

TEST: Xerox 820 II, Victor

ORDENADOR

POPULAR

AÑO I - Núm. 4 - Junio 1983 • 250 Ptas.

Suplemento
BITE

**LOGO, un language
para aprender**

**Astrologia:
El Zodiaco en 8 bits**



COMMODORE 64

EPSON

QX-10

GAMA PROFESIONAL



III
TROFEO
INTERNACIONAL
LIDERES EN
TECNOLOGIA
1982



EPSON CENTER

Provenza, 89-91
Tels. 3220354-3220444
BARCELONA

EPSON CENTER

Infanta Mercedes, 62, 2º, 8º
Tels. 270 3707 - 270 3658
MADRID

DISTRIBUIDOR AUTORIZADO

ORDENADOR POPULAR

Director: Norberto Gallego

Año I. Núm. 4. Junio 1983.

Hace sólo dos meses, en el número 2 de esta revista, comentábamos el fenómeno de la proliferación de tiendas de ordenadores en España. Desde entonces, hemos sabido que algunas tiendas —felizmente no aquéllas que entrevistamos— no andan bien. El negocio no es la fiesta que esperaban sus animosos propietarios. Mientras esto ocurre, varios importantes nombres, de la informática mundial tropiezan con dificultades a la hora de montar sus redes comerciales en nuestro país.

¿Exceso de optimismo? ¿Inmadurez del mercado? Creemos, para decirlo con términos económicos, que se trata más de inadecuación de la oferta que de insuficiencia de demanda. Porque todos los datos que manejamos indican que el mercado español de ordenadores personales es pujante, que el parque crece a un ritmo semejante (o superior) al de la media europea.

Traemos a cuento estas reflexiones para situar al lector en las condiciones en que habrán de producirse, a lo largo de este año, interesantes anuncios de productos, cada vez más potentes y cada vez más baratos. De una oferta amplia sólo puede salir ganando el usuario.

Es el caso, concretamente, del **Commodore 64**, al que dedicamos la nota principal de este número. Es el último producto de una firma que busca su camino hacia franjas más altas del mercado, después de haber alcanzado éxito con sus modelos anteriores. Pero el 64 llega a un mercado más exigente, y su éxito dependerá del *software* que sea capaz de ofrecer a los usuarios para

aprovechar sus particulares características. De ello nos ocupamos en las páginas 16 y siguientes.

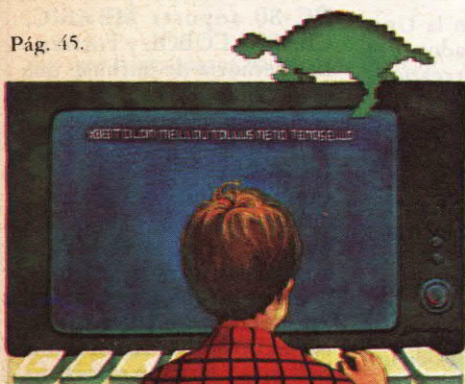
Como decimos siempre, esta revista está dedicada a todas las manifestaciones de la penetración de la informática en la vida cotidiana de la gente. ¿Qué hay de más cotidiano que el deseo de conocer el futuro? ¿Qué hay de más banal que los horóscopos de los diarios? Precisamente, la revista se cierra con un reportaje sobre la conjunción entre astrología y ordenador.

Hasta el próximo número.



Pág. 16.

Pág. 45.



Paulo

Pág. 91.

Ordenador Popular es una publicación mensual de Ediciones y Suscripciones, S. A. • Presidente: Fernando Bolín. Hermosilla, 100. Telf. (91) 435 63 55. Madrid. • Diseño: A. Gordillo • Colaboradores de redacción: Ricardo García, Aníbal Pardo, Andrea Bea, Marisa Cortazzo, Bernardo Díaz, Gumersindo García • Circulación: Luis Carrero • Suscripciones: Antonio Zurdo • Publicidad Madrid: María José Martín. Avda. Cardenal Herrera Oria, 144. Madrid-34. Telf. (91) 734 68 90 • Publicidad Barcelona: Enrique Aliet. Talleres, 62-64. Barcelona-1. Telf. (93) 302 36 48 • Distribuye: Sociedad General Española de Libería. Av. Valdelaparra, s/n, Alcobendas, Madrid • Imprime: Novograph, S. A. Ctra. de Irún, Km. 12,450. Fuencarral, Madrid • Depósito Legal: M-6522-1983.

Solicitado control de



COPYRIGHT © 1983. La reproducción de todos los textos e ilustraciones de esta revista sin autorización previa del editor está prohibida. En el caso de aquellos artículos a cuyo pie figuran las leyendas "© Popular Computing/Ordenador Popular" o "© Byte/Ordenador Popular", los derechos de reproducción están reservados por McGraw Hill Inc. Toda traducción y publicación debe ser autorizada por McGraw Hill Inc., 1221, Avenue of the Americas, New York, NY 10020, USA. La reproducción completa o parcial, por cualquier procedimiento o en cualquier idioma, sin autorización previa, está prohibida. **COPYRIGHT © 1983.** In the case of the articles with following notices: "© Popular Computing/Ordenador Popular" or "© Byte/Ordenador Popular", all rights are reserved by McGraw Hill Inc., 1221 Avenue of the Americas, New York, NY 10020, USA. Reproduction in any manner, in any language, in whole or in part without prior written permission is prohibited.



Fortune 32:16



■ El **Fortune 32:16**, uno de los más originales microordenadores que hayansido lanzados al mercado americano en los últimos dos años, será comercializado en España a partir de las próximas semanas. Por lo que hemos podido saber, una firma con sede en Cataluña ha conseguido de **Fortune Systems** la exclusividad de la importación y, a su vez, ha cedido a otra, la distribución a nivel nacional y soporte técnico de aplicaciones.

Este ordenador, basado en el potente microprocesador **Motorola 68000**, con su arquitectura de 32 bits proporciona cualidades notables de flexibilidad. Se presentará en España en cinco configuraciones distintas, que van desde los 800 Kbytes en discos *floppies* hasta una unidad de disco duro con 20 Mbytes. En principio, los precios deberían ser francamente competitivos con los de otros ordenadores más banales, oscilando entre el millón y los dos millones de pesetas.

Una de las originalidades del **Fortune 32:16** es la de contar con el sistema operativo **UNIX**, desarrollado por Bell Laboratories y que, hasta el momento, se perfila como un futuro estándar de facto para los ordenadores de 16 bits.

El *software* de aplicaciones que acompañará al **Fortune 32:16** en su lanzamiento al mercado español está siendo desarrollado actualmente e incluirá, seguramente, un potente programa de tratamiento de textos, una base de datos capaz de sacar partido de las capacidades del hardware y un *spreadsheet* **Multiplan**.

■ Apenas cerrado nuestro número anterior, en el que dábamos cuenta de los desarrollos que varias compañías estaban haciendo en el campo de la tecnología de los discos ópticos, llega la noticia de que **Burroughs** ha decidido abandonar sus proyectos en esa materia, debido al alto costo de la operación. Esta es, al menos, la explicación oficial. Pero algunas

fuentes de la industria aseguran que **Burroughs** ya tenía listo para entrar en producción un sistema de almacenamiento basado en rayos láser, pero no pudo encontrar la forma de desarrollar simultáneamente un sistema capaz de procesar la información así almacenada.

Hasta el momento de dejar de lado el proyecto, **Burroughs** había invertido en él una suma que la propia compañía se niega a revelar pero que otras fuentes estiman en no menos de 20 millones de dólares. Es posible, sin embargo, que la operación vuelva a actualizarse, si se confirman los pasos dados por otras empresas en este mismo campo, como parece indicarlo la noticia —que publicamos en esta misma sección— acerca de un nuevo sistema de almacenamiento en disco óptico diseñado por **Matsushita**.

■ Entre el 18 y el 29 de julio próximo funcionará la Escuela de Verano de Informática en la Residencia Universitaria de Jarandilla de la Vera, en la provincia de Cáceres. Organizada por la Asociación Española de Informática y Automática,

en colaboración con la Universidad de Extremadura y el Departamento correspondiente de la Facultad de la Complutense, la escuela veraniega va por su quinta edición.

Este año, se ofrecerán cursos sobre Tolerancia de Fallos. Criptografía, Seguridad de la Información, Lenguaje Natural y Representación del Conocimiento, además de actividades de intercambio de información entre grupos, presentación de trabajos, mesas redondas y conferencias sobre temas actuales de informática. Es de esperar que el nutrido programa deje tiempo para gozar del paisaje de la comarca de la Vera.

Los cursos de la Escuela de Verano están dirigidos a profesores universitarios, investigadores y profesionales de la materia. Podrán llevar acompañantes. El precio por participante ha sido fijado en 35.000 pesetas (alojamiento y comida incluidos). Para información adicional e inscripciones, dirigirse a los teléfonos 244 040 (y 244063, correspondientes a la Facultad de Física de la Universidad Complutense de Madrid.

■ Para finales de este año, **Commodore** piensa introducir un equipo para hacer frente al **Lisa** de **Apple**. El nuevo sistema informático tendrá un precio de 4.000 dólares. El nuevo sistema tendrá un "ratón" manual y se comercializará a través de la actual red de **Commodore**, poniéndose en el mercado con la "serie B" una nueva línea de ordenadores comerciales que se presentará durante el verano.

■ El nuevo ordenador personal **BC-80**, fabricado por **Fujitsu**, incluye memoria de burbuja, gráficos de color y una impresora. El **BC-80** consiste en un microprocesador **Z80A**, de 8 bits, que está contenido en el teclado y 64K de RAM, pudiendo funcionar con *software* standar CP/M. Es opcional un **BASIC** en ROM incorporado. El **BC-80** soporta **MBASIC**, **CBASIC**, **COBOL** y **Fortran**. La memoria de burbuja, que es un dispositivo liviano de almacenamiento no volátil, no ha sido aún utilizada en forma intensiva por las empresas de microordenadores debido a su alto-costo. Los precios para la memoria de burbujas han comenzado a bajar, posibilitando su uso en los microordenadores. Los cartuchos de memoria de burbujas permiten al teclado del **BC-80**, convertirse en un ordenador portátil que puede ser enchufado a un televisor de color.

■ **IBM** acaba de presentar el tan esperado **Sistema 36**. Se trata de un pequeño sistema orientado a la gestión, cuyos principales usuarios son las pequeñas y medianas empresas que por primera vez deciden informatizarse, pequeños negocios y departamentos de grandes empresas que incluso ya dispongan de un gran ordenador.

El nuevo sistema se sitúa entre el antiguo **Sistema 34**, que en España comenzó a instalarse en abril del 77, y el **Sistema 38**, considerando como ordenador grande de la gama baja. El usuario potencial del **Sistema 36** es el mis-



Sistema 36, de IBM

mo que hasta ahora ha venido instalando el 34, del cual existen aproximadamente 2.000 unidades funcionando en nuestro país, así como los usuarios del todavía gran parque instalado del Sistema 3, que podrían encontrar así respuesta a sus requerimientos sin necesidad de cambiar de proveedor. Aunque el sustituto para el S/3 debiera ser en muchos casos el S/38, buena cantidad de usuarios optaron por el S/34 y otra parte cambiaron de marca.

El Sistema 36 es totalmente compatible con el 34, pudiendo además controlar hasta 30 estaciones de trabajo locales y 64 remotas, no necesitando ningún tipo de sala especialmente acondicionada.

La memoria central disponible es de 128 K ampliables a 512 K, donde se presenta la interesante característica de que los módulos de memoria son independientes agrupándose de 2 en 2 K, con lo que se obtiene la doble ventaja de que el sistema no se colapse por un fallo en uno de los módulos y la rapidez en el servicio cuando esto ocurra.

Los lenguajes de progra-

mación de que dispone el S/36 son RPGII, Basic, Cobol, Fortran y Ensamblador.

Soporta hasta cuatro líneas de comunicaciones de alta velocidad, pudiendo utilizar los protocolos SNA/SDLC y BSC. El precio medio del sistema es 4.200.000 de pesetas.

■ **American Telephone and Telegraph**, el gigante americano de las telecomunicaciones, acaba de anunciar una serie de acuerdos tecnológicos con tres de los principales productores de semiconductores de Estados Unidos. **Motorola**, **Intel** y **National Semiconductor**, cada uno por su lado, han acordado producir microprocesadores conforme al diseño del sistema operativo **UNIX**, propiedad de **Bell Laboratories**, filial de **AT & T**.

La movida confirma la intención de **AT & T** de estar presente como un activo suministrador de *hardware* y *software*, desde que quedó liberada de las ataduras que, en virtud de la legislación americana antitrust, le impedía invadir el campo de la informática.

Western Electric, la rama de **AT & T** que, en principio, se encargará de la manufactura de *hardware*, lanzará probablemente un conjunto de microprocesadores (tal vez se llamará **Bell Mac**) al mercado a finales de este año, incorporando el sistema operativo desarrollado por **Bell Laboratories**.

■ **Matsushita Electric Industrial Company** ha anunciado lo que al parecer es el primer sistema de disco óptico en el que la información puede borrarse y volver a escribirse.

El desarrollo con éxito de un sistema "borrable", probablemente mejorará la imagen del disco óptico como soporte alternativo a los dispositivos magnéticos, como el disco flexible, para el almacenamiento masivo de información, especialmente para aplicaciones de proceso de archivos en oficinas.

A largo plazo, también podría resultar interesante para los futuros productos de consumo basados en disco óptico, que actualmente incluyen el video-disco láser y el disco digital de audio.

Hasta el presente, estos otros discos no podían ser regrabados, y en consecuencia no podían competir con productos basados en cinta magnética que el usuario puede reprogramar a voluntad.

El disco de **Matsushita**, de 20 cm de diámetro, puede almacenar hasta 15.000 imágenes en color o 10.000 documentos de tamaño A4, lo que equivale a 1000 veces la capacidad de un disco flexible de 8 pulgadas (1 megabyte), o 150 veces la capacidad de muchas unidades de disco duro. La información puede ser borrada y vuelta a escribir un millón de veces.

La capacidad de borrado se ha conseguido gracias a la adición de diversas sustancias, entre otras germanio e indio, al substrato de subóxido telúrico que forma la superficie básica de registro de los sistemas de disco óptico convencionales de "grabar y reproducir".

Al hacer pasar por la superficie "disparos" de un rayo láser grabador, es posible cambiar su estado de una fase cristalina a una fase amorfa (no cristalina) y viceversa.

Estas fases ofrecen alta y baja reflectividad luminosa respectivamente, y es esta información la que puede ser "leída" por un rayo láser de cierta potencia y longitud de onda, mientras el disco gira a una velocidad de 1.800 rpm.

El equipo **Matsushita** emplea dos rayos láser. El rayo grabador y reproductor tiene una longitud de onda de 0,83 μ m y una potencia incidente de 8 mw, mientras que el rayo de borrado tiene 0,78 μ m y 10 mw respectivamente. No obstante, utilizando un sistema de cabeza óptica única y las correspondientes lentes de enfoque, las funciones de borrado y grabación se consiguen hacer simultáneamente.

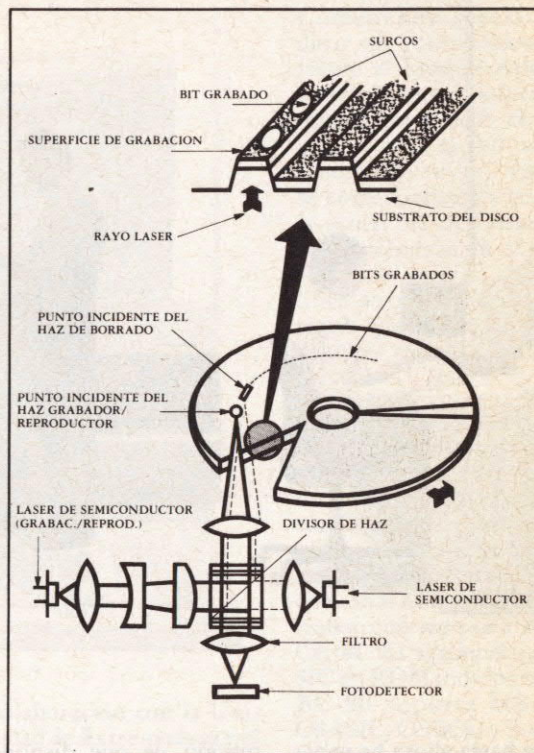
El disco borrable ha sido el fruto de un programa de investigación y desarrollo con un presupuesto de 6000 millones de dólares, llevado a cabo por **Matsushita**, y refleja la convicción de la compañía de que el disco óptico es

una tecnología clave para los futuros sistemas de automatización de oficinas. La compañía afirma que los primeros elementos de "muestra" estarán disponibles el año próximo.

Matsushita había comercializado ya, en 1981, una unidad de memoria óptica de disco no borrable, con especificaciones básicas similares a las del nuevo disco, y el año pasado lanzó su primer sistema de archivo óptico de documentos.

El disco óptico ofrece ventajas sobre la cinta magnética, más delicada. Su calidad de reproducción no se ve afectada por los pequeños defectos de la superficie o la

El diagrama muestra, arriba, la superficie del disco con los surcos en los que el rayo láser graba dígitos binarios (bits). Debajo, un esquema del sistema óptico, donde aparecen el disco, los dos láser, uno para registro/reproducción y el otro para borrado, y el conjunto divisor de haz.



CASIO FP 1000. PRIMER ORDENADOR PERSONAL CON ESTRUCTURA MULTIPROCESADOR.

Para Casio, innovar a buen precio no es ninguna innovación. Porque, en tecnología, Casio ha estado siempre por delante y al alcance de todos.

Siguiendo esta filosofía, Casio presenta ahora el FP 1000; el primer ordenador personal con estructura multiprocesador.

La estructura multiprocesador es la última

innovación de Casio: dos procesadores dentro de una misma estructura, que funcionan simultáneamente y hacen posible una alta velocidad de proceso. El FP 1000 cuenta con 64 KB de memoria, 2 x 320 K útiles en diskettes, impresora y pantalla gráfica de alta resolución lo que permite operar, opcionalmente, con 8 colores diferentes.

ACTUALIDAD

posible suciedad debida a su utilización diaria; además, su producción masiva resultaría más sencilla y económica.

Un disco óptico puede "archivar" los documentos a medida que se reciben, sin necesidad de proceso previo, por lo que promete ser el auténtico sistema de archivo "sin papeles". La posibilidad de borrado hará también posible la actualización de los ficheros y de los bancos de datos para consulta.

La llegada de los discos ópticos borrables y reutilizables, supone un nuevo paso hacia la convergencia de las tecnologías de video, audio y de datos, y, especialmente en el campo del video, ha llevado a especular sobre la futura continuidad de los fabricantes de sistemas no ópticos.

Un estudio publicado este mes por la *Nippon Administrative Management Association* sobre el progreso de la automatización de oficinas, revela el extraordinario

interés que la aplicación de los discos ópticos ha despertado entre las compañías japonesas. El 39,9 % de las 382 compañías consultadas manifiestan estar utilizando ya o pensando en introducir los sistemas de disco óptico.

■ Hace casi diez años que el tema surge a la luz, como rumor nunca confirmado. Honeywell, el poderoso grupo americano especializado en equipos de instrumentación y control, estaría a punto de abandonar sus actividades informáticas, cerrando las puertas de su filial **Honeywell Information Systems**. El bulo nunca se confirmó por una razón muy sencilla: Honeywell, para mantenerse al día en las actividades que le dejan más dólares de beneficios, necesita apoyarse en un *know how* que sólo la informática propia puede darle.

Lo que no impide que la filial informática de Honey-

well sufra crónicas dificultades financieras y que, en los últimos tiempos, se perfila en su horizonte un retroceso en su mercado, el de los *mainframes*.

■ Ciertamente, mientras IBM supo diversificarse a tiempo y mantener altas tasas de crecimiento en todas sus áreas de actuación, sus competidores de segunda línea, entre ellos **Honeywell Information Systems**, han tardado en encontrar el camino para salir del estancamiento. No es que le falten bazas: según estudios de IDC, hay en Estados Unidos un parque instalado de 18000 grandes ordenadores **Honeywell**, pero ese segmento de mercado ha sido invadido por los miniordenadores más potentes y tanto IBM como los japoneses están penetrando en la clientela de Honeywell.

Edson Spencer, el *chairman* de Honeywell, no pien-

sa en retirarse de la informática, aseguran los portavoces de la firma, que tiene su base en Minneapolis. Por el contrario, constituyó una *task force*, encabezada por **Janes J. Renier**, para salir del marasmo de su filial. El primer paso ha sido la presentación de un microordenador, el modelo 610, que debiera permitir a la firma satisfacer las necesidades de su clientela, atraída por la oferta de otros fabricantes.

La comercialización europea de ese modelo es improbable, habida cuenta de la supervivencia (aunque con menor compromiso financiero para los americanos) de los acuerdos con la empresa francesa **Cii-Honeywell Bull** (que, precisamente por ello, entre otras razones, vuelve a llamarse **Bull** a secas).

Ahora se dice que el repliegue de Honeywell tomaría una forma audaz, la asociación con la firma japonesa

CASIO AVANZA SIN QUE AVANCEN LOS PRECIOS.

Precio: 448.000 Ptas. (sin impresora)



Deseo recibir mayor información, sobre el Casio FP 1000

NOMBRE _____

EMPRESA _____

TEL. _____

CALLE _____

N.º _____

POBLACIÓN _____

PROVINCIA _____

Envíe este cupón a Gispert, P.R.P. Provenza, 206-208, Barcelona 36

CASIO

red de distribuidores oficiales.

Por gama. Por precios. Por servicio. Y porque hacemos bien nuestro trabajo.



GISPERT



Micro-ordenador ATARI

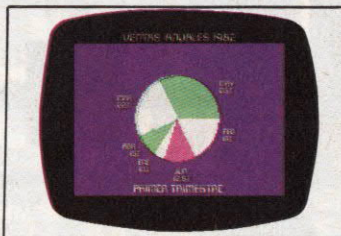
¡Capaz de Todo!

Ahora más que nunca hablamos su lenguaje

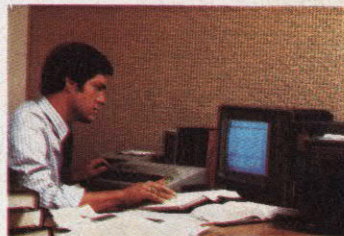
Basic, Pilot, Assembler, Microsoft Basic, MacroAssembler, Pascal.
Capaz de resolverle todo en casa o en la oficina.



- Compositor musical.
- Biorritmos.
- Iniciación a la Programación.
- Idiomas.
- Simulación Planta Nuclear.
- Entretenimientos (Ajedrez.
- Comando de Misiles, Comecocos Etc).



- Gráficos (3 dimensiones).
- Contabilidad.
- Estadísticas.



- Procesador de textos.
- Análisis de stock.
- Manejo de correspondencia y Editor de Listados.
- Etc.

SISTEMAS DE ORDENADORES

ATARI® 400/800™



CARACTERISTICAS ESPECIALES:
Alta resolución gráfica (320.192).
128 colores y 128 tonos. Módulos de memoria expansibles hasta 128 K RAM (conectables directamente con su televisor). Interfase para conexión de Modem, Plotter, conexión a otros ordenadores para bases de datos. Sonido (4 sintetizadores simultáneos e independientes).



Para mayor información dirigirse a: AUDELEC (División Ordenadores) Apartado 597. MALAGA

**Usted puede encontrar
los Micro Ordenadores
ATARI en los siguientes
establecimientos:**

ALBACETE

Sr. ALFONSO CEBRIAN GARCIA
C/. Isaac Peral, 9.
TEL: 22 63 69

TECOL, S. L.

C/. María Marín, 13
TEL: 23 61 27

ALCOY - ALICANTE

TESS

Avda. País Valenciá, 54
TEL: 54 52 86

BARCELONA

BERENGUERAS

C/. Diputación, 219
TEL: 323 36 51

REDISA GESTION, S. A.

Avda. Sarriá, 52/54.
TEL: 321 85 58

BILBAO

SOLO VIDEO

C/. Colón de Larreategui, 26
TEL: 424 17 97

GRANADA

MUEBLES

ELECTRODOMESTICOS SANCHEZ

Avda. José Antonio, 98
TEL: 26 77 11

MADRID

COMPUSTORE

C/. 12 de Octubre, 32
TEL: 409 36 74

C.O.S.E.S.A.

C/. Barquillo, 25
TEL: 231 29 18

D.I.D.I.S.A.

Paseo Pintor Rosales, 26

DINSA

C/. Gaztambide, 49
TEL: 244 34 00

ELECTRONICA LUGO, S. A.

C/. Barquillo, 40
TELS: 419 87 42/51

ELECTRONICA SANDOVAL

C/. Sandoval, 3
TEL: 445 18 33

MICROTEC

C/. Duque de Sesto, 30
TEL: 431 00 84

SONEX, S. A.

C/. Barquillo, 49

VISOCOLOR, S. A.

C/. Diego de León, 52

MALAGA

SUMINISTROS

ELECTRICOS MORENO

C/. Salitre, 13

PALMA DE MALLORCA

Sr. ANTONIO MARTORELL

C/. Via Roma, 5 B
TEL: 21 50 96

VALENCIA

EQUIPO DRAC

C/. Botánico Cavanilles, 30. B.
TEL: 361 39 50

**y además, en todos los
centros de El Corte
Inglés**

ACTUALIDAD

Nippon Electric (NEC) para producir y comercializar en común productos de alta gama. No está confirmado todavía, pero parece coherente con la política escogida por Reiner, que apunta a establecer una red de *joint ventures* con firmas rivales. Así, por ejemplo, Honeywell está asociada con Control Data (y parece que también lo estará con Sperry) en la producción de cierto tipo de periféricos.

Edson Spencer



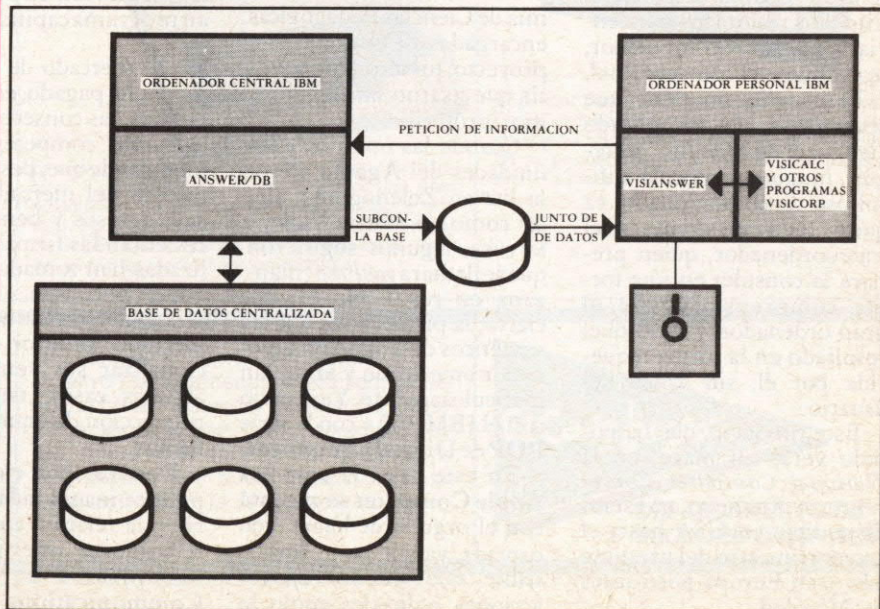
■ Visicorp e Informatics General Corporation acaban de lanzar conjuntamente una serie de productos *software* que proporcionan una conexión "inteligente" entre el ordenador personal IBM y los grandes equipos de esta marca.

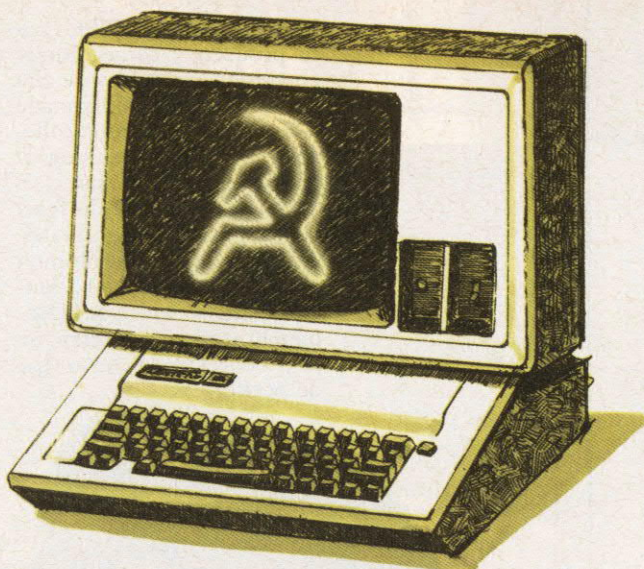
El paquete de *software* para el IBM/PC creado por Visicorp y llamado VisiAnswer, permite al usuario formular consultas y enviarlas al equipo central. Asimismo, acepta los resultados enviados por éste bajo una forma que permite visualizarlos, imprimirlos y, si es preciso, tratarlos.

En el otro extremo, un paquete llamado Answer/DB, permite que el equipo central pueda recibir las consultas procedentes del IBM/PC, extraer selectivamente la información de la base de datos, y prepararla para ser transmitida al ordenador personal. Esta información se almacena en los discos flexibles del IBM/PC en un formato inmediatamente utilizable por Visicalc y otros programas de Visicorp.

Este desarrollo reconoce

El sistema de enlace inteligente entre el Ordenador Personal y los grandes equipos IBM, presentado por Visicorp/Informatics.





la necesidad de los grandes equipos centralizados para el soporte, actualización y proceso de grandes volúmenes de datos, pero prevé que cada vez más los ejecutivos serán partidarios de tener sus propios teclados y pantallas sobre las mesas de sus despachos.

El nuevo *software* pretende ayudar a eliminar la confusión causada por la duplicación e inconsistencia de los datos manipulados por usuarios del IBM/PC. Asegurará también que todos los usuarios acceden a los mismos datos, actualizados, precisos y consistentes de la base de datos central.

Como explican las dos compañías, el despliegue de los ordenadores personales de IBM, aunque importante, no sería suficiente por sí mismo para satisfacer los requerimientos de calidad, prevención contra la duplicación, edad, comparabilidad y seguridad de los datos. Por otra parte, el enlace mediante el clásico terminal implica largos períodos de conexión y elevados costes de comunicación.

Con el nuevo sistema, la comunicación sólo se precisa brevemente al comienzo de la sesión, y después también brevemente cuando la consulta está compuesta y lista para cargarla en el equipo central. Cuando éste tiene listos los resultados para enviarlos al microcomputador, se establece el contacto final.

El usuario no tiene que aprenderse un complicado "lenguaje de consulta" ni seguir ningún protocolo dificultoso de comunicación, ya que de todo esto se encarga el microordenador, quien prepara la consulta en una forma comprensible para el gran ordenador, y recibirá el resultado en la forma requerida por él, sin ayuda del usuario.

Este producto, que ha podido verse en mayo, en la *National Computer Conference*, de Anaheim, no estará disponible en USA hasta el tercer trimestre del presente año y en Europa poco antes de Navidad.

■ A las heladas estepas rusas llegó una tentadora Manzanita que hizo caer en pecado a los ortodoxos burócratas soviéticos. Es cierto que no figura en los Mandamientos. Pero copiar es, seguramente, pecar. Lo decimos a propósito de la aparición, discreta pero notoria, en la Unión Soviética de un ordenador personal (que sepamos, el primero que se fabrica en el mundo comunista) calcado del Apple II. Tan parecido es el Agatha (que así se llama el producto soviético) al ordenador personal americano que los propios miembros de la Academia de Ciencias Pedagógicas, encargadas de supervisar el proyecto, tuvieron que admitir que usaron un Apple como "prototipo".

Cuando las iniciales 1.000 unidades del Agatha vieron la luz en Zelenograd, algo así como un Silicon Valley a la rusa, algunos sugirieron que se llamara *yabloko* (manzana en ruso). No es, por cierto, la primera vez que los soviéticos destripan un ordenador americano y lo copian meticulosamente. Ya ocurrió con el IBM 360 y con la serie PDP de Digital Equipment.

En este caso, la empresa Apple Computer se quedará con el orgullo de haber sido copiada, ya que no es imaginable que intente ejercer acciones judiciales como lo

hizo frente a las imitaciones de sus modelos fabricadas en Taiwan.

Plagio o no, el Agatha servirá para que los estudiantes soviéticos se entrenen en los grandes centros de enseñanza informática de Moscú y Novosibirsk. Según los planificadores, para 1999 habrá 5.000 aulas equipadas con ordenadores en las escuelas soviéticas.

Otras unidades de Agatha se instalarán en institutos y laboratorios de investigación. Es de temer que a los soviéticos les ocurra lo que a TRON, si el "prototipo" de Apple llevaba en su interior un programa capitalista.

■ El mercado de los video juegos ha pagado, en Estados Unidos, las consecuencias de la fuerte competencia. Temerosas de que, pese al crecimiento del mercado global, sus ingresos y beneficios se resientan las firmas especializadas han tomado medidas que les permitan, a la vez que afianzar posiciones en el mercado interior, internacionalizar sus ventas y, en algunos casos, también su producción de equipos y cartuchos.

Destaca, por ejemplo, la política inaugurada por Atari en la relación con sus distribuidores americanos. La subsidiaria de Warner Communications trata de

llegar a acuerdos exclusivos con ellos, ofreciéndoles descuentos de entre el 25 y el 40 por ciento sobre el precio de sus *cassettes* si se comprometen a no distribuir marcas competidoras. Precisamente por este motivo, Parker Brothers (un fabricante de juegos de sociedad reciclado a los *videogames*) inició demanda judicial contra Atari por prácticas monopólicas, y un tribunal le dio la razón. Atari, que presentó recurso, mantiene su política de negociar contratos de exclusividad con 40 por lo menos de sus distribuidores más importantes.

Por otra parte, tanto Atari como sus competidores de diversos tamaños tratan de colocarse en posición expectante de cara a mercados internacionales que hasta ahora descuidaban. Piensan, no sin razón, que los próximos años serán propicios para lograr tasas de crecimiento más altas fuera que dentro de Estados Unidos. La explicación es que, mientras en el mercado americano, el desarrollo que está alcanzando la televisión por cable, habrá de estrechar el mercado para los *videogames* (u obligar a los diseñadores y fabricantes a aguzar la imaginación), en otros países el mismo fenómeno está todavía lejano.

Stan Peters, director de ventas internacionales de Imagic, afirma que su compañía está transformándose en una organización mundial. Atari, a su vez, prevé que al cabo de los próximos cinco años el grueso del volumen total de sus ventas corresponderá al mercado internacional. Y lo cierto es que, en Europa, sólo Philips tiene una campaña de *marketing* agresiva, capaz de resistir la presión de los americanos.

■ La producción de *software* educacional está de moda. Ann Piestrup, doctora en Educación por la Universidad de Berkeley, ha creado una firma, a la que dio el sencillo nombre de Learning Corporation, para

Creemos en los ordenadores aplicados a la vida cotidiana.



OSBORNE

He aquí el máximo exponente de los ordenadores profesionales portátiles. Un compacto autónomo -puede funcionar con baterías o a la red- con monitor y minidisquetes incorporados lo que le da unas posibilidades inéditas hasta ahora. Su relación calidad/precio es muy buena.

PRECIO: 395.000 Pts.



M-20 OLIVETTI

El ordenador profesional que nace de la experiencia de la conocidísima firma italiana en el campo de la microinformática. El M-20 ofrece la tecnología más avanzada a gran escala, con un diseño innovador y compacto. Su microprocesador de 16 Bits auténticos permiten resolver cualquier problema de pequeña o mediana empresa con velocidad y soltura.

PRECIO: 599.700 Pts.



VIC-20 COMMODORE PRECIO: 44.950 pts.

Le presentamos un ordenador con el que podrá aprender microinformática muy fácilmente. El ordenador familiar, con color y sonido, que mayor número de programas educativos y de ocio tiene en el mercado. Desde aprender inglés, jugar al ajedrez, llevar la contabilidad casera...

(incluyendo Curso de Introducción Basic y un cartucho)



COMMODORE 64

Una perfecta herramienta para el ingeniero, el médico, el director financiero y, en general, para cualquier Departamento de una Empresa. El Commodore 64 está dotado de periféricos profesionales, gráficos de alta resolución, efectos tridimensionales, sintetizador de música...

PRECIO: 110.000 pts.



DRAGON 32

El ordenador familiar que estimula su imaginación. Con un Dragon 32 usted tendrá una gran posibilidad de realizar gráficos, sonido y edición. Podrá controlar su magnetofón desde el teclado. Su característica más sobresaliente es la velocidad en resolución de problemas.

PRECIO: 75.350 Pts.



SPECTRUM

La última novedad presentada por Investrónica. El ordenador familiar esperado. Una perfecta conjunción y máximas prestaciones. Merece la pena esperar para tener este ordenador compacto, en el que las posibilidades de color, sonido y gráficas son excelentes.

**PRECIO: 16K: 39.900 Pts.
48K: 52.000 Pts.**



SINCLAIR ZX81

El primer paso. El primer contacto con el mundo de los ordenadores familiares. Un buen ejemplo de diseño microelectrónico avanzado en función de su pequeño tamaño. Puede crecer modularmente a medida de sus necesidades.

PRECIO: 14.975 Pts.

Todo lo que usted necesite en el campo de los ordenadores personales -tanto los familiares, como los profesionales para su pequeña o mediana empresa- lo encontrará en El Corte Inglés.

Desde los últimos o más avanzados ordenadores hasta los complementos, libros, revistas y todo aquello que está relacio-

nado con el mundo de la informática.

Nuestro Departamento de Microinformática le ofrece además, instalación, transporte, servicio post-venta, compra a plazos, pago con tarjeta de compra de El Corte Inglés y nuestra garantía: Si no queda satisfecho, le devolvemos su dinero



DEPARTAMENTO DE MICROINFORMATICA

Nixdorf 8870

Manejable como un Mini, versátil como un Jeep, cómodo como un Mercedes y veloz como un Porsche.



Pocos sistemas informáticos poseen la versatilidad, la operatividad y el elevado rendimiento del Sistema de Primera Clase Nixdorf 8870. Ninguno le ofrece además, la simplicidad de manejo, la eficacia ante todo tipo de problemas informáticos, la justa adecuación al tamaño y crecimiento de su empresa y la óptima relación precio/rendimiento, del sistema Nixdorf 8870.

Porque el Sistema Nixdorf 8870 ha sido creado a partir de profundos análisis de las exigencias y necesidades del mercado. Obedece a una filosofía de trabajo y de tecnología muy avanzadas. Es un sistema probado de eficacia contrastada, hasta la total eliminación de factores sorpresa en cuanto a su durabilidad y rendimiento. Está respaldado por una organización experimentada y conocedora en profundidad de los pro-

blemas de los clientes a quienes va dirigido. Y su rendimiento le convierte en el más económico del mercado.

Pero más que las palabras, son los hechos los que colocan al Sistema Nixdorf 8870 a la cabeza de las decisiones informáticas inteligentes: más de mil clientes satisfechos en toda España pueden atestiguarlo, lo que para usted puede ser toda una garantía.

Clientes, que han visto, cómo el Sistema Nixdorf 8870 ha ido creciendo con ellos, gracias a su sistema modular compatible. Clientes que han comprobado cómo el Sistema Nixdorf 8870 sirve para muchos años, sin cambiar programas ni repetir esfuerzos.

Clientes que han descubierto lo fácil que puede resultar la informática y que se han convencido de que el Sistema Nixdorf 8870 se adapta perfectamente a todos los campos, desde la técnica más compleja, a la gestión empresarial integrada. Clientes, en suma, que han recuperado su inversión en un tiempo récord gracias al alto rendimiento alcanzado por los Equipos Nixdorf 8870.

Clientes y más clientes para los que la informática ha sido una inversión de primera. De Primera Clase.

Remita este cupón a Nixdorf Computer, Capitán Haya, 38
NO SE QUEDE ATRAS. INFORMESE
Madrid-20

Nombre _____
Empresa _____
Dirección _____
Población _____
Teléfono _____

OP-5 ✂

**NIXDORF
COMPUTER**

Primera Clase en Informática



Ann Pietrup

canalizar comercialmente productos de *software* desarrollados por ella. Luego de varias intentonas fallidas, **Ann Pietrup** ha desarrollado programas de enseñanza lectura/escritura, matemáticas y bases de circuitos, para niños de entre 3 y 13 años. El negocio de **Learning Corp** alcanza ya la cifra de 1 millón de dólares de ventas anuales. Se cree, según los analistas de mercado, que dentro de cinco años las ventas de *software* educativo alcanzarán, en total, los 3.000 millones de dólares.

Como se ve, la cifra abre el apetito. Y no son pocas las empresas que se preparan para comer una tajada de semejante mercado. Por ejemplo, **Tandy** ha llegado a un acuerdo con **Walt Disney Productions** y con **Childrens Television Workshop** (los creadores de Barrio Sésamo) para utilizar algunos de sus personajes en programas educativos para los ordenadores de la marca.

Los directivos de **Tandy** piensan que, al menos en su segmento del mercado, los programas educativos serán la clave de la venta de *hardware* en los próximos años. También **Atari** utilizará personajes de Disney y de Barrio Sésamo en su nuevo *software* infantil. Y **Apple** ha creado cuatro juegos del mismo carácter a partir de

los resultados de una investigación de mercado que llegó a la conclusión de que una buena parte de los compradores de sus ordenadores tienen hijos de entre 4 y 13 años de edad. También **Texas Instruments** comercializará más de 300 programas educativos en el futuro próximo.

■ Este año, seguramente, asistiremos al lanzamiento de otra novedad de **Apple Computer**. Tal vez no se trate, todavía del misterioso proyecto **Mackintosh**, del que todo el mundo habla pe-

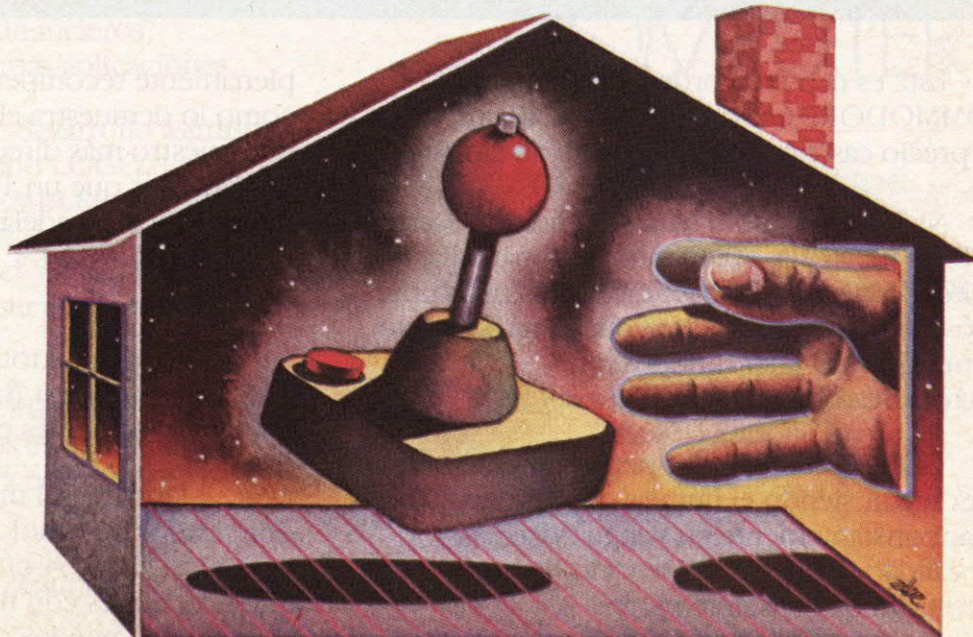
ro nadie sabe gran cosa. Al parecer, los estrategas de la firma de Cupertino han llegado a la conclusión de que deben presentar en sociedad un microordenador de bajo precio, al que por el momento denominan —en código interno de la compañía— **Elf**.

Las versiones más atinadas dicen que **Elf** no será otra cosa que una versión más barata del **Apple IIe**, presentado a principios de este año. Porque, aunque el propósito del rediseño del venerable **Apple II** —nuevo teclado y reducción drástica del núme-

ro de componentes— era el de hacer un modelo más barato que el caballito de batalla de la firma, el resultado, en la práctica, no fue el esperado sino más bien el contrario.

Elf sería, aparentemente, una versión más barata del **Apple IIe**. Resulta difícil imaginar por dónde se recortarán los costes, pero es lógico suponer que el nuevo modelo tendrá menos interfases y tal vez un teclado más simple (ambos componentes son normalmente los más caros de todo *hardware* si se excluye la unidad de disco).

■ El esperado *microdrive* de **Sinclair Research** para el ordenador **ZX Spectrum**, ya está, finalmente, en el mercado británico. Pero, tal vez, por demasiado esperado, está defraudando a los comentaristas de la prensa británica especializada. Según **Computer Answer**, por ejemplo, el nuevo dispositivo inventado por **Sinclair** está lejos de alcanzar ofrecer el prometido tiempo de acceso: 3,5 segundos. Los *test* efectuados por analistas de esa revista indican que el tiempo de acceso es, en la práctica, de 8 segundos. Lo que no puede menos que decepcionar a los fanáticos que han aguardado durante meses.



El mejor ordenador personal del Mundo



EL COMMODORE 64

Este es el nuevo ordenador personal COMMODORE 64. Un gigante de 40 cm, con un precio casi tan pequeño como su tamaño.

Nadie hasta ahora había logrado ofrecerle 64 K de memoria, 40 columnas en pantalla, 8 sprites y un sonido de auténtica maravilla por sólo 110.000,— ptas. Claro que tampoco todo el mundo es el líder mundial en microordenadores.

COMMODORE sabe perfectamente que para seguir siendo el número uno, tiene que estar constantemente en vanguardia. De calidad. De precios. De todo. Para ello investigamos constantemente.

Afortunadamente nuestra labor se ve

plenamente recompensada cuando vemos, como lo demuestra el cuadro comparativo, que nuestro más directo competidor cuesta nada menos que un 100% más caro. Y ello sin reunir todos los adelantos técnicos del COMMODORE 64.

1. Capacidad total de memoria RAM de 64 K. Interpretador BASIC extendido y sistema operativo residentes en ROM.

2. Dotado del más potente chip sintetizador de sonido diseñado hasta hoy, el COMMODORE 64 ofrece 3 veces totalmente independientes con una gama de 9 octavas. El programa puede controlar la envolvente, la afinación y la forma de onda de cada voz,

convirtiendo al COMMODORE 64 en el mejor simulador de instrumentos.

3. Conectable directamente a toda una gama de periféricos, incluyendo unidad de discos, impresora de matriz de puntos o de margarita, plotter, comunicaciones locales y remotas..., y mucho más.

4. Pantalla de alta resolución en color con 320 x 200 puntos directamente direccionables. Capacidad en modo carácter de 25 líneas por 40 columnas.

5. El chip de video, único en su género, permite el uso de 8 «Sprites» (figuras móviles en alta resolución y color). Los «Sprites» pueden moverse independientemente por programa de «pixel» en «pixel».

6. A cada «Sprite» se le asigna por programa un nivel de prioridad en caso de cruce con otro, consiguiendo efectos tridimensionales, existiendo también detección automática de colisiones.

7. Teclado profesional con mayúsculas y minúsculas, más 62 caracteres gráficos, todos ellos disponibles en el teclado y visualizables en 16 colores, en forma normal o bien en video invertido.

8. Encontrará a su disposición una completa gama de programas profesionales, incluyendo proceso de textos, sistemas de información, modelos financieros, contabilidad y muchas más aplicaciones.

9. Están en fase de desarrollo asimismo otros lenguajes tales como LOGO, UCSD PASCAL, COMAL, ASSEMBLER, etc. Todos los programas existentes de la gama COMMODORE, desde el VIC-20 hasta los modelos CBM pueden ser adaptados fácilmente.

10. Posibilidad de inserción de cartuchos con programas grabados en ROM, tanto profesionales como para educación y ocio.

11. Opción de un segundo procesador Z-80 para trabajar con sistema operativo CP/M (R).

EL COMMODORE 64 Y SU MAS DIRECTO COMPETIDOR

OPCION DE BASE	COMMODORE 64	Más directo competidor
Precio	110.000,— ptas.	* El doble
Memoria usuario	64 K	48 K
Teclado profesional	SI	SI
Teclado con caracteres gráficos	SI	NO
Minúsculas	SI	NO
Teclas de función	SI	NO
Máxima capacidad disco	170 K a 1 M	143 K
AUDIO		
Generador de sonido	SI	SI
Sintetizador de música	SI	NO
Salida HI-FI	SI	NO
VIDEO		
Salida monitor	SI	SI
Salida para TV	SI	EXTRA
PERIFERICOS		
Cassette	SI	SI
Periféricos inteligentes	SI	SI
Bus serie	SI	NO
SOFTWARE		
Opción CP/M (R)	SI	SI
Ranura cartucho externo	SI	NO

 **commodore**
COMPUTER

PARA MAS INFORMACION
DEL COMMODORE 64,
LLAMAR O ESCRIBIR A:
MICROELECTRONICA Y CONTROL
c/ Taquígrafo Serra, 7, 5º. Barcelona-29
Tel. (93) 250 51 03
c/ Princesa, 47, 3º, G. Madrid-8
Tel. (91) 248 95 70

Nombre.....
Dirección.....
Tel.....
Población.....



COMMODORE 64

Cuando fue presentado el original "escarabajo" de **Volkswagen**, ofrecía todas las características que podía necesitar un conductor medio y era, al mismo tiempo, un buen vehículo común para el transporte familiar. Era, además, barato. Aquel coche de éxito no repetido cumplió con las exigencias de su apelativo: "el automóvil del pueblo". Ahora, en 1983, acaba de llegar al público el **Volkswagen** de los ordenadores personales. El **Commodore 64** ofrece más bytes por unidad de dinero que

cualquiera de sus competidores, y se nos ocurre que bien puede merecer la comparación con el **Volkswagen**.

El nuevo modelo de **Commodore** ofrece una larga lista de características estándar y de opciones accesibles. Viene con una memoria programable de 64 Kbytes (de allí su nombre), gráficos avanzados con exposición en 16 colores, animación con "duendes" (es decir caracteres gráficos programables, llamados *sprites* en inglés y BOM —bloques y objetos móviles— en la versión castella-

na del manual), un sintetizador musical de tres voces, un microprocesador 6510 (compatible con el 6502) y todo un equipo de accesorios baratos y fáciles de conseguir.

Como ocurre con casi todos los ordenadores nuevos, la fuerza del **Commodore** reside más en su *hardware* que en el *software* que se puede conseguir. El fabricante y su representante en España han anunciado una amplia línea de *software* para su uso en aplicaciones comerciales, educativas y domésticas, con bajos precios de venta por programa, que irá apareciendo en el mercado en los próximos meses. **Commodore** ha tenido por otra parte, la previsión de ofrecer cartuchos opcionales para el sistema



operativo **CP/M**, además del lenguaje Logo, que facilitará la introducción del 64 en el mercado educacional.

Commodore Business Machines (curiosa reminiscencia del nombre de **IBM**) es una de las pocas compañías en el mundo de la informática que fabrica sus propios circuitos integrados. Dado que cuenta con un abastecimiento seguro e inmediato de *chips*, puede crear máquinas avanzadas antes que otros y a precios inferiores. De hecho, fue **MOS Technology**, una subsidiaria de **Commodore**, la firma que creó el microprocesador 6502.

Desde la época de sus humildes comienzos en 1956 como tienda de repa-

ración de máquinas de escribir, **Commodore** ha crecido hasta convertirse en lo que hoy es: una corporación multimillonaria cuyas acciones son muy apreciadas en el mercado bursátil.

Sin dejar de hacernos sentir culpables por comprar máquinas de juegos en lugar de verdaderos ordenadores, **Commodore** ha acumulado suficientes ventas de su **VIC-20** como para convertirse en uno de los pocos fabricantes que forman parte del "club del millón vendido". Y es muy posible que el nuevo modelo, tan agresivamente lanzado al mercado, obtenga un éxito similar al de su predecesor.

El **Commodore 64** es una unidad

liviana, compacta, del tamaño de una pequeña máquina de escribir portátil. Se parece notablemente al **VIC-20**, excepción hecha de la etiqueta y del color de su caja. Junto con él se entrega una fuente de alimentación, una conexión para el receptor de TV y una guía para el usuario.

Este ordenador de una sola pieza alberga un microprocesador 6510, 20 Kbytes de memoria ROM conteniendo el intérprete de lenguaje BASIC, el sistema operativo **Commodore** y su equipo de caracteres estándar, 64 Kbytes de RAM y *chips* especiales que aportan al **Commodore 64** su potente capacidad musical y gráfica. En la parte posterior,

La clave está en el software

La estrategia de **Commodore** al presentar su modelo **64** es despegarse de los juegos para ir ir hacia aplicaciones clásicas de ordenador personal, incluyendo programas destinados a la pequeña gestión. La idea central es arropar el *hard-ware* con programas estándar que respondan a las necesidades de la mesa de trabajo o del profesional libre que aspira a mecanizarse pero no hasta el punto de invertir 800.000 pesetas en un ordenador personal de los que ya existen en el mercado. Con este enfoque, es obvio que la clave del éxito que pueda alcanzar el **Commodore 64** depende en mucho del *software* que lo acompañe. Se necesitan programas concebidos para que el usuario promedio pueda aprovechar las potentes características de este nuevo ordenador. En el mercado americano —al que durante un tiempo descuidó en beneficio de su implantación en otras partes del mundo—

Commodore afronta ahora el problema de que muchos de los programas que ofrece fueron escritos originalmente para el **PET** y, en consecuencia, no sacan partido del **Commodore 64**.

Hay, eso sí, una serie de programas, todos ellos con el prefijo **Easy**, adaptados al nuevo modelo. Algunos de esos programas, de origen americano, están siendo traducidos por equipos de **Microelectrónica y Control**, que representa a la marca en el mercado español. Pero, por lo menos en el caso de **Easycalc** —una versión del "tablero electrónico" **Visicalc**— todo indica que dejará su lugar a un programa de origen sueco, llamado **CalcResult**, más

potente y eficaz. Esto nos lleva a indicar que, probablemente los programas escritos en Europa para los modelos **Commodore** tengan mejor acogida en España que los procedentes de Estados Unidos. Habrá, según señalan los portavoces de la firma que comercializa **Commodore** en España, una auténtica escalada de lanzamiento de *software* para el modelo **64**. El **CalcResult** y el **Easy Script** (un buen procesador de textos) ya están disponibles, junto con los cartuchos de lenguajes **Forth** y **Logo** y un monitor de código de máquina. Antes del verano estará disponible el cartucho para el sistema operativo **CP/M**, además de varios paquetes traducidos y otros que se están desarrollando ahora mismo en España. Entre ellos dos programas de contabilidad y una base de datos. Entre las novedades de *software* que se preparan para el resto de 1983 figuran varias decenas de programas educativos —extraídos de un lote de más de 600 presentados por **Commodore** y, probablemente, el **Wordcraft**.

Aparentemente, los diseñadores del **Commodore 64** piensan que la mayoría de los compradores no harán el gasto para programar el ordenador en **BASIC** sino para ejecutar programas disponibles para aplicaciones en los negocios, en la educación y en el dominio doméstico. Tal vez sea ésta la explicación de lo que nos parece una relativa flojedad del **BASIC** incorporado al **64**, que no es sino una versión adaptada del **Microsoft** clásico, del que difiere, sin embargo, por sus gráficos, por el manejo de

los ficheros y por los comandos *input/output*. La memoria imprescindible para el **BASIC** ha sido minimizada, a expensas de los comandos convenientes para la gestión de ficheros en disco, de gráficos y de sonido. Sin embargo, se dispone de un cartucho de expansión **BASIC** de 8 Kbytes, para mejorar los comandos de gráficos, color y música.

Encender el ordenador implica una pausa de dos segundos, durante la cual la pantalla queda en blanco mientras el **Commodore 64** chequea un cartucho, se inicializa y purea su propia memoria. Luego, los signos **BASIC**, y aparece en pantalla el mensaje que indica que la memoria está libre y dispuesta para programación. Esto es más de diez veces el espacio de trabajo del programa que podía obtenerse con el **VIC-20**. Por otra parte, el **Commodore 64** podría tolerar fácilmente un **BASIC** más potente, pero el fabricante sigue pensando en términos de programas pequeños a pesar de que nos acercamos al final de la época de las memorias limitadas.

Las líneas de programa, por ejemplo, no pueden tener una longitud superior a los 80 caracteres, y sólo las dos primeras letras de las variables de programa son usadas. Dicho esto, también es de rigor señalar que resulta un placer trabajar con el editor de pantalla en **BASIC**, pues permite cambiar con toda facilidad cualquier línea. No hay más que usar el cursor sobre el error y escribir la corrección. También se puede agregar o borrar caracteres con la tecla

a la derecha, están los conectores de periféricos y cartuchos de expansión.

Otros ordenadores de bajo costo suelen proveer memoria programable de 16 Kbytes (ampliable a 48 Kbytes). Su memoria programable de 64 Kbytes permite al último modelo de **Commodore** ofrecer más sofisticación de funcionamiento y mejor gestión de programas. La memoria adicional, por otra parte, realza los gráficos. Y vale la pena subrayar que 8 de los 35 circuitos integrados que conforman el **Commodore 64** son *chips* de RAM, probablemente la parte más cara del *hardware* de este ordenador.

El microprocesador **6510**, de 8 bits, es el primer consumidor de toda esa me-

moria, que emplea para almacenar programas y datos. Internamente, el **6510** es idéntico al **6502** y usa las mismas instrucciones, pero tiene una puerta de I/O de 8 bits adicionales para gestión de memoria. Esto permite al microprocesador acceder a una memoria superior a los 84 Kbytes conectando con diferentes áreas y tipos de memoria en los **64** Kbytes a los que el **6510** puede acceder en cualquier momento.

Ya es tiempo de que presentemos a los lectores el **VIC** y el **SID**. El **Video Interface Chip 6567 (VIC-II)** es una versión mejorada del circuito integrado que dio al **VIC-20** no sólo su nombre sino también su capacidad gráfica y su bajo costo. El **VIC-II** es una versión de

40 columnas del antiguo *chip* **VIC** y maneja todo el proceso de *video-display*, dejando al microprocesador central libre para efectuar otras tareas. El *display* de 40 columnas por 25 líneas del **64** de texto, y la alta resolución de 300 × 200 puntos así como los "duendes" gráficos son todos producidos por este *chip*.

El **Sound Interface Device 6581 (SID)** es un sintetizador musical electrónico de tres voces y generador de efectos sonoros. Provee un excelente control sobre el tono y el volumen. La capacidad de música programable que provee este *chip* es suficiente razón para adquirir el **Commodore 64**.

Estos tres *chips* —**6510**, **6567** y

INSERT/DELETE igual que si se estuviera usando un procesador de textos. Al pulsar RETURN mientras el cursor está en cualquier lugar de la línea que será modificada, la nueva línea reemplaza a la anterior. Y hasta el número de línea de un programa BASIC puede ser modificado, haciendo posible la inmediata duplicación de líneas similares.

Muchas de las palabras clave en BASIC son accesibles en forma abreviada pulsando la primera o las dos primeras letras y la tercera en SHIFT. Si se usa con frecuencia el BASIC, una vez memorizado los códigos, se ahorra mucho tiempo. Es posible almacenar cadenas de caracteres entrecomillas, comandos de símbolos para limpiar la pantalla, cambiar el color de los caracteres y

mover el cursor. Todo ello es de agradecer, pero no siempre queda claro el sentido de los símbolos invertidos por estos comandos. Es, por cierto, fácil recordar que el inverso de la forma corazón limpia la pantalla, pero las demás combinaciones de caracteres invertidos y su significado son, en definitiva, una fuente de confusión. **Commodore** tiende a usar mucho en sus programas estos códigos invertidos y, por consiguiente, es bastante difícil operar los programas si no se conoce el comando que crea en el listado los caracteres invertidos. Y, lo que no es menos importante, obliga a usar una impresora **Commodore** para imprimir los listados.

Los archivos y los datos son manejados según las mismas reglas que las usadas en los BASIC anteriores para el PET y el VIC-20. Luego de abierto, las instrucciones son similares, ya sea que se use un grabador de cinta, un disco o una impresora. Fácilmente se puede redirigir el *input* o el *output* cambiando el número de dispositivo o usando un canal de comando. Puesto que tanto el BASIC como el *drive* inteligente están involucrados en el manejo de archivos, hay que poner cuidado en que los cierren simultáneamente. De otro modo, puede haber problemas por haber dejado huecos en el disco en los que caerán los datos. Por ahora, no parece que **Commodore** tenga planes para ofrecer su BASIC 4.0, más potente y extenso, que gasta más memoria pero presenta mejores comandos para un sistema en disco. Otras



6581— son la base para la compatibilidad entre el 64 y otros ordenadores de **Commodore**. Muchos de los cartuchos de juego o programas del 64 pueden usarse sin modificaciones en otras máquinas de la marca. El cartucho se conecta por medio de una clavija en la parte posterior del ordenador, cerca del cable de alimentación. La expansión BASIC para CP/M y los cartuchos Logo se adaptan también a las ranuras. Por su parte, los cartuchos de juego del VIC-20, que son apenas un poco más grandes, no funcionan en el 64, puesto que ese modelo de ordenador no lleva ninguno de los *chips* arriba mencionados.

Dos puertas de control de juego a un lado del ordenador aceptan la conexión

de *joysticks*, *paddles* y lápices ópticos. Detrás está la conexión para TV, la salida para sistema estéreo y monitor, una puerta serial para hasta cinco dispositivos (cuatro *drives* de disco y una impresora), la salida para grabador de *cassettes* y otra para accesorios tales como un *modem* VIC o interfaces estándar. Un detalle interesante: es imposible equivocarse porque todas las conexiones son diferentes unas de otras.

El **Commodore 64** tiene un excelente teclado para ser un ordenador barato. A las mecanógrafas habituadas a trabajar al tacto les puede llevar algún tiempo habituarse a la ubicación de las teclas de punto, coma, signos varios y elevación para los caracteres de la fila superior.

Sobre todo, se echa de menos una tecla de tabulación para aplicaciones de proceso de textos.

Entre las 66 teclas hay 4 especiales de programación, que pueden usarse con la de elevación para lograr un total de ocho teclas de función, característica que ningún otro ordenador de bajo costo ofrece.

Dos teclas de control de cursor se usan junto con la tecla de elevación para mover el cursor en las cuatro direcciones. Mientras tanto, probablemente los usuarios de programas de proceso de texto o de "tablero electrónico" probablemente preferirían cuatro teclas separadas. El movimiento de cursor, el espaciador y la tecla de insert/delete repiten automáticamente si se las mantiene ba-

La clave está en el software

compañías están estudiando esa posibilidad. La opción **CP/M** provee un mejor sistema, pero hay que desconectarla para poder aprovechar el *software* de bajo precio diseñado especialmente para el **Commodore 64**.

Se sabe que Logo es el lenguaje preferido para aplicaciones en educación. Del mismo modo que gracias a **Visicalc** se han vendido mejor los **Apple II** para el mundo de los negocios, la disponibilidad de Logo puede facilitar las ventas de **Commodore 64** a las escuelas. Porque muchas de éstas han decidido que sólo comprarán *hardware* apropiado para rodar Logo.

El usar los gráficos de tortuga del Logo para ilustrar conceptos del programa, simplifica el aprendizaje de un lenguaje de computación. Logo hace más visible lo que el programa está ejecutando, a medida que la tortuga dibuja el patrón en la pantalla.

Al enseñarle al ordenador nuevas órdenes Logo, se les da a los estudiantes el sentido de control sobre el ordenador. Este procedimiento también puede ser empleado para fraccionar un problema en partes, para realizar una aproximación modular a la programación. Los procedimientos pueden exigir otros procedimientos o pueden incluso pedir copias de sí mismos en la potente técnica llamada *recursion*.

Commodore dice haber vendido más *software* educativo en todo el mundo que cualquier otro fabricante. La compañía ha tenido mucho éxito —en Estados Unidos— en imponer sus ordenadores con el recurso de entregar tres equipos por el precio de dos a las escuelas. Ahora bien,

esta amplia base de instalaciones ha generado miles de programas de carácter educativo. Estos programas, accesibles para escuelas y distribuidores, cubren desde la administración escolar hasta la enseñanza de idiomas, geografía, historia, matemáticas y ciencias. Además de juegos didácticos, naturalmente.

Lo cierto es que el uso más habitual de los ordenadores domésticos son los juegos. **Commodore** ha obtenido licencias para juegos tales como **Gorf**, **Omega**, **Clowns** y **Sea Wolf**, para hacer de ellos versiones especiales para el modelo **64**. Los cartuchos y *cassettes* ya están a disposición de los usuarios. También hay programas prácticos para uso doméstico, como **Word Machine** para notas y correspondencia, y **The Name Machine**, que es una agenda de nombres y direcciones. El emulador **PET** permite al **Commodore 64** utilizar muchos de los programas escritos para el **PET**. Una opción anunciada algo prematuramente por una empresa americana es un emulador que podría permitir al **Commodore 64** cargar y ejecutar programas escritos para el **Apple II**. Si la cosa funciona, podría constituirse en una gran ventaja, habida cuenta de que el **Apple II** sigue teniendo la colección más caudalosa de programas. Pero desarrollar ese emulador quizás sea imposible. Los emuladores son siempre algo lentos, pues deben traducir instrucciones y simular el funcionamiento del ordenador original. Para un simple programa **BASIC** esto no ofrece mayores dificultades, pero cuando se intenta copiar las complejidades de gráficos de alta resolución en un juego de movimientos rápidos, lo más posible es que se acabe realizando un juego lento que, por esto mismo, resulte poco divertido.

jo presión, pero a diferencia de la mayor parte de los ordenadores, las demás teclas no repiten.

Una memoria de teclado (*buffer*) ayuda al **Commodore 64** para que pueda satisfacer a los mecanógrafos rápidos y permite una nueva orden antes que el ordenador haya acatado la anterior. Dado que el *buffer* no almacena más de 10 teclas a la vez, un mecanógrafo muy veloz puede llegar a perder un carácter alguna que otra vez.

En varios sentidos, el **Commodore 64** podría ser visto como un reacondicionamiento muy mejorado del **PET** de la misma marca. Esto se hace evidente cuando se comienza a emplear el teclado y se comprueba un gran ordenamiento

de caracteres gráficos que se logran con los alfanuméricos comunes en mayúsculas y minúsculas y los signos de puntuación. Los 66 caracteres gráficos marcados en el borde del frente de las teclas son los mismos que los usados en el **PET** original. Mientras esto permite al **Commodore 64** trabajar con casi todo el *software* del **PET** (con la ayuda de un cartucho emulador opcional), también le crea algunas complicaciones adicionales.

Al encender el **64**, comienza a funcionar en modo mayúsculas/gráficos. Cualquier signo alfabético que se pulse aparece en la pantalla como mayúscula. Si se pulsa un signo en **SHIFT**, se verá el que corresponde a la derecha sobre la

tecla. Usando la tecla **Commodore** (en el rincón inferior a la izquierda del teclado) en vez de **SHIFT**, verá el signo gráfico a la izquierda del borde frontal de la tecla.

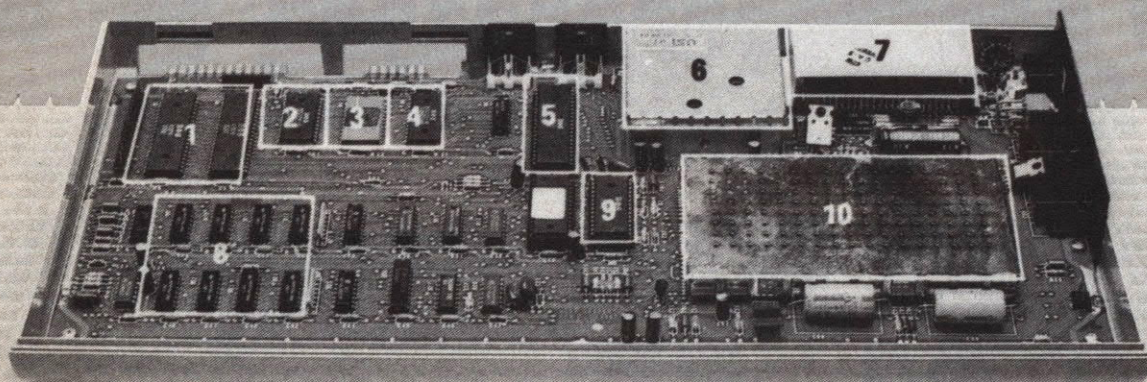
¿Dónde están las minúsculas? Para pasar el ordenador a letras normales minúsculas y mayúsculas, debe pulsarse la tecla **Commodore** y la **SHIFT** simultáneamente. Pero este modo no le permite exponer los caracteres gráficos del frente derecho de las teclas. Y, lo que es peor aún, en este modo las mayúsculas no son reconocidas por el intérprete de **BASIC**, provocando así la aparición del aborrecido mensaje "Error de sintaxis". Por esto, si usted está programando en **BASIC**, mejor será que se limite al modo mayúsculas/gráficos, que también da acceso a los caracteres gráficos. Si lo que le interesa es hacer proceso de textos, recurra al convencional modo mayúsculas/minúsculas.

Para volver al modo mayúsculas/gráficos, simplemente pulse la tecla **Commodore** y la **SHIFT** simultáneamente otra vez. Pero, tenga cuidado. Cualquier minúscula que ya estuviera en pantalla se convertirá en mayúscula y todas las mayúsculas se tornarán caracteres gráficos. Es superfluo especificar que estos dos conjuntos de caracteres pueden llegar a ser una buena causa de irritación al comienzo, por ser tan confusos, aunque la mayoría de la gente parece habituarse a su manejo en poco tiempo.

El **64** no usa el **ASCII**, los 128 caracteres del *American National Standard Code for Information*. Genera todos los caracteres del **ASCII** excepto el retroceso, subrayado, paréntesis, barra vertical y acento (que no es muy usado) pero no en el orden común. Debido a esta única disposición, habrá que traducir si se realizan comunicaciones con otros ordenadores no fabricados por **Commodore**. Esto puede hacerse con un *software* de comunicaciones como el **64-Term**, que acompaña al *modem* **VIC**.

Pueden imprimirse 157 caracteres distintos como parte de los dos conjuntos de 256 caracteres. Muchos de los caracteres de uno de los conjuntos están repetidos en el otro o son meramente caracteres de control que no imprimen. Los caracteres pueden ser exhibidos en imagen invertida y en cualquiera de los 16 colores. Además, los caracteres son programables para poder definir la forma de la rejilla de 8 x 8 puntos que configuran cada uno de los caracteres. Esto permite una enorme flexibilidad para los juegos y las aplicaciones comerciales.

En cuanto a la pantalla, el **Commodore 64** exhibe 25 líneas de 40 caracteres en un receptor de TV o en un monitor. El límite de 40 columnas se debe a las pantallas de televisión, que son el medio más barato de lograr una exhibición en color. Muchos ordenado-



1. Dos chips 6526 controlan las conexiones entre el procesador central, el *joystick*, el reloj y los accesorios.
2. Memoria ROM que contiene el BASIC.
3. *Chip* de 8 Kbytes que alberga el sistema operativo Commodore.
4. *Chip* de 4 Kbytes del conjunto de caracteres.
5. Unidad central de proceso, microprocesador 6510.
6. Convertidor de radiofrecuencia que traduce las señales de video para la conexión a un receptor estándar de TV

- color a través de una puerta de salida de radiofrecuencia.
7. Port para cartucho.

8. Ocho *chips* de memoria, cada uno de ellos de 8 Kbytes de RAM.

9. *Sound Interface Device* (SID) 6581, que provee al Commodore 64 su capacidad de generación musical.

10. *Video Interface Chip* (VIC) 6567, clave de la capacidad gráfica del ordenador.

res ofrecen actualmente 80 caracteres por línea en monitores de video, pero no parece estar en los planes de **Commodore** ofrecer un adaptador para igualar esa oferta. Es posible, sin embargo, que otras compañías —dependiendo de la introducción que logre el **Commodore 64** en el mercado— lancen adaptadores que permitan visionar un documento en toda su amplitud, lo que resulta un factor importante para aplicaciones de gestión y para proceso de textos.

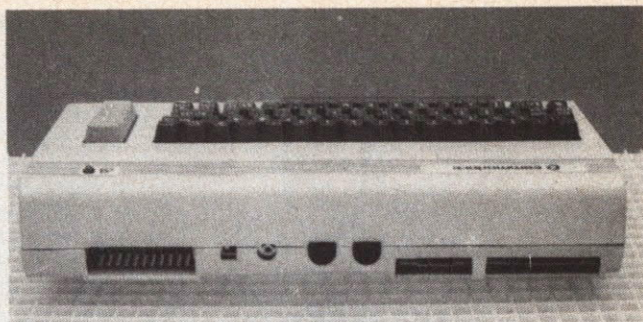
La pantalla comienza con letras en azul claro sobre fondo azul oscuro. Esto, aunque de apariencia atractiva, no da un buen contraste. En consecuencia, cambiar el fondo azul a blanco facilita la lectura.

Cambiar el color del carácter siguiente es muy fácil, por medio del control CTRL o de una tecla Commodore como una de SHIFT con los dígitos del 1 al 8. Los colores son hermosos para trabajar, y pensamos que más ordenadores deberían adquirir esta riqueza de expresión. El espectro incluye: negro, rojo, siena, púrpura, verde, azul oscuro, amarillo, naranja, marrón, rosa, gris claro, gris medio, verde claro, celeste y gris oscuro. Dependiendo del ajuste del color del receptor de TV, el púrpura puede resultar rosado y el marrón limitarse a una sombra naranja.

El fondo puede obtenerse en cualquiera de los dieciséis colores, lo que facilita una gran cantidad de combina-

ciones de color de fondo/letra. En esta situación, la tecla **RESTORE** es una ayuda para el usuario. Si se pulsa la tecla **RESTORE** simultáneamente con la **RUN/STOP**, el ordenador se acomoda a su situación original y la pantalla se limpiará sin perder por ello el programa en memoria. Esta es una característica muy útil si se está usando alguna de las características gráficas o de sonido y se quiere despejar la pantalla para volver a la característica normal del comienzo. También se puede dar cualquiera de los 16 colores al borde de la zona del texto de 40 columnas por 25 líneas.

El **Commodore 64** tiene una impresionante colección de capacidades gráficas. Los tres modos generales incluyen



caracteres gráficos de 40 columnas por 25 líneas, gráficos de alta resolución de 300×200 puntos y los bloques y objetos móviles. Estos son grandes caracteres programables que pueden moverse por la pantalla con gran facilidad. También pueden aparecer simultáneamente combinaciones de estos modos.

La base para gran parte de esta capacidad gráfica es la memoria del **Commodore 64**. La ubicación de cada carácter en la pantalla (40×25) tiene una correspondencia en la memoria del ordenador, creando así lo que llamamos *memory-mapped display* (pantalla con configuración de memoria). Cada vez que uno de los 256 posibles caracteres es ubicado en una de las 1.000 localizaciones de la memoria, el carácter aparece en el lugar correspondiente de la pantalla. Una zona de memoria-color mantiene la huella del color de cada uno de los caracteres.

Al poner los caracteres gráficos apropiados en la memoria del video y el color de cada uno de los caracteres gráficos en la memoria/color, se puede llegar a obtener resultados artísticos con el **Commodore 64**. Con el inconveniente de que puede ser muy difícil de lograr cuando se está usando BASIC.

La versión BASIC 2.0 de **Commodore** sólo tiene un comando para hacer todo esto: **POKE**. Tomará un valor (un byte) y lo ubicará en una localización especial de memoria. La instrucción

POKE 1024,1 pondrá la letra A (carácter 1) en el rincón superior a la izquierda de la pantalla (localización de memoria 1024).

Esta retorcida aproximación numérica no es la más acertada para enseñar a pintar con el **Commodore 64**. El cartucho opcional de expansión BASIC ofrece un paquete de soporte musical y de video con simples comandos color como **COLOR**, **PLOT**, **CIRCLE** y **LINE**, que facilita enormemente la manera de usar los gráficos.

¿Quiere usted enseñar ruso o griego a su ordenador? El conjunto de caracteres programables le permite el rediseño de cada uno de los caracteres de 8×8 puntos pulsables desde el teclado. También son posibles los caracteres de varios colores.

Los caracteres programables se usan también para crear gráficos de alta resolución con ciertas limitaciones de color. Cada cuadro vale 800 palabras (bytes). Se necesitan 8 bytes para definir los caracteres de 8×8 puntos de las 1000 posiciones de la pantalla. Esto permite una imagen de 320×200 puntos en la cual cada rejilla de 8×8 puntos está limitada a dos colores. También existe la posibilidad de una imagen de resolución media multicolor. Cada uno de los caracteres 8×8 del modo de resolución media de 160×200 puntos puede tener cada uno de los cuatro colores. Es posible, por otra parte, el lento y suave

desplazamiento de toda la pantalla en horizontal o en vertical, de a un punto a la vez.

Supérfluo es decir que al especificar todos estos valores en la versión BASIC 2.0 de **Commodore** puede ser toda una tarea. Pero con un poco de paciencia se pueden alcanzar resultados asombrosos.

Los bloques y objetos móviles (*sprites*) de composición gráfica no necesitan muchas directivas. Un "duende" es un carácter grande (24×21 puntos) y programable, cuya forma y color es posible definir. Es muy fácil moverlos por la pantalla y pueden desplazarse independientemente de otros *sprites* y de otros gráficos. También pueden pasar por delante o por detrás de uno u otro o por partes del fondo para darle a la animación gráfica calidad tridimensional. Aunque son una de las realizaciones más potentes de los gráficos creativos, son pocos los ordenadores que disponen de ellos. El **Texas Instruments 99/4A** los usa para Logo y otras aplicaciones. El **Atari** de las series 400 y 800 (que los llama misiles) los emplea en sus juegos interplanetarios de excelente calidad.

Por su parte, el **Commodore 64** tiene 8 *sprites* que pueden alternar con otros modos gráficos. Simplemente, hay que darle a cada uno de ellos forma y color, y una localización x-y, para que se pongan a trabajar.

Si los *sprites* de 24×21 puntos no le

AUMENTE LAS POSIBILIDADES DEL **sinclair** ZX81



PERIFERICOS YA DISPONIBLES DE FABRICACION NACIONAL PARA EL ORDENADOR SINCLAIR ZX81

- **TECLADO PROFESIONAL.** 52 TECLAS EN DOS SECCIONES: NUMERICAS Y ALFANUMERICAS GRABADAS CON TERMOIMPRESION. INTERRUPTOR ON/OFF CON PILOTO DE AVISO. SALIDAS PARA MIC, EAR, TV, MONITOR, ETC. INCORPORACION DE MEMORIA INTERNA O EXTERNAMENTE. SENCILLO ACOPLAMIENTO INTERNO DEL SINCLAIR SIN SOLDADURAS. **P.V.P. 18.900**
- **MEMORIA DE 16 K.** AMPLIACION DE MEMORIA PARA EL ZX81 DE ALTA TECNOLOGIA. NO NECESITA MODIFICACION DE RAMTOP. PILOTO/AVISO DE FUNCIONAMIENTO. **P.V.P. 12.150**
- **MEMORIA DE 32 K.** APILABLE DIRECTAMENTE CON LA MEMORIA DE 16 K DE SINCLAIR OBTENIENDO UN TOTAL DE 48 K. CONEXION PARA OTROS PERIFERICOS. PILOTO/AVISO DE FUNCIONAMIENTO. **P.V.P. 15.800**
- **MEMORIA DE 64 K.** MAXIMA CAPACIDAD DEL ZX81. ALMACENAMIENTO PARA DATOS BASIC CODIGO MAQUINA Y DIRECCIONAMIENTO DE DISTINTOS INTERFACES. PILOTO/AVISO DE FUNCIONAMIENTO. **P.V.P. 19.200**
- **GENERADOR DE SONIDO.** UTILIZA EL CHIP. AY-3-8910 CON 16 REGISTROS INTERNOS. 3 GENERADORES INDEPENDIENTES MAS UNO DE RUIDO Y OTRO ENVOLVENTE CON AMPLITUD VARIABLE. ESCALA DE 5 OCTAVAS. PILOTO DE FUNCIONAMIENTO. POTENCIOMETRO DE VOLUMEN Y SALIDA AMPLIFICADOR. **P.V.P. 7.950**
- **GENERADOR DE GRAFICOS.** PERMITE PROGRAMAR HASTA 128 NUEVOS CARACTERES GRAFICOS. DISPONE DE 1 K RAM PARA ALMACENAMIENTO DE DATOS E INTERRUPTOR ON/OFF. **P.V.P. 7.500**
- **D. SAVE.** CONTIENE 3 FUNCIONES PRINCIPALES: CARGAR (LOAD), GUARDAR (SAVE) Y VERIFICAR (VERIFY) DE PROGRAMAS EN 30 SEGUNDOS. ALCANZANDO UNA TASA DE 4000 BAUDIOS EN LUGAR DE LOS 250 DEL ZX81. **P.V.P. 6.900**
- **SET DE GRABACION.** COMPUESTO POR UN MEDIDOR DE NIVEL, UNA CINTA PARA ALINEACION DE LA CABEZA REPRODUCTORA DEL CASSETTE Y UN MANUAL DE 48 PAGINAS PARA RESOLVER LOS COMPLEJOS PROBLEMAS DE GRABACION DEL ZX81. **P.V.P. 3.500**
- **INVERSOR DE VIDEO.** INVIERTA LA SALIDA DE VIDEO DEL ZX81 PRESENTANDO LA PANTALLA CON FONDO NEGRO Y CARACTERES EN BLANCO. **P.V.P. 2.100**
- **INTERFACE CENTRONICS.** PERMITE CONECTAR EL ZX81 A CUALQUIER IMPRESORA PARALELO TIPO CENTRONICS. **P.V.P. 14.950**

BOLETIN DE PEDIDO

CANTIDAD	DESCRIPCION	PESETAS
EMPAQUETADO Y TRANSPORTE		PEDIDOS SUPERIORES A 10.000 PTAS. 500
		PEDIDOS INFERIORES A 10.000 PTAS. 300
		TOTAL

• INCLUYO CHEQUE NOMINATIVO A FAVOR DE INDESCOMP, S.A. POR _____ PTAS.

• REMITAN EL PEDIDO CONTRA REEMBOLSO A:

D. _____ DIRECCION _____

N.º _____ PROVINCIA _____ TEL. _____ PROFESION _____

INCLUYANME EN SU BANCO DE MAILING ☐ ENVIEME CATALOGO GENERAL ☐

De venta en



Sonytel

y distribuidores autorizados o directamente en INDESCOMP, S.A. por teléfono 279 31 05 (5 líneas) o remitiendo el cupón adjunto.

¡ Música, maestro!

Hasta cierto punto, se puede formar una orquesta gracias al sintetizador musical de tres voces del **Commodore 64**, opción que convierte al ordenador en instrumento musical. El **Commodore 64** simula los sonidos de varios instrumentos musicales mejor que cualquier otro ordenador personal existente actualmente. Las tres voces independientes del sintetizador implican que se puede dar tres notas distintas simultáneamente. Cada una de estas tres voces puede sonar como un instrumento distinto. Puede ser órgano, violín, cobre, madera, cuerda, piano, arpa, tambor, guitarra o sonidos de efectos especiales. Se logra así una casi infinita combinación de sonidos gracias a la flexibilidad del *chip* SID. El sintetizador da un gran control sobre la calidad de las tres voces. Cada voz tiene un margen de ocho octavas, un generador de sonido envolvente con control programable y elección de onda. La oferta culmina con un control de volumen para todas las voces y un filtro para diferentes frecuencias.

La escala musical en ocho octavas es índice de que el **64** tiene aspiraciones. El generador ADSR (*attack/decay/sustain/release*) define la velocidad con la que una nota se eleva desde el silencio a su máximo volumen (*attack*), la velocidad promedio en la que pasa de su volumen máximo al sostenido (*decay*), el nivel del volumen sostenido (*sustain*) y la velocidad promedio en que pasa de éste al silencio (*release*). Un redoble de tambores, por ejemplo, tiene una forma ADSR muy simple: ataque instantáneo y decaimiento sin sostenido ni *release*. Un violín, en cambio, es un instrumento sostenido con diferentes tasas de *attack*, *decay* y *release*. Las tres formas de onda del sintetizador afectan la calidad de un sonido. Suave y aflautado, súbito y estridente, brillante (tirando a nasal) y agudo. El generador de ruidos produce sonidos que van desde el bajo cálido hasta el alto y estridente. El filtro del sonido programable permite tres tipos de filtro para lograr el exacto sonido que se desea. Puede ser usado para un

pase a alta frecuencia, a baja frecuencia, frecuencia de banda y como filtro de rechazo. Este último impide el paso de frecuencias medias. También es posible el vibrato y el tañir de campanas. El sintetizador puede producir sonidos que los instrumentos musicales son incapaces de reproducir. Al aplicar a la estructura armónica de un instrumento la forma de onda de otro, se pueden crear, de hecho, instrumentos nuevos. Los efectos de sonido tales como explosiones, viento o campanas pueden ser usados en juegos y en otros programas interactivos. Tocar una de las 94 notas exige encender el *master* de volumen, establecer las pautas de *attack/decay* y los niveles de *sustain/release*, entrar los valores de alta y baja frecuencia de la nota y dar comienzo por fin a la forma de onda. Para apagar una nota, soltar simplemente la forma de onda. Para tocar una segunda nota sólo deben ser cambiados los valores de la nota y la forma de onda. El BASIC de **Commodore** carece también de comandos para música, así que se empleará el POKE para localizaciones de memoria. El paquete de apoyo de música y video, que es parte del BASIC opcional de expansión, provee todo un conjunto de comandos fáciles de usar para poder hacer música. El conector de la parte posterior del ordenador ofrece un *output* de radio a un amplificador y altavoz externo. Conectarlo al propio sistema estéreo da una mucho mejor audición que el altavoz del receptor de TV. También se da un *input* de audio para quien se aventure a combinar o filtrar sus instrumentos o voces por medio del sintetizador.

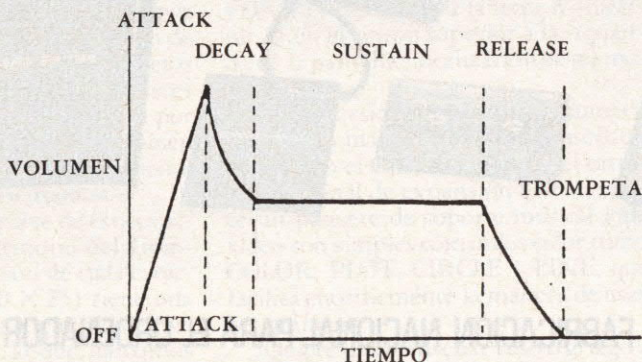


Fig. 1: Esquema de sonido envolvente de un instrumento típico, en este caso la trompeta.

resultan de tamaño suficiente, puede duplicar el tamaño en sentido horizontal, vertical o en ambas direcciones. Pero esto en detrimento de la resolución, porque simplemente se expande el *sprite* existente. Un modo de *sprite* multicolor implica resolución horizontal para hasta cuatro colores en cada *sprite*.

Cuáles son los *sprites* que pasan frente a otros está determinado por el número asignado a cada uno de ellos. El *sprite* 0, que tiene la mayor prioridad, aparecerá más cercano que el 1, porque pasará por encima de éste cada vez que se encuentren. También se puede hacer pasar a cada *sprite* por delante o por detrás de objetos que están sobre el

fondo. Las colisiones entre *sprites* o entre *sprites* y objetos de fondo se detectan por separado y los resultados pueden ser empleados en juegos o en otros tipos de animación gráfica.

Es muy posible escribir aceptables juegos de estilo *arcade* en BASIC, porque el *chip* especial de video maneja rápidamente la posición y desplazamiento de los *sprites*. Estos son tan independientes que permanecerán en pantalla aun cuando haya interrumpido el programa BASIC al pulsar la tecla RUN/STOP. Sin embargo, si pulsa simultáneamente las teclas RUN/STOP y RESTORE, limpiará la pantalla.

Al igual que con otros modos gráficos, la versión 2.0 de BASIC de **Com-**

modore no tiene ningún comando específico para los *sprites*. Toda la actividad de estos "duendes" se maneja por medio de los comandos POKE a las localizaciones de memoria, a menos que tenga conectado el paquete de apoyo de video y música en el cartucho de expansión BASIC, que provee comandos para un más fácil desplazamiento de los *sprites*.

Mucha gente compra sus grabadores a *cassette* o sus *drives* de discos con la idea de almacenar programas y datos. El grabador VIC-1530 Datasette se adecúa sólo a un uso limitado, debido a su lentitud de acceso. Permite almacenar información en un audio *cassette* barato a expensas de la velocidad. Mucha gente preferiría el grabador de cintas para la

GRAN PROMOCION, EN MICROPROCESADORES



CASIO FX900P

- Lenguaje basic • Memoria 4 K, ampliable a 32 K • Alta resolución
- Teclado profesional • Periféricos disponibles: cassette - impresora - unidad de disco.



65.800 Ptas.

Genie color

- Memorias RAM 16 K ampliables internamente a 32 K • Lenguaje Extended BASIC Microsoft • Salida para Monitor Audio, Video y de RF para TV Color y B/N.



Sinclair ZX Spectrum

- RAM-16 K o 16 K o 48 K
- 8 Colores disponibles para primer plano fondo y borde • Sonido-orden BEEP con intensidad y duración variables • gráficos en alta resolución en pantalla e impresora de 256 x 192 puntos. • Grabación y carga en cassette a alta velocidad para programas y/o datos.



**A = 75.000 Ptas.
A.D. = 83.000 Ptas.**

NewBrain

- Lenguaje Basic • Memoria RAM 32 K • Memoria ROM 29 K • Alta Resolución 640 x 250 D. • Salida para TV, Monitor, 2 cassettes, Disquettes e impresora.

SOLICITE INFORMACION



**ELECTRONICA
SANDOVAL S.A.**

COMPONENTES ELECT. PROFESIONALES
VIDEO — TV. COLOR — RADIO
Sandoval. 3 — Telef. 445 75 58 — 445 76 00
Sandoval. 4 — Telef. 447 42 01 — 445 18 33
Sandoval. 6 — Telef. 447 45 40 — 445 18 70
Telex: 47784 SAVL MADRID-10

cantidad de juegos disponibles, más baratos que los de cartucho. El **Datasette** tiene un conector especial que ajusta el interfase del *cassette* en la parte posterior del ordenador. Otros grabadores en cambio presentan dificultades para su adaptación.

La configuración más popular a la venta es el **Commodore 64** con un **VIC-1541** de un solo *drive*. Almacena 170 Kbytes por disco de 5 1/4 pulgadas de una sola densidad. Es el doble de la capacidad del *drive* del **Atari** (88 Kbytes) y sustancialmente más que el del **Apple II** (114 Kbytes). Sólo difiere del *drive* **VIC-1540** por un nuevo *chip* ROM. Los discos realizados en el **VIC-1540** y en el más antiguo *drive* dual, el **CBM 4040**, pueden ser leídos con el 1541.

Hasta unos cuatro *drives* de *floppys* pueden ser conectados con una serie de cables desde la puerta serial bus. Al usar comunicación serial (un bit por vez) para los discos, se supone que disminuye la velocidad de rendimiento del *drive*, pero los del **Apple II**, que usan comunicaciones paralelas (8 bits enviados simultáneamente por 8 cables) tampoco son más rápidos.

El controlador de disco (con su propio microprocesador 6502) y del sistema operativo están incorporados al *drive*. Esta inteligencia permite al ordenador pasar una orden al *drive* y luego seguir ocupándose de su propio trabajo mientras el *drive* cumple con lo ordenado. No resulta tan conveniente en la eventualidad de un error: el *drive* titila, pero el ordenador desconoce el error

hasta que recibe la orden expresa de interrogar al *drive*. Por ejemplo, si trata de conservar un programa en un disco que ya estuviera completo, el único resultado sería un centelleo en el *drive*. Sólo cuando preguntara por el error específico tendría una respuesta inmediata y clara.

Un programa BASIC puede ser duplicado con un solo *drive*, cargando el programa y lo puesto de lado en otro disco.

La impresora gráfica de matriz de puntos **VIC-1525** reproduce todo el conjunto de caracteres del **Commodore** y ofrece gráficos de puntos, pero es tan lenta que no imprime más de 30 caracteres por segundo. La habilidad de imprimir caracteres gráficos es muy conveniente si se los usa en sus programas BASIC, y debiera darse la posibilidad de imprimir cualquier texto o gráfico que aparezca en la pantalla. Esta impresora no tiene verdaderas minúsculas en letras como g, p, y, sino que eleva estos caracteres en la línea de impresión. Se puede conectar esta impresora a la puerta serial o al último *drive* conectada a la puerta serial.

También es posible añadir otras impresoras **Commodore** usando el interfase estándar IEEE-488. Esta opción conecta por medio de una clavija en la ranura de los cartuchos y le permite usar la **Commodore 4022** (una versión de la **Epson MX-80**). Hay interfases paralelos de impresora, disponibles en otras marcas, que también pueden servir para conectar impresoras estándar, realizan-

do la conversión necesaria de los caracteres estándar a los ASCII.

El **Commodore 64** provee todas las señales de comunicación serial estándar RS-232C, pero no a los voltajes estándar. Por ello, se hace necesario un interfase RS-232C, que permite al **Commodore 64** comunicarse con muchos *modems*, impresoras, sintetizador y demás accesorios que usan este interfase estándar.

El *modem* **VIC-1600**, un cartucho de interfase de teléfono que se conecta a la puerta del usuario y le permite comunicar por medio de la red telefónica con servicios de datos y con otros ordenadores a 300 bits por segundo, estará, al menos en teoría, en el catálogo español de **Commodore**. Pero lo cierto es que el único acoplador acústico homologado por la **Compañía Telefónica Nacional** es el modelo 8010, ya conocido por los usuarios de la marca.

Por otra parte, ya está disponible un cartucho, basado en un microprocesador **Z80**, que permite al **Commodore 64** usar el sistema operativo **CP/M**. El disco que acompaña al **CP/M** da 52 Kbytes y un área de programa de la misma magnitud. Una buena parte de los usuarios de **Apple II** en todo el mundo ha adquirido una opción conectable del mismo tipo, lo que permite presagiar que el cartucho **CP/M** puede difundirse bastante como opción para el 64.

Mientras el **CP/M** se ha establecido, de hecho, como un estándar en ordenadores de 8 bits, no puede decirse lo mismo de los diferentes formatos de disco que se encuentran en el mercado. El caso es que los *drives* **VIC-1541** presentan otro formato no estándar para los fabricantes de software en **CP/M**. Y esto también puede ser un problema para el usuario.

Y así llegamos al momento de las conclusiones. Al igual que el original "escarabajo" de Volkswagen, al que aludíamos al comienzo, el **Commodore 64** ofrece la posibilidad de una pequeña inversión inicial, gran economía y accesorios baratos. No a todos les gustará su diseño, pero su precio deja a otros ordenadores mal parados. Lo cierto es que sus características estándar le otorgan al **Commodore 64** una posición muy sólida en el mercado durante los próximos años, a condición de que el fabricante y sus distribuidores sepan cumplir con la promesa de ofrecer *software* abundante y servicio garantizado. Es muy posible que la aparición de este modelo ayude a relanzar la guerra de precios entre los ordenadores de baja gama.

Bruce Roberts

© Popular Computing/Ordenador Popular

FICHA

Nombre: Commodore 64

Fabricante: Commodore Business Machines

Representante en España:

Microelectrónica y Control

Taquígrafo Serra, 7 - 5.º

Barcelona-29

Tfno.: (93) 250 51 03

Características estándar:

- medidas: 40,4 × 21,6 × 7,5 centímetros
- peso: 1,820 kilos
- microprocesador MOS 6510
- memoria 20 Kbytes de ROM y 64 Kbytes de RAM
- teclado Qwerty de 62 teclas mayúsculas y minúsculas más 4 de función
- pantalla de 40 columnas, 25 filas. 16 colores, con 8 bloques y objetos móviles definibles por el usuario.
- Gráficos de alta resolución de 320 × 200 puntos

- sintetizador de sonido. Tres voces de ocho octavas cada una
- Entradas/salidas para discos e impresoras (RS-232C). Port para cartuchos. 2 E/S para joystick, paddles y lápiz óptico.

Periféricos:

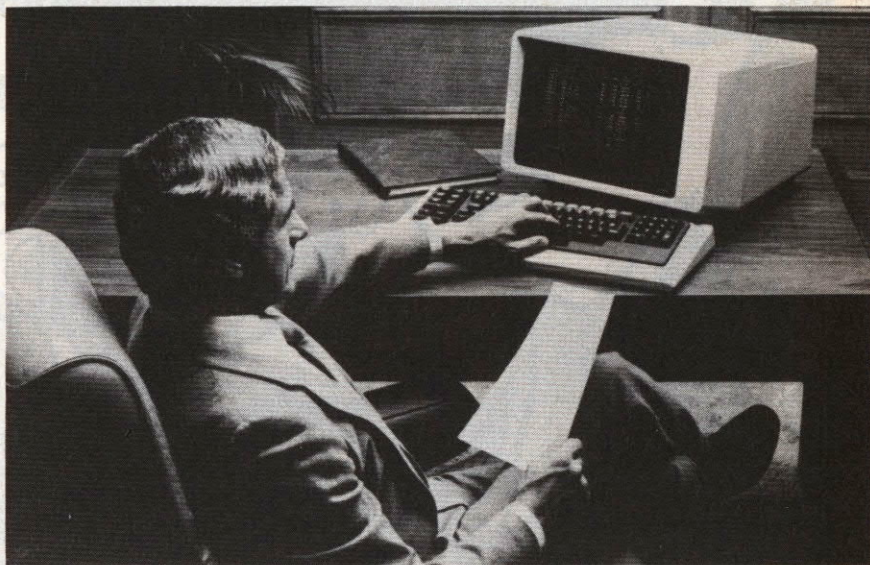
- hasta cuatro drives de discos floppy de 5 1/4 pulgadas (170 Kbytes cada uno)
- impresora VIC 1525
- grabador de audio cassette VIC 1530 Datasette
- acoplador acústico 8010
- cartucho Z80, para sistema operativo CP/M (disponible a partir de julio de 1983)

Lenguaje: intérprete interno de BASIC

Precios:

unidad básica: 110.000 pesetas
unidad de disco 89.620 pesetas
impresora 59.850 pesetas

Escuche cómo su ordenador rompe las barreras del lenguaje.



Es digno de verse: las caras de los directores utilizando nuestro Sistema MAPPER™.

¡Conexión! El hombre y la máquina están hablando el mismo lenguaje. Aislemos estos datos y comparémoslos. Siga esta idea. Examine esta relación.

Infórmese para conseguir una mente penetrante. Particularmente, después de años de operaciones frustradas a causa de un departamento de datos bloqueado que tarda dos días en facilitar las respuestas. O dos meses. Y cuando la respuesta sugiere otra pregunta, de nuevo el lento proceso.

Se acabó.

Se ha derribado la barrera del lenguaje. Ha llegado una nueva era. La del acceso directo, de obtener justamente lo que necesitaba saber, de respuesta instantánea, de capacidad inmediata

para explorar, de efectuar referencias cruzadas y obtener datos masivos útiles.

Los directores financieros, los de fabricación y los de marketing, pueden llevar a la práctica todo el potencial creativo de sus trabajos.

Señalar problemas. Resolverlos mientras sean todavía pequeños. Examinar modelos y relaciones que sugieren nuevas estrategias. Explorar opciones y trayectorias alternativas de decisión.

Y, como oímoslo decir a nosotros tiene menos utilidad que una experiencia real, hemos diseñado



Sabemos cuán importante es escuchar

el "MAPPER". Usted nos propone un problema actual y nosotros le demostramos cómo resolverlo por sí mismo. Cómo tratar directamente con el ordenador. En su propio lenguaje.

Si desea llevar a cabo una experiencia que podría alterar definitivamente su eficacia profesional, póngase en contacto con nosotros.

Seguimos escuchando.

MAPPER



Aprenda Basic con Sherlock Holmes

IV. Estudio de la ceniza del tabaco

"Verá, Watson —observó Sherlock Holmes una fría tarde en que estábamos analizando un informe según el cual la desaparecida baronesa Whitelsey había sido vista en Viena— creo que gran parte de mi éxito profesional se debe al hecho de que encaro mi trabajo tomándolo como un arte y, a la vez, como una ciencia. A diferencia de la mayoría de mis colegas detectives, nunca he pensado en mi labor como una esclavitud. En su forma más pura, el trabajo de un detective es razonamiento deductivo, sólo comparable a las matemáticas, por su elegancia y por lo que tiene de desafío intelectual. Esta es la razón de que la Máquina Analítica, basada en principios matemáticos, me haya parecido la herramienta más adecuada para mi tarea".

"¿Cuál es su próximo plan para hacer uso de la Máquina?", pregunté, presintiendo que mi amigo estaba a punto de lanzar alguna idea.

"Mi plan, Watson, es usar esta notable Máquina como almacén para algunos detalles que abarrotan mi cabeza. Tomemos, por ejemplo, mi monografía sobre "La diferencia entre las cenizas de varios tabacos" —dijo, señalando un

polvoriento volumen que se encontraba frente a nosotros— en la que describo y clasifico 140 tipos de cigarrillos, cigarrillos y tabacos para pipa, con rótulos coloreados que ilustran las diferentes clases de ceniza. Pese a que me esmeré especialmente en recordar detalles tales como la exacta apariencia de las cenizas de un McDuffy, en contraste con las de los cigarrillos Lunkah, lo cierto es que a la mayoría de los investigadores que usted y yo conocemos les faltaría paciencia para hacer lo mismo. Y no los culpo por ello.

"Después de todo —añadió— el cerebro es muy parecido a un ático, amplio pero de capacidad limitada, que llenamos con todo lo que nos parece importante para el futuro. Dado que las paredes de ese ático no son elásticas, a medida que seguimos acopiando más y más información, la más antigua debe dejar lugar a la más nueva. Sería muy útil tener un dispositivo para recordar una cantidad de datos complementándolos con informaciones sobre esos mismos datos cada vez que los necesitemos. Y estoy seguro, Watson, que la Máquina Analítica está maravillosamente estructurada para esa tarea.

"Como mero ejercicio para poner a prueba esta idea, he preparado una tabla con las diez mejores clases de cigarrillos más comúnmente fumadas en Londres".

Mientras estudiaba este documento, Holmes se acercaba al fuego y, cogiendo una cajita de bronce de encima de la chimenea, la abrió y olió el único cigarro que en ella había.

"Me gustaría diseñar un programa para identificar los cigarrillos con determinadas y específicas características. Además, si las características establecidas no concordaran con ninguna de estas diez marcas de cigarro, la Máquina podrá hacerlo notar para que entonces pueda investigar yo mismo el tema".

"¿Puedo dar por supuesto que tiene usted algún método para instruir a la Máquina en la manera de buscar datos?"

Apenas lo dije, me di cuenta de que esta pregunta obedecía a mi inexperiencia en la construcción de programas.

"Precisamente, Watson. El punto clave es la necesidad de tomar decisiones y de emprender las acciones consecuentes en una determinada condición. Claro que este es un problema muy común en la labor detectivesca".

"Hay cierto número de combinaciones que podemos hacer con las varias condiciones. Podemos, por ejemplo, especificar que determinada acción sea tomada bajo determinadas circunstancias. Por ejemplo:

si textura es migosa
ejecutar acción A"

"Otra situación que surge es la de dos

Bull, así de sencillo

**El árbol más famoso de la informática
se poda el nombre
para crecer todavía más fuerte.**



USTED ya conoce el árbol. Es el emblema de una de las compañías internacionales que más han extendido la informática por el mundo. Y quizá, con el tiempo y con la práctica, haya llegado a aprenderse también su nombre. Se llamaba, nos llamábamos, Cii Honeywell Bull, un nombre casi tan extenso como nuestra organización, pero mucho menos asequible.

No era lógico. Sobre todo para una compañía como la nuestra, que ha basado

su éxito en hacer sencillo lo complejo. Y sobre todo ahora que seguimos acelerando nuestra expansión con más entu-



HONEYWELL BULL, S. A.
ARTURO SORIA, 107 - MADRID-33
TELS. (91) 413 12 13 - 41332 13

siasmo, con más recursos y con más soluciones que nunca.

Así es que lo hemos podado. Ahora la compañía del árbol se llama Bull, una sola sílaba, inequívoca y con la misma pronunciación evidente en cualquier parte del mundo.

Bull, así de sencillo. Puestos a simplificar, reconozca que se lo hemos puesto fácil. Y casi tan corto como el tiempo de respuesta de nuestros ordenadores.

TIPO DE CIGARRO	TEXTURA	COLOR	PARTICULAS	NICOTINA
Espanada	migoso	oscuro	no	++
Heritage	escamoso	gris claro	no	++
Latino	mezcla	oscuro	si	+
Londoner	migoso	amarronado	no	++
Lunkah	granulado	gris oscuro	no	++
MacDuffy	escamoso	gris oscuro	no	++
Old Wood	mezcla	amarronado	variado	+++
Top Hat	migoso	gris oscuro	no	++
Trichinopoly	escamoso	oscuro	no	++
West Country	esponjoso	gris claro	no	++

acciones posibles con una elección entre ambas que dependa de una sola combinación. Por ejemplo:

si textura es migosa

ejecutar acción A

si no

ejecutar acción B

"Por fin, podemos encarar un cierto número de cursos de acción posibles, en los cuales nuestra elección dependa de una entre varias condiciones:

si textura es migosa

ejecutar acción A

si no

si textura es escamosa

ejecutar acción B

si no

si textura es granulada, entonces

ejecutar acción C

si no (si no se cumple ninguna de las condiciones), entonces

ejecutar acción D

"Comprendo lo que quiere decir, Holmes, pero ¿cómo podemos resolver el problema en términos comprensibles para la Máquina Analítica?"

"Muy simple, Watson. Todo lo que tenemos que hacer es organizar la decisión en forma de un algoritmo consistente y traducirlo luego al lenguaje de la Máquina. Aquí tengo una lista de todas las opciones de propiedades de un cigarro. Observe que el cigarro normal no tiene partículas y tiene un contenido de nicotina marcado con dos cruces. Además, hay dos clases básicas de ceniza de cigarro. Está el Stock 1, que es tanto escamoso como migoso, y el Stock 2, cuyas características son esponjoso y granulado".

La siguiente es la reproducción de la clasificación de cenizas hecha por Holmes:

COLOR: oscuro, gris oscuro, gris claro, amarronado.

TEXTURA: escamosa, migosa, granulada, esponjosa.

PARTICULAS: no, sí.

NICOTINA: +, ++, +++.

FUERZA NORMAL: ++, sin partículas.

CIGARRO STOCK 1: escamoso o migoso.

CIGARRO STOCK 2: granulado o esponjoso.

"Mi estrategia, Watson, es ordenarle a la Máquina leer una lista de las propiedades pertenecientes a la ceniza del cigarro. Entonces, el programa determinará si el cigarro tiene una fuerza normal y si pertenece al grupo 1 o al grupo 2. Una vez establecidas estas respuestas, será capaz de determinar si el cigarro es de uno de los diez tipos que figuran en la lista. De ser así, nombrará el cigarro; de no ser así, lo informará y se limitará a señalar la clase o fuerza del cigarro".

Luego, Holmes realizó su algoritmo, que publico como figura 1. Me pareció muy claro y casi instantáneamente comprendí su lógica.

"Pero, Holmes, la Máquina no reconoce las propiedades de la ceniza de cigarro. ¿Cómo podrá distinguir entre la

Figura 1: Algoritmo de Holmes para la identificación de cenizas de cigarro. El algoritmo define un método para resolver un problema. El programa de la figura 2 incorpora este algoritmo para su uso en ordenador.

Definiciones:

COLOR: un color de ceniza.

TEXTURA: una textura de ceniza.

PARTICULAS: una indicación de partículas.

NICOTINA: un resultado de test de nicotina.

STOCK: una clase de cigarro.

NORMALIDAD: una indicación de partículas y nicotina.

Algoritmo:

Input COLOR, TEXTURA, PARTICULAS, NICOTINA.

Si TEXTURA = ESCAMOSO o TEXTURA = MIGOSO entonces

sea STOCK = 1

sino sea STOCK = 2.

Si NICOTINA = ++ y PARTICULAS = NO entonces

sea NORMALIDAD = NORMAL sino

sea NORMALIDAD = ANORMAL.

Si NORMALIDAD = NORMAL y STOCK = 1 entonces.

Si COLOR = OSCURO y TEXTURA = MIGOSO entonces

imprimir "CIGARRO ES UN TRICHINOPOLY".

Si no si COLOR = OSCURO y TEXTURA = MIGOSO entonces

imprimir "CIGARRO ES UN ESPANADA".

sino si COLOR = GRIS OSCURO y TEXTURA = MIGOSO entonces

imprimir "CIGARRO ES UN MAC DUFFY".

Si no si COLOR = GRIS OSCURO y TEXTURA = MIGOSO entonces

imprimir "CIGARRO ES UN TOP HAT".

Si no si COLOR = GRIS-CLARO y TEXTURA = ESCAMOSO entonces

imprimir "CIGARRO ES UN HERITAGE".

Si no si COLOR = AMARRONADO y TEXTURA = MIGOSO entonces

imprimir "CIGARRO ES UN LONDONER".

Si no imprimir "CIGARRO NO IDENTIFICADO NORMAL DE STOCK 1".

Si NORMALIDAD = NORMAL y STOCK = 2 entonces.

Si COLOR = GRIS-OSCURO y TEXTURA = GRANULADO entonces

imprimir "EL CIGARRO ES UN LUNKAH".

Si no si COLOR = AGRISADO y TEXTURA = ESPONJOSA entonces

imprimir "CIGARRO ES UN WEST COUNTRY".

Si no imprimir "CIGARRO NO IDENTIFICADO NORMAL DE STOCK 2".

Si NORMALIDAD = ANORMAL entonces

si COLOR = AMARRONADO y NICOTINA = +++ entonces

imprimir "CIGARRO ES UN OLD WOOD".

Si no si COLOR = OSCURO y NICOTINA = + y PARTICULAS = SI

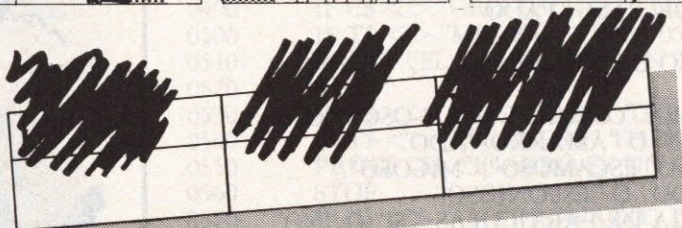
entonces imprimir "EL CIGARRO ES UN LATINO".

Sino imprimir "CIGARRO NO IDENTIFICADO ANORMAL".

VICTOR.ORDENADOR PROFESIONAL.



FINANCIACION
EN CONDICIONES
ESPECIALES
DE LA CAJA DE AHORROS
DE MADRID



Unos venden amigos, otros competencia. Nosotros: el mejor ordenador profesional.

Si usted está interesado en comprar un ordenador profesional habrá visto muchos anuncios sobre el tema. Habrá observado que en muchos de ellos se habla de que el ordenador es un amigo (vaya usted a saber por qué), en otros se dedican a compararse con la competencia (no con toda, claro) y, en el mejor de los casos, hablan con un lenguaje que no entiende nadie más que el propio fabricante.

Nosotros, en cambio, vamos a decirle las cosas para que las entienda. Nosotros sólo fabricamos ordenadores profesionales, es decir, somos especialistas y, por eso, fabricamos el mejor: el Víctor. El que más capacidad de memoria tiene, el de más sencillo manejo y mayores posibilidades.

Y para acabar de una vez, se lo servimos completo. Desembalar, conectar y listo. Sin accesorios y facturas aparte. Pero con enormes posibilidades de ampliación.

No queremos aburrirle con datos sobre nuestra pantalla de alta resolución, capacidad y otras ventajas. Mejor nos escribe.

Nos entenderemos porque, como nuestro ordenador se fabrica totalmente en Estados Unidos, no hablamos en chino.

VICTOR



GRUPO BAHRC

POP 1

Si quiere que le informen a fondo sobre el computador profesional VICTOR, envíe este cupón a Computronic.

Nombre

Empresa Tel.

Dirección D. P.

Ciudad



Computronic, S.A.
Para más datos.

Figura 2: Listado del programa-BASIC para identificar cenizas de cigarros. La línea de unión en subrayado es usada sólo en el caso del aspecto, y puede ser reemplazada por un espacio o un guión, según la preferencia. También se puede obviar la marca O en las referencias del número de línea.

```

0010 REM —ESTE PROGRAMA LEE CUATRO PROPIEDADES DE
      CENIZAS DE CIGARRO
0020 REM —LAS PROPIEDADES ESTAN DADAS COMO CADENAS
      DE CARACTERES
0030 REM —EL PROGRAMA TRATA DE IDENTIFICAR LA CENIZA
0040 REM —SEGUN SUS PROPIEDADES IMPRIMIR UN
      MENSAJE CON RESULTADO
0050
0060 REM —DICCIONARIO DE NOMBRES:
0070 REM
0080 REM —C$ COLOR: SEA "OSCURO", "GRIS-OSCURO"
0090 REM —"GRIS-CLARO" O "AMARRONADO"
0100 REM —T$ TEXTURA: "ESCAMOSO", "MIGOSO",
      "GRANULADO" O "ESPONJOSO"
0110 REM —P$ PRESENCIA DE PARTICULAS: "SI" O "NO"
0120 REM —N$ CONTENIDO NICOTINA: "+", "++", O "+++"
0130 REM —S STOCK DE CIGARRO: 1 O 2
0140 REM —M$ NORMALIDAD DEL CIGARRO: "NORMAL" O
      "ANORMAL"
0150 REM
0160 REM
0170      PRINT INTRODUZCA COLOR, TEXTURA,
      PARTICULAS, NICOTINA
      INPUT C$, T$, P$, N$
0180      IF T$ = "ESCAMOSO" THEN 0230
0190      IF T$ = "MIGOSO" THEN 0230
0200      LET S = 2
0210      GOTO 0270
0220      LET S = 1
0230      GOTO 0270
0240
0250 REM FINAL DE IF
0260 REM
0270      IF N$ <> "++" THEN 0310
0280      IF P <> "NO" THEN O
0290      LET M$ = "NORMAL"
0300      GOTO 0350
0310      LET M$ = "ANORMAL"
0320      GOTO 0350
0330 REM FINAL DE IF
0340 REM
0350      IF M$ <> "NORMAL" THEN 0640
0360      IF S <> 1 THEN 0640
0370      IF C$ <> "OSCURO" THEN 0410

```

que tiene textura granulada y la que la tiene escamosa?"

"Cierto, Watson, la Máquina no reconoce las propiedades de la ceniza de cigarro. Es al programador a quien le cabe distinguir las propiedades de la ceniza de cigarro y luego interpretar esta información para la Máquina en términos comprensibles para ella. Las características distintivas estarán cifradas en cadenas de caracteres y leídas en esa forma por el dispositivo. Así, por ejemplo, usaré ESCAMOSO por escamoso y GRIS—OSCURO por gris oscuro. Y uso el subrayado en vez de espacio para poner especial énfasis en que dos o

más palabras van unidas para formar una frase. La Máquina leerá cadenas de caracteres que corresponden a cada una de las propiedades: color, textura, partículas y nicotina respectivamente, como le demostraré".

Y tomando una hoja de papel esbozó Holmes el siguiente ejemplo:

Ejemplo de entrada: OSCURO, ESCAMOSO, NO, ++

Ejemplo de salida: CIGARRO ES TRICHINOPOLY.

"Dicen, Watson, que el genio es una capacidad infinita para trabajar con afán. Es una mala definición, aunque se aplica a la programación. ¡Cuánto alivio



nos va a proporcionar la Máquina Analítica! Propongo que dediquemos algunos años a la composición de un texto que reúna todo el arte de la labor detectivesca y los especiales usos de la Máquina Analítica, en un solo volumen".

"Menuda tarea, Holmes. Seguramente constituirá su mayor contribución a la ciencia y a la humanidad".

Estas palabras mías produjeron un súbito flujo de sangre en las mejillas de Holmes. Arrebolado, hizo una pequeña reverencia, a la manera del director de escena que recibe el homenaje de su público. Ese mismo hombre orgulloso y reservado, que desdeñaba la fama y la popularidad, era capaz de sentirse conmovido ante la espontánea maravilla y aplauso de su amigo.

En cuanto a nuestro estudio sobre la ceniza de los cigarros, Holmes aplicó el programa de la figura 2 para probar la efectividad de la Máquina medida con su propia capacidad. Pocas cosas me han producido más placer que seguir a Sherlock Holmes en sus investigaciones y admirar sus deducciones, siempre fundadas en la misma base lógica con que operaba la Máquina, con la que desvela-



ba los muchos problemas que se le planteaban.

La habilidad para tomar decisiones es fundamental en programación. Dependiendo de una o más circunstancias, queremos emprender acciones apropiadas. El mecanismo básico para efectuar opciones en BASIC es la sentencia IF. Esta sentencia tiene las siguientes formas:

IF condición THEN número de línea
Esta sentencia significa:

Si la condición es cierta, transferir la ejecución a la sentencia con el número de línea dado, de otra manera no hacer nada y seguir a la próxima sentencia.

La mayor parte de las veces una sentencia IF aparecerá en una secuencia de sentencias. Como resultado de esto, si la condición no es cierta, se procesa la próxima sentencia. Por ejemplo, podemos tener sentencia 1.

IF condición THEN número de línea
sentencia 2.
lo cual significa:

1. Ejecute sentencia 1.
2. Ejecute la sentencia IF: es decir, si la condición es cierta transfiera a la línea indicada.

```

0380 IF T$ <> "ESCAMOSO" THEN 0410
0390 PRINT "EL CIGARRO ES UN TRICHINOPOLY"
0400 STOP
0410 IF C$ <> "OSCURO" THEN 0450
0420 IF T$ <> "MIGOSO" THEN 0450
0430 PRINT "EL CIGARRO ES UN ESPANADA"
0440 STOP
0450 IF C$ <> "OSCURO" "GRIS-OSCURO" THEN 0490
0460 IF T$ <> "ESCAMOSO" THEN 0490
0470 PRINT "EL CIGARRO ES UN MACDUFFY"
0480 STOP
0490 IF C$ <> "GRIS-OSCURO" THEN 0530
0500 IF T$ <> "MIGOSO" THEN 0530
0510 PRINT "EL CIGARRO ES UN TOP HAT"
0520 STOP
0530 IF C$ <> "GRIS-CLARO" THEN 0570
0540 IF T$ <> "ESCAMOSO" THEN 0570
0550 PRINT "EL CIGARRO ES UN HERITAGE"
0560 STOP
0570 IF C$ <> "AMARRONADO" THEN 0610
0580 IF T$ <> "MIGOSO" THEN 0610
0590 PRINT "EL CIGARRO ES UN LONDONER"
0600 STOP
0610 PRINT "****CIGARRO NO IDENTIFICADO NORMAL
DE STOCK 1"
0620 STOP
0630 REM
0640 IF M$ <> "NORMAL" THEN 0770
0650 IF S <> 2 THEN 0770
0660 IF C$ <> "GRIS-OSCURO" THEN 0700
0670 IF T$ <> "GRANULADO" THEN 0770
0680 PRINT "EL CIGARRO ES UN LUNKAH"
0690 STOP
0700 IF C$ <> "GRIS-CLARO" THEN 0740
0710 IF T$ <> "ESPONJOSO" THEN 0740
0720 PRINT "EL CIGARRO ES UN WEST COUNTRY"
0730 STOP
0740 PRINT "+++ CIGARRO NO IDENTIFICADO NORMAL
DE STOCK 2"
0750 STOP
0760 REM
0770 IF M$ <> "ANORMAL" THEN 0900
0780 IF C$ <> "AMARRONADO" THEN 0820
0790 IF N$ <> "+++" THEN 0820
0800 PRINT "EL CIGARRO ES UN OLD WOOD"
0810 SOTP
0820 IF C$ <> "OSCURO" THEN 0870
0830 IF N$ <> "+" THEN 0870
0840 IF P <> "SI" THEN 0870
0850 PRINT "EL CIGARRO ES UN LATINO"
0860 STOP
0870 PRINT "**** CIGARRO NO IDENTIFICADO ANORMAL"
0890 REM
0900 END

```

3. De no ser así, ejecute la sentencia 2.

Obsérvese que si la condición es falsa, las sentencias 1 y 2 son ejecutadas en secuencia. Si la condición es cierta, la sentencia 1 es ejecutada y luego la ejecución prosigue en otra parte.

La habilidad para incluir la sentencia IF en un programa tiene posibilidades

de muy largo alcance. Mientras que el mecanismo básico es de una simplicidad extrema, podemos lograr importantes resultados (tómese como testigo el programa de Holmes para identificar la ceniza de tabaco).

Henry Ledgard y Andrew Singer
© Random House/Ordenador Popular

Cómo programar la impresora MX-80 (primera parte)

"¿Por qué no puedo conectar mi nueva impresora y trabajar con ella como si fuera un tostador o una máquina de afeitar? Este manual dice que he de sentarme y aprender como si fuera un escolar. Espera que le ponga las manos encima a ese vendedor".

Realmente usted puede enchufarlo y hacer algunas cosas como sacar el listado de un programa o imprimir algo. Si le apetece. Pero no es hora de perder el tiempo con impresiones vulgares... Estamos sentados en la cumbre de la explosión del mercado de impresoras. Las nuevas impresoras de bajo coste están pisando fuerte y por eso es muy importante que las conozcamos.

La impresora **Epson MX-80** nos da una oportunidad de aprender más acerca de la nueva tecnología en impresoras, y ese es el propósito de esta serie de dos artículos.

"Muy bien —diga usted— estoy casi impresionado, pero, ¿por qué tengo que aprender todo esto? ¿Puede alguien escribir un programa que lo haga por mi automáticamente?".

Desafortunadamente, hay demasiadas aplicaciones para un solo ordenador y un rápido crecimiento del número de impresoras. Incluso si nuestras necesidades son estándar, llevará tiempo a un profesional escribir programas de calidad que hagan exactamente lo que queremos, que trabajen en nuestra impresora particular, y con nuestro ordenador en particular.

En este primer artículo, desarrollaremos un programa en BASIC para controlar la impresora. Tecleando y ejecutando el programa aprenderá cómo funciona la impresora y cómo el ordenador se comunica con ella.

¿Por qué la **MX-80**? Porque es la más vendida y tiene un precio razonable.

El programa de muestra

Cualquier empresa que tiene un producto que vender debe responder a las posibles preguntas de los clientes por medio de un folleto informativo. En este artículo desarrollaremos una carta de información para el cliente. En el proceso tendrá oportunidad de poner a su **MX-80** a trabajar y descubrir sus múltiples habilidades.

La carta será una publicidad de un producto ficticio llamado **MXWRITER**. ¿Qué hace este producto? Tendrá que seguir leyendo para saberlo.

La carta muestra la mayoría de las características de **MX-80** de tal forma que al finalizar tendrá una idea precisa de lo que son capaces de hacer usted, su ordenador y su impresora.

OSBORNE 1

LA RENTABILIDAD DE UNA PEQUEÑA INVERSION...

...PARA SU EMPRESA

...PARA SU PROFESION

...PARA SU TRABAJO

El ORDENADOR OSBORNE 1, ha sido diseñado para proporcionarle mayor eficacia en su trabajo ya que:

1. Su total portabilidad le permite llevarle con Vd. a cualquier lugar.
2. Con el programa **WORDSTAR** suministrado gratuitamente, se convierte en un tratamiento de textos ahorrando tiempo a Vd. y su secretaria.
3. Con el programa **SUPERCALC**, suministrado gratuitamente, proporciona una tabla de 256 filas y 64 columnas para datos alfanuméricos que pueden interrelacionarse entre si. El conocer en el momento ¿qué pasaría si...? es de esta forma un juego.
4. Su precio de 395.000 ptas., incluyendo **WORDSTAR**, **SUPERCALC**, **MAILMERGE**, **CPM**, **CBASIC**, y **MBASIC**, le permite amortizarlo a muy corto plazo.

- Disponemos de la aplicación específica que Vd. necesite.
- Consulte en distribuidores autorizados.

PARA INFORMACION ADICIONAL DIRIGIRSE A:



INVESTRONICA

Tomás Bretón, 21 Madrid-7
Tels. 468 01 00/468 03 00
Telex 23399 IYCO E

Muntaner, 565 - Barcelona
Tel. 212 68 00

Nombre

Domicilio

Ciudad

Provincia

Con el ordenador que utilizamos para reproducir este ejemplo (un HP-83) la sentencia que ordena a la impresora que imprima es PRINT. De acuerdo al ordenador que usted utilice esta sentencia será diferente. Apple utiliza PR#n donde n es el lugar donde la tarjeta de la impresora reside. Atari utiliza LPRINT o PRINT#7 con una sentencia OPEN. TRS-80 utiliza LPRINT, y así para las distintas marcas.

Impresión normal

Empecemos. Conecte su impresora, conecte su ordenador utilizando la versión del PRINT que su ordenador reconozca e introduzca estas líneas:

```
50 PRINT "S T A R F I R E"
60 PRINT TAB(4);"O P E R A T I O N"
  S"
70 PRINT TAB(8);"S O F T W A R E ,
  I N C ."
```

Presione Enter o Return después de cada línea de programa. Luego teclee RUN y presione Enter.

¿Qué ha impreso? Si el *output* se parece al primer trozo de la figura 1, es que ha encontrado la equivalencia de PRINT para su ordenador y que todas las conexiones están bien. Si no vuelva al manual de su ordenador y busque la sentencia de impresión, compruebe las conexiones y el papel. Le esperamos.

La salida de la impresora le muestra el modo normal del MX-80. La sentencia PRINT manda cadenas de caracteres entrecorridas a la impresora.

Observe que la función TAB del BASIC funciona igual que en la pantalla. Desafortunadamente, algunas versiones de BASIC no tienen la sentencia TAB, así que prescindiremos de ella siempre que sea posible. El MX-80 tiene sus propios medios para la tabulación que luego veremos.

El *output* queda muy bien pero está todo en mayúsculas. ¿Qué pasa con las minúsculas? El MX-80 las trata con facilidad pero algunos ordenadores requieren una modificación especial o un *software* especial para mandar minúsculas a la pantalla o a la impresora. Consulte con su manual.

Mientras tanto, para saber lo que hace nuestro sistema, teclee y ejecute estas pocas líneas. Mientras tecla podrá saberlo muy rápido, si su ordenador puede manejar minúsculas. Si no puede, use mayúsculas.

```
130 PRINT
190 PRINT "Norberto Gallego"
200 PRINT "Ordenador Popular"
210 PRINT "c/ Hermosilla 100"
220 PRINT "Madrid"
230 PRINT
240 PRINT "Estimado Sr. Gallego:"
250 PRINT
```

Una pequeña teoría de números

¿Qué es lo que hace el programa realmente cuando manda esas líneas a la impresora? ¿Empaqueta la línea 190 la K y la manda por correo aéreo? Difícilmente. De hecho la K no se manda como tal. En su lugar se manda un número que representa a la K. Toda la información se manda a la impresora como números entre el 0 y el 255 (aunque algunos ordenadores pueden mandar sólo del 0 al 127). El apéndice A de su manual de Epson le muestra cómo interpreta la MX-80 cada uno de esos números.

Para imprimir las líneas tal como se ha hecho, el BASIC convierte automáticamente cada carácter del *string* a un número y lo lanza a la impresora. La impresora recibe el número, lo convierte al carácter asociado y lo imprime.

Ahora que tratamos del envío de números a la impresora desde el BASIC, debe saber también las malas noticias. El BASIC no es muy seguro cuando transfiere números de nuestro programa a la impresora. Cada ordenador tiene sus propias particularidades, pero en general los números desde el 0 al 31 son a menudo interceptados por el BASIC y manda números completamente diferentes de los que se había intentado. Las únicas soluciones viables son evitar esos códigos problemáticos una vez que se han descubierto u obtener alguna otra forma de captar los códigos por la impresora (es decir POKE, OUT, lenguaje máquina, etc.). Contemplaremos esto en el próximo artículo.

¿Por qué todos estos líos y molestias con los números? Principalmente porque la MX-80 contempla algunos números como códigos de control especiales. La clave para manejar la impresora es conocer las secuencias de números que activan esas habilidades ocultas. Con los códigos apropiados podemos decir a la impresora qué imprimir, dónde imprimirlo, cuándo imprimirlo y cómo imprimirlo.

Anchora comprimida

Los códigos de control se envían nor-

malmente vía la función CHR\$ de BASIC. Añada estas líneas:

```
10 C$=CHR$(15) @ C2$=CHR$(18)
470 PRINT C$;"Comprimido"
510 PRINT C2$
```

y ejecútelas. La palabra "Comprimido" debería estar impresa en el modo comprimido.

El código de control que sitúa en el modo comprimido (15) se almacena en C\$ en la línea 10. El código que lo desactiva (18) se almacena en C2\$. En la línea 470 el modo comprimido se activa imprimiendo C\$. Como en la mayoría de las impresoras el modo comprimido permanece hasta que éste se desactiva, que es lo que se hace en la línea 510.

Anchora doble

Tanto la anchora normal como la comprimida pueden ser expandidas al doble de sus anchuras normales. Añada y ejecute las siguientes líneas:

```
20 D$=CHR$(14) @ D2$=CHR$(20)
130 PRINT C$
140 PRINT TAB(14);D$;"75 Long Stre"
  et"
150 PRINT TAB(14);D$;"New York, Ne"
  w York"
160 PRINT C2$
170 PRINT "1 de junio de 1.983"
180 PRINT
260 PRINT "Gracias por el interes"
  demostrado en nuestro ";D$;"MXWRIT"
  ER.";D2$;" Dejeme decirle"
270 PRINT "algo mas en relacion a"
  sus importantes características."
280 PRINT
```

La línea 20 establece los códigos para activar (12) y desactivar (20) el modo de doble anchora. Los *strings* de las líneas 140 y 150 se imprimen en doble anchora del modo comprimido. El modo de doble anchora es el único que se desactiva automáticamente al final de la línea. Es por lo que necesitamos D\$ (14) en las líneas 140 y 150. Por otra parte el modo comprimido no se desactiva hasta que se lo decimos. Lo activamos en la línea 130 y lo desactivamos en la 160. La línea 170 imprime en anchora normal, confirmando que la impresión comprimida se ha desactivado en 160.

Descubra en COMPUCENTRO ARGÜELLES el mundo del ordenador personal

y la importancia de llamarse **Tandy Radio Shack**

Tandy Radio Shack, un auténtico gigante americano, dispone de las líneas más avanzadas y representativas en el mundo del Ordenador Personal y a través de Compucentro Argüelles le captará sus problemas de gestión y le asesorará profesionalmente la manera más idónea de resolverlos.

APLICACIONES

- Contabilidad General.
- Control de Pedidos.
- Control de Almacén.
- Facturación.
- Edición de Textos.
- Control de Clientes y Pedidos.
- Control de Presupuestos.
- Declaración de la Renta (Simplificada).
- Nóminas (TC 1, TC 2 y IRPF).

LA IMPORTANCIA DE SABER ELEGIR
EL SERVICIO DE UN ORDENADOR PERSONAL



COMPUCENTRO ARGÜELLES

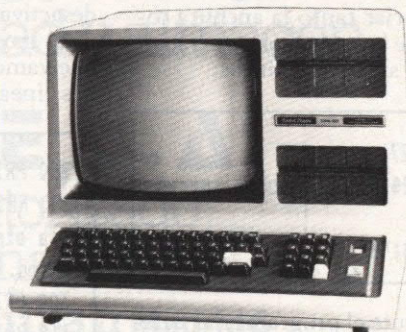
Martin de los Heros, 57-Madrid 8-Teléfs. 247 34 31-247 34 41



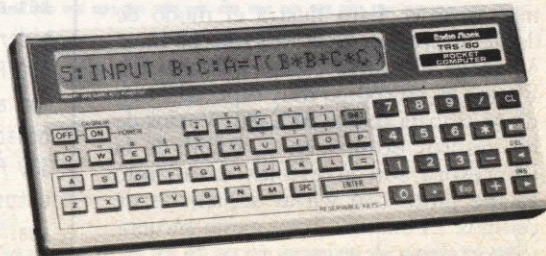
NUEVO ORDENADOR COLOR



MODELO II



MODELO III



PC-2

La línea 260 demuestra una de las características más agradables del modo de doble anchura: puede ser activado y desactivado dentro de una línea de impresión. Usando esta característica hacemos que la palabra **MXWRITER** resalte.

El modo de doble anchura es el único que puede ser activado y desactivado en la misma línea de impresión. Los otros que trataremos (incluyendo en modo comprimido) son proposiciones de todo o nada. Por ejemplo no podemos mezclar el modo normal con el comprimido en la misma línea. Pero podemos pasarnos al modo de doble anchura cuando queramos. Recuerde que el modo de doble anchura se desactiva al final de cada línea (ver línea 460 y 490). He aquí más instrucciones que utilizan el modo de doble anchura para nuestra carta:

```
350 PRINT "el ";D$;"MXWRITER";D2$;
" para poder ver los deslumbrantes
colores."
430 PRINT
450 PRINT "MXWRITER con impresion
normal"
460 PRINT D$;"Doble anchura"
490 PRINT C$;D$;"Comprimido con do
ble anchura"
```

Ejecute el programa expandido y vaya descubriendo las habilidades de su impresora.

Doble impresión

Podemos variar tanto la anchura como la calidad de la impresión de caracteres. Teclee las siguientes líneas:

```
30 E$=CHR$(27)
510 PRINT C2$;E$;"G";"Doble impres
ion"
520 PRINT E$;"H"
```

Cuando ejecute el programa, las palabras "Doble impresión" aparecerán más oscuras. Esto ilustra el modo de doble impresión de la **MX-80**. La línea 30 carga el código escape (27) en **E\$**. Escape es un código muy especial cuando se trabaja con la **MX-80**. Varios de los tesoros ocultos en la ROM de la impresora son accesibles mandando Escape junto con varias secuencias de códigos. El manual llama a esas secuencias códigos de escape.

La línea 510 utiliza nuestro primer código de escape. Escape (27 decimal) seguido por "G" (o **CHR\$(71)**) sitúa a la impresora en el modo de doble impresión. Escape "H" lo desactiva. Como en el modo comprimido la doble impresión continúa hasta que se desactiva.

En el modo de doble impresión la impresora hace dos pasadas por la misma línea con un pequeñísimo avance del papel entre las pasadas. El resultado son caracteres más resaltados. Ahora bien: la cinta se gasta antes, pero es el precio del estilo.

Impresión enfatizada

Otra opción que afecta a la calidad de los caracteres impresos es la impresión enfatizada. Haga estos cambios:

```
520 PRINT E$;"H";E$;"E";"Enfatizad
o"
540 PRINT E$;"H";E$;"F";"IMPRESION
EN COLOR"
```

Ejecute el programa y podrá ver la calidad de lo impreso con estas líneas.

Escape "E" activa la impresión enfatizada y Escape "F" la desactiva. Como puede ver la calidad es incluso mejor que la que producía la doble impresión. En modo enfatizado los caracteres se imprimen a mitad de la velocidad normal (40 caracteres por segundo), y la impresora golpea el papel con más fuerza. Si, este modo también desgasta demasiado la cinta pero produce una impresión de alta calidad.

Como ocurre con la doble impresión, la impresión enfatizada permanece hasta que se desactiva. La secuencia de códigos **E\$;** "H" de las anteriores líneas desactiva el modo de doble impresión.

El modo enfatizado es tan agradable que vamos a utilizarlo para añadir algunas líneas más:

```
280 PRINT E$;"E"
290 PRINT "BAJO PRECIO: No encontr
ara otro programa como este a su p
recio."
300 PRINT
310 PRINT "OPCION DE COLOR: Si, he
mos encontrado el secreto para com
binar codigos de soft-"
320 PRINT "ware que producen color
en el MX-80. Es nuestro genial pr
oceso patentado, la im-"
330 PRINT "presora convierte los p
untos negros en los colores primar
ios."
```

```
340 PRINT "Desgraciadamente, este
anuncio esta en blanco y negro, as
ique tendra que comprar"
360 PRINT
370 PRINT "COMPATIBILIDAD: Gracias
a nuestras tecnicas de programaci
on innovadoras, nuestro"
380 PRINT "software se ejecuta en
cualquier ordenador."
390 PRINT
410 PRINT E$;"F";E$;"G";"CARACTERI
STICAS DE IMPRESION"
430 PRINT E$;"H"
530 PRINT E$;"G";"Doble impresion
y enfatizado"
```

Tabulación horizontal

Una de las características que se encuentran en la **MX-80** es su capacidad para almacenar tabulaciones horizontales internas. Al principio esto puede parecer redundante para una impresora, tanto más cuando la mayoría de los **BASIC** tienen la función **TAB**. ¿Por qué no utilizarla?

Desafortunadamente, la función **TAB** de **BASIC** en la mayoría de los ordenadores afecta sólo a la pantalla, no a la impresora. Como resultado muchos usuarios no pueden tabular sino hasta la columna 40 o la 64 (el número de caracteres por línea de la pantalla). Esto no es adecuado para una impresora que puede imprimir 80 columnas (132 en el modo comprimido). Afortunadamente, **Epson** nos provee de la forma de solucionar estos inconvenientes de algunos ordenadores. Añada la siguiente línea al programa:

```
400 PRINT E$;"D";CHR$(39);CHR$(52)
;CHR$(0)
```

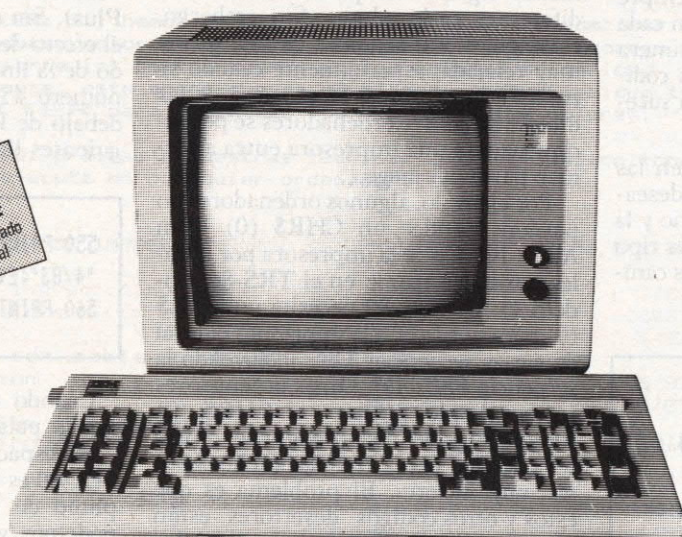
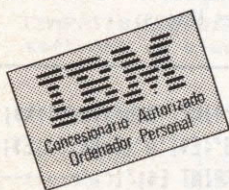
Esta línea establece las tabulaciones igual que en una máquina de escribir. Antes, en el programa, establecimos que **E\$** igual al código Escape (**CHR\$(27)**). El **ESC "D"** que los valores de tabulación horizontal le seguirán. El **CHR\$(39)** y **CHR\$(52)** sitúan las paradas de tabulación en las columnas 39 y 52. Pueden darse hasta 112 tabulaciones. **CHR\$(0)** se utiliza para indicar al ordenador que ha finalizado la secuencia de valores de tabulación.

Una vez que las paradas de tabulación se han situado, el código **ASCII HT (9 decimal)** puede activar la acción tabuladora en cualquier línea. Haga estos cambios y adiciones:

Y... ¿QUIÉN VA A EXPLICARME TODO ESTO?



ORANGE



Demostraciones y programas para
facturación, contabilidad y almacén.

E.I. ORANGE, S.A.

Concesionario autorizado
Ordenador Personal IBM

Príncipe de Vergara, 90 Teléf. 411 63 11 Madrid 6

Para más información sobre el Ordenador Personal IBM llame al
411 63 11 o envíe este cupón a c/. Príncipe de Vergara, 90 - Madrid-6

Nombre: _____
Empresa: _____ Dpto.: _____
Dirección: _____
Ciudad: _____ Teléf. _____


```
30 E$=CHR$(27) @ N$=CHR$(9)
410 PRINT E$;"F";E$;"G";"CARACTERI
STICAS DE IMPRESION";
420 PRINT N$;"PRECIO";N$;"FECHA DI
SPONIBILIDAD"
```

Ahora ejecute el programa

En la línea 420, el primer N\$ (recuerde que N\$ es CHR\$ [9]) causa una tabulación a la primera parada, columna 39. De la misma forma el segundo N\$ causa una tabulación a la columna 52. Ninguna tabulación más se puede hacer en la línea 420 por que sólo se han definido dos paradas.

Observe que la tabulación siempre empieza con la primera parada en cada línea. Si deseamos saltarnos la primera parada tenemos que mandar dos códigos de tabulación (CHR\$ [9]) en sucesión.

Las líneas 410 y 420 establecen las cabeceras para las columnas que deseamos imprimir. Añadimos el precio y la fecha de disponibilidad para cada tipo de impresión. Haga los siguientes cambios y adiciones:

```
430 PRINT E$;"H";
440 PRINT E$;"D";CHR$(40);CHR$(60)
;CHR$(0)
450 PRINT "MXWRITER con impresion
normal";N$;"$95";N$;"3/82"
460 PRINT D$;"Doble anchura";D2$;N
$;"$25";N$;"4/83"
510 PRINT C2$;E$;"G";"Doble impres
ion";N$;"$15";N$;"6/82"
520 PRINT E$;"H";E$;"E";"Enfatizad
o";N$;"$16";N$;"2/83"
530 PRINT E$;"G";"Doble impresion
y enfatizado";N$;" ND"
540 PRINT E$;"H";E$;"F";"IMPRESION
EN COLOR";N$;"$50";N$;"6/82"
```

Vaya comprobando la ejecución del programa con la figura 1 como hacemos habitualmente.

Se sitúan nuevas paradas de tabulación en la línea 440. Cada vez que se imprime un N\$ (CHR\$ [9]), la cabeza de impresión tabula a la siguiente parada. La implementación de estas tabulaciones horizontales en la MX-80 es una buena forma de burlar las limitaciones de las funciones TAB de algunos BASIC.

Códigos de software desertores

Alguno de ustedes estará moviendo la cabeza y preguntándose por qué las nuevas líneas no trabajan adecuadamente en su sistema. Antes de señalar con el dedo y acusar a la impresora, a este artículo o a otras fuentes inocentes, veamos algunos desagradables hechos. Primero y ante todo, no hay un BASIC estándar. Un grupo de gente se reunió hace unos años y decidieron un esquema de códigos estándar para los caracteres y símbolos utilizados en BASIC. Se llama Código Nacional Americano Estándar para el Intercambio de Información (*American National Standard Code for Information Interchange*), conocido como ASCII. Resultado de ello tenemos al menos algunas compatibilidades entre diferentes ordenadores. Sin embargo, estas guías son seguidas de una forma muy relajada, especialmente cuando se trata de impresoras. La mayoría de las diferencias entre ordenadores se potencian cuando una impresora entra a formar parte del juego.

Por ejemplo, algunos ordenadores no pueden mandar un CHR\$ (0) —un ASCII NUL— a la impresora por canales normales (como en el TRS-80 modelo I). La MX-80 necesita un CHR\$ (0) para notificar a la impresora el final de varias secuencias ESC, incluyendo la secuencia ESC "D". Otros ordenadores (como Apple) utilizan el CHR\$ (9) como código de inicialización de impresora, lo cual hace la tabulación horizontal muy incierta. El problema es que estos y otros códigos "desertores" están reservados para funciones especiales por el ordenador y por tanto chocan con los códigos que se necesitan para manejar la MX-80. Evidentemente este conflicto puede ser una constante irritación cuando se trabaja con la impresora.

¿Qué podemos hacer para aliviar este problema? La última solución es hurgar en el manual del ordenador y encontrar otras formas de obtener información del ordenador para mandar a la impresora. Esto requiere paciencia, pero es la forma más segura de obtener el pleno control de la impresora. Muchos sistemas le permiten mandar directamente la información a la impresora prescindiendo del intérprete BASIC y utilizando las instrucciones POKE o OUT. El manual del usuario del ROM Grafrax Plus de Epson da también algunas soluciones de ayuda para los ordenadores más populares.

Otra solución es evitar los códigos conflictivos cuando los encontramos. A menudo hay bastantes formas de obtener el mismo resultado. Por ejemplo, si situamos una parada del tabulador en 12

este causará una paginación indeseada, intentemos tabular a 10 u 11 y luego imprimamos un blanco o dos. Tómelo como un desafío a sus habilidades creadoras.

Como último recurso queda escribir los programas en lenguaje máquina. Sin embargo, a menos que usted sea un muy serio estudiante del arte de la programación, el tiempo que hay que invertir en aprender lenguaje ensamblador puede no merecer la pena.

Subrayado y espaciado de líneas variable

La MX-80 original no tiene modo de subrayado automático como las últimas versiones (Grafrax-80 y Grafrax Plus). Sin embargo, podemos obtener el efecto deseado cambiando el espaciado de la línea para que el guión (ASCII número 45) se imprima directamente debajo de los caracteres. Añada las siguientes líneas:

```
550 PRINT "Subrayado";N$;"$12";N$;
"4/83";E$;"A";CHR$(5);E$;"2"
560 PRINT E$;"1";"-----"
```

Cuando ejecute el programa podrá ver la palabra "Subrayado"

El espaciado de línea normal de la MX-80 es de 12/72 pulgadas (una amplitud de 12 puntos). Con ESC "A", podemos variar el espaciado de las líneas con incrementos de 1/72 pulgadas. En la línea 550, ESC "A" está seguido por CHR\$ (5) de tal forma que el salto es de 5 puntos en lugar de los 12 normales. Esta línea le dice a la impresora que esté preparada para hacer el cambio, pero no sucede nada hasta que nosotros no activemos el nuevo espaciado de línea con el código ESC "2". La línea 550 manda entonces todos los códigos necesarios para activar el nuevo espaciado de línea. Después de imprimir la línea 550, el papel salta 5 puntos para imprimir la secuencia de guiones debajo de la palabra "Subrayado".

Utilizando ESC "A", podemos variar el espaciado entre líneas desde una amplitud de 1 a 85 puntos. La MX-80 todavía nos provee de una forma rápida de situar el espaciado de líneas. Así ESC "1" cambia el espaciado de la línea a 7/72 pulgadas (equivalente a ESC "A" CHR\$ [7]). En la línea 560 esto causa un salto del papel de 7 puntos lo que le sitúa en su estado original. Observe que los dos saltos de línea consecutivos de 5 y 7 puntos suman un salto de 12 puntos.

STARFIRE
OPERATIONS
SOFTWARE, INC.

75 Long Street
New York, New York

1 de junio de 1.983

Norberto Gallego
Ordenador Popular
c/ Hermosilla 100
Madrid

Estimado Sr. Gallego:

Gracias por el interes demostrado en nuestro **MXWRITER**. Dejeme decirle algo mas en relacion a sus importantes caracteristicas.

BAJO PRECIO: No encontrara otro programa como este a su precio.

OPCION DE COLOR: Si, hemos encontrado el secreto para combinar codigos de software que producen color en el MX-80. Es nuestro genial proceso patentado, la impresora convierte los puntos negros en los colores primarios. Desgraciadamente, este anuncio esta en blanco y negro, asique tendra que comprar el **MXWRITER** para poder ver los deslumbrantes colores.

COMPATIBILIDAD: Gracias a nuestras tecnicas de programacion innovadoras, nuestro software se ejecuta en cualquier ordenador.

CARACTERISTICAS DE IMPRESION	PRECIO	FECHA DISPONIBILIDAD
MXWRITER con impresion normal	\$95	3/82
Doble anchura	\$25	4/83
Comprimido	\$12	5/82
Comprimido con doble anchura	ND	
Doble impresion	\$15	6/82
Enfatzado	\$16	2/83
Doble impresion y enfatzado	ND	
IMPRESION EN COLOR	\$50	6/82
Subrayado	\$12	4/83
Super indices	\$15	5/82
Sub indices	\$23	3/82
Variable espaciamiento de linea	\$18	9/82

ELOGIOS DE LOS CRITICOS:

Utilize MXWRITER para preparar todos mis discursos y correspondencia personal.
... Maria Jesus Aguado, Profesora.

Disfruto con los graficos. Los colores que creo con el MXWRITER son fantasticos!
... Ricardo Garcia, experto en graficos por ordenador.

SOPORTE:

Tenemos los tecnicos mas cualificados del ramo. Pregunte por Juan Lopez. El esta siempre disponible para ayudarle con sus problemas en software.

RESPONSABILIDADES:

Starfire ofrece este programa en perfectas condiciones y no asume ninguna responsabilidad en los danos causados como resultado del uso del programa.

GARANTIA:

MXWRITER tiene una garantia de tres anos, durante los cuales reemplazaremos los diskettes danados totalmente gratis.

ATRACCIONES FUTURAS:

Preste atencion a nuestro proximo software: graficos para su EPSON.

James T. Smith, Director ejecutivo

Superíndices

Con una técnica similar la **MX-80** puede también imprimir subíndices y superíndices. Los superíndices se utilizan para imprimir exponentes en las ecuaciones matemáticas o para marcar notas a pie de página en un texto. Los subíndices se utilizan para etiquetar variables en las fórmulas científicas. Añada y ejecute estas pocas líneas:

```
570 PRINT C$;"Super";E$;"A";CHR$(3);E$;"2"
580 PRINT TAB(3);C2$;"indices";N$;"$15";N$;"5/82";E$;"A";CHR$(140);E$;"2"
```

Fácil, ¿verdad? Utilizando unos pequeños trucos como cambiar al modo comprimido y variar el salto del carro, obtenemos buenos efectos. Los códigos para activar y desactivar el modo comprimido están almacenados en C\$ y C2\$ respectivamente.

La línea 570 cambia el espaciado de la línea a 3 puntos, y el final de la línea 580 lo vuelve a colocar en 12, que es su posición normal. (Utilizamos 140 porque el 12 es otro de esos códigos desértos. Algunas veces añadiendo 128 a un código podemos pasar el intérprete BASIC sin problemas).

Subíndices

Podemos también imprimir subíndices variando el salto del carro y utilizando opcionalmente el modo comprimido. Añada estas líneas:

```
590 PRINT TAB(3);"indices";N$;"$23";N$;"3/82";E$;"A";CHR$(3);E$;"2"
600 PRINT C$;TAB(3);"Sub";E$;"A";CHR$(140);E$;"2" @ PRINT C2$;
```

Más cerca del espaciamento de línea variable

Hemos visto cómo subrayar y cómo crear subíndices y superíndices cambiando el espaciamento de la línea tal y como lo necesitásemos. Añada las siguientes cuatro líneas que traten el mismo tema:

```
610 PRINT TAB(4);"ia";E$;"A";CHR$(3);E$;"2"
```

```
10 C$=CHR$(15) @ C2$=CHR$(18)
20 D$=CHR$(14) @ D2$=CHR$(20)
30 E$=CHR$(27) @ N$=CHR$(9)
40 Z$=CHR$(42) @ B$=CHR$(42)
50 PRINT "S T A R F I R E"
60 PRINT TAB(4);"O P E R A T I O N S"
70 PRINT TAB(8);"S O F T W A R E, I N C ."
130 PRINT C$
140 PRINT TAB(14);D$;"75 Long Street"
150 PRINT TAB(14);D$;"New York, New York"
160 PRINT C2$
170 PRINT "1 de junio de 1.983"
180 PRINT
190 PRINT "Norberto Gallego"
200 PRINT "Ordenador Popular"
210 PRINT "c/ Hermosilla 100"
220 PRINT "Madrid"
230 PRINT
240 PRINT "Estimado Sr. Gallego:"
250 PRINT
260 PRINT "Gracias por el interes demostrado en nuestro";D$;"MXWRITER.";D2$;" Dejeme decirle"
270 PRINT "algo mas en relacion a sus importantes características."
280 PRINT E$;"E"
290 PRINT "BAJO PRECIO: No encontrara otro programa como este a su precio."
300 PRINT
310 PRINT "OPCION DE COLOR: Si, hemos encontrado el secreto para combinar codigos de soft-"
320 PRINT "ware que producen color en el MX-80. Es nuestro genial proceso patentado, la im-"
330 PRINT "presora convierte los puntos negros en los colores primarios."
340 PRINT "Desgraciadamente, este anuncio esta en blanco y negro, asique tendra que comprar"
350 PRINT "el ";D$;"MXWRITER";D2$;" para poder ver los deslumbrantes colores."
360 PRINT
370 PRINT "COMPATIBILIDAD: Gracias a nuestras tecnicas de programacion innovadoras, nuestro"
380 PRINT "software se ejecuta en cualquier ordenador."
390 PRINT
400 PRINT E$;"D";CHR$(39);CHR$(52);CHR$(0)
410 PRINT E$;"F";E$;"G";"CARACTERISTICAS DE IMPRESION";
420 PRINT N$;"PRECIO";N$;"FECHA DISPONIBILIDAD"
430 PRINT E$;"H";
440 PRINT E$;"D";CHR$(40);CHR$(60);CHR$(0)
450 PRINT "MXWRITER con impresion normal";N$;"$95";N$;"3/82"
460 PRINT D$;"Doble anchura";D2$;N$;"$25";N$;"4/83"
470 PRINT C$;"Comprimido";E$;"A";CHR$(1);E$;"2"
480 PRINT C2$;N$;"$12";N$;"5/82";E$;"A";CHR$(11);E$;"2"
490 PRINT C$;D$;"Comprimido con doble anchura";E$;"A";CHR$(1);E$;"2"
500 PRINT C2$;D2$;N$;" ND";E$;"A";CHR$(140);E$;"2"
510 PRINT C2$;E$;"G";"Doble impresion";N$;"$15";N$;"6/82"
```



```

520 PRINT E$;"H";E$;"E";"Enfatisado";N$;"$16";N$;"2/83"
530 PRINT E$;"G";"Doble impresion y enfatisado";N$;" N
D"
540 PRINT E$;"H";E$;"F";"IMPRESION EN COLOR";N$;"$50";
N$;"6/82"
550 PRINT "Subrayado";N$;"$12";N$;"4/83";E$;"A";CHR$(5
);E$;"2"
560 PRINT E$;"I";"-----"
570 PRINT C$;"Super";E$;"A";CHR$(3);E$;"2"
580 PRINT TAB(3);C2$;"indices";N$;"$15";N$;"5/82";E$;"
A";CHR$(140);E$;"2"
590 PRINT TAB(3);"indices";N$;"$23";N$;"3/82";E$;"A";C
HR$(3);E$;"2"
600 PRINT C$;TAB(3);"Sub";E$;"A";CHR$(140);E$;"2" @ PR
INT C2$;
610 PRINT TAB(4);"ia";E$;"A";CHR$(3);E$;"2"
620 PRINT TAB(2);"r";TAB(5);"b";" espaciamento de lin
ea";N$;"$18";N$;"9/82"
630 PRINT TAB(1);"a";TAB(6);"l"
640 PRINT "v";TAB(7);"e";E$;"A";CHR$(140);E$;"2"
800 A$="ELOGIOS DE LOS CRITICOS:" @ GOSUB 1170
810 PRINT "Utilizo MXWRITER para preparar todos mis di
scursos y correspondencia"
820 PRINT "personal.
.. Maria Jesus Aguado, Profesora."
830 PRINT
840 PRINT "Disfruto con los graficos. Los colores que
creo con el MXWRITER son"
850 PRINT "fantasticos! ... Ricardo Garcia, e
xperto en graficos por ordenador."
860 PRINT
870 A$="SOPORTE:" @ GOSUB 1170
880 PRINT "Tenemos los tecnicos mas cualificados del r
amo. Pregunte por Juan"
890 PRINT E$;"A";CHR$(5);E$;"2";
900 PRINT "Lopez. El esta siempre disponible para ayud
arle con sus problemas en software."
910 PRINT TAB(15);E$;"I";"-----"
920 PRINT E$;"A";CHR$(140);E$;"2"
925 FOR I=1 TO 7 @ PRINT @ NEXT I
930 A$="RESPONSABILIDADES:" @ GOSUB 1170
940 PRINT C$;"Starfire ofrece este programa en perfect
as condiciones y no asume ninguna ";
950 PRINT "responsabilidad en los danos causados como
resultado del uso del programa."
960 PRINT
970 A$="GARANTIA:" @ GOSUB 1170
980 PRINT "MXWRITER tiene una garantia de tres anos, d
urante los cuales reemplazaremos"
990 PRINT "los diskettes danados totalmente gratis."
1000 PRINT
1010 A$="ATRACCIONES FUTURAS:" @ GOSUB 1170
1020 PRINT "Preste atencion a nuestro proximo software
: graficos para su EPSON."
1030 PRINT @ PRINT N$;"James T. Smith, Director ejecut
ivo" @ PRINT
1160 END
1170 PRINT C$;D$;A$
1180 PRINT C2$
1190 RETURN

```

```

620 PRINT TAB(2);"r";TAB(5);"b";"
espaciamento de linea";N$;"$18";N
$;"9/82"
630 PRINT TAB(1);"a";TAB(6);"l"
640 PRINT "v";TAB(7);"e";E$;"A";CH
R$(140);E$;"2"

```

Mezclando el modo normal y el comprimido

Vamos a ver qué ocurre cuando añadimos el precio y la fecha de disponibilidad a nuestra tabla de características de impresión. Ejecute el programa después de hacer las siguientes adiciones y cambios:

```

470 PRINT C$;"Comprimido";
480 PRINT C2$;N$;"$12";N$;"5/82"
490 PRINT C$;D$;"Comprimido con do
ble anchura";
500 PRINT C2$;D2$;N$;" ND"

```

Parece que aquí tenemos pocos problemas. En primer lugar, ¿qué es lo que pasa con la impresión comprimida anunciada en las líneas 470 y 490? (Recuerde que C\$ activa el modo comprimido y C2\$ lo desactiva). Como sabemos, no podemos combinar el modo normal y el comprimido en la misma línea. Si se envían varios códigos de amplitud de impresión a la impresora el último determinará la amplitud de la línea completa. Las líneas 470 y 480 ilustran este punto. La impresora recibe un C\$ seguido por un C2\$. El último código enviado es C2\$ con lo cual la línea entera se imprime con anchura normal. Esta es una limitación que hemos de conocer para trabajar teniéndola en cuenta.

Impresión comprimida versus tabulación horizontal

Nos encontramos con otro problema si eliminamos simplemente los códigos C2\$. Nuestra tabulación horizontal está situada para la impresión normal. Si la línea se imprime en el modo comprimido, las tabulaciones no coincidirán con las de las otras características de impresión. Así el desafío es imprimir las dos líneas en modo comprimido manteniendo las tabulaciones anteriores.

El espaciado de línea variable viene de nuevo a rescatarnos. La impresora no nos permitirá mezclar el modo compri-

mido y el normal, pero con un pequeño truco podemos hacer que parezca que están en la misma línea. Cambie estas líneas:

```
470 PRINT C$;"Comprimido";E$;"A";CHR$(1);E$;"2"
480 PRINT C2$;N$;"$12";N$;"5/82";E$;"A";CHR$(11);E$;"2"
490 PRINT C$;D$;"Comprimido con doble anchura";E$;"A";CHR$(1);E$;"2"

500 PRINT C2$;D2$;N$;" ND";E$;"A";CHR$(140);E$;"2"
```

Cuando ejecute el programa la salida debe parecerse a la de la figura 1.

En la línea 470, añadimos el código para que se dé un salto del carro de un punto. Esto nos permite imprimir "Comprimido" en el modo comprimido y luego el carro salta un punto. La línea 480 cancela el modo comprimido, luego tabula a nuestras posiciones predefinidas e imprime \$12 en la columna del precio y 5/82 en la columna de fecha de disponibilidad, todo con impresión normal. El pequeño salto de carro da la impresión de que todo está en la misma línea. La misma técnica se utiliza en las líneas 490 y 500. Al final de la línea 500 volvemos a poner la impresora en su salto de carro original de 12 puntos.

Una subrutina útil

Las siguientes líneas nos proveen de una subrutina que nos ahorra teclear durante el resto del programa.

```
1170 PRINT C$;D$;A$
1180 PRINT C2$
1190 RETURN
```

Esta subrutina se utiliza para imprimir un *string* (almacenado en A\$) en los modos comprimido y de doble anchura. Las siguientes líneas añaden testimonio a la utilización de la subrutina:

```
800 A$="ELOGIOS DE LOS CRITICOS:"
@ GOSUB 1170
810 PRINT "Utilizo MXWRITER para reparar todos mis discursos y correspondencia"
820 PRINT "personal."
```

```
... Maria Jesus
Aguado, Profesora."
830 PRINT
840 PRINT "Disfruto con los graficos. Los colores que creo con el MX WRITER son"
850 PRINT "fantasticos!"
.. Ricardo Garcia, experto en graficos por ordenador."
```

Como siempre, ejecute el programa y compárelo con la figura 1. La línea 800 utiliza nuestra subrutina, lo que causa que el *string* almacenado en A\$ (es decir, "ELOGIOS DE LOS CRITICOS") sea impreso en el modo comprimido y en el modo de doble anchura.

La letra pequeña

El modo comprimido es ideal para imprimir en esa letra pequeña que a nadie le gusta leer. Añada y ejecute estas líneas:

```
930 A$="RESPONSABILIDADES:" @ GOSUB 1170
940 PRINT C$;"Starfire ofrece este programa en perfectas condiciones y no asume ninguna ";
950 PRINT "responsabilidad en los danos causados como resultado del uso del programa."
960 PRINT C2$
```

Sus resultados han de coincidir con los de la figura habitual. Las líneas 940 y 950 se combinan para crear una sola línea de impresión en modo comprimido.

Los toques finales

Añada estas líneas a nuestra carta:

```
860 PRINT
870 A$="SOPORTE:" @ GOSUB 1170
880 PRINT "Tenemos los tecnicos mas cualificados del ramo. Pregunte por Juan"
890 PRINT E$;"A";CHR$(5);E$;"2";
900 PRINT "Lopez. El esta siempre disponible para ayudarle con sus problemas en software."
910 PRINT TAB(15);E$;"1";"-----"
```

```
920 PRINT E$;"A";CHR$(140);E$;"2"
925 FOR I=1 TO 7 @ PRINT @ NEXT I
960 PRINT
970 A$="GARANTIA:" @ GOSUB 1170
980 PRINT "MXWRITER tiene una garantia de tres anos, durante los cuales reemplazaremos"
990 PRINT "los diskettes danados totalmente gratis."
1000 PRINT
1010 A$="ATRACCIONES FUTURAS:" @ GOSUB 1170
1020 PRINT "Preste atencion a nuestro proximo software: graficos para su EPSON."
1030 PRINT @ PRINT N$;"James T. Smith, Director ejecutivo" @ PRINT
1160 END
```

Nuestra amiga la figura 1 muestra la carta completa. Para su referencia, el programa completo se imprime en la figura 2.

Las últimas adiciones ayudan a rellenar la carta. Observe que las líneas 890, 910 y 920 cambian temporalmente el salto de carro para que se pueda subrayar la palabra "siempre". Comenzamos en la línea 890 con un salto de carro de 5 puntos. En la línea 910 se sitúa en 7, luego tabula a 15 e imprime la línea de guiones debajo de "siempre". La línea 920 retorna el salto de carro a la posición normal (Nota: se utiliza 140 en lugar de 12 para evitar una paginación indeseada).

Para imprimir más de una página, es necesario paginar para evitar imprimir encima de la perforación entre páginas. La forma de solucionar el problema es enviar un código de paginación (12 decimal). Hemos elegido insertar una línea temporal (925) para forzar a hacer varios saltos de carro. Esta línea la eliminaremos en el siguiente artículo.

El segundo y último artículo de esta serie nos conducirá al aprendizaje de la realización de caracteres gráficos con la MX-80. Utilizaremos caracteres bloque de gráficos para crear un logotipo y algunas letras sobre dimensionadas, así como algunos efectos visuales.

David Kater
© Popular Computing/Ordenador Popular

suplemento
BYTE



GUIA PARA EL PRINCIPIANTE EN LOGO

En los años 60, los ordenadores eran muy caros y tenían una cantidad muy limitada de memoria. Un ordenador como el IBM 1620 podía almacenar un máximo de 24 Kbytes (ó 60.000 dígitos decimales). Incluso los ordenadores más grandes dedicados a la investigación podían manejar como mucho 6 veces esa cantidad. Al estar los programas obligados a hacer un uso restrictivo de la memoria, los lenguajes de ordenador estaban diseñados teniendo en cuenta esta limitación.

Los lenguajes debían ser sencillos para el ordenador, aunque fuesen muy latosos para el programador. Por ejemplo, para ayudar al compilador a controlar adecuadamente la ocupación de memoria, muchos lenguajes de programación insistían en la relación muy próxima entre los nombres utilizados por el programa y las celdas de la memoria del ordenador. Como consecuencia, los únicos datos que podían ser denominados directamente y manipulados en las operaciones del programa eran aquellos que podían almacenarse en una celda individual. El ta-

maño de las estructuras tenía que especificarse necesariamente en tiempo de compilación. Muchos lenguajes necesitaban también que el programador incluyera las instrucciones "declarativas" y aceptase otras restricciones en el uso de nombres para hacer sencillo al compilador la determinación del tipo de almacenamiento que requería cada variable. (Por ejemplo, en algunos lenguajes era preciso que ciertos nombres comenzaran con I o J cuando se trataba de enteros; que las matrices se tuvieran que declarar junto con su tamaño, o que las funciones definidas tuviesen nombres que comenzaran con FN, seguido de un dígito).

El interés por ahorrar memoria impregnaba no sólo el lenguaje, sino todo el ordenador como conjunto. Así por ejemplo, si el sistema incluía un editor de programas, corregir una línea de código requería que el operador abortase el programa, cargara el editor, leyera un fichero, realizase la edición, escribiese un nuevo fichero, saliese el editor, recompilara de nuevo, y volviera a cargar el programa. Todo esto por- que el editor y el lenguaje no

cabían simultáneamente en la memoria principal.

Los lenguajes de los años 60 se desarrollaron con los ordenadores personales de los 70, que, aunque ya no tan caros, todavía no tenían demasiada memoria. A medida que los ordenadores personales se convirtieron en algo cada vez más popular, la gente comenzó a confundir la idea de que un lenguaje que es sencillo para un ordenador sería también sencillo para el usuario. ("BASIC tiene sólo unas pocas funciones primitivas; por consiguiente debe ser sencillo de aprender"). Algunas personas llegaron incluso a racionalizar los aspectos molestos e incómodos de tales lenguajes hasta el punto de considerarlos ventajas. ("Al tener que declarar los tipos de datos de las variables puedes organizar mejor tus programas"). "Si fuera demasiado sencillo editar los programas no los escribirías con cuidado la primera vez". Por otra parte cuando los educadores empezaron a explorar los usos potenciales de los ordenadores, frecuentemente aceptaron los inconvenientes de estos lenguajes como parte integral de la programación.

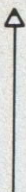
Durante los últimos 12 años, el Grupo Logo del MIT (*Massachusetts Institute of Technology*) bajo la dirección de Seymour Papert, junto con algunos colegas de universidades y centros de investigación de todo el mundo, adoptó una postura diferente de cara a la enseñanza del uso de los ordenadores. En lugar de aceptar las limitaciones de los equipos normales por aquellos días, comenzamos a trabajar con los ordenadores más grandes de que se disponía para investigación. El sistema que utilizamos, llamado Logo, es esencialmente un dialecto de LISP, un potente lenguaje desarrollado para investigar sobre inteligencia artificial, que utilizaba una gran cantidad de memoria comparando con lo que era usual en los años 60.

Trabajando con Logo descubrimos algunas cosas importantes. Un lenguaje de ordenador puede ser al mismo tiempo sencillo y potente. De hecho estos dos aspectos son complementarios más que conflictivos, ya que es la ausencia casi total de posibilidades expresivas en los lenguajes primitivos como BASIC, la que hace difícil

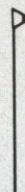
Turtle starts



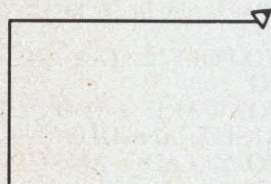
FORWARD 100



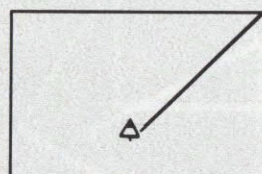
RIGHT 90



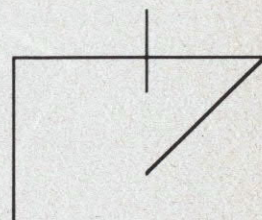
FORWARD 150
LEFT 45



BACK 100
LEFT 45



PENUP
FORWARD 50
PENDOWN
FORWARD 50
HIDETURTLE



que los principiantes escriban programas sencillos que realicen algo interesante. Más importante aún, encontramos que es posible dar al usuario un control sobre potentes recursos de cálculo, que pueden ser utilizados como medios para aprender, jugar y explorar. Esto nos ha llevado en ocasiones más allá de las consideraciones ordinarias sobre diseño de lenguajes de ordenador, para entrar en el terreno de la creación de imágenes seductoras que expresaran cómo los ordenadores pueden aportar una perspectiva nueva para reformular las ideas tradicionales de las ciencias y de las matemáticas, de manera que las hagan más accesibles, más de acuerdo con los modos intuitivos del pensamiento. En nuestra investigación en el MIT, trabajando con preescolar, educación básica, bachilleres, técnicos y universitarios, y con sus profesores, utilizamos Logo para introducir la programación y la perspectiva informática a todos los niveles. En este artículo, me gustaría mostrar al lector en qué consiste la programación utilizando Logo, ese sencillo pero potente sistema, que le

permitirá explorar el mundo del ordenador.

Dibujar con la tortuga

Vamos a empezar con algo que llamaremos gráficos con la tortuga. La tortuga es un pequeño puntero triangular en la pantalla que responde a unas órdenes muy sencillas. FORWARD mueve la tortuga hacia adelante un número determinado de unidades. Si usted escribe, por ejemplo, la orden Logo FORWARD 50, la tortuga responderá avanzando hacia adelante 50 pasos de tortuga (aproximadamente 1/4 de la altura de la pantalla). RIGHT hace que la tortuga gire en el sentido horario un determinado número de grados. BACK y LEFT hacen los movimientos opuestos de FORWARD y RIGHT. La tortuga lleva también una pluma que, cuando se ha bajado, va dejando una traza de su recorrido por la pantalla cuando se mueve y el lápiz está bajo. Las órdenes PENUP y PENDOWN hacen que la tortuga levante y baje la pluma. La figura 1 muestra el resultado de una sencilla secuencia de órdenes Logo.

Resulta muy entretenido hacer dibujos utilizando estas instrucciones (junto con otras pocas más como CLEARSCREEN, que limpia la pantalla). Pero para hacer progresos reales es preciso enseñar al ordenador nuevos términos. Por ejemplo, usted puede enseñar al ordenador a que la tortuga dibuje un cuadrado haciéndole que repita 4 veces la secuencia de ir hacia adelante 50 pasos y girar 90 grados. Las instrucciones de Logo serían:

```
TO CUADRADO
REPEAT 4 (FORWARD 50
RIGHT 90)
END
```

CUADRADO es un ejemplo de procedimiento Logo. La primera línea (señalada por el término TO) especifica el nombre del procedimiento. A este procedimiento le hemos dado el nombre CUADRADO (porque dibuja precisamente un cuadrado), pero podría haberse llamado con cualquier otro nombre. El resto del procedimiento especifica la lista de instrucciones que el ordenador llevará a cabo en respuesta a la orden CUADRA-

Figura 1: Desplazamiento de la tortuga con una serie de sencillas órdenes en Logo. FORWARD avanza la tortuga hacia adelante. RIGHT y LEFT la hacen girar. PENUP y PENDOWN se emplean para subir o bajar la pluma virtual (con la pluma bajada, la tortuga va dejando el rastro de su movimiento).

DO; la palabra END indica el final de la definición.

Una vez definido así, CUADRADO entra a formar parte del vocabulario de nuestro ordenador. Cada vez que se le dé orden CUADRADO la tortuga dibujará un cuadrado.

Procedimientos con entradas

Entre CUADRADO y FORWARD existe una importante diferencia. CUADRADO siempre dibuja un cuadrado de 50 pasos de lado. Pero FORWARD es más versátil; acepta una entrada que determina el tamaño del desplazamiento de la tortuga. Es posible cambiar el procedimiento CUADRADO de manera que también acepte una entrada a parámetro que determine el tamaño del lado del cuadrado que va a dibujar. Así por ejemplo:

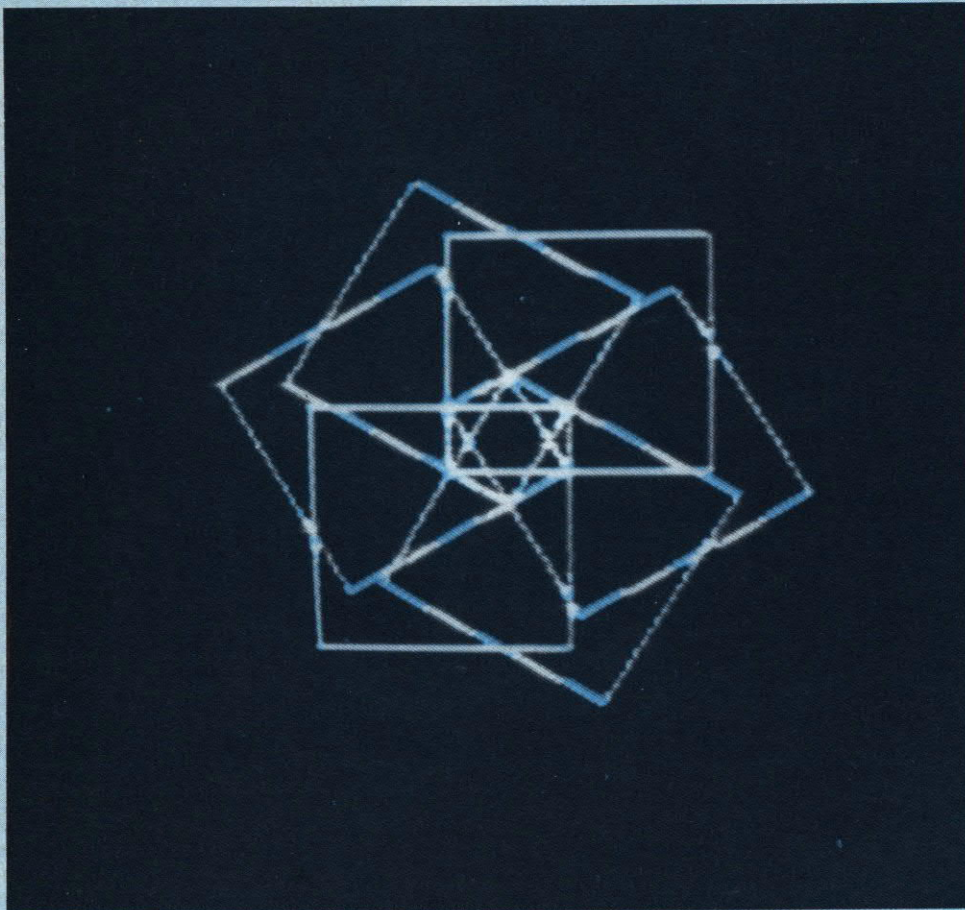


Foto 1: Dibujo creado por un sencillo programa Logo de sólo una línea que hace que la tortuga repita 6 veces estos pasos: Hacia adelante 20 unidades (FORWARD 20), girar 60 grados a la derecha (RIGHT 60) y dibujar un cuadrado de 75 unidades de lado.

TO CUADRADO: LADO
REPEAT 4 (FORWARD:
LADO RIGHT 90)
END

A partir de este momento usted utilizará CUADRADO de la misma manera que cualquier orden Logo que precisa una entrada. Así para dibujar un cuadrado de 100 pasos de lado teclearíamos:

CUADRADO 100

Para dibujar un cuadrado de 50 pasos de lado introduciríamos:

CUADRADO 50

La definición de CUADRADO ilustra una regla general para definir procedimientos que precisan entradas. Se elige un nombre para la entrada y se incluye este nombre en la línea de títulos del procedimiento precedida por dos puntos. En el cuerpo del procedimiento se pone el

nombre de la entrada (con los dos puntos) allí donde normalmente se utilizaría el valor de la entrada.

Puesto que un procedimiento, una vez definido, se convierte en otra palabra que el ordenador "conoce", es posible utilizar estos procedimientos como parte de las definiciones de nuevos procedimientos. He aquí un procedimiento que realiza un dibujo repitiendo la secuencia de avanzar primero, girar, y dibujar un cuadrado (ver foto 1):

TO DESIGN
REPEAT 6 (FORWARD 20
RIGHT 60 CUADRADO
75)
END

Procedimientos Recursivos Simples

El siguiente procedimiento también dibuja un cuadrado de un tamaño determinado:

TO CUAD: LADO
FORWARD: LADO
RIGHT 90
CUAD: LADO
END

Aunque CUAD y CUADRADO dibujan ambos cuadrados, se comportan de manera muy distinta.

En lugar de dibujar un cuadrado y detenerse, CUAD hace que la tortuga vuelva a recorrer el mismo camino una y otra vez hasta que usted diga al ordenador que se pare. Veamos por qué sucede esto. Cuando usted da la orden:

CUAD 100

la tortuga debe avanzar FORWARD 100, girar RIGHT 90, y luego realizar de nuevo CUAD 100, y así sucesivamente.

Añadiendo una segunda entrada a CUAD se obtiene un procedimiento que llamaremos POLY, que repite una y otra vez esta secuencia:

avanzar una distancia fija y girar un determinado ángulo. El procedimiento acepta como entradas el número de pasos que ha de avanzar y el ángulo que ha de girar:

TO POLY: LADO: ANGULO
FORWARD: LADO
RIGHT: ANGULO
POLY: LADO: ANGULO
END

Para utilizar el Procedimiento POLY, escriba la palabra POLY seguido de los valores específicos para las entradas:

POLY 60 144

La figura 2 muestra algunos de los muy diversos diseños que pueden obtenerse llamando al procedimiento POLY con diversas entradas.

Recursión es un término de programación que describe la posibilidad de utilizar un procedimiento como parte de su propia definición. CUAD y POLY son procedimientos recursivos de un tipo muy sencillo —simplemente repiten un ciclo fijo una y otra vez. Pero la recursión es una idea mucho más potente y puede utilizarse para obtener efectos mucho más complicados. Examine el siguiente ejemplo que va un poco más allá del tipo de recursión puramente repetitiva:

TO POLYSPI: LADO: ANGULO
FORWARD: LADO
RIGHT: ANGULO
POLYSPI: LADO + 3: ANGULO
END

La instrucción:

POLYSPI 1 120

da lugar a esta secuencia de movimientos de la tortuga

Figura 2: Todas estas formas se consiguen con el programa Logo de 3 líneas POLY, que hace avanzar a la tortuga cierta distancia (FORWARD), gira a la derecha un ángulo fijo (RIGHT), y repite una y otra vez este proceso. Las figuras dibujadas por POLY siempre son cerradas, pero el número de lados que hay que trazar antes de cerrar la figura depende del parámetro o entrada ANGULO del procedimiento.

```
FORWARD 1
RIGHT 120
FORWARD 4
RIGHT 120
FORWARD 7
RIGHT 120
FORWARD 10
RIGHT 120
```

que produce una espiral triangular en la que cada uno de los lados es 3 pasos mayor que el anterior. La figura 3 muestra algunas de las formas que pueden generarse con el procedimiento POLYSPI por una instrucción que dibuje un cuadrado:

```
TO CUADRESPI: LADO:
ANGULO
CUADRADO: LADO
RIGHT: ANGULO
CUADRESPI: LADO + 3:
ANGULO
END
```

El resultado de la ejecución de

```
CUADRESPI 1 10
```

aparece en la figura 4 y como puede verse es una secuencia de cuadrados de tamaño creciente que comienza con uno de 1 paso de lado. Cada cuadrado es 3 unidades mayor que el anterior y girado con respecto a él 10 grados. El procedimiento continúa ejecutándose y los cuadrados continúan creciendo hasta que usted le diga a Logo que se pare. El procedimiento puede modificarse de manera que se detenga cuando los cuadrados alcancen un determinado tamaño (por ejemplo 100 pasos), incorporando un criterio de parada:

```
TO CUADRESPI: LADO:
ANGULO
IF: LADO > 100 THEN
STOP
CUADRADO: LADO
RIGHT: ANGULO
CUADRESPI: LADO + 3:
ANGULO
END
```

Parte de la potencia de la recursión reside en el hecho de que programas tan simples como los mostrados pueden llevar a resultados tan variados.

Un Entorno para Explorar

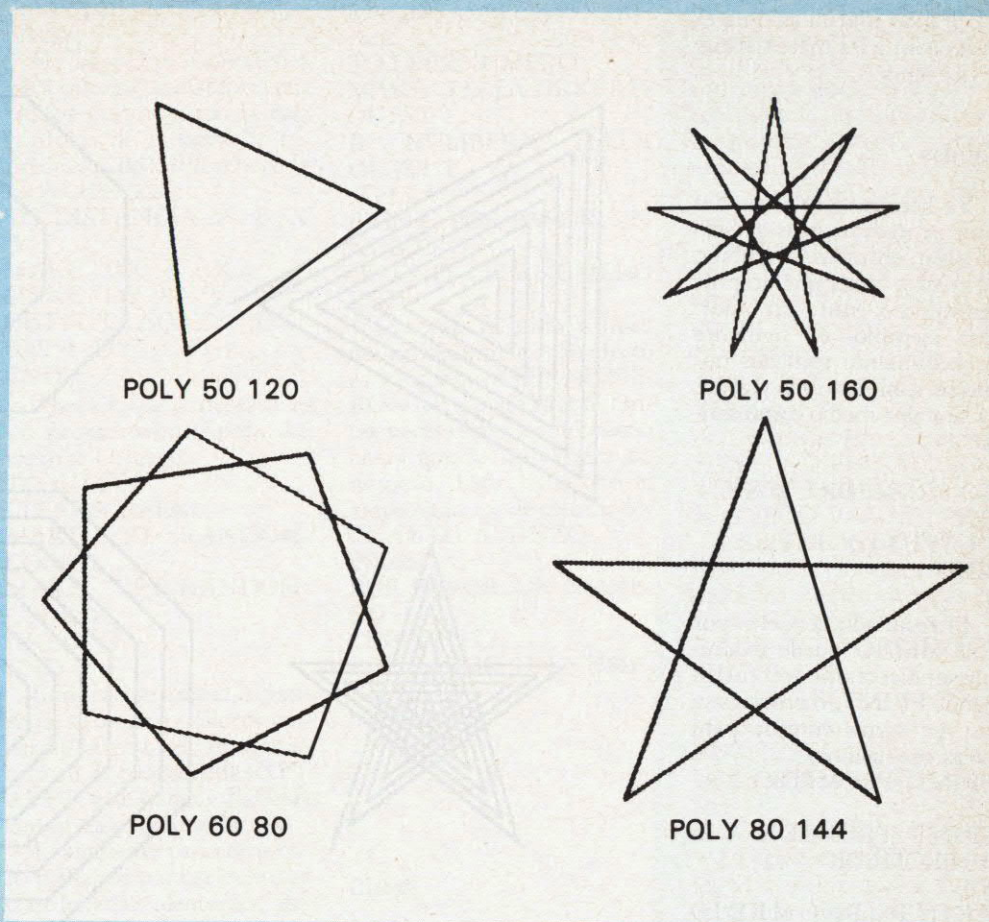
Como puede ver por los ejemplos presentados hasta ahora, es realmente muy sencillo comenzar a programar con los gráficos de la tortuga. Ello es debido en parte al tema en sí. Las órdenes básicas tienen efectos sencillos y visibles. Al mismo tiempo los gráficos con la tortuga son un área increíblemente rica para la exploración en la cual, incluso programas sencillos, pueden tener resultados a primera vista inesperados, y a veces muy bonitos. La pequeña cantidad de Logo que hemos visto hasta aquí es suficiente para soportar semanas de actividades en programación y matemáticas explorando

cuestiones del tipo de "¿cómo depende la figura generada por POLY del ángulo de entrada?" o "¿por qué programas tan repetitivos producen dibujos simétricos?" o simplemente dedicándose a crear bellas configuraciones geométricas. Andrea diSessa y yo explicamos las matemáticas que aparecen al investigar este acercamiento a la geometría basado en el ordenador, en el libro *Turtle Geometry: The computer as a medium for Exploring Mathematics* (Cambridge, MA, MIT Press, 1981).

Aparte de la materia misma, la interacción del sistema juega también un papel crucial. Cuando la gente investiga utilizando Logo, está definiendo continuamente nuevos procedimientos y modificando los antiguos. Un sistema típico orientado a un compilador, en el que cambiar una definición requiere andar siempre entrando y saliendo de editores, compiladores. Gran par-

te del esfuerzo en la creación de Logo ha consistido en proporcionar un entorno de programación que agilice la definición y modificación de los procedimientos. Las implementaciones de Texas Instruments y Apple incluyen editores de pantalla integrados. Al escribir el comando TO o EDIT se activa el editor de pantalla, con la definición de procedimiento apropiada lista para ser modificada. Con solo pulsar una tecla la nueva definición queda incluida con un procedimiento Logo.

El éxito de la geometría de tortuga es debida en gran parte al hecho de que al diseñarlo no pensábamos en nosotros aisladamente como matemáticos y educadores intentando inventar una nueva forma de explicar la geometría, ni como científicos del ordenador intentando crear un sistema. Por el contrario, lo que intentamos era contemplar simultáneamente ambas perspectivas,



adaptando continuamente el ordenador a las matemáticas y viceversa.

Salidas

Ya hemos visto cómo definir procedimientos que requieren entradas. También es posible hacer que un procedimiento emita un valor. Por ejemplo, el siguiente procedimiento tomados números como entrada y genera su valor medio como salida:

```
TO PROMEDIO: X XY
:XY
OUTPUT (:X + :Y)/2
END
```

El resultado devuelto por PROMEDIO puede examinarse directamente (utilizando PRINT) o utilizarse a su vez como entrada para otras operaciones:

```
PRINT (PROMEDIO 2 3)
2,5
PRINT (PROMEDIO 1 2)
+ (PROMEDIO 3 4)
5,0
PRINT (PROMEDIO
PROMEDIO 1 2) 3)
2,25
```

Observe el convenio utilizado por Logo de utilizar paréntesis para agrupar un procedimiento con sus entradas. Aunque los paréntesis son casi siempre opcionales en líneas individuales de Logo es una buena idea incluirlos porque facilitan la lectura.

Programación con Procedimientos

Un programa Logo típico consiste en un grupo de procedimientos. Estos procedimientos se pasan información entre ellos por medio de las entradas y las salidas. La ventaja de este tipo de organización es que divide el programa en trozos manejables, cada uno de ellos sencillo en sí mismo. Incluso en un programa complejo, es poco corriente tener un procedimiento individual que tenga muchas líneas. Además, el editor integrado de Logo y la

naturaleza interactiva general del sistema, permite definir y poner a prueba separadamente cada procedimiento individual.

Para ilustrar la organización en procedimientos a que se prestan los programas Logo vamos a diseñar un sencillo juego cuyas reglas se explican a continuación. El ordenador elige un "punto misterioso" aleatorio de la pantalla y se trata de que el jugador llegue a él con sucesivos movimientos LEFT y FORWARD con la tortuga. Antes de cada movimiento, el ordenador imprime la distancia de la tortuga al punto misterioso. El éxito consiste en dejar la tortuga muy cerca del punto misterioso en el menor número de movimientos posibles. He aquí una transcripción del juego en acción.

DISTANCIA AL PUNTO
67,6

CUANTO GIRO A LA IZ-
QUIERDA?

0
CUANTO AVANZO?

25
DISTANCIA AL PUNTO
90,25

CUANTO GIRO A LA IZ-
QUIERDA?

180
CUANTO AVANZO?

50
DISTANCIA AL PUNTO
47,38

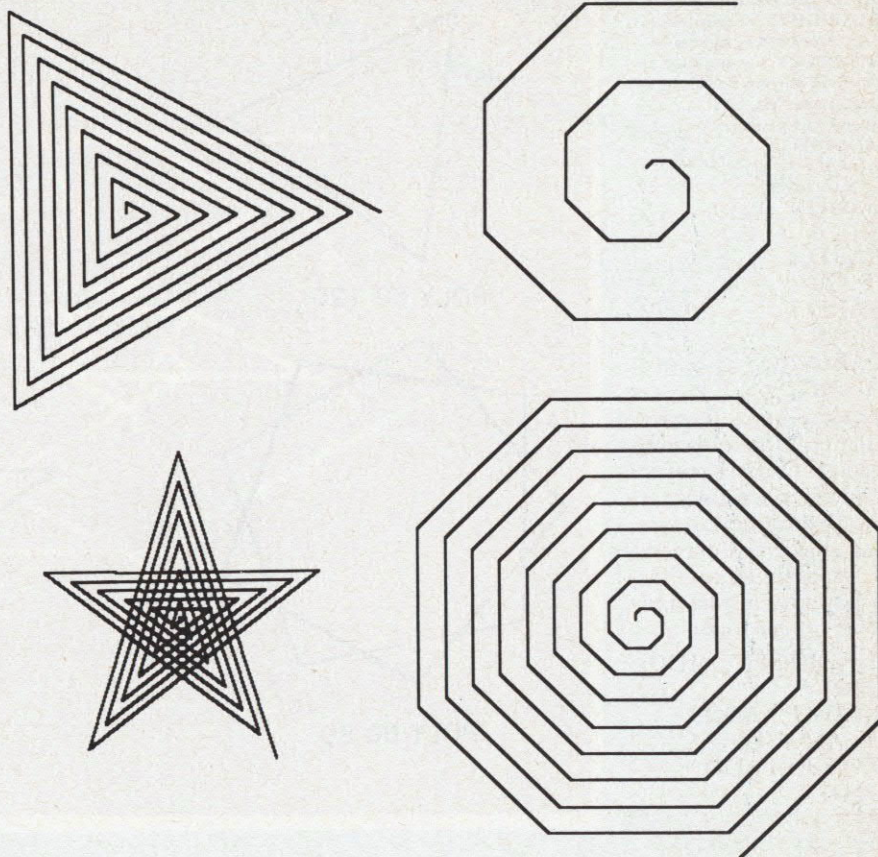
Y finalmente:
DISTANCIA AL PUNTO
12,08
LO CONSIGUIO EN 11 TI-
RADAS!

El núcleo del programa es un procedimiento llamado PLAY. Este procedimiento acepta como entrada un número M que indica el núme-

Figura 3: Figuras creadas mediante POLYSPI. Una variante de POLY, saca partido de la recursión para incrementar el avance de la tortuga en cada llamada que el procedimiento se hace a sí mismo. El resultado es una espiral poligonal. Lo mismo que con POLY, variando el ángulo se modifica la simetría.

ro de tiradas hasta el momento. PLAY comprueba primero si el jugador ya ha terminado. Si es así, imprime un mensaje que indica en cuántos movimientos lo ha hecho y se para. Si no, invita al jugador a realizar una tirada y continúa con el siguiente ciclo aumentando M en 1:

```
TO JUGAR :M
TEST TERMINO?
IFTRUE (PRINT (LO
CONSIGUIO EN))
:M (TIRADAS))
IFCIERTO STOP
TIRADA
JUGAR :M +1
END
```



JUGAR es un procedimiento sencillo en sí mismo porque delega los problemas de comprobación de la terminación del juego y la realización de los movimientos a los procedimientos TERMINO?, y TIRADA.

Este es el procedimiento TIRADA que pide al usuario los datos de ángulo y distancia y desplaza la tortuga de acuerdo con esos datos. Utiliza un subprocedimiento LEERNUMERO, que toma el valor que se escribe por teclado:

```
TO TIRADA
PRINT (CUANTO GIRO A
LA IZQUIERDA?)
LEFT LEERNUMERO
PRINT (CUANTO GIRO A
LA DERECHA?)
FORWARD LEERNU-
MERO
END
```

Para comprobar si el jugador ha terminado, el programa debe comprobar si la posición de la tortuga está suficientemente próxima a un punto predeterminado (por ejemplo 20 pasos). Las primitivas de Logo XCOR y YCOR devuelven las coordenadas x e y de la tortuga. Supondremos que las coordenadas x e y del punto oculto vienen dadas por las variables XPT e YPT. Supuesto que el procedimiento DISTANCIA es quien calcula la distancia entre los dos puntos, el procedimiento TERMINO? podría escribirse así:

```
TO TERMINO?
MAKE "D DISTANCIA
XCOR YCOR
:XPT :YPT
(PRINT (DISTANCIA AL
PUNTO))
:D)
IF :D (20 OUTPUT "CIE-
RTO
OUTPUT "FALSO
END
```

TERMINO? devuelve como valor bien CIERTO o bien FALSO, que son los resultados comprobados por JUGAR para determinar si el juego ha terminado o no. Observe también la utilización de la instrucción MAKE para asignar valores a las variables. En este caso D se

utiliza para designar la distancia.

He aquí el procedimiento para calcular la distancia entre los puntos como la raíz cuadrada de la suma de los cuadrados de la diferencia de las coordenadas:

```
TO DISTANCIA :A :B :X
:Y
MAKE "DX :A - :X
MAKE "DY :B - :Y
OUTPUT SQRT (:DX*
:DX + :DY* :DY)
END,
```

Ahora lo que se necesita es un procedimiento para comenzar el juego:

```
TO JUEGO
CLEARSCREEN
MAKE "PT RANDOM-
COORD
MAKE "YPT RANDOM-
COORD
JUGAR O
END
```

Con esto se limpia la pantalla, se asignan valores (tomados al azar) al punto misterioso de coordenadas XPT e YPT, y se llama a JUGAR con el valor inicial M = 0.

El siguiente procedimiento utilizado para seleccionar coordenadas aleatorias, devuelve un número entre -75 y +75. Funciona llamando a la función de Logo RANDOM para obtener un número aleatorio entre 0 y 150, y le resta luego 75:

```
TO RANDOMCOORD
OUTPUT (RANDOM 150)
-75
END
```

Lo único que se necesita ya para completar el programa es LEER NUMERO que se encarga de tomar el valor introducido por el teclado:

```
TO LEERNUMERO
OUTPUT FIRST REQUEST
END
```

LEERNUMERO utiliza la función de Logo REQUEST, que espera a que el usuario teclee una línea y devuelve luego una lista de todos los items de esa línea. El número deseado se extrae como primero (FIRST) item de la lista introducida (más adelante hablamos sobre listas).

Realmente sería mejor escribir LEERNUMERO de manera que comprobase si el item que toma es un número

y en caso contrario lo rechaza:

```
TO LEERNUMERO
MAKE "TECLA FIRST RE-
QUEST
IF NUMBER? :TECLA
OUTPUT
:TECLA
PRINT (POR FAVOR, UN
NUMERO)
OUTPUT LEERNUMERO
END
```

Observe la última línea del procedimiento. Su efecto es haecer que LEERNUMERO vuelva a ejecutarse tantas veces como sea necesario hasta que se introduzca un número. Esto hace que el juego aparezca de este modo:

```
CUANTO AVANZO?
FJKSL
POR FAVOR, UN NUME-
RO
CINCUENTA
POR FAVOR, UN NUME-
RO
50
...
```

Listas

Hemos visto que la organización de Logo a base de procedimientos lo convierte en un lenguaje sencillo y cómodo para escribir programas. Muchos lenguajes modernos de programación están de hecho organizados de este modo, pero son muy pocos aquellos en los que los procedimientos pueden definirse y modificarse de una manera tan sencilla como en Logo.

Un aspecto mucho más especial de Logo es el modo en que trata las colecciones o series de datos. Esto se hace utilizando listas. Una lista es la secuencia de datos. Por ejemplo:

```
(1 2 MARCHA SIN TE-
MOR)
es una lista de 5 cosas. Los
elementos de la lista pueden
a su vez ser listas. Como por
ejemplo:
((DIEGO VALOR) CAPE-
RUCITA ROJA)
```

que es una lista de tres items, el primero de los cuales es así mismo una lista de 2 items. De manera similar podemos

tener listas cuyos items sean listas y así sucesivamente. Las listas, por consiguiente, son una manera natural de representar estructuras jerarquizadas, esto es estructuras compuestas de partes que a su vez tienen partes compuestas.

Logo incluye una serie de operaciones para manejar estas listas. FIRST extrae el primer item de la lista. En este ejemplo:

```
FIRST (1 2 MARCHA SIN
TEMOR)
es 1 y en siguiente ejemplo:
FIRST ((DIEGO VALOR)
WENDY HOHN)
es (DIEGO VALOR)
```

La operación BUTFIRST devuelve la lista que consta de todos los elementos a excepción del primero, esto es BUTFIRST (1 2 MARCHA SIN TEMOR) es (2 MARCHA SIN TEMOR), mientras que BUTFIRST ((DIEGO VALOR) CAPERUCITA ROJA) es (CAPERUCITA ROJA).

La operación FPUT toma los dos elementos x e y y construye una lista cuyo FIRST es X y cuyo BUTFIRST es y. Por ejemplo: FPUT 5 (2 MARCHA SIN TEMOR)

produce la lista (5 2 MARCHA SIN TEMOR) y FPUT (DIEGO VALOR) (MARCHA SIN TEMOR) produce la lista ((DIEGO VALOR) MARCHA SIN TEMOR).

La operación SENTENCE, como FPUT, construye listas más grandes a base de otras más pequeñas, pero de una manera ligeramente diferente. SENTENCE toma una serie de listas como entradas y combina sus elementos para producir una única lista. Por ejemplo: SENTENCE (DIEGO VALOR) (MARCHA SIN TEMOR) produce la lista (DIEGO VALOR MARCHA SIN TEMOR).

Un aspecto significativo sobre las listas en el sistema Logo es que pueden ser manipuladas como lo que los científicos del cálculo mate-

mático llaman "objetos de datos de primera clase". Esto significa que las listas Logo (al contrario de lo que ocurre, por ejemplo, con las matrices en BASIC) pueden ser:

- Asignadas como valores de variables.
- Pasadas como entradas a procedimientos.
- Devueltas como salidas de procedimientos.

Así por ejemplo, se pueden asignar nombres a listas: NAME "X (OOM PAH) MAKE "Y (HEIGH HO)

Y luego referirse a los valores de esas variables, es decir, BURTFIRST: x es la lista (PAH). Puede también combinar operaciones con listas para crear operaciones más complejas. Por ejemplo: FIRST FIRST ((DIEGO VALOR))

(CAPERUCITA ROJA) Devuelve la palabra DIEGO.

Se pueden asimismo escribir procedimientos para manipular listas:

```
TO DOBLAR :L
  OUPUT SENTENCE :L :L
END
```

```
PRINT DOBLAR (OOM PAH)
(OOM PAH)
OOM PAH OOM PAH
PRINT DOBLAR DOBLAR (OOM PAH)
OOM PAH OOM PAH
OOM PAH OMM PAH
```

Lo que esto implica es la posibilidad de combinar operaciones con listas de la misma manera que se combinan operaciones con números en los lenguajes corrientes. Por ejemplo una operación muy útil con listas es PICKRANDOM, que elige un elemento al azar de una lista de entrada. PICKRANDOM no está disponible como operación primitiva pero se construye fácilmente a base de operaciones más simples, como buscar la longitud de la lista, seleccionar un número aleatorio en un margen determinado y extraer el enésimo ítem de la lista.

Jugar con Textos

Para ilustrar cómo se manejan las listas vamos a exa-

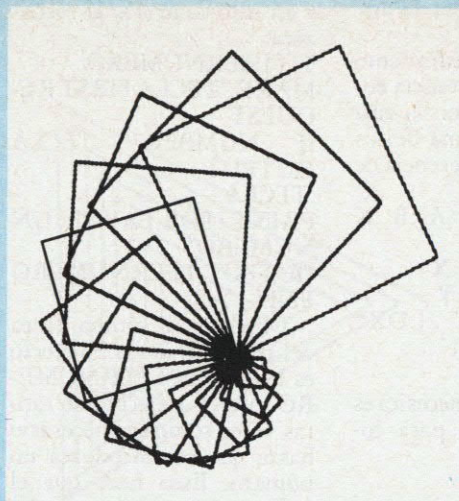


Figura 4:
CUADRESPI es un sencillo programa recursivo que dibuja un cuadrado de un tamaño dado, lo gira, aumenta el tamaño y repite el proceso.

minar un programa que compone tarjetas postales de vacaciones como ésta:

```
QUERIDA CARMEN
ME GUSTARIA QUE ESTUVIESES AQUI
UN BESO JUAN
QUERIDA MARIA
TODO EL MUNDO BIEN.
ESCRIBE PRONTO —TIA PEPA
```

Se empieza por fijar las listas de nombres y frases de las cuales vamos a extraer los elementos de la tarjeta postal:

```
MAKE "NOMBRES
(JUAN CARMEN)
(TIA PEPA)
(INQUILINO)
```

```
MAKE "FRASES
(ME GUSTARIA QUE ESTUVIESES AQUI)
(EL TIEMPO ES MAGNIFICO)
(NOS DIVERTIMOS CON EL SURF)
(TODO EL MUNDO BIEN))
MAKE "CIERRES
(UN BESO)
(HASTA PRONTO)
(ESCRIBE PRONTO))
```

He aquí el programa principal de composición de la tarjeta:

```
TO POSTAL
  PRINT SENTENCE (QUERIDA) NOMBRE
  PRINT TEXTO
  PRINT (SENTENCE)
  DESPEDIDA
  (—) NOMBRE)
  POSTAL
END
```

La llamada recursiva de la última línea hace que el pro-

cedimiento pueda seguir imprimiendo nuevas tarjetas postales continuamente. (Comparar con los procedimientos SQ y POLY).

Los procedimientos NOMBRE, TEXTO y DESPEDIDA generan los elementos de la tarjeta postal seleccionando ítems de las listas apropiadas:

```
TO NOMBRE
  OUTPUT PICKRANDOM :NOMBRES
END
```

```
TO TEXTO
  OUTPUT PICKRANDOM :FRASES
END
```

```
TO DESPEDIDA
  OUTPUT PICKRANDOM :DESPEDIDAS
END
```

Se puede cambiar el programa de las tarjetas postales de manera que aumente automáticamente su repertorio en frases con una cierta frecuencia (diagramas una vez de cada tres veces) pidiendo al usuario que teclee una frase y añadiéndola a la lista FRASES. Para hacerlo, añada al procedimiento POSTAL una línea:

```
IF 1,IN,3 INCORPORAR, FRASE
El procedimiento 1, IN, 3 da como resultado CIERTO con probabilidad de una sobre tres y FALSO en los otros casos. Una posible manera de escribir este procedimiento es:
TO 1, IN, 3
```

```
IF (RANDOM 3) = 0 OUTPUT "CIERTO
OUTPUT "FALSO
END
```

Este es el programa que aprende una nueva frase:

```
TO INCORPORAR, FRASE
  PRINT (TECLEE UNA NUEVA FRASE)
  MAKE "FRASES FPUT REQUEST :FRASES
END
```

La idea es que REQUEST devuelva (como una lista) la frase que el usuario teclea en respuesta al mensaje. Esta se añade a FRASES (por medio de FPUT) de manera que el programa podrá utilizarla en futuras postales, como por ejemplo:

```
TECLEE UNA NUEVA FRASE
NO TE OLVIDES DE DAR DE COMER AL PERRO.
```

```
QUERIDA TIA PEPA
NO TE OLVIDES DE DAR DE COMER AL PERRO
HASTA PRONTO —JUAN
```

Otro cambio que puede hacerse es generar tarjetas postales más largas, cuyo TEXTO, consta de una o más frases. Una manera de hacerlo es alterar el procedimiento TEXTO como sigue:

```
TO TEXTO
  IF1, IN, 2 OUTPUT SINGLE, FRASE
  OUTPUT SENTENCE TEXTO
  SINGLE, FRASE
```

```
TO SINGLE, FRASE
  OUTPUT PICKRANDOM :FRASES
END
```

Aquí se utiliza la recursión de una manera indirecta. La mitad de las veces que se llama a TEXTO producirá una sola frase, como anteriormente. (El procedimiento 1. IN. 2 es análogo al 1.IN.3 descrito antes). Pero la otra mitad de las veces, genera recursivamente un nuevo TEXTO y lo combina (utilizando SENTENCE) con una frase elemental. El nuevo TEXTO (llamado recursivamente) generará él mismo una frase sencilla sólo la mitad de las veces. El

Fotos 2a y 2b: Dos esquemas de trenzado contruidos a partir de una figura básica (un triángulo dentro de un cuadrado). Las diferencias entre ambos se deben a que se han utilizado diferentes orientaciones cuando la figura básica se combina en configuraciones de "nivel superior".

resto llamará a un tercer TEXTO. El resultado final es que una llamada a TEXTO generará una frase sencilla aproximadamente la mitad de las veces, 2 frases 1/4 de las veces, 3 frases sobre 1/8 de las veces, y así sucesivamente.

He aquí el programa final de las tarjetas postales en acción:

QUERIDA TIA PEPA
NOS DIVERTIMOS CON
EL SURF. NO TE OLVI-
DES DE DAR DE COMER
AL PERRO.
HASTA PRONTO —CAR-
MEN

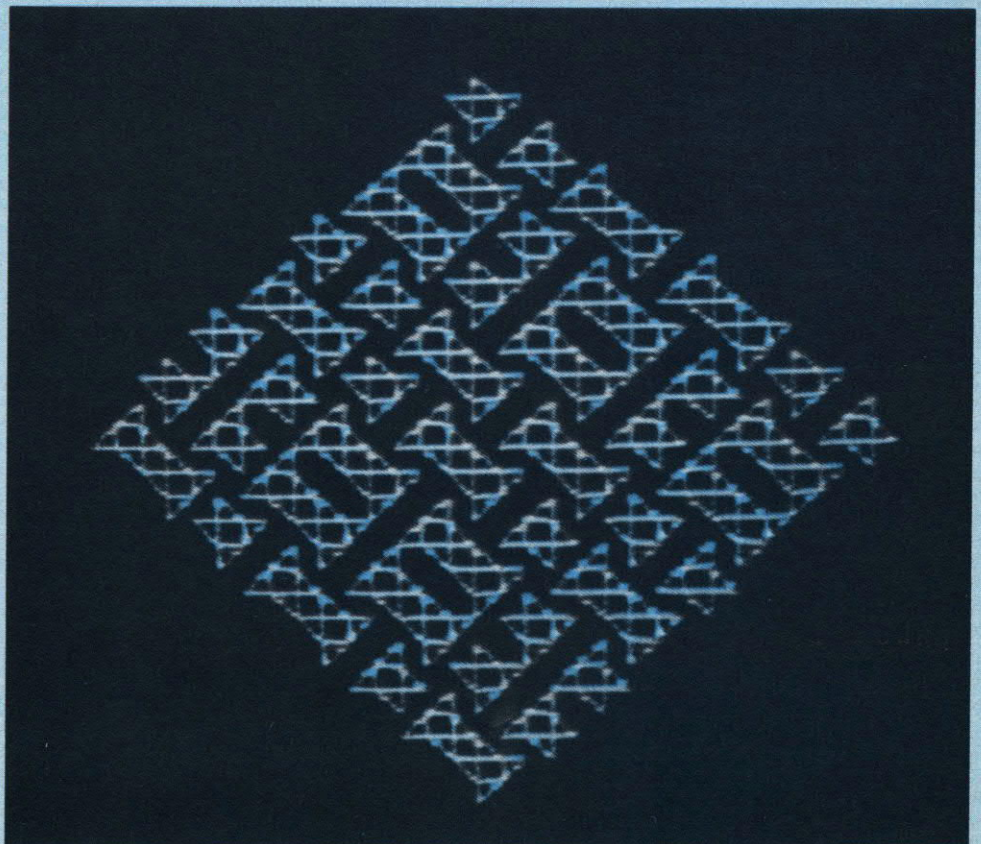
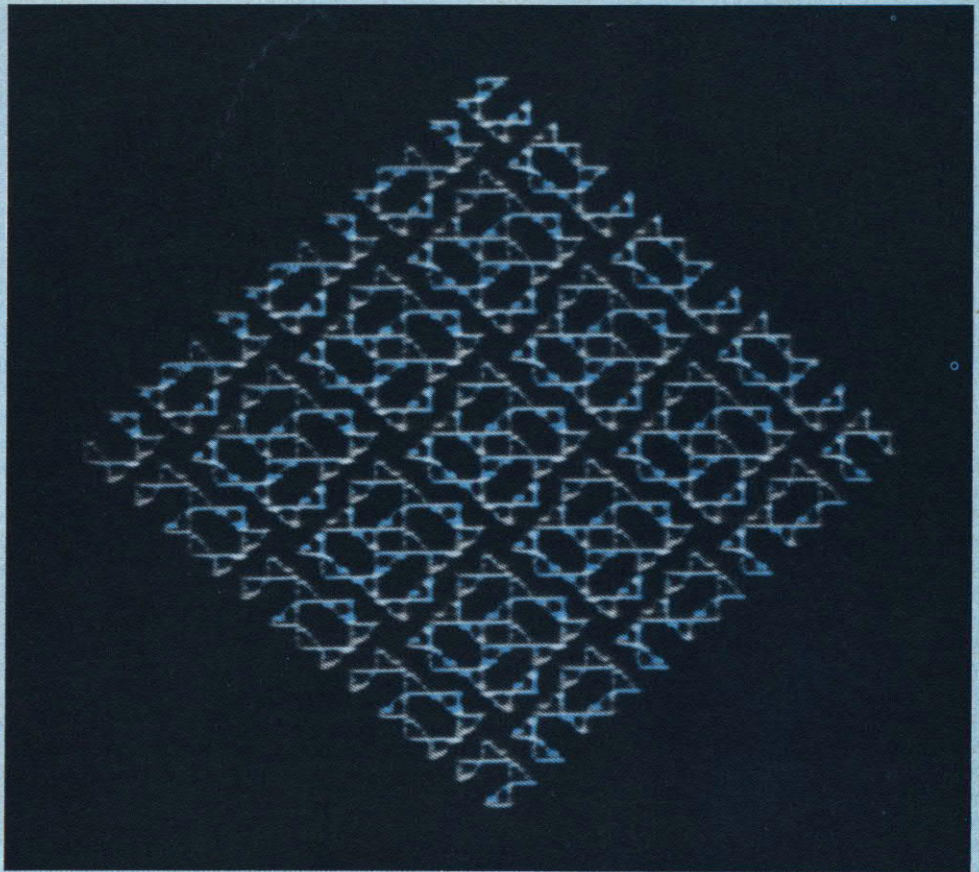
TECLEE UNA NUEVA
FRASE.
NECESITO DINERO

QUERIDA MARIA
EL TIEMPO ES MAGNIFI-
CO
ME GUSTARIA QUE ES-
TUVIERAS AQUI
NECESITO DINERO
ESCRIBE PRONTO —JUAN

Los procedimientos de generación de textos entreteni- dos de escribir y de ejecutar y también son sencillos de modificar. Pueden hacerse tan elaborados o sencillos como se desee y aplicar las mismas ideas para producir ensayos, poemas, cartas de amor, o cualquier otra cosa.

El Tejido Recursivo

El éxito de Logo como catilizados en el aprendizaje va más allá del lenguaje en sí, aunque el lenguaje juega un papel crucial. El mejor tipo de actividad con Logo es una síntesis de programación, matemáticas, estética y, sobre todo, oportunidad de explorar. Un ejemplo particularmente llamativo es "el tejido recursivo", un programa inventado por Andreadis- say Doug Hill. Este esquema permite escribir sencillos



procedimientos que dibujan esquemas como los mostrados en las figuras 2 y 3, proporcionando literalmente miles de millones de posibilidades para examinar y explorar.

La idea es la siguiente. Suponga un programa que hace un dibujo dentro de un cuadrado de un tamaño determinado. Reduciendo a la mitad la escala y juntando cuatro copias del dibujo, puede obtener una configuración más compleja en un cuadrado del tamaño original, como muestra en la figura 5a. Con el mismo dibujo original se puede dar lugar a muchas configuraciones diferentes, porque cada copia colocada en una esquina del cuadrado puede girarse un múltiplo arbitrario de 90 grados como se muestra en la figura 5b.

Para convertir esta idea en un programa de ordenador, suponga que tiene un procedimiento llamado PROC que dibuja un determinado diseño en un cuadrado. Admitamos que PROC toma como entrada un valor S que especifica el tamaño del cuadrado, determinado por la mitad de su diagonal. Admitamos también que PROC está pensado para que la tortuga comience a dibujar en el centro del cuadrado apuntando a un vértice y terminando el dibujo en el mismo estado.

Supongamos ahora que tenemos un cuadrado de tamaño S dividido en cuatro "esquinas cuadradas" cada una de ellas de tamaño S/2. El proceso siguiente está diseñado para comenzar con la tortuga en el centro del cuadrado, mirando hacia una de las esquinas. Dibuja una copia del diseño previsto en PROC en el cuadrado correspondiente a esa esquina, y hace que la tortuga vuelva al centro del cuadrado grande. Luego gira la tortuga 90 grados para que apunte al si-

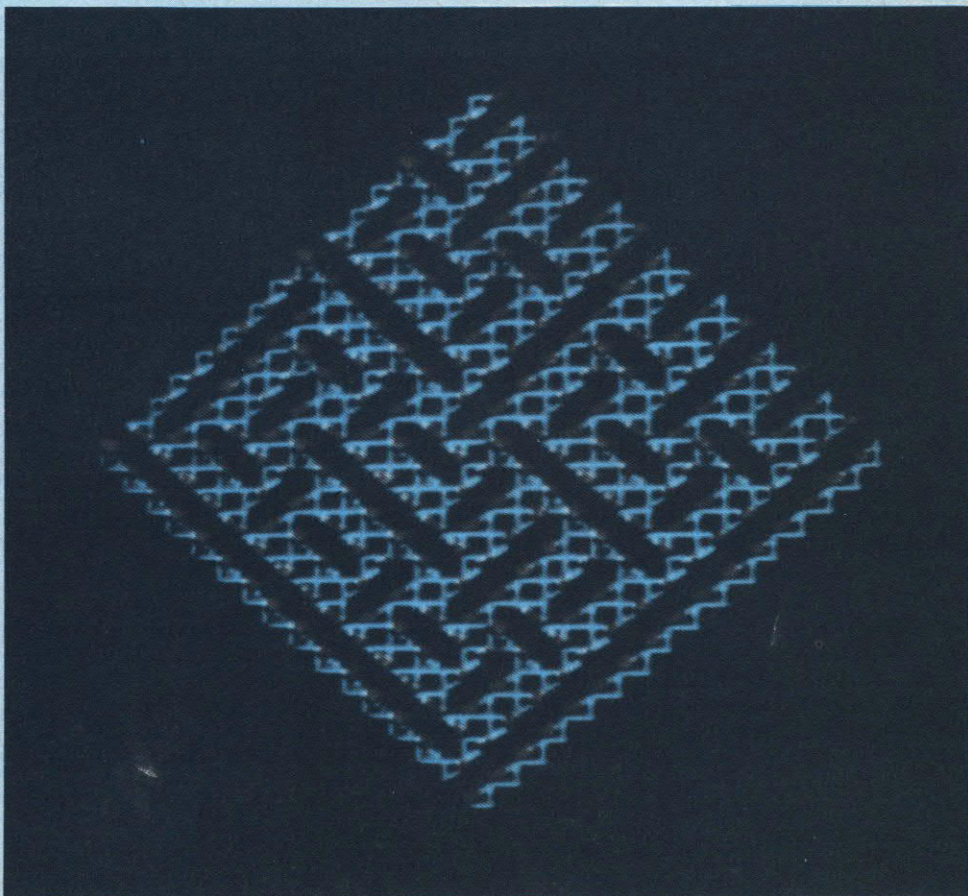
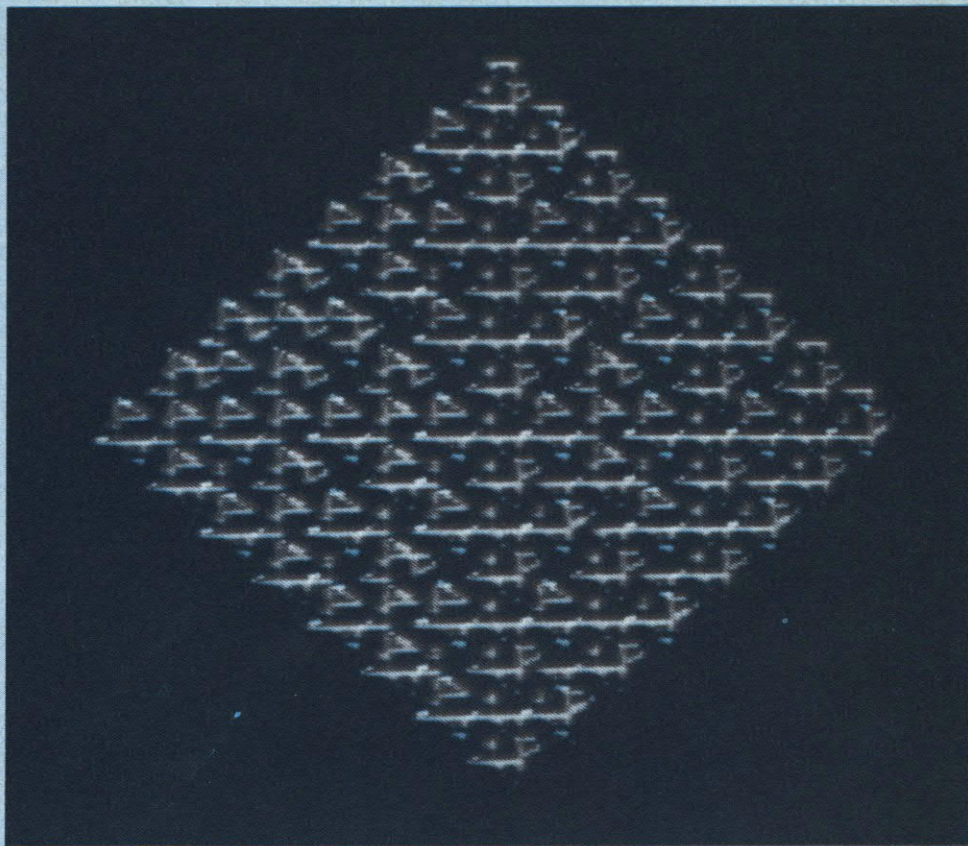


Foto 3a y 3b: Dos dibujos de tejido contruidos a partir del mismo triángulo básico que los mostrados en las fotos 2a y 2b. Los cuatro utilizan el esquema de montaje de las figuras 5a y 5b y las fotos 4a y 4c, extendido a cuatro niveles.

guiente vértice. Los pasos en este proceso son:

1. Mover la tortuga hacia adelante (FORWARD) una distancia $S/2$. Con esto se lleva la tortuga al centro del pequeño cuadrado, apuntando a un vértice de ese cuadrado.

2. Ejecutar el procedimiento PROC con una entrada de $S/2$ (la mitad de la diagonal del cuadrado menor). Esto hace el dibujo y deja la tortuga en el centro del cuadrado pequeño.

3. Hacer retroceder la tortuga (BACK) una distancia de $S/2$ para que vuelva al centro del cuadrado grande.

4. Girar la tortuga 90 grados.

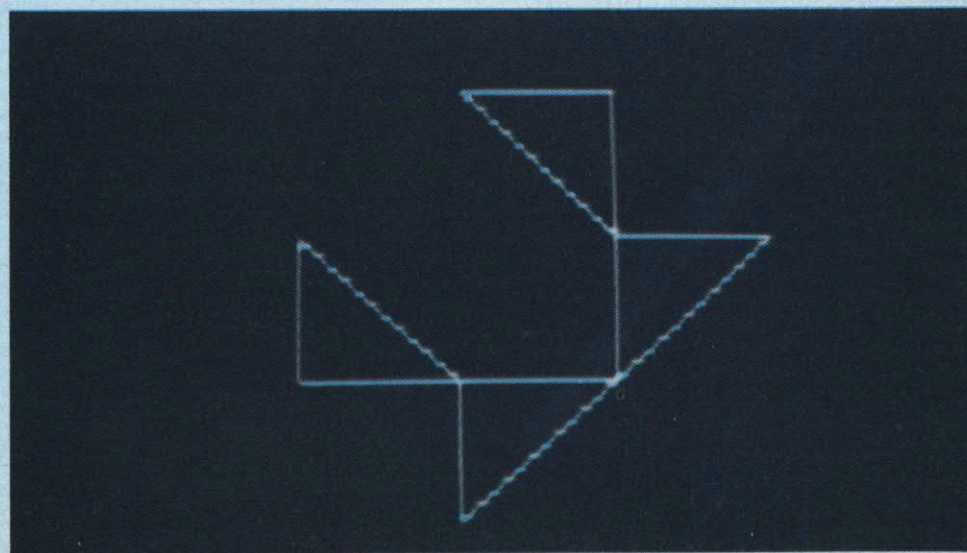
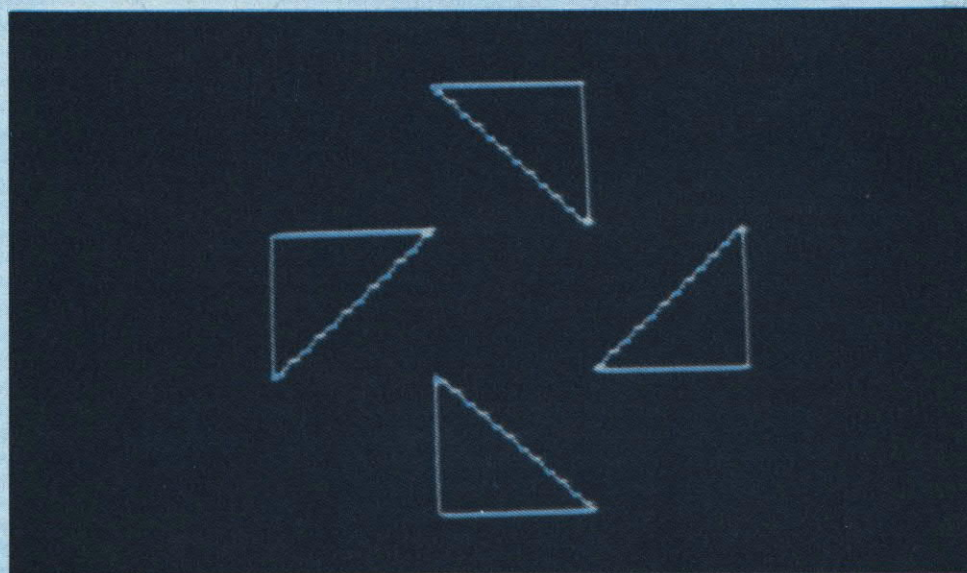
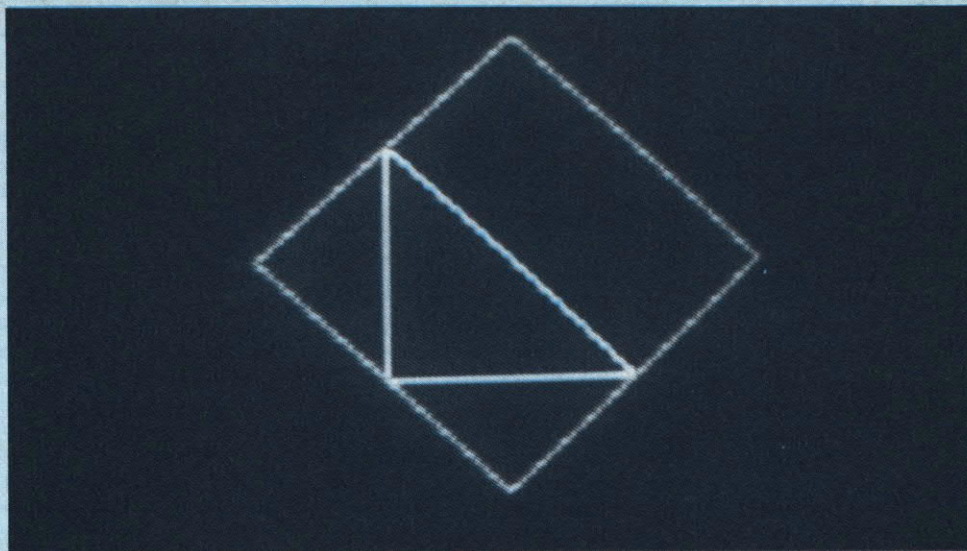
Además, antes de realizar el paso 2, se puede girar el dibujo un múltiplo de 90 grados. Si lo hace así deberá realizar el giro opuesto al final del paso 2 de manera que la tortuga acabe mirando en la misma dirección en que comenzó.

El procedimiento CORNER que se da a continuación lleva a cabo esta estrategia. CORNER acepta tres entradas. La primera, A, es un múltiplo de 90 grados que corresponde al ángulo que se gira antes de realizar el dibujo (A es un entero de 0 a 3). La siguiente entrada PROC es el nombre del procedimiento que hace efectivamente el dibujo.

PROC acepta a su vez una entrada que especifica el tamaño del dibujo. La tercera entrada de CORNER es un número S que especifica el tamaño del cuadrado. He aquí las instrucciones:

```
TO CORNER :A :PROC :S
FORWARD :S/2
RIGHT 90 * :A
```

Figura 5: Dado un dibujo en el interior de un cuadrado, cuatro copias del mismo, cada una de ellas a escala mitad de la original, producirán dentro del mismo cuadrado un dibujo más complejo. Variando las orientaciones relativas de las cuatro copias se consiguen diseños completos diferentes. Este es el esquema básico de montaje de los dibujos de las fotos 2a, 2b, 3a y 3b, comenzando con un triángulo y repitiendo el proceso de montaje cuatro veces (esto es, trabajando con un elemento final a escala $1/16$ del tamaño del cuadrado original).




```
FIGURA :PROC :S/2
LEFT 90 * :A
BACK :S/2
RIGHT 90
END
```

El programa CORNER utiliza un subprograma FIGURA, que acepta como entradas un nombre de procedimiento y un número ejecutando el procedimiento con el número como entrada. FIGURA está implementada en base a la primitiva de Logo RUN, que ejecuta una lista como si fuera una línea de comandos que se hubiera introducido:

```
TO FIGURA :PROC :INPUT
RUN SENTENCE :PROC
:INPUT
END
```

Por ejemplo, si ejecuta: FIGURA (CUADRADO) 100 lo que se hace es combinar (CUADRADO) y 100 para obtener la lista (CUADRADO 100) y ejecutarla como si se tratara de una línea de instrucciones de Logo introducida desde el teclado, es decir, como si se hubiera ordenado ejecutar CUADRADO 100.

El procedimiento CORNER comienza con la tortuga apuntando a una esquina del cuadrado grande, y termina apuntando a la siguiente esquina. Esto significa que llamando cuatro veces a CORNER se obtiene un montaje completo de las 4 repeticiones del dibujo. Cada una de estas llamadas a CORNER puede especificar un múltiplo de 90 grados diferente que se aplicará al dibujo en esa esquina en concreto. Puesto que cada montaje tiene 4 esquinas y cada esquina puede estar girada un ángulo diferente existen 4 elevado a 4 o sea 256 posibles montajes para un dibujo determinado. He aquí dos de ellos:

```
TO MONTAR1 :PROC :S
REPEAT 4 (CORNER 0 :PROC :S)
END
```

```
TO MONTAR2 :PROC :S
CORNER 0 :PROC :S
CORNER 1 :PROC :S
CORNER 2 :PROC :S
```

```
CORNER 3 :PROC :S
END
```

La figura 4B muestra el resultado producido cuando se introduce:

```
MONTAR1 (TRI) 100
```

donde TRI es un procedimiento que dibuja un pequeño triángulo dentro del cuadrado como muestra la foto 4a. (De acuerdo con la estrategia general de montaje, TRI debe ser un procedimiento que acepta una entrada que especifica el tamaño del cuadrado). A efectos comparativos, la foto 4c muestra un montaje diferente).

```
MONTAR2 (TRI) 100
```

Veamos ahora la idea feliz. Los procedimientos MONTAR permiten montar juntos cuatro copias de un dibujo. Por otra parte, introduciendo:

```
MONTAR1 (TRI) 100
```

es también una orden que dibuja un esquema en un cuadrado. De hecho, la lista (MONTAR1 (TRI)) cuando se combina con un tamaño (por medio del procedimiento FIGURA), produce un comando que dibuja un esquema en un cuadrado de un tamaño determinado. Por consiguiente, usted puede utilizar un procedimiento MONTAR para montar juntos cuatro de esos dibujos, por ejemplo:

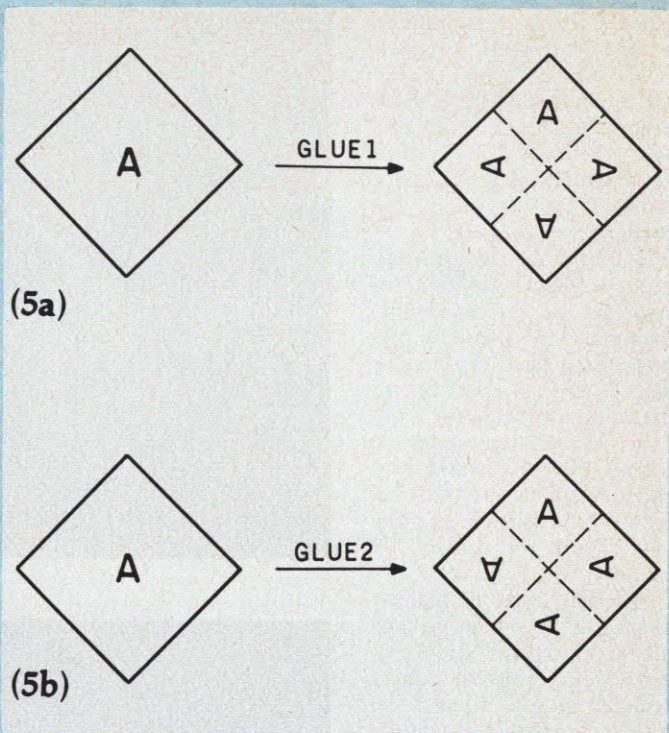
```
MONTAR1 (MONTAR1 (TRI)) 100
```

```
MONTAR2 (MONTAR1 (TRI)) 100
```

Pero, cada uno de estos montajes de "nivel 2" es a su vez algo que puede ser también montado, consiguiéndose así configuraciones de nivel 3 como:

```
MONTAR1 (MONTAR1 (MONTAR2 (TRI)) 100
```

y así sucesivamente. Los esquemas mostrados en las fotos 2a, 3a, y 3b son, de hecho, montajes de nivel 4 basados en el mismo procedimiento TRI, que utilizan diferentes rotaciones en los diversos niveles.



Se dispone así de una gama de posibilidades para investigar. Cuatro niveles de montaje con 256 posibles orientaciones a cada nivel dan 256 elevado a 4 o más de 4.000 millones de posibles montajes de nivel 4, todos ellos a partir de un dibujo elemental. (El número de configuraciones diferentes es menor debido a las simetrías en el proceso de montaje, que de por sí es un fenómeno interesante de investigar). Para más variedad puede probar con diferentes dibujos de base o incluso desarrollar diferentes esquemas de montaje, como el que se deriva de dividir un triángulo equilátero en cuatro triángulos equiláteros menores.

Logo se describe a veces como un lenguaje de programación. Quienes hemos diseñado Logo tendemos a pensar que se trata más bien de un entorno de enseñanza basado en ordenador, en que las actividades (como la exploración de simetrías de POLY) son parte tan integral del mismo como los medios de programación utilizados (recursión y listas).

Los próximos años van a ser muy interesantes en lo

Foto 4: En la foto 4a aparece un triángulo dentro de un cuadrado, tal como lo dibuja el procedimiento TRI. (El cuadrado en color no es dibujado por TRI). Las fotos 4b y 4c muestran dos montajes de nivel uno diferente que consiguen una figura más complicada dentro del cuadrado. Los complejos diseños de las fotos 2a, 2b, 3a y 3b son montajes de nivel cuatro basados en la figura básica dibujada por TRI.

que se refiere a la enseñanza con ayuda de ordenador porque los ordenadores personales están empezando a ser suficientemente potentes como para explotar sistemas diseñados más para la comodidad del usuario que para la conveniencia de los compiladores. Si somos capaces de disipar el error de que aprender a utilizar los ordenadores consiste en perder el tiempo preocupándose con los índices de las matrices o con disquisiciones acerca de si x es entero o real, podremos empezar a pensar en la programación como un medio de generación de ideas. La programación es una actividad que describe cosas. Las descripciones están hechas en términos que el ordenador puede entender.

Harold Abelson
©Byte/Ordenador Popular

¿POR QUE LOGO?

Logo está diseñado para estimular el desarrollo de la capacidad de resolución de problemas

Logo es un lenguaje para aprender. Esta sentencia, una de las máximas del movimiento Logo, encierra un sutil juego de palabras. El significado obvio es que se trata de un lenguaje para aprender a programar; Logo está diseñado para hacer que la programación de ordenadores sea lo más sencilla de entender que sea posible. Pero Logo es también un lenguaje para aprender en general. Dicho a lo grande, es un lenguaje para aprender a pensar. Su historia está fuertemente ligada a la investigación en cálculo automático, y especialmente en inteligencia artificial. Pero también hunde sus raíces en las investigaciones de Jean Piaget sobre el modo en que los niños desarrollan su capacidad de pensamiento.

En cierto sentido todos los lenguajes de programación son lo mismo. Dicho de otro modo, si usted puede resolver un problema en un lenguaje, generalmente es también capaz de resolverlo en otro. Lo que diferencia a los lenguajes es que algunos tipos de problemas son más sencillos de resolver en un lenguaje que en otro. Los diseñadores de lenguajes deciden qué tipo de problemas resolverá mejor el suyo en particular y en función de eso eligen ciertas opciones de diseño.

Logo como lenguaje de programación

Vamos a posponer por un momento los aspectos edu-

cativos en sentido amplio. Primero, vamos a considerar a Logo simplemente como un lenguaje de programación. ¿En qué se parece a otros lenguajes y en qué se diferencia? Dejando a un lado los detalles sintácticos, existen varios puntos sustanciales del diseño de un lenguaje, en base a los cuales Logo puede ser comparado con otros lenguajes.

Logo es procedimental

Un proyecto de programación en Logo no se escribe como un gran programa. Por el contrario, el programa se divide en piezas pequeñas, y para cada pieza se escribe un procedimiento individual. En este aspecto, Logo se parece a los lenguajes más modernos. Pascal, APL, LISP, C, e incluso FORTRAN, permiten la división de un programa en procedimientos independientes. Entre los lenguajes más populares de propósito general, sólo BASIC carece de esta capacidad. (Los programas de muestra para este artículo están escritos en el Logo de Apple, un dialecto creado por Logo Computer System Inc. Otras versiones de Logo difieren de ésta en pequeños detalles).

Considérese el programa Logo del listado 1a. Incluso si usted no sabe nada de Logo, es probable que vea con facilidad lo que hacen este par de procedimientos. Compárelo con la versión en BASIC del listado 1b.

La estructura GOSUB del BASIC es más débil que la auténtica capacidad de división en procedimientos, por diversas razones. Para empezar, la subrutina en BASIC

no es un programa independiente; si se omitiera la línea 100, el flujo normal del programa "entraría" en la subrutina. Más importante aún, no existe en BASIC el concepto de entradas (o parámetros) para procedimientos, como PREGUNTA y RESPUESTA en el programa Logo. En su lugar, deben utilizarse instrucciones extras para asignar valores a las variables Q\$ y A\$, explícitamente.

Esta asignación explícita no es simplemente un inconveniente; significa que la parte principal del programa debe "conocer", detalles internos de las subrutinas. En la versión Logo, el procedimiento llamado QUIZ sólo sabe que el procedimiento QA tiene dos entradas, una pregunta y una respuesta. Si QA se modificara utilizando diferentes nombres para las variables, QUIZ continuaría funcionando igual. De la misma manera, aunque este ejemplo en particular no lo muestre, los procedimientos Logo pueden tener una salida que es comunicada al procedimiento que efectúa la llamada. (La sentencia DEF del BASIC proporciona una versión limitada de procedimientos con salidas; las limitaciones estriban en que entradas y salidas han de ser numéricas y la definición ha de hacerse en una sola línea, sin bifurcaciones condicionales).

Logo es interactivo

Al igual que BASIC y a diferencia de Pascal, Logo permite teclear un comando y que éste se realice de manera inmediata. Igualmente es muy rápido y sencillo cam-

biar cualquier línea del programa. Otros lenguajes interactivos son LISP y APL; ejemplos de programas no interactivos son C y FORTRAN.

Que un lenguaje sea o no interactivo tiene que ver con su eficiencia.

De manera esquemática, digamos que el desarrollo de programas es generalmente más rápido si se hace en un lenguaje interactivo, pero los programas ya escritos generalmente se ejecutan más deprisa en un lenguaje que no sea interactivo. La diferencia tiene que ver con el mecanismo mediante el cual el ordenador "entiende" el programa del usuario.

Cada ordenador entiende por construcción un lenguaje particular. Este "lenguaje de máquina" es diferente para cada tipo de ordenador. Como las instrucciones de lenguaje de máquina se representan como números, no son sencillas de leer para la gente. Por ejemplo, el número 23147265 podría significar "sumar el número de la posición de memoria 147 al número que figura en la posición de memoria 265". Los programas escritos en lenguaje de alto nivel, incluyendo Logo y los otros lenguajes que hemos mencionado, deben ser traducidos al lenguaje máquina antes de que el ordenador pueda ejecutarlos. Esta traducción se lleva a cabo por otro programa de ordenador, que puede ser de dos tipos: compilador o intérprete.

Un compilador de Pascal, por ejemplo, toma un programa escrito en Pascal y lo traduce (compila) a lenguaje de máquina del ordenador que estamos utilizando. El programa traducido se guar-

da en un medio permanente ya en lenguaje de máquina del ordenador que estemos utilizando. El programa traducido se guarda en un medio permanente ya en lenguaje de máquina (probablemente como un fichero dentro de un disco flexible). A partir de este momento, el programa en lenguaje de máquina puede ser ejecutado directamente. El proceso de compilación dura bastante tiempo. Pero una vez que está terminado, ejecutar el programa compilado es muy rápido, porque ya no necesita ser compilado de nuevo.

Un intérprete de Logo, por el contrario, no crea una versión permanente de lenguaje máquina de un programa. En cambio, cada instrucción de Logo es traducida y ejecutada cada vez que el programa pasa por ella. El intérprete no genera una representación en lenguaje máquina del programa, sino que simplemente ejecuta instrucciones de lenguaje máquina. Si una instrucción de Logo ha de ser ejecutada 6 veces, es traducida 6 veces. (En la actualidad, algunos intérpretes, incluyendo el de Logo de Apple, guardan una traducción parcial de cada procedimiento de manera que la segunda ejecución es algo más rápida que la primera; este proceso es demasiado complicado para explicarlo detenidamente en el espacio de este artículo).

Los lenguajes interpretados pueden ser interactivos. Supongamos que queremos hallar el valor de $2 + 2$ en Pascal. Primeramente hay que utilizar el editor de textos del sistema Pascal para escribir un fichero en disco que contenga el programa fuente. A continuación entra en acción el compilador de Pascal, que traducirá el programa a lenguaje de máquina. Finalmente, se ejecuta el programa compilado y el ordenador contestará que la suma de $2 + 2$ es 4. En un lenguaje interpretado como Logo, simplemente hay que escribir `PRINT 2+2` para obtener inmediatamente el mismo resultado.

La situación en la que la interacción es más impor-

tante es la fase del desarrollo de programas. Si estamos escribiendo un programa complicado, probablemente no funcionará con toda corrección la primera vez que se prueba. Habrá que probarlo, ver dónde falla, cambiar el programa, y probarlo de nuevo. Para ver que es lo que iba mal, lo ideal es poder utilizar una depuración interactiva. (Paramos el programa cuando ocurre el error e introducimos los comandos que sean necesarios para examinar los valores de la variable en ese momento). Este ciclo de depuración ha de repetirse las veces que sean necesarias hasta que el programa funcione correctamente por completo. Aunque un compilador pueda hacer que el programa se ejecute más rápidamente, un intérprete es lo más adecuado para llevar a cabo el proceso completo de depuración con mayor rapidez, porque es mucho más fácil localizar y corregir los errores. Sólo cuando el programa funciona bien y hay que ejecutarlo todos los días sin ninguna modificación es cuando la versión compilada es realmente más rápida.

La flexibilidad y sencillez de utilización de un lenguaje interactivo resulta particularmente valiosa en un entorno educativo. Para un estudiante de programación no existe normalmente la fase de producción, ya que el programa sigue siendo un objetivo mientras no funciona. Cuando ya lo hace, el estudiante pasa al siguiente problema. En este tipo de ambiente, la ventaja de velocidad del compilador no tiene objeto. En el marco de una empresa, por el contrario, la utilización de un programa para la producción real, es probablemente más importante, por lo que en este caso un compilador resulta más deseable.

Algunos lenguajes utilizan esquemas mixtos. BASIC (normalmente, un lenguaje interpretado) tiene compiladores que permiten al usuario cambiar interacción por eficacia. Algunos compiladores de LISP pueden coexistir con intérpre-

```
(1a)
TO QUIZ
QA (¿CUAL ES LA MEJOR PELICULA DE TODOS
LOS TIEMPOS?)
(CASABLANCA)
QA (¿CUANTO ES 2+2?) (5)
QA (¿QUIEN ESCRIBIO "LA VUELTA AL MUNDO
EN 80 DIAS?")
(JULIO VERNE)
END
```

```
TO QA: PREGUNTA :RESPUESTA
TYPE: PREGUNTA
TEST EQUALP: RESPUESTA READLIST
IF TRUE (PRINT ("CORRECTO!))
IF FALSE (PRINT SENTENCE (NO; LO CORRECTO
ES): RESPUESTA)
END
```

```
(1b)
10 Q$="¿CUAL ES LA MEJOR PELICULA DE TO-
DOS LOS TIEMPOS?"
20 A$="CASABLANCA"
30 GOSUB 1000
40 Q$="¿CUANTO ES 2+2?"
50 A$X="5"
60 GOSUB 1000
70 Q$="QUIEN ESCRIBIO LA VUELTA AL MUNDO
EN 80 DIAS?"
80 A$="JULIO VERNE"
90 GOSUB 1000
1000 GOTO 9999
1000 PRINT Q$
1010 INPUT R$
1020 IF R$=A$ THEN GOTO 1100
1030 PRINT "NO; LO CORRECTO ES";A$
1040 RETURN
9999 END
```

Listado 1: Comparación de Logo y BASIC. Ambos programas hacen el mismo juego de preguntas y compara las respuestas del usuario con las del autor. En la versión BASIC (listado 1b), la subrutina de "preguntar" (líneas 1000-11000) no es un programa independiente. En la versión Logo (listado 1a), el procedimiento QA podría muy bien ser un programa aparte, utilizable por otros programas.

tes, de manera que algunos procedimientos pueden ser compilados mientras otros se depuran de manera interactiva. Algunas versiones de Pascal son compiladas a un lenguaje intermedio llamado código-p, que luego es interpretado. FORTH utiliza un sistema similar de compilación parcial, pero el compilador es parte del entorno en tiempo de ejecución, de manera que las instrucciones individuales pueden ser compiladas y ejecutadas interactivamente.

Logo es recursivo

En un programa, un procedimiento puede utilizar otro como subprocedimiento para realizar parte de su trabajo. Un lenguaje es recursivo cuando un procedimiento puede ser subprocedimiento de sí mismo.

Todos los modernos lenguajes "procedimentales" permiten la recursión. Entre los utilizados más ampliamente, sólo FORTRAN permite los procedimientos pero no la recursión. (BASIC,

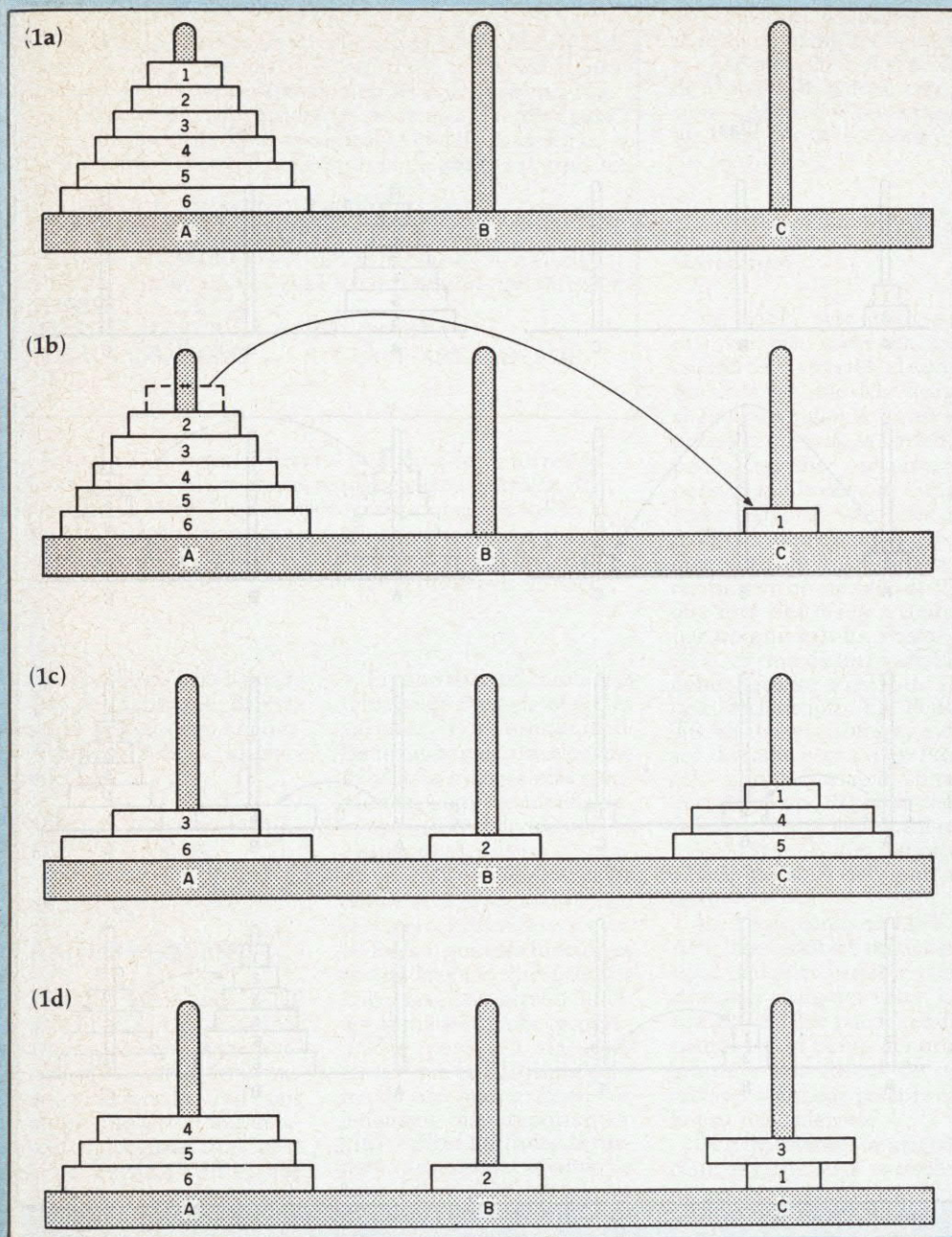


Figura 1: Movimientos típicos en el juego de las Torres de Hanoi. La figura la muestra la posición inicial, la 1b el primer movimiento y la 1c la situación tras varios movimientos. En la 1d se muestra una situación ilegal, con un disco grande situado sobre otro más pequeño.

Supongamos ahora que son 6 los discos que entran en juego (ver figura 2b). De nuevo, hemos de comenzar por llevar el disco 6, el mayor de todos, de la guía A a la guía B. Pero ahora son 5 los discos que tenemos antes que él, no sólo 1. Esto nos conduce a resolver un subproblema: llevar los cinco discos de la guía A a la guía C. Pero esto no es sino exactamente el mismo problema de la Torre de Hanoi pero planteado ahora con cinco discos en lugar de seis. El subproblema no es más que una versión más sencilla del problema principal. Esta reflexión sugiere la solución recursiva que se muestra en el listado 2.

Al trabajar en este problema, no debe olvidarse que cada utilización del procedimiento HANOI tiene sus propias variables privadas; el valor de NUMERO, por ejemplo, permanece constante durante una utilización particular del procedimiento, aunque haya por medio otro uso de HANOI con un valor diferente de NUMERO.

Además de Logo, son varios los lenguajes que permiten la recursión (por ejemplo Pascal, C, LISP, y APL). El estilo de Logo, sin embargo, pone mayor énfasis en la utilización de la recursión que algunos de estos lenguajes. C y Pascal permiten la recursión, pero en realidad priman la iteración (iteración significa decirle al ordenador que ejecute algo repetidamente. La construcción FOR... NEXT del BASIC es un ejemplo típico). En Logo es justamente al revés: la iteración es posible pero se prefiere la recursión. En muchas situaciones, no está clara cuál es la solución ideal. Para los casos en que puede utilizarse la iteración, probablemente resulta más simple, pero en algunas ocasiones

como se dijo antes, no permite ninguna de las dos facetas). Podría pensarse que la recursión no es demasiado importante. ¿Qué lo hace diferente de cualquier otro tipo de utilización de subprocedimientos? No es fácil de explicar de una manera sencilla por qué la recursión es importante. La idea que encierra la recursión, sin embargo, tiene una importancia matemática profunda. Al permitir que un problema complicado se pueda describir en términos de versiones

más sencillas de sí mismo, la recursión permite expresar la solución de una gran cantidad de problemas de una forma compacta.

Un ejemplo muy conocido de problema que se resuelve bien utilizando la recursión es el juego llamado las Torres de Hanoi. Este juego consta de una serie de discos de diferente tamaño apilados inicialmente en una de las tres guías existentes, con el más pequeño situado arriba. El problema consiste en cambiar los discos de guía,

moviendo uno a uno y de manera que nunca se ponga uno más grande sobre uno más pequeño (ver figura 1).

Para resolver el problema, observe primero que es muy sencillo si sólo existen dos discos (ver figura 2a). Es fácil darse cuenta de que necesitamos poner el disco 2 en la guía B. Para hacerlo, hemos de librarnos primero del disco 1. Por consiguiente lo primero que hacemos es llevar el disco 1 a la guía C, luego el disco 2 a la guía B, y el disco 1 a la guía B.

Figura 2: He aquí como se resuelve el juego dividiéndolo en subproblemas más sencillos con soluciones similares. En la figura 2a aparece el caso más sencillo: el juego reducido a dos discos. La figura 2b muestra el resultado de aplicar la misma estrategia a más discos.

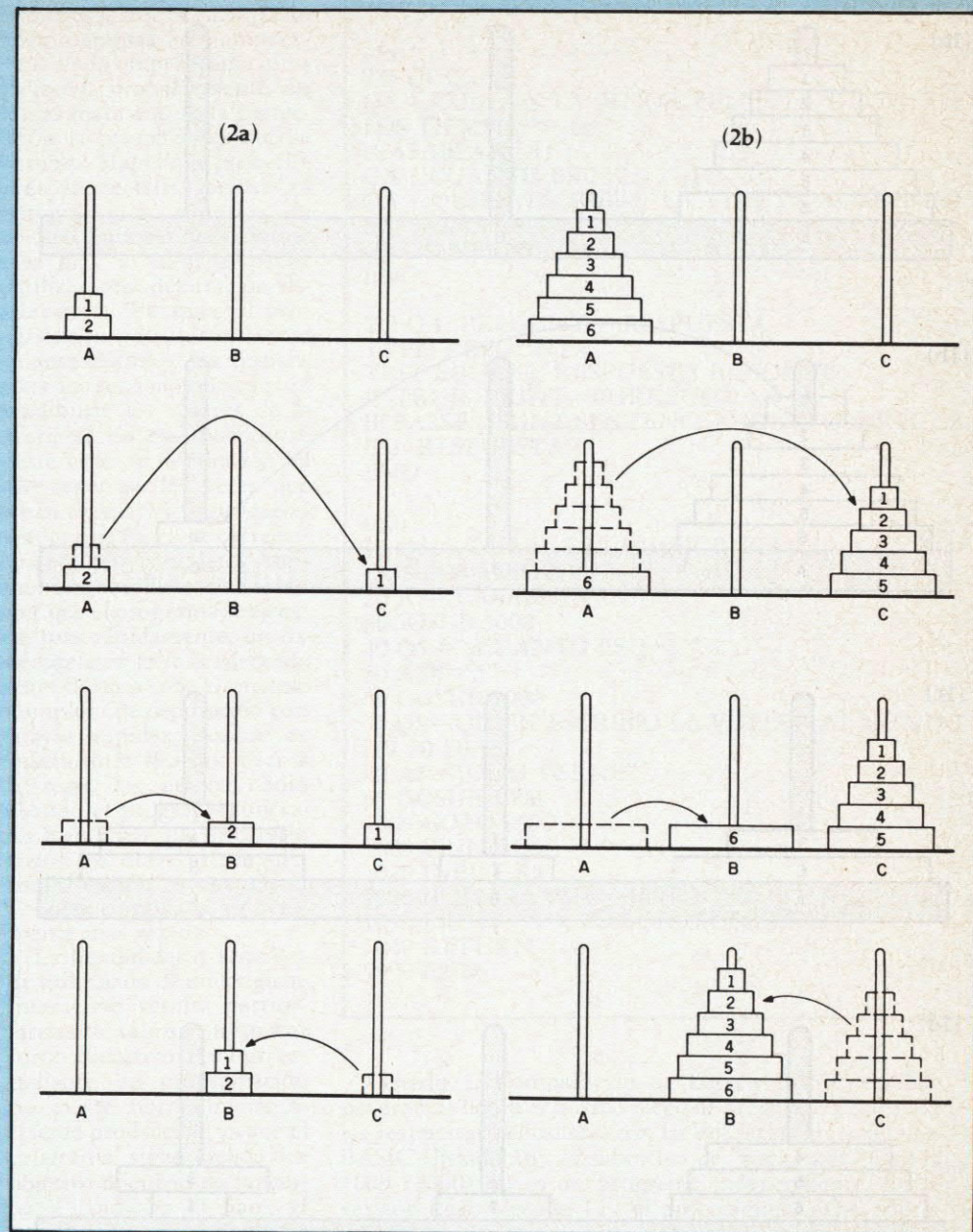
nes, como en el juego de las Torres de Hanoi, no se puede utilizar otra cosa que la recursión.

Hasta hace poco tiempo, la iteración era mucho más eficiente que la recursión, tanto en velocidad como en la utilización de la memoria. Un importante avance de las implementaciones recientes de Logo, incluidas las de los microordenadores Apple II y Texas Instruments TI-99/4A, es que "la recursión de cola" o recursión terminal es reconocida por el intérprete y tratada como si fuera una iteración. La recursión de cola es la situación en la cual el uso recursivo del procedimiento es la última cosa que hace dicho procedimiento. En general, sólo los programas con este tipo de recursividad pueden ser realizados tan fácilmente como si se tratara de iteraciones. El procedimiento HANOI, por ejemplo, no utiliza recursión de cola porque en su cuerpo existen dos llamadas recursivas, y sólo una está situada al final.

Logo posee procesamiento de listas

Todos los lenguajes de programación importantes disponen de algún medio para agrupar diversas piezas de información (números, por ejemplo), en una unidad de tamaño superior. En FORTRAN y BASIC, este mecanismo es la matriz o *array*. En Pascal y C, también se utilizan las matrices, junto con agrupaciones más complejas llamadas registro en el caso del Pascal, o estructura en el lenguaje C. En Logo, el principal mecanismo de agrupación se llama lista.

Las listas y las matrices tienen dos importantes diferencias. En primer lugar, las matrices tienen un tamaño fijo, mientras que las listas pueden agrandarse o dismi-



nir de tamaño a medida que se ejecuta el programa. (No existe un equivalente en Logo a la instrucción DIM del BASIC, utilizada para especificar el tamaño máximo de una matriz). La segunda diferencia es que las matrices deben de ser uniformes, es decir, es posible tener una matriz de 12 números o una matriz de cadenas, cada una de ellas con 23 caracteres, pero no se puede tener una mezcla de ambas. (Un registro en Pascal o una estructura en C pueden mezclar ambas, pero solo en un formato predeclarado). Cada elemen-

to de una lista Logo puede ser cualquier objeto Logo: un número, una palabra o incluso otra lista. He aquí algunos ejemplos de listas:

(VAINILLA, CHOCOLATE, MOKA)

(VAINILLA (CHOCOLATE A LA MENTA) (BATIDO DE FRESA))

(PLATANO 3.14159 (ROJO AZUL AMARILLO) 2.71828)

(SABORES (VAINILLA CHOCOLATE) TAMAÑOS

(GRANDE PEQUEÑO) OPCIONES ((BLANDO DURO) (CUCURUCHO ESPECIAL)))

El primero de estos ejemplos es una lista de tres palabras. La segunda es también una lista con tres miembros, pero el primero es una palabra, mientras que los otros son listas. El tercer ejemplo muestra que pueden incluirse números. El último ejemplo demuestra cómo una lista puede contener como elemento otra lista, que a su vez contiene una nueva lista.

El último ejemplo es un

Se llega a la solución dividiendo el problema en una serie de subproblemas más sencillos, todos los cuales pueden resolverse repitiendo una serie de movimientos. Primero, llevar todos excepto el de abajo a la tercera guía; a continuación llevar el que quedó a la guía de destino; y finalmente, llevar todos excepto el de abajo a la guía de destino.

```

TO HANOI: NUMERO :DESDE :HASTA : OTRA
IF :NUMERO = 0 (STOP)
HANOI :NUMERO — 1 :DESDE :HASTA :OTRA
PRINT (SENTENCE (LLEVAR DISCO) :NUMERO
(DESTE LA GUIA)
:DESDE (HASTA LA GUIA) :HASTA)
HANOI :NUMERO —1 :OTRA :HASTA :DESDE
END

```

Listado 2: Solución general al problema de las torres de Hanoi en Logo. El programa requiere cuatro entradas. La variable NUMERO le dice al programador cuantos discos hay; las otras tres entradas son los nombres de las guías. La instrucción IF detecta el subproblema trivial de tener que mover cero discos, caso en el que el programa no hace nada.

tipo especial de lista llamada Lista de Atributos. Si esta lista se asociara con el nombre HELADOS, la instrucción Logo

```
PRINT GPROP "HELADOS" TAMAÑOS
```

Imprimiría:

GRANDE PEQUEÑO

(GPROP viene de "GET PROProperty"). Las listas de atributos son una manera cómoda de agrupar información relacionada. Así por ejemplo, un programa relativo a un juego del tipo "guerra del espacio" en el que existen diversas naves espaciales, cada una de ellas podría tener una lista de atributos especiales. Estas propiedades podrían ser la posición de la nave, su velocidad, forma, energía que le queda y demás.

La razón de que algunos lenguajes sólo dispongan de las matrices es que al ser uniformes y de tamaño fijo, se manejan más eficientemente. Si el ordenador conoce cuál es la posición de comienzo de la matriz en la memoria, puede calcular fácilmente la posición del enésimo elemento, independientemente de los valores actuales que tenga la matriz en ese momento.

En una lista, en cambio, el tamaño de cada elemento es variable. Por consiguiente las listas han de almacenarse de alguna manera más complicada. Como resultado, para encontrar el decimocuarto elemento es preciso comenzar por el primero, calcular donde está el segundo luego el tercero, y así sucesivamente. Puesto que esta función es realizada por el intérprete de Logo, las listas no son difíciles de manejar por el programador, pero son algo más lentas que la búsqueda dentro de una matriz. Entre los lenguajes más importantes, LISP utiliza las listas de manera muy similar a como lo hace Logo. (De hecho, las estructuras de datos de Logo están basadas en las de LISP. El término LISP procede de *List Processing*). APL utiliza una estructura de datos que se parece a las listas en que no es de tamaño fijo, y a las matrices en que es de composición uniforme. En otras palabras, un vector APL puede agrandarse o acortarse, pero debe estar compuesto siempre de números o de caracteres exclusivamente. Pascal y C no tienen listas, pero disponen de variables punteros, que utilizadas con las estructuras o registros, permiten construir el equivalente a las listas. FORTRAN y BASIC no

disponen de asignación dinámica de memoria —no es posible aumentar el tamaño de una tabla durante la ejecución— por lo que no es posible disponer de listas en estos lenguajes.

En Logo las variables no tienen tipo

En BASIC, si se quiere que una variable contenga una cadena de caracteres, el nombre de la variable debe llevar al final un signo \$. Si no se utiliza el signo \$, la variable puede contener un número pero no una cadena de caracteres. (Algunas versiones de BASIC tienen un tercer tipo, una variable cuyo nombre termina en un símbolo de %, que está destinada a contener un entero). En Pascal y en C, el tipo de una variable debe hacerse explícito en una declaración. En FORTRAN las variables han de ser declaradas como en Pascal; si una variable no se declara, su tipo depende de la primera letra del nombre. Las letras I a N son indicativas de variables del tipo entero.

En Logo, como en LIST y APL, las variables no tienen tipo, cualquier variable puede tomar cualquier valor. La misma variable puede ser un entero en un punto del programa, y una cadena de caracteres (llamada palabra en Logo) más adelante.

Originalmente, la asignación de tipos a las variables no eran un asunto de la filosofía de diseño de un lenguaje. Esta cuestión se introdujo para facilitar la tarea de quienes tenían que escribir los compiladores. Como para sumar números enteros se utilizan, por ejemplo, distintas instrucciones de lenguaje de máquina, que para sumar números fraccionarios, es más fácil traducir al lenguaje de máquina la expresión "A + B" si se sabe previamente si A y B son enteros o no.

Recientemente, algunos diseñadores de lenguajes han adoptado la posición de que la asignación de tipos a las variables es algo positivo, aparte de las cuestiones relativas a su implementación,

ya que crea en el programador la disciplina de utilizar una variable para una cosa solamente. Al rechazar la asignación de tipos a las variables, los diseñadores de Logo no han pretendido, por supuesto, animar a la utilización gratuita de las mismas variables para diferentes propósitos; lo que han hecho, por el contrario, es construir un lenguaje basado en procedimientos, en el cual las variables están afectadas a un procedimiento en particular, y no disponibles en todo el programa. Con ello se promueve la misma disciplina de una manera diferente.

Como ejemplo en el que las variables con tipo son problemáticas de manejar, ciertos números introducidos por usuario, realice algún cálculo con ellos y repita el proceso hasta que el usuario señale que ya no hay más problemas a resolver.

Este programa se ha escrito de manera que el usuario pueda introducir la palabra BASTA cuando no quedan más números por añadir. En un lenguaje que utiliza variables con tipo fijo, los números tienen que leerse en variables de tipo numérico, no en variables de tipo cadena de caracteres. Si se introduce una palabra no numérica, se produce un error. Por tanto, un programa en FORTRAN se verá obligado a sacar un mensaje al usuario que diga algo parecido a "para indicar el final, teclee 9999". En Pascal tendríamos parecidas dificultades.

Logo es extensible

Todos los lenguajes de programación tienen ciertas operaciones incorporadas que llamamos primitivas. La gran mayoría de los lenguajes, por ejemplo, incluyen operaciones aritméticas con números, y algún medio de escribir los resultados. Los lenguajes basados en procedimientos permiten al programador crear una serie de operaciones, que amplían la capacidad del lenguaje. En este sentido, se dice que estos lenguajes son extensibles.

Pero el término "extensible" es utilizado por los diseñadores de lenguajes en un sentido especial.

Un lenguaje extensible es aquel en el cual los procedimientos definidos por el usuario son "como" los procedimientos primitivos. Se trata en parte de una cuestión de notación y en parte de una cuestión de auténtica potencia. En muchos lenguajes las operaciones aritméticas primitivas pueden aplicarse a diferentes tipos de variables (enteros y reales, por ejemplo), obteniéndose los resultados apropiados para cada tipo. Sin embargo, son muchos en los que, los procedimientos definidos por el usuario han de especificar en su definición el tipo particular de variable al que se aplican. Esta restricción viola el principio de extensibilidad.

Los lenguajes extensibles son particularmente valiosos para la enseñanza porque el profesor puede ir creando ampliaciones del lenguaje y enseñarlas como si se tratase de auténticas primitivas. LISP, Logo, APL y FORTH son extensibles, con algunas restricciones de menor orden en determinados casos. Logo viola el principio puro de extensibilidad, por ejemplo, puesto que algunas de sus operaciones aritméticas primitivas están representadas de forma infija (con el símbolo de operación entre los dos operandos, como en $3 + 2$), mientras que los procedimientos definidos por el usuario pueden representarse sólo en forma prefija (esto es con el símbolo de operación precediendo a los operandos, como en $SUM\ 3\ 2$). Casi todas las primitivas de Logo se utilizan en forma prefija.

Como ejemplo del uso de la extensibilidad de Logo, muchas versiones no tienen procedimientos primitivos para hacer iteraciones equivalentes a las estructuras FOR, DO, o WHILE de otros lenguajes. Sin embargo, es muy fácil definir estos procedimientos, si es que se desea, de manera que sean sintácticamente similares a

la instrucción IF, que es una primitiva de Logo.

Logo como lenguaje de aprendizaje

Entre los lenguajes que hemos citado, se habrá podido observar que se han hecho dos grupos: Logo, LISP, APL son interpretados orientados a listas, y sin tipos fijos para las variables. Pascal y C son compilados, orientados a matrices, y con declaración de tipos. (Todos ellos son basados en procedimientos por definición). De esta clasificación son culpables muy diversas circunstancias: accidentes históricos, intentos de facilitar la implementación y filosofías de diseño. Así por ejemplo, C y Pascal son muy similares porque ambos se derivan de un lenguaje anterior, ALGOL, que inauguró un estilo que siguieron otros muchos lenguajes.

Los compiladores se adaptan mejor a los lenguajes con variables de tipo declarado, mientras que a los intérpretes les sucede lo contrario. En el diseño de los lenguajes orientados a listas intervinieron investigadores que eran básicamente matemáticos, más que programadores de ordenador.

Dentro de cada grupo, sin embargo, las diferencias tienden a reflejar la utilización particular que el diseñador pensaba dar al lenguaje. Así por ejemplo, las diferencias entre C y Pascal estriban en que C fue diseñado como lenguaje de programación de sistemas. En el grupo de los lenguajes orientados a listas, LISP se desarrolló para la investigación sobre inteligencia artificial, mientras que APL fue desarrollado para enseñar álgebra y otros aspectos de las matemáticas, como el cálculo infinitesimal, relacionados con el álgebra. Logo, sin embargo, se desarrolló como lenguaje de aprendizaje, no como una rama de resolución de problemas. Logo se dirige particularmente a estudiantes más jóvenes que los destinatarios de APL, aunque también ha sido utilizado con éxito por estudiantes univer-

sitarios de física del MIT (Massachusetts Institute of Technology).

Desde el punto de vista "puramente" científico, Logo es LISP. Quienes han desarrollado Logo, de hecho, han sido investigadores en inteligencia artificial, para los cuales LISP es su segunda naturaleza. Las diferencias entre los dos lenguajes se basan en el intento específico de hacer que Logo resulte especialmente útil como lenguaje de aprendizaje. A continuación vamos a describir precisamente las propiedades especiales de Logo desde este punto de vista.

Logo está especialmente adaptado para algunas aplicaciones interesantes

Probablemente el aspecto más famoso de Logo es la idea de la geometría de la tortuga. Este modo de abordar los gráficos con ordenador ha sido incorporado luego a otros lenguajes, como Pascal y PILOT, pero su origen es el lenguaje Logo.

Muchas modalidades de gráficos con ordenador se basan en el sistema de coordenadas cartesianas (el sistema "X, Y" de representación de funciones que se aprende en la escuela; ver figura 3). En este esquema cada línea que se quiere dibujar se especifica en términos de las posiciones de sus extremos, relativas a un sistema de coordenadas fijo. Con las coordenadas cartesianas, no es demasiado difícil dibujar un cuadrado con los lados paralelos a los ejes en una posición conocida, pero si el cuadrado está girado, sus coordenadas hay que calcularlas utilizando trigonometría. La potencia de la geometría de la tortuga reside en que las líneas se describen, no en términos de posiciones absolutas dentro de un sistema coordenado, sino relativas a la posición y dirección de la tortuga, un animal conceptual que se mueve por la pantalla del televisor. En este tema, no se sabe dónde comienza o termina la tortuga, sino cómo se mueve y en qué dirección:

TO CUADRADO: LADO
REPEAT 4 (FORWARD:
LADO RIGHT 90)
END

Los realmente interesantes del asunto es que utilizando este poderoso instrumento, los principiantes pueden programar fácilmente gráficos en su primer contacto con el ordenador.

En el pasado, la programación de ordenadores atraía solamente a un pequeño número de personas porque había una auténtica carencia de prácticas que fuesen interesantes y suficientemente fáciles para principiantes. Los cursos de programación tradicionales estaban basados en buena parte en cuestiones algebraicas ("escribir un programa para resolver ecuaciones cuadráticas"), y en consecuencia, no atraían a la gente que no estaba interesada en el programa tradicional de matemáticas.

La geometría de la tortuga no es la única aplicación especial que puede hacerse con Logo. Otra aplicación importante es el tratamiento del lenguaje. Letras, palabras y frases, son la jerarquía natural de los objetos de Logo. (En muchos lenguajes de programación, por el contrario, una sentencia no es una lista de palabras sino una cadena de caracteres. Si se quiere manejar las palabras de la frase, es preciso escribir un complicado programa sólo para ir buscando los espacios en la cadena que permitan dividir la frase en palabras). Como ejemplo sencillo, el listado 4 es un programa en Logo que traduce una frase en "Latín casero".

PALABRALAT se utiliza como un subprocedimiento para traducir una palabra individual en base a este regla: si la palabra comienza con vocal, se añade AY. Si no, pasar la primera letra al final y volver a probar.

En el programa WORD y SENTENCE son procedimientos para encadenar dos objetos y formar uno mayor. FIRST y BUTFIRST, por el contrario, separan un objeto en sus partes componentes. El procedimiento primitivo FIRST aplicado a una sen-

tencia devuelve la primera palabra de dicha sentencia. Cuando se aplica a una palabra devuelve la primera letra. Ningún otro lenguaje de programación incorpora de manera tan nítida esdta jerarquía de objetos tan propia del lenguaje humano.

Logo es agradable de utilizar

Un lenguaje para aprendices debe estar diseñado de manera que se ocupe de problemas, que quizá son de menor importancia cuando se trata de lenguajes para programadores experimentados. Así por ejemplo, cuando se comete un error, es muy deseable obtener un mensaje detallado y orientativo. Los lenguajes que comunican cosas como SYNTAX ERROR o ERROR NUMBER 259 no ayudan demasiado al principiante. Logo dispone de mensajes como:

```
+ DOESN'T LIKE HELLO
AS INPUT
(HELLO NO LE GUSTA
COMO OPERANDO)
```

Este mensaje quiere decir que se intentó sumar algo no numérico, la palabra HELLO, a otra cosa. Cuando se presenta este mensaje. I DON'T KNOW HOW TO FRIST

(NO SE QUE ES FRIST)

Es que se ha utilizado un procedimiento, FRIST, que no se había definido. El mensaje

```
NOT ENOUGH INPUTS
TO MAKE
(MAKE NECESITA MAS
PARAMETROS)
```

significa que el procedimiento MAKE necesita más entradas y sólo se le ha dado una. Si se produce el error durante la ejecución de un procedimiento, Logo imprime también el nombre del procedimiento y la línea que contiene el error.

Desde los primeros tiempos (en 1974), los estudiantes de programación se han sentido confusos ante instrucciones muy corrientes en programación como $X = X+1$, una asignación utilizada muy frecuentemente que parece ir en contra de la intuición algebraica. El signo $:$ = que utiliza el lenguaje Pascal en lugar del signo $=$ a secas, representa sin duda una mejora, y la flecha utilizada por APL es incluso mejor. Pese a esto la notación no deja patente que $X - 3$, que aparentemente son muy similares. En Logo la asignación se hace de este modo: MAKE "X:X+1. Aunque menos conciso que la asignación a base de un sólo carácter, la palabra MAKE evoca de una vivida la noción de que algo se está cambiando, y no que se realice un cálculo exclusivamente.

Son muchos los aspectos en los cuales Logo hace explícitamente las cosas que otros lenguajes dejan implícitas. Así por ejemplo, Logo utiliza los dos puntos para expresar que "quiero el valor de esa variable", la misma palabra sin estar precedida de los dos puntos designa un procedimiento. En LISP, los notorios paréntesis hacen posible la distinción entre llamadas a procedimientos simplemente prohíben utili-

zar la misma palabra para ambas cosas. (Esta solución sería incómoda en Logo porque muchas palabras como WORD no son sólo populares nombres de variables, sino también nombres de procedimientos primitivos).

En cualquier caso de acuerdo con la filosofía de diseño de Logo, la notación de los dos puntos es buena, aparte de su necesidad técnica, porque llama la atención sobre el hecho de que el valor de una variable es diferente de su nombre; también resalta que una variable es diferente de un procedimiento. Por ejemplo, en la situación $X = X+1$ las dos apariciones de un X idéntico tienen diferentes significados. El segundo representa el antiguo valor de X, mientras que el primero denomina la variable a la que se va a dar nuevo valor. En la versión de Logo, estos dos significados quedan diferenciados por la notación. Al primero se le llama "X; al segundo, :X.

En cualquier caso de acuerdo con la filosofía de diseño de Logo, la notación de los dos puntos es buena, aparte de su necesidad técnica, porque llama la atención sobre el hecho de que el valor de una variable es diferente de su nombre; también resalta que una variable es diferente de un procedimiento. Por ejemplo, en la situación $X = X+1$ las dos apariciones de un X idéntico tienen diferentes significados. El segundo representa el antiguo valor de X, mientras que el primero denomina la variable a la que se va a dar nuevo valor. En la versión de Logo, estos dos significados quedan diferenciados por la notación. Al primero se le llama "X; el segundo, :X.

Otro ejemplo de una distinción que se hace explícitamente en Logo y no en algunos otros lenguajes, es la división de los procedimientos en comandos y operaciones. Una operación es un procedimiento que calcula cierto valor que resulta ser la salida del procedimiento. Por ejemplo, las operaciones aritméticas pertenecen a esta categoría. Un comando no produce una salida, pero en

cambio tiene un efecto: imprime algo, desplaza la tortuga, o cambia el valor de una variable.

En Pascal se hace la misma distinción: las operaciones se llaman funciones y los comandos procedimientos. FORTRAN los llama funciones y subrutinas. LISP, APL y C, sin embargo, son menos exigentes. C trata todos los procedimientos como operaciones, pero permite que una operación se utilice como si fuera un comando. En ese caso, se ignora el resultado de la operación. En LISP y APL, el resultado de tal operación de "nivel superior" se imprime. (En LISP, todo procedimiento tiene una salida y todo comando de nivel superior imprime algo. En APL, algunos procedimientos no tienen salida y, por consiguiente, no imprimen nada). En Logo, utilizar una operación sin un comando es considerado un error; si desea que algo se imprima debe utilizar el comando PRINT.

El uso de aritmética infija en Logo es una concesión a los hábitos de los usuarios. Todos los demás procedimientos Logo se utilizan en forma prefija, con el nombre del procedimiento antes de sus parámetros o entradas. Las operaciones aritméticas pueden utilizarse también en forma prefija. Las dos expresiones de Logo, $3+2$ y SUM 3 2, son equivalentes.

La forma infija parece más natural a la gente acostumbrada a expresarse aritméticamente al margen del entorno de Logo. La forma prefija, sin embargo, es mejor por muchas razones. Por ejemplo, elimina la necesidad de un criterio de precedencia en las operaciones (esto es, que se haga antes la división que la suma, etcétera). Asimismo, elimina la necesidad de los paréntesis para indicar agrupamientos. En LISP sólo se utilizan formas prefijas.

Otro aspecto de Logo en cuanto a su comodidad de utilización, es su facilidad para la definición interactiva de procedimientos. Las primeras versiones de Logo utilizaban una técnica de nume-

```
TO LAZOSUMA
PRINT (TECLEE LOS DOS NUMEROS A SUMAR)
MAKE "NUMEROS READLIST
IF FIRST :NUMEROS = "BASTA (STOP)
PRINT SENTENCE (LA SUMA ES) (FIRST :NUMEROS)+(LAST :NUMEROS)
LAZOSUMA
END
```

Listado 3: Las variables en Logo no tienen tipo prefijado. La variable NUMEROS contiene lo que el usuario introduzca. Primero, es examinada como una lista de palabras, para ver si contiene el valor BASTA; luego se utiliza como una lista de números que hay que sumar.

ración de líneas: dentro de cada procedimiento se numeraban las líneas y podían ser reemplazadas del mismo modo que se hace con las líneas de BASIC. Las implementaciones actuales de Logo utilizan un editor en pantalla, en el cual se utilizan caracteres especiales de control para mover el cursor por la pantalla y cambiar caracteres individuales en cualquier sitio de la definición del procedimiento.

Logo no tiene umbral ni techo

Esto significa que Logo es suficientemente sencillo para que lo pueda utilizar cualquiera, pero es también lo suficientemente potente para cualquier clase de proyecto; no es un lenguaje de "juguete". Logo es bien conocido como lenguaje para chavales de la escuela elemental, pero está diseñado para principiantes de cualquier edad y cualquier nivel de sofisticación.

¿Cuál puede ser la edad mínima del aprendiz de Logo? Los críos muy jóvenes podrían tener problemas con las teclas y con la escritura correcta de los nombres de los procedimientos. Por eso, hace algunos años *Radia Perlman* construyó en el MIT una serie de teclados especiales con grandes botones etiquetados mediante dibujos en lugar de palabras. Con este hardware especial, consiguió enseñar las ideas de la geometría de la tortuga a niños de cuatro años. Este proyecto incorporaba incluso la idea de procedimientos, con botones llamados "comienza a recordar" y "deja de recordar" para delimitar la definición de un procedimiento, y otro llamado "hazlo" para ejecutar el procedimiento. Utilizando botones de diferentes colores, podían denominarse múltiples procedimientos.

Y por el otro lado, ¿hasta qué edades puede resultar útil la enseñanza de Logo? Los profesores *Harold Abelson* y *Andrea diSessa* han utilizado Logo para enseñar física a los alumnos preuni-

versitarios del MIT. Utilizaban programas de simulación en Logo, para demostrar no sólo los mecanismos newtonianos sencillos, sino incluso la teoría general de la relatividad. Su libro "Turtle Geometry"; The Computer as a Medium for Exploring Mathematics" (Cambridge, MA: MIT Press, 1981), explica su sistema, que puede también ser utilizado con éxito por estudiantes universitarios.

Logo se ha utilizado también para un grupo especial de alumnos: los disminuidos de diversos tipos. En el pasado, muchos niños de inteligencia normal o superior, pero con su capacidad de comunicación disminuida han sido diagnosticados de retrasados. Con estos chicos los ordenadores pueden utilizarse tanto como "prótesis" de comunicación y como campo de interés en el que puede exhibir autonomía en la persecución de sus propios objetivos. El uso de Logo en la educación de disminuidos se examina en el libro del doctor *E. Paul Goldenberg*: "Special Technology for Special Children".

Otros lenguajes que fueron diseñados pensando en los estudiantes fueron BASIC, Pascal y APL. (Omito PILOT, porque fue diseñado no tanto para los estudiantes como para los profesores; en su diseño original se suponía que los estudiantes utilizarían programas de formación con ayuda de ordenador escritos en PILOT, más que el PILOT en sí). ¿Cómo pueden compararse esos lenguajes con Logo en cuanto a su aplicabilidad en educación?

BASIC fue concebido como una modificación de FORTRAN para principiantes. El avance más importante de BASIC lo constituyó su método interactivo. Esto fue mucho más importante que lo que ahora pueda parecer, cuando la gente ya está acostumbrada a que todos los ordenadores personales incorporen esta posibilidad. En los primeros tiempos de BASIC, los únicos ordenadores disponibles eran los grandes y caros. Aunque el tiempo comparti-

do, que permitía a diversas personas utilizar simultáneamente un gran ordenador, se había inventado recientemente, eran muchos se puede pintar cualquier cosa.

Para el uso que se pensó, APL es un lenguaje muy potente. Muchos cálculos que requieren bucles iterativos y variables auxiliares en otros lenguajes, pueden hacerse en una sola instrucción en APL. Mucha gente ve esta potencia principalmente como una cuestión de compacidad; APL es famoso (o por lo menos notorio) por sus programas de una sola línea. La cualidad real del diseño de APL es que permite que la atención del estudiante se centre sobre el problema de matemáticas, más que sobre las necesidades del ordenador. APL se diseñó para ser utilizado no en un curso especial de programación o como un bloque especial en otro curso de matemáticas, sino para utilizarlo con toda naturalidad en un curso de álgebra, como si se tratase de utilizar un calculador.

Los objetivos de Logo son diferentes de los de estos lenguajes. No se trata de que sea una fácil introducción a nada, ni tampoco es un lenguaje específicamente para los estudiantes de informática, ni una herramienta para enseñar el mismo programa de matemáticas que ya se está enseñando con otros medios. Por el contrario, es una puerta abierta al territorio de los ordenadores, como

un objeto para la exploración intelectual. Volviendo a lo dicho al comienzo del artículo, Logo es para aprender a aprender.

¿Por qué Logo?

En su libro "Midsdorms: Children, Computers, and Powerful Ideas" (New York: Basic Books, 1980), *Seymour Papert* dice: "no se puede decir que la imagen de la relación de un niño con un ordenador que voy a desarrollar aquí vaya más allá de lo que es común en la escuela de hoy en día. Mi esquema no va más allá: camina en la dirección opuesta". Logo no es sólo un lenguaje de programación; es también una filosofía de educación. El libro de Papert es la mejor explicación de esa filosofía. Lo que sigue no es más que un breve resumen.

Un niño aprende parcialmente captando hechos y estrategias específicas. Gran parte de la educación formal existente versa sobre hechos y capacidades: lectura, escritura, la tabla de multiplicación, etcétera. Pero un tipo de aprendizaje más profundo es el aprendizaje del modo de aprender en sí, que implica la construcción de modelos mentales del mundo, de uno mismo, y del proceso de aprendizaje. Estos modelos pueden empezar de una manera casi por azar, pero enseguida un buen

```
TO LATIN :FRASE
IF EMPTY :FRASE (OUTPUT ())
OUTPUT SENTENCE (PALABRALAT FIRST :FRASE) (LATIN BUTFIRST: FRASE)
END
```

```
TO PALABRALAT :PALABRA
IF VOCAL FIRST :PALABRA (OUTPUT WORD :PALABRA "AY")
OUTPUT PALABRALAT WORD BUTFIRST :PALABRA FIRST :PALABRA
END
```

```
TO VOCAL :LETRA
OUTPUT MEMBERP :LETRA (A E I O U)
END
```

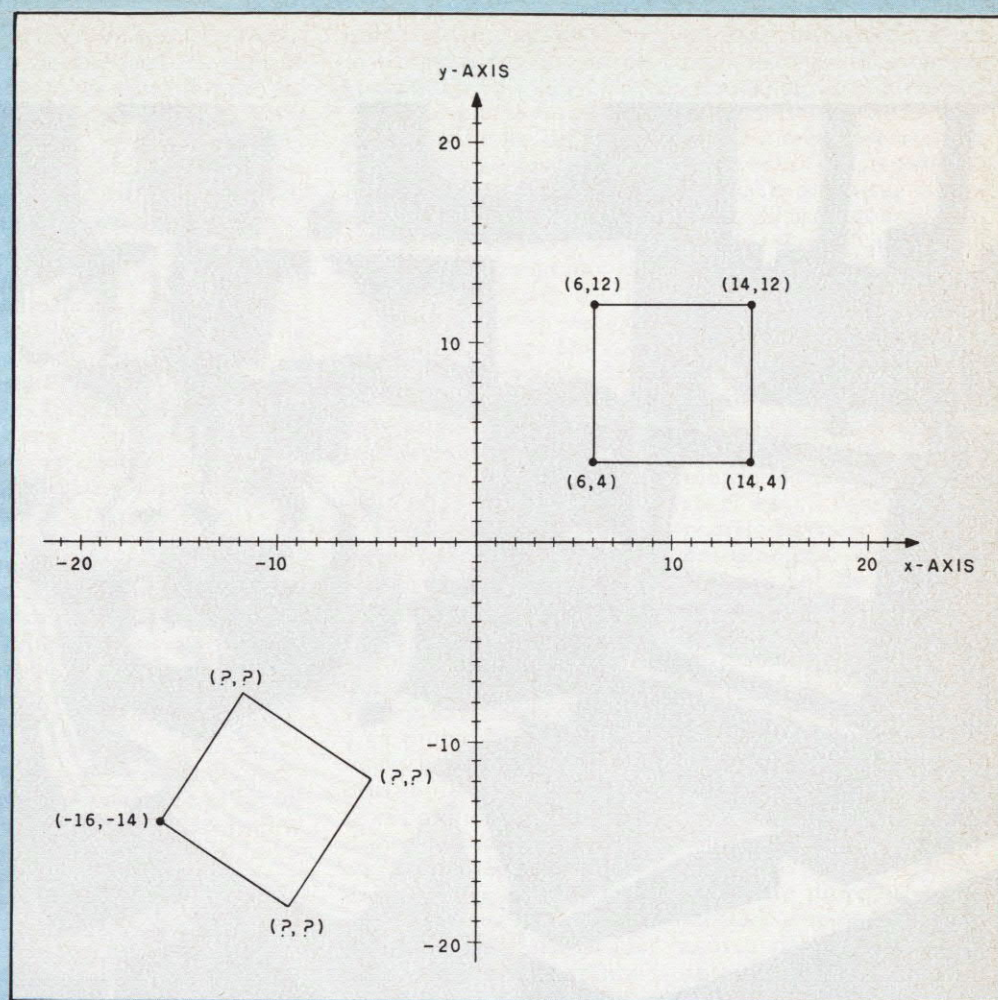
Listado 4: Traducción a Latín casero con Logo.

aprendiz desarrolla estrategias para una exploración totalmente orientada hacia sus propósitos específicos. Cuanto más va aprendiendo, mejor es el modelo de aprendizaje, y más capacidad de aprendizaje se desarrolla".

En este proceso de crecimiento, realmente no importa el aspecto particular del mundo que se explore. En la introducción de su libro, Papert menciona que a los dos años él estaba enamorado de las cajas de cambios de los automóviles. Cuando yo estaba en cuarto curso, era un completo enamorado del hipnotismo. La cuestión sobre la utilización de ordenadores en la educación no es que todo el mundo deba saber sobre ordenadores sino simplemente que para mucha gente, la programación de ordenadores pueda ser un terreno de su proceso general, debe ser una herramienta para muy diversos intereses. Puede dibujar cuadros, hacer música, escribir cuentos, o mover robots.

"Yo quiero un trabajo como programador de ordenadores. ¿Para qué voy a aprender Logo y no algo más útil como por ejemplo COBOL?" Esta es una pregunta corriente. Y existen dos posibles respuestas a ella. La primera es que Logo, como se explicó antes, está diseñado para explicar muchas de las ideas fundamentales sobre la programación de ordenadores. Quien aprende Logo tendrá una idea muy clara de la naturaleza de las variables, procedimientos, y muchas otras construcciones de la programación. Por eso, Logo puede ser una base mejor, incluso para luego aprender COBOL, que si se empezara por el COBOL mismo. Pero la segunda respuesta es que el propósito de Logo no es formar programadores de ordenador. Logo no está para reemplazar a los otros lenguajes de programación.

Logo se asocia generalmente con los niños porque mucha gente tiene un modelo del proceso de aprendizaje en que los niños aprenden pero los adultos no. Este mo-



delo es desafortunado. Logo puede ser útil para gente de cualquier edad, pero será útil si la aproximación se hace de modo lúdico y exploratorio.

Es importante distinguir entre el lenguaje Logo y cualquier implementación particular del mismo. Ciertas cosas no pueden hacerse en las versiones de Logo de Apple o Texas Instruments, simplemente porque sus máquinas no son lo suficientemente grandes y rápidas o porque la implementación del lenguaje no incluye algunas capacidades. Así por ejemplo, ninguna versión de Logo para microordenador tiene un buen procedimiento de almacenamiento de datos en disco, aunque todas ellas lo pueden hacer de alguna manera.

El intérprete de Logo cabe con dificultad en un *Apple II* con 64 Kbytes, y la implementación tiende a favore-

cer lo que se precisa para las actividades educativas, y no aquello que es preciso para un proceso de datos práctico. Pero en principio, Logo es un buen lenguaje para desarrollar cualquier clase de aplicación, gracias a su depuración interactiva y su estilo orientado a los procedimientos.

Si quiere hacer un programa de video-juego, probablemente se ejecutará demasiado lentamente en el Logo de *Apple*, aunque resulte sencillo de hacer. Pero quizá resultara interesante desarrollarlo en Logo, probar con él diferentes ideas para el juego en un entorno que permite una rápida y sencilla modificación del programa, y escribirlo de nuevo en cualquier otro lenguaje. La ventaja de Logo puede ser descrita parcialmente en términos puramente técnicos como "interactivo". Otra ma-

Figura 3: Dificultades de la representación gráfica en coordenadas cartesianas. es fácil dibujar un cuadrado de lados paralelos a los ejes, pero para cualquier otra orientación se necesita acudir a la trigonometría.

nera de verlo, sin embargo, es que Logo anima a probar todas las posibilidades que se necesitan para conseguir el mejor juego posible. Ahora bien, si todo lo que usted quiere es hacer una copia exacta del juego de los Asteroides, las ventajas de Logo son menos importantes.

En suma: Logo es un lenguaje del tipo de LISP y un laboratorio para aprender de manera divertida cómo se debe aprender.

Brian Harvey
©Byte/Ordenador Popular



LOGO EN LA ESCUELA

En sus 15 años de historia, el lenguaje de ordenador Logo se ha utilizado en diversas actividades educativas y de investigación. Desde estudiantes de preescolar a universitarios, y desde alumnos con diversos problemas físicos, mentales o emocionales a estudiantes con excepcional capacidad para la ciencia y las matemáticas, han sido usuarios de Logo. En actividades educativas, Logo se ha utilizado para:

- Proporcionar un entorno para la enseñanza experimental de las matemáticas.

- Promover el desarrollo de la capacidad de resolución de problemas.

- Servir como lenguaje de programación introductorio que ayuda a los estudiantes a aprender los principios de la programación estructurada.

- Servir como vehículo de difusión para la enseñanza de los principios básicos del ordenador, ayudando a los estudiantes a desarrollar un sentido del control personal sobre el mismo.

- Ayudar al aprendizaje a los estudiantes que por una razón u otra no han tenido

éxito con los medios tradicionales.

- Proporcionar el entorno básico de aprendizaje para una serie de materias entre las que están música, literatura, artes plásticas, física, biología y matemáticas.

- Crear un fundamento para un tipo de escuela totalmente nueva basada en las ideas de Jean Piaget sobre el aprendizaje y la enseñanza, utilizando ordenadores como herramientas de uso general para facilitar el aprendizaje.

En cada experiencia con

Logo en la Escuela, el interés se ha centrado en determinados objetivos a expensas de otros. Este artículo habla de cuatro proyectos diferentes con Logo y describe los planteamientos y los objetivos de cada uno de ellos, además de los resultados conocidos. En algunos casos hago referencia a los informes ya publicados. Cuando no existen informes publicados me remito a la información obtenida en visitas y entrevistas con las personas directamente implicadas. Cada uno de los proyectos ha tenido muchas dimensiones

que no se incluyen aquí por razones de espacio. Cada proyecto ha tenido un tipo diferente de población estudiantil, diferentes objetivos, y diferentes tipos de resultados.

El Proyecto Logo de Edimburgo. Departamento de Inteligencia Artificial de la Universidad de Edimburgo, en Escocia, se dirigía a muchachos de 12 y 13 años de edad asistentes a una escuela privada próxima a la universidad. Se centraba en la utilización de Logo para crear un entorno para aprender a pensar matemáticamente y a desarrollar nuevos métodos de enseñanza de los contenidos de las matemáticas escolares convencionales.

El Proyecto Logo de Brookline, realizado como una colaboración entre el Grupo Logo del MIT y la escuela pública de Brookline, Massachusetts, tuvo dos fases muy diferentes. En la primera se preparó un laboratorio con cuatro ordenadores. El interés de la investigación se centró en observar y documentar lo que el grupo de los estudiantes de sexto grado aprendían realmente, más que en evaluar si alcanzaban o no unos objetivos previamente planificados.

La segunda fase del Proyecto Logo de Brookline consistió en introducir ordenadores en aulas correspondientes a los grados cuarto al octavo durante varias semanas simultáneamente. Este proyecto hizo hincapié en el desarrollo de las diversas asignaturas de los programas, para apoyar el aprendizaje de Logo como una actividad con múltiples usos en una clase.

El Proyecto Ordenador en la Escuela dirigido por la Academia de Ciencias de Nueva York en colaboración con las Escuelas Públicas de esta ciudad, proporcionó formación para profesores de bachillerato con la intención de que utilizarán Logo como

una herramienta permanente de sus clases. El enfoque básico de este proyecto ha sido la implementación en la escuela, la formación y el apoyo a los profesores para asegurar una utilización con éxito de Logo en las clases.

El Proyecto Logo de la Escuela de Lamplighter llevado a cabo en una escuela privada en Dallas, Texas, con estudiantes de 3 a 9 años es el proyecto más ambicioso hasta la fecha. Dirigido como un esfuerzo conjunto con la escuela, el Grupo Logo del MIT, y la compañía *Texas Instruments*, el proyecto pretendía dotar a la escuela con un hardware suficiente como para que la posibilidad de acceso a los ordenadores, no fuese una limitación en lo que los estudiantes y profesores, desde el parvulario hasta el cuarto grado. Eventualmente se esperaba que el proyecto mejorase el aprendizaje en diversas áreas, en la medidas que facilitaba la utilización del ordenador como una herramienta de apoyo para todos los programas. La escuela de Lamplighter sirvió también como una primera puesta a prueba para el desarrollo de la implementación que *Texas Instruments* había hecho de Logo.

El Proyecto Logo de Edimburgo

El objetivo del proyecto Logo de Edimburgo consistía en descubrir si los estudiantes "... experimentaban algún cambio en sus destrezas matemáticas y en su opinión acerca de las mismas, cuando abordaban los problemas matemáticos mediante la programación con Logo". Este es el propósito que literalmente figura en el informe "Teaching Mathematics Through Logo Programming: An Evaluation Study, by Howe, Oshea y Plane. Los estudiantes fueron un grupo de 11 chicos de sexto grado de la escuela privada George Herriot School, próxima a la universidad. Fueron seleccionados del

grupo con menor nivel en matemáticas de entre los de la escuela.

El proyecto ha durado dos años, durante los cuales los estudiantes asistieron a un laboratorio Logo en la Universidad. Durante el primer año, los estudiantes trabajaron con un programa de ejercicios graduales con el fin de aprender los elementos básicos de Logo. Durante el segundo año hicieron ejercicios especiales de Logo, diseñados para enseñar aspectos seleccionados de sus programas normales de matemáticas.

El proyecto estaba altamente estructurado en diversos aspectos. Las experiencias de aprendizaje de los estudiantes se estructuraban por medio de ejercicios planificados que debían realizar en orden y cada uno a su propio ritmo. De este modo, los investigadores podían controlar efectivamente el progreso de cada estudiante. Durante el segundo año del proyecto, las actividades con Logo se dirigieron hacia algunos puntos del programa de matemáticas tales como áreas de rectángulos, factores y múltiplos, números positivos y negativos y dibujo de gráficas en el plano coordenado.

El aspecto investigador del proyecto estaba también cuidadosamente estructurado. Antes y después del proyecto los estudiantes completaron una serie standard de pruebas de matemáticas. Su progreso era comparado con el de un grupo de control (extraído de muchos con un nivel en matemáticas inmediatamente superior). Ambos grupos de chicos, así como sus profesores, recibieron también unos cuestionarios pensados para medir sus actitudes hacia las matemáticas. Se observó gran cuidado en vigilar que el diseño de la investigación se mantuviera a lo largo de toda la experiencia.

Los resultados publicados relativos a los logros de los estudiantes no fueron excesivamente sorprendentes. Después de dos años, el gru-

po experimental mejoró un poco más que el grupo de control, de acuerdo con el test de "matemáticas básicas". En relación con el test "destrezas matemáticas" sucedió exactamente lo contrario. El hallazgo más interesante se produjo en relación con algunas apreciaciones de los profesores sobre los estudiantes que habían tomado parte en las clases de Logo se les notaba más dispuestos a "la argumentación sobre cuestiones de matemáticas" así como "a explicar con claridad sus dificultades en el tema". Este resultado podría depender tanto del procedimiento de enseñanza utilizado por los profesores de Logo, comparado con los que utilizaban los profesores de matemáticas en las clases— y de la asistencia personalizada que recibieron los estudiantes de Logo, como de las actividades en Logo como tales.

Diversas conversaciones con algunas de las personas implicadas indican que durante este proyecto se han acopiado interesantes datos acerca de la manera de aprender de los estudiantes. Desgraciadamente muy poca de esa información ha sido analizada o publicada. Para la gente interesada en la enseñanza de Logo, los resultados más tangibles del proyecto pueden ser el conjunto de prácticas desalladas para el aprendizaje de los conceptos de Logo y sus aplicaciones matemáticas. Estas prácticas representan un conjunto muy útil de ideas para la enseñanza de Logo, incluso si no se utiliza en la forma estrictamente secuencial en que se concibieron. El grupo Logo de Edimburgo dispone de copias de las prácticas utilizadas por los estudiantes en los dos años que duró el proyecto.

El grupo Logo de Edimburgo ha estado también implicado en varios proyectos educativos más. En uno de ellos, profesores de los estudiantes que no eran especialistas en matemáticas recibieron formación en Logo con la intención de estudiar

cómo les afectaría para la enseñanza de las matemáticas. En otro proyecto actualmente en desarrollo, se han instalado ordenadores en diversas escuelas, de manera que los profesores que han seguido un curso de formación Logo puedan a su vez enseñarlo o utilizarlo en sus asignaturas. Este proyecto está orientado a obtener resultados esclarecedores acerca del impacto de Logo en la mejora del rendimiento de la clase en matemáticas.

Para llevar a cabo el estudio actual, el grupo Logo de Edimburgo implementó una versión de Logo en el ordenador *Terak*, un sistema basado en un LSI-11 con gráficos de alta resolución. En el grupo de Edimburgo están disponibles los discos de esta versión de Logo. En la actualidad, están en desarrollo otra serie de proyectos de implementación de Logo para otros sistemas de microordenadores ampliamente difundidos en Gran Bretaña.

El Proyecto Logo de Brookline

El primer Proyecto Logo de Brookline, financiado por la **National Science Foundation** y dirigido por el grupo Logo del MIT en colaboración con una escuela pública de Brookline en Massachusetts, tenía unos propósitos muy diferentes y obtuvo también diferentes resultados. En este caso, 50 estudiantes de sexto grado tuvieron la oportunidad de aprender Logo en un laboratorio instalado dentro de la propia escuela. Se seleccionó para su estudio el trabajo de 16 estudiantes, que representaban un abanico completo de capacidades e intereses académicos.

La experiencia completa de aprendizaje de Logo de estos estudiantes fue conscientemente controlada y analizada, controlando y documentando qué aprendían los estudiantes, qué estilos de aprendizajes utilizaban y qué tipo de elecciones hacían. Al comienzo del pro-

yecto se presentaron a los alumnos algunas ideas y materiales comunes y se hizo especial hincapié en algunas cuestiones de la geometría de la tortuga. Los estudiantes tuvieron también la oportunidad de elegir sus propias actividades y llegaron a desarrollar muchos proyectos Logo diferentes, incluyendo juegos matemáticos, juegos de palabras y conversaciones, figuras animadas, exploraciones geométricas, tres en raya, y juegos de acción dinámica. Eran los estudiantes quienes se encargaban de estos proyectos, mientras que los profesores estaban allí para ayudarlos a lograr sus objetivos.

Los resultados del proyecto indican que el entorno de aprendizaje con Logo es adecuado para tipos muy variados de estudiantes. Todos ellos, de los muy dotados a los de resultados académicos más pobres, salieron airoso de las clases de Logo. El sorprendente éxito entre los estudiantes con problemas de aprendizaje, dio lugar a una propuesta separada para proporcionar formación en Logo y equipo a profesores especializados en este área. El informe final del proyecto resume los estilos de aprendizaje de los estudiantes y analiza lo que aprendieron en las materias de la programación de ordenadores y de matemáticas. Un segundo volumen del informe hace un examen individual de las experiencias de aprendizaje de cada uno de los 16 estudiantes. El informe puede servir de base para un plan de estudios inicial con Logo, y como fuente rica en ideas sobre proyectos para proponer a estudiantes con intereses y destrezas notablemente divergentes.

El proyecto Logo de Brookline no tuvo demasiado éxito en la obtención de datos objetivos acerca de las mejoras en el aprendizaje conseguidas por los estudiantes. Las pruebas o exámenes habituales se rechazaron por irrelevantes para los objetivos del proyecto (la destreza en el uso de la geometría de

la tortuga no queda medida por un examen de matemáticas de sexto grado). Las pruebas sobre estrategia de resolución de problemas y los tests matemáticos diseñados y aplicados por los responsables del proyecto, no obtuvieron resultados concluyentes. El problema del desarrollo de pruebas objetivas en áreas tales como la capacidad de resolución de problemas o el pensamiento en términos de procedimientos, sigue siendo una cuestión pendiente para los investigadores en formación.

Otra limitación del proyecto fue que requería profesores extremadamente sensibles e inteligentes, con mucho tiempo disponible para considerar las necesidades de cada estudiante. La esperanza de los autores del proyecto es que este informe, de dos volúmenes, con sus análisis sobre el aprendizaje de los estudiantes y muchos ejemplos de proyectos realizados, se convierta en un recurso efectivo para profesores que trabajen actividades más concretas. El informe se proponía también servir de base para un plan de estudios con Logo que desarrollarían proyectos posteriores.

El segundo proyecto Logo Brookline, también sostenido por la **National Foundation**, tenía el objetivo de desarrollar un plan de estudios basado en la utilización de Logo en las clases. Se introdujeron ordenadores en las aulas de los grados cuarto al octavo. Los profesores recibieron un pequeño entrenamiento y los medios didácticos desarrollados para profesores y alumnos. Durante el proyecto, dos ordenadores circularon entre diversas aulas. Cada clase hizo un uso exclusivo del ordenador durante un período de 8 a 12 semanas. Durante este tiempo, los estudiantes trabajaron por su cuenta en el ordenador, individualmente o en parejas, mientras el resto de la clase seguía con su trabajo normal. Una vez a la semana, la clase entera se

reunía para una lección en la que se compartían ideas, se introducían nuevos conceptos y se asignaban nuevos trabajos.

El material para el desarrollo del plan de estudios generado por el proyecto está a dos niveles diferentes: una programación introductoria para los grados cuarto al sexto, y un conjunto de proyectos Logo avanzados basados en la ejecución y modificación de un conjunto de juegos con la "dinatortuga" (la tortuga de Logo dotada de velocidad y aceleración). El plan de estudios introductorio incluye instrucciones paso a paso para los estudiantes, así como una serie de prácticas de diferentes tipos. Los profesores reciben información sobre todo lo concerniente a la disposición física de los ordenadores en la clase, los conceptos que los estudiantes deben aprender, sugerencias sobre cómo dividir las lecciones y una lista de comprobaciones para ir controlando el progreso de los estudiantes.

Las actividades avanzadas se centran en una serie de juegos con la dinatortuga que pueden utilizarse de dos maneras diferentes. Los juegos constituyen un micro-mundo en el cual los estudiantes pueden explorar el comportamiento de la "dinatortuga" —la tortuga Logo que ha sido programada para que siga las leyes del movimiento de Newton—. Cada juego introduce un nuevo factor a considerar. El primer juego consiste en hacer que la dinatortuga choque contra un objetivo, lo que fuerza al estudiante a aprender a controlar su momento y a entender algo sobre cómo se combinan las cantidades vectoriales de fuerza y momento. El segundo juego consiste en conducir la tortuga alrededor de una pista circular, introduciendo así algunos de los conceptos que intervienen en el movimiento de rotación.

(continuará en el próximo número)

El más vendedor...

**Su Anuncio en
LA VANGUARDIA**

**Ó COMO HACER
UN BUEN
NEGOCIO
CADA DIA EN
CATALUÑA**

Cataluña Política Interna-
cional Nacional Cultura
Economía Deportes Espec-
táculos Pasatiempos Anun-
cios económicos Bolsa del auto-
móvil Ir de compras Pisos Al-
quileres Demandas Ofertas
Sucesos Guía médica Servi-
cios del hogar Cine Teatro
Religión Guía de la enseñan-
za Avisos oficiales Bolsa del
trabajo Día a día El tiempo
Televisión Cataluña Política
Internacional Nacional Cultura
Economía Deportes Espec-
táculos Pasatiempos Anun-
cios económicos Bolsa del auto-
móvil Ir de compras Pisos Al-
quileres Demandas Ofertas
Sucesos Guía médica Servi-
cios del hogar Cine Teatro
Religión Guía de la enseñan-
za Avisos oficiales Bolsa del
trabajo Día a día El tiempo
Televisión.

Cataluña Política Interna-
cional Nacional Cultura
Economía Deportes Espec-
táculos Pasatiempos Anun-
cios económicos Bolsa del auto-
móvil Ir de compras Pisos Al-
quileres Demandas Ofertas
Sucesos Guía médica Servi-
cios del hogar Cine Teatro
Religión Guía de la enseñan-
za Avisos oficiales Bolsa del
trabajo Día a día El tiempo
Televisión Cataluña Política
Internacional Nacional Cultura
Economía Deportes Espec-
táculos Pasatiempos Anun-
cios económicos Bolsa del

automóvil Ir de compras Pi-
sos Alquileres Demandas
Ofertas Sucesos Guía médi-
ca Servicios del hogar Cine
Teatro Religión Guía de la
enseñanza Avisos oficiales
Bolsa del trabajo Día a día El
tiempo Televisión Cataluña
Política Internacional Nacio-
nal Cultura Economía De-
portes Espectáculos Pasa-
tiempos Anuncios económicos
Bolsa del automóvil Ir de com-
pras Pisos Alquileres De-
mandas Ofertas Sucesos Guía
médica Servicios del hogar
Cine Teatro Religión Guía

de la enseñanza Avisos oficia-
les Bolsa del trabajo Día a
día El tiempo Televisión.

Cataluña Política Interna-
cional Nacional Cultura
Economía Deportes Espec-
táculos Pasatiempos Anun-
cios económicos Bolsa del auto-
móvil Ir de compras Pisos Al-
quileres Demandas Ofertas
Sucesos Guía médica Servi-
cios del hogar Cine Teatro
Religión Guía de la enseñan-
za Avisos oficiales Bolsa del
trabajo Día a día El tiempo
Televisión Cataluña Política
Internacional Nacional Cultu-

ra Economía Deportes Es-
pectáculos Pasatiempos Anun-
cios económicos Bolsa del
automóvil Ir de compras Pi-
sos Alquileres Demandas
Ofertas Sucesos Guía médi-
ca Servicios del hogar Cine
Teatro Religión Guía de la
enseñanza Avisos oficiales
Bolsa del trabajo Día a día El
tiempo Televisión Cataluña
Política Internacional Nacio-
nal Cultura Economía De-
portes Espectáculos Pasa-
tiempos Anuncios económicos
Bolsa del automóvil Ir de com-
pras Pisos Alquileres De-
mandas Ofertas Sucesos Guía
médica Servicios del hogar
Cine Teatro Religión Guía
de la enseñanza Avisos oficia-
les Bolsa del trabajo Día a
día El tiempo Televisión.

Cataluña Política Interna-
cional Nacional Cultura
Economía Deportes Espec-
táculos Pasatiempos Anun-
cios económicos Bolsa del auto-
móvil Ir de compras Pisos Al-
quileres Demandas Ofertas
Sucesos Guía médica Servi-
cios del hogar Cine Teatro
Religión Guía de la enseñan-
za Avisos oficiales Bolsa del
trabajo Día a día El tiempo
Televisión Cataluña Política
Internacional Nacional Cultura
Economía Deportes Espec-
táculos Pasatiempos Anun-
cios económicos Bolsa del
automóvil Ir de compras Pi-
sos Alquileres Demandas
Ofertas Sucesos Guía médi-
ca Servicios del hogar Cine

...el que más vende



DONDE LA PUBLICIDAD ES NOTICIA

GUIA PRACTICA DE ORDENADOR POPULAR

ANUNCIESE

Telfs.

91 734 68 90
91 734 33 42
93 302 36 48



ELECTRONICA
SANDOVAL S.A.

DISTRIBUIDORES DE

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

ELECTRONICA SANDOVAL, S. A.
 C/. SANDOVAL 3, 4, 6 - MADRID-10
 Teléfonos: 445 75 58-445 76 00-445 18 70-
 447 42 01



Primera Clase en informática.

CENTRAL:

MADRID-20
 Capitán Haya, 38
 Tel. 27027 08
 Télex: 23586 nimad E

DELEGACIONES:

ALICANTE-2
 Rambla de Méndez Núñez, 42, Ptt. 2.º
 Tel. 21 05 00

BARCELONA-36
 Urgel, 240-1.º
 Tel. 25032 06

BILBAO-10
 Aureliano Valle, 4
 Tels. 443 1666 - 443 05 12

LOGROÑO
 Avda. de Portugal, 2-5.º
 Tel. 22 59 05

MADRID-20
 Capitán Haya, 38-4.º
 Tels. 27937 03/78/08

MALAGA
 Compositor Lhemberg Ruiz, 6
 Edificio "Galaxia", Ptt. 1.º
 Tel. 39 13 54

MURCIA
 Jaime I el Conquistador, 3
 Tel. 23 99 76

P. DE MALLORCA
 Juan Maragall, 17
 Tel. 46 43 45

PAMPLONA
 Navarro Villoslada, 1 bis
 Tels. 24 76 00 - 24 76 58

S. SEBASTIAN
 Catalina de Erauso, 23
 Tel. 45 18 11

SEVILLA-5
 Avda. S. Francisco Javier, s/n.
 Edificio "Sevilla-2", Ptt. 10.º
 Tel. 65 21 11

VALENCIA-10
 Pza. Legión Española, 12 bis
 Tel. 36 18 12 (6 líneas)

ZARAGOZA-1
 Coso, 100. Edificio "César Augusta"
 Tel. 21 29 55

Y servicio técnico para toda España.



DATA
PROCESSING 2000,
S. A.

EN MICROINFORMATICA, INFORMESE ANTES

Sabino Arana, 22-24, bajos.
Barcelona-28.
Teléfono 330 77 14.

VENTA DE MICROORDENADORES PARA LOS SECTORES:

- PROFESIONAL.
- HOGAR PERSONALES.
- ENSEÑANZA.
- HOSPITALARIO

ESPECIALIZADOS EN MEDIMATICA.
COMPLETOS SERVICIOS
EMPRESARIOS IMFORMATICOS.

P en propio edificio.



ICL España
International
Computers, SA

CENTRAL

Luchana, 23, 3.º
 Teléf. 445 20 61 (*)
MADRID-10

DELEGACIONES.

BARCELONA-6
 Tuset, 19
 Teléf. 209 55 22/57 43

MALAGA-10
 Avda. de Andalucía, 25
 Oficina 17
 Teléf. 34 90 90

SEVILLA
 Avda. República Argentina, 68
 Teléf. 45 05 48

VALENCIA-4
 Avda. Navarro Réverter, 2, 8.º
 Teléf. 334 88 98/89 66

SUSCRIBASE A

ORDENADOR POPULAR

TELEFONO 435 63 55

GUIA PRACTICA DE ORDENADOR POPULAR



Programas específicos para arquitectura, construcción y obra civil, sobre microordenadores Hewlett-Packard.

Pídanos Catálogo gratuito.

SOFT biblioteca de programas

Apartado de Correos, 10.048. Tel. (91) 440 35 40. Madrid.



INVEST MICROSTORE

De tu formación en Informática depende tu futuro, cualquiera que sea tu profesión.

- MICROORDENADORES: NEWBRAIN, ORIC, VIC-20, DRAGON-32.
- COMMODORE: 64, 4032 Y 8032-SK.
- PERIFERICOS: IMPRESORA SEIKOSHA, NEWPRINTER Y EPSON, PANTALLAS, DISCOS, CASSETTES.
- PROGRAMAS DOCENTES Y DE JUEGOS PARA: NEWBRAIN, ORIC, VIC-20 Y COMMODORE-64.
- PROGRAMAS PROFESIONALES.
- PROGRAMAS A MEDIDA.
- Asesoramiento permanente.
- Cursos gratuitos de entrenamientos, cursos periódicos de Basic, Pascal, ensamblador y lenguaje máquina.

GENOVA, 7. 2.º
MADRID-4

(91) 419 96 64
(91) 410 17 44

Computerland®

La mayor cadena mundial de tiendas de microordenadores, periféricos, software y accesorios etc., le ofrece a partir de ahora en España, los mismos productos y servicios que en el resto del mundo (más de 350 puntos de venta)

(abierto sábado mañanas)

Computerland® madrid

Castelló, 89 (Esq. Juan Bravo)

Tfno: (91) 435 29 38

Computerland® barcelona

Infanta Carlota, 89 (Esq. Entenza)

Tfno: (93) 322 06 66

Computerland® tenerife

Méndez Núñez, 104 B

Tfno: (922) 28 20 58

Computerland® las palmas

Carvajal, 4

Tfno: 23 07 08

Computerland® valencia

Gran Vía Marqués del Turia, 53

Teléf. (96) 322 46 01



los profesionales que saben escuchar

Computer Systems

Martínez Villergas, 1

Teléf. 403 60 00 y 403 61 00

MADRID-27

Oficinas en:

Avinguda Diagonal, 618

Teléf. 322 25 11

BARCELONA-21

Alameda de Recalde, 36-7-8

Teléf. 424 59 27 y 424 56 24

BILBAO-9

Cabo Santiago Gómez, 3-1.º

Teléf. 26 01 00

LA CORUÑA

Ventura Rodríguez, 2

Teléf. 24 37 77, 24 19 86 y 24 19 90

OVIEDO

República Argentina, 24-13.º

(Torre de los Remedios)

Teléf. 27 78 00

SEVILLA-11

Colón, 43

Teléf. 378 26 08 y 378 27 12

VALENCIA-4

Coso, 100-8.º

Teléf. 23 16 13 y 23 64 39

ZARAGOZA-1



LA PRIMERA TIENDA
DE ORDENADORES
PERSONALES

- DIGITAL • ALTOS
- EDSON HX 20 • GENIE
- APPLE • NEW BRAIN

Libros - Revistas

COMPUSTORE I

C/ DOCE DE OCTUBRE, 32 - MADRID-9

Tels. 274 68 96 - 409 36 74

COMPUSTORE II

C/ GALILEO, 90 - MADRID-15

Tels. 254 38 52

SUSCRIBASE A

ORDENADOR POPULAR

TELEFONO 435 63 55

GUIA PRACTICA DE ORDENADOR POPULAR

ANUNCIESE

Telfs. 734 68 90
734 33 40



**HEWLETT
PACKARD**

MADRID

Ctra. de La Coruña, Km. 16,400
Las Rozas
Tel. 637 00 11. Telex 23515

BARCELONA-29

c/. Entenza, 321
Tel. (93) 322 24 51. Telex 52603

SEVILLA-5

Avda. San Francisco Javier, s/n.
Planta 10. Edificio Sevilla 2
Tel. (954) 64 44 54

BILBAO-1

c/. San Vicente, s/n.
Edificio Albia II
Tel. (94) 423 83 06

VALENCIA-10

c/. Ramón Gordillo, 1
Tel. (96) 361 13 54



Cii Honeywell Bull

SEDE SOCIAL

Arturo Soria, 107
Tel.: 413 32 13. MADRID-33.

DELEGACIONES

Avinguda Diagonal, 633.
Tel.: 330 66 11. BARCELONA-29.

Arturo Soria, 107.
Tel.: 413 12 13. MADRID-33.

Menéndez Pelayo, 5 bis.
Tel.: 361 79 12. VALENCIA-10.

Gran Vía, 89
Tel.: 441 28 50. BILBAO-11.

Paseo de los Olmos, 9.
Tel.: 39 53 91. SAN SEBASTIAN.

Madre Rafols, 2.
Tel.: 43 87 00. ZARAGOZA-4.

Santa Catalina, 13.
Edificio Las Nieves.
Tel.: 22 28 64. LA CORUÑA.

Avda. San Francisco Javier, s/n.
Edificio Sevilla II
Tel.: 64 41 61. SEVILLA.

Avda. de Maisonnave, 33, 39.
Tel.: 12 10 63. ALICANTE.



FLOPPY

INFORMATICA DE GESTION S.A.

SOFTWARE

Desarrollo de programas para las necesidades concretas de su empresa en su propio ordenador

SERVICIOS

Si su empresa no dispone de ordenador podemos cubrir sus necesidades desde nuestros propios ordenadores

(gestión, gráficos, calculos científicos, etc.)

C/. DE LAS FUENTES, 10 - 3º D - TNOS. 248 17 08 / 07
MADRID - 13

ANUNCIESE

Telfs.

91 734 68 90
91 734 33 42
93 302 36 48

INDICE DE ANUNCIANTES

Pág. Anuncio

2	● Epson
6/7	● Casio
8/9	● Atari
11	● Sonimag
12	● Nixdorf
14/15	● Commodore
23	● El Corte Inglés
25	● Electrónica Sandoval
27	● Sperry
29	● Bull
31	● Victor
35	● Osborne
37	● Computecro Argüelles
39	● E.T. Orange
73	● Entel
77	● Otesa
81	● HM Informática
87	● ZX 81
89/90	● ZX Spectrum
99	● Seikosha
100	● Dragon

SUSCRIBASE A

**ORDENADOR
POPULAR**

TELEFONO 435 63 55

SERVICIOS **ENTEL**

CUZCO IV

CONSULTORIA • INGENIERIA DE SISTEMAS
DISEÑO Y DESARROLLO DE SISTEMAS • TRATAMIENTO DE LA INFORMACION

SELECCION Y FORMACION DE PERSONAL • COMERCIALIZACION DE EQUIPOS Y LOGICAL
GESTION DE CENTROS • INDUSTRIA DE LA INFORMACION • NUEVAS TECNOLOGIAS

P.º CASTELLANA 141 • EDIFICIO CUZCO IV • TELS. 450 90 96 • 450 70 00 • MADRID. 16.

VIA AUGUSTA 200 • EDIFICIO ZURICH • TELS. 209 03 77 • 200 85 33 • BARCELONA. 21

XEROX 820 II

Casi un año después de haber puesto en el mercado su primer ordenador personal, Xerox lanza una versión mejorada del mismo. Se trata del modelo 820-II, sucesor del 820 a secas. El primer ordenador de uso personal de Xerox no aparece con la intención de ser un ordenador más, sino que tiene un objetivo perfectamente definido por su fabricante: ser un puesto de trabajo personal dentro de un ambiente tipo Ethernet (este es el nombre de la red local desarrollada hace tiempo por Xerox). Aún más, el 820 fue diseñado para ampliar las posibilidades del sistema star para la gestión de la información, integrándose en él, lo cual no es ningún obstáculo para que el sistema 820 y también por supuesto el 820-II proporcionen magníficas prestaciones como ordenadores personales.

La utilización del 820-II se hace extremadamente sencillo debido a que se maneja principalmente por medio de menús que aparecen en la pantalla. Además, se trata de un sistema con un alto grado de estandarización. En primer lugar conviene recordar que está controlado por el clásico microprocesador Z80 A y que corre bajo el sistema operativo CP/M de Digital Research.

Físicamente, el microordenador se compone de un teclado inteligente, que va separado de la carcasa, que contiene el tubo catódico y la tarjeta principal. Las unidades de almacenamiento masivo elegidas (discos de 5 1/4 y 8 pulgadas, así como discos duros) poseen su propia caja separada del conjunto principal.



La Unidad Central de Proceso está incluida en la tarjeta principal, un circuito impreso de dimensiones medias fabricado con tecnología multicapa, lo que permite optimizar el diseño disminuyendo el tamaño final de la tarjeta. El corazón de la UCP es el microprocesador Z-80A de Zilog, una versión rápida que funciona con un reloj patrón de 4MHz, en contraste con el Z-80 del 820 anterior que funcionaba más lentamente, 2,5 Mhz.

La memoria principal de que dispone el 820-II son 64 Kbytes de RAM (memoria de acceso aleatorio), que puede ser incrementada en 32 Kbytes más por medio del bus del sistema. Es interesante observar que en esta nueva versión se puede acceder al bus del sistema, ampliándose las posibilidades para aquellas firmas independientes que fabrican accesorios para microordenadores, que de esta forma pueden acceder de manera más sencilla a la electrónica interna del sistema. Los *chips* de memoria utilizados en la tarjeta principal son de 64 Kbits, reduciéndose el número necesario de estos a solamente ocho.

Existen además 8 Kbytes de memoria ROM (memoria que sólo puede ser leída) en ella va grabado el *firmware* que permite, entre otras cosas, la carga automática del sistema operativo en el ordenador a partir del disco, ejecutando las órdenes pertinentes.

Un circuito interesante que aparece en esta tarjeta es el temporizador de la familia Z-80; sirve como reloj cuando se reclama la función TIME por pantalla que no hace otra cosa que actuar como el "reloj de pulsera" del 820-II, cuando sean necesarias las tareas de funcionamiento en tiempo real, entre otras posibilidades.

A la hora de elegir pantalla se ofrecen dos opciones, fósforo blanco y fósforo verde. Esto quiere decir que se podrán tener caracteres blancos o verdes sobre fondo negro o viceversa si se elige el video inverso.

El formato de pantalla es el clásico de 24 filas por 80 columnas. A su vez, el formato de los caracteres individuales es una matriz de 7 x 10 puntos.

El brillo total de la pantalla es ajustable por medio de un control dispuesto en la parte inferior del lado izquierdo de la carcasa que contiene a la pantalla.

Los caracteres que aparecen en la pantalla, que pueden ser hasta un total de 1.920 (24 x 80); se almacenan en una memoria RAM de 2 Kbytes que es continuamente leída por la pantalla, descargando así a la memoria principal. En cuanto a las comunicaciones con el mundo exterior, el 820-II dispone de dos *ports* serie, una para la impresora y la otra para cualquier otro tipo de periférico que lo requiera. La velocidad de transmisión de ambas puede establecer-

se al definir la configuración del sistema, mediante el *software*, eligiendo las distintas opciones en la pantalla. Asimismo, el 820-II también dispone de dos *ports* paralelas.

El teclado, que como se vio antes conforma una unidad independiente, es lo que ha dado en llamarse teclado inteligente.

Dispone de 55 teclas en una configuración estándar tipo máquina de escribir. Se complementa con otras 20 teclas que son el teclado numérico para uso rápido (*keypad* numérico), muy útil cuando se quieren introducir grandes cantidades de cifras. En este *keypad* se dispone también de las cuatro teclas de control del cursor en pantalla y la tecla ESCAPE, como elementos interesantes para optimizar las tareas de manejo.

Se añade que 32 de las teclas son redefinibles por *software*. El teclado también puede elegirse totalmente por *software* (SETUP), pudiéndose escoger a través del menú correspondiente si se desea un teclado español, inglés, americano, alemán, italiano, francés, sueco, holandés, noruego o ASCII. El juego de caracteres viene a ser un total de 94, con letras mayúsculas y minúsculas; característica esta muy importante cuando se desea trabajar con paquetes para proceso de textos. Al decir, antes que el teclado es inteligente, la afirmación se refiere a que dispone de su propio microprocesador y una pequeña memoria RAM de 64 bytes, además de 1K de ROM.

Todo el conjunto recibe el suministro de corriente a partir de una fuente de alimentación dispuesta en el interior de la carcasa de monitor. La particularidad de la misma es su diseño; corresponde a la línea (que cada vez están siguiendo más fabricantes), de las fuentes de alimentación conmutadas. La ventaja de este tipo de diseño se centra en dos características fundamentales cuando se desea aumentar la fiabilidad de los sistemas. El calor disipado disminuye en gran medida y la tensión de alimentación es más estable (los dos mayores enemigos de los componentes electrónicos son el calor y las tensiones que exceden sus capacidades).

Por ahora, Rank Xerox ha venido haciendo mayor hincapié en los *floppies* de 8 pulgadas, aunque dispone también de unidades de *diskette* de 5 1/4 pulgadas. El disco duro Winchester complementa las configuraciones con mayores necesidades de almacenamiento.

La configuración básica del 820 II está compuesto, como hemos dicho más arriba, por la CPU, teclado y monitor, que se vende a 370.000 pesetas actualmente. A ello debe añadirse, desde luego, el precio de las unidades de disco que se escoja. La estrategia comercial de

Xerox apunta a imponer los *drives* de 8 pulgadas, en desmedro de los de 5 1/4 pulgadas que, sin embargo, están en catálogo. Una configuración típica con unidad de disco de 8 pulgadas doble cara/doble densidad sale por unas 577.500 pesetas, lo que está francamente bien de precio.

Pero, según manifestaciones de los portavoces de Rank Xerox Española, la demanda se ha disparado en una configuración más amplia, que lleva un disco duro Winchester y que representa, en este momento el 20 por ciento de las ventas en España de este ordenador.

La cuestión de la impresora plantea una gama de opciones poco habitual. Rank Xerox, por razones propias, recomienda los modelos de Epson y Oki en matriciales o, en las de margarita, la Xerox 620 o la Diablo 630, bastante menos corriente en el mercado español, a decir verdad.

Cuando se conecta el sistema, mediante un conmutador hábilmente camuflado en la parte inferior derecha, en la pantalla aparece el siguiente mensaje: 820-II V 4.00 1982 Xerox Corp (RX v006)

L - Load System

H - Host Terminal

* ■

Lo cual indica que el sistema funciona correctamente. También indica que el sistema está puesto a cero, se obtiene la misma pantalla si se oprime el botón de reset (puesta a cero del sistema) que está situado en la parte posterior del monitor, probablemente para evitar un fortuito accidente que manda al garete ese caro programa o la aplicación que tanto tiempo nos ha llevado desarrollar.

Si previamente se ha introducido el *diskette* de Digital Research en la unidad correspondiente (es muy importante conectar la alimentación a la unidad antes de meter el disco y desconectarla después de haberlo sacado. De esta forma se evitarán desagradables sorpresas) estaremos en disposición de requerir al ordenador para que cargue (*boot* le llaman) la parte del sistema operativo que corresponde en la memoria central, simplemente tecleando la L.

Si por el contrario se elige presionar la H, el 820-II actuará como terminal de otro ordenador.

Supongamos que se introduce el *diskette* con el sistema operativo en la ranura A de la unidad de disco y que hemos elegido pulsar la L. Después de oír graznar al mecanismo de la unidad aparece en la pantalla otro mensaje: XEROX 60K CP/M vers 2.2 a 2-294 - ACOOOOOO

A > ■

Lo cual indica que ya está cargado el CP/M en el sistema. En el disco, ade-



más del CP/M existen una serie de comandos residentes (que se quedan permanentemente en la memoria RAM una vez cargados, durante todo el período de operación) otros son transitorios (se les va llamando según se les requiera o no) y las utilidades (sirven para ejecutar algunas tareas específicas).

Entre los comandos residentes existe uno llamado DIR, muy útil, pues proporciona la lista en la pantalla de los ficheros contenidos en el disco.

Con la ayuda de varios de estos comandos se puede definir la configuración del sistema que deseamos. Desde nacionalizar el teclado con la ayuda de SETUP, hasta configurar el "protocolo de entendimiento" con la impresora.

Pero todo esto le sobra a quien dis-

ponga de un *diskette* que contenga una aplicación específica que corra bajo esta versión del CP/M o del lenguaje del alto nivel elegido, que por ahora podría ser alguna de las versiones existentes del BASIC, el COBOL-80, el PROPCAL, etc.

El 820 II viene arropado con un amplio paquete de conjunto de paquetes de *software*, al que se irán añadiendo en los próximos meses nuevos programas adaptados a las necesidades del mercado español. Para Rank Xerox Española, el catálogo de *software* en castellano disponible es el arma más contundente de cara a su penetración en el mercado.

Ese catálogo incluye, como es obvio, el sistema operativo CP/M 2.2E y los paquetes Q Sort y M Sort (que funcio-

nan respectivamente en BASIC y en COBOL, además de los ya clásicos "tableros electrónicos" Supercalc y Multiplan. En tratamiento de textos destaca Wordstar en sus versiones 2.6 y 3.0, además del Select (más sencillo que Wordstar y que, además, empalma con el Multiplan). Tanto el Wordstar 3.0 como el Select ofrecen la posibilidad de *mailmerge*.

Está también, aunque todavía no se ha vendido en España, la posibilidad que ofrece la base de datos dBASE II (sólo en inglés, por el momento).

En materia de comunicaciones, Rank Xerox ofrece dos programas de emulación 3720 y 3780, bisincronos, ambos en castellano.

Pero tal vez sean los paquetes de aplicaciones de gestión el mayor atractivo que el Xerox 820 II puede ofrecer en materia de *software*. Hay, en primer lugar, tres contabilidades. Una general, en dos versiones (una de ellas elaborada por la *software house* STS y la otra de Aplicaciones Informáticas), además de otro paquete más sofisticado, que ofrece la posibilidad de calcular ratios financieros, balances presupuestarios y se aproxima a lo que es una contabilidad analítica.

A ello se añade programas de nómina de empresa, agenda electrónica, gestión de almacén, facturación inmediata y gestión comercial. Además, paquetes de aplicaciones verticales dirigidos a administración de fincas, notarios, ayuntamientos, videoclubs, restaurantes y distribuidores. No faltan en el catálogo programas de tratamiento de textos de origen español, un programa de impuesto a la Renta y dos paquetes de aplicaciones técnicas: cálculo de estructuras y gestión de obras.

FICHA

Nombre: Xerox 820 II

Fabricante: Xerox Corp.

Representante en España:

Rank Xerox Española, S. A. Josefa Valcárcel, 26. Madrid-27.

Tfno: (91) 742 41 11.

Características estándar:

- Microprocesador Z80A.
- Memoria RAM: 64 Kbytes ampliable a 96 Kbytes.
- Memoria ROM total: 8 Kbytes.
- Pantalla de 24 filas x 80 columnas.
- Teclado de 75 teclas.
- Keypad numérico de 20 teclas.
- Dos ports en serie y dos en paralelo.
- Unidad de disco floppy de 8 pulgadas.

Opcional: unidad de disco duro Winchester de 10 Mbytes.

Periféricos: impresora matricial (Oki) o de margarita (Xerox/Diablo).

Sistema Operativo: CP/M 2.2 E.

Lenguajes: BASIC, COBOL 80 y COBOL 4.6. Próximamente Fortran.

Precios:

CPU con teclado y pantalla:

370.000 pesetas.

Unidad de disco 8 pulgadas simple cara: 205.000 pesetas.

Unidad de disco 8 pulgadas doble cara: 307.000 pesetas.

Unidad de disco rígido Winchester: 705.000 pesetas.

Con precios de micros prestaciones reales de minis.

 **MONROE**
Litton Systems For Business

OC-8880/88



- 128 K RAM - Standard
- 256 K RAM - Opcional
- Una o dos unidades de 320 Kb. FD 5 1/4"
- Opcional 5 ó 10 Mb. Hard Disk, expansibles a 20 Mb.
- Sistema Operativo, MS-8 de Monroe con ISAM (Indexed Sequential Access Method)
- Sistema Operativo CP/M de digital Research
- Lenguajes: Basic, Pascal, Ensamblador
- Comunicaciones Sincronas y Asíncronas: 2780/3780 BSC, 3270 BSC, 3270 SNA/SDLC, TTY, Ascom.
- Local area Networks (LAN)
- Multi-Usuario
- Multi-Tasking

... DESDE 575.000 PTAS....

VEALOS EN LA FERIA DE MUESTRAS DE BARCELONA
2-10 JUNIO - STANS 144, 145, 146, 154 y 155
BUSCAMOS DISTRIBUIDORES PARA TODA ESPAÑA

IMPORTADOR Y DISTRIBUIDOR
EXCLUSIVO PARA ESPAÑA:



OTESA

RED COMERCIAL:
VALENCIA - BILBAO - SEVILLA - VIGO - ALICANTE - SAN SEBASTIAN

Gran Vía Carlos III, 37-39
BARCELONA
Tel. 330 34 74

Miguel Yuste, 16
MADRID-17
Tels. 754 33 00

Desee más amplia información sobre sus equipos

Nombre _____
Empresa _____
Dirección _____
Población _____
Tel. _____

He aquí un ordenador que, como muchos humanos, trata de recuperar sus señas de identidad. Nacido de la mente imaginativa de **Chuck Peddle** (el creador del *chip* 6502, que al bajar los costes abrió el camino a la difusión de los microordenadores), fue bautizado con un doble nombre: **Sirius 1** para el mercado europeo, **Víctor 9000** para su venta en Estados Unidos. Ocurrió que el **Sirius** alcanzó un extraordinario éxito en este lado del Atlántico, pero el *marketing* americano no resultó brillante. Y así, **Sirius Computer**, la firma creada por Peddle para fabricar el ordenador, terminó absorbiendo a **Víctor United**, que en principios era su casa matriz. A partir de entonces, el **Sirius 1/Víctor 9000** se impuso también en su país de origen. Pero, cuando parecía que el ordenador llevaría en adelante un solo nombre universal —**Sirius**— un complicado litigio sobre la propiedad de la marca obligó a quedarse con el otro apelativo bautismal: **Víctor**.

Es algo más que una anécdota, porque el problema que ahora afrontan fabricantes y distribuidores europeos es cómo fijar en el público la nueva marca sin perder la imagen de calidad que habían ganado con la anterior. Es un problema que no tiene, en verdad, el representante en España, **Computronic**, porque en nuestro país la comercialización y la campaña de imagen comenzó demasiado tarde para haber establecido la marca en la memoria de los usuarios.

Chuck Peddle concibió el **Sirius** —habrá que acostumbrarse a llamarlo **Víctor**— como un ordenador "de tercera generación". La primera —el **Apple I**— era en realidad un producto de aficionados. La segunda —los **Apple II**, **PET** y **TRS-80**— fue diseñada para gente que quería tener un ordenador sobre su mesa. La tercera, que es la actual, representa productos más acabados, pensados para ser usados como equipos profesionales, en una oficina o en una red de oficinas. El futuro, opina Peddle, apunta hacia los ordenadores trabajando en redes, lo que desplazará a los sistemas multiusuarios.

Y con estas ideas en mente fue diseñado el **Víctor**. Para tocar por arriba el segmento de los minis y empalmar por abajo con los ordenadores personales de segunda generación. Su configuración va dirigida a los usuarios que encuentran insuficientes las prestaciones de estos últimos, buscando capacidades más profesionales. Su memoria de base empieza en 128 Kbytes y llega a 896 Kbytes, siendo la configuración media, pues, de 256 Kbytes con dos *floppys* doble cara/doble densidad.

Físicamente, tres módulos conforman el **Víctor**: la carcasa principal (que contiene el procesador, la RAM, la fuente de alimentación y los *drives* de disco),

una unidad de *display* ubicado encima de la carcasa principal, y el teclado.

El procesador, la RAM de 128 Kbytes y todos los demás componentes electrónicos se sitúan en un gran PCB al fondo de la carcasa principal. La CPU está basada en un *chip* **Intel 8088** funcionando a 5 MHz. Es el **Víctor** un microordenador de 16 bits, pero internamente se parece a uno de 8 bits. Lo que, en términos de *marketing*, significa ofrecer la potencia de los primeros con el bajo coste de los segundos. En cualquier caso, su ventaja reside en la compatibilidad con el **Intel**.

Se le pueden añadir tarjetas, gracias a las ranuras que para ello existen en su unidad central. Tres de ellas están allí para permitir la expansión del sistema a su plena capacidad mediante sucesivas adiciones de 128 Kbytes. Si se pretende subir esa capacidad a un megabyte, es necesaria una unidad de expansión externa.

El **Víctor** tiene tres puertas con las cuales se comunica hacia el mundo exterior. Una de ellas para impresora paralela, que comparte a la vez el estándar **Centronics** y el **IEEE-488**, así como una puerta para impresora asíncrona **RS-232C** con baudios programables. Una segunda puerta serial provee entre 1200 y 9600 baudios para comunicaciones.

Una característica inusual en esta gama de ordenadores es un digitizador de sonidos, que permite grabar la voz en disco y oírla a través del altavoz. Desafortunadamente, la máquina no dispone de un dispositivo para introducir la voz. Sorprende, al observar el **Víctor**, que le falte un reloj-calendario interno permanente. Sin embargo, también es posible conseguir una tarjeta que añade esa función.

Dos áreas de memoria se reservan al *display*, una de las características más importantes de la máquina que estamos comentando. En modo normal la pantalla tiene 25 líneas de 80 caracteres cada una. El monitor es de fósforo verde de 12 pulgadas y da una imagen clara y nítida.

Como casi todo lo demás, la pantalla está bajo completo control del *software*. Un área de 4 K de RAM mantiene los caracteres que se impresionan en pantalla mientras que un área de más abajo contiene los modelos de puntos que construyen los caracteres. Los caracteres de la pantalla estándar 80 × 25 son contruidos con matrices de 10 × 16, lo que hace el texto claro y elegante. Puesto que los modelos de puntos están en RAM se puede acceder a ellos y cambiarlos utilizando un programa llamado **Efont**. Se pueden almacenar hasta 2048 caracteres en RAM y en cualquier momento un conjunto de caracteres se pueden almacenar en disco y retraerse a la memoria a través del programa.

Naturalmente el **Víctor** tiene capacidad gráfica. Y, por cierto, es alta. La pantalla admite una matriz de puntos de 800 × 400, más que cualquier otro ordenador de su precio. En el modo gráfico es posible no sólo realizar espectaculares gráficos sino además, escribir textos de una forma más comprimida: hasta 50 líneas de 132 caracteres perfectamente legibles.

Desde el teclado o a través del programa se puede controlar la doble intensidad, el video invertido, el subrayado, el contraste y el brillo de la pantalla, así como el altavoz. El monitor no tiene ningún control, es de pantalla antirreflexiva y rotatorio.

El teclado tiene 95 teclas divididas en varios grupos. El grupo principal es, evidentemente, el teclado *qwerty*. Cuando se conecta el ordenador este se sitúa en minúsculas, pero por supuesto hay una tecla de elevación (**SHIFT**), que por cierto no tiene un indicador luminoso que indique cuando está activada. Dos teclas se encuentran al final de la barra espaciadora, justo donde uno espera encontrarse la tecla de elevación. Estas son las teclas "**ALT**" y "**CONT**". La primera de ellas no es sino una tecla de control. La segunda lo que hace es interrumpir el programa a la primera pulsación y continuarlo a la segunda.

En la parte superior del teclado hay una fila de 7 teclas de funciones programables y a la derecha el bloque de teclas numéricas que incluye los signos de tanto por ciento, más, menos, plor y dividido, pero no el signo igual.

Entre el bloque de teclas numéricas y el teclado principal hay dos columnas de teclas que controlan el cursor, limpian la pantalla e insertan y borran caracteres y líneas. Presionando **ALT** y algunas de estas teclas obtenemos el control del brillo y el contraste de la pantalla así como del volumen del altavoz. A la izquierda del teclado hay otra columna donde están contenidas las teclas de **Escape** y de repetición así como el control de la doble intensidad, video invertido y subrayado.

El sistema de conexión es muy simple. Todos los conectores están en la parte posterior de la carcasa principal y todo lo que hay son las conexiones para el video y el teclado y la tecla de encendido. No hay una luz indicadora de que el ordenador esté funcionando pero el ventilador hace tanto ruido que realmente no lo necesita. Insertando el disco del sistema operativo en el *drive* de la izquierda, el ordenador se encarga de leerlo sin mayor necesidad de instrucciones por parte del usuario. Si el disco no está situado aparece un mensaje de ello en la pantalla.

El **Víctor** viene con dos sistemas operativos, el **CP/M-86** y el **MSDOS**.



VICTOR

Nuestros lectores habituales deben conocer ya el CP/M puesto que publicamos un artículo sobre él en nuestro número 2. Pues bien, el CP/M-86 es la versión para el 8086 del CP/M, escrito por Digital Research. El MSDOS de Microsoft es virtualmente idéntico al sistema operativo que produjo Microsoft para el ordenador Personal de IBM, del que también hablamos en nuestro número 1. No merece, pues, la pena detenernos más en este tema.

El Victor admite la implementación de distintos lenguajes con la ventaja que esto lleva consigo. De esta forma, puede trabajar con MS-COBOL, MS-PASCAL, MS-FORTRAN, CBASIC/86, MS-BASIC, Level II COBOL y GW-BASIC. Todos ellos son de Microsoft, excepto CBASIC/86 que es de Digital Research y Level II COBOL que es de Microfocus.

Una amplia colección de *software* está disponible para trabajar con el ordenador y se podría decir que la mayoría de los programas que integran esa colección son "hermosos". Podemos dividir el *software* disponible en dos grupos: el *software* de utilidad (lo que la casa llama Utility Software) que en la mayoría de los casos son ayuda al programador, y el *software* de aplicaciones. Comencemos con el primer grupo, aunque sabemos que el éxito que pueda alcanzar el ordenador dependerá de la

calidad de sus programas de aplicaciones.

Programmer's toolkit es un conjunto de diversos programas de utilidad, a saber: PMATE-86; un editor de textos diseñado especialmente para el programador; MS-LINK es un linkeditor relocizable diseñado para linkeditar módulos objeto producidos separadamente; MS-LIB crea y modifica archivos utilizados con MS-LINK; DEBUG es un programa de rastreo y depuración de programas utilizado para desarrollar el chequeo controlado para los ordenadores basados en el 8086/8088; MSCREF produce una lista alfabética de todos los símbolos utilizados en un programa MACRO-86; FABS/86 es un programa en ensamblador diseñado para mantener archivos claves para grabación y captación en grandes archivos de datos; y AUTOSORT/86 es un clasificador (sort) de datos.

Dentro del *software* de aplicaciones podemos hacer otros dos grupos: el *software* de aplicaciones "universal", es decir, aquel que se puede utilizar en cualquier parte del mundo en un Victor, y el *software* de aplicaciones "nacional", que es aquel sólo utilizable en España y que ha sido elaborado para Computronic bajo el nombre genético de Prisma. El representante de Victor en nuestro país está convencido —y pensamos que

con razón— de que el 70 por ciento por lo menos de su mercado está en el mundo de las empresas. El *software disponible* debe, pues, dirigirse a ese mercado. Concretamente, el paquete Prisma se compone de cinco elementos. Dos programas de Contabilidad General, otro de Facturación/Almacén, un tercero de Gestión Laboral y un generador de archivos. Se está elaborando actualmente otros cinco o seis productos *desoftware* nacional, dirigidos esta vez a aplicaciones verticales. Hay o habrá pronto programas para administración de colegios, transportes, exportación e importación de frutos, gestión hotelera, administración de granjas porcinas, fabricación de muebles, clínicas. Y un paquete de liquidación del Impuesto sobre la Renta, además de otro para diseño gráfico en tres dimensiones.

El *software* "universal" incluye el célebre programa dBASE II, que permite crear un potente sistema de base de datos), el de sobra conocido Multiplan y Benchmark (programa de tratamiento de textos con gráficos, diccionario y comunicaciones). Todos ellos para rodar en el sistema operativo MS DOS. En cuanto al CP/M 86 se dispone de dBASEII, de Supercalc (otra variante actual de "tablero electrónico", el editor de textos WordStar, MailMerge correo para Wordstar) y Select (tratamiento de textos con diccionario).

En opinión de su fabricante americano, el Victor es un ordenador que puede aspirar a colocar el 50 por ciento de su producción en grandes cuentas. Concretamente, se insiste en destacar la operación, se jugará la baza de las comunicaciones —a este ordenador como el modelo que utilizarán —en esta gama, por supuesto— sus filiales de todo el mundo. En coherencia con esa convicción, se jugará la baza en las comunicaciones —tanto sincronicas como asincronicas— un rasgo que diferencia el Victor de otros ordenadores personales. Por otra parte, se dice que en el curso de este año habrá nuevos anuncios. Concretamente, la conexión a la red Ethernet y una posibilidad de ampliar la capacidad de almacenamiento externo a 20 y 40 megabytes mediante el aprovechamiento de los últimos avances en la tecnología de discos Winchester. Y con estas armas, Chuck Peddle cree posible aspirar a que su producto llegue a ser el número 3 en la industria americana de microordenadores. Ambiciones no le faltan.

FICHA

Nombre: Victor

Fabricante: Sirius Computer

Representante en España:

Computronic, S. A.

Marqués de Riscal, 11

Madrid-4

Características estándar:

- Microprocesador Intel 8088

- Memoria RAM: 128 Kbytes

- Memoria ROM interna: 16

- Kbytes

- Pantalla de 12 pulgadas. 25 líneas por 80 caracteres, con posibilidad de modo comprimido. Resolución gráfica de 800 × 400 puntos.

- Teclado QWERTY con 18 teclas numéricas y 7 de función.

- Unidad de dos discos flexibles doble densidad de 5 1/4 pulgadas. 600 Kbytes formateado.

- Dos puertas seriales RS 232-C. Salida en paralelo para impresora.

Opciones: con 1,2 Mbytes formateado en doble cara/doble densidad.

Unidad de disco duro Winchester de 10,6 Mbytes formateado más 1,2 Mbytes en floppy de 5 1/4 pulgadas para backup.

Sistema Operativo: CP/M 86 y MS/DOS.

Lenguajes: C BASIC 86, M BASIC 86, MS COBOL, CIS COBOL, MS Pascal, MS Fortran, PL/1 y PL/M, además de GW Graphic Basic.

Software disponible: Prisma (paquete de contabilidad general, facturación y almacén, gestión laboral, generador de ficheros), Multiplan, Wordstar, Select. Paquetes verticales de aplicación verticales (ver texto).

Precios: configuración mínima (CPU, teclado, pantalla, dos unidades de discos flexibles con 1,2 Mbytes de almacenamiento total), 795.000 ptas. Idem con 2,4 Mbytes de almacenamiento total en discos flexibles, 948.000 ptas. Idem con 10,6 Mbytes de almacenamiento total en disco duro, 1.375.000 ptas.

PONEMOS LA INFORMATICA AL ALCANCE DE VD.

**EQUIPOS CII HONEYWELL BULL (1.º FABRICANTE EUROPEO)
MAS DE 200 EMPRESAS MECANIZADAS POR NOSOTROS
COSTES REDUCIDOS (DESDE 25.000 PTAS. MES)**

TENEMOS FUNCIONANDO ACTUALMENTE

APLICACIONES GENERALES

*Contabilidad.
Gestión Comercial.
Nóminas.*

APLICACIONES SECTORIALES

*Administración de fincas.
Agentes de aduanas.
Gestorías y asesorías.
Distribución alimentación.
Confección.
Publicidad.
Calzado.
Agencias de viajes.
Constructoras e Inmobiliarias.
Limpieza.
Cristalerías.
Declaración de renta y Patrimonio*

ADEMAS CONFECCIONAMOS SU APLICACION



VENGA Y SE LO DEMOSTRAREMOS

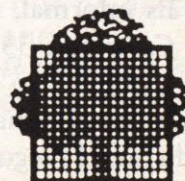


INFORMATICA S.A.

**Colombia, 53
Madrid-16
Tels. 457 30 99/457 30 73**

**General Ruiz, 4 bis
Valladolid-4
Tels. 22 21 97/22 44 02**

DISTRIBUIDOR DE



CII HONEYWELL BULL

CHRIS CRAWFORD:



Así diseño mis juegos

Es informal, imaginativo y tiene sentido del humor.

Chris Crawford, 33 años, trabaja para Atari como diseñador de video games, y lo hace bien. Está considerado como el mejor inventor en el tema y le consultan sobre sus ideas, pues cree que mediante el juego se accede al conocimiento y a la reflexión. Es físico, autodidacta en informática, y opina que su tarea está cercana al arte. En su mochila guarda cartuchos de sus juegos predilectos. Además, sus creaciones son tan buenas que puede reconocer que no son perfectas.

¿Por qué algunos juegos de ordenadores son mejores que otros? Tanto los fabricantes de juegos como los autores tratan constantemente de responder a esta pregunta. Son muchos los factores que inciden en la atracción de un juego, incluyendo calidad técnica, gráficos, sonido, la marcha, el desarrollo y la acción. Es imposible enumerar la lista de propiedades de un juego para decidir si tendrá o no éxito. El diseño de juegos es una forma de arte y, al igual que cualquiera de las formas del arte, los componentes individuales son menos importantes en sí que la manera en que están vinculados, es decir, la técnica general.

Todos los artistas tienen su propia técnica especial y un ideal para la ejecución de su arte. El pintor se preocupa de sus pinceladas, la mezcla de su pintura y la textura; el compositor musical aprende procedimientos de orquestación, tiempo y contrapunto. El diseñador de juegos también aprende toda una variedad de habilidades especializadas, técnicas e ideales para la ejecución de ese arte. En este artículo describiré algunas de las técnicas que uso en mis juegos. No comprendo lo suficientemente bien mi arte como para presentar los elementos técnicos en secuencias jerárquicas o lógicas; me limitaré a ofrecer un *potpourri* de ideas interesantes agrupadas bajo el concepto de balance, relación, monotonicidad —dado que tiene una curva de aprendizaje decididamente monotonica— y posibilidades de ganar.

Balance de juegos solitarios

Un juego solitario enfrenta al jugador con el ordenador. El ordenador y el humano son dos creaturas diferentes; donde el proceso del pensamiento humano es difuso, asociativo e integrado, el proceso de pensamiento de la máquina es directo, lineal y aritmético. Esta diferencia es lo que causa el problema. Un diseñador crea un juego de computación para beneficio del humano y por tanto el juego se lleva a cabo en el territorio intelectual del humano más que en el territorio intelectual del ordenador. Por tanto, el ordenador está, naturalmente, en desventaja. Pese a que el ordenador podría vencer con facilidad al humano en juegos que impliquen computación, clasificación o funciones similares, esos juegos serían de muy poco interés para el jugador humano en la mayoría de los casos. El ordenador debe jugar en el campo de juegos del humano, lo que le resulta muy difícil. ¿Cómo, entonces, diseñar un juego que capacite al ordenador para competir y desafiar al humano? Cuatro son las técnicas posibles: vastedad de recursos, cómputos artificiales, informaciones limitadas y marcha.

ORDENADOR POPULAR

Cada mes, *ORDENADOR POPULAR* le traerá los últimos adelantos en el mundo de los ordenadores personales —nuevos equipos, cómo usarlos, análisis de los mismos, software, etc.— y toda clase de información acerca de este apasionante mundo. Todo ello en una forma amena y fácil de leer.
Suscribase y no se pierda ningún ejemplar.

OFERTA DE INTRODUCCION



GRATIS PARA USTED si se suscribe a ORDENADOR POPULAR

Una obra imprescindible en la biblioteca de todo aficionado a la informática. El libro de más éxito de McGraw Hill en España. Un regalo de 240 páginas de gran formato (210 x 275 mm.) cuyo precio de venta es de 1.170 Ptas.

Aproveche ahora esta irrepetible oportunidad para suscribirse a *ORDENADOR POPULAR*. Envíe HOY MISMO la tarjeta adjunta, que no necesita sobre ni franqueo. Depositela en el buzón más cercano. Inmediatamente recibirá su primer ejemplar de *ORDENADOR POPULAR* más el REGALO y así durante un año (11 ejemplares).

**ORDENADOR
POPULAR**

Hermosilla, 100
Tel. 275 30 04
MADRID-9

Vastedad de recursos

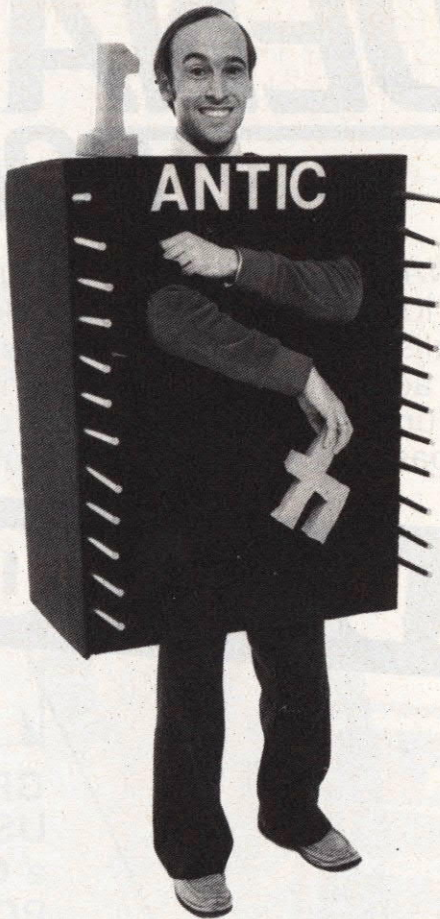
En lo que resulta de lejos la más manida técnica para balancear un juego, el diseñador provee al ordenador de vastos recursos que éste usa muy estúpidamente. Estos recursos pueden consistir en un gran número de oponentes que operan con una inteligencia sólo rudimentaria. Muchos juegos usan el siguiente esquema, por ejemplo, *Invasores del Espacio*, *Comando Misil*, *Asteroides*, *Centípede* y *Tempestad*. También es posible equipar al ordenador con un pequeño número de oponentes más poderosos que los jugadores humanos, tales como supertanques en la zona de batalla. El efecto en ambos casos es similar: la ventaja del jugador humano en inteligencia se equilibra con las ventajas materiales del ordenador.

La técnica de la vastedad de recursos tiene cierto número de ventajas. Primero, el conflicto entre el humano y el ordenador adquiere un aire de David contra Goliat. La mayoría de la gente prefiere ganar como aparente infradotado que como un igual. Segundo, esta técnica es la más fácil de implementar. Darle inteligencia al ordenador puede ser difícil, pero hacer que cantidad de jugadores repitan un simple proceso puede ser apenas más complicado que un simple *loop*. Por supuesto, la facilidad de implementación conlleva una desventaja: todos pueden hacerlo. Estamos muy sumergidos en esos juegos. La pereza y la indecisión tienen más que ver con la prevalencia de esta técnica que las consideraciones de diseño de juego.

Cómputos artificiales

La alternativa obvia para darle una ligera ventaja al jugador-ordenador es proveerlo de la inteligencia adecuada como para enfrentar al jugador humano en igualdad de condiciones. Desgraciadamente, la técnica de inteligencia artificial no se comprende tanto como para resultar útil en este contexto. El desarrollo de las técnicas de acorralamiento nos permite producir ajedrez, ajedrecistas y jugadores de ordenador *Othello*. Cualquier otro juego que pueda ser expresado directamente en técnicas de acorralamiento puede ser manejado con estas mismas técnicas. Pero son muy pocos los juegos apropiados para este tipo de tratamiento.

Una posible alternativa es el desarrollo *ad hoc* de rutinas artificiales de inteligencia. Es el método que usé en *Tanktics*, *Legionarios* y *Frente Este 1941*, con diferentes grados de éxito. Aun cuando esta estrategia exija un gran esfuerzo del diseñador del juego, las rutinas son tan primitivas que refe-



irse a ellas con el término de inteligencia artificial lleva a error y casi resulta arrogante. Prefiero el término mucho menos ambicioso de computación artificial.

La primera meta de todo sistema de cómputo artificial es producir una conducta razonable. El ordenador no debe conducir sus tanques por despeñaderos ni estrellar sus naves espaciales unas contra otras, o ponerse a descansar cuando está directamente frente a las armas de los humanos. En otras palabras, el sistema de cómputo artificial no debe permitir movidas obviamente estúpidas. Esta exigencia nos lleva a calcular toda movida posiblemente estúpida, redactar un código que las ponga a prueba y excluirlas. Esta es la manera equivocada de encarar el problema porque el ordenador puede demostrar una sorprendente creatividad en la estupidez de sus errores. Un mejor (aunque más difícil) método es crear un algoritmo más general que obvie las movidas más absurdas.

Una segunda exigencia de una rutina de cómputo artificial es la impredecibilidad. El jugador humano no debe poder

nunca adivinar la conducta del ordenador, pues esto le ahogaría la impresión de inteligencia facilitándole demasiado la victoria. Esta segunda exigencia parecería contradecir la primera, de conducta razonable, porque la conducta razonable sigue pautas que deben ser predecibles. Esta aparente contradicción podemos resolverla por medio de una más profunda comprensión de la naturaleza interactiva del juego. Para llegar a esa comprensión, debemos combinar tres premisas: Primero, la reacción a un opositor es en cierta manera un reflejo de ese opositor. Un jugador razonable trata de anticipar los movimientos de su oponente percibiendo la personalidad de este oponente. Segundo, la interacción es una reacción mutua: ambos jugadores tratan de prever las movidas del otro. Tercero, el nivel de interactividad es la medida de la calidad del juego. Podemos combinar estas tres premisas en una analogía. Pensemos en un juego en los términos de dos espejos frente a frente, con un jugador mirando desde cada uno de ellos. Un juego débilmente interactivo es como dos espejos casi frente a frente. Cada jugador puede ver e interactuar a uno o dos niveles de reflejo. Un juego ideal, perfectamente interactivo es análogo a dos espejos exactamente enfrentados. Cada uno de los dos jugadores cambia de lugar infinitamente en el túnel de reflejos anticipados. Por más razonable que sea la conducta, el infinitamente complejo patrón de la anticipación y la contraanticipación desafía toda posible predicción. El patrón es razonable aunque impredecible, satisfaciendo así el requerimiento del cómputo artificial.

La experiencia me ha demostrado que los algoritmos de los juegos son en su mayor parte predecibles cuando son particulares, cuando ponen el énfasis por sobre la esencia del juego todo, en un solo elemento. Por ejemplo, en los juegos de guerra, algoritmos tales como "determinar la unidad enemiga más próxima y dispararle" son muy particulares y comportan una conducta predecible.

Creo que los algoritmos mejores para conducta razonable e impredecible toman en consideración una gran cantidad de información en un contexto más amplio. Es decir, tomarán en cuenta a efectos de una decisión, un amplio número de consideraciones más que enfocar un pequeño número de elementos clave. Comparando con el ejemplo anterior, un mejor algoritmo puede ser "determine la unidad enemiga con la mayor combinación de amenaza y vulnerabilidad (basado en alineamiento, actividad, enfrentamiento, alineamiento con los otros tanques del ordenador, cubierta y vista); en ese caso, al disparar sobre la unidad serán mayores las probabilidades de matar que de ser matado".

¿Cómo se implementan esos principios en algoritmos específicos, programables? Dudo mucho que pueda llegar a ser descubierto un sistema que sirva en todos los casos. La mejor solución general que puede encontrar hasta ahora es una combinación de sistemas de puntaje, análisis de campo y cambios en la estructura del juego.

Primero, establezco un sistema de puntaje para cuantificar el mérito de cada posible movida. Esta es una técnica altamente respetada por varios sistemas de inteligencia artificial. El sistema de puntaje lleva mucha dedicación. Una complicación inicial es de orden dinámico: el diseñador debe garantizar que la probabilidad de dos movidas accesibles acumulará un valor de puntaje igual al máximo valor permitido por el tamaño de la palabra (8-bits) cercano a 0. En otras palabras, no podemos tener dos movidas cada una de ellas de 255 puntos, o no tendremos manera de saber cuál fue la mejor.

Otro problema que presenta el sistema de puntaje es el de balancear los factores uno contra otro. Supongamos que tenemos un juego de tanques en el que sabemos que trepar a la cima de una colina es bueno, pero que también sabemos que ir sobre una carretera también es bueno. ¿Cuál es mejor? Si la posición en la cima de una colina vale 15 puntos ¿cuánto debe valer una posición sobre la carretera? ¿10 puntos? ¿20? Es realmente imprescindible una profunda familiaridad con el desarrollo del juego para poder responder a estos interrogantes. Desgraciadamente, esa familiaridad es imposible de alcanzar con un juego que aún no se ha completado. Debe uno fiarse de la experiencia, de la comprensión del problema planteado, de un análisis agobiante y mucha experimentación.

Otra técnica de computación artificial, el análisis del campo, se aplica a los juegos de relación espacial. En esos juegos, el humano se basa en el reconocimiento de patrones para analizar posiciones y planear movidas. El microordenador es incapaz de verdaderos patrones de reconocimiento al mismo nivel que los humanos. Pero, de todas maneras, el análisis de campo le permite al ordenador alcanzar algo que podemos llamar patrón de reconocimiento. El esfuerzo fundamental del diseñador es la creación de una cantidad de campo calculable que pueda expresar correctamente la información crítica que necesita el ordenador para realizar una movida razonable. Por ejemplo, en varios de mis juegos de guerra, he usado los campos de seguridad y de peligro para decirle a la unidad el nivel de riesgo en una situación dada. El peligro se calcula dividiendo la fuerza de cada una de las unidades del enemigo por su distancia sumándole el cociente de las diferentes

unidades; así, las unidades grandes o muy cercanas son muy peligrosas y las pequeñas o distantes lo son menos. Un cálculo similar con unidades amigas da un factor de seguridad. Una unidad decide si dar muestras de conducta arriesgada o prudente, comparando desde su posición los valores de peligro y de seguridad. Una vez tomada esta decisión, la unidad mira en derredor y mide el peligro neto menos la seguridad de cada una de las posiciones a las que puede desplazarse. Si la unidad se siente arriesgada, se mueve hacia el peligro; si se siente tímida, se aleja de él. En este caso particular, encuentro que el campo vector es más informativo que la escala de campo; aunque en algunos casos este último resulta el más indicado. El campo vector dice la magnitud y la dirección del peligro. El campo a escala sólo dice la magnitud. Si el algoritmo es lo suficientemente inteligente como para usar la información de dirección, lo más útil es el campo vector.

Coordinar las movidas de las diversas unidades bajo el control del ordenador es un dilema muy especial. ¿Cómo lograr que el ordenador asegure la movilización de las diferentes unidades de una manera coordinada y que no sucedan embotellamientos de tránsito? Habitualmente, el diseñador de juegos responde a esto usando un sistema de planificación secuencial sumado a un simple test para determinar la posición de las demás unidades. Así, la unidad n.º 1 se mueve primero, luego la n.º 2, seguida por la unidad n.º 3, evitando cada una de ellas la colisión. Puedo asegurar —por experiencia propia— que este sistema sólo sirve para reemplazar las colisiones por los más tremendos embotellamientos de tránsito.

Un método mejor es usar un virtual sistema de movidas, en el cual cada unidad planea una virtual (hipotética) movida basada en las posiciones virtuales de las demás unidades. Funciona de esta manera: Comenzamos con una matriz de las verdaderas posiciones de todas las unidades del ordenador. Creamos la matriz de las posiciones verdaderas e inicializamos todos los valores virtuales a los reales. Luego, cada unidad planifica su movida, evitando las colisiones con las posiciones virtuales. Luego de que la unidad haya planificado su movida, la planeada posición final es ubicada dentro de la matriz virtual. Luego, otras unidades planifican a su vez sus movidas. Luego de que todas las unidades hubieron planificado una movida virtual, se repite el proceso, con cada unidad planificando su movida en la base de una matriz de movida-virtual interina. Este gran *loop* exterior deberá ser convergente; luego de un suficiente número de repeticiones, termina la rutina y las posiciones virtuales forman la base de las movidas realizadas por las

unidades del ordenador. Esta técnica es útil para coordinar las movidas de varias unidades y prever los embotellamientos de tránsito.

Otra técnica para lograr algoritmos razonables e impredecibles es tan simple que parece una broma: cambiar el juego. Si un elemento crucial del juego no es tratable con computación artificial, elimínalo. Si no puede dar con una buena manera de sacarle provecho a una característica, realmente no queda más opción que desecharla. Por ejemplo, mientras yo diseñaba *Tanktics*, me vi frente a un problema con algunos lagos. Si el ordenador se acercaba a un lago en forma de U desde la dirección equivocada, llevaría sus tanques hasta el final de la península formada por el lago, viendo que el agua los bloquearía de seguir adelante, se unen, dan vuelta cuidadosamente y vuelven a la península sólo para quedar otra vez bloqueados por el lago. El lago en forma de U creaba una trampa para mi algoritmo de computación artificial. Pasé una buena parte de mi tiempo trabajando con una rutina de cómputo artificial más inteligente que no se dejara atrapar por esos



lagos y haciendo, de paso, una buena economía de movimientos. Luego de ingentes esfuerzos descubrí la mejor solución: eliminar del mapa los lagos en forma de U. En el plano ideal, el diseñador de juegos con experiencia tiene suficiente intuición de los algoritmos como para presentir cuales son los factores de juego intratables y evitarlos durante las distintas etapas de diseño de los juegos. Sin embargo, la mayoría de nosotros llegamos a esos descubrimientos de la manera más ardua, por medio de modificaciones y retrazado de juegos.

No importa cuán bueno sea un algoritmo, su aplicabilidad es limitada. La realidad es que un algoritmo específico siempre será mejor trabajando en un número limitado de condiciones. Para que resulte realmente interesante, un buen diseño de juego deberá ofrecer una amplia gama de condiciones. Así, con varios juegos, un diseñador puede crear un determinado número de algoritmos y empalmar los unos con los otros, como cambio de condiciones. La transición de un algoritmo al otro implica cierto riesgo; de no mantenerse una continuidad durante la transición, las unidades del ordenador pueden llegar a demostrar pautas de conducta altamente irracionales. Recuerdo muy bien una transición de algoritmo muy frustrante en **Legionnaire**. Los bárbaros del ordenador estaban controlados por tres algoritmos que tanto los llevaban a correr en busca de resguardo, a acercarse para contactar o a atacar. Bajo determinadas condiciones, un bárbaro que operara bajo el algoritmo acercarse - para - contactar, decidiría una conducta intrépida: arrojarse hacia adelante para hacer contacto con el humano y hacer la transición para el algoritmo de ataque, que declararía poco seguro el ataque. El bárbaro frustraría así el ataque convirtiéndose al algoritmo ponerse-a-salvo, que le indicaría dar la vuelta y echar a correr. El jugador humano estaba destinado a un espectáculo de feroz carga y despavorida retirada de los bárbaros, ninguno de los cuales hasta ahora se había negado a combatir. Eventualmente me rendí y rediseñé los algoritmos fusionando los tres en un sólo algoritmo de avance-ataque sin transiciones.

Las técnicas de computación artificial descritas están diseñadas para juegos que impliquen relaciones espaciales. Muchos juegos son no-espaciales y exigen otras técnicas de computación. Un tipo común de juego no espacial es un complejo sistema de simulación, que a menudo involucran ecuaciones diferenciales acopladas. **Lunar Lander**, **Hammurabi**, **Energy Czar** y **Scram** son ejemplos de este tipo de juego. En juegos así, el primer problema que enfrenta el diseñador no es tanto la derrota del humano como un modelo astuto de sistema de conducta. Prevengo al

diseñador de juegos en general para que sea especialmente cuidadoso con los juegos que impliquen amplios sistemas de ecuaciones diferenciales acopladas. **Hammurabi** usa tres ecuaciones diferenciadas acopladas de primer orden y la mayoría de los programadores encuentran que es un número fácil de manejar. Pero la complejidad del problema aumenta con una rapidez proporcional al número de ecuaciones diferenciales usadas. **Energy Czar** usó la fantástica cantidad de 48 ecuaciones diferenciales, hecho sólo creíble por la cantidad de impedimentos que se les imponía. En general, el consejo es cuidarse de no usar más de cuatro ecuaciones diferenciales acopladas. Si es imprescindible usar muchas, hay que tratar de usar ecuaciones paralelas, en las cuales la misma ecuación fundamental se aplica a cada uno de los elementos de un conjunto de valores.

Cada ecuación diferencial deberá tener un factor de disminución de onda que deberá ser empíricamente ajustado:

Un pequeño factor de disminución de onda produce sistemas vívidamente simulados que saltan salvajemente. Un factor de disminución amplio produce sistemas que cambian lentamente. Desgraciadamente, el recurrir a simples factores de disminución puede resultar un tiro por la culata cuando existe una relación negativa de retroceso entre el nuevo valor y la fuerza de conducción. En este caso, un amplio factor de disminución inhibe el retroceso negativo y una de las variables cambia erráticamente. La conducta de los sistemas de las ecuaciones diferenciales es compleja; sugiero que los diseñadores interesados en estos problemas se familiaricen con las matemáticas de la sobredisminución, subdisminución y los sistemas oscilatorios críticamente disminuidos. Por información más generalizada sobre solución de sistemas de ecuaciones diferenciales, cualquier texto de análisis numérico servirá de guía útil.

La aplicación de todos estos métodos bien puede producir un juego de cierta inteligencia, pero las expectativas no deben ser exageradas. Ni siquiera un denodado esfuerzo basta para producir un juego verdaderamente inteligente. En realidad, hasta la fecha ninguno de mis juegos es lo suficientemente inteligente como para vencer a un jugador humano en igualdad de condiciones. En realidad, aún necesitan una ventaja de relación de fuerza de 2 a 1 para poder luchar con un jugador humano.

Información limitada

Otra manera de compensar la debilidad de la inteligencia del ordenador es limitar la cantidad de información accesible al jugador humano. Si el humano

carece de información para procesar, no puede ampliar su poder superior de procesamiento al problema. Aplicar esta táctica hasta el extremo, puede llegar a reducir el juego al puro azar. Pero usándola con discreción, la limitación de información puede igualar las probabilidades entre el jugador y el ordenador. Si la información es retenida dentro de un contexto razonable (ejemplo: el jugador deberá enviar exploradores), las restricciones de información sólo serán naturales, sino que aumentarán el realismo y el suspenso del juego.

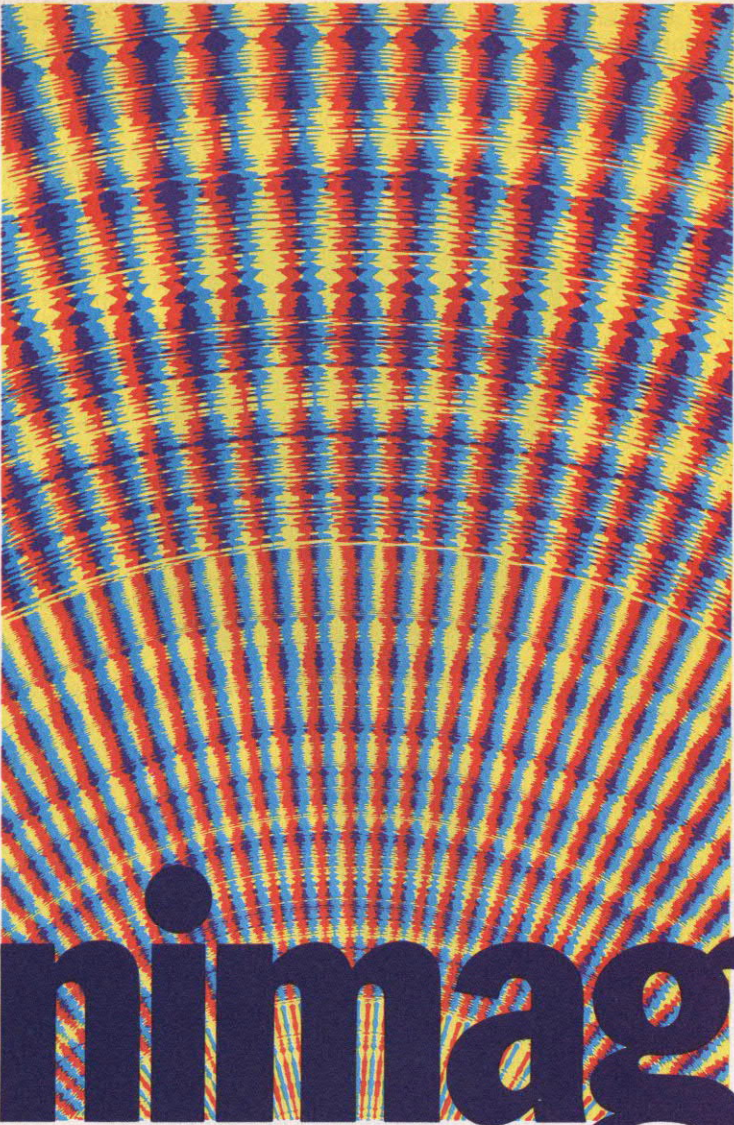
Los diseñadores de juegos suelen sobervaluar o desvalorizar la importancia de la información limitada. La información limitada puede ser un acicate para la imaginación del jugador, sugiriéndole detalles sin llegar a confirmárselos. Hay que elegir con cierto arte, con todo, las limitaciones de la información. Los datos dados al azar suelen ser más confusos y frustrantes que acicateantes.

Otra manera de balancear la fuerza entre el ordenador y el jugador humano es controlar el desarrollo, la marcha del juego. El humano puede ser listo, pero el ordenador es más rápido al efectuar simples computaciones. Si la marcha es lo suficientemente rápida, los jugadores humanos no tendrán suficiente tiempo para aplicar sus pautas superiores de procesamiento y quedarán confundidos. Esta es una técnica fácil de aplicar, pero eso a nadie sorprende que muchos diseñadores de juegos de acción y astucia la usen a menudo.

Estas cuatro técnicas —vastedad de recursos, computación artificial, información limitada y marcha— nunca se emplean aislados. Cada juego usa una combinación de los cuatro. La mayoría de los juegos se basan primero en la marcha y en la vastedad de recursos para balancear, con muy poca computación artificial o información limitada. La razón para este énfasis es muy simple: la marcha y la vastedad de recursos son fáciles de implementar, mientras que la computación artificial y la información limitada son más difíciles. La economía de esfuerzos ya no se justifica en el lucrativo mercado de los juegos. No hay razón alguna por la que los juegos no puedan incorporar las cuatro técnicas. Al contrario, esto tendría que coadyuvar al éxito del juego, porque al usar un poco de cada uno de los cuatro métodos, el juego no necesita constreñirse a las limitaciones de ninguno de ellos. El diseñador deberá decidir el balance apropiado de cada una de las metas de un juego en particular.

Relación entre los oponentes

Los conceptos planteados hasta ahora se aplican fundamentalmente a los juegos computados. Consideremos ahora alguna teoría sobre el juego en genera-



Sonimag 21

21 Salón Internacional de la Imagen, el Sonido y la Electrónica

Barcelona, 26 de Septiembre al 2 de Octubre 1983

**ANTENAS, EMISORAS DE RADIO Y TV,
HI-FI DOMESTICO Y PROFESIONAL,
ILUMINACION ESPECTACULAR,
INSTRUMENTOS MUSICALES,
ORDENADORES PERSONALES,
RADIOAFICION, TV Y RADIO PROFESIONAL,
TV Y VIDEO DOMESTICO E INDUSTRIAL,
VIDEOJUEGOS, VIDEOPRODUCCIONES,
FOTOGRAFIA.**

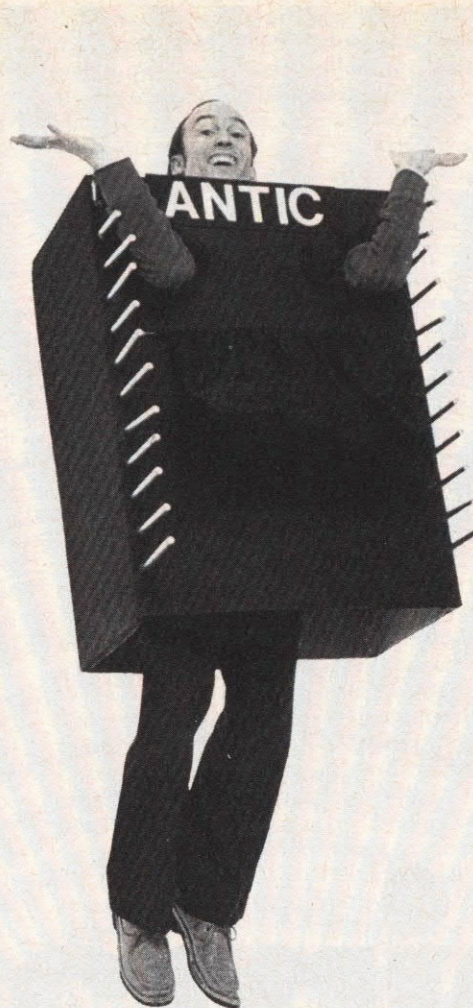
Todo juego establece una relación entre oponentes que cada jugador deberá explotar al máximo a efectos de la ventaja. La arquitectura fundamental de esta relación, **simétrica o asimétrica**, tiene un rol central en el juego. La arquitectura más simple establece una **relación simétrica** entre los dos jugadores. Ambos tienen las mismas propiedades, la misma fuerza, la misma debilidad. Los juegos simétricos tienen una obvia característica deseable: son automáticamente balanceados. Porque el mismo proceso se aplica a cada uno de los jugadores, los juegos computados simétricos tienden a ser más fáciles de programar. Finalmente, estos juegos son más fáciles de aprender y comprender. Ejemplos de juegos simétricos son **Combate**, de Atari 2000, **Basketball** y **Dog Daze**.

Los juegos simétricos padecen una variedad de debilidades, la mayor de ellas su relativa simplicidad. Cualquier estrategia que prometa ser realmente efectiva será usada por ambos bandos simultáneamente. En ese caso, el éxito deriva, no del planeamiento sino de la ejecución. En cambio, el éxito del juego se basa en detalles muy sutiles. Un buen ejemplo, de ellos es el ajedrez: una ventaja de sólo un peón puede redundar en victoria.

Dada la debilidad de los juegos simétricos, muchos diseñadores de juegos tienden a establecer una relación asimétrica entre los oponentes. Cada jugador tiene una única combinación de ventajas y de desventajas. El diseñador del juego tiene que balancear, en cierto modo, las ventajas, para que la victoria sea posible para ambos jugadores con un nivel igual de riesgos. La manera más simple de lograrlo es por medio de la asimetría plástica. Los juegos que incorporan esta relación son formalmente simétricos, pero los jugadores eligen rasgos iniciales acordes a determinado conjunto de restricciones.

Otros juegos tienen una relación asimétrica más explícita. Casi todos los juegos computados solitarios establecen una relación asimétrica entre el jugador computado y el humano, dado que el computado no puede esperar jugar con el humano en igualdad de condiciones de inteligencia. Así, el diseñador le da recursos al jugador humano para que use su superior capacidad de planeamiento, mientras que el ordenador es provisto de recursos que compensan la debilidad de su inteligencia.

La ventaja de los juegos asimétricos estriba en la habilidad de establecer relaciones triangulares o no transitivas dentro del juego. La no transitividad es una propiedad matemática bien definida. En este contexto, se la ilustra mejor con el juego roca-tijera-papel. Son dos los jugadores. Cada uno de ellos elije secretamente un rol y lo anuncian simultáneamente, comparando su elección. Si ambos realizan la misma elección, hay empate y se repite el juego. Si



la elección es diferente, la roca parte la tijera, la tijera corta el papel y el papel envuelve la piedra. Esta relación en la que cada uno de los componentes puede vencer al otro y ser vencido a su vez, es una relación no transitiva. Porque el hecho de que la roca rompa las tijeras y de que las tijeras corten el papel no implica que la roca venza al papel. Esta particular relación no transitiva sólo produce resultados claros con tres componentes, para que cada uno de ellos tenga relación con los otros dos solamente; vence a uno y pierde con el otro. Un juego de roca-tijeras-papel trasladado al lenguaje binario —gana o pierde— no puede ser llevado a cabo con más de tres componentes. Se podría hacer un juego con múltiples componentes si se incluyera distintos niveles de victoria (tal vez con sistema de puntaje).

La no-transitividad es una propiedad matemática muy interesante eso no permite gran riqueza de juegos si nos atenemos al sentido estrictamente matemático del término. Su valor para el diseño de juegos reside en la generalización del principio en áreas menos exactamente definidas. Empleo el término triangular para describir esas relaciones asimétricas que extienden los conceptos

de no-transitividad más allá de su definición formal.

Un simple ejemplo de relación triangular surge en el juego **Battlezone**. Cuando aparece un platillo volador, el jugador puede perseguirlo en vez de perseguir a un tanque enemigo. En ese caso, hay tres componentes: jugador, platillo volador y tanque enemigo. El jugador persigue al platillo volador (lado uno del triángulo) y es perseguido por el tanque enemigo (lado dos). El tercer lado del triángulo (platillo volador al tanque enemigo) no atañe directamente al humano —el ordenador maneja el platillo para llevar al humano a una peor posición. Este ejemplo es fácil de comprender porque la triangularidad asume tanto una forma espacial como una estructural.

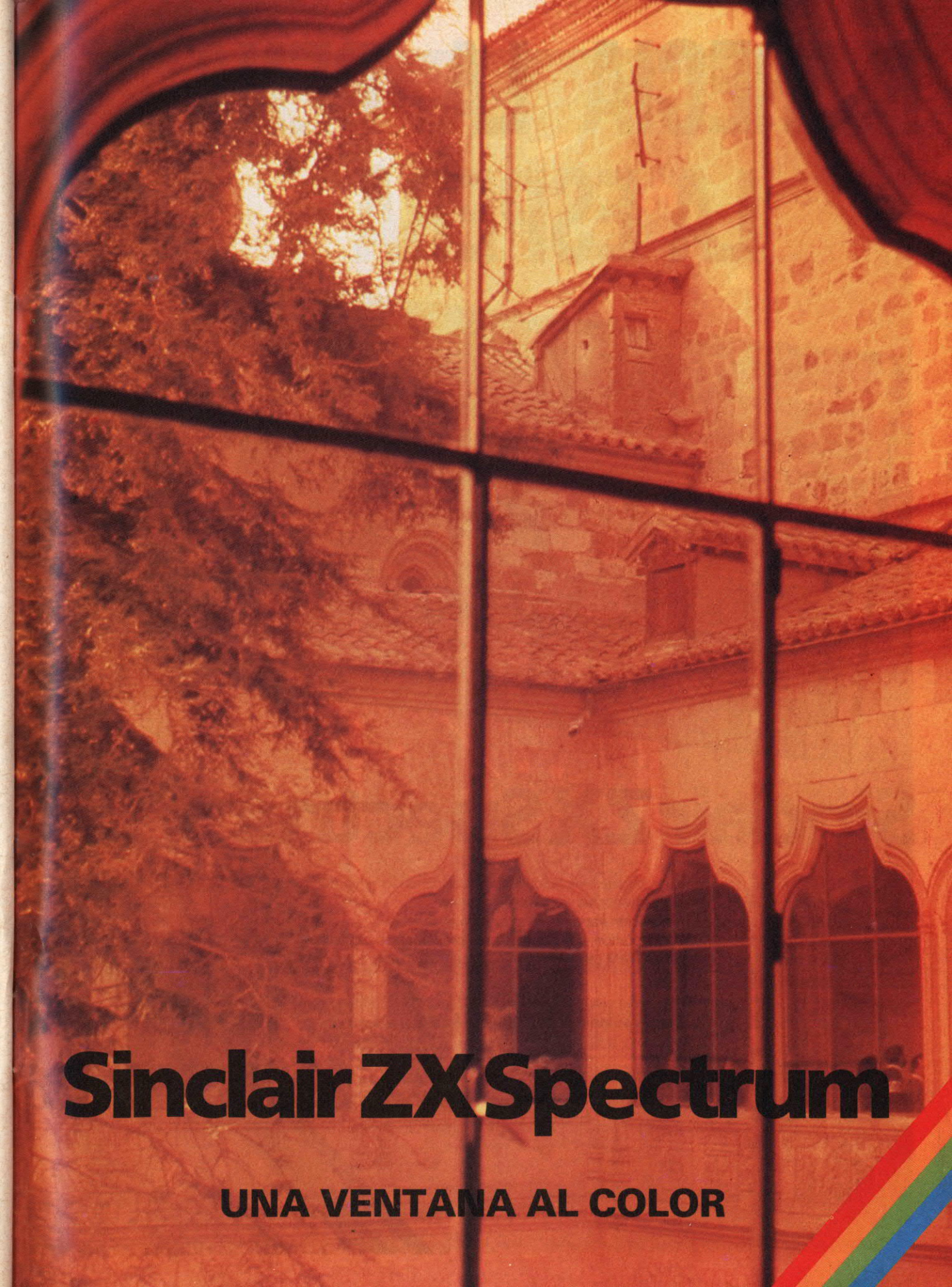
Ilusión de posibilidad de ganar (competitividad)

Una característica importante de todo juego es la ilusión de posibilidad de ganar. Si un juego va a presentar un continuo desafío a los jugadores, también debe presentar una continua motivación para seguir jugando. El juego debe evidenciar la posibilidad de ganar para todos los jugadores, principiantes y expertos, aunque nunca debe podersele vencer completamente, pues perdería su atractivo. Esta es una ilusión muy difícil de mantener. Algunos juegos la mantienen para los expertos, pero no lo logran con los novatos; intimidan a todo el que no sea un avezado jugador. El juego más exitoso en este aspecto es **Pac-Man**, que evidencia posibilidades de ganarle para la mayoría de los jugadores, aunque en realidad no se lo puede vencer completamente.

La simplicidad o claridad del juego es el factor más importante en la impresión de poder ganar. Un juego sucio intimida a quienes comienzan, por el exceso de detalles. Muchos principiantes no se sobreponen nunca a la inhibitoria sospecha de que en algún momento del juego va a surgir una situación gordiana, algún detalle complicado o factor oculto que aún no ha descubierto. En contraste, un juego nítido alienta a todos los jugadores a experimentar con el juego en cuanto éste aparece.

Otro factor fundamental es mantener la esperanza de poder ganar que aumenta a medida que el jugador analiza sus errores. En todos los juegos se espera que el jugador cometa errores. ¿Qué lo alienta a seguir adelante? Si cree que su error se debió a un defecto de los controles o del curso del juego, se sentirá frustrado y furioso con lo que en todo su derecho considera una situación injusta que nunca podrá superar. Si el jugador percibe que su error es resultado de sus propias limitaciones y decide que ganar le exigirá un esfuerzo.

Chris Crawford
© Byte/Ordenador Popular



Sinclair ZX Spectrum

UNA VENTANA AL COLOR

MERECIA LA PENA LA ESPERA



Sinclair ZX Spectrum

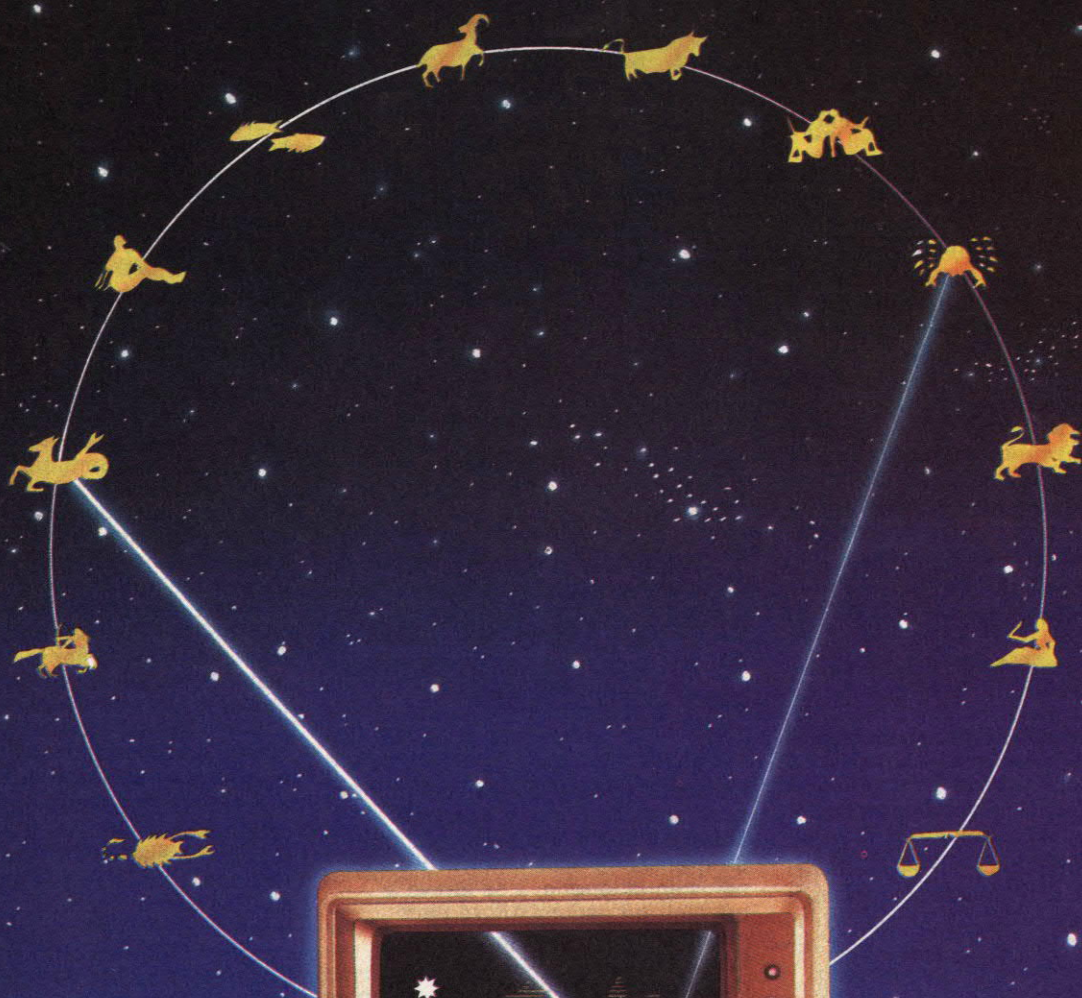
- **Color total:** ocho colores disponibles para primer plano, fondo y bordes, independientes entre sí. Control de intensidad de brillo e intermitencia.
- **Sonido** de intensidad y duración variable. Altavoz incorporado.
- **Alta definición:** 256 × 192 puntos, con ocho colores presentes en pantalla.
- **Dos modelos:** — 16 K de RAM (ampliables). — 48 K de RAM.
- **Teclado móvil.**

Solicite mayor información a su distribuidor habitual autorizado.



INVESTRONICA

DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA
Tomás Bretón, 21. Teléfono (91) 468 03 00. MADRID-7
Muntaner, 565. Teléfono (93) 212 68 00. BARCELONA



EL ZODIACO EN 8 BITS

Se dice que los grandes cambios previstos para 1983 —que algunos relacionan con guerras y catástrofes— en realidad pueden interpretarse como el anuncio de la revolución informática. Así opinan algunos astrólogos que reniegan de una visión pesimista del futuro del mundo y que, a la vez, consideran a la astrología como un método fiable —que no infalible— de conocer las tendencias del universo y los individuos.

La astrología ha generado, desde un principio, una ardua polémica sobre si es o no es una ciencia. Según los entendidos, es difícil determinar el peso de la intuición en la interpretación de una carta astral. Sin embargo, al basarse en la astronomía los datos empíricos y al contar con millones de casos aportados por centenios de experiencias le permite diferenciarse de las ciencias ocultas, que trabajan fundamentalmente a partir de ciertas propiedades especiales de determinadas personas.

Para la informática es importante determinar si la astrología es una ciencia, un timo, o un don. En realidad, de lo que se crea que es depende mucho el futuro de las aplicaciones para el universo astral.

Actualmente nadie duda de la utilidad de los ordenadores y de programas especiales para facilitar los cálculos. Algunos creen en el acierto de las interpretaciones obtenidas por ordenadores y muy pocos confían en el desarrollo de la inteligencia artificial aplicada a programas expertos de astrología.

No es cuestión de resolver ahora una discusión que se remonta a siglos. Desde Tolomeo, que situó a la Tierra en el centro del sistema, la astrología fue endiosada y denostada por períodos y mucho más aún luego de Copérnico, a pesar de los estudiosos que reivindican el valor de la astronomía aunque dicen que las influencias astrales dependen del lugar donde se ha nacido.

Tampoco es cuestión de desdenar un aporte de datos recabados en milenios y que recién ahora, en algunos aspectos, mediante el uso de ordenadores, pueden ser sometidos a comprobación. Por lo menos así lo cree el matrimonio Gauquelin, que ha realizado dos estudios en base a 25.000 fechas distribuidas por profesiones y sobre herencia natal, en las que se ha confirmado el funcionamiento tradicional. Incluso el personal de la NASA que elaboró el más completo libro de efemérides —*The Concise Planetary Ephemerides*— con un IBM 370/168, y que sirve de material básico en el trabajo de los astrólogos. Existen muchos investigadores que, sin ser astrólogos, realizan estudios en la materia. En Estados Unidos existe un servicio de control de acontecimientos históricos que hacen la carta astral de personajes relevantes y establecen una comparación con la coyuntura del período.

En Wall Street es bien sabido que muchas operaciones han sido anuladas



Elisabeth Teissier: Al mundo le espera una gran crisis

Francesa, una de las más conocidas astrólogas de Europa, **Elisabeth Teissier** trabaja, por supuesto, con programas de ordenador. No tiene pelos en la lengua, y se atreve a predecir un oscuro futuro para nuestro plante planeta.

P.: ¿Qué dice la astrología acerca del futuro de la humanidad?

R.: Reflexiono mucho para averiguar a dónde debemos ir. Esta es la cuestión. La astrología es mucho más precisa en el cuándo. El dónde es mucho más difícil de determinar. Contamos con reglas para ello, pero no es seguro que las tradicionales sean buenas. De ahí que recurramos a los ordenadores.

P.: Cuando habla de dónde debemos ir, ¿a qué se refiere? ¿A un lugar geográfico?

R.: Precisamente a eso. No es tan fácil. Hay una historieta macabra de alguien que en la última Guerra Mundial se dijo: "Me huelo que las cosas van realmente mal". Y se refugió en una isla solitaria del Pacífico, Pearl Harbour, que al principio era un escondrijo pequeño y tranquilo.

P.: Pero entonces usted cree que es inevitable la Tercera Guerra Mundial?

R.: Ya en 1975 pronostiqué en Francia que entre 1981 y 1984 se produciría un cambio de alcance mundial. Pienso que el cambio proporcionará al mundo, a la Humanidad, la ocasión de un gran

progreso, a la vez que una gigantesca crisis que puede dar lugar a una Guerra Mundial. A finales de este año se dará una conjunción de Saturno y Plutón que es muy grave y que de hecho en el pasado ha estado ligada a guerras. Con las dos últimas guerras mundiales. Hay buenos motivos para sentirse preocupado.

P.: O sea que la Humanidad puede desaparecer.

R.: No, no lo creo así. He estudiado en mi ordenador la fecha 11 de agosto de 1999 —señalado por Nostradamus— y la forma de presentarse los astros en ese día es absolutamente terrible. He observado esa fecha con un miedo cerval, porque todos nosotros, los que estamos entre los 20 y 50 años de edad, normalmente viviremos para entonces. Será ciertamente un gran día.

P.: ¿Y qué sucederá?

R.: Me pregunto qué puede significar ese fenómeno, pues Nostradamus afirma que puede ser el fin del mundo o el principio de otra civilización. Habla también del Anticristo. Puede ser la fecha de su nacimiento. Pero todavía estamos bastante lejos de ese día, y creo que sobreviviremos al período 1981 a 1984. Quizás no todos nosotros.

P.: Volvamos a temas menos dramáticos. ¿Ha elaborado usted su programa?

R.: No, no estoy muy fuerte en matemáticas. Las de la astrología son tan complicadas que nunca hubiera podido elaborar mi programa. Lo he comprado en New York a la empresa **Digicomp**. Lo hice porque así puedo obtener horóscopos desde 10.000 años antes de Cristo hasta otros 10.000 después de Cristo. Los ordenadores ayudan, ya que los pronósticos requieren muchísimo tiempo. La obtención de pronósticos ha costado a los astrólogos del pasado noches y semanas, mientras que con el ordenador el resultado se obtiene en pocos minutos. Es algo fantástico, pues en otro caso no podrían utilizarse algunos de los métodos. No se puede estudiar a cada cliente días o semanas enteras. Habría que cobrar una fortuna.

P.: ¿Qué podría hacer un especialista en ordenadores que comprara un programa sin tener conocimientos astrológicos?

R.: Nada en absoluto. El ordenador imprime un papel con un dibujo que es la esquematización del horóscopo, por ejemplo. En él aparecen los ascendentes, el zenit, las casas, las seis puertas con los diferentes planetas en ellas. Todo eso hay que interpretarlo. Este es el trabajo más pesado pero ahora sin las incomodidades del cálculo.

por culpa de un astrólogo, y sin ir más lejos, nuestros entrevistados confirmaron —sin dar nombres— que entre sus clientes españoles hay un gran número de directivos de banca y hombres de empresa que les consultan sobre operaciones puntuales. Gerardo Sánchez, conocido astrólogo español, tiene un contrato con algunas estaciones de invierno para "predecir" el futuro meteorológico.

Sally Quispert: Las ventajas de usar ordenador

Es norteamericana, pero vive en Madrid desde hace más de una década. Estudió en la Escuela de Astrología de Estados Unidos, internacionalmente reconocida. Se ha especializado en las aplicaciones de la astrología a la predicción económica. Lo que le ha valido ser consultada por personajes del mundo financiero a los que, desde luego, omite nombrar.

P.: ¿En qué le beneficia usar este ordenador (un Apple II) en esta profesión?

R.: Gano mucho tiempo y me puedo dedicar más a la interpretación y a hacer estudios, como el de la influencia de los astros sobre una persona en distintos lugares del planeta. Es un tipo de trabajo que antes dejaba de lado por la cantidad de cálculos que exigía. Estoy encantada desde que uso el ordenador.

P.: ¿No hay cierta resistencia de los astrólogos a usar computadores?

R.: Nadie discute ahora su utilidad, sólo hay polémica sobre qué modelo conviene usar. Yo no entiendo mucho del tema, pero estoy contenta con el Apple II, aunque a veces me

Toda la acumulación de datos sobre las estrellas y sus influencias no alcanzan, sin embargo, para establecer interpretaciones causales. Es decir aunque se sabe que la conjunción Saturno-Plutón, que se dará a finales de año, preanuncia situaciones graves, no es posible determinar con veracidad que se desencadenará una guerra. Si agosto de 1999 se presenta malo (porque todos los astros mantendrán un mal ángulo con los

pregunto cómo hacer para aprovecharlo más.

P.: ¿Qué programas usas?

R.: Los **Astro-Graphics** que me envían de Estados Unidos y le he mandado a hacer una adaptación al castellano. Uso de los programas más avanzados para cálculos y también otros que incluyen una interpretación.

P.: ¿Son correctas esas interpretaciones?

R.: No se las puede tomar demasiado en serio, pero a alguna gente le divierte tenerlas y se las facilito aclarándoles que no tiene un verdadero trabajo de interpretación.

P.: ¿Hace mucho que se dedica a la astrología?

R.: Pues sí. Antes trabajaba de cosmetóloga, pero comencé a estudiar en la escuela de profesionales de mi país, y desde ese momento me dediqué de lleno a la astrología.

P.: ¿Cuánto de verdad hay en la astrología?

R.: Puedo asegurar sin duda, que en determinado período a una persona le conviene estar en Nueva York en vez de España, o que un negocio tiene grandes posibilidades de salir bien.

P.: ¿Y que se acaba el mundo?

R.: No. Eso nadie puede afirmarlo con seriedad.

demás) depende de las tendencias más o menos catastróficas de los astrólogos que vean en ello el anuncio del fin del mundo o un momento de profundo cambio. Ciertamente, de acuerdo a los expertos, es más fácil establecer la influencia astral en los aspectos psicológicos de una persona y el peso que determinadas conjunciones tienen sobre las enfermedades.

Esta dificultad de establecer hechos ciertos más que tendencias influye en el desarrollo de los programas de astrología. En Estados Unidos, **Digicomp** de Nueva York y **Astro-Graphics Services** de Massachusetts, se especializan en *software* para uso de amateurs y profesionales.

El **Astro-Scope** sobre la base de hora de nacimiento, fecha y lugar brinda una carta natal elaborada sobre la combinación de 1.800 palabras. Se puede adquirir otros programas adicionales que permitirá establecer relaciones entre cartas de dos personas para conocer el grado de afinidad.

El **Superprogram** es específicamente para profesionales y ofrece una amplia posibilidad de cálculos referidos a carta natal, influencia lunar y otros.

Los profesionales a la vez que reconocen la utilidad de los ordenadores en cuanto al ahorro de tiempo invertido en cálculos, la posibilidad de almacenar un archivo y otros aspectos de consulta e investigación que manualmente resultarían muy difíciles de realizar, expresan el escaso valor de las interpretaciones que actualmente ofrecen las máquinas y el profundo escepticismo sobre un futuro en el que éstas lograrán reemplazarlos. Difícilmente un astrólogo profesional negará la importancia del aspecto intuitivo en la interpretación de una carta.

Existe un cierto rechazo del profesional al uso masivo de estos programas que sirven para montar negocios de carta natal al paso, aunque no dejan de reconocer el aspecto de juego que permite desmitificar una temática ancestralmente relacionada con lo desconocido.

Si, por un lado, la introducción de la informática en el campo de la astrología es beneficiosa, por otro pone en peligro, para algunos, el valor de los años de experiencia acumulados, por reducir a escasos textos una interpretación que se convierte en una ensalada de astros sin manos expertas para condimentarla.

Para los catastrofistas la cuestión es aún más peligrosa, ya que el avance en la comprobación de los datos aportados por la astrología marcará más claramente la barrera entre ciencia e intuición. Aunque se reconozca el don de alguna gente, también se sabe que es un buen terreno para el timo. Predecir guerras, revoluciones, fin del mundo, genera miedo. Y el temor, a veces, es rentable.



17.45 / 27-11-52
A.R.M.C. = 22 h. 7.50 m.

Casas	Signo
M C	Acuario
11	Aries
12	Tauro
Asc	Géminis
2	Cáncer
3	Leo

Planetas

	El Sol	Sagitario
	La Luna	Aries
R	Mercurio	Sagitario
	Venus	Capricornio
R	Marte	Acuario
	Júpiter	Tauro
R	Saturno	Libra
	Urano	Cáncer
	Neptuno	Libra
R	Plutón	Leo

Una muestra hecha en NewBrain del gráfico que proporciona el ordenador con especificación de casas, planetas y signos para elaborar una carta natal.

Arturo González: Ni recetarios ni catastrofistas

Profesor de la Escuela de Cultura Astrológica de Madrid, miembro del colectivo que edita la revista especializada Kepler, **Arturo González** es también parte de Urania, una empresa dedicada a la elaboración de *software* para astrólogos.

P.: ¿Cómo nace un astrólogo?

R.: Yo, personalmente, a los 15 años compré mi primer libro de astrología. Luego, pasé nueve años estudiando y todavía sigo. En un primer momento, me dediqué al estudio y la investigación, pero luego, comencé a hacer cartas como profesional. En este sentido, es una profesión como cualquier otra. Incluso es preferible a muchas, porque no hay que aguantar a un jefe.

P.: ¿Y desde cuándo trabaja con ordenador?

R.: Comencé hace ahora cuatro años, cuando salió el **Texas 59**, que tenía un juego de programas con el que me inicié. Actualmente, trabajo con un **Sinclair ZX81**, con mis propios programas, que elaboramos



en colaboración con los integrantes de Urania. Y en este momento estoy en un proyecto para el **HP 41C**.

P.: ¿Qué opina de las interpretaciones por ordenador?

R.: Pues que las que ahora se ofrecen son un recetario. No aportan mucho, y cualquier persona que se compra un libro de astrología elemental puede hacerlo mejor. El problema es que esos programas calculan cartas en determinados aspectos y van empalmando textos, pero no combinan significaciones de modo inteligente.

P.: Pero en un futuro, con el desarrollo de la inteligencia artificial...

R.: Tengo mis dudas sobre los resultados de esos proyectos. En astrología se trabaja con elementos cualitativos. No se manejan números sino símbolos...

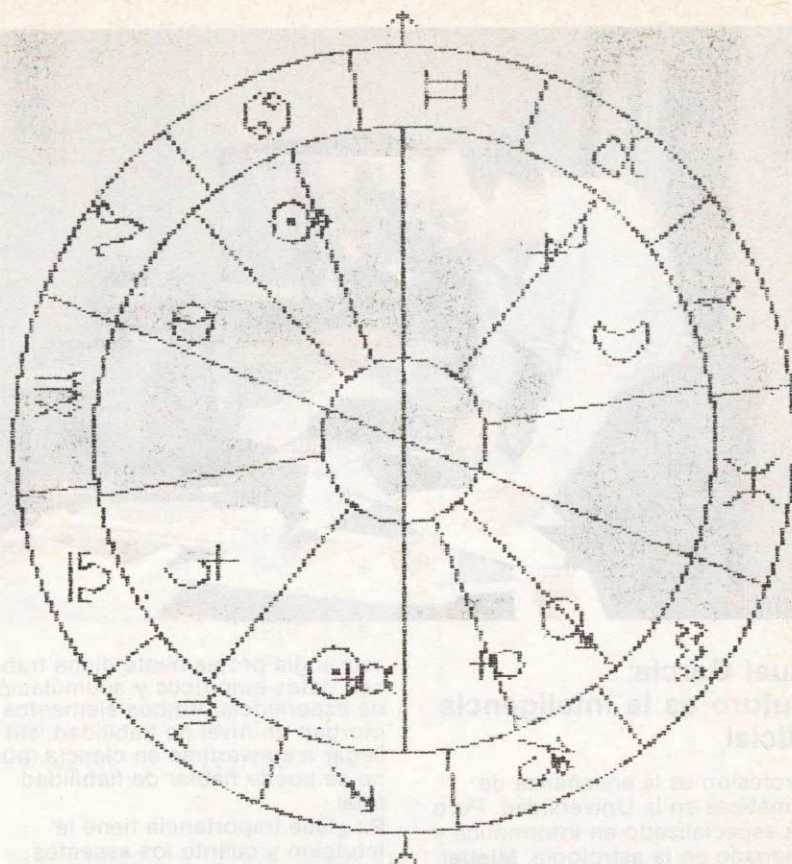
P.: Pero la interpretación de fechas suele llevar a anuncios de guerras, catástrofes, atentados...

R.: Como en todas partes, hay astrólogos chapuceros que viven de hacer preguntas de ese tipo. La astrología no está en el nivel de hacer tales apreciación predicciones; sólo ve las tendencias que pueden ser de violencia, y en las épocas de tensión a nivel mundial los planetas tienden a estar juntos. Lo que sucede es que si dices que habrá una gran catástrofe en 1983, seguro que vendes.

Gr	Mn
29	46.35
2	18.16
12	46.17
22	9.10
13	28.34
4	45.32

Velmin/día)

5	26.8	60.76
17	45.7	836.34
11	45.4	— 71.82
14	23.8	72.27
4	29.8	45.78
13	17.4	— 6.99
23	29.7	6.08
18	3.1	— 1.69
22	58.5	1.76
23	15.8	— .09



Pedro Chacón: De dones y razón

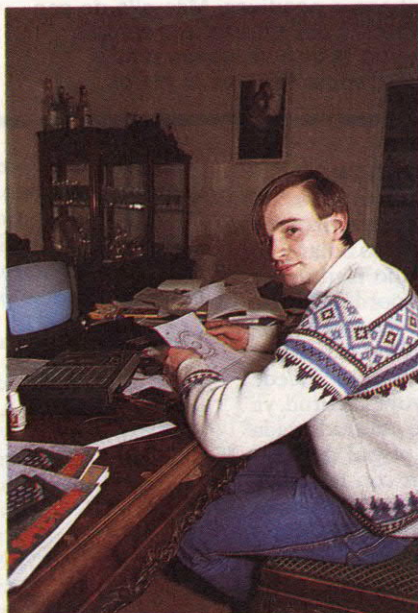
Es un Aries con ascendiente Géminis. **Pedro Chacón** ha estudiado astrología y actualmente trabaja en programas de *software* especializado para la investigación. Es el más joven de los entrevistados por esta revista, y un forofo del **Sinclair ZX81**.

P.: ¿Cómo entraste en contacto con la astrología?

R.: Tenía afición por las ciencias ocultas. Así, en general. Me atraía el tarot, la parapsicología, etc., y la astrología sólo la conocía por los periódicos. En principio, me parecía imposible que tantas personas, tal como mostraba el horóscopo, pudieran coincidir en lo que les fuera a pasar, por pertenecer a un mismo signo astral. Entonces cayó en mis manos un manual de astrología, que me aclaró las ideas. De esto hace ya cinco años.

P.: ¿Se necesita ser vidente para interpretar a los astros?

R.: No creo que la astrología sea una ciencia adivinatoria. Quien practica las ciencias ocultas posee un don, mientras que la astrología se estudia,



es una ciencia, entre comillas, empírica.

P.: ¿Y qué papel juega un ordenador en este contexto?

R.: A mí siempre me ha gustado las

máquinas y utilizo el ordenador como una herramienta que me ayuda a hacer cálculos y otras tareas como la que estoy realizando actualmente que es el desarrollo de un programa con el objeto de archivar cartas natales.

P.: ¿Posees el don de la interpretación?

R.: Me considero un investigador astrológico, no un astrólogo y no hago cartas personales.

P.: ¿Estamos o no en la era Acuario?

R.: No se sabe. Desde el comienzo del Zodíaco se marcan 2000 años por constelación, pero muchas veces se solapan dos constelaciones, como ahora sucede con Piscis —Acuario, y es imposible establecer el comienzo exacto de una era.

P.: Se dice que los astros, coincidiendo con el comienzo de la era Acuario, provocarán peligrosos estallidos de violencia.

R.: Realmente nos hallamos en un período de cambio, en el que se han producido y van a producirse grandes conjunciones de los planetas. Cambio no quiere decir necesariamente guerra o destrucción. De hecho, así lo ha sido ya que cada determinados períodos se dan situaciones semejantes a la actual.



Miguel García: El futuro es la inteligencia artificial

Su profesión es la enseñanza de matemáticas en la Universidad. Pero se ha especializado en informática y apasionado en la astrología. **Miguel García**, conjuga todas estas vocaciones desarrollando *software* astrológico que comercializa con un grupo de amigos. Realizan programas a medida.

P.: ¿Cree usted que la astrología es una ciencia?

R.: Sería muy discutible. Tiene mucho de arte. La parte de cálculo es astronomía, mientras que la

astrología propiamente dicha trabaja con datos empíricos y acumulación de experiencia. Ambos elementos le otorgan un nivel de fiabilidad, sin llegar a convertirse en ciencia, pues no se puede hablar de fiabilidad total.

P.: ¿Qué importancia tiene la intuición y cuánto los aspectos científicos?

R.: No se puede decir. En astrología se pueden determinar en el nivel psicológico tendencias generales, al igual que en lo que llamaríamos destino se pueden mostrar períodos críticos o positivos. Más allá de eso depende del astrólogo y en este campo el que más sabe y más experiencia tiene es el mejor.

P.: ¿Cómo te acercaste al campo de la astrología?

R.: Además de las matemáticas

siempre me interesó la psicología, y la astrología es un buen modo, el más fácil de conocer a la gente. Tenía esa inquietud cuando comencé a trabajar con un ordenador **IBM/360** que tenía un programa que dibujaba un mapa del cielo. Casi como un juego, en momentos libres intentaba hacer cálculos astrológicos. Pensaba que podía utilizarlo, pero no fue así: el ordenador no estaba preparado para ello aunque ayudó a que me iniciara en el tema.

P.: ¿Cómo pasaste a hacer programas?

R.: Estuve un par de años como autodidacta. Compré las tablas de la NASA hasta que a través de un amigo me vinculé al Centro de Cultura Astrológica. Una cosa trae la otra, porque comencé a hacer programas para nuestro uso.

P.: ¿Y actualmente?

R.: Tengo un programa de cálculo completo de cartas sin necesidad de pasar a efemérides y otro de comparación de cartas.

Generalmente trabajo con **New Brain**, aunque también realizo programas a medida para otros modelos.

P.: ¿Pero crees que se puede ir lejos usando ordenadores en la interpretación?

R.: Esa es mi intención. El terreno que me gusta es el de la inteligencia artificial y creo que se puede hacer un programa experto. No se cuándo pero espero acabar haciendo eso e ir sacando conclusiones técnicas.

Cálculo del signo solar y del ascendente

Este programa ha sido desarrollado para permitir a los lectores de Ordenador Popular averiguar su Signo Solar y su Ascendente. Hemos procurado hacer un listado fácilmente adaptable al BASIC de la mayoría de los microordenadores.

El primer escollo que encontrarán los usuarios es averiguar la hora de nacimiento. La fuente más fiable es la familia, pero una "copia literal de la partida de nacimiento" puede valer. Debe determinarse esta hora con la mayor exactitud, ya que el Ascendente se mueve a una media de un grado cada cuatro minutos. La diferencia entre la hora oficial de nacimiento y la de Greenwich se puede averiguar con facilidad, excepto cuando rijan hora de verano. En este caso la consulta de un manual de iniciación a la Astrología en cualquier librería puede proporcionarle el dato. La posición geográfica del lugar de nacimiento puede obtenerse con un mapa (basta aproximarse en medio grado).

El programa presentará el Signo Solar y el Signo Ascendente, especificados en grados y minutos. Cada signo tiene 30° y podemos considerar los tres últimos de transición al siguiente. Así, un Sol a 28° de Escorpio ya recibe una cierta influencia Sagitario, un Ascendente a 29° 1/2 de Cáncer prácticamente se puede considerar Leo.

Uso del programa

El programa pide los datos por orden. La fecha se debe introducir como AÑOSMEDIA (año: cuatro dígitos; mes: dos; día: dos). La hora de 0 h. a 24 h. Al emplear notación internacional Oeste se abrevia como "W".

Ejemplo: Ana, nacida en Madrid (40° 24' norte, 3° 42' Oeste) el 18 de junio de 1958 (1958/618) a las 2 y media de la tarde (14,30). No regía hora de verano.

Introducimos los datos en el ordenador en los formatos especificados, a medida que lo solicita. Ana resulta SGR Géminis con Ascendente Libra.

```
10 REM *****
20 REM **      URANIA                      **
30 REM **      SOFTWARE ASTROLOGICO        **
40 REM **      APDO. 10112 MADRID          **
50 REM *****
```



```

60 REM
100 DIM N$(12)
110 N$( 1)="ARIES      "
120 N$( 2)="TAURO      "
130 N$( 3)="GEMINIS    "
140 N$( 4)="CANCER     "
150 N$( 5)="LEO        "
160 N$( 6)="VIRGO      "
170 N$( 7)="LIBRA      "
180 N$( 8)="ESCORPIO   "
190 N$( 9)="SAGITARIO  "
200 N$(10)="CAPRICORNIO"
210 N$(11)="ACUARIO    "
220 N$(12)="PISCIS     "

1099 REM ** Adquisicion de los datos de nacimiento
1100 PRINT("NOMBRE      ");:INPUT Q$
1102 PRINT("FECHA (AAAAMMDD)");:INPUT DT
1120 PRINT("LUGAR        ");:INPUT LU$
1130 PRINT("LATITUD (GG.MM)");:INPUT LT
1132 PRINT("NORTE O SUR (N/S)");:INPUT LT$
1134 PRINT("LONGITUD (GGG.MM)");:INPUT LN
1136 PRINT("ESTE U OESTE (E/W)");:INPUT LN$
1140 PRINT("HORA OFICIAL (HH.MM)");:INPUT H
1150 PRINT("DIFERENCIA CON GREENWICH ");:INPUT GW
1160 H=H-GW
1170 PRINT

1999 REM ** Paso de la fecha (aaaaammdd) a dias transcurridos
2000 AA=INT(DT/10000):D4=DT-10000*AA:MM=INT(D4/100):DD=D4-100*MM
2030 TI=365*AA+DD+31*(MM-1)
2040 IF MM<3 THEN TI=TI+INT((AA-1)/4-INT(INT((AA-1)/100+1)*3/4))
2050 IF MM=3 THEN TI=TI-INT(0.4*MM+2.3)+INT(AA/4)-INT((INT(AA/100)+1)*3/4)
2060 TJ=TI-693960:N=TJ-365
2070 N=N+(INT(H)+100*(H-INT(H))/60)/24
2149 REM ** Tiempo siderico
2150 TS=86400*(N-INT(N))+23749.7298+236.5553605*N:TS=TS-86400*INT(TN/86400)
2160 TS=TS/3600
2199 REM ** Posicion del Sol
2200 L=278.965+0.985647342*N:L=L-360*INT(L/360):W=281.235+0.0000469*N:M=L-W
2210 MM=PI*M/180:U9=M+1.91934*SIN(MM)+0.02*SIN(2*MM):L9=U9+W
2220 R9=0.999721/(1+0.01675*COS(PI*U9/180))
2500 SS=INT(L9/30)+1:GS=INT(L9)-30*(SS-1):MS=60*(L9-INT(L9))
2610 PRINT"SIGNO SOLAR" = ";N$(SS);"("";INT(GS);" GRADOS ";INT(MS);" MINUTOS )"
2699 REM ** Calculo del ascendente
2700 L1=INT(LN)+100*(LN-INT(LN))/60:IF LN$="W" THEN L1=-L1
2710 L2=INT(LT)+100*(LT-INT(LT))/60:IF LT$="S" THEN L2=-L2
2720 T=TS+L1*4/60:H=PI*(15*T+3*30)/180:QT=PI*23.45/180
2730 Q=COS(QT):F=TAN(PI*L2/180)*TAN(QT)
2740 S=PI*15*T/180:DV=1
2750 FOR I=1 TO 18:H=ACS(-F*SIN(H))/DV+S:NEXT I
2800 H=H-2*PI*INT(H/(2*PI)):I=ATN(TAN(H)/Q)
2820 IF ABS(H-I)>0.5 THEN I=I+PI:I=I-2*PI*INT(I/(2*PI)):GOTO 2820
2830 H=180*I/PI:IF H<0 THEN H=H+360
2850 SS=INT(H/30)+1:GS=INT(H)-30*(SS-1):MS=60*(H-INT(H))
2860 PRINT"ASCENDENTE" = ";N$(SS);"("";INT(GS);" GRADOS ";INT(MS);" MINUTOS )"
2900 END

```




MICHAEL GEORGE

Como máximo responsable del *marketing* de ordenadores personales Hewlett-Packard en Europa, Michael B. George está en inmejorable posición para apreciar las tendencias del mercado (en realidad, de la pluralidad de mercados) en que se refleja de este lado del Atlántico el fenómeno de la vertiginosa generalización del uso de microordenadores. Aprovechando su breve paso por Madrid, Ordenador Popular dialogó con él sobre ese y otros temas.

P: En cierto modo, el ordenador personal es un producto de la manera americana de trabajar. ¿Acabaremos los europeos trabajando como los americanos? ¿Qué diferencias hay, según usted, entre el mercado americano y los mercados europeos?

R.: Creo que el mercado de los ordenadores personales ya está bien desarrollado en Europa, como una proyección de la introducción de la informática en los métodos laborales y, más recientemente, en otros aspectos de la vida cotidiana. Yo sé que mucha gente piensa que Europa está rezagada respecto de Estados Unidos pero no estoy tan seguro de que sea así.

P.: ¿Por qué no?

R.: En Europa, los usuarios se interesan en los ordenadores personales para el mismo tipo de aplicaciones, hablando en general, que son comunes en Estados Unidos. Pienso en nuestro mercado, quiero decir, en el que persigue Hewlett-Packard, que es fundamentalmente el de las aplicaciones técnicas, y de los profesionales

de los negocios. No veo diferencias importantes a ese nivel. En cambio, el *software* debe ser diferente porque los *standards* profesionales son diferentes, incluso en cada país dentro de Europa.

P.: ¿Cuál es la importancia de los mercados europeos en las cifras de negocios global de Hewlett-Packard?

R.: Aproximadamente la mitad de nuestros negocios son fuera de Estados Unidos, y a Europa le corresponde más o menos el 35 por ciento. Ya puede ver usted por qué es importante no pensar en la imposición de un producto puramente americano.

P.: ¿Cuál es, desde su punto de vista, el ritmo de crecimiento de los mercados del mercado europeo? En este caso, podemos considerarlo como un conjunto...

R.: A cada momento nos llegan todo tipo de estudios de mercado en los que las cifras dependen, por supuesto, de quien haya hecho el análisis, pero sobre todo del sector de mercado de que se esté hablando. Hay mucha confusión cuando se trata de ordenadores personales a secas. Hewlett-Packard está en un segmento definido, el de los ordenadores de sobremesa para profesionales y en las aplicaciones empresariales. Hay otros dos segmentos en los que no entramos, que son el *home computer* y el mercado educacional. Así definido, el mercado crece a tasas de entre el 45 y el 50 por ciento en valor. Como hay una tendencia acentuada al abaratamiento de los equipos, esto quiere decir que el ritmo de crecimiento en volumen es muy superior.

P.: ¿Cree usted que sobrevivirán las pequeñas compañías de microinformática nacidas al calor del *boom* actual?

R.: Me parece que tal vez no haya lugar para todos en este barco, pero tampoco creo en una concentración muy acentuada del mercado. Ocurre que hay pequeñas compañías que deben su éxito a la habilidad con que supieron ofrecer soluciones para determinados mercados particulares. Habrá necesariamente una racionalización de la industria del *personal computer*. Y, desde luego, es más fácil pensar en la desaparición de algunos fabricantes pequeños que imaginar el ocaso de los grandes.

P.: Pero, en términos de marcas, ¿el usuario seguirá teniendo a una multiplicidad como la actual?

R.: Pensándolo bien, creo que hubiera podido responder su pregunta anterior diciendo que algunos fabricantes pequeños, y por tanto algunas marcas, no sobrevivirán más de tres o cinco años en adelante.

P.: ¿Cuál es la estrategia que seguirá en los próximos años Hewlett-Packard?

R.: Creo que no es ningún secreto que vamos a desarrollarnos en dos áreas. Una, la que está representada por empresas de tamaño mediano a grande, en las que los profesionales harán uso creciente de estaciones de trabajo conectadas a la capacidad instalada de computación de la compañía. Otra, las aplicaciones individuales (siempre de carácter profesional) fuera de este tipo de organizaciones, o más propiamente dentro de las pequeñas empresas.

SEIKOSHA

IMPRESORAS

Comunicamos la salida al mercado de la nueva IMPRESORA de COLOR GP-700 Fricción, tracción, todos los COLORES, todas las INTERFACES y sólo por 98.500. En la próxima revista en esta misma página le diremos donde puede adquirirla.

IMPORTADORES EXCLUSIVOS PARA ESPAÑA:

DiRAC S.L.

AV. BLASCO IBAÑEZ, 114-116
TEL. 372 88 89 - VALENCIA-22
TELEX 62220

SORPRENDE A TU FAMILIA METIENDO UN DRAGON EN CASA



 **DRAGON** Data Ltd. **32**

tu ordedragón personal.

- Para aplicaciones comerciales y juegos.
- Posibilidad de tratamiento de textos y base de datos.
- Con 32 K de memoria RAM. (ampliable a 64 K).
- Con gráficos de alta resolución (26 K RAM).
- Con lenguaje Basic ampliado de Microsoft.
- Sin necesidad de magnetofón especial.
- Con color y sonido.
- Con salidas a televisor y monitor.
- Con salida a impresora papel continuo.
- Para trabajar con cassettes y cartuchos ROM.
- Salida para conexión de mandos de juego.
- Hasta 1 Mb de memoria externa con 4 disquetes de 5 1/4" de 250 Kb cada uno (proximamente).

DE VENTA EN DISTRIBUIDORES AUTORIZADOS



DISTRIBUIDOR
EXCLUSIVO:

INVESTRONICA

MADRID

TOMAS BRETON, 60
TELEF. 468 03 00
TELEX 23399 IYCO E

BARCELONA

MUNTANER, 565
TELEF. 212 68 00