

Enero 1985 250 Ptas.

Todospectrum

AÑO 1 - NUMERO 5.

REVISTA EXCLUSIVA PARA USUARIOS



Diseño gráfico por ordenador

OPUS o la rapidez de los floppys

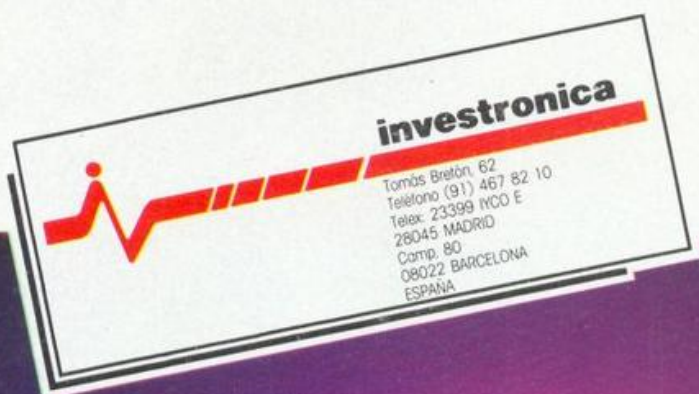
64 caracteres en código máquina

Programas en BASIC, Pascal y Forth

Simulador de vuelo, damas chinas ...

ZX Spectrum + (64 K.)

Para los que exigen +



A buen seguro que el saco lleno de Spectrums que portaba nuestro Santaclair del número anterior, se quedó rápidamente vacío. Para los nuevos usuarios, nuestro más caluroso recibimiento, y para los demás, seguro que tampoco se olvidó de ellos Santaclair y les trajo algún periférico.

Y como hablamos de periféricos, nada mejor que empezar el año echando una miradita al Opus, la unidad de disco que se hiciera famosa en Inglaterra, ahora comercializada en España.

En el capítulo de *software*, dos son los programas más significativos con los que podrá presumir de realizar el CAD/CAM con su Spectrum (es decir, diseño gráfico por ordenador) y si usted es profesor, modernizarse en el tratamiento de evaluaciones.

Los aficionados a los juegos verán una pequeña reducción en el número de programas, que se compensa con la calidad de los mismos, y que fue necesario para dar cabida a otros temas que se nos pedían. En este sentido, esperamos saciar la curiosidad de quienes nos pedían que explicásemos cómo hacíamos las pantallas.

Feliz año nuevo.

- 4 **OPUS, O LA RAPIDEZ DE LOS FLOPPYS.** Analizamos la primera unidad de discos disponible para el Spectrum.
- 8 **DISEÑO ASISTIDO POR... SPECTRUM.** Un programa en BASIC permite el diseño gráfico.
- 18 **64 CARACTERES POR LINEA.** Una corta rutina en código máquina cambia completamente el aspecto de su pantalla.
- 26 **JUEGO DE LA VIDA.** BASIC comentado a fondo mediante este popular juego.
- 30 **DESCUBRIMIENTO DE UN NUEVO LENGUAJE: PASCAL (2).** Segunda parte del lenguaje que hace furor en Telecomunicaciones.
- 35 **ASI HACEMOS LAS PORTADAS.** Hablamos con Víctor Ruiz sobre cómo realiza las portadas de TODOSPECTRUM, que tanto han llamado la atención de los lectores.
- 38 **CONTROL DE EVALUACIONES.** Programa de pedagogía especialmente recomendado para profesores.
- 48 **PROGRAMAS: Simulador de vuelo, Damas chinas, Romanos en Forth y Pascal.**
- 55 **GUSANEZ.** Los reyes le traen un Plus.
- 64 **PREGUNTAS Y RESPUESTAS.** Un lector nos propone una nueva rutina en código máquina: esta vez relocalizable.
- 66 **PROGRAMANDO ENTRE AMIGOS.** Entrevistamos al grupo de programadores de Software Centre en Barcelona.



Con la carta a sus Majestades,
y un poquito de ayuda del programa
Artist, no nos faltaron los regalos.

Número

DIRECTOR:

Simeón Cruz

COORDINADOR EDITORIAL:

J. A. Sanz

REDACCIÓN:

Juan Arencibia, Fernando García, José C. Tomás, Gumersindo García, Luis M. Brugarolas, Ricardo García.

DISEÑO:

Ricardo Segura

Editado por PUBLINFORMATICA, S. A.

Presidente: Fernando Bolin.

Director Editorial: Norberto Gallego.

Administración:

INFODIS, S. A.

Gerente de Circulación y Ventas:

Luis Carrero.

Producción:

Miguel Onieva.

Director de Marketing:

Antonio González.

Servicio al Cliente:

Julia González. Tel. 733 79 69.

Administración:

Miguel Atance y Antonio Torres.

Jefe de Publicidad:

María José Martín.

Dirección y Redacción:

Bravo Murillo, 377, 5.º A Tel. 733 74 13

Telex: 48877OPZX e 28020 Madrid

Administración y Publicidad:

Bravo Murillo, 377, 3.º E Tels. 733 96 62/96

Publicidad Madrid:

Nieves Clemente.

Publicidad Barcelona:

María del Carmen Ríos, Jorge González Pelayó, 12.

Tel. (93) 301 47 00 Ext. 27 y 28 08001 Barcelona

Depósito legal: M-29041-1984

Distribuye: S.G.E.L.

Avda. Valdelaparra, s/n Alcobendas - Madrid.

Fotomecánica: Karmat, C/ Pantoja, 10, Madrid.

Fotocomposición: Artecomp.

Imprime: Héroes, C/ Torrelara, 8, Madrid.

Esta publicación es miembro de la Asociación de

Revistas de Información ARI, asociada a la

Federación Internacional de Prensa Periódica,

FIPP.

ari

SUSCRIPCIONES:

Rogamos dirijan toda la correspondencia relacionada con suscripciones a:

TODOSPECTRUM

EDISA: Tel. 415 97 12

C/ López de Hoyos, 141-5.º

28002 MADRID

(Para todos los pagos reseñar solamente TODOSPECTRUM)

Para la compra de ejemplares atrasados

diríjanse a la propia editorial

TODOSPECTRUM

C/ Bravo Murillo, 377 - 5.º A

Tel. 733 74 13

28020 MADRID



OPUS

Desde el pasado mes de junio, más de cien unidades han sido comercializadas por la firma Silog. Acostumbrados a otros periféricos y al mismo Spectrum, su precio puede parecer ligeramente elevado. Pero la relación precio-prestaciones lo hacen más que aconsejable. Con ellos se cierra la oferta de periféricos de almacenamiento de datos: cintas, cartuchos microdrive o wafadrive y discos. Toda una panoplia de ofertas diferentes en calidad y precio.

Del mismo color negro que el Spectrum, la unidad de discos **Opus** y el *interface* están preparados para una cómoda y fácil conexión, sin desentonar con su ordenador. Las dimensiones del *interface* (14 x

9,5 x 2,5 cm.) imposibilitan la entrada de corriente, previsto por la nueva entrada vía *interface*, pero no así la conexión MIC, de difícil o imposible acceso con el *interface* conectado (aunque se trabaje con disco, siempre puede ser interesan-

te grabar datos en un *cassette*). Si piensa en el *interface* 1 como separador, mejor olvídelo: no son compatibles. El resto de las conexiones no se ven afectadas, quedando abierto un nuevo *port* de expansión en la parte posterior, para la incorporación de nuevos periféricos.

El *interface* puede controlar discos de 40 a 80 sectores, densidad normal o doble. En otras palabras, se puede disponer de discos con capacidades que oscilan entre las 100 y 400 Kbytes en sus distintas configuraciones. Por el momento, la firma **Silog** sólo comercializa los de 40 pistas-100 Kbytes, objeto de nuestro análisis. Teniendo en cuenta que se pueden conectar si-

multáneamente hasta un máximo de cuatro unidades, esto arroja la respetable cifra de 1.600 Kbytes o 1,6 Megabytes de memoria externa.

Un amplio cable de conexión con el *interface* permite trabajar cómodamente, a lo que se unen las pequeñas dimensiones de la unidad de discos (15 x 23 x 4,5 cm.) para lo que viene siendo normal. Pero no es este su mayor atractivo. Al margen de la capacidad antes descrita, está la posibilidad de trabajar con ficheros de datos aleatorios. Una simple comparación entre los distintos sistemas de almacenamiento, nos da una idea de la velocidad: cargar el extenso programa BASIC de pedagogía que se incluye en este número, supuso los siguientes tiempos según el medio empleado:

Cassette.....3'34"
Microdrive0'09"
Disco Opus0'11"

El dato del *microdrive* corresponde al promedio de cinco lecturas distintas, ya que según la posi-

nuevamente en el DOS, se teclea RANDOMIZE USR 15360.

Algunos comandos son ya conocidos, como es el caso de FORMAT, ERASE, LOAD, SAVE, MOVE y CAT, de utilización con el *microdrive*. Al igual que éstos, los discos han de inicializarse para poder trabajar con ellos. RUN "format" CODE permite trabajar con discos de cuatro formatos distintos:

* Mini-floppy Format Ver 2.3*
© 1984 Technology Research Ltd
1 _____ 40 track single sided
2 _____ 80 track single sided
3 _____ 40 track double sided
4 _____ 80 track double sided
Q _____ Quit

Which format

Suponiendo que elige la opción de 40 sectores densidad normal, aparecerá el siguiente mensaje:

* 40 track single side *
Which drive? (A, B, C, D, Menu) A

Please insert disk in drive A
When you are ready, type
F for format
V for verify
Q for quit

El comando CAT informa del contenido del disco. Las tres primeras líneas indican el nombre del disco, el número de archivos existentes y borrados, respectivamente. Las siguientes líneas muestran los nombres de los programas y algo muy importante, si están grabados en BASIC o en código máquina <C>. El número que le acompaña indica la longitud del programa en sectores (1 Kbyte = 4 sectores). La última línea informa de la memoria disponible.

Title: Utility
3 File(s)
0 Del. File(s)
A: format <C> 10 : copy 2
A: copy <C> 14
364 Free

SAVE permite la grabación de programas, del mismo modo que

o la rapidez del floppy

ción de la cinta los tiempos de acceso pueden diferir significativamente. La velocidad es, por tanto, similar al *microdrive*, si tenemos en cuenta su mayor capacidad.

Para ver las restantes características, hemos de analizar los nuevos comandos que incorpora el *interface*. Conectado al Spectrum, aparece el mensaje de la versión de *interface* y el nombre de Technology Research en la parte superior. En la parte inferior aparece el símbolo "A>", que indica que se está en el DOS (Disk Operating System o Sistema Operativo por Disco), y no en el BASIC de Sinclair. Para volver al BASIC se presiona RETURN y ENTER. Para entrar



estamos acostumbrados con *casette* o *microdrive*.

SAVE "archivo"
SAVE "archivo" CODE 36000

Del mismo modo, LOAD realiza la carga del programa. Una vez finalizada esta, se pasa automáticamente al BASIC, listándose el programa.

LOAD "archivo"
LOAD "archivo" CODE 36000

Comando que se puede sustituir por RUN para la carga y ejecución automática del programa (aunque no estuviese grabado con la instrucción SAVE "LINE XXXX").

RUN "archivo"
RUN "archivo" CODE

También es posible utilizar la instrucción MERGE para combinar dos o más programas en BASIC.

MERGE "anterior"
MERGE "b:nuevo:

en este segundo caso se cargaría el programa "nuevo" de la segunda unidad de discos conectada, en su caso, combinándose con el residente en memoria.

La eliminación de ficheros es igualmente realizada por ERASE, del mismo modo que si se tratase del *microdrive*.

ERASE "archivo"
ERASE "archivo" CODE
ERASE "archivo" DATA

pero con una notable diferencia: los "huecos" resultado de eliminar ficheros pueden eliminarse por efecto de una nueva instrucción. MOVE (no confundir con la utili-

zación que de este comando hace el *interface* 1) reorganiza la información para evitar la dispersión de datos típicos del *microdrive*, causante de los incrementos en tiempos de acceso conforme se va utilizando.

MOVE

Otro comando que trabaja de forma diferente a lo que estamos habituados es NEW. En esta ocasión no se elimina nada de la memoria. Sirve para cambiar la denominación de un archivo o programa (recuerde que en el *microdrive* esto supone cargar el programa, grabarlo con el nuevo nombre y eliminar el nombre antiguo, sin perjuicio de la imposibilidad de reorganizar la cinta, salvo que se vuelva a formatear).

NEW "nuevo nombre", "nombre anterior"
NEW "nuevo nombre", "nombre anterior" CODE
NEW "nuevo nombre", "nombre anterior" DATA

Y finalmente nos encontramos con los comandos PEEK y POKE. Potentes herramientas que nos facilitarán el acceso aleatorio al disco. PEEK lee los sectores del disco (un sector por acceso), colocando la información en un *buffer* RAM.

PEEK "nombre archivo" dirección *buffer*, número registro
PEEK "TEST" 30023,1

Lee el primer sector de TEST y queda en *buffer* RAM con dirección 30023.

POKE realiza la operación inversa, es decir, graba la información por sectores:

POKE "nombre archivo" dirección *buffer*, número registro
POKE "TEST", 30023,1

La información contenida en la dirección 30023 se graba en el primer sector del archivo TEST.

Se puede, por tanto, trabajar con archivos aleatorios y olvidarse de los secuenciales (única posibilidad con la cinta y *microdrive*), aunque la elección de un tipo u otro de archivo dependerá de las aplicaciones a realizar. Sin embargo, el manejo de archivos por este sistema dista mucho de ser sencillo. Para facilitar esta labor, las instrucciones OPEN#, CLOSE#, PRINT#, INPUT#, acuden en ayuda del programador, por el momento no disponibles para esta unidad, aunque según pudimos informarnos, la firma que lo distribuye está construyendo un EPRON al efecto.

Si le parece interesante y decide reforzar su Spectrum, cuando pregunte precios no olvide especificar *interface* y unidad de discos. En el caso del Opus, la unidad son 55.200 pesetas, más las 29.850 pesetas del *interface*.

Una característica importante es la posibilidad de utilizar los comandos DOS desde el BASIC e incorporarlos así, a nuestros programas, mediante el uso de la instrucción RANDOMIZE:

RANDOMIZE USR 15363 : REM
: (Comando del DOS)

así por ejemplo, si se desea cargar y ejecutar un programa llamado "disco" la instrucción LOAD "disco" pasaría a ser:

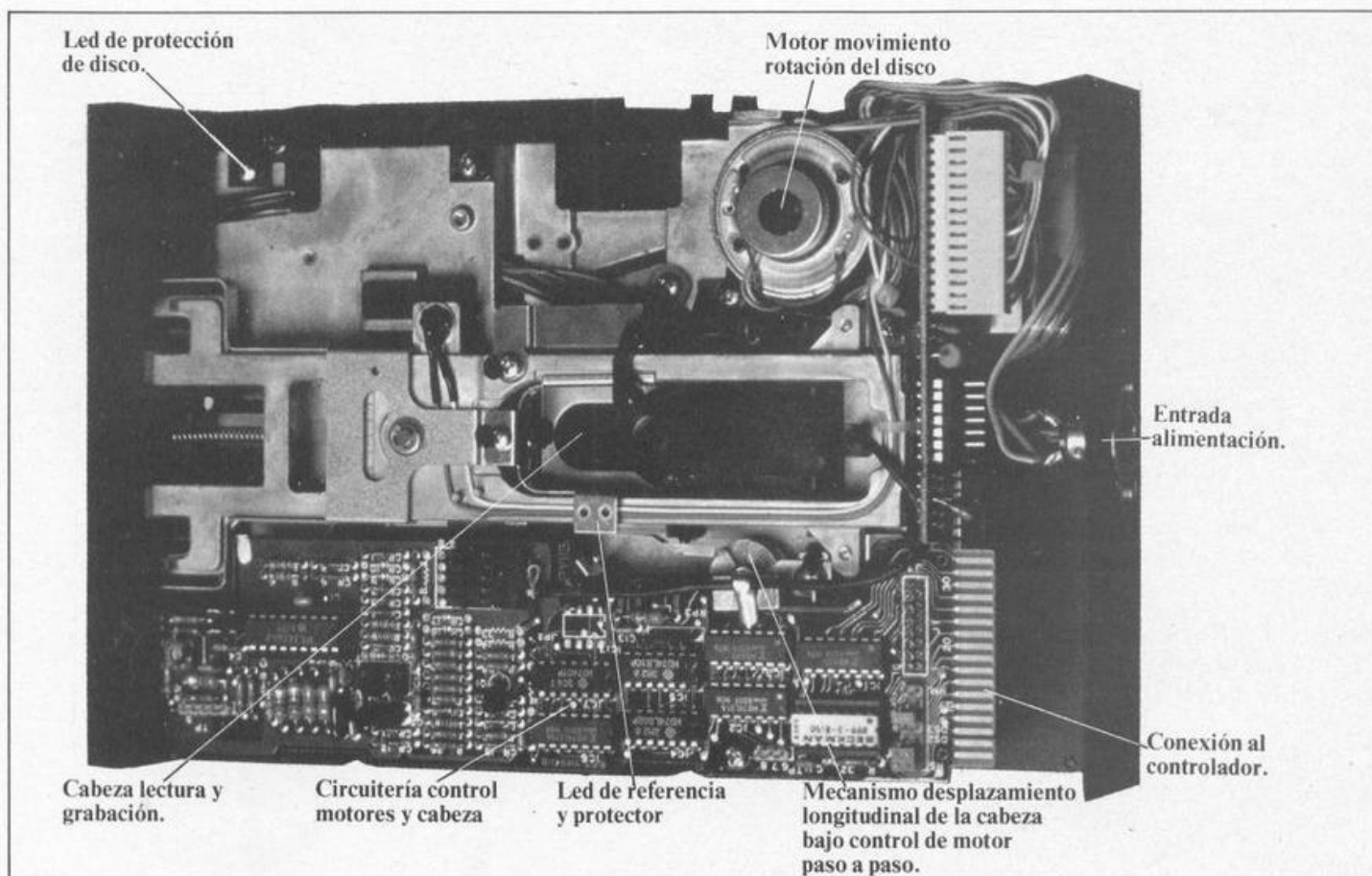
RANDOMIZE USR 1536 : REM :
RUN "disco"

únicamente ha de tener en cuenta que la instrucción que contenga un comando DOS ha de ser la última, dentro de una línea.

Pero no sólo se puede acceder desde el BASIC. También desde código máquina es posible. Para ello se ha de alterar la variable del sistema CHADD. Veámoslo con un ejemplo:

dirección	código	
40000	234	REM
40001	58	:
40002	239	LOAD





40003	34	"	40008	112	p
40004	101	e	40009	108	l
40005	106	j	40010	111	o
40006	101	e	40011	34	"
40007	109	m	40012	13	ENTER

utilizando la dirección 40000, puede escribirse el código máquina como sigue:

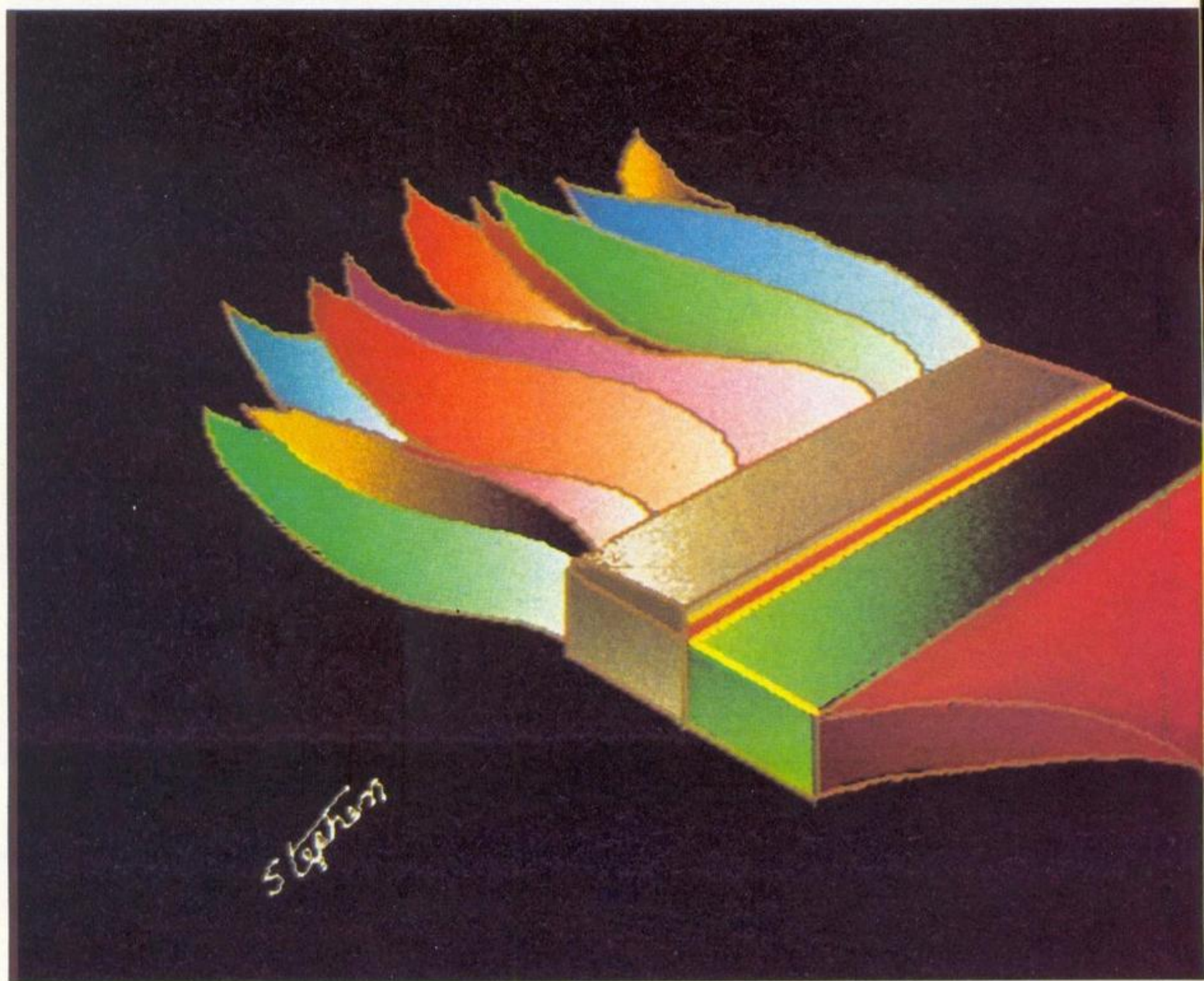
```
LD HL,(CHADD)      Siendo
                    CHADD 23645
LD (TEMP),HL       Se guarda en
                    una posición
LD (CHADD),40000   Se direc-
                    ciona CHADD a la 40000
CALL 15363          Llamada al
                    DOS
LD HL,(TEMP)
LD (CHADD),HL      Se devuel-
                    ve su contenido inicial
RET
```

Finalmente, junto con el interfa- ce se acompaña un disco con los *utility programs* o programas de utilidad. En él hay cuatro progra- mas en BASIC que llama a otros cuatro programas en código má- quina: BACKUP, COPY, FOR- MAT, SCOPY.

Format permite el formateo de discos, como ya vimos. Diversas modalidades de copias, tanto par- ciales de programas o archivos, como totales de disco a disco, se realizan por los tres programas res- tantes.

NUEVOS COMANDOS DEL DOS

***A:"	Selección unidad A.
***B:"	Selección unidad B.
***C:"	Selección unidad C.
***D:"	Selección unidad D.
CAT	Catálogo.
ERASE	Eliminación archivo.
LOAD	Carga de un programa.
MERGE	Combinación de dos o más pro- gramas.
MOVE	Reorganización de ficheros.
NEW	Cambio del nombre de un fichero.
PEEK	Lectura aleatoria de un fichero.
POKE	Escritura aleatoria de un fichero.
RANDOMIZE	
USR	
15360	Paso al DOS
RETURN	Retorno al BASIC
RUN	Carga y ejecución de un programa.
SAVE	Grabación de un programa.
USR	Cambio de la clave de acceso.



Diseño asistido SPECTRUM

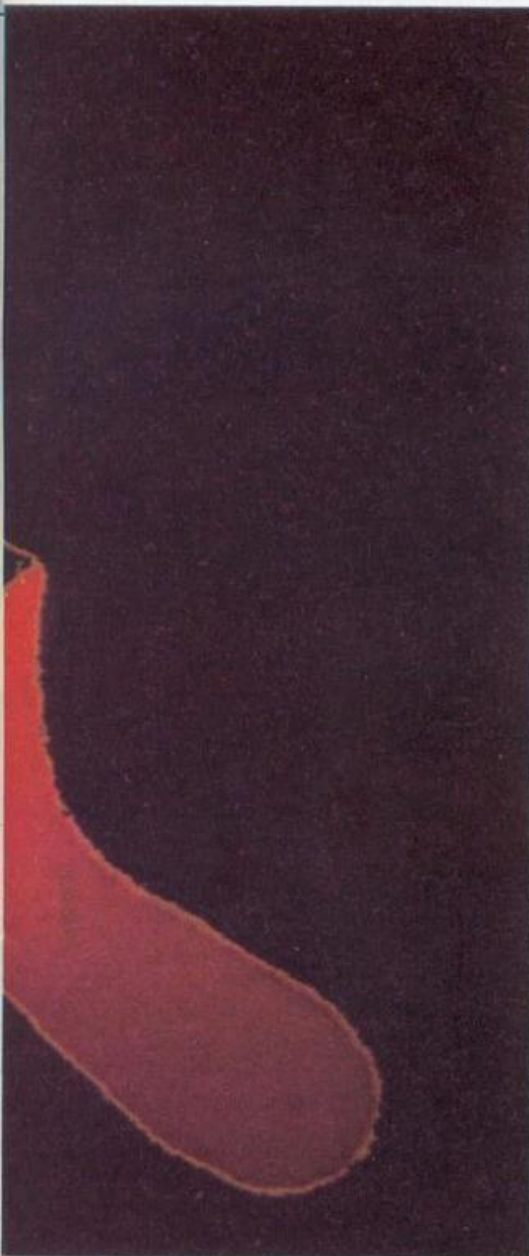
Para muchos usuarios, las matemáticas son fascinantes, primero, por la facilidad con que se resuelven problemas bastante complejos y en segundo lugar como herramienta que posee fuerza propia. Siempre y cuando no se alteren las reglas del juego se podrá conseguir el resulta-

do más inesperado de un problema que parezca sencillo. (Como por ejemplo, los resultados de la mecánica cuántica y la relatividad, donde las respuestas obtenidas a través del uso de las matemáticas se contradicen con el sentido común.)

Hasta la llegada del micro, esta fascinación por las matemáticas

era algo pasajero. Con demasiada frecuencia resolver un problema con papel y lápiz se hacía demasiado difícil como para poder continuar. Era como estar luchando contra un pulpo: te librabas de un brazo para caer en el siguiente.

¿Pero, cómo nos ayudan los ordenadores? Hasta la computadora



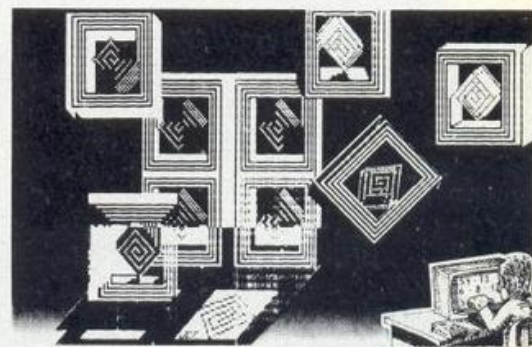
por...

más barata está hecha para hacer operaciones matemáticas a unas velocidades muy superiores de lo que podemos imaginar. Miles de cálculos iterativos son llevados a cabo por una simple línea de programación en BASIC. Los programas no sólo se hacen para efectuar dichos cálculos, sino que también

pueden tomar decisiones "lógicas" acerca de qué método hay que seguir para realizar dicha operación.

En ninguna parte se notan estas ventajas tanto como en el campo de los gráficos generados por ordenador. De todos modos, las demandas en cuanto a la potencia de la computadora, tanto en capacidad de almacenamiento como en velocidad, son enormes y por esto los programas de gráficos diseñados por computadora (CAD) han sido probados en ordenadores muy avanzados, normalmente máquinas con procesadores de 32 y 48 bits, los cuales al trabajar en paralelo los hacen ultrarrápidos. Estos programas en código máquina son asistidos por *hardware* especial para el rápido tratamiento de los números y procesamiento de la pantalla.

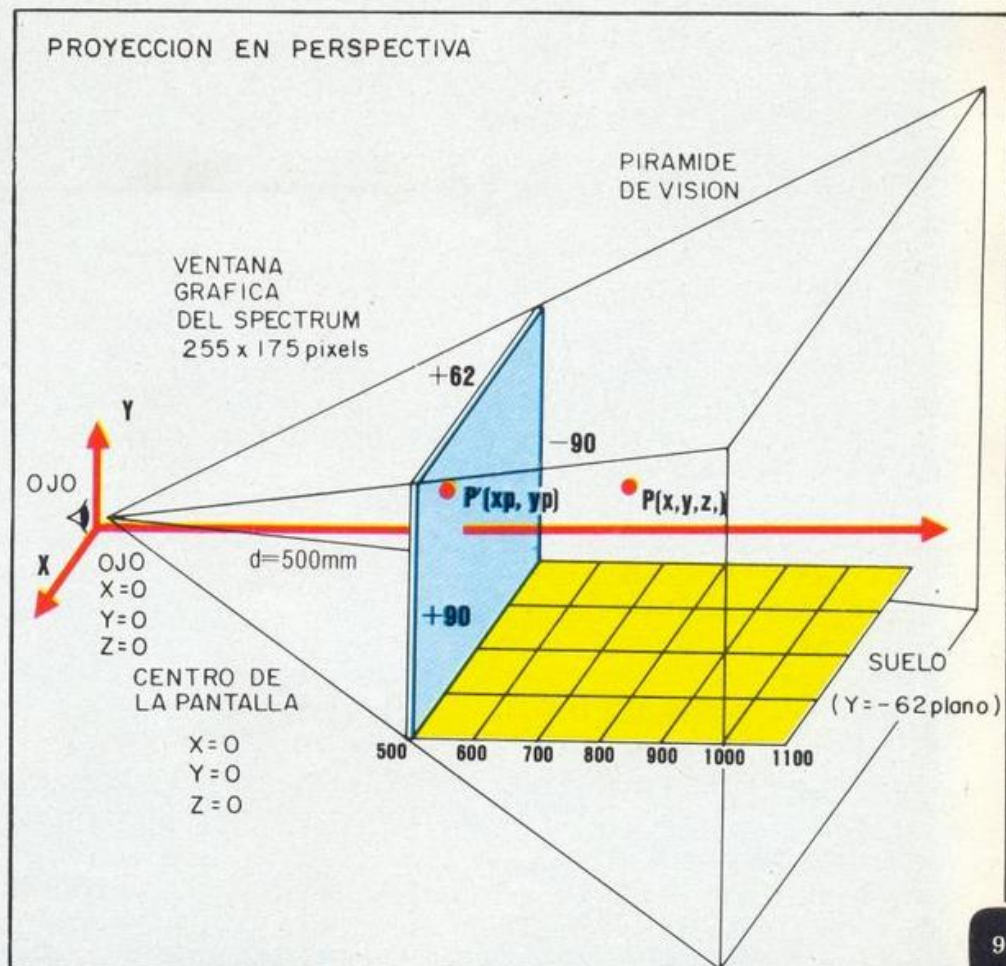
Por lo tanto no es de extrañar la escasez de programas de gráficos para ordenadores cuya configuración sea de 8 bits que son la inmensa mayoría de los ordenadores domésticos. Aunque el CAD parezca estar más allá de las posibili-

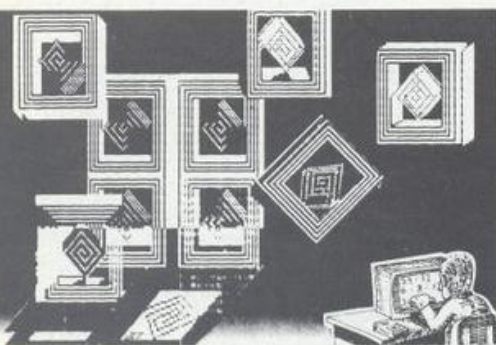


dades del micro, han sido diseñados varios programas didácticos a la vez que entretenidos.

Planteamiento del proyecto

El problema reside en la creación de diversos contornos y sombras mediante la aplicación de funciones matemáticas en vez de "plotear" la pantalla con puntos diversos. Con la elección de coordenadas adecuadas, se podrán hacer girar determinadas figuras geométricas, trasladarlas o dibujarlas





a escala sin la necesidad de transformaciones matriciales.

La mejor forma de comprenderlo todo es a través de un programa

que genera gráficos en 3 dimensiones. Para los genios matemáticos con experiencia en programación BASIC, entender el ejemplo será coser y cantar y para aquellos a los que no les interesan tanto las matemáticas y que quieran teclear el programa y pasárselo bien tienen un ejemplo para que comprueben los bonitos gráficos que se pueden hacer con el Spectrum. Habiendo llegado a ambos extremos del usuario típico, el resto está hecho desde el punto de vista de aquellos que deseen aprender, utilizar y desarrollar el concepto principal usa-

do en el programa y sus ejemplos. Ahora entremos en los detalles del programa, una vez entendido pruebe con algunos ejemplos expuestos.

Observando en perspectiva

La proyección en perspectiva se puede introducir en la rutina 520, que es la que calcula la perspectiva del suelo.

Hay una serie de datos que se asumen:

1. El origen del sistema rectan-

```

10 REM objetos en perspectiva
20 GO SUB 440: STOP
30 PRINT AT 21,13;"v=-62mm";AT
11,26;"x=90mm";AT 0,13;"y=62mm"
;AT 11,0;"x=-90mm"
40 LET p=1.41
50 LET d=500
60 GO SUB 510
70 INPUT AT 0,0;"Interseccion
de planos: v/h";a$
80 IF a$="h" THEN GO TO 660
90 INPUT "Dist. del obj. en mm
;z0=";z0
100 INPUT "'Radio' obj. en mm;
R=";R
110 INPUT "Punto central obj. (m
m);x0=";x0
120 INPUT "y0=";y0
130 INPUT "Num. de sec. vert.;s
b=";sb
140 INPUT "Num. lados de cada s
ec.;sa=";sa
150 INPUT "Long. inicial en gra
d.;b0=";b0
160 INPUT "Lat. inicial en grad
.;a0=";a0
170 LET db=360/sb
180 LET da=360/sa
190 FOR b=b0 TO 360+b0 STEP db
200 FOR a=a0 TO 360+a0 STEP da
210 LET x=R*COS (PI/180*a)*COS
(PI/180*b)+x0
220 LET v=R*SIN (PI/180*a)+y0
230 LET z=R*COS (PI/180*a)*SIN
(PI/180*b)+z0
240 LET dx=R*COS (PI/180*b)*(CO
S (PI/180*(a+da))-COS (PI/180*a)

```

```

)
250 LET dy=R*(SIN (PI/180*(a+da
))-SIN (PI/180*a))
260 LET dz=R*SIN (PI/180*b)*(CO
S (PI/180*(a+da))-COS (PI/180*a)
)
270 IF x>z/5.6 THEN LET x=z/5.
6
280 IF (x+dx)>z/5.6 THEN LET d
x=z/5.6-x
290 IF y>z/8.3 THEN LET y=z/8.
3
300 IF (y+dy)>z/8.3 THEN LET d
y=z/8.3-y
310 IF x<-z/5.6 THEN LET x=-z/
5.6
320 IF (x+dx)<-z/5.6 THEN LET
dx=-z/5.6-x
330 IF y<-z/9 THEN LET y=-z/9
340 IF (y+dy)<-z/9 THEN LET dy
=-z/9-y
350 LET xp=x*d/z: LET yp=y*d/z
360 PLOT xp*p+128,yp*p+87
370 LET dk=d/(z+dz)-d/z
380 LET dxp=dx*d/z+x*dk: LET dy
p=dy*d/z+y*dk
390 LET e=ABS ((z-d)/200)
400 DRAW INK e;dxp*p,dyp*p
410 NEXT a
420 NEXT b
430 GO TO 70
440 PLOT 0,0
450 DRAW 255,0
460 DRAW 0,175
470 DRAW -255,0
480 DRAW 0,-175
490 BORDER 5

```


gular de coordenadas es el ojo del observador.

2. Todas las distancias se expresan en mm. (otras unidades se pueden especificar fácilmente, mientras se mantenga a lo largo del programa).

3. Los milímetros son convertidos en pixels sólo en las líneas del programa donde haya PLOT y DRAW.

4. El número de pixels/mm (p) para la ventana gráfica del Spectrum en una TV de 12 pulgadas se asume igual a 1.41 (255/180=1.41). Este valor nomi-

nal es el que se utiliza a lo largo del programa.

5. La pantalla de la TV se tendrá que dibujar mediante la rutina 450, y entonces habrá que medirla para poder teclear el valor real de la pantalla en mm. en la línea 50. Si su pantalla tiene alrededor de 12 pulgadas entonces los valores de la línea 50 son los correctos, y la ejecución del programa se podrá continuar pulsando la tecla CONT. El centro de la pantalla está en $z=0$, $y=0$ y $z=d=500$.

6. El objeto se puede situar en cualquier parte de la zona determi-

nada por la pirámide que se forma entre la distancia entre el tamaño de la pantalla y la formada por los ojos y la pantalla (d). Una pantalla grande y una distancia de la vista hacia ella demasiado corta, hará que la pirámide diverja rápidamente. Las líneas 290 a la 360 cortan los objetos que van más allá de la pirámide de visión.

7. La condición general que tiene que cumplir un punto para que esté entre los límites establecidos son:

a) $-z * w/d \leq x \leq z * w/d$
(donde w es la mitad de la anchura

```

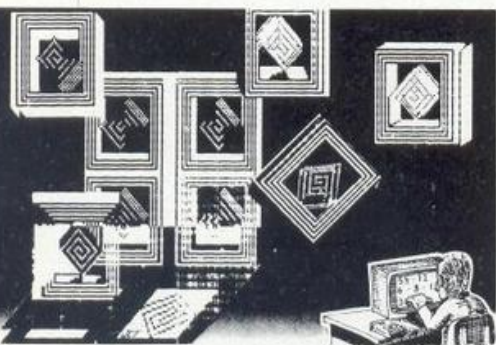
500 RETURN
510 FOR z=d TO d+600 STEP 100
520 FOR x=-90 TO 90 STEP 22.5
530 LET xp=x*d/z
540 LET yp=-62*d/z
550 LET f=INT ((z-d)/200)
560 PLOT xp*p+128,yp*p+87
570 LET dk=d/(z+70)-d/z
580 DRAW INK f;x*dk*p,-62*dk*p
590 PLOT xp*p+128,yp*p+87
600 LET dxp=20*d/z
610 IF x+22.5>90 THEN LET dxp=
0
620 DRAW INK f;dxp*p,0
630 NEXT x
640 NEXT z
650 RETURN
660 INPUT "Dist. del obj. en mm
;z0=";z0
670 INPUT "'Radio' obj. en mm:
R=";R
680 INPUT "Punto central obj. (m
m);x0=";x0
690 INPUT "y0 = ";y0
700 INPUT "Num. de secciones;sa
=";sa
710 INPUT "Num. lados de cada s
ec.;sb=";sb
720 INPUT "Long. inicial en gra
d.;b0=";b0
730 INPUT "Lat. inicial en grad
.;a0=";a0
740 LET db=360/sb
750 LET da=360/sa
760 FOR a=a0 TO 360+a0 STEP da
770 FOR b=b0 TO 360+b0 STEP db
780 LET x=R*SIN (PI/180*b)+x0

```

```

790 LET y=R*COS (PI/180*b)*SIN
(PI/180*a)+y0
800 LET z=R*COS (PI/180*b)*COS
(PI/180*a)+z0
810 LET dx=R*(SIN (PI/180*(b+db
))-SIN (PI/180*b))
820 LET dy=R*SIN (PI/180*a)*(CO
S (PI/180*(b+db))-COS (PI/180*b
))
830 LET dz=R*COS (PI/180*a)*(CO
S (PI/180*(b+db))-COS (PI/180*b
))
840 IF x>z/5.6 THEN LET x=z/5.
6
850 IF (x+dx)>z/5.6 THEN LET d
x=z/5.6-x
860 IF y>z/8.3 THEN LET y=z/8.
3
870 IF (y+dy)>z/8.3 THEN LET d
y=z/8.3-y
880 IF x<-z/5.6 THEN LET x=-z/
5.6
890 IF (x+dx)<-z/5.6 THEN LET
dx=-z/5.6-x
900 IF y<-z/9 THEN LET y=-z/9
910 IF (y+dy)<-z/9 THEN LET dy
=-z/9-y
920 LET xp=x*d/z: LET yp=y*d/z
930 PLOT xp*p+128,yp*p+87
940 LET dk=d/(z+dz)-d/z
950 LET dxp=dx*d/z+x*dk: LET dy
p=dy*d/z+y*dk
960 LET e=ABS ((z-d)/200)
970 DRAW INK e;dxp*p,dyp*p
980 NEXT b
990 NEXT a
1000 GO TO 70

```

de la pantalla en milímetros), y
 $b) -z * h/d \leq y \leq z * h/d$
 (donde h es la mitad de la altura de la pantalla en milímetros).

Los valores en el programa (líneas 290 a 360) se corresponden a la pirámide de visión de $d=500$ mm y la ventana gráfica del Spectrum en una TV de 12 pulgadas.

8. La condición puesta en práctica para cortar determinados gráficos es un poco complicada debido a que se trabaja con dx y dy , que son valores que incrementan la x y y respectivamente, utilizados en las instrucciones DRAW. Esta instrucción en el Spectrum no es estándar en lo que se refiere a que sólo determina la dirección y longitud de una línea mientras que el punto de partida viene expresado por la última instrucción PLOT o DRAW. Teniendo que situar antes de cada sentencia DRAW una ins-

trucción PLOT que será la que inicialice el punto de partida.

9. La proyección en perspectiva de un punto $P(x, y, z)$ en la pantalla $P(xp, yp)$ se basa en la relación existente entre triángulos equivalentes, $Odxp$ y $Ozxp$, según se ve en la figura 1.

Después de una simple manipulación:

$$xp = x * d/z \quad (1)$$

$$yp = y * d/z \quad (1)$$

(la proyección xp e yp son inversamente proporcionales a la profundidad de z .)

La perspectiva queda demostrada con la rutina utilizada en el programa para generar el suelo. En la

notación matemática esta representa al plano de $y=-62$ mm como se puede ver en el diagrama de la proyección en perspectiva.

Viendo el programa línea a línea, tendremos lo siguiente:

Líneas 530-540, definen el tamaño del plano, por ejemplo, z (profundidad) entre 500 mm y 1.100 mm y x (anchura) entre -90 mm y 90 mm.

Líneas 550-560, son aplicaciones directas de las ecuaciones (1).

Línea 580, proyecta xp e yp , expresado en pixels (multiplicado por p) y fija el origen de PLOT en el centro de la pantalla, (256 por 175 pixels).

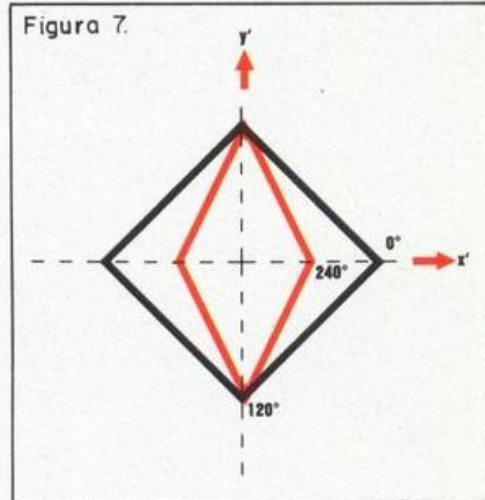
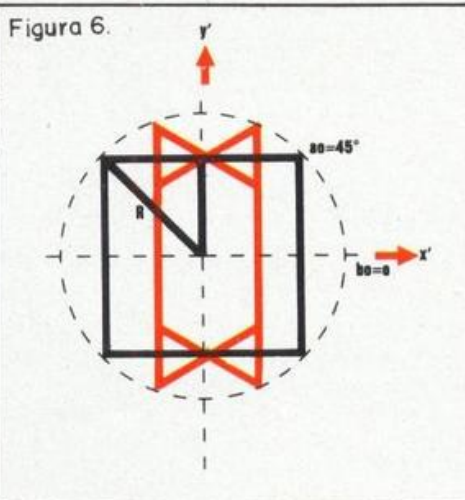


Figura 5.

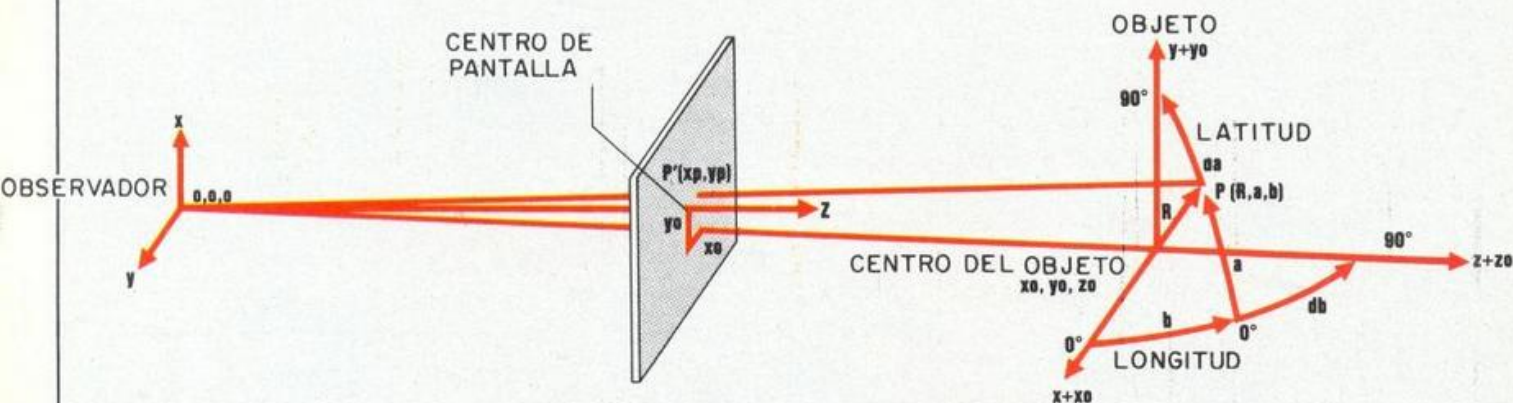


Figura 1.
TRANSFORMACION DE UN PUNTO

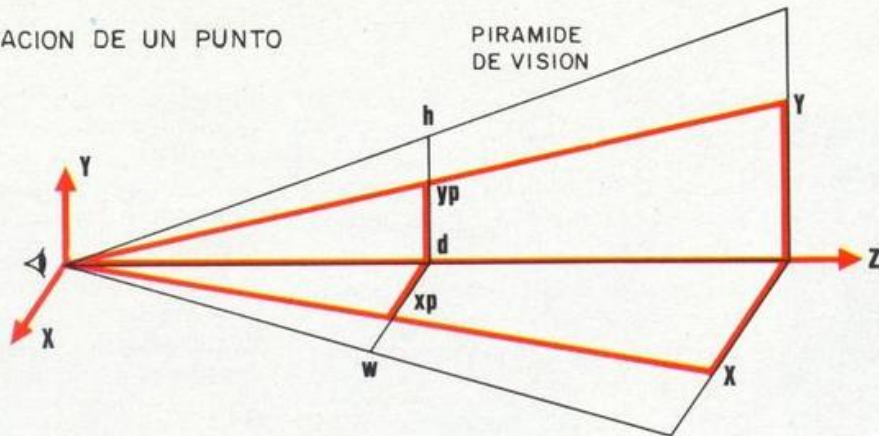


Figura 2.
PROYECCION DE UN VECTOR

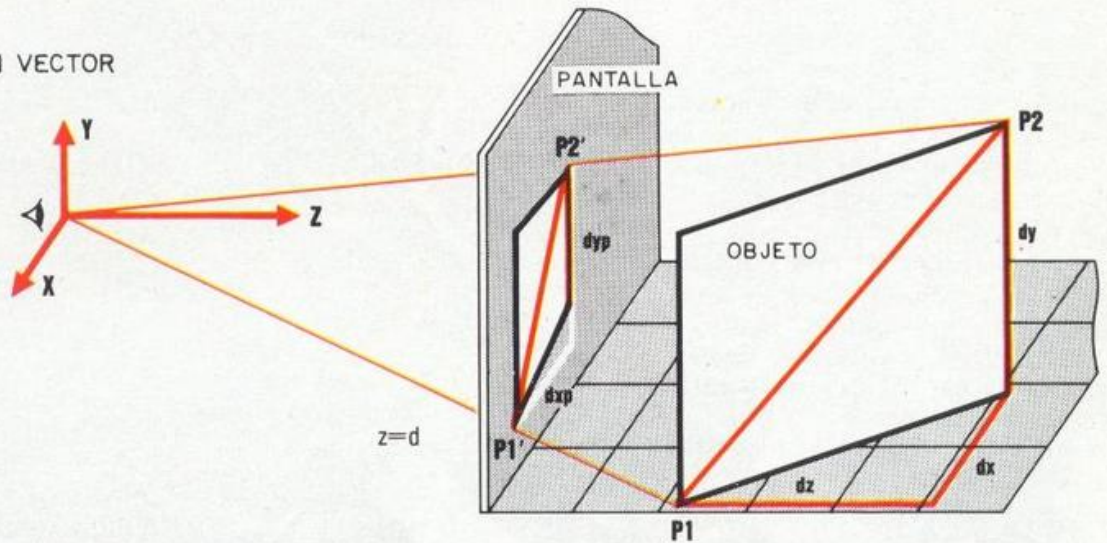


Figura 3.
PROYECCION DE PUNTOS INDIVIDUALES

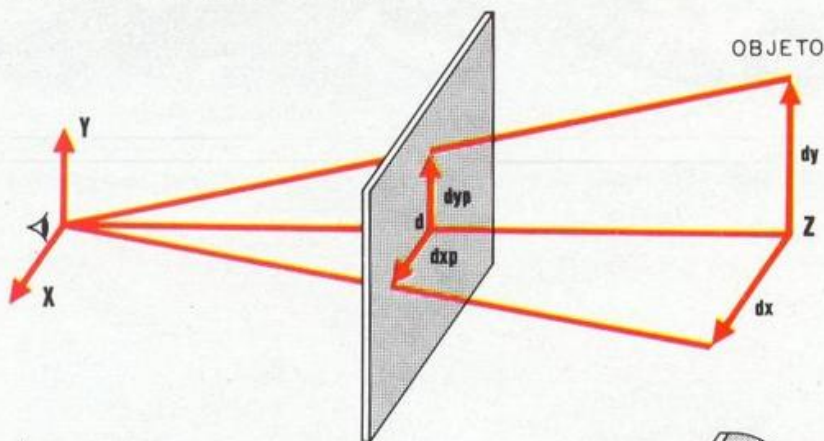
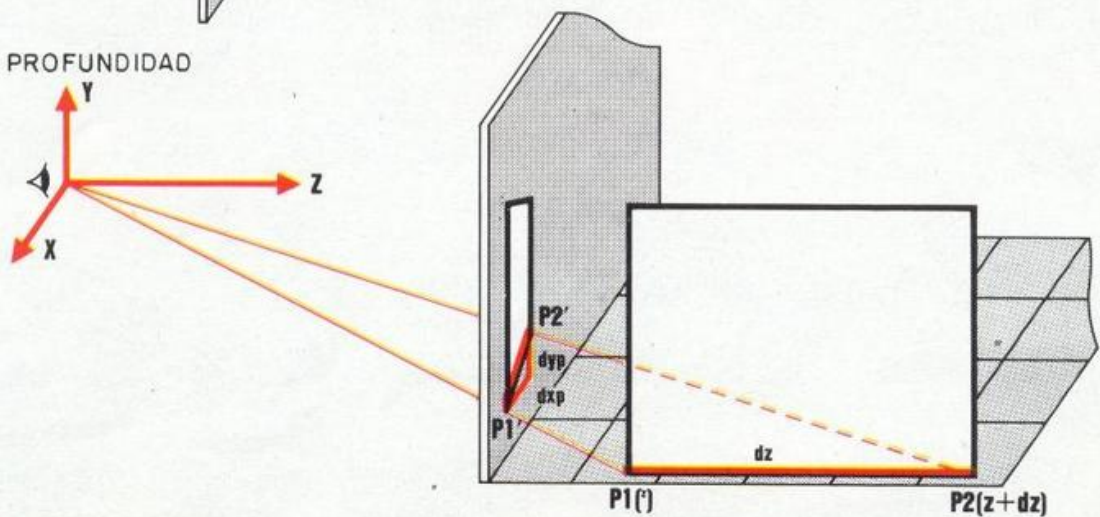
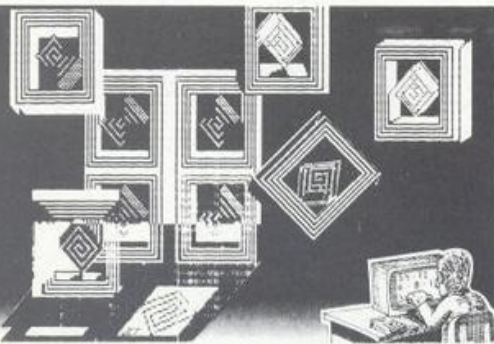


Figura 4.
PROYECCION DE PROFUNDIDAD





Línea 590, define el factor de decrecimiento de perspectiva dk, a la vez que z se va incrementando por dz = 70. Esto es necesario debido a que se tienen que reducir los valores de x e y a la vez que la z aumenta en profundidad, dz.

Línea 600, contiene la primera instrucción DRAW que dibuja una línea correspondiente a la proyección del incremento de dz, después de aplicar el incremento dk.

Línea 610 inicializa el vector DRAW a 580.

Línea 620 convierte el incremento de x (dx=20) a su proyección, dyp.

Línea 640 contiene la segunda instrucción DRAW y dibuja la proyección correspondiente al incremento dx=20.

Las secciones del suelo se obtuvieron a través de la aplicación directa de la ecuación (1) y la sentencia PLOT. La sentencia DRAW se utiliza para dibujar las proyecciones de los incrementos de dz y dy, que son vectores ya que no sólo tienen un valor sino que también poseen una dirección en el espacio. En realidad no sería difícil

proyectar cualquier tipo de marco de cualquier plano. Por ejemplo, el plano x=90 se puede formar sustituyendo la y por x en el siguiente programa:

```
520 REM Plano de la pared derecha, x=90
540 FOR y=-62 TO 62 STEP 12
550 LET yp=1*d/z
560 LET xp=90*d/z
...
600 DRAW INK f; 90*dk*p, y*dk*p
...
620 LET dyp=8*d/z
630 IF y+12 > 62 THEN LET dyp=0
640 DRAW INK f; 0, dyp*p
650 NEXT y
660 NEXT z
```

De la misma forma, el plano z = 1100 se puede dibujar con el programa siguiente:

```
520 REM Pared del fondo
525 LET z=1100
530 FOR y=-62 TO 62 STEP 12
...
560 LET yp=y*d/z
...
590 LET dyp=12*d/z
595 IF y+12 > 62 THEN LET dyp=0
600 DRAW INK f; 0, dyp*p
...
650 NEXT x
660 NEXT y
```

Como alternativa, los tres pla-

nos se pueden combinar hacia la esquina inferior derecha de la habitación por mediación de un programa con los tres ejemplos como subrutinas.

Se podrá visualizar lateralmente una casa con tres pisos, desde el plano original, y = -62, añadiendo las líneas siguientes:

```
517 REM Tres pisos
520 FOR y=-62 TO 62 STEP 50
...
560 LET yp=y*d/z
...
600 DRAW INK f; x*dk*p, y*dk*p
...
665 NEXT y
```

La analogía se puede llevar más lejos, por ejemplo, dibujar un plano orientado hacia cualquier parte y en combinación con varios planos, se podrían generar cajas en el espacio. Es obvio que el programa podría resultar demasiado complejo.

Basado en la experiencia hasta el momento, la rutina que dibuja el plano se recomienda en los siguientes casos:

1. Pared o suelo único.
2. Paredes o suelos paralelos múltiples.
3. Planos paralelos simples o múltiples planos paralelos de los que exponemos un ejemplo a continuación:

```
520 REM techo y=+40
530 FOR z=d TO d+600 STEP 100
```

PARAMETROS PARA EL PROGRAMA DE 3D

Lámpara	Ventana en la pared					Esquina entre...					Puerta	Decoración	
	Trasera	Derecha	Izquierda	La Pared izda. y trasera	La Pared dcha. y trasera	Pared trasera y techo	Pared izda. y techo	Pared dcha. y techo				Techo	Pared
z0	900	900	1100	800	600	1100	1100	1100	800	800	1000	900	700
R	15	3.5	25	25	30	62	62	90	300	300	50	90	40
x0	0	0	0	90	-90	-90	90	0	-90	90	-90	0	90
y0	40	58.5	0	0	0	0	0	62	62	62	-27*	62	0
sb	3	1	1	1	1	1	1	1	1	1	1	10	1
sa	8	2	4	4	4	2	2	2	2	2	4	2	4
b0	45	0	0	90	90	0	0	0	90	90	90	0	90
a0	0	90	45	45	45	90	90	0	0	0	45	0	0

* (-67+45)


```

540 FOR x = - 90 TO 90 STEP
22.5
555 LET y=40+10 * SIN (x / 10)
...
600 DRAW INK f; x * dk * p, y *
dl * p
...
625 LET dy = 10 * SIN ((x + 20) /
10) - SIN (x / 10))
628 LET dyp = dy * d/z
...
635 IF y + dy > 62 THEN LET
dyp=0
...
640 DRAW INK f; dyp * p, dyp *
p

```

Ahora veremos las coordenadas esféricas elegidas para crear una diversa variedad de objetos simétricos en 3D y también para ejecutar transformaciones simples (escalas, rotación y traslación). Las rutinas se utilizan en el programa principal y el sistema de coordenadas se puede observar en la Figura 2.

La generación de los objetos tiene lugar a partir de la línea 90 con una serie de instrucciones INPUTs, con los parámetros que dé el observador. Los parámetros que se piden en el programa son:

- el número de secciones verticales, sb
- el número de lados en cada sección, sa
- la longitud de comienzo, esto es, la longitud de la primera sección vertical, b0
- la latitud de comienzo, es de-

cir, la latitud del origen del vector DRAW que representa el primer lado del polígono, a0.

Esto se puede comprobar en la figura 3, donde sb=3, sa=4, b0=0 y a0=45 (los valores de a0 y b0 vienen en grados).

La figura 3 generará 3 secciones verticales (a0, 120 y 240 grados) cada sección se representa mediante una figura cuadrada orientada de manera que tenga dos lados horizontales y dos verticales. Si hubiera que hacer a igual a 0, se producirán cuadrados con lados dispuestos a 45 grados con la horizontal (figura 4).

El tamaño del objeto obviamente viene determinado por R (en la línea 100) y el tamaño aparente dependerá de la distancia del objeto desde el punto de observación, z0 (línea 90). Variar el valor de z0, equivale a la función de "zoom" en una cámara de fotos — traslación en profundidad. La traslación en el eje x y en el y se consiguen variando x0 e y0, respectivamente (el centro de la pantalla se encuentra en x0=0 e y0=0).

A continuación comentaremos la función de las líneas del programa dedicando especial atención a la parte que hace referencia a las coordenadas esféricas.

Las líneas 170-180 convierten un número elegido, sb y sa, en incrementos angulares en grados, db y da, utilizados en los bucles FOR-NEXT de las líneas 190-200.

Las líneas 210-270, convierten la coordenada esférica R, a y b en coordenadas rectangulares x e y (y sus incrementos dx, dy y dz), de

forma que el objeto se pueda dibujar en perspectiva de la misma manera que las rutinas que dibujan el plano.

Las líneas 220-240, son ecuaciones estándar que convierten coordenadas esféricas en coordenadas rectangulares y se derivan de la figura 2 utilizando trigonometría. Observe que todos los ángulos en las sentencias SIN y COS se han convertido a radianes ($\text{PI}/180 * \text{ángulo en grados}$).

Las líneas 250-270, calculan el incremento cartesiano (vectores) dx, dy y dz, correspondiente al incremento de latitud, da. (Esto es un buen ejemplo de cómo un ordenador personal puede hacer cálculos de alto nivel, aunque no sea tan rápido como las máquinas dedicadas a hacer CAD). Observe que todos los resultados se expresan en milímetros.

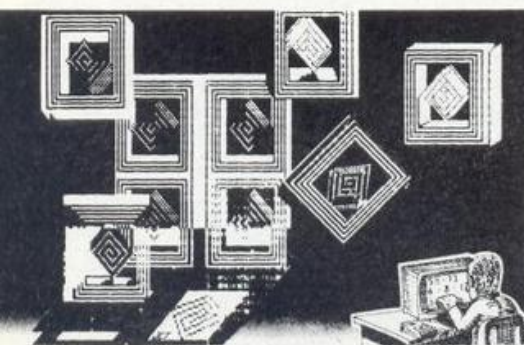
Las líneas 280-360, evitan que la figura se corte fuera de la pirámide de visión y los límites se corresponden a d=500 mm y el tamaño de la «ventana» del Spectrum medido en una televisión con una pantalla de 12 pulgadas.

La línea 370 es la ecuación estándar de la perspectiva (ecuación 1) utilizada en la consiguiente sentencia PLOT de la línea 380. Observe que los milímetros se convierten en pixels en la línea 380.

La expresión en la instrucción 380 ya se mencionó en la rutina que dibuja el suelo; dk representa el decremento del factor de la escala (d/z) cuando varía la profundidad por dz. La línea 390 es algo más complicada de lo visto hasta el momento.

BANCO										MESA REDONDA				
Parte superior					Patás				Estanterías			Alfombra/sombra	Mesa plana	
z0	700	700	500	900	500	900	600	600	600	600	600	600	600	600
R	200	200	10	10	8	8	37	31	8	8	8	40	30	12
x0	-90	-70	-80	-80	-80	-80	90	80	85	85	85	0	0	0
y0	-45	-45	-45	-45	-53	-53	-40	-40	-40	-20	-50	-62	-45	-53.5
sa	1	1	1	1	1	1	1	1	1	1	1	1	1	1
sb	2	2	2	2	4	4	4	4	4	4	4	11	15	4
b0	90	90	0	0	90	90	90	90	90	90	90	0	0	45
a0	0	0	0	0	45	45	45	45	0	0	0	0	0	45
VERTICAL (V)					HORIZONTAL (H)									
														(H)

Algunos ejemplos de las figuras que se pueden crear con este programa, y que pueden modificarse con una pequeña traslación ($R = R_{(y)}$).



Vamos a detenernos un poco para comentar la importancia de dos elementos diferentes que combinados forman la variable dyp:

— $dx * d/z$ es el componente de la perspectiva debido a un incremento en el vector, dx.

— $x * dk$ es el componente de la perspectiva debido al vector, dz

De la misma forma se pueden explicar los componentes de la variable dyp.

La línea 400 dibuja la proyección en perspectiva de un vector en función de dx, dy y dz, medido en pixels. Las líneas 410 y 420 completan el bucle FOR-NEXT que calcula la latitud y longitud. La línea 430 es una bifurcación incondicional que manda el programa a la rutina principal y pide los datos del siguiente objeto a dibujar.

Seguido a la explicación de cómo son generados los objetos, explicaremos algunos parámetros y sus efectos:

1. Selecciona el tipo de figura utilizando la variable sa.

- sa=1; dibuja un punto.
- sa=2; dibuja una línea.
- sa=3; dibuja un triángulo.
- sa=4; dibuja un cuadrado.
- sa=5; dibuja un pentágono.

Cuando el valor de "sa" sea superior a 15, se encontrará dibujando un círculo en el espacio.

2. Determina la orientación de la figura con la latitud inicial, a0, en grados.

a) a0=0; se dibuja el comienzo y el final de la figura en el ecuador.

b) a0=45; se dibuja el comienzo y el final de la figura en la latitud de 45 grados.

3. Selecciona el número de figuras verticales (secciones), sb, comenzando en la longitud b0, donde;

a) b0=0; dibuja el primer y último plano de la figura en el plano z=0, esto es, paralelo a la pantalla.

b) b0=90; sitúa la figura en el plano de x=x0, esto es, en la esquina de la pantalla.

Esto permite dibujar figuras desde un haz de planos verticales — donde la intersección de los planos se sitúa en las coordenadas, x0 y z0.

El programa se puede modificar con facilidad para que dibuje una serie de haces de planos con una intersección horizontal cambiando la longitud por la latitud y viceversa. Se han incluido unos ejemplos de parámetros que se piden en las sentencias INPUT que puede ser efectivo en combinación con la traslación y rotación.

Recuerde que todas las dimensiones están en milímetros y se dibujarán con exactitud en perspectiva dentro de las coordenadas de la pirámide de visión. Esto significa que el valor de R tendrá que ser menor o igual que 62 para aquellos objetos que se encuentren en z0=500.

Observe que seleccionar z0=500

hará que la parte frontal de los objetos se encuentren en frente de la pantalla y que la parte superior e inferior del objeto tienda a cortarse.

La traslación (a lo largo del eje x, por ejemplo) se puede demostrar con efectividad añadiendo un bucle para z0 en el programa de la manera siguiente:

90 REM Ejemplo de ZOOM

```

...
185 LET z0=500
...
440 LET z0= z0 + 100; IF z0 >
1100 THEN GO TO 90
443 GO TO 190

```

Lo mismo se puede hacer para la traslación por los ejes x e y.

Para combinar la rotación alrededor del eje vertical (longitud) con la traslación (en el eje x) se puede proceder de la forma siguiente:

```

110 REM Combinación de trasla-
cion/rotacion
...
185 LET x0=-45
188 LET b0= b0 + 10: REM Rota-
cion de 10 grados
...
440 LET x0= x0 + 22.5: IF x0 >
45 THEN GO TO 90
443 GO TO 188

```

Transformación de perspectivas: proyecciones de los puntos

$$xp / d = x / z \quad \dots\dots\dots (1a)$$

$$xp = x * d / z \quad \dots\dots\dots (1a)$$

$$x / z \leq w / d \quad \dots\dots\dots (2a)$$

$$x \leq z * w / d \quad \dots\dots\dots (2a)$$

BANCO												MESA REDONDA		
Parte superior				Patas				Esteras			Alfombra/sombra	Mesa plana		
z0	700	700	500	900	500	900	600	600	600	600	600	600	600	600
R	200	200	10	10	8	8	37	31	8	8	8	40	30	12
x0	-90	-70	-80	-80	-80	-80	90	80	85	85	85	0	0	0
y0	-45	-45	-45	-45	-53	-53	-40	-40	-40	-20	-50	-62	-45	-53.5
sa	1	1	1	1	1	1	1	1	1	1	1	1	1	1
sb	2	2	2	2	4	4	4	4	4	4	4	11	15	4
b0	90	90	0	0	90	90	90	90	90	90	90	0	0	45
a0	0	0	0	0	45	45	45	45	0	0	0	0	0	45
VERTICAL (V)												HORIZONTAL (H)		

Algunos ejemplos de las figuras que se pueden crear con este programa, y que pueden modificarse con una pequeña traslación ($R = R(y)$). Vale la pena intentarlo.

$$\begin{aligned} y_p / d &= y / z \\ y_p &= y * d / z \dots\dots\dots (1a) \\ y / z &\leq h / d \\ y &\leq z * h / d \dots\dots\dots (2a) \end{aligned}$$

La transformación de los puntos (figura 1) se deriva de triángulos equivalentes, Odx_p y Oz_x , para x ; esto también se puede comprobar para y . Esta transformación se utiliza para "plotear" puntos con las sentencias PLOT. La pirámide de observación determina el espacio en 3D en la parte anterior y posterior de la pantalla que se puede utilizar sin problemas de cortar el objeto (ecuación 2a).

Proyección de vectores

$$\begin{aligned} dx_p &= dx * d / z + x * dk \\ dyp &= dy * d / z + y * dk \dots\dots\dots (3a) \end{aligned}$$

La combinación total de la proyección de vectores se puede observar en la figura 5.

La transformación de vectores se

utiliza con la sentencia DRAW del Spectrum para poder dibujar las proyecciones de los incrementos (vectores) dx , dy y dz .

1. La proyección de dx y dy se derivan de la misma manera que con la proyección de puntos aislados (ver figura 6).

$$\begin{aligned} dx_p &= dx * d / z \\ dyp &= dy * d / z \dots\dots\dots (4a) \end{aligned}$$

2. La proyección de dz se deriva vía dk . dk se define como el decremento del factor de la perspectiva ($k = d / z$) debido a sólo un incremento de z por dz .

$$dk = (d / (z + dz)) - d / z \dots\dots\dots (5a)$$

Refiriéndonos a la figura 7 y aplicando la transformación estándar del punto aislado (ver ecuación 1a) para los puntos $P1(x1, y1, z)$ y $P2(x1, y1, z + dz)$, podremos deducir lo siguiente:

$$\begin{aligned} x_{p1} &= x1 * d / z, x_{p2} = x1 * d / (z + dz) \\ y_{p1} &= y1 * d / z, y_{p2} = y1 * d / (z + dz) \dots\dots\dots (6a) \end{aligned}$$

$$\begin{aligned} \text{Como, } dx_p &= x_{p2} - x_{p1} \\ &= x1 * d / (z + dz) - x1 * d / z \\ &= x1 * (d / (z + dz) - d / z) \\ &= x1 * dk \end{aligned}$$

$$\begin{aligned} \text{por ejemplo, } dx_p &(\text{en función de } dz \text{ solamente}) = x * dk \\ dyp &(\text{en función de } dz \text{ solamente}) = y * dk \dots\dots\dots (7a) \end{aligned}$$

Los efectos combinados de los tres incrementos, dx , dy y dz , es como consecuencia de ello la suma de todos los componentes de la proyección (ver ecuaciones 3a y la figura 5). Las ecuaciones se exponen de nuevo por conveniencia:

El total de:

$$\begin{aligned} dx_p &= dx * d / z + x * dk \\ dyp &= dy * d / z + y * dk \end{aligned}$$

debido a dx o a dy debido a dz solamente

SINCLAIR QL



APLICACIONES

- BASE DE DATOS
- TRAT TEXTOS
- HOJA DE CALCULO
- GRAFICOS
- LENGUAJES
- PASCAL
- FORTH
- ENSAMBLADOR

¡CONÓCELOS!
¡APRENDE A USARLOS!

CURSILLOS DE FORMACION
PARA DISEÑO DE PROGRAMAS

VISÍTANOS



World-Micro S.A.

Avda. del Mediterráneo, 7
Tels. 251 12 00 y 251 12 09 - MADRID 7

SI TE INTERESA LA ELECTRONICA

Resuelve los problemas de:

- Diseño de filtros
- Análisis de redes
- Comunicaciones por ionosfera y por satélite
- Antenas y acopladores
- Lineas de transmision
- Bobinas y transformadores
- Interferencias etc.

Con mas de 100 PROGRAMAS TECNICOS, perfectamente documentados en castellano, con ejemplos practicos.

Operativos en ordenadores:
HEWLETT PACKARD serie 200
ZX Spectrum 48K
Consultar para otros

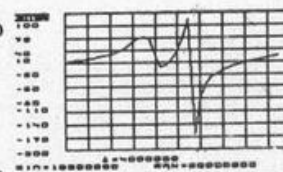
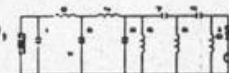
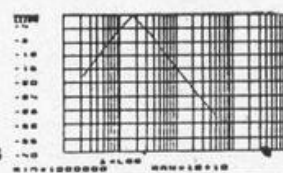
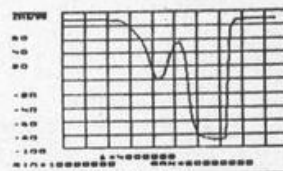
Asistencia técnica post venta

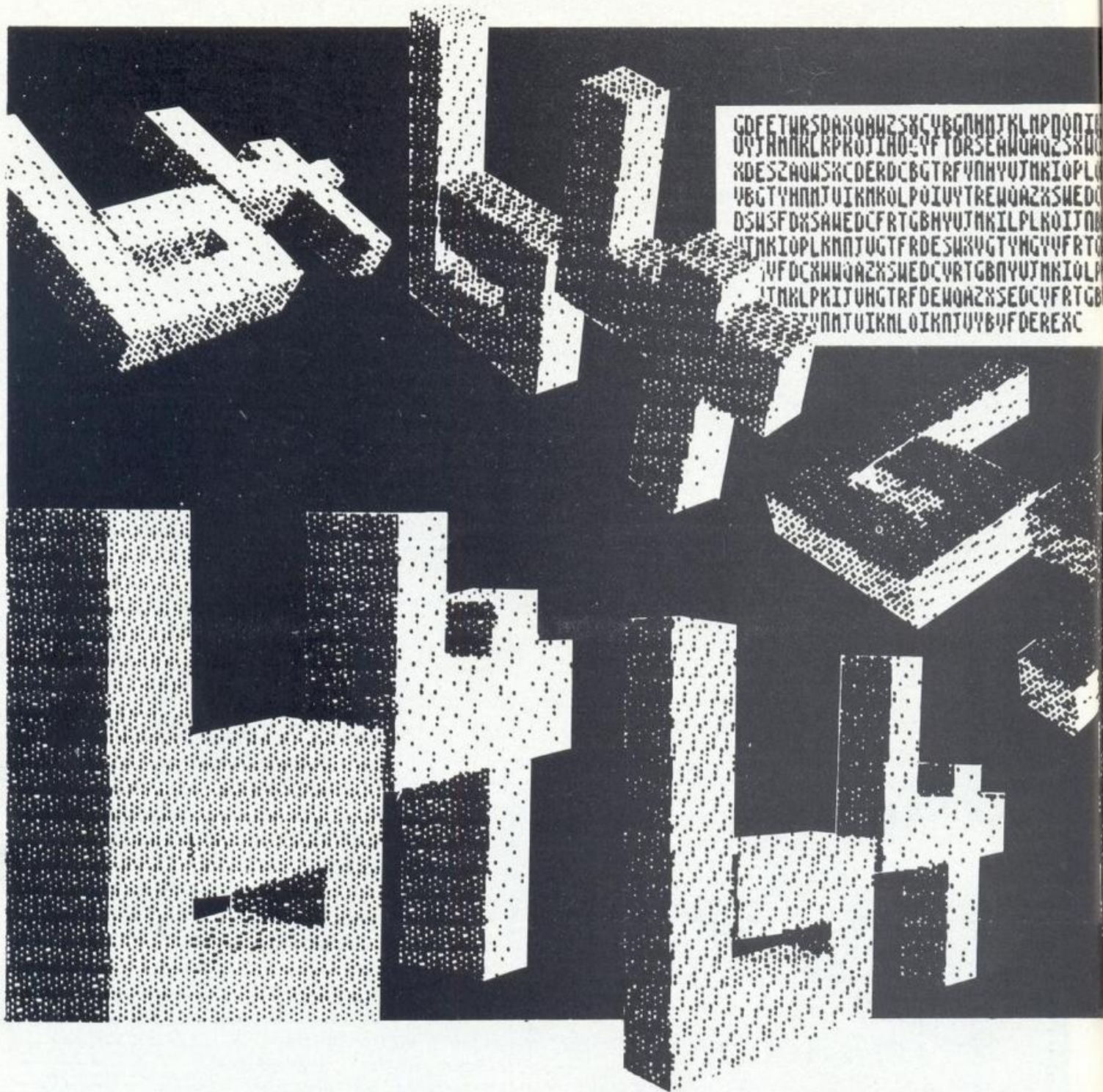
Pide información a:

SOFTRONICA S.A.®

C/José Abascal, 52
MADRID 28003

Tel. (91) 441 38 46
(91) 450 18 24





64 caracteres por línea

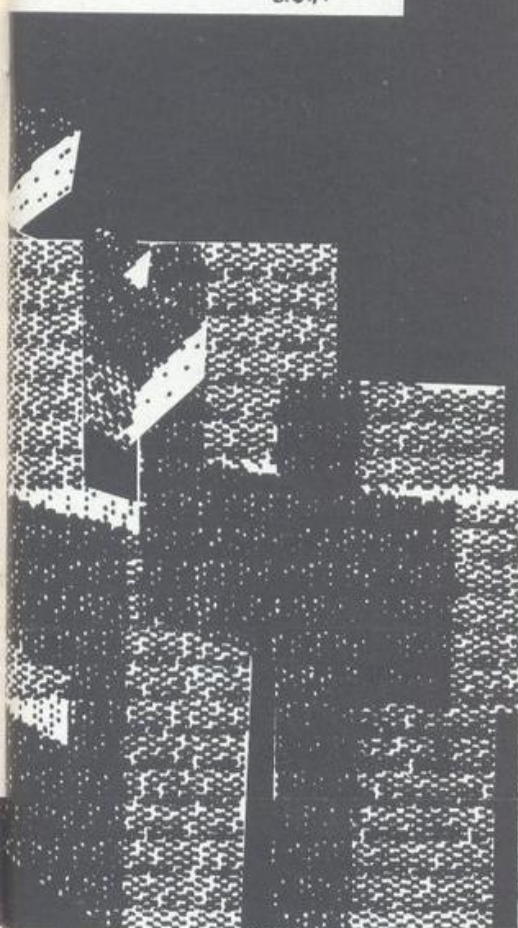
En el número 1 de **TODOSPECTRUM**, y con el título "64 caracteres", ofrecíamos un pequeño editor de textos cuya principal característica era trabajar con 64 caracteres por línea. Si le agradó la idea

y la resolución de su pantalla le permite trabajar cómodamente con 64 caracteres, le ofrecemos ahora la rutina que lo hace posible y que podrá incorporar a sus programas.

Una aproximación al problema nos da las dos soluciones posibles: cambiar completamente la circuitería de video, o redefinir el ancho de los caracteres a la mitad del estándar. La primera es prácticamente inalcanzable para el aficionado medio, aunque actualmente existe una firma comercial inglesa desarrollando una tarjeta de 80 columnas en un sistema de expansión para el Spectrum. La se-

TAGDBENKASQZAWXSCDUEFGHTUY
 ABGHIJKLMNOPQRSTUVWXYZ
 UMBVGYTFCKDRESZAQWSEDF
 GBNHYUTNKIOLPQITPBHUYGVCFA
 REDCKSWQAZXSWEDCVFRTGBNHY
 UTKILOPLKNNHYVXXKFCDEKSAF
 KOTPBHUYGCFTRDXZAWSESDXCFY
 KIOPLKNTIUMBGYTFCDREZAWX

JCT.



la separación entre caracteres, por lo que sólo nos restan 3 pixels para definir cada caracter. Es posible que los caracteres no estén perfectamente formados, pero son suficientemente legibles. Una nota sobre este tema: debido a que la resolución no se cambia, el ancho de banda de la señal de video no es incrementado como se podría pensar, por lo que el televisor normal

nos servirá como pantalla en este modo de visualización al igual que con el formato de 32 columnas. Sin embargo si se usa un monitor con entrada de video en lugar del televisor, se conseguirá un incremento notable en legibilidad y definición en cualquier tipo de pantalla (texto, dibujos, etc.) generada por el ordenador. El problema es que el Spectrum no dispone de sa-

7530		10	ORG 30000
5091		15 PFLAG	EQU 23697
7530	00	20 XPOS	DEFB 0
7531	AF	30 YPOS	DEFB 175
7532	00	35 X	DEFB 0
7533	00	36 Y	DEFB 0
7534	01	70 CAR	DEFB 1
7535	0000	75 DIRCAR	DEFW 0
7537	ED4B3075	80 PRINT	LD BC, (XPOS)
753B	78	83	LD A, B
753C	D608	84	SUB 8
753E	47	85	LD B, A
753F	ED433275	86	LD (X), BC
7543	21C076	140	LD HL, CARSET
7546	3A3475	145	LD A, (CAR)
7549	5F	150	LD E, A
754A	1D	160	DEC E
754B	1600	170	LD D, 0
754D	CB23	180	SLA E
754F	3001	190	JR NC, L1
7551	14	200	INC D
7552	CB23	210 L1	SLA E
7554	3001	220	JR NC, L2
7556	14	230	INC D
7557	CB23	240 L2	SLA E
7559	3001	250	JR NC, L3
755B	14	260	INC D
755C	19	270 L3	ADD HL, DE
755D	223575	280	LD (DIRCAR), HL
7560	0608	290	LD B, 8
7562	0E00	295	LD C, 0
7564	C5	300 LAZ01	PUSH BC
7565	2A3575	310	LD HL, (DIRCAR)
7568	48	320	LD C, B
7569	0600	330	LD B, 0
756B	0B	340	DEC BC
756C	09	350	ADD HL, BC
756D	ED4B3275	360	LD BC, (X)
7571	7E	370	LD A, (HL)
7572	CB7F	380	BIT 7, A
7574	2005	390	JR NZ, P1
7576	CDF675	400	CALL BORRA

gunda solución no nos da un juego de caracteres perfectamente formado, o el formato más deseable de 80 columnas, pero es barata y sencilla de poner en funcionamiento.

La solución final fue una rutina en código máquina que imprime texto a razón de 64 caracteres por línea. Ya que no es posible cambiar la resolución estándar de 256 pixels (elementos de pantalla) horizontales, tenemos que dividirlos en grupos de 4 para conseguir las 64 columnas deseadas. Estos 4 elementos de pantalla deben incluir

lida de video estándar. Únicamente disponemos de video directo en el bus de conexiones de la parte trasera del ordenador. Esta señal no es estándar en su amplitud y tampoco lo es la impedancia de salida del circuito generador de video. La solución es intercalar un *buffer* o amplificador entre la máquina y el monitor. Este puede ser un simple transistor NPN en configuración de seguidor de emisor (base directamente a la salida, emisor a masa a través de una resistencia de unos 3 K, y colector directamente a cualquier punto de alimentación) con un ancho de banda suficiente (10-20 Mhz.). Aquellos que hayan manejado un soldador alguna vez no tendrán ningún problema.

Volviendo al problema del *software*, se precisó que sería necesario un programa en lenguaje máquina para realizar la impresión de caracteres a una velocidad razonablemente alta y para poder llamar la rutina desde el sistema operativo del Spectrum. Se escribió una rutina de impresión que imprime caracteres en cualquier punto de la pantalla. Se diseñó un nuevo juego de caracteres, pero algunos fueron prácticamente imposibles de definir debido al ancho permitido. Afortunadamente estos caracteres son irrelevantes, como el símbolo de "copyright". Por lo que se sustituyen por caracteres del castellano no presentes en el juego estándar del Spectrum.

Se programó una segunda rutina que aceptase el código de un carácter en el acumulador de la máquina y llamase a la rutina de impresión para imprimirlo en pantalla. Esta subrutina actualiza automáticamente la posición de impresión incrementándola en 4 pixels horizontales u 8 verticales (si se alcanza el final de la línea), cada vez que se imprime un carácter. La rutina no realiza *scrolling* al llegar a la última línea, aunque no se descarta esta posibilidad en un futura ampliación del programa. El código

de carácter de "AT" (CHR\$ 22) hace que la rutina tome los dos siguientes caracteres que reciba en el registro A como dos números que determinan las coordenadas del elemento de pantalla donde se imprimirá el próximo carácter (empezando por la esquina superior

izquierda). El carácter ENTER (CHR\$ 13) es ignorado, y cualquier otro que no esté en el rango 32-127 dará impresiones sin sentido, ya que la rutina tomará como modelo de carácter una zona de memoria donde no existe tal.

Esta rutina fue escrita teniendo

7579	1803	410	JR	C1
757B	CDE475	420 P1	CALL	PINTA
757E	03	430 C1	INC	BC
757F	CB77	440	BIT	6,A
7581	2005	450	JR	NZ,P2
7583	CDF675	460	CALL	BORRA
7586	1803	470	JR	C2
7588	CDE475	480 P2	CALL	PINTA
758B	03	490 C2	INC	BC
758C	CB6F	500	BIT	5,A
758E	2005	510	JR	NZ,P3
7590	CDF675	520	CALL	BORRA
7593	1803	530	JR	C3
7595	CDE475	540 P3	CALL	PINTA
7598	03	550 C3	INC	BC
7599	CB67	560	BIT	4,A
759B	2005	570	JR	NZ,P4
759D	CDF675	580	CALL	BORRA
75A0	1803	590	JR	C4
75A2	CDE475	600 P4	CALL	PINTA
75A5	03	610 C4	INC	BC
75A6	CB5F	620	BIT	3,A
75AB	2005	630	JR	NZ,P5
75AA	CDF675	640	CALL	BORRA
75AD	1803	650	JR	C5
75AF	CDE475	660 P5	CALL	PINTA
75B2	03	670 C5	INC	BC
75B3	CB57	680	BIT	2,A
75B5	2005	690	JR	NZ,P6
75B7	CDF675	700	CALL	BORRA
75BA	1803	710	JR	C6
75BC	CDE475	720 P6	CALL	PINTA
75BF	03	730 C6	INC	BC
75C0	CB4F	740	BIT	1,A
75C2	2005	750	JR	NZ,P7
75C4	CDF675	760	CALL	BORRA
75C7	1803	770	JR	C7
75C9	CDE475	780 P7	CALL	PINTA
75CC	03	790 C7	INC	BC
75CD	CB47	800	BIT	0,A
75CF	2005	810	JR	NZ,P8
75D1	CDF675	820	CALL	BORRA
75D4	1803	830	JR	C8
75D6	CDE475	840 P8	CALL	PINTA

SPECTRUM COMPUTING

*¡Ya está a la venta!
Compela en su quiosco
habitual o solicítala a:
INFODIS S.A.
Bravo Murillo, 377-5, s.a.
28020 MADRID
enviando el cupón que encontrará
en esta revista*

para 16K y 48K

**Como programar
aventuras**

**Graficos
tridimensionales**

Jugando al Poker

El castillo embrujado



**875
pts.**

Misión suicida

**64 columnas
en su pantalla**

**y mucho
mas**

BIENVENIDO A

SPECTRUM

COMPUTING

PRESENTA

- BITMAN
- COMO PROGRAMAR AVENTURAS
- MISSION SUICIDA
- JUGANDO AL POKER
- SCROLLER
- GRAFICOS TRIDIMENSIONALES
- EL CASTILLO ENDIABLADO
- COWBOY
- RANDIT
- SKIING
- 64 COLUMNAS EN PANTALLA

GARANTIA
Calidad de carga
asegurada



en cuenta la posibilidad de ser llamado desde el sistema operativo del Spectrum. La rutina de impresión de un carácter (RST 16) en la ROM del ordenador puede ser forzada a llamar nuestra subrutina introduciendo su dirección de comienzo en 2 bytes del área de información de canales, de forma que cada vez que se ejecute una sentencia PRINT desde el BASIC el mensaje será impreso en pantalla en el formato de 64 columnas. Por desgracia estos dos bytes son renovados cada vez que se realiza un listado de programa, ya sea mediante el comando oportuno o apretando la tecla ENTER. Esto es debido a que la rutina de listado envía a la de impresión los códigos de las palabras claves (*Tokens*) directamente sin decodificador, tarea que ha de realizar esta última. Si utilizásemos este sistema, sería necesario renovar la dirección cada vez que se fuera a imprimir algo. Sin embargo la rutina LPRINT posee la misma facilidad, con la diferencia de que los bytes que contienen la dirección de llamada a la rutina no son cambiados una vez alterados. Esto nos permite usar el comando LPRINT para usar nuestra rutina desde el BASIC. Una subrutina que es llamada en primer lugar, almacena la dirección apropiada en los bytes que contienen la información del canal de impresora (correspondientes al valor contenido en la variable del sistema CHANS + 15). El formato de una instrucción LPRINT para nuestros propósitos será:

LPRINT AT x,y; "(texto a imprimir)"

Donde x e y son las coordenadas del pixel allí donde se quiere comenzar la impresión. Si se omite esto, el texto será impreso después del último carácter enviado a la rutina o en la posición 0,175 si la rutina es llamada por primera vez.

Este sistema no deja a la impresora ZX totalmente fuera de uso, ya que puede ser usada restauran-

do los valores originales en CHANS +15 y CHANS +16. Estos son los bytes 244 y 9. Es decir, cada vez que queramos usar la impresora bastará con "pokear" en CHANS +15 el valor 244, y en CHANS +16 el valor 9. De esta forma el canal LPRINT queda direccionado a la rutina existente en

la ROM. Para volver a usar nuestra rutina habrá que volver a llamar la subrutina de inicialización explicada anteriormente. También es posible imprimir con la impresora a 64 columnas efectuando el mismo proceso pero manteniendo el texto a imprimir en pantalla, de forma que podamos realizar un

75D9	3A3375	850	CB	LD	A, (Y)
75DC	3C	860		INC	A
75DD	323375	870		LD	(Y), A
75E0	C1	880		POP	BC
75E1	10B1	890		DJNZ	LAZ01
75E3	C9	900		RET	
75E4	F5	910	PINTA	PUSH	AF
75E5	C5	915		PUSH	BC
75E6	3A915C	920		LD	A, (PFLAG)
75E9	CB87	925		RES	0, A
75EB	CB97	930		RES	2, A
75ED	32915C	935	PLOT	LD	(PFLAG), A
75F0	CDDF22	940		CALL	#22DF
75F3	C1	945		POP	BC
75F4	F1	950		POP	AF
75F5	C9	955		RET	
75F6	F5	960	BORRA	PUSH	AF
75F7	C5	965		PUSH	BC
75F8	3A915C	970		LD	A, (PFLAG)
75FB	CB87	975		RES	0, A
75FD	CB07	980		SET	2, A
75FF	C3ED75	985		JP	PLOT
7602	00	1110	AT	DEFB	0
7603	00	1120	ACENTO	DEFB	0
7604	FE16	1130	USER	CP	22
7606	2006	1140		JR	NZ, E1
7608	210276	1150		LD	HL, AT
760B	3601	1160		LD	(HL), 1
760D	C9	1170		RET	
760E	47	1180	E1	LD	B, A
760F	210276	1190		LD	HL, AT
7612	7E	1200		LD	A, (HL)
7613	FE01	1210		CP	1
7615	2007	1220		JR	NZ, E2
7617	3602	1230		LD	(HL), 2
7619	213075	1240		LD	HL, XPOS
761C	70	1250		LD	(HL), B
761D	C9	1260		RET	
761E	FE02	1270	E2	CP	2
7620	2007	1280		JR	NZ, E3
7622	3600	1290		LD	(HL), 0
7624	213175	1300		LD	HL, YPOS
7627	70	1310		LD	(HL), B

"COPY" una vez direccionada la impresora.

El nuevo juego de caracteres definido admite las letras del castellano "ñ", "ü", y "í"; así como acentos en todas las vocales. Los cuatro símbolos sustituyen a los caracteres *hash* (código 35), *ampersand* (código 38), *at* (código

64), y barra vertical (código 124); de forma que podemos obtener estos nuevos caracteres incluyendo los correspondientes que sustituyen en la sentencia LPRINT que ejecutemos. El acento se obtiene tecleando el "apóstrofo" (código 39) antes de la vocal deseada. Una vez ejecutada la sentencia aparece-

rá la vocal correspondiente acentuada. Un ejemplo de esto puede ser esta sentencia:

LPRINT "El árbol"

Que dará como resultado:

El árbol

en la pantalla del ordenador.

El programa completo ocupa alrededor de 1K de memoria. El listado se ha dado en ensamblador, de forma que aquellos que dispongan de esta herramienta puedan introducir la rutina cómodamente. Para los más modestos se da el ensamblado en hexadecimal en el lateral izquierdo del propio listado. Este ensamblado ha sido realizado en la dirección de memoria 30000, por lo que al cargarse el código habrá de hacerse estrictamente a partir de esta dirección, ya que la rutina no es relocizable. Un sencillo cargador hexadecimal bastará para la tarea. Se adjunta también una tabla de números en decimal que constituyen el nuevo juego de caracteres. Esta tabla deberá introducirse en memoria a partir de la etiqueta del listado, del ensamblador "CARSET". Se puede introducir en el propio ensamblador con un pseudonemónico del tipo DEFB (*Define Byte*), o bien directamente en memoria a partir de la dirección indicada por la etiqueta susodicha. Los que utilicen ensamblador podrán, por supuesto, ensamblar el código en la dirección de memoria que deseen, cuidando siempre de que esté por encima del RAMTOP. Para los que introduzcan el ensamblado directamente en la dirección 30000, CARSET se halla en la 30400. CHANEL es la subrutina que hay que llamar para inicializar el canal de LPRINT, y se halla en la dirección 30386. Es decir, cada vez que se cargue el programa hay que ejecutar la sentencia:

RANDOMIZE USR 30386

Y la rutina queda preparada para usarse desde BASIC. USER es

7628	C9	1320	RET
7629	7B	1330 E3	LD A,B
762A	FE0D	1340	CP 13
762C	2001	1350	JR NZ,E4
762E	C9	1360	RET
762F	FE27	1370 E4	CP 39
7631	2006	1380	JR NZ,E5
7633	210376	1390	LD HL,ACENTO
7636	3601	1400	LD (HL),1
7638	C9	1410	RET
7639	47	1420 E5	LD B,A
763A	210376	1430	LD HL,ACENTO
763D	7E	1440	LD A,(HL)
763E	FE01	1450	CP 1
7640	2026	1460	JR NZ,E6
7642	7B	1470	LD A,B
7643	C6C0	1480	ADD A,192
7645	3003	1490	JR NC,E7
7647	7B	1500	LD A,B
7648	1802	1510	JR E8
764A	D6E0	1520 E7	SUB 224
764C	5F	1530 E8	LD E,A
764D	1600	1540	LD D,0
764F	CB23	1550	SLA E
7651	3001	1560	JR NC,E9
7653	14	1570	INC D
7654	CB23	1580 E9	SLA E
7656	3001	1590	JR NC,E10
7658	14	1600	INC D
7659	CB23	1610 E10	SLA E
765B	3001	1620	JR NC,E11
765D	14	1630	INC D
765E	21C076	1640 E11	LD HL,CARSET
7661	19	1650	ADD HL,DE
7662	E5	1660	PUSH HL
7663	3620	1670	LD (HL),32
7665	23	1680	INC HL
7666	3640	1690	LD (HL),64
7668	7B	1700 E6	LD A,B
7669	C6C0	1710	ADD A,192
766B	3004	1720	JR NC,E12
766D	7B	1730	LD A,B
766E	3C	1740	INC A

la rutina que recibe el código del carácter en el acumulador, y que es llamada por el sistema operativo. Este se halla en la 30212. Aquellos que lo deseen pueden llamar la rutina desde otro programa en código máquina haciendo un CALL 30212 (o la dirección pertinente si se ensambla en otra zona), habiendo cargado previamente el registro A con el código del carácter a imprimir.

El funcionamiento del programa a grandes rasgos es como sigue: la rutina USER recibe en el acumulador el código del carácter a imprimir. En primer lugar lo compara con el código AT (22). Si es éste, retorna el control al programa de llamada, fijando previamente una "bandera" o *flag* que le indica que el próximo código es una coordenada de impresión y no un carácter. Posteriormente compara con el código 13 (ENTER), realizando un simple retorno si se trata de éste. Por último compara con 29 (apóstrofo). Si se ha enviado un apóstrofo el programa entiende que se quiere acentuar la próxima letra, y por lo tanto fija una "bandera" de control (ACENTO) que se lo indicará en la próxima llamada. Al recibir el siguiente carácter, modifica el modelo de éste (hallando previamente su dirección en el juego de caracteres) de forma que incluya el acento. Por último llama a la rutina de impresión de un carácter (PRINT) pasándole los parámetros de código del carácter y posición de impresión. Si la última letra fue acentuada, remodifica el modelo de su carácter. Antes de devolver el control al programa de llamada actualiza la posición de impresión, incrementándola horizontalmente en 4 pixels o verticalmente en 8, según corresponda. La rutina de impresión al ser llamada, calcula la dirección del carácter a imprimir en el juego de caracteres. Después toma éste como modelo y entra en un bucle (LAZO1) de 8 iteraciones en el que cada byte del carácter es explorado bit por bit,

llamándose a las rutinas BORRA o PINTA según el bit valga 1 o 0. Las rutinas BORRA y PINTA llaman a la rutina PLOT de la ROM con distintos parámetros INVERSE y OVER para que borre o pinte un punto según el caso. Una vez terminado el bucle, el carácter ya está impreso en la pantalla, y por

lo tanto se devuelve el control a USER.

Y esto es todo, una sencilla rutina con la que puede dar un toque profesional a sus programas en BASIC si se usa para imprimir mensajes, o para soportar la salida en código máquina.

Javier Cancela

766F	1802	1750	JR	E13
7671	D6DF	1760	SUB	223
7673	323475	1770	LD	(CAR), A
7676	C5	1780	PUSH	BC
7677	CD3775	1790	CALL	PRINT
767A	3A3075	1800	LD	A, (XPOS)
767D	C604	1810	ADD	A, 4
767F	3805	1820	JR	C, E14
7681	323075	1830	LD	(XPOS), A
7684	180D	1840	JR	E15
7686	3A3175	1850	LD	A, (YPOS)
7689	D608	1860	SUB	8
768B	323175	1870	LD	(YPOS), A
768E	3E00	1880	LD	A, 0
7690	323075	1890	LD	(XPOS), A
7693	3A0376	1900	LD	A, (ACENTO)
7696	C1	1910	POP	BC
7697	FE01	1920	CP	1
7699	2016	1930	JR	NZ, E16
769B	E1	1940	POP	HL
769C	7B	1950	LD	A, B
769D	FE69	1960	CP	105
769F	2004	1970	JR	NZ, E17
76A1	3E40	1980	LD	A, 64
76A3	1802	1990	JR	E18
76A5	3E00	2000	LD	A, 0
76A7	3600	2010	LD	(HL), 0
76A9	23	2020	INC	HL
76AA	77	2030	LD	(HL), A
76AB	00	2040	NOF	
76AC	3E00	2050	LD	A, 0
76AE	320376	2060	LD	(ACENTO), A
76B1	C9	2070	RET	
76B2	2A4F5C	2080	LD	HL, (23631)
76B5	010F00	2090	LD	BC, 15
76B8	09	2100	ADD	HL, BC
76B9	010476	2110	LD	BC, USER
76BC	71	2120	LD	(HL), C
76BD	23	2130	INC	HL
76BE	70	2140	LD	(HL), B
76BF	C9	2150	RET	
76C0	00	2160	DEFB	0
76C0		3000	ENT	CARSET

Spectrum puede con todos.

¿Quién nos gana en gama? Estamos por asegurar que ninguno. No olvides que tenemos un Spectrum para cada exigencia: dos capacidades diferentes (16K y 48K) y tres modelos con dos tipos de teclado (doméstico y profesional).

¿Quién nos gana en programas? Spectrum cuenta con más de 5.000 títulos publicados a nivel internacional, cien de ellos están traducidos al castellano.

Naturalmente estos crecen casi de forma constante. Una buena muestra es el voluminoso catálogo de software que puedes solicitar a tu distribuidor de confianza.

¿Quién nos gana en periféricos? Ya son más de 50 los periféricos creados especialmente para el Spectrum, pero no creas que eso termina ahí. Es muy raro el día que no aparece en el mercado una novedad. Así tu Spectrum guardará para ti el mismo interés del primer día.

¿Quién nos supera en número? Otro factor a tener en cuenta: te diremos que ya son más de tres millones los microordenadores Sinclair vendidos en todo el mundo (y más de 100.000 Spectrum vendidos en España) ¿no te parece esto una buena razón para confiar en tu Spectrum?

Decídate; este año tener un Spectrum es todo un regalo.

Los concesionarios INVESTRONICA tienen para ti un montón de novedades.

* De venta en la Red de Concesionarios INVESTRONICA.

*Feliz
Spectrum
1985*

• Con el Spectrum de 16K te regalamos 6 estupendos juegos.



• Con el Spectrum de 48K te regalamos 8 estupendos juegos.



SINCLAIR RESEARCH LIMITED hace constar que no está en condiciones de garantizar el origen y calidad de aquellos productos que no hayan sido comercializados en España a través de su distribuidor exclusivo INVESTRONICA, S.A.

El juego de

Llamado en general *Life*, aunque también recibe distintos nombres y modalidades, es el programa típico de obligada incorporación en todos los paquetes de demostración, independientemente del ordenador. Este conocido programa nos servirá en esta ocasión para analizar las técnicas de programación BASIC a emplear en el Spectrum.

El juego de la vida no es en realidad un juego sino un proceso evolutivo que el usuario prepara y observa.

Inventado por John Conway, el programa simula una colonia de células, que se desarrollan a partir de una colonia inicial de acuerdo con tres reglas básicas. Las células se sitúan en un tablero compuesto de casillas individuales y cada generación siguiente se determina según el número de casillas adyacentes que contengan células. Las reglas son las siguientes:

1) Si una célula posee, en los 8 casilleros adyacentes, dos o tres células vecinas, sobrevivirá a la siguiente generación.

2) Una célula muere (desaparecerá del tablero en la siguiente generación) si tiene 4 o más células vecinas (por superpoblación); y uno o ningún vecino (por aislamiento).

3) En cada casilla que esté vacía que tenga tres vecinos nacerá una célula en la siguiente generación.

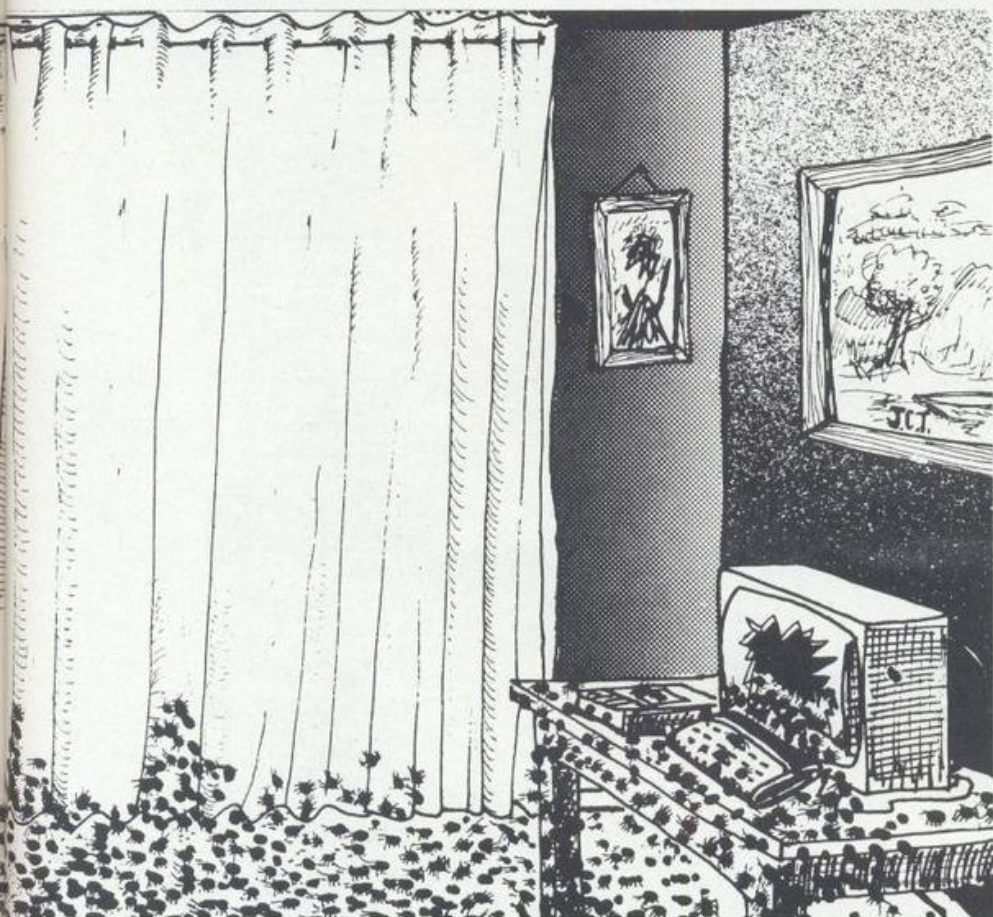
Para implementar esto en el ordenador, es obvio que deberán manejarse tablas bidimensionales debido a que el tablero está compuesto de filas y columnas. También es necesario tener otra tabla extra debido a que el tablero de la generación (n+1) se define de acuerdo con la generación (n), y ninguna célula



```
5 REM "LIFE"
10 PRINT TAB 12: "*LIFE*"
20 PRINT
30 DIM A(16,16)
40 DIM B(16,16)
50 DIM A$(6,6)
60 LET GEN=0
70 REM *COLONIA INICIAL*
80 PRINT "TECLEE LA COLONIA INICIAL QUE SE SITUARA EN LAS CASILLAS CENTRALES DE 6 X 6. TECLEE 6 CADENAS DE 6 CARACTERES, YA SEAN ASTERISCOS (*) O ESPACIOS."
90 PRINT
100 FOR F=1 TO 6
110 PRINT "CADENA ":F;
120 INPUT A$(F)
130 PRINT " ":A$(F)
```

```
140 NEXT F
145 REM *SITUA LA COLONIA EN LA CADENA*
150 FOR F=1 TO 6
160 FOR Z=1 TO 6
170 IF A$(F,Z)="*" THEN LET B(F+5,Z+5)=1
180 NEXT Z
190 NEXT F
200 GO SUB 1000
205 REM *INCREMENTO DE GENERACION*
210 LET GEN=GEN+1
230 FOR X=2 TO 15
240 FOR Y=2 TO 15
250 REM *INICIALIZA CONTADOR*
260 LET C=0
265 REM *COMPRUEBA LAS CELULAS VECINAS*
```


e la vida



de la generación (n) se puede alterar hasta que no se completa el proceso de comprobación.

En la tabla de la generación (n) hay que hacer una comprobación de todos los casilleros, y contabilizar el número de células vecinas. Según las reglas, el futuro de la célula en esa posición está determinado, y si está vacío nacerá una nueva célula. Estos datos se almacenan en una cadena, y seguidamente la cadena de la generación se actualiza según los cambios.

El programa se divide en las siguientes secciones:

1) Inicialización (líneas 30-60). Preparación de dos tablas de 16 x 16, con 6 cadenas para almacenar la colonia inicial. El contador de generaciones se inicializa a 0.

2) Entrada de los datos de la colonia inicial (líneas 70-140). Se introducen y se imprimen 6 cadenas de 6 caracteres.

3) La colonia inicial se almacena en la cadena. Las líneas 150-190 sitúan la célula "1" en la cadena B cuando se detecta un asterisco "*" en la cadena inicial teclada en el punto 2, en el bloque central de 6 x 6.

4) Subrutina de impresión. La línea 200 manda el control a la sub-

```

270 IF A(X-1,Y)=1 THEN LET C=C
+1
290 IF A(X-1,Y+1)=1 THEN LET C
=C+1
300 IF A(X,Y+1)=1 THEN LET C=C
+1
310 IF A(X,Y-1)=1 THEN LET C=C
+1
320 IF A(X+1,Y-1)=1 THEN LET C
=C+1
330 IF A(X+1,Y)=1 THEN LET C=C
+1
340 IF A(X+1,Y+1)=1 THEN LET C
=C+1
345 REM *DECIDE SI NACE*
350 IF A(X,Y)=0 AND C=3 THEN L
ET B(X,Y)=1
355 REM *DECIDE SI MUERE*
360 IF A(X,Y)=1 AND (C>3 OR C<2
) THEN LET B(X,Y)=0

```

```

370 NEXT Y
380 NEXT X
400 GO TO 200
1000 CLS
1005 PRINT AT 0,1:"GENERACION ";
GEN
1010 FOR X=1 TO 16
1020 FOR Y=1 TO 16
1025 REM *ACTUALIZACION DE LA CA
DENA A*
1030 LET A(X,Y)=B(X,Y)
1035 REM *IMPRIME LA CADENA*
1040 IF A(X,Y)=1 THEN PRINT AT
X+2,Y+6;"*"
1050 IF A(X,Y)=0 THEN PRINT AT
X+2,Y+6;" "
1060 NEXT Y
1070 NEXT X
1080 COPY
1090 RETURN

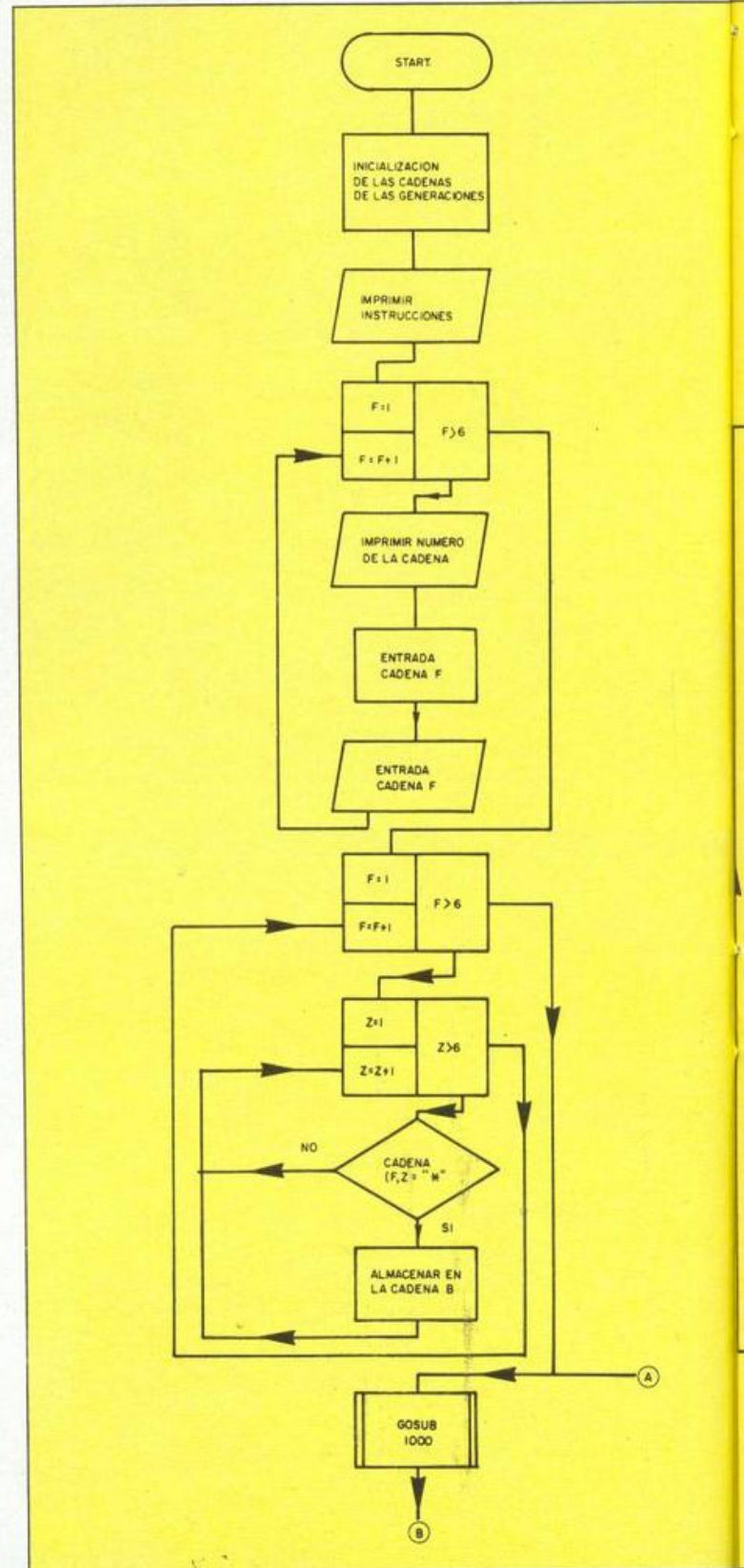
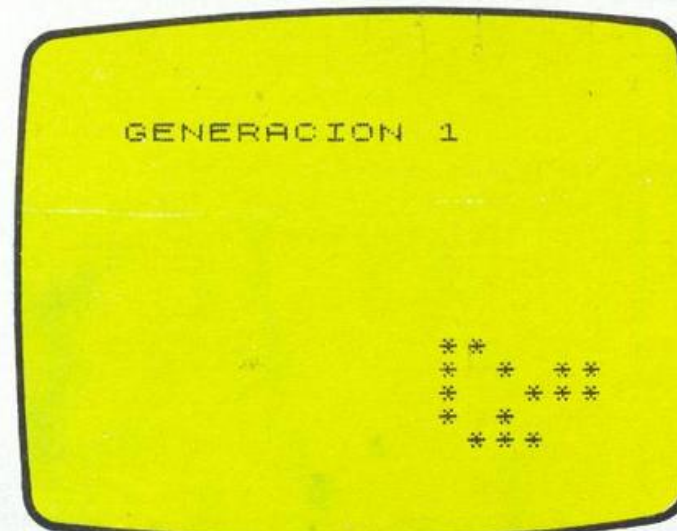
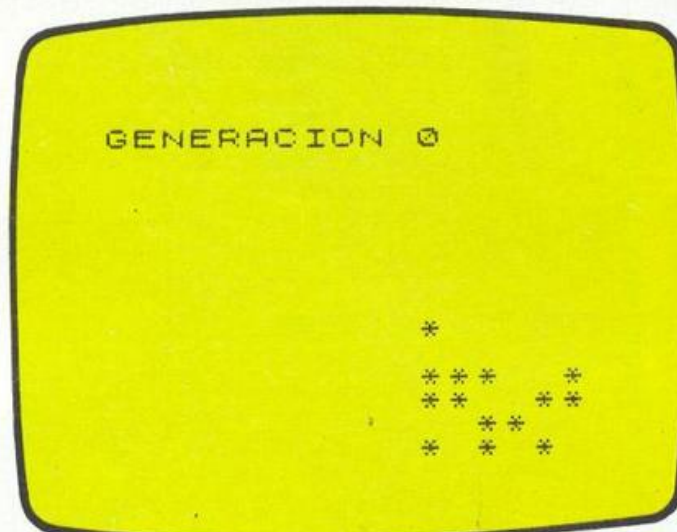
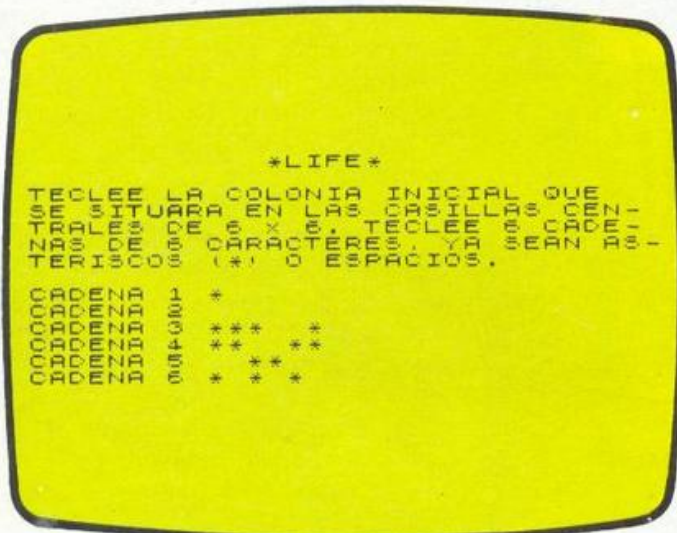
```


rutina. Esta utiliza un bucle doble para igualar las cadenas A y B, y a continuación imprime un asterisco en el tablero por cada "1" que encuentra en la cadena A. Observe que al final de esta subrutina, las cadenas A y B contendrán la misma información. La pantalla se

copia y se devuelve el control a la línea 210. Aquellos que no tengan la impresora tendrán que omitir la línea 1080.

5) Comprobación de la población para determinar la generación siguiente. La generación se incrementa en una unidad. El bucle do-

ble se inicializa y cada célula en la cadena A se comprueba para ver el número de células vecinas que posee. El contador C se incrementa en una unidad por cada vecino que tenga. La línea 350 sitúa una célula "recién nacida" en la cadena B si la célula de la cadena A está en



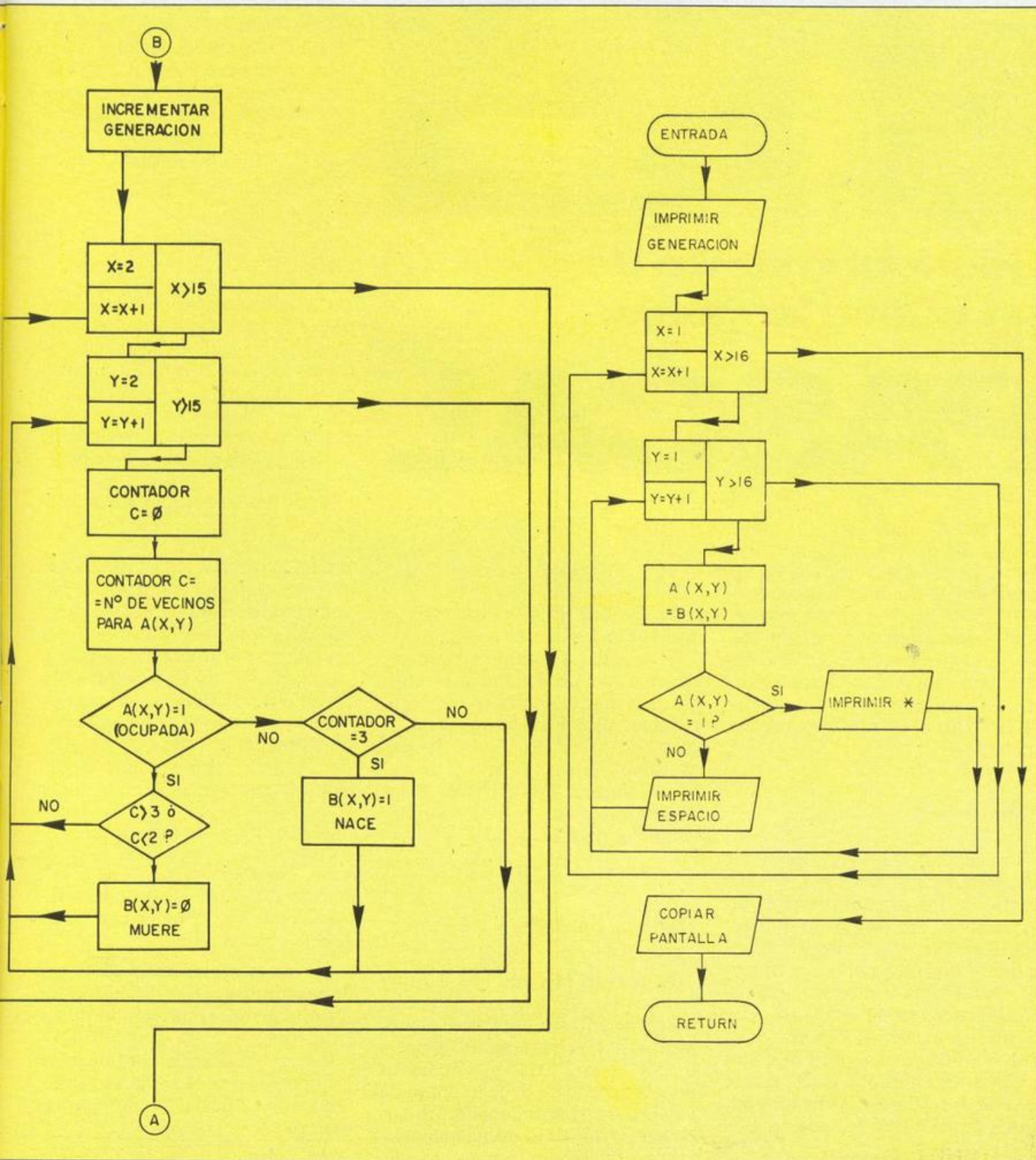
blanco, y el número de vecinos es 3. La línea 360 mata cualquier célula que esté rodeada de 3 o menos de dos vecinos. Al elemento correspondiente en la cadena B se le pone un 0. Llegados aquí, la cadena B contiene la futura generación de células. La cadena A se deja

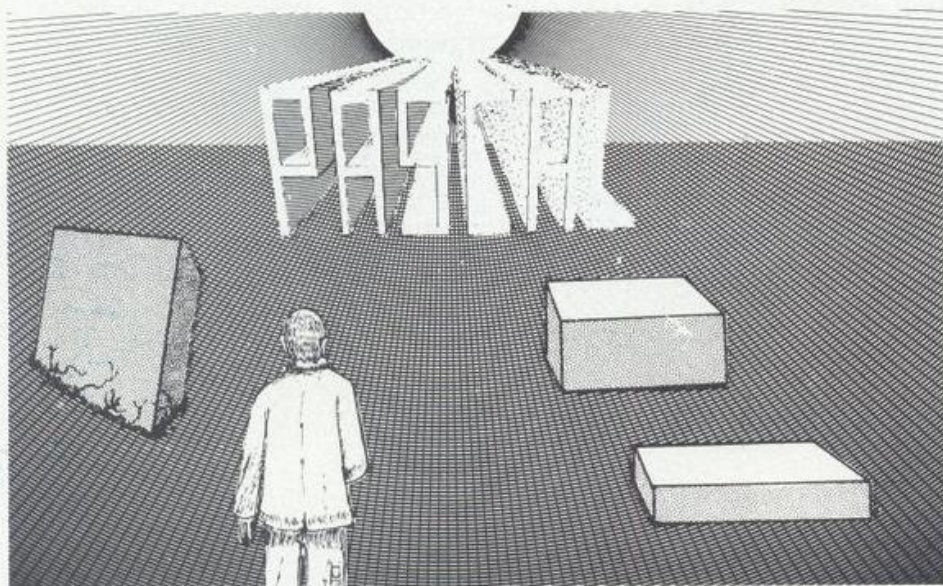
para el proceso de comprobación. Esta es la razón por la que las dos cadenas se igualan en el punto 4.

6) El control se devuelve a la línea 200 y se repiten los puntos 4 y 5.

La forma de situar los asteriscos al comienzo, determinará la mane-

ra de desarrollarse las células. Algunos planteamiento iniciales mueren después de unas cuantas generaciones, mientras otros entran en un ciclo estable que se repite y hay una tendencia general hacia la simetría si la población sobrevive el tiempo suficiente.





Descubrimiento de un nuevo lenguaje:

PASCAL

En este segundo capítulo vamos a tratar de explicar más detenidamente la manera de formar las expresiones de las que hablamos anteriormente, y una visión más general sobre los tipos básicos de datos. Veremos los datos de tipo entero, boolean, real, y el tipo carácter, las expresiones que podemos formar con ellos y los operadores correspondientes.

Expresiones

En general las expresiones las utilizamos en diferentes contextos dentro de los programas, pero todas tienen un propósito básico, que es el obtener un valor como resultado. Por otra parte, los distintos operadores dentro de una expresión tienen una jerarquía, esto es, un sistema de prioridades, o sea que dentro de una expresión se ejecutan unos antes que otros. Así la lista de operadores clasificados por orden de prioridad decreciente es:

- 1) NOT, y cambio de signo: —

- 2) DIV, MOD, *, /, AND
- 3) +, -, OR
- 4) <, ≤, =, ≥, >, <>

La sintaxis de una expresión se observa en las figuras 1 a 4.

Si queremos modificar este orden natural, tendremos que utilizar paréntesis, siendo la parte encerrada entre éstos la que toma la más alta prioridad. Todas las expresiones tienen un tipo y un valor. Vamos a estudiar ahora los distintos tipos de expresiones, junto con los tipos de datos.

Datos de tipo entero

El tipo INTEGER nos permite representar, almacenar y procesar números enteros en Pascal (cero, positivos y negativos). El tamaño de los enteros con que podamos operar dependerá de la longitud de palabra de cada ordenador. Si suponemos que el identificador *maxint* es el máximo valor entero per-

mitido, el Pascal operará con valores enteros dentro del rango:

— *maxint*, ..., -1, 0, 1, ..., *maxint*

Si se excede del máximo valor *maxint* se producirá la condición de *overflow* (exceso de capacidad).

En lo referente a las constantes enteras, éstas se escriben como una secuencia de dígitos decimales, opcionalmente precedida por un signo: “-” indicará un número negativo, y el signo “+” o sin signo indicará un número positivo (figuras 5 y 6). Ejemplos de constantes enteras serían:

máximo = 100
mínimo = -máximo
n = -245

Un ejemplo de constante errónea sería 12.05, pues contendría el carácter especial “.”. Para usar variables enteras dentro de un programa tendremos que especificarlo en la parte de declaración de variables de tipo entero. Un ejemplo de declaración de variables enteras es:

VAR número 1, número 2: integer;

Una expresión entera en el caso más general está constituida por una serie de constantes enteras y variables enteras ligadas por los operadores aritméticos de suma (+), resta (-), multiplicación (*) y división entera (DIV).

Recordemos que también es posible obtener el resto de una división usando el operador MOD. Este operador viene definido por:

$$a \text{ MOD } b = a - (a \text{ DIV } b) * b$$

En una expresión entera el orden de prioridad de las operaciones se realiza de acuerdo con las siguientes reglas:

1. Se efectúan las operaciones de multiplicación y/o división indistintamente, comenzando por la izquierda.

2. Se realizan a continuación las operaciones de suma y/o resta, también empezando por la izquierda.

Así, por ejemplo, la expresión

entera: $a + b * c - d$, se evaluará de la siguiente forma:

- 1) $b * c = Y$
- 2) $a + Y = Z$
- 3) $Z - d$

En Pascal no existe el operador de exponenciación. Sin embargo, disponemos de la función estándar $SQR(x)$ que nos permite calcular el cuadrado de x . Por último, para terminar con los datos de tipo entero debemos decir que el valor de una expresión entera puede asignarse a una variable entera o a una variable real.

Así, si hemos declarado:

VAR

velocidad, tiempo: INTEGER;
espacio: REAL;

la asignación `espacio := velocidad * tiempo` es perfectamente posible.

Datos de tipo boolean

Las constantes y variables booleanas pueden tomar únicamente dos valores: TRUE o FALSE (verdadero o falso).

El conjunto lógico está ordenado en Pascal por la relación de falso < verdadero.

Para utilizar una variable lógica dentro de un programa la debemos declarar con el atributo BOOLEAN. Por ejemplo:

VAR

lógica: BOOLEAN;

Existen tres operadores booleanos o lógicos: AND(\wedge), OR(\vee) y NOT(\neg). La tabla de verdad que representa la lógica de actuación de estos operadores aplicados a las variables booleanas p y q es:

p	q	$p \vee q$	$p \wedge q$	$\neg p$
true	true	true	true	false
true	false	true	false	false
false	true	true	false	true
false	false	false	false	true

Igualmente un valor lógico lo podemos obtener como resultado de cualquier relación o compara-

ción, donde una relación viene definida por:

e_1 operador de relación e_2

siendo e_1 y e_2 expresiones aritméticas y los operadores de relación los ya conocidos $<$, \leq , $=$, $>$, \geq , $<>$. Veamos un ejemplo para com-

prender mejor lo dicho. Sean p y q dos variables booleanas, y X , Y , Z tres variables enteras que tienen los valores 5, 8 y 13, respectivamente.

Las sentencias:

$p := x = y$

$q := (x < y) \wedge (y \leq z)$

Figura 1

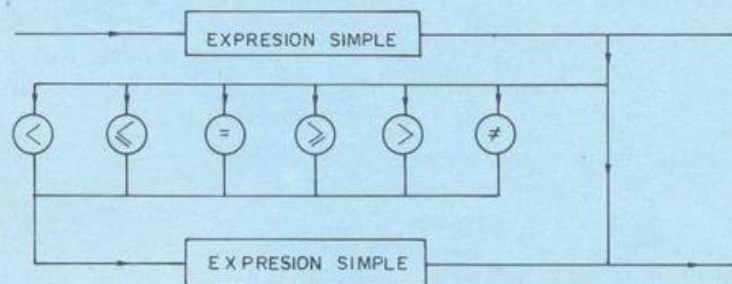


Figura 2

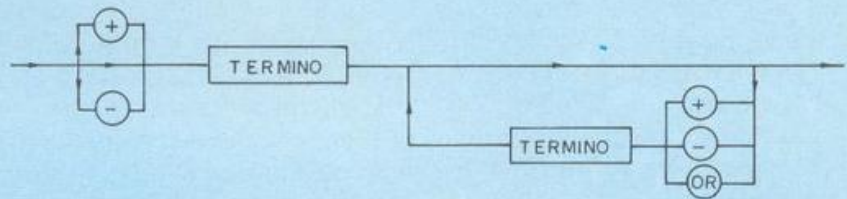


Figura 3

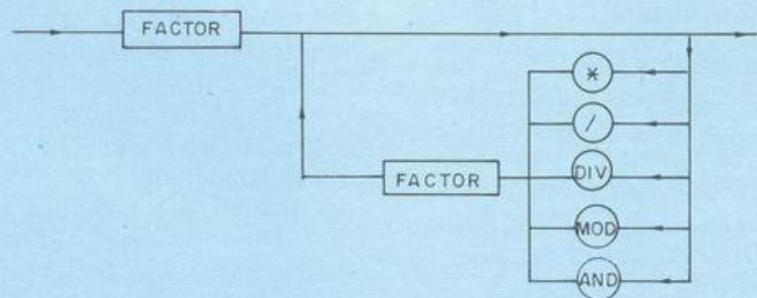


Figura 4



PASCAL

definen los valores lógicos de $p = \text{false}$ y $q = \text{true}$.

La sintaxis de las expresiones lógicas puede verse en las figuras 7 a 9.

La sentencia de lectura **READ** no acepta variables booleanas. Por tanto, los valores de estas variables se deberán asignar a lo largo del programa, nunca mediante una sentencia de lectura.

Las expresiones de Boole son aquellas que al evaluarlas dan como resultado un valor lógico (*true* o *false*). Como hemos visto, un caso particular de una expresión es una comparación. Por ejemplo: $1 < 6$ da *true* al evaluarla. Pero una expresión booleana propiamente dicha consiste en una combinación de constantes lógicas (*true* o *false*), una variable lógica, una función estándar lógica y una función lógica de usuario. Uniéndolos mediante uno de los operadores booleanos **NOT**, **AND** y **OR**, advirtiendo que el operador **NOT** sólo afecta al operando que le sigue, es decir, es unario, mientras que **AND** y **OR** son binarios.

Para terminar con el tipo boolean vamos a comentar tres funciones estándar de Pascal que entregan un valor lógico al ser llamadas:

a) La función **ODD(x)** para x entero pregunta si es impar, dando *true* en este caso y *false* en caso contrario.

b) La función **EOF(f)** indica si se ha alcanzado el final del fichero f dando *true* en este caso y *false* en caso contrario.

c) La función **EOLN(f)** que indica si se ha alcanzado el fin de línea del fichero f .

Estas dos últimas funciones serán vistas más adelante.

Datos de tipo real

Cuando deseamos medir cantidades físicas, como distancias o tiempos, los números enteros nos resultan inadecuados. Estas mediciones dan lugar a números del

tipo 15.77, 10^5 , 0.001, etc., a los que llamaremos números reales. Al igual que en los números enteros, existe un límite para el número de dígitos de un número real. Este límite dependerá de cada ordenador.

Existen dos formas de representar las constantes reales: en forma de notación decimal (coma fija) y en forma científica o coma flotante.

Así, por ejemplo, $11E3$ equivale a 11000.0 y $1E-3$ equivale a 0.001, donde la letra **E** representa el factor de escala 10.

Es imprescindible que el punto sea precedido y seguido de una cifra como mínimo. Por ejemplo, las constantes 4. .0 250 son incorrectas.

Las variables reales deben ser declaradas mediante el atributo real. Un ejemplo de una declaración de variables reales es:

```
VAR  
N1, N2: REAL;
```

La sintaxis de un número real se representa en la figura 10.

Las expresiones aritméticas con números reales se construyen de manera similar a las expresiones enteras. Los operadores aritméticos utilizados son: +, -, *, /, siendo "/" división real, es decir, el cociente es siempre un valor real.

Veamos ahora las restricciones con respecto a las asignaciones.

Mediante una sentencia de asignación se puede asignar a una variable real un valor o expresión que sea *integer* o real.

En cambio, no podemos asignar a una variable *integer* un valor o expresión real. Así, por ejemplo, si hemos declarado la variable **I** como *integer* la sentencia $I := 2.345$ es incorrecta.

En cuanto a las expresiones, una expresión entera no puede contener operandos reales, mientras que una expresión real puede contener operandos reales o enteros indistintamente.

Además de la función **SQR** que ya hemos visto, el lenguaje Pascal tiene incorporadas un conjunto de funciones estándar en las cuales se acepta un determinado tipo de parámetro (*integer* o real) y devuelve

un determinado tipo de resultado (*integer* o real), según la función. La lista de las funciones es la siguiente:

- **ABS(x)** valor absoluto de x .
- **SQR(x)** cuadrado de x .
- **SIN(x)** seno de x (x en radianes).
- **COS(x)** el coseno de x para x en radianes.
- **ARCTAN(x)** arcotangente de x (el resultado en radianes).
- **EXP(x)** e^x .
- **LN(x)** el logaritmo neperiano de x .
- **SQRT(x)** raíz cuadrada de x (x mayor que 0).
- **ROUND(x)** valor de x redondeando al entero más próximo, por ejemplo, $\text{ROUND}(3.7) = 4$.
- **TRUNC(x)** parte entera de x , por ejemplo, $\text{TRUNC}(3.7) = 3$.

Datos de tipo caracter

Los datos de tipo caracter los forman un conjunto finito y ordenado de caracteres en el que están incluidas las letras, los dígitos, el caracter blanco y los caracteres especiales. Por supuesto, el juego de caracteres reconocidos dependerá de cada ordenador.

Una constante de tipo caracter es un elemento del conjunto de caracteres disponibles. Se representa dentro de un programa por el caracter encerrado entre apóstrofes (comilla simple), excepto el propio apóstrofe que a efectos sintácticos necesita ser duplicado. Por ejemplo:

```
'1' '/' 'V' ''
```

representan a los caracteres 1, /, V y ' , respectivamente.

Estas constantes también pueden ser una cadena de caracteres delimitadas por dos apóstrofes. Un ejemplo de una cadena de caracteres sería: **'CADENA'** que constaría de 6 caracteres.

En cuanto a las variables de tipo caracter deben ser declaradas en la parte de declaración de variables con el atributo **CHAR**. Por ejemplo:

VAR

caracter, palabra: CHAR;

La lectura de datos de tipo caracter debe realizarse caracter a caracter. Por tanto, si queremos leer una cadena de caracteres lo deberemos hacer con sentencias repetitivas las cuales veremos más adelante.

Vamos a ver, por último, cuatro funciones estándar que actúan sobre datos de tipo caracter. Hemos dicho que el conjunto de caracteres es un conjunto ordenado. La función ORD(x), siendo x un caracter obtiene el valor ordinal de x en el conjunto ordenado de caracteres. Evidentemente el valor obtenido con esta función depende del conjunto de caracteres empleados por el sistema. En cualquier caso el máximo valor es el 255. En el sistema de codificación ASCII:

$$\text{ORD}('1') = 48$$

$$\text{ORD}('3') = 50$$

La función CHR(x) es la inversa de ORD(x). Obtiene el caracter que corresponde al valor entero x dentro del conjunto ordenado de caracteres. Por tanto, se cumplen las relaciones:

$$\text{CHR}(\text{ORD}(x)) = x$$

$$\text{ORD}(\text{CHR}(i)) = i$$

Existen también otras dos funciones que operan dentro del conjunto de caracteres. La función PRED(x) obtiene el caracter precedente al caracter x dentro del conjunto ordenado de caracteres. x no puede ser el primer elemento del conjunto.

Por ejemplo:

$$\text{PRED}(5) = 4$$

$$\text{PRED}('B') = A$$

y la función SUCC(x) obtiene el caracter que sigue al caracter x dentro del conjunto ordenado de caracteres. También deben cumplirse las identidades:

$$\text{PRED}(\text{SUCC}(x)) = x$$

$$\text{SUCC}(\text{PRED}(x)) = x$$

Dentro del conjunto de valores booleanos:

$$\text{PRED}(\text{true}) = \text{false}$$

$$\text{SUCC}(\text{false}) = \text{true}$$

$$\text{SUCC}(\text{true}) = \text{ERROR no existe.}$$

Vamos a incluir a continuación un pequeño programa para ver

más claramente el uso de las expresiones aritméticas, y de algunas de las funciones estándar de las que ya hemos hablado. El programa calcula las raíces de una ecuación de segundo grado. Incluye una sentencia IF de la que hablaremos en próximos capítulos.

También se ha incluido una

Figura 5

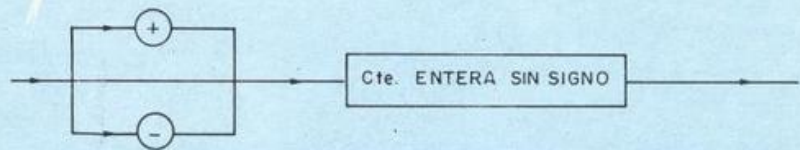


Figura 6

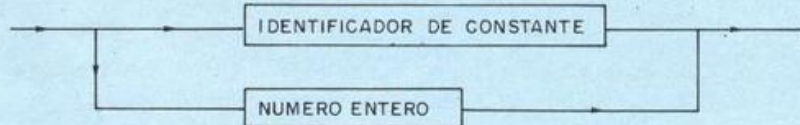


Figura 7

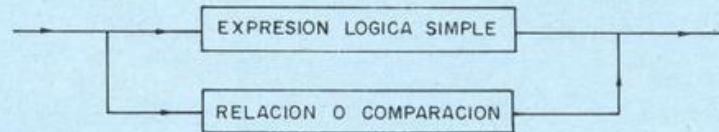


Figura 8

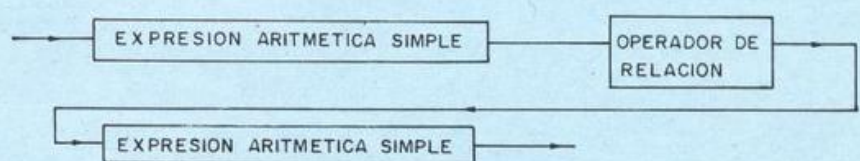


Figura 9

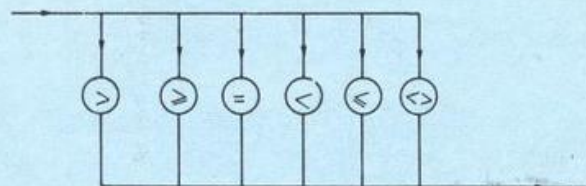
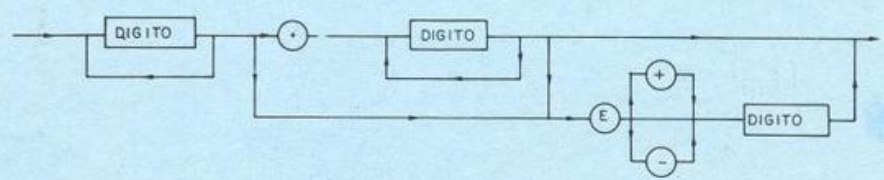


Figura 10



PASCAL

constante de tipo caracter (cadena de caracteres).

El programa de ejemplo realiza la resolución de ecuaciones de segundo grado por la fórmula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

En él aparecen las tres sentencias básicas que hemos explicado

hasta ahora (lectura, escritura y asignación).

La parte de declaración comprende las líneas 20 a la 60. En ella hemos declarado una constante de tipo cadena de caracteres, y a su vez todas las variables que hemos usado durante el programa.

Todas las variables son de tipo real.

La parte principal del programa (cuerpo) comprende las líneas 70 a la 280. Lo primero que deberemos tener en cuenta para este programa es que la raíz cuadrada tiene que ser siempre positiva. Por lo tanto, en la variable DISCRI, efectuamos

la operación aritmética del cálculo del discriminante de la ecuación, con las funciones estándar que ya hemos explicado. Si esta variable DISCRI es mayor o igual a cero, pasamos a calcular las raíces de la ecuación y a imprimirlas (líneas 120 a 190), y en caso contrario, imprimimos un literal indicando que no existen raíces reales para la ecuación dada. La instrucción IF será explicada en el próximo capítulo.

Por último, recordar que la instrucción (WRITE(CHR(16))) es propia del compilador que hemos usado.

```

AC73 10 PROGRAM ECUACION;
AC73 20 CONST
AC73 30 SEPARADOR='-----'
AC7F 40 VAR
AC88 50 A,B,C,X1,X2:REAL;
AC88 60 DISCRI:REAL;
AC88 70 BEGIN
AC91 80 READ(A,B,C);
AC9F 82 WRITE(CHR(16));
ACB6 85 WRITELN('LA ECUACION
PEDIDA ES:');
ACDA 86 WRITELN(A:4:1,'X2+',
B:4:1,'X+',C:4:1);
AD34 90 WRITELN(SEPARADOR);
AD3F 100 DISCRI:=SQRT(B)-4*A*C
AD77 110 IF DISCRI>0 THEN
AD9D 120 BEGIN
AD9D 130 X1:=(-B+SQRT(DISCRI
))/2 *A;
ADD9 140 X2:=(-B-SQRT(DISCRI
))/2 *A;
AE19 150 WRITELN('LAS RAICES
DE LA ECUACION SON:');
AE45 160 WRITELN;
AE48 170 WRITELN('X1=',X1:7:
3);
AE6D 180 WRITELN;
AE70 190 WRITELN('X2=',X2:7:
3);
AE92 200 END
AE95 210 ELSE
AE98 220 BEGIN
AE98 230 WRITELN;
AE98 240 WRITELN('NO EXISTEN
RAICES REALES');
AEC1 250 WRITELN
AEC1 260 END;
AEC4 270 WRITELN(SEPARADOR)
AECC 280 END.
End Address: AED1
    
```

Ejemplo de asignación, entrada-salida de datos, declaración de las constantes (N=3), variables y utilización de las instrucciones BEGIN-END. WRITELN sirve para escribir por líneas.

```

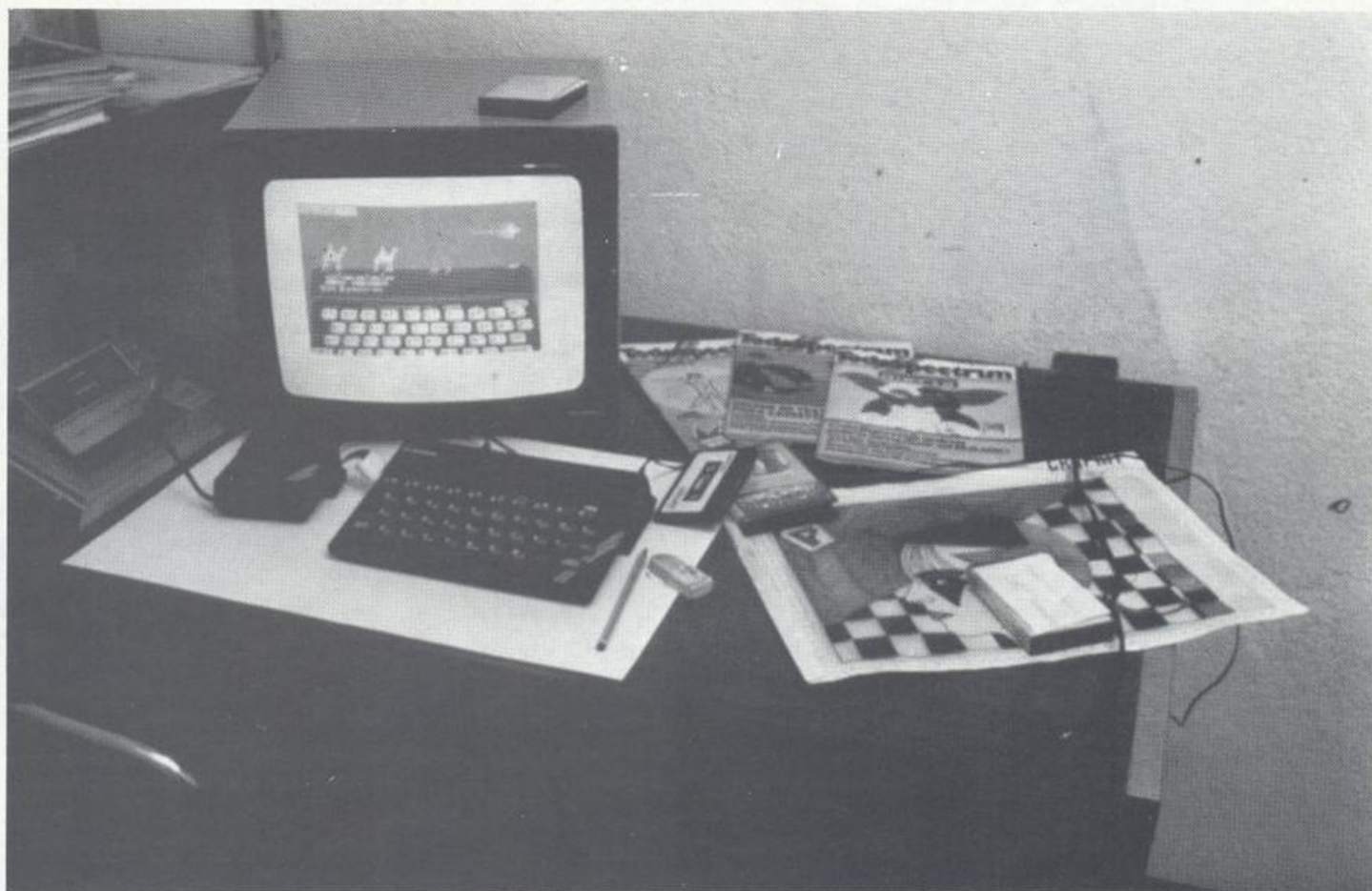
LA ECUACION PEDIDA ES:
1.0X2+ 7.0X+ 4.0
-----
LAS RAICES DE LA ECUACION SON:
X1= -0.628
X2= -5.372
-----
    
```

```

LA ECUACION PEDIDA ES:
1.0X2+ 5.0X+ 3.0
-----
LAS RAICES DE LA ECUACION SON:
X1= -0.697
X2= -4.303
-----
    
```

```

LA ECUACION PEDIDA ES:
1.0X2+ 2.0X+ 3.0
-----
NO EXISTEN RAICES REALES
-----
    
```

Así hacemos la portada

Mucho se ha hablado y se seguirá hablando sobre las capacidades gráficas del Spectrum. Una forma de utilizarle al máximo en este sentido, es trabajar con los programas de diseño, cada vez más abundantes, y que requieren estudio detallado que emplazamos para un próximo número. Por el momento, nos fijamos en uno en particular: **El Artist**, gracias al cual hemos venido realizando las portadas de **TODOSPECTRUM**.

Sea cual sea el programa elegido, realizar un gráfico de estas características requiere grandes dosis

de paciencia y, por supuesto, un boceto previo. En esta etapa inicial de dibujo no interviene el Spectrum y, es más, el dibujante ni siquiera lo conoce, concentrado en la tarea creadora en su estudio de Barcelona. Una vez realizado el dibujo, viene la segunda parte: pasarlo al Spectrum. Resulta prácticamente imposible realizar una copia exacta y uno de los mayores problemas reside en la combinación de tres o más colores en una cuadrícula o carácter. El objetivo es sacar el mayor provecho del Spectrum y del programa.

Para ser lo más fieles posibles al

dibujo, se siluetea en la pantalla, momento a partir del cual entra en juego **el Artist**. Veamos brevemente sus principales características.

El programa permite tres modos de utilización: **ACTIVO**, para trazar líneas a su paso; **PASIVO**, para desplazamientos sin modificar el dibujo; y **BORRADO**, para eliminar el dibujo a su paso.

El movimiento de cursar se logra presionando diversas teclas: (Q) subir, (A) bajar, (Z) izquierda, (X) derecha, (W) diagonal ascendente izquierda, (R) diagonal ascendente derecha y (D) diagonal descendente.



te derecha. El tamaño y la velocidad del cursor es seleccionable.

Estos reducidos conocimientos de funcionamiento del programa son suficientes para trazar el cursor por la silueta previamente marcada sobre la pantalla, es decir, para memorizar los primeros datos. A continuación vienen los detalles. En el caso del SANTA-CLAIR, inicialmente se dibujó a Papa Noel arrastrando un saco de regalos. Después vendría la ciudad del fondo, el reno con el trineo, los Spectrum de regalo y, finalmente, la figura de **Sir Clive Sinclair**, quien a buen seguro se quedó con el saco vacío durante las pasadas navidades.

El contorno o silueta de las figuras no encierra grandes problemas. El mayor contratiempo es la introducción del color. El Spectrum sólo admite dos colores por carácter, pero se deja engañar un poco. Siguiendo los colores empleados por el dibujante (en base a la selección que ofrece el Spectrum) se colorea carácter a carácter con el color que mayoritariamente aparezca, además de la silueta dibujada anteriormente, la silueta carácter a



carácter del color que toma cada parte del dibujo.

Para poner el color hemos de estar en modo texto. Una vez aquí aparecerá un cursor del tamaño de un carácter que podemos mover con las teclas usuales (punto 3). Al mismo tiempo en las líneas de información vemos el color en el que dibujaremos al poner ACTIVO el cursor, este puede ser alterado fácilmente pulsando:

- 6: selecciona INK
- 7: selecciona PAPER
- 8: BRIGHT/NO BRIGHT
- 9: FLASH/NO FLASH

Una vez que seleccionemos lo que nos sea necesario pondremos el cursor en modo ACTIVO pulsando "1" y dejaremos ese color en el carácter en donde se encuentre. Terminado esto, tendremos que amoldar el dibujo inicial para aprovechar al máximo el color que corresponde al carácter.

Ahora solamente queda delimitar qué zona va a ser INK o PAPER. Nos vamos a valer de los dibujos de la figura 1, correspondientes a la frente de SANTA-CLAIR, para verlo mejor.

El carácter que tiene un solo color no se modifica (A1).



Los caracteres B1, B2 y A2 tienen dos colores, por lo que hay que elegir qué parte de ellos va a ser INK o PAPER.

Para utilizar el FILL la parte a rellenar debe estar cerrada por todos sus lados y para saber qué partes del dibujo están en INK o PAPER podemos utilizar el GRID (es INK todo lo que aparece en negro).

Otro pequeño truco es utilizar la potencia de los gráficos definidos. Para ello del modo texto volvemos al menú principal y seleccionamos la opción UDG. Aparece un GRID para diseñar gráficos y un papel editor, sobre el cual se des-



En una moderna urbanización de las afueras de Madrid, Víctor Ruiz desarrolla su trabajo en lo que un joven grupo de programadores ha denominado la "Mansión Dinámica", compartiendo su código máquina con los estudios universitarios. Estuvimos allí un domingo "que es cuando coges a todos".

Con su Spectrum, Víctor comenzó a trabajar con el Artist cuando el grupo aún no se había formado.

"Lo hice para mí. Conocía el Paint Box y quería mejorarlo en aquellos puntos que veía flojos, especialmente en el movimiento del cursor que es muy lento, así como tener la posibilidad de simultanear las representaciones gráficas con la definición de bloques gráficos". Estuvo cuatro meses desarrollando la idea, "sin prisas, porque era para mí. Los mayores problemas los tuve con la rutina tridimensional 3D. Cada punto son cuatro bytes y



plaza un cursor (controles en punto 3) con el que pondremos o quitaremos pixels del GRID pulsando CAPS SHIFT. Una vez que nos convence el gráfico diseñado lo podemos definir en el caracter y banco seleccionado.

Además de definir tenemos una serie de posibilidades: Rotar, Invertir y Reflejar.

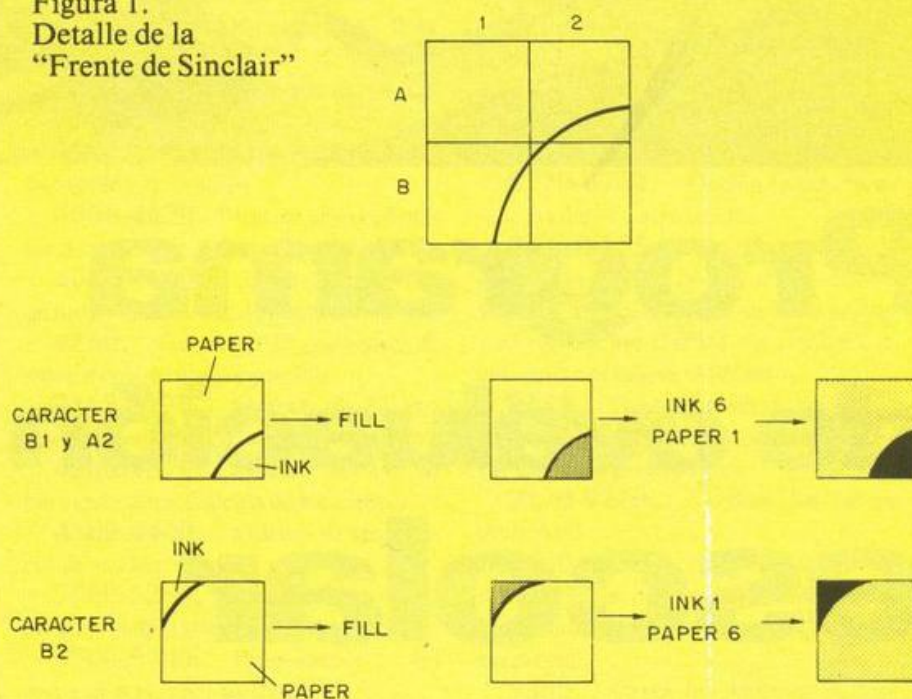
Para introducir estos gráficos en pantalla volvemos al modo texto y seleccionamos la opción gráficos. Nos aparece el cursor que controlamos con CAPS SHIFT y los cursores. Los situamos en el lugar que nos interesa y presionamos la tecla que corresponda al gráfico diseñado. En el caso del SANTACLAIR las "firmas" se introdujeron de esta manera.

Finalmente destacar que el **Artist** permite la impresión de "textos gigantes", así como gráficos en tres dimensiones. En el primer caso, se presiona "5" en el modo texto, apareciendo el cursor que situaremos en el lugar donde empezaremos a escribir. A continuación presionamos "0", introducimos el ancho y el alto (en caracteres) y el texto a insertar. En el segundo caso, se pueden hacer rotar los grá-

ficos, sobre todos sus ejes, y acercarlos o alejarlos (en forma de ZOOM). Pulsando la tecla "0", se mezcla la figura con el dibujo que tuviéramos realizado.

Y esto es todo. Lo más importante es disponer de grandes dosis de paciencia. Seguro que le quedarán preciosos. ¿Cuándo dijo que nos lo mandaba?

Figura 1.
Detalle de la
"Frente de Sinclair"



como se guardan las pantallas para saltar entre menús, la rutina que había realizado no cabía. Tuve que

"Por el mismo trabajo haces un juego que vendes cinco veces más"

ubicarla en otra dirección inferior pero el direccionamiento no era el

adecuado. Cuando por fin ves rotar un cubito... se siente uno bien".

Con lo que no se sentía tan bien es con la respuesta del público. "Por cada cinco juegos te piden un Artist. No compensa hacer programas de utilidades, por el mismo trabajo haces otro que vendes cinco veces más. Por eso ahora hemos abandonado las aplicaciones".

Una empresa de Barcelona quiere diseñar mochilas y son precisamente sus posibilidades de diseño

las que le hacen especialmente interesante para ellos mismos en su confección de carátulas y pantallas de los juegos que más tarde salen al mercado "Es muy importante la pantalla de carga, como se ve en todos los juegos de Ultimate. Si las pantallas son malas, algo marcha mal. Salvo la pantalla del Fred, somos los únicos que nos estamos preocupando por las pantallas de carga. Es una cuestión de imagen muy importante".



Programa de evaluación escolar

EL proceso educativo consta de cinco etapas: fijación de metas; objetivos; estrategias metodológicas; método, aprendizaje, evaluación y recuperación.

No se puede pensar en la realización completa de un acto educativo si falta cualquiera de dichas etapas. Por tanto, la evaluación se halla integrada en el proceso educativo, siendo eslabón inseparable de toda una cadena de actividades. La evaluación podría definirse como un proceso sistemático para determinar hasta qué punto los objetivos educativos han sido logrados por los alumnos. Con lo dicho resulta fácil concluir que la evaluación es un término en plena vi-

gencia y de rico contenido; siendo un proceso no fácil pues toma en algunos casos significado de juicio a los alumnos con el fin de sólo medir sus conocimientos. Sin embargo, justifica su significado e importancia en los siguientes factores:

— La preocupación por la eficacia docente, que busca obtener el máximo provecho de los medios disponibles y lograr información constante del proceso educativo.

— La nueva concepción del rendimiento en la enseñanza. La evaluación es tarea más compleja que una simple medida. La medición sólo es una fase de la evaluación que suministra datos, pero evaluar significa además emitir

Cuando se habla de un ordenador en la escuela, lo primero que se piensa es en una clase llena de ordenadores.

Esto puede ser factible en un futuro no muy lejano.

Por el momento, nos centramos en el profesor.

Dada su longitud, disponemos de copias grabadas que podrá solicitar retirando el cupón que se adjunta.

juicios de valor sobre ellos, es decir, obtener parámetros cualitativos y cuantitativos del alumno, consiguiendo una información lo más completa y objetiva posible que sirva de conocimiento para el profesor.

— La necesidad de individualización pedagógica para alcanzar la eficacia deseada. Es decir, los juicios valorativos que se emiten sobre las mediciones han de estar basadas en la personalidad del alumno y en las necesidades sociales.

— La necesidad de superar la ineficacia del sistema clásico de control. Este, basado sólo en exámenes conlleva, tanto en la preparación del examen como en la calificación, un alto nivel de subjetivi-

dad, además de obligar al alumno solamente a pasar la prueba, y no a estimular su formación individual.

En definitiva, la evaluación ha de tener los medios adecuados, que los hay actualmente (aunque no siempre se usan), para prestar atención continua, racional, específica y objetiva hacia el alumno.

Instrumento de evaluación

Los instrumentos de evaluación deben estar adaptados a cada etapa del sujeto. Así como para el preescolar, ciclo inicial y ciclo medio es interesante un control basado en la observación, entrevistas, pruebas orales (sin carácter de examen), e incluso algún control escrito, aunque éstos incluyan un cierto nivel de subjetividad, son los instrumentos más provechosos y adaptables para estas etapas, en el ciclo superior y otras etapas posteriores (BUP, COU, universitarios, etc.) se pueden utilizar los ya citados, siempre con menos intensidad, pero sobre todo los trabajos en grupo, temas a desarrollar y los test. Debe tenerse en cuenta que dependiendo del grupo de sujetos, estos instrumentos han de ser flexibles y acomodarse al nivel de desarrollo de comprensión lectora. Así, por ejemplo, pueden utilizarse pruebas tipo test en sujetos que se encuentren en la etapa final del ciclo medio, o, por el contrario, no poderse utilizar los test en sujetos del ciclo superior.

De entre todos los instrumentos de evaluación, el test proporciona una información más completa y objetiva del sujeto. Aunque su elaboración, para que sea eficaz, es complicada para el profesor, lo que supone una alta preparación por su parte. Sus ventajas son múltiples e innegables:

- Permiten una rápida corrección y calificación: plantillas o computadoras.

- Posibilidad de realizar controles con frecuencia.

- Los resultados son comparativos, puesto que todos los sujetos se juzgan con iguales criterios.

- Los resultados son independientes del estado de ánimo y subjetividad del profesor al corregir.

1135-1140. Subrutina de entrada válida en un INPUT.

1145-1150. Subrutina de verificación en una entrada por INPUT.

1162-1173. Dimensiona todas las variables con subíndice que se usarán en los cálculos, excepto F() y G().

1174-1320. Entrada de las respuestas de cada *item* para cada sujeto.

1400-1450. Presentación de parámetros.

1500-1735. Cálculo de los aciertos, errores, nulos y de la puntuación de cada sujeto.

2000-2240. Cálculo de la Z de cada sujeto.

3000-3265. Cálculo de los aciertos, errores, nulos y de la puntuación de cada sujeto.

3400-3675. Cálculo de la Z de cada sujeto.

3800-3885. Cálculo del coeficiente de correlación.

3890. Cálculo del coeficiente de fiabilidad.

4000-4070. Rutina para ordenar las puntuaciones en orden decreciente.

4085-4140. Calcula los sujetos en orden decreciente de puntuación.

4160. Guarda el 27 por ciento de sujetos con mayor puntuación.

4165-4170. Guarda el 27 por ciento de sujetos con menor puntuación.

4300-4370. Cálculo de As, Ai para cada *item*. Cálculo de los errores.

4380-4440. Cálculo de los índices de validez.

5000-5200. Presentación del menú principal por pantalla.

5500-5700. Presentación del menú de tablas por pantalla.

6100-6160. Presentación de la tabla de aciertos, errores y nulos.

6200-6260. Presentación de la tabla de puntuación.

6300-6380. Presentación de la tabla de la Z.

6400-6470. Presentación de la tabla de aciertos, errores y nulos por series.

6500-6570. Presentación de la tabla de puntuación por series.

6600-6690. Presentación de la tabla de la Z por series.

6700-6760. Elección del tipo de listado.

6770-6813. Entrada por INPUT del sujeto por donde comienza el listado U, y del sujeto en que termina, H.

6814-6830. Presentación de comandos por pantalla.

6835-6890. Presentación del listado por sujeto por pantalla.

6891-6899. Bifurcación según el comando seleccionado para el listado por sujeto.

6900-6955. Entrada por INPUT del *item* por donde comienza el listado, U, y del *item* en que acaba, H.

6960-6990. Presentación por pantalla del listado por *item*.

6991-6999. Bifurcación según el comando seleccionado para el listado por *item*.

7800-7880. Coeficiente de correlación y del coeficiente de fiabilidad.

7900-7940. Copiar por impresora de los coeficientes anteriores.

8200-8300. Presentación por pantalla de los datos iniciales y la nomenclatura de los índices de validez.

8310-8360. Presentación de la tabla con los índices de validez.

8720-8760. Opción para cambiar los datos a otro sujeto.

9000-9125. En caso de verificarse la grabación de datos, dimensiona y define las variables que se van a grabar y da entrada por INPUT del nombre con que quiere grabarse el fichero.

9130. Grabación datos.

9140-9160. Da opción a verificar los datos grabados.

9170-9220. Verifica los datos grabados.

9400-9490. En caso de verificarse la carga de datos, inicializa todas las variables y dimensiona las que van a cargarse.

9500. Carga los datos en la misma secuencia que fueron grabados.

9510-9560. Definición de las variables que va a emplear el programa, devolviendo el control del programa a la línea 1171 después de llamar a la subrutina que calcula el 27 por ciento de los sujetos.

9700-9750. Presentación por impresora de cualquiera de las tablas.

9800-9840. Subrutina para detener una tabla durante su presentación.

9850-9890. Subrutina para verificar si se quiere continuar con las opciones 4, 5, 6, 7 u 8 del menú principal.

9900-9925. Cálculo del 27 por ciento de sujetos.

9930-9970. Cálculo del espacio libre de memoria y el número máximo de *items* que se pueden introducir.

— Eliminan multitud de factores ajenos al contenido mismo de las pruebas, presentación, caligrafía, estilo, limpieza, etc., que se pueden evaluar con otros instrumentos.

— Ofrecen la posibilidad de cubrir la materia en su máxima amplitud.

Adentrándonos un poco en el tema que nos ocupa, tres son las condiciones que deben adornar al test: validez, fiabilidad y tipificación.

La validez se refiere al grado de exactitud con que el test mide lo que pretende medir. La fiabilidad se refiere a obtener resultados semejantes aplicando la prueba en iguales condiciones para los sujetos. Finalmente, la tipificación se refiere a una escala o baremo que permite interpretar el significado de los resultados. Para ello se pueden tomar distintos criterios.

El programa determina estas tres condiciones primordiales además



de ofrecer otros datos que serán de gran interés para el profesor.

Evaluación de una prueba tipo test

Las preguntas de una prueba tipo test se denominan *items*, y por lo general suelen ser pares. Cada *item* se compone, al menos, de dos respuestas posibles (RP). Para cada item:

— Se considera acierto (A), cuando se contesta la/s respuesta/s correcta/s.

— Se considera nulo (N), cuando no se contesta.

— Se considera error (E), en cualquier otro caso.

Una vez hecho el recuento de aciertos, errores y nulos de cada sujeto se calculan los parámetros en base a dos clasificaciones: por sujeto y por *items*.

POR SUJETO

1. Puntuabilidad (P).

$$P = A - \frac{E}{RP - 1}$$

2. Puntuabilidad tipificada o tipificación (Z).

$$Z = \frac{P_i - \bar{P}}{\sigma_p}$$

P_i = Puntuabilidad del sujeto.

\bar{P} = Puntuabilidad media = $\frac{\sum P_i}{N}$

σ_p = Desviación típica de las puntuabilidades.

$$\sigma_p = \frac{\sum (P_i - \bar{P})^2}{N}$$

N = N.º total de sujetos.

```

O>REM
*****
* Juan Carlos Fdez. Romero
* 17/9/1984
*****
1 REM PARA GRABAR EL PROGRAMA
*USA : SAVE "nombre" LINE 10
*****
10 PAPER 1: INK 7: FLASH 0: BR
IGHT 0: OVER 0: INVERSE 0: BORDE
R 1: CLS
20 PRINT " INK 6: INVERSE 1: "
P E D A G O G I A "
30 PRINT AT 10,7: " EVALUACION
DE UNA ";AT 12,12: " PRUEBA ";AT
14,11: " TIPO TEST "
40 FOR F=19 TO 21: PRINT AT F,
0: PAPER 5:
": NEXT F
50 PRINT AT 20,4: PAPER 3: INK
0: BRIGHT 1: OVER 1: " FIABILID
AD Y VALIDEZ "
60 POKE 23658,8: POKE 23609,20
70 PAUSE 500
80 LET D=65536-USR 7962
90 IF D<15000 THEN GO TO 5000
100 CLS
110 PRINT " QUIERES CARGAR DATO
S, S/N ? "
115 FOR F=0 TO 50: NEXT F
120 IF INKEY$="S" THEN GO TO 9
450
130 IF INKEY$="N" THEN GO TO 1
000
140 GO TO 120
1000 REM ** ENTRADA DE DATOS **
1005 PAPER 1: INK 7: BORDER 1: C
LS
1010 LET J=0
1015 PRINT AT 2,5: "No. DE SUJETO
S = ";

```

```

1020 INPUT "Introduce el No. de
SUJETOS "; LINE I$: LET D=0
1025 GO SUB 1135: IF D=1 THEN G
O TO 1020
1030 PRINT VAL I$: " "
1035 GO SUB 1145: IF D=1 THEN G
O TO 1045
1040 GO TO 1010
1050 LET S=VAL I$
1060 PRINT AT 3,5: "No. DE ITEMS
= ";
1065 INPUT "Introduce el No. de
ITEMS "; LINE I$: LET D=0
1070 GO SUB 1135: IF D=1 THEN G
O TO 1065
1072 GO SUB 9900
1073 IF J=1 THEN PRINT AT 20,0:
INVERSE 1: " MAXIMO = ";H$: " ";AT
3,22: GO TO 1065
1075 IF VAL I$>H THEN CLS: GO
TO 1000
1080 PRINT VAL I$: " "
1085 GO SUB 1145: IF D=1 THEN G
O TO 1095
1090 GO TO 1060
1095 LET I=VAL I$
1096 PRINT AT 20,0: "
"
1100 PRINT AT 4,0: "RESP. POS. PO
R ITEM = ";
1105 INPUT "Introduce el No. de
respuestas posibles en cada ITE
M "; LINE I$: LET D=0
1110 GO SUB 1135: IF D=1 THEN G
O TO 1105
1115 PRINT VAL I$: " "
1120 GO SUB 1145: IF D=1 THEN G
O TO 1130
1125 GO TO 1100
1130 IF VAL I$<2 THEN GO TO 110
0

```

```

1132 LET RP=VAL I$
1133 GO TO 1170
1135 IF CODE I$=0 THEN LET D=1:
RETURN
1137 FOR F=1 TO LEN I$: IF CODE
I$(F)<48 OR CODE I$(F)>57 THEN
LET D=1: RETURN
1138 NEXT F
1139 IF D=0 THEN IF VAL I$<2 TH
EN LET D=1: RETURN
1140 RETURN
1145 POKE 23658,8: PRINT #0: " ES
CORRECTO S/N ? "
1147 IF INKEY$="N" THEN RETURN
1149 IF INKEY$="S" THEN LET D=1
: RETURN
1150 GO TO 1147
1162 REM ** DIM. VARIABLES **
1170 DIM I(S,I)
1171 DIM A(S): DIM E(S): DIM N(S
): DIM P(S): DIM Z(S): DIM X(S):
DIM V(S): DIM Y(S): DIM W(S): D
IM S(S): DIM Q(S): DIM R(S): DIM
M(S): DIM T(S): DIM H(SF): DIM
L(SF): DIM B(I): DIM C(I): DIM J
(I): DIM K(I): DIM O(I)
1172 IF D=2 THEN GO TO 1180
1173 IF D=9450 THEN GO TO 1310
1174 REM **ENTRADA RESPUESTAS **
1175 LET U=1: LET H=S
1180 FOR F=18 TO 20: PRINT AT F,
0: PAPER 3:
": NEXT F
1185 INPUT PAPER 5: INK 1: "REFE
RENCIA DE LOS DATOS?(18 max)"; L
INE R$: IF LEN R$>18 THEN GO TO
1185
1186 IF CODE R$=0 THEN LET R$="
Sin Referencia "
1187 PRINT AT 20,0: PAPER 5: INK
1: " REFERENCIA DE LOS DATOS :

```


Se aplica el baremo elegido a la Z. Uno de los criterios más usuales es el siguiente:

$Z > 2.5$	Sobresaliente/9, 10
$2 < Z \leq 2.5$	
$1.5 < Z \leq 2$	Notable/7, 8
$1 < Z \leq 1.5$	
$0.5 < Z \leq 1$	Bien/6
$0 \leq Z \leq 0.5$	Suficiente/5
$Z < 0$	Suspense/.

3. Se dividen los *items* en pares e impares, formándose así dos series, y se calcula la P y la Z en cada una de las series.

Se obtendrá:

Pi = Puntuación del sujeto en la serie impar.

Pp = Puntuación del sujeto en la serie par.

Zi = Puntuación tipificada del sujeto en la serie impar.

Zp = Puntuación tipificada del sujeto en la serie par.

Para calcular la fiabilidad, primero se obtiene el coeficiente de

TABLA DE ACIERTOS, ERRORES Y LOS			
ITEM NO. 1	A=24	E=11	Na=6
ITEM NO. 2	A=22	E=14	Na=4
ITEM NO. 3	A=26	E=8	Na=6
ITEM NO. 4	A=21	E=15	Na=4
ITEM NO. 5	A=20	E=16	Na=4
ITEM NO. 6	A=24	E=12	Na=4
ITEM NO. 7	A=21	E=15	Na=4
ITEM NO. 8	A=21	E=16	Na=4
ITEM NO. 9	A=20	E=16	Na=4
ITEM NO. 10	A=20	E=16	Na=4

correlación, con el que después obtener el coeficiente de fiabilidad.

1. Se calcula el coeficiente de correlación (R).

$$R = \frac{\sum(Z_i \times Z_p)}{N}$$

2. Cálculo del coeficiente de fiabilidad (CF).

$$CF = \frac{2 \times R}{1 + R}$$

POR ITEMS

En el caso de trabajar con clasificaciones por *items*, primero se ha-

lla el 27 por ciento de sujetos con mayor puntuación y el 27 por ciento con menor puntuación. Y después se determinan para cada *item* los aciertos del 27 por ciento de S. con mayor puntuación, y del 27 por ciento de S. con menor puntuación, As y Ai, respectivamente.

Los índices que se dan para cada *item* son el de facilidad, discriminación, discriminación máxima y de eficacia.

Si no desea teclear el programa, disponemos de una copia en cassette al precio de 200 ptas. Solicitela a TODOSPECTRUM, Bravo Murillo, 377, 5.º A, adjuntando el presente cupón.

SUSCRIBASE POR TELEFONO

- * más fácil,
- * más cómodo,
- * más rápido

Telf. (91) 733 79 69

7 días por semana, 24 horas a su servicio

SUSCRIBASE A

Todospectrum




```

"; PAPER 2; INK 7;R$
1188 POKE 23658,8
1190 PRINT AT 5,0; PAPER 4; INK
0; BRIGHT 1;" PULSA : "; INVERSE
1;" <A> "; INVERSE 0;" si es AC
IERTO ";AT 7,9; INVERSE 1;" <E>
"; INVERSE 0;" si es ERROR ";AT
9,9; INVERSE 1;" <N> "; INVERSE
0;" si es NULO "
1200 FOR J=U TO H
1210 PRINT AT 12,1; INVERSE 1;"
SUJETO No.":J;" "
1220 FOR F=1 TO I
1230 PRINT AT 14,1;"ITEM No.":F;
" "
1240 PRINT AT 18,12; PAPER 3;"
"
1250 IF INKEY$="" THEN GO TO 12
50
1260 IF INKEY$="A" THEN LET I(J
,F)=1: LET I$="ACIERTO": GO TO 1
295
1270 IF INKEY$="E" THEN LET I(J
,F)=-1: LET I$="ERROR ": GO TO
1295
1280 IF INKEY$="N" THEN LET I(J
,F)=0: LET I$="NULO ": GO TO 1
295
1290 GO TO 1250
1295 PRINT AT 18,12; PAPER 8; IN
K 6;I$: BEEP .3,(CODE I$)/3
1300 NEXT F: NEXT J
1305 IF D=2 THEN GO TO 8720
1310 PRINT AT 18,0; FLASH 1; PAP
ER 8;" PREPARA CAFE MIENTRAS CAL
CULO "
1320 FOR F=1 TO 400: NEXT F
1400 REM ***** PARAMETROS *****
1410 PAPER 6; INK 0; BORDER 6; C
LS
1420 PRINT INVERSE 1;" P A
R A M E T R O S "
1430 PRINT AT 3,1;" S = SUJETO";
AT 3,16;" I = ITEM";AT 4,1;" RP =
RESP. POS.POR ITEM"
1440 PRINT AT 6,0; PAPER 3; INK
0; BRIGHT 1;" F I A B I L
I D A D " : PRINT " A = Aci
ertos,E=Errores,N=Nulos": PRINT
" P = Puntuabilidad": PRINT " Z
= Puntuabilidad TIPIFICADA"
1445 PRINT " R = Coeficiente de
Correlacion": PRINT "CF = Coefi
ciente de Fiabilidad"
1450 PRINT " PAPER 3; INK 0; BRI
GHT 1;" V A L I D E Z
": PRINT " I.F. = Indice
de Facilidad": PRINT " I.D. = I
ndice de Discriminacion": PRINT
"I.D.M. = I.D. Maximo": PRINT " I
.E. = Indice de Eficacia"
1500 REM ***** CALC. PUNT. *****
1510 LET J=0
1520 LET J=J+1: LET F=0
1530 LET F=F+1
1540 IF I(J,F)=1 THEN LET A(J)=
A(J)+1
1550 IF I(J,F)=-1 THEN LET E(J)
=E(J)+1
1560 IF I(J,F)=0 THEN LET N(J)=
N(J)+1
1570 IF J=S THEN IF F=I THEN L
ET J=0: GO TO 1710
1580 IF F=I THEN GO TO 1520
1590 GO TO 1530
1710 LET J=J+1
1720 LET P(J)=A(J)-(E(J)/(RP-1))
1730 IF J=S THEN LET J=0: GO TO
2010
1735 GO TO 1710
2000 REM ** CALC. DE Z **
2010 LET PRM=0: LET DIF=0
2020 LET J=J+1
2030 LET PRM=PRM+P(J)
2035 IF J=S THEN LET J=0: GO TO
2050

```

```

INDICES DE VALIDEZ
No. DE SUJETOS = 50
No. DE ITEMS = 40
RESPUESTAS POSIBLES POR ITEM = 4
Los Indices de Validez se dan
para cada Item y son :
I.F. = Indice de FACILIDAD (%)
I.D. = Indice de DISCRIMINACION(%)
I.D.M. = I.D. MAXIMO (%)
I.E. = Indice de EFICACIA

```

```

2040 GO TO 2020
2050 LET PUM=PRM/S
2070 LET J=J+1
2080 LET DIF=DIF+((P(J)-PUM)*(P(
J)-PUM))
2085 IF J=S THEN LET J=0: GO TO
2210
2090 GO TO 2070
2210 LET SIGMA=SQR (DIF/S)
2215 IF SIGMA=0 THEN LET SIGMA=
1E-33
2220 LET J=J+1
2230 LET Z(J)=(P(J)-PUM)/SIGMA
2235 IF J=S THEN LET J=0: GO TO
3050
2240 GO TO 2220
3000 REM ** CALC. PUNT. SERIES**
3050 LET J=J+1: LET F=-1
3060 LET F=F+2: IF F>I THEN GO
TO 3095
3070 IF I(J,F)=1 THEN LET X(J)=
X(J)+1
3080 IF I(J,F)=-1 THEN LET Y(J)
=Y(J)+1
3090 IF I(J,F)=0 THEN LET S(J)=
S(J)+1
3095 IF J=S THEN IF F>I THEN
LET J=0: GO TO 3120
3100 IF F>I THEN GO TO 3050
3105 GO TO 3060
3120 LET J=J+1: LET F=0
3125 LET F=F+2: IF F>I THEN GO
TO 3150
3130 IF I(J,F)=1 THEN LET V(J)=
V(J)+1
3140 IF I(J,F)=-1 THEN LET W(J)
=W(J)+1
3150 IF J=S THEN IF F>I THEN
LET J=0: GO TO 3230
3155 IF F>I THEN GO TO 3120
3160 GO TO 3125
3230 LET J=J+1
3240 LET Q(J)=X(J)-(Y(J)/(RP-1))
3250 LET R(J)=V(J)-(W(J)/(RP-1))
3260 IF J=S THEN LET J=0: GO TO
3430
3265 GO TO 3230
3400 REM *** CALC. Z SERIES ****
3430 LET PRQ=0: LET PRR=0: LET D
IFQ=0: LET DIFR=0
3440 LET J=J+1
3450 LET PRQ=PRQ+Q(J): LET PRR=P
RR+R(J)
3460 IF J=S THEN LET J=0: GO TO
3500
3470 GO TO 3440
3500 LET PUMQ=PRQ/S: LET PUMR=PR
R/S
3510 LET J=J+1
3520 LET DIFQ=DIFQ+((Q(J)-PUMQ)*
(Q(J)-PUMQ))
3530 LET DIFR=DIFR+((R(J)-PUMR)*
(R(J)-PUMR))
3540 IF J=S THEN LET J=0: GO TO
3610
3545 GO TO 3510
3610 LET SIGI=SQR (DIFQ/S): LET
SIGP=SQR (DIFR/S)

```

```

3620 IF SIGI=0 THEN LET SIGI=1E
-33
3630 IF SIGP=0 THEN LET SIGP=1E
-33
3640 LET J=J+1
3660 LET M(J)=(Q(J)-PUMQ)/SIGI:
LET T(J)=(R(J)-PUMR)/SIGP
3670 IF J=S THEN LET J=0: GO TO
3840
3675 GO TO 3640
3800 REM ** CALC. COEF. CORR. **
3840 LET PCC=0
3850 LET J=J+1
3860 LET PCC=PCC+(M(J)*T(J))
3865 IF J=S THEN GO TO 3880
3870 GO TO 3850
3880 LET R=PCC/S
3885 IF R=-1 THEN LET CF=-1E36:
GO TO 4000
3890 LET CF=(2*R)/(1+R)
4000 REM ** ORDENA PUNTUAB. **
4010 DIM F(S): DIM G(S)
4020 FOR J=1 TO S: LET F(J)=P(J)
: NEXT J
4030 LET J=1: LET C=0
4040 IF F(J)<F(J+1) THEN LET D=
F(J): LET F(J)=F(J+1): LET F(J+1
)=D: LET C=C+1
4050 LET J=J+1
4055 IF J=S THEN IF C=0 THEN L
ET J=0: GO TO 4085
4060 IF J=S THEN GO TO 4030
4070 GO TO 4040
4085 LET J=J+1: LET F=0
4090 LET F=F+1
4100 IF F(J)=P(F) THEN GO TO 41
30
4101 GO TO 4105
4102 LET G(J)=F
4105 IF J=S THEN IF F=S THEN G
O TO 4150
4110 IF F=S THEN GO TO 4085
4120 GO TO 4090
4130 FOR K=1 TO J-1: IF G(K)=F T
HEN GO TO 4105
4140 NEXT K: GO TO 4102
4150 REM *27% S.con M.y m.PUNT.*
4160 FOR J=1 TO SF: LET H(J)=G(J
): NEXT J
4165 LET K=1
4170 FOR J=S TO S-SF+1 STEP -1:
LET L(K)=G(J): LET K=K+1: NEXT J
4300 REM ** CALC. IND. VAL. **
4310 DIM F(I): DIM G(I)
4320 LET F=0
4325 LET F=F+1: LET K=0
4330 LET K=K+1
4340 IF (I(H(K),F)=1) THEN LET
B(F)=B(F)+1
4350 IF (I(L(K),F)=1) THEN LET
C(F)=C(F)+1
4353 IF I(H(K),F)=-1 THEN LET O
(F)=O(F)+1
4356 IF I(L(K),F)=-1 THEN LET O
(F)=O(F)+1
4360 IF F=I THEN IF K=SF THEN
LET F=0: GO TO 4380
4365 IF K=SF THEN GO TO 4325
4370 GO TO 4330
4380 LET F=F+1
4390 LET F(F)=50*(B(F)+C(F))/SF
4400 LET G(F)=50*(B(F)-C(F))/SF
4410 LET J(F)=2*(F)
4420 IF J(F)=0 THEN LET K(F)=1E
33: GO TO 4435
4430 LET K(F)=G(F)/J(F)
4435 IF F=I THEN GO TO 5010
4440 GO TO 4380
5000 REM ** M E N U **
** P R I N C I P A L **
*****
5010 POKE 23658,8
5020 INK 7: BRIGHT 0: PAPER 1: B
ORDER 1: CLS
5030 FOR F=0 TO 2: PRINT AT F,0;
PAPER 2;"

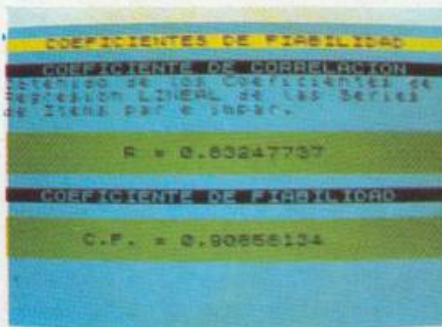
```



```

6710 PAPER 3: INK 7: BORDER 3: C
LS
6720 PRINT PAPER 6; INK 0; "
L I S T A D O S "
6730 PRINT AT 5,0; " Listado por
SUJETO "; FLASH 1; "< S >"; FLASH
0; AT 7,0; " Listado por ITEM "
; INVERSE 1; FLASH 1; "< I >"; #0;
FLASH 0; " < M > = M E N
U"
6740 IF INKEY$="S" THEN GO TO 6
770
6745 IF INKEY$="M" THEN GO TO 5
510
6750 IF INKEY$="I" THEN GO TO 6
900
6760 GO TO 6740
6770 PRINT AT 3,3; "27% de Sujeto
s = "; SF; " "; FLASH 1; ">"; FLAS
H 0; AT 5,19; "(1-"; S; " ) "; AT 7,0;
PAPER 2; " Desde el SUJETO No.
"; LET C=6795
6780 INPUT LINE I$
6785 GO SUB 1135: IF D=1 THEN L
ET D=2: GO TO 6780
6790 GO TO C
6795 LET U=VAL I$: PRINT AT 7,22
; PAPER 2; BRIGHT 1; U: IF U>S OR
U<1 THEN GO TO 6770
6800 PRINT AT 9,0; PAPER 2; " Has
ta el SUJETO No. "; LET C=6
805: GO TO 6780
6805 LET H=VAL I$: PRINT AT 9,22
; PAPER 2; BRIGHT 1; H: IF H>S OR
H<1 THEN GO TO 6800
6808 LET D=6835
6810 IF H<U THEN GO TO 6700
6813 POKE 23658,8
6814 FOR F=15 TO 21: PRINT AT F,
0; PAPER 1; BRIGHT 1; "
"; NEXT F
6815 PRINT AT 13,7; PAPER 4; INK
0; BRIGHT 1; " C O M A N D O S "
6820 PRINT AT 16,0; PAPER 8; INK
6; " < A > para Adelante
"
6822 PRINT " PAPER 8; INK 6; " <
B > para Atras
6824 PRINT " PAPER 8; INK 6; " <
M > para volver al M E N U "
6826 PRINT #0; INK 0; " Pu
lsa una tecla "
6828 IF INKEY$="" THEN GO TO 68
28
6830 GO TO D
6835 PAPER 1: BORDER 1: FOR J=U
TO H
6840 CLS : PRINT AT 0,8; INVERSE
1; " SUJETO No."; J; " "
6845 PRINT " A="; A(J); "
"; "E="; E(J); " "; "N="; N(
J)
6847 PRINT " Puntuabilidad = ";
P(J); " "
6849 PRINT " Punt. MEDIA = "; PU
M
6851 PRINT " Z = "; Z(J)
6852 PRINT AT 8,17; LET D=Z(J):
GO SUB 6853: GO TO 6865
6853 IF D<0 THEN PRINT INVERSE
1; INK 2; PAPER 7; " SUSPENSO
"
6855 IF D>=0 AND D<.5 THEN PRIN
T INVERSE 1; INK 5; " APROBAD
O"
6857 IF D>=.5 AND D<1 THEN PRIN
T INVERSE 1; INK 5; " BIEN"
6859 IF D>=1 AND D<2 THEN PRINT
INVERSE 1; INK 5; " NOTABLE"
"
6861 IF D>=2 AND D<2.5 THEN PRI
NT INVERSE 1; INK 5; " SOBRESALI
ENTE"
6863 IF D>2.5 THEN PRINT PAPER
6; INK 0; " SOBRESALIENTE"
6864 RETURN
6865 PRINT " PAPER 3; INK 0; BRI

```



```

GHT 1; "FOR SERIES DE ITEMS(IMP
AR Y PAR)"
6868 PRINT "Ai="; X(J), "Ap="; V(J)
6869 PRINT "Ei="; Y(J), "Ep="; W(J)
6870 PRINT "Ni="; S(J), "Np="; N(J)
-S(J)
6873 PRINT "Pi="; Q(J), "Pp="; R(J)
; " "
6876 PRINT "Pmi="; PUMQ, "Pmp="; P
UMR
6879 PRINT "Zi="; M(J), "Zp="; T(J)
)
6880 LET D=M(J): GO SUB 6853: LE
T D=T(J): GO SUB 6853
6882 PRINT AT 21,0; FOR K=1 TO
SF: IF H(K)=J THEN PRINT PAPER
7; INK 2; " Sujeto "; INVERSE 1;
" "; K; " "; INVERSE 0; " con Mayor
Puntuab."; TAB 31; " "
6883 POKE 23692,-1
6885 IF L(K)=J THEN PRINT PAPE
R 7; INK 2; " Sujeto "; INVERSE 1
; " "; K; " "; INVERSE 0; " con Meno
r Puntuab."; TAB 31; " "
6888 NEXT K: IF SCREEN$ (21,1)<>
"S" THEN PRINT AT 21,0; PAPER 7
; INK 2; " SIN COMENTARIO
"; TAB 31; " "
6890 BEEP .2,10
6891 PRINT #0; PAPER 4; INK 0; B
RIGHT 1; "Pulsa un COMANDO o "; I
NVERSE 1; "< C >"; INVERSE 0; " para
COPY"
6894 IF INKEY$="M" THEN GO TO 5
510
6895 IF INKEY$="A" AND (J>=H) TH
EN PRINT #0; FLASH 1; " F I N
D E L L I S T A D O "; FOR F=
1 TO 50: NEXT F: GO TO 6700
6896 IF INKEY$="B" AND (J>U) THE
N LET J=J-1: GO TO 6840
6897 IF INKEY$="A" AND J<H THEN
NEXT J
6898 IF INKEY$="C" THEN COPY
6899 GO TO 6894
6900 REM ** LISTADO POR ITEMS**
6910 PRINT AT 5,0; " 27% de Su
jetos = "; SF; " "; AT 7,19; "(1-"; I
; " ) "
6915 PRINT AT 9,0; PAPER 2; " Des
de el ITEM No. "; LET C=693
5
6920 INPUT LINE I$
6925 GO SUB 1135: IF D=1 THEN L
ET D=2: GO TO 6920
6930 GO TO C
6935 LET U=VAL I$: PRINT AT 9,20
; PAPER 2; BRIGHT 1; U: IF U>I OR
U<1 THEN GO TO 6910
6940 PRINT AT 11,0; PAPER 2; " Ha
sta el ITEM No. "; LET C=69
45: GO TO 6920
6945 LET H=VAL I$: PRINT AT 11,2
0; PAPER 2; BRIGHT 1; H: IF H>I O
R H<1 THEN GO TO 6940
6950 LET D=6960: POKE 23658,8
6955 GO TO 6810
6960 PAPER 1: BORDER 1: FOR F=U

```

```

TO H
6970 CLS : PRINT INVERSE 1; TAB
31; " "; AT 0,11; PAPER 8; "ITEM No
."; F
6972 FOR K=1 TO 3: PRINT PAPER
5; TAB 31; " "; NEXT K: FOR K=1 TO
3: PRINT PAPER 6; TAB 31; " "; N
EXT K: FOR K=1 TO 3: PRINT PAPE
R 3; TAB 31; " "; NEXT K
6973 PRINT AT 10,0; FOR K=1 TO
6: PRINT PAPER 4; TAB 31; " "; NE
XT K: FOR K=1 TO 6: PRINT PAPER
2; TAB 31; " "; NEXT K: PRINT AT
1,0;
6974 PRINT PAPER 8; INK 0; "Aci
ertos del 27% de Sujetos con Mayo
r Puntuabilidad = "; B(F); " = As
"; PRINT INK 5; BRIGHT 1; "Suj. N
o."; FOR K=1 TO SF: IF I(H(K),F
)=1 THEN PRINT INK 5; BRIGHT 1
; " "; H(K);
6975 NEXT K
6976 PRINT " PAPER 8; INK 1; "Aci
ertos del 27% de Sujetos con Men
or Puntuabilidad = "; C(F); " = Ai
"; PRINT INK 5; BRIGHT 1; "Suj.
No."; FOR K=1 TO SF: IF I(L(K),F
)=1 THEN PRINT INK 5; BRIGHT
1; " "; L(K);
6977 NEXT K: PRINT " PAPER 8; "Er
rores del conjunto de Sujetos an
teriores = "; O(F); " "; PRINT IN
K 5; BRIGHT 1; "Suj.No."; FOR K=
1 TO SF: IF I(H(K),F)=-1 THEN P
RINT INK 5; BRIGHT 1; " "; H(K);
6978 IF I(L(K),F)=-1 THEN PRINT
INK 5; BRIGHT 1; " "; L(K);
6979 NEXT K
6980 PRINT " "; PAPER 8; INK 0; " I
.F. = "; F(F); " %"
6982 PRINT PAPER 8; INK 0; " I.D
. = "; G(F); " %"
6984 PRINT PAPER 8; INK 0; " I.D
.M. = "; J(F); " %"
6986 PRINT PAPER 8; INK 0; " I.E
. = "; K(F); " %"
6987 PRINT AT 16,0; PAPER 8; "Ace
rtaron"; FOR J=1 TO S: IF I(J,
F)=1 THEN PRINT PAPER 8; J; " ";
6988 NEXT J: PRINT CHR$ 8; PAPER
8; " "; PRINT PAPER 8; " Erraro
n"; FOR J=1 TO S: IF I(J,F)=-1
THEN PRINT PAPER 8; J; " ";
6989 NEXT J: PRINT CHR$ 8; PAPER
8; " "; PRINT PAPER 8; " Hicier
on Nulos"; FOR J=1 TO S: POKE
23692,-1: IF I(J,F)=0 THEN PRIN
T PAPER 8; J; " ";
6990 NEXT J: PRINT CHR$ 8; PAPER
8; " "; BEEP .2,10
6991 PRINT #0; PAPER 4; INK 0; B
RIGHT 1; "Pulsa un COMANDO o "; I
NVERSE 1; "< C >"; INVERSE 0; " para
COPY"
6992 IF INKEY$="M" THEN GO TO 5
510
6994 IF INKEY$="C" THEN COPY
6995 IF INKEY$="A" AND (F>=H) TH
EN PRINT #0; FLASH 1; " F I N
D E L L I S T A D O "; FOR F=
1 TO 50: NEXT F: GO TO 6700
6996 IF INKEY$="B" AND (F>U) THE
N LET F=F-1: GO TO 6970
6997 IF INKEY$="A" AND F<H THEN
NEXT F
6999 GO TO 6992
7800 REM **** CORR. Y FIAB. ****
7810 PAPER 5: BORDER 5: INK 0: C
LS : PRINT " PAPER 6; " COEFICI
ENTES DE FIABILIDAD "
7820 PRINT " INVERSE 1; " COEFI
CIENTE DE CORRELACION "
7830 PRINT "Obtenido de los Coef
icientes de Regresion LINEAL de
las Series de Items par e impar
."
```


No se caliente la "CABEZA"

SEIKOSHA

IMPRESORAS



Nuestra calidad es "SEIKO";
nuestros precios, únicos.
Si desea más información,
consulte con nuestro distribuidor
más cercano, o llame o escriba a:

DiRAC S.L.

Dirección comercial:
Av. Blasco Ibáñez, 114-116.
46022-Valencia.
Tel. (96) 372 88 89.
Télex 62220

Delegación en Cataluña:
C/ Muntaner, 60, 4, 1.
08011-Barcelona.
Tel. (93) 323 32 19.

ESTOS SON NUESTROS MODELOS:

Modelo	Velocidad	Columnas	Tipos de letra	Interface	P.V.P.
GP-50	40 cps	46	2	A-Paralelo AS-Serial S-Spectrum	A-25.900 AS-29.900 S-28.900
GP-500	50 cps	80	2	A-Paralelo AS-Serial	A-47.900 AS-49.900
GP-550	86 cps	80-136	18	A-Paralelo	A-59.900
GP-700	50 cps	80-106	3	A-Paralelo	A-89.900
BP-5200	200 cps	136-272	18	Paralelo y serial	199.000
BP-5420	420 cps	136-272	18	Paralelo y serial I-IBM PC	299.000 I-299.000

Disponemos de interfaces opcionales para todos los modelos: IBM PC, COMMODORE 64, ZX SPECTRUM, ATARI, DRAGON 64, SHARP MZ 700, SPECTRAVIDEO, NEW BRAIN, APPLE, ETC...

FIABILIDAD

A = Aciertos, E=Errores, N=Nulos
P = Puntuabilidad
Z = Puntuabilidad TIPIFICADA

R = Coeficiente de Correlacion
CF = Coeficiente de Fiabilidad

VALIDEZ

I.F. = Indice de Facilidad
I.D. = Indice de Discriminacion
I.D.M. = I.D. Maximo
I.E. = Indice de Eficacia

7840 FOR F=8 TO 10: PRINT AT F,0
; PAPER 4;"

" : NEXT F

7850 PRINT AT 9,8; PAPER 8;" R =
"R;" "

7860 PRINT AT 12,0; INVERSE 1;"
COEFICIENTE DE FIABILIDAD "

7870 FOR F=14 TO 16: PRINT AT F,
0; PAPER 4;"

" : NEXT F

7880 PRINT AT 15,5; PAPER 8;" C.
F. = ";CF;" "

7900 PRINT #0; PAPER 2; INK 7;"P
ulsa:<C> para COPY,<M>=M E N U "

7910 IF INKEY#="" THEN GO TO 79
10

7920 IF INKEY#="M" THEN GO TO 5
000

7930 IF INKEY#="C" THEN COPY

7940 GO TO 7910

8200 REM **** IND. DE VAL. ****

8210 PAPER 1: BORDER 1; INK 7; C
LS : PRINT PAPER 6; INK 0;"I N
D I C E S D E V A L I D E Z"

8220 FOR F=12 TO 20: PRINT AT F,
0; PAPER 5;"

" : NEXT F

8230 PRINT AT 3,3; INK 7;" No. D
E SUJETOS = ";S;" ";AT 5,3;" No.
DE ITEMS = ";I;" ";AT 7,0;"RE
SPUESTAS POSIBLES POR ITEM= ";RP

8240 PRINT AT 9,0; INK 6;" Los I
ndices de Validez se dan para
cada item y son : "

8250 PRINT " "I.F. = Indice de F
ACILIDAD (%)"

8260 PRINT " "I.D.=Indice de DISC
RIMINACION(%)"

8270 PRINT " "I.D.M. = I.D. MAXIM
O (%)"

8280 PRINT " "I.E. = Indice de EF
ICACIA "

8290 PRINT #0; INK 6; BRIGHT 1;"
Pulsa una tecla

"

8295 LET RUTA=5; OPEN #5,"S": LE
T D=8310; LET U=158; LET C=32

8300 IF INKEY#="" THEN GO TO 83
00

8310 PAPER 5: PAPER 5: INK 0; BO
RDER 5; CLS : PRINT #RUTA; PAPER
6;"I N D I C E S D E V A L I D
E Z"

8320 FOR F=1 TO I

8325 POKE 23692,-1

8330 PRINT #RUTA; PRINT #RUTA; I
NVERSE 1; BRIGHT 1;"ITEM No.:";F;
PRINT #RUTA; PAPER 2; INK 7;"I.
F.=";INT (F(F)*1E4)/1E4;" %";TAB
18;"I.D.=";INT (G(F)*1E4)/1E4;T
AB 31;"%"; PRINT #RUTA; PAPER 2;
INK 7;"I.D.M.=";INT (J(F)*1E4)/
1E4;" %";TAB 18;"I.E.=";INT (K(F)
)*1E4)/1E4;TAB 31;" "

8340 GO TO 9810

8350 NEXT F

8360 GO TO 9700

8600 REM *** CAMB. DE DATOS ***

8610 CLS : PRINT " PAPER 6; INK
0;" C A M B I O D E D A T O
S "

8615 LET U=167; LET C=33; LET K=
5; GO TO 9850; PRINT AT 5,3;"

8620 PRINT AT 7,0; INVERSE 1;" S
UJETO No."

8630 INPUT "Introduce el No. del
Sujeto (1-";(S);") " ; LI
NE I#; LET D=2

8640 GO SUB 1135; IF D=1 THEN G
O TO 8630

8650 IF VAL I#>S OR VAL I#<1 THE
N GO TO 8630

8660 PRINT INVERSE 1;AT 7,11;VA
L I#;" "

8670 GO SUB 1145; IF D=1 THEN G
O TO 8690

8680 GO TO 8630

8690 LET U=VAL I#; LET H=U

8710 PRINT AT 7,0;"
" : LET D=2; GO TO 1171

8720 PRINT #0; PAPER 6; INK 0;"
QUIERES CAMBIAR OTRO S/N ? "

8730 FOR F=1 TO 50: NEXT F

8740 IF INKEY#="S" THEN CLS : G
O TO 8620

8750 IF INKEY#="N" THEN GO TO 1
310

8760 GO TO 8740

9000 REM **** GRABAR DATOS ****

9010 PAPER 2: INK 7: BORDER 2: C
LS

9020 PRINT INVERSE 1;" G R A
B A R D A T O S "

9030 LET U=105; LET C=35; LET K=
1; GO TO 9850

9065 PRINT AT 5,12;" : SI "

9070 DIM D(3); DIM U\$(1,LEN R#)

9080 RESTORE 9100; FOR F=1 TO 3:
READ D: LET D(F)=D: NEXT F

9090 LET U\$(1)=R#

9100 DATA S,I,RP

9110 PRINT AT 7,0; PAPER 5; INK
0;" NOMBRE DEL FICHERO "; FLASH
1;"?"; PRINT AT 21,0;" 10 LETRAS
MAXIMO "; PRINT AT 9,3;

9120 INPUT LINE I#; IF CODE I#=
0 THEN GO TO 9120

9121 IF LEN I#>10 THEN GO TO 91
20

9122 PRINT INVERSE 1;I#;AT 7,20
; INK 2;" "

9125 POKE 23658,8

9130 SAVE I# DATA D(); SAVE I# D
ATA U\$(); SAVE I# DATA I()

9140 CLS : PRINT INVERSE 1;"G R
A B A C I O N D E D A T O S";
PRINT AT 10,0;"QUIERES VERIFICA
R LOS DATOS "; FLASH 1;"S"; FLAS
H 0;" / "; INVERSE 1; FLASH 1;"N"

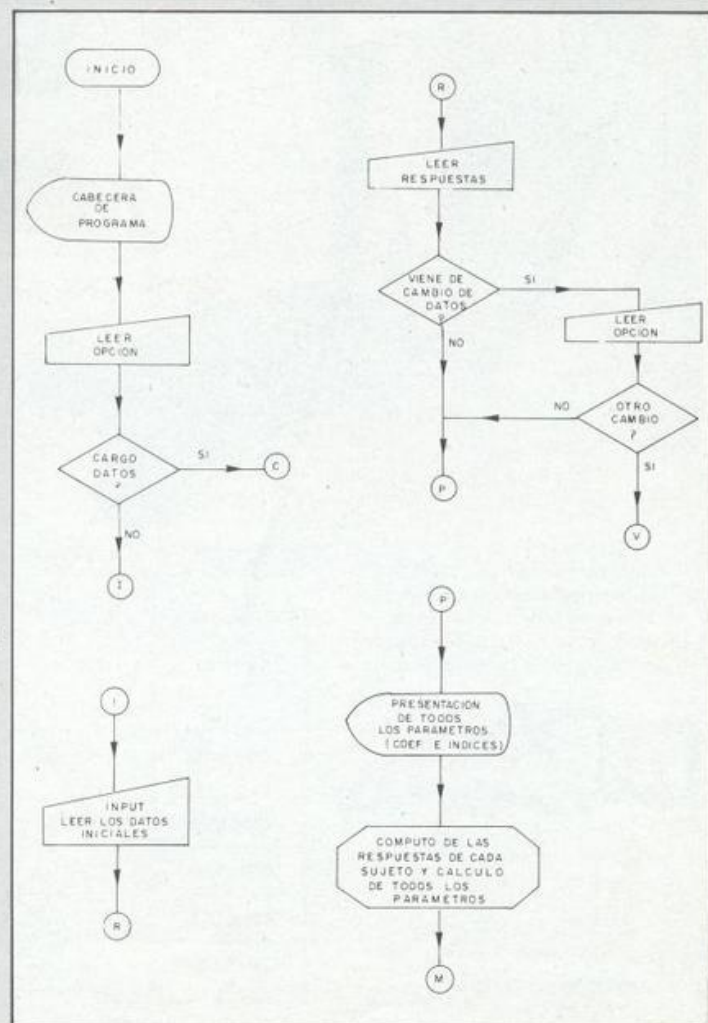
9150 IF INKEY#="S" THEN GO TO 9
170

9155 IF INKEY#="N" THEN GO TO 5
020

9160 GO TO 9150

9170 PRINT AT 10,27; INVERSE 1;"
:SI "

Organigrama de operación




```

9180 PRINT AT 12,0;"SI APARECIES
E UN CODIGO DE ERROR DURANTE LA
VERIFICACION, PULSA: "; INVERSE
1;" GOTO 5000"
9190 PRINT AT 20,0; PAPER 5; INK
1;" SI HAS REBOBINADO LA CINTA
, PON EN MARCHA EL CASSETTE
"
9200 VERIFY I$ DATA D(): VERIFY
I$ DATA U$(): VERIFY I$ DATA I()
9210 CLS : PRINT AT 11,0; PAPER
1; FLASH 1;" LOS DATOS ES
TAN PERFECTAMENTE G
RABADOS " : PAUSE 200
9220 GO TO 5020
9400 REM **** CARGAR DATOS ****
9410 PAPER 4: INK 0: BORDER 4: C
LS
9420 PRINT INVERSE 1; BRIGHT 1;
" C A R G A R D A T O S
"
9440 LET U=234: LET C=36: LET K=
1: GO TO 9850
9450 CLEAR : DIM U$(1,18): DIM D
(3)
9455 PRINT INVERSE 1; BRIGHT 1;
" C A R G A R D A T O S
": PRINT AT 3,3;"
"
9460 PRINT AT 7,0;" NOMBRE DEL F
ICHERO " : FLASH 1;"?"
9470 INPUT LINE I$: PRINT AT 7,
21;" : AT 9,2; INVERSE 1; BRIGHT
1;I$
9475 POKE 23658,8
9480 PRINT AT 15,0; PAPER 1; INK
7;"SI APARECIESE UN CODIGO DE E

```

```

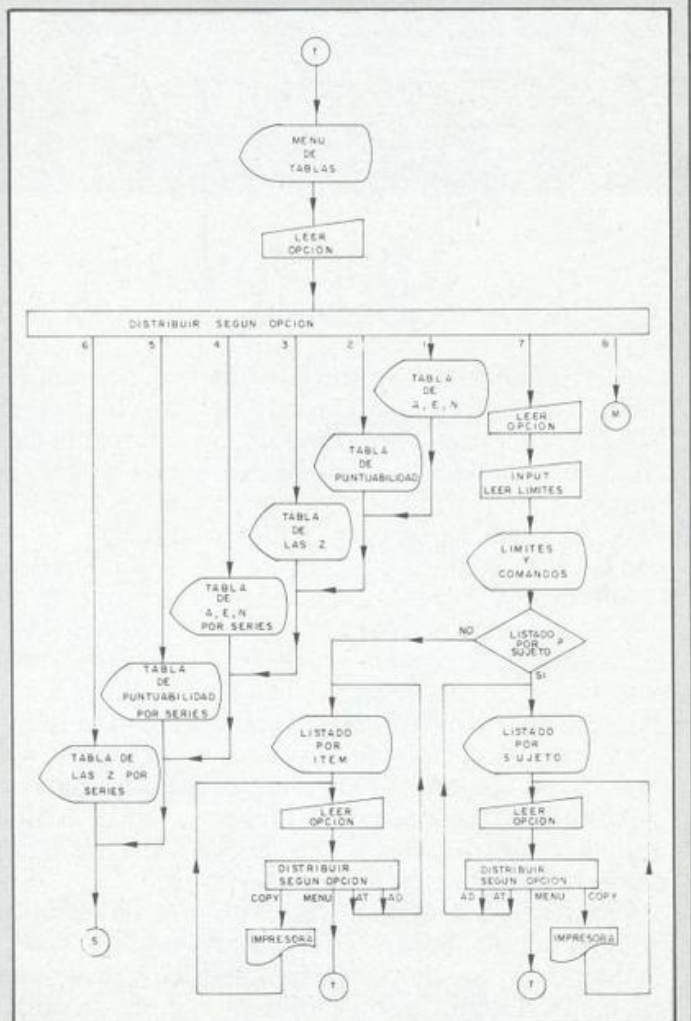
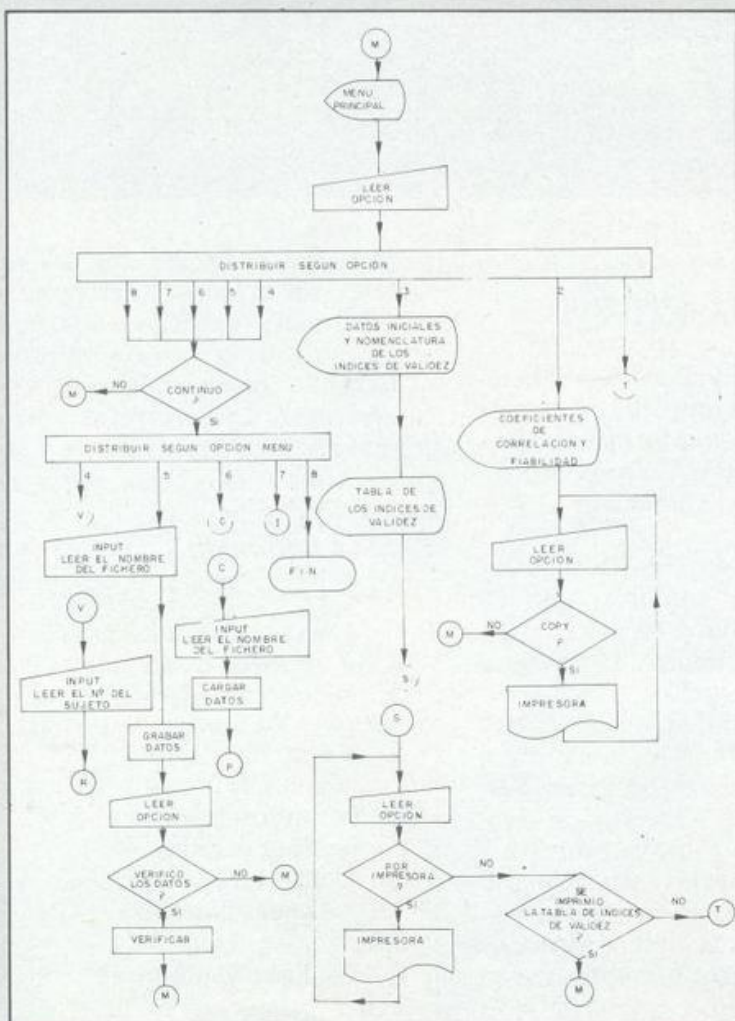
RROR DURANTE LA CARGA , PULSA :
"; INVERSE 1;" GOTO 9450 "
9490 PRINT AT 21,4;"PON EN MARCH
A EL CASSETTE"
9500 LOAD I$ DATA D(): LOAD I$ D
ATA U$(): LOAD I$ DATA I()
9510 CLS : PRINT AT 10,0; PAPER
3; INK 7;" C A R G A C O R R
E C T A "
9520 LET S=D(1): LET I=D(2): LET
RP=D(3): LET R$=U$(1)
9530 PAUSE 100
9540 LET D=9450
9550 GO SUB 9900
9560 GO TO 1171
9700 REM "OFCION:MENU-IMPRES."
9710 PRINT #0; BRIGHT 1; PAPER 4
; INK 0;"Pulsa:" : FLASH 1;"<M>";
FLASH 0;"=" : PAPER 5;" MENU " :
PAPER 4;" " : INVERSE 1; FLASH 1;
"<I>"; FLASH 0; INVERSE 0;"=" : P
APER 6;" IMPRESORA "
9720 IF INKEY$="M" AND D=8310 TH
EN CLOSE #RUTA: GO TO 5010
9730 IF INKEY$="M" AND D<>8310 T
HEN CLOSE #RUTA: GO TO 5510
9740 IF INKEY$="I" THEN LET RUT
A=4: CLOSE #5: OPEN #4,"P": GO T
O D
9750 GO TO 9720
9800 REM *SUB. DE PARADA EN T.**
9810 IF INKEY$="" THEN POKE 236
18,U: POKE 23619,C: POKE 23620,1
9815 BEEP .5,0
9820 IF INKEY$="" THEN GO TO 98
20
9825 IF INKEY$="M" AND D=8310 TH
EN CLOSE #RUTA: GO TO 5010

```

```

9830 IF INKEY$="M" AND D<>8310 T
HEN CLOSE #RUTA: GO TO 5510
9835 BEEP .5,0
9840 POKE 23618,U: POKE 23619,C:
POKE 23620,1
9850 REM *** SUB. DE VERIF. ***
9860 PRINT AT 5,2;" CONTINUO " :
FLASH 1;"S"; FLASH 0;" / "; INVER
SE 1; FLASH 1;"N"
9870 IF INKEY$="N" THEN GO TO 5
020
9880 IF INKEY$="S" THEN POKE 23
618,U: POKE 23619,C: POKE 23620,
K
9890 GO TO 9870
9900 REM ** 27% DE SUJETOS **
9910 LET SR=27*S/100: LET SE=INT
(27*S/100)
9915 IF SR=SE THEN LET SF=SE: G
O TO 9925
9920 LET SF=SE+1
9925 IF D=9450 THEN RETURN
9930 REM *SUB. DE LONG.VARIABLES*
9935 LET U=65000-(PEEK 23627+256
*PEEK 23628)
9940 IF VAL I$>S THEN GO TO 996
0
9945 LET H=INT ((U-439-80*S-10*S
F)/(25+5*S))
9950 IF VAL I$>INT H THEN LET J
=J+1: RETURN
9955 LET J=0: RETURN
9960 LET H=INT ((U-439-70*S-10*S
F)/(35+5*S))
9965 IF VAL I$>(INT H) THEN LET
J=J+1: RETURN
9970 LET J=0: RETURN

```



SE LE ACABO LA PISTA



Simulador de vuelo

A estas alturas —nunca mejor dicho— seguro que ya dispone de un programa simulador de vuelo, o en el peor de los casos, estará examinando los distintos tipos de “vuelos” que se pueden realizar con su Spectrum. El programa que le ofrecemos tiene la particularidad de correr en el Spectrum de 16 K, gracias a la separación de instrucciones en dos programas. El primero define los caracteres gráficos e informa acerca de la utilización de su aparato. El segundo es el simulador propiamente dicho, que le permitirá llegar tan lejos como su imaginación le permita.

Comienza el programa “en el aire”, por lo que habrá de concentrar su atención en el aterrizaje. Inicialmente tomará tierra fácilmente,

pero si es amante de las emociones fuertes, el completo cuadro de control le dará la información precisa en todo momento para controlar el avión... o intentarlo al menos.

¿Quiere una ayudita? Relájese y desconecte dos motores, uno de cada lado del avión (de otra forma perdería estabilidad). Disminuya la altura a 100 metros, a una velocidad comprendida entre 301 y 550 km/h. Cuando se encuentre a 2 km. de la cabeza de pista, descienda a 30 m. y recorra 3 km. Después baje a 5 m. y recorra 2,5 km. Una vez hecho esto, baje a 0 m. y disminuya la velocidad parando los motores. Si recordó sacar el tren de aterrizaje, habrá conseguido su objetivo.

Finalmente indicar que los caracteres subrayados corresponden a la definición gráfica creada en el primer programa (por ejemplo A = A en modo gráfico). Habrá de ejecutarse solamente el primer programa, desde el cual se llama al segundo (línea 9999) que habrá de estar grabado en cinta.

Roberto Lacámara

16 K

CUADRO DE MANDOS

- C Quitar piloto automático.
- K Aumentar velocidad.
- J Disminuir velocidad.
- SPACE Estabilizar velocidad.
- 6 Aumentar altura.
- 7 Disminuir altura.
- 5 Inclinar izquierda.
- 8 Inclinar derecha.
- T Control tren aterrizaje.
- 1,2 Apagar motores.
- 3,4
- Q,W Encender motores
- E,R

Programas

PROGRAMA 1

```

10 REM *grafic.-presentacion*
20 REM * Roberto Lacamara *
100 POKE USR "a"+0,BIN 00000001
110 POKE USR "a"+1,BIN 00000010
120 POKE USR "a"+2,BIN 0
130 POKE USR "a"+3,BIN 0
140 POKE USR "a"+4,BIN 0
150 POKE USR "a"+5,BIN 0
160 POKE USR "a"+6,BIN 0
170 POKE USR "a"+7,BIN 0
190 REM -----
200 POKE USR "b"+0,BIN 0
210 POKE USR "b"+1,BIN 0
220 POKE USR "b"+2,BIN 0
230 POKE USR "b"+3,BIN 0
240 POKE USR "b"+4,BIN 0
250 POKE USR "b"+5,BIN 0
260 POKE USR "b"+6,BIN 01000000
270 POKE USR "b"+7,BIN 10000000
290 REM -----
300 POKE USR "c"+0,BIN 10000000
310 POKE USR "c"+1,BIN 01000000
320 POKE USR "c"+2,BIN 0
330 POKE USR "c"+3,BIN 0
340 POKE USR "c"+4,BIN 0
350 POKE USR "c"+5,BIN 0
360 POKE USR "c"+6,BIN 0
370 POKE USR "c"+7,BIN 0
390 REM -----
400 POKE USR "d"+0,BIN 0
410 POKE USR "d"+1,BIN 0
420 POKE USR "d"+2,BIN 0
430 POKE USR "d"+3,BIN 0
440 POKE USR "d"+4,BIN 0
450 POKE USR "d"+5,BIN 0
460 POKE USR "d"+6,BIN 00000010
470 POKE USR "d"+7,BIN 00000001
490 REM -----
500 POKE USR "e"+0,BIN 00000001
510 POKE USR "e"+1,BIN 00000010
520 POKE USR "e"+2,BIN 00000100
530 POKE USR "e"+3,BIN 00001000
540 POKE USR "e"+4,BIN 00010000
550 POKE USR "e"+5,BIN 00100000
560 POKE USR "e"+6,BIN 01000000
570 POKE USR "e"+7,BIN 10000000
590 REM -----
600 POKE USR "f"+0,BIN 10000000
610 POKE USR "f"+1,BIN 01000000

```

```

620 POKE USR "f"+2,BIN 00100000
630 POKE USR "f"+3,BIN 00010000
640 POKE USR "f"+4,BIN 00001000
650 POKE USR "f"+5,BIN 00000100
660 POKE USR "f"+6,BIN 00000010
670 POKE USR "f"+7,BIN 00000001
690 REM -----
700 POKE USR "g"+0,BIN 0
710 POKE USR "g"+1,BIN 0
720 POKE USR "g"+2,BIN 0
730 POKE USR "g"+3,BIN 11111111
740 POKE USR "g"+4,BIN 0
750 POKE USR "g"+5,BIN 0
760 POKE USR "g"+6,BIN 0
770 POKE USR "g"+7,BIN 0
790 REM -----
800 POKE USR "h"+0,BIN 10000000
810 POKE USR "h"+1,BIN 01111100
820 POKE USR "h"+2,BIN 01000010
830 POKE USR "h"+3,BIN 01000010
840 POKE USR "h"+4,BIN 01000010
850 POKE USR "h"+5,BIN 01000010
860 POKE USR "h"+6,BIN 00111110
870 POKE USR "h"+7,BIN 00000001
890 REM -----
900 POKE USR "i"+0,BIN 00000001
910 POKE USR "i"+1,BIN 00111110
920 POKE USR "i"+2,BIN 01000010
930 POKE USR "i"+3,BIN 01000010
940 POKE USR "i"+4,BIN 01000010
950 POKE USR "i"+5,BIN 01000010
960 POKE USR "i"+6,BIN 01111100
970 POKE USR "i"+7,BIN 10000000
990 REM -----
1000 PAPER 7: BORDER 7: INK 0: C
LS
1005 PRINT AT 12,9;"NO PARE LA C
INTA": PAUSE 150: CLS
1010 PRINT AT 2,2;"Instrucciones
sobre el manejo";AT 3,2;"-----
";AT 5,4;
"C- Quitar piloto automatico.";A
T 6,4;"K- Aumentar velocidad.";A
T 7,4;"J- Disminuir velocidad.";
AT 8,0;"SPACE- Estabilizar veloc
idad.";AT 9,4;"6- Aumentar altur
a.";AT 10,4;"7- Disminuir altura
.";AT 11,4;"5- Inclinar izquierd
a."

```

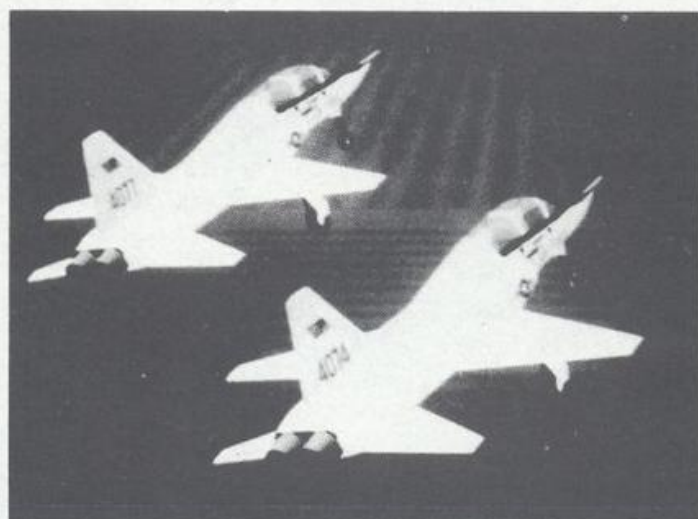


```
1020 PRINT AT 12,4;"8- Inclinar
derecha.";AT 13,4;"T- Controlar
tren de ater.";AT 14,4;"1\.";AT 1
5,4;"2 ! Apagar motores";AT 16,
4;"3 !";AT 17,4;"4/.";AT 18,4;"Q\
";AT 19,4;"W ! Encender motores
";AT 20,4;"E !";AT 21,4;"R/"
9999 LOAD "AIR RICH"
```

PROGRAMA 2

```
1000 REM *      air rich      *
1500 REM Roberto Lacamara*
2000 CLEAR : INK 6: PAPER 0: BOR
DER 3: CLS
2010 FOR x=0 TO 31: PRINT AT 0,x
;"■";AT 1,x;"■": NEXT x
2020 PLOT 0,30: DRAW 255,0
2040 PLOT 81,152: DRAW 85,0
2050 DRAW 0,-88: DRAW -85,0: DRA
W 0,87
2060 CIRCLE 123,108,42
2070 PLOT 83,30: DRAW 0,-30
2080 PLOT 170,30: DRAW 0,-30
2090 PLOT 180,154: DRAW 0,-110:
DRAW 70,0: DRAW 0,110: DRAW -70,
0
2100 PLOT 7,154: DRAW 0,-110: DR
AW 65,0: DRAW 0,110: DRAW -65,0
2110 PLOT 80,38: DRAW 92,0: DRAW
0,20: DRAW -92,0: DRAW 0,-20
2120 PLOT 7,115: DRAW 65,0
2130 PLOT 180,125: DRAW 70,0
2140 PLOT 180,65: DRAW 70,0
2150 PLOT 180,95: DRAW 70,0
2155 INK 7: PRINT AT 8,11;"
-";AT 8,19;" -": PRINT AT 9,2
1;"1";AT 7,21;"1";AT 5,21;"2";AT
11,21;"2"
2165 PRINT AT 19,2;"CONTROL";AT
15,10;"Long. PISTA";AT 19,11;"Ca
b. PISTA"
2170 PAPER 0: INK 7: PRINT AT 6,
5;"Km/h";AT 9,2;"Vel.";AT 12,2;"
Resi.";AT 11,5;"Km/h";AT 15,5;"g
/cm"
2175 PRINT AT 21,0;"Aut■ Manu■ "
;AT 16,20;"m";AT 21,20;"m";AT 21
,30;"sg";AT 5,30;"m"
2180 PRINT AT 15,29;"1";AT 9,23;
"On■ Off■"; INK 7: PAPER 2;AT 12
,23;"1";AT 12,25;"2";AT 12,27;"3
";AT 12,29;"4"
```

```
2190 INK 5: PRINT AT 8,13;"GGGG
";AT 7,15;"I"
2195 PAPER 1: INK 7: PRINT AT 3,
2;"VELO.";AT 8,2;"AIRE"
2200 PRINT AT 19,23;"TIEMPO";AT
11,23;"MOTORES";AT 14,23;"FUEL";
AT 7,23;"TREN ATE";AT 3,23;"ALTU
RA"
3000 LET yu=0: LET ti=0: LET pa=
0: LET vm=0: LET vn=0: LET v=750
: LET a=5000: LET m1=1: LET m2=1
: LET m3=1: LET m4=1: LET f=2000
: LET lp=7000: LET cp=30000: LET
ba=INT (RND*101)-50: LET i=2: L
ET t=0: LET h=0: LET vl=0: LET e
```



```
u=1: LET ev=1: LET m=0: LET er=0
: LET gj=0: LET cf=0: LET oo=0
3010 POKE 23674,0: POKE 23673,0:
POKE 23672,0
3020 LET a$=INKEY$: IF a$="c" TH
EN GO TO 3090
3030 GO SUB 3390
3035 LET ti=(PEEK 23672+256*PEEK
23673)/50
3040 IF oo=1 THEN GO TO 3070
3050 LET cp=cp-250: IF cp<=0 THE
N LET cp=0: PRINT AT 21,14;"0
": LET oo=1: GO TO 3080
3060 IF cp>0 THEN GO TO 3080
3070 LET lp=lp-250
3080 INK 7: PAPER 2: PRINT AT 5,
2;v;" ";AT 5,23;a;" ";AT 21,13;c
p;" ";AT 16,14;lp;" ";AT 15,23;f
;" ";AT 21,23;ti;" ";AT 10,2;va;
" ";AT 14,2;ra;" ": GO TO 3020
```


Programas

```

3090 PAPER 5: PRINT AT 21,3;" ":
PAPER 2: PRINT AT 21,9;" "
3100 GO SUB 3200
3110 GO SUB 3390
3120 GO SUB 4150
3130 GO SUB 3790
3140 GO SUB 3770
3150 GO SUB 3910
3160 GO SUB 3710
3170 GO SUB 3380
3175 LET ti=(PEEK 23672+256*PEEK
23673)/50
3180 GO SUB 4010
3190 GO TO 3100
3200 LET a$=INKEY$
3210 IF a$="6" THEN GO TO 3560
3220 IF (a$="7") AND (a<>0) THEN
GO TO 3590
3230 IF a$="8" THEN GO TO 3630
3240 IF a$="5" THEN GO TO 3650
3250 IF a$="k" THEN GO TO 3670
3260 IF a$="j" THEN GO TO 3690
3270 IF a$="t" THEN GO TO 3540
3280 PAPER 5: INK 0: IF a$="1" T
HEN LET m1=0: PRINT AT 12,23;"1
"
3290 IF a$="2" THEN LET m2=0: P
RINT AT 12,25;"2"
3300 IF a$="3" THEN LET m3=0: P
RINT AT 12,27;"3"
3310 IF a$="4" THEN LET m4=0: P
RINT AT 12,29;"4"
3320 IF (a$="q") AND (f>0) THEN
LET m1=1: INK 7: PAPER 2: PRINT
AT 12,23;"1"
3330 IF (a$="w") AND (f>0) THEN
LET m2=1: INK 7: PAPER 2: PRINT
AT 12,25;"2"
3340 IF (a$="e") AND (f>0) THEN
LET m3=1: INK 7: PAPER 2: PRINT
AT 12,27;"3"
3350 IF (a$="r") AND (f>0) THEN
LET m4=1: INK 7: PAPER 2: PRINT
AT 12,29;"4"
3360 IF a$=" " THEN LET v1=0
3370 RETURN
3380 INK 7: PAPER 2: PRINT AT 5,
2;v;" ";AT 5,23;a;" ";AT 21,13;c
p;" ";AT 16,14;lp;" ";AT 15,23;f
;" ";AT 21,23;ti;" ";AT 10,2;va;
" ";AT 14,2;ra;" ": RETURN
3390 LET va=ba+INT (RND*5)
3400 LET mt=m1+m2+m3+m4: LET f=f

```

```

-mt*3: IF f<0 THEN LET f=0
3410 IF f=0 THEN LET mt=0: INK
0: PAPER 5: PRINT AT 12,23;"1";A
T 12,25;"2";AT 12,27;"3";AT 12,2
9;"4": INK 7: PAPER 2
3420 LET ra=INT (v/5*va)
3430 IF ra<0 THEN LET ra=-ra: L
ET ra=ra+300
3440 LET vm=300*mt: GO TO 3880
3450 LET vn=30*mt
3460 IF er=1 THEN GO TO 3490
3470 IF v<=vn THEN LET v=vn: LE
T v1=0
3480 IF (v>=vm-50) AND (pa=0) TH
EN LET v=v-INT (v1/2): LET v1=0
3490 LET a=a+m: LET v=v+v1
3500 IF a<=0 THEN LET a=0
3510 IF a>=12000 THEN LET a=120
00
3520 IF lp<=0 THEN GO TO 4180
3530 RETURN
3540 IF t=1 THEN LET t=0: PAPER
5: PRINT AT 9,25;" ": PAPER 2:
PRINT AT 9,30;" ": RETURN
3550 LET t=1: PAPER 2: PRINT AT
9,25;" ": PAPER 5: PRINT AT 9,30
;" ": RETURN
3560 LET h=h-1: IF h=-3 THEN LE
T h=-2
3570 LET m=-5*h: IF h=-2 THEN L
ET m=m+40
3580 RETURN
3590 IF a=0 THEN LET m=0: LET h
=0: RETURN
3600 LET h=h+1: IF h=3 THEN LET
h=2
3610 LET m=-5*h: IF h=2 THEN LE
T m=m-40
3620 RETURN
3630 LET i=i+1: LET u=1: IF i=4
THEN LET i=3
3640 RETURN
3650 LET i=i-1: LET u=-1: IF i=0
THEN LET i=1
3660 RETURN
3670 LET v=v+9: LET v1=9: LET pa
=0
3680 RETURN
3690 LET v=v-9: LET v1=-9: LET p
a=1
3700 RETURN
3710 IF cp<=0 THEN GO TO 3750
3720 IF mt=0 THEN RETURN

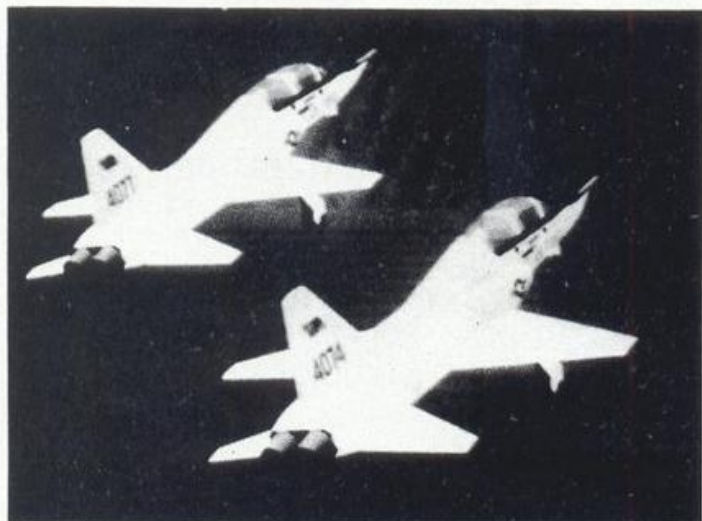
```



```

3730 LET cp=cp-INT (v/3)
3740 RETURN
3750 IF oo=1 THEN GO TO 3755
3752 LET cp=0: INK 7: PAPER 2: P
RINT AT 21,14;"0": LET oo=1
3755 IF mt<>0 THEN LET lp=lp-IN
T (v/3)
3760 RETURN
3770 INK 0: PAPER 0: PRINT AT 8+
yu,13;z$
3771 PRINT AT 10+yu,13;x$
3772 PRINT AT 9+yu,13;c$
3773 PRINT AT 7+yu,13;v$
3774 PRINT AT 6+yu,13;b$
3775 INK 5: PAPER 0: PRINT AT 8+
h,13;w$: LET yu=h

```



```

3776 PRINT AT 8+yu,13;z$
3777 PRINT AT 10+yu,13;x$
3778 PRINT AT 9+yu,13;c$
3779 PRINT AT 7+yu,13;v$
3780 PRINT AT 6+yu,13;b$
3785 RETURN
3790 IF i=1 THEN GO TO 3820
3800 IF i=2 THEN GO TO 3840
3810 IF i=3 THEN GO TO 3860
3820 LET w$=" I ": LET f$=" E
": LET g$=" E ": LET h$=" \
E ": LET i$=" E"
3830 LET z$=w$: LET x$=f$: LET c
$=g$: LET v$=h$: LET b$=i$: RETU
RN
3840 LET w$="GGGGG": LET h$=" :
": LET g$=" ": LET f$="

```

```

": LET i$=" "
3850 LET z$=w$: LET x$=f$: LET c
$=g$: LET v$=h$: LET b$=i$: RETU
RN
3860 LET w$=" H ": LET f$="
E": LET g$=" E ": LET h$=" E
/ ": LET i$=" D "
3870 LET z$=w$: LET x$=f$: LET c
$=g$: LET v$=h$: LET b$=i$: RETU
RN
3880 IF er=1 THEN GO TO 3450
3890 IF v>=vm THEN LET v=vm: LE
T vl=0
3900 GO TO 3450
3910 IF a=0 THEN RETURN
3920 IF m1+m2>m3+m4 THEN LET im
=1: GO TO 3950
3930 IF m1+m2<m3+m4 THEN LET im
=-1: GO TO 3950
3940 RETURN
3950 LET we=1+INT (RND*10)
3960 IF we<10 THEN RETURN
3970 LET i=i+im
3980 IF i=4 THEN LET i=3
3990 IF i=0 THEN LET i=1
4000 RETURN
4010 IF a=0 THEN GO TO 4070
4020 LET gj=0
4030 IF (a<=30) AND (a>0) AND (i
<>2) THEN GO TO 4190
4040 IF (a=10) OR (a=5) THEN LE
T eu=eu+1: RETURN
4050 IF (a>=15) AND (a<=30) THEN
LET ev=ev+1: RETURN
4060 LET eu=1: LET ev=1: RETURN
4070 IF gj=1 THEN GO TO 4200
4080 IF ((cp=0) AND (t=1) AND (v
>300) AND (v<551)) AND ((eu>29)
AND (ev>34)) OR (eu>69) THEN LE
T gj=1: LET vr=lp-v*2000/300: LE
T m=0: LET h=0: GO SUB 4350: RET
URN
4090 PRINT AT 5,23;"0 "
4100 IF mt=0 THEN LET k$="
LOS MOTORES PARADOS": LET j$="
CAYO EN PICADO POR TENER": GO T
O 4250
4110 IF cp>0 THEN LET k$=" ": L
ET j$=" NO HA LLEGADO A LA PIS
TA": GO TO 4250
4120 IF t=0 THEN LET k$="
DE ATERRIZAJE ": LET j$="
NO HA SACADO EL TREN ": GO TO 42

```


Programas

```

50
4130 IF v<400 THEN LET j$="
      SE HA ESTRELLADO," : LET k$="
      ATERRIZO MUY DESPACIO": GO TO
4250
4140 LET j$="          SE HA ESTRELL
ADO," : LET k$="          ATERRIZO BRUS
CAMENTE": GO TO 4250
4150 IF (mt=0) AND (a>0) THEN L
ET er=1: LET v=v+90: LET h=2: LE
T m=-200: GO SUB 4320: RETURN
4160 IF er=1 THEN LET er=0: LET
v1=0
4170 RETURN
4180 PAPER 2: INK 7: PRINT AT 16
,14;"O " : LET k$=" " : LET j$="
      SE LE ACABO LA PISTA": GO TO
4250
4190 LET j$=" SE ESTRELLLO POR IR
      INCLINADO": LET k$="          A
      BAJA ALTURA": GO TO 4250
4200 IF v=0 THEN GO TO 4220
4210 RETURN

```

```

4220 IF 1p>vr THEN GO TO 4240
4230 LET k$=" " : LET j$="ENHORAB
UENA , LO HA CONSEGUIDO": GO TO
4250
4240 LET k$="          EN LA PIST
A": LET j$=" HA FRENADO DEMASIAD
O DEPRISA"
4250 PAPER 6: INK 0: PRINT AT 0,
1;j$;AT 1,1;k$
4260 IF cf=1 THEN GO TO 4290
4270 FOR w=1 TO 400: NEXT w
4280 LET cf=1: LET k$="
      (S/N)          " : LET j$=
"Desea intentar otro aterrizaje"
: GO TO 4250
4290 LET a$=INKEY$: IF a$="s" TH
EN GO TO 2000
4300 IF a$="n" THEN GO TO 4340
4310 GO TO 4290
4330 RETURN
4340 STOP
4350 BEEP .25,-10: BEEP .25,-10
4360 RETURN

```

SUSCRIBASE A

Todospectrum

(12 NUMEROS)

TARIFA DE PRECIOS DE SUSCRIPCION

	CORREO ORDINARIO		CORREO CERTIFICADO		CORREO AEREO		CORREO AEREO-CERTIF.	
	PTAS.	\$	PTAS.	\$	PTAS.	\$	PTAS.	\$
ESPAÑA	3.000	21	3.273	23	3.055	22	3.333	24
EUROPA, MARRUECOS, TUNEZ, TURQUIA, ARGELIA Y CHIPRE	3.456	25	4.272	31	3.600	26	4.418	31
COSTA RICA, CUBA, CHILE, PARAGUAY Y REP. DOMINICANA	3.396	24	4.212	30	4.164	30	4.980	36
GIBRALTAR Y PORTUGAL	3.264	23	4.080	29	3.149	22	3.965	28
FILIPINAS	3.264	23	3.540	25	3.775	27	4.050	29
RESTO DEL MUNDO	3.456	25	4.272	31	4.224	30	5.040	36

CUPON DE PEDIDO

Recorte y envíe este cupón a: **Todospectrum** EDISA. Lopez de Hoyos, 141 - 28002 - MADRID

El importe lo abonaré: POR CHEQUE ☐ CONTRA REEMBOLSO ☐

CON TARJETA DE CREDITO ☐ American Express ☐ Visa ☐ Interbank ☐

Número de mi Tarjeta: Fecha de caducidad:

NOMBRE

DIRECCION

CIUDAD D.P. PROVINCIA



Romanos

En este mes son multitud a quienes "no les salen las cuentas". Para eso el Spectrum no tiene ninguna solución, pero con ayuda de estos programas, podrá realizar sus cuentas en números romanos. Escoja: ¿Pascal o Forth?

El listado 1 ha sido realizado en Pascal, gracias al compilador HYPs.

Se declara un tipo T que será

una matriz de 16 elementos donde se guardan los caracteres que se van convirtiendo. Las variables del programa son NUM (el número que se va a transformar), I (un contador para movernos por la matriz), y ROMAND que será el tipo T previamente declarado.

En las líneas 40 a 70 se inicializa a blancos la matriz donde vamos a guardar los caracteres romanos. El

programa principal (restantes líneas) el algoritmo consiste en ir comparando el número con diversas cantidades, primero con 1.000, 900, 500, etc. hasta 1. Cuando el número es mayor que el que se compara se le asigna a la matriz el caracter romano que corresponde y se le resta el número con el que se ha comparado.

Hay dos clases de bucles: a) cuando se puedan poner 3 caracteres iguales tales como 3M, 3C, 3X o 3I, se utiliza el bucle WHILE (líneas 115-140). b) Si sólo se puede poner un caracter, caso de la D, L o V, se utiliza un bucle IF ya que sólo se pasa una vez por él (líneas 180-195). En los casos tales como 900 o 400 en los que no se pueden escribir 4 caracteres iguales, habrá que asignar a la matriz 2 caracteres en el mismo bucle, y avanzar el contador el doble (líneas 145-175).

El listado 2 tiene análogos resultados, con la diferencia de que está realizado en Forth.

Programa en Pascal: José Ramón

Programa en Forth: Alfonso Martín

PROGRAMA 1

```

3018 10 PROGRAM ROMANOS;
3018 15 TYPE
3018 20 T=ARRAY [1..16] OF CH
AR;
3018 25 VAR
3021 30 NUM, I: INTEGER;
3021 35 ROMANO: T;
3021 40 PROCEDURE INICIALIZAR (
VAR ROMANO: T);
3024 45 VAR
3024 50 J: INTEGER;
3024 55 BEGIN
303C 60 FOR J:=1 TO 16 DO
305F 65 ROMANO[J]:= ' ';
3084 70 END;
3090 75 BEGIN (*PP*)
3099 80 READ (NUM);
309F 85 I:=1;
30B8 87 WRITE (CHR(16));
30A5 90 WRITELN;
30A8 95 WRITELN ('EL NUMERO P
EDIDO ES, ', NUM); WRITELN;
30D6 100 IF NUM>4000 THEN WRI
TELN ('EL NUMERO ES DEMASIADO GRA
NDE')
3111 105 ELSE BEGIN
3117 110 INICIALIZAR (ROMANO);
3120 115 WHILE NUM>1000 DO
3139 120 BEGIN
3139 125 ROMANO[I]:='M';
3156 130 NUM:=NUM-1000;
3166 135 I:=I+1;
316C 140 END;
316F 145 IF NUM>900 THEN
3185 150 BEGIN

```

```

3185 155
31A2 160
31C0 165
31C8 170
31D0 175
31D7 180
GIN
31ED 185
320A 190
3221 195
3220 200
3236 205
3236 210
3253 215
3271 220
3280 225
3288 230
3298 235
32A1 240
32A1 245
32BE 250
32CD 255
32D5 260
32D7 265
32ED 270
32ED 275
330A 280
3328 285
3330 290
3338 295
333F 300
3355 305
3355 310
3372 315
3381 320

```

```

ROMANO[I]:='C';
ROMANO[I+1]:='M';
I:=I+2;
NUM:=NUM-900
END;
IF NUM>500 THEN BE
ROMANO[I]:='D';
NUM:=NUM-500; I:=I+1
END;
IF NUM>400 THEN
BEGIN
ROMANO[I]:='C';
ROMANO[I+1]:='D';
NUM:=NUM-400;
I:=I+2
END;
WHILE NUM>100 DO
BEGIN
ROMANO[I]:='C';
NUM:=NUM-100;
I:=I+1
END;
IF NUM>90 THEN
BEGIN
ROMANO[I]:='X';
ROMANO[I+1]:='C';
I:=I+2;
NUM:=NUM-90
END;
IF NUM>50 THEN
BEGIN
ROMANO[I]:='L';
NUM:=NUM-50;
I:=I+1

```


Programas

28

```

8389 325 END;
8388 330 IF NUM>=40 THEN
839E 335 BEGIN
839E 340 ROMANO[I] := 'X';
838B 345 ROMANO[I+1] := 'L';
83D9 350 NUM := NUM-40;
83E8 355 I := I+2
83F0 360 END;
83F0 365 WHILE NUM>=10 DO
8409 370 BEGIN
8409 375 ROMANO[I] := 'X';
8425 380 NUM := NUM-10;
8435 385 I := I+1
843D 390 END;
843F 395 IF NUM=9 THEN
8451 400 BEGIN
8451 405 ROMANO[I] := 'I';
846E 410 ROMANO[I+1] := 'X';
848C 415 NUM := NUM-9;
849B 420 I := I+2
84A3 425 END;
84A3 430 IF NUM>=5 THEN
84B9 435 BEGIN
84B9 440 ROMANO[I] := 'V';
84D5 445 NUM := NUM-5;
84E5 450 I := I+1
84ED 455 END;
84EC 460 IF NUM=4 THEN
84FE 465 BEGIN
84FE 470 ROMANO[I] := 'I';
851B 475 ROMANO[I+1] := 'V';
8539 480 NUM := NUM-4;
8543 485 I := I+2
854B 490 END;
854B 495 WHILE NUM>=1 DO
8564 500 BEGIN
8564 505 ROMANO[I] := 'I';

```

```

8581 510 NUM := NUM-1;
8588 515 I := I+1
8590 520 END;
8592 525 WRITELN('LA CONVERS
ION A NUMEROS ROMANOS ES:'); WRIT
ELN;
85C6 530 FOR I:=1 TO 16 DO
85E0 535 WRITE(ROMANO[I])
85FF 540 END
8602 545 END.
End Address: 8604

```

Run?

```

EL NUMERO PEDIDO ES,1234
LA CONVERSION A NUMEROS ROMANOS
ES:
MCCXXXIV
EL NUMERO PEDIDO ES,3652
LA CONVERSION A NUMEROS ROMANOS
ES:
MMMDCCLII
EL NUMERO PEDIDO ES,7654
EL NUMERO ES DEMASIADO GRANDE

```

Salida por pantalla del programa en Pascal.

PROGRAMA 2

```

: V @ VARIABLE ;
: U A V R V C U D U U U M U P U
NO
: INPUT PAD 11 11 EXPECT .0 PAD
(NUMBER) DROP DROP ;
: DEC 0 @ 10 /MOD SWAP A 10 /M
00 SWAP 0 10 ;
: ERR CR 2 INK . 11 EL NO CS MA
YOR DE 1000, NO VALE 11 0 INK ;
: NOPU 73 U 1 86 M 1 88 P 1 0 @
NO ;
: NORD 58 U 1 74 M 1 67 P 1 0 @
NO ;
: NORD 67 U 1 68 M 1 77 P 1 0 @
NO ;
: NORD 14 U @ EMIT M @ EMIT ;
: NORD NO @ 4 = IF NORD1 ELSE NO
@ 00 U @ EMIT LOOP THEN ;
: NORD NO @ 0 = IF ELSE NORD1 THE
N ;
: NORD M @ EMIT NO @ 5 = 0 1 C @

```

```

0 DO U @ EMIT LOOP ;
: NORD U @ EMIT P @ EMIT ;
: NORD NO @ 0 = IF NORD1 ELSE NORD
1 THEN ;
: NORD NO @ 5 > IF NORD1 ELSE M @
EMIT THEN ;
: NORD NO @ 1 L IF NORD1 ELSE NORD0
THEN ;
: NORDM DEC NORD NORD NORD NORD NO
PU NORD ;
: INC 0 @ 1000 L IF ERR ELSE 0 5
PAPER NORDM THEN OF
: INK 1 PAPER 2 BORRER 11 INTRO
DUCE EL NO 11 CR INPUT DUP 0 1
INC ;
: LISTNUM CR 7 PAPER 000 0 00 1
ABRAPER 1 0 1 NORDM CR 100 1
: SEE KEY 74 - 15 LISTNUM ELSE 1
M1 THEN CR ;
: NP 1 INK 11 QUIERES INTRODUCIR
P NO (N) O VER LISTA(L) 11 CR OF
E 0 INK ;

```

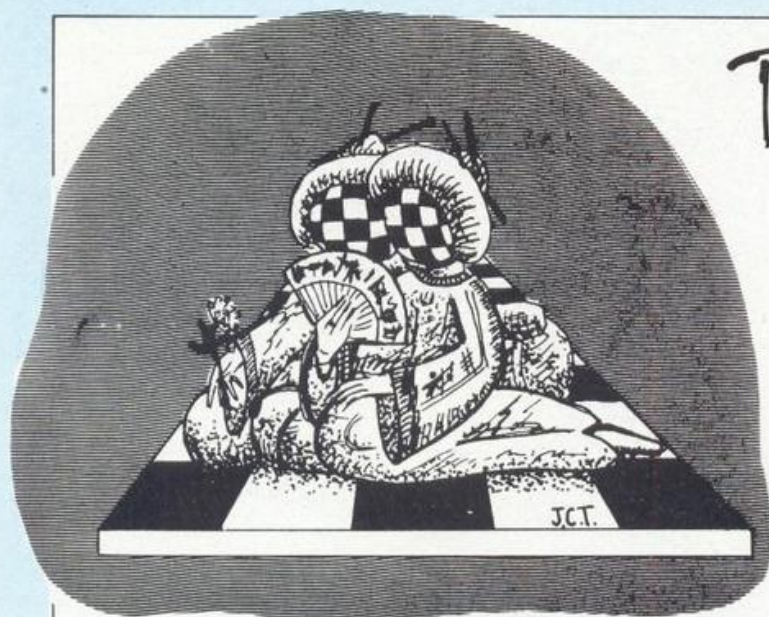
```

OK
NR QUIERES INTRODUCIR NO.(N) O
VER LISTA(L)
INTRODUCE EL NO.
12 XII
OK
NR QUIERES INTRODUCIR NO.(N) O
VER LISTA(L)
INTRODUCE EL NO.
400 CD
OK
NR QUIERES INTRODUCIR NO.(N) O
VER LISTA(L)
INTRODUCE EL NO.
777 DCCLXXVII
OK
NR QUIERES INTRODUCIR NO.(N) O
VER LISTA(L)
INTRODUCE EL NO.
888 DCCCLXXXVIII
OK
21 EMIT

```

Salida por pantalla del programa en Forth.





Damas chinas

Si no teme ser derrotado por una máquina, le proponemos una lucha "inteligente" contra su Spectrum. Para ganar a su oponente, tendrá que colocar sus fichas en el extremo del tablero en forma de triángulo, antes de que él haga lo propio en el otro extremo. Se avanza en diagonal, siempre hacia adelante, y una sola casilla, a excepción de que delante exista una ficha y que la posición siguiente esté libre. Si alguno de los dos jugadores no puede mover, se producen tablas. (La indicación de tablas sólo se da cuando el ordenador queda bloqueado). Las instrucciones de manejo se dictan claramente en pantalla.

Un consejo final: no se desespere si le resulta difícil ganar y fíjese cómo actúa su oponente.

Autor: Gregorio Ferrero

16 K

VARIABLES UTILIZADAS

x(6), y(6): Posiciones de las fichas del Spectrum.

a(6), b(6): Posiciones de las fichas del jugador.

p(6,2): Almacena la prioridad de movimiento de las fichas del Spectrum en las dos direcciones.

pu: Puntos jugador.

pus: Puntos Spectrum.

ink: Color fichas jugador.

inks: Color fichas Spectrum.

Tur: Indica quién empieza.

a\$: Nombre jugador. En la línea 1300, a\$ = a "s" o "n".

fi: Fila de la ficha del jugador a mover.

co: Columna de la ficha del jugador a mover.

mov: Indica la dirección del avance y si hay salto.

sal: Indica la dirección del salto.

ba, bb: Posición a la que puede saltar el jugador, y que el Spectrum debe bloquear.

ZX SPECTRUM



Fila de ficha a mover L

```
10 CLS : REM DAMAS CHINAS
  Gregorio Ferrero 1984
20 GO TO 9000
50 INPUT "Dime tu nombre":a$
60 DIM x(6): DIM y(6): DIM a(6)
  ): DIM b(6): DIM p(6,2): LET pu=
  0: LET pus=0
70 RESTORE 90: FOR f=1 TO 6
80 READ x,y,a,b: LET x(f)=x: L
  ET y(f)=y: LET a(f)=a: LET b(f)=
  b: NEXT f
90 DATA 0,16,20,16,1,15,19,15,
  1,17,19,17,2,14,18,14,2,16,18,16
```

```
,2,18,18,18
100 INPUT "Que color eliges? (1
  -6)":ink
110 IF ink<1 OR ink>6 THEN GO
  TO 100
120 INPUT "Que color para mi? (
  1-6)":inks
130 IF inks<1 OR inks>6 OR inks
  =ink THEN GO TO 120
140 INPUT "Quien empieza? (Yo=1
  :Tu=2)":tur
150 IF tur<1 OR tur>2 THEN GO
  TO 140
```



```

160 LET tur=INT (tur)*1000
170 REM Dibuja tablero
180 INK 0: PAPER 0: BORDER 0: C
LS
190 PRINT PAPER 7;AT 0,16;" ";
AT 20,16;" "
200 FOR n=1 TO 19 STEP 2
210 PRINT PAPER 7;AT n,15;" ";
AT n,17;" "
220 IF n=19 THEN GO TO 240
230 PRINT PAPER 7;AT n+1,14;"
";AT n+1,16;" ";AT n+1,18;" "
240 NEXT n
245 FOR n=0 TO 20: PRINT AT n,1
2; PAPER 0; INK 7;n: NEXT n
248 FOR a=0 TO 4: PRINT AT 21,a
+14; PAPER 0; INK 7;a: NEXT a
250 PRINT AT 0,0; INK inks;"ZX
SPECTRUM"
260 PRINT AT 0,32-LEN a$; INK i
nk;a$
270 FOR n=1 TO 6: PRINT AT x(n)
,y(n); PAPER 7; INK inks;"●";AT
a(n),b(n); INK ink;"●": NEXT n
300 GO TO tur
1000 REM Juega Spectrum
1010 FOR n=1 TO 6
1020 IF ATTR (x(n)+1,y(n)-1)=0 T
HEN LET p(n,1)=0
1025 IF ATTR (x(n)+1,y(n)-1)=56
THEN LET p(n,1)=1
1030 IF ATTR (x(n)+1,y(n)-1)=56+
inks OR ATTR (x(n)+1,y(n)-1)=56+
ink THEN LET p(n,1)=2
1040 IF ATTR (x(n)+1,y(n)+1)=0 T
HEN LET p(n,2)=0
1050 IF ATTR (x(n)+1,y(n)+1)=56
THEN LET p(n,2)=1
1060 IF ATTR (x(n)+1,y(n)+1)=56+
inks OR ATTR (x(n)+1,y(n)+1)=56+
ink THEN LET p(n,2)=2
1065 IF p(n,1)=2 AND ATTR (x(n)+
2,y(n)-2)=56 AND p(n,2)=2 AND AT
TR (x(n)+2,y(n)+2)=56 THEN LET
i=INT (RND*2)-2: IF i=-1 THEN L
ET i=2: GO TO 3000
1070 IF p(n,1)=2 AND ATTR (x(n)+
2,y(n)-2)=56 THEN LET i=-2: GO
TO 3000
1080 IF p(n,2)=2 AND ATTR (x(n)+
2,y(n)+2)=56 THEN LET i=2: GO T
O 3000
1090 NEXT n

```

```

1100 FOR n=1 TO 6
1110 IF (ATTR (a(n)-1,b(n)-1)=56
+inks OR ATTR (a(n)-1,b(n)-1)=56
+ink) AND ATTR (a(n)-2,b(n)-2)=5
6 THEN LET i=-2: GO TO 4000
1120 IF (ATTR (a(n)-1,b(n)+1)=56
+inks OR ATTR (a(n)-1,b(n)+1)=56
+ink) AND ATTR (a(n)-2,b(n)+2)=5
6 THEN LET i=2: GO TO 4000
1130 NEXT n
1140 FOR n=1 TO 6
1145 IF ATTR (x(n)+1,y(n)-1)=56
AND ATTR (x(n)+1,y(n)+1)=56 THEN
LET i=INT (RND*2)-1: IF i=0 TH
EN LET i=1: GO TO 5000
1150 IF ATTR (x(n)+1,y(n)-1)=56
THEN LET i=-1: GO TO 5000
1160 IF ATTR (x(n)+1,y(n)+1)=56
THEN LET i=1: GO TO 5000
1170 NEXT n
1180 PRINT AT 10,13; INK 7;"TABL
AS"
1300 INPUT "Otra partidita? (s/n
) ";a$
1310 IF a$(1)="s" OR a$(1)="S" T
HEN CLS : GO TO 50
1320 STOP
2000 REM Juega Humano
2010 INPUT "Fila de ficha a move
r ";fi: INPUT "Columna de ficha
a mover ";co: LET co=co+14
2020 FOR n=1 TO 6
2030 IF a(n)=fi AND b(n)=co THEN
GO TO 2050
2040 NEXT n
2045 PRINT AT 10,0; INK 7;"NO HA
Y FICHA": PAUSE 50: PRINT AT 10,
0;" ": GO TO 2010
2050 INPUT "Movimiento?0-IZQ;1-S
ALTO;2-DER";mov
2055 IF mov=1 THEN GO TO 2200
2060 IF mov<>0 AND mov<>2 THEN
GO TO 2050
2070 LET mov=mov-1
2080 IF ATTR (a(n)-1,b(n)+mov)=5
6 THEN GO TO 2090
2085 PRINT AT 13,0; INK 7;"MOVIM
IENTO";AT 15,0; INK 7;"NO VALIDO
": PAUSE 50: PRINT AT 13,0;"
";AT 15,0;" ": GO
TO 2050
2090 LET a(n)=a(n)-1: LET b(n)=b
(n)+mov

```



```

2100 REM Impresion
2105 PRINT PAPER 7; FLASH 1; IN
K ink; AT a(n)+1, b(n)-mov; "●": BE
EP .2, 10; BEEP .2, 15; FLASH 0
2110 PRINT PAPER 7; AT a(n)+1, b(
n)-mov; " "; AT a(n), b(n); INK ink
; "●": BEEP .2, 10; BEEP .2, 15
2120 IF a(n)<3 THEN LET pu=pu+1
2130 IF pu=10 THEN GO TO 2500
2140 GO TO 1000
2200 REM Salto
2210 INPUT "Salto? (0-IZQ; 4-DER)
"; sal
2215 IF sal<>0 AND sal<>4 THEN
GO TO 2210
2220 LET sal=sal-2
2230 IF ATTR (a(n)-1, b(n)+SGN sa
l)<>56 AND ATTR (a(n)-2, b(n)+sal
)=56 THEN GO TO 2300
2240 PRINT AT 13, 0; "MOVIMIENTO";
AT 15, 0; "NO VALIDO"
2250 PAUSE 50; PRINT AT 13, 0; "
"; AT 15, 0; " "
2260 GO TO 2050
2300 LET a(n)=a(n)-2; LET b(n)=b
(n)+sal
2305 PRINT PAPER 7; FLASH 1; AT
a(n)+2, b(n)-sal; INK ink; "●": BE
EP .2, 10; BEEP .2, 15; FLASH 0
2310 PRINT PAPER 7; AT a(n)+2, b(
n)-sal; " "; AT a(n), b(n); INK ink

```

```

; "●": BEEP .2, 10; BEEP .2, 15
2320 IF a(n)<2 THEN LET pu=pu+2
2325 IF a(n)=2 THEN LET pu=pu+1
2330 IF pu=10 THEN GO TO 2500
2340 GO TO 1000
2500 REM Victoria jugador
2510 FLASH 1; INK 7; PRINT AT 10
, 20; INVERSE 1; "HAS GANADO"
2530 FOR b=0 TO 255
2540 OUT 254, b: OUT 254, 7
2550 NEXT b
2560 FLASH 0
2570 GO TO 1300
3000 REM Salto
3010 LET x(n)=x(n)+2; LET y(n)=y
(n)+i
3015 PRINT PAPER 7; FLASH 1; IN
K inks; AT x(n)-2, y(n)-i; "●": BEE
P .2, 20; BEEP .2, 25; FLASH 0
3020 PRINT PAPER 7; AT x(n)-2, y(
n)-i; " "; AT x(n), y(n); INK inks;
"●": BEEP .2, 20; BEEP .2, 25
3030 IF x(n)>18 THEN LET pus=pu
s+2
3035 IF x(n)=18 THEN LET pus=pu
s+1
3040 IF pus=10 THEN GO TO 3500
3050 GO TO 2000
3500 REM Victoria Ordenador
3510 FLASH 1; INK 7; PRINT AT 10
, 2; INVERSE 1; "HE GANADO"

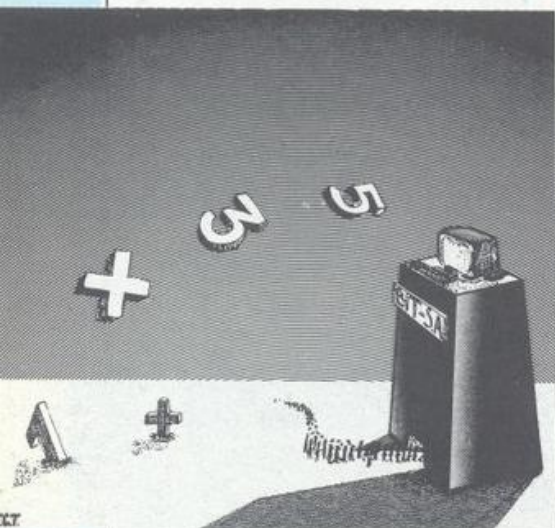
```

A*azmorø&?

¿Por qué correr el riesgo de que los curiosos lean sus cartas? Antes de que deje volar su imaginación a la películas de espionaje, veamos qué puede hacer este programa y decida después su utilización.

Este programa le permitirá codificar un mensaje basándose en una clave determinada. El programa se basa simplemente en una tabla compuesta por caracteres (en principio se han incluido desde el CHR\$ 32 hasta el CHR\$ 127, inclusive). Gracias a esta tabla y ba-

sándonos en la clave podrá codificar un mensaje, y también posteriormente partiendo del código creado, reconstruirlo. La tabla inicial se crea colocando la lista de caracteres de la variable 1\$ como referencia de fila y columna, la primera línea es la propia lista de caracteres en el orden en que está la variable 1\$. La siguiente línea se obtiene cambiando de lugar los caracteres, el último carácter de la línea anterior pasa a ser el primero de esta línea y así sucesivamente hasta completar la tabla.



Programas

```

3520 FOR b=255 TO 0 STEP -1
3530 OUT 254,b: OUT 254,7
3540 NEXT b
3550 FLASH 0: GO TO 1300
4000 REM Bloqueo
4010 LET ba=a(n)-2: LET bb=b(n)+
i
4020 FOR f=1 TO 6
4030 IF x(f)=ba-1 AND y(f)=bb-1
THEN GO TO 4060
4035 IF x(f)=ba-1 AND y(f)=bb+1
THEN GO TO 4080
4040 NEXT f
4050 GO TO 1130
4060 LET x(f)=x(f)+1: LET y(f)=y
(f)+1
4065 PRINT PAPER 7: FLASH 1: IN
K inks;AT x(f)-1,y(f)-1;"●": BEE
P .2,20: BEEP .2,25: FLASH 0
4070 PRINT PAPER 7:AT x(f)-1,y(
f)-1;" ": GO TO 4100
4080 LET x(f)=x(f)+1: LET y(f)=y
(f)-1
4085 PRINT PAPER 7: FLASH 1: IN
K inks;AT x(f)-1,y(f)+1;"●": BEE
P .2,20: BEEP .2,25: FLASH 0
4090 PRINT PAPER 7:AT x(f)-1,y(
f)+1;" "
4100 PRINT INK inks;AT x(f),y(f
);"●": BEEP .2,20: BEEP .2,25
4105 IF x(f)>17 THEN LET pus=pu

```

```

s+1
4108 IF pus=10 THEN GO TO 3500
5000 REM Movimiento
5005 PRINT AT x(n),y(n); FLASH 1
; INK inks;"●": BEEP .2,20: BEEP
.2,25: FLASH 0
5010 PRINT AT x(n),y(n); PAPER 7
;" "
5020 LET x(n)=x(n)+1: LET y(n)=y
(n)+1
5030 PRINT AT x(n),y(n); PAPER 7
; INK inks;"●": BEEP .2,20: BEEP
.2,25
5035 IF x(n)>17 THEN LET pus=pu
s+1
5040 IF pus=10 THEN GO TO 3500
5050 GO TO 2000
9000 RESTORE 9030: FOR b=0 TO 7
9010 READ c: POKE USR CHR$ 144+b
,c
9020 NEXT b
9030 DATA 0,60,126,126,126,126,6
0,0
9040 GO TO 50

```

NOTAS GRAFICAS

A B C ...

● B C ...

Los menús que incluye el programa permitirán: grabar el código o el mensaje en una cinta, cargar uno previamente grabado, sacarlo por impresora, operar con un texto cargado en la memoria o introducirlo por el teclado. También intercambiar el texto residente en memoria, por su correspondiente código o su mensaje.

Una vez creada la tabla, basándose en los caracteres de la clave podrá crear el código. Así, si la clave es "PAN" y quiere codificar la palabra "COSA" (el resultado debería ser "s/q") el proceso es el siguiente: Buscar en la tabla la línea correspondiente a la "P" (primer carácter de la clave), y la columna correspondiente al "C"

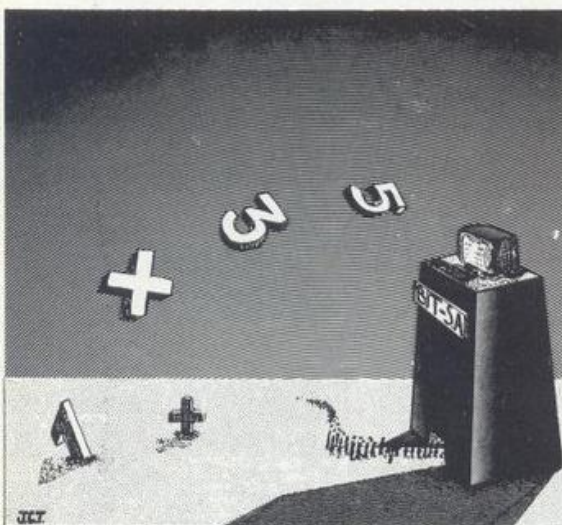
(primer carácter del mensaje); la intersección en la tabla en este caso de la "s" será el primer carácter de código. Y así el proceso se repite hasta el final del texto; si como es normal los caracteres de la clave se agotan antes que los del texto, se comienza otra vez por el primero de la clave. Así, para poder codificar la "A" del mensaje comenzamos nuevamente la clave, siendo "P" el carácter, y así todas las veces necesarias.

El método para traducir es el inverso; basándonos también en la clave primitiva (con la que se codificó el mensaje) y siguiendo con nuestro ejemplo, para traducir la "s" buscaríamos la línea de "P" (primer carácter de la clave), y

ahora buscaríamos dentro de esa línea el carácter, y traducir "s" (si la tabla es correcta solo habrá una "s"); una vez encontrado vemos en qué columna está y la letra que marca esa columna sería la traducción de "s", en este caso la "C". Como antes, si se termina la clave antes que el código volveremos al primer carácter de la clave.

Un mismo mensaje se codificará distinto según la clave. También podemos someter a un mensaje a un cierto número de codificaciones repetidas. Para recuperar el texto primero, debemos someter al último código al mismo número de traducciones también continuadas, esto se consigue gracias a la utilización adecuada de la opción

Programas



"S" del primer menú (intercambio de entrada por salida).

Para mayores complicaciones en el código, se puede cambiar el orden de los caracteres en la tabla (con lo que se crea una tabla distinta); línea 180, variable I\$. Siempre que no haya ninguno repetido, ni falte ninguno de los que vamos a utilizar. Aunque es aconsejable que sean correlativos (sólo letras mayúsculas, sólo números...) ya que, si no, se complica mucho el resto del programa.

Si hace esto deberá retocar las líneas 110, 210, 220, 490, 500, 740, 750, 760; adecuando los valores de

la longitud de I\$, y del primer carácter de la lista para tomarlo como referencia (inicialmente la longitud es de 96 caracteres, y el primer carácter es el CHR\$ 32, que corresponde al espacio; por ésto se suma 31 al valor de la posición en I\$ para hallar el código absoluto correspondiente en el manual de Sinclair, apéndice A).

Por último, si reduce el número de caracteres ahorrara memoria, siendo incluso posible correr en la versión de 16K, y también se reduciría el tiempo de ejecución.

Autor: José Félix Alvarez 16K.

```

10 REM
20 REM *****
30 REM JOSE FELIX ALVAREZ
40 REM CODIGOS 14/10/1984
50 REM *****
60 REM
70 REM *****
80 REM Dimensiona tabla de ca
   racteres. Inicializa,
   texto en memoria (texme
   mo).
90 REM *****
100 REM
110 DIM t$(96,96)
120 LET texmemo=0
130 REM
140 REM *****
150 REM Define caracteres;l$.
   Y crea tabla;t$.
160 REM *****
170 REM
180 LET l$=" !"+CHR$ 34+" $%&?' (
) *+, -./0123456789:;<=>?@ABCDEFGH
IJKLMNOPQRSTUVWXYZ[\]^_#abcdefgh
ijklmnopqrstuvwxyz{ }"
190 LET t$(1)=l$
200 PRINT AT 12,4; INVERSE 1; F
LASH 1;"CREANDO TABLA DE CARACTE
RES"
210 FOR N=2 TO 96
220 LET t$(N)=t$(N-1,96)+t$(N-1
)
230 NEXT N
240 CLS
250 REM

```

```

260 REM *****
270 REM Introduce clave y mues
   tra el primer menu.
280 REM *****
290 REM
300 INPUT " Introduce la C
LAVE" LINE c$: IF c$=
" " THEN GO TO 300
310 INPUT " 1 Codificar mens
   aje"
320 INPUT " 2 Traducir codigo"
330 INPUT " 3 Cargar texto desde cin
   ta"
340 INPUT " 4 Borrar el texto en
   memoria"
350 INPUT " 5 Cambia tex.:
   memo. X salida" (elige opci
   on), LINE m$
360 IF m$="2" AND texmemo=1 THE
N GO TO 700
370 IF m$="2" THEN GO TO 690
380 IF m$="3" THEN GO SUB 1140
: PRINT AT 10,6; FLASH 1;"CARGAN
DO DEL CASSETTE";AT 13,15-LEN (n
$)/2; FLASH 0;n$: LOAD n$ DATA e
$( ); LET texmemo=1; CLS : GO TO
310
390 IF m$="4" THEN LET texmemo
=0; GO TO 310
400 IF m$="5" THEN LET texmemo
=1; LET e$=s$
410 IF m$<>"1" THEN GO TO 310
420 IF texmemo=1 THEN GO TO 45
0
430 REM
440 REM *****
450 REM Introduce mensaje y
   lo codifica.

```


ESPECIFICAMENTE PARA QL



INMEDIATAMENTE
DISPONIBLE

ESPECIFICACIONES

MODELO:
CUB 1451/DQ3
14" QL MONITOR
Entrada RGB-TTL
Resolución (PIXELS)
653 (H) x 585 (V)
DOT PITCH 0.43 mm
Bandwidth 18 MHz
Específicamente diseñado
para el QL que
aprovecha su facilidad
única de 85 columnas.

Mod. 1451/DQ3
P.V.P. 96.000 Ptas.

MICROVITEC 653
cub
COLOUR DISPLAYS

DISTRIBUIDO EN EXCLUSIVA POR MULTILÓGIC, Ramón de Santillán, n.º 15 28016 MADRID Tel. 458 74 75




```

420 REM *****
430 REM
440 INPUT "      Introduce el M
MENSAJE"***** LINE e$: IF e
$="" THEN GO TO 440
450 DIM s$(LEN e$): LET z$="":
LET chrclave=1
460 PRINT AT 12,7; FLASH 1;"CO
DIFICANDO MENSAJE"
470 FOR N=1 TO LEN e$
480 LET y$=c$(chrclave)
490 LET fila=CODE y$-31
500 LET columna=CODE e$(N)-31
510 LET x$=t$(fila,columna)
520 LET z$=z$+x$
530 IF chrclave=LEN c$ THEN LE
T chrclave=1: GO TO 550
540 LET chrclave=chrclave+1
550 NEXT N
560 LET s$=z$
570 REM
580 REM *****
590 REM Imprime codificacion.
600 REM *****
610 REM
620 CLS : PRINT "MENSAJE(entrad
a):";e$;"CODIGO(salida):";s$
630 GO TO 940
640 REM
650 REM *****
660 REM Introduce codigo y lo
traduce.
670 REM *****
680 REM
690 INPUT "      Introduce el
CODIGO"***** LINE e$: IF e
$="" THEN GO TO 690
700 DIM s$(LEN e$): LET z$="":
LET chrclave=1
710 PRINT AT 12,7; FLASH 1;"TRA
DUCION DE CODIGO"
720 FOR N=1 TO LEN e$
730 LET y$=c$(chrclave)
740 LET fila=CODE y$-31
750 FOR J=1 TO 96
760 IF t$(fila,J)=e$(N) THEN L
ET y$=CHR$(J+31): GO TO 780
770 NEXT J
780 LET z$=z$+y$
790 IF chrclave=LEN c$ THEN LE
T chrclave=1: GO TO 810
800 LET chrclave=chrclave+1

```

```

810 NEXT N
820 LET s$=z$
830 REM
840 REM *****
850 REM Imprime traduccion.
860 REM *****
870 REM
880 CLS : PRINT "CODIGO(entrada
):";e$;"MENSAJE(salida):";s$
890 REM
900 REM *****
910 REM Imprime y ejecuta el
segundo menu.
920 REM *****
930 REM
940 PRINT #1;"Pulsa una tecla p
ara MENU": PAUSE 0: CLS
950 INPUT "      A Copiar el text
o en cinta"***** B Escribir t
exto (impresora)***** C Sali
r del programa***** D Nueva
clave***** E Menu anterior"
***** (elige opcion)", LINE m$
960 IF m$="C" OR m$="c" THEN S
TOP
970 IF m$="D" OR m$="d" THEN
GO TO 300
980 IF m$="B" OR m$="b" THEN L
PRINT s$: GO TO 950
990 IF m$="E" OR m$="e" THEN G
O TO 310
1000 IF m$<>"A" AND m$<>"a" THEN
GO TO 950
1010 REM
1020 REM *****
1030 REM Graba el texto.
1040 REM *****
1050 REM
1060 GO SUB 1140: PRINT AT 10,6;
FLASH 1;"VOLCANDO AL CASSETTE";
AT 13,15-LEN (n$)/2; FLASH 0;n$
1070 SAVE n$ DATA s$():
1080 CLS : GO TO 950
1090 REM
1100 REM *****
1110 REM Nombra el texto.
1120 REM *****
1130 REM
1140 CLS : INPUT "Nombre del tex
to ? "; LINE n$: IF n$="" THE
N GO TO 1140
1150 RETURN

```


ULTIMO AVISO

¿Eres aficionado a la programación?
¿Dominas el código máquina?
¿Tienes programas originales?
¿Puedes escribir un buen juego?
¿Quieres ganar dólares, libras, francos o pesetas desde tu casa, en tus horas libres?

NO TE LO PIERDAS!

Contacta inmediatamente con:

CIBERCOMP, S. A.

Tels. (91) 200 21 00
(91) 759 22 44

Especialistas en software para Home Computers,
asociados con primeras firmas internacionales.



- Ordenadores personales Hard y Soft.
- Cursos de Basic.

Oficinas: **RENOVACION EN MARCHA, S.A.**

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

Teléfono (91) 441 24 78

Tienda: **REM SHOP 1**

c/ Galileo, 4 - MADRID-15

Teléfono (91) 445 28 08



MULTISYSTEM, S. A.

BOUTIQUE INFORMATICA

- Ordenadores Personales.
- Micro-ordenadores de gestión.

Todas las novedades en:

Programas - Periféricos - libros

(nacionales y de importación)

**Para: Spectrum - Dragón - Base 64
Spectravideo - Oric - Commodore, etc.**

C/ San Vicente, 53. **ALICANTE.** Tel. (965) 21 55 66.

TRANS-EXPRESS

Copiará tus programas de:

- 1) CASSETTE A MICRODRIVE
- 2) MICRODRIVE A MICRODRIVE
- 3) CASSETTE A CASSETTE
- 4) MICRODRIVE A CASSETTE

En Basic, C/M. o mixtos, protegidos o no, con o sin
cabecera, hasta 48,4K de long. y de "una sola vez".

TRANS-EXPRESS - 1.600 ptas. + gastos envío.

compumania

Pelayo, 12 - 08001 BARCELONA

Tfno.: 301 47 00 - Ext. 20

ANUNCIESE por MODULOS

MADRID (91) 733 96 62 BARCELONA (93) 301 47 00



microgesa

**ESPECIALISTAS EN SINCLAIR
SAQUE RENTABILIDAD AL SPECTRUM**

PROGRAMAS EN MICRODRIVE ZX GESTION:

Contabilidad (P.N.C.)	12.000	pts.
Base de Datos	6.000	"
Proceso de textos (español)	6.500	"
Calc (hoja electrónica)	4.000	"
Control stock y facturación	8.500	"

P. TECNICOS

Agente de Bolsa	6.500	"
Mediciones y presupuestos	24.000	"

PROGRAMAS EN CASSETTES EDUCATIVOS:

Geografía I	1.900	pts.
Geografía II	1.900	"
Curso de Contabilidad I	2.200	"
Curso de Contabilidad II	2.200	"
Curso de Contabilidad III	2.200	"
Geometría y Trigonometría	2.200	"
Superdesarrollos 1X2	3.900	"

(imprime boletos con Impresora Admate)

ORDENADORES: Spectrum, Spectrum + Spectravideo, Commodore, Oric, Katson... desde 1.239 pesetas al mes

IMPRESORAS: Star, New Print, Seikosha desde 774 ptas. al mes. Monitores. Accesorios.

MODEM TELEFONICO - AMPLIACIONES DE MEMORIA - LAPIZ OPTICO - CURRAH (S. DE VOZ)

ENVIOS CONTRA REEMBOLSO, GIRO O TALON CONFORMADO C/ Silva, 5-4.º. Tel. 242 24 71 - 28013 MADRID

DE TODA CONFIANZA

ASI ES **HISSA**

Por algo es el Servicio Oficial INVESTRONICA para los productos SINCLAIR

SIN SOBRESALTOS.

Gracias al "COSTE ESTANDAR POR REPARACION" siempre sabes, de antemano, lo que cuesta el reparar tu microordenador SINCLAIR, una vez caducada la garantía de tu equipo.

Sin presupuestos previos, sin gastos adicionales, tenga lo que tenga tu microordenador, por mucho que sea, el coste siempre será el mismo según el siguiente cuadro:

Además tienes la garantía de que tu equipo será reparado por expertos técnicos y con piezas originales SINCLAIR

ZX 81:	3.150 Ptas.
Spectrum 16K:	5.250 Ptas.
Spectrum 48K:	6.300 Ptas.

DELEGACIONES HISSA

C/. Aribau, n.º 80, piso 5.º 1.º
Telfs.: (93) 323 41 65 - 323 44 04
08036 BARCELONA

P.º de Ronda, n.º 82, 1.º E
Telf.: (958) 26 15 94
18006 GRANADA

C/. Universidad, n.º 4 - 2.º 1.º
Telf.: (96) 352 48 82
46002 VALENCIA

C/. San Sotero, n.º 3
Telfs.: 754 31 97 - 754 32 34
28037 MADRID

C/. 19 de Julio, n.º 10 - 2.º local 3
Telf.: (985) 21 88 95
33002 OVIEDO

Avda. de Gasteiz, n.º 19 A - 1.º D
Telf.: (945) 22 52 05
01008 VITORIA

C/. Atares, n.º 4 - 5.º D
Telf.: (976) 22 47 09
50003 ZARAGOZA

C/. Avda. de la Libertad, n.º 6. Bloq. 1.º Entf. Izq. D.
Telf.: (968) 23 18 34
30009 MURCIA

C/. Hermanos del Río Rodríguez, n.º 7 bis
Telf.: (954) 36 17 08
41009 SEVILLA

C/. Travesía de Vigo, n.º 32 - 1.º
Telf.: (986) 37 78 87
6 VIGO

HORARIO DE ATENCION AL PUBLICO: de 9 h. a 13 h. (excepto Madrid: de 8 1/2 h. a 17 1/2 h.)

AMPLIAMOS POR UN AÑO LA GARANTIA DE TU SINCLAIR

Si tu microordenador SINCLAIR aún está con la GARANTIA INVESTRONICA vigente y deseas ampliarla por un año más (a partir de la fecha de caducidad de la misma), nada más sencillo:

HISSA te amplía la garantía por el mismo importe de lo que te costaría una reparación.

Rellena el cupón con todos los datos y envíalos, con todo lo que se te indica, a la delegación HISSA de MADRID.

A los pocos días recibirás tu NUEVA GARANTIA.

CUPON

D. con domicilio en
calle/plaza n.º teléfono D.P.
desea ampliar en UN AÑO la garantía de su equipo SINCLAIR, cuya GARANTIA INVESTRONICA aún está vigente.
La fecha de compra del microordenador fue el día de de 198
Para ello adjunta, a este cupón, la GARANTIA INVESTRONICA y un talón nominal a HISSA por el siguiente importe, que señala con una X.

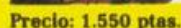
<input type="checkbox"/>	ZX 81:	3.150 Ptas.
<input type="checkbox"/>	Spectrum 16K:	5.250 Ptas.
<input type="checkbox"/>	Spectrum 48K:	6.300 Ptas.

"Enviar el cupón
a HISSA.
C/. San Sotero, 3.
28037 MADRID".

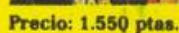
Firmado:

Todospectrum

Ha seleccionado ocho excelentes programas de juego para su ordenador **SPECTRUM**



Precio: 1.550 ptas.



Cada envío llevará 95 Ptas. en concepto de gastos.



Programar, lo que se dice programar, lo puede hacer cualquiera. Hacerlo bien ya es un poco más difícil. Y si

además se pretende la comercialización del producto, lo que empezó siendo una diversión puede fácilmente convertirse en una pesadilla. Lejos de los ejemplos conocidos más allá de nuestras fronteras, donde el joven programador confiesa con amplia sonrisa su rápido enriquecimiento, aquí tendrá que luchar por el reconocimiento de su trabajo.

Programando entre amigos

UN pequeño apartamento céntrico en Barcelona es el cuartel general de este nutrido grupo de jóvenes programadores. A primera vista parece sólo un grupo de amigos que se reúnen para hablar sobre una pasión común. "Empezamos tres, haciendo programas para nosotros. Uno dominaba el inglés, otro el código máquina... Ahora hemos crecido considerablemente en número, pero seguimos trabajando un poco aislados, viéndonos mensualmente para comentar los problemas y discutir los programas en conjunto. Esperamos centralizar aquí todo el tema de máquinas y programas durante este año".

Combinando su tiempo con sus estudios o con el trabajo, hay veces en que deben quitar horas al sueño, especialmente durante los meses pesados, con motivo de las fiestas navideñas. "Es el mayor problema. Trabajamos a contrarreloj, sacando horas de donde podemos. En vacaciones hemos llegado a trabajar trece horas diarias, incluidos los domingos". Los más pequeños tienen alguna limitación. "Yo lo que me dejen, necesito la televisión de casa y la empleo siempre que mis padres no la

ven". Otros van más lejos: "Empecé mañana y tarde hasta que se reventó el transformador...". Una y otra vez se sucedían anécdotas, quizá tan numerosas como sus propios programas.

"El problema es encontrar un tema atractivo. A los que programamos nos es muy difícil porque nos falta un poco de imaginación. Tenemos unos cauces muy marcados en cuanto a la lógica, pero la imaginación es algo muy distinto. A veces ves por dónde van los tiros, pero hay que hacer un cuidadoso estudio". Una vez elegido el tema, se pasa al código máquina, pero tenemos más ilusión que medios. Este año esperamos contar con más medios, con lo que se nos facilitará bastante la labor. Trabajar en código máquina es muy duro tal como lo venimos haciendo. A veces hemos dado con las rutinas de gráficos corriendo un programa que se colgaba siempre. Al final te dabas cuenta de que te habías olvidado definir una variable".

De lo que desde luego no se olvidaban era del gran esfuerzo a realizar en la comercialización de sus programas. "El distribuidor prefiere hacer el dinero fácil, buscar el programa que ya tiene éxito fuera

y traerlo aquí. Lo que se debe hacer es buscar programadores aquí en España, puesto que los hay y muy buenos. Los de fuera tienen las mentes más cuadradas. Piensa que las rutinas de grabación rápida sólo existen en España. Es un error pensar que el programa número 1 en Inglaterra tenga que gustar aquí. Una vez fuimos a unos grandes almacenes y les enseñamos los cassettes y las carátulas nuevas, y al ver las cajas nos dijeron que era fabuloso. Sólo pudimos responderles que el programa estaba dentro. Lo único que les interesa es la presentación".

A pesar de las dificultades, se respiraba un ambiente de confianza en el futuro. Antonio Pascual, en cierta medida responsable del grupo, comparte las relaciones públicas del mismo con su trabajo en un banco y parece conocer bien el sector. "Lo que más nos preocupa del mercado es el abuso que se hace de los programadores. En el momento en que un programa se vende mínimamente, como un programa extranjero, la tirada mínima alcanza los 500 cassettes. Y si éstos se venden, los programadores deben llevarse 200.000 pts., y esto ha de estar claro. El que ahora existan personas que paguen 40.000 pts. por programa y hagan tiradas de 500 o más cassettes, es engañar al programador. Y el programador no es tonto. El resultado es un abandono de la programación. Se debe ir con la verdad por delante y ellos han de tener un *royalty* sobre el programa. En todas partes está funcionando así menos en España. Los pocos que se deciden a aprovechar al programador están abusando. Pero no sólo se abusa del programador, sino también del comprador. Existe una casa aquí en Barcelona que te vende un programa que no funciona. Cuando vas a reclamar te dice que te lo cambia por otro del mismo precio, con lo que en realidad aseguran la venta. Esto es un fraude".

Quizá aquí resida el secreto de su éxito. Que el usuario sepa lo que compra y que funcione, y que el programador reciba lo que se merece por su trabajo.

LX

REVISTA PARA LOS USUARIOS
DE ORDENADORES SINCLAIR

Programas
de ajedrez

Gráficos
de funciones

Conexiones
con el P I/O

YA ESTA A LA VENTA



REGALAMOS
ESTE EQUIPO

El Hobbit



LIBERA TU IMAGINACION.

Da rienda suelta a tu fantasía
porque las diversiones
más emocionantes te esperan en:

sinclair store
SOMOS PROFESIONALES

BRAVO MURILLO, 2 (aparc. gratuito en C/. Magallanes, 1). Tel.: 446 62 31
DIEGO DE LEON, 25 (aparc. gratuito en C/. Núñez de Balboa, 114). Tel.: 261 88 01 MADRID