

MICROHOBBY

## AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

Semanal

AÑO II N.º 24

160 Ptas.

(incluido I.V.A.)

Canarias, Ceuta y Melilla 165 ptas.

EXISTE LA LOGICA  
DENTRO DE TU  
ORDENADORAsí  
SUENA  
TU  
AMSTRADOrdena  
tus libros  
y autores  
preferidos  
con  
BibliotechHACIA LA  
SOLUCION OPTIMA  
CON DECISION MAKER

SOFTWARE

Los atletas de élite  
compiten en Hypersport  
y Supertest

# MICROHOBBY

# AMSTRAD

## Semanal

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

**LE OFRECE AHORA SUS PROGRAMAS YA GRABADOS PARA QUE VD. NO TENGA QUE TECLEARLOS. TOTALMENTE DESPROTEGIDOS CON EL OBJETO DE FACILITAR SU COPIA EN DISCO.**

**T**odos los programadores y aficionados a la microinformática, sabemos lo tedioso y propenso a errores que resulta el teclear un listado de un programa. Para facilitar tu labor al máximo y evitar que malgastes largas horas sobre el teclado de tu ordenador tratando de descifrar incomprensiblemente mensajes de error.

AMSTRAD SEMANAL te ofrece cada mes los programas publicados en los cuatro números correspondientes, en una cinta de cassette desprotegida, que te permitirá copiar los programas en disco y tener acceso a los listados para su estudio y posterior edición de rutinas.

#### Programas incluidos en la cinta número 1

Título	Rev. n.	Título	Rev. n.
TEXTDRAW	1	MAD-ADDER	3
EGGFLUTZ	2	HEXER	3
CODIGO SECRETO	2	CHARGEN	4
VENTANAS	2	PROGRAMACION	4
BORRACHOS	2		

#### Programas incluidos en la cinta número 2

Título	Rev. n.	Título	Rev. n.
GRAFICOS	6	INCOGNITON	7
MUSICA	6	MONITOR	5
TRON	6	ANALISIS	5B
ENSAMBLADOR	8	CEDIC	8
HEXERL	8	ANIMACION1	7
TOCKETT	8	ANIMACION2	8
PRIMEROS PASOS	7	SMILEY	7

#### Programas incluidos en la cinta número 3

Título	Rev. n.	Título	Rev. n.
ANALISIS	9-12	SPRITE12	11
FRUTAS	9	AMSCARD	11
MENUDISC	9	OREO	12
REV-01	10	EVENT121	12
REV-102	10	EVENT122	12
CUAIBORATA	10	FRUITIES	12
SPRITE11	11		



Por sólo 675 pts.  
(incluidos gastos de envío)



COMPATIBLES  
CON LOS MODELOS  
CPC-464, CPC-664  
y CPC-6128

Recíbelos en tu casa  
cómodamente enviándonos  
con la menor demora  
posible, el cupón que se  
encuentra en la última  
página de la revista

# AMSTRAD

## sumario

**Director Editorial**

José I. Gómez-Centurión

**Director Ejecutivo**

Victor Prieto

**Subdirector**

José María Díaz

**Redactora Jefe**

María García

**Diseño**

José Flores

**Colaboradores**

Francisco Portalo, Pedro Sudón

Miguel Sepúlveda,

Francisco Martín,

Jesús Alonso, Pedro S. Pérez

Amadio Gómez,

Juan J. Martínez,

David Sopuerta, Alberto Suárez,

Eduardo R. Velasco

**Secretaría Redacción**

Carmen Santamaría

**Fotografía**

Carlos Candel

Javier Martínez

**Portada**

M. Barco

**Ilustradores**

Javier Igual, J. Pons, F. L.

Frontón, J. Septén, Pejo, J. J.

Mora, Luigi Pérez

**Edita**

HOBBY PRESS S.A.

**Presidente**

María Andino

**Consejero Delegado**

José I. Gómez-Centurión

**Jefe de Publicidad**

Concha Gutiérrez

**Publicidad Barcelona**

José Galán Cortes

Tel: (93) 303 10 22/313 71 62

**Secretaría de Dirección**

Marisa Cogorro

**Suscripciones**

M.ª Rosa González

M.ª del Mar Calzada

**Redacción, Administración****y Publicidad**

La Granja, 39

Polígono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 480 HOPR

**Dto. Circulación**

Carlos Peropadre

**Distribución**

Coedis, S. A. Valencia, 245

Barcelona

**Imprime**

ROTEDIC, S. A. Ctra. de Irún.

Km. 12,450 (MADRID)

**Fotocomposición**

Novocomp, S.A.

Nicolás Morales, 38-40

**Fotomecánica**

GROF

Ezequiel Solana, 16

**Déposito Legal:**

M-28468-1985

**Derechos exclusivos**

de la revista

**COMPUTING with**

the AMSTRAD

Representante para Argentina, Chile,

Uruguay y Paraguay, Cia.

Americana de Ediciones, S.R.L. Sud

América 1.532. Tel.: 21 24 64. 1209

BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace

necesariamente solidaria de las

opiniones vertidas por sus

colaboradores en los artículos

firmados. Reservados todos los

derechos.

Se solicitará control OJD

Año II • Número 24 • 11 al 17 de Febrero de 1986

160 ptas. (incluido I.V.A.)

Precio Canarias, Ceuta y Melilla 155 + 10 ptas. sobretasa aérea.

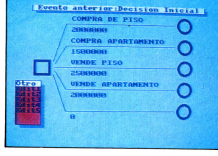
## 5 Primera plana

Nace LUCID, un nuevo lenguaje de programación. Lenguajes IA, están aquí.

## 6 Primeros pasos

Programar con lógica es una necesidad imprescindible para todo «artista» que se precie, y ello es algo mucho más amplio que el mero hecho de estructurar adecuadamente los programas.

El Amstrad Basic posee para ayudarnos unos operadores, llamados lógicos. Aprendamos cómo se utilizan.

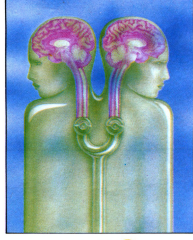


## 20 Banco de Pruebas

Decisión Maker es un interesante programa de gestión, pensado para asistir al hombre de negocios en la compleja tarea de escoger un camino entre varias alternativas posibles, basándose en estimaciones y criterios mensurables de las mismas.

## 24 ProgramAcción

¿Qué es el sonido en el Amstrad? ¿Cómo se produce? y, sobre todo, cómo podemos aprovechar las excelentes posibilidades sonoras del ordenador para nuestras propias creaciones.



## Serie Oro 12

Biblioteca es un extenso programa, con una presentación y sencillez de manejo fuera de lo común. Nos servirá para llevar nuestra biblioteca personal perfectamente actualizada.

## 16 Análisis

Presentamos un pequeño programa generador de ventanas, que graba y lee del disco los datos que definen una.

## Mr. Joystick 18

Toda la emoción de los Juegos Olímpicos con Hypersport y Supertest.



## Código Máquina 28

Tercera y última parte del estudio sobre las instrucciones de rotación y desplazamiento.

# ...descubre el N.º 3

ya está en  
tu quiosco

**AMSTRAD**

también disponible  
para

**SPECTRUM, PLUS, 128**

**COMMODORE 64**

## CARNIVAL

Si te gusta el tiro al blanco, con este programa podrás practicar sin necesidad de salir de casa. Si tienes buena puntería obtendrás disparos gratis.

## BLOCKER

Demuestra tu habilidad esquivando las paredes y a tus enemigos. Cuantos más destruyas, más aparecerán ante ti.

## SPACE

Al cargar este programa aparecerá ante ti un batallón de alienígenas. Tu misión es destruirlos, pero cuidado, su intención es eliminarte lo antes posible.

## HAUNTED

En este caso debes coger todos los puntos que aparecen en el laberinto. ¡Atención!, los fantasmas están enfadados e intentarán deshacerse de ti a toda costa.

## VAMPIRO

Es un programa en el cual pueden participar dos jugadores. La misión de cada uno será pintar las lápidas de un color distinto. La destrucción del enemigo significa la victoria.

## SPLIT

Es una rutina en código máquina, que te permitirá siete colores en pantalla en MODO 1, en el cual normalmente sólo se pueden utilizar cuatro.

Si no lo encontrara en su quiosco, solicítelo directamente a nuestra editorial.

**SINTAX, S.A.**

Paseo de la Castellana, 268.  
28046 Madrid. Tel. (91) 733 25 99

**795** pts.  
(Incluido IVA)

# YOUR

# COMPUTER

La mejor selección de programas de juegos y utilidades, publicados en la revista de mayor difusión de ordenadores de Europa. Ahora reproducidos en cassette, en auténtica exclusiva mundial.



## DOS PARA SPECTRUM



**Watford Electronics** vende en Inglaterra un sistema operativo en disco (*Disc Operating System, DOS*) para el Spectrum.

La capacidad de los drives oscila entre 200 Kbytes y 3.2 Megabytes.

El SP-DOS permite acomodar hasta 144 ficheros en el directorio del disco y la extensión de los ficheros de datos alcanza 895 Kbytes.

Otra característica útil del DOS estriba en que cualquier programa, Basic o código máquina, si se salva con autoejecución, podrá ser ejecutado desde el disco mediante la orden RUN.

Es decir, aquello a lo que los usuarios de **Amstrad** y otros ordenadores basados en disco están acostumbrados a hacer.

El SP-DOS ha sido diseñado para lograr una gran facilidad de uso, evitando en lo posible la aparición de comandos nuevos y adaptándose a los ya existentes del Sinclair Basic, especialmente los relacionados con el microdrive.

Watford regala, al adquirir su sistema, tras programas de utilidad: un proceso de textos, el Tasword, una base de datos, Masterfile y, cómo no, una hoja de cálculo, Omnicalc, completa la trilogía.

Los precios dependen de la capacidad de almacenamiento del disco, ofreciéndose 4 configuraciones: 200 K, 400 K, 800 K y 1 Mega.

## AMSTRAD EN LA RADIO



La informática cada vez causa un mayor impacto en nuestra sociedad, y los medios de comunicación tratan de hacerse eco de ello.

En este sentido, la cadena **COPE** y **Radio Miramar de Barcelona** van a poner en marcha, en Onda Media y

a partir del día 15 de febrero, un programa de dos horas de duración aproximadamente dedicado a los ordenadores **Amstrad**, Spectrum, Commodore y MSX.

La idea es crear un espacio informativo en constante contacto con todos los radioyentes, para potenciar al máximo la participación y aproximación entre la gente, usuarios o interesados en ordenadores.

El programa tendrá lugar todos los sábados de 5 a 7 de la tarde, inmediatamente antes del espacio dedicado al baloncesto, ofreciendo todo tipo de información acerca de hardware y software, a todos los niveles y del máximo interés.

## Primera plana

### LOS DISCOS LASER APRENDERAN A ESCRIBIR

**P**hilips y Verbatim, dos conocidas firmas en el campo de la informática, aseguran que para últimos del 87 producirán discos ópticos de lectura/escritura.

Hasta ahora, se han ido vendiendo CD (*Compact Disc*) para unos cuantos ordenadores personales y existen proyectos (*Amstrad* y *MSX*) de hacer la tecnología láser accesible a máquinas más pequeñas, a los «home computers».

Sin embargo, estos periféricos, que pueden almacenar cientos de Megabytes, presentan el espinoso problema de que no pueden ser **escritos** por el usuario; son de sólo lectura.

Verbatim proyecta un drive de lectura/escritura basado en una tecnología mezcla del magnetismo y la óptica, que será capaz de almacenar 150 Megabytes en sus primeras versiones y 600 cuando la técnica usada se refine un poco.

Tal vez lo más impresionante sea que el drive tendrá unas dimensiones análogas a las de un disco floppy de 3,5 pulgadas, esto es, un poco mayor que los diskettes empleados por **Amstrad**.

Otra idea que tal vez permita crear discos ópticos de lectura/escritura se basa en que ciertos materiales pueden existir en forma cristalina (*sus moléculas están ordenadas*) o amorfa (*sin orden*). Las características de reflexión de la luz en ambos estados son lo bastante diferentes para ser detectadas por un láser. Para cambiar el estado del material se le calienta ligeramente y luego, rápidamente, se enfría.

Los primeros prototipos de ambas tecnologías están ya funcionando.

# OPERADORES LOGICOS

Javier Igual

**Hasta aquí hemos sido capaces de comparar o relacionar un número con otro, ver si dos cadenas son iguales y alguna otra cosilla. Ahora vamos a intentar ver una serie de operadores que nos permitan unir varias de estas comparaciones y vigilen su cumplimiento.**



En muchas ocasiones nos ha sido necesario comparar un número con otro o una palabra con otra. Para ello hemos utilizado sentencias como:

IF condición THEN...

incluyendo en la condición cualquier operador «relacional» —que relaciona— de los que conocemos. Eran, por tanto, comparaciones del tipo:

A < B    A menor que B  
C = D    C igual que D

o

E < > F    E distinto de F

por ejemplo.

El resultado de estas comparaciones nos da un valor tal que nuestro **Amstrad** será capaz de saber si han sido correctas o falsas y actuar en consecuencia (siempre bien, por supuesto).

La instrucción:

IF A < B THEN PRINT «A es menor que B»

hará que salga en la pantalla el mensaje «A es menor que B» solamente cuando los valores que con anterioridad hayamos dado a A y B cumplan la condición A < B. En cualquier otro caso, nuestro ordenador quedaría mudo y no nos daría información alguna.

Pruébelo tecleando las siguientes órdenes:

A=3  
B=5

y si ahora introduce la anterior instrucción IF

IF A < B THEN PRINT «A es menor que B»

comprobará que es cierto lo que le dijimos anteriormente: saldrá el mensaje.

Sin embargo, si invierte los valores dados a A y B tecleando:

A=5  
B=3

y después repite al instrucción IF

IF A < B THEN PRINT «A es menor que B» no aparecerá ningún mensaje en la pantalla excepto el conocido READY.

Resumiendo, sólo se ejecutan las órdenes que siguen a la palabra THEN en el caso que se cumpla la condición A < B. Pruebe con varias diferentes mediante instrucciones IF... THEN de la forma en la que ya se nos está convirtiendo en familiar. Es el mejor modo de comprender cómo funcionan estas comparaciones y cuál son los resultados que producen. ¡Animo!

Pero, ¿qué ocurre si tenemos que analizar si se cumplen o no dos condiciones a la vez? O, ¿qué tipo de comparación tendríamos que hacer para que realicemos algo —sacar un mensaje en pantalla por ejemplo— cuando se cumplan una o varias condiciones de una serie de ellas y no lo realicemos sólo cuando no se cumpla ninguna?

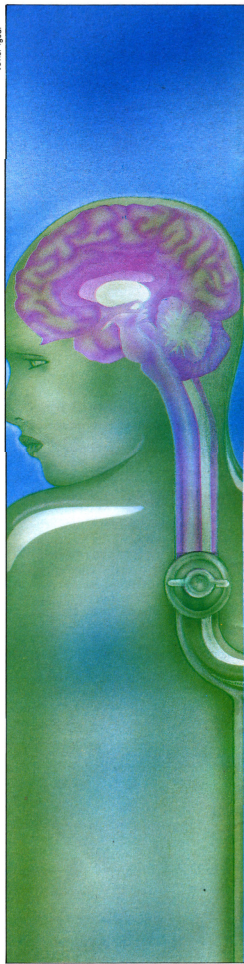
Para sacarnos de este atolladero nuestro **Amstrad** posee un tipo de operadores, llamados «operadores lógicos», que nos va a permitir actuar entre condiciones de modo que se puedan combinar varias dentro de una sola instrucción IF... THEN.

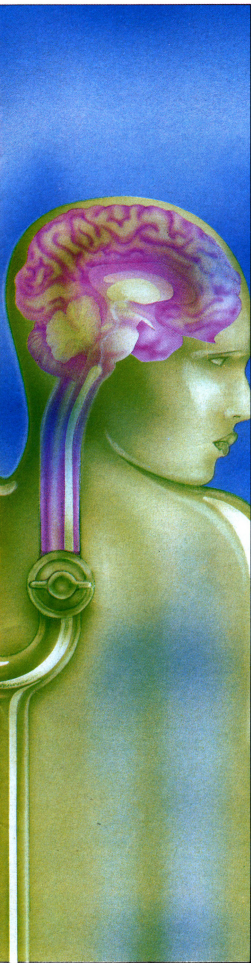
Intentemos averiguar si un número es mayor que 10 o no lo es. Con las herramientas que hasta ahora disponemos sólo nos sería posible comprobarlo de una forma semejante a la del programa 1. Echele un vistazo.

## Programa uno

El camino que sigue es sencillo. Primero nos pide un número que introduce en la variable «número» (línea 30).

A continuación investiga si el valor contenido en dicha variable es mayor que cero. Si es mayor que 0, ve si es mayor que 10 y en caso de ocurrir así saca un mensaje de VERDADERO.





En el caso de que no se cumplan ninguna de las dos condiciones, el mensaje final nos dirá que es FALSO que el número que hemos metido sea menor que 10 y mayor que 0.

Hemos utilizado, como habrá podido observar, dos instrucciones IF... THEN para analizar en cada una de ellas las dos condiciones que delimitan el rango de valores de la variable «número».

Ahora, como ya nos vamos convirtiendo en unos expertos programadores, puede parecernos que es un derroche de instrucciones hacerlo de esta manera. Vamos a intentar resumirlo y simplificarlo.

Ha llegado la hora de introducir en nuestro lenguaje informático el primero de los operadores lógicos: AND. ¿Qué hará?

Vamos a explicarlo. AND es un operador lógico que actúa sobre dos condiciones y su resultado será VERDADERO solamente cuando se cumplan ambas condiciones —cuando las dos son verdaderas—.

En el momento en que una de las dos condiciones sea falsa, este operador diría que no se cumple la condición general y por tanto su resultado sería FALSO.

Supongamos que dentro de un programa nos encontramos con una línea parecida a:  
IF número < 10 AND número > 0 THEN  
PRINT

«Valor correcto» ELSE PRINT  
«Valor incorrecto»

¿Qué hará?

Respuesta: Cuando el valor contenido en «número» sea a la vez mayor que 0 y menor que 10 nos saldrá en la pantalla un mensaje que nos dice que han cumplido las dos condiciones y que, por tanto, el número es correcto.

Si la variable «número» no tiene valor menor que 10 o contiene uno menor que 0, no se cumple una de las dos condiciones o no se verifican ninguna de las dos, el ordenador nos avisaría de que «número» no está comprendido entre los valores que hemos especificado.

Visto esto, hemos encontrado la solución a nuestro problema. Metemos una línea de este estilo en nuestro programa y punto. Si observamos el programa 2 verá la forma de hacerlo y la reducción de instrucciones que nos ha supuesto con relación al programa 1.

### Programa dos

Podrá comprobar que además de ser más corto, el programa es también bastante más claro. Hemos condensado el análisis del cumplimiento de las dos condiciones en una sola instrucción y así es más fácil valorar qué es lo que se está cumpliendo en realidad.

Esto es algo que siempre hemos de tener en cuenta a la hora de programar. Es muy conveniente reflejar las cosas de la misma forma que suceden en la realidad. Todo lo que sea añadir claridad a los programas hasta conseguir que sean más y más legibles ¡bienvenido!

## Primeros pasos

La tabla 1 nos resumiría todo lo anteriormente expuesto referente al operador AND. Es lo que llamamos su TABLA DE VERDAD.

AND	CONDICION1	CONDICION2	RESULTADO
	FALSO	FALSO	FALSO
	FALSO	CIERTO	FALSO
	CIERTO	FALSO	FALSO
	CIERTO	CIERTO	CIERTO

TABLA 1

Vamos ahora a intentar combinar dos condiciones de otra manera diferente. Supongamos que tenemos que ejecutar una determinada acción cuando se cumpla una condición de dos posibles o quizá sean verdaderas las dos a la vez, por ejemplo: los domingos o días de fiesta vamos a ir de excursión.

El programa 3 sería la forma tradicional de resolver semejante problema.

### Programa tres

Nos pedirá que introduzcamos si el día es laborable, domingo o festivo (línea 30) y lo mete en la variable «día».

En la línea 40 comparamos este valor con el literal «festivo». Si ambos son iguales, nuestro programa saltará a la línea 70 que es la que hace que aparezca en la pantalla el feliz mensaje que nos anuncia nuestro día de excursión.

Si esta primera comparación resulta que es falsa, continúa la ejecución de nuestro programa en la línea 50. En ella comparamos el valor de «día» con el literal «domingo». Si la comparación es verdadera, saltamos a la línea 70 que, como antes, nos comunicará la agradable noticia.

Si por el contrario, no se cumple esta segunda condición nos encontraríamos con que, al no ser el día elegido un domingo ni festivo, se nos estropearía la jornada de campo, conseguiríamos un mensaje desolador y terminaríamos la ejecución del programa.

En esta ocasión también hemos utilizado dos instrucciones IF... THEN, una para evaluar si se cumple o no cada una de las condiciones.

Como antes vamos a intentar dar algo más de claridad al programa a la vez que le hacemos más sencillo.

Nos toca el turno de presentar al segundo de los operadores lógicos más utilizados: OR.

Su forma de funcionamiento es muy sencilla. El operador OR —O en castellano— relaciona dos condiciones pero esta vez de un mo-

do diferente —claro— a como lo hacía el operador AND.

En este caso basta que una de las condiciones sea cierta para que la proposición que incluye el OR sea también verdadera.

Solamente será FALSA cuando ambas condiciones no se cumplan: las dos son FALSAS también.

La manera de reflejar esto en instrucciones BASIC sería semejante a:

IF condición1 OR condición2 THEN acción

Pero vamos a concretar. Apliquemos esta línea a nuestro programa campestre ya que es justamente la instrucción que estábamos buscando.

La primera condición será:

dia\$ = «festivo»

y la segunda:

dia\$ = «domingo»

La acción que debemos realizar será escribir el mensaje de EXCURSION si se cumple una de las dos condiciones.

En la línea 40 del programa 4 queda resumida toda la literatura anterior.

### Programa cuatro

Como antes primero pedimos que se teclée un tipo de día —laborable, domingo o festivo—. La línea 40 analiza si es domingo o festivo.

Si se cumple una de las dos condiciones, iremos de excursión (*atención al mensaje*), pero si no se cumple ninguna de las dos nos tocará quedarnos en casa.

Como ven es muy sencillo. Hemos vuelto a concretar el análisis de las dos condiciones en una sola instrucción IF... THEN, lo que de momento ya nos hace ahorrar instrucciones.

Les repetimos que quizá el englobar otra vez un grupo de condiciones en una sola hace que podamos seguir la ejecución del programa con mucha más claridad sobre el listado que si no fuera así. No olvide que es muy importante que todos nuestros programas, además de funcionar, puedan ser leídos sin problemas.

OR	CONDICION1	CONDICION2	RESULTADO
	FALSO	FALSO	FALSO
	FALSO	CIERTO	CIERTO
	CIERTO	FALSO	CIERTO
	CIERTO	CIERTO	CIERTO

Tabla II

Resumamos. El operador lógico AND nos da como condición VERDADERA el que «todas» las relaciones sobre las que actúa sean también VERDADERAS.

Para el operador OR basta con que una de las relaciones sea VERDADERA. ¿Ven la diferencia?

La Tabla de Verdad del operador OR nos la muestra la Tabla 2.

Puede ocurrir que en algún momento necesitemos analizar una condición de forma que ejecutemos una serie de instrucciones cuando dicha condición sea FALSA.

Por ejemplo, si tecléa:

A=2  
B=3

y  
PRINT NOT (A=B) THEN PRINT «HOLA»

observará que le aparecerá en la pantalla el mensaje «HOLA». ¿Qué educado!

¿Qué ha ocurrido? Vayamos analizando por partes.

Hemos dado el valor 2 a la variable «A» y 3 a la variable «B». Por tanto la condición:

A=B

no se cumplirá.

Sin embargo, en la pantalla nos ha aparecido la palabra «HOLA» como si la condición se hubiera cumplido. ¿Nos está engañando el Amstrad? No. Nuestro ordenador nos sigue permaneciendo fiel.

Todo ha sido debido a la palabra NOT que hemos puesto en la última instrucción.

Resultado que se trata de un nuevo operador lógico. A simple vista se aprecia que este operador solamente actúa sobre una única condición —aunque ésta pueda estar compuesta por varias que estén unidas por los operadores anteriormente vistos AND y OR—.

La misión de este operador es transformar, más bien invertir, el resultado de la evaluación de la condición. Cuando el resultado es VERDADERO, NOT actúa de forma que el resultado de la proposición sea FALSO y cuando el operador es FALSO, el resultado que nos da este operador es verdadero.

En nuestro caso, al encontramos la instrucción:

IF NOT (A=B) THEN PRINT «HOLA»

si el contenido de «A» es igual al de la variable «B», el operador NOT nos dará como resultado «NO VERDADERO» que es el equivalente a decir «FALSO». Por tanto el ordenador a no saludará —no aparecerá «HOLA» en la pantalla—.

Si, por el contrario, el contenido de «A» fuera distinto de el de «B» si que nos diría «HOLA» ya que NOT daría «NO FALSO», es decir, «VERDADERO».

Vamos a ver cómo funciona en la práctica este operador con el programa 5.

### Programa cinco

Mediante la línea 30 introducimos en la variable «revista\$» el nombre de la que estamos leyendo en este momento.

A continuación comparamos dicho nombre con el de la nuestra y aplicamos el operador NOT a dicha relación. Si no son iguales, la condición será FALSA y por tanto, el opera-

dor NOT hará que la proposición se vuelva verdadera sacando el mensaje correspondiente —línea 40—.

La Tabla de Verdad de este operador es muy sencilla. Ojea la tabla 3 y compruébalo.

Solamente nos queda ya conocer el último de los operadores lógicos con los que cuenta el Amstrad para asociar condiciones. En esta ocasión volvemos a tener dos relacionadas por medio del operador XOR.

¿En qué consiste este operador? Creemos necesario decirle que es un poco especial. Su función consiste en darnos como resultado VERDADERO cuando una de las condiciones se cumpla —sea VERDADERA— y la otra no —sea FALSA— o viceversa.

En el caso en que no se cumplan ninguna de las dos condiciones o por el contrario, se realicen ambas, este operador nos devuelve como resultado de la comparación el valor FALSO.

NOT	CONDICION	RESULTADO
	FALSO	CIERTO
	CIERTO	FALSO

Tabla III

Seguramente que viendo la ejecución del programa 6 entenderá mejor lo que le queremos decir.

### Programa seis

Que no se le olvide que en este caso, la aplicación del operador XOR nos dará VERDADERO solamente cuando la valoración de las dos condiciones sea diferente.

La Tabla 4 nos muestra los resultados que produce este operador al aplicarlo al cumplimiento o no de dos condiciones.

Por último, le diremos que todos estos operadores pueden utilizarse combinados dentro de una línea de programa. No es muy extraño encontrar una que presente el siguiente aspecto:

IF (A < B OR C > D) AND (NOT B = D)  
THEN PRINT «ESTO ES UN LIO»

XOR	CONDICION1	CONDICION2	RESULTADO
	FALSO	FALSO	FALSO
	FALSO	CIERTO	CIERTO
	CIERTO	FALSO	CIERTO
	CIERTO	CIERTO	FALSO

Tabla IV

Esperamos que a pesar del mensaje anterior, la utilización de estos operadores no resulte liosa para usted. Le proponemos como ejercicio que intente averiguar con paciencia las condiciones que han de cumplirse para que la proposición anterior sea VERDADERA. Coméntenos sus deducciones.

Y todo «LOGICAMENTE» llega a su fin. Ejercítense en el uso de todos estos operadores y le esperamos la semana próxima.

# Primeros pasos

```
10 REM PROGRAMA I
20 CLS
30 INPUT "TECLEA UN NUMERO ",numero
40 IF numero>0 THEN IF numero<10 TH
EN PRINT"EL NUMERO ES MAYOR QUE 0 Y
MENOR QUE 10": END
50 PRINT"EL NUMERO NO ESTA COMPREND
IDO ENTRE 0 Y 10"
```

---

```
10 REM PROGRAMA II
20 CLS
30 INPUT "TECLEA UN NUMERO ",numero
40 IF numero>0 AND numero<10 THEN P
RINT"EL NUMERO ES MAYOR QUE 0 Y MEN
OR QUE 10" ELSE PRINT"EL NUMERO NO
ESTA COMPRENDIDO ENTRE 0 Y 10"
```

---

```
10 REM PROGRAMA III
20 CLS
30 INPUT "TECLEA EL DIA(LABORABLE/F
ESTIVO/DOMINGO)",dia$
40 IF dia$="festivo" THEN GOTO 70
50 IF dia$="domingo" THEN GOTO 70
60 PRINT"NOS TOCA QUEDARNOS EN CASA
":END
70 PRINT"!QUE BELLO ES EL CAMPO!"
```

---

```
10 REM PROGRAMA IV
20 CLS
30 INPUT "TECLEA EL DIA(LABORABLE/F
ESTIVO/DOMINGO)",dia$
40 IF dia$="festivo" OR dia$="domin
go" THEN PRINT"!QUE BELLO ES EL CAM
PO!" ELSE PRINT"NOS TOCA QUEDARNOS
EN CASA"
```

```
10 REM PROGRAMA V
20 CLS
30 INPUT "QUE REVISTA ESTAS LEYENDO
?",revista$
40 IF NOT(revista$="microhobby") TH
EN PRINT "ES HORA DE CAMBIAR" ELSE
PRINT "!BUENA ELECCION!"
```

---

```
10 REM PROGRAMA VI
20 CLS
30 PRINT"TABLA DE VERDAD DE XOR"
40 PRINT
50 PRINT"A > B XOR A < C"
60 PRINT
70 INPUT "VALOR DE A: ",a
80 INPUT "VALOR DE B: ",b
90 INPUT "VALOR DE C: ",c
100 PRINT
110 IF a>b THEN PRINT"A>B ES VERDAD
ERO" ELSE PRINT"A>B ES FALSO"
120 IF a<c THEN PRINT"A<C ES VERDAD
ERO" ELSE PRINT"A<C ES FALSO"
130 PRINT
140 IF a>b XOR a<c THEN PRINT"EL RE
SULTADO DE A>B XOR A<C ES VERDADERO
" ELSE PRINT"EL RESULTADO DE A>B XO
R A<C ES FALSO"
```

## ESCUELA de INFORMATICA APLICADA

*"Mister Chip"*

CENTRO HOMOLOGADO Y COLABORADOR DEL INEM

- **CURSO de INICIACION**  
(6 meses)  
Diploma: PROGRAMADOR BASIC-1
- **INFORMATICA BASICA**  
(96 horas)  
Diploma: PROGRAMADOR EN BASIC
- **PROGRAMACION AVANZADA**  
(110 horas)  
Diploma: MASTER EN PROGRAMACION

Dirigido a mayores de 12 años.  
CIUDAD de los PERIODISTAS. Avda. Herrera Oria, 171 bajo  
Frente al Instituto N. Herrera Oria. Tels.: 201 64 09 - 201 93 85

TODAS LAS CLASES SON PRACTICAS  
CON ORDENADORES AMSTRAD O SPECTRUM

# COMENZAR POR EL PRINCIPIO

Programar es mucho más sencillo de lo que se piensa. Mediante una serie de conceptos clave y un poco de práctica, nuestras propias aplicaciones comenzarán a funcionar en nuestro AMSTRAD muy pronto.

# R

esulta obvio que el teclado de capital importancia, ya que a través suyo podemos comunicarnos con la máquina.

Obsérvese que existen teclas separadas para la letra «O» y el número «0»; éste siempre aparecerá cruzado por una línea, sea teclada desde el teclado principal o desde el número. Es muy importante recordar esta diferencia entre 0 y O; podemos garantizar que gran cantidad de «**inexplicables**» errores en sus primeros programas sucederán por confundirlos.

Volviendo al teclado principal, verá que además de las letras y números (teclas «alfanuméricas», en jerga de ordenadores), hay otras que destacan por un color diferente al resto. Algunas, como Caps Lock y Shift, nos serán familiares de los Máquins de escribir. Otras, como Ctrl y Esc, deberían ser algo nuevo.

Antes de continuar, vamos a introducir una pequeña convención que nos facilitará las cosas. Si es necesario pulsar la tecla SHIFT, en el texto se indicará [SHIFT]

Si, por el contrario, aparece SHIFT, sin corchetes, queremos indicar que tecléis primero la «S», luego la «H», etc.

La tecla [ENTER] es muy importante, y hay dos: una grande en el teclado principal y otra pequeña en el numérico. Se usa de manera similar a la de una máquina de escribir, para asegurarse de que la escritura continúa en la línea siguiente.

También, y esa es la diferencia, sirve para indicarle al **Amstrad** que hemos terminado de introducirle información y puede comenzar a procesarla.

No preocuparse por el mensaje de «error de sintaxis» (SYNTAX ERROR en inglés); el **AMSTRAD** no puede estropearse por escribir cosas en él. Podemos experimentar con toda libertad.

## No sabe, no contesta

Todo lo que el «**SYNTAX ERROR**» quiere decir es que el ordenador no comprende las palabras u órdenes que le hemos introducido (los **COMANDOS**). Necesita que se lo digamos en su propio lenguaje, BASIC. El Basic es muy similar al inglés, pero sólo usa unas cuantas palabras de ese idioma para hacer la vida un poco más fácil al **Amstrad**; como hemos comentado antes, esas palabras se denominan **COMANDOS** (en inglés, **KEYWORDS**, palabras-clave).

Por ejemplo, podemos decirle al ordenador que un programa Basic ha terminado con el comando END (en español, FIN). Tecleemos END y su equivalente FIN en nuestra lengua, pulsado [ENTER] después de cada palabra.

La máquina sólo acepta la primera, que ya tiene incorporada en el «**vocabulario**» **COM TODOS LOS COMANDOS BASIC**.

## También en mayúsculas

Los impacientes ya habrán descubierto que si se pulsa simultáneamente [SHIFT] y una letra, ésta aparece en mayúsculas. Intentado, por favor, si no lo habéis hecho antes. Es como esperábamos, igual que una máquina de escribir. Tampoco produce mucha sorpresa el hecho de que, si pulsamos a la vez [SHIFT] + 6, por ejemplo, obtenemos el signo «&». En general, aparecerá en pantalla el signo que se encuentra en la mitad superior de la tecla, si lo tiene; de lo contrario obtendremos una mayúscula. Análogamente, la tecla [CAPS LOCK], al pulsarla, actúa como si presionáramos [SHIFT] continuamente; inténtalo. Si volvemos a pulsarla, volvemos al estado normal, es decir, pulsando «d», por ejemplo, aparecerá en minúscula (en el lenguaje informático, estos comandos que *permutan entre dos estados diferentes suelen llamarse «conmutadores», traducción libre de la palabra inglesa «toggling»*).

## Oír es obedecer

Supongamos que queremos sumar 2 + 2, y llenos de alegría por haber encontrado un uso para nuestra máquina, le ordenamos que lo haga. El **Amstrad** lo hará a una velocidad de vértigo, con una precisión de docenas de decimales, pero pasará por alto el pequeño detalle de **COMUNICARNOS EL RESULTADO**. Recuerden, **SOLO** le hemos ordenado que sume dos números, ¡no que nos diga cuántos suman!

Para decirle al ordenador que escriba cosas en la pantalla, tenemos que usar el comando Basic PRINT.

Así, para VER el resultado de 2 + 2, debemos teclear: print 2+2 [ENTER]

Observad que no se necesita el signo «=» como en el caso de una calculadora. De la misma forma que el micro no permite sustituir el 0 por la O, tampoco se puede usar «x» o «>» como signo de multiplicar; sólo reconoce el signo «\*», como en el caso de:

PRINT 4\*3 [ENTER]

El signo «-», de restar, permanece inalterado; se encontrará en la misma tecla que

# Primeros repastos

«=». Para dividir, sin embargo, se utiliza una barra vertical «/» como en:

PRINT 12/4 [ENTER]

Por favor, probad ahora la siguiente secuencia de órdenes: PRINT 2+8-3  
PRINT 4\*8/2  
PRINT 4\*8+2  
PRINT 4\*(8+2)

Pensando un poco acerca de los resultados, descubriremos algo importante y muy de agradecer: el **Amstrad** interpreta las secuencias de operaciones en el mismo orden en el que lo aprendimos en la escuela, esto es, lo que va entre paréntesis, primero, multiplicación y división después y, por último, suma y resta.

Ahora teclear: print 2/3

print 1000\*1000\*1000

print 1/100/100/100/100/100

## Dentro de un orden

El punto a destacar es que el micro posee un límite de precisión. Por ejemplo, 2/3 no es exactamente 0.666666667. El error está por debajo del millón, pero existe y habrá algunos programas en los que tengamos que tenerlo en cuenta.

De forma similar, con números muy grandes o muy pequeños, el ordenador ahorra memoria representándonos en **NOTACION CIENTIFICA**, también llamada **FORMATO EXPONENCIAL**; así, 1000\*1000\*1000, en lugar de aparecer como: 1000000000

se muestra en pantalla como: 1 E9 que debe leerse como «**uno multiplicado por diez elevado a la novena potencia**», esto es, un 1 seguido de 9 ceros.

## No sólo números

Vamos a intentar ahora algo un poco más alegre: escribir frases en la pantalla. Si tecleamos: print «AMSTRAD» el ordenador escribirá **Amstrad** en la posición del cursor.

Obsérvese que la palabra que queremos escribir va entrecomillada y, sin embargo, las comillas no aparecen en pantalla. A la frase entrecomillada se la denomina «**cadena alfanumérica**», nada menos, o «**string literal**» en inglés. Cada palabra que el **Amstrad** lo trata como una serie de caracteres, uno detrás de otro y literal porque se escribe en pantalla lo que aparece entre comillas exclusivamente. Así: print «AMSTRAD»  
print « AMSTRAD»  
print « AMSTRAD»

produce una salida diferente porque el número de espacios que preceden a la palabra es distinto.

# ¡GUERRA al I.V.A.!

Suscríbete ahora a

**MICROHOBBY  
AMSTRAD**

y benefíciate de un

# 26%

# DESCUENTO

(Oferta válida sólo hasta el 28 de febrero de 1986)

Microhobby AMSTRAD **te ofrece** ahora una oportunidad excepcional. Hasta el 28 de febrero de 1986 podrás suscribirte a nuestra revista sin tener que pagar el recargo correspondiente al Impuesto sobre el Valor Añadido.

Microhobby AMSTRAD **lo abona por ti.** Ahora puedes recibir Microhobby AMSTRAD en tu domicilio durante todo un año por sólo 5.900 ptas., es decir, 2.100 ptas. menos de su valor real.

**¡APROVECHA ESTA OPORTUNIDAD!**

RECORTA O COPIA ESTE CUPON Y ENVIALO A HOBBY PRESS, S.A. APODO. DE CORREOS 232 ALCOBENDAS (MADRID)

Deseo suscribirme a **Microhobby AMSTRAD** durante un año (50 números) por **sólo 5.900 ptas.**, lo que me supone un ahorro de **2.100 pesetas.**

El primer número que deseo recibir es el \_\_\_\_\_

NOMBRE \_\_\_\_\_ EDAD \_\_\_\_\_

APELLIDOS \_\_\_\_\_

DOMICILIO \_\_\_\_\_

CIUDAD \_\_\_\_\_ PROVINCIA \_\_\_\_\_

C. POSTAL \_\_\_\_\_ TELEFONO \_\_\_\_\_ PROFESION \_\_\_\_\_

Marco con una (x) en el casillero correspondiente la forma de pago que más me conviene.

Talón bancario adjunto a nombre de HOBBY PRESS, S. A.

Giro Postal a nombre de HOBBY PRESS, S. A., N.º \_\_\_\_\_

Contra reembolso del primer envío

VISA N.º \_\_\_\_\_

Fecha de caducidad de la tarjeta \_\_\_\_\_

Domiciliación bancaria (50 NUMEROS MAS 1 DF REGALO)

Banco \_\_\_\_\_ Sucursal \_\_\_\_\_ Localidad \_\_\_\_\_

N.º de cuenta \_\_\_\_\_ Firma y fecha \_\_\_\_\_

Si lo prefiere puede  
suscribirse por teléfono:  
(91) 654 28 98

# BIBLIOTECH

**Este programa está pensado para llevar a cabo eficientemente la gestión de una biblioteca, es decir, se considera a los libros y a los datos relacionados con ellos como los elementos integrantes de una base de datos, en la cual se pueden hacer las cosas usuales de este tipo de programas y que posteriormente el autor explica con detalle.**

# B

ibliotech, aparte de enseñarnos cómo diseñar una base de datos, y cómo aprovechar las capacidades gráficas del **Amstrad** para lograr una soberbia presentación, nos permitirá llevar perfectamente una clasificación actualizada y ordenada de todos los libros que tengamos en nuestra casa.

**Ordenador:** CPC-464 (monitor fósforo verde).  
**Memoria de masa externa:** cassette.  
**Tipo:** utilidad.  
**Lenguaje:** Basic.  
**Longitud:** 14237 bytes (7 bloques).  
**Nombre del programa:** GESTION DE UNA BIBLIOTECA.  
**Capacidad:** 350 libros (autor, título, tema y comentario). 200 autores.  
(Esta capacidad puede aumentarse disminuyendo al mínimo el número y longitud del comentario dejando esta opción para recordar préstamos, premios, etc. Particularmente poseo un fichero con 576 libros, 189 autores con una memoria libre de cerca de 8 K.)

Con monitor en color: En la presentación del programa hay una rutina que hace el efecto de **fundido** a la inversa variando el valor de INK 1,x desde x:1 hasta x:26. Este efecto es visible con el monitor en fósforo verde, pero con el de color no lo es. Para ello, se debe modificar lo siguiente.

- De la línea 25 quitar INK 1,1.
- Borrar la línea 1360.
- La letra eñe está redefinida en la tecla «». ».

**Programa realizado por:**  
**José Antonio Serrano Bergali**  
**(Sevilla)**



## Instrucciones de manejo

Una vez cargado el programa y después de hacer «run» aparecen dos opciones:

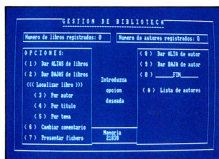
- A) Cargar ficheros:** con esta opción se cargan ficheros ya salvados anteriormente y almacenados en cinta. Cada fichero está dividido en dos: Fichero Autores y Fichero Máster.
- B) Iniciar:** si se pretende iniciar un nuevo fichero.

Una vez acabado de cargar los ficheros o de iniciar uno nuevo, aparece el menú del programa. En la zona superior de la pantalla aparecen los dos recuadros con el número de libros y de autores registrados. Esta información está presente en la mayor parte del programa.

En la zona inferior central aparece la memoria libre existente hasta el momento. Aunque no es totalmente exacta da una idea muy aproximada de ello.

Las opciones del menú son las siguientes:

- 1) Dar alta de libro:** con esta opción es posible introducir:



— **Autor:** para que el programa lo acepte éste ha debido de ser dado de alta anteriormente utilizando la opción B.

— **Título:** es indiferente si se introduce en minúsculas o mayúsculas ya que el programa lo convierte en mayúsculas y así lo almacenará. Esto es para que a la hora de búsqueda no existan problemas.

— **Tema:** el programa tiene almacenado 83 temas más una opción de «sin clasificar». Para verlos se pulsa la tecla espaciadora y van apareciendo los temas en bloques de 21. Cuando lo haya localizado pulse (DEL) e introduzca el número del tema.

— **Comentario:** se puede introducir un total de 58 caracteres. Es recomendable reducirlos al mínimo ya que de ello depende en gran medida la capacidad final del fichero.

Una vez introducida la ficha nos da la posibilidad de introducir otra ficha o volver al menú.

- 2) Dar baja de libro:** nos permite borrar de la memoria algún libro del fichero mediante su título.

- 3) Localizar libro por autor:** nos presenta todos los libros de un determinado autor junto con su tema y el número de ellos.

- 4) Localizar libro por título:** podemos localizar un libro por su título completo o libros que contengan una determinada cadena.

- 5) Localizar libro por tema:** nos presenta los libros cuyo tema coincida con el introducido por nosotros.

- 6) Cambiar comentario:** nos permite modificar el comentario de algún libro.

## LISTA DE VARIABLES

1mx:	Número máximo de libros-1
amx	Número máximo de autores
l	Número de libros introducidos
f	Número de autores introducidos
fl	Verifica si se ha variado el Fichero Máster
fa	Verifica si se ha variado el Fichero Autores
c	Contador de libros por autor
d	Contador de libros por tema
t	Tono del sonido
i	Verifica si se ha encontrado algún libro del autor requerido
t\$ (lmx)	Título del libro
a (lmx)	Número del autor
t (lmx)	Número del tema
c\$ (lmx)	Comentario
a\$ (amx)	Nombre del autor
n (amx)	Número de cada autor
b\$ (83)	Tamas
IS (0) - 1\$ (1)	Cadenas para crear el rótulo gigante
- 1\$ (2)	

## Serie Oro

cuenta que primero hay que dar de baja los libros de ese autor.

**O) Fin:** cuando queremos finalizar pulsamos esta opción y el programa pasa a grabar en cinta los ficheros que hayan sido modificados. Esto lo detecta el programa por sí solo.

**A) Lista de autores:** nos presenta los autores por orden alfabético.



**7) Presentación del fichero:** nos presenta por orden de introducción el título, autor, tema y comentario, si lo hubiese. También nos presenta el número del libro. Si queremos parar pulsamos (*espacio*). Si pulsamos (CLR) podemos saltarnos parte del fichero. Para retroceder pulsamos (TAB). Para volver al menú (DEL).

**8) Dar alta de autor:** podemos añadir un autor más a la lista de autores. El programa lo ordena alfabéticamente.

**9) Dar baja de autor:** nos da la opción de suprimir un autor de la lista de autores. Hay que tener en

## VARIABLES DE USO GENERAL

ss, tt, a\$, t\$, c\$, r, s, x, y, q, p, k, r\$

### Rectificación Serie Oro del número 20

En el programa **Amster** de Serie Oro de AMSTRAD SEMANAL se deslizaron dos errores de impresión, debido a que el autor del programa colocó en dos líneas (1270 y 1300) un carácter de control no reconocido por nuestras impresoras.

Las dos líneas, completas, se ofrecen más abajo.

El símbolo «%» que aparece en ellas, debe sustituirse por el que se obtiene pulsando simultáneamente CTRL + x.

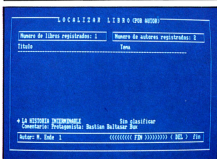
Lamentamos las molestias que esto haya podido causarles.

```
1270 LOCATE 6,2:PEN 12:PRINT"9-INFO
RNE"9*:LOCATE 2,3:PEN 2:PRINT"9-FIN
DEL COMBATE"9*:LOCATE 2,6:PEN 1
1:PRINT"-CAUSA":IF ZONA=21 THEN 13
00
```

```
1300 LOCATE 2,10:PEN 2:PRINT"9-LLEG
ASTE A LA ZONA"SPC(9)"20"SPC(10)"9"
:LOCATE 2,21:PEN 4:PRINT"Quieres su
gar de nuevo?":GOTO 1320
```

## TABLA DE SUBROUTINAS

10-40	Se dimensionan matrices. Opción de cargar ficheros o iniciar
100-165	Introducción de una nueva ficha (autor, título, tema y comentario)
200-245	Localización del autor. Si existe se fija su número
250-285	Dar baja de autor.
300-345	Localizar libros de un determinado autor
350-390	Localizar libros por título o cadena
400-445	Localizar libros por temas
450-488	Cambiar comentario
500-670	Carga de los ficheros
700-790	Grabación de los ficheros
850-890	Presentación del fichero
900-948	Dar alta de autor
950-985	Dar baja de autor
1000-1118	Menú
1120-1155	Presentación autores
1220-1299	Fin
1300-1400	Redefine la eñe. Dibuja el rótulo gigante. Hace el «fundido» de la presentación. Carga los 84 temas.
1500-1600	Subrutinas de recuadros, ventanas, presentación de temas y sonido
2000-2030	Datos







# Análisis

# GENERADOR DE VENTANAS

**Al ver el programa de Análisis de esta semana, tal vez el título del mismo os parezca algo pretencioso, ya que el programa es muy sencillo.**

**Sin embargo, muestra un método de trabajo de gran utilidad, que sorprendentemente no se usa demasiado; parece que tuviera mala prensa o fuera desconocido.**

# E

El tema consiste en que más de un programa pueda utilizar los mismos datos; el procedimiento es muy simple: se graban los datos en un fichero de disco y luego ese mismo programa u otro los lee y los aprovecha.

Lo hemos aplicado al diseño de ventanas, en el sentido de crear en el disco un fichero con todos los parámetros que necesita una ventana, canal incluido, y luego volverlo a leer y reutilizarlo. Estoy seguro que vuestra vivaz imaginación os sugerirá muy pronto otras formas de usar esta idea.

Ahí va el programa:

**20-90** Bucle principal que maneja a todas las subrutinas del programa. Como de costumbre, nos confesamos adictos del bucle principal **forever**.

**120-200** Entrada de datos mediante sentencias INPUT. Observad que las variables que definen la ventana se piden en el mismo orden que el comando WINDOW requiere, excepto el canal.

**210-250** Esta subrutina muestra simplemente la ventana; se ha colocado aparte porque el programa la usa dos veces, la segunda para «demostrar» que los datos se han leído correctamente del disco.

**260-350** Se pregunta el nombre de fichero, se abre para escritura (línea 310), se escriben las variables en el fichero (línea 320) y se cierra con CLOSEOUT. Ya que la ventana existe, se ha aprovechado para pedir los datos.

**360-440** Prácticamente como la anterior, sólo que aquí los leemos del fichero y se vuelve a crear la ventana llamando a la rutina que comienza en 220.



**P**asa que los datos no miden el trabajo duro. M.H. AMSTRAD lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un coste mensual, reducido.

# Juegos & ESTRATEGIA

Juegos de Simulación Estratégica  
para poner a prueba su inteligencia  
y vivir la emoción de situaciones reales!

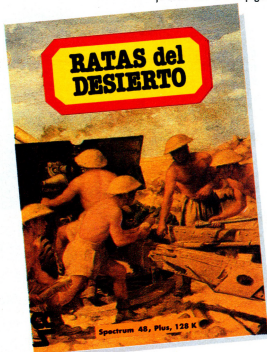
¿Se atreve Ud. a  
dirigir la Campaña  
del Desierto y  
derrotar a Rommel  
antes de lo que lo  
hizo el General  
Montgomery en el  
Alamein?

Un WAR GAME de estrategia que reúne todos  
los condimentos necesarios para hacer de él un auténtico  
«plato fuerte», inteligencia, emoción, sorpresa y realismo.

## RATAS del DESIERTO

### Operación Norte de Africa

¡Desde uno a tres jugadores!



Disponible para  
Spectrum y  
Amstrad

**HOBBY PRESS**  
Para gente inquieta.



Recorte o copie este cupón y envíelo a Hobby Press, S. A. Apdo. de Correos 54.095 Madrid.  
Deseo recibir en mi domicilio, sin gastos de envío alguno por mi parte, la cinta **RATAS DEL DESIERTO** al  
precio de 1.125 ptas. (IVA incluido).  
Nombre .....  
Dirección .....  
Localidad .....  
Código .....  
El medio que elijo es para:  
 Amstrad  Spectrum  
 Giro postal n.º .....  
 Press S. A.  Tarjeta Visa n.º .....  
Fecha de caducidad de la tarjeta .....  
Provincia .....  
Teléfono .....  
Echad .....  
Fecha y firma: .....

# HYPER SPORTS

El deporte de alta competición solamente está hecho para aquellos que, compitiendo en varias pruebas, son capaces de obtener los mejores resultados.

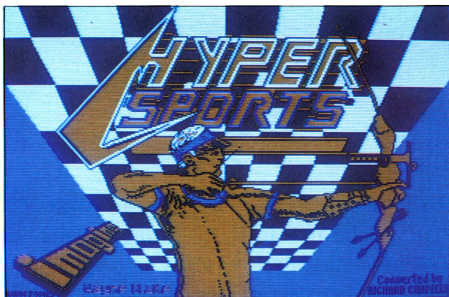
# H

yper Sports es un programa que vio la luz por primera vez en las máquinas de billares. Con él se puede decir que nació la era de los juegos deportivos.

En su versión para mega-ordenadores, incluía las siguientes pruebas: cien metros lisos, salto de longitud, cien metros vallas, lanzamiento de jabalina, lanzamiento de martillo y salto de altura.

Todas ellas, realizadas con un colorido y una ambientación fantásticas, la música y los efectos sonoros son excelentes, con unos gráficos que reproducen exactamente el desarrollo de cada prueba deportiva, consiguiendo una animación alucinante.

Este programa, al ser adaptado para el Amstrad, salió con el nombre de Decathlon, y, francamente, cualquier parecido de la versión me-



ga-arcade (máquina de billares) con la versión para nuestro ordenador, era pura coincidencia.

Nuestro atleta tenía una constitución física más parecida a la de un hombre y el movimiento era bastante deficiente. Defectos éstos que no se pueden achacar exclusivamente a la falta de memoria.

La continuación del Hyper Sports en los mega-arcade, fue el Hyper Medallist, otro superprograma con gráficos y animación aún mejores que los de su predecesor.

En esta ocasión, la competición se desarrollaba sobre los siguientes deportes: natación, tiro al plato, salto de potro, tiro con arco, triple salto, levantamiento de pesas y salto con pértiga.

Su versión en programa para Amstrad, es el Hyper Sports, en él se han incluido las seis primeras pruebas.

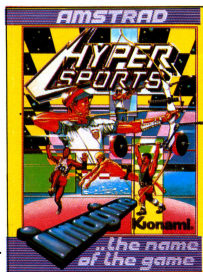
La competición es por eliminatorias, de forma que no pasamos a la siguiente prueba hasta habernos clasificado en la que estamos compitiendo actualmente.

El control del juego solamente puede realizarse por teclado, y nuestro nombre quedará impreso en el marcador del estadio en caso de que superemos alguna de las tres mejores marcas de todos los tiempos.

Las claves para lograr el oro: moral de triunfo, habilidad y un duro entrenamiento a base de joystick.



Compatible: CPC/464, CPC/664 y CPC/6128



# SUPER TEST

## Mr. Joystick

**Agotadas las pruebas de estadio olímpico, los programadores han tenido que recurrir a otros deportes menos conocidos y pruebas de habilidad.**



Tras el Decathlon e

Hyper Sports, la competición sigue con Super-Test.

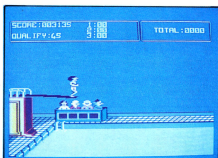
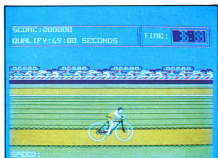
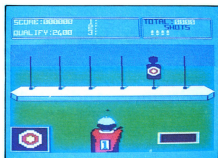
En esta ocasión la competición deportiva se desarrolla en dos días, cada uno de los cuales se encuentra en una cara de la cinta.

Como su antecesor, es obligado el uso del joystick, pues no existe opción de teclado.

La competición comienza con el tiro de pistola.

Ante nosotros se encuentran las ocho siluetas que constituyen nuestro blanco, de las cuales una se vuelve hacia nosotros, dándonos tres segundos para acertarle: una prueba de reflejos y puntería combinadas.

**Ciclismo:** a base de firme pedaleo y voluntad inquebrantable, debemos recorrer los tres kilómetros que nos separan de la meta. Salto de trampolín: tomando impulso en la tabla, volamos hacia las nubes y comenzamos a girar, hasta caer completamente verticales en el agua.



**Slalom gigante:** deslizándonos a la máxima velocidad, por la inclinada pendiente, sorteando las banderas que marcan el recorrido.

Nuestra lucha contra el reloj determina si podemos pasar a la siguiente prueba o continuar en ésta.

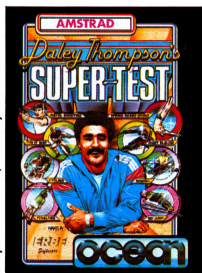
**Remo:** en nuestra embarcación de banco móvil, luchamos contra nuestro adversario deslizándonos por las sosegadas aguas del lago artificial que sirve de escenario a las pruebas de remo.

**Lanzamiento de penaltis:** una prueba de habilidad que se sale del marco olímpico; tomamos impulso, corremos con energía hacia el balón y chutamos con todas nuestras fuerzas hacia la portería. El guardameta se encargará de detener nuestro tiro; solamente los lanzamientos ajustados a los palos son gol.

**Salto de esquí:** nos dejamos caer por la pendiente de la rampa, acumulando la máxima energía posible, y, al llegar al borde de la misma, tiramos del cuerpo hacia arriba y comienza el vuelo que nos dará la clasificación.

**Cuerda:** solos ante nuestro contrincante nos agarramos al suelo desesperadamente intentando desplazarle, si le hacemos cruzar la raya.

Si después de tanto derroche de energía física, no te encuentras exhausto, el mismísimo Daley Thompson te envidiará.



Compatible: CPC/664, CPC/664 y CPC/6128



# DECISION ASISTIDA POR ORDENADOR

**Decision Maker es un programa de ayuda a la toma de decisiones, esto es, el Amstrad, asistido en este caso por usted mediante una serie de datos, es capaz de indicarle el camino más lógico (¡jojo!, no el más correcto necesariamente) para cumplir un objetivo basado en determinadas premisas. El programa está bien documentado, y lleva en el disco un programa tutor para el no iniciado en la teoría del Análisis de Decisión.**



Esta metodología es un poco abstrusa, y requiere algo de paciencia por parte del usuario, para acostumbrarse a esa forma de obrar, y a esa terminología.

En este sentido, el manual y el programa tutor será de gran ayuda.

Hemos pensado que, esta vez, además de describir el programa con detalle, merecería la pena incluir una breve introducción acerca del Análisis de Decisión, y luego, el resto. Así se ha hecho, y esperamos que resulte comprensible.

## Análisis de Decisión

Las técnicas presentadas en este **Decision Maker** pueden usarse para efectuar un análisis explícito de una situación concreta y revelar la estrategia apropiada que ha de adoptarse. Tu papel como «tomador de la decisión» se simplifica porque puedes ver cómo las diversas acciones tomadas pueden generar diferentes resultados en el futuro. Es esta estructuración del problema de las decisiones, combinada con el instrumento para medir el valor de la decisión, lo que constituye el atributo clave de Análisis de Decisión. Puedes desglosar el problema —rompiéndolo en la serie de partes que lo componen—, y ese proceso te llevará a una perspectiva más clara de las opciones, permitiéndote tener un control mayor sobre el modelo de los futuros eventos.

Es importante apreciar que no todos los problemas en la toma de decisiones pueden resolverse fácilmente por este método. Otras técnicas tales como la Programación Lineal, el Flujo de Caja Neto, el Análisis del Camino Crítico y similares, son más apropiadas para algunas situaciones. Para usar el Análisis de Decisión debe ser posible desglosar el problema

en un número finito de elementos de la siguiente manera:

1. Hay una decisión inicial de tomar, y hay un número limitado de alternativas claramente definidas entre las que escoger.

2. Para cada decisión, hay un número limitado de posibles resultados alternativos.

3. Cada resultado pudiera llevarnos a otra decisión con otra serie de posibles resultados alternativos de la misma, y así sucesivamente. La técnica del Análisis de Decisiones implica el trazado de un diagrama del proceso de una forma simbólica. De hecho el proceso real de construir el diagrama, conocido como Arbol de Decisión, ayudará a que una decisión muy compleja parezca algo más sencilla.

Una vez que se ha desglosado el problema de esta manera, necesitamos un medio de juzgar el valor de la decisión. En el mundo de la gestión, habitualmente se mide por el coste o por el beneficio, pero en otras circunstancias, el valor puede estimarse por ejemplo en términos del número de vidas salvadas, o el número de puestos de trabajo creados, o en alguna otra cantidad no monetaria. Debemos, por tanto, ser capaces de estimar el **valor** de cada decisión y de cada resultado, de manera que pueda elegirse la decisión inicial que nos conduzca al valor más alto posible.

La aplicación de las técnicas **Decision Maker** a un amplio surtido de situaciones es un instrumento potente de gestión por diversas razones:

1. Te compete a ti que eres la persona que toma la decisión, a reconocer la estructura y las relaciones entre los diversos elementos implicados.

2. Debes sistemáticamente evaluar todas las acciones posibles y los resultados, y también eso te exige que revises concienzudamente cada aspecto de la decisión.

3. Puedes modificar los costes, los valores y los riesgos de las diversas acciones implicadas y comprobarlos para ver cuántas cosas necesitan cambiarse para hacerte alterar tu decisión inicial.

## El proceso en seis pasos

El Análisis de Decisión puede considerarse como un proceso en seis etapas, que llevan desde el concepto general de la naturaleza de las decisiones hasta una solución específica. Los pasos pueden definirse de la forma siguiente:

1. **ESTRUCTURA:** Analiza y clasifica las diversas decisiones que han de tomarse, en qué orden surgen, y cuáles son los eventos aleatorios que pueden ocurrir entre ellas. Luego las diversas decisiones y resultados se trazan de una forma simbólica en la que se denomina un «Arbol de Decisión».

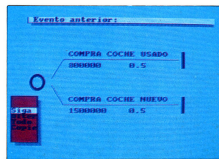
2. **EVALUACION:** Calcular los costes y valores asociados a todos los eventos y casos circunstanciales involucrados en la decisión.

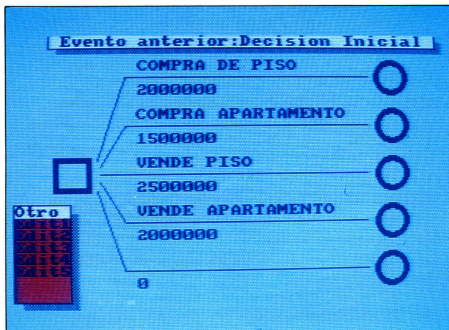
3. **PROBABILIDADES:** Estimar el grado de posibilidad de que ocurran cada uno de los eventos aleatorios.

4. **ANÁLISIS REGRESIVO:** Usar las reglas para el análisis de los árboles de decisión con el propósito de calcular la decisión. Esta regresión es la clave del proceso.

5. **ANÁLISIS DEL RIESGO:** Comprueba la solución calculada para asegurar que pueden tolerarse todos los posibles resultados de la decisión.

6. **ANÁLISIS DE SENSIBILIDAD:** Halla cuánto tienes que cambiar las hipótesis para que cambie la solución.





### Descripción del programa Inicio

Cuando el programa esté en marcha, mostrará el menú de opciones iniciales:

COMIENCE UN NUEVO ARBOL  
CARGUE UN ARBOL ANTERIOR  
FINALICE ESTE PROGRAMA

### Construcción de un Arbol

Cuando se elige la opción de Comenzar un Nuevo Arbol, la pantalla se queda en blanco y el programa trazará el primer nodo decisivo en la parte izquierda. En la parte superior se ven los mensajes «Evento Anterior; Decisión Inicial». En pantallas sucesivas, la función de esta línea se hará mucho más comprensible, pero inicialmente, actúa como un recordatorio de que es el primer nodo del árbol.

En la parte inferior de la pantalla hay un mensaje que solicita que introduzcas la cantidad de acciones diferentes que van a emerger de este nodo. Dicha cantidad está limitada a cinco en virtud del tamaño de la pantalla, y eso debiera ser lo suficiente en la mayoría de las circunstancias.

Las preguntas se repiten sucesivamente para cada una de las ramas. Cuando se hayan definido todas las ramas, aparece un menú en la parte inferior izquierda de la pantalla. El primer menú ofrece cuatro opciones:

**Siga:** El ordenador pasará al siguiente nodo, buscando la información requerida para completar el árbol.

**Altere:** Apela a la rutina de «edición» de manera que puedas revisar y cambiar algo de lo que veas expuesto en pantalla.

**Todo:** Visualiza toda la imagen del árbol completado hasta ese momento.

**Copie:** Si tienes una impresora Amstrad DMPI, esta opción hará que se imprima en papel una copia de lo que hay en pantalla, y luego continúe con el programa.

### Edición

Si eliges en el menú la opción de Alterar, en cualquier momento, aparecerá un menú diferente de opciones disponibles:

**Otra:** Hace que regreses al menú de Sigal/Altere/Copie.

**Edite:** Hay una línea de edición para cada rama que aparece en pantalla. Eligiendo una cualquiera haremos que entre en acción el menú de edición y podamos variar esa línea concreta.

**Sume:** Esta opción sólo aparecerá si hay menos de 5 ramas en el nodo. Si la eliges, la pantalla volverá a ser trazada, pero no con una rama adicional, y te pedirá los datos pertinentes de esa nueva rama.

El menú de edición completo aparecerá, cuando se haya elegido editar una de las líneas, en la forma siguiente:

**Otro:** Hace que regreses al menú previo. Si no deseas editar la línea que está «activa» en ese momento (**mostrada en la parte inferior de la pantalla**), usa esta opción para regresar al menú anterior y poder editar otra línea.

**Etique:** Te permite volver a introducir la descripción de la rama.

**Valor:** Para volver a introducir el coste o valor.

**Proba:** Para volver a introducir la probabilidad.

**Nodo:** Para cambiar la clase de nodo de la que emerge una rama.

# Banco de pruebas

**Borre:** Para suprimir completamente una rama. Si es de un nodo aleatorio, las probabilidades de los otros resultados tendrán que volver a ser re-inscritas.

### Visualización del Arbol

Al concluir la introducción de datos, puedes aprovechar la opción incluido en el menú de exponer **Todo**. Eso trazará en pantalla un diagrama esbozando la parte del árbol de decisión que hayas definido hasta ese momento. La pantalla incluirá tres posibilidades:

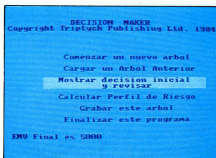
EL PROPIO DIAGRAMA DEL ARBOL  
UN CURSOR  
UN MENU DE OPCIONES

El diagrama por sí mismo es una representación directa del árbol, con las tres clases diferentes de nodos: decisorios, aleatorios y terminales, distinguiéndose los unos de los otros.

### Búsqueda de la solución

Cuando el árbol ha sido completamente definido, al elegir en cualquiera de los menús la opción de seguir hará que el ordenador proceda a calcular el **valor medio esperado** en el nodo inicial del árbol y que regrese al menú principal. Se te ofrecerán las siguientes alternativas:

COMENZAR UN NUEVO ARBOL  
CARGAR UN ARBOL ANTERIOR  
MOSTRAR DECISION INICIAL  
Y REVISAR  
CALCULAR PERFIL DE RIESGO  
GRABAR ESTE ARBOL  
FINALIZAR ESTE PROGRAMA



Puedes revisar cada nodo en detalle si lo deseas, eligiendo la opción de Revisar y cuando ya estés preparado, puedes regresar al menú principal mediante la opción de Seguir. Si consideras que debes hacer alteraciones a los datos del árbol, puedes hacerlo mediante las opciones de Revisar y Juntar, pero en este caso la opción de Seguir sólo te volverá al menú principal si el árbol queda completamente definido, y en ese caso además volverá a calcular el valor medio esperado.

Cuando estés de nuevo ante el menú principal, puedes elegir la opción de Calcular el Perfil de Riesgo. Las probabilidades y los costes o valores para todos los posibles resultados de los nodos aleatorios, aparecerán entonces en pantalla. Puedes regresar al menú principal o sacar una copia de este **Perfil de Riesgo** mediante la impresora, si dispones de ella.

**Valor de la información de Muestreo/Sondeo**

Finalmente, puede obtenerse del árbol una información adicional. Siempre y cuando haya sido calculado el valor medio esperado, aparecerá una opción adicional en el menú denominada **EVSI (valor esperado según información de «sondeo»)**. Eligiendo esta opción hará que se calcule el valor esperado según la nueva información adicional correspondiente a ese nuevo nodo aleatorio. El ordenador te hará dos preguntas:

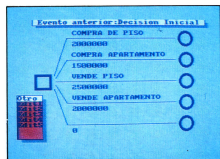
**¿Fiabilidad del sondeo?**

— Tecllea la fiabilidad que tú estimes como un número de 0 a 1. El valor 1 representará la **información perfecta**, y si lo das lo que obtiene realmente es el valor **EVPI** (valor esperado según información perfecta).

**¿Valor de la opción cero?**

— **¿Cuál es el coste de evitar este riesgo?** i.e. si se predice un resultado con un valor medio esperado negativo, **¿cuánto costará hacer algo más, o no hacer nada en lugar de eso?** Típicamente, ante esta pregunta se puede contestar que **0**.

Luego se calculará el valor esperado según información de sondeo, y aparecerá en la parte inferior de la pantalla. Eligiendo esta opción **EVSI** otra vez haremos que vuelva a calcularlo según otros valores, o alternatively, puedes continuar con tu inspección del árbol, usando la opción de **Otro** menú.



**Manipulaciones adicionales**

Un parámetro importante en todo proceso de Análisis de Decisión es el del **«Análisis de la Sensibilidad»** de la solución alcanzada; es decir, comprobar cómo se comportan las decisiones en cuanto a ajustes incrementales en las diversas hipótesis —particularmente en cuanto a los costes, los valores y las probabilidades—. El ordenador te permite primeramente que introduzcas tus **«mejores estimaciones»**, y luego regresa al árbol para cambiar sucesivamente las variables y observar cuál es la que tiene un mayor impacto sobre la solución.

Fundamentalmente, sólo hay dos restricciones en el programa: no puedes tener más de 150 nodos en el árbol, y cada nodo decisivo o aleatorio no puede dar lugar a más de 5 ramas. Si realmente tropiezas con una u otra de estas restricciones, debes en primer lugar preguntarte a ti mismo si no estás haciendo las cosas demasiado complicadas. Si estás convencido que tu enfoque es el correcto, entonces puede hacer algo de lo siguiente:

- Si tienes más de 150 nodos: Entonces desglosa el árbol en dos o más sub-árboles, cada uno con menos de 150 nodos, y calcula las soluciones por separado.
- Si hay más de 5 ramas por nodo: Introduce un nodo ficticio. Coloca sólo 4 ramas en el nodo original y traza una quinta rama hasta el nodo ficticio. Coloca las ramas que te quedan sobre el ficticio. Cada uno de los nodos ficticios te proporcionará hasta 5 ramas adicionales —usa justamente tantas como desees, pero ten cuidado con los nodos aleatorios y las probabilidades de sus ramas, porque dichas probabilidades deben sumar la unidad en cada nodo y al mismo tiempo preservar las probabilidades globales deseadas.

Finalmente, algunas pistas sobre re-ediciones importantes de los datos del árbol. La característica clave es la posibilidad de reservar una zona en memoria. A menudo un árbol tendrá el mismo subárbol en unos cuantos nodos diferentes. Así que, definiéndolo al principio una vez y luego grabándolo en memoria, puede posteriormente ser juntado en el sitio del árbol donde se requiera. No olvides, sin embargo, que puede que necesites usar las rutinas de edición para cambiar los valores concretos de dicho subárbol, al colocarlo en diferentes sitios. Una vez que un subárbol particular ha sido depositado, puede recuperarse tantas veces como se desee.

El otro uso importante de la memoria aparece cuando se desea insertar algunos pasos adicionales en la mitad del árbol, o suprimir algunos pasos intermedios. Simplemente basta grabar el final del árbol en memoria y luego editar el árbol principal tal y como se precisa, y posteriormente volver a encadenar el final del árbol de nuevo en su sitio.

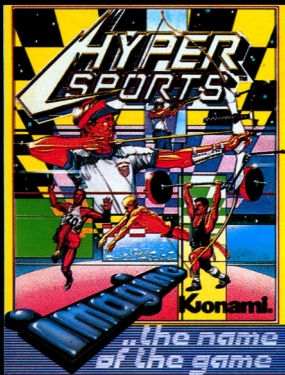
**FICHA DEL PROGRAMA DECISION MAKER**

Comercializa: **Indescomp.**  
 Soporte: **Disco.**  
 Sistema operativo: **CP/M 2.2.**  
 Compatible con: **CPC464, CPC664, CPC6128.**  
 Precio: **5.500 ptas.**

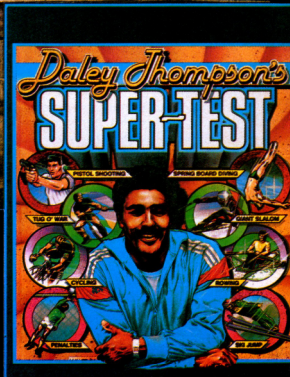
SI BUSCAS LO MEJOR **ERBE** Software LO TIENE

# TODO EL DEPORTE EN TU ORDENADOR

NATACION, TIRO AL PLATO, SALTO DE POTRO,  
LEVANTAMIENTO DE PESO,  
TIRO AL ARCO, TRIPLE SALTO



**PENALTIES, CICLISMO, TIRO DE PISTOLA,  
REMO, SALTO DE ESQUI, SLALOM GIGANTE  
TIRO DE CUERDA, SALTO DE TRAMPOLIN**



DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE, SANTA ENGRACIA, 17. Tel. 447 34 10.  
DELEGACION BARCELONA, Avd. MISTRAL, 10. Tel. (93) 432 07 31

# ASÍ SUENA EL AMSTRAD

*Los ordenadores, aparte de poder bregar con texto y gráficos, cada vez más incorporan un tratamiento del sonido más potente y sofisticado. El Amstrad no es una excepción, como podrá verse en estas páginas.*

# H

oy en día, la mayor parte de los micros son capaces de emitir sonidos. La forma más sencilla de conseguir que nuestro Amstrad haga ruido es introducirle:

PRINT CHR\$(7)

y pulsar ENTER. Lo malo es que ¿a quién le emociona un solitario beep?

Para permitirnos crear sonidos más interesantes, el Amstrad tiene tres comandos muy potentes—SOUND, ENT y ENV.—. Sin embargo, una mayor sofisticación en los comandos supone una mayor dificultad en su correcto uso, por lo menos al principio.

Así que durante las próximas semanas iremos describiendo esos comandos detalladamente y construiremos paso a paso una guía de **fabricar ruidos** con el Amstrad. Procuraremos que sea un curso práctico con muchos ejemplos, por lo tanto, asegúrese de que su micro esté encendido y listo para utilizarlo mientras lea estos artículos.

¡Ah!, no se conforme con entender lo que le decimos: ¡**pruébelo!** El comando SOUND es un tema muy amplio y solamente experimentando con él podrá conocerlo en su totalidad.

Dicho esto, vamos a producir sonidos. Teclee:

SOUND 1, 200, 100, 7

y nos sentiremos muy emocionados ante las excitantes tonalidades que emite nuestro Amstrad.

Bueno, quizá se lo hemos exagerado un poco. No hemos producido el sonido más importante y armónico del mundo. Una vez que conocamos cómo funciona el comando SOUND todavía tendremos que recorrer un largo camino para dominar el sonido del Amstrad.

La forma más sencilla del comando SOUND es cuando está seguido de cuatro parámetros. Que no le desanimen, son precisamente los números que siguen al comando SOUND y afectan a su forma de trabajar.

Los parámetros del último sonido que hicimos eran 1, 200, 100 y 7. El primer número (1) selecciona el canal por el que se reproducirá la nota, el segundo (200) indica al micro cuál será el tono. El tercer parámetro (100) determina la duración de la nota mientras que la última cifra (7) decide el volumen de reproducción.

Al teclear:

SOUND 1, 200, 100, 7

producimos una nota por el canal 1 que durará un segundo, reproducida al volumen más alto y con un tono bastante agudo.

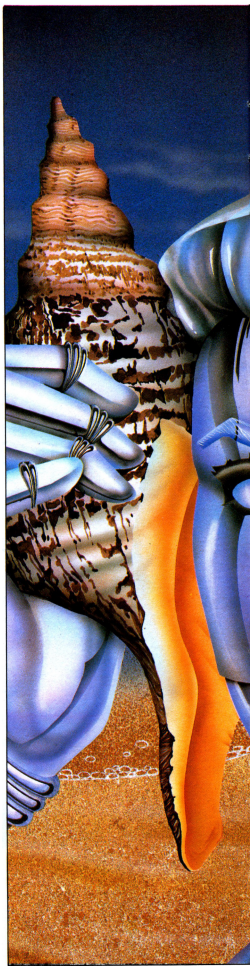
No se preocupe si todavía le cuesta un poco entender estos términos, volveremos a hablar sobre ellos a lo largo de este artículo.

## El comando SOUND

La estructura básica del comando SOUND es:

SOUND canal, tono, duración, volumen, esto es, la palabra clave SOUND seguida de cuatro parámetros que especifican el sonido concreto que buscamos.

Le sugerimos probar nuevos sonidos introduciendo en el ordenador:





SOUND 2, 600, 20, 4

SOUND 4, 20, 50, 3

o conviértase en un aventurero y fabrique los suyos propios.

Como hemos visto (*y escuchada*) por encima, los valores que siguen al comando SOUND pueden variar y, al hacerlo, cambian también las notas producidas. Sin embargo, no podemos elegir cualquier parámetro, sólo podemos escogerlos entre ciertos límites.

Tomemos el parámetro del canal, el primer número que viene detrás del comando SOUND. Se usa para decidir cuál de los canales de sonido del micro es el que se utiliza. El **Amstrad** tiene tres canales similares y cada uno sólo puede reproducir una nota al mismo tiempo.

No obstante, al poder estar los tres canales trabajando a la vez, el micro puede estar reproduciendo simultáneamente tres notas diferentes. Los conocemos como canales A, B y C, y los parámetros que los seleccionan son 1, 2 y 4 respectivamente.

El programa 1 nos muestra estos parámetros en acción.

Primero reproduce una nota por el canal A (*parámetro 1*), después espera que pulsemos una tecla. La siguiente nota se reproduce por el canal B ya que el parámetro que sigue al comando SOUND es la cifra 2.

### Programa uno

Cuando ha terminado esta nota y pulsamos una tecla, la última nota, con 4 como parámetro de canal, suena por el C.

Por supuesto que estas notas se reproducen una detrás de la otra. Esperamos que los lectores más inquietos se estén preguntando: **¿Cómo distinguimos los tres canales?** Podría tratarse solamente de un canal por el que suenan las notas una tras otra.

Si tiene alguna duda, pase al programa 2 que reproduce las tres notas anteriores pero ahora a la vez, por cada uno de los diferentes canales. No estamos escuchando a Mozart precisamente, **¿verdad?** Aunque no sea así, por lo menos nos habrá convencido sobre la existencia de los tres canales.

Vamos a dejar, por ahora, este parámetro —ya volveremos sobre él— y pasamos a ver el siguiente parámetro de tono. Como podemos suponer controla el tono de la nota producida por el comando SOUND.

## Programa Acción

El tono de una nota nos indica si es grave o aguda. La nota aguda —semejante a las producidas por una soprano o un molesto gato— se corresponde con un tono alto. La nota grave —como la de un contrabajo o una sirena de barco— es la del tono bajo.

### Programa dos

Ejecutemos el programa 3 y escucharemos una nota que cambia de tono, siendo éste cada vez más grave. Pulse ESC un par de veces cuando tenga suficiente.

Lo que está ocurriendo es que cada vez que se ejecuta el bucle FOR... NEXT, el comando SOUND de la línea 30 produce una nota. Pero ahora varía el valor del parámetro de tono, «tono», en cada vuelta del bucle. Al principio vale 32, después 33, después 34 y así sucesivamente. Como varía el parámetro del tono, cambia también la nota producida.

### Programa tres

Podemos observar que cuanto más alto sea el valor del parámetro, la nota tiene un tono más bajo (*el sonido es más grave*). Queremos decir con esto que un parámetro de tono igual a 100 produce una nota mucho más grave que uno igual a 30.

A propósito. Si hemos utilizado ESC para salir del programa anterior podríamos haber dejado alguna nota en el micro, esto puede provocar un sonido extraño (*mejor dicho, muy extraño*) en los programas siguientes, así que nos deshacemos de ellas introduciendo:

SOUND 129, 100, 0, 0

Parece que el parámetro de canal tiene un valor equivocado, pero ya hablaremos sobre ello en un artículo posterior. Por el momento lo utilizamos precisamente como cubo de la basura para eliminar todos estos restos de sonidos.

Volviendo al parámetro de tono, el programa 4 usa un paso de —4 para decrementar el valor de este parámetro en cada ciclo del bucle FOR... NEXT. El resultado es una nota que incrementa su tonalidad.

El parámetro de tono puede tomar valores comprendidos dentro de un rango de 0 a 4095 aunque funciona de una forma irregular en los extremos del intervalo. Los valores han de ser números enteros. Si utilizamos números con parte decimal para el parámetro de tono el CPC, sencillamente, ignora el decimal.

No se limite a seguirnos, compruébelo por usted mismo. Se puede apreciar alguna diferencia entre:

SOUND 1, 200.5, 100, 7  
SOUND 1, 200, 100, 7?

### Programa cuatro

Para aquellos que tengan pretensiones musicales, el Apéndice 7 del Manual del Usuario nos proporciona la tabla completa de los valores del parámetro de tono y las correspondientes notas musicales. Se ha llamado tono al parámetro aunque en realidad es el período del mismo, pero a partir de ahora vamos a considerar que ambos son la misma cosa.

La tabla 1 nos muestra 24 de estas notas y sus correspondientes valores para este parámetro.

Nota	Pitch Parámetro	Nota	Pitch Parámetro
C	239	F#	84
C#	225	G	80
D	213	G#	75
D#	201	A	71
E	190	A#	67
F	179	B	63
F#	169	C	60
G	159	C#	56
G#	150	D	53
A	142	D#	50
A#	134	E	47
B	127	F	45
C	119	F#	42
C#	113	G	40
D	106	G#	38
D#	100	A	36
E	95	A#	34
F	89	B	32

Tabla 1.

El programa 5 reproduce una selección de estas notas ordenadamente, leyendo los valores de «tono» desde la instrucción DATA de la línea 60.

### Programa cinco

Podemos observar que esta serie de notas nos proporcionan una sensación más «completa» que la que teníamos con las anteriores. También volveremos sobre esto en un próximo artículo.

De momento avancemos al siguiente parámetro que nos determina la longitud de la nota.

Este tercer parámetro, con el que se controla la duración de la nota, puede alcanzar valores comprendidos entre -32768 y 32767 pero ahora sólo utilizaremos los que van desde 1 a 32767 (valores positivos). Este número es la cantidad de centésimas de segundo que dura una nota.

Si el parámetro de duración es 100, la nota debe sonar durante un segundo. Si el parámetro es 1000, debe durar diez segundos. Teclean-do:

SOUND 1, 200, 300, 7  
oiremos una nota que dura tres segundos.

El programa 6, una variante del anterior, utiliza nuestro familiar bucle FOR... NEXT para cambiar el valor del parámetro de duración por medio de la variable «duración».

### Programa seis

Las notas cambian de tono cada vez que hacemos el bucle y además incrementamos su duración desde

una décima hasta 1.2 segundos. Esto nos da una sensación semejante al sonido de un objeto que se está parando.

En el programa 7 se decrementa al parámetro de duración en cada giro del bucle, produciéndose así el efecto contrario.

### Programa siete

Quedamos que el parámetro final era el del volumen. Como su propio nombre indica, decide cuál ha de ser el volumen de la nota y su valor será cualquier número entero comprendido entre 0 y 7. Si el parámetro es 1, la nota casi no se escucha. Cuando el parámetro sea 2, el sonido se escucha un poco más alto y alcanzaremos el máximo cuando el parámetro de volumen sea igual a 7. El programa 8 pone en marcha todo esto.

### Programa ocho

La mayor parte del programa se sigue con bastante facilidad. Como se repite el bucle FOR... NEXT, el parámetro del volumen se incrementa



y la nota se escucha cada vez más fuerte.

Pero, ¿qué ocurre en la línea 40 en la que hay un comando SOUND con un parámetro que vale 0? ¿Qué pinta aquí?

La contestación es que mantiene un intervalo de silencio entre las otras notas. Cuando este parámetro vale 0 significa que el sonido no tiene volumen. No lo podemos oír.

No obstante, la nota se reproduce (aunque silenciosamente) durante todo el tiempo indicado por el parámetro de duración. Lo cual quiere decir que la línea 40 produce un segundo de silencio entre cada nota.

Si bien puede parecerse un poco extraño a primera vista utilizar un parámetro que vale 0 para producir una nota que no podemos oír, hay ocasiones en las que resulta muy útil. Podemos usarlo para lograr las pausas que encontramos frecuentemente en las melodías. También hace que las cosas se vean más claras a veces. Quitemos la línea 40 del último programa y comprenderemos lo que acabamos de decir.

Y hemos llegado al final de este recorrido por los cuatro parámetros del comando SOUND. Se nos olvidaba una cosa. ¿Puede explicarnos por qué las notas producidas por:

SOUND 1, 200, 20, 4

y

SOUND 1, 200

son exactamente iguales?

La razón es que si no facilitamos al comando SOUND el parámetro de duración o de volumen él mismo lo asume. El valor del parámetro de duración por defecto es 20 y el de volumen es 4, por lo tanto, los dos comandos SOUND anteriores producen la misma nota.

Y es todo por el momento. La tabla 2 resume todo nuestro recorrido, mientras que el programa 9 lo utiliza para tocar una conocida melodía. Quién sabe lo que nos «encontraremos» al escucharla.

	Canal	Pitch	Duración	Volumen
rango	1=A	0	1	0
	2=B	to	to	to
	4=C	4095	32767	15
defecto	ninguno	ninguno	20	4

Tabla 2. Rango de los parámetros del comando SOUND.

¿Por qué no escribe alguna canción sencillita usted mismo?

## Program Acción

```
10 REM PROGRAMA I
20 REM Canal A
30 SOUND 1,478,100,15
40 WHILE INKEYS="" :WEND
50 REM Canal B
60 SOUND 2,379,100,15
70 WHILE INKEYS="" :WEND
80 REM Canal C
90 SOUND 4,301,100,15
```

```
10 REM PROGRAMA VI
20 FOR duracion=10 TO 120 STEP 10
30 READ tono
40 SOUND 1,tono,duracion,15
50 NEXT duracion
60 DATA 119,113,106,100,95,89
70 DATA 84,80,75,71,67,63
```

```
10 REM PROGRAMA II
20 REM Canal A
30 SOUND 1,478,100,15
50 REM Canal B
60 SOUND 2,379,100,15
80 REM Canal C
90 SOUND 4,301,100,15
```

```
10 REM PROGRAMA VII
20 FOR duracion=120 TO 10 STEP -10
30 READ tono
40 SOUND 1,tono,duracion,15
50 NEXT duracion
60 DATA 119,113,106,100,95,89
70 DATA 84,80,75,71,67,63
```

```
10 REM PROGRAMA III
20 FOR tono=32 TO 60
30 SOUND 1,tono,100,15
40 NEXT tono
```

```
10 REM PROGRAMA VIII
20 FOR volumen=1 TO 15
30 SOUND 1,100,100,volumen
40 SOUND 1,100,100,0
50 NEXT volumen
```

```
10 REM PROGRAMA IV
20 FOR tono=478 TO 253 STEP -4
30 SOUND 1,tono,100,15
40 NEXT tono
```

```
10 REM PROGRAMA IX
20 SOUND 1,239,100,15
30 SOUND 1,213,100,15
40 SOUND 1,268,100,15
50 SOUND 1,536,100,15
60 SOUND 1,358,100,15
```

```
10 REM PROGRAMA V
20 FOR nota=1 TO 12
30 READ tono
40 SOUND 1,tono,100,15
50 NEXT nota
60 DATA 119,113,106,100,95,89
70 DATA 84,80,75,71,67,63
```

# INSTRUCCIONES DE ROTACION Y DESPLAZAMIENTO (III)

**En los últimos capítulos hemos estudiado con detenimiento una serie de instrucciones que tratan de la manipulación de los bits que componen un byte.**

**Como habréis podido observar el contenido teórico ha sido bastante extenso debido a la cierta complejidad que representa el manejo y manipulación de bits. Por este motivo hemos creído conveniente dedicar este capítulo al estudio de algunos ejemplos prácticos que nos aclararán las posibles dudas que aún podamos tener sobre dichas instrucciones.**



El primer programa que hemos preparado nos va a resultar de gran ayuda para acabar de comprender todos las instrucciones referentes a las manipulaciones de bits. Se trata de una rutina que nos va a permitir ver en pantalla cuál es el comportamiento de un registro o posición de memoria al aplicarle cualquiera de las instrucciones de rotación y desplazamiento.

## Programa uno

Vamos a estudiar paso a paso cada una de las líneas que componen el programa número 1, para ver qué función cumple cada una de las instrucciones.

En primer lugar, dado que lo que deseamos es imprimir algo en pantalla, deberemos seleccionar el modo en el que desamos trabajar. Esto se realiza en las dos primeras líneas; en primer lugar cargamos en el acumulador el modo elegido (en este caso el modo 1), y luego llamamos a la rutina SET MODE, que se encuentra en el firmware, que será la encargada de seleccionarlo.

A continuación entramos en un bucle que se encargará de imprimir en pantalla el siguiente texto:

CARRY (IX+0)

ya que lo que nos va a mostrar será el contenido de flag Carry y el contenido de la posición de memoria a la que apunta el registro doble IX.

Seguidamente, un segundo bucle será el encargado de rotar los bits del registro deseado. Para ello cargamos el acumulador con el contenido del registro doble IX; como sabemos, este registro indexado apunta al último valor dado en la llamada al código máquina por el comando CALL. Por lo que deberemos llamar a la rutina en código máquina de la siguiente forma:

CALL &A000,x

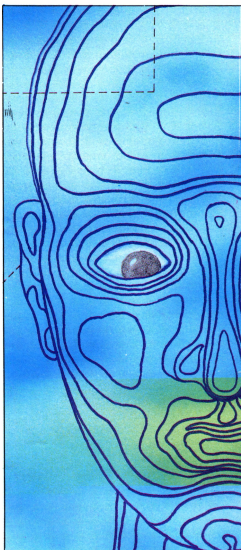
donde el operando 'x' será cualquier número entre 0 y 255, que será el valor que se imprimirá en pantalla y sobre el cual se producirán las sucesivas rotaciones.

El siguiente paso es cargar en el registro D el contenido del acumulador; esto debe ser así ya que la rutina que se encarga de imprimir en pantalla toma los datos de dicho registro. A continuación preservamos todos los flags mediante la instrucción PUSH AF para poder recuperar más tarde el valor del Carry que es el que aquí nos interesa, y poder imprimirlo en pantalla. La siguiente línea es la que se encarga de llamar a la subrutina de reposición de coordenadas para poder, más tarde, imprimir en pantalla.

Ahora volvemos a recuperar AF y seguidamente lo preservamos de nuevo; llamamos a la rutina que se encarga de impresión en pantalla. Después de esto hacemos una llamada a la rutina del firmware WAIT KEY que esperamos hasta que se pulse una tecla.

En la línea 270 deberemos poner la instrucción de rotación que deseamos observar. Así pues, si queremos ver en pantalla los sucesivos efectos que produce la instrucción RR

F. L. Fernández



(IX+0), deberemos poner en dicha línea lo siguiente:

270 RR (IX+0)

por lo tanto, en esta línea deberemos poner la instrucción que deseamos observar en pantalla. A partir de aquí se produce ocho veces este bucle y volvemos al Basic.

Las líneas 310-320 se encargan de mirar el contenido del Carry, si éste es cero se llama a la rutina que imprime un cero y si es uno se llama a la encargada de imprimir un uno. Las líneas 389-470 se encargan de observar cada uno de los bits que componen en ese momento el registro y llaman a las rutinas de impresión de uno a cero según sea uno o cero el contenido del bit en ese momento.

A continuación nos encontramos con las rutinas encargadas de imprimir un uno o un cero en pantalla, y finalmente la rutina encargada de reponer las coordenadas.

## Programa dos

Vamos ahora a centrarnos en el segundo programa que hemos incluido para la explicación de esta serie de instrucciones. Se trata

de una rutina que produce un scroll a la derecha de toda la pantalla.

Diremos en primer lugar, que puede utilizarse para cualquier modo de pantalla, pero dado que los modos de baja resolución de pantalla, modos uno y cero, utilizan una estructura de color un tanto especial, únicamente conseguiremos un buen efecto trabajando en el modo de alta resolución, o sea el modo 2.

Pasaremos ahora a describir el programa empezando por la línea 30, aquí cargamos el registro doble HL con la posición de memoria &C000, que es donde empieza la memoria de pantalla. Luego entramos en un bucle que se repetirá 25 veces, que es el número de líneas de que está compuesta la pantalla.

Al entrar en este bucle lo primero que debemos hacer es preservar los registros BC y HL, ya que luego se corromperán. Acto seguido nos encontramos con otro bucle que deberá repetirse ocho veces, ya que éstos son los bytes que componen cada carácter, y preservamos de nuevo HL para poder recuperarlo más tarde.

Por último, nos encontramos con el bucle que será el encargado de producir el efecto de scroll. Este bucle deberá repetirse 80 veces, ya que éste es el número de columnas de

que está compuesta la pantalla. Dentro de este bucle podemos ver la instrucción:

```
110 BUC: RR (HL)
```

ésta es la encargada realmente de producir el efecto de scroll hacia la derecha de la pantalla. Si quisiéramos obtener un scroll hacia la izquierda, únicamente deberíamos cambiar esta línea por la siguiente:

```
110 BUC: RL (HL)
```

ya que dicha instrucción produce una rotación hacia la izquierda de los bits que componen un byte.

Una vez completada la rotación de los primeros 80 bytes, deberemos pasar a los de la siguiente fila, para ello deberemos sumar a la posición inicial de pantalla el valor 2048, ya que ésta es la separación entre cada dos bytes que forman un carácter.

Cuando se haya completado la rotación de los ocho bytes del primer carácter, deberemos sumar a la posición inicial de pantalla, el valor 80, ya que ésta es la distancia que separa a los bytes de dos filas consecutivas.

Así pues, cuando se hayan completado las 25 pasadas habremos conseguido un desplazamiento de un byte hacia la derecha de todos los caracteres que forman la pantalla.

## Programa tres

El tercer programa que hemos creído conveniente incluir en este artículo, se trata de una rutina que produce una multiplicación por dos del número dado.

En realidad este programa está constituido por dos rutinas bien diferenciadas, una de ellas es la encargada de la multiplicación propiamente dicha, y la segunda es la que se encargará de imprimir en pantalla el resultado de la operación realizada. Empezaremos la descripción del programa explicando en primer lugar el bloque que tiene como misión realizar la operación de multiplicar.

Dado que el programa necesita un dato para poder trabajar, que es el número que queremos multiplicar, éste se lo proporcionaremos, en la llamada a la rutina desde el Basic, de la siguiente forma:

```
CALL &A000,m
```

donde la variable 'm' será un número comprendido entre 0 y 65535, que es el mayor valor que puede contener un registro de 16 bits.

Una vez hemos llamado a la rutina, éste se encarga en primer lugar de asignar al regis-

## PROGRAMA 1

```

AR00 39 39 0RG AR00R
AR00 3081 49 LD A,I
AR02 C08EC 54 CALL #CEC
AR05 2187C 69 LD HL,#8C7
AR00 C0780 75 LD HL,#8C
AR00 2175A 88 LD HL,#8C
AR0C 84F 95 LD R,15
AR10 7E 100 REP: LD A,(HL)
AR11 C0680 114 CALL #8C4
AR14 23 129 INC HL
AR15 18F 136 SINC REP
AR17 2189E 148 LD HL,#8E7
AR10 2078A 156 LD HL,(HL)
AR10 2187F 168 LD HL,#788
AR20 84E 176 LD B,8
AR22 C5 188 BUC1: PUSH BC
AR22 D0788 195 LD A,(IX)
AR24 57 204 LD D,A
AR27 E5 214 PUSH HL
AR20 F5 228 PUSH AF
AR0F C08A8 236 CALL C08D0
AR2C F1 248 POP AF
AR2C F5 256 POP AF
AR2C C05A8 264 CALL TANT
AR31 C0180 270 CALL #8B10
AR24 F1 288 POP AF
AR30 E1 298 POP HL
AR2C D0C882E 310 SRA IX
AR3A C1 318 POP BC
AR30 18C 328 SINC BUC1
AR3C C9 338 RET
AR3C D0C08 346 PINT: CALL C,IND
AR41 D42A8 358 CALL #C,CERO
AR44 84A 368 LD B,A
AR44 3E28 378 PUSH: LD A,-
AR00 C0480 388 CALL #8B0A
AR40 18F 396 SINC PAS
AR40 84E 408 LD B,D
AR4F C87A 418 BUC1: BIT 7,D
AR40 84E 428 LD B,D
AR54 C4C88 438 CALL Z,CERO
AR57 C12 448 SINC HL
AR59 18E4 458 SINC BUC
AR58 C9 468 RET
AR5C 3E31 478 AND: LD A,"1"
AR5C D05A8 488 CALL #8B0A
AR61 C9 498 RET
AR62 3E38 508 SINC: LD A,"1"
AR6A C05A8 518 CALL #8B0A
AR6C C1 528 RET
AR6C 2C 538 CORO: LD HL,(POS)
AR6C 273A8 548 INC L
AR6F C0780 558 CALL #8B75
AR72 C9 568 RET
AR75 273 578 POP: DEFS 2
AR76 4341523 598 HEXL: DEFN "ARRY (IX)"

```

```

BUC AR4F BUC1 AR22 CERO AR42
C08D0 AR4F #8C AR75 PAS AR44
PINT AR6E POS AR75 REP AR18
IND

```

## PROGRAMA 2

```

AR00 39 39 0RG AR00R
AR00 2184C 48 LD HL,#888
AR03 C5 58 LD B,2
AR05 C5 68 BUC2: PUSH BC
AR04 C5 78 PUSH HL
AR07 8E9B 88 LD L,C
AR09 E5 98 BUC1: PUSH HL
AR0A 8458 100 LD B,8
AR0C C81C 118 BUC1: INC HL
AR0E 23 120 INC HL
AR07 18F 128 SINC BUC
AR11 E1 136 POP HL
AR12 11888 138 LD HL,(280)
AR15 19 148 ADD HL,DE
AR16 80 158 DEC C
AR17 79 168 LD A,C
AR18 8E7 178 LD HL,#8C
AR1A E1 200 POP HL
AR1B 11888 218 LD HL,(280)
AR1E 19 228 POP HL
AR1F C1 238 POP BC
AR20 18E3 248 SINC BUC
AR22 C9 258 RET

```

```

BUC AR0C BUC1 AR07 BUC2 AR05

```

## PROGRAMA 3

```

AR00 39 39 0RG AR00R
AR00 D04E8 48 LD C,(1048)
AR03 D04E8 58 LD A,(1048)
AR0A C821 68 SIA C,POS
AR0C 388D 78 JR C,POS
AR0A C828 88 SIA B,POS
AR0C 388E 98 JR C,POS
AR0E C078A 100 CALL #8A
AR11 C9 110 CALL #8A
AR12 C928 128 PMS: SIA B,B
AR14 C8B 138 SINC B,B
AR16 38E4 148 JR C,POS1
AR18 C078A 158 CALL #8A
AR1B C9 168 RET
AR1C 2187F 178 PMS1: LD HL,#788
AR1F C8E4 188 SIA HL
AR21 C8C2 198 SET B,(HL)
AR22 C078A 208 CALL #8A
AR25 C9 218 RET
AR27 7E 228 HEXL: LD A,(HL)
AR28 C078A 238 CALL #8A
AR28 7E 248 LD HL,(HL)
AR2C C078A 258 CALL #8A
AR2F 79 268 LD HL,(HL)
AR30 C078A 278 CALL #8A
AR33 38D0 288 LD HL,(HL)
AR35 D0C88 298 CALL #8B57
AR38 C9 308 RET
AR39 F5 318 PMS1: PUSH AF
AR3A E4F8 328 SRA A
AR3C C8F 338 SRA A
AR3E C8F 348 SRA A
AR40 C8F 358 SRA A
AR44 C04A8 370 CALL #8E7
AR47 F1 388 POP HL
AR48 E4F 398 AND MF
AR4C FE3A 410 OR CF,"1"
AR4E 280 420 LD C,(POS)
AR4B C8C 438 AND A,-4
AR52 C05A8 448 CALL #8B0A
AR55 C9 458 RET

```

```

HEVA AR27 #8B1 AR52 PAS AR12
PMS1 AR1C #8A AR39 #8E AR4A

```

```

10 REM * CARGADOR PROGRAMA 1 *
20 FOR N=&A000 TO &A0E4
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>&39D9 THEN PRINT "ERROR
EN DATAS"
70 DATA 62,1,205,14,188,33,7
80 DATA 12,205,117,187,33,117,160
90 DATA 6,15,126,205,90,187,35
100 DATA 16,249,33,9,14,34,115
110 DATA 160,33,0,112,6,8,197
120 DATA 221,126,0,87,229,245,205
130 DATA 104,160,241,245,205,62,160
140 DATA 205,24,187,241,225,221,203
150 DATA 0,46,193,16,229,201,220
160 DATA 92,160,212,98,160,6,4
170 DATA 62,32,205,90,187,16,249
180 DATA 6,8,203,122,204,98,160
190 DATA 196,92,160,203,18,16,244
200 DATA 201,62,49,205,90,187,201
210 DATA 62,48,205,90,187,201,42
220 DATA 115,160,44,34,115,160,205
230 DATA 117,187,201,0,0,67,65
240 DATA 82,82,89,32,32,32,40
250 DATA 73,88,43,48,41,32,0

```

```

10 REM * CARGADOR PROGRAMA 2 *
20 FOR N=&A000 TO &A023
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>&9C9 THEN PRINT "ERROR
EN DATAS"
70 DATA 33,0,192,6,25,197,229
80 DATA 14,8,229,6,80,203,30
90 DATA 35,16,251,225,17,0,8
100 DATA 25,13,121,32,239,225,17
110 DATA 80,0,25,193,16,227,201

```

```

10 REM * CARGADOR PROGRAMA 3 *
20 FOR N=&A000 TO &A056
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>&2B52 THEN PRINT "ERROR
EN DATAS"
70 DATA 221,78,0,221,70,1,203
80 DATA 33,56,8,203,32,56,14
90 DATA 205,39,160,201,203,32,203
100 DATA 192,56,4,205,39,160,201
110 DATA 33,0,112,203,38,203,198
120 DATA 205,39,160,201,126,205,57
130 DATA 160,120,205,57,160,121,205
140 DATA 57,160,62,13,205,93,187
150 DATA 201,245,230,240,203,63,203
160 DATA 63,203,63,203,63,205,74
170 DATA 160,241,230,15,198,48,254
180 DATA 58,56,2,198,7,205,90
190 DATA 187,201,0,0,0,0,0

```

# MICRO-1

C/ Duque de Sesto, 50  
28009 Madrid  
Tels. (91) 275 96 16  
MICROLID: Gregorio Fdez., 6  
47006 Valladolid

el IVA lo paga  
MICRO-1

## SOFTWARE: Por cada programa GRATIS ¡¡1 bolígrafo con reloj de cuarzo!!

RAID _____	2.300 ptas.	EXPLODING FIST _____	2.300 ptas.
JUMP JET _____	2.495 ptas.	FIGHTER PILOT _____	1.975 ptas.
TALES OF ARABIAN _____	1.950 ptas.	MASTER OF T. LAMP _____	1.950 ptas.
SABREWULF _____	1.650 ptas.	NIGHTSHADE _____	1.950 ptas.
GHOSTBUSTERS _____	1.950 ptas.	HACKER _____	1.950 ptas.
HIGHWAY ENCOUNTER _____	1.750 ptas.	MAPGAME _____	2.700 ptas.

### IMPRESORAS

¡¡20% DE DESCUENTO SOBRE P.V.P.!!

LAPIZ OPTICO  
¡¡4.900 ptas.!!

YOYSTICK QUICK SHOT II \_\_\_\_\_ 2.395 ptas.  
JOYSTICK QUICK SHOT V \_\_\_\_\_ 2.795 ptas.

CASSETTE ESPECIAL  
ORDENADOR 5.295 ptas.

TOSHIBA MSX 64 K  
¡¡34.900 ptas.!!

CINTA C-15 ESPECIAL ORDENADOR \_\_\_\_\_ 85  
DISKETTE 3" \_\_\_\_\_ 990

PC-COMPATIBLE IBM 256 K  
MONITOR FOSFORO VERDE  
2 BOCAS DISKETTE 360 K  
SOLO ¡¡243.900!!

PC-PORTATIL  
2 BOCAS 360 K  
¡¡174.900!!

TAPA METACRILATO PARA  
TECLADO ¡¡2.300 ptas.!!

PRECIOS SUPER-EXCEPCIONALES PARA  
AMSTRAD CPC-472 y CPC-6128  
¡¡LLAMANOS, TE ASOMBRARAS!!

UNIDAD DE DISCO CON  
CONTROLADOR: 49.900

SINTETIZADOR DE VOZ  
¡¡8.495 ptas.!!

MODULADOR TV  
8.400 ptas.

INTERFACE DISCO  
5 1/4" 5.300

### LIBROS:

Curso autodidáctico Basic I _____	2.525 ptas.
Curso autodidáctico Basic II _____	2.525 ptas.
Programando con Amstrad _____	2.195 ptas.
Juegos sensacionales Amstrad _____	1.950 ptas.
Hacia la inteligencia artificial _____	1.295 ptas.
Música y sonidos con Amstrad _____	995 ptas.

IMPRESORA MARGARITA  
¡¡49.900 ptas.!!

Pedidos contra reembolso sin ningún gasto de envío. Tels. (91) 275 96 16/274 53 80 o escribiendo a Micro-1.  
C/ Duque de Sesto, 50. 28009 Madrid.

# Sin duda alguna

A través de esta sección se pretende resolver, en la medida de lo posible, todas las posibles dudas que «atormenten» a todas las personas interesadas en el mundo del AMSTRAD, sean o no poseedores de uno y, si lo son, se encuentren en cualquier nivel de destreza en su manejo.

Semanalmente, aparecen en estas páginas las consultas de la mayor cantidad de usuarios posible; ello redundará en un mejor servicio y en un contacto más estrecho entre todos nosotros a través de la revista.

**SIN DUDA ALGUNA** está abierta a todos.

En el ordenador **Amstrad** las operaciones lógicas se valoran como -1 si son ciertas y 0 si son falsas. Otros ordenadores las valoran respectivamente como +1 y 0, lo que a primera vista parece más lógico. ¿Tiene alguna ventaja particular el sistema adoptado por **Amstrad**?

**J. Esteve (Barcelona)**

Que nosotros sepamos, no presenta ninguna ventaja en particular, y la elección de un tipo u otro de «lógica de verdad depende exclusivamente de los diseñadores de la máquina a nivel de hardware.

Por tanto, la razón de la elección de lógica negativa para el **Amstrad** probablemente haya que buscarla en las «tripas» del ordenador.

1.º ¿Hay posibilidad de conectar una impresora tipo "SERIE", al CPC 664, en caso afirmativo indicar la referencia (Código), del producto y su precio actual?

2.º Capacidad de ampliación de la memoria RAM del CPC 664, así como la forma de la (s) ampliación

(es), ya sea por tarjetas, interna (KITS), cartuchos, etc... Indicar precio de éstas.

3.º ¿Existe alguna posibilidad de que en un futuro no muy lejano se fabrique el CPC 664, pero con un MZ-80 de 16 bits?

Quedo a la espera de vuestras noticias, y a ver si me toca en el sorteo, el tan nombrado, y no sin razón, CPC 664 de **Amstrad**.

**Luciano Pérez Coello (Tenerife)**

1.º La posibilidad de poder conectar o no una impresora serie a un **Amstrad** depende de que exista para él una interface RS232. Una vez cumplido este requisito, cualquier impresora que se adapte a este protocolo de comunicación puede servirte. Actualmente, que sepamos, existen 2 en nuestro mercado: una comercializada por Indescomp, distribuidora oficial de **Amstrad** en España, y la segunda la vende M.H.T. ingenieros. Encontrarás cumplida información de direcciones y teléfonos de ambos en las páginas de publicidad de la revista.

2.º Todavía no se comercializan las prometidas ampliaciones de memoria para el 664 a 128 y 256 Kbytes. Desconocemos el formato que adoptarán, aunque presumiblemente sea en forma de cartucho conectable al bus de expansión.

3.º Creemos que no. De aparecer un **Amstrad 10 bits**, probablemente la casa se incline por la compatibilidad IBM, lo cual significa un microprocesador Intel 8088/8086.

Poseo un **Amstrad 464**. Lo he implementado con la unidad de disco y lo utilizo, principalmente, para tratamiento de textos. Quisiera utilizarlo para llevar un fichero de artículos técnicos muy extenso, pero me dejaron para probar la base de datos que viene en el disco del 664 y a la hora de clasificar el fichero me daba overflow en 6DF4, con sólo 150 fichas de unos doscientos caracteres repartidos en 6 campos. Teóricamente tiene capacidad para más de 30.000 fichas pero con 162 me da, en la clasificación overflow en 64D6.

Otra consulta que os dirijo es si conocéis algún reubicador para el **Amstrad 464**, y si lo conocéis, dónde puedo conseguirlo.

**Angel Dehesa (Santander)**

Por la descripción del error que el programa de base de datos le da, creemos que tal programa ha sido hecho con cierto compilador de Pascal existente en nuestro mercado, y que dicho fallo sea uno de diseño en el programa, difícilmente reparable por usted. En cuanto al tema del reubicador, su carta no aclara lo suficiente si esta pregunta está relacionada o no con el problema de la base de datos, pero en cualquier caso la respuesta es no: no existe por el momento.

Sin embargo, algunos de los programas de la casa Hisoft emplean una técnica muy ingeniosa para reubicar código de máquina, siempre que el programa a reubicar presente un determinado formato.

## Amstrad Ideas

**AMSTRAD** Semanal comunica a todos sus lectores la apertura de una nueva sección dedicada a recoger las mejores ideas que exploten al máximo las posibilidades del ordenador, materializadas en programas claros y cortos (máximo 25 líneas). Los mejores de entre todos ellos serán publicados con el nombre de su autor en la revista, recibiendo como premio, gratuitamente en su domicilio los cuatro primeros números de nuestra cita mensual. Los programas enviados deberán incluir:

— Cinta de cassette con el programa o programas grabados.

— Explicación detallada del funcionamiento y propósito del programa, mecanografiado a 2 espacios o con letra clara.

Es imprescindible indicar en el sobre claramente: **AMSTRAD IDEAS**.

La dirección es:

**Hobby Press, S. A.**

La Granja, s/n.

Polígono Industrial de Alcobendas.

Madrid.



DISTRIBUIDORES DE PRODUCTOS INFORMATICOS Y ELECTRONICOS

Tenemos todo lo que necesitas en informática...

## AL MEJOR PRECIO

- AMSTRAD (ordenadores y periféricos)
- APRILLOT
- IMPRESORAS (New Print, Riteman, Star, C. Itoh, etc.)
- COMPATIBLES IBM
- OTROS

...y el mejor asesoramiento para ti y tu ordenador. **Llámanos y te informaremos**

Hermosilla, 75, 1 - Oficina 14 - 28001 MADRID - Tel. (91) 276 43 94/435 04 70

# ORDEMANIA

SOFT

## TE OFRECEMOS EL NUEVO PLAN GENERAL CONTABLE CON I.V.A.

- Contabilidad CPC 664 y CPC 6128 13.900 ptas.
- Contabilidad PCW 8256 37.500 ptas.

Disponemos de un equipo de software a tu servicio.  
Hacemos programas a medida.

**RECUERDA:** —Damos solución a la pequeña y mediana empresa.

Torres Quevedo, 34

Tel. 967/22 794

02003 ALBACETE

## MANTENGA SU AMSTRAD COMO NUEVO CON ESTA PRACTICA FUNDA.

POR SOLO: **2.250** ptas.

Ahora puede recibir la suya.  
Rellene el cupón y envíelo a:  
BAZAR POPULAR  
Apartado 27.500  
08080 BARCELONA

Deseo recibir el siguiente pedido:

- Funda AMSTRAD 464. 2.250 ptas.
- Funda AMSTRAD 664. 2.250 ptas.
- Funda AMSTRAD 6128. 2.250 ptas.

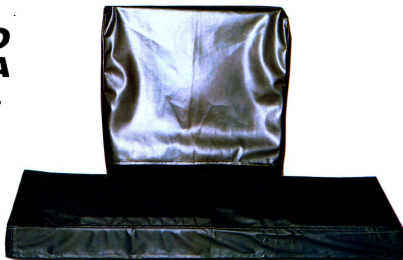
Indique su monitor:  Verde  Color

Disponemos también de los siguientes modelos: SEIKOSHA SP-800/1000 (1.200 ptas.). SPECTRUM 16/48 (430 ptas.). SPECTRUM PLUS (560 ptas.). COMMODORE 64 y VIC-20 (780 ptas.). SAGA-1 EMPEROR (650 ptas.). IMPRESORA AMSTRAD DMP-1 (1.400 ptas.). Indique la que desee.

Forma de pago:  Contra-reembolso  Sellos de correos adjuntos.

Gastos de envío: 150 ptas.

NOMBRE ..... EDAD .....  
DOMICILIO ..... TELEF. ....  
POBLACION .....  
CODIGO POSTAL ..... PROVINCIA .....



# Mercado común

Con el objeto de fomentar las relaciones entre los usuarios de AMSTRAD, **MERCADO COMUN** te ofrece sus páginas para publicar los pequeños anuncios que relacionados con el ordenador y su mundo se ajusten al formato indicado a continuación.

En **MERCADO COMUN** tienen cabida, anuncios de ventas, compras, clubs de usuarios de AMSTRAD, programadores, y en general cualquier clase de anuncio que pueda servir de utilidad a nuestros lectores.

Envíanos tu anuncio mecanografiado a: **HOBBY PRESS, S.A. AMSTRAD SEMANAL.**

Apartado de correos 54.062  
28080 MADRID

**¡ABSTENERSE PIRATAS!**

Desearía contactar con usuarios de **Amstrad** de Zaragoza de forma especial para comprar, vender, intercambiar, juegos, información ideas sobre el ordenador. (Programas prácticos, juegos...). Compraría joystick en buen estado y barato. Mi dirección: Pilar Gayán Claro. C/ San Ignacio de Loyola, 1 - 6.º A. Tel. (976) 22 23 61. Llamar a ser posible 14 a 17 y 20 a 22,30 H.

Vendo impresora Compute Mate-100 modelo CPA-80. A estrenar. Doble tracción-fricción, 100 CPS cable centronic. Funda. Garantía. Muy barato. Tardes-noches. Ana Saiz. Madrid. Tel. (91) 200 57 01.

Cambio monitor fósforo verde GT-64 y alimentador/modulador MP-1, por monitor color CTM-640. Abono diferencia precio. Fernando Corredera. Tel. 759 25 28. Madrid.

Vendo Amstrad CPC-664 con unidad de disco y monitor en color. Nuevo. 2 meses, muy poco uso. Garantía Indescomp. Con manuales, disco CP/M, Logo, Disco Utilidades, Base de Datos, Procesador de Texto, Random Files, Gráficos, Puzzle... Un disco Virgen 3". Por cambio de ordenador, 117.000 ptas. Tel. (981) 59 34 43. Pedro. Por las tardes. Santiago.

PUBLICIDAD



GABINETE DE INFORMÁTICA

- Clases de Informática sobre AMSTRAD
- Exclusivamente individuales.
- Ordenadores AMSTRAD y periféricos
- Los mejores precios
- Software a la medida

ZURBANO, 4 ☎ 410 47 63  
28010 MADRID

## DATA BECKER APUESTA FUERTE POR AMSTRAD



Otros una colección muy interesante de sugerencias, ideas y soluciones para la programación y utilización de su CPC-464. Desde la estructura del hardware, sistema de funcionamiento - Tokens Basic, dibujos con el joystick, aplicaciones de ventanas en pantalla y otros muchos interesantes programas como el procesamiento de datos, editor de textos, generador de caracteres, monitor de código máquina hasta listados de interesantes juegos.  
**CPC-464 Consejos y Trucos. 263 págs. P.V.P. 2.290,- ptas.**



Escrito para alumnos de los últimos cursos de EGB y de BUP, este libro contiene muchos programas para resolver problemas y de aprendizaje, descritos de una forma muy completa y fácil de comprender. Temática de Programación, progresiones geométricas, escritura de cartas, crecimiento exponencial, verfos irregulares, quédulas cuadráticas, movimiento pendular, estructura de moléculas; cálculo de interés y muchas cosas más.  
**CPC-464 El libro del colegio. 388 págs. P.V.P. 2.290,- ptas.**



PEKES, POKES y CALLS se utilizan para introducir al lector de una forma fácilmente accesible al sistema operativo y al lenguaje máquina del CPC. Proporciona además muchos e interesantes posibilidades de aplicación y programación de su CPC.  
**PEKES Y POKES del CPC 464/6128. 160 págs. P.V.P. 1.990,- ptas.**



La técnica y programación del Procesador Z80 son los temas de este libro. Es un libro de estudio y de consulta imprescindible para todos aquellos que poseen un Commodore 128, CPC, MSX u otros ordenadores que trabajan con el Procesador Z80 y desean programar en lenguaje máquina.  
**El Procesador Z80. 560 págs. P.V.P. 3.800,- ptas.**



El LIBRO DEL FLOPPY del CPC lo explica todo sobre la programación con discos y la gestión relativa de ficheros mediante el floppy DDI 1 y la unidad de discos incorporada del CPC 664/6128. La presente obra, un auténtico estándar, representa una ayuda incomparable tanto para el que desee iniciarse en la programación con discos como para el más curtido programador de ensamblado. Especialmente interesante resulta el listado exhaustivamente comentado de DOS y los muchos programas de ejemplo, entre los que se incluye un completo sistema de gestión de ficheros.  
**El libro del Floppy del CPC. 353 págs. P.V.P. 2.800,- ptas.**



¿Domar CP/M por fin! Desde explicaciones básicas para almacenar numéricos, la protección contra la escritura, o ASCII, hasta la aplicación de programas auxiliares de CP/M, así como «CP/M interno» para invadidos, cada usuario del CPC rápidamente encontrará las ayudas e informaciones necesarias para el trabajo con CP/M. Este libro tiene en cuenta las versiones CP/M 2.2, así como CP/M Plus (3.0), para el AMSTRAD CPC 464, CPC 664 y CPC 6128.  
**CP/M. El libro de ejercicios para CPC. 204 págs. P.V.P. 2.800,- ptas.**



**TEXTOMAT 8.800 ptas.**

¡El procesador de textos más vendido en Alemania, ahora también disponible para

**AMSTRAD**

BOLETIN DE PEDIDO

**FERRE-MORET S.R.L.**

Turati n.º 8, entlo. 2 - Tel. 218 02 93  
BARCELONA 08006

Devolvo aduando 300 ptas.

NOMBRE

DIRECCION

Adquiro cheque  Reembolso más gastos del mismo



# NOVEDADES

## Amsoft

La mayor colección de programas para AMSTRAD



### 3D STUNT RIDER.



Participa en uno de los más excitantes deportes con riesgo: el motorismo acrobático. Intenta batir el record mundial de salto sobre autobuses! P.V.P. CASSETTE: 1.600 pts.

### BRAXX BLUFF.



Lánzate a explorar mundos desconocidos con los Centuriones. En cada lugar que aterrices te aguardan peligros y desastres inesperados. P.V.P. CASSETTE: 1.6000 pts.

### HOCKEY.



¡Pon a prueba tus reflejos, rapidez y precisión compitiendo en un emocionante partido de hockey sobre hielo, contra un amigo o contra tu AMSTRAD. P.V.P. CASSETTE: 1.600 pts. DISCO: 2.500 pts.

### ASALTO.



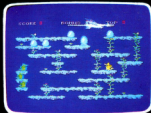
Si te gusta la emoción de la guerra, desarrolla toda una batalla aeronaval. Tus enemigos no te darán cuartel y no olvidas lo más importante: ¡ataca! P.V.P. CASSETTE: 1.600 pts.

### AIRWOLF.



Cinco científicos están retenidos en las profundidades de una base. ¿Puedes rescatarlos pilotando el inigualable y sofisticado helicóptero? ¡Destruye los copas de control de la defensa! P.V.P. CASSETTE: 1.600 pts. DISCO: 2.500 pts.

### DRAGONS.



Apasíonate en una inquietante aventura aniquilando dragones. Camina, trepa y salta para salvar tu vida de los dragones rojos. P.V.P. CASSETTE: 1.600 pts.

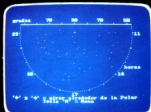
### GRAND PRIX RALLY II.



Participa en el mundo de los Rallies con circuitos en tres dimensiones. Atraviesa hielo, lluvia, desiertos, niebla y evita los choques con tus competidores. Sorpréndete creando tus propios recorridos. P.V.P. CASSETTE: 1.600 pts. DISCO: 2.500 pts.



### CONSTELACIONES.



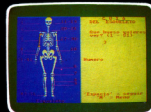
Observa las más importantes estrellas del universo en su posición real (grados y horas). Descubre sus nombres y aprende a identificarlos y conocerlos. P.V.P. CASSETTE: 1.600 pts.

### ESTIMATOR RACER.



Desarrolla tu capacidad de cálculo numérica. Conduce tu coche por el carril, en una carrera contra reloj, con la respuesta matemática más aproximada. 4 niveles de dificultad. P.V.P. DISCO: 2.500 pts.

### EL CUERPO HUMANO. EL ESQUELETO.



¿Cuántos nombres de huesos de nuestro esqueleto eres capaz de recordar? Aprende y diviértete con este excelente programa educativo. P.V.P. CASSETTE: 1.600 pts.

# AMSTRAD

ESPAÑA