ANO II
NÚMERO 15
CR$ 1.500,00

interface

Hardware - Software - Computadores - Mini - Micro - Teleprocessamento

MANAUS
SANTARÉM
RIO BRANCO
ALTAMIRA
BOA VISTA
MACAPÁ
PORTO VELHO
CR$ 1.600,00



JOGUE DEZ DEZENAS NA LOTO

TRUQUES PARA ARRUMAR OS ALGORITMOS

TODOS FABRICAM E NINGUÉM CONSERTA

DANIEL REMI

Preço de lançamento
Cr$ 649.850,00
Preço sujeito à alteração.



Em breve nas lojas

Novo TK 2

Pelo preço dele você só compra a I

Até hoje você tinha só duas alternativas: ou pagava caro um computador de alta performance ou se conformava com um equipamento de preço e possibilidades limitadas.

Agora a Microdigital lança o TK 2000 color, um micro de alta performance pela metade do preço do seu concorrente mais próximo.

Veja: ele tem 64 kbytes de memória RAM e 16 kbytes de memória ROM, trabalha com alta resolução gráfica à cores podendo ser ligado ao seu TV colorido ou P&B, diskette de 5 1/4", impressora (ele já vem com interface), gravador, cartuchos gravados, joystick.

E o que é muito importante: tem excelentes software disponível,



2000 color

metade do concorrente mais próximo.

inclusive planilha eletrônica, editor de textos, controle de estoques, jogos animados em cores e muito mais. Conheça o novo TK 2000 color nas lojas especializadas e magazines de todo o país.

E leve um micro avançado, de alta performance, praticamente pela metade do preço.

Se na sua cidade você não encontrar o computador TK 2000, ligue gratuitamente para (011) – 800 255.8583 e teremos o prazer de informar-lhe sobre o revendedor Microdigital mais próximo.

MICRODIGITAL

Microdigital Eletrônica Ltda.
Caixa Postal - 54121 - CEP 01000 - São Paulo - S.P.

À venda nas boas casas do ramo, lojas especializadas de fotovídeo-som e grandes magazines.

Cobra adere ao álcool.

O leitor sabe da importância que a indústria de cana de açúcar adquiriu para o Brasil, principalmente depois que o álcool passou a ser nossa principal fonte alternativa de energia.

Isso levou o Instituto de Açúcar e do Álcool a promover um rápido processo de modernização de toda a indústria canavieira. E os computadores Cobra entraram em ação.

Uma experiência pioneira está sendo realizada na Cooperativa Regional dos Produtores de Açúcar de Alagoas, onde trabalham um Cobra 530 e dois Cobra 305. A primeira tarefa destes computadores é determinar o teor de sacarose de toda a cana que entra nas usinas. Em vez de se pagar ao produtor pelo peso da cana, como se fazia antes, agora se paga pelo teor de sacarose, ou seja, pela quantidade de açúcar e álcool que a cana pode produzir.

Além de servir de incentivo para que o produtor melhore cada vez mais a qualidade de sua cana, esse procedimento ainda permite que tão logo a cana entre nas usinas, já se possa fazer um planejamento perfeito de produção. Os computadores Cobra fornecem dados que determinam, com segurança, a quantidade de açúcar e álcool que cada partida de cana vai produzir.

Mas não pense você que é só na área de produção que os computadores Cobra dão conta do recado. Eles ainda se encarregam da área administrativa, cuidando do faturamento, das contas a pagar e mantendo, com rigorosa precisão, as contas correntes dos cooperados. Como você vê, com computadores Cobra a vida na usina fica muito mais doce.



Nota do editor

NAILA GLÓRIA O. CORTES

É COMO O SOL DE VERÃO

Interface entra em seu 15.º número com grandes novidades: a reformulação editorial e gráfica e a mudança de sua sede, agora na **rua Barão do Flamengo, 22/703.** As duas iniciativas atendem a pedidos de nossos leitores e colaboradores, que nos escreviam há meses, reclamando de nosso alheiamento em Jacarepaguá (vocês são de difícil acesso) e do sistema de apresentação dos trabalhos publicados (vocês estão mais tecnocráticos do que uma balzaquiana que já esqueceu como é bom um chopinho na Barra). E nós atendemos nossos leitores e colaboradores: estamos mais acessíveis e prontos para qualquer programa de computação. Podem vir quentes que nós estamos fervendo. Mantemos o nível de informação tecnológica, mas preparamos a balzaquiana para poder entrar em qualquer discoteca.

Para começo de conversa, o curso Basic já chegou ao ponto de poder levar aqueles que começaram conosco a programar um jogo da Loto. O Caderno Especial, destinado aos assuntos propriamente didáticos, prossegue com seu Curso Z-80, já na 14.ª lição, trata da linguagem Logo e do sistema Filia.

Mantemos o objetivo de fazer os usuários de micros capazes de progressos constantes. Temos um artigo contando como você pode chegar a ser um grande cozinheiro de algoritmos, capaz de navegar nos mares nada estáveis onde a racionalidade precisa ser temperada com uma grande dose de intuição.

Também continuamos a preparar os terrenos para os novos tempos que estão chegando: todos já sabem que o 68000, o poderoso microcomputador, está chegando para ficar. Nosso leitor já está sabendo disso há muito tempo e agora recebe mais informações de como se preparar para o acontecimento.

Nossas dicas de hardware e de software continuam atuais, como sempre. Temos uma série imensa de pequenas novidades, de novos produtos, de iniciação a novos processos, que sempre manteremos como nosso objetivo de grandes apaixonados pela informática e de órgão primordial de sua divulgação.

E temos também as nossas novas preocupações, todas com o leitor, com o usuário da informática: apresentamos um questionário suscinto para você preencher e nos enviar. Contamos com isso para saber que serviços você espera de nós.

Mas temos também uma noticiazinha ruim: o custo do papel nos força a aumentar os preços do balcão, aquela paginazinha tão baratinha que fazia de cada leitor de Interface um grande anunciante perante todos os usuários de micros. Essa presença é ainda garantida, mas nossa tabela ficou um pouco mais salgada. Não mais do que o necessário para ela poder continuar existindo.

Não somos mornos: nosso interesse pela informática não nos permite ficar em cima do muro. Também não somos frios: mataríamos os assuntos que sempre estudamos. Esta é uma revista quente. Muito quente. Como o verão de 84.



Diretores responsáveis:
Nalla Glória O. Côrtes
Cesar da Costa

Editores científicos:
Cesar da Costa
Nalla Glória O. Côrtes

Editor de texto:
Jaime Collier Coell

Editor de arte:
Daniel Remi van Paemel

Assistente:
Nilton Luiz Côrtes

Arte-final:
Jorge Gatto
Sandra Maria P. dos Santos

Revisão:
Nelma Bernadete Osório

Diretor Comercial:
José Augusto R. Pereira

Publicidade:
Rio:
Rua Barão de Flamengo, 22, conjunto 703
Telefone: (021) 285-5047
Roberto Passeri
José Augusto Nunes Rodrigues
São Paulo:
Avenida Paulista 1159, conjunto 801
Márcia Padovan de Moraes
José Caetano dos Santos

Assinaturas:

Gerente:

Rio de Janeiro
Edison Teixeira Pedroso
Telefone: (021) 285-5047

São Paulo:
José Silva Santos
Tel.: (011) 284-8384

Preços:
Brasil: Cr$ 10.000,00 (1 ano)
Cr$ 20.000,00 (2 anos)
Exterior: US$ 50 (1 ano)
US$ 100 (2 anos)

Distribuição:

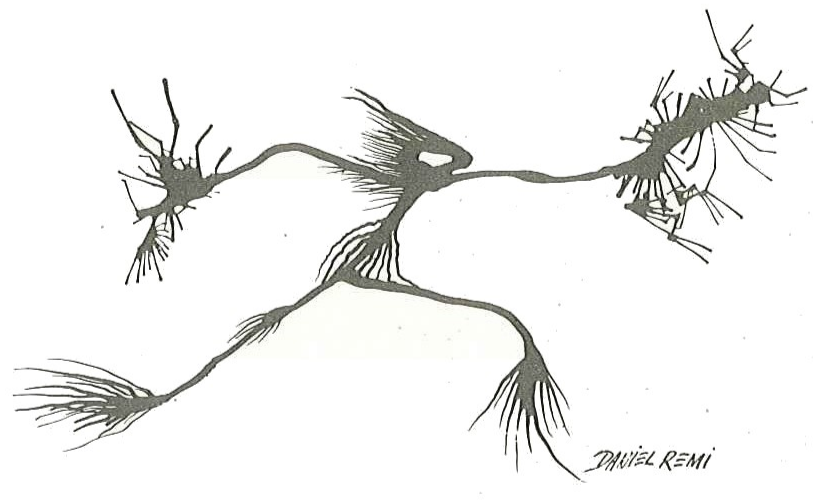
Fernando Chinaglia
RJ:
rua Teodoro da Silva, 907
Tel.: (021) 268-9112
SP:
Av. Casper Líbero, 152

INTERFACE

é uma publicação de propriedade da PRODIGT - Processamento, Tecnologia e Comunicação Ltda. Redação, Administração e Publicidade: Rua Barão do Flamengo, 22 - Conj. 703 - Flamengo - CEP. 22220 - Rio de Janeiro - Tel.: (021) 285-5047; Av. Paulista, 1159 - Conjunto 801 - CEP. 01311 - São Paulo - Tel.: (011) 284-8384.
1984 - PRODIGT - Processamento, Tecnologia e Comunicação Ltda. Todos os direitos reservados; proíbe-se a reprodução parcial ou total dos textos e ilustrações desta publicação, assim como traduções e adaptações, só podendo ser reproduzidos com fins comerciais, mediante autorização prévia. Os artigos assinados são de responsabilidade do autor, as opiniões emitidas não são necessariamente coincidentes com as da revista.

Interface

Algoritmo o primeiro passo para solucionar um problema em computação



Página 8

CALENDÁRIO

Página 12

Usuário sem socorro na hora da manutenção

Página 14

INFORME

Página 18

NOVOS PRODUTOS

Página 20

68000 - o poderoso microprocessador de 16 bits

Página 21

DESENVOLVIMENTO DE SOFTWARE PARA SISTEMAS BASEADOS EM MICROPROCESSADORES

Página 24

RESERVANDO ESPAÇO PARA ROTINAS EM CÓDIGO DE MÁQUINAS

Página 26

CARTAS

Página 27

CADERNO ESPECIAL

LINGUAGEM LOGO: A FERRAMENTA DE ENSINO POR COMPUTADOR

Página 40

EM BUSCA DE DEZ DEZENAS PARA A LOTO

Página 29

FILIA-UM SISTEMA DE PROGRAMAÇÃO

Página 32

O Z-80 já chega à 14.ª lição.

Página 38

TCI - a interface que testa circuitos integrados

Página 45

BALCÃO

A partir do número 16 o anúncio publicado no Balcão, com três linhas de 32 toques cada, passará a custar Cr$ 4.000,00. A linha adicional custará Cr$ 1.200,00. As reservas podem ser feitas em nossos escritórios ou por carta mediante o envio de cheque ou vale postal no valor correspondente.

Página 55

DIFERENÇAS ENTRE O BASIC E O TRS-80, APPLE E SINCLAIR

Página 56

LIVROS

Página 58

PROGRAMAS DE LAZER

Página 64

GLOSSÁRIO

Página 67

ERRATA

Página 68

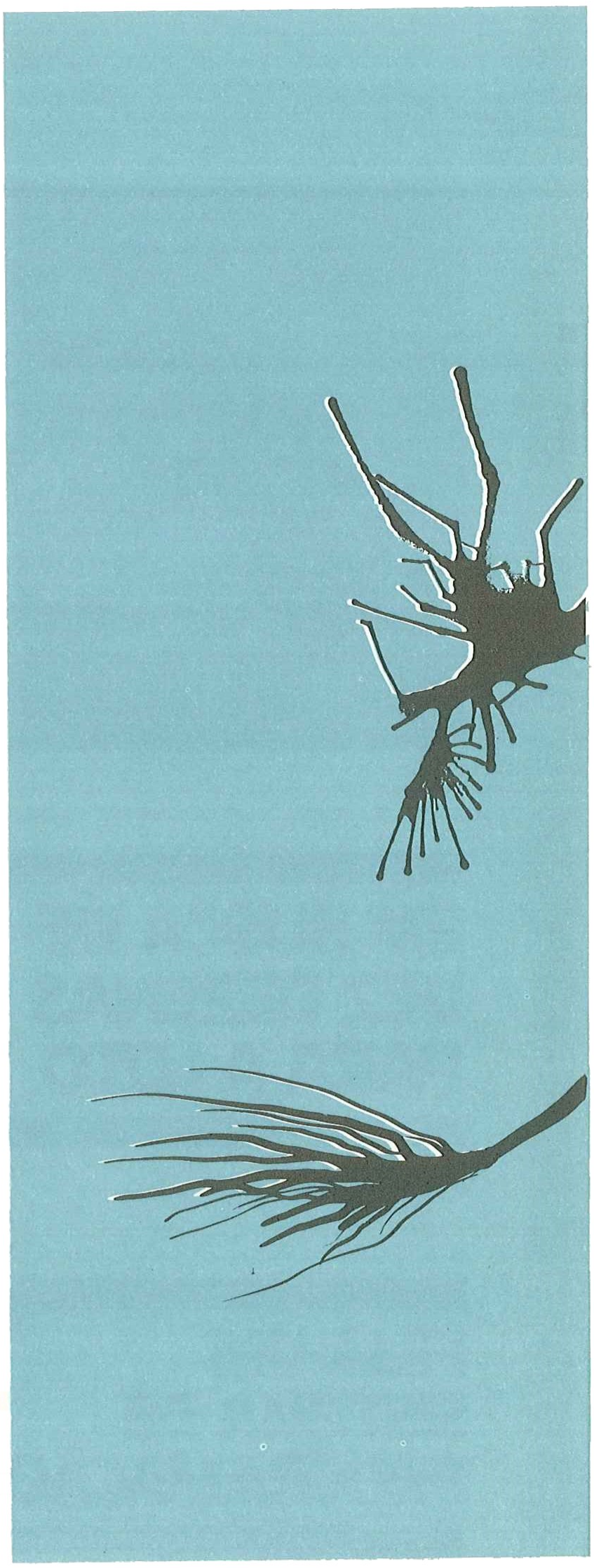


INTERFACE

PROCESSOS

MAX STEPHANO

Algoritmo o primeiro passo para solucionar um problema em computação

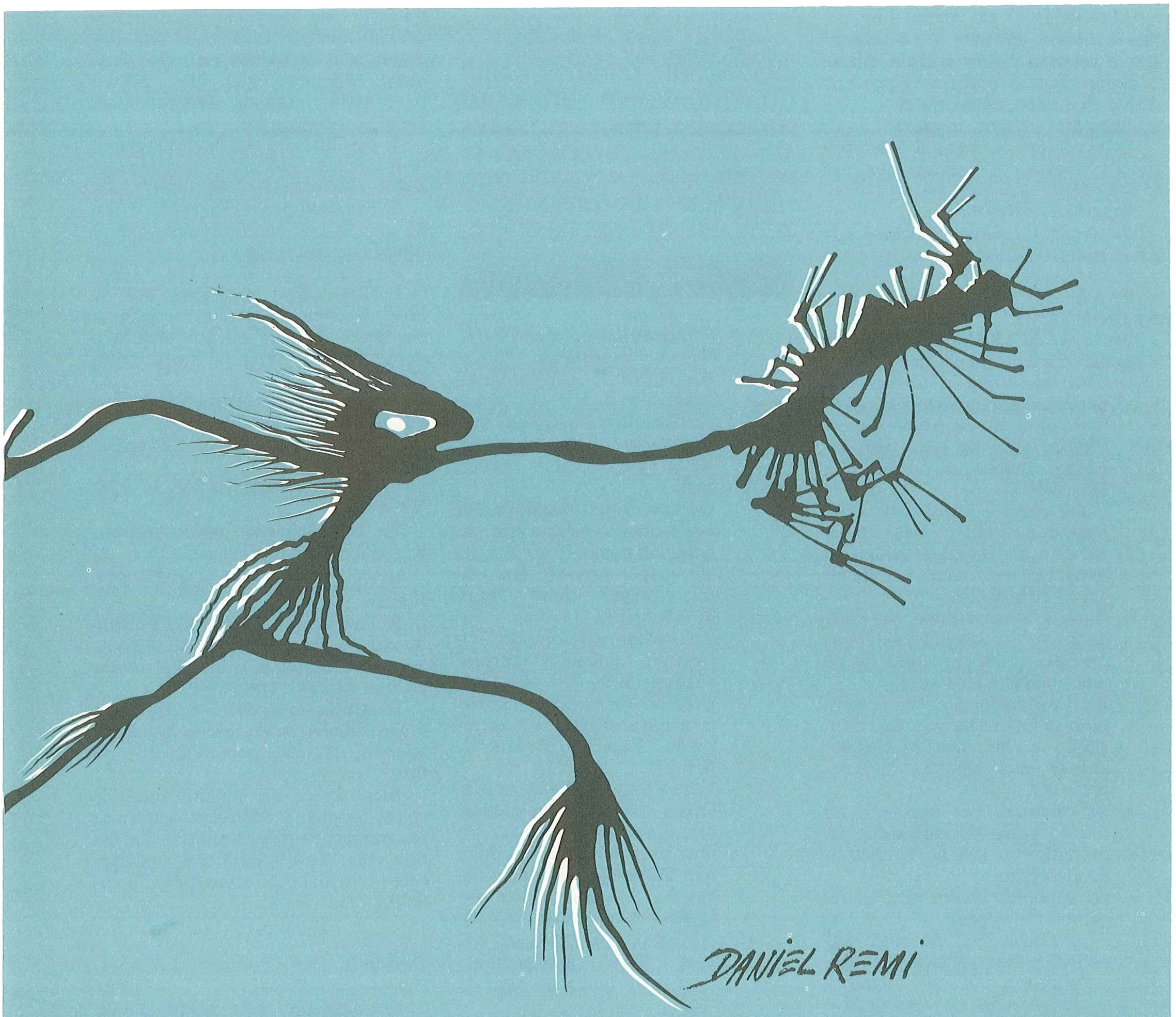


Durante séculos foi um lugar comum a crença na possibilidade de resolver certas questões matemáticas por meio de procedimentos mecânicos de cálculo — os algoritmos. Enquanto os matemáticos discutiam o problema da decisão e a impossibilidade de um procedimento universal para a teoria dos números, o computador trouxe essa problemática para o terreno operacional. E algoritmo, o primeiro passo para solucionar qualquer problema no computador, passou a ter o significado de qualquer processo computacional mecânico ou repetitivo. Max Stephano dá as dicas de como trabalhar com o altoritmo e saber, de ouvido afinado, quando a técnica precisa ser suplementada com a intuição de um bom cozinheiro.

É importante fazer uma análise histórica do que seja *algoritmo* pois sua noção é a base de toda programação de computadores.

A palavra não aparece no *Webster's New World Dictionary* antes de 1957, sendo encontrada na forma *algorism*, com o significado de fazer cálculos aritméticos com algarismos arábicos. A palavra surgiu no Ocidente no século XI, quando foram traduzidos para o latim os livros de Abu Mohammed ibn Musa al-Khowarizmi (século IX). Ela vem diretamente da versão latina intitulada *Liber Algorismi de numero Indorum*, livro contendo uma teoria do cálculo numérico com algarismos indianos. Outra obra de al-Khowarizmi tem o título de *Al-Gabr w'almuqualabala*, de onde vem a palavra *álgebra*. *Al-Gabr* quer dizer *transposição*, isto é, a passagem de um elemento da equação da esquerda para a direita do sinal de igual ou vice-versa, mudando o sinal. *Muqualabala* quer dizer *redução*.

A palavra *algoritmo* nada tem a ver com o *al-Gabr* de al-Khowarismi, mas sim com a obra conhecida apenas na versão latina *Liber Algorismi de numero Indorum*. O nome inteiro do matemático, Abu Ja'far Mohammed ibn Musa al-Khowarizmi, que quer dizer pai de Ja'Far, filho de Moisés, natural de Khowarizmi (a atual cidade de Khiva na Rússia) também nada tem com *algoritmo*, que vem diretamente de sua última parte, a cidade em que nasceu. *Natural de Khowarizmi* (al-Khowarizmi) se pronuncia, em portu-



guês, *al-Huuarizmi*, que é de onde surgem *algorismo e algarismo*. *Algorismus* e *algorithmus* são duas palavras que aparecem praticamente juntas, no latim medieval, após a tradução dos livros do matemático persa. Mas *algoritmo* tem influência do grego *arithmós* (número) de onde vem a palavra aritmética.

A expressão latina *algorithmus infinitesimalis* foi usada para indicar meios de cálculo com quantidades infinitamente pequenas, desenvolvidas por Leibnitz (1646-1716). No dicionário alemão Vollstandiges Mathematisches Lexicon (Leipzig, 1747), há a seguinte definição da *algorithmus*: "Sob esta designação estão combinadas as noções dos quatro tipos de cálculo aritmético, chamados adição, multiplicação, subtração e divisão".

A partir da segunda metade do nosso século a palavra *algoritmo* passou a dizer respeito a disposições especiais que se fazem com elementos matemáticos com o objetivo de efetuar cálculos, tais como o meio de fazer a divisão, o crivo de Eratóstenes para chegar a números primos, o algoritmo das divisões sucessivas para obter o máximo divisor comum de dois números dados, e o dispositivo de Briot e Rufini para divisão de um polinômio por binômio do primeiro grau. O *Algoritmo de Euclides* é um método para achar o máximo divisor comum de dois números (ver a obra *Elementos* no livro 7, proposições 1 e 21) que apresentamos abaixo:

ALGORITMO M: Dado dois números inteiros positivos M e N, encontrar o máximo divisor comum, isto é, o maior número positivo inteiro que divide ao mesmo tempo M e N.

Passo M1 – (Achar o resto)

Divida M por N, e que R seja o resto da divisão. (Teremos que ter $O \leqslant R < N$).

Passo M2 – (Resto é zero)

Se R = O, termina o algoritmo e N é a resposta.

Passo M3 – (Troca)

Faça M ← N, N ← R, e retorne para o passo M1. (**).

Este tipo de apresentação obedece a algumas regras bastante simples:

- Cada algoritmo é identificado por uma letra, no caso a letra M.
- Cada passo do algoritmo é identificado além da letra, por um número seqüencial, no caso M1, M2 e M3.
- Cada passo começa com uma frase que resume a idéia principal da ação. Esta frase pode aparecer também dentro do fluxograma do algoritmo.
- Após a frase resumo, há uma descrição em palavras e símbolos da ação a ser executada ou de uma decisão a ser tomada.
- Há, ocasionalmente, um comentário entre parênteses, que é incluído como uma explicação, apenas para auxiliar na compreensão.
- Os asteriscos que aparecem no fim do passo M3 são usados para indicar o fim do enunciado do algoritmo.
- Os símbolos usados têm as seguintes finalidades:
  - A flecha "←" no passo M3 indica uma operação de substituição. "M N" significa que o valor da variável M é substituído pelo valor atual da variável N.
  - A flecha é usada para distingüir a operação de substituição da relação de igualdade; não dizemos "Faça M = N", mas podemos testar "É verdade que M = N". O sinal "= " indica uma condição que pode ser testada, enquanto o sinal "←" indica uma ação que pode ser executada. Em geral, a expressão "Variável ← Fórmula" significa que a fórmula é efetuada usando os valores correntes das variáveis que nela aparecem, e o resultado substitui o valor antigo da variável que aparece à esquerda da flecha.

Aqui você tem a oportunidade para o "jeitinho": o significado moderno de algoritmo é bem similar ao de receita, processo, método, técnica, procedimento, rotina. Mas é verdade que não é isso tudo junto: às vezes é um, às vezes é outro.

  - A ordem das ações no Passo M3 é importante: "Faça M ← N, N ← R" é diferente de "Faça N ← R, M ← N", pois o valor anterior da variável N é perdido antes que possa ser usado para substituir o valor de M.
  - Quando diversas variáveis têm que assumir o mesmo valor de outra, tal como "Faça M ← R, N ← R", pode-se usar outra notação como "Faça M ← N ← R".
  - Para trocar os valores de duas variáveis, podemos escrever "Troque M ↔ N". Esta ação pode ser especificada usando uma nova variável T e escrevendo "Faça T ← M, M ← N, N ← T".
  - Falta a notação usada para variáveis subscritas ou indexadas, que são elementos de uma lista ordenada. Suponha que tivéssemos "N" quantidades, $V_1, V_2, V_3, \ldots, Vn$. Ao invés de escrevermos "V_j" para o j-ésimo elemento escrevemos "V(j)". Similarmente, "A(I,J)" é usado no lugar de "Aij".
  - Algumas vezes as variáveis recebem nome com várias letras (usualmente em maiúsculas). Por exemplo, HORA pode ser o nome de uma variável usada para guardar o valor da HORA num determinado passo de um algoritmo. A variável primo (K) pode ser usada para guardar o K-ésimo número primo.

O melhor modo de aprender o que é um algoritmo é executando um. Este é o método mais prático e menos doloroso. Qualquer outra abordagem é geralmente mal-sucedida.

Portanto, vamos trabalhar no algoritmo M. É importante neste momento se municiar com lápis e papel para acompanhar, calculando cada passo do algoritmo. (Note que todas as operações no algoritmo M são operações de número inteiro, ou seja, são desprezadas as frações e as casas decimais). Supondo que M = 238 e N = 1088, estamos prontos para começar, no Passo M1.

1. Dividindo M por N, neste caso é bem simples, desde que o cociente seja zero e o resto seja 238. Então, R ← 238.

2. Passando para o Passo M2, e desde que R é diferente de zero, não é tomada nenhuma ação.

3. No Passo M3, M ← 1088, N ← 238. Aqui percebe-se que se M < N como no início, o cociente no Passo M1 será sempre zero, e o algoritmo vai fazer a troca entre M e N.

4. De volta ao Passo M1 temos que (1088/238) = 4 e R ← 136.

5. Novamente M2 não é aplicável.

6. Em M3 fazemos M ← 238, N ← 136.

7. No próximo ciclo fazemos R ← 102, M ← 136, N ← 102.

8. Na seguinte, R ← 34 e M ← 102, N ← 34.

9. Finalmente, quando 102 é dividido por 34, R ← O e no Passo M2 o algoritmo termina. E o máximo divisor comum de 238 e 1088 é 34.

O significado moderno de algoritmo é bem similar ao de receita, processo, método, técnica, procedimento, rotina.

Além de ser um conjunto finito de regras ao qual é dado uma seqüência nas operações para a solução de um tipo específico de problema, um algoritmo tem cinco características importantes:

Finidade

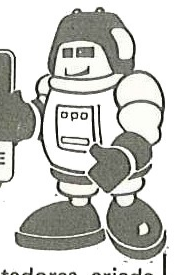
Um algoritmo deve sempre terminar após ser executado um número determinado de vezes. O algoritmo M satisfaz esta condição porque no Passo M1 o valor de R é menor que N. Assim, se R é diferente de O, o valor de N diminui cada vez que o Passo M1 é executado. Assim, o Passo M1 é executado somente um determinado número de vezes para qualquer que seja o valor inicial de N. Observe que o número de vezes pode tornar-se muito grande; certos valores escolhidos para N e M farão com que o Passo M1 seja executado milhares de vezes.

Um procedimento que tem todas as características de um algoritmo, menos a capacidade de finidade, pode ser chamado de *método computacional*. Além do seu algoritmo para o máximo divisor comum de dois inteiros, Euclides também desenvolveu

NÃO SE ARRISQUE

CIBERNE

O SOFTWARE QUE VOCÊ MERECE



CIBERNE é a mais avançada e sofisticada linha de sotfware para microcomputadores, criada para livrar você dos inconvenientes de produções amadoras ou de origem duvidosa. Com CIBERNE você terá uma diversificada linha de programas novos, lançados periodicamente em pacotes econômicos. Com gravação profissional e em embalagem inviolável, CIBERNE oferece a você garantia total em qualquer lugar do Brasil.

PROGRAMAS EM FITA PARA TK-82, 83, 85, CP-200 E COMPATÍVEIS

Com o exclusivo FLASH-SISTEM que permite um carregamento 6 vezes mais rápido que o normal, sem qualquer modificação no equipamento.

JOGOS

Sem qualquer acréscimo no preço você encontra em cada fita 5 excitantes jogos para seu lazer e entretenimento. Use joystick ou teclado.

BICHOS & CIA.	PATRULHA GALÁCTICA	COMBATE	AVENTURA & MISTÉRIO
■ CASCA, A COBRA	■ NAVE MÃE	■ NIGHT GUNNER	■ PIRÂMIDE INCA
■ CRAZY KONG	■ FUNGOS MUTANTES	■ ALERTA VERMELHO	■ SABOTAGEM
■ CENTOPÉIA	■ GALÁTICA	■ POLARIS	■ O AVENTUREIRO
■ FROGGER	■ SOS VEGA III	■ DUELO	■ MAZOGS
■ SUCURI	■ PERSEGUIDOR	■ SCRAMBLE	■ USS ENTERPRISE

UTILITÁRIOS

ROT I - Plus

- SOG - Uma nova e mais poderosa versão de ROT I e seu sensacional Sistema Operacional Gráfico.
- MERGE

ROT II

- ASSEMBLER
- DESASSEMBLER
- COMPILADOR BASIC

PARA BREVE

● O MERCADOR	● ESTRELA NEGRA	● ORÇA I (Orçamento doméstico)
● STARQUEST	● CRISTAL MÁGICO	● ARQ I (Arquivo de Dados)
● ZARAKS	● GUERRILHA CÓSMICA	● CASH-FLOW

EM TODO O BRASIL NAS MELHORES LOJAS DO RAMO.

Informações, Distribuição e Vendas:

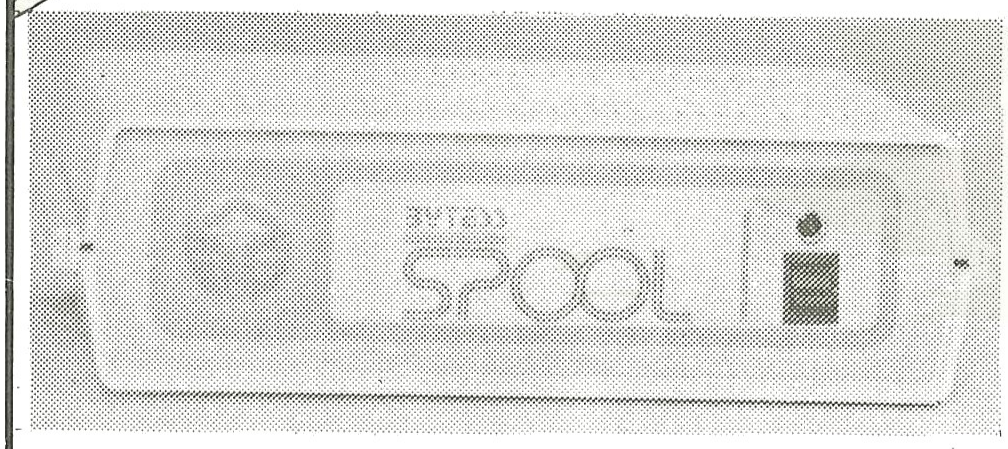
JVA – MICROCOMPUTADORES LTDA.

Av. Graça Aranha, 145 s/loja 01 - Rio de Janeiro - RJ

CEP 20.030 Tel.: (021) 262-6968

Lançamento 83

Liberte-se de sua impressora



O bytesspool é uma memória que otimiza o uso da impressora, permitindo que o computador continue a processar enquanto ela imprime, ganhando-se até 50% do tempo de rodagem de um programa. Versão serial/paralela, de 64 KBytes até 1 MBytes. Permite cópias do conteúdo da memória, pausa de operação, hierarquização de impressão, compressão de dados automática. Opcional: multiusuário dinâmico, que possibilita conectar até 4 computadores em até 4 impressoras.

CLARITRON IND. E COM. LTDA.
01455 - Rua Hungria, 526 São Paulo SP
Tel: (011) 210-7681

BYTESS

MICRO NEWS

Microcomputadores com crédito direto ou leasing

COLOR 64	—	Cr$ 495.000, x 2	(Grátis aplicativos)
CP-200	—	Cr$ 157.000, x 2	(Grátis 20 jogos)
DGT-1000	—	Cr$ 320.000, x 3	(Grátis 20 jogos)
TK-85	—	Cr$ 149.000, x 2	(Grátis 20 jogos)
CP-500	—	Cr$ 749.000, x 2	(Grátis 20 jogos)

Preços sujeitos a alteração

CURSOS DE BASIC COM ATÉ 100% DE DESCONTO
ENTREGA RÁPIDA EM TODO BRASIL

Aplicativos: controle de estoque; contabilidade; folha de pagamento; contas a receber pagar; mala direta; cadastro de clientes e desenvolvimento de software para cada necessidade.

Temos toda linha de periféricos e suprimentos para acompanhar o crescimento de sua empresa.

VISITE-NOS OU SOLICITE UM REPRESENTANTE

MICRONEWS COMÉRCIO E SERVIÇOS LTDA.
R. Assembléia 10 Gr. 3317 - Ed. Centro Cândido Mendes
Tel.: (021) 252-9420 - CEP 20011/RJ.

SoftKristian®

Revendedores Autorizados

Rio de Janeiro

Seletronix
Republica do Líbano, 25-A
Rio de Janeiro - RJ
CEP: 20061

Gachet
R: Dr. El'Jaick, 25 S/5
Nova Friburgo - RJ
tel.: 22.4208

VGC
Av. Brasil, 10 S/07
Araruama - RJ
CEP: 28970

ENTRELIVROS
Av. Rio Branco, 156 - térreo
Rio de Janeiro - RJ

M.C.S.
Visc. de Pirajá, 303/217
Rio de Janeiro - RJ
tel.: 267.8597

Pernambuco

Eletrônica Isabele
R: Porto Alegre, 112
Caruaru - PE
CEP: 55100

Alagoas

Expoente
Av. Siqueira Campos, 838
Maceió - AL
tel.: (082) 223.3979

São Paulo

Imarés
Av. dos Imarés, 457
São Paulo - SP
tel.: 61.4049 - 61.0946

Fotoleo
R: Boa Vista, 314 - 3.º andar
São Paulo - SP
tel.: 35.7131 R/32

Memocards
R: Amador Bueno, 855
Ribeirão Preto - SP
tel.: (016) 636.0586

Fotoptica
Alameda Juruá, 434
São Paulo - SP
tel.: 421.5211

Ritz
R: Frei Caneca, 7
Santos - SP
tel.: 35.1792

Computerland
Av. Angélica, 1996
São Paulo - SP
CEP: 01228

Livraria Poliedro
R: Aurora, 704
São Paulo - SP
tel.: 221.6764

RC Microcomputadores
Av. Estados Unidos, 983
Piracicaba - SP
tel.: 33.7018

Rio Grande do Sul

Advancing
R: Andradas, 1560 galeria
Malcon 518 Porto Alegre - RS
tel.: 26.8246

J.H. Santos
Pça. Otavio Rocha, 41
Porto Alegre - RS
CEP: 90000

India Center
R: Floriano Peixoto, 1112 conj.
33/43 Santa Maria - RS
tel.: (055) 221.7120

Geremia Ltda.
Av. Julio de Castilhos, 1872
Caxias do Sul - RS
tel.: 221.1299

Nordemaq
Av. Julio de Castilhos, 3240
Caxias do Sul - RS
tel.: 221.3516

Micromega
R: Julio de Castilhos, 441 -
1.º andar Novo Hamburgo - RS
tel.: (0512) 93.4721

Bahia

Oficcina
Shopping Center Itaigara
lj40 - 1.º piso
Salvador - BA
tel.: (071) 248.6666

Santa Catarina

Supermicro Show
R: dos Ilheus, 10 lj 6
Florianópolis - SC
tel.: 22.8770

Paraná

Computique
Av. Batel, 1750
Curitiba - PR
tel.: 243.1731

Madison
Av. Mal. Deodoro, 311
Curitiba - PR
tel.: 224.3422

Minas Gerais

Computronix
R: Sergipe, 1422
Belo Horizonte - MG
tel.: (031) 225.3305

Eletrorádio
R: Aquiles Lobo, 441-A
Belo Horizonte - MG
tel.: (031) 222.8903

Micro Poços
R: Assis Figueiredo, 1072
Poços de Caldas - MG
tel.: (035) 721.1883

Blow-Up
Av. Floriano Peixoto, 396
Uberlândia - MG
tel.: 235.1413 - 235.7359

Brasília

Digitec
SCLN 302 bl.A lj.63
Brasília - DF
tel.: (061) 225.4534

• CREDENCIAMOS NOVOS REVENDEDORES PARA TODO O BRASIL

uma construção geométrica que é praticamente idêntica ao algoritmo M, exceto pelo fato de ser um procedimento para se obter a maior medida comum de dois segmentos de reta, e que também é um método computacional que não termina se os tamanhos fornecidos forem incomensuráveis.

Definibilidade

Cada passo de um algoritmo tem que ser precisamente definido; as ações indicadas devem ser definidas com rigor e sem ambigüidade. No algoritmo M supõe-se que o leitor saiba exatamente o que significa dividir M por N e o que é resto. Como não há nada que fale sobre o que aconteceria se M e N não fossem inteiros positivos e como o algoritmo trabalha apenas com números inteiros positivos, devemos assegurar (dentro do critério de definibilidade) que os valores de M e N sejam sempre inteiros e positivos, cada vez que entrar no Passo M1.

Entradas

Um algoritmo não tem nenhuma ou tem uma ou mais entradas, que são as quantidades fornecidas inicialmente antes de começar. Estas entradas são tiradas de um conjunto de objetos. No algoritmo M há duas entradas, chamadas M e N, que são tiradas do conjunto dos números inteiros positivos.

Saídas

Um algoritmo tem uma ou mais saídas, que são quantidades que tem uma relação com as entradas. O algoritmo M tem uma saída chamada N no Passo M2, que é o máximo divisor comum das duas entradas.

Efetividade

Um algoritmo deve ser efetivo, isto é, todas as operações a serem executadas no algoritmo devem ser básicas o bastante para que possam (a princípio, numa quantidade de tempo determinada) ser feitas por um homem usando lápis e papel. O algoritmo M usa as operações de divisão entre dois números inteiros, testa se um número inteiro é zero e assinala o valor de uma variável com o valor de outra. Estas operações são efetivas, pois números inteiros podem ser representados no papel e há pelo menos um método (algoritmo da divisão) para dividir um pelo outro.

Pensando bem, esqueça um pouco o método e fique com a receita. A melhor comparação é mesmo com a receita de um livro de culinária.

Podemos comparar o conceito de algoritmo com uma receita de um livro de culinária. Numa receita, presumivelmente, há as qualidades de finidade (um prato de comida), entradas (farinha, ovos etc) e saída (o jantar, um lanche), mas perceptivelmente não há muita definição. Há momentos em que uma receita foge a uma definição mais precisa, por exemplo, "Adicione uma pitada de sal"; uma pitada pode ser definida como 1/8 de uma colher de chá. "Sal" é suficientemente definido, mas onde adicionar o sal (em cima, ou ao lado). Instruções como "Agite suavemente até a mistura se dissolver", e "Aqueça o conhaque numa pequena caçarola", são explanadas adequadas para um cozinheiro treinado, talvez, mas um algoritmo tem que ser especificado num grau tal de modo que qualquer computador possa seguir suas ordens.

A restrição de finidade não é forte o suficiente para uso prático: um algoritmo não requer somente um número limitado de passos, mas o menor número possível de passos. Na prática, não queremos somente algoritmos mas queremos bons algoritmos (inclusive com algum senso estético).

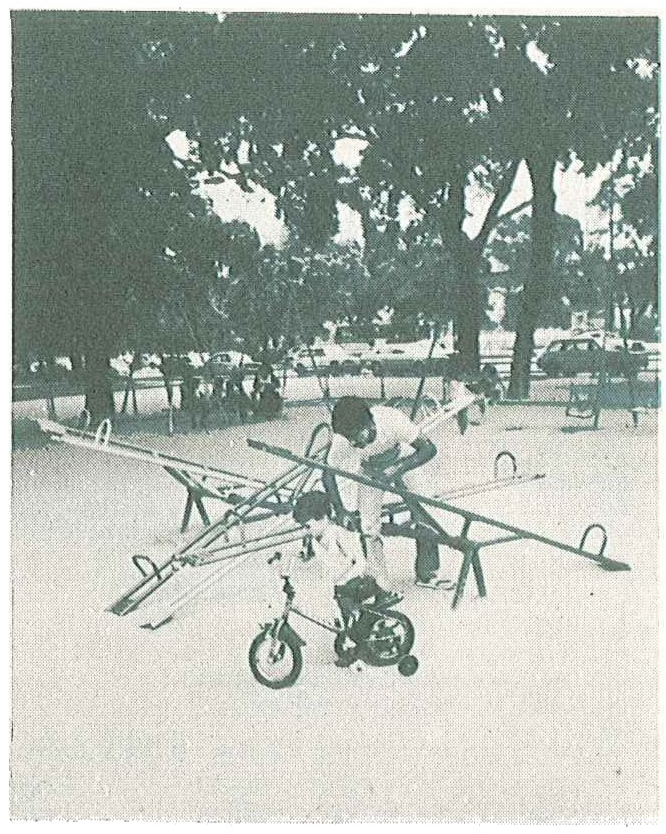
Um dos critérios para determinar se um algoritmo é bom é a quantidade de tempo que este leva para ser executado; outro critério é a sua adaptabilidade em vários tipos de computadores além, naturalmente, da simplicidade e da elegância.

Bibliografia

- *The Theory of Algorithms (A. A. Markov).*
- *Fundamental Algorithms (Donald E. Knuth) – The Art of Computer Programming.*
- *Communications of the Assossiation for Computing Machinery.*
- *Journal of the Assossiation for Computing Machinery.*
- *Computer Journal.*

CALENDÁRIO

INFORMÁTICA NAS FÉRIAS



O CBI – Centro Brasileiro de Informática apresenta para o período de janeiro e fevereiro uma série de cursos objetivos para aperfeiçoamento profissional em segmentos específicos da área de Informática. São os seguintes os cursos programados: *Teleprocessamento, Programação Pascal, Técnicas de Programação Estruturada, Análise Estruturada, Manutenção DGT 100/1000, Microprocessador 8086/8088 e Controle de Processos.*

Informações complementares sobre os cursos podem ser obtidas no CBI – Av. Passos, 115 – sala 215 – RJ – Tel.: (021) 233-1123.

SCI

A SCI está promovendo cursos para janeiro e fevereiro: *Microcomputadores para Uso na Empresa, Gerenciando o Profissional de Processamento de Dados, Análise e Projeto Estruturado de Sistemas – Workshop, Panorama da Teleinformática Brasileira, Planejamento Estratégico e Tático de Informática, Fundamentos de Bancos de Dados, Fundamentos de Teleprocessamento e Informática para Usuários.*

Informações pelos telefones (011) 289-0099 e (021) 294-7438.

BASIC PARA CRIANÇAS

A Benny e a Feira Permanente de Microcomputadores estão voltando a atenção para as crianças e promovendo curso especial para elas. Para este curso foi elaborado um livro especial e as crianças aprenderão brincando e se di-

vertindo em micros, com professores especialmente treinados.

O preço do curso é de Cr$ ..... 45.000,00, incluindo o material didático.

Informações pelo telefone (011) 570-1555.

CP / M & MP / M AVANÇADO

O Núcleo de Computação Eletrônica da UFRJ está promovendo em janeiro um curso de *CP/M & MP/M Avançado,* que se destina àqueles que desejam se aprofundar no estudo da estrutura interna desses sistemas operacionais, de suas compatibilidades e potencialidades.

As inscrições estão sendo realizadas, não havendo possibilidade de reserva de vaga. É necessário comprovação de conhecimentos de Assembler e noções da utilização de CP/M.

Para maiores informações dirija-se à Secretaria de Cursos na sala C-1010 do Núcleo de Computação Eletrônica, no bloco C do CCMN – Cidade Universitária – C.P 2324 – RJ – Tel.: 290-3212, Ramal 224.

DATAMICRO

A Datamicro está promovendo diversos cursos. Estão programados os de *Linguagem Basic, Introdução aos Microcomputadores, Aplicação dos Microcomputadores na Engenharia Civil,* e outros. Todos em diversos horários, atendendo à conveniência dos interessados.

Existe programado também os cursos de *Microcomputador para Crianças* e *Aplicação do Microcomputador na Engenharia.*

GRANDE ABC

A EMSISA promove regularmente cursos de processamento de dados nas linguagens *Cobol* e *Basic.* As aulas são teóricas e práticas, em laboratório (TK82 – D8000 – D8100 e CP200), nos períodos da manhã, tarde e noite, e aos sábados das 9:00 às 12:00 horas. As turmas são compostas de, no máximo, 12 alunos, que recebem apostilas completas e certificado de conclusão. A Escola também desenvolve cursos fechados para empresas e escolas. Informações e inscrições à Av. Getúlio Vargas, 334 – Tel.: 414-3151 CEP 09700 – São Bernardo do Campo – SP.

ERKLA

A Erkla já está com sua programação de cursos para o 1º semestre de 1984 pronta. Destinam-se a técnicos e engenheiros e têm por finalidade o aperfeiçoamento técnico desses profissionais, sendo totalmente apostilados e utilizando recursos audiovisuais.

Em janeiro inicia-se o *Basic I,* no dia 10; o *Basic II* no dia 12 e *Microprocessadores Z80* no dia 9. Em fevereiro, *Componentes Eletrônicos* no dia 6.

Os telefones da Erkle são (011) 67-8339 e 826-1499.

STANDARD MUMPS

Um curso de Standard Mumps para mini e microcomputadores, com duração de 40 horas, será ministrado pela Matrix Sistemas e Serviços Ltda., na Rua Maestro Elias Lobo, 70, CEP 01433, São Paulo, capital, telefone 640688.



MICRO-KIT

A Micro-Kit Educacional continua oferecendo com sucesso os cursos de *Basic Intensivo* para adultos e pessoal de empresas e *Basic I e Avançado* para crianças entre 10 e 14 anos.

Maiores informações na Rua Visconde de Pirajá, 303 s/loja 210 – Tel.: (021) 521-4638 e 267-8291 – RJ.

ADP SYSTEMS

A ADP Systems, através de sua divisão de ensino, está divulgando sua programação para janeiro: com início no dia 16 oferece *Básico/Lógica Estruturada e DOS/VS,* das 8:30hs às 11:30hs, às segundas, quartas e sextas-feiras.

Para maiores informações contacte a ADP pelos telefones (011) 285-3283, 285-4238 e 283-3157.

SEMINÁRIOS DE FEVEREIRO

A Compucenter tem programados para fevereiro dois seminários: o Data Entry, de 22 a 24 e o de Administração de Dados, de 20 a 22. O Data Entry é uma das atividades-chave das operações de processamento de dados e se mantém como uma das mais desafiantes para uma gerência efetiva. Os gerentes de entrada de dados continuamente tratam de problemas como alta rotatividade de pessoal, absenteismo, baixa moral e outros. O Data Entry – técnicas para aumentar a produtividade – é um seminário projetado especialmente para debater os problemas dos gerentes de entrada de dados. O consultor, Phillip W. Bugg, da Myers & Associates é o instrutor desse seminário. Já o seminário sobre Administração de Dados é destinado aos profissionais diretamente envolvidos na problemática de modelar os dados de forma a obter o melhor rendimento de seu banco de dados. Entre eles, os gerentes PD, analistas de sistemas e de software, programadores de aplicações e demais profissionais que estejam envolvidos com projeto, implementação e operação de bancos de dados. A Compucenter, com sede em São Paulo, atende pelo telefone (011) 255-5988, telex 011 21689 CPUT BR e pela Caixa Postal 51674. Para este ano a Compucenter promete um plano especial para as empresas que inscreverem seus técnicos em seu programa de seminários.

CESAR DE ARAUJO LIMA

Usuário sem socorro na hora da manutenção

Uma pesquisa feita pela revista *Interface* revelou que o usuário leigo ficaria em apuros caso tentasse consertar seu microcomputador numa firma anunciante de grandes jornais, páginas amarelas ou jornais especializados em classificados. A grande maioria dos anunciantes é composta de fabricantes de equipamentos, *Software house* e postos de venda de microcomputadores, enquanto a maior parte das empresas de manutenção dedica-se a jogos computadorizados ou equipamentos de vídeo. Cesar de Araújo Lima, do Centro Brasileiro de Informática, expôs, no *I Encontro Regional dos Usuários de Microcomputadores*, realizado em setembro de 83 pela revista *Interface* no Clube de Engenharia do Rio de Janeiro, a deficiência da estrutura de manutenção, que deveria dar garantia plena de funcionamento aos equipamentos comercializados.



Há um esforço muito grande dos fabricantes e comerciantes de microcomputadores para vender o maior número possível de equipamentos. Todos os dias aparece um novo fabricante ou uma nova loja, sem a devida atenção à estrutura de manutenção. Com isso o equipamento vendido não tem garantia de funcionamento contínuo.

Também já está bem claro que os grupos de micros — quaisquer entidades que congreguem um grande número de usuários, como cooperativas, por exemplo — têm um poder de barganha bastante significativo junto aos fabricantes ou vendedores, conseguindo preços e condições de manutenção bem mais vantajosos que os dos usuários independentes.

Manutenção preventiva — Determinadas partes dos sistemas dos microcomputadores necessitam de tempos em tempos de uma manutenção dita preventiva, isto é, para evitar que o equipamento apresente uma falha grave no futuro. Como exemplo, cita-se as unidades de disquete, cassete, impressoras.

Manutenção corretiva — Sempre que ocorre um defeito que torne inoperante o equipamento, na parte ou no todo, faz-se a manutenção corretiva. Os defeitos ocorrem aleatoriamente e o atendimento deve ser feito o mais rápido possível. Por isso a empresa prestadora de tais serviços deve contar com um corpo técnico de bom nível, instalações adequadas, peças de reposição e módulos de reposição.

As características que influem no tempo médio entre falhas (MTBF) são as seguintes: projeto de equipamento, qualidade dos componentes a ele associados, cuidados de produção, *Burn-in*, controle de qualidade, embalagem e transporte, rede de alimentação AC do usuário, condições de umidade, maresia, poeira, temperatura das instalações do usuário e cuidados com o equipamento.

Cuidados do usuário — Diversos fatores que influenciam o MTBF do equipamento são de responsabilidade exclusiva do fabricante. Mas existem alguns fatores que são de responsabilidade do usuário. Se há alguma dúvida quanto à qualidade da rede de alimentação, deve-se pedir a um profissional para avaliá-la e, se for o caso, instalar um regulador de tensão e um eliminador de transientes; o equipamento deve estar instalado em ambiente isento de impurezas e de preferência com temperatura controlada. O usuário deve cuidar das manutenções preventivas recomendadas pelo fabricante.

Qualidade e burn-in — Das variáveis que influenciam diretamente na performance dos microcomputadores, o controle de qualidade (CQ) e o *burn-in* são os mais importantes.

Todos os insumos que compõem o microcomputador devem passar por controle de qualidade mais ou menos rigoroso, de acordo com seus antecedentes.

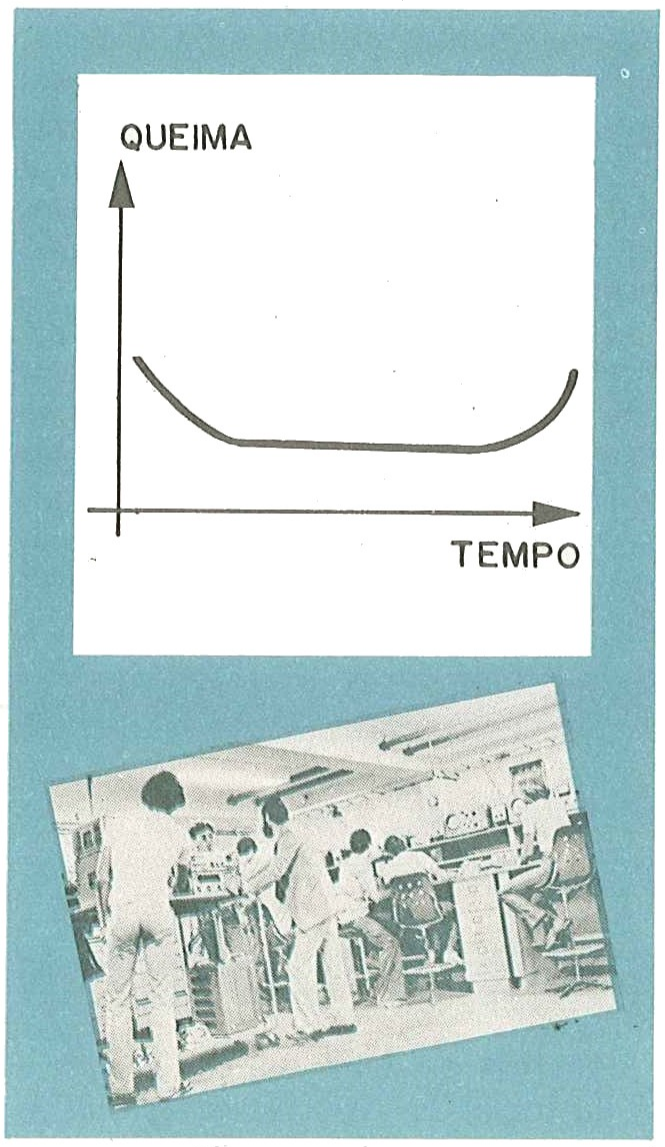


Figura
queima x tempo

O burn-in ou envelhecimento — É uma etapa de estudo necessário nos componentes eletrônicos, que têm característica de queima, variável no tempo, de acordo com o gráfico mostrado na figura

Ou seja, tendem a apresentar um grande número de defeitos no início de sua utilização, caracterizando uma "mortalidade infantil" que o *burn-in* acelera, fazendo com que os componentes que passem por esta etapa sejam mais confiáveis. Já o CQ do equipamento montado, deverá ser muito criterioso em suas diversas etapas de fabricação, a fim de evitar que as falhas de produção e/ou componentes sejam repassadas ao consumidor, gerando um MTBF compatível com o elevado preço desses equipamentos.

Garantia — Até que ponto seu micro está seguro? Os fabricantes de microcomputadores fornecem certificados de garantia que variam de três meses a dois anos. Os fabricantes operam em dois grupos distintos: os que mantém filiais nos grandes centros, verticalizando a assistência técnica, e os que operam com empresas credenciadas na prestação de assistência técnica. Estes últimos constituem maioria e delegam a terceiros a responsabilidade de atender os usuários.

Comparando as duas políticas de assistência técnica a primeira é, sem dúvida, a mais vantajosa para o usuário, que terá de tratar com uma única empresa que é a própria fabricante do equipamento, sem a intermediação de terceiros. No segundo caso o usuário eventualmente depara-se com alguns credenciados sem muita preparação ou sem estoque suficiente para fazer uma manutenção rápida, ou ainda sem módulos de reposição capazes de atender a um defeito de solução mais demorada. Com isso passa a ter um período longo com o equipamento inoperante.

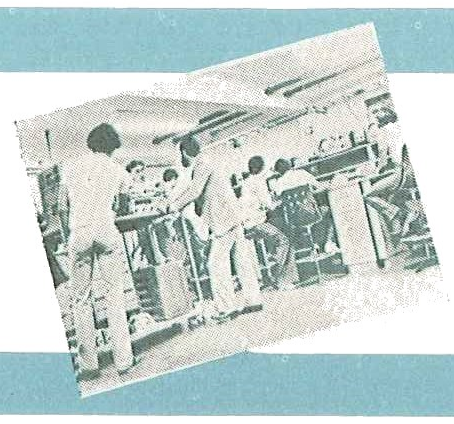
Prazo de garantia — Durante o período de garantia os problemas que ocorrem são minimizados, pois o usuário não paga diretamente pelos serviços que tenham de ser feitos.

Findo o período de garantia, os usuários têm a seguinte alternativa: primeiro, só recorrer ao serviço de assistência técnica quando o equipamento apresentar defeito, pagando cada vez o serviço executado; segundo, fazer um contrato de manutenção com uma empresa, com um custo fixo mensal, um prolongamento da garantia oferecida pelo fabricante.

Contrato de manutenção — Um contrato de manutenção deve conter de forma clara o preço a ser pago pelo usuário, forma de pagamento, todas as peças cobertas pelo contrato, as circunstâncias em que o contrato não poderá cobrir eventuais defeitos, período estipulado como máximo para a entrega do equipamento reparado.

O usuário deve observar o binômio *custo x benefício*, pois o contrato mais barato não é necessariamente o melhor negócio. Por isso é bom que o usuário faça uma pesquisa entre as empresas que possam prestar esse serviço.

Uma saída: associações de usuários



Há um número significativo de empresas aptas a prestar bons serviços de manutenção a microcomputadores. O custo não é fator determinante da escolha da empresa, mas sim a qualidade e eficiência dos serviços. É aconselhável que o usuário conheça as instalações e o equipamento utilizados na manutenção de seu microcomputador, bem como a capacidade técnica das pessoas envolvidas no processo.

Outra questão a ser considerada é o tipo de manutenção: se a nível de componentes ou a nível de módulos.

A manutenção a nível de componentes normalmente só é realizada nas instalações da empresa de manutenção, pois exige equipamentos especiais. Mas é a melhor escolha para o usuário, pois terá sua placa de circuito impresso de volta e não um módulo de origem desconhecida. De qualquer maneira, é bom que disponha de módulos *back up* para o caso de determinadas falhas que exijam um estudo mais profundo de sua causa, a fim de que o usuário não tenha de ficar aguardando indefinidamente a reposição.

As empresas de manutenção devem ter um serviço de atendimento telefônico, para evitar que o usuário leigo requisite a presença do técnico ou vá com o equipamento até a empresa, procurando solucionar um defeito que poderia ser reparado pelo próprio usuário ou às vezes até inexistente. Os "defeitos" mais comuns são: erros de operação, cabos não conectados ou mau conectados, teclados travados mecanicamente, fusíveis queimados, periféricos desligados e toda a sorte de pequenos problemas que um pouco de paciência e boa vontade resolveriam até por telefone.

Para o pessoal técnico possuidor de microcomputadores e não familiarizado com a tecnologia e a arquitetura de microcomputadores, o Centro Brasileiro de Informática – CBI – está realizando curso de manutenção de microcomputador pessoal DGT 100, em convênio com a fabricante Digitus, possibilitando aos usuários a abertura do *hardware* de seu equipamento, por meio de esquemas claros, iniciativa que deveria ser imitada pelos outros fabricantes.

A comunidade de usuários de determinado modelo de microcomputador, por sua vez, deve se unir em grupos, associações, cooperativas ou qualquer outro meio, para pressionar o fabricante a solucionar possíveis problemas de projeto ou qualidade nos equipamentos. Isso é do interesse tanto do usuário quanto do fabricante. Este teria retorno das informações de campo, que nem sempre são reportadas pelas empresas concessionárias. Com isso a performance dos equipamentos e seu tempo médio entre falhas seriam melhoradas gradativamente.

Outra vantagem é a troca de informações, programas e experiências, enfim, o conhecimento técnico, que se propagaria em benefício da própria comunidade, que se tornaria mais exigente com os fabricantes e também com a manutenção, elevando o nível profissional destes últimos.

O veículo das reivindicações

Abertos os debates pelo presidente da mesa, Tarcisio Neves da Cunha, Cesar da Costa, diretor da *Interface*, declarou que os usuários têm na revista que dirige o veículo natural de apresentar suas reivindicações junto aos fabricantes, salientando que foi uma pesquisa elaborada pela Interface que tornou claro o problema.

O representante da Tecnoccop, Eduardo Silva, citou a experiência da empresa que representa, que negocia com a Prológica e outros fabricantes seu credenciamento como empresa de assistência técnica. Sugeriu que os usuários se unam em grupos para tratar de manutenção ao nível de módulos *back-ups*.

Bernardo, representante do *Grupo de Micro da Telerj*, contou que esse grupo é composto de 90% de usuários do DGT 100 e que tem recebido bom atendimento do fabricante. Mas ressaltou que é uma situação atípica, pois trata-se de um cliente que comprou mais de 500 equipamentos. O grupo avalia fornecedores de manutenção pois pretende credenciar um único representante.

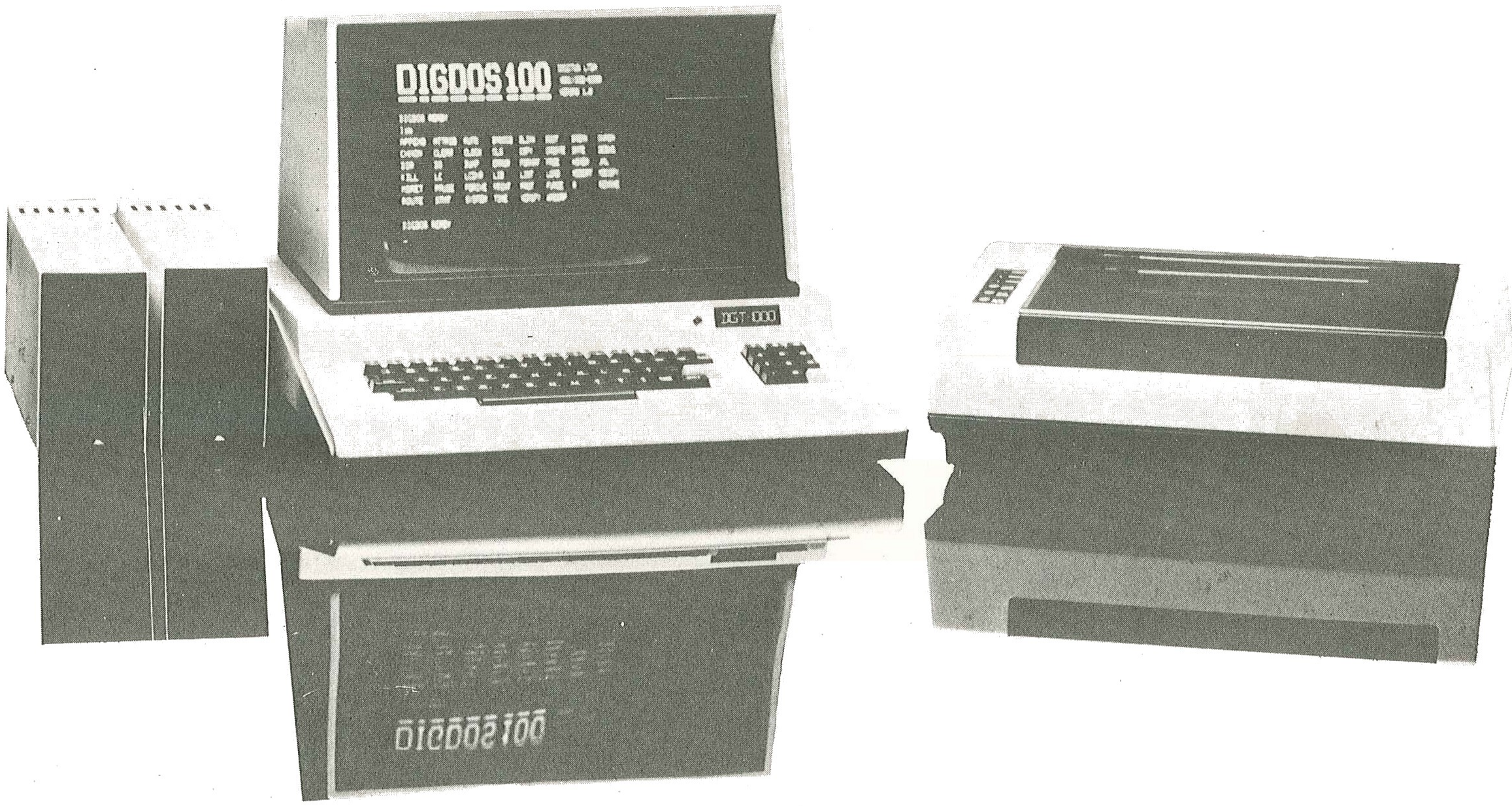
Luís Carlos, representante da *Microshow*, disse aconselhar seus clientes a praticar a utilização plena do equipamento durante o exígüo prazo de garantia, para que os componentes com vício de origem danifiquem-se durante esse período, sem ônus para o usuário. Também os aconselha a usarem desreguladores de tensão. A *Microshow* substitui módulos e eventualmente o próprio equipamento. Quanto a peças de reposição, Luís Carlos acredita que o fabricante tem estoque reduzido de peças e, se tiver de optar entre entregar a peça à produção ou à assistência técnica, entrega-a à produção.

Flávio, da firma *Sistema,* informou que sua empresa mantém um plantão telefônico para atender os clientes e que os contratos de manutenção que realiza cobram apenas a mão-de-obra, sendo as peças cobradas à parte. Ele acha que o usuário deve fazer limpeza periódica no equipamento. A *Sistema* manda o técnico até o cliente. Se o defeito não for reparado em duas horas o microcomputador é retirado para laboratório. Se possível, troca somente o módulo por um *back-up.* Após o reparo, o módulo original é devolvido.

Júlio Bastos, da *BCD Engenharia,* garantiu que seus clientes não percebem quando a garantia têrminou e começou um contrato de manutenção, a não ser no que diz respeito ao ônus que este representa. A *BCD Engenharia* orça o contrato de manutenção de acordo com a utilização da máquina, ou seja, a perda física ou jurídica definem a utilização maior ou menor. Acredita que os usuários devem criar grupos para lutarem por preços mais atrativos.

Ari, da *Yatek,* disse que o contrato de manutenção de sua empresa funciona como um seguro contra defeitos e que confecciona bancadas de teste automáticas, que permitem reparos rápidos em DGT 100 e CP 500. Criticou os fabricantes que não fornecem informações sobre modificações e não fornecem esquemas para os usuários, pois com poucos conhecimentos técnicos não é difícil manter um microcomputador.

Homero, do *Projeto Ciranda,* criticou o prazo de garantia de alguns equipamentos – de apenas três meses – e solicitou aos fabricantes a exemplo dos montadores de automóveis, que garantissem os componentes por um período de dez anos, pois com a evolução tecnológica os componentes têm rápida tendência à obsolescência.



SINTA NOS DEDOS ESTA NOVA CONQUISTA

Já não é preciso escolher. Agora você tem o microcomputador DGT–1000, com design moderno e novas incorporações que lhe conduzirão à decisão certa.

O DGT–1000 é modular e dependendo da sua necessidade ele se expande até um grande sistema. A DIGITUS lhe assegura a possibilidade de expansão do DGT–1000 através das interfaces que comercializa:

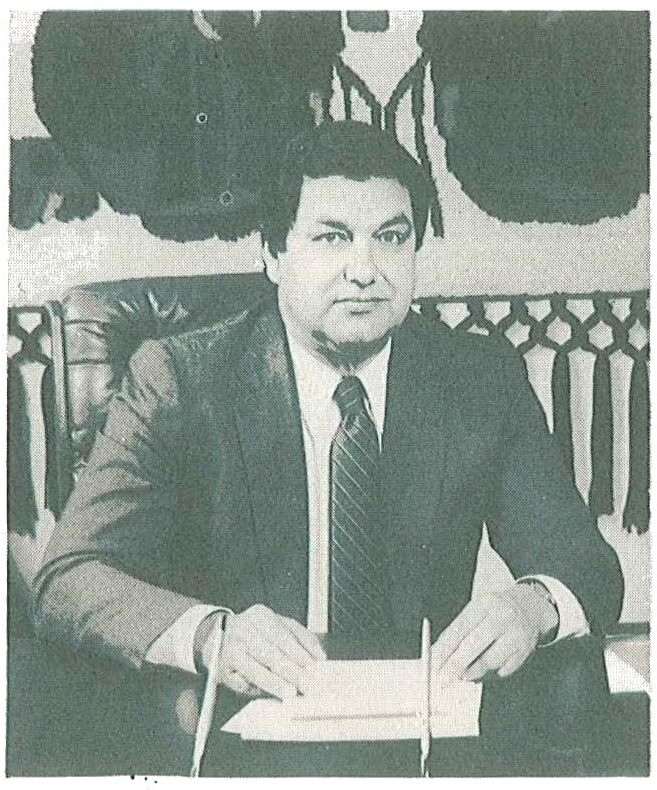
- Interface printer, paralela para impressora tipo centronis.
- Interface para unidade de disco flexivel 5 1/4 *(até 4 unidades de 184 KB cada).*
- Expansão de memória até 64 KB.
- Interface colorida com alta resolução gráfica.
- Interface RS 232.
- Sintetizador de voz.
- Sistema operacional DGP/M totalmente compatível com CP/M *(com o DGP/M você poderá usar os famosos programas WORDSTAR e dBASE II).*

DIGITUS

Rua Gávea, 150 Belo Horizonte
Tel.: (031) 332-8300 Telex: 3352

Aracaju (079) 224-1310/224-6111 **Barueri** (011) 421-5211/282-3274 **Brasília** (061) 273-2128/225-4534/226-9201/226-8701/242-6344 **Belo Horizonte** (031) 223-6947/223-9445/222-7889/225-3305/225-6239/226-6336/226-5734 **Bélem** (091) 225-4000/225-3000 **Campinas** (0192) 32-6322 **Curitiba** (041) 232-1750/243-1731 **Divinópolis** (037) 221-2004/221-9194 **Fortaleza** (085) 224-3923/231-4822/224-0544/244-4691/227-5878/224-4235/226-4922 **Florianópolis**,(0482) 23-1039 **Goiânia** (062) 223-1165 **João Pessoa** (083) 221-6743 **Londrina** (0432) 23-7110 **Maceió** (082) 223-3979 **Montes Claros** (038) 221-2599 **Niterói** (021) 714-0112/710-2780 **Novo Hamburgo** (0512) 93-4721 **Ouro Preto** (031) 551-1931 **Poços de Caldas** (035) 721-5810 **Porto Alegre** (0512) 26-8246/26-1194/26-0194/22-9782/21-4189 **Recife** (081) 326-9969/326-9318 **Ribeirão Preto** (061) 636-0586 **Rio de Janeiro** (021) 288-2650/551-8942/257-4398/228-0734/264-0143/248-8159/284-5649/252-9057/263-1241/252-9420/292-0033/236-7175/252-2050/222-4515/252-2752/253-3395/267-1093/267-8291/247-1339/285-5947/295-8194/259-1516/247-7842/332-4166 **Salvador** (071) 248-6666/235-4184/242-9394/243-2684 **Santa Maria** (055) 221-7120 **São Paulo** (011) 231-3922/815-0099/852-7700/258-7311/222-1511/258-3954/282-2105/853-9288/280-2322/881-0200/61-4049/61-0946 **Taubaté** (0122) 32-9807 **Vitória** (027) 223-5147

SOLUÇÃO GLOBAL E SCON ON LINE



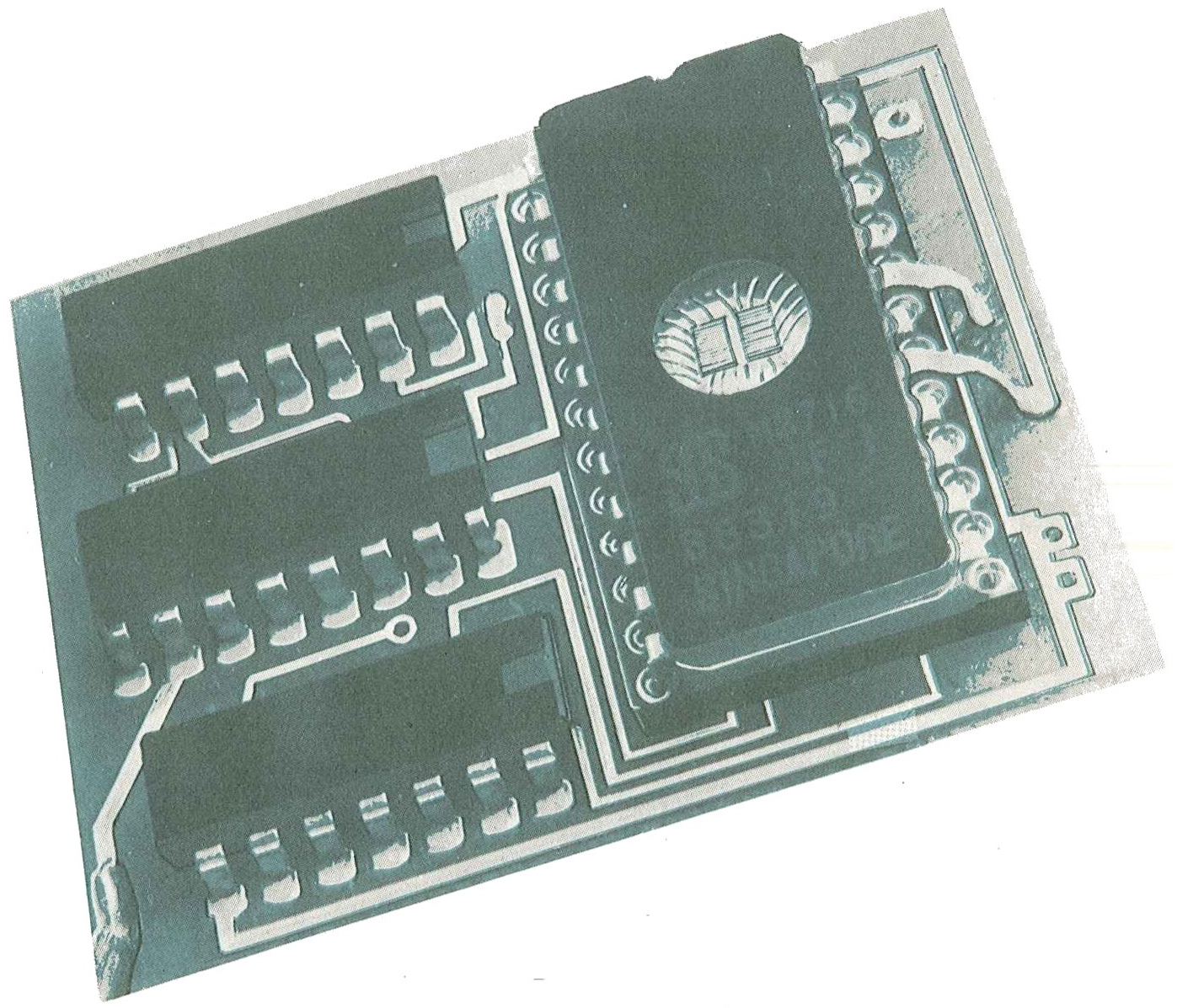
Edes Landim (foto), presidente da ADP Systems, Empresa de Computação Ltda., informou que dois novos serviços oferecidos pela empresa que dirige – *Solução Global* e *Scon on Line* – custaram o investimento de 250 milhões de cruzeiros desde 1980. A ADP põe à disposição do usuário um microcomputador conectado diretamente a um computador de grande porte, instalado no birô.

COMPUTIQUE & ITAUTEC

A Computique está agora vendendo o microcomputador I-7000 da Itautec, resultado de projeto feito por técnicos e engenheiros brasileiros, que tem os seguintes softwares aplicativos: contas a pagar, contabilidade, contas a receber, folha de pagamento, controle de estoque, planilha eletrônica, mala direta, gerenciador de banco de dados, editor de texto, traçador de gráficos, linguagens etc. Permite a ligação do micro a computadores maiores e o teleprocessamento. Tem os suprimentos: fitas para impressoras, disketes 5 1/4 e 8", formulários contínuos, etiquetas adesivas, fitas magnéticas, arquivos para disketes e outros.

IVANITA GERA CARACTERES

A Sacco Microcomputadores, situada na avenida Euzébio Matoso, 167, São Paulo (telefone 814-0598) já está vendendo Ivanita, que complementa o gerador Ascii de um Aple, permitindo escrever corretamente em português, acentuando todas as vogais minúsculas e maiúsculas, assim como a utilização dos caracteres especiais ç, º, ª e §.



SOFTWARE DA DIGITUS

A Digitus já dispõe de programas elaborados para seus equipamentos, nas mais diversas áreas.

- *Pacote de comunicação*, que propiciará ao usuário a utilização da tecnologia da comunicação serial de dados.
- *Digcalc*, programa que combina a conveniência e a familiaridade de uma calculadora de bolso, oferecendo facilidades de uma memória poderosa e tela para o acompanhamento da operação.
- *Jogos em fita ou disco*, Demofita ou Demodisco, nas modalidades I e II, são jogos criativos: o I em linguagem de máquina e o II em Basic.
- *Digfile e Banco de Dados* – sistemas de bancos de dados de enorme versatilidade para empresas.
- *DGP/M*, um sistema operacional totalmente compatível com o CP/M.
- *Controle de estoque*, para aplicação em empresas de pequeno e médio porte.
- *Processador de textos*, um programa criado para o processamento de palavras.
- *Editor Assembler*, um software que permite programação em linguagem de máquina usando mnemônicos do Z-80.

TAMBÉM PARA CEGOS

Dois programadores da Prodam – Cia. de Processamento de Dados do Município de São Paulo – ambos portadores de deficiência visual, desenvolveram o Projeto Libra – Listador Braille –, tornando capaz o computador de imprimir, em listagens, textos

em linguagem braille. Com os recursos disponíveis em qualquer computador IBM de pequeno, médio ou grande porte, o projeto, elaborado em linguagem Cobol-Ans, contendo uma sub-rotina em linguagem Assembler, permite listar arquivos com registros de diferentes tamanhos sem utilizar qualquer alteração de hardware e software. Na impressão braille a linha de 132 caracteres passa a comportar 44 caracteres. Antes do Projeto Libra, os profissionais cegos da Prodam necessitavam de seus companheiros de trabalho para desempenhar suas atividades. Agora, com a utilização de uma máquina de escrever em braille, eles demonstram performances consideradas normais para os padrões da empresa e são submetidos aos mesmos testes de avaliação quanto à produtividade, ao desenvolvimento técnico e ao comportamento profissional. Lothar Bazanelle, um dos idealizadores do projeto, já venceu muitas etapas e dentro dos critérios da empresa já trabalha como analista de software. A Prodam fornece gratuitamente uma cópia (listagem) do programa Cobol e um manual de utilização, com as instruções sobre o projeto.

O MICRO DE 16 BITS

A Softec Engenharia de Sistemas e Comércio Ltda., fabricante do EGO, o primeiro microcomputador nacional de 16 bits, está mudando sua fábrica para o bairro do Jaguaré, em São Paulo, em prédio de 1000 m^2. Nas novas instalações estarão funcionando também o departamento de desenvolvimento de software, o administrativo, compras, desenho e arte final. O objetivo é aumentar a produção, atingindo no primeiro trimestre deste ano 100 unidades por mês.

RTTY E CW

O Py2CME Homero de Paula Lima Jr. informa que já é possível transmitir e receber em telegrafia (CW) e em telex via rádio (RTTY), em diversas velocidades, com o auxílio de um microcomputador pessoal da linha Sinclair e interfaces que podem ser encontrados com ele mesmo, na Caixa Postal 202, telefone (0152) 713759, na Rua Pedro de Toledo, 75 (Estrelas), Itapetininga, SP, CEP 18200. Homero está QRV para os colegas.

ESTUDANTAÇO

Um pequeno grupo de estudantes de engenharia eletrônica de São Paulo entrou no mercado de assistência técnica a microcomputadores, abrindo a DN Craft Eletrônica de Sistemas, na avenida Indianópolis, 503, telefone 570-1690. A empresa presta assistência técnica para máquinas de qualquer marca, e especialmente para as das linhas Sinclair (modelos TK, NE e CP 200), TRS-80 (Dismac série 8000, Prológica CP 300 e 500 e Sysdata, por exemplo) e Apple (Unitron, Microengenho etc.).

FERRAMENTA DIDÁTICA

O Colégio Etapa (SP) vem utilizando microcomputadores como instrumento de apoio programado em diversas disciplinas, iniciando os alunos no campo da computação com máquinas de pequeno porte como o TK 83 da Microdigital. O professor Bindi, do Colégio Etapa, assegura que os resultados têm sido muito bons. Desenvolvendo o raciocínio e a criatividade, os microcomputadores transformaram-se rapidamente em ferramentas didáticas.

VALES DE REFEIÇÃO

Com um crescimento de 130% em seu faturamento nos últimos 12 meses, a empresa paulista Vale Refeição Ltda. executa o controle de reembolso, emissão de cheques e todo o serviço de faturamento em seu próprio centro de processamento de dados, de forma a agilizar todas essas rotinas e oferecer aos clientes um serviço muito mais dinâmico.

SOFT-HOUSE NO ABC

Para desenvolver software nas áreas de jogos de entretenimento, jogos educativos, sistemas científicos e sistemas administrativos, acaba de ser criada, na avenida Getúlio Vargas, 334, Vila Baeta Neves, São Bernardo do Campo, em São Paulo, a EMSISA, equipada com um microcomputador Alfa 3003 da Dismac. Os aplicativos são: contábil, folha de pagamento, crediário, transportadoras, ativo imobilizado, faturamento, contas a receber e a pagar e controle de estoque.

BANCO DE 24 HORAS

Sete empresas nacionais e estrangeiras apresentaram proposta na concorrência aberta pelo Banco 24 Horas para expandir sua rede de ATM. As propostas pertencem à ABC Sistemas, Digirede, IBM, Itautec, NCR, Racimec e SID. A análise durará 40 dias, segundo informou Gilberto Dib, diretor da Tecnologia Bancária, empresa prestadora de serviços de informática responsável pelo Banco 24 Horas, que já opera em São Paulo, Rio e Curitiba. O equipamento ATM permite a operação on-line e off-line, through-the wall e facilidades operacionais. O banco tem uma ampla gama de serviços que permite ao cliente realizar saques até o limite de seu cartão, fazer depósitos ou transferências em conta corrente ou caderneta de poupança e pagar contas, carnês e tributos com cheque ou debitando diretamente na sua conta, além do saque de valor fixo.

NOVIDADES DA COBRA

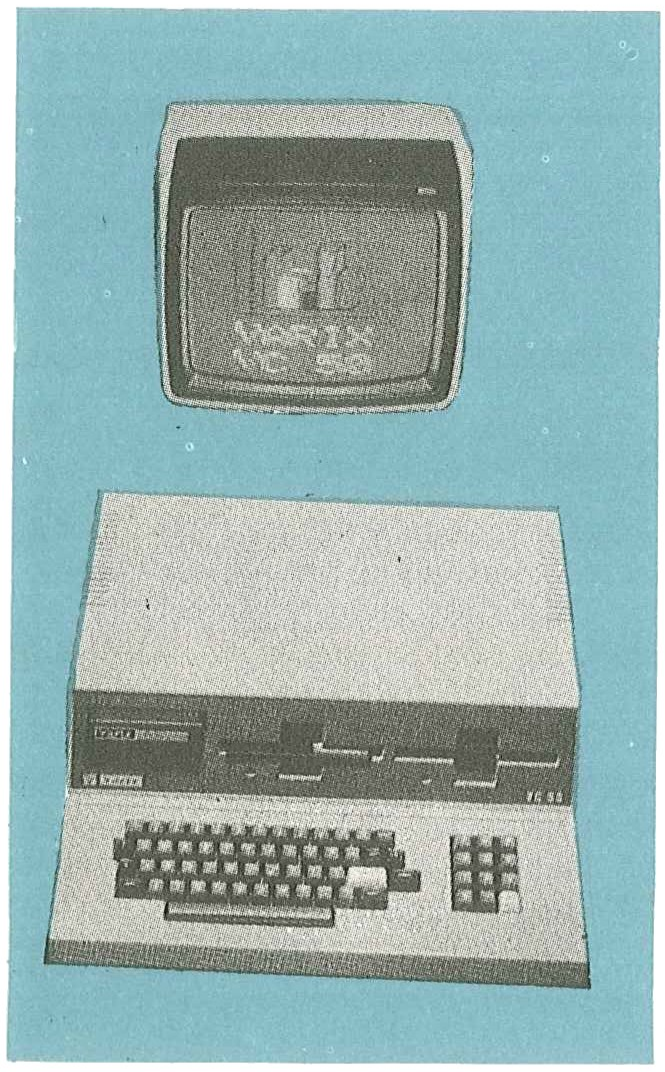
A Cobra apresenta algumas novidades relacionadas ao seu microcomputador Cobra 305, a começar pelo seu vídeo, agora em fósforo verde. Além disto, está sendo entregue com unidades de leitura/gravação de dupla cabeça, possibilitando a utilização de discos flexíveis de dupla face. Eleva-se assim sua capacidade de armazenamento a 4M bytes, utilizando 4 unidades.

A Cobra está também comercializando uma nova versão do Cobra 305, com uma só unidade de disco flexível e custo 30% inferior ao da versão com duas unidades, objetivando aplicações de entrada e comunicação de dados. Outra inovação é o lançamento de seu terceiro sistema operacional, denominado SPM (SISTEMA-PADRÃO PARA MICROPROCESSADORES), compatível com o CP/M (versão 2.2), suportando as linguagens de programação COBOL ANS, FORTRAN IV e LPS. Assim, o usuário tem ampliado seu leque para escolher o sistema operacional mais adequado a cada aplicação:

- MUMPS, para manipulação de bancos de dados, aplicações interativas com múltiplos usuários;
- SOM, para aplicações comerciais, administrativas, científicas, entrada e transmissão de dados, utilizando as linguagens COBOL I, LPS, FORTRAN e LTD, além de vasta biblioteca de utilitários;
- SPM, cuja grande vantagem é a compatibilidade com o sistema operacional mais popular para micros de 8 bits.

A Cobra liberou também para comercialização a interface multiplexadora, de até 8 portas assíncronas, com velocidades entre 75 e 9600 bps. Este dispositivo permite utilizar o Cobra 305 em diversas aplicações que exigem maior capacidade de comunicação. Por exemplo, o uso do Cobra 305 como concentrador de terminais financeiros em agências bancárias, sua conexão a equipamentos de laboratório em usinas de cana-de-açúcar, etc.

NOVOS PRODUTOS



O VARIX VC 50

A Engetécnica está lançando o microcomputador VARIX VC 50, que tem grande versatilidade para a sua classe, incluindo em sua configuração mínima, 64 K RAM Dinâmica, com refresh transparente; 16 K EPRON com Varix Basic compatível com Rádio Shack Color Computer; 16 K de ROM PACK (Cartuchos), intercambiáveis com Visicalc, processamento de textos, jogos etc.; Interface para cassete; Interface para impressora; Interface RS 232C; 2 joysticks; CPU 6809 (16 bits de configuração interna); saídas de vídeo para monitor e para TV normal ou modificada; saída de áudio, juntamente com vídeo ou separada; teclado completo, profissional com numérico separado; linguagem basic e assembler; sistemas operacionais VDOS, VFLEX e VOS9; periféricos até 4 disk driver, sendo dois internos e dois externos; impressora modelo VP 50 100 cps, bidirecional, 132 colunas com procura lógica; modem e cassete.

CP / M NO CP-500

A Chip Shop Computadores lançou novos produtos na área de micros pessoais. Entre esses produtos destaca-se o cartão CP/M para o CP-500. Também está em lançamento Comunicação Apple-IBM. Comunicação Apple-Apple e pacotes de automação de empresas.

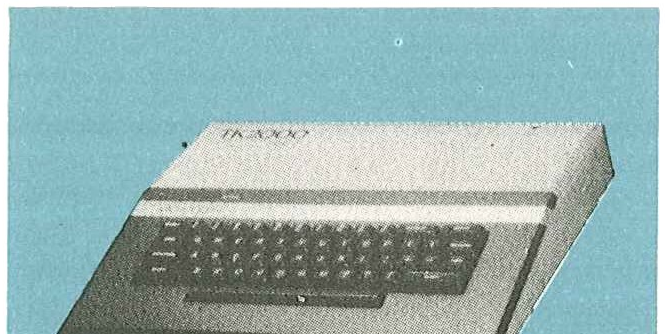
SERVIDOR DE ARQUIVOS

A Cetus está lançando o Servidor de Arquivos CS-1200, o primeiro de uma série de produtos que prestarão suporte especializado aos usuários ligados à rede local. O CS-1200 é um modo especial capaz de gerenciar até quatro unidades de disco rígido de 5 ou 10 Mbytes permitindo o compartilhamento físico e lógico de memória de massa entre diversos úsuários da rede Cetus. Utilizando discos de tecnologia Winchester pode-se expandir a capacidade de memória de um determinado sistema. O CS-1200 realiza o gerenciamento completo do espaço em disco, dividindo-o em volumes lógicos de diferentes tamanhos. Oferece rotinas de comunicação que permitem a criação, remoção e bloqueio (lock) dos referidos volumes por parte de usuários de diferentes sistemas. A rede Cetus possibilita ligar computadores de diferentes fabricantes.

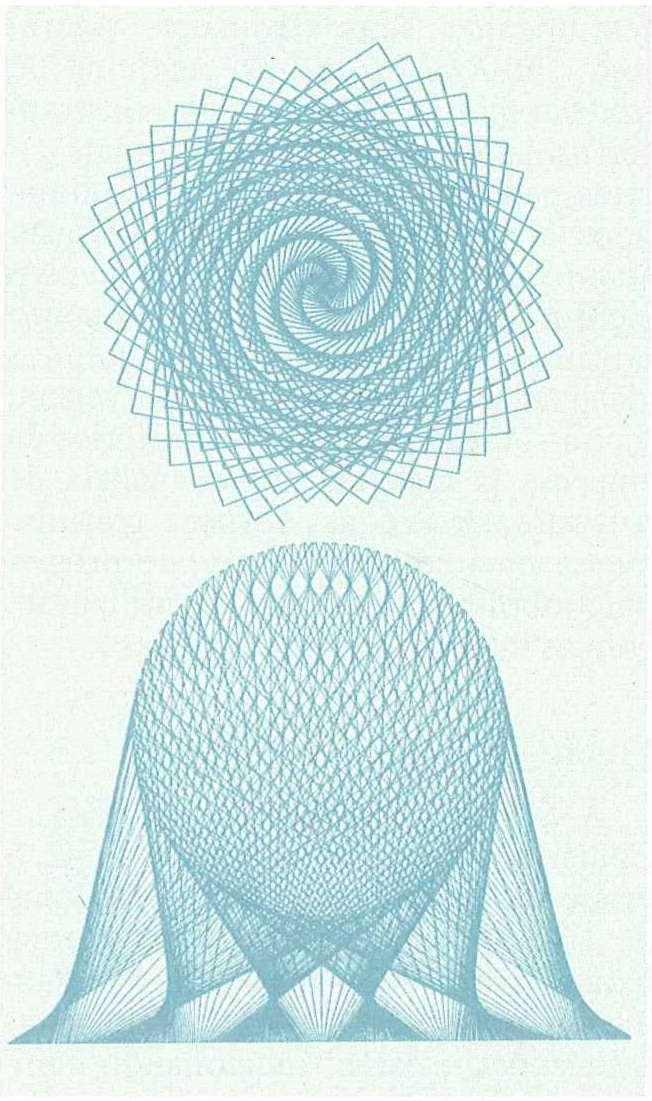
NOVO MICRO DA DISMAC

A Dismac lançou o microcomputador pessoal D-8100. Ele torna mais produtivo todo o trabalho de profissionais das mais diversas áreas, de pequenas empresas, de estudantes ou pesquisadores, pois além de ter operação simples é compacto. Utiliza o microprocessador 6502, capacidade de memória RAM de 48 Kbytes expandível, capacidade de memória ROM de 12 kbytes expandível, teclado de 53 teclas, saída para monitor de vídeo, saída para TV em cores com 15 cores programáveis ou P&B.

O TK 2000 - COLOR



A Microdigital está lançando o TK 2000 – COLOR, seu novo microcomputador, para servir pequenas empresas, profissionais liberais, pessoas que gostem de computador. Tem um microprocessador 6502, memória de 16 Kbytes ROM e 64 Kbytes RAM, display gráfico de baixa e alta resolução, unidade de disquete para armazenamento de programas, além de conexão paralela para impressora. Seu teclado é composto de 54 teclas. O TK 2000 – COLOR permite a gravação e leitura de programas em formato APPLESOFT.



SURGE A PLOTTER

Plotter – Traçador gráfico digital que representa uma evolução em desenhos por computador. A Plotter possui 12 comandos que permitem ao usuário fazer gráficos, fórmulas, mapas, blocos de letras, desenhos geométricos, desenhos técnicos, circuitos eletrônicos, etc. A Plotter, que pode ser acoplada a micros pessoais, micros profissionais do tipo Sistema 700 ou 600 e minicomputadores, executa funções complexas a uma alta velocidade, por ser um traçador inteligente controlado por um microprocessador combinado a um poderoso programa fornecido pela Prológica. As características da plotter são as seguintes: velocidade de 2,5 polegadas por segundo; uma resolução de 0,005 polegadas; uma pena; mudança de pena manual; e interface serial RS 232-C. Além disso, possui também 93 caracteres (maiúsculos e minúsculos) e quatro ângulos de rotação.

68000 - o poderoso microprocessador de 16 bits

Na terceira — e também a última — parte de seu artigo, Cesar da Costa trata da estrutura dos barramentos e do sistema de interrupção do microprocessador 68000.

Estrutura do barramento – O 68000 é disponível em invólucro com 64 pinos, a pinagem é apresentada na figura 1, requer para sua alimentação uma simples fonte de 5 volts. O barramento de dados bidirecional, não é multiplexado e tem capacidade (tamanho) para 16 bits. O barramento de endereço tem capacidade para 24 bits, permitindo o processador acessar diretamente 2^{24} ou 16 megabytes de memória, Observe que "*A0*", o bit menos significativo do barramento de endereço, não é disponível na saída do "*chip*". Este bit é usado internamente em conjunção com o tamanho do dado especificado por cada instrução, para gerar os sinais $\overline{UDS}$ (UPPER DATA STROBE) e $\overline{LDS}$ (LOWER DATA STROBE). Este barramento é assíncrono, isto é, não é sincronizado com o clock. Em vez disto, usa um método de "*handshake*" para acesso ao barramento. O sinal $\overline{AS}$ (ADRESS STROBE) indica o início de um acesso a memória e o sinal $\overline{DTACK}$ (data transfer acknowledge) retorna da memória para sinalizar o fim de uma transferência de dados. O diagrama de temporização de um byte de um ciclo de leitura é apresentado na figura 2. No tempo 1, o endereço é colocado na linha de endereço (A1 – A23) e a linha R/$\overline{W}$ (read/write) é "*setada*" para apresentar uma leitura de memória. No tempo 2, o sinal $\overline{AS}$ (address strobe) muda de estado para indicar o início de um ciclo de barramento, e o sinal $\overline{LDS}$ (Lower data strobe) muda de estado para indicar que a leitura está nos oito bits de menor ordem do barramento de dados. A memória responde colocando os dados a serem lidos, nas oito linhas de dados de menor ordem (D0-D7), como mostra o tempo 3. Ao mesmo tempo, o sinal $\overline{DTACK}$ muda de estado para indicar que os dados estão disponíveis. O processador então armazena os dados no chip e muda o estado do sinal $\overline{AS}$ para indicar que o ciclo terminou, como indica o tempo 4. A vantagem deste esquema é que permite utilizar memórias com diferentes velocidades. Portanto, o processador pode acomodar uma larga variedade de memórias e periféricos, que podem estar completamente afastados do próprio processador.

Alguns sinais são fornecidos para permitir interfaceamento e sincronismo com os dispositivos periféricos do 68000. São eles:

- Enable (E) – é um sinal padrão para os dispositivos periféricos da família 68000.
- $\overline{VPA}$ (Valid peripheral address) – entrada que indica que o dispositivo ou região endereçada é um dispositivo da família 68000 e que a transferência de dados será sincronizada pelo sinal de "*enable*".
- $\overline{VMA}$ (Valid memory address) – saída usada para indicar aos dispositivos periféricos do 68000 que existe um endereço válido no barramento de endereço e o

Figura 1 - Pinagem e barramento do 68000

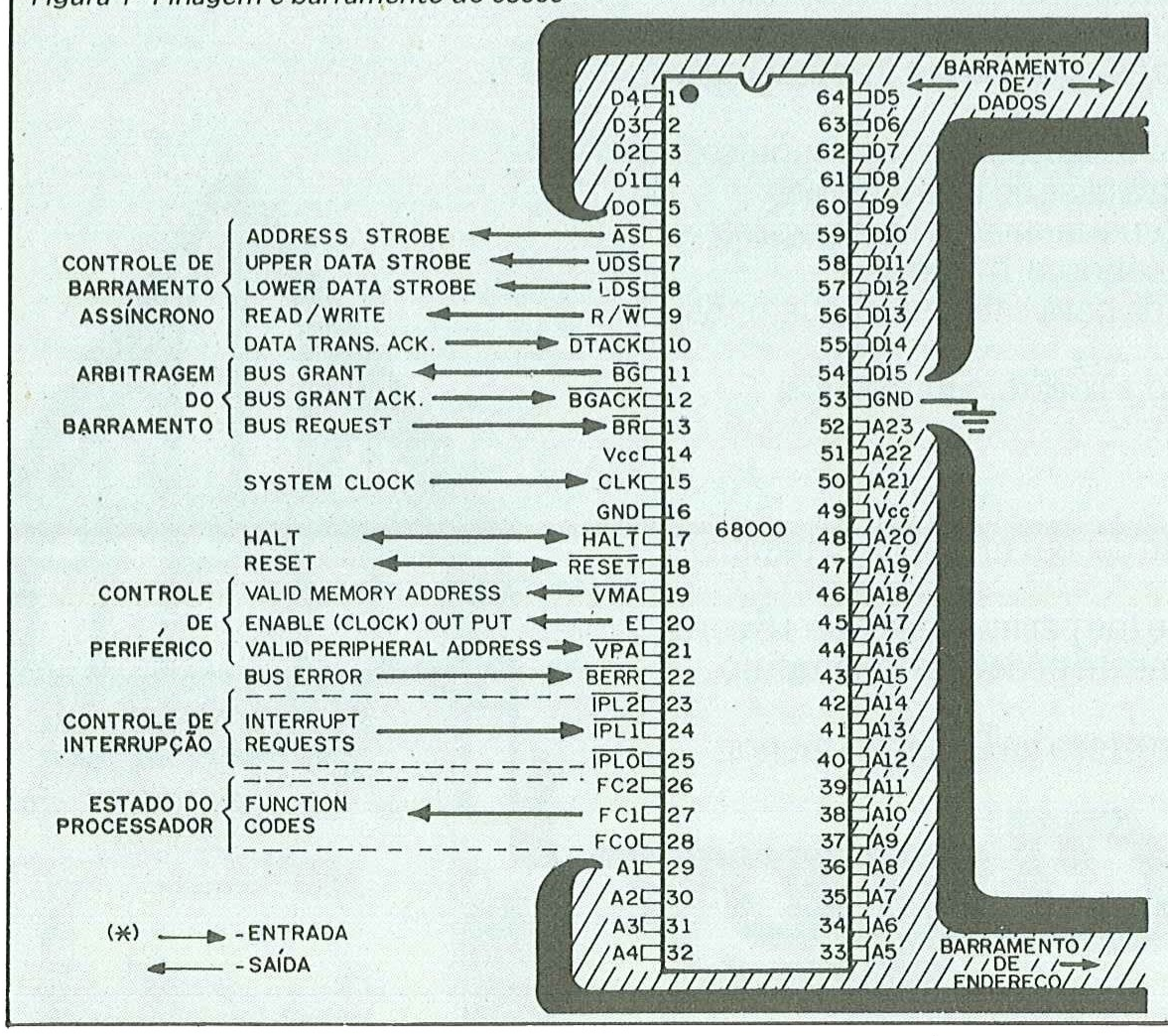
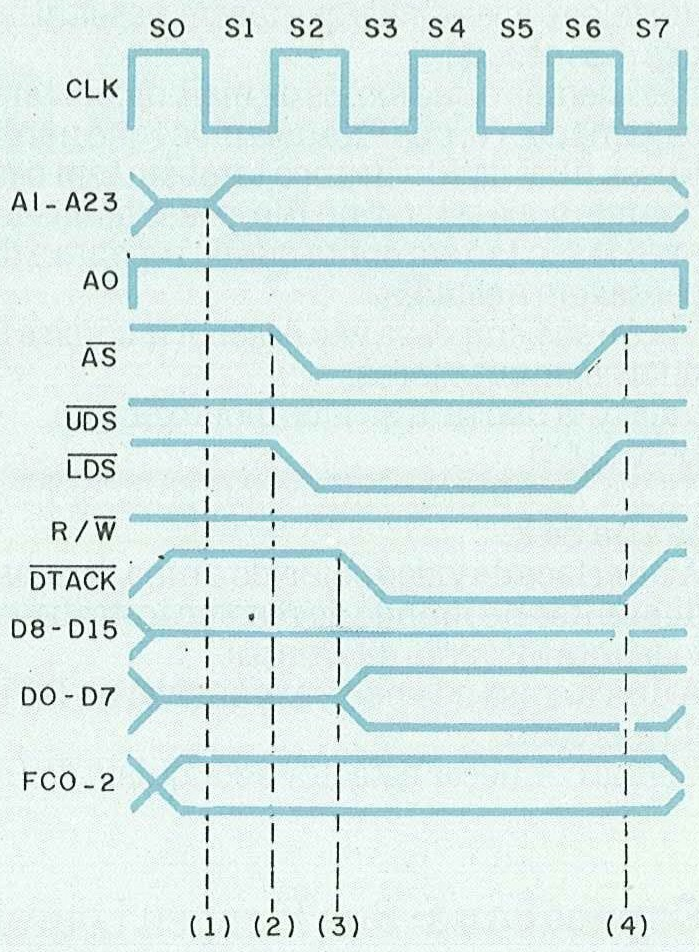


Figura 2 - Diagrama de tempo de um byte de ciclo de leitura.



processador está sincronizado pelo sinal *"enable"*. Este sinal somente responde a um sinal $\overline{VPA}$, o qual indica que o periférico é um dispositivo da família 68000.

Interrupção — Em aplicações de *"tempo-real"*, alguns recursos devem ser oferecidos para interrupção do processamento normal. Por exemplo, se uma impressora quer sinalizar ao processador que tem que imprimir uma linha de texto e necessita mais dados, o controlador da impressora transmitirá para o processador um sinal requisitando uma interrupção.

O 68000 tem sete níveis de prioridade de interrupção. A máscara de prioridade de interrupção no byte de supervisão do registro de estado identifica os níveis de interrupção; e pode ser programada para reconhecer qualquer nível e aqueles posteriores. A interrupção propriamente dita é obtida nas linhas $\overline{IPL0}$ – $\overline{IPL2}$. O processador parará a execução do programa corrente e lerá o vetor de interrupção do barramento de dados, numa operação chamada *"ciclo de reconhecimento de interrupção"*. O dispositivo periférico que requisita a interrupção coloca o número do vetor no barramento de dados. Este número será usado com um ponteiro na localização da memória. O processador lerá este endereço e iniciará a execução da rotina ali contida. Para retornar a execução do programa interrompido, a instrução RTE é executada.

Os microprocessadores de 16 bits estão sendo utilizados em um grande número de projetos porque comportam mais funções e possuem velocidade de operação muito maior que os microprocessadores convencionais de 8 bits. Podem endereçar diretamente 16 megabytes de memória, permitindo a utilização de programas extensos sem ter que esperar por acesso a disco. Esta facilidade abre as portas para aplicações gráficas de alta resolução e controle de processos.

Conclusão — Os microprocessadores de 16 bits estão sendo utilizados em um grande número de projetos, porque comportam mais funções e possuem velocidade de operação muito maior que os microprocessadores convencionais de 8 bits. Podem endereçar diretamente 16 megabytes de memória, permitindo a utilização de programas extensos sem ter que esperar por acesso a disco. Esta facilidade abre as portas para aplicações gráficas de alta resolução e controle de processos.

Os três maiores fabricantes do 68000 (Motorola, Mostek e Signetics) estão produzindo chips periféricos para o 68000.

Por exemplo, o Motorola MC68008 é uma versão de 8 bits do 68000; qualquer software escrito para este chip pode ser utilizado sem modificações no microprocessador 68000. Outra CPU típica de 16 bits é a Mostek MK68200 otimizada para aplicações de controle; contém 4 K bytes de ROM, 256 bytes de RAM e três temporizadores de 16 bits, podendo ser escrava do 68000.

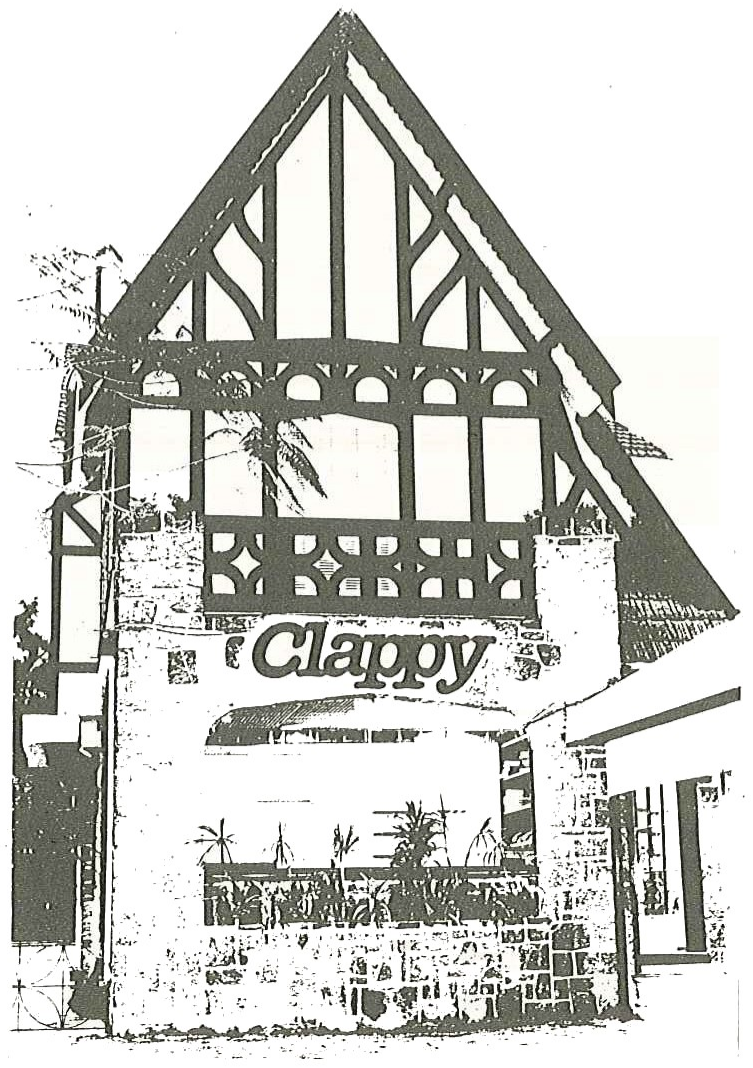
O MC68010 suporta memória virtual e máquina virtual, termo em tempos passados associado somente com *"mainframes"* (computador de grande porte).

No campo do software o 68000 tem mais de 40 distribuidores independente de pacotes. Em termos de linguagem de alto nível alguma coisa de *"FORTH"* a *"ADA"* é disponível. Em termos de sistema Operacional a Motorola dispõe do *"UNIDOS"*, a versão 7 do *"UNIX"*, sistema operacional compatível, e a DIGITAL RESEARCH está produzindo o CP/M-68K.

A melhor casa do Rio para

Os executivos que vêm ao Rio, principalmente a negócios, agora podem contar com uma casa que transforma sua rápida passagem pela cidade maravilhosa em momentos inesquecíveis. Em pleno coração de Copacabana, estamos de braços abertos e prontos para oferecer dos mais simples aos mais sofisticados modelos e acessórios que fazem nossa atividade tão excitante e tão imprescindível nos dias atuais.
Oferecemos o que existe de melhor, em termos de qualidade. E a preço e condições de pagamento (é, nós financiamos) que nenhuma outra casa do ramo oferece.
Nossa filial da Rio Branco também tem o mesmo atendimento e o mesmo preço.
Quando você estiver no Rio, passe bons momentos conosco. Nossos preços são tão em conta que de repente a diferença dá para cobrir seus custos de passagem e estadia.
Você e sua empresa vão descobrir como é fantástico, e barato, o mundo dos microcomputadores.
Veja esta oferta aí ao lado, por exemplo.
DGT-1000
Agora você tem um microcomputador modular com design moderno expansível de 16 a 64 k.
A interface de vídeo colorido é uma de suas opções, e lhe permite usar até 16 cores diferentes no mundo de resolução gráfica. Você também pode movimentar no vídeo até 32 áreas diferentes.
A única coisa que não é expansível no DGT-1000 é o preço na Clappy: o menor do mercado.
Venha conhecer esta nova conquista da DÍGITUS.

Clappy



Copacabana: Rua Pompeu Loureiro, 99
Centro: Av. Rio Branco, 12 ● loja e sobreloja ● Tels.: (021) 222-5517 ● 234-0214 ● 264-2096 ● 253-3395 ● 222-5721 ● 257-4398 ● 236-7175

A CERTEZA DE UM BOM NEGÓCIO

FAIRCHILD — **FAIRCHILD SEMICONDUTORES LTDA.**
Transístores, Diodos de Sinal e Zeners. . .

GENERAL SEMICONDUCTOR INDUSTRIES INC.
Transístores, Diodos Transzorb. . .

IBRAPE — **IBRAPE IND. BRAS. DE PRODUTOS ELETRÔNICOS E ELÉTRICOS LTDA.**
Transístores, Diodos de Sinal e Zeners. . .

ICOTRON — **ICOTRON S/A IND. DE COMPONENTES ELETRÔNICOS**
Transístores, Capacitores de Poliéster Metalizado e Eletrolítico. . .

MOTOROLA SEMICONDUCTOR PRODUCTS INC.
Transístores, Circuitos Integrados, Retificadores, Tiristores. . .

SOLID STATE SCIENTIFIC INC.
Transístores, Circuitos Integrados. . .

TECCOR — **TECCOR ELECTRONICS INC.**
Tiristores, DIACS, SCR, TRIACS. . .

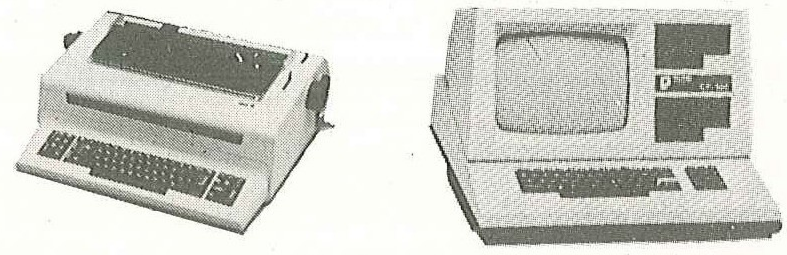
TELEDYNE SEMICONDCUTOR
Transístores, Diodos de Sinal e Zeners. . .

TEXAS INSTRUMENTS INC.
Transístores, Circuitos Integrados. . .

Kapraro

Teleimport / Eletrônica Ltda.
Rua Sta. Ifigênia, 402, 8/10º andar - CEP 01207 - São Paulo
Fone: 222-2122 - Telex (011) 24888 TLIM-BR
(Solicite nosso catálogo geral de componentes)

DIMERJ — A TECNOLOGIA SEM SOFISTICAÇÃO ANTES E DEPOIS DA COMPRA.

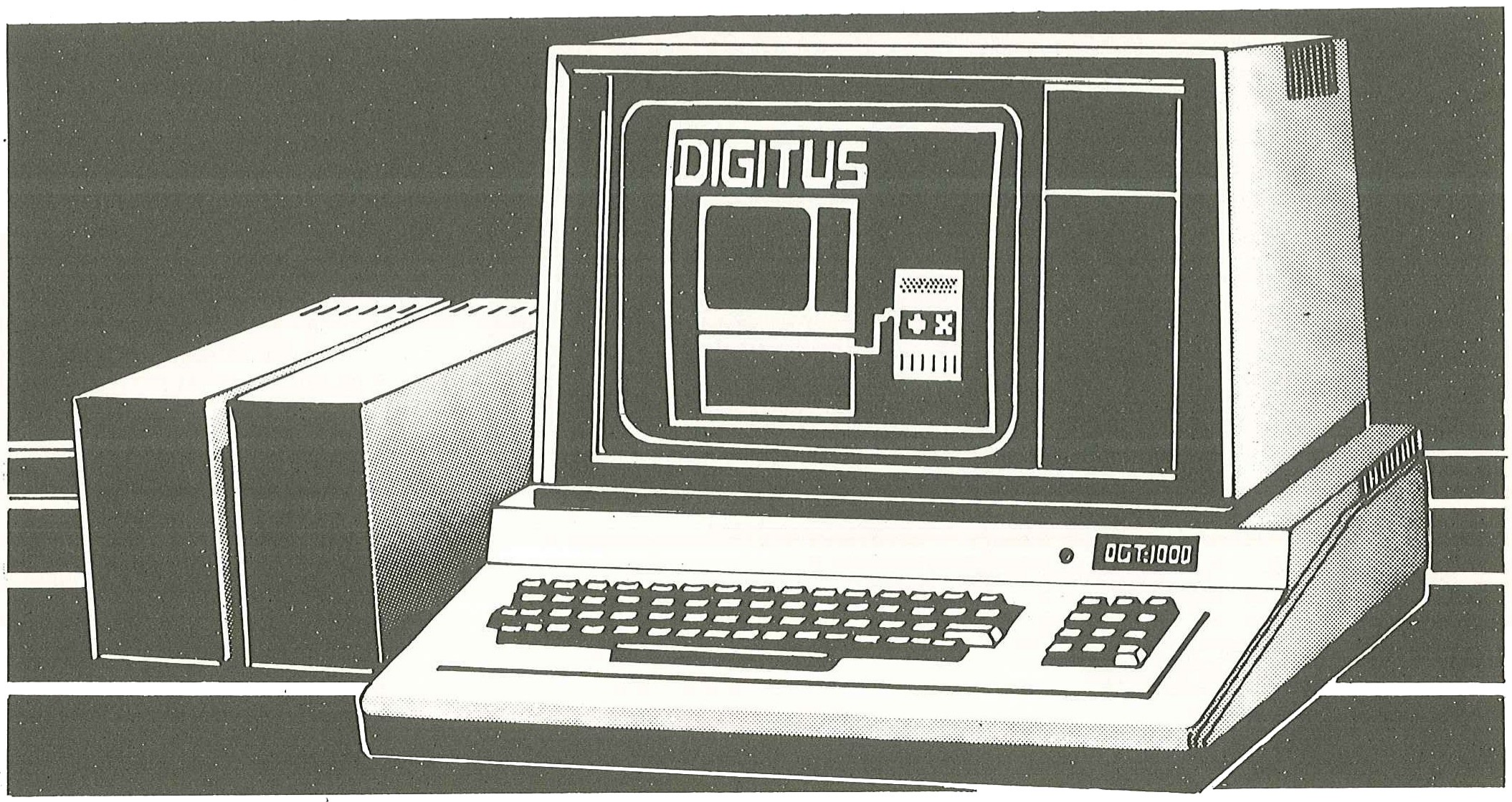


Agora o now-how e a sólida estrutura da DIMERJ no atendimento aos consumidores de máquinas e equipamentos FACIT estão à sua disposição também para venda, assistência técnica e manutenção dos micros da PROLÓGICA.
A DIMERJ tem programas de manutenção para manter os seus equipamentos com o máximo de rendimento e o mínimo de desgaste.
Os nossos técnicos são treinados na própria fábrica e o micro entregue aos nossos cuidados passa por uma cuidadosa revisão geral, com utilização dos mais modernos recursos.
Sem dúvidas, a DIMERJ possui a qualidade de serviços que a PROLÓGICA garante com alta tecnologia e sem sofisticação, antes e depois da compra.

DIMERJ produtos e serviços de qualidade — PROLOGICA microcomputadores

DIMERJ DISTRIBUIDORA DE MÁQUINAS DO RIO DE JANEIRO LTDA.
Av. Rodrigues Alves, 153 - Centro - Rio de Janeiro - CEP: 20081
Tel.: PABX 283.3522

a executivos de alto nível.



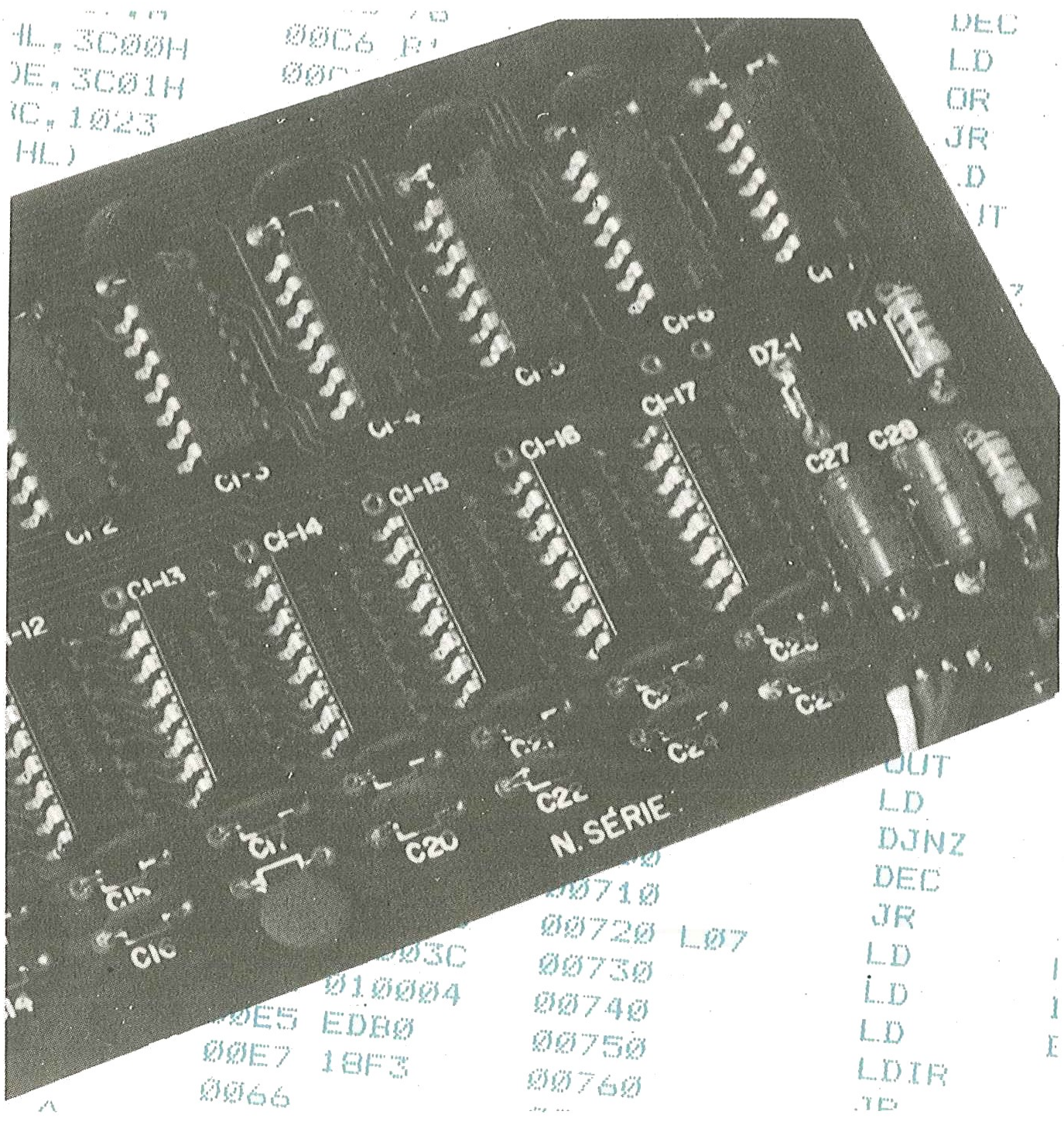
CPU com 64 k, placa CP/M, monitor, 2 drives, impressora Monica da Elebra.

DIGITUS

Entregamos em todo Brasil pelo reembolso Varig.
COPACABANA: Aberta diariamente das 10 às 20 horas e aos sábados das 10 às 15 hora

DESENVOLVIMENTO DE SOFTWARE PARA SISTEMAS BASEADOS EM MICROPROCESSADORES

Nesta segunda parte do artigo cuja primeira parte foi publicada na INTERFACE 14, mostra-se que as especificações do software não podem ser expressas clara e resumidamente numa linguagem natural. Uma correta e adequada especificação das exigências do software evita uma série de dores de cabeça já conhecidas de sistemas baseados em microcomputadores, desenvolvidos principalmente na área de telecomunicações.



O primeiro passo no desenvolvimento do projeto de um sistema baseado em microprocessadores, como havíamos dito na primeira parte deste artigo (INTERFACE número 14), é selecionar o microprocessador, que não é tarefa tão simples como possa parecer, pois as especificações do projeto (exigências) influenciam diretamente no hardware e no software do sistema. Assim sendo, quando um projetista seleciona um determinado microprocessador como base para o hardware do sistema, obrigatoriamente terá que examinar as funções críticas que serão implementadas no software. O desempenho dessas funções no microprocessador serão analisadas e comparadas com as especificações do projeto. Esta comparação, juntamente com a avaliação do custo, determinará a validade do projeto. Se o projeto não for válido o projetista deverá modificá-lo, porém se nenhuma alternativa em termos de projeto ajustar-se às especificações do sistema, estas deverão ser modificadas. Este aspecto iterativo de definição das especificações ajuda a identificar e prevenir erros antecipadamente em todas as etapas de projeto, quando ainda são menos dispendiosas.

PROJETO DO SOFTWARE

Após as exigências do software serem especificadas, as características são agrupadas logicamente num pequeno número de funções interdependentes com interfaces definidas. Este é o estágio do projeto de alto nível, onde as funções são decompostas de maneira hierárquica, juntamente com as descrições detalhadas das funções e suas interface. Este refinamento, de maneira prudente, conduz a um ponto de detalhamento que posteriormente facilitará a implementação em códigos.

O estágio de *"implementação"* propriamente dita, transforma o projeto em uma linguagem, que poderá ser traduzida eficientemente em linguagem de máquina. Tradicionalmente utiliza-se a linguagem assembler, onde um programador transcreve o projeto em passos, um a um, transformando-o num programa.

Após a implementação o software é testado. O teste envolve várias fases:

primeira-fase – depuração dos módulos de software individualmente, consiste na remoção de erros de linguagem de programação e erros de lógica, introduzidos no estágio de implementação.

Os erros de lógica são detectados pelo andamento do seu próprio código ou simulando o código

em um sistema de microcomputador (sistema desenvolvimental).

segunda fase – após o teste de módulos individuais, os módulos devem ser integrados em subsistemas de software, que também são testados por simulação.

terceira fase (última) – depuração por completo do sistema de software num sistema de microcomputador múltiplo. Note que esta fase de teste só poderá ser bem sucedida, se o projeto desenvolvido no estágio anterior estiver correto.

O estágio final do desenvolvimento de software é a "*manutenção*", que consiste em duas atividades distintas:

Correção de erros que não foram detectados durante o estágio de teste.

Otimização de software de modo que possa desempenhar novas funções, que mais tarde se tornará impossível, porque o acréscimo dessas funções exigirá novo interfaceamento, e se as interfaces não permitirem, então será necessário um novo projeto.

Muitos sistemas baseados em microprocessador foram desenvolvidos principalmente na área de telecomunicações, supervisão e controle de centrais telefônicas, com maior ênfase nas exigências de hardware, deixando as necessidades de software para fase posterior. O resultado foi catastrófico: 60% dos sistemas não atendem às exigências do software e 90% não atendem as suas funções críticas.

DEFINIÇÃO DAS EXIGÊNCIAS DE UM PROJETO DE SOFTWARE

Tem-se observado uma quantidade muito grande de sistemas baseados em microprocessador, que foram desenvolvidos no país principalmente na área de telecomunicações, supervisão e controle de centrais telefônicas, onde os projetistas deram maior ênfase às exigências do hardware, deixando as exigências de software para uma fase posterior. O resultado foi catastrófico: 60% dos sistemas não atendem às exigências do software e 90% não atendem às suas funções críticas.

Na maioria das vezes, esses problemas são resultantes de definições de exigências de software, posterior ao desenvolvimento do projeto. Uma definição incompleta força o projetista a fazer suposições e decisões que não são documentadas e freqüentemente diferem das especificações prometidas. Outro problema, que resulta da tentativa de não incorrer no primeiro, é especificação em excesso. Detalhes excessivos estendem desnecessariamente o projeto. Um documento de definição de exigência tem que ser completo, claro, consistente e testável. Novos instrumentos e técnicas foram desenvolvidas para dar suporte às especificações de exigências, uma vez que é difícil para descrições em linguagem natural exibir estas características.

Deixaremos para próximos artigos, onde mostraremos em detalhe, algumas técnicas e instrumentos para descrição de exigências de software.

A Nasajon Sistemas lança um novo conceito para agilizar a sua empresa :

A PRESSA É AMIGA DA PERFEIÇÃO.

A Nasajon Sistemas está lançando no mercado uma série de programas específicos que podem dinamizar ainda mais as diversas áreas de sua empresa.
São mais de 50 programas diferentes para DGT 1000, CP 500, D 8002, TRS 80, NAJA, JR e outros.
Com os programas da Nasajon Sistemas você vera porque a pressa é amiga da perfeição.

PREÇO ESPECIAL DE LANÇAMENTO		
Programa	Fita (Cr$)	Diskette(Cr$)
Contabilidade	40.000,00	239.000,00
Controle de Estoque	64.000,00	188.000,00
Mala Direta	47.000,00	109.000,00
Mala Direta c/ Ed. Texto	–	194.000,00
Contas a pagar/receber	47.000,00	116.000,00
Tesouraria (C. Saldo bancário)	–	116.000,00
Crediário (p/ D 8002)	–	129.000,00
Admin. de Imóveis	–	430.000,00
Editor de Texto	36.000,00	–
Arquivo de Processos	24.000,00	–
Controle de Livros	19.000,00	–
Controle de Cheques	19.000,00	–
Biorritimo	16.000,00	26.000,00
Decisão	17.000,00	27.000,00
Obstáculo	17.000,00	27.000,00
Kit Matemátic. c/6 progr.	62.000,00	72.000,00
Jogos Americ. (Fita c/4)	25.000,00	35.000,00

Você também encontra esses programas em nossos revendedores credenciados.

Nasajon Sistemas:
Um jeito fácil de resolver os problemas de seu computador.

- Desenvolvemos qualquer tipo de software de acordo com as necessidades de sua empresa.
- Antes de comprar seu computador solicite nossa assessoria, sem compromisso, para análise, implantação e apoio.
- Descontos para revenda.
- Atendimento por reembolso para todo Brasil.

Próximos lançamentos: Administração de Consultórios, Crediário II e Folha de Pagto. II.

HÁ

nasajon sistemas

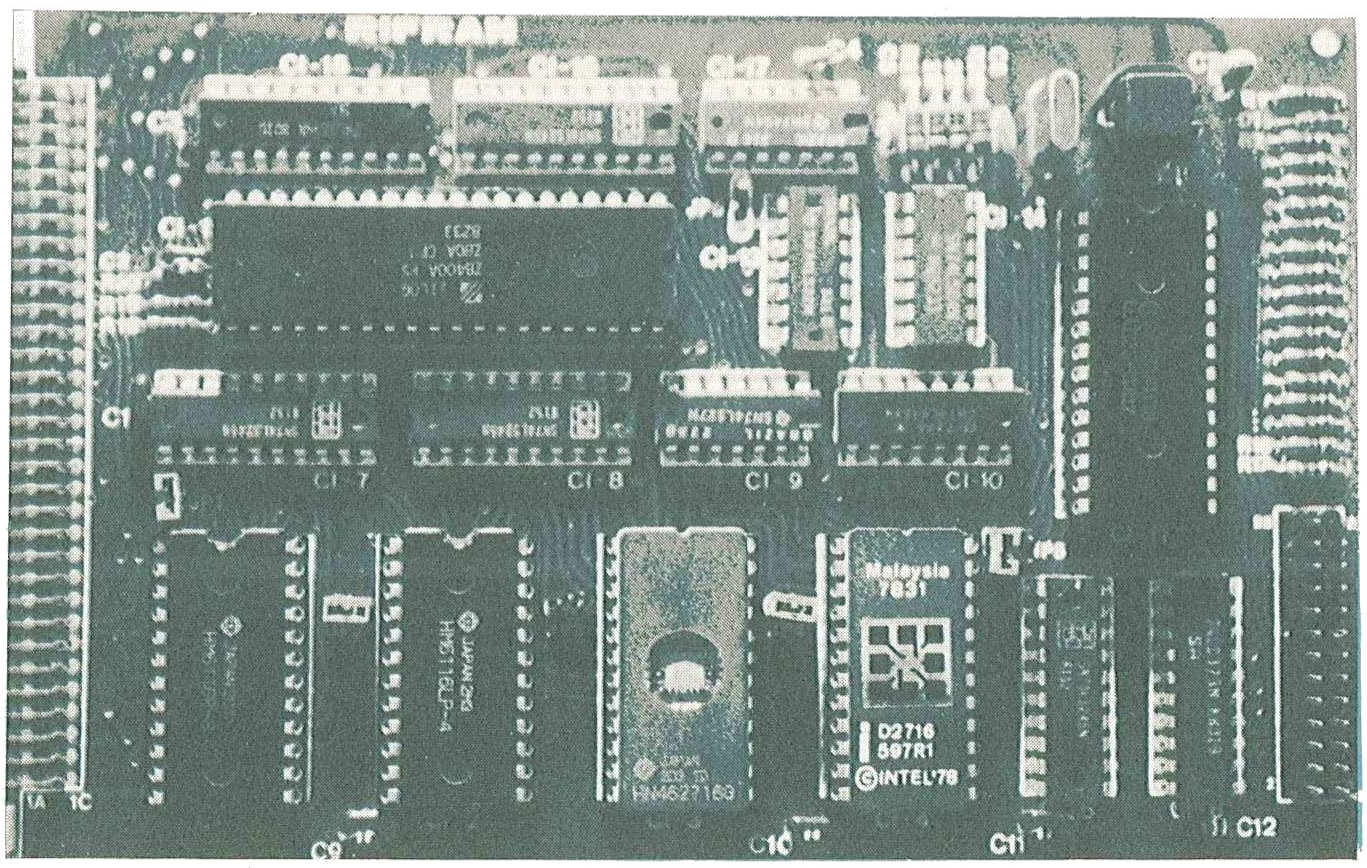
Av. Rio Branco, 45 gr. 1311 CEP 20090
Tel. (021) 263.1241 – Rio de Janeiro

DICAS DE SOFTWARE

Eduardo Augusto M. Andrade.

RESERVANDO ESPAÇO PARA ROTINAS EM CÓDIGO DE MÁQUINAS

Eduardo Augusto M. Andrade, engenheiro eletricista e professor de Basic que cursa Análise de Sistemas na U.F.M.G., conta como arranjar locais para armazenar rotinas de apoio.



Dentre os locais disponíveis para armazenar rotinas de apoio feitas em código de máquinas, o espaço após uma instrução REM no início do programa parece ser mais adequado. Isto porque as rotinas localizadas no final da memória (em área reservada através da variável do sistema RTP) simplesmente não são salvas pelo comando SAVE e o espaço reservado por uma instrução DIM, na área de variáveis, é flutuante na memória, podendo bagunçar tudo se for digitado RUN ou CLEAR acidentalmente.

Entretanto, ao se optar pela utilização de uma instrução REM vai-se deparar com um problema sério. Imagine-se por exemplo – como é o caso do programa simulador de vôo – ter que digitar 3.000 caracteres após a instrução REM. Nem precisa ser tanto, 500 caracteres já dá para esgotar a paciência de qualquer um.

Entretanto, existe um meio de burlar o monitor do TK e usar o comando EDIT para reproduzir a 1ª linha. Assim, no caso do simulador, basta digitar uma linha com 250 caracteres e reproduzi-la 12 vezes, utilizando o comando EDIT. Vamos ver como.

Ao digitar uma linha REM, o monitor a colocará na memória da seguinte maneira: usará 2 bytes para guardar o nº da linha, mais 2 bytes para o tamanho do texto, n bytes para o texto onde n é o tamanho do texto e finalmente um byte indicativo de final de linha (NEW LINE).

1	INVISÍVEL	REM XX. . . X (FINDA COM NEW LINE)

2 BYTES	2 BYTES	N BYTES (TEXTO)

Nº DA LINHA	TAMANHO DO TEXTO + NEW LINE	TEXTO COM N BYTES + + NEW LINE

Sabemos também que a área de programa inicia na posição 165Ø 9. Logo, se a instrução REM for a primeira linha do programa, os 2 bytes que indicam o tamanho do texto estarão na posição 16511 e 16512 (lembre-se que primeiro vem o byte menos significativo).

Como exemplo digite:

1 REM XXXX 4 X

Agora digite PRINT 256 * PEEK 16512 + PEEK 16511 e você obterá como resposta 6. Por que?

Agora duplique a linha 1 usando o comando EDIT e digite POKE 16511, 16. A linha 2 não existe mais, pois ela passou a fazer parte da linha 1. Tente mover o cursor para a linha 2, e verá que não irá conseguir. Por que 16? Simples. Como a linha 2 é igual a 1, ela terá os 6 bytes do texto mais os 4 de tamanho do texto e número da linha, perfazendo então um total de 1Ø bytes. Esses 10 bytes somados aos 6 bytes da linha 1 darão os 16 bytes, que será o comprimento da linha 1.

Este processo poderá ser usado para quantas linhas se queira copiar. Entretanto, NUNCA digite mais de 25Ø caracteres em uma linha antes de copiá-la usando o comando EDIT. Abaixo temos os passos a serem seguidos para se chegar ao resultado desejado.

1. LET A = PEEK 16511 * N + 4 * (N – 1)
   (N é o total de linhas REM. A que você digitou mais as que copiou utilizando EDIT).
2. LET B = INT (A/256)
3. LET C = INT (256 * (A/256-INT (A/256)))
4. POKE 16512,B
5. POKE 16511,C

Após ter executado os passos acima, você poderia editar a linha 1 e acrescentar ou retirar alguns caracteres.

CARTAS

SAUDAÇÕES GOIANAS

Em primeiro lugar, meus parabéns, os mais efusivos e sinceros votos de felicidades pelo transcurso do 1º ano da nossa revista INTERFACE. Digo nossa porque, como assinante, realmente me sinto orgulhoso em pertencer a tão brilhante família.

Informo que estou tentando formar um modesto clube, onde alguns hobistas possam tomar uma cervejinha bem gelada e trocar idéias sobre informática. E desde já nos colocamos à disposição de vocês todos.

Em seguida, um abraço especial a Tarcísio Cunha. O seu artigo "A Crise Econômica e a Reserva de Mercado" é um *senhor trabalho*. Analisei a exposição comedida de um drama que causa uma revolta incontrolável na maioria dos brasileiros esclarecidos.

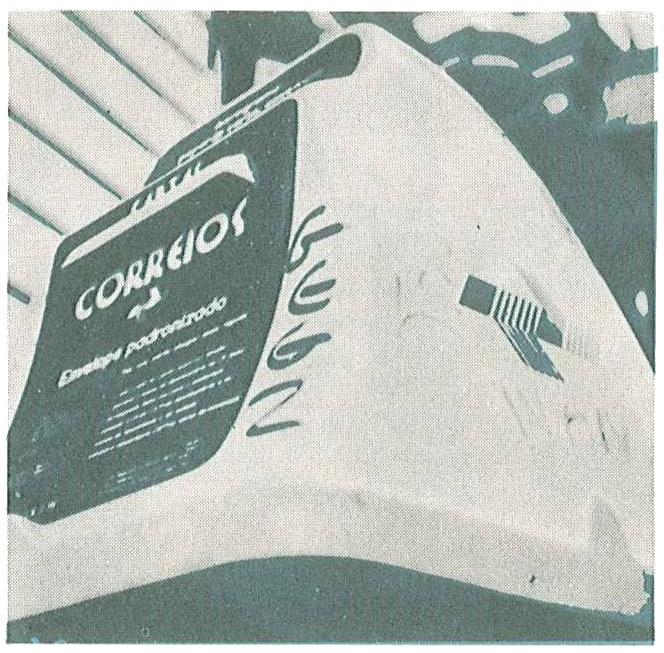
Ontem, 16 de novembro, o comentarista econômico Joelmir Beting abordou as tentativas, de leve, da desestatização da Petrobrás, em favor das multinacionais, face ao desenvolvimento do nosso Próalcool, que deu certo. Ao final, o comentarista disse que não havia motivos para preocupações etc. Pois bem, só em tocar nesse assunto das forças alienígenas tentando abocanhar o que de fato e de direito é nosso, me enerva tanto que, passo mal. Se houvesse um plebiscito, as urnas diriam "Não" a qualquer concessão, quer no campo do Petróleo, da Informática, até palito de fósforo nas mãos estrangeiras. A resposta do povo seria "Não". Alguém diria: Com que certeza eu digo isso? Respondo: Por que no Brasil só há eleições livres quando as elites dominantes deixam? Justamente quando não vão atrapalhar muito os interesses estrangeiros. O povo brasileiro tem uma intuição mística tão aprimorada que as elites econômicas jamais conseguirão resolver qualquer problema sem o aval mental, emanado pelo magnetismo positivo do povo sofrido. Um povo se conquista com amor. Um povo prefere ser mau governado por quem ele gosta do que ser mau governado por quem *não gosta*.

O artigo de Tarcísio é um esplendor à defesa da reserva de mercado para a nossa indústria de informática. Não tão contundente, mas sente-se em todo o trabalho o cuidado na colocação dos termos corretos e o rebuscamento de uma terminologia de alta qualidade literária. Creio que Tarcísio gostaria de esbravejar, gritar, brigar em defesa da nossa indústria de informática. Sinto isso devido ao excesso de comedimento em suas colocações primorosas. A obra é de alta literatura. Posso até estar errado, mas ainda creio ser Tarcísio um grande gladiador esperando o momento oportuno.

Meu temperamento é explosivo. Embora assim o seja, respeito profundamente a maneira de ser das palavras de Tarcísio, com seu estilo sério e clássico, pois também tenho que reconhecer que INTERFACE não é uma arena romana, e colocações explosivas poderiam prejudicar a bela imagem da revista.

Por tudo isso, obrigado a toda a equipe, obrigado especial ao Tarcísio Cunha. Obrigado ao José Arthur da Rocha pelo Curso Basic; se não fosse você eu estaria perdido. Caí num curso "picareta" que dá dó.

Obrigado ao André Gil e ao Ney Acyr pelo Z-80. Como sou neófito, estou na 2ª lição (ainda), mas creiam — estou empolgado.



Os subordinados são o espelho dos chefes. A hierarquia é a base fundamental de toda força, seja militar, religiosa ou política. Até na espiritualidade existe hierarquia, tudo com amor e respeito mútuo. O bom timoneiro leva o barco a salvo através das tempestades. Parabéns a essa Direção de INTERFACE, com Naila e Cesar, que estão conseguindo atravessar as intempéries de um ano duro, mas vitoriosos.

Wiles Amaral
Goiânia — GO.

Você chega a dar saudades das corruptelas *e da Vila Boa velha de guerra, das conversas do Bar Central em Goiânia, e desse jeitão brasileiro de ser texano. O seu exercício de tiro ao alvo vai publicado, com dois lembretes: é preciso saber fazer para poder querer e o lindo gado que se cria no campo também faz fila, mas nem por isso escapa do curral e do matadouro. Junto com o abraço vai a lembrança que depois do infravermelho e dos fuzis de quatro quilômetros de alcance, o saber fazer ficou mais importante ainda. Remember Malvinas.*

TEMA NÃO CONCLUÍDO

Gostaria de parabenizá-los pelo excelente trabalho desenvolvido em todos os números até agora lançados, com uma organização primorosa de assuntos, que são abordados de forma muito sucinta por outras revistas do gênero, mas que ganham um aspecto mais detalhado, sem economia de palavras ou mesmo de ilustrações, o que permite uma visão mais ampla do assunto, colocando desse modo a revista INTERFACE numa posição privilegiada.

Em apenas um exemplar, refiro-me ao número 08, encontrei uma falha clamorosa, se me permitem assim dizer, no tocante ao assunto "Teleprocessamento", que sumiu desse exemplar. O tema, que é de meu interesse e que acompanhei no que pude com os exemplares que tenho, é citado no exemplar número 07 do seguinte modo: "Serviço Internacional de Comunicação de Dados" (1ª Parte). E o assunto termina da seguinte forma: "No próximo número (o número 08) abordaremos os meios de acesso ao serviço Interdata, através da Rede Pública de Telex, Rede Transdata e Renpac", o que não veio a acontecer pois o assunto desapareceu sem explicação.

Mario Sergio O. de Jesus
Salvador — BA

Realmente, Mário, inúmeros problemas impediram a continuação do tema em questão. Entretanto, você pode aguardar, pois muito em breve publicaremos a conclusão. Agradecemos em nome de toda a equipe suas amáveis palavras.

INTERFERÊNCIA PROVEITOSA

Com referência a minha carta de 20 de outubro de 1983, comunico-lhes que fui atendido pela DIGITUS quanto ao programa PROTEX, que não funcionava direito em meu DGT-100, comprovando-se defeito na fita por mim adquirida.

Ainda não sei se foi tomada alguma providência quanto ao envio do número 10 de INTERFACE, não recebido por mim nos meios normais (sou assinante).

Agradeço sua interferência junto a DIGITUS, que me atendeu muito bem.

Pedro Paulo Luz Cunha Filho
Brasília — DF

Muito nos alegra o fato de termos podido ajudá-lo. Acreditamos que já tenha recebido o exemplar da nº 10. Conte conosco.

FUTURO ASSINANTE

Venho através desta parabenizá-los pela excelente revista que é INTERFACE e desejar-lhes muito sucesso nos outros números e nos outros anos de sua existência.

Tomei conhecimento da existência da INTERFACE lá pelo número 3, mas acontece que "passaram a mão" na minha coleção e sou obrigado a pedir a vocês todos os números que faltam, ou seja: 1, 2, 3, 4, 5, 6, 7, 8 e o 9. Quero receber pelo reembolso postal e, se puderem, o mais breve possível.

Gostaria que os senhores colocassem na revista a data em que ela foi feita, para se evitar tais distorções, e se possível agilizassem os meios de entrega da revista para não chegar tão atrasada assim, motivo pelo qual ainda não assinei esta brilhante revista.

Alexandre Vieira Machado
Belo Horizonte – MG

Agradecemos os seus parabéns e estimamos que já tenham chegado às suas mãos os exemplares que solicita. Quanto ao mês na capa e à distribuição, estão sendo providenciados.

NOTÁVEL QUALIDADE

Queiram receber meus cumprimentos por sua excelente revista INTERFACE. Todos os assuntos são de qualidade notável. Como assinante, apenas acuso o não recebimento do nº 12, até o presente momento. Peço que seja providenciado, caso já não o tenham feito.

Flavio Arnaldo
São Leopoldo – RS

Agradecemos suas palavras e esperamos que você já tenha recebido seu exemplar de nossa edição de aniversário.

COMPARÁVEL AO INTERNACIONAL

Desejo parabenizá-los pela publicação de uma revista séria, objetiva e com um elevado nível técnico. De fato, INTERFACE pode ser considerada a melhor revista nacional no gênero, e é comparável à publicação internacional.

Gostaria de sugerir-lhes a continuação do artigo "Conheça o Hardware do seu Micro", desta vez para a linha Sinclair (CP-200 e TK-85). . .

Herman Poliakoff
Belo Horizonte – MG

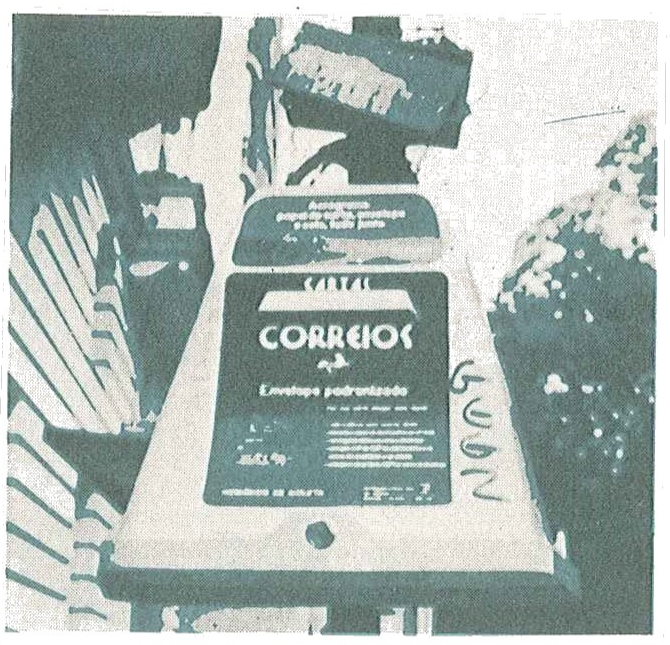
Obrigado, Herman Poliakoff. A família agradece e vai fazer um esforço para merecer seus elogios cada vez mais.

MANUAIS PARA SOFTWARE

Venho fazer uma reclamação: a de que vários programas existentes no mercado não possuem manual de instruções para, ao menos, mostrar como funciona um certo programa e, como todos nós sabemos, ninguém nasce sabendo. Programas tais como o *Monitor e Disassembler* da Microdigital, que deveriam, obrigatoriamente, ter um manual, já que o programa não tem instruções, não deveriam ser colocados no mercado, pois ninguém sabe como operá-los. Aí fica o meu apelo aos fabricantes de software. Aproveito a oportunidade para externar minha satisfação com o trabalho que INTERFACE vem nos mostrando a cada número, e desejo sinceros votos de sucesso.

Eduardo Medeiros
Ribeirão Preto – SP

Você está certo, Eduardo. Os programas existem para ajudar e não para atrapalhar.



A ELEITA

Diante de diversas edições sobre computação elegi a INTERFACE e mais uma outra, as de melhor e mais completo conteúdo sobre o assunto, sendo bastante objetivas. Por este motivo resolvi colecionar INTERFACE, fazendo assim a solicitação, há algumas semanas, dos números que me faltavam. Juntamente seguiu cheque nominal, conforme indicações, mas até o presente momento não as obtive.

Gostaria do parecer dos senhores quanto à questão, seguindo novamente em anexo a cópia da referida solicitação.

Carlos Roberto do Nascimento
Belo Horizonte – MG

Esperamos que o problema já tenha sido sanado; as providências já foram tomadas. Agradecemos em nome de toda a equipe as suas palavras.

COMPROMISSO COM A FELICIDADE

Assíduo leitor desta revista, venho trazer algumas sugestões e os meus votos para seu corpo editorial de "que em 1984 seu maior compromisso seja com a felicidade". Quanto às sugestões são: um ou mais artigos sobre software de apoio à programação (compiladores, assemblers, assemblers cruzados (Z80-8080) e uma análise sobre *add-on's* ou modificações de hardware que começam a ser mais freqüentes no mercado (teclados, circuitos leitores de eprom's, alta resolução gráfica, conversores AD/DA etc.).

Estas sugestões se aplicam principalmente aos micros da linha TK onde, após um pequeno investimento inicial, o usuário pode chegar a um sistema previamente poderoso, de uma forma gradativa, sem ter que trocar de computador.

João Carlos Santiago Filho
Rio de Janeiro – RJ

Obrigado pela mensagem de carinho. E que possamos ter como lema o compromisso com a felicidade *a que você alude. Suas sugestões foram ótimas e foram anotadas.*

FELIZ NATAL COM LED-METER

Gostaria de pedir-lhes uma cópia xerox (ou semelhante) do diagrama completo do LED-METER publicado na página 31, da revista número 11, pois meu exemplar foi extraviado. Meu melhor muito obrigado e votos de Feliz Natal!

Roberto de Araújo Santos
Rio de Janeiro – RJ

Já foi enviada a xerox da página 31, bem como da errata do LED-METER. Agradecemos e retribuimos seus votos.

COMANDOS NUMÉRICOS

Gostaria de parabenizar este magnífico mensário INTERFACE pelos esclarecimentos e conhecimentos obtidos, querendo lembrar o seguinte: *quem está com a informática está com o futuro garantido.* Aproveito o ensejo desejando que continuem cada vez melhor e, se possível, alguns artigos referentes a controle numérico; seria uma publicação inédita (ou não concordam?).

Angelo da Silva
Valinhos – SP

Muito obrigado pelos elogios. A sugestão já foi para a agenda.

BASIC – 6.ª LIÇÃO

JOSÉ ARTHUR DA ROCHA

EM BUSCA DE DEZ DEZENAS PARA A LOTO

José Arthur da Rocha prossegue, nesta 6.ª lição, o que já tinha começado na 5.ª: gerar um palpite de cinco dezenas para o jogo da Loto. Depois também faz um programa para gerar um único palpite de dez dezenas.

Na lição anterior apresentamos o comando READ que, muitas aplicações, facilita a entrada de dados num programa. Vimos também que dados poderiam ser criados dentro do próprio programa, aleatoriamente, utilizando a função RND. Você observou como esta função é implementada no seu microcomputador?

O exercício proposto na lição anterior (gerar um palpite de 5 dezenas para o jogo da LOTO será resolvido agora.

A solução do problema envolve os seguintes passos:

1. gerar uma dezena entre 0 e 99
2. imprimir esta dezena
3. testar se já foram impressos 5 dezenas; se não, voltar ao início

Como gerar um número dentro de um determinado intervalo foi um dos assuntos abordados na última lição.

Vimos que a função RND gera um número no intervalo [0,1], isto é, um número maior que 0 (zero) e menor ou igual a 1 (um). Para aumentar este intervalo até [0,99] devemos inicialmente "multiplicá-lo" por 100. Em outras palavras, todo número fracionário gerado será multiplicado por 100. Se pegarmos somente a parte inteira deste número teremos o resultado no intervalo [0,100], onde existem apenas os números inteiros de 1 a 100 inclusive. Se subtrairmos uma unidade dos números gerados excluiremos o 100 e incluiremos o 0 (zero) entre os números possíveis, isto é, o intervalo passa a ser [0,99], conforme o desejado.

O algorítimo solução do problema é apresentado no fluxograma da figura 1.

O programa em BASIC que implementa o fluxograma da figura 1 é apresentado a seguir.

```
10 REM ** PALPITE DA LOTO**
20 RANDOMIZE
30 C = 1
40 N = RND
45 D = INT (N* 100) – 1
50 PRINT D;"     " ;
60 F C = 5 THEN GOTO 90
70 C = C + 1
80 GOTO 40
90 PRINT
100 INPUT "NOVO JOGO? (SIM/NÃO) ; X$
110 IF X$ = "NÃO" THEN STOP ELSE GOTO 20
```

O algoritmo e o programa em BASIC poderá gerar inúmeros jogos de 5 dezenas para o LOTO, bastando para isso que responda à pergunta final "NOVO JOGO?" com qualquer coisa diferente de "NÃO".

Façamos algumas observações no programa. As linhas 40 e 45 podem ser reduzidas a uma só linha pois a função END pode fazer parte de uma expressão. A linha 45 sumiria e a 40 seria:

40 D = INT (RND * 100) – 1

Observem que na linha 50 temos a impressão da variável D (dezena "sorteada") e em seguida um ponto e vírgula (;) seguido de uma STRING composta por 1 ou mais espaços, e mais um ponto e vírgula. Com isto as dezenas serão escritas uma ao lado da outra, separadas por 1 ou mais espaços. Recordamos assim a função do ponto e vírgula no comando PRINT.

Para que a impressão que seguirá a quinta dezena (a impressão seguinte é "NOVO JOGO? (SIM/NÃO)") não seja colocado ao lado daquela dezena, é utilizado o PRINT vazio da linha 90. Como depois do PRINT nada existe, s programa simplesmente aponta como local da próxima impressão a linha seguinte.

No programa "PALPITE DA LOTO" a variável C "conta" quantas dezenas já foram sorteadas e nela é feito o teste se este número já chegou a 5. E observem que, ao se pedir um novo jogo, seu valor retorna a 1 para que novamente sejam sorteadas outras 5 dezenas. A ação de se realizar uma determinada função ou processo, composto de uma ou mais instrução um determinado número de vezes chama-se processo ou fluxo controlado. O controle se baseia em termos uma variável, chamada variável de controle, que tem um valor inicial (vi) e será somada a um incremento t toda vez que o processo for realizado, até que a variável de controle seja maior que o valor final (vf).

No programa "PALPITE DA LOTO", a variável de controle do fluxo que gera e imprime uma dezena é o C. Sem valor inicial (v_i) é 1, seu incremento (t) é 1 e seu valor final (v_f) é 5.

Poderíamos resumir processo controlado no fluxograma da figura 2.

Os processos ou fluxos controlados (em inglês LOOPS) aparecem na maioria das funções que se quer implementar, em BASIC ou em outra linguagem qualquer. E suas aplicações e importância são tão grandes que em todas as linguagens existem comandos que facilitam a implementação destes "LOOPS": No BASIC temos o comando FOR.

COMANDO FOR

A função do comando FOR é exatamente aquela descrita pelo fluxograma da figura 2. Nele é definida a variável de controle, seus valores inicial e final e o incremento. Os valores v_i, v_f e t, respectivamente, são números reais. Podem aparecer também na forma de variável ou expressão. Cada

Figura 1 - Fluxograma para gerar 5 dezenas para o jogo da LOTO, repetidas vezes

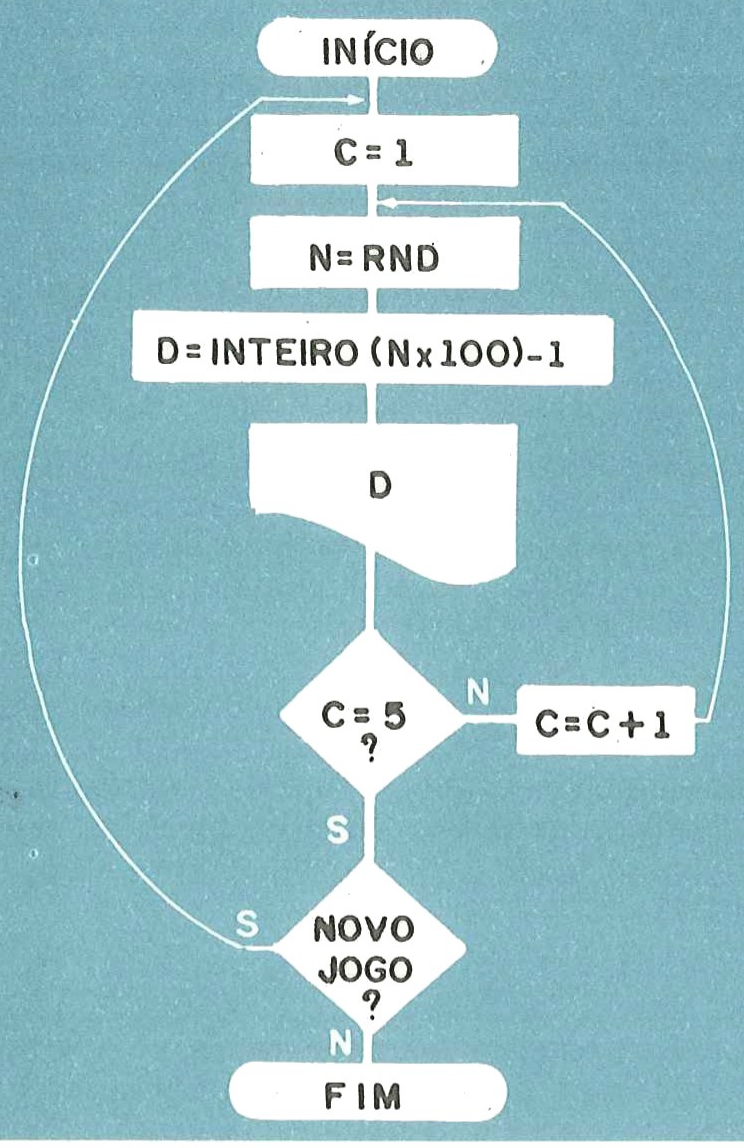


Figura 2 - Algoritmo geral de um fluxo controlado. Os pontos 1 e 2 podem aparecer em qualquer ponto do fluxograma principal.

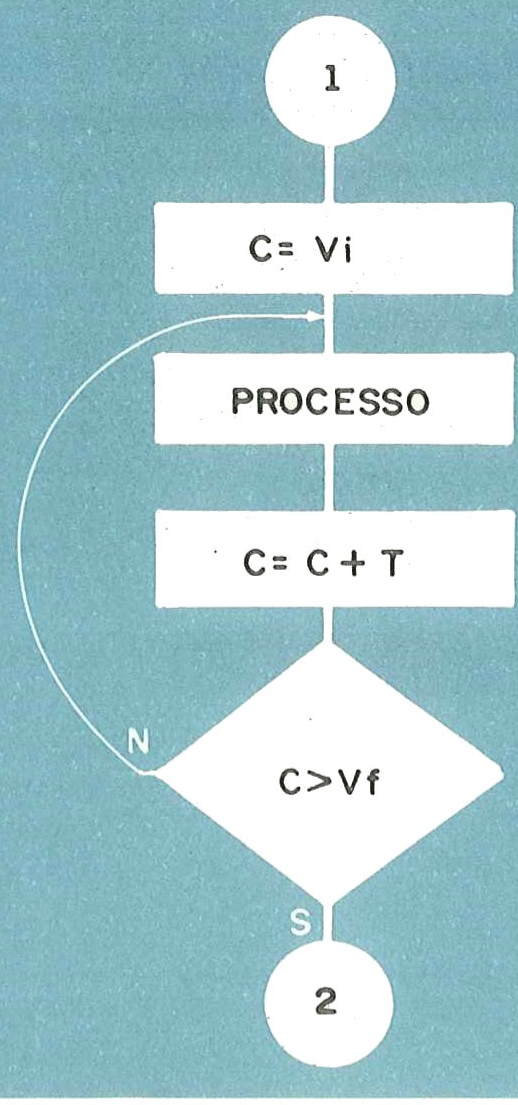
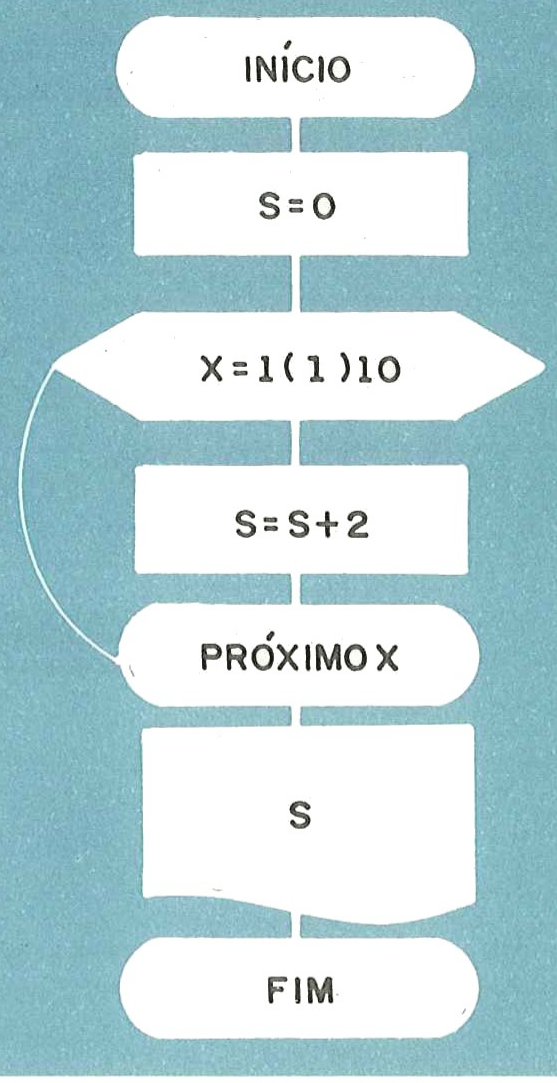


Figura 3 - Fluxograma e programa em Basic que calculam a soma dos 10 primeiros números pares, positivos.



fluxo tem uma única variável de controle.

O fluxograma da figura 2 diz que:
"para C variando de v_i a v_f com incremento de t, execute o processo ou, "resumidamente", para C = v_i até v_f com passo t" que em inglês daria a forma do comando FOR:

FOR C = V_I TO VF STEP T

PROCESSO

NEXT C

Observem que o processo é limitado pelo FOR e pelo comando NEXT, que pede o "máximo" valor para a variável de controle C, que será C = C + T se este valor de C ainda for menor ou igual a VF. Se T = 1 não é necessário escrever o STEP e o comando ficaria: FOR C = VI TO VF

Vejamos os exemplos abaixo:

Exemplo 1) Achar a soma dos 10 primeiros números pares inteiros, o algoritmo solução e o programa em BASIC estão na figura 3 Observem que a terceira caixa representa o controle do fluxo pela variável X, de valor inicial v_i = 1, incremento t = 1 e valor final v_f = 10

```
INÍCIO      10 REM TOMA 10 PRIMEIROS NÚMEROS
                PARES
S = 0       20 S = 0
            30 FOR X = 1 TO 10
X = 1 (1)10 40 S = S + 2
            50 NEXT X
S = S + 2   60 PRINTS
            70 STOP
PRÓXIMO X
     S
   FIM
```

Exemplo 2) Escrever na ordem decrescente todos os números inteiros positivos, múltiplos de 3, a partir de 27.

A solução está na figura 4.

```
INÍCIO      10 REM NÚMEROS MÚLTIPLOS DE 3
            20 FOR A = FOR A = 27 TO 0 STEP – 3
A = 27 (–3)0 30 PRINT A;"     " ;
            40 NEXT A
     A      50 STOP
PRÓXIMO A
   FIM
```

Para o programa da figura 4 aparecerá na tela como resposta:

27 24 21 18 15 12 9 6 3 0

Exemplo 3) Programa da LOTO para gerar um único palpite de 10 dezenas usando o comando FOR

```
10 REM ** PALPITE DA LOTO**
20 RANDOMIZE
30 FOR C = 1 TO 10
40 D = INT (RND * 100) – 1
50 PRINT D;"     " ;
60 NEXT C
70 STOP
```

Nos três exemplos acima vários programas que continham cada qual um único "loop". No entanto podemos ter dois ou mais fluxos dentro de um programa e ainda fluxos dentro de fluxos, cada qual com a sua variável de controle. Vejamos algumas regras de uso do comando FOR

1. Cada fluxo deve ter uma única variável de controle e a sua ação vai desde o FOR até o NEXT, até que a variável de controle seja maior que o valor final determinado.
2. Dentro do fluxo poderá haver um desvio para fora

Figura 4 - Fluxograma e programa Basic, solução do problema do exemplo 2.

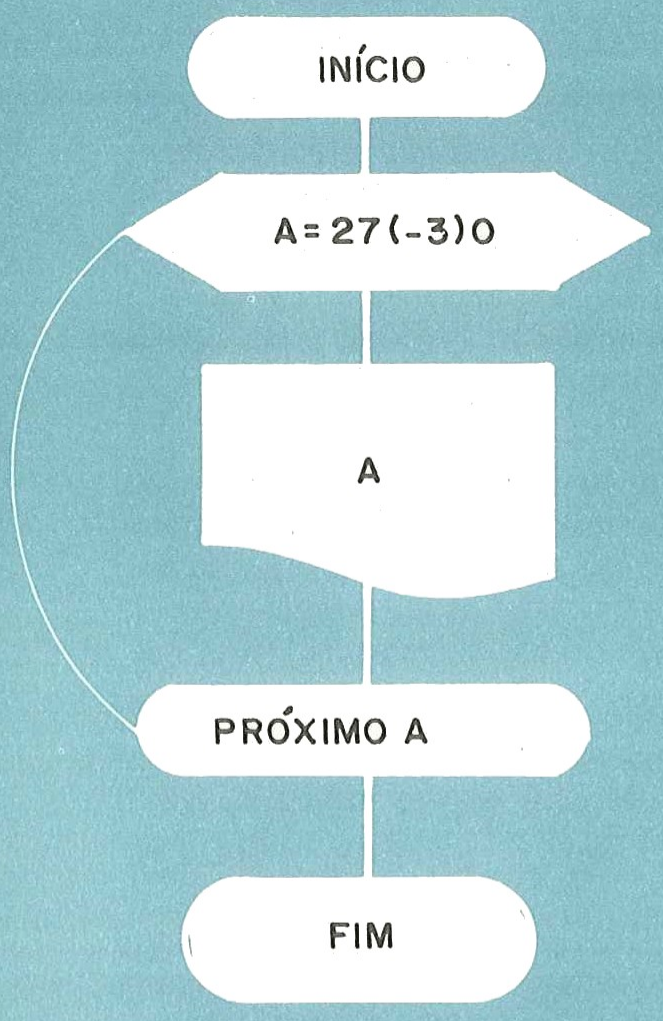


Figura 5 - Formas de utilização do comando FOR em fluxos compostos: a correta (a) e a proibida (b)

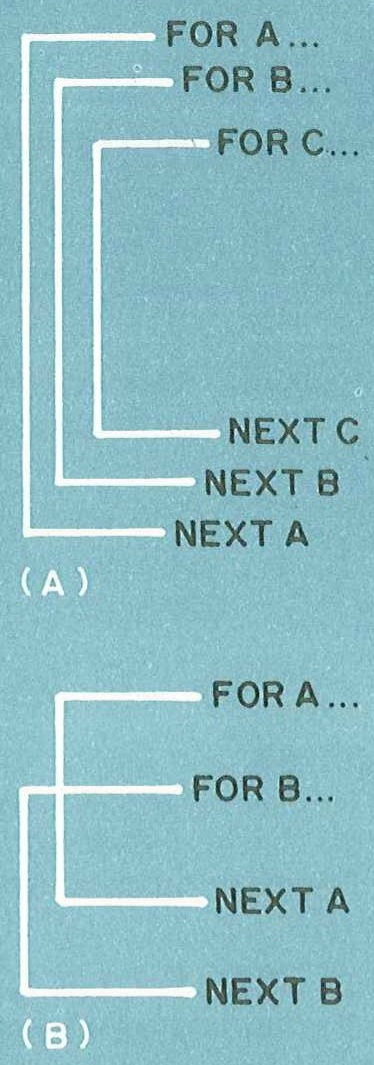
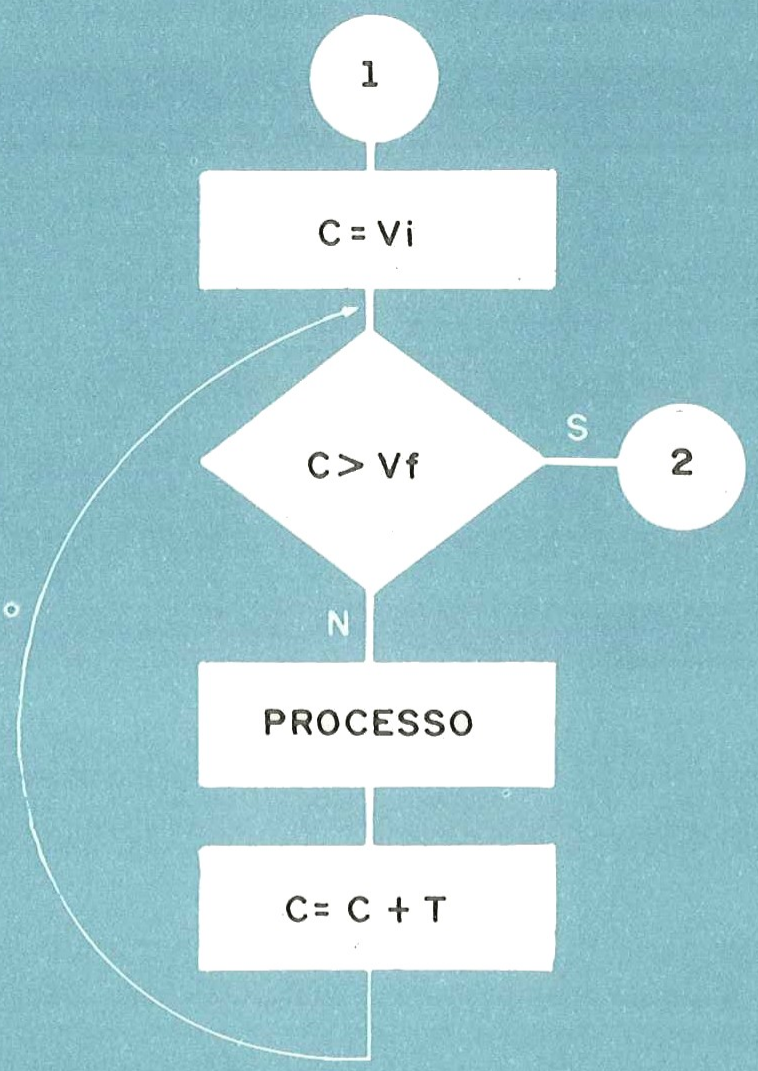


Figura 6 - Outro algoritmo que implementa o comando FOR



dele caso alguma condição de permanência no fluxo não seja cumprida. No entanto não poderá haver um desvio para dentro de um fluxo sem passar pelo comando FOR. Neste caso o computador acusaria erro ao encontrar o NEXT sem o respectivo comando FOR.

3. É possível a existência de fluxos compostos, isto é, fluxos dentro de fluxos, cada um com a sua variável de controle. Eles devem obedecer o diagrama apresentado na figura 5a, onde o primeiro fluxo a ser "aberto" deve ser o último a ser "fechado" e o último a ser "aberto" deve ser o primeiro a ser "fechado". A figura 5b mostra uma forma proibida de utilização do FOR onde o computador acusaria erro.

```
FOR A...          FOR A...
FOR B...          FOR B...
FOR C...
NEXT C            NEXT A
NEXT B            NEXT B
NEXT A
```

(a) (b)

SUMÁRIO

Vimos nesta lição um dos processos mais poderosos e de larga aplicação em programação: o controle de fluxos. No BASIC ele é implementado no comando FOR, que segue, quase sempre, o fluxograma da figura 2. Algumas versões do BASIC seguem o fluxograma da figura 6, cujo resultado é, na grande maioria das aplicações, o mesmo. Verifique no seu microcomputador como é um implementado o comando FOR (prob. 2).

EXERCÍCIOS

1) Modifique o programa da LOTO de maneira a possibilitar ao operador que ele diga quantas dezenas serão sorteadas por vez (5 a 10, segundo as regras da LOTO) e seja dado no final o valor da aposta, segundo uma tabela genérica.
2) Faça um programa que mostre qual dos dois algoritmos apresentados implementa o comando FOR em seu microcomputador.
3) Analise o programa abaixo e diga o que aparecerá no vídeo depois que ele for executado.

```
 10 REM PROGRAMA TESTE
 20 A = 1
 30 READ X,Y
 40 IF (X/Y) 2 OR Y X THEN GOTO 110
 50 IF X – Y 10 THEN GOTO 130
 60 FOR I = 1 TO (X – Y)
 70 PRINT "*";
 80 NEXT I
 90 PRINT Y
100 GOTO 140
110 PRINT "INVÁLIDO"
120 GOTO 140
130 PRINT "———"; X
140 A = A + 1
150 IF A = 5 THEN STOP ELSE GOTO 20
160 DATA 54, 40, 17, 12, 12, 17, 2, 1, 3, 1, 23, 3
```

FILIA

RILDO PRAGANA

FILIA-UM SISTEMA DE PROGRAMAÇÃO

Rildo Pragana, do Departamento de Física da Universidade Federal de Pernambuco (Recife) chega à 7.ª parte de seu artigo.

Esta parte do artigo nos mostrará um interpretador externo simplificado para Filia e alguns outros programas utilitários.

O Interpretador externo tem grande importância pela interface que realiza entre o sistema e o operador/programador. Um interpretador mal escrito significa dificuldade na hora de utilizar os programas disponíveis, ou insistentes "crashes" do sistema. É responsabilidade do interpretador externo a tradução dos nomes de dicionário dos átomos pelo seu endereço de execução. Suas rotinas básicas são descritas sumariamente abaixo:

★RECEBA – Faz a entrada de caracteres do terminal (ou teclado) ser bufferizada, e permite correção de erros datilográficos. Encerra a introdução quando recebe um "retorno de carro" (ENTER ou RETURN ou CR). Dependendo do nível de sofisticação desta rotina podemos ter repetição de linha rebatida, eco na limpeza de um caracter etc.

★ÁTOMO – Retira do buffer de recepção um "token" ou "átomo", ou seja, um conjunto de caracteres finalizados e/ou precedidos de delimitadores. Normalmente estes delimitadores são espaços mas podem ser modificados antes da chamada de ÁTOMO, para fins especiais. O Átomo é colocado logo no final do dicionário, sendo precedido pelo número de caracteres significativos (não contando com os delimitadores) e um espaço para a colocação do elo quando estivermos compilando.

★PROCURE – Verifica se o átomo existente no final do dicionário tem correspondente no próprio dicionário. Devolve um flag indicando "sim" ou "não" e caso o átomo seja encontrado devolverá também na pilha o endereço do átomo.

★ABORTE – Reinicializa as pilhas, esquece o último átomo definido se abortou durante uma compilação e retorna o controle ao interpretador EXTERNO. Esse programa é usado quando um átomo não é encontrado, quando um número não é possível na base corrente, ou em outras ocasiões de erro. O ABORTE também imprime a palavra contida no final do dicionário (sem fazer parte ainda do mesmo), seguida por um símbolo de interrogação. Um exemplo de sua execução é quando digitamos algo errado, não reconhecível pelo sistema e este nos responde com o que digitamos seguido de '?'.

★NÚMERO – Verifica se o átomo existente no final do dicionário é um número válido na base corrente, devolvendo um flag e o valor do número quando for possível. O flag indica se o número é válido ou não (1 = válido).

★?PILHA – Função para testar se as pilhas estão dentro dos seus limites, retornando um flag.

★?COMPILÁVEL – Função que indica se um átomo dado pelo seu endereço é compilável ou imediato. Átomos imediatos são os que devem ser executados mesmo durante uma compilação. Para compilar estes átomos precisamos de um expediente especial, o uso do par de compiladores [e].

★EXEC – Função que causa a execução de um átomo dado seu endereço na pilha. Retorna para o programa que a chamou após essa execução.

★PROCESSO-INTEPRETADO – Função secundária que executa uma linha de código em fonte já introduzida no buffer de teclado ou de outro buffer qualquer, desde que tenha um ponteiro colocado na variável LEIT.

★EXTERNO – É o programa principal do interpretador externo. Normalmente é o programa que fica sempre executando numa instalação de Filia, a não ser quando sai para executar um programa do usuário. É o programa mais externo de todos e tem todos os outros programas do sistema como suas subrotinas, inclusive os do usuário.

★HEAD – Ponteiro para o final da lista ligada de átomos no dicionário. Já foi discutida em partes anteriores do artigo.

★DP – Ponteiro para a próxima posição livre do dicionário. Seu conteúdo é devolvido também pela função AQUI, devido à sua freqüente utilização em outros programas.

★LEIT – Variável usada como apontador para a leitura e também para a escrita nos buffers do sistema. Usada por ÁTOMO para ler o próximo átomo existente no buffer de teclado.

★RBUFF – Buffer de teclado. Normalmente de tamanho um pouco maior que uma linha na tela de vídeo. Local onde são armazenados os caracteres digitados.

★ESTADO – Variável de sistema (flag) que indica se o estado corrente é de compilação, isto é, se estamos compilando, ou de execução. Controla o funcionamento do interpretador externo.

★DELIM – Variável (byte) que contém o delimitador corrente usado na separação de átomos. É modificada por ÁTOMO no final da execução, retornando ao valor "default" que é espaço.

★VAZIO – Flag de sistema que indica se o buffer de recepção (RBUFF) se encontra vazio.

Agora passaremos à implementação das rotinas acima. Algumas delas são escritas diretamente em assembly devido à grande velocidade de interpretação conseguida. Em muitos casos é tolerável ser substituída por uma implementação em Filia (alto-nível), especialmente se o dicionário está quase vazio ou o sistema tem pouca utilização a nível de interpretador externo. Para desenvolvimento de programas contudo, a velocidade é uma das virtudes que diferenciam Filia da maioria das outras linguagens e devemos preservá-la.

Iniciaremos com uma discussão de ÁTOMO. Esta rotina deve retirar caracteres do buffer de recepção um a um, analisando-os até encontrar algo diferente de um delimitador corrente. Nesse ponto são transferidos os caracteres do buffer para AQUI + 3 (no final do dicionário) e contados quantos caracteres são transferidos até ser encontrado um outro delimitador. Assim ÁTOMO quebra a linha introduzida em partes que devem ser processadas independentemente. Nos compiladores tradicionais essa operação é conhecida como "parsing" ou análise sintática. Em Filia nenhum erro normalmente é analisado nesta fase, por simplicidade, apesar de que não seria difícil introduzir um interpretador externo mais específico com esta função.

Daremos uma implementação em assembly de ÁTOMO, usando o assembler residente do sistema.

```
CODE ÁTOMO
LEIT LHLD       (carregue em HL um ponteiro para o buffer de leitura)
XCHG            (deixe-o em DE)
DP LHLD         (agora em H1 um ponteiro para o final do dicionário)
H INX H INX     (+ 2 para ficar um espaço para um possível elo)
B PUSH          (salve o BC que é o pseudo-PC do FILIA)
H PUSH          (salve também o ponteiro para o local)
                (. . . aonde ficará o tamanho do "token")
H INX           (aponte H1 para o local onde serão transferidos os caracteres)
8Ø B MVI        (tamanho máximo de um "token")
DELIM LDA       (tome o delimitador corrente)
AC MOV          (. . . e deixe-o em C)
INÍCIO,         (laço para a procura de algo diferente de um delimitador)
D LDAX          (leia o primeiro caracter)
D INX           (avance para o próximo)
C CMP           (é delimitador?)
NZ/ ATE,        (até encontrar um caracter diferente)
(até aqui poderiam existir múltiplos espaços ou outros delimitadores)
INÍCIO,         (laço de transferência do átomo)
A M MOV         (mova o primeiro caracter não delimitador)
H INX           (avance o ponteiro para o final do dicionário)
D LDAX          (deixe em A o próximo caracter do buffer)
B DCR           (conte se estourou o tamanho máximo permitido)
Z/SE,           (estourou então finalize assim mesmo)
AQUI TROQ       (observe que isto introduz um endereço na pilha)
                (. . . em outras palavras, não é código estruturado)
C M MOV         (ponha um delimitador no final do átomo)
XCHG
LEIT SHLD       (retorne o ponteiro de leitura no buffer atualizado)

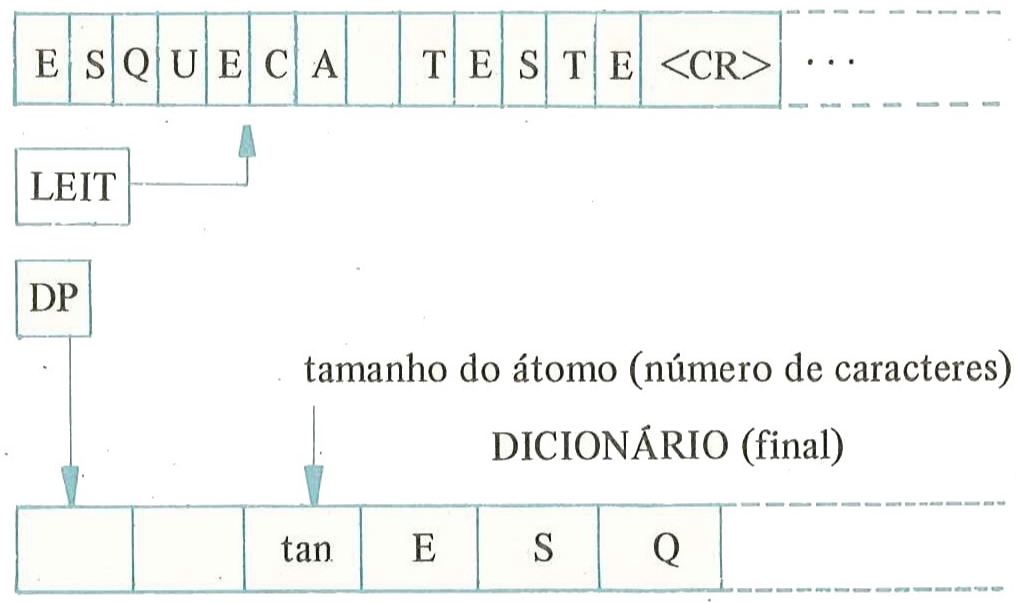
8Ø A MVI
B SUB           (calcule o tamanho do átomo lido)
H POP
A M MOV
B POP
32 A MVI        (código ASCII de espaço)
DELIM STA       (será o novo delimitador)
PROX JMP        (retorne ao interpretador interno)
                (é o fim de ÁTOMO)

ENTÃO,

TROQ            (finalidade: restaurar a ordem do INÍCIO . . . ATÉ,)
D INX           (avança na leitura do buffer)
C CMP           (é outro delimitador no fim?)
Z/ATÉ,
JMP             (esse JMP sem endereço vai saltar para o endereço deixado na pilha logo após o SE')
```

OPERAÇÃO DE "ÁTOMO"

BUFFER DE RECEPÇÃO



A segunda rotina a ser mostrada, PROCURE, também é normalmente codificada em linguagem de máquina devido à velocidade necessária. Talvez seja esta rotina do interpretador filia a que precise ser mais rápida, pois sua operação fica mais lenta a cada vez que introduzimos uma nova definição no dicionário. A implementação corrente de Filia presume o dicionário como uma lista ligada linear, ou seja, com todos os átomos numa única cadeia. Isto torna a pesquisa de um átomo muito demorada, se não tomarmos os devidos cuidados. PROCURE assume o átomo a ser pesquisado já na posição padrão deixada por ÁTOMO, para facilitar o trabalho do interpretador externo. O final da cadeia (lista) do dicionário está na variável HEAD.

Vejamos a codificação de PROCURE:

```
CODE PROCURE
B PUSH          (salve o pseudo-PC temporariamente)
HEAD LHLD       (toma o ponteiro para o final da lista de átomos)
INÍCIO, H PUSH  (guarda o ponteiro sem modificações)
XCHG
DP LHLD         (toma o ponteiro p/o dicionário)
XCHG
D INX           ("AQUI"+2 é onde está o tamanho do átomo)
D INX
M A MOV         (verifica se elo = Ø, isto é, final da lista)
```

```
H INX
M ORA
Z/ SE,
H POP B POP Ø H LXI H PUSH PROX JMP
                    (termina a pesquisa devolvendo que
                    não encontrou pois chegou ao final
                    do dicionário)
ENTÃO,
H INX               (passa a apontar para o tamanho do
                    átomo no dicionário)
D LDAX
M SUB
7F ANI              (verifica se tamanhos coincidem, exce-
                    to flag de IMEDiato)
Z/ SE,
D LDAX 5 CPI        (se tamanho maior que 4)
P/ SE, 4 A MVI
ENTÃO,              (force 4, que é o máximo)
A B MOV             (use o registrador B como contador)
INÍCIO,
H INX D INX D LDAX M CMP
                    (compare um dos caracteres)
AQUI 1+ Ø NZ/ JC    (prepare um salto se NZero deixando o
                    endereço deste salto na pilha)
B DCR               (conte até o fim)
TROQ                (evita o endereço de salto deixado na
                    pilha)
Z/ ATÉ,             (repita enquanto o contador não for
                    zero)
H POP B POP         (restaure ponteiro da lista e também o
                    BC)
7 D LXI D DAD       (avance 7 para ter o endereço do des-
                    critor do átomo)
H PUSH              (devolva esse descritor)
1 H LXI H PUSH      (devolva um flag dizendo que achou o
                    átomo)
PROX JMP            (. . . e retorne)
AQUI TROQ =         (aquele salto acima deverá saltar para
                    cá)
ENTÃO,
H POP               (restaure o ponteiro para a lista)
M E MOV H INX M D MOV XCHG
                    (leia o elo para o próximo da lista)
JMP                 (volte ao INÍCIO, do laço)
```

(observe que INÍCIO, é usado apenas para deixar um endereço)

(o código acïma é não-estruturado e deve ser analisado com cuidado)

ABORTE usa na sua definição algumas funções em assembly e uma rotina de impressão que pode ser facilmente modificada. As funções são #CARACTERES, que devolve o tamanho do token lido por ÁTOMO, que se encontra em "AQUI + 2"; e a função ØPILHAS que reinicializa as pilhas e devolve o controle para o interpretador EXTERNO.

```
CODE #CARACTERES
DP LHLD H INX H INX
                    ("aqui + 2" contém o número de ca-
                    racteres)
M L MOV Ø H MVI H PUSH PROX JMP
```

Esta rotina poderia ser codificada já em Filia sem prejuízo de velocidade, devido à sua pequena utilização:

```
: #CARACTERES AQUI 2 + B @
```

A segunda rotina, ØPILHAS, deve reinicializar as pilhas.

```
CODE ØPILHAS SPØ SP LXI (SPØ é a posição original da
                    pilha de usuário)
SSØ H LXI SS SHLD (SSØ é a posição original da segunda
                    pilha)
EXTERNO B LXI       (carrega pseudo-PC com o início do
                    EXTERNO)
PROX JMP
```

Agora vejamos a implementação de ABORTE:

```
: ABORTE #CARACTERES
                    (toma o número de caracteres do últi-
                    mo "token")
AQUI 3 +            (local onde o token errado se encon-
                    tra, no dicionário)
TYPE                (rotina em baixo-nível para imprimir
                    n caracteres)
." ?"               (imprima uma interrogação em seguida
                    ao nome)
Ø ESTADO B =        (faça o ESTADO ser execução, e não
                    compilação)
(aqui poderiam estar outras reinicializações necessárias)
Ø PILHAS ;          (reinicializa as pilhas devolvendo o
                    controle ao interpretador externo)
```

Dependendo do sistema implantado, o ABORTE poderia ter um comportamento diferente, mas sempre deve se ter em mente deixar como função desta rotina a restauração das variáveis mais importantes do sistema após um erro fatal. Assim, num sistema com pilha de strings, por exemplo, seria reinicializado o ponteiro desta pilha; num sistema com memória virtual, seriam verificados os buffers do disco, e assim por diante.

No interpretador externo existem também a rotina NÚMERO, que identifica números válidos na base corrente e empilha-os se for o caso. Para bases numéricas até o valor 1Ø (decimal), os algarismos arábicos são suficientes para codificar qualquer constante numérica, porém no caso de base maiores que dez, teremos que usar como símbolos os caracteres do alfabeto, AZ, para bases até o valor 36, quando então a quantidade disponível de símbolos torna confusa a entrada de números. A implementação de NÚMERO dada é em linguagem assembly, como usual para programas que desejamos alta velocidade, mas poderá ser facilmente recodificada em Filia para se tornar mais compacta.

```
CODE NÃO-DIGITO (programa auxiliar para saída com
                    erro)
D POP B POP Ø H LXI H PUSH PROX JMP
                    (restaura registradores e retorna com
                    um zero na pilha, indicando número
                    não válido)
CODE NÚMERO B PUSH (salve o pseudo PC)
DP LHLD             (tome o ponteiro para o dicionário)
H INX H INX         (mais dois para obter o número de ca-
                    racteres do átomo)
M B MOV             (guarde o tamanho em B)
H INX               (aponte para a string numérica, presu-
                    mivelmente)
Ø D LXI             (formaremos o número, se válido, em
                    DE)
M A MOV             (leia um caracter)
20 CPI              (verifique se é o sinal ", para números
                    negativos)
Z/ SE, PSW PUSH     (salve os flags indicando este fato)
H INX B DCR         (atualize o ponteiro e o contador)
SENÃO, PSW PUSH     (salve apenas o flag, o primeiro carac-
                    ter é um dígito)
ENTÃO,
BASE LDA A C MOV (carregue a BASE numérica em C)
INÍCIO, H PUSH      (salve HL para efetuar uma multipli-
                    cação)
C A MOV             (obtenha a base)
Ø H LXI             (inicialmente h1 = Ø)
INÍCIO,             (multiplique dígito em E pela base)
RAR                 (obtém o bit mais significativo do mul-
```

```
                tiplicador)
C/ SE,          (se houve carry)
D DAD           (produto = produto-multiplicando)
ENTÃO,
XCHG H DAD XCHG
                (duplique o multiplicando)
A ANA           (teste se o multiplicador é zero)
Z/ ATÉ,
D POP           (restaure o HL salvo no início do laço como DE)
D LDAX          (leia um caracter do número)
3Ø SUI          (3Ø em hexadecimal é o código ascii do '0')
M/ SE,          (se for menor, não é dígito válido)
'NÃO-DÍGITO JMP (retorne com um zero na pilha)
ENTÃO,
ØA CPI P/ SE,   (se for maior que 9, então deve ser entre A e Z)
7 SUI ØA CPI M/SE, (se está entre '9' e 'A' também é errado)
NÃO-DÍGITO JMP
ENTÃO,
ENTÃO,
C CMP           (verifique se é maior ou igual à base)
P/ SE,          NÃO-DÍGITO JMP ENTÃO,
D PUSH          (salve o ponteiro temporariamente)
A E MOV         (ponha u dígito em E)
Ø D MVI         (com D igual a zero)
D DAD           (some número-parcial=número-parcial + dígito)
D POP XCGH      (volte o ponteiro para HL)

H INX B DCR     (avance para o próximo dígito, decremente contador)
Z/ ATÉ,         (repita o mesmo para todos os dígitos)
PSW POP         (verifique se o número continha um sinal " à frente)
Z/ SE,          (se tinha, complemente-o)
D A MOV CMA A D MOV E A MOV CMA A E MOV D INX
ENTÃO,
B POP           (restaure o pseudo PC)
D PUSH 1 H LXI H PUSH PROX JMP
                (retorne na pilha o número e um flag = 1 indicando que o número foi verificado com sucesso)
```

Analisaremos agora as rotinas mais externas do interpretador. Estas são mais facilmente entendidas e modificadas pelo próprio usuário para satisfazer as suas necessidades particulares.

A primeira delas, PROCESSO-INTERPRETADO, interpreta e executa (ou compila, conforme o caso) o conteúdo do buffer apontado por LEIT. É sua responsabilidade também evitar que a pilha sofra um "underflow", quando deverá ser impressa uma mensagem de estouro da pilha e abortada a execução da linha de comandos. A execução de PROCESSO-INTERPRETADO termina quando o buffer fica VAZIO.

PROCESSO-INTERPRETADO

```
INÍCIO
ÁTOMO PROCURE (verifique se é um átomo já definido)
SE DUP          (salve o ponteiro para o mesmo)
?COMPILÁVEL     (teste se é compilável ou imediato)
ESTADO B @ E/   (e o estado é compilação)
SE,             (lembre-se que ',' compila o topo da pilha)
SENÃO EXEC      (senão execute o átomo)
ENTÃO
SENÃO           (se não encontrou no dicionário corrente)
NÚMERO          (verifique se é um número, convertendo)
SE ESTADO B @   (está compilando?)
SE ' NÚMERO-COMPILADO 2,
                (deixe a função NÚMERO-COMPILADO)
                , (seguida pelo número)
ENTÃO           (se não estiver compilando, simplesmente. . .)
                (deixe o número na pilha, onde já está)
SENÃO APAG ABORTE
                (não é função nem número válido, erro)
ENTÃO
ENTÃO
?PILHA          (teste se a pilha está acima do mínimo)
SE ." ESTOUROU A PILHA " ABORTE (imprima mensagem de erro)
ENTÃO
VAZIO B @ ATÉ ; (repita enquanto o buffer contiver caracteres para serem processados, o que é indicado pelo flag VAZIO)
```

Finalmente a rotina principal do interpretador externo é EXTERNO:

```
: EXTERNO INÍCIO RECEBA PROCESSO-INTERPRETADO
Ø ATÉ           (repita indefinidamente) ;
```

IMPLEMENTAÇÃO DE 'RECEBA'

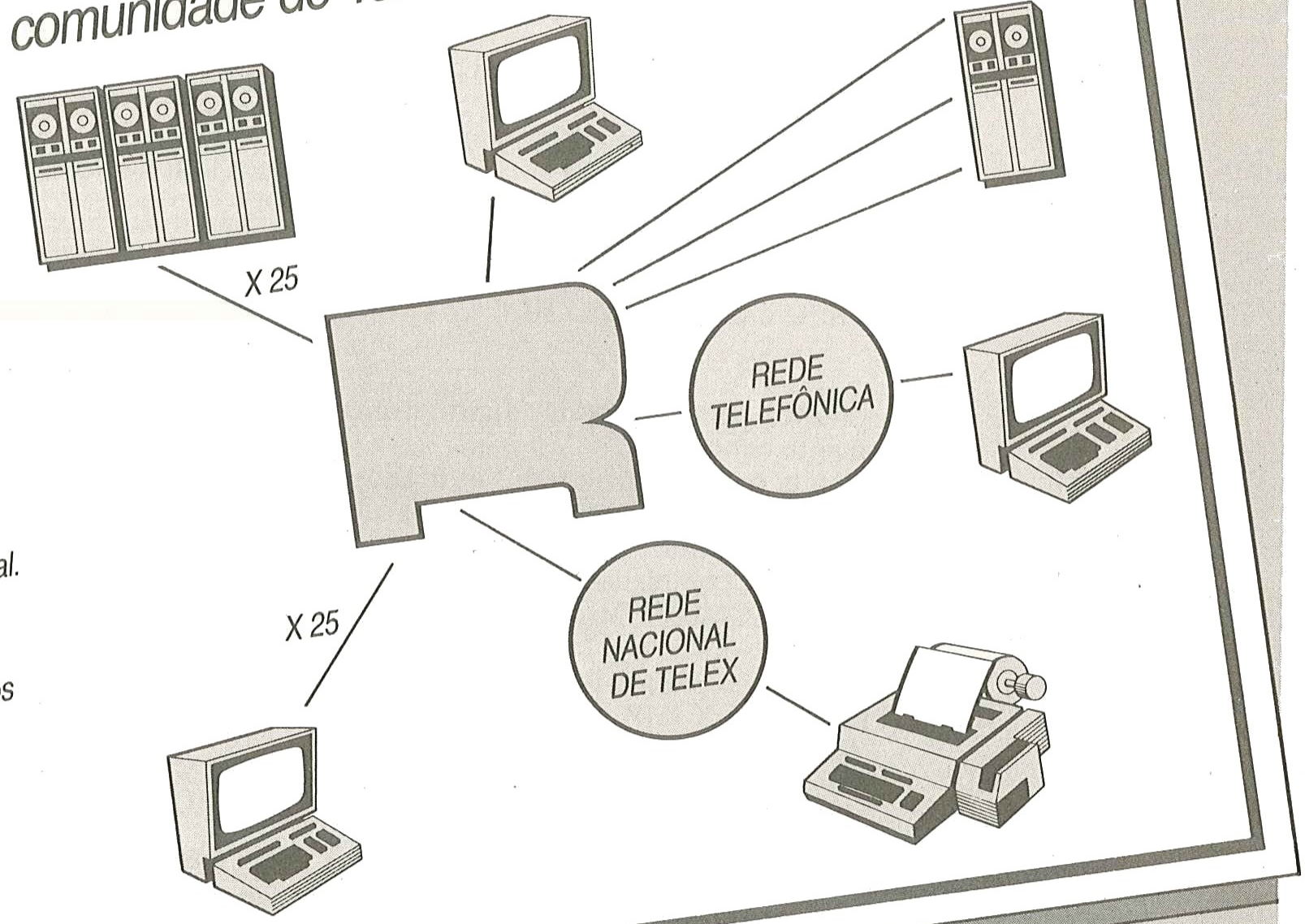
A rotina RECEBA é talvez a mais difícil de ser escolhida. Podemos colocar o preenchimento do buffer com características mínimas, contendo apenas um 'backspace' para corrigir teclas introduzidas erroneamente, ou colocar uma série de caracteres de controle diferentes para rebater a linha introduzida, para editar a linha quando ainda está sendo introduzida, para apagar tudo o que foi digitado até então etc. Fica muito ao gosto pessoal de cada um essa implementação, mas mesmo assim daremos uma rotina relativamente simples para servir de esboço aos que quiserem aperfeiçoá-la.

```
CODE RECEBA
RBUFF H LXI     (aponte para o buffer de recepção)
LEIT SHLD
5Ø D MVI        (tamanho máximo do buffer)
INÍCIO, LETECL CALL (chama a rotina que lê tecla do terminal)
ØD CPI Z/SE, 2ØM  MVI H INX 25 M MVI H INX 2Ø M MVI
                (deixa no buffer o nome '%' do programa de "fimdebuffer)
ØDA MVI ' VID CALL ØA A MVI ' VID CALL
                (envia um 'crlf' para o vídeo)
A XRA VAZIO STA (indica buffer não vazio)
PROX JMP        (fim da rotina)
ENTÃO,
8 CPI           (é um caracter de backspace?)
Z/ SE,
5Ø A MVI D CMP  (já está no início da linha?)
NZ/ SE, D INR H DCX
                (atualize ponteiro e contador)
8 A MVI ' VID CALL 2Ø A MVI ' VID CALL
8 A MVI ' VID CALL
                (apague o caracter anterior, recuando o cursor)
ENTÃO,
SENÃO,          (se não for nenhum caracter de controle, entre-o)
A M MOV H INX   (guarde o caracter no buffer e incremente o ponteiro)
```

A Comutação revolucionar o quadr

Veja algumas vantagens que os Serviços Renpac trazem à comunidade de Teleinformática.

- ☐ Custo em função do volume de dados transmitidos, com redução dos custos fixos.
- ☐ Interconexão de terminais de dados e de computadores segundo procedimentos padronizados.
- ☐ Viabilidade de aplicações de baixo/médio volume de dados.
- ☐ Maior controle das informações do seu negócio e maior integração com os seus clientes.
- ☐ Viabilização do uso temporário e eventual.
- ☐ Estímulo à consulta a Bancos de Dados e recuperação das informações.
- ☐ Fomento à criação de Bancos de Dados ampliando o volume de informações recuperáveis.
- ☐ Viabilização de novas aplicações de teleinformática.
- ☐ Cobertura nacional.



A Embratel conta para dar os últimos

de Pacotes vai o da Teleinformática.

1984 está pintando um quadro muito promissor para quem vive o mundo da teleinformática.

1984 é o ano do lançamento dos Serviços Renpac, oferecidos através da Rede Pública de Comunicação por Comutação de Pacotes, via Embratel.

Com os Serviços Renpac, os usuários compartilham recursos de transmissão por mensagens divididas em blocos-padrão, os pacotes.

De imediato, isso quer dizer inúmeros benefícios que dão maior flexibilidade e melhor desempenho à comunicação de dados.

No quadro ao lado estão algumas das principais vantagens dos Serviços Renpac.

Mas ainda há alguns pontos muito importantes a serem decididos. A serem decididos com você.

Procure a Embratel e fale das necessidades da sua empresa, seja ela grande, média ou pequena. A Embratel acredita que ninguém melhor do que você para saber o que é melhor para você mesmo. E se você é um profissional liberal, faça também o mesmo. Há sempre uma aplicação dos Serviços Renpac para atender economicamente ao seu problema. E tudo com a conhecida eficiência da Embratel.

Isso vale também para você que produz equipamentos. Afinal, o lançamento dos Serviços Renpac vai criar uma demanda por novos itens que a sua indústria pode fornecer.

Para facilitar este nosso diálogo, aqui abaixo, à direita, está um cupom para você preencher e enviar à Embratel. Envie hoje mesmo. Nós entraremos em contato com você. Dê sua participação efetiva para o melhor uso dos Serviços Renpac.

Com ela, e só com ela, a Embratel pode dar os retoques finais no quadro de vantagens que os Serviços Renpac vão trazer para você.

Ministério das Comunicações
EMBRATEL
Empresa do Sistema TELEBRÁS
a sua Empresa de Comunicação de Dados

com você retoques.

À
Empresa Brasileira de Telecomunicações S. A. - Embratel
Departamento de Coordenação Comercial
Av. Presidente Vargas, 1.012 - sala 912 - CEP 20071 - Rio de Janeiro - RJ

NOME .....

EMPRESA .....

CARGO/FUNÇÃO .....

ENDEREÇO .....

CEP ..... *CIDADE* ..... *ESTADO* .....

```
' VID CALL          (ecoe o caracter no vídeo)
ENTÃO,
D DCR
Z/ ATÉ,             (até encher o buffer)
(aqui deve entrar, se desejado um procedimento para quando o buffer encher)
(o mais simples é apenas ignorar o fato, como se nada fosse introduzido)
RBUFF H LXI         (aponte para o início do buffer)
2Ø M MVI H INX 25 M MVI H INX 2Ø M MVI H INX
(deixe um "fimdebuffer" ai')
PROX JMP
```

Com esse interpretador você poderá executar qualquer programa em Filia, apenas fica menos flexível a sua interface com o sistema. A vantagem de um interpretador mais sofisticado é essa interface, que evitaria o rebatimento inteiro de uma introdução de dados errada. Apesar disto, o interpretador fornecido é completo o suficiente para ser usado em qualquer sistema.

Nos próximos artigos traremos um resumo das funções normalmente disponíveis nas implementações de Filia e uma relação da bibliografia disponível, a maioria para o Forth (linguagem "irmã" do Filia), já que Filia ainda é bastante nova para ter uma vasta literatura a seu respeito.

Várias pessoas, especialmente de outras universidades, e alguns usuários de micros pessoais tem me perguntado sobre como poderiam implementar Filia em seus sistemas. Evidentemente, não possuo Filia para todos os micro-computadores diferentes, mas os artigos da revista INTERFACE, já são um bom começo para quem quiser assumir a árdua tarefa de implementar. Para o TK82 e similares, bem como para os sistemas CP/M, e também para o TRS8Ø, temos disponíveis atualmente, estando apenas o manual em conclusão. Espero que isto possa satisfazer à maioria. . .

CURSO Z-80

André Gil Rubens e Ney Acyr de Oliveira

O Z-80 já chega à 14.ª lição.

Nesta lição, André Gil Rubens e Ney Acyr de Oliveira ensinam as instruções do bit carry e as instruções de controle do microprocessador Z-80.

GRUPO DE INSTRUÇÕES DO BIT CARRY

O *Grupo de instruções do bit carry* realiza operações que manipulam o bit carry.

À tabela I mostra as instruções deste grupo.

A primeira instrução (CCF) complementa o bit carry e a segunda instrução (SCF) seta o bit carry.

Observe que o bit carry pode ser testado após a instrução CCF.

Exemplo # 1

O programa da figura 1 reseta o bit carry, primeiramente setando-o, e em seguida complementando-o.

```
SCF     ; SETA O BIT CARRY
CCF     ; COMPLEMENTA O BIT CARRY
HALT
```

Figura 1 - Programa que reseta o bit carry

GRUPO DE INSTRUÇÕES DE CONTROLE DE Z-80

As instruções de controle do Z-80 provocam a sua entrada no estado operacional NOP ou no estado HALT e, também, determinam os modos de atendimento de interrupções.

O estado operacional NOP é destituído de qualquer operação e o estado HALT é caracterizado pela paralisação da busca de instrução.

Para efeito de estudo dividiremos este grupo em dois subgrupos: o subgrupo 1 envolve as instruções de controle dos estados operacionais do Z-80 e o subgrupo 2 envolve as instruções relativas ao atendimento de interrupções.

SUBGRUPO 1 - INSTRUÇÕES DE CONTROLE OPERACIONAL

O Z-80 tem cinco estados operacionais: WAIT, HOLD, NOP, HALT e NORMAL. A passagem para os estados WAIT ou HOLD se faz por hardware e, para os estados NOP e HALT, por software.

O estado WAIT permite que o Z-80 troque informações com periféricos lentos.

O estado HOLD é próprio para a realização de operações DMA.

O estado NORMAL é aquele em que o Z-80 fica no processo contínuo de busca, interpretação e execução de instruções. Consideramos o atendimento de interrupções como parte integrante do estado NORMAL do Z-80. Podemos ter ainda estados WAIT ou SOLD intercalados ao estado NORMAL.

O Z-80 entra nos estados NOP e HALT por software e pelas instruções do mesmo nome, que fazem parte do subgrupo que estamos estudando.

A Figura 2 mostra o diagrama de transição entre os estados NORMAL, HALT e NOP.

O Z-80 se mantém no estado NORMAL até que ocorra uma busca de uma instrução HALT ou NOP.

A instrução HALT coloca o Z-80 no estado HALT, que se caracteriza por paralisar o avanço do contador de programa e a busca de instruções. A saída do estado HALT só é possível através de uma interrupção ou de um reset.

A instrução NOP coloca o Z-80 no estado NOP, que se caracteriza por executar uma instrução destituída de qualquer operação. O estado NOP é um estado quase estável, pois finda a execução da instrução NOP, uma nova instrução é apanhada e executada, retornando o Z-80 ao seu estado normal.

Geralmente se usa a instrução NOP quando se deseja provocar atrasos de tempo (delay) em um programa ou reservar posições de memória intercaladas entre programas.

SUBGRUPO 2 - INSTRUÇÕES DE CONTROLE DE INTERRUPÇÕES

A instrução CI bloqueia qualquer solicitação de interrupção que chegue à entrada INT (pino 16) do Z-80. Efetivamente, esta instrução reseta o flip-flop IFF1 interno ao Z-80.

A instrução EI habilita as interrupções que chegam à entrada INT, setando o flip-flop IFF1.

As instruções geradas pela entrada NMI (pino 17) não são afetadas pelas instruções EI ou DI e ficam sempre habilitadas.

As instruções IMO, IM1 e IM2 habilitam o Z-80 a atender interrupções INT nos modos 0, 1 e 2, respectivamente. E são intrínsecas à arquitetura do Z-80.

Após o reset o modo - é automaticamente assumido (default)

A tabela 2 mostra as instruções desse grupo.

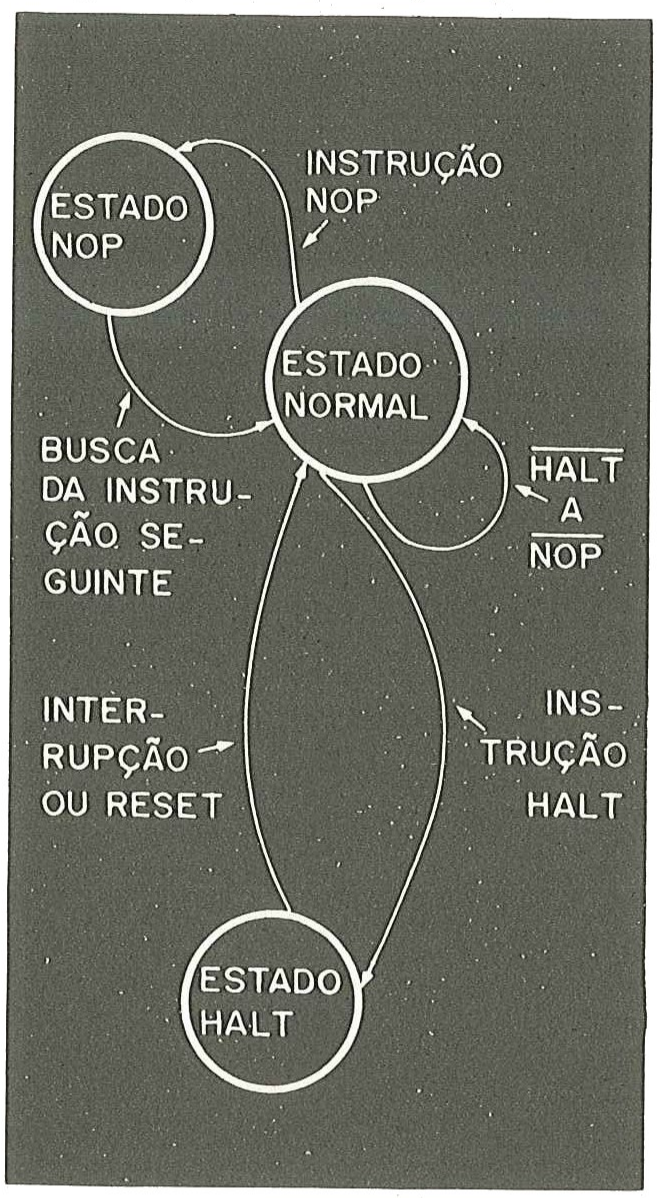


Figura 2 - Diagrama de transição entre os estados NORMAL, HALT e NOP

TABELA 1 INSTRUÇÕES DO BIT CARRY

MNEMÔNICOS 8085	MNEMÔNICOS Z-80	OPERAÇÃO SIMBÓLICA	FLAGS C	Z	P/V	S	N	H	CÓDIGO DE MÁQUINA BINÁRIO 76 543 210	Nº DE BYTES	Nº DE M-CICLOS	Nº DE T-CICLOS
CMC	CCF	CY — $\overline{CY}$	↕	●	●	●	0	X	00 111 111	1	1	4
STC	SCF	CY — 1	1	●	●	●	0	0	00 110 111	1	1	4

NOTAS:

1- CY SIMBOLIZA O FLIP-FLOP DO CARRY BIT

2- NOTAÇÃO DOS FLAGS: ● = NÃO AFETADO, 0 = RESETADO
1 = SETADO , X = INDEFINIDO
↕ = O FLAG É AFETADO EM FUNÇÃO DO RESULTADO DA OPERAÇÃO

Tabela 1 - Instruções do bit carry

TABELA 2 INSTRUÇÕES DE CONTROLE DA UCP

MNEMÔNICO 8080/85	MNEMÔNICO Z-80	OPERAÇÃO SIMBÓLICA	FLAGS C	Z	P/V	S	N	H	CÓDIGO DE MÁQUINA BINÁRIO 76 543 210	Nº DE BYTES	Nº DE M-CICLOS	Nº DE T-CICLOS	SUB-GRUPO
OPERAÇÕES DE CONTROLE DE ESTADOS OPERACIONAIS													
NOP	NOP	NO OPERATION	●	●	●	●	●	●	00 000 000	1	1	4	1
HLT	HALT	CPU HALTED	●	●	●	●	●	●	01 110 110	1	1	4	
OPERAÇÕES DE CONTROLE DE INTERRUPÇÕES													
DI	DI	IFF ← 0	●	●	●	●	●	●	11 110 011	1	1	4	
EI	EI	IFF ← 1	●	●	●	●	●	●	11 111 011	1	1	4	
—	IM 0	SETA MODO Ø DE INTERRUPÇÃO	●	●	●	●	●	●	11 101 101 01 000 110	2	2	8	2
—	IM 1	SETA MODO 1 DE INTERRUPÇÃO	●	●	●	●	●	●	11 101 101 01 010 110	2	2	8	
—	IM 2	SETA MODO 2 DE INTERRUPÇÃO	●	●	●	●	●	●	11 101 101 01 011 110	2	2	8	

NOTAÇÃO DOS FLAGS : ● = NÃO AFETADO

Tabela 2 - Instruções de controle da UCP

LOGO

Claudio Esperança

LINGUAGEM LOGO: A FERRAMENTA DE ENSINO POR COMPUTADOR

Uma linguagem fácil de aprender, que dá total liberdade aos alunos e que lhe permite explorar todas as atividades que podem ser desenvolvidas num computador é aqui explicada por Cláudio Esperança.

Já há algum tempo ouvimos falar na linguagem LOGO, quase sempre associada a comentários do tipo: "LOGO é uma linguagem muito interessante para crianças!". Este preconceito a respeito do LOGO é muito comum, e reflete um pensamento generalizado não só no Brasil como também no exterior. Invariavelmente, entretanto, frases assim costumam deixar irritadas as pessoas que trabalham no dia-a-dia com essa linguagem. Na verdade, como iremos explicar adiante, o LOGO é uma linguagem muito séria.

A história do LOGO

A linguagem LOGO foi idealizada pelo Prof. Seymour Papert, no Instituto Tecnológico de Massachusetts (MIT), na década de 70, com o intuito de servir como ferramenta de ensino utilizando computadores. *A idéia do Prof. Papert foi a de montar um ambiente de aprendizado onde o aluno tivesse total liberdade de criação, ao contrário dos moldes tradicionais de ensino.* A linguagem teria de ser, é claro, muito fácil de aprender, no entanto, deveria ser suficientemente poderosa a fim de permitir que o aluno explorasse todas as atividades que se pode fazer com um computador.

Sendo as aplicações gráficas as que mais interesse despertavam entre os alunos, criou-se inicialmente um método pelo qual figuras e desenhos pudessem ser gerados utilizando ao mínimo possível noções de matemática. Surgiu então o que chamamos hoje de "geometria da tartaruga" ou, como é conhecida em inglês, "turtle geometry". *Neste método, as linhas são traçadas no papel por uma tartaruga que leva consigo uma pena que é arrastada sobre o papel à medida que ela se move.* O movimento da tartaruga é comandado por instruções simples do tipo: Avance, Vire à esquerda, Vire à direita, Levante a pena, Abaixe a pena, etc. . .

Inicialmente a "tartaruga" foi construída fisicamente sob a forma de um robô controlado por rádio a partir de um computador. Mais tarde passou-se a utilizar um terminal gráfico onde a tartaruga assumia a forma de um triângulo que ia se movimentando na tela.

O desenvolvimento da idéia da "geometria da tartaruga" levou naturalmente à sua integração dentro de uma linguagem de computador, surgindo então o LOGO. Como linguagem em si, *o LOGO pode ser considerado como um descendente do LISP, outra linguagem desenvolvida no MIT para ser usada em aplicações voltadas para inteligência artificial.*

Características da Linguagem

Em LOGO, o desenvolvimento de um programa é feito ensinando-se o computador como executar um determinado procedimento. Para tanto, designa-se um nome para esse procedimento e escreve-se:

```
TO <nome do procedimento>
   <comando>
   <comando>

   <comando>
END
```

Daí em diante, sempre que for citado o <nome do procedimento>, o sistema LOGO se encarregará de executar todos os <comandos> de que consistem o procedimento <comando>, que aparecem entre a palavra "TO" e a palavra "END".

Desenhando Figuras

Vejamos então como faríamos para ensinar o computador a desenhar um quadrado:

```
TO QUADRADO
   FORWARD 100
   LEFT 90
   FORWARD 100
   LEFT 90
   FORWARD 100
   LEFT 90
   FORWARD 100
END
```

A instrução FORWARD <n> comanda a tartaruga para avançar 100 posições na tela gráfica; a instrução LEFT <n> faz com que a tartaruga gire <n> graus à esquerda. A definição da palavra "QUADRADO" é feita, portanto, ordenando-se que a tartaruga avance 100 posições, gire 90 graus à esquerda, ande mais 100 posições, e assim por diante, até completar o quadrado. Veja como isto seria feito na tela do computador:

Chama-se de recursividade ao ato de um procedimento chamar a si mesmo, isto é, fazer com que um dos comandos que fazem parte da definição de um procedimento seja um comando para executar a si próprio.

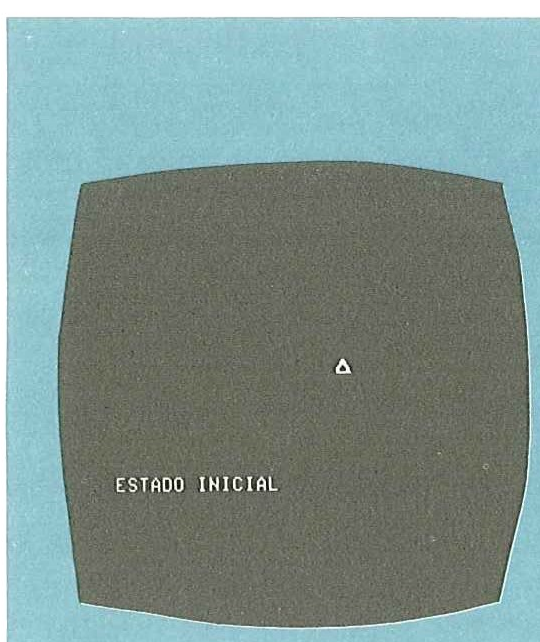


Figura 1

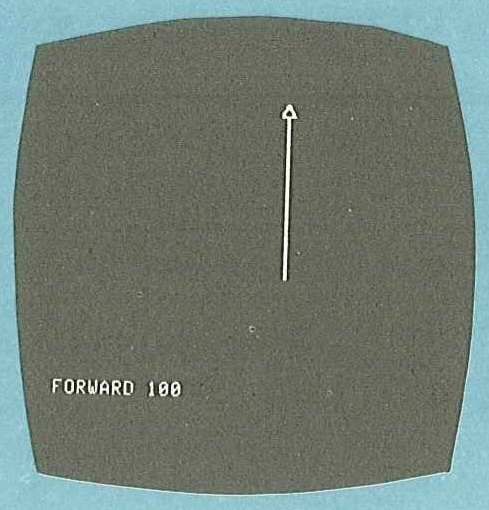


Figura 2

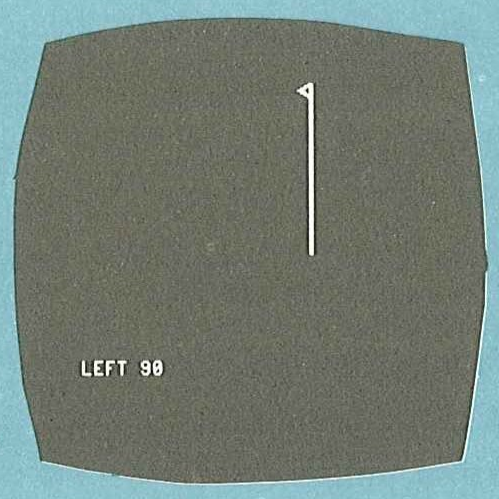


Figura 3

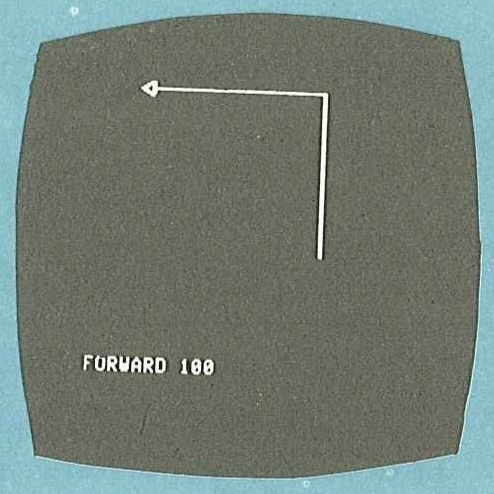


Figura 4

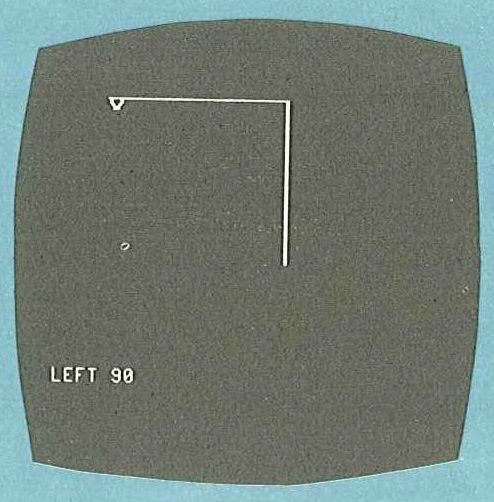


Figura 5

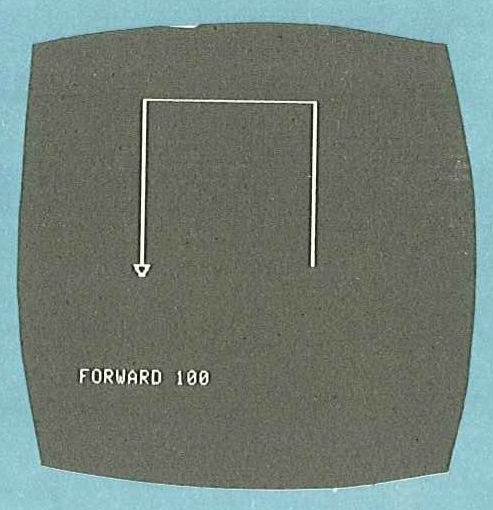


Figura 6

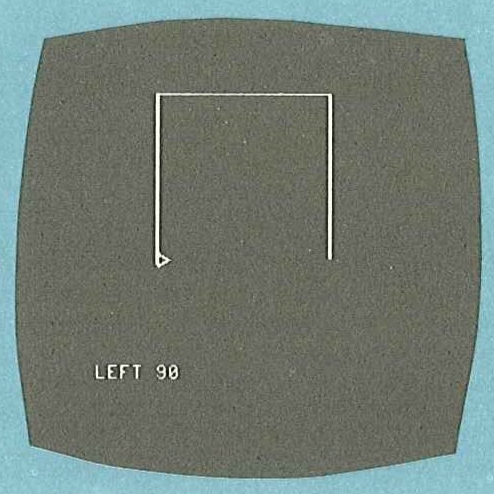


Figura 7

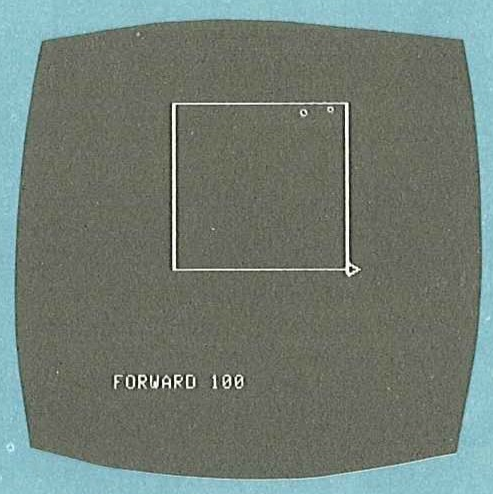


Figura 8

O procedimento "QUADRADO" efetivamente, executa a tarefa a que se propõe, no entanto, ele o faz de uma maneira fixa, isto é, ele sempre traça quadrados com lados de tamanho 100. Melhor seria se pudéssemos modificá-lo para que ele pudesse traçar quadrados com lado "n", onde "n" seria um valor qualquer. Para isso, utilizamos uma das mais importantes facilidades da linguagem: a habilidade de receber valores passados de fora da rotina.

Na definição de qualquer procedimento em LOGO podemos especificar ao lado do <nome do procedimento> o nome de um ou mais <parâmetros> que serão utilizados pelo procedimento:

TO<nome do procedimento> <parâmetro>. . .<parâmetro>
<comando>
<comando>

<comando>

Neste ponto, precisamos esclarecer que em LOGO os nomes de valores são sempre precedidos pelo sinal de dois-pontos (":"). Isto visa diferençar os nomes de valores dos nomes de procedimentos. Assim, poderíamos modificar o procedimento "QUADRADO" para aceitar um parâmetro chamado "LADO".

```
TO QUADRADO :LADO
   FORWARD :LADO
   LEFT 90
   FORWARD :LADO
   LEFT 90
   FORWARD :LADO
   LEFT 90
   FORWARD :LADO
END
```

O LOGO também possui estruturas de controle que permitem efetuar a repetição de um comando ou lista de comandos um determinado número de vezes – esta é a tarefa do comando REPEAT. Para usar o comando REPEAT escreve-se:

REPEAT <número de vezes> [<comando>. . <comando>]

A razão de necessitarmos do uso de um par de colchetes circundando os comandos é que o REPEAT espera sempre uma *lista* de comandos para repetir. O conceito de "lista" é um elemento muito importante utilizado pelo LOGO. Informalmente, poderíamos dizer que uma lista é um conjunto ordenado de elementos. Voltaremos a esse assunto mais tarde.

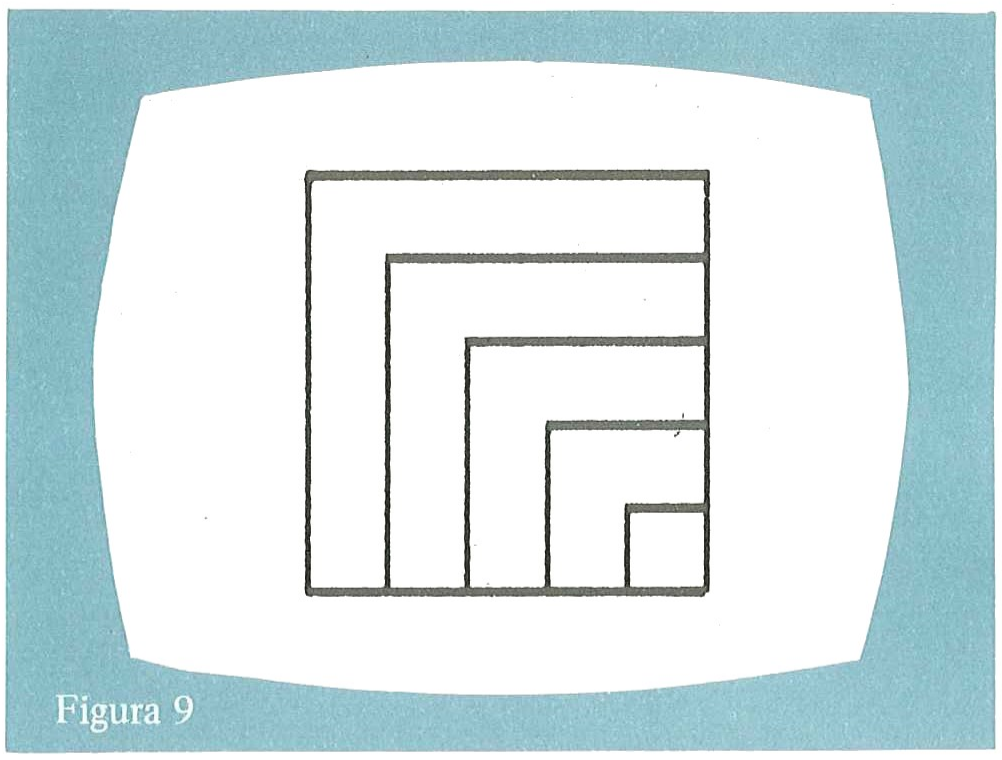
Usando então esse nosso recém-aprendido comando – o REPEAT – poderíamos reescrever o procedimento "QUADRADO" da seguinte forma:

```
TO QUADRADO :LADO
   REPEAT 4 [FORWARD :LADO LEFT 90]
END
```

Vamos agora experimentar esse novo procedimento para ver o que acontece. Para fazer isso, basta ordenar ao computador que execute o procedimento QUADRADO algumas vezes, com valores diferentes. Tentemos, por exemplo, o seguinte:

A idéia do professor Papert foi a de montar um ambiente de aprendizado onde o aluno tivesse total liberdade de criação, ao contrário dos moldes tradicionais de ensino.

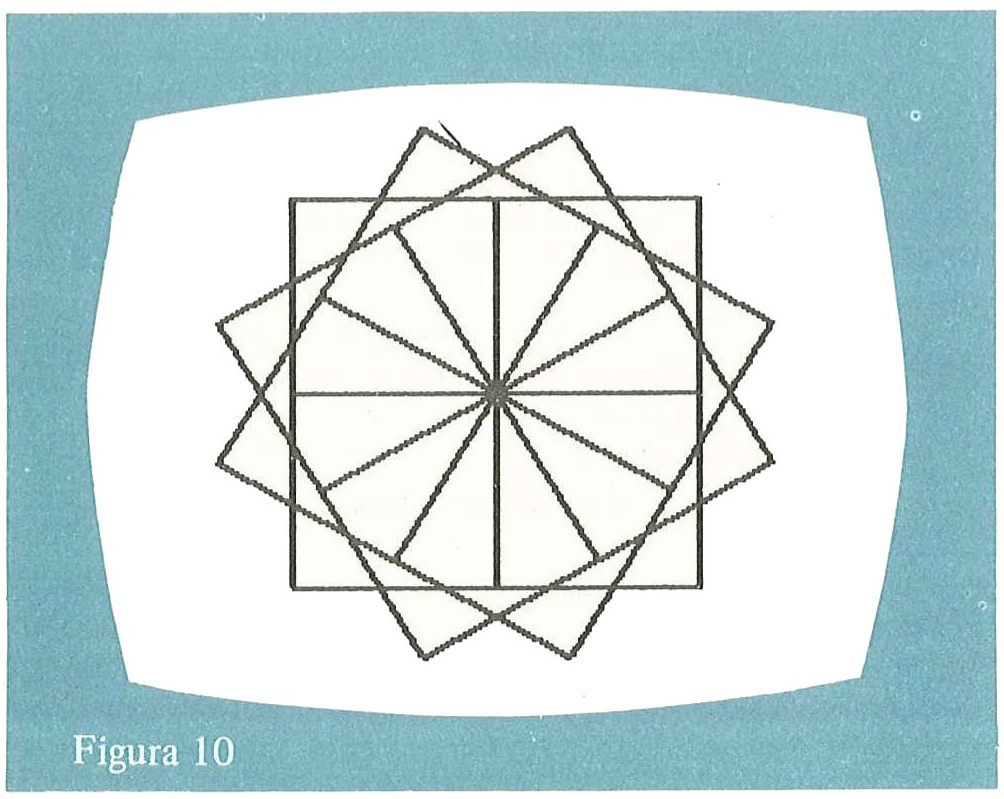
QUADRADO 20
QUADRADO 40
QUADRADO 60
QUADRADO 80
QUADRADO 100
Eis o que teríamos na tela após essa seqüência:


Figura 9

Uma outra experiência seria efetuar repetidamente o procedimento "QUADRADO", mas agora girando a tartaruga a cada vez. Vamos montar então um novo procedimento chamado GIRA-QUADRADO:

```
TO GIRAQUADRADO
   REPEAT 12 [QUADRADO 80 LEFT 30]
END
```

Efetuando esse procedimento no computador faria aparecer na tela:


Figura 10

Recursividade

É hora de sermos apresentados a mais interessante das características do LOGO: sua habilidade de recursão. *Chama-se de recursividade ao ato de um procedimento chamar a si mesmo, isto é, fazer com que um dos comandos que fazem parte da definição de um procedimento seja um comando para executar a si próprio.*

Para experimentar esta nova habilidade, vamos construir um procedimento que é, em essência, uma generalização do procedimento "QUADRADO" – vamos chamá-lo de "POLÍGONO". Para traçar um polígono, precisamos conhecer o tamanho do seu lado e o ângulo existente entre dois lados sucessivos; conseqüentemente, POLÍGONO terá esses dois parâmetros. Vejamos:

```
TO POLÍGONO :LADO :ÂNGULO
   FORWARD :LADO
   LEFT :ÂNGULO
   POLÍGONO :LADO :ÂNGULO
END
```

Nas figuras 11, 12 e 13 temos os resultados obtidos com esse procedimento.

Podemos obter figuras ainda mais interessantes se modificarmos POLÍGONO de forma a incrementar o valor de LADO a cada vez que é executado, gerando assim figuras parecidas com espirais – a este novo procedimento vamos chamar então de ESPIRAL. ESPIRAL necessita de um parâmetro adicional que indicará de quanto o lado deverá ser incrementado a cada vez:

```
TO ESPIRAL :LADO :ÂNGULO :INC
   FORWARD :LADO
   LEFT :ÂNGULO
   ESPIRAL :LADO+:INC :ÂNGULO :INC
END
```

Exemplos de ESPIRAL estão nas figuras 14, 15 e 16.

O grande problema com a recursividade é saber quando parar. No procedimento abaixo, ESPIRAL chama a si mesmo indefinidamente – o computador só pára quando se aperta uma tecla especialmente designada para abortar qualquer procedimento que esteja sendo executado. Isto não tem importância quando se está simplesmente fazendo experiências, porém, quando se está programando a sério, deve ser levado em conta. Para programar ESPIRAL "seriamente", poderíamos incluir um teste impedindo que o procedimento chamasse a si mesmo quando LADO ultrapassasse o valor de, digamos, 100. É aqui que necessitamos usar um comando bem ao "estilo BASIC" – o IF – que tem a forma:

IF <condição> <comando> . . . <comando>
Vamos então colocar um IF em ESPIRAL:

```
TO ESPIRAL :LADO :ÂNGULO :INC
   FORWARD :LADO
   LEFT :ÂNGULO
   IF :LADO<100 ESPIRAL :LADO+:INC :ÂNGULO
      :INC
END
```

Brincando com palavras

O LOGO não está confinado a aplicações gráficas. Pelo contrário, uma das atividades às quais o LOGO se sai muito bem é no processamento de texto que sem dúvida é uma das aplicações mais exploradas pelos computadores.

Os textos são representados num sistema LOGO sob a forma de listas de palavras. Este é um conceito simples, po-

O LOGO pode ser considerado um descendente do LISP, outra linguagem desenvolvida no MIT para ser usada em aplicações voltadas para a inteligência artificial.

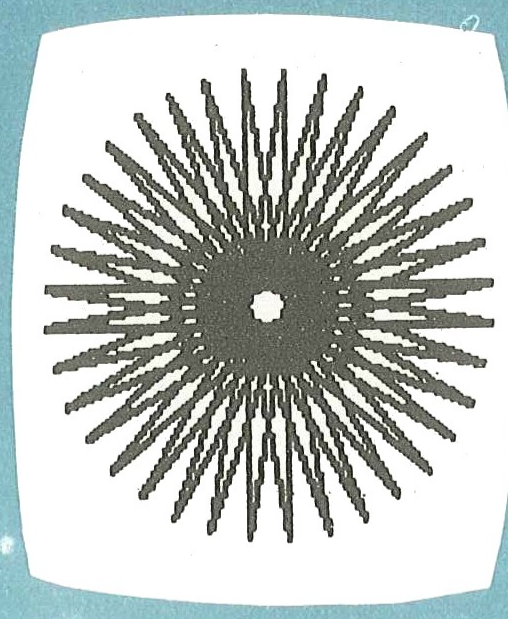


Figura 11

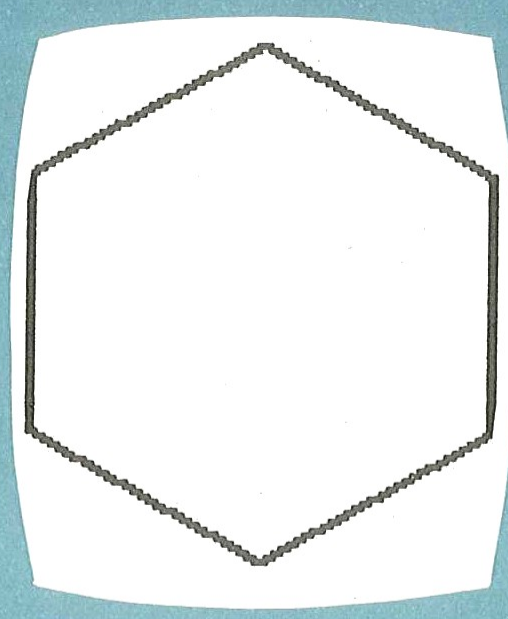


Figura 12

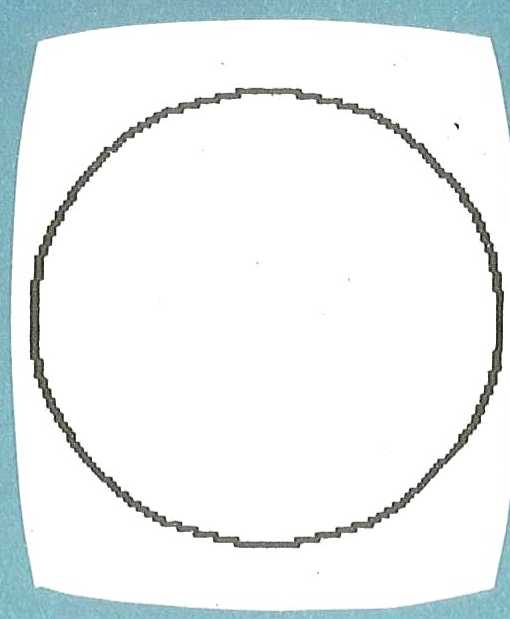


Figura 13

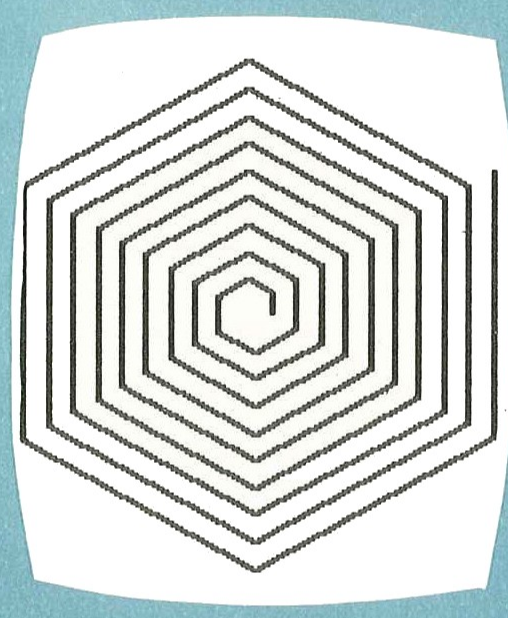


Figura 14

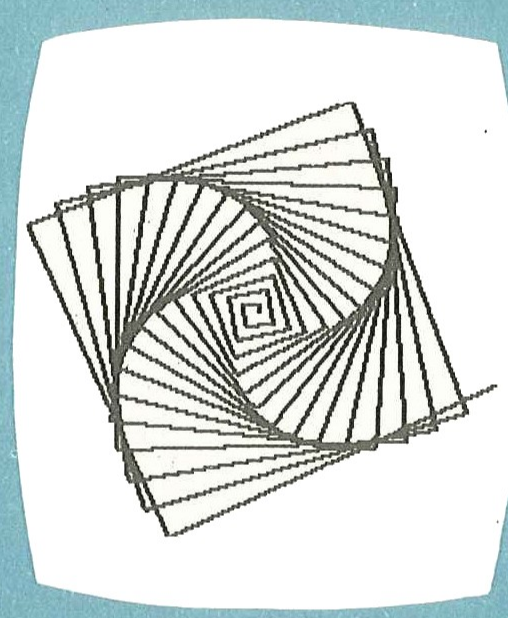


Figura 15

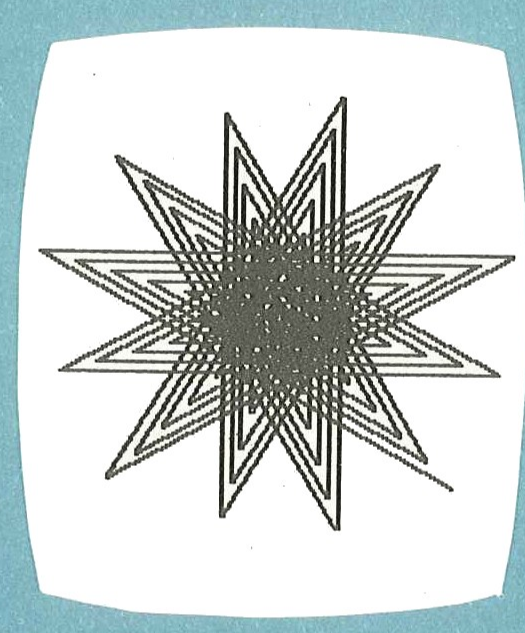
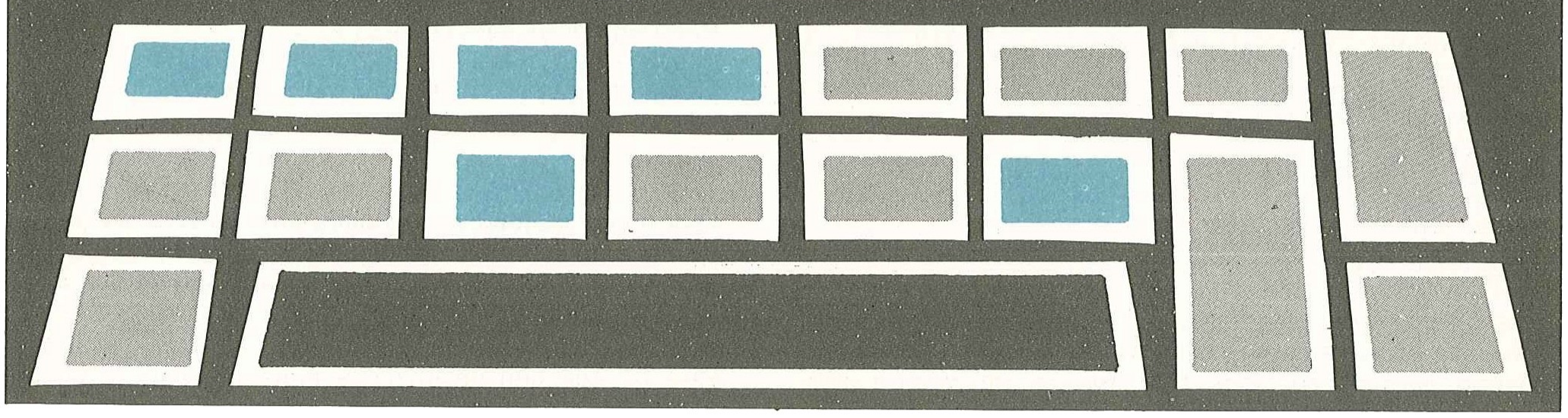


Figura 16



rém poderoso, uma vez que permite generalizar todas as operações que se pode fazer com listas para tratar igualmente letras, palavras, frases, parágrafos, etc. . .

Explicando melhor: é fácil pensar numa frase como sendo uma lista de palavras, mas é igualmente possível encarar uma palavra como sendo uma lista de letras, um parágrafo como uma lista de frases e também (por que não?) um número como uma lista de algarismos.

Semelhantemente aos primitivos que lidam com gráficos (FORWARD, LEFT, etc.), em LOGO existem vários primitivos que efetuam operações com listas (um primitivo é um procedimento que o sistema LOGO já sabe como executar, não precisando portanto ser definido). Veremos alguns deles:

Vamos experimentar um pouco esses primitivos. Para isto vamos usar o comando PRINT para imprimir na tela os resultados obtidos:

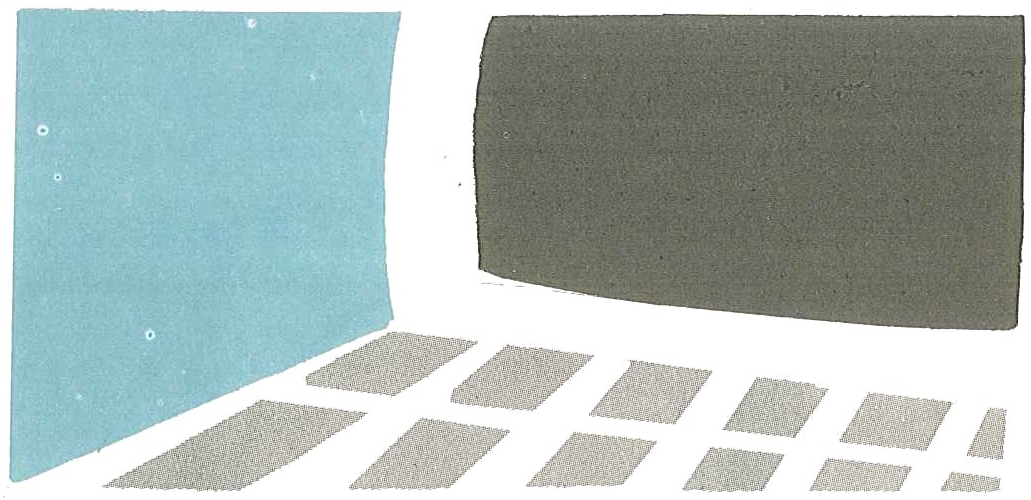
PRINT FIRST [VAMOS A PRAIA?]
Resultado: VAMOS
PRINT BUTFIRST [VAMOS A PRAIA?]
Resultado: A PRAIA?

FIRST <lista> – dá o primeiro elemento de uma lista.

LAST <lista> – dá o último elemento de uma lista.

BUTFIRST <lista> – dá a lista integral menos o primeiro elemento.

BUTLAST <lista> – dá a lista integral à exceção do último elemento.



PRINT LAST [VAMOS A PRAIA?]
Resultado: PRAIA?
PRINT BUTLAST [VAMOS A PRAIA?]
Resultado: VAMOS A.

Precisamos agora aprender uma coisa ou duas a respeito da maneira com que o LOGO trabalha com listas.

Nos exemplos acima, VAMOS A PRAIA? é uma lista composta de três elementos, a saber: VAMOS, A e PRAIA?. Não é sempre, entretanto, que os elementos das listas são palavras. Podemos, por exemplo, ter como elementos de listas outras listas. No caso das próprias palavras, vimos acima que poderíamos encará-las como sendo listas de letras. Veja os seguintes exemplos:

FIRST [[CARLOS SOUZA] [STELA ALVES]]
Resultado: CARLOS SOUZA
FIRST FIRST [[CARLOS SOUZA] [STELA ALVES]]
Resultado: CARLOS
FIRST FIRST FIRST [[CARLOS SOUZA] [STELA ALVES]]
Resultado: C
FIRST LAST [[CARLOS SOUZA] [STELA ALVES]]
Resultado: STELA

Para escrever uma palavra sem colocá-la numa lista, isto é, sem que seja preciso construir uma lista cujo único elemento é essa palavra, o LOGO utiliza uma (e apenas uma) aspas. Assim, "STELA é uma palavra, enquanto que [STELA] é uma lista que contém um elemento: "STELA. Veja o seguinte:

PRINT FIRST "STELA
Resultado: S
PRINT LAST "STELA
Resultado: A
PRINT BUTFIRST "STELA
Resultado: TELA

Este conceito é importante pois o LOGO também pode utilizar as palavras como sendo variáveis, isto é, entidades que têm valores associados. Para associar um valor a uma determinada variável (palavra) utiliza-se o comando MAKE, que tem a forma:

MAKE <palavra> <valor>

<Valor> pode ser qualquer coisa: uma lista, um número, uma palavra e assim por diante. Note a grande diferença que o LOGO estabelece entre o *nome* de uma variável e o seu *valor. Isto é bem diferente do que é seguido por outras linguagens como o* BASIC por exemplo, onde a variável e o seu valor são coisas praticamente idênticas. Já vimos que o LOGO adota a convenção de que os valores são escritos precedidos pelo sinal de dois-pontos. Para tornar isto mais claro, veja os exemplos abaixo:

MAKE "JOÃO 10
MAKE "V "JOÃO
MAKE "X: JOÃO
PRINT "JOÃO
Resultado: JOÃO
PRINT :JOÃO
Resultado: 10
PRINT "V
Resultado: V
PRINT :V
Resultado: JOÃO
PRINT "X
Resultado: X
PRINT : X
Resultado: 10

É útil, neste ponto, testar esses novos conhecimentos construindo um procedimento que os ponha em prática. O procedimento em questão irá tomar uma frase e imprimi-la ao contrário, isto é, imprimir a última palavra em primeiro lugar, a penúltima em segundo e assim por diante. Vamos chamá-lo de INVERTE:

```
TO INVERTE = FRASE
   IF :FRASE = [ ] TOPLEVEL
   PRINT1 LAST = FRASE
   INVERTE BUTLAST = FRASE
END
```

Em INVERTE usamos dois comandos ainda não explicados, mas que são fáceis de entender:

TOPLEVEL é um comando que faz o computador parar de executar qualquer procedimento.

PRINT1 é semelhante ao comando PRINT já visto, com a diferença de não ocasionar o salto para a próxima linha, isto é, PRINT1 "PALAVRA por exemplo faz com que o cursor pare após o último "A" de PALAVRA.

Vamos experimentar INVERTE:
INVERTE [FUI A PRAIA]
Resultado: PRAIA A FUI
INVERTE [ESTOU COM SONO]
Resultado: SONO COM ESTOU
INVERTE [MEU CARRO ESTÁ PIFADO]
Resultado: PIFADO ESTÁ CARRO MEU
INVERTE [CHOVE]
Resultado: CHOVE
INVERTE [[ESTÁ CHOVENDO]]
Resultado: CHOVENDO ESTÁ
INVERTE [[CARLOS SOUZA] ME CHAMOU]
Resultado: CHAMOU ME CARLOS SOUZA

Conclusão:

Esta breve exploração da linguagem não teve o intuito, é claro, de ensinar LOGO a ninguém, o que seria muita pretensão. O que procuramos fazer é examinar levemente o que seja programar em LOGO e dar uma idéia da filosofia por trás da linguagem.

Para os que continuam encarando o LOGO como um simples brinquedo para crianças, só posso dizer mais isso — Vocês não pegaram o espírito da coisa!

Para quem me seguiu alegremente até este ponto, e ficou motivado e/ou impressionado com esta breve explicação, dou meu sincero voto de encorajamento.

Gostaria também de recomendar a esses últimos alguns livros que poderão ser úteis a quem se proponha a aprender LOGO, ou simplesmente quer conhecer mais a respeito. A propósito, são os mesmos livros que utilizei para confeccionar este texto.

— LOGO for the Apple][, Apple LOGO de Harold Abelson — McGraw-Hill
— Learning with LOGO de Daniel Watt — BYTE Books
— Turtle Geometry de Harold Abelson e Andy diSessa — MIT Press
— Mindstorms: Children, Computers and Powerful Ideas de Seymour Papert — Basic Books

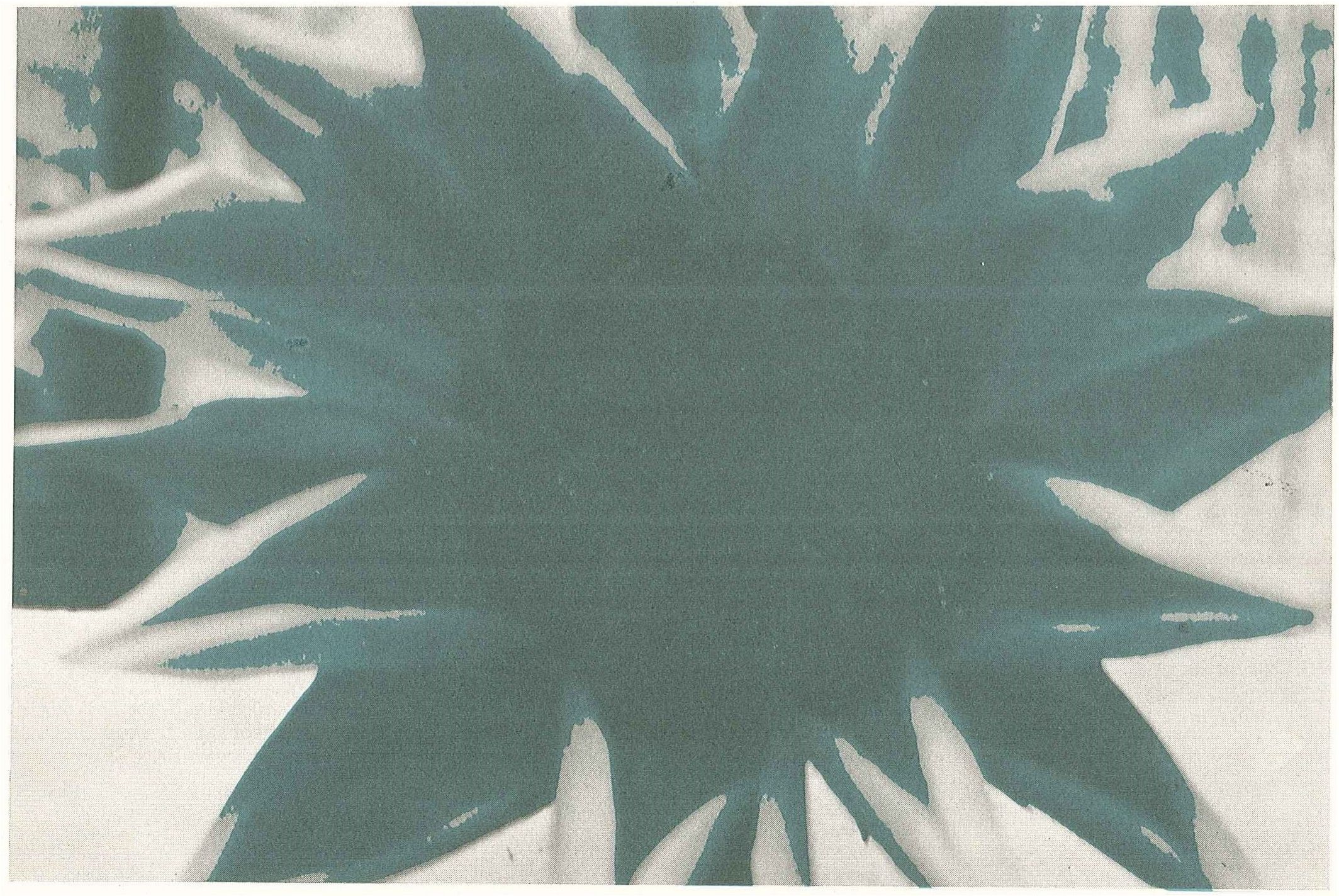
INTERFACE

Dicas de hardware

Carlos Teixeira Dias.

TCI - a interface que testa circuitos integrados

O TCI é uma interface para microcomputadores baseados no microprocessador Z-80, que permite testar qualquer tipo de circuito integrado através de programas específicos, desenvolvidos pelo próprio usuário. Foi desenvolvida por Carlos Teixeira Dias, estudante de engenharia do Instituto Nacional de Telecomunicações (Inatel), de Santa Rita de Sapucaí, com o auxílio de seus professores. O artigo foi dividido em duas partes: a primeira descreve o software utilizado para implementar o interface; a segunda descreve o hardware da interface.



DESCRIÇÃO DO SOFTWARE

O TCI (Testador de Circuito Integrado) necessita de uma rotina em Basic, a qual terá a função de identificar o tipo de CI a ser testado e executar a sua própria sub-rotina que se encontra em linguagem de máquina.

No entanto, para escrevê-la necessitamos de uma área específica de memória e, para reservá-la podemos utilizar dois métodos.

O primeiro é indicar ao microcomputador o endereço que será o topo da memória, ficando então liberada uma área acima deste endereço na qual poderemos escrever as sub-rotinas em linguagem de máquina.

Supondo que queremos reservar uma área acima do endereço 30000 (de 30000 a 32767), para isto deveremos transformar 30000 em binário e dividi-lo em dois bytes; o primeiro será o byte de baixa ordem (L) e o segundo o byte de alta ordem (H). Então teremos: L = 48 e H = 117

Estes dois bytes deverão ser escritos nos endereços 16388 e 16389, que são os endereços que especificam ao microcomputador onde está localizado o topo da memória (RAMTOP). Logo, teremos:

```
POKE 16388,48
POKE 16389,117
NEW
```

Após darmos o NEW, teremos do endereço 30000 ao 32767 livres para escreyer qualquer programa em linguagem de máquina. A única desvantagem de utilizar este método é que as sub-rotinas que se encontram acima do endereço 30000 não poderão ser gravadas em fitas cassetes.

O segundo método consiste em utilizarmos a instrução REM, precedida de uma série de caracteres alfanuméricos. O número de caracteres deverá ser igual ao número de bytes utilizados pelo programa em linguagem de máquina e, então, teremos os endereços livres a partir do endereço 16514, como no exemplo:

REM 1 2 3 4 5 S D CASA HOJE. . .
. . . etc.

A rotina em BASIC a ser utilizada é mostrada a seguir, ficando a cargo do leitor efetuar algumas mudanças na estrutura da mesma, caso seja necessário para a finalidade que o mesmo possa vir a utilizar.

```
1 REM 1 2 3 4 . . . etc
10 PRINT "ENTRAR COM O NÚMERO DO CIRCUITO INTEGRADO A SER TESTADO"
20 INPUT A$
30 IF A$ = "7404" THEN GOTO 400
40 IF A$ = "7405" THEN GOTO 400
50 IF A$ = "7408" THEN GOTO XXX
60 IF A$ = "7410" THEN GOTO YYY
.. .. .. .. .. .. .. ..
..... .. .. .. .. .. ..
100 PRINT "ESTE CIRCUITO INTEGRADO NÃO PODE SER TESTADO, DEVIDO A FALTA DA SUB-ROTINA DE TESTE DO MESMO"
110 GOTO 10
120 LET A = PEEK 32756
130 LET B = PEEK 32757
140 LET C = PEEK 32758
150 LET D = PEEK 32759
160 IF A = 1 THEN GOTO 180
170 GOTO 230
180 IF B = 1 THEN GOTO 200
190 GOTO 230
200 IF C = 1 THEN GOTO 220
210 GOTO 230
220 IF D = 1 THEN GOTO 240
230 PRINT "CIRCUITO INTEGRADO SE ENCONTRA DANIFICADO"
240 PRINT "CIRCUITO INTEGRADO EM PERFEITAS CONDIÇÕES"
250 RETURN
260 . . .. . . . . .
270 . . .. . . . . .
.. . . . .. . . . . .
.. . . .. . . . . .
```

No programa acima, as linhas 30 e 40 perguntam se A$ é igual a 7404 ou 7405. Caso seja, o computador passa a executar imediatamente a linha 400, onde se encontra a sub-rotina de teste destes circuitos integrados.

Os X's e os Y's que se encontram no final das linhas 50 e 60 indicam, respectivamente, o número das linhas em que deverá iniciar a sub-rotina de teste dos CI's 7408 e 7410, respectivamente.

Os endereços 32756, 32757, 32758 e 32759 são utilizados pela sub-rotina em linguagem de máquina para armazenar resultados obtidos durante a comparação (teste) do CI 7404 ou 7405. Dependendo do tipo de CI a ser testado, serão utilizados mais endereços para armazenar estes resultados obtidos durante os testes. Normalmente utiliza-se dois endereços para cada tipo de combinação a ser testada, por exemplo: para o 7404 e 7405 temos duas combinações possíveis.

Entrada	Saída
0	1
1	0

Logo, necessitaremos de 4 endereços para armazenar os dados.

Para o 7408 temos quatro tipos de combinações:

E1	E2	S
0	0	0
0	1	0
1	0	0
1	1	1

Logo, necessitaremos de 8 endereços para armazenar os dados. Para que não tenhamos que modificar a rotina em Basic faremos uma outra sub-rotina em Basic para efetuar a análise dos endereços utilizados durante o teste de cada tipo de CI.

A partir da linha 260 deve-se colocar as outras sub-rotinas (em Basic), que verificarão os endereços utilizados durante o teste para que possamos obter na tela do vídeo um resultado real do teste realizado.

Façamos agora a sub-rotina em linguagem de máquina, a qual é responsável pelo teste do circuito integrado; logo após o programa faremos uma análise detalhada da mesma.

```
400 LET R = 0
405 POKE 16514,33
410 POKE 16515,212
415 POKE 16516,255
420 POKE 16517,54
425 POKE 16518,85
430 POKE 16519,33
435 POKE 16520,2
440 POKE 16521,254
445 POKE 16522,54
450 POKE 16523,207
455 POKE 16524,54
460 POKE 16525,85
465 POKE 16526,6
470 POKE 16527,168
475 POKE 16528,33
480 POKE 16529,0
485 POKE 16530,254
490 POKE 16531,54
495 POKE 16532,128
500 POKE 16533,33
505 POKE 16534,3
510 POKE 16535,254
515 POKE 16536,54
520 POKE 16537,207
525 POKE 16538,54
530 POKE 16539,43
535 POKE 16540,14
540 POKE 16541,212
545 POKE 16542,33
550 POKE 16543,1
555 POKE 16544,254
560 POKE 16545,54
565 POKE 16546,212
570 POKE 16547,33
575 POKE 16548,0
```

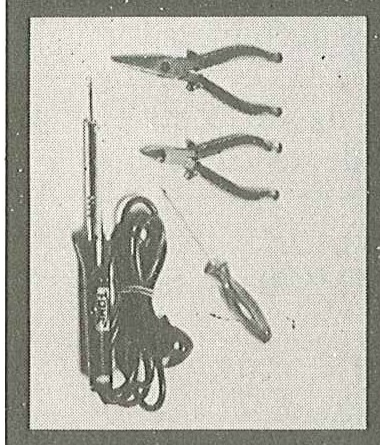
CEDM
CURSOS DE APERFEIÇOAMENTO

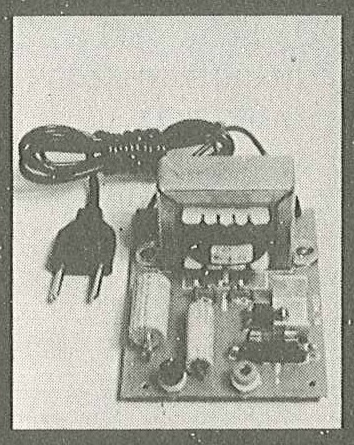
MAIS SUCESSO PARA VOCÊ!

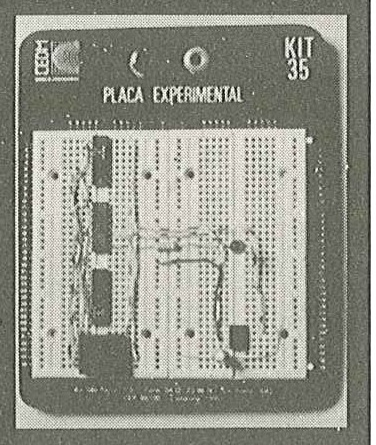
Comece uma nova fase na sua vida profissional.
Os CURSOS CEDM levam até você o mais moderno ensino técnico programado e desenvolvido no País.

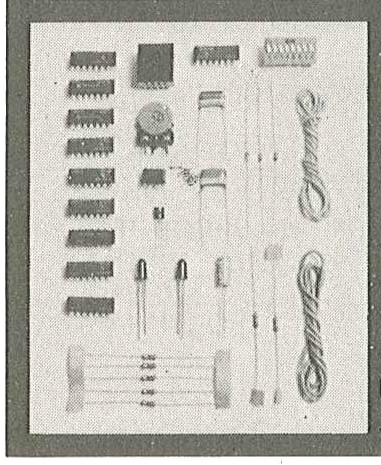
CURSO DE ELETRÔNICA DIGITAL E MICROPROCESSADORES

São mais de 140 apostilas com informações completas e sempre atualizadas. Tudo sobre os mais revolucionário CHIPS. E você recebe, além de uma sólida formação teórica, KITS elaborados para o seu desenvolvimento prático. Garanta agora o seu futuro.

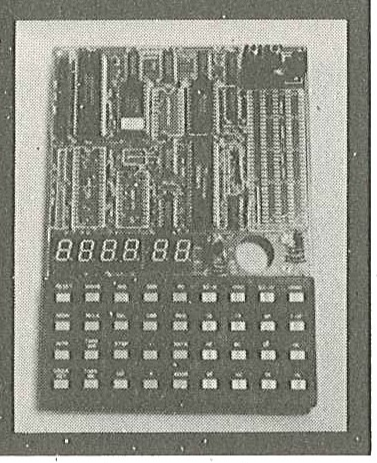






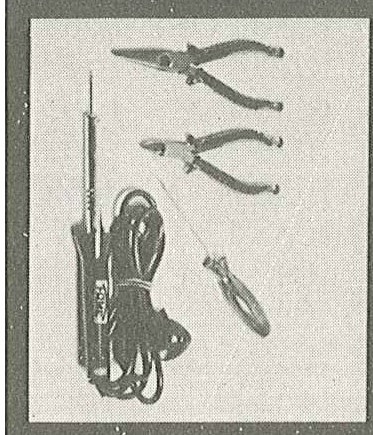


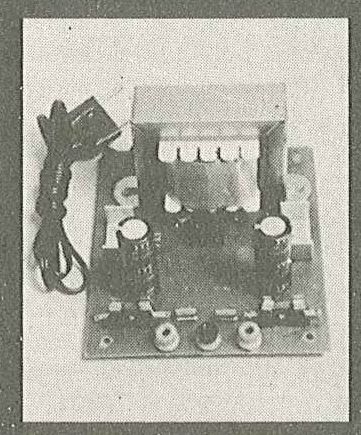
CEDM-20 - KIT de Ferramentas.
CEDM-78 - KIT Fonte de Alimentação 5v/1A. CEDM-35 KIT Placa Experimental
CEDM-74 - KIT de Componentes.
CEDM-80 MICROCOMPUTADOR Z80 ASSEMBLER.

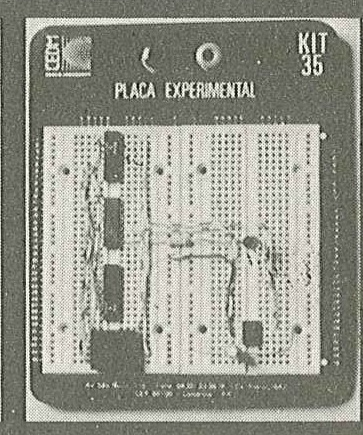


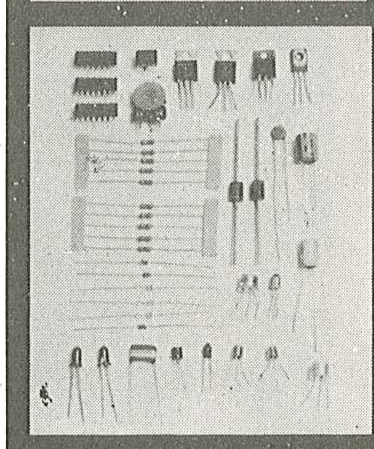
CURSO DE ELETRÔNICA E ÁUDIO

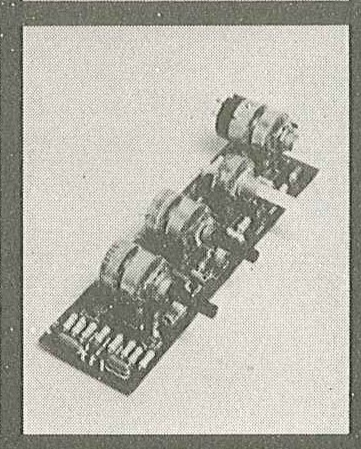
Métodos novos e inéditos de ensino garantem um aprendizado prático muito melhor. Em cada nova lição, apostilas ilustradas ensinam tudo sobre Amplificadores, Caixas Acústicas, Equalizadores, Toca-discos, Sintonizadores AM/FM, Gravadores e Toca-Fitas, Cápsulas e Fonocaptadores, Microfones, Sonorização, Instrumentação de Medidas em Áudio, Técnicas de Gravação e também de Reparação em Áudio.

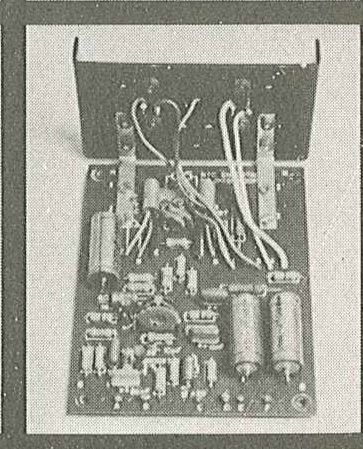








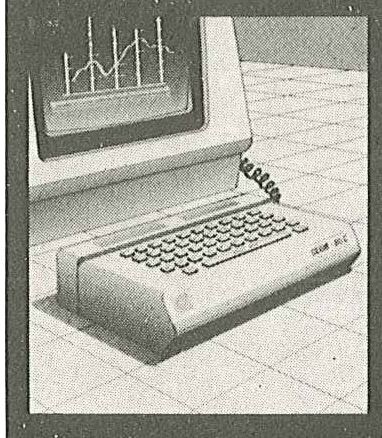


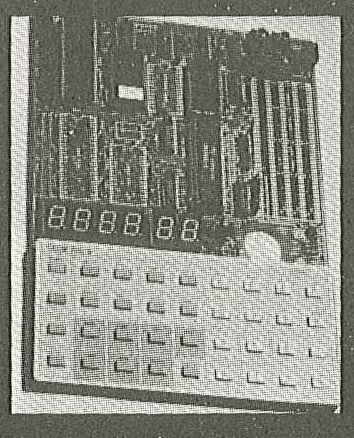


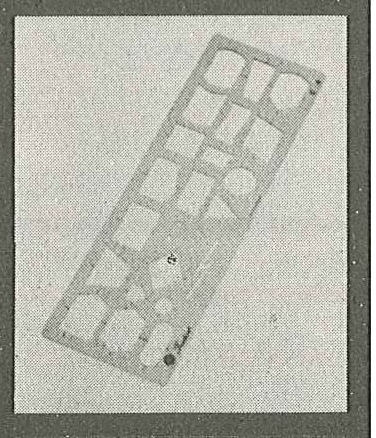
CEDM-1 - KIT de Ferramentas. CEDM-2 - KIT Fonte de Alimentação + 15-15/1A. CEDM-3 - KIT Placa Experimental
CEDM-4 - KIT de Componentes. CEDM-5 - KIT Pré-amplificador Estéreo. CEDM-6 - KIT Amplificador Estéreo 40w.

CURSO DE PROGRAMAÇÃO EM BASIC

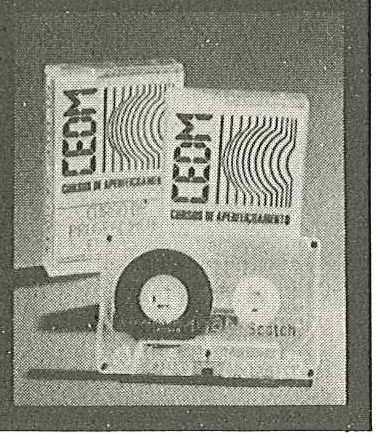
Este CURSO, especialmente programado, oferece os fundamentos de Linguagem de Programação que domina o universo dos microcomputadores. Dinâmico e abrangente, ensina desde o BASIC básico até o BASIC mais avançado, incluíndo noções básicas sobre Manipulação de Arquivos, Técnicas de Programação, Sistemas de Processamento de Dados, Teleprocessamento, Multiprogramação e Técnicas em Linguagem de Máquina, que proporcionam um grande conhecimento em toda a área de Processamento de Dados.







KIT CEDM Z80 BASIC Científico.
KIT CEDM Z80 BASIC Simples.
Gabarito de Fluxograma E-4. KIT CEDM SOFTWARE Fitas Cassete com Programas.



Você mesmo pode desenvolver um ritmo próprio de estudo. A linguagem simplificada dos CURSOS CEDM permite aprendizado fácil. E para esclarecer qualquer dúvida, o CEDM coloca à sua disposição uma equipe de professores sempre muito bem acessorada. Além disso, você recebe KITS preparados para os seus exercícios práticos.

Ágil, moderno e perfeitamente adequado à nossa realidade, os CURSOS CEDM por correspondência garantem condições ideais para o seu aperfeiçoamento profissional.

GRÁTIS

Você também pode ganhar um MICROCOMPUTADOR.

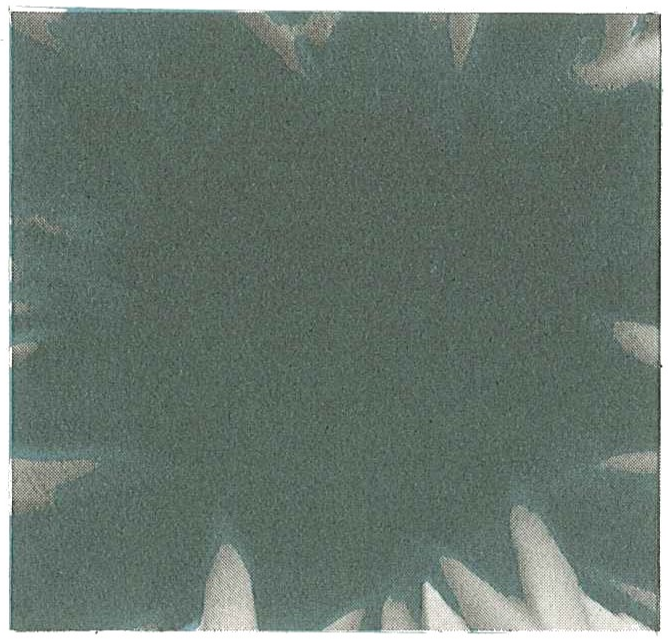
Telefone (0432) 23-9674 ou coloque hoje mesmo no Correio o cupom CEDM.
Em poucos dias você recebe nossos catálogos de apresentação.

CEDM
Avenida São Paulo, 718 - Fone (0432) 23-9674.
CAIXA POSTAL 1642 - CEP 86100 - Londrina - PR

CURSO DE APERFEIÇOAMENTO POR CORRESPONDÊNCIA

Solicito o mais rápido possível informações sem compromisso sobre o CURSO de . . . . . .
Nome. . . . . .
Rua . . . . . .
Cidade . . . . . .
Bairro. . . . . CEP . . . . . .

```
580 POKE 16549,254
585 POKE 16550,126
590 POKE 16551,184
595 POKE 16552,54
600 POKE 16553,244
605 POKE 16554,127
610 POKE 16555,194
615 POKE 16556,178
620 POKE 16557,64
625 POKE 16558,54
630 POKE 16559,1
635 POKE 16560,195
640 POKE 16561,180
645 POKE 16562,64
650 POKE 16563,54
655 POKE 16564,0
660 POKE 16565,33
665 POKE 16566,1
670 POKE 16567,254
675 POKE 16568,126
680 POKE 16569,185
685 POKE 16570,33
690 POKE 16571,245
695 POKE 16572,127
700 POKE 16573,194
705 POKE 16574,196
710 POKE 16575,64
715 POKE 16576,54
720 POKE 16577,1
725 POKE 16578,195
730 POKE 16579,198
735 POKE 16580,64
740 POKE 16581,54
745 POKE 16582,0
750 POKE 16583,22
755 POKE 16584,84
760 POKE 16585,33
765 POKE 16586,0
770 POKE 16587,254
775 POKE 16588,54
780 POKE 16589,0
785 POKE 16590,30
790 POKE 16591,170
795 POKE 16592,33
800 POKE 16593,1
805 POKE 16594,254
810 POKE 16595,54
815 POKE 16596,128
820 POKE 16597,33
825 POKE 16598,0
830 POKE 16599,254
835 POKE 16600,126
840 POKE 16601,186
845 POKE 16602,33
850 POKE 16603,246
855 POKE 16604,127
860 POKE 16605,194
865 POKE 16606,228
870 POKE 16607,64
875 POKE 16608,54
880 POKE 16609,1
885 POKE 16610,195
890 POKE 16611,230
895 POKE 16612,64
900 POKE 16613,54
905 POKE 16614,0
910 POKE 16615,33
915 POKE 16616,1
920 POKE 16617,254
925 POKE 16618,126
930 POKE 16619,187
935 POKE 16620,33
940 POKE 16621,247
945 POKE 16622,127
950 POKE 16623,194
955 POKE 16624,245
960 POKE 16625,64
965 POKE 16626,54
970 POKE 16627,1
975 POKE 16628,195
980 POKE 16629,247
985 POKE 16630,64
990 POKE 16631,54
995 POKE 16632,0
1000 POKE 16633,33
1005 POKE 16634,0
1010 POKE 16635,255
1015 POKE 16636,54
1020 POKE 16637,0
1025 POKE 16638,201
1030 RAND USER 16514
1035 LET R = R + 1
1040 IF R = 5 THEN GOTO 1050
1045 GOTO 405
1050 GOSUB 120
1055 GOTO 10
```



ANÁLISE DO PROGRAMA EM LINGUAGEM DE MÁQUINA

Da linha 405 a 425 o microcomputador habilita o decodificador 1, figura 1, através dos bits de alta ordem (A8 a A15); carrega o primeiro conjunto de flip-flops através dos bits de baixa ordem (A0 a A7) e carrega o segundo conjunto de flip-flops através do barramento de dados (D0 a D7). Os dados armazenados nestes conjuntos de flip-flops são responsáveis pela indicação de entrada ou saída de dados nos amplificadores de corrente.

Da linha 430 a 450 o microcomputador habilita o decodificador 2, figura 1, através do byte H (A8 a A15); o byte L (A0 a A7) habilita o controle da porta A na PIO-Z80 e, através do barramento de dados habilita o controle bit a bit na porta A da PIO-Z80.

Da linha 455 a 460 o microcomputador habilita os *tri-states* internos da porta A da PIO-Z80, especificando se cada saída da porta trabalhará como entrada ou como saída de dados ("1" (um) saída de dados e "0" (zero) entrada de dados).

Da linha 465 a 470 carrega o registro B do microcomputador com os dados que deverão ser obtidos do CI a ser testado.

Da linha 475 a 495 habilita o decodificador 2 através do byte H e, através do byte L habilita o registro A da porta A da PIO-Z80. Através do barramento de dados armazena no registro A a combinação a ser testada no CI (terminal 1 ao 8).

Da linha 500 a 520 o decodificador 2 é habilitado através de H; L habilita o controle da porta B da PIO-Z80 e, através do barramento de dados habilita o controle de bit a bit na porta B.

Da linha 525 a 530 os *tri-states* internos da porta B da PIO-Z80 são habilitados.

Da linha 535 a 540 é armazenado no registro C do microcomputador o dado correto a ser obtido do CI durante o teste.

Da linha 545 a 565 o decodificador 2 é habilitado através de H; o registro da porta B é habilitado através do byte L e através do barramento de dados armazena na porta B da PIO-Z80 a combinação a ser testada no CI (do terminal 9 ao 16).

Da linha 570 a 585, H habilita o decodificador 2; L habilita o registro da porta A da PIO-Z80 e realiza a leitura do registro A da PIO, armazenando o conteúdo da mesma no acumulador do microcomputador.

Da linha 590 a 605 é realizada uma comparação do dado contido no acumulador com o dado contido no registro B do microcomputador ("0" (zero) para igualdade e "1" (um) para não-igualdade), carregando o par de registradores HL com o endereço 32756.

Da linha 610 a 655, se o valor do acumulador após a comparação for "0" (zero) coloca "1" (um) no endereço 32756.

Da linha 660 a 675, H habilita o decodificador 2; L habilita o registro da porta B e realiza a leitura do registro da porta B.

Da linha 680 a 695 compara o valor contido no acumulador com o registro C do microcomputador e carrega os registradores HL com o endereço 32757.

Da linha 700 a 745, caso o valor do acumulador seja "0" (zero) coloca "1" (um) no endereço 32757 e, sendo "1" (um) coloca "0" (zero) no endereço 32757.

Da linha 750 a 755 o registro D do microcomputador é carregado com o

dado correto a ser obtido do CI durante o teste.

Da linha 760 a 780, H habilita o decodificador 2; L habilita o registro da porta A e através do barramento de dados escreve no registro da porta A da PIO-Z80 uma nova combinação a ser testada.

Da linha 785 a 790 carrega o registro E do microcomputador com o dado correto a ser obtido durante o teste do CI.

Da linha 795 a 815, H habilita o decodificador 2; L habilita o registro da porta B e através do barramento de dados armazena um novo dado para efetuar o teste do CI.

Da linha 820 a 835, H habilita o decodificador 2; L habilita o registro da porta A da PIO e realiza a leitura do registro da porta A da PIO-Z80.

Da linha 840 a 855 compara o acumulador com o registro do microcomputador e carrega o par de registradores HL com o endereço 32758.

Da linha 860 a 905, caso o valor contido no acumulador após a comparação seja "0" (zero), coloca "1" (um) no endereço 32758 e se for "1" (um) coloca "0" (zero) no endereço 32758.

Da linha 910 a 925, H habilita o decodificador 2; L habilita o registro da porta B e realiza a leitura da porta B da PIO-Z80.

Da linha 930 a 945 compara o valor do acumulador com o registro E do microcomputador e carrega o par de registradores HL com o endereço 32759.

Da linha 950 a 995, caso o valor do acumulador após a comparação seja "0" (zero) coloca "1" (um) no endereço 32759 e, caso seja "1" (um) coloca "0" (zero) no endereço 32759.

Da linha 1000 a 1020 reseta os dois conjuntos de flip-flops.

Na linha 1025 informa ao microcomputador para retornar ao Basic.

Na linha 1030 indica ao microcomputador que a partir do endereço 16514 ao endereço anterior (16638) deve ser interpretado como um programa em linguagem de máquina.

As linhas 400, 1035, 1040 e 1045 são responsáveis por um ciclo de 5 testes seguidos no CI, antes de apresentar o resultado final no vídeo.

Caso o leitor esteja familiarizado com o ASSEMBLER (*linguagem de máquina*) o mesmo poderá modificar a estrutura da sub-rotina de teste, de maneira que o final do teste apresente um resultado mais detalhado sobre as condições do CI; caso o mesmo se encontre danificado, poderá ser indicado qual ou quais as partes do mesmo que se encontram defeituosas. O programa em ASSEMBLER é apresentado a seguir.

PROGRAMA EM ASSEMBLER
SUB-ROTINA DE TESTE DO 7404 e 7405

Endereço	Código	Instrução	Byte H	Byte L	Comentário
16514	33	LD HL, NN	11111111	11Ø1Ø1ØØ	H - Habilita decodificador 1 L - Carrega Flip-Flop 1
16515	212				
16516	255				
16517	54	LD (HL), N		Ø1Ø1Ø1Ø1	Carrega Flip-Flop 2
16518	85				
16519	33	LD HL, NN	1111111Ø	ØØØØØØ1Ø	H - Habilita decodificador 2 L - Habilita controle A
16520	2				
16521	254				
16522	54	LD (HL), N		11ØØ1111	Habilita em A controle bit a bit
16523	207				
16524	54	LD (HL), N		Ø1Ø1Ø1Ø1	Habilita tri-state interno de A
16525	85				
16526	6	LD B, N		1Ø1Ø1ØØØ	Carrega registro B com dado correto a ser comparado
16527	168				
16528	33	LD HL, NN	1111111Ø	ØØØØØØØØ	H - Habilita decodificador 2 L - Habilita registro A
16529	0				
16530	254				
16531	54	LD (HL), N		1Ø1Ø1ØØØ	Escreve no registro A
16532	168				
16533	33	LD HL, NN	1111111Ø	ØØØØØØ11	H - Habilita decodificador 2 L - Habilita controle B
16534	3				
16535	254				
16536	54	LD (HL), N		11ØØ1111	Habilita em B controle bit a bit
16537	207				
16538	54	LD (HL), N		ØØ1Ø1Ø11	Habilita tri-state interno de B
16539	43				
16540	14	LD C, N		11Ø1Ø1ØØ	Carrega registro C com dado correto a ser comparado
16541	212				
16542	33	LD HL, NN	1111111Ø	ØØØØØØØ1	H - Habilita decodificador 2 L - Habilita registro B
16543	1				
16544	254				
16545	54	LD (HL), N		11Ø1Ø1ØØ	Escreve no registro B
16546	212				
16547	33	LD HL, NN	1111111Ø	ØØØØØØØØ	H - Habilita decodificador 2 L - Habilita registro 2
16548	0				
16549	254				
16550	126	LD A, (HL)			Lé conteúdo do Registro A (PIO-Z80)

16551	184	CP B			Compara Registro A com Registro B (µP)
16552	54	LD HL, NN			Carrega HL com endereço 32756
16553	244				
16554	127				
16555	194	JPNZ NN			Se não for zero GOTO 16562 Se for zero prossegue
16556	178				
16557	64				
16558	54	LD (HL), N		ØØØØØØØ1	Coloca 1 no endereço 32756
16559	1				GOTO endereço 16564
16560	195	JP NN			
16561	180				
16562	64				
16563	54	LD (HL), N		ØØØØØØØØ	Coloca Ø no endereço 32756
16564	0				
16565	33	LD HL, NN	11111110	ØØØØØØØ1	H - Habilita decodificador 2 L - Habilita registro B
16566	1				
16567	254				Lê conteúdo do Registro B (PIO)
16568	126	LD A, (HL)			Compara Registro B com Registro C (µC)
16569	185	CP C			Carrega HL com endereço 32757
16570	33	LD HL, NN			
16571	245				
16572	127				
16573	194	JPNZ NN			Se não for zero GOTO 16580 Se for zero prossegue
16574	196				
16575	64				
16576	54	LD (HL), N		ØØØØØØØ1	Coloca 1 no endereço 32757
16577	1				GOTO 16582
16578	195	JP NN			
16579	198				
16580	64				
16581	54	LD (HL), N		ØØØØØØØØ	Coloca Ø no endereço 32757
16582	0				
16583	22	LD D, N		Ø1Ø1Ø1ØØ	Carrega Registro D com dado correto a ser comparado
16584	84				
16585	33	LD HL, NN	11111110	ØØØØØØØØ	H - Habilita decodificador 2 L - Habilita registro A (PIO)
16586	0				
16587	254				
16588	54	LD (HL), N		ØØØØØØØØ	Escreve no Registro A da PIO-Z80
16589	0				
16590	30	LD E, N		1Ø1Ø1Ø1Ø	Carrega Registro E com dado correto a ser comparado
16591	170				
16592	33	LD HL, NN	11111110	ØØØØØØØ1	H - Habilita decodificador 2 L - Habilita Registro B da PIO-Z80
16593	1				
16594	254				
16595	54	LD (HL), N		1ØØØØØØØ	Escreve no Registro B da PIO-Z80
16596	128				
16597	33	LD HL, NN	11111110	ØØØØØØØØ	H - Habilita decodificador 2 L - Habilita Registro A
16598	0				
16599	254				
16600	126	LD A, (HL)			Lê conteúdo do Registro A da PIO
16601	186	CP D			Compara Registro D com Registro A da PIO
16602	33	LD HL, NN			Carrega HL com endereço 32758
16603	246				
16604	127				
16605	194	JPNZ NN			Se não for zero GOTO 16612 Se for zero prossegue
16606	228				
16607	64				
16608	54	LD (HL), N		ØØØØØØØ1	Coloca 1 no endereço dado por HL
16609	1				GOTO 16614
16610	195	JP NN			
16611	230				
16612	64				
16613	54	LD (HL), N		ØØØØØØØØ	Coloca Ø no endereço 32758
16614	0				
16615	33	LD HL, NN	11111110	ØØØØØØØ1	H - Habilita decodificador 2 L - Habilita Registro B
16616	1				
16617	254				
16618	126	LD A, (HL)			Lê conteúdo do Registro B da PIO
16619	187	CP E			Compara Registro E com Registro B da PIO
16620	33	LD HL, NN			Carrega HL com endereço 32759
16621	247				
16622	127				
16623	194	JPNZ NN			Se não for zero GOTO 16629 Se for zero prossegue
16624	245				
16625	64				
16626	54	LD (HL), N		ØØØØØØØ1	Coloca 1 no endereço 32759
16627	Í				GOTO 16631
16628	195	JP NN			
16629	247				
16630	64				
16631	54	LD (HL), N		ØØØØØØØØ	Coloca zero no endereço 32759
16632	0				
16633	33	LD HL, NN	11111111	ØØØØØØØØ	H - Habilita decodificador 1 L - Carrega Flip-Flop 1
16634	0				
16635	255				
16636	54	LD (HL), N		ØØØØØØØØ	Carrega Flip-Flop 2 (Reset nos Flip-Flops)
16637	0				Retorna para ROTINA BASIC
16638	201	RET			

DESCRIÇÃO DE HARDWARE

O hardware utilizado no testador de CI's é bastante simples e tem como elemento principal a PIO-Z80, que é um importantíssimo dispositivo de entrada/saída. A figura 1 apresenta o diagrama em blocos do Testador de Circuito Integrado.

Vamos fazer uma pequena análise do funcionamento da PIO-Z80, para que o leitor possa compreender melhor o funcionamento do circuito e do programa em linguagem de máquina.

A PIO-Z80 é constituída por duas portas A e B, sendo que cada porta possui um registro de dados e controles. Os controles são responsáveis pela definição se as portas trabalharem como saídas ou entradas, ou ainda, se trabalharem com controle bit a bit Neste caso, deve-se habilitar os *tri-states* internos das portas, especificando quais deles serão saída e quais serão entrada. O registro de dados tem a função de armazenar os dados provenientes do barramento de dados, após ser habilitado, de modo que possa ser obtido na saída da porta durante o tempo desejado.

As entradas B/$\overline{A}$ e C/$\overline{D}$ da PIO-Z80 são ligadas aos bits A0 e A1 do barramento de endereço do microcomputador e selecionam em que porta trabalharão, porta A ou porta B. Assim sendo, utilizamos o byte L (byte de baixa ordem) para habilitar e controlar as

BYTE L				FUNÇÃO
00	00	00	00	Habilita registro da porta A
00	00	00	01	Habilita registro da porta B
00	00	00	10	Habilita controle de A
00	00	00	11	Habilita controle de B
A7 A6	A5 A4	A3 A2	A1 A0	

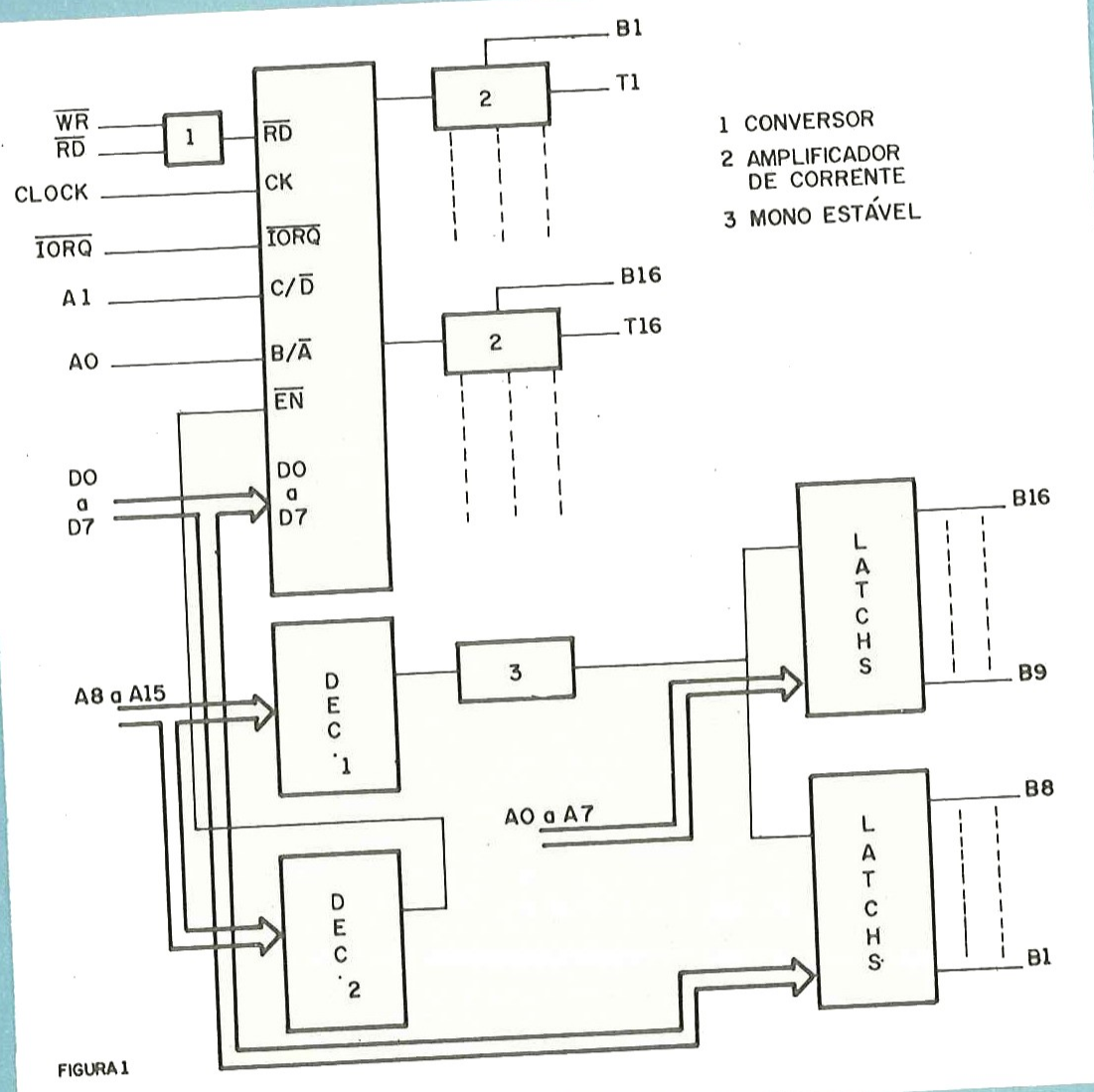


FIGURA 1

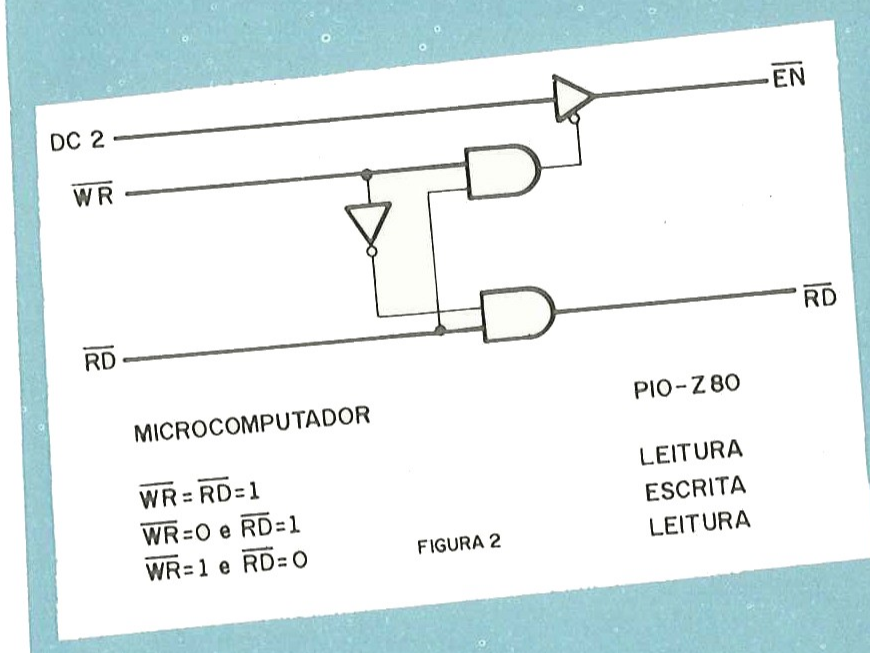


FIGURA 2

Figura 1 - Diagrama em bloco do Testador de Circuito Integrado (TCI)
Figura 2 - Conversor
Figura 3 - Diagrama em bloco simplificado da PIO-Z80

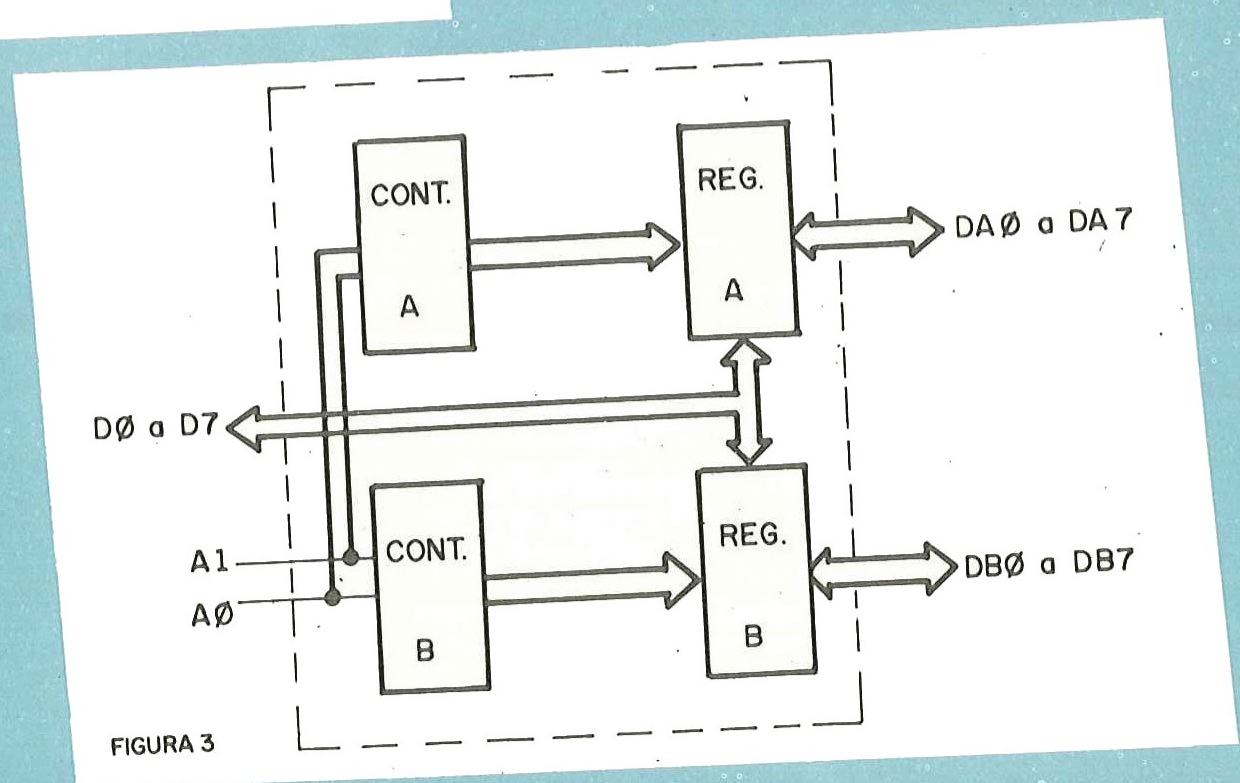


FIGURA 3

Para indicar uma operação de leitura ou escrita na PIO-Z80 necessita-se de um pequeno circuito conversor, pois o microcomputador utiliza duas saídas para indicar uma operação de leitura ou escrita ($\overline{WR}$ e $\overline{RD}$) e a PIO-Z80 só tem uma entrada ($\overline{RD}$) para indicar estas operações. A figura 2 apresenta o circuito conversor e a sua tabela verdade.

Note que no circuito da figura 2 o sinal EN é ligado à entrada de habilitação da PIO, que só é habilitada quando $\overline{WR}$ ou RD são diferentes de 1, isto é, a PIO só é habilitada durante uma operação de leitura ou escrita.

Para a gravação de dados na porta A da PIO-Z80 utiliza-se os seguintes passos:

1º Passo: Habilita-se o controle de A, através do byte L (byte de baixa ordem), com o endereço 00 00 00 10.

2º Passo: Habilita-se o controle bit a bit em A, através do barramento de dados, por 11 XX 11 11, onde XX representa qualquer estado lógico.

3º Passo: Habilita-se os *tri-states* internos da porta A, através do barramento de dados, "1" (um) para saída de dados e "0" (zero) para entrada de dados.

4º Passo: Habilita-se o registro A, através do byte L, com o endereco 00 00 00 00.

5º Passo: Escreve-se o dado a ser utilizado no registro A.

Para gravar dados na porta B deve-se mudar os passos 1 e 4.

A figura 3 apresenta o diagrama em bloco interno (simplificado) da PIO-Z80.

Outra parte importante a ser mencionada é o valor de corrente que a PIO pode fornecer. Como alguns CI's consomem até 120 mA e a PIO não consegue fornecer esta corrente, necessita-se de um amplificador de corrente. A figura 4 apresenta o circuito de um amplificador de corrente.

No circuito onde Q1 é um transistor PNP e Q2 é NPN, pode-se utilizar qualquer transistor que forneça uma corrente em torno de 150 mA. O resistor R é utilizado para testar CI's de coletor aberto, no caso do 7405. A entrada E é a entrada do amplificador de

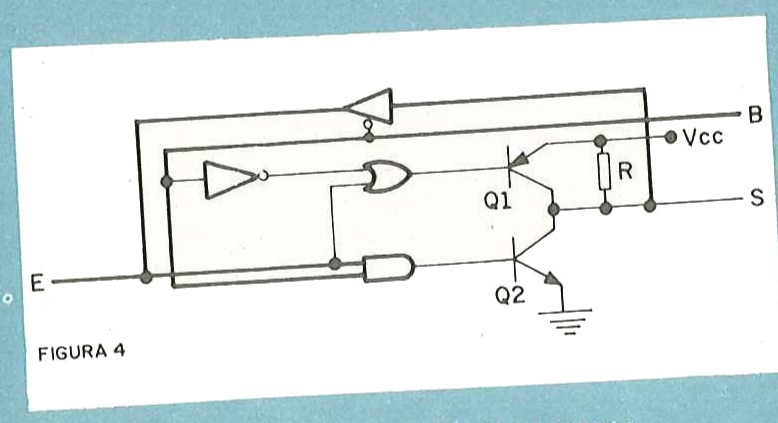


FIGURA 4

Figura 4 - Amplificador de corrente.
Figura 5 - Decodificador 1 e 2.
Figura 6 - Monoestável
Figura 7 - Circuito dos latches.

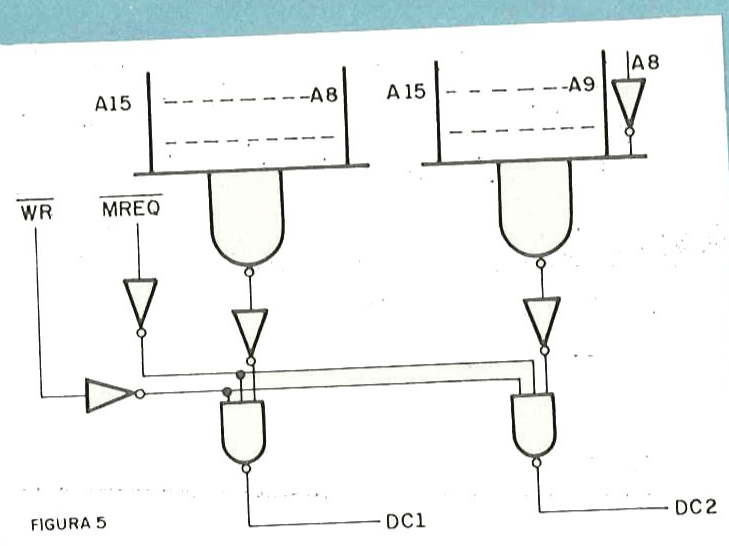


FIGURA 5

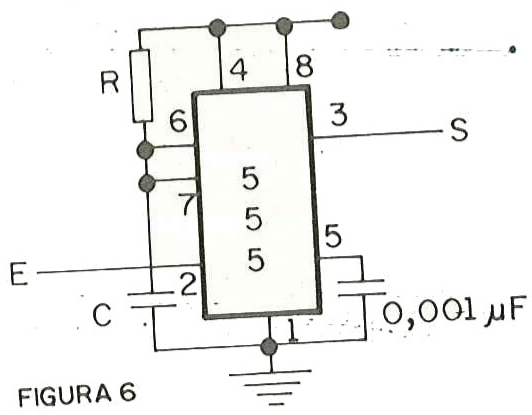


FIGURA 6

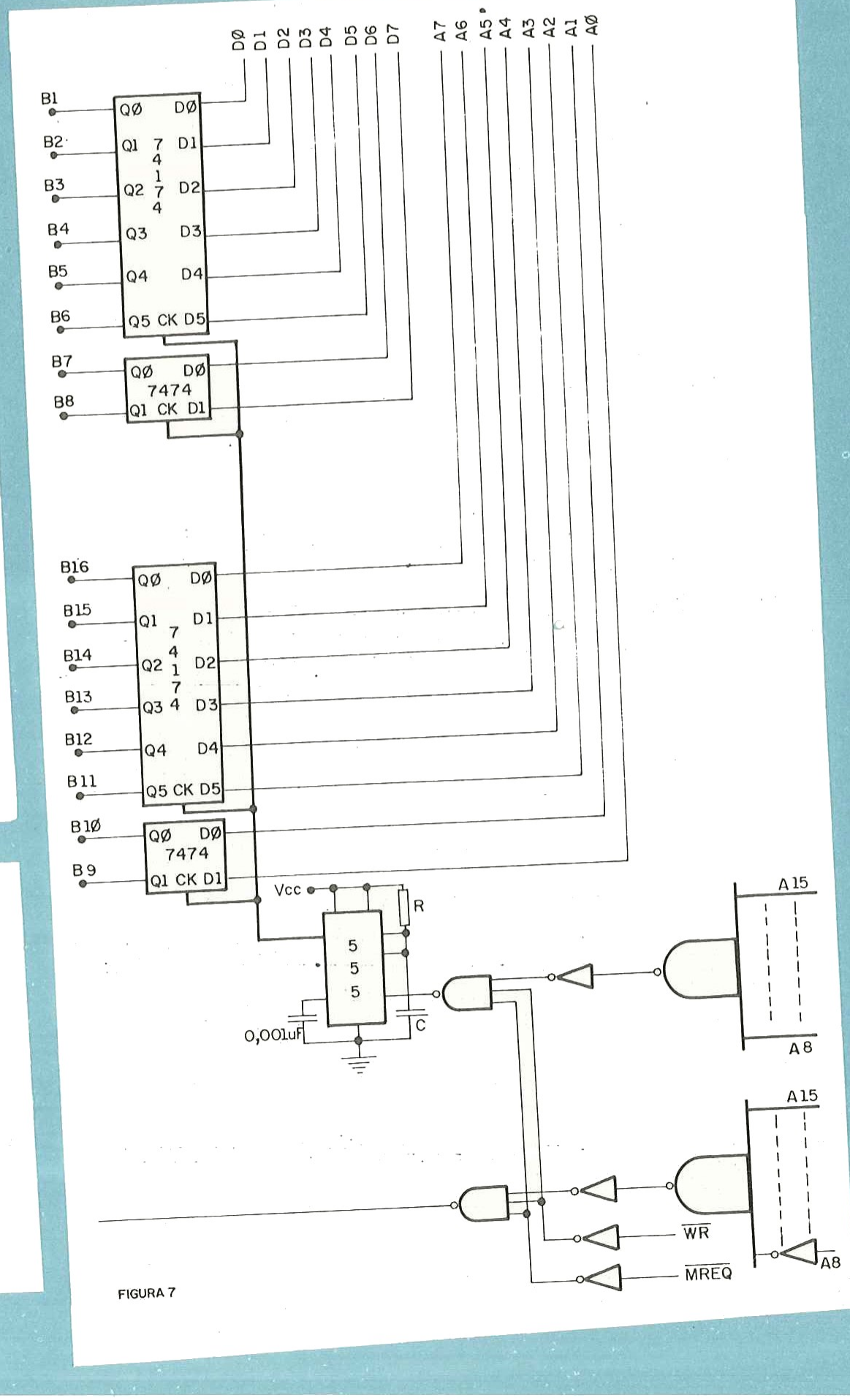


FIGURA 7

corrente e provém das portas da PIO; S é a saída do amplificador de corrente, sendo conectada diretamente aos terminais do Circuito Integrado a ser testado; e B provém dos flip-flops(latches), sendo responsável pela definição do amplificador de corrente como uma entrada ou saída de dados.

Devido ao fato de que os terminais de alimentação dos CI's a serem testados variam de CI para CI, necessita-se repetir o circuito de amplificação de corrente para cada saída de cada porta da PIO-Z80 (16 vezes).

Os decodificadores 1 e 2 são constituídos por um conjunto de duas portas NAND de oito entradas, cinco portas inversoras e duas portas NAND de três entradas. A figura 5 apresenta os decodificadores 1 e 2.

A saída do decodificador 1 é conectada à entrada de um monoestável, de maneira que quando for realizada a decodificação o monoestável será ativado, produzindo um pulso de nível alto e de pequena duração. Este pulso é utilizado pelos flip-flops para carregar os dados, que se encontram em suas entradas e necessitam permanecer por um determinado tempo, após o pulso do monoestável ter retornado ao nível "0".

Pode-se utilizar como monoestável o 555. A figura 6 apresenta uma configuração; note que R e C são calculados em função do tempo em que o pulso deve permanecer em nível alto.

O circuito dos latchs é constituído de flip-flops tipo D, por serem de fácil aquisição e de baixo custo. Como são necessários dois conjuntos de flip-flops, onde cada conjunto utiliza oito flip-flops, optou-se por utilizar dois 74174 e dois 7474. A figura 7 apresenta o circuito dos latchs (também poderia utilizar-se dois 8212).

A figura 8 apresenta o diagrama elétrico de todo o circuito do Testador de Circuito Integrado (TCI). O leitor que possuir um conhecimento razoável de eletrônica digital poderá implementar e ampliar este Testador de Circuitos Integrados, de maneira que o mesmo possa testar CI's de 24 pinos. Para isso aconselhamos a utilização de uma outra PIO-Z80.

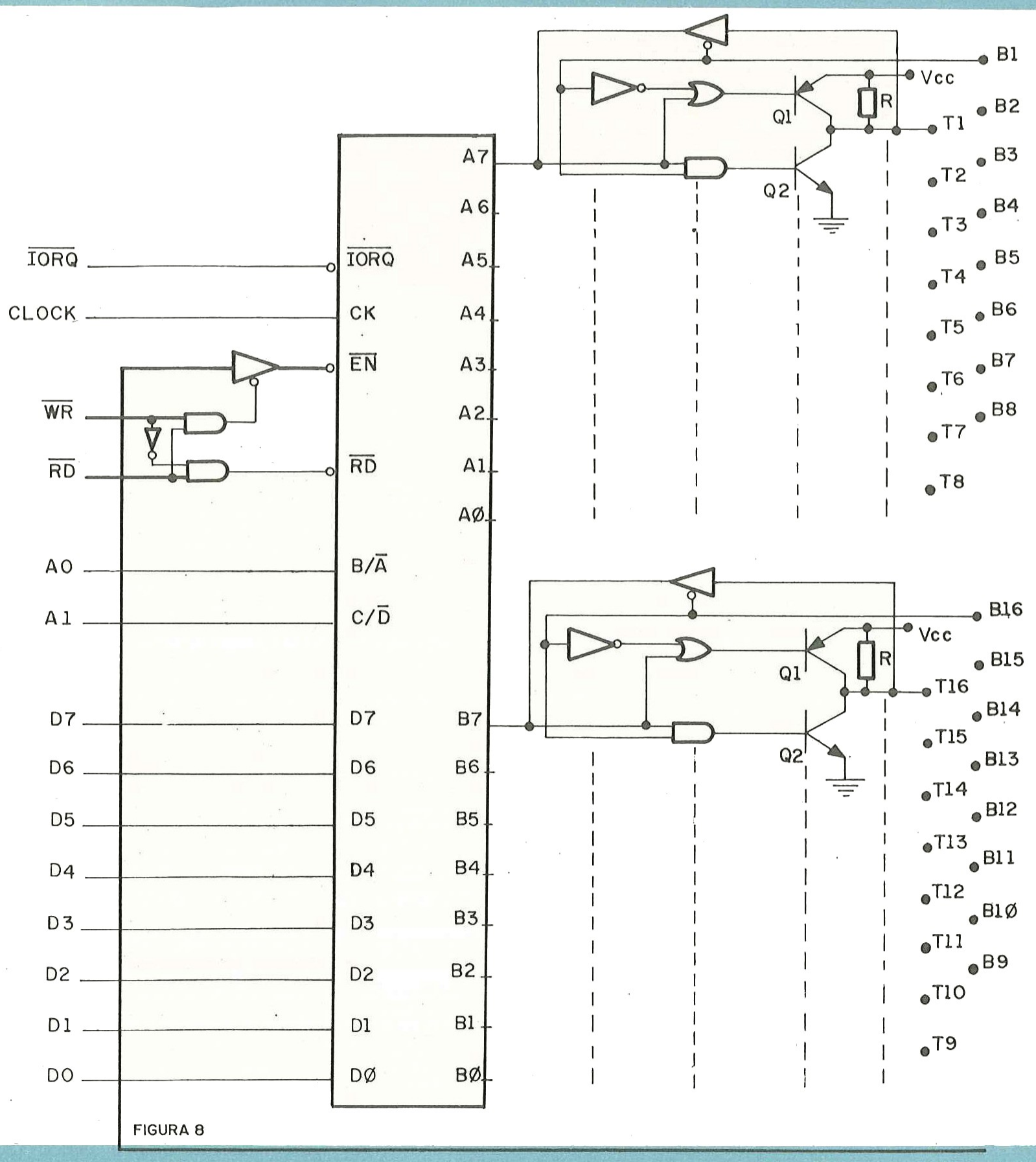


FIGURA 8

Figura 8 - Diagrama elétrico do TCI

filcres



SISTEMA 600

- 64 K DE MEMÓRIA
- DUAS UNIDADES DE DISCO
- ACOMPANHA OS APLICATIVOS: DB II, CALCSTAR E WORDSTAR
- ACOMPANHA IMPRESSORA P-600 DE 132 COL. E 130 CPS

Kapram PROPAGANDA

Na Filcres quando você adquire um microcomputador, pessoal ou profissional, você não só leva seu equipamento como também a certeza de um suporte em software (cursos de programação), hardware (assistência técnica) e uma completa linha de acessórios e periféricos ao seu dispor, para um bom desempenho de seu equipamento.

Peça a visita de nosso representante.

INSTRUMENTOS

- OSCILOSCÓPIOS
- MULTÍMETROS DIGITAIS
- FREQÜENCÍMETROS
- GERADORES
- TRAÇADORES GRÁFICOS

APROVEITE

- PROMOÇÕES ESPECIAIS
- FINANCIAMENTO
- BLOCK-TIME
- LEASING
- CONSÓRCIO
- CARTÕES DE CRÉDITO

Filcres Importação e Representações Ltda.

Rua Aurora, 165 — CEP 01209 — São Paulo — SP

Telex 1131298 FILG BR — PBX 223-7388 — Ramais 2, 4, 12, 18, 19 — Diretos: 223-1446, 222-3458, 220-5794 e 220-9113

Reembolso VARIG — Direto: 222-0016, 220-7718

BALCÃO

☐ ESQUEMA ELÉTRICO NEZ-8000 —Esquema elétrico completo (com slow e expansão de memória). Cheque nominal de Cr$ 9.500,00, para Jan Martin Lund.
Rua Frederico Ozanan, 16 – Bloco A/21 – CEP 11100 – Santos-SP.

☐ VÍDEO REVERSO E JOYSTICK P/TK-82C —Coloco vídeo reverso e vendo joystick sob encomenda p/TK-82C.
Silvio (011) 247-8967.

☐ SLOW/HIGH SPEED — Instalo slow, alta resolução, High Speed nos micros do tipo Sinclair (CP200, TK, NE); vendo programas.
Luciano – Telefone: 523-3921 (à noite).

☐ VENDO CP-500 —Vendo CP-500 da Prológica. Novo, com garantia. Versão cassete; por Cr$ 950.000,00.
Heloisa – Caixa Postal 90878 ou Telefone: 325-0872 – Rio de Janeiro-RJ.

☐ MULTIPLEXADOR PARA IMPRESSORA — Vendo multiplexador para impressora – Interface – paralela, segundo a convenção Centronics.
Flavio R. Paulon – Rua Flávio Fongaro, 94 – V. Marlene – São Bernardo do Campo-SP – CEP 09700 – Telefone: (011) 458-8360

☐ TROCO/VENDO PROGRAMAS — Troco ou vendo programas para o TK82-C. Possuo vários programas importados e em linguagem de máquina.
Edison Bueno da Silva – Av. Manoel Pontes Junior, 295 – Vila Rami – Jundiaí-SP – CEP 13200.

☐ PROGRAMAS – TROCO, COMPRO OU VENDO —Troco, compro ou vendo programas para os micros TK82/83/85, CP200 e similares. Jogos e aplicativos.
Ricardo Pieranti – Rua Cap. Cesar de Andrade, 168/806 – CEP 22431 – Rio de Janeiro-RJ.

☐ TROCO PROGRAMAS — Troco programas p/TK, NE e Sinclair. Mande uma fita com dois programas e receba-a de volta com mais dois programas à sua escolha. Escreva anexando uma lista de seus programas e receba a minha lista para sua escolha.
Carlos Souza Oliveira – Rua Artur Prado, 57/113 – CEP 01322 – Bela Vista – São Paulo-SP – Telefone: 289-3416.

☐ TROCO INFORMAÇÕES E JOGOS —Desejaria trocar informações e jogos com pessoas que tenham micros compatíveis com o DGT-100, com gravador. Tenho jogos em Assembler e Basic.
André Luís Nascimento Vianna – Rua Cap. Enilton França, 106 – CEP 21941 – Galeão-RJ – Telefone: 393-2520 e 393-7374 (de preferência à noite).

☐ PROGRAMAS EM K7 —Vendo os seguintes programas em fitas K7, para ZX's, TK's e CP-200: "Monstro das Trevas", "TXadrez I", "Forca", "Graphmaker I", "Jornada nas Estrelas" e listagem de outros 50. Solicite informações.
Jorge Eider – Village dos Mares – Quadra 18 – Bloco B – Casa 15 – Capim Macio – CEP 59000 – Natal-RN.

☐ VÍDEO REVERSO —Colocamos vídeo reverso, acionador de gravador, funções especiais e gerador de sons, nos micros TK82/83/85 e CP200.
Wellington – Rua Tomaz Acioly, 1053 – CEP 60000 – Fortaleza-CE – Telefone: 224-2776.

☐ 1º CLUBE NACIONAL DO ZX-SPECTRUM —Troca de Programas; trocas de informações; transcodificações para o sistema brasileiro e outros.
André Luiz Behrensdorf Derraik – Rua Senador Vergueiro, 198/101 – Flamengo – Rio de Janeiro – CEP 22230 – Telefone: (021) 551-1828.

☐ PROGRAMAS PARA MICROS —Vendo, compro ou troco programas para microcomputadores TK-82, TK-85, NEZ-8000, ZX-81 ou CP-200. Aplicativos, comerciais e de jogos animados, nacionais ou importados. Possuo 160 programas diferentes.
Carlos Sciarretti – Caixa Postal 5567 – CEP 01051 – São Paulo-SP – Telefone: (011) 522-8586.

☐ PROGRAMAS PARA LINHA SINCLAIR — Venda e troca de programas para TK-85, CP-200 e compatíveis com linha Sinclair. Solicite relação
HIGH-SPEED SOFT – Caixa Postal 44 – CEP 07000 – Guarulhos-SP.

☐ PROGRAMAS APLICATIVOS & JOGOS — Vendo programas aplicativos e jogos inteligentes e animados p/os micros CP200, TK82-85, Sinclair etc. Faço adaptação de auto-resolução gráfica do tipo T-Plack e Função H.Speed por Hardware.
Mario Vicente – Rua Galvão Bueno, 470 – Liberdade – São Paulo-SP – Telefones: (011) 270-8574 ou 278-9504.

☐ VENDA DE PROGRAMAS — Vendo programas de origem européia para TK82, TK83, TK85, CP200 e APPLY300, lacrados em plástico e contendo instruções de uso. Para o SPECTRUM: programas em K7, tais como o "Frogger", "The Tomb of Drácula", "Simulador de Vôo" e outros. Os jogos têm gráficos coloridos em alta resolução e som.
Alexandre – Caixa Postal 17005 – CEP 02399 – Telefone: 203-4277 – Tremembé-SP.

☐ VENDA DE TK82-C —Vendo um microcomputador TK82-C com expansão para 16KRAm e 6 meses de uso, por Cr$ 100.000,00. Acompanha uma fita com vários programas, entre eles "Defensores", "Monstro das Trevas", "Invasores do Espaço", etc.
Sérgio Luiz da Silva – Av. Conselheiro Nébias, 710 – Telefone: 35-1125 – Boqueirão – Santos-SP, horário comercial.

☐ FITA COM JOGOS —Vendo uma fita com os seguintes programas: "Monstro das Trevas", "Pac-Man", "Redalert", "Simulador de Vôo II", "Space Invaders"; "TXadrez I ou II", "Defensor 3D", "Labirinto 3D", "Asteróides" e "TK-Man". Para TK-82, TK-85, CP-200 etc., por apenas Cr$ 12.000,00.
João Bosco de Castro – Rua Licinio Leite Machado, 59 – Santana – São José dos Campos-SP – CEP 12200 – Telefone: (0123) 22-0420.

☐ MICRO CLUBE —Aos usuários de micros compatíveis com o Sinclair, anunciamos a criação do *Micro Clube*, que distribuirá 100 programas aos associados. Dispomos de uma Biblioteca de 200 programas, além de kits para os micros TK e CP.
Wellington – Rua Tomaz Acioly, 1053 – Fortaleza-CE – CEP 60000.

☐ INTERFACE PARA DECODIFICAÇÃO DE CW —Vendo interface para decodificação de CW, para micros com tecnologia Sinclair.
Renato Strauss PY2-EMI – Rua Cardoso de Almeida, 654/32 – CEP 05013 – São Paulo-SP.

☐ JOGOS EM CASSETE —Vendo uma fita cassete com pequeno manual explicativo, contendo 10 (dez) programas de 1K de memória. Todos de minha autoria, dos quais 7 (sete) são jogos. Podem ser carregados em qualquer equipamento da linha Sinclair, com ou sem slow.
Marcos – Telefone: (011) 63-3324 – São Paulo-SP.

☐ TK85 – COMPRO —Estou interessado na compra de um TK85 em bom estado, com vídeo ou não. Márcio de Carvalho – Rua Alagoas, 392 – CEP 15800 – Catanduva – SP – Telefone: (0175) 22-1098.

☐ VENDA DE PUBLICAÇÕES
Nova Eletrônica do nº 3 ao 81, exceto o 37; *Interface* do nº 1 ao 10; *Data News*, do período de Maio/81 a Maio/83, com os suplementos especiais sobre Automação de Escritórios, Mercado de Informática no Sul, etc; *Micromundo*, do nº 1 ao 3 quando revista e, do nº 1 ao 18 quando jornal; *Experiências e brincadeiras com Eletrônica*, 3º Vol.; livro "Guia para Programadores", de Marilyn Bohl; livro "Programação Cobol", de Alex C. Bastos; revista *Dados e Idéias*, edição especial sobre o mercado de informática: revista *Flap Internacional*, nº 106 e 108; revista *Aviação*, do nº 477 a 492; *Micro Sistemas* nº 6; *Som Três*, nºs 16, 17, 25, 31, 34 e edição especial sobre Caixas Acústicas e Auto-falantes; folhetos e catálogos das últimas Feiras de Informática e Eletrônica; blocos de programação RPG e COBOL. Preços a combinar.
Silvio – Telefone: 293-4608 – São Paulo.

☐ PROGRAMAS PARA LÓGICA SINCLAIR —Vendo programas para o TK82-C, TK-85, CP-200 e qualquer computador da lógica Sinclair. Jogos, educativos, utilitários, etc. Possuo mais de 160 programas; também vendo xerox de livros importados e faço programas sob encomenda.
Marcelo Rodrigues – Rua J. Carlos, 90/701 – CEP 22461 – Rio de Janeiro-RJ – Telefone: (021) 286-4765.

CONSULTORIA

DIFERENÇAS ENTRE O BASIC E O TRS-80, APPLE E SINCLAIR

Nossos consultores respondem, neste número, a três questões: uma, sobre diferenças entre o Basic e outros micros; a segunda, a função do Basic que transforma um certo valor em seu correspondente ASCII; a terceira, como implementar um bit por software num microcomputador pessoal.

Característica	Sinclair	Apple	TRS-80
If. . . Then. . . Else	não	não	sim
If. . . Goto. . .	não	sim	sim
INPUT (cadeia);. . .	não	sim	sim
AND/OR	sim	não	sim
CLS (clear screen)	sim	não	sim
OPEN	não	não	não
CLOSE	não	não	não
PRINT #	não	PR#	PRINT #
INPUT #	não	IN#	INPUT #
SAVE/LOAD	vide manual	vide manual	vide manual
INKEY$/GET	INKEY$	GET	INKEY $
Tecla substituindo comando	sim	não	não
Precisão	9 dígitos	9 dígitos	17 dígitos
Números	entre 10^{-39} a 10^{38}	entre 10^{-39} a 10^{38}	entre 10^{-39} a 10^{38}
Variáveis	1 letra + 1 dígito	até 238 caracteres	2 letras ou 1 letra + dígito

P *Gostaria de saber quais as principais diferenças entre os* Basics *que rodam nos micros compatíveis com TRS80, APPLE e SINCLAIR.*

R As principais diferenças, em termos de comando, entre os *Basic's* destes micros são apresentadas na tabela acima.

P *Qual a função do basic que transforma um certo valor em seu correspondente ASCII, e se possível publicar um programa que gere uma tabela dos caracteres ASCII.*

R A função CHR$ do basic transforma um certo valor em seu correspondente ASCII. Apresentamos a seguir um programa de teste para gerar uma tabela dos caracteres ASCII, que lista em colunas os códigos numéricos e os caracteres correspondentes. A saída do programa é também apresentada. Por exemplo, o número decimal 121 (linha 120 com coluna 1) representa o ca-

Microchip
COMPUTAÇÃO

- Vendas
- Cursos de Basic
- Basic Avançado
- Fitas
- Disquetes
- Software

EM COPACABANA
Rua Miguel Lemos, 41/606
Tel. 227-8803
Rio de Janeiro — RJ

MICRO WARE
MICRO WARE
O MICRO DO CHOPPING ICARAÍ

CURSOS
INICIAÇÃO AOS MICROCOMPUTADORES
BASIC BÁSICO E AVANÇADO
ASSEMBLER

VENDAS
• MICRO-COMPUTADORES • LIVROS • REVISTAS
• PROGRAMAS • FITAS • DISQUETES • SUPRIMENTOS

CONSULTORIA
DESENVOLVIMENTO DE HARDWARE
DESENVOLVIMENTO DE SOFTWARE
ESPECIFICAÇÕES DE SISTEMAS

ASSISTÊNCIA TÉCNICA
EM NITERÓI
Rua Moreira Cesar, 229 Conjunto, 1713
Tel.: (021) 710-2780

Consultoria Projetos
Profissionais do ramo de Informática/Eletrônica oferecem-se para consultoria de Hardware e/ou Software ou Desenvolvimento Integral de Projetos
CONTATOS:
258-4171/Eng.ª Jane
Eng.º João
284-6014/Eng.º Jorge

```
*** TABELA ASCII ***
+-----------------------------------------------------+
!     ! 0 ! 1 ! 2 ! 3 ! 4 ! 5 ! 6 ! 7 ! 8 ! 9 !
+-----------------------------------------------------+
! 30  !   !   !   !   !   !   !   !   !   !   !   !   !
! 40  ! ( ! ) ! * ! + ! , ! - ! . ! / ! 0 ! 1 !
! 50  ! 2 ! 3 ! 4 ! 5 ! 6 ! 7 ! 8 ! 9 ! : ! ; !
! 60  ! < ! = ! > ! ? ! @ ! A ! B ! C ! D ! E !
! 70  ! F ! G ! H ! I ! J ! K ! L ! M ! N ! O !
! 80  ! P ! Q ! R ! S ! T ! U ! V ! W ! X ! Y !
! 90  ! Z ! [ ! ' ! ] ! ^ ! _ ! ` ! a ! b ! c !
! 100 ! d ! e ! f ! g ! h ! i ! j ! k ! l ! m !
! 110 ! n ! o ! p ! q ! r ! s ! t ! u ! v ! w !
! 120 ! x ! y ! z ! { ! | ! } ! ~ !   !   !   !
Ok
```

ractere y (minúsculo).

P *Como implementar um bip por software, num microcomputador pessoal?*

R A instrução *PRINT CHR$ (7)* quando executada pelo microcomputador fará soar um bip, se ele estiver conectado a um alto-falante. A função CHR$ transforma um certo valor, no caso o decimal 7, em seu correspondente ASCII, BEL, que toca o bip.

```
10  REM ********************************************************
20  REM *                                                      *
30  REM *       PROGRAMA PARA GERAR UMA TABELA ASCII           *
40  REM *                                                      *
50  REM *       AUTOR: JOSE ANTONIO BORGES                     *
60  REM *                                                      *
70  REM *       EM JULHO/83                                    *
80  REM *                                                      *
90  REM *       VARIAVEIS:                                     *
100 REM *               L       CONTROLA AS LINHAS           *
110 REM *               C       CONTROLA AS COLUNAS          *
120 REM *               T$      LINHA DE TRACINHOS           *
130 REM *                                                      *
140 REM ********************************************************
150 REM
160 REM *** IMPRIME O CABECALHO ***
170 REM
180     PRINT
190     PRINT "*** TABELA ASCII ***"
200     PRINT
210     T$ =  "+-----------------------------------------------+"
220     PRINT T$
230     PRINT "!     ! 0 ! 1 ! 2 ! 3 ! 4 ! 5 ! 6 ! 7 ! 8 ! 9 !"
240     PRINT T$
250 REM
260 REM *** IMPRIME AS LINHAS DA TABELA
270 REM
280     FOR L = 30 TO 120 STEP 10
290         REM *** IMPRIME O NUMERO DA LINHA ***
300         PRINT "!"; L; TAB(7); "! " ;
310         REM *** IMPRIME OS CODIGOS ***
320         FOR C = L TO L + 9
330             K$ = " "
340             IF C >= 32 AND C <= 126 THEN   K$ = CHR$(C)
350             PRINT K$; " ! ";
360     NEXT C
370     PRINT
380     NEXT L
390 REM
400     PRINT T$
410 END
```

— Programa para gerar tabela ASCII.

sistema

ASSISTÊNCIA TÉCNICA

☐ Assistência a Micros Nacionais
Todas as marcas e modelos Importados:
Sinciair - Trs-80 - Apple-IBM

Outras marcas poderão ser atendidas

Assistência Técnica autorizada
PROLOGICA microcomputadores

Av. Presidente Vargas, 542
Sala 2111 - Tel.: 253-0645
Rio de Janeiro

Proteja-se da Pirataria



- Impossibilita a Utilização do Programa por Usuário não Autorizado
- Permite Cópias de Segurança

PROTESIS®

Um Produto da
Engesis Engenharia e Sistemas Ltda.
Rua Anchieta, 18, cj. 401,
tel. (011) 36-0389
CEP 01016 — São Paulo — SP
* Pat. Req.

ASSEL

ASSISTÊNCIA ELETRÔNICA LTDA.

ASSISTÊNCIA TÉCNICA
CALCULADORAS ELETRÔNICAS
MICROCOMPUTADORES E ACESSÓRIOS
AUTORIZADO: TEXAS E DISMAC

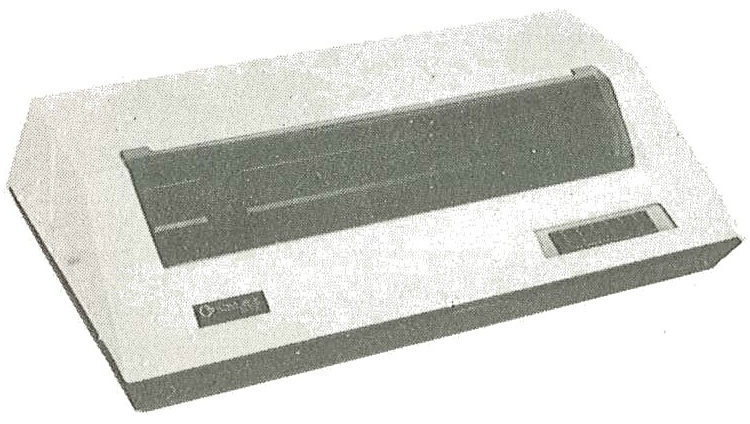
RUA DA LAPA, 107 - 10º AND. - RIO - RJ.
TELS.: 222-7137 E 222-2278

AV. MINISTRO EDGARD ROMERO, 81
- MADUREIRA - SOBRELOJA 307

GLOBUS

Ideal para qualquer microcomputador existente no mercado.

Impressora M-100 X de 132 colunas



Impressora matricial de impacto com velocidade de 100 cps e impressão de 132 colunas.

Finalmente em produto Nacional com performance inigualável:

- Velocidade real de 100 cps
- Cabeça de impressão de 9 agulhas
- Sustentador de cabeça de impressão em duplo eixo
- Capacidade gráfica
- Flexibilidade de software
- Interface dupla: Paralela e Série
- Compatibilidade Epsom e pacotes Apple II

A Globus produz linha completa de impressoras para todas as aplicações, além de dispor de peças de reposição para entrega imediata.

GLOBUS DIGMAL S.A.
EQUIPAMENTOS PERIFÉRICOS

MUITO MAIS DO QUE MÁQUINAS SIGNIFICA SERVIÇO

Matriz e Fábrica: Rua Isidro Rocha, 1057
CEP 21241 - Tel.: 372-4385 - Telex (021) 31991
GLDG - BR - Vigário Geral - RJ - Brasil
Filial: Rua Stella, 515 - Bloco H - Grupo 21
CEP 04011 - Tels.: (011) 231-2586/571-4642
Vila Mariana - São Paulo - SP

VASTA REDE DE DISTRIBUIÇÃO

Polícom/ZEZ

LIVROS

Basic - Aplicações Comerciais. José Antônio Borges. 1.ª Edição. 208 páginas - Cr$ 4.500,00. L.T.C.

A abordagem geral do livro se baseia em exemplos voltados para aplicações do tipo gerencial. Partindo de programas completos, que são minuciosamente analisados, o leitor é guiado ao entendimento do computador e da atividade de programação. Essa abordagem é exatamente oposta aos padrões habituais de livros de linguagens de programação.

O livro cobre minuciosamente todos os detalhes da linguagem BASIC incluindo os aspectos que não são padronizados nas diversas versões para diferentes computadores. Especialmente, o capítulo sobre arquivos fornece a base para desenvolvimento das aplicações de uso na administração em geral. Embora a fase seja no BASIC para micros, esse texto poderá ser utilizado para formação de programadores em BASIC para outros computadores de maior porte.

VELOSO, P.S.; SANTOS, Clesio dos, AZEREDO, Paulo e FURTADO, Antônio - ESTRUTURAS DE DADOS - Editora Campus

Os computadores servem para armazenar informação e programas para manipulá-la. Assim, um programador consciente não pode ignorar a importância dos dados e como estruturá-los.

Este livro apresenta as estruturas de dados mais importantes para a "programação não numérica", servindo como texto ou referência para cursos de graduação ou, com o possível acréscimo de algum material, de pós-graduação.

A apresentação reflete a experiência dos autores em lecionar este assunto durante vários anos, principalmente na PUC-RJ e na UFRGS, onde versões iniciais do livro foram utilizadas como notas de aula. De grande interesse para alunos e professores são os inúmeros exercícios propostos, com graus de dificuldade bastante variados.

ABREU, Carlos Alberto e REIS, Denise - CURSO DE BASIC volume I - Micro-kit

O curso de Basic Volume I é o resultado da experiência de aproximadamente oitenta cursos de Linguagem Basic, promovidos pela Micro-Kit Educacional.

Segundo os autores, este volume é dedicado à linha Sinclair por ser mais popular aos que iniciam.

O livro é composto por dez aulas, e no final de cada uma são propostos exercícios de verificação de aprendizado, solucionados.

A primeira aula é uma introdução à programação e, em seguida, são explanados os elementos de linguagem Basic, conduzindo o leitor a elaborar um programa aplicativo passo-a-passo.

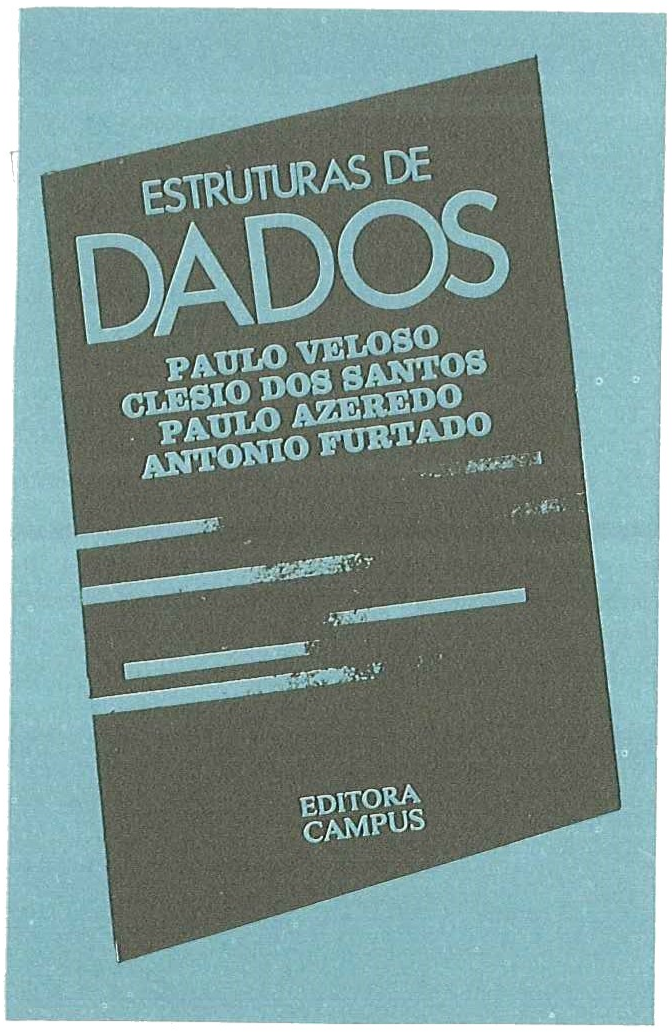
A parte final é dedicada aos gráficos e desenhos na tela para jogos.

Como é comum surgir muitas dúvidas na transformação de comandos de outras máquinas para os da linha Sinclair, foi criado o anexo I, que dá DICAS de como proceder em tais casos.

O ponto alto deste livro é o planejamento das aulas, com programas-exemplos correspondentes à parte teórica, o que possibilita ao leigo aprender rapidamente a Linguagem Basic.

O curso de Basic Volume I é uma excelente ferramenta para o aprendizado de programação de computador da linha Sinclair, e está planejado para um curso de vinte horas.

Este é o primeiro livro de uma série a ser lançada.

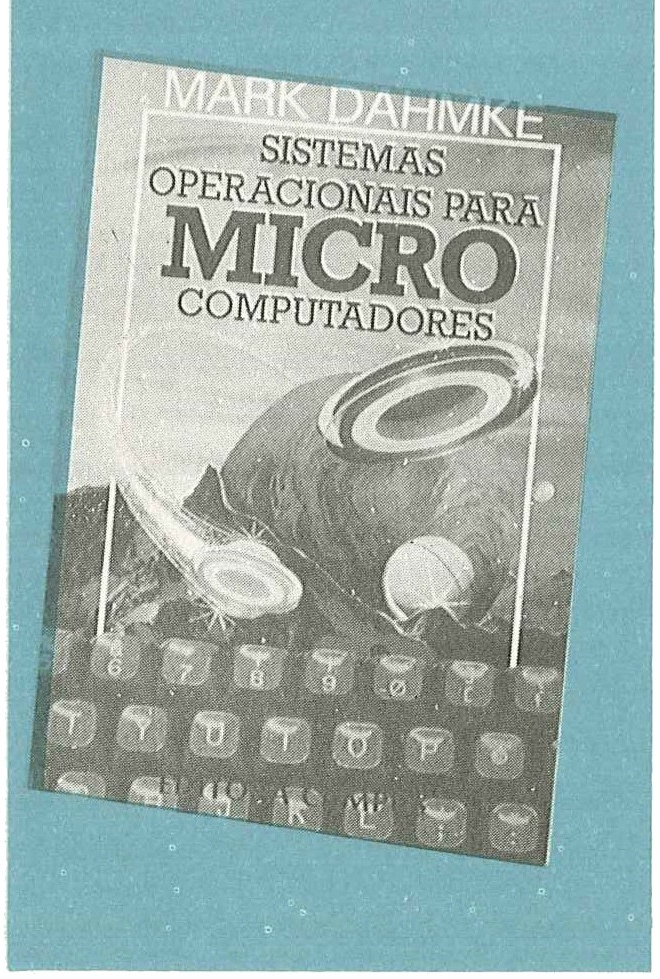


MIRSHAWKA, Victor - BASIC SEM SEGREDOS - Editora Nobel

BASIC SEM SEGREDOS" é o livro que dará ao leitor as condições de conhecer a linguagem Basic e suas aplicações. O Basic, uma das linguagens de computação caracterizada pela simplicidade e facilidade de aprendizado, constitui-se na grande porta de entrada da Informática. Usado em todos os tipos de computadores – micros, minis, médios e grandes – o Basic permite a aproximação, o uso e o entendimento desta nova tecnologia, principalmente no que diz respeito aos microcomputadores.

Este livro apresenta à linguagem Basic em 3 estágios, elementar, intermediário e avançado, tendo como objetivo final ensinar a programar. Através de exemplos e ilustrações, de uma forma clara e objetiva, o autor traz ao leitor os conceitos do Basic e um completo conhecimento da sintaxe da linguagem de programação – no caso, da linguagem Basic.

Dividido em 6 capítulos, "BASIC SEM SEGREDOS" aborda: Aplicações e um pouco de história sobre os computadores; A linguagem Basic Elementar; O Basic Intermediário – com destaque para o par Read-Data, as variáveis alfanuméricas e indexadas, e as funções matemáticas especiais; O Basic Avançado; O sofisticado MAXXI da linha APPLE e um capítulo abordando o uso do microcomputador na Educação, destacando os seus benefícios e alguns possíveis malefícios.



A Construção de um compilador. Autor: V.W. Setzer e I.S.H. de Melo. 200 páginas. CAMPUS.

Com finalidade didática e informativa, esta obra tanto poderá ser empregada como livro-texto para a disciplina de teoria e construção de compiladores, quanto como referência para profissionais de processamento de dados que desejem informar-se sobre este campo da Ciência da Computação e Informática.

Dividida em duas partes, trata inicialmente de análise léxica e sintática, apresentando o método de análise sintática simples e eficiente desenvolvido por V. W. Setzer. Aborda, em seguida, análise de contexto, geração de código e detalhes de implementação, sugerindo e orientando, como trabalho prático, a execução de um projeto completo de compilador para a Linguagem Pascal.

Sistemas operacionais para microcomputadores. Autor: M.Dahmke. 122 páginas. CAMPUS.

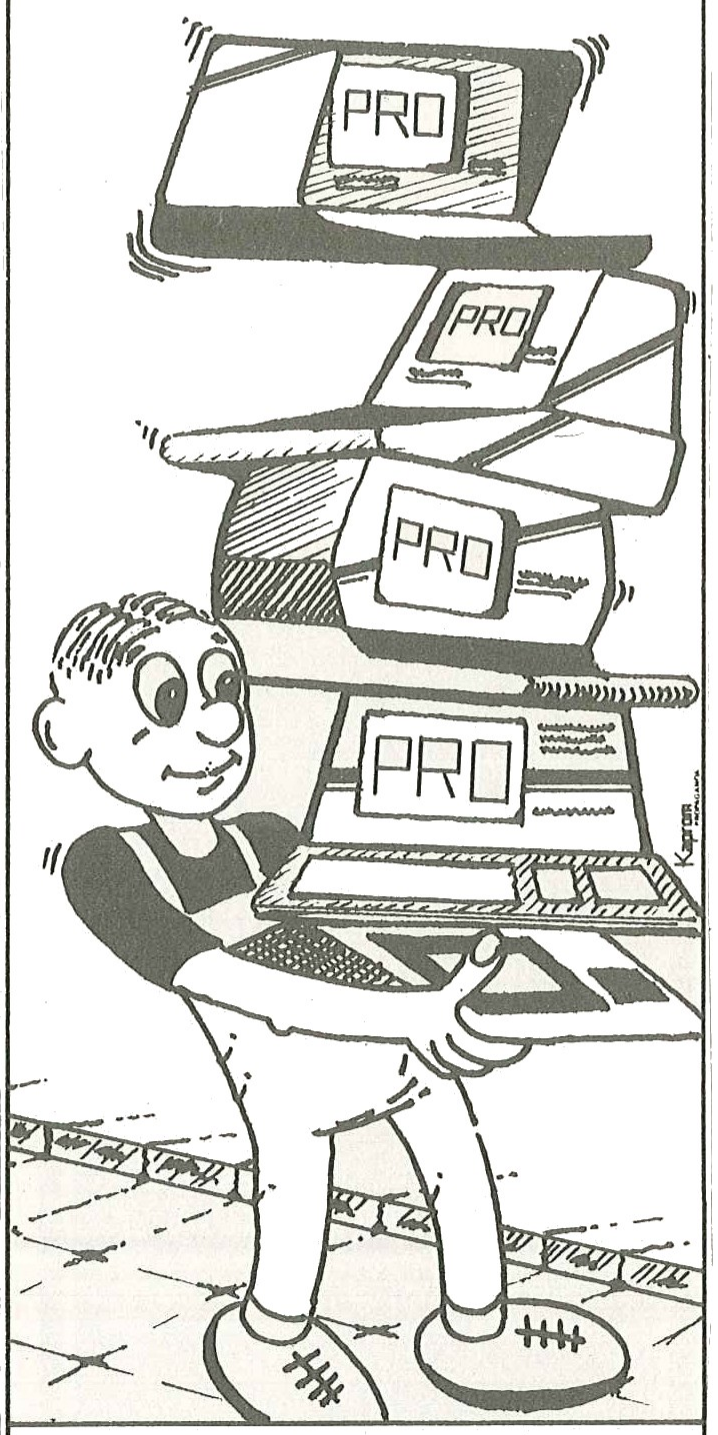
Texto interessante e original que aborda o assunto de forma inédita introduzindo o conceito de projeto de sistemas de microcomputadores do ponto de vista do software, em vez de enfocar os detalhes familiares do projeto do hardware.

Os aspectos envolvidos na criação de sistemas operacionais são discutidos e ilustrados com base nos sistemas atualmente disponíveis. Após uma breve revisão da tecnologia e da terminologia associadas com o moderno projeto de microprocessadores, a matéria começa a ser apresentada de forma evolutiva, descrevendo o modo pelo qual os sistemas operacionais dos grandes computadores apareceram e pelo qual foram adaptados para os modernos micros. Segue-se uma descrição das características e dos recursos de um sistema operacional bastante semelhante ao DOS, de uso comum hoje em dia pela maior parte dos microcomputadores, e detalhes sobre multiprogramação, multiprocessamento e redes.

O ENDEREÇO DE TODOS OS MICROS

Em nossa loja somos todos Pró-informática, Pró-didática e Pró-eletrônica.

Sysdata ZIROK FLEXIDISK
MICRODIGITAL DACTARI
Polymax Unitron
ELEBRA ACECO
PROLOGICA microcomputadores
apple



PRÓ ELETRÔNICA
COMERCIAL LTDA.
Rua Santa Efigênia, 568 – CEP 01207 – São Paulo – SP
Tels.: 220-7888 – 221-9055 – Telex (011) 34901 – POEC

INVESTIMENTO EM ESTOQUE GARANTE UM BOM LUGAR

Estoque bem abastecido e permanentemente atualizado são os fatores que Alberto Matos, presidente da Clappy considera importantes no crescimento da empresa que dirige, que atualmente ocupa a 89ª posição entre as 100 maiores da área de informática do país, sendo a única loja a figurar nesse ranking. "Nosso investimento maior é em estoque, levando ao cliente o benefício de encontrar sempre tudo aquilo que procura e custos altamente competitivos, já que compramos em escala e repassamos para o cliente as vantagens obtidas nas negociações com a indústria", afirma Alberto Matos. Com isso, seu faturamento deve chegar a 2,5 milhões em 83.

A Clappy é a maior loja do país voltada para a comercialização de microcomputadores e periféricos, embora tenha lojas apenas no Rio de Janeiro: em Copacabana e na Avenida Rio Branco, no Centro. Ambas têm o clima adequado ao usuário que vai escolher um sistema de computação. Todos os equipamentos podem ser testados e operados pelo cliente. Os vendedores são técnicos treinados, capazes de orientar o cliente para a melhor opção. Depois de escolhido o equipamento a equipe da Clappy oferece ao cliente opções quanto à forma de aquisição. Ele pode prestar assessoramento na montagem das operações de leasing de equipamentos, recomendando instituições financeiras com as quais usualmente trabalha nesse tipo de operação e oferece sistema de locação de equipamentos e programas para as empresas. Com isso as empresas podem testar sistemas de computação. Para as pessoas físicas, além da venda à vista, há a possibilidade de financiar as compras até em 24 meses.

A Clappy inicia agora a locação de programas e aplicativos e mantém sistema de informações sobre software, podendo encaminhar o cliente a escritórios e software-houses capazes de desenvolver os programas desejados.



LITEC
Livraria Editora Técnica Ltda.

A maior livraria da América Latina especializada em INFORMÁTICA, COMPUTAÇÃO E ELETRÔNICA.
Mais de 3.000 títulos em português, espanhol e inglês em permanente exposição.

Rua dos Timbiras 257 - 01208 São Pauio Tel. (011) 220-8983 cx. postal 30.869

APPLE II PLUS



PRODUTOS DISPONÍVEIS:

- Computador AP x II (equivalente APPLE II)
- Placa interface diskette 5¼
- Cartão 80 colunas (Videx)
- Cartão CP/M (Softcard Z80)
- Cartão interface dupla: sevial e paralelo
- Programador de EPROM (2708, 2716, 2732 2764; não precisa de unidade de disco)
- Monitor de video profissional de fósforo verde (video direto e modulador)
- Teclado compatível c/ Apple (Jan/84)
- Fonte chaveada (Jan/84)

FONE: (011) 276-9773

— Para pequenas quantidades, entrega imediata
Fábricado por:
SIGMA TECNOLOGIA ELETRÔNICA LTDA.

ENGESIS
Software

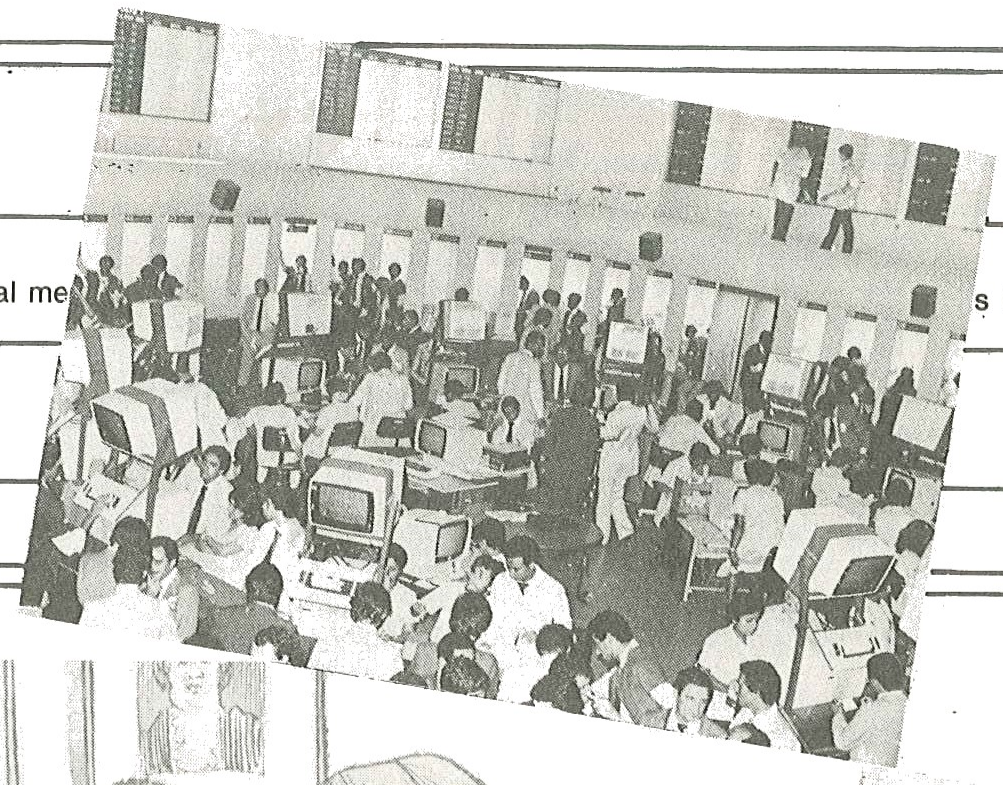
- Contabilidade
- Controle de Estoques
- Contas à Pagar
- Contas à Receber
- Agenda
- Mala Direta
- Arquivo Eletrônico (Banco de Dados)
- Orçamentação de Obras
- Tráfego de Elevadores
- Coletores Solares
- Planilha de Cálculo
- Gráficos Comerciais
- Jogos

- DESENVOLVIMENTO DE SOFTWARE
- IMPLANTAÇÃO DE SISTEMAS

Engesis Engenharia e Sistemas Ltda.
Rua Anchieta, 18, cj. 401,
tel. (011) 36-0389
CEP 01016 — São Paulo — SP

AÇÃO

Jornal me s Ltda.



JORNAL AÇÃO - Os dados que faltavam na sua memória

Em 1983, quem aplicou o seu dinheiro no mercado de ações teve, em média, um retorno acima de 400% e, em certos casos a valorização chegou a 5.000%, como as ações da Correia Ribeiro.

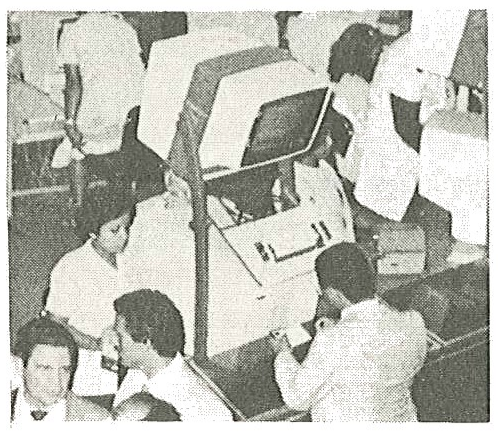
Com a introdução da informática as Bolsas de Valores, Corretoras, Bancos de Investimento e Distribuidoras estão aparelhadas para, com agilidade e segurança, rentabilizar o seu capital.

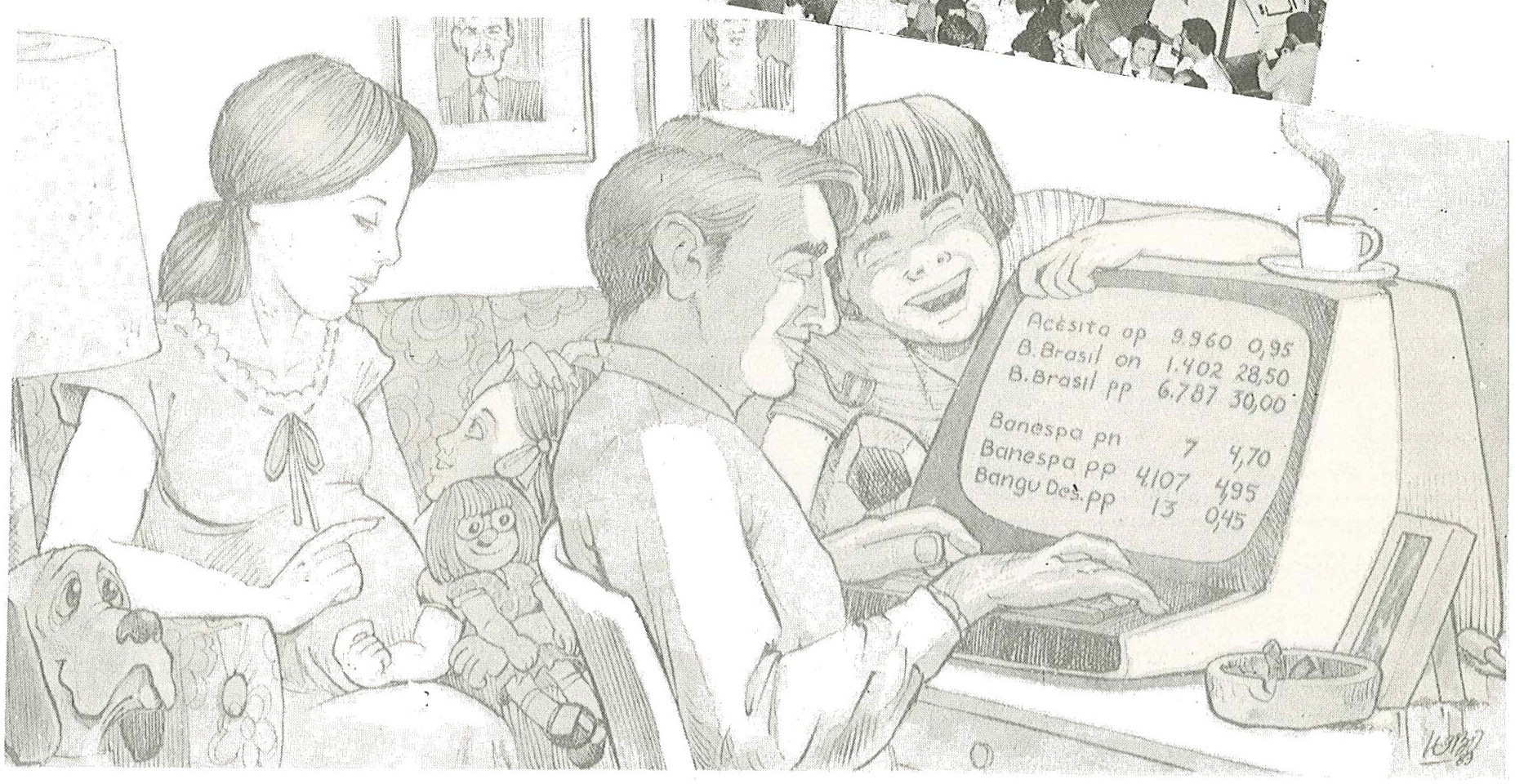
Hoje, aplicar no mercado oferece uma segurança, mas também envolve risco, sendo fundamental, para se obter os ganhos desejados, estar bem informado quanto às opções de investimento.

Por apenas Cr$ 10.000,00 anuais você faz uma assinatura do Jornal AÇÃO — o único jornal do mercado de capitais, terá resposta a todas as suas dúvidas, e mensalmente, as melhores informações para programar o seu investimento.

Há três anos o Jornal AÇÃO é o jornal que os participantes do mercado lêem e consultam. Suas páginas mostram a realidade que há por trás da notícia, analisam o desempenho das empresas, mostram os casos de sucesso, explicam como funcionam as entidades e quais são as opções de investimento. Seus colunistas, profissionais de larga experiência e alto conceito no setor, analisam e opinam sobre as técnicas das empresas, como investir com segurança e rentabilidade, os novos investimentos e como avaliar o risco de uma aplicação.

Preencha o cupom e envie um cheque de Cr$ 10.000,00 nominal à C&MARKETING COMUNICAÇÃO E ARTES GRÁFICAS LTDA., RUA DO CARMO Nº. 17, 10º ANDAR — CEP 20.011, CENTRO, RIO DE JANEIRO, RJ. Receba os dados que faltavam para você programar com segurança e rentabilidade o seu investimento.





A revolução silenciosa da Informática no mercado de capitais

A Informática, que já faz parte do dia-a-dia de executivos, donas de casa e mesmo crianças, poderá influenciar decisivamente o conceito de trabalho, ou seja, quebrar regras de horários e procedimentos. Simplesmente ela possibilitará que um profissional execute suas tarefas, através de um terminal de computador, sem precisar diariamente ir ao escritório. Isto a princípio pode parecer uma utopia, mas quando se verifica a revolução silenciosa que esta ciência causou na sociedade em um curto espaço de tempo, de nada se pode duvidar.

Acabar com pregão de ações de uma Bolsa, por exemplo, há alguns anos, não passava de uma quimera. No entanto, até o final deste ano a Informática poderá iniciar a quebra desta tradição centenária através da implantação do Telepregão da Bolsa de Valores do Rio de Janeiro. Este será um marco na história do mercado de capitais, setor onde a Informática mais se desenvolveu no Brasil.

Silencioso e eficiente, o Telepregão irá funcionar como uma espécie de extensão do pregão físico, ampliando a possibilidade de negócios da Bolsa do Rio para todo o País simultaneamente, através de terminais de vídeo. Isto significa que os investidores de outros estados poderão realizar os mesmos negócios que os do Rio estão efetuando.

Mais uma vez, portanto, a Informática quebra barreiras e estabelece novas regras. Para alguns técnicos, dentro de pouco tempo o mercado de capitais estará totalmente automatizado dando possibilidade às instituições que atuam no setor de prestarem serviços para outros estados sem ficarem restritas a seu lugar de origem.

Com o advento da Informática, portanto, o mercado de capitais obteve uma nova dinâmica, as corretoras desenvolveram uma nova forma de marketing e muitas outras mudanças virão até o ano 2000.

NOME _____

ENDEREÇO _____

CEP _____ CIDADE _____ ESTADO _____

TELEFONE _____ PROFISSÃO _____

EMPRESA ONDE TRABALHA _____ CARGO _____

DATA DE NASCIMENTO _____ DATA _____

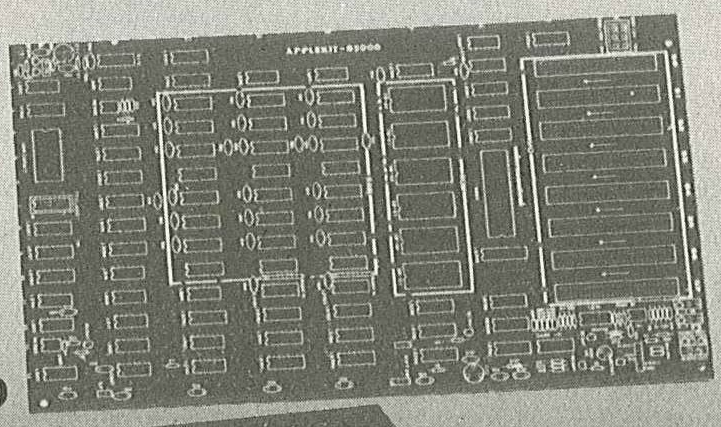
..... ASSINATURA _____

Os Kits de Micro Chegaram!

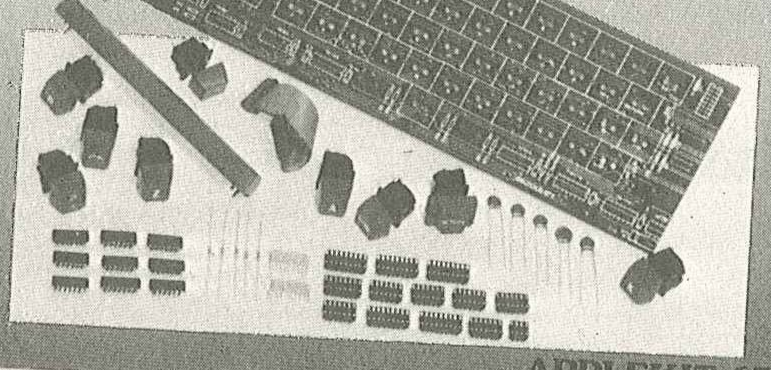
APPLEKIT - Kit de microcomputador tipo Apple®

Componentes para montagem de um microcomputador APPLEKIT completo.

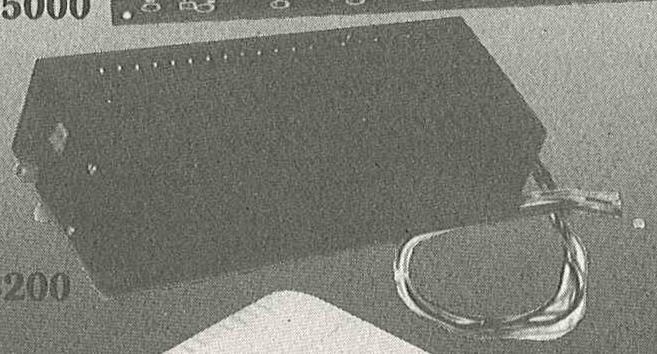
microcontrol
Sistemas de Controles
Tels.: (011) 814-0446 e 814-1110
São Paulo - Brasil.



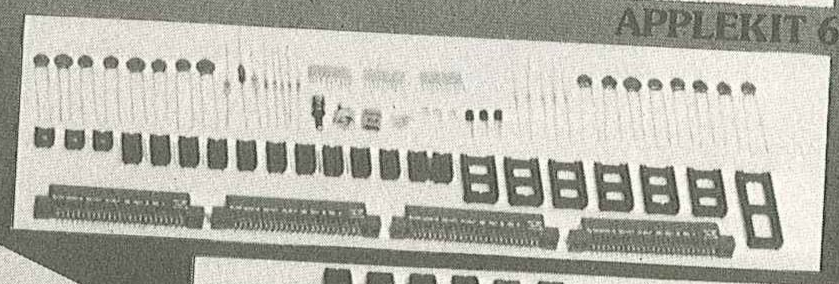
APPLEKIT 65000



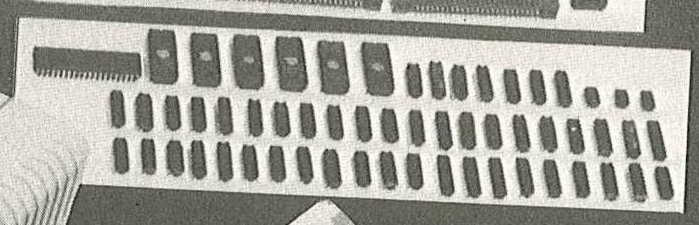
APPLEKIT 65100



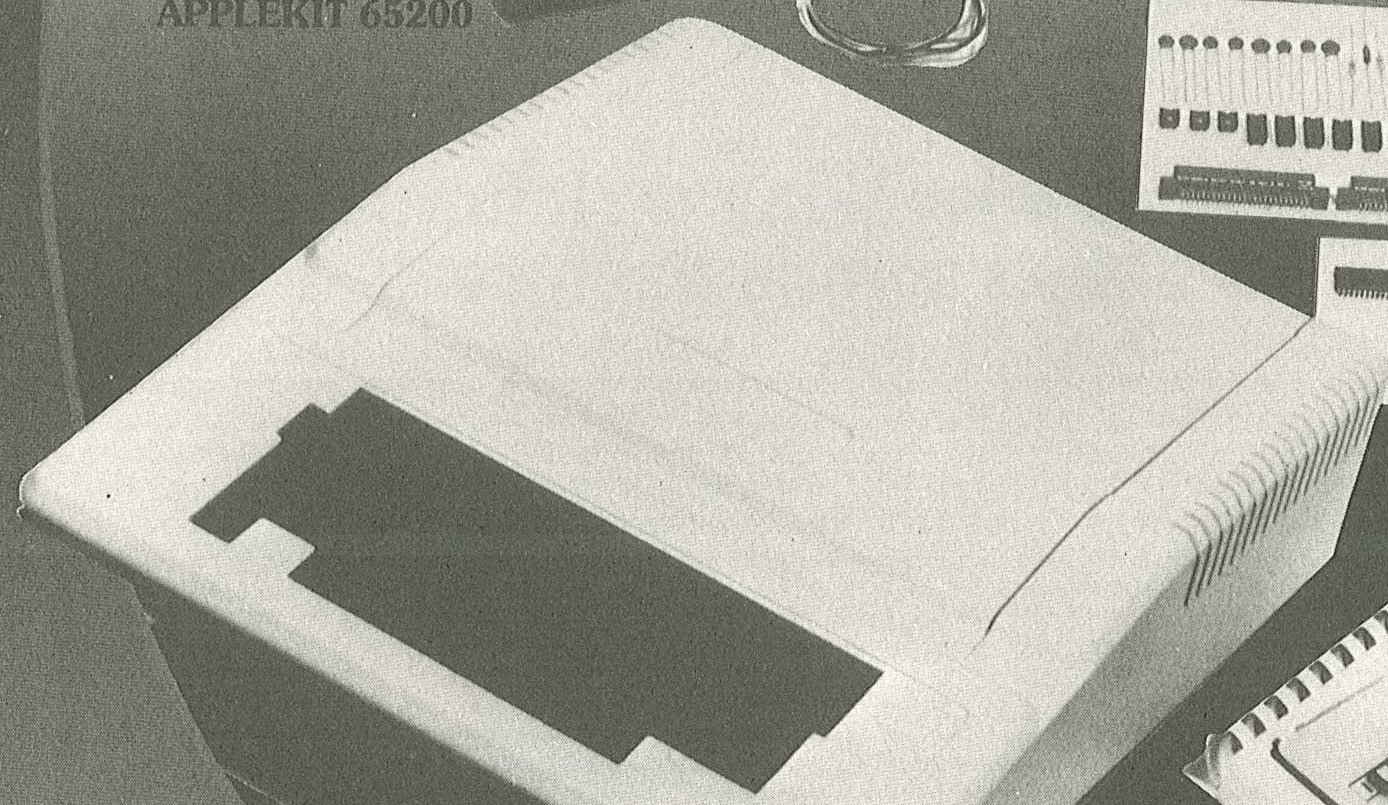
APPLEKIT 65200



APPLEKIT 65010



APPLEKIT 65020



APPLEKIT 65300



APPLEKIT 65400

APPLEKIT 65000 Placa de circuito impresso. **APPLEKIT 65010** Conjunto de soquetes, conectores, resistores e capacitores. **APPLEKIT 65020** Conjunto de semicondutores, TTL's, LSI e memórias (As memórias EPROM são fornecidas com gravação). **APPLEKIT 65100** Conjunto de teclado alfanumérico com 52 teclas e componentes, circuito impresso. **APPLEKIT 65200** Fonte de alimentação tipo chaveado. **APPLEKIT 65300** Caixa de microcomputador em poliuretano. **APPLEKIT 65400** Manual de montagem e teste de micro.

APPLEKIT é 100% compatível com os cartões periféricos da MICROCRAFT.

® Apple é marca registrada de Apple Inc.

COLEÇÃO

REVISTA interface



CAPA RÍGIDA PARA ENCADERNAÇÃO DE SUA COLEÇÃO

* *2 volumes — encaderna 6 edições por volume*
* *Proteção rígida para todos os seus exemplares*
* *Prática para bibliotecas, laboratórios, cursos, hobbistas, estudantes e professores.*
* *Uma proteção indispensável para sua coleção*
* *Para reservas preencha o cupom abaixo*

ESTOU INTERESSADO EM ADQUIRIR:

☐ Volume 1 ou volume 2 (Cr$ 2.500,00) ☐ Volume 1 + volume 2 (Cr$ 5.000,00)

GOSTARIA TAMBÉM DE ADQUIRIR OS SEGUINTES EXEMPLARES ATRASADOS PARÁ COMPLETAR MINHA COLEÇÃO:

Preço unitário: Cr$ 1.300,00

1 2 3 4 5 6 7 8 9 10 11 12

PARA ISTO ESTOU ENVIANDO CHEQUE NOMINAL À PRODIGT - PROCESSAMENTO, TECNOLOGIA E COMUNICAÇÃO LTDA, ESTRADA DO TINDIBA, 2380 - CEP 22700 - RJ.

NOME: . . . . . .
ENDEREÇO: . . . . . .
EMPRESA: . . . . . .
CARGO: . . . . . TEL: . . . . . .

PROGRAMAS DE LAZER

PROGRAMA LABIRINTO

Estre programa foi eleborado para o TK 82-C, CP 200, TK 85 e similares

O leitor Francisco J.D. Figueiredo apresenta duas colaborações: um Programa Labirinto e um Programa Intrusos.

```
1 REM LABIRINTO/F.
  FIGUEIREDO
3 LET LET U = Ø
5 GOSUB 96ØØ
10 FAST
15 FOR B = 2 TO  2/STEP 2
20 FOR A = Ø TO 19
30 PRINT AT A,/B;"■"
40 NEXT A
45 PRINT AT RND * 14 + 3,
  B;" "
50 NEXT B
60 FOR A = Ø TO 3Ø
70 PRINT AT Ø, A;"▦"; AT
  19, A;"▦"
80 NEXT A
90 FOR B = 1 TO 18
100 PRINT AT B, Ø;"▦"; AT
  B, 30;"▦"
110 NEXT B
120 GOSUB 9ØØØ
125 GOTO 167
150 PRINT AT A, B;"[$]"
155 LET Z = Z – 673
160 PRINT AT 2Ø, Ø;
  "PLACAR:"; Z;"     "
165 RETURN
167 LET Z$ = INKEY$
170 IF Z$ = "" THEN LET
  Z$ = A$
180 LET Z = Z – 5Ø
200 LET Y = A
210 LET X = B
217 LET R = RND * RND *
  RND
220 LET A = A + (Z$ = "Z")
  – (Z$ = "Q")
230 LET B = B + (Z$ = "L")
232 PRINT AT Y, X;" "
233 IF PEEK (PEEK 16396 +
  256 * PEEK 16397 + 33 *
  A + B + 1) = 128 THEN
  GOSUB 150
235 PRINT AT A, B; "$"
255 IF A > 18 OR A < 2 OR
  B < 1 THEN LET Z = INT
  (Z/3)
260 IF A > 18 OR A < 2 OR
  B < 1 OR B > 29 THEN
  GOTO 51Ø
490 LET A$ = Z$
500 GOTO 167
510 PRINT AT 2Ø,Ø; "FIM DE
  JOGO PLACAR: "; Z
520 IF Z > U THEN LET
  U = Z
530 FOR G = 1 TO 6
540 PRINT AT 21,3;
  "MELHOR PLACAR: "; U
545 PRINT AT A, B;"[$]"; AT
  A, B;"$"; AT A, B;"[$]"
550 PRINT AT 21,3;
  "[MELHOR]"
560 NEXT G
570 CLS
580 GOTO 1Ø
9000 LET A = 1Ø
9010 LET B = 1
9020 LET Z = 2ØØØØ
9030 LET Y = A
9040 LET X = B
9050 LET A$ = "Z"
9490 SLOW
9500 RETURN
```

```
9600 C L S
9605 PRINT TAB 1Ø;
     "LABIRINTO"
9610 PRINT ,, "OBJETIVO DO
     JOGO: ", "VOCÊ DEVE
     CHEGAR NO LADO
     DIREITO DO
     LABIRINTO EVITANDO
          PASSAR POR ■,
     QUE DIMINUI SEUS
     PONTOS."
9620 PRINT,, "CONTROLES:",,
     "Q → PARA CIMA",, "Z
     → PARA BAIXO",, "L
     → PARA DIREITA"
9630 PRINT AT 21,Ø;
     "PRESSIONE NEW LINE
     P/JOGAR"
9640 INPUT U$
9650 CLS
9660 RETURN
```

Descrição de linhas com caracteres gráficos

```
30      SPACE invertido
70 e 90 GRAPHICS SHIFT A
150     $ invertido
550     MELHOR invertido
9605    LABIRINTO invertido
9610    SPACE invertido
```

PROGRAMA INTRUSOS

Este programa foi elaborado para rodar nos microcomputadores compatíveis com o Sinclair / Timex.

```
 1 REM INTRUSOS
 5 REM POR FRANCISCO
   FIGUEIREDO
10 PRINT TAB 10;
   "  INTRUSOS  "
13 PRINT,, "INTRUSOS
    =   Y   ",, "VOCE = A"
20 PRINT,, "CONTROLES: "
25 PRINT "5 → MOVE
   PARA A ESQUERDA"
30 PRINT "8 → MOVE
   PARA A DIREITA"
35 PRINT "B → DISPARA"
40 PRINT,,,, "OBS.: O
   PLACAR FINAL E""
   A – – – – –
   – – – – – – – –
   DIFERENÇA ENTRE OS
   SEUS
   – – – – – – – – PONTOS
   E OS DO
   COMPUTADOR".
45 PRINT AT 21,Ø;
   "PRESSIONE NEW LINE
   P/JOGAR"
46 INPUT U$
50 LET U = Ø
100 GOSUB 9ØØØ
400 FOR N = 1 TO 7Ø
410 IF N = 1 THEN GOTO
    1Ø65
500 LET M = Ø
700 LET Z$ = INKEY$
800 LET B = B + (INKEY$ =
    "8" AND B < 29) –
    (INKEY$ = "5" AND
    B > 1)
950 LET M = (RND >.4)
1000 IF Z$ < > "B" THEN
     GOTO 1Ø52
1010 FOR A = 18 TO 2 STEP-2
1020 IF NOT M THEN GOTO
     1053
1022 LET M = Ø
1025 LET Q = B
1027 IF A$ (1, Q) = "  " THEN
     GOTO 1053
1030 FOR E = 3  TO19  STEP  4
1040 PRINT AT E,Q; "▭■▭";
     AT E, Q ;  "_ _ _"
1045 LET B = B + ( INKEY$
     = "8" AND B < 29) –
     (INKEY$ = "5" AND
     B > 1)
1047 PRINT AT 2Ø, B-1;"
     "_A_"
1050 NEXT E
1051 IF B = Q THEN GOSUB
     5000
1052 IF M = 1 THEN GOTO
     1Ø22
1053 LET A$ (1) = A$ (1, 2
     TO) + A$ (1, 1)
1054 LET A$ (2) = A$ (2,2
     TO) + A$ (2,1)
1055 LET A$ (3) = A$ (3, 2
     TO) + A$ (3, 1)
1065 PRINT AT 2, Ø;  A$ (1);
     AT 4, Ø;  A$ (2);  AT
     6, Ø;A$ (3);  AT 2Ø, B;
     "_A_"
1067 IF Z$ < > "B" THEN
     GOTO 25ØØ
1068 LET V = (PEEK (A * 33 +
     B + D) = 19Ø)
1072 IF V THEN GOTO 1076
1073 PRINT AT A, B + 1;  "* ";
     AT A, B + 1;  "_ "
1075 NEXT A
1076 LET S = S + (V < > Ø)
```


ADVANCING
CONSULTORIA DE PESSOAL E TREINAMENTO

ADVANCING
COMPUTER SHOP

ADVANCING
SOFTWARE HOUSE

EMPRESAS DO GRUPO ADVANCING:
CONSULTORIA - TREINAMENTO - DIREÇÃO Andradas 1560 - Cj. 518 - 5º and.
COMPUTER SHOP - SOFTWARE HOUSE Sarmento Leite, 248 - Fones (0512) 26-8246/26-0194/26-1194 P. Alegre - RS

```
1090 IF V THEN LET A$
     (A/2, B + 2) = " _ "
1092 IF A < 2 THEN GOTO
     25ØØ
1095 PRINT AT A, B + 1;
     " * "; AT A, B + 1; " _ ";
     AT ØØ; "VOCE"; S *
     641
2500 NEXT N
2520 PRINT AT Ø,16; "EU";
     C * 439
3000 PRINT AT 1Ø,1Ø; "FIM
     DO JOGO"
3005 LET P = S * 641 – C *
     439
3010 IF P > U THEN LET
     U = P
3020 PRINT "SEU MAiOR
     PLACAR – "; U
3080 FOR N = 1 TO 60
3090 NEXT N
3095 PRINT AT 21,Ø ;
     "OUTRA PARTIDA?
     (SIN)"
3096 INPUT I$
3097 IF I$ < > "S" THEN
     STOP
3098 GOSUB 9ØØØ
3990 CLS
4000 GOTO 4ØØ
5000 LET C = C + 1
5005 FOR J = 1 TO 3
5010 PRINT AT 1Ø,1Ø;
     " BUUM " ; AT 1Ø,1Ø;
     "BUUM";AT 1Ø,1Ø;
     "_____"
5020 PRINT AT 2Ø, Q + 1;
     " ■ "; AT 2Ø, Q + 1;
     " * ";AT 2Ø, Q + 1; "X";
     AT 2Ø, Q + 1; "M"; AT
     2Ø, Q + 1; "A"
5050 NEXT J
5060 PRINT AT Ø, Ø;
     " VOCE _"; S *641,
     " EU –"; C * 439
5100 RETURN
9000 FAST
9010 LET Z$ = " "
9020 LET S = Ø
9070 LET C = Ø
9090 LET B = 15
9100 DIM A$ (3,32)
9110 FOR N = 1 TO 3
9120 FOR M = 1 TO 32
9130 LET A$ (N, M) = CHR$
     ((RND > . 6) *  190)
9140 NEXT M
9150 NEXT N
9160 LET D = PEEK 16396
     + 256 *  PEEK 16397 + 2
9170 CLS
9200 SLOW
9900 RETURN
```

Descrição de linhas com caracteres gráficos

```
1Ø  INTRUSOS em fundo
    invertido
 13 Y e A invertidos
1Ø4Ø SHIFT 7, SPACE,
    SHIFT 7 (gráficos)
1Ø47 A invertido
1Ø65 A invertido
5Ø1Ø BUUM invertidos
5Ø2Ø SPACE, SHIFT B, A
    invertidos
5Ø6Ø VOCÊ E EU invertidos
```


LOJA MICRO-KIT
TUDO SOBRE MICROCOMPUTADOR

FAÇA AGORA A SUA ENCOMENDA

LANÇAMENTO DO LIVRO "CURSO DE BASIC" – VOL. I – EDIÇÃO PRÓPRIA

CURSOS

Basic p/adultos e crianças, com método próprio comprovadamente eficiente; Professores c/mestrado em ENGENHARIA DE SISTEMAS, mais de 20 cursos aplicados. Turmas pequenas, aulas práticas com MICROCOMPUTADOR.

VENDA DE MICROCOMPUTADOR

Unitron AP II, Digitus, TK e CP 200.
Financiamento em até 24 meses.

PROGRAMAS

Comerciais e Jogos p/APPLE, Unitron, Polymax, Digitus TK e CP 200.

SUPRIMENTOS

Disquetes, Caixa p/Disquetes, Formulários Contínuos etc.

VENDA DE LIVROS E REVISTAS

Despachamos para todo o Brasil.

Rua Visconde de Pirajá, 303 S/Loja 210
Tels. (021) 267-8291 521-4638
CEP 22410 – Rio de Janeiro
Rua Visconde de Pirajá, 365 – Sobreloja 209
Ipanema

BITS & BYTES
COMPUTADORES

- VENDAS
- PROGRAMAS
- DISKETTS
- FITAS
- SERVIÇOS
- CURSOS DE BASIC
- FORMULÁRIOS
- ASS. TÉCNICA (AUTORIZADA PROLÓGICA)

CONSERTOS EM 24 HORAS (COM GARANTIA) PARA O CP-500 e DGT-100

EM SÃO CONRADO
Estrada da Gávea, 642 – Lj. B
TEL.: 322-1960

FILIA
A SUPER LINGUAGEM

Agora disponível para os micros de lógica Sinclair, Trs-80, CP-500, DGT-100 e outros

Transforme seu micro pessoal numa poderosa máquina

MAIORES INFORMAÇÕES:
Prof. Rildo Pragana
Rua Condessa do Livramento, 54A
Derby – Recife – PE

GLOSSÁRIO

CARRIER SYSTEM
(Sistema de portadoras)
Um sistema que permite que vários sinais de comunicação independentes entre si compartilhem o mesmo circuito.

CARTRIDGE
(Cartucho)
Cartão de circuito impresso inserido numa caixa plástica com um conector, no qual estão uma ou mais memórias EPROM's contendo programas e informações. Muito utilizado para vídeo jogos. Os discos magnéticos rígidos também são disponíveis em *cartridge* (vide *discos magnéticos*).

CARTRIDGE WRITE LOCKOUT
Habilitação e proteção de gravação em disco magnético rígido de cartucho.

CASCADE CONTROL
(Controle em cascata)
Sistema automático em que as unidades de controle organizam-se em seqüência, de maneira que cada uma controle sua sucessora, sendo, por sua vez, controlada por sua antecessora.

CASSETE
(vide áudio cassete)

CAT
(Computer Assisted Teaching)
Treinamento auxiliado por computador.

CATALOG (UE)
(Catálogo)
Lista de itens arranjados para fácil referência.

CATASTROPHIC ERROR
(Erro catastrófico)
Quando muitos erros ocorrem em um programa de computador em que o *job* é terminado (vide *job*).

CATHODE RAY TUBE
(CRT)
Tubo de raios catódicos, similar aos utilizados em televisão, usado para exibir informações incluindo gráficos.

CB
(Citizens'band radio)
Radiotransmissão na faixa do cidadão.

CBASIC
(Basic comercial)
O *CBASIC* é uma linguagem compilada que tem mais flexibilidade do que o BASIC.

CCD
(Charge Coupled Device)
Dispositivo de armazenamento para informação, que usa somente capacitores especiais.

CCITT
(Comité Consultatif International Télégraphique et Téléphonique)
Comitê de Consultoria Internacional de Telegrafia e Telefonia.

CCTV
Circuito fechado de televisão.

CELL
(Célula)
Menor unidade de armazenamento, com capacidade de armazenar somente um bit.

CENTRAL PROCESSING UNIT
(Unidade Central de Processamento)
O coração de um computador, que contém a *Unidade Lógica Aritmética* (ULA), a *memória* e a *unidade de controle*, as quais dirigem e coordenam a operação do computador e seus periféricos.

CHAIN INDEXING
(Cadeia indexada)
Índice alfabético no qual cada item aparece abaixo do título de cada termo escolhido, dentro de uma classificação hierárquica. Os termos de menor ordem na classificação são listados depois dos termos de maior ordem.

CHAIN SEARCH
(Cadeia de busca)
Método de busca que conduz de uma gravação a outra, até a gravação desejada ser encontrada; caso contrário, o fim da cadeia é prolongado.

CHANNEL
(Canal)
Caminho no qual os sinais podem ser transmitidos.

CHARACTER
(Caractere)
Um único número de 0 (zero) a 9 (nove); uma letra; um sinal de pontuação ou qualquer outro símbolo gráfico.

CHARACTER CODE
Código usado para representar um caractere em um computador.

CHARACTER DENSITY
(Densidade de caractere)
O número de caracteres armazenados em uma unidade de comprimento, área ou volume.

CHARACTER GENERATOR
(Gerador de caracteres)
Dispositivo eletrônico (circuito) que gera os caracteres a serem exibidos em um CRT ou vídeo.

CHARACTER PRINTER
(Impressora de caracteres)
Impressora que transfere totalmente letras, números ou símbolos gráficos a cada movimento de impressão, imprimindo caracteres mais legíveis do que os criados pelas impressoras de matrix de pontos.

CHARACTER SET
(Conjunto de caracteres)
Conjunto de números, letras, gráficos e símbolos que possam ser gerados por um sistema particular.

CHARACTER SKEW
Alinhamento de caracteres.

CHARACTER STRING
Uma seqüência de caracteres.

CHECK BIT
Dígito binário de teste.

CHECK CHARACTER
Caractere usado em transferências de informações para testar erros, como um dígito de teste.

CHECK DIGIT
(Dígito de teste)
Dígito de teste utilizado para detecção automática de erros em transferências de informações.

CHECK EVEN/ODD PARITY
Teste de paridade par ou ímpar.

CHECKOUT TEST ROUTINE
Rotina de teste verificadora.

CHECK POINT
(Ponte de teste)
Ponto de controle em uma rotina, onde realiza-se uma verificação ou comparação.

CHIP
(Circuito integrado)
Pequena pastilha medindo entre 1 a 5 cm de comprimento, podendo ter entre 14 a 40 pinos de conexões externas, onde integram-se, condensados, um conjunto de circuitos e componentes eletrônicos.

CICS

(Customer Information Control System)
Sistema de controle de informação de clientes. Um sistema de software usado pelos bancos americanos para possibilitar o acesso das agências locais ao computador central (contas de clientes).

CIM

(Computer Input Microfilm)
Microfilme usado para dar entrada de informações em alta velocidade num computador.

CIRCUIT BOARD

(Placa de circuito)
Uma fina placa retangular de fibra de vidro, na qual são montados um ou mais chips. Existem muitos tipos de placas de circuitos, também conhecidas como *PC boards* e *Cards*, incluindo placas de memórias, placas de processadores, placas de I/O (input/output) etc. . .

C-LANGUAGE

(Linguagem C)
Linguagem para microcomputador, desenvolvida pelos laboratórios Bell, dos Estados Unidos.

CLEAR MEMORY WORD

Palavra de memória composta de um zero.

CLOCK

Gerador de pulsos elétricos, que sincroniza todos os sinais em um computador.

CLUSTER

Unidades múltiplas.

CMC

Código para caracteres magnéticos.

CMOS

(Complementary Metal Oxide Semiconductor)
Transistor usado em circuitos integrados; é utilizado onde o baixo consumo de potência é exigido.

COAXIAL CABLE

(Cabo coaxial)
Cabo de comunicação consistindo de um condutor central interno, normalmente de cobre, isolado de um outro condutor (malha de cobre). Estes cabos são utilizados para altas freqüências e apresentam muito baixa perda de energia.

COBOL

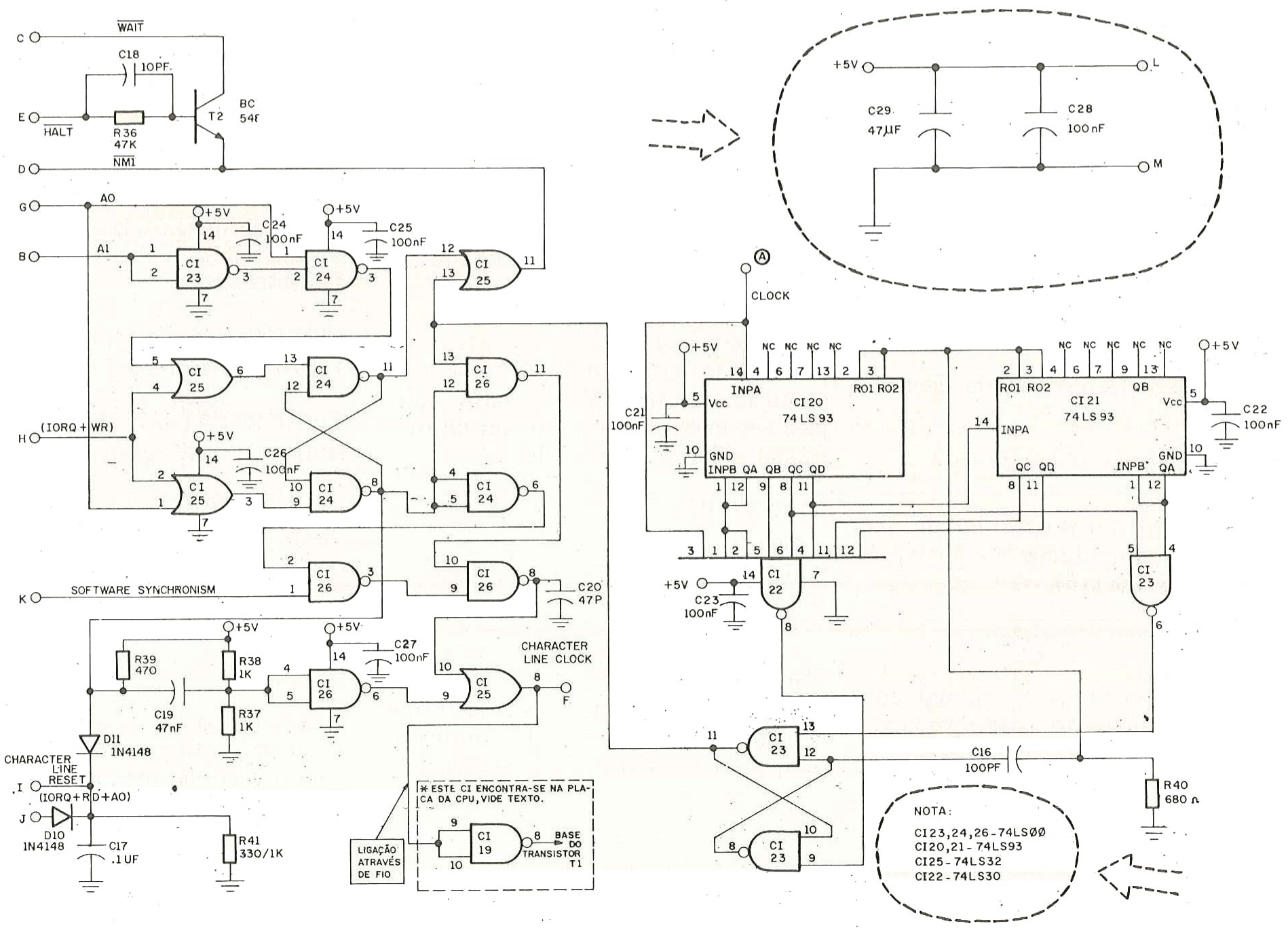
(Commom Business Orientated Language)
Linguagem de programação de alto nível, desenvolvida especialmente para manipulação de dados comerciais.

ERRATA

ERRATA DO PROGRAMA CONTAS A PAGAR

Falta Linha 415 GOSUB 1500

Linha 895 – alterar para PRINT A (Y, 5)
Linha 1130 – alterar para PRINT "ANO";
Linha 1166 – LET T$ = "" (não há espaço entre as aspas)
Linha 1167, 1169 e 1172 – não foi impresso o sinal de < antes do 10
Linha 1190 LET M = M + INT (365.25*(AX-1)) + 1
Linha 1300 – após E digite Shift Q (aspas dupla)
Linha 1315 – PRINT AT 21,0; "(digite 22 SPACE)"
Linha 1625 – LET T$ = STR$ A (Y, 1)



Você tem a informação. Nós temos os meios para fazê-la andar

A revista Interface chega ao seu 15.º número.
Muita coisa mudou, no mundo, no país, na informática.
Queremos que você nos diga se falta alguma coisa para a revista Interface acompanhar os progressos que você tem feito na informática e na sua concepção de mundo. Precisamos saber o que está mudando em nossa cultura. Por isso pedimos a você que dê uma nota (até 10) para cada um dos quesitos propostos.
Depois escolha sua maior crítica, sua sugestão mais elaborada, e a resuma no quadrado que deixamos em branco. Pedimos que se identifique.
Nada é menos humano e menos inteligente do que o anonimato. E Interface não é uma revista da média de opiniões. Antes pretende insuflar cada pesquisador individual que exista entre seus leitores.

DICAS DE SOFTWARE ____
DICAS DE HARDWARE ____
PROGRAMAS DE LAZER ____
PROGRAMAS APLICATIVOS ____
CONTROLADORES / INTERFACE ____
GLOSSÁRIOS DE TERMOS TÉCNICOS ____
MERCADO DE COMPUTADORES ____
CONSULTORIA ____
CURSO Z-80 ____
MUMPS ____
FILIA ____
PERIFÉRICOS ____
CURSO BASIC ____
CP / M ____
HARDHOUSE ____
CHIPS ____

DETALHE AQUI SUAS CRÍTICAS E/OU SUGESTÕES:

Nome:

End:

CEP.

RECORTE E ENVIE PARA A REVISTA INTERFACE



O micro NAJA foi desenvolvido utilizando os mais modernos padrões de arquitetura de Microcomputador, atingindo uma ampla faixa, desde os computadores pessoais até os utilizados em empresas de pequeno e médio porte. Uma de suas grandes vantagens é a sua versatilidade, ou seja, você poderá adquiri-lo na sua versão mais simples, podendo você mesmo expandi-lo à medida de suas necessidades, a um baixo custo

CARACTERÍSTICAS TÉCNICAS

- *48K bytes de memória RAM*
- *16K bytes de memória ROM*
- *Clock de 3,6 MHz ou 2,1 MHz comutado por Soft*
- *Saída para impressora paralela*
- *6 conectores para expansão no próprio gabinete*
- *Microprocessador Z-80A*
- *Vídeo de 16 linhas por 64 ou 32 colunas*
- *Interface de cassete para 1.500 ou 500 BPS*
- *Linguagem Basic na ROM do sistema*
- *Software compatível com TRS-80 mod. III*

ACESSÓRIOS

- *Monitor de vídeo de 12" verde profissional*
- *Interface para 4 unidades de disco de 51/4" de dupla dens.*
- *Unidade de disco face simples ou dupla*
- *Interface para 4 MHz de Clock*

AGORA TAMBEM COM:

- CP/M
- CLOCK DE 6 MHZ
- UNIDADE DE DISCO DE 8"
- VIDEO A CORES
- SINTETIZADOR DE VOZ

KEMITRON LTDA.

Av. Contorno, 6048 - Savassi - Fone: 225-0644
Telex - (031) 3074 - KEMI - BR Belo Horizonte - MG

AS-1000

o micro que cresce com você.



O Microcomputador AS-1000 é uma ótima escolha para quem está iniciando na ciência da computação. Seus recursos de programação e sua concepção modular, porém, permitem que ele o acompanhe até as aplicações mais sofisticadas.

O AS-1000 já nasce com uma biblioteca de milhares de programas para jogos, administração doméstica, aplicações comerciais e profissionais.

O AS-1000 é fabricado com a qualidade ENGEBRÁS e garantido por um ano.

Entre na era da informática com a escolha certa. AS-1000, o seu micro pessoal.

Escreva-nos, sua correspondência não ficará sem resposta.

CARACTERÍSTICAS TÉCNICAS:

- 16 K bytes de memória iniciais
- Expansão interna para 32 e 48 K bytes
- 8 K bytes de memória EPROM
- Microprocessador Z-80A
- Teclado de membrana com ação mecânica positiva
- 40 teclas e 154 funções
- Basic e linguagem de máquina
- Video normal ou reverso
- Saída para qualquer impressora
- Manipula até quatro cassetes com geração de arquivo
- Modem
- Joystick
- Speed File
- Fonte de alimentação embutida (110/220 volts)
- Nível de leitura de gravação automático

EB ENGEBRÁS
ELETRÔNICA E INFORMÁTICA LTDA.
Rua do Russel, 450 - 3.º andar
cep 22210 Rio de Janeiro - RJ
Tel.: (021) 205-4898

NOVO CP 300 PROLÓGICA.

O pequeno grande micro.





Compatível com programas em fita cassete ou em disco.

Agora, na hora de escolher entre um microcomputador pessoal simples, de fácil manejo e um sofisticado microcomputador profissional, você pode ficar com os dois.

Porque chegou o novo CP 300 Prológica.

O novo CP 300 tem preço de microcomputador pequeno. Mas memória de microcomputador grande.

Ele já nasceu com 64 kbytes de memória interna com possibilidade de expansão de memória externa para até quase 1 megabyte.



Pode ser acoplado a uma impressora.

E tem um teclado profissional, que dá ao CP 300 uma versatilidade incrível. Ele pode ser utilizado com programas de fita cassete, da mesma maneira que com programas em disco.



O único na sua faixa que já nasce com 64 kbytes de memória.



Permite conexão telefônica.

Pode ser ligado ao seu aparelho de TV, da mesma forma que no terminal de vídeo de uma grande empresa.

Com o CP 300 você pode fazer conexões telefônicas para coleta de dados, se utilizar de uma impressora e ainda dispor de todos os programas existentes para o CP 500 ou o TRS-80 americano. E o que é melhor: você estará apto a operar qualquer outro sistema de microcomputador.



Pode ser ligado a um televisor comum ou a um sofisticado terminal de vídeo.

Nenhum outro microcomputador pessoal na sua faixa tem tantas possibilidades de expansão ou desempenho igual.

CP 300 Prológica.

Os outros não fazem o que ele faz, pelo preço que ele cobra.

PROLOGICA
microcomputadores

Av. Eng.º Luis Carlos Berrini, 1168 - SP

RESERVA DE MERCADO

DEFESA DOS VALORES NACIONAIS

AM
Manaus - 234-1045
• BA-Salvador - 247-8951
• CE-Fortaleza - 226-0871 - 244-2448
• DF-Brasília - 226-1523 - 225-4534 • ES-Vila Velha 229-1387 - Vitória - 222-5811 • GO-Goiânia - 224-7098 • MT Cuiabá - 321-2307 • MS-Campo Grande - 383-1270 - Dourados - 421-1052
• MG-Belo Horizonte - 227-0881 - Betim - 531-3806 - Cel. Fabriciano - 841-3400 - Juiz de Fora - 212-9075 - Uberlândia - 235-1099 • PA-Belém - 228-0011 • PR-Cascavel - 23-1538 - Curitiba - 224-5616 - 224-3422 - Foz do Iguaçu - 73-3734 - Londrina - 23-0065 • PE-Recife - 221-0142 • PI-Teresina 222-0186 • RJ-Campos - 22-3714 - Rio de Janeiro - 264-5797 - 253-3395 - 252-2050 • RN-Natal - 222-3212 • RS-Caxias do Sul - 221-3516 - Pelotas - 22-9918 - Porto Alegre - 22-4800 - 24-0311 - Santa Rosa - 512-1399 • RO-Porto Velho - 221-2656 • SP Barretos - 22-6411 - Campinas - 2-4483 - Jundiaí - 434-0222 - Marília - 33-5099 - Mogi das Cruzes - 469-6640 - Piracicaba - 33-1470 - Ribeirão Preto - 625-5926 - 635-1195 - São Joaquim da Barra - 728-2472 - São José dos Campos - 22-7311 - 22-4740 - São José do Rio Preto - 32-2842 - Santos - 33-2230 Sorocaba - 33-7794 • SC-Blumenau - 22-6277 - Chapecó - 22-0001 - Criciúma - 33-2604 - Florianópolis - 22-9622 - Joinvile - 33-7520 • SE-Aracajú - 224-1310.

Solicite demonstração nos principais magazines.