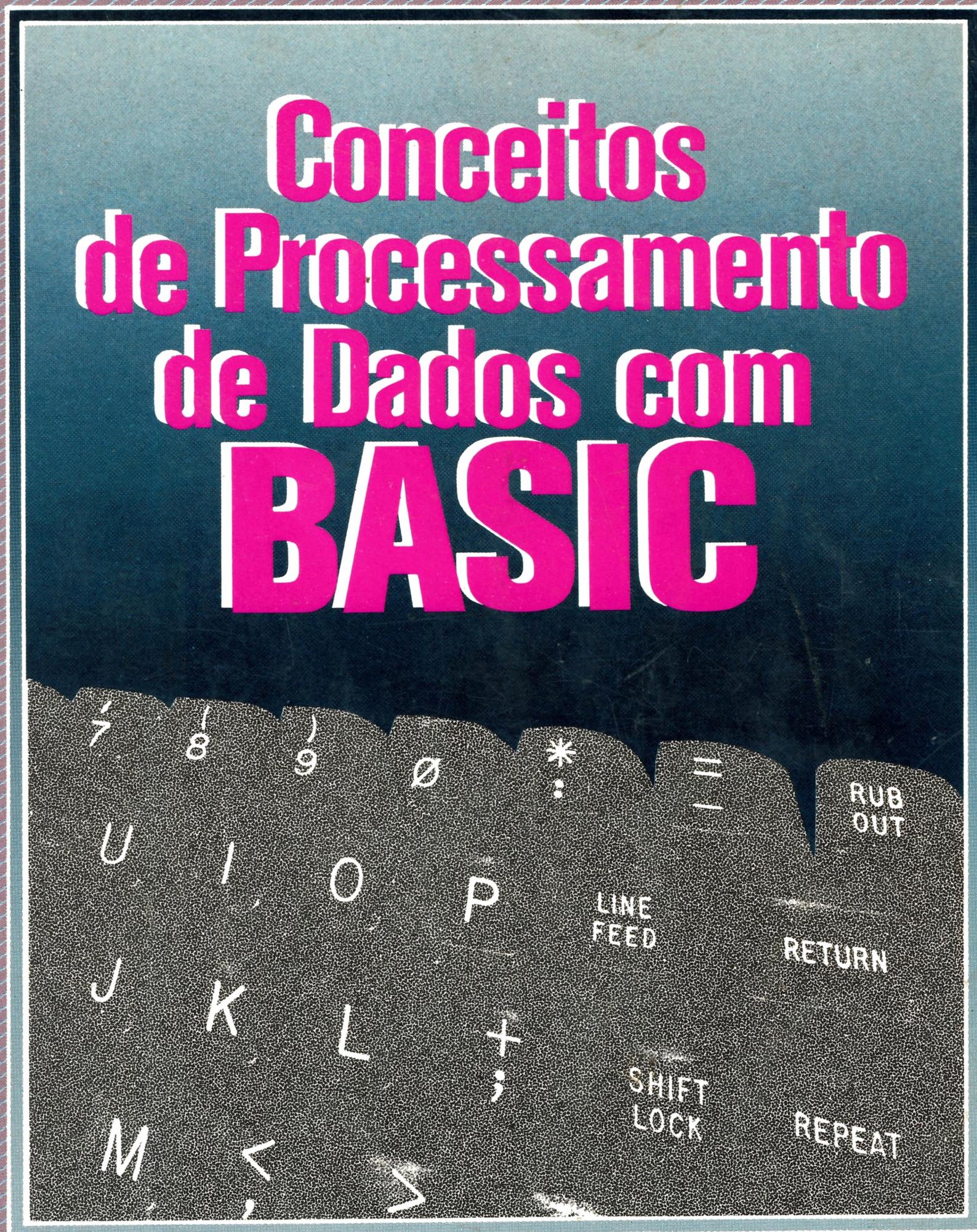


Robert A. STERN
Nancy

Conceitos de Processamento de Dados com **BASIC**



EDITORA CAMPUS

Títulos de Computação da Editora Campus

ADMINISTRAÇÃO DE SISTEMAS DE INFORMAÇÃO – *P. Ein-Dor & E. Segev*

ANÁLISE DE SISTEMAS PARA SISTEMAS DE INFORMAÇÃO POR COMPUTADOR – *J. C. Wetherbe*

APLICAÇÕES DE MICROPROCESSADORES – *J. A. Kuecken*

APLICAÇÕES DO COMPUTADOR NA MEDICINA – *N. F. Kember*

APLICAC; UM SOFTWARE EDUCACIONAL, PESSOAL E PROFISSIONAL EM BASIC – *E. A. Meili*

BASIC BÁSICO, – 4ª ed. *J. C. Pereira Filho*

BASIC PARA APLICAÇÕES COMERCIAIS – *D. Hergert*

BASIC PARA MICROS PESSOAIS – 2ª ed. *J. C. Pereira Filho*

BASIC SINCLAIR – *R. U. Christmann*

BRINCANDO COM O COMPUTADOR – *J. A. Moreira*

COBOL PARA ESTUDANTES – 4ª ed. *A. Parkin*

COMO LIDAR COM O COMPUTADOR – *H. C. Lucas Jr.*

COMPUTADORES BRASILEIROS; INDÚSTRIA, TECNOLOGIA E DEPENDÊNCIA – *P. B. Tigre*

COMPUTADORES PARA USUÁRIOS – *J. C. Pereira Filho, Coordenador*

Vol. I – *Aplicação de Computadores*

Vol. II – *Equipamentos e Sistemas de Computação*

Vol. III – *Programas e Programação de Computadores*

Vol. IV – *Seleção de Sistemas de Computação*

CONCEITOS DE PROCESSAMENTO DE DADOS COM BASIC – *R. A. Stern & N. Stern*

A CONSTRUÇÃO DE UM COMPILADOR – *V. W. Setzer & I. S. H. Melo*

CRIANÇA TAMBÉM FAZ PROGRAMAS, 2ª ed. – *J. A. Moreira*

DOCUMENTAÇÃO DE SOFTWARE – *J. D. Lomax*

ESPECIFICAÇÃO DE SISTEMAS – *S. J. Waters*

ESTRUTURAS DE DADOS, 2ª ed. – *P. S. Veloso et al.*

FUNDAMENTOS DE ESTRUTURAS DE DADOS – *E. Horowitz & S. Sahni*

FUNDAMENTOS DE PROCESSAMENTO DE DADOS – *W. T. Price*

GERÊNCIA DE BASES DE DADOS PARA MICROCOMPUTADORES – *E. G. Brooner*

GRAFOS E ALGORITMOS COMPUTACIONAIS – *J. L. Szwarcfiter*

GUIA DE LINGUAGENS DE COMPUTADORES – *H. L. Helms Jr.*

GUIA PARA PROGRAMADORES, 2ª ed. – M. Bohl

IMPLANTAÇÃO DE MICROS E MINICOMPUTADORES COMERCIAIS – P. A. Knight

INTRODUÇÃO À ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES – H. Lorin

INTRODUÇÃO À CIÊNCIA DA COMPUTAÇÃO COM WATFIV E FORTRAN, 2ª ed. – S. E. R. Carvalho

INTRODUÇÃO À PROGRAMAÇÃO COM PASCAL, 2ª ed. – S. E. R. Carvalho

INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES, 3ª ed. – H. V. R. Corrêa Silva et al.

INTRODUÇÃO À PROGRAMAÇÃO FORTRAN – J. C. Pereira Filho

INTRODUÇÃO À SEGURANÇA DO COMPUTADOR – M. B. Wood

INTRODUÇÃO A SISTEMAS DE BANCOS DE DADOS – C. J. Date

INTRODUÇÃO AO PROCESSAMENTO DE TEXTOS – G. L. Simons

INTRODUÇÃO AO VISICALC, 2ª ed. – E. A. Garbin

JCL/SISTEMA 370, 3ª ed. – G. D. Brown

JOGOS EM BASIC PARA MICROS – A. J. L. Botelho

LCP – LÓGICA DE CONSTRUÇÃO DE PROGRAMAS, 2ª ed. – J. D. Warnier

LCS – LÓGICA DE CONSTRUÇÃO DE SISTEMAS – J. D. Warnier

LINGUAGEM DE PROGRAMAÇÃO ALGOL, 2ª ed. – L. M. Segre

LINGUAGEM PASCAL – V. L. Strube de Lima

MANUAL DE CP/M INCLUINDO MP/M – R. Zaks

MANUAL DE COBOL ESTRUTURADO – D. D. McCracken

MICRO-MINICOMPUTADORES BRASILEIROS – E. L. Passos

MICROCOMPUTADORES PARA APLICAÇÕES COMERCIAIS – W. Barden Jr.

- MICROPROCESSADORES DE 16 BITS – C. A. Titus et al.

MICROPROCESSADORES/MICROCOMPUTADORES – A. J. Khambata

Vol. I – Arquitetura

Vol. II – Software e Sistemas

1001 APLICAÇÕES PARA O SEU COMPUTADOR PESSOAL – M. Sawusch

ORGANIZAÇÃO DE BANCOS DE DADOS, 4ª ed. – A. L. Furtado e C. S. Santos

PRINCÍPIOS DE SISTEMAS OPERACIONAIS, 3ª ed. – C. C. Guimarães

PROGRAMAÇÃO EM ASSEMBLER E LINGUAGEM DE MÁQUINA – D. C. Alexander

PROGRAMAÇÃO ESTRUTURADA EM COBOL – A. B. Furtado

PROGRAMAÇÃO SISTEMÁTICA EM PASCAL, 2ª ed. – N. Wirth

PROGRAMAS ADMINISTRATIVOS EM BASIC SINCLAIR – *L. Karsten*

RPG-II – *J. C. Pereira Filho*

REDES DE COMPUTADORES; ASPECTOS TÉCNICOS E OPERACIONAIS

– *D. A. Menascé e D. Schwabe*

O SEU COMPUTADOR PESSOAL – *M. Waite e M. Pardee*

SISTEMA OPERACIONAL UNIX – *K. Christian*

SISTEMAS DE VIDEOCASSETE; TEORIA E MANUTENÇÃO – *G. P. McGinty*

SISTEMAS OPERACIONAIS PARA MICROCOMPUTADORES – *M. Dahmke*

TÉCNICAS E PRÁTICA DE PROGRAMAÇÃO – *A. Chantler*

**TECNOLOGIA DA INFORMAÇÃO: UM GUIA PARA EMPRESAS, GERENTES
E ADMINISTRADORES** – *J. Eaton e J. Smithers*

30 PROGRAMAS EM BASIC PARA COMPUTADORES PESSOAIS – *D. Chance*

USANDO CP/M: UM GUIA EM ENSINO PROGRAMADO – *J. N. Fernandez
& R. Ashley*

MICROBITS

Macroinformações e Programas para Micros TK 82-83-85, CP-200 e compatíveis.
*Uma publicação bimestral com análises de hardware, aplicativos, artigos sobre
linguagem de máquina, dicas de programação e respostas para suas dúvidas.*

MICROINFORMÁTICA INTERNACIONAL

*Uma publicação bimestral sobre Segurança e Prevenção de Fraudes em
Computadores, Aplicação de Microprocessadores e Microcomputadores e
Microeletrônica.*

Assinaturas: 10 ORTN's

Procure nossas publicações nas boas livrarias ou comunique-se diretamente com:

EDITORA CAMPUS LTDA.

Livros Científicos e Técnicos

Rua Barão de Itapagipe, 55

20261 Rio de Janeiro – RJ – Brasil

Telefone: (021) 284 8443

Atendemos também pelo reembolso postal.

**Conceitos
de Processamento
de Dados com
BASIC**

Robert A. **STERN**
Nancy

Nassau Community College
Garden City, N.Y.

Hofstra University
Hempstead, N.Y.

Conceitos de Processamento de Dados com **BASIC**

Tradução

Mariza Saad Rodrigues da Cunha

EDITORA CAMPUS LTDA.

Rio de Janeiro

Do original:
Concepts of Information Processing With BASIC.

Copyright © 1983 by John Wiley & Sons, Inc. All Rights Reserved.
Autorizada tradução da edição publicada por John Wiley & Sons, Inc.

© 1985, Editora Campus Ltda.

Todos os direitos para a língua portuguesa reservados e protegidos pela Lei 5988 de 14/12/1973.

Nenhuma parte deste livro poderá ser reproduzida ou transmitida sejam quais forem os meios empregados: eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.

Todo o esforço foi feito para fornecer a mais completa e adequada informação. Contudo a editora não assume responsabilidades pelo uso da mesma.

A Editora Campus não é filiada a nenhum fabricante de sistemas computacionais.

Capa
Otavio Studart

Diagramação, composição, paginação e revisão
Editora Campus Ltda.

Rua Barão de Itapagipe 55 Rio Comprido

Tel.: (021) 284 8443

20261 Rio de Janeiro RJ Brasil

Endereço telegráfico: CAMPUSRIO

ISBN 85-7001-202-0

(Edição original: ISBN 0-471-87617-8 John Wiley & Sons, Inc., New York.)

Ficha Catalográfica
CIP-Brasil. Catalogação-na-fonte
Sindicato Nacional dos Editores de Livros, RJ.

S858c Stern, Robert A.
Conceitos de processamento de dados com BASIC / Robert A. Stern e Nancy Stern ; tradução de Mariza Saad Rodrigues da Cunha. — Rio de Janeiro : Campus, 1985.
Glossário
ISBN 85-7001-202-0

1. BASIC (Linguagem de programação para computadores) 2. Processamento eletrônico de dados I. Stern, Nancy II. Título

84-0608

CDD — 001.642

PREFÁCIO

Este livro proporciona ao leitor uma visão geral dos conceitos de processamento da informação. Pode ser usado como um guia para os usuários de computador ou usuários em potencial, como suplemento para qualquer curso que utilize computadores, ou como texto único em cursos introdutórios que enfoquem os conceitos do processamento da informação mas que não necessitem de uma abordagem mais detalhada. Serve também como uma breve exposição aos princípios básicos de computação para qualquer pessoa que esteja interessada somente em aprender alguma coisa sobre este assunto ou que esteja pretendendo adquirir um computador.

O livro fornece uma análise gradual dos principais aspectos do processamento da informação, bem como das principais questões relacionadas à computação. Apresenta uma visão realista dos computadores, esclarecendo as limitações destas máquinas, assim como as suas vantagens. É ilustrado com inúmeras fotografias, diagramas e gráficos que servem de reforço e ajuda para que o leitor possa entender melhor o material. No final de cada capítulo aparecem sumários bastante concisos para consolidar os conceitos apresentados. Para cada unidade incluem-se formulários-testes para estudo individual, com o objetivo de auxiliar os leitores na avaliação de sua compreensão. Há um glossário no final do texto.

O Capítulo 8, *Programação em BASIC*, apresenta uma visão mais profunda do assunto. Os leitores descobrirão não somente as características da linguagem BASIC, mas serão capazes de codificar programas simples e de nível intermediário depois de lerem o capítulo.

Evitamos utilizar o jargão técnico, exceto onde as definições dos termos-chave são imprescindíveis. A importância não está nos termos, mas nos conceitos e fundamentos do processamento da informação.

**ROBERT A. STERN
NANCY STERN**

NOTA DO TRADUTOR

Procurei traduzir para o português a maioria dos termos técnicos com as suas respectivas abreviaturas, baseando-me no Dicionário de Informática da SUCESU e em pesquisas feitas na área de Processamento de Dados. Porém, foi necessário abrir exceção para as declarações e comandos que geralmente não são traduzidos. Permaneceram também em inglês os termos *Hardware*, *Software*, *Firmware*, *Hard Copy* e *Soft Copy*, por não haver ainda um correspondente preciso para denominá-los na língua portuguesa.

Adotei este método de tradução com o objetivo de facilitar a compreensão dos conceitos do processamento da informação, principalmente por se tratar de um manual introdutório, de fácil acesso a todas as pessoas interessadas em adquirir noções básicas de computação.

Tentei ainda, através da tradução, incentivar o combate ao uso de termos técnicos em inglês, já que na maioria dos casos existem no português termos correspondentes para os mesmos que poderiam ser adotados pelo pessoal de Informática.

Elaborei um glossário no final do livro com a tradução dos comandos e declarações. Estes, na realidade, não são traduzidos, pois trata-se da nomenclatura específica do assunto. Porém, meu intuito foi mais uma vez o de tentar facilitar a compreensão do leitor, sem pretender, evidentemente, que tais comandos sejam traduzidos para aplicação prática.

Mariza Saad Rodrigues da Cunha

SUMÁRIO

UNIDADE 1 DESCRIÇÃO GERAL

- 1 Introdução ao Processamento da Informação, 11

UNIDADE 2 *HARDWARE*

- 2 O Processamento de Dados por Computador, 26
- 3 *Hardware* Comum e os Meios de Entrada e Saída, 41
- 4 Características e Aplicações das Comunicações de Dados, 66
- 5 Mini e Microcomputadores, 81

UNIDADE 3 *SOFTWARE*

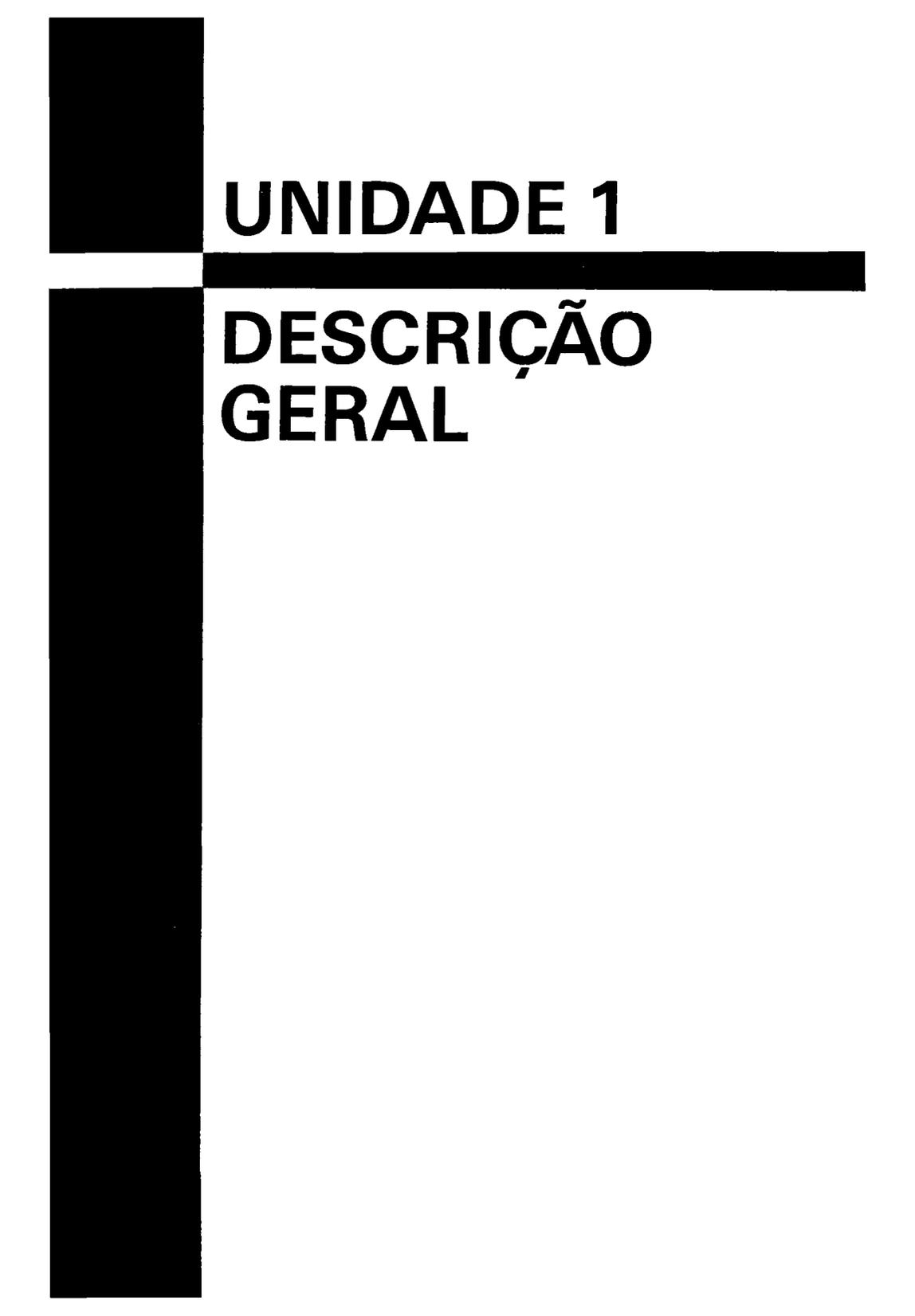
- 6 Conceitos de *Software*, 98
- 7 Instrumentos de Planejamento para Programadores, 117
- 8 Programação em BASIC, 131

UNIDADE 4 SISTEMAS

- 9 Sistemas de Informação, 183
- 10 Sistemas Computadorizados na Sociedade, 197

GLOSSÁRIO, 210

ÍNDICE ANALÍTICO, 223



UNIDADE 1

DESCRIÇÃO GERAL

1

INTRODUÇÃO AO PROCESSAMENTO DA INFORMAÇÃO

- I. INTRODUÇÃO
 - A. O QUE É O PROCESSAMENTO DA INFORMAÇÃO?
 - B. POR QUE USAR COMPUTADORES PARA O PROCESSAMENTO DA INFORMAÇÃO?
 - 1. Velocidade
 - 2. Precisão
 - 3. Capacidade de Memória e Acessibilidade
 - 4. O Baixo Custo do Equipamento de Computador
 - C. POR QUE OS COMPUTADORES ÀS VEZES FALHAM?
 - 1. Erros de Entrada
 - 2. Erros na Instrução a um Computador
 - 3. Comunicação Inadequada entre os Usuários e o Pessoal de Computação
 - 4. Controles Inadequados
 - 5. Custo Elevado da Programação
 - II. CONCEITOS BÁSICOS DE COMPUTAÇÃO
 - A. COMO OS COMPUTADORES OPERAM
 - 1. O sistema de Computador
 - 2. O Conceito de Programa Armazenado
 - 3. Sistemas de Informação: *Hardware* e *Software* Integrados
 - B. TIPOS DE COMPUTADOR
 - III. ORGANOGRAMA E ORGANIZAÇÃO DE UM CENTRO DE COMPUTAÇÃO
 - A. PESSOAL
 - 1. Analista de Sistemas
 - 2. Programador
 - 3. Operador de Computador e de Entrada de Dados
 - 4. Administrador de Banco de Dados
 - 5. Auditor de PED*
 - B. ORGANIZAÇÃO DE UM EQUIPAMENTO DE COMPUTADOR
 - C. TIPOS DE INSTALAÇÕES DE COMPUTADOR
- TERMOS-CHAVE
QUESTIONÁRIO

* Processamento Eletrônico de Dados, em inglês EDP (Electronic Data Processing).

I. INTRODUÇÃO

Nos últimos 30 anos o computador vem causando um profundo impacto no setor de negócios e na sociedade em geral. Não há praticamente nenhuma área de negócios que não possa ser beneficiada, de alguma maneira, pelo uso de um computador. Contudo, os resultados da computadorização nem sempre têm atendido às expectativas e não têm sido sempre satisfatórios. O principal objetivo deste texto é analisar as maneiras pelas quais os computadores podem ser utilizados com eficácia e as razões pelas quais são às vezes usados inadequadamente.

A. O QUE É O PROCESSAMENTO DA INFORMAÇÃO

A principal razão para usar computadores, sem levar em conta a aplicação, é processar os dados de chegada denominados *entrada* e produzir informação de saída denominada *saída*. O termo *processamento da informação* refere-se ao método rápido e eficiente de ler a entrada, processá-la e produzir uma informação de saída precisa e oportuna (veja a Figura 1.1).

Portanto, o processamento da informação consiste geralmente no seguinte:

Operações do Computador

1. Entrada
Leitura de dados de chegada.
2. Processamento e Armazenamento
Execução de operações de aritmética e lógica; armazenamento e recuperação de dados.
3. Saída
Produção de informação de saída.

O processamento do computador é executado por uma série de instruções denominadas *programa*, que lê a entrada, indica as operações a serem realizadas e registra a saída.

O termo **Processamento Eletrônico de Dados (PED)** é usado para indicar o processamento de dados por computadores eletrônicos.

B. POR QUE USAR COMPUTADORES PARA O PROCESSAMENTO DA INFORMAÇÃO?

Algumas vezes é a única maneira de realizar o trabalho.

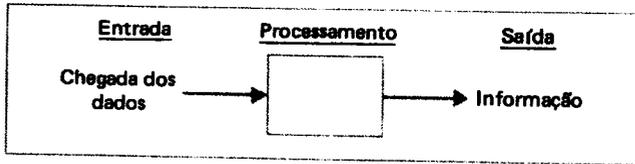


Figura 1.1
Ilustração do processamento de informação.

Algumas operações simplesmente não seriam viáveis hoje em dia sem os computadores. Pense-se no Departamento de Censo Demográfico dos EUA (U. S. Census Bureau), que realiza o recenseamento a cada 10 anos e que em 1980 processou 120 milhões de formulários de recenseamento, contendo respostas para 3,3 bilhões de perguntas. Estes formulários tinham que ser processados até 1º de janeiro de 1981, para que as cifras da população do estado fossem aplicadas na determinação do número proporcional de membros no congresso. Teria sido simplesmente impossível levar a cabo tal processamento sem o auxílio do computador.

Considere-se também a indústria bancária que, só nos Estados Unidos, processa 35 bilhões de cheques por ano. Os cheques têm que ser processados com rapidez suficiente para que possam produzir mensalmente os extratos de conta e manter registros de clientes que sejam pontuais e corretos.

Portanto, o que faz com que os computadores sejam tão importantes são os seguintes elementos:

1. Velocidade

Quando foram usados pela primeira vez em negócios, os computadores processavam os dados a velocidades medidas em milésimos de segundo. Atualmente os computadores processam os dados em *nanossegundos* – bilionésimos de segundo. Em alguns casos os computadores podem processar os dados em *picossegundos* – trilionésimos de segundo – o que começa a se aproximar da velocidade da luz! Esta velocidade de processamento surpreendentemente rápida significa que os computadores podem executar, em apenas alguns segundos, milhares de operações em milhares de entradas de dados.

2. Precisão

Quando as máquinas são programadas corretamente e quando a entrada não contém erro, está praticamente garantido que o computador produzirá resultados corretos. Portanto, os erros produzidos pelo computador simplesmente não ocorrerão se a programação e a entrada de dados forem precisas. Assim, as “histórias pavorosas” sobre falhas de computador em geral não são resultado do funcionamento inadequado da máquina, mas do erro humano – pessoas que escrevem instruções incorretas ou que inserem os dados de maneira inadequada.

3. Capacidade e Acessibilidade de Memória

Há uma enorme variedade de dispositivos com capacidade de armazenar milhões de registros de forma compacta. Além do mais, pode-se ter acesso a estes registros com

rapidez, muitas vezes em questão de segundos. As companhias aéreas, por exemplo, utilizam a capacidade de memória dos meios de computação para manter informações sobre vôos e atualizar registros de vôos quando os clientes solicitam passagens.

4. O Baixo Custo do Equipamento de Computador

A tendência atual na tecnologia de computadores é diminuir o custo e o tamanho do equipamento. Por outro lado, a segurança e a velocidade aumentam. Sendo assim, as companhias que em certa época não podiam justificar a aquisição de um computador, agora consideram viável comprar ou alugar um.

C. POR QUE OS COMPUTADORES ÀS VEZES FALHAM

Apesar das vantagens associadas à computadorização há muitos casos em que os sistemas de computador produzem resultados deletérios e não atendem ao objetivo fixado. Eis aqui alguns dos motivos:

1. Erros de Entrada

A razão mais freqüente para erros de saída é o fato de os dados inseridos no computador não terem sido preparados corretamente. Por exemplo, se um funcionário encarregado do setor de reservas de uma companhia aérea pressiona as teclas para um número de vôo errado, o computador produzirá uma passagem errada.

2. Erros na Instrução a um Computador

Um computador processa dados quando é instruído por um programa. O programador é o técnico de computador que escreve o conjunto de instruções. Às vezes o programador comete erros ou deixa de considerar todas as formas possíveis de entrada. Suponha-se que um programa de Folha de Pagamento, emissor de cheques de pagamento, determine que o SALÁRIO seja de 6 algarismos; se o presidente da companhia ganha Cr\$ 1.180.000 o computador processará simplesmente Cr\$ 180.000 porque foi programado para trabalhar somente com seis algarismos. Este tipo de erro de programação ocorre quando os programadores não examinam todas as condições possíveis.

3. Comunicação Inadequada Entre os Usuários e o Pessoal de Computação

Uma razão importante para os procedimentos computadorizados não cumprirem muitas vezes os seus objetivos é que os técnicos de computação e as pessoas que em realidade utilizam a informação, chamados *usuários*, não se comunicam de modo eficaz. Para que seja satisfatória, a computadorização deve ser entendida e utilizada por todos os usuários dentro de uma organização — isto é, tanto o pessoal de operações quanto o pessoal a nível de gerência.

Os usuários muitas vezes não entendem as limitações de um computador e conseqüentemente as suas expectativas podem não ser totalmente razoáveis. Um problema da mesma gravidade é o fato de o técnico de computador muitas vezes não compreender bem as necessidades ou solicitações dos homens de negócio.

4. Controles Inadequados

À medida que o uso de computadores continua prosperando, a necessidade de controle adequado e de medidas de segurança aumenta assustadoramente. À medida que terminais em locais remotos vão sendo conectados aos computadores, com frequência cada vez maior, maior se torna a necessidade de tal controle e segurança. Porém, muitas organizações não conseguem encarar esta realidade. O resultado tem sido um aumento no número de erros, alguns provenientes de equívocos sem intenção, outros causados por tentativas propositais de fraudar ou de sabotar a companhia através do computador. Além do mais, dependendo da organização, controles inadequados podem levar a uma invasão de privacidade. O governo, a indústria bancária e os hospitais são exemplos de organizações que devem tomar muito cuidado para evitar a possível invasão da privacidade individual.

5. Custo Elevado da Programação

Embora os custos dos equipamentos diminuam significativamente, o custo da programação de um computador está aumentando. A aquisição de pacotes de programação ou a contratação de pessoas para gravar programas personalizados é uma despesa que as companhias freqüentemente subestimam. Além disto, se os usuários não tiverem cuidado, estarão sujeitos a adquirir pacotes ou obter programas personalizados que não satisfazem perfeitamente as suas necessidades.

II. CONCEITOS BÁSICOS DE COMPUTAÇÃO

A. COMO OS COMPUTADORES OPERAM

1. O Sistema de Computador

O termo *sistema de computador* é usado para descrever o grupo de máquinas que executa conjuntamente as tarefas de processamento de informação.

Um sistema de computador é projetado ou configurado para atender às necessidades de cada usuário em particular. Alguns dos dispositivos de computador disponíveis são: unidades de fita magnética, leitoras de cartões perfurados, terminais, unidades de disco magnético etc. Nem todos são requisitados por todos os usuários. Por outro lado, às vezes o equipamento especializado deve ser encomendado para necessidades específicas. Entretanto, todos os sistemas de computador devem ter uma *unidade central de processamento*, ou UCP*, para processar dados.

Um sistema de computador consta dos seguintes elementos:

Elementos de um Sistema de Computador

1. Dispositivos de Entrada
O conjunto de máquinas que lêem diferentes formas de dados de chegada.
2. Unidade Central de Processamento (UCP)
O dispositivo central que armazena e processa a entrada.
3. Dispositivos de Saída
O conjunto de máquinas que produzem a informação de saída.

* Em inglês CPU (Central Processing Unit).

As unidades que compõem um determinado sistema de computador são geralmente ligadas à UCP por um cabo. Os terminais não precisam ser ligados à UCP através de cabo, mas podem se comunicar com o computador usando uma linha telefônica ou outra rede de teleprocessamento. Isto será discutido detalhadamente no Capítulo 4.

2. O Conceito de Programa Armazenado

Antes de poder realmente ler os dados, processá-los e produzir informação, um computador tem que ler um conjunto de instruções, denominado *programa*, que indica o processamento solicitado. O *conceito de programa armazenado* significa que os programas, assim como os dados, são armazenados no computador. O conjunto global de programas que permite que o sistema de computador processe os dados é denominado *software*. Observe-se que alguns sistemas de computador possibilitam que diversos programas sejam executados simultaneamente.

3. Sistemas de Informação: *Hardware* e *Software* Integrados

O processamento de informação necessita, portanto, de:

Hardware — dispositivos físicos em um sistema de computador.

Software — programas usados para processar dados.

Um *sistema de informação* é uma integração do *hardware* e *software* para computadorizar eficazmente os procedimentos negociais. Sistemas de informação podem ser aplicados, por exemplo, no processamento de contas a receber, contas a pagar, de estoques e folhas de pagamento. O Capítulo 9 fornece uma discussão mais detalhada sobre os sistemas de informação.

Como foi ressaltado anteriormente, os custos do *hardware* diminuíram rapidamente nos últimos anos, mas os custos do *software* estão aumentando, porque eles exigem mão-de-obra intensiva. Portanto, o *software* — que está fortemente apoiado na comunicação perfeita entre o usuário e o técnico de computador — deve ser tão eficiente quanto possível. Assim, um sistema de informação deve ter um equilíbrio adequado entre *hardware* e *software* para conseguir os resultados desejados.

B. TIPOS DE COMPUTADORES

Os sistemas de computador são geralmente classificados como grande, médio, pequeno, mini (ou *minicomputador*) e micro (ou *microcomputador*). “Supercomputador” é uma outra categoria reservada para os sistemas mais sofisticados. A Tabela 1.1 fornece uma comparação básica destes sistemas.

Os minicomputadores tornam-se cada vez mais populares, não somente em negócios de pequeno porte, mas também em grandes organizações (Figura 1.2). Muitas vezes uma grande companhia amplia a potência do computador em uso com minicomputadores, ou “troca” um sistema grande por uma rede de minicomputadores.

Tanto os micro como os minicomputadores tornaram-se populares em negócios de pequeno porte e escritórios profissionais, bem como em grandes organizações e no lar.

Os mini e microcomputadores são considerados por muitos como “a onda do futuro”, com potencial para substituir totalmente os sistemas maiores.

Tabela 1.1 Comparação das Várias Categorias de Computadores

<i>Características</i>	<i>Sistemas de Computadores Grandes</i>	<i>Sistemas de Computadores Médios</i>	<i>Sistemas de Computadores Pequenos</i>	<i>Sistemas de Minicomputadores</i>	<i>Sistemas de Microcomputadores</i>
Preço de compra (US\$)	1.000.000 +	250.000 a 1.000.000	50.000 a 250.000	2.000 a 50.000 (forma básica)	Várias centenas a 10.000
Capacidade de memória principal	Extremamente grande	Grande	Média	Média	Limitada
Velocidade de processamento	Extremamente rápida	Rápida	Moderada	Moderada	Lenta
Número de dispositivos de Entrada/Saída que podem ser fornecidos	Extremamente grande	Grande	Moderado	Moderado	Limitado
Apoio de <i>Software</i>	Amplio	Considerável	Moderado	Moderado	Limitado

Nota: Os termos relativos "grande", "rápido", "moderado" etc. estão incluídos aqui para fornecerem uma compreensão geral de como os sistemas de computador se diferenciam.¹

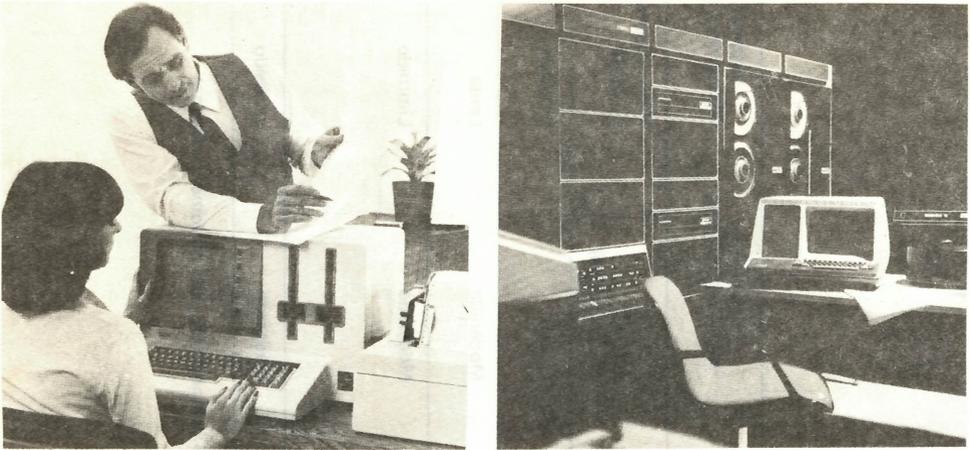


Figura 1.2

Dois tamanhos diferentes de minicomputadores (à esquerda – cortesia da IBM; à direita – Digital Equipment Corporation).

III. ORGANOGRAMA E ORGANIZAÇÃO DE UM CENTRO DE COMPUTAÇÃO

A. PESSOAL

1. Analista de Sistemas

O *analista de sistemas* é o técnico de computador encarregado de analisar os procedimentos administrativos existentes, determinar as ineficácias ou áreas problemáticas básicas e projetar um conjunto de métodos ou sistema de computação mais eficaz. Em geral, o analista de sistemas, além de conhecer muito bem o computador, deve ter experiência administrativa, capacidade de comunicação e às vezes um diploma de nível avançado, de preferência em administração de empresa, ciência administrativa, ou ciência da computação.

O gerente de sistemas determina tarefas específicas para cada analista e supervisiona o progresso destes.

2. Programador

O *programador* recebe os pedidos de serviço do analista de sistemas e é responsável pelo registro e teste do programa que integrará o sistema como um todo. O programador deve geralmente ter formação universitária, com prática em administração ou ciência da computação.

O gerente de programação é o indivíduo que supervisiona as atividades de um grupo de programadores.

3. Operador de Computador e de Entrada de Dados

O *operador de computador* supervisiona uma operação de computador apertando as teclas apropriadas e monitorizando o *hardware*. O *operador de entrada de dados* é

responsável pela inserção de dados nos cartões perfurados, fita, disco, ou um terminal, de maneira que eles possam ser lidos pelo computador. Cargos como o de operador de computador e operador de entrada de dados são geralmente considerados trabalhos semi-especializados, exigindo na maioria das vezes somente um diploma de nível secundário.

O gerente de operações é responsável pela supervisão das atividades gerais de todos os operadores e por manter uma margem mínima de erros.

4. Administrador de Banco de Dados

O *administrador de banco de dados* é responsável pela estrutura, planejamento e controle de todos os arquivos de processamento de dados. O administrador de banco de dados organiza e planeja o conjunto de dados usados pela organização como um todo. Ele é tido como o responsável pelo projeto eficiente do banco de dados e pela implementação dos controles e técnicas adequados necessários para acessar o banco de dados.

5. Auditor de PED

O *auditor de PED* é um especialista em contabilidade e computação, responsável pela avaliação da eficiência do sistema de computador e pela garantia da sua integridade total.

B. ORGANIZAÇÃO DE UM EQUIPAMENTO DE COMPUTADOR

As Figuras 1.3 e 1.4 indicam as duas maneiras principais de organizar um centro de computação. Em algumas organizações os programadores estão subordinados aos analistas de sistemas; em outras, os programadores estão subordinados aos gerentes de programação e não respondem diretamente ao analista de sistemas. Quando os programadores estão subordinados aos analistas de sistemas, há mais unidade dentro do sistema, mas o programador tem menos independência e às vezes menos mobilidade dentro da hierarquia.

C. TIPOS DE INSTALAÇÕES DE COMPUTADOR

As instalações de computador estão organizadas tipicamente de três maneiras principais:

Tipos de Instalação de Computador

- | | |
|------------------------|---|
| Centralizada | Há uma instalação central de computador que fornece serviços de computação para todos os usuários. |
| Descentralizada | Cada departamento tem a sua própria instalação de computador, que funciona autonomamente com o seu equipamento próprio e pessoal de apoio. |
| Distribuída | Há instalações independentes para cada departamento, mas há também uma instalação central para controlar e ordenar as atividades de cada uma. |

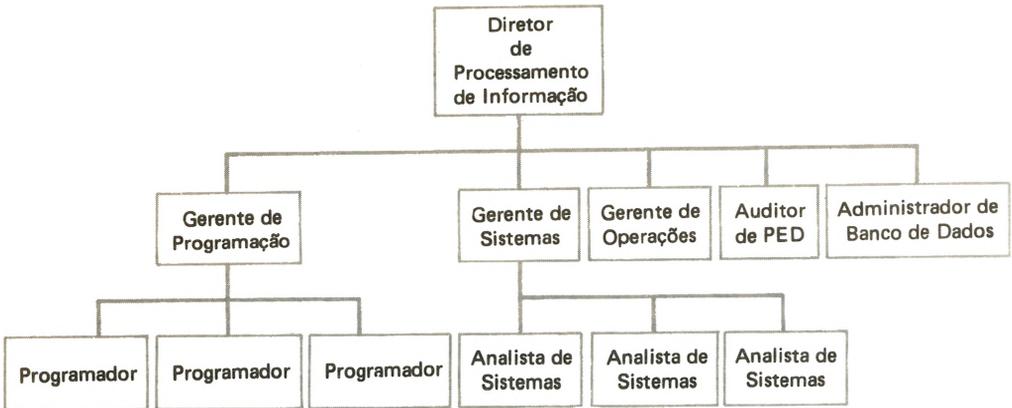


Figura 1.3
Organograma organizacional: os programadores e analistas estão subordinados a gerentes diferentes.

Em uma *instalação centralizada de processamento de dados* há geralmente um computador principal que, para a maioria das organizações de grande porte, é de tamanho grande ou médio. Este tipo de organização significa que todos os usuários têm que utilizar este único sistema e os pedidos de computadorização devem ser tratados com base em uma escala de prioridades.

O desenvolvimento dos minicomputadores foi em grande parte responsável pelos dois tipos de instalações de computador mais recentes: descentralizada e distribuída.

Figura 1.4
Organograma organizacional: os programadores estão subordinados aos analistas.

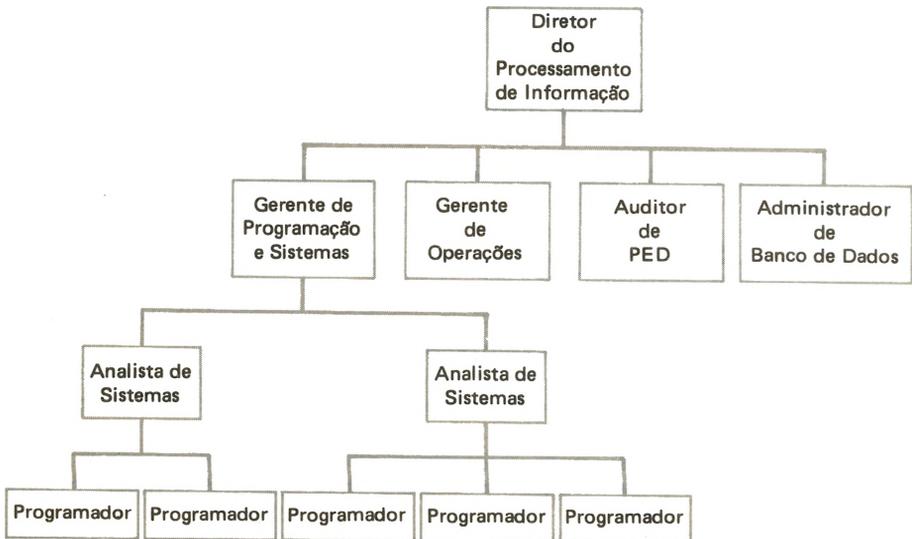


Tabela 1.2 Resumo

	Vantagens	Desvantagens
PD centralizado	<ol style="list-style-type: none"> 1. Resulta instalações e procedimentos padronizados para a companhia como um todo. 2. Os especialistas de computador trabalham em grupo — os procedimentos do PD centralizado tornam-se mais profissionais; melhor supervisão das atividades. 3. Reduz a duplicação de esforço. 	<ol style="list-style-type: none"> 1. É difícil avaliar as necessidades do PD de cada usuário, os custos e a eficiência do PD para cada usuário. 2. É difícil determinar as prioridades para o uso do computador.
PD descentralizado	<ol style="list-style-type: none"> 1. O controle direto pelos usuários minimiza a defasagem de comunicação tradicional; os técnicos de computador entendem melhor as necessidades do departamento. 2. Resposta mais direta às necessidades dos usuários; não existe pressão de outros departamentos. 3. É mais fácil determinar a eficiência de cada instalação do PD. 	<ol style="list-style-type: none"> 1. Há duplicação de arquivos, processamento e relatórios. 2. Não existe padronização. 3. Pode ser mais caro ou não.
PD distribuído (PDD)	<ol style="list-style-type: none"> 1. Reúne as vantagens do processamento de dados centralizado e descentralizado. 2. Utiliza uma abordagem integrada para os sistemas. 3. O melhor método para dar lugar ao crescimento. 	<ol style="list-style-type: none"> 1. Exige um controle sofisticado de <i>hardware</i> e <i>software</i>. 2. Exige uma fidelidade rigorosa aos padrões. 3. Pode ser caro.

Uma *instalação descentralizada de processamento de dados* permite que cada departamento ou unidade dentro de uma companhia estabeleça o seu próprio sistema de computador, geralmente um minicomputador, e forneça ao centro técnicos de computador que conheçam as necessidades do departamento. As necessidades superpostas entre os departamentos, tais como Folhas de Pagamento e Pessoal ou Marketing e Vendas, são ignoradas, pois cada área provavelmente terá a sua própria instalação. Isto freqüentemente resulta em uma duplicação de esforços.

Uma rede de *Processamento de dados distribuída* (PDD) é uma organização planejada para somar as vantagens de ambas as instalações: centralizada e descentralizada. Em uma rede de PDD, cada departamento tem acesso aos seus próprios dados por meio de um minicomputador ou um grupo de terminais. No entanto há também uma instalação centralizada que supervisiona e coordena as atividades de cada unidade administrativa. O computador central usado em uma *rede* de PDD deve ser suficientemente grande para permitir o acesso de muitas estações.

Para um resumo das características dos sistemas de computador centralizado e distribuído, consulte a Tabela 1.2.

TERMOS-CHAVE

Administrador de banco de dados
Analista de sistemas
Auditor de PED
Conceito de programa armazenado
Entrada
Unidade Central de Processamento (UCP)
Instalação centralizada de processamento de dados
Instalação descentralizada de processamento de dados
Instalação distribuída de processamento de dados
Hardware
Microcomputador
Microsegundo
Minicomputador
Nanosegundo
Operador de computador
Operador de entrada de dados
Processamento de informação
Processamento eletrônico de dados (PED)
Programa
Programador
Rede
Saída
Sistema de computador
Sistema de informação
Software
Usuário

QUESTIONÁRIO

(As respostas para as questões de números pares acompanham este teste.)

A. Questões do Tipo Verdadeiro-Falso

V F

_____ 1. O termo "dados" refere-se à informação que foi processada pelo computador.

- _____ 2. A informação de saída que foi processada pelo computador é denominada saída.
- _____ 3. Os computadores mais modernos operam dados em velocidades medidas em milésimos de um segundo.
- _____ 4. Uma instalação centralizada de computação significa geralmente o uso de uma única unidade central de processamento.
- _____ 5. As instalações descentralizadas de computação geralmente possibilitam a cada grupo de usuários o acesso ao seu próprio computador.
- _____ 6. Uma instalação distribuída de computação requer geralmente o uso de mini-computadores ou terminais em localizações-chave da companhia.
- _____ 7. Os usuários em um ambiente comercial consistem de pessoal de operações bem como de gerentes.
- _____ 8. A razão básica pela qual os computadores às vezes não produzem a entrada correta é o mal funcionamento da máquina.
- _____ 9. *Software* refere-se aos dispositivos periféricos usados com uma UCP.
- _____ 10. Os mini e microcomputadores tornaram-se populares recentemente.

B. Preencha as Lacunas

1. Os computadores lêem os dados de chegada denominados
2. O termo refere-se ao processamento global dos dados para produzir uma informação significativa.
3. Os computadores operam dados usando uma série de instruções denominadas
4. O dispositivo de computador que realmente executa o processamento solicitado é denominado
5. O conjunto de dispositivos de entrada e saída associado a um sistema de computador é chamado
6. A falta de adequada entre os usuários e os técnicos de computador é uma razão importante para os sistemas computadorizados deixarem de produzir às vezes uma saída significativa.
7. Um é um especialista em contabilidade e computação, responsável pela eficiência do sistema de computador e pela garantia de sua integridade total.
8. O técnico de computador responsável pela determinação das áreas problemáticas em um conjunto de procedimentos administrativos existentes e pelo replanejamento desses procedimentos administrativos de maneira que eles sejam mais eficientes é chamado
9. O técnico de computador responsável pela estrutura, planejamento e controle geral dos arquivos de processamento de dados é o
10. O conjunto de programas que permite ao sistema de computador processar os dados é chamado

Respostas

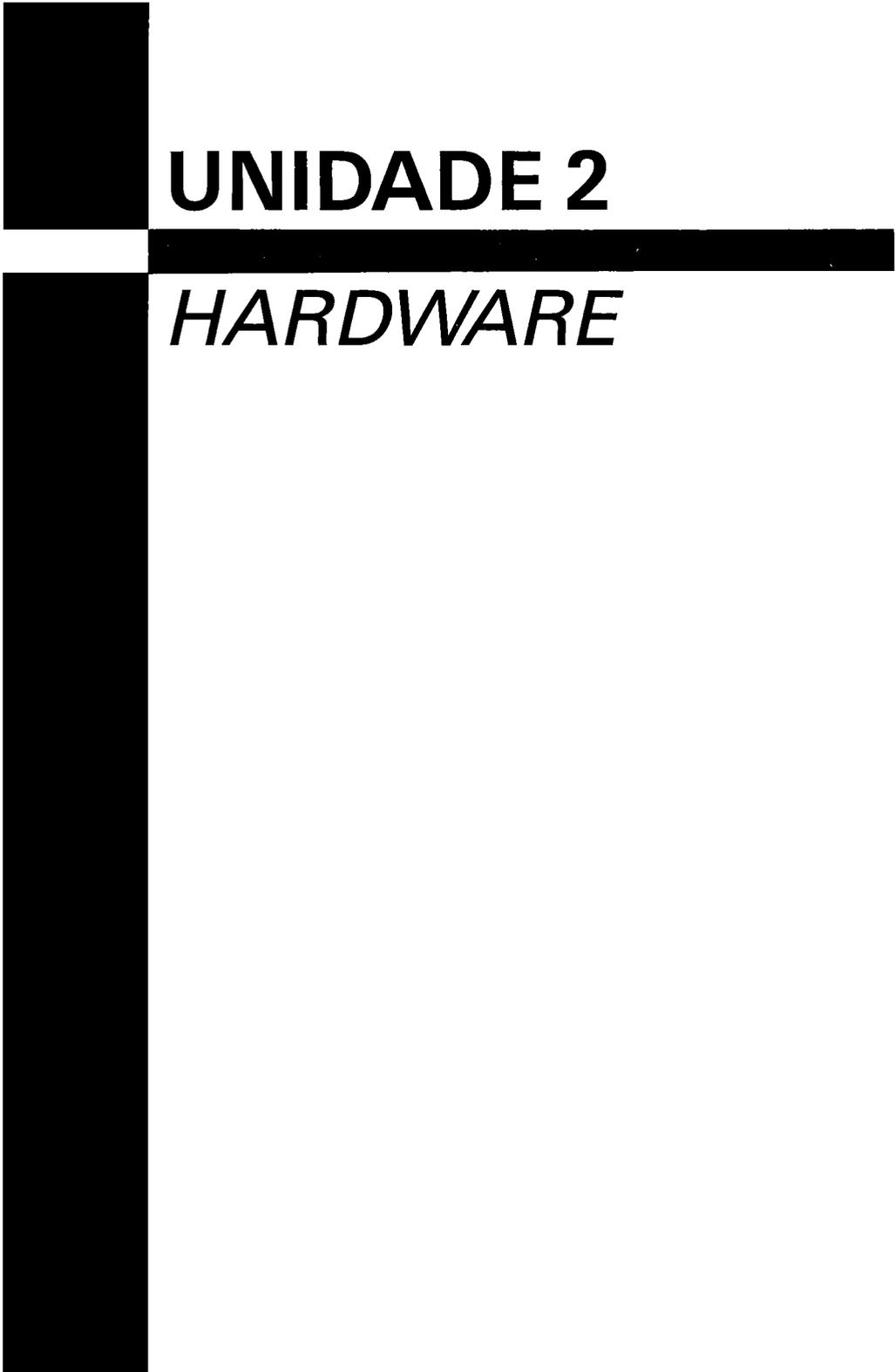
(Fornecemos respostas somente para as questões de números pares.)

A. Questões do Tipo Verdadeiro-Falso

2. V 4. V 6. V
8. F — erros de entrada e programação são os responsáveis na maioria das vezes.
10. V

B. Preencha as lacunas

2. Processamento de informação
4. Unidade central de processamento (UCP)
6. Comunicação
8. Analista de sistemas
10. *Software*



UNIDADE 2

HARDWARE

2

O PROCESSAMENTO DE DADOS POR COMPUTADOR

- I. O SISTEMA DE COMPUTADOR
 - A. DISPOSITIVOS DE ENTRADA E SAÍDA
 - B. UNIDADE CENTRAL DE PROCESSAMENTO
 - 1. Memória Primária
 - 2. Unidade de Controle
 - 3. Unidade Aritmética e Lógica
 - C. MEMÓRIA AUXILIAR OU SECUNDÁRIA
- II. MÉTODOS DE PROCESSAMENTO DE DADOS
 - A. PROCESSAMENTO POR LOTES
 - B. PROCESSAMENTO EM LINHA
 - C. PROCESSAMENTO FORA DE LINHA
- III. REPRESENTAÇÃO DOS DADOS NA UCP
- IV. UMA VISÃO MAIS DETALHADA DA MEMÓRIA
 - A. TIPOS DE MEMÓRIA DA UCP
 - 1. Memória de Núcleo Magnético
 - 2. Circuitos Integrados
 - 3. Avanços na Tecnologia de Memória
 - B. CARACTERÍSTICAS DA MEMÓRIA PRIMÁRIA
 - 1. Memória de Acesso Aleatório (MAA)*
 - 2. Memória de Leitura Somente (MLS)**

TERMOS-CHAVE

QUESTIONÁRIO

* Em inglês RAM (Random Access Memory).

** Em inglês ROM (Ready Only Memory).

I. O SISTEMA DE COMPUTADOR

Cada equipamento de computador é composto de uma série de dispositivos que operam em conjunto, como uma unidade integrada ou *sistema de computador*. Cada sistema de computador consiste de máquinas individuais que incluem os seguintes elementos:

Elementos de um Sistema de Computador

1. Unidades de Entrada
Lêem os dados de entrada transmitindo-os à unidade central de processamento.
2. Unidade Central de Processamento (UCP)
Controla todas as operações do sistema de computador; fornece memória primária.
3. Unidade de Memória Auxiliar ou Secundária
Fornecer memória adicional para aumentar a memória primária.
4. Unidades de Saída
Transmitem os dados processados da UCP para o formulário de saída.

O esquema na Figura 2.1 ilustra a integração dos elementos básicos em um sistema de computador. Veja o Quadro 2.2 para uma ilustração de um sistema de computador típico.

Observe que um sistema de computador consiste de uma série de máquinas independentes, ou *hardware*, que funcionam de maneira integrada para produzir a saída desejada. Observe também que os sistemas de computador possuem uma enorme variedade de dispositivos de entrada/saída (E/S) que podem ser ligados à *unidade principal* ou UCP.

Neste capítulo discutiremos as características básicas de um sistema de computador. É bom lembrar que para processar os dados, em qualquer momento dado, um sistema de computador tem que estar sob o controle de um *programa*, ou seja, uma seqüência de instruções indicando quais as unidades a serem usadas de modo exato como os dados devem ser processados. Este programa, escrito por um programador, deve ser lido na UCP antes que o processamento tenha início.

A. DISPOSITIVOS DE ENTRADA E SAÍDA

Este capítulo descreve a maneira pela qual a unidade de processamento central opera com os dados. Os dois capítulos seguintes fazem referência aos dispositivos de entrada e saída. A Figura 2.2 fornece uma descrição simplificada de um dispositivo de

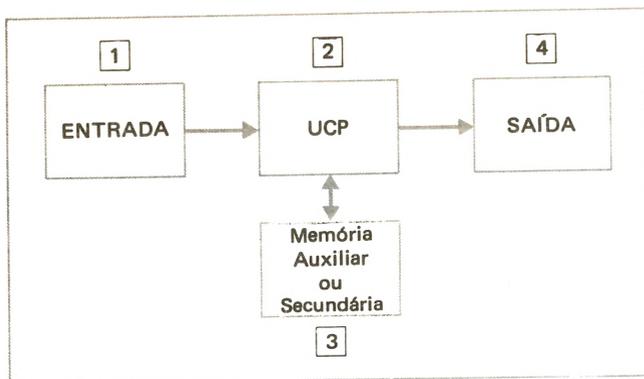


Figura 2.1
Os elementos básicos em um sistema de computador.

entrada típica para um computador. Para uma breve descrição de um dispositivo de saída típico veja a Figura 2.4

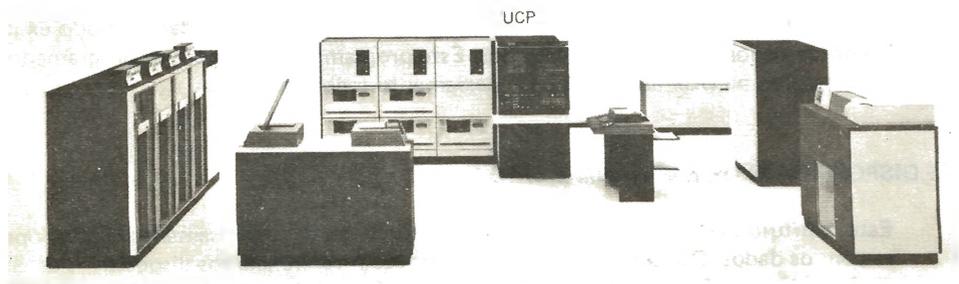
B. A UNIDADE CENTRAL DE PROCESSAMENTO

Todas as operações executadas pelo sistema de computador são controladas pela *unidade central de processamento ou UCP*. A UCP é ligada a cada dispositivo de entrada/saída através de cabo e quando programada ela controla o seguinte:

Operações da UCP

1. Leitura de um dispositivo de entrada.
2. Processamento dos dados de entrada, incluindo:
 - operações aritméticas
 - testes lógicos em dados
3. Registro da informação usando um dispositivo de saída.

Figura 2.2
Sistema IBM 370. (Cortesia da IBM)



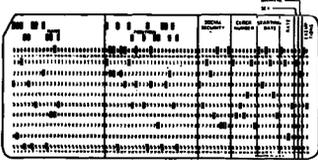
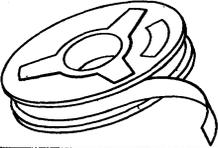
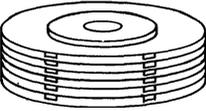
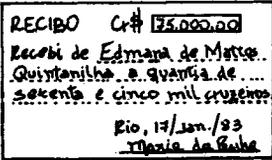
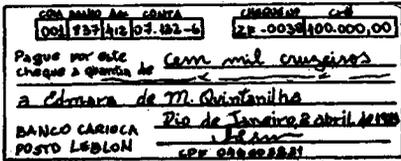
DISPOSITIVO	FORMA DE ENTRADA	DESCRIÇÃO
Leitora de Cartão	<p>Cartão perfurado</p> 	Os dados são registrados sob forma de perfurações.
Unidade de fita magnética	<p>Fita magnética</p> 	Os dados são registrados sob forma de pontos magnetizados ou <i>bits</i> .
Unidade de disco magnético	<p>Disco magnético</p> 	Os dados são registrados sob forma de pontos magnetizados ou <i>bits</i> .
Terminal	<p>Entrada de dados pelo teclado</p> 	Os dados são inseridos a partir de localizações remotas.
Fita de papel perfurada	<p>Fita de papel Alimentação de linha Retorno do carro Espaço</p> 	Os dados são registrados sob forma de perfurações em uma fita de papel contínua.
Leitora ótica	<p>Caracteres datilografados ou manuscritos</p> 	Os dados são registrados em formulários impressos ou manuscritos.
Leitora de reconhecimento de caracteres em tinta magnética	<p>Cheques bancários</p> 	Os dados são registrados em tinta magnética na parte inferior dos cheques.

Figura 2.3 Dispositivos de entrada de um sistema de computador.

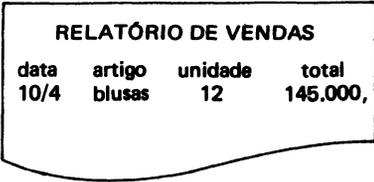
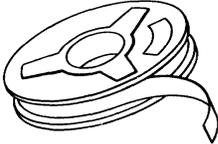
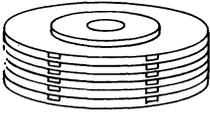
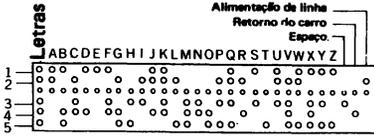
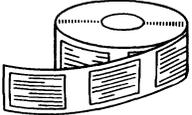
Dispositivo	Forma de Saída	Descrição
Perfuradora de cartões		Perfura cartões como saída.
Impressora		Produz um relatório impresso.
Unidade de fita magnética		Este é um dispositivo de entrada-saída que pode ler ou gravar a fita magnética.
Unidade de disco magnético		Este é um dispositivo de entrada-saída que pode ler ou gravar o disco magnético.
Terminal		Este é um dispositivo de entrada-saída que pode ser usado para inserir dados ou obter informação produzida pelo computador.
Perfuradora de fitas de papel		Produz perfurações em uma fita de papel contínua.
Microfilme de saída de computador		Produz dados de saída sob a forma de registros microfilmados em miniatura.

Figura 2.4
Dispositivo de saída de um sistema de computador.

Os elementos da unidade central de processamento consistem de três partes:

1. Memória Primária

A memória primária é algumas vezes denominada *memória principal*. É a unidade da UCP onde se coloca o programa para ser rodado ou executado. De maneira semelhante, os dados devem estar na memória primária para poderem ser processados.

A *memória primária* consiste de posições de memória endereçáveis denominadas *bytes* de memória. Cada *byte* possui um único caracter, isto é, uma letra ou um dígito. Um registro de dados que tem 150 caracteres necessitaria de 150 bytes de memória quando lido no computador.

A capacidade de memória primária dos computadores é denominada *tamanho de memória*. O tamanho de memória dos microcomputadores pequenos pode variar de 4000 a 16000 *bytes*, escritos simbolicamente como 4K a 16K, onde K é aproximadamente igual a 1000 *bytes* (na verdade, equivale a 1024 *bytes*). Os sistemas em grande escala têm capacidades de memória superiores a 1 milhão de *bytes*, comumente denominado um *megabyte*. Há uma enorme variedade de computadores com capacidades de memória que se estendem de alguns milhares de *bytes* a vários *megabytes*.

2. Unidade de Controle

A *unidade de controle* da UCP supervisiona ou controla as funções desempenhadas por todo o sistema de computador. Um programa de controle denominado *supervisor* controla e supervisiona as operações do sistema de computador.

Um programa fornecido pelo programador é lido na memória primária, mas cada instrução é interpretada e executada pela seção de controle da UCP.

3. Unidade Aritmética e Lógica

A *unidade aritmética e lógica* da UCP controla todas as operações de aritmética e de comparação.

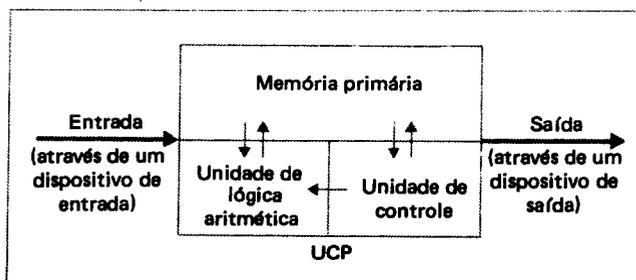
A Figura 2.5 fornece uma ilustração dos componentes básicos da UCP.

C. MEMÓRIA AUXILIAR OU SECUNDÁRIA

As *unidades de memória auxiliar* que consistem geralmente de um tambor ou disco magnético, e algumas vezes de uma fita magnética, fornecem à UCP uma capacidade

Figura 2.5

Componentes básicos de uma UCP.



de memória adicional. Os dispositivos de armazenamento de massa são freqüentemente usados como memória secundária, aumentando consideravelmente a capacidade de memória de um sistema de computador.

Apresentamos a seguir um resumo dos componentes de um sistema de computador.

RESUMO: COMPONENTES DE UM SISTEMA DE COMPUTADOR

A. Dispositivos de Entrada

Exemplos: Terminal

Unidade de fita magnética

Unidade de disco magnético

Leitora de cartões

B. Unidade Central de Processamento (UCP) ou Unidade Principal

1. Memória primária

Consiste de localizações endereçáveis

Tamanho de memória — número de localizações endereçáveis

Tamanho de memória descrito em termos de:

K — 1000 *bytes* aproximadamente (1024) *megabytes* — um milhão de *bytes* de memória.

2. Unidade de controle

Fornece controle central à UCP.

3. Unidade aritmética e lógica

Realiza todas as operações de controle aritmético e lógico.

C. Memória Auxiliar ou Secundária

Fornece memória adicional para processamento.

Ligada à UCP.

Geralmente em fita, disco ou tambor magnéticos, ou dispositivo de armazenamento de massa.

D. Unidade de Saída

Exemplos: Impressora

Terminal

Unidade de fita magnética

Unidade de disco magnético

Perfuradora de cartões

II. MÉTODOS DE PROCESSAMENTO DE DADOS

A. PROCESSAMENTO POR LOTES

O método tradicional para processar dados é acumular um grande volume de dados de entrada e processá-los periodicamente para produzir algum formulário de saída. Por exemplo, as notas das vendas diárias são acumuladas e, no final de cada mês, são usadas para imprimir a fatura de cada freqüês. Este método de processamento de dados em intervalos regulares em grupos, ou lotes, é denominado *processamento por lotes*.

Uma grande desvantagem do processamento por lotes é que ele não é sempre útil para fornecer informação a tempo. O processamento mensal das notas de vendas significa que a relação de gastos de um freqüês não pode ser determinada a cada dia.

B. PROCESSAMENTO EM LINHA

O *processamento em linha* proporciona um acesso imediato à UCP, seja para entrada de dados ou para fins de consulta. Desta maneira, as notas de vendas podem ser processadas no momento da venda; assim, o arquivo do freguês conteria sempre a informação atual. Os terminais são freqüentemente usados para inserir dados em linha em uma UCP.

É evidente que o processamento em linha é mais flexível que o processamento por lotes, mas por outro lado exige mais da UCP. Em uma operação por lotes, o arquivo do freguês estaria à disposição da UCP uma só vez — durante o momento específico reservado para o processamento de dados. Em uma operação em linha, o arquivo do freguês deve estar sempre disponível para poder ser imediatamente alterado quando se realiza uma venda.

Um sistema em linha que funciona com rapidez suficiente que possibilite ao usuário tomar decisões é classificado como um *sistema de processamento em tempo real*. Um sistema de reservas de uma companhia aérea que fornece informação sobre a condição de cada voo e emite imediatamente as passagens, caso solicitadas, é um sistema de processamento em tempo real. Os passageiros podem obter respostas do sistema em linha com rapidez suficiente para que possam decidir que voo reservar.

C. PROCESSAMENTO FORA DE LINHA

Não há dúvida de que o processamento por computador é caro. Portanto, qualquer operação que consegue evitar o uso inútil do computador pode economizar quantias significativas para a companhia. O *processamento fora de linha* é o que não está sob controle direto da UCP principal e é, conseqüentemente, mais barato.

Por exemplo, um terminal pode ser usado para, através do teclado, fazer entrar dados não *diretamente* na UCP, mas em um disco magnético pequeno, denominado disco flexível. No final do dia, ou em algum intervalo determinado, o disco pode ser inserido, em alta velocidade e por lotes, na UCP principal. Esta operação teclado-disco é denominada processo fora de linha e é amplamente usada em aplicações de entrada de dados.

I. REPRESENTAÇÃO DOS DADOS NA UCP

Todos os computadores usam alguma variação do sistema numérico binário para representar cada *caracter*, onde um caracter é definido como qualquer letra, dígito ou símbolo especial.

No *sistema numérico binário* podem-se usar somente dois dígitos: 0 e 1. Este sistema é ideal para processamento por computador porque o "1" é usado para indicar a presença de um pulso ou sinal elétrico no circuito do computador, e o "0" é usado para indicar a ausência de tal sinal. A Figura 2.6 ilustra a maneira pela qual o computador utiliza o sistema numérico binário.

Cada dígito decimal pode ser representado usando números binários:

<i>Decimal</i>	<i>Binário</i>
0	0
1	1
2	10
3	11
4	100
5	101
6	110
⋮	⋮

Cada dígito binário ou *bit* indica a presença ou ausência de um dígito binário com valor decimal 2^N , onde $N = 0, 1, 2, \dots$

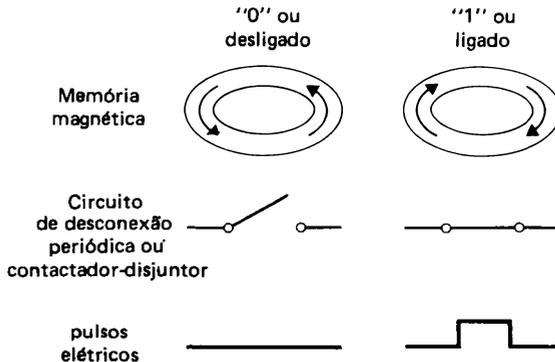
...	2^3	2^2	2^1	2^0	dígitos binários ou <i>bits</i>
...	8	4	2	1	valor decimal

11 em binário, por exemplo, é equivalente ao decimal 3:

valor exponencial	2^1	2^0	
valor decimal	2	1	
valor binário	1	1	11 em binário indica a presença do <i>bit</i> 2 e do <i>bit</i> 1; portanto, é igual ao decimal número 3.

Figura 2.6

Como o computador utiliza o sistema numérico binário.



Apresentamos a seguir um resumo das características da representação de dados.

RESUMO: REPRESENTAÇÃO DE DADOS

1. Os computadores utilizam alguma forma de sistema numérico binário.
 - a. Uso de 0 e 1 para representar todos os caracteres.
 - b. 0 = desligado.
 - c. 1 = ligado.
2. Todos os números decimais podem ser convertidos em binários e os números binários podem ser usados para representar qualquer número decimal.
3. Os computadores usam freqüentemente 8 dígitos binários, denominados *bits*, para representar cada caractere.
4. Códigos de computador
 - a. EBCDIC — Código Ampliado de Caracteres Decimais Codificados em binário para o intercâmbio da informação (um código de 8 *bits*).
 - b. ASCII — Código Padrão Americano para Intercâmbio de Informações (pode ser um código de 7 ou 8 *bits*).

IV. UMA VISÃO MAIS DETALHADA DA MEMÓRIA

A. TIPOS DE MEMÓRIA DA UCP

1. Memória de Núcleo Magnético

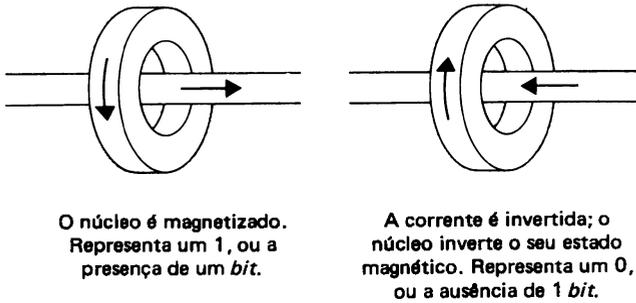
No passado, a maioria dos computadores usava os *núcleos magnéticos* para representar dados e instruções na memória. Um núcleo magnético é um minúsculo elemento ferrite com formato de rosca, mais ou menos do tamanho de um grão de sal.

Um núcleo magnético pode ser magnetizado em uma de duas direções, para a direita ou para a esquerda. Se um núcleo é magnetizado para a direita diz-se que está “ligado” e se é magnetizado para a esquerda diz-se que está “desligado”. Os núcleos “ligados” representam o dígito binário 1 e os núcleos “desligados”, o dígito binário 0. Os planos ou grupos de núcleos são empilhados para formar os dígitos binários que representam todos os caracteres na memória, geralmente oito dígitos binários (*bits*) para uma posição de memória (*byte*) — veja a Figura 2.7.

2. Circuitos Integrados

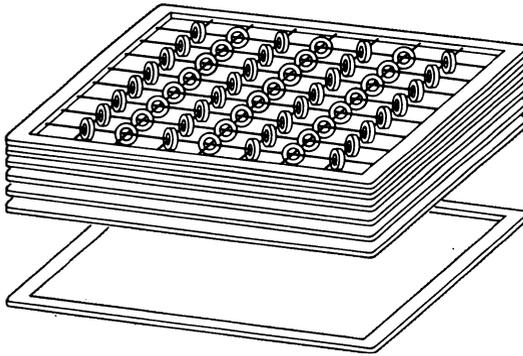
Os computadores mais modernos utilizam memórias que consistem de circuitos integrados. Os *circuitos integrados* são compostos de centenas, ou até milhares, de componentes eletrônicos em uma minúscula placa de silicóne. O fluxo de corrente através de um circuito representa um estado “ligado”, usado para especificar o dígito binário 1.

Milhares de circuitos integrados podem ser colocados em um único *chip* não maior do que uma unha do polegar (veja a Figura 2.8). Estes *chips* deram origem a computadores menores com memórias muito amplas. Na verdade, podem-se



O núcleo é magnetizado. Representa um 1, ou a presença de um *bit*.

A corrente é invertida; o núcleo inverte o seu estado magnético. Representa um 0, ou a ausência de 1 *bit*.



Núcleo em um plano.

Figura 2.7
Memória de núcleo magnético.

encontrar *chips* isolados que possuem a capacidade de memória dos sistemas em grande escala da década passada. Um *chip* pode armazenar 64,000 *bytes* de memória ou mais. Além disto, os *chips* permitem que a manutenção do computador seja fácil e eficiente. Quando ocorrem erros, os *chips* podem ser testados e os que estão funcionando mal são retirados e substituídos facilmente. Além disso, o tamanho de memória de um computador pode ser aumentado com facilidade pela simples adição de *chips* ou placas, onde uma placa contém diversos *chips*. A Figura 2.9 mostra duas placas, cada uma contendo 64,000 *bytes* de memória.

3. Avanços na Tecnologia de Memória

Há inúmeras organizações que estão desenvolvendo outros tipos de memórias que podem vir a substituir os circuitos integrados nos próximos anos. A *memória de bolha magnética*, desenvolvida recentemente, tem uma grande vantagem em relação aos outros tipos de memória: os dados podem ser mantidos na memória mesmo se a força estiver desligada. Atualmente, a memória de bolha magnética é

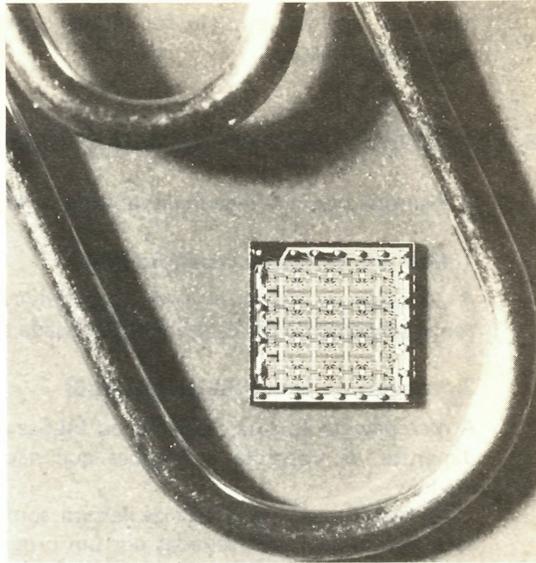
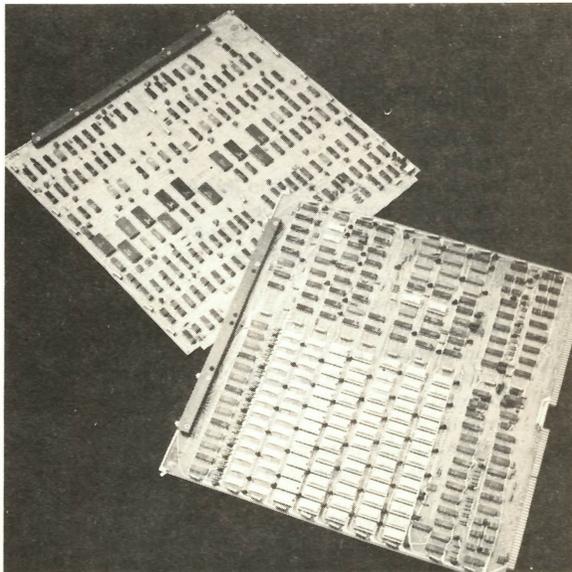


Figura 2.8
Um *chip* comparado em tamanho com um grampo para papéis.
(Cortesia da IBM).

Figura 2.9
Duas placas, cada uma com memória de *byte* de 64K. (Cortesia de Data General).



cara, mas quando o custo for reduzido poderá tornar-se um importante tipo de memória no futuro.

B. CARACTERÍSTICAS DA MEMÓRIA PRIMÁRIA

A memória primária é geralmente subdividida em duas áreas:

1. Memória de Acesso Aleatório (MAA)

A *Memória de Acesso Aleatório* ou MAA é a parte da memória usada para armazenar programas e dados. Pode ser acessada e modificada conforme necessário.

2. Memória de Leitura Somente (MLS)

A *Memória de Leitura Somente* ou MLS é a parte da memória que contém procedimentos de controle e funções que não podem ser modificadas pelo programador.

As funções em memória de leitura somente são armazenadas permanentemente e não podem ser alteradas por um programa. A expressão "Memória de leitura somente" indica que esta área de armazenamento pode ser acessada, mas não pode ser usada para armazenar instruções ou dados. A MLS pode ser modificada somente se os cabos dos circuitos forem trocados.

TERMOS-CHAVE

ASCH

Bit

Byte

Caractere

Chip

Circuito integrado

EBCDIC

Hardware

Megabyte

Memória de acesso aleatório (MAA)

Memória de bolha magnética

Memória de leitura somente (MLS)

Memória de núcleo magnético

Memória primária

Memória principal

Número binário

Processamento em linha

Processamento fora de linha

Processamento por lotes

Programa

Sistema de computador

Sistema de processamento em tempo real

Supervisor

Tamanho de memória

Unidade aritmética e lógica

Unidade Central de Processamento (UCP)

Unidade de controle
Unidade de memória auxiliar
Unidade principal

QUESTIONÁRIO

(As respostas para as questões de números pares acompanham o teste.)

A. Questões do Tipo Verdadeiro-Falso:

V F

- _____ 1. Todo sistema de computador deve ter uma UCP.
- _____ 2. Todos os sistemas de computador IBMs/38, por exemplo, teriam os mesmos dispositivos de entrada e saída.
- _____ 3. Alguns dispositivos são capazes tanto de ler como de escrever os dados.
- _____ 4. Para arquivos que necessitam de uma atualização periódica usa-se o processamento por lotes.
- _____ 5. Uma desvantagem do processamento em tempo real é que os arquivos mestres não estão sempre atualizados.
- _____ 6. Cada instrução em memória é considerada como um *byte*.
- _____ 7. A capacidade total de memória de um computador é conhecida como o seu tamanho de *bit*.
- _____ 8. A maioria dos computadores utiliza de algum modo o sistema numérico binário para seus códigos internos.
- _____ 9. O processamento em linha refere-se às operações que não estão sob controle direto de uma UCP.
- _____ 10. A memória de bolha magnética faz com que o computador conserve os conteúdos de memória mesmo depois que a máquina é desligada.

B. Preencha as Lacunas

- 1. O programa armazenado para processar dados está geralmente contido no(a) da UCP.
- 2. A parte da memória que pode ser acessada mas não pode ser alterada por cada programa é denominada
- 3. A forma mais comum de memória de computador em uso atualmente é
- 4. Uma capacidade de memória de 1 milhão de *bytes* é chamada de memória.
- 5. A notação é freqüentemente usada como abreviação de cerca de 1000 posições de memória.
- 6. Quando as operações em linha são executadas com rapidez suficiente para influir na tomada de decisões são consideradas como operações
- 7. O termo é uma abreviatura para dígito binário.
- 8. Quando o computador precisa executar uma operação aritmética ou fazer comparações, a UCP ativa a unidade
- 9. O da UCP supervisiona ou controla as funções executadas por todo o sistema de computador.
- 10. As ligadas à UCP fornecem memória adicional para processar dados.

Respostas

(As respostas são apenas para as questões de números pares.)

A. Questões do Tipo Verdadeiro-Falso

2. F — cada sistema pode ser configurado de forma a incluir somente os dispositivos entrada/saída que o usuário necessita.
4. V
6. F — cada posição de memória é um *byte*; as instruções normalmente ocupam numerosas posições de memória.
8. V
10. V

B. Preencha as Lacunas:

2. memória de leitura somente (MLS)
4. *megabyte*
6. tempo real
8. lógica-aritmética
10. unidades de memória auxiliar ou secundária.

3

HARDWARE COMUM E OS MEIOS DE ENTRADA E SAÍDA

- I. DISPOSITIVOS COMUNS DE ENTRADA DE DADOS E MEIOS DE ENTRADA E SAÍDA
 - A. TERMINAIS
 - B. DISPOSITIVOS DE CARTÕES PERFURADOS
- II. PROCESSAMENTO DO ARQUIVO MESTRE USANDO FITA OU DISCO
 - A. CARACTERÍSTICAS DA FITA MAGNÉTICA E DAS UNIDADES DE FITA
 - 1. As Fitas só Podem ser Processadas em Seqüência
 - 2. Cada Processamento de Atualização Cria uma Nova Fita-Mestre
 - 3. Os Controles são Necessários para Manter a Integridade dos Arquivos de Fita
 - B. CARACTERÍSTICAS DOS DISCOS MAGNÉTICOS E DAS UNIDADES DE DISCOS
 - 1. Os Discos Podem ser Acessados Aleatória ou Seqüencialmente
 - 2. Os Arquivos de Disco Podem ser Facilmente Acessados e Modificados Conforme Necessário
- III. SAÍDA IMPRESSA
 - A. CARACTERÍSTICAS DA SAÍDA IMPRESSA
 - B. TIPOS DE IMPRESSORAS
 - C. CARACTERÍSTICAS DAS IMPRESSORAS
 - 1. Impacto *Versus* Não-Impacto
 - 2. Impressoras de Matriz de Ponto *Versus* Impressoras de Caractere Padronizado
- IV. DISPOSITIVOS ADICIONAIS DE E/S
 - A. DISPOSITIVOS DE RECONHECIMENTO ÓTICO DE CARACTERE (ROC)*
 - B. DISPOSITIVOS DE RECONHECIMENTO DE CARACTERE EM TINTA MAGNÉTICA (RCTM)**

TERMOS-CHAVE

QUESTIONÁRIO

* Em inglês OCR (Optical Character Recognition Devices); doravante usaremos a notação portuguesa.

** Em inglês MICR (Magnetic Ink Character Recognition Devices); doravante usaremos a notação portuguesa.

Conforme mencionado, os computadores lêem e processam os dados e produzem informação acerca deles. Há uma enorme variedade de dispositivos que podem ler os dados de entrada para um sistema de computador e, de forma semelhante, uma grande variedade de dispositivos que podem produzir dados de saída de um sistema de computador. Neste capítulo consideramos os dispositivos mais comuns e os meios de entrada/saída (E/S) correspondentes.

Observe que os dados são representados em muitas formas diferentes, dependendo do dispositivo usado. Portanto, além de considerarmos os dispositivos mais comuns de entrada e saída, levaremos em conta os conceitos de arquivo de dados referentes a estes dispositivos.

Entre os dispositivos mais comuns usados atualmente temos os seguintes:

Dispositivos Comuns de E/S

1. Terminais.
2. Dispositivos de cartão perfurado.
3. Unidades de fita magnética.
4. Unidades de disco magnético.
5. Impressoras.

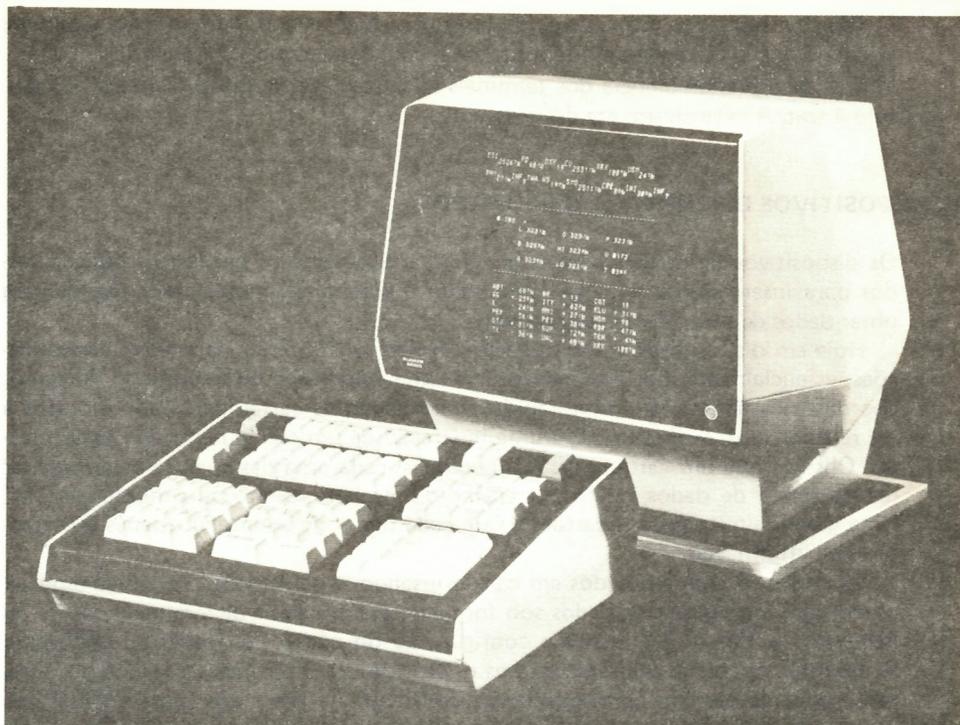
I. DISPOSITIVOS COMUNS DE ENTRADA DE DADOS E MEIOS DE ENTRADA E SAÍDA

A. TERMINAIS

Um *terminal* é qualquer dispositivo que pode acessar um computador de uma localização remota. Isto é, com o uso de um meio de comunicação, tal como uma linha telefônica, um terminal pode ser ligado a uma UCP localizada em qualquer parte do mundo.

Os terminais são freqüentemente usados para:

1. Entrada de dados — no ponto de transação.
Os dados podem ser inseridos no local da transação, isto é, em um magazine, aeroporto, banco etc.
2. Obter dados de saída ou respostas imediatas de um computador.
O computador pode fornecer respostas a perguntas ou produzir relatórios propriamente ditos no terminal.



(a)



(b)

Figura 3.1

(a) Tubo de raios catódicos que fornece aos operadores de mercado uma representação simultânea dos dados a partir de teleimpressoras, transmissão por cabo e bases de dados para mercado de ações. O sistema também pode ser usado para transmitir ordens de compra e venda. (Cortesia do Bunker Ramo Corp.) (b) Terminal de máquina de escrever. (Cortesia da IBM.)

Existem inúmeros dispositivos de terminais, mas os mais comuns são o tubo de raio catódico com teclado e o terminal de máquina de escrever (Figura 3.1).

Devido à predominância dos terminais nos sistemas de entrada de dados, o Capítulo 4 tratará exclusivamente deste assunto.

B. DISPOSITIVOS DE CARTÕES PERFURADOS

Os dispositivos de cartões perfurados eram antigamente os periféricos básicos usados para inserir dados em um computador e eram ainda amplamente usados para obter dados de saída de um computador.

Hoje em dia, os dispositivos de cartões foram substituídos, na maioria das aplicações, essencialmente por equipamentos de alta velocidade ou interativos. Portanto, abordaremos apenas de forma rápida o dispositivo de cartão perfurado e outros a ele relacionados. A Figura 3.2 é uma ilustração de um *cartão perfurado* padronizado. Observe que um cartão possui 80 colunas, cada uma das quais pode armazenar um caractere de dados, como por exemplo uma letra, dígito ou símbolo especial. Uma grande limitação dos cartões é que cada um pode armazenar apenas 80 caracteres de informação.

Os dados são representados em cartões usando-se perfurações. O código especial usado para representar os dados sob forma de perfurações é denominado *código de Hollerith*. A Figura 3.3 ilustra a configuração do código de Hollerith usado para representar caracteres. Observe que os dígitos de 0 a 9 são representados por uma perfuração na carreira correspondente, de 0 a 9. As letras são representadas por duas perfurações em uma coluna e os caracteres especiais podem ter até três perfurações.

Os cartões perfurados podem ser produzidos por *máquinas perfuradoras de teclado*, isto é, dispositivos manuais que funcionam como máquinas de escrever. Um operador aciona um caractere e os dados são perfurados automaticamente através da configuração do código de Hollerith (Figura 3.4).

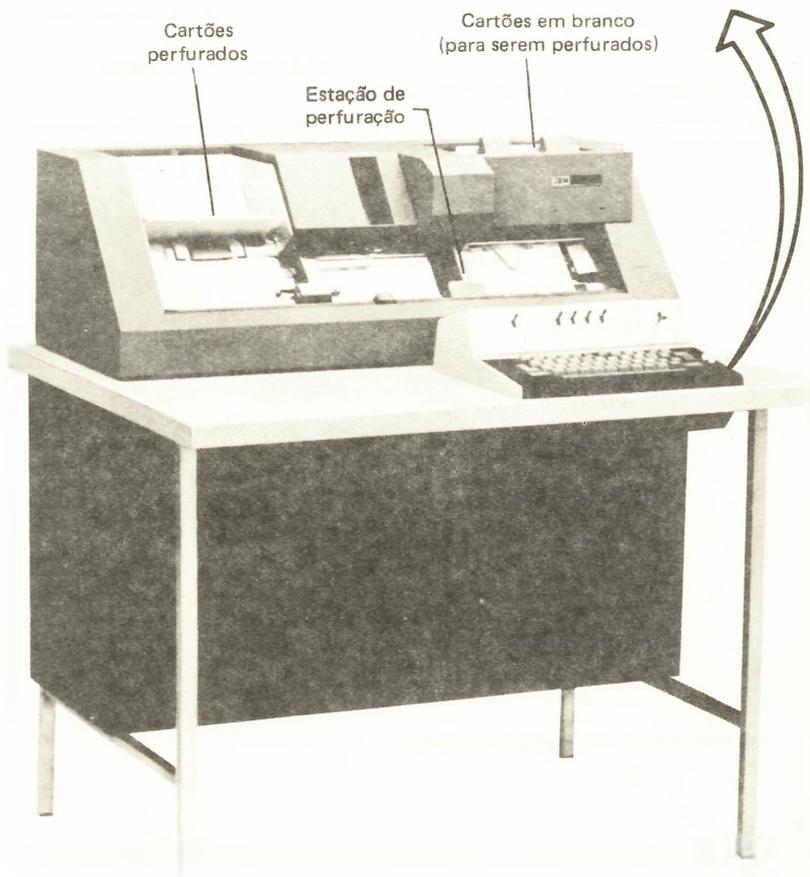
Os dados de um cartão perfurado podem ser lidos por um computador através de uma leitora de cartões. Algumas leitoras de cartões são dispositivos mecânicos que detectam as perfurações usando escovas elétricas. Outros usam células fotoelétricas para perscrutar os dados. Todas as leitoras de cartões são dispositivos relativamente lentos que nem de longe chegam a atingir a velocidade de outros equipamentos tais como unidades de fita e de disco. Esta velocidade lenta, associada à limitação de 80 colunas e ao fato dos cartões poderem ser dobrados, grampeados, mutilados ou inutilizados de algum outro modo, faz do processamento através de cartões algo ultrapassado, embora seja ainda bastante usado para muitas aplicações.

Os dispositivos de perfurar cartões são usados somente em um número relativamente pequeno de aplicações; produzem cartões perfurados como saída de um sistema de computador. Algumas vezes, entretanto, o cartão de saída é útil porque pode ser reinserido posteriormente como entrada. O cartão, neste caso, é denominado um *documento de retorno*. Contas de serviços públicos que são enviadas ao cliente e retornam depois com o pagamento são exemplos de cartões perfurados usados como documentos de retorno (Figura 3.5).

Até agora, falamos dos dispositivos de terminal e de cartões perfurados como equipamento comum de entrada/saída. Quando servem como entrada, tais equipa-

MULT PCH	# @	. %	\$ *	<		DUP -	0 /	PROG ONE	REL	PROG TWO		
	+ Q	- W) E	¢ R	0-8-2 T	I Y	1 U	2 I	3 O	& P	FEED	LEFT ZERO
ERROR RESET	A	> S	: D	: F	⌋ G	' H	4 J	5 K	6 L	SKIP	REG	
NUMERIC	Z	? X	" C	' V	=	! B	(N	7 M	8 ,	9 .	AUX DUP	ALPHA
SPACE BAR												

(a)



(b)

Figura 3.4
 (a) Teclado de uma máquina perfuradora. (b) Máquina perfuradora IBM029. (Cortesia da IBM.)

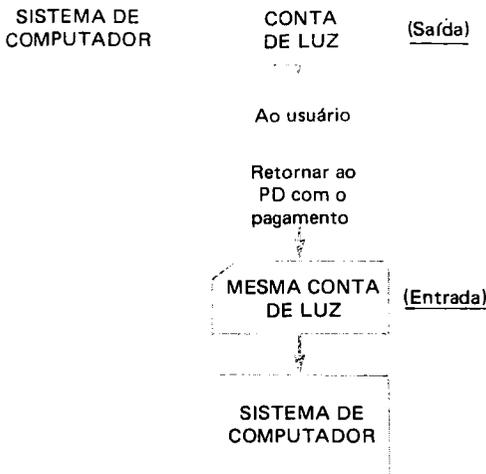
Light
Serviços de Eletricidade S.A.

Av. Marechal Floriano, 166 - Rio de Janeiro - RJ - C.C.N. N.º 60.444.437.0001-46

Consumidor	Nome do Cliente	Endereço	Relatório	Letra	Apresentação	Conta do
HELOISA PIRES MESQUITA	1753801753403001418092132601					JAN 63
M MASCARENHAS 50 F 206	Medidor	Consumo	Estado	Patente		
	610809563	1 3468	137	*****		
Q34	168027*****64130*****	232191		Vencimento		160285
<p>Utilize os serviços do Ligue-Light Tel. 283-2422 das 8h às 20h.</p>						
Total a pagar sem acréscimo Cr\$		2321,00	Acrescimo Cr\$	16800	Total a pagar com acréscimo Cr\$	2489,00
1983-MAIS PRODUCAO, MAIS EXPORTACAO,						

O recibo só será válido com autenticação mecânica ou carimbo do recebedor autorizado.

(a)



Atualizar a importância paga pelo usuário.

(b)

Figura 3.5

(a) Conta de luz em um cartão perfurado que serve como documento de retorno. (Cortesia da IBM.)
(b) O processamento dessa conta.

mentos são usados tipicamente para inserir dados como faturas, ordens de compra, dados de folha de pagamento etc. Estes dados são geralmente processados pelo computador junto com o *arquivo-mestre*. Um *arquivo-mestre* é a maior compilação de

registros referente a uma aplicação específica. Por exemplo, um arquivo-mestre de folha de pagamento é a principal fonte de informação sobre a remuneração dos empregados. Os arquivos-mestre são na maioria das vezes muito grandes e, portanto, são armazenados de maneira que possam ser lidos e escritos em altas velocidades.

Um objetivo básico da entrada de dados a partir de terminais ou cartões é *atualizar* ou colocar em dia um arquivo-mestre. Isto é, cartões de ponto podem ser inseridos semanalmente como entrada junto com um arquivo-mestre de folha de pagamento para atualizar o arquivo-mestre; ordens de compra podem ser inseridas em um terminal para atualizar o arquivo de contas a receber.

PROCESSAMENTO DO ARQUIVO-MESTRE USANDO FITA OU DISCO

A fita e o disco magnéticos são os meios mais comuns usados para armazenar arquivos-mestre, pelas seguintes razões:

Características do Processamento por Fita e Disco

1. Uma unidade de fita ou disco pode ler e escrever os dados.
2. Uma unidade de fita ou disco pode processar os dados em altas velocidades.
3. Uma fita ou um disco pode armazenar centenas de milhares de registros.
4. Um registro de fita ou disco pode ser de qualquer tamanho.

Existem várias distinções entre o processamento de fita e disco, porém a principal é a seguinte:

Principal Distinção Entre a Fita e o Disco

1. As fitas são processadas em seqüência, geralmente por lote.
2. Os discos podem ser processados de maneira aleatória ou direta, o que os torna extremamente flexíveis para o processamento em linha.

Portanto, se um terminal tem que acessar a informação de um arquivo-mestre de reserva de uma companhia aérea, tal arquivo será provavelmente armazenado em disco, da seguinte forma:

(Arquivo-mestre
de reserva da
companhia aérea)

(Terminal para
pergunta-resposta)

UCP

Unidade de
disco

Vamos agora considerar as principais características de ambos, fita e disco, e os dispositivos correspondentes, conforme se segue:

A. CARACTERÍSTICAS DA FITA MAGNÉTICA E DAS UNIDADES DE FITA

Uma *fita magnética* tem geralmente 720 m de comprimento, mas existem outros tamanhos. A maioria das fitas tem 1,27 cm de largura e são feitas de plástico, com uma camada de óxido de ferro que é magnetizada para representar os dados.

Por serem os pontos ou *bits* magnetizados extremamente pequenos e invisíveis a olho nu, é possível condensar grandes quantidades de dados em uma área de fita relativamente pequena. Por exemplo, os dados que são perfurados em um cartão de 80 colunas podem perfeitamente ser armazenados em 2,54 mm de fita magnética ou menos. É bom lembrar, entretanto, que as fitas podem ter registro de tamanho variado, e que os dados podem ser lidos ou escritos em velocidades que variam de 100.000 a 300.000 caracteres por segundo, ou aproximadamente 0,5 m de fita por segundo.

A fita média, que custa mais ou menos US\$20, pode armazenar mais de 20 milhões de caracteres. Depois que um arquivo de fita for processado e já não for necessário, a mesma fita pode ser reusada repetidamente para armazenar outras informações.

Uma unidade de fita magnética é um dispositivo que pode ler ou escrever em uma fita magnética (Figura 3.6). Os dados podem também ser registrados diretamente em uma fita, usando-se um dispositivo de impressão de fita que é semelhante a uma máquina perfuradora de teclado.

As fitas são usadas freqüentemente para arquivos-mestre volumosos, pelo fato de poderem ser processadas com muita rapidez e armazenarem grande quantidade de dados. Entretanto, possuem as seguintes limitações intrínsecas:

1. As Fitas Só Podem Ser Processadas em Seqüência

Para acessar um registro com um número de transação 254, por exemplo, de um arquivo de fita que se mantém na seqüência do número de transação, o computador tem que ler cada registro ordenadamente até que encontre o solicitado. Isto é chamado *Processamento Seqüencial*.

Se precisamos acessar e processar a maioria dos registros em um arquivo em seqüência, esta característica não será uma grande limitação. Por exemplo, se atualizarmos uma folha de pagamento lendo seqüencialmente o cartão de ponto de cada funcionário e adicionando os salários semanais ao arquivo-mestre, então o processamento seqüencial é adequado. Mas se atualizássemos um arquivo de contas a receber com compras que são inseridas a partir de um terminal sem nenhuma seqüência específica, o processamento seqüencial seria lento e ineficaz.

2. Cada Processamento de Atualização Cria uma Nova Fita-Mestre

As unidades de fita não podem ler uma fita-mestre e depois fazer modificações diretamente nesta mesma fita. Portanto, para atualizar uma fita-mestre, é necessário recriar todos os registros, com todas as modificações, em um novo arquivo-mestre de fita.

3. Os Controles são Necessários Para Manter a Integridade dos Arquivos de Fita

A maioria das instalações de computador em grande e média escala tem centenas ou até milhares de fitas magnéticas, cada uma utilizada para uma aplicação específica.

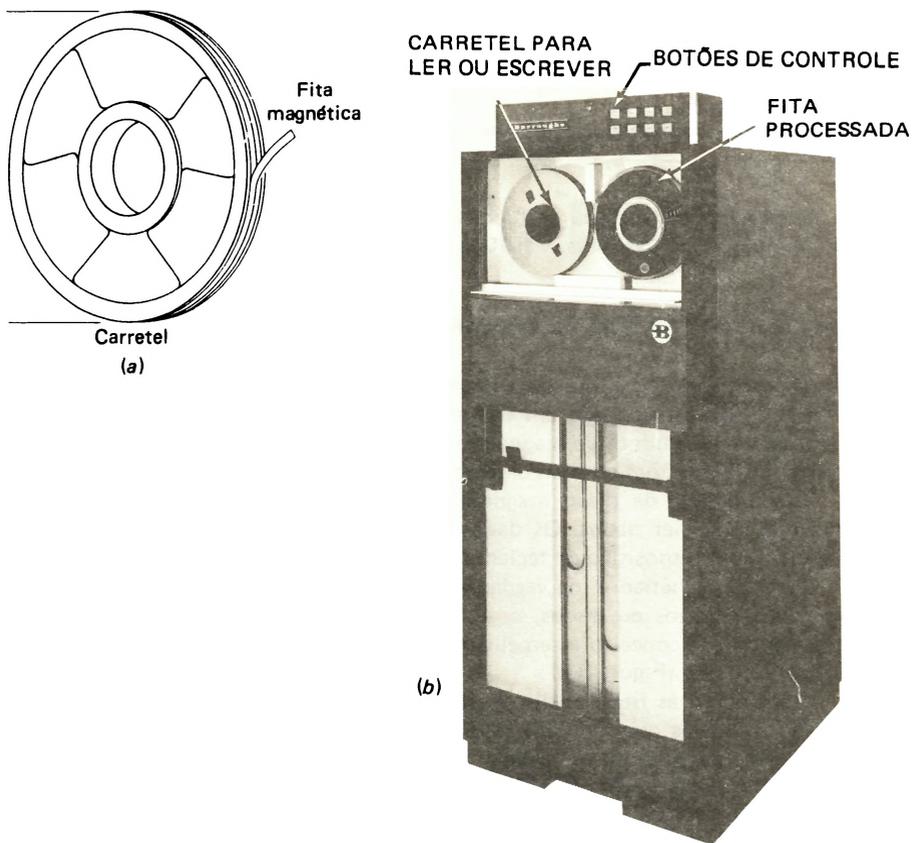


Figura 3.6
 (a) Carretel de fita magnética. (b) Unidade de fita magnética. (Cortesia da Burroughs Corp.)

É muitas vezes difícil manter o controle, pois os dados registrados nestas fitas não são "legíveis" ou visíveis a olho nu. Por exemplo, se uma fita-mestre de contas a receber for "reescrita" inadvertidamente, ou usada como saída para algum outro trabalho, o resultado poderia ser um processo de recriação caro, pois a escrita dos dados de saída destruiria a informação existente. Diversas medidas estão sendo adotadas na maioria das instalações para evitar tais ocorrências, ou reduzir a extensão dos prejuízos, caso eles venham a ocorrer. Veremos que estes mecanismos de controle são também usados para disco.

Mecanismos de Controle para Processamento de Fita e Disco

1. Rótulos gomados externos
 Identificam claramente a fita (ou disco).
2. Bibliotecário
 Controla o acesso a todas as fitas (e discos).

3. Rótulos programados
Os rótulos de identificação são programados para serem o primeiro e o último registro na fita (ou disco).
4. Arquivos de reserva
São guardados caso a fita (ou disco) seja destruída acidentalmente.

As fitas em cassetes ou cartuchos são fitas magnéticas em miniatura, usadas com mini e microcomputadores. Veja Capítulo 5.

B. CARACTERÍSTICAS DOS DISCOS MAGNÉTICOS E DAS UNIDADES DE DISCO

Um *disco magnético* é outro meio de informação de alta velocidade que pode servir como entrada ou saída de um sistema de computador (Figura 3.7). Da mesma maneira que a fita, um disco magnético tem uma capa de óxido de ferro usada para armazenar 100 milhões ou mais caracteres de dados.

Uma unidade de disco magnético, como uma unidade de fita, pode escrever informação ou ler dados. Os dados podem ser também registrados em um disco usando-se um dispositivo de teclado para discos.

O disco magnético é, na verdade, um conjunto de discos que consiste de uma seqüência de pratos ou discos, organizados em uma pilha vertical e ligados por um eixo central. O conceito é semelhante a um grupo de discos fonográficos empilhados em um eixo (Figura 3.8).

Assim como as fitas, os discos podem ser processados com muita rapidez, armazenando grandes quantidades de dados e tornando-se bastante apropriados para o processamento do arquivo-mestre. Os dados têm os mesmos problemas de controle que as fitas, e empregam-se mecanismos semelhantes para evitar que estes sejam acidentalmente usados de maneira errada.

Entretanto, o disco possui várias vantagens intrínsecas que o tornam mais adequado para algumas aplicações. Consideraremos algumas destas vantagens a seguir:

1. Os Discos Podem Ser Acessados Aleatória ou Seqüencialmente

Visto que um disco possui inúmeras superfícies de registro endereçáveis e cabeças de leitura e escrita, os registros podem ser acessados tanto seqüencial como aleatoriamente, sem necessidade de pesquisar todo o arquivo.

Os dois métodos de organização que possibilitam o *acesso aleatório* são os seguintes:

a. Organização Seqüencial Indexada de Arquivo

A *organização seqüencial indexada de arquivo* é um método comum para acessar os registros de disco magnético diretamente (ou de forma aleatória) através de um índice. Durante a criação dos registros, o computador usa rotinas programadas de manuseio de arquivo para organizar um índice no próprio disco. Este índice indica basicamente a localização de cada registro. O conceito é semelhante ao do índice encontrado no final do livro, que especifica a página onde cada item de informação pode estar localizada.

O índice do disco especifica os endereços ou localizações dos registros que estão armazenados no disco. Em termos básicos, o endereço refere-se à área física no dis-



(a)



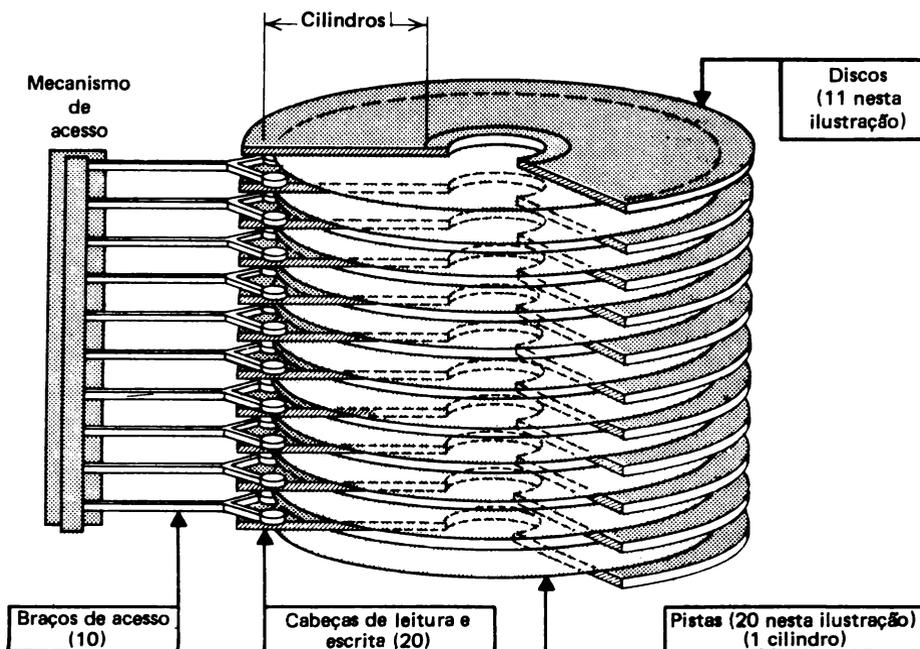
(b)

Figura 3.7

(a) Unidade de disco. (Cortesia da NCR.) (b) Conjunto de discos. (Cortesia da Burroughs Corp.)

Figura 3.8

Como os dados são acessados de um conjunto de discos. Cada cabeça de leitura/escrita acessa uma superfície específica. As cabeças de leitura e escrita se movimentam juntas para dentro e para fora em função do mecanismo de acesso.



co onde se encontra um determinado registro. O computador usa um campo de dados codificados em cada registro, conforme indicado pelo programador, como base para organizar a informação de endereço no índice. Por exemplo, se um arquivo de folha de pagamento é armazenado em um disco, o campo codificado seria provavelmente o Número do Seguro Social* ou o Número do Empregado, se este for usado como meio para identificar os registros.

Portanto, para acessar qualquer registro de disco, o usuário precisa somente fornecer um campo de dados codificados, tal como o Número do Empregado 17537; o computador "procura" no índice o endereço do disco correspondente para este registro e vai diretamente em busca desse registro.

b. Outros Métodos de Organização de Arquivo

Os registros podem ser armazenados em localizações relativas no arquivo. O campo codificado é usado para determinar cada localização relativa. No caso mais simples, o campo codificado faria a conversão diretamente para a posição real no disco. Isto é, em um arquivo de folha de pagamento com números de empregados de 0001 a 1999, o registro com um número de empregado de 0525 seria realmente o 525º registro no disco. Isto é denominado *organização relativa de arquivo*.

Na maioria das vezes, a conversão do campo codificado para a localização real não é tão simples. Frequentemente torna-se necessário converter um campo codificado, através de algum cálculo aritmético, em um endereço real que identifique a localização do registro. Por exemplo, em um campo codificado relativo a um número de conta de três dígitos, o primeiro dígito pode fornecer o número de superfície, e os dois últimos dígitos o número de pista. Este esquema usaria 10 superfícies e 100 pistas, e é mais rápido que o acesso seqüencial indexado porque não é necessário procurar um endereço em um índice. Entretanto, necessita de uma programação mais ampla, pois o programador tem que determinar a melhor fórmula matemática para converter os campos codificados em endereços. Isto é denominado *organização direta de arquivo*.

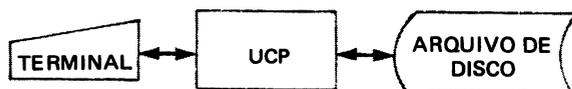
Visto que os arquivos de disco podem ser processados diretamente, conforme os dois exemplos acima, sem necessidade de um índice, as organizações diretas e relativas de arquivo são ideais para o sofisticado processamento em tempo real.

2. Os Arquivos de Disco Podem ser Facilmente Acessados e Modificados Conforme Necessário

Os discos têm sobre as fitas a vantagem de permitirem atualizações ou modificações dos registros existentes no mesmo disco. Desta maneira, não é necessário criar um disco novo para incluir as modificações em curso, como em geral é necessário fazer com o processamento por fita. Isto é, o mesmo disco pode ser usado tanto para entrada como para saída. Pode-se ler um registro de um disco e modificá-lo no mesmo disco; podem-se acrescentar registros ao disco; e apagar registros do disco. A Figura 3.9 ilustra uma operação de atualização. Entretanto, observe que é necessário criar arquivos de reserva para o disco no caso deste ser avariado.

* Em inglês Social Security Number; equivale ao nosso número de registro na Previdência Social.

a. Pedido de informação sobre um voo.



b. Emissão de uma passagem aérea e conseqüente atualização do disco.

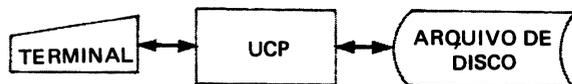


Figura 3.9

Solicitação de dados e atualização de um arquivo-mestre de disco.

Os *discos flexíveis* são discos em miniatura usados com mini e microcomputadores. Veja o Capítulo 5.

As Tabelas 3.1 e 3.2 apresentam sumários do processamento de arquivos de cartões, discos e fitas.

II. SAÍDA IMPRESSA

A. CARACTERÍSTICAS DA SAÍDA IMPRESSA

O relatório impresso é uma das principais formas de saída do computador. É a forma de saída mais comumente distribuída aos usuários.

O documento impresso é em geral usado basicamente como um produto final. É o resultado final de um programa rodado em computador e geralmente é revisado pela gerência de alto nível. É preparado por um dispositivo de computador denominado impressora (Figura 3.10).

Outras formas de saída são geralmente produtos intermediários, cuja função primordial é a de serem reinseridas no fluxo do computador como entrada para outro trabalho. Estes tipos de saída são planejados para serem tão eficientes quanto possível. Por exemplo, os campos e registro deste tipo são condensados de forma a permitir a máxima utilização do computador e de suas capacidades de armazenamento. Entretanto, o relatório impresso é escrito levando-se em conta a sua utilização pelo usuário. Visto que muitos relatórios produzidos pelo computador são lidos por diretores de companhias, tais formulários têm que ser nítidos, limpos e fáceis de interpretar. Os usuários devem lembrar-se sempre de que os relatórios de saída têm que fornecer de maneira precisa a informação solicitada, naturalmente, dentro das limitações de custo.

As impressoras podem imprimir tipicamente 100, 120 ou 132 caracteres por linha, dependendo do dispositivo.

O papel fornecido à impressora é denominado um *formulário contínuo* pois é emendado, sendo separado somente por perfurações (Figura 3.11).

Os relatórios impressos podem ser em papel normal padrão, pautado ou não, ou em *formulários pré-impressos* (Figura 3.12). Pode-se utilizar papel em várias vias, com carbonos, ou um papel especial sem carbono para fornecer cópias.

Tabela 3.1 Características Físicas dos Meios de Arquivo Usuais

<i>Características</i>	<i>Cartão perfurado</i>	<i>Fita Magnética</i>	<i>Disco Magnético</i>
Comprimento do registro	80 [ou 96] caracteres	Ilimitado	Ilimitado
Capacidade total	80 caracteres por cartão	1 a 45 milhões de caracteres	1 a 100 milhões de caracteres
Método de acesso	Sequencial	Sequencial	Direto ou sequencial
Equipamento usado para criar o arquivo	Perfuradora de teclado ou perfuradora de cartão.	Sistema de impressão de fita ou unidade de fita	Sistema de acionamento de disco ou unidade de disco.
Reusável	Não	Sim	Sim
Legibilidade manual	Não	Não	Não
Velocidade de transferência (para a UCP)	Sim 300 + caracteres por segundos	15.000 a 320.000 caracteres p/seg.	100.000 a 1.5 milhão de caracteres por segundo.

Tabela 3.2 Sumário dos Meios de Arquivo Usuais

Tipo de Arquivo	Características	Vantagens	Desvantagens
Cartão perfurado	<ol style="list-style-type: none"> 1. Cartão de 80 colunas 2. Dados registrados nos cartões pela máquina perfuradora de teclado ou perfuradora de cartão de um computador. 	<ol style="list-style-type: none"> 1. Pode ser processado por máquina elétrica de contabilidade, bem como por computadores. 2. Relativamente barato para ser- viços de pouco volume. 3. Legível maquinal e manual- mente 	<ol style="list-style-type: none"> 1. Velocidade lenta dos dispositi- vos de cartão. 2. Os cartões deformam, dobram ou rasgam facilmente. 3. Ineficaz para arquivos de gran- de volume.
Fita magnética	<ol style="list-style-type: none"> 1. Dados registrados sob forma de <i>bits</i> magnetizados em capa de óxido de ferro. 2. Dados registrados por disposi- tivos de impressão ou unidade de fita. 	<ol style="list-style-type: none"> 1. Muito eficaz para trabalhos de grande volume em um ambien- te de processamento por lote. 2. Pode armazenar milhões de ca- racteres em um carretel. 3. Tamanho de registro irrestrito. 	<ol style="list-style-type: none"> 1. Problemas de controle porque os dados não são legíveis ma- nualmente. 2. Usado principalmente para pro- cessamento sequencial. 3. Não se pode adicionar dados a uma fita existente.
Disco magnético	<ol style="list-style-type: none"> 1. Dados registrados sob forma de <i>bits</i> magnetizados em capa de óxido de ferro. 2. Dados registrados por disposi- tivos de impressão ou unidade de disco. 	<ol style="list-style-type: none"> 1. Muito eficaz para trabalhos de grande volume em um ambien- te em linha. 2. Pode armazenar milhões de ca- racteres em um conjunto de discos. 3. Podem-se adicionar dados a um disco existente. 4. Tamanho de registro irrestrito. 	<ol style="list-style-type: none"> 1. Discos de reserva devem ser criados para controle. 2. Os choques de cabeças de lei- tura/escrita podem destruir os arquivos de disco.



Figura 3.10
Uma impressora. (Cortesia da IBM.)

As características da saída impressa podem ser resumidas conforme se segue:

RESUMO

1. Os programas que produzem saídas impressas devem incluir instruções para:
 - a. Impressão de cabeças.
 - b. Alinhamento de dados.
 - c. Edição.
 - d. Espacejamento de formulários.
2. Dependendo da impressora, pode-se obter um máximo de 100, 120 ou 132 caracteres por linha.
3. Os formulários contínuos são usados para saídas impressas de computador.
4. Tipos de formulários impressos:
 - a. (1) Pré-impressos — usados para extrato de conta de comprador, cheques de pagamento e relatórios externos distribuídos fora da organização.
 - (2) Formulários padronizados — usados para relatórios internos.
 - b. O papel em várias vias é geralmente encontrável em duas formas:
 - (1) Papel com folhas de carbono — os carbonos devem ser retirados.
 - (2) Papel sem carbono — mais caros, porém eliminam a necessidade de retirar os carbonos.

B. TIPOS DE IMPRESSORAS

Há uma grande variedade de dispositivos de impressoras disponíveis. Alguns são dispositivos de alta velocidade; outros são usados predominantemente como dispositivos de terminal para saídas de pouco volume, cópias de leitura direta ou saídas impressas. A lista abaixo indica os principais tipos de impressoras usadas atualmente.

Tipos de Impressoras

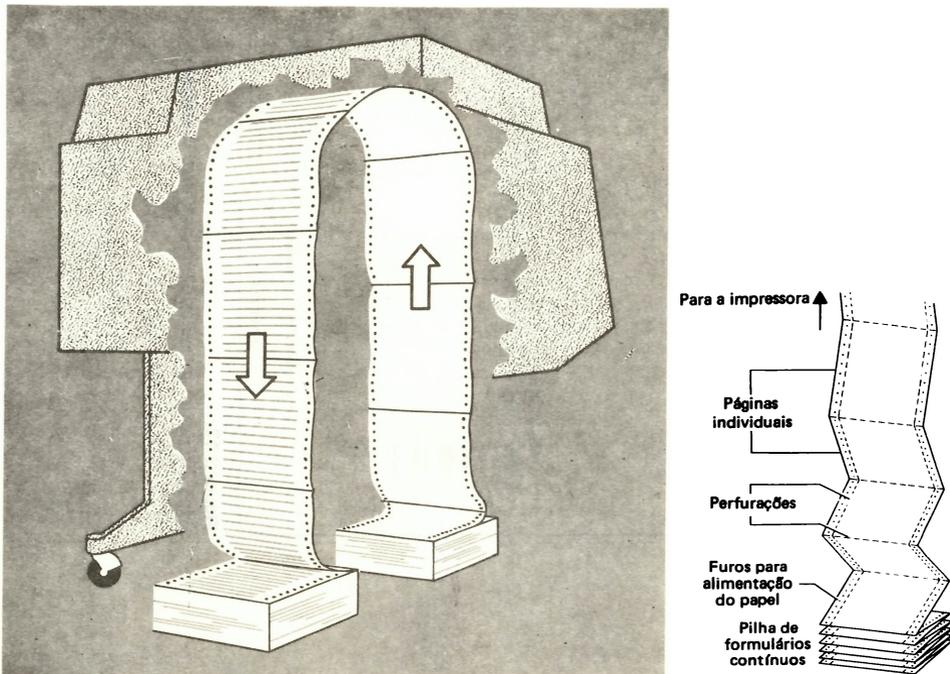
1. Impressora em série
 - Dispositivo relativamente lento.
 - Funciona exatamente como uma máquina de escrever, imprimindo um caractere de cada vez.
 - Usada basicamente para terminais interativos de impressão de teclados (a serem discutidos na próxima seção) ou minicomputadores.
 - Relativamente barata.
2. Impressora por linha
 - Dispositivo de rapidez moderada.
 - Acumula uma linha completa de saída e depois imprime uma linha de cada vez.
 - Esta é a impressora mais comum para operações de impressão com velocidade moderada.
3. Impressora por página
 - Dispositivo extremamente rápido e caro.
 - Imprime uma página inteira de uma só vez.

C. CARACTERÍSTICAS DAS IMPRESSORAS

1. Impacto *Versus* Não-Impacto

Uma impressora de impacto é aquela que opera geralmente usando um martelo para bater um caractere contra uma fita de impressão; o impacto, então, faz com que a imagem do caractere seja impressa.

Figura 3.11
Formulários contínuos. (Cortesia da IBM.)



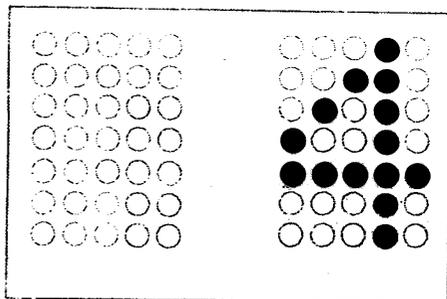
As impressoras de impacto funcionam exatamente como as máquinas de escrever. As principais desvantagens são o fato de serem relativamente lentas, barulhentas e sujeitas a falhas mecânicas. Apesar destas desvantagens, a maioria das impressoras usadas atualmente são dispositivos de impacto.

Existem, no entanto, impressoras sem impacto, que usam métodos eletrônicos ou fotoelétricos para produzir relatórios em altas velocidades. Os exemplos incluem os dispositivos eletrossensíveis, eletrostáticos e laser-xerográficos.

2. Impressoras de Matriz de Ponto Versus Impressoras de Caractere Padronizado

Algumas impressoras imprimem imagens de caractere a partir de uma seqüência de pontos. A configuração de cada caractere é criada por fios estendidos que atingem uma fita de carbono, produzindo a imagem desejada (Figura 3.13).

As impressoras de caractere padronizado usam um disco, cadeia ou tambor de impressão que posiciona o caractere desejado no ponto exato e geralmente usam um martelo para bater a imagem do caractere desejado no papel.



Matriz de ponto 5 X 7

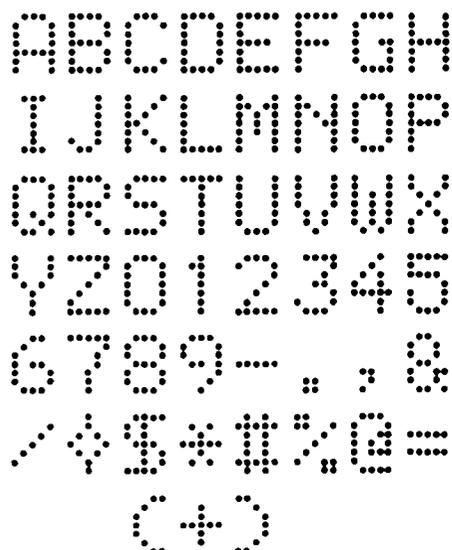
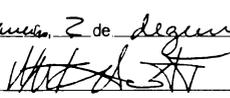
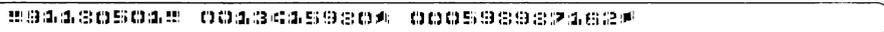
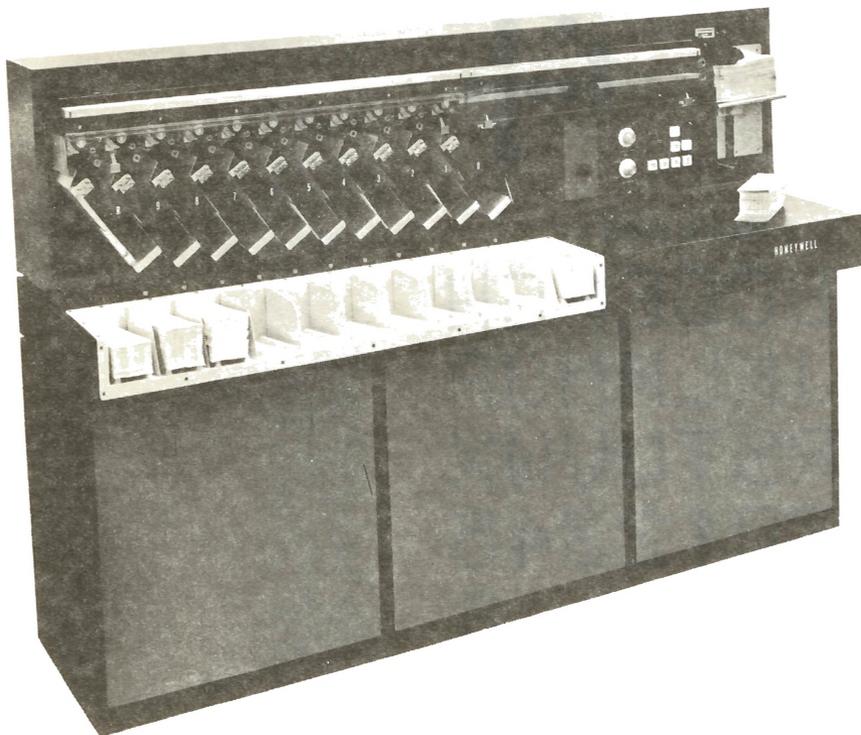


Figura 3.13
Impressora de matriz de ponto.

Identificação do banco	Número da conta	Número do cheque	Identificação do banco	Valor
0598	98716-2	Cheque N.º CB-180501	Banco 341	Cr\$ 50.000,00
Pague por este cheque a quantia de <u>cinquenta mil reais</u>				
A <u>José Roberto Rego Cadato</u> ou à s/ordem				
<u>Rio de Janeiro, 2 de dezembro de 1983</u> 				
				

(a)



(b)

Figura 3.14

(a) Cheque. (b) Leitora de Classificação de RCTM. (Cortesia de Honeywell.)

Algumas impressoras de impacto com caractere padronizado são conhecidas como *Impressoras de letra de qualidade* porque produzem imagens nítidas no papel.

. DISPOSITIVOS ADICIONAIS DE E/S

Os terminais considerados no Capítulo 4 e os dispositivos discutidos aqui estão entre os dispositivos periféricos usados com mais frequência. Entretanto, não são os únicos disponíveis.

Dentre os dispositivos projetados de forma a minimizar a necessidade de inserir dados por teclado em um computador estão incluídos os seguintes:

1. DISPOSITIVOS DE RECONHECIMENTO ÓTICO DE CARACTERE (ROC)

Estes dispositivos de entrada podem de fato ler a entrada impressa ou manuscrita. Porém, são caros e sujeitos a erros causados por rasuras, erros de tipografia, alinhamento equivocado dos dados etc.

3. DISPOSITIVOS DE RECONHECIMENTO DE CARACTERE EM TINTA MAGNÉTICA (RCTM)

A indústria bancária, que processa mais de 50 bilhões de cheques por ano, usa os dispositivos de RCTM para ler e classificar os cheques processados. Os cheques são codificados na parte inferior com uma tinta magnética especial que pode ser lida por estes dispositivos (Figura 3.14).

Observe que existe uma grande variedade de outros meios usados para funções especializadas de entrada e saída que não foram discutidos aqui. Entretanto, consideramos os dispositivos e meios mais comuns usados em pequena, média e larga escala nos centros de computador.

— TERMOS-CHAVE —

Acesso aleatório
Arquivo
Arquivo de reserva
Arquivo-mestre
Atualizar
Cartão perfurado
Código de Hollerith
Disco flexível
Disco magnético
Dispositivo de Reconhecimento de Caractere em Tinta Magnética (RCTM)
Dispositivo de Reconhecimento Ótico de Caractere (ROC)
Documento de retorno
Fita magnética
Formulário contínuo
Formulário pré-impresso
Impressora de impacto

Impressora de letra de qualidade
 Impressora de matriz de ponto
 Impressora por caracteres
 Impressora sem impacto
 Máquina perfuradora de teclado
 Organização direta de arquivo
 Organização relativa de arquivo
 Organização seqüencial indexada de arquivo
 Processamento seqüencial
 Terminal

QUESTIONÁRIO

(As respostas para as questões de números pares acompanham o teste.)

A. Questões do Tipo Verdadeiro-Falso

- | V | F | |
|-------|-------|---|
| _____ | _____ | 1. Um cartão perfurado é um formulário interativo comum de entrada e saída. |
| _____ | _____ | 2. Os cartões perfurados são a forma predominante de entrada e saída. |
| _____ | _____ | 3. O tipo mais rápido de impressora é uma impressora por página. |
| _____ | _____ | 4. Os tubos de raios catódicos são usados como terminais. |
| _____ | _____ | 5. Se fosse necessário obter informações sobre o estado do arquivo em um ambiente em linha, o arquivo seria provavelmente armazenado em fita. |
| _____ | _____ | 6. Os processamentos de fita e disco são usados para a maioria dos arquivos de grande volume. |
| _____ | _____ | 7. As fitas são mais apropriadas para processamento em tempo real. |
| _____ | _____ | 8. Os dados são representados em fita e disco sob a forma de <i>bits</i> magnetizados. |
| _____ | _____ | 9. Um arquivo de disco pode armazenar tipicamente milhões de caracteres de informação. |
| _____ | _____ | 10. É possível apagar ou "reescrever" uma fita. |

B. Preencha as lacunas

1. Os discos são mais apropriados para processamento de arquivo-mestre em um ambiente
2. Se um terminal tivesse que acessar a informação de um arquivo-mestre, este arquivo seria geralmente armazenado em
3. Visto que as fitas podem ser às vezes reescritas ou podem mesmo ser danificadas, é necessário manter fitas para minimizar o risco de um processo de recriação caro.
4. Os minicomputadores usam que são versões menores de unidades de fita magnética de grande escala.
5. Os minicomputadores usam dispositivos que são exatamente análogos às unidades de disco magnético de grande escala.
6. Uma fita magnética é feita de plástico com uma camada de óxido de ferro que pode ser para representar os dados.
7. Os bancos usam para ler os dados codificados com tinta magnética especial.
8. Um dispositivo que pode ler um documento datilografado é denominado um
9. Os relatórios impressos que contêm o nome de uma companhia usam formulários especiais.
10. O código usado para representar dados em um cartão perfurado é denominado

Respostas

(Fornecemos respostas apenas para as questões de números pares.)

A. Verdadeiro - Falso

2. F 4. V 6. V 8. V 10. V

B. Preencha as lacunas

2. Discos
4. Gravadores cassete
6. Magnetizada
8. Dispositivo de reconhecimento ótico de caracteres
10. Código de Hollerith

4

CARACTERÍSTICAS E APLICAÇÕES DAS COMUNICAÇÕES DE D

- I. INTRODUÇÃO
- II. EQUIPAMENTOS DE COMUNICAÇÃO DE DADOS
 - A. TERMINAIS MAIS USADOS
 - 1. Tubo de Raios Catódicos (TRC) com Teclado
 - 2. Terminal de Máquina de Escrever
 - 3. Unidade de Resposta Audível
 - 4. Dispositivo de Reconhecimento de Voz
 - 5. Outros Tipos de Terminais
 - B. ELO DE COMUNICAÇÃO
 - 1. Linhas de Comunicação
 - 2. Interface
 - C. REQUISITOS DO COMPUTADOR
- III. PRINCIPAIS APLICAÇÕES DO TERMINAL E DA COMUNICAÇÃO DE DADOS
 - A. SISTEMAS DE PONTO DE VENDA (SPV)*
 - B. SISTEMAS ELETRÔNICOS DE TRANSFERÊNCIA DE FUNDOS (TEF)**
 - C. PROCESSAMENTO DE PALAVRAS E AUTOMATIZAÇÃO ADMINISTRATIV
 - 1. Sistemas de Processamento de Palavras
 - 2. Correio Eletrônico

TERMOS-CHAVE

QUESTIONÁRIO

* Em inglês POS (Point-of-Sale Systems).

** Em inglês EFT (Electronic Funds Transfer Systems).

I. INTRODUÇÃO

Comunicação de dados é o termo usado para descrever a transmissão eletrônica de dados por linhas telefônicas ou telegráficas, satélites ou estações de microondas. Os *terminais* em uma localização remota fora do centro de computação são ligados à UCP através de uma linha de comunicação. O *teleprocessamento* é o uso combinado dos equipamentos de comunicação, tais como linhas telefônicas, e equipamento de computador para transmitir dados.

Nos sistemas de comunicação de dados, um terminal localizado praticamente em qualquer lugar do mundo pode inserir dados ou obter respostas de qualquer UCP.

A maioria dos sistemas de comunicação de dados é usada para os seguintes tipos de aplicações:

Aplicações da Comunicação de Dados

1. Entrada de Dados a Distância

Usa-se um terminal no local onde os dados são produzidos, de maneira que estes possam ser inseridos diretamente na UCP.

Exemplo: as ordens de compra são enviadas diretamente para a UCP através de um terminal.

2. Pergunta-Resposta

Usa-se um terminal para se fazer uma consulta sobre o estado de um arquivo.

Exemplo: o serviço ao consumidor em uma loja possui terminais que dão acesso ao arquivo de contas a receber para responder às consultas dos clientes.

3. Tempo Compartilhado

Usa-se um terminal para inserir dados e programas em uma UCP que é compartilhada por vários usuários ou programadores.

Exemplo: os estudantes de uma universidade podem inserir os seus programas BASIC em um terminal que tem acesso a um grande equipamento de tempo compartilhado.

Observe que muitas aplicações usam os sistemas de comunicação de dados tanto para entrada de dados a distância como para perguntas-respostas. Por exemplo, um sistema de reservas de uma companhia aérea utiliza a comunicação de dados para inserir informações sobre passagens e para indagar acerca das situações de determinados vôos.

II. EQUIPAMENTOS DE COMUNICAÇÃO DE DADOS

Os três elementos principais do sistema de comunicação de dados são os seguintes:

Elementos dos Sistemas de Comunicação de Dados

1. Dispositivos de terminal.
2. Linhas de comunicação.
3. Sistema de computador.

Discutiremos detalhadamente cada um dos elementos mencionados acima.

A. TERMINAIS MAIS USADOS

1. Tubo de Raios Catódicos (TRC) com Teclado

O *Tubo de Raios Catódicos* (TRC) (Figura 4.1) é um dispositivo de exibição visual semelhante a uma tela de televisão. A saída é exibida instantaneamente na tela. A saída de exibição visual pode ser digitada ou gerada em forma gráfica.

Alguns TRC podem exibir os dados em cores. Por não haver nenhum registro impresso, a saída é considerada como uma *soft copy*.

Um teclado é geralmente ligado ao TRC para transmitir dados à unidade central de processamento. Uma *caneta luminosa* ou *bastão de comando* podem também ser usados para fazer modificações na tela. Veja novamente a Figura 4.1 exemplificando uma caneta luminosa usada juntamente com um TRC.

Os TRC são usados com mais freqüência quando se deseja obter rapidamente os dados de saída em localizações remotas e onde não se exigem *hard copy* ou saídas

Figura 4.1

Terminal de raios catódicos com teclado. O operador pode fazer modificações nos dados exibidos usando uma caneta luminosa ou teclado. (Cortesia da IBM.)





Figura 4.2
Terminal de máquina de escrever (Cortesia da IBM.)

impressas. Por exemplo, as firmas distribuidoras de ações usam os TRC com teclado para obter informações sobre as últimas cotações do mercado de ações.

2. Terminal de Máquina de Escrever

Com um *terminal de máquina de escrever* (Figura 4.2) um operador envia os dados para a UCP através de um teclado, e a saída de *hard copy* é produzida através de um dispositivo semelhante a uma máquina de escrever.

Os terminais de máquina de escrever são usados para inserir encomendas de compra, recibos de mercadoria, passagens aéreas etc. O terminal possibilita que:

1. Os dados sejam transmitidos à UCP.
2. Os usuários guardem a encomenda, recibo ou passagem em forma de *hard copy*.

De maneira semelhante, os terminais de máquina de escrever são usados para obter relatórios de *hard copy* ou respostas da UCP.

3. Unidade de Resposta Audível

Uma *unidade de resposta audível* é um terminal que pode fornecer aos usuários uma resposta verbal. O computador é equipado com frases pré-gravadas que são extraídas de um arquivo quando forem necessárias para responder a uma pergunta específica. As mensagens verbais são transmitidas ao usuário por meio de uma unidade de resposta audível.

Muitos bancos usam unidades de resposta audível. Suponha que um cliente quer descontar um cheque em uma agência filial. O caixa registra o número da conta do cliente e a importância do cheque usando um fone com discagem por multifrequência. A UCP "examina" a conta do cliente e em seguida uma unidade de resposta audível instrui o caixa para descontar o cheque ou recusá-lo delicadamente (Figura 4.3). De maneira semelhante, os serviços de "pagamento por telefone" oferecidos



Figura 4.3

Telefone com discagem por multifreqüência usado em sistema de teleprocessamento. (Cortesia de Bell System Data Communications Service.)

por alguns bancos têm unidades de resposta audível para responder às solicitações do cliente.

4. Dispositivo de Reconhecimento de Voz

O *equipamento de reconhecimento de voz* pode interpretar a linguagem oral e transmitir os dados à UCP. O usuário discar para o computador usando um fone com discagem por multifreqüência como terminal e fala com o dispositivo de reconhecimento de voz. Isto tem uma grande utilidade potencial para entrada de ordens de venda, controle de estoque e muitas outras aplicações (Figura 4.4).

5. Outros Tipos de Terminais

A relação acima descreve alguns dos principais terminais usados atualmente. Observe que existe uma enorme variedade de outros dispositivos que servem tipicamente como terminais ou que podem ser modificados para serem usados como terminais, incluindo os seguintes:

a. Terminais inteligentes

Terminais padronizados com capacidade incorporada de microprocessamento para edição, formatação e alguns processamentos limitados.

b. Leitoras de código de barra

Usadas em bancos, supermercados e outros estabelecimentos varejistas (veja a seção III deste capítulo).



Figura 4.4

Verificação de fonte/entrada — o pessoal do almoxarifado insere os dados de controle do estoque diretamente no computador principal e verifica a precisão nesse mesmo momento. (Cortesia da Interstate Electronics Corp.)

c. Terminais manuais

Utilizados para entrada de encomendas e uso do consumidor (veja a Figura 4.5).

- d. Qualquer dispositivo de E/S que não esteja no mesmo local que a UCP pode servir como terminal.

Figura 4.5

Exemplo de um terminal manual. (Cortesia da Norand Corp.)



B. ELO DE COMUNICAÇÃO

Um terminal que se comunicará com uma UCP necessita dos seguintes elementos:

Requisitos para a Comunicação	
1. Linhas de Comunicação	Conectam o terminal à UCP
2. Interface	Possibilita a comunicação da UCP com o terminal.

1. Linhas de Comunicação

Dentre os tipos de *linhas de comunicação* normalmente disponíveis para teleprocessamento estão os seguintes:

Linhas de Comunicação			
<i>Tipo</i>	<i>Exemplo</i>	<i>Característica</i>	<i>Velocidade de Transmissão (bps = bits por segundo)</i>
Estreita (velocidade lenta)	Linhas de teletipo	Mais barata, mais lenta	Até 150 bps
Classe vocal (alcance médio)	Linhas de telefone	Mais comum; o ruído na linha interfere na transmissão	Algumas atingem 9600 bps
Larga (de banda larga)	Linhas de telefone particulares ou alugadas	Cara, particular, menos ruído	Até 56.000 bps

A velocidade da linha de comunicação é algumas vezes denominada *taxa de bauds*.

A maioria dos sistemas utiliza um terminal e uma linha telefônica normal para transmitir dados a distância para uma UCP. Algumas vezes, quando um terminal não está muito longe da UCP, ele pode ser ligado por um cabo ou *conectado por fio* à unidade principal.

2. Interface

Existem 2 tipos de *interfaces* (Figura 4.6) que permitem a comunicação entre o terminal e a UCP.

Interfaces	
<i>Tipo</i>	<i>Descrição</i>
Modulador/ Demodulador	Ligado por fio. Possibilita uma conexão permanente entre o terminal e a UCP.
Acoplador acústico	Usado para transmitir dados através das linhas de comunicação.

(a)



(b)

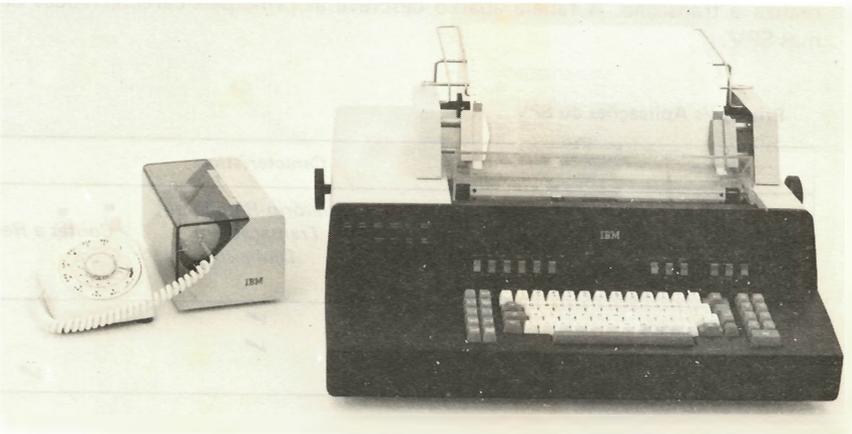


Figura 4.6

(a) Um modulador-demodulador. (Cortesia da Burroughs Corp.) (b) Terminal com um acoplador acústico. (Cortesia da IBM.)

C. REQUISITOS DO COMPUTADOR

Praticamente qualquer computador pode ser usado para a comunicação de dados. Entretanto, para usuários múltiplos em localizações remotas, a UCP deve ter os seguintes elementos:

1. Capacidade de acesso direto.
2. Capacidade de multiprogramação — capacidade de processar mais de um programa ou solicitação ao mesmo tempo.
3. Características complexas de controle.

Os minicomputadores, bem como sistemas maiores, são comumente usados para a comunicação de dados.

Observe que a manutenção da segurança do sistema de comunicação de dados é um grande problema, pois terminais remotos em várias localizações podem ter acesso a uma UCP. Os analistas de sistemas devem ter muito cuidado para planejar os sistemas de comunicação de dados de modo a assegurar a sua integridade e restringir o uso não-autorizado do computador.

III. PRINCIPAIS APLICAÇÕES DO TERMINAL E DA COMUNICAÇÃO DE DADOS

A. SISTEMAS DE PONTO DE VENDA (SPV)

Uma aplicação comum, orientada para o consumidor, do equipamento e terminais de comunicação de dados é o *Sistema de Ponto de Venda (SPV)*. Estes sistemas são usados em estabelecimentos varejistas para inserir dados no ponto exato em que se realiza a transação. A tabela abaixo descreve as principais características dos sistemas SPV.

Principais Aplicações do SPV

<i>Área</i>	<i>Características</i>		
	<i>Controle de Estoque</i>	<i>Relatório Rápido da Transação em Dinheiro</i>	<i>Contas a Receber</i>
Supermercados	✓	✓	
Lanchonetes	✓	✓	
Lojas	✓		✓

As caixas registradoras eletrônicas são freqüentemente usadas junto com os sistemas SPV para determinar o preço líquido e o imposto no total, para controlar cupons ou descontos etc.

Normalmente, o operador insere o código de barra dos produtos. Para artigos de supermercados, este código de barra é denominado *Código de Produtos Universal (CPU)* e contém dados identificando o produto e o fabricante. O computador então "procura" o preço do artigo e o registra. São também mantidos dados de estoque sobre o artigo (Figura 4.7).

O operador pode inserir o código de barra dos produtos usando:

1. Uma barra que é esfregada no código de barra.
2. Um raio laser que examina o código de barra.
3. Um dispositivo de teclado.

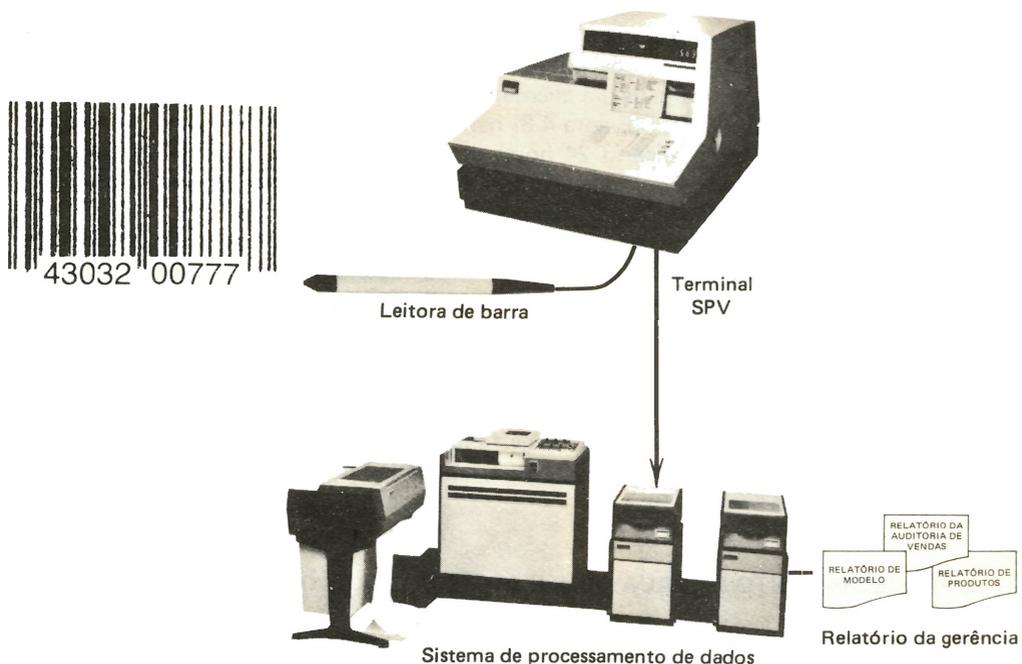


Figura 4.7
 Código de Produto Universal. (b) Exemplo de um sistema típico de ponto de venda. (Cortesia da National Register.)

Dentre as vantagens de um sistema de ponto de venda estão as seguintes:

Vantagens de um Sistema SPV

1. Minimizar o risco de erro do operador.
2. Serve como um dispositivo automático de entrada de dados para processamento por computador.
3. Por ser usado para atualizar imediatamente os registros de estoque e as contas de cliente (em lojas).
4. Pode ser usado para verificar a situação do crédito do cliente antes de debitar um artigo na conta (em lojas).
5. Pode ser usado juntamente com um sistema eletrônico de transferência de fundos, conforme será discutido, para efetuar automaticamente a transferência de fundos da conta bancária de um cliente para a conta de uma loja.

B. SISTEMAS ELETRÔNICOS DE TRANSFERÊNCIA DE FUNDOS (TEF)

Os *Sistemas Eletrônicos de Transferência de Fundos* (TEF) usam computadores em bancos para inserir dados no ponto de transação e para atualizar automaticamente os registros bancários. O sistema TEF pode operar em uma única agência, em uma

rede ligando todas as agências filiais a uma grande instalação bancária central, e em conjunto com os sistemas de ponto de venda para descontar automaticamente os fundos da conta bancária de um cliente, cada vez que ele comprar alguma coisa.

As *caixas automáticas* (Figura 4.8) nas agências bancárias são terminais que podem prestar serviço bancário 24 horas por dia. O cartão plástico de identificação bancária do cliente é inserido na máquina e um computador central irá primeiramente verificar e depois autorizar um depósito ou retirada.

A operação bancária pré-autorizada é uma característica do sistema TEF que possibilita a realização de tal operação eletronicamente. Cheques de salário, pagamentos de Assistência Social ou dividendos de ações podem ser depositados direta e automaticamente no banco. De maneira semelhante, o banco fará automaticamente pagamentos de hipotecas, ou carros, ou transferirá qualquer quantia a pedido do cliente.

Os sistemas TEF usam uma carteira de compensação automatizada, isto é, uma rede eletrônica que facilita a troca de fundos entre as instituições financeiras. Isto reduz a necessidade de troca de cheques, bem como o tempo necessário para se processar as transações.

C. PROCESSAMENTO DE PALAVRAS E AUTOMATIZAÇÃO ADMINISTRATIVA

Muitos sistemas estão sendo planejados para facilitar o processamento de informação no escritório. Consideraremos os dois mais comuns: processamento de palavras e correio eletrônico.

1. Sistemas de Processamento de Palavras

Um *sistema de processamento de palavras* (Figura 4.9) possui geralmente uma UCP, um teclado de TRC e uma impressora. Os relatórios, correspondências e outros do-

Figura 4.8

Caixa automática. (Cortesia da NCR.)





Figura 4.9
Exemplo de um sistema de processamento de palavras. (Cortesia de Wang Laboratories, INC.)

documentos são inseridos através de teclado e exibidos na tela. Quando estão totalmente corretos, podem ser armazenados ou impressos. Os textos podem ser usados, reformatados, editados e atualizados conforme o necessário.

2. Correio Eletrônico

Com o uso de um *dispositivo telefac-símile* e uma linha de comunicação de dados é possível transmitir cópias de um documento para uma ou mais localizações remotas, numa questão de minutos ou mesmo segundos (Figura 4.10). Estes dispositivos

Figura 4.10
Dispositivos telefac-símile. (Cortesia da Graphic Sciences, Inc.)



telefac-símiles são usados para uma grande variedade de aplicações de *correio eletrônico*.

De modo semelhante, os dados gerados por um terminal podem ser exibidos em outras localizações através das linhas de comunicação de dados. Esta capacidade de compartilhar dados reduz a necessidade de despachar documentos importantes de um local para outro.

Os sistemas de processamento de palavras e correio eletrônico diminuem o trabalho manual feito geralmente pelas secretárias. A necessidade de rebater documentos, enviar memorandos etc. é minimizada. Portanto, no futuro, as secretárias atuarão mais como administradoras de escritório, organizando e controlando a informação ampliada que estará disponível no escritório.

TERMOS-CHAVE

Acoplador acústico
Aparelho telefac-símile
Bastão de comando
Caixa automática
Caneta luminosa
Código de Produto Universal (CPU)
Comunicação de Dados
Conectado por fio
Correio eletrônico
Entrada de dados a distância
Equipamento de reconhecimento de voz
Hard copy
Interface
Leitora de código de barra
Linhas de comunicação
Modulador/demodulador
Pergunta-resposta
Sistema eletrônico de transferência de fundos (SETF)
Sistema de ponto de venda (SPV)
Sistema de processamento de palavras
Soft copy
Taxa de bauds
Teleprocessamento
Tempo compartilhado
Terminal
Terminal de máquina de escrever
Terminal inteligente
Terminal manual
Tubo de raio catódico (TRC)
Unidade de resposta audível

QUESTIONÁRIO

(As respostas para as questões de números pares acompanham o teste.)

A. Questões do Tipo Verdadeiro-Falso

V F

_____ 1. Os terminais são ideais para o processamento em linha por estarem localizados no ponto de transação.

- _____ 2. Um terminal pode ser qualquer dispositivo de E/S que está localizado longe da UCP.
- _____ 3. Uma unidade de resposta audível é o tipo mais comum de terminal.
- _____ 4. Um tipo comum de terminal de *hard copy* é o TRC.
- _____ 5. Os terminais podem transmitir sinais através de linhas telefônicas usando um acoplador acústico.
- _____ 6. A capacidade de edição de textos pode ser conseguida com o uso de equipamento de processamento de palavras.
- _____ 7. Os sistemas de ponto de vendas são usados basicamente em supermercados para lidar com as contas a receber.
- _____ 8. A caixa automática pode proporcionar ao cliente um serviço bancário de 24 horas por dia.
- _____ 9. Os estudantes que usam um computador para inserir programas BASIC de um terminal estão freqüentemente acessando um equipamento de tempo compartilhado.
- _____ 10. A taxa de bauds refere-se à velocidade da linha de comunicação.

B. Preencha as lacunas

1. O termo usado para descrever a transmissão eletrônica de dados por telefone, telégrafo, satélite ou estações de microondas é
2. Um(a) é um terminal que pode fornecer ao usuário uma mensagem verbal ao invés de uma mensagem impressa ou exibida na tela.
3. Um dispositivo que faz com que a UCP se comunique com um terminal por telefone é denominado um(a)
4. é a possibilidade de processar mais de um programa simultaneamente.
5. é o termo usado para descrever o processamento de dados no ponto onde os dados se originam realmente.
6. Um dispositivo pode interpretar a linguagem oral e transmitir os dados à UCP.
7. Um terminal com capacidade de microprocessamento embutida é denominado
8. Às vezes, quando um terminal não está muito longe da UCP, pode ser ligado por um cabo ou à unidade principal.
9. SPV é uma abreviação para
10. usa computadores em bancos para inserir dados no momento de uma transação e para atualizar automaticamente os registros bancários.

Respostas

(Fornecemos respostas somente para as questões de números pares.)

A. Questões do Tipo Verdadeiro-Falso

- 2. V
- 4. F — os TRC produzem *soft copy*
- 6. V
- 8. V
- 10. V

B. Preencha as lacunas

2. Unidade de resposta audível
4. Multiprogramação
6. Reconhecimento de voz
8. Conectado por fio
10. Sistemas eletrônicos de transferência de fundos (TEF)

5

MINI E MICROCOMPUTADORES

- I. INTRODUÇÃO AOS MINI E MICROCOMPUTADORES
 - A. DEFINIÇÃO DE UM MINI E UM MICROCOMPUTADOR
 - B. APLICAÇÕES DOS MINICOMPUTADORES
 - C. DESENVOLVIMENTO DOS MICROCOMPUTADORES
 - 1. Educação
 - 2. Pequenas Empresas
 - 3. Grandes Organizações
 - 4. Uso Doméstico
 - D. DISPOSITIVOS DE ENTRADA/SAÍDA USADOS COM MINI E MICROCOMPUTADORES
 - 1. Sistema Básico
 - 2. Unidades Adicionais
 - E. AVALIAÇÃO DOS MINI E MICROCOMPUTADORES
 - 1. Velocidade
 - 2. Comprimento de Palavra
 - 3. Tipo de Microprocessador Usado com um Microcomputador
 - 4. Tipo de Cabo de Interconexão
 - 5. *Software* Disponível
 - II. MICROCOMPUTADORES MAIS VENDIDOS
 - III. SISTEMAS OPERACIONAIS
 - IV. MICROCOMPUTADORES E O CONSUMIDOR
- TERMOS-CHAVE
- QUESTIONÁRIO

I. INTRODUÇÃO AOS MINI E MICROCOMPUTADORES

A. DEFINIÇÃO DE UM MINI E UM MICROCOMPUTADOR

Não há nenhum consenso real na definição de minicomputador, mas nós o definiremos como um computador que custa de US\$10,000 a US\$50,000 na sua forma básica e tem um tamanho básico de memória de pelo menos um megabyte. Isto os diferencia dos *microcomputadores*, que definimos como unidades autônomas que são vendidas por várias centenas de dólares, na sua forma básica, e têm capacidade mais limitada.

Mesmo com estas definições é às vezes difícil categorizar os computadores, especialmente porque pode-se adquirir capacidade adicional para qualquer sistema, o que pode torná-lo tão potente quanto uma máquina maior.

B. APLICAÇÕES DOS MINICOMPUTADORES

Existem três usos básicos dos minicomputadores.

Aplicações dos Minicomputadores

1. Computador Autônomo Financeiro para Uso Geral (veja a Figura 5.1)
Os minicomputadores podem ser usados para aplicações financeiras padrão, tais como folhas de pagamento, contas a receber, estoque etc.
2. Sistema Exclusivo, Completo, ou para Uso Especial
Um minicomputador pode ser usado, por exemplo, como um sistema exclusivo de processamento de palavras, que já vem provido tanto de *software* como *hardware* e o usuário simplesmente "liga" o sistema para operá-lo. Um minicomputador pode ser usado de maneira semelhante em uma fábrica ou indústria para manter os dados em controle de qualidade.
3. Módulo em um Sistema de Processamento de Dados Distribuído
Os minicomputadores podem ser ligados em uma *rede* para processarem os dados no local e transmiti-los posteriormente a uma instalação centralizada. Neste caso, o minicomputador serve como um processador terminal.

Os minicomputadores estão entre os computadores que mais se desenvolvem atualmente, abrangendo mais de um quinto de todo o mercado. Em cerca de um milhão de microcomputadores vendidos, 60% são usados como aplicações financeiras autônomas ou como sistemas completos em companhias pequenas, e 40% são usados como parte de uma rede distribuída.



Figura 5.1
Computador financeiro autônomo para uso geral. (Cortesia da Honeywell.)

C. DESENVOLVIMENTO DOS MICROCOMPUTADORES

O desenvolvimento dos microcomputadores nos últimos anos foi muito além das expectativas iniciais. As áreas em que eles mais se desenvolveram são:

1. Educação

A maioria das escolas de primeiro e segundo graus, bem como as universidades*, possuem atualmente microcomputadores disponíveis para ensinar aos alunos a linguagem BASIC ou qualquer outra linguagem de programação, ou mesmo para instrução com assistência do computador.

2. Pequenas Empresas

Várias empresas de pequeno porte adquiriram microcomputadores, principalmente para o processamento de palavras e outras aplicações administrativas típicas tais como contabilidade, estoque e folha de pagamento.

3. Grandes Organizações

Muitas companhias de grande porte adquiriram microcomputadores como centros operacionais para gerentes, módulos em um ambiente de processamento de dados distribuído, ou como unidades autônomas de processamento de palavras.

* O autor se refere às escolas e Universidades dos Estados Unidos.

4. Uso Doméstico

Um número cada vez maior de microcomputadores está sendo usado pelo consumidor. Já é possível prever que até o ano 2000 praticamente todas as casas terão um minicomputador, que poderá ter as seguintes utilizações:

1. Controlar o uso de energia.
2. Servir como um sistema de segurança.
3. Processamento de palavra.
4. Telecomunicação com o sistema de reembolso postal, corretores da bolsa, farmácia local etc.
5. Correio eletrônico.
6. Manutenção de registros pessoais.
7. Instrução com assistência do computador.
8. Jogos.

Consideraremos com maiores detalhes o uso dos microcomputadores pelos consumidores na seção IV deste capítulo.

D. DISPOSITIVOS DE ENTRADA/SAÍDA USADOS COM MINI E MICROCOMPUTADORES

1. Sistema Básico

As unidades básicas dos mini e microcomputadores incluem um TRC, UCP e teclado. Um monitor de televisão pode ser usado no lugar do TRC. A UCP tem uma capacidade de memória que varia enormemente: desde 4K, para sistemas de microcomputadores bem pequenos, que são vendidos por algumas centenas de dólares, até vários megabytes, para sistemas maiores, cujo preço chega às vezes a várias centenas de milhares de dólares, incluindo os equipamentos periféricos.

2. Unidades Adicionais

Além da configuração básica de um TRC, UCP e teclado, adquirem-se frequentemente unidades adicionais. Estas incluem (1) um dispositivo de armazenamento externo, (2) uma impressora de cópia e (3) outros dispositivos para ampliar o sistema.

a. Armazenamento Externo

Uma *fita cassete ou cartucho* é semelhante a uma minifita magnética e é usada basicamente para processamento por lote em um microcomputador; isto é, os minicomputadores raramente usam dispositivos de fita.

Os gravadores cassete comuns de uso doméstico podem ser utilizados para armazenar dados em uma fita cassete ou ler dados nela contidos. Os dispositivos de cartucho, desenvolvidos especificamente para computadores, são semelhantes a gravadores cassete (Figura 5.2). As principais desvantagens dessas duas unidades são a velocidade lenta de transmissão e a incapacidade de processar dados aleatoriamente.

Um *disco flexível*, de forma semelhante a um conjunto de discos padrão, armazena dados utilizando métodos de acesso direto e é bastante usado com mini e microsistemas. Portanto, o tempo de acesso de uma unidade de disco flexível é bem mais rápido que o de um cassete ou cartucho.



Figura 5.2
Dispositivo de cartucho de fita. (Cortesia da 3M Company.)

As unidades de disco flexível são os meios de armazenamento externo mais populares tanto para mini como para microcomputadores (Figura 5.3). Existem dois tamanhos básicos de disco flexível: (1) versão de 20 cm e (2) versão de 13,3 cm.*

Um disco flexível de 20 cm (Figura 5.4) na sua forma básica tem uma capacidade de armazenamento de 250K de caracteres. Os discos flexíveis de 20 cm de densidade dupla podem armazenar o dobro do número de *bits* e conseqüentemente têm o dobro de capacidade de armazenamento. Os discos flexíveis de 20 cm de *dupla face* lêem e registram dados em ambas as faces do disco; isto também dobra a sua capacidade de armazenamento. Assim, um disco flexível de 20 cm com densidade e face duplas tem uma capacidade de armazenamento de mais de 1 milhão de caracteres.

Para que uma unidade de disco possa processar discos com densidade e face duplas, tem que ter um controlador especial e *software* adicional.

O disco flexível de 20 cm tende a ser mais utilizado para negócios do que para uso doméstico, e mais com mini do que com microcomputadores. Em geral, um grande problema com discos flexíveis é a sua falta de compatibilidade, particularmente se são discos de densidade dupla. A maioria dos discos flexíveis de 20 cm de densidade simples pode ser lida por uma grande variedade de computadores.

O minidisco flexível de 13,3 cm pode armazenar 80K de caracteres na versão de densidade simples, mas existem no mercado versões de densidade dupla. Estes dis-

* 8 e 5 1/4 polegadas, respectivamente.



Figura 5.3
Unidade de disco flexível. (Cortesia da Mohawk Data Sciences.)

cos de 13,3 cm não são em geral compatíveis, e por isso um disco produzido por um Apple, por exemplo, não pode ser lido por um TRS80. Apesar desta falta de compatibilidade, os discos de 13,3 cm são os mais populares para microcomputadores.

Observe que muitos usuários consideram inadequada a capacidade de armazenamento dos discos flexíveis. Ainda que os discos flexíveis de densidade e face duplas possam armazenar mais de um milhão de caracteres, isto não é muitas vezes suficiente. Visto que uma página de texto usa tipicamente uma média de 2K de caracteres, um disco flexível pode armazenar no máximo 500 páginas de dados. Isto pode

Figura 5.4
Disco flexível. (Cortesia da BASF.)



não ser suficiente para algumas aplicações, particularmente para as que exigem a manutenção de grandes arquivos.

Como resultado, as *unidades de disco rígido* tornaram-se recentemente acessíveis para mini e microcomputadores. Um disco rígido padrão pode armazenar entre 5 e 70 milhões de caracteres, mas existem alguns com capacidades bem maiores. A maioria destas unidades de disco rígido usa uma inovação da IBM denominada *tecnologia de Winchester*.

As *unidades de disco de Winchester* fornecem maior capacidade de armazenamento e maior confiabilidade do que os outros tipos de unidades de disco. Fornecem maior proteção com relação a problemas do meio ambiente; tais como poeira, por exemplo, pois com a tecnologia de disco de Winchester o disco e a cabeça de leitura/escrita ficam selados dentro de um único módulo. O mecanismo de leitura/escrita é muito leve, proporcionando uma proteção maior contra os "*choques de cabeça de gravação*". Estes choques ocorrem às vezes em outros dispositivos de disco quando poeira e outras partículas provocam a colisão da cabeça com a superfície do disco, resultando na perda de todos os dados contidos no disco. A Figura 5.5 mostra uma unidade de disco de Winchester.

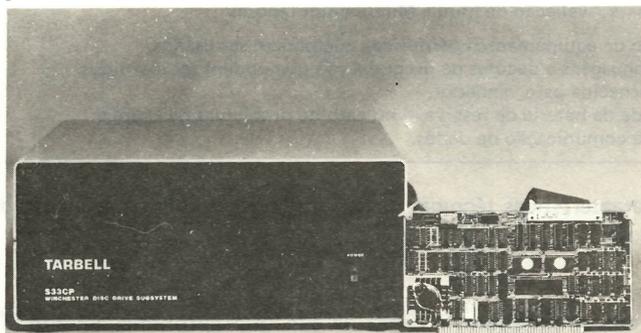
Observe que os discos rígidos têm uma capacidade muito maior do que os flexíveis, mas há um preço a pagar por esta vantagem: as unidades de disco rígido são de 2 a 10 vezes mais caras que unidades de disco flexível.

b. Impressora

Se um sistema requer cópias impressas, é necessário que tenha uma impressora. Se as cópias impressas são usadas para produzir relatórios ou documentos formais, é necessário que o sistema possua uma impressora de letra de qualidade. A impressão produzida por impressoras de letra de qualidade é bastante clara e fácil de ler. Diablo e Qume estão entre os maiores fabricantes de impressoras de letra de qualidade.

Apesar destes dispositivos produzirem saída de alta qualidade, são geralmente muito mais lentos e mais caros que as impressoras de matriz de ponto de alta velocidade. Uma impressora típica de letra de qualidade, que produz de 45 a 55 caracteres por segundo, custa de US\$1000 a US\$2000. A maioria possui uma unidade

Figura 5.5
Unidade de disco de Winchester. (Cortesia da Tarbell Electronics.)



de *avanço de tração* para alimentar formulários contínuos; podem em geral ser adquiridas com um adaptador para inserir folhas individuais que são impressas através de um mecanismo de *avanço de fricção*.

Alguns usuários possuem duas impressoras, um dispositivo de alta velocidade para controle ou fiscalização, onde a qualidade da impressora não é crucial, e uma impressora de letra de qualidade para gerar relatórios.

Na escolha de uma impressora, deve-se considerar os seguintes critérios:

1. É necessário letra de qualidade?
2. Velocidade.
3. Tamanho de papel necessário.
4. Caracteres de impressão disponíveis junto com a impressora (i.e., são necessários símbolos matemáticos, alfabeto estrangeiro etc.?).
5. Fontes de tipos disponíveis.
6. Exige-se capacidade gráfica?
7. Confiabilidade de funcionamento.
8. Tipo de papel necessário.

c. Outros Dispositivos

Assim como os sistemas maiores, os minicomputadores muitas vezes possuem dispositivos periféricos tais como leitoras de cartão, unidades de fita, dispositivos de reconhecimento ótico etc.

De forma semelhante, quase não há restrição quanto ao tipo de unidades de *hardware* que podem ser ligadas a um microcomputador. Por exemplo, os bastões e barras de comando podem ser usados para jogos bem como para sintetizadores de música e de voz.

Muitos mini e microcomputadores têm *capacidade de multiprogramação*, permitindo que vários terminais compartilhem da mesma UCP.

E. AVALIAÇÃO DOS MINI E MICROCOMPUTADORES

Os usuários que adquirem minicomputadores devem considerar os seguintes critérios gerais para avaliar estes sistemas:

Critérios Gerais para Avaliação de Mini e Microcomputadores

1. Número e tipo de equipamentos periféricos que podem ser usados.
2. Número de linguagens e pacotes de programação que podem ser mantidos.
3. Assistência fornecida pelo vendedor.
4. Disponibilidade da bateria de reserva — no caso de problemas de energia.
5. Capacidade de comunicação de dados.

As seguintes características técnicas devem ser consideradas na avaliação de mini e microcomputadores:

1. Velocidade

A maioria dos microcomputadores é capaz de executar aproximadamente 250.000 ou mais instruções por segundo, ao passo que os minicomputadores podem executar 500.000 ou mais instruções por segundo.

2. Comprimento de Palavra

A capacidade de um mini ou microcomputador não depende somente da velocidade, mas da quantidade de dados que pode manusear em uma única operação. Tal quantidade denomina-se palavra. Cada palavra consiste de uma seqüência de *bits*. Quanto maior o comprimento de palavra, mais rápido o processamento e maior a capacidade e poder de manipulação do computador.

A maioria das unidades principais, incluindo minicomputadores, tem um comprimento de palavra de 32 a 64 *bits*, mas existem dispositivos de potência elevada, atualmente disponíveis, com comprimentos de palavra ainda maiores.

Quando introduzidos, os microcomputadores tinham um comprimento de palavra de 4 *bits*. Por isto eram lentos, de certo modo ineficientes e difíceis de programar. Pouco depois, os comprimentos de palavras de 8 *bits* tornaram-se populares. Os microcomputadores mais novos, tais como o computador IBM de uso pessoal, possuem um comprimento de palavra de 16 *bits*.

3. Tipo de Microprocessador Usado com um Microcomputador

Cada minicomputador tem um tipo diferente de microprocessador ou *chip* usado para memória. Os chips mais usados incluem:

<i>Chips</i>	<i>Exemplos de Computadores</i>
Intel 8080	North Star, Heath
Intel 8088	Computador IBM de Uso Pessoal
MOS 6502	Apple, Atari, PET
Zilog Z80	Radio Shack, Cromemco

4. Tipo de Cabo de Interconexão

As conexões internas usadas para transmitir dados de uma parte do computador para outra são denominadas *cabo de interconexão*. Um dos cabos de interconexão mais comuns é o S-100.

5. Software Disponível

Existem inúmeros pacotes de programação disponíveis para mini e microcomputadores. Por exemplo, o BASIC é apresentado em várias formas. O CBASIC é uma versão específica para uso comercial; o Tarbell BASIC é uma versão totalmente interativa que interpreta cada linha de código quando esta é inserida, em vez de interpretá-las depois que todas as instruções foram inseridas. Outras linguagens de programação disponíveis para muitos mini e microcomputadores, em inúmeras versões, são: COBOL, FORTH, FORTRAN, Pascal etc.

Três pacotes de processamento de palavra populares são: Word Star, Magic Wand e Electric Pencil, sendo que todos podem satisfazer as necessidades de muitos usuários para edição de textos. O Word Star tem também um complemento Spell/Star que verifica a ortografia e uma opção Mail Merge que é útil na preparação de letras de impressão.

II. MICROCOMPUTADORES MAIS VENDIDOS

Os microcomputadores que têm sido mais vendidos até agora e que têm condições de manter essa tendência são:

1. Apple
2. Computador IBM de Uso Pessoal (Figura 5.6)
3. TRS 80 (Radio Shack) (Figura 5.7)
4. Commodore PET
5. Atari

O Apple II Plus e o Apple III têm uma capacidade básica de 48K e custam menos de US\$2.000.

A IBM introduziu a vertente do computador de uso pessoal no final de 1981 e já detém uma parcela considerável do mercado. O IBM PC, como é denominado, é um dos primeiros minicomputadores a usar um *chip* de 16 *bits* que, como já foi mencionado, proporciona-lhe o maior potencial em termos de capacidade de armazenamento e velocidade. A versão básica de 48K tem um preço semelhante ao do Apple.

O TRS 80 também possui modelos diferentes. O Model 16 é a sua versão de 16 *bits*. Tanto o Apple como o IBM e o TRS 80 têm modularidade incorporada, de maneira que outras unidades podem ser acrescentadas conforme necessário.

O Commodore PET é bastante usado em escolas de primeiro e segundo graus, assim como o Atari 800 e o Apple. O Atari vende o seu modelo 400, uma versão de 4K, por várias centenas de dólares e o Atari 800 básico, uma unidade básica de 16K, por US\$1.000.

O TRS 80 é encontrado em qualquer distribuidora da Radio Shack. A maioria dos outros microcomputadores pode ser adquirida em uma loja especializada no ramo de computadores ou outra distribuidora local de computadores. A IBM assinou também um acordo com a SEARS permitindo que essa cadeia de lojas venda seus

Figura 5.6
Computador IBM de Uso Pessoal. (Cortesia da IBM.)





Figura 5.7
TRS 80. (Cortesia da Radio Shack.)

computadores, e vários outros fabricantes fizeram contrato com outros grandes estabelecimentos de vendas a varejo.*

Outros fabricantes importantes de computador, incluindo a Digital Equipment Corporation, acompanharam a IBM no setor de computadores de uso pessoal. As previsões para 1985 com relação à participação no mercado de minicomputadores são as seguintes:

Previsões Referentes à Participação no Mercado em 1985

IBM	23%
Xerox	17%
Tandy/Radio Shack	15%
Hewlett-Packard	14%
Apple	12%
DEC	11%
Commodore	3%
Todas as outras	5%

Fonte: International Resource Development Inc.

* Tais dados dizem respeito ao mercado norte-americano.

III. SISTEMAS OPERACIONAIS

Para que qualquer computador seja usado eficientemente, este deve ter um programa de controle ou *sistema operacional* para facilitar o armazenamento de arquivos, execução de programas, impressão etc. Alguns microcomputadores possuem sistemas operacionais incorporados. Entretanto, a maioria deles fornece o seu próprio sistema operacional como *software*, ou possibilita aos usuários adquiri-lo. Por exemplo, o computador IBM de uso pessoal tem o seu próprio sistema operacional, mas também pode utilizar outros mais sofisticados.

Há um sistema operacional para microcomputadores que se tornou um tipo de padrão na indústria e pode ser utilizado em uma enorme variedade de computadores. O *CP/M*, que significa Programa de Controle para Microprocessadores, é um sistema operacional em discos produzido pela Digital Research Company. Existem versões do CP/M para sistemas de disco flexível de 20 cm e 13,3 cm, bem como para sistemas de disco rígido. O CP/M-86 destina-se a microcomputadores de *16-bits*, tais como o IBM. Além disto, muitos fabricantes importantes estão oferecendo o CP/M aos seus usuários.

Para utilizar o CP/M, o microcomputador deve ter pelo menos 16 K de memória principal e algum tipo de disco. Alguns comandos básicos no CP/M que facilitam o processamento de programas incluem os seguintes:

- ERA — apague um arquivo.
- DIR — liste todos os nomes do arquivo.
- STAT — indique os números das posições de memória usados para um programa ou arquivo específico.
- SAVE — conserve todas as informações armazenadas em um arquivo de disco.

O CP/M tem também um comando de edição, ED, que permite ao usuário editar todos os arquivos e programas. Entretanto, se a edição de texto for uma necessidade importante, recomenda-se editoras mais sofisticadas.

Há também uma versão do CP/M denominada MP/M, um sistema operacional que possibilita a vários usuários rodar programas simultaneamente usando uma UCP. O MP/M é um programa de multiprocessamento com controle monitorizado.

IV. MICROCOMPUTADORES E O CONSUMIDOR

A disponibilidade de microcomputadores de baixo custo teve como resultado um mercado totalmente novo para o computador: o consumidor. Prevê-se que até 1985 3.8 milhões de computadores de uso pessoal sejam vendidos, somando-se um total de US\$ 4.9 bilhões.*

O computador doméstico é tipicamente usado para os seguintes tipos de aplicações:

* Os dados dizem respeito ao mercado norte-americano.

Aplicações do Computador de Uso Pessoal para o Consumidor

1. Diversão

Os computadores de uso pessoal podem jogar bridge, xadrez, gamão e milhares de outros jogos. O usuário simplesmente adquire um programa pré-fabricado em uma fita cassete ou disco flexível. Naturalmente, os usuários podem escrever os seus próprios programas de jogos.

2. Instrução com Assistência do Computador

Há uma enorme variedade de assuntos que podem ser ensinados utilizando-se um computador de uso pessoal. Estão à venda programas que treinam os estudantes em multiplicação, ortografia, geografia etc. Muitos destes incluem espetáculos de desenho animado, projetados para prender a atenção de crianças mais novas, bem como dos usuários mais velhos. Há também sintetizadores de música que possibilitam às pessoas aprender a tocar e fazer experiências com música. A programação também pode ser ensinada através de instrução com assistência do computador.

3. Aplicações Financeiras e Pessoais

Há programas disponíveis para controlar talões de cheque, fazer devolução de imposto de renda, guardar receitas e cardápios de dieta etc. A lista de usos pessoais do computador é praticamente ilimitada. Existem também no mercado programas tais como o VisiCalc, que oferecem capacidade de "tabelamento eletrônico".

4. Processamento de Palavra

Os computadores podem ser usados para armazenar relatórios, documentos e correspondências e para imprimi-los. O TRC é usado para exibir uma cópia de um documento e corrigi-la até que fique perfeita; a cópia é então impressa usando-se uma impressora. O arquivo pode ser então armazenado para posterior atualização, edição ou reformulação.

5. Elo de Comunicação com um Serviço de Tempo Compartilhado

Inúmeras organizações fornecem as últimas cotações da bolsa de valores, informações sobre serviços aéreos, notícias, pesquisas literárias etc. O microcomputador pode ser usado como um terminal que acessa um ou todos estes principais bancos de dados.

Existem já em funcionamento muitos serviços que permitem aos consumidores que usam mini ou microcomputadores comunicar-se, por linhas telefônicas, com outro usuário ou para serviços específicos tais como:

Resumo de notícias do "New York Times".

Informações da UPI.

Informações sobre ações no mercado.

Horário de serviços aéreos.

Reservas de hotel.

Reservas de restaurante.

Serviços de entrega a domicílio.

Correio eletrônico.

Jogos.

Operações bancárias em casa.

O usuário de um mini ou microcomputador disca um número de telefone local e registra o código e a senha da conta adequada. O usuário pode então escolher um dos serviços de um "menu" variado. Os preços típicos para este serviço são os seguintes:*

* Os dados são referentes aos serviços existentes nos Estados Unidos.

Taxa inicial de US\$100

Preços em relação ao horário:

De 7 às 18h	US\$18 por hora
De 18 às 24h	US\$5.75 por hora
De 24 às 7h	US\$4.25 por hora

Há uma taxa mínima de US\$10 por mês.

Espera-se que estes preços possam sofrer uma redução à medida que aumentar o número de usuários contribuintes destes serviços.

SUMÁRIO

A. "Prefira o Pequeno"

A tendência atual na indústria de PED gira em torno do desenvolvimento de mini e microcomputadores para utilização em grandes e pequenos negócios, bem como para o mercado doméstico.

B. Diferença entre Mini e Microcomputadores

1. Não existe nenhuma diferença absoluta entre um mini e um microcomputador. Os diversos fabricantes usam termos diferentes para descrever equipamento que às vezes é bastante semelhante.

C. Aplicações dos Minicomputadores

1. Unidades autônomas.
2. Sistemas "completos" para uso especial.
3. Módulos em um sistema distribuído de processamento de dados.

D. Aplicações dos Microcomputadores

1. Diversão em casa.
2. Educação.
3. Aplicações financeiras e pessoais.
4. Processamento de palavra.
5. Elo de comunicação com um serviço de tempo compartilhado.

E. *Hardware* usado com Mini e Microcomputadores

1. UCP — geralmente de mesa.
2. Terminal — com teclado e TRC.
3. Unidades de fita cassete ou cartucho — versões menores das unidades de fitas.
4. Unidades de discos flexíveis ou discos rígidos projetadas para mini e microcomputadores — versões menores das unidades de disco.
5. Impressora.

F. Problemas com Mini e Microcomputadores

1. Os dispositivos periféricos ainda não são compatíveis com unidades maiores.
2. Os dispositivos periféricos não são padronizados na indústria.
3. A assistência do fabricante é muitas vezes inadequada.
4. É muitas vezes difícil encontrar um bom *software*.

TERMOS-CHAVE

Cabo de interconexão

Cartucho de fita

Choque da cabeça de gravação

Compatibilidade

Computador exclusivo

CP/M

Disco de densidade dupla

Disco de dupla face

Disco de Winchester
Disco flexível
Fita cassete
Mecanismo do avanço de fricção
Microcomputador
Minicomputador
MP/M
Multiprogramação
Processamento de palavra
Sistema completo
Sistema operacional
Software
Terminal processador
Unidade de avanço de tração

QUESTIONÁRIO

(As respostas para as questões de números pares acompanham o teste.)

A. Questões do Tipo Verdadeiro-Falso

- | V | F | |
|-------|-------|---|
| _____ | _____ | 1. Um microcomputador tem geralmente uma capacidade maior que um mini-computador. |
| _____ | _____ | 2. A diferença entre mini e microcomputadores é bastante nítida. |
| _____ | _____ | 3. Os minicomputadores têm excelente potencial para negócios de pequeno porte, mas têm realmente muito pouco potencial em grandes organizações. |
| _____ | _____ | 4. Uma unidade de fita cassete padrão pode ser usada com um microcomputador. |
| _____ | _____ | 5. Os discos flexíveis são versões em miniatura dos discos rígidos em grande escala. |
| _____ | _____ | 6. Um sistema completo é o que se usa freqüentemente em um ambiente de processamento de dados distribuído. |
| _____ | _____ | 7. É possível adquirir <i>chips</i> que aumentam a capacidade de armazenamento de um sistema de computador. |
| _____ | _____ | 8. Os minicomputadores podem servir de terminais processadores para sistemas maiores de computação. |
| _____ | _____ | 9. Os microcomputadores são freqüentemente usados em casa para aplicações de processamento de palavra. |
| _____ | _____ | 10. É tão fácil adquirir <i>software</i> totalmente compatível para minicomputadores quanto <i>software</i> para sistemas em grande escala. |

B. Preencha as Lacunas

1. Os minicomputadores usados para aplicações especiais, nas quais o usuário não necessita de nenhuma habilidade para lidar com computador, são denominados
2. Um minidisco magnético usado com minicomputadores denomina-se
3. Uma minifita magnética usada com minicomputadores denomina-se
4. O conjunto de dispositivos que integram um sistema de computador denomina-se
5. O programa de controle mais popular para microcomputadores denomina-se
6. Um dispositivo que pode ser usado para modificar dados em um TRC é um

7. Um dispositivo que faz com que um microcomputador possa tocar músicas denomina-se
8. Um fornece aos minicomputadores capacidade de acesso aleatório.
9. O tipo de computador que provavelmente será adquirido para uso pessoal ou doméstico é
10. A linguagem de programação normalmente existente para a maioria dos mini e microcomputadores é

Respostas

(Fornecemos respostas somente para as questões de números pares.)

A. Questões do Tipo Verdadeiro-Falso

2. F 4. V
6. F — os sistemas completos são geralmente autônomos.
8. F 10. F

B. Preencha as Lacunas

2. disco flexível
4. *hardware*
6. teclado, caneta luminosa, bastão de comando, barra
8. disco flexível ou disco rígido
10. BASIC



UNIDADE 3

SOFTWARE

6

CONCEITOS DE SOFTWARE

- I. TIPOS DE SOFTWARE: DESCRIÇÃO GERAL
 - A. SOFTWARE DE APLICAÇÃO
 - B. SOFTWARE DE SISTEMA OPERACIONAL
 - II. PROGRAMAS DE APLICAÇÃO
 - A. COMPREENSÃO DO SISTEMA QUE INCORPORA O PROGRAMA
 - B. PREPARAÇÃO DO PROGRAMA
 - 1. Fluxograma de Programas
 - 2. Programação Estruturada
 - C. CODIFICAÇÃO DO PROGRAMA
 - 1. Código Próprio da Máquina: Linguagem de Máquina
 - 2. Programação Simbólica
 - D. FUNCIONAMENTO DO PROGRAMA
 - 1. Fase de Tradução
 - 2. Fase de Execução
 - E. IMPLEMENTAÇÃO DO PROGRAMA E DO SISTEMA COMPLETO
 - F. DOCUMENTAÇÃO DO PROGRAMA
 - III. PRINCIPAIS LINGUAGENS DE PROGRAMAÇÃO DE ALTO NÍVEL
 - A. APL
 - B. BASIC
 - C. COBOL
 - D. FORTRAN
 - E. PASCAL
 - F. PL/1
 - G. RPG
 - H. ADA: A ELABORAÇÃO DE UMA NOVA LINGUAGEM
 - IV. AVALIAÇÃO DO SOFTWARE DE APLICAÇÃO
 - V. CARACTERÍSTICAS DO SOFTWARE DE SISTEMAS
 - A. SUPERVISOR
 - B. COMUNICAÇÃO COM O SUPERVISOR
 - C. MULTIPROGRAMAÇÃO: EXECUÇÃO DE MAIS DE UM PROGRAMA SIMULTANEAMENTE
 - D. MULTIPROCESSAMENTO
 - E. MEMÓRIA VIRTUAL (MV)*
 - F. QUEM FORNECE O SOFTWARE DE SISTEMAS?
 - VI. SUPORTE LÓGICO INALTERÁVEL: ONDE O HARDWARE E O SOFTWARE SE INTERCALAM
- TERMOS-CHAVE
- QUESTIONÁRIO

* Em inglês VS (VIRTUAL STORAGE)

TIPOS DE SOFTWARE: DESCRIÇÃO GERAL

A. SOFTWARE DE APLICAÇÃO

O *software de aplicação* consiste de programas destinados a suprir as necessidades dos usuários individuais. Pode constar de programas de folha de pagamento, programas de controle de estoque, solicitações a um banco de dados etc.

B. SOFTWARE DE SISTEMA OPERACIONAL

O *software de sistema operacional* consiste de programas que são geralmente fornecidos pelo fabricante ou comerciante de computador que vende, aluga ou arrenda o equipamento para o usuário final. Estes programas otimizam o uso adequado do sistema de computador como um todo. O sistema operacional consiste de um *supervisor* ou programa de controle, que controla todas as operações do computador. O sistema operacional consiste ainda de uma biblioteca de funções, à qual o supervisor pode ter acesso quando necessitar.

Esta seção começa com uma discussão sobre a programação de aplicação, que é a principal tarefa dos programadores contratados por organizações de usuários.

I. PROGRAMAS DE APLICAÇÃO

A lista abaixo representa a seqüência de tarefas necessárias para preparar e completar os programas de aplicação para usuários:

Etapas Envolvidas na Programação de Aplicação:

1. Compreensão do sistema que incorpora o programa.
2. Preparação do programa.
3. Codificação do programa.
4. Funcionamento do programa.
5. Implementação do programa e do sistema completo.
6. Documentação do programa.

A. COMPREENSÃO DO SISTEMA QUE INCORPORA O PROGRAMA

Na maioria das vezes, um departamento ou divisão de usuário dentro de uma organização se encarrega de computadorizar um sistema completo, ou conjunto de pro-

cedimentos. Isto é, em uma organização administrativa, um sistema de Contas a Receber pode ser computadorizado; em uma indústria aeroespacial, o projeto de um novo sistema de aeronave pode ser computadorizado. Quando o analista de sistemas projeta um novo sistema, ele designa a cada programador os programas de aplicação necessários. O programador tem a responsabilidade de compreender como o programa integrará o sistema como um todo.

B. PREPARAÇÃO DO PROGRAMA

Desta forma, a primeira grande responsabilidade do programador é entender perfeitamente as necessidades básicas do sistema como um todo e determinar precisamente como o programa será integrado nesse sistema.

Depois de determinadas todas as especificações, há diversos instrumentos ou técnicas de planejamento que podem ser usados para planejar a lógica a ser incorporada no programa. Estes instrumentos ou técnicas de planejamento são tipicamente utilizados antes de escrever o programa, para minimizar a possibilidade de erros lógicos.

Instrumentos para Planejar a Lógica do Programa

1. Fluxograma de programas.
2. Instrumentos de programação estruturada.
 - a. Pseudocódigo
 - b. Diagrama HIPO

1. Fluxograma de Programas

Um *fluxograma de programa* é uma representação pictórica da lógica a ser usada em um programa. Indica, em forma de diagrama, os principais elementos do programa e como estes se integram logicamente.

Um fluxograma é traçado com o uso de um *gabarito de programação* (Figura 6.1) que contém os símbolos padrões. Cada fluxograma usa os símbolos que representam as operações específicas a serem realizadas. A Figura 6.2 ilustra um fluxograma de programa, especificando a lógica a ser usada em um programa.

Observe que os fluxogramas são traçados *antes* do programa ser escrito, para assegurar que o programador entenda quais são as tarefas especificamente solicitadas e qual a seqüência destas.

2. Programação Estruturada

É uma técnica projetada para padronizar programas, tornando-os, portanto, mais fáceis de codificar, depurar e modificar. Os instrumentos usados para programas estruturados são:

a. Pseudocódigo

Este instrumento de planejamento utiliza um código semelhante a um código de programa para representar a lógica, em vez de um diagrama por blocos. O *pseudo-*

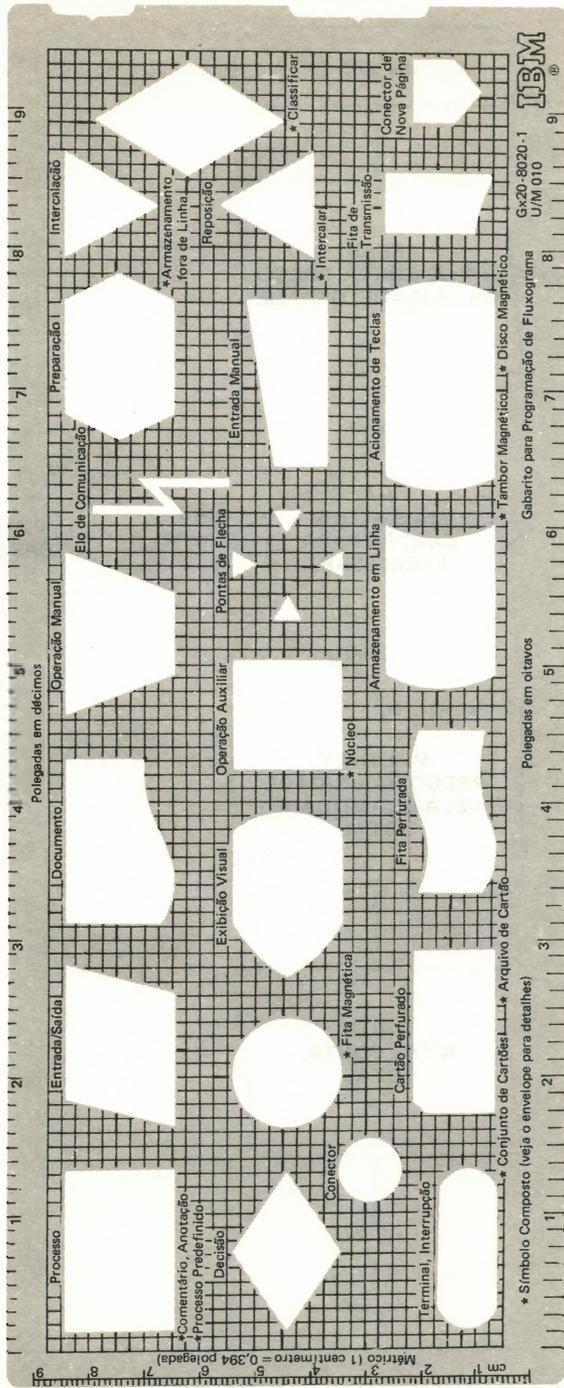


Figura 6.1
Gabarito de programação para fluxograma

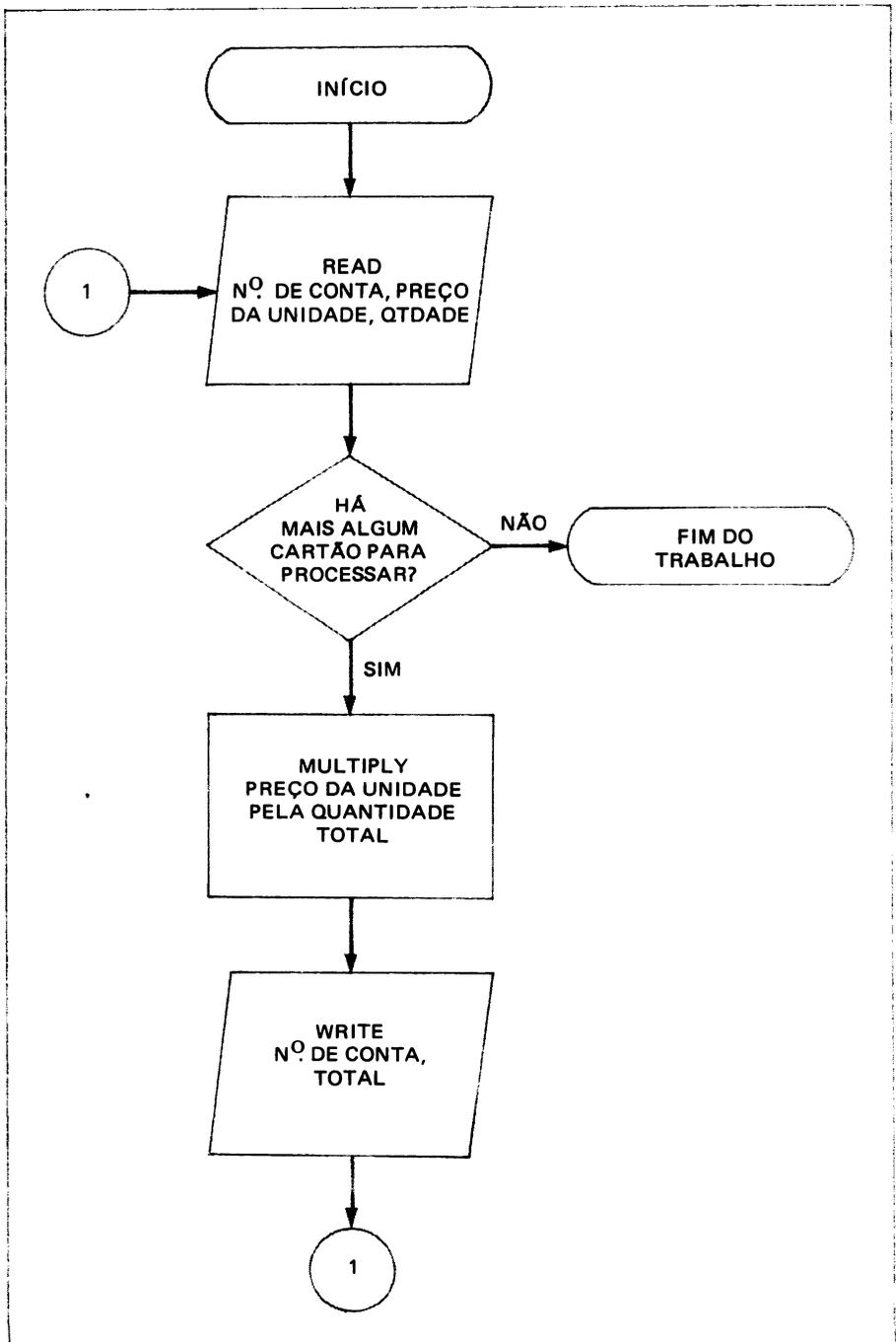


Figura 6.2
Uma amostra de fluxograma

código é amplamente usado para aplicações que possuam procedimentos lógicos de controle que sejam complexos; nestes casos é mais útil uma abordagem estruturada.

b. Diagramas HIPO

Estes são diagramas orientados para sistemas, enfocando a relação hierárquica entre entrada, processamento e saída. HIPO é, na verdade, um acrônimo para Hierarquia e Entrada – Processo – Saída*. Os diagramas HIPO são muitas vezes usados juntamente com a representação mais detalhada do pseudocódigo.

O Capítulo 7 aborda estes instrumentos detalhadamente.

C. CODIFICAÇÃO DO PROGRAMA

Uma vez planejado o fluxo lógico de um programa, usando-se um fluxograma, pseudocódigo ou diagrama HIPO, pode-se escrever a seqüência de instruções de um programa. A escrita destas instruções é denominada codificação do programa.

Os programas podem ser codificados em folhas de papel denominadas folhas de programa ou de codificação. Estas são depois inseridas no computador, usando-se tipicamente um dos seguintes métodos:

Métodos Típicos para a Inserção de Programas no Computador:

1. Digitar o programa em um terminal que é ligado em linha a uma UCP.
2. Digitar o programa em um microcomputador e depois armazená-lo em fita cassete ou disco flexível.
3. Perfurar por teclado o programa em cartões e inseri-lo na modalidade de processamento por lote.

Em cada caso, uma linha do código é teclada sobre uma linha do programa ou um cartão perfurado.

1. Código Próprio da Máquina: Linguagem de Máquina

Para ser executado, um programa de computador tem que estar em *linguagem de máquina*. Este é um código interno altamente complexo que os programadores raramente aprendem, pelas seguintes razões:

Dificuldades Relacionadas com a Programação em Linguagem de Máquina:

1. A linguagem de máquina utiliza códigos de operação complexos.
2. A linguagem de máquina utiliza endereços de máquina reais.

2. Programação Simbólica

Uma *linguagem simbólica de programação* é a mais fácil para o programador codificar. Usa códigos simbólicos de operação, tais como ADD ou +, e endereços simbó-

* Hierarchy – Input – Process – Output.

licos tais como HOLD, RESULT e SUM para representar as posições reais de armazenamento. A grande maioria dos programas são codificados em linguagem simbólica ao invés de linguagem de máquina.

Um programa simbólico deve ser convertido ou traduzido para linguagem de máquina antes de ser executado pelo computador. Existem duas fases necessárias para operar um programa.

Duas Fases Necessárias para Programas Simbólicos

1. Fase de Tradução

O computador traduz o programa simbólico, denominado *programa-fonte*, para um programa de máquina, denominado *programa-objeto*.

2. Fase de Execução

Depois que o programa é traduzido, o computador opera ou o executa.

D. FUNCIONAMENTO DO PROGRAMA

1. Fase de Tradução

As principais linguagens de programação simbólica serão discutidas na Seção III deste capítulo. Cada uma requer um processo de tradução.

Sem levar em conta o processo de tradução usado, existem formas de saída (Figura 6.3).

Resultados de uma Tradução

1. Listagem do Programa-Fonte:

Esta é uma saída impressa, gerada pelo computador, de todas as instruções fornecidas pelo programador.

2. Lista das Violações de Normas, Denominadas *Erros de Sintaxe*.

3. Programa Objeto:

É o equivalente, em linguagem de máquina, ao programa fonte.

Os seguintes tipos de programas tradutores são usados para traduzir programas fonte:

Tipos de Programas Tradutores:

1. Compilador – os programas fonte são compilados.

2. Montador – os programas fonte são montados.

a. Compilador: Compilação de um Programa-Fonte

As linguagens de programação simbólicas de alto nível, tais como FORTRAN e COBOL, são relativamente fáceis de codificar, mas necessitam de processos de tradução complexos, denominados compilações. Quanto mais fácil for a codificação de um programa, mais difícil será para a máquina traduzi-lo para a sua própria linguagem. O programa tradutor é denominado *compilador*.

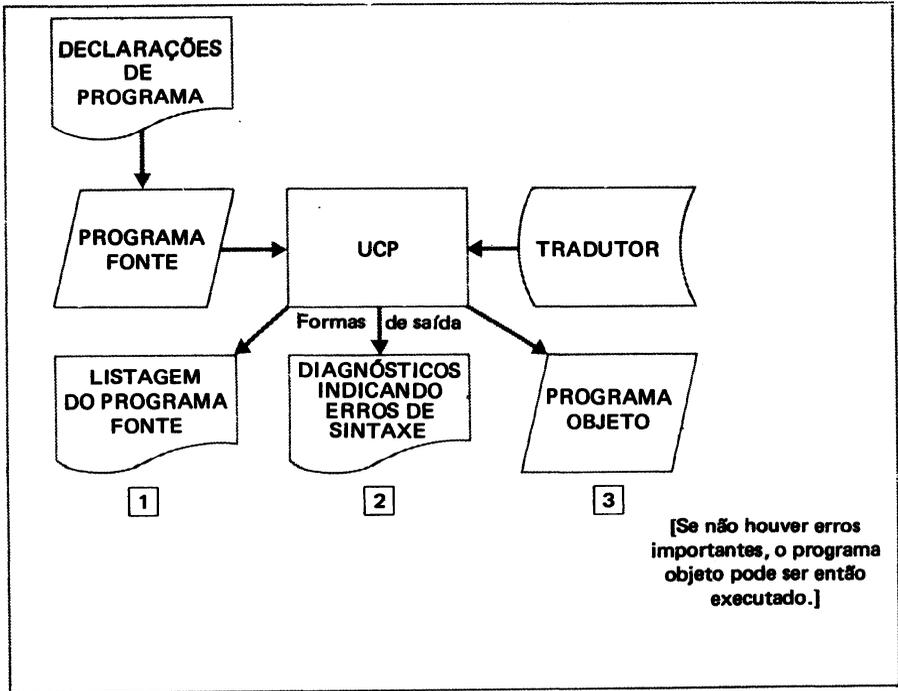


Figura 6.3

Etapas envolvidas na tradução de um programa. O tradutor pode ser um compilador ou um montador.

Cada linguagem de alto nível tem o seu próprio compilador. Portanto, há um compilador FORTRAN, um compilador COBOL, um compilador PL/1 etc.

Cada computador tem a sua linguagem de máquina exclusiva. Assim, um compilador FORTRAN que traduz programas FORTRAN para o IBM 3033 não seria o mesmo que um compilador FORTRAN usado para o UNIVAC 1100, mesmo que o programa-fonte fosse o mesmo.

Ao compilar um programa, insere-se inicialmente todo o programa-fonte; em seguida, inicia-se o processo de tradução. No final da compilação imprime-se, então, uma listagem do programa e quaisquer violações de normas ou erros de sintaxe.

Alguns tradutores, chamados *intérpretes*, podem traduzir programas *interativamente*. À medida que cada linha de codificação é inserida, o computador verifica os erros e comunica imediatamente ao usuário a ocorrência de qualquer erro de sintaxe.

b. Montador: Montagem de Um Programa-Fonte

As linguagens de programação de baixo nível, ou linguagens montadoras, são semelhantes à linguagem de máquina e não necessitam de traduções complexas. Estas linguagens são muito eficientes e são usadas quando é importante otimizar o desempenho do computador. Portanto, os sistemas operacionais ou procedimentos de

controle complexos serão provavelmente programados em uma linguagem montadora. Entretanto, tais linguagens são um tanto quanto difíceis de serem aprendidas e utilizadas.

Neste caso, o tradutor é denominado *montador*. O processo de tradução, denominado montagem, é uma operação de alta velocidade.

Cada computador mantém uma variedade de linguagens de programação. Os grandes sistemas provavelmente mantêm todas ou a maioria das linguagens a serem abordadas, ao passo que os pequenos sistemas, mini ou microcomputadores mantêm somente algumas.

Para sistemas em grande escala, os tradutores fazem parte do sistema operacional e são chamados quando necessário. Para programação interativa, o intérprete está constantemente em linha; isto é, está na memória principal sempre disponível para uso.

2. Fase de Execução

A fase de execução é usada para *depurar* um programa, isto é, eliminar todos os erros ou “falhas de funcionamento”. Opera-se ou executa-se o programa com dados de teste ou de amostra. Compara-se em seguida a saída produzida com a saída que foi preparada manualmente a partir dos mesmos dados. Se tudo conferir, o programa é considerado depurado ou isento de erros, e está pronto para ser executado em termos rotineiros.

E. IMPLEMENTAÇÃO DO PROGRAMA E DO SISTEMA COMPLETO

Uma vez escritos e completamente depurados todos os programas para um sistema específico, o analista de sistemas, juntamente com o programador devem transformar o velho sistema no novo sistema computadorizado de modo a assegurar a não ocorrência de erros.

F. DOCUMENTAÇÃO DO PROGRAMA

Depois que um programa foi totalmente testado e implementado, o programador deve relatar as especificações completas para todos os usuários, documentando, assim, o programa.

III. PRINCIPAIS LINGUAGENS DE PROGRAMAÇÃO DE ALTO NÍVEL

A. APL

APL, um acrônimo para *A Programming Language* (Uma Linguagem de Programação), é uma linguagem de programação interativa de alta potência, que é mais bem inserida através de um terminal.

Um teclado de *APL*, com inúmeros símbolos especiais, é necessário para programação nesta linguagem. A necessidade destes símbolos especiais para a programação torna a *APL* uma linguagem de certo modo difícil de ler e, para alguns, difícil de codificar.

B. BASIC

BASIC é uma das linguagens de programação mais usadas para mini e microcomputadores, bem como para aplicações em tempo compartilhado em computadores

maiores. É também, muitas vezes, uma linguagem interativa e por isso fácil de depurar e corrigir.

BASIC é um acrônimo para *Beginner's All-purpose Symbolic Instruction Code*. O termo "*Beginner*" (Iniciante) significa que é ideal para ser ensinada a pessoas que não tenham nenhum conhecimento prévio de programação. Além de ser a principal linguagem de programação para sistemas de mini e microcomputadores, é também a linguagem de programação ensinada com maior frequência, particularmente para cursos de nível inicial.

A linguagem BASIC possibilita ao iniciante escrever um código simples que usa instruções baseadas em matemática. Por exemplo:

$$\text{LET F} = 9/5 * \text{C} + 32$$

é equivalente à determinação de que a temperatura F (Fahrenheit) é igual a $9/5C + 32$, onde C significa Celsius. Devido à sua notação matemática e estilo de forma livre, BASIC é uma linguagem popular tanto para cientistas como para engenheiros.

C. COBOL

COBOL, um acrônimo para *Common Business Oriented Language* (Linguagem Comum para Atividades Comerciais) é, como o nome indica, uma linguagem comercial que pode ser operada em muitos computadores. Foi criada para satisfazer as necessidades habituais das atividades comerciais e é uma linguagem parecida ao inglês. Portanto, em COBOL, podemos codificar, por exemplo, a seguinte instrução que dispensa explicação:

ADD SALES TAX TO PRICE
(ADICIONE TAXA DE VENDAS AO PREÇO)

Visto que a maioria dos problemas voltados para a atividade comercial opera com grandes quantidades de dados, necessitando de processamento em alta velocidade, uma linguagem comercial tem que ser capaz de manusear, de maneira rápida e eficiente, meios de armazenamento de alta velocidade, tais como fita e disco magnético. COBOL usa instruções que tornam a programação para estes dispositivos de alta velocidade uma tarefa simples.

D. FORTRAN

A linguagem de programação simbólica *FORTRAN* é uma abreviação para *Formula Translator* (Tradutor de Fórmulas). FORTRAN é uma linguagem matemática ideal para descrever fórmulas. Por exemplo, para somar dois campos, A e B, e colocar a resposta em um campo C, usa-se a instrução FORTRAN:

$$\text{C} = \text{A} + \text{B}$$

Observe que esta instrução parece uma fórmula matemática e é bastante semelhante à linguagem BASIC.

FORTRAN é usada com mais freqüência para aplicações em ciência ou engenharia, devido à sua natureza matemática. Inclui propriedades para determinar logaritmos, funções trigonométricas etc. Da mesma forma que as outras linguagens, FORTRAN foi padronizada para poder ser usada em muitos computadores diferentes.

E. PASCAL

PASCAL foi assim denominada por causa do matemático e inventor Blaise Pascal (1623-1662), que desenvolveu uma das primeiras máquinas de calcular.

Considera-se que a linguagem de programação Pascal tem uma grande quantidade de potencial para usuários de computador, pois facilita o uso de técnicas de programação estruturadas. É ensinada atualmente como primeira linguagem em muitos cursos de ciência de computador.

F. PL/1

PL/1, uma abreviação para *Programming Language/1* (Linguagem de Programação/1), é uma linguagem simbólica projetada para satisfazer às necessidades dos usuários de computador tanto na área científica quanto comercial. Isto é, foi projetada para combinar as principais vantagens e características de COBOL e FORTRAN de tal forma que o usuário possa empregá-la tanto para problemas científicos como comerciais. Existem outras versões de *PL/1*, tais como *PL/C* e *PL/M*, que são usadas em alguns sistemas.

G. RPG

Muitas organizações comerciais, particularmente as de menor porte, ou as que utilizam minicomputadores, não necessitam das opções variadas disponíveis em COBOL ou *PL/1*. As necessidades podem ser satisfeitas com uma linguagem simplificada que é usada com maior freqüência para imprimir saídas a partir de cartões, fita ou disco, ou realizar operações simples relacionadas com a atividade comercial.

RPG, uma abreviação para *Report Program Generator* (Gerador Automático de Programas), é uma linguagem simbólica ideal para criar relatórios impressos a partir dos meios de entrada. *RPG* requer um esforço mínimo de programação.

H. ADA: A ELABORAÇÃO DE UMA NOVA LINGUAGEM

Ada é uma linguagem de programação relativamente nova que foi desenvolvida com o apoio do Departamento de Defesa. A linguagem é assim denominada em referência a Ada, a Condessa de Libertino, que projetou para as máquinas de computação de Charles Babbage, no século XIX, o que podemos chamar atualmente de "programa".

Um objetivo importante da linguagem Ada é incentivar a prática da boa programação com estruturas de controle bem desenvolvidas. O projeto estruturado é um elemento crucial desta linguagem.

O Departamento de Defesa tem incentivado ativamente o uso de Ada, mas ela possui também potencial para ser empregada em muitos tipos de aplicações comerciais. Está ainda em fase de desenvolvimento, embora já existam alguns compiladores.

O Quadro 6.1 resume as linguagens de programação de alto nível aqui discutidas. Note-se que existem muitas outras linguagens acessíveis, mas estas são as mais comuns.

Tabela 6.1 Linguagens de Programação de Alto Nível

<i>Nome</i>	<i>Significado</i>	<i>Usos Principais</i>	<i>Características</i>
APL	<i>A Programming Language</i> (Uma Linguagem de Programação).	<ol style="list-style-type: none"> 1. Usada em mini-computadores. 2. Usada em instalações de tempo compartilhado. 	<ol style="list-style-type: none"> 1. Interativa 2. Requer teclado de gráfico especial. 3. Relativamente complexa. 4. Orientada cientificamente.
BASIC	<i>Beginner's All purpose Symbolic Instruction Code</i> (Código de Instruções simbólicas de Caráter Geral para Principiantes).	<ol style="list-style-type: none"> 1. Usada em mini-computadores para operações de produção. 2. Usada em inúmeros sistemas para propósitos de ensino. 	<ol style="list-style-type: none"> 1. Interativa. 2. Fácil de aprender. 3. Usa notação matemática. 4. Pode ser usada para problemas científicos e comerciais.
COBOL	<i>Common Business Oriented Language</i> (Linguagem Comum para Atividades Comerciais).	<ol style="list-style-type: none"> 1. Usada em sistema de pequeno, médio e grande porte. 2. Não está ainda disponível para a maioria dos mini ou microcomputadores (o compilador é demasiado complexo). 	<ol style="list-style-type: none"> 1. Ideal para aplicações comerciais. 2. Manuseia facilmente características de disco e fita. 3. Parecida à língua inglesa.
FORTRAN	<i>Formula Translator</i> (Tradutor de fórmula).	Usado em sistemas de pequeno, médio e grande porte.	<ol style="list-style-type: none"> 1. Ideal para aplicações científicas. 2. Usa notação matemática.
PASCAL	Assim denominada por causa do matemático Blaise Pascal.	Usada em todos os tipos de computadores.	<ol style="list-style-type: none"> 1. Voltada para a ciência. 2. Concentra-se em técnicas de programação estruturadas.
PL/1	<i>Programming Language One</i> (Linguagem de Programação Um).	Usada em sistemas de médio e grande porte.	<ol style="list-style-type: none"> 1. Combina as características de COBOL e FORTRAN.

Tabela 6.1 (continuação)

<i>Nome</i>	<i>Significado</i>	<i>Usos Principais</i>	<i>Características</i>
RPG	<i>Report Program Generator</i> (Gerador Automático de Programas).	Usada basicamente em mini-computadores, mas pode ser usada também em sistemas maiores.	<ol style="list-style-type: none"> 2. Um tanto complexa para codificar. 1. Linguagem muito simples. 2. Ideal para imprimir relatórios.

IV. AVALIAÇÃO DO SOFTWARE DE APLICAÇÃO

Existem três métodos principais para se obter o *software* de aplicação:

Fontes de Software

1. Programadores domésticos ou internos.
2. Consultores ou fornecedores externos de *software*.
3. Aplicações já prontas vendidas pelos fabricantes ou "firmas de *software*".

A Figura 6.4 resume as vantagens e desvantagens de cada fonte de *software*.

V. CARACTERÍSTICAS DO SOFTWARE DE SISTEMAS

Até agora, consideramos o aspecto do *software* relacionado à programação de aplicação. O *software* de sistemas inclui os seguintes elementos:

Software de Sistemas

- Rotinas de diagnóstico.
- Sistema operacional.
- Compiladores, montadores, intérpretes.
- Rotinas de controle de entrada/saída.
- Rotinas de controle de serviço e contabilidade.
- Geradores de relatório.
- Sistemas de administração de banco de dados.

Observe que o próprio sistema de computador opera em programas de aplicação sob o controle de um *sistema operacional*, que é uma parte importante do *software* de sistemas.

Figura 6.4
Aquisição do *software*.

<i>Método de Aquisição do Software</i>	<i>Vantagens</i>	<i>Desvantagens</i>
Programador Interno	<ol style="list-style-type: none"> 1. O programador possui um conhecimento mais profundo das necessidades do usuário. 2. As manutenções e modificações são facilmente executadas. 	<ol style="list-style-type: none"> 1. O custo do projeto e da depuração é relativamente alto. 2. O programador pode não estar muito bem informado da aplicação específica.
Fornecedores externos de <i>software</i>	<ol style="list-style-type: none"> 1. O fornecedor pode proporcionar uma avaliação objetiva das necessidades do usuário. 2. O fornecedor pode proporcionar manutenção do <i>software</i> para usuários de mini e microcomputadores que não possuem pessoal interno. 	<ol style="list-style-type: none"> 1. Mais barato que a programação doméstica (não é necessário pagar obrigações sociais etc.) mas ainda é caro. 2. Depois que o programa está completo, é difícil obter apoio para manutenção e modificação.
Pacotes de Aplicações	<ol style="list-style-type: none"> 1. O mais barato — o preço real do pacote é absorvido por vários usuários, em vez de um. 2. Os programadores têm um amplo conhecimento das aplicações específicas. 	<ol style="list-style-type: none"> 1. Nenhuma flexibilidade. 2. As modificações são extremamente caras. 3. A organização do usuário não tem controle dos procedimentos que devem ser incluídos.

1. SUPERVISOR

O *supervisor* é o programa de controle do sistema operacional. Ele coordena, correlaciona e supervisiona todos os aspectos do sistema operacional. É um elemento muito importante do *software*.

O sistema operacional completo consiste tipicamente de uma biblioteca de compiladores, montadores, intérpretes e outros programas, além do supervisor. São todos armazenados em um dispositivo de memória auxiliar, como por exemplo um disco ou um tambor, denominado *dispositivo de residência do sistema*.

O programa supervisor é introduzido na memória principal antes que qualquer processamento possa ser iniciado. Para computadores que operam 24 horas por dia o supervisor reside sempre na memória principal.

Funções do Supervisor

1. Controla o processamento de dados através de cada programa de aplicação.
2. Controla a chamada de compiladores, montadores e intérpretes para a tradução dos programas fonte.

3. Controla o processamento de solicitações do tipo pergunta-resposta.
4. Comunica as necessidades de cada operação ao operador de computador, por meio de uma máquina de escrever de console.
5. Mantém informações sobre a contabilidade do serviço, isto é, dados de custo, tempo, data e momento de cada operação.

B. COMUNICAÇÃO COM O SUPERVISOR

Uma linguagem especial orientada para a máquina, projetada para a máquina, projetada para possibilitar que os operadores, usuários e programadores comuniquem suas necessidades ao supervisor e, através dele, ao sistema operacional, denomina-se *Linguagem de Controle de Trabalho (LCT)*.* A LCT é usada inicialmente para acessar dados, programas ou outras características do sistema de computador.

C. MULTIPROGRAMAÇÃO: EXECUÇÃO DE MAIS DE UM PROGRAMA SIMULTANEAMENTE

A maioria dos sistemas de computadores mais modernos, inclusive muitos minicomputadores, tem geralmente capacidade de controle para processar mais de um programa em qualquer momento. Isto é, vários terminais podem ter acesso ao computador simultaneamente e têm a seguinte capacidade:

Multiprogramação: Execução de Mais de um Programa Simultaneamente

1. Alguns terminais podem ser usados para inserir dados sob o controle de um programa de entrada de dados.
2. Outros terminais podem ser usados para fazer perguntas sobre uma base de dados específica, sob o controle de um programa de pergunta-resposta.
3. Outros terminais e/ou unidades de E/S locais podem também ser usados para inserir, depurar e operar outros programas simultaneamente.

Visto que a velocidade de processamento dos computadores modernos é tão alta, os programas que operam em um ambiente de multiprogramação apresentam-se ao usuário como se estivessem todos sendo operados simultaneamente. Na verdade eles são intercalados, e as instruções são executadas uma de cada vez.

A multiprogramação requer um supervisor sofisticado e um sistema operacional completo, e é uma característica integrante e essencial das aplicações de comunicações de dados e de tempo compartilhado.

* Em inglês JCL (Job Control Language); doravante usaremos a abreviação portuguesa.

Memória Principal Supervisor
Partição 1 – Programa 1
Partição 2 – Programa 2
Partição 3 – Programa 3
Partição 4 – Programa 4

D. MULTIPROCESSAMENTO

Multiprocessamento é o uso de duas ou mais unidades centrais de processamento interligadas para otimizar o processamento de dados. Em um ambiente de multiprocessamento, pode-se executar mais de uma instrução ao mesmo tempo, através de duas ou mais UCP. Esta técnica é diferente da multiprogramação, onde uma UCP pode processar simultaneamente instruções diferentes de um ou mais programas, intercalando-as. Com o multiprocessamento, existe na verdade mais de uma UCP; portanto, pode-se executar, de fato, mais de uma instrução simultaneamente.

Uma aplicação típica do multiprocessamento é o uso de um minicomputador para lidar com cronogramação, formatação de dados, impressão e totais de resumo, de maneira que a UCP ou unidade principal possam ser usadas para tarefas mais complexas ou de grande prioridade.

E. MEMÓRIA VIRTUAL (MV)

A *memória virtual* faz com que um sistema de computador opere como se possuísse maior capacidade de memória primária do que realmente possui. Isto aumenta a capacidade de multiprogramação de um sistema.

Esta técnica é executada segmentando-se um programa em uma série de seções ou módulos que são armazenados fora da UCP, tipicamente em um dispositivo de acesso direto tal como disco magnético. Em vez de chamar todo o programa para a UCP para ser executado de uma só vez, o sistema de controle permite que seções do programa sejam lidas e executadas, uma de cada vez. Depois que uma seção foi executada, coloca-se outra seção na UCP e usam-se as posições de memória ocupadas pela seção anterior. Desta maneira, o computador pode operar como se tivesse uma capacidade de memória eficiente, muito superior à sua capacidade de memória real. Veja a Figura 6.5.

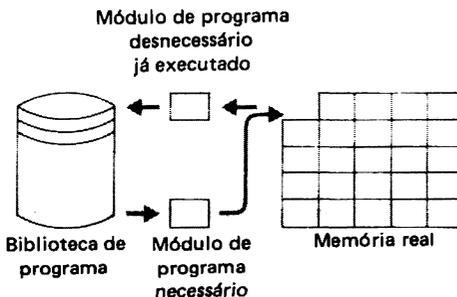


Figura 6.5
Conceito de memória virtual.

F. QUEM FORNECE O SOFTWARE DE SISTEMAS?

Antigamente, o fabricante ou agente locador de computador sempre fornecia o *software* de sistemas juntamente com o sistema de computador, como parte de um pacote. Portanto, mesmo se o usuário pudesse obter *software* de sistemas mais eficientes, de outro modo, não haveria nenhuma maneira simples de contratar a firma de computadores para fornecer somente o *hardware*. Atualmente, os fabricantes e agentes locadores têm que fornecer serviços *desvinculados*. Isto é, o usuário pode encomendar cada parte do *hardware* ou *software* individualmente.

Assim, pode-se obter um sistema de computador de um vendedor e o *software* de sistemas, para controlar as operações de *hardware*, desse mesmo vendedor ou de uma "firma de *software*", especializada somente em pacotes de *software*.

VI. SUPORTE LÓGICO INALTERÁVEL: ONDE O HARDWARE E O SOFTWARE SE INTERCALAM

Conforme mencionado, é possível adquirir, como parte de um sistema de computador, circuitos lógicos integrados, em um *chip* miniaturizado, contendo já um pacote de *software*. Tais *chips* são conhecidos como MLS, ou *Memória de Leitura Somente*.

Em sistemas pequenos onde a memória principal é limitada, o uso da MLS é bastante comum. Enquanto os programas ocupam *bytes* da memória principal, a MLS pode ser inserida no sistema, não exigindo, portanto, nenhuma memória de usuário. Conseqüentemente, a MLS pode conter características de sistema operacional que de outro modo precisariam ser programadas.

A lista abaixo representa os tipos mais comuns de MLS:

Tipos de MLS

MLS	— Memória de Leitura Somente (padrão): Um compilador BASIC, por exemplo, pode servir como MLS em sistemas de mini e microcomputadores.
MLSP*	— Memória de Leitura Somente Programável: A MLS pode ser programada pelo usuário ou vendedor. Este <i>chip</i> proporciona flexibilidade aos usuários que gostariam de produzir as suas próprias MLS.
MLSPA**	— Memória de Leitura Somente Programável e Apagável: Um <i>chip</i> é programável não somente para incluir um conjunto de funções, mas o programa pode ser "apagado" ou superposto a outro programa através de um procedimento de microcódigo.

* Em inglês PROM (Programmable Read-Only Memory); doravante usaremos a abreviação portuguesa.

** Em inglês EPROM (Erasable Programmable Read Only Memory); doravante usaremos a abreviação portuguesa.

TERMOS-CHAVE

Ada
APL
BASIC
COBOL
Compilador
Depurar
Desvinculado
Dispositivo de residência do sistema
Erro de sintaxe
Firma de *software*
Fluxograma de programa
FORTRAN
Gabarito de programação
Gabarito de programação de fluxograma
HIPO
Intérprete
Linguagem de Controle de Trabalho (LCT)
Linguagem simbólica de programação
Memória de Leitura Somente (MLS)
MLSP (PROM)
MLSPA (EPROM)
Memória Virtual (MV)
Montador
Multiprocessamento
Multiprogramação
Pascal
PL/1
Programação estruturada
Programa fonte
Programa objeto
Pseudocódigo
RPG
Software
Software de aplicação
Software de sistema operacional
Supervisor
Suporte lógico inalterável

QUESTIONÁRIO

(As respostas para as questões de números pares acompanham o teste.)

A. Questões do Tipo Verdadeiro-Falso

- | V | F | |
|-------|-------|---|
| _____ | _____ | 1. Um programa de aplicação é escrito para maximizar o uso eficiente do sistema operacional. |
| _____ | _____ | 2. O programa que controla as operações de computador é denominado supervisor. |
| _____ | _____ | 3. O programador deve sempre codificar um programa antes de escrever um fluxograma; isto minimizará os erros posteriores. |
| _____ | _____ | 4. Para ser executado, um programa tem que estar em linguagem simbólica. |
| _____ | _____ | 5. Os programas em linguagem de máquina necessitam tanto de tradução como de execução. |

- _____ 6. Se a eficiência é um fator crucial, é provável que a linguagem montadora seja usada.
- _____ 7. Uma linguagem de programação apoiada pelo Departamento de Defesa e ideal para rotinas estruturadas é a PL/1.
- _____ 8. COBOL é a linguagem de programação que será provavelmente usada para calcular a trajetória de um foguete em direção à lua.
- _____ 9. Quando um programa não tem erros de sintaxe, então está totalmente depurado.
- _____ 10. APL e BASIC são linguagens de programação interativas.

B. Preencha as Lacunas

- 1. O suporte de programa projetado para maximizar o uso eficiente do computador é denominado.....
- 2. Os programas projetados para satisfazer as necessidades de cada usuário são chamados.....
- 3. As organizações menores que não possuem seu próprio pessoal de programação contratam para escrever programas.
- 4. O sistema operacional consiste de um programa especial denominado..... que controla todas as operações de computador.
- 5. Uma representação pictórica da lógica a ser usada em um programa denomina-se
- 6. O instrumento de planejamento que enfatiza uma abordagem estruturada para a resolução de problemas de computador é um.....
- 7. Para ser executado, um programa de computador tem que estar em linguagem.
- 8. Os erros que ocorrem em um programa quando as regras são violadas são classificados como erros.....
- 9. O processo de eliminação de erros em um programa denomina-se.....
- 10. A linguagem de programação utilizada com mais frequência para mini e microcomputadores é.....

Respostas

(Fornecemos respostas somente para as questões de números pares.)

A. Questões do Tipo Verdadeiro - Falso

- 2. V
- 4. F – tem que estar em linguagem de máquina.
- 6. V
- 8. F – FORTRAN ou linguagem montadora será provavelmente usada.
- 10. V

B. Preencha as Lacunas

- 2. Programas de aplicação
- 4. Supervisor
- 6. Pseudocódigo
- 8. Sintaxe
- 10. BASIC

7

INSTRUMENTOS DE PLANEJAMENTO PARA PROGRAMADORES

- I. FLUXOGRAMA DE PROGRAMA
 - A. ELEMENTOS DE UM FLUXOGRAMA DE PROGRAMA
 - B. FLUXOGRAMAS ILUSTRATIVOS
- II. PROGRAMAÇÃO ESTRUTURADA
 - A. CONCEITOS
 - B. PSEUDOCÓDIGO: UM INSTRUMENTO USADO PARA REPRESENTAR CONCEITOS DE PROGRAMAÇÃO ESTRUTURADA
 - C. HIPO: OUTRO INSTRUMENTO DE PLANEJAMENTO ESTRUTURADO

TERMOS-CHAVE

QUESTIONÁRIO

Os três instrumentos seguintes são os mais usados para formular e descrever a lógica de um programa antes que este seja escrito:

Instrumentos para Especificar a Lógica de Programa

1. Fluxograma de programa.
2. Pseudocódigo.
3. Diagramas HIPO.

Para se aprender a programar, será extremamente útil saber como usar estes instrumentos.

I. FLUXOGRAMA DE PROGRAMA

A. ELEMENTOS DE UM FLUXOGRAMA DE PROGRAMA

Um *fluxograma de programa* é uma representação pictórica do fluxo lógico de dados que serão usados na elaboração de um programa. O fluxograma é desenhado *antes* do programa ser escrito para se ter certeza de que as instruções se integrarão de maneira lógica. Antes de codificar o programa, os usuários revisam os fluxogramas com os programadores para se assegurarem de que estes conhecem perfeitamente os requisitos do trabalho.

Um instrumento de plástico ou metal, denominado *gabarito de programação*, é usado para representar símbolos padrão em um fluxograma. A Figura 6.1 do capítulo anterior é uma ilustração de um gabarito de programação.

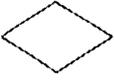
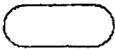
Os cinco símbolos definidos e descritos no Quadro 7.1 são os mais comumente usados em fluxogramas de programa. Cada símbolo especifica a função a ser executada.

B. FLUXOGRAMAS ILUSTRATIVOS

EXEMPLO 1

O fluxograma na Figura 7.1 indica a lógica para se imprimir em um arquivo o nome e o número de telefone de cada pessoa do sexo feminino. A Figura 7.2 descreve a saída resultante do programa codificado a partir deste fluxograma.

Quadro 7.1 – Principais Símbolos de Fluxograma

	Símbolo	Explicação
1.		<p>Símbolo de ENTRADA/SAÍDA (E/S)</p> <p>Este símbolo representa qualquer operação de entrada ou saída, como READ A NAME, READ A CARD, WRITE A LINE etc. A operação específica a ser executada é indicada por um sinal dentro do símbolo.</p>
2.		<p>PROCESSAMENTO</p> <p>Este símbolo indica qualquer processamento interno de computador, isto é, qualquer seqüência de transferência de dados ou operações aritméticas. Pode-se ter, por exemplo:</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px; text-align: center;">ADD QUANTIA AO TOTAL</div> ou <div style="border: 1px solid black; padding: 5px; text-align: center;">COMPUTE IMPOSTO = .05 X VENDAS</div> ou <div style="border: 1px solid black; padding: 5px; text-align: center;">TOTAL X + Y + Z</div> ou <div style="border: 1px solid black; padding: 5px; text-align: center;">MOVE INPUT TO OUTPUT</div> </div>
3.		<p>DECISÃO</p> <p>Este símbolo é usado para testar uma comparação lógica. É usado basicamente quando se quer que o computador faça uma pergunta. Exemplos de testes de decisão incluem:</p> <p>(a) A QUANTIDADE DE VENDAS É MAIOR QUE CR\$ 100.000,00?</p> <p>(b) A QUANTIDADE DE VENDAS É MENOR QUE A QUANTIDADE DE CRÉDITO?</p> <p>(c) CAMPO SEXUAL "M"?</p> <p>(d) TOTAL = ZEROS?</p>
4.		<p>CONECTOR</p> <p>Este símbolo denota um ponto de referência cruzada, indicando onde o fluxograma deve continuar. É usado para indicar uma alteração no fluxo normal de dados.</p>
5.		<p>TERMINAL</p> <p>Este símbolo é usado para indicar o ponto inicial ou final de um programa.</p>

Lógica de Programa

1. Ler um registro de fita.
2. Determinar se o registro se refere a uma pessoa do sexo feminino (isto é, determinar se o campo de código é igual a 2).
3. Se o campo de código não for igual a 2, então não continuar a processar este registro, pois ele contém dados para uma pessoa do sexo masculino. Em vez disto, ler o próximo registro; isto é, voltar para a etapa 1.
4. Se o registro contém dados para uma pessoa do sexo feminino (i.e., o código é igual a 2), então imprimir os seguintes campos: primeiro nome, último sobrenome, número do telefone.
5. Voltar à etapa 1 para ler o próximo registro.

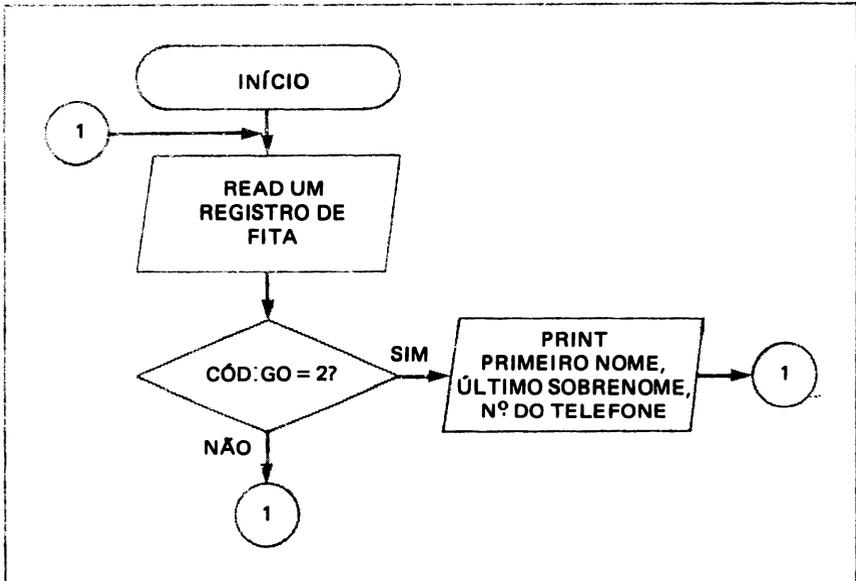


Figura 7.1
Exemplo de fluxograma.

MARIA	CUNHA	836-6210
JOANA	CODATO	697-1127
ALICE	REGO	665-3300

Figura 7.2
Amostra de saída para o Exemplo 1.

As convenções de fluxogramas são indicadas da seguinte maneira:

Convenções de Fluxograma

1. Cada símbolo indica um tipo de operação:
Entrada/saída
Processamento
Decisão
Transferência ou desvio
Terminal
2. Há uma nota escrita em cada símbolo para indicar a função específica a ser executada.

3. Os fluxogramas são lidos de alto a baixo.
4. Executa-se uma seqüência de operações até que um símbolo de terminal determine o fim da operação ou um conector de desvio transfira o controle.

EXEMPLO 2

Imprimir nome e salários semanais no registro de fita de cada empregado, onde se lê o Nome, Horas de Trabalho e Salário/Hora. O fluxograma para o Exemplo 2 está na Figura 7.3.

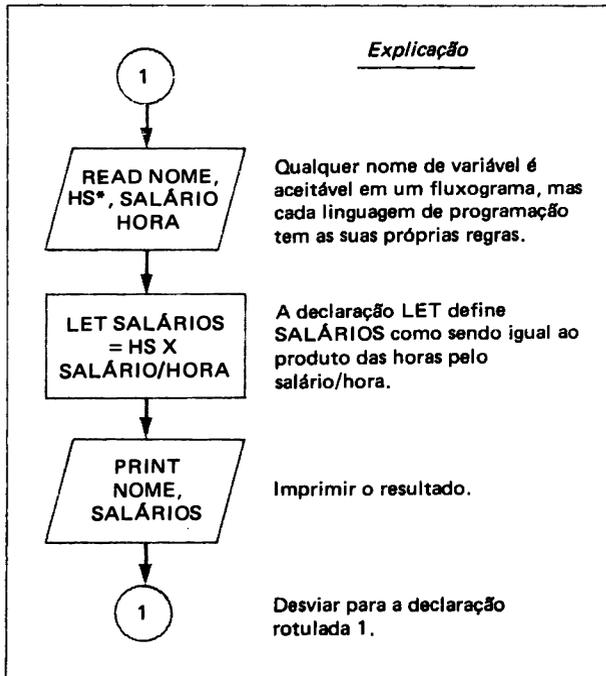
EXEMPLO 3

Vamos utilizar o problema do exemplo anterior: ao final da operação, imprimir o número total de registros processados, além do nome e salários:

- a. Como se determina o final da operação?

Em uma fita ou disco há um indicador de final de arquivo. Na ausência deste indicador, como em um terminal, usa-se um pseudo-registro ou *registro-reboque* para indicar o final do arquivo. Portanto, pode-se indicar arbitrariamente um final de arquivo como um registro de 999 H de trabalho. Visto que pode-se ter uma certeza razoável de que ninguém trabalha 999 horas, um registro com 999 significará a in-

Figura 7.3
Fluxograma para o Exemplo 2.



tenção de desviar-se uma rotina de final de arquivo onde se imprime o número total de registro.

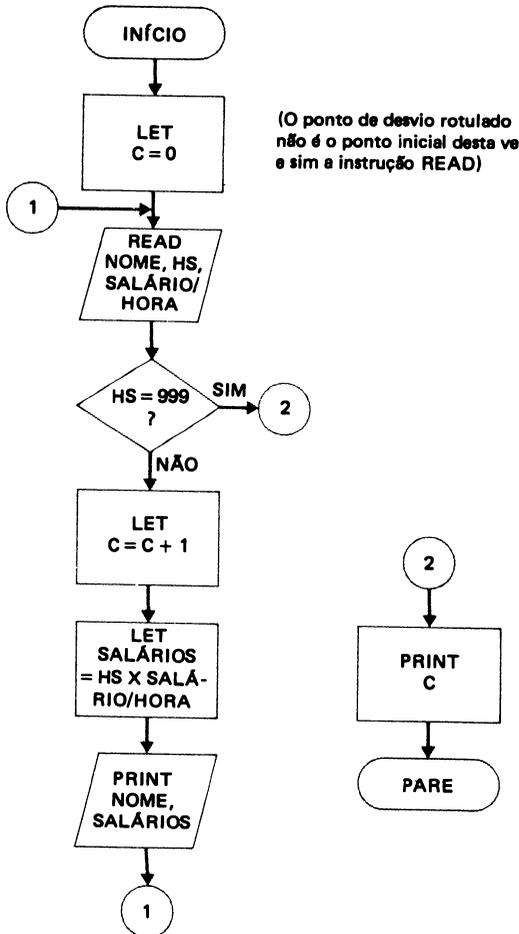
b. Como se conta o número total de registros?

Inicia-se com um campo denominado C, que servirá como contador. C deve começar em 0. Cada vez que um registro é lido, soma-se 1 a C. O símbolo de processamento poderia especificar:



O último símbolo significaria: coloque em C o valor anterior de C mais 1. Isto é o mesmo que acrescentar 1 a C. O fluxograma para o exemplo 3 está na Figura 7.4.

Figura 7.4
Fluxograma para o Exemplo 3.



EXEMPLO 4

Usando o problema acima, imprimir Nome e Salários para cada registro lido, sendo que as horas extras deverão ser contadas uma vez e meia. Isto é, para mais de 40 horas, o salário/hora pago é o salário/hora \times 1,5. Imprimir também, no início, um cabeçalho denominado RELATÓRIO DO SALÁRIO SEMANAL. O fluxograma está ilustrado na Figura 7.5.

Os seguintes símbolos são usados para determinar se deve ocorrer um desvio ou não:

Símbolos Usados na Tomada de Decisões	
<i>Símbolo</i>	<i>Significado</i>
<	Menor que
>	Maior que
=	Igual
<>	Diferente
<=	Menor ou igual a
>=	Maior ou igual a

EXEMPLO 5

Ler um arquivo com dados de registro de vendas. Para registros com valor de venda acima de Cr\$ 500.000,00, a comissão é 5% do valor das vendas. Para valor de vendas entre Cr\$ 200.000,00 e Cr\$ 500.000,00 a comissão é 2% deste valor. Para valor de vendas menor que Cr\$ 200.000,00 não há comissão. Imprimir o cabeçalho RELATÓRIO MENSAL DE VENDAS, nome e comissão para cada registro de venda. A Figura 7.6 mostra o fluxograma para este exemplo.

EXEMPLO 6

Cada registro contém a nota de um aluno. Há três alunos em cada turma. Os 10 primeiros registros são para os alunos da turma 1, os 10 registros seguintes são para os alunos da turma 2 etc. Imprimir a média de cada turma.

A maneira mais eficiente de processar os dados é através do uso de um procedimento denominado *circuito*. É ineficiente ler e somar 10 vezes e depois repetir o processo. Em vez disto estabelece-se um contador que começa em 0. A cada leitura acrescenta-se 1 ao contador. Quando o contador atingir 10, pode-se então imprimir uma média e recomeçar. Isto denomina-se circuito (Figura 7.7).

II. PROGRAMAÇÃO ESTRUTURADA

A. CONCEITOS

A *programação estruturada* é uma técnica importante usada para padronizar e melhorar programas de modo a torná-los mais fáceis de avaliar, depurar e modificar.

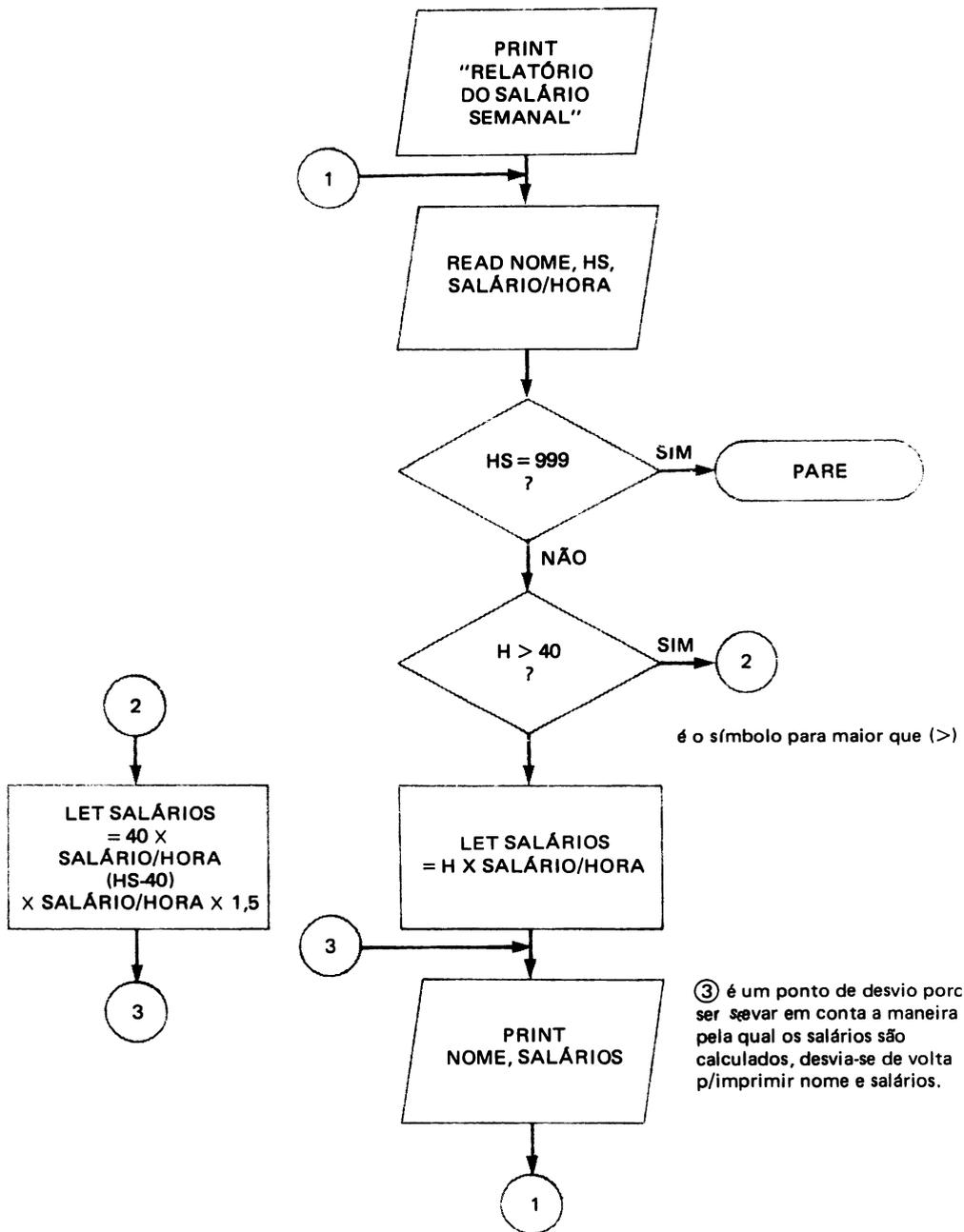


Figura 7.5
Fluxograma para o Exemplo 4.

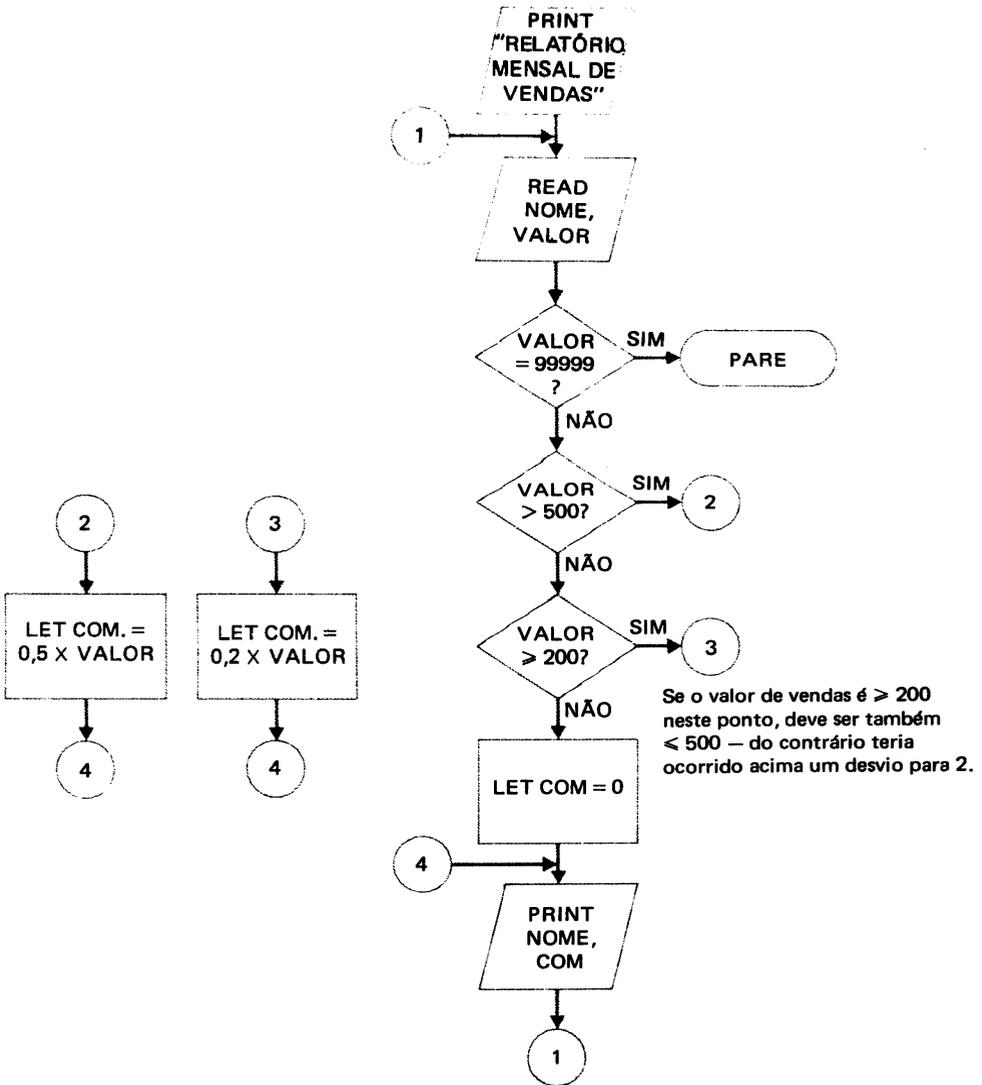


Figura 7.6 Fluxograma para o Exemplo 5.

As técnicas de programação estruturada são resumidas da seguinte maneira:

RESUMO:

1. Programação Estruturada
Uma tentativa de padronizar programas e melhorar sua confiabilidade.
2. Vantagens
 - a. Torna-se mais fácil avaliar programas.

- b. Torna-se mais fácil depurar programas.
- c. Torna-se mais fácil modificar programas já existentes.
- d. Permite-se que as equipes de programadores trabalhem juntas.

3. Características

- a. Os programas são escritos como uma série de módulos.
Cada programa consiste de blocos, segmentos ou módulos independentes.
- b. A programação denomina-se programação GO TO-menos. Evitam-se as instruções de desvio GO-TO; em vez disto, usa-se:
Circuitos DO em FORTRAN ou PL/1.
PERFORM ... UNTIL em COBOL
- c. Os programas usam uma abordagem de cima para baixo.
O módulo principal é codificado primeiro, seguido de módulos subordinados.
Todos são conectados por PERFORM ... UNTIL ou codificação de circuito DO.

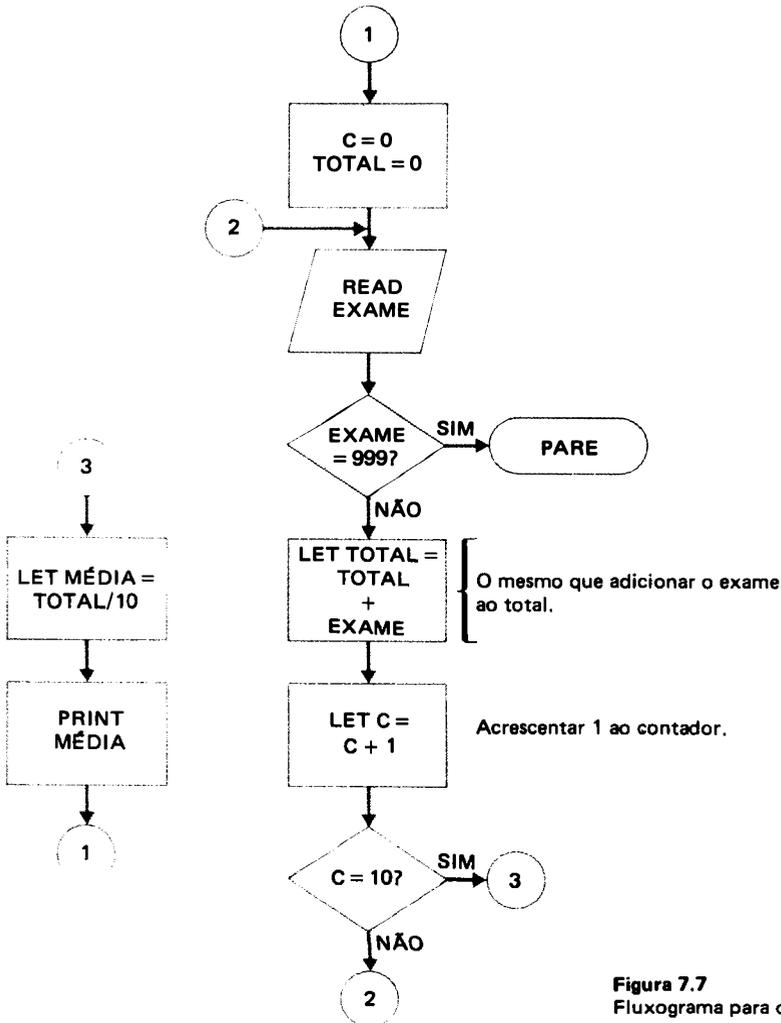


Figura 7.7
Fluxograma para o Exemplo 6.

B. PSEUDOCÓDIGO: UM INSTRUMENTO USADO PARA REPRESENTAR OS CONCEITOS DE PROGRAMAÇÃO ESTRUTURADA

Os fluxogramas não são muito úteis para programas estruturados, pois não contêm símbolos padronizados para indicar módulos ou blocos de instruções. O *pseudocódigo* é usado com mais freqüência como um instrumento de planejamento para programas estruturados.

Segue-se o pseudocódigo para imprimir todos os registros de fita:

Ilustração do Pseudocódigo

Inicializar as operações.

Ler um registro de fita; no final transferir 1 para EOF*

PERFORM ... UNTIL EOF = 1.

Transferir os dados da fita para a área de impressão.

Escrever uma linha.

Ler um registro de fita; no final transferir 1 para EOF

EXECUÇÃO FINAL

Funções de fim de trabalho

Parar de operar

Lê-se um registro de fita. O registro é processado e o próximo registro é lido sob o controle de PERFORM. Esta seqüência continua até EOF = 1, que ocorrerá quando não houver mais registro para processar.

Especificações IF THEN ELSE

Para indicar o teste de condições, usa-se o seguinte formato em um pseudocódigo:

IF (condição)

THEN (operação a ser realizada se a condição for satisfeita)

ELSE (operação a ser realizada se a condição não for satisfeita)

ENDIF

C. HIPO: OUTRO INSTRUMENTO DE PLANEJAMENTO ESTRUTURADO

HIPO é uma abreviação para Hierarchy plus Input-Process-Output.** Assim como os fluxogramas, o HIPO é usado para representar graficamente um sistema ou pro-

*Em inglês EOF (End of File MARK = MARCA de Fim de Arquivo). Conservaremos a abreviação inglesa por ser esta a usada pelo computador.

** Em português Hierarquia/entrada-processo-saída. Manteremos a abreviação inglesa por ser a adotada universalmente.

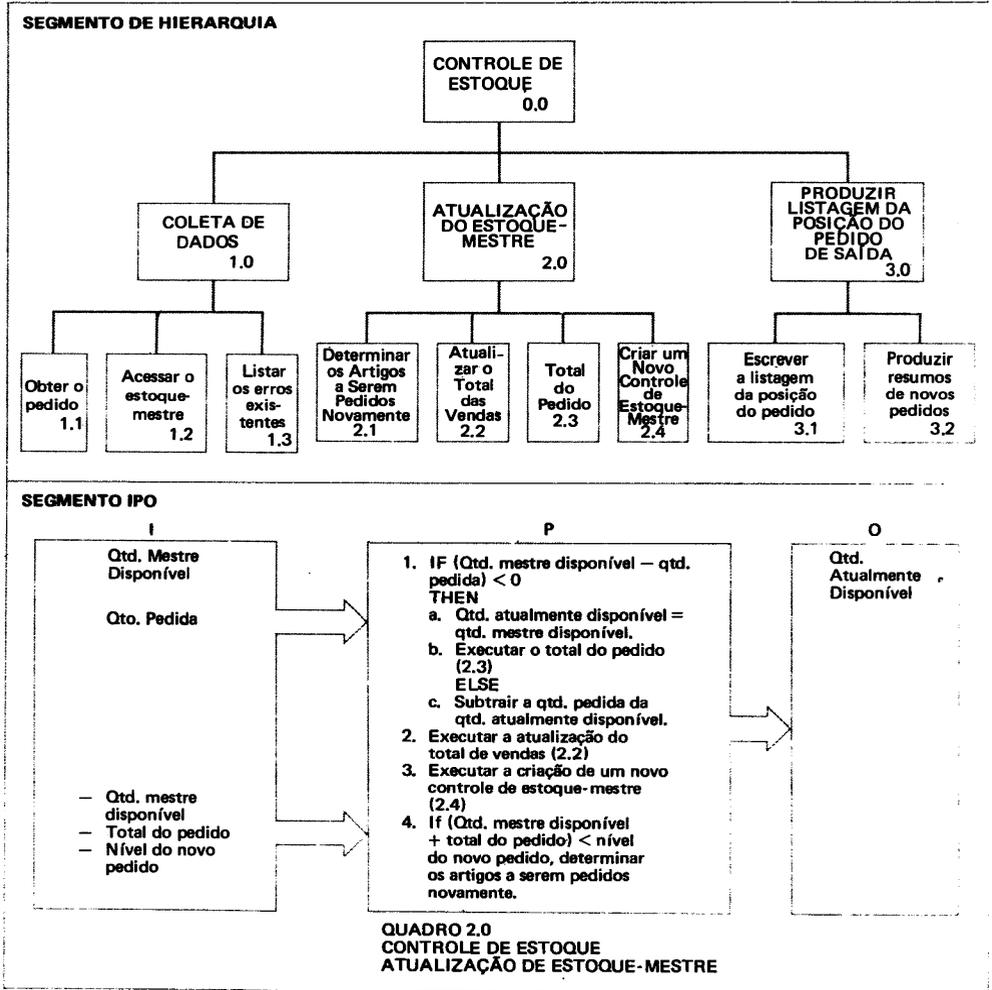


Figura 7.8
 Exemplo de diagrama HIPO

grama. Mas por ser hierárquico, o HIPO é mais adequado para representar programação estruturada e técnicas de projeto de sistemas. A Figura 7.8 ilustra a hierarquia e os segmentos do processo de entrada e saída de um diagrama HIPO.

TERMOS-CHAVE

Circuito
 Circuito DO

Codificação GO TO-menos
Desvio
Diagrama HIPO
Fluxograma de programa
Gabarito de programação
PERFORM ... UNTIL
Programação estruturada
Pseudocódigo
Registro-reboque

QUESTIONÁRIO

(As respostas para as questões de números pares acompanham o teste.)

A. Questões do Tipo Verdadeiro-Falso

V F

- _____ 1. Os fluxogramas são lidos da esquerda para a direita, a não ser que ocorra um desvio.
- _____ 2. Os fluxogramas devem ser completados depois que o programa estiver escrito.
- _____ 3. Se um programador tem uma concepção equivocada sobre algum aspecto do programa, a representação de um fluxograma pode ajudá-lo a esclarecer tal concepção.
- _____ 4. Em geral, os programas estruturados são mais fáceis de depurar do que os tradicionais.
- _____ 5. É mais fácil avaliar a eficiência de um programa estruturado do que de um tradicional.
- _____ 6. Um fluxograma é considerado o melhor instrumento para representar a lógica em um programa estruturado.
- _____ 7. Usa-se geralmente um gabarito de programação na elaboração de um pseudocódigo.
- _____ 8. Os diagramas HIPO são de estrutura hierárquica.
- _____ 9. Ao desenhar um fluxograma é aconselhável incluir um teste para fim de arquivo.
- _____ 10. O circuito é um procedimento usado para executar uma série de operações um determinado número de vezes.

B. Preencha as Lacunas

- Um é um diagrama ou representação pictórica do fluxo lógico de dados em um programa.
- Um instrumento especial de plástico ou metal denominado é usado para representar os símbolos em um fluxograma.
- Escreve-se uma dentro de cada símbolo do fluxograma para indicar a função específica a ser executada.
- A todo conector de desvio deve corresponder um conector
- HIPO é uma abreviação para
- Os dois segmentos de um diagrama HIPO são o segmento e o segmento
- Outro termo para programação estruturada é

8. Os programadores devem trabalhar bem próximos aos no projeto de fluxogramas; isto reduzirá erros posteriores.
9. Para possibilitar que cada módulo funcione como uma entidade autônoma em um programa estruturado, as instruções devem ser evitadas.
10. Um instrumento usado para representar a lógica em um programa estruturado é

Respostas

(Fornecemos respostas somente para as questões de números pares.)

A. Questões do Tipo Verdadeiro-Falso

2. F — deve ser escrito antes do programa.
4. V.
6. F — o pseudocódigo é melhor.
8. V.
10. V.

B. Preencha as Lacunas

2. Gabarito de programação
4. Conector de entrada
6. Hierarquia/entrada-processo-saída
8. Usuários ou analistas de sistemas
10. Pseudocódigo

8

PROGRAMAÇÃO EM BASIC

- I. SUMÁRIO DA LINGUAGEM BASIC
- II. INTERAÇÃO COM UM SISTEMA DE COMPUTADOR USANDO UM TERMINAL
 - A. INÍCIO DE COMUNICAÇÃO
 - 1. Sistemas Fixos ou de Discagem Manual
 - 2. Etapas para Iniciar a Comunicação
 - B. TRANSMISSÃO À UCP
 - C. CORREÇÃO DE ERROS DE TIPOGRAFIA
 - 1. Retrocesso
 - 2. Correção ou Supressão de uma Linha
 - 3. Inserção de uma Linha em um Programa BASIC
 - D. EXECUÇÃO DE UM PROGRAMA
 - E. LISTAGEM DE UM PROGRAMA
 - F. CONSERVAÇÃO DE UM PROGRAMA
- III. ELEMENTOS ESSENCIAIS DE UM PROGRAMA BASIC
 - A. DESCRIÇÃO GERAL
 - B. REGRAS FUNDAMENTAIS PARA PROGRAMAS BASIC
 - C. HIERARQUIA DAS OPERAÇÕES ARITMÉTICAS
 - D. DECLARAÇÕES CONDICIONAIS
 - 1. Formato
 - 2. Exemplos de Procedimentos Lógicos de Controle
 - E. VARIAÇÕES SOBRE UM TEMA
 - 1. Opções de Impressão
 - 2. Inserção de Dados Alfanuméricos Denominados Variáveis de Série
 - 3. Outros Usos da Declaração LET
 - F. DECLARAÇÃO REM
 - G. SUMÁRIO
 - TESTE DE AUTO-AVALIAÇÃO
 - H. DECLARAÇÕES READ E DATA
 - I. DECLARAÇÕES FOR ... NEXT
 - J. MATRIZES
- IV. TÓPICOS AVANÇADOS EM BASIC
 - A. OPÇÃO DE TABULAÇÃO
 - B. DECLARAÇÃO PRINT USING
 - 1. Alinhamento de Campos Numéricos
 - 2. Inserção de Vírgulas em Campos Numéricos
 - 3. Inserção de Sinais de Dólar em Campos Numéricos
 - 4. Uso de um Sinal de Dólar Flutuante

- 
- 5. Impressão de Números com Asterisco
 - 6. Impressão de Sinais Negativos
 - 7. Impressão de Variáveis e Constantes de Série
 - 8. Utilização de Mais de Uma Expressão de Série em Uma Declaração
PRINT USING
 - 9. Uso de Ponto-e-vírgula para Separar Itens em uma Declaração PRINT

C. FUNÇÕES DE BIBLIOTECA

D. USO DE RND EM BASIC

E. SUB-ROTINAS

F. FUNÇÕES DE SÉRIE

G. CLASSIFICAÇÃO DE DADOS EM BASIC

V. VERSÕES DE BASIC

VI. SUMÁRIO DA LINGUAGEM BASIC

TERMOS-CHAVE

QUESTIONÁRIO

PROGRAMAS PRÁTICOS EM BASIC

I. SUMÁRIO DA LINGUAGEM BASIC

Conforme observado, BASIC é a linguagem de programação mais popular para cursos introdutórios de computador, sobretudo porque é fácil de aprender e codificar. A lista que se segue é um resumo das características do BASIC, conforme o que foi discutido no Capítulo 6.

RESUMO

1. BASIC é uma abreviação para Beginner's All-purpose Symbolic Instruction Code (Código de Instruções Simbólicas de Caráter Geral para Principiantes).
2. BASIC é geralmente inserida em um sistema de computador através de um terminal (ou em um sistema de mini ou microcomputador) de um modo interativo.
3. BASIC é uma linguagem relativamente simples de aprender e codificar.
4. Embora um programa BASIC seja, na maioria das vezes, inserido em um terminal de um computador através de um teclado, pode ser também inserido em cartões perfurados ou fitas, numa modalidade de processamento por lote, e depois compilado, se for o caso.
5. Todos os programas têm que ser primeiro codificados e depois traduzidos. Estes programas têm que ser então testados com dados de amostra para garantir a sua precisão.
6. Em BASIC, os dados de teste, como as instruções, são geralmente inseridos em um terminal, embora possam ser inseridos através de fita ou disco.

A versão padrão de BASIC pode ser executada em qualquer computador. Existem várias versões de BASIC ampliadas, que fornecem maiores recursos mas não são padronizadas. Enfatizaremos a versão padronizada.

II. INTERAÇÃO COM UM SISTEMA DE COMPUTADOR USANDO UM TERMINAL

Na verdade, acessar um sistema de mini ou microcomputador é simplesmente ligar o dispositivo. Nesse momento, pode-se começar a codificar o programa. Para acessar um sistema maior com o uso de um terminal, necessita-se de um procedimento de "*início de comunicação*". Os leitores que estão usando um mini ou microcomputador podem passar por cima da Seção A, que considera os procedimentos de início de comunicação.

A. INÍCIO DE COMUNICAÇÃO

1. Sistemas Fixos ou de Discagem Manual

Um terminal é ligado a uma UCP em local distante, de forma direta, por meio de um cabo, ou pode também acessar a UCP usando um serviço telefônico de discagem manual. No primeiro caso, o usuário simplesmente liga o terminal e está apto a iniciar a comunicação.

No segundo caso, os usuários têm que receber um número de telefone para que possam "discar" o sistema. O telefone é geralmente colocado em um acoplador

acústico de maneira que os sinais digitais provenientes do computador e dirigidos ao computador possam ser convertidos em sinais telefônicos. Depois de discar para o computador, o usuário ouvirá um sinal agudo, indicando que o computador está pronto. Se o usuário ouvir um sinal de ocupado, deve discar novamente.

2. Etapas para Iniciar a Comunicação

Uma vez obtido o acesso através de um terminal, deve-se indicar o seguinte:

1. O código de autorização.
2. O objetivo de escrever um programa BASIC.

Estes procedimentos fazem parte do *controle de trabalho* e dependem do *protocolo* estabelecido na instalação onde tais procedimentos são executados.

A seguir, um exemplo típico de um procedimento de início de comunicação:

```
ZE222 PLEASE LOGON
/LOGON CSA010,A2129
% E223 LOGON ACCEPTED FOR TEN 2202, ON 04/11/81 AT 1434, LINE #050.
***HOFSTRA UNIVERSITY COMPUTING FACILITY***
***UNIVAC 90/60 MOD 2 ---VS/9 VER 3.5***
/EXEC BASIC
% F500 LOADING VER# 009 OF BASIC.
BASIC 09, NEW OR OLD
*NEW
NEW PROGRAM NAME---STERN
READY
```

NOTA: As entradas colocadas dentro de retângulos na ilustração são fornecidas pelo usuário.

B. TRANSMISSÃO À UCP

A maioria dos terminais e microcomputadores têm um teclado padrão para digitar instruções. A impressão destas instruções e as respostas do computador variam; em alguns casos teremos uma exibição visual, em outras uma impressão digitada.

O programador digita qualquer mensagem ou qualquer instrução BASIC e depois pressiona uma tecla de controle para indicar o fim da linha. Geralmente esta é uma tecla RETURN. Entretanto, em alguns sistemas o usuário tem que pressionar tanto a tecla CONTROL como a letra C, por exemplo, para indicar o fim da linha.

A linha será transmitida à UCP somente depois que a tecla de controle for pressionada.

C. CORREÇÃO DE ERROS DE TIPOGRAFIA

É bastante possível que ocorram erros de tipografia durante os procedimentos de início de comunicação ou durante a própria codificação do programa BASIC. Em ambos os casos, os mesmos métodos podem ser usados para corrigir estes erros. Os procedimentos de correção de erro incluem as seguintes técnicas:

1. Retrocesso

Se uma tecla foi pressionada erradamente, é possível apagar um caractere ou retroceder antes que a linha seja transmitida à UCP.

Como o Retrocesso é Executado em Alguns Sistemas

1. DEL — pressionando a tecla DEL em um teclado, apaga-se o caractere totalmente.
2. Tecla (←) — cada seta retrocede um caractere.
3. Tecla @ — @ retrocede um caractere, por exemplo:
LOGG @ ON é transmitido como LOG ON.

2. Correção ou Supressão de Uma Linha

Suponha uma situação em que só se percebe um erro cometido depois de transmitir uma linha à UCP. Isto é, já se pressionou a tecla de controle. Neste caso deve-se simplesmente reinserir a linha toda.

3. Inserção de uma Linha em um Programa BASIC

Todas as instruções em BASIC necessitam de números de linha. Se se omitiu uma instrução inadvertidamente, é possível inseri-la, mesmo fora de seqüência.

Suponha que, ao se inserir um programa, as seguintes linhas foram digitadas:

```
20 LET D = A + B - C
30 PRINT D
```

A linha 10, que leria valores para A, B e C, foi omitida inadvertidamente. A linha omitida (linha 10) pode ser digitada diretamente depois da linha 30, mas ela deve conter o seu número de linha apropriado indicando a sua localização exata na seqüência de instruções. Portanto, a seqüência seguinte executará as instruções na ordem do número de linha; isto é, a linha 10 será executada em primeiro lugar, embora esteja colocada por último.

```
20 LET D = A + B - C
30 PRINT D
10 INPUT A, B, C
```

O computador executa automaticamente as declarações BASIC em seqüência pelos seus números de linha, sem levar em conta a ordem na qual elas são digitadas.

Numerando as instruções de 10 em 10 (10, 20, 30 etc.), torna-se possível fazer inserções, caso seja necessário.

D. EXECUÇÃO DE UM PROGRAMA

Depois de inserido o programa, ele deve ser testado ou depurado. Usam-se dados de amostra para determinar se a lógica do programa produz resultados corretos.

Para operar ou executar um programa, digitar:

RUN

EXEMPLO

```
10 INPUT A, B, C
20 LET D = A + B - C
30 PRINT D
40 GO TO 10
50 END
RUN
?1, 2, 3
0
?2, 2, 1
3
```

E. LISTAGEM DE UM PROGRAMA

O programa pode ser difícil de ler, caso tenha havido alterações ou inserções no momento da entrada. Digitando um comando LIST, a listagem impressa será "limpa".

```
10 PRRN@@@INT "ALÔ"
20 NND@@@END
LIST
10 PRINT "ALÔ"
20 END
RUN
ALÔ
```

F. CONSERVAÇÃO DE UM PROGRAMA

Depois que o programa foi inserido, o programador pode querer conservá-lo para processamento posterior. Para isto, aperta-se geralmente no comando SAVE com o nome do programa.

SAVE TESTE 1

Deste modo pode-se chamar este programa, na medida do necessário, digitando posteriormente um comando de sistema como:

LOAD TESTE 1

Observe mais uma vez que alguns destes comandos terão uma pequena variação de um sistema para outro.

III. ELEMENTOS ESSENCIAIS DE UM PROGRAMA BASIC

A. DESCRIÇÃO GERAL

Vamos iniciar examinando algumas instruções simples em BASIC:

Instruções Simples Codificadas em BASIC		
<i>Tipo de instrução</i>	<i>Formato BASIC</i>	<i>Exemplo</i>
Entrada	INPUT (nomes de campo)	INPUT A, B, C
Processamento	LET (nomes de campo) = (expressão aritmética)	LET D = A + B + C
Saída	PRINT (nome de campo)	PRINT D

Na instrução descrita acima, lêem-se no computador três números denominados A, B e C. Somam-se estes números usando uma declaração LET, e imprime-se o resultado da soma, denominado D. A, B, C e D são chamados *variáveis de nomes de campo*. Usando as instruções acima, podem-se inserir quaisquer números para A, B e C e obter os resultados corretos ou soma, em D. Os campos variáveis podem conter valores inteiros ou decimais. Portanto, 10, 36 e -5.2 seriam entradas válidas. Entretanto, as variáveis não podem conter uma vírgula. Assim, 1000 é uma variável válida mas 1,000 não é.

Observe, porém, que raramente se escreve um programa de computador para operar somente com um conjunto de dados. Os computadores geralmente utilizam grandes volumes de dados para processamento.

Dessa maneira, normalmente se repetirá a seqüência das etapas acima não somente para um grupo de três números, mas para vários grupos de números. Para isto, deve-se colocar um número de linha em cada instrução. Depois da impressão, o computador é simplesmente instruído para repetir ou GO TO o número de linha associado à primeira instrução.

```

10 INPUT A, B, C
20 LET D = A + B + C
30 PRINT D
40 GO TO 10

```

Na maioria dos computadores, os programas BASIC têm que terminar com uma declaração END para informar ao computador que não há mais instruções. Portanto, o primeiro programa BASIC completo aparece conforme se segue. Lembre-se de que são utilizados números de linhas múltiplos de 10, no caso de ser necessário fazer inserções.

```

10 INPUT A, B, C
20 LET D = A + B + C
30 PRINT D
40 GO TO 10
50 END

```

Para se executar o programa no computador, aperta-se a tecla RUN. Eis uma amostra dos resultados possíveis:

```

10 INPUT A, B, C
20 LET D = A + B + C
30 PRINT D
40 GO TO 10
50 END

```

RUN
 ? 1,2,3
 6
 ? 2,2,2
 6
 ? 0,1,2
 3
 ? STOP

A palavra STOP determina um encerramento de trabalho. Os programas BASIC bem projetados incluem um teste para o final dos dados. Isto é, geralmente não é aconselhável determinar um encerramento de operação digitando "STOP", como se fez acima, embora isto funcione na maioria das vezes. Depois que o leitor se tornar mais familiarizado com a linguagem BASIC, incluiremos testes apropriados para final de dados.

B. REGRAS FUNDAMENTAIS PARA PROGRAMAS BASIC

Regras													
1. Cada instrução deve começar com um número de linha. (Os números de linhas são tipicamente codificados em múltiplos de 10; isto é, 10, 20, 30 etc. são usados para permitir inserções no caso de uma instrução ser omitida inadvertidamente.)													
2. As instruções são executadas em seqüência de número de linha, a menos que se encontre um GO TO.													
3. Todos os programas terminam com uma declaração END.													
4. Os nomes de variáveis numéricas são representados por uma letra ou uma letra seguida de um dígito (por ex.: A, A1, B6 etc.).													
5. As operações aritméticas são codificadas em uma declaração LET com os seguintes símbolos:													
<table border="1"> <thead> <tr> <th>Símbolo usado</th> <th>Operação</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">+</td> <td>Adição</td> </tr> <tr> <td style="text-align: center;">-</td> <td>Subtração</td> </tr> <tr> <td style="text-align: center;">*</td> <td>Multiplicação</td> </tr> <tr> <td style="text-align: center;">/</td> <td>Divisão</td> </tr> <tr> <td style="text-align: center;">** (ou ↑)</td> <td>Potenciação</td> </tr> </tbody> </table>		Símbolo usado	Operação	+	Adição	-	Subtração	*	Multiplicação	/	Divisão	** (ou ↑)	Potenciação
Símbolo usado	Operação												
+	Adição												
-	Subtração												
*	Multiplicação												
/	Divisão												
** (ou ↑)	Potenciação												
6. As constantes numéricas podem ser usadas em declarações aritméticas:													
10 LET D = .05 * C													
7. Os desvios incondicionais são codificados com declarações GO TO.													

Usando as sete regras básicas descritas acima, pode-se codificar uma enorme variedade de programas.

EXEMPLO 1

Para uma seqüência de dados de entrada que indicam temperaturas em graus Celsius, escrever um programa para calcular as temperaturas em graus Fahrenheit.

$$\text{Fahrenheit} = 9/5 \text{ Celsius} + 32$$

```
10 INPUT C
20 LET F = 9/5 * C + 32
30 PRINT F
40 GO TO 10
50 END
```

Os valores 9/5 e 32 são considerados constantes numéricas. Visto que a fração 9/5 poderia ser escrita como 1.8, a linha 20 acima também poderia ser codificada da seguinte maneira:

```
20 LET F = 1.8 * C + 32
```

EXEMPLO 2

Para cada campo de entrada representando um TOTAL, calcular um PREÇO que permita um desconto de 3%.

```
10 INPUT T
20 LET F = T - .03 * T
30 PRINT F
40 GO TO 10
50 END
```

Observe que qualquer quantidade T menos .03 de T = .97T. Portanto, a linha 20 poderia ser codificada assim:

```
20 LET F = .97 * T
```

EXEMPLO 3

Serão inseridos dois campos de entrada, representando HORAS DE TRABALHO e SALÁRIO/HORA. Calcular os SALÁRIOS TOTAIS dos campos de entrada para cada grupo.

```
10 INPUT H, R
20 LET W = H * R
30 PRINT W
40 GO TO 10
50 END
```

O usuário deste programa inserirá os dados para os quais se imprimirá a informação de saída. Quando não houver mais dados, o usuário digitará "STOP".

E/S para o Problema Anterior

?40,2
80
?30, 3.50
105
?50, 4
200
?STOP

Observe novamente que o uso de "STOP", neste caso para determinar um fim de operação, é uma simplificação. Posteriormente, serão incluídos testes apropriados de final de dados.

Note que BASIC é uma linguagem de estilo livre, o que significa que é permissível qualquer espaço entre palavras e expressões. Esta liberdade é uma grande vantagem que a linguagem BASIC tem sobre outras como COBOL, por exemplo.

C. HIERARQUIA DAS OPERAÇÕES ARITMÉTICAS

Considere o seguinte:

```
10 INPUT B, C, D
20 LET A = B - C * D
30 PRINT A
40 GO TO 10
50 END
```

Suponha que $B = 10$, $C = 5$, $D = 2$. Uma questão que precisa ser resolvida é se o computador executará:

$$B - C * D = 10 - 5 \times 2$$

como

$$(10 - 5) \times 2 = 10$$

ou

$$10 - (5 \times 2) = 0$$

É evidente que se o computador executar *primeiro* a operação de subtração e depois a de multiplicação, o resultado será diferente do resultado em que a multiplicação é executada primeiro.

A lista seguinte representa as regras hierárquicas usadas por computadores.

Ordem das Operações

1. Potenciação.
2. Multiplicação e divisão.
3. Adição e subtração.

Se uma instrução tem duas ou mais operações no mesmo nível, estas operações são executadas seqüencialmente, da esquerda para a direita.

Portanto, na seguinte instrução:

```
20 LET A = B - C * D
```

executa-se primeiro a multiplicação e depois a subtração. Isto é, temos:

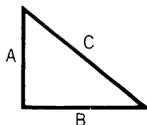
$$10 - 5 * 2 = 10 - 10 = 0$$

As regras tradicionais de hierarquia podem ser omitidas com o uso de parênteses. Assim, para obter-se $B - C$ multiplicado por D , codifica-se:

```
20 LET A = (B - C) * D
```

EXEMPLO

Determinar o valor da hipotenusa de um triângulo retângulo, em que os outros dois lados são inseridos como entrada.



Lembre-se de que C é igual à raiz quadrada de $A^2 + B^2$. As raízes quadradas podem ser calculadas em BASIC elevando-se uma expressão à potência $1/2$ ou $0,5$, ou usando-se a função especial SQR.

```
10 PRINT "INSERIR DOIS LADOS DE UM RETÂNGULO"  
20 PRINT "FAREI A COMPOSIÇÃO DA HIPOTENUSA"  
30 PRINT  
40 INPUT A, B  
50 LET C = (A ** 2 + B ** 2) **.5  
60 PRINT "A HIPOTENUSA É", C  
70 PRINT  
80 GO TO 40  
90 END
```

O uso de parênteses é necessário para se obter a ordem apropriada das operações. A linha 50 também poderia ser substituída usando-se a função SQR:

```
50 LET C = SQR (A ** 2 + B ** 2)
```

Outras funções serão consideradas posteriormente.

EXEMPLO

Ler três conceitos de prova E para cada aluno de uma turma e depois calcular a média.

```
10 PRINT "INSERIR TRÊS CONCEITOS DE PROVA"  
20 PRINT  
30 INPUT E1, E2, E3  
40 LET A = (E1 + E2 + E3)/3  
50 PRINT "A MÉDIA É ", A  
60 GO TO 20  
70 END
```

Aqui, os parênteses são novamente necessários para a ordem correta da avaliação. A instrução seguinte está incorreta:

CODIFICAÇÃO INCORRETA:

```
40 LET A = E1 + E2 + E3/3
```

Somente E3 seria dividido por 3 se os parênteses fossem omitidos.

Em resumo, quando se desejar que uma operação ou uma seqüência de operações seja executada primeiro, deve-se colocá-la entre parênteses.

EXEMPLO

Ler um valor de capital, uma determinada taxa de juros e vários períodos de investimento.

Computar o capital e os juros ganhos após N períodos de investimento, usando a seguinte fórmula:

$$C(1 + J)^N = \text{capital} + \text{juro ganho}$$

```
10 INPUT C1, N, J  
20 LET C = C1 * (1 + J) ** N  
30 PRINT C  
40 GOTO 10  
50 END
```

D. DECLARAÇÕES CONDICIONAIS

1. Formato

Um aspecto importante da programação consiste em testar condições específicas. Pode-se querer calcular uma comissão para vendedores com vendas totais superiores a Cr\$ 100.000,00, ou conceder descontos a fregueses especiais, ou então verificar se o arquivo de entrada chegou ao fim.

A declaração IF-THEN é usada para testar condições específicas. Seu formato é:

Formato da Declaração IF-THEN

IF (condição) THEN (número da declaração para a qual se deverá desviar)

EXEMPLO

IF A = 15 THEN 60

significa "se A é igual a 15 vá para o número de linha 60".

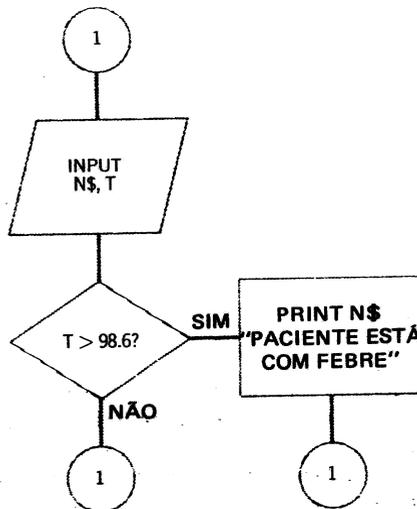
As condições que podem ser especificadas em uma declaração IF THEN são as seguintes:

Símbolos Usados em Declarações IF-THEN	
Símbolo	Significado
=	Igual a
<	Menor que
>	Maior que
≤	Menor ou igual
≥	Maior ou igual
<>	Diferente (menor ou maior que)

EXEMPLO

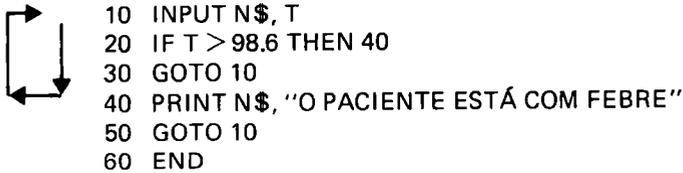
Suponha que seja necessário ler nomes de pacientes e imprimir a mensagem "O PACIENTE ESTÁ COM FEBRE" para pacientes com temperaturas acima de 98.6. Se a temperatura não está acima de 98.6, não se imprime nada.

O fluxograma e o programa para este problema são os seguintes:

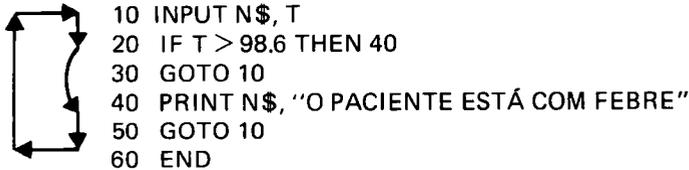


```
10 INPUT N$, T
20 IF T > 98.6 THEN 40
30 GOTO 10
40 PRINT N$, "O PACIENTE ESTÁ COM FEBRE"
50 GOTO 10
60 END
```

Observe que existem duas seqüências possíveis de serem executadas. Para pacientes com temperaturas de 98.6 ou menos, temos:



Para pacientes com temperaturas acima de 98.6, temos:



Sem alterar o resultado, a declaração acima poderia ter sido codificada da seguinte maneira:

```
10 INPUT N$, T
20 IF T <= 98.6 THEN 10
30 PRINT N$, "O PACIENTE ESTÁ COM FEBRE"
40 GOTO 10
50 END
```

As declarações IF-THEN fornecem ao programador bastante flexibilidade e capacidade de controle lógico. Uma dessas capacidades é o *circuito* onde se executa um determinado conjunto de operações até que uma condição específica seja atingida.

As declarações IF-THEN são comumente usadas para testar um fim de arquivo. Quando se inserem dados em um terminal usa-se um reboque ou pseudo-registro para indicar o final do arquivo. Suponha que seja necessário imprimir o número total de contas processadas em um determinado dia. O extrato do programa abaixo é um bom começo.

```
10 INPUT A
20 LET C = C + 1
30 GO TO 10
```

Este extrato de programa servirá como entrada de um campo denominado A, que consiste de um número de conta, e acumulará em C a quantidade total de números de conta lidos. A instrução $C = C + 1$ é usada para acrescentar 1 a C. Mas como se consegue imprimir C depois que todos os registros forem processados? Pode-se instruir o operador de terminal para inserir um número de conta fictício, 99999, para indicar o final do arquivo. O programa completo seria então o seguinte:

```
10 INPUT A
20 IF A = 99999 THEN 50
30 LET C = C + 1
40 GO TO 10
50 PRINT C
60 END
```

2. Exemplos de Procedimentos Lógicos de Controle

EXEMPLO

Calcular a soma de todos os números inteiros de 1 a 100. Neste caso, não será necessário nenhuma declaração INPUT, pois já conhecemos as variáveis a serem processadas.

À primeira vista, o seguinte programa poderia parecer adequado:

```
10 LET T = 1 + 2 + 3 + ... + 100
20 PRINT T
30 END
```

Observe, entretanto, que as reticências (...) teriam que ser substituídas pelos números inteiros reais e, na melhor das hipóteses, esta seria uma tarefa cansativa.

Evidentemente, o mais adequado seria o seguinte:

<i>Declaração</i>	<i>Significado</i>
10 LET N = 1	Iniciar N em 1.
20 LET S = S + N	ADD N a S.
30 LET N = N + 1	Acrescentar 1 a N.
40 IF N <= 100 THEN 20	Se N é menor ou igual a 100, repetir.
50 PRINT S	Imprimir S quando N for maior que 100.
60 END	

Na declaração acima, N varia de 1 a 100 e S é o nome de variável para o acumulador que vai reter a soma dos N números inteiros. Repete-se o procedimento até $N > 100$. O termo usado para este processamento é denominado *circuito*.

Portanto, temos:

<i>Número de Vezes da Sequência</i>	<i>Valor de N</i>	<i>Valor de S</i>
0	1	0 (todas as variáveis iniciam em 0)
1	1	1
2	2	1 + 2
3	3	1 + 2 + 3
⋮	⋮	
100	100	1 + 2 + 3 + ... + 100

Seria possível ter codificado o circuito de soma de outras maneiras. Por exemplo:

```
10 LET N = 1
20 LET S = S + N
30 LET N = N + 1
40 IF N > 100 THEN 60
50 GOTO 20
60 PRINT S
70 END
```

Este procedimento de circuito possibilita a execução de uma enorme variedade de operações semelhantes, com pequenas modificações.

EXEMPLO

Somar todos os números inteiros de 100 a 2000.

```
10 LET N = 100
20 LET S = S + N
30 LET N = N + 1
40 IF N <= 2000 THEN 20
50 PRINT S
60 END
```

EXEMPLO

Somar todos os números ímpares de 1 a 1001. Neste caso o número inteiro, N, começa em 1 e é aumentado em 2 a cada vez.

```
10 LET N = 1
20 LET S = S + N
30 LET N = N + 2
40 IF N <= 1001 THEN 20
50 PRINT S
60 END
```

Os circuitos podem ser usados para muitos tipos diferentes de operações. Examinaremos alguns destes tipos.

EXEMPLO

Ler um valor para N. Computar N!, onde $N! = N \times (N - 1) \times (N - 2) \dots \times 1$. (N! é denominado N fatorial.)

Por exemplo:

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

```
10 LET F = 1
20 INPUT N
30 LET F = F * N
40 LET N = N - 1
50 IF N >= 1 THEN 30
60 PRINT F
70 END
```

Suponha que $N = 4$; a seqüência do processamento seria a seguinte:

<i>Número de Vezes da Declaração 30</i>	<i>Valor de F</i>	<i>Valor de N</i>
1	1	4
2	4	3
3	4 × 3	2
4	4 × 3 × 2	1
5	4 × 3 × 2 × 1	

$F = 4 \times 3 \times 2 \times 1 = 24$ quando impresso

As operações de circuito geralmente incluem os elementos mencionados abaixo. Observe que a palavra "LET" é opcional em uma declaração aritmética.

EXEMPLO	
<i>Somar os Números Ímpares de 1 a 101</i>	<i>Elementos em um Circuito</i>
10 N = 1	1. Iniciar o campo.
20 S = S + N	2. Executar as operações solicitadas.
30 N = N + 2	3. Modificar o campo cada vez que passar por ele.
40 IF N <= 101 THEN 20	4. Testar o campo: se a operação foi executada o número necessário de vezes, o trabalho está pronto; se não, repetir.

EXEMPLO

Calcular salários semanais com base na entrada de SALÁRIO/HORA e HORAS DE TRABALHO. Os empregados que trabalharam mais de 40 horas devem receber uma vez e meia por hora extra.

```

10 INPUT H, S/H
20 IF H > 40 THEN 60
30 LET SS = H * S/H
40 PRINT SS
50 GOTO 10
60 LET H = H - 40
70 LET SS = 40 * S/H + (H * S/H * 1,5)
80 PRINT SS
90 GOTO 10
100 END

```

Economizaríamos uma etapa se a linha 80 fosse codificada como:

```
80 GO TO 40
```

e a linha 90 omitida. Visto que as linhas 80 e 90 no exemplo acima são cópias da 40 e 50, pode-se simplesmente codificar GO TO 40 no número de linha 80, economizando assim uma etapa.

E. VARIAÇÕES SOBRE UM TEMA

1. Opções de Impressão

a. Impressão de Constantes Alfanuméricas

Observe que nos exemplos anteriores, lemos os dados e imprimimos os resultados computados. Em geral, os dados que são processados são variáveis; isto é, dependem totalmente dos números que estão sendo inseridos no momento e não são conhecidos durante a fase de codificação.

Suponha, entretanto, que se queira imprimir uma mensagem alfanumérica descrevendo o conteúdo real do campo de saída. Visto que a mensagem que se deseja imprimir é conhecida antecipadamente, ela é denominada *constante*; isto é, não depende da entrada, mas é sempre a mesma para um determinado programa.

Codificam-se as constantes alfanuméricas em BASIC colocando-as entre aspas. Por exemplo, considere a seguinte declaração:

```
10 PRINT "ALÔ – SOU O SEU AMIGO COMPUTADOR"
```

A declaração PRINT fará com que, no momento em que a declaração 10 for executada, o computador imprima o seguinte:

```
ALÔ – SOU O SEU AMIGO COMPUTADOR
```

Uma declaração PRINT pode imprimir variáveis constantes ou uma combinação de ambas, usando uma vírgula.

```
10 PRINT "O TOTAL É", T
```

Esta instrução imprimirá a constante juntamente com o valor de T. Sendo assim, pode-se aumentar os programas apresentados anteriormente, conforme se segue:

EXEMPLO

Suponha que desejamos computar os salários semanais baseados em horas de trabalho e salário/hora; não considere o pagamento de hora extra neste exemplo. Além disto, desejamos imprimir, no final da operação, o total de salários pagos a todos os empregados, juntamente com as constantes adequadas.

Para isto, informamos ao computador que, se um campo de HORAS DE TRABALHO é inserido como 99, isto é uma entrada de reboque e significa que não há mais dados. Portanto, temos:

```
10 INPUT N$, H, S/H
20 IF H = 99 THEN 70
30 LET SS = H * S/H
40 LET TS = TS + SS
50 PRINT "SALÁRIOS SEMANAIS PARA", N$, "É", SS
60 GOTO 10
70 PRINT "TOTAL DE SALÁRIOS PAGOS É", TS
80 END
```

A linha 20 provocará um desvio para a 70 somente quando se insere 99 para HORAS DE TRABALHO; 99 seria inserido apenas no final do trabalho. Isto é, o último registro conteria 99 para HORAS DE TRABALHO.

A linha 40 computa o TOTAL DE SALÁRIOS adicionando todos os SALÁRIOS SEMANAIS, SS, cada vez que estes são computados.

EXEMPLO

Ler três conceitos para cada aluno. Imprimir a média de cada aluno e uma média da turma, juntamente com as constantes apropriadas.

Para imprimir a média de uma turma é necessário determinar o número de registros de alunos que foram processados. Para determinar o número de registros processados em qualquer procedimento, inclui-se a seguinte declaração LET depois que o computador tiver lido a entrada e executado o teste de fim de trabalho:

```
20 LET C = C + 1
```

Cada vez que uma declaração INPUT é executada, acrescenta-se 1 a C. C indicará sempre o número de registros de entrada processados.

Um conceito de prova 0 indicará uma condição de fim-de-dados. Portanto, temos:

```
10 INPUT E1, E2, E3
20 IF E1 = 0 THEN 80
30 LET C = C + 1
40 LET A = (E1 + E2 + E3)/3
50 PRINT "A MÉDIA DO ALUNO É", A
60 LET T = T + A
70 GOTO 10
80 LET T1 = T/C
90 PRINT "A MÉDIA DA TURMA É", T1
100 END
```

b. Impressão de Linhas em Branco

A seguinte instrução imprimirá uma linha em branco:

```
10 PRINT
```

2. Inserção de Dados Alfanuméricos Denominados Variáveis de Série

Até agora, usou-se uma declaração INPUT para inserir variáveis numéricas. Relembre a seguinte regra para designar variáveis numéricas:

Regra para Nomes de Variáveis Numérica

Usar uma única letra ou uma letra seguida de um dígito (i.e. A, A1, A9 etc.).

E1, T, N são todos nomes de variáveis numéricas válidas. Mas suponha que seja necessário inserir dados alfanuméricos tais como o nome de uma pessoa, seu endereço, sexo ou outra informação qualquer. Deve-se usar uma variável de série. A regra para o nome de uma variável de série é a seguinte:

Regra para Nomes de Variável de Série

Nome de variável seguido de \$ (letra ou letra mais dígito, e \$) (i.e., A\$, N\$, A1\$, etc.).

EXEMPLO

Ler como entrada o nome de um aluno e três resultados de prova (E). Imprimir o nome do aluno e a média.

```
10 PRINT "INSERIR O NOME DO ALUNO E 3 RESULTADOS DE PROVA"  
20 PRINT  
30 INPUT N$, E1, E2, E3  
35 IF E1 = 999 THEN 70  
40 LET A = (E1 + E2 + E3)/3  
50 PRINT N$, "A MÉDIA É" A  
60 GO TO 20  
70 END
```

RUN

INSERIR NOME DO ALUNO E 3 RESULTADOS DE PROVA

? JOSÉ ROBERTO, 80, 70, 75
A MÉDIA DE JOSÉ ROBERTO É 75

? EDUARDO CUNHA, 90, 95, 85
A MÉDIA DE EDUARDO CUNHA É 90

? END, 999, 999, 999

Todas as declarações IF-THEN codificadas até agora comparam variáveis numéricas com constantes numéricas. Suponha, entretanto, que se deseje ler nomes, datas de nascimento (mês e ano), e sexo (M para masculino, F para feminino). Neste caso, deseja-se imprimir datas de nascimento de pessoas do sexo feminino. Isto é, deve-se comparar uma variável de série — sexo — com uma literal alfanumérica "F". Observe que as literais alfanuméricas aparecem entre aspas.

```
10 INPUT N$, M, A, S$  
20 IF S$ = "F" THEN 40  
30 GO TO 10  
40 PRINT "A DATA DE NASCIMENTO É", M, "/" , A  
50 GO TO 10  
60 END
```

Vamos modificar o programa acima. Suponha que, em vez disto, queiramos imprimir o número total de pessoas do sexo feminino nascidas em abril. O fim da operação será indicado por "LAST" ("ÚLTIMO") no campo de NOME. O mês é indicado por um número de 1 a 12.

```
10 INPUT N$, M, A, S$  
20 IF N$ = "LAST" THEN 80  
30 IF M <> 04 THEN 10  
40 IF S$ = "F" THEN 60  
50 GOTO 10  
60 LET C = C + 1  
70 GOTO 10  
80 PRINT "O Nº TOTAL DE PESSOAS DO SEXO FEMININO NASCIDAS  
EM ABRIL É", C  
90 END
```

3. Outros Usos da Declaração LET

EXEMPLO

Para todos os salários lidos como entrada, acrescentar um aumento de Cr\$100.000,00.

```
10 INPUT S
20 LET S = S + 100.000,00
30 PRINT S
40 GO TO 10
50 END
```

Observe que as instruções seguintes produziram os mesmos resultados:

```
10 INPUT S
20 LET ST = S + 100
30 PRINT ST
40 GO TO 10
50 END
```

A declaração LET também pode ser usada para deslocar um valor para um campo. Por exemplo:

```
10 LET A = 10.3
```

tem como resultado o deslocamento de 10.3 para A.

F. DECLARAÇÃO REM

Um número de linha, seguido de REM, seguido de qualquer outra coisa, será considerado como uma observação ou comentário no programa. Por exemplo,

```
5 REM ESTE PROGRAMA CALCULA O TOTAL DE PESSOAS DO SEXO
  FEMININO
7 REM NASCIDAS EM ABRIL
```

poderia ser acrescentado ao programa anterior a título de esclarecimento ou comentário.

As declarações REM podem ser codificadas em qualquer parte do programa. O computador não considera as declarações REM durante a execução, mas imprime-as como parte da listagem do programa.

G. SUMÁRIO

O programa seguinte resumirá todos os conceitos vistos até agora.

EXEMPLO

Os registros de dados para uma companhia de seguros são lidos juntamente com os seguintes itens:

```
NOME
SEXO (M = MASCULINO, F = FEMININO)
DATA DE NASCIMENTO (M, A)
ESTADO (1 para RIO DE JANEIRO)
```

Imprimir:

```
% DE MOTORISTAS DO RJ  
% DE MOTORISTAS COM MENOS DE 25 ANOS  
% DE MOTORISTAS DO SEXO FEMININO
```

Dica: Suponha que o ano em curso seja 1984; portanto, qualquer pessoa nascida depois de 1959 é considerada com menos de 25 anos.

```
10 INPUT N$, S$, M, A, Q  
20 IF S$ = "Q" THEN 130  
30 REM Q NO CAMPO DE SEXO INDICA O FIM DO PROGRAMA  
40 LET T = T + 1  
50 REM CONTAR N° DE REGISTROS NA INSTRUÇÃO ACIMA  
60 IF S <> 1 THEN 80  
70 LET T1 = T1 + 1  
80 IF A <= 58 THEN 100  
90 LET T2 = T2 + 1  
100 IF S$ <> "F" THEN 10  
110 LET T3 = T3 + 1  
120 GOTO 10  
130 LET T1 = T1/T * 100  
140 PRINT "% MOTORISTAS DO RJ", T1  
150 LET T2 = T2/T * 100  
160 PRINT "% MOTORISTAS COM MENOS DE 25 ANOS", T2  
170 LET T3 = T3/T * 100  
180 PRINT "% MOTORISTAS DO SEXO FEMININO", T3  
190 END
```

TESTE DE AUTO-AVALIAÇÃO

1. (V ou F) A seguinte declaração é válida:
100 LET X = SQR (15.000)
2. Indicar quais dos seguintes itens são nomes de variáveis válidas.
 - a. A1
 - b. QTDAD
 - c. QUANTIDADE
 - d. D
 - e. Z 53
 - f. AB
3. Um nome de campo N representaria um campo ao passo que N\$ representaria um(a)
4. Indicar os resultados que serão obtidos caso se execute o seguinte:
30 LET X = V + W/S - E + C * D
onde V = 30, W = 10, S = 5, E = 20, C = 3 e D = 6
5. Indicar os resultados caso se execute o seguinte:
20 LET V = W + C ** 2 - E + F/S
onde W = 20, C = 10, E = 50, F = 20 e S = 5

6. Escrever um programa para calcular F, uma temperatura Fahrenheit, quando C, uma temperatura Celsius, é lida como entrada. Observe que a temperatura Fahrenheit equivale a $9/5$ da temperatura Celsius + 32.
7. Escrever uma seqüência de declarações para prosseguir para a linha 90 se A estiver entre 98.2 e 100.6, incluindo os dois extremos. Se A não estiver entre estes dois pontos, prosseguir para a linha 190.
8. Escrever uma seqüência de declarações para descobrir qual o maior dos três números A, B, C; colocar o resultado em H e prosseguir para a linha 200.
9. Codificar um programa para ler um valor para A e imprimir o valor absoluto de A. O valor absoluto de qualquer número é a sua distância em relação a 0. Portanto, o valor absoluto de $-3 = 3$; o valor absoluto de $25 = 25$. Considerar um valor de entrada de 999 para indicar o fim da operação.
10. Ler um conjunto de notas de prova de alunos. Imprimir a média. Um valor de 999 indica o final dos dados.
11. Cada conjunto de entradas contém o nome de um aluno e quatro notas atribuídas a ele. Determinar a média e imprimir um conceito baseado na média, conforme se segue:

<i>Média</i>	<i>Conceito</i>
90 +	A
80 a 89	B
70 a 79	C
60 a 69	D
< 60	F

Um nome "LAST" indica o fim do trabalho.

Soluções

1. Falso — os números não podem conter vírgulas.
2. a, d — somente uma única letra, ou uma letra seguida de um dígito, pode ser usada para um nome de variável em muitos sistemas.
3. Variável ou numérico; variável de série ou campo alfanumérico.
4. 30.
5. 74.
6.

```

10 INPUT C
20 LET F = 9/5 * C + 32
30 PRINT F
35 GO TO 10
40 END

```
7.

```

10 IF A > 100.6 THEN 190
20 IF A < 98.2 THEN 190
30 GO TO 90
   ou
10 IF A <= 100.6 THEN 30
20 GO TO 190
30 IF A <= 98.2 THEN 90
40 GO TO 190

```
8.

```

10 LET H = A
20 IF B > H THEN 50
30 IF C > H THEN 70
40 GO TO 200
50 LET H = B

```

```

60 GO TO 30
70 LET H = C
80 GO TO 200
9. 10 INPUT A
20 IF A = 999 THEN 80
30 IF A < 0 THEN 60
40 PRINT "O VALOR ABSOLUTO DE", A, "É", A
50 GOTO 10
60 PRINT "O VALOR ABSOLUTO DE", A, "É", -A
70 GOTO 10
80 END
10. 10 INPUT E1
20 IF E1 = 999 THEN 60
30 LET C = C + 1
40 LET T = T + E1
50 GOTO 10
60 PRINT "A MÉDIA É", T/C
70 END
11. 10 INPUT N$, E1, E2, E3, E4
20 IF N$ = "LAST" THEN 190
30 LET T = (E1 + E2 + E3 + E4)/4
40 IF T >= 90 THEN 110
50 IF T >= 80 THEN 130
60 IF T >= 70 THEN 150
70 IF T >= 60 THEN 170
80 LET C$ = "F"
90 PRINT N$, "O CONCEITO É", C$
100 GOTO 10
110 LET C$ = "A"
120 GOTO 90
130 LET C$ = "B"
140 GOTO 90
150 LET C$ = "C"
160 GOTO 90
170 LET C$ = "D"
180 GOTO 90
190 PRINT "ISTO É TUDO, AMIGOS! !"
200 END

```

H. DECLARAÇÕES READ E DATA

Até agora, consideramos a declaração INPUT como um comando de entrada de dados. Esta declaração é usada quando não se sabe, de antemão, que entrada será processada.

Às vezes, porém, tem-se um processamento complexo para executar, onde se sabe *exatamente* quais os dados com os quais se quer operar. Neste caso, usam-se declarações READ e DATA ao invés do comando INPUT. Observe que, com as declarações READ e DATA, não há necessidade de inserir dados adicionais quando a palavra RUN é digitada.

EXEMPLO

Suponha que para cada conjunto de valores de X e Y, quiséssemos calcular a raiz quadrada de $X^2 + Y^2$. Considere o seguinte programa:

```

10 READ X, Y
20 IF A = 99 THEN 90
30 LET A = SQR (X ** 2 + Y ** 2)
40 PRINT A
50 GO TO 10
60 DATA 3, 4
70 DATA 5, 7
80 DATA 99, 99
90 END

```

A declaração READ instrui o computador para obter um valor para X e um valor para Y a partir de uma declaração DATA. Quando a instrução READ é executada pela primeira vez, o computador obterá os valores da *primeira* declaração DATA (os primeiros dados serão 3 e 4). Depois de executar os cálculos desejados e de imprimir a resposta, o computador retorna à declaração 10. Recebe então instrução para obter outro conjunto de valores – um para X e um para Y. Visto que os valores da primeira declaração DATA já foram usados, o computador sabe automaticamente que deve obter dados da declaração 60, a próxima declaração DATA. Quando o computador retorna à declaração 10 pela terceira vez, encontra um 99 em A. Segue-se um desvio para a declaração END.

Seria igualmente válido ter uma declaração DATA como a seguinte:

```
50 DATA 3,4,5,7,99,99
```

Aqui, usa-se somente uma declaração DATA. A primeira declaração READ acarreta o uso apenas dos dois primeiros valores na declaração 50, visto que somente duas variáveis devem ser lidas. Na segunda vez que a declaração READ é executada, usam-se os dois valores seguintes na declaração DATA. Nesta forma de declaração DATA pode-se usar o maior número possível de elementos de dado que couberem em uma linha.

Observe que a localização típica das declarações DATA é no fim do programa, logo antes da declaração END. Entretanto, visto que as declarações DATA não são propriamente instruções, mas servem unicamente para fornecer dados para as declarações READ, elas podem ser introduzidas em qualquer lugar em um programa BASIC.

O exemplo que acabamos de discutir mostra como os dados numéricos podem ser incluídos em um programa. Os dados alfanuméricos também podem ser incluídos em declarações DATA, embora as regras variem ligeiramente, dependendo do computador em que o programa é executado. Alguns compiladores BASIC exigem que os dados alfanuméricos sejam colocados entre aspas, conforme exemplificado no seguinte trecho de programa:

```

10 READ N$, S
   :
180 DATA "NOÊMIA CUNHA", 10250
190 DATA "ANA SAAD", 13175
200 END

```

A declaração 10 instrui o computador para obter os valores de uma declaração DATA para dois campos – N\$, uma variável de série representando o nome, e S, um campo numérico representando o salário. Lembre-se de que quando se forma um

nome de variável de série usa-se uma letra seguida de um sinal de dólar. A regra para formar nomes ou variáveis de campos numéricos é usar uma letra ou uma letra seguida de um dígito. Se um compilador BASIC não requer o uso de aspas em torno dos dados alfanuméricos, elas devem ser eliminadas.

I. DECLARAÇÕES FOR ... NEXT

Conforme observado anteriormente, um circuito é uma seqüência de instruções que deve ser repetida um determinado número de vezes. Consideramos um problema que ilustra a maneira pela qual se executa um circuito. Suponha que se queira computar a quantia de dinheiro acumulada ao final de um, dois e três anos, caso seja depositada em banco uma quantia específica com uma determinada taxa de juros. Suponha que todo o dinheiro (capital e juros) permanece no banco. A fórmula geral apresentada a seguir é útil para resolver problemas envolvendo juros compostos.

$$C_n = C_0 (1 + J)^n$$

C_n é igual à quantia total depois de n anos de investimento de uma quantia inicial (C_0) com uma determinada taxa de juros (j).

Escreveremos agora um programa para calcular a fórmula acima, onde os valores de C (capital) e J (taxa de juros) devem ser inseridos como entrada. Interessam-nos os resultados em que n é igual a 1, 2 e 3.

a. Método 1 – Sem Circuitos

```
10 INPUT C, J
20 PRINT "RESULTADOS DOS JUROS COMPOSTOS"
30 PRINT
40 PRINT "TAXA DE JUROS = ", J, "QUANTIA INICIAL = ", C
50 PRINT
60 LET A = C * (1 + J) ** 1
70 PRINT "QUANTIA APÓS 1 ANO É IGUAL", A
80 LET B = C * (1 + J) ** 2
90 PRINT "QUANTIA APÓS 2 ANOS É IGUAL", B
100 LET C = C * (1 + J) ** 3
110 PRINT "QUANTIA APÓS 3 ANOS É IGUAL", C
120 END
```

Observe que as linhas 60 a 110 são muito semelhantes. As linhas 60, 80 e 100 utilizam a mesma fórmula básica com apenas uma modificação cada vez que o cálculo é realizado – o valor do expoente é 1, 2 ou 3. As linhas 70, 90 e 110, que imprimem a saída, são também muito semelhantes, com a diferença de que cada mensagem indica um ano diferente – 1, 2 ou 3.

Note que poderia se estabelecer uma rotina, em vez de três rotinas separadas, para computar os três valores. Esta última rotina poderia utilizar, por exemplo, um expoente variável N , que começa como um e é cada vez aumentado em 1, até ultrapassar três. Isto é, pode-se estabelecer um circuito para ser executado três vezes, variando N de 1 a 3.

b. Método 2 – Usando um Circuito

```
10 INPUT C, J
20 PRINT "RESULTADOS DOS JUROS COMPOSTOS"
30 PRINT
40 PRINT "TAXA DE JUROS = ", J, "QUANTIA INICIAL = ", C
50 PRINT
60 LET N = 1
70 LET A = C * (1 + J) ** N
80 PRINT "QUANTIA APÓS ANO", N, "É IGUAL", A
90 IF N = 3 THEN 120
100 LET N = N + 1
110 GOTO 70
120 END
```

Repare que este programa utiliza um circuito. A seqüência de instruções a ser repetida começa com a linha 70. As linhas 70 e 80 são executadas pela *primeira* vez com um valor de N igual a 1. Na linha 90, o computador realiza um teste para ver se a seqüência deve ser repetida. Visto que N não é igual a 3 na primeira vez que a declaração é executada, o computador prossegue automaticamente para a linha seguinte, onde acrescenta-se 1 a N, de maneira que N seja igual a 2. Com N igual a 2, então, o computador é instruído na linha 110 para retornar à linha 70 e processar novamente a seqüência de instruções.

c. Método 3 – Usando Instruções FOR e NEXT

Vimos a partir das soluções anteriores que desejamos executar uma seqüência de operações usando uma variável N que começa como 1 e é aumentada até chegar a 3. As declarações FOR e NEXT fornecem alguma flexibilidade para lidar com este tipo de problema.

A declaração FOR estabelece uma variável e especifica a faixa na qual ela deve variar. Portanto, para o nosso problema teríamos:

```
FOR N = 1 a 3
```

com um número de linha apropriado.

As declarações a seguir serão executadas, pela primeira vez, com N = 1. Depois de codificar todas as operações necessárias para N = 1, codifica-se

```
NEXT N
```

com um número de linha apropriado. Em essência, a declaração NEXT instrui o computador para repetir estas etapas com N no valor seguinte (N = 2).

Este procedimento seria repetido até que N fosse igual a 3 e que todas as instruções fossem executadas para esse valor. Então o programa continuaria com a declaração imediatamente seguinte à declaração NEXT.

Vamos considerar agora um método mais eficaz para lidar com circuitos:

```
10 INPUT C, J
20 PRINT "RESULTADOS DE JUROS COMPOSTOS"
30 PRINT
40 PRINT "TAXA DE JUROS = ", J, "QUANTIA INICIAL = ", C
```

```

50 PRINT
60 FOR N = 1 a 3
70 LET A = C * (1 + J) ** N
80 PRINT "QUANTIA APÓS ANO", N, "É IGUAL", A
90 NEXT N
100 END

```

A declaração FOR na linha 60 indica que a seqüência de instruções até a declaração NEXT (linha 90) deve ser repetida três vezes, com $N = 1$ na primeira vez, $N = 2$ na segunda vez, e $N = 3$ na última vez, até o final das declarações.

A declaração NEXT (linha 90) faz com que o campo N, especificado na declaração FOR (linha 60), seja aumentado automaticamente para o próximo valor cada vez que a seqüência for executada.

A declaração FOR usada no programa acima vai iniciar com N em 1 e resultará em N aumentada de 1 até que seja igual a 3.

Suponha, entretanto, que se queira iniciar N em 10 e a cada vez aumentar N em 10 e não em 1. Isto é, deseja-se atribuir a N os valores 10, 20, 30 etc., até que N seja igual a 100. A declaração FOR seria especificada do seguinte modo:

```
FOR N = 10 a 100 STEP 10
```

A opção STEP é necessária somente se vamos acrescentar ao valor de N algum número diferente de 1. No exemplo acima iniciamos N em 10 e o aumentamos em 10 e "pulamos" 10 cada vez que as operações seguintes são executadas, até que N seja igual a 100.

A seguir, exemplos de instruções FOR ... NEXT.

EXEMPLO 1

Somar os números ímpares de 101 a 1001 usando um circuito FOR ... NEXT.

```

10 FOR N = 101 a 1001 STEP 2
20 LET T = T + N
30 NEXT N
40 PRINT "A SOMA DOS NÚMEROS ÍMPARES DE 101-1001 É", T
50 END

```

EXEMPLO 2

Ler um valor para M. Determinar $M!$, onde $M! = M \times (M - 1) \times (M - 2) \dots \times 1$

```

10 LET T = 1
20 INPUT M
30 FOR N = 1 a M
40 LET T = T * N
50 NEXT N
60 PRINT "M É", M
70 PRINT "M! =", T
80 GOTO 10
90 END

```

Observe que a linha 30 poderia ser codificada como:

```
30 FOR N = M a 1 STEP - 1
```

É possível codificar circuitos FOR ... NEXT dentro de circuitos FOR ... NEXT. Considere o seguinte exemplo:

EXEMPLO

Existem 10 turmas de matemática na Faculdade ABC. Cada turma tem 25 alunos. Cada linha de entrada tem a média de um aluno. Escrever um programa BASIC usando circuitos FOR ... NEXT para imprimir as médias das 10 turmas:

```
10 FOR I = 1 a 10
20 LET T = 0
30 FOR J = 1 a 25
40 INPUT N1
50 LET T = T + N1
60 NEXT J
70 PRINT "MÉDIA DA TURMA", I, "=", ", ", T/25
80 NEXT I
90 END
```

J. MATRIZES

Existem muitos tipos de problemas onde é conveniente, e às vezes necessário, armazenar todos os dados de entrada antes de executar outro processamento. Para isto, deve-se armazenar no computador uma lista ou tabela de dados. É necessário um *subíndice* ou número para indicar a posição de um valor específico na lista ou tabela. Considere o seguinte programa, onde cada aluno em um determinado curso fez seis provas (E), cuja média será calculada pelo computador.

```
10 INPUT E1, E2, E3, E4, E5, E6
20 LET A = (E1 + E2 + E3 + E4 + E5 + E6)/6
30 PRINT A
40 GOTO 10
50 END
```

Este programa simplificado permite ao instrutor sentar-se no terminal, inserir as notas de um determinado aluno e em seguida obter a média impressa, antes de passar para o próximo aluno.

Observe que seis nomes diferentes de variáveis foram usados nas linhas 10 e 20: E1, E2, E3, E4, E5 e E6. Suponha que o instrutor tivesse dado 12 provas; neste caso, pode-se incorrer em vários problemas ao escrever o programa. Em primeiro lugar, não se pode usar nomes de variáveis tais como E10, E11 e E12, pois um nome de variável pode consistir somente de uma letra, ou uma letra acompanhada de um dígito. (Alguns compiladores aceitam nomes mais detalhados, mas o mínimo é uma letra ou letra e dígito.) Mesmo que seja possível escolher nomes de variável que sejam significativos, as instruções se tornam um tanto complicadas de digitar. Em segundo lugar e mais importante, torna-se cansativo e ineficiente codificar uma sequência de variáveis como a que mostramos no exemplo abaixo:

```
10 INPUT E1, E2, E3, E4, E5, E6, E7, E8, E9, T1, T2, T3
20 LET A = (E1, E2, E3, E4, E5, E6, E7, E8, E9 + T1 + T2 + T3)/12
```

Uma forma de contornar estes problemas é usar uma variável subscriptada para os resultados das provas. Considere novamente o problema em que se deve levar em conta, para cada aluno, resultados de seis provas. Suponha que seja necessário que o computador organize uma lista, ou uma *matriz* unidimensional. O termo matriz unidimensional significa simplesmente que os dados podem ser visualizados enquanto são armazenados em uma coluna no computador. O nome da lista será E. Ela terá seis fendas ou células nas quais os números podem ser colocados. Depois que os dados são armazenados na lista, a referência a qualquer item específico pode ser feita indicando-se a posição adequada na lista com o uso de um subíndice. Assim, podemos nos referir aos valores na lista como E(1), E(2), E(3), E(4), E(5) e E(6). E(1) refere-se ao primeiro número da lista denominada E, E(2) refere-se ao segundo número e assim por diante. A Figura 8.1 ilustra a maneira pela qual a lista descrita pode ser visualizada.

As instruções para ler os resultados e a média poderiam ter então a seguinte forma:

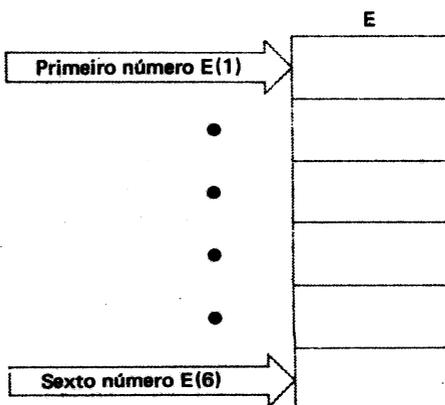
```
10 INPUT E(1), E(2), E(3), E(4), E(5), E(6)
20 LET A = (E(1) + E(2) + E(3) + E(4) + E(5) + E(6))/6
```

Apesar de estas instruções exemplificarem uma variável subscriptada, o uso deste método não facilitou em nada. Entretanto, talvez tenha ocorrido ao leitor a possibilidade de se usar as declarações FOR e NEXT para facilitar a programação. As seguintes instruções simplificam a codificação:

```
5 LET T = 0
10 FOR I = 1 A 6
20 INPUT E(I)
30 LET T = T + E(I)
40 NEXT I
50 PRINT T/6
60 GOTO 5
70 END
```

Figura 8.1

Lista das variáveis subscriptadas visualizada no computador.



A linha 5 estabelece uma variável T para acumular as notas totais de cada estudante. A linha 10 indica que as linhas subsequentes (20 e 30) devem ser repetidas seis vezes, com I variando de 1 a 6, com acréscimos de 1. As linhas 20 e 30 processam o "iésimo" valor de E; isto é, na primeira rodada do circuito, insere-se como entrada e armazena-se na primeira fenda da lista — E(1) — o primeiro valor de E, ou seja, primeira nota de prova. Este valor é então adicionado ao valor atual de T, que é zero. A linha 40 aumenta I em 1 e o circuito é então repetido.

Na segunda rodada a segunda nota de prova é inserida e armazenada como E(2), e assim por diante. Depois que as seis notas foram inseridas e acumuladas, o computador passa automaticamente para a linha 50, onde o total acumulado para o estudante cujos dados acabaram de ser processados é dividido por 6 e a resposta é então impressa. Em seguida, repete-se o processo para todos os outros estudantes.

Agora deve estar evidente o motivo pelo qual as variáveis subscritadas são tão úteis. Sem levar em consideração o número de valores que podem ser lidos em uma lista específica, pode-se usar essencialmente as mesmas instruções simples. Entretanto, existe uma outra instrução que é necessária no caso do subíndice exceder 10. Tal instrução é a declaração de dimensão (DIM). Seu objetivo é o de informar explicitamente ao computador quantos espaços reservar para uma variável subscritada quando se sabe que haverá subíndices maiores que 10. O compilador BASIC reserva automaticamente espaço suficiente para acrescentar um subíndice até 10 toda vez que uma variável subscritada é encontrada no decorrer da compilação. Se sabemos, por exemplo, que haverá 12 notas para cada estudante, será necessário uma declaração de dimensão como

```
10 DIM E(12)
```

no início do programa. Esta declaração informa explicitamente ao computador que deve reservar espaço suficiente para uma lista denominada E e deixar espaço para um subíndice até 12.

V. TÓPICOS AVANÇADOS EM BASIC

Um dos principais objetivos de usar computadores para o processamento de informação é produzir relatórios ou impressão de saída. A impressão de saída deve ser tão clara e significativa quanto possível, devendo ser projetada para satisfazer adequadamente às necessidades do usuário.

Utilizam-se tipicamente duas características especiais para que a saída se torne "acessível" ao usuário, ou fácil de ser interpretada pelo mesmo.

1. Deve-se incluir cabeçalhos que ajudem a identificar o relatório geral de saída e os campos específicos que devem ser impressos.
2. Os campos de saída devem ser *editados* de maneira a incluir símbolos tais como sinais de dólar, vírgulas, sinais negativos etc. para torná-los tão legíveis quanto possível.

Os cabeçalhos e os campos editados são utilizados sem levar em consideração se a saída é impressa em um relatório que serve como *hard copy* ou cópia exibida em uma tela de TRC. Em ambos os casos é fundamental que o programador torne a saída clara e significativa.

A. OPÇÃO DE TABULAÇÃO

Os cabeçalhos que consistem de constantes alfanuméricos ou literais são colocados em locais específicos em uma linha de saída, para identificar o relatório e os campos de saída. Para efeitos deste exemplo, considere a seguinte saída de TRC, supondo uma tela de TRC com largura de tela de 75 posições.

NOTAS DE ALUNOS

ÚLTIMO SOBRENOME	PRIMEIRO NOME	NOTA
REGO	JOSÉ	85
CODATO	ROBERTO	95
CUNHA	MARIZA	90

Para centralizar o cabeçalho, deseja-se que o A de ALUNO comece na posição 31. Isto foi determinado conforme se segue:

1. A linha de saída pode conter 75 caracteres.
2. O título NOTAS DE ALUNOS contém 17 caracteres incluindo dois espaços (um entre NOTAS e DE, outro entre DE e ALUNOS).
3. Para centralizar o título, gostaríamos de ter um número igual de espaços de cada lado da constante alfanumérica. Isto significa que deve-se ter 30 espaços de cada lado do cabeçalho:

$$\begin{array}{r} 75 \text{ posições na linha} \\ - 17 \text{ caracteres no título} \\ \hline 60 \text{ espaços à esquerda} \\ 60/2 = 30 \text{ espaços de cada lado do título} \end{array}$$

Portanto, começa-se o título na posição 31.

Pode-se usar a opção de tabulação na declaração IMPRESSA para se obter a saída nas localizações específicas desejadas. Por exemplo, a declaração

```
20 PRINT TAB (31); "NOTAS DE ALUNOS"
```

fará com que a constante alfanumérica comece na posição 31. Observe que um ponto-e-vírgula separa a opção de tabulação do cabeçalho real ou literal.

As posições iniciais para os cabeçalhos da coluna ÚLTIMO SOBRENOME, PRIMEIRO NOME e NOTA também são computados, de forma que cada um destes cabeçalhos de campo tenha um espaçamento uniforme na página. Um Gráfico de Espaçamento da Impressora ou um Formulário de Apresentação de TRC são úteis para ajudar o programador a estabelecer margens adequadas.

O seguinte trecho extraído de um programa exemplifica a maneira pela qual a opção de tabulação pode ser usada para se obter os resultados discutidos acima:

```
10 PRINT
20 PRINT TAB (31); "NOTAS DE ALUNOS"
30 PRINT
40 PRINT TAB (18); "ÚLTIMO SOBRENOME"; TAB (44); "PRIMEIRO
   NOME"; TAB (71); "NOTA"
50 PRINT
   :
```

```
120 PRINT TAB (18); U$; TAB (44); P$; TAB (73); N
130 END
```

Observe que na linha 120 a tabulação para o último item, N (de NOTA), foi fixada em 73, ao passo que a tabulação para o cabeçalho da coluna "NOTA" foi fixada em 71. Fez-se isto para ilustrar a maneira pela qual a nota real poderia ser centralizada abaixo do cabeçalho.

Antes de se utilizar a opção de tabulação, o programador deve determinar o tamanho da tela do terminal ou impressora na qual a saída deverá aparecer. Normalmente os tamanhos de linha variam de 72 a 132 posições de impressão por linha, dependendo do dispositivo usado.

3. DECLARAÇÃO PRINT USING

Considere o seguinte programa:

```
10 READ A
20 IF A = 9 THEN 60
30 PRINT A
40 GO TO 10
50 DATA 12.75,8.60,125000.50,9
60 END
RUN
12,75
8,60
125000,50
```

Os resultados seriam mais legíveis se fossem impressos da seguinte maneira:

```
12,75
8,60
125.000,50
```

Observe que (1) os números foram alinhados pela casa decimal, e (2) uma vírgula foi inserida no último número para facilitar a leitura. Veremos como se pode conseguir estes resultados com a declaração PRINT USING.

Dois dos principais objetivos da declaração PRINT USING são fornecer o seguinte:

1. Alinhamento correto dos resultados.
2. Capacidade de editar os resultados. *Editar* significa tornar os resultados mais legíveis e mais úteis adicionando caracteres de edição tais como sinais de dólar, vírgulas e asteriscos.

Antes de abordarmos detalhadamente a declaração PRINT USING, o leitor deve observar que a forma desta declaração varia ligeiramente de uma versão de BASIC para outra. Sugere-se uma revisão do manual da linguagem BASIC referente ao sistema que está sendo usado para determinar as necessidades exatas do programa a ser executado. Na discussão que se segue, chamaremos a atenção para duas diferenças principais entre uma versão e outra.

A fórmula geral da declaração PRINT USING é: PRINT USING [EXPRESSÃO(ÕES) DE SÉRIE, LISTA] onde a expressão de série indica o formato a ser

usado na impressão do(s) item(ns) da lista. (O BASIC TRS 80 utiliza um ponto-e-vírgula, em vez de vírgula, para separar a(s) expressão(ões) de série e a lista.)

Discutiremos agora algumas das expressões de série comumente usadas, para exemplificar a maneira pela qual pode-se conseguir o alinhamento e a edição de resultados.

1. Alinhamento de Campos Numéricos

Usa-se o símbolo # na expressão de série da declaração PRINT USING para indicar que deve-se imprimir um campo numérico. Por exemplo, se A tem um valor de 35, a declaração

```
30 PRINT USING "#####", A
```

fará com que o valor de A seja impresso como ~~0000~~35, onde ~~0~~ indica um espaço. Isto é, o número é ajustado à direita, sendo as posições precedentes preenchidas com lacunas.

Se o número a ser impresso contém um ponto decimal, pode-se incluir um ponto decimal na expressão de série da declaração PRINT USING.

```
30 PRINT USING "#####.##", A
```

o ponto decimal fará com que os resultados sejam arredondados em vez de truncados. Assim, se A tem um valor de 12,755 será impresso como 12,76 quando se executar a declaração acima. Observe que se A tem um valor de ,2, será impresso como 0,20.

O programa seguinte exemplifica o uso dos símbolos "#" e ".#".

```
10 READ A
20 IF A=9 THEN 60
30 PRINT USING "#####.##", A
40 GO TO 10
50 DATA 18,758, 7.352, 125000, .2.9
60 END
RUN
    18,76
     7,35
125000,00
   0,20
```

2. Inserção de Vírgulas em Campos Numéricos

Caso seja necessário imprimir uma ou mais vírgulas, em um número, pode-se simplesmente incluir estas vírgulas na expressão da série da declaração PRINT USING:

```
30 PRINT USING "#,###,###,###", C
```

Se C tem um valor de 1052000,50, será impresso como 1.052.000,50. Imprimem-se as vírgulas somente se o valor de C contém posições significantes em número suficiente; caso contrário, as vírgulas serão suprimidas. Portanto, se C tem um valor de 532,50, por exemplo, será impresso *sem as vírgulas*, como ~~000000~~532,50, onde ~~0~~ indica uma lacuna.

3. Inserção de Sinais de Dólar em Campos Numéricos

Se desejamos imprimir um sinal de dólar juntamente com os conteúdos de um campo, inclui-se o sinal de dólar na declaração PRINT USING.

```
10 PRINT USING "$ #,###.##", D
```

Se o valor de D fosse 3546,23, seria impresso como \$3.546,23.

4. Uso de um Sinal de Dólar Flutuante

Às vezes não é suficiente imprimir um sinal de dólar conforme mostrado acima. Por exemplo, se D contivesse 1,93, a saída seria impressa da seguinte maneira:

```
$ 1,93
```

No exemplo acima existem muitos espaços em branco separando o sinal de dólar do primeiro caractere significativo. Incluindo *dois* sinais de dólar no início da expressão de série da declaração PRINT USING, pode-se assegurar de que o sinal de dólar será impresso *ao lado* do primeiro caractere significativo. O uso de dois sinais de dólar fará com que o sinal de dólar *flutue*, ou seja impresso ao lado do primeiro dígito significativo do número. Considere o seguinte programa:

```
10 READ A
20 IF A = 9 THEN 60
30 PRINT USING "$$ #,###.##", A
40 GO TO 10
50 DATA 5200,50, 325,55, 85320,75.9
60 END
RUN
$5.200,50
 $325,55
$85.320,75
```

Examine o último valor de A a ser impresso. Observe que a posição ocupada pelo *segundo* sinal de dólar na expressão de série da declaração PRINT USING pode ser substituída por um dígito, caso seja necessário. O primeiro sinal de dólar nunca pode ser substituído por um dígito. Portanto, a expressão de série "\$\$ #,###.##" pode permitir que um número tão grande quanto 99999,99 seja impresso como \$99.999:9.

5. Impressão de Números com Asterisco

Às vezes deseja-se imprimir números com asteriscos à esquerda, em vez de espaços. Isto é conhecido como preenchimento com asterisco. Para este fim, pode-se incluir dois asteriscos (**) no início da expressão de série da declaração PRINT USING. Por exemplo, se D tem um valor de 25,75, a declaração

```
30 PRINT USING "*** #,###.##", D
```

imprimirá *****25,75.

Devem-se observar os seguintes aspectos referentes ao uso do asterisco na expressão de série:

1. Se vírgulas não significativas forem indicadas na série, serão substituídas por asteriscos, conforme foi mostrado no exemplo acima.
2. Os dois asteriscos podem ser substituídos por dígitos, se necessário, para imprimir um número grande. Por exemplo, com relação à declaração PRINT USING mencionada acima, se D tiver um valor de 850750,35, este valor será impresso como 850.750,35, *sem* nenhum asterisco.

6. Impressão de Sinais Negativos

Se um campo é negativo, o sinal negativo será impresso à esquerda do número, a menos que seja especificado de outro modo. Por exemplo, se E tem um valor de -25,50, a declaração

```
30 PRINT USING "###.##", E
```

fará com que E seja impresso como -25,50. Entretanto, pode-se querer que o sinal negativo seja impresso à *direita* do número, como acontece às vezes com relatórios financeiros. Para conseguir isto, coloca-se um sinal negativo à direita da expressão de série na declaração PRINT USING:

```
30 PRINT USING "###.##-", E
```

Se o valor do campo é positivo, aparecerá um *espaço em branco* na saída, em vez do sinal negativo.

7. Impressão de Variáveis e Constantes de Série

Até agora, consideramos as variações da declaração PRINT USING que podem ser usadas para imprimir campos numéricos. Para imprimir variáveis e/ou constantes de série, pode-se determinar o tamanho do campo usando o caractere de barra inclinada para trás (\) na declaração PRINT USING. Cada (\) permite a impressão de um caractere. Cada espaço entre as barras também permite a impressão de um caractere. Portanto, a declaração PRINT USING

```
20 PRINT USING "\ \", "NOME"
```

possibilitará a impressão dos quatro caracteres da palavra NOME. Se a expressão de série tivesse sido escrita como "\ \", somente NO seria impresso. De forma semelhante, para se imprimir os conteúdos de uma variável de série, seria possível escrever:

```
50 PRINT USING "\ \", B$
```

No lugar do caractere de barra inclinada para trás (\), algumas versões de BASIC usam um dos seguintes símbolos:

1. Para BASIC TRS 80: um símbolo de porcentagem (%). Por exemplo:


```
20 PRINT USING "% %", "NOME"
```
2. Para sistemas que usam BASIC-PLUS-2: a letra L, com o primeiro L em uma série precedido por uma única aspa. Por exemplo:


```
20 PRINT USING "'LLLL", "NOME"
```
3. Para sistemas que usam C BASIC:

- a. *Constantes* de série usam uma barra inclinada para a frente (/). Por exemplo:

```
20 PRINT USING "//", "NOME"
```

- b. *Variáveis* de série usam um & para imprimir um campo de qualquer comprimento. Por exemplo:

```
30 PRINT USING "&","A$"
```

Observe, então, que existe uma grande disparidade no formato das declarações PRINT USING, dependendo do compilador usado.

8. Utilização de Mais de uma Expressão de Série em uma Declaração PRINT USING

Muitas vezes se torna necessário imprimir uma combinação de campos numéricos e variáveis ou constantes de série em uma linha específica. O exemplo seguinte ilustra a maneira pela qual duas ou mais expressões de série podem ser especificadas em uma declaração PRINT USING.

```
150 PRINT USING "\      \###,###", "O NÚMERO É", N
```

Observe que as expressões de série são escritas simplesmente na ordem em que elas serão usadas, com pelo menos um espaço entre cada expressão de série.

9. Uso de Ponto-e-Vírgula para Separar Itens em uma Declaração PRINT

Até agora, usaram-se vírgulas para separar itens em uma declaração PRINT. Isto imprime os itens em áreas separadas na página, conforme se segue:

```
10 PRINT 1,2,3
RUN
1           2           3
```

Para imprimir os itens de maneira mais compacta, pode-se usar um ponto-e-vírgula em vez de uma vírgula para separá-los.

```
10 PRINT 1;2;3
RUN
1 2 3
```

C. FUNÇÕES DE BIBLIOTECA

Existem funções disponíveis em BASIC que permitem ao programador chamar rotinas pré-programadas que podem ser usadas com o programa. Por exemplo, suponha que se quer determinar a parte inteira de um número A. A instrução seguinte armazenará essa parte inteira na variável rotulada T

```
LET T = INT (A)
```

INT () é uma função de biblioteca que pode ser usada em qualquer programa BASIC.

EXEMPLOS

```
10 LET T = INT (A)
      A           T
      3,7         3
      4,16        4
      - 7,23      - 7
      6,0         6
```

A instrução usando INT () chama a rotina para determinar a parte inteira de um número.

EXEMPLO

Leia um número N. Determine se ele é ímpar ou par.

```
10 INPUT N
20 IF N = 99999 THEN 100
30 LET T = N/2
40 LET Q = INT (T)
50 IF N = 2 * Q THEN 80
60 PRINT "O NÚMERO"; N; "É ÍMPAR"
70 GO TO 10
80 PRINT "O NÚMERO"; N; "É PAR"
90 GO TO 10
100 END
```

Além da função INT, existem várias outras funções de biblioteca disponíveis. Conforme se observou anteriormente, a função SQR pode ser usada para determinar a raiz quadrada de um número.

```
5 REM ESTE PROGRAMA CALCULA A HIPOTENUSA DE UM TRIÂNGULO
7 REM RETÂNGULO
10 INPUT A, B
20 LET C = SQR (A ** 2 + B ** 2)
30 PRINT C
```

Observe que os resultados impressos serão exatamente os mesmos que

```
20 LET C = (A ** 2 + B ** 2) **.5
```

ABS () é uma função de biblioteca para calcular o valor absoluto de um número. Desta forma, uma solução para determinar o valor absoluto de A é a seguinte:

```
10 INPUT A
20 IF A = 999 THEN 60
30 LET T = ABS (A)
40 PRINT "O VALOR ABSOLUTO DE"; A; "É"; T
50 GO TO 10
60 END
```

As funções normais de biblioteca incluem:

FUNÇÕES NORMAIS DE BIBLIOTECA

<i>Função</i>	<i>Significado</i>
SQT (TR)	Determina a raiz quadrada de T.
INT (T)	Determina a parte inteira de T.
ABS (T)	Determina o valor absoluto de T.
SGN (T)	Determina o sinal de T: se $T = 0$, $SGN (T) = 0$ se $T < 0$, $SGN (T) = -1$ se $T > 0$, $SGN (T) = +1$
EXP (T)	Determina o logaritmo de T.
SIN (T)	Determina o seno de T.
COS (T)	Determina o co-seno de T.
TAN (T)	Determina a tangente de T.
RND (T)	Determina um número aleatório.

A discussão seguinte será sobre o gerador de números aleatórios, devido ao seu uso difundido, especialmente para construção de modelos e para programas de jogos.

D. USO DE RND EM BASIC

A função de biblioteca RND é uma função usada com freqüência, particularmente para codificar um jogo ou construir um modelo onde os *números aleatórios* são freqüentemente utilizados. A função RND varia um pouco de sistema para sistema. Dependendo do sistema, pode-se usar o seguinte:

1. LET Y = RND
2. LET Y = RND (T), onde T é um inteiro, ou, onde T deve ser 0 ou 1.

EXEMPLO

```
10 REM ARMAZENAR OS RESULTADOS DE DOIS DADOS
15 REM ONDE RND (0) GERA UM NÚMERO ALEATÓRIO
20 REM DE 0 A 1
25 LET Y = RND (0)
30 LET X = INT (6 * Y + 1)
40 LET T = RND (0)
50 LET Z = INT (6 * T + 1)
60 PRINT X, Z
```

O trecho de programa acima funcionará adequadamente em sistemas que geram um número aleatório de 0 a 1 com o uso de RND (0).

Se Y é o número aleatório de 0 a 1, então $6 * Y$ será um número de 0 a 5,9999. $6 * Y + 1$ será um número de 1 a 6,9999. O INT ($6 * Y + 1$) vai gerar então um inteiro de 1 a 6. As declarações 40 e 50 produzem um segundo número aleatório de 1 a 6.

EXEMPLO

Codificar o programa para um jogo *Mastermind* simples, no qual o jogador deve adivinhar quatro dos números gerados pelo computador de 1 a 6 e colocá-los na seqüência correta de acordo com a escolha feita pelo computador.

```

10 PRINT "ISTO É MASTERMIND"
12 PRINT
14 PRINT
15 PRINT "PARA INICIAR, ESCOLHER 4 NÚMEROS DOS DÍGITOS 1-6--"
18 PRINT "NÃO DUPLICAR"
23 PRINT "X SIGNIFICA POSIÇÃO À DIREITA, INTEIRO À DIREITA"
25 PRINT "I SIGNIFICA INTEIRO À DIREITA, POSIÇÃO ERRADA"
26 PRINT "-- SIGNIFICA NÚMERO ERRADO"
28 PRINT
110 FOR I = 1 A 4
120 LET A(I) = 1 A 4
130 NEXT I
140 FOR I = 5 A 8
150 A(I) = INT (RND (I) * 6 + 1)
160 IF A (I) = A (I - 1) THEN 150
170 IF A (I) = A (I - 2) THEN 150
180 IF A (I) = A (I - 3) THEN 150
190 IF A (I) = A (I - 4) THEN 150
200 NEXT I
210 LET N = 0
220 LET N = N + 1
230 PRINT
240 LET M = 0
250 INPUT B
260 LET B = B/1000
270 FOR I = 1 A 4
280 LET A (I) = INT (B)
290 LET B = (B - A(I))* 10
300 IF I = 3 THEN B = B + .0005
310 NEXT I
320 FOR I = 1 A 4
330 LET J = I + 4
340 IF A (I) <> A (J) THEN 380
350 PRINT "X";
360 LET M = M + 1
370 LET A (I) = 17
380 NEXT I
390 IF M = 4 THEN 550
400 FOR I = 1 A 4
410 FOR J = 5 A 8
420 IF A (I) <> A (J) THEN 470
430 IF J = I + 4 THEN 470
440 PRINT "I";
450 LET M = M + 1
460 GOTO 480
470 NEXT J
480 NEXT I
490 IF M = 4 THEN 220

```

```

500 LET M = M + 1
510 FOR I = M A 4
520 PRINT "--";
530 NEXT I
540 GOTO 220
550 PRINT
560 PRINT "VOCÊ TRAPACEOU";N;" LANCES"
570 GOTO 110

```

EXEMPLO

Escrever um exercício e módulo prático para ajudar alunos de curso primário a aprender tabuadas de multiplicação. Usar RND para obter números ao acaso.

```

10 DIM A$(10)
20 FOR I = 1 A 10
30 READ A$(I)
40 NEXT I
50 LET I = 1
60 PRINT "OI, QUAL O SEU NOME?"
70 INPUT N$
80 PRINT "PRAZER EM CONHECÊ-LO; N$"
90 PRINT "VOCÊ GOSTARIA DE PARTICIPAR DE UM JOGO DE
    ARITMÉTICA?"
100 INPUT Y$
110 IF Y$ = "SIM" THEN 130
120 GOTO 510
130 PRINT "BEM, VAMOS COMEÇAR"
140 PRINT "DIGITAR AS SUAS RESPOSTAS ATÉ"
150 PRINT "VOCÊ QUER PARAR – ENTÃO APERTAR A TECLA –1"
160 PRINT
170 LET C = INT (RND (1) * 9)
180 LET D = INT (RND (1) * 9)
190 PRINT "QUANTO É";C;"X";D;"?"
200 INPUT Z
210 IF Z = -1 THEN 360
220 IF V = 0 THEN Y = Y + 1
230 IF Z = C * D THEN 280
240 PRINT "VOCÊ COMETEU UM ERRO, TENTAR NOVAMENTE"
250 IF V = 0 THEN X = X - 1
260 LET V = 1
270 GOTO 200
280 LET V = 0
290 PRINT A$(1)
300 PRINT
310 LET X = X + 1
320 LET I = I + 1
330 IF I = 11 THEN I = 1
340 GOTO 170

```

```

350 PRINT "QUE BOM JOGAR COM VOCÊ" ; N$
360 PRINT "SEU RESULTADO É"; X; "CORRETAS EM"; Y; "PERGUNTAS"
370 GOTO 510
380 DATA "EXCELENTE, VAMOS TENTAR OUTRO"
390 DATA "NÃO OLHE, MAS O SEU CÉREBRO ESTÁ APARECENDO!"
400 DATA "VOCÊ É REALMENTE UM GÊNIO"
410 DATA "EINSTEIN DEVE TER SIDO SEU IRMÃO!"
420 DATA "DÃO PRÊMIO NOBEL EM CÉREBRO??"
430 DATA "BRILHANTE, BRILHANTE! !"
440 DATA "MANTENHA-SE FIRME!"
450 DATA "QUE MENTE REFINADA!"
460 DATA "ESTOU IMPRESSIONADO!"
470 DATA "OH! !"
510 END

```

E. SUB-ROTINAS

Uma *sub-rotina* é uma seqüência de instruções que devem ser executadas de pontos diferentes de um programa. No lugar de codificar o conjunto de instruções cada vez que for necessário, pode-se codificar uma única rotina e depois chamá-la e executá-la cada vez que for necessária. Este processo é executado utilizando-se a declaração GOSUB, que tem a seguinte fórmula:

GOSUB (número de linha)

Cada vez que aparecer a declaração GOSUB, transfere-se o controle para o número de linha indicado. Executam-se, então, as declarações que começam nesse número de linha, até aparecer uma declaração RETURN. O controle, então, retorna à declaração *imediatamente seguinte* à GOSUB.

EXEMPLO

```

10 _____
20 _____
30 _____
40 GOSUB 1000
50 _____
60 _____
70 _____
80 GOSUB 1000
90 _____
100 _____
110 _____
120 _____
1000 _____
1010 _____
1020 _____
1030 RETURN
:
1040 END

```

As etapas entre 1000 e 1030 são executadas quando se encontra o comando GOSUB.

Depois do RETURN, o controle passa para a declaração posterior à declaração GOSUB.

Depois de executar o GOSUB no número de linha 40, executam-se as linhas 1000 até 1030 e o controle volta então à declaração número 50. Após a execução do GOSUB na linha 80, as linhas 1000 até 1030 são novamente executadas e o controle retorna à declaração número 90.

EXEMPLO

Ler a entrada que consiste de três campos: quantidade 1, quantidade 2 e quantidade 3. Imprimir a palavra "ERRO" e o campo específico que provocou um erro, se existir qualquer uma das seguintes condições:

quantidade 1 > 1000
quantidade 2 < 500
quantidade 3 = 750

Se houver mais de um erro, imprimi-los separadamente.

```
10 INPUT A1, A2, A3
20 IF A1 = 999 THEN 9999
30 LET T$ = "QTDADA 1"
40 IF A1 = 1000 THEN 60
50 GOSUB 1000
60 LET T$ = "QTDADA 2"
70 IF A2 >= 500 THEN 90
80 GOSUB 1000
90 LET T$ = "QTDADA 3"
100 IF A3 <> 750 THEN 120
110 GOSUB 1000
120 GOTO 10
1000 PRINT "ERRO EM"; T$
1010 RETURN
9999 END
```

As seqüências GOSUB ... RETURN podem também ser úteis para produzir resultados variáveis, dependendo de quando elas são executadas em um programa.

EXEMPLO

```

:
50 LET A = 1
60 LET Z = 10
70 LET S = 1
80 GOSUB 900
90
:
:
200 LET A = 5
210 LET Z = 20
220 LET S = 5
230 GOSUB 900
```

```

900 FOR I = A A Z STEP S
910 PRINT I
920 NEXT I
930 RETURN
940 END

```

Este segundo procedimento é um método útil para produzir diagramas ou gráficos. Codifica-se uma rotina. A produção da saída real dependerá dos valores dos parâmetros específicos quando a declaração GOSUB é executada.

F. FUNÇÕES DE SÉRIE

Lembre-se que os dados alfanuméricos podem ser armazenados em um campo cujo nome consiste de um caractere alfabético seguido de um \$. Portanto, A\$, B\$ etc. podem ser usados como nomes de campos alfanuméricos ou variáveis de série.

Podem-se ler, imprimir e comparar campos alfanuméricos. Para se comparar um campo com "ABC", por exemplo, e desviar para a declaração 50 numa mesma condição, codifica-se:

```
10 IF F$ = "ABC" THEN 50
```

Quando se lida com sistemas interativos, às vezes é útil manipular variáveis de série de maneiras mais sofisticadas. Por exemplo, talvez seja necessário separar o campo em caracteres individuais para fins classificatórios ou para tipos específicos de análise.

As *funções de série* de BASIC permitem aos programadores modificar, comparar, analisar e combinar (denomina-se concatenar) dados alfanuméricos. Para que os computadores sejam úteis em ambientes interativos onde os usuários talvez não estejam particularmente voltados para o computador, é necessário que tais usuários possam inserir dados que talvez não estejam na forma exata desejada. Com o uso de funções de série, o programa BASIC pode analisar a entrada para ver se ela contém uma resposta adequada, e em seguida operar nela, conforme o necessário.

EXEMPLO 1

Suponha que se peça a um operador de terminal uma resposta "S" para indicar SIM e "N" para indicar NÃO. É possível que o operador responda com a palavra completa em vez da letra S ou N conforme solicitado. A codificação seguinte eliminará todas as dificuldades relacionadas com uma resposta de terminal que talvez não seja exatamente a que foi pedida.

```

10 PRINT "HOJE É O PRIMEIRO DIA DO MÊS?"
20 PRINT "INSERIR S PARA SIM E N PARA NÃO"
30 INPUT A$
40 IF A$ = "S" THEN 500
50 IF A$ = "N" THEN 700
60 IF LEFT$(A$) = "S" THEN 500
70 IF LEFT$(A$) = "N" THEN 700

```

A função de série LEFT\$(AB) permite ao programador testar o caractere mais à esquerda de A\$ para determinar se é um S ou N. Desta maneira, mesmo que o ope-

rador de terminal tivesse digitado a palavra toda (SIM ou NÃO), o programa prosseguiria corretamente.

EXEMPLO 2

Suponha que se peça ao operador de terminal para inserir a descrição de um item. O tamanho deste campo é variável e pode alterar o alinhamento dos dados de saída. É possível determinar o comprimento de uma variável de série usando-se uma função de série LEN.

```
10 PRINT "INSERIR A DESCRIÇÃO DE ARTIGO E O PREÇO DO ARTIGO"
20 INPUT D$,P
30 LET T = LEN (D$)
40 IF T > 15 THEN 80
  :
```

Eis aqui algumas das funções de série que podem ser usadas em BASIC:

<i>Função de Série</i>	<i>Objetivo</i>	<i>Exemplo</i>
LEN (série)	Determinar o comprimento da série	se F\$ é igual a "FRED" LEN (F\$) = 4
LEFT\$ (série, nº de caracteres)	Determinar os caracteres em série à extrema esquerda	LEFT\$ ("NOTHING", 2) = NO
RIGHT\$ (série, nº de caracteres)	Determinar os caracteres em série à extrema direita	RIGHT\$ ("AB", 1) = B
série 1 + série 2	Combinar ou concatenar duas séries	"A-" + "OK" = "A - OK"
MID\$ (série, expressão 1, expressão 2)	Começar com o caractere na expressão 1 até a expressão 2	MID\$ ("ABCDE", 3,2) = "CD"

Nota: Quando o número de caracteres é omitido, presume-se que seja 1.

G. CLASSIFICAÇÃO DE DADOS EM BASIC

Uma aplicação comum de computadores em negócios é a classificação de dados. Os dados freqüentemente devem ser classificados em seqüência ascendente, isto é, do valor mais baixo para o mais alto. De maneira semelhante, os dados às vezes devem ser classificados em seqüência descendente, isto é, do valor mais alto para o mais baixo.

Em geral os dados a serem classificados são colocados desordenadamente em uma matriz e em seguida organizados de forma adequada. Uma técnica comum de classificação é denominada *classificação de bolha*. Usando uma classificação de bolha, comparamos dois valores adjacentes em uma matriz. Se desejamos uma seqüência ascendente, então colocamos primeiramente o valor mais baixo entre dois que estão sendo comparados, seguido do valor mais alto.

Através deste procedimento, duas quantidades são comparadas até que toda a matriz tenha sido processada, em seguida compara-se o próximo grupo de dois valores na matriz. Este procedimento continua até que se consiga colocar todas as entradas na seqüência adequada. Observe que a classificação de bolha pode ser usada para classificar valores numéricos ou alfanuméricos. Mostraremos abaixo um trecho de programa em BASIC que classifica 15 variáveis em seqüência ascendente. Suponha que os números foram lidos em uma matriz B(15).

```
10 LET A = 0
20 FOR I = 1 A 14
30 IF B(I) < B(I + 1) THEN 80
40 LET C = B(I)
50 LET B(I) = B(I + 1)
60 LET B(I + 1) = C
70 LET A = 1
80 NEXT I
90 IF A = 1 THEN 10
100 REM OS NÚMEROS AGORA ESTÃO EM SEQÜÊNCIA
  :
```

V. VERSÕES DE BASIC

Note que consideramos a forma de BASIC padrão, que pode operar na maioria dos modelos e tamanhos de computadores. O programa em BASIC é muitas vezes interpretado, e não compilado, de modo que os erros são indicados logo após a linha que contém os erros de sintaxe.

```
10 INPUT A,B,C
20 LET D = A + B + C
30 PRINT D(
```

SYNTAX ERROR D(

Existem várias versões de BASIC ampliado. Em uma versão, por exemplo, uma declaração IF-THEN não causa necessariamente um desvio; em vez disto, pode-se executar um comando.

```
10 IF C = 0 THEN T = T + 1
```

Além disso, as declarações podem ser agrupadas em uma única linha, separadas somente por um ponto-e-vírgula ou outro delimitador, dependendo da versão específica do BASIC ampliado.

```
10 INPUT A,B,C; LET D = (A + B + C)/3; PRINT D
```

Algumas versões de BASIC permitem nomes de variáveis mais longas. Portanto, HORAS ou CONTAGEM podem ser usados no lugar de H ou C.

A versão de BASIC que ensinamos aqui funcionará em todos os computadores. Entretanto, ao examinar o manual do usuário o leitor poderá descobrir que existem inúmeras extensões disponíveis.

VI. SUMÁRIO DA LINGUAGEM BASIC

A Linguagem BASIC

Vantagens

1. Fácil de aprender e de codificar.
2. Forma livre — sem regras de estrutura ou pontuação.
3. Operações e funções matemáticas de fácil execução.

Desvantagens

1. As operações E/S são de certo modo limitadas.

TERMOS-CHAVE

Acessível ao usuário
Circuito
Classificação de bolha
Código de autorização
Constante
Edição
Função de biblioteca
Função de série
Hierarquia de operações
Início de comunicação
Matriz
Número aleatório
Protocolo
Sinal de dólar flutuante
Subíndice
Sub-rotina
Variável
Variável de série

QUESTIONÁRIO

(As respostas para as questões de números pares acompanham o teste.)

1. Uma declaração INPUT é usada quando
2. Para incluir os dados como parte de um programa usam-se declarações e
3. Quantos valores se imprimirão para D no seguinte programa?
10 READ A, B, C
20 LET D = A + B + C
30 PRINT D
40 GO TO 10
50 DATA 10,5,20,30,8,50,60,20,100
60 DATA 30,40,50,20,10,100,40,60
70 END
4. Escrever um programa, usando declarações READ e DATA, para ler um campo de código e uma quantidade. Se o código for "A", calcular e imprimir T, taxa, como 4,5% da quantidade. Se o código for "B", calcular e imprimir T, taxa, como 3,7% da quantidade. Se o código for "C", calcular e imprimir T, taxa, como 2,2% da quantidade. Usar este procedimento para os seguintes valores:

<i>Código</i>	<i>Quantidade</i>	<i>Código</i>	<i>Quantidade</i>
C	5200	B	3372
A	463	A	495
B	8211	B	625
		C	42

5. Codificar um programa para ler nomes e pesos* de cada um dos vários indivíduos e imprimir somente os nomes dos que pesam menos de 54 kg ou mais de 110 kg. Usar declarações READ e DATA.
6. Revisar o programa acima para ler o peso e a altura (em metros) de cada pessoa. Imprimir o nome de cada pessoa que pesa menos de 54 kg e mede menos de 1,60 m, ou que pesa mais de 110 kg e mede mais de 1,85 m. No final, imprimir o número total de nomes no relatório.
7. Codificar um programa para ler os campos de quantidade e imprimir a quantidade média depois de ler a entrada, supondo que uma quantidade de 999,99 indica uma condição de fim de trabalho. Usar uma declaração INPUT.
8. Escrever uma declaração FOR variando X de modo que ele atribua todos os valores de números ímpares a 99.
9. Escrever uma rotina, usando uma declaração INPUT e declarações FOR e NEXT, para ler 20 notas de alunos e depois determinar e imprimir uma média da turma.
10. Indicar quantas vezes o circuito será executado com a seguinte declaração e quais os valores que serão atribuídos a X.
10 FOR X = -5 A 8 STEP 3
11. O objetivo de uma declaração de dimensão (DIM) é
12. Um subíndice é usado para indicar em uma lista ou matriz unidimensional.
13. Escrever um programa usando declarações FOR-NEXT para ler os pesos de 11 alunos e imprimir a média. Os pesos serão fornecidos em uma declaração DATA. Armazenar todos os dados em uma lista antes de prosseguir os cálculos.

Respostas

(Fornecemos respostas somente para as questões de números pares.)

2. READ
DATA

4. 10 READ C\$, A
20 IF C\$ = "A" THEN 70
30 IF C\$ = "B" THEN 100
40 LET T = .022 * A
50 PRINT T
60 GOTO 10
70 LET T = .045 * A
80 PRINT T
90 GOTO 10
100 LET T = .037 * A
110 PRINT T
120 GOTO 10
130 DATA "C", 5200, "A", 463, "B", 8211
140 DATA "B", 3372, "A", 495, "B", 625, "C", 42
150 END

6. 10 LET T = 0
20 READ N\$, P, A
30 IF P = 999 THEN 140

```

40 IF P < 54 THEN 70
50 IF P > 110 THEN 90
60 GOTO 20
70 IF A < 1.60 THEN 110
80 GOTO 20
90 IF H > 1.85 THEN 110
100 GOTO 20
110 LET T = T + 1
120 PRINT N$
130 GOTO 20
140 PRINT T
150 DATA "JOAN CONNORS", 56, 1.48
160 DATA "MARK DORBS", 125, 1.63
170 DATA "KATHY FRANKS", 53, 1.56
180 DATA "ROBERT PETERS", 83, 1.61
190 DATA "ÚLTIMO", 999,999
200 END

```

8. FOR X = 1 A 99 STEP 2
10. Cinco vezes: X = -5, X = -2, X = 1, X = 4, X = 7.
12. A posição de um valor específico.

PROGRAMAS PRÁTICOS EM BASIC

1. Escrever e executar um programa para ler cinco quantidades de um terminal, calcular e imprimir a média destas quantidades, e em seguida repetir o processo.
2. Escrever um programa para ler um valor de vendas em um terminal. Se o valor das vendas for maior que Cr\$ 500.000,00, a comissão é computada como 5% do valor das vendas. Se o valor das vendas não for maior que Cr\$ 500.000,00 porém maior que Cr\$ 200.000,00, então a comissão é 3% do valor das vendas. Se o valor das vendas for Cr\$ 200.000,00 ou menos, então a comissão é zero. Imprimir a comissão.
3. Escrever um programa para calcular e imprimir um determinado imposto sobre o salário, para ser lido em um terminal.

Imposto = 6,70% do salário até Cr\$ 32.400,00

(Nenhum imposto é computado acima dos primeiros Cr\$ 32.400,00.)

4. Escrever um programa para calcular e imprimir o salário a partir de um número referente ao salário/hora e outro referente às horas de trabalho. O salário é computado do seguinte modo:

salário = salário/hora × horas, para horas < 40
 salário = salário/hora × (40) + salário/hora × 1,5 × (horas - 40) para horas > 40.
5. Escrever e executar um programa para ler os seguintes dados de acidente e produzir um relatório com a informação apresentada. Usar declarações READ e DATA para a entrada. Fornecer cabeçalhos adequados no relatório.

ENTRADA: Informação para cada motorista envolvido em um acidente no ano passado.

- a. Nome do motorista.
- b. Código de estado (1 para Rio de Janeiro).
- c. SEXO (M para Masculino, F para Feminino).
- d. Data de nascimento (mês, ano).

SAÍDA: Um relatório que apresente os seguintes resultados:

- a. A porcentagem de motoristas que eram do sexo masculino e tinham mais de 30 anos.
- b. A porcentagem de motoristas do Rio de Janeiro que eram do sexo feminino.

6. Modificar a saída no problema acima, conforme se segue. Fazer um relatório que relacione os nomes de todas as pessoas do sexo feminino, do Rio de Janeiro, que estavam envolvidas em um acidente. Indicar no final do relatório o número total de motoristas do Rio de Janeiro envolvidos em acidentes. Supor que um quinto campo é adicionado com 1 para acidentes, e 0 para a não ocorrência de acidentes.
7. Escrever e executar um programa para determinar a quantidade mais econômica de cada produto para uma companhia manufatureira produzir. A quantidade de encomenda econômica, Q , pode ser determinada pela fórmula:

$$Q = \text{raiz quadrada de } \frac{2RC}{E}$$

A entrada para cada produto consistirá de:

- a. Nome do produto.
- b. Exigência de produção anual total (R).
- c. Controle de estoque contendo o custo por unidade (E).
- d. Custo da organização por encomenda (C).

A saída deve ser um relatório que relacione os dados acima para cada produto, juntamente com o valor de Q que foi determinado. Incluir informação de identificação adequada. Considerar 25 produtos.

8. Escrever um programa para ler um número decimal de dois dígitos e convertê-lo em binário.
9. Escrever um programa para ler um número binário de cinco dígitos e convertê-lo em decimal.
10. Escrever um programa para calcular os honorários de um advogado. Três campos são lidos como entrada.

- NOME DO CLIENTE
- TIPO DE HONORÁRIO
- QUANTIDADE RECUPERADA

Se o tipo de honorário é = 1, valor do honorário = um terço da quantidade recuperada.

Se o tipo de honorário = 2, isto indica uma escala móvel; o valor do honorário será:

- 50% do primeiro Cr\$ 1.000.000,00 recuperados
- 40% dos próximos Cr\$ 2.000.000,00 recuperados
- 35% dos próximos Cr\$ 22.000.000,00 recuperados
- 25% de qualquer quantia recuperada acima de Cr\$ 25.000.000,00

Se o tipo de honorário não é a 1 ou 2, imprimir "DADOS INCORRETOS" e parar.

11. Escrever um programa para calcular a quantia total que se vai gastar para sustentar uma criança desde o nascimento até 18 anos, supondo um gasto de Cr\$ 3.700.000,00 por ano e uma taxa de inflação de 12% ao ano.

12. Ler dados que consistem dos seguintes campos para cada encomenda:

Descrição do Artigo; Preço por Unidade; Quantidade Encomendada.

Escrever um programa para imprimir a seguinte para cada encomenda:

Descrição do Artigo; Preço Total; Desconto; Preço Líquido.

O Preço Total é igual à quantidade Encomendada vezes Preço por Unidade.

O desconto é de 2% para encomendas com uma quantidade de 20 ou menos; 5% para encomendas com uma quantidade entre 20 e 50; 7% para encomendas com uma quantidade de 50 ou mais.

O Preço Líquido é igual ao Preço Total menos o Desconto.

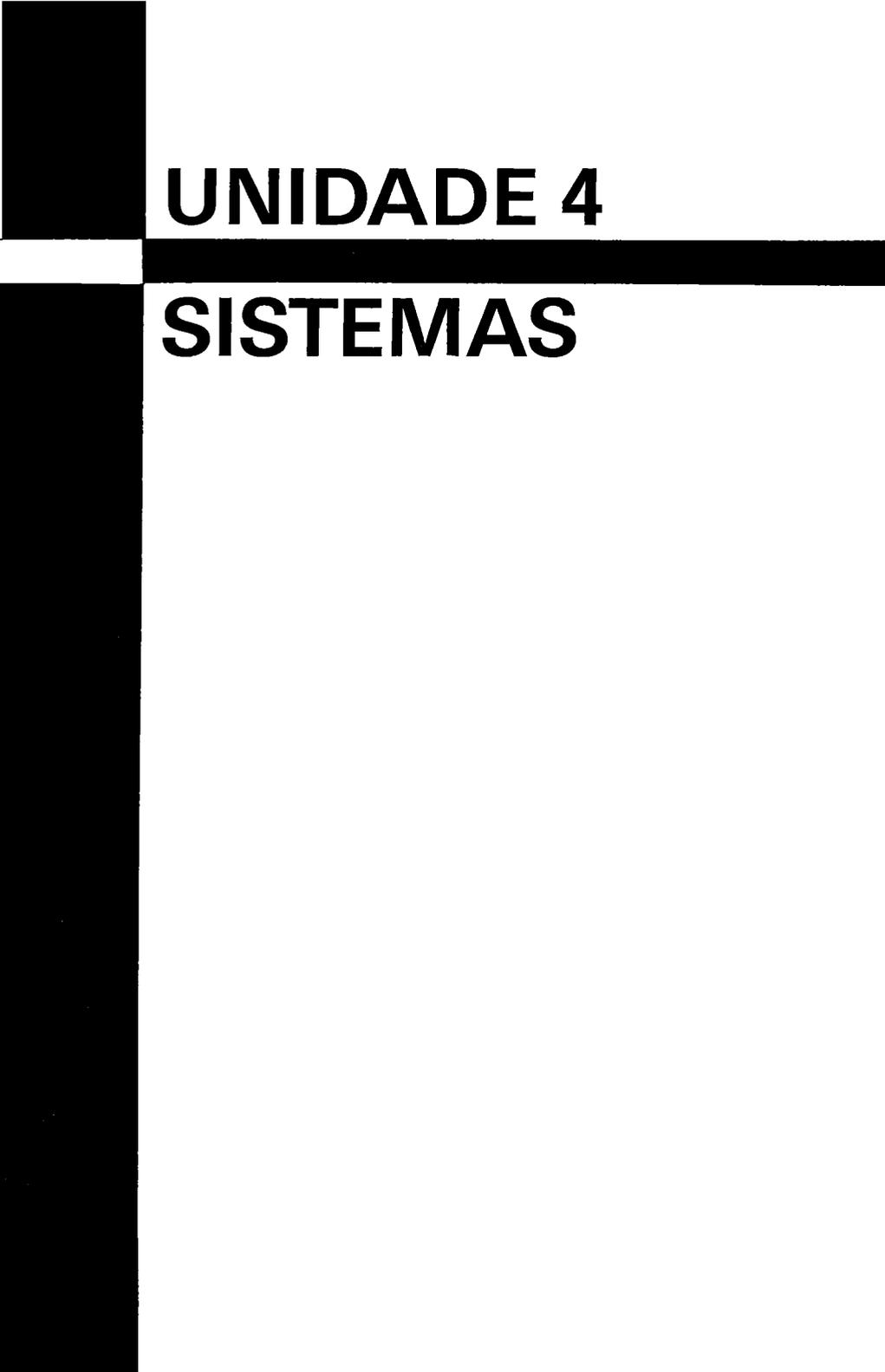
A saída impressa deve incluir cabeçalhos e resultados editados.

13. Escrever um programa usando a função INT para arredondar um campo denominado A para o inteiro mais próximo.

14. Escrever um programa usando a função TAB para imprimir uma linha diagonal de X em uma página.

```
X
 X
  X
   X
    X
```

15. Escrever um programa que simulará a decisão de cara ou coroa de uma moeda aleatoriamente, imprimindo os resultados de cada decisão – CARA ou COROA.



UNIDADE 4

SISTEMAS

9

SISTEMAS DE INFORMAÇÃO

- I. ANÁLISE E PROJETO DE SISTEMAS
 - A. O SISTEMA DE NEGÓCIOS
 - B. DESCRIÇÃO GERAL DE ANÁLISE E PROJETO
 - C. ETAPAS ENVOLVIDAS NA ANÁLISE E PROJETO
 - D. CICLO DE VIDA DOS SISTEMAS
 - E. REQUISITOS PARA A FUNÇÃO DE ANALISTA DE SISTEMAS
- II. PROBLEMAS COM SISTEMAS DE NEGÓCIOS COMPUTADORIZADOS
 - A. FALTA DE INTEGRAÇÃO ADEQUADA ENTRE SISTEMAS DENTRO DE UMA COMPANHIA
 - B. FALTA DE PADRONIZAÇÃO
 - C. COMUNICAÇÃO INEFICIENTE ENTRE ANALISTA E USUÁRIO
 - D. CONTROLES INADEQUADOS
- III. ESTUDO DE VIABILIDADE DO EQUIPAMENTO
 - A. INTRODUÇÃO
 - B. EQUIPE DE VIABILIDADE
 - C. OPÇÕES PARA ADQUIRIR EQUIPAMENTO
 - 1. Aquisição através do Fornecedor de Equipamento Específico (FEE*)
 - 2. Aquisição através de Plano de Aluguel, Compra ou Arrendamento (*Leasing*)
 - 3. Aquisição de um Computador para Uso Geral ou Sistema Completo
 - 4. Uso de Serviços Externos de Computador como Alternativa para a Computação Interna
- IV. SISTEMAS DE INFORMAÇÃO DE GERÊNCIA
 - A. UMA ALTERNATIVA PARA SISTEMAS CONVENCIONAIS
 - B. CARACTERÍSTICAS DO SIG** (Sistema de Informação de Gerência)
 - 1. Banco de Dados
 - 2. Sistema de Gerência de Banco de Dados (SGBD)***
 - 3. Dicionário de Dados
 - C. ADMINISTRADOR DE BANCO DE DADOS
 - D. LIMITAÇÕES DO SIG
 - 1. Custo

* Em inglês, OEM (Original Equipment Manufacturer).

** Em inglês, MIS (Management Information System).

*** Em inglês, DBMS (Data Base Management System).

2. Falta de Comunicação Adequada entre Analistas de Sistemas e Gerência
3. Problemas de Controle e Segurança

V. SISTEMAS DE SUPORTE DE DECISÃO: UMA NOVA PERSPECTIVA
PARA O SIG
TERMOS-CHAVE
QUESTIONÁRIO

I. ANÁLISE E PROJETO DE SISTEMAS

A. O SISTEMA DE NEGÓCIOS

Um *sistema de negócios* é um método organizado para executar uma função de negócios. Na maioria das organizações, alguns sistemas de negócios são computadorizados para maximizar o fluxo eficiente de informação, enquanto outros sistemas são em grande parte manuais.

Um *analista de sistemas* é um técnico de computador responsável pela computo-rização de operações de negócios. Analisa os procedimentos em curso e projeta sistemas mais econômicos e eficientes para melhor executar determinadas tarefas em uma companhia. O analista tem que trabalhar perto do usuário para que possa atingir este objetivo básico. A causa mais freqüente para o fracasso dos sistemas computadorizados em atingirem seus objetivos é a comunicação inadequada entre analista e usuário.

A maioria das faculdades oferece cursos de nível superior em análise e projeto de sistemas e existem livros técnicos que tratam exclusivamente deste assunto. Portanto, o nosso objetivo é apenas o de introduzir este assunto, não o de fornecer uma visão aprofundada do mesmo.

B. DESCRIÇÃO GERAL DE ANÁLISE E PROJETO

A gerência de uma organização encarrega o analista de sistemas de projetar um novo sistema ou buscar meios de melhorar um sistema já existente. Há duas razões principais para tal encargo:

1. Não existe um conjunto de procedimentos gerais para executar uma tarefa específica, mas a gerência gostaria de introduzir um procedimento computadorizado.

2. Existe um conjunto de procedimentos gerais (manual ou computadorizado) para executar uma tarefa específica, mas a gerência não está satisfeita com ele e quer uma reformulação.

A seguir fornecemos uma lista de razões comuns para a insatisfação da gerência com um sistema:

Por que os Sistemas de Negócio Fracassam

1. Incapacidade de processar dados de maneira que a informação seja oportuna.
2. Procedimentos manuais ineficientes e caros.
3. Alta taxa de erros.
4. Incapacidade do sistema manual de adaptar-se ao crescimento.
5. Incapacidade da gerência de obter as informações necessárias para tomar decisões.
6. Padrões e/ou controles inadequados.

C. ETAPAS ENVOLVIDAS NA ANÁLISE E PROJETO

Independentemente de ser o analista encarregado de instalar um sistema totalmente novo ou de reformular um já existente, as tarefas envolvidas são exatamente as mesmas. Incluem as seguintes etapas:

Tarefas da Análise e Projeto de Sistemas

1. Planejamento
Desenvolver um plano ou curso de ação que esboce os procedimentos a serem usados.
2. Análise e Projeto
 - a. Analisar o sistema de negócios existente para determinar as principais áreas problemáticas.
 - b. Projetar um novo sistema que opere de forma mais eficaz e eficiente que o anterior.
3. Implementar o novo sistema de modo que não existam maiores problemas de transição ou transformação.
4. Operar o novo sistema.
Ter certeza de que o novo sistema funciona adequadamente.

D. CICLO DE VIDA DOS SISTEMAS

Os quatro elementos discutidos acima são conhecidos como *ciclo de vida do sistema* (Figura 9.1). O termo "ciclo de vida" é usado com referência a um sistema porque todos os projetos são mais bem considerados enquanto *dinâmicos* ou sujeitos a transformação. Isto é, um sistema necessitará de uma revisão para atingir novos objetivos de maneira mais eficaz. Portanto, mesmo depois que um analista projeta um novo sistema, serão necessárias algumas modificações periodicamente.

Observe que o novo projeto geralmente envolve a computadorização de alguns aspectos do sistema de negócios. O analista é, acima de tudo, um técnico de computador que encara o computador como um instrumento que pode ser usado para tornar os sistemas de negócios mais eficientes. Entretanto, existem casos isolados nos quais um analista projeta um novo sistema de negócios *sem* recomendar o uso de computadores, porque acha que o custo, naquele caso específico, não se justifica.

O analista de sistemas pode ser:

1. Um funcionário da organização.
2. Um consultor externo que dá sugestões para a gerência.

Em ambos os casos, o papel do analista é essencialmente consultivo — pode apenas sugerir. A decisão de adotar um novo sistema de negócios fica a cargo da gerência.

A fase de análise e projeto é a parte mais difícil do trabalho do analista. São estas as tarefas que necessitam da aprovação da gerência. Depois que um conjunto de procedimentos e objetivos existentes foi analisado, o analista prepara um relatório à gerência, denominado *definição do problema*. Em termos exatos, este documento indica as áreas de problemas ou dificuldades gerais, e menciona de modo geral as partes essenciais que o analista irá projetar para eliminar tais problemas.

Se a gerência concordar, o analista projeta um novo conjunto de procedimentos. Antes que o novo projeto seja implementado, a gerência deve aprovar a sua execução bem como a proposta de custo.

E. REQUISITOS PARA A FUNÇÃO DE ANALISTA DE SISTEMAS

O analista de sistemas é geralmente um técnico de computador que tem competência técnica em *hardware* e *software*, experiência significativa na área de negócios e grande capacidade de comunicação, necessária para lidar com os usuários — isto é, tanto com o pessoal de operações quanto com a gerência. No decorrer de um projeto para um novo sistema os programadores ficam geralmente subordinados ao analista na tarefa de escrever o conjunto de instruções de computador para as aplicações necessárias ao novo projeto.

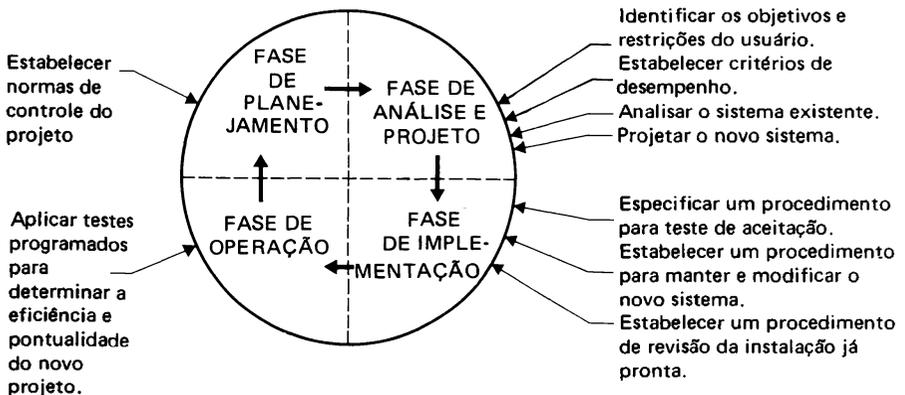


Figura 9.1
Ciclo de vida do sistema.

. PROBLEMAS COM SISTEMAS DE NEGÓCIOS COMPUTADORIZADOS

Apesar da abordagem sistemática usada pelos analistas ao projetar novos sistemas, há muitas vezes problemas sérios em novos projetos, dentre os quais destacamos os seguintes:

A. FALTA DE INTEGRAÇÃO ADEQUADA ENTRE SISTEMAS DENTRO DE UMA COMPANHIA

Visto que cada sistema é projetado independentemente quando se usa uma abordagem tradicional de sistemas, muitas vezes ocorre uma duplicação de esforços e uma falta de integração entre os sistemas. Como resultado, a gerência de alto nível às vezes não consegue obter uma visão completa das operações de toda a companhia. Os sistemas de informação de gerência minimizam este problema, mas também têm desvantagens inerentes, tais como custo e o longo tempo necessário para o planejamento.

B. FALTA DE PADRONIZAÇÃO

Pelo fato de cada sistema funcionar independentemente, em geral há poucas normas abrangendo toda a companhia com referência às aplicações de computador. Este problema também pode ser minimizado se houver uma abordagem integrada com relação ao projeto de sistemas.

C. COMUNICAÇÃO INEFICIENTE ENTRE ANALISTA E USUÁRIO

Conforme foi visto, um projeto de sistemas eficiente requer comunicação efetiva entre analista e usuários. Os usuários são os gerentes e o pessoal de operação. Se a comunicação é inadequada, o sistema fracassará e os usuários não seguirão a estrutura estabelecida.

Para minimizar este problema, os usuários devem entender de processamento de dados e os analistas devem ter experiência considerável na área de negócios. Na verdade, alguns especialistas sugerem que tanto o analista como o usuário sejam responsáveis pelo novo projeto (Figura 9.2).

Vários estudos foram realizados para determinar com precisão as áreas específicas que retardam a implementação de um sistema ou provocam insatisfação do usuário. A Figura 9.3 ilustra os resultados deste estudo.

D. CONTROLES INADEQUADOS

Mesmo os melhores sistemas podem eventualmente tornar-se pouco confiáveis se os controles adequados não forem implementados. Estes controles devem cumprir os seguintes objetivos:

Objetivo dos Controles

- 1. Minimizar os erros de entrada.**
- 2. Reduzir o risco de sabotagem.**
- 3. Minimizar o risco de manipulação intencional do sistema para fins de lucro pessoal.**

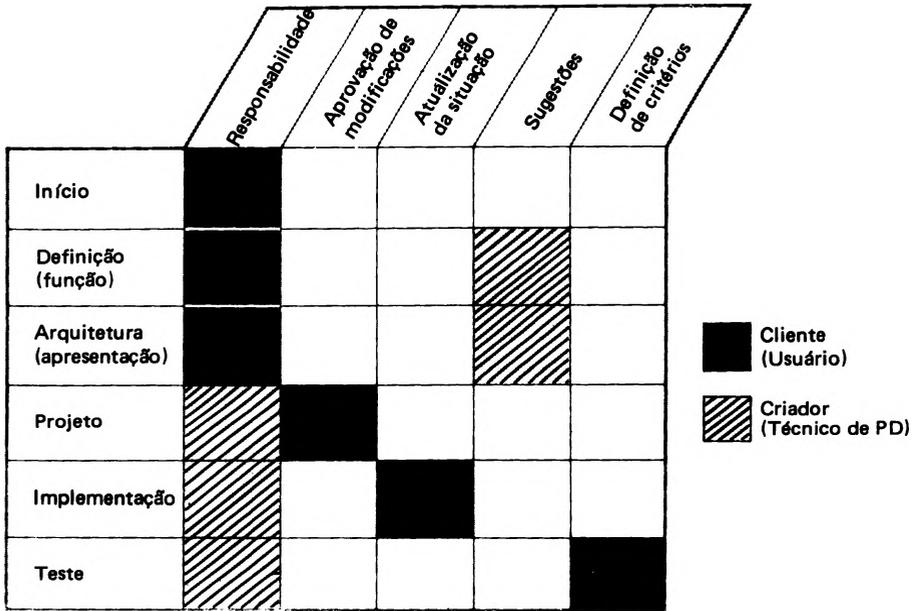
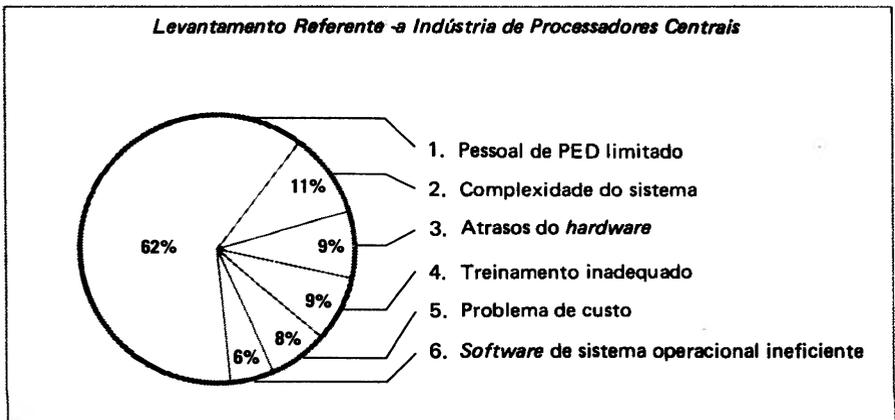


Figura 9.2
 Uma sugestão para a divisão de responsabilidades entre o técnico de PD e o usuário, durante o projeto e implementação de novos sistemas.

Figura 9.3
 Causas que retardam a implementação de novas aplicações.



Fonte: Datamation

I. ESTUDO DE VIABILIDADE DO EQUIPAMENTO

A. INTRODUÇÃO

Além de projetar sistemas de negócios, o analista de sistemas é freqüentemente chamado para indicar novos equipamentos de computador para uma companhia. Se for necessário um *hardware* alternativo ou adicional, deve-se realizar um *estudo de viabilidade* para determinar o equipamento que melhor satisfaz as necessidades da companhia como um todo.

B. EQUIPE DE VIABILIDADE

Um estudo de viabilidade é conduzido na maioria das vezes por uma equipe que consiste de um analista, um executivo da companhia, representantes dos departamentos e um consultor externo ou empregado da firma com experiência considerável em *hardware*.

As funções de uma equipe de viabilidade incluem:

Funções de uma Equipe de Viabilidade

1. Analisar as necessidades atuais.
2. Determinar se um novo equipamento é justificável.
3. Preparar um pedido de concorrência e distribuí-lo aos vendedores solicitando a apresentação de propostas.
4. Avaliar as propostas ou ofertas e tomar uma decisão.
5. Idealizar um plano de instalação.

C. OPÇÕES PARA ADQUIRIR EQUIPAMENTO

Há uma enorme variedade de opções disponíveis para a equipe de viabilidade.

1. Aquisição através do Fornecedor de Equipamento Específico (FEE)

Pode-se adquirir o equipamento através do fabricante ou de um FEE (*Fornecedor de Equipamento Específico*). Um FEE compra equipamento por atacado com descontos extraordinários, organiza um sistema de modo que ele possua o melhor *hardware* para uma aplicação específica, e fornece-o ao usuário por um preço bem mais econômico. Tanto os fabricantes como os FEE fornecem manutenção, *software* ou outro tipo de assistência necessária.

2. Aquisição através de Plano de Aluguel, Compra ou Arrendamento (*Leasing*)

A Figura 9.4 fornece uma descrição geral destes três métodos de aquisição. Observe que o plano de compra é o único meio de aquisição disponível para minicomputadores de baixo custo e para todos os microcomputadores.

3. Aquisição de um Computador para Uso Geral ou Sistema Completo

Se a programação vai ser executada pelo pessoal interno, os computadores de uso geral são os mais flexíveis. Se o sistema vai ser usado apenas para fins específicos, tais como processamento de palavras ou contas a receber, talvez seja melhor obter

COMPRA	
Vantagens	<ol style="list-style-type: none"> 1. Tende a ser o meio de aquisição mais barato. 2. Fornece ao usuário a possibilidade de um desconto considerável em termos de imposto.
Desvantagens	<ol style="list-style-type: none"> 1. É possível que a obsolescência do equipamento possa deixar o usuário com um <i>hardware</i> ultrapassado. 2. Grande investimento de capital. 3. A manutenção não está incluída.
ARRENDAMENTO	
Vantagens	<ol style="list-style-type: none"> 1. Tende a ser mais barato que aluguel. 2. Há uma obrigação por parte do fabricante ou seu representante de fornecer assistência ao sistema durante um certo tempo. 3. Opções de compra disponíveis. 4. Inclui manutenção.
Desvantagens	<ol style="list-style-type: none"> 1. Mesmo se estiver insatisfeito, o usuário é obrigado a manter o equipamento durante o tempo de arrendamento.
ALUGUEL	
Vantagens	<ol style="list-style-type: none"> 1. É o método que fornece maior flexibilidade ao usuário. 2. Proporciona a melhor proteção contra a obsolescência do equipamento.
Desvantagens	<ol style="list-style-type: none"> 1. Tende a ser caro. 2. Proporciona muito pouca proteção contra aumentos de custo. 3. Geralmente a assistência técnica do equipamento passa a ser um custo adicional.

Figura 9.4
Meios de aquisição de computador.

um *sistema completo* que possua todo o *hardware*, *software* e *firmware* necessários para processar essa aplicação. O usuário tem simplesmente que "apertar o botão" para fazer com que o sistema opere.

4. Uso de Serviços Externos de Computador como Alternativa para a Computação Interna

Pode ser mais barato e mais prático alugar um tempo de computador de uma grande firma de computador em vez de sujeitar-se a despesas e demora para adquirir o seu próprio computador. Se a organização no momento não possui pessoal de computação, a aquisição do equipamento significará, naturalmente, uma despesa adicional de contratação de pessoal. Os serviços externos de computador proporcionam assistência tanto para *hardware* como para *software*; isto é, não apenas operam programas de computador, como também os escrevem.

IV. SISTEMAS DE INFORMAÇÃO DE GERÊNCIA

A. UMA ALTERNATIVA PARA SISTEMAS CONVENCIONAIS

A análise e projeto de sistemas convencionais têm as seguintes desvantagens principais:

Desvantagens da Análise e Projeto de Sistemas Convencionais

1. Falta de Integração entre Sistemas.
Cada conjunto de procedimentos de negócios — contabilidade, folha de pagamento, pessoal, vendas etc. — é projetado independentemente.
2. Duplicação de Esforços.
Muitos sistemas poderiam facilmente compartilhar recursos, mas quando cada um é projetado independentemente nem sempre isto é possível.
3. Insatisfação da Gerência.
Visto que cada sistema tem seus arquivos e saídas individuais, a gerência não pode obter uma visão geral da companhia como um todo.

Um Sistema de Informação de Gerência (SIG) é uma alternativa para sistemas convencionais, projetado para eliminar as desvantagens mencionadas acima. O SIG abrange toda a companhia como um sistema principal. Deste modo, a gerência pode utilizar o sistema para obter informação integrada e facilitar o processo de tomada de decisão (Figura 9.5). Um SIG reduz a duplicação de esforços e fornece um conjunto de procedimentos integrados para toda a companhia.



Figura 9.5
As necessidades de cada subsistema são satisfeitas através de um processo de filtragem.

B. CARACTERÍSTICAS DO SIG (Sistema de Informação de Gerência)

1. Banco de Dados

Em vez de cada departamento ter o seu próprio conjunto específico de arquivos, todos os departamentos têm acesso aos dados de um *banco de dados* centralizado (Figura 9.6).

2. Sistema de Gerência de Banco de Dados (SGBD)

Se cada companhia resolver projetar todos os requisitos necessários para processar um banco de dados e mantê-lo atualizado, o projeto total será extremamente dispendioso e consumirá muito tempo. É típico das aplicações do SIG levarem anos

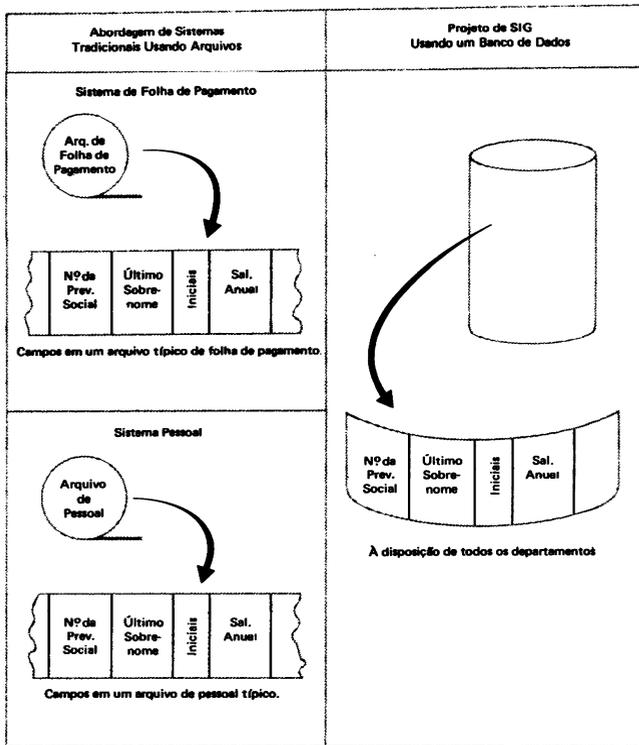


Figura 9.6

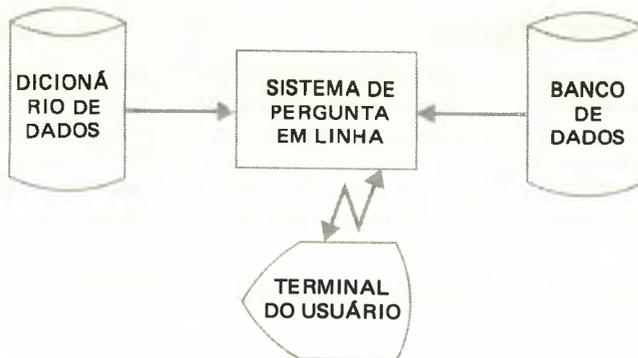
Comparação de um banco de dados para o armazenamento convencional de arquivo. (a) Abordagem de sistema tradicional usando arquivo. (b) Projeto de sistema de informação de gerência usando um banco de dados.

para serem projetadas e custarem milhões de dólares. Um *Sistema de Gerência de Banco de Dados (SGBD)* é um pacote fornecido pelo vendedor que pode ser usado para armazenar um banco de dados e para recuperar, solicitar e relatar dados. Um SGBD pode ser modificado pelos usuários conforme necessário. Os preços dos SGBD variam de dezenas de milhares de dólares a bem mais de US\$100,000 mas isto ainda custa muito menos do que projetar um sistema particular.

Os Sistemas de Gerência de Banco de Dados fornecem relatórios programados e permitem aos usuários pesquisar a situação do banco de dados "a pedido". O usuário acessa o banco de dados utilizando uma *linguagem de solicitação de dados* "acessível ao usuário", isto é, uma linguagem fácil de se usar.

3. Dicionário de Dados

Visto que todos os departamentos de usuários terão acesso a um banco de dados centralizado, é imprescindível que se estabeleça um único método padronizado para a referência aos elementos de dados. Um *dicionário de dados* fornece informação descritiva sobre os elementos de dados armazenados no banco de dados.



1. O usuário solicita dados de um registro.
2. O sistema localiza o registro através de um dicionário de dados.
3. O registro é recuperado do banco de dados.
4. O registro é exibido no terminal do usuário.

Figura 9.7

Uso de um dicionário de dados.

O dicionário de dados torna-se assim a única fonte para toda a informação e documentação sobre a função de processamento de dados. Tal dicionário geralmente inclui um conjunto de controles que exige senhas para que se possa ter acesso aos dados vitais ou sensíveis (Figura 9.7).

C. ADMINISTRADOR DE BANCO DE DADOS

É um técnico de computador, responsável pela manutenção do banco de dados, e deve assegurar-se de que os recursos de dados sejam administrados com eficiência.

D. LIMITAÇÕES DO SIG

É evidente, portanto, que o SIG tem grande capacidade e potencial para a tomada de decisão e o estabelecimento de diretrizes em uma grande companhia. No entanto, há várias razões pelas quais o SIG às vezes deixa de preencher as expectativas.

1. Custo

Além do *software* necessário para um SIG, ele requer um sistema de computador sofisticado, com capacidade de multiprogramação, o que proporcionaria acesso a inúmeros usuários.

2. Falta de Comunicação Adequada entre Analistas de Sistemas e Gerência

Às vezes a gerência deixa de explicar as suas necessidades ao analista. Por outro lado, os analistas algumas vezes não entendem as necessidades da companhia. Em ambos os casos, se houver um problema de comunicação, o SIG projetado não será satisfatório.

3. Problemas de Controle e Segurança

É difícil manter a segurança de qualquer sistema de computador. Entretanto, o problema de segurança se multiplica quando um único banco de dados abrangente é colocado à disposição de um grande número de usuários. A segurança de um sistema pode ser frequentemente violada como resultado de um ato criminoso ou um simples acidente. Portanto, é imperativo que o SIG inclua controles de *hardware* e *software* adequados para evitar que o sistema seja utilizado sem a devida autorização.

V. SISTEMAS DE SUPORTE DE DECISÃO: UMA NOVA PERSPECTIVA PARA O SIG

Um *Sistema de Suporte de Decisão* (SSD)* é projetado especificamente para facilitar o processo de tomada de decisão da gerência. O SSD usa modelos matemáticos para prever o resultado de diversos cursos de ação alternativos. Por exemplo, um sistema de suporte de decisão pode ser usado para prever os resultados de várias promoções de vendas que estejam sendo consideradas por uma companhia. Os executivos terão então o recurso de perguntar ao SSD "o que acontecerá se" um determinado curso de ação for adotado, e isto lhes permitirá tomar decisões de modo mais eficaz.

Para satisfazer às necessidades de tomada de decisão da gerência, vários projetos novos de SSD incluem capacidades de suporte de decisão.

TERMOS-CHAVE

Administrador de banco de dados
Analista de Sistemas
Banco de Dados
Ciclo de vida dos sistemas
Definição de solicitação de dados
Dicionário de Dados
Estudo de viabilidade
Fornecedor de Equipamento Específico (FEE)
Sistema acessível ao usuário
Sistema completo
Sistema de Gerência de Banco de Dados (SGBD)
Sistema de informação
Sistema de Informação de Gerência (SIG)
Sistema de negócios

QUESTIONÁRIO

(As respostas para as questões de números pares acompanham o teste.)

A. Questões do tipo Verdadeiro-Falso

V F

- _____ 1. Um analista de sistemas é um técnico de computador cuja responsabilidade é a de analisar os procedimentos existentes na companhia e dar sugestões para melhoramento dos sistemas.

*Em inglês DSS (Decision Support System).

- _____ 2. Um estudo de viabilidade é um estudo realizado para determinar se um SIG é melhor que um sistema tradicional.
- _____ 3. Um FEE é um programador que escreve programas de alto nível.
- _____ 4. Se a programação vai ser executada por um funcionário interno, então um computador para propósitos gerais fornece maior flexibilidade.
- _____ 5. Um SIG às vezes falha porque proporciona integração entre sistemas.
- _____ 6. Um SIG armazena todos os dados em um banco de dados em vez de estabelecer arquivos individuais.
- _____ 7. Os sistemas de suporte de decisão usam modelos matemáticos para prever os resultados de cursos de ação alternativos.
- _____ 8. Para uma companhia com um grande número de pessoal de computação, um sistema completo é geralmente o mais adequado.
- _____ 9. Um ambiente de processamento de dados distribuído é útil para o SIG.
- _____ 10. Um administrador de banco de dados, como um supervisor, faz parte do sistema operacional.

B. Preencha as Lacunas

1. Em um projeto de SIG, o objetivo básico é satisfazer as necessidades do/a.....
2. Um SIG processa dados através da modalidade.....
3. A maioria dos sistemas de gerência de banco de dados possibilita ao usuário o acesso ao banco de dados usando uma linguagem.....
4. O sistema de *software* usado para proporcionar um SIG totalmente integrado denomina-se
5. Muitos projetos de SIG deixam de funcionar corretamente devido à comunicação inadequada entre.....
6. Em um projeto de SIG todos os subsistemas têm acesso aos dados de um..... centralizado.
7. Um..... adquire equipamento de computador por atacado e em seguida arrenda os sistemas para os usuários finais.
8. Um..... é um sistema fornecido por um vendedor em forma de pacote, incluindo todo o *hardware* e *software*.
9. Quando a gerência considera necessário um sistema de computador totalmente novo, realiza-se então um estudo.....
10. O método mais barato de aquisição de computador é..... que também requer o maior investimento de capital.

Respostas

(Fornecemos respostas somente para as questões de números pares.)

A. Questões do tipo Verdadeiro-Falso

2. F — um estudo de viabilidade determina a viabilidade de se adquirir um equipamento de computador.
4. V
6. V
8. F — os sistemas completos são usados por companhias com pouca capacidade de programação interna.

10. F — um administrador de banco de dados é uma pessoa responsável pela manutenção do banco de dados.

B. Preencha as Lacunas

- 2. em linha
- 4. Sistema de Gerência de Banco de Dados (SGBD)
- 6. Banco de dados
- 8. Sistema completo
- 10. Compra

10

SISTEMAS COMPUTADORIZADOS NA SOCIEDADE

- I. NOVOS USOS DO COMPUTADOR NA SOCIEDADE
 - A. INSTRUÇÃO COM ASSISTÊNCIA DO COMPUTADOR (IAC)
 - B. COMPUTADORES NA ÁREA DA SAÚDE
 - 1. Diagnóstico de Doenças
 - 2. Controle dos Sinais Vitais de um Paciente
 - 3. Sistemas de Informação Hospitalar
 - C. COMPUTADORES NA ÁREA DE CIÊNCIA E TECNOLOGIA
 - 1. O Computador como Instrumento Experimental
 - 2. Inteligência Artificial
 - 3. Robôs
 - D. COMPUTADORES NA ÁREA DE CIÊNCIAS HUMANAS
 - 1. Computadores nas Artes
 - 2. Computadores na Música
 - 3. Computadores em Bibliotecas
- II. QUESTÕES ATUAIS SOBRE COMPUTAÇÃO
 - A. COMPUTADORES E AUTOMATIZAÇÃO
 - B. CRIME POR COMPUTADOR E INVASÃO DA PRIVACIDADE
 - C. "TELECOMUTAÇÃO": UM NOVO RUMO PARA OS TRABALHADORES?
 - D. O QUE O FUTURO PROPORCIONARÁ?

TERMOS-CHAVE

QUESTIONÁRIO

I. NOVOS USOS DO COMPUTADOR NA SOCIEDADE

Este livro enfatizou, na sua maior parte, o uso de computadores na área de negócios. Nesta seção abordaremos o uso de computadores na sociedade.

A. INSTRUÇÃO COM ASSISTÊNCIA DO COMPUTADOR (IAC)

A **Instrução com Assistência do Computador (IAC)** descreve o uso de computadores para ensinar aos alunos uma enorme variedade de assuntos. A IAC utiliza tipicamente terminais de teclado de TRC (Figura 10.1). Estes terminais fornecem exibição instantânea, capacidade gráfica e têm a vantagem de eliminar o consumo de papel.

Os módulos de cursos com IAC são utilizados em uma grande variedade de matérias para alunos de todos os níveis. A Figura 10.2 mostra um exemplo de exibição visual de um módulo de curso com IAC. São os seguintes os usos da IAC:

Módulos IAC

1. **Exercício e Prática**
Ajuda os alunos a praticarem tabuadas de multiplicação, regras de subtração etc.
2. **Simulação**
Simula uma experiência da "vida real" para testar as reações dos alunos (i.e., simula um caso clínico para testar estudantes de medicina, simula uma experiência química para testar estudantes de química etc.).

As vantagens da IAC incluem as seguintes:

Vantagens da IAC

1. Serve como um instrumento auxiliar de ensino, reduzindo, conseqüentemente, a carga de trabalho do professor.
2. Permite que os alunos aprendam no seu próprio ritmo.
3. Melhora a compreensão dos alunos, pois permite que utilizem diversos módulos para explicar o mesmo assunto de maneiras diferentes.
4. Minimiza a pressão de rivalidade que normalmente ocorre em um ambiente de sala de aula.
5. Motiva os alunos que preferem a comunicação impessoal com uma máquina ao invés de um ambiente de sala de aula.
6. Possibilita a exibição de imagens gráficas e desenhos animados para ilustrar o material.

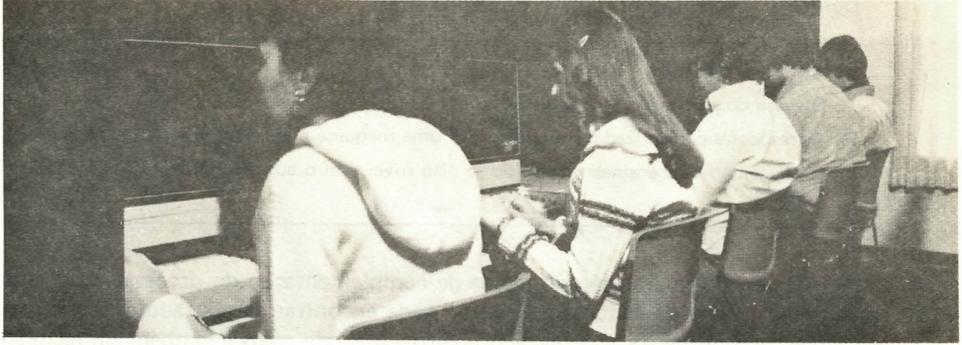
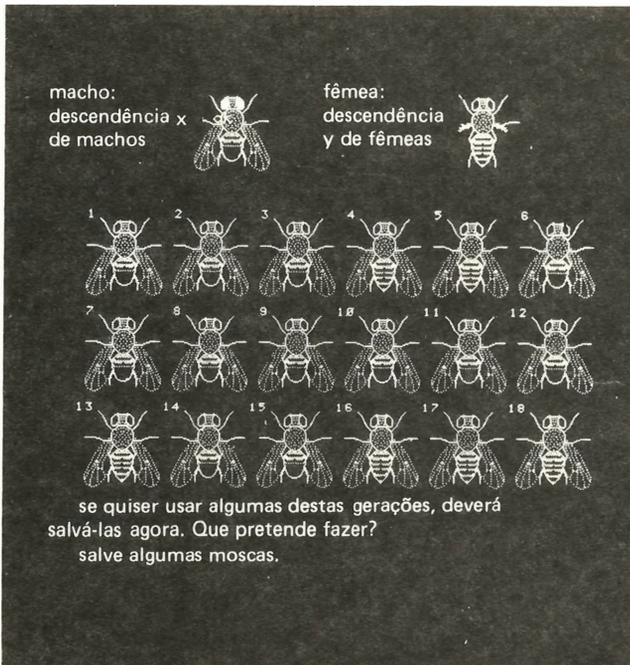


Figura 10.1
Alunos usando terminais para instrução com assistência do computador. (Cortesia de Goal Systems.)

Figura 10.2
Exibição visual exemplificando a IAC para o ensino de genética.
(Cortesia da Universidade de Illinois, Urbana.)



Dentre as desvantagens da IAC podemos citar:

Desvantagens da IAC

1. O custo do *software* é bastante alto.
2. As respostas padronizadas solicitadas por uma máquina podem inibir a criatividade.
3. A capacidade de ensinar conceitos de alto nível sem o auxílio do professor é questionável.

A IAC tornou-se um método eficaz de complementação do processo de ensino em todos os níveis da educação. O leitor pode encontrar computadores sendo usados em escolas primárias, escolas de primeiro e segundo graus e também em faculdades, como um instrumento auxiliar de ensino em matemática e em uma grande variedade de outras matérias.

A **Instrução Dirigida por Computador (IDC)** é uma técnica usada por instrutores para controlar a eficiência da IAC. As características da IDC incluem as seguintes habilidades:

Características da IDC

1. Testar o nível dos alunos e determinar o método de ensino adequado.
2. Manter registros do aproveitamento do aluno.
3. Determinar a confiabilidade das perguntas de testes.
4. Determinar a eficiência global de cada módulo de ensino.

B. COMPUTADORES NA ÁREA DA SAÚDE

1. Diagnóstico de Doenças

Os computadores são usados por médicos para auxiliar no diagnóstico de doenças. O médico insere dados com os sintomas de um paciente e o computador imprime uma lista de possíveis doenças, do material a ser examinado, ou outros testes de diagnóstico a serem realizados.

Os computadores são amplamente usados em laboratórios para analisar sangue, examinar amostras etc. Os instrumentos de perscrutação de TAC* (Tomografia Axial Computadorizada) e outros instrumentos sofisticados para diagnóstico possuem microprocessadores que podem detetar várias doenças.

2. Controle dos Sinais Vitais de um Paciente

Os pacientes hospitalizados em estado grave são freqüentemente ligados a um computador que controla ininterruptamente os seus sinais vitais e alerta a enfermeira se houver alguma anormalidade.

* Em inglês CAT scanners

3. Sistemas de Informação Hospitalar

Muitos hospitais possuem computadores que desempenham não somente as funções de controle de estoque, folha de pagamento e contabilidade, mas também mantêm registros integrados de dados para cada paciente. Um registro de dados de um paciente inclui:

- Histórico médico.
- Contabilidade.
- Necessidades laboratoriais.
- Necessidades farmacêuticas.
- Necessidades dietéticas.
- Instruções especiais.

Deste modo, controlam-se todos os dados relativos a um paciente, e o arquivo de pacientes pode ser usado para fornecer informações estatísticas sobre todos os pacientes (Figura 10.3).

C. COMPUTADORES NA ÁREA DE CIÊNCIA E TECNOLOGIA

1. O Computador como Instrumento Experimental

Quando usado como instrumento experimental, o computador funciona de maneira semelhante a um microscópio, termômetro ou barômetro. Isto é, possibilita aos cientistas avaliarem os resultados de uma experiência (Figura 10.4).

Os computadores fornecem ao cientista capacidade de computação em alta velocidade, sem a qual certas aplicações não seriam viáveis. Considere o gigantesco complexo de computador da National Aeronautics and Space Administration (NASA) projetado para determinar os parâmetros para viagens espaciais (Figura 10.5). Outra maneira de se usar computadores como instrumentos científicos é para a manutenção de dados na análise de cromossomos. Foram também desenvolvidos simuladores de vôo por computador para testar a viabilidade de novos projetos de aeronaves (Figura 10.6).



Figura 10.3

Exemplo de um sistema computadorizado de informação hospitalar. (Cortesia da IBM.)



Figura 10.4
Equipamento de computador que pode ser usado para análise científica. (Cortesia da Tektronix.)

2. Inteligência Artificial

Inteligência artificial é um campo científico que tenta esclarecer a natureza do pensamento através da simulação do processo de raciocínio. Foram criados programas de computador para solucionar problemas de raciocínio, com a finalidade de demonstrar que o processo de raciocínio se dá de acordo com um conjunto de regras sistemáticas. Muitas vezes projetam-se programas de jogos para demonstrar que o raciocínio humano é totalmente sistemático. A inteligência artificial é um assunto bastante controverso pois muitas pessoas acreditam que intuição, julgamento, expe-



Figura 10.5
Uma instalação de computador na National Aeronautics and Space Administration. (Cortesia da NASA.)

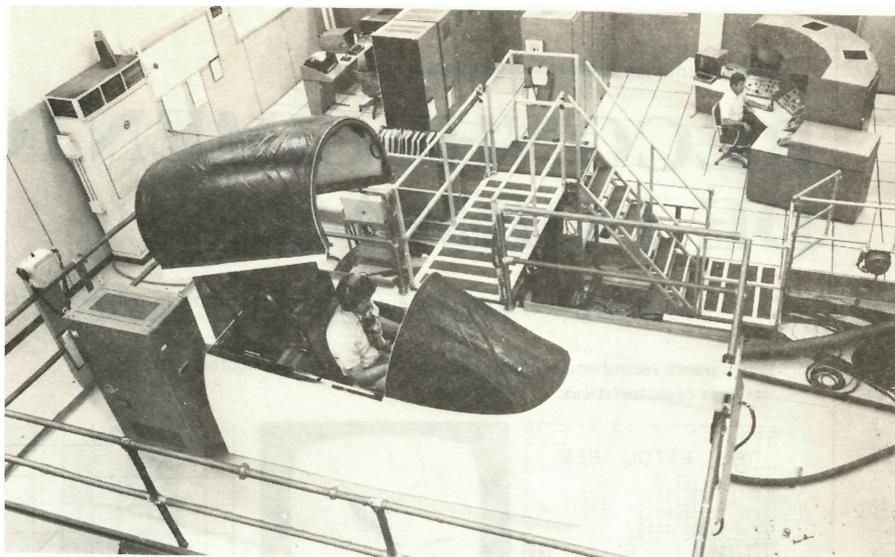


Figura 10.6
Simulador de voo. (Cortesia da Goodyear Aerospace.)

riência e emoção — todos eles fatores não mensuráveis — exercem influência sobre o pensamento; por este motivo, acham que a inteligência artificial está destinada a fracassar. Entretanto, uma área de pesquisa em inteligência artificial que tem obtido sucesso considerável é a do reconhecimento de formas (Figura 10.7).

3. Robôs

O robô industrial está sendo usado com frequência cada vez maior para executar operações de linha de montagem em oficinas mecânicas e fábricas de automóveis (Figura 10.8). Os robôs são muitas vezes usados para executar tarefas mecânicas enfadonhas, que desagradam à maioria dos trabalhadores.

D. COMPUTADORES NA ÁREA DE CIÊNCIAS HUMANAS

1. Computadores nas Artes

Os computadores podem ser usados nas artes de duas maneiras básicas:

1. Como um aparelho que, dado um conjunto de regras básicas, pode gerar um produto artístico, como por exemplo uma fotografia.
2. Como um instrumento auxiliar no projeto ou estruturação de um produto final.

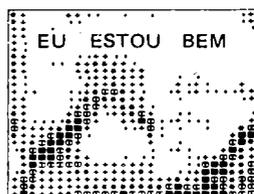
A Figura 10.9 fornece um exemplo de arte gerada por computador.

Quando um computador é usado para o

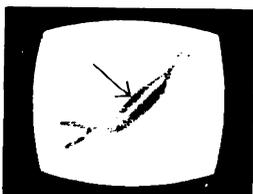
RECONHECIMENTO DE FORMAS

A identificação e classificação automática de formatos, formas e relações.

O programa reconhece objetos, eventos ou fatores significativos através das suas características.



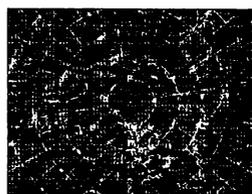
Uma máquina pode ouvir uma frase e identificar as palavras,



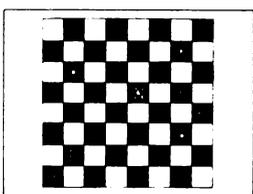
ou ler uma fotomicrografia de cromossomos e identificar uma anormalidade.



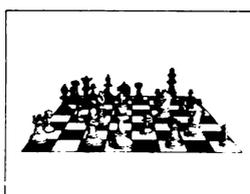
ou examinar uma impressão digital e identificar o seu dono,



ou perscrutar fotografias de satélite e localizar um possível furacão,



ou examinar uma tábua de jogo de xadrez e através dela determinar



uma boa jogada.

Todos estes são exemplos de reconhecimento de formas.

Figura 10.7

Exemplo de reconhecimento de formas. (Cortesia da IBM.)

2. Computadores na Música

Os sintetizadores de música usados juntamente com mini ou microcomputadores podem ser usados para gerar música, testar várias composições musicais e imprimir música por computador.

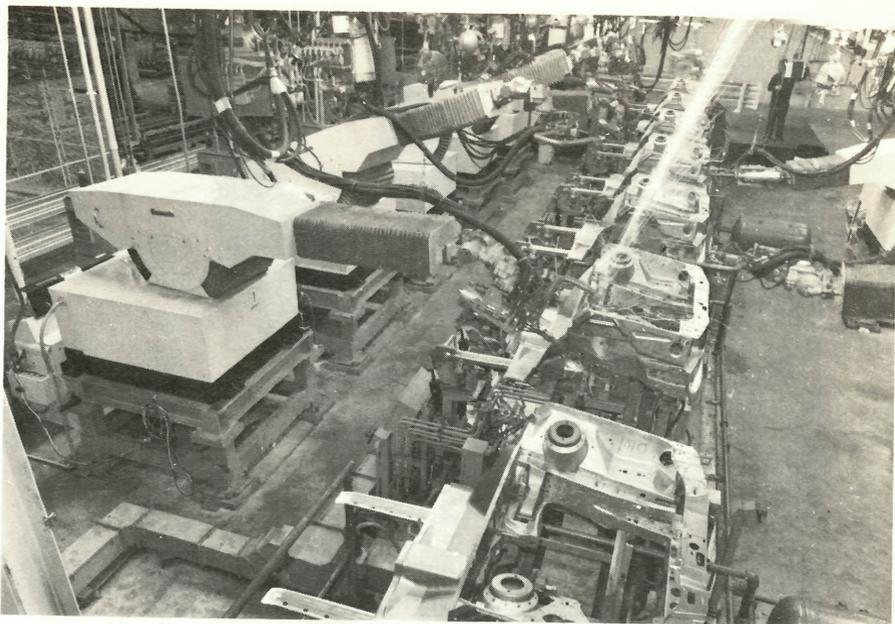


Figura 10.8

Robôs trabalhando em uma oficina de montagem da Ford Motor Company. Cada robô é usado para executar uma média de sete soldas por ponto na parte dianteira de um carro à medida que ele se desloca por uma linha de montagem. (Cortesia da Ford Motor Company.)

3. Computadores em Bibliotecas

Muitas bibliotecas estão associadas a uma variedade de bancos de dados de informação que mantêm registros de um grande número de artigos, livros e outros recursos. Alguns destes bancos de dados de informação contêm dados acerca de uma só disciplina específica, como medicina, história ou direito.

I. QUESTÕES ATUAIS SOBRE COMPUTAÇÃO

A. COMPUTADORES E AUTOMATIZAÇÃO

Há bastante controvérsia sobre o resultado prático da computadorização na força de trabalho. Ainda não está claro se o efeito final da computadorização e automatização é o aumento ou a diminuição do número total de empregos.

Vários fatores devem ser considerados quando se tenta avaliar o impacto da automatização. Por um lado, os problemas do desemprego resultantes da automatização podem muitas vezes desaparecer se os empregados estiverem dispostos a submeter-se a um novo treinamento para executar tarefas relacionadas com o computador. No entanto, em muitos casos os empregados continuam a opor-se por medo de que a mecanização trará como resultado a perda de empregos. Mesmo que este medo não

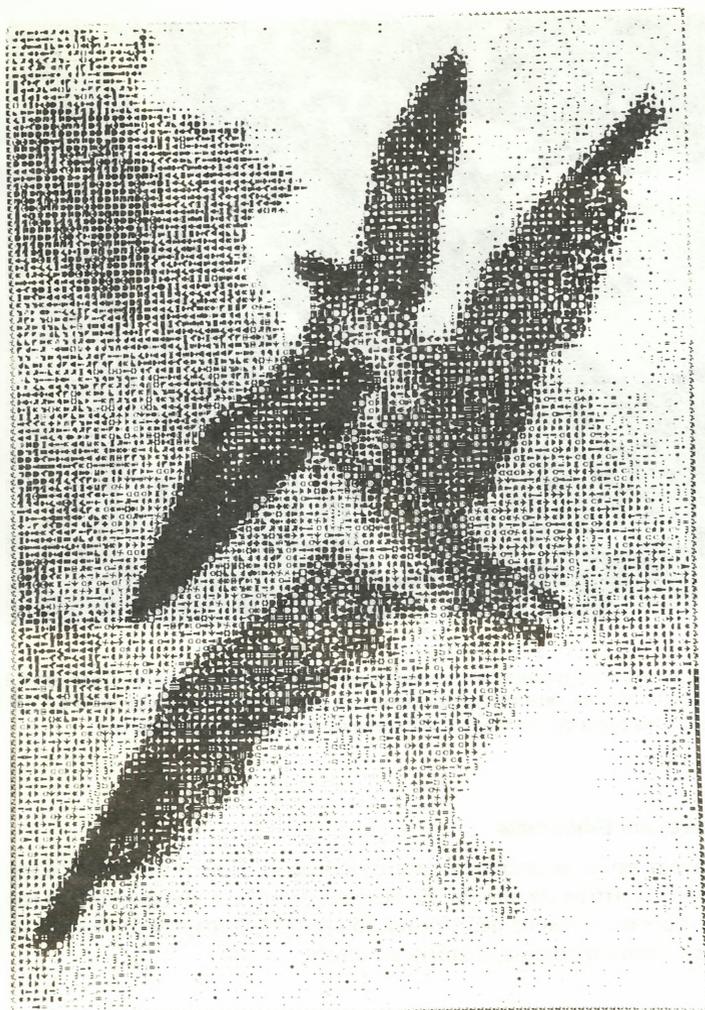


Figura 10.9
Uma fotografia gerada por computador. (Cortesia da Bell Labs.)

se justifique, sempre existe a possibilidade de que os empregados tenham que ser novamente treinados ou então demitidos como consequência da computadorização.

Muitas companhias perceberam que este é um problema sério e têm criado programas destinados a minimizar os problemas relacionados com o deslocamento de mão-de-obra.

B. CRIME POR COMPUTADOR E INVASÃO DA PRIVACIDADE

A possibilidade de invasão da segurança de um sistema de computador, deliberada

ou acidentalmente, é bastante grande. Estima-se que, em média, a perda resultante de roubos ou fraudes relacionados com o computador é de US\$600.000.

Dentre as técnicas usadas para minimizar os crimes por computador podemos citar as seguintes:

Técnicas Usadas para Minimizar o Crime por Computador

1. Códigos de autorização concedidos a usuários autorizados.
2. Senhas que são periodicamente modificadas.
3. Técnicas de escrita em cifra ou "mistura de dados" que tornam os dados ilegíveis se ocorrer a tentativa de interceptá-los ilegalmente.
4. Auditoria de PED executada por pessoal treinado.

Além dos crimes por computador, as companhias devem se preocupar com a possibilidade de que o computador possa invadir a privacidade individual. Os bancos de dados mantidos pelos governos estaduais e federais, hospitais, bancos etc. todos contêm informações importantes que, se divulgadas, podem significar uma invasão da privacidade do indivíduo. Várias leis federais e estaduais foram decretadas* sobre a questão da privacidade. Por exemplo, o Decreto sobre o Direito à Privacidade Financeira, de 1979, limita o tipo de informação pessoal que as agências governamentais podem obter de instituições financeiras, e exige que o indivíduo seja notificado antes que qualquer informação pessoal possa ser divulgada.

C. "TELECOMUTAÇÃO": UM NOVO RUMO PARA OS TRABALHADORES?

Com a proliferação do teleprocessamento e dos sistemas em linha, tornou-se bastante comum para técnicos de computador e usuários trabalharem em casa, usando um terminal para interação com o local de trabalho. Os benefícios deste sistema denominado "telecomutação" são os seguintes:

Vantagens da Telecomutação

1. Reduz a necessidade do transporte de massa.
2. Reduz a necessidade de energia no local de trabalho e nos sistemas de transporte.
3. Melhora as condições de trabalho.
4. Empregados mais satisfeitos têm uma produção mais eficiente.

D. O QUE O FUTURO PROPORCIONARÁ?

A Terceira Onda (The Third Wave), de Alvin Toffler, e *A Sociedade Telemática (The Telematic Society)*, de James Martin, fizeram previsões sobre o impacto dos computadores no futuro. É indiscutível que os computadores serão usados, no futu-

* Tais leis se aplicam aos EUA.

ro, praticamente em todas as casas. Além disso seus usos serão ilimitados. Por exemplo, será possível uma pessoa telefonar para o seu computador doméstico, de qualquer parte do mundo, e instruí-lo para preparar a casa para a sua chegada: ligar o aquecedor, o forno, as luzes etc. Já é possível usar um computador doméstico para se obterem preços de mercadoria, reservas de hotel, vendas a varejo, notícias e assim por diante.

É evidente que a chamada revolução do computador tem tido, e continuará tendo, um profundo impacto não somente na área de negócios mas na sociedade em geral, mas deixamos para os futurólogos a pressão dos efeitos desta revolução e do aspecto que ela tomará nos próximos anos.

TERMOS-CHAVE

Crime por computador
Instrução com Assistência do Computador (IAC)
Instrução Dirigida por Computador (IDC)
Inteligência artificial
Reconhecimento de formas
Robôs
Sistema de informação hospitalar
Telecomutação

QUESTIONÁRIO

(As respostas para as questões de números pares acompanham o teste.)

A. Questões do Tipo Verdadeiro-Falso

- | V | F | |
|-------|-------|---|
| _____ | _____ | 1. A IAC não é especialmente útil para a aprendizagem através de exercício ou prática. |
| _____ | _____ | 2. Um dos possíveis problemas com computadores é que eles podem ser desumanizadores. |
| _____ | _____ | 3. Os TRC são freqüentemente usados para IAC porque fornecem <i>hard copy</i> . |
| _____ | _____ | 4. Uma possível desvantagem da IAC é que as respostas padronizadas, exigidas pelo computador, podem inibir a criatividade do aluno. |
| _____ | _____ | 5. É possível que os computadores possam algum dia substituir os médicos no tratamento de pacientes. |
| _____ | _____ | 6. Os simuladores de voo por computador foram desenvolvidos para testar a viabilidade de novos projetos de aeronaves. |
| _____ | _____ | 7. É provável que, sem o auxílio dos computadores, não teria sido possível a aterrissagem de um foguete na lua. |
| _____ | _____ | 8. Alguns cientistas desenvolveram programas de jogos sofisticados, numa tentativa de demonstrar que o raciocínio humano é totalmente sistemático. |
| _____ | _____ | 9. Alguns cientistas acreditam que a inteligência artificial não resultará em nenhuma descoberta útil porque o pensamento humano não pode ser medido de modo sistemático. |
| _____ | _____ | 10. Os robôs têm sido usados com freqüência cada vez maior para substituir os funcionários administrativos. |

B. Preencha as Lacunas

1. O termo usado para descrever o processo de uso de um terminal em casa para interagir com o local de trabalho é
2. A de PED executada por contadores experientes pode minimizar as fraudes por computador.
3. Muitos computadores usam técnicas para tornar incompreensíveis os dados de computador transmitidos por uma linha telefônica, no caso de ocorrer a instalação de uma conexão secreta para interceptá-las.
4. O está sendo usado com frequência cada vez maior para executar operações de linha de montagem em oficinas mecânicas e fábricas de automóveis.
5. é um campo científico que tenta esclarecer a natureza do pensamento através da simulação do processo de raciocínio.
6. Uma área de pesquisa em inteligência artificial que tem obtido sucesso considerável é a do reconhecimento de
7. Um é um sistema que não só executa as funções de controle de estoque, folha de pagamento e contabilidade em um hospital, mas também mantém registros integridade dos dados de cada paciente.
8. é uma técnica usada por instrutores para controlar a eficiência de uma IAC.
9. IAC é uma abreviação para
10. Os terminais de com capacidades de fornecer imagens gráficas são usados com frequência em IAC porque proporcionam divertimento e exemplos ilustrativos que ajudam a reforçar o material apresentado.

Respostas

(Fornecemos respostas somente para as questões de números pares.)

A. Questões do Tipo Verdadeiro-Falso

2. V.
4. V.
6. V.
8. V.
10. F.

B. Preencha as Lacunas

2. auditoria
4. robô industrial
6. forma
8. Instrução Dirigida por Computador (IDC)
10. Tubo de Raio Catódico (TRC)

GLOSSÁRIO

A

Acesso Aleatório – A capacidade de acessar aleatoriamente um registro em um arquivo, sem necessidade de pesquisar todo o arquivo seqüencialmente.

Acoplador Acústico – Dispositivo conectado a um terminal que possibilita o acesso do terminal à UCP, usando qualquer telefone padrão.

Ada – Uma nova linguagem de programação que incentiva o projeto de programa estruturado; apoiada ativamente pelo Departamento de Defesa.

Administrador de Banco de Dados – A pessoa responsável pela manutenção do banco de dados e pela segurança de que os recursos de dados estão sendo administrados eficazmente; responsável também pelo planejamento adequado de controles de segurança e pela proibição do uso não autorizado do banco de dados.

Analista de Sistemas – O técnico de computador responsável pelos sistemas ou procedimentos eficientes e econômicos para executar com mais eficácia as tarefas determinadas em uma companhia.

APL – APL é um acrônimo para *A Programming Language* (Uma Linguagem de Programação); é uma linguagem de programação interativa de alta potência.

Arquivo – Uma coleção de registros individuais que são considerados como uma unidade. Um arquivo de folha de pagamento, por exemplo, refere-se a uma coleção completa de registros de empregados de uma companhia.

Arquivo de Reserva – Uma cópia de um arquivo pode ser usada para recriar o arquivo original no caso deste ser avariado, perdido ou roubado.

Arquivo Mestre – Um arquivo principal usado para manter registros para uma determinada aplicação.

ASCII – Um código comum de computador usado para comunicação de dados; uma abreviação para American Standard Code for Interchange of Information (Código Padrão dos EUA para Intercâmbio de Informação).

Atualização – O processo pelo qual se atualiza um arquivo de dados.

Auditor de PED – A pessoa responsável pelas operações de processamento de informação de auditoria.

B

Banco de Dados – Um arquivo de dados individual e principal que a companhia mantém e que pode ser acessado por cada departamento. Este banco de dados é muito mais eficiente do que cada departamento manter o seu próprio arquivo, um procedimento que, muitas vezes, resulta na duplicação de esforço e falta de controle adequado.

BASIC – Uma linguagem de programação simbólica de alto nível; um acrônimo para *Beginner's All-Purpose Symbolic Instruction Code* (Código de Instruções Simbólicas de Caráter Geral para Principiantes); semelhante à linguagem FORTRAN; ideal

para programação com o uso de terminais, mini e microcomputadores.

Bastão de Comando – Um dispositivo que pode ser usado para fazer modificações na tela de um TRC; comumente usado em jogos de computador.

Bit – Uma abreviação para *Binary Digit* (Dígito Binário). O termo refere-se à representação de dados na forma binária, como uma seqüência de ligado-desligado ou dígitos 1-0.

Byte – Um determinado número de *bits* que representa um caractere. O comprimento mais comum do *byte* é oito *bits*.

C

Cabo de Interconexão – As conexões internas usadas para transmitir dados de uma parte do computador para a outra.

Caixa Automático – Um caixa que presta serviços bancários 24 horas por dia.

Caneta Luminosa – Um dispositivo usado com terminais de TRC para fazer alterações nos dados exibidos na tela.

Caractere – Uma unidade de dados que consiste de um dígito, letra, ou símbolo especial.

Cartão Perfurado – Um cartão com perfurações para representar dados.

Cartucho de Fita – Um cartucho de fita é semelhante a uma minifita magnética e é usado para armazenar dados e programas que são processados seqüencialmente em uma modalidade por lote.

Cassete de Fita – Semelhante a um cartucho de fita. Um cassete de fita utilizado em gravações de música pode também ser usado para armazenar dados e programas.

Chip – Um dispositivo de memória feito de plaquetas finas de silicone e usado para compor a memória principal de um computador.

Choque de Cabeça de Gravação – A poeira e outras partículas às vezes forçam a colisão da cabeça de montagem de leitura/escrita de uma unidade de disco com uma superfície de disco, resultando na perda de todos os dados no disco.

Ciclo de Vida dos Sistemas – Consiste de quatro fases do desenvolvimento de um sistema: planejamento, análise e projeto, implementação, operação.

Circuito – Uma seqüência de etapas em um programa ou fluxograma a ser executada um determinado número de vezes.

Circuito DO – Uma seqüência de instruções a ser executada um determinado número de vezes, controladas por uma declaração DO. Os circuitos DO são usados em algumas linguagens de alto nível como FORTRAN.

Circuito Integrado – Consiste de inúmeros transistores e outros componentes eletrônicos em uma placa fina de silicone.

Classificação de Bolha – Um método de classificação pelo qual se comparam dois valores adjacentes em uma matriz, que são depois intercambiados de acordo com a seqüência desejada.

COBOL – Uma linguagem de programação simbólica de alto nível; um acrônimo para *Common Business Oriented Language* (Linguagens Comerciais); parecida com o inglês; ideal para atividades comerciais.

Codificação GO TO Menos – Outro termo para Programação Estruturada; a programação estruturada faz com que cada módulo funcione como uma entidade autônoma e evita desvios ou comandos "GO TO".

Código Ampliado de Caracteres Decimais Codificados em Binário para o Intercâmbio da Informação (EBCDIC) – Um código de computador usado para representar

caracteres; usado com maior freqüência na terceira e quarta gerações de computadores e em fitas de nove pistas.

Código de Autorização – Um código que permite o acesso do usuário de terminal a um determinado sistema de computador.

Código de Hollerith – O sistema de perfurações usado em um cartão para representar dados; assim denominado por Herman Hollerith.

Código de Produto Universal (CPU) – Um código de barra que aparece praticamente em todos os artigos do consumidor; pode ser lido por um dispositivo de perscrutação ou dispositivo de barra; usado em sistemas de ponto de venda.

Compatibilidade – Dois sistemas são compatíveis quando o *software* preparado para um computador pode ser utilizado por outro sem necessidade de uma conversão rigorosa.

Compilação (Compilador) – O processo de tradução que gera uma linguagem de máquina equivalente a uma linguagem de programação simbólica de alto nível; requer processamento mais complexo que uma montagem. O programa que executa esta operação denomina-se compilador.

Computador Dedicado – Um computador usado na maioria das vezes somente para uma aplicação específica.

Comunicação de Dados – A utilização das linhas de comunicação para manusear o fluxo de dados entre os terminais remotos e uma instalação central de computador.

Conceito de Programa Armazenado – Este conceito faz com que os programas bem como os dados sejam armazenados na UCP.

Conectado por Fio – Um dispositivo, como um terminal, pode ser ligado por cabo diretamente à UCP; deste modo, não se necessita de linhas telefônicas para transmitir mensagens de localizações remotas a uma UCP.

Constante – Um valor fixo que é parte de um programa; este valor não se altera durante a execução do programa.

Correio Eletrônico – O uso de dispositivos ou terminais telefac-símile para transmitir informação de uma localização para outra.

CP/M – Um acrônimo para *Control Program for Microprocessors* (Programa de Controle para Microprocessadores); um sistema operacional em discos produzido pela Digital Research Company, usado por muitos microcomputadores.

Crime por Computador – O uso de um computador para desviar fundos ou defraudar uma organização.

D

Definição de Problema – Um relatório à gerência, preparado por um analista de sistemas depois de analisar um conjunto de objetivos e procedimentos existentes.

Depurar – Eliminar os erros ou “falhas” de um programa.

Desvinculação – Um usuário pode adquirir cada parte do *hardware* ou *software* separadamente, em vez de ter que adquirir tudo como parte de um pacote.

Desvio – Uma transferência de controle em um programa ou fluxograma.

Diagramas HIPO – São diagramas orientados para sistemas que enfocam a relação hierárquica entre entrada, processamento e saída.

Dicionário de Dados – Fornece uma informação descritiva sobre as unidades de dados armazenados no banco de dados; é a única fonte para toda a informação e documentação sobre a função de processamento de dados.

Disco de Densidade Dupla — Um disco flexível que pode armazenar o dobro do número de *bits* e tem o dobro da capacidade de armazenamento de um disco comum.

Disco de Dupla Face — Um disco flexível que pode ler e registrar dados em ambas as faces; duplica a capacidade de armazenamento.

Disco de Winchester — Um disco rígido disponível para mini e microcomputadores.

Disco Flexível — Um disco flexível é uma versão em miniatura dos discos magnéticos maiores de cabeça fixa. É usado em mini e microcomputadores para armazenar dados e programas. Um disco flexível usa o método de processamento de acesso aleatório.

Disco Magnético — Um meio de entrada/saída que pode ser usado para armazenamento em grande volume e alta velocidade, onde os dados são armazenados sob forma de *bits* magnetizados. Um disco tem a característica adicional de acessibilidade direta ou aleatória.

Dispositivo de Avanço de Tração — Um dispositivo de impressão em que o papel se move pelo avanço dos dentes de roda dentada que se encaixam nos orifícios correspondentes no papel.

Dispositivo de Reconhecimento de Caracteres em Tinta Magnética (DRCTM) — *Hardware* de computador capaz de ler caracteres impressos com tinta magnética, tais como caracteres de cheques.

Dispositivo de Residência do Sistema — Um dispositivo de memória auxiliar, como por exemplo um disco ou um tambor, que armazena a biblioteca de compiladores, montadores, intérpretes e outros programas, além do supervisor.

Dispositivo de Reconhecimento Ótico de Caracteres (ROC) — *Hardware* de computador capaz de ler documentos manuscritos ou datilografados.

Documento de Retorno — O cartão de saída produzido pelo computador algumas vezes pode ser reinserido como entrada.

E

EBCDIC — *Veja*: Código ampliado de caracteres decimais codificados em binário para o intercâmbio de informação (*Extended Binary Coded Decimal Interchange Code*).

Edição — Processo pelo qual se verifica se os dados inseridos no computador estão relativamente isentos de erro. As operações de edição minimizam os erros gerados pelo computador.

Entrada (E) — Dados de chegada lidos em um sistema de computador para processamento.

Entrada de Dados a Distância — A utilização de terminais para inserir dados a distância; estes dados são transmitidos automaticamente à UCP através de linhas de comunicação.

Equipamento Telefac-símile — Dispositivos usados para transmitir cópias de um documento para uma ou mais localizações remotas, em questão de minutos ou mesmo segundos. Estes dispositivos são usados para uma enorme variedade de aplicações de correio eletrônico.

Erro de Sintaxe — Todas as regras de programação que são violadas fazem com que o computador imprima uma mensagem de erro durante a fase de tradução do programa.

Estudo de Viabilidade — Um estudo usado para determinar o equipamento de com-

putador que a companhia deve escolher para satisfazer as suas necessidades de processamento de informação.

F

Firmware — *Hardware* pré-conectado que foi programado para executar funções específicas. O usuário pode muitas vezes adquiri-lo com um sistema de computador, algum *software* ou suporte de programa que está embutido no *hardware*.

Fita Magnética — Um meio de entrada-saída de alta velocidade, que armazena dados sob forma de *bits* magnetizados. É comumente usada para armazenar arquivos de grande volume.

Fluxograma de Programa — Uma representação pictórica da lógica a ser usada em um programa.

Formulário Contínuo — O formulário de saída geralmente produzido por uma impressora de computador. Apesar da possibilidade de um formulário contínuo ser separado em folhas individuais, ele é alimentado pela impressora em uma folha contínua para aumentar a velocidade e facilitar o processamento.

Formulários Pré-Impressos — Formulários que possuem cabeçalhos e outras informações de identificação já impressos; a informação produzida pelo computador é acrescentada a estes dados pré-impressos.

Fornecedor de Equipamentos Específico (FEE) — Organizações especializadas que compram muitos sistemas e dispositivos a preço de atacado, e fornecem os sistemas prontos aos usuários de PED.

FORTRAN — Linguagem de programação simbólica de alto nível; um acrônimo para *Formula Translator* (Tradutor de Fórmula); ideal para programas científicos ou matemáticos.

Função de Biblioteca — Funções em BASIC que possibilitam o programador de chamar rotinas pré-programadas que podem ser usadas com o programa.

Funções de Série — Permitem ao programador modificar, comparar e analisar um campo alfanumérico em um programa BASIC, através da segmentação do campo em caracteres individuais.

G

Gabarito de Programação de Fluxograma — Um instrumento especial de plástico ou metal usado para desenhar símbolos em um fluxograma. Cada símbolo representa uma operação específica.

H

Hard Copy — Um registro permanente dos dados de saída de um sistema de computador. Contrasta com *Soft copy* (saída de exibição visual).

Hardware — Equipamento físico usado conjuntamente com a UCP; o *hardware* constitui, coletivamente, um sistema de computador. Contrasta com *software*.

Hierarquia de Operações — A ordem de operações efetuadas pelo computador quando se executa uma instrução envolvendo cálculos.

I

Impressora de Avanço de Fricção — Um dispositivo de impressão que permite a inserção de folhas individuais bem como a impressão em formulários contínuos.

Impressora de Impacto — Um dispositivo que funciona como uma máquina de es-

crever; usa um martelo para bater um caractere contra uma fita de impressão que faz com que a imagem do caractere seja impressa.

Impressora de Letra de Qualidade – Uma impressora de impacto padrão que produz imagens nítidas no papel.

Impressora em Série – Um dispositivo semelhante a uma máquina de escrever que é usado basicamente para terminais interativos de impressão de teclado. Este tipo de impressora imprime um caractere de cada vez.

Impressora por Caracteres – Um dispositivo que utiliza uma roda, cadeia ou tambor de impressão que posiciona o caractere desejado no ponto exato e geralmente usa um martelo para imprimir a imagem deste caractere no papel.

Impressora por Linha – Um dispositivo de saída de computador que imprime uma linha de informação de cada vez. Contrasta com impressora por página ou por série.

Impressora de Matriz de Ponto – Um dispositivo que imprime imagens de caractere a partir de uma seqüência de pontos.

Impressora por Página – Uma impressora que pode imprimir uma página inteira de informação de uma vez.

Impressora Sem Impacto – Um dispositivo em um sistema de computador que usa métodos eletrônicos ou fotoelétricos para produzir relatórios em velocidades extremamente rápidas. Contrasta com impressora de impacto.

Início de Comunicação – O procedimento usado para estabelecer comunicação com o sistema de computador.

Instalação Centralizada de Processamento de Dados – Há uma instalação central de computador que fornece serviços de computação a todos os usuários.

Instalação Descentralizada de Processamento de Dados – Cada departamento possui a sua própria instalação de computador, que funciona com seu próprio equipamento e pessoal de apoio.

Instalação de Gerência – Uma companhia que é contratada para maximizar o uso eficiente de uma organização interna de computador.

Instrução com Assistência do Computador (IAC) – Uma técnica educacional que utiliza computadores; o computador geralmente imprime a informação textual em um terminal e em seguida faz uma série de perguntas para o aluno responder; dependendo da exatidão das respostas, o computador passará a um tópico mais avançado ou então repetirá o assunto.

Instrução Dirigida por Computador (IDC) – Uma técnica usada para controlar a eficiência da instrução com assistência do computador.

Instrumentos de Perscrutação de TAC – Um dispositivo de diagnóstico usado para produzir raios X seccionais dos órgãos internos de uma pessoa; um acrônimo para tomografia axial computadorizada.

Inteligência Artificial (IA) – Um campo científico relacionado com a simulação do processo de raciocínio para esclarecer a natureza do pensamento racional.

Interface – Um dispositivo que possibilita a comunicação da UCP com um terminal através de linhas de comunicação.

Intérprete – Um tradutor que compila cada instrução à medida que ela é inserida; contrasta com compiladores padrão que traduzem os programas somente depois que todas as instruções foram inseridas.

L

Leitora de Código de Barra – Um dispositivo de reconhecimento ótico que pode ler códigos de barra e transmiti-los à UCP.

Linguagem de Controle de Trabalho (LCT) – A linguagem na qual o programador comunica ao supervisor os requisitos para a execução do trabalho. As especificações da LCT incluem o tipo de tradução desejada, equipamento de entrada/saída a ser usado, indicadores de início e fim de dados etc.

Linguagem de Máquina – O código próprio de máquina; um programa escrito em linguagem de máquina não necessita do processo de tradução.

Linguagem de Solicitação de Dados – Uma linguagem de alto nível que permite aos usuários o acesso a um banco de dados.

Linguagem Simbólica – Usa códigos simbólicos de operação tais como ADD ou +, e endereços simbólicos como HOLD e SUM para representar posições reais de armazenamento.

Linhas de Comunicação – Linhas de telefone ou teletipo usadas em um sistema de comunicação de dados para conectar um terminal à UCP.

M

MAA (Memória de Acesso Aleatório) – Memória usada para armazenar programas e dados.

Máquina Perfuradora de Teclado – Um dispositivo semelhante à máquina de escrever, usado por um operador para perfurar dados em um cartão, geralmente de um documento fonte.

Matriz – Um método para armazenar dados usando uma variável subscriptada.

Megabyte (MB) – Um milhão de *bytes* de memória.

Memória Auxiliar – Uma unidade de memória adicional que completa a memória primária da UCP.

Memória de Acesso Aleatório – Veja: MAA.

Memória de Bolha Magnética – Consiste de pontos magnetizados em um filme fino de material semicondutor; os dados podem ser mantidos na memória mesmo se a força estiver desligada.

Memória de Leitura Somente – Veja: MLS.

MLS – Memória de Leitura Somente – A parte da memória de computador que contém funções pré-conectadas; não pode ser alterada por instruções programadas.

Memória de Núcleo Magnético – Um tipo de memória que consiste de núcleos magnéticos, ou minúsculos elementos de ferrite com formato de rosca, mais ou menos do tamanho de um grão de sal.

Memória Primária – A capacidade de memória de um sistema de computador que está localizada na UCP; contrasta com Memória Auxiliar.

Memória Principal – Veja: Memória primária.

Memória Virtual – Uma técnica para maximizar e otimizar a memória disponível usando técnicas de superposição.

Microcomputador – Um computador pequeno geralmente com TRC de teclado e memória de 16 a 32 K na sua forma básica; os preços variam a partir de várias centenas de dólares. A maioria dos computadores de "uso pessoal" são microcomputadores.

Microsegundo – Um trilionésimo (10^{-12}) de um segundo. Alguns supercomputadores podem operar dados em velocidades medidas em microssegundos.

Minicomputador – Um computador pequeno com a maioria das características iguais às de um computador de tamanho padrão; usado com frequência por compa-

nhas de pequeno porte, ou para redes de comunicação de dados em organizações maiores.

MLSP – Memória de Leitura Somente Programável; MLS que pode ser programada para aplicações específicas.

MLSPA – Memória de Leitura Somente Programável e Apagável. Um *chip* não é programável somente para incluir um conjunto de funções, mas o programa pode ser “apagado” ou superposto com outro programa através de um procedimento de microcódigo.

Modulador/Demodulador – Um dispositivo que modula e demodula sinais transmitidos por linhas de comunicação.

Montador – Um tipo de programa tradutor que é usado para traduzir linguagens de programação de baixo nível em linguagem de máquina.

MP/M – Uma versão do CP/M; um acrônimo para *Multiprocessing Monitor Control Program* (Programa de Multiprocessamento com Controle Monitorizado); um sistema operacional que possibilita que vários usuários operem programas simultaneamente usando uma UCP.

Multiprocessamento – O uso de duas ou mais UCP que estão conectadas.

Multiprogramação – Capacidade que tem o sistema de computador de processar dois ou mais programas simultaneamente; esta é uma característica comum dos sistemas de comunicação de dados e de tempo compartilhado.

N

Nanossegundo – Um bilionésimo (10^{-9}) de um segundo. A maioria dos computadores pode operar dados com velocidades medidas em nanossegundos.

Número Aleatório – Um número gerado aleatoriamente, usado por computadores para jogos, construção de modelo e simulação.

Número Binário – Um número representado por uma combinação de 0 e 1; perfeitamente adequado para uso em computadores, onde 0 representa o estado “desligado” e 1 “ligado”.

O

Operador de Computador – A pessoa que supervisiona uma operação de computador pressionando as teclas adequadas e controlando o *hardware*.

Operador de Entrada de Dados – A pessoa responsável pela inserção de dados nos cartões perfurados, terminal, fita ou disco, de modo que eles possam ser lidos pelo computador.

Organização Direta de Arquivo – Os registros de dados são acessados convertendo um campo codificado, através de um cálculo aritmético, em um endereço real que identifica superfície, pista e possivelmente cilindro ou número de setor.

Organização Relativa de Arquivo – Os registros podem ser armazenados em localizações relativas no arquivo. O campo codificado é usado para determinar cada localização relativa; o campo codificado faz a conversão diretamente para a posição real no disco.

Organização Seqüencial Indexada de Arquivo – Um método de organizar dados em um arquivo para facilitar o processamento em linha. O disco usa um índice para localizar os registros.

P

Pascal — Uma linguagem de programação que facilita o uso de técnicas de programação estruturada.

PERFORM ... UNTIL — Em programas COBOL estruturados, o desvio é substituído por PERFORM ... UNTIL. Esta instrução é usada para indicar uma seqüência a ser processada até que se encontre uma condição específica.

Pergunta-Resposta — Quando se usa um terminal para solicitar informação sobre o estado de um arquivo.

PL/1 — Linguagem simbólica de Programação de alto-nível; um acrônimo para *Programming Language 1* (Linguagem de Programação 1); reúne as principais características da FORTRAN e COBOL.

Processador Terminal — Os dados de uma rede de minicomputadores ou terminais são transmitidos em altas velocidades para um sistema de grande porte que coordena os dados.

Processamento de Dados Distribuído (PDD) — Um método de processamento destinado a incorporar as vantagens de ambos processamentos de dados centralizado e descentralizado. Um minicomputador ou terminal em uma localização remota é usado para processar dados e alimentá-los em uma unidade principal. Vários usuários têm acesso ao banco de dados usando uma rede de terminais e minicomputadores ligados a uma UCP principal.

Processamento de Informação — O processo de produção de informação significativa que a gerência pode usar para propósitos de tomada de decisão e o pessoal de operação, para as suas atividades diárias.

Processamento Eletrônico de Dados (PED) — Na nomenclatura atual, PED refere-se ao processamento de dados por computadores eletrônicos digitais.

Processamento em Linha — O processamento de dados com dispositivos tais como terminais que estão sob controle direto da UCP principal.

Processamento Fora de Linha — O processamento de dados que não está sob controle direto da UCP principal.

Processamento por Lote — O processamento de dados em grupos ou lotes, em oposição ao processamento de dados imediato.

Processamento Seqüencial — O método de processar dados seqüencialmente à medida que eles aparecem no meio de entrada. Este método é usado com maior freqüência em processamento de cartão e fita. Contrasta com Acesso Direto.

Programa — Uma seqüência de instruções que possibilita o computador de ler os dados de entrada, processá-los e convertê-los em dados de saída.

Programação Estruturada — Um termo usado para descrever uma técnica de programação eficaz que pode facilitar o processamento de programas em todas as linguagens; denominada programação GO TO menos.

Programador — O técnico de computador que escreve e depura o programa, ou conjunto de instruções, que opera em entrada e o converte em saída.

Programa Fonte — Um programa escrito em uma linguagem simbólica de programação; os programas fonte devem ser traduzidos antes de ser executados.

Programa Objeto — Uma linguagem de máquina equivalente a um programa fonte. Os programas objeto são os únicos que podem ser processados sem ser traduzidos previamente.

Protocolo — Os procedimentos de início de comunicação usados para acessar um sistema de computador.

Pseudocódigo — Este instrumento representa especificamente o fluxo lógico de um programa estruturado. É um instrumento de planejamento que utiliza um código semelhante a um código de programa para representar a lógica. É usado para aplicações que possuam procedimentos lógicos de controle que sejam complexos; nestes casos é mais útil uma abordagem estruturada.

R

Reconhecimento de Formas — A capacidade de um computador reconhecer objetos, eventos ou fatores significativos através de suas características. Por exemplo, um computador pode ser programado para reconhecer linguagem oral, impressões digitais, tipos de clima através de fotografias da terra tiradas por satélites.

Rede — Uma série de dispositivos de computador que operam de modo integrado.

Registro-Reboque — O último registro de um arquivo. Quando um arquivo é inserido em um terminal, este registro, às vezes denominado pseudo-registro, é usado para indicar o fim do arquivo.

Robôs — O uso de robôs industriais para executar operações de linha de montagem em oficinas mecânicas e fábricas de automóveis.

Rótulo — Usado para identificar arquivos em meios tais como fita e disco magnético. Existem dois tipos: (1) um rótulo gomado externo que é afixado na fita ou disco; (2) um rótulo programado que é incluído como parte do próprio arquivo.

RPG — Um gerador de programa de alto nível; um acrônimo para *Report Program Generator* (Gerador Automático de Programas); ideal para imprimir relatórios simples.

S

Saída (S) — Dados que foram processados por um sistema de computador; a informação obtida de um sistema de computador denomina-se saída.

Sinal de Dólar Flutuante — Quando um sinal de dólar é impresso como um caractere imediatamente à esquerda do primeiro dígito em um campo; \$15.26 em vez de \$ 15.26.

Sistema Acessível ao Usuário — Um sistema que é projetado para ser semelhante ao inglês e tão fácil de utilizar quanto possível.

Sistema Completo — Um microcomputador usado como um dispositivo para uso especial que não necessita de programação ou de especialista em computador para utilizá-lo; o sistema tem geralmente que ser pré-programado para a aplicação específica; um sistema de computador que inclui, na forma de pacote, *hardware*, *software* e *firmware* necessários ao processamento de uma aplicação específica.

Sistema de Computador — Consiste de uma unidade central de processamento, ou unidade principal, e dispositivos de entrada e saída.

Sistema de Gerência de Banco de Dados (SGBD) — O sistema *software* usado para fornecer um sistema de informação de gerência totalmente integrado; inclui o *software* necessário para armazenar, recuperar, solicitar e relatar dados.

Sistema de Informação — Um sistema computadorizado que combina as características do *hardware* e *software* em uma aplicação integrada.

Sistema de Informação de Gerência (SIG) — Uma abordagem de sistemas que considera os departamentos de negócios como partes integradas de um sistema completo, em vez de entidades separadas. Esta abordagem tem como objetivo facilitar o fluxo

de informação e fornecer um gerenciamento com maior poder de tomada de decisão.

Sistema de Informação Hospitalar — Os computadores são usados basicamente para propósitos de manutenção de registro. Contrasta com Sistemas de Informação Médica.

Sistema de Negócio — Um método organizado para realizar uma função negociável.

Sistema de Ponto de Venda (SPV) — Um avanço em teleprocessamento que permite aos balconistas atualizar automaticamente os arquivos de computador enquanto preparam os recibos de vendas.

Sistema de Processamento de Palavra — Um sistema que automatiza tarefas executadas por secretárias na maioria dos escritórios; em vez de datilografar documentos, relatórios e cartas usando uma máquina de escrever comum, a secretária usa um terminal que pode imprimir o documento, fazer as alterações adequadas quando necessárias e armazenar a informação para impressão posterior.

Sistema de Processamento em Tempo Real — Um sistema com capacidade de acessar e atualizar os arquivos de computador com rapidez suficiente para influenciar na tomada de decisão.

Sistema de Suporte de Decisão (SSD) — Um sistema que ajuda a prever o resultado das decisões propostas pela gerência; utiliza modelos matemáticos para prever o resultado das diversas alternativas de cursos de ação.

Soft Copy — Dados de saída de um sistema de computador sob a forma de exibição visual; não pode ser conservada a não ser que esteja ligada a uma impressora de *hard copy*.

Software — Suporte de programação que permite ao sistema de computador operar de maneira eficaz. Este suporte inclui supervisor, programas do usuário, programas utilitários etc.

Software de Aplicação — Programas destinados a satisfazer as necessidades dos usuários.

Software de Sistema Operacional — Consiste de programas que otimizam o uso eficiente do sistema de computador como um todo; manuseia automaticamente tarefas tais como compilação, escalonamento e controle de entrada/saída.

Subíndice — Um número usado para indicar a posição de um valor específico em uma lista ou tabela.

Sub-rotina — Uma seqüência de instruções que devem ser processadas em pontos diferentes de um programa.

Supervisor — Uma característica do sistema operacional; um programa especial que reside na UCP para controlar as operações de todo o sistema.

T

Tamanho de Memória — O número de posições de memória que podem ser acessadas em um sistema de computador.

Taxa de Bauds — A unidade usada para medir velocidades de transmissão de dados; geralmente medida em função de *bits* por segundo.

Telecomunicação — A possibilidade dos técnicos de computador e usuários trabalharem em casa, usando um terminal para interação com o local de trabalho.

Teleprocessamento — O uso combinado de equipamentos de comunicação, tais como linhas telefônicas, e equipamento de processamento de dados para transmitir e processar dados.

Tempo Compartilhado — O termo usado para descrever um sistema de computador que é compartilhado por vários usuários, geralmente através do uso de terminais. As companhias de pequeno porte que não têm recursos para alugar ou comprar os seus próprios computadores consideram este sistema vantajoso.

Terminal — O dispositivo usado em localizações a distância em um ambiente de teleprocessamento para inserir ou receber dados de uma UCP.

Terminal de Máquina de Escrever — Um terminal que tem um teclado e o equipamento para produzir saída de *hard copy* através de um dispositivo semelhante a uma máquina de escrever.

Terminal Inteligente — Um dispositivo de terminal que contém instalação embutida para executar funções lógicas tais como edição de dados de chegada, mesmo antes que os dados sejam transmitidos ao computador.

Terminal Manual — Um pequeno terminal utilizado para entrada de encomendas e uso do consumidor.

Transferência Eletrônica de Fundos (TEF) — Um sistema que elimina a necessidade de descontar cheques, transferindo automaticamente os fundos de uma conta para outra.

Tubo de Raio Catódico (TRC) — Um dispositivo de terminal que imprime mensagens em um tubo semelhante a um cinescópio; a saída é denominada "*soft copy*", pois não é mantida como um registro permanente.

U

Unidade Aritmética e Lógica — A unidade de uma UCP que executa operações e comparações aritméticas.

Unidade Central de Processamento (UCP) — O dispositivo de computador que controla as operações reais do sistema de computador; estas operações incluem entrada, transferência de dados, lógica/aritmética e saída, que fazem parte de cada programa do usuário.

Unidade de Controle — A parte de uma UCP que controla as operações de um computador.

Unidade de Entrada — Um dispositivo que lê dados de um formulário específico e os converte em pulsos elétricos, que são transmitidos a uma área de entrada na UCP.

Unidade de Reconhecimento de Voz — Um dispositivo capaz de interpretar a linguagem oral e transmitir os dados à UCP.

Unidade de Resposta Audível — Um dispositivo de saída que transmite mensagens em forma verbal; um computador é equipado com várias palavras ou frases chaves pré-gravadas que são extraídas, conforme necessário, para responder a perguntas específicas.

Unidade de Saída — Um dispositivo que recebe informação da UCP e converte os impulsos elétricos em uma forma de saída adequada.

Unidade Principal — Veja: Unidade Central de Processamento.

V

Variável — Em um programa BASIC, este é um campo a ser usado para conservar dados numéricos.

Variável de Série — Um campo em um programa BASIC que será usado para conservar caracteres alfanuméricos.

TRADUÇÃO DOS COMANDOS E DECLARAÇÕES

- ADD → SOMAR
- COMPUTE → COMPUTAR
- DATA → DADOS
- END → FIM
- END OF JOB → FIM DE TRABALHO
- ERROR → ERRO
- FOR → PARA
- FOR ... NEXT → PARA ... PRÓXIMO
- GOSUB → EXECUTAR SUB-ROTINA
- GOTO → IR PARA
- HOLD → MANTER
- IF → SE
- IF ... THEN → SE ... ENTÃO
- INPUT → ENTRADA
- LET → SEJA
- LIST → LISTAR
- LOAD → CARREGAR
- MOVE INPUT TO OUTPUT → TRANSFERIR ENTRADA PARA SAÍDA
- MULTIPLY → MULTIPLICAR
- NEXT → PRÓXIMO
- PERFORM ... UNTIL → EXECUTAR ... ATÉ QUE
- PRINT → IMPRIMIR
- PRINT USING → IMPRIMIR COM FORMATO
- READ → LER
- READ A CARD → LER UM CARTÃO
- READ A NAME → LER UM NOME
- REM → OBSERVAÇÃO
- RETURN → RETORNAR
- RUN → OPERAR
- SAVE → CONSERVAR
- START → INICIAR
- STEP → PASSO
- STOP → PARAR
- SUM → SOMAR
- WRITE → ESCREVER
- WRITE A LINE → ESCREVER UMA LINHA

ÍNDICE ANALÍTICO

- Acesso aleatório, e discos, 52
- Acoplador acústico, 73, 210
- Ada, 108-109, 210
- Adição, em BASIC, 137-138
- Administrador de Banco de Dados, 19, 28, 210
- Alinhamento de campos numéricos, em BASIC, 163-164
- Análise, sistemas, 184-185
- Apagamento de linha de programa em BASIC, 135
- APL, 106, 109, 210
- Aquisição de computador, métodos, 185-190
- Área de ciências humanas, computadores na, 203
- Armazenamento externo, e mini e microcomputadores, 84
- Arquivo de reserva, 50, 210
- Arquivo mestre, 48, 210
 - processamento, 50-54
- Arte, computadores em, 203, 206
- ASCII (Código americano para Intercâmbio de Informações), 35, 210
- Asterisco, impressão de, em BASIC, 165
- Auditor de PED, 19, 210
- Automatização administrativa, 76-78
- Automatização, e computadores, 205

- Banco de Dados, 191, 210
- Bastão de comando, 68, 211
- Biblioteca de funções, 99
- Bibliotecário, de fita e disco, 51
- Bit, 34, 211
- Braço de acesso, do disco, 53
- Byte, 31, 211

- Cabeça de leitura/escrita, 53
- Cabo de interconexão, 89, 211
- Cabo de interconexão S 100, 89
- Caixa automática, 76, 211
- Campo de dado codificado, do disco, 54
- Caneta luminosa, 68, 211
- Capacidade de memória, por categoria de computador, 17
 - de computadores, 31
- Caractere, de dados, 33, 211
 - em cartão perfurado, 44
- Cartão perfurado, 44
- Categorias de computador, 17
- CBASIC, 89

- Célula fotoelétrica e máquina perfuradora de teclado, 44
- Chip, 35-36, 211
- Choques de cabeça, disco, 87, 211
- Ciclo de vida, sistemas, 185-186
- Cilindro, de disco, 53
- Circuito de desconexão periódica, 34
- Circuito de computador, 34
- Circuito, 123, 211
 - em BASIC, 144-147
- Circuito integrado, 35-36, 211
- Classificação de bolha, em BASIC, 175-176, 211
- Codificação, programa, 103-104
- Código ampliado de caracteres decimais codificado em binário para o intercâmbio da informação (EBCDIC), 35, 211
- Código de autorização, em programação, 134
- Código de Hollerith, 44-47, 212
- Código Padrão Americano para Intercâmbio de Informações, (ASCII), 35, 210
- Coluna, de cartão perfurado, 44
- Comando de edição, em CP/M, 92
- Compilador, 104-105, 212
- Comprimento de palavra, de mini e microcomputador, 89
- Computador Apple, 89-90
- Computador Atari, 89-90
- Computador IBM de uso pessoal, 89-90
- Computador Commodore PET, 89-90
- Computador TRS 80, 89-90
- Computadores na área de ciência, 201-203
- Computadores na área de ciências humanas, 203-205
- Computadores na área da saúde, 200-201
- Computadores na sociedade, 198-208
- Comunicação, entre o usuário e o pessoal de computação, 14
- Conceito de programa armazenado, 16, 212
- Constantes alfanuméricas, em BASIC, 148-149
- Constantes numéricas, em BASIC, 138
- Consumidor, e microcomputadores, 92-94
- Controle de trabalho, e programação, 134
- Correio eletrônico, 77-78, 212
 - e microcomputadores, 84
- CP/M, sistema operacional, 92, 212
- CPU (Código de Produto Universal), 74-75, 212
- Crime, computador, 206-207

Crime por computador, 206-207
 Custo, por categoria de computador, 17
 de computadores, 14
 de programação, 15

Dados concatenados, 174
 Declaração condicional, em BASIC, 142-147
 Declaração DATA, em BASIC, 154-155
 Declaração DIM, em BASIC, 161
 Declaração READ, em BASIC, 154-155
 Declarações FOR ... NEXT, em BASIC, 156-159
 Depurar, programa, 106, 212
 Desvio, em programa, 121, 212
 Desvio incondicional, em BASIC, 138
 Diagnósticos, em programa, 105
 Dicionário de dados, 192, 212
 Digital Equipment Corporation (DEC), 91
 Digital Research Company, 92
 Dígitos binários, 34
 Disco de Winchester, 87, 213
 Disco flexível, 55, 84-86, 213
 Disco flexível de densidade dupla, 85, 213
 Disco flexível de dupla face, 85, 213
 Disco magnético, 49-57, 213
 Disco rígido, 87
 Dispositivo de avanço de fricção, 88, 213
 Dispositivo de avanço de tração, 88, 213
 Dispositivo de entrada de dados, 42-44
 Dispositivo de Reconhecimento de caracteres em Tinta Magnética (RCTM), 62-63, 213
 Dispositivo de residência do sistema, 111, 213
 Dispositivo Óptico de Reconhecimento de Caracteres (ROC), 63, 213
 Divisão, em BASIC, 138
 Documentação de programa, 106
 Documento de retorno, 44, 48, 213

EBCDIC (Código ampliado de caracteres decimais codificado em binário para o intercâmbio da informação), 35, 213
 Edição, definição, 163, 213
 em BASIC, 161-167
 Electric Pencil, 89
 END, em BASIC, 137
 Entrada de dados à distância, 67, 213
 Entrada, 12, 213
 dispositivo, 16, 27-32, 213
 Erro de sintaxe, programação, 104-105, 213
 Erros, entrada, 14
 programação, 14
 E/S (Entrada-Saída), 42-65, 119

Fase de execução, da programação, 105
 Fase de tradução, da programação, 104-106
 (FEE) Fornecedor de Equipamento Específico, 189-214
 Fita magnética, 49-57, 214
 Fluxograma, programa, 118-123
 Fonte de tipo, 88

Formulários pré-impressos, 55, 61, 214
 Fornecedor de Equipamento Específico (FEE), 189, 214
 Fornecedor externo de software, 110-111
 FORTH, 89
 FORTRAN, 107-108, 109, 214
 Funções de biblioteca, em BASIC, 167-172, 214
 Função de série, em BASIC, 174-175, 214
 Função INT, em BASIC, 167

Gabarito de programação de fluxograma, 100-101, 118, 214
 Gerador de número aleatório, em BASIC, 169-172
 Gerente de operações, 19
 Gerente de programação, 18
 Gerente de sistemas, 18
 GOSUB, em BASIC, 172-174
 GO TO em BASIC, 137
 Gráficos, e TRC, 68-69

Hardware, 16, 25-28, 214
 Hewlett-Packard, 91

IAC (Instrução com Assistência do Computador), 198-200, 215
 IDC (Instrução Dirigida por Computador), 200, 215
 IF... THEN, em BASIC, 142-147
 Implementação do programa, 106
 Impressora, 55-63
 e mini e microcomputadores, 87-88
 Impressora de caractere, 215
 Impressora de impacto, 59, 215
 Impressora de letra de qualidade, 59, 87, 215
 Impressora de matriz de ponto, 61, 215
 Impressora em série, 59, 215
 Impressora por linha, 59, 215
 Impressora por página, 59, 215
 INPUT (entrada), em BASIC, 137
 Intel 8088, chip, 89
 Intérprete, 105, 215
 Inserção de linha em programa BASIC, 135
 Instalação centralizada de computador, 19-21, 215
 Instalação descentralizada de processamento de dados, 19-21, 215
 Instalação distribuída de processamento de dados, 19-21, 215
 Instrução com Assistência do Computador (IAC), 198-200, 215
 Instrução Dirigida por Computador (IDC), 200, 215
 Inteligência artificial, 202-203
 Interface, com terminal, 72, 215

Lanchonetes, e sistema de ponto de venda, 74
 LCT (Linguagem de Controle de Trabalho), 112, 216

- Leitora de cartão, 44-45
- Leitora de código de barra, 70, 215
- LEFT, em BASIC, 175
- LEN, em BASIC, 175
- Linguagem de Controle de Trabalho (LCT), 112, 216
- Linguagem de solicitação de dados, 192, 216
- Linguagens, programação, 106-109
- Linhas de comunicação, 72, 216
- Linhas de comunicação de banda larga, 72
- Linhas de teletipo, 72
- Linhas em branco, impressão em BASIC, 149
- Listagem (LIST) do programa, 136
- Loja, e sistema de ponto de venda, 74
- LOAD, programa, 136

- MAA (Memória de Acesso Aleatório), 38, 216
- Magic Wand, 89
- Mail Merge, 89
- Máquina perfuradora de teclado, 44, 47, 216
- Martin, James, 207
- Matriz, em BASIC, 160-161, 216
- Megabyte, 31, 216
- Meios, de entrada-saída, 42-65
- Memória auxiliar, 28, 32
- Memória auxiliar ou secundária, 28, 32, 216
- Memória de Acesso Aleatório, 38, 216
- Memória de bolha magnética, 36, 216
- Memória de Leitura Somente (MLS), 114, 216
- Memória de Leitura Somente Programável (MLSP), 114, 216
- Memória do computador, 35-38
- Memória primária, 31, 216
- Memória principal, 31, 216
- Memória, 31, 216
 - tipos, 35-38
- Memória virtual, 113, 216
- Microcomputador, 17-18, 82-96, 216
- Microcomputador de dezesseis bits, 89
- Microcomputador de oito bits, 89
- Minicomputador, 17-21, 82-96, 216
- Minicomputador autônomo, 82
- Minicomputador para uso geral, 82
- MLS (Memória de Leitura Somente), 38, 114-115, 216
- MLSPA (Memória de Leitura Somente Programável e Apagável), 114, 217
- Modulador/demodulador, 73, 217
- Montador, 105-106, 217
- MOS 6502, chip, 89
- MP/M, sistema operacional, 92, 217
- Multiplicação, em BASIC, 138
- Multiprocessamento, 113, 217
- Multiprogramação, 112, 217
 - e comunicação de dados, 73
 - e mini e microcomputadores, 88
- Música, computadores e, 204
- MV (Memória Virtual), 113, 216

- Nanossegundo, 13, 217
- NASA, computadores e, 201-202
- Nome de variável, 149
- Núcleo magnético, 35-36, 216
- Número de linha, em BASIC, 137

- Opção TAB (tabulação), em BASIC, 162-163
- Opções de impressão, em BASIC, 148-149
- Operação bancária pré-autorizada, 76
- Operações de computador, 12
- Operador de computador, 18, 217
- Operador de entrada de dados, 18, 217
- Organização, da instalação de computador, 19-20
- Organização direta de arquivo, 54, 217
- Organização relativa de arquivo, 54, 217
- Organização seqüencial indexada do arquivo, 52-53, 217

- Pacotes de aplicações, 111
- Papel em várias vias, 58
- Papel normal padrão, 55
- Papel sem carbono, 58
- Parênteses, em BASIC, 141-142
- Pascal, 108-109, 218
- PDD (instalação distribuída de processamento de dados — processamento de dados distribuído), 19-22
- PED (Processamento Eletrônico de Dados), 12, 218
- Pergunta-resposta, e comunicação de dados, 67, 218
- Pessoal de computação, 18-19
- Pessoal, do centro de computação, 18-19
- PL/1, 108-109, 218
- Placa de silicone, 35
- Precisão, de computadores, 13
- PRINT, em BASIC, 137
- PRINT...USING, 163-167
- Procedimentos de início de comunicação, 133-134, 215
- Processador terminal, 82, 218
- Processamento de fita, 49-57
- Processamento de informação, 12, 218
 - requisitos, 15-16
- Processamento de palavra, 76-77, 220
- Processamento Eletrônico de Dados (PED), 12, 218
- Processamento em linha, 33, 218
 - e disco, 49
- Processamento fora de linha, 33, 218
- Processamento por disco, 49-57
- Processamento por lote, 32-33, 49, 218
- Processamento seqüencial, de fita, 50, 218
- Programa de controle, 31
- Programa fonte, 104-105, 218
- Programação BASIC, 106-107, 109, 133-181, 210
- Programação simbólica, 103-104
- Programador, 18, 218

- Programa-objeto, 104-105, 218
Pseudocódigo, 100, 127, 219
- Reconhecimento de formas, 203, 204, 219
Reconhecimento Ótico de Caractere, (ROC), 63, 219
Rede e processamento de dados distribuído, 19-20
 e minicomputadores, 82
REM, em BASIC, 151
Retrocesso, em terminal, 135
RETURN, em BASIC, 172-174
Rótulo, como controle, 51, 219
Rótulos programados, para controle, 52
RPG, 108-110, 219
- Saída de Hard-copy, 58, 69, 220
Saída de soft-copy, 68, 220
Saúde, computadores e, 200-201
Segurança, e comunicação de dados, 73
Serviços de tempo compartilhado e minicomputadores, 93-94
SGBD (Sistema de Gerência de Banco de Dados), 195, 219
SIG (Sistema de Informação de Gerência), 191, 219
Símbolo e fluxograma, 119
Sinais de dólar, impressão de, em BASIC, 165
Sinal de dólar flutuante, em BASIC, 165
Sistema completo, 184-196
Sistema de computador, 15-16, 27-31, 219
Sistema de computador de tamanho médio, 17
Sistema de computador pequeno, 17
Sistema de computador grande, 17
Sistema de Gerência de Banco de Dados (SGBD), 191, 219
Sistema de informação, 16, 184-196, 219
Sistema de Informação de Gerência (SIG), 191-193, 219
Sistema de informação hospitalar, 200-201, 220
Sistema de Suporte de Decisão (SSD), 194, 220
Sistema de Transferência eletrônica de fundos (TEF), 75, 221
Sistema exclusivo, de minicomputador, 82
Sistema numérico binário, 33-35
Sistema operacional, 110-115
 para mini e microcomputadores, 92
 software e, 99, 220
Software de aplicação, 99-106, 220
Software de apoio, por categoria de computador, 17
- Software de sistemas, 110
Software doméstico, 110
Spell Star, 89
SPV (Sistema de Ponto de Venda), 74-75, 220
SSD (Sistema de Suporte de Decisão), 194, 220
Subíndice, em BASIC, 159-161, 220
Subtração, em BASIC, 138
Supercomputador, 17
Supermercado, e sistema de ponto de venda, 74-75
Supervisor, 31, 99, 111-112, 220
- Taxa de Bauds, 72, 220
Tecla de controle, e programação, 134
TEF (Sistema de Transferência Eletrônica de Fundos), 75, 221
Terminais conectados por fio, 72
Terminais manuais, 221
Terminal, 15, 42-44, 67-71, 221
Terminal de máquina de escrever, 42-44, 69, 221
Terminal inteligente, 70, 221
Tubo de Raios Catódicos (TRC), 43-44, 68-69, 221
 e mini e microcomputadores, 84
- UCP (Unidade Central de Processamento), 16, 27, 28, 31, 221
Unidade Aritmética e Lógica, 31, 221
Unidade Central de Processamento, 16, 27, 28, 31, 221
Unidade de controle, 31, 221
Unidade de fita cassete, 50, 84, 211
Unidade de resposta audível, 69-70, 221
Unidade principal, 27, 221
Usuário, 14-15
- Variáveis numéricas, em BASIC, 138
Velocidade, de computadores, 13
Velocidade de processamento, por categoria de computador, 17
Velocidade de transferência, de meios de arquivo usuais, 57
Versões de BASIC, 176
Vírgulas, impressão de, em BASIC, 164
- Word Star, 89
Xerox, 91
Zilog Z80 chip, 89

Conceitos de Processamento de Dados com **BASIC**

- Um guia para o futuro usuário
- Um livro básico para cursos de introdução ao processamento de informações
- Um complemento para qualquer curso que envolva o uso de computadores
- Uma visão geral dos fundamentos da computação para todos os interessados no assunto

Este é, pois, um texto versátil.

É, também, um texto atual. Apresentando objetivamente os computadores, suas vantagens e limitações, faz uma análise pormenorizada das principais questões nos campos do processamento de informações e da computação. E é, ainda, um texto simples e acessível, fartamente ilustrado com fotografias, diagramas e gráficos, e que inclui, ao final dos capítulos, um breve resumo da matéria e um pequeno questionário de avaliação. De especial interesse para o leitor são o glossário de termos técnicos e o Capítulo 8, inteiramente dedicado à Linguagem Basic, ao final do qual o leitor estará apto a codificar programas menos complexos.

ISBN 85-7001-202-0

(Edição original: ISBN 0-471-87617-8 John Wiley & Sons, Inc.,
New York.)