simplemente añadiendo una sola letra como parámetro al comando **FORMAT**, separándola por un solo espacio en blanco:

Para formato Data:     **FORMAT D**

Para formato IBM:     **FORMAT I**

Para formato Vendor: **FORMAT V**

**\*\*AVISO\*\*** El acuerdo de licencia para tu disco **CP/M**, (que está codificado con número de serie electrónicamente) permite su uso únicamente sobre un solo sistema de cómputo. En particular, eso significa que se te prohíbe dar a cualquier otra persona una disco con **TU** copia del **CP/M** numerada serialmente sobre él. Dado que cada disco de sistema que tú preparas tiene su **CP/M** sobre él, debes por tanto ser cuidadoso y no vender, intercambiar ni en ninguna manera, repartir cualquier disco con formato sistema. En lugar de ello, debes formatear un disco en formato **Vendor**, que es idéntico al formato **Sistema**, exceptuando que las pistas de sistema están en blanco, y luego copiar los programas pertinentes sobre ese disco, usando **FILECOPY** o **PIP**. Ten cuidado en que el programa que tú copies de esta manera, no tenga a su vez derechos reservados o esté sujeto a un acuerdo de licencia.

Si recibes programas en un disco en formato **Vendor**, con el fin de usarlo convenientemente, puedes bien copiarlo a un disco de sistema usando **FILECOPY** o **PIP**, o alternativamente convertir el disco a un disco **Sistema** añadiendo sobre él tu **CP/M**. Eso se consigue con los comandos **BOOTGEN** y **SYSGEN**.

LEE CUIDADOSAMENTE EL ACUERDO DE LICENCIA  
PARA EL USUARIO FINAL EN EL APENDICE 2  
DE ESTE MANUAL

### 3.7.4.2 MOVCPM

Algunas veces se requiere construir una versión del CP/M que no sea cargada en memoria en la posición standard. Puede suceder porque se desea reservar algo de memoria para otro propósito, por lo que el CP/M por sí mismo puede desplazarse a una parte inferior de la memoria. Es posible ubicar el CP/M en cualquier posición en memoria situada sobre un límite de 256 bytes. La posición se especifica por el parámetro tamaño y está en la banda 64 a 179. Este parámetro indica el número de áreas de 256 bytes disponibles para el CP/M y sus programas transeúntes.

El resultante CP/M **reubicado** puede escribirse en el disco de sistema usando **SYSGEN** o bien guardarse usando un comando sugerido por el programa **MOVCPM**.

La sintaxis del comando es **MOVCPM <tamaño> \***

eg. **MOVCPM 178 \*** hará un CP/M 256 bytes más abajo en la memoria que la versión standard (que se crea con el máximo tamaño posible de 179).

### 3.7.4.3 SYSGEN

**SYSGEN** escribe el resultado de un comando **MOVCPM** sobre las pistas de sistema de un disco de formato Sistema o de formato Vendor ("Vendedor"). Hay tres opciones:

**SYSGEN \*** escribirá el CP/M generado por un comando **MOVCPM** inmediatamente precedente.

**SYSGEN <nombrefichero>** tomará la información del fichero especificado, que probablemente haya sido guardada después de un comando **MOVCPM**, y la escribe en las pistas de sistema. eg. **SYSGEN CPM44.COM**

**SYSGEN** sin ningún parámetro te reclamará un disco Origen y un Destino, y por lo tanto copiará las pistas de sistema desde un disco hasta el otro. Esta es la opción que se usa si estás subiéndolo de grado un disco de formato Vendor a un disco de formato System.

### 3.7.4.4 BOOTGEN

Como se ha comentado en la sección 3.2 y 3.3, hay más en las pistas del sistema que simplemente el CP/M. El programa utilitario **BOOTGEN** -de Boot="Botadura"- te reclamará discos origen y destino, copia el sector 1, pista 0 (el cargador) y el sector de configuración desde un disco hasta otro. Usa **BOOTGEN** como parte del proceso de subir de grado un disco de formato Vendor hacia un disco Sistema, o cuando quieres distribuir un sector de configuración recientemente estipulado entre diversos discos.

### 3.7.5 Programación avanzada

Los siguientes programas son para uso de especialistas y se recomienda que el usuario consulte SOFT 159-Una Guía al CP/M, u otros trabajos de referencia.

<b>ASM</b>	8080 Ensamblador
<b>DDT</b>	Ayuda para la depuración de código ensamblador 8080
<b>DUMP</b>	Utensilio para el <b>volcado</b> de ficheros en hexadecimal
<b>ED</b>	Un editor sencillo de contexto
<b>SUBMIT</b>	Modo de comando de consolas para <b>someter</b> trabajos en lotes
<b>XSUB</b>	Programa transeúnte para los trabajos en lotes.

# CAPITULO 4

## Introducción al LOGO

Esta Sección intenta presentar el tema del lenguaje LOGO, con ejemplos, y proveyendo una guía a los comandos disponibles. No intenta ser una guía de referencia o tutorial completa. Eso se suministra en UNA GUIA al LOGO (SOFT160).

Los temas cubiertos son:

- \* Concepto de LOGO
- \* Carga y Ejecución con Dr LOGO
- \* Los Gráficos de la Tortuga
- \* Escribiendo tus propios procedimientos
- \* Editando tus propios procedimientos

### 4.1 Qué es LOGO

Logo te puede ayudar a 'desarrollarte' como programador, hayas tenido o no experiencia previa en programación.

Logo es un potente **lenguaje de programación** que está rápidamente consiguiendo popularidad dado que es muy fácil su aprendizaje y su utilización.

Para crear programas en Logo, usas **procedimientos** -abreviados 'process'- como bloques constructivos del programa. El propio Dr. Logo es una colección de procedimientos, denominados **primitivos**, que tú empleas para redactar tus propios programas.

Durante la década de los 70, un equipo de científicos, informáticos y educadores bajo la dirección de Seymour Papert, desarrolló Logo incluyéndole los **gráficos con tortuga** para permitir a los niños muy pequeños programar y utilizar una computadora. Desarrollaron la tortuga de manera que los jóvenes discentes pudieran tener -como Papert decía, "un objeto con el que pensar", un instrumento que les ayudara a aprender de una manera nueva y diferente. En la forma de 'punta de flecha', la tortuga puede ser dirigida por toda la pantalla usando simples comandos.

## 4.2 Dr. LOGO

Dr LOGO es una **implementación** inteligente de Logo que ha sido especialmente acomodada al AMSTRAD CPC464, para hacer incluso más fácil el programar. Se han incluido ampliaciones al lenguaje para poder disponer de las potentes facultades sonoras del CPC464 y la **edición** de programas se ha hecho más fácil al incluir el grupo de teclas separadas para el desplazamiento del cursor.

## 4.3 Comenzando

Para operar con Dr. Logo, inserta una copia de la CARA 2 de tu disco maestro en la unidad de disco. (El capítulo de **Fundamentos** contiene instrucciones de cómo copiar este disco maestro).

Pulsa simultáneamente las teclas [CTRL],[SHIFT] y [ESC] para **restaurar** el ordenador.

Teclea |CPM y automáticamente LOGO será traído del disco y cargado en memoria.

En tu monitor aparecerá el mensaje de saludo propio de Dr. Logo:

```
                Welcome to
                Amstrad LOGO V1.1
                Copyright (c) 1983/4, Digital Research
                Pacific Grove, California

                Dr. Logo is a trademark of Digital Research
                Serial No. 6002-1232-123456

                Please Wait
```

Este saludo en el que se te da la **bienvenida** y se te pide que **por favor esperes** desaparecerá pronto y aparecerá el signo de interrogación (?) que es la **divisa** utilizada por Dr. Logo. Este signo de interrogación te dice que Dr. Logo está esperando que le **ordenes** lo que deseas, escribiéndoselo simplemente mediante el teclado.

## 4.4 Primeros pasos

Intenta teclear (usando letras minúsculas): **fd 60** (y concluye el comando pulsando [ENTER])

Y verás que aparece la **tortuga** y luego se mueve hacia **adelante** (forward) 60 unidades dejando detrás de sí una **estela** que es una línea recta trazada desde el punto de comienzo hasta el punto donde está situada ahora. La pantalla se **limpiará** y presenta un área mayor para gráficos y una pequeña área de textos que es donde la **divisa** ? aparece cerca de la parte inferior de la pantalla total.

Dr. Logo decide a menudo cambiar la disposición de la pantalla, ya sea para darte un área de textos mayor o para darte un área de gráficos mayor, según convenga.

Teclea ahora **rt 90** (y concluye el comando pulsando [ENTER] ) y la tortuga **girará** 90 grados hacia la **derecha** (right). (A partir de ahora, suponemos que siempre vas a pulsar la tecla [ENTER] después que hayas dado cada línea de comandos a la computadora).

Ahora teclea **fd 60** y se trazará otra línea de la misma longitud y formando un ángulo recto con la primera línea trazada.

Experimenta con las instrucciones sencillas **fd,bk** (que hace que **retroceda=back**), **rt** y **lt** (que hace que **gire** a la **izquierda=left** 90 grados), y vete analizando lo que sucede en la pantalla.

## 4.5 Procedimientos en Dr. LOGO

Un **procedimiento** -abreviado familiarmente 'proce'- es toda lista de instrucciones que le diga a Dr. Logo cómo llevar a cabo una **tarea** determinada. Probablemente escribirás tu primer procedimiento y lo añadirás a aquéllos que ya están incorporados dentro de Dr. Logo, que son precisamente los que denominamos **primitivos**, porque te permiten **derivar** tus propias maneras de hacer las tareas.

**fd,bk,rt** y **lt** son todos procedimientos primitivos que están incorporados, y que puedes usar en cualquier momento como **bloques constructivos** para describir tus propios procedimientos -derivados- para describir cómo ha de llevar a cabo la tarea que deseas. Otro primitivo incorporado muy útil es **cs** que **limpia** o borra la pantalla (clear screen=pantalla en 'claro'), y coloca además la tortuga en su posición inicial de base.

### 4.5.1 Escribiendo un procedimiento sencillo

Es fácil imaginar que si los movimientos **fd 60 rt 90** fueran **repetidos** 4 veces cada uno, la tortuga trazaría un cuadrado cuyos lados serían de 60 unidades. Esa tarea de dibujar el cuadrado puede conseguirse escribiendo una línea de comandos muy simple:

```
repeat 4 [fd 60 rt 90]
```

Así que borra la pantalla y luego intenta teclear dicha línea para comprobar lo que sucede.

Podemos hacer que Dr. Logo se **aprenda** esta nueva tarea simplemente dándole un **nombre de procedimiento** que le permita identificarla y distinguirla de las demás. Por ejemplo, si queremos llamarla 'square'=cuadrado, le diremos que **para** hacer un 'square' emplee ese procedimiento, tecleando:

```
to square
repeat 4 [fd 60 rt 90]
end
```

Dr. Logo ahora entenderá lo que debe hacer cada vez que se encuentre la palabra 'square', ie. dibujar un cuadrado según ese procedimiento. Podríamos haber dado a ese procedimiento cualquier nombre, pero hemos elegido 'square' para practicar con inglés y recordar la tarea que efectúa.

Dr. Logo nos permite teclear toda una serie consecutiva de comandos, de manera que mediante **square rt 45 square**, dibujaremos dos cuadrados de lado 60, y girados 45 grados uno con respecto a otro.

#### 4.6 Procedimientos con parámetros

Es posible hacer un procedimiento al que le podamos decir **cuánto** de la misma manera que ya sabemos decirle **cómo** ha de efectuar una tarea mediante un procedimiento dado.

Para hacer que **aprenda** a efectuar la tarea de dibujar un **cuadrado de cualquier lado**, y dándole el **lado="side"**, en el momento oportuno, podemos cambiar la definición del procedimiento 'square' y dejarlo en:

```
to squareanysize :side
repeat 4 [fd :side rt 90]
end
```

Este nuevo procedimiento nos presenta la idea de una **variable**, ya que el lado -denominado **:side**- en este caso puede **variar** en cada circunstancia. Habrás notado que la variable **:side** tiene su nombre precedido por **dos-puntos**, lo que indica a Dr. Logo que precisamente **:side** es el nombre de una variable y no el de un comando ni el de un procedimiento.

Cuando queramos que realmente trace un cuadrado tendremos que darle un valor a dicha variable **:side**, de acuerdo con el **lado** que queramos tenga el cuadrado. Por lo tanto, citar simplemente **squareanysize 150** hará que Dr. Logo entienda que ha de trazar un cuadrado según ese procedimiento y que ha de hacer que el lado del cuadrado sea de 150 unidades.

Intenta incluir en la misma línea de comandos dos procedimientos y observa lo que sucede. Por ejemplo, obedeciendo la instrucción:

```
cs squareanysize 100 rt 45 squareanysize 150
```

la tortuga dibujará dos cuadrados de diferente tamaño de lado y el segundo girado 45 grados respecto al primero, todo ello después de borrar lo que hubiera en la pantalla.

Observa cómo Dr. Logo nos recuerda que hemos repartido una **misma línea** de comando en más de una **línea** (renglón) de pantalla.

#### 4.7 Usando Variables para recordar valores

Dr. Logo también nos permite usar variables para recordar valores, además de usarlas para pasar valores a un procedimiento. Define primero un nuevo procedimiento que lleve a cabo la tarea de dibujar **triángulos**=triangle:

```
to triangle
repeat 3 [fd :edge rt 120]
end
```

En que el **borde** o lado del triángulo es variable. Podemos comprobarlo tecleando:

```
make "edge 100
triangle
```

Con lo que le decimos que **haga** que el borde sea 100 y luego lleve a cabo la tarea de dibujar un **triángulo**. Si ahora queremos saber el valor 'recordado' por **:edge** podemos simplemente teclear **:edge** después de la divisa **\*** y Dr. Logo nos **expondrá** el valor de esa variable.

Finalmente, podemos usar nuestra variable **:edge** en cualquier otro procedimiento, como por ejemplo para dibujar según un determinado **patrón**=pattern. Observa cómo se incrementa el valor de **:edge** añadiendo a su valor previo una cantidad, de manera que cada vez que dibujamos el patrón dado lo hacemos en un tamaño mayor.

```
to pattern
triangle lt 60 triangle rt 60
make "edge :edge+4
pattern
end
make "edge 10
cs pattern
```

Cuando hayas visto bastante, pulsa [ESC] para detener el programa.

## 4.8 Edición de programas y procedimientos

Dr. Logo nos permite corregir las equivocaciones cometidas al teclear y alterar los procedimientos que hayamos definido. Las **teclas de edición** que podemos usar son:

Las teclas de cursor [↑][↓][←][→] que mueven el cursor una posición horizontal o vertical cada vez que las pulsemos.

Las teclas de cursor [↑][↓][←][→] pulsadas al mismo tiempo que mantenemos pulsada la tecla [CTRL], harán que el cursor se mueva hacia arriba o hacia abajo una **página**, y hacia la izquierda y a la derecha una **línea** cada vez.

[CLR]hará que se suprima el carácter situado exactamente en la posición del cursor; [DEL] suprime el carácter situado una posición a la izquierda de la del cursor.

[ENTER]le dice a Dr. Logo que has **concluido** la revisión de una línea de comandos, o que ya puede poner **adentro** la línea corregida cuando estás describiendo un procedimiento.

[ESC] significa que abandone la operación y [COPY] le dice cuándo has terminado de editar un procedimiento.

Para teclear comandos o nuevos procedimientos, simplemente edita el texto en la pantalla. Cualquier carácter que no sean los anteriormente mencionados, quedará **inserto** dentro del texto y en la posición señalada por el cursor en ese momento.

Para **editar** un procedimiento ya existente, usa el comando **ed** ('edit'). Dr. Logo mostrará en pantalla la vieja versión del procedimiento, y tú puedes usar todos los comandos mencionados para mover el cursor de un sitio a otro y cambiar lo que está escrito en pantalla.

Ensayá editando el procedimiento para dibujar **patrones**, tecleando simplemente **ed "patterns"**

Experimenta con las teclas de edición. Si cuando has acabado, pulsas [ESC], el Dr. Logo entenderá que quieres **abandonar** la edición hecha, y dejará la versión original que poseías sin llevar a cabo ningún cambio de los que hayas hecho.

Teclea otra vez **ed "patterns"** y después de cambiar el número 4 por un 8, pulsa [COPY]para **salir** del editor -habiendo llevado a cabo la modificación que has hecho en pantalla. Luego vuelve a pedir que ejecute el procedimiento y observa cómo ha cambiado lo que saca en pantalla. Recuerda que debes fijar el valor inicial de la variable **:edge**.

## 4.9 Sugerencias operativas

El **espacio de trabajo** que Dr. LOGO emplea, va siendo dinámicamente dividido en 'cotos' o parcelas, que aquí se denomina **nudos**, porque originalmente se usó la palabra inglesa "nodes". Puedes ver cuántos quedan libres tecleando **nodes**. En ocasiones, cuando Dr. LOGO ha empleado casi todos los nudos permitidos, 'aseará' y arreglará el espacio de trabajo y puedes observar cómo la tortuga hace una pausa mientras se lleva a cabo este trabajo doméstico. Tú también puedes pedir a Dr. LOGO que arregle y ordene el espacio de trabajo si le das el comando **recycle** (para que pase otra vez por dicho **ciclo**). Eso a menudo te permitirá continuar después de que Dr. LOGO se haya quejado de que no tiene libre ningún nudo más.

Asegúrate que hay suficiente espacio disponible en el diskette antes de comenzar con Dr. LOGO, para el caso en que decidas **guardar** tus procedimientos en el disco. Puedes usar el comando **CATálogo** en AMSDOS (véase Lo Primordial del AMSDOS).

Hojea las páginas siguientes y ensaya alguno de los ejemplos -aunque al ojear los ejemplos, no los comprendas completamente a primera vista. A medida que conozcas a Dr. LOGO serás capaz de usar más y más de los comandos permitidos.

Cuando hayas terminado la sesión con Dr. LOGO dile **adiós** mediante el comando **bye**.

## 4.10 Resumen de primitivos para Dr. Logo

Esta sección agrupa por clases de procedimientos y los enuncia en orden alfabético, mostrando además las **entradas** que se usan, a menudo con un ejemplo. Los nombres empleados para los procedimientos **primitivos** pueden darse tanto en letras mayúsculas como en minúsculas.

### 4.10.1 Tratamiento de Palabras y Listas

(Observa que en los siguientes ejemplos aparecen las **divisas** ? y > según se trate directamente con Dr. LOGO o con el editor de procedimientos, respectivamente.

**ascii**

La salida es el valor ASCII del primer carácter de la palabra de entrada.

```
?ascii "G
71
?ascii "g
103
```

## **bf**

(but first=sin primero) Saca todos los elementos del objeto de entrada exceptuando el primero.

```
?bf "smiles  
miles  
?bf [1 2 3]  
[2 3]
```

## **bl**

(but last=sin último) Saca todos los elementos del objeto de entrada exceptuando el último.

```
?bl "smiles  
smile  
?bl [1 2 3 4]  
[1 2 3]
```

## **char**

La salida es el carácter cuyo valor ASCII es el número de entrada.

```
?char 83  
S
```

## **count**

(contar, cuenta) Saca el número de elementos que hay en el objeto de entrada.

```
?count "six  
3  
?count [0 1 2 3]  
4
```

## **empty**

(empty property=propiedad de vacío) Saca TRUE (=cierto) si el objeto de entrada es la palabra vacía o la lista vacía (sin nada, nula); en caso contrario saca FALSE (=falso).

```
?empty "   
TRUE  
?empty []   
TRUE  
?empty [x]   
FALSE  
?make "x []   
?empty :x   
TRUE
```

## first

(primero) Saca el primer elemento -el anterior a todos- del objeto de entrada.

```
?first "zebra
z
?first [1 2 3]
1
```

## fput

(firstput=ponga al principio) Saca un nuevo objeto formado colocando el primer objeto de entrada como primer elemento del segundo objeto de entrada.

```
?fput "s "milesmiles
smilesmiles
?fput 1 [2 3]
[1 2 3]
```

## item

(item, elemento de un conjunto) Saca el elemento especificado del objeto de entrada.

```
?item 4 "dwarf
r
```

## list

Saca una **lista** construida con los objetos de entrada, reteniendo los corchetes externos de la lista (compárala con **se**).

```
?(list 1 2 3 4)
[1 2 3 4]
?list "big [feet]
[big [feet]]
?(list)
[]
```

## se

(sentence=oración gramatical) Saca una lista construida con los objetos de entrada, quitando los corchetes externos de la lista (compárala con **list**).

```
?make "instr_list rl
repeat 4 [fd 50 rt 90]
?run (se "cs :instr_list "ht
```

Observa que el símbolo de subrayado entre **instr** y **list** se obtiene pulsando simultáneamente **SHIFT 0**, y sirve para tratarla como una sola palabra.

## **word**

Saca una **palabra** formada por las palabras de entrada.

```
?word "sun "shine  
sunshine
```

## **wordp**

(propiedad de palabra) Saca TRUE (=cierto) si el objeto de entrada es una palabra o un número.

```
?wordp "hello  
TRUE  
?wordp []  
FALSE
```

### 4.10.2 Operaciones Aritméticas

#### **cos**

Saca el **coseno** del número de entrada, tomado en grados.

```
?cos 60  
0.5
```

#### **int**

(integer=entero) Saca la parte entera del número de entrada.

```
?int 4 / 3  
1
```

#### **random**

(azar, aleatorio) Saca aleatoriamente un entero no-negativo menor que el número de entrada.

```
?random 20
```

#### **sin**

Saca el **seno** del número de entrada, tomado en grados.

```
?sin 30  
0.5
```

+

Saca la suma de los números de entrada.

?+ 2 2

4

?2+2

4

-

Saca la diferencia de los dos números de entrada.

?- 10 5

5

?10-5

5

\*

Saca el producto de los números de entrada.

?\* 4 6

24

?4\*6

24

/

Saca el cociente decimal de los dos números de entrada.

?/ 25 5

5

?25/5

5

### 4.10.3 Operaciones Lógicas

and

(Y Lógico Inclusivo) Saca TRUE (=cierto) si el resultado de **todas** las expresiones de entrada es cierto.

?and(3<4)(7>4)

TRUE

## **not**

(NO Lógico) Saca TRUE (=cierto) si la expresión de entrada es FALSE (=falso). Saca FALSE si la expresión de entrada es TRUE.

```
?not (3 = 4)
TRUE
?not (3 = 3)
FALSE
```

## **or**

(O Lógico Inclusivo) Saca FALSE (=falso) si **todas** las expresiones de entrada son **falsas**.

```
?or "TRUE "FALSE
TRUE
?or (3=4) (1=1)
TRUE
```

=

Saca TRUE (=cierto) si los dos objetos de entrada son idénticos; en otro caso saca FALSE (=falso).

```
?= "LOGO "LOGO
TRUE
?1=2
FALSE
```

>

Saca TRUE (=cierto) si la primera palabra de entrada es mayor que la segunda; en otro caso saca FALSE (=falso).

```
?> 19 20
FALSE
?20>19
TRUE
```

<

Saca TRUE (=cierto) si la primera palabra de entrada es menor que la segunda; en otro caso saca FALSE (=falso).

```
?< 27 13
FALSE
?13<27
TRUE
```

#### 4.10.4 Variables

##### **local**

(local, privada) Hace que la(s) variable(s) mencionada(s) como entrada(s) sea accesible sólo para el procedimiento **actual** y para los procedimientos que dentro de él se citan.

```
>(local "x "y "z)
```

##### **make**

(haga) Asigna a la variable mencionada como entrada el valor del objeto de entrada.

```
?make "side 50  
?:side  
50
```

#### 4.10.5 Procedimientos

##### **end**

(fin) Indica dónde concluye la definición de un procedimiento; debe ser el único comando y al comienzo de la última línea del procedimiento. Con eso se deja de trabajar con el editor de procedimientos (cuya divisa es el símbolo `␣`).

```
?to square  
>repeat 4[fd 50 rt 90]  
>end  
square defined  
?square
```

##### **po**

(print out=exponga fuera) Muestra la definición de los procedimientos o variables especificadas.

```
?po "square  
to square  
repeat 4[fd 50 rt 90]  
end  
?po "x  
x is 3
```

##### **pots**

(print out titles=exponga fuera títulos) Muestra los nombres y títulos de todos los procedimientos que haya en el espacio de trabajo.

```
?pots
```

**to**

(para, hacia) Indica el comienzo de la definición de un procedimiento, y pasa al modo edición.

```
?to square  
>repeat 4[fd 50 rt 90]  
>end  
square defined
```

#### 4.10.6 Edición

**ed**

(edit=editar, revisar) Carga en la zona reservada del editor de pantalla (en el 'buffer') los procedimientos y/o las variables especificadas en el comando.

```
?ed "square
```

#### 4.10.7 Pantalla de Texto

**ct**

(clear text=limpia texto) Borra todo el texto en la ventana que en ese momento contiene al cursor, que después coloca en la esquina superior izquierda de la ventana.

```
?ct
```

**pr**

(print=exponer) Muestra los objetos de entrada en la pantalla de texto, quitando los corchetes externos de la lista; y detrás del último objeto de entrada envía un retorno de carro (párala con **show** y **type**).

```
?pr [a b c]  
a b c
```

**setsplit**

(fija división) Establece el número de líneas a emplear en la pantalla dividida entre texto y gráficos.

```
?ss 10
```

## **show**

(mostrar) Presenta en la pantalla de texto el objeto de entrada, reteniendo los corchetes externos de la lista; y termina la presentación con un retorno de carro (compárala con **pr** y **type**).

```
?show [a b c]
[a b c]
```

## **ts**

(text screen=pantalla de texto) Elige la pantalla completa para texto.

```
?ts
```

## **type**

(tipografiar) Muestra los objetos de entrada en la pantalla de texto, quita los corchetes externos de las listas, y no envía detrás de la última entrada el retorno de carro (compárala con **pr** y **show**).

```
?type [a b c]
a b c
```

### 4.10.8 Pantalla de Gráficos

Observa que la pantalla está en el Modo 1, dando cuatro colores, y que se usa exactamente el mismo sistema de coordenadas que en el AMSTRAD BASIC. En otras palabras, todas las posiciones de pantalla serán redondeadas al número par más próximo a una **mota** de pantalla. Los colores rojo, verde y azul pueden tener cantidades variables de 0, 1 ó 2.

## **clean**

(limpiar, asear) Borra la pantalla de gráficos sin afectar a la tortuga.

```
?fd 50
?clean
```

## **cs**

(clear screen=aclara pantalla) Borra la pantalla de gráficos y sitúa la tortuga en (0,0), con rumbo 0 grados (Norte) y con la pluma bajada (para dejar estela).

```
?rt 90 fd 50
?cs
```

## dot

(punto, mota) Pinta una diminuta 'manchita' de color en la posición especificada por la lista de coordenadas de entrada, y con el color actual de la pluma.

```
?dot[50 10]
```

## fence

(valla, cercado) Establece un 'lindero' o frontera que limita la tortuga a la pantalla visible de gráficos, o saca mensaje de **fuera de límites** en caso contrario. El comando **window** quita esos límites.

```
?fence
```

```
?fd 300
```

```
Turtle out of bounds
```

## fs

(full screen=pantalla completa) Elige la pantalla completa para gráficos.

```
?fs
```

## pal

(palette=paleta) Sacar números que representan la cantidad de rojo, verde y azul que hay en el color de la pluma.

```
?pal 2
```

```
[0 2 2]
```

## setpal

(set palette=fijar paleta) Estipula la mezcla de colores que va a usarse en la pluma. Asigna una cierta cantidad de rojo, verde y azul.

```
?setpal 3 [1 1 2]
```

```
?pal 3
```

```
[1 1 2]
```

## sf

(screen facts='hechos' de pantalla) Sacar la información estipulada para la pantalla de gráficos. El formato del comando es [**<color fondo>** **<estado pantalla>** **<tamaño división>** **<estado ventana>** **<gibosidad>**] siendo **<color fondo>** el número de la pluma para el color de fondo que será siempre 0; **<estado pantalla>** indica **ss** (pantalla dividida), **fs** (pantalla completa) o **ts** (pantalla texto); **<tamaño división>** es el número de líneas de texto estipulado para la ventana de texto cuando la pantalla está dividida; **<estado ventana>** indica el modo ventana (**window**), el modo volteo (**wrap**) o el modo cercado (**fence**).

Y finalmente gibosidad indica la **curvatura** y está permanentemente fijo a 1, que es circular.

```
?sf  
[0 SS 5 FENCE 1]
```

**ss**

(split screen=pantalla dividida) Muestra una ventana de texto sobre la pantalla de gráficos.

```
?ss
```

**window**

(ventana, lienzo) Permite que la tortuga dibuje por fuera de la pantalla visible de gráficos, después de un comando de volteo (wrap) o de uno de cercado (fence).

```
?fence fd 300  
Turtle out of bounds           Tortuga fuera de lindes  
?window  
?fd 300
```

**wrap**

(envolver, voltear) Hace que la tortuga reaparezca por el lado opuesto de la pantalla de gráficos cuando sobrepasa los límites establecidos para la ventana.

```
?cs wrap  
?rt 5 fd 1000  
?cs window  
?rt 5 fd 1000
```

#### 4.10.9 Gráficos con la Tortuga

**bk**

(backwards=retroceda) Mueve la tortuga el número de pasos especificado como entrada, en la dirección opuesta a la que está apuntando (al rumbo).

```
?cs fd 150  
?bk 50
```

**fd**

(forward=avance) Mueve la tortuga el número de pasos dado como entrada, en la dirección en la que está apuntando (siguiendo el rumbo).

```
?fd 80
```

## ht

(hide turtle=oculte tortuga) Hace que la tortuga sea invisible, con lo que se acelera y se hacen más claros los dibujos.

```
?ht  
?cs fd 50  
?st
```

## lt

(left=izquierda) Gira según el número de grados dado como entrada, hacia la izquierda.

```
?lt 90
```

## pd

(pen down=pluma abajo) Hace que la tortuga 'baje la cola' que le sirve para pintar según se mueve.

```
?fd 20 pu fd 20  
?pd  
?fd 20
```

## pe

(pen erase=pluma de borrar) Cambia el color de la pluma de la tortuga al número 0, el color del fondo; la tortuga borra las líneas trazadas.

```
?fd 50  
?pe  
?bk 25  
?fd 50  
?pd fd 25
```

## pu

(pen up=pluma arriba) Hace que la tortuga suba la pluma, con lo que deja de dibujar.

```
?fd 30  
?pu  
?fd 30  
?pd fd 30
```

### **px**

(pen reverse=pluma inversa) Hace que la tortuga cambie el color de cualquier elemento de imagen previamente coloreado, cuando lo cruza en su movimiento, dejándolo en el color inverso o complementario.

```
?fd 20 pu fd 20
?pd setpc 3 fd 20
?px
?bk 80
?fd 80
?pd bk 100
```

### **rt**

(right=derecha) Hace que la tortuga gire el número de grados dado como entrada, hacia la derecha.

```
?rt 90
```

### **seth**

(set heading=ponga rumbo) Hace que la tortuga se oriente hacia un rumbo absoluto, especificado por el número de grados dado como entrada; los números positivos hacen que gire en el sentido de las agujas del reloj; los números negativos hacen que gire en el sentido contrario a las agujas del reloj.

```
?seth 90
```

### **setpc**

(set pen colour=ponga color de pluma) Estipula que la pluma llevada por la tortuga sea la especificada por el número de entrada. El número 0 es el color de fondo.

```
? setpc 1
```

### **setpos**

(set position=fija posición) Mueve la tortuga hasta la posición especificada por la lista de coordenadas de entrada.

```
?setpos[30 20]
```

### **st**

(show turtle=muestre tortuga) Hace que la tortuga sea visible, si estaba oculta.

```
?ht
?fd 50
?st
```

**tf**

(turtle facts='hechos' de tortuga) Saca información sobre la tortuga: el formato es [`<xcor><ycor><rumbo><estado pluma><color pluma/#n><visibilidad>`] siendo `<xcor>` la coordenada x de la posición. `<ycor>` la coordenada y de la posición. `<rumbo>` indica la dirección según la brújula, hacia la que está orientada la tortuga. `<estado pluma>` indica pluma bajada (PD), pluma de borrar (PE), pluma invertida (PX) o pluma alzada (PU). `<color pluma/#n>` indica el número de pluma actual, y finalmente `<visibilidad>` indica TRUE si la tortuga es visible y FALSE en caso contrario.

```
setpos[15 30]
?rt 60
?setpc 3
?pe
?ht
?tf
[15 30 60 PE 3 FALSE]
```

#### 4.10.10 Gestión del espacio de trabajo

**er**

(erase=borrar) Elimina del espacio de trabajo los procedimientos especificados en el comando.

```
?er "square"
```

**ern**

(erase name=borre nombre) Elimina del espacio de trabajo las variables especificadas.

```
?make "side[100]
?make "angle[45]
?:side :angle
[100]
[45]
?ern[side angle]
?:side
side has no value      side no tiene ningún valor
```

**nodes**

Saca el número de  **nudos**  libres que quedan en el espacio de trabajo.

```
?nodes
```

## **recycle**

(reciclar) Deja libres tantos nudos como sean posibles y reorganiza el espacio de trabajo.

```
?recycle  
?nodes
```

### 4.9.11 Listas de propiedades

#### **glist**

(get list=coge lista) Saca una lista de todos los objetos del espacio de trabajo que tengan el nombre de propiedad dado como entrada dentro de sus listas de propiedades.

```
?glist ".DEF
```

#### **gprop**

(get property=coge propiedad) Saca el valor de propiedad que tiene del nombre de propiedad dado como entrada del objeto mencionado como entrada.

```
?make "height "72"  
?gprop "height ".APV  
72"
```

#### **plist**

(property list=lista de propiedades) Saca la lista de propiedades del objeto mencionado como entrada.

```
?plist "height  
[.APV 72"]
```

#### **pprop**

(put property=deje como propiedad) Pone la pareja de propiedades dada como entrada dentro de la lista de propiedades del objeto mencionado como entrada.

```
?pprop "master ".APV "Scott  
?:master  
Scott
```

## **remprop**

(remove property=quite propiedad) Quita la propiedad especificada de la lista de propiedades del objeto mencionado como entrada.

```
?remprop "master ".APV
```

### 4.10.12 Ficheros en Disco

#### **dir**

(directory=directorio) Saca una lista con los nombres de ficheros en Dr. Logo que están registrados en el disco especificado o prescrito para omisiones; acepta símbolos comodín.

```
?dir  
[STARTUP STARS PATTERNS]  
?dir "b:  
[AVERAGE TOOLS ADDRESSES]  
?dir "??AR????  
[STARTUP STARS]
```

#### **load**

(tráiga, cargue) Toma del disco el fichero mencionado como entrada y lo aloja en el espacio de trabajo.

```
?load "myfile  
?load "b:shapes
```

#### **save**

(guardar, salvar) Lleva el contenido del espacio de trabajo hasta el fichero en disco mencionado como entrada.

```
?save "shapes
```

### 4.10.13 Teclado y Joytick

#### **buttonp**

(button pressed=botón apretado) Saca TRUE (=cierto) si el botón del joystick especificado está apretado; los números 0 ó 1 identifican los dos posibles mandos de juego.

```
?to fire  
>label "loop  
>if (buttonp 0)[pr[fire 0!]]  
>if (buttonp 1)[pr[fire 1!]]  
>go "loop  
>end
```

La inclinación de la palanca de un joystick concreto se puede examinar mediante la función **paddle**.

### **keyp**

(key pressed=tecla pulsada) Saca TRUE (=cierto) si se ha tecleado un carácter y todavía está esperando que sea recogido del teclado.

```
?to inkey
>if keyp [op rc][op "]
>end
```

### **paddle**

(timón, palanca) Devuelve el estado de uno u otro joystick, según la inclinación de la palanca, mediante el siguiente convenio:

<u>Valor devuelto</u>	<u>Significado</u>
255	Sin inclinación
0	Arriba
1	Arriba y derecha
2	Derecha
3	Abajo y derecha
4	Abajo
5	Abajo e izquierda
6	Izquierda
7	Arriba e izquierda

```
?paddle 0
255
```

Los botones de disparo se comprueban mediante la función **buttonp**.

### **rc**

(read character=lea carácter) Saca el carácter correspondiente a la primera tecla que se haya pulsado.

```
?make "key rc
```

(y luego si has pulsado la tecla X)

```
? :key
X
```

rl

(read list=lea lista) Sacar una lista que contiene una línea entera introducida por el teclado; lo tecleado debe ir seguido de un retorno de carro.

```
?make "instr_list rl
repeat 4[fd 50 rt 90]
?:instr_list
[repeat 4[fd 50 rt 90]]
```

rq

(read quote=lea 'cita textual') Sacar una **palabra** que contiene una línea introducida por el teclado; lo tecleado debe ir seguido de un retorno de carro.

```
?make "command rq
repeat 3[fd 60 rt 120]
?:command
repeat 3[fd 60 rt 120]
```

#### 4.10.14 Sonido

Los comandos para sonido son únicos en la implementación AMSTRAD de Dr. LOGO, y son similares a sus contrapartidas en el AMSTRAD BASIC.

Consulta el Capítulo 6 de la Guía del Usuario CPC464 para información adicional.

El comando **SOUND** pone una nota dentro de la secuencia a emitir por un canal. El formato es: [**<estado-canal>** **<período-tono>** **<duración>** **<volumen>** **<envolvente-volumen>** **<envolvente-tono>** **<período-ruido>**] Los parámetros que van detrás de **duración** son opcionales.

```
?sound [1 20 50]
```

env

Estipula una envolvente de volumen. El formato es: [**<número-envolvente>** **<sección(es)-envolvente>**]

```
?env [1 100 2 20]
?sound [1 200 300 5 1]
```

ent

Fija una envolvente de tono. El formato es: [**<número-envolvente>** **<sección(es)-envolvente>**]

```
?ent [1 100 2 20]
?sound [1 200 300 5 1 1]
```

## Release

(Soltar, liberar) Deja que sean emitidas las notas que hayan sido estipuladas como **retenidas** en un comando de sonido. Los canales por donde se emiten las notas **liberadas** se indican como sigue:

<u>Valor de entrada</u>	<u>Canales liberados</u>
0	Ninguno
1	A
2	B
3	B y A
4	C
5	C y A
6	C y B
7	C y B y A

?release 1

### 4.10.15 Control de Ejecución

#### **bye**

(adiós) Se termina la sesión actual con Dr. Logo.

?bye

#### **co**

(continúe) Hace que siga la ejecución después de una pausa provocada por 'espere' (**wait**), un CTRL Z o un ERRACT.

?co

#### **go**

(vaya) Ejecuta la línea dentro del procedimiento vigente que vaya detrás de una **etiqueta** que sea idéntica a la palabra dada como entrada.

>go "loop

#### **if**

(si...) Ejecuta una de dos listas de instrucciones dependiendo del valor que tenga la expresión de entrada; las listas de instrucciones dadas como entrada deben ser listas literales encerradas entre corchetes.

```
>if (a>b)[pr [a is bigger]]  
>[pr [b is bigger]]
```

**label**

(etiqueta, rótulo) Identifica la línea que ha de ejecutarse después de un comando GO que tenga esa palabra como entrada.

```
>label "loop
```

**op**

(output=sacar, poner fuera) Hace que el objeto de entrada del comando sea la salida del procedimiento y que en ese momento se termine su ejecución, volviendo al procedimiento que lo llamó.

**repeat**

(repetir, reiterar) Ejecuta la lista de instrucciones dada como entrada el número de veces dado como entrada.

```
?repeat 4[fd 50 rt 90]
```

**run**

(ejecutar) Ejecuta la lista de instrucciones dada como entrada.

```
?make "instr_list[fd 40 rt 90]  
?run :instr_list
```

**stop**

(pare) Hace que cese la ejecución del procedimiento en curso y que vuelva al TOPLEVEL (la divisa ?) o al procedimiento que puso en marcha el que se ha parado.

```
?stop
```

**wait**

(espera, pausa) Detiene la ejecución del procedimiento durante la cantidad de tiempo especificada por el número de entrada. El intervalo de pausa es igual a número de entrada \* 0.22 segundos.

```
?wait 20
```

#### 4.10.16 Tratamiento de excepciones

##### **catch**

(atrape, cace) Detecta errores y condiciones especiales que ocurren durante la ejecución de la lista de instrucciones dada como entrada.

```
>catch "error+[[][]]
>pr [I am here]
I am here
```

##### **error**

Saca una lista cuyos elementos describen el error más reciente.

```
>catch "error[do.until.error]
>show error
```

##### **pause**

Suspende la ejecución del procedimiento en curso para permitir interacción con el interpretador o el editor.

```
>if:size>5[pause]
```

##### **throw**

(lance, eche) Ejecuta la línea identificada por el nombre de entrada dentro de una expresión **catch** previa.

```
?throw "TOPLEVEL
```

#### 4.10.17 Primitivos del sistema

<b>.contents</b>	Muestra el <b>contenido</b> del espacio para símbolos de Dr. LOGO.
<b>.deposit</b>	<b>Deposita</b> el segundo número de entrada dentro de la celdilla de memoria -en direcciones absolutas- especificada por el primer número de entrada.
<b>.examine</b>	Mira o <b>examina</b> el contenido de una celdilla de memoria -en direcciones absolutas- especificada por el número de entrada.

#### 4.10.18 Variables de Sistema

**ERRACT** (Error Activo)

Cuando es TRUE (=cierto) provoca una pausa cuando ocurre un error y luego regresa a TOPLEVEL (=nivel superior).

**FALSE**

Valor de sistema.

**REDEFN**

(Redefine Procedure) Cuando es TRUE (=cierto) permite la **redefinición** de primitivos.

**TOPLEVEL**

**throw "TOPLEVEL** hará que se salga abandonando todos los procedimientos pendientes, y se regrese al **nivel superior máximo**.

**TRUE**

(Cierto) Valor de sistema.

#### 4.10.19 Propiedades de sistema

**.APV**

Associated property value=Valor de propiedad asociado; el valor de una variable **global**.

**.DEF**

Definición de un procedimiento.

**.PRM**

Identifica un PRiMitivo.

# CAPITULO 5

## Información Técnica para el Usuario -Programa Grabado (Firmware)

Este capítulo presupone un conocimiento anterior del programa grabado en ROM del CPC464.

También se presuponen algunos conocimientos sobre los fundamentos del CP/M. Este capítulo no pretende comentar las posibilidades disponibles dentro del entorno CP/M. La información completa concerniente al CP/M, el entorno CP/M en el CPC464, y la especificación técnica del BIOS, está contenido en el SOFT159 Una Guía al CP/M. La Especificación Completa del Programa Grabado (Firmware) del Amstrad DDI-1 está contenida en el manual SOFT 158A, disponible como un Apéndice a la Especificación Concisa del Programa Grabado SOFT158.

### 5.0 Introducción

AMSDOS es un sistema operativo en disco para el AMSTRAD CPC464 equipado con la interface para disco flexible DDI-1. AMSDOS permite que los programas en BASIC accesen ficheros en disco de una manera similar a los ficheros en cassette, y pudiendo desde luego, los programas existentes que actualmente usan el cassette, ser capaces de manejar los ficheros en disco con muy poca o con ninguna modificación. La principal fuente de incompatibilidades serán los nombres de ficheros, dado que para AMSDOS los nombres de ficheros deben cumplir las normas CP/M mientras que los nombres de ficheros en cassette están mucho menos restringidos.

AMSDOS ha sido diseñado para complementar al CP/M, pero no para competir con él. Ambos comparten la misma estructura de fichero y pueden efectuar lectura y escritura en los ficheros del otro sistema. AMSDOS reside en la misma ROM que el CP/M BIOS.

AMSDOS conmuta los cauces de entrada y salida hacia el cassette (#9) hasta y desde el disco. Así, todas las facilidades disponibles en cinta, también están disponibles en disco. Además, también es posible mostrar el directorio del disco, borrar ficheros en disco, cambiar el nombre de ficheros en disco y elegir la unidad y el usuario prescrito para omisiones.

Estas facultades están implementadas bien mediante la interceptación de las citas a las rutinas que tratan el cassette, o bien mediante comandos externos.

## 5.1 Cabeceras

Los ficheros en cinta están subdivididos en bloques de 2K bytes, cada uno de los cuales va precedido de un **registro de cabecera**. Los ficheros en CP/M no tienen cabeceras. Los ficheros en AMSDOS pueden tener o pueden no tener cabecera dependiendo del contenido del fichero. Eso no causará problemas para los programas escritos en BASIC, pero es una diferencia importante entre ficheros en cinta cassette y en disco. Eso podría quizás explotarse para un esquema de protección de ficheros.

Los ficheros en ASCII sin proteger no tienen cabeceras. Todos los otros ficheros en AMSDOS tienen un solo registro de cabecera dentro de los primeros 128 bytes del fichero. Estas cabeceras son detectadas mediante comprobación de la suma de los primeros 67 bytes del registro. Si dicha comprobación es la esperada, entonces es que hay presente un registro de cabecera, si no lo es, no hay cabecera. Por lo tanto, es poco probable pero posible, que un fichero sin cabecera pueda ser tomado por un fichero con cabecera.

## 5.2 Cambio de Discos

Bajo AMSDOS se puede cambiar un disco, o quitar, siempre que la unidad de disco no esté siendo accesada y siempre que no haya ficheros de entrada o de salida abiertos en esa unidad. A diferencia del CP/M no hay necesidad de "inscribir" un disco en el sistema.

El cambio de un disco mientras todavía se está escribiendo en él, puede 'corromper' la información que haya en el disco. Si se cambia un disco mientras todavía hay ficheros abiertos en él, tan pronto como el AMSDOS lo detecte abandonará todos los ficheros abiertos que haya en la unidad, y emitirá un mensaje de error. Cualquier información que todavía no se hubiera escrito, se perderá; y la más reciente reseña del directorio no quedará reflejada en el disco. Sin embargo, AMSDOS sólo puede detectar este cambio cuando efectúa la lectura del directorio, lo que hace cada 16K bytes del fichero y siempre que se abre o cierra un fichero. Por lo tanto, potencialmente, pueden deteriorarse 16K bytes de información al cambiar un disco mientras todavía hay ficheros abiertos en él.

## 5.3 Requisitos de almacenamiento

Cuando se pone en marcha, el AMSDOS reserva 500 bytes de memoria tomados del 'pozo' o fondo general. No está previsto ningún cambio en este valor.

Al cargar un programa en código máquina desde el disco y alojarlo en la memoria, usando la rutina CAS IN DIRECT del AMSDOS, es importante que no se sobre-escriban las variables del AMSDOS.

Eso presenta un problema, dado que en general no es posible descubrir dónde están esas variables. Eso es debido a que las variables para las ROMs externas se asignan dinámicamente. Observa que este problema no surge cuando se trae el programa desde el cassette, dado que las variables para el tratamiento del cassette están dentro del área de variables **pregrabada**.

AMSDOS reserva almacenamiento a partir de la cima de la memoria general, de manera que la solución más simple es siempre cargar los programas en código máquina dentro de la parte inferior del almacenamiento. El programa puede luego reubicarse a sí mismo si se requiere una dirección más alta.

Alternativamente, el programa en código máquina pudiera cargarse en dos etapas: primero cargar y ejecutar un pequeño **cargador** dentro de la parte inferior de la memoria. La acción del MC BOOT PROGRAM habrá cancelado todas las RSXs y las ROMs de ampliación. El programa cargador debiera implantar AMSDOS usando KL INIT BACK, forzando así a que las variables AMSDOS estén donde quiera que desees. El cargador puede ahora cargar el programa en código máquina usando las subrutinas AMSDOS llamadas CAS OPEN IN, CAS IN DIRECT y CAS IN CLOSE, conjuntamente con el MC START PROGRAM.

Con el fin de implantar el AMSDOS usando KL INIT BACK, se requiere el número de la ROM en AMSDOS. Para determinar el número de la ROM en AMSDOS observa cualquiera de las entradas interceptadas a los bloques de saltos para el cassette, con las rutinas DISC seleccionadas. Cada entrada es una 'llamada lejana' cuya parte de dirección apunta hacia tres bytes de la dirección lejana, y el tercer byte de dicha dirección lejana es el número de ROM. Obviamente, eso debe hacerse antes de que el AMSDOS esté completamente 'cerrado'.

## 5.4 Mensajes de Error

AMSDOS usa el CP/M BIOS con el fin de efectuar los accesos al disco. Por tanto, en el caso de un error de los circuitos se mostrarán los mensajes BIOS.

En lo siguiente <DRIVE > significa el identificativo de unidad A o B. <FILENAME> significa un nombre de fichero según las normas del AMSDOS.

### 5.4.1 Mensajes del AMSDOS

#### **Bad command**

Ese es un **comando malo** y ha fallado en alguna manera. Puede ser un error sintáctico en el comando o en el nombre del fichero, o que el BIOS ha detectado un error y lo comunica con la opción C de cancelación.

#### <FILENAME> already exists

El usuario está intentando cambiar el nombre de un fichero usando un **nombre que ya existe**.

#### <FILENAME> not found

El usuario está intentando abrir como fichero de entrada, borrar o cambiar el nombre de un fichero que **no se ha encontrado**.

#### Drive <DRIVE>: directory full

En la **unidad** de disco está el **directorio lleno**. (Recuerda que en el directorio sólo puede haber 64 reseñas de fichero).

#### Drive <DRIVE>: disc full

En la **unidad** mencionada está el **disco lleno**, es decir, no hay más bloques disponibles para información.

#### Drive <DRIVE>: disc changed, closing FILENAME

El usuario en la **unidad** mencionada **ha cambiado el disco**, y el sistema está **cerrando** el fichero de ese nombre.

#### <FILENAME> is read only

El usuario está intentando borrar o cambiar el nombre de un fichero que está estipulado como de **únicamente lectura**. Puede ser provocado también al cerrar un fichero cuando la versión existente de ese fichero es de **únicamente lectura**.

### 5.4.2 Mensajes del BIOS

Los mensajes del BIOS (sistema 'basilar' de entrada/salida) van seguidos de la pregunta **Retry, Ignore or Cancel?** para que elijas entre reintentar, descartar o cancelar la operación mandada. El sistema desecha entonces cualquier carácter que haya pendiente, muestra el cursor, y espera que el usuario le teclee **R, I** o **C**, según desee. Cualquier otra cosa que teclee, producirá un pitido de aviso.

Tecleando **R** para reintentar, hace que el BIOS repita la información mandada.

Tecleando **I** por 'ignorar', hace que el BIOS continúe como si no hubiera ocurrido el percance.

Tecleando **C** para cancelar, hace que el BIOS abandone la operación mandada. Eso a menudo da como resultado un mensaje de error emitido por el BDOS.

Después de que el usuario haya tecleado **R, I** o **C**, el cursor se oculta de nuevo.

Los mensajes del BIOS AMSTRAD son los siguientes:

**Drive <DRIVE >: disc missing**

El mensaje se produce cuando el BIOS intenta acceder la **unidad** mencionada en la que **falta el diskette**, ya sea porque no está o porque no está bien cerrada la portezuela de la unidad.

**Failed to load boot sector**

Este mensaje se produce durante un 'arranque en frío' (exigido mediante **CPM**) e indica que el sistema **ha fallado en cargar el sector de "botadura"**, ya sea porque no pueda hacer una lectura correctamente o porque todos los bytes de dicho sector tienen el mismo valor.

**Failed to load CP/M**

Este mensaje se produce durante un 'arranque en caliente' cuando un sector del CCP o del BDOS no ha sido leído correctamente, o si todos los bytes del primer sector CCP tienen el mismo valor; en ambos casos el mensaje indica que **ha fallado al cargar CP/M**.

**Drive <DRIVE >: disc is write protected**

El mensaje indica que en la **unidad** mencionada el **disco está protegido ante escritura**, y se produce cuando el BIOS intenta grabar algo en un disco que está protegido ante escritura. Si el usuario desea escribir en ese disco, debe primeramente quitar el disco, facultarle para que pueda escribirse en él, y volver a meterlo en la unidad; y luego teclear **R** para reintentar la operación.

**Drive <DRIVE >: read fail**

Este mensaje indica que en la **unidad** mencionada ha habido un **fallo de lectura**, y es un error detectado por los circuitos al efectuar la lectura desde el disco. Puede estar provocado al intentar tomar información de un disco que tiene el formato erróneo, por ejemplo: intentando efectuar la 'botadura' desde un disco cuyo formato es DATA ONLY (únicamente información).

**Drive <DRIVE >: write fail**

Este mensaje indica que en la **unidad** mencionada ha habido un **fallo de escritura** y se produce cuando los circuitos han detectado un error mientras estaban escribiendo información en el disco.

En el caso de un fallo de lectura o de escritura se recomienda al usuario que quite y vuelva a meter el disco en la unidad, y luego teclee **R** para reintentar la operación. Eso puede ayudar en los casos en que el disco estaba mal colocado o puede trasladar cualquier polvillo que esté adherido al cabezal. La importancia de las **copias de respaldo** nunca está de más resaltarla.

## 5.5 Organización del Disco

El BIOS admite tres formatos diferentes para los discos: formato SYSTEM, formato DATA ONLY (únicamente información) y formato IBM. Bajo el AMSDOS el formato del disco se examina automáticamente cada vez que se efectúa un acceso al disco con ningún fichero abierto en él. Para permitir esta detección automática, cada formato tiene números de sector **únicos** que lo identifican.

Los discos de 3 pulgadas son de **doble cara** pero sólo puede efectuarse el acceso a una cara en cada momento, y dependiendo de la manera en que el usuario haya insertado el disco en la unidad. Puede haber diferentes formatos en una y otra cara del disco.

### COMUN A TODOS LOS FORMATOS

Simple cara (las dos caras de un disco de 3 pulgadas se tratan separadamente).

Tamaño de sector físico de 512 bytes.

40 pistas numeradas de 0 a 39.

Tamaño de bloque CP/M de 1024 bytes.

64 reseñas o entradas en el directorio.

### FORMATO SYSTEM

9 sectores por pista numerados #41 a #49.

2 pistas reservadas.

El formato **sistemat** es el principal formato de los admitidos, dado que el CP/M sólo puede ser traído y cargado en memoria ('arranque en caliente o en frío') desde un disco con el formato **sistemat**. Las pistas reservadas se usan de la manera siguiente:

Pista 0 sector #41: sector de 'botadura'.

Pista 0 sector #42: sector de 'configuración'.

Pista 0 sectores #43..#47 sin usar.

Pista 0 sectores #48..#49 y pista 1 sectores #41..#49: CCP y BDOS.

**Nota:** El formato 'VENDOR' (vendedor) es una versión especial del formato 'sistemat' que no contiene ninguna información en las dos pistas reservadas. Está pensado para que lo usen las casas de distribución de programas.

### FORMATO DATA ONLY (únicamente información)

9 sectores por pista numerados #C1..#C9.

0 pistas reservadas.

Este formato pretende servir para mejoras futuras y no se recomienda su utilización con el CP/M, dado que no es posible efectuar un 'arranque en caliente' a partir de él.

Sin embargo, si solamente va a usarse el AMSDOS con este formato se dispone de un poco más de espacio de almacenamiento en el disco.

## FORMATO IBM

8 sectores por pista numerados 1..8.

1 pista reservada.

Este formato es obviamente el mismo que el formato de simple cara usado por el CP/M en el IBM PC. Está pensado para ser usado sólo por especialistas y no se recomienda en otro caso.

### 5.6 Interceptación de los Accesos al Bloque de Saltos - por AMSDOS

Cuando se implanta el AMSDOS en el sistema, en el primer momento copia lo reseñado en el **bloque de saltos** relativos a las operaciones con el cassette, dentro de su propia área de datos. Cuando se elige DISC, lo reseñado en el bloque de saltos para el cassette queda **sobre-escrito** por las propias reseñas del AMSDOS; cuando se elige TAPE vuelve a reponerse en su sitio las reseñas originales relativas a la **cinta** en cassette.

En el momento inicial, está prescrito que se elijan las rutinas correspondientes al trabajo con discos.

### 5.7 Re-Interceptación de los Accesos al Bloque de Saltos - por el Usuario

Con el fin de interceptar cualquier intento de acceso al bloque de saltos, deberá observarse el siguiente procedimiento: copiar los tres bytes obtenidos de la reseña requerida del **bloque de saltos** dentro de tu propia área de datos - y sin hacer ninguna suposición en cuanto a lo que significan esos tres bytes. Sustituye lo reseñado en el bloque de saltos con tus propias instrucciones de salto (JMP), reinicio (RST) o la que sea. Cuando se te ceda el control, vuelve a reponer lo reseñado en el bloque de saltos, y da una instrucción de apelación o llamada (CALL). Cuando recibas el control de nuevo, guarda la reseña del bloque de saltos y sustitúyela con la tuya propia. Este procedimiento funcionará cualquiera que sea lo reseñado en el bloque de saltos.

**Nota:** Al interceptar las apelaciones a las rutinas del AMSDOS, debe seguirse el procedimiento anteriormente mencionado. Si simplemente efectúas una copia de la reseña del bloque de saltos, no funcionará; debe volver a almacenarse en su sitio original dentro del bloque de saltos.

## 5.8 Parámetros devueltos por la rutina

Hasta donde es posible, las rutinas del AMSDOS tienen el mismo **nexo** de comunicación que sus contrapartidas para cinta en cassette, aunque en algunos casos la interpretación de los parámetros devueltos es diferente. Los errores que son detectados tanto por las rutinas en cinta o en disco, hacen que se devuelva el testigo de acarreo como falso, y el testigo de cero como falso. Los errores que únicamente han sido detectados por las rutinas para el disco, se devuelven con el testigo de acarreo como falso y el de cero como cierto. Este último caso corresponde a la condición de **ruptura** (BREAK) de la rutina para la cinta en cassette. En ambos casos, el registro A del microprocesador contiene el número de error pertinente.

Cuando falla una rutina (testigo de acarreo falso) entrega en el registro A del microprocesador, un número de error de 6 bits. El bit 7 está alzado si ese error ya ha sido comunicado al usuario. Los números de error son como siguen:

#0E	el fichero no está abierto como se esperaba.
#0F	marca <b>firme</b> del final del fichero.
#1A	marca <b>suave</b> del final del fichero.
#20	mal comando, usualmente provocado por un nombre de fichero incorrecto.
#21	fichero ya existe.
#22	fichero no existe.
#23	directorio lleno.
#24	disco lleno
#25	se ha cambiado el disco con ficheros abiertos en él.
#26	fichero de únicamente lectura.

Los errores detectados por los circuitos controladores de la unidad de disco flexible se reportan **calibrados** en binario, con un valor entre #40.. #7F, i.e. con el bit 6 siempre alzado. Los otros bits corresponden a lo siguiente:

bit 5 error en datos	error CRC en los datos o en el campo ID.
bit 4 error desbordamiento	
bit 3 unidad no dispuesta	no hay disco en la unidad
bit 2 sin datos	no puede encontrar el sector
bit 1 no escribible	el disco está protegido ante escritura
bit 0 falta la marca de dirección	

## 5.9 Interceptación de las Apelaciones al Programa en ROM

Cuando se recurre al programa en ROM para que efectúe una tarea determinada en relación con la cinta en cassette, las **llamadas** interceptadas son:

### Lectura de fichero

125	#BC77	CAS IN OPEN	Abre un fichero de entrada
126	#BC7A	CAS IN CLOSE	Cierra apropiadamente el fichero de entrada.
127	#BC7D	CAS IN ABANDON	Cierra inmediatamente el fichero de entrada.
128	#BC80	CAS IN CHAR	Ingresa un carácter desde el fichero de entrada.
129	#BC83	CAS IN DIRECT	Ingresa en memoria el fichero de entrada.
130	#BC86	CAS RETURN	Devuelve el último carácter ingresado
131	#BC89	CAS TEST EOF	Comprueba si se ha alcanzado el final del fichero de entrada.

### Escritura en ficheros

132	#BC8C	CAS OUT OPEN	Abre un fichero de salida
133	#BC8F	CAS OUT CLOSE	Cierra apropiadamente el fichero de salida.
134	#BC92	CAS OUT ABANDON	Cierra inmediatamente el fichero de salida.
135	#BC95	CAS OUT CHAR	Escribe un carácter en el fichero de salida.
136	#BC98	CAS OUT DIRECT	Escribe el fichero de salida directamente desde memoria.

### Exposición de catálogo

137	#BC9B	CAS CATALOG	Obtiene el catálogo de ficheros en la cinta.
-----	-------	-------------	--

# APENDICE 1

## GLOSARIO DE TERMINOS

### Ambiguous File Name:

**Ambiguo**

Se aplica a un nombre de fichero que contiene uno o más símbolos comodínes. Los nombres de ficheros ambigüos estrictamente hablando son nombres **colectivos** ya que hacen referencia no a un fichero concreto, sino a más de un fichero de lo registrados en el disco.

### AMSDOS:

Sistema Operativo en Disco de AMStrad. El programa **sistemat** que permite al BASIC Locomotive poder efectuar accesos a los ficheros grabados en disco.

### Backup:

**Respaldo**

Se aplica a la copia de la información que se utiliza como **resguardo** o salvaguardia en caso de que se pierda o se dañe accidentalmente la información original. Obtener una copia de respaldo se refiere al proceso de duplicar un disco o un fichero de un disco.

### BDOS:

Sistema Operativo Basilar en Disco. Es la parte del sistema operativo CP/M que constituye el lazo de unión entre el programa de usuario y las funciones disponibles en el CP/M para operar con discos.

### BIOS:

Sistema Basilar de Entrada/Salida. Es la parte del CP/M que depende de los circuitos concretos de un ordenador, y que se programa específicamente para cada clase de ordenador. Toda la información de entrada y salida en relación con la pantalla, y el teclado se efectúa a través de este sistema.

### Boot:

**Botadura**

El proceso de cargar un sistema operativo en la memoria del ordenador. Normalmente llamado **implantación**. Cuando se inicia la operación con el CP/M a partir del BASIC se trae automáticamente desde el disco y se carga en memoria un programa de **botadura** o de **carga inicial**, que se encarga a su vez de traer a la memoria desde el disco el resto del sistema operativo.

**Buffer:**

Una zona reservada de memoria para almacenar temporalmente, para **depositar** en ella, información cuando se están efectuando transferencias entre los equipos periféricos y la memoria.

**Built-in commands:****Incorporados**

Se aplica a los comandos que son parte intrínseca de un sistema operativo. Son siempre más rápidos que los comandos **transeúntes** porque no tienen que ser traídos desde el disco a la memoria. También se llaman residentes y permanentes.

**CCP:**

Procesador de Comandos de Consola. Es un módulo o parte del programa CP/M encargado de interpretar y cumplimentar las órdenes que el usuario introduce a través del teclado. Usualmente, los comandos son los mandatos dados por el operador y que permiten que se carguen programas en memoria y se les ceda el control del ordenador.

**Cold start:****Arranque en frío**

El proceso de efectuar la implantación en memoria de **todo** el sistema operativo. Se efectúa un arranque en caliente del CP/M en este ordenador cuando se da el comando **CPM**.

**Console mode:****Modo Consola**

Es el modo directo del CP/M caracterizado porque aparece en pantalla el símbolo **A** , que indica la unidad de disco prescrita como activa en ese momento y que el sistema está preparado para recibir comandos del CP/M intrínsecos, o correspondientes a los programas utilitarios (utensilios) que están depositados en el disco.

**Corruption:****Corrupción**

Se aplica a la información, sea programas o datos, e indica que se ha destruido, alterado o deteriorado el contenido de un fichero en disco o en memoria, de una manera no deseable y potencialmente no puede recuperarse esa información.

**CP/M:**

Programa de Control para Microcomputadoras. Es un sistema operativo basado en disco -una colección de programas para operar el sistema- creada por Digital Research y que constituye el nexo o lazo de unión standard con los programas escritos para una amplia variedad de sistemas de cómputo basados en microprocesadores.

**Default:****Prescrito para omisiones**

Se aplica a cualquier valor o nombre que es el que se toma en ausencia de cualquier valor explícito dado por el usuario en un comando. Por ejemplo, cuando se inicia la operación con el CP/M, la unidad prescrita es la A: se dice también que es la vigente y la activa.

**Directory:****Directorio**

Una sección del disco que contiene **reseñas** de cada fichero que esté depositado en ese disco. Es por tanto, una lista o catálogo, del contenido de un disco.

**Disc (or disk):****Disco**

Una pieza circular delgada y plana de plástico, recubierto sobre una u otra cara con una capa de óxido magnético que se usa como soporte para el almacenamiento de información. El disco está alojado en un sobre protector, con aberturas para que los cabezales de lectura/escritura puedan tener acceso a la superficie magnética. En un disco de tres pulgadas esa abertura está protegida por un obturador metálico que automáticamente se desliza a lo ancho cuando el disco se saca de la unidad.

**Disc drive:****Unidad de disco**

El equipo electrónico mecánico usado para efectuar el giro del disco y para efectuar el acceso a la información grabada en la superficie magnética del disco. Cada acceso puede ser para efectuar una lectura o una escritura de información.

**Disc Interface:****Interface**

Los circuitos electrónicos necesarios para permitir que el equipo de almacenamiento en disco pueda comunicarse tanto con los programas como con los circuitos internos del ordenador al que está acoplado.

**Double sided:****Doble cara**

Un disco que puede almacenar información sobre ambos lados de la superficie. Una unidad para discos de doble cara puede hacer los accesos a ambas caras del disco sin necesidad de quitarlo de la unidad y volverlo a poner por la cara opuesta.

**Dr. Logo:**

La versión de Digital Research del lenguaje de programación Logo desarrollado en el MIT por Seymour Papert, que emplea gráficos de tortuga y desglose de programas en procedimientos.

<b>Edit:</b>	<b>Edición</b>
Revisar, corregir o hacer cambios en datos, programas o textos.	
<b>Expression:</b>	<b>Expresión</b>
Se usa en Dr. Logo para indicar una <b>frase</b> que consta de un nombre de procedimiento seguido de cualquier dato necesario de <b>entrada</b> al procedimiento.	
<b>File:</b>	<b>Fichero</b>
Una colección de datos, que guardan cierta relación entre sí, y que generalmente se almacenan o depositan en cinta magnética o en disco magnético. También se llama archivo.	
<b>File name:</b>	<b>Nombre de fichero</b>
La serie de caracteres que permite identificar a un fichero con respecto a los otros. En Dr. Logo, un nombre de fichero puede tener hasta 8 caracteres alfabéticos o numéricos. En CP/M está permitido una parte adicional en el nombre para identificar la clase de fichero -el apellido- que va separada del nombre <b>propio</b> por un punto (.).	
<b>Hex file:</b>	<b>Fichero en hexadecimal</b>
Una representación ASCII de un fichero de comandos o de un fichero con un programa en código máquina.	
<b>Integer:</b>	<b>Entero</b>
Un número según el sistema decimal que no tiene parte fraccionaria (sin punto).	
<b>Interface:</b>	<b>Interface</b>
Un circuito electrónico que permite <b>acoplar</b> dos sistemas independientes, tal como un CPC464 y una unidad de almacenamiento en disco, para que pueda intercambiarse información. Constituye el <b>nexo</b> o lazo de unión entre los dos.	
<b>Logical device:</b>	<b>Equipo logical</b>
Es un <b>artificio</b> de programación, con el que se logra la representación de un equipo <b>fisical</b> cuando se está trabajando con un programa de sistema o de usuario. Por ejemplo, el equipo logical del CP/M que usa como <b>listador</b> se denomina LIST, y en un equipo concreto y en un momento dado puede estar asociado concretamente al equipo fisical que es la pantalla o cualquier periférico de salida. Se denomina también equipo simbólico y equipo abstracto.	

**Logo:**

El nombre de un lenguaje de programación derivado de la palabra griega **logos** que significa palabra, razonamiento, 'lógica'. Fue diseñado en el MIT bajo la dirección de Seymour Papert pretendiendo servir para enseñar los fundamentos de la programación de computadoras.

**Node:****Nudo**

Una unidad de almacenamiento dentro del espacio de trabajo utilizado por Dr. Logo. Típicamente un nudo ocupa 4 bytes de memoria.

**Page zero:****Página cero**

Con esto nos referimos a la región de memoria dentro de la ambientación del CP/M entre las direcciones 0000Hex y 0100Hex que el sistema CP/M emplea para conservar en ella los parámetros vitales de su funcionamiento.

**Physical device:****Equipo físico**

Es realmente el **artefacto** -el aparato- que consta de circuitos electrónicos y componentes, que existe material y realmente. Son estos equipos físicos los que están representados en abstracto mediante los equipos lógicos usados al programar.

**Primitives:****Primitivos**

Se aplica a los procedimientos con que está dotado originalmente el Dr. Logo. Son los intrínsecos o ya incorporados en el sistema. Los que el usuario emplea para construir sus procedimientos **derivados**.

**Procedure:****Procedimiento**

Un párrafo, una serie de expresiones o instrucciones que describen cómo ha de llevarse a cabo una **tarea** particular. Se abrevia llamándole **proce** y es similar a las rutinas y subrutinas de otros lenguajes.

**Prompt:****Aviso, mensaje**

Por un lado se refiere a los breves mensajes o series de caracteres que recuerdan al usuario que está esperando se le teclee algo. Pero además, cuando es el símbolo característico de un programa sistémico, tal como el del CP/M, o el ? del Dr. Logo, se le suele llamar **divisa** del sistema.

**Random access:****Acceso referencial**

La operación de efectuar una lectura o una escritura en un determinado periférico de almacenamiento, sea disco o cinta.

La habilidad de efectuar acceso a un determinado punto de la memoria o de un disco (pero no de una cinta) porque está **identificado** de una manera directa, normalmente mediante un número correlativo. Dado que el acceso es a discreción del usuario, sin estar supeditado a ningún otro acceso previo, puede parecer a un ajeno que lo efectúa 'a lo loco', de manera que en un principio recibió el nombre de acceso aleatorio (random).

**Read only R/O:**

**R/O Unicamente lectura**

Es un atributo asignado a un disco, a un fichero en un disco o a una unidad de disco, que le impide que se efectúe escritura o alteración de la información contenida.

**Read write R/W:**

**R/W Escritura/Lectura**

Un atributo asignado a un disco, a un fichero de un disco o a una unidad de disco, que le permite efectuar tanto la lectura de la información contenida como la escritura de nueva información.

**Record:**

**Registro**

Un grupo consecutivo de bytes dentro de un fichero que se trata como una sola entidad. El CP/M usa registros de 128 bytes.

**Sector:**

**Sector**

Un bloque de datos en un disco. Físicamente, ocupa un segmento curvilíneo o trozo de corona en la superficie del disco. El sistema de almacenamiento en disco AMSTRAD usa un tamaño para el sector de 512 bytes.

**Single sided:**

**Simple cara**

Se refiere a un disco que sólo tiene uno de sus lados disponible para grabar información.

**System tracks:**

**Pistas del sistema**

Son las pistas -coronas circulares- reservadas en el disco para el propio sistema CP/M.

**TPA:**

Area de Programas Transeúntes (Transient Program Area). Es un área dentro de memoria gestionada por el CP/M, y que comienza a partir de la dirección 0100Hex, donde se alojan los datos y programas que funcionan bajo control del CP/M. El propio CP/M la emplea para cargar en ella los programas utilitarios -los transeúntes o utensilios- que se le piden.

**Track:****Pistas**

Las pistas son los anillos concéntricos -las coronas- en que está dividida la superficie magnética de un disco. Cada pista se divide a su vez según un número prefijado de sectores. Las pistas y los sectores se definen realmente dentro de todo el área del círculo cuando se efectúa la operación de formateo del disco.

**Transient program:****Programa transeúnte**

Un programa de utilidad del CP/M tal como **FILECOPY** que permite ser traído del disco y cargado en la TPA, y al que se le cede control para que efectúe la tarea correspondiente; simplemente basta teclear su nombre para que el CP/M lo ponga en marcha. Se denominan también utensilios de programación y programas utilitarios.

**Turnkey:****Llave en mano**

Es un adjetivo que se usa para describir un programa que automáticamente se ejecuta cuando se implanta el sistema en memoria. El disco de Dr. Logo es un ejemplo de esta clase de programas.

**Turtle:****Tortuga**

Un símbolo gráfico que tiene forma de cabeza de flecha, y que funciona como un cursor de gráficos dentro de la pantalla de gráficos manejada por Dr. Logo. Podría llamarse **pincel**.

**Turtle graphics:****Gráficos de tortuga**

La imagen que va dejando en la pantalla -la estela- la tortuga cuando se desplaza a través de ella. Se obtienen básicamente por analogía del trayecto seguido por un barco al navegar.

**Turtle step:****Paso de tortuga**

(No es que sea muy lento) simplemente es la distancia más pequeña que se toma como unidad para hacer que la tortuga se mueva. Normalmente corresponde a un **elemento de imagen** (pixel) cuando se habla de pantallas de gráficos.

**Utility program:****Utensilio de programación**

Un programa en disco que permite al usuario efectuar ciertas tareas con él, que normalmente son de las denominadas domésticas: copiar, formatear, etc. Véase programa transeúnte.

**Warm start:****Arranque en caliente**

Se efectúa al pulsar [CTRL] C al estar trabajando directamente bajo CP/M. Un arranque en caliente efectúa la implantación en memoria de **parte del sistema operativo** -porque el resto se supone que no ha cambiado al haber estado trabajando antes-, y al hacerlo se entrega de nuevo el control al CP/M que se coloca en el modo directo para recibir comandos.

**Wildcard character:****Comodín**

Uno de los símbolos asterisco o interrogación (\* ?), que está admitido en el CP/M y sirve para **concordar** con cualquier otro símbolo. Dr. Logo sólo admite el símbolo interrogación ? como comodín. El símbolo asterisco \* significa simplemente cualquier número de interrogaciones. Al hacer referencia a un fichero, se usa el símbolo comodín para constituir lo que se llama un nombre de fichero **ambigüo**. Cualquier ? incluida en el nombre de un fichero hace referencia a cualquier símbolo numérico o alfabético del repertorio, que realmente haya en un nombre de fichero en esa posición.

**Write Protection:****Protección ante escritura**

Una medida preventiva usada para impedir que se sobre-escriba -o simplemente se escriba- en un disco o en un fichero de un disco. Por tanto, un disco o un fichero protegido ante escritura, se dice que tiene el atributo de **únicamente lectura**.

# APENDICE 2

## DIGITAL RESEARCH & AMSTRAD ACUERDO DE LICENCIA PARA EL USUARIO FINAL DEL PROGRAMA

NOTA PARA EL USUARIO - POR FAVOR LEE CUIDADOSAMENTE ESTE TEXTO. NO ABRAS EL PAQUETE CON EL DISKETTE HASTA QUE HAYAS LEIDO ESTE ACUERDO DE LICENCIA.

ABRIR EL PAQUETE CON EL DISKETTE INDICA QUE TU ACUERDO QUEDA REGIDO POR ESTOS TERMINOS Y CONDICIONES.

### 1. DEFINICIONES

En este Acuerdo de Licencia, los términos:

1. DRI significa DIGITAL RESEARCH (CALIFORNIA) INC., P.O. Box 579, Pacific Grove, California 93950, propietario de los derechos, o licenciatario autorizado del programa.
2. Máquina significa el microcomputador singular en el que usas el programa. Los sistemas de múltiples CPU requieren licencias adicionales.
3. Programa significa el conjunto de programas, la documentación y todos los materiales relacionados dentro de este paquete, juntamente con todas las actualizaciones y mejoras auxiliares que te suministra DRI sin tener en cuenta la manera en que puedes usarlas subsecuentemente, y sin tener en cuenta ninguna de las modificaciones que puedas efectuar a ellas.
4. AMSTRAD significa AMSTRAD CONSUMER ELECTRONICS PLC., Brentwood House, 169 Kings Road, Brentwood, Essex CM14 4EF.

El usuario final asume la responsabilidad durante la selección del programa para conseguir los resultados pretendidos, y para la instalación, utilización y resultados obtenidos del programa.

### 2. LICENCIA

Puedes:

1. Usar el programa en una máquina singular, de una sola CPU.
2. Copiar el programa en cualquier máquina, en forma impresa o legible para respaldo o con propósitos de modificación en apoyo de tu uso del programa sobre una máquina singular. Puedes sacar hasta tres (3) copias del programa para tales propósitos. (Ciertos programas pueden sin embargo, incluir mecanismos para limitar o prohibir el copiado. Son los que están marcados 'copy protected'). La copia de documentación y de otro material impreso está prohibida.

Está prohibido desensamblar el código recibido.

3. Modificar el programa y/o congregarlo con otro programa para tu uso en una máquina singular. (Cualquier porción de este programa incluida dentro de otro programa continuará estando sujeta a los términos y condiciones de este Acuerdo).
4. Transferir el programa y la licencia a un tercero si notificas a DRI el nombre y la dirección de dicho tercero y este acuerda a) aceptar los términos y condiciones de este acuerdo; b) firmar y enviar a DRI una copia de la Carta de Registro y c) pagar la tarifa de transferencia vigente en ese momento. Si transfieres el programa, debes al mismo tiempo transferir también todas las copias, incluyendo el original, sea en forma impresa o legible por máquina para el tercero, o destruir cualquier copia no transferida; eso incluye todas las modificaciones y porciones del programa contenidas en o congregadas con otros programas.

Debes reproducir e incluir la nota de derechos de copia en cualquier copia, modificación o porción incluida en otro programa.

**CADA DISKETTE TIENE SU NUMERO DE SERIE, Y NO PUEDES USAR, COPIAR, MODIFICAR, TRANSFERIR, O EN NINGUN OTRO CASO HACER QUE ESTE DISPONIBLE A UN TERCERO, EL PROGRAMA, O CUALQUIER COPIA, MODIFICACION O PORCION CONGREGADA EN TODO O EN PARTE, EXCEPTO TAL Y COMO EXPRESAMENTE ESTA PROVISTO PARA ELLO EN ESTE ACUERDO DE LICENCIA.**

**SI TRANSFIERES LA POSESION DE CUALQUIER COPIA, MODIFICACION O PORCION CONGREGADA DEL PROGRAMA A UN TERCERO, TU LICENCIA QUEDA AUTOMATICAMENTE CANCELADA.**

### 3. PLAZO

La licencia es efectiva hasta que esté cancelada. Puedes concluir la en cualquier otro momento destruyendo el programa conjuntamente con todas las copias, modificaciones y porciones congregadas en cualquier forma. También termina mediante las condiciones estipuladas en cualquier parte de este contrato, o si fallas en el cumplimiento de cualquier término o condición de este Acuerdo. También acuerdas que en el caso de tal cancelación destruirás el programa juntamente con todas las copias, modificaciones, y porciones congregadas en cualquier manera.

#### 4. GARANTIA LIMITADA

**ESTE PROGRAMA SE SUMINISTRA "TAL COMO ESTA". NI DRI NI AMSTRAD DAN NINGUNA GARANTIA DE NINGUNA CLASE, NI EXPLICITA NI IMPLICITA, INCLUYENDO PERO NO LIMITADO A LAS GARANTIAS IMPLICITAS DE COMERCIABILIDAD Y ADECUACION A UN PROPOSITO PARTICULAR. EL RIESGO COMPLETO EN CUANTO A LA CALIDAD Y FUNCIONAMIENTO DEL PROGRAMA ESTA EN TI. SI EL PROGRAMA SE DEMOSTRARA QUE ESTA DEFECTUOSO, TU (Y DRI O AMSTRAD) ASUMIRAN EL COSTE ENTERO DE TODO EL SERVICIO, REPARACION O CORRECCION NECESARIO.**

Ni DRI ni AMSTRAD garantizan que las funciones contenidas en el programa satisfarán tus requisitos, ni que la operación del programa sea ininterrumpida o libre de errores.

Sin embargo, AMSTRAD garantiza que el diskette en que se proporciona el programa, está libre de defectos en materiales y mano de obra, bajo el uso normal durante un período de noventa (90) días a partir de la fecha en que se te entregue, tal y como quede evidenciada por la copia de tu recibo.

#### 5. LIMITACIONES DE REMEDIOS

La responsabilidad entera de AMSTRAD y su remedio exclusivo será la sustitución de cualquier diskette que no satisfaga esta "Garantía Limitada" y que sea entregado a AMSOFT con una copia de tu recibo.

**EN NINGUN CASO SERA RESPONSABLE DRI O AMSTRAD POR NINGUN DAÑO, INCLUYENDO CUALQUIER PERDIDA DE BENEFICIOS, PERDIDA DE AHORROS, O CUALQUIER OTRO DAÑO INDIRECTO O SUBSECUENTE ESPECIAL, INCLUSO AUNQUE DRI O AMSTRAD HAYAN SIDO AVISADOS DE LA POSIBILIDAD DE TALES DAÑOS, O POR CUALQUIER RECLAMACION DE UNA TERCERA PARTE.**

#### 6. CARTA DE REGISTRO

DRI puede, de vez en cuando, actualizar sus programas. Las actualizaciones se te suministrarán sólo si en el fichero está una Carta de Registro adecuadamente firmada, en la oficina principal de DRI o en el remite autorizado por la carta de registro. DRI no está obligado a hacer ninguna actualización del programa, ni a suministrarte ninguna de tales actualizaciones.

## 7. GENERAL

No puedes sublicenciar, asignar o transferir la licencia, o el programa, excepto como expresamente está provisto en este acuerdo. De lo contrario, cualquier intento para sublicenciar, asignar o transferir cualquiera de los derechos, deberes u obligaciones aquí estipulados es nulo.

Este acuerdo estará regido por, y construido de acuerdo con las leyes de Inglaterra.

Si tienes alguna pregunta concerniente a este Acuerdo, puedes ponerte en contacto con DRI, escribiendo a Digital Research Inc., P.O. Box 579, Pacific Grove, California 93950.

**ESTE ACUERDO NO PUEDE Y NO SERA MODIFICADO POR ORDENES DE COMPRA, ANUNCIOS U OTRAS REPRESENTACIONES DE ALGUIEN, Y SOLAMENTE PUEDE MODIFICARSE POR UNA ENMIENDA ESCRITA EJECUTADA POR TI Y UN REPRESENTANTE AUTORIZADO DE DRI Y AMSTRAD.**

**RECONOCES QUE HAS LEIDO ESTE ACUERDO, LO HAS COMPRENDIDO Y ACEPTAS ESTAR SUPEDITADO A SUS TERMINOS Y CONDICIONES. ADEMAS, ACUERDAS QUE ES COMPLETO Y QUE EL ESTABLECIMIENTO EXCLUSIVO DEL ACUERDO ENTRE TU Y DRI Y AMSTRAD, PREDOMINA SOBRE CUALQUIER ACUERDO PROPUESTO O PREVIO, ORAL O ESCRITO, Y CUALQUIER COMUNICACION ENTRE TU Y DRI O AMSTRAD EN RELACION CON EL TEMA SUJETO DE ESTE ACUERDO:**

**ESTE ACUERDO NO AFECTA A TUS DERECHOS ESTATUTARIOS.**

# APENDICE 3

## INDICE

A	F4.2 C2.8	Data only formato	C5.6
AMSDOS	F4.1 C2.1	DDT	C3.12
AMSDOS resumen comandos	C2.8	.DEF	C4.28
AMSDOS mensajes		.DEPOSIT	C4.27
de error	C2.14 C5.3	DIR (CP/M)	C3.4
AMSDOS nombres de fichero	C2.2	DIR (LOGO)	C4.22
AMSDOS cabeceras	C2.3 C5.2	DIR	C2.9
AMSDOS comodínes	C2.4	Directos comandos	
AND	C4.11	de consola	C3.3
.APV	C4.28	Discos	F2.1 C5.2
Aritméticas operaciones	C4.10	DISC	F4.2
Aritméticas operadores		DISCCHK	F4.13
(+ - * /)	C4.11	DISCCOPY	F4.9
ASCII	C2.12 C4.7	Disco directorio	C2.2
ASM	C3.12	Disco comandos de fichero	
B	F4.2 C2.8	(LOGO)	C4.22
BASIC disco	C1.2	DISC.IN	F4.3
BF	C4.8	Disco organización	C5.6
BIOS mensajes	C5.4	DISC.OUT	F4.3
BK	4.17	DOT	C4.16
BL	C4.8	DRIVE	C2.9
BOOTGEN	C3.9	DUMP	C3.12
BUTTONP	C4.22	ED	C4.14
BYE	C4.25	Edición	C4.6 C4.14
CAT	F4.8	EMPTYP	C4.8
CATCH	C4.27	END	C4.13
CHAR	C4.8	ENV	C4.24
CLEAN	C4.15	ENT	C4.24
CLOAD	C3.7	ER	C4.20
CO	C4.25	ERA	C3.4
Configuración	C3.2	ERA	C2.10
Configuración de un programa		ERN	C4.20
CP/M	C1.5	ERRACT	C4.28
.CONTENTS	C4.27	ERROR	C4.27
Control códigos de	C3.2	Error mensajes de	
COPYDISC	F4.9 C1.3	(AMSDOS)	C2.14 C5.3
COS	C4.10	Error mensajes de	
COUNT	C4.8	(CP/M BIOS)	C5.4
CP/M	C3.1	.EXAMINE	C4.27
CPM	F4.4 C2.9	Excepciones tratamiento de	C4.27
CP/M nombres de fichero	C3.3	FALSE	C4.28
CP/M Pistas de Sistema	C3.2	FD	C4.28
CS	C4.15	FENCE	C4.16
CSAVE	C3.7	FILECOPY	C1.3
CT	C4.14	FIRST	C4.9

FORMAT	F4.5 C3.10	RANDOM	C4.10
FPUT	C4.9	RC	C4.23
FS	C4.16	RECYCLE	C4.21
GLIST	C4.21	DEREDFP	C4.28
Glosario de términos	Apl.1	RELEASE	C4.25
GO	C4.25	REMPROP	C4.22
Gráficos comandos		REN	C3.5
de pantalla de	C4.15	REN	C2.10
GPROP	C4.21	REPEAT	C4.26
HT	C4.18	RL	C4.24
IBM formato	C5.7	RQ	C4.24
IF	C4.25	RT	C4.19
INT	C4.10	RUN (AMSDOS)	F4.8
ITEM	C4.9	RUN (LOGO)	C4.26
KEYP	C4.23	SAVE (AMSDOS)	F4.7
LABEL	C4.26	SAVE (LOGO)	C4.22
LIST	C4.9	SE	C4.9
LOAD (AMSDOS)	C4.9	SETH	C4.19
LOAD (LOGO)	C4.22	SETPAL	C4.16
LOCAL	C4.13	SETPC	C4.19
Lógicas operaciones	C4.11	SETPOS	C4.9
Lógicos operadores (= > <)	C4.12	SETSPLIT	C4.14
LOGO	C4.1	SETUP	C3.8
LT	C4.18	SF	C4.16
MAKE	C4.13	SHOW	C4.15
MOVCPM	C3.11	SIN	C4.10
NODES	C4.20	SOUND	C4.24
NOT	C4.12	Sonido comandos de	
OP	C4.26	(LOGO)	C4.24
OR	C4.12	SS	C4.17
PADDLE	C4.23	ST	C4.19
PAL	C4.16	STAT	C3.8
PAUSE	C4.27	STOP	C4.26
PD	C4.8	SYSGEN	C3.11
PE	C4.18	SUBMIT	C3.12
Periféricos gestión de	C3.5	Sistema disco	C1.2
PIP	C3.4	Sistema formato	C5.6
PLIST	C4.21	Sistema gestión	C3.8
PO	C4.13	Sistema primitivos (LOGO)	C4.27
POTS	C4.13	Sistema propiedades	
PPROP	C4.21	(LOGO)	C4.28
PR	C4.14	Sistema variables (LOGO)	C4.28
Primitivos	C4.7 C4.27	TAPE	F4.2 C2.10
.PRM	C4.28	TAPE.IN	F4.3 C2.10
Procedimientos	C4.6 C4.13	TAPE.OUT	F4.3 C2.10
Propiedades comandos		TF	C4.20
de lista de	C4.21	THROW	C4.27
PU	C4.18	TO	C4.14
PX	C4.19	TOPLEVEL	C4.28

Transeúnte comandos	C3.5
TRUE	C4.28
TS	C4.15
Tortuga comandos	
para gráficos con	C4.17
TYPE (CP/M)	C3.5
TYPE (LOGO)	C4.15
USER	C2.10
Utensilios/Utilidades	C1.2
Variables	C4.5 C4.13 C4.28
WAIT	C4.26
WINDOW	C4.17
WORD	C4.10
WORDP	C4.10
XSUB	C3.12

**AMSTRAD**