tu Micro

programas
lenguajes
ordenadores

AÑO 1/N.º 3/4
JULIO/AGOSTO 1984
450 Ptas.



¿QUIEN TEME AL ENSAMBLADOR FEROZ?

LOS PSEUDO-RANDOMS



NUMERO DOBLE CON ¡MAS DE 25 PROGRAMAS!... PARA SPECTRUM, COMMODORE-64, VIC-20, ATARI, ORIC, DRAGON, SPECTRAVIDEO, MPF-II, NEW BRAIN...

SPECTRAVIDEO

El sistema con el que usted crecerá



Los ordenadores personales de Spectravídeo, gracias a su perfecto diseño y gran facilidad de manejo, se pondrán enteramente en sus manos desde el momento que los desembale. Y, aún más, son suficientemente capaces y ampliables como para cubrir todas las necesidades durante mucho tiempo



SV-318 SV-328

CUANDO SE TRATA DE ORDENADORES PERSONALES SPECTRAVIDEO ESTA MAS ALLA DE CUALQUIER COMPARACION

BASIC extendido de Microsoft, integrado, convierte a Spectravídeo en los primeros ordenadores verdaderamente programables y asequibles del mercado.

Extraordinaria memoria. 32 Kbytes de ROM, ampliables a 96 Kbytes y, 32 u 80 Kbytes de RAM, ampliables a 256 Kbytes.

Expansión diagonal. Un sistema de 14 periféricos plenamente soportado. Incluyendo el adaptador de juegos Colecovisión. Unidad de expansión de 7 slots, controlador de disco flexible, cassette, interface para cartucho, etc.

Gráficos avanzados. El sistema SV ofrece 16 colores en alta resolución y, más importante aún, 32 sprites programables que permiten unas impresionantes posibilidades de control de animación en pantalla.

Otras muchas características atractivas. Tales como un microprocesador Z80A con un rápido (3,6) reloj interno, slot para la conexión de cartuchos, diez teclas de función programables por el usuario, tres canales de sonido (8 octavas por canal), perfil bajo y estilizado diseño.

TABLA DE COMPARACION DE FUNCIONES CON DISTINTAS MARCAS

	SPECTRAVIDEO SV-328	SPECTRAVIDEO SV-318	APPLE IIE	ATARI 800	COMMODORE 64	BBC MODEL B	DRAGON 32	EL MAS VENDIDO HASTA AHORA
CAPACIDAD DEL ORDENADOR								
ROM INCORPORADA	32K	32K	16K	10K	20K	16K	16K	16K
AMPLIABLE A	96K	96K	?	42K	?	64K	?	?
MBASIC INCORPORADO	SI	SI	SI	ADIC.	NO	NO	SI	NO
RAM INCORPORADA	80K*	32K* * *	64K	48K	64K	32K	32K	16K
AMPLIABLE A	256K* *	256K* *	64K	NO	N/A	32K	64K	48K
CARACTERISTICAS DEL TECLADO								
NUMERO DE TECLAS	87	71	63	61	66	73	53	40
TECLAS DEFINIBLES POR EL USUARIO	10	10	?	4	8	10	?	?
PROCESO DE TEXTOS	SI	SI	NO	NO	NO	NO	NO	NO
GRAFICOS (DESDE TECLADO)	SI	SI	NO	SI	SI	SI	SI	
TECLAS MAYUSCULAS Y MINUSCULAS	SI	SI	SI	SI	SI	SI	SI	SI
CARACTERISTICAS DE JUEGO Y SONIDO								
RANURAS DE CARTUCHOS SEPARADAS	SI	SI	NO	SI	NO	NO	SI	NO
JOYSTICK INCORPORADO	NO	SI	NO	NO	NO	NO	NO	NO
COLORES	16	16	15	128	16	16	9	8
RESOLUCION (PUXELK)	256×192	256×192	280×180	320×192	320×200	256×640	256×192	256×192
SPRITES	32	32	N/A	4	8	?	16	?
CANALES DE SONIDO	3	3	1	4	3	1	3	?
OCTAVAS POR CANAL	8	8	4	4	9	3	5	3
ENVOLTURA A.D.S.R.	SI	SI	NO	NO	SI	SI	NO	NO
CARACTERISTICAS DE LOS PERIFERICOS								
CASSETTE	2 CANALES	2 CANALES	1 CANAL	2 CANALES	1 CANAL	2 CANALES	?	?
AUDIO 110	SI	SI	NO	SI	NO	?	?	?
MC INCORPORADO	SI	SI	NO	NO	NO	NO	NO	NO
CAPACIDAD DE LOS DISCOS	256K	256K	143K	92K	170K	100K	?	?
LINEA SENCILLA	SI	SI	NO	NO	NO	NO	NO	NO
COMPATIBILIDAD CON CP/M (programas standard de 80 columnas)								
CP/M 2.2	SI	SI	NO....	NO	NO....	SI	NO	NO
CP/M 3.0	SI	SI	NO	NO	NO	NO	NO	NO
MSX	SI	SI	NO	NO	NO	NO	NO	NO

Microsoft es una marca registrada de Microsoft Corporation
CP/M, es una marca registrada de Digital Research, Inc.

. 64 K disponibles con CP/M 2.2, más 16 K de soporte gráfico
.. 240 K disponibles con CP/M 3.0, más 16 K de soporte gráfico
... 16 K disponibles, más 16 K de soporte gráfico
.... Apple II acepta con una tarjeta de modificación de 40 u 80 columnas CP/M
..... Commodore 64 acepta 40 columnas CP/M
? Dato no confirmado

Las especificaciones están sujetas a cambios sin previo aviso.

● Garantizamos el mantenimiento de nuestros equipos. ● Póliza de seguro, a todo riesgo. ● Programas especiales a necesidades del cliente.

DISTRIBUIDOR OFICIAL PARA ESPAÑA

DYNADATA INFORMATICA

CENTRAL: Sor Angela de la Cruz, 24. Madrid-20
Teléfonos: (91) 279 21 85 - 270 01 93 - 279 28 01

DELEGACION: Aribau, 61, entlo. Barcelona-11
Teléfono: (93) 254 65 48

tu Micro
Año 1/n.º 3/4
julio/agosto 1984

SUMARIO

lenguajes

Estandarización del BASIC (3) 19
Entrada de datos.

programación

Los pseudo-randoms 39
Cómo sacar partido a la función RND...

¿Quién teme al ensamblador feroz? 71
La docilidad del lenguaje ensamblador al descubierto.

equipos

Casadapter 110
Los magnetófonos convencionales al servicio de los ordenadores ATARI.

YIS-503 89
Un representante de la familia MSX con vocación musical.

Juego informático

Misioneros y caníbales 83
ZX-SPECTRUM

... y además

TU MICRO noticias	5, 8 y 9
Libros	6
Bazar Soft	63
TU MICRO diversión	95
Glosario micro-informático	99
Tarjetas de servicios	99
Directorio	112

programas

Campo minado 11
COMMODORE 64

El juego de los barcos 15
VIC-20

Agenda telefónica 24
DRAGON 32/64

ATARI-filler 28
ATARI 600XL/800XL

6 subrutinas de gestión 31
ZX-SPECTRUM

El caso del juego de la vida 44
VIC-20

Front-ball 49
DRAGON 32

Las cuatro en raya 53
ZX-SPECTRUM

Crono-alarm 55
COMMODORE 64

Ping-pong 57
ZX-SPECTRUM

El compatible 60

Adivina un número 65
VIC-20

Poligonales 68
ORIC 1/ ORIC ATMOS

El montón 77
LASER 200

Simón numérico 80
MPF-II

Fichero biblioteca 91
SPECTRAVIDEO SV-318

Basket-bit 96
NEW BRAIN

Oratrón 101
ZX-SPECTRUM

Hijo-poker 105
COMMODORE 64

Literales numéricos 108
ZX-SPECTRUM

¡Mas de 25 programas para compartir el ocio estival con TU-MICRO-ordenador! Lo prometido es deuda y, tal como adelantamos en el número anterior, aquí está el extra de verano de TU MICRO. Una tarea ardua en la que hemos intentado no relegar al olvido a su ordenador: Spectrum, VIC-20, Commodore 64, Atari, MPF-II, Laser 200, Dragon, New Brain, Oric y Spectravideo... Son muy pocos los microordenadores populares que han quedado en el tintero, sin dedicarles al menos un programa.

Y no sólo hemos pensado en ofrecerle material con el que calmar la sed de programas de su equipo... ¡también el lector es importante!, de ahí que hayamos ilustrado algunos programas con historietas amenas, divertidas y —¡desde luego!— con una marcada inclinación didáctica.

Por lo que respecta a los restantes artículos, seguimos avanzando en la serie dedicada a la estandarización del BASIC, este mes ocupándonos de los comandos y funciones para la entrada de datos; también le descubrimos algunos métodos interesantes y efectistas para sacar un mayor partido a la pseudoaletoriedad de la función RND, e incluso le prevenimos de la sana docilidad del «ensamblador feroz»... Pero vaya oteando a su alrededor y localice el árbol más frondoso y con mejor sombra, ¡no importa que no haya enchufe al que conectar el ordenador! primero lea... luego ya compartirá el contenido con su MICRO-disciplinado amigo...

EDITORIAL

AÑO 1 **N.º 3/4** **JULIO/AGOSTO 1984**

Director:
Antonio M. Ferrer Abelló

Redactor-jefe:
Francisco Lara

Redacción:
Fernando López
Juan Manuel López
Inmaculada de la Torre

Colaboradores:
Xavier Alamán, Antonio Carvajal, Ricardo García-Loygorri, Daniel Geijo, Ignacio Moregrega, Pedro del Moral, Manuel Muñoz, Eduardo Pérez, Santiago Ruiz, David Santaolalla, JLM Vázquez de Parga

Producción gráfica:
Vicente Robles

Ilustración:
José Ochoa

Maquetación:
Carlos González Amezúa

Fotografía:
Equipo Gálata

Publicidad:
Lola González
Fernando Castellanos
Teléf. 457 69 23

Redacción, administración, publicidad y suscriptores:
Avda. Alfonso XIII, 141. 28016 MADRID
Teléf. 250 58 20. Télex 49371 ELOC E

Dirección para correspondencia:
Aptdo. de Correos 61.294.
28016 MADRID

Administración:
María Antonia Buitrago

Distribución:
Santiago Ferrer

Suscriptociones:
María Teresa Conde

TU MICRO es una publicación mensual de **Ediciones INGELEK, S. A.**

Presidente:
Ernesto Medina
Copyright 1984
Ediciones INGELEK, S. A.
(MADRID-E)

Reservados todos los derechos. Prohibida la reproducción total o parcial, aún citando su procedencia, de textos, dibujos, fotografías y programas sin autorización escrita de Ediciones INGELEK, S. A. Los programas publicados en TU MICRO sólo pueden ser utilizados para fines privados
Queda prohibido su empleo para fines comerciales.

Imprenta:
Héroes, S. A. MADRID

Distribución:
COEDIS, C/ Valencia, 245. BARCELONA

Precios para España:
Ejemplar: 250 ptas.
Suscripcion anual: 2.500 ptas.
(La suscripción anual incluye 10 números simples más uno doble correspondiente a los meses de julio/agosto.)

Depósito legal:
M. 12.533-1984
ISSN: 0212-9027
Impreso en España

SIMON'S BASIC PARA EL COMMODORE 64

Microelectrónica y Control, distribuidora nacional de los equipos COMMODORE, anuncia el nuevo cartucho SIMON'S BASIC destinado al COMMODORE 64. El nuevo cartucho permite sacar un mayor provecho del mencionado ordenador, al completar el BASIC estándar del equipo con 114 comandos adicionales. El total de comandos incorporados puede clasificarse en doce áreas:

Ayudas a la programación, como KEY y TRACE que dan velocidad y más eficacia a la programación en BASIC.

Comandos de manejo de cadenas de caracteres, como INSERT y PLACE. Dan control total sobre las cadenas alfanuméricas.

Comandos de texto como CENTRE y PRINT AT que facilitan el formatear la pantalla.

Comandos para mejorar la entrada de datos, como FETCH e INKEY que dan control total sobre lo que se entra a través del teclado.

Operadores aritméticos adicionales como MOD y DIV que proporcionan un método simple de división de números enteros.

Comandos de conversión numérica que permiten pasar de código binario o hexadecimal a su equivalente decimal.

Comandos de programación estructurada como PROC e IF..THEN..ELSE que permiten escribir un programa de forma más inteligible.

Ayudas al manejo de la pantalla como SCRSV y COPY que permiten guardar/cargar los datos de la pantalla y/o imprimir sobre papel la información que contiene tanto en alta como en baja resolución.

Comandos de dibujo gráfico como CIRCLE y PAINT que permiten dibujar con facilidad figuras simples y pintar áreas de la pantalla.

Comandos sprite y gráficos definidos por el usuario como DESIGN, MOB SET, DETECT y CHECK que permiten crear y animar «SPRITES» o definir caracteres gráficos.

Comandos de música como WAVE y ENVELOPE que permiten realizar efectos sonoros y componer y ejecutar música.

Comandos de operación con diskette como DIR que simplifican el tratamiento de ficheros.

CLUB USUARIOS DE SPECTRAVIDEO

ISIS, S. A. ha creado el club SV, orientado a que los usuarios de los microordenadores SPECTRAVIDEO a nivel nacional tengan un medio de enlace, intercambio de experiencias e información.

El Club proporcionará el asesoramiento necesario para aprender a utilizar correctamente el SPECTRAVIDEO, aprovechando sus posibilidades y conociendo sus posibles ampliaciones. Existe, además, el propósito de crear y mantener una biblioteca de programas destinados a múltiples aplicaciones, confeccionados por los propios socios y a disposición de los mismos. Dispondrá también de una guía de publicaciones (libros, revistas...) específicas para los equipos SV y que puedan resultar útiles para el usuario, e incluso organizará cursos de programación especializados en el SPECTRAVIDEO.

El club enviará a los socios información acerca de novedades, trucos y consejos de programación y pondrá su disposición el teléfono y el buzón del club para contestar a sus preguntas y resolver cualquier tipo de dudas.

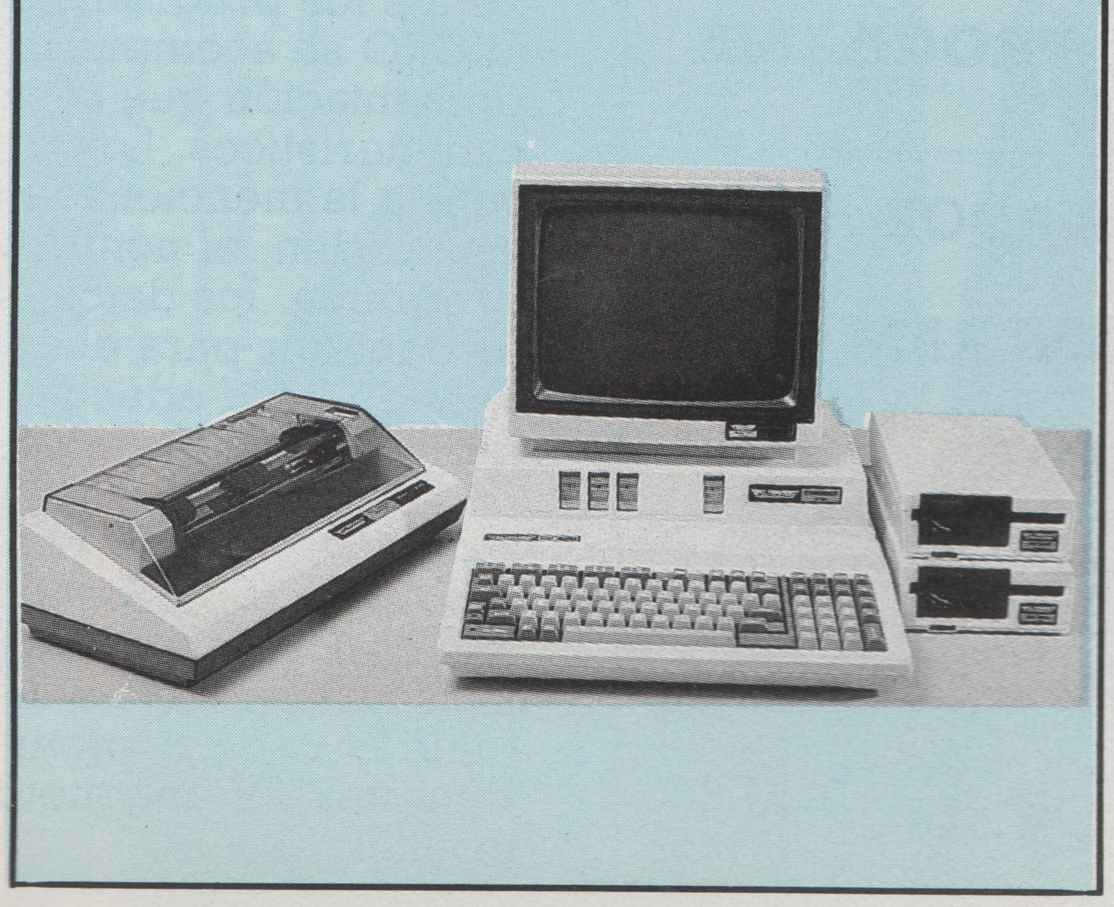
La inscripción al club tiene carácter gratuito y debe realizarse contactando con:
ISIS, S. A.
Lagasca, 125-6.º dcha.
28006 MADRID
Tel. 411 55 61 ó 411 55 43

NUEVOS PRECIOS DE SPECTRAVIDEO

Dynadata, S. A. firma distribuidora en España de la gama de microordenadores y periféricos SPECTRAVIDEO, ha actualizado la lista de precios de algunos de sus productos; los nuevos precios son los siguientes:
SV-318: 49.900 Ptas.
SV-328: 76.000 Ptas.
SV-806 (tarjeta para 80 columnas): 23.800 Ptas.
SV-904 (unidad de casete): 7.900 Ptas.

Por lo demás, la gama de productos SPECTRAVIDEO se ha visto incrementada con dos nuevas incorporaciones que ya pueden adquirirse en los concesionarios autorizados por Dynadata:
SV-605: Super-expander con controlador de discos, una unidad para disco flexible e interface centronics: 99.900 Ptas.

«SV-605 A: Super-expander con controlador de discos, dos unidades para disco flexible e interface centronics: 148.000 Ptas.



SECOINSA Y LOGIC CONTROL

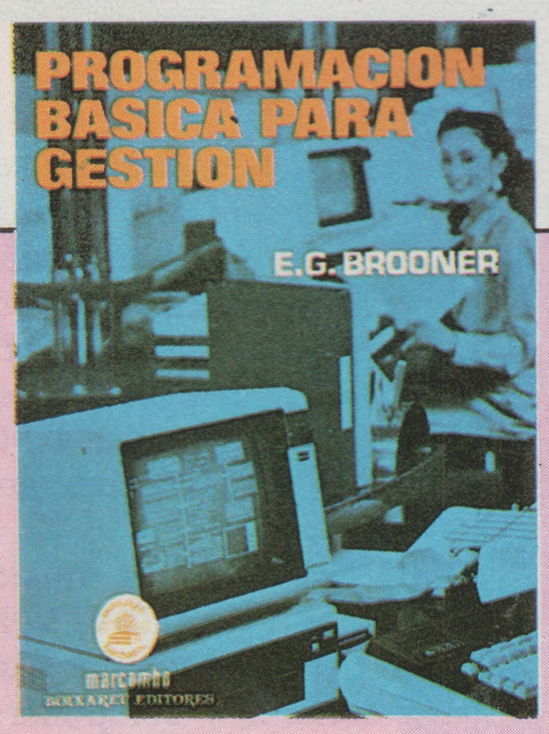
Dos empresas españolas de informática, Secoinsa y Logic Control, han suscrito un contrato para la comercialización de un ordenador personal de tipo profesional, así como del software necesario para abastecerlo de aplicaciones.

Secoinsa, empresa dependiente del INI con experiencia en el desarrollo de hardware diseñado por técnicos españoles (actualmente está exportando microordenadores de la Serie-20 y sus modelos de impresoras 1555 y 1556), será quien se ocupe de la fabricación del nuevo ordenador personal.

Logic Control, empresa 100% nacional de capital privado que desarrolla su actividad en el sector informático y que exporta software a varios países americanos, se ocupará de alimentar la biblioteca de software del nuevo producto con programas de creación propia.

En la línea de crecimiento que caracteriza al sector informático (en España se prevé un parque de 200.000 equipos para el año 1987), no cabe la menor duda que el fruto de esta colaboración entre las dos empresas nacionales puede ser un producto netamente competitivo en el mercado internacional.

LIBROS



Programación básica para gestión

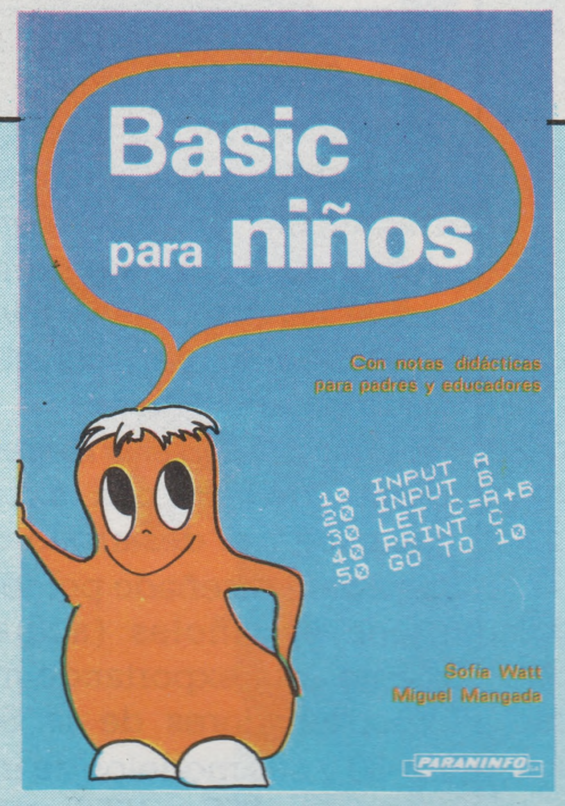
La definición de la obra queda patente en una de las frases que el autor vierte en el prólogo: «La finalidad de este libro, programación básica para gestión, es ayudar al ejecutivo a entender algunos de los principios del desarrollo del software comercial...». En consonancia con este objetivo, el texto elude cualquier tipo de referencia técnica o alusión al aspecto tecnológico del ordenador, no obstante, sí supone en el lector algunos conocimientos sobre el lenguaje BASIC.

Aunque su destinatario inmediato sea la persona que desea evaluar programas de tipo comercial, para adquirir una idea de la idoneidad de los mismos a sus necesidades, también el programador encontrará en las descripciones y en los programas ejemplo material útil para la confección de programas específicos.

La obra se desarrolla en un total de 10 capítulos más un apéndice dedicado al código ASCII y un glosario final.

Autor: E. G. Brooner
Editorial: Marcombo
Idioma: Castellano
Edición: 1984
Páginas: 178
Precio: 980 Ptas.

BASIC



BASIC para niños

La estructura del libro está orientada a que el niño y el adulto descubran juntos el mundo de la programación. La función del adulto se concreta en actuar de guía y, a su vez, estimular al niño para que este vaya descubriendo los secretos del diálogo con el ordenador a través de su propia experiencia, avanzando progresivamente en el conocimiento del lenguaje BASIC.

Quince son los capítulos que integran el libro. El primero —«Unas notas previas» define el contenido y el tratamiento otorgado al material que compone la obra y plantea los requisitos necesarios para que la lectura se convierta en un eficaz y ameno aprendizaje. El segundo capítulo —«Para empezar»— resume la idea básica de ordenador y de programación. Los trece capítulos restantes —excepto los dos últimos, dedicados a diagramas de flujo y a programas de juego— abordan el estudio del lenguaje BASIC, encadenando sucesivos comandos acompañados de los oportunos comentarios y ejemplos.

Autores: Sofía Watt y Miguel Manglada.
Editorial: Paraninfo.
Idioma: Castellano.
Edición: 1984
Páginas: 128
Precio: 500 Ptas.

BASIC



Todos los programas publicados en TU MICRO se acompañan de una tabla de presentación que resume sus datos característicos. Los tres bloques de que consta la mencionada tabla reflejan, en este orden, el nombre o título dado al programa, los datos básicos del mismo (ordenador para el que se ha concebido, versión RAM adecuada, lenguaje en el que se ha confeccionado el programa y tipo del mismo) y la configuración de sistema adecuado para ejecutarlo y almacenarlo. En esta última zona la configuración se define recurriendo a los símbolos de las unidades implicadas (unidad central, monitor de vídeo o receptor TV, magnetófono a casete, joystick...).

Cuando se es COMMODORE es muy difícil ser modesto



COMMODORE 64

Cuando se tiene 64 K de memoria, una magnífica resolución, 16 colores, efectos tridimensionales con sprites, un sonido equivalente al de un sintetizador, un teclado profesional con 62 caracteres gráficos, toda una amplia gama de periféricos, la más completa gama de programas educativos, profesionales y de video-juegos...; en resumen, cuando se es un ordenador personal como no existe ningún otro en el mercado y el más vendido mundialmente, es muy difícil decir sin orgullo que eres un Commodore-64.

Claro que más difícil todavía es decir sin orgullo que tienes un Commodore-64. ¿Por qué no lo comprueba?

El ordenador personal de la familia más potente

commodore COMPUTER

MICROELECTRONICA Y CONTROL, S.A.
c/. Taquígrafo Serra, 7, 5.º Barcelona-29
c/. Princesa, 47, 3.º, G Madrid-8

HP-110: EL PERSONAL PORTATIL DE HEWLETT-PACKARD

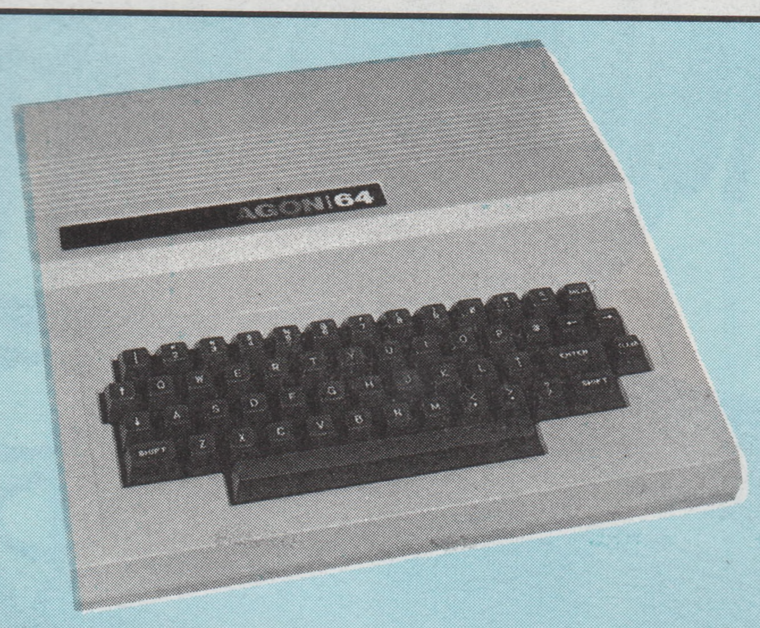
La conocida firma americana Hewlett-Packard acaba de presentar su nuevo ordenador personal portátil: el HP-110. Un equipo cuyo peso es inferior a los 4 kilos, con unas dimensiones de tan sólo 32×25×7 cm y alimentado a partir de baterías.

Esta presentación de HP en el mercado de los microordenadores portátiles de gestión constituye el peldaño siguiente al HP-150 (ordenador con pantalla sensible al tacto) presentado en Europa a principios de este mismo año.

El HP-110 se ofrece como un producto de funcionalidad múltiple, utilizable como ordenador portátil autónomo para profesionales, como equipo auxiliar conectado a otros ordenadores HP o incluso IBM, o como terminal inteligente.

El equipo incorpora algunos conocidos programas de gestión como el paquete Lotus 1-2-3, el MemoMaker y el gestor de aplicaciones personales P.A.M. La biblioteca creada entorno al HP-110 incluye otros programas de aplicación suministrables en disco (hojas electrónicas, tratamiento de textos, programas para la generación de gráficos...). Un dato reseñable es que su memoria residente duplica a la del IBM-PC, al igual que su capacidad de almacenamiento externo.

La introducción en el mercado europeo del HP-110 está prevista para el próximo otoño, una vez hayan concluido los trabajos necesarios para la adaptación del equipo a las características de los diversos países en los que va a comercializarse.



NACE UN NUEVO DRAGON

El popular DRAGON-32 ya tiene un hermano mayor: el DRAGON-64. Aún conservando las características esenciales de su predecesor, el nuevo DRAGON-64 no aparece como un simple sustituto del anterior modelo, sino que aporta nuevas y atractivas posibilidades.

Siguiendo en la línea trazada por el DRAGON-32, el nuevo equipo está basado en el 6809 E: un microprocesador con arquitectura externa de 8 bits, aunque dotado de una capacidad de cálculo con palabras de 16 bits que facilita la implantación de versiones evolucionadas de lenguajes de alto nivel como el BASIC, Pascal y C.

La presentación del nuevo equipo ha sido parálela a la del Disk Drive (unidad para disco flexible de 5 y 1/4 pulgadas) y a la del sistema operativo en disco OS-9 que ampliará las posibilidades del DRAGON-64 hacia el campo de las aplicaciones profesionales y de gestión.

El DRAGON-64 permite tres modos de funcionamiento dependiendo de la zona de memoria RAM en explotación. El modo «32 K» equipará su funcionalidad a la del DRAGON-32. El modo «48 K» que facilita el acceso a un total de 48 Kbytes de RAM y dispone de un intérprete BASIC de 16 K. Y, por último, el modo «64» a través del que se hace un empleo intensivo de los 64 K de memoria RAM incorporados al equipo. Por lo que respecta al apartado de comunicaciones, el DRAGON-64 dispone de un acceso de entrada/salida ajustado al estándar de interface RS/232C, lo que abre la compatibilidad del equipo, con una amplia gama de impresoras y otros dispositivos periféricos, al tiempo que permite crear redes con la asociación de varios equipos.

Un punto a señalar es que el nuevo DRAGON-64 comparte las prestaciones de su predecesor; así, por ejemplo, se mantienen los 256×192 pixels en la pantalla de alta resolución y todo el software disponible para el modelo 32, tanto en casete como en cartucho, es compatible con el DRAGON-64.



NUEVO CLUB DE MICRO-ORDENADORES

El «Círculo Ourensan Vigués», con sede en Vigo, c/ Venezuela, 48 (Tel. 42 25 19) amplía sus actividades al campo informático creando un club de microordenadores para SPECTRUM, COMMODORE-64, VIC-20, ORIC-1 y ZX-81. Un club cuyo objetivo esencial es intercambiar experiencias, mantener reuniones periódicas de sus miembros, recopilar información y ayudar a los no iniciados en el uso del microordenador. En la actualidad, el club pone a disposición de sus miembros un gran número de publicaciones especializadas y más de 400 programas en cinta.

Cualquier interesado en incorporarse al club —sea o no usuario de algún microordenador— puede dirigirse a la sede del mismo por carta o mediante llamada telefónica que puede realizar cualquier sábado de 4 a 6 de la tarde.

SX-64: EL PORTATIL DE COMMODORE

El nuevo COMMODORE SX-64 es una versión portátil del conocido COMMODORE 64. Lleva integrados un monitor en color de 127 mm. de diagonal, una unidad de disco 1541 y amplificador y altavoz para audio. Todo el software del COMMODORE 64 puede ejecutarse en el nuevo modelo, desde el proceso de texto EASY SCRIPT hasta los juegos de «marcianitos», pasando por contabilidades, gestión de ficheros, ampliaciones de BASIC, lenguajes como ASSEMBLER, LOGO, etc. Los periféricos del COMMODORE 64 y el VIC-20 (unidad de disco, impresora y plotter) pueden conectarse directamente al COMMODORE SX-64. Varios interfaces le permiten usar como periféricos una gama completa de unidades de disco de gran capacidad, plotters para trabajos científicos e impresoras matriciales rápidas y de alta calidad (impresoras de margarita). También puede conectarse el monitor en color de 13" COMMODORE 1702, cuando el usuario lo necesite. Sus dimensiones son de 368×368×127 mm.

ACUERDO DIGITAL-VISUALDATA

DIGITAL y VISUALDATA han firmado un acuerdo para comercializar conjuntamente los sistemas de VIDEOTEX privado IVS-3 de AREGON Int. Ltd., en Norma CEPT.

El sistema IVS-3 reside en un ordenador VAX de DIGITAL y es el sistema abierto de VIDEOTEX privado de mayor penetración en el mercado mundial y el primer sistema operativo en España; orientado a la Norma CEPT (futura normativa de VIDEOTEX en España), ofrece también el protocolo de conexión a los ACPS (Centros Privados de Servicio VIDEOTEX).

El IVS-3 puede funcionar como un sistema independiente de VIDEOTEX con aplicaciones específicas del usuario, o como un FRONT-END a un HOST del usuario, para la explotación a través del VIDEOTEX de las aplicacions existentes.

Existe un paquete de demostración a partir del que los interesados en estos sistemas de VIDEOTEX privados pueden realizar las oportunas valoraciones del programa.

DIGITAL es la segunda compañía a nivel mundial de informática y el principal fabricante de equipos de informática para oficinas, fábricas, laboratorios, informática personal y pequeña gestión.

VISUALDATA, S. A. es la pionera en España dentro del campo empresarial de proveedores de servicios VIDETOTEX.



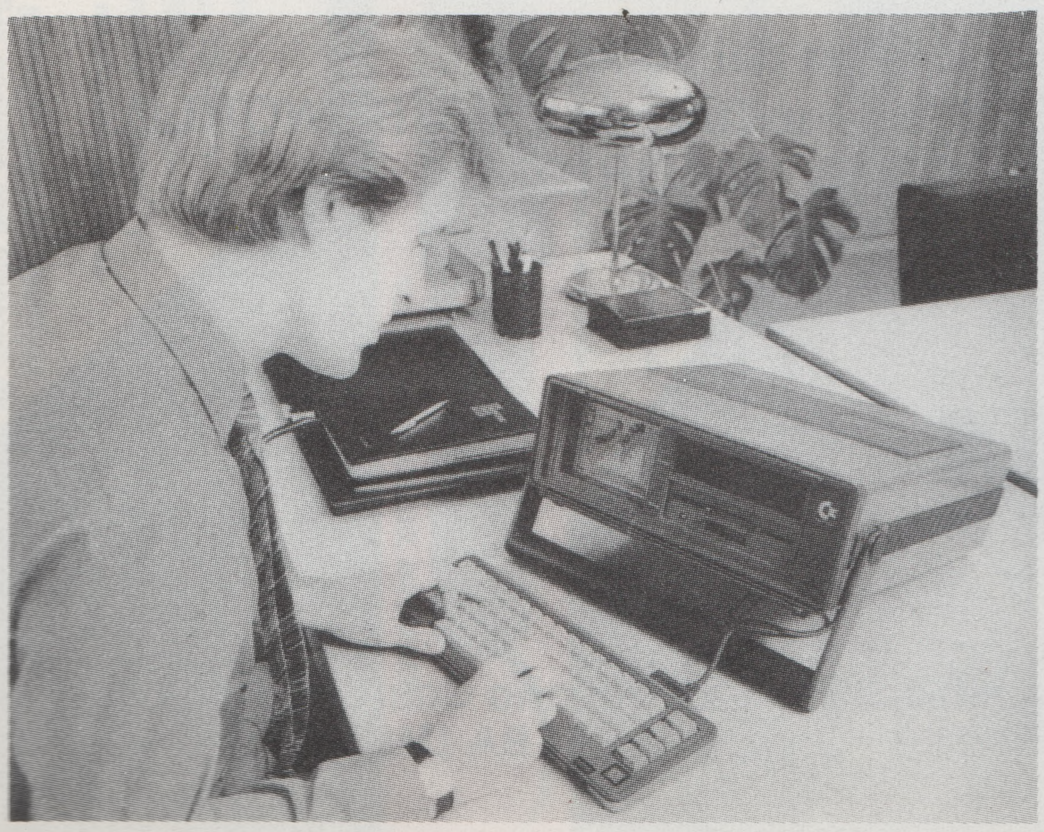
NUEVA SEDE SOCIAL DE DIGITAL EQUIP. CORP. S. A.

DIGITAL ha trasladado sus oficinas centrales en el mes de Julio al nuevo edificio «DIGITAL» ubicado en la Urbanización Mirasierra de Madrid.

El edificio «DIGITAL» cuenta con más de 5.000 m² dedicados exclusivamente al servicio a los clientes, amigos y colaboradores. Dispone de amplias salas para cursos, diversas áreas de demostración y presentaciones.

DIGITAL España, fundada en 1975 ha venido observando a lo largo de estos años un crecimiento acumulativo de más de un 63%. En el año fiscal que finalizó en junio de 1983, el crecimiento de DIGITAL fue de un 85% con respecto a la facturación del año anterior.

Edificio «DIGITAL»
Cerro del Castanar, 72
Mirasierra/28034 MADRID
Teléf.: 734 70 02 y 734 10 00



COMUNICADO DE TEXTRONIC, S. A.

A raíz del estudio sobre el ORIC-ATMOS publicado en el número anterior de nuestra revista, la firma importadora del equipo —TEXTRONIC, S. A.— nos ha remitido las siguiente carta en la que se aportan algunas precisiones:

«Muy Sres. nuestros:

Una vez leído el estudio hecho por Vds. sobre el ordenador ORIC Atmos en el n.º 2 de su estimada revista, y del que somos distribuidores en exclusiva, me permito hacerles dos puntualizaciones mínimas, pero a la vez muy interesantes tanto de cara al distribuidor como al posible usuario y que son:

— TEXTRONIC, S. A. está sita en c/ Preciados, 39

— El completo manual inglés al que Vds. hacen referencia está *literalmente* traducido al castellano, siendo éste apartado de vital importancia destacarlo por ser una de las pocas medidas para defender de las importaciones paralelas la comercialización del aparato, ya que cualquier ORIC Atmos que no sea suministrado por un Distribuidor Oficial no podrá contar con el citado manual traducido, ni por supuesto con la garantía del fabricante.

TEXTRONIC, S. A.

MICRO WORLD

I CONCURSO NACIONAL DE PROGRAMACION

1.000.000

de pesetas en premios.

1er Premio: 500.000 pts.

Premios de 50.000 pts. para los 10 finalistas.

Solicite más información y bases del concurso,
directamente en los centros MICROWORLD o escribiendo a MICROWORLD
c/ MODESTO LAFUENTE Nº 63 - MADRID - 3

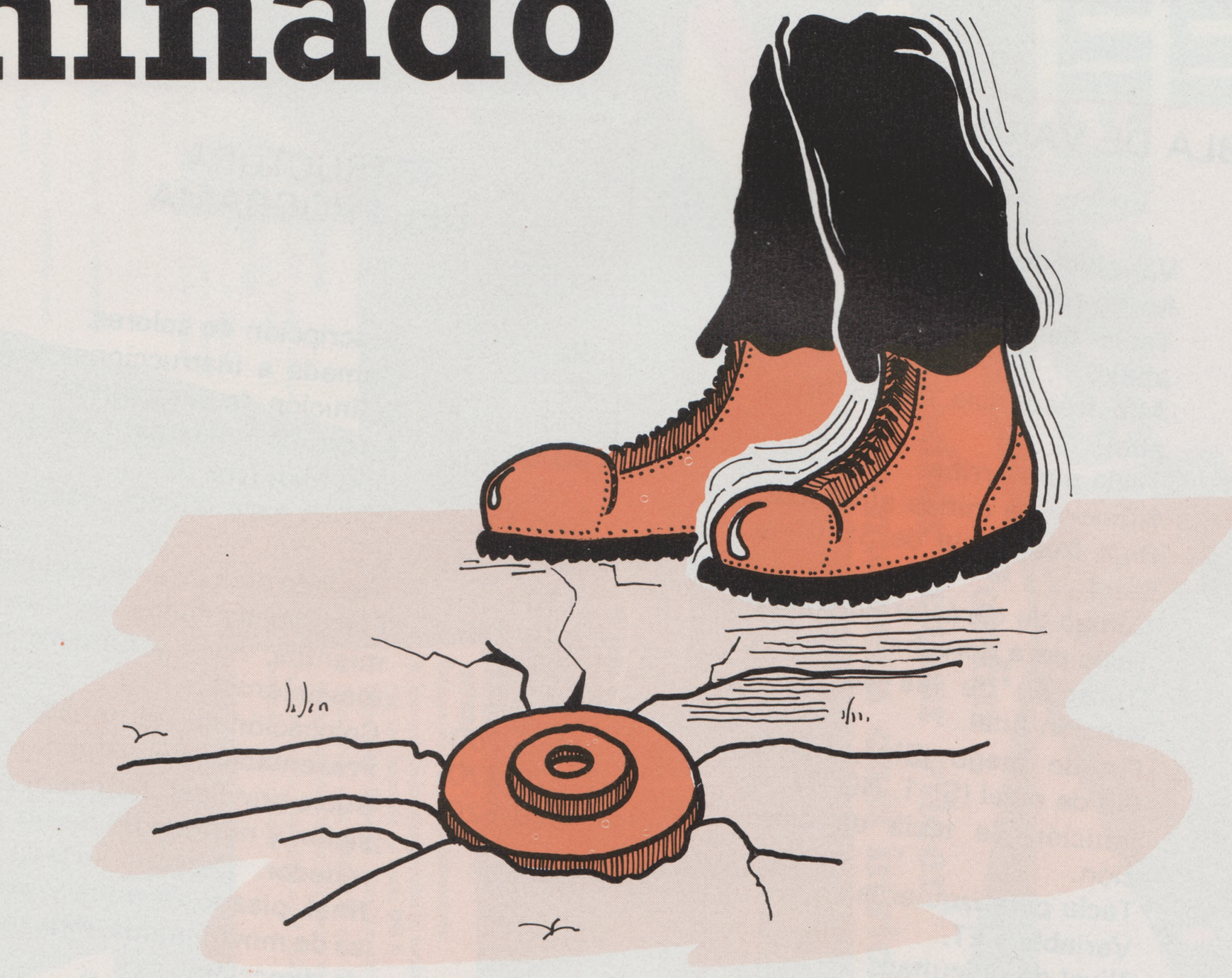
MICRO WORLD

c/ Modesto Lafuente, 63
MADRID-3

c/ Colombia, 39
MADRID-16

c/ Honorio Gonzalo, 2
VILLALBA (Madrid)

Campo minado



Campo minado

COMMODORE-64

VERSION BASE

BASIC

JUEGO

Un juego cuyo objetivo consiste en cruzar un campo de minas sin perecer en el intento... ¡Toda una prueba de valor y sangre fría!

El jugador parte de la esquina inferior izquierda del terreno minado y debe cruzarlo, llegando ileso a la casilla superior derecha. Al llegar a cada casilla, el ordenador emitirá un sonido característico si existe alguna mina en cualquiera de las casillas vecinas... ¡Cuidado al avanzar! Si no percibe el aviso de proximidad de mina, puede saltar a cualquiera de las casillas situadas alrededor de la que ocupa.

Según el nivel de dificultad elegido por el propio jugador al empezar su arriesgada travesía, será mayor o menor el número de minas con el que puede tropezar (¡a mayor nivel, más minas!). En cualquier caso, siempre hay cuatro casillas libres de artefactos explosivos; estas son las situadas a la derecha y encima de la casilla de salida y a la izquierda y debajo del punto de llegada.

El juego permite utilizar para el desplazamiento las teclas que más se acomoden a cada jugador; antes de empezar el juego, éste puede elegir las teclas que desea utilizar para moverse en las cuatro direcciones del campo minado (arriba, abajo, izquierda y derecha).

Una buena táctica para intentar salir con buen pie en los niveles altos de dificultad, corroborada por varias horas de andar entre minas, consiste en ir anotando en cada casilla recorrida el número de minas que hay alrededor; de esta forma —combinando lógica y paciencia— puede ir elaborando un mapa cada vez más perfecto del campo minado.

Como podrá observar al ejecutar el programa, este incorpora detalles muy interesantes y agradables como, por ejemplo, la ambientación sonora del desarrollo del juego (con tonadilla final incluida) y la «repetición de la jugada», partiendo de la casilla inicial, cada vez que tenga un desafortunado propiezo con una mina.

Antes de pasar al juego, una última observación: debido a la existencia de un buffer de teclado en el COMMODORE-64 es recomendable no andar con prisa en los saltos, de lo contrario, para cuando suene el aviso de mina próxima a una cuadrícula de distancia, la «inercia» del movimiento puede ser fatal.

¡Buena suerte!...

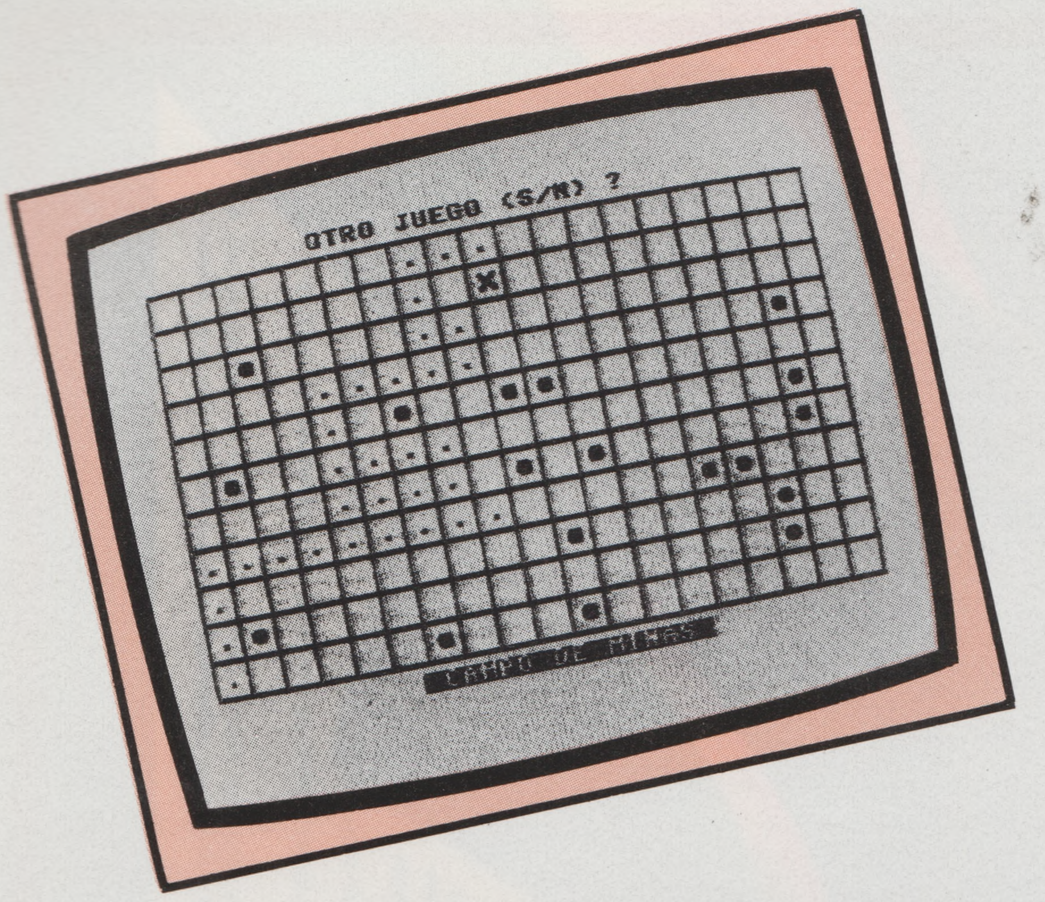
tu Micro

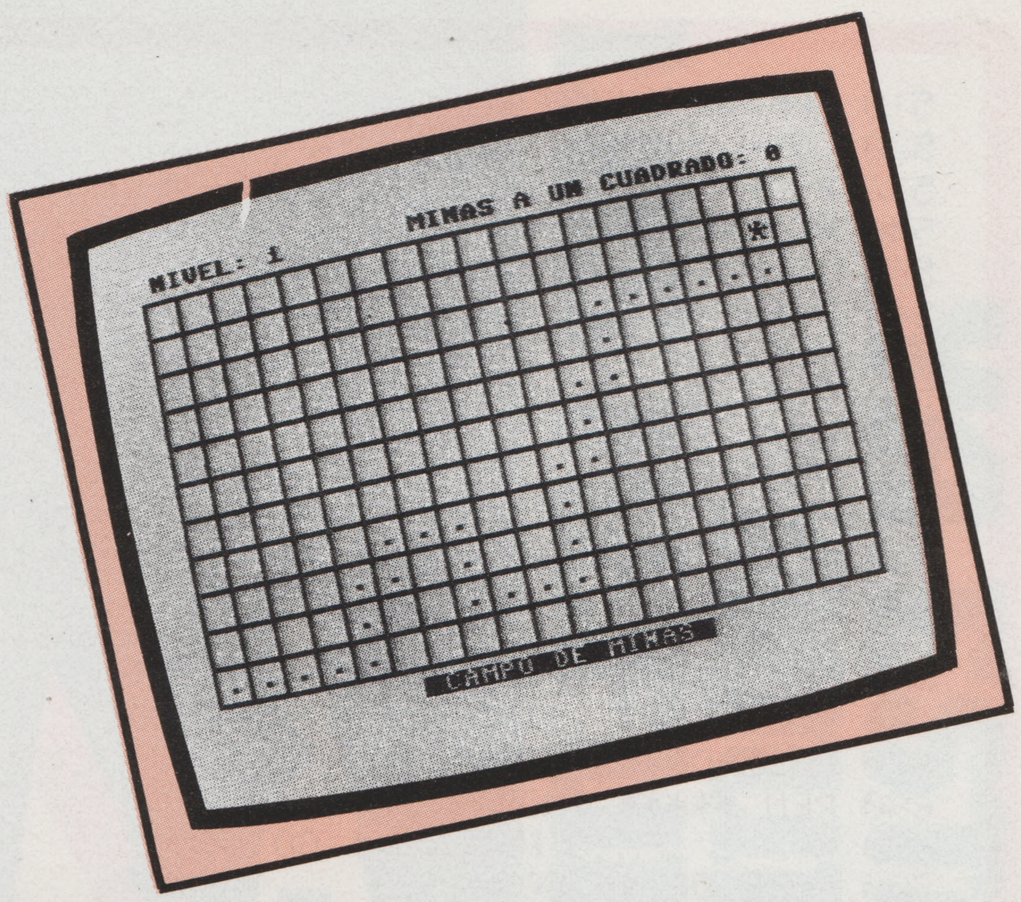
TABLA DE VARIABLES

A$	Variable de vaciado del buffer de teclado.
AB$	Tecla desplazamiento hacia abajo.
AF(*)	Alta frecuencia de SS (Byte alto).
AR$	Tecla para arriba.
B	Número de minas alrededor.
BF(*)	Baja frecuencia de SS (Byte bajo).
CR	Código de carácter en salida.
DR$	Tecla para derecha.
DU(*)	Duración de las notas en canción final.
FJ	Fin de juego (SI=1, NO=0).
FL	Fin de nivel (SI=1, NO=0).
F$	Petición de tecla de dirección.
IZ$	Tecla para izquierda.
K$	Variable GET.
L	Nivel de dificultad.
MI(*,*)	Matriz de minas (O=sin mina, 1=con mina).
N	Número de jugada actual.
N(*)	Guarda todas las posiciones del recorrido para repetición.
OP	Se iguala a PO para poner un punto en la posición anterior.
P	Número máximo de movimientos.
PO	Pasa de coordenadas x,y a POKE de pantalla.
SS	Frecuencia de las notas en canción final.
T$	Variable GET para volver o no.
X	Posición x actual.
x1, y1	Comprobación de minas en los 8 puntos colindantes.
x2, y2	Transformación de x1, y 1 al punto actual.
Y	Posición y actual.
H,I,J,K, S, T, W	Variables FOR.

ESTRUCTURA DEL PROGRAMA

60	Descripción de colores.
70	Llamada a instrucciones y a definición de caracteres.
80-90	Ajuste de notas para la canción final (voz 3).
100-120	POKEs para música, voces 1 y 2.
130	Vaciado del buffer de teclado.
140-230	Elección de todos los movimientos, nivel y número de movimientos.
240-260	Colocación de las minas.
270-370	Presentación del campo.
380-520	Bucle principal. Actualiza posición y número de minas alrededor. Detecta llegada al final, pisado de mina y exceso de movimientos. Pide tecla de dirección.
530	Ruido de los pasos al cruzar el campo.
540-640	Detecta minas alrededor, enseña nivel y número de minas. Si hay mina próxima activa la alrma.
650-760	Realiza los cuatro movimientos.
770-890	Efectos de fin de juego, explosión y flash de «DESTRUIDO».
900-970	Subrutina de repetición de la jugada.
980-1040	Subrutina de incremento del nivel de dificultad.
1060-1120	Subrutina de cambio de caracteres.
1130-1150	Subrutina para tocar la música final.
1160-1270	Subrutina de presentación de instrucciones.
1280-1290	Datos para música.





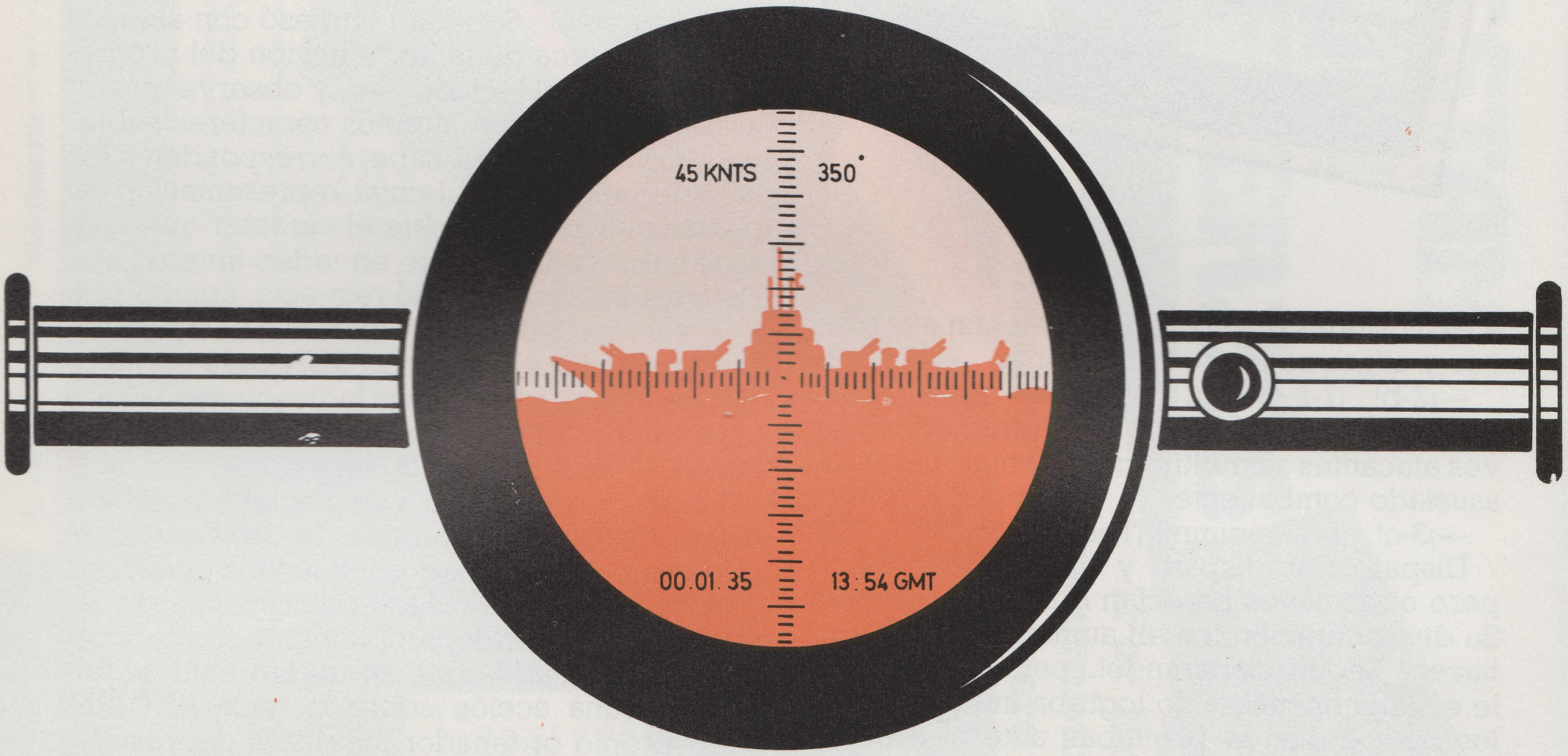
LISTADO

```
10 REM ************************************
20 REM *          CAMPO MINADO            *
30 REM * POR   ANTONIO CARVAJAL  6/84 *
40 REM ************************************
50 :
60 POKE 53281,5:POKE 53280,5:POKE646,0
70 GOSUB 1170:GOSUB 1070
80 CLR:FORS=0TO7:READSS:AF(S)=INT(SS/256):BF(S)=SS-AF(S)*256:NEXT
90 FORT=0TO7:READDU(T):NEXT
100 CR=127:PRINT"■":POKE198,0
110 POKE54296,15:POKE54277,2:POKE54273,50:POKE54272,223
120 POKE54284,12:POKE54280,51:POKE54279,97
130 FORI=1TO10:GETA$:NEXT
140 PRINT"■PULSE TECLA PARA IZQUIERDA_":GETIZ$:IFIZ$=""THENPRINT"■■■":GOTO140
150 PRINT"■PULSE TECLA PARA DERECHA_":GETDR$:IFDR$=""THENPRINT"■■■":GOTO150
160 PRINT"■PULSE TECLA PARA ARRIBA_":GETAR$:IFAR$=""THENPRINT"■■■":GOTO160
170 PRINT"■PULSE TECLA PARA ABAJO_":GETAB$:IFAB$=""THENPRINT"■■■":GOTO170
180 INPUT"■PULSE NIVEL (1-20)";L:IFL<1 OR L>20 THEN PRINT"■■■":GOTO180
190 INPUT"■NUMERO DE MOVIMIENTOS";P:DIM MI(19,11),N(P)
200 POKE198,0:PRINT"■■■■■"IZ$"_:IZQUIERDA":PRINT"■■■■■"DR$"_:DERECHA
210 PRINT"■■■■■"AR$"_:ARRIBA":PRINT"■■■■■"AB$"_:ABAJO
220 PRINT"■■■PARA COMENZAR,PULSE CUALQUIER TECLA_"
230 GETK$:IFK$=""THEN230
240 FOR I=1 TO L*4:X=INT(RND(8)*19):Y=INT(RND(8)*11):MI(X,Y)=1
250 IF(X=0ANDY=0)OR(X=18ANDY=10)OR(X=0ANDY=1)OR(X=1ANDY=0)THENMI(X,Y)=0
260 NEXTI:X=0:Y=0
270 PRINT "■"
280 PRINT "┌┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┐"
290 FOR I=1 TO 10
300 PRINT "│ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │"
310 PRINT "├┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┤"
320 NEXT I
330 PRINT "│ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │"
340 PRINT "└┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┘"
350 PRINT"■■■■■■■■■■■R CAMPO DE MINAS _■"
360 IF FJ=1 THEN RETURN
370 IF FL=1 THEN RETURN
380 REM ***** BUCLE PRINCIPAL *****
390 PO=1905+2*X-80*Y:POKE PO,0:POKE1141,:R:POKE55413,1
400 IFX=18ANDY=10THENFL=1:N=N+1:N(N)=PO :GOSUB1140:GOTO840
410 N(N)=PO
420 IFCR=127THENCR=255:GOTO440
430 IFCP=255THENCR=127
440 OP=PO
450 GOSUB 550
460 IFN=PTHENPRINT"■DEMASIADOS MOVIMIENTOS !A EMPEZAR!":FORT=0TO999:NEXT:GOTO80
470 GET F$
480 IFF$=DR$THENN=N+1:GOSUB530:GOTO660
490 IFF$=IZ$THENN=N+1:GOSUB530:GOTO690
500 IFF$=AR$THENN=N+1:GOSUB530:GOTO720
```

```
510 IFF1=AR$THENN=N+1:GOSUB520:GOTO750
520 GOTO 390
530 POKE 54276,129:FORI=0TO50:NEXT:POKE54276,128:RETURN
540 REM ***** MINAS ALREDEDOR *****
550 IF MI(X,Y)=1 THEN 780
560 FORX1=-1TO1:X2=X-X1:IFX2<0THEN600
570 FORY1=-1TO1:Y2=Y-Y1:IFY2<0THEN590
580 IFMI(X2,Y2)=1THENB=B+1
590 NEXT Y1
600 NEXT X1
610 PRINT"S NIVEL:"L"       MINAS A UN CUADRADO:"B
620 IFB>0THENPOKE54283,17:FORI=0TO50:NEXT:POKE54283,16
630 B=0
640 RETURN
650 REM ***** DERECHA *****
660 X=X+1:IFX>18THENX=X-1:GOTO390
670 POKE OP,46:GOTO 390
680 REM ***** IZQUIERDA *****
690 X=X-1:IFX<0THENX=X+1:GOTO390
700 POKE OP,46:GOTO 390
710 REM ***** ARRIBA *****
720 Y=Y+1:IFY>10THENY=Y-1:GOTO390
730 POKE OP,46:GOTO 390
740 REM ***** ABAJO *****
750 Y=Y-1:IFY<0THENY=Y+1:GOTO390
760 POKE OP,46:GOTO 390
770 REM ***** DESTRUIDO *****
780 POKE54277,12:POKE54276,129
790 FOR I=1 TO 5:PRINT"*QQQQQQQQQQ                    RDESTRUIDO!"
800 FORH=1TO200STEP10:POKE54273,H:NEXT:POKE54276,128
810 PRINT"*QQQQQQQQQQ                    _DESTRUIDO!":POKE54276,129
820 FORH=1TO200STEP10:POKE54273,255-H:NEXTH,I
830 POKE54277,2:POKE54273,50:FJ=1
840 GOSUB270:GOSUB 910
850 IF FL=1 THEN 990
860 PRINT"S          OTRO JUEGO (S/N) ?"
870 GET T$:IF T$="S" THEN GOSUB1000:GOTO80
880 IF T$="N" THEN PRINT"*":END
890 IF T$="" THEN 870
900 REM ***** REPETICION *****
910 FOR I=0 TO 18:FOR J=0 TO 10
920 IF MI(I,J)=1 THEN M=1905+2*I-80*J:POKEM,81
930 NEXT J,I
940 FORW=0TON:POKEN(W),0:POKE54276,129:FORH=1TO100:NEXT:POKE54276,128
950 POKEN(W),46:NEXT:POKEN(W-1),86
960 FORI=1TO2000:NEXT
970 RETURN
980 REM ***** CAMBIO DE NIVEL *****
990 PRINT"*QQ]]LO HICISTE BIEN EN EL NIVEL"L
1000 FORI=0TO18:FORK=0TO10:MI(I,K)=0:NEXT K,I
1010 FOR W=0TON:N(W)=0:NEXT W:N=0:FL=0
1020 IFFJ=1THENRETURN
1030 PRINT"QQ]]A VER COMO LO HACES EN EL NIVEL"L+1:L=L+1
1040 FOR I=0TO1999:NEXT:PRINT"*":GOTO200
1060 REM ***** CAMBIO CARACTERES *****
1070 POKE52,48:POKE56,48:POKE56334,PEEK(56334)AND254
1080 POKE1,PEEK(1)AND251:FORI=0TO2047:POKEI+12288,PEEK(I+53248):NEXT
1090 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1:POKE53272,(PEEK(53272)AND240)+12
1100 POKE12288,24:POKE12289,152:POKE12290,145:POKE12291,255:POKE12292,56
1110 POKE12293,56:POKE12294,68:POKE12295,198
1120 RETURN
1130 REM ***** MUSICA FINAL *****
1140 POKE54291,9:FORT=0TO7:POKE54290,17:POKE54287,AF(T):POKE54286,BF(T)
1150 FOR DU=0 TO DU(T):NEXTDU:POKE54290,16:NEXTT:POKE54290,16:RETURN
1160 REM ***** INSTRUCCIONES *****
1170 PRINT"*R                    CAMPO MINADO                    ="
1180 PRINT"QUSTED SE ENCUENTRA SITUADO EN EL EXTREMO"
1190 PRINT"DE UN CAMPO DE MINAS Y DEBE ATRAVESARLO."
1200 PRINT"PARA ELLO CUENTA CON UN DETECTOR DE"
1210 PRINT"QMINAS.ESTE LE INDICA EN TODO MOMENTO LAS"
1220 PRINT"MINAS QUE HAY JUSTO A SU ALREDEDOR."
1230 PRINT"QLA ENTRADA ESTA EN LA PARTE INFERIOR"
1240 PRINT"QIZQUIERDA,Y LA SALIDA EN LA PARTE"
1250 PRINT"QSUPERIOR DERECHA MARCADA CON "CHR$(34)"■"CHR$(34)
1260 PRINT"QQQ]]]]]]]]]]]]R!BUENA SUERTE!="
1270 RETURN
1280 DATA 5530,4650,4650,4927,5530,9301,9301,7382
1290 DATA 50,100,25,25,25,100,100,500
```

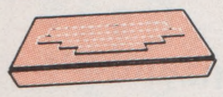
END

EL JUEGO DE

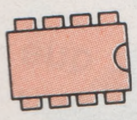


LOS BARCOS

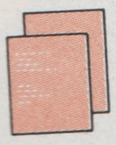
El juego de los barcos



VIC-20



VERSION BASE

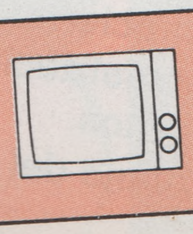


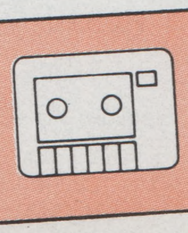
BASIC



JUEGO







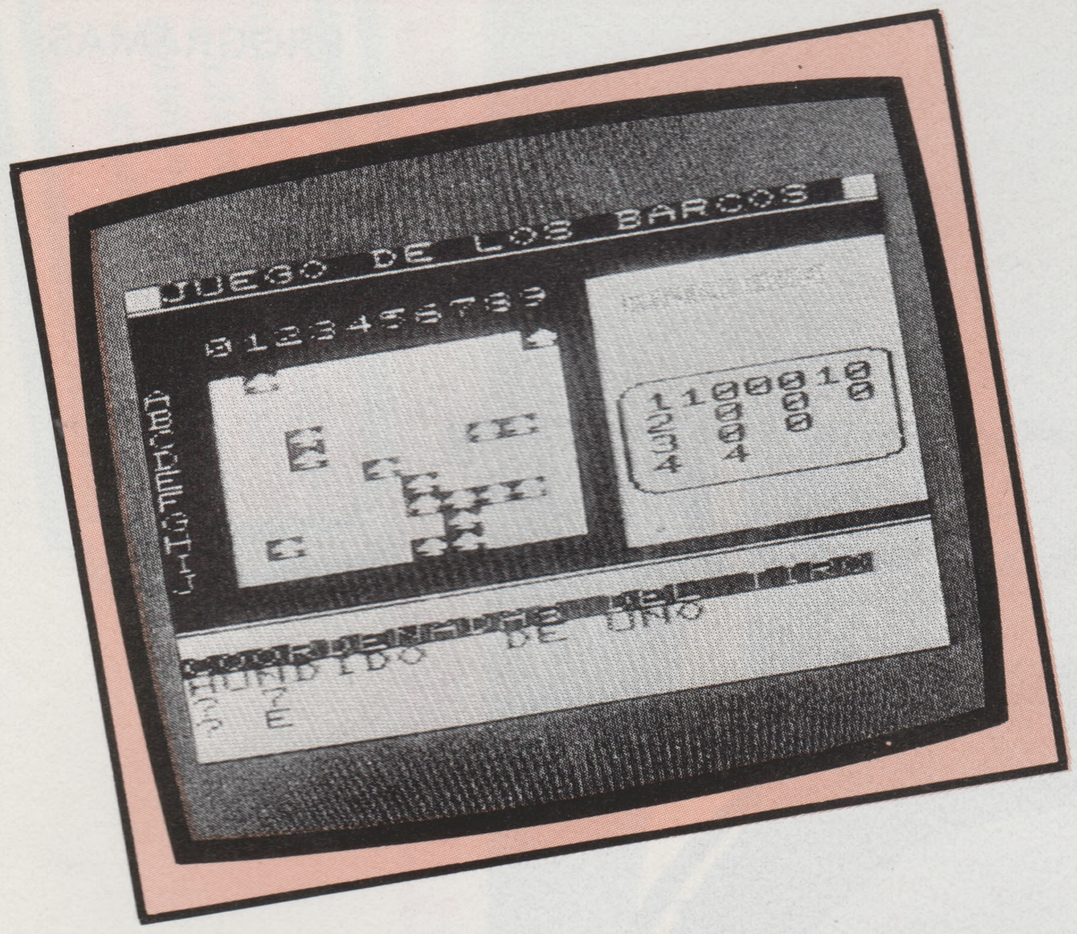
Fortunato Pérez, un quinto más del último reemplazo puesto de verde, aguantaba a pie firme y vigilante dando escolta a una robusta pieza de artillería costera. Hoy, el día se levantó con resaca, desapacible y anegado por una densa niebla que no permitía a Fortunato ver más allá de la punta del cañón. A pesar de ello, la jornada se presentaba tranquila, como una de tantas en las que nuestro amigo compartía su monólogo con el ecléctico y leal pedazo de hierro... Hasta que, de repente, sonó la alarma que anunciaba la presencia de naves enemigas. ¡La orden era disparar!... pero ¿hacia dónde?

La pose hierática de Fortunato se tornó en torbellino oceánico.

Vueltas y más vueltas alrededor del cañón...

—¡Será posible! ¡Cómo no tire al azar o barriendo la zona a diestro y siniestro!...

Y el valiente Fortunato empezó con el vals de andanadas, confiado en que los fogonazos



y explosiones le permitieran detectar sus blancos y atinar con la próxima carga.

—¡2-b!, ¡7-f!, ¡4-d!... ¡Otra vez agua!... ¡5-a!...

Los minutos volaban y las sirenas de las naves atacantes se multiplicaban en los oídos del asustado combatiente.

—¡3-c!... ¡Bouuummm! ¡Tocado!...

Disparo tras disparo y blanco tras blanco, pero otras naves parecían brotar de la niebla. Su desesperación iba en aumento ¿Qué podía hacer? ¡Seguir disparando!... pero el tiempo se le echaba encima y no lograba hundir a todos los barcos que se plantaban ante el acantilado... Y, de repente, Fortunato percibió el seco impacto de un golpe en pleno rostro...

—¡No, no!... ¡5-b! ¡Aguuuaaa!... ¿Dónde estoy? ¡Dispara, dispara que se acercan!...

Fortunato siguió gritando, mientras se incorporaba firmemente agarrado a uno de los barrotes de la cama, y al abrir los ojos se encontró sin naves a las que abatir: la niebla se había disipado por completo y tras ella sólo estaba la mirada entre incrédula y sorprendida de su amigo Paco.

—¿No te da verguenza? ¿Te parece bonito caerte en pleno ataque enemigo?...

Al ver la expresión de su vecino de litera, Fortunato terminó de despertar de su agitado sueño. Con el toque de fagina, nuestro amigo seguía aun acariciándose la dolorida barbilla y dando vueltas al bélico sueño de la noche anterior —«Al fin y al cabo, todo ha sido un sueño... pero ¿qué hago yo si en la guardia de mañana me encuentro con semejante escena...?»— Precavido donde los haya, Fortunato se dispuso a salir de dudas, y se encaminó hacia la Furrielería —«O mucho me equivoco o lo que trajo ayer el brigada era un VIC-20»—. La fácil palabra del quinto logró que el ordenador quedara a su disposición durante el próximo mes, por supuesto, después del toque de paseo.

Al empezar la tarea parecía fácil pero, poco a poco, el optimismo se convirtió en dura pelea con la máquina. El mayor problema era la escasez de memoria. Pero nuestro amigo no se arredró; después de muchas tardes frente al teclado y tras ser rebautizado con el significativo nombre de Fortunato «el ojeras», puso a punto un programa que apuraba hasta tal punto la memoria del ordenador que no quedaba en el un byte libre... ¡Todo un éxito!

Días más tarde, Fortunato recibió la visita de un comandante al que habían llegado noticias del «afortunato» acontecimiento y venía a estudiar la posibilidad de incluir el programa en el plan de preparación de futuros artilleros. Tras los saludos de rigor, el militar pidió a su subordinado que le relatara los pormenores del programa. Su relato empezó con algunos detalles acerca de la introducción del programa a través del teclado: —«... y observe que en el listado aparecen algunos caracteres subrayados que, como Ud. sabe, corresponden a los caracteres de control cuya representación en pantalla es precisamente el caracter que aparece subrayado, aunque en vídeo inverso...»—. Despejadas las dudas al respecto, apuntó que una vez cargado el programa, sólo quedaba por introducir el comando RUN para que empezara el juego. Según sus indicaciones, la pantalla empezaba mostrando un texto de presentación; al accionar cualquiera de las teclas, la cabecera desaparecía y la pantalla pasaba a estar ocupada por el mapa de combate en el que estaban distribuidos los navíos enemigos que hay que hundir. Las andanadas se disparaban introduciendo, simplemente, las coordenadas de la casilla que se quiere batir seguidas por una acción sobre la tecla RETURN. —«El propio ordenador informará del resultado de cada disparo y, además, un marcador localizado bajo la palabra puntos indicará el número de blancos realizado en cada uno de los barcos, señalando de qué tipo de barco se tra-

ESTRUCTURA DEL PROGRAMA

10-90	Cabecera de presentación.
500-600	Inicializa a cero los marcadores de aciertos.
1010-1190	Dibujo del tablero.
1200-1205	Introducción del tiro.
1210	Situación del lanzamiento en el tablero.
1220-1500	Comprobación del lanzamiento.
2000-2010	Inicialización de la matriz que representa el pedazo de mar en que transcurre el juego.
2030-2900	Situación de los barcos en la matriz.
6000-6320	Generación de mensajes tras un acierto.
6500-6700	Actualización del recuadro de puntos.
7500-7800	Dibujo del recuadro puntos.

ta en función de su tamaño»... «En efecto, mi comandante. Sobre el propio tablero de juego puede visualizar los aciertos, ya que los puntos en los que se ha realizado un blanco estarán identificados por caracteres rojos en lugar de azules».

—Bien, muy bien —replicó el comandante— un programa afortunado... ¿Qué tiene Ud. que añadir acerca de las variables que se utilizan?

—¡A sus órdenes, mi comandante!... No vaya Ud a creer que el programa es muy complicado; son muy pocas las variables que intervienen. Las más significativas son D y E que almacenan los valores efectivos del disparo y, como no, la matriz H(10) que representa la zona de mar que se considera en el juego; y, por último, tal vez deba hablarle de AU (6), AD(3) y AT(3), matrices cuya misión se limita a memorizar cierta información relativa a los blancos realizados enar en cada uno de los navíos enemigos. Las restantes variables no cumplen una función específica y se utilizan para asignaciones auxiliares o para almacenar valores intermedios.

Esbozando una sonrisa, el comandante reiteró su interés por estudiar la posibilidad de incluir el programa en el plan de instrucción y, sin añadir una palabra más, desapareció tras la puerta seguido por su displicente ordenanza.

Fortunato quedó sumido en la duda.

—¿Creéis que adoptarán el programa?... ¡Mis buenos sudores me ha costado para que ahora...!

Su inseparable Restituto, compañero de reemplazo que compartió con él las duras marchas del periodo de instrucción, se dispuso a consolarle.

—¡Animo Fortunato!... ¡Oye! y si el pograma no cuela ¿te imaginas la que podemos organizar en la «tele» entrenándonos con la maquinita?... ¿A que le hundimos la flota entera al Navarro?

¡Sí hombre... a ese del programa «La tarde»!...

Micro

LISTADO

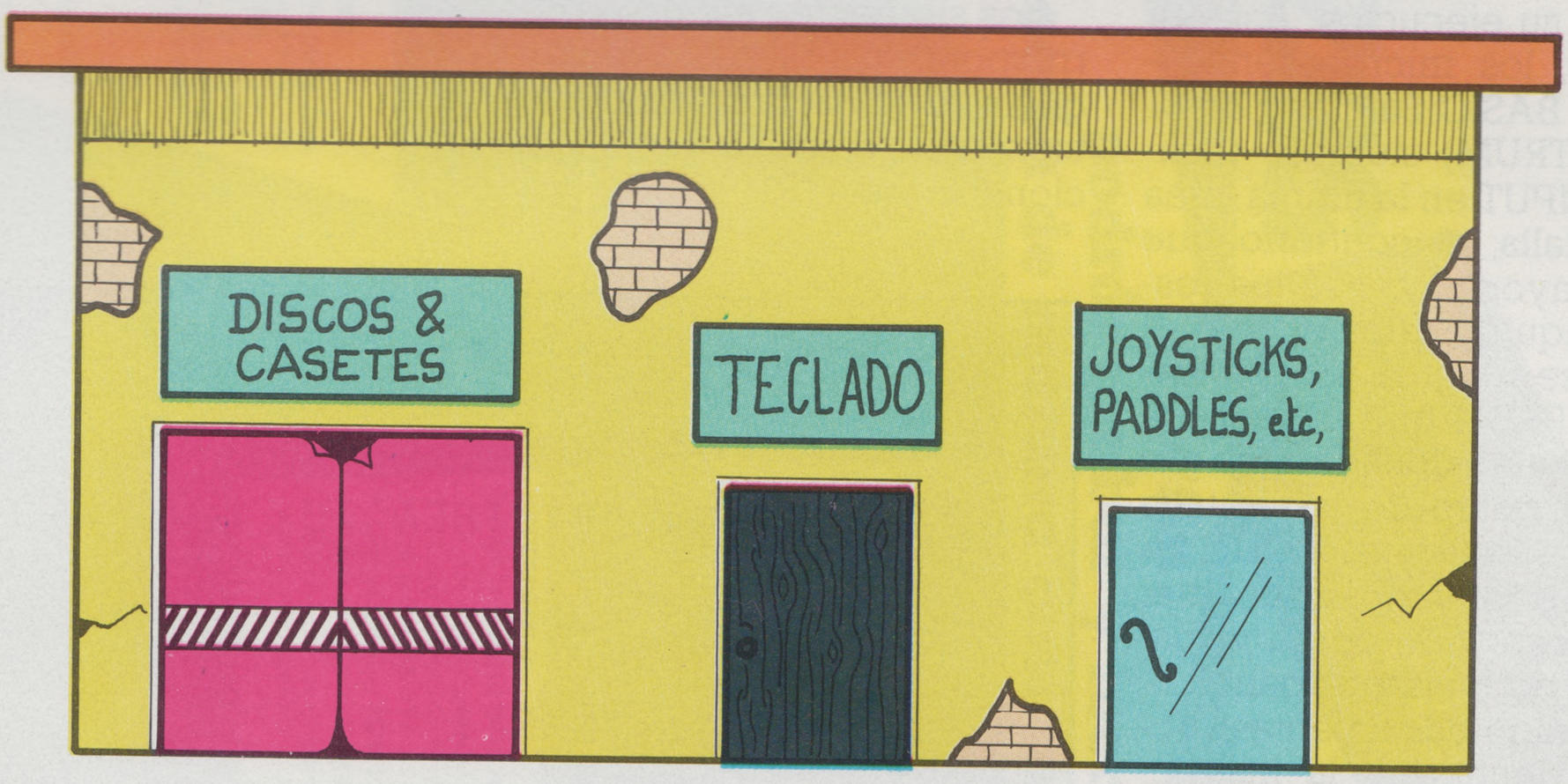
```
10 REM BARCOS
80 PRINT"■QQQQDAVID SANTAOLALLA      SANTIAGO RUIZ"
90 GETA$:IFA$=""THEN90
500 DIM AD(3):DIM AT(3)
520 FOR I=1 TO 3:AD(I)=0:AT(I)=0:NEXTI
560 AC=0
600 DIM AU(6):FOR I=1 TO 6:AU(I)=0:NEXTI
900 POKE 36879,30
990 PRINT"■
1000 GOSUB 7500
1010 PRINT"■£ RJUEGO DE LOS BARCOS _■"
1012 FOR P=1 TO 10:A=7747+P:B=175+P:POKE A,B:NEXTP
1015 FOR P=38466 TO 38742
1017 POKE P,0:NEXTP
1020 FOR M=7769TO 7780:I=242+M
1040 POKEM,96+128
1050 POKEI,96+128
1060 NEXTM
1070 FOR P=7791 TO 8022 STEP22
1080 C=P+11
1090 POKE P,96+128
1100 POKE C,96+128
1120 POKEC,96+128:NEXTP
1130 FOR C=1 TO 10:H=7790+(C-1)*22:POKEH,C+128
1140 NEXTC
1143 POKE38730,0:POKE38488,0
1145 POKE8010,160:POKE7768,160:POKE7758,160
1150 GOSUB 2000
1170 PRINT "R                                    =
1175 PRINTSPC(14)"↑RPUNTOS■-
1180 PRINT "QQQQQQQQQQR                          -
1190 PRINT"RCOORDENADAS DEL TIRO_Q
1200 INPUT S$:D=ASC(S$)-48
1205 INPUT M$:E=(ASC(M$))-65:PRINT"■■■":IF D<=10ANDE<=9ANDE>=0THEN 1210
1208 PRINT"■":GOTO 1200
1210 POKE 7792+D+(22*E),193
1220 FOR INDA=1 TO 12
1222 IF H(D,E)=INDA  THEN GOSUB 6000
1225 IF H (D,E)=0 THEN PRINT  "■AGUA              ←
1300 NEXT INDA
1500 GOTO 1200
1990 END
2000 FOR AN=1 TO 10
2003 FOR IN=1 TO 10
2005 H(AN,IN)=0
2010 NEXT IN:NEXTAN
```

```
2030 FOR I=1 TO 6
2040 A=INT(10*RND(1)):B=INT(10*RND(1))
2045 FOR K=1 TO I
2050 IF H (A,B)=K THEN GOTO 2040
2055 NEXTK
2060 H(A,B)=I
2080 NEXTI
2090 FOR I=7 TO 9
2100 A=INT(10*RND(1)):B=INT(9*RND(1))
2120 FOR K=1 TO I
2130 IF H(A,B)=K OR H(A,B+1)=K THEN GOTO 2100
2132 NEXTK
2150 H(A,B)=I
2155 V=B+1: H(A,V)=I
2160 NEXTI
2170 FOR I=10 TO 11
2210 A=INT(10*RND(1)):B=INT(8*RND(1))
2220 FOR K=1 TO I
2225 IF H(A,B)=K OR H(A,B+1)=K OR H(A,B+2)=K THEN GOTO 2210
2240 NEXTK
2250 H(A,B)=I:H(A,B+1)=I: H(A,B+2)=I
2260 NEXTI
2300 A=INT(10*RND(1)):B=INT(7*RND(1))
2310 FOR M=1 TO 11
2315 IF H(B,A)=M OR H(B+1,A)=M OR H(B+2,A)=M OR H(B+3,A)=M THEN GOTO 2300
2320 NEXTM
2330 H(B,A)=12:H(B+1,A)=12:H(B+2,A)=12: H(B+3,A)=12
2900 RETURN
6000 FOR P=1 TO 6
6010 IF H(D,E)<>P THEN GOTO 6030
6020 PRINT "■£HUNDIDO  DE UNO   ←":H(D,E)=20:AU(P)=1:GOTO 6500
6030 NEXTP
6100 FOR P=1 TO 3
6110 IF H(D,E)<>P+6 THEN GOTO 6140
6115 AD(P)=AD(P)+1
6120 IF AD(P)=2 THEN PRINT "■£HUNDIDO  DE DOS    ←":H(D,E)=20:GOTO 6500
6130 IF AD(P)=1 THEN PRINT "■πTOCADO DE DOS     ←π":H(D,E)=20:GOTO 6500
6140 NEXT P
6200 FOR P=1 TO 2
6210 IF H(D,E)<>P+9 THEN GOTO 6240
6215 AT(P)=AT(P)+1
6220 IF AT(P)=3 THEN PRINT"■£HUNDIDO DE TRES     ←":H(D,E)=20:GOTO6500
6230 IF AT(P)<>3 AND AT(P)<>0 THEN PRINT "■πTOCADO DE TRES     ←":H(D,E)=20:GOTO
 6500
6240 NEXTP
6300 IF H(D,E)=12 THEN AC=AC+1
6310 IF AC=4 THEN PRINT"■£HUNDIDO DE CUATRO ←":H(D,E)=20
6320 IF AC<>4 AND AC<>0 THEN PRINT "■πTOCADO DE CUATRO←  ":H(D,E)=20
6500 FOR J=14 TO19
6510 POKE(38576+J),0
6520 POKE(38598+J),0
6530 POKE(38620+J),0
6540 POKE(38642+J),0
6550 NEXT J
6560 POKE7916,(48+AD(1))
6570 POKE7918,(48+AD(2))
6580 POKE7920,(48+AD(3))
6590 POKE 7938,48+AT(1):POKE7960,48+AC:POKE7940,48+AT(2)
6595 POKE38512+D+(22*E),2
6600 FOR T=1 TO 6:POKE7892+T,48+AU(T):NEXTT
6650 FOR I=1 TO 6:IF AU(I)<>1 THEN RETURN
6670 NEXTI
6700 IF AD(1)=2ANDAD(2)=2ANDAD(3)=2ANDAT(1)=3ANDAT(2)=3ANDAC=4 THEN GOTO 9000
6800 RETURN
7500 PRINT"S":PRINTSPC(13)"QQQQQQQ ┌──────┐":PRINTSPC(13)"■|1000000|"
7510 PRINTSPC(13)"■|2 0 0 0|"
7515 PRINTSPC(13)"■|3 0 0  |"
7520 PRINTSPC(13)"■|4 0    |"
7525 PRINTSPC(13)"■└──────┘S"
7800 RETURN
9000 PRINT"SACABO":END
```

END

Estandarización del BASIC (3)

Entrada de datos



Todos sabemos que el cometido de un ordenador es el tratamiento de la información, tratamiento que exige tres etapas operativas: adquisición o entrada de los datos, procesado de los mismos y salida de resultados... ¿Alguién duda todavía de la importancia que tienen las instrucciones para la toma de datos dentro de un lenguaje informático?

Del mismo modo que un edificio tiene diferentes entradas, destinadas a distintos usos (entrada de garaje para los coches, entrada de servicio para las mercancías, entrada principal para los vecinos...), el ordenador también dispone de varias entradas especializadas en los distintos tipos de informaciones que pueden ingresar: entrada para datos masivos (lectura de cinta o disco), entrada para datos especiales (lápiz óptico, joysticks...) y entradas para datos a través de teclado. Esta última es la más utilizada y de la que vamos a ocuparnos en primera instancia.

El teclado: órgano fundamental de comunicación

El teclado o consola es sin duda el principal periférico de comunicación entre el usuario y la máquina. A través del mismo el ordenador recibe la mayor parte de la información, de ahí que los lenguajes informáticos —como es el caso del BASIC— cuenten entre sus recursos con varios métodos para la exploración del teclado. La instrucción de entrada de datos por excelencia es INPUT, presente en todos los dialectos BASIC con más o menos variantes. En su forma

más simple, la instrucción va seguida tan solo por una variable a la que se asigna el valor introducido. En algunos casos, entre la instrucción INPUT y la variable de asignación, se puede situar un texto o literal entrecomillado que servirá de título o de referencia para el INPUT; el efecto de este literal es semejante a la inserción de un PRINT «título» antes de la propia instrucción INPUT escrita en su formato simple.

Dada la sencillez de su formulación, parece obvio que la instrucción INPUT presente muy ligeras diferencias de uno a otro dialecto BASIC. La variantes se localizan principalmente en el comportamiento del intérprete al proceder a su ejecución. Así, por ejemplo, los microordenadores con BASIC Sinclair (ZX-81 y SPECTRUM) siempre realizan el INPUT en la última línea de pantalla; al contrario que en la mayor parte de los restantes equipos, en los que el INPUT se efectua en la posición de PRINT que corresponde. En el mencionado caso general, el punto de la pantalla donde se realiza el INPUT, ya sea con o sin texto, suele quedar referenciado por la visualización de un interrogante.

En resumidas cuentas, podemos decir que la instrucción INPUT permite introducir un grupo de caracteres cuyo final está marcado por la tecla de fin de línea (RETURN, NEW LINE, ENTER, etc.).

De uno en uno ¡por favor!

El segundo procedimiento para la entrada de datos a través del teclado es el que podemos denominar «mono-caracter». Normalmente, éste método permite la introducción de un caracter desde el teclado, caracter que puede ser nulo. Este último detalle es muy significativo puesto que, de esta forma, no se fuerza la detención del programa hasta la introducción de un dato (como ocurre en el INPUT), por el contrario, si en el momento de realizar la investigación del teclado no hay ninguna tecla pulsada, se asignará la cadena «vacía» a la variable de entrada. Son dos los tipos de lectura monocaracter habituales en los dialectos del lenguaje BASIC. El primero de ellos recurre al comando GET (BASIC Commodore, por ejemplo), mientras que la segunda variante opta por una función sin argumento: INKEY$ (Sinclair, Dragón...) o KEY$ (Oric). El comando debe estar seguido por una variable en la que se almacena el dato introducido. Sin embargo, la función permite una interrogación directa del teclado que se ejecuta asociada a una sentencia de asignación; esta última almacenará el dato de entrada en una variable que se analizará posteriormente.

Los siguientes ejemplos corresponden a rutinas que congelan la secuencia de ejecución aguardando a que se accione la tecla "R":

Commodore

```
10 GET X$
20 IF X$<>"R" THEN 10
```

Sinclair

```
10 LET X$=INKEY$
20 IF X$<>"R" THEN GOTO 10
```

Oric

```
10 LET X$=KEY$
20 IF X$<>"R" THEN 10
```

En los dos últimos casos (BASIC Sinclair y ORIC), debido a que la entrada de datos puede realizarse por medio de una función sin argumento, la rutina al efecto puede reducirse a una sola línea.

Sinclair

```
10 IF INKEY$<>"R" THEN GOTO 10
```

Oric

```
10 IF KEY$<>"R" THEN 10
```

El BASIC de Oric dispone además de un comando GET que actúa de forma distinta a como lo hace en el BASIC Commodore; diferencia lógica puesto que si se tratara simplemente de hacer un KEY$ en forma de comando, este podría realizarse de forma fácil —como ya hemos visto— asociando la propia función KEY$ a una sentencia de asignación. En efecto, el GET de Oric (al igual que el del MPF-II) tiene una particularidad: no admite la cadena vacía, por lo que la asignación sólo se realizará cuando se accione una tecla de introducción de dato; mientras, la ejecución se detendrá sin avanzar a la instrucción siguiente.

Existe otro caso un tanto especial, el del NewBrain, que exige la apertura de un fichero para el teclado antes de realizar el GET. En este caso la rutina propuesta en los anteriores ejemplos se convierte en:

```
10 OPEN#2,6
20 GET#2,X$:IF X$<>"R" THEN 20
30 CLOSE#2
```

Con Atari nos aguarda aún otra sorpresa más. La lectura se realiza por medio de un PEEK, aplicado directamente al buffer del teclado (dirección 764); si el contenido de esta posición de memoria es 255, no existe ninguna tecla pulsada.

Ficheros secuenciales

Los microordenadores suelen utilizar casetes y discos como medios de soporte para el almacenamiento masivo de los datos. Estos suelen memorizarse en forma de ficheros o archivos secuenciales.

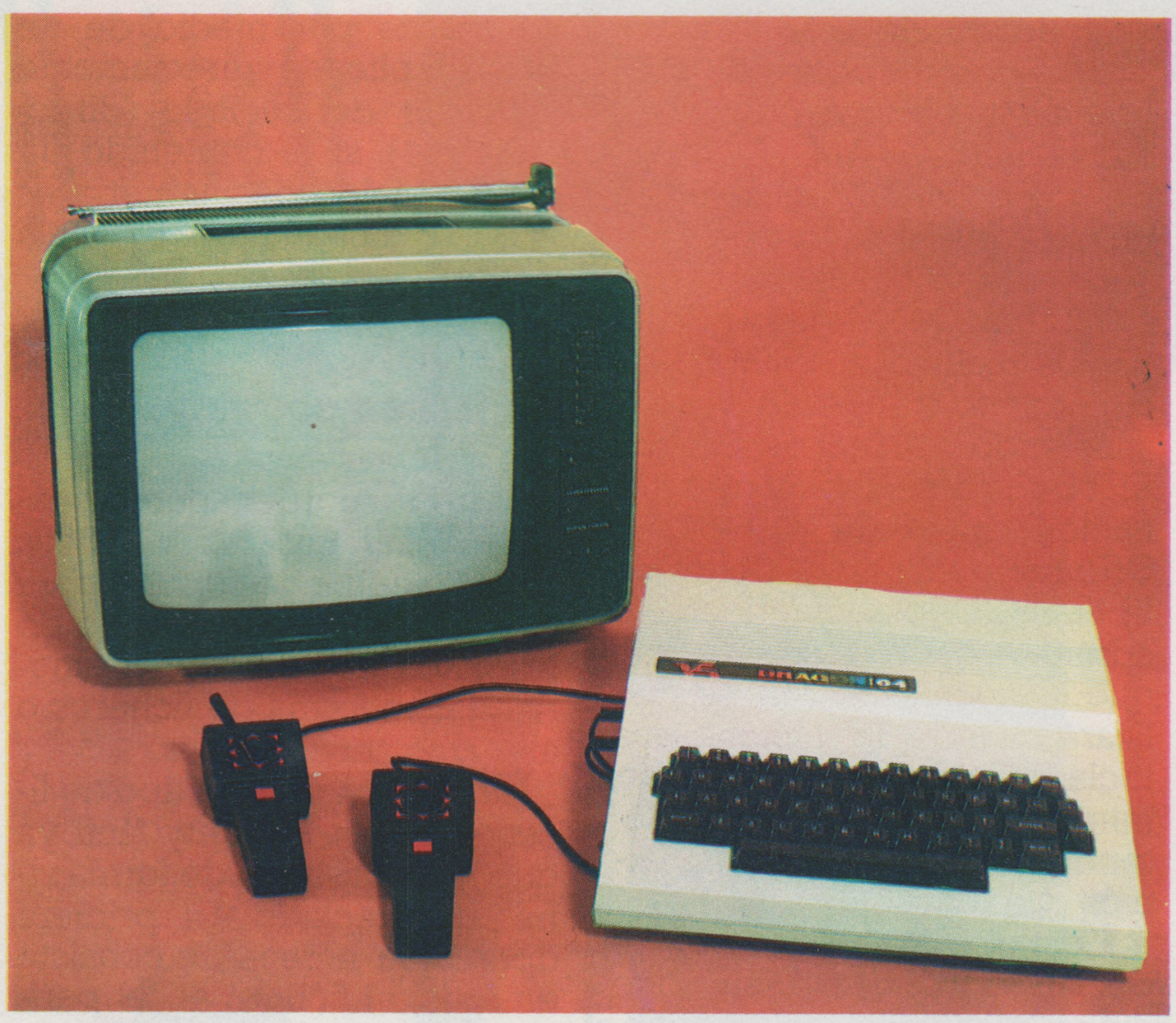
La gestión de este tipo de ficheros suele ser muy fácil ya que se comportan de forma similar al teclado. La primera operación a realizar consiste en abrir el fichero en el periférico deseado (casete o disco) y, a continuación, se procede a la lectura, tras la cual es conveniente cerrar el fichero... ¡para que no se escape el gato, ya saben!

La lectura se realiza utilizando las mismas intrucciones que se emplearían en el caso del teclado, aunque indicando el número de fichero del que se lee. Así, por ejemplo, INPUT efectuará una lectura del fichero hasta localizar un carácter de fin de línea; mientras

TABLA DE COMPATIBILIZACION

Símbolo	ORIC	COMMODORE	SINCLAIR	NEW BRAIN	DRAGON	ATARI	Descripción
GET	x(1)	x		x		x(2)	Toma un carácter del teclado, sin detener la ejecución del programa.
INKEY$			x		x		Como la anterior.
KEY$	x						Como la anterior.
PEEK (764)						x	Como la anterior.
INPUT	x	x	x	x	x	x	Toma un dato del teclado, hasta encontrar un carácter de fin de línea.
GET#		x	x		x		Lectura de un carácter de fichero.
INKEY$#			x		x		Como la anterior.
INPUT#		x	x	x	x	x	Lectura de un dato de un fichero hasta el próximo carácter de fin de línea.
JOYSTK (x)					x		Lectura de la posición del joystick.
STICK (x)						x	Lectura de la posición del joystick x.

(1) En el Oric produce la detención del programa hasta que se pulsa una tecla.
(2) Carece del sentido de lectura del teclado y se convierte en una lectura de códigos de la pantalla (color o carácter).



Algunos dialectos BASIC incorporan un grupo de instrucciones especializadas en la captación de datos procedentes de periféricos especiales. Este es el caso, por ejemplo, del intérprete BASIC del Dragón-32 que dispone de una función orientada a la lectura del estado de los joystickes conectados al equipo.

que la lectura caracter a caracter se realizará con el auxilio de la correspondiente instrucción)GET, INKEY$...) ya sea o no función.

En definitiva, los comandos descritos en los párrafos anteriores se transforman, según el periférico afectado por la operación de lectura, en: INPUT (número de canal) y, para lectura monocaracter, en GET (número de canal) o INKEY$ (número de canal).

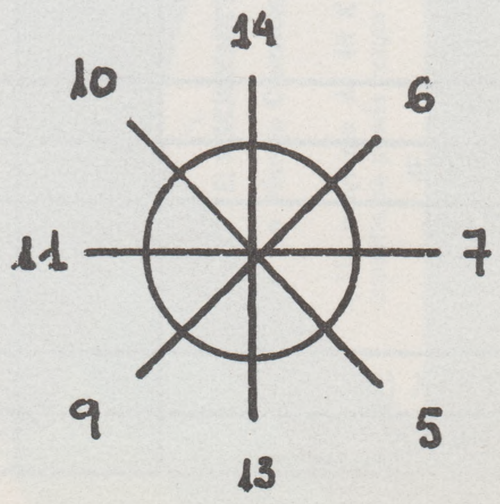
Periféricos especiales

Aún nos queda por hablar de ciertos dispositivos periféricos, algo menos frecuentes, a los que nos vamos a referir como especiales: lápices ópticos, joysticks, lectores de código de barras... La entrada de datos suele adquirir unas características específicas y particulares para cada dispositivo. Lo más común es que se recurra directamente a la máquina, mediante consultas a direcciones de memoria RAM (PEEKs) o del port de usuario (INs), en lugar de utilizar instrucciones BASIC al efecto. No obstante, hay equipos —como es el caso del Atari o Dragón— que disponen de comandos específicos para el manejo de los joysticks. Por ejemplo, el intérprete BASIC del Dragón incorpora una función BASIC que informa cómodamente del estado del joystick: JOYSTK (x). Donde x puede adoptar los valores 1, 2, 3 ó 4.

x=0: Consulta de la posición horizontal del Joystick izquierdo.
x=1: Consulta de la posición vertical del Joystick izquierdo.
x=2: Consulta de la posición horizontal del Joystick derecho.
x=3: Consulta de la posición vertical del Joystick derecho.

Al ejecutar la función, se obtiene como respuesta un número comprendido entre 0 y 63, dato a partir del que puede escalonarse la resolución de los saltos del joystick según desee el usuario. Debido a la rapidez de manejo de este tipo de periférico y, por ende, a las célebres alteraciones que sufre el valor de la función, es muy recomendable utilizar la mencionada función asociada a sentencias de asignación que almacenen los valores de posición para su posterior análisis (por ejemplo, A=JOYSTICK (0).

El intérprete BASIC de Atari también dispone de una función especializada en la lectura de los joysticks: STICK (x). El argumento que figura entre paréntesis define el número de palanca de juego objeto de examen (joystick 0 a 3, de izquierda a derecha). La respuesta a la función es un código que indica el sentido de desplazamiento de la palanca del joystick escrutado; adoptando como referencia los puntos cardinales, la correspondencia es:



Secoinsa amplía su gama de ordenadores

Por sus hechos los conoceréis

CON SECOINSA, LA RENTABILIDAD ES UN HECHO Todos los ordenadores Secoinsa, con la mejor relación precio/potencia, han sido pensados para rendir al máximo y capitalizar la inversión en Informática.

CON SECOINSA, LA EFICACIA ES UN HECHO Eficacia es fuerza y poder de resolución. Secoinsa ofrece el hardware y el software más adecuado a las particularidades de cada empresa.

CON SECOINSA, LOS RESULTADOS SON UN HECHO Son un hecho decisivo, confirmado por la experiencia informática en miles de empresas españolas que tienen ordenadores Secoinsa.

CON SECOINSA, SUS BENEFICIOS SON UN HECHO Beneficios a largo plazo, por rentabilidad, y a corto plazo, por sencillez de adaptación a todos los procesos.

LA AMPLIACION DE LA GAMA SECOINSA YA ES UN HECHO Diversidad de soluciones a todos los niveles. Alta tecnología, oferta completa y la mejor relación precio/potencia del mercado.

Pídanos información, pregunte, y juzgue usted mismo.

secoinsa

La Empresa Nacional de Informática.

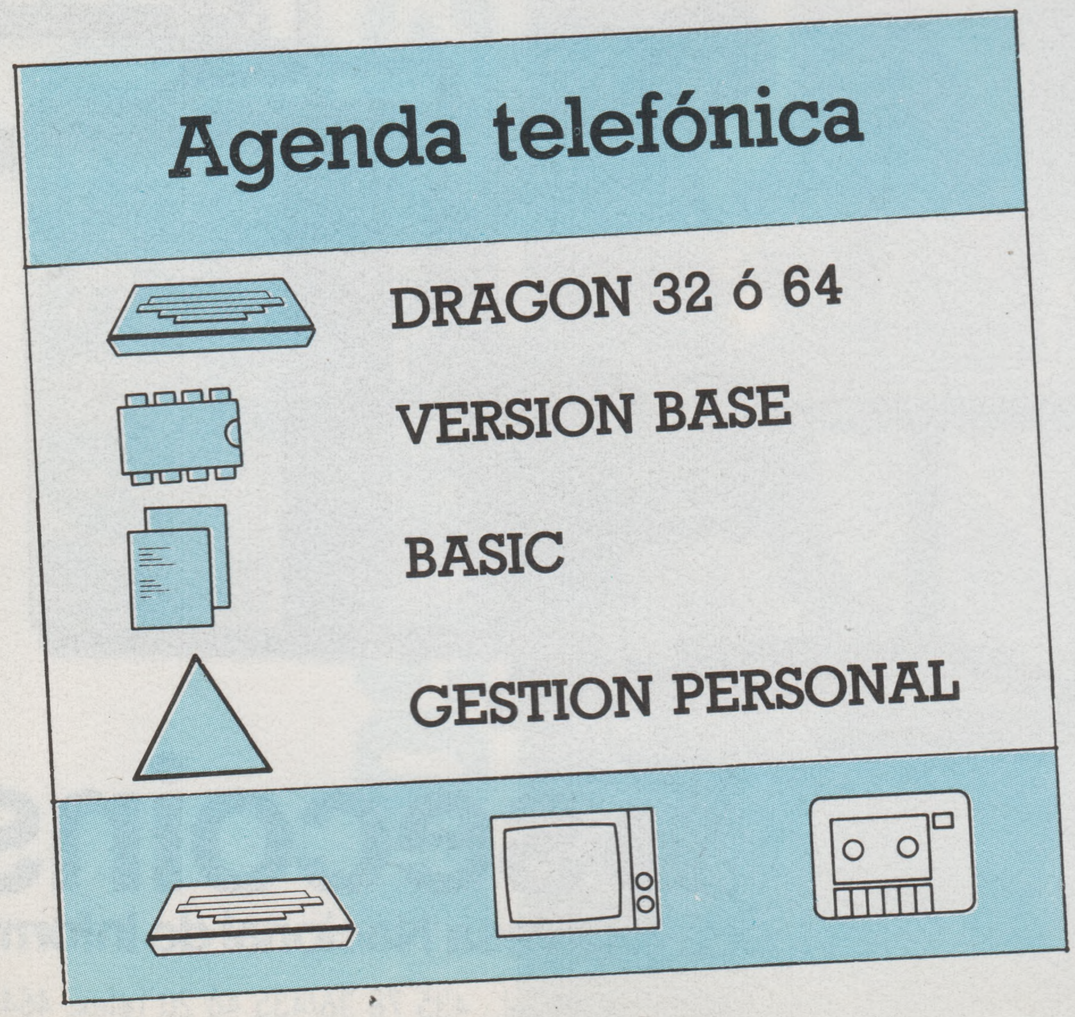
Almagro, 40 Tel.: 435 78 36/435 48 20 Telex: 46454 SECI E.
Apartado de Correos 10.238 Madrid-4



Una completa y automatizada guía telefónica para el DRAGON-32, con tres apartados: el primero dedicado a los posibles teléfonos de interés, otro a los prefijos nacionales (no todos) y un tercer apartado destinado a los teléfonos de uso particular.

AGENDA TELEFONICA

La explotación del programa resulta de gran comodidad y sencillez debido a que su desarrollo está organizado a base de menús. El primero de ellos es el menú general del programa, que da paso a las tres opciones básicas.



1. Teléfonos de interés.
2. Prefijos nacionales.
3. Teléfonos particulares.

La opción tercera da entrada al submenú de «Teléfonos particulares», cuyo repertorio de opciones permiten, además de la lectura de un teléfono particular, la introducción de nuevos teléfonos de archivar así como la baja de los teléfonos que el usuario desee suprimir.

El programa se encarga de presentar en la pantalla las indicaciones pertinentes para que el usuario pueda realizar sin dificultad las operaciones necesarias.

Estructura del programa

La zona funcional del programa que presentamos empieza con el bloque comprendido entre las líneas 110 a la 170, en las que se lee el fichero «Teléfono». El bloque siguiente (líneas 180 a la 260) muestra el menú principal y deriva la ejecución hacia la opción seleccionada. De la línea 270 a la 360 se procede a la representación y ejecución de la opción primera

«Teléfonos de interés»; de la 370 a la 450 se desarrolla la opción 2 «Prefijos nacionales» mientras que el submenú asociado a la opción 3, «Teléfonos particulares», es procesado por las líneas 460 a la 530. Este submenú dispone, a su vez, de tres opciones que son ejecutadas por las siguientes zonas del programa. Las líneas 540 y 630 se ocupan de buscar el registro deseado (opción 1: lectura de teléfonos particulares). Entre la 640 y la 830 se procede a la creación de nuevos registros y a su posterior grabación en el nuevo archivo «Teléfono» actualizado (opción 2). El bloque comprendido entre las líneas 840 y 940 se encarga de cancelar los registros que solicite el usuario y, por último, las líneas 1.000 a la 1.020 son las que tienen encomendada la creación del archivo inicial «Teléfono». Esta última zona sólo se ejecutará la primera vez que utilicemos el archivo en cuestión; en lo sucesivo la mencionada zona no intervendrá en absoluto dentro del programa.

Utilizando la agenda telefónica

Antes de utilizar por primera vez el programa de «Agenda telefónica», es preciso inicializar el archivo «Teléfono»; para ello basta con lanzar la orden de ejecución RUN 1.000.

Una vez inicializado el citado archivo puede ya ordenar la ejecución del programa en la forma habitual, introduciendo el comando RUN.

El programa es capaz de operar con una tabla de 100 nombres, 100 números telefónicos y 100 domicilios; amplitud que hemos considerado suficiente para satisfacer las exigencias de una agenda telefónica doméstica o personal. En cualquier caso, si desea ampliar el volumen de datos de su agenda telefónica, le bastará con modificar la línea 10 del programa; más concretamente, los parámetros de definición del número máximo de nombres, teléfonos y domicilios procesables. En el listado adjunto, la definición de los referidos parámetros es NO$(100), TE$(100) y DO$(100), con lo que se establece la capacidad de la agenda en 100 «fichas telefónicas».

Consejos finales

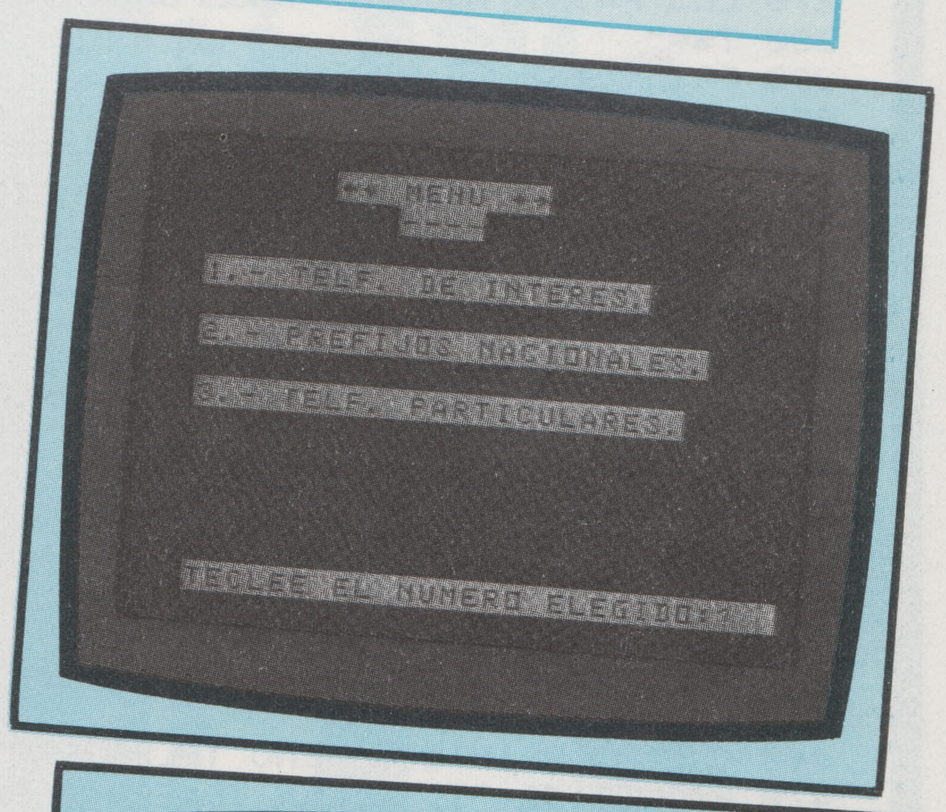
En la línea 280 nos encontramos con la variable T$ que contiene 8 caracteres en blanco, mientras que en la línea 380 la asignación otorgada a la misma variable es de 3 caracteres blancos.

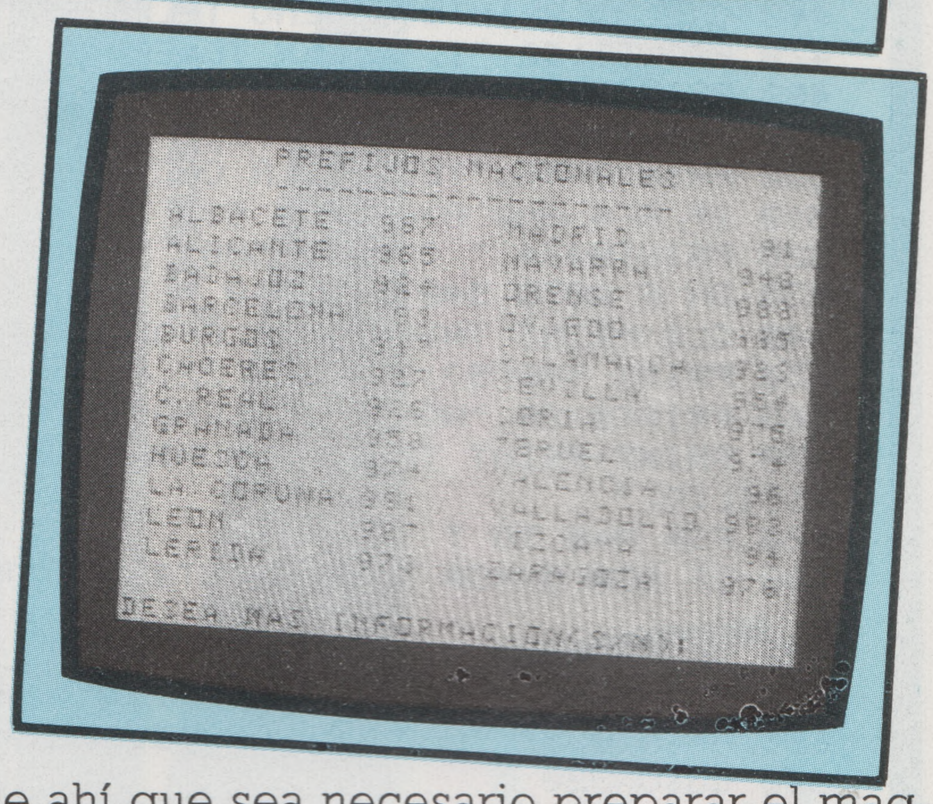
Cuando se disponga a grabar un nuevo registro o a dar de baja algún registro que desee suprimir, es recomendable que grave en la cinta una referencia verbal (una palabra o, simplemente, la fecha en la que realiza la modificación) antes de seleccionar la opción 2 («Crear teléfono nuevo») ó 3 («Dar de baja») del submenú de «Teléfonos particulares». Esta referencia, grabada en la cinta en modo audio, le permitirá identificar, sin lugar a error, el archivo de teléfonos más actualizado y que incorpora las últimas modificaciones realizadas.

Recuerde que su agenda reside en una cinta; de ahí que sea necesario preparar el magnetófono a casetes para que actúe en modo lectura o reproducción (tecla PLAY accionada), antes de ejecutar el programa por medio del comando RUN. Por supuesto, cuando el programa le informe a través de la pantalla que va a realizar una grabación (mensaje: «Prepárese para grabar»), debe preparar el magnetófono al efecto, accionando simultáneamente las teclas PLAY y RECORD.

TABLA DE VARIABLES

A	Variable numérica para el INPUT de las líneas 230 y 500.
B$	Variable utilizada para la lectura de los datos impares del DATA localizado en la línea 970.
C$	Variable utilizada pra la lectura de los datos pares del DATA localizado en la línea 970.
D$	Variable de INPUT utilizada en las líneas 330, 700, 740, 760 y 910.
E$	Variable para la lectura de los datos impares del DATA de las líneas 980 y 990.
F$	Variable para la lectura de los datos pares del DATA de las líneas 980 y 990
G$	Variable INPUT para almacenar el nombre que se quiere localizar.
NO$	Variable perteneciente al nombre.
TE$	Variable perteneciente al teléfono.
DO$	Variable perteneciente al domicilio.
T$	Variable auxiliar con caracteres blancos.
T	Contiene el número de registros que hay en el archivo de teléfonos particulares.





LISTADO



```
1 REM *AGENDA TELEFONICA*
2 REM *I.MORAGREGA V.*
10 DIM B$(12),C$(12),E$(24),F$(24),NO$(100),TE$(100),DO$(100)
20 CLS(3)
30 FOR Z=1 TO 12:READ B$(Z),C$(Z):NEXT Z
40 FOR Z=1 TO 24:READ E$(Z),F$(Z):NEXT Z
50 FOR X=19 TO 46
60 SET(X,9,8):SET(X,22,8):NEXT X
70 FOR Y=10 TO 21
80 SET(19,Y,8):SET(46,Y,8):NEXT Y
90 PRINT@205,"AGENDA";
100 PRINT@299,"TELEFONICA";
110 OPEN"I",#-1,"TELEFONO"
120 INPUT#-1,T
130 Z=0
140 IF EOF(-1) THEN 170
150 INPUT#-1,NO$(Z),TE$(Z),DO$(Z)
160 Z=Z+1:GOTO 140
170 CLOSE#-1
180 CLS(3):PRINT@43,"** MENU **";
190 PRINT@78,"----";
200 PRINT@131,"1.-";TAB(7);"TELF. DE INTERES.";
210 PRINT@195,"2.-";TAB(7);"PREFIJOS NACIONALES.";
220 PRINT@259,"3.-";TAB(7);"TELF. PARTICULARES.";
230 PRINT@450,"TECLEE EL NUMERO ELEGIDO:";: INPUT A
240 IF A<4 AND A>0 THEN 260
250 PRINT@476," ":GOTO 230
260 ON A GOTO 270,370,460
270 CLS:PRINT TAB(6);"TELEFONOS DE INTERES"
280 T$="          "
290 PRINT TAB(6);STRING$(20,45)
300 FOR Z=1 TO 12
310 PRINT TAB(4);B$(Z);TAB(19);LEFT$(T$,8-LEN(C$(Z)));C$(Z):NEXT Z
320 PRINT@480,"DESEA MAS INFORMACION(S/N):";
330 LINE INPUT D$
340 IF D$="S" THEN 180
350 IF D$="N" THEN 950
360 PRINT@475," ";:GOTO 330
370 CLS:PRINT TAB(6);"PREFIJOS NACIONALES":X=65:Y=11
380 T$="    "
390 PRINT TAB(6);STRING$(19,45)
400 FOR Z=1 TO 24
410 PRINT@X,E$(Z);TAB(Y);LEFT$(T$,3-LEN(F$(Z)));F$(Z)
420 X=X+32
430 IF Z=12 THEN X=81:Y=28
440 NEXT Z
450 GOTO 320
460 CLS(3):PRINT@7,"MENU TELEF.PARTIC.";:PRINT@39,STRING$(18,45);
470 PRINT@162,"1.- LEER TELEFONO PARTICULAR.";
480 PRINT@226,"2.- CREAR TELEFONO NUEVO.";
490 PRINT@290,"3.- DAR DE BAJA.";
500 PRINT@450,"TECLEE LA OPCION DESEADA";:INPUT A
510 IF A<4 AND A>0 THEN 530
520 PRINT@474," ":GOTO 500
530 ON A GOTO 540,640,840
540 CLS:PRINT TAB(5);"TELEFONOS PARTICULARES":W=208
550 PRINT TAB(5);STRING$(22,45)
560 PRINT@128,"NOMBRE DESEADO:";:INPUT G$
570 PRINT:PRINT TAB(3);"NOMBRE.....":PRINT TAB(3);"TELEFONO...":PRINT TAB(3);"DO
MICILIO."
580 FOR M=0 TO T
590 IF NO$(M)=G$ THEN PRINT@208,NO$(M):PRINT@240,TE$(M):PRINT@272,DO$(M):GOTO 32
0
600 NEXT M
610 FOR Z=1 TO 50:PRINT@327,STRING$(15,32)
620 PRINT@327,"NOMBRE NO REGISTRADO":NEXT Z
630 GOTO 320
640 CLS:PRINT TAB(9);"TELEFONOS NUEVOS":PRINT TAB(9);STRING$(16,45)
650 T=T+1
660 PRINT@97,"C.NOMBRE....";:INPUT NO$(T)
670 PRINT@161,"C.TELEFONO..";:INPUT TE$(T)
680 PRINT@225,"C.DOMICILIO.";:INPUT DO$(T)
690 PRINT@353,"SON LOS DATOS VALIDOS(S/N):";
700 LINE INPUT D$
710 IF D$="S" THEN 740
```

```
720 IF D$="N" THEN T=T-1:GOTO 640
730 GOTO 690
740 PRINT@417,"DESEA OTRO (S/N):";:INPUT D$
750 IF D$="S" THEN 640
760 CLS(3):PRINT@134,"PREPARESE PARA GRABAR";:PRINT@224,"CUANDO ESTE LISTO PULSE
 ENTER";:INPUT D$
770 OPEN"O",#-1,"TELEFONO"
780 PRINT#-1,T
790 FOR Z=0 TO T
800 PRINT#-1,NO$(Z),TE$(Z),DO$(Z)
810 NEXT Z
820 CLOSE#-1
830 GOTO 320
840 CLS:PRINT TAB(10);"DAR DE BAJA":PRINT TAB(10);STRING$(11,45)
850 PRINT@128,"NOMBRE A DAR DE BAJA:";:INPUT G$
860 FOR M=0 TO T:IF NO$(M)=G$ THEN 880
870 NEXT M
880 FOR Z=M+1 TO T:NO$(Z-1)=NO$(Z):TE$(Z-1)=TE$(Z):DO$(Z-1)=DO$(Z):NEXT Z
890 T=T-1
900 PRINT@417,"DESEA ALGUNA BAJA MAS(S/N):";
910 LINE INPUT D$
920 IF D$="S" THEN 840
930 IF D$="N" THEN 760
940 GOTO 900
950 CLS(4):PRINT@171,"NO OLVIDE";
960 PRINT@260,"DESCONECTAR EL ORDENADOR";:PRINT@480," ":END
970 DATA POLICIA,091,POL.MUNICIAL,092,GUARDIA CIVIL,2331100,BOMBEROS,2323232,URG
.MEDICA,2616199,AMBULANCIAS,2523264,RENFE,7333000,AEROPUERTO,2054372,RADIO TAXI,
2478200,INFORM.TELF.,003,AVERIAS TELF.,002,INFORM.GENERAL,098
980 DATA ALBACETE,967,ALICANTE,965,BADAJOZ,924,BARCELONA,93,BURGOS,947,CACERES,9
27,C.REAL,926,GRANADA,958,HUESCA,974,LA CORUNA,981,LEON,987,LERIDA,973
990 DATA MADRID,91,NAVARRA,948,ORENSE,988,OVIEDO,985,SALAMANCA,923,SEVILLA,954,S
ORIA,975,TERUEL,974,VALENCIA,96,VALLADOLID,983,VIZCAYA,94,ZARAGOZA,976
1000 OPEN"O",#-1,"TELEFONO"
1010 PRINT#-1,-1
1020 CLOSE#-1
```

END

EL PROGRAMA DEL LECTOR
EL PROGRAMA DEL LECTOR
EL PROGRAMA DEL LECTOR

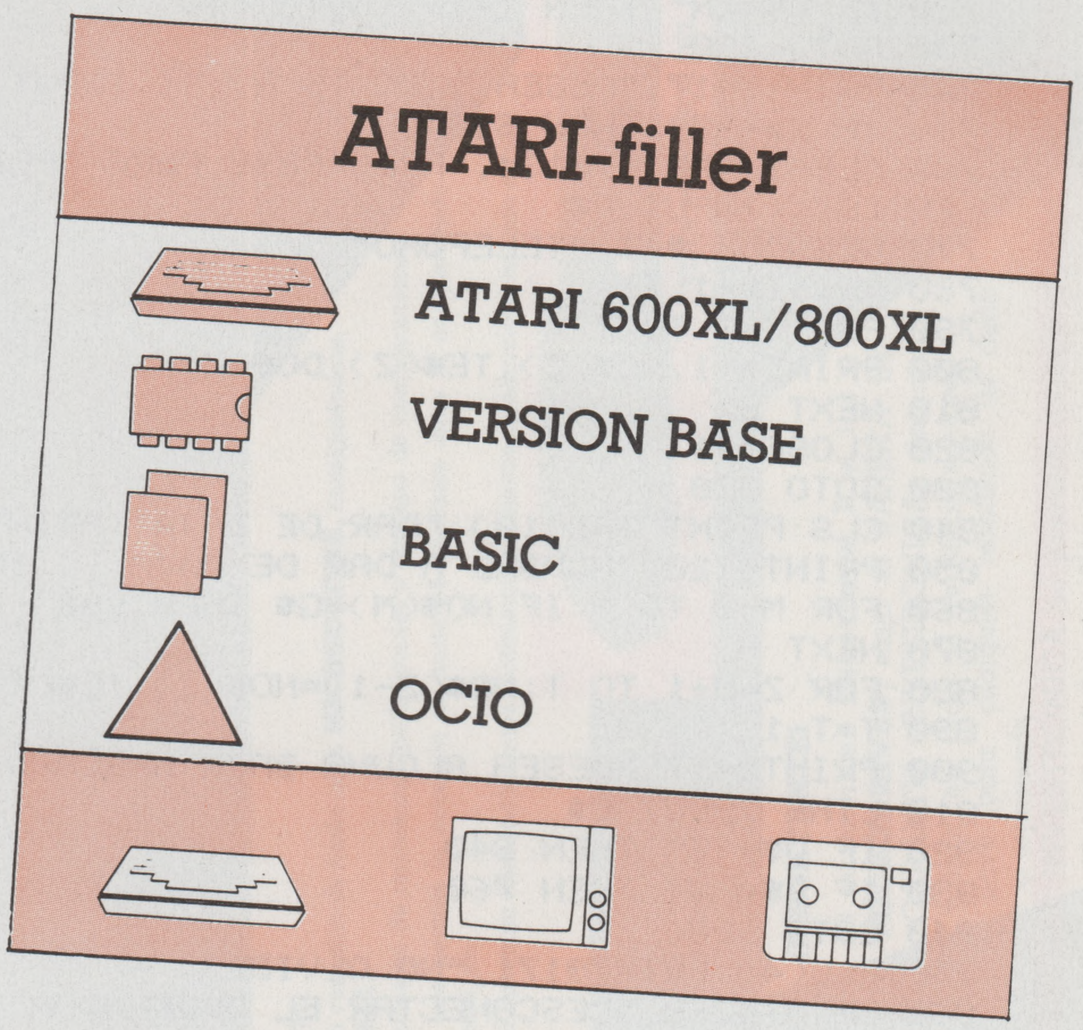
EL PROGRAMA DEL LECTOR

EL PROGRAMA DEL LECTOR
EL PROGRAMA DEL LECTOR
EL PROGRAMA DEL LECTOR

El fallo del jurado se hará público
en el próximo número de TU MICRO:
Nº 5, septiembre-1984.

UN PRIMER PREMIO DE 60.000 PESETAS EN MATERIAL INFORMATICO

ATARI filler



Dibujando con el joystick sobre la pantalla

Los aficionados al análisis de las características de los modernos microordenadores, conocen las notables posibilidades gráficas de los nuevos modelos de ATARI, 600 XL y 800 XL; sobre todo en lo relativo a su resolución máxima en modo gráfico.

El programa que vamos a proponerle en este artículo echa mano de estas posibilidades gráficas y ofrece al usuario una útil herramienta para el dibujo sobre la pantalla del TV. Utilizando el joystick para mover el pincel, el usuario puede trazar en la pantalla cualquier dibujo, e incluso rellenar de color las superficies cerradas por líneas. Un simple menú le permite elegir en cualquier momento el color adecuado para trazar las líneas del dibujo o dar color a la zona seleccionada.

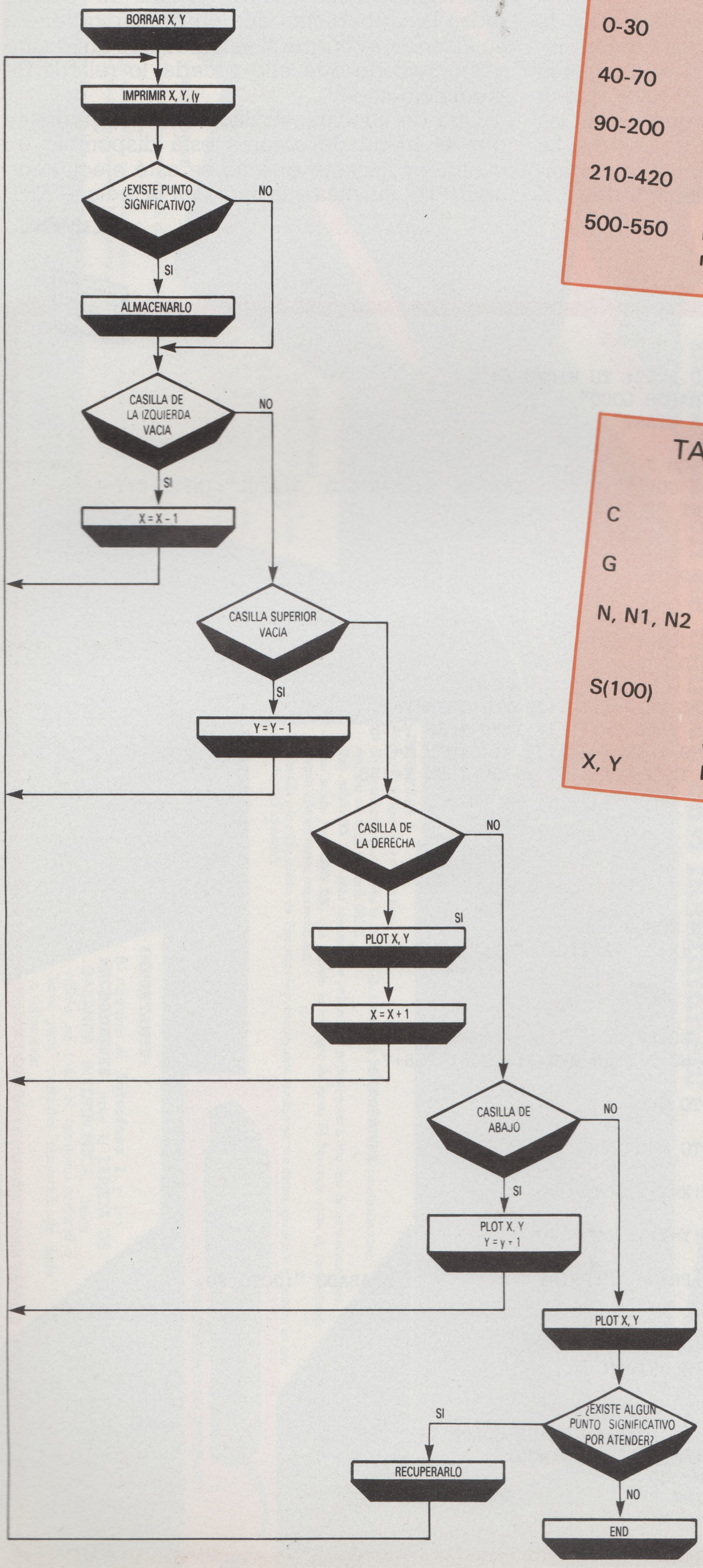
Al ejecutar el programa, este plantea como primer interrogante «¿Cuál es el color con el que quieres dibujar?». A continuación, y por medio del joystick conectado al «port 1», puede ya dar rienda suelta a su arte y crear el dibujo en la pantalla. Cuando se haya encerrado una superficie dentro de la línea de dibujo, puede pedir al equipo que la rellene de color; para ello, solo hay que accionar el botón de disparo del joystick y, como respuesta, el ordenador permitirá elegir el color que se estime oportuno.

¿Cuántos colores están disponibles para rellenar superficies?... Solo tres y por una razón que sin duda conocerán los usuarios de ATARI: en modo «alta resolución» únicamente están disponibles cuatro colores, el negro que utilizamos como fondo y los tres restantes que llenan los botes en los que «mojar» el pincel.

¿Cómo detecta el programa si se encuentra dentro de una superficie cerrada... y cuándo detecta que ha salido de ella? El diagrama de flujo adjunto ilustra la secuencia de operaciones que rige la actuación del programa.

El mayor problema radica en los «puntos singulares» que ha de recordar el ordenador para no dejarse atrás superficies sin rellenar de color... pero, vayamos al programa. La primera zona del algoritmo para dar color a las superficies cerradas (líneas 215 a la 420), se encarga de comprobar si la pluma de dibujo está situada en un punto significativo o «singular» y, en caso afirmativo, de almacenarlo. A continuación, realiza desplazamientos hacia la izquierda, tantos como sean necesarios para llegar a colocarse en una posición inmediata a otra ocupada. Luego, desplaza de nuevo el punto de dibujo, esta vez hacia arriba, hasta llegar a la posición anterior a la primera ocupada. Acto seguido, si el pixel situado a la derecha está «vacío de color», lo rellenará, y seguirá dando

ORDINOGRAMA DEL ALGORITMO CENTRAL DEL RELLENADOR LOCO



ESTRUCTURA DEL PROGRAMA

0-30	Prolegómenos: selección de color, cartel de entrada, etc.
40-70	Dibujo del cuadrado y del punto.
90-200	Chequeo del joystick y trazado del dibujo en la pantalla.
210-420	Rutina para rellenar superficies.
500-550	Rutina de presentación del menú.

TABLA DE VARIABLES

C	Almacena el color con el que se está trabajando.
G	Variable intermedia que da el estado del joystick.
N, N1, N2	Variables intermedias que almacenan el contenido de determinados puntos.
S(100)	Matriz que almacena de puntos singulares a medida que se van detectando.
X, Y	Posición del cursor.

Diagrama de flujo ilustrativo de la secuencia de operaciones que rige la actuación del programa "ATARI-filler".

color hasta que llegue al tope por la derecha. A lo largo del camino recorrido el programa irá memorizando los puntos singulares que encuentre.

A estas alturas del comentario, no estaría de más explicar en qué consisten los puntos singulares. Estos puntos se caracterizan porque el pixel de la derecha está vacio, al igual que el de abajo, si bien no lo están los que inician las diagonales hacia abajo y hacia la derecha. La razón de que estos puntos sean tan importantes es que, dada la naturaleza del algoritmo, si no se tuvieran en cuenta quedarían vacías de color las superficies localizadas debajo de ellos... ¡cosa que no podemos consentir! Así que cuando el ordenador termina de rellenar todas las superficies que en principio tiene a su alcance, averigua si se ha dejado atrás algo y en caso de que ello suceda, lo rellena inmediatamente.

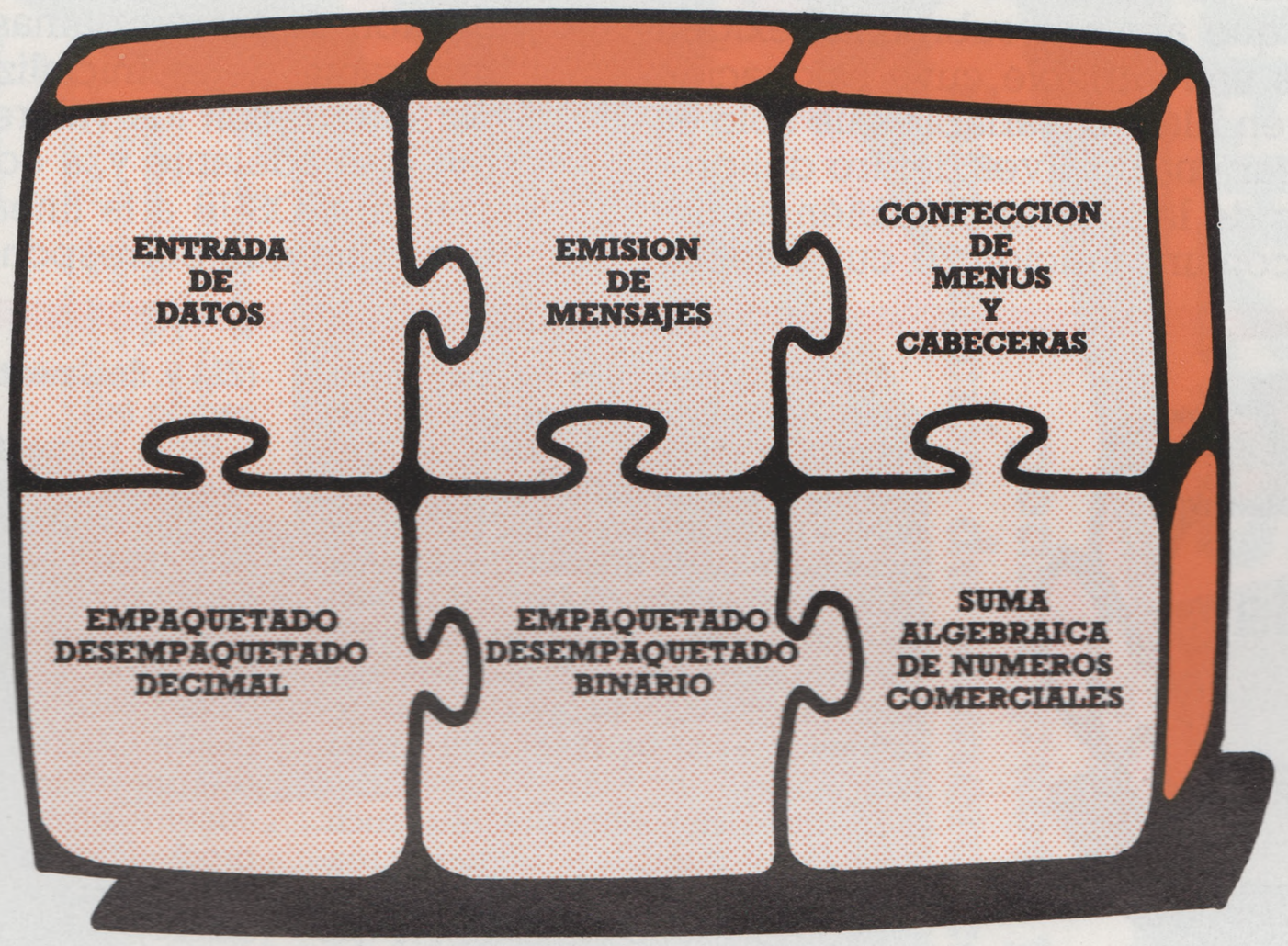
Otra de las características del programa es que el menú de colores está disponible en cualquier instante que no se esté ejecutando un INPUT, sin más que accionar la tecla START.

LISTADO

```
1 GRAPHICS 0:REM SANTIAGO RUIZ: TU MICRO 84
2 POSITION 10,8:? "RELLENADOR LOCO"
3 POSITION 12,18:? "PULSA START"
5 IF PEEK(53279)<>6 THEN 5
10 DIM S(100):X=80:GRAPHICS 7
15 POSITION 2,20:? "ELIGE COLOR":? " 1:NARANJA  2:AMARILLO  3:AZUL":INPUT C:Y=40
20 PRINT "":PRINT "":PRINT ""
30 COLOR C:PLOT X,Y
40 PLOT 0,0
50 DRAWTO 159,0:DRAWTO 159,79
70 DRAWTO 0,79:DRAWTO 0,0
90 G=STICK(0)
95 IF PEEK(53279)=6 THEN GOSUB 500
100 IF STRIG(0)=0 THEN 210
110 IF G=15 THEN 90
120 IF G=14 OR G=10 OR G=6 THEN Y=Y-1:IF Y<1 THEN Y=1
130 IF G=13 OR G=9 OR G=5 THEN Y=Y+1:IF Y>78 THEN Y=78
140 IF G=11 OR G=10 OR G=9 THEN X=X-1:IF X<1 THEN X=1
150 IF G=7 OR G=6 OR G=5 THEN X=X+1:IF X>158 THEN X=158
200 PLOT X,Y:GOTO 90
210 REM
215 POSITION 2,18:INPUT C
220 G=0:COLOR 0
230 PLOT X,Y:COLOR C
240 POSITION 2,20
245 IF PEEK(53279)=6 THEN GOSUB 500
250 PRINT "  HORIZONTAL=";X;"  VERTICAL=";Y;"  G=";G
260 LOCATE X+1,Y+1,N
270 LOCATE X,Y+1,N1
280 LOCATE X+1,Y,N2
285 REM SAVE POINT IF NECESSARY
290 IF N<>0 AND N1=0 AND N2=0 THEN G=G+1:S(G)=X*256+Y
300 LOCATE X-1,Y,N
310 IF N=0 THEN X=X-1:GOTO 240
320 LOCATE X,Y-1,N
330 IF N=0 THEN Y=Y-1:GOTO 240
340 LOCATE X+1,Y,N
350 IF N=0 THEN PLOT X,Y:X=X+1:GOTO 240
360 LOCATE X,Y+1,N
370 IF N=0 THEN PLOT X,Y:Y=Y+1:GOTO 240
380 PLOT X,Y
390 IF G=0 THEN PRINT "":PRINT "":PRINT "           ACABADO ":GOTO 90
400 X=INT(S(G)/256)
410 Y=S(G)-X*256
420 G=G-1:GOTO 240
500 IF PEEK(53279)<>6 THEN RETURN
510 GRAPHICS 7+32:POKE 752,1
515 POSITION 2,20
520 PRINT CHR$(125);"    ////SELECCION DE COLOR\\\\     "
530 PRINT "1:NARANJA   2:AMARILLO   3:AZUL"
540 IF PEEK(53279)=6 THEN 540
545 PRINT "":PRINT "":PRINT ""
550 RETURN
```

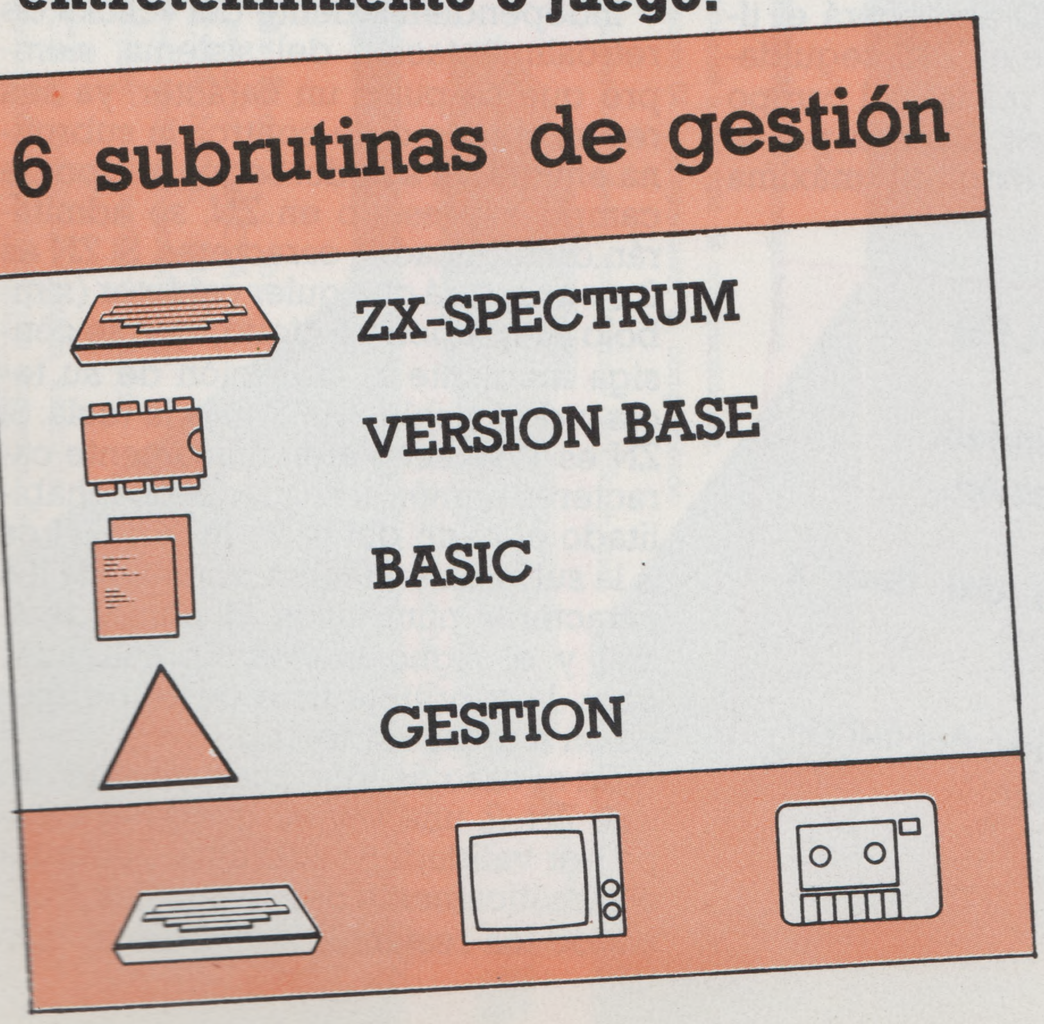
END

6 subrutinas de gestión



Un puzle para confeccionar programas

El Spectrum no es un ordenador orientado a la gestión comercial, sin embargo, en ningún caso debemos menospreciar sus cualidades y destinarlo a un uso de mero entretenimiento o juego.



Partiendo de una base de programación propia, podemos potenciar las cualidades del Spectrum, convirtiéndolo en un aparato apto para la realización de aplicaciones comerciales.

A continuación, presentamos un conjunto de 6 subrutinas programadas en BASIC, diseñadas para facilitar las tareas de gestión en este ordenador:

— Entrada de datos
— Emisión de mensajes
— Confección de menús e impresión de cabeceras
— Empaquetado/desempaquetado decimal
— Empaquetado/desempaquetado binario
— Suma algebraica de números comerciales.

El conjunto de subrutinas ocupa 82 líneas de programa, numeradas de la 2 a la 83, lo que permite la utilización ensamblada de todo el grupo o de sólo algunas de ellas.

A efectos de interacción con nuestros programas, hay que tener en cuenta que las mencionadas subrutinas manejan ciertas variables (esencialmente las del final del abecedario)

que hemos de considerar como reservadas; de ahí que no puedan utilizarse dentro del programa principal, so pena de distorsionar totalmente la operatividad del mismo.

Gracias a la rapidez de carga de que gozan los usuarios del microdrive, estos encontrarán muy útil comenzar el área de programa principal a partir de la línea 100, con lo que siempre se podrá efectuar un *MERGE* del bloque de subrutinas comerciales.

Para la correcta actuación de las subrutinas, el usuario ha de proporcionarles determinados parámetros en forma de variables BASIC. Estas variables serán interpretadas por la subrutina afectada, la cual brindará los resultados, a su vez, a través de variables de este tipo.

Cabe añadir que algunas subrutinas exigen la presencia de una variable cuya asignación es 32 espacios en blanco, variable que sólo es preciso inicializar una sola vez dentro del programa. Además de la anterior, también es necesario contar con una función Z0, cuya definición debe realizar el usuario en una de las líneas del programa por medio de la instrucción:
DEF FN Z(Z)=Z-INT (Z/256)*256

En resumidas cuentas, un método muy recomendable para construir programas de gestión basados en las subrutinas que presentamos es el siguiente:

1. Empezar creando la línea de programa que sigue:
   1 LET W$»
   DEF FN Z(Z)=Z-INT(Z/256)*256: GOTO 100
2. Incluir el conjunto de subrutinas (líneas 2 a la 83).
3. Situar a partir de la línea 100 el programa BASIC elaborado por el usuario para la gestión de las subrutinas necesarias. Este programa debe inicializar las variables-parámetro con los valores que se detallarán más adelante y, a continuación, saltar con un GOSUB a la línea de entrada a la subrutina que corresponda.



Esta es la subrutina de mayor longitud de todo el bloque (41 líneas). Permite la entrada de datos en cualquier punto de la pantalla, de forma similar a como lo hace el INPUT del sistema en la parte inferior de la pantalla, aunque, quizás, de forma más completa en algunos aspectos. Para utilizar esta subrutina hay que empezar inicializando las variables que caractericen la toma de datos que deseamos realizar.

Z$: título del campo a introducir.
ZF: fila de pantalla en la que se realiza la toma de datos.
ZC: columna de pantalla en la que se efectúa la toma de datos.
ZL: longitud máxima del dato a entrar.
Y$: contenido inicial del campo (puede ser Y$=".
ZN: tipo de campo a entrar. Si ZN es cero el dato se considera alfanumérico, si ZN es uno sólo se admiten números naturales y si ZN es dos se consideran números reales con dos decimales.

Una vez inicializados estos parámetros, se ejecuta la subrutina mediante *'GOSUP 2.* A partir de la posición expresada por ZF y ZC aparecerá el título del campo dado en Z$S; seguidamente el contenido inicial del campo (Y$), completado con blancos, por la derecha hasta su longitud máxima (ZL) y, finalmente, una marca de fin de campo (<).

En la primera posición del campo aparecerá un carácter de subrayado (_), que representa al cursor que marca nuestra posición. Para desplazar el cursor por el campo, sin modificar su contenido, utilizaremos las flechas a izquierda y derecha ya habituales (CAPS SHIFT y CAPS SHIFT 8). Mediante DELETE (CAPS SHIFT 0), podremos borrar el carácter que se encuentre sobre el cursor, aunque sin producir el arrastre de los demás, como estamos acostumbrados, sino sustituyéndolo por un carácter de espacio en blanco y retrocediendo el cursor un espacio.

Existen dos formas de dar por concluida la entrada. Mediante ENTER, se considerará que el dato ingresado está constituido por el conjunto de caracteres introducidos, desde el principio del campo hasta el carácter inmediatamente anterior al cursor. Pulsando EDIT (CAPS SHIFT 1), se considerará el dato como el campo completo (de longitud ZL) sin tener en cuenta la posición del cursor en el momento de la salida.

Independientemente del «click» de retroalimentación del sistema, siempre que se pulse un carácter, ya sea representado o de control, la subrutina emitirá un sonido. Según el tipo de campo expresado en ZN, se admitarán determinados caracteres. Si ZN es 0, se aceptará cualquier carácter (símbolo simple, letra o dígito) que se consiga mediante la pulsación de su tecla o de un solo SHIFT más la tecla. Si ZN es 1, se aceptarán únicamente caracteres numéricos, quedando inhabilitado el resto del teclado. Y si ZN es 2, la subrutina admitirá, además de los caracteres numéricos, el punto decimal y el signo menos. En cualquier caso, la subrutina dota de autorrepetición a todas las teclas.

En el caso de introducir un dato del tipo ZN=2, una vez se quede el campo por validado, la subrutina obtendrá automáticamente dos decimales, truncando los restantes si existieran más

CUADRO DE VARIABLES

Z	FOR de diversa utilidad.
ZC	Columna de impresión.
ZF	Fila de impresión.
ZL	Longitud máxima del dato a entrar.
ZN	Tipo del dato.
ZP	Puntero de posición del cursor.
Y$	Campo a modificar.
Z$	Dato entrado.

LISTADO

```
   1 REM Fernando Lopez Martinez # Subrutinas comerciales (entrada de datos) # Z
X Spectrum
   2 FOR z=1 TO zl: LET y$=y$+" ": NEXT z: LET y$=y$( TO zl): PRINT AT zf,zc;z$;
y$;"<": LET zc=zc+LEN z$: LET zp=1: PRINT AT zf,zc;
   3 PRINT  OVER 1;"_"; OVER 0;CHR$ 8;
   4 LET z$=INKEY$: IF z$="" THEN  GO TO 4
   5 IF CODE z$>31 THEN  GO TO 13
   6 IF z$=CHR$ 8 AND zp>1 THEN  PRINT  OVER 1;("_"+CHR$ 8 AND zp<zl+1);CHR$ 8;"
_"; OVER 0;CHR$ 8;: LET zp=zp-1: BEEP .01,32: GO TO 4
   7 IF z$=CHR$ 9 AND zp<zl+1 THEN  PRINT  OVER 1;"_";CHR$ 9;("_"+CHR$ 8 AND zp<
zl); OVER 0;: BEEP .01,32: LET zp=zp+1: GO TO 4
   8 IF z$=CHR$ 12 AND zp>1 THEN  PRINT  OVER 1;("_"+CHR$ 8 AND zp<zl+1);CHR$ 8;
 OVER 0;"_";CHR$ 8;: BEEP .01,32: LET zp=zp-1: LET y$(zp)=" ": GO TO 4
   9 IF z$=CHR$ 7 THEN  PRINT  OVER 1;("_"+CHR$ 8 AND zp<zl+1); OVER 0;: LET zp=
zl+1: GO TO 12
  10 IF z$<>CHR$ 13 THEN  GO TO 4
  11 PRINT ("_" AND zp<zl+1)
  12 BEEP .1,32: GO TO 29
  13 IF z$<" " OR z$>" THEN  GO TO 4
  14 IF zp>zl THEN  GO TO 4
  15 IF NOT zn THEN  PRINT z$; OVER 1;("_"+CHR$ 8 AND zp<zl); OVER 0;: GO TO 28
  16 IF zn>1 THEN  GO TO 19
  17 IF z$<"0" OR z$>"9" THEN  GO TO 4
  18 PRINT z$; OVER 1;("_"+CHR$ 8 AND zp<zl); OVER 0;: GO TO 28
  19 IF z$>"/" AND z$<":" THEN  PRINT z$; OVER 1;("_"+CHR$ 8 AND zp<zl); OVER 0;
  GO TO 28
  20 IF z$="-" THEN  GO TO 23
  21 IF z$="." THEN  GO TO 25
  22 GO TO 4
  23 IF zp>1 THEN  GO TO 4
  24 PRINT z$; OVER 1;"_"; OVER 0;CHR$ 8;: GO TO 28
  25 FOR z=1 TO zl: IF y$(z)="." THEN  GO TO 4
  26 NEXT z: IF zp=zl THEN  GO TO 4
  27 GO TO 24
  28 LET y$(zp)=z$: BEEP .01,32: FOR z=0 TO 7: NEXT z: LET zp=zp+1: GO TO 4
  29 IF zp=1 THEN  LET y$="": GO TO 31
  30 LET y$=y$( TO zp-1)
  31 IF NOT zn THEN  GO TO 41
  32 IF zn=2 THEN  GO TO 35
  33 IF zp=1 THEN  LET y$="0"
  34 GO TO 41
  35 IF y$="" THEN  LET y$="0.00": GO TO 41
  36 IF y$(1)<>"-" THEN  LET y$=" "+y$
  37 FOR z=1 TO LEN y$: IF y$(z)="." THEN  GO TO 39
  38 NEXT z: LET y$=y$+"."
  39 LET y$=y$+"00": LET y$=y$( TO z+2): IF y$(2)="." THEN  LET y$=y$(1)+"0"+y$(
2 TO )
  40 IF y$(1)=" " THEN  LET y$=y$(2 TO )
  41 LET z$=y$: FOR z=0 TO zl: LET y$=y$+" ": NEXT z: PRINT AT zf,zc;y$: IF zn=2
 THEN  LET y$=z$( TO LEN z$-3)
  42 RETURN
```

END

o completando a ceros si hubiera menos.

Al regresar de la subrutina, en Z$ estará el dato entrado y, en el caso especial de ZN=2, Y$ portará además la parte entera del dato introducido.



Dos de los métodos más habituales para guiar al usuario a través de lo programas de gestión, son la emisión de mensajes y la confección de menús. Su frecuente uso hace importante tanto la eficacia como el aspecto estético de los mensajes.

Estos mensajes siempre plantean una disyunción al usuario, a la que éste responderá mediante la pulsación de una tecla. El caso más frecuente es el de la exposición de cierta cantidad de datos, a continuación de los cuales aparece un mensaje de validación.

La subrutina de emisión de mensajes es muy simple en cuanto a los parámetros de entrada requeridos: Z$ es el título del mensaje a emitir y ZF es una bandera o indicador sobre la forma de emisión del mismo. Si ZF es 0 el mensaje aparecerá fijo y si ZF es 1, el mensaje parpadeará en la pantalla, acompañándose de un sonido.

La subrutina centrará automáticamente el mensaje en horizontal, y elegirá como línea de impresión la segunda a partir de la última escrita en pantalla.

La subrutina está diseñada con el fin de presentar en pantalla la información, definir los parámetros del mensaje y, por último, ejecutar *GOSUB 43*, lo cual presentará el mensaje en pantalla de forma estética, dejando una línea vacía entre la última posición escrita y el mensaje, y centrando este al ancho de la pantalla.

De la subrutina no se retornará hasta que el usuario haya pulsado una te-

LISTADO

```
  1 REM Fernando Lopez Martinez # Subrutinas comerciales (emision de mensajes)
# ZX Spectrum
 43 LET y$=z$: IF LEN y$>32 THEN  LET y$=y$( TO 32)
 44 PRINT : LET z1=24-PEEK 23689: LET zc=(32-LEN y$)/2: IF zf THEN  GO TO 48
 45 PRINT AT z1,zc;y$
 46 LET z$=INKEY$: IF z$="" THEN  GO TO 46
 47 PRINT AT z1,0;w$: BEEP .1,32: RETURN
 48 LET zf=-1*zf: PRINT (CHR$ 22+CHR$ z1+CHR$ zc+y$ AND zf<0)+(CHR$ 22+CHR$ z1+
CHR$ 0+w$ AND zf>0): BEEP .5,65: LET z$=INKEY$: IF z$="" THEN  GO TO 48
 49 GO TO 47
```

END

CUADRO DE VARIABLES

Z	FOR de diversa utilidad.
ZC	Columna de impresión del mensaje.
ZF	Bandera de modo de impresión.
ZL	Línea de impresión del mensaje.
$YS	Mensaje a imprimir.
$ZS	Tecla pulsada.
$WS	Cadena compuesta por 32 espacios en blanco.

cla, la cual puede ser investigada al regresar al programa principal mediante la variable Z$.

Hay que hacer notar que el parámetro ZF no conecta el modo habitual de FLASH del sistema, sino que el parpadeo se genera mediante el borrado y reescritura del mensaje, lo que se traduce en un resultado más elegante.



La forma más comúnmente utilizada en los programas de gestión para presentar disyunciones múltiples a un usuario es el menú. Menú, corrresponde en el argot informático, a una relación de opciones, a cada una de las cuales se le adjudica un número que ha de ser seleccionado por el usuario.

Debido a su simple estructura, es fácil programar una subrutina que confeccione los menús en pantalla. Gracias a ello ganamos en ocupación, economizando un buen puñado de PRINTS, y en estética, por la uniformidad que obtenemos al realizar todos los menús conforme a un patrón común.

De nuestro grupo de subrutinas comerciales, una de ellas está destinada precisamente a la confección de menús en pantalla. Estos aparecerán dejando una línea en blanco entre la primera opción del menú y la última línea escrita en la pantalla y dejando, asimismo, una línea en blanco entre cada una de las opciones que componen el menú. Cada una de las opciones irá precedida de un número, presentado en video inverso, que aparecerá en la segunda columna de la pantalla y, a continuación, a partir de la cuarta columna se visualizará el título de la opción correspondiente. Dadas las dimensiones de altura de la pantalla del Spectrum, el número máximo de opciones que se admiten en los menús es 9.

Para utilizar la subrutina se han de inicializar dos variables: ZL con el número de opciones del menú y Z con el número de línea en la que se encuentra el DATA con el título de cada una de las opciones del menú. Por lo tanto, es necesario que en la línea de programa Z exista una instrucción DATA con los títulos de las opciones. Acto seguido, realice *GOSUB 52.*. De la subrutina no se regresará hasta que el usuario pulse un número entre el 1 y el correspondiente a la última opción del menú. Al volver al programa principal, Z$ almacenará el código de la tecla accionada y Z el valor numérico de la misma.

CUADRO DE VARIABLES

Z	FOR y puntero para DATA con títulos de opción.
ZL	Número de opciones del menú.
Z$	Tecla pulsada.
W$	Cadena compuesta por 32 espacios en blanco.

Impresión de cabeceras

Lo más frecuente es que los menús se representen sobre una pantalla limpia y con una cabecera general o título del menú. Con tal finalidad, nuestro conjunto de subrutinas incluye una de cabeceras que realiza el borrado de la pantalla, y escribe en la primera línea un título centrado, en video inverso.

La única entrada de variable que precisa la subrutina es Z$, con el título de la cabecera, ejecutándose la subrutina mediante *GOSUB 50.*

Un ejemplo de confección de menú en base a las dos subrutinas presentadas sería:

```
10 LET Z$="MENU GENERAL": GOSUB 50
20 DATA"Opción uno", "Opción dos", "Opción tres", "Opción cuatro"
30 LET ZL=4: LET Z=20: GOSUB 52
40 REM A este punto no se llegará hasta que el usuario no haya pulsado las teclas 1, 2, 3 ó 4
```

ORIC-ATMOS



el orid ahora con un nuevo teclad.

Pidalo en
tiendas especializadas.

solicite
TARJETA DE
GARANTIA DSE

Comte d'Urgell, 118 - Tel. 323 00 66 - Tlx. 97760 DSIE-E - BARCELONA - 11
Infanta Mercedes, 83 - Teléfono 279 11 23 MADRID - 20

DSE, SA DISTRIBUIDORA DE SISTEMAS ELECTRONICOS, S. A.

LISTADO

```
  1 REM Fernando Lopez Martinez # Subrutinas comerciales (confeccion de menus e
 impresion de cabeceras) # ZX Spectrum
  50 CLS : IF LEN z$>32 THEN  LET z$=z$( TO 32)
  51 PRINT  INVERSE 1;(w$( TO 16-(LEN z$/2))+z$+w$( TO 16-(LEN z$/2)))( TO 32);
INVERSE 0: RETURN
  52 RESTORE z: FOR z=1 TO z1: READ z$: PRINT : PRINT " "; INVERSE 1;z; INVERSE
0;" ";z$: NEXT z
  53 LET z$=INKEY$: IF z$<"1" OR z$>STR$ z1 THEN  GO TO 53
  54 LET z=VAL z$: BEEP .1,32: RETURN
```

END

50 REM En Z$ tendremos el carácter pulsado y en Z el valor numérico del mismo

El menú elaborado por este programa ejemplo es el que reproducimos a continuación.

Empaq./ desempaq. decimal

Empaquetado decimal:

El ahorro de memoria en los pequeños microordenadores es un punto muy a tener en cuenta, sobre todo cuando. tratamos programas de gestión que suelen estar encaminados al manejo y almacenamiento de grandes volúmenes de datos.

Quizás el caso más notorio de desperdicio de memoria lo tenemos cuando almacenamos dígitos (0-9) en una variable numérica BASIC, ya que se les asigna un espacio de nada más y nada menos que 5 bytes, cuando en teoría sólo serían necesarios 4 bits. A nadie se le oculta que un desaprovechamiento de las nueve décimas partes de la memoria es absolutamente despilfarrador, de ahí que los números comerciales (entendiendo por números comerciales, aquellos que pueden alcanzar la cantidad de 99 millones incluyendo dos decimales, esto es: de hasta 10 cifras sin incluir el punto decimal), que no pueden ser tratados habitualmente por el BASIC Sinclair sin pasar a notación científica, suelen ser considerados como cadenas de simples caracteres, con lo cual el «despilfarrado» se reduce tan solo a un nibble (4 bits).

El método informático de ahorrar la máxima cantidad de memoria, mediante la codificación especial de la información, se denomina EMPAQUETADO. En nuestro bloque de gestión contamos con subrutinas establecidas a tal fin y, por supuesto, con las que realizan el efecto inverso, es decir, decodificación de la información para adoptar su configuración habitual (DESEMPAQUETADO).

Para utilizar la subrutina de empaquetado decimal, debemos establecer en Y$ la cadena que contiene el número a empaquetar (sin punto decimal) y ejecutar *GOSUB 55*. A regreso, tendremos en Z$ la cadena con el número ya codificado.

La subrutina considera si el número es negativo y consigue reducir a la mitad el espacio ocupado por la cadena (en el caso de cadena de longitud par) o, en el peor de los casos, a la mitad más un byte (cadena original de longitud impar).

CUADRO DE VARIABLES

Z	FOR de diversa utilidad.
ZN	Signo del dato a empaquetar o desempaquetar.
Y$	Dato a procesar.
Z$	Dato procesado.

Desempaquetado decimal

La subrutina de desempaquetado decimal realiza la función inversa a la anteriormente expresada, restableciendo a su forma original una cadena empaquetada mediante la subrutina de empaquetado decimal.

Como en su opuesta, esta subrutina sólo necesita de una entrada: la cadena que contiene el número empaquetado que estará en Y$, y sólo tiene una salida: Z$, que contiene la cadena original desempaquetada. La ejecución se efectúa con un *GOSUB 58*.

LISTADO

```
  1 REM Fernando Lopez Martinez # Subrutinas comerciales (empaquetado/desempaqu
etado decimal) # ZX Spectrum
  55 LET z$="": LET zn=0: IF VAL y$<0 THEN  LET y$=y$(2 TO ): LET zn=1
  56 LET y$="0"+y$: FOR z=LEN y$-1 TO 1 STEP -2: LET z$=CHR$ VAL (y$(z)+y$(z+1))
+z$: NEXT z: IF zn THEN  LET z$(1)=CHR$ (CODE z$(1)+128)
  57 RETURN
  58 LET z$="": LET zn=0: IF CODE y$(1)>128 THEN  LET y$(1)=CHR$ (CODE y$(1)-128
): LET zn=1
  59 FOR z=1 TO LEN y$: LET v$="0"+STR$ CODE y$(z): LET z$=z$+v$(LEN v$-1 TO ):
NEXT z: IF z$(1)="0" THEN  LET z$=z$(2 TO )
  60 IF zn THEN  LET z$="-"+z$
  61 RETURN
```

END



Empaquetado binario:

En los programas de gestión, el sistema habitual para el acceso a registros multicampo (conjuntos de información interrelacionada, con varios datos o campos) es el de otorgar un número de registro o clave; éste suele contener, normalmente, información significativa sobre el tipo de registro al que nos referimos, codificada en formato numérico para el ahorro de espacio.

Para aclarar el punto anterior comentaremos un ejemplo: La empresa para la que trabajamos lleva su contabilidad de forma mecanizada y cada una de las cuentas se configura como un registro multicampo, pues contiene diversa información (datos) sobre un mismo objeto de información (la cuenta). Así por ejemplo, el registro podría contener el nombre de un cliente, el saldo inicial que mantiene con la empresa al principio del ejercicio, la cantidad de movimientos que se han realizado al «debe» durante el ejercicio y la cantidad de movimientos realizados al «haber» durante el ejercicio.

La forma más coherente de clasificar la información antedicha, sería otorgar un número de registro con el que poder almacenarla. Usualmente este número, en el caso de una aplicación contable, será un código de 7 cifras, denominado subcuenta, cuyos tres primeros dígitos indicarán el tipo de cuenta (430=Cliente), los dos siguientes la letra del alfabeto a que pertenece el título (01=A, 02=B, etc...) y los dos últimos, el número de orden del registro dentro de los que pertenecen a la misma letra.

Por lo que hemos visto, la codificación de la clave es absolutamente *sui generis,* pero lo que *sí* que es habitual es que sea un número natural y no de gran longitud. Para el empaquetado de estos números especiales, nuestro grupo de subrutinas dispone de un empaquetado que denominamos binario, puesto que utiliza la base 2. Gracias a ésta podremos empaquetar números naturales de la forma más ventajosa. Por ejemplo, la cantidad 16000000 que sería empaquetada por la subrutina decimal en 5 bytes, es empaquetada por la subrutina binaria en tan solo 3 bytes.

Para la ejecución de la subrutina sitúe en Y$ la clave a empaquetar y efectúe un *GOSUB 62.* Al regresar al programa principal Z$ contendrá el resultado empaquetado.

CUADRO DE VARIABLES

Z	Valor del dato a empaquetar en la subrutina de tal fin.
Y$	Dato a procesar.
Z$	Dato procesado.
Z (Z)	Función para desempaquetado.

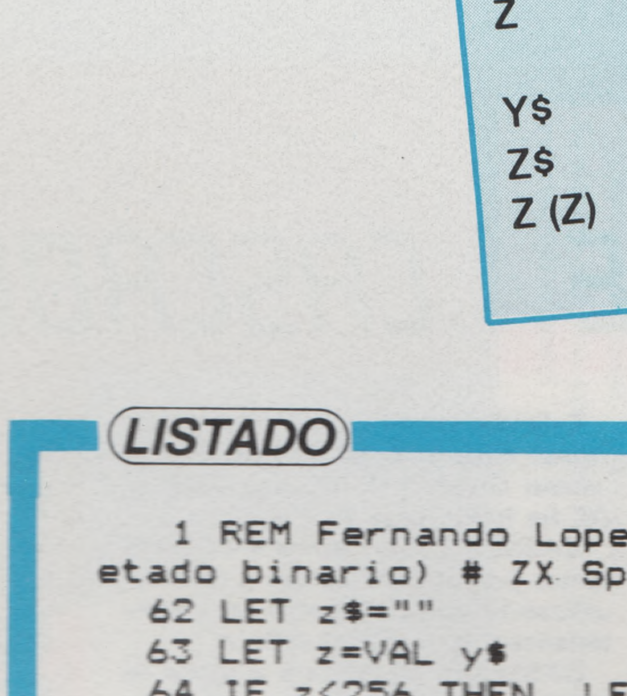
Desempaquetado binario:

La subrutina de desempaquetado binario funciona de forma similar a la explicada para el desempaquetado decimal. En Y$ debe entrar la clave empaquetada mediante la subrutina de empaquetado binario, y en Z$ saldrá el resultado ya desempaquetado; para la ejecución utilice *GOSUB 67.*

LISTADO

```
   1 REM Fernando Lopez Martinez # Subrutinas comerciales (empaquetado/desempaqu
etado binario) # ZX Spectrum
  62 LET z$=""
  63 LET z=VAL y$
  64 IF z<256 THEN  LET z$=CHR$ z+z$: GO TO 66
  65 LET z$=CHR$ FN z(z)+z$: LET z=INT (z/256): GO TO 64
  66 RETURN
  67 LET z$=""
  68 LET y$=CHR$ 0+CHR$ 0+y$: LET y$=y$(LEN y$-2 TO ): LET z$=z$+STR$ (CODE y$(1
)*65536+CODE y$(2)*256+CODE y$(3)): RETURN
```

END



La operación fundamental en las aplicaciones de gestión comercial (la única en la contabilidad general, esto es, no analítica, presupuestaria o de costes), es la suma algebraica, entendiéndose por suma algebraica, la suma y resta (suma con signo).

Esta operación con números comerciales no podemos encomendarla al BASIC, puesto que es muy posible que pase el resultado a notación científica (en base a potencias de 10), de ahí que optemos por dedicar la última de las subrutinas a gestionar la suma algebraica de números comerciales.

CUADRO DE VARIABLES

Z	Valor numérico de parte de la cadena de los sumandos.
ZN	Como la anterior.
ZP	Bandera de signo negativo del resultado.
Y$	Sumando.
Z$	Sumando.
V$	Cadena puente en la obtención del resultado.

La subrutina exige dos entradas: los sumandos, que serán introducidos en las variables Y$ y Z$ sin punto decimal. Así, por ejemplo, si se quieren sumar las cantidades: 87538810,34 y 40563591,00 el programa adecuado será el siguiente:

```
10 LET Y$="8753881034": LET Z$="4056359100": GOSUB 69.
```

Una vez que se regrese de la subrutina tendremos en Z$ el resultado, sin punto decimal; según el ejemplo anterior: 12810275134, lo que supondría 128.102.751.34.

La subrutina considera automáticamente los signos, de manera que una resta se realizará, simplemente, enviando el sumando con su correspondiente signo menos. tuMicro

LISTADO



```
1 REM Fernando Lopez Martinez # Subrutinas comerciales (suma algebraica) # ZX
 Spectrum
  69 LET zp=0: LET zn=0: IF z$(1)="-" OR y$(1)="-" THEN  GO TO 77
  70 LET z=VAL z$(LEN z$-2 TO )+VAL y$(LEN y$-2 TO ): IF z>999 THEN  LET zn=1: L
ET z=z-1000
  71 LET v$="00"+STR$ z: LET v$=v$(LEN v$-2 TO ): IF ABS VAL y$<1000 THEN  LET y
$="0"+y$
  72 IF ABS VAL z$<1000 THEN  LET z$="0"+z$
  73 LET z$=STR$ (zn+VAL z$( TO LEN z$-3)+VAL y$( TO LEN y$-3))+v$
  74 IF z$(1)="0" THEN  LET z$=z$(2 TO )
  75 IF zp THEN  LET z$="-"+z$
  76 RETURN
  77 IF z$(1)=y$(1) THEN  LET zp=1: LET z$=z$(2 TO ): LET y$=y$(2 TO ): GO TO 70
  78 IF ABS VAL z$<ABS VAL y$ THEN  LET v$=y$: LET y$=z$: LET z$=v$
  79 IF z$(1)="-" THEN  LET zp=1: LET z$=z$(2 TO ): GO TO 81
  80 LET y$=y$(2 TO )
  81 IF z$=y$ THEN  LET z$="000": RETURN
  82 LET z=VAL z$(LEN z$-2 TO )-VAL y$(LEN y$-2 TO ): IF z<0 THEN  LET zn=1: LET
 z=1000+z
  83 LET v$="00"+STR$ z: LET v$=v$(LEN v$-2 TO ): LET z$="0"+z$: LET y$="0"+y$:
LET z$=STR$ (VAL z$( TO LEN z$-3)-VAL y$( TO LEN y$-3)-zn)+v$: GO TO 74
```

END

CASCADE CASCADE CASCADE CASCADE

LA CINTA DEL AÑO
por sólo 3.700 Pts.

Dirección:
C/ CRISTOBAL BORDIU, 35
MADRID-3 -TFNO. 254 24 30

N.° Y NOMBRE DEL JUEGO

1. MUNCHER (MASTICADOR)
2. SKI JUMP (SALTO DE SKI)
3. BASKETBALL (BALONCESTO)
4. FROGGER (RANAS)
5. BREAKOUT (ESCAPE)
6. CRUSHER (APLASTADOR)
7. STARTREK (VIAJE A LAS ESTRELLAS)
8. MARTIAN KNOCK OUT (KAO MARCIANO)
9. BOGGLES (DESORBITADO)
10. ALIEN ATTACK (ATAQUE EXTRATERRESTRE)
11. LUNAR LANDER (ATERRIZAJE LUNAR)
12. MAZE EATER (COMEDOR DE LABERINTO)
13. MICROTRAP (PEQUEÑA TRAMPA)
14. MOTORWAY. (AUTOPISTA)
15. LABYRINTH (LABERINTO)
16. SKITTLES (JUEGO DE BOLOS)
17. RACE TRACK (PISTA DE CARRERAS)
18. SKI RUN (SLALON)
19. TANKS (TANQUES)
20. SOLAR SHIP (NAVE SOLAR)
21. TEN PINS (10 ALFILERES)
22. CARS (COCHES)
23. STOMPER (BEISBOL)
24. PINBALL (FLIPPER)
25. CAVERN (CUEVA)
26. LASER (LASER)
27. ALIEN (EXTRATERRESTRE)
28. CARGO (CARGA)
29. THE RACE (CARRERA)
30. THE SKULL (EL CRANEO)
31. ORBIT (ORBITA)
32. MUNCH (MUNCH)
33. BOWLS (BOLOS)
34. RAIDERS (SAQUEADORES)
35. FIELD (EL CAMPO)
36. DRAGGOLD (DRAGON DE ORO)
37. SPACE SEARCH (BUSQUEDA EN EL ESPACIO)
38. INFERNO (INFIERNO)
39. NIM (NIM)
40. VOYAGER (VIAJANTE)
41. SKETCH PAD (CUADERNO DE ESBOZOS)
42. BLITZ (ALARMA ANTIAEREA)
43. FISHING MISSION (MISION DE PESCA)
44. MYSTICAL DIAMONDS (DIAMANTES MISTICOS)
45. GALAXY DEFENSE (DEFENSA DE LA GALAXIA)
46. CYPHER (CIFRAR)
47. JETMOBILE (COCHE JET)
48. BARREL JUMP (SALTAR EL BARRIL)
49. ATTACKER (ATRACADOR)
50. SPACE MISSION (MISION ESPACIAL



Felicidades. Ya existe una sensacional cinta con 50 juegos que se llaman CASSETTE-50, que CASCADE ha traido para tí.

CASSETTE-50, contiene 50 excitantes juegos para cargar y disfrutar utilizando completamente las características de tu computadora.

CASSETTE-50, atraerá a gente de todas las edades, y los juegos proporcionarán muchas horas de entretenimiento para toda la familia, a la fracción del coste de otros juegos de computadora.

La mayoría de los juegos que contiene CASSETTE-50, pueden jugarse con la computadora de tamaño standard. Solamente algunos de los juegos más grandes requieren una memoria más larga (MORE RAM).

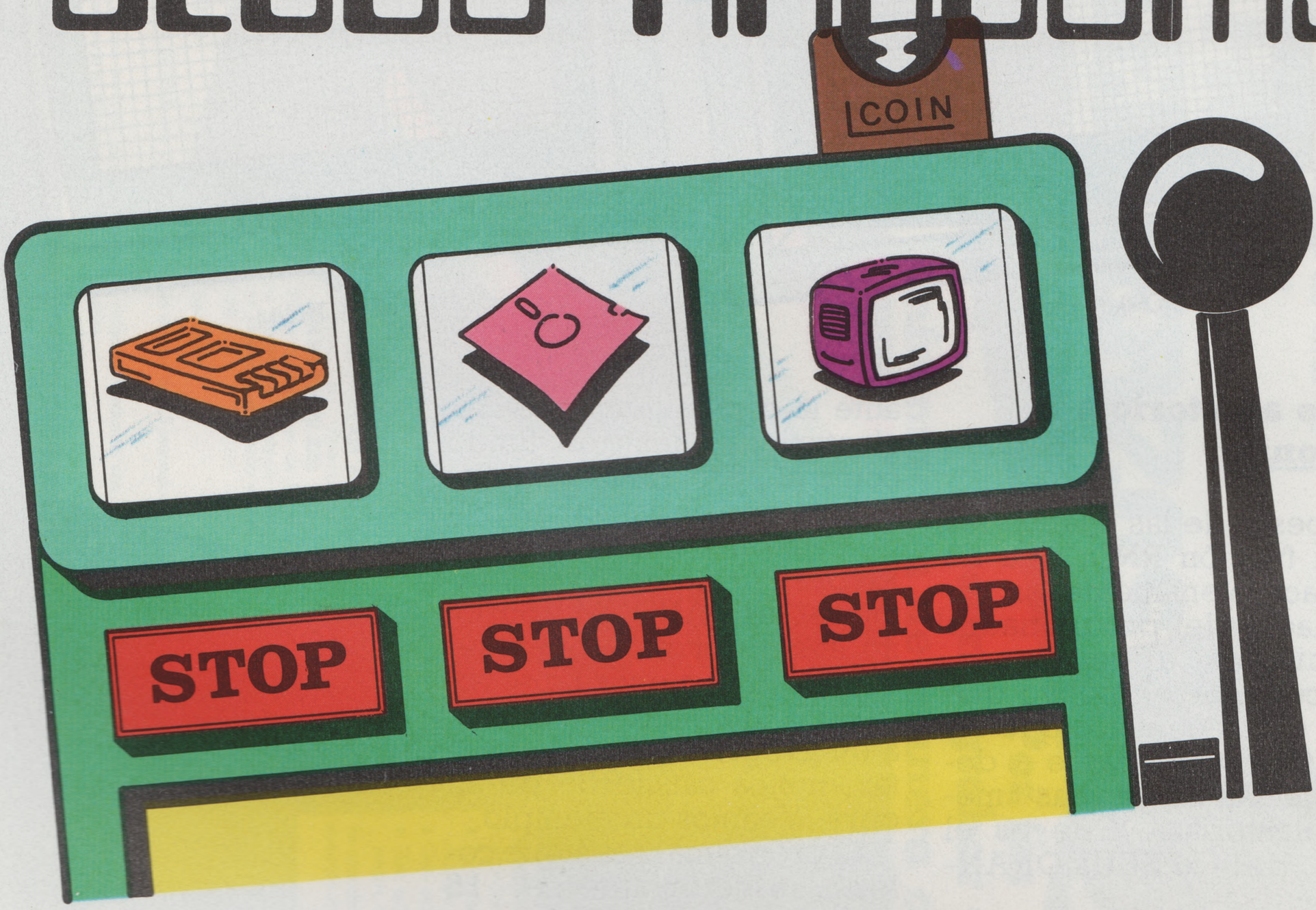
VIC-20 / ORIC
SPECTRUM / BBC / APPLE
DRAGON
ATARI / COMMODORE-64
en Julio 84

Remita este cupón a Cascade, C/ Cristóbal Bordiu, 35. Madrid-3

NOMBRE .....
APELLIDOS .....
DIRECCION .....
POBLACION ..... D.P. .....
PROVINCIA ..... TELEFONO .....
FORMA DE PAGO: ☐ TALON CONFORMADO ☐ GIRO POSTAL ☐ CONTRA REEMBOLSO

CASCADE CASCADE CASCADE CASCADE

PSEUDO-AAOOOMS



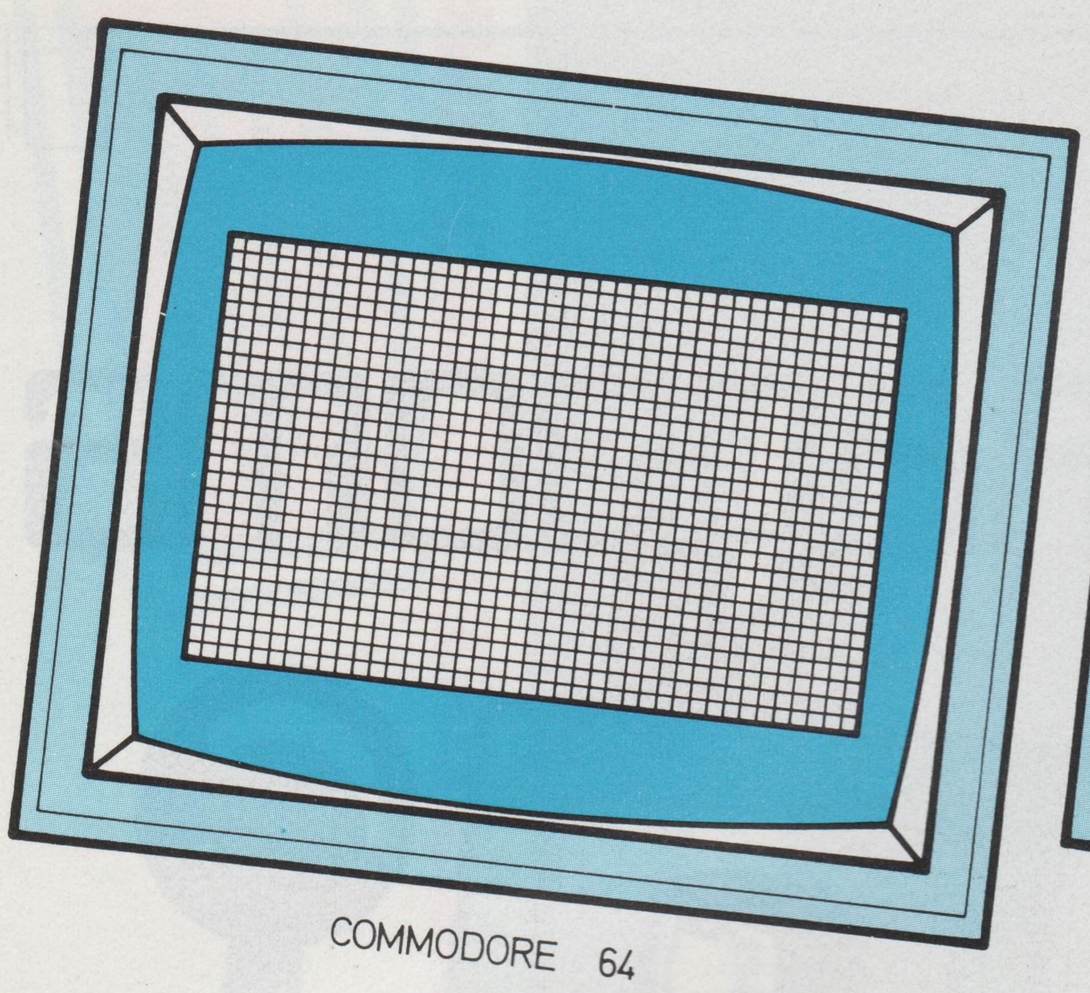
La aleatoriedad puede hacernos ricos, aunque también puede llevarnos a la catástrofe... ¡Quién pudiera controlar el azar! Los ordenadores nos permiten la enorme ventaja de poder controlar la suerte, suprimiéndole su «porción» de riesgo. No estaría nada mal utilizarlo en la vida real y que siempre nos tocara la lotería, ¿verdad?

La generación de números aleatorios, característica casi exclusiva del lenguaje BASIC, resulta un gran aliado en la programación de juegos y simulaciones en tan popular lenguaje. No obstante, las peculiaridades de esta función en los diferentes dialectos obliga al usuario a bucear en el manual de cada equipo concreto en aras a un correcto uso de la generación aleatoria.

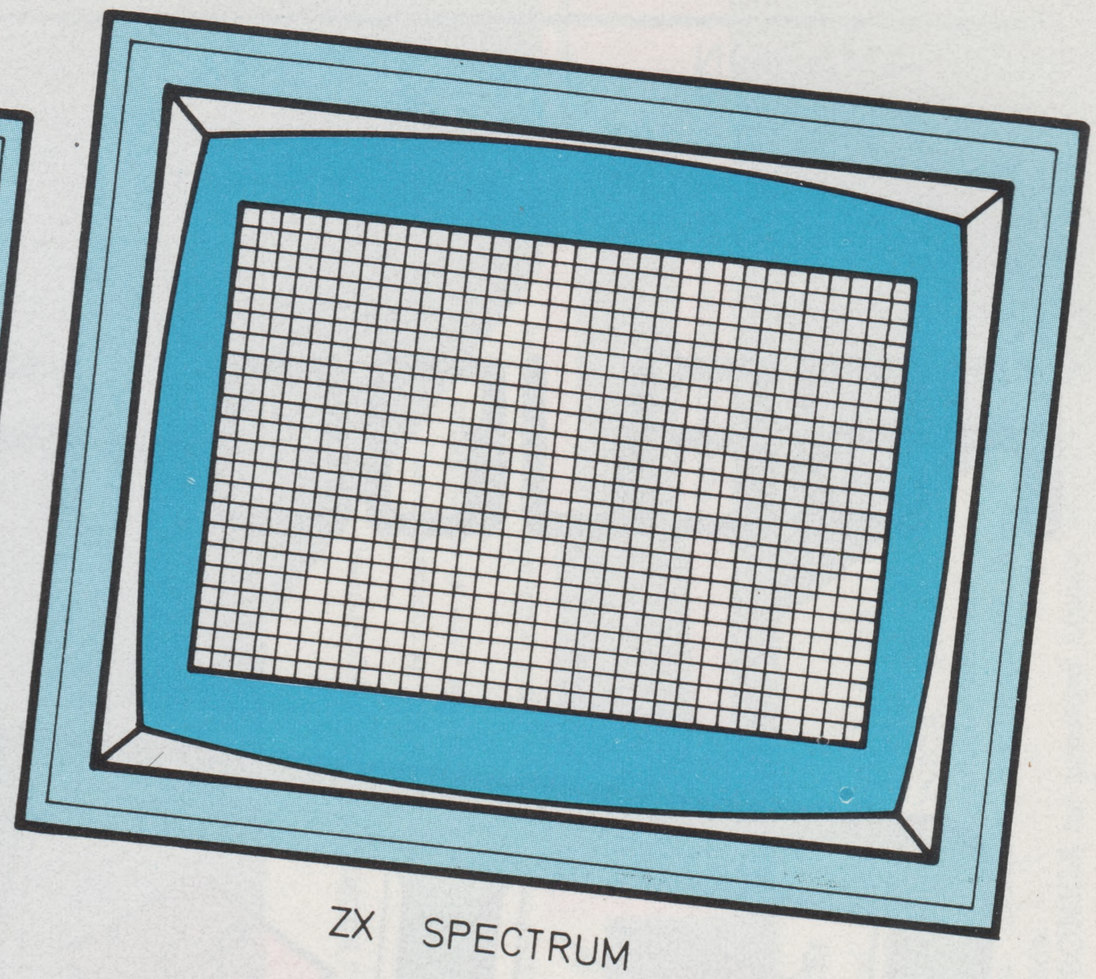
Explicar los fundamentos matemáticos del RND no es cosa fácil ni común a todos los microordenadores aunque, normalmente, el método es como sigue: se parte de un número base («semilla») generado por el sistema, tomado del contador de cuadros de TV o introducido por el usuario, y se calcula una secuencia de cocientes, aparentemente aleatorios, llegando a un resultado que puede adoptar un valor comprendido entre 0 y 1 (no pudiendo ser 1, ya que el numerador de la fracción es siempre menor que el denominador).

¿Cómo diseñar pantallas de juego controlando la aleatoriedad de la función RND?

Tu Micro N.º 3/4, Pág. 39



COMMODORE 64



ZX SPECTRUM

De lo aleatorio a lo racional

A pesar de las cualidades de la función RND, ésta no satisface plenamente las necesidades del programador en ciertas aplicaciones. A veces, por el motivo que sea, es necesario que lo «aleatorio» se adapte a determinadas exigencias «menos aleatorias». Este es el caso del «PSEUDORANDOM».

Esta técnica, que podríamos considerar como un RANDOM «racional», ya que combina las prestaciones de la función RND con las exigencias concretas de programación, se convierte en una eficaz herramienta para el diseño de pantallas en juegos de animación. Es posible que el lector no iniciado en las tareas de programación de juegos no comprenda muy bien en qué consiste este sistema de «racionalización». Para explicarlo mejor, forzaremos un poco la imaginación suponiendo que queremos diseñar una pantalla de simulación para un juego del tipo LUNAR LANDER.

Si el fondo de la pantalla (cielo estrellado) y el perfil del suelo (crestas lunares) van a ser siempre los mismos, reservando los mismos lugares para el despegue y alunizaje, el juego no va a resultar muy vistoso.

Llegados a este punto, y ante la conveniencia de dar aleatoriedad tanto al fondo como al perfil de la pantalla, nos vemos en la obligación de echar mano de la función RND. Pero, desgraciadamente, el RANDOM «puro» no suele dar muy buen resultado..., al menos no el resultado que puede apetecer a un programador «detallista», nunca satisfecho de su obra. Veamos un ejemplo.

El programa que sigue coloca, aleatoriamente, 16 puntos luminosos sobre la pantalla de un COMMODORE 64 (25 filas × 40 columnas); pantalla que previamente el usuario habrá «limpiado» y a la que habrá asignado el color negro.

```
FOR I=Ø TO 15
X=RND(Ø)*1ØØØ+1Ø24
POKE X,46
POKE X+54272,1
NEXT I
```

¡Ya tenemos el cielo estrellado! Para los menos familiarizados con este microordenador, añadiremos que la pantalla de caracteres ocupa las posiciones de memoria comprendidas desde 1024 a 2023, mientras que la memoria de color está localizada entre las posiciones 54272 a 55271.

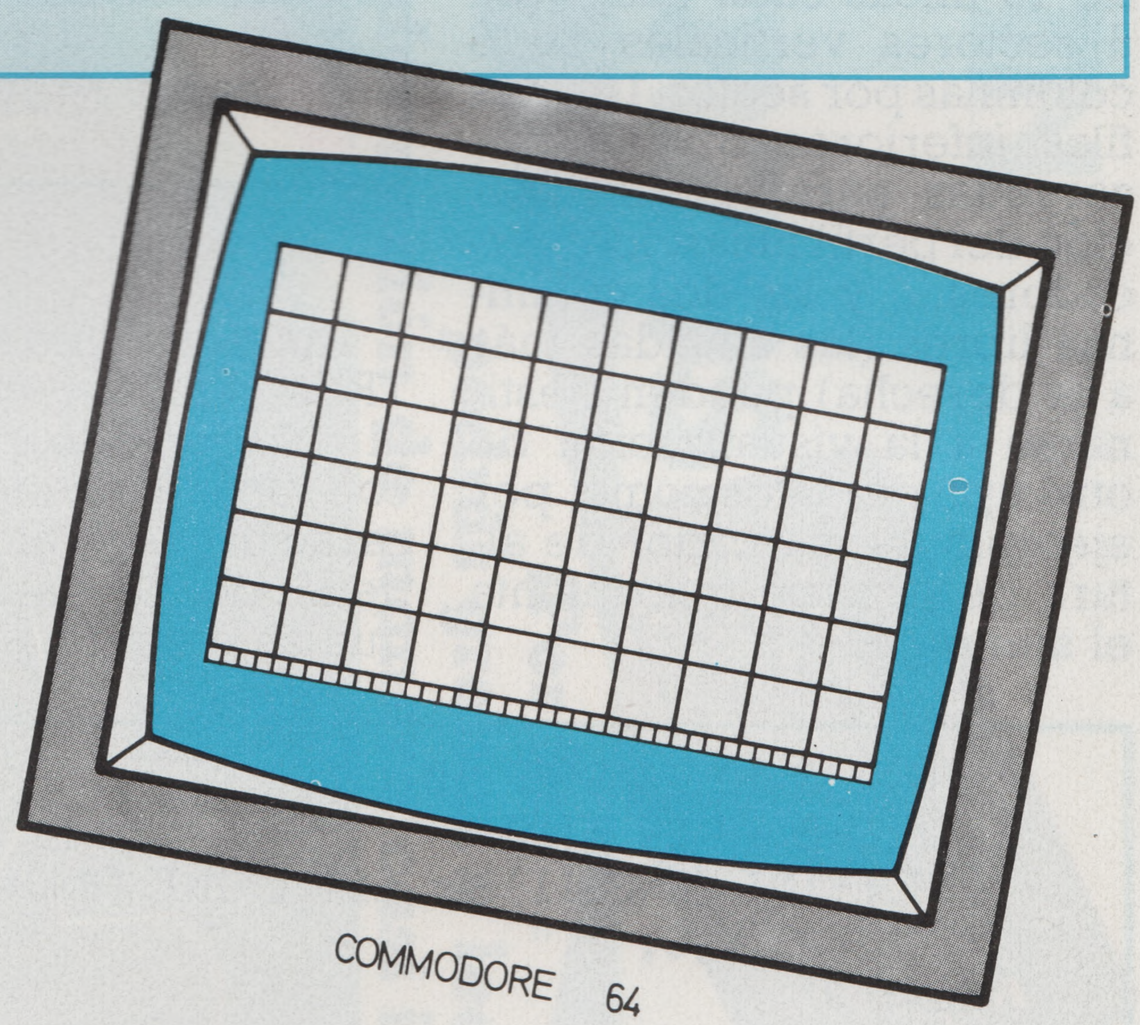
Un ejemplo semejante para los incondicionales del SPECTRUM: en este caso, el

programa en cuestión coloca 11 puntos aleatorios sobre la pantalla que, en el caso del SPECTRUM es de 22 filas por 32 columnas. Por supuesto, las 11 «estrellas» se visualizarán en las posiciones de coordenadas F,C (F=fila y C=columna) que se generarán al recorrer sucesivamente el bucle que constituye el programa.

Observe que no hemos utilizado la fila 25 (fila inferior de la pantalla). A la hora de crear un programa, por ejemplo de «alunizaje», tampoco emplearíamos las 24 filas que se consideran en el programa anterior: usaríamos tan sólo las 18 primeras filas para el fondo con estrellas, reservaríamos las filas 19 a 24 para la presen-

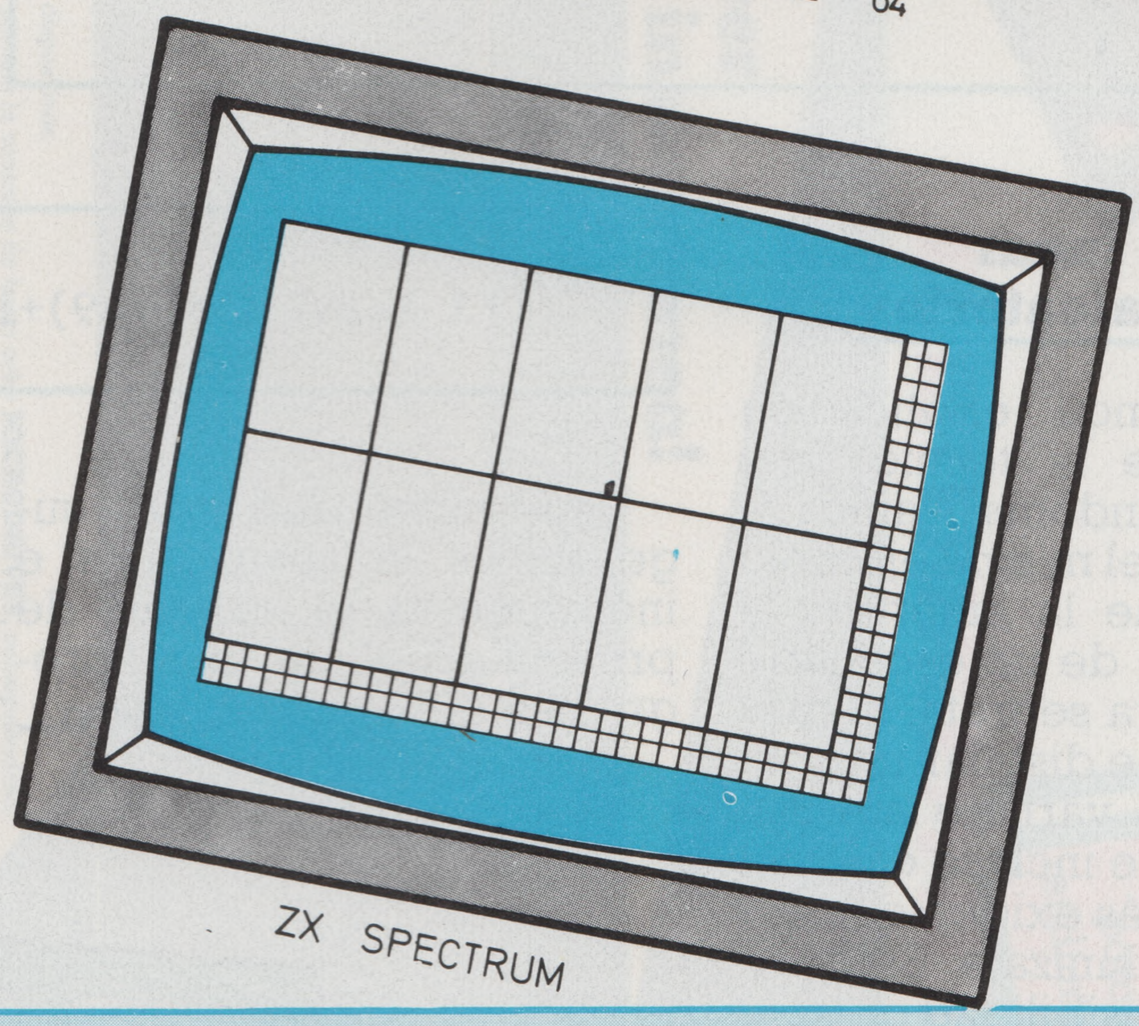
```
FOR I=Ø TO 1Ø
LET F=INT (RND*22): LET C=INT (RND*32)
PRINT AT F,C;"."
NEXT I
```

En ambos casos, cada vez que se accede a la rutina propuesta se obtienen resultados diferentes: con un simple programa podemos generar innumerables fondos estrellados. Y ahora toca la de «arena». El entusiasmo se viene abajo al comprobar que la propia aleatoriedad del RND nos condena a que la distribución de estrellas sobre la pantalla no quede tan uniforme como cabría desear: aparecerán zonas con alta densidad de puntos y otras sin ninguno..., ¡o calvos, o con tres pelucas...! ¡Hay que dar con otra solución más estética!



Una solución estética...

... aunque algo más complicada es —en el caso del COMMODORE 64— dividir la pantalla en 16 sectores (6 filas × 10 columnas), y aplicar luego la función RND en cada uno de ellos, con lo que el resultado será una distribución de estrellas mucho más uniforme.




```
FOR I=1Ø24 TO 1744 STEP 24Ø
FOR J=Ø TO 3Ø STEP 1Ø
X=I+J+INT(RND(Ø)*6)*4Ø+INT(RND(Ø)*1Ø):POKE X,46:POKE X+54272,1
NEXT J,I
```

tación del perfil lunar, y la 25 para visualizar los marcadores y demás indicaciones relativas a la evolución del juego. En cualquier caso, la rutina propuesta —«experta» en racionalizar la función RND—, sirve como punto de partida para lograr el fondo estelar que desee el usuario.

Tampoco en este caso vamos a olvidarnos de los usuarios de SPECTRUM. El progama siguiente ilustra el método comentado, si bien, en este caso, la pantalla del SPECTRUM la hemos dividido en 2 zonas horizontales, de 10 líneas cada una, y en 5 sectores verticales de 6 columnas por sector. Las dos filas inferiores quedan reservadas para la visualización del perfil lunar, los marcadores..., y las dos columnas libres (las situadas más a la derecha) pueden destinarse a la visualización de otro tipo de datos como, por ejemplo, el marcador de altura de la nave que intenta el alunizaje.

```
FOR I=Ø TO 1Ø STEP 1Ø
FOR J=Ø TO 24 STEP 6
LET F=INT (RND*11)+I: LET C=INT (RND*7)+J
PRINT AT F,C; "."
NEXT J
NEXT I
```

Un perfil lunar «pseudo-aleatorio»

La segunda explicación práctica de la técnica del «pseudo-random» vamos a extraerla del mismo ejemplo general que la anterior: la generación de un escenario lunar. Ahora se trata, ni más ni menos, de diseñar un perfil selenita, variado aunque coherente, e incluso «móvil» (para los más exigentes), apto para el alunizaje. Está claro que la posibilidad de configurar el perfil que deseamos recurriendo a un simple RND de caracteres es ilusoria y máxime cuando pretendemos disponer, además, de determinadas zonas de alunizaje. A pesar de ello, la solución no es compleja. Para empezar, hay que definir una cadena de caracteres de longitud apropiada —por ejemplo, de 80 caracteres— que configure el perfil lunar e incluya los puntos de alunizaje. A partir de aquí, la creación de un perfil lunar pseudoaleatorio resulta de lo más fácil: basta con elegir una subcadena de 40 caracteres por medio de la función RND y escribirla en la pantalla. Una simple línea de programa es capaz de realizar esta tarea en el COMMODORE 64:

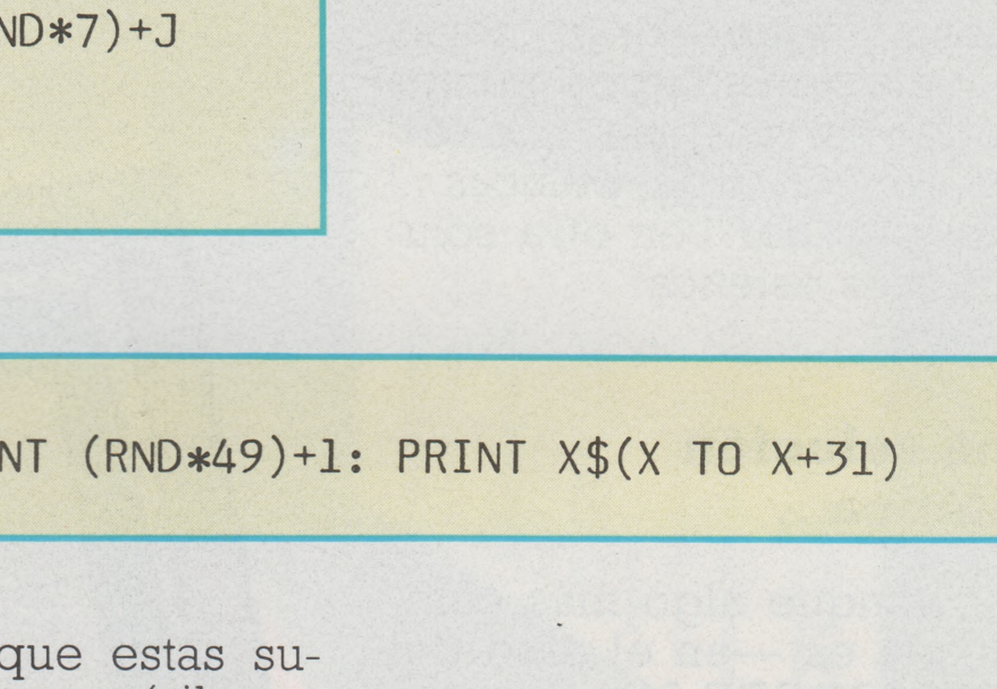
```
PRINT MID$(X$,RND(Ø)*41+1,4Ø)
```

Tan fácil como en el SPECTRUM, en cuyo caso el perfil coherente lo construimos con una subcadena de 32 caracteres extraída de la cadena de 80 caracteres X$ que se definirá a tal efecto.

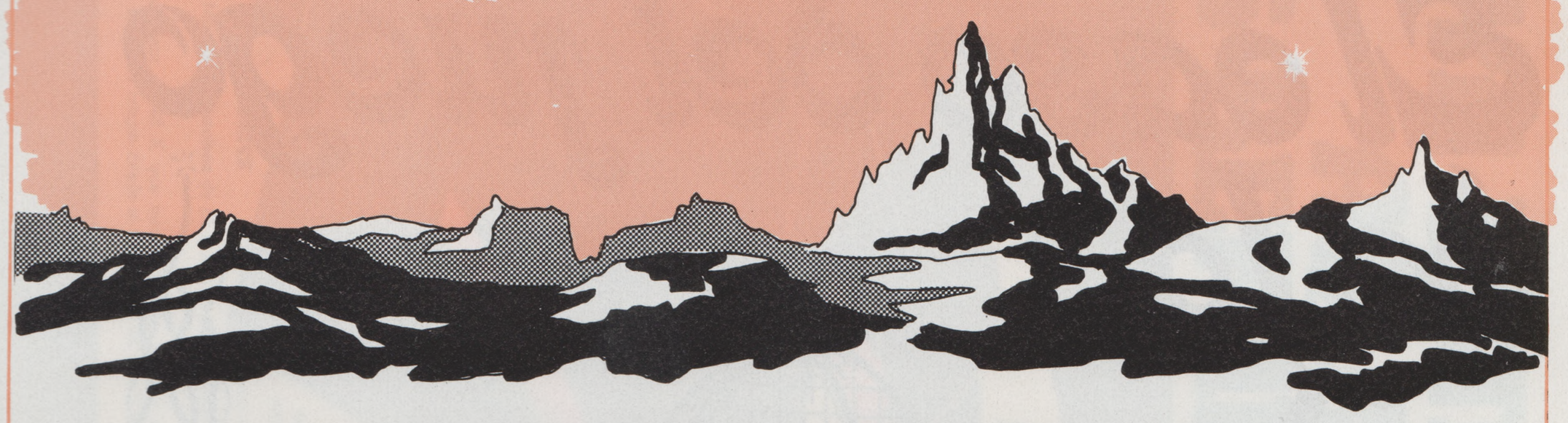
```
LET X=INT (RND*49)+1: PRINT X$(X TO X+31)
```

Esperamos que estas sugerencias le sean útiles e incrementen la calidad de presentación de sus programas.

tu Micro



Paisajes lunares en el SPECTRUM

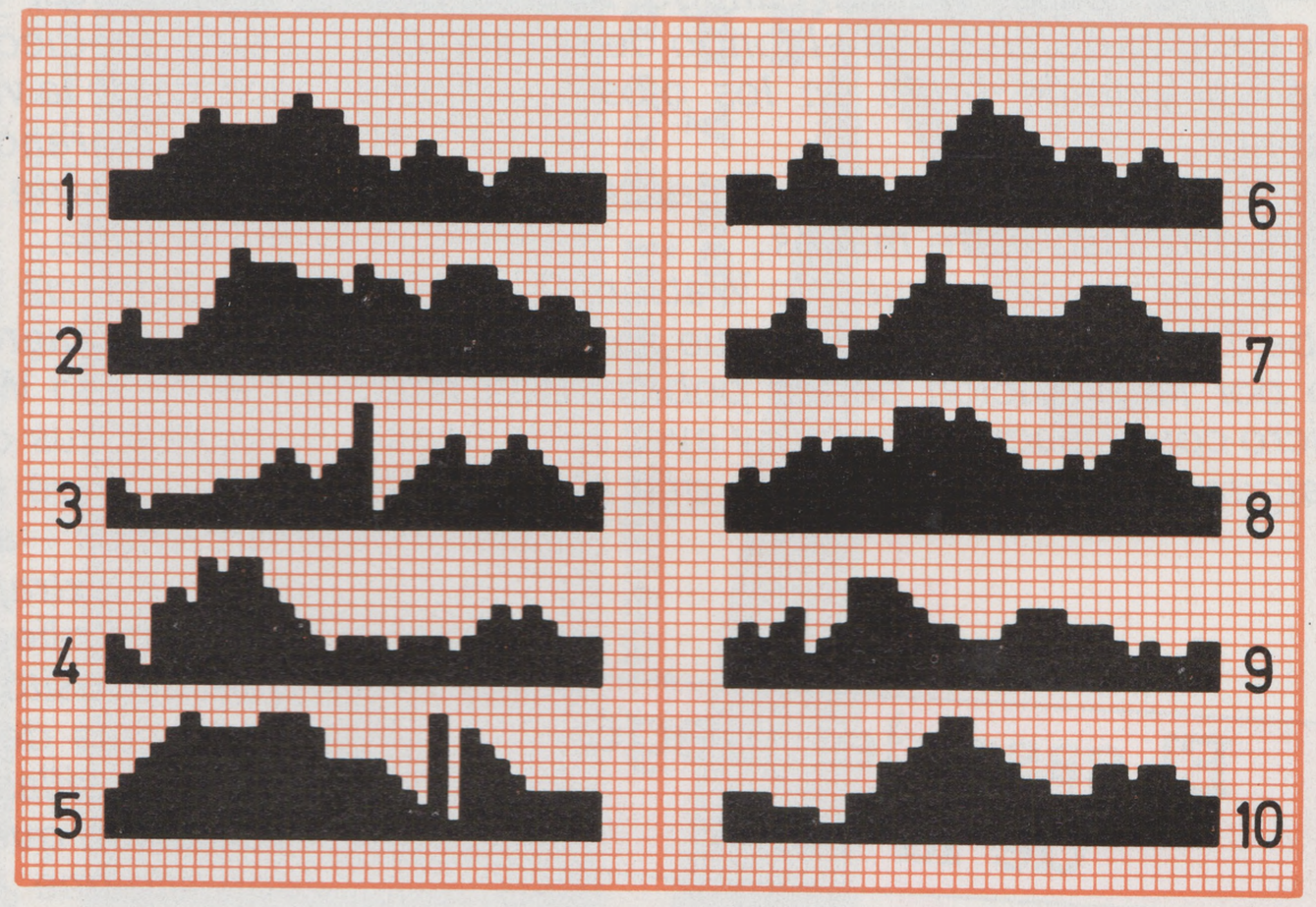


Otra de las técnicas habituales para la construcción de cadenas pseudo-aleatorias es la que podemos denominar como «síntesis de subcadenas coherentes». El método se basa en la definición de un determinado número de subcadenas cuyo enlace sea coherente en cualquier orden y el posterior montaje de la cadena principal por medio de la función RND.

La figura ilustra que la simplicidad del método es bastante más fácil que el nombre que le hemos otorgado. Aplicando esta técnica al ejemplo del perfil lunar, podemos observar en la figura el diseño de un cierto número de subcadenas que pueden ser ensambladas en cualquier orden, dando una sensación de continuidad.

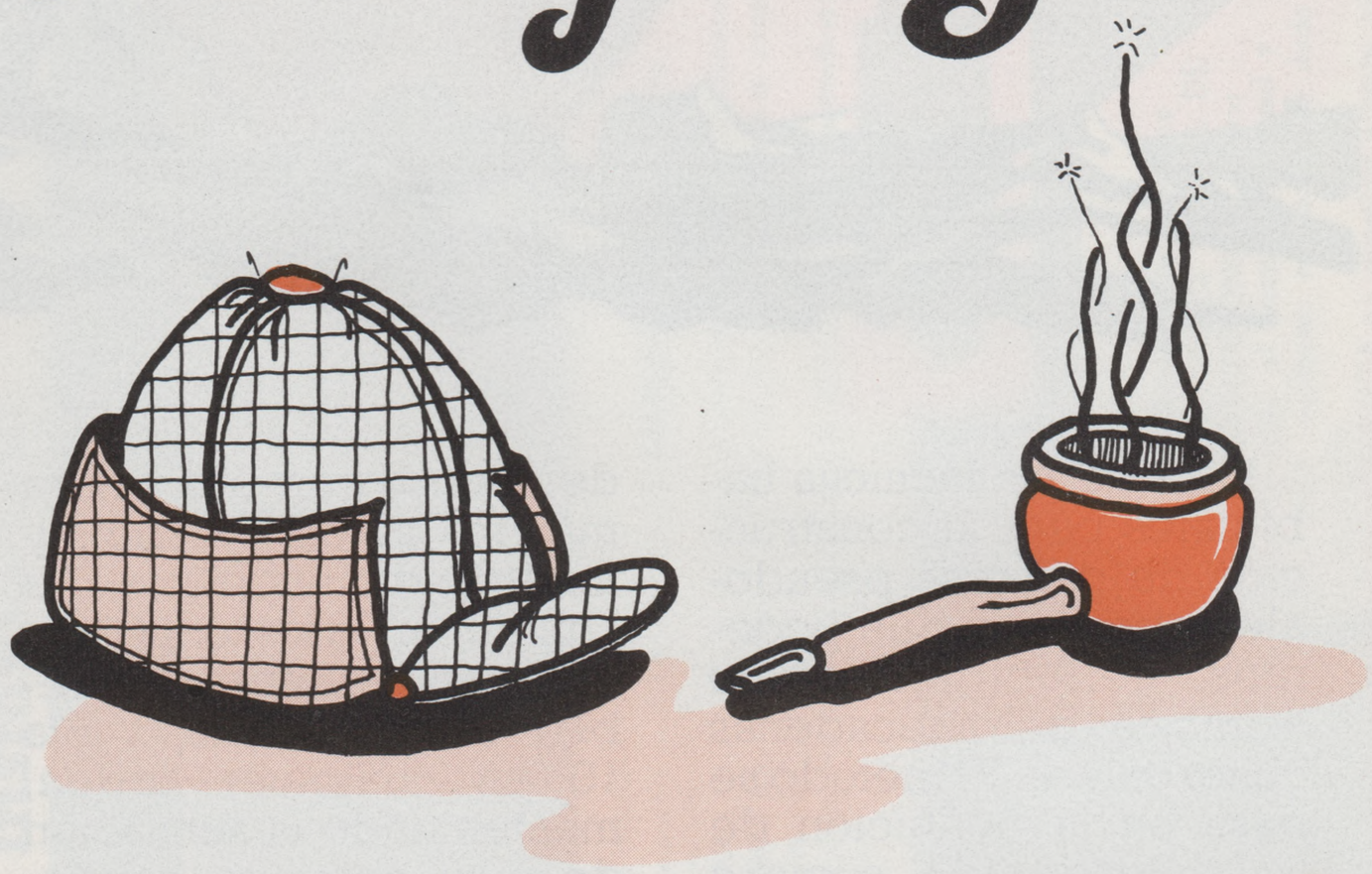
En el ejemplo representado en la figura, hemos definido 10 subcadenas o series de 32 puntos de ancho, con una altura máxima de ocho puntos y mínima de un punto. La condición de «coherencia» que permitirá ensamblar las subcadenas 1 a la 10 en cualquier orden, manteniendo la sensación de perfil continuo, consiste en hacer coincidente la altura de las columnas primera y última de cada subcadena. En nuestro caso, esta altura se ha fijado en tres puntos.

Eligiendo de forma aleatoria 8 de las subcadenas o series definidas —ya sean repetidas o distintas—, y combinándolas de acuerdo a un orden también aleatorio, obtendremos un perfil continuo de 256 puntos de ancho, coincidente con los 32 caracteres del SPECTRUM.





El caso del juego de la vida



Aquella tarde, cuando acudí a la mansión de mi querido amigo Sherlock Holmes, tuve la sensación de que un nuevo caso acechaba. Al aparecer su rostro tras la curtida puerta de caoba vi la excitación reflejada en su cara.

—¡Pase, pase Watson, amigo mío! —dijo, acompañándome hacia el fondo de la estancia, con un extraño brillo en los ojos— ... Tome asiento.

—Creo no equivocarme al pensar que tiene un nuevo caso entre manos...

—¡Algo mejor mi querido Watson! Como Ud. sabe, hace ya algún tiempo que ando interesado por la evolución de las poblaciones, tanto micro como macroscópicas, y ayer tarde llegué a algunas conclusiones. Creo que sus amplios conocimientos médicos van a sernos de mucha utilidad.

—No alcanzo a comprender a qué se refiere Sr. Holmes —confesé con un tono de voz que denotaba mi sorpresa.

—¡Vamos, vamos, Watson! Creo que usted, como médico, tiene suficientes conocimientos acerca de la vida y, además, en los años que lleva colaborando conmigo sus conocimientos sobre la muerte han debido incrementarse de forma notable.

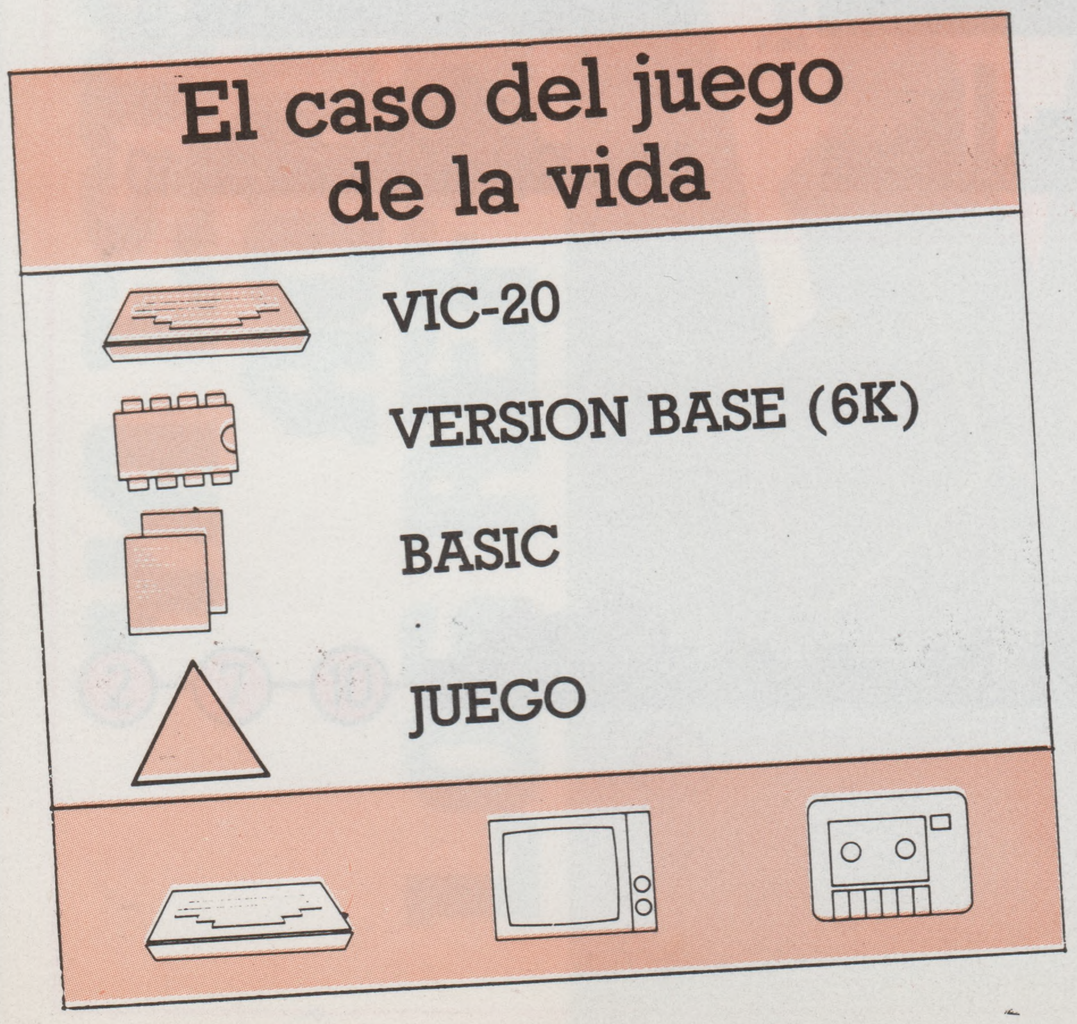
—Sigo sin entenderle mi querido amigo... ¿Dónde quiere ir a parar?

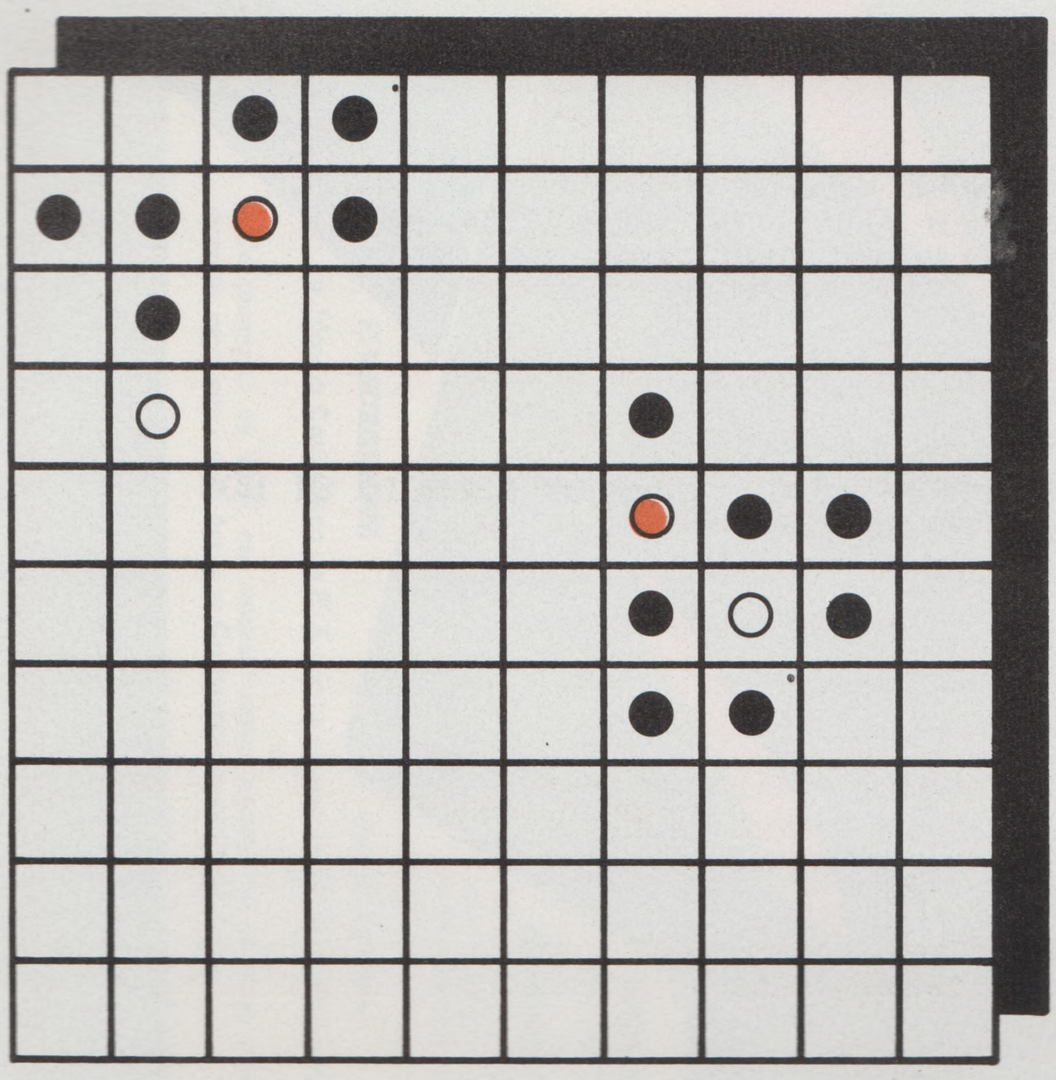
—Ya veo Watson que persiste en no querer emplear sus células grises... Como de costumbre me veré obligado a ayudarle en sus razonamientos —añadió en tono compasivo, al tiempo que extrajo del bolsillo de su inseparable batín un pliego de papel, repleto de dibujos, que depositó sobre la mesa—. Observe esta cuadrícula de 10 por 10, e imagine que en algunas casillas existen individuos o entes de la población que vamos a estudiar... Y ahora Watson, dígame ¿cuál cree que debe ser la condición para que en un punto de la cuadrícula nazca un nuevo individuo?

—Bien... ¡Me coge Ud. por sorpresa querido amigo! Sin embargo, creo que para que nazca un nuevo individuo en determinada casilla bastaría con que tenga al menos dos individuos de la generación anterior en casillas adyacentes.

—¡Bravo Watson! Creo que me va comprendiendo. Supongamos que para que nazca un nuevo individuo es necesario que haya tres individuos de la generación anterior, en lugar de dos, en las casillas vecinas. Más adelante, entenderá por qué. Bien, pasemos ahora al tema de la muerte. Dígame: ¿cuándo cree usted que morirá un individuo?

El caso del juego de la vida





INDIVIDUO
NACIMIENTO DE UN INDIVIDUO EN LA PROXIMA GENERACION
INDIVIDUO QUE PERECERA (POR AISLAMIENTO O POR ASFIXIA) EN LA PROXIMA GENERACION

—Desde luego, mi experiencia me indica que un individuo aislado tiene más oportunidades de perecer...

—¡Muy bien! Pongamos que para que un individuo permanezca vivo necesita otros dos a su alrededor. Por lo tanto, aquel individuo que tenga menos de dos entes a su alrededor perecerá irremisiblemente... ¡Piense un poco más! ¿Se le ocurre otra razón para que un individuo muera?

—Ummm... Si no está aislado no veo como ha de perecer.

—¡Mi querido Watson!... creo que ha olvidado Ud. la muerte por asfixia. Piense en un individuo rodeado por tantos otros que se ve abocado a la muerte por asfixia o, sencillamente, por falta de recursos para su subsistencia.

—Sí, creo que tiene razón... podría darse el caso. Pero ¿cuál es el número de entes vecinos que produce la asfixia del sujeto?

—¡Intentemos deducirlo! Revisemos las condiciones que hemos supuesto hasta ahora. Si hemos establecido que son dos individuos los que deben rodear a un tercero para que no perezca y tres para generar un nacimiento... ¡Elemental, mi querido Watson! nada más lógico que prever que con cuatro individuos colindantes nuestro sujeto va a morir por asfixia o por falta de recursos.

—Cierto, no cabe duda que parece lo más acertado...

—Y si no tiene Ud. ninguna objeción, creo que estas leyes son suficientes para el estudio de la evolución de las poblaciones —replicó, con evidentes muestras de satisfacción.

—Efectivamente —le contesté— aunque me parece excesivamente laborioso empezar a desarrollar sus conclusiones con lápiz y papel.

—Desde luego, está usted en lo cierto, Watson. Pero, como ya sabe, procuro no dejar cabos sin atar...

Con una velocidad más propia de un joven alocado que de un reposado e inteligente detective, Holmes saltó del sillón y desapareció tras la puerta del laboratorio.

Recuerdo que eran ya las siete de la tarde cuando recibí su inesperada llamada telefónica. Ahora, tumbado en el mullido sillón testigo de tantos casos resueltos por la aguda capacidad deductiva de mi amigo Holmes, vino a mi mente la agitación de su voz a través del teléfono. Un hombre inteligente, de rostro frío e insondable, que los años y aventuras a su lado me habían enseñado a conocer. Al cruzar el umbral, su rostro excitado confirmó mis secretas expectativas de encontrarme ante un nuevo y fascinante caso.

—Como Ud. recordará, amigo Watson, hace algunas fechas le puse al corriente de la última carta que recibí de Lord Jennings —la voz de Holmes llegaba desde el laboratorio, acompasada por el ruido de sus pasos que se encaminaban de nuevo hacia el salón—. En ella me relataba las fantásticas posibilidades de una nueva máquina que acababa de recibir procedente de América. Pues bien, amigo Watson, aquí está el referido artilugio. La lectura de su misiva me dejó tan atónito que no dudé un instante en pedir a Lord Jennings que tuviera a bien cederme por unos días tan fascinante aparato.

—... Mmmm ¡VIC-20!... ¿Tal vez un nuevo analizador de huellas dactilares?

—Nada de eso mi querido Watson, se trata de algo mucho más interesante y novedoso, que espero resulte de gran utilidad en la investigación que me propongo realizar.

—¿No pretenderá Ud. que este ingenio es capaz de estudiar la evolución de una comunidad según las leyes que acaba de referirme?

—En efecto, amigo mío ¡eso es lo que propongo!

La seguridad que había en sus palabras me obligó a silenciar mi escepticismo.

—He examinado con detalle el legajo de instrucciones que acompaña al aparato y he llegado a la conclusión que no ha de ser difícil educar a la máquina por medio de un programa que simule las leyes que hemos enunciado.

En medio de la perplejidad con la que reaccionaba a sus aseveraciones, mi vista se posó en el reloj que adornaba la estancia. La hora de mi cita con Sir Ian Spechs estaba al caer y tuve que despedirme de Sherlock Holmes.

La curiosidad que sentía por el nuevo invento que obraba en las manos dé mi sagaz amigo y la apasionante investigación que se porponía acometer, me incitó a volver a casa de Holmes la mañana siguiente. Esta vez mi sorpresa fue mayor le encontré con el pelo desaliñado, ojeroso y medio oculto entre la densa nube de humo que inundaba la habitación.

—¡Querido Watson! ¡Adelante! Tendrá que disculpar mi aspecto, puede imaginar mi actividad durante la pasada noche. Creo que mi vo-

LISTADO



```
10 REM (C) RUSA # JUEGO DE LA VIDA # COMMODORE VIC-20
20 PRINT"♥QQQ  ESTE ES EL JUEGO DE  Q      LA VIDA","QQ  REALIZADO POR
30 PRINT"Q  SANTIAGO RUIZ","Q  DAVID SANTAOLALLA","QQQ  PARA TUMICRO
40 PRINT"QQQ  RPULSA UNA TECLA_
50 GETA$:IFA$=""THEN50
110 H=10:V=10
120 DIM A(H,V),C(H,V),A$(H,V)
135 PRINT "♥QQQQ  DIBUJA LA PRIMERA          GENERACION"
140 PRINT"QCADA LINEA DEBE ESTAR    ENTRE PARENTESIS","QDAR UN ESPACIO SI NO"
150 PRINT"   HAY PEON Y UN *","      SI LO HAYQQ"
210 G=1
220 FORI=1 TO H
240 PRINT "LINEA ";I
250 INPUTB$
260 B$=LEFT$(B$,12)
270 FOR J=0 TO V
280 A(I,J)=0
290 NEXTJ
300 FOR J=1 TO (LEN(B$)-1)
310 IF MID$(B$,J,1)<>"*" THEN 330
320 A(J-1,I)=1
330 NEXTJ
340 NEXTI
1000 FOR I=1 TO H
1010 FOR J=1 TO V
1020 C(I,J)=0
1030 NEXTJ
1040 NEXTI
1050 G=G+1:P=0
1060 FOR I=1 TO H
1070 FOR J=1 TO V:N=0
1090 IF J=0 THEN 1130
1100 IF A(I,J-1)=1 THEN N=N+1
1110 IF I=H THEN 1170
1120 IF A(I+1,J-1)=1 THENN=N+1
1130 IF I=H THEN1170
1140 IF A(I+1,J)=1 THENN=N+1
1150 IF J=V THEN 1210
1160 IF A(I+1,J+1)=1 THEN N=N+1
1170 IF J=V THEN 1210
1180 IF A(I,J+1)=1 THEN N=N+1
1190 IF I=0 THEN 1300
1200 IF A(I-1,J+1)=1 THEN N=N+1
1210 IF I=0 THEN 1300
1220 IF A(I-1,J)=1 THEN N=N+1
1230 IF I=0 THEN 1300
1240 IF A(I-1,J-1)=1 THEN N=N+1
1300 IF N<2 THEN 1350
1310 IF N>3 THEN 1350
1320 IF N=3 THEN 1340
1330 IF A(I,J)=0 THEN 1350
1340 C(I,J)=1:P=P+1
1350 NEXTJ
1360 NEXTI
1370 FOR I=0 TO H
1380 FOR J=0 TO V
1390 A(I,J)=C(I,J)
1400 NEXTJ
1410 NEXTI
1420 IF P=0 THEN 1600
1425 GOSUB 8000
1430 PRINT"♥"
1435 PRINT"     ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1440 FOR I=1 TO V
1460 PRINT "     ▒£ ,A$(1,I)A$(2,I)A$(3,I)A$(4,I)A$(5,I)A$(6,I)A$(7,I)
1465 PRINTSPC(13)"■";A$(8,I)A$(9,I)A$(10,I);"←▒"
1500 NEXTI
1510 PRINT"     ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1520 PRINT"QQQGENERACION ";G,"POBLACION=";P
1530 GOSUB2000
1540 GOTO 1000
1600 PRINT "♥QQQQ  ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒","  ▒  NO HAY NINGUN ▒","  ▒INDIVIDUO EN L
A ▒"
1610 PRINT"  ▒ GENERACION     ▒","  ▒ SIGUIENTE      ▒"
1620 PRINT"  ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1645 PRINT"QQQQQQQQQQR■QUIERES VOLVER A JUGAR_←"
1650 GET G$:IF G$="" THEN GOTO1650
1660 IF G$="S" THEN RUN
1700 END
2000 POKE36878,7:POKE36875,240
2010 FOR I=1TO75:NEXT
```

```
2020 POKE36878,0:RETURN
8000 FOR I=1 TO H
8020 FOR J=1 TO V
8030 IF A(I,J)=1 THEN A$(I,J)="♥"
8040 IF A(I,J)=0 THEN A$(I,J)=" "
8060 NEXTJ:NEXTI
8080 RETURN

READY.
```

END

luntaria vigilia ha sido fructífera y plenamente satisfactoria... Observe —añadió, mostrándome una ristra de hojas de papel empalmadas y delicadamente plegadas una sobre otra—, aquí está el programa.

No sé cuanto tiempo se prolongó la escena, aunque creo que el suficiente para que mi amigo Sherlok Holmes percibiera mi mirada atónita paseando sin rumbo alguno por entre los extraños símbolos y las inescrutables palabras que rebosaban del papel.

—¡Ni más ni menos que el listado del programa que hará que la nueva máquina realice la investigación por nosotros!

Listado, programa... palabras que para mí no tenían significado alguno y que jamás había oido a Holmes pronunciar con anterioridad. En un intento de forzar su explicación sin verme obligado a reconocer mi total ignorancia, repliqué, mostrando un convincente interés por los papeles.

—Un completo éxito Mr. Holmes, no cabe duda... si bien, temo no comprender como desearía alguna de las partes del programa.

La frase dió el resultado que anhelaba. La inclinación al monólogo pedagógico de Sherlock Holmes salió de inmediato a la luz.

—Permítame que le explique, querido amigo. Como verá, la estructura del programa es poco menos que diáfana. Observe, de la línea 130 a la 150 he incluido la cabecera del programa, en la que se explica el procedimiento para introducir en la máquina los datos de la primero generación. Las líneas 210 a 340 forman la rutina de entrada de esta primera generación.
El cuerpo del programa va de la 1000 a la 1040. A su vez, el bloque comprendido entre las líneas 1060 y 1360 es quien realiza casi todo el trabajo, creando la nueva generación en la matriz C. En las líneas 1000 a 1040 se inicializa C a ceros. Y en las 1370 a 1410 se pasan los datos de C a la matriz correspondiente a la generación actual, A.

—Eso está claro... pero ¿cuál es la función de las líneas 8000 y siguientes?

—¡Elemental, mi querido Watson! Esta subrutina forma una nueva matriz, esta vez de caracteres, que se verá reflejada en la pantalla. Las instrucciones 1430 a la 1600 se encargarán de visualizar dicha matriz, así como el número de individuos almacenados en P... Y, por supuesto, la línea 15600 se ejecuta sólo en el caso de que no exista ningún individuo en la nueva generación.

—Clarividente, amigo mío...

—La verdad es que la investigación ha resultado más fácil de lo que suponía, e incluso, durante el proceso de confección y prueba del programa, he llegado a ciertas conclusiones de interés. Sin ir más lejos, he descubierto algunos tipos de poblaciones característicos; por ejemplo las estables, que no evolucionan significativamente, o las oscilantes cuya evolución es alternativa en uno y otro sentido... ¡querido Watson, le desafío a que Ud. localice nuevos tipos!

Mis múltiples obligaciones no me han permitido por el momento, poner en práctica la sugerencia de Sherlock Holmes. Pero quizás algún lector con más tiempo que yo encuentre algunas de estas escurridizas poblaciones cuya evolución sigue un desarrollo característico. ¡Ah! se me olvidaba contarles un último detalle que Holmes me hizo observar: los caracteres que aparecen subrayados en el listado del programa, corresponden a los de control cuya representación en pantalla es el propio caracter subrayado en vídeo invertido... Una investigación más resuelta con un completo éxito... ¡Hasta el próximo caso! tuMicro

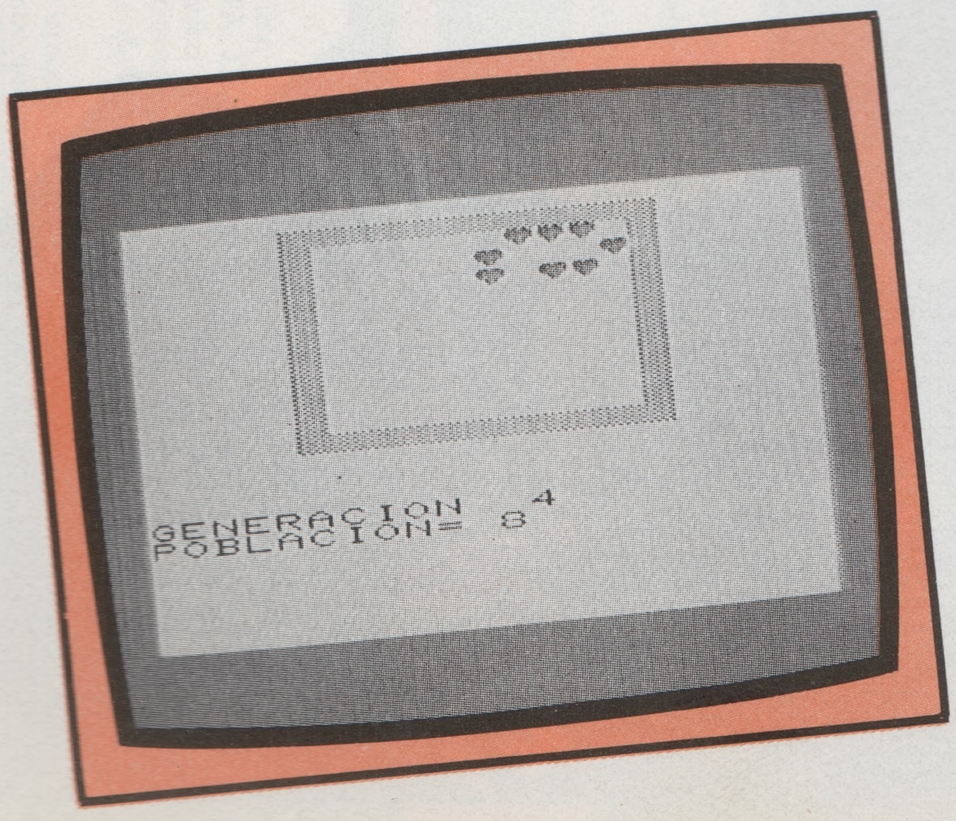


TABLA DE VARIABLES

A(H,V)	Matriz que contiene la generación actual.
C(H,V)	Matriz con la nueva generación.
A$	Contiene la visualización de la generación actual.
G	Cuenta el número de generación.
P	Cuenta el número de individuos.
I y J	Variables que barren el contenido de las matrices dentro de los bucles de programa.

Suscríbase a

Electro OCIO

La revista mensual de electrónica, micro-informática y tecnología... ¡para todos!

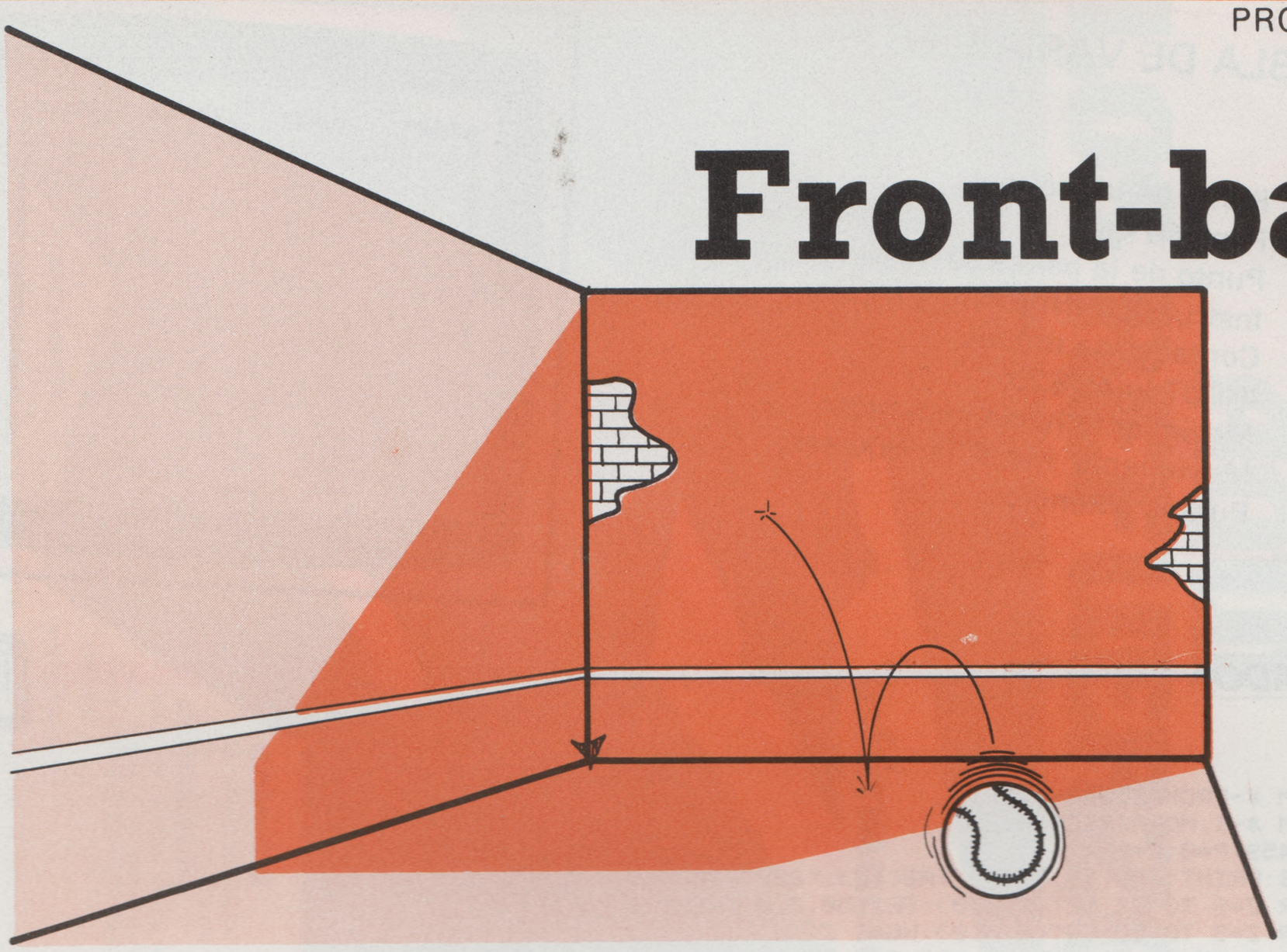
... y recibirá como regalo la

guía de los COMPONENTES ELECTRONICOS



Para formalizar su suscripción llame por teléfono al
91/250 58 20.

Front-ball



Una pelota parte de un punto aleatorio localizado en la zona superior de la pantalla. El jugador debe estar atento para devolverla de un raquetazo —¡dos puntos más que pasan a engrosar su marcador!— y si además de interceptarla, consigue introducirla en las porterías situadas en la pared frontal, su maestría se verá recompensada con 10 ó 20 puntos adicionales.

Una ojeada al listado adjunto evidencia la sencillez de este programa, elaborado a partir de instrucciones BASIC de tipo común y sin complicaciones alguna en su formulación.

La estructura del programa está edificada en base a los siguientes bloques. Entre la línea 10 y la 100 se inicializan las oportunas variables y se dibuja el recinto de juego. La siguiente zona (líneas de la 110 a la 200) gestiona el salto a las diversas rutinas y subrutinas hacia las que puede derivar la secuencia de ejecución, por ejemplo, para el control del desplazamiento de la raqueta a derecha e izquierda o para gobernar el movimiento de la pelota en sentido vertical u oblícuo. El bloque delimitado por las líneas 210 y 240 se ocupa de controlar la progresión de la pelota cuando esta parte de un punto aleatorio situado en la pared frontal.

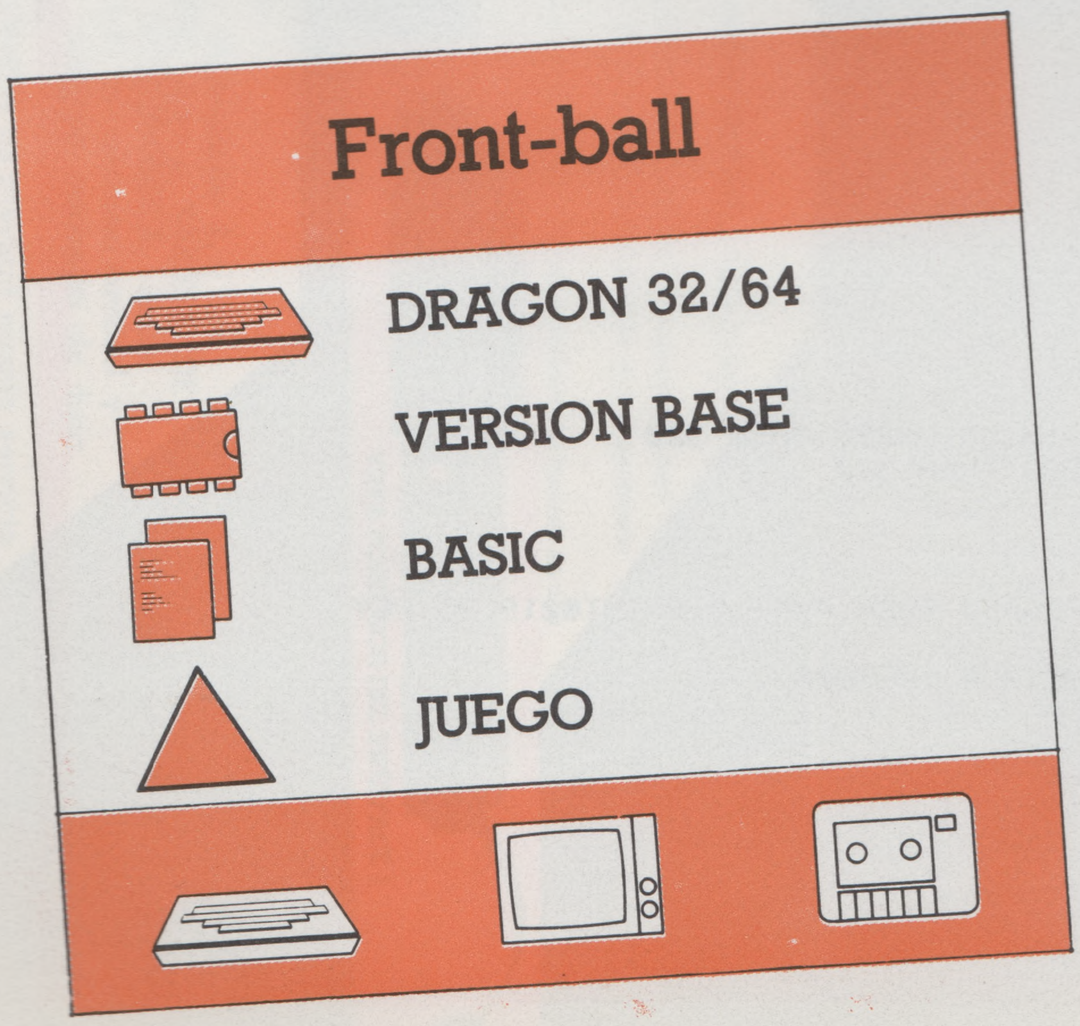
El desplazamiento de la raqueta a derecha e izquierda lo consigue el jugador accionando las teclas «M» y «N», respectivamente; la gestión de ambos movimientos corre a cargo de las líneas de programa 250 a la 270 (acción sobre la tecla «M») y de la 280 a la 300 (acción sobre la tecla «N»).

El control de las evoluciones de la pelota corre a cargo de distintas zonas del programa, según la naturaleza del desplazamiento al que esté sometida. Así, por ejemplo, del movimiento de subida oblícua hacia la izquierda se ocupa el bloque de líneas 310-390, de la subida vertical el bloque 400-460 y del movimiento de subida oblícua hacia la derecha las líneas 470 a la 550.

La zona 560 a la 590 incluye las instrucciones adecuadas para acumular la suma de puntos obtenidos a lo largo del juego, mientras que las líneas 600 a las 640 se ocupan de dar por concluida la partida, después de jugadas las diez pelotas disponibles, y de mostrar la puntuación final, además de brindar al jugador la posibilidad de empezar una nueva partida.

Dentro del bloque delimitado por las líneas 310 a 390, está presente una subrutina (líneas 350 y 360) que entra en actividad cuando la pelota sube en dirección oblícua hacia la izquierda y choca contra la pared, produciéndose un cambio en la dirección de desplazamiento que es asumido por la línea 480.

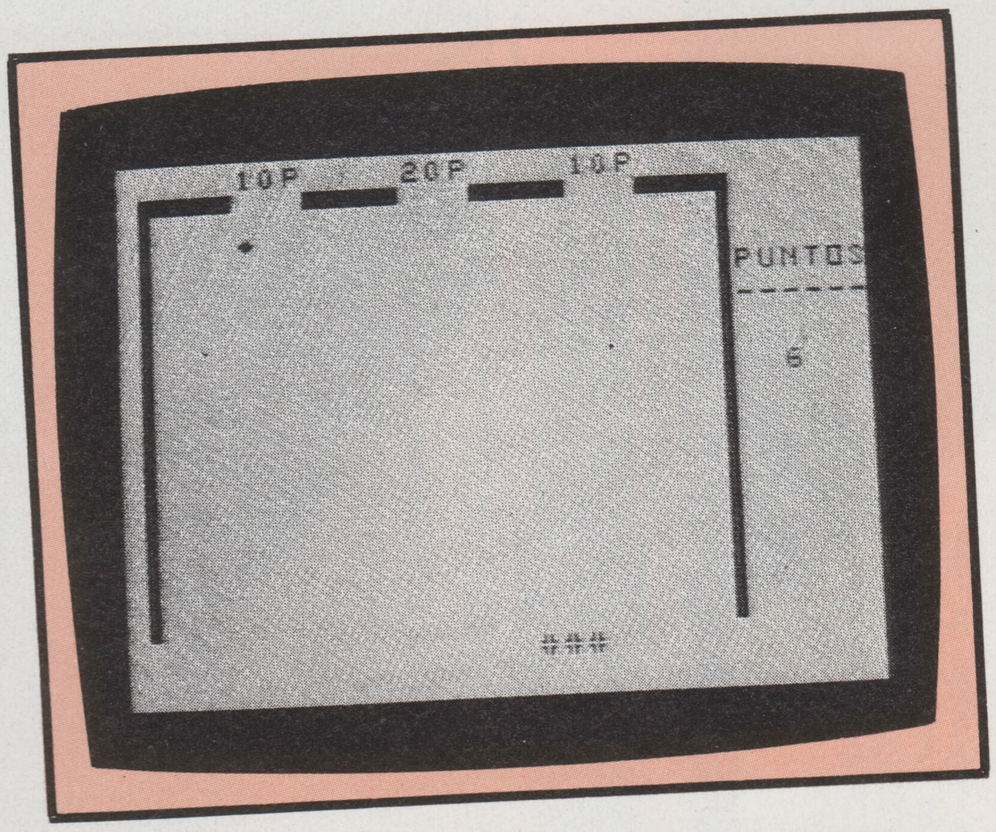
Y ya poco queda por añadir excepto, tal vez, recordarle que el desplazamiento de la raqueta se consigue accionando las teclas M o N, según desee hacerlo a derecha o izquierda.



Tu Micro N.º 3/4, Pág. 49

TABLA DE VARIABLES

A	Punto de la raqueta.
B	Función RND.
D	Punto de la pelota (D=B+C).
E$	Instrucción INKEY$.
F	Contador de tiradas.
G$	INPUT línea 620.
M	Movimiento hacia la derecha.
N	Movimiento hacia la izquierda.
P	Puntos obtenidos.



LISTADO

```
10 REM *-FROMPELOTA-*
20 REM *-I.MORAGREGA V.-*
30 A=459:P=0:F=0
40 CLS:PRINT TAB(5);"10P";TAB(12);"20P";TAB(19);"10P"
50 FOR Z=3 TO 27:SET(3,Z,0):SET(50,Z,0):NEXT Z
60 FOR Z=3 TO 50:SET(Z,3,0):NEXT Z
70 FOR Z=37 TO 51 STEP 7:PRINT@Z,STRING$(3,143);:NEXT Z
80 PRINT@122,"PUNTOS";
90 PRINT@154,"------";
100 PRINT@A,"###"
110 B=RND(21):C=66:D=B+C
120 F=F+1:IF F>10 THEN 600
130 GOSUB 210
140 E$=INKEY$
150 IF E$="M" THEN GOSUB 250
160 IF E$="N" THEN GOSUB 280
170 IF D=A THEN 310
180 IF D=A+1 THEN 400
190 IF D=A+2 THEN 470
200 GOTO 130
210 PRINT@D,"*";:FOR Z=1 TO 50:NEXT Z
220 PRINT@D," ";:D=D+32
230 IF D>473 THEN 110
240 RETURN
250 PRINT@A,"   ":A=A+1
260 IF A=473 THEN A=A-1
270 PRINT@A,"###":RETURN
280 PRINT@A,"   ":A=A-1
290 IF A=447 THEN A=A+1
300 PRINT@A,"###":RETURN
310 GOSUB 590
320 PRINT@D,"*";:PRINT@A,"###"
330 FOR Z=1 TO 50:NEXT Z
340 PRINT@D," ";:D=D-33
350 IF D=417 OR D=418 OR D=386 OR D=354 OR D=322 OR D=290 OR D=258 THEN 480
360 IF D=226 OR D=194 OR D=162 OR D=130 OR D=98 THEN 480
370 GOSUB 560
380 IF D<64 THEN 110
390 GOTO 320
400 GOSUB 590
410 PRINT@D,"*";:PRINT@A,"###"
420 FOR Z=1 TO 50:NEXT Z
430 PRINT@D," ";:D=D-32
440 GOSUB 560
450 IF D<65 THEN 110
460 GOTO 410
470 GOSUB 590
480 PRINT@D,"*";:PRINT@A,"###"
490 FOR Z=1 TO 50:NEXT Z
500 PRINT@D," ";:D=D-31
510 IF D=440 OR D=408 OR D=376 OR D=344 OR D=312 OR D=280 THEN 320
520 IF D=248 OR D=216 OR D=184 OR D=152 OR D=120 THEN 320
530 GOSUB 560
540 IF D<64 THEN 110
550 GOTO 480
560 IF D=37 OR D=38 OR D=39 OR D=51 OR D=52 OR D=53 THEN P=P+10:PRINT@219,P;:RET
URN
570 IF D=44 OR D=45 OR D=46 THEN P=P+20:PRINT@219,P;:RETURN
580 RETURN
590 P=P+2:PRINT@219,P;:RETURN
600 CLS(3):PRINT@102,"HA OBTENIDO";P;"PUNTOS.";
610 PRINT@195,"SI DESEA AUMENTAR SU PUNTO";
620 PRINT@234,"PULSE ENTER";:INPUT G$
630 IF G$<>"" THEN 620
640 GOTO 30
```

END

VEN A LA TIENDA Nº1 DE MADRID

REMSHOP nº1

HORARIO: 10 MAÑANA A 10 NOCHE DE LUNES A SABADO

REM Somos profesionales.
REM Nace para dar mejor servicio.
REM Como oferta de lanzamiento, entrega un Bono del 10% del importe de su compra para adquirir nuevo material o programas.
REM CAMBIO acepta equipos de 2.ª mano al adquirir otro nuevo
REM Consúltanos tus necesidades.

REM

RENOVACION EN MARCHA, S. A.
c/. Espronceda, 34-2.º int. - MADRID-3
Teléfono **(91) 441 24 78**

REM SHOP 1
c/. Galileo, 4 - MADRID-15
Teléfono **(91) 445 28 08**

HARD SPECTRAVIDEO

1 SV 328	76.000 Ptas.
1 Unidad doble discos con controlador y salida centronics	148.000 Ptas.
1 Joystick-3	3.654 Ptas.
1 Impresora GP-500	49.900 Ptas.
TOTAL	277.554 Ptas.

PRECIO PTAS. TOTAL 249.798 PTAS.

HARD ATARI

1 800 XI (64 K)	85.500 Ptas.
1 Disc Drive	99.500 Ptas.
1 Tableta Gráfica	15.900 Ptas.
1 Cassette	16.900 Ptas.
2 Joystick doble con mando a distancia	6.800 Ptas.
TOTAL	224.600 Ptas.

PRECIO PTAS. TOTAL 202.140 PTAS.

HARD ORIC

1 Oric Atmos	59.500 Ptas.
1 Disco 190 K	89.900 Ptas.
1 Impresora GP-500	49.900 Ptas.
TOTAL	199.300 Ptas.

PRECIO PTAS. TOTAL 179.370 PTAS.

HARD MULTITECH

1 MPF II	62.500 Ptas.
1 Joystick	2.500 Ptas.
1 Magnetofón	7.500 Ptas.
1 Impresora GP-500	49.900 Ptas.
TOTAL	122.400 Ptas.

PRECIO PTAS. TOTAL 110.160 PTAS.

SOFT SPECTRAVIDEO

Music Mentor	5.995 Ptas.
Frantic Freddy	3.995 Ptas.
Flipper Sliper	3.995 Ptas.
Armoured Assault	1.995 Ptas.
Introduction to Basic	1.995 Ptas.
TOTAL	17.975 Ptas.

PRECIO PTAS. TOTAL 16.177 PTAS.

SOFT ATARI

Pole Position	8.450 Ptas.
Tennis	8.450 Ptas.
Pac Man	8.450 Ptas.
Basketball	5.925 Ptas.
Jungla Hunt	8.450 Ptas.
TOTAL	39.725 Ptas.

PRECIO PTAS. TOTAL 35.752 PTAS.

5 SOFT ORIC

Zorgons	2.300 Ptas.
Xenon	2.800 Ptas.
Ajedrez	2.800 Ptas.
Ghost Gobbler	2.100 Ptas.
Green Toad	1.900 Ptas.
TOTAL	11.900 Ptas.

PRECIO PTAS. TOTAL 10.710 PTAS.

5 SOFT MULTITECH

Falcons/Fenix	1.700 Ptas.
Panic	1.700 Ptas.
Suicide	1.700 Ptas.
Sabotage/Tiroteo	1.700 Ptas.
Asteroides	1.700 Ptas.
TOTAL	8.500 Ptas.

PRECIO PTAS. TOTAL 7.650 PTAS.

REM NOTICIAS

REM FRANCHISING
Si quieres montar tu propia minitienda de informática, o una tienda especializada, envianos tu dirección y recibirás información completa.

REM SOFT
Pagamos 20% de royalties de programas originales.

REM DETALL
Si quieres vender nuestros productos, envíanos tu dirección y recibirás puntual información.

REM CLUB
Reuniones: Intercambio de programas: Revistas, Libros: Unete a nosotros.

REM INFORMATICA
Tenemos también SPECTRUM y COMMODORE Hard y Soft.

BOLETIN DE PEDIDO

Nombre y apellidos .....

Dirección y Teléfono .....

Deseo recibir más información .....

Deseo adquirir .....

Precio total (incluye 300 Ptas. de gastos de envío)

Giro Postal ☐ Giro Telegráfico ☐ Transferencia Bancaria
Ingreso en cuenta 3769.8 BANCO DE BILBAO. Ríos Rosas, 44 - MADRID-3
Talón adjunto ☐ Talón conformado adjunto

Tarjeta VISA número .....

Fecha Caducidad ..... Firma .....

REM c/. Espronceda, 34-2.º int. - MADRID-3

Las cuatro en raya



Pon en práctica tu habilidad: concéntrate, juega... ¡y gana! Esta es la síntesis más dinámica del objetivo del programa que describimos en los próximos párrafos.

En esta ocasión el juego programado no exige una detallada relación de las reglas, puesto que resulta de lo más familiar. En cualquier caso, para los olvidadizos, aquí está la descripción somera del juego.

Dos jugadores, dos grupos de fichas, cada uno de distinto color y un tablero de juego. Para vencer el jugador debe alinear cuatro fichas de su color en línea recta, bien sea en sentido horizontal, vertical o diagonal. En turnos alternativos, cada uno de los contendientes introducirá una ficha propia indicando al ordenador el número de columna en la que desea realizarlo. Emulando al tablero de juego convencional, la ficha se deslizará a lo largo de la columna hasta ocupar el casillero inferior que se encuentre libre.

El programa cuyo listado se adjunta reemplaza las fichas y el tablero de juego por la pantalla de un televisor doméstico y, a su vez, convierte al ZX-SPECTRUM en un árbitro imparcial e insobornable.

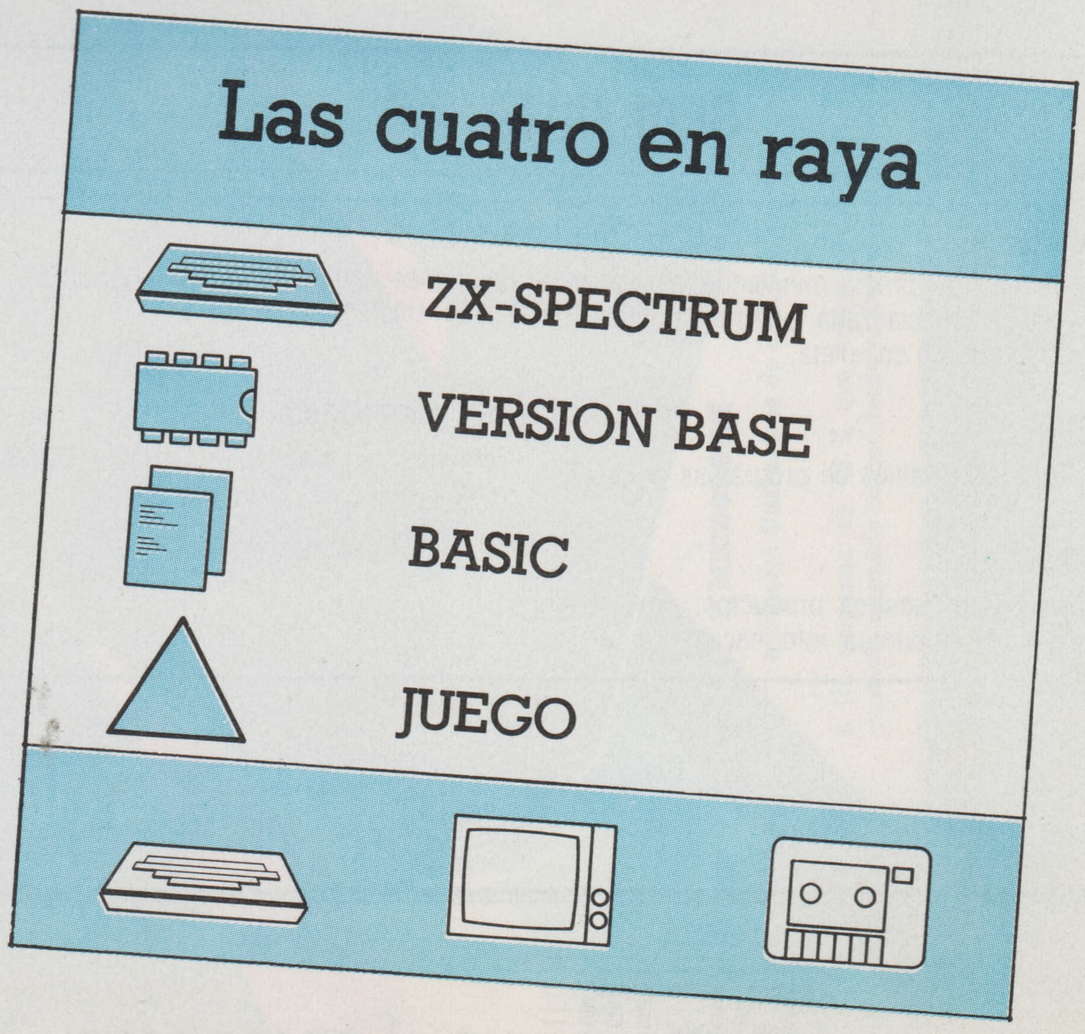
¡A jugar!

El juego empieza con una primera pantalla de captación de datos, a través de la que se piden los nombres de los oponentes cuyo ingenio y habilidad van a entrar en liza. Los nombres, tal como refleja el mensaje visualizado, no pueden exceder la longitud de nueve caracteres.

Una vez realizada esta operación, el ordenador diseña el tablero de juego, asigna un color de ficha a cada jugador y determina aleatoriamente a qué oponente corresponde realizar la primera jugada.

¡Empieza la partida! El jugador agraciado con el primer turno debe pulsar la tecla correspondiente al número de columna en la que desea introducir su ficha; acto seguido, el oponente hará lo propio... y así, turno a turno, hasta concluir la partida.

Cada vez que uno de los jugadores introduce una ficha, el programa comprueba si ha completado un conjunto de cuatro fichas alineadas. Cuando alguno de los conten-



Pág. 52, Tu Micro N.º 3/4

LISTADO

```
  10 REM P.M. DEL MORAL
  20 REM "LAS CUATRO EN RAYA"
  30 BORDER 5: PAPER 5: INK 0: POKE 23658,8
  50 DATA 0,60,126,126,126,126,60,0,255,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,128,128,12
8,128,128,128,128,128,1,1,1,1,1,1,1,255,128,128,128,128,128,128,128,255
  60 FOR I=0 TO 47
  70 READ C: POKE USR "A"+I,C
  80 NEXT I
  90 CLS : DIM T(6,7)
 100 FOR I=1 TO 6
 110 FOR J=1 TO 7
 120 LET T(I,J)=0
 130 NEXT J
 140 NEXT I
 150 PRINT AT 3,5;"**********************"'TAB 5;"* LAS CUATRO EN RAYA *"'TAB 5;
"**********************"
 155 PRINT AT 14,5;"(MAXIMO 9 CARACTERES)"
 160 PRINT AT 12,1;"TECLEE EL NOMBRE DEL JUGADOR 1"
 170 INPUT  LINE P$
 180 IF LEN P$>9 THEN  GO TO 170
 190 BEEP .25,0: PRINT AT 12,1;"TECLEE EL NOMBRE DEL JUGADOR 2"
 200 INPUT  LINE S$
 210 IF LEN S$>9 THEN  GO TO 200
 211 BEEP .25,0: DIM T$(2,9)
 212 LET T$(1)=P$: LET T$(2)=S$
 220 CLS
 230 PRINT AT 7,10;"__________"
 240 FOR I=8 TO 13
 250 PRINT AT I,9;"C"; PAPER 6;" "; PAPER 2; INK 7;" AAAAAAA "; PAPER 6;" ";: PR
INT "D"
 260 NEXT I
 270 PRINT AT 14,7;"C"; INK 6; PAPER 0;"   1234567   ";: PRINT "D";AT 15,8;"BB
BBBBBBBBBBBB";AT 13,8;"_E";AT 13,21;"F_"
 280 PRINT AT 20,5-INT (LEN P$/2); PAPER 4;P$;AT 20,26-INT (LEN S$/2); PAPER 1;
INK 7;S$
 290 LET M=42
 320 LET R=INT (RND*2)
 330 LET R=R+1: IF R=3 THEN  LET r=1
 335 FOR I=1 TO 10: BEEP .01,20: NEXT I
 340 GO SUB 2000
 350 PAUSE 0
 360 IF INKEY$<"1" OR INKEY$>"7" THEN  GO TO 360
 370 LET C=VAL INKEY$
 440 IF T(1,C)<>0 THEN  FOR I=1 TO 6: PRINT AT 6,11+C; OVER 1;"*": BEEP .5,20: N
EXT I: LET M=M+1: GO TO 350
 450 IF R=1 THEN  LET J=4: GO TO 470
 460 LET J=1
 470 FOR I=6 TO 1 STEP -1
 480 IF T(I,C)=0 AND R=1 THEN  LET T(I,C)=1: GO SUB 3000: GO TO 510
 490 IF T(I,C)=0 AND R=2 THEN  LET T(I,C)=2: GO SUB 3000: GO TO 510
 500 NEXT I
 512 FOR K=1 TO 4
 513 LET Q=0
 515 FOR L=K TO K+3
 520 IF T(I,L)=R THEN  LET Q=Q+1
 522 NEXT L
 525 IF Q=4 THEN  GO SUB 4000: GO SUB 5000
 530 NEXT K
 540 FOR K=1 TO 3
 550 LET Q=0
 560 FOR L=K TO K+3
 570 IF T(L,C)=R THEN  LET Q=Q+1
 580 NEXT L
 590 IF Q=4 THEN  GO SUB 4000: GO SUB 5000
 600 NEXT K
 610 LET Y$=STR$ T(I,C): LET I1=I: LET C1=C
 620 LET I1=I1-1: LET C1=C1+1: IF C1>7 OR I1<1 THEN  GO TO 640
 630 LET Y$=Y$+STR$ T(I1,C1): GO TO 620
 640 LET I1=I: LET C1=C
 650 LET I1=I1+1: LET C1=C1-1: IF I1>6 OR C1<1 THEN  GO TO 670
 660 LET Y$=STR$ T(I1,C1)+Y$: GO TO 650
 670 IF LEN Y$<4 THEN  GO TO 700
 680 LET Z$="": FOR K=1 TO 4: LET Z$=Z$+STR$ R: NEXT K: FOR K=1 TO LEN Y$-3: IF
Y$(K TO K+3)<>Z$ THEN  NEXT K: GO TO 700
 690 GO SUB 4000: GO TO 999
 700 LET Y$=STR$ T(I,C): LET I1=I: LET C1=C
 710 LET I1=I1-1: LET C1=C1-1: IF C1<1 OR I1<1 THEN  GO TO 730
 720 LET Y$=STR$ T(I1,C1)+Y$: GO TO 710
 730 LET I1=I: LET C1=C
 740 LET I1=I1+1: LET C1=C1+1: IF I1>6 OR C1>7 THEN  GO TO 760
 750 LET Y$=Y$+STR$ T(I1,C1): GO TO 740
 760 IF LEN Y$<4 THEN  GO TO 330
 770 LET Z$="": FOR K=1 TO 4: LET Z$=Z$+STR$ R: NEXT K: FOR K=1 TO LEN Y$-3: IF
Y$(K TO K+3)<>Z$ THEN  NEXT K: GO TO 330
```

```
 780 GO SUB 4000: GO TO 9999
 997 IF s=0 THEN  GO SUB 5000
1010 LET S=S-1: PRINT AT 18,26;S;" ": RETURN
2000 PRINT AT 3,5;"TE TOCA ";T$(R): RETURN
3000 PRINT AT 7+I,11+C; INK J; PAPER 2;"A": LET M=M-1: IF M=1 THEN  GO SUB 6000:
 GO SUB 5000
3010 RETURN
4000 PRINT AT 6,7;"HAS GANADO ";T$(R)
5000 PRINT AT 16,2;"QUIERES JUGAR OTRA VEZ (S/N)?"
5010 IF INKEY$="S" THEN  GO TO 90
5020 IF INKEY$<>"N" THEN  GO TO 5000
6000 FOR I=1 TO 6: PRINT AT 6,14; OVER 1;"TABLAS": BEEP .5,0: NEXT I: RETURN
```

END

dientes lo consigue —¡cuatro en raya!—, aparecerá en la pantalla un cartel con el nombre del flamante ganador.

La diversión puede continuar. El propio ordenador preguntará en este punto si se desea empezar una nueva partida: basta con responder con una «S» en caso afirmativo o con una «N» si la revancha se pospone para mejor ocasión.

Y por último...

El programa posee un control sobre las columnas, evaluando cualquier posible error en la introducción de fichas. Por ejemplo, si uno de los jugadores intentara colocar su ficha en una columna ya completa, el ordenador detectará el error y avisará a los contendientes emitiendo una señal acústica y colocando un asterisco sobre la columna en cuestión.

A lo largo del desarrollo del juego son varios los sonidos que emite el ordenador revelando determinadas situaciones, una de ellas es el cambio de turno de jugada.

El programa está concebido para el ZX-SPECTRUM en cualquiera de sus versiones, 16 ó 48 K. Una vez introducido el programa en el SPECTRUM —tarea ardua puesto que hay que realizarla a golpe de teclado— es conveniente almacenarlo en casete antes de su ejecución; para ello hay que recurrir al comando SAVE con el siguiente formato:

SAVE b «4 EN RAYA».

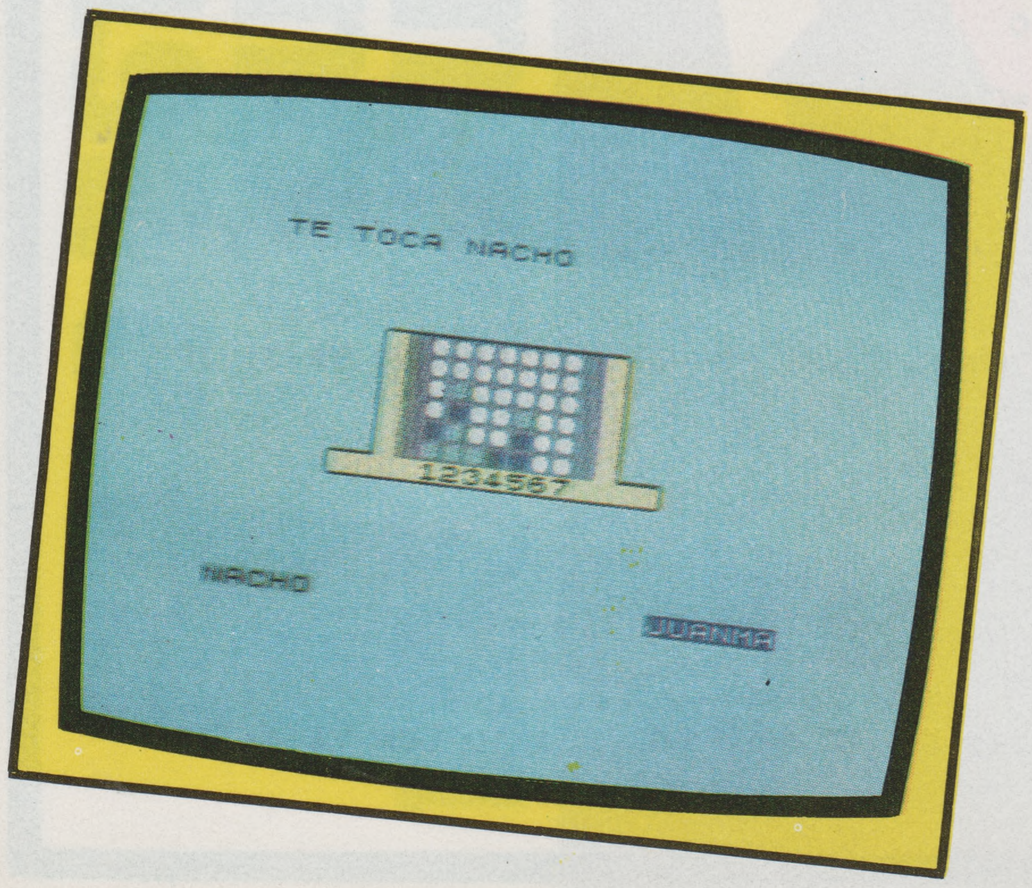
tu Micro

ESTRUCTURA DEL PROGRAMA

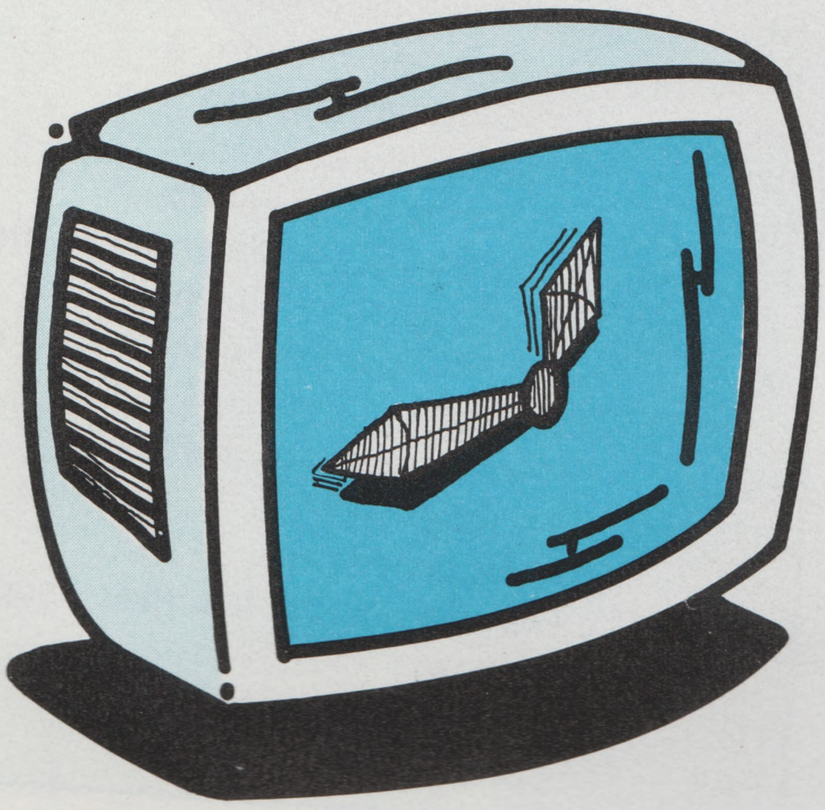
30-80	Definición de gráficos.
90-140	Inicializa la tabla «T» a ceros.
150-212	Diseña la primera pantalla y pide los nombres de los jugadores.
220-280	Diseña la segunda pantalla.
320-370	Adjudica el turno de juego y pide el número de columna.
440-460	Depura jugada no válida.
470-500	Guarda en la tabla «T» el valor correspondiente.
512-530	Comprueba la fila.
540-600	Comprueba la columna.
610-780	Examina las diagonales.

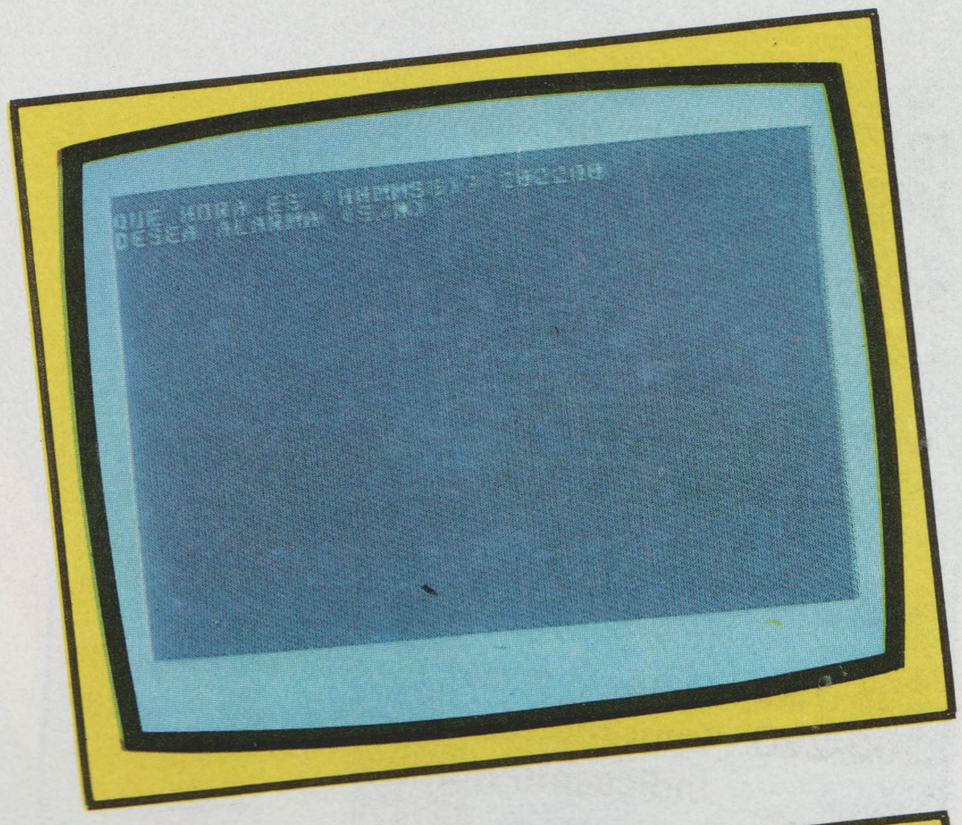
TABLA DE VARIABLES

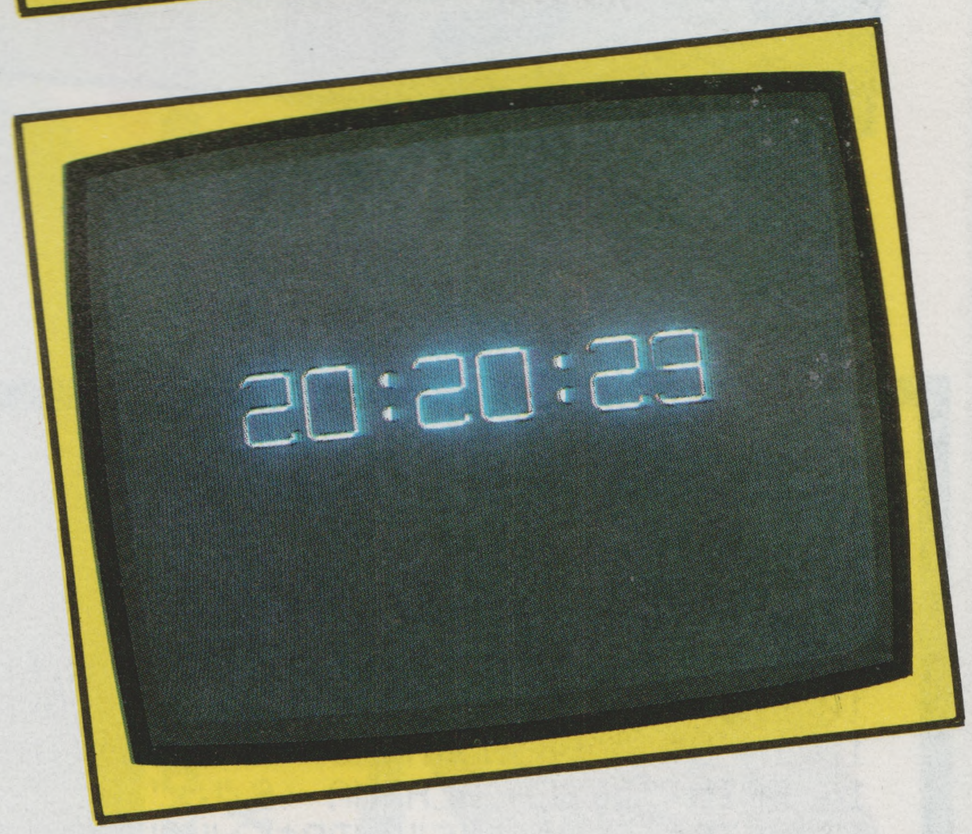
C	Número de columna introducido.
I, J, K, L	Variables auxiliares.
R	Identifica al jugador que acaba de realizar su jugada.
Q	Comprobación de horizontales y verticales.
C1, I1	Parámetros de comprobación de diagonales.
P $	Nombre del primer jugador.
S $	Nombre del segundo jugador.
X $, Y $, Z $	Variables para la comprobación de diagonales.
T S(2)	Tabla que contiene los nombres de los jugadores.
T (6, 7)	Tabla que contiene los valores del tablero de juego.



(1) El tradicional juego de «Las cuatro en raya» trasladado de la mesa de juego a la pantalla del televisor, por obra y gracia de un sencillo programa para el ZX-SPECTRUM.

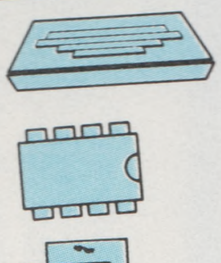






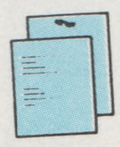
Crono-alarm

Crono-alarm

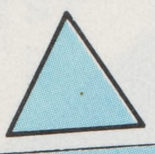


COMMODORE-64

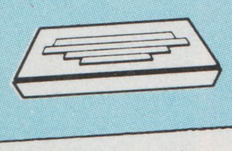
VERSION BASE

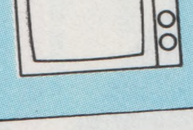


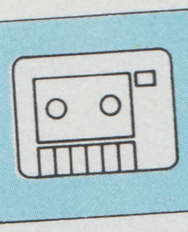
BASIC

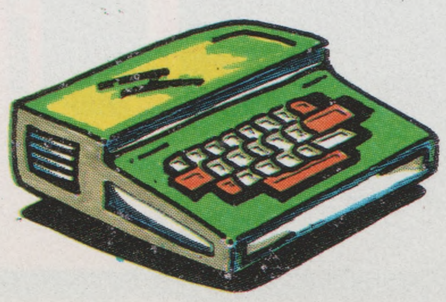


DOMESTICO









Si se le pegan las sábanas por la mañana, no dude en echarle un vistazo a «Rolex de oro con despertador».

El programa que le presentamos convierte la pantalla de su televisor en un auténtico reloj/despertador panorámico, con selector de brillo, contraste y volumen... y todos los demás extras que suele incorporar un receptor de televisión.

Por supuesto, el ordenador no lo sabe todo; así que antes de empezar a disfrutar de su buen hacer, habrá que comunicarle la hora y, en el caso de utilizar la posibilidad de alarma, decirle también a qué hora debe «sonar», para despegarnos de las sábanas o avisarnos de que empieza el partido en la segunda cadena.

El fundamento del programa no deriva de la aplicación de técnicas muy sofisticadas. Sumarialmente, toda su sapiencia —¡displicente y exacta, no cabe la menor duda!— se reduce a extraer de la variable TIME$, una y otra vez, la configuración horaria que corresponda y proyectarla en la pantalla del televisor. Para esta frenética actividad, cuenta con el auxilio de la matriz unidimensional N$(10) que contiene los 10 «superdígitos» que se utilizarán para construir la indicación horaria. tuMicro

Tu Micro N.º 3/4, Pág. 55

ESTRUCTURA DEL PROGRAMA

10	POKEs de sonido.
15-20	Entrada de hora, alarma y colores.
30-120	Define los dígitos gigantes.
130	Define los dos puntos de separación.
140-150	Principio del bucle principal. Separa los seis dígitos horarios en variables.
192-195	Examina si debe sonar la alarma.
196-198	Detecta si debe parar la alarma.
200-220	Imprime los superdígitos con la indicación horaria.

TABLA DE VARIABLES

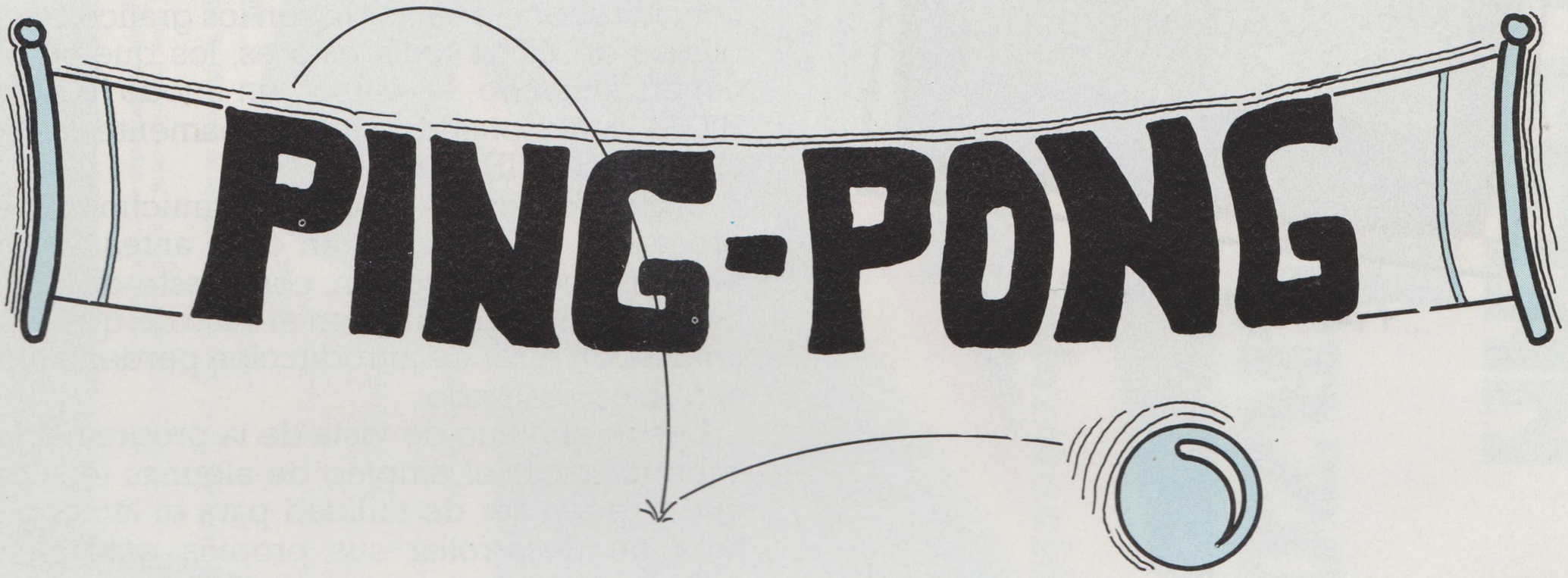
S	Puntero de sonido.
HD	Decenas de hora.
HU	Unidades de hora.
MD	Decenas de minuto.
MU	Unidades de minuto.
SD	Decenas de segundo.
SU	Unidades de segundo.
F	Bandera (0: puede sonar la alarma, 1: corta la alarma).
W	Variable para la generación del sonido.
H$	Hora para pasar a TI$.
R$	Variable GET.
AL$	Hora de alarma.
N$ (10)	Caracteres de los diez «superdígitos».
P$	Puntos de separación.
PA$	Variable GET.

LISTADO

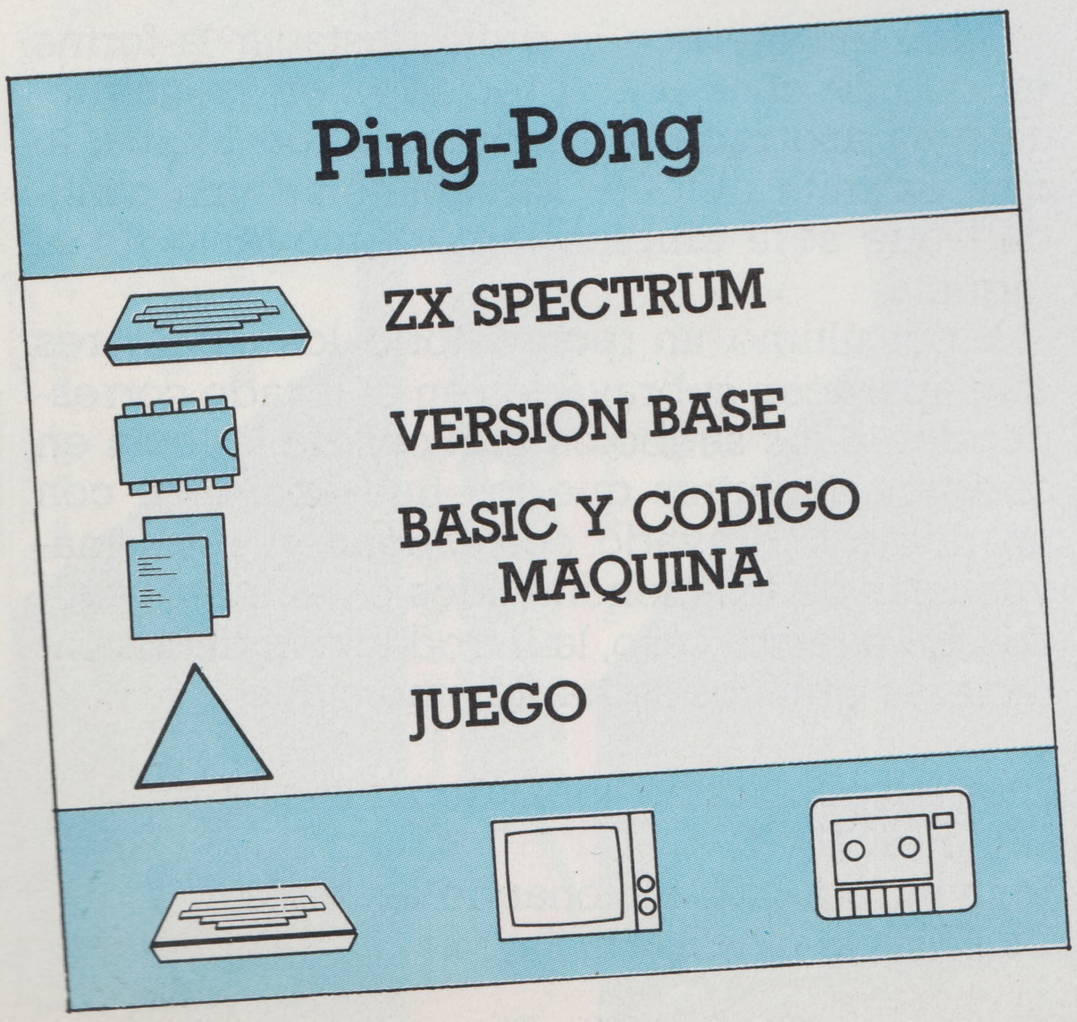
```
1 REM CRONOALARM # CBM 64 (C) A.CARVAJAL
10 PRINT"■":S=54272:POKES,200:POKES+1,40:POKES+24,15:POKES+6,240:POKES+4,16
15 INPUT "QUE HORA ES (HHMMSS)";H$:TI$=H$
16 PRINT"DESEA ALARMA (S/N)"
17 GETR$:IFR$<>"S"ANDR$<>"N"THEN17
18 IFR$="N"THEN20
19 INPUT "HORA DE ALARMA (HHMMSS)";AL$
20 POKE 53281,0:POKE53280,0:PRINT"E■"
30 N$(1)="  Q!!!!   ! Q!!!!   ! Q!!!!   ! Q!!!! ┌┴┐■■■■"
40 N$(2)="┌─┐Q!!!!  !Q!!!!┌─┐Q!!!!!   Q!!!! └─┐■■■■"
50 N$(3)="┌─┐Q!!!!  !Q!!!! ─┤Q!!!!   !Q!!!! └─┐■■■■"
60 N$(4)="│ │Q!!!!!  !Q!!!!└─┤Q!!!!   !Q!!!!   └┐■■■■"
70 N$(5)="┌─┐Q!!!!!  Q!!!!└─┐Q!!!!   !Q!!!! └─┐■■■■"
80 N$(6)="┌─┐Q!!!!!  Q!!!!├─┐Q!!!!!  !Q!!!! └─┐■■■■"
90 N$(7)="┌─┐Q!!!!  !Q!!!!  ─┤Q!!!!   !Q!!!!   └┐■■■■"
100 N$(8)="┌─┐Q!!!!!  !Q!!!!├─┤Q!!!!!  !Q!!!! └─┐■■■■"
110 N$(9)="┌─┐Q!!!!!  !Q!!!!└─┤Q!!!!   !Q!!!! └─┐■■■■"
120 N$(0)="┌─┐Q!!!!!  !Q!!!!!  !Q!!!!!  !Q!!!! └─┐■■■■"
130 P$="   Q!!! ■ Q!!!   Q!!! ■ Q!!!   ■■■■"
140 HD=VAL(MID$(TI$,1,1))
150 HU=VAL(MID$(TI$,2,1))
160 MD=VAL(MID$(TI$,3,1))
170 MU=VAL(MID$(TI$,4,1))
180 SD=VAL(MID$(TI$,5,1))
190 SU=VAL(MID$(TI$,6,1))
192 IFTI$<>AL$THENF=0
195 IFF=0THENIFTI$=AL$THENW=17
196 GETPA$
197 IFPA$=" " THEN W=16:F=1
198 POKES+4,W
200 FORI=0TO7:PRINT:NEXT
210 PRINT TAB(4)N$(HD)N$(HU)P$N$(MD)N$(MU)P$N$(SD)N$(SU)
220 PRINT"■■■■■■■■■":GOTO 110

READY.
```

END



Hace ya algunos años, cuando empezaron a salir a la calle los primeros vídeo-juegos, el popular tenis de mesa olvidó las raquetas, red y pelota para saltar a las pantallas de vídeo de los bares y billares.

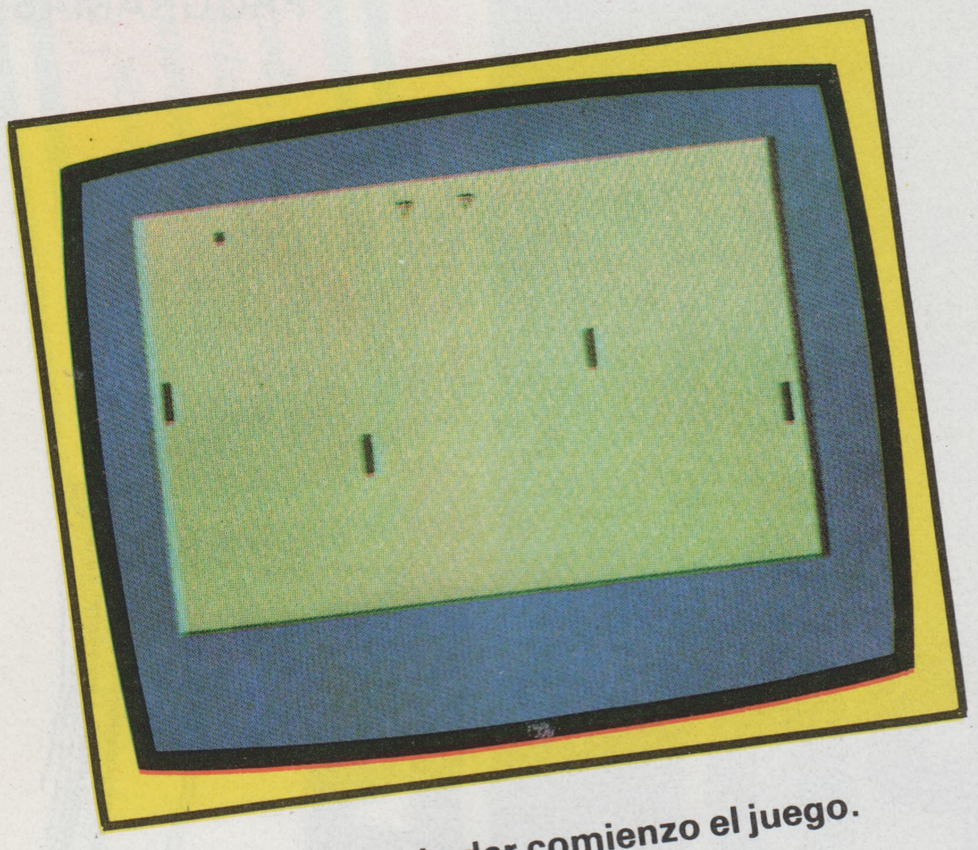


En aquella época, reciente en el tiempo aunque prehistórica en la evolución de los vídeo-juegos, era necesario el delicado y prolongado trabajo de un ingeniero electrónico, soldando una nube de transistores, resistencias y condensadores, para llegar a construir un aparato que jugaba al ping-pong.

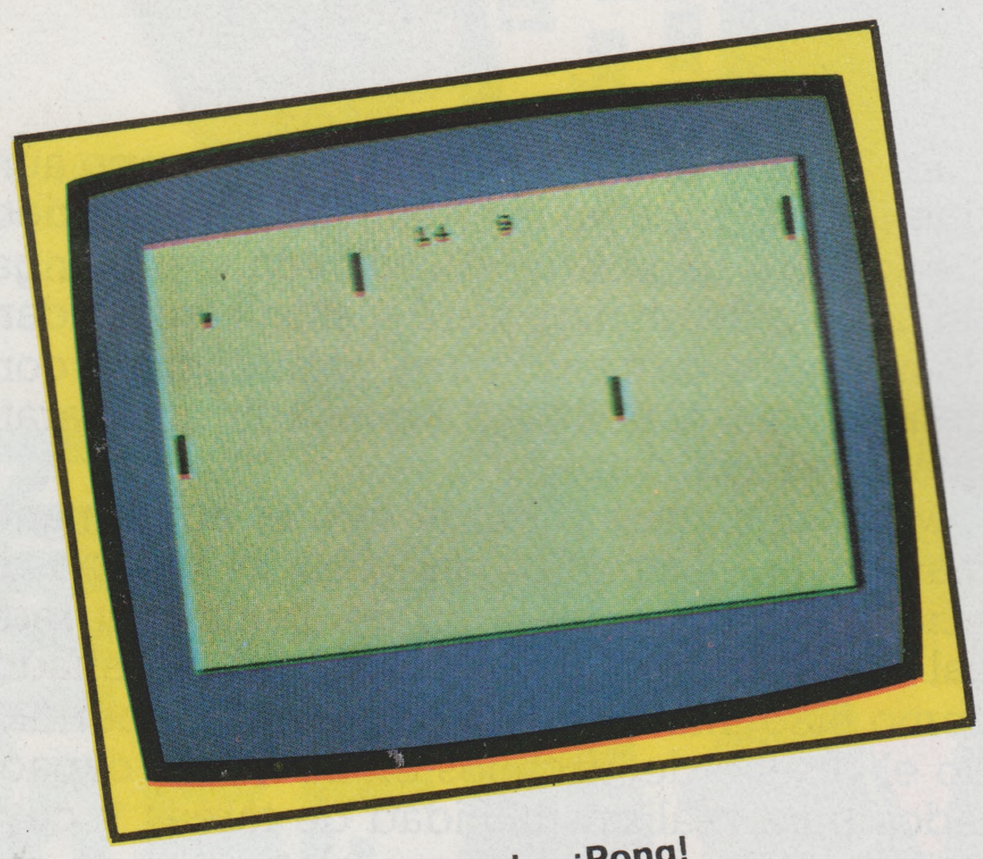
Más adelante, a medida que el apasionante mar de la informática anegaba el tiempo de ocio, las cosas cambiaron de forma casi radical. La moderna concepción del ordenador como máquina de utilidad general, ha permitido el diseño de circuitos electrónicos capacitados para realizar infinidad de tareas y cuya aplicación específica viene determinada por un simple «programa»

Una prueba de la vertiginosa evolución ocurrida la tenemos en este mismo artículo, en el que el tradicional video-juego del ping-pong se convierte en un simple programa para un ordenador doméstico.

Creemos no pecar de optimistas al suponer que las reglas del ping-pong son archiconocidas; en cualquier caso, vamos a resumirlas particularizándolas al caso de nuestro ping-pong doméstico. Dos contrincantes luchan por hacer pasar una bola a través de la línea de fondo del campo contrario, a la vez que evitan que lo haga por el suyo. Para ello, cada jugador dis-



... Y ya puede dar comienzo el juego.



¡Ping!... ¡Pong!

pone de dos teclas para el desplazamiento de la raqueta. El primer jugador utilizará las teclas 1 y 2, mientras que el segundo emplazará las teclas 9 y 0.

La introducción del programa en el SPECTRUM no presenta ninguna complicación especial; si acaso, recordar que los símbolos gráficos que aparecen en el listado con un subrayado simple corresponden a los gráficos de la tecla 5 (GRAPHICS 5); a su vez, los que aparecen con subrayado doble son los gráficos cambiados de dicha tecla esto es, los que se obtienen situando el cursor en modo «GRAPHICS» y accionando simultáneamente las teclas CAPS-SHIFT y 5.

Una precaución importante, y mucho más en programa, es grabarlos en cinta antes de lanzar la primera ejecución; como este que contienen código máquina; en el caso de que existiera algún error de introducción perdería todo el trabajo realizado.

Desde el punto de vista de la programación cabe destacar el empleo de algunas técnicas que pueden ser de utilidad para el lector a la hora de desarrollar sus propios programas.

En primer lugar, para agilizar el juego, se utiliza una subrutina en código máquina que se ocupa de visualizar la pelota. Esta subrutina está almacenada a partir de la dirección 31349 y su efecto es imprimir la primera vez, y borrar la segunda, un cuadrado de 4 por 4 pixels. La precisión de las coordenadas es el doble que las correspondientes a un PRINT AT usual y sus valores se encontrarán almacenados en las posiciones de memoria 23300 y 23301. Esta subrutina es mucho más rápida que un PRINT, de ahí que el lector pueda encontrarle utilidad a la hora de agilizar sus propios ptragramas.

Otro truco interesante es el reflejado en la instrucción que se encuentra en la línea 105:

GOTO 500 — (400*a$="s" OR a$="S")

Esta instrucción simula el efecto de:

IF a$="s" THEN GOTO 60 ELSE GOTO 500

A su vez, GOTO 500 en este programa equivale a un STOP.

También merece la pena destacar la forma en que se chequea el teclado con la instrucción IN asociada a la comparación lógica, lo que permite obtener directamente una cantidad que será empleada para mover o no la raqueta.

Y por último, un recordatorio: los caracteres que aparecen subrayados en el listado corresponden a los símbolos gráficos de la tecla en cuestión, mientras que los que aparecen con un doble subrayado corresponden a las mayúsculas de los mencionados caracteres gráficos. En nuestro caso, la introducción de los caracteres gráficos se hará como sigue:

5: se introducirá accionando las teclas GRAPHICS y 5.
5: se introducirá accionando las teclas GRAPHICS CAPS-SHIFT y 5.

LISTADO

```
  1 REM # XAVIER ALAMAN # PING-PONG # ZX SPECTRUM
 10 DATA 58,4,91,22,15,203,71,32,2,22,240,203,63,230,31,111,38,68,58,5,91,79,20
3,71,32,5,124,214,4,103,121,203,63,230,24,180,103,121,203,39,203,39,203,39,203,3
9,230,225,181,111,6,4,0,122,174,119,36,16,249,201
 20 FOR I=31349 TO 31408
 30 READ A
 40 POKE I,A
 50 NEXT I
 60 BORDER 1: PAPER 6: CLS : LET PUNT1=0: LET PUNT2=0: GO TO 100
 70 BEEP 1,10: CLS
 80 IF NOT PEEK 23300 THEN  LET PUNT2=PUNT2+1: GO TO 100
 90 LET PUNT1=PUNT1+1
100 PRINT AT 1,13;PUNT1;"  ";PUNT2
105 IF PUNT1>14 OR PUNT2>14 THEN  PRINT AT 10,12;"SE ACABO": BEEP 1,15: BEEP .5
,11: BEEP 1,11: INPUT "Quieres jugar otra vez? ";a$: GO TO 500-(440*(a$="s" OR a
$="S"))
110 RANDOMIZE : POKE 23300,31: POKE 23301,40+INT (RND*2)
120 LET I=1*SGN (RND-.5): LET J=1+INT (RND*2): LET X1=10: LET X2=X1
125 PRINT AT 10,0;"5";AT 9,0;"5";AT 10,21;"5";AT 9,21;"5";AT 10,10;"5";AT 9,10;
"5";AT 10,31;"5";AT 9,31;"5"
130 POKE 23300,I+PEEK 23300
140 POKE 23301,J+PEEK 23301
150 RANDOMIZE USR 31349
160 LET A=IN 63486=254
170 LET B=(IN 63486=253)*-1
180 LET X1=X1+(A AND X1<>21)+(B AND X1<>1)
190 IF A THEN  PRINT AT X1,0;"5";AT X1,21;"5";AT X1-2,0;" ";AT X1-2,21;" ": GO
TO 210
200 IF B THEN  PRINT AT X1-1,0;"5";AT X1-1,21;"5";AT X1+1,0;" ";AT X1+1,21;" "
210 LET X=X1+X1-PEEK 23301
220 IF (PEEK 23300=2 OR PEEK 23300=44) AND X>-3 AND X<4 THEN  LET I=1: LET J=1*
SGN X: BEEP .001,40: GO TO 240
230 IF NOT PEEK 23300 THEN  GO TO 70
240 IF PEEK 23301<2 OR PEEK 23301>41 THEN  LET J=-J: BEEP .005,20
250 RANDOMIZE USR 31349
260 POKE 23300,I+PEEK 23300
270 POKE 23301,J+PEEK 23301
280 RANDOMIZE USR 31349
290 LET C=IN 61438=254
300 LET D=(IN 61438=253)*-1
310 LET X2=X2+(C AND X2<>21)+(D AND X2<>1)
320 IF C THEN  PRINT AT X2,10;"5";AT X2,31;"5";AT X2-2,10;" ";AT X2-2,31;" ": G
O TO 340
330 IF D THEN  PRINT AT X2-1,10;"5";AT X2-1,31;"5";AT X2+1,10;" ";AT X2+1,31;"
"
340 LET X=X2+X2-PEEK 23301
350 IF (PEEK 23300=19 OR PEEK 23300=61) AND X>-3 AND X<4 THEN  LET I=-1: LET J=
1*SGN X: BEEP .001,40: GO TO 370
360 IF PEEK 23300=63 THEN  GO TO 70
370 IF PEEK 23301<2 OR PEEK 23301>41 THEN  LET J=-J: BEEP .005,20
380 RANDOMIZE USR 31349
390 GO TO 130
```

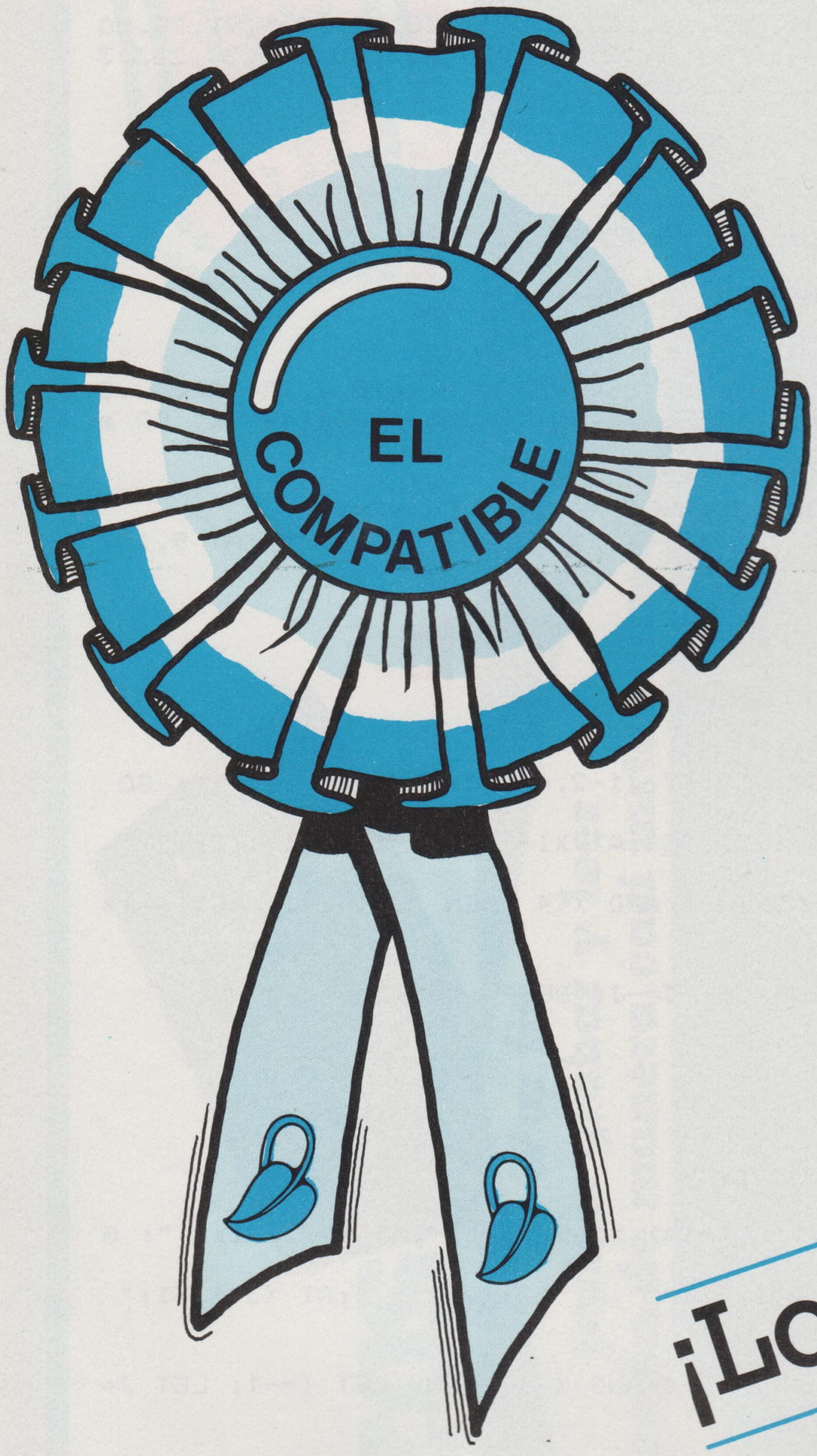
END

CUADRO DE VARIABLES

A	Como la anterior.
A$	Input de nueva partida.
B,C,D	Variable para la lectura del teclado.
I	Variable auxiliar de diversa utilidad.
J	Coordenada de fila de la pelota.
PUNT1	Puntos del jugador 1.
PUNT2	Puntos del jugador 2.
X	Variable para el cambio de sentido de la pelota.
X1	Fila de situación del jugador 1.
X2	Fila de situación del jugador 2.

ESTRUCTURA PROGRAMA

10	Código máquina.
20-50	Introducción del código máquina.
60	Inicialización de parámetros.
70-105	Contador de puntos, final de partida, etc.
110-125	Inicialización de pelota y raquetas.
130-290	Bucle principal.
130-240	Primer jugador.
240-290	Segundo jugador.



A lo largo de su historia, el hombre ha tenido que enfrentarse con enigmas irresolubles: la anhelada Piedra Filosofal que ningún alquimista consiguió encontrar jamás; el fabuloso tesoro de las Minas del Rey Salomón, perdidas aún en la impenetrabilidad de la selva africana, o el increíble Disolvente Universal que ningún químico ha llegado aún a sintetizar (¡aunque bien es cierto que la Coca Cola...!) Tal vez, el mito de la nueva era de la informática sea el programa BASIC compatible con todos los ordenadores, y ahí sí tenemos algo que decir... ¡Eureka!. ¡lo encontramos!

¡Lo conseguimos!



Erase una vez una calurosa tarde de estío y una oficina de redacción en la que el aire acondicionado seguía convencido de que estábamos en plena estación invernal... Poco a poco, las sillas iban cumpliendo su cometido con la llegada de los componentes del equipo de programación.

Nerviosos, cual visitantes de una tensa sala de maternidad, acompasaban su danza alrededor de la silla consumiendo cigarro tras cigarro o devorando un pedazo de cultura yanqui con sabor a menta, fresa o canela.

La tensión crecía paulatinamente hasta que Santi cruzó la puerta y arrojó sobre la mesa un paquete con 25 ejemplares del número 2 de TU MICRO. Con el golpe desapareció el hechizo y en una desaforada carga los presentes se arrojaron sobre el tan esperado montón de papel.

De nuevo se hizo un silencio expectante mientras todos y cada uno de los presentes

escudriñaban con avidez las páginas de la revista, esperando descubrir el fruto de su intelecto. Poco a poco las miradas iban posándose en la editorial e, indefectiblemente, se observaba en todos la misma reacción: primero una leve sonrisa (¡... habré leído mal!); a continuación la cara se iba tornando lentamente en una mueca de espanto, mientras rememoraban el árbol genealógico de Paco y, por último, el rostro rompía en una risa abierta, casi histérica, entre la incredulidad y el pavor...

¡Sí, habían leído bien! Veintiseis, ni uno más ni uno menos, eran los programas que había que pergeñar para el número doble de TU MICRO. En una primera reacción algunos pensaron en solicitar su ingreso voluntario e inmediato en la Legión Extranjera; otros, animados por Fernando, decidieron que su anhelo era ejercer de misioneros en la perdida selva amazónica, y los restantes empezaron a proyectar su expedición hacia el Polo Este (¡... al fin y al cabo, el Norte y el Sur estaban ya descubiertos!) Después de varios minutos de frenética cordura, fue calmándose el ánimo del «epatado» equipo de programación, hasta que, finalmente, su proverbial carencia de sentido común hizo que aceptaran el reto.

Una investigación apasionante

Al día siguiente, el ruido en la sala de programación era atronador. El «trumpita, trumpita...» de las impresoras taladraba los tímpanos, con la facilidad de un martillo hidráulico horadando bloques de cemento, y el chirrido característico del patinar de las neuronas surgía de las cabezas de todos y cada uno de los presentes.

En este ambiente de febril actividad, uno de los programadores alzó su voz sobre la verbena de decibelios:

—¡Lo tengo!... ¡Ya lo tengo!

—¡Oye!... ¿No será contagioso?

Interrogó suspicazmente uno de los vecinos de silla, sin dejar de aporrear su teclado.

—¡Qué digo que ya lo tengo... y esta vez va en serio!

Nadie replicó. Todos sabían que cuando el coordinador de programación hablaba en serio se avecinaban problemas; de nada bueno podía tratarse.

—¡Vais a hacerme caso de una vez! ¡A que quito los plomos!

Un latigazo sacudió al grupo, llevando a más de uno al borde del paro cardíaco. Una docena de punzantes miradas de odio se clavaron como alfileres en el emisor de tan despiadada amenaza.

Los ojos, hasta ahora desorbitados, iban retornando a su expresión normal a medida que la mano de Fernando se apartaba de la caja de fusibles.

—¡Fantástico! Oído al parche: el vigésimo-sexto programa del «extra» debe ser de impresión... ¿Todos de acuerdo?

Al no percibir respuesta alguna, el improvisado orador continuó:

—... Y ya tengo la idea para ese programa... ¡El copatible! ¿Qué os parece?

La sonrisa empezó a dibujarse en el rostro de los programadores esbozando una respuesta sin palabras; una respuesta que no precisaba palabras.

—¡Será iluso el «plomos» este!... ¡La fiebre del compatible ataca de nuevo!

El programa

El programa sintetiza plenamente los objetivos del equipo de programación. Su estructura, un tanto compleja al estar edificada sobre un algoritmo hipotético-deductivo, queda fuera de cualquier posible simplificación, de ahí que no incluyamos en esta ocasión nuestros habituales cuadros de descripción de variables y estructura del programa. En todo caso, las instrucciones adecuadas para el manejo y explotación del programa aparecen en el listado correspondiente.

Como observarán, no existe ningún carácter subrayado en el listado adjunto; el motivo es que el programa no incluye caracteres gráficos y ni tan siquiera de control.

Esperamos que disfruten con las excelencias de nuestro programa —compatible con casi cualquier ordenador presente en el mercado mundial— en cuyo desarrollo y puesta a punto han colaborado, por riguroso orden alfabético:

Xavier Alamán-Planificación de REMs.

Antonio Carvajal-Renumeración.

Francisco Lara-Glosó tan sensacional evento esgrimiendo el lápiz.

Fernando López-Ideó el engendro y «convenció» al grupo de programadores para que prestaran su colaboración (¡... y sino quito los plomos!)

Juan Manuel López-Trajo los bocadillos.

José Luis Martínez Vázquez de Parga-Logró que copiaramos su apellido completo... sin equivocarnos.

Ignacio Moragrega-Creación gráfica.

Pedro del Moral-Dio mucha idem.

Javier Murube-Manejo de impresoras.

Eduardo Pérez-Sonido y efectos especiales.

Carlos Rubio-Claqueta y vaciado de ceniceros.

Santiago Ruiz Javier Santaolalla (RUSA)-Responsables de la ensaladilla hardware.

Estamos plenamente convencidos que apreciarán en lo que vale la ardua y esforzada tarea de compatibilización que ha supuesto la puesta a punto de tan trascendental programa. El éxito con el que ha culminado el proyecto nos ha movido a presentarles el listado en un papel de calidad superior, infrecuente en nuestra revista, lo que les permitirá —¡probablemente!— enmarcarlo una vez introducido en el ordenador y confirmada su total compatibilidad.

```
  1 REM (C) EQUIPO DE PROGRAMACION DE TU MICRO, 1984 # EL COMPAT
IBLE
  5 REM COMPATIBILIDAD COMPROBADA EN LOS EQUIPOS:
 10 REM ATARI 600
 20 REM ATARI 800
 23 REM ORIC 1
 30 REM ORIC ATMOS
 32 REM SPECTRAVIDEO SV-318
 33 REM SPECTRAVIDEO SV-328
 34 REM SINCLAIR ZX-80
120 REM SINCLAIR ZX-81
121 REM SINCLAIR SPECTRUM
124 REM COMMODORE 64
129 REM VIC-20
132 REM CBM 3000
143 REM CBM 4000
149 REM CBM 8000
160 REM DRAGON 32
172 REM DRAGON 64
183 REM NEW BRAIN
191 REM LASER 200
207 REM LASER 2001
210 REM LASER 3000
234 REM APPLE II
239 REM APPLE IIE
240 REM MPF-II
248 REM KATSON
249 REM BBC
256 REM TRS-80 (RADIO SHACK)
400 REM IBM PC
401 REM INSTRUCCIONES: EJECUTE EL COMANDO "RUN"
406 PRINT "TU MICRO LES DESEA A TODOS UNAS MUY FELICES VACACIONE
S"
```

Ghost Gobbler

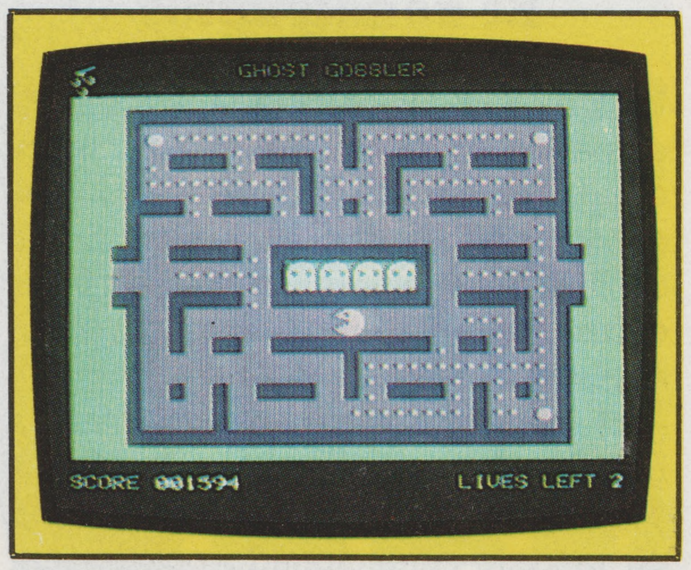
Una nueva versión del popular «come cocos», en esta ocasión creada para el ORIC-1 y ORIC-ATMOS.

El juego consiste en conducir al simpático GHOST a través del laberinto, comiéndose todos los puntos luminosos que aparecen en su camino y evitando que los fantasmas lo atrapen en su loca carrera.

Cuando GHOST logra deglutir alguna de las píldoras proteínicas, su fuerza se ve incrementada hasta el punto que es capaz de cazar y aniquilar a los fantasmas, ganando de este modo bonos y puntos... Pero ¡cuidado! ya que los fantasmas permanecerán tan solo unos segundos en estado de inferioridad respecto a GHOST.

En cada uno de los niveles pueden irse acumulando bonos de puntuación a medida que nuestro personaje va comiéndose las frutas que aparecen en su camino.

Este apasionante juego introduce fantasmas perseguidores, píldoras proteínicas, cajas de fantasmas, túneles, frutas...



Ordenador	ORIC-1/ATMOS
Soporte	Cinta
Tipo	Juego
Origen	IJK Software. Ltd.
Distribuidor	D.S.E.
Precio	

Gráficos/ estética	**
Sonido	**
Facilidad de control	**
Originalidad	*
Adicción	**
Resumen	**

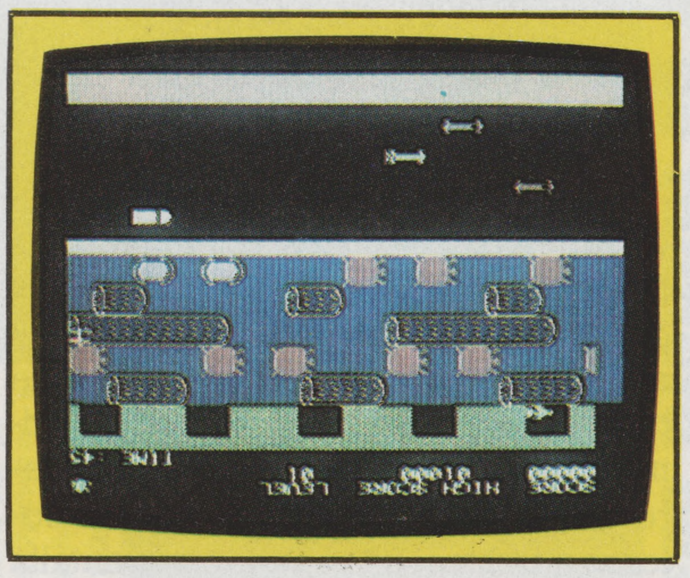
Green Cross Toad

En esta excitante versión del popular FROGGER, el juego empieza, ayudando a un simpático «sapo» a cruzar la carretera, sorteando el intenso tráfico y, a continuación, a cruzar un río. En este último escenario, nuestro personaje contará con la colaboración de troncos flotantes y de un grupo de tortugas que lo auxiliarán en su viaje hacia la seguridad de la guarida.

La primera dificultad es atravesar la carretera, a través de la que circulan coches y camiones en ambas direcciones.

Una vez superada esta fase, se llega a la orilla de un río en la que hay que prestar una gran atención so pena de caer en las fauces de las serpientes, mientras se espera el momento oportuno para lanzarse sobre alguna de las tortugas.

Después hay que continuar saltando sobre los troncos flotantes que viajan a distinta velocidad, extremando las precauciones para no caer al agua. Por último, hay que acertar en una de las cinco guaridas de la rana; y si cuenta con la suerte de encontrarse con las moscas que aparecen y desaparecen en los distintos casilleros, su puntuación se verá engrosada por bonos.



Ordenador	ORIC 1/ATMOS
Soporte	Cinta
Tipo	Juego
Origen	IJK Software, Ltd.
Distribuidor	D.S.E.
Precio	

Gráficos/ estética	*
Sonido	**
Facilidad de control	**
Originalidad	*
Adicción	**
Resumen	**

Escalador loco

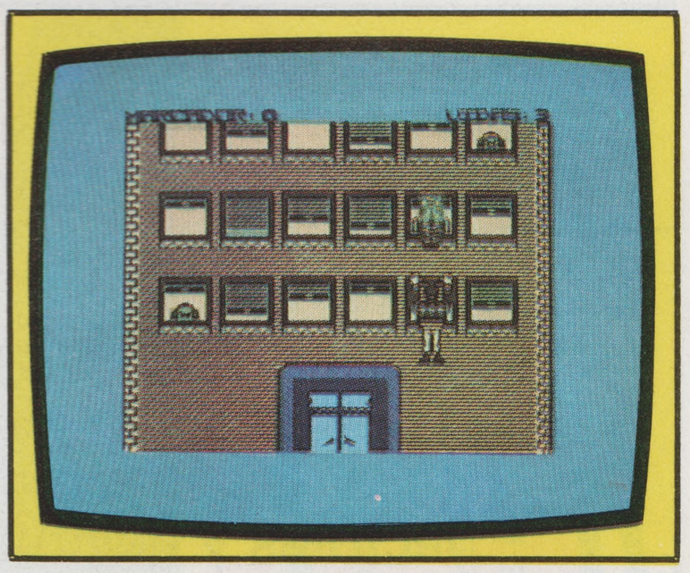
En esta ocasión se trata de nuevo de una versión de otro popular vídeojuego: el CRAZY CLIMBER. El escalador debe superar todo tipo de obstáculos en su duro ascenso hasta la cima, reptando por las verticales paredes de cemento de un gigantesco edificio y con el único apoyo que puede encontrar en las ventanas.

En los primeros pisos del edificio tiene que esquivar las macetas que le arrojan unos individuos poco escrupulosos que, además de los mortíferos lanzamientos, no dudan en precipitar al vacio al escalador si le sorprenden en una de las ventanas.

Para colmo de males, las persianas de las ventanas suben y bajan con evidente peligro.

En la zona intermedia del edificio el escalador se ve liberado del «diluvio vegetal», no obstante, el abrir y cerrar de las ventanas se hace ahora aún más frenético que en la primera parte de la ascensión. Como obstáculo final, antes de alcanzar la cima del edificio aparece un enorme gorila que la emprende a puñetazos con el escalador a nada que este se despiste.

¡Animo y a por el gorila!

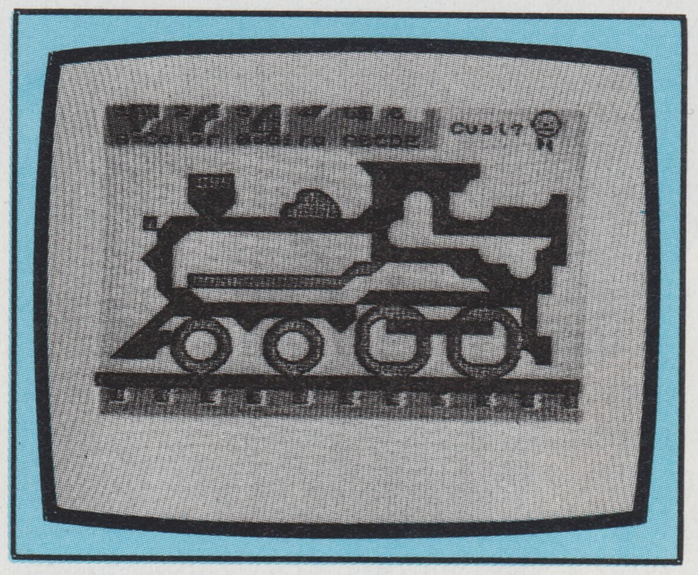


Ordenador	SPECTRUM
Soporte	Cinta
Tipo	Juego
Origen	JGB Software
Distribuidor	Ventamatic
Precio	1.700 ptas.

Gráficos/estética	***
Sonido	***
Facilidad de control	*
Originalidad	*
Adicción	**
Resumen	**

bazar SOFT

El constructor



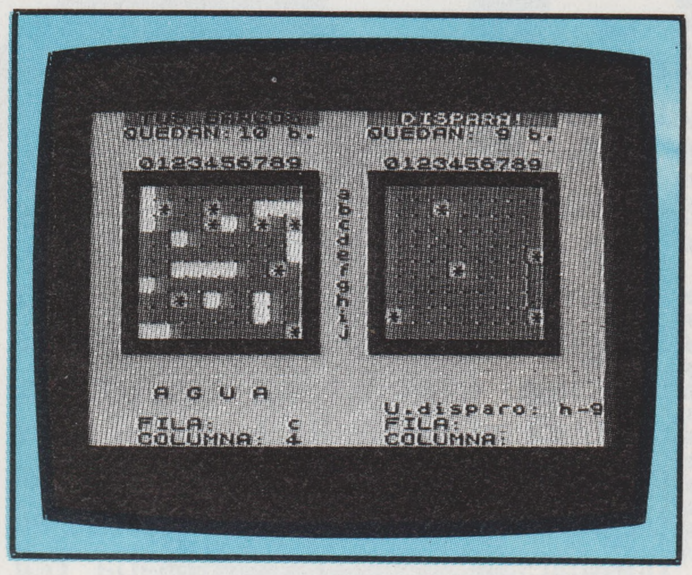
Este es un juego orientado a los aficionados al ZX-Spectrum de menor edad. Está basado en las tradicionales construcciones de piezas de madera de diferentes colores que, además de entretener a los niños, tienen un indudable interés pedagógico.

El juego consiste en construir, a base de 5 tipos diferentes de piezas, y con la ayuda de la goma de borrar, un modelo seleccionado previamente de entre los 5 posibles que suministra el programa. También existe la opción de construir un modelo original, partiendo como base de las mismas piezas empleadas en los modelos fijos. en el caso de decidirse por uno de los modelos que ofrece el programa (A, B, C, D o E), éste aparece en pantalla al pulsar la tecla correspondiente.

Es posible seleccionar libremente el color de las piezas, así como girarlas a cualquier posición. Además, está abierta la posibilidad de grabar los dibujos en cinta para su posterior recuperación. El movimiento del «pincel» de dibujo se logra accionando las teclas de desplazamiento del cursor o por medio de un joystick conectado al SPECTRUM a través de un interface KEMPSTON.

Gráficos/estética	**
Sonido	*
Facilidad de control	***
Originalidad	***
Adicción	*
Resumen	**

Ordenador	**SPECTRUM**
Soporte	**Cinta**
Tipo	**Educativo**
Origen	**OD Software**
Distribuidor	**Ventamatic**
Precio	**1.700 ptas.**



Caramelos y Batalla naval

Caramelos. El juego consiste en deducir, mediante sucesivos intentos, la posición y el color de los caramelos que el ordenador genera de forma aleatoria y de acuerdo al nivel de complejidad seleccionado. En la primera pantalla, se selecciona el número de caramelos y el número de colores a utilizar. En la segunda pantalla se eligen los caramelos, quedando reflejados en la pizarra de anotaciones. La elección de cada caramelo se hace con el movimiento de dos teclas, confirmando el caramelo elegido con ENTER. Cabe la opción de abandonar el juego tecleando «FIN».

Batalla naval. Un desafío entre el Spectrum y el jugador, en el que la puntería y la intuición juegan un papel determinante a la hora de destruir la flota enemiga. Como primer paso, debe introducirse la posición de los barcos en el tablero de juego, indicando si la nave va a colocarse de forma horizontal o vertical. A continuación, se define la fila y columna en que se va a posicionar cada barco. La distribución de los barcos es la siguiente: 1 de 4 casillas, 2 de 3, 3 de 2 y 4 de 1.

Cuando un barco es alcanzado, aparece la oportuna indicación sobre el tablero de juego.

Ordenador	**SPECTRUM**
Soporte	**Casete**
Tipo	**Juego**
Origen	**Investrónica**
Distribuidor	**Investrónica**
Precio	**1.800 ptas.**

Gráficos/estética	*
Sonido	*
Facilidad de control	***
Originalidad	*
Adicción	**
Resumen	**

el correo del lector

La revista **TU MICRO** también tiene un espacio reservado para reflejar las sugerencias de los lectores y responder a sus consultas.

Las cartas para «El correo del lector» debe dirigirlas a:

TU MICRO
Apdo. de Correos 61.294
28080 MADRID

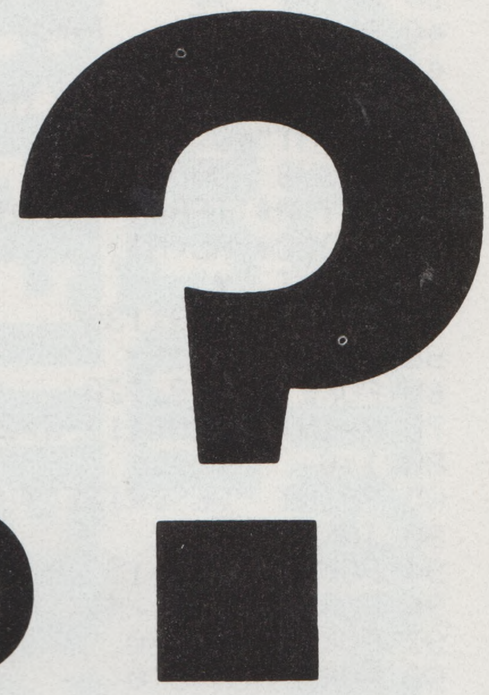
ANUNCIOS BREVES

Un servicio gratuito para los lectores de **TU MICRO**

- Compra/venta e intercambio de equipos, accesorios y programas.
- Avisos y comunicaciones de clubs de usuarios
- ...

(Vea tarjeta ANUNCIOS BREVES en pág. 39)

Adivina un número



Una de las mayores dificultades con las que tropiezan aquellos que intentan dar sus primeros pasos en las tareas de programación es encontrar programas que apoyen su aprendizaje. Programas sencillos, construidos alrededor de una estructura clara e inteligible y que, al tiempo, realicen alguna función medianamente atractiva. «Adivina un número» puede ser uno de estos programas, aptos para no iniciados.

El programa que nos ocupa resulta apto para que el recién llegado al mundo de la informática aprenda a utilizar algunas instrucciones BASIC. Por ejemplo, son varias las instrucciones de salto condicional (IF... THEN) y de creación de bucles (FOR-NEXT) que intervienen en el programa; instrucciones cuyo análisis puede resultar ilustrativo y didáctico para quien pretenda empezar a familiarizarse con el lenguaje BASIC.

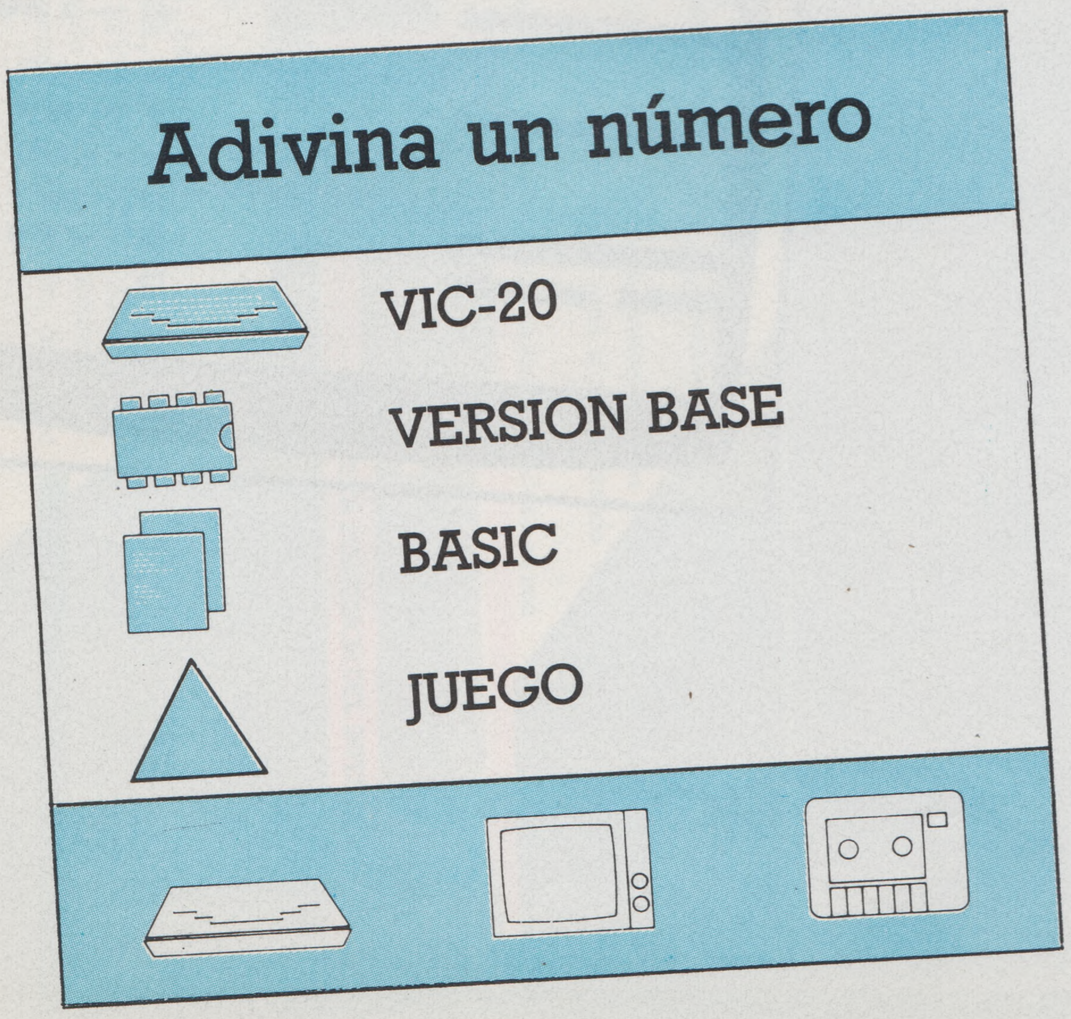
Si el lector es ya un corredor de fondo, poco va a aprender con este programa, aunque sí puede elevar su marca añadiendo un nuevo juego a su biblioteca de programas. En uno y otro caso, no cabe la menor duda de que la primera tarea a realizar coincide: se trata de realizar un ligero ejercicio «dactilar» que terminará al introducir la línea 2900 del listado adjunto.

¡Y el juego puede empezar! Tres pulsaciones adicionales sobre las teclas R, U y N, seguidas por una enérgica acción sobre RETURN. El resultado será la aparición en pantalla de las instrucciones del juego «Adivina un número». Pulsando cualquier tecla, el texto desaparecerá, cediendo su lugar a la interrogación acerca de qué nivel de juego es el que desea seleccionar: 1 ó 2. En el primero de los niveles el número a adivinar estará comprendido entre 100 y -100, mientras que en el segundo el número secreto se situará entre 200 y -200.

Tras esta definición de nivel, da comienzo el juego. La pantalla mostrará la cabecera con el nombre del juego y una interrogación a la que el usuario debe responder escribiendo un número, con o sin signo. En el instante en el que se accione la tecla RETURN, el ordenador aceptará la jugada introducida, la comparará con el número secreto e indicará si el número introducido es inferior o superior al que hay que adivinar.

La pantalla visualizará en todo momento un determinado número de jugadas anteriores, que servirán de referencia y de ayuda para afinar la puntería en la próxima jugada.

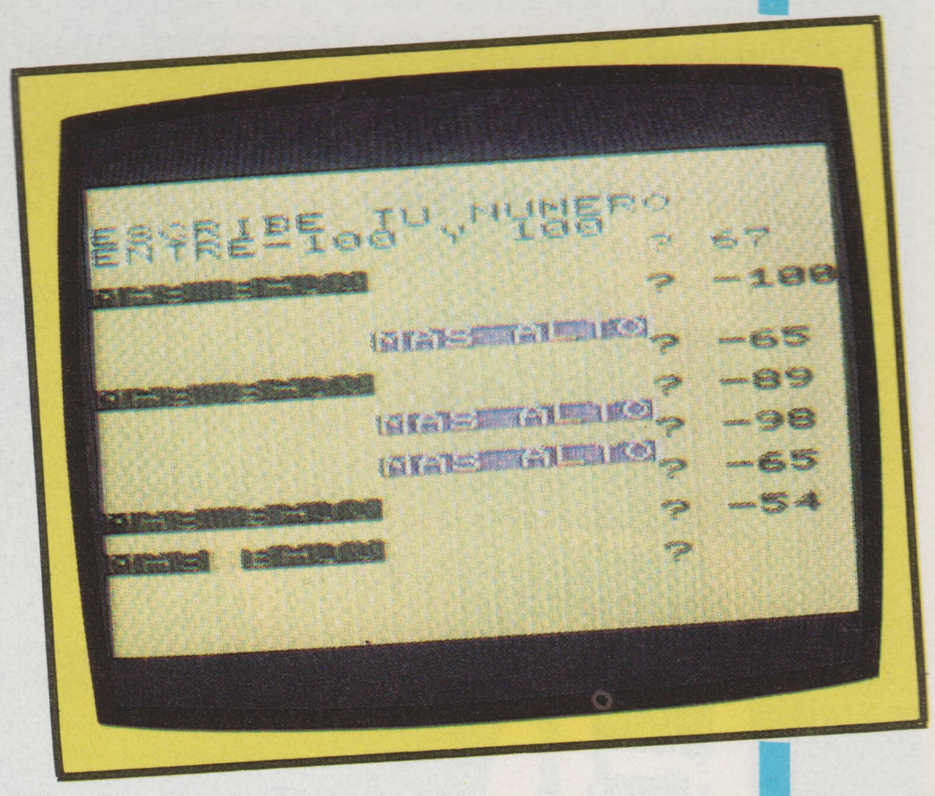
Cuando el usuario logre adivinar la clave secreta, aparecerá en pantalla el número de



LISTADO

```
10 REM RUSA # ADIVINA UN NUMERO # VIC-20
15 GOSUB1000
20 PRINT "■QQQQ    ADIVINA UN","       NUMERO"
30 PRINT "QQ    ESCRIBE ","    TU NUMERO", "  Y PULSA LA TECLA","    RETURN"
40 PRINT "QQ    PARA CONTINUAR ","    PULSA UNA TECLA "
42 GOSUB 2000
43 GET A$:IF A$="" THEN 43
45 PRINT"■QQQQQQQ£    ELIGE GRADO DE","Q       DIFICULTAD","QQ       R1/2←"
47 GOSUB 2000
49 GET A$:IFA$<"1"ORA$>"2"THEN49
50 Z=200*VAL(A$)
60 PRINT "■"
65 POKE 36879,138
66 GOSUB 800
68 POKE 36879,120
70 X=INT(Z*RND(1)-Z/2)
80 I=0
85 PRINT "QESCRIBE TU NUMERO     ENTRE";-Z/2;"Y";Z/2
90 INPUT"                 ";N
94 POKE 36878,15
96 FOR M=0 TO 24 : POKE 36875,234:NEXTM
97 POKE 36878,0
98 POKE 36875,0
100 I=I+1
110 IF N<X THEN PRINT "        ■RMAS ALTO■←":GOTO 90
120 IF N>X THEN PRINT"■RMAS BAJO←":GOTO 90
125 GOSUB500
130 PRINT "■QQQQ    ENHORABUENA" "QQQ    RACERTASTE←","QQ    DESPUES DE";I
135 PRINT "Q    INTENTOS"
140 PRINT "QQQQQQ£RQUIERES REPETIR (S/N)?←"
145 GOSUB 2000
150 GET R$:IF R$=""THEN150
160 IF R$<>"S" AND R$<>"N"THEN150
170 IF R$="S"THEN45
180 PRINT "■ADIOS"
185 POKE 36879,27
190 END
500 POKE36878,15
510 FOR L=148 TO 220 STEP .7
520 POKE 36876,L:NEXTL
530 FOR L=128 TO 200
540 POKE 36876,L:NEXT L
550 FOR L=200 TO 128 STEP-1
560 POKE 36876,L:NEXTL
570 POKE 36878,0
580 POKE 36876,0
590 RETURN
800 POKE 368??,220
810 FOR L=15 TO0 STEP -1
820 POKE 36878,L
830 FOR M=0 TO 69 :NEXT M: NEXT L
840 POKE 36877,0:POKE36878,0
850 RETURN
1000 POKE 36878,15
1010 B=0
1020 FORL=128 TO 255 STEP11
1030 POKE 36874,L
1040 FORM=0 TO 9:NEXTM
1050 B=B+1
1060 IF B=3 THEN B=0:POKE 36874,0
1070 NEXT L
1080 POKE 36874,0
1090 POKE 36878,0
1100 RETURN
2000 FOR N=38446 TO 38463:POKE N,0
2005 I=N+374
2010 POKE I,0:NEXT N
2020 FOR N=7726 TO 7743:POKE N,42:I=N+374
2030 POKE I,42:NEXTN
2040 FOR N=7748 TO 8100STEP 22:POKE N,42
2050 I=N+17:POKE I,42:NEXT N
2060 FOR N=38468 TO 38820 STEP22:POKE N,0
2070 I=N+17:POKE I,0:NEXTN
2900 RETURN

READY.
```



END

ESTRUCTURA DEL PROGRAMA

10-43	Cartel de entrada.
45-68	Cartel para elegir el grado de dificultad.
70	Generación del número aleatorio.
80-120	Comparación del número aleatorio con la jugada introducida.
125-190	Tratamiento en el caso de que se acierte el número secreto.
500-590	Rutina que genera un sonido semejante a un silbido de admiración.
800-850	Rutina que produce un ruido de explosión.
1000-1100	Rutina que genera el sonido de entrada.
2000-2900	Rutina para el trazado del recuadro en los carteles de texto.

TABLA DE VARIABLES

A$	Variable que almacena el grado de dificultad.
B	Variable auxiliar para establecer la ausencia de sonido cada tres notas.
I	Almacena el número de intentos.
L, M	Variables FOR de diversa utilidad.
N	Almacena la jugada.
R$	Variable GET para nueva partida.
X	Número a adivinar, generado por el programa.
Z	Valor absoluto del margen en el que se encuentra el número a adivinar. (Nivel 1, Z=100; Nivel 2, Z=200).

intentos o jugadas que se han realizado. A su vez, el ordenador preguntará si desea empezar una nueva partida; responda accionando la tecla S si desea intentarlo de nuevo.

Tal como ya es habitual en los listados que publicamos, los caracteres que aparecen subrayados corresponden a los caracteres de control de las teclas en cuestión, cuya representación en pantalla coincide con el carácter en negativo (vídeo inverso) del que aparece en el listado. Un detalle final a considerar es que el desarrollo del juego está amenizado con la emisión de un conjunto de sonidos asociados a cada situación.



DISPONIBLE CASETE VIRGEN

250 ptas

Apellidos .....
Nombre .....
Domicilio .....
Localidad.....Teléfono .....
Provincia .....Código Postal .....

Deseo recibir[1] CASETES VIRGENES «TU MICRO», especiales para ordenador, al precio de 250 ptas cada una. El importe lo haré efectivo mediante:

1) Indicar el n.º de casetes solicitado

☐ Talón bancario a nombre de TU MICRO. n.º de talón .....Banco .....
☐ Giro postal n.º .....Fecha .....Localidad.....
☐ Reembolso del pedido más gastos de envio.

FIRMA

Recortar o copiar este cupón y enviar a:
Ediciones INGELEK Apartado de Correos 61294 28080 MADRID

Poligonales

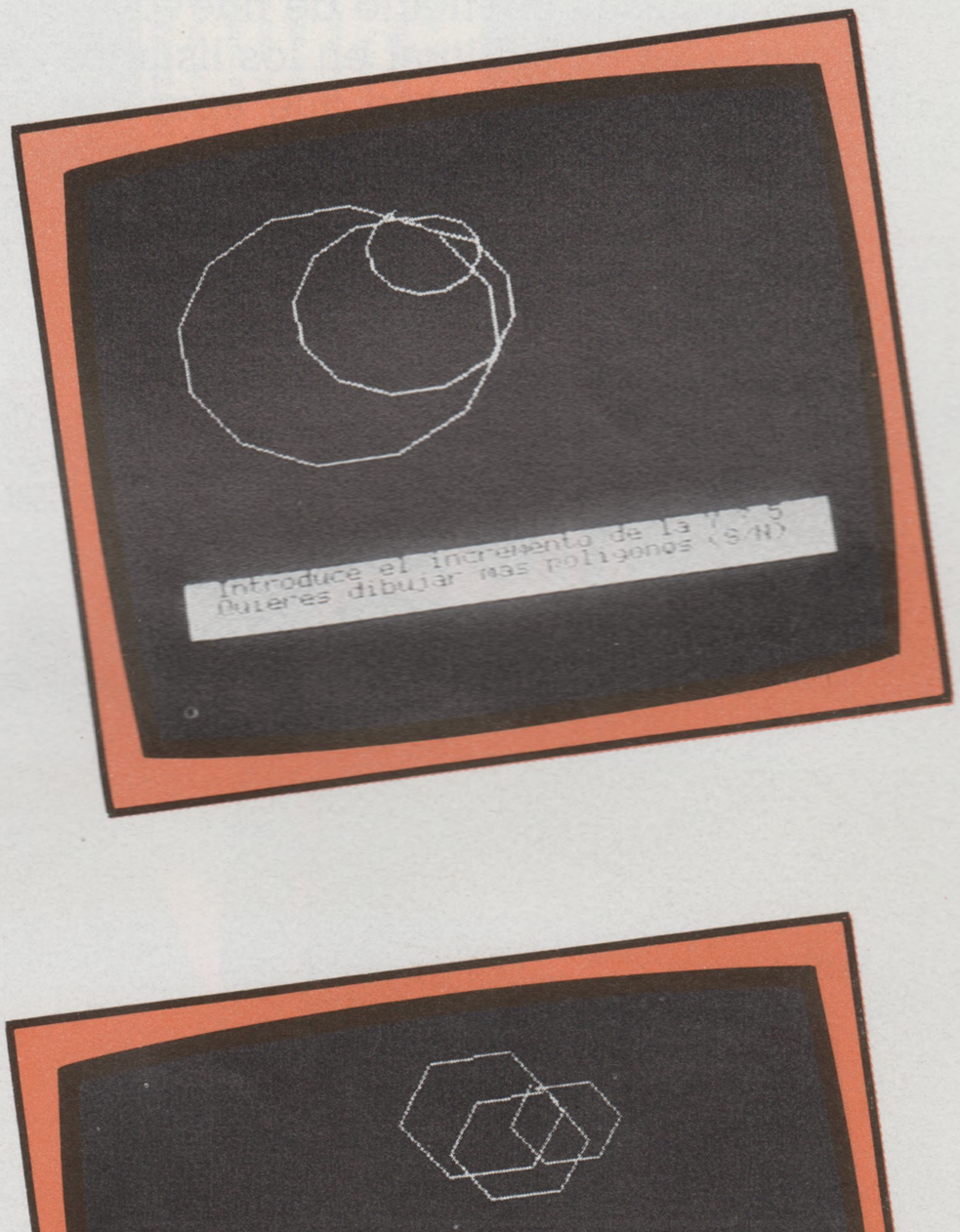
Entre juego y juego, nunca está de más coleccionar algún que otro programa destinado a aplicaciones más serias. Este es el caso de POLIGONALES: un programa que exprime la capacidad de cálculo y representación gráfica del ORIC, orientándola al dibujo de desarrollos poligonales.

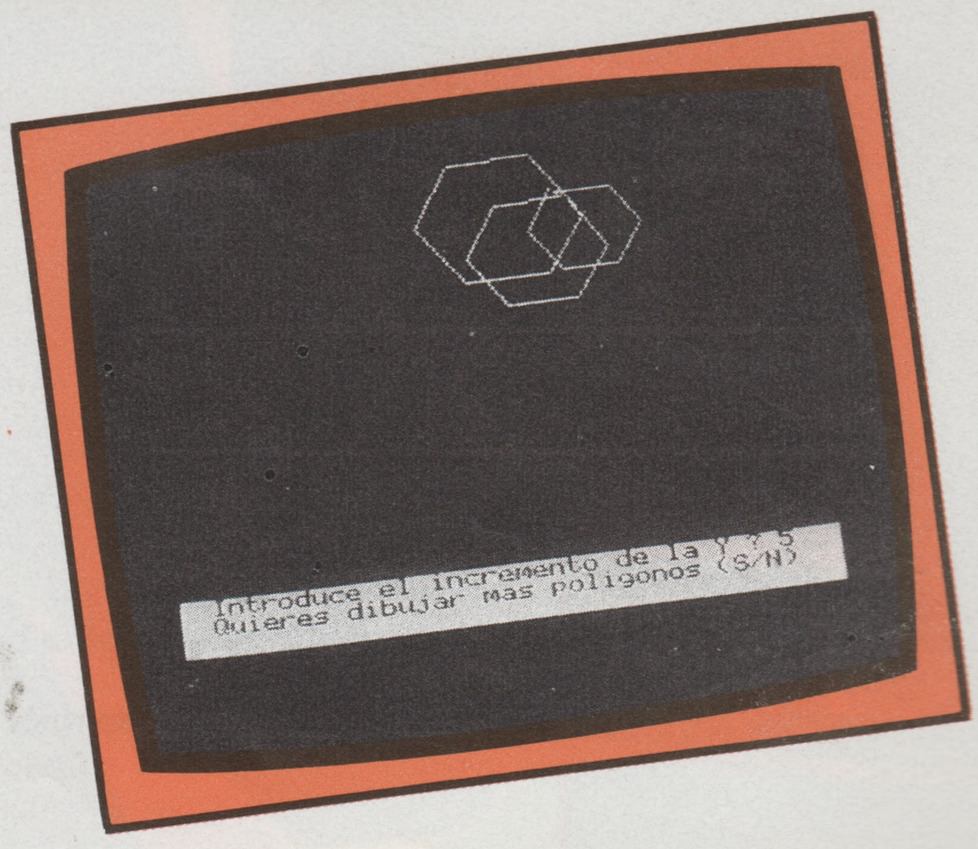
Los polígonos a dibujar y su posición relativa son definidos por el usuario respondiendo a las preguntas que irá formulando el ordenador.

Son 9 los interrogantes que plantea el ordenador y que determinarán las características del desarrollo gráfico:

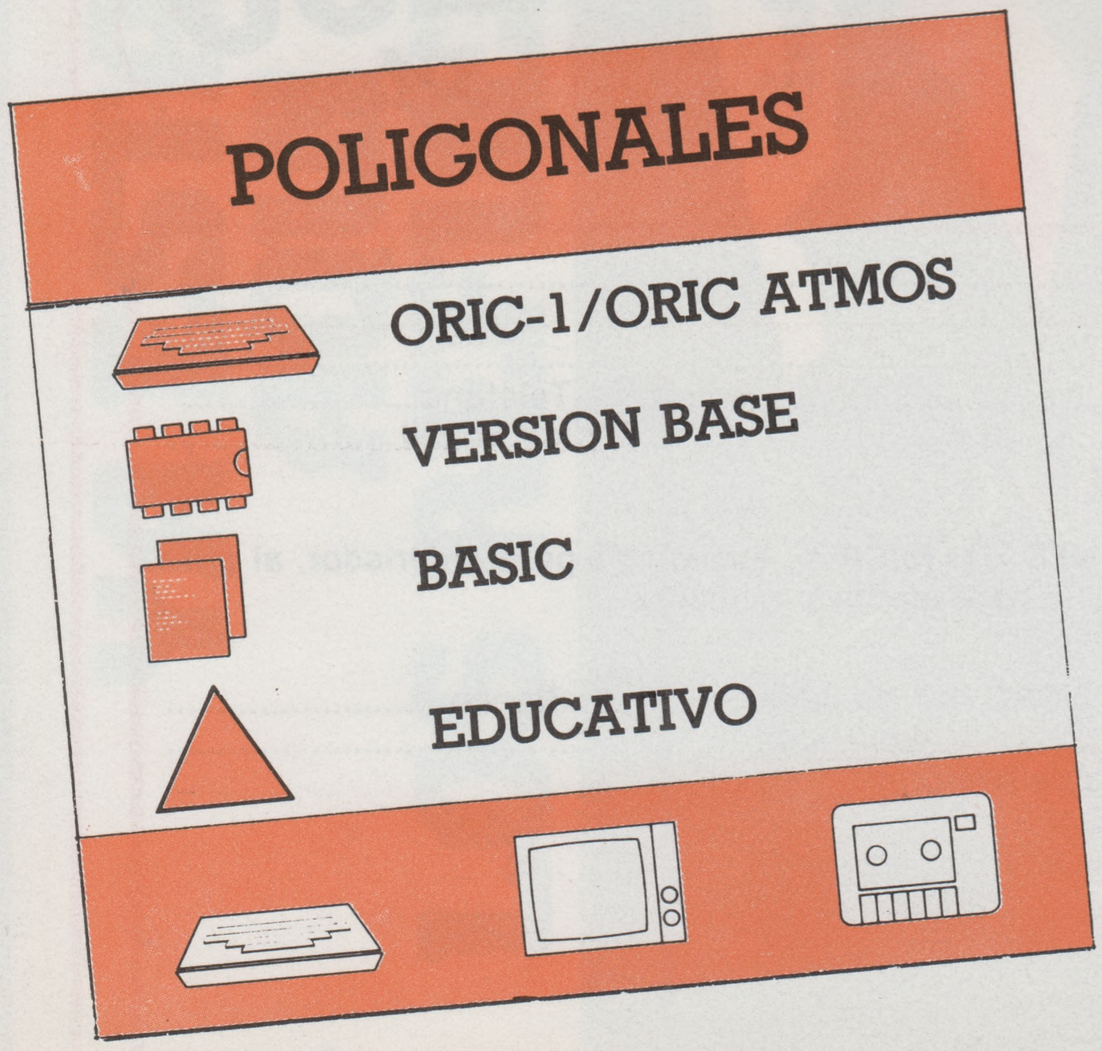
1. Número de lados del polígono.
2. Longitud inicial del lado.
3. Incremento del giro en grados.
4. Incremento del lado en cada giro.
5. Número de polígonos a dibujar.
6. Coordenada A del primer polígono.
7. Coordenada B del primer polígono.
8. Incremento de X.
9. Incremento de Y.

Como podemos deducir de inmediato, la respuesta a la pregunta «¿Número de lados?» establece qué tipo de polígono hay que dibujar. Así, por ejemplo, si la respuesta es 3 se dibujarán triángulos, si es 4 cuadrados... A partir de este polígono básico definido, el





Dos ejemplos de desarrollos poligonales obtenidos sobre la pantalla de TV conectada al ORIC-1.



ESTRUCTURA DEL PROGRAMA

20-100 Introducción de datos y definición de las variables que definen el gráfico a realizar.

140 Conversión del ángulo de giro de grados a radianes.

160 Llamada a la subrutina para el dibujo de un polígono.

170-200 Actualiza las variables que preparan el dibujo de un nuevo polígono.

1000 Principio de la subrutina para el dibujo de un polígono.

1030 Llamada a la subrutina para la conversión de coordenadas polares a cartesianas.

1040 Dibujo de un lado del polígono.

1050 Actualiza el ángulo del nuevo vértice del polígono.

1060 Comprueba si se han dibujado todos los lados del polígono.

2000 Principio de la subrutina para la conversión de polares a cartesianas.

TABLA DE VARIABLES

A Coordenada X del primer polígono.

ALFA Valor en grados de cada vértice de los polígonos.

AVA Incremento del lado entre dos polígonos dibujados.

B Coordenada Y del primer polígono.

GIR Giro entre cada dos polígonos dibujados.

LI Lado inicial del polígono básico.

NC Número total de polígonos a dibujar.

NL Número de lados de los polígonos a dibujar.

PUN Contador del número de lados del polígono.

S Contador del número de polígonos dibujados.

X Incremento de la coordenada X entre dos polígonos.

X1 Coordenada X de cada vértice del polígono.

Y Incremento de la coordenada Y entre dos polígonos.

Y1 Coordenada Y de cada vértice del polígono.

programa permite construir muy diversos tipos de desarrollos (con disposición en espiral), para lo cual bastará con que el usuario defina el incremento del lado y del giro. Si, además, se quiere que los polígonos se desplacen a través de la pantalla, habrá que fijar los valores del incremento de las coordenadas X e Y, por supuesto, una vez establecidas las coordenadas A y B del primer polígono a dibujar. Observando el programa se deduce que estos incrementos pueden ser tanto positivos como negativos; con lo cual, el desarrollo poligonal puede trazarse de derecha a izquierda (de arriba hacia abajo), o viceversa.

La sencillez del programa la evidencia el reducido número de líneas que lo integran. La zona principal del programa inicializa las variables que definen el dibujo a realizar, ac-

Ejemplos de desarrollos poligonales

Dibujo poligonal	Datos a introducir como respuesta a los sucesivos interrogantes que plantea el programa.								
	1	2	3	4	5	6	7	8	9
Espiral triangular	3	10	20	3	30	120	100	0	0
Espiral con cuadrados	4	10	10	5	30	120	100	0	0
Anillos esféricos	20 20	15 05	18 18	0 0	20 20	120 180	100 50	0 0	0 0
Anillos regulares	4	40	9	0	40	120	100	0	0
Gráfico tubular	4	2	1	1	30	10	10	2	1

LISTADO

```
  5 REM POLIGONALES - ORIC 1 - Eduardo Perez
 10 HIRES
 20 S=1
 30 INPUT "Numero de lados ";NL
 40 INPUT "Longitud inicial del lado ";LI
 50 INPUT "Incremento del giro en grados ";GIR
 60 INPUT "Incremento del lado en cada giro ";AVA
 70 INPUT "Numero de poligonos a pintar ";NC
 80 INPUT "Introduce las coordenadas del primer
    poligono (A,B)";A: INPUT B
 90 INPUT "Introduce el incremento de la X ";X
100 INPUT "Introduce el incremento de la Y ";Y
140 GIR=(GIR*PI)/180
150 ALFA=0
160 GOSUB 1000
170 ALFA=ALFA+GIR
180 A=A+X
190 B=B+Y
200 LI=LI+AVA
210 S=S+1
220 IF S=<NC THE 160
230 PRINT "Quieres dibujar mas poligonos (S/N)  "
240 GET A$
250 IF A$="S" THEN 10
260 END
1000 REM DIBUJA UN POLIGONO
1010 PUN=1
1020 CURSET  A,  B,1
1030 GOSUB 2000
1040 DRAW X1,Y1,1
1050 ALFA=ALFA+((2*PI)/NL)
1060 IF PUN=NL THEN 1090
1070 PUN=PUN+1
1080 GOTO 1030
1090 RETURN
2000 REM CAMBIO DE CARTESIANAS A POLARES
2010 X1=LI*COS(ALFA)
2020 Y1=LI*SIN(ALFA)
2030 RETURN
```

END

tualiza las coordenadas de cada polígono y va contando los polígonos trazados hasta concluir la representación de estos en la pantalla. La primera de las subrutinas (línea 1000) se ocupa de dibujar cada polígono, mientras que la segunda (línea 2000) efectúa el cambio de coordenadas polares a cartesianas.

Para familiarizarse con el funcionamiento del programa puede definir los desarrollos que, a título de ejemplo, se relacionan en la tabla 1. Las series de valores que se indican para cada desarrollo poligonal reflejan los números que debe introducir como respuesta a los nueve interrogantes que plantea el programa.

Una posibilidad interesante consiste en superponer varias representaciones, con objeto de lograr trazados más complejos. Para ello basta con responder afirmativamente a la pregunta final «¿Quieres dibujar más polígonos?» y, a continuación, introducir un nuevo grupo de datos. La práctica conducirá a resultados espectaculares.

¿Quién teme al ensamblador feroz?



El aficionado a bucear en los programas suele tropezarse con harta frecuencia con rutinas o zonas de aplicaciones programadas en lenguaje máquina o en ensamblador. Si el buceador es experto, el encuentro no se traduce en mayores dificultades que las que supone la puesta en práctica de sus conocimientos del lenguaje de bajo nivel. No obstante, cuando es un profano quien ve desaparecer el camino BASIC para entrar en el sendero del ensamblador, es muy posible que el resultado sea una pura y simple pérdida de guión y termine desorientado.

La colisión con una zona de programa en lenguaje ensamblador puede crear en el no iniciado en las lides de la disección de programas una natural sensación de temor a tropezarse con el feroz ensamblador. Puestos a colaborar en su aprendizaje, vamos a hablarle en esta ocasión de ferocidades que no son más que una distorsionada interpretación de la docilidad del lenguaje ensamblador.

En el fondo, todo es código máquina

El cerebro de operaciones de nuestro microordenador, sea cual fuere su marca, modelo y procedencia, es una pequeña «cucaracha» plástica, nublada de patitas metálicas y contra la que no tienen nada que hacer los eficaces mata-insectos en aerosol. Su nombre es microprocesador y su apellido —Z/80, 6502, 6510, 8086, 6809...— denota su pertenencia a alguna de las pocas y rancias familias que aportan la inteligencia a los microordenadores. Parece obvio, pues, que cualquier instrucción que deba ejecutar nuestro equipo, debe pertenecer al lenguaje propio del microprocesador que constituye su cerebro.

Estos avanzados circuitos integrados dialogan en un lenguaje cuyo alfabeto se reduce a dos únicos caracteres: 0 y 1. Y al igual que ocurre en cualquier otro lenguaje, combinando los caracteres de su alfabeto (ceros y unos, en el caso que nos ocupa) se construyen las palabras con las que se establecerá el diálogo. Como quiera que éste es el lenguaje propio de las máquinas, nada parece más evidente que bautizarlo con el nombre de «Lenguaje Máquina»... ¿A que el nombre le resulta familiar?

El lenguaje máquina comparte ciertas características propias de los lenguajes convencionales. Por ejemplo, presentan una diversidad idiomática que hace que el lenguaje de un tipo de microprocesador no sea entendido por otro, y viceversa. ¡Nada sería más cómodo que franceses, ingleses, alemanes

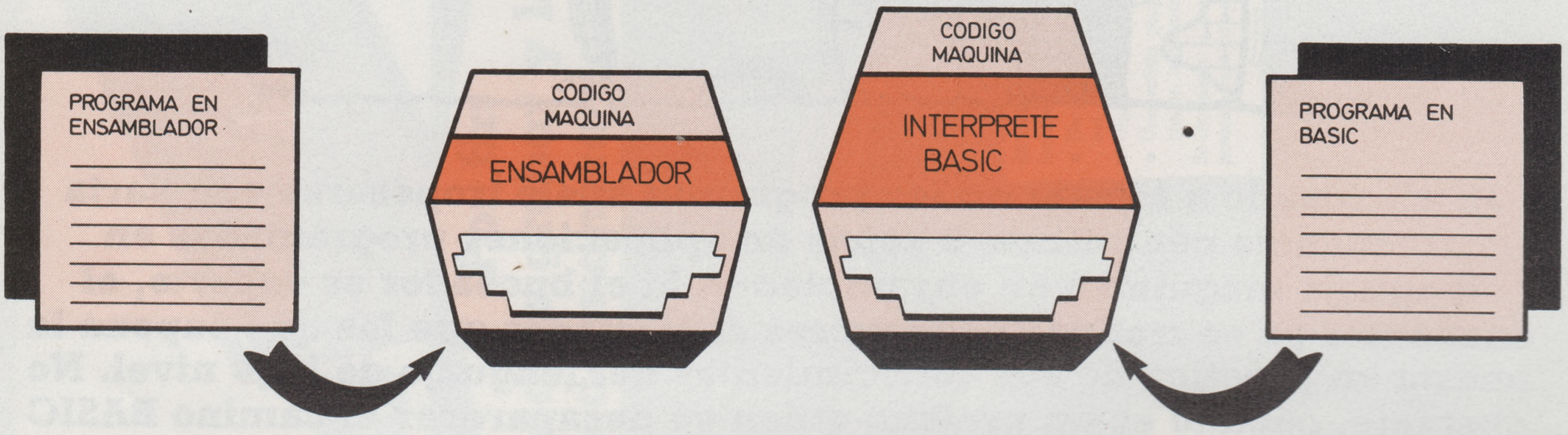
y españoles habláramos el mismo idioma! También es cierto que el mundo de las máquinas presenta algunas particularidades exclusivas. Sin ir más lejos, y con el ánimo de facilitar su labor de diálogo, en el lenguaje de las máquinas las palabras se construyen con un determinado número de «letras». Número que está perfectamente establecido y que en la «sociedad» de los microprocesadores suele ser de 8 ó de 16. Los microordenadores más populares en la actualidad (SPECTRUM, ORIC, COMMODORE-64...) están basados en microprocesadores de 8 bits y, en consecuencia, utilizan palabras de 8 letras, a lo que es lo mismo: de 8 bits, o en nomenclatura más simplificada, de 1 byte (un octeto).

de un intérprete BASIC (residente en el interior de su memoria ROM) que se ocupa de la ineludible tarea de traducción.

Facilidades todas... pero pagando un precio: la lentitud. Poco hay más engorroso y dilatante que mantener una conversación con un obcecado súbdito del Camerún a través de un intérprete. Llevada la situación al marco de la máquina, podemos imaginar fácilmente la pérdida de tiempo que supone la fase de traducción.

¿Qué pinta en todo esto el ensamblador?

Aun aceptando la rapidez que permitiría dialogar directamente en código máquina,



El lenguaje ensamblador, bastante más próximo al lenguaje máquina que el BASIC, presenta la ventaja de reducir el tiempo que debe invertir el microordenador en la fase de traducción.

Sabemos ya cual es el lenguaje que hablan los microordenadores y conocemos la simplicidad de su alfabeto (ceros y unos)... Entonces ¿por qué hablamos con nuestro microordenador en BASIC en lugar de utilizar directamente el lenguaje máquina? La respuesta es simple: ¡arriba la comodidad! Convendrán con nosotros que para el usuario humano representa un esfuerzo mucho mayor hablar en lenguaje máquina que en BASIC; además, si se trata de hacer un esfuerzo de traducción ¡qué lo haga la máquina!... ¡para eso están a nuestro servicio!

Fruto de este razonamiento (¡de nuevo la ciencia al servicio de la comodidad!), la gentil máquina pone a nuestra disposición los servicios

eliminando intermediarios, hay que reconocer que sigue en pie el inconveniente de su tedioso empleo. Y aquí es donde aparece el lenguaje ensamblador (denominado también lenguaje de ensamble o lenguaje mnemónico), como puente entre el rudimentario código máquina y el BASIC de alto nivel.

El ensamblador es un lenguaje muy próximo en su estructura al código máquina, aunque más fácil de aprender y recordar. Su vocabulario no es ya una críptica e inexpresiva cadena de ceros y unos, sino que se trata de un repertorio de «mnemónicos» o abreviaturas —de tres o cuatro letras, normalmente— que recuerdan, con mayor o menor propiedad, la operación u orden que representan.

Sigue sin ser el lenguaje propio de la máquina, pero sí es un pariente próximo del anterior que puede dar mayor rapidez al diálogo... ¡no cabe la menor duda que es más fácil entenderse con un portugués que con nuestro ínclito nativo del Camerún!... ¿El traductor? Sigue siendo necesario, aunque esta vez su tarea es mucho menos compleja y, por ende, más rápida.

Los traductores expertos en este cometido (los denominados programas ensambladores; no confundir con programas escritos en lenguaje ensamblador) no actúan como intérpretes o traductores simultáneos que intervienen tras cada frase de la conversación. Por el contrario, aguardan discretamente a que el usuario haya concluido su retahíla de instrucciones redactadas en lenguaje ensamblador. Acto seguido, ejecutan velozmente la traducción encomendada por el usuario, de un golpe, entregando como resultado la versión completa en lenguaje máquina del programa fuente redactado en lenguaje ensamblador. Semejante operación traductora «de un golpe» se denomina «compilado».

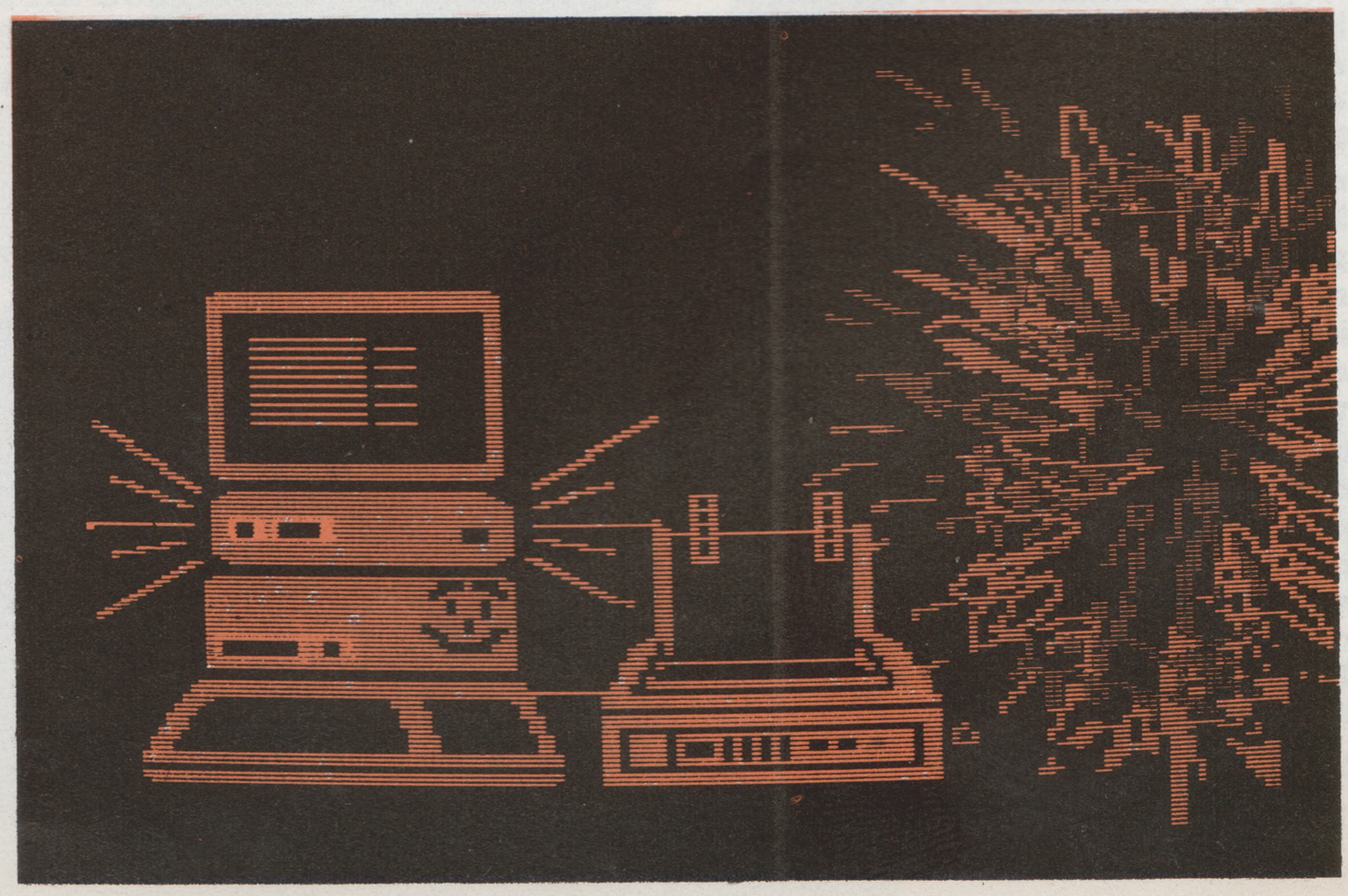
Al compilar el programa en lenguaje ensamblador se llega al mismo resultado que si el programa se hubiera escrito originalmente en lenguaje máquina. Visto el resultado y la mayor facilidad de aprendizaje y empleo del lenguaje ensamblador, resulta evidente el hecho de que casi nunca se recurra directamente al código máquina, sino que el diálogo de bajo nivel este donimado por el ensamblador. Un dato que corrobora la docilidad de este lenguaje es que algunos usuarios suelen elaborar rutinas en lenguaje ensamblador para, a continuación, compilarlas «a mano», siguiendo los mismos pasos que andaría el programa compilador.

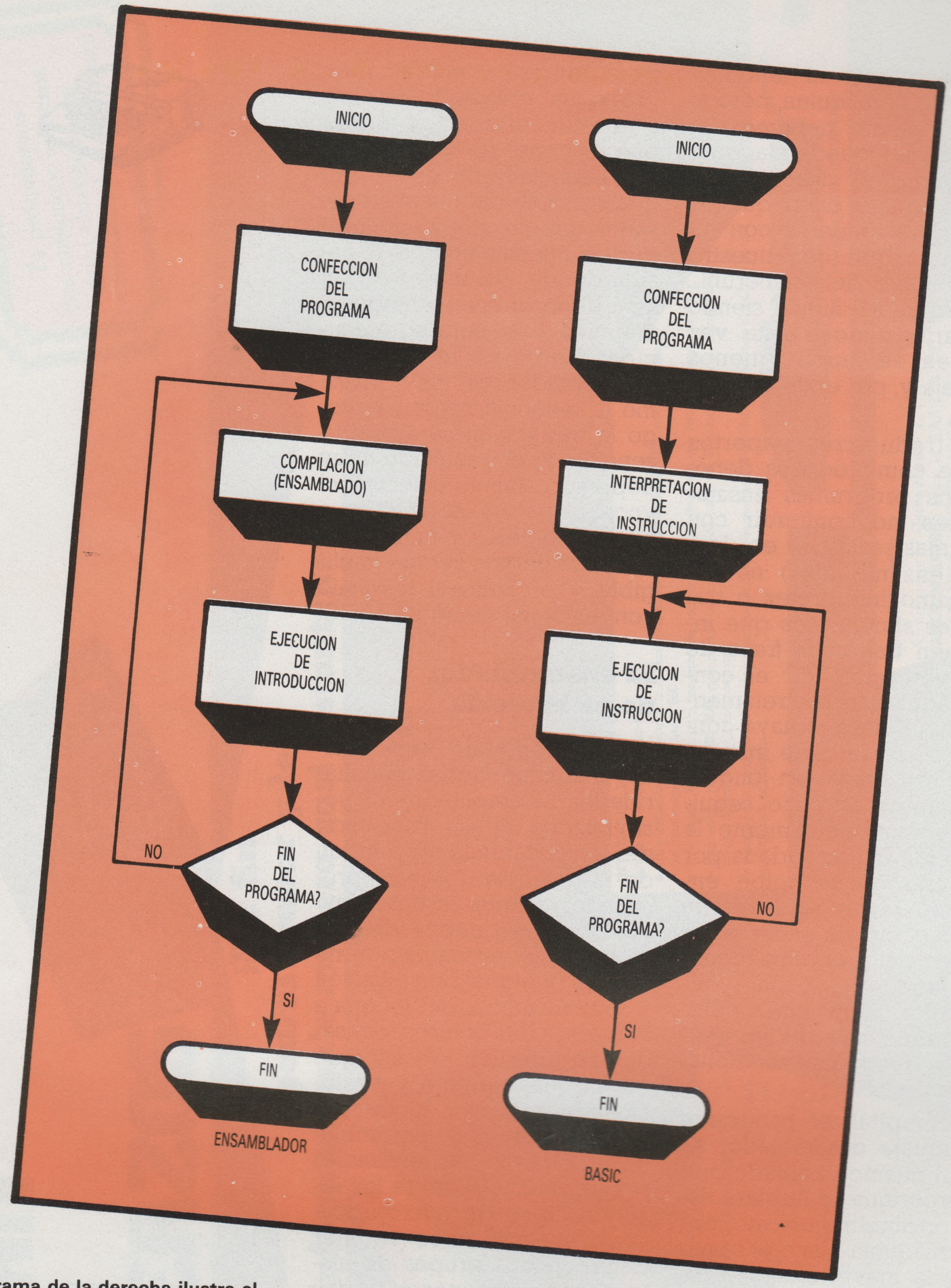
Retomando el hilo por el cabo del microprocesador, cabe añadir que para convertir a lenguaje máquina un programa redactado en lenguaje ensamblador, habrá que disponer de un compilador especializado en el lenguaje propio del microprocesador, que constituye el cerebro de nuestro equipo. El resultado de su intervención será el programa en lenguaje máquina, listo para su ejecución. Ventajas: el lenguaje ensamblador es muy próximo al lenguaje máquina, luego la traducción será rápida, sencilla y el resultado muy depurado... Inconvenientes: el vocabulario no se reduce a hileras de ceros y unos, pero tampoco llega a ser tan inteligible y próximo al lenguaje convencional como el BASIC.



El ensamblador no es brujería

Cada lenguaje ensamblador mantiene ciertas peculiaridades. La amplitud y las posibilidades de su repertorio de instrucciones dependen de la potencia del microprocesador al que está destinado. El motivo deriva de que el lenguaje ensamblador no es más que una transposición a «mnemónicos» del lenguaje máquina del microprocesador en cuestión, de ahí que cuanto más potente y evolucionado sea éste, tanto más lo será su lenguaje ensamblador. Salvando las distancias —¡que las hay!— entre los diversos tipos de microprocesadores, podemos señalar determinados grupos de instrucciones comunes a los distintos lenguajes ensambladores:





El diagrama de la derecha ilustra el proceso de traducción y ejecución de un programa redactado en lenguaje ensamblador. A la izquierda aparece reflejado el mismo proceso, aunque partiendo en este caso de un programa en lenguaje BASIC. Por supuesto, nos remitimos a la secuencia de trabajo habitual en el campo de los pequeños microordenadores dotados de un intérprete de lenguaje BASIC. Como se observa en la figura, el inconveniente de la traducción del programa BASIC por medio de un intérprete radica en que si una instrucción se ejecuta, por ejemplo, un millón de veces dentro del programa, el intérprete se verá obligado a traducirla un millón de veces a código máquina. El intérprete no recuerda las instrucciones que ya ha sometido a un proceso de traducción.

— Instrucciones de lectura/almacenamiento en memoria (similares al PEEK y POKE del BASIC).
— Instrucciones de asignación de registros (similares al LET del lenguaje BASIC).
— Instrucciones de bifurcación (similares al IF del BASIC).
— Instrucciones de transferencia de control (similares al GO TO y GOSUB del BASIC).

Como observamos, el lenguaje ensambldor dispone de funciones similares a las habituales en un lenguaje evolucionado como es el BASIC, aunque a un nivel mucho más bajo, más próximo a la máquina. En consecuencia, no cabe pensar en qué extraña magia se desencadena cuando realizamos un SYS o un USR; recordemos que estas instrucciones BASIC representan tan solo un «RUN» u orden de ejecución para determinadas rutinas en código máquina y nada más... ni misterios, ni prestidigitación informática, sino que simplemente salta a escena un lenguaje que nos resulta desconocido.

VENTAMATIC NEWS

VENTAMATIC Venta por correo: Avda. De Rhode, nº 253 ☎ (972) 255 616 **ROSES** (Girona)

Tienda y cursos de **BASIC** y código máquina: C/ Córcega, nº 89, entlo. **BARCELONA-29**

INAUGURACIÓN DE LA NUEVA TIENDA VENTAMATIC EN BARCELONA.

Desde principios de abril, la nueva tienda VENTAMATIC en Barcelona, situada en la calle Córcega, nº 89, entlo., abre sus puertas al público, ofreciendo la más amplia gama de accesorios, programas y libros para el **ZX-SPECTRUM** y el **ZX81**, nacionales y de importación, además de extraordinarias novedades exclusivas de **VENTAMATIC**. Todo ello con los fantásticos precios **VENTAMATIC**, incluyendo los propios ordenadores y con garantía **VENTAMATIC** válida en todo el territorio nacional.

También están disponibles los micro-ordenadores **COMMODORE 64** y **ORIC ATMOS**, con fantásticas ofertas y todos sus accesorios, programas y libros, incluyendo novedades exclusivas de **VENTAMATIC.**

El horario comercial de la nueva tienda es de 10 a 1,30 de la mañana y de 4 a 8 de la tarde.

SPECTRUMANIA, REVISTA EN CASSETTE. NOVEDAD EN ESPAÑA.

VENTAMATIC acaba de lanzar el primer número de su nueva revista en cassette para el **ZX-SPECTRUM, SPECTRUMANIA**. Se trata de la primera publicación de este tipo editada en España y sus ventajas son extraordinarias: se evita el fastidioso trabajo de teclear los programas de las revistas que pueden contener errores difíciles de detectar o no responder a lo que se esperaban, se pueden ver pantallas reales de los juegos y programas comercializados en la sección de revisiones de programas, se pueden aprovechar directamente las rutinas utilizadas en la elaboración y presentación de los programas, etc. En el primer número encontramos los siguientes programas: **INVADERS, COMPOSITOR, SUPERSTORE, MAQUINA DE FRUTAS, GENERADOR DE CARACTERES**, etc. y pantallas de algunos de los mejores juegos comercializados, además de revisiones de accesorios, cartas, trucos, rutinas, etc. Todo para los modelos de **16K** y **48K** al increíble precio de **750.- ptas.** Suscripción por tres números: **1.500.- ptas.** Suscripción por **6 números: 3.000.- ptas.** Distribuidores interesados en distribuir este producto, contacten con **VENTAMATIC.**

****

TECLADO PROFESIONAL PARA ZX-SPECTRUM

Por fín ha llegado el teclado **STONECHIP** que entre sus teclas de tamaño normal de máquina de escribir, serigrafiadas en distintos colores igual que en el teclado original del **ZX-SPECTRUM**, incluye barra espaciadora, tecla simple de **DELETE**, tecla simple de **MODO EXTENDIDO** y doble tecla de **RESET**. Además lleva incorporado amplificador de sonido con altavoz y controles de tono y volumen y commutador **SAVE/BEEP/LOAD**. El **ZX-SPECTRUM** *con caja y todo, sin desmontarlo ni anular la garantía*, se monta en el interior de la caja del teclado, permitiendo la conexión del **ZX-INTERFACE** 1 y los **ZX-MICRODRIVES** sin absolutamente ningún problema.

Precio: 16.000.- ptas.

NUEVA ETAPA DEL CLUB NACIONAL DE USUARIOS DE LOS ZX

Con el boletín nº 10, *El Club Nacional de Usuarios de los ZX* inaugura una nueva etapa en formato revista editado e impreso profesionalmente, con un contenido más ameno y práctico, más sorteos, más premios, más secciones, más programas y mucha mayor dedicación al **ZX-SPECTRUM**. En el número 11 se puede encontrar una entrevista con **MATTEW SMITH**, autor de **MANIC MINER** y **JET SET WILLY**, la primera parte del listado de toda la bibliografía del **ZX-SPECTRUM**, revisión del tablero digital para **ZX-SPECTRUM** y un informe detallado sobre el funcionamiento del **ZX-INTERFACE 1** y **ZX-MICRODRIVE**, además de las habituales secciones de *Correo de los Socios*, Realizaciones, Bolsa del Club, Novedades, Soft-bank, Soft-Stars, Consultas, etc. Suscripción boletines **11-16** (ambos inclusives): **2.500.- ptas.** Boletines atrasados: **300.- ptas** cada uno. También más descuentos para los socios en compras a **VENTAMATIC.**

DISPONIBLES LOS MEJORES PROGRAMAS INGLESES EN VERSION ORIGINAL

Ya están aquí los números 1 en Inglaterra:

- ▶ JETPAC, PSSST, LUNAR JETMAN, ATIC ATAC, COOKIE y TRANZ-AM de ULTIMATE.
- ▶ CHUCKIE EGG de A&F.
- ▶ CYRUS-IS-CHESS de INTELLIGENT SOFTWARE.
- ▶ THE HOBBIT de MELBOURNE HOUSE.
- ▶ VALHALLA de LEGEND.
- ▶ THE PYRAMID y DOOMSDAY CASTLE de FANTASY.
- ▶ ROMMEL'S REVENGE de CRYSTAL.

Existencias limitadas. Solicita lista de precios vigente.

También a punto de llegar:

- ▶ FIGHTER PILOT de DIGITAL INTEGRATION.
- ▶ DEATHCHASE y CODENAME MAT de MICROMEGA.
- ▶ POGO de OCEAN.
- ▶ WHEELIE de MICROSPHERE.

LOS MEJORES JUEGOS PARA EL ZX-SPECTRUM, PRODUCIDOS AHORA EN ESPAÑA.

SOFTWARE PROJECTS, la firma inglesa que comercializa los video-juegos para el **ZX-SPECTRUM** considerados como los mejores en Inglaterra, **MANIC MINER** y **JET SET WILLY**, ha concedido licencia exclusiva de producción de las versiones españolas a **VENTAMATIC**, así como de sus demás video-juegos para el **ZX-SPECTRUM: THRUSTA, PUSH-OFF, TRIBBLE TROUBLE, ORION** y **OMETRON.** Precio de cada programa: **1.700.-ptas.** Versiones de **MANIC MINER, JET SET WILLY** y otros video-juegos de **SOFTWARE PROJECTS** para el **COMMODORE 64** en preparación.

EL ZX-MICROVIDE, DISPONIBLE POR FIN

Precio **ZX- INTERFACE 1: 17.500.- ptas.,**
Precio **ZX-MICRODRIVE: 17.500.- ptas.**
Con garantía **VENTAMATIC**, válida en todo el territorio nacional.

EXTRAORDINARIA OFERTA ZX81

El **ZX81** con módulo de **16K** más **5** cassettes de programas por sólo **17.950.- ptas.** Completo con alimentador, cables para TV y cassette, manuales en inglés y en castellano, y garantia **VENTAMATIC** válida en todo el territorio nacional.

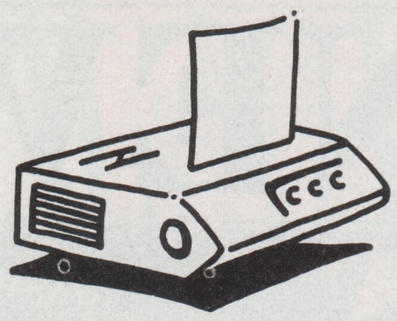
NOVEDADES VENTAMATIC

✪ **BYTE DRIVE 500** para **ORIC 1** y **ATMOS**: Drive de 3'' con capacidad de **440K** formateados y muchos comandos avanzados de disco, incluyendo ficheros secuenciales y aleatorios. Precio: **90.000.- ptas** (incluye Drive, Fuente de alimentación para 2 drives, interface, disco del D.O.S. y manual). Con solo cambiar el interface será compatible con **ZX-SPECTRUM** o **COMMODORE 64.**

✪ **ORIC ATMOS: 59.000.- ptas. ORIC COLOUR PRINTER: 59.000.- ptas.** Disponibles también otros periféricos, libros y programas.

✪ **COMMODORE 64: 79.900.- ptas.**. Disponibles también todos sus periféricos, libros y programas.

✪ **TABLA DIGITALIZADORA** para **ZX-SPECTRUM**: Para dibujar sobre ella y que aparezca el dibujo en pantalla. Se suministra con un completo programa de diseño asistido por ordenador con una extensísima gama de comandos y ayudas gráficas. También disponible para **ORIC** y **COMMODORE 64: 39.000.- ptas.**



BASIC, ENSAMBLADOR Y CODIGO MAQUINA:

En algunas ocasiones, el equivalente en lenguaje ensamblador de un programa BASIC puede ser incluso sencillo. Un ejemplo lo constituye la siguiente rutina, destinada a alterar el color del borde de la pantalla de presentación en el SPECTRUM. Al ejecutarla, el color del marco exterior a la zona de visualización de datos pasa del blanco al negro, atravesando toda la escala de colores disponibles en el equipo.

Dada la sencillez del programa en lenguaje ensamblador podemos realizar su conversión a código máquina «a mano», sin la colaboración de un experto ensamblador. Para facilitar la tarea, no vamos a traducirlo directamente a hileras de ceros y unos, sino que expresaremos las palabras o códigos del lenguaje máquina en su equivalente en el sistema de numeración decimal.

Lenguaje ensamblador	Lenguaje máquina CO	OP	Descripción
LD B, $7	6	7	Carga del registro B (código de operación 6) con el dato 7.
LD A, B	120		Carga del acumulador con el contenido del registro B (C. O. = 120).
OUT 254, A	211	254	Salida del contenido del acumulador (C. O. = 211) a través del «port» localizado en la dirección 254.
DJNZ LOOP	16	252	Decrementar el contenido del registro B (C. O. = 16) y regresar a LOOP mientras el contenido sea distinto de cero. El operando 252 es el complemento a dos del salto a realizar para llegar a LOOP (-4 = 256-4).
LD A, B	120		Carga del acumulador con el contenido del registro B (C. O. = 120).
OUT 254, A	211	254	(Igual que la tercera instrucción).
RET	201		Retorno al control del intérprete BASIC (C. O. = 201).

Notas:
CO = Código de operación.
OP = Operando.

PROGRAMA BASIC:

```
FOR I = 7 TO O STEP -1: BORDER I: NEXT I
```

PROGRAMA EN LENGUAJE ENSAMBLADOR:

```
BEGIN LD B, $7;     Carga el registro B con el código de color blanco.
LOOP LD A, B;       Hace el acumulador igual a B.
OUT 254, A;         Cambia el color del borde de pantalla.
DJNZ LOOP;          Decrementa en 1 el registro B y salta a LOOP si B
                    aun no es cero.
LD A, B;            Hace el acumulador igual a B, esto es, igual a cero.
OUT 254, A;         Cambia por última vez el color del borde.
RET;                Retorna al control del BASIC.
```

Hay que desterrar la idea de que aprender lenguaje ensamblador o código máquina exige una capacidad extraordinaria y amplios conocimientos informáticos. La base suficiente para coronar con éxito el aprendizaje es conocer nuestro equipo y armarse de ganas y paciencia.

El montón



El comentario sobre este programa va a ser algo más que somero por nuestra parte; en efecto, vamos a dejar que el ordenador se ocupe tanto de instruir a los jugadores como de convertirse, de pedirlo el usuario, en un duro contendiente. Simplemente, le adelantaremos que los dos jugadores se encontrarán ante un montón de piezas, del que deben ir retirando un número de ellas comprendido entre una y M (número máximo que establecerá el ordenador), por riguroso turno. El jugador que se hará acreedor a los laureles de la victoria será aquel que obligue al contrario a llevarse la última pieza del montón.

Y una advertencia: ¡cuidado cuando elija al LASER 200 como contrincante! Si la partida se desarrolla entre dos «humanos», el ordenador se limitará a instruirlos en las reglas del juego y a gestionar que la partida transcurra sin trampa alguna. No obstante, cuando se tiene al ordenador como oponente, la cosa es muy distinta. Este empezará pidiendo al usuario que introduzca el grado de dificultad del juego (niveles de 0 a 4) y, según el nivel elegido, así será su contundencia. El hecho de que el ordenador opte por realizar sus jugadas de forma aleatoria o evaluando antes cuál es la jugada óptima, depende del nivel de dificultad seleccionado: cuanto más alto sea éste, más difícil le resultará vencer a la máquina y, desde luego, en el nivel más alto muy rara vez logrará Ud. subir al pódium.

Dado que el programa se ha confeccionado con la finalidad de que también actúe como instructor, está capacitado para dialogar en castellano. Así, a las preguntas que requieran una respuesta afirmativa o negativa, el jugador debe responder pulsando las telcas «S» o «N», respectivamente.

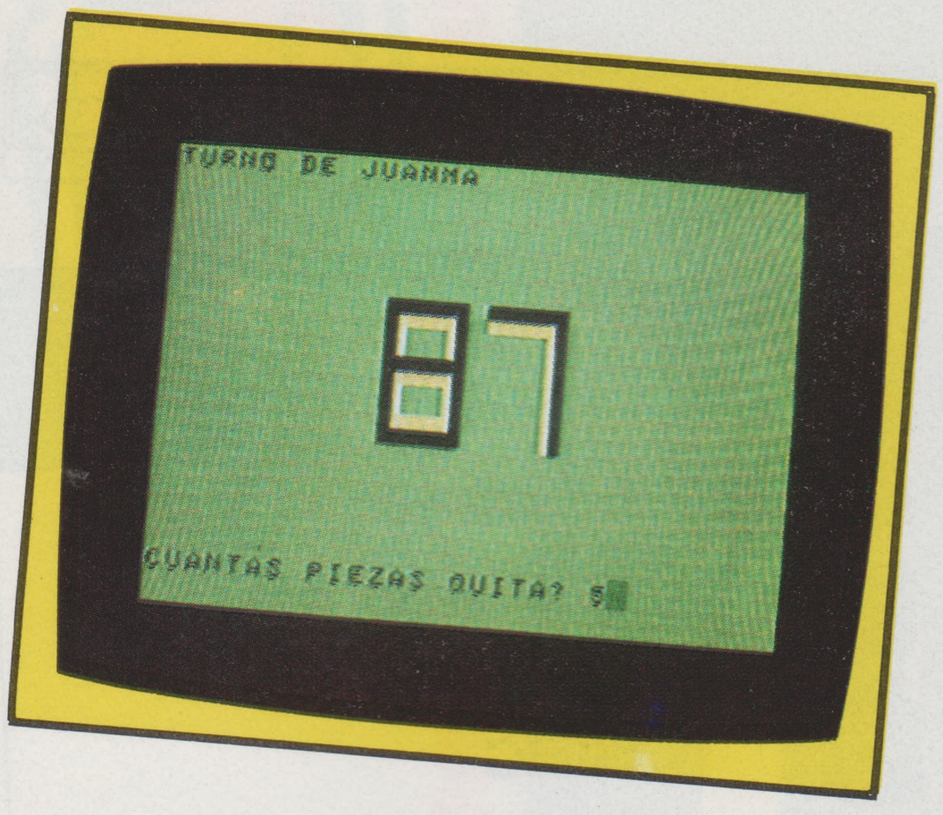
¡Suerte y a por el montón!

TABLA DE VARIABLES

A	Selecciona el turno de juego.
A$	Recogida del nombre de los jugadores.
B%	Número de piezas a descontar.
D$	Grado de dificultad.
I	Variable utilizada en las estructuras FOR.
N	Número de piezas que restan.
M	Movimiento máximo permitido.
S	Apunta a la posición donde escribir los caracteres gigantes.
X	Recoge el valor numérico a mostrar en pantalla.
X$	Recogida de datos diversos.

ESTRUCTURA DEL PROGRAMA

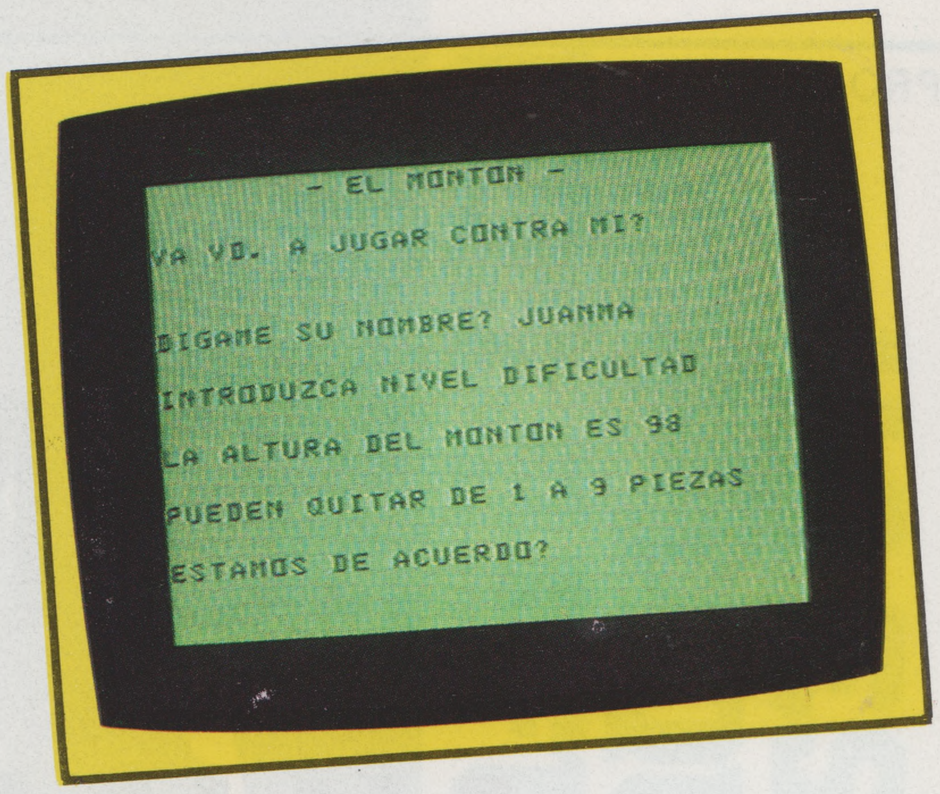
10-140	Introducción de datos.
150-250	Presentación de las condiciones iniciales del juego.
260-440	Selección de turno y comprobación de fin de juego.
500-520	Ejecución de un movimiento aleatorio.
450-490	Ejecución de un movimiento calculado.
530-800	Gestión de la presentación en caracteres gigantes.
802-804	Control del número de piezas a quitar en el caso de jugar contra un adversario distinto de la máquina.
810-1260	Selección del número a mostrar en la pantalla.



En efecto, el mismísimo programa se encarga de instruir al jugador. La fotografía muestra la pantalla de entrada al juego, en donde se solicita la introducción de los datos previos (nombre del jugador y grado de dificultad) y se comunica a los contendientes la altura del montón y el número de piezas que pueden retirarse en cada turno.

LISTADO

```
10 REM P.M. DEL MORAL - LASER 200 - EL MONTON
20 CLS
30 PRINT@8,"- EL MONTON -"
40 PRINT
45 IF INKEY$<>"" THEN 45
50 PRINT "VA A JUGAR CONTRA MI?"
60 X$=INKEY$ : IFX$<>"S" AND X$<>"N" THEN 80
70 PRINT : PRINT
80 INPUT "DIGAME SU NOMBRE";A$(0)
90 IF X$="S" THEN A$(1)="LASER":GOTO 110
100 INPUT "Y EL DE SU CONTRARIO";A$(1)
110 PRINT
120 IF X$="N" THEN 160
130 PRINT "INTRODUZCA NIVEL DE DIFICULTAD"
140 D$=INKEY$ : IF D$<"0" OR D$>"4" THEN 140
160 N=20+RND(79)
170 M=5+RND(10)
180 PRINT
190 PRINT "LA ALTURA DEL MONTON ES "N
200 PRINT
210 PRINT "PUEDE QUITAR DE 1 A "M" PIEZAS"
220 PRINT
230 PRINT "ESTAMOS DE ACUERDO?"
240 Z$=INKEY$: IF Z$<>"S" AND Z$<>"N" THEN 240
250 IF Z$="N" THEN PRINT@224,"": GOTO 160
260 A=RND(2)-1
270 CLS
280 PRINT "TURNO DE " A$(A) : SOUND 3,3
290 GOSUB 530
300 IF X$="S" AND A THEN GOSUB 450: GOTO 330
310 PRINT@447,"": INPUT "CUANTAS PIEZAS QUITA"; B%
320 IF B%<1 OR B%>M THEN 310
330 N=N-B%
340 IF N<1 THEN 370
350 A=A+1: IF A=2 THEN A=0
360 GOTO 270
370 CLS
380 A=A+1: IF A=2 THEN A=0
390 SOUND 1,3: SOUND 4,2: SOUND 3,3: SOUND 4,2: SOUND 1,3
400 PRINT@ 233-INT(LEN(A$(A))/2), "ENHORABUENA "A$(A)"!"
410 PRINT@201, "DESEA JUGAR OTRA VEZ?"
420 Z$=INKEY$: IF Z$<>"S" AND Z$<>"N" THEN 420
430 IF Z$="S" THEN 20
440 END
450 IF RND(0)*4>VAL(D$) THEN 500
460 E=INT((N-1)/(M+1))
470 B%=N-1-E*(M+1): IF B%=0 THEN 500
480 GOSUB 602
490 RETURN
500 B%=RND(M)
510 GOSUB 602
520 RETURN
```



¡Muy importante!: los caracteres que aparecen subrayados en el listado del programa corresponden a los símbolos gráficos de las teclas indicadas. En consecuencia, un caracter que aparezca subrayado se introducirá en el LASER 200 accionando, simultáneamente, la tecla SHIFT y la tecla cuyo símbolo principal es la letra subrayada.

```
530 N$="0"+MID$(STR$(N), 2, LEN(STR$(N))-1)
540 N$=RIGHT$(N$,2)
550 FOR I=1 TO 2
560 S=5+5*1
570 X=VAL(MID$(N$,1,1))
580 GOSUB 610
590 NEXT
600 RETURN
602 PRINT@448, "CUANTAS PIEZAS QUITA?"; B%;
603 FOR I=0 TO 1999: NEXT
604 RETURN
610 IF X=0 THEN 1240
611 IF X=1 THEN 1000
612 IF X=2 THEN 1020
613 IF X=3 THEN 1050
614 IF X=4 THEN 1072
615 IF X=5 THEN 1090
616 IF X=6 THEN 1120
617 IF X=7 THEN 1150
618 IF X=8 THEN 1180
619 IF X=9 THEN 1210
1000 PRINT@163+S,"U": PRINT@195+S,"U": PRINT@227+S,"U"
1005 PRINT@259+S,"U"; PRINT@291+S,"U"
1010 RETURN
1020 PRINT@161+S,"YYYS": PRINT@196+S,"U": PRINT@226+S,"YY"
1030 PRINT@257+S,"I": PRINT@289+S,"DTTT"
1040 RETURN
1050 PRINT@161+S,"YYYS": PRINT@196+S,"U": PRINT@225+S,"YYYS"
1060 PRINT@260+S,"U": PRINT@289+S,"TTTF"
1070 RETURN
1072 PRINT@161+S,"I": PRINT@193+S,"I": PRINT@225+S,"DTTT"
1074 PRINT@260+S,"U": PRINT@292+S,"U"
1076 RETURN
1090 PRINT@161+S,"AYYY": PRINT@193+S,"I": PRINT@225+S,"YYYS"
1100 PRINT@260+S,"U": PRINT@289+S,"TTTF"
1110 RETURN
1120 PRINT@161+S,"AYYY": PRINT@193+S,"I": PRINT@225+S,"AYYS"
1130 PRINT@257+S,"I  U": PRINT@289+S,"DTTF"
1140 RETURN
1150 PRINT@161+S,"YYYS": PRINT@196+S,"U": PRINT@228+S,"U"
1160 PRINT@260+S,"U": PRINT@292+S,"U"
1170 RETURN
1180 PRINT@161+S,"AYYS": PRINT@193+S,"I  U": PRINT@225+S,"AYYS"
1190 PRINT@257+S,"I  U": PRINT@289+S,"DTTF"
1200 RETURN
1210 PRINT@161+S,"AYYS": PRINT@193+S,"I  U": PRINT@225+S,"DTTF"
1220 PRINT@260+S,"U": PRINT@289+S,"TTTF"
1230 RETURN
1240 PRINT@161+S,"AYYS": PRINT@193+S,"I  U": PRINT@225+S,"I  U"
1250 PRINT@257+S,"I  U": PRINT@289+S,"DTTF"
1260 RETURN
```

END

SIMON NUMERICO

Después de un duro enfrentamiento con el ordenador, intentando vencer su fría y rápida eficacia en los juegos de acción, nuestros reflejos necesitan un bien merecido reposo. Descanso que no exige el abandono de tan incondicional acompañante en los ratos de ocio y que podemos aprovechar para comprobar cómo anda nuestra memoria recordando un número.

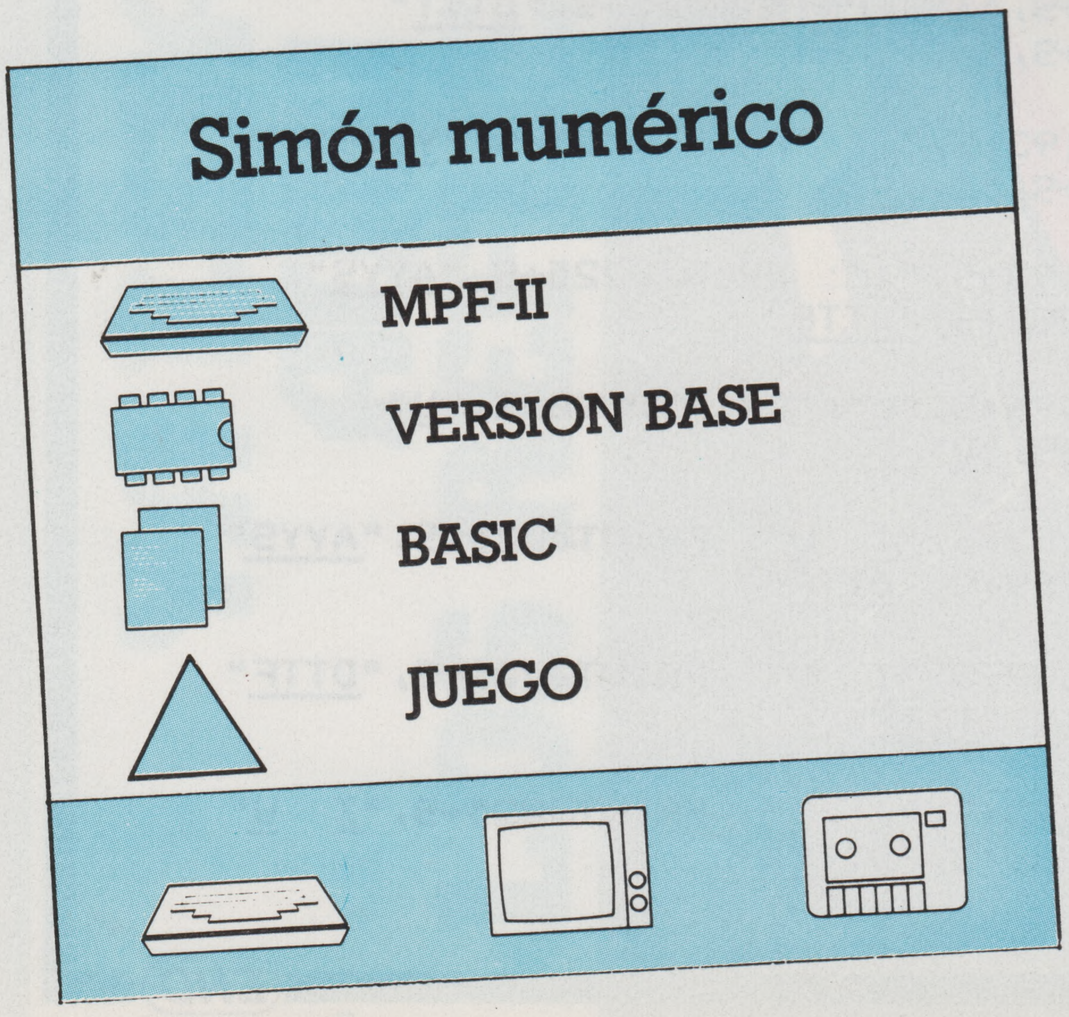
El propósito de este programa no es otro que comprobar hasta que punto la suya es una «memoria de elefante». Desde luego que tan voluminoso paquidermo no puede exhibir sus facultades retentivas accionando las teclas oportunas... ¡cuestión de deditos! No obstante Ud. sí puede, aunque su capacidad de memoria deje aún bastante que desear. Es posible que su actuación empiece siendo muy floja, pero, al fin y al cabo, es con la práctica como se puede llegar a educar a tan interesante rincón de la mente... ¡No dude en intentarlo!

Recordando números

El programa es una versión del popular SIMON que, en este caso, va a poner a prueba su capacidad para recordar una serie de números, por supuesto, concediéndole un tiempo limitado que puede seleccionar el propio jugador.

Convendrá con nosotros en que éste no es mal momento —recién terminada la declaración de Hacienda— para inundar de números su cabeza, una vez más... ¡Y no piense en rendirse!, recuerde que, en el fondo, el ordenador no es más que una caja de tuercas y cables; el ser humano —¡Ud. por ejemplo!— es muy superior, capaz de regalarse con ideas abstractas e incluso de hacerle trampa al mismísimo ordenador.

Hablemos de trampas, ¿por qué no? Dentro del programa hemos colocado una interesante subrutina cuya presencia pasa totalmente desapercibida para el propio SIMON. Suponga que, después de repetidos intentos, su empa-



cho de números amenaza con convertirse en indigestión. En tal caso, nuestro más preciado consejo es que aguarde a que SIMON le pida que introduzca la serie de números, en este punto responda con un STOP (CONTROL-C) y, a continuación, teclee la instrucción GOTO 2000. A partir de ahí, la astuta subrutina se ocupará de demostrar dónde está el ingenio: le indicará cual es la serie de números correcta, antes de devolverle al punto del programa en el que lo abandonó... ¡La victoria está ya en su mano!

Si intenta dar una exhibición de su capacidad de memoria ante un grupo de amigos, no dude en utilizar la mencionada subrutina... el efecto puede ser contundente.

El programa

La zona de programa que finaliza en la línea 400 se ocupa exclusivamente de generar las pantallas de presentación y de visualizar los créditos. A partir de la línea 520, hasta la 565, el ordenador pide al usuario que defina las condiciones del juego, seleccionando el número de términos, número de jugadas y velocidad a la que aparecerán los números.

El desarrollo del juego está programado en la zona delimitada por las líneas de instrucción 565 y 727, en las que el ordenador crea la serie de números y los visualiza en la pantalla (líneas 570-650, auxiliadas por las subrutinas localizadas entre 1000 y 1090). La petición de jugada —repetición de la serie de números propuesta— corre a cargo de las líneas 680 a la 720. Unicamente en el caso de que el jugador introduzca una serie de números equivocada, el programa quedará interrumpido. En estas condiciones, la secuencia de ejecución saltará a la línea 780 y sucesivas que constituyen la salida del programa, con opciones finales como la de ver cual es la serie de números correcta o volver a realizar una partida. Por último queda nuestra «tramposa» subrutina, localizada a partir de la línea 2000.

Unos consejos para terminar

Los lectores que deseen sustituir los números por colores, en un intento de aproximar nuestro juego al tradicional SIMON, pueden empezar introduciendo una instrucción COLOR = A(I) en la línea 586. A partir de ahí, con paciencia y recordando la codificación de colores del equipo, el usuario puede variar la gama de colores añadiendo la instrucción COLOR y el número de color deseado antes de cada subrutina. Una tarea algo delicada pero que puede dar mayor fidelidad al juego.

TABLA DE VARIABLES

A	Número de términos de cada serie de números.
A ( )	Vector conteniendo la serie de números de cada jugada.
I	Variable auxiliar contadora de bucles.
K	Variable para determinar la velocidad del juego.
T	Variable auxiliar para gráficos y bucles.
Z	Variable auxiliar como retardadora del juego.
J	Número máximo de jugadas.
H	Variable auxiliar contadora de jugadas.
S N $	Variable utilizada como opción para volver a jugar.
M	Término que introduce el jugador.

LISTADO

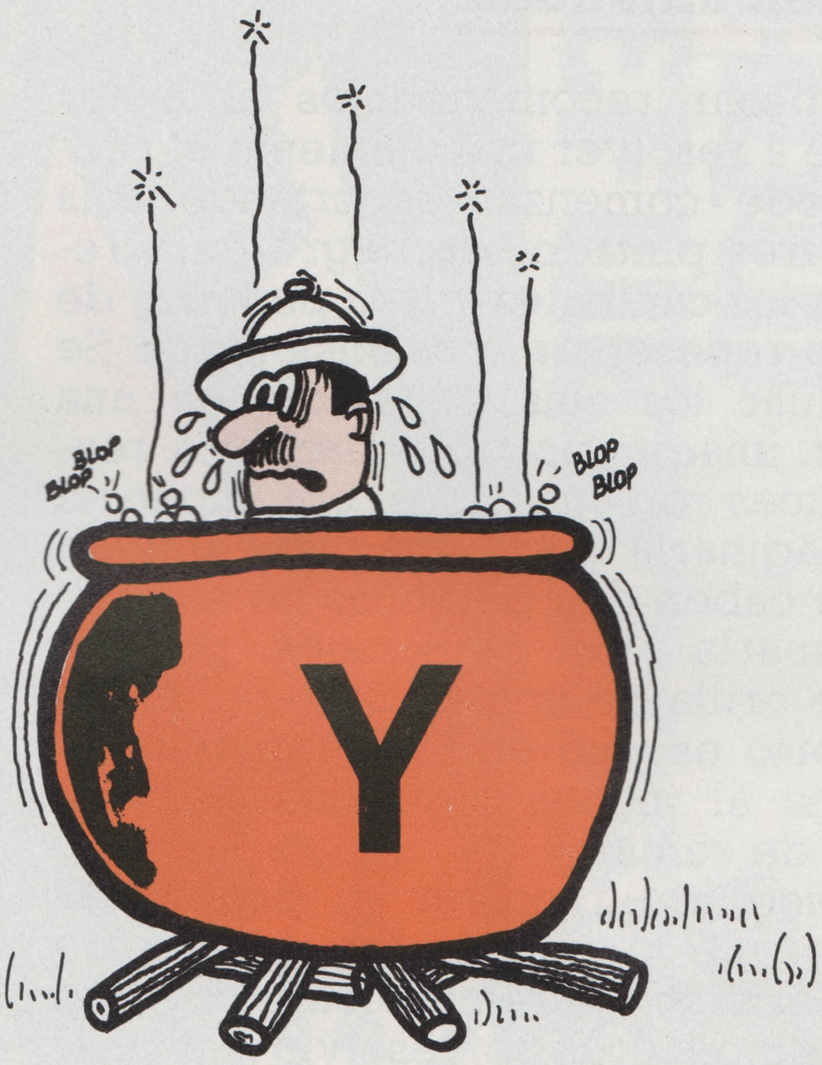


```
10 REM JUEGO SIMON NUMERICO *JLM VAZQUEZ
 DE PARGA* MPF II
20 HOME:DIM A(50)
30 GR
40 COLOR=15
50 VLIN 2,6 AT 8
60 VLIN 6,9 AT 12
70 FOR X=1 TO 8
80 READ Z
90 VLIN 2,9 AT Z
100 NEXT X
110 DATA 14,16,19,22,24,27,29,33
120 HLIN 8,12 AT 2
130 HLIN 8,12 AT 6
140 HLIN 8,12 AT 9
150 HLIN 16,22 AT 2
160 HLIN 24,27 AT 2
170 HLIN 24,27 AT 9
180 HLIN 29,33 AT 2
190 FOR Z=1 TO 1000:NEXT Z
200 FOR X=1 TO 8
210 READ T
220 VLIN 12,17 AT T
230 NEXT X
240 DATA 7,12,14,19,23,27,30,32
250 HLIN 7,9 AT 12
260 HLIN 7,9 AT 17
280 HLIN 15,17 AT 12
290 HLIN 15,17 AT 15
300 HLIN 15,17 AT 17
310 PLOT 17,16
320 HLIN 21,25 AT 12
330 HLIN 28,29 AT 12
340 HLIN 28,29 AT 15
350 HLIN 33,35 AT 17
355 VLIN 13,16 AT 10
360 FOR Z=1 TO 1000:NEXT Z
370 VTAB 16
380 PRINT:PRINT:PRINT TAB(5)"JOSE LUIS M
 VAZQUEZ DE PARGA"
390 FOR Z=1 TO 2000:NEXT Z
400 TEXT
520 HOME
530 PRINT"CUANTOS TERMINOS DESEA? (2/8)
":GET A
532 PRINT:PRINT:PRINT"PUEDE ESCOGER LA V
ELOCIDAD A LA QUE APA RECE LA SECUENCIA
DE NUMEROS"
534 PRINT:PRINT TAB(6)"1. MUY LENTO"
536 PRINT:PRINT TAB(6)"2. LENTO":PRINT:P
RINT TAB(6)"3. NORMAL": PRINT:PRINT TAB(
6)"4. RAPIDO":PRINT:PRINT TAB(6)"5. PROF
ESIONAL":PRINT:PRINT
537 PRINT"ELIJA OPCION (1...5):":GET T
538 IF T=1 THEN K=1000
540 IF T=2 THEN K=650
541 IF T=3 THEN K=500
542 IF T=4 THEN K=300
543 IF T=5 THEN K=170
545 FOR Z=1 TO 1500:NEXT Z
550 HOME
555 INPUT"NUMERO DE JUGADAS? (5/10/20):"
;J
560 HOME
562 FOR H=1 TO J
567 FOR Z=1 TO 1000:NEXT Z
570 FOR I=1 TO A
580 A(I)=INT(RND(1)*10)
585 GR
590 ON A(I) GOSUB 1010,1020,1030,1040,10
50,1060,1070,1080,1090
595 IF A(I)=0 THEN GOSUB 1000
620 FOR Z=1 TO K:NEXT Z
625 TEXT
630 HOME:PRINT TAB(30)"TERMINO ":A-I+1
640 FOR Z=1 TO K:NEXT Z
650 NEXT I
655 TEXT
660 PRINT"AHORA ES SU TURNO PARA DAR LA
SECUENCIA DE NUMEROS"
670 PRINT:PRINT" (UNO A UNO Y EN ORDEN)"
680 FOR I=1 TO A
690 GET M
700 IF M <>A(I) GOTO 780
705 PRINT M:
710 PRINT CHR$(7)
720 NEXT I
724 K=K-30
727 NEXT H
730 PRINT:PRINT:HTAB(20):INVERSE
735 PRINT TAB(19)"FELICIDADES":NORMAL
740 PRINT:PRINT"SERIE CORRECTA. ";
750 PRINT"DESEA JUGAR DE NUEVO?        (S
/N)":GET SN$
760 IF SN$="S" GOTO 530
770 PRINT:PRINT"FIN DE PROGRAMA"
775 END
780 FOR I=1 TO 5
790 PRINT CHR$(7)
800 NEXT I
810 HOME
820 HTAB(20)
830 INVERSE
840 PRINT TAB(20)"ERROR"
850 NORMAL
860 PRINT:PRINT:PRINT"DESEA VER LA
SECUENCIA CORRECTA? (S/N):":GET B$
870 IF B$="N" GOTO 750
880 FOR I=1 TO A
890 PRINT A(I)
900 NEXT I
910 PRINT:PRINT:GOTO 750
920 END
1000 VLIN 8,16 AT 17:VLIN 8,16 AT 23:HLI
N 17,22 AT 8:HLIN 17,22 AT 16:RETURN
1010 VLIN 8,16 AT 20:RETURN
1020 VLIN 12,16 AT 17:VLIN 8,12 AT 22:HL
IN 17,22 AT 8:HLIN 17,22 AT 12:HLIN 17,2
2 AT 16:PLOT 17,9:PLOT 22,15:RETURN
1030 VLIN 8,16 AT 23:HLIN 18,23 AT 8:HLI
N 21,23 AT 12:HLIN 17,23 AT 16:PL  . ..
5:RETURN
1040 VLIN 8,12 AT 17:VLIN 8,16 AT 22:HLI
N 17,22 AT 12:RETURN
1050 VLIN 8,12 AT 18:VLIN 12,16 AT 23:HL
IN 18,22 AT 8:HLIN 18,23 AT 12:HLIN 17,2
3 AT 16:PLOT 17,15:RETURN
1060 VLIN 8,16 AT 17:VLIN 12,16 AT 23:HL
IN 17,22 AT 8:HLIN 17,23 AT 12:HLIN 17,2
3 AT 16:RETURN
1070 VLIN 8,16 AT 22:HLIN 18,22 AT 8:HLI
N 20,22 AT 12:RETURN
1080 VLIN 8,12 AT 18:VLIN 8,12 AT 22:VLI
N 12,16 AT 17:VLIN 12,16 AT 23:HLIN 19,2
1 AT 8:HLIN 19,21 AT 12:HLIN 18,22 AT 16
:RETURN
1090 VLIN 8,12 AT 18:VLIN 8,16 AT 23:HLI
N 18,23 AT 8:HLIN 18,23 AT 12:HLIN 17,23
 AT 16:PLOT 17,15:RETURN
2000 VTAB15:PRINT"SSSCH! (HACEMOS TRAMPA
S)":FOR T=1 TO A:PRINT A (T)::NEXT
2010 PRINT"MEMORIZALOS":FOR Z=1 TO 1000:
NEXT Z
2020 GOTO 690
```

END

MISIONEROS





CANIBALES

Un divertido juego informático

Sin duda, una de las partes mas fascinantes del cuerpo humano es el cerebro. Mediante él se pueden desarrollar actividades inteligentes, y, aun cuando ha sido estudiado con profusión, el conocimiento que tenemos acerca de su funcionamiento es muy limitado. Un problema más sencillo que comprender los procesos que desencadenan el funcionamiento mental es la cuantificación de la inteligencia humana. Los psicólogos han diseñado diversos tests y problemas que tan sólo pueden ser resueltos por seres inteligentes y según aumenta la complejidad del problema solucionado mayor es la inteligencia de la persona que lo ha conseguido resolver.

El problema que vamos a plantear a continuación está dentro de los que citábamos en el párrafo anterior, de ahí que para resolverlo sea necesario estar dotado de una cierta inteligencia. Supongamos que en una orilla de un río hay un determinado número de *misioneros* y el mismo número de *caníbales*. El río tan sólo se puede atravesar mediante una barca cuya capacidad es de dos personas, siendo una de las dos personas la encargada de remar. Tanto los misioneros como los caníbales saben remar y obedecerán ciegamente las órdenes dadas por uno de los misioneros.

El objetivo final es muy simple: cruzar a todos los misioneros y caníbales hasta la

otra orilla con el menor número posible de cruces. Para complicar un poco el problema tendremos que tener en cuenta que, en cualquier momento y orilla, el número de misioneros debe ser mayor o igual al de caníbales, de no cumplirse esta condición los pobres frailes serán engullidos por los feroces caníbales.

La solución «manual»

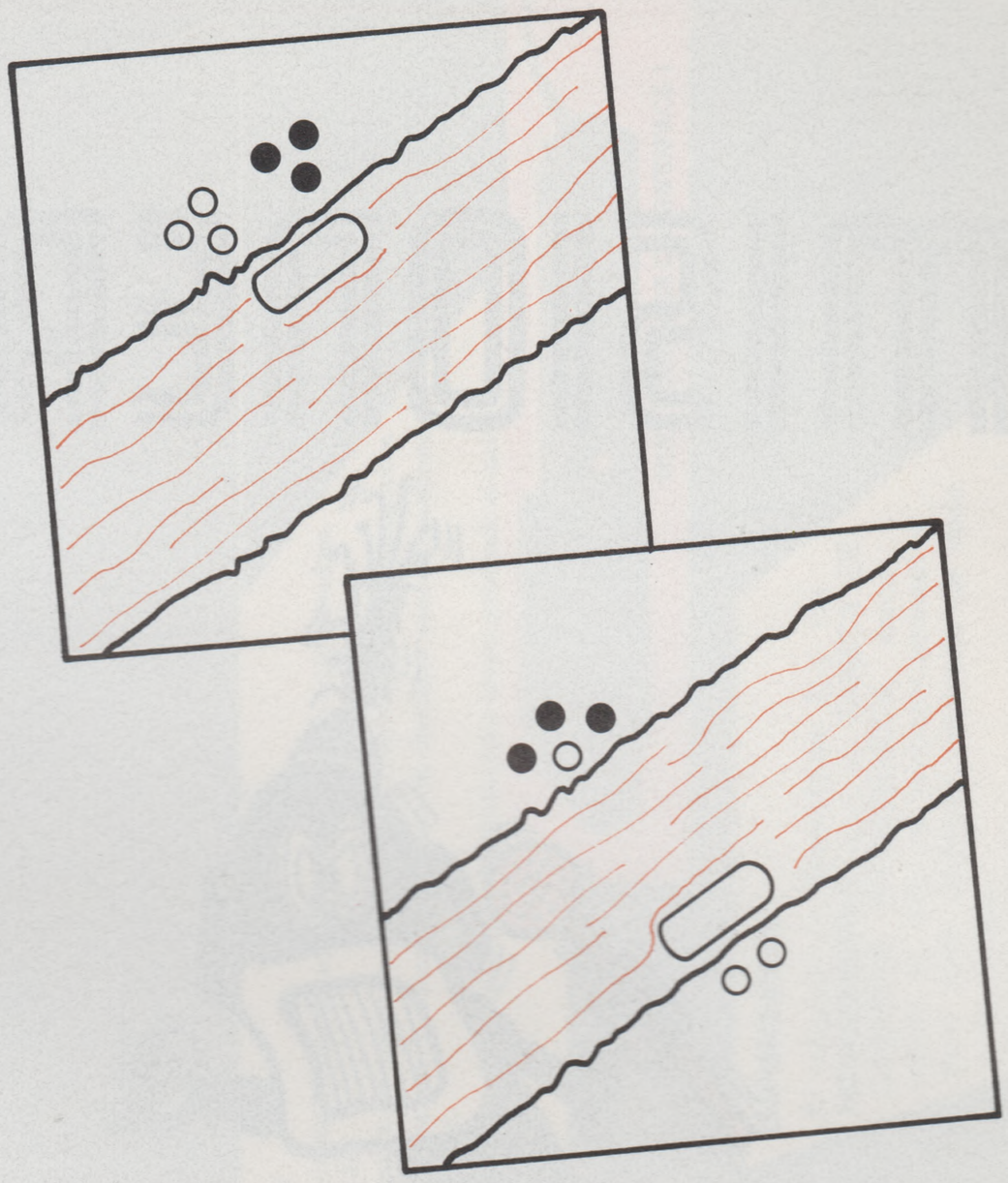
Para empezar recomendamos al lector que pruebe a resolver manualmente el problema; puede comenzar recortando seis papelillos, tres pintados de negro para representar a los caníbales y tres pintados de blanco para repesentar a los misioneros. Se pueden situar los seis papelillos en una orilla de un imaginario río y probar a realizar los cruces con una barca, por supuesto también imaginaría ...¡ah!, una pista: aunque en la barca caben dos personas no es obligatorio llenarla, los personajes pueden cambiar de orilla de uno en uno. Lo que sí es obligatorio es que en todo movimiento de la barca al menos vaya una persona encargada de remar.

¿Ha conseguido resolver el problema? Suponemos que sí. El número óptimo de cruces, en el caso de tres misioneros y tres caníbales, es once. Si lo resolvió con menos, comuníquenoslo... (¡es imposible!).

Como mencionábamos antes, para resolver este pequeño problema es imprescindible estar dotado de una cierta inteligencia. Partiendo de esta base, si conseguimos escribir un programa que realice los cruces oportunos la conclusión es clara: habremos escrito un programa inteligente. Es importante destacar que la mayoría de los programas de juegos para micro-ordenadores no son inteligentes, se limitan a representar gráficamente la situación y permiten al usuario que resuelva el problema, nosotros intentaremos que sea nuestro programa quien solucione el problema. Al aplicar el adjetivo de inteligente al programa —aunque parezca exagerado— lo hacemos con el máximo rigor de la palabra. En la rama matemática de las ciencias de la computación existen varios grupos dedicados al estudio de la *Inteligencia Artificial* y hoy no es ridículo hablar de programas de ordenador cuyo funcionamiento tiende a emular al de una mente humana.

Inteligencia programada

Centrémonos en la realización del programa. En primer lugar, tendremos que describir las variables que serán necesarias para representar el estado del problema. Por un lado, necesitaremos dos variables que representen el número de misioneros y caníbales, respectivamente, que



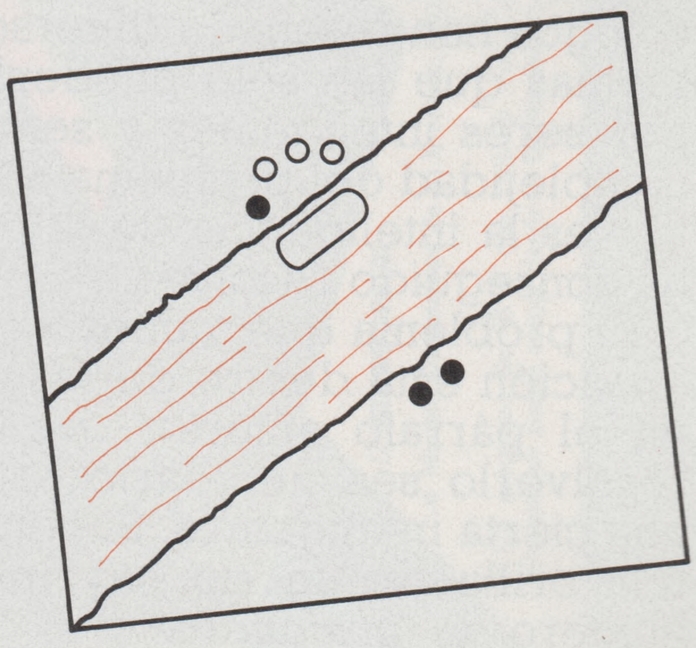
Si en el primer cruce con la barca desplazamos a dos misioneros desde la orilla-1 a la orilla-2, la tituación se convierte en caótica. En la orilla-1 quedan tres caníbales y un misionero; al estar los hambrientos caníbales en superioridad numérica se merendarán al indefenso fraile.

están situados en la orilla-1; a estas variables las llamaremos M y C.

Para contar el número original de misioneros y caníbales utilizaremos la variable N. No es necesario utilizar dos nuevas variables para representar a los misioneros y caníbales de la orilla-2, ya que se pueden deducir sin más que efectuar las restas N—M y N—C.

Por último, será necesaria una variable para saber en cuál de las dos orillas se encuentra la barca; a ésta la llamaremos B y valdrá 1 ó 2, según se encuentre en la primera o segunda orilla.

Así, pues, con el conjunto de variables M, C, B tenemos perfectamente determinada la situación en que nos encontramos.



Para representar la situación ilustrada utilizamos el conjunto (3, 1, 1,) que indica que en la orilla-1 hay tres misioneros y un caníbal y que la barca se encuentra también en la orilla-1.

TABLA 1

Situación inicial	M 1	C 1	B 1
Operación-1 (Pasar un misionero)	0	-1	2
Operación-2 (Pasar dos misioneros)	-1	1	2
Operación-3 (Pasar un caníbal)	1	0	2
Operación-4 (Pasar dos caníbales)	1	-1	2
Operación-5 (Pasar un misionero y un caníbal)	0	0	2

De las cinco posibles jugadas u operaciones, a realizar partiendo de la situación inicial indicada, las operaciones 2 y 4 son desechadas ya que conducen a situaciones imposibles (no puede haber misioneros o caníbales negativos en ninguna orilla). La situación 3 también es eliminada puesto que da lugar a una situación harto peligrosa para los misioneros (en la orilla-2 quedarán dos misioneros y tres caníbales). En definitiva, sólo son aceptables las operaciones 1 y 5, en el supuesto de que la situación a la que conducen no sea coincidente.

El programa tendrá que ser capaz de generar una sucesión de conjuntos de variables que termine en 0, 0, 2 , que significa: «*no quedan ni misioneros ni caníbales en la orilla-1, y la barca se encuentra en la orilla-2*».

Tanto si resolvemos el problema manualmente como si lo hacemos con un microordenador, en cada instante habrá que analizar a cuántas situaciones se puede pasar desde el estado en el que nos encontremos. Para ello habrá que generar cinco conjuntos de posibles situaciones, que se obtendrán realizando alguno de los siguientes cruces:

1. Pasar un misionero.
2. Pasar dos misioneros.
3. Pasar un caníbal.
4. Pasar dos caníbales.
5. Pasar un misionero y un caníbal.

Ahora bien, no todos los conjuntos generados con estas cinco operaciones corresponderán a situaciones permitidas, bien porque sea imposible realizar la operación, bien porque la situación alcanzada conduzca a un desenlace trágico o bien porque la situación producida es una de las que ya se ha elegido previamente y, por tanto, volver a elegirla no produciría ningún beneficio, en cambio podría hacer que el programa se metiera en un bucle infinito sin solución.

La siguiente operación a realizar consiste en suprimir los conjuntos incoherentes o insatisfactorios. Si se eliminan todas las operaciones posibles, bien sea por uno u otro motivo, será señal de que el problema no tiene solución o que no somos capaces de resolverlo. Esto es otra prueba de que el programa en cuestión es inteligente; al igual que en el ser humano no se sabe a ciencia cierta si es que el problema no tiene solución o que el ordenador no es lo suficientemente inteligente como para resolverlo.

El siguiente paso es el más delicado y, sin duda alguna imposible de dar correctamente sin recurrir a procedimientos semejantes a los usados por el cerebro humano. Hay que decidir, guiados por la intuición o la experiencia, cuál de las posibles situaciones válidas se elige. El proceso puede ser fatal si elegimos una situación que conduzca a punto sin solución. Nuestro inteligente

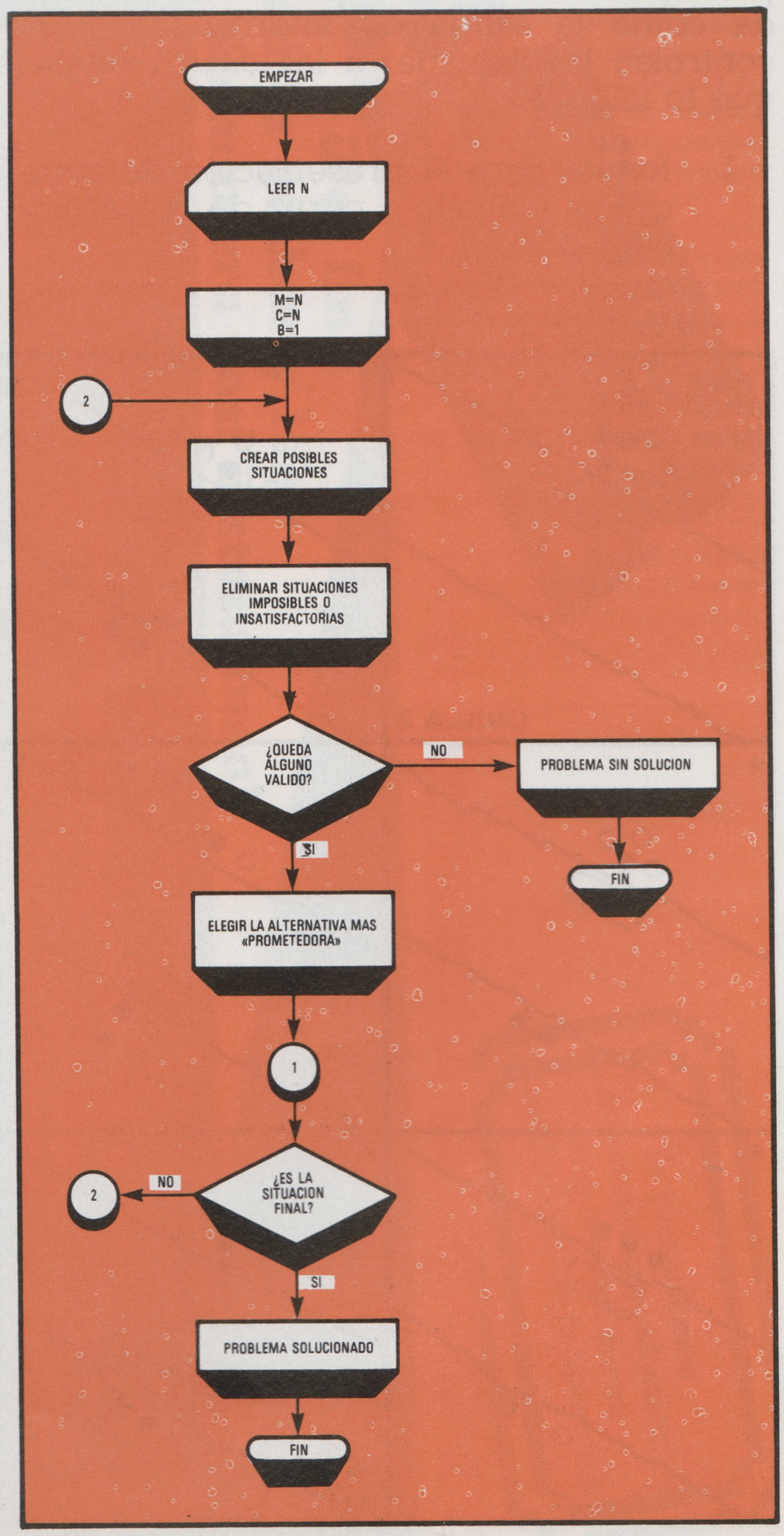


Diagrama de bloques básico del funcionamiento del programa que resuelve el problema de los misioneros y los caníbales.

programa tomará siempre la opción más próxima a la situación final, esto es: calculará C+M para cada una de las operaciones factibles y eligirá la que produzca un valor mas próximo a 0 en ambos casos. En caso de empate, se decidirá por la situación en que más igualado esté el número de misioneros y caníbales.

Una vez tomada la decisión anterior, sólo resta comprobar si la situación obtenida es la final. En caso afirmativo, el programa se enorgullecerá de sí mismo dado que ha conseguido resolver el problema; de no ser así, seguirá generando nuevos conjuntos de posibles situaciones.

Misioneros y caníbales en BASIC

Para escribir y ejecutar el programa en cuestión, hemos elegido el micro-ordenador SINCLAIR ZX-SPECTRUM y, por supuesto, el lenguaje utilizado será el BASIC.

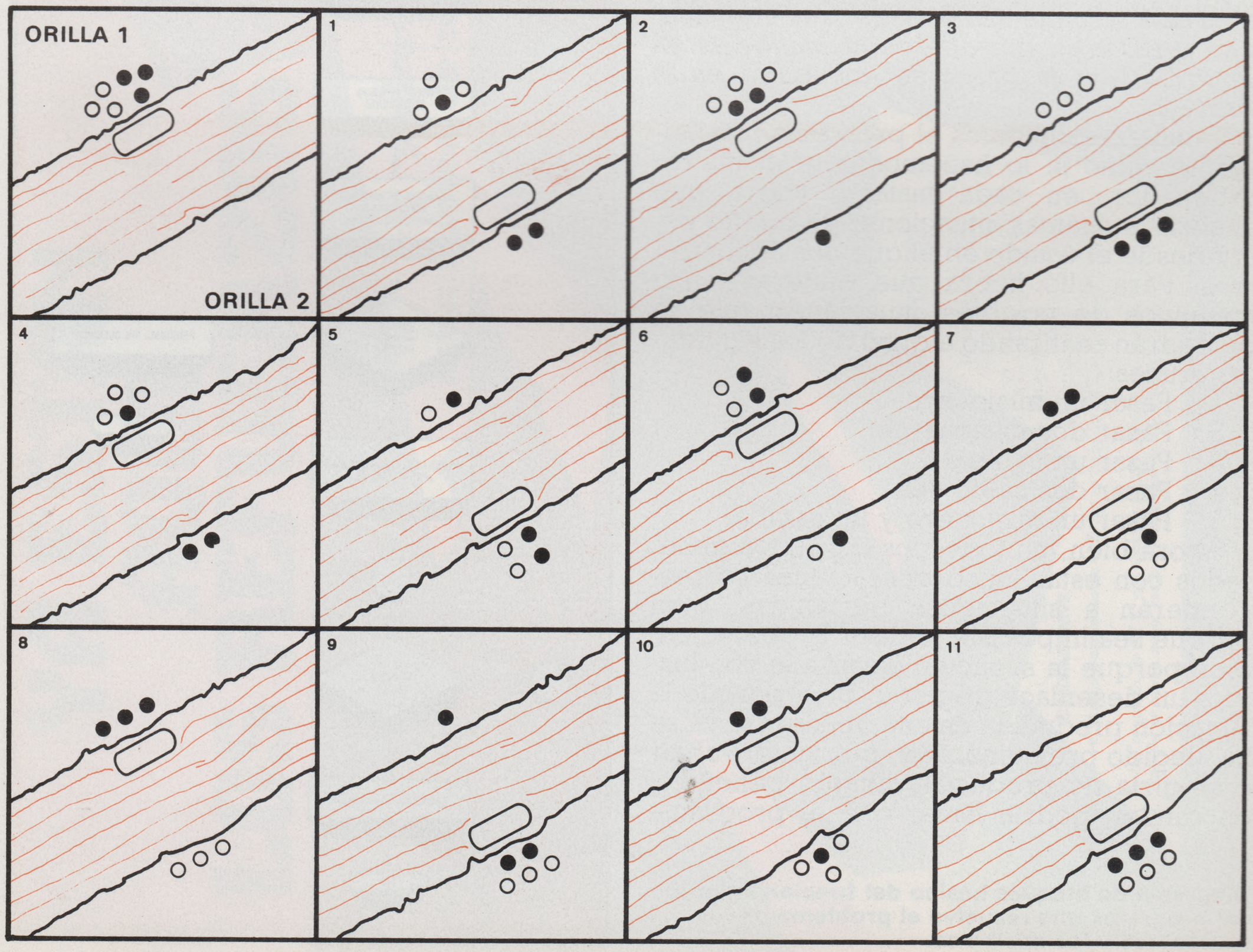
El programa (situado a partir de la línea de instrucción 10) consta de un cuerpo principal en el que se realiza tanto la entrada como la salida de información y se controlan las llamadas a las cuatro rutinas que lo integran:

1. Rutina para la presentación del programa (situada a partir de la instrucción 2000).

```
.
.
.
.
5050 IF s(i,4)=0 AND s(i,1)+s(i,2)<x THEN
     LET x=s(i,1)+s(i,2):
     LET j=i:
     LET d=ABS(s(i,1)-s(i,2))
5055 IF s(i,4)=0 AND s(i,1)+s(i,2)=x AND
     ABS (s(i,1)-s(i,2))<d THEN
     LET x=s(i,1)+s(i,2):
     LET j=i:
     LET d=ABS (s(i,1)-s(i,2))
.
.
.
.
.
```

La rutina que elige el «sucesor» o alternativa más prometedora es el auténtico «cerebro» del programa. Dentro de ella, las instrucciones 5050 y 5055 son las más importantes. Si el criterio establecido en estas instrucciones fuera erróneo, el programa puede llegar a ser incapaz de resolver el problema... ¡el programa sería poco inteligente!

La figura ilustra los once cruces de barca representativos de una de las diversas soluciones que admite el problema de los misioneros y los caníbales.



2. Rutina generadora de posibles sucesores (localizada a partir de la instrucción 3000).
3. Rutina para la eliminación de las situaciones imposibles o insatisfactorias (a partir de la instrucción 4000).
4. Rutina para la elección de la opción más «prometedora» (situada a partir de la instrucción 5000).

Si el lector quiere realizar un trasplante de cerebro al programa no tiene mas que modificar la rutina situada a partir de la instrucción 5000. Después, el programa podrá comportarse de tres formas:

1. No sabrá resolver los problemas que antes sí sabía, con lo cual el transplante no es satisfactorio... ¡hay rechazo al órgano del donante!
2. Los resolverá pero en un número superior de pasos, con lo que el programa seguirá siendo inteligente, pero menos que antes.
3. Los resolverá de forma mas eficiente, luego será mas inteligente que el programa original.

Por supuesto, si alguien consigue mejorar la inteligencia del programa esperamos que nos lo comunique para poder hacer público su algoritmo.

En todo el planteamiento realizado en este artículo hemos considerado variable el número de misioneros y caníbales que se encuentran inicialmente en la orilla -1. Si ejecutamos el programa propuesto para resolver el problema en un SINCLAIR ZX-SPECTRUM, podemos comprobar que cuando dicho número es 1, 2 ó 3 el programa es capaz de dar con la solución, en cambio, si el número es mayor de tres tan sólo es capaz de dar seis pasos y acaba respondiendo que el problema no tiene solución. Quedan tres preguntas abiertas:

1.ª ¿Verdaderamente el problema no tiene solución para N>3? o... ¿el programa no tiene inteligencia suficiente para resolverlo?
2.ª ¿Se pueden dar más de seis pasos, en el caso de que realmente no tenga solución el problema?
3.ª ¿Qué pasaría si el número inicial de misioneros y caníbales no fuera el mismo?

Pretendemos hacer de esta sección una ventana abierta a través de la que puedan participar y relacionarse los usuarios de micro-ordenadores; por ello, nos agradaría recibir tanto sus respuestas a los interrogantes planteados como cualquier sugerencia o variación relativa a los problemas que les iremos proponiendo. Con ello conseguiremos dos cosas: pasar un rato agradable y —¡como no!— mejorar nuestra técnica de programación.

tuMicro

LISTADO

```
 10 REM Canibal # Manuel Munoz #
 20 DIM z(50,3)
 30 DIM s(5,4)
 31 BORDER 6
 32 PAPER 3
 33 INK 6
 34 GO SUB 2000
 35 CLS
 40 FOR j=1 TO 3
 50 FOR i=1 TO 50
 60 LET z(i,j)=0
 70 NEXT i
 80 NEXT j
140 PRINT AT 1,3;"Teclee numero de "
150 PRINT AT 2,3;"canibales y misioneros"
160 INPUT n
170 CLS
180 PRINT AT 1,3;"     ORILLA 1  ORILLA 2"
190 PRINT AT 2,3;"     --------- ---------"
200 PRINT AT 3,3;"PASO MIS. CAN. MIS. CAN. BAR."
210 PRINT AT 4,3;"==== ==== ==== ==== ==== ===="
220 LET p=0
230 LET m=n
240 LET c=n
250 LET b=1
260 LET l=5
270 LET z(1,1)=m
280 LET z(1,2)=c
290 LET z(1,3)=b
300 PRINT AT 1,4;p
301 PRINT AT 1,9;m
302 PRINT AT 1,14;c
303 PRINT AT 1,19;n-m
```

```
304 PRINT AT 1,24;n-c
305 PRINT AT 1,29;b
310 LET l=l+1
312 IF l=22 THEN  LET l=5
315 LET p=p+1
320 GO SUB 3000
330 GO SUB 4000
340 GO SUB 5000
350 IF m<>0 OR c<>0 OR b<>2 THEN  GO TO 300
352 IF l>20 THEN  LET l=5
353 PRINT AT l,4;p
354 PRINT AT l,9;m
355 PRINT AT l,14;c
356 PRINT AT l,19;n-m
357 PRINT AT l,24;n-c
358 PRINT AT l,29;b
359 LET l=l+1
360 PRINT AT l+1,3;"Me felicito. Resuelto."
390 PAUSE 0
391 CLS
392 PRINT "Desea continuar ?"
393 INPUT y$
394 IF y$<>"NO" AND y$<>"no" AND y$<>"No" AND y$<>"n" AND y$<>"N" THEN  CLS : G
O TO 140
395 STOP
2000 CLS
2010 PLOT 35,120: DRAW -20,0: DRAW 0,-40: DRAW 20,0
2020 PLOT 45,120: DRAW 20,0: DRAW 0,-40: DRAW 0,20: DRAW -20,0: DRAW 0,20: DRAW
0,-40
2030 PLOT 75,80: DRAW 0,40: DRAW 20,-40: DRAW 0,40
2040 PLOT 105,120: DRAW 20,0: DRAW -10,0: DRAW 0,-40: DRAW -10,0: DRAW 20,0
2050 PLOT 135,120: DRAW 0,-40: DRAW 15,0: DRAW 5,4: DRAW 0,12: DRAW -5,4: DRAW 5
,4: DRAW 0,12: DRAW -5,4: DRAW -15,0: DRAW 0,-20: DRAW 15,0
2060 PLOT 165,120: DRAW 20,0: DRAW 0,-40: DRAW 0,20: DRAW -20,0: DRAW 0,20: DRAW
 0,-40
2070 PLOT 195,120: DRAW 0,-40: DRAW 20,0
2080 PLOT 15,75: DRAW 200,0
2090 PRINT AT 19,2;"Pulse <ENTER> para continuar"
2095 PRINT AT 1,2;"                "
2100 PRINT AT 2,2; Manuel MUNOZ GARCIA"
2110 PAUSE 0
2120 RETURN
3000 FOR i=1 TO 2
3010 LET s(i,1)=m+i*((b=2)-(b=1))
3020 LET s(i,2)=c
3030 LET s(i,3)=b+1
3040 IF s(i,3)=3 THEN  LET s(i,3)=1
3050 NEXT i
3060 FOR i=1 TO 2
3070 LET s(i+2,1)=m
3080 LET s(i+2,2)=c+i*((b=2)-(b=1))
3090 LET s(i+2,3)=b+1
3100 IF s(i+2,3)=3 THEN  LET s(i+2,3)=1
3110 NEXT i
3120 LET s(5,1)=m+1*((b=2)-(b=1))
3130 LET s(5,2)=c+1*((b=2)-(b=1))
3140 LET s(5,3)=b+1
3150 IF s(5,3)=3 THEN  LET s(5,3)=1
3160 FOR i=1 TO 5
3170 LET s(i,4)=0
3180 NEXT i
3200 RETURN
4000 LET k=0
4010 FOR i=1 TO 5
4020 IF s(i,1)<0 OR s(i,1)>n THEN  LET s(i,4)=1
4030 IF s(i,2)<0 OR s(i,2)>n THEN  LET s(i,4)=1
4040 FOR j=1 TO p
4050 IF s(i,1)=z(j,1) AND s(i,2)=z(j,2) AND s(i,3)=z(j,3) THEN  LET s(i,4)=2
4060 NEXT j
4070 IF s(i,2)>s(i,1) AND s(i,1)<>0 THEN  LET s(i,4)=3
4080 IF n-s(i,2)>n-s(i,1) AND n-s(i,1)<>0 THEN  LET s(i,4)=3
4090 IF s(i,4)=0 THEN  LET k=k+1
4100 NEXT i
4200 RETURN
5000 IF k=0 AND l>20 THEN  LET l=5
5010 IF k=0 THEN  PRINT AT l+1,2;"Lo siento. No existe solucion.": STOP
5020 LET j=0
5025 LET d=n+1
5030 LET x=2*n+1
5040 FOR i=1 TO 5
5050 IF s(i,4)=0 AND s(i,1)+s(i,2)<x THEN  LET x=s(i,1)+s(i,2): LET j=i: LET d=A
BS (s(i,1)-s(i,2))
5055 IF s(i,4)=0 AND s(i,1)+s(i,2)=x AND ABS (s(i,1)-s(i,2))<d THEN  LET x=s(i,1
)+s(i,2): LET j=i: LET d=ABS (s(i,1)-s(i,2))
5060 NEXT i
5070 LET m=s(j,1)
5080 LET c=s(j,2)
5090 LET b=s(j,3)
5100 LET z(p,1)=m
5110 LET z(p,2)=c
5120 LET z(p,3)=b
5130 RETURN
```

END

YIS-503

... o la simbiosis músico-informática

La tecnología japonesa nos tiene ya acostumbrados a constantes —cuando no sorprendentes— innovaciones en equipos de ésos que hacen más agradables nuestros ratos de ocio. en este caso, la novedad procede de la síntesis de dos mundos tan distantes tradicionalmente como la música y la informática.

Como cabía esperar, han sido los japoneses los primeros en plasmar, definitivamente, la unión de la música con la informática bajo el aspecto de un ordenador. El YIS-503 no es un equipo más con alguna que otra posibilidad sonora: ¡es un músico por vocación!

La filosofía de su diseño está adscrita al novedoso estándar MSX, adoptado por varias firmas niponas para sus ordenadores personales. El MSX establece una normalización básica tanto en el aspecto hardware como software, de tal forma que cualquier ordenador acogido a esta filosofía es compatible con los restantes miembros de la familia MSX.

La firma Yamaha es la creadora del invento, bautizado con las siglas YIS-503. Con un cerebro basado en el popular microprocesador de 8 bits Z80 A, incorpora, en su versión base, 32 Kbytes de memoria ROM y una capacidad de memoria RAM que se eleva a los 48 Kbytes. A través de posteriores ampliaciones, el equipo puede llegar a acoger un volumen de memoria RAM nada desdeñable: ¡1 Megabyte! distribuidos en 16 bloques de 64 Kbytes.

En cualquier caso, al conectar la versión libre del YIS-503, el usuario se encuentra con una zona de RAM libre de 28.815 bytes.





El volumen de memoria RAM que queda hasta completar los 48 K. los reserva el sistema para aplicaciones específicas; por ejemplo, un bloque de 16 Kbytes se destina a memoria de pantalla.

El teclado —de tipo QWERTY y semejante al de una máquina de escribir—, consta de 63 teclas, incluidas las de desplazamiento del cursor, más otras cinco de control localizadas en la parte superior izquierda del mueble. Como se observa en la fotografía, en el ángulo superior derecho de la caja aparece una ranura para el alojamiento de cartuchos con programas.

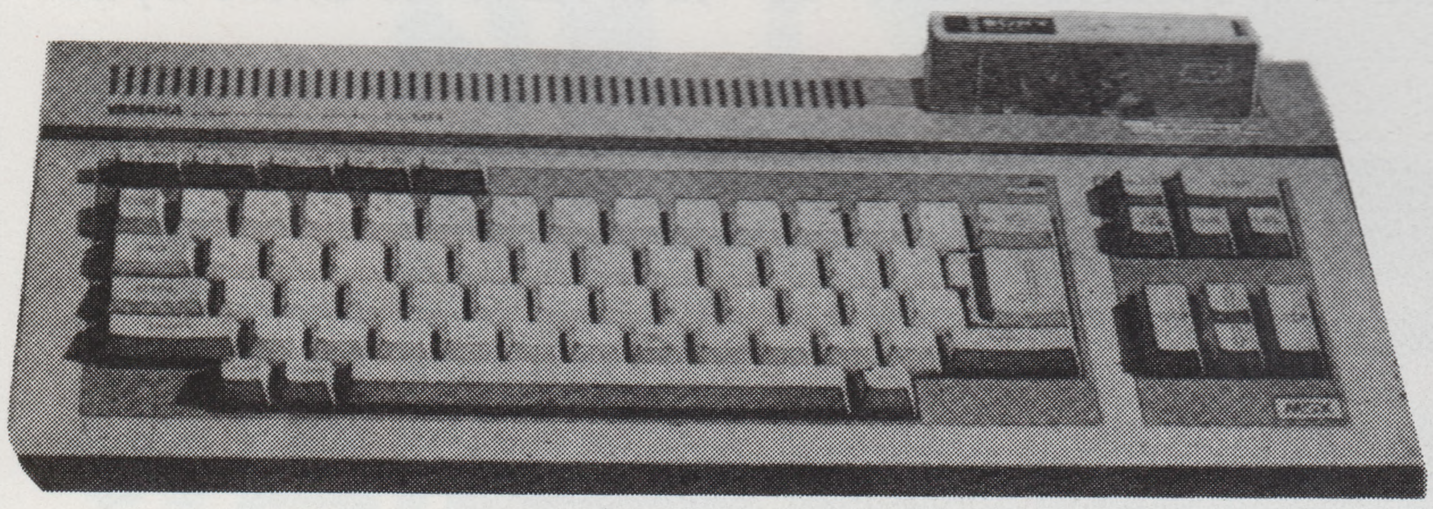
Los conectores para comunicación externa están dispuestos en el lateral izquierdo del mueble y a lo largo del panel posterior; al respecto, cabe mencionar la presencia de dos conectores destinados a la incorporación de sendos joysticks y varios accesos para la conexión de la impresora y los restantes periféricos.

La presentación en pantalla en modo texto adopta una distribución de 24 líneas de 32 ó 40 columnas, mientras que la resolución gráfica se eleva a 256 × 192 pixels. Como es habitual en los equipos adscritos a la norma MSX, el intérprete BASIC del YIS-503 dispone de los tradicionales comandos gráficos PLOT, LINE, CIRCLE, además del comando FILL destinado a rellenar de color determinadas zonas de la pantalla. Uno de los puntos fuertes del equipo es la posibilidad de operar simultáneamente con 32 «sprites» y hasta 16 colores, características más que reseñables y atractivas para dar versatilidad a los juegos.

Color, resolución, posibilidades de comunicación..., parece que no hay aspecto alguno en el que el YIS-503 permita objeciones. Aunque bien es cierto que la característica que destaca por encima de todas es el sonido. Dispone de ocho octavas y tres notas por acorde con control de envolvente. El genio que orquesta al conjunto sonoro es el sintetizador SFG-01 FM. Un auténtico virtuoso que, por si fuera poco, permite la elección de 48 voces distintas: guitarra eléctrica y clásica, clarinete, piano..., además de generar una ingente variedad de sonidos especiales (sirenas policiacas, sonido de la lluvia, etc.).

Todo este cúmulo de propiedades sonoras permiten al usuario del YIS componer música en modo monofónico o polifónico tocando a la vez ocho notas. La gama de «ritmos» de acompañamiento tampoco es escasa: jazz, rock... Y una opción adicional más que interesante: puede acoplarse al equipo un teclado de tipo piano como si se tratara de un periférico más. Por supuesto, la composición musical se va registrando de forma automática en la memoria RAM, para su posterior grabación en cinta de casete, disco... El inconveniente con el que se pueden topar los melómanos radica en el uso obligado del receptor de TV para deleitarse con sus melodías. No obstante, existen ya algunas firmas especializadas en equipos de sonido que están muy interesadas en crear ampliaciones con etapas de potencia y altavoces destinados a complementar esta peculiar inclinación musical del YIS-503.

El equipo ya está en la calle y sólo cabe esperar que el fabricante se decida a introducir el modelo en el mercado europeo y, por supuesto, nacional, para dar satisfacción a los usuarios..., y quebraderos de cabeza a la competencia.



El YIS-503: un novedoso representante japonés del estándar MSX que combina el virtuosismo musical con la eficacia y versatilidad de un potente microordenador.

Pág. 90, Tu Micro N.º 3/4

Fichero biblioteca

Un bibliotecario ordenado y diligente: el SPECTRAVIDEO SV-318

Tambie.. el verano es una época adecuada para poner orden en los cerros de libros que aguardan arrinconados en cajones y estanterías. Abriendo un hueco en los juegos, vamos a dedicar estas páginas a contarle cómo puede convertir al SPECTRAVIDEO en un ordenado y eficaz bibliotecario.

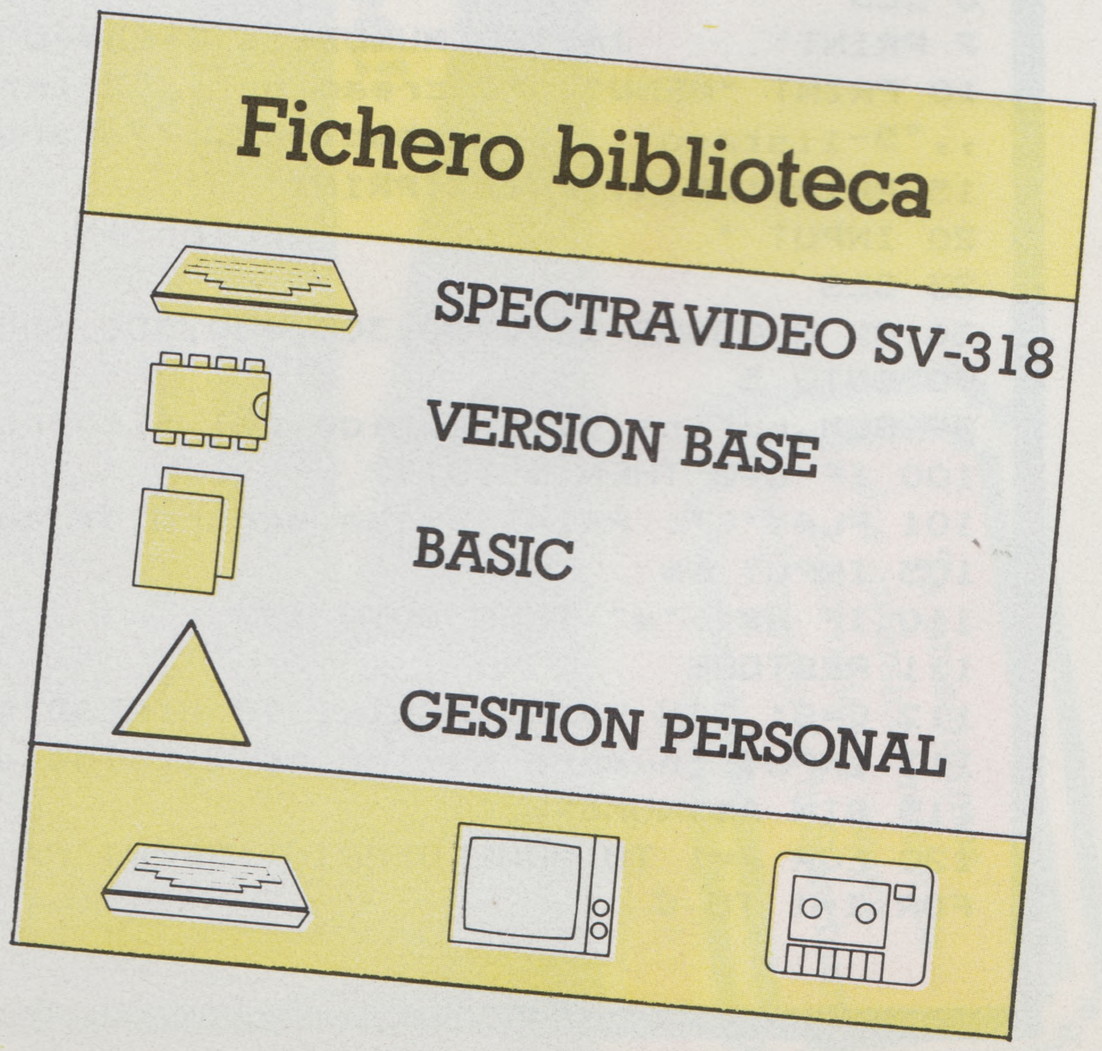
La primera tarea que vamos a exigirle es que se abandone a un sano ejercicio dactilar, introduciendo el programa cuyo listado reproducimos en este artículo. Una vez tecleado el programa, puede ya almacenarlo en casete para su posterior reutilización y ejecutarlo por medio del comando RUN.

Tal como observará sobre la pantalla, el programa está organizado entorno a un menú de opciones que dan paso a las 7 tareas que dan cuerpo a la aplicación.
La utilidad de cada una de las opciones es la que se define a continuación.

Opción 1: Creación del fichero

Esta opción se utiliza una sola vez, para crear el fichero que almacenará las fichas bibliográficas. en las sucesivas reutilizaciones del programa el fichero estará ya creado, de ahí que si se ejecutara de nuevo esta rutina el resultado sería, simplemente, un retorno al menú.

Sí, en efecto, es la primera vez que se utiliza el programa, el ordenador preguntará al usuario «si está seguro de su decisión de crear el fichero». Una respuesta negativa conducirá de nuevo al menú principal; por el contrario, si la respuesta es afirmativa, el programa pedirá al usuario «cuál es el número máximo de títulos que va a controlar el programa» (número máximo de títulos que va a controlar el programa» (número máximo de fichas, una por libro).



Tu Micro N.º 3/4, Pág. 91

En función de la memoria disponible, éste, puede definir un volumen de fichas más o menos amplio, por ejemplo, 100 fichas para la versión base del SV-318.

A continuación, a través de la pantalla, se solicitarán los datos según el número de campos disponibles. En el caso de que no se quiera llenar el fichero, el usuario puede abandonarlo introduciendo ZZZ.

Si se desea alterar el número de campos que constituyen cada ficha, basta con modificar el valor de C en la línea de instrucción 112, dando un nuevo valor a esta variable, e introducir las correspondientes cabeceras en la instrucción DATA de la línea 150. Una última solicitud del programa es el número de campo por el que va a realizarse la ordenación del fichero.

Opción 2: Inserción

A través de esta segunda rutina, asociada a la opción 2, se introducen los datos de las nuevas fichas que causan alta en el fichero de biblioteca. Una vez realizada la inserción, el programa pregunta al usuario si la introducción es correcta y termina efectuando una nueva ordenación del fichero.

Opción 3: Buśqueda

El programa empieza pidiendo el campo por el que debe realizarse la búsqueda, acto seguido, se efectúa la oportuna ordenación y se pide el título a localizar. El resultado del proceso es la presentación en pantalla de la ficha buscada o el mensaje de ficha inexistente. Para regresar al menú basta, sencillamente, con accionar la tecla ENTER. El proceso de búsqueda se realiza de forma secuencial por comparación.

ESTRUCTURA DEL PROGRAMA

1-90	Programa principal. Menú.
99-190	Rutina de creación.
199-290	Rutina de inserción.
299-300	Rutina de búsqueda.
399-490	Rutina de eliminación.
499-590	Rutina de impresión de todo el fichero.
599-690	Rutina de grabación.
699-790	Rutina de carga.
899-1160	Rutina de ordenado.

LISTADO



```
1 REM ENRIQUE JAVIER MURUBE SOTO
2 REM Programa principal
3 PLAY"AB"
5 CLS
7 PRINT"      JAVIER MURUBE's fichero"
10 PRINT "MENU",,"1-creacion",,"2-insercion",,"3-busqueda",,"4-eliminacion"
,,"5-listado",,"6-grabacion",,"7-carga"
15 PRINT:PRINT:PRINT:PRINT
20 INPUT "                    opcion";N
25 CLS
30 ON N GOSUB 100,200,300,400,500,600,700
90 GOTO 5
99 REM rutina de creacion del fichero
100 IF C=8 THEN GOTO197
101 PLAY"C": PRINT"Estas seguro de querer crear? s/n  "
105 INPUT N$
110 IF N$<>"s" THEN GOTO 197
111 RESTORE
112 C=8: DIM V$(C):FORI=1 TO C:READ V$(I):NEXT I
113 INPUT "numero maximo de titulos a guardar ";NUMERO
115 DIM A$(NUMERO,C)
120 FOR J=1 TO NUMERO:PRINT:PRINT"PARA TERMINAR: zzz ":PRINT: PRINT:
FOR I=1 TO C
```

TABLA DE VARIABLES

A$	Almacena la información del fichero.
C	Número de campos.
I, J, K	Variables auxiliares para bucles.
INDICE	Fichas existentes.
N	Variable auxiliar.
NUMERO	Número máximo de fichas.
N$, T$	Variables auxiliares.
RN	Define posición en las operaciones de búsqueda.
S	Almacena el campo por el que se busca y ordena.
V$	Encabezamientos de los campos.

Opción 4: Eliminación

La rutina que ejecuta este proceso empieza acometiendo una búsqueda del título que debe causar baja en el fichero, con el fin de comprobar su presencia o no. Si el título está en el fichero, el ordenador pide al usuario conformidad antes de proceder al borrado.

Opción 5: Listado del fichero

Esta opción presenta en pantalla un listado completo del contenido del fichero de biblioteca, según la última ordenación realizada. Para detener el «scroll» (desplazamiento) de la pantalla debe utilizarse la tecla STOP; para proseguir con el listado bastará con una segunda acción sobre la tecla STOP.

Opción6: Grabación

Graba el fichero con el que se está trabajando en ese instante. Si no se estuviera operando sobre un fichero, se retornará la menú.

Opción7: Carga de un fichero almacenado en casete

A diferencia con las restantes opciones, ésta resulta operativa aun en el caso de que no esté creado el fichero bibliográfico. Al ejecutarla, la rutina asociada crea en memoria un fichero de trabajo que incluye el fichero recuperado del casete más un total de 10 fichas adicionales, libres, para posibles inserciones de nuevos títulos. Esta característica resulta de gran interés puesto que de esta forma se van creando espacios libres a medida que resultan necesarios. Por supuesto, el usuario no debe recurrir a la opción 1 de creación de fichero si lo que va a realizar es una recuperación del fichero almacenado en casete.

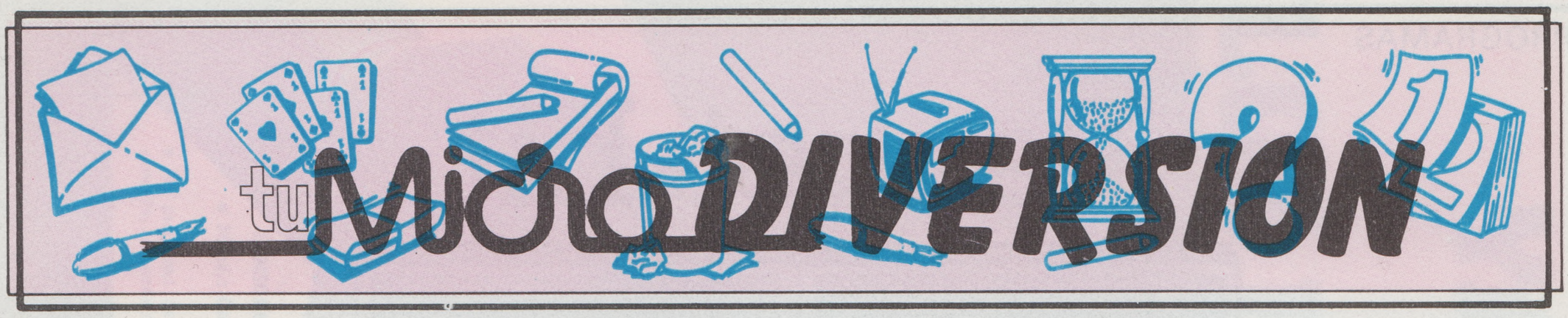
A título ilustrativo, el programa viene preparado para almacenar las fichas de los libros que publicamos en la correspondiente página de TU MICRO.

Micro

```
125 PRINT V$(I):INPUT A$(J,I)
130 IF A$(J,I)="zzz"THEN  GOTO 195
135 NEXT I:CLS:NEXT J
145 INDICE=NUMERO
150 DATA titulo,autor,editorial,idioma,edicion,paginas,precio,notas
155 GOTO 196
195 INDICE=J-1
196 GOSUB 900
197 RETURN
199 REM rutina de insercion de 1 elemento
200 PLAY"D": IF INDICE=NUMERO THEN GOTO290
205 PRINT "INSERCION"
210 FOR I=1 TO C
215 PRINT V$(I):INPUT A$(INDICE+1,I)
220 NEXT I
225 PRINT:INPUT"CORRECTO? s/n ";N$
230 IF N$<>"s"THEN GOTO 291
235 INDICE=INDICE+1
240 GOSUB 900
250 GOTO 291
290 PRINT"FICHERO COMPLETO":INPUT J
291 RETURN
```

```
299 REM rutina de busqueda de un elemento
300 PLAY"E": IF INDICE=NUMERO THEN GOTO340
301 GOSUB 900
302 PRINT "Introduce ";V$(S);" a buscar"
303 INPUT N$
305 FOR I=1 TO INDICE
310 IF A$(I,S)=N$ THEN GOTO 375
315 IF A$(I,S)>N$ THEN GOTO 330
320 NEXT I
330 CLS:PRINT"NO ESTA ";N$:INPUT J
332 RN=0
340 RETURN
375 RN=I
380 FOR I=1 TO C :PRINT V$(I);": ";A$(RN,I):NEXT I:INPUT J
385  GOTO 340
399 REM rutina de eliminacion de 1 elemento
400 PLAY"F": IF INDICE=NUMERO THEN GOTO490
405 GOSUB 300
410 IF RN=0 THEN GOTO 490
415 INPUT"Seguro de querer borrarlo? s/n  ";T$:IF T$<>"s" THEN GOTO 490
420 A$(RN,S)="zzzzzz"
425 GOSUB 1000
430 INDICE=INDICE-1
490 RETURN
499 REM rutina de impresion de todo el fichero
500 PLAY"G": FOR I=1 TO INDICE:FOR J=1 TO C
510 PRINT V$(J);": ";A$(I,J)
520 NEXT J:FOR K=1 TO 100:NEXT K:PRINT:PRINT:NEXT I
525 INPUT J
530 RETURN
599 REM RUTINA DE GRABACION
600 PLAY"D": IF INDICE=NUMERO THEN GOTO690
601 OPEN"cas:t"FOR OUTPUT AS 1
602 PRINT#1,INDICE,C
605 FOR I=1 TO INDICE:FOR J=1 TO C
610 PRINT#1,A$(I,J)
615 NEXT J:NEXT I
620 FOR I=1 TO C:PRINT#1,V$(I):NEXT I
625 CLOSE 1
690 RETURN
699 REM rutina de carga
700 PRINT"PONGA EN MARCHA EL CASETE PARA CARGAR   EL FICHERO":
OPEN"cas:t"FOR INPUT AS 1
705 INPUT#1,INDICE,C:DIM A$(INDICE+10,C):DIM V$(C)
710 FOR I=1 TO INDICE:FOR J=1 TO C
715 INPUT#1,A$(I,J)
720 NEXT J:NEXT I
725 FOR I=1 TO C:INPUT#1, V$(I):NEXT I
730 CLOSE 1
790 RETURN
898 REM
899 REM rutina de ordenacion (no visible)
900 CLS:PRINT"Por que numero de campo desea la busqueda/ordenacion":
FOR K=1 TO C:PRINT K;"-";V$(K):NEXT K:INPUT"numero";S
1000 PLAY"ABC"
1005 J=0
1010 FOR I=1 TO INDICE-1
1020 IF A$(I,S)>A$(I+1,S)THEN GOSUB 1100
1030 NEXT I
1040 IF J<>OTHEN GOTO 1005
1050 RETURN
1100 J=1
1110 FOR K=1 TO C
1120 N$= A$(I,K)
1130 A$(I,K)=A$(I+1,K)
1140 A$(I+1,K)=N$
1150 NEXT K
1160 RETURN
```

END



una de lenguajes

Intente poner orden en este revoltijo de lenguajes, haciendo corresponder el nombre de cada lenguaje con su descripción.

1.	**BASIC**	a) Lenguaje de alto nivel con una sintaxis y con un conjunto de caracteres orientados a aplicaciones matemáticas, sobre todo de tipo matricial. Fue desarrollado por Ivenso de IBM.
2.	**PASCAL**	b) Cualquier lenguaje simbólico que se utiliza para la confección de programas fuente muy próximos al lenguaje máquina y cuya traducción a programa objeto es muy directa.
3.	**ENSAMBLADOR**	c) Lenguaje de programación interactivo de alto nivel, conocido por la utilización de una «tortuga» en el trazado de gráficos.
4.	**APL**	d) Lenguaje de alto nivel estructurado en bloques. Incorpora las estructuras de control de programación (secuencia, selección y repetición) y las estructuras de datos (matrices, registros, ficheros y componentes definidos por el usuario). Su nombre corresponde al de un matemático francés que diseñó una máquina sumadora en el siglo XVII.
5.	**FORTRAN**	e) Lenguaje de alto nivel concebido para desarrollar programas en modo conversacional y programación en línea. Inicialmente estaba destinado a principiantes no especializados en Informática. Goza de una gran aceptación en el campo de los ordenadores personales.
6.	**LOGO**	f) Lenguaje de alto nivel que se concibió originalmente como un instrumento a utilizar por los matemáticos para representar algoritmos. Permite una expresión concisa de procesos aritméticos y lógicos.
7.	**ALGOL**	g) Lenguaje multitarea interactivo de alto nivel que se creó a mediados de la década de los años 60 y que es esencialmente, un lenguaje para control de procesos. Sus operandos se mantienen en una pila y los programas adoptan la forma de cadenas en notación polaca inversa. Tiene gran aceptación entre los usuarios de microordenadores.
8.	**COBOL**	h) Lenguaje de programación muy utilizado para el cálculo científico; ideal para la expresión de fórmulas.
9.	**FORTH**	i) Lenguaje de alto nivel para problemas de procesos en tiempo real. Se creó inicialmente para control de aviones militares. Se le dió su nombre en honor de quien se consideró fue el primer programador.
10.	**ADA**	j) Lenguaje de alto nivel orientado hacia los problemas de procesos de datos, en actividades comerciales.

sopa informática

Trate de localizar en nuestro «sopigrama» el mayor número de lenguajes informáticos.

Calificación:
- 5 o menos — ¡Hay que leer más TU MICRO!
- De 6 a 8 — Buena culturilla.
- De 9 a 10 — ¡Ud. es todo un políglota!
- Más de 10 — Pueden proponerle para «Académico de los lenguajes».

Nota: Salvo «espontáneos», son 13 los lenguajes ocultos.

N	T	R	O	F	R	A	T	B
A	O	I	C	D	L	E	A	R
P	G	W	O	G	A	S	D	L
L	O	B	O	C	I	I	A	P
D	L	L	A	C	S	A	P	C
P	A	N	A	R	T	R	O	F

Soluciones a los pasatiempos de TU MICRO número 2.

revoltijo de siglas
1-D, 2-H, 3-G, 4-I, 5-F, 6-3. 7-M, 8-C, 9-J, 10-K, 11-1, 12-B, 13-N, 14-O, 15-A.

el juego de los números
A-4, B-5, C-7, D-3, E-1, F-2, G-6, H-10, 1-8, J-9.

jeroglífico
En tiempo compartido
(EN/T/I/E/M/PO/CO/M/PARTIDO).

Basket-bit



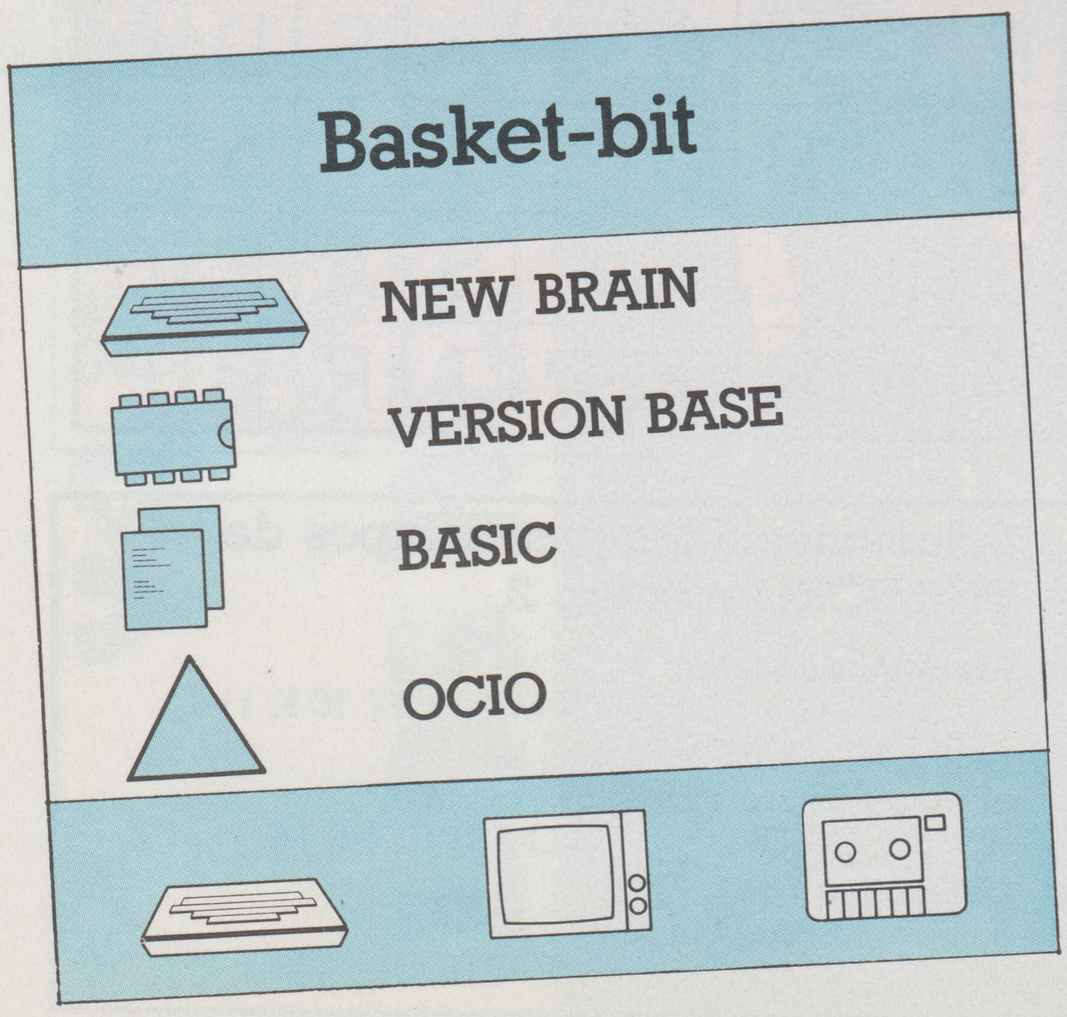
¡El programa olímpico de Tu MICRO!

¡Ya están aquí las olimpiadas! y, por supuesto, no podíamos olvidarnos de ello en TU MICRO. Barajando ideas y proyectos, nos hemos decidido por el Basket-bit: un programa con el que podrá seguir al detalle las evoluciones de los emocionantes partidos de baloncesto.

Control de las alineaciones y cambios durante el partido, evolución del marcador, personales... todos los datos relativos al partido serán gestionados por el NEW BRAIN, instruido por el programa que presentamos en este artículo. La única exigencia para el usuario es la rapidez en el manejo del teclado, para que no llegue a escaparse ni un sólo detalle del olímpico acontecimiento.

En empleo del programa es de una sencillez proverbial. Todas las instrucciones relativas a su empleo las encontrará el usuario en la propia pantalla del televisor conectado al NEW BRAIN.

Una vez introducido el programa a través del teclado y almacenado en casete para su posterior reutilización, puede ya lanzar la ejecución auxiliado por el comando RUN... y ¡que gane el equipo español!



Estructura del programa

El funcionamiento del programa está organizado en torno a un grupo de tablas, de una sola dimensión, que son inicializadas en la línea 10.

Las instrucciones comprendidas entre las líneas 50 y 190 gestionan la construcción de las mencionadas tablas; para ello se pide al usuario el nombre de dos equipos en liza y la relación de jugadores que intervienen, indicando si estos forman parte de la alineación inicial.

A continuación aparece en pantalla un menú que ofrece tres opciones seleccionables por el usuario: 1-canasta, 2-personales y 3-cambios de alineación. La gestión de las tres opciones disponibles corre a cargo de las subrutinas que empiezan en las líneas 1000, 2000 y 3000, respectivamente. La bifurcación a la subrutina correspondiente la realiza la instrucción ON M GOSUB 1000, 2000, 3000 localizada en la línea 270.

Al hacer uso de las variables NA$ y NB$, hay que respetar la limitación impuesta por su longitud, lo que significa que no pueden introducirse nombres de jugadores que excedan de diez caracteres.

En las opciones 2 y 3, es posible que el cursor regrese al principio de la línea de opción después de que el usuario introduzca los datos que desee. Esta circunstancia ocurrirá cada vez que se cometa un error de entrada de datos; por ejemplo, cuando al definir un cambio en el cinco titular, el usuario introduzca como número del jugador que sale del terreno de juego el correspondiente a un jugador que ya se encuentra en el banquillo o que se encuentra expulsado por acumulación de personales.

Un último dato a reseñar es que para comprimir la extensión del listado se han sustituido los comandos PRINT por su símbolo simplificado?

¡Felices olimpiadas y mejor baloncesto!... y, por supuesto, ¡que gane España!

LISTADO

```
1 REM *-TABLA DE BALONCESTO-*
2 REM *-I.MORAGREGA V.-*
3 PUT 31:PUT 23,67
4 PUT 22,12,9,"":?"******************"
5 PUT 22,9,11,"":?"*** EURO TUMICRO-84 ***"
6 PUT 22,12,13,"":?"******************"
7 FOR Z=0 TO 1300:NEXT Z:PUT 31:PUT 23,66
10 DIM NA$(14),NB$(14),TA$(14),TB$(14),PA(14),PB(14)
20 Z$="                    "
30 PUT 22,11,2,"":?"PARTIDO DE BALONCESTO"
40 ? TAB(11);"----------------------"
50 PUT 22,2,7,"":INPUT("NOMBRE DEL EQUIPO (A):") A$
60 PUT 22,2,14,"":INPUT("NOMBRE DEL EQUIPO (B):") B$
80 PUT 31:? TAB(11);"PLANTILLA DEL ";A$ :A=0
90 FOR N=5 TO 14
100 PUT 22,4,7,"":? "NUMERO DEL JUGADOR ";N
110 PUT 22,4,9,"":INPUT("NOMBRE DEL JUGADOR:") NA$(N)
115 IF LEN(NA$(N))>10 THEN X=1:GOTO 300
120 PUT 22,1,13,"":INPUT("SI ES TITULAR, TECLEE (*) SINO BLANCO :") TA$(N)
123 IF TA$(N)="*" THEN A=A+1
125 IF TA$(N)="*" OR TA$(N)="" THEN 130
129 GOTO 120
130 PUT 22,23,9,"":? Z$:PUT 22,40,13,"":? " ":NEXT N
135 IF A<>5 THEN X=1:GOTO 500
140 PUT 31:? TAB(11);"PLANTILLA DEL ";B$:B=0
150 FOR N=5 TO 14
160 PUT 22,4,7,"":? "NUMERO DEL JUGADOR ";N
170 PUT 22,4,9,"":INPUT("NOMBRE DEL JUGADOR:") NB$(N)
175 IF LEN(NB$(N))>10 THEN X=0:GOTO 300
180 PUT 22,1,13,"":INPUT("SI ES TITULAR, TECLEE (*) SINO BLANCO :") TB$(N)
183 IF TB$(N)="*" THEN B=B+1
185 IF TB$(N)="*" OR TB$(N)="" THEN 190
187 GOTO 180
190 PUT 22,23,9,"":? Z$:PUT 22,40,13,"":? " ":NEXT N
195 IF B<>5 THEN X=0:GOTO 500
200 PUT 31
210 ? TAB(5);A$;TAB(25);B$
220 ?:? TAB(7);MA;TAB(30);MB
230 ?:?: FOR C=5 TO 14
240 ? TAB(1);C;TAB(5);TA$(C);TAB(7);NA$(C);TAB(18);PA(C);TAB(21);C;TAB(25);TB$(C
);TAB(27);NB$(C);TAB(38);PB(C)
250 NEXT C
260 PUT 22,0,23,"":INPUT("MENU (1 CANASTA, 2 PERSONAL, 3 CAMBIO):") M
265 IF M<1 OR M>3 THEN 260
270 ON M GOSUB 1000,2000,3000
280 GOTO 200
300 PUT 22,1,13,"":?"NOMBRE SUPERA 10 CARACTERES.DISMINUYALO"
310 PUT 22,23,9,"":? Z$
320 IF X=1 THEN 110
```

```
330 GOTO 170
500 PUT 31:PUT 22,2,8,"":?"NO HAY EL NUMERO EXACTO DE TITULARES."
510 ? TAB(5);"REPITA DE NUEVO LA OPERACION."
520 ? TAB(8);"RECUERDE QUE TIENEN QUE SER 5."
530 PUT 22,3,15,"":INPUT("PULSE NEW LINE PARA REPETIR.") C$
540 IF X=1 THEN 80
550 GOTO 140
1000 PUT 31:?:? TAB(9);"CANASTA-CANASTA-CANASTA"
1010  PUT 22,3,9,"":INPUT("EQUIPO A o B:") EQ$
1020 PUT 22,3,15,"":INPUT("PUNTOS CONSEGUIDOS:") P
1030 IF EQ$="A" THEN MA=MA+P
1040 IF EQ$="B" THEN MB=MB+P
1050 IF EQ$<>"A" AND EQ$<>"B" THEN 1000
1060 RETURN
2000 PUT 31:?:? TAB(8);"PERSONAL-PERSONAL-PERSONAL"
2010 PUT 22,3,9,"":INPUT("EQUIPO A o B:") EQ$
2020 PUT 22,3,15,"":INPUT("NUMERO DEL JUGADOR:") N
2030 IF EQ$="A" AND TA$(N)="*" THEN 2100
2040 IF EQ$="B" AND TB$(N)="*" THEN 2200
2050 GOTO 2000
2100 PA(N)=PA(N)+1
2110 IF PA(N)<>5 THEN 2140
2120 IF TA$(N)="*" THEN GOSUB 3000
2130 TA$(N)="E"
2140 GOTO 200
2200 PB(N)=PB(N)+1
2210 IF PB(N)<>5 THEN 2240
2220 IF TB$(N)="*" THEN GOSUB 3000
2230 TB$(N)="E"
2240 GOTO 200
3000 PUT 31:?:? TAB(11);"CAMBIO-CAMBIO-CAMBIO"
3010 PUT 22,3,7,"":INPUT("EQUIPO A o B:") EQ$
3020 PUT 22,3,11,"":INPUT("NUMERO DEL JUGADOR QUE SALE:") N1
3030 PUT 22,3,15,"":INPUT("NUMERO DEL JUGADOR QUE ENTRA:") N2
3040 IF EQ$="A" THEN 3100
3050 IF EQ$="B" THEN 3200
3060 GOTO 3000
3100 IF TA$(N1)="E" OR TA$(N1)=""  THEN 3020
3110 IF TA$(N2)="E" OR TA$(N2)="*"  THEN 3030
3120 TA$(N1)="":TA$(N2)="*"
3130 RETURN
3200 IF TB$(N1)="E" OR TB$(N1)=""  THEN 3020
3210 IF TB$(N2)="E" OR TB$(N2)="*"  THEN 3030
3220 TB$(N1)="":TB$(N2)="*"
3230 RETURN
```

END

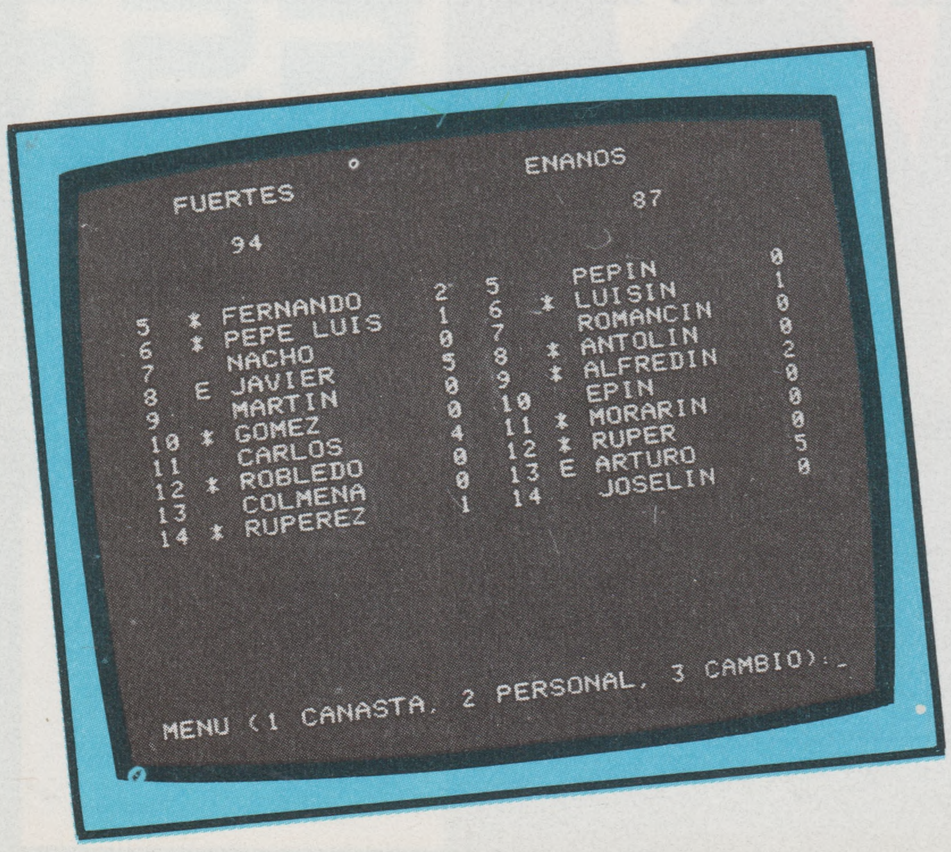


TABLA DE VARIABLES

A	Contador de jugadores equipo A.
B	Contador de jugadores equipo B.
A$	Nombre del equipo A.
B$	Nombre del equipo B.
NA$	Nombre de jugador equipo A.
NB$	Nombre de jugador equipo B.
TA$	Titular del equipo A.
TB$	Titular del equipo B.
MA	Marcador equipo A.
MB	Marcador equipo B.
PA	Personales equipo A.
PB	Personales equipo B.
Z$	Cadena de caracteres blancos (15).
Z	Variable auxiliar para bucle cerrado.

tuMicro

anuncios breves

TEXTO DEL ANUNCIO

Escriba de forma clara y en mayúsculas una sola letra por casilla. No olvide indicar su nombre y dirección o teléfono en la zona de datos personales. Evite abreviaturas. No olvide poner en el sobre las siglas ABS.

DATOS PERSONALES

Nombre: .....

Dirección: .....

.....

El servicio de ANUNCIOS BREVES se rige por las normas que se detallan en la página de «Servicios TU MICRO».

Analógico: Se refiere a la representación de un fenómeno físico mediante una magnitud eléctrica (tensión, corriente...), que varía de forma continua. Un dispositivo analógico es aquel que trabaja con las representaciones analógicas de las funciones a estudiar. Se asimila al concepto de variación continua en oposición a la variación de tipo digital con paso de un estado a otro de forma discontinua.

APL: Abreviatura de *«A Programming Language»*.
Se trata de un lenguaje de programación de alto nivel concebido por IBM en 1962 con fines netamente científicos. Tiene las características de ser sencillo, conciso y adaptable a las expresiones matemáticas (lo que no es de extrañar puesto que su creador, Iverson, lo concibió inicialmente como una notación matemática).
A veces resulta «impenetrable» para quien no lo domina, lo que, en ciertas ocasiones, resulta ventajoso.

tuMicro

suscripciones

Apellidos:
Nombre:
Domicilio:
Localidad: **D.P.:**
Provincia: **Teléfono:**
País: **Profesión:**

Deseo suscribirme a TU MICRO por el período de un año a partir del número ....., al precio de ☐ 2.500 ptas. (correo ordinario) o ☐ 2.800 ptas. (correo certificado).

El importe lo haré efectivo mediante:

Firma

☐ Talón bancario a nombre de INGELEK.
☐ Reembolso (por el importe de la suscripción más gastos de envío) al recibir el primer ejemplar.
☐ Giro postal n.º ..... Fecha .....
Localidad .....

(No olvide poner en el sobre las siglas STM.)

(ANSI). Se utiliza para representar las informaciones en la mayor parte de las máquinas informáticas.

Asíncrono: Método de transmisión de datos en el que los caracteres del mensaje se transmiten uno a uno entre bits de sincronización de reloj y en donde los intervalos entre caracteres transmitidos pueden ser de longitud desigual. En el caso de circuitos lógicos electrónicos se refiere a un sistema en el que las operaciones lógicas no se realizan bajo el control de una señal de reloj y se producen retardos significativos entre señales.

Assembler: Ver Ensamblador.

Autocode: Autocódigo, código automático.
Un nombre genérico utilizado para un grupo de lenguajes que está comprendido entre ensambladores y lenguajes de alto nivel.

Argumento: Valor de una variable independiente que se utiliza en el cálculo de una función.
También se refiere a un dato conocido incluido en una tabla y que sirve para su referencia o indicativo.
En programación suele ser sinónimo de parámetro formal de una función o de un procedimiento.

Arquitectura: Conjunto de características estructurales y de interconexiones de los componentes principales de un sistema. En el caso concreto de un microprocesador se refiere a los registros (número y funciones), medios de direccionamiento, disposición de los buses y elementos de secuenciamiento.

ASCII: Abreviatura de *American Standard Code for Information Interchange.*
Un código de caracteres (de siete bits más el dígito de paridad) establecido por el American Standards Institute.

(Las referencias cruzadas se indican con un asterisco.)

(Las referencias cruzadas se indican con un asterisco.)

Mediante el empleo de instrucciones nemónicas tiene como misión facilitar la programación de un ordenador. Se puede considerar como precursor de los modernos lenguajes de alto nivel.
Se le conoce también como «lenguaje clásico».

Autómata: Dispositivo animado y controlado por mecanismos internos que trata de imitar los movimientos de un ser viviente. Se trata de una tentativa tangible de mejorar la comunicación hombre-máquina.

Automática: Técnica de la automatización o de los sistemas automáticos.

AZERTY: Tipo de teclado. Se utiliza principalmente en los países europeos y se deriva de las letras que figuran en la fila superior.

anuncios breves

tuMicro

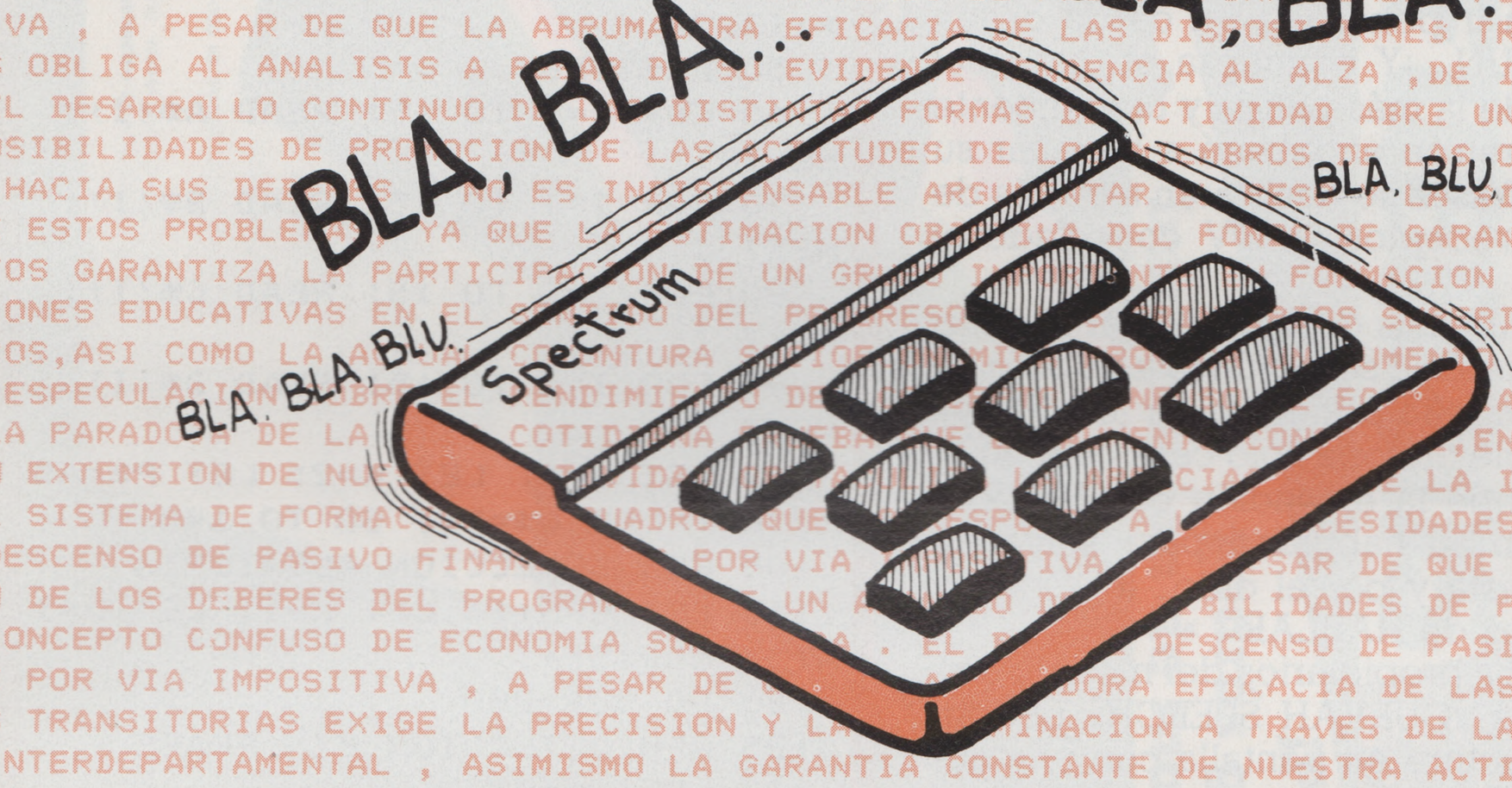
Recorte esta ficha y envíela dentro de un sobre a:
Ediciones INGELEK, S. A.
Apto. de Correos 61.294
28080 MADRID

suscripciones

tuMicro

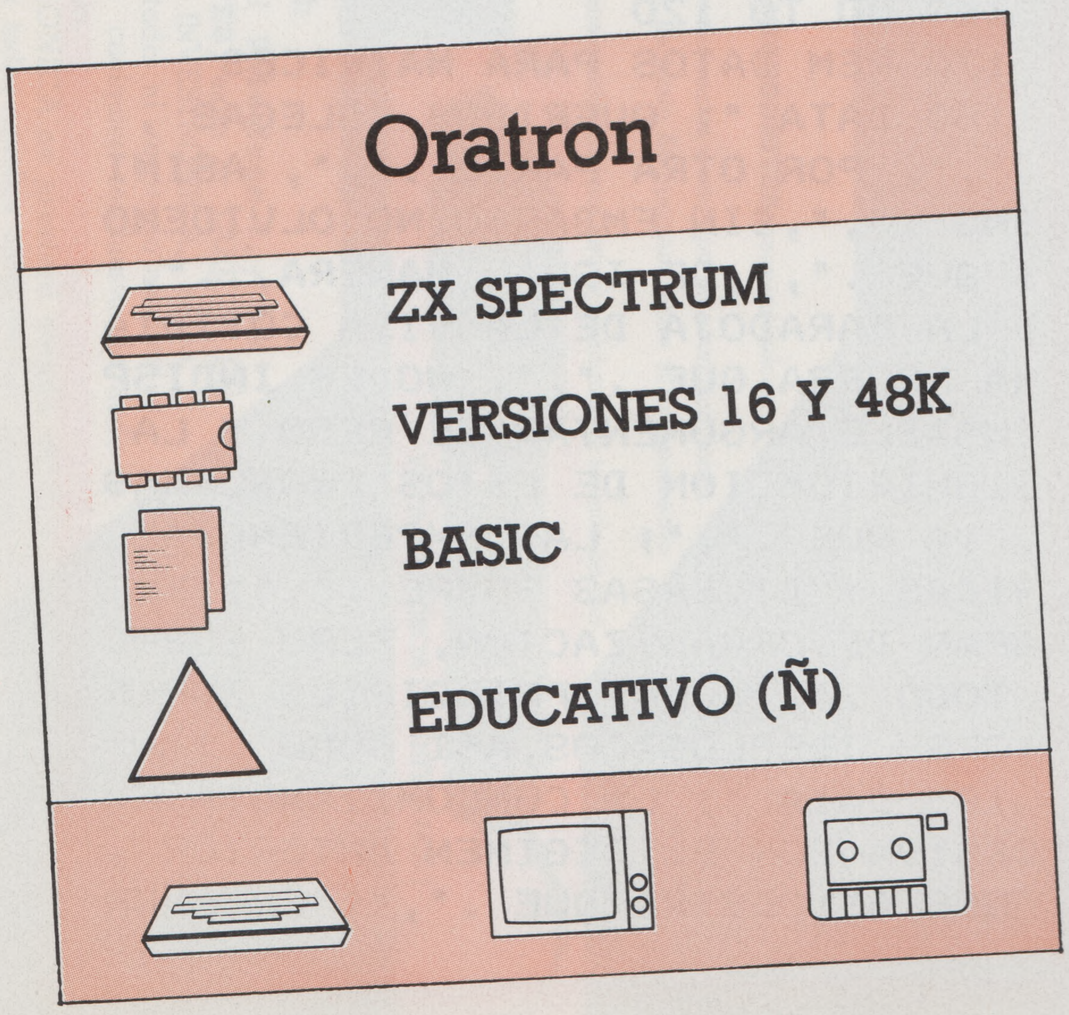
Recorte esta ficha y envíela dentro de un sobre a:
Ediciones INGELEK, S. A.
Apto. de Correos 61.294
28080 MADRID

Oratrón



¡Convierta al SPECTRUM en un orador inquebrantable!

Si Ud. cree que lo suyo es la política, pero no sabe a ciencia cierta de que va, y quiere entrar en ellas con buen pie, aquí está su gran oportunidad: el Oratrón. Un método definitivo para alcanzar puestos de elevada responsabilidad y no menos elevada remuneración.



Por si no lo ha adivinado, se trata de un programa generador de discursos que convierte al SPECTRUM en un brillante e inquebrantable orador, maestro en las artes de la dialéctica insustancial aunque efectista.

Su interés práctico para la confección de comprometidos discursos queda fuera de cualquier duda; aunque tampoco son despreciables sus aptitudes como instigador al ejercicio mental del incauto oyente que pretenda extractar las sustanciosas conclusiones de su parlamento.

El Oratrón frente al atril

El podio desde el que el Oratrón imparte su densa sabiduría dialéctica es el popular ZX-SPECTRUM. La primera tarea que debe realizar el usuario es, en consecuencia, cargar el programa cuyo listado se adjunta a través del

teclado del equipo. Una vez introducido, resulta muy conveniente grabarlo en una cinta; para ello basta con recurrir a la instrucción siguiente:

SAVE "..." LINE 1

Por supuesto, en lugar de los puntos suspensivos encerrados entre comillas debe introducir el nombre que elija para identificar al programa.

Con el programa grabado y puesto el casete a buen recaudo para evitar una nueva sesión de carga «digital» (en este caso «digital» procede de la raíz castellana «dedo»... ¡ya saben!), puede ya ir pensando en dar la orden de ejecución RUN... ¡Y comienza el discurso! Haciendo gala de la más pulida y convincente verborrea, el Oratrón empezará a hilvanar su discurso cuyas frases se deslizarán a través de la pantalla del receptor de TV conectado al SPECTRUM.

Tal vez, la virtud más destacable del Oratrón sea su constancia. Esta se manifiesta no sólo en un enconado empeño por eternizar los discursos (su inagotable oratoria sólo cesa cuando el usuario interrumpe la ejecución con BREAK), sino también en su habilidad para atacar al oyente por la vista y por el oído... ¡El Oratrón también es capaz de HABLAR!

Los afortunados propietarios del módulo sintetizador de voz «CURRAH SPEECH» podrán acompañar la lectura en pantalla con una audición directa de los discursos generados por el Oratrón.

Sin lugar a dudas, las posibilidades del Oratrón son notables; he aquí una breve síntesis:

ESTRUCTURA DEL PROGRAMA

10-100	Inicialización.
120-160	Selección y visualización.
180-210	Datos.
290-370	Selección tipo datos.
400-500	Rutina impresión.

TABLA DE VARIABLES

G	Selección tipo frase.
I	Usada para bucle FOR/NEXT.
J	Número de frase en la matriz.
K	Número de carácter en la frase.
Q	Contador de carácteres.
A$	Matriz de frases.
B$	Matriz de frases.
C$	Matriz de frases.
D$	Matriz de frases.
P$	String de actualización.
S$	String salida voz.
W$	String auxiliar voz.
Y$	String de acumulación.
Z$	String de visualización.

LISTADO

```
  10 REM by LOYSOFT'84 & THE
                 CRAZY REDACTOR
  20 REM INICIALIZACION
  30 BORDER 1: PAPER 1: INK 7: C
LS
  35 LET Q=0: LET Y$="": LET Z$=
"
  "
  40 LET G=1: DIM A$(15,90): DIM
 B$(15,90): DIM C$(15,90): DIM D
$(15,90)
  50 RESTORE
  60 FOR I=1 TO 15: READ A$(I):
NEXT I
  70 FOR I=1 TO 15: READ B$(I):
NEXT I
  80 FOR I=1 TO 15: READ C$(I):
NEXT I
  90 FOR I=1 TO 15: READ D$(I):
NEXT I
 100 LET J=1: LET K=2: LET P$=A$
(J,K): LET W$=P$: GO TO 145
 105 REM SELECCION ALEATORIA
 110 LET G=G+1: LET K=1: LET J=I
NT (RND*15)+1
 120 GO SUB 290
 125 REM DETECCION FIN FRASE
 130 LET W$=P$
 135 IF P$=CHR$ 46 THEN  GO TO 1
10
 140 IF P$=CHR$ 59 THEN  LET P$=
CHR$ 46
 145 LET Q=Q+1: LET Z$=Z$+P$: LE
T Z$=Z$(2 TO )
 150 PRINT  INK 7;AT 12,0;Z$: BE
EP .05,INT (RND*80)-39: LET K=K+
1: IF K>89 THEN  GO TO 110
 151 REM  PRINT  INK 7;AT 12,0;Z
$: PAUSE 1: LET s$=W$: PAUSE 1:
_ET K=K+1: IF K>89 THEN  GO TO 1
10
 155 IF Q=32 THEN  GO SUB 400
 160 GO TO 120
 170 REM DATOS PARA MATRICES
 180 DATA "; QUERIDOS COLEGAS ,
.","," POR OTRA PARTE .","," ASIMI
SMO .","SIN EMBARGO,NO OLVIDEMO
S QUE .","DE IGUAL MANERA, .","
; LA PARADOJA DE LA VIDA COTIDIA
NA PRUEBA QUE .","," NO ES INDISP
ENSABLE ARGUMENTAR EL PESO Y LA
SIGNIFICACION DE ESTOS PROBLEMAS
, YA QUE .","; LAS EXPERIENCIAS
RICAS Y DIVERSAS SOBRE .","; EL
AFAN DE ORGANIZACION, PERO SOBRE
 TODO .","; LOS PRINCIPIOS SUPER
IORES IDEOLOGICOS,ASI COMO ."
 185 DATA "; LA CONCOMITANCIA EX
CESIVA CON EL REGIMEN ANTERIOR ,
 NOS DEMUESTRA QUE .","; EL ADOR
```

```
MECIMIENTO DE LAS CONCIENCIAS Y
.","; LA DESCALIFICACION PERMANE
NTE DE LA OPOSICION , ASI COMO .
","; EL BRUTAL DESCENSO DE PASIV
O FINANCIABLE POR VIA IMPOSITIVA
 , A PESAR DE QUE .","PUESTO QUE
 , Y LO DIGO SIN LA MAS MINIMA A
CRITUD , ."
 190 DATA "LA REALIZACION DE LOS
 DEBERES DEL PROGRAMA .","LA COM
PLEJIDAD DE LOS ESTUDIOS DE LOS
DIRIGENTES .","EL AUMENTO CONSTA
NTE,EN CANTIDAD Y EN EXTENSION D
E NUESTRA ACTIVIDAD .","LA ESTRU
CTURA ACTUAL DE LA ORGANIZACION
.","EL NUEVO MODELO DE LA ACTIVI
DAD DE LA ORGANIZACION .","EL DE
SARROLLO CONTINUO DE LAS DISTINT
AS FORMAS DE ACTIVIDAD .","LA GA
RANTIA CONSTANTE DE NUESTRA ACTI
VIDAD DE INFORMACION Y PROPAGAND
A .","EL REFORZAMIENTO Y DESARRO
LLO DE LAS ESTRUCTURAS .","EL IN
ICIO DE LA ACCION GENERAL DE FOR
MACION DE LAS ACTITUDES .","LA A
CTUAL COYUNTURA SOCIOECONOMICA .
"
 195 DATA "LA OFERTA COHERENTE D
EL FUTURO MODELO DE SOCIEDAD .",
"LA ABRUMADORA EFICACIA DE LAS D
ISPOSICIONES TRANSITORIAS .","EL
 ABANDONO EN QUE SE ENCUENTRA LA
 ESTRUCTURA PRODUCTIVA .","LA ES
TIMACION OBJETIVA DEL FONDO DE G
ARANTIA DE DEPOSITOS .","EL INDI
CE DE INESTABILIDAD DE LA BALANZ
A DE PAGOS ."
 200  DATA "NOS OBLIGA AL ANALIS
IS .","CUMPLE UN ROL ESENCIAL EN
 LA FORMACION .","EXIGE LA PRECI
SION Y LA DETERMINACION .","AYUD
A A LA PREPARACION Y REALIZACION
 .","GARANTIZA LA PARTICIPACION
DE UN GRUPO IMPORTANTE EN FORMAC
ION .","FACILITA LA CREACION .",
"OBSTACULIZA LA APRECIACION DE L
A IMPORTANCIA .","OFRECE UN ENSA
YO INTERESANTE DE VERIFICACION .
","IMPLICA EL PROCESO DE REESTRU
CTURACION Y DE MODERNIZACION .",
"GENERA EL DESENCANTO ."
 205 DATA "INDUCE UNA RECONVERSI
ON ESPECIFICA DE LA ESCALA DE VA
LORES .","ABRE UN ABANICO DE POS
IBILIDADES DE PROMOCION .","PROV
OCA UN AUMENTO EN LA TASA DE ESP
ECULACION SOBRE EL RENDIMIENTO .
","ALEJA EL FANTASMA DE LOS TIPO
S DE INTERES HASTA OPTIMIZAR LA
RESOLUCION .","NOS CONDUCE A UNA
```

— Permite ejercitar la memoria visual del lector. Para llegar a ser un consumado maestro de la retentiva, basta con someterse a tres sesiones diarias de «Oratrón visual», de un par de horas de duración, durante un periodo de seis meses.

— Si lo que desea es ejercitar su memoria tanto visual como auditiva, se recomienda poner en práctica el mismo método aunque, esta vez, incorporando al SPECTRUM el sintetizador de voz «CURRAH SPEECH».

— Los oradores profesionales —bien sean políticos en ejercicio o ciudadanos a pie que «quieren llegar...»—, pueden encomendar al programa la redacción de sus discursos, con la total seguridad de obtener un resultado exento de contenido, de elevado purismo formal y generosamente surtido de vocablos «al uso».

El usuario puede elegir el formato en el que desee imprimir el texto del discurso (con la ZX-printer o a través de una impresora convencional de 80 columnas).

— Por último, cabe destacar la idoneidad del Oratrón como amenizador de cócktеles, reuniones, comidas de confraternización y cenas pre y post-congresuales. Su alta especialización en la frase vana y el párrafo multi-sentido, garantizan la completa satisfacción de cualquier oyente, sea cual fuere su adscripción política.

Anatomía del Oratón

La cabecera del programa (líneas de la 10 a la 50), se encarga de inicializar las variables y dimensionar las cuatro matrices cuyo contenido coincidirá con los cuatro tipos de frases que figuran en los DATA de las líneas 180 a la 210. Estas últimas se cargan a través de la ejecución de los bucles FOR/NEXT (líneas 60 a 90).

En la línea 100 se selecciona una porción de la primera frase (con la que el programa empieza cualquier discurso) en la matriz A$, bifurcando la secuencia de ejecución a la rutina de visualización localizada en la línea 150. Esta rutina presenta el texto del discurso en pantalla desplazándolo de derecha a izquierda y acompañándolo de un «BEEP» aleatorio, o bien, de la síntesis vocal del texto en el caso de que se conecte al SPECTRUM el módulo «CURRAH SPEECH».

La primera visualización lleva a la pantalla la cadena de caracteres constituida por la suma de Z$ (variable inicializada en la línea 35 con 32 espacios en blanco) y P$. A continuación, el programa incremente el contador de caracteres K, interroga si se ha llegado al final de la matriz y, en caso negativo, transfiere la secuencia de ejecución a la línea 120, tras haber comprobado en la línea 155 si Q es igual a 32. En la línea 120 se produce una bifurcación a la 290, lugar en el que se encuentra la subrutina que selecciona el carácter siguiente (en la frase y matriz adecuadas) dependiendo del valor de G. Al regresar, se ejecuta en 135 una comparación para detectar si se ha llegado al final de la frase (hay que tener en cuenta que las fra-

ses son de longitud variable). Si la comparación es positiva se produce un salto a 110 con el fin de acceder a la siguiente matriz; en el caso contrario, se incrementa el contador de caracteres Q y se prepara la cadena de caracteres a visualizar (línea 145).

La variable Q es la base de un contador para la detección de bloques de 32 caracteres, de tal forma que cuando se ha visualizado este número de caracteres (línea 155) entra en operación la subrutina 400 que los acumula en la cadena Y$. Cuando la longitud de Y$ es de 160 caracteres, se ejecuta la subrutina de impresión (línea 500) que imprime cinco líneas de texto en la ZX-Printer o dos si se trata de una impresora de 80 columnas. Si tiene una impresora conectada al equipo, el usuario observará que en el instante de imprimir se produce una ligera detención del texto visualizado en pantalla... ¡un breve y merecido descanso para el sufrido espectador!

Consejos y sugerencias

En un párrafo anterior hemos afirmado que el Oratrón es capaz de hablar —¡no le quepa la menor duda!—, aunque, bien es cierto, que tal posibilidad sólo podrán utilizarla los propietarios del módulo sintetizador de voz «CURRAH SPEECH». El listado adjunto corresponde a la versión general del programa, ejecutable en un SPECTRUM de 48 K y sin la colaboración del sintetizador vocal. En cualquier caso, el programa es fácilmente adaptable a cualquiera de las alternativas que se relacionan a continuación.

Para que el Oratrón sea ejecutable en un SPECTRUM de 16K, basta con realizar las tres operaciones siguientes:

— Modificar el dimensionado de las matrices de frases (A$, B$, C$ y D$) de la línea 40, limitándolo a (10,90) en lugar de (15,90).

— En la tercera instrucción de la línea 110, reemplazar INT (RND*15) por INT (RND*10).

— Suprimir las líneas 185, 195, 205 y 215 (instrucciones DATA) del programa original adjunto.

La forma de acondicionar el programa para que pueda operar con el sintetizador de voz «CURRAH SPEECH» es bastante más simple y, por lo demás, aplicable tanto a la versión 16 como 48 K. El lector avezado habrá descubierto la presencia de un REM en la línea 151 que inhibe su ejecución. Esta es precisamente la línea que debe sustituir a la 150 en el caso en el que quiera dar voz al Oratrón. La solución más fácil, cómoda e inmediata consiste en eliminar el REM situado al principio de la línea 151 y colocarlo al principio de la 150. Con esta ligera modificación, el Oratrón puede controlar al módulo sintetizador de voz para que éste deleite el oído de los presentes con su densa palabra. Hay que advertir que la vocalización del discurso va a sonarle con un «acentillo» un tanto raro, pero todo es acostumbrarse.

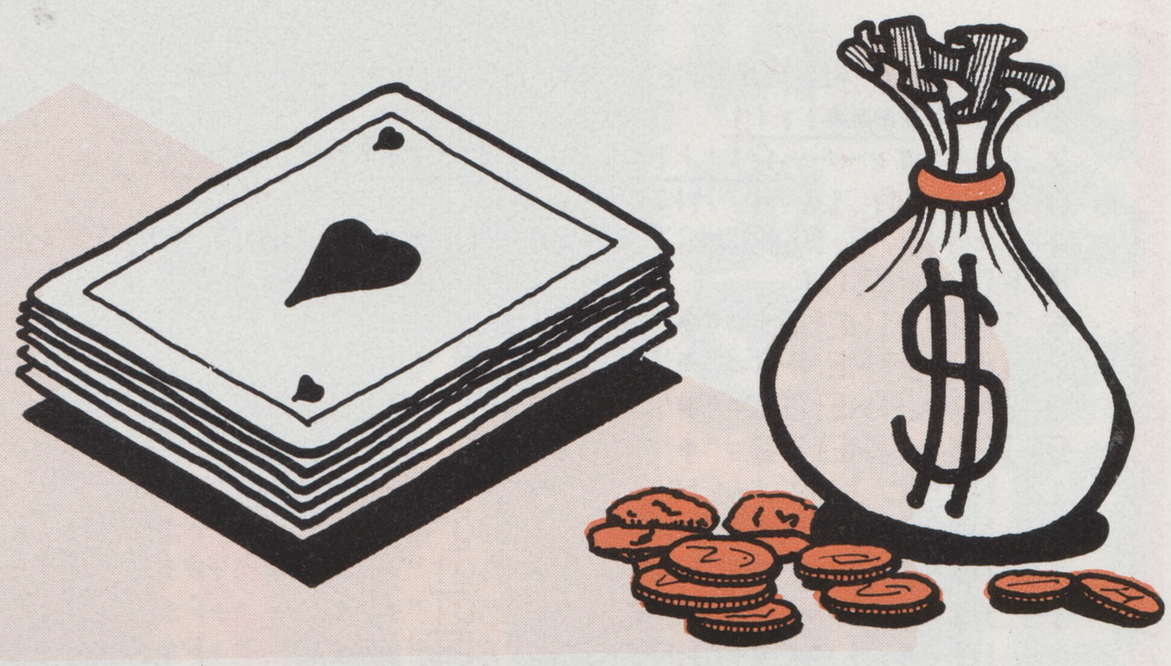
Por último, ni que decir tiene que cabe la posibilidad de que el usuario modifique las frases tipo localizadas en los DATA; por supuesto, teniendo en cuenta que las matrices se han dimensionado de tal forma que las frases tengan un número máximo de 90 caracteres. Aquí, la mezcla puede producir matrices y a costa de mucho teclear, puede obtener una mayor variación en las frases que se traducirá en discursos más perfectos (¡y tal vez convincentes!) Es posible que más adelante nos dé por ingeniar nuevas, interesantes e instructivas aplicaciones en las que intervenga el inquebrantable Oratrón.

Micro

```
 CONCATENACION ."
 210 DATA "DE LAS CONDICIONES FI
NANCIERAS Y ECONOMICAS EXISTENTE
S ."," DE LAS DIRECTIVAS DE DESAR
ROLLO PARA EL FUTURO ."," DEL SIS
TEMA DE PARTICIPACION GENERAL ."
," DE LAS ACTITUDES DE LOS MIEMBR
OS DE LAS ORGANIZACIONES HACIA S
US DEBERES ."," DE LAS NUEVAS PRO
POSICIONES ."," DE LAS DIRECCIONE
S EDUCATIVAS EN EL SENTIDO DEL P
ROGRESO ."," DEL SISTEMA DE FORMA
CION DE CUADROS QUE CORRESPONDA
A LAS NECESIDADES ."," DE LAS CON
DICIONES DE LAS ACTIVIDADES APRO
PIADAS ."," DEL MODELO DE DESARRO
LLO ."," DE LAS FORMAS DE ACCION
."
 215 DATA "DEL CONCEPTO CONFUSO
DE ECONOMIA SUMERGIDA ."," A TRAV
ES DE LA COOPERACION INTERDEPART
AMENTAL ."," ACORDE CON LA IDIOSI
NCRASIA AL USO ."," COMPATIBLE CO
N LAS ESPECIFICACIONES REQUERIDA
S ."," A PESAR DE SU EVIDENTE TEN
DENCIA AL ALZA ."
 280 REM SELECCION TIPO FRASE
 290 IF G=1 THEN  GO TO 340
 300 IF G=2 THEN  GO TO 350
 310 IF G=3 THEN  GO TO 360
 320 IF G=4 THEN  GO TO 370
 330 LET G=1
 340 LET P$=A$(J,K): RETURN
 350 LET P$=B$(J,K): RETURN
 360 LET P$=C$(J,K): RETURN
 370 LET P$=D$(J,K): RETURN
 380 REM
 390 REM IMPRESION 32/80 COL
 400 LET Q=0: LET Y$=Y$+Z$: IF L
EN Y$=160 THEN  GO SUB 500
 410 RETURN
 500 LPRINT Y$: LET Y$="": RETUR
N
```

END

Hijo-poker



De apuestas va el programa... ¿Qué tal anda su bolsillo? ¿Se precia de tener una buena visión de la jugada? Pues aquí está su oportunidad: ¡apueste al «Hijo-poker» y pronto descubrirá cuan apropiado resulta el nombre?

El programa desarrolla un conocido juego español (aunque, en nuestro caso, con baraja francesa). Se trata, ni más ni menos, de superar a la carta que extraiga el ordenador al azar con las cuatro cartas que le hayan tocado en suerte; desde luego, si con cualquiera de sus cuatro cartas es capaz de superar a la del ordenador, el importe de la apuesta pasará a engrosar su bolsa. La escala de valores o puntuaciones que determina qué carta predomina sobre otra es la siguiente:

A>K>Q>J>10>9>8>7>6>5>4>3>2

Y, por supuesto, para ganar a la carta extraída es necesario que el jugador posea una carta del mismo palo y puntuación superior.

Como suele ser habitual al tener al ordenador como compañero de juego, éste se ocupará de casi todo: barajará las cartas y hará el reparto, mostrando en la pantalla las cuatro cartas que han correspondido al jugador. Acto seguido, éste puede ya realizar su apuesta (y ¡ojo! que no se admiten trampas: las apuestas negativas serán consideradas positivas por el ordenador). Una vez cerrada la apuesta, el propio ordenador extraerá la carta oculta y comunicará a los presentes quién ha resultado vencedor. La pantalla visualizará el texto oportuno y se emitirán dos sonidos distintos según le haya sonreído o no la fortuna.

Antes de cada apuesta el jugador puede observar sus disponibilidades monetarias en la parte superior de la pantalla. Por lo demás, el ordenador rechazará aquellas apuestas que excedan de la máxima permitida; esta se define antes de empezar el juego, al tiempo que se establece el importe de cada jugada.

El funcionamiento del programa es simple. En primer lugar se construye la baraja y, a continuación, se redefinen los caracteres. Concluida esta primera etapa se salta al bucle principal que se repite indefinidamente.

Para introducir el programa a través del teclado es recomendable aprovechar las posibilidades que brinda el editor de pantalla del COMMODORE -64; las siguientes instrucciones en modo directo facilitarán el trabajo en gran medida. Veamos:

a) introduzca las siguientes instrucciones:

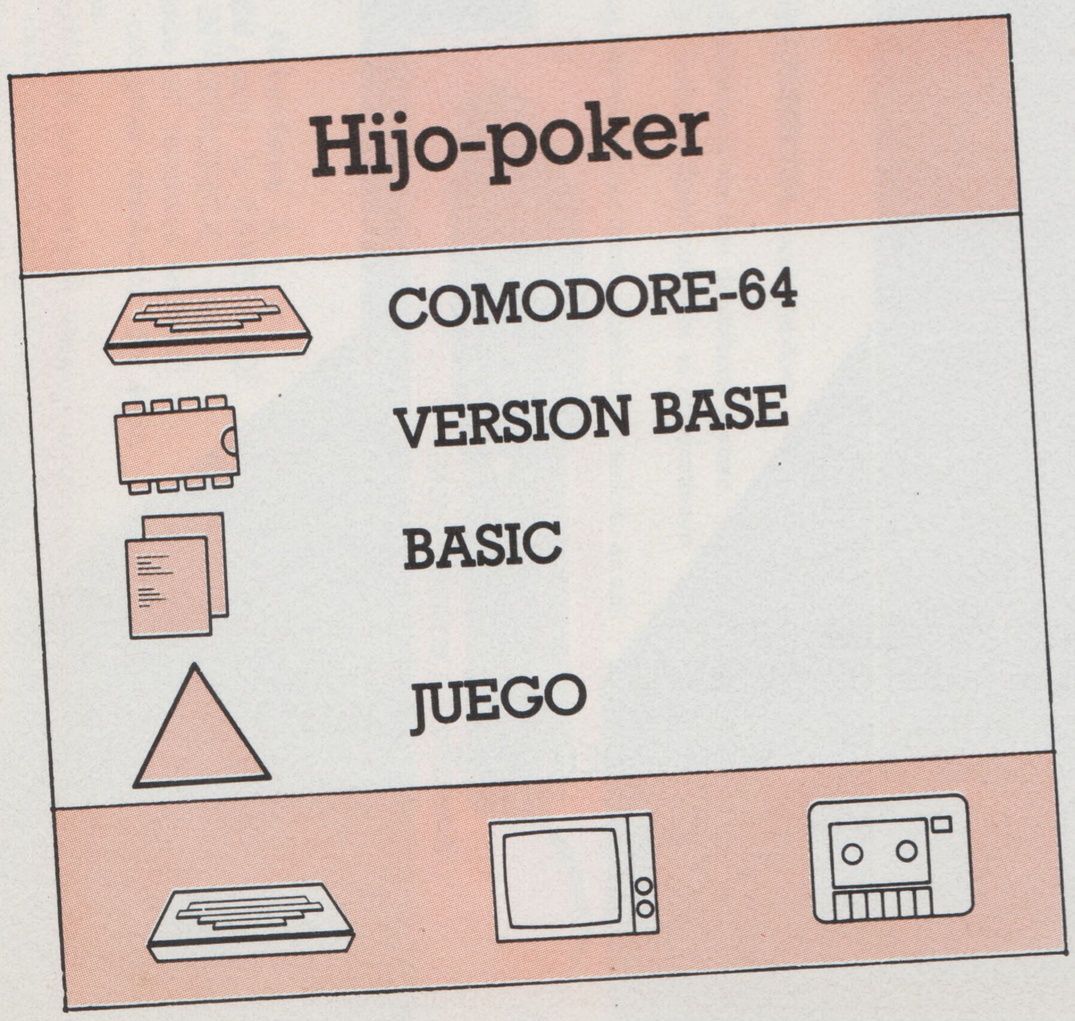
POKE 650, 12B (Autorrepetición de teclas).

FOR J=12 TO 24: PRINT J "N$("; J—10; ",J)=C$+CHR$(C)+": NEXT

Ahora, lleve el cursor al primer PRINT y mantenga pulsada la tecla RETURN.

Con esta simple receta el ordenador escribirá automáticamente casi la mitad de las sentencias de las líneas 12 a la 24. Por supuesto, luego debe volver sobre ellas para completarlas.

b) Una instrucción más:



LISTADO



```
0 REM ** HIJOPOKER ** A. CARVAJAL **
1 S$="●●●●||||":S=54272
2 C$="■ ┌──┐Q|||||| |Q|||||| |Q|||||| |Q|||||| |Q|||||└──┘":C$=C$+S$
3 DIM CA(14,4),N$(14,4)
4 S=54272:FORL=STOS+24:POKEL,0:NEXT:POKES+5,13:POKES+24,15
5 FORJ=1TO4
6 IFJ=1THENP$="♠":C=144
7 IFJ=2THENP$="♣":C=144
8 IFJ=3THENP$="♥":C=28
9 IFJ=4THENP$="♦":C=28
12 N$(2,J)=C$+CHR$(C)+"2Q"+P$+"QQ|"+P$
13 N$(3,J)=C$+CHR$(C)+"3Q"+P$+"Q|"+P$+"Q|"+P$
14 N$(4,J)=C$+CHR$(C)+"4Q|"+P$+" "+P$+"QQ|||"+P$+" "+P$
15 N$(5,J)=C$+CHR$(C)+"5Q|"+P$+" "+P$+"Q||"+P$+"Q||"+P$+" "+P$
16 N$(6,J)=C$+CHR$(C)+"6Q|"+P$+" "+P$+"Q|||"+P$+" "+P$+"Q|||"+P$+" "+P$
17 N$(7,J)=C$+CHR$(C)+"7Q|"+P$+" "+P$+"Q|||"+P$+P$+P$+"Q|||"+P$+" "+P$
18 N$(8,J)=C$+CHR$(C)+"8Q|"+P$+" "+P$+"Q|||"+P$+P$+P$+"Q|||"+P$+P$+P$
19 N$(9,J)=C$+CHR$(C)+"9Q|"+P$+P$+P$+"Q|||"+P$+P$+P$+"Q|||"+P$+P$+P$
20 N$(10,J)=C$+CHR$(C)+"10"+P$+"Q|||"+P$+P$+P$+"Q|||"+P$+P$+P$+"Q|||"+P$+P$+P$
21 N$(11,J)=C$+CHR$(C)+"JQ|"+P$+"[]"+"Q|||"+" ▒▒ "+"Q|||←$"+P$
22 N$(12,J)=C$+CHR$(C)+"QQ|"+P$+"&("+"Q|||"+" ▒▒ "+"Q|||+*"+P$
23 N$(13,J)=C$+CHR$(C)+"KQ|"+P$+"@£"+"Q|||"+"↑↑↑"+"Q|||')"+P$
24 N$(14,J)=C$+CHR$(C)+"AQQ"+P$+"Q"
25 NEXTJ
26 POKE53281,7:POKE53280,5
27 PRINT"♥■Q"TAB(15)"RHIJOPOKER═"
28 PRINT"Q]EL OBJETO DEL JUEGO ES GANAR A LA CARTA"
29 PRINT"]QUE SACA EL ORDENADOR CON UNA DE LAS"
30 PRINT"Q]CUATRO NUESTRAS TENIENDO EN CUENTA QUE:"
31 PRINT"]A>K>Q>J>10>9>8>7>6>5>4>3>2."
32 PRINT"Q]Y QUE LA CARTA SEA DEL MISMO PALO."
33 PRINT"Q]EN ESTE JUEGO NO HAY COMODINES."
34 PRINT"Q]USTED DETERMINA EL VALOR DE CADA JUGADA"
35 PRINT"]ASI COMO EL VALOR DE LA APUESTA MAXIMA."
36 PRINT"]NO SE ADMITEN APUESTAS NEGATIVAS"
37 PRINT"Q            ESPERE POR FAVOR"
38 PRINT"      SE ESTAN CREANDO LAS CARTAS"
46 POKE52,48:POKE56,48
47 POKE56334,PEEK(56334)AND254
48 POKE1,PEEK(1)AND251
49 FORI=0TO2023:POKEI+12288,PEEK(I+53248):NEXT
50 POKE1,PEEK(1)OR4
60 POKE56334,PEEK(56334)OR1
65 POKE53272,(PEEK(53272)AND240)+12
70 FORI=12288TO12295:READA:POKEI,A:NEXT
71 FORI=12504TO12543:READA:POKEI,A:NEXT
72 FORI=12568TO12639:READA:POKEI,A:NEXT
80 DATA 170,124,84,127,108,68,127,255
81 DATA 255,126,84,124,108,68,120,255
82 DATA 0,0,32,252,44,30,63,63
83 DATA 64,96,112,120,88,24,24,24
84 DATA 255,239,239,239,239,239,239,255
85 DATA 24,24,24,26,30,14,6,2
86 DATA 60,60,60,60,60,60,60,60
87 DATA 255,62,34,54,62,42,126,255
88 DATA 255,247,247,247,247,247,247,255
89 DATA 120,124,86,127,111,71,127,255
90 DATA 252,252,120,52,63,4,0,0
91 DATA 0,24,60,60,60,24,24,24
92 DATA 255,254,33,54,254,42,62,85
93 DATA 255,254,226,246,254,106,62,30
95 DATA 24,24,24,60,60,60,24,0
450 INPUT"♥■Q]]]PRECIO DE LA JUGADA";PJ
460 INPUT"Q]]]APUESTA MAXIMA";AM
480 FOR TT=1 TO 2000:NEXT TT:T=9
482 IF G>=0 THEN PRINT "♥←]]]]]]]]]]GANAS:"G"PESETAS"
484 IF G<0 THEN PRINT "♥£]]]]]]]]]]PIERDES:"ABS(G)"PESETAS"
490 FOR K=1 TO 5
500 X=INT(RND(8)*13)+2:Y=INT(RND(8)*4)+1:   IF CA(X,Y)=1 THEN 500
510 CA(X,Y)=1:P(K)=Y:V(K)=X
```

```
515 NEXT K
517 FOR K=1 TO 4
520 PRINT"SQQQQQQQ"TAB(T);N$(V(K),P(K));S$;:T=T+6
525 NEXT K
526 INPUT "SQQQQQQQQQQQQQ]]■APUESTA";A:A=ABS(A):IFA>AM THEN 526
530 FOR K=1 TO 4
540 IF P(5)=P(K) THEN IF V(5)<V(K) THEN     JU=1:GOTO 565
545 NEXT K
546 JU=-1
565 PRINT"SQQQQQQQQQQQQQQQ"TAB(18)N$(V(5),P(5))
570 IF JU=1 THEN PRINTTAB(16)"■GANASTE!!":G=G+A-PJ:II=6E3:FI=16E3:GOSUB700
580 IF JU=-1THEN PRINTTAB(15)"■PERDISTE!!":G=G-A-PJ:II=12E3:FI=2E3:GOSUB700
600 FOR I=1 TO 14:FOR J=1 TO 4:CA(I,J)=0:    NEXT J,I:GOTO480
700 FORM=IITOFISTEPJU*200:HF=INT(M/256):LF=M-HF*256:POKES+1,HF:POKES,LF
710 POKES+4,17:NEXT:POKES+4,16:RETURN
```

END

FOR J=27 TO 38: PRINT J "DATA": NEXT

Actúe de igual forma que en el caso anterior y se encontrará con que ya tiene escritas parte de las instrucciones de las líneas 27 a la 38.

c) Para escribir de forma automática la parte común de las líneas 80 a la 94, debe aplicar el referido método después de introducir:

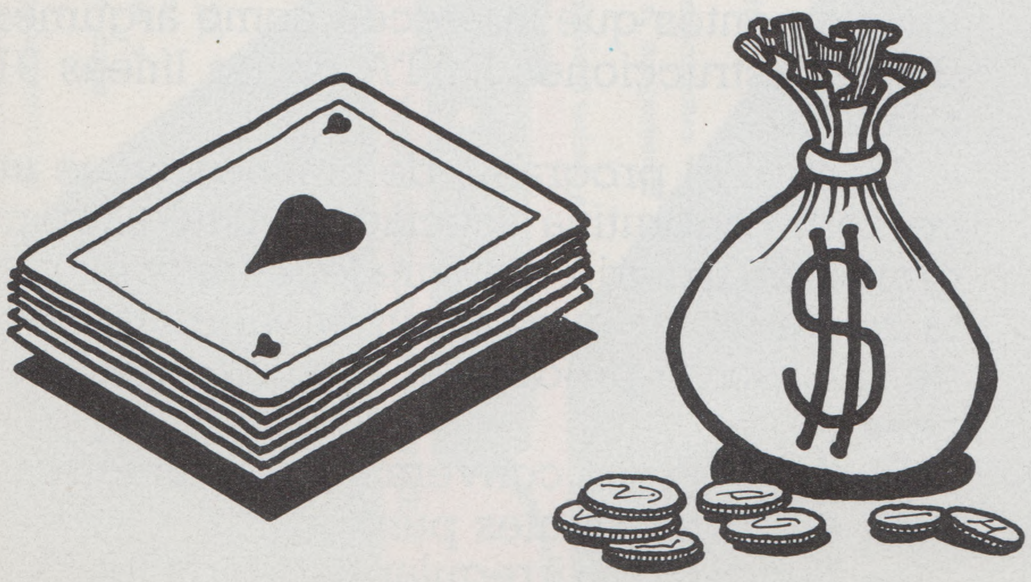
FOR J=80 TO 94: PRINTT J "DATA": NEXT

Con este recetario el propio COMODORE-64 ocupará el lugar de sus dedos introduciendo una buena parte del programa... ¡Más facilidades, imposible!

tuMicro

ESTRUCTURA DEL PROGRAMA

1-3	Definición de variables.
4	Inicialización del chip de sonido.
5-25	Construcción de la baraja.
26-38	Instrucciones y colores.
46-72	Redefinición de caracteres.
80-95	Datos sobre los caracteres.
450-460	INPUTs precio y apuesta máxima.
480-600	Bucle principal.
700-710	Subrutina de sonido.



CUADRO DE VARIABLES

A	Variable para la lectura de DATAs. Valor de la puesta.
AM	Valor de la apuesta máxima.
C	Color del palo correspondiente.
C$	Variable string para dibujar el rectángulo de la carta.
CA(14,4)	Matriz de cartas que han salido o no.
FI	Final frecuencias.
C	Plan de ganancias o pérdidas.
HF	Alta frecuencia (Byte alto).
II	Inicicio frecuencias.
I, J, L	Variables FOR.
JU	(1, ganamos. –1, perdemos).
K, TT	Variables FOR.
LF	Baja frecuencia (Byte bajo).
N$(14,4)	Todas las cartas construidas.
P$	Palo de la carta.
PJ	Precio de la jugada.
P (5)	Palo de cada una de las cinco cartas.
S	Puntero de sonido.
S$	Variable string auxiliar.
T	Variable para la colocación de la carta (TAB(T)).
V(5)	Valor de cada una de las cinco cartas.
X	Número de carta, aleatorio.
Y	Palo de la carta, aleatorio.

Literales numéricos

En la edición de documentos comerciales es necesario, con frecuencia, que aparezcan los importes o cantidades tanto en números como en letras. El tratamiento numérico es muy sencillo para el ordenador, cosa que no ocurre con el tratamiento de literales; por ello, siempre resulta de utilidad tener a mano una subrutina que permita la conversión de números a su expresión alfabética.

La subrutina que presentamos, a pesar de su relativa sencillez y reducida extensión, es capaz de convertir en literales números naturales comprendidos entre 1 y 999.999. Aunque este margen de trabajo suele ser más que suficiente en las aplicaciones habituales, no resulta difícil extenderlo para lograr que admita números hasta 999 millones.

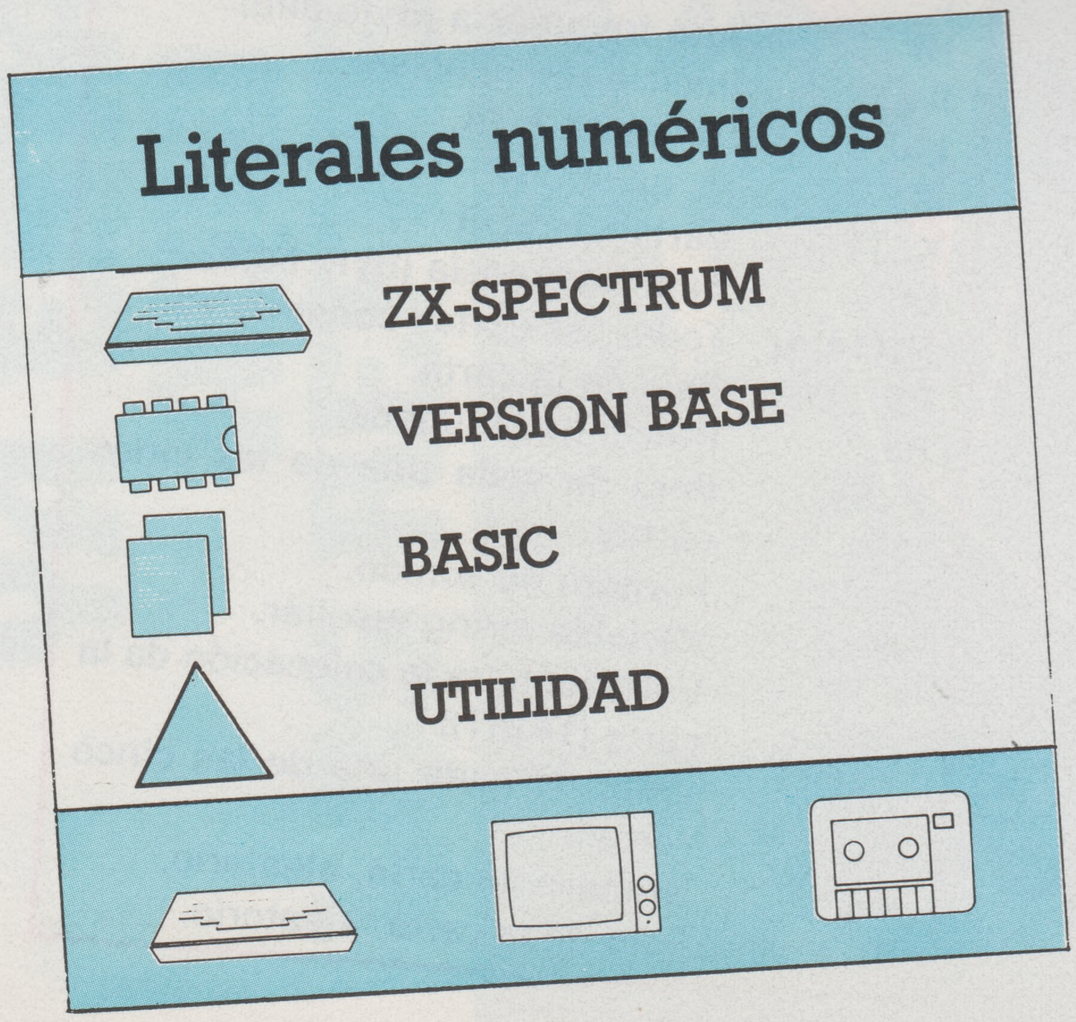
Las líneas 10 a la 60 del listado adjunto no forman parte integrante de la subrutina conversora propiamente dicha, sino que han sido incluidas para permitir la comprobación de la efectividad de la misma por parte del usuario. Una vez introducido el listado, el programa puede ya ejecutarse por medio del comando RUN. La primera tarea que realizará el ordenador será pedir al usuario que introduzca el número a convertir (recuerde que debe estar comprendido entre 1 y 999.999); de inmediato, el número en cuestión será procesado por la rutina conversora y el resultado aparecerá sobre la pantalla.

La rutina que nos ocupa, confeccionada para el ZX-SPECTRUM, toma el número a procesar dentro de la variable de cadena de caracteres A$ y emite el resultado alfabético en la cadena. S$. Un dato reseñable es que el bloque inicial del programa inicializa y crea la tabla alfanumérica T$, tabla que estará integrada por los elementos que aparecen como argumento de las instrucciones DATA de las líneas 9140 a la 9170.

Cuando el programa de conversión se utilice como subrutina asociada a cualquier programa del usuario, hay que tener muy en cuenta las variables que entran en juego con el fin de que no se produzcan reutilizaciones erróneas.

El proceso de conversión alfanumérica se basa en los siguientes puntos:

— Respetar las irregularidades de los 29 primeros números.



LISTADO

```
  10 REM Fernando Lopez Martinez # Literal numerico # ZX Spectrum 16/48 Kb.
  20 DIM T$(45,13)
  30 FOR I=1 TO 45: READ T$(I): NEXT I
  40 INPUT "NUMERO?": LINE A$: IF VAL A$<1 OR VAL A$>999999 THEN  GO TO 40
  50 PRINT A$;" = PESETAS: ";: GO SUB 9000: PRINT S$
  60 GO TO 10000
9000 LET S$="": LET Z=0
9010 IF LEN A$>3 THEN  LET Z=1: LET B$=A$( TO LEN A$-3): GO TO 9030
9020 LET B$=A$: IF B$(1)="0" THEN  LET B$=B$(2 TO )
9030 IF LEN B$<3 THEN  GO TO 9070
9040 IF B$="100" THEN  LET S$=S$+"CIEN": GO TO 9110
9050 LET Z$=T$(VAL B$(1)+36): GO SUB 9180: LET S$=S$+Z$+" "
9060 LET B$=B$(2 TO )
9070 IF VAL B$=0 THEN  GO TO 9110
9080 IF VAL B$<30 THEN  LET Z$=T$(VAL B$): GO SUB 9180: LET S$=S$+Z$: GO TO 9110
9090 LET Z$=T$(VAL B$(1)+27): GO SUB 9180: LET S$=S$+Z$: IF B$(2)="0" THEN  GO T
O 9110
9100 LET Z$=T$(VAL B$(2)): GO SUB 9180: LET S$=S$+" Y "+Z$
9110 IF Z THEN  LET S$=(S$ AND S$<>"UNA")+" MIL ": LET Z=0: LET A$=A$(LEN A$-2 T
O ): GO TO 9020
9120 LET Z$=S$: GO SUB 9180: RETURN
9140 DATA "UNA","DOS","TRES","CUATRO","CINCO","SEIS","SIETE","OCHO","NUEVE","DIE
Z","ONCE","DOCE"
9150 DATA "TRECE","CATORCE","QUINCE","DIECISEIS","DIECISIETE","DIECIOCHO","DIECI
NUEVE","VEINTE"
9160 DATA "VEINTIUNA","VEINTIDOS","VEINTITRES","VEINTICUATRO","VEINTICINCO","VEI
NTISEIS","VEINTISIETE","VEINTIOCHO","VEINTINUEVE"
9170 DATA "TREINTA","CUARENTA","CINCUENTA","SESENTA","SETENTA","OCHENTA","NOVENT
A","CIENTO","DOSCIENTAS","TRESCIENTAS","CUATROCIENTAS","QUINIENTAS","SEISCIENTAS
","SETECIENTAS","OCHOCIENTAS","NOVECIENTAS"
9180 FOR I=LEN Z$ TO 1 STEP -1: IF Z$(I)<>" " THEN  RETURN
9190 LET Z$=Z$( TO I-1): NEXT I
```

END

— Composición de las decenas y unidades.
— Complemento de las centenas sobre el grupo de decenas y unidades.
— Estudio del grupo de centenas-decenas-unidades de millar, siguiendo la misma rutina y limitándose, simplemente, a agregar a continuación el literal «MIL».

El análisis se efectúa de izquierda a derecha, recortando la cadena según se va codificando en la variable de almacenamiento S$.

Cabe destacar que el programa incluye una subrutina anidada (subrutina de subrutina), cuya finalidad es suprimir los blancos que el intérprete de BASIC Sinclair deja para homogeneizar las longitudes de los elementos de las tablas alfanuméricas.

Mida

ESTRUCTURA DEL PROGRAMA

10-20	Inicialización y lectura de la tabla T$ a partir de la información contenida en las instrucciones DATA.
30-60	Toma de dato y llamada a la subrutina de conversión.
9000	Comienzo de subrutina. Inicialización.
9010-9120	Tratamiento de la cadena numérica.
9140-9170	DATA con estructuras alfabéticas básicas.
9180-9190	Subrutina truncadora de blancos por la derecha.

CUADRO DE VARIABLES

I	Variable FOR de diversa utilidad.
Z	SWITCH para analizar los millares.
A$	Cadena de entrada con el número a procesar.
B$	Cadena puente en el tratamiento de A$.
S$	Cadena para construcción del literal alfabético.
Z$	Cadena para el truncado de blancos por la derecha.
T$	Tabla alfanumérica de longitud 45, que contiene las estructuras alfabéticas básicas, para la construcción del literal final.

Un canal para el diálogo de los ordenadores ATARI con cualquier magnetófono a casetes.



Casadapter

Equipo:
CASADAPTER
Interface para magnetófono a casetes.

Fabricante:
SAR-AN

Distribuidor:
UNIMPORT
Dos amigos, 3 Madrid

¡Y al fin llegó! Un interface largamente esperado por los usuarios de microordenadores ATARI obligados, hasta ahora, a utilizar la unidad de casetes ATARI 1010 como periférico de almacenamiento externo.

El atractivo económico de este nuevo dispositivo de interface es obvio: 5.900 ptas. más un magnetófono de tipo común (¡valen incluso los magnetófonos estéreo!), frente a las 16.900 de la unidad ATARI 1010.

Una alternativa que el usuario desearía que se extendiera a otros microordenadores como, por ejemplo, VIC-20, COMMODORE-64 o SPECTRAVIDEO.

Aunque distribuido en España por la firma UNIMPORT, importadora de la gama de microordenadores ATARI, el CASADAPTER es un producto cuyo origen es una sociedad americana dedicada a la fabricación de dispositivos y unidades complementarias para sistemas microordenadores: SAR-AN.

El interface para casete se presenta en una caja de plástico, a través de cuyos laterales acceden al exterior dos mangueras de cable. Una dispone en su extremo de un conector, compatible con los presentes en las unidades centrales de ATARI o en su familia de periféricos, para la conexión en cadena de unidades periféricas. La segunda manguera, destinada a conectar el interface con el magnetófono a casete, incor-

pora en su extremo tres conectores de tipo jack y uno de tipo americano «plug». De los tres primeros, el microjack es para conectar a la toma de control remoto del motor (REMOTE), el jack negro esta destinado a la toma EAR del magnetófono y el gris a MIC. El cuarto conector, de tipo «plug americano», es para la toma de AUDIO en el caso de utilizar un magnetófono estéreo.

la Realizadas las oportunas conexiones, la unidad central ATARI queda en disposición de utilizar como memoria de masa cualquier magnetófono convencional. La forma de operar coincide exactamente con la habitual para lectura y escritura en la unidad de casete ATARI 1010.

Y llegó el momento de hablar de la cal y de la arena del CASADAPTER. Veamos algunos detalles significativos. En primer lugar, cabe mencionar que los casetes grabados a través del CASADAPTER son compatibles con la unidad ATARI 1010 y, por supuesto, en ambos sentidos, esto es: un casete grabado con el CASADAPTER puede ser leído por el ATARI 1010 y viceversa. Por lo demás, el software de ATARI suministrado en cinta puede ser cargado en la unidad central recurriendo al CASADAPTER asociado a un magnetófono convencional.

Otro punto a señalar, aunque esta vez no tan favorable, es que no es posible bobinar o rebobinar la cinta a casete mientras esté conectada la clavija REMOTE; lo cual obligará al usuario a conectar y desconectar la clavija en cuestión cuando trate de situar la cinta en un determinado punto de su recorrido.

Es posible que al realizar las primeras pruebas con el CASADAPTER surja algún que otro problema a la hora de cargar los programas... ¡Que no cunda el pánico! En esta tesitura, recuerde que debe ajustar los controles de volumen y tono del magnetófono, de tal forma que la señal que entregue sea la idónea para garantizar la carga.

GRABACION de programas en casete

Introducir el comando CSAVE seguido por una acción sobre la tecla RETURN. Una señal acústica recordará al usuario la necesidad de poner en marcha el magnetófono en modo grabación (teclas REC y PLAY accionadas simultáneamente); acto seguido basta con una nueva acción sobre RETURN para que empiece la grabación.

CARGA de programas almacenados en casete

Introducir el comando CLOAD y pulsar seguidamente la tecla RETURN. La respuesta del ordenador será un pitido recordatorio de que debe conectar el magnetófono en modo reproducción (tecla PLAY accionada).



DIRECTORIO

LG LUGO

COMPONENTES
AUTOSERVICIO
BARQUILLO, 40
4198742-4198751

ELECTRONICA SANDOVAL S.A.

DISTRIBUIDORES DE:
ORIC-1
CASIO FP200
ROCKWELL-AIM-65
VIDEO GENIE-EG-2000
CASIO FX-9000P
SINCLAIR ZX81
OSBORNE 1
DRAGON-32
NEW BRAIN
EPSON HX-20

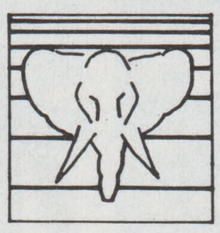
ELECTRONICA SANDOVAL, S. A.
C/ SANDOVAL, 3, 4, 6 - MADRID-10
TELEX 47784 SAVL-E

ALICANTE
A. W.

Oric-1	Spectrum
New Brain	Spectravideo
M.P.F. 1, 2, 3	Sharp
XEROX	T.R.S.-80
Vic	Impresoras

Cursos de programación todos los meses
C/ Calderón de la Barca, 2
Teléfono 21 91 28

CONTERSA



·Xerox ·Oric Atmos
·Alphatronic ·Toshiba
·New Brain ·Atari

Impresoras Microdrives
Programas

General Ricardos, 117-3°D
☎ 471 22 14/23 62

MICRO M WORLD

ORDENADORES PERSONALES Y MICROORDENADORES DE GESTION

- SPECTRUM
- ORIC ATMOS
- NEW BRAIN
- KATSON
- APPLE
- ALTOS

SOFTWARE STANDARD Y A MEDIDA

CURSILLOS Y FORMACION

En Madrid:
MODESTO LAFUENTE, 63. Teléf. 253 94 54

D digital s.a.

— Los ordenadores personales son demasiado importantes como para permitirse el lujo de ignorarlos.

— Ordenadores personales software, hardware, libros periféricos.

LASER COLOR COMPUTER 200 · COMMODORE 64
ZX SPECTRUM · ORIC-1
VIC 20 · New Brain

Distribuimos el software
publicado por tu MICRO

C/ BERLIN, 5 DPDO. - MADRID-28
TELEFS.: 246 49 90 - 246 56 63



COMPTER'S
Todo en Microcomputadores

ORDENADORES.
— SHARP
— TOSHIBA
— EPSON
— STAR

Programas garantizados para todas las gestiones de la Empresa.

Estación de Chamartín
Planta Comercial S-14 B
Teléf. (91) 215 51 60 - Madrid-16

secoinsa

DIRECCION CENTRAL
Almagro, 40. MADRID-4.
Tels.: 435 78 36/37. Télex: 7901.
Apartado de Correos 36215.

CENTROS DE INVESTIGACION Y FABRICAS

BARCELONA-15. Rocafort, 98-100. Teléfonos (93) 325 41 00. Telex 51688.

MALAGA. Polígono Industrial Guadalhorce. Parcela, 21. Teléf. (952) 330000. Telex 77142.

DELEGACIONES Y CENTROS DE SERVICIO EN ESPAÑA

ALICANTE-7. Alona, 21. Teléf. (965) 22 03 02/03.

BARCELONA-28. Gran Vía de Carlos III, 105. 1.ª planta. Teléf. (93) 330 62 53.

BILBAO-10. Dr. Areilza, 31-33. Teléfs. (94) 432 44 06/07/08.

GERONA. Calle de la Cruz, 2 (bis), entresuelo, 8. Teléf. (972) 20 31 28.

LAS PALMAS. Alcalde Ramírez Bethancourt, 45, 1.º. Edificio Rocamarín. Teléf. (928) 36 49 11

MADRID-20. Pedro Teixeira, 8, 1.ª planta Teléf. (91) 455 40 04.

MANRESA (Barcelona). Paseo de Pedro III, 20. Teléf. (93) 874 61 54.

MURCIA. Junterones. Edificio Sierra Espuña (entresuelo). Teléf. (968) 24 05 01

OVIEDO. Plaza de América, 10. Teléfono (985) 24 46 23/24.

PALMA DE MALLORCA. Avda. A. Roselló, 15, 2.º B. Edificio Minaco. Teléfono (971) 22 13 28.

PAMPLONA. Emiliano Arrieta, 16, 5.º, departamento 2. Teléfonos (948) 22 15 04/22 39 05.

SAN SEBASTIAN. Paseo de la Concha, 14 Telf.: (943) 42 47 51/6

SANTA CRUZ DE TENERIFE. Almirante Díaz Pimienta, 8. Edificio Isla de Tenerife. Teléf. (922) 21 14 88.

SEVILLA-5. Avda. San Francisco Javier, sin número. Edificio Sevilla II, 2.º. Teléfs. (954) 64 76 00/04.

TARRAGONA. Avda. de Andorra, s/n. CONSTANTI
TARRASA (Barcelona). Gutenberg, 3-13, 4.ª planta. Teléfs. (93) 780 06 00/0308/01 8...

VALENCIA-10. Alvaro de Bazán, 19. Teléfs. (96) 360 29 50/54/58.

VALLADOLID. P.º Isabel la Católica, 3,, 1.º Izq. Teléf. (983) 92 45 08

VIGO (Pontevedra). Manuel Núñez, 2, 2.ª planta. Teléfs. (986) 22 56 15/22 86 30.

VITORIA. Francia, 21
Telf.: (945) 26 95 98

ZARAGOZA. Plaza de Nuestra Señora del Carmen, 7-8, 5.º. Edificio Mercurio. Teléfs. (976) 21 95 63/21 34 63/22 84 90.

OTROS CENTROS DE SERVICIO

BARCELONA-18. Pedro IV, 29-35. Teléf. (93) 309 53 00.

MADRID-17. Valentín Beato, 11, 3.º. Teléf. (91) 754 41 16.

División de Electrónica e Informática del INI

tuMicro

digital s.a.

NUEVO

LASER™ COLOR COMPUTER 200

UNIDAD PRINCIPAL 4 K RAM 16 K ROM. 8 COLORES EN PANTALLA. SONIDO GRAFICOS ALTA RESOLUCION 128×64 SUPERECONOMICO. AMPLIABLE. PROGRAMAS Y PERIFERICOS DISPONIBLES.

indescomp

New Brain

VIC-20

ADEMAS:

Periféricos; soft ware hardware; Biblioteca Técnica, etc.
PARA TODOS ELLOS.

COMMODORE 64

ORIC-ATMOS

DRAGON 64

Distribuimos todas las cintas que aparecen en la revista tu MICRO.

Silicítenoslas, se las enviaremos cómodamente a su casa contra reembolso o con cargo a su tarjeta de crédito (costos de envío 200 ptas.).

ESPECIALISTAS EN VENTA POR CORREO ¡¡CONSULTENOS!!

YA DISPONIBLE, LA NUEVA EDICION DEL CATALOGO GENERAL DE ELECTRONICA E INFORMATICA PARA VENTA POR CORREO DE MAYOR DIFUSION DE ESPAÑA

Solicite hoy su ejemplar adjuntando 200 pesetas en sellos de correos, nuevos, enviando el cupón a:

digital, s.a.

C/ BERLIN, 5
DPDO.
MADRID-28

Teléfonos
246 49 90 - 246 56 63

APELLIDOS .....
NOMBRE .....
DIRECCION .....
POBLACION .....
PROVINCIA ..... TFNO. .....

DESEO RECIBIR, SIN COMPROMISO, EN MI DOMICILIO, EL CATALOGO GENERAL DE ELECTRONICA E INFOMATICA 1984, PARA LO CUAL ADJUNTO 200 PESETAS EN SELLOS DE CORREOS NUEVOS.

CUPON DE PEDIDO
SOLICITELO A

digital s.a.

APDO. DE CORREOS
8287 - MADRID

Enciclopedia Práctica de la

INFORMATICA

52 *fascículos coleccionables y encuadernables en* **4** *tomos*

VIVIMOS EN EL MUNDO DE LA INFORMATICA



A LA VENTA EN SU QUIOSCO LOS MIERCOLES

¡DOMINALO!

En cada fascículo 6 secciones

- INFORMATICA BASICA: Los conocimientos fundamentales.
- HARDWARE: Los 52 ordenadores más vendidos.
- SOFTWARE: Los programas: como son y como se hacen.
- PERIFERICOS: Como comunicarse con el ordenador.
- APLICACIONES: Presentación y comentario de programas.
- EL MUNDO DE LA INFORMATICA: Lo insólito y lo práctico.

1.048 PAGINAS A TODO COLOR ✱ MAS DE **3.000** ILUSTRACIONES

UNA OBRA EXCEPCIONAL REALIZADA POR LOS MEJORES ESPECIALISTAS

Si desea recibir los fascículos comodamente en su casa recorte o copie este cupón y remítalo a:
Ediciones INGELEK, S.A. Avd. Alfonso XIII, 141. Madrid-16.

Deseo suscribirme a la obra: Enciclopedia Práctica de la INFORMATICA (52 fascículos más 4 tapas para la encuadernación) por el precio de 7.400 Ptas.
Dicha obra la recibiré: ◯ Dos fascículos cada 15 días y las tapas junto con los fascículos 13, 26, 39 y 52.
◯ Un tomo cada 13 semanas.

NOMBRE: _____
DOMICILIO: _____
CIUDAD: _____
PROVINCIA _____

El importe de dicha suscripción lo abonaré:

◯ Talón a nombre de INGELEK, S.A.
◯ Giro Postal nº:
◯ Reembolso del importe de la suscripción más gastos de envio.

Fecha y firma

NUEVA INFORMATICA

Lo curioso es que todos los puntos que hacen a APD altamente rentable son debidos a su nueva tecnología. Lógico.

El costo de los equipos APD es sensiblemente más bajo. Compárelo.
Sus posibilidades de crecimiento son superiores. Descúbralo.
Sin contactos, sin cables, sin problemas de refrigeración...
Un diseño MONOPLACA, compacto, exclusivo y fiable.
APD: Máxima rentabilidad.

El costo, el crecimiento y un sistema monoplaca.

Comparta la opinión de todos los expertos, ponga a su alcance el Sistema Operativo más evolucionado y poderoso (OASIS®).
Tan sencillo que en pocas horas podrá dominarlo.
Tan poderoso que nunca se le quedará "corto".
Un sistema operativo fuerte para un equipo fiable y potente es la fórmula APD.

La superioridad del sistema operativo OASIS®

Un año es un período de garantía muy largo. Sólo equipos muy probados, muy duros y muy fiables pueden mantenerlo.
APD se lo ofrece de entrada para todos nuestros productos. Y para Vd., un año de confianza y seguridad.

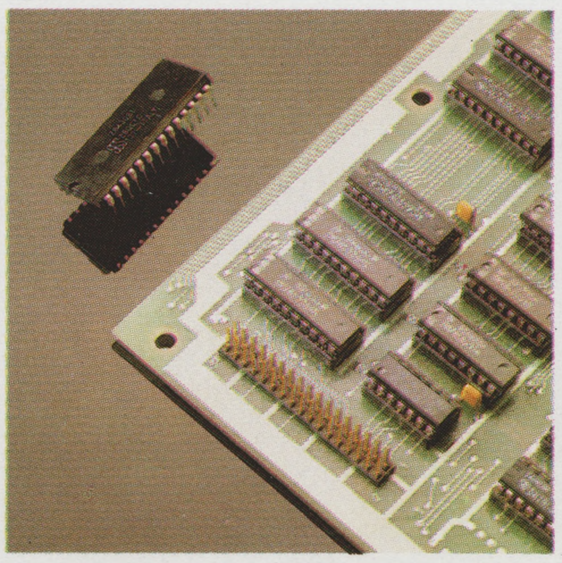
¿Por qué 1 AÑO de garantía?

Un equipo APD puede controlar muchos puestos de trabajo por el mismo precio que otros sólo controlan uno. Puede realizar varios trabajos a la vez, por muy diferentes que éstos sean: CIENTIFICOS, TECNICOS, DE GESTION... Y SIMULTANEAMENTE.
APD: Tecnología, versatilidad y potencia.

Unidades inteligentes y muy versátiles.









APD©

APD es más que un ordenador, es un sistema basado en la experiencia y en las necesidades reales. Pida información.

TECNOLOGIA RENTABLE

Si desea recibir mayor información, no dude en llamarnos.

SU CONTACTO CON APD©

MADRID 435 22 65
435 23 76
435 26 77
Castelló, 63. MADRID-1

BARCELONA 209 70 00
Aribau, 267 / Párroco Ubach
BARCELONA-21

VALLADOLID 22 53 51
Acera de Recoletos, 11-4.º D
VALLADOLID

SEGUIMOS BUSCANDO DISTRIBUIDORES



PONTE A LOS MANDOS DE UN SPECTRUM.

Ahora tu microordenador SPECTRUM es, aún, MAS
con sus nuevos refuerzos: Microdrive, Interface 1, Interface 2...

¡Por fín podrás grabar y leer información de manera casi instantanea!.

¡O disfrutar a lo grande con la más extensa variedad de programas
tanto educativos como de mero entretenimiento!.

Y sobre todo vas a tener la posibilidad de aprender a programar
(que siempre te será muy útil) de una manera fácil y divertida.

No dejes pasar esta ocasión,
ahora que puedes obtener mayor rendimiento de tu SPECTRUM.

Solicita información en la Red de Concesionarios Autorizados Investrónica.

J M PUBLICIDAD

ESTE VERANO PONTE A LOS MANDOS DE UN SPECTRUM



IMPORTANTE:

Al adquirir los productos **SINCLAIR** exija la TARJETA DE GARANTIA INVESTRONICA, única válida en todo el territorio nacional y llave para cualquir resolución de duda o reparación.
INVESTRONICA no prestará ningún servicio técnico a todos aquellos aparatos que carezcan de la correspondientes garantía.

DE VENTA EN CONCESIONARIOS AUTORIZADOS.

DISTRIBUIDOR EXCLUSIVO:
INVESTRONICA
CENTRAL COMERCIAL: Tomás Bretón, 60
Tel. 468 03 00 Telex: 23399 IYCO E Madrid
DELEGACION CATALUÑA: Camp, 80 - Barcelona - 22