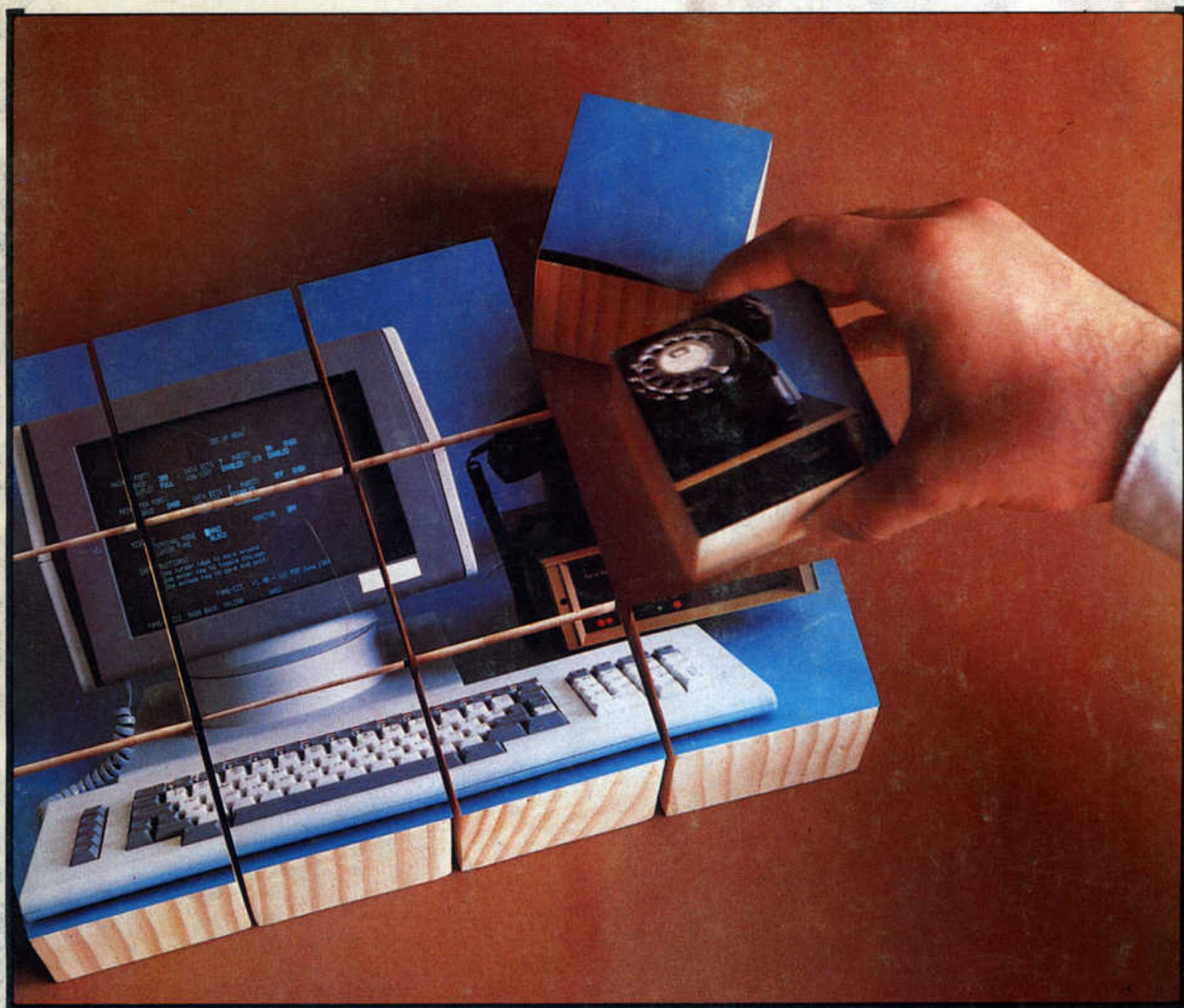


MICROBYTE

TODO COMPUTACION

MARZO 1985
Nº 10 \$ 150



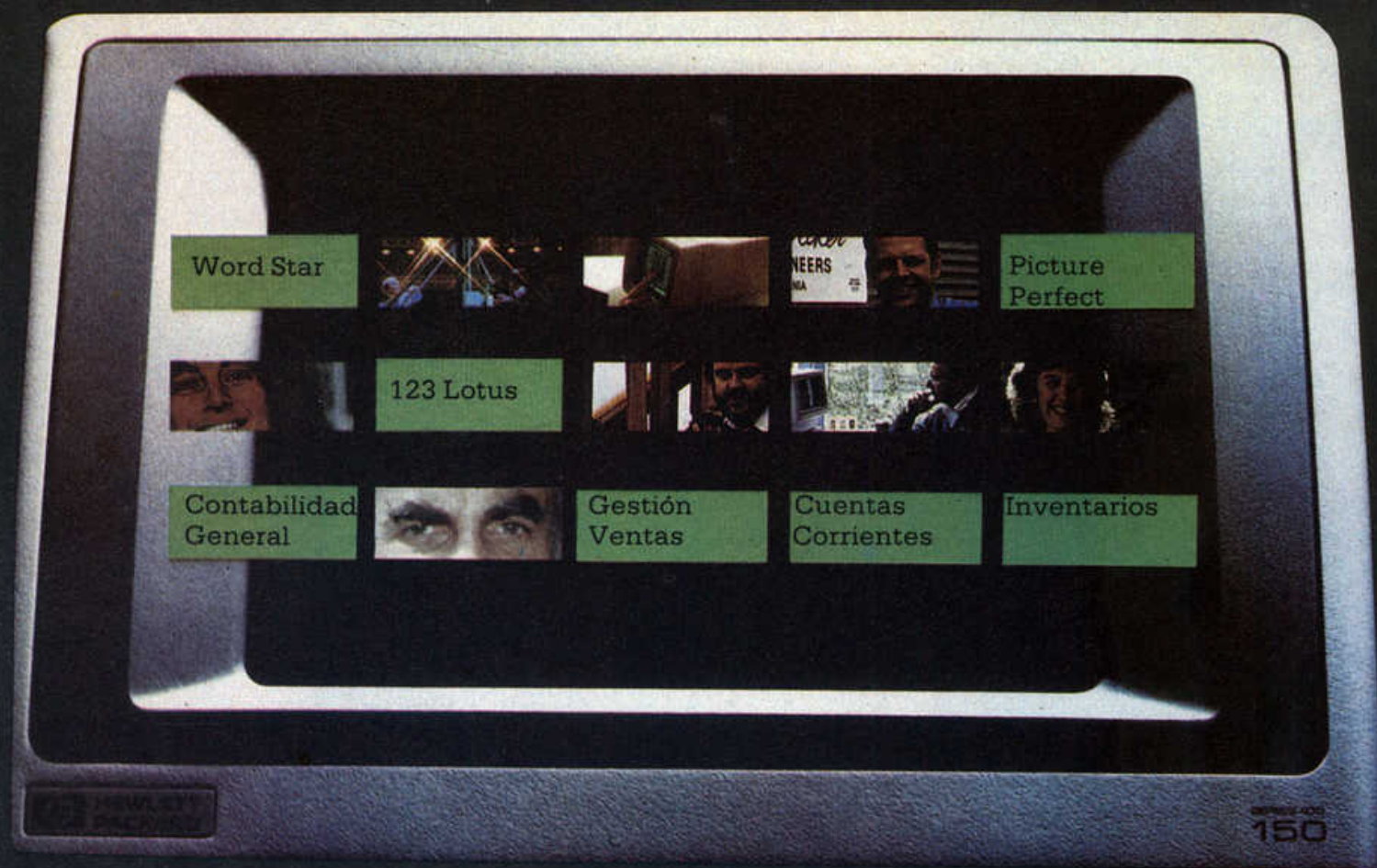
Control de Calidad

CP/M: Primera Parte

Teoría de colas

Construya su propio compilador 6502

HEWLETT-PACKARD



Toque Mágico
Pantalla
Sensible

16 bitz 8 MHZ

Teclado en
castellano

Disco
Winchester
hasta 15 Mb

256 Kb. RAM.
ampliable
a 640 Kb.

Programas
en castellano.

diskette 3 1/2
con 270 Kb.
Formateado

Graficadores
Impresoras.



HEWLETT
PACKARD

OLYMPIA



OLYMPIA (Chile) LTDA. - Avda. Rodrigo de Araya 1045 - ☎ 22 55 044 - 39 22 43 - Santiago.

ARICA 3 28 74	QUIQUE 2 66 56 2 31 84	ANTOFAGASTA 22 23 25	V. DEL MAR 68 07 32	VALPARAISO 25 67 72	SANTIAGO 22 55 044 39 22 43	RANCAGUA 2 42 15	TALCA 3 34 11	CONCEPCION 2 17 02	TEMUCO 3 17 82	OSORNO 49 56	PUNTA ARENAS 2 15 37 2 51 34
------------------	------------------------------	-------------------------	------------------------	------------------------	-----------------------------------	---------------------	------------------	-----------------------	-------------------	-----------------	------------------------------------

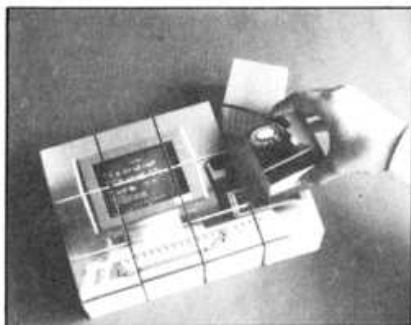


Foto Portada

Ud. lo adivinó. Sin capacidad para comunicarse, un computador es una figura incompleta.

Director Responsable

Jorge Carrera R.

Coordinador General

José Kaffman T.

Director Publicidad y RR.PP.

Ariel Leporatti P.

Publicidad

Juan Pablo Novoa

Ventas

Orlando Zepeda

Redacción Periodística

Myriam Pinto M.

Directora Arte

Paz Barba

Fotografía

González y Geoffroy

Cuerpo Editorial

Jaime Aravena

Jorge Cea

Carlos Contreras

Corresponsales en el exterior

Luis Kaffman T. (Londres)

Alfredo Zarowsky (Paris)

Victor Kahan (Ohio)

Fotocomposición

Laser Ltda.

Representante Legal

Jorge Carrera R.

Dirección: Merced 346 - Of. F

Fono: 393866

Distribución

Antártica S.A.

Impresión

Tamarcos, quien sólo actúa como impresor

Microbyte es una publicación mensual de KVC Asociados.

Ninguna parte de esta revista puede ser reproducida, archivada en sistemas de clasificación o recuperación de datos, transmitida en modo alguno, electrónico o químico, mecánico, óptico, fotográfico o cualquier otro sin el permiso previo de KVC Asociados.

Microbyte no puede asumir ninguna responsabilidad por errores en artículos, programas o avisos publicitarios.

Las opiniones expresadas en estas páginas corresponden a sus autores y no representan necesariamente el pensamiento de sus editores.

Colaboraciones de los lectores son bienvenidas y serán publicadas previa revisión, con un pago de acuerdo a tipo de colaboración y calidad.

Las colaboraciones deben venir tipeadas o impresas a doble espacio y, si es posible, acompañadas de material gráfico. En el caso de listados de programas mayores de 15 líneas, es preferible enviar cassette o disco y una explicación de su contenido.

SUBSCRIPCIONES

Valor subscripciones semestral: (6 Ejs.)

Correo Certif. Stgo. y Prov. \$ 850

Entrega por mano Stgo. \$ 800

Valor subscripciones anual: (12 Ejs.)

Correo Certif. Stgo. y Prov. \$ 1.600

Entrega por mano Stgo. \$ 1.500

Solicite un representante al fono 393866, en Merced N° 346, Of. F. Santiago - Chile.

3 Editorial

24 Control de Calidad Un desarrollo de software nacional

4 Noticias Novedades

30 Construya su propio compilador 6502

13 Curso Sistema Operativo CP/M 1ª parte

38 Educación: Computadores y la enseñanza del Inglés

17 Fichas técnicas N° 2

40 Open File Cartas al lector

20 Teoría de Colas

43 Sección por marcas Atari Casio Sinclair

Rainbow 100

Doble capacidad, doble versatilidad y doble simplicidad para un mismo precio... o casi.

El nuevo computador personal DEC Rainbow 100 de DIGITAL es, sencillamente, sorprendente.

Sorprendente en su diseño ergonómico, largamente estudiado para facilitarle y ordenarle a usted su operación, acuciosamente construido para adaptarse a cualquier lugar de trabajo.

Sorprendente en su ingeniería.

El Rainbow 100 incorpora en forma standard 2 procesadores, de 8 y 16 BITS respectivamente. Esto le permite a usted multiplicar las aplicaciones disponibles, haciendo uso del experimentado y abundante software de 8 BITS para CPM/-80, incorporando todo el emergente software de la nueva y revolucionaria arquitectura de 16 BITS en CPM/-80 o en MS-DOS.

Pero lo que quizás a usted más llame la atención sea su sorprendente versatilidad y facilidad

de uso. El Rainbow 100 le instruye a usted todo lo que necesita saber de su operación, mediante programas de instrucción especialmente incorporados a su sistema, evitándole la lectura de tediosos y voluminosos manuales.

Su impresionante capacidad le permitirá abordar y resolver en él sus problemas de administración de información, de contabilidad, de finanzas, de control de producción, de cuentas corrientes, de planificación, etc.

Finalmente, el versátil Rainbow 100 puede además transformarse en un terminal de los computadores centrales de su empresa, o multiplicar enormemente sus tareas, mediante la incorporación de la más completa gama de periféricos y equipos auxiliares.

Reconocemos que el Rainbow 100 tardó en aparecer en el nuevo y sorprendente mercado de los "personal computers" ...pero pensamos que valió la pena esperar.

Entrega inmediata.


Rainbow 100
Personal Computer



 **SONDA**

Sociedad Nacional de
Procesamiento de Datos Ltda.
CASA MATRIZ: Teatinos 574
Fono: 62277 Santiago - Chile.

digital

Digital
Equipment
Corporation
U.S.A.

Pasada ya la euforia del fin de año y la somnolencia de los primeros meses de uno nuevo, las empresas distribuidoras de computadores personales ya están sacando sus primeros balances.

Si bien aún no se cuenta con cifras detalladas, en general 1984 mostró un fuerte crecimiento en la importación de equipos de computación, alcanzando una suma cercana a los 38 millones de dólares, incluyendo equipos, suministros y periféricos. En 1983 se importaron tan sólo 26 millones de dólares mientras que en 1981, el año record, la cifra alcanzó a los 60 millones.

En el rubro computadores personales profesionales, que corresponde a un 30% de la importación total de computadores, IBM fue la marca de mayor venta seguido por Apple y Hewlett-Packard.

Un indicador interesante, aunque aún por confirmar, apunta a un mayor crecimiento en el área de los computadores personales profesionales en desmedro de los equipos más pequeños, tendencia bastante lógica por lo demás.

En efecto, al igual que en el resto del mundo, también en Chile está pasando de moda la época en que cualquier equipo, por el solo hecho de llamarse computador se convertía en un producto de fácil venta.

Hoy, es necesario que un computador disponga de una gran variedad de software y periféricos que lo hagan apto para servir en distintas tareas. Pasando el entusiasmo inicial, es necesario que los computadores justifiquen la inversión que significan.

Las sucesivas rebajas de precio a que se han visto sometidos prácticamente todos los computadores, han acercado cada vez más en precio a equipos pequeños con micro-computadores más poderosos, haciéndose éstos cada vez más competitivos. Todo esto no hace más que acelerar lo que algunos denominan "el fin de los computadores de juguete".

No es extraño entonces, que los primeros meses de este año se hayan visto inundados de anuncios de nuevos modelos entre las marcas hogareñas más populares. Atari anunció la introducción de cinco nuevos modelos en los próximos meses, uno de los cuales está dirigido directamente a competir con el Macintosh de Apple a una fracción de precio. También Commodore anunció la introducción de dos equipos, el C-128 y el LCD portátil. Sinclair por su parte, desde el año pasado comercializa el QL y ahora además introdujo el Spectrum Plus.

Para el usuario, todas estas novedades son positivas pues la relación costo-capacidad lo favorece cada vez más. Para las empresas distribuidoras sin embargo, esta situación las obligará a especializar aún más su atención al cliente, entregando una información más completa y un mejor soporte. No está mal para ser sólo un comienzo de año.

NOTICIAS NOVEDADES

Olivetti y Ecom firman acuerdo

En un reciente convenio entre Olivetti de Chile y la Empresa Nacional de Computación e Informática (Ecom), ambas firmas determinaron establecer vínculos no exclusivos de soporte recíproco de comercialización. La medida tiene por finalidad ampliar los mercados para equipos y servicios de las partes.

Como primer paso, Olivetti hizo entrega a Ecom de un compu-

tador M40, en el cual Ecom desarrollará software para los usuarios de Olivetti.

El M-40 es un equipo diseñado para ambientes financieros y gestionales, sirviendo como base para la automatización de oficinas. Este convenio, así como otro anterior entre Olivetti y la AT&T a nivel mundial, sin duda reforzará la posición de Olivetti en el mercado chileno.



NCR invita a Congreso en Buenos Aires

NCR está invitando a la banca y al comercio nacional al Cuarto Congreso Latinoamericano de Transferencia Electrónica de Fondos a realizarse en Buenos Aires entre el 22 y 24 de abril próximo.

Entre los temas a tratar destacan los Servicios de Cajeros Automáticos, Aplicación de Precios a Servicios Bancarios y estado actual de la tecnología de telecomunicaciones.

En este congreso en el que se espera un fructífero intercambio de experiencias, participarán además expertos europeos y norteamericanos.

Como invitados chilenos, asistirán representantes de instituciones financieras, grandes tiendas de departamentos, supermercados e instituciones de gobierno.

Computadores franceses

Goupil, el conocido fabricante de micros domésticos en Francia, está ahora representado en Chile. Esperan lanzar en marzo, simultáneamente que en París, un PC-Compatible con características mejoradas denominado Goupil 3.

NEC portátil

NEC anunció un nuevo computador portátil, el PC-8401A basado en un procesador Z-80 con 64K de RAM de los cuales 32 están disponibles para el usuario.

El ROM, este equipo trae 80K, los que incluyen Wordstar y una planilla electrónica. Como monitor, el PC-8401A viene con una pantalla plana (LCD) capaz de desplegar 16 líneas de 80 columnas.

Convenio entre Entel y Ecom.

A comienzos de este año se logró finalmente el acuerdo, dentro de Corfo, que permite el traspaso de la red pública de transmisión de datos de ECOM a ENTEL.

En efecto, este importante proyecto de telecomunicaciones, indudablemente beneficioso para el país, había sufrido serios retrasos debido a los problemas financieros de ECOM que impedía la extensión de la red. Al hacerse cargo del proyecto, ENTEL está dispuesta a darle un impulso definitivo que permita en muy corto plazo poner este servicio al alcance de todas las regiones.

Para alcanzar estos objetivos, ENTEL ha solicitado concesión de servicio intermedio, manteniendo ECOM su actual concesión de servicio público que le permite comercializar el servicio a los usuarios. Naturalmente, cualquier otra empresa puede solicitar concesión tanto para servicios intermedios como públicos, aunque conociendo las características de nuestro mercado, se estima difícil que más de una pueda operar rentablemente, especialmente en los primeros años.

Entre las inversiones a realizar por ENTEL en el transcurso del año se encuentra un centro de gestión de red y varios nodos y concentradores que, agregándose al equipamiento de ECOM, permitirá cubrir todo el territorio nacional con una red autónoma.

Es así como en el transcurso del año, diversas empresas podrán empezar a emplear los servicios de esta red, aprovechando las innegables ventajas, versatilidad y economía que este tipo de herramientas tecnológicas ya han mostrado en otros países.

Sony impone su standard

Cuando en 1981 comenzaron a aparecer los floppy drive compactos, la gran duda era respecto al tamaño final que estos tendrían. En un breve período aparecieron drives de 3,5 pulgadas, de 3 e incluso de 3,25, lo que parecía (y aún parece) un cuento de locos. Naturalmente, un tamaño de diskettes no sirve para ser corrido indistintamente en los tres modelos.

Los principales fabricantes de estos drives (japoneses todos) sabían que sólo sobreviviría aquel que logre convencer a los grandes fabricantes de computadores de incorporar sus drives a sus equipos y, en

esta batalla, a los 3 años recién podemos decir que hay un vencedor absoluto.

En 1982, Hewlett-Packard adoptó el modelo de 3,5 pulgadas de Sony, dando comienzo al irresistible ascenso que tendrían éstos. Luego NEC introdujo un modelo de procesador de texto con drives de 3,5; Apple siguió luego con el Macintosh. A fines de 1984, IBM terminó de dar el último espaldarazo a Sony, eligiendo sus drives para su modelo JX, el PC-Junior japonés.

Sony vende ahora más de 200.000 compact-drives mensuales, mientras que sus antiguos competidores de Matsus-

hita e Hitachi han debido replegar sus banderas, anunciando que debido al bajo volumen de ventas de drives de 3 pulgadas han sacado un nuevo modelo de 3,5. El principal perjudicado, por una errada elección, es Sharp que los utilizaba para su computador personal.

Ahora que el standard ya ha sido fijado, se espera que un gran número de fabricantes de drives comiencen a producir modelos de 3,5 pulgadas. Entre estos ya se encuentra Brother, el afamado fabricante de máquinas de coser, quien se está lanzando a este mercado de alta tecnología con sus propios drives.

Compatibilizando procesadores de texto

De acuerdo a estudios realizados por la marina norteamericana, un documento puede llegar a ser retipeado hasta 20 veces a lo largo de su vida debido a la falta de compatibilidad entre los procesadores de texto de distintas empresas, e incluso dentro de una misma empresa.

Como un medio de evitar este traspaso manual de textos en la época de escritura electrónica, Keyword Office Technologies creó un conversor de diskettes a diskettes que funciona como periférico en un IBM PC.

El Keyword 7000 consiste en una tarjeta que se inserta en uno de los slots del PC y una unidad que contiene dos drives de 5,25 y dos de 8 pulgadas. La tarjeta contiene un integrado capaz de simular el controlador de distintos tipos de diskettes.

De este modo, basta con insertar un diskette fuente en un drive y un diskette vacío en otro. El software se basa en menús múltiples en los que se ingresa el tipo de procesador de texto que se quiere homologar en cada drive. De acuerdo a sus fabricantes, toma aproximadamente 13 segundos convertir y traspasar una página de texto de un diskette al otro.

Hasta el momento, ya hay software disponible para traspasar textos entre procesadores IBM Displaywriter, Xerox 320, Lanier, Philips Micom (serie 200 y 300), Decmate II, Multimate en modo PC-DOS, Wordstar en modo PC-DOS o CP/M y Wang corriendo WPS, OIS y VS.

El precio del Keyword 7000, no es sin embargo muy económico. El hardware con software apropiado para compatibilizar una pareja de procesadores alcanza a los US\$ 9.995 y el costo del software adicional para otras parejas de procesadores asciende a US\$ 495.

Chaos Computer Club

Después de la moda del rock, Estados Unidos había dejado de producir modas que impactaran a la juventud. Hoy nuevamente están ocupando un lugar destacado con la exportación de "hackers", jóvenes especializados en romper todas las claves de seguridad de los sistemas de transferencia electrónica de información.

Inspirados en algunas películas, todas las semanas están llegando noticias de los modernos vándalos que en todo el mundo se dedican, desde sus casas o clubes de aficionados, a poner los pelos de punta a los encargados de seguridad de los sistemas de teletext y viewdata.

En Inglaterra, un hacker, educado naturalmente en Estados Unidos, logró acceso a la clave del príncipe Felipe al sistema Prestel, accediendo así varias páginas de información supuestamente confidencial.

En Alemania, ceñudos funcionarios de la Bundespost, empresa de correo y telecomunicaciones encargada de la red de videotex nacional (BTX), aún tratan de averiguar cómo hizo un club de aficionados, el Chaos Computer Club para averiguar la clave del Banco Hamburger Sparkasse, para luego utilizarla para solicitar servicios del propio club.

En este caso, publicitado por el propio grupo para demostrar sus habilidades y la escasa seguridad de los sistemas, al apropiarse de la clave del banco, el club fue capaz de hacer que éste requiriera 13.000 veces durante el fin de semana unas páginas de información del propio club, lo que le costaría al banco la módica suma de US\$ 50.000, aproximadamente.

Los funcionarios del banco no hicieron comentarios aparte de afirmar su convicción de que no recibirían ninguna factura del club.

Computadores del año

Al igual que todos los años, al finalizar 1984, seis revistas europeas de computación más Popular Computing de Estados Unidos, con la organización de la revista Chip, de Alemania, se reunieron para elegir a los computadores más exitosos del año.

Como computador hogareño, resultó nuevamente elegido el Commodore 64, seguido por el QL de Sinclair y el Alphatronic PC.

Computador personal del año salió elegido el Macintosh de Apple, seguido por el HP-150 y el IBM PC/XT.

En la categoría portátiles, el ganador por estrecha mayoría fue el Sharp PC-5000 con una pequeña ventaja sobre el Epson PX-8 y el Olivetti M-10.

Entre los transportables, una categoría bastante difícil de definir, el vencedor fue el Compaq Plus. Su más cercano rival fue el Apple IIc.

Cabe destacar que estas nominaciones se refieren más que a una excelencia tecnológica, al hecho de que estos equipos ya han salido al mercado, cuentan con una gran variedad de periféricos y sobre todo de una amplia gama de software.

Otros equipos pueden ser mejores tecnológicamente, pero por razones de mercado o por ser muy recientes no cuentan aun con mayor respaldo de software y periféricos.

Lo novedoso esta vez es que también se eligió al mejor software del año, dividido en varias categorías.

Como Juego del Año salió nominado el Flight Simulator de Sublogic/Microsoft. En el rubro comercial, el ganador por lejos fue el Lotus 1-2-3.

Autocad, un programa para dibujo y diseño fue elegido Software Técnico del año mientras que Logo de Digital Research recibió el premio al mejor software de sistema del año. Segundo en esta categoría salió favorecido el CP/M concurrente también de Digital Research.

IBM enseña computación a hijos de su personal

Cincuenta jóvenes, de entre 16 y 20 años, hicieron un alto en sus vacaciones para asistir en dos grupos, durante cinco medios días, al segundo curso de computación que para hijos de su personal organizó, a través de su Departamento de Educación, la IBM de Chile.

Los óptimos resultados de la experiencia los revelan los juicios de los instructores.

"Los alumnos demostraron una gran inquietud por aprender y muchos volvían en las tardes junto a los terminales para practicar por su cuenta", señala Valentina Veloso.

Por su parte Juan Santibáñez acotó: "a diferencia de los adultos, los jóvenes y los niños se enfrentan sin miedo a la pantalla y preguntan sin complejos cuando una duda los acomete".

Los instructores explicaron también que los cursos no intentaron influir en los alumnos para que sigan una carrera de computación. "Se trata de iniciarlos en una ciencia que constituye, cada día más, una herramienta indispensable, cualquiera que sea la carrera universitaria y, por ende, la profesión que elijan".

El primer ensayo de este tipo se puso en práctica en el verano del 84, cuando 125 niños de 10 a 15 años, hijos también de funcionarios de la IBM, recibieron similar instrucción. El éxito de ambos cursos entusiasmó a organizadores y padres que ahora apuntan hacia otro componente familiar: las cónyuges del personal.



Hijos del personal de la IBM se familiarizan con el teclado de los terminales.

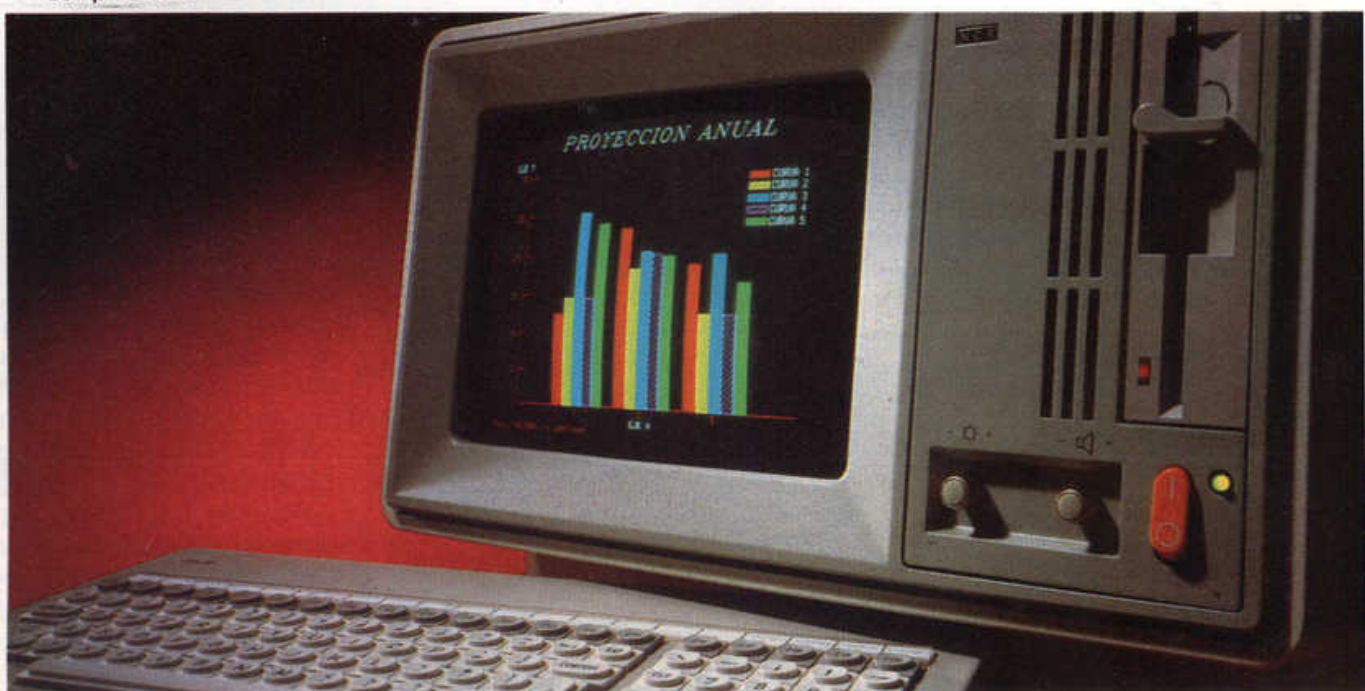
Asociación LOGO

Con su característico entusiasmo, la Asociación LOGO está finiquitando los preparativos para un ampliado a realizarse los días 29 y 30 de marzo.

Los objetivos de este ampliado son intercambiar experiencias mediante la presentación de trabajos y programas en LOGO, integrar nuevos miembros a la asociación y elegir una nueva directiva.

El 5 de marzo es la fecha tope de recepción de los trabajos a ser expuestos, los que hay que hacer llegar a la Asociación a su local en calle Victoria Subercaseaux 201 - Santiago, a nombre de Gustavo Jiménez.

Este encuentro servirá además para preparar una eventual participación en la Conferencia Internacional de LOGO que organiza el M.I.T. en Cambridge entre el 22 y el 25 de julio próximos.

NCRInnovadora tecnología
computacional

ERA DE ESPERAR QUE EL COMPUTADOR PERSONAL NCR FUERA ASI.

Con todas las ventajas de la tecnología y experiencia de NCR.

- Procesador de datos de 8 bits o de 8/16 bits
- Procesador destinado a gráficos
- Pantalla con alto poder de resolución
- Teclado fácil de utilizar (numérico y alfanumérico)
- Memoria principal de hasta 512 KB.

Fácil de manejar y con un alto rendimiento en el procesamiento de datos. Era de esperar de NCR, una empresa que cumple 100 años innovando.

Lo esperamos con nuestra red de distribuidores.

NCRInnovadora tecnología
computacional

Mac-Iver 370 - Fono 380013 - Santiago

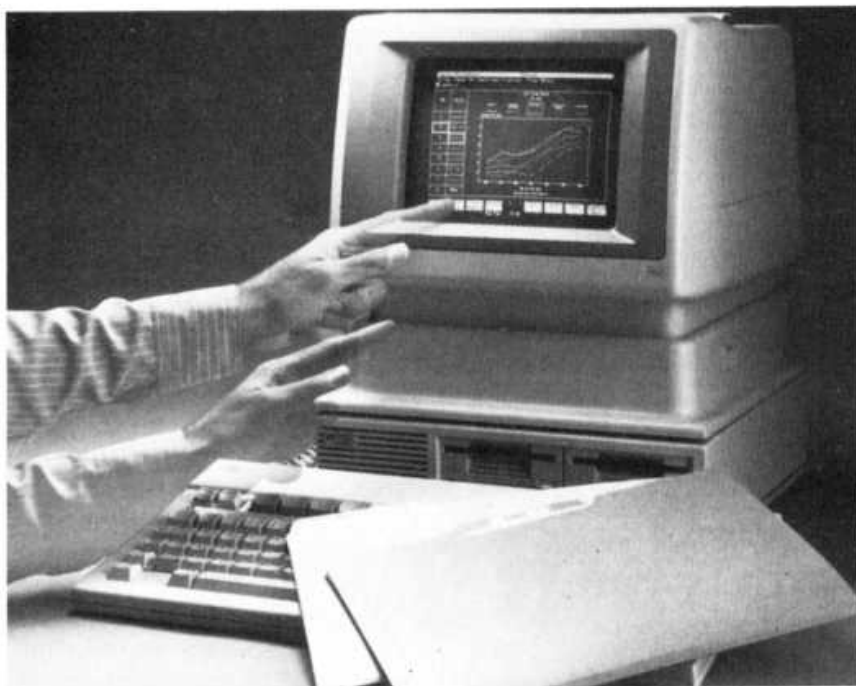
Nuevo modelo HP-150-B

Desde comienzos de este año fue puesto en venta la versión B del popular computador HP-150, el del "toque mágico".

Entre las mejoras hechas al equipo, cabe destacar una mayor capacidad de almacenamiento en diskettes, que se eleva de 540 Kb a 1.4 Mb. En disco fijo, su capacidad llega a 15 Mega.

Aumenta también la velocidad de transferencia, posibilidad de conectar unidades de soporte en cartridge de 67 Mb, incluye disco RAM y permite formar redes de computadores.

Los propietarios de un HP 150-A, deben saber que su equipo puede convertirse en un modelo B, cambiando las piezas correspondientes a un costo de 120 a 500 dólares de acuerdo a la capacidad de almacenamiento deseado.



HP-150

WordStar 2000

MicroPro Int. Corp. es una empresa que se elevó a la fama y la fortuna con un sólo producto que se convirtió en best-seller y sinónimo de procesador de palabras en microcomputadores, el WordStar.

Sin embargo, nuevos programas, más sencillos de utilizar, y el auge de la piratería de software han dejado a esta empresa al borde del colapso económico.

Ahora, MicroPro, quiere volver a su perdido sitio con WordStar 2000, un procesador de texto que incluye por US\$ 425 un corrector de ortografía tipo SpellStar y la capacidad para dividir la pantalla en hasta tres ventanas para diferentes documentos y traspasar información de una a otra.

Gráficos para Wordstar

Doodle es un programa gráfico concebido para ser usado por su cuenta o en asociación con el popular procesador de textos Wordstar. De este modo se pueden producir textos ilustrados sobre la base de computadoras como Apricot y el IBM-PC.

Los gráficos se crean sobre la pantalla usando los controles del cursor e instrucciones como las del lenguaje Logo (también se puede usar una "pizarra" electrónica), y para mezclar los gráficos creados con el texto se usan instrucciones como las de Wordstar. Doodle genera entonces el texto ilustrado usando cualquier impresora de matriz.

A.S.C instala equipo en Valparaíso

La firma A.S.C. ganó la licitación a que llamó la Universidad Federico Santa María de Valparaíso, consistente en un equipo de computación para funciones administrativas y manejo de biblioteca.

El equipo, que será instalado en el presente mes de marzo, es un modelo H.P. 3.000 configurado con dos MB en memoria principal, un disco de 132 mega, 16 terminales, dos impresoras de 200 caracteres por segundo, una impresora de 600 líneas por minuto y una cinta magnética de 1.600 CPI.

Ricardo Borzutzky, gerente general de Avanzados Sistemas de Conocimientos Ltda. (ASC), comunicó que el moderno sistema lo instalarán ingenieros enviados ex profeso por el Gobierno canadiense. Este servicio se hace sin cargo alguno para la universidad porteña, de acuerdo a un convenio suscrito entre Hewlett-Packard y el Gobierno de Ottawa.

El equipo de computación para la biblioteca utiliza el programa Minisis, usado, entre otras, por las bibliotecas de las Naciones Unidas y de la Organización Internacional del Trabajo.

Texas Pro-Lite

Texas anunció la próxima introducción de un nuevo computador portátil para competir con similares de Data General y Hewlett Packard.

El nuevo computador, conocido como Pro-Lite pesará alrededor de 5 kilogramos, incluyendo una pantalla de doce pulgadas capaz de desplegar 25 líneas de 80 caracteres y un disk drive de 3,5 pulgadas.

Este equipo, con 256 K de memoria principal, tendrá un precio de venta aproximado de US\$ 3.000. En principio, Texas espera producir una cantidad limitada de estos equipos este año debido a razones de capacidad de producción y además porque estima que el mercado de los portátiles aún no madura.

Banco Integrado de Proyectos



Asistentes al seminario de monitores del BIP. Cincuenta personas de Santiago y provincias, fueron capacitadas en enero por ODEPLAN, para esa función.

—Nuevo sistema computacional es presentado como único en el mundo

—Está destinado a poner fin a los derroches de la inversión pública

—Su base de datos es la mayor del país

Con el propósito de mejorar la asignación de los recursos financieros destinados a la inversión pública comenzó, en febrero recién pasado, la operación definitiva del Banco Integrado de Proyectos del Sector Público (BIP). En su gestación, que duró seis laboriosos años, participaron, mancomunadamente, la Oficina Nacional de Planificación (ODEPLAN) y el Programa de las Naciones Unidas para el Desarrollo (PNUD).

Para el ingeniero Sandro Raggi, uno de los promotores de la iniciativa, el banco es único en el mundo. En muchos países existe interés por conocer el sistema de informática computacional que representa. Sus bondades, al decir de Raggi, han despertado el deseo del Banco Interamericano de Desarrollo y del Banco Mundial, de invertir en proyectos que antes no financiaban por dificultades de control y seguimiento.

El derroche que se busca evi-

tar tiene numerosos ejemplos, productos de la falta de coordinación adecuada entre los centenares de instituciones en que se encuentra dividida la Administración Centralizada del Estado. Esta necesaria parcelación la obliga, con frecuencia, a duplicar, sin motivo, los gastos de inversión. Es el caso de la acera recién pavimentada, que se rompe para colocar tuberías de alcantarillado de mayor diámetro. Y luego, cuando ya se había rellenado la excavación y recompuesto el pavimento, otra cuadrilla de obreros, de otro servicio público, se presenta en el mismo sitio para romper de nuevo. Esta vez con el propósito de cambiar los conductos de agua potable.

El BIP impedirá estas situaciones al contener toda la información técnica y financiera relativa a los proyectos de inversión pública. Para ello cuenta con la memoria electrónica más poderosa del país, que consiste en una base de datos única ubicada en el Registro Civil.

Se trata de un equipo Digital 1091 con una capacidad de memoria de 1,2 mega word (word = 36 bits). Por otra parte su capacidad de almacenamiento en disco es de 1,471 mega word.

La base de datos se conecta en teleproceso y en línea a terminales que se hallan en cada

capital regional. A nivel central hace lo mismo con terminales instalados en Ministerios y en la Contraloría General de la República. Para las comunicaciones entre la base y la red se utilizan tres "front-end processors".

En la base de datos y los terminales los usuarios pueden ingresar u obtener la información que requieran para cumplir sus funciones específicas. Al ser en línea el sistema permite conocer en forma rápida la situación al momento de invertir.

La información contenida sólo representa la inversión financiera y nada relativo a la contabilidad gubernamental de responsabilidad de la Contraloría.

El potencial del sistema y los beneficios que aporta se traducen en los distintos usos que se le puede dar. Entre los más importantes se cuentan:

—Planificar la inversión, considerando una eficiente planificación de la preinversión.

—Entregar información puntual de un proyecto o agregada de muchos, en forma rápida y oportuna.

—Analizar demandas de financiamiento para inversión o preinversión a futuro o conocer la generación de empleo en paquetes de proyectos alternativos de invertir.

—Conocer la inversión agregada por sectores económicos y sociales, por regiones o por comunas, de manera rápida y actual.

Participaran en el sistema, en calidad de usuarios, 800 instituciones entre Ministerios, Intendencias, Gobernaciones y Municipalidades. Esto es, todas las instituciones de la Administración Centralizada del Estado, exceptuando en una primera etapa a las empresas estatales y a las universidades.

Paralelo al desarrollo del BIP se cumplió una intensa labor de normatización del proceso de inversión, de estandarización de la información a ingresar al sistema y de capacitación de funcionarios públicos en su manejo.

Arte y Computación

El avance de la tecnología también está poniendo en duros aprietos a críticos y filósofos del arte. En efecto, cada nuevo adelanto en las formas de producir arte han traído consigo áridas discusiones respecto a que es y que no es arte.

La fotografía es un buen ejemplo de lo anterior. Hubo de pasar un respetable tiempo hasta que se logró un acuerdo mínimo de considerar a algún tipo de fotografía como artística, frente al absoluto rechazo de pintores y escultores.

Hoy, el mismo problema se plantea con la aplicación de las computadoras al arte. Un computador, con un software adecuado puede componer música

grata al oído no muy diferente a composiciones humanas. Lo mismo ocurre con poesía, pintura e incluso escultura. Quien sabe si en pocos años más, al visitar el Louvre no nos encontremos junto a la Gioconda con una máquina de Videogame con imágenes de Pacman o Donkey Kong.

En la fotografía, cortesía de los Servicios de Informaciones de la Embajada de Estados Unidos, se aprecia al artista Harold Cohen, de la Universidad de California en San Diego dibujando en grandes telas con ayuda de un computador, las que serán expuestas en museos de arte.



Breves

** Si bien el nuevo PC-AT de IBM es aun un producto nuevo, no han tardado en aparecer fabricantes de tarjetas para insertar en este diversas capacidades.

Entre los primeros, esta AST quien produce una tarjeta que permite mediante el uso de integrados de 64kbit o 256Kbit agregar hasta 3 Megabytes de RAM, conteniendo además dos puertas seriales y una paralela para impresora. Por US\$ 495, parece una buena compra.

** Si los japoneses hacen ejercicio físico durante su jornada laboral y llevan conquistado medio mundo con sus productos, los norteamericanos piensan emularlos con Breakdance.

Esa es al menos la intención de la conocida empresa de software Creative Software de California que acaba de liberar un programa que por US\$ 25 le asegura convertirse en la admiración de sus amigos como campeón de Breakdance. Por el momento, el programa solo corre en Commodore, pero luego serán liberadas versiones para correr en sistemas operativos multi-tarea.

Todos contra uno...

En absoluta reserva se reunieron a fines del año pasado en París los principales fabricantes de computadores norteamericanos y europeos. Entre los asistentes se contaban Sperry, Burroughs, NCR, DEC, Hewlett-Packard, Bull, Thomson, ICL, Nixdorf, Olivetti, Philips, Siemens, etc.

Paradojalmente, el único que no asistió y sobre el que más se habló, fue IBM. La razón, simple. La reunión se centró en el estudio de desarrollar y concretizar nuevos standards que faciliten la interconexión de sus equipos que les permita enfrentar mejor el creciente dominio de IBM en el mercado de los computadores.

Este es en realidad el primer encuentro luego de la reunión efectuada a comienzos de 1984 entre los principales doce fabricantes europeos en que acordaron la adopción de un standard, llamado Open System Interconnection (OSI).

En todo caso, si bien luego de la reunión todos los asistentes expresaron su unánime acuerdo en proseguir con la adopción del nuevo standard, aún no está clara la factibilidad de éste ni los plazos en que se podría alcanzar.

Apple hace morder el polvo a sus imitadores asiáticos

Dicen que el plagio es el más sincero de los elogios, pero Apple Computers no se ha dejado enternecer, y su ejército de abogados e investigadores encargados de poner fin a la competencia de empresas de Taiwán están propinando golpes demoledores a los productores de "Apple-compatibles" que violan sus patentes.

El Unitrón 2200 fue discretamente retirado del mercado británico (con pérdidas considerables para sus distribuidores, quienes habían gastado alrededor de US\$ 120.000 publicitando el hecho de que esta maquina costaba menos de la mitad del Apple equivalente). Y en Australia el Wombat (un Apple II apenas disfrazado) debió retirarse con la cola entre las piernas luego que la justicia australiana determinara en septiembre pasado que se trataba efectivamente de una copia (los plagiarios taiwaneses dejaron incluso las iniciales de los programadores de Apple en sus equivalentes del ROM Appleso (intérprete de Basic) y el Autostart (que controla al monitor).



Multitech

MAS COMPUTADOR POR MENOS PRECIO

1 **MPF III** Compatible con APPLE II

Memoria RAM : 64 KB
Sistemas Operativos : DOS y CP/M
Almacenamiento : 1/2 Disketeras 140 KB

2 **Mic 504**

Memoria RAM : 64 KB
Sistemas Operativos : CP/M
Almacenamiento : 2 Disketeras 1000 KB c/u

3 **MPF- PC** Compatible con IBM PC

Memoria RAM : 256 KB - 640 KB
Sistemas Operativos : DOS y CP/M-86 Concurrente
Almacenamiento : 2 Disketeras 360 KB c/u
MPF-PC : 1 Disketera y Disco 10 MB
MPF-PC XT : 1 Disketera y Disco 10 MB



CIENTEC

INSTRUMENTOS CIENTIFICOS LTDA.
Departamento Computación
Antonio Varas 754
Fonos 225 7350 - 74 7028
Santiago

REPRESENTANTE PARA CHILE

DISTRIBUIDORES RESPALDADOS POR CIENTEC:

SANTIAGO	: ADCOM	F: 2237426
SANTIAGO	: COMPUTER MARKET	F: 2243474
SANTIAGO	: COMPUTER MARKET	F: 399828
SANTIAGO	: E. CHILENA COMP.	F: 392254
SANTIAGO	: ING. SER. ELECT.	F: 776991
RANCAGUA	: ASCOMING LTDA.	F: 21869
ANTOFAGASTA	: INFOCOM LTDA.	F: 224762
LA SERENA	: E. CHILENA COMP.	F: 213222
VIÑA DEL MAR	: VECOM LTDA.	F: 882490
TALCA	: ABECAR LTDA.	F: 35837
CONCEPCION	: E. CHILENA COMP.	Caupol. 567
TEMUCO	: STG LTDA.	A. Prat 837
OSORNO	: STG LTDA.	F: 5254

Inglaterra y Francia aceleran alfabetización computacional.

En Inglaterra, ya existen más de 100.000 computadores en las escuelas primarias y secundarias, un promedio de diez por escuela secundaria, alcanzando el primer lugar en densidad de computadores por estudiante en el mundo.

En 1981, año en que el gobierno comenzó con su campaña "Micros in Schools", menos de un 50% de las escuelas secundarias británicas contaban con al menos un computador. En febrero de este año, más de un 97% del total de escuelas ya contaban con un computador. De acuerdo al sistema para equipar a sus escuelas, el gobierno central corre con la mitad de los gastos, mientras que las autoridades educacionales locales deben conseguir la parte restante del financiamiento. Este esquema ha sido más exitoso en algunas regiones que en otras, reflejando directamente el mayor o menor interés de las autoridades locales por hacer acceder a sus alumnos al alfabetismo computacional.

En Francia, de acuerdo a los proyectos gubernamentales, recién en 1988 se alcanzará la meta de 100.000 computadores instalados en las escuelas. Sin embargo, los planes franceses son aún más ambiciosos. De acuerdo a un programa elaborado por una comisión de siete ministerios, los requerimientos de modernización tecnológica franceses no serían satisfechos mediante la sola introducción de computadores a las escuelas, sino que es necesario que también los adultos puedan tener acceso masivamente a ella.

De ser aprobado este proyecto, en alrededor de 36.500 pueblos, aldeas y ciudades se instalarían durante 1985 talleres de capacitación, abiertos a todos los interesados de cualquier edad y condición. Este proyecto, en que participan como asesores, expertos de la Universidad de Carnegie-Mellon de Estados Unidos, no tiene parangón con ninguna experiencia conocida en ningún país.

MS/DOS + GEM = MACINTOSH

Digital Research liberará próximamente un sistema operativo que opera como extensión del MS-DOS que le permite a cualquier equipo tipo PC de 16 bits emular las capacidades de un Macintosh.

Con GEM (Graphics Environment Manager) se pueden utilizar iconos, menús desplegables, ventanas de distintos formatos e incluso dibujar gráficos en colores.

Windows

Microsoft anunció que el largamente esperado Windows, un sistema operativo multitarea, será liberado finalmente a mediados de este año. Este producto, del que comenzó a hablarse en 1983, entrará a competir con retraso con productos similares de Digital Research (CP/M 86 Concurrente) e IBM (Topview), los que permiten correr hasta cuatro aplicaciones simultáneas, dividiendo la pantalla en ventanas.

Europa trata de recuperar el tiempo perdido

La empresa alemana Siemens y la holandesa Philips acaban de unir sus fuerzas para tratar de alcanzar a las empresas japonesas y estadounidenses en cuanto a diseño y fabricación de microchips de 1 y 4 megabits (millones de caracteres binarios). Estos serán los semiconductores básicos de las computadoras de la próxima década (los de mayor capacidad actualmente son capaces de almacenar un máximo de 256.000 caracteres, y recién ahora comienzan a ser producidos a escala comercial).

Estos dos gigantes de la electrónica europea han venido cooperando en proyectos conjuntos desde 1982, pero nunca antes en una operación de esta envergadura. Este programa de investigación y desarrollo costará alrededor de US\$ 1.000 m en los próximos 7 años, y la mitad

de esta suma será provista por los gobiernos de Alemania y Holanda.

Para poder llegar a producir este tipo de chips hay que miniaturizar aún más los ya microscópicos circuitos integrados, reduciendo el grosor de cada hilo conductor a alrededor de 0,5 micrones (milésimas de milímetro). Esto ya ha sido logrado en el plano experimental, pero el desarrollo de una capacidad de producción masiva requiere un salto tecnológico.

Philips está trabajando en el desarrollo de nuevos equipos de producción basados en la litografía de rayos X.

Los chips de memoria RAM son el más ubicuo de los componentes microelectrónicos, con un mercado mundial estimado en US\$ 5.000 m por año (de un mercado global de US\$ 22.000 m para todo tipo de

semiconductores). De más está decir que la mayor parte de él es hoy controlado por empresas japonesas.

Se calcula que el mercado llegará a US\$ 10.000 m hacia 1990, y Philips-Siemens se han planteado como objetivo conquistar entre 5-10% de él para ese entonces. Eso no parece una ambición desmedida. El problema será lograrlo en base a competitividad genuina, y no a través de subsidios y medidas proteccionistas. El mercado de estos chips "standard" es ferozmente competitivo, y los precios tienden a bajar con pasmosa rapidez.

Curso: Uso del Sistema Operativo CP/M

Jaime Aravena L.

Temario:

1. 1.1. Qué es un S.O.
- 1.2. Características de CP/M.
2. Operación básica.
3. Nombres de Archivos.
4. Comandos de CP/M.
Básicos
Transitorios
5. Detalle de algunos comandos transitorios:
STAT, ASM y DDT.
Estructura física de los archivos CP/M.
6. Estructura del sistema operativo CP/M.
7. Fabricación de programas usando CP/M.

1.1. ¿Qué es un sistema operativo?

Un Sistema Operativo es un tipo de programa que sirve como un enlace entre el computador (el hardware, la máquina misma) y los programas del usuario. Es un programa que organiza el manejo de las unidades de diskette, de la impresora y la pantalla, liberando al operador y al programador de los problemas más comunes, simplificando sus acciones.

Este tipo de programa permite configurar el equipo, es decir, definir la cantidad de periféricos y modo de tratar a cada uno de ellos. Además, incluye Programas utilitarios necesarios para formatear diskettes, efectuar respaldos de diskettes, transferir archivos, borrar archivos, verificar espacio libre, etc.

El concepto de Sistema Operativo se desarrolló al final de la década del 50, al fin de la Segunda Generación, cuando los computadores se construían con transistores. La adopción de estos programas para controlar la máquina en la Tercera Generación —computadores hechos con circuitos integrados—, facilitó mucho el uso de los computadores. Este concepto es, en parte, responsable de la popularización de la informática desde entonces. Esta idea de utilizar un programa intermediario entre el programa del usuario y la máquina es un avance tan grande en la computación como lo fue la adopción del transistor para la construcción del Hardware.

El Sistema Operativo es un programa complejo y debe estar adaptado al hardware específico de la máquina que controla.

La idea de utilizar un programa como el "Siste-

ma Operativo" es otra de las geniales ideas que rodean la evolución de la computación. En efecto, la computación se basa en la dualidad Hardware-Software, que permite construir máquinas estandarizadas, sin un objetivo específico predefinido y luego asignarles una función determinada mediante la adición de una información, los programas, para lograr una finalidad deseada.

Todas las máquinas, como ayudas que el Hombre crea, se construyen con un objetivo. Si el objetivo es complejo, la máquina realiza una secuencia de operaciones que le permiten obtener un resultado predefinido a partir de un material base.

En un telar, por ejemplo, el objetivo a cumplir es tejer y este objetivo se logra mediante operaciones que están determinadas por los engranajes, palancas y poleas que controlan los movimientos necesarios para realizar el tejido. Son estos dispositivos los que conforman la máquina misma, de modo que no puede alterarse el objetivo (tejer) sin modificar fundamentalmente la máquina. Puede decirse que es la máquina la que determina, en sus elementos constructivos, el objetivo a realizar.

Todo lo dicho es obvio, pero pone énfasis en un aspecto esencial de los computadores.

Si se denomina PROGRAMA a la secuencia de operaciones que permite lograr un objetivo, se visualiza que en el caso del telar, el programa está definido por los materiales que conforman la máquina misma. Es decir, el programa de la máquina está determinado por la forma en que se interconectan las partes de la misma máquina.

El conjunto de materiales que componen una máquina se conoce, en la jerga computacional, con el nombre de HARDWARE; palabra inglesa que significa ferretería o quincalla.

De este modo se dice que, en el ejemplo del telar, el PROGRAMA de la máquina está definido por HARDWARE. Se comprende también que, en este caso, ésta determinación es rígida, en el sentido que cualquier modificación del objetivo es difícil ya que significa rehacer la máquina misma.

El ejemplo indicado es válido también para máquinas de cálculo: una máquina sumadora sólo sirve para eso, y esto está fijado así desde el momento de fabricación.

La característica más sorprendente de los computadores radica en que lo anterior no es válido

para ellos; existe INDEPENDENCIA entre el programa y la máquina.

En efecto, el objetivo a realizar por una computadora no está determinado por la máquina (hardware) sino que por un programa que se agrega posteriormente a la fabricación y es independiente de ella.

De hecho, al fabricársele, no se le construye para obtener un fin predeterminado, sino que éste debe ser definido por el usuario. De ésta forma un mismo computador puede calcular la trayectoria de un cohete o una planilla de sueldos, según el programa que se le agregue posteriormente a su fabricación, en el momento de uso. De este modo, puede decirse que una máquina computadora por sí sola no sirve para nada específico sin un programa que le defina sus funciones.

Podemos decir entonces que el PROGRAMA es pues independiente del HARDWARE, y se puede modificar fácilmente el objetivo de la máquina, cambiando el programa.

Por ésto, en la terminología usual se denomina SOFTWARE a los programas de un computador.

Esta palabra se obtiene modificando la palabra inglesa hard-ware reemplazando la partícula HARD, que significa duro o rígido, por la partícula SOFT, que significa blando o flexible.

De esta forma se obtiene un neologismo inglés que es de amplio uso internacional. Como tal no tiene una traducción directa a nuestro idioma y su significado, como ya se indicó, es el acervo de programas que se pueden introducir en un computador para dotarlo de un objetivo específico.

La característica principal de los computadores radica entonces en la propiedad que tienen de poder cambiar de objetivo fácilmente, ya que separan la máquina de los programas, o en otras palabras, el HARDWARE es independiente del SOFTWARE.

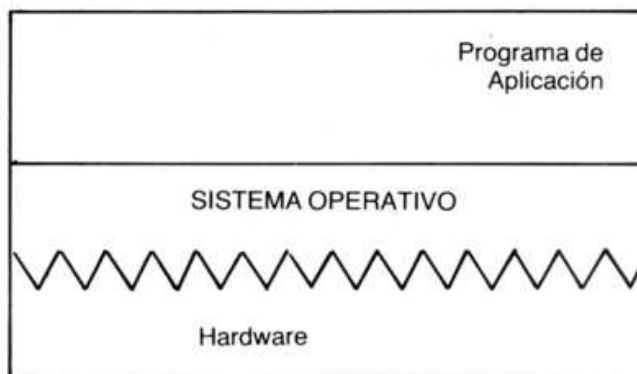
Debe notarse que esta independencia no implica autonomía, ya que para que un sistema computacional trabaje es indispensable la PERFECTA COORDINACIÓN de hardware y software.

La fabricación de este software, necesario para que el sistema informático funcione, es un problema que cada usuario debe resolver. Más aún, es justamente éste problema el más crítico en el desarrollo de la informática.

Una imagen gráfica puede aclarar la idea y plantear la función realizada por el sistema operativo: Si consideramos un sistema informático como un todo, distinguimos en él dos partes independientes, pero que deben coordinarse perfectamente, como dos piezas de un rompecabezas, con un calce difícil de lograr. La dificultad de este ajuste, antes del invento de los S.O., significó que los programas sólo pudieran ser realizados por científicos y matemáticos con preparación especial. Tal vez desde ésa época persiste la imagen del programador como un ser misterioso y superdotado (imagen que muchos aún cultivan con dedicación). En aquel entonces, la coordinación hardware-software requería un profundo conocimiento de matemáticas y electrónica que permitiera, en LENGUAJE BINARIO, expresar los obje-

tivos que componían el programa.

La idea para simplificar esto se basó en la agrupación de todas las subrutinas que se relacionan con la máquina en un sólo conjunto coordinado y coherente, de modo que el programa de aplicación del usuario no tenga que reescribirlas, sino sólo utilizarlas de una manera preestablecida.



De éste modo, el S.O. opera como un puente entre los programas del usuario y la máquina misma. El programa de aplicación no maneja directamente a la máquina, sino que maneja al sistema operativo quien, a su vez, maneja a la máquina. De este modo, se simplifica la construcción de los programas y se facilita el intercambio de programas entre máquinas diferentes. Así, cuando se hace una modificación o mejora del hardware, no se requiere reescribir todo el software sino redefinir al Sistema Operativo (operación que se denomina "Generación del Sistema").

1.2. El Sistema Operativo CP/M.

CP/M es el nombre de un sistema operativo bastante popular en el campo de los computadores personales. Fue desarrollado por quien llegaría a fundar la firma Digital Research en 1974, Gary Kildall. Actualmente está adaptado a más de 3.000 marcas de computadores diferentes, residiendo en más de medio millón de instalaciones en el mundo. Se usa, pues, en varios modelos de máquinas que dispongan de la CPU Z80, 8080 u 8085 y unidades lectoras de diskettes. Es un sistema operativo "mono-usuario", es decir, para una máquina con un sólo terminal. Existen versiones para la CPU 8086 que emplea el ya popular IBM-PC. También existe un sistema operativo multiusuario, llamado MP/M, que es compatible con CP/M.

CP/M administra el uso de programas de aplicación de una vasta colección de software de calidad profesional y de costos razonables. Existe un gran surtido de ensambladores, compiladores, bases de datos (dBase), hojas de cálculo (Supercalc), simuladores, etc. que sólo pueden ser utilizados con CP/M.

Los programas para CP/M son "transportables", es decir, que si funciona en un computador, puede también ser utilizado por cualquier otro que tenga CP/M. Esto, además de su popularidad, trae como consecuencia que un programa preparado para ser ejecutado bajo su supervisión, podrá ser adquirido por muchos usuarios diferentes, aunque no tengan el mismo computador, con la

consiguiente baja de costo por amplitud de mercado. Incluso algunos computadores que no poseen la CPU Z80, pueden agregar una tarjeta adicional, con costo extra, que permite ejecutar programas CP/M. Es el caso de los computadores Apple y Commodore 64, que sin esta extensión son casi un juguete. Este sistema operativo se ha transformado en un estándar de facto para el software de calidad profesional.

Existen varias versiones mejoradas de CP/M, todas ellas compatibles: la primera que se popularizó se denominó CP/M 1.4 y sólo operaba con diskettes de 8 pulgadas. La versión 2.0 y 2.2 permiten utilizar diskettes de 5"1/4, además de los de 8". Actualmente se comercializa la versión 3.0, con varias mejoras con respecto a las anteriores.

El nombre CP/M significa "Programa de Control para Microcomputadores" y no debe ser confundido con la sigla CPM con que designa al "Método del Camino Crítico", utilizado para coordinar actividades complejas.

Aun cuando existen varios otros sistemas operativos de uso corriente, casi todos siguen las líneas de CP/M, que aun siendo uno de los primeros en el terreno de la computación personal, es también uno de los más eficientes.

2. Operación básica.

El Sistema Operativo CP/M es un programa que viene grabado en un diskette junto con varios "Programas Utilitarios" (a los que se les denomina, como se verá más adelante, "comandos transientes"). Este programa es capaz, básicamente de:

- Mostrar la lista de nombres de archivos presentes en un diskette (Comando DIR).
- Cambiar de nombre un archivo (Comando REN).
- Desplegar en la pantalla el contenido de un archivo, de modo de poder leerlo si está escrito en ASCII (Comando TYPE).
- Borrar un archivo (Comando ERA).
- Guardar en un archivo de diskette el contenido de una zona de la memoria de trabajo (Comando SAVE). Y,
- Cargar desde un archivo de diskette un programa escrito en lenguaje de máquina e inmediatamente ejecutarlo. Para esto se escribe simplemente el nombre del programa a ejecutar. Esta operación se denomina en la jerga computacional "invocar un programa". El realizar esta carga y ejecución directa, permite agregar comandos a la lista anterior, en base a programas residentes en discos. Estos nuevos comandos no residen en el código del Sistema Operativo CP/M, sino que son cargados transitoriamente durante su ejecución, en la memoria principal. Por esto se denominan "Comandos Transientes".

Para utilizar este Sistema Operativo, se carga el diskette que lo contiene en la Unidad de Diskette Principal. Luego de identificarse, en la pantalla aparece:

A>

lo que significa que el Sistema Operativo CP/M

está cargado y espera instrucciones. Este símbolo "A>" es una invitación a responder con una orden o comando para realizar algunas de las funciones ya mencionadas de CP/M.

El Programa CP/M identifica a las unidades lectoras de diskettes mediante letras: A, B, C, etc.

La letra A de la invitación a responder (Mensaje A) ya mencionado, nos indica la designación de la Unidad de Diskette sobre la cual está trabajando CP/M. Al mencionar un archivo, cuando no se especifique en qué unidad lectora se está operando, se subentiende que es la unidad indicada por la letra ya mencionada.

Las Unidades de Diskettes tienen las designaciones A; B; etc. Al referirse a una Unidad lectora de Diskette deben colocarse dos puntos después de la letra correspondiente.

Así, por ejemplo, si se quiere utilizar la Unidad de Diskette B como Unidad principal, se procede como sigue:

A> B:

Y se obtiene como respuesta:

B>

Es importante observar siempre que si se quiere grabar o modificar información en un diskette, éste debe estar "montado" o "aceptado" por el Computador. Al cargar CP/M desde un diskette, éste, automáticamente queda "aceptado". Si se cambia el diskette, se debe forzar su aceptación mediante el comando:

ctrl-C (oprimir la tecla CTRL y sin soltarla, pulsar la tecla C). Si no se da este comando, al querer escribir en el nuevo diskette, se recibirá este mensaje:

Bdos Err On A: R/O

lo que significa que el programa Bdos, que es el que escribe la información sobre los diskettes, no reconoce el nuevo disco, no lo acepta como disponible para escribir, sólo lo acepta para leer y de allí la expresión R/O que significa "Read Only" o sea, "sólo lectura".

Todos los comandos operan sobre un archivo o un grupo de archivos. Es decir, los únicos objetos reconocidos por el CP/M son los archivos. Dicho de modo formal, los "operandos" de las "Operaciones básicas", indicados en el comienzo del párrafo, son siempre archivos. Gramaticalmente, se puede decir que los Comandos corresponden a los verbos, las acciones, y los "complementos directos", a los archivos, identificados por su nombre.

3. Nombres de los archivos.

Un archivo es un conjunto de información relacionada, puede ser un programa o puede ser una colección de datos. Un archivo utiliza un área de memoria que depende de su tamaño.

Para CP/M, cada archivo debe tener un "nombre" y un "tipo". El "nombre" debe ser único y debe tener un máximo de 8 caracteres. El "tipo" define la clase de archivo y puede tener hasta 3 caracteres. Ambos elementos se separan mediante un punto.

La información de tipo es optativa, pero es con-

veniente emplearla, puesto que permite clasificar los archivos según sus contenidos. Los tipos más comunes son:

.COM = Archivo de Comando. Es ejecutable directo desde CP/M, no necesita lenguaje. Se invoca con el sólo Nombre. Es un programa en Lenguaje de Máquina.

.BAS = Programa en BASIC. Para ejecutarlo debe cargarse primero el lenguaje BASIC de Microsoft.

.BAK = Archivo original de un archivo actualizado o modificado. Es un reemplazo (BACKUP) automático.

.ASM = Archivo que contiene un programa en Asembler 8080 o Z80.

.\$\$\$ = Archivo de uso transitorio por el propio computador. Normalmente es borrado por el propio Sistema.

.TXT = Texto para un procesador de palabras. No es obligatorio utilizarlo, sino conveniente.

Ejemplos de nombres válidos:

BASIC.COM = Programa en lenguaje de máquina. Probablemente un programa intérprete del lenguaje BASIC.

CASO1.ASM = Programas en Asembler, en versión fuente.

XXX = Archivo sin indicación de tipo. Contenido desconocido.

Se distingue en qué "Unidad de Diskette" está un archivo, especificando la letra (A - B) seguido de ":".

Ejemplo: B:XYZ.COM y A:XYZ.COM son dos archivos que tienen el mismo nombre y el mismo tipo pero están en "Unidades lectoras de diskette" diferentes.

Las letras minúsculas son siempre interpretadas como MAYÚSCULAS.

Nombres Ambiguos y No-Ambiguos.

Se entiende como "Nombre NO ambiguo" al nombre que identifica un archivo en particular, único. Todos los ejemplos anteriores corresponden a esta categoría. Sin embargo, existen algunos comandos que se pueden aplicar a muchos archivos simultáneamente. Para esto es necesario realizar una referencia genérica al grupo de archivos. Esto se conoce como un "Nombre Ambiguo". Los comandos que se aplican de esta forma, usan ciertos símbolos que permiten identificar a un grupo de archivos que cumplen con las especificaciones del nombre ambiguo.

Los símbolos empleados para este fin son:

- * = todos los casos.
- ? = reemplaza una letra.

Ejemplos:

C*.COM = Corresponde a todos los archivos presentes en el diskette cuyo nombre comienza con la letra "C" seguida de cualquier combinación de letras y que son de tipo ".COM". Por ejemplo: "CASA.COM", "COS.COM", "C.COM", etc.

C?.COM = Corresponde a todos los archivos cuyo nombre comience con "C" y luego tengan UNA sola letra adicional. Por ejemplo: CA.COM, C2.COM, etc.

. = Corresponde a todos los archivos de todos los tipos.

*.BAK = Corresponde a todos los archivos de respaldo.

Continúa en la próxima edición

LA INTEGRIDAD DE SU INFORMACION, es una inversión para su empresa.

Elegante gabinete de fina madera.

Medidas: 29 x 17 x 16 cms. de alto.

Terminaciones de lujo.

Cierra con llave para mayor seguridad.

Amplia capacidad: más de 90 diskettes en forma holgada.

Seis divisiones ajustables.

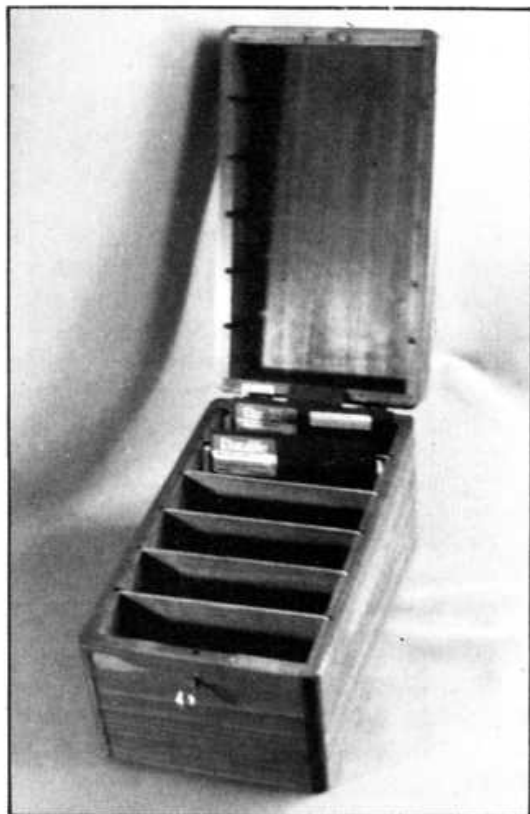
Placa metálica para identificación.

Un producto indispensable junto a cualquier computador.

Adquiéralo por \$ 3.400 IVA incl. en MICROBYTE

Merced 346 of. "F" tel. 393866

Envíos a provincia, agregar \$ 200 para gastos de franqueo.



8086

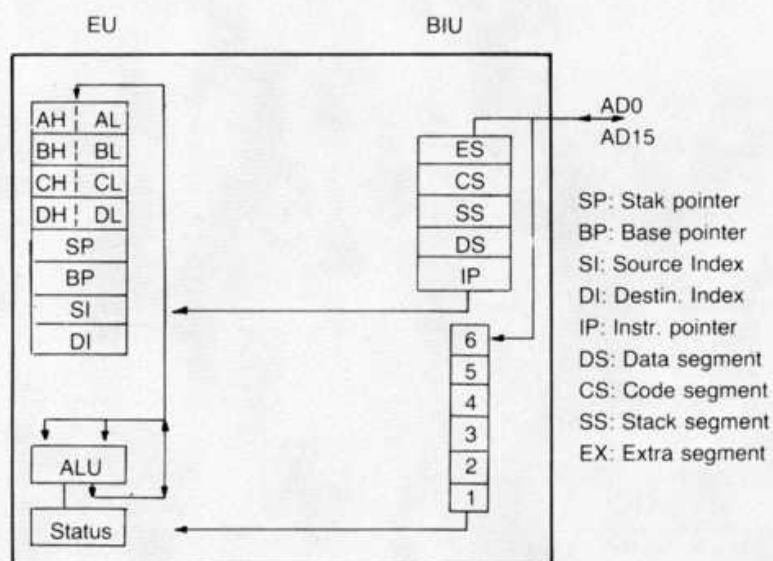
GND	1	40
AD14 ()	2	39
AD13 ()	3	38
AD12 ()	4	37
AD11 ()	5	36
AD10 ()	6	35
AD9 ()	7	34
AD8 ()	8	33
AD7 ()	9	32
AD6 ()	10	31
AD5 ()	11	30
AD4 ()	12	29
AD3 ()	13	28
AD2 ()	14	27
AD1 ()	15	26
AD0 ()	16	25
NMI)	17	24
INTR)	18	23
CLK)	19	22
GND	20	21

VCC	() AD15
() A16/93) A17/S4
) A18/S5) A19/S6
) BHE/S7	(MN/MX
) RD	() RQ/GT0,HOLD
() RQ/TG1,HLDA) LOCK,WR
) S2,M/IQ) S1,DT/R
) SO,DEN) QS0/ALE
) QS1,INTA	(TEST
(READY	(RESET

GND,VCC	— Tierra, alimentación
AD0/AD15	— Bus datos y direcciones
A16-A19	— Dirección
S0-S7	— Status
RD-WR	— Read-Write
READY	— Petición de espera
TEST	— Fin de ocio
INTR	— Interrupción
NMI	— Interr. no enmascarable
RESET, CLK	— Partir, reloj
MN/MX	— Modo mínimo/máximo
RQ/GT0-1	— Control bus local
QS0-1	— Estado de la cola
LOCK	— Control sujetar bus
M/IO	— Memoria o entrada/salida
ALE	— Enganche dirección
DT/R	— Data transmite/recibe
DEN	— Data habilitada
INTA	— Reconocer interrupción
HOLD-HLDA	— Pedir-reconocer entrega bus

8086

Estructura de la CPU



FABRICANTES: INTEL, AMD, NEC, SIEMENS FUJITSU, MITSUBISHI, HARRIS y OKI

MEMORIA: Hasta un megabyte en segmentos de 64KB. La dirección real se obtiene sumando el registro de puntero con el correspondiente registro de segmento, desplazado 4 bits a la izquierda.

CONSUMO: 340 MA. a 5 volts. versión CMOS consume 10MA/MHz.

PRECIOS: Entre US\$ 14 y US\$ 83 según velocidad.

VERSION 5MHz es 12 veces más veloz que INTEL 8080A.

Familia 86:

8087 Coprocesador numérico
 8089 Control de entrada/salida.
 8088 CPU reducida, para 8 bits.
 80186 CPU mejorada y completa.
 80286 CPU mejorada y expandida.
 8284A Reloj.
 8282/3 Puertas de E/S de 8 bits.
 8286/7 Control de buses de 8 bits.
 8288 Controlador buses del sistema.
 8259 Controlador de interrupciones.
 8237A Controlador DMA.
 80150 CPU con Sistema Operativo CP/M.
 M.

Costos 85 de referencia:

8086-5MHz : US\$ 25
 8086-8MHz : US\$ 33
 8086-10MHz : US\$ 83
 8088-5MHz : US\$ 14
 8088-8MHz : US\$ 25
 8086-CMOS : US\$ 31

Tecnología HMOS con implantación de iones en silicio. La pastilla tiene 29.000 transistores en un área de 15 x 15 mm. Todas las líneas son compatibles TTL (excepto el reloj, que es de nivel MOS).

Descripción general

CPU con estructura de tubería (pipeline) para lograr alta capacidad de traspaso. Aunque un tanto primitiva comparada con M68000 o Z8000, su permanencia en el mercado está asegurada por su uso por IBM.

La CPU es de 16 bits con un área de memoria de un megabyte, segmentada en áreas de 64 kB.

La CPU 8088 es una versión reducida, de 8 bits para competir con la CPU Z-80 de Zilog.

La CPU 80186/80188 es la versión completa y madura de esta familia, con reloj, controlador de sistema, interrupciones 2 canales DMA y 3 temporizadores.

La CPU 80286/80288 es un producto evolucionado, orientado al multiproceso y la multitarea, con 16 MB de memoria real y control de hasta 1 GB en memoria virtual (= 1024 megabytes).

A fines del 85 aparecerá el 80386, de 32 bits que con un coprocesador numérico, tendrá una capacidad de 4 millones de instrucciones por segundo.

Diferencias entre 86 y 88

1. La cola FIFO de instrucciones es de sólo 4 bytes.
2. Las operaciones de 16 bits requieren 4 ciclos de reloj extra.
3. Sólo 8 líneas de dirección están multiplexadas.

Modos de Direccionamiento

IMPLICITO
 INMEDIATO
 DIRECTO
 INDEXADO
 REGISTRO
 REGISTRO INDIRECTO
 SEGUN BASE
 INDICE BASE
 CADENA.

LAS BANDERAS O "FLAGS" SON LAS MISMAS DEL 8080 MAS 4 ADICIONALES. ES COMPATIBLE A NIVEL DE ENSAMBLADOR CON 8085.

Las funciones de los distintos alambres de la pastilla cambian según la arquitectura de la configuración en que se emplee. Para esto, existe una pata de la pastilla que conmuta las funciones de 8 líneas desde el "modo mínimo" al "modo máximo". El segundo se caracteriza por existir más de una CPU en la configuración. La CPU tiene 2 partes:

EU: Execution Unit para ejecutar las instrucciones.

BIU: Bus Interface Unit para tomar las instrucciones desde memoria.

Existen 4 registros de uso general:

AX: Acumulador principal.

BX: Puntero base, como HL del Z-80.

CX: Cuenta vueltas, como BC del Z-80.

DX: Data, como el par DE del Z-80.

Cada uno de estos registros es de 16 bits, y se puede utilizar en mitades de 1 byte. Así el registro AX se descompone en los registros AL y AH. En forma similar, existen también los pares de registros BH,BL; CH,CL y DH,DL. Las letras H y L vienen de High y Low.

Todos estos registros se pueden emplear como acumuladores.

Además existen dos punteros y dos registros índices. Los punteros son el del Stack o pila y el puntero Base. Los registros índices son el Índice Fuente, SI y el registro Índice de destino, DI.

Además de los registros de segmentos:

CS: Programa

DS: Datos

SS: Stack o Pila

ES: Extra

Existe una cola FIFO de 6 lugares para tener ya tomadas las instrucciones a ejecutar.

DATAAMERICA

Estado 139

CORONA , MEGA PC

Sistema Multiusuario
Multitarea
Multiprogramación
hasta 8 terminales
Full compatible IBM -PC^{MR}



Teoría de Colas

Guillermo Beuchat
Ing. Civil Industrial U. de Ch.

En diversos artículos anteriores, hemos mostrado una amplia gama de aplicaciones de la computación para resolver problemas de la vida real, siempre con la intención de hacer notar que la computación no tiene por qué ser un fin en sí misma, sino que más bien constituye una herramienta de trabajo. Generalmente, la computación en las empresas se transforma en una actividad absorbente, que merma los recursos financieros y humanos que debieran dedicarse al giro propio del negocio. Por ello en esta serie de artículos he querido presentar aplicaciones no tradicionales, que se salgan del esquema de "Sistemas Administrativos" tan vigente en la actualidad.

Uno de los problemas más interesantes que es posible resolver mediante un computador, es el análisis de las "colas" que se forman frente a los centros en que se presta algún servicio. Por ejemplo, cuando debemos hacer cola en un banco para cobrar un cheque, esperar nuestro turno en un peaje o tratamos de efectuar una llamada telefónica en horas de congestión, estamos en presencia del fenómeno que nos interesa. Es posible analizar matemáticamente las variables que intervienen en estas situaciones, para determinar parámetros de interés económico, tales como el largo promedio de la cola, tiempo medio de espera de las personas que llegan a un sistema, etc.

Sin embargo el análisis matemático, que involucra un cálculo de probabilidades bastante complejo, resulta demasiado engorroso cuando se tiene un sistema formado por más de una unidad de servicio. Por ejemplo, un supermercado en el que hay 20 cajas para pagar los productos.

Tipos de Sistemas de Espera:

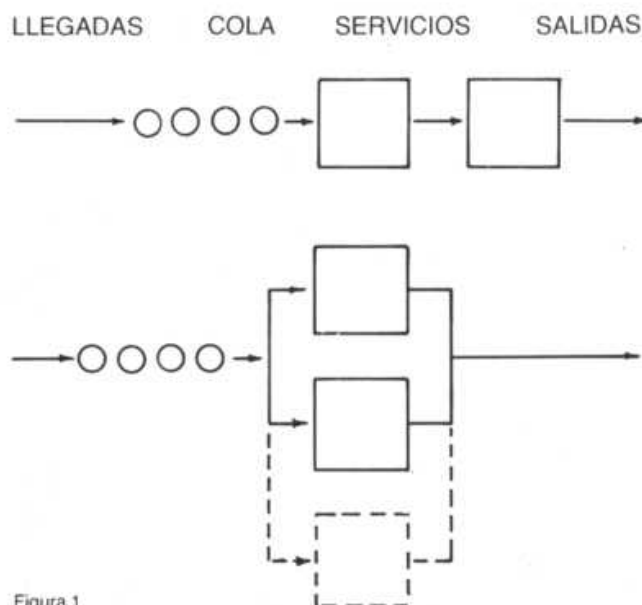


Figura 1

La figura 1 ilustra las 2 clases básicas de sistemas de servicio que pueden construirse. Estos reciben el nombre de Sistema Serie y Sistema Paralelo, de acuerdo a la ubicación de las diferentes unidades de servicio con respecto a la cola. Por ejemplo, las cajas de un supermercado corresponden a un Sistema Paralelo, mientras que las diferentes etapas de una línea de ensamblado de automóviles constituyen un Sistema Serie. Como se puede apreciar, el caso más simple de una sola unidad de servicio corresponde a un caso particular de cualquiera de las dos formas generales enunciadas.

El uso de la simulación computacional nos permite entonces superar las barreras del cálculo probabilístico en estos sistemas complejos, en que existen N unidades de servicio. La idea es que dados los parámetros de operación de un sistema, es posible determinar el número óptimo de unidades de servicio que se deben instalar, la eficiencia que se debe lograr en las mismas, etc. Las variables de decisión son generalmente el número de unidades de servicio y la tasa de atención o despacho. Por otra parte, los criterios de optimización pueden ser minimizar el tiempo de espera de los usuarios, minimizar el número de llegadas al sistema que no se atienden por tener una cola demasiado larga, etc.

El programa BASIC adjunto permite realizar una simulación para un Sistema Paralelo, en que el número de unidades de servicio es variable y lo proporciona el usuario. Pese a que el simulador está diseñado para sistemas en paralelo, bastan algunas modificaciones para modelar un Sistema Serie.

Antes de explicar el funcionamiento del programa, es necesario analizar algunos aspectos teóricos que ayudan a comprender el fenómeno de las colas y permiten una correcta utilización del programa.

Tasas de Llegada y servicio

La tasa de llegada corresponde a la frecuencia promedio con que llegan usuarios a solicitar servicio. Por ejemplo, llegan 50 personas por hora a la ventanilla de un banco. Por otra parte, la tasa de servicio corresponde al promedio de usuarios atendidos o servidos. Por ejemplo, una caja de supermercado es capaz de atender, en promedio, 20 personas por hora.

Sin embargo, no basta para modelar el problema el hecho de conocer las tasas promedio, sino que además es necesario conocer la forma en que se producen las llegadas o las salidas. Es decir, debemos conocer cómo se distribuyen éstas en el tiempo.

Una gran cantidad de estudios estadísticos realizados para diferentes situaciones ha permitido concluir que las llegadas y salidas en la mayoría de los casos se pueden expresar mediante una distribución de probabilidades de Poisson de parámetro T, en que T corresponde a la tasa promedio de llegada o salida. Esto significa, en términos prácticos, que es posible generar computacionalmente los "tiempos" entre las diversas llegadas y salidas, mediante la simple utilización del generador de números aleatorios del BASIC y una fórmula de conversión. Como ya hemos mencionado en otros artículos, los números aleatorios generados por la función RND tienen una distribución de probabilidad uniforme entre 0 y 1.

Para calcular el tiempo entre 2 llegadas sucesivas a la cola, podemos entonces usar la siguiente fórmula matemática, cuya deducción escapa al objetivo de este artículo:

$$t = \frac{-\text{Log}(1-R)}{T}$$

en que t es el tiempo entre 2 llegadas sucesivas a la cola, R es un número aleatorio uniforme entre 0 y 1, y T es la tasa promedio de llegada por unidad de tiempo. Si la unidad de tiempo es horas, entonces t estará dado en horas. La fórmula anterior se aplica también a las salidas, cambiando la tasa de llegada T por la de salida o servicio, S. La función Log utilizada en la fórmula corresponde al logaritmo natural o neperiano, que en BASIC se denomina LOG(x).

Es importante notar que aunque las llegadas y salidas se pueden expresar casi siempre mediante una distribución de Poisson, ello no es un axioma y pueden darse casos que no cumplen lo anterior. Un ejemplo típico se produce cuando se tiene una máquina automática como unidad de servicio. En este caso, la máquina se demorará siempre lo mismo en realizar cada atención, independientemente del largo de la cola u otros factores. Por ello la tasa de servicio S será un valor constante, sin ninguna distribución de probabilidad. En general, podemos decir que la distribución de Poisson representa bastante bien el comportamiento humano, esencialmente aleatorio.

Una suposición importante en el modelo de simulación que se plantea más adelante, es que la tasa de servicio total S debe ser mayor que la tasa de llegada T, de lo contrario, el largo de la cola crecería indefinidamente y la simulación no tendría sentido.

Variables y Parámetros

En un sistema de espera como el descrito, intervienen diversas variables y parámetros:

- factor de utilización: se define como el cociente entre la tasa de llegada y la de salida, para cada unidad de servicio del sistema. Este es el valor teórico, pero para nuestro modelo de simulación consideraremos el factor real de utilización como el cociente entre las llegadas y salidas que efectivamente ocurrieron.

- largo promedio de la cola.
- tiempo promedio de espera en la cola.
- tiempo promedio de espera en el sistema.
- número de personas que fueron atendidas sin hacer cola.
- número de personas que abandonaron el sistema por encontrar la cola en el largo máximo permitido.
- número de unidades de servicio.
- tasas de llegada y salida por unidad de tiempo.
- status inicial del sistema: al iniciar la simulación, un sistema puede tener algunas unidades ocupadas, puede haber usuarios haciendo cola, etc.
- tiempo de simulación: corresponde al tiempo total que está operando el sistema de servicio, por ejemplo 3 horas. Esta variable está relacionada con las tasas de llegada y salida. En efecto, 2 horas de operación no son lo mismo en un supermercado si ocurren a mediodía un día de semana o a mediodía un día sábado. En este último caso, la tasa de llegadas probablemente sea mucho mayor.

Modelo de Simulación

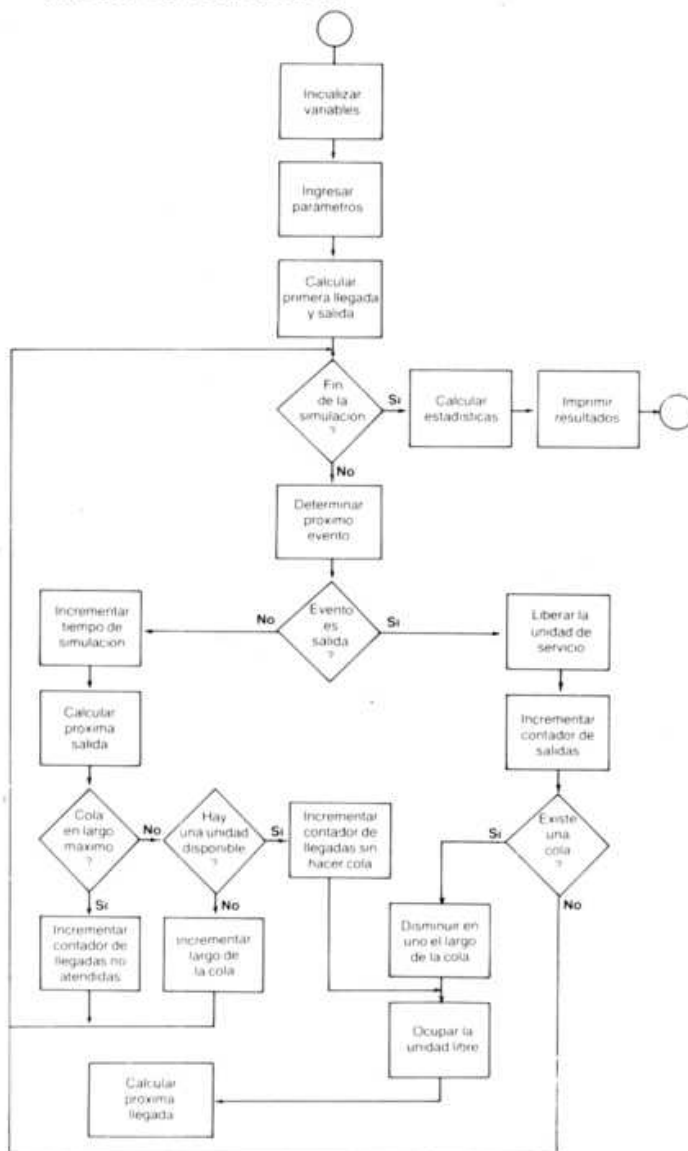


Figura 2. Diagrama de flujo

Para poder realizar la simulación computacional de este problema usaremos la técnica denominada "simulación por próximo evento", que

consiste en mover un "reloj" imaginario, con un algoritmo que va determinando y procesando los eventos a medida que éstos se producen. Por ejemplo, el próximo evento puede ser una llegada a la cola, o un despacho de la unidad de servicio #2, etc. El reloj se hace avanzar generando "tiempos" entre los sucesos usando las fórmulas enunciadas anteriormente.

La figura 2 muestra el diagrama de flujo con la secuencia de procesamiento del algoritmo de simulación, que se implementó directamente en el programa BASIC adjunto.

Descripción del Programa

Las líneas 100-220 permiten ingresar al programa los parámetros de una corrida de simulación. Se debe digitar las tasas de llegada y servicio, el largo inicial y largo máximo de la cola, el número de unidades de servicio y el tiempo de simulación. Además, es necesario especificar si hay alguna unidad de servicio ocupada al iniciar la simulación.

La "rutina inicial" corresponde al cálculo de los tiempos iniciales de llegada y salida. Luego la rutina "avance" verifica si se ha llegado al término de la simulación (línea 355), y además determina el próximo evento, indicando mediante la variable N si es una llegada o una salida.

La rutina "llegada" procesa una llegada a la cola, verificando que ésta no supere el largo máximo permitido e incrementando el largo de la cola si no hay un servicio desocupado.

La rutina "salida" desocupa la unidad de servicio correspondiente y hace avanzar la cola en caso que corresponda, antes de continuar simulando.

Finalmente, las líneas 620-715 calculan las estadísticas y parámetros del sistema, las que se muestran en la pantalla mediante las líneas 720-870. La subrutina de la línea 900 corresponde al generador de números aleatorios.

El ejemplo muestra los resultados obtenidos para un sistema con cinco unidades de servicio, trabajando durante cinco horas. La tasa de llegada es de 50 usuarios por hora, y cada unidad de servicio capaz de atender 11 usuarios por hora. Además, la cola puede llegar a tener un máximo

de ocho usuarios, y su largo inicial es de cinco usuarios.

TASA DE LLEGADA (TOTAL) =? 50
TASA DE SERVICIO (POR UNID.) =? 11

LARGO INICIAL COLA =? 5
LARGO MAXIMO COLA =? 8

UNIDADES SERVICIO =? 5
HAY ALGUNA UNIDAD OCUPADA S/N? S

LA UNIDAD	1	ESTA OCUPADA	S/N? N
LA UNIDAD	2	ESTA OCUPADA	S/N? N
LA UNIDAD	3	ESTA OCUPADA	S/N? S
LA UNIDAD	4	ESTA OCUPADA	S/N? S
LA UNIDAD	5	ESTA OCUPADA	S/N? N

TIEMPO DE SIMULACION =? 5

UNID	SALIDAS	%UTIL	TPROM	STATUS
1	43	94.3	9.11	OCUP.
2	48	88.68	10.82	OCUP.
3	51	90.26	11.29	OCUP.
4	48	85.43	11.23	OCUP.
5	43	85.68	10.03	OCUP.

CONTENIDO COLA			LLEGADAS %	
----------------	--	--	------------	--

ACT	MAX	PROM	TOTAL	S/COLA	%
4	8	2.22	244	81	33.2

ESPERA PROMEDIO (TOTAL) = .05
ESPERA PROMEDIO (COLA) = .07
LLEGADAS SIN ATENDER = 9

La tabla de resultados muestra para cada unidad de servicio, el número de salidas o despachos, el factor de utilización, la tasa promedio real de atención y el status al finalizar la simulación. Además, se muestra el número total de llegadas al sistema, incluyendo aquellas que fueron atendidas sin tener que hacer cola; el número máximo y promedio de usuarios que tuvo la cola; el tiempo de espera promedio para todos los usuarios que ingresaron al sistema (en horas); el tiempo de espera promedio para los usuarios que hicieron cola; y finalmente el número de llegadas que no se pudieron atender.

```

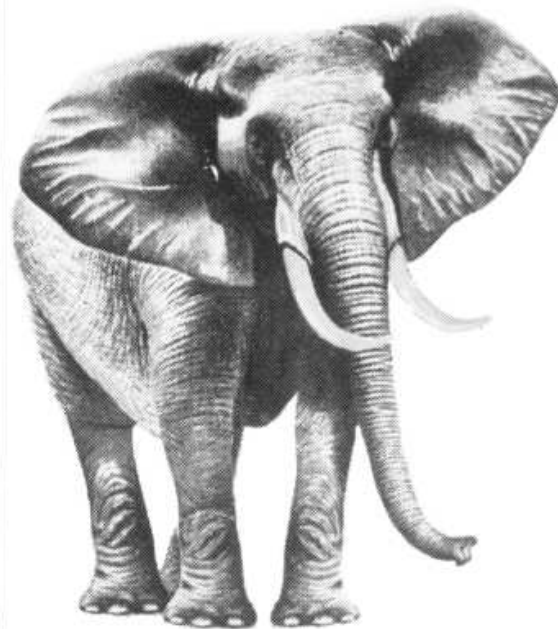
10 REM *****
20 REM * SIMULACION DE UN SISTEMA *
30 REM * DE ESPERA DE MULTIPLES *
40 REM * SERVICIOS EN PARALELO. *
50 REM *
60 REM * GUILLERMO BEUCHAT S. *
70 REM *****
80
85 REM ++ ENTRADA DE PARAMETROS ++
90
100 PRINT CHR$(147): REM BORRA PANTALLA
110 DIM TD(10),S(10),ST(10),CS(10),AS(10)
120 INPUT "TASA DE LLEGADA (TOTAL) =";AI:AI=1/AI
130 INPUT "TASA DE SERVICIO (POR UNID.) =";SI:SI=1/SI
135 PRINT
150 INPUT "LARGO INICIAL COLA =";QL
155 INPUT "LARGO MAXIMO COLA =";LM
156 PRINT
157 INPUT "# UNIDADES SERVICIO =";U
160 INPUT "HAY ALGUNA UNIDAD OCUPADA S/N";D$
170 IF D$="N" THEN 215
180 IF D$<>"S" THEN 160
185 PRINT

```

```

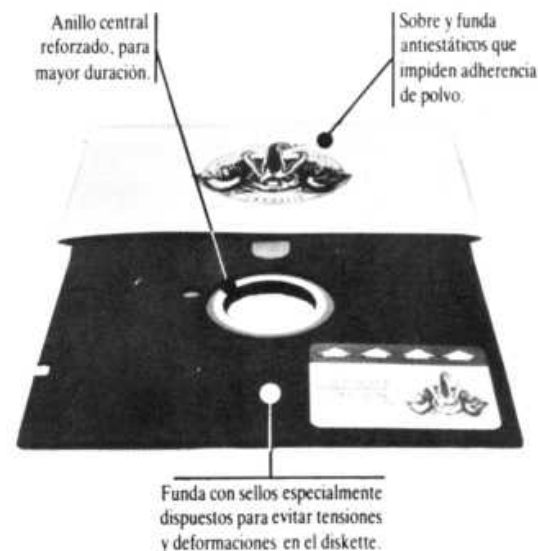
190 FOR K=1 TO U
200 PRINT "LA UNIDAD ",K," ESTA OCUPADA S/N"; INPUT A$(K)
210 NEXT K
215 PRINT
220 INPUT "TIEMPO DE SIMULACION =" ; TT
224
225 REM ++ RUTINA INICIAL ++
226
230 GOSUB 900
240 TA=F*AI
250 FOR K=1 TO U
260 IF A$(K)="S" THEN S(K)=1
270 IF S(K)=0 THEN TD(K)=U*100*AI
280 NEXT K
285
290 REM ***** AVANCE *****
295
300 A=TA-T N=0
310 FOR K=1 TO U
320 IF S(K)=0 THEN 340
330 IF A>TD(K)-T THEN A=TD(K)-T N=N+1 Z=K
340 NEXT K
350 T=T+A
355 IF T>TT THEN 620
360 QT=QT+QL*A
370 FOR K=1 TO U
380 IF S(K)=1 THEN ST(K)=ST(K)+A
390 NEXT K
400 IF N=1 THEN 500
404
405 REM ***** LLEGADA *****
406
410 C1=C1+1
420 GOSUB 900
430 TA=T+F*AI
435 Z2=QL+1
436 IF Z2>LM THEN SN=SN+1 GOTO 300
440 FOR K=1 TO U
450 IF S(K)=0 THEN Z=K C0=C0+1 GOTO 570
460 NEXT K
470 QL=QL+1
480 IF QM<QL THEN QM=QL
490 GOTO 300
495
500 REM ***** SALIDA *****
505
510 S(Z)=0
520 CS(Z)=CS(Z)+1
530 C2=C2+1
550 IF QL=0 THEN 300
560 QL=QL-1
565
570 REM *** UTILIZA SERVICIO LIBRE ***
575
580 S(Z)=1
590 GOSUB 900
600 TD(Z)=T+F*AI
610 GOTO 300
615
620 REM ***** CALCULA ESTADISTICAS *****
625
630 FOR K=1 TO U IF CS(K)=0 THEN 660
640 TS(K)=INT(100*CS(K)/ST(K)+0.5)/100
650 US(K)=INT(10000*ST(K)/T+0.5)/100
660 NEXT K
670 Q0=INT(10000*C0/C1+0.5)/100
680 Q1=INT(100*QT/C1+0.5)/100
690 IF C1=C0 THEN Q2=0 GOTO 710
700 Q2=INT(100*QT/(C1-C0)+0.5)/100
710 QA=INT(100*QT/T+0.5)/100
715
720 REM ***** IMPRIME RESULTADOS *****
725
730 PRINT CHR$(147); REM BORRA PANTALLA
740 PRINT "UNID  SALIDAS  %UTIL  TPROM - STATUS"
745 PRINT "-----"
750 FOR K=1 TO U
760 IF S(K)=0 THEN S$(K)="LIBRE"
770 IF S(K)=1 THEN S$(K)="OCUP."
780 PRINT TAB(8)CS(K)TAB(16)US(K)TAB(24)TS(K)TAB(33)S$(K)
790 NEXT K
800 PRINT
810 PRINT "CONTENIDO COLA          LLEGADAS % "
815 PRINT "-----"
820 PRINT "ACT  MAX  PROM          TOTAL S/COLA  % "
825 PRINT "-----"
830 PRINT QL,TAB(5),QM,TAB(11),QA,TAB(21),C1,TAB(27),C0,TAB(34),Q0
840 PRINT
850 PRINT "ESPERA PROMEDIO (TOTAL)=";Q1
860 PRINT "ESPERA PROMEDIO (COLA) =" ;Q2
865 PRINT "LLEGADAS SIN ATENDER =" ;SN
870 END
890
900 REM ***** GENERA POISSON *****
905
910 F=-LOG(1-RND(1))
920 RETURN

```



Entre un "ELEPHANT" y otros diskettes, las diferencias son del tamaño de un ELEFANTE.

Para que la información procesada por su computador sea exacta y permanezca inalterable, es preciso contar con un diskette altamente confiable. Fabricado en U.S.A., "ELEPHANT" excede con creces las normas internacionales de calidad.



Todas las partidas son homogéneas y están sujetas a los más estrictos controles de calidad.

Compatible con todas las marcas. (Apple, Atari, IBM-PC, Commodore, Epson, NCR, etc.)



Diskettes ELEPHANT
nunca olvidan!

Importa y Distribuye: SINCLAIR CHILE LTDA.

El desarrollo de Software conquista nuevas áreas

"Sistema interactivo de control de calidad"

Jaime Rojas Orrego
Ingeniero Civil Industrial
Universidad de Chile



Independientemente de las condiciones y ámbito en que se encuentre inserta la economía de un país, siempre será importante para el desarrollo de éste, el logro de la calidad tanto en los productos, como servicios ofrecidos por las empresas. Es en los momentos actuales, en que las empresas están buscando formas de reactivación, cuando el cumplimiento de normas de calidad a un bajo costo se convierte en un factor clave. Esto llega a ser más importante y más bien vital en aquellas empresas con una fuerte actividad exportadora, las cuales se enfrentan con mercados cada vez más exigentes y competitivos, los cuales deben conquistar y mantener.

Por otro lado, la función de controlar la calidad en cualquier tipo de empresa es una actividad compleja, produciéndose a menudo altos volúmenes de información, escaso procesamiento y análisis posterior de datos y un precario apoyo a la gerencia en la toma de decisiones oportunas, confiables y financieramente lógicas.

NCR-CHILE emprendió el desarrollo de un sistema computacional de apoyo a la gestión gerencial del Control de Calidad. Este sistema está especialmente dimensionado para la realidad de las empresas chilenas y fue desarrollado inicialmente como tema de tesis por dos Ingenieros Civiles Industriales; los Srs. Jaime Rojas Orrego y Kamel Lahsen Robres asesorados por el Ingeniero Consultor Sr. Pablo Cereceda Bravo, especialista en Control de Calidad.

El sistema será comercializado en breve por NCR y sus principales características son las siguientes:

- Posee cuatro subsistemas o módulos, enlazados en una modalidad de menús, lo cual permite al usuario sin experiencia computacional elegir la actividad de procesamiento necesaria en forma simple y natural. Además en el Menú Principal se puede optar por un módulo especial que tiene como finalidad administrar el sistema, permitiendo obtener auditorías, manejar claves secretas y otras actividades de apoyo al procesamiento de datos.

- Cada opción de procesamiento está adecuadamente protegida tanto en el ingreso a ella (Vía palabras claves e identificación de usuario), como en su operación rutinaria (Vía validación de procedimientos).

- El diseño del sistema está orientado hacia un manejo interactivo, con énfasis en la actualización inmediata de la información. Este aspecto es de especial importancia, debido a que en el área del control de calidad la oportunidad en las decisiones es un factor clave.

- La principal fuente de datos, corresponde a muestreos estadísticos, realizados en cada etapa de la producción. La información obtenida al procesar dichos datos permite tomar acción sobre lotes completos de materiales y materias primas o productos terminados, sobre procesos productivos, sobre proveedores, etc., en base a la información obtenida de pequeñas muestras.



Fig. 1.- Estructura modular del Sistema.

El sistema está construido de forma que cada módulo pueda funcionar como un sistema completo, si la aplicación en particular lo requiere.

Las características principales de cada subsistema se muestran a continuación.

A. Subsistema de Control de Materiales y Materias Primas

- Provee en forma mecanizada planes de muestreo por atributos correspondientes a la norma oficial chilena 44 o su equivalente la norma Americana

Ya somos terceros!

ASC/Hewlett-Packard

A menos de un año de la introducción en Chile de la línea completa de Computadores Hewlett-Packard, importantes empresas manufactureras, comerciales y financieras han adquirido nuestros equipos (especialmente nuestro Computador Personal HP 150) para solucionar sus problemas de información.

Su preferencia por nuestra solución ha permitido que en este corto tiempo, ASC/Hewlett-Packard haya alcanzado un privilegiado **tercer lugar** en el mercado de Computadores Personales en Chile, con su innovador modelo HP 150.

Este auspicioso resultado de ASC en Chile, unido a logros **financieros** (aumento de 4.7, a sobre 6.0 Billones de Dólares en ventas, un 54% en las utilidades y un 20% en el presupuesto de investigación y

desarrollo durante el período 83-84), **tecnológicos** (exitosa introducción de 8 nuevos productos en un año) y de **soporte** (por 4º año consecutivo HP fue catalogado N° 1 en soporte y satisfacción de usuarios por DATAPRO) de Hewlett-Packard en los EE.UU., confirman a ASC/Hewlett-Packard como una inversión con futuro.

Las razones del éxito se deben principalmente a la suma de dos factores: la excelente calidad de los productos Hewlett-Packard y la probada experiencia de ASC en el área de la Computación. Si además nuestros equipos tienen un precio razonable y adecuado a la disponibilidad de recursos financieros de las empresas chilenas, estamos seguros de ofrecer la mejor solución para aumentar la productividad de su empresa.

Conozca la excelencia de nuestra línea de productos.

Computadores Personales: Series 80 y 100

Computadores Comerciales: HP 3000

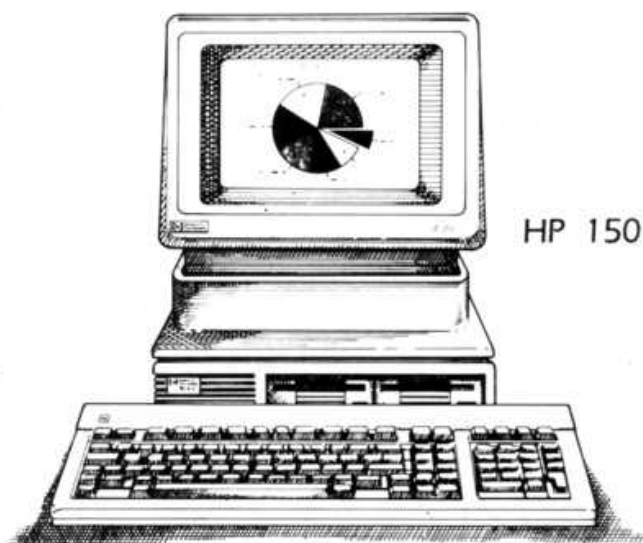
Computadores de Control de Procesos: HP 1000

Computadores Científicos: HP 9000

Periféricos: Plotters, Impresoras, Terminales, Digitalizadores.

Le invitamos a integrarse a la Familia ASC/Hewlett-Packard.

Llámenos, queremos contribuir con nuestra experiencia al crecimiento de su empresa.



futuro con experiencia.



REPRESENTANTE OFICIAL PARA CHILE DE LA LINEA
COMPLETA DE COMPUTADORES HEWLETT-PACKARD

Militar Standard 105 D (ver recuadro). Estos planes se manejan en forma computacional, simplificando enormemente las operaciones de muestreo de materiales y materias primas, y evitando errores en la aplicación de los mismos.

— Entrega informes sobre calificación de proveedores, confiabilidad de los planes de muestreo y costos de la calidad (inspección, transporte, almacenamiento, etc.)

NORMA CHILENA 44

La Norma Chilena Oficial 44 es una adopción y traducción de la MILITARY STANDARD, SAMPLING PROCEDURES AND TABLES FOR INSPECTION BY ATTRIBUTES, MIL - STD 105-D y concuerda totalmente con la norma ISO 2859-1974.

Consiste en un conjunto de tablas y procedimientos para realizar muestreo por atributos. Los planes de muestreo contenidos en la norma son aplicables especialmente, entre otros, a la Inspección de: a) Productos terminados, b) Componentes o materias primas, c) Materiales en proceso, d) Abastecimientos en bodega, e) Operaciones de mantenimiento, f) Informaciones o registros y g) Procedimientos administrativos.

La Inspección por atributo es la inspección mediante la cual cada unidad inspeccionada es clasificada simplemente en defectuosa o no defectuosa, o se cuenta el número de defectos por unidad, de acuerdo a una especificación determinada o conjunto de ellas. Para realizar esta clasificación se recomienda clasificar los defectos posibles de un ítem según gravedad. Los defectos se agrupan generalmente en una o más de las categorías siguientes: 1) Defecto Crítico, 2) Defecto Mayor, 3) Defecto Menor, 4) Defecto Secundario.

Para operar con estas tablas de muestreo es necesario definir previamente algunos parámetros:

a) Nivel de Inspección: Define una relación entre el tamaño de la muestra y el tamaño del lote sometido a inspección.

b) Tipo de Muestreo: Dobles y Múltiples.

c) Tipo de Inspección: Existen tres tipos: Normal, Rigurosa y Reducida.

d) Nivel aceptable de calidad (AQL): en %

Con todos estos parámetros predefinidos, más el tamaño del lote sometido a inspección, se ingresa a la tabla correspondiente (de múltiples entradas) y se encuentra el plan de muestreo, que consiste en un tamaño de muestra, un número de aceptación y un número de rechazo (para el caso de muestreo simple). Si el número de defectuosos o de defectos (según corresponda) encontrados en la muestra, es menor o igual al número de aceptación, el lote es aceptado; si es mayor o igual al número de rechazo, el lote es rechazado. Para el caso de muestreo doble y múltiple se puede caer en regiones de indiferencia que obliguen a tomar muestras adicionales.

B. Subsistema de Control de Procesos Fabriles

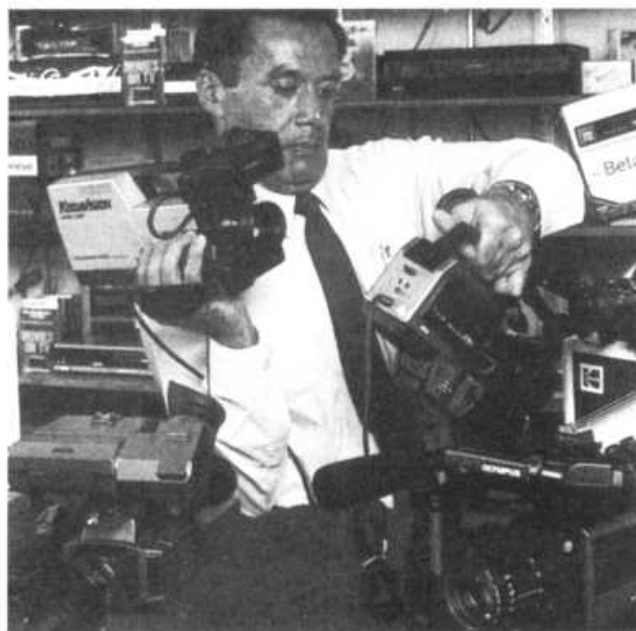
— Procesa los datos obtenidos por operaciones de muestreo a las líneas de producción, realizando los cálculos necesarios para la confección de gráficas de control.

— Entrega información sobre cumplimiento de los productos en procesos, con respecto a las especificaciones de ingeniería y diseño, estimando los costos de la calidad por incumplimiento de especificaciones.

C. Subsistema de Control de Productos Terminados

— Provee en forma mecanizada planes de muestreo por atributos correspondientes a la norma oficial chilena 44 para ser aplicados a la inspección de productos terminados.

— Entrega informes sobre calidad de los productos terminados, confiabilidad de los planes de muestreo y costos de la calidad (inspección, desecho, reproceso, calificación en calidades inferiores).



D. Subsistema de Control de Fallas en los usuarios

— Procesa los datos obtenidos de las notas de servicio de garantía y post-venta de los productos terminados que han fallado después de vendidos.

— Obtiene informes de: Fallas de los productos vendidos (clasificados según fecha de producción), tiempo promedio de fallas de los productos vendidos, costo de atención, costo de garantía, etc.

El sistema completo se encuentra construido en dos versiones: Una para funcionar bajo el control de los sistemas operativos IMOS III e IMOS V y otra bajo el control de ITX. Esto permite la implantación del sistema en cualquiera de los equipos NCR-8250, 8270, 9030, 9040 y 9300.

Además se ha planeado la obtención de una tercera versión para funcionar bajo el control del sistema operativo UNIX, lo cual permitirá la implementación del sistema en equipos NCR - TOWER y otros.

DETALLES DE CONSTRUCCION

El sistema se encuentra totalmente programado en Lenguaje Cobol, consta de 36 programas para el usuario y 4 programas adicionales que se usan para crear las tablas estadísticas (planes de muestreo, curva normal, factores, etc). Los programas para el usuario suman un total de 52.800 líneas de programación. Siendo los de mayor longitud, aquellos que manejan las tablas de muestreo (simulan la operación de la Norma 44). Tienen una longitud aproximada de 3.000 líneas de programación.

Debido a la estructura de Menú que posee el sistema, cada programa se especializa en una actividad determinada (creación y mantención, proceso y listadores), lo cual implica que el tiempo de corrida de cada programa esté limitado por la velocidad del operador, el cual percibirá respuestas casi instantáneas. La excepción a este punto está dada por los programas listadores, que están limitados por la velocidad de la impresora. Esto se soluciona usando archivos SPOOL que independizan el tiempo de corrida del programa con la impresión.

El realizar toda la programación del Sistema de Lenguaje Cobol trajo algunas dificultades en el módulo de procesos que requiere del cálculo de raíces cuadradas. Estos problemas se solucionaron usando tablas en los casos de valores enteros y ciertos

algoritmos numéricos de convergencia rápida en los casos de valores fraccionarios. Se estima que esta solución es menos costosa y de igual eficiencia que el haber usado rutinas en otros lenguajes (por ejemplo BASIC) combinadas con los programas en COBOL.

El sistema maneja un total de: 17 archivos maestros, 10 archivos de transacciones y 6 archivos de tablas (sólo lectura).

De acuerdo a los antecedentes de que dispone el autor, existe en algunas empresas chilenas cierto grado de computarización en el área del Control de Calidad, pero comprenden sólo algunas de las actividades del área. A nivel internacional, al parecer ocurre lo mismo, conociéndose algunos desarrollos puntuales y a la medida de la empresa que lo desarrolló.

De lo que se está seguro es de que este sistema es único a nivel mundial como paquete computacional generalizado de apoyo al Control Total de la Calidad.

El sistema hasta el momento, no ha sido aún probado en una experiencia práctica, pero se espera realizar algunas instalaciones durante este año, las cuales si son exitosas, podrán convertir este software en un producto de exportación.

5 razones por las cuales vale la pena pagar por la diferencia Dysan

1. Superficie
testeada al 100%.
2. Técnicas avanzadas
de pulido.
3. Lubricante Dy¹⁰™
4. Certificación de Auto Carga.
- 5^a Representante oficial
para Chile
respaldo y garantía.



INFORNA LTDA.
Cía. de Informática Nacional Limitada



Números Romanos

BIENVENIDOS AL BASIC

En general, convertir números decimales a números romanos ha sido una tarea tediosa pero extremadamente interesante como ejercicio intelectual. Si bien la mayor parte de los números se pueden transformar directamente, algunos, o más específicamente los 4 y 9 con sus respectivas decenas o centenas tienen una construcción especial, debiendo agregar un prefijo al número inmediatamente superior.

Así, si 3 equivale a III, el número 4 es representado por IV. Lo mismo $30 = XXX$ mientras que cuarenta es XL. El 8 es VIII y el 9 es IX, etc.

Hacer un programa que realice esta transformación es complicado, especialmente por esas excepciones, sin embargo, nada es imposible para nuestros diestros programadores, por lo que aquí les presentamos el problema resuelto y la explicación de cómo se hace.

En este programa, sólo aceptaremos números entre 0 y 3999. La razón, es que para números superiores, se utilizan otras letras para representar los 5.000, 10.000 etc. y la verdad es que no nos son conocidas. Como en general no se usan números romanos mayores a los límites que nos fijamos, no tiene mayor importancia esta limitación.

En la línea 5, partimos definiendo dos cadenas de caracteres. Una, N\$, para las letras que representan a los números romanos y la otra, D\$, vacía en la cual llenaremos el resultado de la transformación.

De las líneas 20 a 45, determinamos si el número contiene unidades, decenas, centenas o miles y de acuerdo a eso asignamos diferentes valores a las variables S y T. S va a representar la posición dentro de la cual comenzamos a trabajar en la cadena N\$. Por ejemplo, si el



número contiene miles, entonces S valdrá 1 y esto representa al primer carácter de la cadena, la M = 1000 en romano. La variable T, nos va a servir para determinar el orden de magnitud del número, miles, centenas, decenas o unidades para luego poder separarlo en los dígitos que lo componen.

De la línea 50 en adelante comenzamos un ciclo, en el cual debemos ir separando el número elegido primero en miles, luego en centenas, decenas y unidades. Para esto, utilizamos la instrucción INT, la cual nos va a dar la parte entera del número dividida por T. Si el número era 1984 entonces la parte entera del número dividido por T (1000) va a ser igual a 1.

Obteniendo ese número, el flujo del programa salta a la línea 200 donde se transforma ese número a su equivalente romano, el que es guardado a continuación en D\$. Entre las líneas 210 y 240 se realiza este proceso, tomando en cuenta si el número es 4 o 9, que es un caso especial.

Para ubicar el equivalente romano, utilizamos la instrucción MID\$, la que nos permite dividir una cadena de caracteres, en este caso N\$, para obtener de ésta un sólo carácter, cuya posición en la cadena está indicada por S.

En la línea 210, revisamos para el caso de que el dígito ob-

tenido en la línea 50 sea 4 o 9. De ser así, su equivalente romano lo obtenemos inmediatamente, sumando a nuestra cadena resultado el carácter indicado por "S" en N\$ y el carácter anterior a éste de ser 4 el dígito o el carácter ubicado dos posiciones antes de S si es 9.

En esta instrucción utilizamos un operador lógico ($R = 9$) como elemento de una suma. Si es verdadero que R es 9 entonces el resultado de esa operación lógica es 1, por lo que el resultado en los paréntesis es 2. Si R no es 9 entonces el resultado de la operación lógica es 0, por lo que el resultado de los paréntesis es sólo 1, encontrando así, cuantos espacios hacia la izquierda hay que mover el puntero "S" para encontrar los valores correspondientes a dígitos 4 o 9.

En la línea 220, verificamos si el dígito es igual o mayor que cinco. Si es así, entonces con el puntero S-1 llegamos al carácter que representa los 5, 50 o 500 en la cadena N\$. Luego le restamos 5 al dígito (R). Si R era 5, entonces el resultado es 0, lo que significa que encontramos la equivalencia romana del dígito. Si R era mayor que 5 entonces debemos completar la equivalencia con el ciclo en la línea 230. En este, S va a apuntar a las unidades, decenas o centenas (I, X, C) las que hay que agregar al carácter anterior.

Por ejemplo, si nuestro dígito es 8 y corresponde a las decenas, entonces el programa primero ubica a L = 50 y para los 30 restantes, escribe 3 veces X, obteniendo "LXXX".

Después de obtener la equivalencia final de un dígito, el programa vuelve a buscar el siguiente si lo hay en la línea 70. En esta rutina, entre las líneas 70 y 80, se le resta al número (NUM) la parte entera de éste dividida por T y luego multiplica-

da por T. Si tenemos el caso de 1984, entonces la operación es:

$$\begin{aligned}\text{NUM} &= 1984 - (\text{INT} (1984/1000) * 1000) \\ &= 1984 - \text{INT} (1,984) * 1000 \\ &= 1984 - 1000 \\ &= 984\end{aligned}$$

Luego el programa separa el número obtenido, tomando el primer dígito, el 9 para buscar su equivalencia y así sucesivamente hasta completar el número.

Para saber cuando se terminó el número se tiene que cumplir una de dos condiciones en la línea 80. Si S es 9, entonces el puntero ya está apuntando al último dígito, por lo que el número terminó, o si al efectuar la resta en la línea 70, el número ya es cero.

Si bien este programa no es excesivamente de mucha utilidad, las lecciones que extraemos de él pueden ser muchas. En primer lugar, aprendimos a separar un número en sus respectivos dígitos mediante la ins-

trucción INT y a utilizar una cadena de caracteres para luego ir accedendo sus componentes con la instrucción MID\$. Luego, ocupamos operadores lógicos dentro de una operación matemática para así obtener diferentes resultados de acuerdo a si la operación lógica es verdadera o falsa. Por último, mostramos que para resolver problemas complejos es indispensable una gota de ingenio. En el próximo número seguiremos revisando programas interesantes.

```
5 N$ = "MDCLXVI":D$ = ""
10 INPUT "INGRESE NUMERO ";NUM
20 IF NUM >= 1000 THEN S = 1:T = 1000: GOTO 50
30 IF NUM >= 100 THEN S = 3:T = 100: GOTO 50
40 IF NUM >= 10 THEN S = 5:T = 10: GOTO 50
45 S = 7:T = 1
50 R = INT (NUM / T)
52 PRINT R
60 GOTO 200
70 NUM = (NUM - INT (NUM / T) * T):S = S + 2:T = INT (T / 10)
80 IF S = 9 OR NUM = 0 THEN 500
85 GOTO 50
200 IF R = 0 GOTO 70
210 IF R = 4 OR R = 9 THEN D$ = D$ + MID$ (N$,S,1) + MID$ (N$,S - (1 + (R = 9)),1): GOTO 70
220 IF R >= 5 THEN D$ = D$ + MID$ (N$,S - 1,1):R = R - 5: IF R = 0 THEN 240
230 FOR I = 1 TO R:D$ = D$ + MID$ (N$,S,1): NEXT I
240 GOTO 70
500 PRINT D$: GOTO 5
```

276	CCLXXVI	1517	MDXVII
57	LVII	1761	MDCCLXI
1584	MDLXXXIV	1115	MCXV
830	DCCCXXX	1224	MCCXXIV
890	DCCCXC	1979	MCMLXXIX
717	DCCXVII	485	CDLXXXV
146	CXLVI	1472	MCDLXXII
254	CCLIV	775	DCCLXXV
1034	MXXXIV	1197	MCXCVII
1471	MCDLXXI	1681	MDCLXXXI
1274	MCCLXXIV	1222	MCCXXII
1774	MDCCLXXIV	1868	MDCCLXXVIII
1792	MDCCXCII	454	CDLIV
830	DCCCXXX	1087	MLXXXVII
196	CXCVI	1077	MLXXVII
1154	MCLIV	674	DCLXXIV

Programa su propio ensamblador

Eduardo Ahumada M.

La intención de este artículo es explicar como funciona el Ensamblador para VIC-20 que apareció en el número anterior de MICROBYTE, de tal forma que usted, amigo lector, pueda programar su propio Ensamblador, sin importar el computador que tenga, pues los principios básicos de diseño son similares en todas las máquinas.

El Ensamblador es un programa que traduce un texto escrito en Lenguaje Assembler, al que denominaremos "Fuente", a un texto expresado en Lenguaje de máquina, y que llamaremos "Objeto". Al proceso de traducción lo denominaremos "Compilación". Un Editor es el programa que permite ingresar el texto Fuente al computador y posteriormente modificarlo (editar) para hacer correcciones.

La primera decisión que se debe tomar es cómo distribuir los recursos del computador entre todos estos elementos. Para el caso del VIC, sólo se disponía de 3,5 Kbytes de RAM para almacenar el Ensamblador, el Editor, el Fuente y el Objeto. Claramente nos falta espacio, de modo que llegamos a un compromiso.

Se decidió que dada la naturaleza de este Ensamblador no se compilarían programas muy grandes, de tal forma que se reservaron 0.25K de RAM para almacenar el Objeto. El texto fuente se almacenaría en un archivo en cassette. En esta forma el Ensamblador es cargado a memoria desde cassette y luego al ejecutar lee una a una las instrucciones del Fuente y va colocando su traducción en el área reservada para el objeto.

El Editor sería un programa aparte que se cargaría desde cassette permitiendo a continuación crear o modificar el Fuente y luego grabarlo en cassette para su posterior uso por el Ensamblador.

De esta forma nuestro sistema para programar en Assembler se compone de dos programas: un Ensamblador y un Editor. En equipos con suficiente memoria ambos programas podrían fundirse en uno solo, y así ahorrar los tiempos que demoran en cargarse a memoria. Estudiemos primero al Ensamblador.

Para entender mejor los procesos internos de un Ensamblador debemos definir las características de la entrada y de la salida. Generalmente la sintaxis del Assembler está ya definida por el fabricante de la CPU, MOS Technology en el caso de la 6502. Las instrucciones de Assembler tienen tres elementos: un rótulo opcional, que sirve para identificar la instrucción y así poder hacer referencia a ella desde otra parte del programa, un código mnemotécnico, que identifica el tipo de operación a realizar, y un operando, que determina sobre qué información o dato actuará la instrucción.

El Assembler de la 6502 permite una amplia variedad de tipos de operandos, cuya sintaxis resumimos en la Tabla I. Estos operandos determinan el modo de direccionamiento que se usará en la instrucción traducida a lenguaje de máquina.

Tabla I: Sintaxis de Operandos en Assembler 6502

Operando	Formato
Implícito	(se omite el operando)
Absoluto	Valor
Relativo	Rótulo
Indexado por X	Valor,X
Indexado por Y	Valor,Y
Indirecto Pre-indexado	(Valor,X)
Indirecto Post-indexado	(Valor),Y
Inmediato	\$Valor
Indirecto	(Valor)

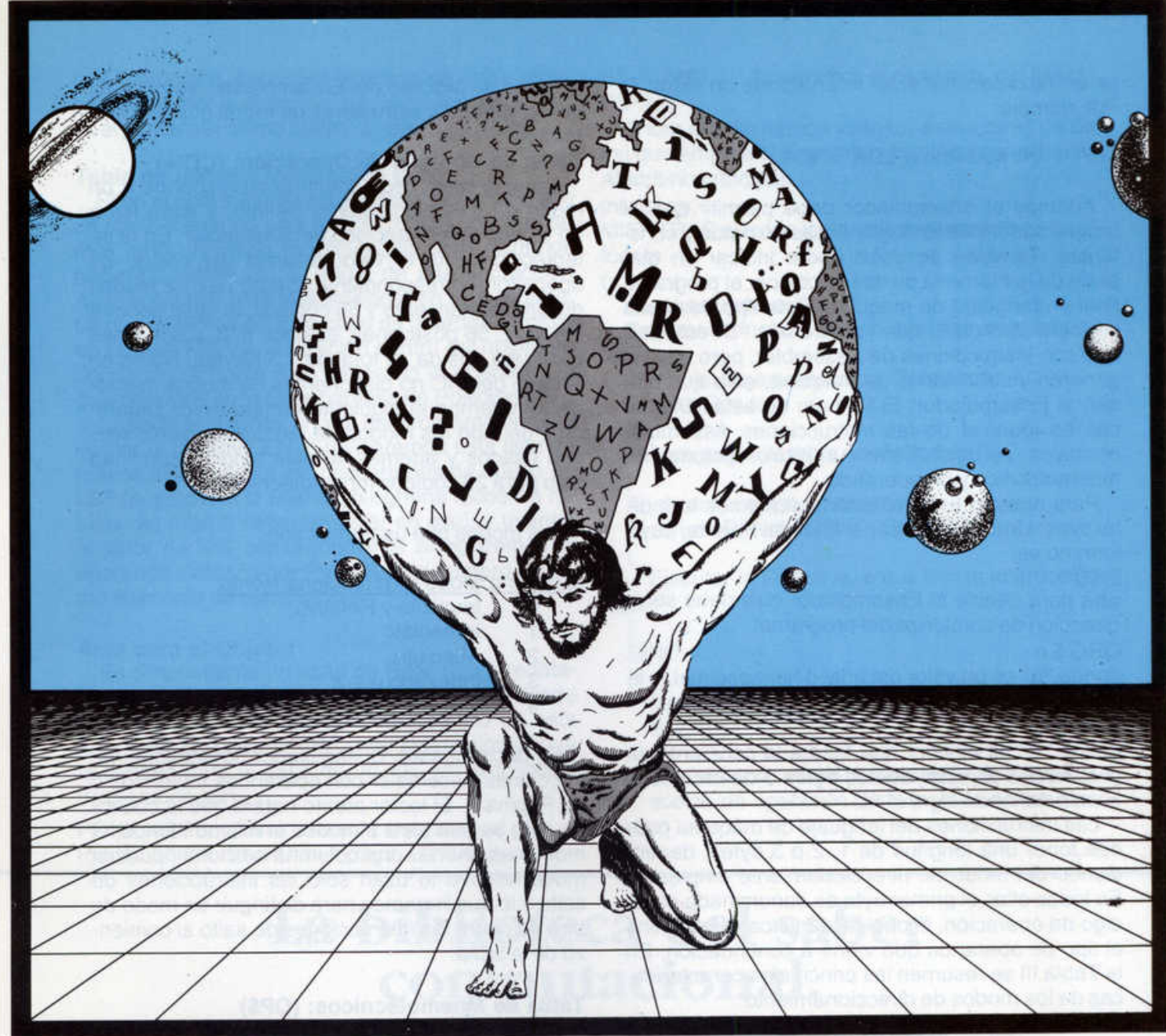
Donde *Valor* puede ser un Rótulo o una expresión aritmética.

Los códigos mnemotécnicos para cada tipo de operación aparecen en la primera columna de la Tabla II. Las columnas restantes indican el código de operación de la instrucción en lenguaje de máquina que corresponde según el modo de direccionamiento empleado.

Tabla II: Códigos de Operación 6502

Modo	0	1	2	3	4	5	6	7	8	9	10	11	12
ABC	AD	65	69	7D	79	61	71	75					
AND	2D	25	29	3D	39	31	35						
ASL	0A	0E	06		1E		16						
BCC								90					
BCS								80					
BEQ								F0					
BIT	2C	24											
BRL								30					
BNE								D0					
BPL								10					
BRE	00												
BVC								50					
BVS								70					
CLC	18												
CLD	DB												
CLI	5B												
CLV	BB												
CMP	ED	65	69	DD	D9	C1	D1	D5					
CPX	ED	EA	ED										
CPY	ED	FA	ED										
DEC	CE	CA		DE			DA						
DEX	CA												
DEY	88												
EOR	40	45	49	5D	59	41	51	55					
INC	EE	EA		FE			FA						
INX	EE												
INY	CB												
JMP	4C							6C					
JBR	70												
JBA	AD	65	69	DD	D9	C1	D1	D5					
JBR	AE	AE		7E				BA					
JBY	AC	64	68	BC			BA						
LSR	46	4E	4A		5E		5A						
NOP	EA												
ORA	0B	05	09	1D	19	01	11	15					
PHA	40												
PHP	48												
PLA	58												
PLP	50												
RDL	7A	7E	7A		7E		7A						
ROR	6A	AE	6A		7E		7A						
RTL	60												
RTS	60												
SBC	ED	E5	E9	FD	F9	E1	F1	F5					
SEC	38												
SED	F8												
SET	78												
STA	BD	65	69	DD	D9	C1	D1	D5					
STX	BE	EA					9A						
STY	BC	EA					9A						
TAX	AA												
TAY	9A												
TXA	9A												
TXB	9A												
TXC	9A												
TYA	9A												

Para simplificar nuestro Ensamblador es mejor que el operando de la instrucción sea tan sólo un número o un rótulo. Los números podrán escribir-



GAD BARTOV S.P.

En el mundo de los procesadores de palabras, CPT lleva todo el peso.

CPT es la única compañía en el mundo dedicada exclusivamente al desarrollo de equipos y sistemas para el procesamiento de palabras y la automatización de oficinas, siendo sus equipos los más completos del mercado. Estos permiten trabajar en 18 idiomas con pantalla de fondo blanco y letras negras, simulando perfectamente la hoja de papel.

CPT es el procesador ejecutivo e inteligente con capacidad suficiente para llevar todo el peso de su oficina.



... aporta soluciones!

se en hexadecimal si se le antepone un signo \$. Por ejemplo:

LDA # 10 equivale a LDA # \$ A

Además el ensamblador debe permitir que el programador pueda definir áreas de datos y constantes. También se debe poder indicar en qué parte de la memoria se desea colocar el programa final en lenguaje de máquina. Este tipo de cosas se logra usando lo que llamaremos "Directivas", que son instrucciones de Assembler, pero que no generan instrucciones de máquina, sino que dirigen al Ensamblador. El formato de estas Directivas es igual al de las instrucciones Assembler normales, es decir, tienen un rótulo opcional, un mnemotécnico y un operando.

Para nuestro Ensamblador definiremos tres directivas: Una para indicar el final del Fuente, cuyo formato es:

END

otra para decirle al Ensamblador cuál debe ser la dirección de comienzo del programa:

ORG \$ n

donde "n" es un valor decimal o hexadecimal, y finalmente una directiva para definir constantes:

Rótulo DC # N

donde "n" es el valor que tendrá la constante, la cual tendrá un largo de dos bytes o de un byte si se especifica el signo "#".

Las instrucciones del lenguaje de máquina pueden tener una longitud de 1, 2 ó 3 bytes, dependiendo del modo de direccionamiento empleado. En todas ellas el primer byte es denominado el código de operación, el cual la identifica y determina el tipo de operando que viene a continuación. En la Tabla III se resumen las principales características de los modos de direccionamiento.

Tabla III: Modos de Direccionamiento para la CPU 6502.

Modo	Largo	Descripción
1) Implícito	1	El operando es omitido.
2) Absoluto	3	El operando es una dirección de 2 bytes, el byte menos significativo precede al más significativo, dicha dirección contiene el dato con el cual opera la instrucción.
3) Página 0	2	El operando es el byte menos significativo de una dirección cuyo byte más significativo se asume cero. Es decir la dirección está entre 0000 y 00FF.
4) Inmediato	2	El operando es el byte que la instrucción usará para procesar.
5) Absoluto indexado por X	3	Es como el modo Absoluto, pero a la dirección que indica el operando se le suma el contenido del registro X, para determinar la dirección que tiene el dato que usará la instrucción.
6) Absoluto indexado por Y	3	Como el anterior, pero usando el registro Y.
7) Indirecto pre-indexado	2	Es como el modo de página 0, pero se suma el valor del registro X al operando, y el contenido de la dirección así obtenida es la dirección que tiene el dato a usar.
8) Indirecto post-indexado	2	Igual al modo de página 0, pero la dirección contiene una dirección a la cual se suma el valor del registro Y, para obtener la dirección que contiene el dato a usar.
9) Página 0 indexado por X	2	Es como el modo Absoluto Indexado pero con las características del modo de Página 0.
10) Página 0 indexado por Y	2	Como el anterior, pero usando el registro Y.
11) Relativo	2	El operando es un byte expresado en notación de complemento de 2, entre -128 y +127. Este valor se suma a la dirección de la instrucción siguiente para así obtener la dirección que será usada.
12) Indirecto	3	Es como el modo Absoluto, pero la dirección indicada por el operando contiene la dirección que ocupará la instrucción.

Ahora que ya hemos estudiado las características de la entrada y de la salida, podemos analizar

la forma de proceso del Ensamblador. Veamos en primer lugar las estructuras de datos que emplearemos:

Tabla de Códigos de Operación: (CO%)

En esta tabla cada columna corresponde a un modo de direccionamiento distinto, y cada fila a un código mnemotécnico del Assembler. En la intersección está el valor decimal del código de operación correspondiente. Como hay 12 modos de direccionamiento y 56 códigos, la tabla debiera tener 12*56 posiciones, es decir 672, a dos bytes cada una nos da un total de 1.3 Kbytes. Como en el caso del VIC no disponemos de mucha memoria, tomaremos el siguiente compromiso: Dejaremos de lado los modos de direccionamiento menos usados y además la tabla tendrá capacidad sólo para 26 códigos mnemotécnicos.

Los modos que usaremos son:

Columna Modo de direccionamiento

0	Implícito y Relativo
1	Inmediato
2	Absoluto
3	Indexado por X
4	Indexado por Y

Es decir, nuestro Ensamblador no admitirá instrucciones Assembler con operandos Indirectos o de Página 0. El lector atento notará que la columna cero se usa para 2 modos al mismo tiempo, el motivo es ahorrar una columna adicional, pues el modo relativo lo usan sólo las instrucciones de salto y lo que haremos para distinguir un modo de otro es poner las instrucciones de salto al comienzo de la tabla.

Tabla de Mnemotécnicos: (OPS)

Esta tabla contiene los códigos mnemotécnicos que es capaz de traducir nuestro Ensamblador. Es paralela a la tabla anterior. Gráficamente:

OPS		CO%				
		0	1	2	3	4
PS —>	0 BNE	208	0	0	0	0
	1 BEQ	240	0	0	0	0
	2 STA	0	0	141	157	153
	3 LDA	0	169	173	189	185
P1 —>	—	—	—	—	—	—
	25 INC	0	0	238	254	0

PS es un puntero que indica hasta qué fila llegan las instrucciones de salto. P1 indica el final de la tabla. La ventaja de usar este tipo de tablas es que el programa es muy fácil de modificar. Por ejemplo, si necesitamos una instrucción que no está dentro de las 26 de la tabla sólo necesitamos ponerla en el sitio de una que no usemos.

Tabla de rótulos definidos (LB\$):

Para cada instrucción que tenga rótulo, el Ensamblador lo almacenará en la tabla LB\$, y la dirección de la instrucción la colocara en la tabla paralela de Direcciones (V%). De esta forma cada vez que se encuentre un rótulo en el operando de

una instrucción, ésta será buscada en dicha tabla para así saber la dirección que le corresponde. P2 es el puntero al último elemento de estas tablas.

Tabla de Rótulos Indefinidos (RI\$):

Si en el operando de alguna instrucción aparece un rótulo y éste no está en LB\$, diremos que estamos ante un rótulo indefinido, es decir estamos haciendo referencia al rótulo de una instrucción Assembler que está más adelante en el programa. Lo que haremos en este caso es guardar el rótulo en la tabla RI\$ y a la posición dentro del objeto en donde debía haber sido generado el operando la guardaremos en la tabla paralela de Posición P%.

Cuando lleguemos al final del programa (indicado por la directiva END) revisaremos esta lista de rótulos indefinidos y los buscaremos de nuevo en LB\$ (puesto que al final del programa todos los rótulos del mismo deberían estar en ella), y usando el valor de V% correspondiente arreglaremos el operando indicado por P%. P3 es el puntero al último elemento de estas tablas.

Area para el Objeto:

Es simplemente un trozo de RAM que arrebatamos al BASIC, y cuyo comienzo está indicado en S0. La próxima posición libre dentro de esta área esta indicada por PC. Las instrucciones que la definen para el caso de un VIC sin ampliación de memoria son:

POKE 52,29: Disminuimos la memoria del
POKE 56,29: BASIC en 1 página (256 bytes).

S0 = 7424: Apuntamos al comienzo del área.
PC = 0

Ahora que ya hemos visto las estructuras de datos que emplea el Ensamblador, veamos cuál es el Algoritmo principal:

Inicializar

MIENTRAS queden instrucciones assembler en el fuente

(

Leer una instrucción.

Separar la instrucción en sus tres componentes:

Rótulo, Mnemotécnico y Operando.

Si tiene Rótulo

(

Agregarlo a la tabla de rótulos (LB\$).

)

Analizar el código de operación.

Si es una Directiva (END, DC u ORG)

(

Ejecutar la Directiva, e ir a leer la instrucción sgte.

)

SINO

(

Buscar el mnemotécnico en la tabla OP\$.

Si está

(

Indexarlo con el tipo de operando para obtener el código de operación de la instrucción de máquina.

na.

)



La biblioteca del saber computacional.

CompuDiccionario

El más completo glosario de términos computacionales traducidos al español con una clara explicación de los conceptos.

Si entender un manual de operación de equipos o software le significa horas de arduo trabajo, esto ya serán cosas del pasado.

Una herramienta de trabajo indispensable junto a todo computador y una excelente ayuda para quienes se inician en este fascinante mundo. **\$ 980**



CompuEscolar

Desde los aspectos más cotidianos a los más complejos, la computación está revolucionando nuestra forma de vivir y de pensar.

En compuEscolar, encontrará la mejor introducción al tema, guiándolo desde los aspectos más simples -historia, aplicaciones diarias, etc.- a detalles de la construcción y programación de los modernos ordenadores. **\$ 870**

CompuLenguajes

Los primeros computadores se programaban en códigos binarios. Desde entonces, han sido desarrollados decenas de nuevos lenguajes.

Su evolución, estructura de los principales lenguajes clásicos (APT, Fortran, Algol, Lisp, Cobol, APL, PL/I, etc), los lenguajes modernos, declarativos y los desarrollos en inteligencia artificial. **\$ 980**

CompuBasic

Basic es el lenguaje universal de los microcomputadores. Sus aplicaciones van desde programas educativos a complejos sistemas administrativos y financieros. Manuales Basic ya conocíamos pero ninguno tan completo como CompuBasic, con más de 180 páginas de amena instrucción, con ilustraciones y numerosos ejemplos y ejercicios para poner en práctica de inmediato sus conocimientos.

\$ 980

Distribuye para todo Chile Revista MICROBYTE.

Adquiera su ejemplar en Microbyte Merced 346 Of. "F" fono 393866 o en sus distribuidores autorizados:

Asicom Mac Iver 115

Infogroup Providencia 2623

Latindata Nueva York 68

Logos Mac Iver 372

Sinclair Chile L. Thayer Ojeda 1234

Teorema Agustinas 1169.

Sres. Microbyte, Merced 346, Of. "F".

Sírvase enviar a mi dirección..... Ejemplar (es) de.....

Adjunto, el valor, más \$ 100 por ejemplar para gastos de franqueo por correo certificado.

NOMBRE:

DIRECCION:

```

SINO
(
    Error, mnemotécnico inválido.
)
)
Colocar en memoria el código de operación.
Colocar en memoria el valor del operando (1 ó
2 bytes).
Imprimir línea del listado.
)
Revisar rótulos indefinidos.
FIN.

```

Cada programador posee su propio estilo para desarrollar programas, el mío consiste en tener claro el algoritmo principal y a continuación comenzar a programar pequeñas subrutinas que parezca que pueden ser útiles más tarde.

Estas subrutinas van formando un grupo de "herramientas" que permiten hacer en forma más fácil el resto del programa. En esta forma, voy a ir mostrando a continuación los componentes más básicos del Ensamblador y después los de mayor nivel. Las implementaciones en BASIC que pongo de ejemplos son las correspondientes al Ensamblador para VIC-20, pero usando una notación indentada para facilitar su lectura.

Subrutina 1: Retorna en P\$ la próxima palabra contenida en F\$, a partir de la posición P.

```

805 P$ = " "; Supongamos que la pa-
L = LEN(F$); labra no está.
806 IF P <= L AND MID$(F$, P, 1) = " " THEN
    P = P + 1; Buscamos la primera le-
    GOTO 806 tra de la palabra.
807 C$ = MID$(F$, P, 1);
    IF P <= L AND C$ <> " " THEN Mientras hay letras las
        P$ = P$ + C$; agregamos a P$.
        P = P + 1;
    GOTO 807
808 RETURN

```

Note que P, al terminar la subrutina, queda apuntando a la posición inmediatamente siguiente al término de la palabra P\$.

Subrutina 2: Usando la subrutina anterior, retornaremos en L\$, I\$ y O\$ el Rótulo, Mnemotécnico y Operando respectivamente, de la instrucción Assembler contenida en F\$. El rótulo es opcional, pero si está presente debe comenzar en la primera posición de F\$.

```

800 P = 1; Comenzamos a partir de la pos. 1
L$ = " "; y suponemos que no hay rótulo.
IF LEFT$(F$, 1) <> " " THEN Si la primera posición está con una
    GOSUB 805; letra, buscamos un Rótulo.
L$ = P$:
801 GOSUB 805; Buscamos el Mnemotécnico.
I$ = P$:
GOSUB 805; Buscamos el Operando.
O$ = P$:
RETURN

```

Subrutina 3: Calcula el valor de un operando numérico contenido en X\$ y devuelve su valor en B1 y B2 (byte más significativo y byte menos significativo). El formato de X\$ puede ser: (\$)n

donde el signo \$ indica que n es un número hexadecimal.

```

820 P = 1; Suponemos un valor cero al operando.
B1 = 0; Si no comienza con $, el operando es
IF LEFT$(X$, 1) <> "$" decimal y lo convertiremos de carácter
THEN a numérico con la función VAL.
    B2 = VAL(X$); Si es hexadecimal hacemos un ciclo
    GOTO 824 por todos sus dígitos.
821 P = P + 1; Ejemplo: la letra A tiene el ASCII 65, al
IF P > LEN(X$) THEN 824 restarle 55 queda un valor de 10.
822 L = ASC(MID$(X$, P, 1)); El 0 tiene un ASCII 48, al restar 48 nos
IF L > 64 THEN da el valor correcto.
    B2 = B2 * 16 + L - 55;
    GOTO 821 Si es posible representar el valor en un
823 B2 = B2 * 16 + L - 48; solo byte, retornamos, en caso contra-
    GOTO 821 rio lo dividimos en byte más significati-
824 IF B2 < 255 THEN vo (B1) y menos significativo (B2).
    RETURN
825 B1 = FNA(B2);
B2 = FNB(B2);
RETURN

```

Funciones empleadas:

5 DEF FNA(X) = INT(X/256) Calcula byte más significativo.
6 DEF FNB(X) = X - B1*256 Calcula byte menos significativo.

Subrutina 4: Recibe en X\$ un Rótulo y devuelve en B1 y B2 la dirección que le corresponde. Si el rótulo no está en la Tabla de Rótulos definidos (LB\$) entonces no retorna ningún valor, sino que lo agrega a la tabla de rótulos indefinidos (RI\$), y en P% coloca la posición actual dentro del objeto.

```

830 FOR I = 0 TO P2: Buscamos X$ en LB$.
IF LBS(I) = X$ THEN 832
831 NEXT: Si no está, agregamos X$ en la tabla
P3 = P3 + 1; ri$, y colocamos en P% la posición ac-
RI$(P3) = X$: tual dentro del objeto (+ 1 para saltarse
P%(P3) = S0 + PC + 1; el código de op.).
RETURN
832 B1 = FNA(V%(I)); Si está, retornamos la dirección del ró-
B2 = FNB(V%(I)); tulo.
RETURN

```

Subrutina 5: Esta subrutina recibe el operando de la instrucción Assembler en O\$ y retorna un valor en O según el tipo de operando. Este valor corresponde a la columna de la tabla CO%. En B1 y B2 devuelve el valor de dicho operando. Si el operando es de un solo byte este es devuelto en B2.

```

810 B1 = 0; Suponemos un valor 0 al operando.
B2 = 0; Si O$ es nulo, se ha omitido el operando
IF O$ = "" THEN op. implícito.
O = 0;
RETURN
811 IF LEFT$(O$, 1) = "*" THEN Si O$ comienza con un "*" entonces es
    THEN un op. inmediato, colocamos el resto en
        O = 1; X$ y calculamos su valor con la subrutina
        XS = RIGHTS(O$, LEN tres
        (O$)-1); Suponemos operando absoluto
        GOSUB 820; Si termina en "X"
        RETURN - indexado por X.
812 O = 2; Si termina en "Y"
XS = O$: - indexado por Y.
IF RIGHTS(O$, 2) = ",X"
    THEN
        O = 3;
        XS = LEFT$(O$, LEN
        (O$)-2);
        GOSUB 820;
        RETURN
813 IF RIGHTS(O$, 2) = ",Y" THEN
    THEN
        O = 4;
        XS = LEFT$(O$, LEN
        (O$)-2);
814 L$ = LEFT$(O$, 1);
IF L$ = "$" OR (L$ >=
    "0" AND L$ <= "9") THEN
    GOSUB 820;
    RETURN
815 GOSUB 830;
RETURN

```

Subrutina 6: Recibe en X1 un número entre 0 y 15, y retorna en C\$ el equivalente hexadecimal.



Innovadora Tecnología
Computacional

**Una Educación
Superior para una
técnica en
permanente
superación.**

CREZCA

PROFESIONALMENTE

La empresa moderna requiere hoy, y demandará en el futuro, calificados especialistas en el campo de la informática y computación.

NCR, empresa líder en el mundo, elabora la tecnología computacional.

Venga a nosotros.

FORMACION PROFESIONAL PLAN MODULAR

- OPERACION COMPUTADORES IMOS ITX. UNIX®
- PROGRAMACION BASIC
- PROGRAMACION COBOL: BASICO AVANZADO - TALLER
- PROGRAMACION C
- ANALISTA DE SISTEMAS: * BASICO * ANALISTA PROGRAMADOR NCR
- DISEÑADOR BASE DE DATOS
- AUTOMACION DE OFICINAS

REQUISITOS

CUARTO MEDIO Y
APROBAR TEST DE
APTITUDES

* PLANES ESPECIALES PARA PROFESORES

INICIO DE CLASES

25/03/1985

ESCUELA DE INFORMATICA Y NEGOCIOS NCR

ANDRES DE FUENZALIDA 147, PROVIDENCIA

FONOS: 380013 anexo 273 - 2515685

```

837 IF X1 > 9 THEN
    CS = CHR$(X1 + 55):
    RETURN
838 CS = CHR$(X1 + 48):
    RETURN

```

Subrutina 7: Recibe en X un valor que debe colocar en la próxima posición disponible dentro del objeto, y además debe imprimir en la pantalla la configuración hexadecimal del byte generado.

```

835 X1 = INT (X / 16):
    GOSUB 837:
    XS = CS:
    X1 = X - X1 * 16:
    GOSUB 837:
    XS = XS + CS
836 PRINT XS:
    POKE S0 + PC, X:
    PC = PC + 1:
    RETURN

```

Colocamos en X1 el medio byte más significativo de x, y lo convertimos a hexadecimal. Hacemos lo mismo con el medio byte menos significativo.

Imprimimos configuración hexadecimal de X y luego colocamos x en la próxima posición del objeto. Actualizamos PC para apuntar a la posición siguiente.

Subrutina 8: Calcula el valor de un operando relativo. Recibe en O\$ el nombre del rótulo hacia el cual se desea saltar. Hay dos casos: el rótulo ya estaba definido (es decir está en la tabla LBS) en cuyo caso se retorna en B1 la diferencia entre la dirección de la instrucción actual y la dirección correspondiente a O\$, y el otro caso es que este es un salto hacia adelante, y el rótulo no ha sido definido aún, en este caso se ingresa O\$ a la tabla de rótulos indefinidos, pero con signo negativo para distinguir que esta es una referencia de 1 byte (a diferencia de referencias para operandos de 2 bytes).

```

840 B1 = 0:
    FOR I = 0 TO P2:
        IF LBS (I) = O$ THEN 842
841 NEXT I:
    P3 = P3 + 1:
    RIS(P3) = O$:
    P%(P3) = -(S0 + PC + 1):
    RETURN
842 B1 = V%(I) - (O0 + PC + 1) + 256:
    RETURN

```

Buscamos el rótulo en la tabla LBS. Si no está, lo guardamos en la tabla RIS, y su posición en P% (en forma negativa para indicar operando de 1 byte). Si está, devolvemos el desplazamiento (sumamos 256 para colocarlo en complemento de 2).

Subrutina 9: Coloca el cursor en la fila Y, columna X. Esta es una función sólo para el VIC. En otras versiones de BASIC hay una función equivalente (y probablemente más simple). A muchos dueños de VICs les parecerá muy complicada, pero la uso porque es la forma más económica de hacerlo (en términos de RAM).

```

900 Z = 7680 + Y * 22:
    W = INT (Z / 256):
    Z = Z - W * 256:
    POKE 209, Z:
    POKE 210, W:
    POKE 211, X:
    POKE 214, Y:
    RETURN

```

Subrutina 10: Espera hasta que el operador presione alguna tecla, y entonces retorna en X\$ el carácter ingresado. Permite detener el proceso cuando la pantalla se llena y así dar tiempo al operador para que la lea.

```

910 GET X$:
    IF X$ = "" THEN 910
911 RETURN

```

La función GET retorna un nulo en caso de que no se ha presionado nada, de modo que hay que repetir hasta tener respuesta.

Ahora que ya hemos visto las subrutinas básicas, veamos cómo se implementa el algoritmo principal que expuse anteriormente:

Módulo 1: Leer una instrucción, separarla, ver si

tiene rótulo, y si es así agregarlo a la tabla de rótulos.

```

80 INPUT #1, F$:
    X = 0:
    GOSUB 900:
    PRINT F$:
    IF LEN (F$) > 15 THEN
        Y = Y + 1
90 X = 16:
    GOSUB 900:
    GOSUB 800:
    IF L$ <> "" THEN
        P2 = P2 + 1:
        LBS (P2) = L$:
        V% (P2) = O0 + PC

```

Leer instrucción del Assembler en F\$.

Posicionamos el cursor en la col. 0 e imprimimos la instrucción fuente. Si esta instrucción es mayor de 15 letras avanzamos una línea (recuerde que el VIC tiene sólo 22 col.) Colocamos el cursor en la col. 16. Separamos la instrucción. Si el rótulo existe, lo colocamos en la tabla de rótulos, junto con la dirección actual dentro del objeto.

Módulo 2: Analizar el código de operación, si es una directiva ejecutarla y si no buscarla en la tabla OP\$. Si está indexarlo con el tipo de operando y obtener el código de operación de la instrucción de máquina.



**ESCUELA DE CIENCIAS
DE LA COMPUTACION
DR. JAIME MICHELOW**

Centro de Formación Técnica Decreto N° 11 del 13/1/83

CARRERAS

**Programación de Microcomputadores (2 años)
* Análisis de Sistemas (4 años)**

CARACTERISTICAS ESPECIALES

- 26 años de experiencia
- Becas de estudio
- Laboratorio con 23 Computadores
- Cursos con 30 alumnos
- Profesores Full-time
- Cursos permanentes de extensión y capacitación (SENCE)
- Bibliotecas, Casinos, Central de Apuntes, Equipos de Audio, Equipos de Fotocopiado, Equipos de Video.
- Contactos académicos internacionales.

Universidad de Columbia New York U.S.A.
Universidad de Denver Colorado U.S.A.

REQUISITOS DE ADMISION

Licencia de Ed. Media. 2 fotos carnet.
Certificado de nacimiento.

SEDES

**Jorge VI 185 Tel.: 2125631-2463283
Las Condes (Apoquindo Alt. 5100)
* París 823 Tel.: 381980
Stgo. Centro.**



**INSTITUTO DE EDUCACION Y
ENSEÑANZA SUPERIOR**

```

105 IF IS = "END" THEN
    PRINT:
    GOTO 200
107 IF IS = "ORG" THEN
    GOSUB 810:
    O0 = B1*256 + B2:
    GOTO 150
110 IF IS<> "DC" THEN 115
112 GOSUB 810:
    X = B2:
    GOSUB 835:
    IF O <> 1 THEN
    X = B1:
    GOSUB 835
114 GOTO 150
115 FOR K = 0 TO P1:
    IF IS = OPS(K) THEN 125
120 NEXT:
    PRINT "ERR-2":
    GOTO 150
125 IF K<= PS THEN
    X = CO% (K, 0):
    GOSUB 835:
    GOSUB 840:
    X = B1:
    GOSUB 835:
    GOTO 150
135 GOSUB 810:
    X = CO% (K, 0):
    IF X = 0 THEN
    PRINT "ERR-3":
    GOTO 150
136 GOSUB 835:
    IF O > 0 THEN
    X = B2:
    GOSUB 835
140 IF O > 1 THEN
    X = B1:
    GOSUB 835
150 Y > Y = Y + 1:
    GOTO 75

```

Si es la directiva END salimos del ciclo principal.

Si es la directiva ORG colocamos la variable de comienzo lógico del objeto (O0) en el valor solicitado en su operando.

Si es la directiva DC, vemos el valor de su operando y generamos el primer byte del mismo. Luego vemos si el operando era de 2 bytes, generamos el segundo.

Buscamos el mnemotécnico en la tabla OPS.

Si no está en ella imprimimos un mensaje de error, y continuamos. Si el mnemotécnico corresponde al de una instrucción de salto (indicado por PS), generamos el objeto correspondiente a su cod. de operación y luego el de su operando (el cual es un operando relativo).

Es una instrucción cualquiera. Indexamos la tabla CO% para obtener su cod. de operación. Si no tiene código válido para ese tipo de operando, imprimimos un error. Generamos objeto de cod. de operación. Si tiene operando, generamos el primer byte del mismo. Si el operando es de dos bytes, generamos el segundo byte.

Avanzamos una línea en la pantalla.

Módulo 3: Revisar rótulos indefinidos.

```

200 CLOSE 1:
    IF P3< 0 THEN 250
205 FOR I = 0 TO P3:
    FOR J = 0 TO P2:
    IF LBS (J) = RIS(I) THEN
    215
210 NEXT J:
    PRINT "ERR-4", RIS(I):
    Y = Y + 1:
    GOTO 225
215 X = P%(I):
    IF X > 0 THEN
    B1 = FNA (V%(J)):
    B2 = FNB (V%(J)):
    POKE X, B2:
    POKE X + 1, B1:
    GOTO 225
220 B1 = X + S0-O0 +
    V%(J):
    POKE -1-X, B1
225 IF Y > 20 THEN
    GOSUB 910:
    PRINT HS:
    Y = 2
230 NEXT I

```

Cerramos el archivo del Fuente. Si no hay rótulos indefinidos nos saltamos este módulo. Revisamos para cada rótulo inde- toda la lista de rótulos.

Si no aparece en ella, imprimimos un error, y el rótulo.

Analizamos la posición donde quedó el operando sin definirse. Si es positiva, era un operando absoluto de 2 bytes. Lo generamos.

Si no, era un operando relativo de un byte. Lo generamos.

Si se está acabando la pantalla esperamos que el operador presione una tecla y luego la borramos para continuar.

y con esto terminamos nuestro estudio del Ensamblador. Por supuesto, no hemos visto detalles como la inicialización de las tablas o la impresión del objeto pues hacerlo obligaría a extender demasiado este artículo, sin embargo, el lector interesado puede fácilmente identificarlos en el listado del Ensamblador para VIC-20 y adaptarlos a su propio caso.



Software

Administración, finanzas

Multiplan
Inventory
General Ledger
Account Payable
Account Receivable
Easy Finance
Easy Calc 64
Data Manager 2

Ajedrez

Colossus Chess
Sargon III

Procesamiento de texto

Wordpro 3 Plus
Easy Script
Easy Mail
Script 64

Lenguajes

Logo
Pilot
Assembler
Pascal
Simons Basic

Educativos

EasyMath - Easy Count

Servicio Técnico Especializado

Kinder Konzept

A little much/or less
Hodge Podge
Monkey Math
Sword of Sagoal
Easy quiz Easy Lesson
The Word Machine
The Name Machine

Juegos

Apple Cider
Archon
Audio Video
Axis Assassin
Aztec Challenge
Bagit Man
Bandits
Bat Attack
Boulder Dash
Beach-Head
Benji
Blue Max
Break Dance
Bristles
Buck Rogers
Burge Time
Camels
Castle of Dr.
Castle Wolfenst
Chopliter

Congo Bongo

Cosmic Tunnels
Crossfire
Crossfire
Cyclons
Davis Midnight
Decatblon
Defender
Dig Dug
Dino Eggs
Donkey Kong
Dragons Den
Drelbs
Droll
Dungeon of
Fair Warning
Falcon Patrol
Flight Simulator
Flying Ace
Football
Forbidden Forest
Fort Apocaly
Froger II
Frogger
Galaxian
Galaxion
Galaxy
Gangbusters
Geisterhaus
Gvruss

Hard hat Mack

Hover Bovver
I am the C-64
J Bird
Jammin
Juice
Jumpman
Jumpman Junior
Jungle Hunt
Kindercomp
Kong
Laser Zone
Le Mans
Lode Runner
M.U.L.E.
Maze Man
Miner 2049
Missile Comand
Montezuma's
Moon Buggy
Moon Patrol
Moon Shuttled
Motor Mania
Mountain King
Mr Robot
Mr Wimpy
Munchy
Murder on the Zinderne
Neutral Zone
Night Mission

Oil's Well

One on One
Pakakuda
Perplexian
Pitfall
Pitstop
Pitstop II
Pogo Joe
Pole Postion
Pool
Pooyan
Popeye
Pro Golf
Protector II
Puc Man
Quintic Warrior
Radar Rat Race
Raid On
Raid Over Moscow
River Chase
Rob. of the Lost Tomb
Sam
Sat Mat I
Saucer Attack
Seven Cities
Shamus Case
Skier
Skramble
Slalom
Solo Flight

Space Pilot

Speed Racer
Spelunker
Spy vs Spy
Spyhunter
Squish'em
Star Wars
Stellar 7
Suicide Strik
Summer Games
Super Pipeline
Survivor
Sword of Fargoal
Telegrand
The Factory
The Heist
Tooth Invaders
Train
Tri-Math
Triad
Upper Reaches
Wall Street
Zaxxon
Zeppelin
Zylogon

Estos Programas para su Commodore 64 y además gran variedad de Cartridges para el VIC 20.

Para servicio técnico, confíe su equipo a buenas manos.

Electroquin Av. Bdo. O'Higgins 980 - Of. 304 Fono: 382224 - Santiago.

ELECTROQUIN

RELAX!!!
El Software del año
\$ 14.400
Incluye: Hardware
Software Manual

Computadores y la enseñanza del inglés

Eddie Edmundson

DIRECTOR

Instituto Chileno Británico de Cultura - Concepción



(Este artículo se basa en una charla y demostración que el autor dio en el Sinclair Centre: Santiago a miembros del "English Teachers Group" el 4 de diciembre de 1984, en la Universidad Austral de Valdivia a profesores del Instituto de Lenguas Extranjeras el 17 de diciembre).

Ha habido recientemente mucho interés en las aplicaciones de microcomputadores en la enseñanza del Inglés como lengua extranjera. Este artículo pretende resumir lo que se ha logrado, especialmente respecto al trabajo de investigadores británicos.

Hasta 1979, el trabajo con computadores y el aprendizaje de lenguas fue casi exclusivamente reservado a las universidades. Hubo cerca de 20 años de experimentación en las universidades en este campo basada en grandes computadores tipo "mainframe" equipados con terminales y lenguajes complejos de computación. La mayor parte del trabajo realizado en este período fue asociado con el aprendizaje programado (o "instrucción programada"), con un énfasis en el computador como máquina de enseñanza.

Ahora que hay más disponibilidad de microcomputadores ha aparecido una variedad más amplia de tipos de programas y actualmente hay menos énfasis en la instrucción programada. Dos destacados expertos en microcomputadores en la educación, John Higgins y Tim Johns, han sugerido la siguiente tipología de programas:

INTERACTIVO	TUTORIAL
	NO TUTORIAL

(Por "interactivo" se entiende que existe una interacción entre el/los estudiante(s) y el teclado).

Tutorial: Este tipo de programa está asociado con el aprendizaje programado. Ejemplos de programas tutoriales son ejercicios de práctica controlada y pruebas de conocimientos. Generalmente los estudiantes trabajan solos con tales programas. Estos programas tienden a ser muy inflexibles, ya que sólo pueden controlar y evaluar.

No tutorial: Ha habido una reacción fuerte a las limitaciones autoimpuestas del programa tipo tutorial. La mayor parte de la investigación reciente, especialmente en Gran Bretaña, se concentra en el tipo de programa donde el computador asume un papel más creativo y se exige un papel más creativo del estudiante. Hay dos avances muy interesantes en esta área; programas **analíticos** (que analizan textos de algún modo) y programas **sintéticos** (que se basan en una gramática 'sintética' del lenguaje).

Programas Analíticos. Estos programas analizan un texto para realizar algo con él, y el texto es la base para la actividad de aprendizaje. Por ejemplo, el computador puede ser programado para mostrar en la pantalla un texto en inglés con algunas de las palabras omitidas. Los alumnos tienen que adivinar cuáles son las palabras que faltan buscando ayuda en el contexto general. Frecuentemente son la base de juegos con los estudiantes 'comprando' palabras, pero perdiendo puntos si lo hacen, y siendo premiados cuando adivinan correctamente.

Programas Sintéticos. Un programa sintético entrega al computador una gramática ya formada que le hace posible construir el material para la enseñanza. Generalmente el resultado toma la forma de un juego, una actividad donde se resuelven problemas, o una simulación. Esto constituye un avance muy promisorio, especialmente porque tales programas tienden hacia la 'inteligencia artificial' si incluyen la habilidad para aprender de errores detectados por los estudiantes.

S-Endings. Este breve programa es un ejemplo de un programa sintético simple, y fue escrito por Tim Johns de la Universidad de Birmingham, Inglaterra. Puede ser usado en cualquier microcomputador equipado con 'Sinclair BASIC' tales como el ZX81, TS 1000 y hasta el Spectrum y TS 2068 con pequeñas modificaciones. Este programa está diseñado para dar la terminación 's' para cualquier verbo o sustantivo en Inglés. Por ejemplo

TABLE-TABLES (plural del sustantivo)

WAIT-WAITS (3ª persona del singular del verbo)

Lo interesante es que existen límites al 'conocimiento' del computador, y los estudiantes (trabajando como un grupo con el profesor) pueden ser desafiados a hacer dos cosas:

- i) descubrir las reglas pertenecientes al Inglés.
- ii) 'pillar' al computador. Por ejemplo en este programa, si usted escribe 'INFORMATION' dará 'INFORMATIONS', lo cual es incorrecto en Inglés. El plural debería ser 'pieces of information'.

Sin embargo, existen excepciones tales como LOCH/LOCHS y QUIZ/QUIZZES, donde el computador logra acertar).

Veamos las reglas de esta gramática sintética



Ejemplos:

CHURCH (reglas 1,2,4) = CHURCHES
 TABLE (reglas 1,4) = TABLES
 CRISIS (reglas 1,2,4,) = CRISES
 STAY (reglas 1,3,4) = STAYS

El Instituto Chileno-Británico de Concepción ha comenzado a experimentar con microcomputadores para ayudar en el aprendizaje del Inglés, y es un miembro de una investigación a escala mundial organizada por el Consejo Británico.

Me gustaría saber de otras instituciones en Chile que hayan trabajado con microcomputadores con miras de un intercambio de información e ideas.

Bibliografía

- BALTRA, A. **Computers in EFL: Present day possibilities and limitations** English Teaching Newsletter, Chile. 8/2 Julio 1984 pp.5-10
- Edmundson, E. **Aprenda Inglês com Animals**. 'Micro-Sistemas', Brasil III/30. Marzo, 1984, pp.14-15
- Edmundson, E. **Newsdesk: Or "How would you CALL that?"**, 'Cultura', Rio de Janeiro, 3/1. Julio 1984, pp.35-36
- Higgins, J. y Davies. **Computers, Language Learning and Teaching**. CILT Information Guide N° 22. London (1982)
- Higgins J. y John T. **Computers in Language Learning**. Addison Wesley/Collins 1984.
- John, T. **The uses of an analytic generator**. ELT Documents, N° 112. El Consejo Británico, Londres (1982) pp.96-105.
- Kenning, M.J. y Kenning, M.M. **An introduction to computer assisted language teaching**. OUP (1983)

```

1  REM S-ENDINGS: TIM JOHNS (1982)
100 PRINT AT 10,0;"PLEASE TYPE IN A VERB OR NOUN"; AT 12,0;"AND PRESS NEWLINE"

110 INPUT A$
115 CLS
120 LET B$ = A$
130 LET L = LEN B$
140 IF B$(L) < > "S" AND B$(L) < > "X" AND B$(L) < > "Z" AND B$(L - 1 TO ) < > "CH" AND B$(L - 1 TO ) < > "SH" OR B$ = "LOCH" THEN GOTO 180
150 IF B$ = "QUIZ" THEN LET B$ = B$ + "Z"
160 IF B$(L - 2 TO ) = "SIS" THEN LET B$ = B$( TO L - 2)
170 LET B$ = B$ + "E"
175 GOTO 190
180 IF B$(L) = "Y" AND B$(L - 1) < > "A" AND B$(L - 1) < > "E" AND B$(L - 1) < > "O" AND B$(L - 1) < > "U" THEN LET B$ = B$( TO L - 1) + "IE"
190 PRINT AT 11,14 - LEN A$;A$; AT 11,17;B$;"S"
200 RUN
  
```

OPENFILE

Cartas del lector



M BASIC

Señores Revista Microbyte:

Nuestras sinceras felicitaciones por su revista y al excelente material que publican.

Deseamos saber los cambios que hay que hacer al programa Monto Escrito, publicado en el N° 9, pues no resultó en nuestro Apple IIe que tenemos. Primero: el computador no acepta la instrucción ELSE, no la reconoce, y segundo: con algunas modificaciones que efectuamos al programa tampoco funciona.

Bueno, quisiéramos que publicaran el programa que adjuntamos, funciona en el Apple y sin fallas. Lo que hacen en el fondo es agregar un punto separando las cantidades en miles, cientos de miles, etc. y agregar ",00".

Reiterando nuestras felicitaciones a Uds. les saludan con toda atención.

José Guevara N.
Samuel Holbut V.
Casilla 34-V
Valparaíso

En realidad, sin conocer las modificaciones que hicieron ustedes al programa, es imposible descubrir por qué no les funcionó.

La instrucción ELSE (en castellano, "de lo contrario") la reconocen sólo algunas versiones del Basic y permite agregar en una sentencia IF-THEN la acción a tomar en caso de que la condición del IF no se cumpla.

Veamos el siguiente ejemplo:

```
10 INPUT A
20 IF A > 5 THEN PRINT
   "MAYOR" ELSE PRINT "ME-
   NOR"
```

Si el número que ingresamos es mayor que cinco, el computador escribirá MAYOR. De lo contrario, escribirá menor. Sencillo y muy útil para estructurar más la programación.

Cuando la versión del Basic, no acepta esa instrucción, entonces la solución está en repetir otra instrucción IF, pero para el caso contrario. Nuestro ejemplo quedaría:

```
10 INPUT A
20 IF A > 5 THEN PRINT
   "MAYOR"
30 IF A <= 5 THEN PRINT
   "MENOR"
```

En el programa del monto escrito, entonces para que funcione en el caso de ustedes, basta que modifiquen las líneas de instrucción que contienen un ELSE agregando una nueva sentencia IF-THEN que contemple el caso en que la primera condición no se cumple.

Respecto al programa que enviaron, lo encontramos un tanto demasiado complicado para un problema bastante simple. La rutina que presentamos, en cambio resuelve el mismo problema sin ninguna limitación de número decimal o largo del número. Esperamos la encuentren apropiada.



NOS ESCRIBEN

Miguel Pérez M. de La Florida nos ha escrito para consultar cómo se carga el programa "carre-ras" para el ZX-81, publicado en Mundo Electrónico en 1983, el que contiene rutinas en lenguaje de máquina.

En realidad, el programa viene con varios errores de imprenta y las explicaciones que lo acompañan son poco claras. Como éste fue un aporte de Sinclair Chile a dicha revista, quizás ellos mismos vuelvan a re-editar el programa, pero corregido.

Juan M. Collao C. de Arica, solicita información respecto a literatura disponible en relación a los más diversos temas de la computación, desde programación y análisis a telecomunicaciones y diseño de hardware.

Como sus necesidades son tan amplias, no nos es posible recomendarle algún texto sobre cada uno de esos temas. Nos llevaría un año completo investigar y seleccionar. ¿Algún lector tendrá preparada alguna lista bibliográfica general que sirva para este caso?

Nelson Jara, de Pto. Montt, felicita a Jorge Cea e insiste en que no nos olvidemos del 6502.

Jorge Gaete, de Linares, solicita información respecto a las direcciones de memoria en el Power 3000. Como no las tenemos, trasparamos su consulta a nuestros lectores. ¿Alguien tiene esa información?

```
10 INPUT "NUMERO :";A$
20 FOR I = 1 TO LEN (A$)
30 IF MID$ (A$,I,1) = "." THEN C$ = "." + RIGHT$
   (A$, LEN (A$) - I):A$ = LEFT$ (A$,I - 1)
40 NEXT I
50 IF C$ = "" THEN C$ = ",00"
60 IF LEN (C$) < 3 THEN C$ = C$ + "0"
70 IF LEN (A$) < 4 THEN GOTO 90
80 IF LEN (A$) > 3 THEN B$ = "." + RIGHT$ (A$,3)
   + B$:A$ = LEFT$ (A$, LEN (A$) - 3): GOTO 80
90 B$ = A$ + B$ + C$: PRINT B$
```

¿RUN
NUMERO :1234
1.234,00

NUMERO :75397.56
75.397,56

NUMERO :9754326432.7
9.754.326.432,70

LENGUAJES

Junto con saludarle, deseo hacerle llegar mis felicitaciones por el excelente nivel alcanzado en cada uno de los números de su revista, que logra educar, informar y entretener.

Quisiera sí, plantear mi inquietud respecto del punto siguiente: la gran mayoría de los artículos, informaciones y programas que se entregan a través de Microbyte, corresponden al lenguaje BASIC. Si bien es cierto que la inmensa mayoría de los microcomputadores actuales traen BASIC como lenguaje residente y es actualmente uno de los más conocidos, no es menos cierto que existen otros lenguajes de innegable importancia tanto en áreas específicas, como de aplicación general.

En las páginas de su revista pude leer un Editorial en que se

OPENFILE

Cartas del lector

plantea la inquietud por el desarrollo de la computación y la relación de ésta con la educación. ¿Porqué no entonces mayor información por los lenguajes LOGO o PILOT, que se encuentran orientados a acercar a los educandos a la computación?

Tampoco deja de ser cierto que en la actividad comercial, COBOL resulta fundamental y nadie que se desenvuelva en el área de trabajo computacional puede desconocerlo.

Resumiendo, creo que sería de mucho interés para muchos lectores, que se pudiera incluir en sus páginas comentarios e información sobre lenguajes como FORTH, PASCAL, los más arriba citados y otros de

uso frecuente.

Esperando que las presentes líneas tengan una buena acogida de parte suya y reiterando mis congratulaciones, se despidió atentamente de Ud.

C. Osorio M.
Viña del Mar

Efectivamente, y tal como Ud. lo plantea, el Basic no es ni el único ni el mejor lenguaje disponible, aunque sea por lejos el más difundido.

Le agradecemos su sugerencia (y recordatorio) y esperamos ponernos al día pronto publicando material relacionado con los otros lenguajes.



CENTRO DE FORMACION TECNICA

RECONOCIDO POR EL MINISTERIO DE EDUCACION SEGUN DECRETO EXENTO N° 33
CARRERAS CONDUCENTES A TITULO TECNICO DE EDUCACION SUPERIOR

CARRERAS

- Análisis de Sistemas
- Programación
- Contador general
- Auditoría
- Adm. de empresas
- Proyectos y finanzas
- Comercio exterior
- Instituciones financieras.

CAPACITACION

- Secretariado ejecutivo
- Lenguaje basic
- Lenguaje cobol
- Operación y programación de microcomputadores
- Dbase.

Matrícula \$ 950 Cuotas fijas sin reajuste de \$ 3.600 Currículum flexible

ALMIRANTE PASTENE 93^{al} 99 Providencia altura 1.300
Estación Metro Manuel Montt - Fono: 2512682-2512741

Señores de Microbyte:

Me es grato dirigirme a Uds. para comunicar mi agrado por vuestra revista, es de muy buen nivel y los temas allí tratados son de una claridad incomparable.

A pesar de que me cuesta un poquito conseguir su revista, soy un asiduo lector de ella.

La razón por la cual he decidido escribirles es que deseo intercambiar programas con usuarios del Texas Instruments TI 99/4A (en TI BASIC o TI BASIC EXTENDED indistintamente).

Muchas gracias, se despide atentamente.

Alvaro F. Amengual
Agustín Alvarez 412 cdad.
Mendoza
República Argentina
C.P. 5500

En el próximo número, Microbyte está de aniversario.

- Raíces de ecuaciones
- Interpolación de Newton
- Control de proyectos 1.ª parte:
El método del camino crítico (CPM)
- La computación en Chile:
Su evolución en términos de equipos y mercado
- Continúa curso CP/M: Comandos básicos y transistores y como siempre, programas para Atari, Sinclair, Commodore, Texas y otros.
No se lo pierda. Reserve su ejemplar.

EFFECTOS VISUALES TI 99-4A

En el programa publicado en el número anterior en la sección Texas, se escaparon algunos errores que impiden su ejecución. Para quienes no supieron cómo arreglarlo, las instrucciones que hay que modificar son:
310 CALL HCHAR (12, 16, 97)
320 FOR Y = 13 TO 24
330 X1 = Y + 3
370 NEXT Y
460 NEXT I

Con estas modificaciones, su programa deberá correr sin ningún problema. Para ustedes, nuestras disculpas.

Señores de la revista Microbyte

Lo felicito a Ud. y a todos los que trabajan en esta revista.

Yo estoy interesado en aprender programas para el Sinclair 1000.

En la revista número 8 he leído el openfile y estoy interesado en aprender a hacer programas. Yo tengo un Timex 1000 y espero que siga publicando programas.

Llamár al fono 258 (Vallenar).
Espero que me contesten.

Patricio Villar S.
(12 años)
Arturo Prat 2152
Vallenar

VIDEO INVERSO

Felicitaciones por su excelente revista, que a mí, y estoy seguro que a muchos más, nos ha acercado al fantástico mundo de la computación.

Referente a su nueva "Sección de consultas técnicas", la cual la encuentro maravillosa, mis dudas son:

1) ¿Cómo confeccionar un programa para SINCLAIR ZX-81 para visualizar en video-inverso, es decir, fondo negro y caracteres blancos?

2) ¿Podrían proporcionar una breve reseña o introducción a la programación en lenguaje de máquina (USR) para el ZX-81?

De antemano, muchas gra-

cias, esperando sus respuestas,

Erik Sacre M.
La Recova, Local 237
La Serena

Respecto a un curso para programar en lenguaje de máquina, ése era precisamente el tema de ocho largos capítulos de Programando el Z-80 escrito por Jorge Cea, desde el número dos al número nueve de Microbyte. Le será de gran utilidad.

Para colocar la pantalla en video-inverso en el ZX-81, el método más elegante es por hardware, existiendo un circuito que lo permite. Sin embargo, también con software es posible hacerlo, para lo cual le resultará útil la siguiente rutina:

```
200 LET X = PEEK 16396
210 LET Y = 16397
220 LET Z = X + 256 * Y
230 FOR J = 1 TO 33 * 24
240 LET K = Z + J
250 LET L = PEEK (K)
260 IF L = 118 THEN GOTO 300
270 LET L = L + 128
280 IF L > 255 THEN LET L = L - 256
290 POKE K, L
300 NEXT J
```

*D-FILE ver Cap. 26 en el manual.
Z es el comienzo del archivo de Display.
Recorre 24 líneas de 32 caracteres + Return.
Si es Return, no procesa Coloca bit 7 en 1 si hay 0 y viceversa.*

En Basic es bastante lento, pero funciona y es muy entretenido. En lenguaje de máquina lo hará mucho más rápido.

Factores

Continuando con nuestros programas educativos en esta sección, esta vez les presentaremos dos programas aptos para lucirse como expertos en matemáticas.

Como todos sabrán, es posible descomponer un número en factores y eso es algo que todos hacemos a diario, especialmente cuando compramos algo en docenas (huevos por ejemplo) y nos interesa averiguar el valor unitario.

El primer programa hace precisamente eso, descomponiendo un número en todos sus factores enteros posibles.

El segundo programa es bastante más complejo, obteniéndose sólo los factores primos de un determinado número, vale decir, no sólo encuentra todos los factores sino que antes de desplegarlos, verifica que ambos factores sean números primos. Aquellos interesados en la teoría de los números, encontrarán este segundo programa de mucha actualidad.

Para correr estos programas,

luego de tipearlos y echarlos a correr, aparece un signo de interrogación mediante el cual el

programa nos está preguntando el número que queremos descomponer.

```

10 REM FACTORES
20 FAST
30 INPUT A
40 PRINT A,
45 PRINT 1,
50 FOR B = 2 TO INT (A / 2)
60 IF A / B = INT (A / B) THEN PRINT B,
70 NEXT B
75 PRINT ".*"
80 RUN

```

```

RUN
?122
122      1      2
61      .
?323
323      1      17
19      .
?178
178      1      2
89      .
?76
76      1      2
4      19     38
.
?260
260      1      2
4      5      10
13     20     26
52     65     130

```

Programa 1

1 REM FACTORES PRIMOS	¿RUN
2 REM DESCOMPONE UN NUMERO EN SUS	?10
3 REM FACTORES, SIEMPRE QUE ESTOS	2*5
4 REM SEAN AMBOS NUMEROS PRIMOS	5*2
10 INPUT A	?121
20 FOR I = 1 TO INT (A / 2)	11*11
25 IF I > 3 THEN I = I + 1	?123
30 IF A / I = INT (A / I) THEN GOSUB 400	3*41
40 NEXT I	41*3
100 GOTO 10	?513
400 LET K = 0	?511
410 LET C = I	7*73
420 GOSUB 500	73*7
430 IF K = 0 THEN RETURN	?713
440 LET C = A / I	23*31
450 GOSUB 500	31*23
460 IF K = 2 THEN GOSUB 600	?77
470 RETURN	7*11
500 IF C = 1 OR C = 2 THEN GOTO 535	11*7
510 FOR N = 2 TO INT (C / 2)	?87
520 IF C / N = INT (C / N) THEN GOTO 550	3*29
530 NEXT N	29*3
535 LET K = K + 1	?101
550 RETURN	1*101
600 PRINT I;"*";C	?671
610 RETURN	11*61
	61*11

Programa 2



Aerocaza Submarino

José C. Periale Molina

Un programa simple y, sin embargo, muy entretenido es el que nos envía José Cristián para nuestros lectores.

En este juego estamos al mando de un sistema de defensa radiocontrolado, mediante el cual controlamos una escuadrilla de aviones encargados de defender las costas de los continuos asedios de submarinos que vienen a destruir el mundo de la superficie.

En la pantalla podemos observar el desplazamiento de nuestro avión y en las profundidades el submarino que se acerca. Para destruir al submarino invasor debemos dejar caer una carga de profundidad que acierte directamente en el centro del submarino.

No es fácil sin embargo. Las leyes de la inercia hacen que la bomba caiga con una inclinación de 45 grados, por lo que

hay que tener la mente clara y el pulso frío para disparar exactamente en el momento preciso. Por otro lado, los enemigos acuáticos no son tontos, navegando cada uno a diferente profundidad, haciendo más difícil aún hacer blanco.

En la línea 65, entre las comillas debe ir un carácter gráfico "Q" y dos caracteres gráficos "6".

```
10 LET J=0
20 FOR B=1 TO 5
24 CLS
30 PRINT AT 10,0;"-----"
40 LET Y=0
45 LET H=INT (RND*(18-11+1))+11
50 FOR X=29 TO 0 STEP -1
60 PRINT AT 7,X;"<"
65 PRINT AT H,(29-X);"  "
70 IF Y=0 AND INKEY$<>" " THEN LET Y=7
80 IF Y<>0 THEN GOTO 150
90 PRINT AT 7,X;" "
95 PRINT AT H,(29-X);"  "
100 NEXT X
130 NEXT B
135 PRINT AT 5,5;"PUNTAJE: ";J;" DE ";B-1
140 PAUSE 4E4
145 RUN
157 LET Y=Y+1
160 PRINT AT Y,X;". "
165 IF Y=H AND X=(29-X)+1 THEN GOTO 250
170 IF Y<H THEN GOTO 90
200 LET Y=0
210 GOTO 90
250 PRINT AT H,(29-X);"HAY"
251 PAUSE 30
252 LET J=J+1
253 FOR F=H TO 20
255 PRINT AT F,(29-X);"GLU";AT F,(29-X);"  "
260 NEXT F
270 GOTO 130
```

Conversión de base decimal a base cualquiera

Héctor A. Miranda Riquelme

Sin ánimo de hacer filosofía, quiero mencionar que uno de los aspectos más interesantes de la computación es que muchas veces nos lleva a preguntarnos el por qué de las cosas y a encontrar nuevas facetas a situaciones que nos parecen cotidianamente muy triviales.

Esto fue lo que me sucedió cuando aprendí y entendí la diferencia entre un número y... un número. La gran confusión al hablar de este tema está muy ligada al lenguaje, puesto que para nosotros decir EL NUMERO "11" es lo mismo que decir EL NUMERO ONCE (es decir once veces una unidad). Lo que sucede es que nosotros estamos acostumbrados a pensar "en decimal" y cuando vemos el número (aquí estoy abusando del lenguaje) 11, decimos, sin pensarlo dos veces, "once"; pero, un ser "no decimal" puede tener la noción de que el símbolo "11" represente al número "tres" o al número "nueve" o al número "diecisiete" o cualquier otra cantidad.

Todo es un problema de bases (matemáticamente hablando). Nuestra noción de número está basada en el 10. Es decir, cuando escribimos 236 y leemos "doscientos treinta y seis" y pensamos en doscientos treinta y seis veces una unidad, estamos haciendo (sin darnos cuenta) un proceso matemático en nuestra mente, que en símbolos se traduce:

$$2 * 10^2 + 3 * 10^1 + 6 * 10^0 = 200 + 30 + 6 = 236$$

donde el símbolo " \wedge " representa la elevación a potencia.

Por definición, toda cantidad elevada a la potencia cero, es igual a uno (la unidad).

Ahora bien, vemos que en la fórmula de arriba se repite un factor de ponderación, elevado a potencias consecutivamente decrecientes, y ese factor es nuestro querido diez. A este

diez (o más bien, a este factor) le llamamos "base", y definimos un conjunto numérico $S(X)$ en base X como:

$$S(X) = (\text{todos los números "n" pertenecientes a } N / n < X)$$

lo que significa que es el conjunto de todos los números "n" pertenecientes al conjunto de los números naturales que son menores que nuestra base X .

Por lo tanto, tenemos que el conjunto $S(10)$ es:

$$S(10) = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) \text{ y el conjunto } S(2) \text{ es:}$$

$$S(2) = (0, 1)$$

Este conjunto $S(X)$ nos provee de los "símbolos" necesarios para escribir cualquier número expresado en nuestra base X . Pero, ¿qué pasa con los conjuntos de base mayor que diez? En este caso ya se complica un poco la notación, más que nada por nuestras nociones intuitivas sobre los símbolos usados. Por ejemplo, el conjunto $S(16)$ estaría formado por:

$$S(16) = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$$

¿Cómo expresar los números del diez al quince sin confundir la notación? La respuesta es simple, se determina la siguiente equivalencia:

$$\begin{aligned} 10 &< \text{---} > A \\ 11 &< \text{---} > B \\ 12 &< \text{---} > C \\ 13 &< \text{---} > D \\ 14 &< \text{---} > E \\ 15 &< \text{---} > F \end{aligned}$$

Por lo tanto, nuestro conjunto (de símbolos) queda:

$$S(16) = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)$$

En computación son muy im-

portantes cuatro conjuntos numéricos: el decimal (en base diez), el binario (en base dos), el octal (en base ocho) y el hexadecimal (en base dieciséis). Hoy en día, el conjunto en base ocho se encuentra un poco en desuso; pero los otros tres tienen plena vigencia. Para la notación, entonces, usaremos el siguiente esquema: $N(X)$, donde N es el número y X es la base en que se encuentra expresado.

De esta forma, $11101(2)$ representa al $29(10)$, es decir:

$$1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 29$$

Y el $F3D4(16)$ es el $62420(10)$:

$$15 * 16^3 + 3 * 16^2 + 13 * 16^1 + 4 * 16^0 = 62420$$

Dominar estas conversiones implica un proceso mental de abstracción, con lo cual terminamos hablando de los "efemeros eficientes" al referirnos al número $FF00(16)$.

Para los que todavía nos cuesta esta conversión y hacer el cálculo "a manopla" nos resulta tedioso, existen unas calculadoras muy útiles, que vienen con las conversiones entre bases 2, 8, 10 y 16 en "firmware" (o software en estado sólido, programas en ROM). Y para los que no tenemos esas simpáticas máquinas también existe la posibilidad de ejercitar un poco nuestras artes de programadores y escribir un programita en nuestra calculadora programable.

Esto es lo que hice yo en mi CASIO FX-702P. En realidad, el programa no lo he ocupado mucho, pero la verdad es que me entretuve una barbaridad haciéndolo, sobre todo porque, dadas las limitaciones del BASIC de la CASIO debí simular algunas útiles instrucciones de otros BASIC's.

Este es un programa utilitario que servirá sobre todo a los pro-

gramadores más expertos que se "codean" todos los días con sus amigos binarios y hexadecimales. Lo que hace es transformar entre números decimales (base 10) y números en una base B cualquiera (B menor o igual que 16).

Las instrucciones para usarlo son sencillas: supongamos que queremos saber a qué número decimal corresponde el hexadecimal FF. Procedemos como sigue:

1. Se selecciona DEFM 2.
2. Se invoca el programa, el

que se presenta con el mensaje:

CON.: DEC < — > BASE B

3. Se presiona CONT (para continuar).
4. El programa pregunta BASE = ¿, y se le debe ingresar la base numérica hacia y desde la cual se desee hacer la conversión, en este caso 16.
5. Luego, nos pregunta nuestra opción:

1) 16 - > 10; 2) 10 - > 16

Presionamos el 1 (sin presionar EXE).

6. Nos pregunta:

NUM (16) =

y le respondemos: FF y terminamos con un "." (punto).

7. Al cabo de un par de segundos, nos contesta:

NUM (10) = 255

8. AL DARLE CONT, vuelve al menú del punto (4.).
9. Si se quiere cambiar de base, al estar en este menú se presiona el punto ".".

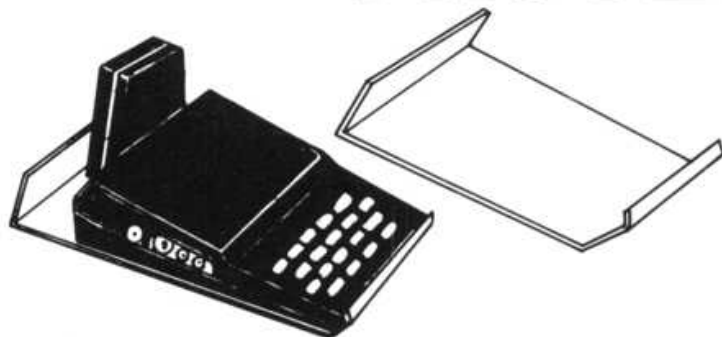
LISTADO

```

10 PRT "CONV.:DEC<-->BASE B":WAIT 0
15 INP "BASE=";B:B=INT B:D=10:IF B>16 THEN 15
17 IF B<2 THEN 15
20 S=0:PRT "1)";B;"->10;2)10->";B:GSB 500
30 IF KEY="" THEN 30
40 IF KEY="2" THEN 120
45 IF KEY="." THEN 15
50 IF KEY="1" THEN 30
60 S="0123456789ABCDEF":GSB 300:GOTO R
70 FOR K=M TO 0 STEP -1:FOR L=1 TO B:IF AS(K)*MID(L,1);NEXT L
80 IF L>B:PRT "ERROR":STOP:GOTO 20
90 S=S+(L-1)*B^(M-K):NEXT K
100 PRT S:STOP:GOTO 20
120 S="0123456789":D=B:B=10:GSB 300:GOTO (SGN (R-20)*110+20)
130FOR K=M TO 0 STEP -1:FOR L=1 TO 10:IF AS(K)*MID(L,1);NEXT L
135 IF L>10:PRT "ERROR":STOP :B=D:D=10:GOTO 20
140 A(K)=L-1:NEXT K:N=0
150 FOR I=0 TO M:N=N+A(I)*10^(M-I):NEXT I
155 T=RND(LN N/LN D,-1):S=S+"ABCDEF":F=N
160 FOR I=0 TO T:P=INT (F/D):G=F-P*D:F=P:A(I)=G
165 IF F<0:NEXT I
170 FOR K=I TO 0 STEP -1:H=A(K):PRT MID(H+1,1);:NEXT K
180 STOP :PRT " ":B=D:D=10:GOTO 20
300 PRT "NUM(";B;" )=";
305 FOR J=0 TO 19
310 CS=KEY:IF CS="" THEN 310
320 IF CS=".";PRT CS;:AS(J)=CS:NEXT J
330 M=J-1:PRT " ":PRT "NUM(";D;" )=";
340 IF J=0:PRT "ERROR":STOP :R=20:RET
350 R=70:RET
500 FOR I=0 TO 19:AS(I)="" :A(I)=0:NEXT I:RET

```

PROTEKTORTM



No más pérdidas de Programas o riesgos de quemar su expansión de memoria de 16 K en el ZX81 y Timex-Sinclair 1000. Adaptable para expansión de memoria de 64K Mantenga su computador y memoria firmemente unidos y no tema mover el computador.

Adquiéralo por \$ 960 en Microbyte

Merced 346 Of. E. Pedidos a provincias agregar \$ 100 para gastos de franquicia

Cuadrados mágicos

Si hay quien no haya llenado un cuadrado mágico en su vida, ha perdido gran parte de ella. Es una diversión muy antigua y que ha llamado la atención de todos aquellos que gustan de las matemáticas. Ejemplos de esto se encuentran en la literatura china y en los escritos de Platón. También los egipcios los utilizaban para los horóscopos.

Un cuadrado mágico es aquel en el cual al sumar sus líneas, columnas y diagonales principales se obtiene un mismo resultado. En la figura 1, vemos un cuadrado mágico en su forma más simple. En éste, la suma de cada columna, fila y diagonal es 15.

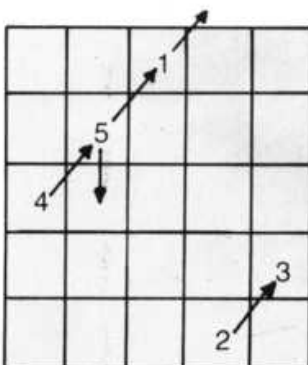
8	1	6
3	5	7
4	9	2

Figura 1

El método para generar este tipo de cuadrados es muy simple para el caso de número impar de filas y columnas, sea 3x3, 5x5, 7x7, etc.

En la figura 2, podemos apreciar cómo se va construyendo un cuadrado de 5x5. Las reglas para esto son las siguientes:

- En la casilla central de la primera fila, partimos colocando un uno.
- A continuación, debemos movernos siempre hacia arriba y hacia la derecha.
- En caso de no haber más filas hacia arriba o columnas hacia la derecha, se debe continuar en el lado opuesto. El número dos, lo ponemos entonces en la última fila y en la columna que corresponde.
- El tres va luego en la celda de arriba a la derecha y para poner el cuatro, se vuelve a la primera columna en la fila de arriba.
- El cinco continúa en la fila y columna siguiente, pero para continuar, la próxima celda ya está ocu-



a

		1	8	
	5	7		
4	6			
10				3
11			2	9

b

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

c

Figura 2

Naturalmente, y esa es una de las gracias de los cuadrados mágicos, no es necesario comenzar siempre del número 1 y tampoco el incremento debe ser obligatoriamente una unidad. En la figura 3 vemos dos ejemplos de cuadrados mágicos de esa naturaleza.

11	4	9
6	8	10
7	12	5

20	6	16
10	14	18
12	22	8

Figura 3

También es factible generar cuadrados mágicos con un número par de columnas y filas (4x4, 6x6, 8x8), pero este tipo lo veremos en el próximo número.

El programa que presentamos permite generar los cuadrados del lado impar, no importa el núme-

ro de filas y columnas que sea. El POKE en la línea 5, sirve para que al tabular, el computador no se salte los diez espacios que deja normalmente entre campo y campo, con el objeto de facilitar la impresión en pantalla de cuadrados de hasta 9 filas.

Al ejecutarlo, el programa parte preguntando por el número de filas del cuadrado y luego por el

número inicial. Modifique usted el programa para que además pregunte por el intervalo entre los números para generar cuadrados aún más interesantes.

Use este programa para generar todos los cuadrados mágicos que quiera y luego úselos para dar algunas pistas a otra persona para que éste lo complete. Es un sano y estimulante ejercicio.

```

1 POKE 201,4
5 OPEN #1,4,0,"K"
6 GRAPHICS 0:PRINT "CUANTAS COLUMNAS (3-7)"
7 INPUT COL
8 PRINT "NUMERO INICIAL (1-9)"
9 INPUT D
10 DIM A(COL,COL)
15 E=D
20 FOR I=1 TO COL
30 FOR J=1 TO COL
40 A(I,J)=0
45 V=V+E
46 E=E+1
50 NEXT J
60 NEXT I
70 GOSUB 1000
80 A(1,INT(COL/2+1))=D
90 GOSUB 1000
100 L=1:M=INT(COL/2+1):N=D
105 IF S=V THEN END
106 L1=L:M1=M
110 IF L=1 THEN L=COL+1
120 IF M=COL THEN M=0
130 L=L-1:M=M+1
140 IF A(L,M)(>)0 THEN L=L1+1:M=M1
150 A(L,M)=N+1:N=N+1
160 GOSUB 1000
170 GOTO 105
999 END
1000 GRAPHICS 0
1002 S=0
1005 X=5
1010 FOR I=1 TO COL
1015 POSITION 5,X
1020 FOR J=1 TO COL
1030 PRINT A(I,J),
1035 S=S+A(I,J)
1040 NEXT J
1050 X=X+1:PRINT :PRINT
1060 NEXT I
1065 REM GET#1,Z
1070 RETURN

```

SMITH-CORONA®

La respuesta americana en impresoras.

DISTRIBUIDORES TUCAN INGENIERIA AREA SMITH-CORONA a lo largo del país.

ASC • Austria 2041

COELSA S.A. • Av. Vicuña Mackenna 1705

- Andrés de Fuenzalida 79 y su Red de Distribuidores a lo largo del país.

COMPUTERMARKET • Pueblo del Inglés Loc. 66 - San Antonio 87

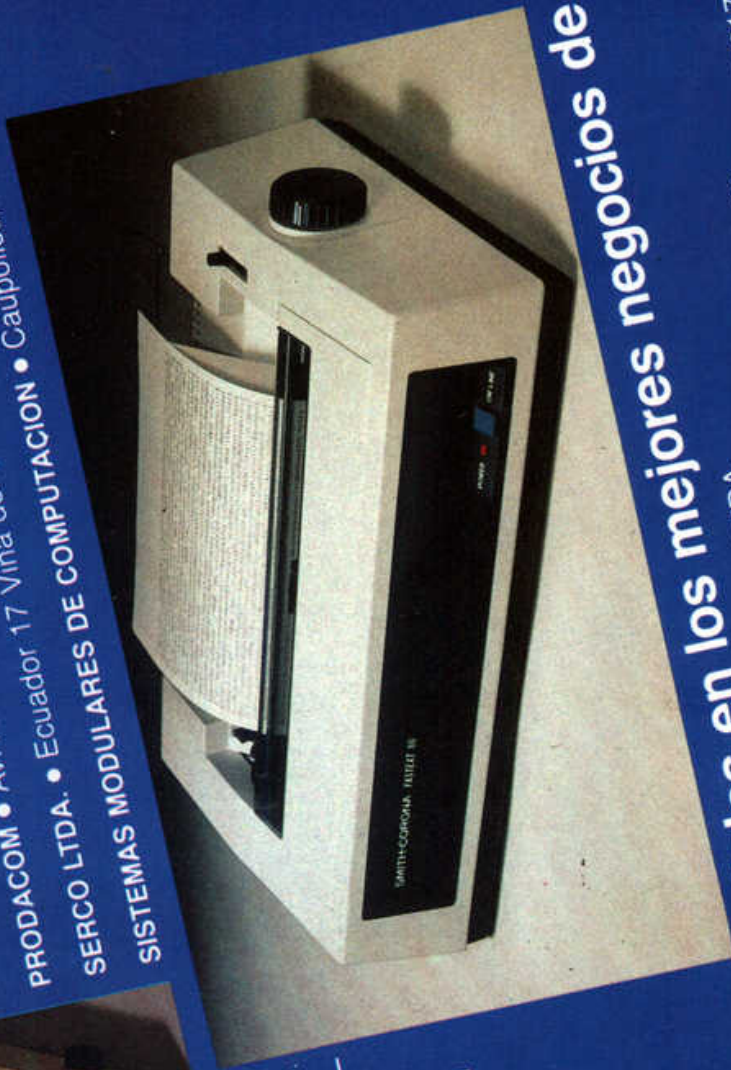
CPC Ltda. • Matías Cousiño 82 Of. 403

INGENIERIA DE SISTEMAS • Av. Gral. Bulnes 107 Of. 67

PRODACOM • Av. Andrés Bello 1581

SERCO LTDA. • Ecuador 17 Viña del Mar

SISTEMAS MODULARES DE COMPUTACION • Caupolicán 567 Of. 402 - Concepción



CARACTERISTICAS	ANCHO		VELOC.	BIDIREC.
	NOR. COL.	MAX. COLOR		
F - 80	80	132	80	S
D - 200	80	132	160	S
D - 300	132	233	180	S
TP - II	100	-	12	N

Exíjalas en los mejores negocios de computación.

TUCAN INGENIERIA Y CIA. LTDA. Chile.
Representante oficial para 2125, 494085 Telex: 240177 VOAG-CL.
Luis Thayer Ojeda 742453, Correo Central, Santiago.
Fonos: Casilla 1261.

Auspicia revista MICROBYTE

El evento computacional de 1985



SOFTTEL'85

**CONVENCION INFORMATICA SOFTEL'85 JUNIO 27 - JULIO 3, 1985
HOTEL HOLIDAY INN CROWNE PLAZA**

LA INFORMACION A NIVEL ORGANIZACIONAL

• EFECTOS • ALTERNATIVAS • TECNOLOGIA

- Informática y Gestión Administrativa.
- Muestra de soluciones reales y prácticas.
- Tendencias y perspectivas.
- Empresas proveedoras de la más reciente tecnología computacional, software y telecomunicaciones.

ORGANIZA: FACULTAD DE CIENCIAS ECONOMICAS Y ADMINISTRATIVAS