

AÑO II N.º 46

MICROHOBBY

AMSTRAD

Semanal

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

AÑO II N.º 46

160 Ptas.

Canarias 165 pts.

ORDEN EN EL CAOS: UTILIZA DBASEII

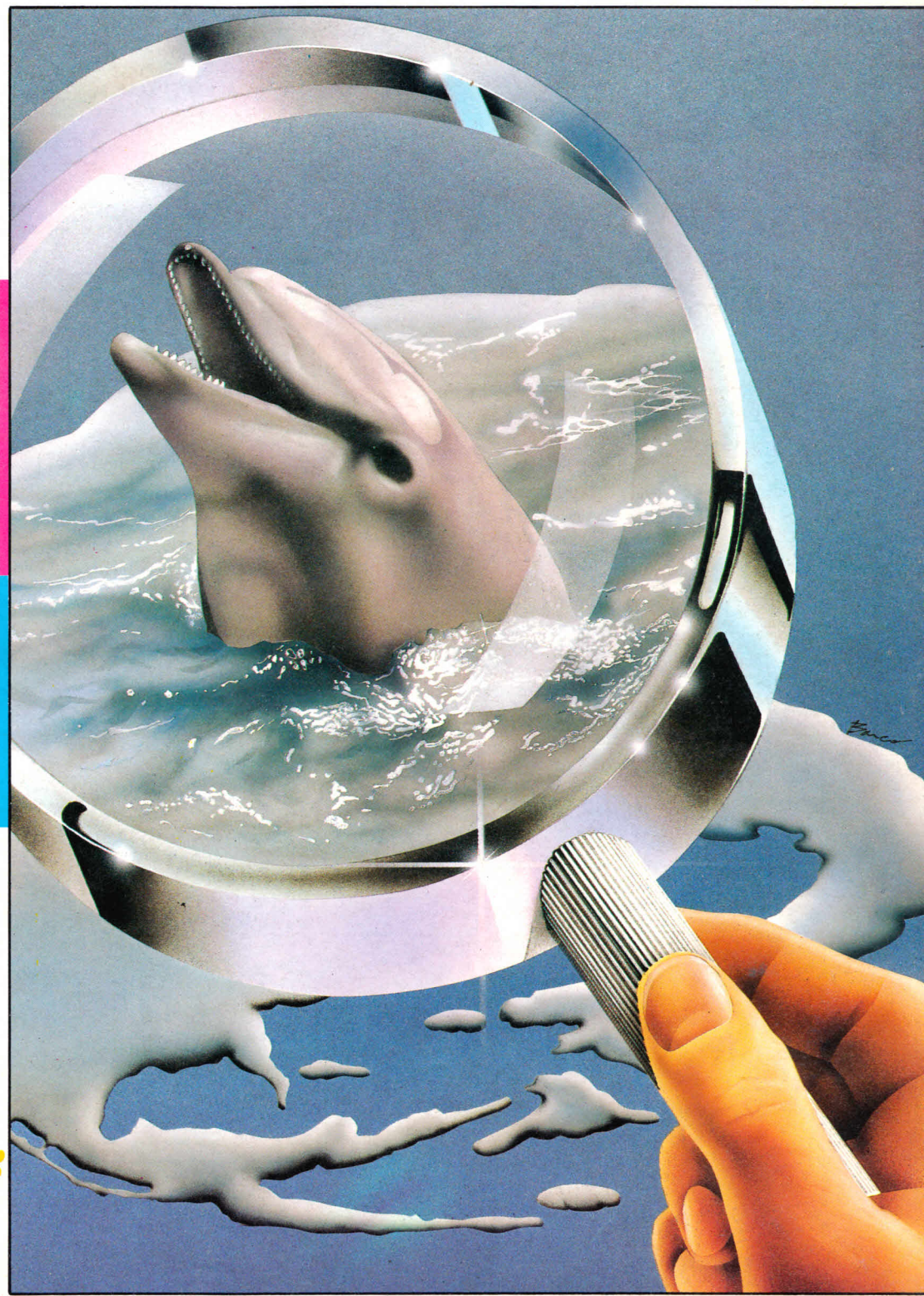
AMPLIA TUS GRAFICOS CON MAGNISCREEN HASTA EL INFINITO

¿FABRICARA AMSTRAD EN ESPAÑA?

COLOR A TOPE EN LENGUAJE MAQUINA

SOFTWARE

Alien Highway: El retorno de los vortones



FUTBOL



MEXICO 70



ALEMANIA 74



ARGENTINA 78



ESPAÑA 82



MEXICO 86

TOTAL

¿Alguna vez pensó en organizar su propio campeonato de fútbol?

Ha llegado ese momento.

Invente un uniforme para sus jugadores, elija entre sus equipos favoritos y reclute a los mejores futbolistas, creando así un equipo campeón, dotado con los mejores jugadores de todos los tiempos...

"SU EQUIPO SERA EL CAMPEON".

**PREPARESE
PARA EL 90**



DISTRIBUIDO en
Cataluña y Baleares por:
DISCLUB, S. A.
Balmes, 58
BARCELONA
Teléf. (93) 302 39 08

Fabricado y
distribuido bajo
licencia por:



Santa Cruz de Marcenado, 31 - 28015 Madrid - Teléf. 241 1063

AMSTRAD

sumario

Año II • Número 46 • 15 al 21 de Julio de 1986
 160 ptas. (incluido I.V.A.)
 Canarias, 155 ptas. + 10 ptas. sobretasa aérea.
 Ceuta y Melilla, 155 ptas.

22

Serie oro

Nuevamente gestión, pero con una calidad y una presentación que se salen fuera de lo común. DATABASE, un programa que difícilmente pasará desapercibido.

26 **Código máquina**

Aprende cómo utilizar la zona del firmware que controla los colores de la pluma y el papel de la pantalla.



30

MR Joystick

La aventura contra los vortones continúa. Un programa de asombrosos gráficos y que acabará con la paciencia de más de uno, pero que con su espectacularidad y adicción compensa, compensa y mucho.



6 **Inteligencia artificial**

Comprender las estructuras de datos que puede manejar el LISP, es absolutamente esencial para dominar las técnicas de la IA. La más importante, sin duda, las listas, de la cual se derivan directamente los árboles. Descubre como manejarlas y aplicarlas en tus propias creaciones.



12 **Primeros pasos**

Muchas y variadas son las instrucciones que semana a semana, hemos ido descubriendo y aplicando a nuestro ordenador, ahora es el momento de recapitular y comprobar nuestros conocimientos siguiendo, paso a paso, el diseño de un conocido programa de juego: el frontón.

16

Programación

¿Recuerdan el reductor de pantallas? Bien, pues ahora lo hacemos al contrario. MAGNISCREEN es capaz de ampliar hasta el infinito, modifique sus gráficos con este magnifico zoom.



Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

José M.ª Díaz

Redactor Jefe

Juan José Martínez

Diseño gráfico

Fernando Chaumel

Colaboradores

Eduardo Ruiz

Javier Barceló

David Sopena

Robert Chatwin

Francisco Portalo

Pedro Sudón

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amalio Gómez

Alberto Suñer

Secretaría Redacción

Carmen Santamaría

Fotografía

Carlos Candel

Chema Sacristán

Portada

M. Barco

Ilustradores

J. Igual, J. Pons, F. L. Frontán,

J. Septièn, Pejo, J. J. Mora

Edita

HOBBY PRESS, S.A.

Presidente

María Andrión

Consejero Delegado

José I. Gómez-Centurión

Jefe de Producción

Carlos Peropadre

Marketing

Marta García

Jefe de Publicidad

Concha Gutiérrez

Publicidad Barcelona

José Galán Cortés

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

Ctra. de Irún km 12,400
 (Fuencarral) 28049 Madrid
 Teléfonos: Suscrip.: 734 65 00
 Redacción: 734 70 12

Dto. Circulación

Paulino Blanco

Distribución

Coedis, S. A. Valencia, 245
 Barcelona

Imprime

ROTEDIC, S. A. Crta. de Irún.
 Km. 12,450 (MADRID)

Fotocomposición

Novocomp, S.A.

Nicolás Morales, 38-40

Fotomecánica

GROF

Ezequiel Solana, 16

Depósito Legal:

M-28468-1985

Derechos exclusivos
 de la revista

COMPUTING with the AMSTRAD

Representante para Argentina, Chile,
 Uruguay y Paraguay, Cia.
 Americana de Ediciones, S.R.L. Sud
 América 1.532. Tel.: 21 24 64. 1209
 BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace
 necesariamente solidaria de las
 opiniones vertidas por sus
 colaboradores en los artículos
 firmados. Reservados todos los
 derechos.

COMPUTIQUE

Te da más

GARANTIA **AMSTRAD**
ESPAÑA

66.900 Ptas.
Amstrad 464 f.
verde

IVA incluido



Al comprar tu Amstrad te regalamos

- Estuche con ocho programas originales

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

- Joystick Gunshot I

- Un estupendo libro de Basic

- Los cuatro mejores programas:

- | | |
|-------------|-----------------|
| ● Decathlon | ● Jet Set Willy |
| ● Sabrewulf | ● Beach-Head |

- Guía de referencia del programador

- y además te obsequiamos con un curso de introducción al Basic.

VENTA A PLAZOS HASTA 36 MESES



Nuevo Amstrad CPC6128: 94.900 ptas.

COMPUTIQUE

Servimos a tiendas
Abrimos sábados por la tarde

Embajadores, 90 Tfno. 2270980
28012 Madrid

FACTURACION 256

Aplicación:

Facturación, almacén y presupuestación.

Equipo:

Amstrad PCW8256

Capacidad:

200 clientes.

1.000 artículos.

1.500 apuntes de albarán.

Informes y documentos:

Listado general de clientes.

Listado de albaranes pendientes de facturar.

Listado general de artículos.

Listado de stocks bajo mínimos.

Confección de albaranes.

Confección de abonos.

Confección de facturas.

Confección de presupuestos/propuestas de pedido.

por Zelig Software

coste y el precio de venta. Permite altas, bajas y modificaciones. Al facturar, el programa da como salida la cantidad facturada. Para tener permanentemente actualizado el almacén, existe la opción de **entrada de artículos en almacén**.

El proceso de facturación se realiza a través de la introducción de apuntes de albarán. Es optativa la confección por impresora de los albaranes.

El programa permite realizar abonos a un cliente. Aparecerán en la factura disminuyendo el importe.

La numeración de facturas y albaranes puede ser automática o manual.



Descripción:

El programa de facturación Zelig proporciona al usuario una herramienta de trabajo fiable, de fácil manejo y de gran utilidad para la empresa. Las opciones están organizadas en forma de árbol —directorios y subdirectorios—, proporcionando un acceso lógico que facilita el aprendizaje sobre su manejo.

El programa utiliza dos ficheros maestros:

1. Fichero de clientes, con los siguientes campos: Código, nombre, CIF/DNI, Régimen Fiscal, domicilio, población, provincia, Código Postal, teléfono y persona de contacto. Permite altas, bajas y modificaciones.

2. Fichero de artículos, que contempla el stock en almacén, la cantidad mínima en stock, el precio de

La opción de confección de presupuestos proporciona 15 líneas en las que puede anotar artículos existentes, artículos no dados de alta o líneas de texto.

Los presupuestos contemplan el recargo por IVA y los descuentos.

Todos los listados se pueden realizar por impresora o por monitor.

Otras características:

Los ficheros son en «tiempo real», proporcionando mayor seguridad en el manejo de los datos.

Para aprovechar la máxima capacidad del disco, los programas se cargan en la memoria virtual (RAM-DIS).

La carga del programa se realiza de forma automática a través del fichero «PROFILE SUB».

Primera PLANA

¿SE FABRICARAN AMSTRADS EN ESPAÑA?

El rumor, en el sentido de la posible fabricación de ordenadores **Amstrad** en nuestro país a cargo de **Indescomp**, distribuidora oficial de la compañía británica, ha cobrado últimamente gran eco, hasta el punto de convertirse en uno de los famosos «**secretos a voces mejor guardados**».

Puestos al habla con los máximos responsables de Indescomp, nos comunicaron que, efectivamente, el proyecto existe, pero se encuentra en la fase correspondiente al estudio de viabilidad económica.

Como todos nuestros lectores saben, hoy por hoy, los **Amstrad** se fabrican en **Corea** por razones de abaratamiento de costes, especialmente la mano de obra. En España dicha mano de obra saldría a un precio superior al coreano, pero influyen otros muchos factores como el del transporte, por ejemplo, que deben ser profundamente estudiados. Por ello, siempre según Indescomp, el plan se encuentra en el estadio que podríamos llamar «**jornada de reflexión**».

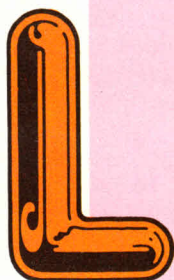
Si todo esto sale adelante, España podría verse convertida en el centro de distribución de productos **Amstrad** para toda Europa.

Por otra parte, **Indescomp** se mostró entre divertida y sorprendida por las declaraciones que aparecieron en algunos periódicos en el sentido de que tal proyecto se ha desencadenado a instancias de la Administración, cuando ellos, personalmente le comunicaron la idea a Alan Sugar hace más de un año. Parece que, con esto de los mundiales de diversos deportes, mucha gente busca apuntarse tantos.

LISTAS, LISP Y RECURSION

Por Roberto Garrote Bernal

Los que conozcáis algo del lenguaje de programación LISP o hayáis seguido los artículos de Inteligencia Artificial que han aparecido en esta revista ya sabréis que la principal estructura de datos que maneja LISP es la lista (en Basic son las matrices). Para aclarar las ideas y comprender mejor las explicaciones que siguen, contaré primero cómo se representan las listas en LISP.



La memoria que reserva para su uso el geniecillo de LISP (el intérprete) está dividida en nodos, cada uno de los cuales está formado por dos punteros (dos direcciones de la memoria): uno se llama **CAR** y el otro **CDR**. (Estos nombres, que en principio pueden parecer caprichosos, tuvieron su origen en la primera implementación que se hizo de LISP, sobre un IBM 704. Algunas de las instrucciones de esa máquina hacían referencia al «contenido del registro de direcciones» (Contents of Address Register, CAR) y al «contenido del registro de disminución» (Contents of Decrement Register, CDR)). Excepto los **átomos** (literales o números), cualquier otro objeto que maneje LISP está construido mediante la agregación de estos nodos (en LISP se llaman pares). Es como si fueran los ladrillos de una casa. Veremos que esta estructura es muy versátil y que con ella se puede «edificar» casi cualquier cosa.

En el artículo del **Especial n.º 2** ya vimos que la función CONS servía para construir pares. Por ejemplo, (CONS 'A 'B) crea un nuevo nodo con el puntero **CAR** apuntando al átomo A y el puntero **CDR** apuntando al átomo B. El resultado se puede ver en la fig. 1. La función CAR aplicada a un par nos devuelve aquello que está siendo apuntado por el puntero **CAR** del par. Siguiendo con el ejemplo anterior, (CAR '(A.B)) nos da el átomo A. El caso de la función CDR es semejante. Veamos ahora cómo se construyen listas usando pares. Daré de nuevo la definición de lista:

1.—La lista vacía () es una lista (también es un átomo, el átomo NIL).

2.—Si S1 es una lista a un átomo y S2 es una lista entonces el par que tiene como CAR S1 y como CDR S2 es una lista. (Se dice entonces que S1 es el primer elemento de la lista y S2 es el resto de la lista).

Además los únicos objetos que son listas son los descritos por los puntos 1 y 2.

Algunos ejemplos de listas son:

() Por el punto 1 de la definición. Es la lista vacía.

NIL Es exactamente el mismo objeto que ().

(A. NIL) Por el punto 2: A es un átomo y NIL es una lista.

(A. ()) Igual que el anterior.

(A) Es una representación en forma de lista de (A. NIL).

(A. (B)) Por el punto 2: A es un átomo y (B) es una lista.

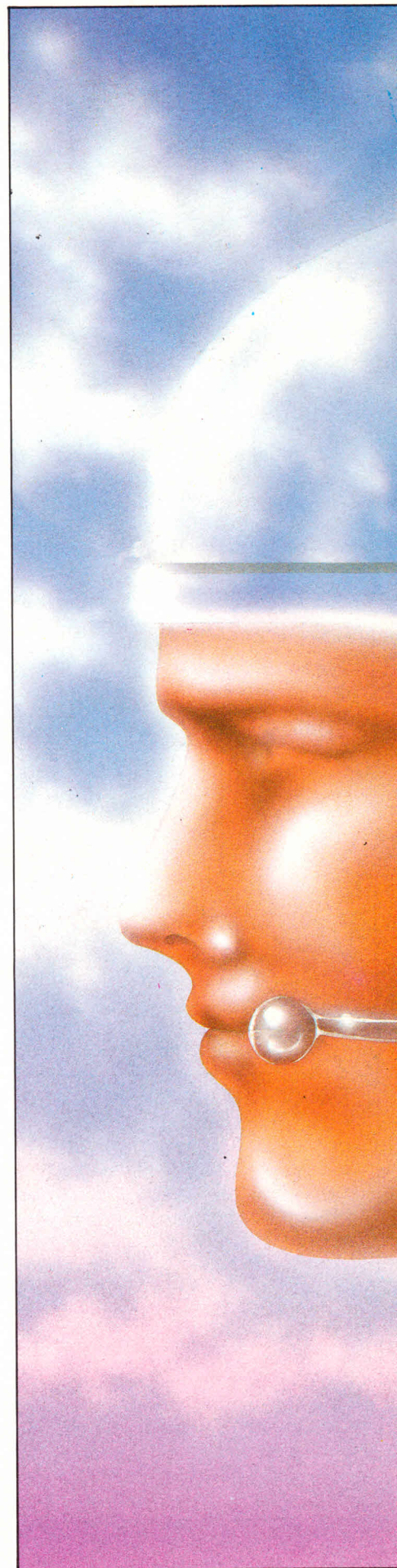
(A B) Es una representación de (A. (B)).

((A). (A B)) Por el punto 2: (A) es una lista y (A B) también.

((A) AB) Es la representación en forma de lista del anterior.

Átomos y pares constituyen las **S-expresiones**. A partir de esta definición podemos imaginarnos las S-expresiones como árboles donde las hojas son los átomos y los puntos de ramificación son los pares. Algunos ejemplos se pueden ver en la fig. 2.

Las **listas**, como S-expresiones que son, también se pueden representar de esta forma. De hecho, la fig. 2. (c) representa una lista, pues NIL es una lista y C es un átomo, luego el par (C. NIL) es una lista. Considerado como lista, este par se escribe como (C). De la misma forma (B. (C. NIL)) es la lista (B C) y (A. (B. (C. NIL))) es la lista (A B C). Algunas personas prefieren representar las listas de forma diferente, como en la fig. 3, pero ambas son equivalentes.





Con todos estos conocimientos podemos comenzar ya a manipular listas. En lo que resta de artículo utilizaré para los ejemplos el intérprete de MINILISP que apareció en el Especial AMSTRAD n.º 2. No es imprescindible su uso para la comprensión de lo que voy a contar, pero sí puede ser una gran ayuda.

Modificar el lenguaje es fácil

En primer lugar vamos a empezar cambiando algunos nombres. LISP es un lenguaje que facilita en gran medida la comprensión de los programas. Para aumentar la «legibilidad» de los mismos, LISP permite la utilización de identificadores (átomos) bastante largos (en MINILISP el tamaño máximo es de 255 caracteres, pero se ha calculado una media de 8 caracteres por cada átomo, de modo que en principio no te preocupes por esto). La función **CAR** aplicada a una lista nos da su primer elemento. El nombre CAR no nos dice mucho, de modo que definiremos una función PRIMERO que nos dará el primer elemento de una lista.

```
> (DE PRIMERO (LIS) (CAR LIS))
```

La función CDR nos da el resto de una lista, así es que démosle este nombre.

```
> (DE RESTO (LIS) (CDR LIS))
```

Espero que os aparezca más clara la expresión (PRIMERO '(A B C)) que (CAR '(A B C)), aunque ambas valen lo mismo, como puedes comprobar haciéndole ambas preguntas a tu geniecillo de MINILISP. Podríamos ahora escribir funciones que calculen los elementos segundo tercero, etc. de una lista, pero en su lugar vamos a escribir una función general para que podamos acceder a cualquier elemento. Esta función habrá de tener dos argumentos: el primero será el lugar que ocupa el elemento de la lista que queremos conocer y el segundo será la propia lista. Supongamos que llamamos a esta función **N-ESIMO**. Entonces

```
> (N-ESIMO 4 '(A B C D E))
```

será D, y

```
> (N-ESIMO (+2 1) '(A B (C D) (E F)))
```

será (C D), ya que (+2 1) vale 3 y el tercer elemento de la lista es (C D). Recordemos que LISP evalúa los argumentos de las funciones que se definen con DE. Aquí llega la definición de la función N-ESIMO

```
> (DE N-ESIMO (N LIS)
```

```
  > (COND
```

```
    > ((=N 1) (PRIMERO LIS))
```

```
    > (T (N-ESIMO (SUB N 1) (RESTO LIS)))
```

```
  >))
```

Inteligencia ARTIFICIAL

Esta función está construida de la forma en la que habitualmente se hacen las funciones en **LISP**:

1.—Busca el caso más sencillo posible y resuélvelo.

2.—Para los casos que no se resuelvan en el punto 1, reduce el problema a términos un poco más sencillos y luego aplica a este problema reducido el algoritmo general.

Seguro que con un ejemplo se aclara todo esto. Tomemos el caso de la función N-ESIMO. En LISP sólo se puede acceder directamente al primer elemento de una lista. Por tanto, el caso más sencillo es cuando queremos conocer el primer elemento de la lista, y esta es la primera condición que hemos puesto. Además ya sabemos calcular el primer elemento de una lista: con la función PRIMERO. La expresión (COND ((=N 1) (PRIMERO LIS))... se podría interpretar como lo siguiente: «Si N es igual a 1 entonces el N-ESIMO elemento de una lista es su primer elemento». El resto de los casos se consideran más complicados. ¿Cómo podemos reducirlos a casos más sencillos? Bien, pues supongamos que queremos calcular el segundo elemento de una **lista**. Para ello sólo disponemos de las funciones que nos dan el primer elemento de una lista y el resto de la misma. Tomemos una lista, por ejemplo:

```
(A B C)
```

Esta lista tiene como primer elemento el átomo A y como resto la lista.

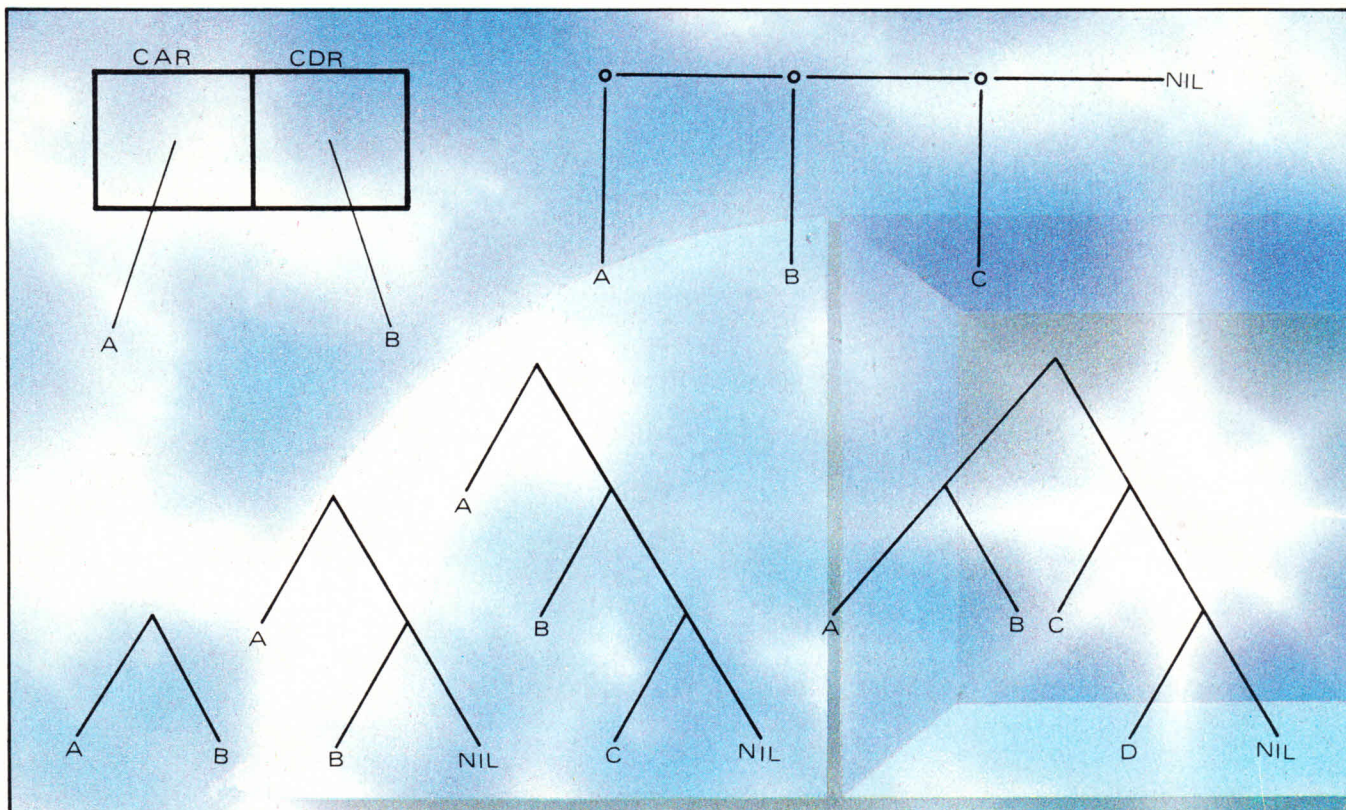
```
(B C)
```

El segundo elemento de la lista (A B C) es B, ¡pero B es el primer elemento de la lista (B C)! Es decir, que el segundo elemento de una lista es el primer elemento del resto de la lista. Vamos a definir la función que nos da el segundo elemento de una lista para que lo entiendas mejor:

```
> ((DE SEGUNDO (LIS) (PRIMERO (RESTO LIS)))
```

Y para calcular el tercero, ¿qué habrá que hacer? El tercer elemento de la lista (A B C) es C, que a su vez es el segundo elemento de la lista (B C). O sea, el tercer elemento de una lista es el segundo elemento del resto de la lista. ¡Y ya sabemos calcular el segundo elemento de una lista!

```
> ((DE TERCERO (LIS) (SEGUNDO (RESTO LIS)))
```



En resumen:

1.—Sabemos calcular el primer elemento de una lista utilizando la función PRIMERO.

2.—Para calcular el segundo elemento de una lista basta con calcular el resto de una lista y su primer elemento. Para calcular el tercer elemento de una lista basta con calcular el resto de la lista y su segundo elemento...

«El elemento que ocupa la posición N en una lista es el mismo elemento que ocupa la posición N-1 en el resto de la lista (siempre que N sea mayor que 1)». Esto es justamente lo que dice la segunda parte de la expresión condicional que define la función N-ESIMO. Podemos ahora unir las piezas para entender el significado de la expresión condicional.

```
(COND
  ((= N 1) (PRIMERO LIS))
  (T (N-ESIMO (SUB N1) (RESTO LIS))))
```

«Si N es igual a 1 entonces el elemento que ocupa la posición N de una lista es su primer elemento. En otro caso (si N es mayor que 1), el elemento que ocupa la posición N de una lista es el que ocupa la posición N-1 en el resto de la lista».

¡Recursos recursivo!

Con esto he cumplido en parte con lo que os prometí la semana pasada: hablar de la recursión. Sí, porque la función N-ESIMO también es recursiva. El esquema de construcción de funciones LISP que dimos antes también es

recursivo, más aun, es el fundamento de los algoritmos recursivos. Es casi seguro que aún habrá alguien que se pregunte cómo es posible que estas definiciones, que utilizan lo que se quiere definir, funcionen; ¡si es que de verdad funcionan, claro! Como nos gusta demostrar nuestras afirmaciones, vamos a comprobar que sí que funcionan y además vamos a ver cómo lo hacen. Si no lo habías hecho ya, escribe la definición de la función N-ESIMO junto con las de las funciones PRIMERO y RESTO en tu intérprete de MINILISP, en el orden en que más te guste porque a tu geniecillo le da igual. ¿Ya está? Perfecto. Ahora teclea:

```
> (TRACE '(N-ESIMO))
```

Recuerda que el símbolo « > » es el indicativo de que MINILISP está dispuesto, así que tú no debes escribirlo. Vamos a ejecutar ahora la función **N-ESIMO** para unos datos concretos e iremos observando cómo cambian los valores de los parámetros a lo largo de la ejecución y cómo se va devolviendo hacia arriba el valor calculado. Escribe:

```
> (N-ESIMO) 4 '(A B C D E F)
```

Y poco a poco irá apareciendo lo siguiente en la pantalla de tu computadora:

```
...
ARGUMENTO DE EVAL:
(N-ESIMO 4 (QUOTE (A B C D E F)))
ARGUMENTO(S) DE N-ESIMO:
(4 (A B C D E F))
ARGUMENTO(S) DE N-ESIMO:
(3 (B C D E F))
ARGUMENTO(S) DE N-ESIMO:
```

```
(2 (C D E F))
ARGUMENTO(S) DE N-ESIMO:
(1 (D E F))
VALOR DE N-ESIMO:
D
VALOR DE N-ESIMO:
D
VALOR DE N-ESIMO:
D
VALOR DE N-ESIMO:
D
TIEMPO 8.14 VALOR:
D
```

Es posible que el tiempo no sea el mismo, ya que a veces el geniecillo de MINILISP se muestra un poco perezoso, pero todo lo demás debería ser igual. Se observa bien cómo según bajamos en la secuencia de llamadas va disminuyendo el parámetro numérico (4, 3, 2, 1) a la vez que disminuye el tamaño de la lista y nos acercamos más al elemento que queremos obtener: D. Finalmente, cuando se alcanza la condición de terminación —el caso más simple, o sea, cuando N vale 1— se calcula el valor de la función y este valor se transmite hacia arriba, a través de la secuencia de llamadas, para emerger finalmente.

La recursión el **LISP** funciona bien siempre que se verifiquen dos condiciones:

(a) Que la acción que hay que realizar cuando la condición de terminación se cumple no dependa de la función que se pretende definir.

(b) Que cada vez que se haga una llamada recursiva los argumentos de la función se «parezcan» un poco más a los argumentos de la condición de terminación.

¿Qué se parece a qué?

El concepto de «parecido» entre argumentos depende del problema de que se trate. En el caso de la función N-ESIMO la condición de terminación en que N sea igual a 1. Por tanto, podemos considerar que un argumento «se parece» más que otro al argumento de la condición de terminación si está más próximo a 1. Por ejemplo, 3 está más próximo que 5.

La acción a realizar cuando la condición de terminación se cumple es calcular el primer elemento de la lista. Para hacer esto usamos la función **PRIMERO**, que no depende de la función N-ESIMO. Por lo tanto, la condición (a) se cumple. La condición (b) se cumple siempre que el valor que asignemos a N (a través del mecanismo de paso de parámetros, del que hablamos la semana pasada) sea mayor o igual que 1. Si N vale 1 entonces se verifica la condición de terminación. Si N es mayor que 1 entonces el intérprete de MINILISP va a intentar calcular:

```
(N-ESIMO (SUB N 1) (RESTO LIS))
```

(SUB N 1) Vale N-1, luego en la siguiente iteración N valdría una unidad menos y, por lo tanto, estaría más próximo a 1 que antes. Los problemas surgen cuando N es menor que 1. En ese caso, al hacer (SUB N 1), nos alejamos cada vez un poco más de 1. Los resultados son **catastróficos** como puede observarse en la ejecución de (N-ESIMO -1 '(A B C D))

```
...
ARGUMENTO DE EVAL:
(N-ESIMO -1 (QUOTE (A B C D)))
ARGUMENTO(S) DE N-ESIMO:
(-1 (A B C D))
ARGUMENTO(S) DE N-ESIMO:
(-2 (B C D))
ARGUMENTO(S) DE N-ESIMO:
(-3 (C D))
```

```
ARGUMENTO(S) DE N-ESIMO:
```

```
(-4 (D))
```

```
ARGUMENTO(S) DE N-ESIMO:
```

```
(-5 NIL)
```

```
ARGUMENTO(S) DE N-ESIMO:
```

```
(-6. * * * POCOS ARGUMENTOS O DE TIPO ERRONEO
```

```
* * * POCOS ARGUMENTOS O DE TIPO ERRONEO
```

```
* * * POCOS ARGUMENTOS O DE TIPO ERRONEO
```

En ese momento presionamos la tecla [ESC] dos veces para cortar la ejecución del programa, pues en otro caso jamás hubiese acabado. (Si tú has probado a hacerlo antes que nosotros debes escribir **GOTO 61000** para continuar en MINILISP. Si esperas más tiempo hasta que aparezca el error de BASIC «Memory full» sólo tendrás una alternativa: apagar la computadora y volver a cargar el intérprete de MINILISP).

Para que nos ocurran este tipo de cosas podemos hacer dos cosas: una es definir **N-ESIMO** de la siguiente forma (no lo escribas todavía que esta solución es «la mala»)

```
> (DE N-ESIMO2 (N LIS)
> (COND
> ((> N 1) '(NADIE ME HA PROGRAMADO PARA QUE CALCULE ESE ELEMENTO))
> ((= N 1) (PRIMERO LIS))
> (T (N-ESIMO2 (SUB N 1) (RESTO LIS)))
> ))
```

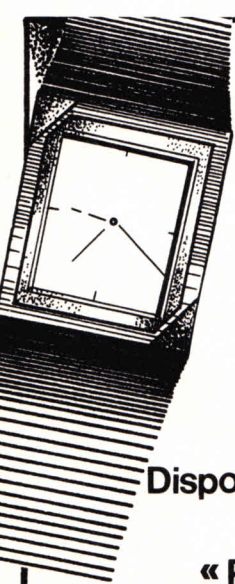
¿Por qué digo que esta solución es mala? En caso de que N sea menor que 1 la función aborta inmediatamente, pero si N es mayor o igual que 1 la primera vez nunca podrá ser menor que 1, luego cada vez que recursivamente se llama a **N-ESIMO2** hay que preguntar si N es menor que 1, aunque ya sabemos que no puede serlo. Para ahorrarnos el

Inteligencia ARTIFICIAL

tiempo que se tarda en hacer esta pregunta, la haremos sólo la primera vez, de la siguiente forma (ahora ya la puedes copiar porque ésta es «la buena»). Para hacer esto debes borrar antes la antigua definición con (REMPROP 'N-ESIMO 'EXPR) y desdoblar entonces el cursor para copiar el trozo de definición que te valga. Para **desdoblar** el cursor debes pulsar simultáneamente una de las teclas que mueve el cursor y [SHIFT]

```
> (DE N-ESIMO (N LIS)
> (COND
> ((> N 1) '(NADIE ME HA PROGRAMADO PARA QUE CALCULE ESE ELEMENTO))
> (T (N-ESIMO-AUX N LIS)
> ))
> (DE N-ESIMO-AUX (N LIS)
> (COND
> ((= N 1) (PRIMERO LIS))
> (T (N-ESIMO-AUX (SUB N 1) (RESTO LIS)))
> ))
```

Hasta el próximo artículo te puedes ir entreteniendo comparando los tiempos de ejecución de las dos nuevas versiones de la función que calcula el n-esimo elemento de una lista: N-ESIMO y N-ESIMO2. Otro ejercicio interesante puede ser arreglar la función **FAC-TORIAL** que dimos la semana pasada de forma que para los números negativos de un mensaje de error, de igual forma que hemos hecho con N-ESIMO.



¡¡ RECUPERA

LAS ASIGNATURAS PENDIENTES !

■ SOCIALES ■ NATURALES ■ MATEMATICAS ■ LENGUAJE ■

DE : 5' 6' 7' y 8' EGB

Programa para ejercicios de ortografía **EDORTO**

Programa para ejercicios de atención y comprensión **EDACLE**

Programa para ejercitar el instrumento lector **EJEILE**

Disponibile en MSX y AMSTRAD.. 1800 ptas Cassette ___ 2750 Disco

«PRECIO ESPECIAL» CURSO COMPLETO EGB... 3500 ptas.

PEDIDOS : **GOSLINE** CUARTELES, 43-1°-29002-MALAGA-TEL. 311877

... ..regalo de un reloj por cada cassette o disco

En tu kiosco te espera algo muy inteligente

El AMSTRAD Especial número 2 incluye una cinta de cassette adherida a la portada con un lenguaje Lisp completo que

te permitía comprender y dominar las técnicas más complejas de inteligencia artificial. Por si fuera poco, en nuestra

cinta se incluyen también dos concursos: uno, de diseño gráfico de pantallas, para artistas, te permitirá ganar hasta 170.000 pesetas en premios. En el segundo regalamos un ordenador Amstrad CPC6128. El número 2 de AMSTRAD Especial trata un amplio espectro de interesantes temas, como un comparativo de impresoras, que le ayudará a elegir la más adecuada a sus necesidades, como multitud de programas y rutinas de utilidad en lenguaje máquina como un paquete de soft integrado, con tres programas en uno y un largo etcétera que sería demasiado prolijo detallar.

Si no lo encuentras en tu kiosco, solicítalo directamente a nuestra Editorial.

MICROHOBBY
AMSTRAD
Especial Año I N.º 2

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

475 ptas.

SOFTWARE INTEGRADO
3 PROGRAMAS EN UNO

TABLETA GRAFICA
GRAFPAD II:
EL ARTE POR ORDENADOR
A TU ALCANCE

TE OFRECEMOS UN
LENGUAJE DE
PROGRAMACION LISP
COMPLETO EN CINTA
DE CASSETTE

NUESTRO PROGRAMA CREADOR
DE CRUCIGRAMAS DESAFIA A TU INGENIO

IMPRESORAS:
COMO HACER LA
MEJOR ELECCION

ATENCION A NUESTRO
FABULOSO CONCURSO:
PUEDES GANAR UN
CPC-6128 CON
SOLO CARGAR LA CINTA

Rellena este cupón y envíalo a **HOBBY PRESS, S. A.** Ap. de Correos 232. Alcobendas. Madrid.

Nombre _____ Apellidos _____
Domicilio _____ C. Postal _____
Localidad _____ Provincia _____
Teléfono _____ Profesión _____ Fecha de nacimiento _____

¿Eres suscriptor de **MICROHOBBY AMSTRAD**? Sí No
Deseo recibir el Especial de **MICROHOBBY AMSTRAD** n.º 2 al precio de 475 ptas. (IVA incluido)

FORMA DE PAGO
 Talón bancario adjunto a nombre de **HOBBY PRESS, S. A.**
 Mediante Tarjeta VISA. N.º _____ Fecha de caducidad _____
 Contra reembolso (supone 75 ptas. de gastos de envío).
Fecha y firma: _____

A D 1

AMSTRAD CPC-6128



- MICROPROCESADOR Z80A.
- 128 K DE MEMORIA RAM (41 K DE USUARIO EN BASIC Y 61 K EN CP/M PLUS)
- 48 K DE MEMORIA ROM QUE INCLUYEN EL LOCOMOTIVE BASIC Y EL SISTEMA OPERATIVO.
- 76 TECLAS, TECLADO NUMERICO Y DE CURSOR INDEPENDIENTE.
- TEXTO EN MONITOR DE 20, 40 U 80 COLUMNAS Y GRAFICOS CON DEFINICION DE HASTA 640 X 200 PUNTOS. 27 COLORES DISPONIBLES.
- HASTA 8 VENTANAS EN PANTALLA.
- GENERACION DE SONIDOS EN 3 VOCES Y 8 OCTAVAS.
- UNIDAD DE DISCO DE 3" (169 K BYTES)
- SISTEMAS OPERATIVOS AMS-DOS Y CPM/PLUS
- CONECTORES PARA IMPRESORA, JOYS-TICKS, CASSETTE, SEGUNDA UNIDAD DE DISCO, ETC.

SISTEMA COMPLETO CON MONITOR EN FOSFORO VERDE, MANUAL EN CASTELLANO, GARANTIA OFICIAL AMSTRAD ESPAÑA, DISCO CON SISTEMA OPERATIVO CP/M 2.2 Y LENGUAJE DR. LOGO, DISCO CON SISTEMA OPERATIVO CP/M PLUS (CP/M 3.0) Y UTILIDADES, DISCO CON SIETE PROGRAMAS DE OBSEQUIO

84.900 Pts. + I.V.A.

SISTEMA COMPLETO IGUAL AL ANTERIOR PERO CON MONITOR EN COLOR.

119.900 Pts. + I.V.A.

AMSTRAD[™]
ESPAÑA

Avd. de Mediterráneo, 9, 28007 MADRID.
Tels. 433 45 48 - 433 48 76

Delegación Cataluña: C/. Tarragona, 110,
08015 BARCELONA - Tel. 325 10 58

¿SABE JUGAR AL FRONTON?

Hemos pensado que la mejor forma de cerrar esta serie de artículos donde se nos contaban los distintos modos de mover un objeto por la pantalla y, además, cómo aplicar correctamente las distintas funciones de lectura del teclado, es ver un ejemplo práctico en el que esté recogida la mayor parte de los distintos casos comentados.



Hay mejor resumen, o por lo menos más divertido, que el diseño y realización de un juego?

Por supuesto que no se va a tratar de una cosa muy sofisticada, modelo de perfección y complicación, pero seguro que servirá para despejar las «poquititas» dudas que tenga sobre el tema, a la par que podrá utilizarlo para pasar agradablemente alguna que otra tarde ociosa.

¿Qué le parece si nos ponemos a trabajar? Lo primero que haremos será definir con la mayor claridad cuál es nuestra idea.

Diseño de una idea

Aunque se trate de un «clásico» de los juegos de ordenador, se nos ha «ocurrido» que nuestro juego será el típico frontón (o muro), en el que nosotros golpearemos con una paleta una bola que irá rebotando en las paredes del mismo, hasta que nuestros reflejos no sean lo suficientemente rápidos y no podamos devolverla.

La partida terminará cuando se nos hayan terminado las cinco bolas que poseemos al principio de la jugada.

Además vamos a intentar que podamos elegir las teclas con las que movemos la paleta. Así adaptaremos el juego a nuestro gusto para batir los «récords» con mayor facilidad.

Para su diseño seguiremos, en la medida que sea posible, el método de descomponer el programa en pequeños submódulos o subrutinas que realicen acciones muy concretas y sencillas. Todas ellas van a ser controladas desde un programa llamado «programa principal» que se compondrá, casi exclusivamente, de instrucciones de llamadas a dichas subrutinas (GOSUB). No es la primera vez que aplicamos este sistema, ¿verdad?

Estructura del programa principal

Comencemos con este programa principal que actuará como un director de orquesta. ¿Qué hace?

Como siempre, pensemos un poquito. Confeccionemos el proceso lógico a seguir para luego plasmarlo gráficamente en un organigrama que nos será de gran ayuda.

Lo primero sería inicializar una serie de datos que manejaremos más tarde (ya veremos cómo), y también el teclado. Mientras el ordenador está haciendo cualquier cosa, cuando pulsamos una tecla su carácter asociado queda guardado en un almacén —hasta un cierto límite. Si no queremos tener problemas,

resultará conveniente vaciar, en cuanto el programa comience a ejecutarse, dicha «despensa».

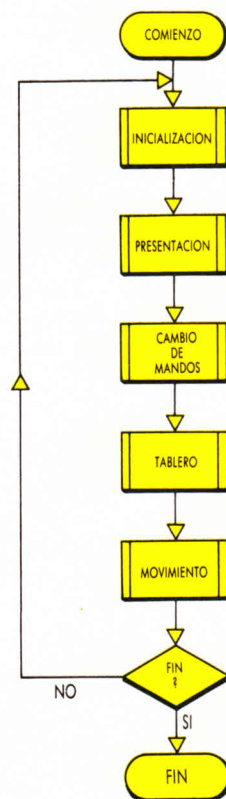
A continuación, tendrán que aparecer en la pantalla unos rótulos informativos que nos cuenten las teclas que necesitaremos pulsar para jugar. Pero aquí, además, hemos incluido la posibilidad de cambiarlas por otras que se adapten más a nuestro modo de juego. No lo olvide.

Si ya hemos seleccionado los «mandos», ¡adelante! Borrarnos todo lo que haya sobre la pantalla, dibujamos el terreno de juego y nos ponemos en movimiento; bueno, se pondrán en movimiento la bola y la paleta.

Cuando se nos agoten las bolas —o termine la partida— el **Amstrad** nos informará de esta situación y nos preguntará si queremos continuar o no para obrar en consecuencia. Si nuestra respuesta es «sí», continuamos con la diversión y en caso contrario: ¡Fin y se acabó!

¿Está suficientemente claro lo que queremos hacer? Pues, manos a la obra. El organigrama, o gráfico que refleje este proceso lógico, es el que nos muestra la figura 1. En realidad es uno de los muchos que podemos hacer. Recuerde que en esto de la informática la solución a un problema nunca es única.

FIG. 1: Organigrama del programa principal.



De ahí a codificar el programa principal sólo hay que dar un pequeño paso. Es lo bueno que tienen estos métodos gráficos.

Observe que no está decidida la forma de realizar cada una de estas acciones. De momento nos basta con saber que habrá una rutina para realizar cada una de ellas. Tenemos una visión general del problema.

Y como hemos dicho que es muy fácil, una ojeada al Programa I le tendrá que resultar familiar, ¿o no?

Programa uno

Tiene dos cosas importantes: las distintas llamadas a las subrutinas (líneas 30 a 70) y el empleo de la función INKEY en las líneas 120 y 130. Analicemos esta última.

INKEY(60) nos devuelve un valor indicativo de si hemos pulsado la tecla «S» o no (60 es el número asociado a la «S»). Este número lo almacenamos en la variable «valor». A continuación, en la línea 130, analizamos si

```
10 REM PROGRAMA I
20 REM PROGRAMA PRINCIPAL
30 GOSUB 2000:REM INICIALIZACION
40 GOSUB 3000:REM PRESENTACION
50 GOSUB 4000:REM CAMBIO DE MANDOS
60 GOSUB 5000:REM CUADRADO
70 GOSUB 1000:REM MOVIMIENTO DE LA BOLA
80 LOCATE 9,12
90 PRINT"HAS TERMINADO, FORASTERO"
100 LOCATE 12,14
110 PRINT"OTRA PARTIDA? S/N"
120 valor=INKEY(60)
130 IF valor<>"S" THEN RUN ELSE IF INKEY(46)<>"N" THEN :END ELSE GOTO 120
```

la hemos presionado para comenzar de nuevo la ejecución del programa o, en caso contrario, explorar si ha sido la «N» (que tiene asociado el número 46) y el programa terminó.

En caso de no ser ninguna de las dos, volveríamos a realizar un nuevo chequeo del teclado saltando otra vez a la línea 120.

Recuerde que «S» es el valor que nos devuelve INKEY si no hemos colocado ninguno de nuestros «deditos» sobre el teclado. Y eso es todo, sólo nos queda desarrollar cada una de las subrutinas. Fácil, ¿verdad?

¿Qué tendremos que inicializar? Ya hemos contado que es muy conveniente vaciar el almacén donde se guardan las teclas pulsadas. Después, parece lógico dar un valor inicial a cada una de las coordenadas de la posición donde aparecerá la bola cuando comience a ejecutarse el programa. Y podemos extender éstos a las de la paleta, puntos y bolas iniciales.

También determinaremos, de un modo aleatorio, el sentido de desplazamiento respecto

a los ejes X e Y. Todo ello lo recogemos en la figura II.

FIG. II: Rutina de inicialización.



Vea que aquí las cosas son ya mucho más concretas. Pasar a instrucciones Basic este organigrama no debe presentar ya ningún problema. El Programa II es la muestra.

Programa dos

Hay una novedad en la línea 2030: CLEAR INPUT. Es la instrucción que empleamos para inicializar el teclado.

CLEAR INPUT

desecha todos los caracteres que hayamos teclado mientras el ordenador haya estado realizando algún trabajo.

Con esta orden conseguimos que el ordenador comience a analizar sólo los caracteres teclados a partir de ella, ¿entendido?

```
2000 REM PROGRAMA II
2010 REM INICIALIZACION
2020 CLS
2030 CLEAR INPUT
2040 xbola=20:ybola=12
2050 xpal=20
2060 bolas=5
2070 puntos=0
2080 IF RND>0.5 THEN despX=1 ELSE despX=-1
2090 IF RND>0.5 THEN despY=1 ELSE despY=-1
2100 RETURN
```

Primeros PASOS

En las líneas 2080 y 2090 se puede observar cómo se determinará el sentido de desplazamiento de la bola de un modo aleatorio (función RND).

Ya tenemos el sistema, o los datos, inicializados, así que pasemos a la siguiente tarea que consistirá en sacar en la pantalla información sobre las teclas que tendremos que pulsar para conseguir el movimiento.

Se trata simplemente de una rutina de presentación en la que sólo vamos a utilizar instrucciones LOCATE y PRINT. El Programa III nos muestra una sencilla forma de hacerlo pero no se deje influenciar por ella, es mucho mejor que intente cambiarla y mejorarla experimentando nuevas técnicas. Es la mejor forma de aprender.

Programa tres

Ya hemos avanzado, ¿no? Corresponde ahora contemplar la posibilidad de cambiar las teclas asignadas al movimiento. Es otra rutina muy simple, pero en ella tendremos que investigar también si hemos pulsado o no alguna que otra tecla. Refresque su memoria.

Habrà que analizar si contestamos con una «S» o con cualquier otra letra a la pregunta que nos aparecerá en la pantalla. Cuando nuestra respuesta sea afirmativa, procederemos a cambiar la asignación de las teclas. Si por el contrario es negativa, seguiremos con las teclas iniciales.

También es conveniente analizar que se ha teclado uno de esos valores («S» o «N») pa-

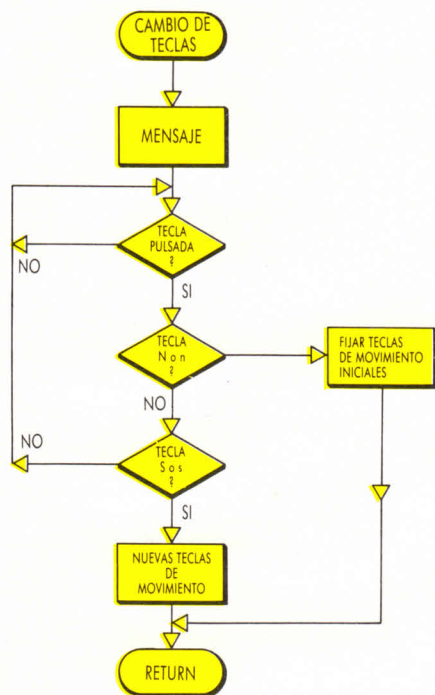
```
3000 REM PROGRAMA III
3010 REM PRESENTACION
3020 CLS
3030 LOCATE 5,9
3040 FOR i=1 TO 30
3050 PRINT" * ";
3060 NEXT i
3070 LOCATE 17,7
3080 PRINT"MANDOS"
3090 LOCATE 5,12
3100 PRINT"?: 17OUTERDA";TAB(25)"M: DERECHA"
3110 LOCATE 5,15
3120 FOR i=1 TO 30
3130 PRINT" * ";
3140 NEXT i
3150 RETURN
```

ra evitar posibles errores, por jugar con el teclado por ejemplo, que hagan que nuestro programa no funcione tan correctamente como es nuestro deseo.

¿Qué opina del organigrama que hemos pensado y qué responde a la Figura III? Se admiten mejoras.

Le sugerimos, como siempre, que intente codificar por sí mismo en instrucciones Basic este proceso lógico. Ya tiene los suficientes datos para hacerlo correctamente. Nosotros le proponemos una solución que no es la única,

FIG. III: Gráfico de la rutina de cambiar las teclas del movimiento.



pero que seguramente le sacará de ese atasco **«gordo»** en el que quizá se encuentra después de darle miles de vueltas a la cabeza sin encontrar una solución válida. Se trata del Programa IV. ¿Lo analizamos?

Programa cuatro

Lo que a nosotros nos afecta más son las distintas aplicaciones de la función INKEY\$ que hay a lo largo de la rutina. Por ejemplo, el bucle WHILE... WEND (líneas 4040 a 4060) nos sirve para almacenar en **«opcion\$»** el carácter generado por INKEY\$ mientras sea igual a la cadena vacía. En el momento que sea diferente, al pulsar nosotros una tecla, el programa sale del bucle y sigue la ejecución de la rutina.

A continuación se analiza el contenido de **«opcion\$»** para darle el tratamiento apropiado. Observe que la línea 4080 nos manda de nuevo al lugar donde elegimos una opción, otra vez, si el carácter tecleado no es ninguno de los válidos (**«S»** o **«N»**).

Empleamos nuevamente la función INKEY\$ en los dos bucles WHILE... WEND de las líneas 4130 y a 4150 y 4210 a 4230. En esta ocasión el programa se detiene hasta que pulsemos una tecla y los caracteres generados por esta función se almacenan respectivamente en las variables **«izquierda\$»** y **«derecha\$»** que más tarde servirán para comprobar si la paleta se ha de mover o no.

Imaginamos que no se le habrá pasado por alto la línea 4180. En ella está la instrucción

CLEAR INPUT que nos borra todos los caracteres tecleados que estén todavía almacenados. Intente quitar esta línea de la rutina y observe lo que ocurre. ¿Es válido?

```

4000 REM PROGRAMA IV
4010 REM CAMBIO DE MANDOS
4020 LOCATE 4,17
4030 PRINT"QUIERES OTRAS TECLAS PARA
A MOVER?"
4040 WHILE opcion$=""
4050 opcion$=INKEY$
4060 WEND
4070 IF opcion$="n" OR opcion$="N"
THEN izquierda$="z";derecha$="m":RE
TURN
4080 IF opcion$<>"s" AND opcion$<>"
S" THEN opcion$="":GOTO 4040
4090 LOCATE 4,17
4100 PRINT
"
4110 LOCATE 5,12
4120 PEN 2:PRINT"#"
4130 WHILE izquierda$=""
4140 izquierda$=INKEY$
4150 WEND
4160 LOCATE 5,12
4170 PRINT izquierda$
4180 CLEAR INPUT
4190 LOCATE 25,12
4200 PRINT"#"
4210 WHILE derecha$=""
4220 derecha$=INKEY$
4230 WEND
4240 LOCATE 25,12
4250 PRINT derecha$:PEN 1
4260 FOR i=1 TO 1000:NEXT i
4270 RETURN

```

Una vez hayamos cambiado (o no) las teclas de movimiento, podemos comenzar a construir ya el juego propiamente dicho. Construiremos el **«terreno»** con el Programa V.

Programa cinco

Dibujamos los bordes a base de bucles que visualizan el CHR\$(206) en líneas horizontales y verticales, líneas 5030 a 5080.

Después colocamos y visualizamos la paleta y la bola en sus posiciones iniciales —líneas 5090 5120. La primera está formada por una cadena de 3 caracteres de código 208, que nos devuelve la función STRING\$ de la línea 5120, la segunda es el carácter 202.

Y, cómo no, en el borde inferior aparecerán el número de bolas que nos quedan y la puntuación obtenida (5130 a 5160). Todo esto ya lo conocíamos.

```

5000 REM PROGRAMA V
5010 REM CUADRADO
5020 CLS
5030 FOR i=1 TO 40
5040 PRINT CHR$(206);
5050 NEXT i
5060 FOR i=1 TO 23
5070 PRINT CHR$(206);SPC(38);CHR$(2
06);
5080 NEXT i
5090 LOCATE xbola,ybola
5100 PRINT CHR$(202)
5110 LOCATE xpal,24
5120 PRINT STRING$(3,CHR$(208))
5130 LOCATE 4,25
5140 PRINT"PUNTOS:";puntos
5150 LOCATE 30,25
5160 PRINT"BOLAS:";bolas
5170 RETURN

```

Siguiendo el organigrama del programa principal podemos ver que la próxima rutina a desarrollar es la del movimiento.

El movimiento de la bola es semejante al que ya analizamos detenidamente en un artículo publicado en el número 43 de MICROHOBBY AMSTRAD. Le remitimos al Programa I de dicho artículo en el caso que crea oportuno un pequeño «refresco» de memoria. IF no es así, THEN podemos continuar.

Tenemos que hacer un par de modificaciones a este programa. Por un lado será necesario repetir el movimiento hasta que se nos hayan acabado las bolas (final de la partida). Pensemos en un bucle que se repita mientras una variable llamada **«bolas»**, por ejemplo, no sea cero.

Además habrá que detectar de alguna manera si hemos pulsado o no alguna de las teclas asignadas al movimiento y cuando haya sido así, saltar a la rutina que se encargue de desplazar la paleta.

Compruebe si el Programa VI se parece en algo a lo que usted había pensado.

Programa seis

Conseguimos estar jugando mientras tengamos bolas mediante el bucle WHILE... WEND de las líneas 1020 y 1130. Esta girando mientras el número de bolas que nos quedan —o el contenido de **«bolas»**— sea distinto de cero.

```

1000 REM PROGRAMA VI
1010 REM MOVIMIENTO DE LA BOLA
1020 WHILE bolas<>0
1030 LOCATE xbola,ybola
1040 PRINT CHR$(202)
1050 FOR i=0 TO 50:NEXT i
1060 LOCATE xbola,ybola
1070 PRINT" "
1080 xbola=xbola+despx
1090 ybola=ybola+despy
1100 GOSUB 1200:REM CAMBIO
1110 tecla$=INKEY$
1120 IF tecla$<>" " THEN GOSUB 1300
1130 WEND
1140 RETURN

```

Con la línea 1110 almacenamos en **«tecla\$»** el valor que nos da INKEY\$. En el caso que hayamos pulsado una tecla, dicha función nos devuelve al carácter asociado a la misma, así que saltaremos a la rutina de la línea 1300 (la del movimiento de la paleta) ya que **«tecla\$»** contendrá un valor distinto de la cadena vacía. Cuando no sea así, la rutina continuará ejecutándose.

Observe también que en la 1100 hacemos una llamada a la subrutina de la 1200 para cambiar el sentido de desplazamiento si es necesario.

Y nada más. ¿No le habrá parecido muy complicada la manera de hacer que la bola se mueva, verdad?

En cuanto al cambio de trayectoria, lo tenemos «chupado». El Programa I del número 43, como le dijimos antes, le da la solución ampliamente comentada. Una sola salvedad: ahora tendremos además que detectar si la bola choca o no contra la paleta al llegar al borde inferior.

Para ello en nuestro actual Programa VII incluimos una llamada a la rutina que se encargará de hacerlo.

Programa siete

Por lo demás todo es igual. Fíjese en la línea 1250 y comprenderá lo que queremos decir: hay un salto de llamada a la rutina que analiza si hay o no rebote y que tendrá su comienzo en la 1500.

```
1200 REM PROGRAMA VII
1210 REM CAMBIO TRAYECTORIA
1220 IF xbola=1 THEN xbola=2:despx=
despx*(-1)
1230 IF xbola=40 THEN xbola=39:desp
x=despx*(-1)
1240 IF ybola=1 THEN ybola=2:despy=
despy*(-1)
1250 IF ybola=24 THEN GOSUB 1500
1260 RETURN
```

Para mover la paleta no habrá encontrado ningún problema. Pero por si un casual, revise el Programa VIII y seguro que concuerda con su idea primitiva.

Programa ocho

Y llega la hora de ver si somos capaces de devolver o no la bola. La condición a investigar será que la coordenada «x» —contenida en la variable «xbola»— coincida con la de alguna de las partes de la paleta.

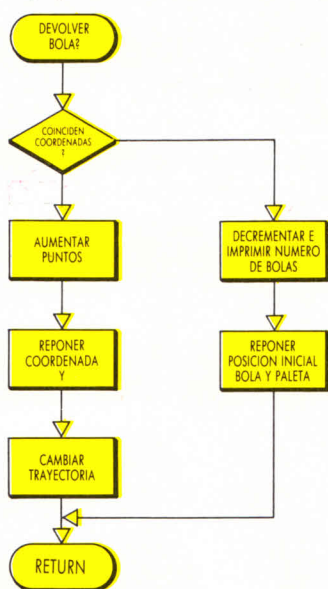
Cuando se cumpla la igualdad, el programa tomará un camino que logra que la bola rebote en el paleta y siga su trayectoria, que nuestra puntuación aumente y que continúe el juego.

Sin embargo, si no es así las acciones a realizar serán muy distintas. Ahora perderemos una pelota, por tanto el, número de bolas disminuirá en una, tendremos que visualizar esta situación y devolver tanto la pelota como la raqueta a su posición inicial para continuar jugando (siempre que esto sea posible).

```
1300 REM PROGRAMA VIII
1310 REM MOVIMIENTO PALETA
1320 LOCATE xpal,24
1330 PRINT " "
1340 IF tecla%=izquierda% THEN xpal=
xpal-1
1350 IF tecla%=derecha% THEN xpal=x
pal+1
1360 IF xpal=1 THEN xpal=2
1370 IF xpal=38 THEN xpal=37
1380 LOCATE xpal,24
1390 PRINT STRING$(3,CHR$(208))
1400 RETURN
```

Gráficamente este proceso quedaría reflejado en el organigrama de la figura IV.

FIG. IV: Organigrama rutina devolver bola.



Con un poco de paciencia imaginamos que verá la correspondencia entre cada una de las cajas de este gráfico y las instrucciones Basic del Programa IX.

Programa nueve

El primero de los caminos que hemos comentado será el comprendido entre las líneas 1530 y 1580. En esta última observará que existe un RETURN con el que volveremos a la instrucción siguiente al lugar desde donde hemos llamado a esta subrutina.

El otro sería el que abarca las líneas 1590 a 1680. Hay una ligera diferencia entre los dos, ¿verdad? Analice ambos y saque sus conclusiones.

```
1500 REM PROGRAMA IX
1510 REM DEVOLVER BOLA?
1520 IF xbola>=xpal+3 OR xbola<=xpa
l-1 THEN GOTO 1590
1530 puntos=puntos+1
1540 ybola=23
1550 despy=despy*(-1)
1560 LOCATE 11,25
1570 PRINT puntos
1580 RETURN
1590 bolas=bolas-1
1600 LOCATE 36,25
1610 PRINT bolas
1620 LOCATE xpal,24:
1630 PRINT " "
1640 xbola=20:ybola=12
1650 xpal=20
1660 LOCATE xpal,24
1670 PRINT STRING$(3,CHR$(208))
1680 RETURN
```

Y hablando de caminos, el nuestro ha llegado a su fin. Le deseamos que con sumo cuidado teclee cada uno de estos programas y después lo mezcle convenientemente (con MERGE, claro) para dar forma a uno solo que contenga todos los demás, cuidando el número de línea.

GANA 100.000 PESETAS CON MICROHOBBY AMSTRAD SEMANAL

Porque 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.

— No se admitirán programas que contengan caracteres de control, debido a que no son correctamente interpretados por las impresoras.

— 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.

¡ENVIANOS TU PROGRAMA!

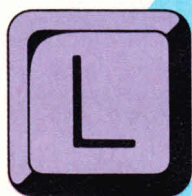
Adjuntando los siguientes datos:
Nombre y apellidos,
dirección y teléfono.

Indicando claramente en el sobre:

AMSTRAD SEMANAL
a **HOBBY PRESS, S. A. La Granja, 39**
Pol. Ind. Alcobendas (Madrid)

MAGNIFICADOR DE PANTALLAS

En un número anterior ofrecíamos un reductor de pantallas; en esta ocasión os presentamos un magnificador que trabaja en cualquiera de los modos de pantalla que posee el Amstrad.



La magnificación de pantallas puede ser una herramienta muy útil para el tratamiento de gráficos, ya que permite obtener de una figura relativamente pequeña, otra que es cuatro veces mayor, o bien, si ésta se magnifica de nuevo, se obtendría una figura que sería 16 veces mayor que la primitiva, y así sucesivamente si deseamos volver a magnificar.

De este modo, si en un programa realizado por nosotros, deseamos imprimir en pantalla una figura suficientemente grande, podemos archivar una figura de menor tamaño y en el momento de imprimirla, podemos magnificarla en pantalla.

Podemos preguntarnos qué ventajas ofrece esta operación. Pues bien, **la ventaja fundamental es el ahorro de memoria.** Si, por ejemplo, deseamos imprimir un gráfico que ocupe 2.000 bytes en condiciones normales, utilizando el sistema que proponemos, se utilizan únicamente 500 bytes. Así pues el ahorro de memoria es de 1.500 bytes.

Funcionamiento del magnificador

Vamos a ver a continuación cuál es el funcionamiento de esta rutina en

código máquina que provoca la magnificación de pantallas.

Lo primero que hacemos al entrar en el programa, es hacer una copia de la pantalla en un buffer situado en memoria; seguidamente, hacemos una llamada a la rutina del firmware que nos dará el modo de pantalla en el que nos encontramos, para adaptar las variables del programa a dicho modo.

Inmediatamente después entramos en el bucle principal del programa, que es el encargado de la lectura de teclado, y de enviar el programa a las diferentes partes de que consta según la tecla que se haya pulsado.

En primer lugar chequea la pulsación de las teclas de cursor, que serán las que permitirán mover el cuadrado por la pantalla, para poder decidir cuál es la zona de pantalla que se desea magnificar.

Así pues, si se pulsa la tecla de cursor arriba, se envía el programa a la rutina que repone la posición vertical de pantalla, incrementando dicha posición. Las restantes teclas de cursor actúan de la misma forma, incrementando las coordenadas correspondientes.

Seguidamente se produce la lectura de la tecla «COPIA», si ésta no está pulsada, no produce ningún efecto, si dicha tecla se encuentra pulsada, hacemos una llamada a la rutina encargada de producir la magnificación de la pantalla propiamente dicha, y una vez hecha esta operación se retorna al bucle principal del programa.

Otra de las teclas que se chequean, es la tecla «P», si ésta está pulsada, el programa toma la pantalla original del buffer de memoria y la coloca en la pantalla visible, perdiendo de este modo el contenido de la pantalla anterior.

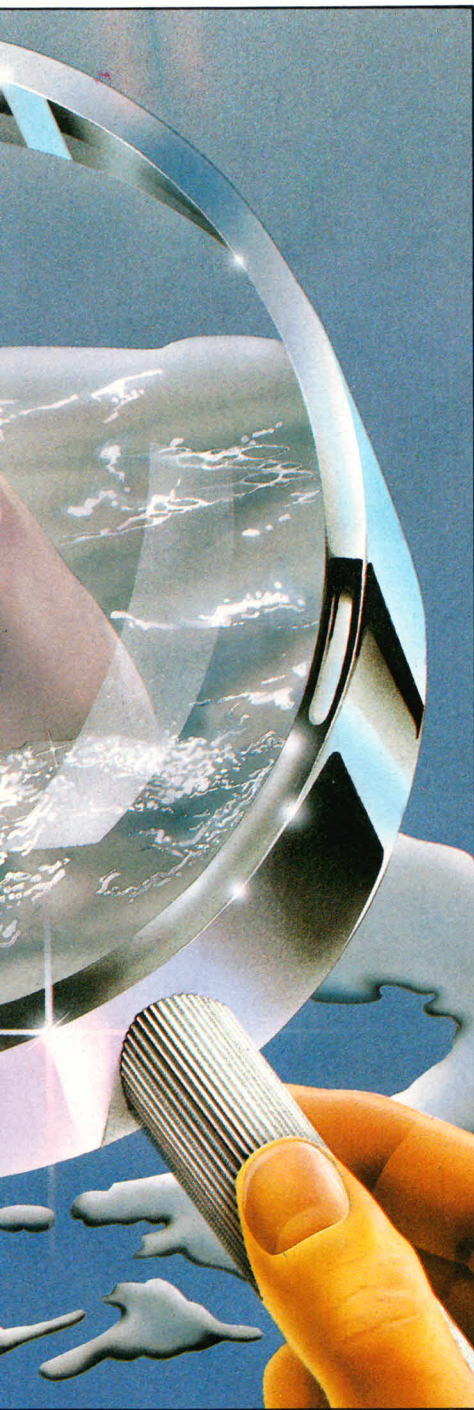


La imagen ampliada se puede salvar en cinta/disco

Existe una tecla que nos permitirá salvar en disco o cassette la pantalla actual, esta tecla es la «S».

Por último también se chequea la pulsación de la tecla «ESC», para la posibilidad de retornar al Basic.

Vamos a centrarnos ahora en la rutina que produce la magnificación, intentaremos ver cuál es el efecto que se produce.



que componen el byte, ya que todos ellos llevan información de pixels encendidos o apagados.

Así pues, **observamos cada uno de los bits** mencionados anteriormente; cuando uno esté a 1 (encendido), enviaremos el programa a una rutina especializada en agrandar horizontalmente dicho bit, así pues, obtendremos un bit doble que el original (o sea se pondrán dos bits a 1).

Si por el contrario dicho bit se encuentra a 0 (apagado), el programa se enviará a una rutina especializada en hacer dobles dichos ceros, de este modo se pondrán dos bits a cero.

Una vez agrandados horizontal-

PROGRAMACION

mente cada uno de los bits, se procederá a efectuar el agrandamiento de forma vertical, para lo cual se efectuará una copia de la primera línea de pantalla en la segunda, con lo que se habrá conseguido el efecto de alargamiento.

Si se está trabajando en **modo 1**, se observarán también los ocho bits de que se compone cada byte, pero

MAGNISCA BB

```

10 REM PROGRAMA CARGADOR
20 FOR N=#A000 TO #A2BC
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<>#2159 THEN PRINT "ERRO
R EN DATAS"
70 DATA 33,160,31,34,142,162,62
80 DATA 200,50,108,161,62,80,50
90 DATA 135,161,33,160,95,34,144
100 DATA 162,33,0,192,205,103,161
110 DATA 62,100,50,108,161,62,40
120 DATA 50,135,161,33,0,16,34
130 DATA 144,162,205,17,188,50,122
140 DATA 162,254,1,40,8,254,2
150 DATA 40,8,62,12,24,6,62
160 DATA 22,24,2,62,42,50,158
170 DATA 160,33,5,5,34,156,160
180 DATA 62,0,205,30,187,40,3
190 DATA 205,159,160,62,2,205,30
200 DATA 187,40,3,205,173,160,62
210 DATA 0,205,30,187,40,3,205
220 DATA 204,160,62,1,205,30,187
230 DATA 40,3,205,188,160,62,9
240 DATA 205,30,187,40,3,205,240
250 DATA 160,62,27,205,30,187,40
260 DATA 3,205,123,162,62,60,205
270 DATA 30,187,40,3,205,146,162
280 DATA 62,66,205,30,187,192,24
290 DATA 178,201,0,0,42,42,156
300 DATA 160,45,125,167,200,34,156
310 DATA 160,205,218,160,201,42,156
320 DATA 160,44,125,254,15,200,34
330 DATA 156,160,205,218,160,201,42
340 DATA 156,160,36,58,158,160,188
350 DATA 200,34,156,160,205,218,160
360 DATA 201,42,156,160,37,124,167
370 DATA 200,34,156,160,205,218,160
380 DATA 201,42,156,160,205,117,187
390 DATA 205,129,187,205,231,160,20
1
400 DATA 1,40,35,11,120,177,32
410 DATA 251,201,205,132,187,42,156
420 DATA 160,37,45,205,252,160,201
430 DATA 205,100,161,205,6,161,205
440 DATA 50,161,201,17,160,31,33
450 DATA 0,16,58,122,162,254,2
460 DATA 202,150,161,254,1,202,226
470 DATA 161,195,46,162,0,0,203
480 DATA 199,203,207,201,203,215,20
3
490 DATA 223,201,203,231,203,239,20
1
500 DATA 203,247,203,255,201,33,0
510 DATA 192,221,42,142,162,6,200
520 DATA 24,21,124,230,56,254,56
530 DATA 40,6,124,198,8,103,24
540 DATA 8,17,80,0,124,238,56
550 DATA 103,25,197,229,6,80,221
560 DATA 126,0,119,221,35,35,16
570 DATA 247,225,193,16,218,201,205
580 DATA 26,188,221,42,144,162,6
590 DATA 100,24,21,124,230,56,254
600 DATA 56,40,6,124,198,8,103
610 DATA 24,8,17,80,0,124,238
620 DATA 56,103,25,197,229,6,40
630 DATA 126,221,119,0,221,35,35
640 DATA 16,247,225,193,16,218,201
650 DATA 6,100,197,237,83,28,161
660 DATA 6,40,175,203,126,196,45
670 DATA 161,203,118,196,40,161,203
680 DATA 110,196,35,161,203,102,196
690 DATA 30,161,18,175,19,203,94
700 DATA 196,45,161,203,86,196,40
710 DATA 161,203,78,196,35,161,203
720 DATA 70,196,30,161,18,19,35
730 DATA 16,207,229,42,28,161,1
740 DATA 80,0,237,176,225,193,16
750 DATA 187,237,83,28,161,201,6
760 DATA 100,197,237,83,28,161,6
770 DATA 40,175,203,126,196,45,161
780 DATA 203,118,196,40,161,203,94
790 DATA 196,35,161,203,86,196,30
800 DATA 161,18,175,19,203,110,196
810 DATA 45,161,203,102,196,40,161
820 DATA 203,78,196,35,161,203,70
830 DATA 196,30,161,18,19,35,16
840 DATA 207,229,42,28,161,1,80
850 DATA 0,237,176,225,193,16,187
860 DATA 237,83,28,161,201,6,100
870 DATA 197,237,83,28,161,6,40
880 DATA 175,203,126,196,45,161,203
890 DATA 110,196,40,161,203,94,196
900 DATA 35,161,203,78,196,30,161
910 DATA 18,175,19,203,118,196,45
920 DATA 161,203,102,196,40,161,203
930 DATA 86,196,35,161,203,70,196
940 DATA 30,161,18,19,35,16,207
950 DATA 229,42,28,161,1,80,0
960 DATA 237,176,225,193,16,187,237
970 DATA 83,28,161,201,0,205,132
980 DATA 187,33,160,95,34,142,162
990 DATA 205,50,161,33,160,31,34
1000 DATA 142,162,201,0,0,0,0
1010 DATA 62,1,205,107,188,6,8
1020 DATA 33,180,162,17,0,169,205
1030 DATA 140,188,33,0,192,17,0
1040 DATA 64,1,0,0,62,2,205
1050 DATA 152,188,205,143,188,201,8
0
1060 DATA 65,78,84,46,65,83,82
1070 DATA 0

```

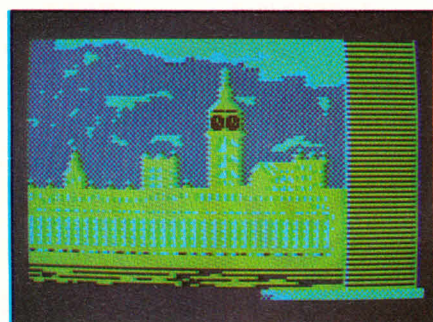
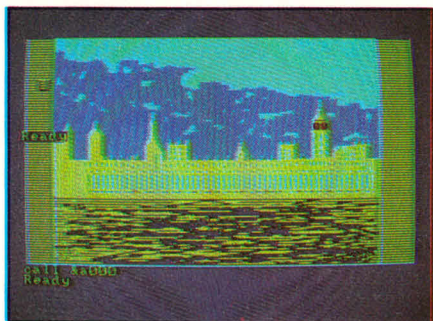
Dado que cada modo de pantalla debe tratarse de diferente forma, lo primero que hacemos es ver cuál es el modo de pantalla en que se está trabajando: para ello observamos cuál es el valor de la variable «modo» y enviamos el control del programa a la rutina especializada en dicho modo.

En detalle

Para trabajar en modo 2, debemos observar cada uno de los bits



P ara que tus dedos no realicen el trabajo duro, M.H. AMS-TRAD lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un cassette mensual, solicitanoslo.



en este caso se tratarán de diferente forma, ya que **los bits que dan información** de encendido o apagado, únicamente **son cuatro**, ya que los otros cuatro se utilizan para colocar los diferentes colores que se utilizan en dicho modo.

En **modo 0**, únicamente llevan información de pantalla dos de los ocho bits que componen el byte, ya que **los otros seis llevan información del color**. Así pues, también se producirán distinto tratamiento que en los anteriores modos.

La opción de recuperar la pantalla inicial, únicamente consta de una rutina de impresión, a través de la cual se procede a situar en la pantalla visual, la pantalla que se tenía almacenada en memoria.

La última opción de que disponemos, es la que nos permite salvar la pantalla actual en disco o cinta, esta opción consta únicamente de varias llamadas al firmware, que provocan el efecto de «SAVE».

Vamos a ver a continuación cómo podemos utilizar el programa indicando cuáles son las teclas correspondientes a cada una de las opciones.

Controles del magnificador

Arriba	Cursor arriba
Abajo	Cursor abajo
Derecha	Cursor derecha
Izquierda	Cursor izquierda
Magnificar	COPIA
Pantalla original	P
Salvar pantalla	S

Una vez que hayamos ejecutado el programa, para proceder a la magnificación de un trozo de pantalla, deberemos colocar el cursor en la parte superior izquierda del bloque de pantalla que se desea magnificar.

Una vez decidida dicha posición, pulsaremos la tecla «COPIA» para obtener el agrandamiento de dicho trozo de pantalla.

Si una vez efectuada dicha operación, no se obtiene el efecto deseado, podremos volver a la pantalla original, con sólo pulsar la tecla «P».

En el caso de que la magnificación se haya producido, si deseamos salvar dicha pantalla en disco o cassette bastará con pulsar la tecla «S».

Usando imágenes grabadas desde Basic

Para poder trabajar con una pantalla que está almacenada en disco o cassette, podemos actuar del siguiente modo:

1. Cargaremos en memoria la rutina de magnificación de pantalla.
2. Escribiremos un programa Basic como el que ofrecemos a continuación.

```
10 MODE N
20 LOAD "PANTALLA", &C000
30 CALL &A000
```

donde «N» será el modo de pantalla en el cual deseamos trabajar o en modo de pantalla en el cual está salvada dicha pantalla en disco o cinta.

Si en cualquier momento decidimos dejar de trabajar con la pantalla actual y deseamos cargar otra, únicamente **deberemos pulsar la tecla «ESC»** y volver a ejecutar el anterior programa Basic, cambiando el nombre de la pantalla a cargar.

Como hemos dicho anteriormente en cualquier momento estamos en condiciones de salvar la pantalla con la que se está trabajando.

Para poder salvar dicha rutina en cinta o disco, únicamente deberemos copiar el listado ensamblador que aparece al final del artículo.

Para aquellos que no posean ensamblador, pueden copiar el listado del programa cargador y ejecutarlo, una vez hecho esto si no nos ha dado ningún error, podemos salvar el programa de la forma siguiente:

```
SAVE "MAGNI", B, &A000, 700
```

Para cargar dicha rutina en memoria en el momento en que deseemos trabajar, deberemos ejecutar el siguiente programa Basic:

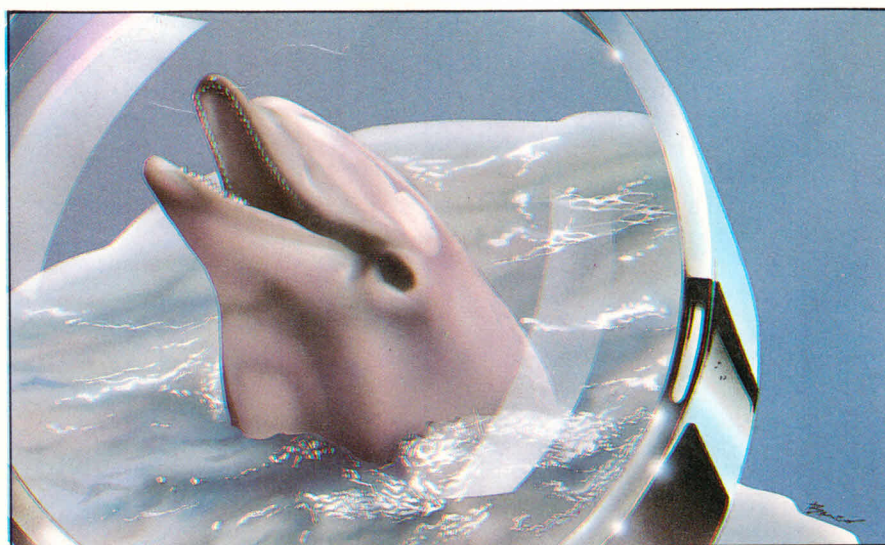
```
10 MEMORY &FFF
20 LOAD "MAGNI", &A000
```

Una vez hecho esto, estaremos en condiciones de ejecutar el primero de los programas Basic que hemos propuesto.

Si cualquiera de las pantallas que hemos almacenado de forma magnificada se desea retocar para limpiarla de las brusquedades propias de la magnificación, deberemos cargarla en el diseñador de gráficos de que dispongamos para dejarla finalmente tal como se desee.

Debemos decir también que utilizando esta rutina de magnificación, no podremos tener nada en memoria a partir de la dirección &FFF, ya que a partir de esa dirección es donde se almacenan los buffers de la pantalla actual y de la pantalla inicial.

Así pues a partir de esa dirección y con una longitud de 16.384 bytes está almacenada la pantalla inicial. Existe otro buffer que es donde se almacena la pantalla magnificada.



LISTADO DE ENSAMBLADO

Hisoft GENA3.1 Assembler. Page 1.

Pass 1 errors: 00

```

A000      10      DRG      #A000
A000 21A01F      20      LD      HL,#1FA0
A003 22BEA2      30      LD      (PANTX),HL
A006 3ECB        40      LD      A,200
A008 326CA1      50      LD      (ETIL+1),A
A00B 3E50        60      LD      A,B0
A00D 32B7A1      70      LD      (ETIN+1),A
A010 21A05F      80      LD      HL,#SFA0
A013 2290A2      90      LD      (PANTY),HL
A016 2100C0     100      LD      HL,#C000
A019 CD67A1     110      CALL   LEERE
A01C 3E64        120      LD      A,100
A01E 326CA1     130      LD      (ETIL+1),A
A021 3E2B        140      LD      A,40
A023 32B7A1     150      LD      (ETIN+1),A
A026 210010     160      LD      HL,#1000
A029 2290A2     170      LD      (PANTY),A
A02C CD11BC     180      CALL   #BC11
A02F 327AA2     190      LD      (MDDO),HL
A032 FE01        200      CP      1
A034 2B08        210      JR      Z,PAMDD1
A036 FE02        220      CP      2
A03B 2B08        230      LD      Z,PAMDD2
A03A 3E0C        240      LD      A,12
A03C 1B06        250      JR      PDLNAR
A03E 3E16        260      PAMDD1: LD A,22
A040 1B02        270      JR      PDLNAR
A042 3E2A        280      PAMDD2: LD A,42
A044 329EAO     290      PDLNAR: LD (LARGA),A
A047 210505     300      CALL   HL,#C0505
A04A 229CA0     310      LD      (POSIC),HL
A04D 3E00        320      TEC:  LD A,0
A04F CD1EBB     330      CALL   #BB1E
A052 2B03        340      JR      Z,TEC1
A054 CD9FA0     350      CALL   UP
A057 3E02        360      TEC1:  LD A,2
A059 CD1EBB     370      CALL   #BB1E
A05C 2B03        380      JR      Z,TEC2
A05E CDADA0     390      CALL   DOWN
A061 3E08        400      TEC2:  LD A,B
A063 CD1EBB     410      CALL   #BB1E
A066 2B03        420      JR      Z,TEC3
A068 CDCCA0     430      CALL   LEFT
A06B 3E01        440      TEC3:  LD A,1
A06D CD1EBB     450      CALL   #BB1E
A070 2B03        460      JR      Z,TEC4
A072 CD8CA0     470      CALL   RIGHT
A075 3E09        480      TEC4:  LD A,9
A077 CD1EBB     490      CALL   #BB1E
A07A 2B03        500      JR      Z,TEC5
A07C CDFOAO     510      CALL   COPIA
A07F 3E1B        520      TEC5:  LD A,27
A081 CD1EBB     530      CALL   #BB1E
A084 2B03        540      JR      Z,TEC6
A086 CD7BA2     550      CALL   RECUP
A089 3E3C        560      TEC6:  LD A,60
A08B CD1EBB     570      CALL   #BB1E
A08E 2B03        580      JR      Z,TEC7
A090 CD92A2     590      CALL   SAVE
A093 3E44        600      TEC7:  LD A,66
A095 CD1EBB     610      CALL   #BB1E
A098 C0          620      RET      NZ
A099 1B02        630      JR      TEC
A09B C9          640      RET
A09E 2A          650      POSIC: DEFS 2
A09F 2A9CA0     660      LARGA: DEFB 42
A0A2 20          680      UP:    LD HL,(POSIC)
A0A3 7D          690      DEC      L
A0A4 07          700      AND      A
A0A5 C7          710      RET      Z
A0A8 2A9CA0     720      LD      (POSIC),HL
A0AA CDAAA0     730      CALL   PINT
A0AC C9          740      RET
A0AD 2A9CA0     750      DOWN: LD HL,(POSIC)
A0B0 7C          760      INC      L
A0B1 7D          770      LD      A,L
A0B2 C2F0F      780      CP      15
A0B4 7B          790      RET      Z
A0B5 2A9CA0     800      LD      (POSIC),HL
A0B8 CDAAA0     810      CALL   PINT
A0B9 C9          820      RET
A0BB 2A9CA0     830      RIGHT: LD HL,(POSIC)
A0BC 7A          840      INC      H
A0BD 7A9EAO     850      LD      A,(LARGA)
A0C3 BC          860      CP      H
A0C4 C8          870      RET      Z
A0C7 2A9CA0     880      LD      (POSIC),HL
A0C8 7DAAA0     890      CALL   PINT
A0CB C7          900      RET
A0CC 2A9CA0     910      LEFT:  LD HL,(POSIC)
A0CE 2E          920      DEC      H
A0D0 7C          930      LD      A,H
A0D1 07          940      AND      A
A0D2 7B          950      RET      Z
A0D4 2A9CA0     960      LD      (POSIC),HL
A0D6 CDAAA0     970      CALL   PINT
A0D9 C9          980      RET
A0DA 2A9CA0     990      PINT:  LD HL,(POSIC)
A0DD C775EB     1000     CALL   #BB75
A0FE CDB1BE     1010     CALL   #BB1
A0FF CDF700     1020     CALL   FAUSA
A0FA C9          1030     RET
A0FB 012P23    1040     PAUSA: LD BC,9000
A0FC 0B          1050     PAUS: DEC BC
A0FD 7B          1060     LD      A,B
A0FE B1          1070     OR      C
A0FF 20FF      1080     JR      NZ,PAUS
A100 C9          1090     RET
A102 CDB4BB     1100     COPIA: CALL #BBB4
A103 2A9CA0     1110     LD      HL,(POSIC)
A104 2E          1120     DEC      H
A107 2D          1130     DEC      L
A108 CDFCA0     1140     CALL   MAGNI
A10B C9          1150     RET
A10C CD64A1    1160     MAGNI: CALL LEER
A10F CD06A1    1170     CALL   MIRA
A102 CD32A1    1180     CALL   IMPRE

```

```

A105 C9          1190     RET
A106 11A01F    1200     MIRA:  LD DE,#1FA0
A109 210010    1210     LD      HL,#1000
A10C 3A7AA2    1220     LD      A,(MDDO)
A10F FE02      1230     CP      2
A111 CA96A1    1240     JP      Z,MODE2
A114 FE01      1250     CP      1
A116 CAE2A1    1260     JP      Z,MODE1
A119 C32EA2    1270     JP      M,MODE0
A11C          1280     POSIN: DEFS 2
A11E CBC7      1290     POSIN: SET 0,A
A120 C8CF      1300     SET 1,A
A122 C9        1310     RET
A123 C8D7      1320     PON23: SET 2,A
A125 C8DF      1330     SET 3,A
A127 C9        1340     RET
A128 C8E7      1350     PON45: SET 4,A
A12A C8EF      1360     SET 5,A
A12C C9        1370     RET
A12D C8F7      1380     PON67: SET 6,A
A12F C8FF      1390     SET 7,A
A131 C9        1400     RET
A132 2100C0    1410     ;RUTINA-IMPRESION
A135 DD2AFA2   1420     ;H-POSICION-VERTICAL-INITIO-1
A137 7C        1430     ;L-POSICION-HORIZONTAL-INITIO-1
A139 06C8      1440     ;DE-DIRECCION-GRAFICO
A13B 1B15      1450     ;
A13D 7C        1460     IMPRE: LD HL,#C000
A13E E638      1470     LD      IX,(PANTX)
A140 FE38      1480     LD      B,200
A142 2B06      1490     JR      CDLDC
A144 7C        1500     P_BUC: AND 56
A145 C608      1510     AND 56
A147 67        1520     CP      56
A148 1B08      1530     JR      Z,P_PAS
A149 7C        1540     LD      A,H
A14B EE38      1550     YOR 56
A14D 67        1560     LD      H,A
A14E 19        1570     ADD    HL,DE
A14F 115000    1580     P_PAS: LD DE,00B0
A150 67        1590     LD      A,H
A151 19        1600     LD      H,A
A152 C5        1610     COLDC: ADD HL,DE
A153 E5        1620     PUSH   HL
A154 0650      1630     LD      D,B0
A155 DD7E00    1640     P_BUC1: LD A,(IX+0)
A156 77        1650     LD      (HL),A
A15A DD23      1660     INC IX
A15C 23        1670     INC HL
A15D 10F7      1680     DJNZ P_BUC1
A15F E1        1690     POP    HL
A160 C1        1700     POP    BC
A161 10DA      1710     DJNZ P_BUC
A163 C9        1720     RET
A164 CD1ABC    1730     LEER:  CALL #BC1A
A167 DD2A90A2  1740     LEERE: LD B,100
A168 0664      1750     ETIL: LD B,100
A169 1B15      1760     LD      L,B
A170 E338      1770     L_DUS: LD A,H
A171 E338      1780     AND 56
A172 FE38      1790     LD      B,100
A174 2B06      1800     JR      Z,L_PAS
A176 7C        1810     LD      A,H
A177 C608      1820     ADD    A,B
A179 67        1830     LD      H,A
A17A 1B08      1840     LD      H,A
A17C 115000    1850     L_PAS: JR LEETE
A17F 7C        1860     LD      A,H
A180 EE38      1870     YOR 56
A182 67        1880     LD      H,A
A183 19        1890     LD      H,A,DE
A184 C5        1900     LEETE: PUSH BC
A185 E5        1910     PUSH   HL
A186 0628      1920     ETIN: LD B,40
A188 7E        1930     L_DUS1: LD A,(HL)
A189 DD7700    1940     LD      (IX+0),A
A18C DD23      1950     INC IX
A18E 23        1960     INC HL
A18F 10F7      1970     DJNZ L_DUS1
A191 E1        1980     POP    HL
A192 C1        1990     POP    BC
A193 10DA      2000     DJNZ L_DUS
A195 C9        2010     RET
A196 0664      2020     M,MODE2: LD B,100
A198 C5        2030     LLL2:  PUSH   BC
A199 ED531CA1  2040     LD      (POSIN),DE
A19D 0628      2050     LD      B,40
A19F AF        2060     LLL1:  XOR A
A1A0 C87E      2070     BIT 7,(HL)
A1A2 C42DA1    2080     CALL   NZ,PON67
A1A5 C876      2090     BIT 6,(HL)
A1A7 C42BA1    2100     CALL   NZ,PON45
A1AA C86E      2110     BIT 5,(HL)
A1AC C423A1    2120     CALL   NZ,PON23
A1AF C866      2130     BIT 4,(HL)
A1B1 C41EA1    2140     CALL   NZ,PON1
A1B4 12        2150     LD      (DE),A
A1B5 AF        2160     XOR A
A1B6 13        2170     INC DE
A1B7 C85E      2180     BIT 3,(HL)
A1B9 C42A1    2190     CALL   NZ,PON67
A1BC C85A      2200     CALL 2,(HL)
A1BE C42BA1    2210     CALL   NZ,PON45
A1C1 C84E      2220     BIT 1,(HL)
A1C3 C423A1    2230     CALL   NZ,PON23
A1C6 C846      2240     BIT 0,(HL)
A1C8 C41EA1    2250     CALL   NZ,PON1
A1CC 12        2260     LD      (DE),A
A1CD 23        2270     INC DE
A1CE 10CF      2280     DJNZ LLL1
A1D0 E5        2290     PUSH   HL
A1D1 2A1CA1    2300     LD      HL,(POSIN)
A1D4 015000    2310     LD      BC,B0
A1D7 E8B0      2320     LDIR
A1D9 E1        2330     POP    HL
A1DA C1        2340     POP    BC
A1DB 10BB      2350     DJNZ LLL2
A1DD ED531CA1  2360     LD      (POSIN),DE
A1E1 C9        2370     RET

```

```

A1E2 0664      2410     MODE1: LD B,100
A1E4 C5        2420     PUSH   BC
A1E5 ED531CA1  2430     LD      (POSIN),DE
A1E9 0628      2440     LD      B,40
A1EB AF        2450     M,MODE1: XOR A
A1EC C87E      2460     BIT 7,(HL)
A1EE C42DA1    2470     CALL   NZ,PON67
A1F1 C876      2480     BIT 6,(HL)
A1F3 C42BA1    2490     CALL   NZ,PON45
A1F6 C85E      2500     BIT 5,(HL)
A1FB C423A1    2510     CALL   NZ,PON23
A1FD C856      2520     BIT 2,(HL)
A1FD C41EA1    2530     CALL   NZ,PON1
A200 12        2540     LD      (DE),A
A201 AF        2550     XOR A
A202 13        2560     INC DE
A203 C86E      2570     BIT 5,(HL)
A205 C42DA1    2580     CALL   NZ,PON67
A208 C866      2590     BIT 4,(HL)
A20A C42BA1    2600     CALL   NZ,PON45
A20D C84E      2610     BIT 1,(HL)
A20F C423A1    2620     CALL   NZ,PON23
A212 C846      2630     BIT 0,(HL)
A214 C41EA1    2640     CALL   NZ,PON1
A217 12        2650     LD      (DE),A
A218 13        2660     INC DE
A219 23        2670     INC HL
A21A 10CF      2680     DJNZ M,MODE1
A21C C5        2690     PUSH   HL
A21D 2A1CA1    2700     LD      HL,(POSIN)
A220 015000    2710     LD      BC,B0
A223 E8B0      2720     LDIR
A225 E1        2730     POP    HL
A226 C1        2740     POP    BC
A227 10BB      2750     DJNZ M,MODE1
A229 ED531CA1  2760     LD      (POSIN),DE
A22D C9        2770     RET
A22E 0664      2780     MODE0: LD B,100
A230 C5        2790     P,MODE2: PUSH BC
A231 ED531CA1  2800     LD      (POSIN),DE
A235 0628      2810     LD      B,40
A237 AF        2820     P,MODE1: XOR A
A23B C87E      2830     BIT 7,(HL)
A23A C42DA1    2840     CALL   NZ,PON67
A23D C86E      2850     BIT 5,(HL)
A23F C42BA1    2860     CALL   NZ,PON45
A242 C856      2870     BIT 2,(HL)
A244 C423A1    2880     CALL   NZ,PON23
A247 C84E      2890     BIT 1,(HL)
A249 C41EA1    2900     CALL   NZ,PON1
A24C 12        2910     LD      (DE),A
A24D AF        2920     XOR A
A24E 13        2930     INC DE
A24F C876      2940     BIT 6,(HL)
A251 C42DA1    2950     CALL   NZ,PON67
A254 C866      2960     BIT 4,(HL)
A256 C42BA1    2970     CALL   NZ,PON45
A259 C856      2980     BIT 2,(HL)
A25B C423A1    2990     CALL   NZ,PON23
A25E C84E      3000     BIT 0,(HL)
A260 C41EA1    3010     CALL   NZ,PON1
A263 12        3020     LD      (DE),A
A264 13        3030     INC DE
A265 23        3040     INC HL
A266 10CF      3050     DJNZ P,MODE1
A268 E5        3060     PUSH   HL
A269 2A1CA1    3070     LD      HL,(POSIN)
A26C 015000    3080     LD      BC,B0
A26F E8B0      3090     LDIR
A271 E1        3100     POP    HL
A272 C1        3110     POP    BC
A273 10BB      3120     DJNZ P,MODE1
A275 ED531CA1  3130     LD      (POSIN),DE
A279 C9        3140     RET
A27A 3150      3150     M,MODE2: DEFS 1
A27B CDB4BB    3160     RECUP: CALL #BBBA
A27E 21A05F    3170     LD      HL,#SFA0
A281 22BEA2    3180     LD      (PANTX),HL
A284 C832A1    3190     CALL   IMPRE
A287 21A01F    3200     LD      HL,#1FA0
A28A 22BEA2    3210     LD      (PANTX),HL
A28D C9        3220     RET
A28E C84E      3230     PANTY: DEFS 2
A290          3240     PANTY: DEFS 2
A292 3E01      3250     SAVE: LD A,1
A294 CDB8BC    3260     CALL   #BC6B
A297 0608      3270     LD      B,B
A299 21B4A2    3280     LD      HL,NAME
A29C 1100A9    3290     LD      DE,#A900
A29F CDB8BC    3300     CALL   #BC6C
A2A2 2100C0    3310     LD      HL,#C000
A2A5 1100A0    3320     LD      DE,#A000
A2A8 010000    3330     LD      BC,0
A2AB 3E02      3340     LD      A,2
A2AD CDB8BC    3350     CALL   #BC6C
A2B0 CDB8BC    3360     CALL   #BC6B
A2B3 C9        3370     RET
A2B4 50414E54  3380     NAME:  DEFM "PANT.ASR"

```

Table used: 661 from 1000

RELOJ ANALOGICO

José Espinosa Rodríguez

El programa, bien resuelto, nos ofrece la posibilidad de utilizar un Reloj funcional. Este cuenta con agujas para las horas, minutos y segundos. Funciona sin usar la sentencia Time.

Variables principales

10-40 Presentación.

40-90 Deberemos introducir las horas, los minutos y los segundos para poder lograr dejar el reloj en la hora que deseemos.

90-140 Dibuja exteriormente el reloj.

140 Se realiza la adaptación de horas, minutos y segundos a los 360 grados que tiene la circunferencia.

150 Creamos una ventana.

160-250 Realizamos los bucles necesarios para hacer funcionar el reloj. Le ponemos sonido y dibujamos el Reloj Interior.

Estructura del programa

- d Horas.
- e Minutos.
- f Segundos.
- c Horas adaptadas a los 360°.
- b Minutos adaptados a los 360°.
- a Segundos adaptados a los 360°.

Manejo

El manejo no implica ninguna dificultad. Cuando arrancamos el programa, se nos pide las horas y debemos dar un número comprendido entre el 0 y el 12.

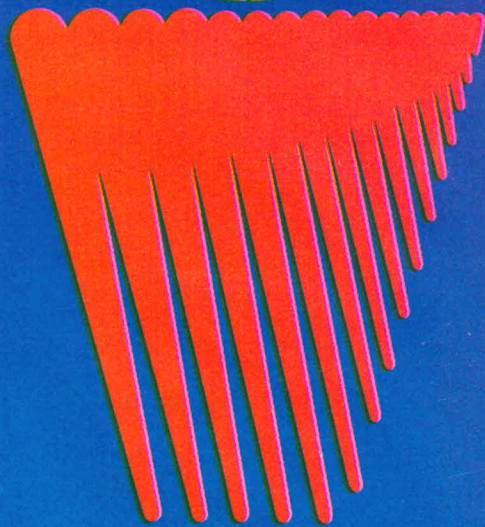
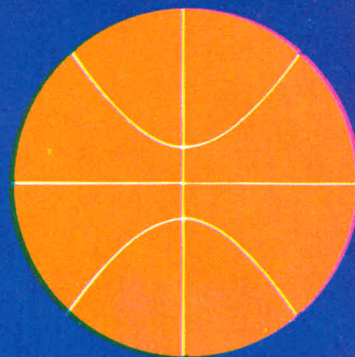
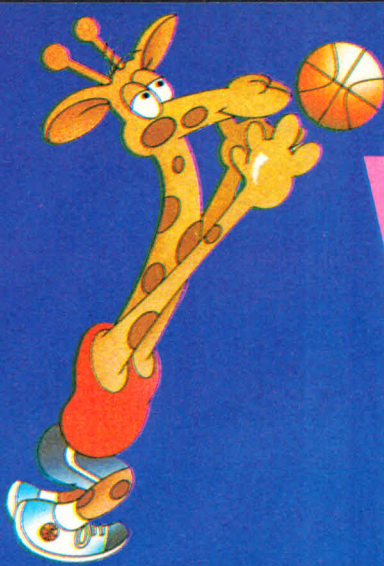
Cuando se nos pide los minutos, el número que debemos dar estará comprendido entre el 0 y el 60. Igual para los segundos.

Una vez que hallamos hecho esto nuestro reloj estará puesto en la hora apetecida.



```

10 MODE 0:REM "agujas"
20 FILL 2:LOCATE 1,5:PRINT " RELOJ
FUNCIONAL,":PRINT:PRINT" POR JOSE E
SPINOSA "
30 PRINT:PRINT
40 PRINT:PRINT
50 PRINT:PRINT:PRINT:INPUT "hora: "
,d
60 PRINT:PRINT:PRINT:INPUT "minuto:
",e
70 PRINT:PRINT:PRINT:INPUT "segundo
":",f
80 MODE 1:FILL 1
90 FOR y=1 TO 25
100 LOCATE 3,y:PRINT STRING$(4,134)
;
110 LOCATE 35,y:PRINT STRING$(4,134
):NEXT
120 TAG:MOVE 306,385:PRINT "12":MO
VE 415,385:PRINT CHR$(138):MOVE 49
8,294:PRINT CHR$(140):MOVE 498,207
:PRINT "3":
130 MOVE 498,112:PRINT CHR$(140):M
OVE 415,29:PRINT CHR$(138):MOVE 31
2,29:PRINT "6":MOVE 220,29:PRINT C
HR$(138):MOVE 127,112:PRINT CHR$(1
40):MOVE 127,207:PRINT "9":MOVE 1
27,294:PRINT CHR$(140):MOVE 212,38
5:PRINT CHR$(138):TAGOFF
140 c=(d*30)+(e/2):b=e*6:a=f*6
150 WINDOW 10,30.5,2.9,22.5:PAPER 1
1:CLS
160 WHILE b<359
170 WHILE a<359
180 SOUND 1,0,100.499999,15:a=a+6:O
RIGIN 320,200:DEG:DRAW 160*SIN(b),1
60*COS(b)
190 ORIGIN 320,200:DEG:DRAW 120*SIN
(c),120*COS(c):ORIGIN 320,200:DEG:P
LOT 160*SIN(a),160*COS(a)
200 WEND
210 b=b+6:c=c+0.5:IF a=360 THEN CLS
220 a=0:ORIGIN 320,200:DEG:DRAW 160
*SIN(b),160*COS(b):ORIGIN 320,200:D
EG:DRAW 120*SIN(c),120*COS(c):PLOT
160*SIN(a),160*COS(a):IF b=360 THEN
240
230 GOTO 170
240 a=0:WEND
250 GOTO 170
    
```



Vive El Mundial con

GIGANTES del Basket



¡Pídela
en tu
kiosco!

La
única revista
de basket que te
hará vivir, semana a
semana, este gran aconte-
cimiento deportivo.
Información caliente, actual, diná-
mica... todo lo que te interesa saber de
las grandes figuras y las grandes jugadas lo
encontrarás en «Gigantes». Y sólo por 150 pesetas.

ORDEN EN EL CAOS CON DBASEI

Dbasel es un programa de creación y posterior tratamiento de ficheros indexados. El programa reside íntegramente en memoria y deja libres 13 k de memoria aproximadamente

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



El programa tiene todas las opciones de una base de datos «profesional» aunque es totalmente comprensible para cualquiera que quiera dedicar un poco de tiempo en teclear el listado, con lo cual comprenderá la estructura del mismo y podrá modificarlo y amoldarlo a sus necesidades.

En efecto, una de las principales características del programa es que, si bien ocupa una gran parte de memoria al residir totalmente en ésta, es adaptable a distinto tipo de usos; basta cambiar las dimensiones de las tres matrices que se utilizan.

Por ejemplo, si el máximo número de campos que tenéis pensado utilizar en cualquier fichero es menor que 6, podéis disminuir la matriz de los campos y aumentar la de las fichas con lo cual aumentará el número máximo de fichas a introducir.

Descripción del programa

El menú nos muestra las 8 opciones principales de que disponemos, algunas de las cuales tienen a su vez submenús. El programa nos informa en todo momento de lo que podemos hacer y dispone de una ventana inferior en la cual explica el modo de introducir la opción elegida.

Opción crear fichero. Nos pide el número de campos y sus nombres. A continuación nos informa del número máximo de fichas que podemos introducir y nos pide confirmación. Si no estamos de acuerdo regresa al menú, de lo contrario comienza el proceso de introducir fichas del cual podemos salir en cualquier momento ya que cada vez que introducimos una se nos muestra la memoria que nos queda y la posibilidad de volver al menú. Si ya hay un fichero en memoria nos avisa de ello pues será destruido si creamos otro por lo que es aconsejable elegir la opción SALVAR FICHERO una vez lo hayamos creado.

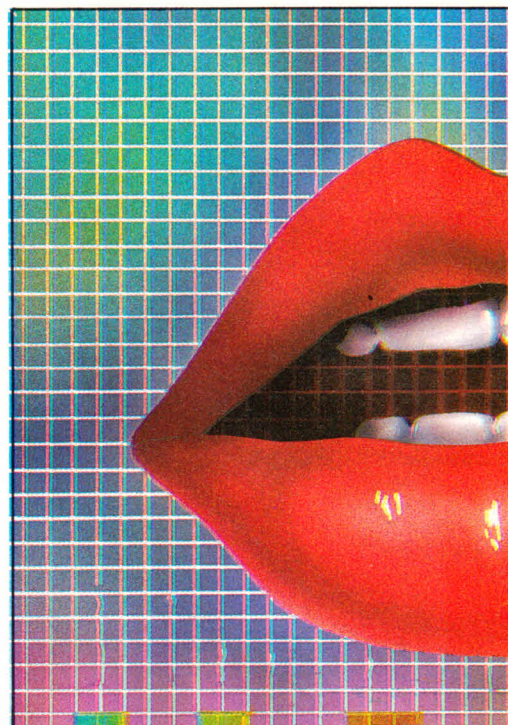
Salvar fichero. Salva el fichero que está en memoria.

Cargar fichero. Carga en memoria el fichero que le digamos. Si hay ya un fichero en memoria (bien porque lo hemos cargado previamente o bien porque lo hemos creado) nos avisa de ello ya que se destruirá. Si el fichero a cargar no está en el disco también nos informa.

Buscar dato. Nos pide el dato buscado y el campo a buscar. Lista todas las fichas que contengan ese dato.

Clasificación alfabética. Nos pide el campo por el que queremos clasificar y clasifica. Para ver las fichas clasificadas hay que elegir la opción apropiada en el submenú de LISTADO DE FICHAS.

Listado de fichas. Tiene a su vez un submenú, ya que se pueden listar las fichas desde la primera, desde el número que queramos u ordenadas (si previamente se ha hecho la ordenación).



Modificación. Nos visualiza las fichas una por una y nos pregunta si queremos modificar algún campo, o suprimir o añadir alguna ficha.

Salida de basic. Nos avisa si hay algún fichero en memoria.

VARIABLES

REGI\$	Fichas
INDICE	Clasificación
CAMPO	Nombre de los campos
OPC	Entrada de menús
SUPER	Número máximo de fichas
FIC	Número de fichas
AVOM	Aviso de fichero de memoria
NAME\$	Nombre de fichero a salvar o cargar
ENCTR	Contador de datos encontrados
ACME\$	Dato a buscar
CONTAUX	Indicador de rutina de modificación
FIZ	Contador de fichas visualizadas o a visualizar
X	Número de ficha a listar
B	Número de campo a clasificar
CLAS	Indicador de clasificación
NUMERO	Número de campos del fichero



NOTA: Los usuarios del CPC 464 deben sustituir la orden CLEAR INPUT de las líneas 280, 2580, 2590 y 1560 por: WHILE INKEY\$ < > "" : WEND

SUBROUTINAS

- 100-340** Menú principal
- 330-570** Crear ficheros
- 580-740** Rellena fichas
- 750-920** Rutina de salvar fichero
- 930-1130** Carga fichero
- 1140-1340** Rutina de búsqueda
- 1350-1450** Rutina de listado de ficha encontrada
- 1460-1620** Imprime menú de listado y entrada de opciones
- 1630-1790** Lista fichas desde la primera
- 1800-1860** Salta fichas
- 1870-2060** Rutina de clasificación
- 2070-2270** Lista fichas clasificadas
- 2280-2410** Imprime menú de actualización y entrada de opciones
- 2420-2530** Imprime y da entrada a rutina de modificación individual.
- 2540-2650** Modifica la ficha
- 2660-2730** Rutina de borrado de fichas
- 2740-2790** Subrutina de espera
- 2800-2890** Tratamiento de errores
- 2900** Salida a basic

BB

```

10 REM
20 ? PROGRAMA CREADO POR
30 ?
40 ? LUIS JAVIER GOMEZ GONZALEZ
50 ?
60 ON BREAK CONT
70 ON ERROR GOTO 2800
80 DIM REGI$(500,6) :DIM INDICE(500)
: DIM CAMPO$(6)
90 BORDER 13
100 ?
110 ? MENU PRINCIPAL
120 ?
130 MODE 2:CLS
140 INK 0,26:PAPER 2:WINDOW 29,55,5
,22
150 PRINT "      MENU"
160 PRINT:PRINT "1  CREAM FICHERO"
170 PRINT:PRINT "2  SALVAR FICHERO
"
180 PRINT:PRINT "3  LISTAR FICHAS "
190 PRINT:PRINT "4  ORDENACION ALFA
BETICA"
200 PRINT:PRINT "5  BUSQUEDA DE UN
DATO"
210 PRINT:PRINT "6  CARGAR FICHERO"
220 PRINT:PRINT "7  ACTUALIZAR"
230 PRINT:PRINT "8  SALIDA A BASIC"
240 WINDOW #2,1,80,24,25:PAPER #2,
13:INK 1,0:PEN #2,0
250 PRINT #2,"      SELECCIONE O
PCION CON LAS TECLAS DEL CURSOR Y P
ULSE [RETURN]      "
260 N=2
270 CURSOR 1:LOCATE 29,N
280 IF INKEY(0)<>-1 AND N>2 THEN N=
N-2:FOR B=1 TO 200:NEXT B
290 IF INKEY(2)<>-1 AND N<16 THEN N
=N+2:FOR B=1 TO 200:NEXT B
300 IF INKEY(18)<>-1 THEN OPC=(VPOS
(#0)-1)/2:CLEAR INPUT:GOTO 320
310 GOTO 270
320 CURSOR 0:CLS:WINDOW 1,80,1,23:CL
S #2
330 ON OPC GOSUB 380,780,1460,1870
,1140,960,2280,2930
340 GOTO 130
350 ?
360 ? CREA FICHERO
370 ?
380 IF AVOM>=1 THEN LOCATE 1,12:INP
UT"HAY UN FICHERO EN MEMORIA.CONTIN
UO?(INTRODUZCA OPCION) [N] NO,OTRA
TECLA SI":OPC$
390 IF AVOM>=1 AND OPC$="N" OR OPC$
="n" THEN RETURN
400 ERASE CAMPO$,INDICE,REGI$:DIM C
AMPO$(6):DIM REGI$(600,6):DIM INDIC
E(600)
410 LOCATE #2,22,1: PRINT #2,"EL NU
MERO MAXIMO DE CAMPOS ES 6"
420 CLS:LOCATE 15,3:INPUT " --POR FA
VOR INTRODUCZA NUMERO DE CAMPOS":NU
MERO
430 IF NUMERO>6 THEN CLS:CLS #2:GOT
O 420
440 CLS #2: SUPER=INT(13000/(17*NUME
RO))
450 PRINT #2,"EL NUMERO MAXIMO DE F
ICHAS SUPONIENDO MEDIA DE 15 CARACT
.POR CAMPO ES ":SUPER
460 FOR N=1 TO NUMERO
470 PRINT:PRINT TAB(19);" NOMBRE DE
L CAMPO NUMERO":N;
480 INPUT COMPO$(N):CAMPO$(N)=UPPER
$(COMPO$(N))
490 NEXT N
500 CLS
510 PRINT "      VA A CONFECCIONAR
USTED UN FICHERO CON":NUMERO;"CAMP
OS QUE SE LLAMARAN : "
520 FOR N=1 TO NUMERO
530 LOCATE 30,1+(N*2):PRINT N;"....
..":CAMPO$(N)
540 NEXT N
550 CLS #2:PRINT #2," SI NO ESTA D
E ACUERDO PULSE /N/... ..CUALQUIE
R OTRA TECLA PARA CONTINUAR      "
560 IF INKEY$="" THEN 560
570 IF INKEY(46)<>-1 THEN AVOM=AVO
M-1:RETURN
580 ?
590 ? RELLENA FICHA
600 ?
610 FIC=1
620 CLS #2:CLS:LOCATE 25,1:PRINT "R
ELLENANDO FICHA NUMERO ";FIC
630 FOR N=1 TO NUMERO
640 PRINT
650 PRINT "      ";CAMPO$(N);" : ";
660 HOR=POS(#0):VER=VPOS(#0):INPUT
ROGI$(FIC,N):REGI$(FIC,N)=UPPER$(RO
GI$(FIC,N))
670 IF REGI$(FIC,N)="" THEN CAMBIO$
=REGI$(FIC-1,N):REGI$(FIC,N)=CAMBIO
$:LOCATE HOR,VER:PRINT REGI$(FIC,N)
680 CLS #2:PRINT #2,"MEMORIA LIBRE
EN BYTES=";FRE("")-4000;" [RETURN]
IMPRIE EL DATO DE LA FICHA ANTERIO
R"
690 NEXT N
700 CLS #2:PRINT #2," PULSE /N/ P
ARA MENU Y CUALQUIER OTRA TECLA PAR
A RELLENAR UNA NUEVA FICHA      "
710 IF INKEY$="" THEN 710
720 IF INKEY(46)<>-1 THEN AVOM=AVO
M+1:RETURN
730 FIC=FIC+1
740 GOTO 620
750 ?
760 ? SALVA FICHERO
770 ?
780 CLS #2:CLS:LOCATE 22,11:INPUT"
NOMBRE DEL FICHERO A SALVAR":NAME$
790 IF LEN(NAME$)>8 THEN CLS:LOCATE
29,11:PRINT"NOMBRE DEMASIADO LARGO
":GOSUB 2770:RETURN
800 OPENOUT NAME$
810 PRINT #9,FIC
820 PRINT #9,NUMERO
830 FOR N=1 TO NUMERO
840 PRINT #9,CAMPO$(N)
850 NEXT N
860 FOR N=1 TO FIC
870 FOR Z=1 TO NUMERO
880 PRINT #9,REGI$(N,Z)
890 NEXT Z
900 NEXT N
910 CLOSEOUT
920 RETURN
930 ?
940 ? CARGA FICHERO
950 ?
960 IF AVOM>=1 THEN LOCATE 1,12:INP
UT "HAY UN FICHERO EN MEMORIA.CONTI
NUO?(IN] NO,[OTRA TECLA] SI ,INTRODU
ZCA OPCION":OPC$:IF OPC$="N" OR OPC
$="n" THEN RETURN
970 CLS:CAT
980 LOCATE 22,22:INPUT"NOMBRE DEL F
ICHERO A CARGAR ";W$
990 ERASE INDICE,REGI$,CAMPO$:DIM I
NDICE(600):DIM REGI$(600,6):DIM CAM
PO$(6)
1000 OPENIN W$
1010 INPUT #9,FIC,NUMERO
1020 FOR N=1 TO NUMERO
1030 INPUT #9,CAMPO$(N)
1040 NEXT N
1050 FOR R=1 TO FIC
1060 FOR C=1 TO NUMERO
1070 INPUT #9,REGI$(R,C)
1080 NEXT C:NEXT R

```

Serie
ORO

```

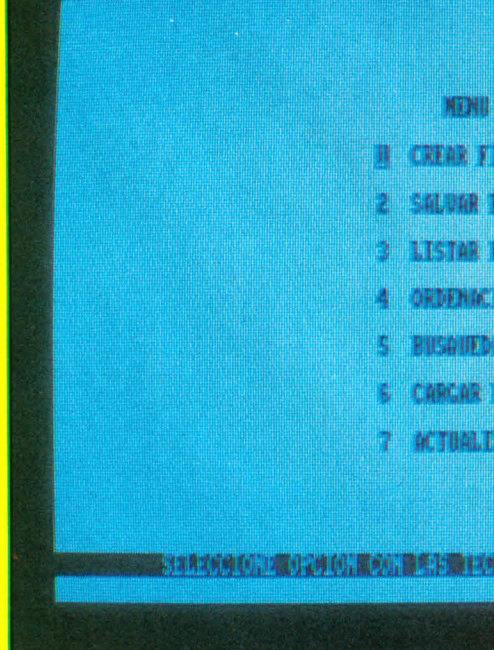
1090 CLOSEIN
1100 CLS:LOCATE 30,12:PRINT"*** CARG
A TERMINADA ***"
1110 GOSUB 2770
1120 AVOM=AVOM+1
1130 RETURN
1140 ?
1150 ? BUSCA DATOS
1160 ?
1170 CLS
1180 LOCATE 11,2:PRINT"POR FAVOR EN
TRE EL NUMERO DE CAMPO A BUSCAR : "
1190 PRINT
1200 FOR C=1 TO NUMERO
1210 PRINT TAB(30) C;".....";CAMPO
$(C)
1220 PRINT
1230 NEXT C
1240 INPUT #2,"CAMPO";B:CLS #2:PRIN
T #2,CAMPO$(B);
1250 CLS
1260 LOCATE 30,12:PRINT"INTRODUZCA
DATO A BUSCAR"
1270 INPUT #2,OCME$:ACME%=UPPER$(OC
ME$)
1280 ENCTR=0
1290 FOR N=1 TO FIC
1300 IF REGI$(N,B)=ACME$ THEN ENCTR
=ENCTR+1:GOSUB 1350
1310 NEXT N
1320 IF ENCTR>0 THEN CLS:LOCATE 22,
12:PRINT ACME$;" NO FIGURA MAS VECE
S":GOSUB 2770
1330 IF ENCTR=0 THEN CLS:LOCATE 22,
12:PRINT ACME$;" NO FIGURA EN EL AR
CHIVO":GOSUB 2770
1340 RETURN
1350 ?
1360 ? LISTA FICHA BUSCADA
1370 ?
1380 CLS
1390 LOCATE 1,30:PRINT "FICHA NUMER
O:";N
1400 FOR Z=1 TO NUMERO
1410 PRINT:PRINT " ";CAMPO$(Z);"
:";
1420 PRINT REGI$(N,Z)
1430 NEXT Z
1440 GOSUB 2770
1450 RETURN
1460 ?
1470 ? IMPRIME MENU DE LISTADO
1480 ?
1490 CLS:LOCATE 24,2:PRINT"RUTINA D
E LISTADO DE FICHAS"
1500 CLS #2:LOCATE #2,11,24:PRINT #
2,"COLOQUE EL CURSOR SOBRE LA OPCIO
N ELEGIDA Y PULSE [RETURN]"
1510 LOCATE 27,10:PRINT"1. LISTA TO
DAS LAS FICHAS"
1520 LOCATE 27,12:PRINT"2. LISTA SO
LO FICHA PEDIDA"
1530 LOCATE 27,14:PRINT"3. LISTA FI
CHAS CLASIFICADAS"
1540 LOCATE 27,16:PRINT"4. MENU PRI
NCIPAL"
1550 CURSOR 1:N=10
1560 LOCATE 27,N
1570 IF INKEY(0)<>-1 AND N>10 THEN
N=N-2:FOR B=1 TO 150:NEXT
1580 IF INKEY(2)<>-1 AND N<16 THEN
N=N+2:FOR B=1 TO 150:NEXT
1590 IF INKEY(18)<>-1 THEN OPC=(VPO
S(#0)-B)/2:CLEAR INPUT:GOTO 1610
1600 GOTO 1560
1610 IF OPC=4 THEN RETURN ELSE ON O
PC GOTO 1630,1800,2070
1620 GOTO 1490
1630 ?
1640 ? LISTA FICHAS DESDE EL PRINCI
PIO
1650 ?
1660 CURSOR 0:FIZ=1:CONTAUX=0
1670 IF FIZ>FIC THEN CLS:LOCATE 15,
12:PRINT "*** FINAL DE FICHERO. NO H

```

```

AY MAS REGISTROS ***:GOSUB 2770:RET
URN
1680 CLS #2:CLS:LOCATE 35,1:PRINT"F
ICHA :";FIZ;
1690 FOR N=1 TO NUMERO
1700 PRINT:PRINT CAMPO$(N);" : ";
1710 PRINT REGI$(FIZ,N)
1720 NEXT N
1730 FIZ=FIZ+1
1740 IF CONTAUX>0 THEN GOTO 2480
1750 PRINT #2," /S/ LISTA FICHA SI
GUIENTE, /A/ LISTA FICHA ANTERIOR,
/M/ MENU"
1760 IF INKEY(49)<>-1 THEN FIZ=FIZ-
2:GOTO 1670
1770 IF INKEY(38)<>-1 THEN 1460
1780 IF INKEY(60)<>-1 THEN 1670
1790 GOTO 1760
1800 ?
1810 ? SALTA FICHAS
1820 ?
1830 CLS #2:CLS:LOCATE 23,12:INPUT"
INTRODUZCA EL NUMERO DE LA FICHA ";
X
1840 FIZ=X
1850 IF FIZ>=FIC THEN CLS:LOCATE 20
,12:PRINT"NUMERO EQUIVOCADO. NO HAY
TANTAS FICHAS":FIZ=1:GOSUB 2770:RE
TURN
1860 GOTO 1670
1870 ?
1880 ? RUTINA DE CLASIFICACION
1890 ?
1900 CLAS=1:CLS
1910 LOCATE 10,1:PRINT "CADA REGIST
RO DE SU FICHERO CONTIENE";NUMERO;"
CAMPOS"
1920 FOR N=1 TO NUMERO
1930 PRINT:PRINT TAB(22);N;".....
.....";CAMPO$(N)
1940 NEXT N
1950 INPUT #2,"
NUMERO DE CAMPO A CLASIFICAR";P
1960 CLS:LOCATE 12,12:PRINT"ESPERE
UN INSTANTE POR FAVOR,ESTOY CLASIFI
CANDO"
1970 FOR I=1 TO FIC
1980 INDICE(I)=I
1990 NEXT I

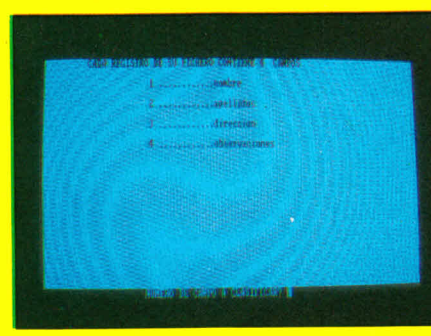
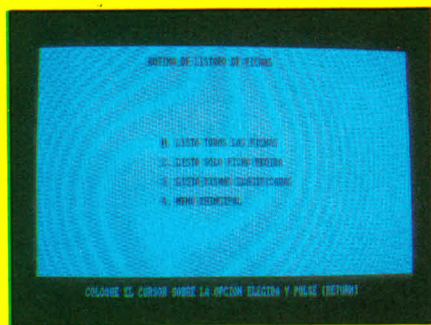
```



```

2000 FOR I=1 TO FIC-1
2010 FOR J=I+1 TO FIC
2020 IF REGI$(INDICE(I),B)>REGI$(IN
DICE(J),B) THEN P=INDICE(I):INDICE(
I)=INDICE(J):INDICE(J)=P
2030 NEXT J
2040 NEXT I
2050 CLS:LOCATE 25,12:PRINT "CLASIF
ICACION TERMINADA":GOSUB 2770
2060 RETURN
2070 ?
2080 ? LISTA FICHAS CLASIFICADAS
2090 ?
2100 CLS
2110 IF CLAS=0 THEN LOCATE 1,12:PRI
NT"PRIMERO TIENE QUE CLASIFICAR EL
FICHERO CON LA OPCION 'CLASIFICACIO
N ALFABETICA":CURSOR 0:GOSUB 2770:R
ETURN
2120 FOR I=1 TO FIC
2130 K=INDICE(I)
2140 PRINT TAB(30)"FICHA NUMERO";I
2150 FOR B=1 TO NUMERO
2160 PRINT:PRINT CAMPO$(B);" :";
2170 PRINT REGI$(K,B)
2180 NEXT B
2190 PRINT #2 ,"SI NO DESEA VER MAS
FICHAS PULSE /N/ DE LO CONTRARIO P
ULSE CUALQUIER OTRA TECLA"
2200 A$=INKEY$
2210 IF A$="" THEN 2200
2220 IF A$="N" OR A$="n" THEN RETUR
N
2230 CLS
2240 CLS #2
2250 NEXT I
2260 LOCATE 22,12:PRINT"FINAL DE FI
CHERO. NO HAY MAS REGISTROS":GOSUB 2
770
2270 RETURN
2280 ?
2290 ? IMPRIME MENU DE ACTUALIZACIO
N
2300 ?
2310 LOCATE 34,5:PRINT"SUBMENU"
2320 LOCATE 28,10:PRINT"1 A:ADIR MA
S FICHAS"
2330 LOCATE 28,12:PRINT"2 MODIFICAR
"
2340 LOCATE 28,14:PRINT"3 MENU PRIN

```



```

ICHERO
ICHERO
ICHAS
ON ALFABETICA
DE UN DATO
ICHERO
AR
ES DE CUANDO PULSE RETURN

```

```

2680 *
2690 FOR K=FIZ TO FIC:FOR N=1 TO NU
MERO
2700 REGI$(K-1,N)=REGI$(K,N)
2710 NEXT N:NEXT K
2720 FIZ=FIZ-1:FIC=FIC-1
2730 GOTO 1670
2740 *
2750 * SUBROUTINA DE ESPERA
2760 *
2770 CLS #2:LOCATE #2,23,25:PRINT #
2,"PULSE CUALQUIER TECLA PARA CONTI
NUAR"
2780 IF INKEY$="" THEN 2780
2790 RETURN
2800 *
2810 * TRATAMIENTO DE ERRORES
2820 *
2830 CLS
2840 IF DERR=146 THEN LOCATE 25,12:
PRINT " FICHERO NO EXISTENTE"
2850 IF DERR=148 THEN LOCATE 30,12:
PRINT"EL DISCO ESTA LLEND"
2860 RESUME 2870
2870 CLS #2:LOCATE #2,19,25:PRINT #
2,"PULSE CUALQUIER TECLA PARA VOLVE
R AL MENU"
2880 IF INKEY$="" THEN 2880
2890 GOTO 130
2900 *
2910 * SALIDA A BASIC
2920 *

```

Serie ORO

```

2930 MODE 2:IF AVOM>=1 THEN LOCATE
5,12:INPUT "EXISTE UN FICHERO EN ME
MORIA. ESTA SEGURO DE QUERER SALIR?
(S/N) ";OPC$:OPC%=UPPER$(OPC$)
2940 IF OPC$="S" OR OPC$="SI" THEN
CLS:NEW
2950 IF AVOM=0 THEN CLS:NEW
2960 RETURN

```



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, solicítanoslo.

```

CIPAL"
2350 PRINT #2,"INTRODUZCA OPCION";:
INPUT #2,OPC
2360 CLS
2370 IF OPC=3 THEN RETURN ELSE ON O
PC GOTO 2380,2420
2380 *
2390 * SUMA 1 AL CONTADOR DE FICHAS
Y MANDA A RUTINA DE RELLENO
2400 *
2410 FIC=FIC+1:GOTO 620
2420 *
2430 * SUBROUTINA DE MODIFICACION
2440 *
2450 CONTAUX=1
2460 FIZ=1
2470 GOTO 1670
2480 CLS #2:PRINT #2,"Q MODIFICAR//
W SUPRIMIR//ENTER NO ALTERAR//M ME
NU// [INTRODUZCA OPCION]";:INPUT #2
,OPC$:opc%=UPPER$(opc$)
2490 IF OPC$="Q" THEN 2540
2500 IF OPC$="W" THEN 2660
2510 IF OPC$="M" THEN RETURN
2520 IF OPC$="" THEN 1670:REM VA A
LA RUTINA DE LISTAR FICHAS
2530 GOTO 2480
2540 *
2550 * MODIFICANDO FICHA
2560 *
2570 CLS #2:CLS
2580 FOR N=1 TO NUMERO
2590 PRINT CAMPO$(N);": ";REGI$(FIZ
-1,N)
2600 CLS #2:PRINT #2,"FICHA/";FIZ-1
;"/M/MODIFICA CAMPO---- [RETURN] DE
JA EL CAMPO COMO ESTA"
2610 IF INKEY(18)<>-1 THEN CLEAR IN
PUT:GOTO 2640
2620 IF INKEY(38)<>-1 THEN CLEAR IN
PUT:CLS #2:LOCATE #2,24,26:PRINT #2
,"ENTRE NUEVOS DATOS":PRINT CAMPO$(
N);:INPUT CAMBIO$:cambio%=UPPER$(ca
mbio%):REGI$(FIZ-1,N)=CAMBIO$:GOTO
2640
2630 GOTO 2610
2640 NEXT N
2650 GOTO 1670
2660 *
2670 * BORRA FICHA IGUALandola A LA
SIGUIENTE

```

GANA 100.000 PESETAS CON MICROHOBBY AMSTRAD SEMANAL

Porque 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.

— No se admitirán programas que contengan caracteres de control, debido a que no son correctamente interpretados por las impresoras.

— 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.

¡ENVIANOS TU PROGRAMA!

Adjuntando los siguientes datos:

Nombre y apellidos, dirección y teléfono.

Indicando claramente en el sobre:

AMSTRAD SEMANAL

a HOBBY PRESS, S. A. La Granja, 39. Pol. Ind. Alcobendas (Madrid)

IMPRESIONA CON TU PANTALLA

Continuando con la serie que iniciamos la semana pasada, intentaremos pasar otro programa que hemos preparado en Basic, a código máquina



El programa de hoy trata también de la impresión de caracteres en pantalla, pero esta vez lo enfocaremos desde una perspectiva distinta.

Trataremos de estudiar la forma en que se pueden realizar distintos marcos o presentaciones de pantallas.

Para ello hemos confeccionado un programa en Basic que prepara una pantalla de presentación, en el cual vamos a colocar un marcó que la rodee.

Siempre que se desee preparar una pantalla para un programa, ya sea de juegos o utilidades, en lo primero que debemos pensar, es qué colores serán los más adecuados para dicho propósito. Así pues, debemos decidir cuáles serán dichos colores.

Debemos tener presente también, el modo de pantalla en el cual deseamos trabajar, ya que para cada modo distinto deberemos elegir un número distinto de tintas. Así pues, si trabajamos en modo 2 únicamente deberemos elegir dos de dichas tintas, si lo hacemos en modo 1 se deberán elegir 4 y si estamos en modo 0 deberemos elegir 16.

Dado que en nuestro caso trabajaremos en modo 1, únicamente deberemos elegir 3 colores, y dado que únicamente imprimiremos en pantalla con las plumas 0, 1 y 3, podemos dejar de definir la pluma número 2, ya que en este caso no nos interesa.

Lo primero que debemos definir en nuestro programa en código máquina es definir una tinta para cada una de las plumas que vayamos a utilizar.

Para ello utilizamos una rutina que el firmware nos proporciona, se trata de la siguiente.

SELECCIONA TINTA. #BC32

Selecciona los dos colores que van a ser utilizados en dicha pluma. Si los dos colores son iguales, la tinta permanece inalterable. Si son distintos, dicha tinta aparece parpadeante en esos dos colores.

Condiciones de entrada.

El registro A debe contener el número de pluma. El registro B, debe contener la primera tinta y el registro C debe contener el segundo número de tinta. Así pues, si el contenido de B y C son iguales la tinta aparecerá inalterable, de lo contrario dicha tinta será parpadeante.

Condiciones de salida.

Los registros AF, BC, DE y HL se corrompen, todos los demás son preservados.

Una vez elegidos los colores con los que vamos a trabajar, deberemos seleccionar con qué papel de fondo deseamos que aparezca dicha pantalla, para ello utilizaremos otra vez una de las rutinas que el firmware nos proporciona.

SELECCIONA PAPEL. #BB96

Selecciona el papel de fondo con el cual queremos trabajar en la ventana elegida.

Condiciones de entrada:

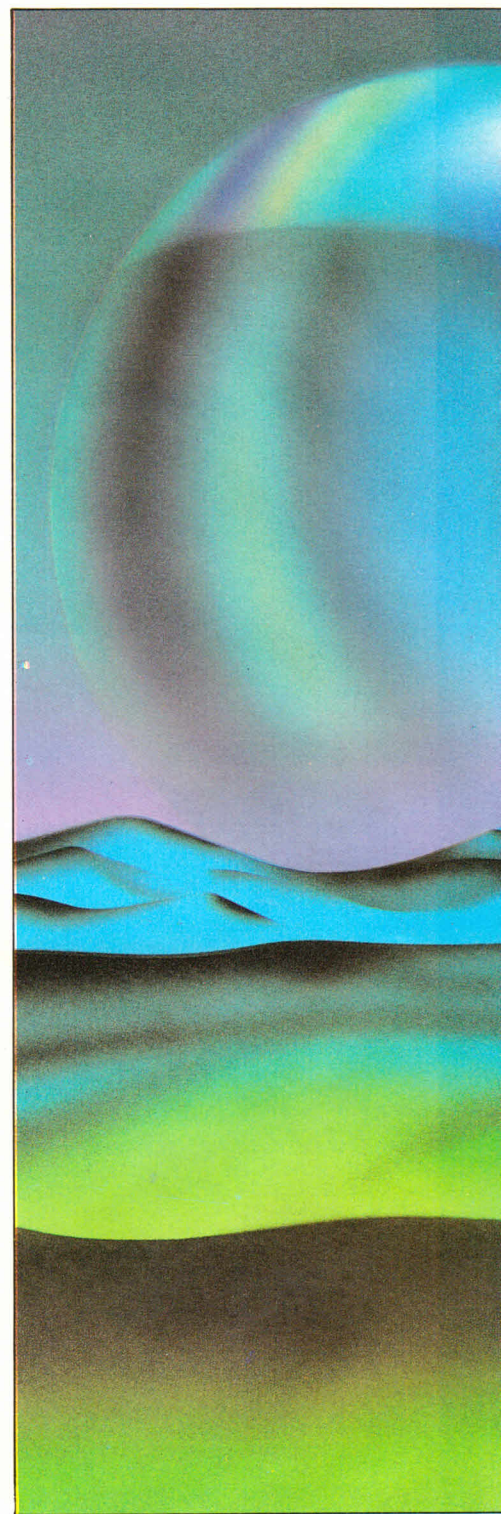
El registro A debe contener la tinta a utilizar.

Condiciones de salida:

Se corresponden los registros AF y HL. Los demás registros se preservan.

Borrado de ventanas

Una vez seleccionado el papel de fondo, deberemos limpiar la ventana en la que se está trabajando, para que dicho papel aparezca realmente como fondo de dicha pantalla.



Así pues, el siguiente paso será provocar un 'CLS' de pantalla. Para este propósito llamamos a la siguiente rutina del firmware:

BORRA LA VENTANA SELECCIONADA. #BB6C

Provoca un 'CLS' en la ventana seleccionada, colocando como fondo el papel elegido previamente.

Condiciones de entrada:

No existe ninguna condición.

Condiciones de salida:

Código MAQUINA

Condiciones de salida:

Se corresponden los registros AF, BC, DE y HL, y se preservan los restantes.

Una vez seleccionadas las condiciones con las que se va a imprimir el texto, ya podemos empezar a preparar las rutinas de impresión.

Rutinas de impresión

Dado que tendremos que imprimir dos tiras horizontales y dos verticales, hemos preparado sendas rutinas que provocarán dicha impresión, **'PINHOR'** que imprime las tiras horizontales y **'PINVER'** que se encargará de pintar las columnas.

Antes de llamar a dichas rutinas, deberemos cargar el registro HL con la posición inicial de impresión de pantalla y llamar a dichas rutinas que se encargarán de incrementar adecuadamente las posiciones de impresión.

Estudiaremos en primer lugar **'PINHOR'** que es la encargada de la impresión de las filas.

Para producirla, preparamos un bucle cuyo contador será el registro B. No hemos elegido el registro B por capricho, sino por la facilidad que nos proporciona, ya que la instrucción:

DJNZ

se encarga de decrementar dicho registro, compararlo con cero, y si es cero se para el bucle y en caso contrario se ejecuta de nuevo hasta que B sea cero.

Así pues, cuando se entra en esta rutina lo primero que hacemos es colocar el cursor en la posición de impresión.

PINHOR: CALL #BB75

A continuación cargamos el registro B (contador de bucle) con 40, ya que estos son los caracteres que entran en una fila en modo 1.

LD B,40

en este momento es donde se inicia el bucle. Ahora deberemos cargar en el acumulador el código ASCII del carácter que deseamos imprimir (en este caso será el 202).

BUCL: LD A,202



Se corresponden los registros AF, BC, DE y HL. Se preservan los restantes registros.

El paso siguiente, será indicar con qué plumas y papel deseamos imprimir el texto en pantalla, lo cual ya sabemos hacer puesto que se ha explicado anteriormente y en el capítulo anterior.

Otra de las cosas que se deben definir antes de pasar a la impresión de cualquier texto, será el color del **'BORDER'**, para ello se llamará a la

apropiada rutina del firmware.

SELECCIONA COLOR DEL BORDE.
#BC38

Selecciona los dos colores que se van a utilizar como **'BORDER'**. Si los dos colores son iguales, el borde aparecerá inalterable, de lo contrario éste aparecerá parpadeante.

Condiciones de entrada:

El registro B debe contener el primer color y el registro C debe contener el segundo color.

para que la rutina del firmware correspondiente ejecute la impresión:

CALL #BB5A

a continuación debemos comprobar si el bucle ha finalizado o no:

DJNZ BUC1

esta instrucción como hemos dicho anteriormente, se encarga de decrementar y comparar el registro B con cero, así pues, este bucle se deberá repetir 40 veces.

La rutina de impresión de columnas, utiliza también el registro B como contador de bucle. Pero en este caso, se deberá calcular la posición de impresión en cada pasada, ya que de lo contrario se imprimiría en forma horizontal.

PINVER: LD B,24
BUC2: PUSH HL

en este caso cargamos el registro B con 24, ya que estas son las posición a imprimir, y además preservamos el registro doble HL, ya que este registro se corrompe cuando se llama a la dirección del firmware encargada de posicionar el cursor.

CALL #BB75

a continuación cargamos el acumulador con el valor del carácter a imprimir y provocamos la impresión en pantalla.

LD A,202
CALL #BB5A

seguidamente recuperamos el contenido del registro HL e incrementamos la posición vertical de pantalla.

POP HL
INC L

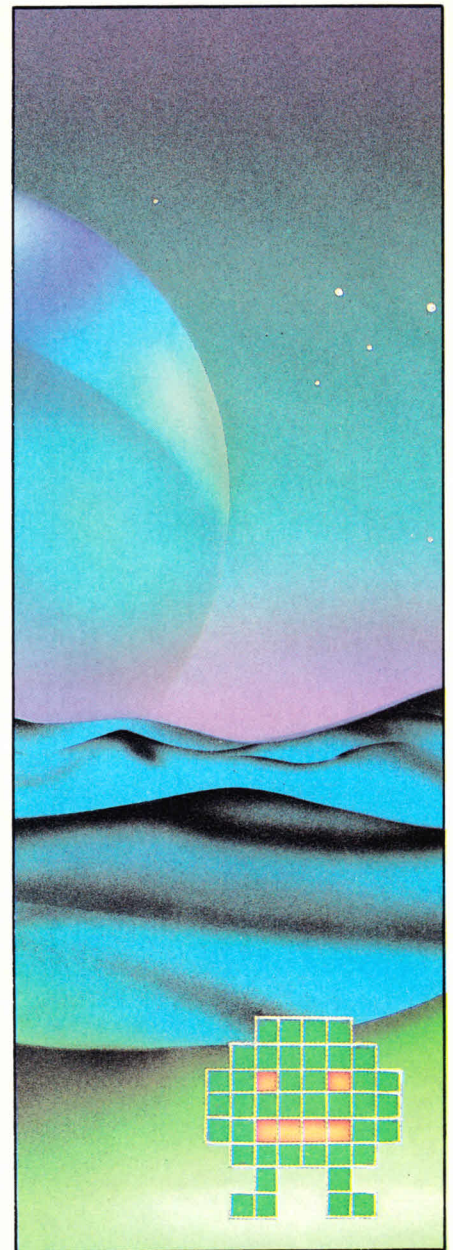
y por último comprobamos si el bucle ha finalizado, si es así, retornamos de la rutina y en caso contrario, éste se repite hasta que el contenido del registro B sea cero.

EUASO 2 BB

```

10      ORG #A000
20 ;
30 ;TNK 3,0
40 ;
50      LD A,3
60      LD BC,0
70      CALL #BC32
80 ;
90 ;TNK 1,24
100 ;
110     LD A,1
120     LD BC,#181B
130     CALL #BC32
140 ;
150 ;TNK 0,1
160 ;
170     XOR A
180     LD BC,#0101
190     CALL #BC32
200 ;
210 ;MODE 1
220 ;
230     LD A,1
240     CALL #BC0E
250 ;
260 ;PAPER 0
270 ;
280     XOR A
290     CALL #RR96
300 ;
310 ;CLS
320 ;
330     CALL #BR6C
340 ;
350 ;PAPER 3
360 ;
370     LD A,3
380     CALL #RR96
390 ;
400 ;PEN 1
410 ;
420     LD A,1
430     CALL #RR90
440 ;
450 ;BORDER 1
460 ;
470     LD BC,#0101
480     CALL #BC3B
490 ;
500 ;IMPRESION DE LA LINEA
510 ;SUPERIOR HORIZONTAL
520 ;
530     LD HL,#0101
540     CALL PINHOR
550 ;
560 ;IMPRESION DE LA LINEA
570 ;INFERIOR HORIZONTAL
580 ;
590     LD HL,#011B
600     CALL PINHOR
610 ;
620 ;IMPRESION DE LA COLUMNA
630 ;VERTICAL IZQUIERDA
640 ;
650     LD HL,#0101
660     CALL PINVER
670 ;

```



```

680 ;IMPRESION DE LA COLUMNA
690 ;VERTICAL DERECHA
700 ;
710     LD HL,#2801
720     CALL PINVER
730 ;
740 ;ESPERA A QUE SE PULSE UNA TECLA
750 ;
760     CALL #BB1B
770     RET
780 ;
790 ;RUTINA DE IMPRESION HORIZONTAL
800 ;
810 PINHOR: CALL #BB75
820         LD B,40
830 BUC1:  LD A,202
840         CALL #BR5A
850         DJNZ BUC1
860         RET
870 ;
880 ;RUTINA DE IMPRESION VERTICAL
890 ;
900 PINVER: LD B,24
910 BUC2:  PUSH HL
920         CALL #BB75
930         LD A,202
940         CALL #BR5A
950         POP HL
960         INC L
970         DJNZ BUC2
980         RET

```

DATCUR 2 BB

```

10 FOR N=&A000 TO &A06C
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<>&2DBD THEN PRINT "ERROR
EN DATAS"
60 DATA 62,3,1,0,0,205,50
70 DATA 188,62,1,1,24,24,205
80 DATA 50,188,175,1,1,1,205
90 DATA 50,188,62,1,205,14,188
100 DATA 175,205,150,187,205,108,187
110 DATA 62,3,205,150,187,62,1
120 DATA 205,144,187,1,1,1,205
130 DATA 56,188,33,1,1,205,79
140 DATA 160,33,24,1,205,79,160
150 DATA 33,1,1,205,92,160,33
160 DATA 1,40,205,92,160,205,24
170 DATA 187,201,205,117,187,6,40
180 DATA 62,202,205,90,187,16,249
190 DATA 201,6,24,229,205,117,187

```

```

200 DATA 62,202,205,90,187,225,44
210 DATA 16,243,201,0,0,0,0

```

```

10 TNK 3,0:TNK 1,24:TNK 0,1
20 MODE 1:PAPER 0:CLS
30 PAPER 3:BORDER 1
40 FOR n=1 TO 40
50 LOCATE n,1:PRINT CHR$(202)
60 LOCATE n,24:PRINT CHR$(202)
70 NEXT
80 FOR n=1 TO 24
90 LOCATE 1,n:PRINT CHR$(202)
100 LOCATE 40,n:PRINT CHR$(202)
110 NEXT
120 WHILE INKEY#="" :WEND

```

SOMOS MAYORISTAS

MICRO-1

PRECIOS INCLUIDO IVA

C/ Duque de Sesto, 50. 28009 Madrid
Tel.: (91) 274 53 80

(Metro O'Donnell o Goya)
Aparcamiento gratuito en Felipe II

Ofertas en software: 2 programas al precio de 1 y además regalo fin de curso una calculadora completamente gratis. ¡¡Asombroso!!
¿Verdad?

BAT MAN _____	2.300	ptas.	KUNG-FU MASTER _____	2.300	ptas.
ROCK'N LUCHA _____	2.300	ptas.	SABOTEUR _____	2.300	ptas.
YIER AR KUNG FU _____	2.300	ptas.	PING PONG _____	2.300	ptas.
THE WAY OF THE TIGER _____	2.300	ptas.	MILLION II _____	0000	ptas.
WEST BANK _____	2.100	ptas.	OLE TORO _____	2.300	ptas.
CAMELOT WARRIORS _____	2.300	ptas.	TURBO ESPRIT _____	2.100	ptas.
RAMBO _____	2.300	ptas.	SABRE WULF _____	1.650	ptas.
WORLD CUP (DISCO) _____	3.300	ptas.	BATALLA DE LOS PLANETAS _____	2.100	ptas.
MILLION II (DISCO) _____	3.300	ptas.	SABOTEUR-COMBAT LINX DISCO _____	3.300	ptas.
RAMBO-MATCH DAY (DISCO) 3.300 ptas.					

SOFTWARE DE REGALO (OFERTA 2x1)
DECATHLON BEACH HEAD SOUTHERN BELLE
DRAGONTORC

LAPIZ OPTICO
3.295 PTAS.

CASSETTE ESPECIAL ORDENADOR
5.295 PTAS.

SINTETIZADOR DE VOZ EN
CASTELLANO
7.650 PTAS.

AMPLIACION DE MEMORIA ANTA 64 K.3
12.500 PTAS.

IMPRESORAS
20% DE DESCUENTO SOBRE P.V.P.

TAPA DE METACRILATO 464	895
CABLE CENTRONICS	3.175
CABLE SEGUNDA UNIDAD D.	1.790
CABLE SEPARADORES 6128	1.975
INTERFACE RS232	9.265
CINTA VIRGEN C15	69

CABLE AUDIO	795
CABLE ADAPTADOR 2 JOYSTICK	2.390
CABLE SEPARADORES 464	1.390
CABLE SEPARADOR 8256	2.900
CABLE RS232	2.500
DISKETTES 3"	990

PRECIOS EXCEPCIONALES PARA TU AMSTRAD
CPC-464, CPC-6128, PCW-8256

¡¡LLEGARON LAS REBAJAS DE VERANO A MICRO 1!!

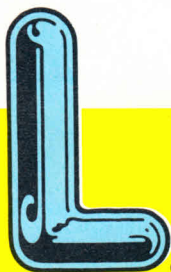
OFERTAS EN JOYSTICKS	
QUICK SHOT I	1.395
QUICK SHOT II	1.695
QUICK SHOT V	1.695

PEDIDOS CONTRA REEMBOLSO SIN NINGUN GASTO DE ENVIO TEL. (91) 274 53 80
O ESCRIBIENDO A: MICRO-1. C/ DUQUE DE SESTO, 50. 28009 MADRID.

Tiendas y distribuidores grandes descuentos.
Dirigirse a Diproimsa. C/ Galatea, 25. Tel. (91) 742 20 19 ó 742 79 68

ALIEN HIGHWAY

Los afortunados que hayan vivido la apasionante aventura del Highway Encounter, tienen ahora la oportunidad de seguir en la brecha con Alien Highway Encounter 2.

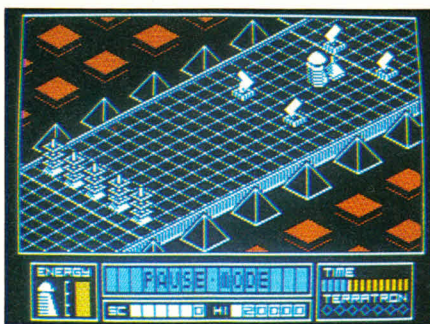


La autopista de la muerte se extiende ante nosotros, rodeada por un tétrico paisaje de formas geométricas. Sobre su superficie de losas regulares, un héroe solitario se enfrenta a los peligros de la larga travesía.

Programado para matar, nada le detendrá hasta que sea completamente destruido; su meta: alcanzar el final de la peligrosa vía.

Todo comenzó con el irresistible aumento del poderío del Imperio Alien, los cuales, en su última factoría de robots de guerra, están consiguiendo unas elevadas cifras de producción.

Estimadas las cifras obtenidas por los servicios de espionaje, se ha llegado a la conclusión que de continuar la fabricación de máquinas de guerra autónomas a ese ritmo, en un



tiempo de 20 días la proporción de robots del Imperio doblaría la fuerza defensiva de Vortones de nuestro territorio.

Después del golpe de mano asestado por los Vortones en Highway Encounter, que tuvo como consecuencia la total destrucción de la base de invasión Alien, no se comprende cómo éstos han conseguido rehacerse con tanta rapidez.

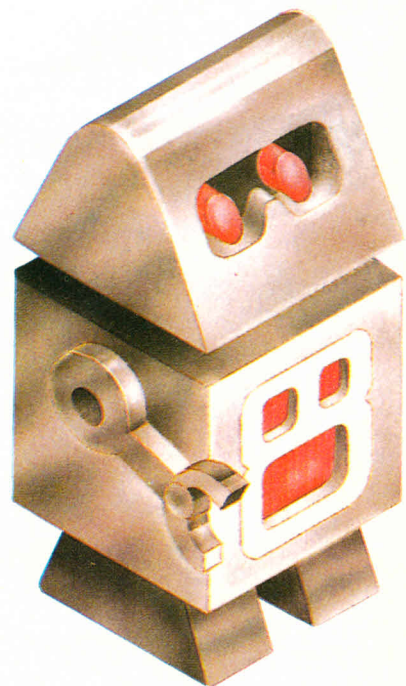
En aquella vez, la misión se llevó a cabo por una columna de cinco robots, que arrastrando la terabomba consiguieron depositarla en el corazón de la base alienígena.

Ahora la situación ha cambiado radicalmente. Con el desarrollo del nuevo modelo de la generación de **Vortones** de guerra, el éxito de la misión se encomienda a un solitario modelo experimental que deberá colocar la bomba en posición de fuego. Mejoradas sus características de disparo y aumentados considerablemente sus blindajes, sus posibilidades de éxito se basan en recargar de energía la terabomba, en las distintas estaciones situadas a lo largo de la autopista.

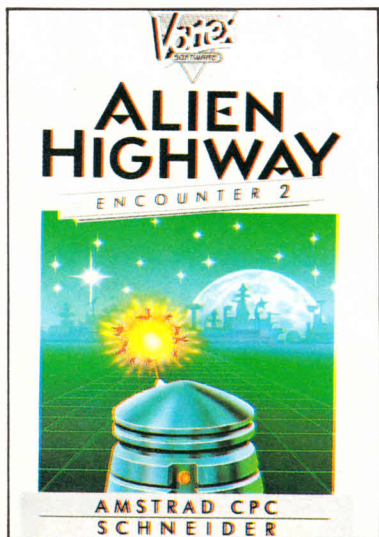
La entrada al complejo industrial no será fácil, y solamente un gran robot de combate podrá conseguir resultados positivos. Acostumbrados a los magníficos juegos de la casa **Vortex**, nos preguntamos qué oscuros motivos habrán llevado a sus creadores de software a realizar una segunda versión del Highway Encounter.

La sequía de ideas puede ser uno de los motivos; otro puede ser el aprovechar el éxito obtenido por **Highway Encounter**, e intentar sacarle nuevos frutos en una segunda versión, como si de una película de **Rocky** se tratara. Esperemos que por lo menos en este caso no se llegue a la cuarta versión, que ya sería excesivo.

Sequía de ideas aparte y demás, Alien Highway es un producto de innegable calidad. En la nueva aventura, el decorado ha cambiado radicalmente, bordeando la autopista



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





un paisaje geométrico que nada tiene que ver con el marco ecologista de la aventura anterior.

El planteamiento de la aventura sigue siendo el mismo, con la diferencia de que ahora, en vez de disponer de cinco robots para completar el recorrido, tenemos un único Vorton con cinco vidas, gracias a su escudo protector.

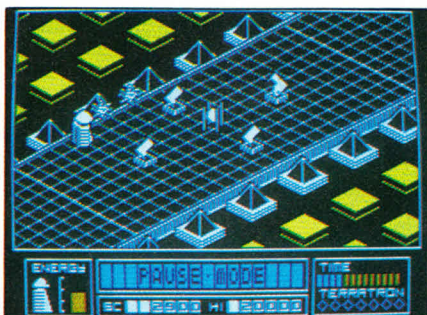
Las trampas que nos esperan en ruta son distintas de las de la versión anterior. Nuevos medios mecánicos y esferas de energía nos aguardan impacientes; en esta versión hemos de ser especialmente cuidadosos con los bordes de la vía, pues el contacto con los mismos es mortal.

También habremos de valernos esta vez de los obstáculos móviles para superar determinadas fases, interponiéndolos entre nosotros y una muerte segura.

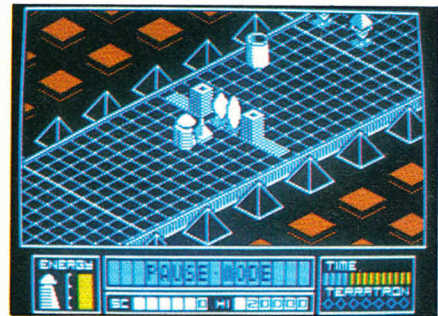
Las zonas de reabastecimiento son imprescindibles para nuestro Vorton,

en ellas se recarga la energía de combate y el escudo protector recupera su compacta estructura molecular, dañada por los anteriores encuentros.

En cuanto al recorrido por la autopista, hemos de mencionar que en Alien Highway, las pantallas no se suceden en un orden concreto como ocurría en la aventura anterior, ahora aparecen al azar ante nuestros ojos, lo cual condiciona en parte nuestra estrategia de lucha.



Mister JOYSTICK



Los gráficos, como cualquier obra de los creadores de **Vortex**, son de una calidad y precisión fuera de lo común, constituyendo el plato fuerte del programa.

Los nuevos alienígenas que nos atacan y las trampas, y demás elementos móviles, tiene un gran atractivo y su **animación** y efectismo son **asombrosos**.

Una pena que tanta maravilla no se haya aplicado a un programa completamente nuevo, casi derrochándola en una simple versión.

Los conocedores de **Highway Encounter** ya saben lo que van a encontrar ante sus pantallas, los que no hayan podido disfrutar de la **maravilla** anterior de **Vortex**, tienen ahora una estupenda posibilidad de comprobar por qué **Vortex** es una de las casas punteras en el mundo de los juegos de ordenador.



Sin duda alguna

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

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

SIN DUDA ALGUNA está abierta a todos.

ACERCA DEL PROGRAMA «HAGGIE»

Tengo un AMSTRAD CPC-464, y les agradecería que me contestaran a las siguientes preguntas:

1. Soy comprador habitual de M.H. **Amstrad**, y en el n.º 29, en la Serie Oro, traen un juego: Haggie, por lo que se ve, debe de ser un buen juego, pero el problema mío es que no me sale. ¿Qué pasa? ¿Está mal editado? Lo he repasado tres veces y no me sale.

2. Cuando haces un programa, lo

deseas grabar y además no quieres que te encuentren el listado, ¿se debe teclear..., save «título del programa», P... ?

Francisco Granero

1) *Que nosotros sepamos, el programa «Haggie», de AMSTRAD Semanal número 29, funciona perfectamente en cualquier Amstrad de la serie CPC. No obstante, como nos comenta que tiene un 464, podría tener problemas si posee una unidad de disco. La razón es la siguiente: cuando se le «endiña» al 464 un programa muy largo, como «Haggie», la ROM del disco comienza a comportarse de manera muy extraña, o al menos las nuestras lo hacen: tienen la desagradable costumbre de comerse, literalmente hablando, un trozo del programa, así, como sueña; de pronto, un grupo de líneas de las últimas, críticas para la buena marcha de la aplicación, desaparecen en las voraces fauces de la ROM del disco.*

Así, al ejecutar el programa, aparecen todo tipo de mensajes de error extraños, fundamentalmente referentes a errores de sintaxis en líneas de programa que no existen.

Sospechamos que esto está mal de «fábrica» en la ROM del disco, pero no podemos asegurar que suceda en todas las unidades. Si después de esta prolija explicación resulta que usted no tiene disco, bien, nuestra más sincera felicitación por haber continuado la lectura hasta aquí; llegados a este punto, por desgracia lo único que puede suceder es que haya usted tecleado mal el programa.

2) *Su segunda pregunta, por fortuna, tiene una respuesta mucho menos evanescente: sí. Para grabar un programa Basic en modo protegido tiene usted que hacerlo como indica, es decir: save "nombre", P*

EJECUTAR BINARIO

Estimados amigos de MICRO-HOBBY **AMSTRAD**, un programa en código máquina se puede hacer funcionar bajando la ramtop al nivel adecuado con el comando memory, cargando el programa con el comando LOAD y ejecutándolo con el comando CALL. ¿Cómo se puede hacer directamente, es decir con RUN "nombre programa"? Yo lo he intentado así y me da Memory full. Lo que les pido, por tanto, es la forma de grabar el código máquina con autoejecución.

¿Cómo se puede quitar la cabecera a los programas?

¿Hay algún error en el programa HAGGIE publicado en su revista n.º 29? Lo he tecleado y comprobado entero, y cuando el tiempo (TIME) llega a 1975 el programa se para poniendo, exactamente: Error... 2 at line y no pone el n.º de línea.

Gracias por su atención.

José A. de la Fuente
Basauri (Vizcaya)

Tu problema es muy sencillo de resolver, y la solución se encuentra utilizando los parámetros de la instrucción SAVE a tal efecto. Esta orden tiene la siguiente estructura:

SAVE "nombre del programa", b, INI, LON, EJEC

La b, le indica que lo que va a grabar es una tira de binario, el primer parámetro INI, es a partir de dónde quieres salvar, el segundo la longitud de la tira que quieres almacenar, y el tercero, clave del asunto para cuando vayas a hacer RUN, a partir de qué dirección ha de comenzar a ejecutarse. De esta forma tan sencilla podrás ya hacer:

RUN "nombre del programa"

Almacenes Méndez



CAPITAN CORTES, 17. TELEF.-228607. ORENSE
Servimos a tiendas y almacenes

DISTRIBUIDOR INDESCOMP PARA ORENSE:



Precios especiales para el
PCW 8256

**LO TENEMOS TODO SOBRE
ORDENADORES**

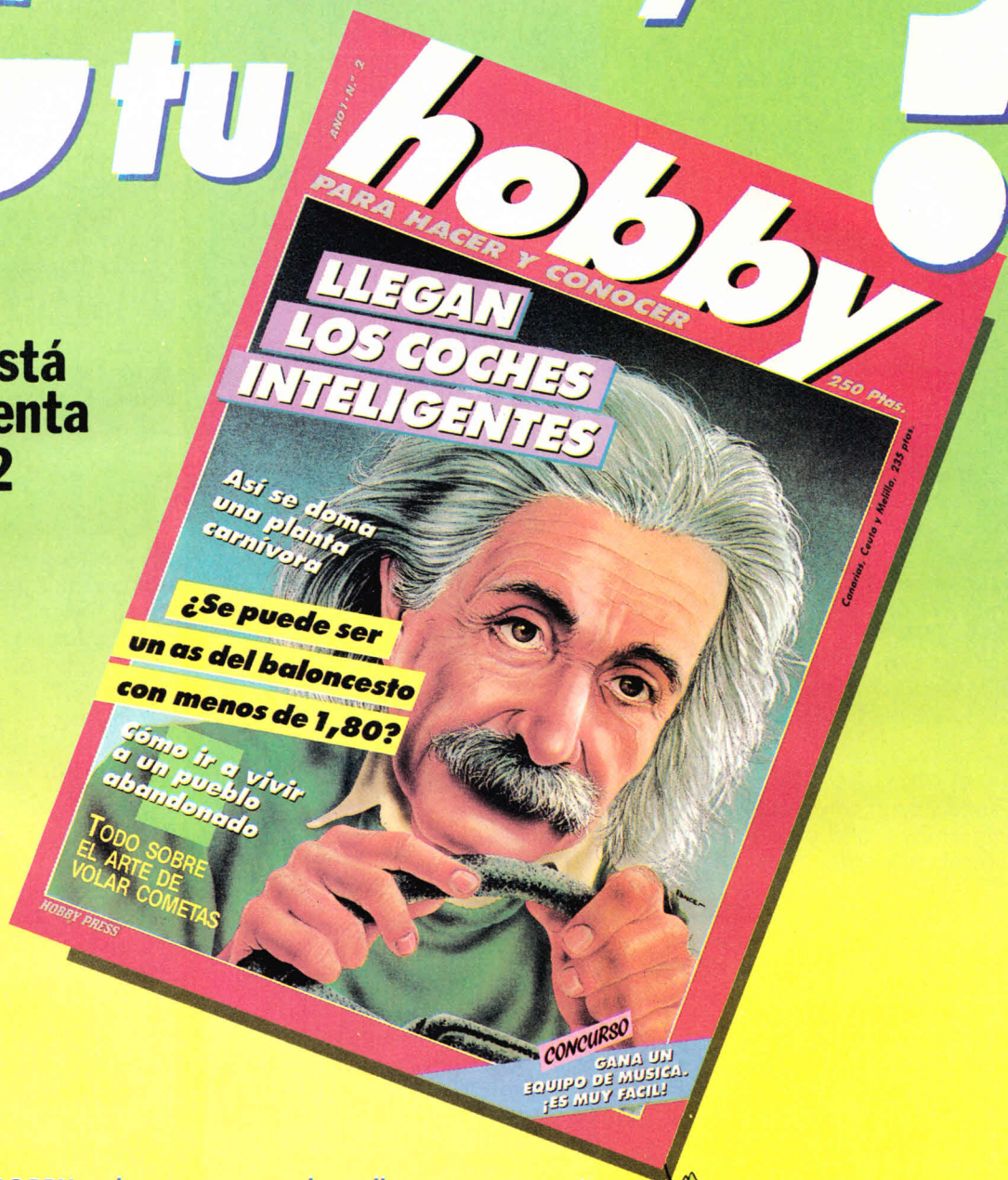
AMSTRAD

**GARANTIA
AMSTRAD
ESPAÑA**

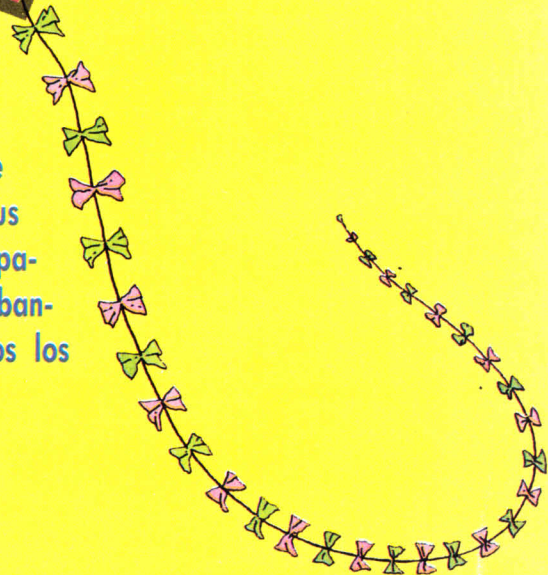
papel de impresora, programas de gestión
libros y todo tipo de accesorios.

¿Tienes ya? tu **hobby**?

Ya está
a la venta
el n.º 2



HOBBY sale este mes a la calle con un montón de temas insólitos que seguro te apasionarán. ¿Sabías, por ejemplo, que puedes ser un campeón de baloncesto aunque midas menos de 1,80? ¿O que es posible domar una planta carnívora y medir sus estímulos? HOBBY te enseña también cosas tan disparates como los pasos a seguir para irte a un pueblo abandonado o el arte de volar cometas. Y, además, todos los meses regalamos una cadena de música. Ten un HOBBY todos los meses. ¡Ya está en tu kiosco el n.º 2!



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!

Vendo gran cantidad de programas a 600 ptas. cada uno para el ordenador Amstrad CPC-464, entre ellos están: Exploding Fist, Raid, Rocky, Dragontorc, etc. Urge por cambio de ordenador. Escribir a: Antonio Bernabéu Pellús. C/ Obispo Rocamora, 6, 2.º B. Orihuela (Alicante).

Desearía contactar con usuarios del **Amstrad** CPC 6128 (unidad de disco). Remitirse a: Belén Soldevilla Peiró. Plaça País Valencià, 7. Potries (Valencia).

Vendo ordenador **Amstrad** con unidad de disco incorporada y monitor en color. Totalmente nuevo y completo. Precio a convenir. También vendería monitor o teclado por separado. Contactar con Gregorio al teléfono de Barcelona (93) 353 68 53. Noches.

Desearía cambiar el Dragontorc (original), por uno de estos programas: Exploding fist, Pole position, Knight lore, Pyjamarama. También me interesan otros programas. Escribir a: Francisco Javier Méndez Serrato. C/ Pedreras, 54. La Línea de la Concepción (Cádiz).

Cambio el siguiente lote de programas: Combat Lynx, Alien 8, Stomp, Multi Gestión; todos por cualquier programa comercial que se interesen ofrecermelo. Escribir a Paco Lozano. C/ Ganduxer, 119. 08022 Barcelona. Contestaré todas las cartas. Tel. 248 03 89.

Desearía contactar con usuarios de **Amstrad** para intercambio de juegos e información general. Preferentemente de Burgos. Puedes dirigirte a José Javier Torres Moreno. C/ Defensores de Oviedo, 1, 5.º izq. 09002 Burgos. Tel. (947) 26 27 99.

Desearía cambiar o vender los siguientes programas: Arnhem, Dummy run, Brian Jacks superstar, Everyone's a wally, Exploding fist y Gremlins.

Llamad al tel. (986) 41 30 56 o escribir a Miguel Angel Casares. Avda. Hispanidad, 59. Vigo (Pontevedra).

Vendo o cambio juegos, prometo respuesta, mandar a poder ser lista, llamar al tel. (976) 86 77 28 o escribir a Borja Sancho. Plaza San Francisco, 7. Borja (Zaragoza).

Desearía contactar con usuarios de **Amstrad** para intercambio de programas, libros, etc., también me interesaría contactar con algún club de **Amstrad**. Respondo a todos. Escribir a Juan Carlos Plaza. C/ Alberto Aguilera, 16. 28015 Madrid.

Vendo programa para **Amstrad** CPC 6128, CPC 664, CPC 464. «Cómo realizar tus propios GDU» más un conversor binario decimal más un conversor decimal binario. Todos originales por sólo 1.500 ptas. Interesados ponerse en contacto a: Javier Sánchez. C/ Segre, 106, escalera B, ático 2.º. 08030 Barcelona o bien llamar al (93) 311 10 47.

Desearía contactar con usuarios del **Amstrad** 6128/664/464 para intercambiar programas, sobre todo utilidades, y trucos. Los interesados escribir a Juan A. Blanco García. Avda. Tomás Giménez, 29 Ent.-2. Hospitalet (Barcelona).

Desearía contactar con usuarios del PCW 8256 para intercambio de programas e ideas y usuarios del CPC 664 o CPC 128 interesados en CP/M. Santi Mas i Plaza. Tvía. Vallés, 5. Igualada (Barcelona).

Desearía contactar con usuarios de **Amstrad** CPC 464 para intercambio de programas de todo tipo. Escribid a: Mary Gómez Olcina. C/ Arzobispo Fabián y Fuero, 21, 11.º. 46009 Valencia. (96) 349 28 53. Enviar lista de programa.

Por cambio de ordenador, vendo o cambio juegos para el **Amstrad** son todos compatibles. Poseo unos 100. Urge por mili. Los interesados llamen al tel. (955) 59 04 58 o escriban a Julián García Barbosa, Los Cantos, chalet 7. Riotinto (Huelva).

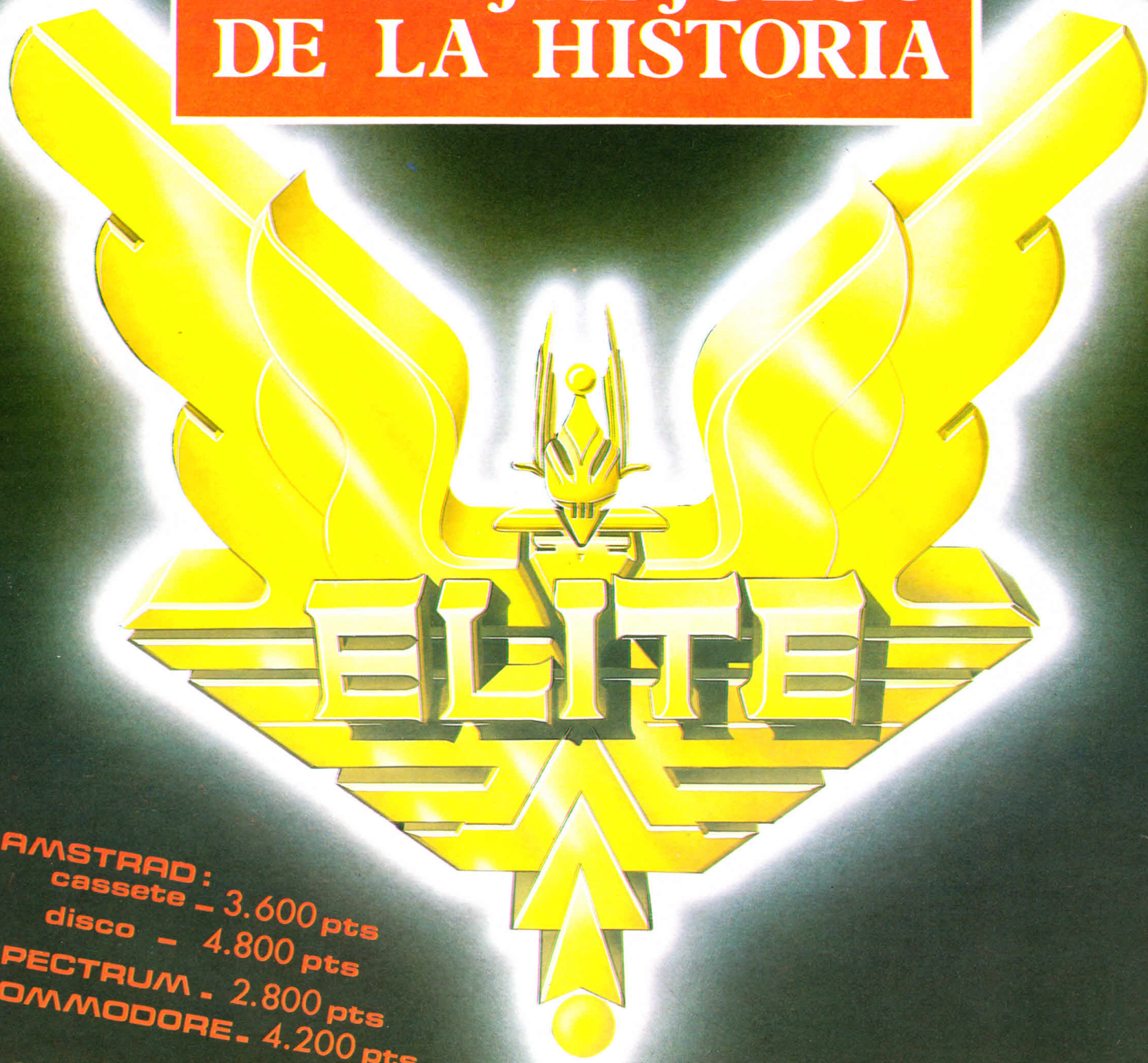
COMERCIAL LEVANTE BAZAR TETUAN

Distribuidor oficial de Amstrad, Spectravideo, Dynadata, Toshiba, etc.

— AMSTRAD 6128 VERDE-DISCO	84.500 PTAS.
— AMSTRAD 6128 COLOR-DISCO	119.500 PTAS.
— AMSTRAD PCW 8256	129.000 PTAS.
— AMSTRAD 472 VERDE-CASSET	59.000 PTAS.
— AMSTRAD 472 COLOR-CASSET	90.000 PTAS.
— ORDENADOR DYNADATA	
— DPC 200-64 K	34.000 PTAS.
— IMPRESORA BROTHER M-1009	36.000 PTAS.
— IMPRESORA BROTHER HR-S	21.000 PTAS.
— JOYSTICK INVESTIK	1.200 PTAS.
— JOYSTICK QUICK SHOT II	1.600 PTAS.
IVA INCLUIDO	

SERVICIO TECNICO
Arenal, 9. 28013 MADRID Tel.: 265 68 55

**EL MEJOR JUEGO
DE LA HISTORIA**



AMSTRAD:
cassete - 3.600 pts
disco - 4.800 pts
SPECTRUM - 2.800 pts
COMMODORE - 4.200 pts

DISPONIBLE PARA TODOS LOS SISTEMAS



SERMA



firebird

Firebird Software - Wellington, New Zealand
L'Esprit du Feu - Paris - France

**Pidelo ya a SERMA C/Bravo Murillo 377
28020 Madrid. Tfos: 7387311-7464.**

**GOLD
EDITION**

SINCLAIR STORE

REGALO SEGURO

**FINANCIACION
HASTA 36 MESES**



POR LA COMPRA DE TU ORDENADOR ESTE EQUIPO COMPLETO DE BASKET ES TUYO. TAMBIEN REGALO DE INTERFACE 2 Y UN JOYSTICK AL COMPRAR TU SPECTRUM 128 O PLUS

- Como siempre curso gratis de informática.
- Spectrum, Q.L., Commodore, Amstrad, Spectravideo y MSX.
- Teclado multifunción con sonido, **13.200** ptas.
- Joystick + Interface + Kempston, **3.200** ptas.
- Lápiz Óptico, **3.500** ptas.
- Bibliografía 25% Dto.
- Tarjeta de Socio Club Sinclair Store.
- Servicio Técnico de Reparaciones, **3.700** ptas.
- Necesitamos distribuidores, somos mayoristas.
- Ampliación de memoria Amstrad 464 ó 6128 256 K, **20.900** ptas.
- Disco de Silicio, **19.900** ptas.
- Por la compra de un spectravideo, 328, **25.000** ptas. o spectravideo 728 MSX, **39.000** ptas. Te regalamos un Joystick Quickshot I más 10 cintas y un cassette.

Amstrad 8512: 169.900 (IVA incluido). Amstrad 8256: 129.900 (IVA incluido). Amstrad 6128: 84.900 (IVA incluido).

*Liquidación Software: West bank, Gremlins, Ole Toro y 200 títulos más, desde **990 ptas.***

sinclair store

SOMOS PROFESIONALES

BRAVO MURILLO, 2
(Glorieta de Quevedo)
Tel. 446 62 31 - 28015 MADRID
Aparcamiento GRATUITO Magallanes, 1

DIEGO DE LEON, 25
(Esq. Núñez de Balboa)
Tel. 261 88 01 - 28006 MADRID
Aparcamiento GRATUITO Núñez de Balboa, 114

AV. FELIPE II, 12
(Metro Goya)
Tel. 431 32 33 - 28009 MADRID
Aparcamiento GRATUITO Av. Felipe II