

VIDEO BASIC

20 LECCIONES DE BASIC
PARA APRENDER CON EL C-64



INGELEK



JACKSON

TV y monitor
Pantalla y memoria de imagen
Colores y atributos
Funciones de control
de la impresión en pantalla
TAB, POS, SPC
Números aleatorios con RND
El modo comillas
Vidoejercicios
Videojuego n.º 6

6

COMMODORE
C-64

VIDEO BASIC

Una publicación de
INGELEK JACKSON

Director editor por INGELEK:

Antonio M. Ferrer

Director editor por JACKSON HISPANIA:

Lorenzo Bertagnolio

Director de producción:

Vicente Robles

Autor: Softidea

Redacción software italiano:

Francesco Franceschini,

Stefano Cremonesi

Redacción software castellano:

Fernando López, Antonio Carvajal,

Alberto Caffarato, Pilar Manzanera

Diseño gráfico:

Studio Nuovaidea

Ilustraciones:

Cinzia Ferrari, Silvano Scolari,

Equipo Galata

Ediciones INGELEK, S. A.

Dirección, redacción y administración,

números atrasados y suscripciones:

Avda. Alfonso XIII, 141

28016 Madrid. Tel. 2505820

Fotocomposición: Espacio y Punto, S. A.

Imprime: Gráficas Reunidas, S. A.

Reservados todos los derechos de reproducción y
publicación de diseño, fotografía y textos.

©Grupo Editorial Jackson 1985.

©Ediciones Ingelek 1985.

ISBN del tomo 2: 84-85831-18-7

ISBN del fascículo: 84-85831-14-4

ISBN de la obra completa: 84-85831-13-6

Depósito Legal: M-15075-1985

Plan general de la obra:

20 fascículos y 20 casetes, de aparición quincenal,

coleccionables en 5 estuches.

Distribución en España:

COEDIS, S. A.

Valencia, 245. 08007 Barcelona.

INGELEK JACKSON garantiza la publicación de todos
los fascículos y casetes que componen esta obra y el
suministro de cualquier número atrasado o estuche
mientras dure la publicación y hasta un año después de
terminada.

El editor se reserva el derecho de modificar
el precio de venta del fascículo,
en el transcurso de la obra, si las circunstancias del
mercado así lo exigen.

Julio, 1985.

Impreso en España.

INGELEK



JACKSON

SUMARIO

HARDWARE 2

Televisores y monitores. Pantalla
y memoria de imagen. Los
atributos y los colores.

EL LENGUAJE 18

Funciones de control de la
impresión en pantalla. SPC, TAB,
POS, RND.

LA PROGRAMACION 26

El modo comillas.

VIDEOEJERCICIOS 32

Introducción

*Entre otras cosas, tu ordenador es
también una emisora de televisión. Es
capaz de transmitir por cable las
informaciones que ha elaborado.*

*Desde este punto de vista, el
programador se convierte en el
«director» de la salida de los datos
tratados.*

*Por lo tanto, él es el responsable de
proporcionarle a las informaciones de
salida el máximo relieve y la mayor
claridad posibles.*

*De aquí deriva la importancia de
conocer, tanto el hardware dedicado a
la visualización de las imágenes,
como las sentencias y funciones del
BASIC capaces de manejar el formato
y los atributos.*

*La mayor parte del éxito de tus
futuros programas depende de tu
familiaridad con estos elementos.*

Televisores y monitores

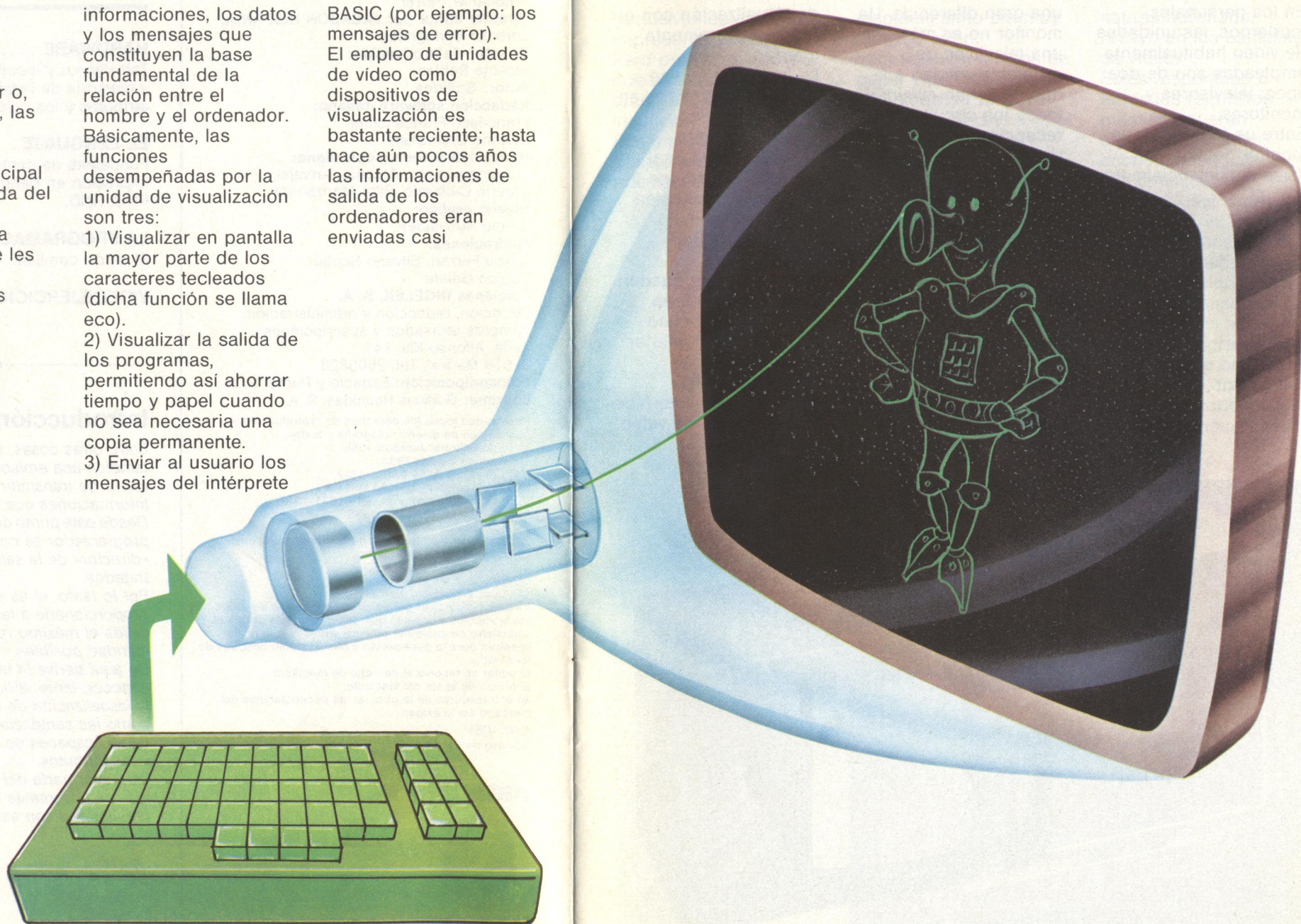
Televisor y monitor o, más generalmente, las unidades de visualización, constituyen el principal dispositivo de salida del ordenador.

Generalmente, es a estas a quienes se les asigna la tarea de visualizar todas las

informaciones, los datos y los mensajes que constituyen la base fundamental de la relación entre el hombre y el ordenador. Básicamente, las funciones desempeñadas por la unidad de visualización son tres:

- 1) Visualizar en pantalla la mayor parte de los caracteres tecleados (dicha función se llama eco).
- 2) Visualizar la salida de los programas, permitiendo así ahorrar tiempo y papel cuando no sea necesaria una copia permanente.
- 3) Enviar al usuario los mensajes del intérprete

BASIC (por ejemplo: los mensajes de error). El empleo de unidades de vídeo como dispositivos de visualización es bastante reciente; hasta hace aún pocos años las informaciones de salida de los ordenadores eran enviadas casi



En los personales modernos, las unidades de vídeo habitualmente empleadas son de dos tipos: televisores y monitores.

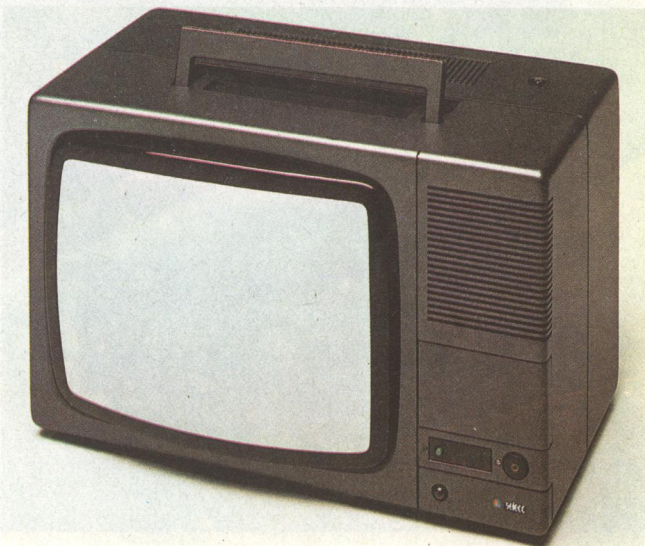
Entre un televisor y un monitor no existe ni física ni sustancialmente

una gran diferencia. Un monitor no es más que una televisión de excelente calidad a la que se le han eliminado todos los circuitos de recepción de la señal a través de antena.

Naturalmente, la calidad de la imagen es superior a la que se puede obtener con un televisor normal. Para un uso no profesional puede no ser necesario (y hasta inútil) recurrir a la compra de un monitor, puesto que el televisor doméstico es capaz de desempeñar perfectamente el trabajo

de visualización con un precio seguramente inferior.

La forma en que se producen los textos en la pantalla de tu C64 es bastante sencilla; existe una zona de la memoria RAM en la cual están depositados —bajo forma de códigos ASCII— todos los caracteres que pueden ser presentados en pantalla. Un circuito especial, en el interior de tu C64, hace referencia a ella, realizando su «interface» con la unidad de vídeo.



exclusivamente a impresoras y teletipos. Pero pronto se advirtió que estos dispositivos eran absolutamente insuficientes para hacer frente a un volumen de trabajo creciente. Además, su coste de

mantenimiento era muy elevado (eran necesarias montañas de papel y permanentes atenciones) y tenían una capacidad y velocidad de visualización bastante limitadas con respecto a las exigencias de los usuarios.

Fue así, como se pensó en complementar las siempre necesarias impresoras con unidades de salida más adecuadas y flexibles en su uso que las empleadas hasta el momento.

La elección, como ya

habrás adivinado, recayó en las pantallas de vídeo.

Estas respondían a todos los requisitos planteados: eran compactas, fiables, económicas (poco mantenimiento y mínimos gastos de uso), y rápidas.

Desde entonces, progresivamente, su empleo ha sido cada vez más intenso; en nuestros días es casi imposible lograr encontrar un ordenador que no esté dotado de su correspondiente unidad de visualización.



Este circuito toma todos los caracteres presentes en las localizaciones de la memoria de pantalla y los envía al circuito del televisor, bajo forma de impulsos eléctricos compatibles con el sistema o «standard» de televisión.

El televisor (o el monitor) produce una imagen visible a partir de estas señales eléctricas.

La manera en que esto ocurre, constituye una de las aplicaciones más interesantes de la física electrónica y tiene como componente

fundamental un dispositivo llamado tubo de rayos catódicos, que es un tubo al vacío, es decir, un contenedor de

vidrio en cuyo interior se ha hecho el vacío. Dentro de este tubo se encuentra un «cañón electrónico» (basado en

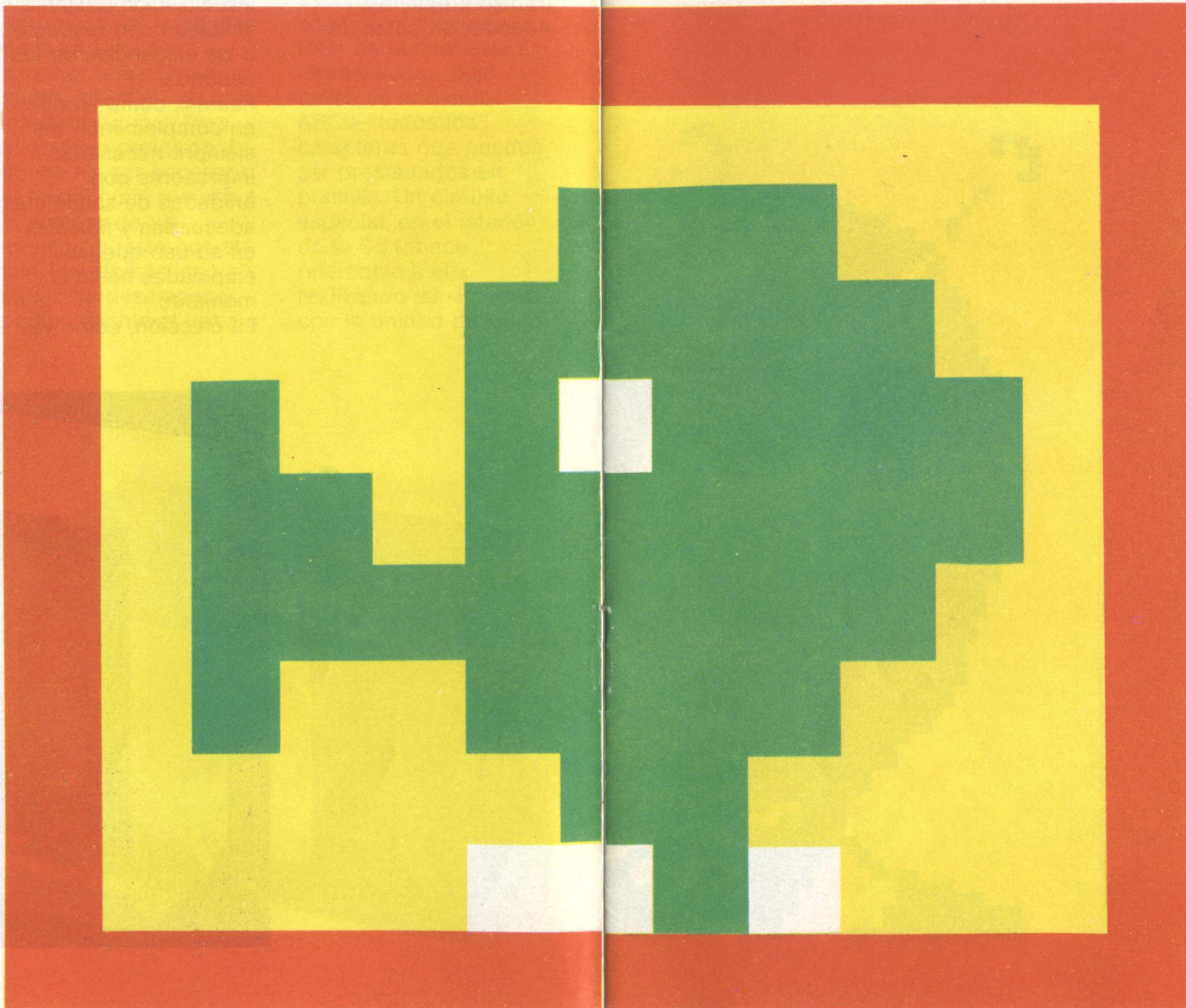
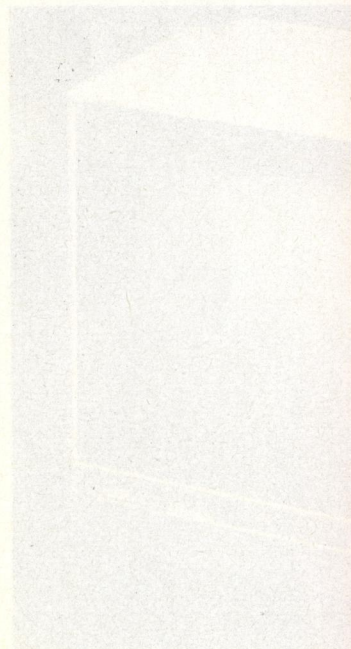
un filamento calentado por la corriente que lo atraviesa, como en las bombillas) que produce un haz (o pincel) de

electrones muy delgado. Los electrones gozan de una propiedad singular: cuando

chocan con determinadas sustancias fluorescentes —fósforo— provocan que éstas generen una luminiscencia, cuya duración puede oscilar entre algunos milisegundos (milésimas de segundo) y algunos segundos, en función del tipo de fósforo y de la intensidad del haz electrónico.

La imagen se reconstruye a raíz precisamente de este hecho: el haz de electrones, «disparado» por el «cañón» y guiado por los oportunos campos eléctricos y magnéticos, desplazándose de derecha a izquierda y de arriba a abajo, aplica mayor o menor intensidad a cada uno de los puntos de una pantalla que ha sido recubierta por una finísima capa de fósforo, provocando una mayor o menor luminosidad.

La baja resolución emplea únicamente los caracteres (normales o gráficos) existentes en el teclado. Las imágenes así obtenidas no resultan muy detalladas.



La imagen, por lo tanto, es reconstruida punto por punto por el pincel electrónico del tubo de rayos catódicos, de la misma forma en la que el ojo humano lee la página impresa de un periódico o de un libro. La velocidad con la que el pincel recorre la totalidad de la pantalla es tan extremadamente elevada, que no puede ser percibida siquiera mínimamente por el observador humano: el estándar europeo de televisión prevé que toda la pantalla, subdividida en 625 líneas horizontales (525

en el estándar americano) sea recorrida completamente cada cincuentavo de

segundo. El fenómeno de la persistencia de la imagen en la retina se encarga de



proporcionar una imagen completa y simultánea. En el interior del televisor (o del monitor)

existen, como ya se ha indicado, circuitos previstos para guiar los movimientos del haz de electrones tanto en

sentido horizontal como vertical.

Para que la imagen sea visible es necesario que estos dispositivos trabajen

simultáneamente.

En la señal de video están compredidas, además de las informaciones referentes a la intensidad de cada punto de la pantalla, también señales especiales para coordinar el movimiento del pincel electrónico. Son las señales de sincronismo.

En las transmisiones de televisión, estas informaciones llegan desde la cámara; en cambio, para los ordenadores existe un circuito que, al igual que una cámara, «lee» la imagen que tiene que visualizar en las distintas posiciones de la memoria de pantalla. El tipo de fósforo aplicado a la superficie de la pantalla determina el color del punto de

En alta resolución puedes direccionar cada punto (pixel) determinando o no su encendido. Las imágenes así obtenidas resultan muy detalladas.

colisión entre el pincel electrónico y el propio fósforo.

Existen en el mercado distintos tipos de monitores: de fósforo verde, amarillo, ámbar, etc. La elección de un color u otro depende normalmente de una cuestión de gustos y preferencias

personales, aunque últimamente lleguen continuos rumores (y desmentidos) sobre la mayor o menor fatiga visual producida por uno u otro tipo de fósforo.

En cambio, es muy importante la elección de las dimensiones de la pantalla. Un error en el que se incurre con frecuencia es creer que cuanto mayor sea la pantalla, mejor es la visión.

La imagen, sea cual sea la dimensión de la pantalla, siempre se subdivide en 625 líneas horizontales: en una pantalla más pequeña las líneas serán más finas que en una grande. La dimensión óptima de la pantalla, será aquélla en la que el ojo humano, situado a una distancia normal de visión, ya no consiga distinguir unas de otras. La dimensión en pulgadas de la pantalla expresa la longitud de la diagonal, indicada precisamente en pulgadas: los ordenadores personales y los terminales de video tienen normalmente pantallas de 9" o 12" (9 o 12 pulgadas).

Las pantallas en color

merecen un tratamiento aparte.

Permaneciendo idénticos los principios de la televisión monocromática, es decir, de un solo color (habitualmente llamada «de blanco y negro» aunque sea verde o amarilla), una pantalla en color basa su funcionamiento en un fenómeno óptico: todos los colores existentes se pueden obtener mediante la mezcla y combinación de tres colores, por esta razón llamados fundamentales: el rojo, el verde y el azul.

En el interior del tubo, existen tres «cañones electrónicos» en lugar de uno. La superficie del tubo está recubierta por completo por centenares de millares de puntos de fósforo, dispuestos en grupos de tres, capaces cada uno de emitir luz roja, verde o azul.

Con el mismo movimiento, ya explicado para la pantalla monocromática (de derecha a izquierda y de arriba a abajo), se mueven ahora tres haces electrónicos en lugar de uno; cada uno de ellos —pasando a través de una pantalla

metálica adecuadamente perforada— es capaz de golpear únicamente el fósforo de un determinado color. Sobre la superficie de la pantalla aparecen por lo tanto tres imágenes

simultáneas, que son, respectivamente, de color rojo, verde y azul, y que, combinándose entre sí proporcionan al ojo humano una sola imagen en colores. Esta técnica se llama síntesis aditiva.

Pantalla y memoria video

Hemos visto ya el proceso mediante el cual las imágenes se visualizan sobre la pantalla de tu televisor. Trataremos ahora de profundizar en la relación que existe entre todo aquello que se encuentra en la memoria de tu C64, y lo que puedes ver representado en pantalla.

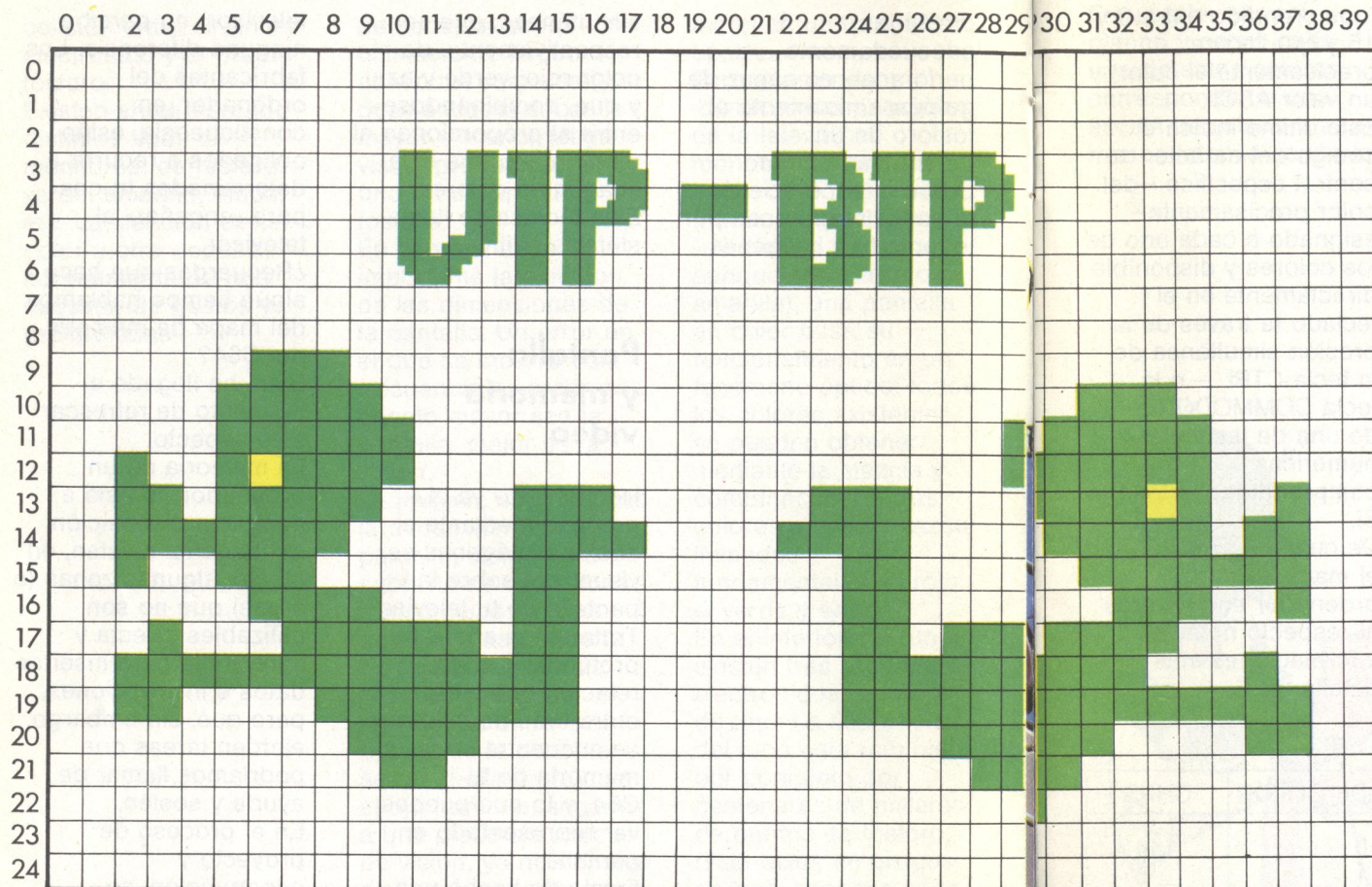
Existe de hecho una estrecha relación entre el contenido de la memoria y la formación de las imágenes. El ordenador, para generar una imagen en pantalla, debe producir (como ya hemos dicho) una señal similar a la de una cámara de televisión, de tal forma que el monitor (o el

televisor) no perciba ninguna diferencia. Los fabricantes del ordenador, en consecuencia, están obligados a recurrir a determinados trucos para «engañar» al televisor.

¿Recuerdas que hace algún tiempo hablamos del mapa de memoria del C64?

Bien, ha llegado el momento de refrescar este aspecto.

La memoria de un ordenador no está a completa disposición del usuario: existen, en efecto, algunas zonas (o áreas) que no son utilizables directa y libremente para insertar datos e instrucciones, pero que, sin embargo, ejercen tareas que podríamos llamar de ayuda y sostén. En el proceso de proyecto y construcción, se confiaron a estas parcelas de la memoria labores no exactamente de «elaboración», pero tan importantes como éstas para el correcto funcionamiento del aparato. Así, algunas localizaciones se han destinado a contener los programas, otras el intérprete BASIC, e incluso otras —y éstas



son las que nos interesan ahora— los caracteres de visualización en pantalla. El área de la memoria de la que nos vamos a ocupar ahora lleva el nombre de memoria de pantalla. Su objeto es

contener todas las informaciones que son necesarias para construir una imagen sobre la pantalla de un televisor. ¿Crees que esta memoria será RAM o ROM? No deberías tener demasiados

problemas para adivinarlo. Forzosamente, si tenemos en cuenta que lo que queremos es modificar el contenido de la pantalla, la memoria vídeo debe pertenecer a la zona de memoria RAM. En ella

deben tener lugar los cambios siempre que, por ejemplo, ejecutes una sentencia PRINT o INPUT, o en el caso de que (excepto en ocasiones particulares) teclees algo que haya de ser visualizado en pantalla.

Pero, por si sola, la memoria de pantalla no puede hacer mucho: es necesario que un circuito especial de vídeo llamado de «refresh» (es decir, refresco) de la imagen, lea —exactamente igual que una cámara— toda la memoria, carácter por carácter. A partir de ese momento, el circuito de vídeo, conociendo ya los códigos de los caracteres a visualizar, consulta a una memoria especial ROM (llamada también generador de caracteres) de la que toma la «descripción» gráfica del carácter mismo. Como resultado final, se obtiene una señal de vídeo que indica si cada punto de la pantalla debe permanecer encendido o apagado. Todo lo que aparece en pantalla es representado bajo la forma de determinadas combinaciones de puntos luminosos, llamados pixel (abreviatura del inglés de picture element). Cada punto, pudiendo estar únicamente apagado o encendido, se puede describir con un único bit. Sin entrar demasiado

en detalles técnicos, te será suficiente con saber que el circuito de refresco «toma» la combinación de bits que contiene la descripción de un carácter y la coloca en la casilla correspondiente de la pantalla, encendiendo o apagando los puntitos como ya se ha indicado. En total, los pixels luminosos de que dispones en tu C64 son 64.000. A ellos les corresponde, en conjunto, una pantalla compuesta por: 25 líneas y 40 columnas, que forman en total $25 \times 40 = 1000$ caracteres.

Los atributos y los colores.

La pantalla es idealmente, y también físicamente, divisible en 1000 (25 líneas de 40 caracteres) posiciones donde los caracteres pueden ser impresos. Cada uno de ellos está representado por un cuadrado de 8×8 puntos. Cada uno de estos caracteres se puede

representar, además, de distintas maneras, por ejemplo: en color, en inverso y en positivo. Es decir, puedes establecer sus atributos. Así, cuando imprimes algo en la pantalla, no haces más que modificar la combinación de alguno de los atributos anteriormente asignados a aquella posición. Normalmente, es decir, cuando no indicas especificaciones concretas —por ejemplo, el color—, el único atributo que sufre modificaciones es aquél referente al encendido o apagado de los pixels que componen el carácter. Todos los demás atributos permanecen inalterados. Pero existe también la posibilidad de variar a placer,

mediante las instrucciones oportunas, todos los atributos. Pero desafortunadamente, tu C64 no dispone de instrucciones específicas destinadas a ello. Es por tanto, necesario recurrir al mágico POKE que, alterando el contenido de determinadas posiciones te permite obtener igualmente los resultados deseados. Hablaremos en primer lugar de los colores disponibles en tu C64. Como podrás ver en la tabla de pie de página, a cada uno de ellos se le asocia un número

(comprendido entre 0 y 15 y que indica precisamente el color) y un valor ASCII. Este último indica el código del carácter de control específico —del color precisamente— asignado a cada uno de los colores y disponible directamente en el teclado (a través de la presión simultánea de la tecla CTRL — o la tecla COMMODORE— y de una de las teclas numéricas comprendidas entre 1 y 8). De cualquier manera, en el manual de tu ordenador encontrarás al respecto notas e informaciones más detalladas.

COLOR	NUMERO DE COLOR	CHR\$
NEGRO	0	144
BLANCO	1	5
ROJO	2	28
CYAN	3	159
PURPURA	4	156
VERDE	5	30
AZUL MARINO	6	31
AMARILLO	7	158
NARANJA	8	129
MARRON	9	149
ROJO CLARO	10	150
GRIS-1	11	151
GRIS-2	12	152
VERDE CLARO	13	153
AZUL	14	154
GRIS-3	15	155

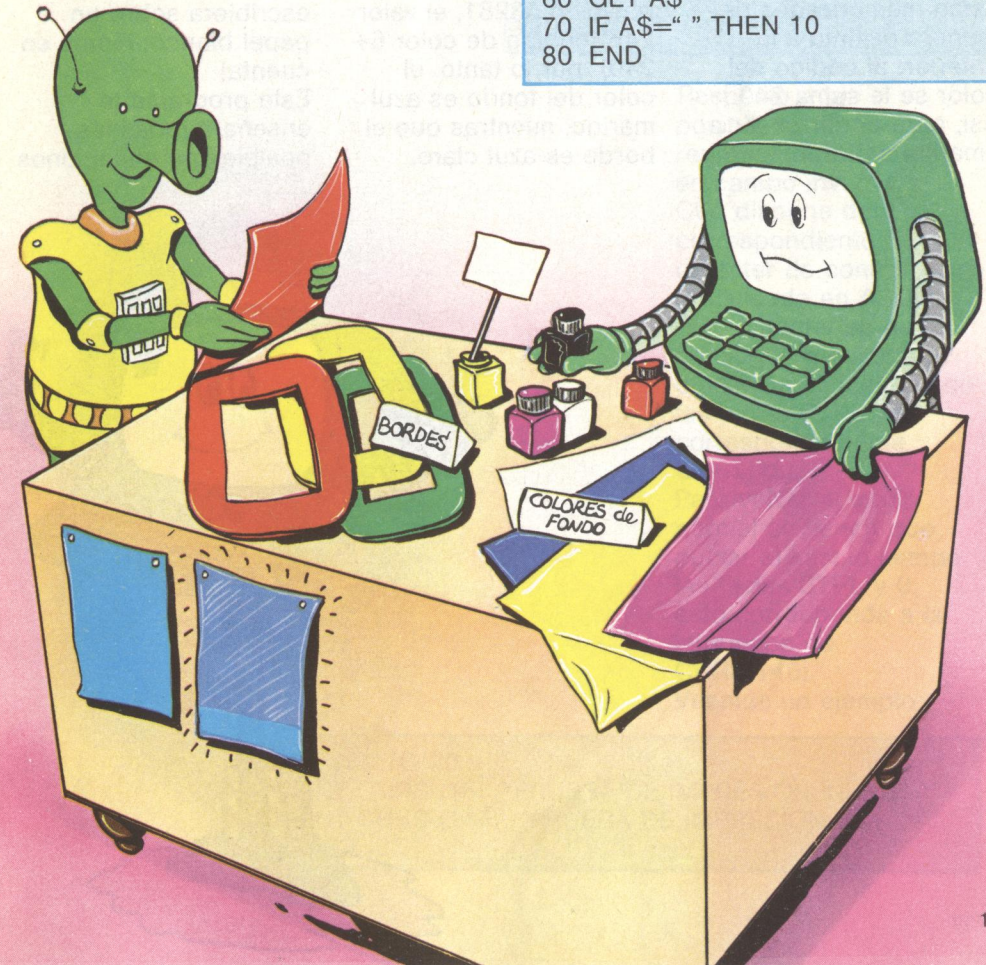
En un televisor en blanco y negro cada uno de estos colores corresponden a una escala de grises, más o menos intensa según la

tonalidad del color. La localización que contiene el valor del color de los caracteres es la 646. Modificando el valor contenido en ella es posible cambiar

el color de todo lo que sea visualizado en adelante. El siguiente programa te aclarará las ideas (para terminar pulsa una tecla cualquiera):

```

10 FOR I=0 TO 15
20 POKE 646,I
30 PRINT "PRUEBA DE IMPRESION"
40 FOR J=0 TO 50:NEXT J
50 NEXT I
60 GET A$
70 IF A$=" " THEN 10
80 END
    
```



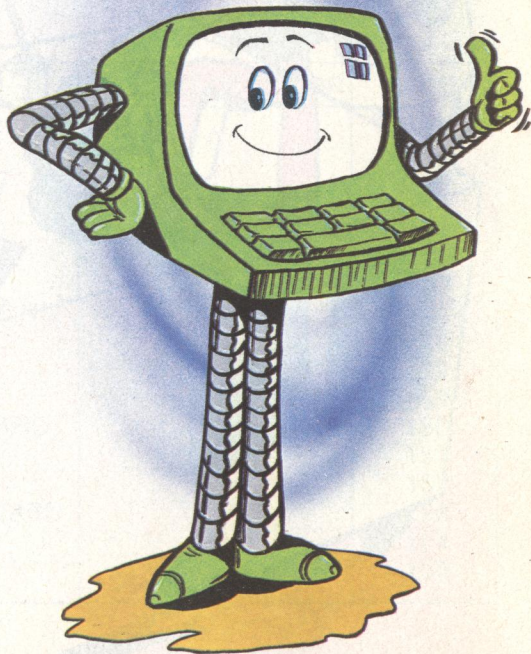
En el momento del encendido, en la posición 646 está presente el valor 14; por lo tanto, el color del cursor será azul marino. Las posiciones que contienen los valores del color del borde y del fondo son respectivamente la 53280 (color del borde) y la 53281 (color del fondo). Pero en estas, en cambio, los códigos están memorizados de manera distinta a la anterior: al código del color se le suma 240. Así, para poder pasar a amarillo el borde, será

necesario sumar al número correspondiente al amarillo —es decir, el 7— el valor 240, y ejecutar a continuación la instrucción:

```
POKE 53280,247
```

En el momento del encendido, en la posición 53280 existe el valor 254 (código de color 14+240) y en la posición 53281, el valor 246 (código de color 6+240): por lo tanto, el color del fondo es azul marino, mientras que el borde es azul claro,

igual que el cursor. Un último detalle. Si el código del fondo y el de los caracteres coincidieran —es decir, si el fondo y los caracteres tienen el mismo color— ocurriría un hecho fácilmente imaginable: no se leerá nada. Los caracteres se escriben con el mismo color del fondo, exactamente igual que si un pincel blanco escribiera sobre un papel blanco. ¡Téno en cuenta! Este programa te enseñará todas las posibles permutaciones



de los colores disponibles en pantalla

(y como antes, para terminar pulsa cualquier tecla):

```
10 FOR I=0 TO 15
20 POKE 53280,I
30 FOR J=0 TO 15
40 POKE 53281,J
50 FOR K=0 TO 15
60 PRINT "PRUEBA DE IMPRESION"
70 GET A$
80 IF A$ <> " " THEN GOTO 120
90 NEXT K
100 NEXT J
110 NEXT I
120 END
```

Respecto a la posibilidad de representar caracteres en campo inverso, tu C64 dispone del correspondiente carácter de control (codificado en ASCII). Este carácter se puede conseguir mediante la pulsación simultánea de las teclas CTRL y 9, correspondiéndole CHR\$(18). Para volver al modo normal se tienen que pulsar al mismo tiempo las teclas CTRL y 0; esto corresponde a la impresión de CHR\$(146). Veamos un ejemplo:

```
10 FOR I=0 TO 20
20 PRINT CHR$ (18); "PRUEBA DE IMPRESION INVERSA"
30 PRINT CHR$ (146); "PRUEBA DE IMPRESION NORMAL"
40 NEXT I
```


Funciones de control de la impresión en pantalla

En el ámbito de las instrucciones de visualización previstas por el BASIC, revisten

una especial importancia todas aquellas sentencias que controlan y modifican, al gusto del programador, la posición del cursor y, en consecuencia, la de los letreros en pantalla. Hasta ahora hemos empleado la instrucción PRINT en numerosas ocasiones, usándola para visualizar todos los resultados, los mensajes y los letreros que hayan sido necesarios cada vez. Pero lo que todavía nos falta es la capacidad de controlar completamente esta instrucción, lo que nos permitiría, por ejemplo, conseguir que los resultados de salida estuvieran colocados en una determinada posición de pantalla, o bien ordenados y encolumnados con formatos no impuestos necesariamente por el ordenador. Dicho de otra manera (con una palabra muy empleada en el campo informático), deseamos saber formatear los textos en pantalla. Es pues éste el objetivo de las instrucciones que ahora nos disponemos a aprender. Y es importante

subrayar que todas estas instrucciones no le indican al ordenador qué imprimir, sino únicamente DONDE imprimir. ¿Has entendido la diferencia? Veámoslas ahora una a una, junto a algunos ejemplos aclaratorios.

SPC

La función SPC() se emplea en el interior de instrucciones PRINT para escribir un cierto número de espacios. El argumento numérico asignado a la función, especifica el número de espacios a escribir. SPC() permite, por lo tanto, desplazar el cursor a lo largo de una línea tantos espacios como se desee. El argumento que le proporcionas habrá de ser un valor numérico entero comprendido entre 0 y 255 (si hubiera cifras decimales, estas serán eliminadas automáticamente); en el caso contrario, tu C64 te enviará el mensaje de error:

```
ILLEGAL QUANTITY  
ERROR
```

Ejemplos

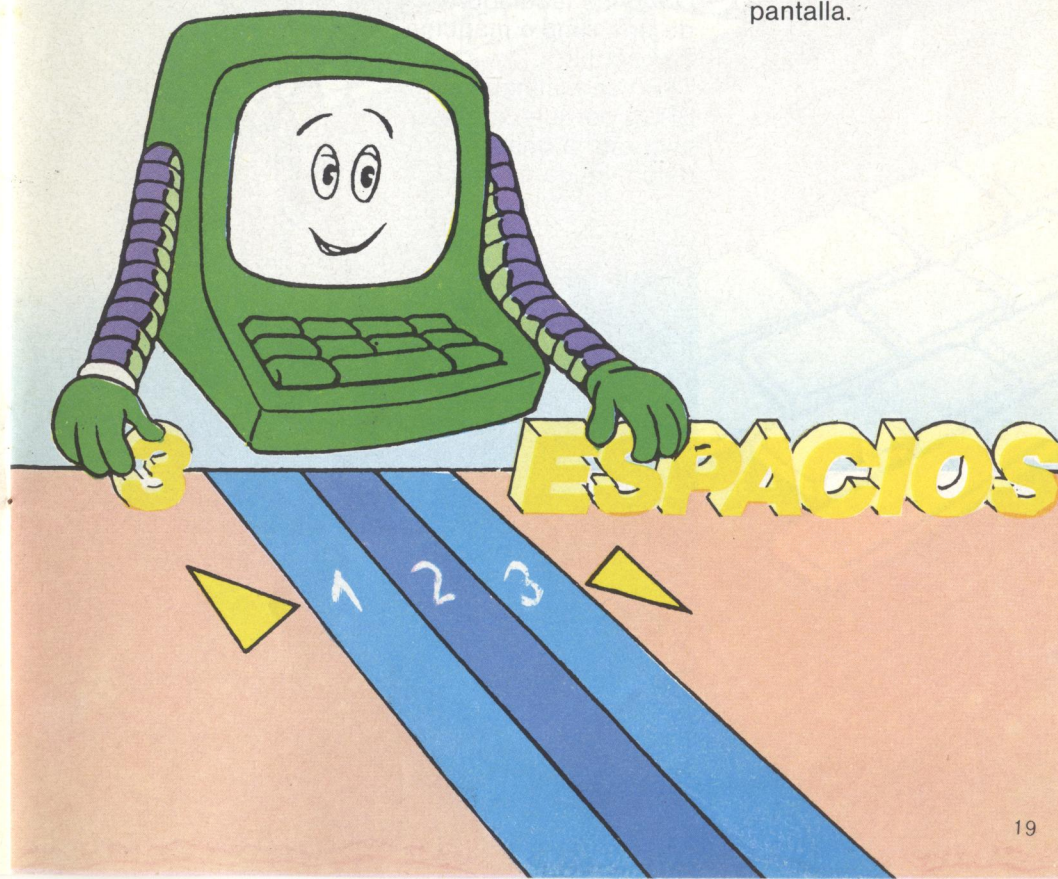
```
PRINT "ROJO"; SPC (4); "DE"; SPC (3); "ATARDECER"
```

Esta instrucción interpone cuatro espacios entre las palabras ROJO y DE, mientras que entre las palabras DE y ATARDECER inserta tres

espacios. Obseva como detrás de cada SPC aparece un punto y coma. Si hubiera existido una coma obtendríamos también su efecto, obteniendo en pantalla unas palabras aún más distanciadas.

```
10 FOR I=0 TO 18  
20 PRINT SPC(I); "ADIOS"  
30 NEXT
```

Este breve programa imprime en cambio una serie de saludos, desplazándolos progresivamente en pantalla.



Sintaxis de la función

SPC (expresión)

donde expresión ha de ser un valor numérico entero entre 0 y 255.

en modo absoluto (es decir, empezando a contar desde la número 0), mientras que SPC() lo desplaza un cierto número de columnas en modo relativo (es decir,

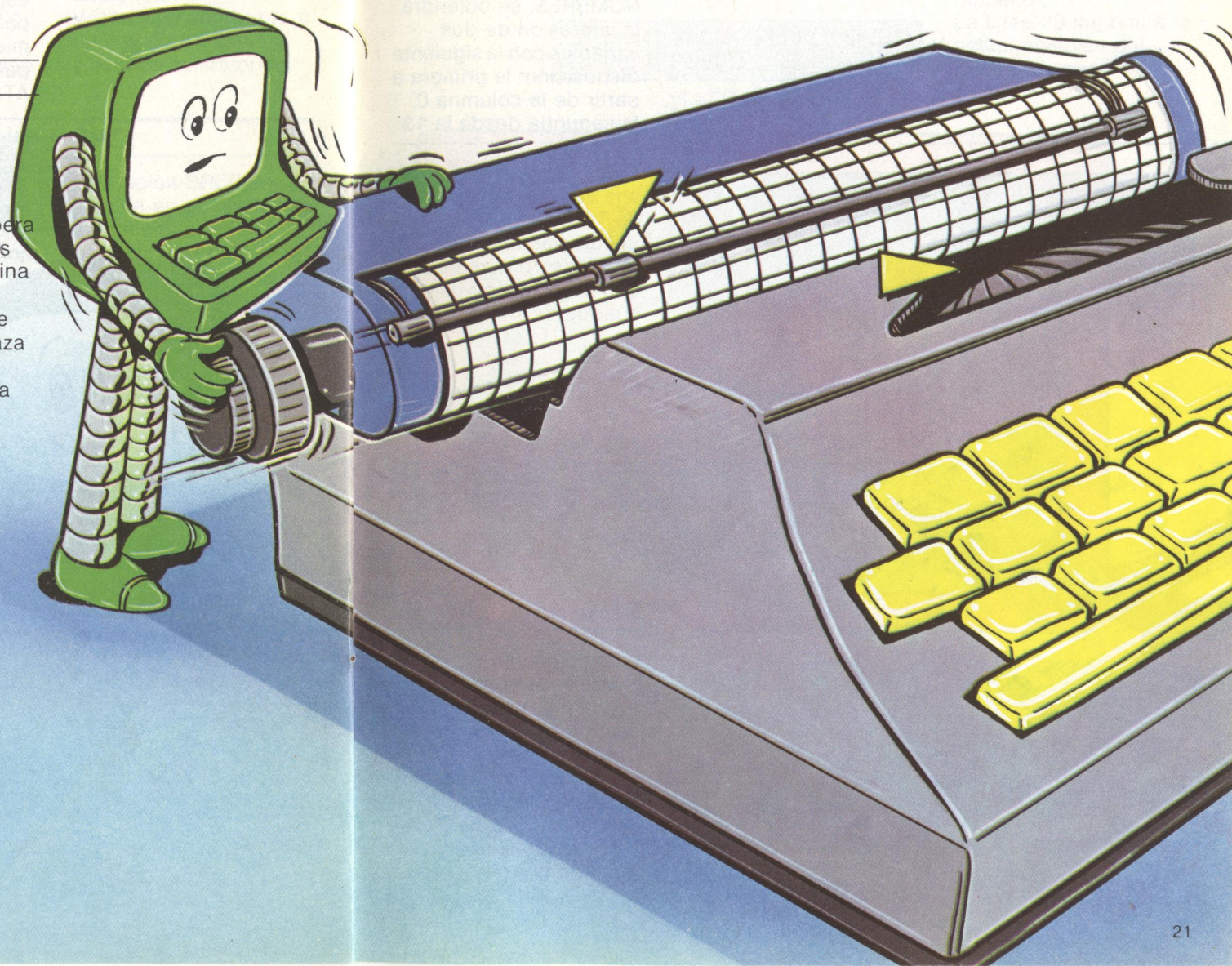
contando los espacios a partir de la posición actual de impresión. La función TAB() se emplea normalmente para alinear las impresiones en

columnas verticales, alineándolas en puntos preestablecidos. Supón que tengas que visualizar algunos datos en pantalla, alineados en un orden

determinado. Gracias a TAB() puedes evitar los aburridos y a veces complicados cálculos para encolumnar con exactitud todos los elementos.

TAB

También la función TAB() se emplea únicamente en el interior de PRINT: opera como los tabuladores de una simple máquina de escribir. TAB() se distingue de SPC() porque desplaza el cursor a una determinada columna



Ejemplos

```
PRINT NOMBRE$; TAB (13); APELLIDO$
```

Con esta instrucción, e independientemente de la longitud de la cadena contenida en NOMBRE\$, se obtendrá la impresión de dos variables con la siguiente disposición: la primera a partir de la columna 0, la segunda desde la 13.

```
10 PRINT TAB (2); "NUMERO"; TAB (12);  
"CUADRADO"  
20 FOR I=1 TO 15  
30 PRINT TAB (4);I;TAB (15);I*2  
40 NEXT
```

Este programa muestra una sencilla aplicación de TAB: imprime encolumnados los primeros 15 números con sus respectivos cuadrados.

Sintaxis de la función

TAB (expresión)

donde expresión es un valor numérico comprendido entre 0 y 255.

POS

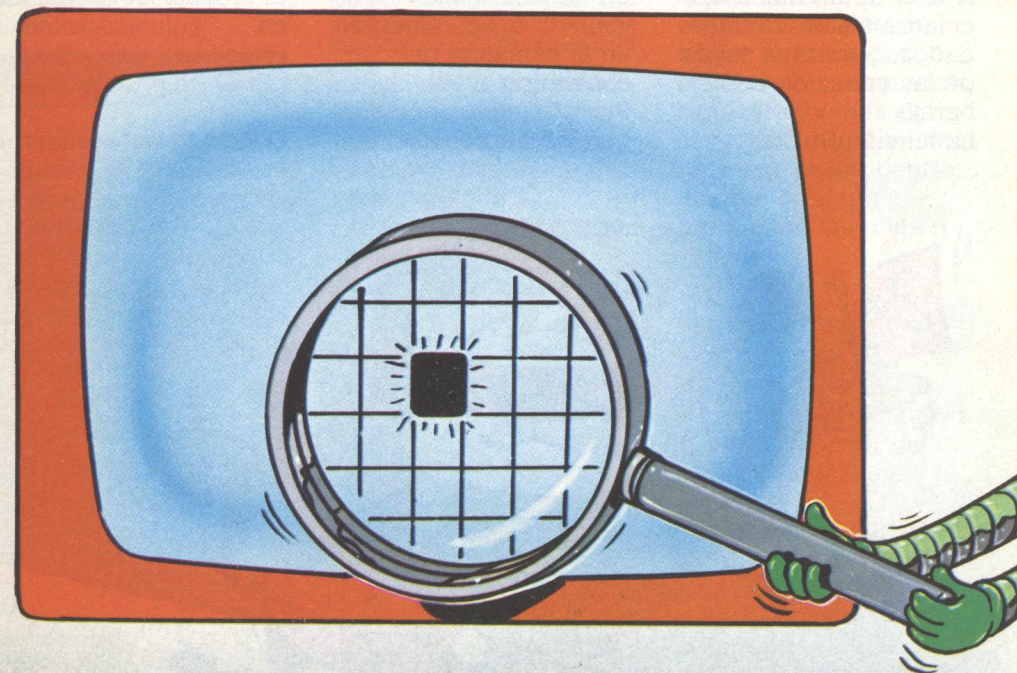
La función POS() devuelve la posición actual del cursor en sentido horizontal, es decir, la columna sobre la que será impreso el siguiente carácter. Pero ésta considera una línea de pantalla como formada por dos líneas

de 40+40 caracteres; da por lo tanto como resultado un número comprendido entre 0 y 79. El argumento de la función (iincreible pero

cierto!) no tiene ninguna importancia, pero habrá de ser sintácticamente correcto (normalmente se emplea el 0). Por ejemplo, el breve programa siguiente:

```
10 PRINT SPC(5);"PERRO";SPC(5)  
20 PRINT POS(0)
```

provocará la visualización de la palabra PERRO precedida y seguida por 5 espacios (que, naturalmente, le resultan invisibles al ojo humano). La línea 20 imprimirá la actual posición del cursor, es decir $5+4+5=14$



Sintaxis de la función

POS (argumento)

donde argumento es un valor numérico puramente formal, sin ninguna función específica.

RND

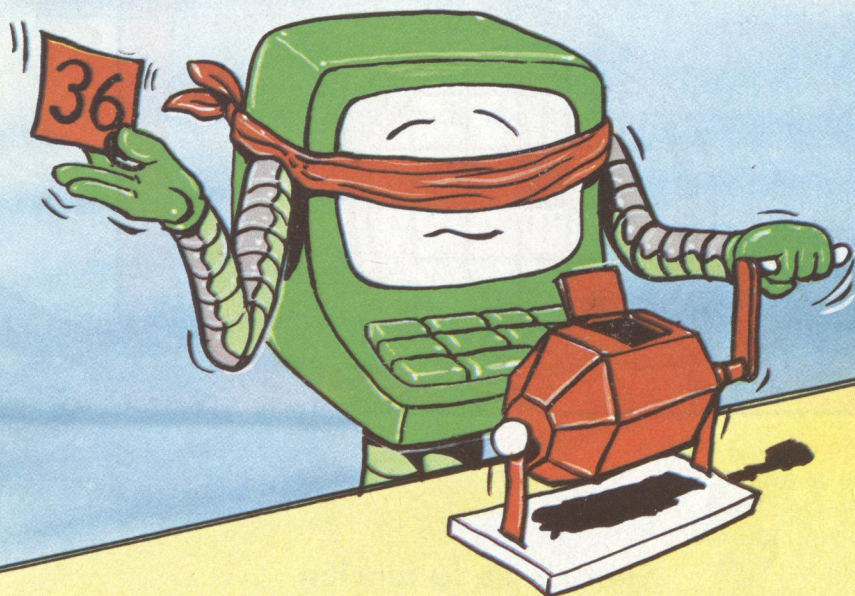
No es raro encontrar casos en los que sea necesario disponer de números aleatorios dentro de un programa. Por ejemplo, podrías necesitar un algoritmo que simule la extracción al azar de un número, o el lanzamiento de unos dados, o quizá la salida de las cartas de una baraja. La función RND te

resuelve excelentemente este tipo de problemas: crea números al azar (random, en inglés) comprendidos entre 0 y 1 ($0 \leq R < 1$). Tu C64 produce una secuencia de números aleatorios realizando una serie de operaciones a partir de un número inicial llamado base, generado en el momento del encendido. Puesto que los números generados por RND son

el resultado de una compleja serie de operaciones, sería más correcto hablar de números pseudoaleatorios. El argumento de RND controla la base de la función y determina el punto de origen de la serie de números aleatorios. El formato de la función es:

```
RND (S)
```

Donde S representa un



número cualquiera. El valor de S determina el comportamiento de la función:

- 1) Si S es un número positivo, RND generará una serie imprevisible de números.
- 2) Con S=0 se obtendrá igualmente una serie de números aleatorios, pero generados mediante distintos cálculos, basados esta vez sobre el reloj interno de tu C64.
- 3) Para un valor de S negativo, RND devolverá siempre la misma serie de números pseudoaleatorios: en otras palabras, puedes obtener la misma serie de números recurriendo a RND y proporcionándole siempre el mismo argumento negativo.

Por ejemplo, el siguiente programa producirá siempre la misma serie de números aleatorios independientemente de cuando hagas el RUN y de cuantas veces lo hagas:

```
10 PRINT RND (-1)
20 FOR I = 1 TO 5
30 PRINT RND (1)
40 NEXT I
```

La llamada a la función RND en la línea 10, gracias al argumento negativo, determina una serie prefijada y previsible de números. Como resultado, los

números aleatorios obtenidos en el interior del BUCLE del ciclo FOR...NEXT serán siempre los mismos 5. Si, en cambio, borras la línea 10, el punto de origen de la serie será distinto cada vez, así como los cinco números generados. Puesto que la gama de números del 0 al 1 no es adecuada para la mayor parte de las aplicaciones, emplea la siguiente fórmula para establecer por ti mismo el mismo rango dentro del que deberán producirse los números aleatorios:

```
LET R = INT ((LS - LI + 1) * RND (1)) + LI
```

Donde R significa el número aleatorio, LS es el límite superior del rango y LI el límite inferior. Para números aleatorios enteros, entre 1 y LS (variable numérica a definir con anterioridad), aplicando la fórmula obtendrás:

```
LET R = INT (LS * RND (1)) + 1
```

Sintaxis de la función

RND (expresión numérica)

El modo comillas

Ya hemos visto en algunas ocasiones como la presión simultánea de algunas parejas de teclas provoca la ejecución inmediata de algunas acciones y operaciones especiales: por ejemplo, borrado de

pantalla, modificación de colores, desplazamiento del cursor. A cada una de estas acciones le corresponden un código ASCII determinado. Por lo tanto, se pueden



emplear libremente en el interior de los programas, especificando su código respectivo. Así, si deseáramos borrar la pantalla y llevar el cursor a la parte superior izquierda, podríamos indicar:

```
PRINT CHR$(147)
```

Pero además siempre es posible incorporar en una cadena los caracteres especiales que corresponden a estas originales (pero indispensables) instrucciones. Cuando en el curso de

un programa la cadena sea impresa, se podrá obtener así el mismo efecto que con el código de control asignado a través de la función CHR\$. En el programa siguiente podrás comprobar todas estas cosas. Opera así:









- coloca dentro de las cadenas A\$, B\$, C\$, D\$ aquellos caracteres que se generan pulsando simultáneamente CTRL y las teclas numeradas del 1 al 4 (y que controlan 4 de los 16 colores disponibles en tu C64).
- imprime estas cadenas una tras otra, alternándolas con un ciclo de espera, para dejar tiempo y darte así cuenta del efecto de cada una de ellas.









```
10 A$="█":B$="▣":C$="▣":D$="▣":E$="ADIOS"
20 PRINT A$; E$: FOR I=0 TO 100: NEXT
30 PRINT B$; E$: FOR I=0 TO 100: NEXT
40 PRINT C$; E$: FOR I=0 TO 100: NEXT
50 PRINT D$; E$: FOR I=0 TO 100: NEXT
60 GO TO 20
```

















Este modo de insertar los caracteres de control en las cadenas, es una peculiaridad típica del C64. Se acostumbra a llamarle "modo comillas",

precisamente porque en el interior de las comillas encuentra la razón de su existencia. En la tabla que sigue, ilustramos el significado de los caracteres que permiten moverse por la pantalla y controlar el

color. En la primera columna aparecen las teclas a pulsar, en la segunda el carácter visualizado en pantalla en modo comillas, y en la tercera, su efecto una vez impresos.

CLR/HOME		Manda el cursor a la esquina superior izquierda de la pantalla, sin borrar su contenido. Corresponde a CHR\$(19).
SHIFT + CLR/HOME		Limpia la pantalla y manda el cursor a la esquina superior izquierda. Corresponde a CHR\$(147).
CRSR/flechas-verticales ↓		Desplaza el cursor una posición hacia abajo. Corresponde a CHR\$(17).
SHIFT + CRSR/ flechas-verticales ↑		Desplaza el cursor una posición hacia arriba. Corresponde a CHR\$(145).
CRSR/flechas-horizontales →		Desplaza el cursor una posición a la derecha. Corresponde a CHR\$(29).
SHIFT + CRSR/ flechas-horizontales ←		Desplaza el cursor una posición a la izquierda. Corresponde a CHR\$(157).
CTRL + 9		Permite visualizar en campo inverso los caracteres siguientes. Corresponde a CHR\$(18).
CTRL + 0		Pone fin a la secuencia de caracteres en campo inverso. Corresponde a CHR\$(146).

CTRL + 1		Negro
CTRL + 2		Blanco
CTRL + 3		Rojo
CTRL + 4		Cyan
CTRL + 5		Púrpura
CTRL + 6		Verde
CTRL + 7		Azul marino
CTRL + 8		Amarillo

 + 1		Naranja
 + 2		Marrón
 + 3		Rojo claro
 + 4		Gris 1
 + 5		Gris 2
 + 6		Verde claro
 + 7		Azul
 + 8		Gris 3

Para modificar los datos presentes en pantalla tienes que llevar el cursor a la posición deseada y después, reescribir encima, o reinsertar, o borrar caracteres. Las teclas para insertar o borrar son:

INST/DEL		Borra el carácter a la izquierda del cursor y desplaza el cursor y todo lo que le siga una posición a la izquierda. Corresponde a CHR\$(20).
SHIFT + INST/DEL		Crea un espacio a la izquierda del cursor, desplazando el cursor y todo lo que le siga una posición a la derecha. Corresponde a CHR\$(148).

Lista de gastos diarios

El siguiente programa totaliza los gastos diarios por diversos conceptos. Para finalizar la entrada de datos, introduce 0, 0.

Obseva que para obtener una correcta visualización de los datos se ha recurrido a la función TAB.

```

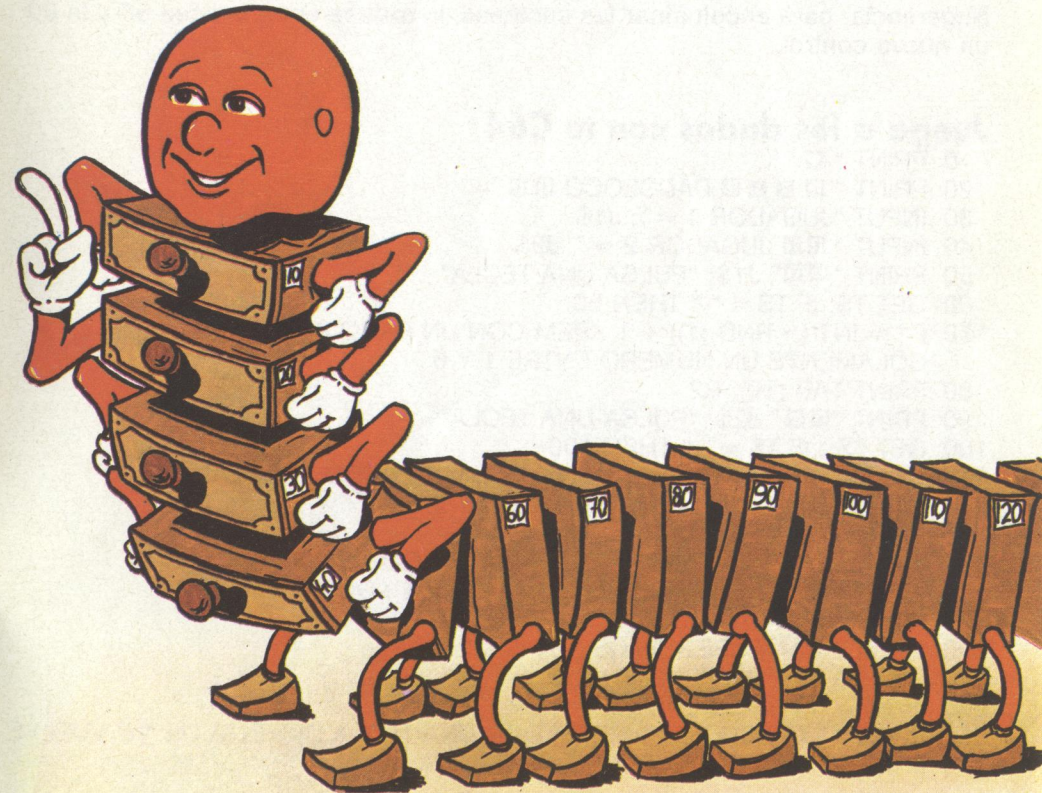
GASTOS DEL DIA
ALIMENTOS 3.200
CINE 300
REVISTAS 600
TELEFONO 1.000
TOTAL 5.100
  
```

GASTOS DEL DIA

ALIMENTOS	3.200
CINE	300
REVISTAS	600
TELEFONO	1.000
TOTAL	5.100

```

10 LET T = 0
20 LET A$ = "  " : REM LA PRIMERA LINEA DE IMPRESION ES LA CUARTA DESDE EL LADO
30 PRINT "  " : REM IMPRIME 24 ESPACIOS PARA BORRAR LA LINEA DE INPUT
40 INPUT "  DESC, IMPORTE": D$, I
50 IF I = 0 THEN PRINT "TOT"; TAB (22); T : END
60 PRINT A$; D$; TAB (22); I
70 LET A$ = A$ + "  " : GO TO 30
  
```



EJERCICIOS

Usando la función TAB, prueba un nuevo método para imprimir la tabla de multiplicar de un número, de forma que las unidades, decenas, etc., de los distintos productos queden perfectamente encolumnados. Para ayudarte he aquí un ejemplo de la tabla del 7.

```
10 C = 10: S = C: REM LAS UNIDADES SE IMPRIMEN EN LA COLUMNA 10
20 PRINT "  "
30 FOR V = 1 TO 10
40 LET P = V * 7
50 IF P > 9 THEN S = C - 1
60 PRINT TAB (S); P
70 NEXT V
80 END
```

Sustituye ahora la línea 30 con: 30 FOR V = 1 TO 20.

Sugerencia: para encolumnar las centenas, introduce entre la línea 50 y la 60 un nuevo control.

Juega a los dados con tu C64

```
10 PRINT "  "
20 PRINT "  DADOLOCO  "
30 INPUT "JUGADOR 1 = "; J1$
40 INPUT "  JUGADOR 2 = "; J2$
50 PRINT "  " J1$; "PULSA UNA TECLA"
60 GET T$: IF T$ = " " THEN 60
70 R1 = INT ( * RND (1)) + 1 : REM CON UN DADO PUEDE SALIR
  SOLAMENTE UN NUMERO ENTRE 1 Y 6
80 PRINT TAB (10); R2
90 PRINT "  " J2$; "PULSA UNA TECLA"
100 GET T$ : IF T$ = " " THEN 100
110 R2 = INT (6 * RND (1)) + 1 : REM CON UN DADO PUEDE SALIR
  SOLAMENTE UN NUMERO ENTRE 1 Y 6
120 PRINT TAB (&10); R2
130 IF R1 = R2 THEN PRINT "  EMPATE": GO TO 200
140 IF R1 > R2 THEN PRINT "  GANA"; J1$ : GO TO 200
150 PRINT "  GANA"; J2$
200 PRINT "  ¿SEGUIMOS? S/N"
210 GET T$ : IF T$ = " " THEN 210
220 IF T$ = "N" THEN END
230 PRINT "  " : GO TO 50 : REM CON UNA TECLA DISTINTA DE "N" VUELVE
  A COMENZAR EL JUEGO
```

Modifica las líneas 70 y 110 para simular el lanzamiento de 2 dados piensa cuáles son los números máximo y mínimo que pueden salir.

SEIKOSHA SP-800

El fruto de la Investigación



La nueva impresora de SEIKOSHA SP-800, con un ordenador personal puede escribir **96 combinaciones de letra diferentes**, desde 96 caracteres por segundo a 20 con muy alta calidad de letra, además es gráfica en alta densidad.

Su precio es de **69.900 R** con **introduccion automático hoja a hoja**.

Con un pequeño ordenador personal, un procesador de textos puede costar alrededor de cien mil pesetas.

Infórmese y comprenderá por qué las máquinas de escribir tienen demasiados años.

Nuestra calidad es "SEIKO";

nuestros precios, únicos

Si desea más información,

consulte con nuestro distribuidor más cercano, llame o escriba a:

DIRECCION COMERCIAL:
AV. Blasco Ibañeta, 114-116
46022 VALENCIA
Tel. (96) 372 88 89
Telex 62224

DIRECCION COMERCIAL EN CATALUNA:
C/Muntaner, 68-2-4Pta
08011 BARCELONA
Tel. (93) 323 32 19

Este pie de página ha sido realizado íntegramente con la nueva impresora:

SEIKOSHA SP-800

CON LA NUEVA SP-800
"LAS MÁQUINAS DE ESCRIBIR PRONTO DESAPARECERÁN"

TRADICIONAL EN CANTONAS ALABARCA SAVALA SE CONSIDERA INSTITUCIONAL EN LAS
CATEGORÍAS DE MÁQUINAS DE ALTA DENSIDAD Y EN LA CATEGORÍA DE MÁQUINAS DE ALTA
DENSIDAD DE LA COMPAÑÍA DE MÁQUINAS DE ESCRIBIR SEIKOSHA SAVALA SAVALA
DEBIDO A SU ALTA CALIDAD Y SU PRECIO. COMO RESULTADO DE SU INVESTIGACIÓN Y DESARROLLO
CONSTRUYÓ LA MÁQUINA DE ESCRIBIR PARA SU PROPIOS Y DE OTROS.

Características

Modelo de introducción automática hoja a hoja.
Funciona en modo de impresión por líneas o por caracteres.
Alimentación por cable para poder usar en
modo de impresión por líneas o por caracteres.
Impresión de 96 combinaciones de letra.
Impresión de 96 caracteres por segundo.
Impresión de 20 caracteres por segundo.
Impresión de 10 caracteres por segundo.
Impresión de 5 caracteres por segundo.
Impresión de 2 caracteres por segundo.
Impresión de 1 carácter por segundo.
Impresión de 1 carácter por segundo.
Impresión de 1 carácter por segundo.

Tarifa de precios

Modelo de introducción automática hoja a hoja.
Funciona en modo de impresión por líneas o por caracteres.
Alimentación por cable para poder usar en
modo de impresión por líneas o por caracteres.

Modelo	Velocidad	Columnas	Tipos de letra	P.V.P.R.
GP-508 LA DEL SPECTRUM	40 cps	32	-	19.900
GP-60 LA PEQUEÑA	40 cps	46	2	25.900
GP-600 LA ECONOMICA	50 cps	50	2	47.900
GP-700 LA DE COLOR	50 cps	98-196	3	69.900
SP-800 LA PERFECCION	96 cps	98-137	20	69.900
BP-6200 LA DE OFICINA	200 cps	136-272	18	199.900
BP-6420 LA MAS RAPIDA	420 cps	136-272	18	299.900

ESTOS SON NUESTROS MODELOS:

MODELO	VELOCIDAD	COLUMNAS	TIPOS DE LETRA	P.V.P.R. INTERFAZ PARALELO
GP-508 LA DEL SPECTRUM	40 cps	32	-	19.900
GP-60 LA PEQUEÑA	40 cps	46	2	25.900
GP-600 LA ECONOMICA	50 cps	50	2	47.900
GP-700 LA DE COLOR	50 cps	98-196	3	69.900
SP-800 LA PERFECCION	96 cps	98-137	20	69.900
BP-6200 LA DE OFICINA	200 cps	136-272	18	199.900
BP-6420 LA MAS RAPIDA	420 cps	136-272	18	299.900

* Los precios indicados son los recomendados para conexión tipo paralelo Centronics, para otro tipo de conexión, sufran un ligero incremento.