

**MICROHOBBY**

AÑO I N.º 11

# AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

*Semanal*

AÑO I N.º 11

**150 Ptas.**

Canarios 160 ptas.

**DISCO LASER  
PARA  
AMSTRAD**

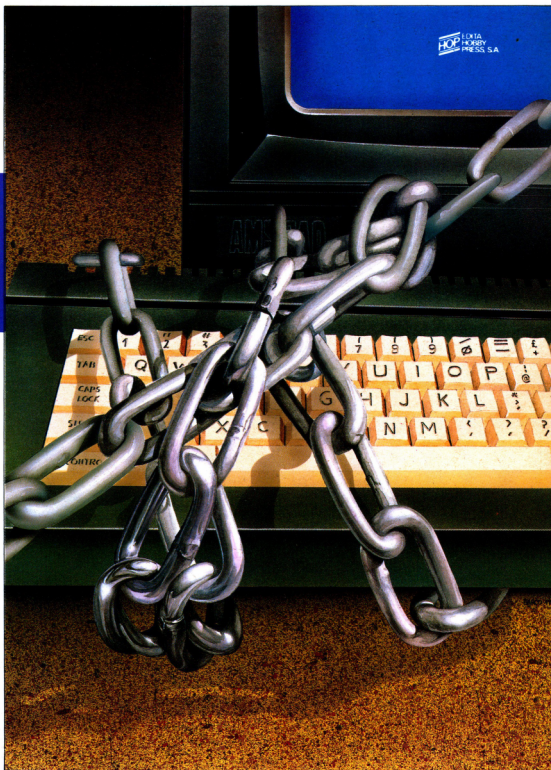
**PRESENTACION  
OFICIAL DEL  
PCW8256 Y  
DEL CPC6128**

**A PARTIR  
DE AHORA,  
EL AMSTRAD  
YA TIENE  
SPRITES.**

**AMSCARD:  
BASE DE DATOS  
EN DISCO COMPATIBLE  
CON TODOS LOS  
AMSTRAD**

**SOFTWARE**

**La aventura continúa.  
La guerra de las galaxias  
con Starstrike.**



# COMPUTIQUE

*Te da más*

**AMSTRAD**  
ESPANA

GARANTIA

**64.900 Ptas.**

Amstrad 464 f.  
verde



## Al comprar tu Amstrad te regalamos

- Estuche con ocho programas originales

- Fruit Machine
- Procesador texto
- Almirante Graf
- Oh Mummy
- Plaga Galáctica
- Amsdraw
- Laberinto Sultan
- Animal, Vegetal, Mineral

- Un estupendo libro de Basic

- Guía de referencia del programador
- y además te obsequiamos con un curso de introducción al Basic.

**Venta a plazos hasta 36 meses.**

Servimos tiendas  
Tel: 227 91 99

Nuevo Amstrad CPC6128



**COMPUTIQUE**

Embajadores, 90 28012 Madrid Tfn. 2270980

# AMSTRAD

## sumario

Año 1 • Número 11 • 12 al 18 de Noviembre de 1985  
150 plas. (sobretasa Canarias, 10 plas.)

**Director Editorial**  
José I. Gómez-Centurió  
**Director Ejecutivo**  
Victor Prieto  
**Subdirector**  
José María Díaz  
**Redactor Jefe**  
Marta García

**Diseño**  
José Flores  
**Colaboradores**  
Francisco Portalo  
Pedro Sudón  
Miguel Sepúlveda  
Francisco Martín  
Jesús Alonso  
Pedro S. Pérez  
Amalio Gómez  
Juan J. Martínez

**Secretaría Redacción**

Carmen Santamaría

**Fotografía**

Carlos Candel

Javier Martínez

**Portada**

J. Igual

**Ilustradores**

Javier Igual, J. Pons, F. L. Frontón, J. Septien, Pejo, J. J. Mora, Luigi Pérez

**Edita**

HOBBY PRESS S.A.

**Presidente**

María Andrino

**Consejero Delegado**

José I. Gómez-Centurió

**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 Grana, s/n

Polígono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 488 HOPR

**Dto. Circulación**

Carlos Peropadre

**Distribución**

Coedis, S. A. Valencia, 245

Barcelona

**Impreme**

ROTEDEC, S. A. C/ta. de Urén,

Km. 12,450 (MADRID)

**Fotocomposición**

Novacom, S.A.

Nicolás Morales, 38-40

**Fotomecánica**

GRGF

Ezequiel Salama, 16

**Depósito Legal:**

M-28468-1985

**Derechos exclusivos**

de la revista

**COMPUTING with**

**the AMSTRAD**

Representante para Argentina, Chile,

Uruguay y Paraguay, Cto.

Americano de Ediciones, S.R.L. Sud

América 1.537, Tel.: 21 24 64. 1209

BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace

necesariamente solidario de las

opiniones vertidas por sus

colaboradores en los artículos

firmados. Reservados todos los

derechos.

Se solicitará control OJD

## 5 Primera plana

Ha llegado el Compact Disc para nuestros ordenadores. El lenguaje C ya está disponible en cinta/discos.

## Primeros 6 pasos

Todas las funciones y comandos relacionados con el tratamiento de cadenas expuestos de forma sencilla y comprensible para el principiante.



## 11 Indescomp

Se presenta oficialmente la nueva gama de productos Indescomp: los Amstrad CPC6128, PCW8256 y el Spectravideo X'Press.

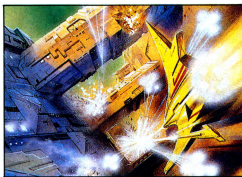
## Análisis 14

Un completo ejemplo de cómo usar y programar las teclas de función del Amstrad, aplicado a tener disponibles, pulsando una sola tecla, las palabras clave más importantes del Basic.

## ProgramAcción 15

Los sprites son unos gráficos muy especiales capaces de desplazarse pixel a pixel por la pantalla sin alterar el fondo de la pantalla por el que se mueven.

El Amstrad Basic permite crearlos fácilmente mediante el uso de interrupciones (el comando EVERY).



## 18 Mr. Joystick

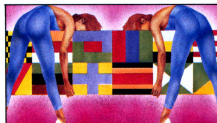
Skywalker, Darth Vader, Obi-Uan-Kenobi, nombres famosos y vivos en tu Amstrad con el programa Starstrike.

## Serie oro 22

Pocos programas son tan conocidos, y tan útiles, como las bases de datos. Amstrad es una implementación de esta idea escrita íntegramente en Basic, que maneja simultáneamente ficheros de lectura y escritura, y es compatible con cualquier Amstrad siempre que esté dotado de unidad de disco.

## 28 Código Máquina

Estudiamos las características y condicionantes del resto de los flags del Z80.



# THE DEVILS CROWN



3.300 pts.  
cassette  
4.300 pts.  
disco

## THE DEVILS CROWN EL MEJOR JUEGO DE AVENTURAS Y ACCION REALIZADO PARA AMSTRAD



SERMA

RECORTA Y ENVIA ESTE CUPON A: ♦ SERMA. BRAVO MURILLO, 377 - 3.º A.  
28020 MADRID TLFs: 733 73 11 - 733 74 64

TITULO	CANTIDAD DISCO	CANTIDAD CASSETTE	NOMBRE Y APELLIDOS:
THE DEVILS CROWN			
CALLE:	N.º:	POBLACION:	
PROVINCIA:	CODIGO POSTAL:		
FORMA DE PAGO: ENVIO TALON BANCARIO <input type="checkbox"/> CONTRA REEMBOLSO <input type="checkbox"/>			

## EL MAS PODEROSO

**T**odo el mundo sabe lo que significan las siglas IBM en el mundo de la informática, a cualquier nivel imaginable.

Pero no todos conocen otro nombre mágico que vive y prospera a la sombra de IBM, y que, sin embargo, es una de las principales bases de la multinacional americana se ha apoyado en los últimos años.

Nos referimos a Intel, la empresa que tuvo la suerte de ver su procesador 8088 dentro del IBM PC.

Desde entonces han pasado muchas cosas; los competidores fabricantes de chips entraron en liza ofreciendo al consumidor microprocesadores más baratos y potentes que el de Intel, de manera cada vez más notoria.

Por supuesto que Intel reaccionó lanzando nuevas cosas, como el 80286, el procesador que constituye el corazón del nuevo IBM AT (*de Advanced Technology*), pero ahora ha dado el salto definitivo creando al sucesor de éste: el Intel 80386, que ha sido llamado, con razón, el microprocesador más poderoso del mundo.

Es de 32 bits, y puede aplicarse igualmente a ordenadores personales, telecomunicaciones y robots industriales. Es dos veces más rápido, como mínimo, que el de sus competidores y puede ejecutar 4 millones de órdenes por segundo.

Más de 30 empresas en USA, incluidas IBM, ATT, Sperry y otras, han decidido adoptarlo para sus próximos modelos.

Excuso decir que, de nuevo, IBM se llevará la parte del león en cuanto a trato privilegiado por parte de Intel.



## SOFTWARE JAPONES

Gracias a un acuerdo entre la casa inglesa de soft Ocean y la japonesa Konami, los mejores juegos tipo arcade de esta última compañía serán adaptados al **Amstrad** en breve plazo.

Los primeros en ser «**transplantados**» serán los famosísimos **Hypersports** y **Tennis** de Konami.

Otros nombres de juegos en cartera son: **Yie Air**, **Kung-Fú**, **Hyper Rally** y **Konami's golf**.

Ocean comercializará estos programas en Europa usando el nombre de **Imagine**.

## Primera plana LENGUAJE C. CORTESIA DE HISOFT

**U**na buena noticia para los programadores. Ya existe un compilador para el lenguaje C desarrollado por Hisoft, creadora, entre otras cosas, del Hisoft Pascal y el Hisoft DEVPAC (*ensamblador/desensamblador*).

El lenguaje C es unánimemente reconocido como el rey para la programación y desarrollo de sistemas operativos; se caracteriza por su extrema concisión (*frente a la verbosidad de lenguajes como Pascal o Ada*) y rapidez (*para juegos*), permitiendo al programador experimentado hacer prácticamente cualquier cosa que se le ocurra, tareas que hasta hace muy poco tiempo sólo podían abordarse mediante el lenguaje ensamblador (código máquina).

El C cuenta además con la enorme ventaja de ser un idioma específicamente creado para ser portable de un ordenador a otro casi sin cambios.

Al parecer, el compilador es una implementación plena del C standard, mejorado con herramientas tales como un depurador interactivo para el principiante.

Hisoft también entrega una guía al C de 150 páginas con el programa, disponible en cinta y disco.

## ¿DISCO LASER PARA AMSTRAD?

**A**mstrad no se ha pronunciado oficialmente sobre la posibilidad de dotar a sus máquinas con un sistema de almacenamiento masivo de datos basado en la tecnología láser, comúnmente denominada «**Compact Disc**».

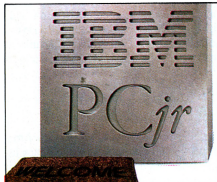
En una reciente feria informática y electrónica, Philips presentó un sistema de este tipo, y William Poel se sabe que lo ha visto y lo ha encontrado «**muy interesante**», suponemos

que por su precio, 450 libras (*ciento y poco mil ptas.*).

Philips ha «**confesado**» que, efectivamente, ha tenido conversaciones con **Amstrad** respecto a este tema.

Se desconoce todavía el alcance de las negociaciones. Por lo visto poca gente experta en medios de almacenamiento masivos, duda de que los discos láser, capaces de atesorar Miles de megabytes en un «**cartucho**» de 12 cm, sustituirán a todas las demás alternativas en un plazo entre 10 y 15 años.

No estaría mal tener uno pronto para nuestro **Amstrad**, ¿verdad?



# CADENAS

**En esta ocasión vamos a ver las variables literales o cadenas y examinaremos algunos comandos Basic para crearlas y manejarlas.**

**R**ecordará que las cadenas literales son las variables cuyo nombre termina con el signo del dólar, \$. Están compuestas por grupos de letras, números, signos de puntuación y espacios, todo ello englobado en un mismo nombre.

Trabajemos con ejemplos. Podemos almacenar la palabra GATOS en la variable literal «cosa\$» usando la siguiente instrucción de asignación:

```
LET cosa$ = «GATOS»
```

y después:

```
PRINT cosa$  
resultará que la palabra:  
GATOS  
aparece en pantalla.
```

## Problema de sintaxis

Por supuesto que no es necesario la palabra clave LET, pero si necesitamos las comillas. Son los delimitadores que señalan el comienzo y el final de la cadena. Probemos a meter:

```
cosa$ = GATOS
```

y veamos que pasa.

Podemos tener números formando parte de la cadena como:

```
cifra$ = «1234»  
PRINT cifra$
```

y

```
número$ = «0567»  
PRINT número$
```

lo demuestran. De cualquier forma, tengamos en cuenta que son variables literales y, por tanto, no podemos hacer cálculos aritméticos con ellas. Probemos:

```
PRINT número$ * cifra$
```

o

```
PRINT cifra$ - número$
```

y veremos lo que decimos. El mensaje de error de incongruencia de tipos (*type mismatch*) quiere decirnos justamente esto: hemos utilizado erróneamente el tipo de variables.

## Sumar cadenas

Sabiendo que no podemos hacer cálculos matemáticos con las variables literales, solo hay un modo de manejarlos que visto, pare-

ce como una forma extraña de suma. Aquí es donde unimos o «concatenamos» dos cadenas. Para mostrar lo que estamos diciendo, vamos a establecer dos variables como:

```
primero$ = «primero»
```

```
segundo$ = «segundo»
```

y después escribimos:

```
unidos$ = primero$ + segundo$
```

El signo más es como el de la suma, pero ¿cómo podemos sumar los valores primero y segundo? Lo que ocurre ahora es que los dos aparecen juntos, como podemos ver si hacemos:

```
PRINT unidos$
```

y obtendremos:

```
primerosegundo
```

como resultado. Las dos variables literales han sido enlazadas una tras otra para formar una tercera cadena.

Ensayemos:

```
PRINT cifra$ + número$
```

y observemos el resultado. La respuesta a esta «suma» es completamente diferente a la solución de:

```
PRINT 1234 + 0567
```

La primera contestación del ordenador es una cadena concatenada, la segunda es un número. Es precisamente la misma diferencia que existe entre:

```
PRINT 2 + 2
```

y

```
PRINT «2» + «2»
```

Hagamos balance, en cuanto a lo que concierne a las variables literales, el signo más quiere decir «unir», no «sumar».

La concatenación de cadenas la encontramos ya en el Programa IX de nuestro último artículo que ahora repetimos como Programa I.

Miremos detenidamente la línea 70. Coge la variable literal «asterisco\$» y le suma —o mejor dicho, concatena— un asterisco. La cadena resultante se almacena otra vez en «asterisco\$», que ahora es un carácter más largo.

Como el bucle FOR... NEXT «gira», la longitud de «asterisco\$» crece en cada línea, resultando un triángulo de asteriscos.

Volvamos atrás, a nuestra cadena original. Si hemos reinicializado nuestro micro, volvemos a crear la variable con:

```
cosa$ = «GATOS»
```

Observemos que cuando hemos hecho:

```
PRINT cosa$
```

hemos obtenido como resultado GATOS y no «GATOS» como cabría esperar. Las comillas han desaparecido. La finalidad de las comillas es marcar el comienzo y el fin de la cadena, pero no son parte integrante de ella.



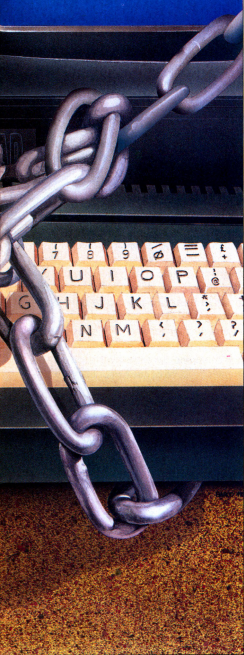
**¿Qué tenemos que hacer si queremos que aparezcan? ¿Podríamos conseguirlo colocando todo ello entre comillas?** Probando lo veremos. A menos que su Amstrad sea muy diferente al nuestro, creemos que descubrirá que:

```
cosa$ = ««GATOS»»
```

da como resultado un mensaje de error sintáctico. Sin embargo, no nos desesperemos, hay un modo de conseguirlo utilizando la función Basic CHR\$. Pero antes de intentarlo vamos a aprender algo acerca del llamado código Ascii.

## Código ASCII y CHR\$

Como probablemente ya sabe, su Amstrad trabaja con números. Todo lo que hace, desde destacar un enojado mensaje de error hasta un ataque alienígena a la Tierra



Así que, recapitulando, cada uno de los caracteres que vemos en la pantalla del **Amstrad** está representado por un número. Las letras mayúsculas tienen los códigos Ascii entre 65 y 90. Podemos convertir estos códigos en caracteres utilizando la función del Basic CHR\$ anteriormente mencionada. Probemos a escribir:

```
PRINT CHR$(65)
```

y tendremos en pantalla una A. Y no nos code de sorpresa encontrar que:

```
PRINT CHR$(66)
```

produzca una B o que:

```
PRINT CHR$(67)
```

haga una C. En cuanto que entendamos cómo la función CHR\$ convierte el código Ascii en una letra del alfabeto podemos seguir algo parecido a una obra maestra como es el Programa II.

Esperamos que se lleve un buen susto por el indiscriminado uso de PRINTS en este último programa. ¡Tranquilo! No tenemos que utilizar un PRINT separado para cada CHR\$, podemos encadenarlos todos juntos como en:

```
PRINT CHR$(71)CHR$(65)CHR$(84)CHR$(79)CHR$(83)
```

Ahora vemos de dónde viene el término cadena...

Hasta aquí, sólo hemos utilizado el código Ascii en un rango de 65 a 90. El Programa III utiliza un bucle FOR... NEXT para mostrar los caracteres cuyos códigos van del 32 al 126.

Aquí no tenemos sólo las letras mayúsculas, hay signos de puntuación, letras minúsculas, números y hasta un espacio 32. Todas estas cosas pueden formar parte de una cadena.

Así que utilizando CHR\$ y el código Ascii oportuno podemos crear cualquier cadena. En efecto, el **Amstrad** tiene un juego completo de caracteres gráficos disponible utilizando el código Ascii. Podemos verlo si cambiamos la cifra del final de la línea 20 a 255.

De todas formas, por el momento, veamos las letras mayúsculas que realiza el Programa IV.

Cada vez que «gira» el bucle FOR... NEXT. «Mayúsculas» incrementa su valor, oscilando entre 65 y 90. El resultado es que el CHR\$ de la línea 30 escribe el alfabeto completo y ordenado usando letras mayúsculas.

## Ahora en mayúsculas

El Programa V hace exactamente lo mismo pero escrito de una forma más técnica.

Aquí después que «desplazamiento» ha tomado el valor 64 en la línea 20, la variable de control del bucle «letra» se extiende del 1 al 26. La línea 40 se asegura del valor actual de «letra» añadiéndole el valor de «desplazamiento» para producir un código Ascii CHR\$ para que lo procese.

Este código oscilará entre 65, cuando «desplazamiento» es 1, hasta 90 cuando «desplazamiento» es 26 y así aparecerán las

# Primeros pasos

letras mayúsculas. Pero si es el mismo resultado que en el Programa III. ¿por qué molestarnos en teclearlo otra vez?

La respuesta es que nos hemos encontrado con un bucle que va desde 1 a 26 para producir el alfabeto de una forma más comprensible que uno que va desde 65 a 90 para conseguir el mismo fin. También, vemos que fácil es hacer letras minúsculas usando el método de «desplazamiento».

Observemos como el Programa VI difiere del Programa V. Aquí, teniendo «desplazamiento» un valor de 96 asegura que los valores con los que trabaja CHR\$ varían en el camino entre 97 y 122.

Estos valores son los códigos Ascii de las letras minúsculas, por lo tanto lo que sale en pantalla es distinto. ¿Puede modificar el programa para que escriba los números de 0 a 9? El rango de los códigos va de 48 a 57.

## La función ASC...

Para librarnos de la necesidad de tener que mirar en la tabla el código Ascii de cada carácter, el Basic del **Amstrad** tiene una función muy útil, el oportunamente llamado ASC, que coge un carácter y devuelve su código Ascii. De este modo:

```
PRINT ASC («A»)
```

devuelve 65 mientras:

```
PRINT ASC («a»)
```

nos da 97. Podemos utilizar variables literales dentro de los paréntesis como:

```
interior$ = «xx»
```

```
PRINT ASC (interior$)
```

nos muestra. También ASC distingue claramente entre números y cadenas como lo demuestran los distintos resultados de:

```
PRINT ASC (7)
```

y

```
PRINT ASC («7»)
```

Pensemos que ASC trabaja sólo con la primera letra de la cadena. Es perfectamente válido hacer algo parecido a:

```
PRINT ASC («GATO»)
```

y sólo conseguiremos que nos devuelva el código de la primera letra. En otras palabras,

```
PRINT ASC («XYX»)
```

da el mismo resultado que:

```
PRINT ASC («X»)
```

la Y y la Z se olvidan.

Sin embargo, ASC es bastante más que únicamente un medio rápido de encontrar un có-

en un juego de arcade, lo hace con números. Incluso cuando tratamos con palabras como:

```
PRINT «GATOS»
```

lo hace con números. Cada carácter posee un código numérico.

El código para A es 65, mientras que el signo de interrogación se representa con el número 63. Todas las letras, número 0-9 y signos de puntuación tienen su código numérico listado en una tabla conocida como código Ascii.

El código Ascii viene respaldado por el American Code for Information Interchange, de ahí su nombre. La tabla 1 nos muestra un breve resumen de los códigos más frecuentes.

El juego completo de códigos está publicado en el Apéndice III del Manual del Usuario. No puede decirse que tenga una lectura fácil, pero hojándolo un poco nos dará una idea de cómo se diseñan.



digo Ascii. Podemos utilizarlo para depurar datos —detectando errores en los usuarios— como se aprecia en el Programa VII.

Como habrá descubierto si lo ha ejecutado —y si no lo ha hecho, debe hacerlo— el programa sólo acepta las letras mayúsculas. La línea 40 revisa el valor Ascii de «entrada\$». Sólo los valores comprendidos entre la serie 65-90 corresponden al alfabeto en mayúsculas, así que si ASC (entrada\$) está por debajo o por encima de estos valores, se mete un dato erróneo.

Esto es otra manera de detectar algún error a quien intente entrar en su programa.

### ... y el control de errores

El GOTO envía el programa de vuelta a la línea 20 para intentarlo otra vez. Solamente cuando el código Ascii de «entrada\$» está dentro del rango de las mayúsculas, el programa se ejecuta hasta conseguir el mensaje final.

No obstante, **¿no cree que el Programa VII podría ser un poco más denso?**

Después de todo, alguien podría introducir una p cuando se necesita una P. Lógicamente el micro señalaría su error —posiblemente se volvería loco de por vida intentan-

do encontrarle— **¿Por qué no tiene que hacerlo con el Amstrad?**

Lo único que hace es un desplazamiento de 32 para admitir los 32 caracteres entre las letras mayúsculas y sus compañeras las minúsculas. El Programa VIII es la demostración de cómo se hace.

Aquí el valor de Ascii de «entrada\$» está metido en «ascii». La línea 60 comprueba que «entrada\$» es cualquier letra mayúscula o minúscula. Si no pasa el filtrado el usuario lo intentará de nuevo.

En el momento en que el programa llega a la línea 80, «entrada\$» debe ser una u otro. Aquí se comprueba, y si son minúsculas —un código mayor que 90— se resta 32 para pasarlo al grupo de las mayúsculas. En efecto. ASC permite a nuestro Amstrad corregir errores humanos.

Sin embargo, esta forma es solamente un modo de solventar el problema y en algunos Basic la única que tiene. La Locomotive Basic tiene dos funciones UPPERS y LOWERS que hacen el trabajo bastante más fácil.

### Trabajando con funciones Basic

Cuando aplicamos UPPERS a una cadena, cambia todas las letras minúsculas a mayúsc-

culas. LOWERS, como puede suponer, hace exactamente lo contrario. Los números y signos de puntuación quedan sin cambiar. Al fin y al cabo, **¿cuál es la mayúscula de ??** Vamos a probar con ejemplos semejantes a:

```
PRINT UPPERS «hijkl»
PRINT LOWERS «ABC12def»
PRINT UPPERS «pqr$&1KL»
```

y conseguiremos comprenderlo pronto.

Podemos utilizar cadenas de variables literales intermedias entre dos paréntesis. Teclee-

```
mixto$ = «aaCCff%%99ON»
PRINT UPPERS (mixto$)
```

si tenemos alguna duda. El Programa IX nos enseña que UPPERS puede ser utilizado en una versión más eficiente del Programa VII.

Como podemos ver, las líneas 70 a 100 del programa antiguo se han reemplazado por una sola utilizando UPPERS. **¿Podemos alterar la línea 70 para que sólo aparezcan las letras minúsculas?**

Antes de abandonar el código Ascii queremos tratar brevemente de los valores de código comprendidos entre 0 y 31. Son muy diferentes de los otros que hemos utilizado hasta aquí.

### Códigos de control

Todos los códigos comprendidos entre 32 y 126 escriben un carácter cuando se usan con CHR\$( ). Los códigos desde el 0 al 31 no influyen en un juego de caracteres, sino que entran en el micro. Son los que se conocen como códigos de control y lo que se hace con ellos es dirigir el micro. Si probamos:

```
PRINT CHR$(12)
```

vemos, o mejor dicho, no vemos que ocurra nada. Así que tendremos que ver, o no, cuál puede ser la causa. 12 es el código de control que borra el texto de la pantalla. En efecto, es lo mismo que CLS.

Intentemos:

```
PRINT CHR$(7)
```

y escuchemos el dulce y tradicional sonido de una campana.

El capítulo IX del Manual del Usuario, contiene todo sobre los códigos de control y sus usos. Compruébelos todos e intente explicarle lo que ocurre.

Nos llaman la atención en particular los códigos 8, 9, 10 y 11 que mueven el cursor del texto retrocediendo, avanzando, bajando y subiéndolo un «carácter espacio» respectivamente. Podemos divertirnos mucho con ello. Intentemos explicar que pasa con:

```
PRINT «GATOS»CHR$(8)CHR$(32)
```

Podemos incorporarlo fácilmente dentro de variables literales sumando —o mejor dicho, concatenando— a una cadena normal. Veamos lo que queremos decir introduciendo: blanco\$ = «GATOS» + CHR\$(8) + CHR\$(8) + CHR\$(8)CHR\$(8) + CHR\$(32) + CHR\$(32) + CHR\$(32) + CHR\$(32) + CHR\$(32)

Tras esto, la cadena de variables blanco\$(32) contiene cinco caracteres, cinco controles de

Cód.	Carac.	Cód.	Carac.	Cód.	Carac.	Cód.	Carac.	Cód.	Carac.	Cód.	Carac.
33	!	49	1	65	A	81	Q	97	a	113	q
34	..	50	2	66	B	82	R	98	b	114	r
35	\$	51	3	67	C	83	S	99	c	115	s
36	%	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	95	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39		55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	>	76	L	92	]	108	l	124	
45	-	61	<	77	M	93	^	109	m	125	~
46	.	62	=	78	N	94	_	110	n	126	°
47	/	63	?	79	O	95	`	111	o		
48	0	64	@	80	P	96	%	112	p		

Tabla 1: códigos ASC II

código y cinco espacios. Ahora cuando hagamos:

```
PRINT blanco$
```

no veremos nada. Los cinco espacios blancos se sobreescriben en GATOS.

Los códigos de control son parecidos a todo lo demás en el **Amstrad**, se empieza a comprender con la práctica (ya sabe lo que queremos decir: debe trabajar, trabajar, trabajar...).

Todo lo que necesitamos por ahora es tener muy clara la idea de que los números o los códigos Ascii pueden representar caracteres.

Antes de dejar CHR\$ por completo. ¿**Recurda nuestro problema con los «GATOS»?** El código Ascii que tenemos a mano. Teclémos:

```
cosa$=CHR$(34)+«GATOS»+CHR$(34)
```

y si hacemos después:

```
PRINT cosa$
```

obtendremos el tan buscado:

```
«GATOS»
```

No debe sorprendernos ya que el código

Ascii para las comillas es 34. Así que, un inteligente uso de CHR\$ nos permite escribir caracteres de una forma que resulta imposible desde el teclado.

De la misma manera:

```
PRINT CHR$(224)
```

y

```
PRINT CHR$(225)
```

nos dan los símbolos que no se encuentran en ninguna tecla.

## La función LEN

Y, finalmente, ¿qué longitud tiene una parte de la cadena? Parece una pregunta tonta, pero no lo es. Como descubriremos en nuestro próximo encuentro, podemos separar nuestras cadenas en partes —es lo que conocemos como cadena cortada— y es importante conocer su longitud. Porque para esto el Basic tiene la función LEN.

Su uso no es difícil. Vamos a suponer que, por cualquier razón, necesitamos conocer la longitud de la cadena ABC utilizando nuestro **Amstrad**.

# Primeros pasos

Todo lo que tenemos que hacer es teclear:

```
PRINT LEN («ABC»)
```

y nos devuelve 3, de modo que ABC tiene tres caracteres de longitud. Esto apenas nos llama la atención, ¿no? Seamos más prácticos, queramos conocer la longitud de una variable literal que ha podido estar cambiando durante todo el tiempo que ha durado la ejecución de un programa. Establezcamos una cadena con:

```
suelección$=«cualquiera»
```

y

```
PRINT LEN (suelección$)
```

nos dirá el número de caracteres que contiene.

Como dijimos, el uso de LEN es bastante sencillo pero hay un par de casos especiales de los que tenemos que ocuparnos. La longitud de un espacio es 1, no 0 como se podría pensar. Si no lo convencemos, teclee:

```
PRINT LEN (« »)
```

y se convencerá por sí mismo. Recuerde, los espacios cuentan como un carácter, así que:

```
vacío$=«HOLA mamá»
```

```
PRINT LEN (vacío$)
```

da como respuesta 9, no 8.

Otro caso especial es el de la cadena nula, la que no contiene nada. Definamos una con:

```
nulo$=«»
```

y descubramos su longitud con:

```
PRINT LEN (nulo$)
```

Tiene sentido que la respuesta sea 0. Después de todo no contiene ningún carácter.

Aunque nos puede parecer un poco tonto hacer una cadena que no contenga nada, nos viene muy bien como la condición de control de una bucle WHILE... WEND cuando cortamos cadenas. De todo esto ya hablaremos. Le dejamos con el Programa X.

Esto es precisamente un «**depurador**» usando LEN para asegurar que la palabra tecleada tiene una longitud correcta.

La línea 20 coloca en «**entrada\$**» la cadena nula. No es rigurosamente necesario, ya que el **Amstrad** asume que una variable literal tiene el valor de una cadena nula mientras no se le diga lo contrario. De todas formas, es bueno en la práctica programar utilizando este tipo de asignaciones, haciendo que el listado sea más claro y, sobre todo, más fácil para su depuración.

Hasta la próxima semana, le dejamos que comprenda cómo trabaja el resto del programa y que anote sus problemas.

El programa funciona con 1234, pero esto no es una palabra, ¿**puede hacer algo para resolverlo?**

## PROGRAMAS

```
10 REM PROGRAMA I
20 CLS
30 asterisco$=""
40 FOR fila:=1 TO 10
50 LOCATE 11-fila,fila
60 PRINT asterisco$
70 asterisco$=asterisco$+"*"
80 NEXT fila
```

```
10 REM PROGRAMA II
20 PRINT CHR$(71);
30 PRINT CHR$(65);
40 PRINT CHR$(84);
50 PRINT CHR$(79);
60 PRINT CHR$(83);
```

```
10 REM PROGRAMA III
20 FOR ascii:=32 TO 126
30 PRINT CHR$(ascii); " ";
40 NEXT ascii
```

```
10 REM PROGRAMA IV
20 FOR mayusculas:=65 TO 90
30 PRINT CHR$(mayusculas); " ";
40 NEXT mayusculas
```

```
10 REM PROGRAMA V
20 desplazamiento:=4
30 FOR letra:=1 TO 26
40 PRINT CHR$(desplazamiento+letra)
+" ";
50 NEXT letra
```

```
10 REM PROGRAMA VI
20 desplazamiento:=96
30 FOR letra:=1 TO 26
40 PRINT CHR$(desplazamiento+letra)
+" ";
50 NEXT letra
```

```
10 REM PROGRAMA VII
20 PRINT "Teclee una letra mayuscul
a ";
30 INPUT entrada$
40 IF ASC(entrada$)<65 OR ASC(entrada$)>90 THEN CLS: PRINT "He dicho u
na letra mayuscula":PRINT:GOTO 20
50 PRINT"Bien hecho!"
```

```
10 REM PROGRAMA VIII
20 PRINT "Teclee una letra " ;
30 INPUT entrada$
40 ascii=ASC(entrada$)
50 REM Comprueba si la entrada esta
dentro del rango de las letras
60 IF ascii<65 OR ascii>122 OR (asc
ii>90 AND ascii<97) GOTO 20
70 REM Si es minuscula, la transfir
ma en mayuscula
80 IF ascii>90 THEN ascii=ascii-32
90 entrada$=CHR$(ascii)
100 PRINT entrada$
```

```
10 REM PROGRAMA IX
20 PRINT "Teclee una letra " ;
30 INPUT entrada$
40 ascii=ASC(entrada$)
50 REM Comprueba si la entrada esta
dentro del rango de las letras
60 IF ascii<65 OR ascii>122 OR (asc
ii>90 AND ascii<97) GOTO 20
70 PRINT LPPR$(entrada$)
```

```
10 REM PROGRAMA X
20 entrada$=""
30 WHILE LEN(entrada$)<4
40 PRINT "Teclee una palabra de cua
tro letras " ;
50 INPUT entrada$
60 WEND
70 PRINT entrada$
```

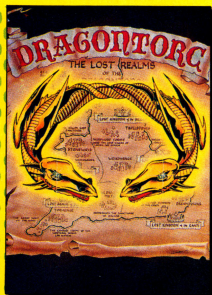
# ¡NUEVO!

## SIEMPRE LOS PRIMEROS EN TENER LO ULTIMO

# circulo de soft

MICROAMIGO S.A.

P.º de la Castellana, 268, 3.º C. 28046-MADRID.  
Tel.: (91) 733 25 00



### DRAGONTORC

Cerca de 200 pantallas con miles de objetos diferentes y más de cien personajes con animación en tres dimensiones, hacen que de este juego la revista inglesa Crash Micro haya llegado a decir «DragonTORC es lo mejor que hemos visto en juegos de acción y aventura».

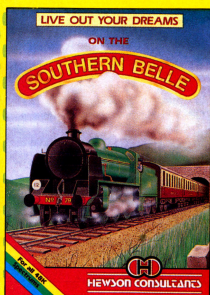
P.V.P.: 2.300 ptas. (Amstrad).  
Precio Socios C. de Soft: 2.070 ptas.  
P.V.P.: 1.900 ptas. (Spectrum).  
Precio Socios C. de Soft: 1.750 ptas.



### RAID OVER MOSCOW

Defiende a USA y Canadá del ataque nuclear que ha lanzado Rusia contra ellos. Con tu escuadrilla habrás de hacer un viaje lleno de peligros hasta llegar al mismísimo Kremlin y destruir las bases de lanzamiento soviéticas. Gráficos y acción sensacionales.

P.V.P.: 2.300 ptas. (Amstrad).  
Precio Socios C. de Soft: 2.070 ptas.  
P.V.P.: 2.100 ptas. (Spectrum).  
Precio Socios C. de Soft: 1.890 ptas.



### SOUTHERN BELLE

Siente la emoción de conducir una locomotora de vapor, através de un maravilloso recorrido desde Londres a Brighton, manejando la caldera, el silbato, atravesando túneles, etc. Estamos ante uno de los juegos más brillantes y originales aparecidos para ordenador.

P.V.P.: 2.300 ptas. (Amstrad).  
Precio Socios C. de Soft: 2.070 ptas.  
P.V.P.: 2.100 ptas. (Spectrum).  
Precio Socios C. de Soft: 1.890 ptas.

... Y LOS TRES PROGRAMAS POR SOLO **5.500 PTAS. (AMSTRAD)**  
**4.950 PTAS. (SPECTRUM)**

**¡HAZTE HOY MISMO SOCIO DEL CIRCULO DE SOFT!** Además de poder adquirir tus programas al mejor precio, recibirás información de forma periódica y gratuita, del mejor software que aparezca en el mercado.

**¿QUE HAY QUE HACER PARA SER SOCIO DEL CIRCULO DE SOFT?** Así de fácil: envíanos por correo tu nombre, dirección y modelo de ordenador, o bien, pide por teléfono o por correo tu primer programa. ¡Y entrarás a formar parte del CIRCULO DE SOFT de forma inmediata!

Si quiero ser SOCIO desde hoy mismo del CIRCULO DE SOFT y recibir periódicamente información de novedades de software, así como beneficiarme desde hoy mismo de los precios reducidos reservados a los SOCIOS y de sus Ofertas Especiales. El ser SOCIO no me obliga a compra alguna.

Si prefieres formalizar tu compra por teléfono puedes hacerlo llamando al (91) 733 25 00. ¡NO SE COBRAN LOS GASTOS DE ENVIO POR CORREO!!

TITULO	P.V.P.	ORDENADOR
_____	_____	_____
_____	_____	_____
_____	_____	_____

Contrarrembolso   
  Giro Postal   
  Taldn adjunto a Microamigo, S.A.   
  Tarjeta VISA n.º \_\_\_\_\_ Fecha caducidad \_\_\_\_\_

Nombre \_\_\_\_\_ Apellidos \_\_\_\_\_ Teléfono \_\_\_\_\_ Edad \_\_\_\_\_

Domicilio \_\_\_\_\_

Localidad \_\_\_\_\_ C.P. \_\_\_\_\_ Provincia \_\_\_\_\_

# PRESENTACION DE LOS NUEVOS PRODUCTOS AMSTRAD

**Indescomp presentó en el Meliá Castilla el CPC6128, el PCW8256 y el SpectraVideo X'PRESS (lease exprés) dentro de su nueva gama de productos. La estrella, sin duda, el PCW, por su relación prestaciones/precio. Según declaraciones del presidente de Indescomp, Amstrad es número 1 en Europa.**

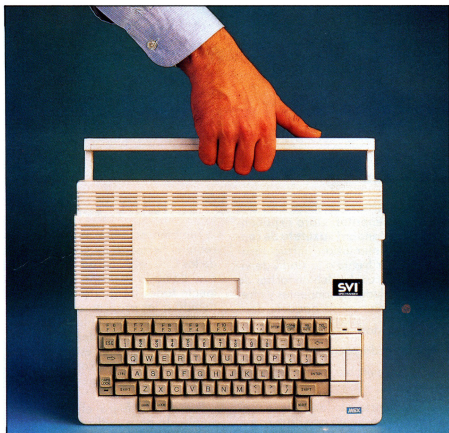
# E

l día 16 de octubre, el mundo del byte se vistió de largo para tomar parte en uno de los rituales más antiguos y de más solera dentro de los círculos comerciales, informáticos o no: la presentación por una compañía de sus nuevos productos.

El escenario para la obra fue escogido respetando al máximo las más puras tradiciones que definen (*¿encasillan?*) este tipo de magnos eventos: el hotel Meliá Castilla.

Tras acertar, en el primer intento, con la sala donde se celebraba una convención de agricultura y ganadería o algo parecido, y escapar rápidamente de allí (*el cocktail aún no estaba servido*), dimos con el salón Calatrava donde Indescomp presentaba su nueva gama de productos.

La concurrencia comenzaba a hacer acto de presencia en la sala, que se encontraba medio vacía.



MSX SPECTRAVIDEO SUJ-738 X'PRESS

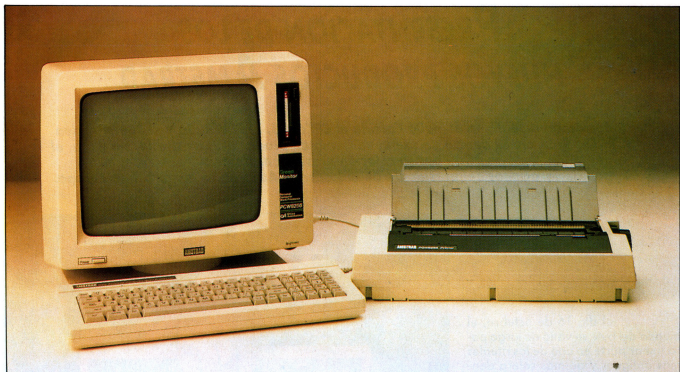
## MICROMANIA. Sólo para adictos

**HOBBY PRESS, S.A.**

**Editamos  
para gente inquieta.**

Una revista con  
marcha para los  
que necesitan  
saber **TODO**  
sobre  
ordenadores.





**AMSTRAD CPC 6128**

En la puerta, una mesa con dos atractivas señoritas, claro exponente del tipo standard «**azafata de convención**», intentaban infructuosamente hacer frente a la riada de visitantes a los cuales debían tachar de una lista, saludarlos sonrientes y ofrecerles un voluminoso paquete de:

- a) Publicidad
- b) Notas de prensa
- c) y el inevitable póster de regalo (*enorme*),

cuyo transporte a lo largo de la presentación constituyó una cruz considerable, soportada en aras de la precisión informativa y de la previosa semiausencia de papeleras.

Pasado el protocolo de admisión, cada uno, tan rápidamente como le fue posible, se dirigió a las hileras de mesas donde se encontraban los nuevos ordenadores, haciendo gala de sus posibilidades y ejecutando diferentes programas de gestión o entretenimiento, según la filosofía de cada uno.

### El disco integrado

Allí estaban el nuevo modelo de Spectraideo, el X'Press, un MSX portable dotado con una unidad de disco integrada de 3,5 pulgadas y capaz de funcionar bajo el sistema operativo CP/M 2.2, además de los sistemas operativos propios de los ordenadores MSX.

### Software, software

También ocupaba un sitio preferente el hermano mayor de la serie de los CPC, el **Amstrad CPC6128**, el cual llamó la atención de prácticamente todos los que estábamos allí debido a los programas de demostración que ofrecía: entre ellos yo calificaría de «cum laude» a una serie de dibujos, tentado estoy de decir cuadros, compuestos con un sorprendente acierto en la elección de formas y colores.

Fuera cual fuera el instrumento empleado para realizarlos, un programa diseñador de pantallas o un digitalizador de imagen o lo que sea, las posibilidades gráficas de este ordenador y la imaginación del artista nos dejaron asombrados. Increíble.

Otro programa que tuvimos ocasión de admirar era uno de ajedrez, el **Cyrus II**, creo, inspirado el programa de ajedrez de Psion para el Sinclair QL. La presentación del tablero es tridimensional, y el movimiento de las piezas se realiza con una suavidad encantadora. La ilusión de realidad está bastante lograda.

Respecto a los programas de gestión, pudimos echarle una ojeada a una contabilidad, bastante atractiva a juzgar por el menú principal, y un curioso programa de planificación de proyectos, que usaba como método de análisis la vía del camino crítico.

La estrella de la presentación, sin

duda, era el **PCW8256**, analizado por **AMSTRAD SEMANAL** en el número anterior. Había unos cuantos ejecutando el procesador de textos **Locscript**, en versión castellana, y mostrando en pantalla e impresora todas las posibilidades de este programa.

Otras máquinas del mismo tipo mostraban el **CP/M Plus**, también en castellano, y una hoja de cálculo.

El nuevo **Spectraideo** tal vez resultó un poco oscurecido por el **PCW**, pero es una máquina con un diseño de lo más atractivo que hemos visto nunca. Se le podía ver ejecutando programas de demostración, a base de sprites y gráficos, y algún que otro de gestión para demostrar la ventaja del disco.

A estas alturas, la sala estaba atestada de gente y del murmullo de las conversaciones de diferentes grupos. Empezaron a aparecer algunos vasos en las manos, y entre sorbo y sorbo podían escucharse los comentarios típicos de estas ocasiones, suavemente.

Justo cuando el ambiente estaba tomando la forma de un bautizo o algo así, una repentina urgencia de tomar asiento en las sillas dispuestas al efecto atravesó la sala como un látigo, y arrullados por el rozar de zapatos contra la moqueta nos dispusimos a escuchar oficialmente lo que casi todos los presentes sabíamos ya: las características de los nuevos ordenadores.

## Oficialmente

Al estrado, porque había un estrado, subió una delegación de Indescomp encabezada por su presidente, José Luis Domínguez.

Tras pedir silencio y saludar a los presentes, habló de las nuevas máquinas por este orden: Spectravideo, CPC6128 y PCW8256. Nuestros lectores ya conocen de que van los nuevos Amstrad, por lo que les ahorraremos repetir los detalles.

Domínguez destacó del Spectravideo su doble uso lúdico y profesional, gracias a la unidad de disco incorporada y al CP/M. También indicó que el precio de la máquina, 99.000 pesetas batía a sus más directos competidores en toda la línea, lo cual efectivamente es cierto (por ahora).

Quedó por aclarar la espinosa cuestión de la disponibilidad del software; como todo el mundo sabe, se ha repetido hasta la saciedad, hay miles de programas que corren bajo CP/M: bien. Pero, ¿cuántos existen en formato de 3,5 pulgadas? ¿Cuántos existirán en un plazo razonable?

Esta misma pregunta podría aplicarse a los dos Amstrad. Pero Domínguez comentó que Indescomp está haciendo un esfuerzo gigantesco para dotar a sus ordenadores de software en disco y que tiene disponibles ya más de 100 programas. En cuanto al PCW8256 es razonable darle tiempo al tiempo, porque prácticamente acaba de nacer. No obstante, el programa Locoscript cubre las necesidades básicas para las que fue diseñado el equipo.



AMSTRAD PCW 8256

El presidente de Indescomp añadió que Amstrad es un número 1 en ventas en casi la totalidad de Europa: España, Francia, Alemania e Inglaterra. Bastante quiso decir.

La defunción del CPC664 fue confirmada por Indescomp en una nota de prensa que se nos facilitó al entrar en el «salón de actos», y durante la charla.

Domínguez aseguró que las rumoreadas ampliaciones de memoria para el 664 ya existen y están prácticamente terminadas (hasta 256 K), pero no comentó la fecha de comercialización de este producto en nuestro país ni el precio estimado del mismo.

Y eso fue todo. Oficialmente, han nacido tres nuevos ordenadores.

NOVEDAD

## Solamente lo mejor



2.300 pts.



1.700 pts.



1.900 pts.



1.900 pts.



1.900 pts.



2.100 pts.



2.100 pts.

PIELOS POR CORREO.

# COMPTIQUE

Embajadores N.º 90  
Madrid 28012. Telf.: 227 09 80

# ¡FANTASTICO!



Si quieres un ordenador de «una pieza» piensa en el AMSTRAD CPC 464. Tendrás un ordenador de una vez por todas. Gracias a sus 64K RAM y 32K ROM y a sus casi ilimitadas posibilidades de crecimiento, tienes garantizado que el ordenador CPC 464 no se te quedará pequeño.

## COMPLETO

Además, gracias a su monitor (color o fósforo verde) de alta resolución (hasta 640 x 200 pixels direccionados individualmente) y a su unidad de cassette incorporada al teclado, podrás disfrutar de tu AMSTRAD de una manera independiente, prescindiendo del televisor y del radiocassette de tu casa (a veces tan solicitados).

## ¿Y QUE ME DICES DE LOS PROGRAMAS?

Actualmente ya hay cientos de ellos disponibles en España. Sin olvidar que son varias las revistas dedicadas sólo a AMSTRAD y que el número de libros y periféricos del CPC 464 crecen día a día, potenciando así la creatividad de tu ordenador personal.

## CARACTERISTICAS TECNICAS

- Microprocesador Z80 (4MHz).
- Memoria de 64K RAM y 32K ROM.
- Gráficos de alta resolución de hasta 640 por 200 pixels direccionables individualmente.
- Unidad de cassette incorporada en el teclado.
- Monitor color o fósforo verde incluido en el Sistema.
- Texto en pantalla de 20.40 y 80 columnas por 25 líneas.
- LOCOMOTIVE BASIC ampliado.
- Paleta de 27 colores y efectos de «flash».
- Teclado profesional tipo QWERTY con bloque numérico y teclas para cursor independientes.
- Salida Centronics paralela.
- Lector de discos de 3" (180K por cara) opcional (con CP/M y Dr. LOGO incluidos junto a la unidad de disco).
- Manuales en castellano.

Al comprar tu ordenador CPC 464, AMSTRAD ESPAÑA te obsequia con 8 cassettes de programas y el libro «Guía de Referencia BASIC para el programador».

Exige la **GARANTIA AMSTRAD ESPAÑA ÚNICA VALIDA PARA ACCEDER AL SERVICIO TÉCNICO OFICIAL.**

## PRECIO:

- 66.900 pts.  
(monitor fósforo verde)
- 95.900 pts.  
(monitor color)

*¡¡ Increíble !!*

# AMSTRAD

ESPAÑA

GARANTIA INDESCOMP

Avd. del Mediterráneo, 9 - 28007 Madrid Tels. 433 45 48 - 433 48 76 - Telex 47660 FAX - 4332450

# SPRITES EN EL AMSTRAD

**Una de las facilidades más impresionantes del Amstrad es la posibilidad de manejar las interrupciones desde Basic. Las aplicaciones de esta técnica abarcan un espectro muy amplio, desde la simulación de multitarea hasta la programación de excitantes juegos tipo arcade. En este último caso, hay una palabra clave que, desde ahora comprenderemos plenamente: los sprites.**

**E**n sprite (en español espíritu, normalmente traducido por «duende» y cosas así) es un carácter que puede moverse libremente por la pantalla, en general pixel a pixel, independientemente de otros caracteres y sin importar la tarea que el programa esté haciendo al mismo tiempo.

Puede salir y/o entrar por cualquier parte de la pantalla y pasar por delante o por detrás de las imágenes que existan en ella, sin borrarlas o modificarlas en forma alguna.

Este milagro, pues lo parece, puede conseguirse recurriendo al hardware del ordenador, al software o a ambos.

Resulta bastante claro que un sprite es casi casi lo ideal para un programa de juegos, basado en la animación.

Desgraciadamente, Amstrad Basic no los incorpora, aunque, como veremos, pueden simularse convincentemente algunas de sus particularidades.

## Creación de sprites

**¿Cuál sería el mejor método para hacerlo?** Obviamente el código máquina sería la solución idónea, porque la velocidad de este lenguaje permitiría mover simultáneamente gran cantidad de sprites.

Sin embargo, el lenguaje máquina tiene unos cuantos peros, como por ejemplo que resulta de difícil comprensión y es mucho más lento de escribir, consume más tiempo, que un lenguaje de alto nivel.

Surge la pregunta: ¿se pueden simular sprites en Basic?

## ProgramAcción

Bueno, si nos contentamos con uno o dos, la respuesta es sí.

Podemos usar las interrupciones desde Basic para llamar, cada cierto tiempo especificado por nosotros, a una subrutina que hará lo siguiente:

a) imprimirá en pantalla el sprite en la posición del cursor gráfico.

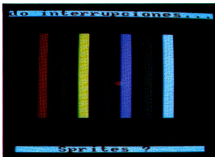
b) realizará la operación lógica conocida como «OR exclusivo», en Basic XOR contra lo que quiera que se encuentre en el fondo sobre el que el sprite se desplaza, asegurándonos así que permanecerá inalterado cuando el sprite sea borrado y ocupe una nueva posición.

## La operación lógica XOR

En el CPC464, la opción XOR se habilita mediante la impresión de un carácter de control (el 664 y 6128 posee sus propios comandos para ello, pero se recomienda usar éste):  
PRINT CHR\$(23);CHR\$(1)

## Cómo trabaja el programa número 2

LINEAS	FUNCION
20	Asigna el modo de pantalla y el borde.
30-35	Definición de los colores de tinta.
60-110	Imprime las barras de color.
120	Posición y dirección de comienzo.
130	Define color.
140	Habilita la impresión de gráficos en XOR.
150	Habilita la rutina de interrupción.
160-170	Mensajes.
180-250	Desplaza (scroll) los mensajes en sentidos opuestos por la pantalla.
290	Une los cursores de texto y gráficos.
300-310	Nuevas coordenadas de impresión.
320	Borra la bola.
330	Imprime la nueva bola en la nueva posición.
340	«Separa» los cursores y actualiza la posición.





A partir de ahora cualquier punto que se imprima en la pantalla será calculado realizando un XOR del valor de PEN del punto del fondo con el nuevo PEN del sprite.

La orden:

```
PRINT CHR$(23);CHR$(0)
```

desactiva el modo XOR.

Para comprender todo esto y ser capaces de predecir el color resultante de la operación XOR, es necesario recurrir un poco a los números binarios. Nuestro **Amstrad** podrá echarnos una mano, en más, demostrará ser un auténtico experto en la materia.

El MODE 0, con 16 colores, realmente es el único útil para nosotros, por lo que vamos a concentrarnos en él.

Los valores de PEN se codifican de 0 a 15, y es binario, esto implica que necesitamos 4 Bits (*no bytes*) para codificarlos.

Para verlos todos, basta con pedirle cortésmente al **Amstrad** que los imprima, mediante:

```
PRINT BIN$(numero, 4)
```

donde, naturalmente, número abarca el rango 0-15.

Ahora bien, para averiguar de una vez por todas qué hace la ope-

ración XOR, podemos decir al **Amstrad** que resuelva:

```
PRINT 0 XOR 0 ---+ DA 0
PRINT 0 XOR 1 ---+ DA 1
PRINT 1 XOR 0 ---+ DA 1
PRINT 1 XOR 1 ---+ DA 0
```

Acabamos, mediante sentencias PRINT, de construir la Tabla de Verdad de XOR. En efecto, los resultados obtenidos son igualmente válidos si imaginamos que los unos y ceros del comando PRINT corresponden a bits de un byte.

### ¿Cuánto vale, por ejemplo, 8 XOR 5?

(en binario)

8 .....	1000
5 .....	0101
8 XOR 5 .....	1101

esto es 8 XOR 5 da 13, como se puede comprobar mediante la instrucción PRINT &X1101.

De modo que si escribimos sobre un punto cuyo valor de PEN sea 8, otro que posea un valor de PEN de 5, obtenemos PEN 13 (por esto, entre otras cosas, usamos el MODE 0).

La inmensa ventaja del XOR es que, si ahora en el punto que posee

PEN 13, dibujamos otro encima con PEN 8, es decir 13 XOR 8:

13 .....	1101
8 .....	1000
13 XOR 8 .....	0101

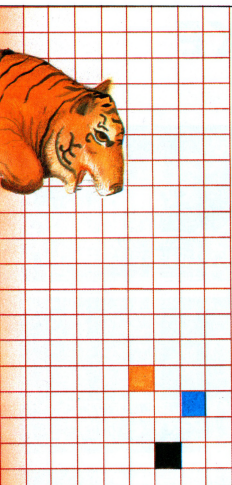
obtenemos 5, el color de fondo original restaurado.

Lo más duro del trabajo ya está hecho, porque podemos prever el color resultante al desplazar un sprite por pantalla y sabemos, también, cómo dejar el fondo que atraviese dicho sprite inalterado. Esto último, si se realiza a velocidad suficiente, nos permite dar al observador la sensación de profundidad, e incluso de volumen, muy fácilmente.

### Mezclar textos y gráficos

La orden Basic TAG permite que caracteres, texto, se impriman en la pantalla en la posición marcada por el cursor de gráficos, usando comandos gráficos, también, para posicionarlos y colorearlos.

Seleccionando cuidadosamente PEN e INK, podemos conseguir el efecto de que un gráfico pasa por delante o por detrás de otros gráficos (*profundidad y volumen*).



Atención a esto: si una pelota se imprime usando PEN 8 encima de un objeto anteriormente impreso con PEN 5, sabemos que el resultado es PEN 13 (en XOR, claro).

Y a esto también: si ahora a PEN 13 se le asigna el mismo color de sprite a PEN 5, la bola estará en pantalla, pero no se verá.

Por último, si asignamos a PEN 13, después, el mismo color que a PEN 8, la bola aparecerá mágicamente por delante del objeto del fondo.

Y esto es todo.

## Los programas

El programa 1 muestra cómo mover una pelota «por delante y por detrás» de una serie de barras verticales.

Primero se asignan los colores, luego se dibujan las barras con números de PEN de 1 a 7. Se activa el XOR, el cursor gráfico y entra el bucle principal de dibujo.

Antes hemos dicho que los sprites deben moverse independientemente de cualquier tarea que el ordenador esté realizando en ese momento.

Para ello, el **Amstrad** tiene disponibles cuatro temporizadores que nos habilitan para hacer que una determinada rutina se ejecute cada cierto tiempo.

Este método se conoce como «**manejo por interrupción**», y lo que sucede es precisamente eso, el programa principal se ve interrumpido por la rutina «**interrupción**», se detiene, la rutina se ejecuta y el principal sigue desde el punto exacto donde se quedó antes.

Si la velocidad con que se realiza este complejo proceso es lo suficientemente rápida, como de hecho ocurre, el observador piensa que más de una tarea se está ejecutando al mismo tiempo.

Por tanto, si escribimos la rutina que mueve el sprite en la línea 1000, por ejemplo, y damos al Basic la orden:

```
EVERY 10 GOSUB 1000
```

dicha rutina se ejecutará 5 veces por segundo, no importa lo que el **Amstrad** esté haciendo (sería aconsejable reparar el manual en los capítulos en los que se explica el funcionamiento de los comandos **EVERY**, **AFTER** y **REMAIN**).

El retardo (el número que va después del comando **EVERY**), puede ser cualquiera, permitiéndonos escoger a nuestro antojo la velocidad del sprite.

El programa 2 muestra la completa interacción de una rutina principal y otra que se ejecuta por interrupción.

## Muy pronto, ¡en máquina!

El lector se habrá percatado de que hemos empezado hablando de sprites y, por necesidad, nos hemos introducido en el mundo mucho más general de las interrupciones.

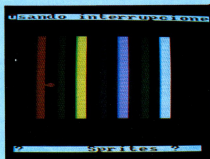
A más de uno este artículo le habrá sabido a poco, y se preguntará como manipular las interrupciones, los «**sucesos**» (events) en el manual del firmware, desde lenguaje máquina.

Este será precisamente el tema de un próximo artículo, que estructurará y relacionará lo que aquí se ha dicho de una forma mucho más útil y general, permitiéndonos aplicar las técnicas de manejo de interrupciones desde lenguaje máquina a un sprite o a lo que se nos ocurra.

Hasta entonces, pensamos que lo expuesto constituye material suficiente de aprendizaje y experimentación.

## PROGRAMAS

```
10 REM Programa...1
20 MODE 0:BORDER 0
30 FOR i%=0 TO 15
40 READ j%:INK i%,j%
50 NEXT
60 FOR i%=1 TO 7
70 PAPER i%:LOCATE 2i%+2,5
80 FOR j%=1 TO 15
90 PRINT " ";CHR$(8);CHR$(10);
100 NEXT
110 NEXT
120 PRINT CHR$(23);CHR$(1)
130 TAG:PLOT -10,-10,8
140 WHILE 1
150 FOR x%=50 TO 440 STEP 4
160 MOVE x%,200:PRINT CHR$(23);
170 FOR delay=0 TO 50:NEXT
180 MOVE x%,200:PRINT CHR$(23);
190 NEXT
200 WEND
210 DATA 0,6,9,24,1,8,10,26
220 DATA 7,6,7,24,7,8,7,26
```



```
10 REM Programa...2
20 MODE 0:BORDER 0
30 FOR i%=0 TO 15
40 READ j%:INK i%,j%
50 NEXT
60 FOR i%=1 TO 7
70 PAPER i%:LOCATE 2i%+2,5
80 FOR j%=1 TO 15
90 PRINT " ";CHR$(8);CHR$(10);
100 NEXT
110 NEXT
120 x%=32;ystep%=8;y%=200;ystep%=0
130 PLOT -10,-10,8
140 PRINT CHR$(23);CHR$(1)
150 EVERY 10,3 GOSUB 290
160 a$="Movemos la bola usando interrupciones...";b$=a$
170 c$="Sprites 2";d$=c$
180 WHILE INKEY("<")=" "
190 IF LEN(d$)<20 THEN b$=b$+a$
200 LOCATE 1,1:PRINT LEFT$(b$,20)
210 b$=MID$(b$,2)
220 IF LEN(d$)<20 THEN d$=c$+d$
230 LOCATE 2,25:PRINT RIGHT$(d$,18)
240 d$=LEFT$(d$,LEN(d$)-1)
250 WEND
260 END
270 :
280 REM Subrutina de Interrupción
290 TAG
300 x%=(x%+ystep%) MOD 640
310 y%=(y%+ystep%) MOD 400
320 MOVE x%,y%:PRINT CHR$(23);
330 MOVE x%,y%:PRINT CHR$(23);
340 TAGOFF:x%=x%+y%*y%:RETURN
350 :
360 DATA 0,6,9,24,1,8,10,26
370 DATA 7,6,7,24,7,8,7,26
```



**P**ara que tu dedo no realice el trabajo duro, M.H. AMSTRAD lo hace por ti. Todos los lectores que incluyan este logotipo se encuentran en tu disposición en un cassette mensual. ¡solicítalo!

# STARSTRIKE

**En los confines de una lejana galaxia, tiene lugar una terrible contienda, las naves enemigas defienden su estrella de combate con denodados esfuerzos, los láser defensivos intentan detener nuestro ataque suicida. Misión: destruir la Estrella de la Muerte.**

**E**l imperio está amenazado por el inmenso poder de una nave colossal, capaz de asolar planetas enteros con sus terribles armas.

La Estrella de la Muerte, domina bajo el yugo del miedo a toda la galaxia, aplastando cualquier insubordinación, solamente con su proximidad; su fuerza y poderío son tan descomunales que cualquier planeta que la divisa teme por su existencia.

En este ambiente de miedo, existen pocos valientes capaces de desafiar el poder de la temible estrella.

Solamente Obi-Uan-kenobi, caballero Yedi conoce su secreto y está en posición de enfrentarse al poder del mal que nadie puede vencer.

Obi-Uan, maestro de guerreros Yedi, instruyó al terrible Darth Vader en el dominio de La Fuerza. Un caballero Yedi en posesión de la fuerza que está por encima de la ambición y el poder y su puesto se encuentran al lado de la justicia y el bien.



Pero Darth Vader sucumbió ante la cara oscura de la fuerza, entrando en el mundo del mal, convirtiéndose gracias a sus poderes en el lugarteniente del emperador el cual le ha otorgado el mando de la temible Estrella de la Muerte.

Sky Walker, es un joven guerrero Yedi, entrenado también por el gran maestro Obi-Uan, está en posesión de la fuerza y es piloto de combate; su instructor le ha elegido como predilecto y le confía el secreto de Darth Vader.

El ambiente de rebelión se extiende por todo el imperio y una fuerza de combate se dispone a intentar el asalto final a la poderosa estrella.

Sky Walker es uno de los pilotos elegidos. En este momento comienza nuestra aventura, estamos a los mandos de nuestro caza intergaláctico.

El primer escollo son las naves enemigas que aparecen en oleadas y disparan sobre nosotros. Podemos detener los disparos con nuestro fuego y debemos destruir el mayor número posible de naves.

Superada esta primera prueba, pasamos al planeta, en el que debemos hacer fuego sobre zonas de energía radiante.

En sucesivas fases, nos encontramos con columnas de energía que crecen delante de nuestra vista disparando láseres cuando su altura ha llegado al máximo.

Por fin entramos en la Estrella de la Muerte, estamos en uno de los corredores, obstáculos a distintas alturas interceptan nuestro paso por ellos, los láseres intentan alcanzarnos mientras destruimos las escotillas de fuego.

Si salimos del túnel, llegaremos al



COMPATIBLE  
CPC 464  
CPC 664  
CPC 6128

sistema central de reactores, un disparo certero es suficiente para destruirlo, si lo conseguimos habremos salvado a la rebelión de las fuerzas imperiales.

Starstrike, es simplemente fantástico, los efectos en tres dimensiones son los mejores logrados en la historia, la acción es trepidante, la velocidad se siente.



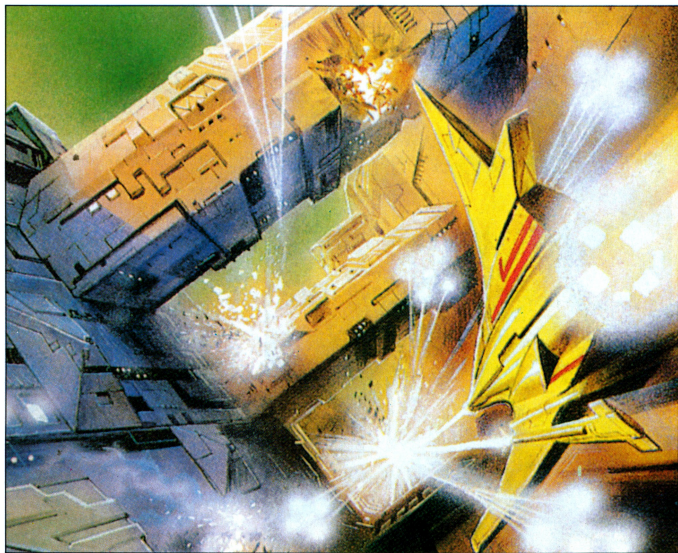
## Mr. Joystick

Los objetos se aproximan y se alejan como si se tratase de la más pura realidad, la sensación de perspectiva está completamente lograda.

El juego es completamente apasionante, solamente tenemos que preocuparnos de aniquilar todo aquello que aparezca delante de nuestro punto de mira.

Programas como éste llevan con orgullo el distintivo de tres dimensiones, no otros que proclaman a voz en grito tres dimensiones y luego nos encontramos con verdaderos adeseos, en los que las tres dimensiones son de verdadera risa.

Un juego para los amantes del Joystick y la acción.



# SUPER TOOLKIT

## Análisis

**A la hora de elaborar nuestros programas, debemos contar con una herramienta que nos evite tener que estar tecleando constantemente comandos como: RUN, LIST, EDIT, etc.**

Comando	Tecla
RUN	F0
GOTO	F1
LIST	F2
EDIT	F3
AUTO	F4
RENUM	F5
CLS	F6
MODE 0	F7
MODE 1	F8
MODE 2	F9

Análisis en esta ocasión, pretende facilitar la labor de programar con Super Toolkit, una herramienta que contiene los comandos más usados en las labores de creación, edición y corrección de programas.

La utilización de éste nos permite con un toque de tecla, la ejecución de los comandos más útiles para trabajar con listados y comprobar el funcionamiento de un programa.

Todos las funciones se obtienen pulsando la correspondiente tecla de función + [CONTROL].

De todos los comandos: RUN, LIST, RENUM, CLS, MODE 0, MODE 1, MODE 2, son de ejecución inmediata.

Mientras que GOTO, EDIT y AUTO, deben completarse con unos parámetros, lo que hace su uso mucho más flexible, adaptándose a nuestras necesidades.

Estos comandos, son de gran utilidad en la edición y corrección de programas, una vez cargado el programa en memoria y puesto en funcionamiento, el listado desaparece, permitiéndonos empezar nuestro programa en la línea que deseamos sin que nos molesten las 14 líneas del Super Toolkit.

**10-30** REMS que identifican nuestro programa.

**40** Define en la tecla de función número 0, el comando RUN, al cual se le suma CHR\$(13) equivalente a ENTER, lo que produce que al pulsar la tecla FO + [CONTROL], el programa que tengamos en memoria se ponga en marcha.

**50** Define el comando GOTO en la tecla de función F1, esta línea no contiene el CHR\$(13), por lo cual el comando anterior no se ejecutará automáticamente.

Simplemente aparecerá en pantalla el mensaje GOTO, seguido de un espacio en blanco, de manera que al teclear el número de línea deseado y pulsar ENTER, el programa transmite la ejecución a esa línea.

**60** La tecla F2, queda definida con el comando LIST, con ejecución automática (debiendo al CHR\$(13), lo que hace que al pulsar esa tecla de función + [CONTROL], el listado del programa se efectúe automáticamente.

**70** Dota a la tecla de función F3, de la orden EDIT, no ésta tiene ejecución automática, permitiéndonos espaciar el número de línea que deseamos editar.

**80** Define la sentencia AUTO, en la tecla de función F4, ésta también viene sin ejecución automática, de manera que podamos especificar los parámetros de número de línea e incremento.

Si pulsamos ENTER sin teclear ninguno de los parámetros, los número de línea se generarán a partir de 10, con incremento también de 10.

**90** Introduce el comando RENUM, en la F5, es de ejecución automática por lo que se efectuó una reenumeración de 10 en 10 a partir de la línea 10.

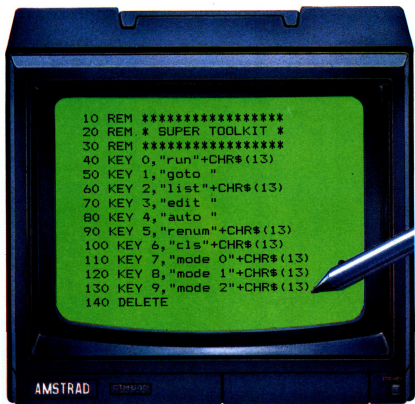
**100** Define el comando CLS, de ejecución automática en la tecla F6.

**110** MODE 0 de ejecución automática en F7.

**120** MODE 1 de ejecución automática en F8.

**130** MODE 2 de ejecución automática en F9.

**140** El DELETE al final del listado, tiene por objeto borrar el listado, de forma que éste no nos moleste para editar y elaborar nuestros listados.



# ¡SENSACIONAL!



Te presentamos un equipo sensacional: el **AMSTRAD CPC 6128**.

Con un sólo cable para enchufar a la red, el Sistema 6128 está listo para funcionar.

## JUEGA Y APRENDE CON EL 6128

Para jugar, el 6128 es un ordenador muy serio; gracias a sus cientos de programas disponibles, tienes aseguradas horas de entretenimiento. Y en el mundo de la enseñanza no es menos.

Gracias a sus sensacionales capacidades gráficas (paleta de 27 colores y hasta 640 x 200 PIXELS) y sonoras (3 voces y 8 octavas, altavoz interior y salida estéreo) el 6128 es una herramienta inguatable. Además, dentro del paquete de programas que se entrega con el sistema, está incluido el lenguaje educativo por excelencia: el **Dr. LOGO** de Digital Research. Y para profundizar en el lenguaje de la informática recuerda que el 6128 es el ordenador idóneo, ya que posee uno de los más rápidos y potentes BASIC —el **LOCOMOTIVE BASIC**—, así como otros muchos lenguajes de programación: **FORTH, PASCAL**, etc.

## TRABAJA CON EL 6128

Haz un sitio en tu negocio al 6128. Planifica presupuestos, lleva contabilidades, gestiona archivos, todo fácilmente gracias a su Sistema Operativo CP/M (en versiones 2.2 y Plus), que (como ya sabes) te permitirá acceder a la más extensa biblioteca de programas profesionales: bases de datos, procesadores de textos, hojas de cálculo, etc.

## CARACTERÍSTICAS TÉCNICAS

- 128K RAM y 48K ROM (incluye Locomotive BASIC y Sistema Operativo).
- Monitor: Color de 14" y fósforo verde de 12".
- Unidad de Disco 3" incorporada (180K por cara).
- Teclado profesional.
- Sistema Operativo: AMS-DOS, CP/M 2.2 y CP/M Plus.
- Salida para segunda unidad de disco y cassette externa.

## El CPC 6128 incluye en su suministro:

- Disco con Sistema Operativo CP/M 2.2 y lenguaje Dr. LOGO.
- Disco con Sistema Operativo CP/M Plus y utilidades.
- Disco con seis programas de obsequio.
- Manuales en castellano.
- **GARANTÍA AMSTRAD ESPAÑA ÚNICA VÁLIDA PARA ACCEDER AL SERVICIO TÉCNICO OFICIAL.**

## TODO POR:

- **109.500 ptas.**  
(monitor fósforo verde)
- **134.500 ptas.**  
(monitor color)

*¡¡Incredible!!*

# AMSTRAD

ESPAÑA

GARANTIA INDESCOMP

Avd. del Mediterráneo, 9 - 28007 Madrid Tels. 433 45 48 - 433 48 76 - Telex 47660 FAX - 4332450

# AMSCARD.V.1/85

**AMSCARD es un programa que, sin duda, le será de gran ayuda para ordenar cualquier cosa, con la única limitación que sus necesidades le impongan o su imaginación le permita. Haciendo uso de las facilidades del Amstrad Basic, podemos manejar hasta 500 fichas de 5 campos implementadas según la técnica de ficheros secuenciales.**

**C**

uando nos surgió la idea de publicar un programa especialmente diseñado para correr en un CPC apoyado por una unidad de disco, nos pareció una restricción lo suficientemente fuerte para que un buen número de lectores—usuarios del 464 se sintiera marginado ante su aparición en estas páginas. Sin embargo, la calidad del programa que presentamos, el incremento de usuarios del 464 que se deciden por una opción más profesional, ampliando su equipo con una unidad de disco (dos?), nos hicieron superar este hándicap inicial y lanzarnos a esta nueva aventura. La aventura del disco.

## DESCRIPCION GENERAL DE AMSCARD

Básicamente AMSCARD es lo que en el mundo informático se denomina una base de datos. La misión de este tipo de programas es la clasificación de un cierto número de datos agrupados en fichas. Cada uno de los datos que componen una ficha recibe el nombre de campo. En el caso concreto de AMSCARD dispondremos de cinco campos, es decir, podremos almacenar hasta cinco datos distintos por cada ficha, más otro campo principal que servirá para realizar las operaciones de búsqueda de una ficha. Cada campo recibe la denominación que deseemos (p. ej. «Teléfono»), salvo el campo principal que siempre recibe la denominación «Nombre». De este modo nosotros definiremos la estructura de cada ficha, en base al tipo de objetos que queramos clasificar. Gracias a esta posibilidad podremos ordenar claramente, desde sellos de correos hasta la lista de nuestros clientes.

COMPATIBLE  
CPC 664  
CPC 6128

Programa realizado por:

**J. J. MARTIN**

El lugar donde se guardan estos datos, se reparte entre la memoria de nuestro ordenador y el soporte de almacenamiento (el disco). En el disco coexisten tres ficheros denominados «CAMPOS», «DIRECT» y «ARCHIVO». Los dos primeros se vuelcan directamente en memoria nada más correr el programa. En ellos se almacenan la denominación de los campos, en el primero; y el primer campo, campo principal de cada ficha, en el segundo. De este modo, cuando solicitamos una ficha, para su lectura o modificación, podemos encontrarla de forma rápida, tengamos en cuenta que el rastreo en la memoria del ordenador es miles de veces más rápido que la búsqueda en un disco.

Una vez encontrado el campo principal de la ficha obtenemos el número que nos da la posición del resto de los campos en el disco. Podría plantearse por qué razón no tenemos, desde un principio, toda la información en memoria. La respuesta es sencillamente que la memoria de nuestro CPC es limitada, ni el potente 6128 podría abordar este problema con éxito. Tenga en cuenta que una ficha de cincuenta letras por campo, ocuparía alrededor de 1/3 de K (1.024 Kbytes). Teniendo en cuenta que disponemos de unos 30 K ibres, una vez cargado el programa, limitaríamos la capacidad del fichero a unas 90 fichas, en contraposición con las 500 fichas de 255 letras por campo que puede gestionar AMSCARD.

## CARGA DEL PROGRAMA

Una vez teclado el programa teclee RUN y pulse ENTER, no se preocupe de salvar el programa,

F. L. FRONTAN





pues éste está preparado para auto-salvarse. Una vez aparecido en pantalla el menú pulsemos para elegir la opción inicializar disco.

## INICIALIZAR DISCO

La pantalla se borrará y aparecerá la pregunta «**Campo1?**». Para comprender esto, describamos primeramente, aunque ya hemos hablado sobre ello, el formato de las fichas que van a integrar nuestro fichero.

Cada ficha se compone de un nombre, el cual será el único elemento válido para posteriores trabajos con la ficha. De la elección del elemento que se va a almacenar en este campo depende el 90 por 100 del éxito de nuestro fichero. A parte del nombre, cada ficha pone, como ya dijimos, de cinco campos redefinibles por el usuario. Que nadie se asuste la cosa es más sencilla de lo que las palabras pudieran hacernos pensar. Si nuestro fichero nos fuese a servir para hacer un archivo de nuestros amigos, en el nombre iría, evidentemente, el nombre. Los campos que definiríamos, por ejemplo, serían los siguientes:

- Campo 1: *Dirección.*
- Campo 2: *Código postal.*
- Campo 3: *Población.*
- Campo 4: *Teléfono.*
- Campo 5: *Fecha de cumpleaños.*

Llegados a esta altura, cualquiera se encontrará en disposición de responder a la pregunta que tenemos en pantalla. La respuesta será el nombre que queramos asignar al campo uno, una vez teclado, pulsamos ENTER y nos encontraremos con «**Campo 2?**», y así sucesivamente hasta que hallamos definido los cinco campos previstos para cada ficha. Terminada esta operación aparecerá el mensaje «**Inserte disco archivo**». Pondremos el disco en el cual queremos se encuentre nuestro archivo y pulsaremos cualquier tecla. En este disco, para una mayor comodidad, se encontrará también grabado AMSCARD. De forma que cuando queramos utilizar el archivo guardado en este disco teclaremos únicamente RUN «**AMSCARD**».

Finalizada esta opción, regresaremos de forma automática al menú principal y único del programa. En esta situación podremos acceder únicamente a tres opciones:

# Serie Oro

Archivar.  
Inicializar disco.  
Inicializar disco con campos.

Cualquier otra elección nos hará regresar al menú con el mensaje «**ARCHIVO VACIO**».

Pulsemos A y decidámonos a crear el archivo.

## ARCHIVAR

A partir de este momento, estamos en disposición de llenar nuestras propias fichas. Primero cargaremos el nombre, luego el campo 1, el campo 2, y así sucesivamente hasta 5. Naturalmente, en pantalla no nos aparecerá Campo 1, sino la definición que hayamos dado al campo número 1; dirección, si nos atenemos a nuestro ejemplo.

De esta forma iremos sucesivamente ampliando nuestro archivo. Cuando queramos finalizar esta opción contestaremos con M o m a la pregunta «**NOMBRE?**». De este modo el archivo se volcará en el disco, regresando después al menú. Para evitar el riesgo de sobrecarga de memoria o corte de luz, es una buena medida efectuar esta operación cada diez o quince fichas realizadas. Después para continuar archivando pulsáramos nuevamente A.

## BUSCAR/VISUALIZAR

Para acceder a esta opción pulsamos V. Se nos pedirá inmediatamente el nombre del titular de la tarjeta. Aquí nos encontramos con una de las potentes ventajas con que cuenta AMSCARD. El nombre no tendrá que ser completo, sino que podrá ser cualquier subcadena contenida en él. Por ejemplo si tenemos una ficha declarada a nombre de JULIAN RODRIGUEZ MARTIN. Nos bastará con teclear JULIAN o RODRIGUEZ o incluso MART. Siempre y cuando, claro está, estemos seguros de que no existe otra ficha a nombre de otro JULIAN u otro RODRIGUEZ u otro MARTIN o MARTINEZ. Incluso a la petición de nombre podríamos responder con un sim-

## LISTA DE VARIABLES

C:	Contiene el número de campo.
CAMP(S)(C):	Nombre del campo C.
ENCONT:	Número de la ficha solicitada en opción búsqueda/visualización.
HEND:	Última ficha almacenada en archivo más uno.
FICH(S)(n,m):	Campo número m de la ficha n.
MOT:	Número de ficha para modificar.
NAMES(N):	Nombre del titular de la ficha n.
NOMBRES:	Nombre del titular solicitada, para visualizar o modificación.
PUL(S):	Se utiliza para almacenar el valor de una tecla pulsada.

ple ENTER (*cadena vacía*), con la aparición indefectible de la primera ficha. Tenga en cuenta que la cadena vacía se encuentra contenida en cualquier cadena.

Una vez introducido el nombre, dos cosas son las que pueden suceder. Que la ficha no sea encontrada, lo que se nos indicará en la parte inferior de la pantalla, a la vez que se nos solicita otro nombre; o que sea encontrada, con lo que la ficha completa aparecerá en pantalla. Ahora si pulsamos M regresamos al menú, si pulsamos P obtendremos una copia por impresora y pulsando cualquier otra tecla, se nos volverá a interrogar sobre la ficha que queremos solicitar.

Ha de tener cuidado con la opción impresora (P), ya que si ésta no está conectada o se encuentra desenchufada, el programa se quedará colgado en una situación de espera. Pulse ESC dos veces y después ejecute AMSCARD con RUN. Esta operación puede hacerla con toda tranquilidad, pues no perderá ninguna información.

### MODIFICAR

Desde esta opción a la que se accede pulsando M, podremos modificar el contenido de una ficha. El sistema de petición es idéntico al descrito en la opción anterior.

Una vez encontrada la ficha se nos mostrará en pantalla, apareciendo a la derecha de nombre una flecha, ahora si pulsamos C, el nombre anterior quedará borrado y podremos introducir el nuevo. Si pulsásemos cualquier otra tecla se conservará el que tenemos y podremos pasar a modificar, sucesivamente, los campos del mismo modo. Tras cada modificación regresaremos automáticamente al menú. Para abortar la opción podremos hacerlo no modificando ninguno de los datos de la ficha que puede ser la primera pulsando ENTER.

### LISTAR ARCHIVO

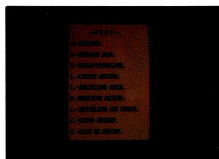
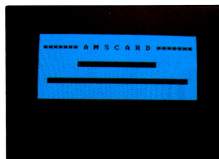
Por medio de esta opción iremos visualizando en pantalla las distintas fichas, desde la 1 hasta la última introducida. Tras la presentación de cada ficha deberemos pulsar una tecla. M para regresar al menú, P para obtener una copia por impresora y cualquier otra tecla para obtener la siguiente ficha.

### LIMPIAR ARCHIVO

Si por algún motivo el archivo hubiese dejado de interesarnos, podemos vaciarlo utilizando esta opción.

Dada su gravedad, se nos pedirá confirmación antes de realizar la limpieza. Sólo se vacía el archivo, por lo que podremos seguir trabajando con el mismo disco el cual contendrá los mismos campos.

Si usted hubiese borrado el archi-



vo y se diese cuenta que éste le interesaba, puede aún recuperarlo, tenga en cuenta que en el disco existe una copia (ARCHIVO.BAK). Cámbele de nombre desde AMSDOS o CP/M por ARCHIVO. No olvide poner un punto tras la palabra ARCHIVO.

### EFFECTUAR BAJA

La utilidad de esta opción está en dar de baja una ficha, permitiendo aprovechar al máximo la memoria suprimiendo fichas inútiles. Procure dar el nombre lo más detallado posible, incluso entero si fuese necesario, tenga en cuenta que puede borrarse una ficha que le interese.

Puede abortar esta opción respondiendo a la petición del nombre del titular de la tarjeta con M.

### INICIALIZAR CON CAMPOS

Esta opción es prácticamente idéntica a la descrita anteriormente, inicializa disco, con la diferencia que toma como campos los definidos ya. El uso de esta opción viene determinado ante la necesidad de tener el archivo lleno y desear continuar archivando. En previsión de esta necesidad, sería aconsejable que si el fichero va a ser muy amplio inicializar varios discos utilizando cada uno de ellos para un grupo de nombres que empiezen por unas letras determinadas. Por ejemplo desde la A a la M un disco y desde la N a la Z en otro disco, o en la otra cara del disco. Esto nos permitiría usar hasta 1.000 fichas.

### PARA TERMINAR

Y si después de todo lo dicho aún le queda alguna duda no lo piense, siértese ante su CPC y decídase a ordenar su biblioteca. El diseño de la ficha podría ser perfectamente el siguiente:

NOMBRE: Aquí iría el título del libro.  
 AUTOR:  
 EDITORIAL:  
 FECHA DE EDICION Y PAGINAS:  
 TEMA:  
 PRESTADO A:

Sin duda alguna su biblioteca se lo agradecerá. Que usted lo ordene bien.

## SIETE CONSEJOS ÚTILES PARA LA UTILIZACIÓN DE AMSCARD

1. Antes de realizar un archivo estudie cuidadosamente el diseño de la ficha, esto es, la denominación de los campos. Si le faltan campos puede utilizar uno para dos datos. Por ejemplo el Teléfono y el Código Postal pueden ir en el mismo, en el caso del archivo descrito anteriormente.

2. No insistiremos lo suficiente en cuanto a la elección del dato que vamos a guardar en el campo principal. Si utilizase este para guardar la dirección, cuando intente averiguar la fecha de cumpleaños de su querido y buen amigo Pepe, deberá meter el nombre de la calle donde vive, lo cual puede resultar bastante incómodo. Tampoco lo utilice si va repetir este dato en otra ficha, por ejemplo para guardar el autor de un libro, el siguiente libro del mismo autor no podría ser encontrado.

3. Si usted desea tener 10 archivos distintos habrá de tener cinco discos, en cada cara de un disco sólo puede haber un fichero, pues si no al utilizar uno inutilizará los otros.

4. En el caso de que no vaya a utilizar el disco para alguna utilidad de CP/M, formatee el disco con FORMAT D. Obtendrá más capacidad de almacenamiento. Naturalmente AMSCARD, puede convivir perfectamente con otros programas en el disco.

5. Cada campo, el principal incluido, admite un máximo de 255 caracteres. No obstante, procure no sobrepasar las tres líneas de texto por campo cuando elabore la ficha, si las sobrepasa el conjunto perderá en estética y comodidad.

6. El campo principal, NOMBRE, procure no llenarlo demasiado, como norma no lo cargue con más de 50 caracteres, a no ser que quiera arriesgarse a quedarse sin memoria. Tenga en cuenta que cada carácter ocupa un byte. Con lo que si tuviésemos 100 caracteres en el campo principal de cada ficha, necesitaríamos 100\*500 bytes, es decir, 50.000 bytes (50 K aprox.). Memoria con la cual no contamos, salvo, claro está, los envidiables usuarios del 6128. Utilice el resto de los campos a tal efecto, pues éstos no ocupan memoria al estar cargados en el disco.

7. Modificando la rutina impresora podremos realizar las direcciones de nuestra correspondencia, imprimiendo en pegatinas que existen comercializadas, a tal efecto, o en el mismo sobre. Obteniendo un resultado elegante y rápido. Por desgracia esto no puede estandarizarse en el programa debido a la infinidad de impresoras existentes, junto a sus diferencias de utilización.

```

10 REM=====
11 N C I D=====
20 ON ERROR GOTO 1390
30 INK 0,0:INK 1,16:BORDER 16:DISC
40 MODE 1:PAPER 1:CLS:WINDOW 1,6,36
6,17:PAPER 1,2:PEN 1,0:CLS:LOCAT
E 1,2,5:PRINT #1,"***** A M S C A
R D *****"PEN 0
50 LOC 0,0,319,0:DRAW 575,319:DRAW
575,129:DRAW 80,129:DRAW 80,319
60 LOCATE 8,14:PRINT "V 2/85 MICRO
HOBBY AMSTRAD":DEFINITE a-z
70 LOCATE 14,11:PRINT "POR J.J. MAR
TIN":LOCATE 7,18:PRINT STRINGS(31,2)
190 FOR n=1 TO 11:LOCATE 37,6:n:PRI
NT CHR$(207):NEXT
80 PEN 1:LOCATE 2,24
90 DIM fich$(500,5):DIM name$(500):
DIM camp$(5):OPENIN "DIRECT":INPUT
#9,HEND:FOR n=1 TO HEND:INPUT #9,na
me$(n):NEXT:CLOSEIN:OPENIN "CAMPOS"
:FOR c=1 TO 5:INPUT #9,camp$(c):NE
X:CLOSEIN
100 MODE 2:PEN #3,2:PAPER #3,1:PE
N 1:PAPER 1:WINDOW #1,19,61,23,25:PEN
#1,1:PAPER #1,2:CLS #1:WINDOW #2,2
2,52,3,23:PEN #2,2:PAPER #2,1:WIND
OW #1,80,3,22:BORDER 0
110 ON ERROR GOTO 0
120 REM=====M E N U=====
130 CLS
140 CLS#2: SOUND 1,300,15,15,2,2:LOC
ATE #2, 8,2:PRINT #2,"--M E N U--"
150 IF INKEY<>="" THEN 150
160 LOCATE #2,2,4:PRINT#2, "A.-ARCH
IVAR.#LOCATE #2,2,6:PRINT #2, "B.-
EFECTUAR BAJA.#LOCATE #2,2,8:PRINT
#2, "C.-BUSCAR/VISUALIZAR.#LOCATE
2,2,10:PRINT #2, "L.-LIMPIAR ARCHIVO
.#LOCATE #2,2,12:PRINT#2, "I.-INICI
ALIZAR DISCO.#
170 LOCATE #2,2,14:PRINT#2, "M.-MODI
FICAR ARCHIVO.#LOCATE#2,2,16:PRINT
#2, "N.-INICIALIZAR CON CAMPOS.#LOC
ATE #2,2,18:PRINT #2, "F.-LISTAR ARCH
IVO.#LOCATE#2,2,20:PRINT#2, "S.-SAL
IR DE AMSCARD.#LOCATE #2,2,6:PRINT
#2, "B.-EFECTUAR BAJA.#
180 PUL=UPPERS(INKEY$):IF PUL$=""
THEN GOTO 180
190 IF PUL$="A" THEN 290
200 IF PUL$="B" THEN 430
210 IF PUL$="C" THEN 1270
220 IF PUL$="I" THEN 590
230 IF PUL$="M" THEN 610
240 IF PUL$="L" THEN 750
250 IF PUL$="F" THEN 790
260 IF PUL$="S" THEN 990
270 IF PUL$="N" THEN 1160
280 GOTO 120
290 REM=====V I S U A L I Z A R=====
300 IF HEND=1 THEN GOTO 800
310 CLS#1
320 :CLS#3:LOCATE#3,14,9:INPUT #3,"
NOMBRE DEL TITULAR DE LA TARJETA":IN
STR$=FOR N=1 TO HEND:ENDONT=N:IF
INSTR(UPPER$(name$(N)),"UPPERS"(noab
r#3))<0 THEN 340
330 NEXT:CLS #1:LOCATE #1,2,2:PRINT
#1, "NOMBRE DESCONOCIDO, NO APAREC
E EN ARCHIVO":GOTO 320
340 CLS#1:OPENIN "archivo.v":FOR n=1
TO endont-1:FOR a=1 TO 5:INPUT #9,a
#NEXT:NEXT:FOR a=1 TO 5:INPUT #9,4
:ch$(c):NEXT:CLOSEIN:CLS
#3
350 PRINT#3,"-----
-----"
360 LOCATE#3,3,2:PRINT#3,"NOMBRE":
:PRINT#3,USING "B";NAME$(ENDONT):FO
R c=1 TO 5:LOCATE#3,3,c+2#3:PRINT
#3,camp$(c):USING"b";:fich$(enco
nt,c):NEXT:CLOSE:LOCATE #3,1,20
370 PRINT #3,"-----
-----"
380 PRINT #1,"-----"
390 PULSE P PARA
PULSE M PARA R
EGRESAR AL MENU",
OTRA TECLA
PARA CONTINUAR OPCION
390 IF INKEY<>="" THEN 390
400 PUL=UPPERS(INKEY$):IF PUL$=""
THEN 400
410 IF PUL$="P" THEN GOSUB 1240
420 IF PUL$="M" THEN 120 ELSE GOTO
310
430 REM ==A R C H I V O R=====
440 com=hend:IF HEND>499 THEN GOTO
1220
450 CLS#3
460 PRINT#3,"-----"
470 CLS#1:LOCATE #1,1,2:PRINT #1,"I
Nserte M PARA REGRESAR AL MENU."
480 INPUT#3, "NOMBRE":name$(HEND):I
F NAME$(HEND)="" OR NAME$(HEND)=""
THEN 520
490 CLS#1:FOR c=1 TO 5:PRINT#3,ca
mp$(c);";":INPUT#3, fich$(hend,c):IN
500 HEND=HEND+1:IF HEND>499 THEN G
TO 520
510 GOTO 450
520 CLS:LOCATE 20,12:PRINT "SALVAND
O NUEVO ARCHIVO, ESPERE"
530 OPENOUT "direct"
540 WRITE #9,hend:FOR n=1 TO hend:M
RITE #9,name$(n):NEXT nn
550 CLOSEOUT
560 OPENOUT "archivo.v:OPENIN "archi
vo.v":IN WHILE N<COMFOR a=1 TO 5:IN
PUT #9,a#;WRITE #9,a#;NEXT M:N=N+
1:WEND
570 LOCATE#3:FOR n=com TO hend:FOR a
=1 TO 5:WRITE #9,fich$(n,m):fich$(n
,m)";":NEXT M:NEXT:CLOSEOUT
580 GOTO 130
590 REM=====E D E O T=====
600 CLS:FOR c=1 TO 5:PRINT " C
AMPO"(c):INPUT camp$(c):PRINT:NE
XT
610 CLS:LOCATE 24,12:PRINT:INSERTE
DISCO PARA INICIALIZAR "
620 CLS#1:PRINT #1,"-----"
630 IF INKEY<>="" THEN 630
640 IF INKEY$="N" THEN 640
650 CLS:OPENOUT "CAMPOS":FOR c=1 TO
5:WRITE #9,camp$(c):NEXT:CLOSEOUT
660 SAVE "AMSCARD"
670 REM=====C R E A R L I M P I A
R=====
680 hend =1
690 CLS:LOCATE 14,12:PRINT "OPERAC
ION LIMPIAR EFECTUANDO, ESPERE."
700 OPENOUT "direct"
710 WRITE #9,hend,hend,hend
720 CLOSEOUT
730 OPENOUT "archivo.v":WRITE #9,FICH
#1,1:CLOSEOUT:CLS:CLS #1:GOTO 120
740 GOTO 120
750 CLS
760 LOCATE 24,12:PRINT "CONFIRMACIO
N ?":PRINT #1,"-----"
770 PUL=UPPERS(INKEY$):IF PUL$<>
"N" AND PUL$<>"N" THEN 770
780 IF PUL$="S" THEN GOTO 680 ELSE
GOTO 120
790 REM=====M O D I F I C A C I O N E
S=====
800 CLS#1:IF HEND=1 THEN LOCATE #1,
8,3:PRINT#1, "ARCHIVO VACIO.#:GOT
0 140

```



¡EXCLUSIVA MUNDIAL!

# YOUR COMPUTER

La Revista de ordenadores de mayor venta en toda Europa

**¡SE PUBLICA DESDE AHORA EN ESPAÑA, EN FORMA DE CASSETTE!**

Sí, ya está confirmada la sensacional noticia. Muy pronto estará en los quioscos de toda España una selección de los mejores juegos y utilidades publicados por la prestigiosa Revista británica «YOUR COMPUTER», editados en cassette de alta calidad y con instrucciones en castellano.

El prestigio alcanzado por Your Computer, tanto en Inglaterra como en España y otros países, se debe, de una forma muy especial, a la gran

calidad de los programas que publica, la mayor parte de ellos en Código Máquina, y con la utilización de rutinas y técnicas de programación muy depuradas.

Ahora, a un precio inmejorable, podéis tener acceso a estos programas, **evitandoos** la difícil tarea de **teclearlos** en vuestro ordenador.

¡Y **cada mes** estará en la calle una nueva cinta!

Si no encontras la cassette de «Your Computer» en tu quiosco o tienda de informática, solicítala a nuestras oficinas:

**SINTAX, S. A.**

«YOUR COMPUTER»

Paseo de la Castellana, 268  
28046 Madrid

Envía tus señas completas, teléfono y **marca de ordenador** e incluye **talón bancario**, o remite **Giro Postal** por el importe.

No te cobraremos gastos por el envío.

Si prefieres pagar **contra reembolso**, entonces incluye, junto a tu pedido, dos sellos de 50 ptas. cada uno para gastos de envío.

TAMBIEN DISPONIBLE  
PARA

**COMMODORE 64** y

**SPECTRUM 48, PLUS, 128**

# YOUR COMPUTER

EL CORAZON DE LA PRIMERA REVISTA EUROPEA DE ORDENADORES

**AMSTRAD**

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

**695.-**  
PTAS.

# SIGUIENDO CON EL FLAG

***Siempre hay más de una manera de hacer las cosas. Este viejo tópico puede aplicarse perfectamente al modo que un programa decide entre distintos caminos en lenguaje máquina. Puede usarse algo más que el Carry flag.***

La última vez nos dedicamos a estudiar el Carry flag y en particular a utilizarlo con la instrucción CP para crear bucles, menús, etc...

Sin embargo, el Carry flag no es el único flag del Z80: hay otros seis más. El que nos interesa en este momento es el flag de Cero.

Realmente el nombre de flag de Cero nos dice lo que hace, puesto que se activa cuando el «respuesta» a cualquier operación en código máquina es cero.

## Programas uno y dos

Por ejemplo, no sólo dejará un 0 en el registro A, sino que activará el flag de Cero al decirnos que ése es el resultado de la operación. Ambos dejarán cero en A y activarán el flag de Cero.

Resumiendo, si un ADD o SUB dejan cero en el registro A, el flag Cero se activa. Si el resultado no es cero, el flag Cero está desactivado.

Pensándolo bien, todo está dispuesto a la inversa: Cuando el resultado «es» cero se activa el flag Cero —a uno. Y cuando el resultado «no es» cero, se desactiva el flag Cero —a cero. ¡Uno cuando es cero, y cero cuando no lo es!

No nos preocupemos más, los verdaderos valores de los flags no nos interesan realmente. Todo lo que necesitamos conocer son las condiciones que los activan y desactivan. En este caso del flag Cero se activa cuando el resultado de la operación es cero y se desactiva cuando no lo es.

No todas las instrucciones afectan al flag Cero, pero pronto usaremos algunas que lo hacen.

```
LD A, &FF
ADD A, &01
RET
```

Programa 1

```
LD A, &DF
SUB &DF
RET
```

Programa 2

Probablemente recordará que teníamos un par de instrucciones de salto basadas en el Carry flag JP C y JP NC.

Bueno, pues tienen su correspondiente par de instrucciones de salto, implicando al flag Cero.

JP Z (opcode&CA)

realiza el salto si el flag Cero ha sido activado, y

JP NZ (opcode&C2)

lo hace cuando no lo está, o sea cuando está desactivado.

## Programa tres

Echemos un vistazo al Programa tres. No nos presentará ningún problema, ya nos lo habíamos encontrado dos veces más anteriormente.

Recordará que íbamos incrementando el registro A y saltando hacia atrás para imprimir el código Ascii correspondiente hasta que dábamos la vuelta al cuentakilómetros y así activar el Carry flag.

No obstante, el flag Cero se activará al mismo tiempo ya que el último ADD A, 1 suma 1 al 255 (&FF) en el registro A, dándonos cero.

## Programa cuatro

Esto quiere decir que podríamos reescribir el Programa tres con un JP NZ, &3002 en lugar del JP NC, &3002 —y esto es lo que hacemos en el Programa cuatro.

Aquí seguimos realizando el bucle hasta que el ADD A, 1 activa el flag Cero cuando A contiene &FF. Entonces salimos del bucle completamente y RET.

No nos quedemos con la idea de que los flags Carry y Cero están conectados de alguna manera. En el ejemplo anterior es precisamente una coincidencia que ambos se activen al mismo tiempo —esto no siempre es así.

## Programa cinco

Para probarlo, veamos el Programa cinco. Utilizaremos INC A (&3C) para incrementar A en 1 en lugar del ADD, 1. Otra vez es idéntico al Programa cuatro, pero un byte más corto.

Como verá, la idea es buena. Sin embargo, si, como en el Programa seis, reemplazamos el JP NZ con JP NC —que haría que los flags Carry y Cero estuvieran relacionados— ocurre un desastre.

La razón es que INC A activa o desactiva el flag Cero dependiendo del resultado del incremento de A a uno. Sin embargo, no afecta en absoluto al Carry flag.

Queremos decir que el Carry flag no se activa cuando A se incrementa de 255 a 0. Por lo tanto, el bucle continúa haciéndose y, efectivamente escribe CHR\$(1), CHR\$(2) y así sucesivamente.



DIRECCION	OPCODE	MNEMONICOS
3000	3E 20	LD A, &20
3002	CD 5A BB	CALL CharOut
3005	C6 01	ADD A, 1
3007	D2 02 30	JP NC, &3002
300A	C9	RET

Programa 3

# Código máquina



No obstante, los códigos Ascii por debajo de 32(&20) son códigos de control, que, como sabemos por el Manual del Usuario, producen algunos efectos bastante «sui generis».

En la práctica, nuestra pantalla se bloqueará: nuestro micro se ha «colgado». Sin embargo, no se ha dañado sólo hay que hacer un reset a la máquina para recuperar el control. Desafortunadamente, haciéndolo así, perderemos cualquier programa que esté en la memoria —en nuestro caso ha sido el Hexer, el cargador hexadecimal.

DIRECCION	OPCODE	MNEMONICOS
3000	3E 20	LD A, &20
3002	CD 5A BB	CALL CharOut
3005	C6 01	ADD A, 1
3007	C2 02 30	JP NZ, &3002
300A	C9	RET

Programa 4

## Programa seis

Nuestro consejo es que no ejecute el Programa seis hasta el final de la sesión, cuando esté preparado para parar de cualquier manera... Desde luego demuestra el hecho de que los flags Carry y Cero son independientes de una manera que no olvidaremos.

Realmente, a pesar del evidente error el Programa seis, posee una atractiva sutileza. Damos por supuesto que el Carry flag está desactivado cuando metemos el programa.

Piénselo. Si cuando ejecutamos el programa —a través de Hexer o directamente desde CALL &3000— el Carry flag ya estuviera activado, se quedaría así durante las tres primeras instrucciones.

Como hemos dicho, las instrucciones LDs no afectan al Carry flag, así que LD A, &20 dejará el Carry activado, lo mismo que al hacer la llamada CharOut. Y, como hemos mencionado, INC A no le afecta tampoco.

Lo cual quiere decir que cuando alcancemos el JP NC, no haremos el salto, puesto que Carry está activado desde el principio. Tendremos que sacar nuestra rutina por la vía del RET final, escribiendo precisamente un espacio (Ascii&20).

En la práctica esto no ocurre. Cuando entramos en la rutina, el Carry está desactivado, y permanece así mientras no le afecte ninguna instrucción. Por lo tanto, el JP NC se realiza siempre y produce un crash. Sin embargo, en el código máquina, es incorrecto suponer cual será el estado que tendrá un flag en particular. Hacerlo así provoca a menudo errores difíciles de encontrar en los programas.

Así que, en el Programa seis, no debemos esperar que se desactive el Carry, debemos hacerlo nosotros. Estaría bien que pudiéramos hacerlo con una sencilla instrucción al Z80. Desafortunadamente, no existen instrucciones semejantes.

DIRECCION	OPCODE	MNEMONICOS
3000	3E 20	LD A, &20
3002	CD 5A BB	CALL CharOut
3005	3C	INC A
3006	C2 02 30	JP NZ, &3002
3009	C9	RET

Programa 5

De todas formas, podemos desactivar el Carry con:

ADD A, O (C6 00)

Después de todo no podemos dar la vuelta al cuentakilómetros sumando cero a cualquier cosa ¿verdad? Y ADD afecta al Carry flag, así que en este caso debe ser desactivado. La ventaja es que el contenido del registro A permanece sin cambiar, piénsalo.

De este modo, el afecto ADD A, O, es que nos desactiva el Carry flag, sin afectar el contenido del registro A.

Bien, ya sabemos como desactivar el Carry flag cuando queramos, pero ¿cómo se activa?

Bueno pues para este caso hay una instrucción que nos viene como anillo al dedo SCF (opcode &37). SCF significa Set Carry Flag (activa el Carry flag).

DIRECCION	OPCODE	MNEMONICOS
3000	3E 20	LD A, &20
3002	CD 5A BB	CALL CharOut
3005	3C	INC A
3006	C2 02 30	JP NC, &3002
3009	C9	RET

Programa 6

## Programa siete

El Programa siete es un ejemplo trivial del uso de estas ideas.

Como podemos ver, tiene dos saltos condicionales que hace que se vuelva al comienzo del programa. Hemos utilizado ADD A, O antes del primero y SCF antes del segundo para manipular las cosas de forma que no se hagan los saltos.

Vamos rectos hacia CALL CharOut, que nos produce un sonido. Si conseguimos que alguno de nuestros saltos condicionales **salte** erróneamente, o si activamos cuando debemos desactivar, quedamos atascados en un bucle sin fin.

## Programa ocho

El Programa ocho muestra otro ejemplo más elaborado. Si lo conseguimos completamente lo veremos como un sendero demasiado tortuoso, debido a que por el modo en que hemos tratado el Carry flag, no aparece nada escrito.

**¿Podemos alterarlo de forma que aparezcan en pantalla dos asteriscos (código Ascii &2A)?**

Bueno, volviendo al flag Cero. La última vez vimos como la instrucción CP n, aunque no altera el registro A, activa y desactiva el Carry flag, como si se hubiese realizado un SUB.

Carry está activado si A < n.  
Carry está desactivado si A + n.  
No nos sorprenderá enterarnos de que CP n también afecta al flag Cero:

Si el byte del registro A es igual a n, se activa el flag Cero.

Si hay dos bytes diferentes, se desactiva el flag Cero. Es lo único que podemos esperar. Si los dos bytes tienen el mismo valor, cuando se hace el CP n simulando A-n, la respuesta deberá ser cero, así que se activa el flag Cero. Si, en otra pasada, los bytes cambian, la respuesta no es cero y el flag se desactiva.

## Programa nueve

Podemos usar esto para «exigir» la entrada de cierto carácter desde el teclado, como en el Programa nueve, donde esperamos un asterisco:

El programa trabaja comparando el carácter de entrada con el código Ascii de un asterisco. Si lo es, el flag Cero se activa e ignora el salto, lo escribirá y devolverá el control.

Si no es un asterisco, los códigos serán diferentes y el flag Cero se desactivará. Entonces el salto vuelve al principio y pedimos otro carácter desde el teclado.

Hasta ahora, cuando usábamos CP, hacíamos una comparación entre el registro A y el número especificado. Además, podemos comparar el registro A con cualquier otro de nuestros registros sencillos, que nos muestra la Tabla I. La anotación de esta forma de instrucción es CP r, donde r es un registro de ocho bits. Por ejemplo, CP B compara el byte del registro A con el que hay en B haciendo un falso A-B y activando el flag correspondiente; **el contenido de ambos registros queda sin cambiar.** Fíjese que el registro especificado se resta del registro A.

De nuevo, si A es igual o mayor que B, se desactiva el Carry. Si A es menor que B, se activa. Además, si los bytes de A y B son iguales, se activa el flag Cero, en otro caso está desactivado.

DIRECCION	OPCODE	MNEMONICOS
3000	3E 07	LD A, 7
3002	C6 00	ADD A, 0
3004	DA 00 30	JP C, &3000
3007	37	SCF
3008	D2 00 30	JP NC, &3000
300B	CD 5A BB	CALL CharOut
300E	C9	RET

Programa 7

DIRECCION	OPCODE	MNEMONICOS
3000	3E 2A	LD A, &2A
3002	C6 00	ADD A, 0
3004	D2 0A 30	JP NC, &300A
3007	CD 5A BB	CALL CharOut
300A	37	SCF
300B	DA 11 30	JP C, &3011
300E	CD 5A BB	CALL CharOut
3011	C9	RET

Programa 8

DIRECCION	OPCODE	MNEMONICOS
3000	CD 18 BB	CALL Charin
3003	FE 2A	CP &2A
3005	C2 00 30	JP NZ, &3000
3008	CD 5A BB	CALL CharOut
300B	C9	RET

Programa 9

DIRECCION	OPCODE	MNEMONICOS
3000	06 00	LD B, 0
3002	3E 2A	LD A, &2A
3004	CD 5A BB	CALL CharOut
3007	04	INC B
3008	3E 08	LD A, &08
300A	B8	CP B
300B	C2 02 30	JP NZ, &3002
300E	C9	RET

Programa 10

## Programa diez

Vamos a usar esta capacidad para comparar nuestro registro de ocho bits al hacer el Programa diez. Aquí nuestra tarea es escribir un número fijo de asteriscos-ocho en este caso.

Hemos utilizado el registro B como un contador. Inicialmente le damos un valor cero con LD B, 0 y entonces escribe un asterisco cargando A con &2A y llamando a CharOut. Incrementamos a continuación B, de este modo se mantiene al tanto del número de asteriscos escritos. Entonces cargamos A con 8, número total de asteriscos, y comparamos el registro B con A.

Si los dos registros no son iguales —o sea, si el flag Cero no está activado— no se han escrito todavía los ocho asteriscos así que saltaremos atrás a &3002, volviendo a cargar A con &2A, llamando a CharOut y así sucesivamente.

Si, en cambio, hemos alcanzado nuestro límite, se activa el flag Cero y no se realiza el salto, sencillamente devolvemos el control.

## Programa once

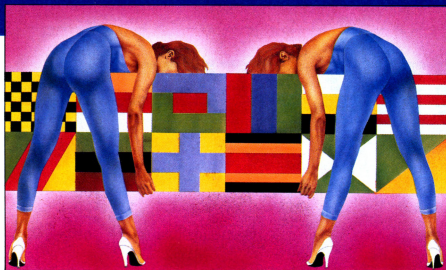
Realmente, el programa es bastante tortuoso. Hay maneras más eficaces de imprimir un número de asteriscos. Solamente lo hemos necesitado para introducir la instrucción CP r. El Programa once presenta una forma alternativa para hacerlo. El truco es la cuenta atrás desde el número que queremos usar DEC B.

Cuando alcancemos cero, se activa el flag Cero, y en cualquier otro caso saltamos atrás y escribimos otro asterisco. En efecto, B actúa como una primitiva variable de bucle o contador.

Podemos utilizar esta idea para crear bucles anidados usando el registro B para el bucle interno y otro para el externo.

## Programa doce

El Programa doce se basa en esta idea para sacar en pantalla un triángulo de asteriscos.



Vamos a hacer ocho líneas, la primera con ocho asteriscos, la segunda con siete y así sucesivamente. Cada vez que gire el bucle externo tendremos una línea de asteriscos—el bucle interno imprime el número de ellos que necesitamos en cada línea.

Usamos C para apuntar el número de líneas—de ahí el inicializarlo con LD C, &08— y B para contar el número de asteriscos. Piense que, después de haberlo utilizado como contador, C rebaja en una su valor cada vez que gira el bucle exterior donde también necesitamos hacérselo a B. Nuestra segunda instrucción, al principio del bucle exterior, es por lo tanto, LD B, C.

Si nos fijamos en el final del programa, veremos el fin del primer bucle:

```
3016      OD      DEC C
3017      C2 02 30  JP NZ, &3002
```

Regresa al principio del bucle exterior, y después de LD B, C cargamos el registro A con el código Ascii del asterisco (LD A, &2A). Entonces llegamos al lazo interior:

```
3005      CD 5A BB  CALL CharOut
3008      05      DEC B
3009      C2 05 30  JP NZ, &3005
```

Como pronto veremos, dibuja «B asteriscos». Ahora el valor de B disminuye una cada vez que se hace el bucle exterior ya que el valor lo recibimos por medio del registro C

DIRECCION	OPCODE	MNEMONICOS
3000	06 08	LD B, 8
3002	3E 2A	LD A, &2A
3004	CD 5A BB	CALL CharOut
3007	05	DEC B
3008	C2 04 30	JP NZ, &3004
3008	C9	RET

Programa 11

DIRECCION	OPCODE	MNEMONICOS
3000	0E 08	LD C, &08
3002	41	LD B, C
3003	3E 2A	LD A, &2A
3005	CD 5A BB	CALL CharOut
3008	05	DEC B
3009	C2 05 30	JP NZ, &3005
300C	3E 0A	LD A, &0A
300E	CD 5A BB	CALL CharOut
3011	3E 0D	LD A, &0D
3013	CD 5A BB	CALL CharOut
3016	0D	DEC C
3017	C2 02 30	JP NZ, &3002
301A	C9	RET

Programa 12

CP A	BF
CP B	BB
CP C	B9
CP D	BA
CP E	BB
CP H	BC
CP L	BD

Tabla 1: CP r

(LD B, C) y C se decrementa por el DEC C al final del bucle exterior en cada vuelta. Esto quiere decir que cada una de las líneas tiene un asterisco menos que la anterior nos da un triángulo efectivamente.

Cuando C llega a cero—después de dibujar ocho líneas—el programa sale a través de JP NZ, &3002 del bucle exterior y hace RET. Fortuitamente las líneas entre los JP NZs de los bucles interior y exterior,

```
300C      3E 0A      LD A, &0A
300E      CD 5A BB  CALL CharOut
3011      3E 0D      LD A, &0D
3013      CD 5A BB  CALL CharOut
```

son precisamente las que producen un salto de línea seguido de la vuelta de carro, asegurándose que nuestras líneas estén separadas. Vamos a intentar cambiar cualquiera de los dos y veremos que pasa. (Recuerde que cambiarán nuestros saltos).

Bueno, creemos que ya es bastante por esta semana. Sin embargo, le dejamos meditando dos cosas.

Si sólo estamos obteniendo el valor de B desde C, ¿por qué no nos preocupamos de C en absoluto? En realidad, es fácil, pero es el tipo de cosas que se nos pueden ocurrir después que nos hemos entumecido al estar durante varias horas programando en código máquina.

Para el final le hacemos una pequeña petición más. Intente dar la vuelta al triángulo, con un asterisco en la parte superior y ocho en la base.

## MICROHOBBY AMSTRAD SEMANAL

LE OFRECE AHORA SUS PROGRAMAS  
YA GRABADOS, PARA QUE VD.  
NO TENGA QUE TECLEARLOS

Todos 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 que no tengas que estar horas sobre el teclado de tu ordenador tratando de descifrar incomprensibles mensajes de error, **AMSTRAD SEMANAL** te ofrece cada mes los programas publicados de los cuatro números correspondientes en una cinta de cassette, sólo por **675 pts. (sin más gastos por envío)**.

Envíanos con la menor demora posible, el cupón correspondiente.

Para que tus dedos no realicen el trabajo duro, M.H. AMSTRAD lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un cassette mensual, solicítalos.

# COMPUTER CENTER

COMANDANTE ZORITA, 13  
28003 MADRID

TELS.: (91) 233 07 35  
(91) 233 07 81

AMSTRAD 464 Verde  
AMSTRAD 6128 Verde

**59.900** ptas.  
**99.900** ptas.

DISKETTE 3"

**1.050** ptas.

AMSTRAD 464 Color  
AMSTRAD 6128 Color  
INTERFACE DISCO 5.25"  
CINTA C-15 ESPECIAL ORDENADOR

**84.900** ptas.  
**117.500** ptas.  
**5.900** ptas.  
**85** ptas.

SOFTWARE ENTRETENIMIENTO (CASSETTE)

COMBAT LYNX  
DEAD PIT  
BEACH HEAD

**1.925** ptas.  
**1.925** ptas.  
**2.395** ptas.

ALIEN-8  
KNIGHT LORE  
JUMP JET

**1.875** ptas.  
**1.875** ptas.  
**2.695** ptas.

EXPLODING FIST  
GREMLINS  
DECATHLON

**2.095** ptas.  
**2.095** ptas.  
**1.920** ptas.

LIBROS

CURSO AUTODIDACTICO BASIC AMSTRAD

(Contiene manual y dos cassettes)

HACIA LA INTELIGENCIA ARTIFICIAL

**2.695** ptas.  
**1.300** ptas.

40 JUEGOS EDUCATIVOS

MUSICA Y SONIDO PARA AMSTRAD

PROGRAMANDO CON AMSTRAD

**1.800** ptas.  
**1.300** ptas.  
**1.900** ptas.

Tu pedido lo puedes recibir contra reembolso (libre de gastos), llamando a los telefonos: (91) 233 07 35 y (91) 233 07 81.

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

POR SOLO: **2.250** ptas.

—También disponemos de fundas para:

- SEIKOSHA SP-800 ..... 1.100 ptas.
- SEIKOSHA SP-1000 ..... 1.100 ptas.
- SPECTRUM 16/48 ..... 330 ptas.
- SPECTRUM PLUS ..... 460 ptas.
- COMMODORE 64 y VIC 20 ..... 680 ptas.
- SAGA 1 EMPEROR ..... 550 ptas.



## INTERESANTES CONDICIONES PARA DISTRIBUIDORES

CUPON DE PEDIDO

Recorte o copie este cupón y envíelo hoy mismo a:

**MICROSOFT-HARD**, Apartado 23.406, 08080-BARCELONA, o bien realice su pedido por teléfono llamando al (93) 348 04 07. (Tardes de 5 a 9). Gastos de envío: 150 ptas.

(firma)

Deseo recibir a vuelta de correo ..... funda(s).

Marca: ..... Modelo: 464-664-6128. Monitor: verde-color

—Contra reembolso: ..... —Talón adjunto a MICROSOFT-HARD, SL:

(MARQUE CON UNA «X» LAS OPCIONES DESEADAS)

—NOMBRE Y APELLIDOS .....

—DOMICILIO .....

—LOCALIDAD .....

—PROVINCIA .....

COD. POSTAL .....

# Suscríbete... y uno de estos tres sensacionales juegos será tuyo... ¡GRATIS!

**M.H. AMSTRAD** te da a elegir entre tres de los mejores juegos existentes en el mercado para AMSTRAD; **COMBAT LYNX**, **DALEY THOMPSON'S DECATHLON** y **BEACH HEAD**, cualquiera de los cuales puede ser tuyo solamente con suscribirte a nuestra revista. **Aprovecha esta ocasión excepcional** y ahorra 2.100 pesetas (precio de venta del programa) más el importante descuento que se produce en el precio de cada número, por el hecho de ser suscriptor. Disfruta de las ventajas que supone recibir cómodamente tu revista a domicilio y de la seguridad de tener tu ejemplar aunque se haya agotado en los quioscos.

**E**nviamos tu boletín de suscripción y no le des más vueltas, el número de juegos para regalos de suscripción, aunque grande, es limitado, y estos se podrían agotar mientras lo estás pensando.

**BEACH HEAD** producido por U.S. GOLD es una misión de desembarco en una costa fuertemente defendida por las fuerzas aeronavales enemigas. Debes conducir tu flota hacia la bahía y repeler el ataque aéreo, si lo consigues tu siguiente obstáculo será una flotilla de destructores y acorazados, superada la cual desembarcarás tus anfibios en las arenas de la bahía, estos deben superar las defensas costeras y llegar a la fortaleza que es el objetivo final.

**COMBAT LYNX** simula una misión de defensa de unas bases atacadas por una división acorazada. Disponemos para enfrentarnos a ellos de un modernísimo helicóptero. Este juego podría incluirse dentro del catálogo de los de estrategia, y su complejidad le dota de una gran dosis de adicción y belleza.

**DALEY THOMPSON'S DECATHLON** con este juego OCEAN enciende la llama olímpica y te reta a superar los récords de los campeonatos de todos los tiempos, el decathlon se desarrolla en dos días de competición y se compone de las siguientes pruebas:  
PRIMER DÍA: 100 m lisos, salto de longitud, lanzamiento de peso, salto de altura y 400 m lisos.  
SEGUNDO DÍA: 110 m vallas, lanzamiento de disco, salto con pértiga, lanzamiento de jabalina y los 1.500 m.

Utiliza el cupón  
adjunto a la revista  
o suscríbete por  
teléfono  
(91) 733 50 12  
(91) 733 50 16

OFERTA  
VALIDA  
SOLAMENTE  
PARA ESPAÑA

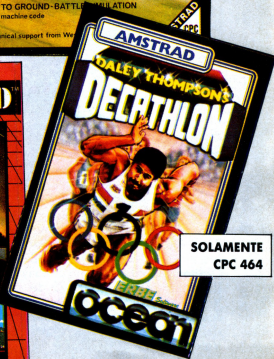
COMPATIBLE  
CPC 464  
CPC 664



COMPATIBLE  
CPC 464  
CPC 664



SOLAMENTE  
CPC 464



# 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!

Quisiera establecer contacto con usuarios de **Amstrad** en Granada, para cambiar listados e ideas. Preguntar por Miguel llamando al Tel. 22 36 66 o bien escribiendo a Cuesta / Escoriaza, 9 - 1.º B. Granada.

De particular a particular **Amstrad**. Por reciente viaje a Inglaterra, **vendo** 12 de los mejores juegos del mercado inglés a precio español. Entre ellos el James Bond 007, A view to a kill, Brian Bloodaxe de los Monthly Piton, Danger Mouse, Defender (versión **Amstrad** del clásico BBC) Dragon's Gold y el famoso Rally II. Llamar al (96) 332 12 72 de Valencia y preguntar por José. José Fco. Sanfeliú Gimeno. Barón de Cárcer, 32, 8.º - 15.º 46001 Valencia.

## K-BITS

Ordenadores personales y de gestión  
Aplicaciones para arquitectura

- Amstrad
- Sinclair
- Commodore
- Philips
- Canon
- Spectravideo
- Dragon
- Impresoras
- Monitores
- Periféricos
- Libros
- Revistas

C/ Barquillo, 15.  
Teléfono: (91) 232 57 37. Madrid.

Por cambio de ordenador **vendo** los siguientes programas a 600 ptas. cada uno: Rocky, Pole position, Decathlon, Gremlins, Knight Lore y algunos otros. Todos son originales. Mario Garcia. Villa Begoña, 6 - 4.º Iza. Haro (La Rioja). (941) 31 14 45.

**Vendo Amstrad** 464 fósforo verde por 35.000 ptas. Está como nuevo (2 meses). También vendería 10 programas de juegos a 500 ptas. cada uno. Entre ellos tengo Rocky, Knight Lore y Combate Libre. Maite Uñarte Murua. Apdo. 1.582. Bilbao.

## COLABORADORES:

Si eres poseedor de un **Amstrad**, tienes conocimientos de programación en Basic u otros lenguajes y te sientes capacitado para escribir acerca de un determinado tema directamente relacionado con el mundo de **Amstrad**, envíanos una carta mecanografiada, indicando los siguientes datos:

**Edad**  
**nombre y dirección**  
**teléfono de contacto**  
**relación de tus conocimientos**  
**lenguaje que dominas**  
Se valorará el conocimiento del inglés.

**Dirigirse a:**  
C/ La Granja, s/n. Polígono Industrial de Alcobendas (Madrid) indicando claramente en el sobre **AMSTRAD SEMANAL**

**Cambio programas Amstrad**, sólo comerciales, más de 100 títulos, entre ellos últimos éxitos. The Way of Exploding Fist, Gremlins, copiones, utilidades, etc. E. Valderas Fdez. Apdo. 41. Puertollano (Ciudad Real).

## GANA 100.000 PESETAS CON MICROHOBBY AMSTRAD SEMANAL

**P**orque pretendemos que **AMSTRAD SEMANAL** sea también vuestra revista, hemos abierto una sección en la que se publicarán los mejores programas originales recibidos en nuestra redacción. Vosotros seréis los encargados de realizar estas páginas, en las que podréis aportar ideas y programas interesantes para otros lectores.

### Las condiciones son sencillas:

- Los programas se enviarán a **AMSTRAD SEMANAL** en una cinta de cassette, sin protección en el software, de forma que sea posible obtener un listado de los mismos.
- Cada programa debe ir acompañado de un texto explicativo en el cual se incluyan:
  - Descripción general del programa.
  - Tabla de subrutinas y variables utilizadas, explicando claramente la función de cada una de ellas.
  - Instrucciones de manejo.

— Todos estos datos deberán ir escritos a máquina o con letra clara para mayor comprensión del programa.

— En una sola cinta puede introducirse más de un programa.

— Una vez publicado, **AMSTRAD SEMANAL** abonará al autor del programa de **15.000 a 100.000** pesetas, en concepto de derechos de autor.

— Los autores de los programas seleccionados para su publicación, recibirán una comunicación escrita de ello en un plazo no superior a dos meses a partir de la fecha en que su programa llegue a nuestra redacción.

— **AMSTRAD SEMANAL** se reserva el derecho de publicación o no del programa.

— Todos los programas recibidos quedarán en poder de **AMSTRAD SEMANAL**.

— Los programas sospechosos de plagio serán eliminados inmediatamente.

## ¡ENVÍANOS TU PROGRAMA!

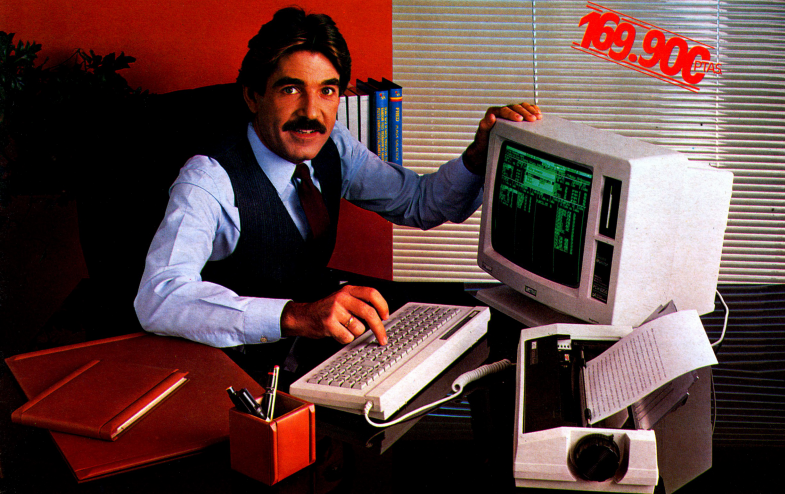
Indicando claramente en el sobre:

## AMSTRAD SEMANAL

a HOBBY PRESS, S. A. La Granja, n.º 8. Pol. Ind. Alcobendas (Madrid)

# ¡EXTRAORDINARIO!

169.900 PTAS



AMSTRAD le propone la solución más completa al ordenador personal y al proceso de textos. El sistema PCW 8256 le ofrece, por el precio de una máquina de escribir, un ordenador personal de 256K con **teclado profesional** y caracteres en castellano (ñ, acentos, etc.). Una pantalla de **fósforo verde de alta resolución** (90 col. x 32 lin.), con una **unidad de disco** de 3" (180K por cara) integrada en el monitor (con opción de integrar un segundo disco de 1Mb) y una **impresora** de textos de alta calidad con diversos tipos de letra.

Imprime tanto hoja a hoja como papel continuo y tiene una alineación automática de papel. Junto con el sistema PCW 8256, se suministra el procesador de textos LOCOSCRIPT (totalmente en castellano) el cual tiene reservadas, teclas específicas de control).

## UN EQUIPO EXTRAORDINARIO

Pero esto no es todo. El PCW 8256, al trabajar con el Sistema Operativo CP/M Plus, con un extra de gráficos GSX, tiene la posibilidad de acceder a los mejores programas profesionales del mercado: SuperCalc 2, Multiplan, dBase II, etc., así como a todo tipo de lenguajes: PASCAL, COBOL, FORTRAM, FORTH, etc.

Asimismo, gracias al programa Dr. LOGO, que se suministra con el Sistema PCW 8256, se tiene un inmenso campo de posibilidades en el mundo de la enseñanza.

## LO MAS NUEVO EN SU AMSTRAD PCW 8256

Disponer del sistema AMSTRAD 8256, es tener en la mano la última tecnología punta a precio de excepción.

Trabajar con «disco virtual» o, simplemente, operar con «ficheros indexados» era, hasta hace poco, solo imaginable en equipos muy sofisticados y de alto precio.

## El Sistema Informático PCW 8256 incluye en su suministro:

- Teclado profesional.
- Unidad de Disco.
- Pantalla de alta resolución.
- Impresora.
- Programas en disco:
  - Mallard BASIC con sistema JETSAM para ficheros indexados.
  - Sistema Operativo CP/M Plus.
  - Procesador de textos LOCOSCRIPT.
  - Lenguaje Dr. LOGO.
  - Diversas Utilidades.
- Completa documentación y manuales en castellano.
- **GARANTIA AMSTRAD ESPAÑA ÚNICA VALIDA PARA ACCEDER AL SERVICIO TECNICO OFICIAL.**

¡¡Inmense!!

ESPAÑA

# AMSTRAD

GARANTIA INDESCOMP

Avd. del Mediterráneo, 9 - 28007 Madrid Tels. 433 45 48 - 433 48 76 - Telex 47660 FAX - 4332450

# Rocky



Simulación tridimensional de un combate de boxeo. Lucha contra CIMBELIN, TED MATARE, JANSEN SINO y FIGHTER BULL para alzarte con el título mundial.

SPECTRUM 48 K 1.800 ptas.  
AMSTRAD CPC 464, 2.100 ptas.

ROCKY



Servimos a tiendas y almacenes telf. (91) 447 34 10 Telex 22542 JAGA E  
Pedidos contrareembolso (Sin G/E y adhesivos regalo) a:

"MANSION" DINAMIC: C/ Tilos, 2.21 MONTEPRINCIPE, BOADILLA DEL MONTE, MADRID TELF. (91) 715 00 67