

NÉLSON CASARI

dBASE II PLUS

PARA



INTERATIVO



2ª Edição

atlas

dBASE II PLUS
PARA MSX
INTERATIVO



EDITORA ATLAS S.A.

Rua Conselheiro Nébias, 1384 (Campos Elísios)
Caixa Postal 7186 – Tel.: (011) 221-9144 (PABX)
01203 São Paulo (SP)

NÉLSON CASARI

dBASE II PLUS PARA MSX

INTERATIVO

CURSO PRÁTICO

2ª EDIÇÃO

SÃO PAULO
EDITORIA ATLAS S.A. — 1990

© 1988 by EDITORA ATLAS S.A.
Rua Conselheiro Nébias, 1384 (Campos Elísios)
Caixa Postal 7186 – Tel.: (011) 221-9144 (PABX)
01203 São Paulo (SP)

1. ed. 1988; 2. ed. 1989; 2.^a tiragem — 1990

ISBN 85-224-0470-4

Impresso no Brasil/Printed in Brazil

Depósito legal na Biblioteca Nacional conforme Decreto nº 1.825, de 20 de dezembro de 1907.

TODOS OS DIREITOS RESERVADOS – É proibida a reprodução total ou parcial, de qualquer forma ou por qualquer meio, salvo com autorização, por escrito, do Editor.

Capa
Paulo Ferreira Leite

**Dados de Catalogação na Publicação (CIP) Internacional
(Câmara Brasileira do Livro, SP, Brasil)**

Casari, Néelson, 1931-
dBASE II PLUS para MSX : interativo : curso prático / Néelson Casari. - -
2. ed. - - São Paulo : Atlas, 1989.

Bibliografia.
ISBN 85-224-0470-4

1. dBASE II PLUS (Programa de computador) 2. MSX (Computadores)
– Programação I. Título.

89-0216

CDD-001.6425
-001.642

Índices para catálogo sistemático:

1. dBASE II PLUS : Computadores : Programas : Processamento de dados
001.6425
2. MSX : Computadores : Programação : Processamento de dados 001.642

SUMÁRIO

INTRODUÇÃO, pág. 15

- 1 - BANCO DE DADOS - O QUE É, 17
- 2 - SGBD - SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS, 21
- 3 - ESTRUTURA DE UM ARQUIVO DE DADOS, 22
 - CAMPOS, 22
 - REGISTRO, 23
 - CHAVE PRIMÁRIA DO REGISTRO, 23
 - CHAVES SECUNDARIAS DO REGISTRO, 24
 - CAMPO CHAVE, 24
- 4 - dBASE II PLUS - GERENCIADOR RELACIONAL DE DADOS, 25
 - 4.1 - SGBD EM LINGUAGENS PADRONIZADAS, 25
 - 4.2 - SGBD EM LINGUAGENS NÃO PADRONIZADAS, 26
 - 4.3 - SGBD RELACIONAL, 26
- 5 - SGBDs EM LINGUAGENS PADRONIZADAS OU DBASE?, 28
 - 5.1 - PORQUE ESCOLHER UM SGBD COMO dBASE, 29
 - 5.2 - dBASE, SGBD UNIVERSAL, 30
 - 5.3 - MODO INTERATIVO, 31

5.4 - MODO PROGRAMÁVEL, 32

6 - COMEÇANDO A USAR O dBASE II PLUS, 33

6.1 - ABRINDO O "PACOTE" dBASE II PLUS, 34

6.2 - COMO PREPARAR DISQUETES DE TRABALHO DO dBASE II PLUS, 35

6.3 - PORQUE COPIAR O DOS (MSX DOS, HBDOS, SOLXDOS)

NO DISQUETE DE TRABALHO DO dBASE II PLUS, 38

6.4 - ATIVANDO O dBASE NO MICROCOMPUTADOR, 39

6.5 - O PONTO DO dBASE, 41

6.6 - MODOS DIRETO E INDIRETO NO dBASE, 41

7 - INFORMAÇÕES ÚTEIS, 42

7.1 - CONVENÇÃO ADOTADA NA APRESENTAÇÃO DOS COMANDOS, 43

7.2 - COMO DIGITAR OS COMANDOS DO dBASE, 44

7.3 - COMANDOS DIGITADOS INCORRETAMENTE, 45

7.4 - ACENTUAÇÃO GRÁFICA NO dBASE II PLUS, 47

7.5 - ESPECIFICAÇÕES DO dBASE II PLUS, 48

7.6 - TECLAS DE CONTROLE DOS COMANDOS

APPEND E EDIT (INDICADOR VISUAL), 49

7.7 - TECLAS DE CONTROLE DO COMANDO BROWSE (INDICADOR VISUAL), 50

7.8 - TECLAS DE CONTROLE DO COMANDO MODIFY (INDICADOR VISUAL), 51

7.9 - OPERADORES VÁLIDOS NO dBASE II PLUS, 52

7.10 - HELP - COMANDO DE AUXÍLIO DO dBASE II PLUS, 54

7.11 - MENSAGENS DE ERRO DO dBASE II PLUS, 56

7.12 - LOCALIZAÇÃO DOS COMANDOS E FUNÇÕES

DO dBASE II PLUS ESTUDADOS NESTE LIVRO, 61

8 - CRIANDO O PRIMEIRO ARQUIVO DE DADOS, 62

- Comando CREATE, 63

- EM CASO DE ERRO NA DEFINIÇÃO DA ESTRUTURA, 65

- Comandos USE, DISPLAY e LIST, 65

- DISPLAY STRUCTURE, 66

- DISPLAY FILES, 67

- Comando QUIT, 68

9 - INSERINDO DADOS NO ARQUIVO, 69

- Comando APPEND, 69

- Comando SET COLON ON/OFF, 70
 - Comando INSERT, 73
 - Comando SET CONFIRM ON/OFF, 73
 - Comando SET CARRY ON/OFF, 74
 - Comando DISPLAY STATUS ou LIST STATUS, 75
 - Comando SET, 75
- 9.1 - ACENTUAÇÃO GRAFICA DE DADOS NO dBASE, 76

10 - PESQUISANDO INFORMAÇÕES NO ARQUIVO, 78

- DIFERENÇA BÁSICA ENTRE DISPLAY E LIST, 78
 - Comando ERASE, 79
- 10.1 - USANDO A IMPRESSORA ATRAVÉS DO COMANDO DISPLAY, 80
- Comandos SET PRINT ON/OFF e CONTROL/P, 81
 - PONTEIRO DE REGISTROS, 83
 - Comando GO ou GOTO, 83
 - Função #, 84
 - ? - Comando de IMPRESSÃO, 84
 - Comando SKIP, 85
- 10.2 - PESQUISAS DIVERSAS EM CAMPOS "C", 86
- Função !, 87
- 10.3 - BUSCAS DE DADOS ACENTUADOS GRAFICAMENTE, 87
- 10.4 - PESQUISAS DE DADOS COM VERIFICAÇÕES EXATAS, 88
- Comando SET EXACT ON/OFF, 89
- 10.5 - PESQUISAS DIVERSAS EM CAMPOS "N", 89
- Função STR, 90
 - Função \$, 90
 - Comandos LOCATE e CONTINUE, 91
 - Comando COUNT, 93
 - Comando SUM, 94
- 10.6 - PESQUISAS DE DADOS ORDENADOS, 95
- 10.6.1 - INDEXAÇÃO DE ARQUIVO, 96
- 10.6.2 - CRIAÇÃO DE ARQUIVO DE ÍNDICE, 97
- Comando INDEX ON, 97
 - Comando SET INDEX TO, 99
 - Comando INDEX, 100
 - Comando FIND, 101
- 10.6.3 - INDEXAÇÃO POR VÁRIOS CAMPOS OU CHAVES, 102
- 10.6.4 - REINDEXAÇÃO DE ARQUIVOS DE ÍNDICE, 103
- Comando REINDEX, 103

10.6.5 - USANDO ALTERNATIVAMENTE DIVERSOS ARQUIVOS DE ÍNDICE, 104

11 - ALTERANDO DADOS DO ARQUIVO, 110

- Comando EDIT, 110
- TECLAS DE CONTROLE DO COMANDO EDIT, 111
- Comando SET SCREEN ON/OFF, 114
- Comando BROWSE, 118
- TECLAS DE CONTROLE DO COMANDO BROWSE, 119
- Comando CHANGE, 123
- Comando REPLACE, 125

11.1 - CONVERTENDO LETRAS MINÚSCULAS
DE DADOS PARA MAIÚSCULAS, 126

12 - DELETANDO REGISTROS DO ARQUIVO, 128

- Comando DELETE, 128
- Função *, 129
- Comando PACK, 129
- Comando RECALL, 129
- Comando SET DELETED ON/OFF, 129

13 - EMITINDO RELATÓRIOS, 135

- Comando REPORT, 135
- 13.1 - EMISSÃO DE RELATÓRIOS ATRAVÉS DO VÍDEO, 136
 - Comando SET HEADING TO, 139
- 13.2 - EMISSÃO DE RELATÓRIOS ATRAVÉS DE IMPRESSORA, 141
 - Comando EJECT, 142
- 13.3 - EMISSÃO DE RELATÓRIOS PARCIAIS, 142
- 13.4 - INSTRUÇÕES PARA A IMPRESSORA, 144
 - 13.4.1 - ALTERAÇÃO DA MARGEM ESQUERDA DE IMPRESSÃO, 146
 - Comando SET MARGIN TO, 146

**14 - UM NOVO ARQUIVO
PARA NOVAS EXPERIÊNCIAS, 147**

- Comando SET DEFAULT TO, 150

15 - ARQUIVOS DE SUBTOTAIIS, 154

- Comando TOTAL ON, 154

16 - TRABALHANDO COM DOIS

ARQUIVOS A UM SÓ TEMPO, 158

- Comando SELECT PRIMARY/SECONDARY, 158

16.1 - EXIBINDO DADOS DE DOIS ARQUIVOS AO MESMO TEMPO, 160

- Prefixos P. e S., 160

16.2 - DESLOCANDO OS PONTEIROS DE REGISTROS

SIMULTANEAMENTE EM DOIS ARQUIVOS EM USO, 162

- Comando SET LINKAGE ON/OFF, 163

16.3 - FECHAMENTO DE DOIS ARQUIVOS DBF EM USO SIMULTANEO, 166

17 - JUNÇÃO DE DADOS DE DOIS ARQUIVOS, 168

- Comando JOIN, 168

18 - ARQUIVO DE MOVIMENTO, 172

- Comando UPDATE, 173

19 - ARQUIVO DE COMANDOS, 181

19.1 - CRIAÇÃO DE ARQUIVO DE COMANDO, 182

- Comando MODIFY, 182

19.2 - EXECUÇÃO DE ARQUIVO DE COMANDOS, 184

- Comando DO, 184

19.3 - ALTERAÇÃO OU CORREÇÃO DE ARQUIVO DE COMANDOS, 184

- TECLAS DE CONTROLE DO COMANDO MODIFY, 185

20 - ARQUIVO ALTERNATIVO

OU ARQUIVO DE FATOS, 187

- Comando SET ALTERNATE TO, 188

- Comando SET ALTERNATE ON/OFF, 188

21 - LIDANDO COM ARQUIVOS, 189

21.1 - COPIAÇÃO DE ARQUIVO E DE ESTRUTURA DE ARQUIVO, 189

- Comando COPY, 189

- CÓPIA DE ESTRUTURA DE ARQUIVO OU DE PARTE DELA, 191

21.2 - MUDANÇA DE NOMES DE ARQUIVOS, 192

- Comando RENAME, 192
 - 21.3 - ELIMINAÇÃO DE ARQUIVOS DO DISQUETE, 193
 - Comando DELETE FILE, 193
 - 21.4 - LISTAGEM DE ARQUIVOS DO DISQUETE, 193
 - Comandos DISPLAY e LIST, 193
 - 21.5 - MODIFICAÇÃO DE ESTRUTURA DE ARQUIVO, 194
 - Comando MODIFY STRUCTURE, 195
 - Comando APPEND FROM, 198
 - 21.6 - CONVERSÃO DE ARQUIVOS dBASE
 - PARA ARQUIVOS NÃO dBASE E VICE-VERSA, 202
 - 21.6.1 - CONVERSÃO DE ARQUIVOS dBASE EM ARQUIVOS NÃO dBASE, 203
 - 21.6.2 - CONVERSÃO DE OUTROS ARQUIVOS EM ARQUIVOS dBASE, 204
- 22 - CONSTANTES E VARIÁVEIS DE MEMÓRIA, 207**
- 22.1 - CRIAÇÃO E INICIALIZAÇÃO DE VARIÁVEIS, 209
 - 22.1.1 - CRIAÇÃO E ATRIBUIÇÃO DE VARIÁVEIS EM MODO DIRETO, 209
 - Comando STORE, 209
 - 22.1.2 - CRIAÇÃO E ATRIBUIÇÃO DE VARIÁVEIS EM MODO INDIRETO, 210
 - Comando INPUT, 210
 - Comando ACCEPT, 211
 - 22.1.3 - ATRIBUIÇÃO A VARIÁVEIS COMO RESULTADO
 - DA EXECUÇÃO DOS COMANDOS COUNT E SUM, 212
 - 22.1.4 - ALTERAÇÃO DO CONTEÚDO DE VARIÁVEIS, 212
 - 22.1.5 - ELIMINAÇÃO DE VARIÁVEIS DE MEMÓRIA, 212
 - Comando RELEASE, 212
 - Comando CLEAR, 213
 - 22.1.6 - GRAVAÇÃO DE VARIÁVEIS DE MEMÓRIA EM DISQUETE, 214
 - Comando SAVE, 214
 - 22.1.7 - RECUPERAÇÃO DE VARIÁVEIS
 - DE MEMÓRIA DE ARQUIVO EM DISQUETE, 214
 - Comando RESTORE, 214
 - 22.1.8 - LISTAGEM DE VARIÁVEIS DE MEMÓRIA, 215
 - Comando LIST MEMORY ou DISPLAY MEMORY, 215
- 23 - OUTRAS FUNÇÕES DO dBASE II PLUS, 216**
- 23.1 - Função @, 216
 - 23.2 - Função &, 217
 - 23.3 - Função CHR, 218

- 23.4 - Função DATE, 219
- 23.5 - Função EOF, 219
- 23.6 - Função FILE, 220
- 23.7 - Função INT, 220
- 23.8 - Função LEN, 221
- 23.9 - Função RANK, 222
- 23.10 - Função TRIM, 222
- 23.11 - Função TYPE, 223
- 23.12 - Função VAL, 224

24 - OUTROS COMANDOS DO GRUPO "SET", 225

- SET BELL ON/OFF, 225
- SET CONSOLE ON/OFF, 225
- SET ESCAPE ON/OFF, 226
- SET INTENSITY ON/OFF, 226

25 - "PROGRAMANDO" COM AUXÍLIO DOS UTILITÁRIOS DO DBASE II PLUS, 227

- DGEN, 228
 - GERAARQ, 229
 - GERAREL, 232
 - GERAETIQ, 233
 - GERAMENU, 234
- 25.1 - COMANDOS PRÓPRIOS DO MODO PROGRAMAVEL, 236

BIBLIOGRAFIA, 237

INTRODUÇÃO

Desde quando foi lançado, o programa dBASE é um grande sucesso.

Tudo faz crer que é o sistema de gerenciamento de banco de dados mais vendido e mais usado em todo o mundo.

Existem atualmente quatro versões do dBASE: dBASE II, dBASE II PLUS, dBASE III e dBASE III PLUS. A versão II serve aos microcomputadores de oito e de dezesseis "bits", enquanto que a versão III só pode ser utilizada em microcomputadores de dezesseis "bits".

No Brasil, só recentemente foi lançada a versão II PLUS para microcomputadores do padrão internacional MSX, os quais são produzidos aqui há cerca de dois anos.

Graças a tal lançamento, esses microcomputadores foram grandemente valorizados e passaram definitivamente a fazer parte do elenco das máquinas profissionais de grande utilidade e versatilidade, já que a modalidade de serviço prestada pelo dBASE é a mais utilizada na atualidade, em todos os setores da atividade humana, mas principalmente na área administrativa empresarial e pública.

Antes do lançamento do dBASE II PLUS para MSX, os microcomputadores deste padrão só podiam operar com bancos de dados gerados por linguagem padronizada como o BASIC, por exemplo, ou linguagem de máquina, "assembly", o que restringia enormemente o seu uso, já que dependiam de programas específicos, de elaboração geralmente demorada e custosa e que não oferecem a condição de universalidade dos arquivos de bancos de dados gerados pelo sistema dBASE.

Mas o manual que acompanha o dBASE II PLUS para MSX é um manual apenas de referência. Não ensina o operador ou usuário de MSX inexperiente a utilizá-lo. Em consequência, por carência de conhecimento, de tempo ou de oportunidade para frequentar cursos especializados, grande parte de seus adquirentes utiliza-o apenas timidamente, não usufruindo toda a sua potencialidade, quando não deixa de utilizá-lo de vez.

Assim, objetivando oferecer humildemente algum subsídio para complementar tal quadro, elaboramos este livro, que pretende servir como manual auxiliar em cursos de dBASE II PLUS.

O livro foi compilado nos moldes de um CURSO PRÁTICO, através do qual a matéria estudada é ministrada de maneira gradativa e objetiva, por meio de experiências detalhadas que simulam o uso real de bancos de dados e que induzem o estudante a efetuar-las no microcomputador com agrado e facilidade, propiciando-lhe a prática necessária para domínio do sistema, sem necessidade dos convencionais exercícios teóricos com lápis e papel, os quais são muitas vezes totalmente improfícuos.

Em locais onde faltem cursos sobre o dBASE II PLUS, temos certeza de que o presente livro será muito útil àqueles que pretendem se especializar no assunto. Para tanto, empregamos toda nossa experiência e modesta capacidade na sua elaboração.

Mas, não obstante todo nosso esforço e empenho, temos também certeza de que nosso modesto livro poderá ainda ser melhorado e escoimado de falhas e lacunas, se aqueles que o utilizarem com proveito nos fizerem suas sugestões e críticas contrutivas. A essas pessoas agradecemos antecipadamente, solicitando-lhes o obséquio de nos escreverem aos cuidados da EDITORA.

São Paulo, janeiro de 1988.

O Autor

1 - BANCO DE DADOS

- O QUE É.

Banco de dados é a denominação que se dá a todo CONJUNTO ORGANIZADO DE DADOS que constitui o núcleo - ou a base - de um SISTEMA DE INFORMAÇÕES.

Pode-se dizer que um banco de dados é um acervo organizado de informações sobre diversos assuntos ou matérias.

Exemplos de bancos de dados:

- Fichário de empregados de uma empresa.
- Arquivo de contas-correntes de clientes de um banco comercial.
- Lista telefônica.

Em cada um desses bancos de dados, as informações são organizadas de maneira a corresponder com sua finalidade precípua.

Assim, um fichário de empregados deve dispor de informações organizadas em moldes que permitam à empresa saber rapidamente e em qualquer momento quantos funcionários mantém em seus quadros, em quais departamentos atuam, seus cargos, salários, idades, datas de admissões e outras informações de seu interesse.

Os dados das contas-correntes de clientes de um banco comercial devem refletir a qualquer momento o saldo atual e médio de cada conta, seu número e natureza, limite para eventual saque a descoberto, nome e

endereço de seu titular etc.

Listas de telefones são bancos de dados públicos onde são obtidas informações específicas sobre localização de pessoas (endereço e número de telefone) - lista de assinantes - ou sobre suprimentos e serviços gerais - lista amarela.

Bancos de dados são utilizados também no lar, reunindo e administrando informações úteis sobre a biblioteca e a discoteca da casa, receitas culinárias, guarda-roupa etc.

Antes do advento do computador, os bancos de dados eram "físicos" ou "concretos", isto é, todas as informações que reuniam eram caligrafadas ou impressas em "fichas" de papel ou de cartolina e armazenadas em ordem alfabética, numérica ou cronológica em pastas, que eram guardadas em gavetões de móveis de arquivo, diretamente nesses gavetões ou em estantes, armários etc.

Um tipo de banco de dados muito corrente no passado era o que fazia uso de enormes livros de capas grossas forradas de tecido cinza ou preto, nos quais os dados eram caligrafados sequencialmente, obedecendo à ordem de numeração das páginas. Exemplos marcantes deles eram os famosos livros de tabeliães, nos quais eram registrados nascimentos, casamentos, transferências de propriedades etc., até hoje conservados.

A consulta para obtenção de informações de tais bancos de dados era geralmente morosa e apresentava muitas falhas quando o sistema era constituído por fichas e pastas. Bastava uma ficha ou pasta ter sido colocada fora de ordem para causar sérios transtornos.

Assim, às deficiências e dificuldades próprias existentes para a criação e organização de bancos de dados, sobrepunha-se o problema de sua manutenção em condições de funcionamento eficiente e perfeito.

Com o advento do computador, a criação, organização e manutenção de bancos de dados passaram a ser feitas eletronicamente, de modo quase que geral, e sofreram grande avanço e aperfeiçoamento, passando a desfrutar de nova conceituação.

Deixaram de ser sistemas de confiabilidade e uso restritos e passaram a ser praticamente imprescindíveis em qualquer sistema adminis-

trativo de grande, médio ou pequeno porte.

Além de sua grande rapidez e confiabilidade, os arquivos eletrônicos de dados - também arquivos magnéticos - oferecem grande economia de espaço e de pessoal. Certos arquivos convencionais que antes ocupavam uma área de 50 metros quadrados, três ou quatro funcionários, diversos móveis para arquivamento de fichas ou de pastas e vários armários e escrivaninhas, podem agora funcionar perfeitamente em área correspondente a um ou dois décimos daquela, exigindo apenas uma unidade de computação eletrônica, um único funcionário e pouquíssimos móveis.

Os dados de bancos eletrônicos são armazenados em discos ou fitas magnéticas, através de processo de gravação e reprodução semelhante ao utilizado em áudio. Em vez de armazenar e reproduzir som, os meios magnéticos utilizados em computação - discos e fitas - armazenam e reproduzem dados (informações). A interação destes com os meios magnéticos é obtida através de modulação e demodulação codificadas de pulsos eletrônicos convertidos em pulsos elétricos e vice-versa.

Nos primeiros tempos da era da computação, os bancos eletrônicos de dados eram disponíveis apenas nas grandes empresas, que podiam suportar o elevado custo gerado pela instalação de um computador ou minicomputador. Microcomputadores também eram produzidos em pequena escala e representavam custos que não estimulavam sua aquisição.

Em fins da década de 70, quando microcomputadores passaram a ser produzidos em maior quantidade e a custos que possibilitavam vendê-los a preços mais acessíveis, bancos eletrônicos de dados passaram a despertar grande interesse também para as médias e pequenas empresas.

Todavia, sua implantação era muitas vezes inviabilizada, pois dependia de "gênios especialistas na matéria", os quais - com exceções, naturalmente - cobravam honorários tão elevados que acabavam desestimulando a maioria dos interessados.

Foi com o surgimento dos

SGBDs

ou "SISTEMAS PARA GERENCIAMENTO DE BANCOS DE DADOS" que, felizmente, tal situação mudou, propiciando não somente às médias e pequenas em-

presas a oportunidade de terem seus bancos de dados funcionais e eficientes, mas também a profissionais liberais e a quaisquer outros interessados.

Quando um banco de dados reúne informações sobre um único tópico, é mais apropriadamente denominado ARQUIVO DE DADOS.

Na sua acepção mais adequada, BANCO DE DADOS significa um CONJUNTO DE ARQUIVOS DE DADOS sobre entidades ou itens DIVERSOS.

É importante notar, todavia, que as denominações BANCO DE DADOS e SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS, ou SGBDs, têm sido amiúde usadas indevidamente para descrever simples agendas de endereços ou de telefones e seus respectivos inexpressivos programas de gerenciamento, os quais nem sempre são providos dos mínimos recursos e características que lhes permitam merecer tal classificação.

2 - SGBD - SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS

Um SGBD é um conjunto de programas utilitários ("software") que permite criar, organizar e manter um banco eletrônico de dados.

Em outras palavras: um SGBD é um sistema de programação para computadores, cuja utilização permite PROCESSAR DADOS ELETRONICAMENTE, desde sua entrada para armazenamento (arquivamento em discos ou fitas magnéticas) até sua liberação na forma de "fornecimento" de informações através de monitor de vídeo ou de impressora.

A entrada de dados pode ser feita por digitação (via teclado) ou por transferência de outro banco de dados.

Depreende-se do exposto, pois, que "processar dados eletronicamente" significa manipular através de um computador "elementos de informação" definidos como tais (dados), sobre quaisquer assuntos.

O processamento de dados de um banco eletrônico compreende:

- entrada e gravação de dados = ARMAZENAMENTO DE DADOS.
- ordenação, atualização ou exclusão de dados; cálculos e avaliações de dados numéricos = ORGANIZAÇÃO E MANUTENÇÃO DE DADOS.
- edição ou liberação de dados para o monitor de vídeo (visualização das informações) ou para a impressora (impressão das informações), gerando relatórios, etiquetas, listas etc. = RECUPERAÇÃO E UTILIZAÇÃO DE DADOS.

3 - ESTRUTURA DE UM ARQUIVO DE DADOS

Para que o processamento eletrônico de dados de arquivos do tipo dos criados e gerenciados pelo "software" dBASE se efetive com proveito, eficiência e confiabilidade, seu sistema requer que os dados sejam armazenados em conformidade com uma estrutura uniforme definida para o arquivo todo, de forma que todos os conjuntos de dados sejam "guardados" numa mesma ordem ou disposição e em espaços predefinidos e regulares para cada um, isto é, iguais para todos os registros (entidades ou "fichas").

Para tanto, deve ser definida previamente uma estrutura para o arquivo a ser criado e processado.

Para definir a estrutura de um arquivo, deve-se:

- determinar o espaço (em número de colunas) a ser destinado a cada dado e dar um nome a esse espaço;
- caracterizar o seu conteúdo: caracteres alfanuméricos (C), números (N) ou lógico (L).

Os espaços destinados a armazenar dados são denominados:

CAMPOS.

Um conjunto de CAMPOS relacionados, isto é, referentes a um mesmo item ou entidade, é denominado:

REGISTRO.

Pode-se deduzir do exposto que um registro de arquivo eletrônico ou de banco de dados é, pois, equivalente a uma "ficha" ou "pasta de cartolina" de arquivo convencional, em que são mantidos "elementos de informação", ou dados, de um item ou de uma entidade.

Uma ficha de arquivo de controle de estoque convencional, com uma configuração como a exemplificada a seguir:

ITEM: _____	CÓDIGO: _____
NOME DO FORNECEDOR: _____	
PREÇO CUSTO: _____, ____	PREÇO VENDA: _____, ____
QUANTIDADE MÍN.: _____	ESTOQUE: _____
ÚLT. COMPRA: __/__/__	VALOR TOTAL: _____, ____

deverá ter sua estrutura definida num arquivo eletrônico de dados com a seguinte configuração:

ITEM, C, 23
Cód, C, 5
FORN, C, 26
CUSTO, N, 8,2
VENDA, N, 8,2
QTMÍN, C, 5
ESTOQ, N, 5
ULTCOMP, C, 8
VLTOT, N, 10,2

Ao campo que identifica com precisão um registro, distinguindo-o de maneira inequívoca dos demais, dá-se a denominação de

CHAVE PRIMÁRIA DO REGISTRO

Exemplo: ITEM ou Cód no registro configurado acima.

Outros campos do registro, que armazenam dados complementares ou atributivos ao dado contido numa chave primária, são considerados

CHAVES SECUNDÁRIAS DO REGISTRO.

Exemplo: FORN ou CUSTO no registro configurado acima.

Arquivos de dados costumam ser ORDENADOS, INDEXADOS, CONCATENADOS, ATUALIZADOS e TOTALIZADOS através de um ou mais campos. Dá-se o nome de

CAMPO CHAVE

ao campo através do qual um arquivo é processado para uma das finalidades enunciadas.

No dBASE, os elementos de estruturação do arquivo são solicitados automaticamente quando é criado o arquivo com o comando CREATE, como será visto oportunamente.

A resumida introdução ora feita sobre estrutura de arquivo eletrônico de dados visa propiciar noções básicas ao leitor ainda não familiarizado com o assunto, já que a estrutura de um arquivo ou banco de dados deve ser considerada como seu arcabouço, em torno do qual gira todo o sistema.

4 - dBASE II PLUS - GEREN- CIADOR RELACIONAL DE DADOS

4.1 - SGBD EM LINGUAGENS PADRONIZADAS

Os bancos de dados criados e gerenciados por linguagens de computação padronizadas, como BASIC, COBOL etc., apresentam alguns sérios inconvenientes quando se pensa em termos de dinâmica de utilização.

Por exemplo, se houver necessidade de acrescentar um novo campo aos registros de um arquivo já existente, será necessário:

- criar um novo arquivo com uma estrutura capaz de receber os dados do arquivo antigo, bem como os do novo campo criado;
- elaborar um programa para copiar e transferir os dados do velho arquivo para o novo;
- modificar o programa gerenciador em função da nova estrutura do arquivo.

O mesmo acontece quando se pensa em alterar o tamanho de um ou de mais campos já existentes.

Dependendo da extensão do arquivo e de sua estrutura, tal tarefa poderá representar considerável dispêndio de tempo e alto custo.

4.2 - SGBD EM LINGUAGENS NÃO PADRONIZADAS

Os SGBDs simples constituídos por linguagens não padronizadas já não apresentam os inconvenientes acima apontados, pois normalmente encerram, isto é, dispõem de programas de conversão já embutidos, capazes de alterar a estrutura de seus arquivos, permitindo acrescentar novos campos ou aumentar os tamanhos dos já existentes. Em tais casos, os dados dos arquivos em alteração são convertidos automaticamente e não exigem a elaboração de programas extras.

Essa, em realidade, é uma das características mais desejáveis e marcantes de um SGBD: propiciar soluções rápidas às necessidades de alterações surgidas no banco de dados.

4.3 - SGBD RELACIONAL

As necessidades cada vez mais crescentes de aperfeiçoar os SGBDs, a fim de otimizar sua dinâmica de utilização, geraram pesquisas e estudos que culminaram no desenvolvimento de um novo SGBD: o RELACIONAL, do qual é expressão máxima o dBASE nas suas diversas versões atuais, entre as quais destacamos o dBASE II PLUS para a linha de computadores do padrão MSX.

O SGBD RELACIONAL dBASE II PLUS é um sistema constituído por linguagem própria. Além de ser dotado de todas as características existentes nos SGBDs simples em linguagens não padronizadas, dispõe de recursos de grande versatilidade, que permitem:

- converter arquivos gerados no formato SDF (Standard Data Format) de linguagens padronizadas para o formato dBASE e vice-versa, possibilitando o intercâmbio de informações entre arquivos de origens e formatos diversos;
- efetuar a fusão ou união de dados de arquivos diversos, mantendo suas características ou revestindo-os de outras.

Merece também especial destaque no dBASE II PLUS a existência nele de uma linguagem de programação, que coloca à disposição do usuário amplos e versáteis recursos para sua utilização, como será visto oportunamente.

5. SGBDs EM LINGUAGENS

PADRONIZADAS OU dBASE?

Para se criar um arquivo eletrônico de dados usando-se linguagens de computação padronizadas, como BASIC, COBOL etc., é preciso que, antes, seja elaborado seu programa gerenciador.

A elaboração de tal programa deverá considerar previamente os seguintes pontos:

- modalidade do arquivo (sequencial ou de acesso direto, com ou sem indexação);
- quantidade, natureza e extensão dos campos;
- extensão e enquadramento dos registros (256 caracteres no máximo, no MSX), visando aproveitamento otimizado do disquete;
- formatação da tela de vídeo para obtenção dos dados de entrada;
- desenvolvimento de fórmulas matemáticas para cálculos de dados dependentes de valores ou de quantidades;
- definição dos campos que serão utilizados para pesquisas ou buscas e constituição de informações a serem prestadas;
- modalidades de fornecimento de informações (relatórios através de monitor ou de impressora, impressão de etiquetas - 1,2,3 ou 4 por carreira -, formato de exibição e de impressão dos dados a serem fornecidos como informações etc.);
- recursos para manutenção dos dados (atualização, alteração, inserção de novos registros ou cancelamento de já existentes);

- detalhes diversos inerentes a quaisquer programas e a programas de gerenciamento de arquivos.

A lista poderá ainda ser acrescida de diversos outros itens, dependendo do tipo de arquivo desejado, do programa para seu gerenciamento e do que é esperado da utilização do arquivo ou banco de dados.

Pelo exposto, parece-nos fácil deduzir que a criação e a organização de bancos de dados com a utilização de linguagens de computação, como BASIC, COBOL etc., exigem conhecimentos técnicos avançados e especializados por parte do programador.

Seu desenvolvimento e depuração são geralmente demorados, chegando a consumir frequentemente meses de trabalho, dependendo de sua extensão e complexidade.

Contudo, um SGBD nessa modalidade geralmente só tem aplicação no arquivo ou banco de dados para o qual foi elaborado. Com adaptações - quase sempre difíceis - poderá eventualmente ser usado em arquivo com estrutura idêntica e finalidades semelhantes.

Por tais razões, o desenvolvimento de SGBDs através de linguagens de computação padronizadas geralmente representam custos elevados.

5.1 - PORQUE ESCOLHER UM SGBD COMO dBASE

A vista do que foi exposto acima e tendo-se em conta que um bom SGBD deve caracterizar-se por:

- permitir rápida e eficientemente a criação e organização de um arquivo ou banco de dados;
- dispor de recursos rápidos e eficientes para manutenção do arquivo ou banco de dados, para fins de atualização ou exclusão dos mesmos, incluindo execução de cálculos e avaliações de dados numéricos;
- possibilitar a ordenação ou indexação dos registros através de um ou mais campos, com a finalidade de facilitar pesquisas ou

fornecimento de informações e elaboração de relatórios;

- liberar rapidamente dados do arquivo ou banco de dados na forma de informações exibidas no monitor ou através de relatórios impressos, facilitando consultas; ou na forma de etiquetas, para endereçamento de mala-direta;
- oferecer soluções rápidas quando houver necessidade de alterar a estrutura do arquivo ou banco de dados, com a finalidade de modificar extensões de campos ou acrescentar novos;
- permitir a incorporação ou transferência de dados de um arquivo para outro, ainda que os mesmos tenham estruturas diferentes;
- ser operacionalmente simples e inteligível,

chegar-se-á, sem dúvida, à conclusão de que, por oferecer todos os recursos mencionados, além de outros mais, o SGBD dBASE merecerá a preferência de escolha, em especial o dBASE II PLUS para a linha de computadores do padrão MSX.

5.2 - dBASE, SGBD UNIVERSAL

Pode-se dizer que o dBASE é um SGBD UNIVERSAL porque ele é capaz de gerar e gerenciar qualquer tipo de arquivo ou banco de dados, eliminando de modo absoluto a necessidade de elaborar um programa específico para cada arquivo ou banco de dados a ser criado e gerenciado.

Todavia, não obstante essa característica de universalidade total, o dBASE oferece também recursos para elaboração de programas específicos, através de sua linguagem própria de programação, embutida no seu conjunto.

Assim, o usuário do dBASE dispõe de duas opções ou modos para sua utilização:

- 1 - MODO INTERATIVO
- 2 - MODO PROGRAMÁVEL

5.3 - MODO INTERATIVO

No MODO INTERATIVO, o usuário simplesmente carrega o dBASE na memória do microcomputador e, fazendo apenas uso de seus comandos, cria um arquivo e começa a operá-lo em conformidade com os padrões próprios do dBASE para entrada de dados e formatações da tela de vídeo.

Depois de o arquivo estar organizado, isto é, carregado de dados, o usuário poderá imediatamente fazer atualizações, pesquisas, consultas, emissão de relatórios etc., sendo todo o processamento e a liberação de informações e de relatórios feitos nos padrões de formatações também próprios do dBASE. O usuário não tem praticamente outra coisa a fazer senão usar os comandos do dBASE.

Não obstante a expressão "padrões do dBASE" parecer sugerir alguma limitação de recursos nas formatações citadas, tal fato não ocorre.

Os padrões de formatação de tela de vídeo para entrada de dados e liberação de informações são inquestionavelmente eficientes e irreprocháveis, estando já embutidos no sistema a fim de poupar ao usuário o trabalho de programá-los, o que, todavia, poderá ser feito através do MODO PROGRAMAVEL, caso o usuário queira personalizar tais formatações.

Quanto à formatação de relatórios, os padrões são apenas operacionais ou de funcionalidade, dependendo seu conteúdo e estética final de instruções solicitadas pelo dBASE através do vídeo e fornecidas pelo usuário por intermédio do teclado.

Pode-se dizer que no MODO INTERATIVO o usuário simplesmente "diz" ao dBASE - usando os comandos próprios do mesmo - o que deve ser feito e o dBASE "faz", sem dificuldades nem complicações.

Por essa razão, o MODO INTERATIVO é referido como um MODO CONVERSACIONAL, sendo a interação entre ele e o usuário feita através do teclado e do vídeo. O usuário se comunica através do teclado com o dBASE e este se comunica com aquele através do vídeo.

Um fato importante a salientar é que, no MODO INTERATIVO, o dBASE não requer que o usuário tenha conhecimentos técnicos avançados, bastando que saiba lidar, pelo menos razoavelmente, com o microcomputador

funcionando com um dos sistemas operacionais para discos citados a seguir, no caso de computadores MSX: MSX DOS, HBDOS ou SOLXDOS.

Este primeiro volume de nosso curso é destinado exclusivamente ao estudo do MODO INTERATIVO do dBASE II PLUS.

5.4 - MODO PROGRAMÁVEL

No MODO PROGRAMÁVEL do dBASE, além de dispor de todos os recursos existentes no MODO INTERATIVO, o usuário poderá personalizar as diversas formatações de entrada e saída de dados e implementar determinadas conveniências específicas para manipulação dos mesmos, como, por exemplo, fazer o dBASE testar a validade de entrada de dados selecionados. No MODO INTERATIVO as únicas validades testadas são as de números e de dados locados em campos lógicos.

Outro recurso muito útil disponível através do MODO PROGRAMÁVEL e que citamos a título de exemplo é o de programar a geração das chamadas "pistas de auditoria", isto é, fazer com que determinadas alterações de dados (como preços, valores de bens etc.) sejam efetuadas como fatos à parte, de modo a poder recuperá-los e reconstituir sua situação anterior ou a primitiva, no caso de dúvida ou conveniência.

Ao contrário do MODO INTERATIVO, o MODO PROGRAMÁVEL do dBASE requer que o usuário possua certo grau de conhecimentos técnicos de programação estruturada, conhecimentos esses que se tornam assimiláveis sem grandes dificuldades quando o usuário passa a dominar o MODO INTERATIVO.

A utilização do MODO PROGRAMÁVEL do dBASE alarga indefinidamente os limites de suas aplicações específicas, propiciando ao programador experiente e imaginoso recursos indescritíveis para a criação e manipulação de arquivos ou bancos de dados.

O MODO PROGRAMÁVEL será estudado no segundo volume deste curso.

5. COMEÇANDO A USAR O dBASE II PLUS

Esperamos que as considerações até agora tecidas tenham demonstrado ao leitor que, para criar arquivos ou bancos de dados manipuláveis através de linguagens padronizadas, como BASIC, COBOL etc., é necessário desenvolver programas gerenciadores para os mesmos, ou SGBDs, na mesma linguagem.

Vimos também que o desenvolvimento de tais programas ou SGBDs:

- exigem do programador conhecimentos técnicos avançados;
- consomem muito tempo em seu desenvolvimento;
- oferecem dificuldades para alterações de sua estrutura inicial;
- não se revestem da condição de universalidade, pois, na maioria dos casos, só podem ser usados com os arquivos para os quais foram desenvolvidos;
- representam custos elevados em razão de tais particularidades.

Veremos agora que, contrariamente ao que parece, criar e manipular arquivos ou bancos de dados com a utilização do SGBD dBASE II PLUS é tarefa fácil, não exigindo do usuário nenhum conhecimento avançado.

Iniciaremos nossa experiência com a criação e organização de um arquivo e, através de seu uso, iremos descrevendo os comandos e recursos do dBASE.

Aconselhamos ao leitor interessado em adquirir rapidamente práti-

ca no uso do dBASE que, a partir deste ponto, passe a fazer a leitura do livro sentado frente ao computador ligado e que execute no mesmo os passos todos que serão descritos.

Antes, porém, devemos abrir o "pacote" dBASE II PLUS.

6.1 - ABRINDO O "PACOTE" dBASE II PLUS

O "pacote" dBASE II PLUS consiste em:

- 1 MANUAL MSX dBASE II.
- 1 DISQUETE COM OS PROGRAMAS DO SISTEMA.

O manual se caracteriza como um dicionário de comandos do dBASE II PLUS e é muito útil como tal: a apresentação dos comandos em ordem alfabética facilita sua consulta para "refrescamento" da memória quando ela "esquenta", dificultando sua rememoração espontânea.

O disquete do sistema contém os programas que constituem o dBASE II PLUS, para duas versões de visualização de vídeo:

- 40 colunas (TV ou monitor monocromático);
- 80 colunas (monitor monocromático).

Os programas existentes no disquete do sistema são os seguintes:

- DBBSE.COM
- DBBSEOVR.COM
- DBBSEMSG.TXT
- DGEN8.CMD
- GER8ARQ.CMD
- GER8MENU.CMD
- GER8ETIQ.CMD
- GER8REL.CMD
- COPIA8.BAT
- DB4SE.COM

- DB4SE0VR.COM
- DB4SEMSG.TXT
- DGEN4.CMD
- GER4ARQ.CMD
- GER4MENU.CMD
- GER4ETIQ.CMD
- GER4REL.CMD
- COPIA4.BAT

A finalidade de cada um desses programas será vista mais adiante.

ATENÇÃO!

O disquete do sistema pode ser utilizado somente para gerar cópias de disquetes de trabalho com o dBASE.

6.2 - COMO PREPARAR DISQUETES

DE TRABALHO DO dBASE II PLUS

No caso de o microcomputador operar com apenas um disk drive:

1. Formatar dois disquetes através de um dos seguintes sistemas operacionais para discos: MSXDOS, HBDOS ou SOLXDOS.
2. Copiar nos disquetes os programas que constituem um dos sistemas operacionais citados.

(Se for usada a versão de 80 colunas com o cartucho - interface - de 80 colunas produzido por EPCOM Equipamentos Eletrônicos Ltda., gravar nos disquetes o sistema HBDOS. Tal cartucho funciona apenas com o HBDOS. Se for usado o cartucho produzido pela MICROSOL Tecnologia Ltda., gravar o sistema MSXDOS ou o SOLXDOS.)

3. Instalar devidamente no disk drive o disquete de sistema do dBASE II PLUS.
4. Digitar o comando

COPIAS A B <RETURN>

Observação: a palavra 'return' usada na forma <RETURN> indicará

doravante que a tecla RETURN deverá ser pressionada uma vez, após a digitação do comando indicado.

5. Quando aparecer no vídeo a mensagem para colocar disquete no disk drive B, retirar o disquete de sistema do dBASE II PLUS e instalar um dos disquetes preparados nos passos 1 e 2.
6. Proceder inversamente quando nova mensagem no vídeo solicitar colocar disquete no disk drive A.
7. Continuar com o mesmo procedimento até ser exibida no vídeo a mensagem de que oito arquivos foram copiados.
8. Retirar do disk drive o disquete em que foram copiados os oito programas, colocar nele uma etiqueta com os dizeres dBASE II PLUS 80 COLUNAS e colocá-lo de volta no envelope protetor.

ATENÇÃO!

Lembrar que a etiqueta deve ser escrita antes de ser colada no disquete. Escrever sobre o disquete pode danificá-lo!

9. Colocar novamente no disk drive o disquete de sistema do dBASE II PLUS.
10. Digitar o comando

COPIA4 A B <RETURN>

11. Repetir os passos 5, 6 e 7.
12. Retirar o disquete em que foram copiados outros oito programas e colocar nele uma etiqueta com os dizeres dBASE II PLUS 40 COLUNAS, guardando-o em seguida no seu envólucro protetor.

No caso de o microcomputador operar com dois disk drives, o disquete de sistema do dBASE II PLUS deve ser instalado no disk drive A e os disquetes formatados para receber as diferentes cópias no disk drive B, devendo ser dados os comandos dos passos 4 e 10. Obviamente, não haverá necessidade de intercambiar os disquetes no processo de cópia.

Após o procedimento descrito, o disquete original de sistema do dBASE II PLUS deverá ser cuidadosamente guardado. Servirá para gerar novas cópias de trabalho quando forem necessárias.

Depois de copiados do disquete original em outro disquete, os nomes dos programas do dBASE II PLUS assumem sua forma normal, já que os números 4 e 8 existentes nos nomes do disquete original são automaticamente substituídos pela letra A nos nomes dos programas copiados. É por força desse fato que as versões para 40 e 80 colunas devem ser copiadas em disquetes distintos, pois os programas que as constituem assumem nomes idênticos e seriam copiados uns sobre os outros se se tentasse copiá-los no mesmo disquete.

Nesta fase de estudo do dBASE, é conveniente que todos os programas de cada versão sejam mantidos no disquete respectivo de trabalho, pois serão ou poderão ser usados para experiências. Todavia, quando o usuário tiver a prática necessária para o domínio do sistema, poderá organizar seus disquetes de trabalho apenas com os programas indicados a seguir:

- DOS (MSX, HB ou SOLX)
- DBASE.COM
- DBASEOVR.COM

liberando dessa forma mais espaço no disquete para arquivos, o que poderá ser necessário se dispuser de um disk drive apenas.

Os dois programas dBASE indicados são os únicos imprescindíveis para o funcionamento do sistema. Pode-se dizer que eles são o próprio sistema dBASE II PLUS.

O programa DBASEMSG.TXT é o que contém o auxílio ao usuário - conhecido como HELP - e as mensagens de erro do sistema, conforme descrições feitas nos itens 7.10 e 7.11.

Os demais programas são utilitários do dBASE II PLUS. São descritos no Capítulo 25.

As duas cópias geradas, uma para operar o dBASE II PLUS em 40 colunas de vídeo e a outra em 80, deverão ser utilizadas a critério do usuário ou em conformidade com o equipamento disponível.

Se dispuser de monitor de vídeo propriamente dito, a versão de 80

colunas deverá ser preferida, embora possa ser usada também a de 40.

No caso de usar aparelho de TV como monitor, deverá ser usada a versão de 40 colunas. Se for usada a de 80, a formatação da tela será prejudicada.

Por oportuno, lembramos que o cartucho de 80 colunas não funciona com aparelho de TV. É válido apenas para monitor propriamente dito.

Quando é usada a versão de 80 colunas com cartucho de expansão de vídeo para 80 colunas, mais dados ou informações podem ser visualizados a um só tempo na tela, obviamente:

Todavia, a versão de 80 colunas não altera nenhum outro ponto ou aspecto do trabalho e rendimento do dBASE II PLUS.

6.3 - PORQUE COPIAR O DOS

(MSX DOS, HBDOS, SOLXDOS) NO DIS-

QUETE DE TRABALHO DO dBASE II PLUS

Para o dBASE II PLUS funcionar, os programas que o constituem devem ser introduzidos (carregados) na memória do computador depois de estar ativado no mesmo um dos sistemas operacionais para disco citados abaixo.

Depois de introduzido na memória do microcomputador, o sistema operacional para disco passa a controlar todas as operações de entrada e saída entre a UPC (unidade de processamento central do microcomputador) e periféricos (disk drive(s), impressora, teclado, vídeo etc.). Nessa condição, possibilita o carregamento do dBASE, que passa a operar também sob o seu controle.

Os sistemas para discos que operam nos computadores de padrão MSX produzidos no Brasil e através dos quais o dBASE pode ser usado são:

- MSX DOS, da MICROSOFT Corporation.
- HBDOS, da EPCOM Equipamentos Eletrônicos Ltda.

Se o disquete de trabalho do dBASE II PLUS não contiver também o DOS, o procedimento para carregar o dBASE no microcomputador exigirá o manuseio de dois disquetes, com o seguinte procedimento:

1. Colocar no disk drive um disquete com o DOS e ativá-lo no microcomputador.
2. Retirar o disquete com o DOS e guardá-lo.
3. Colocar no disk drive o disquete de trabalho do dBASE II PLUS e ativá-lo.
4. Inverter o procedimento quando se quer retornar ao DOS.

Se ambos os sistemas estiverem reunidos no mesmo disquete, apenas esse disquete será manuseado, com economia de operações e de tempo e, conseqüentemente, diminuição de riscos de danos para os disquetes e o equipamento. Neste caso, um AUTOEXEC.BAT poderá ser gravado no disquete, tornando automático também o carregamento do dBASE no microcomputador, o que não só economiza tempo, como também é muito mais cômodo.

Considere-se ainda que o DOS ocupa apenas dez Kbytes no disquete, enquanto que uma só das versões do dBASE II PLUS ocupa quase dezesseis vezes o mesmo espaço.

(Oportunamente, veremos que alguns dos programas que compõem essa versão do dBASE podem ser transferidos para outro disquete, liberando mais espaço para arquivos, o que poderá ser uma necessidade se apenas um disk drive operar com o microcomputador.)

6.4 - ATIVANDO O dBASE NO MICROCOMPUTADOR

Qualquer das versões do dBASE II PLUS - 40 ou 80 colunas - poderá ser usada nas experiências descritas a seguir.

Partindo da pressuposição de que a maioria dos conjuntos iniciais de computação dispõe de apenas um disk drive, todas as experiências básicas serão efetuadas com apenas uma unidade acionadora de disquete,

ficando para a parte mais avançada do livro a realização de experiências com o microcomputador operando com dois disk drives.

PASSOS PARA ATIVAÇÃO DO dBASE II PLUS:

1. Ligar o monitor de vídeo e a fonte do disk drive.
2. Instalar adequadamente no disk drive um dos disquetes preparados de acordo com o item 6.2.
3. Ligar o microcomputador.

Depois de ativado o DOS, digitar:

DBASE <RETURN>

Logo em seguida surgirá no vídeo a mensagem:

Entre com a data de hoje ou tecle <return>.
(DD/MM/AA) :

Digitar a data usando um dos seguintes formatos:

16/7/87 16-7-87 16,7,87 16 7 87
16/07/87 <RETURN>

ou, simplesmente:

<RETURN>

Surgirá no vídeo o texto seguinte:

```
* (C) DATALOGICA dBASE II PLUS MSX *  
* VERSÃO 1.0 JANEIRO 1987 *  
Digite 'HELP', 'HELP dBASE' ou um comando
```

Observar o ponto (.) na linha após o texto e, após ele, o cursor na forma de um quadrado.

6.5 - O PONTO DO dBASE

O ponto (.) do dBASE indica que o sistema está ativado e pronto para receber comandos no modo direto.

Após a execução de qualquer comando do dBASE no modo direto, reaparece sempre na tela de vídeo o ponto, significando que o sistema está em prontidão para receber um novo comando.

6.6 - MODOS DIRETO E INDIRETO NO dBASE

No MODO DIRETO, os comandos do dBASE são executados quando são digitados e é pressionada a tecla RETURN em seguida.

O MODO DIRETO é a condição de funcionamento do MODO INTERATIVO, referido no item 5.3.

No MODO INDIRETO, os comandos do dBASE são executados através de programas, que devem ser elaborados na modalidade estruturada e com os comandos de sua linguagem de programação, com auxílio do compacto EDITOR do dBASE, embutido no próprio sistema, ou com auxílio de qualquer outro editor comercial, como WORD, WORDSTAR etc. O editor do dBASE é simples e fácil de ser operado, mas bom e eficiente.

Através dos outros editores de textos citados, a tarefa se torna sensivelmente mais complicada, como demonstra a prática.

Tais programas são executados pelo próprio dBASE, através do seu comando DO, do modo direto.

O MODO INDIRETO é condição de funcionamento do MODO PROGRAMÁVEL, referido no item 5.4.

7 - INFORMAÇÕES ÚTEIS

Neste capítulo, precedendo propositadamente aqueles em que será descrito e experimentado o dBASE II PLUS de modo tão objetivo e prático quanto possível, são apresentadas as informações preliminares consideradas necessárias para seu manejo adequado e eficiente.

Foram também agrupados neste capítulo indicadores visuais que facilitam a execução de diversos controles e comandos do dBASE, bem como outras informações julgadas úteis para facilitar sua compreensão, possibilitando melhor assimilação de seus recursos e peculiaridades.

Para facilitar consultas, cada tópico deste capítulo é apresentado em página separada.

7.1 - CONVENÇÃO ADOTADA NA

APRESENTAÇÃO DOS COMANDOS

Visando facilitar a descrição da sintaxe dos comandos e funções do dBASE, adotamos a seguinte convenção na sua apresentação:

- As palavras transcritas em letras maiúsculas referem-se ao comando propriamente dito e seus complementos sintáticos, devendo ser digitados como são apresentados. Exemplo:

```
DISPLAY [<escopo>] [FOR  
          <condição>] [TO PRINT]
```

- As indicações feitas em letras maiúsculas entre os sinais [**< >**] são componentes opcionais dos comandos.
- As indicações feitas em letras minúsculas entre os sinais **< >** referem-se a itens do comando que devem ser definidos pelo operador ou usuário.
- As indicações **<escopo>**, feitas na descrição de alguns comandos, referem-se à quantidade de registros que deverão ser processados, assumindo um dos seguintes parâmetros:

ALL - abrangendo todos os registros do arquivo.

NEXT n - abrangendo n registros, a partir do corrente.

RECORD n_o - abrangendo apenas o registro de número indicado.

Obs.: As posições indicadas acima para as cláusulas [**<escopo>**] e [**FOR <condição>**] são meramente casuais. Em comandos efetivos podem ocupar quaisquer posições.

7.2 - COMO DIGITAR OS COMANDOS DO dBASE

- Todos os comandos do dBASE e nomes de arquivos e de campos podem ser digitados em letras maiúsculas ou minúsculas, sendo automaticamente convertidos para letras maiúsculas.
- Os comandos devem ser digitados após o ponto (.) de prontidão do dBASE e finalizados com o pressionamento da tecla RETURN.
- Nomes de comandos propriamente ditos podem ser digitados apenas com as quatro letras iniciais. Exemplo:

DISP STRU

em vez de DISPLAY STRUCTURE.

- Cada linha de comando pode ter até 254 caracteres.
- Não é permitido usar mais de um comando por linha, embora possam constar do mesmo diversas cláusulas.
- Uma linha de comando muito longa pode ser dividida em diversas partes por meio do sinal ponto-e-vírgula (;). Exemplo:

```
DISPLAY FOR SALARIO = 19000 .AND. ;  
CARGO = "GERENTE" .OR. CARGO ;  
= "CONTADOR"
```

- As cláusulas estabelecidas por ALL, FOR e NEXT podem ocupar posição qualquer na linha de comando.
- A execução de um comando pode ser repetida após o retorno do ponto do dBASE, pressionando-se simultaneamente CONTROL e R.
- Um comando em execução pode ser cancelado, pressionando-se ESC.
- Pressionando-se de uma vez as teclas CONTROL e X, apaga-se toda a linha de comando sem executá-lo.

7.3 - COMANDOS DIGITADOS INCORRETAMENTE

Quando é dado um comando incorreto, o dBASE emite imediatamente uma mensagem de advertência e pergunta ao operador se deseja corrigi-lo para nova entrada. Vamos exemplificar o processo:

```
D          USER PESSOAL  <RETURN>
```

O dBASE avverte:

```
*** Comando irreconhecível.  
user pessoal.  
Corrigir e reentrar (Y/N) ?
```

Se a resposta for Y (= sim), é solicitada informação sobre a correção a efetuar e o processo prossegue. Se a resposta for N (= não), o dBASE retorna o ponto de prontidão para novo comando.

Façamos a experiência com Y:

```
D          Y  <RETURN>
```

Surge no vídeo:

```
Mudar de :
```

Devemos digitar:

```
USER  <RETURN>
```

Surge no vídeo:

```
Mudar para :
```

Digitamos:

USE <RETURN>

O dBASE prossegue:

USE PESSOAL

Mais correções (Y/N) ?

Respondemos:

N

Nesse estágio, o comando é executado normalmente.

Observamos que, dependendo da extensão do comando, pode ser mais rápido e conveniente responder N (= não) quando é feita a pergunta

Corrigir e reentrar (Y/N) ?

e digitar novamente o comando.

7.4 - ACENTUAÇÃO GRÁFICA NO dBASE II PLUS

Nomes de ARQUIVOS e de CAMPOS DE REGISTROS não podem ser acentuados graficamente nem usar c com cedilha.

As palavras que constituem conteúdos dos campos - dados - dos registros podem ser acentuadas e usar c com cedilha, mas serão reproduzidas corretamente apenas no vídeo.

ATRAVES DE IMPRESSORA, PALAVRAS ACENTUADAS GRAFICAMENTE E QUE TENHAM C COM CEDILHA SERÃO REPRODUZIDAS COM TRUNCAMENTO E/OU COM OS CARACTERES EM TAIS CONDIÇÕES TROCADOS.

Por tais razões, aconselha-se a não usar palavras acentuadas graficamente e com c com cedilha no dBASE II.

7.5 - ESPECIFICAÇÕES DO dBASE II PLUS

Campos por registro	32
Caracteres por registro	1000
Registros por banco de dados	65535
Caracteres por cadeia alfanumérica (string)	254
Precisão de campos numéricos - dígitos	10
Dígitos por variável numérica	10
Maior número positivo representável	$1,8 \times 10^{63}$
Menor número negativo representável	$1,0 \times 10^{-63}$
Variáveis de memória	64
Caracteres por linha de comando	254
Expressões em comando SUM	5
Caracteres de cabeçalho no comando REPORT	254
Campos permitidos no comando REPORT	24
Caracteres em chave indexada	100
Comandos GET pendentes	64
Arquivos abertos ao mesmo tempo	15
Tamanho de arquivo de comandos	ilimitado

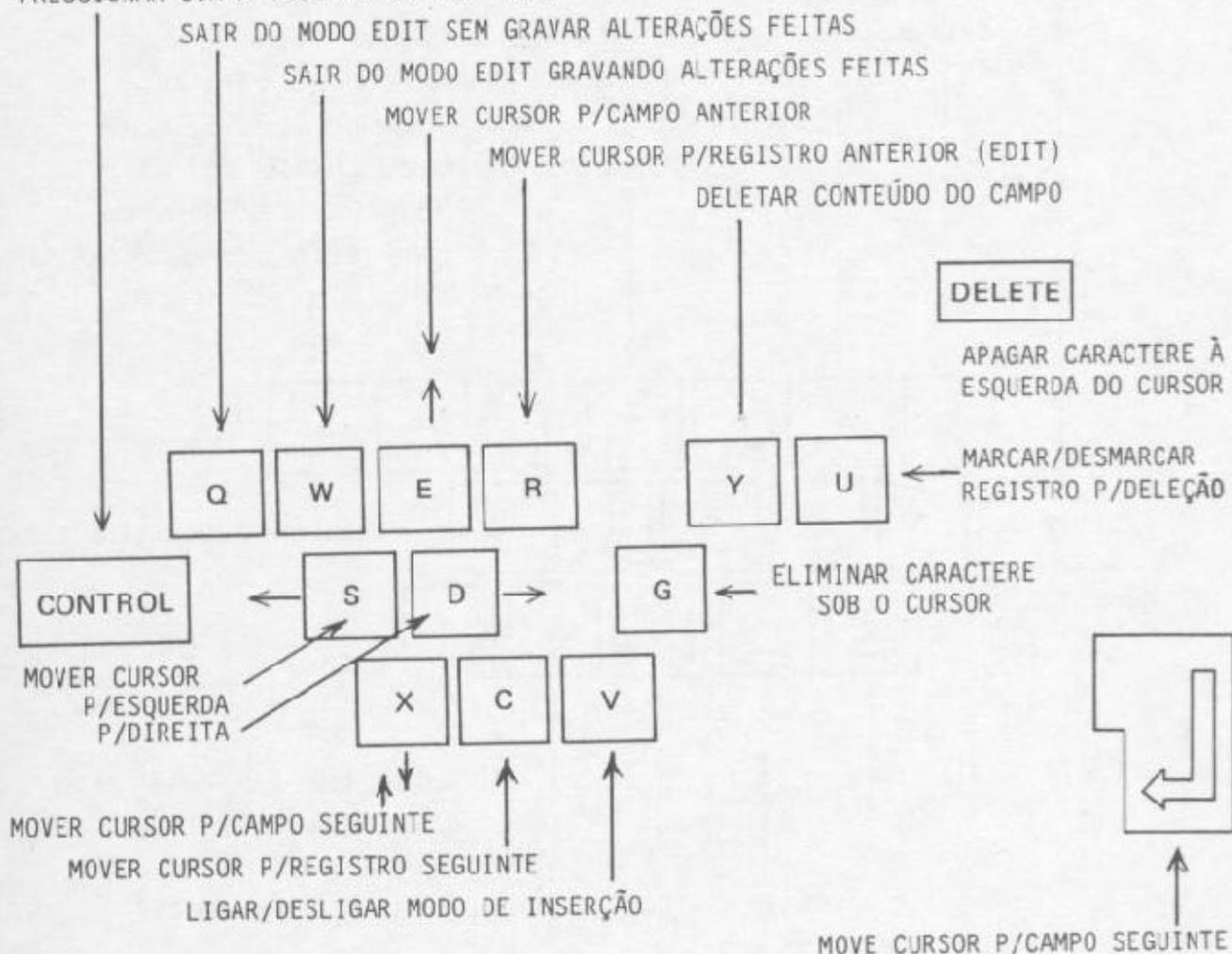
Obs: Os números citados expressam valores máximos permitidos.

7.6 - TECLAS DE CONTROLE DOS COMANDOS

APPEND E EDIT

(INDICADOR VISUAL)

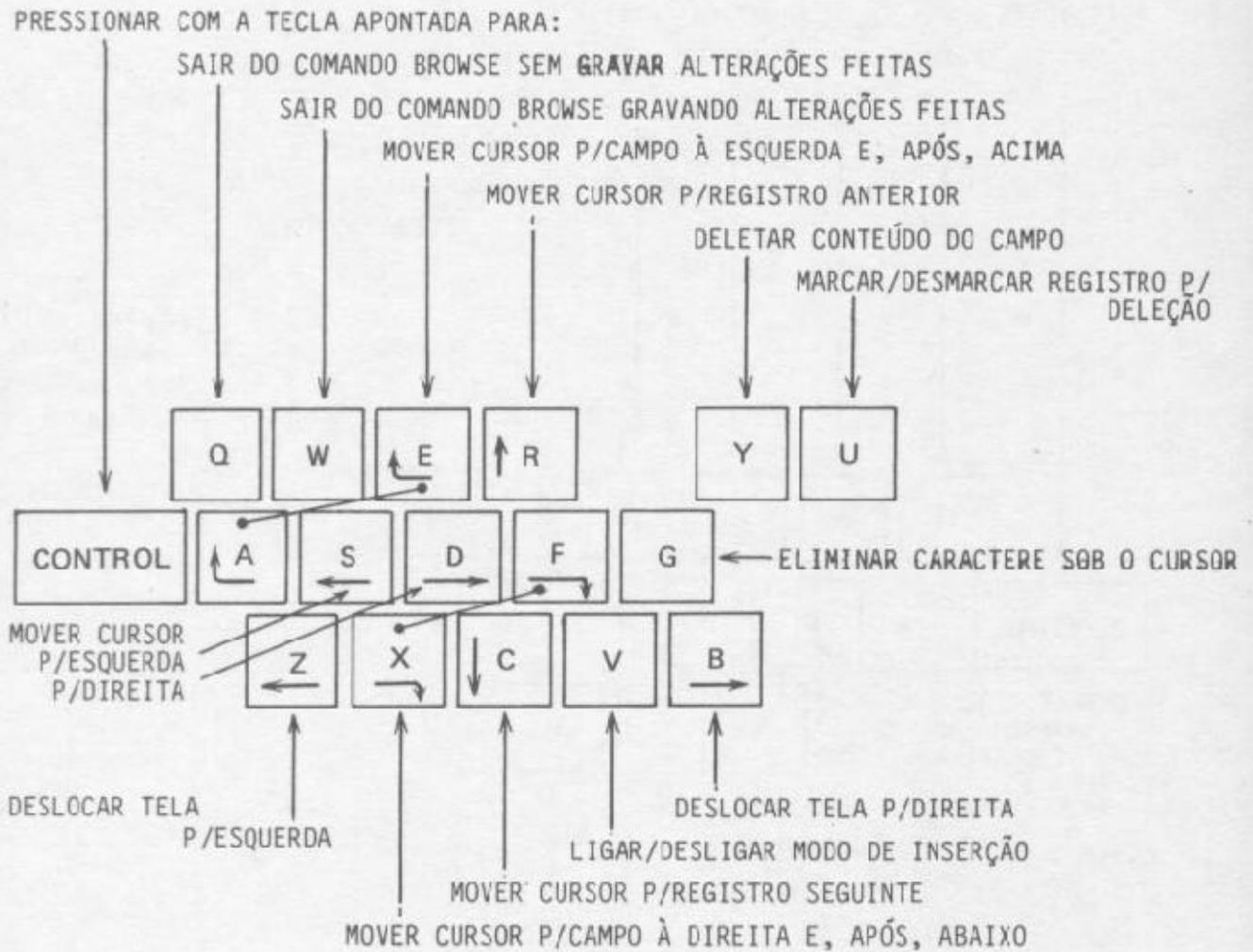
PRESSIONAR COM A TECLA APONTADA PARA:



Nos Capítulos 9 e 11 encontram-se detalhes sobre os comandos acima. A tabela dos mesmos se encontra no Capítulo 11.

7.7 - TECLAS DE CONTROLE DO COMANDO BROWSE

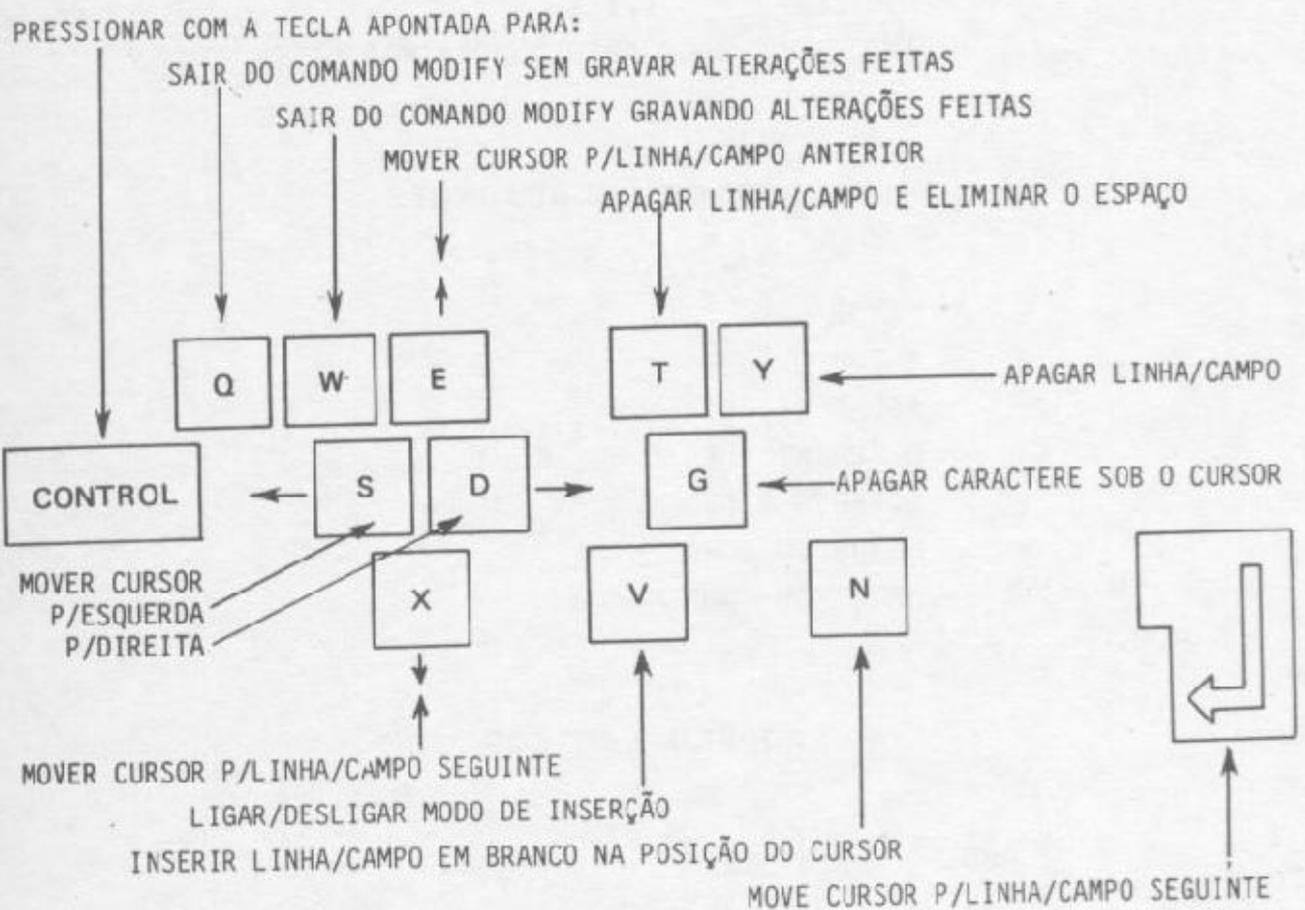
(INDICADOR VISUAL)



No Capítulo 11 encontram-se detalhes sobre o comando BROWSE e a tabela dos comandos acima.

7.8 - TECLAS DE CONTROLE DO COMANDO MODIFY

(INDICADOR VISUAL)



Nos Capítulos 19 e 21 encontram-se detalhes referentes ao comando MODIFY. A tabela dos comandos acima se encontra no Capítulo 19.

7.9 - OPERADORES VÁLIDOS NO dBASE II PLUS

OPERADORES MATEMÁTICOS

- + - ADIÇÃO (3)
- - SUBTRAÇÃO (4)
- * - MULTIPLICAÇÃO (1)
- / - DIVISÃO (2)
- () - INDICADORES DE PRECEDÊNCIA NAS OPERAÇÕES

OPERADORES RELACIONAIS

- < - MENOR QUE
- > - MAIOR QUE
- = - IGUAL A
- <> - DIFERENTE DE
- ≠ - DIFERENTE DE
- <= - MENOR OU IGUAL
- >= - MAIOR OU IGUAL

OPERADORES LÓGICOS

- .AND. - E (2)
- .OR. - OU (3)
- .NOT. - NÃO (1)

OPERADORES DE CONCATENAÇÕES

- + - JUNTA CONTEÚDOS DE CAMPOS, DE CADEIAIS ALFANUMÉRICAS OU DE VARIÁVEIS.
- - IDEM, ELIMINANDO ESPAÇOS EM BRANCO EXISTENTES NOS FINAIS DOS MEMOS.

§ - COMPARA CADEIAS ALFANUMÉRICAS, RETORNANDO T (= TRUE) SE A PRIMEIRA CADEIA É IDENTICA A SEGUNDA OU ESTÁ CONTIDA NELA.

OBSERVAÇÃO: Os números entre parênteses indicam a precedência em cada grupo.

Quando operadores de diversos grupos são usados em uma mesma expressão, a precedência dos mesmos segue a seguinte ordem:

- 1 - MATEMÁTICOS
- 2 - RELACIONAIS
- 3 - LÓGICOS

7.10 - HELP - COMANDO DE AUXÍLIO DO dBASE

Conforme descrito no item 6.4, ao ser ativado, o dBASE exibe no vídeo sua etiqueta de identificação e, logo após a mesma, a frase:

Digite 'HELP', 'HELP DBASE' ou um comando.

Essa frase indica um interessante recurso do dBASE, pouco utilizado, ao que parece, em razão de seu alcance não ser bem conhecido.

Sendo recurso de grande valia especialmente para usuários principiantes do sistema, julgamos oportuno descrever o que é oferecido pelo HELP do dBASE II PLUS e como assessor seus principais serviços.

Os auxílios ao usuário oferecidos pelo dBASE II PLUS são obtidos quando se aciona o comando HELP seguido pela palavra-chave do auxílio desejado.

A seguir são listadas as palavras-chaves dos auxílios do dBASE. O auxílio correspondente fornecido é indicado entre parênteses.

COPIA (Como proceder para obter cópias do dBASE do disco mestre.)

DBASE (Introdução e listagem dos programas do dBASE II PLUS. Informação sobre o comando HELP.)

ERROS (Descrição das mensagens sobre erros de uso e providências cabíveis.)

FUNÇÕES (Listagem e descrição resumida das funções do dBASE.)

LIMITES (Especificação e características do dBASE II PLUS.)

TELA CHEIA (Listagem das teclas de controle do cursor e suas funções principais.)

UTILITARIOS (Indicação sucinta de alguns programas que podem trabalhar com o dBASE.)

Para se obter auxílio sobre o uso das teclas de controle do cur-

sor, por exemplo, comanda-se:

```
HELP TELA CHEIA <RETURN>
```

Além dos citados acima, o dBASE presta auxílio sobre qualquer dos seus comandos, bastando que se digite o nome do mesmo após o comando HELP. Exemplo:

```
HELP BROWSE <RETURN>
```

Se for dado apenas o comando:

```
HELP <RETURN>
```

será exibida uma listagem completa dos comandos do dBASE II PLUS, com descrição sinóptica dos mesmos.

O comando HELP pode ser acionado sempre que o dBASE estiver pronto para receber comandos.

7.11 - MENSAGENS DE ERRO DO dBASE II PLUS

Como foi comentado no item 7.10, as mensagens de erro do dBASE II PLUS podem ser exibidas no vídeo através do comando HELP ERROS.

Mas tal comando não permite selecionar a exibição de uma mensagem determinada, a fim de estudar-lhe o significado. Todas são exibidas no vídeo, em lotes sucessivos de telas cheias e fora de ordem alfabética.

E há situações em que o usuário necessita saber o significado de uma mensagem com certa urgência, sem removê-la do vídeo - onde se encontra próxima do comando ou fato que a originou - a fim de analisar a sua causa. Tal ocorrência é tão mais provável quanto mais novato é o usuário envolvido.

Assim, visando possibilitar acesso rápido a tais mensagens e suas significações, reunimo-las nesta secção, em ORDEM ALFABÉTICA, cada uma com a respectiva significação ou indicação da providência cabível, reproduzidas exatamente como são transmitidas pelo dBASE II PLUS.

- "ALÉM DA STRING" - Reescreva substring (\$) com o parâmetro correto.
- "ARQUIVO DE COMANDO NÃO ENCONTRADO" - Cheque o nome do arquivo ou o drive default.
- "ARQUIVO DE FORMATO NÃO FOI CONFIGURADO" - Use o arquivo .FMT apropriado.
- "ARQUIVO DE FORMATO NÃO PODE SER ABERTO" - Cheque a integridade deste arquivo.
- "ARQUIVO DE ÍNDICE NÃO PODE SER ABERTO" - Cheque nome ou índice do banco de dados.
- "ARQUIVO JÁ EXISTE" - Delete o arquivo desnecessário antes de usar RENAME.
- "ARQUIVO NÃO É BANCO DE DADOS DO dBASE II" - Arquivo DBF aberto não foi criado pelo dBASE II.

- "ARQUIVO NÃO EXISTE" - Use DISPLAY FILE LIKE *.* para ter certeza se o arquivo realmente existe.
- "ARQUIVO NO MOMENTO ESTÁ ABERTO" - Use um comando USE ou CLEAR para fechar este arquivo.
- "BANCO DE DADOS EM USO NÃO É INDEXADO" - FIND permitido somente em banco de dados indexados.
- "CAMPO DECIMAL INVÁLIDO" - Reentre a parte decimal da definição do campo.
- "CHAVES NÃO SÃO DO MESMO TAMANHO" - Comando UPDATE requer chaves idênticas.
- "COMANDO DESCONHECIDO" - Cheque o conteúdo. dBASE não conseguiu interpretar o comando.
- "DADO NÃO ENCONTRADO" - Reescreva o comando REPLACE, ou veja a estrutura do arquivo e o nome correto de campo.
- "DIRETÓRIO ESTÁ CHEIO" - O sistema operacional não pode gravar mais arquivos no disco.
- "DISCO ESTÁ CHEIO" - Não há mais espaço no disco. Use o comando DELETE FILE para apagar arquivos desnecessários e conseguir mais espaço.
- "DIVISÃO POR ZERO" - Houve uma tentativa de divisão por zero.
- "ERRO DE SINTAXE" - dBASE não entendeu o comando.
- "ERRO DE SINTAXE NA ESPECIFICAÇÃO DO FORMATO" - O comando @ SAY GET PICTURE foi usado impropriamente.
- "ERRO DE SINTAXE, REDIGITE" - INPUT, ACCEPT, REPORT requerem automaticamente entrada correta.
- "ERRO INTERNO, NOTIFIQUE SUPORTE" - Contacte o departamento de suporte da DATALÓGICA.
- "EXCESSO DE ARQUIVOS ABERTOS" - Somente 16 arquivos de todos os tipos (.FMT, .NDX ou de COMANDOS) podem ser abertos ao mesmo tempo.
- "EXCESSO DE CARACTERES" - Somente quando não em modo TELA-CHEIA. O

dado entrado excedeu o tamanho alocado para aquele campo.

"EXCESSO DE RETURNS ENCONTRADO" - Provavelmente um erro na estrutura de um arquivo de comandos. Cheque o número e a localização de RETURNS.

"EXCESSO DE VARIÁVEIS DE MEMÓRIA" - Use no máximo 64 variáveis de memória.

"EXCEDIDO O LIMITE DE ENCAIXES" - Não pode haver mais de 16 arquivos de comandos abertos ao mesmo tempo.

"EXPRESSÃO NÃO NUMÉRICA" - Comando SUM necessita de uma expressão numérica.

"FIELD NÃO ENCONTRADO" - Reescreva a linha de comando CHANGE.

"FIM DE ARQUIVO INESPERADO" - O banco de dados em uso (USE) não está no formato correto. dBASE não assegura que o arquivo é do tipo DBF.

"FOR NÃO ENCONTRADO" - Reescreva comando JOIN com a sintaxe correta.

"FROM NÃO ENCONTRADO" - Reescreva comando UPDATE com a sintaxe correta.

"ÍNDICE NÃO COMPATÍVEL COM O BANCO DE DADOS" - dBASE não pode combinar a chave indexada com o banco de dados. Tente outro arquivo indexado.

"MACRO NÃO É CADEIA DE CARACTERES" - A variável a ser expandida pela macro (&) deve ser uma string de caracteres.

"MAIS DO QUE 7 ARQUIVOS INDEXADOS SELECIONADOS" - O número máximo de arquivos indexados abertos ao mesmo tempo é 7. Um número menor dará melhor performance.

"MAIS DO QUE 5 CAMPOS A SOMAR" - SUM limitado a 5 campos ao mesmo tempo.

"NÃO ENCONTRADO" - Mais um diagnóstico do que uma mensagem de erro. dBASE não pode descobrir a chave. O número de registro é colocado em 0.

"NÃO PODE ABRIR ARQUIVO" - Cheque a existência ou integridade de um arquivo MEM ou HEX.

- "NÃO PODE INSERIR, NÃO HÁ REGISTROS NO BANCO DE DADOS" - Use o comando APPEND no lugar.
- "NOME DE ARQUIVO INVÁLIDO" - Erro de sintaxe no nome do arquivo.
- "NOME DE CAMPO INVÁLIDO" - Redefina o nome do arquivo enquanto em CREATE.
- "NOME DE VARIÁVEL ILEGAL" - Somente alfanuméricos e "-" são permitidos em variáveis e nomes de campos. Redefina a variável ou nome do campo.
- "NÚMERO DE ARQUIVO NÃO DESIGNADO" - Erro interno, entre em contato com o suporte. Se HELP é usado, DBASEMSG.TXT deve existir no drive corrente.
- "ON NÃO ENCONTRADO" - Reescreva o comando UPDATE ou INDEX corretamente.
- "REGISTRO FORA DE ALCANCE" - Número de registro chamado é maior que o número de registros do banco de dados. O arquivo indexado não está atualizado. Reindexe-o com o comando REINDEX.
- "REGISTRO NÃO INDEXADO" - Arquivo indexado não atualizado após um registro ter sido incluído. REINDEX o arquivo.
- "SEM ESPAÇO PARA VARIÁVEIS DE MEMÓRIA" - Reduza o número ou tamanho das variáveis de memória.
- "SEM EXPRESSÃO PARA SOMAR" - Comando SUM necessita de uma expressão numérica.
- "TAMANHO DE CAMPO INVÁLIDO" - Redefina o tamanho do campo de entrada.
- "TAMANHO DO REGISTRO ULTRAPASSA O LIMITE DE 1000 BYTES" - Reduza o tamanho de alguns campos ou crie um segundo banco de dados com a mesma chave.
- "TENTATIVA DE GERAR MAIS QUE 65534 REGISTROS" - A cláusula FOR permite ligação de demasiados registros. Faça-o mais comprido.
- "TIPO DE CAMPO INVÁLIDO" - Deve ser C (character), N (numérico), ou L (lógico).
- "TIPO DE DADO ILEGAL" - SORT não pode sortear um campo lógico.

"TO NÃO ENCONTRADO" - Reescreva o comando com a sintaxe correta.

"VALOR ILEGAL PARA GO TO" - Deve ser maior que 0 e menor que 65534.

"VARIÁVEL NÃO FOI ENCONTRADA" - Precisa-se criar a variável ou checar o conteúdo do nome do campo na estrutura do banco de dados.

"WITH NÃO ENCONTRADO" - Reescreva comando REPLACE corretamente.

7.12 - LOCALIZAÇÃO DOS COMANDOS E FUNÇÕES DO dBASE II PLUS ESTUDADOS NESTE LIVRO

COMANDOS:

?, ??, pág. 84	INDEX, pág. 100	SET ALTERNATE, pág. 188
ACCEPT, 211	INDEX ON, 97	SET BELL, 225
APPEND, 69	INPUT, 210	SET CARRY, 74
APPEND FROM, 198	INSERT, 73	SET CONFIRM, 73
BROWSE, 118	JOIN, 168	SET CONSOLE, 225
CHANGE, 123	LIST, 65	SET COLON, 70
CLEAR, 167, 213	LIST FILE, 194	SET DATE, 219
CONTINUE, 91	LIST MEMORY, 215	SET DEFAULT, 150
COPY, 189	LIST STATUS, 75	SET DELETED, 129
COPY STRUCTURE, 190	LIST STRUCTURE, 190	SET EJECT, 142
COUNT, 93	LOCATE, 91	SET ESCAPE, 226
CREATE, 63	MODIFY COMMAND, 182	SET EXACT, 89
DELETE, 128	MODIFY STRUCTURE, 195	SET HEADING, 139
DELETE FILE, 193	PACK, 129	SET INDEX, 99
DISPLAY, 65	QUIT, 68	SET INTENSITY, 226
DISPLAY FILE, 67, 194	RECALL, 129	SET LINKAGE, 163
DISPLAY MEMORY, 215	REINDEX, 103	SET MARGIN, 146
DISPLAY STATUS, 75	RELEASE, 212	SET PRINT, 81
DISPLAY STRUCTURE, 66	RENAME, 192	SET RAW, 226
DO, 184	REPLACE, 125	SET SCREEN, 114
EDIT, 110	REPORT, 135	SKIP, 85
ERASE/? CHR(12), 79	RESTORE, 214	STORE, 209
FIND, 101	SAVE, 214	SUM, 94
GO/GOTO, 83	SELECT, 158	TOTAL, 154
HELP, 54	SET, 75	UPDATE, 173
		USE, 65

FUNÇÕES:

&, 217	CHR, 218	RANK, 222
@, 216	DATE, 219	STR, 90
*, 129	EOF, 219	TRIM, 222
#, 84	FILE, 220	TYPE, 223
\$, 90	INT, 220	VAL, 224
!, 87	LEN, 221	

B - CRIANDO O PRIMEIRO ARQUIVO DE DADOS

A estrutura de nosso primeiro arquivo de dados - simples, com finalidade puramente didática - será baseada nas informações contidas na pequena ficha apresentada a seguir.

(Será conveniente manter uma cópia da mesma juntamente com o disquete, para facilitar a identificação do arquivo e de sua estrutura. A rápida identificação da estrutura do arquivo é muito útil quando se pretende recuperar seus dados por programa gerenciador que não seja o seu próprio.)

Nome do arq.: PESSOAL.DBF		Disco: 123		
Assunto: REGISTRO DE EMPREGADOS				
Campo	T	E	D	Referência
1 - NOME	C	15	-	NOME DO EMPREGADO
2 - CARGO	C	10	-	FUNÇÃO EXERCIDA
3 - SALARIO	N	5	-	REMUNERAÇÃO
4 - DATADM	C	5	-	MÊS/ANO DA ADMISSÃO
T=Tipo / E=Espaço / D=Decimais				

FIGURA FB.1

(Se houver dúvidas quanto às definições de CAMPO, TIPO e ESPAÇO, convirá reler o Capítulo 3.)

Quanto a nomes, são válidos no dBASE:

PARA ARQUIVOS: até 11 caracteres, sendo 8 como parte principal e 3 como extensão, devendo ambas ser separadas por um ponto e começar obrigatoriamente com uma letra. Não são permitidos espaços em branco.

PARA CAMPOS: até 10 caracteres, devendo o primeiro ser obrigatoriamente uma letra. Espaços em branco não valem.

Comando CREATE

*** CREATE cria arquivos. Sua sintaxe é:

```
CREATE <nome do arquivo>
```

Estando o dBASE ativado conforme item 6.4, digitar:

```
CREATE PESSOAL <RETURN>
```

O dBASE responderá com a seguinte mensagem no vídeo:

```
Entre com a estrutura do registro :  
Nome do campo, tipo, tamanho, casas decimais.  
001
```

Digitar, a partir do ponto onde se encontra o cursor:

```
NOME,C,15 <RETURN>
```

Surgirá no vídeo, em seguida:

```
002
```

Digitar, sempre a partir de onde se encontra o cursor:

```
CARGO,C,10 <RETURN>
```

Surgirá no vídeo, em seguida:

003

Digitar:

SALARIO,N,5 <RETURN>

Surgirá no vídeo:

004

Digitar:

DATADM,C,5 <RETURN>

Quando surgir no vídeo o número do campo 005, apenas:

<RETURN>

*** Pressionando-se a tecla RETURN quando o dBASE aguarda definição de um novo campo, encerra-se a entrada de informações para a definição de estrutura do arquivo.

(Os nomes de ARQUIVOS e de CAMPOS devem ser digitados sem acentos e com c sem cedilha. Caso contrário, o dBASE os invalidará.)

O dBASE perguntará em seguida:

Incluir dados agora ?

Como não vamos ainda incluir dados no arquivo, respondemos:

N <RETURN>

O ponto do dBASE reaparecerá, aguardando novo comando.

Nesse ponto, o vídeo estará exibindo a seguinte tela:

Entre com a estrutura do registro :

Nome do campo, tipo, tamanho, casas decimais.

```
001   NOME,C,15
002   CARGO,C,10
003   SALARIO,N,5
004   DATADM,C,5
005
```

Incluir dados agora ? N

-

Em caso de erro na definição da estrutura:

Se ocorrer algum erro durante a digitação dos dados que definem a estrutura, deve-se pressionar a tecla RETURN quando o dBASE exibir no vídeo o número do próximo campo a definir. O dBASE perguntará então:

Incluir dados agora ?

Digitar como resposta:

N <RETURN>

e, em seguida:

DELETE FILE PESSOAL <RETURN>

O dBASE informará:

Arquivo foi eliminado.

Em tal caso, a estrutura do arquivo deve ser refeita desde o início, através do comando CREATE.

Comandos USE, DISPLAY e LIST

*** USE seguido do nome de um arquivo, ativa-o para uso. Sem parâme-

tros, fecha arquivos em uso. Sua sintaxe é:

```
USE <nome do arquivo> [INDEX <nome(s)
do(s) arquivo(s) de índice>]
```

*** DISPLAY e LIST exibem o que lhes for determinado. Sua sintaxe é:

```
DISPLAY [<escopo>] [FOR <condição>] [<no-
me(s) do(s) campo(s)>] [TO PRINT] [OFF]
```

Depois de definida a estrutura do arquivo, é conveniente verificar se ela está certa, antes de iniciar a entrada de dados. Para isso, devemos colocar o arquivo em uso:

```
USE PESSOAL <RETURN>
```

Em seguida, podemos comandar a exibição da estrutura do arquivo:

```
DISPLAY STRUCTURE <RETURN>
```

O dBASE exibirá no vídeo:

```
Número de registros :00000
Data da última atualização :16/07/87
Uso primário do banco de dados.
Cpo      Nome      Tipo Tam.   Dec.
001      NOME      C      015
002      CARGO     C      010
003      SALARIO  N      005
004      DATADM   C      005
** TOTAL **                00030.
```

(Observar que foram definidos 35 espaços para os 4 campos do arquivo, mas o dBASE apresenta o total de 36. Esse espaço extra é acrescentado automaticamente pelo dBASE para locação do CÓDIGO DE DELEÇÃO, de seu controle. Notar a coluna DEC, destinada a registrar casas decimais de campos numéricos.)

*** Todos os comandos do dBASE podem ser digitados com suas 4 primeiras letras apenas, no formato maiúsculo ou minúsculo. Em vez de:

DISPLAY STRUCTURE <RETURN>

pode-se digitar apenas:

DISP STRU ou disp stru <RETURN>

Ainda antes de começarmos a entrada de dados para o arquivo, vamos verificar se ele já se encontra gravado no disquete. Para tanto, comandamos:

DISPLAY FILES <RETURN>

Em seguida, serão exibidas no vídeo as seguintes informações sobre o arquivo criado por nós:

Arquivo de dados	# Regs	Ultima data
PESSOAL DBF	00000	16/07/87

Observar a extensão DBF, não digitada por nós e sim acrescentada automaticamente pelo dBASE.

*** Quando não entramos com EXTENSÃO DE NOME para arquivos criados, o dBASE a acrescenta automaticamente.

No caso de ARQUIVOS DE DADOS, o dBASE adota a extensão .DBF (DATA BASE FILE).

As extensões adotadas automaticamente pelo dBASE são:

- .BAK = Arquivo cópia (BACK UP).
- .CMD = Arquivo de comandos (CoManDos).
- .DBF = Arquivo de dados (DATA BASE FILE).
- .FMT = Arquivo de formatação de tela de vídeo.
- .FRM = Arquivo de formatação de relatórios.
- .NDX = Arquivo de índices de arquivo de dados.
- .MEM = Arquivo de variáveis de memória.
- .PRG = O mesmo que .CMD.

.TXT = Arquivo de texto.

*** Quando a extensão do nome do arquivo é uma das próprias do dBASE, não é necessário digitá-la ao ser o mesmo selecionado para uso, a não ser que o arquivo tenha diversas modalidades (de índice, de formatação etc.) que usem o mesmo nome principal.

*** O usuário poderá definir a seu critério as extensões de nomes para seus arquivos. Em tal caso, todavia, ao ser selecionado um arquivo para uso, seu nome deverá ser digitado sempre com a extensão de nome que o caracteriza.

Para se ter todos os nomes dos arquivos do disquete listados no vídeo quando o dBASE está ativado, deve-se digitar:

DISPLAY FILES LIKE *.* <RETURN>

Comando QUIT

*** QUIT desativa o dBASE, fecha os arquivos eventualmente abertos e retorna ao DOS o controle do microcomputador. Sua sintaxe é:

QUIT

Vamos encerrar agora nossa primeira sessão com o dBASE, para depois começar a inserir dados no nosso arquivo.

Para encerrá-la, devemos apenas digitar:

QUIT <RETURN>

9 - INSERINDO DADOS

NO ARQUIVO

Uma vez criado e estruturado, o arquivo PESSOAL está pronto para receber registros.

Depois de ativar o dBASE conforme item 6.4, digitar:

```
USE PESSOAL <RETURN>
```

Comando APPEND

*** APPEND permite inserir registros no arquivo, a partir do primeiro registro livre no mesmo. (Ver APPEND FROM no item 21.5.) Sua sintaxe é o próprio nome apenas:

```
APPEND
```

Digitar:

```
APPEND
```

O dBASE exibirá no vídeo o seguinte quadro, que chamaremos de FICHA EM BRANCO DO REGISTRO Nº 1:

```

REGISTRO 00001
NOME      :           :
CARGO     :           :
SALARIO   :           :
DATADM    :           :

```

Os sinais : : sinalizam os tamanhos dos campos, em conformidade com a estrutura definida para o arquivo. Sua exibição no vídeo é controlada pelo comando descrito a seguir:

Comando SET COLON ON/OFF

*** SET COLON ON/OFF ativa ou desativa a exibição dos delimitadores dos campos de registros. Seu estado é estabelecido automaticamente em ON ao ser o dBASE ativado. Sua sintaxe é:

```
SET COLON ON ou OFF
```

Ao ser exibido no vídeo um registro "em branco" para entrada de dados, o cursor estará posicionado após o primeiro sinal de dois pontos (:)-na linha correspondente ao primeiro campo do registro, que é, no caso, o campo NOME.

Antes de começar a digitar os dados para preenchimento da ficha, é oportuno lembrar que, nessa fase de entrada de dados, podem ocorrer erros de digitação cujas correções exijam deslocamentos do cursor para diversas direções ou pontos da "ficha".

Para conhecer os diversos recursos oferecidos pelo dBASE nesse sentido, consulte o "INDICADOR VISUAL DAS TECLAS DE CONTROLE DOS COMANDOS APPEND E EDIT", no Capítulo 7.

Começemos a digitar:

```
Marco Antonio
```

Como o nome não preenche todo o espaço destinado a ele (linha li-

mitada por : no começo e : no fim), pressionar a tecla RETURN.

O cursor saltará para a linha referente a CARGO. Digitar então:

Desenhista

A palavra "desenhista" ocupa todo o espaço destinado à descrição do cargo. Nesse caso, o cursor salta automaticamente para a linha destinada a SALARIO. Digitar:

23000

O cursor estará na linha destinada a DATADM. Digitar:

07/77

Todos os campos do registro nº 1 já foram preenchidos. Nessa condição, o dBASE colocará automaticamente no vídeo a FICHA EM BRANCO DO REGISTRO Nº 2.

Vamos preenchê-la adotando o critério descrito com relação à ficha nº 1, utilizando os seguintes dados:

Luis Carlos

Contador

21000

12/78

Ao findar a digitação desses dados, o dBASE colocará no vídeo a FICHA EM BRANCO DO REGISTRO Nº 3. Vamos preenchê-la com os dados:

Maria Cecilia

Pedagoga

20000

06/81

O mesmo procedimento deverá ser adotado com os dados apresentados a seguir, referentes a outros três funcionários:

Maria Beatris
Psicologa
19000
10/82

Julio Cesar
Gerente
30000
04/87

Paulo Sergio
Engenheiro
25000
09/82

Quando for colocada pelo dBASE no vídeo a FICHA EM BRANCO DO REGISTRO Nº 7, digitar apenas:

<RETURN>

*** Encerra-se a entrada de dados no arquivo, pressionando-se a tecla RETURN quando uma nova FICHA EM BRANCO DE REGISTRO é colocada pelo dBASE no vídeo, para preenchimento por digitação.

Como foi dito na sua descrição, APPEND possibilita inserir dados em arquivo a partir do primeiro registro livre do mesmo. Ao ser comandado para uma nova inserção em relação a um arquivo que já tenha seis registros ocupados, por exemplo caso do arquivo PESSOAL recém criado, a inserção será feita automaticamente como registro número sete.

No uso prático, todavia, ocorrem situações em que é preciso inserir um novo registro - ou diversos - em posições intermediárias do arquivo.

Para tais situações, o dBASE dispõe de um comando específico:

Comando INSERT

*** INSERT insere registro em qualquer posição do arquivo. Sua sintaxe é:

INSERT [BEFORE] [BLANK]

Com a opção BEFORE, o novo registro é inserido na posição anterior à do ponteiro de registros no momento do comando, de modo que os dados desejados possam ser inseridos nele no mesmo ato.

Se não for usada a opção BEFORE, a inserção será feita na posição seguinte à do ponteiro de registros, também de modo que o mesmo possa receber dados no mesmo ato.

Usado com a opção BLANK, um registro em branco é inserido na posição posterior à do ponteiro de registros, de modo que o mesmo fique disponível para entrada de dados em outra oportunidade.

Comando SET CONFIRM ON/OFF

Na entrada de dados, para que o cursor se desloque para o campo ou registro seguintes, é necessário teclar RETURN quando o dado digitado não preenche todo o campo respectivo.

Por outro lado, quando o dado digitado preenche o campo todo, o cursor salta automaticamente para o campo seguinte, ou para o próximo registro em branco se não houver mais campos a preencher.

Esta última condição confere ao processo de entrada de dados mais agilidade e rapidez. Todavia, não sendo o dado confirmado para entrada pelo acionamento da tecla RETURN e ocorrendo erro na digitação, será necessário retornar o cursor ao campo para efetuar a correção, ou efetuar a correção posteriormente através dos comandos EDIT ou BROWSE no caso de o cursor já ter pulado para o registro seguinte a preencher.

*** Para que o cursor salte ou não automaticamente para o campo ou registro seguintes durante o processo de entrada de dados, usa-se o comando SET CONFIRM, cuja sintaxe é:

SET CONFIRM ON ou OFF

Ao ser ativado o dBASE, SET CONFIRM está normalmente em OFF.

SET CONFIRM deve ser acionado com o arquivo em uso e antes do comando APPEND. Exemplo:

```
D          SET CONFIRM ON   <RETURN>
D          APPEND
```

Comando SET CARRY ON/OFF

Na fase de entrada de dados para o arquivo, há situações em que os mesmos são repetitivos (dados sobre uma frota de veículos de mesma marca, ano e cor, em que apenas os números de placas são diferentes, por exemplo).

*** O comando SET CARRY ON permite que os dados do último registro efetuado sejam transportados para o próximo a dar entrada. Nessa condição, o cursor deve ser deslocado para o campo do dado repetido a ser alterado e, depois de efetuada a alteração, para o último campo.

Estando o cursor no último campo do registro, teclando-se simultaneamente CONTROL e X, ou simplesmente RETURN, será apresentado no vídeo o próximo registro, para idêntico procedimento.

Para interromper a entrada de dados, bastará teclar RETURN quando um novo registro repetido for exibido no vídeo.

A sintaxe do comando é:

```
SET CARRY ON ou OFF
```

Quando o dBASE é ativado, SET CARRY está normalmente em OFF.

Deve ser ativado com o arquivo em uso e antes de APPEND. Exemplo:

```
D          SET CARRY ON   <RETURN>
D          APPEND        <RETURN>
```

Comando DISPLAY STATUS ou LIST STATUS

Além das controladas por SET CONFIRM e SET CARRY, o dBASE dispõe de outras condições funcionais, as quais são estabelecidas automaticamente quando ele é ativado. Em computação, tais condições são denominadas padrões "default", já que são estabelecidas a título inicial (ou precário, diríamos), podendo ser mudadas pelo usuário a qualquer momento, a seu bel-prazer. Todas essas condições são controladas pelo comando descrito a seguir:

Comando SET

*** SET funciona como uma espécie de CHAVE LIGA/DESLIGA, mantendo ou não determinados padrões iniciais de funcionamento do dBASE.

Alguns desse padrões já foram descritos. Outros o serão à frente.

Para examiná-los todos e verificar em que condição se encontram, isto é, se ligados ou desligados - em ON ou em OFF - usa-se o comando DISPLAY STATUS (ou LIST STATUS) que, também, fornece outras informações sobre a situação atual do processamento, nome de arquivo em uso e nomes de arquivos de índice abertos, com os nomes dos campos chaves respectivos, disk drive default etc. Sua sintaxe é:

DISPLAY STATUS

Vamos experimentá-lo agora:

```
D          DISPLAY STATUS <RETURN>
```

Se não houver nenhum arquivo em uso, o dBASE exibirá o seguinte quadro no vídeo:

Data de Hoje	-	16/07/87		
Drive 'default'	-	A:		
ALTERNATE	-	OFF	BELL	- ON
CARRY	-	OFF	COLON	- ON
CONFIRM	-	OFF	CONSOLE	- ON
DEBUG	-	OFF	DELETE	- OFF
ECHO	-	OFF	EJECT	- ON
ESCAPE	-	ON	EXACT	- OFF
INTENSITY	-	ON	LINKAGE	- OFF
PRINT	-	OFF	RAW	- OFF
STEP	-	OFF	TALK	- ON

Se houver algum arquivo em uso no momento de execução do comando, o dBASE informará primeiramente o seu nome, seu tipo e condição de uso e emitirá a mensagem:

Esperando.

significando que uma tecla qualquer deva ser pressionada, a fim de que sejam informadas as condições funcionais mencionadas (quadro acima).

9.1 - ACENTUAÇÃO GRÁFICA DE DADOS NO dBASE

Vimos anteriormente que os nomes de arquivos e de campos de registros no dBASE não devem ser acentuados graficamente nem usar c com cedilha, sob pena de serem rejeitados.

*** No dBASE II PLUS, os conteúdos de campos de registros (dados) podem ser acentuados e usar c com cedilha.

Todavia, cumpre ressaltar dois pontos importantes a respeito:

1. Dados acentuados graficamente e com c com cedilha serão exibidos normalmente no vídeo.
2. Os mesmos dados, quando enviados para a impressora, SERÃO IM-

PRESSOS COM TRUNCAMENTO OU TROCA DOS CARACTERES ACENTUADOS.

Esse é um problema que ocorre praticamente com todos os programas aplicativos importados de países cujos idiomas não utilizam acentuação gráfica e c com cedilha, diferentemente do que ocorre com a nossa língua portuguesa.

Essa é a razão de não estarmos usando acentuação gráfica nos dados de nosso arquivo PESSOAL.

10 - PESQUISANDO

INFORMAÇÕES NO ARQUIVO

Depois de termos um arquivo pronto, isto é, um arquivo em que haja dados armazenados, podemos obter informações de seus registros.

DISPLAY e LIST são dois dos comandos utilizados para esse fim.

DIFERENÇA BÁSICA ENTRE DISPLAY E LIST

A diferença básica entre ambos é que, em arquivos com mais do que quinze registros, DISPLAY pára a exibição a cada quantidade igual exposta, aguardando qualquer tecla ser pressionada para prosseguir, enquanto que LIST não conta com esse recurso, exibindo os dados ininterruptamente.

Todavia, a exibição de dados por ambos os comandos pode ser interrompida temporariamente e novamente acionada, pressionando-se simultaneamente as teclas CONTROL e S.

Por outro lado, pressionando-se a tecla ESC, cancela-se a execução de qualquer comando do dBASE.

Vamos usar nosso arquivo PESSOAL para as experiências que realizaremos a seguir.

Começemos por "limpar" a tela de vídeo:

Comando ERASE

*** ERASE elimina tudo o que estiver em exibição na tela de vídeo e posiciona o cursor no canto esquerdo superior da mesma. Sua sintaxe é, simplesmente:

ERASE

Estando o dBASE ativado, comandar:

USE PESSOAL <RETURN>

Digitar em seguida:

DISPLAY ALL <RETURN>

O dBASE exibirá no vídeo:

00001	Marco Antonio	Desenhista	23000	07/77
00002	Luis Carlos	Contador	21000	12/78
00003	Maria Cecilia	Pedagoga	20000	06/81
00004	Maria Beatris	Psicologa	19000	10/82
00005	Julio Cesar	Gerente	30000	04/87
00006	Paulo Sergio	Engenheiro	25000	09/82

NOTA: Doravante, a fim de agilizar nossas experiências, não escreveremos mais por extenso a indicação de digitar comandos, a não ser que seja absolutamente necessária, para clareza do texto.

Em vez de "digitar" ou "digitar em seguida" etc., será usada apenas a letra D na margem esquerda da linha em que for indicado o comando, que será sempre seguido de <RETURN>.

Quanto à resposta dada ao comando pelo dBASE através do vídeo ou da impressora, continuará a ser transcrita no texto do livro logo após a citação do comando em discussão, com salto de apenas uma linha e sem a costumeira e redundante frase "o dBASE exibirá no vídeo:".

Em ambos os casos continuarão a ser usados os mesmos caracteres enfatizados de impressão que têm sido empregados até esta parte.

Comandos SET PRINT ON/OFF e CONTROL/P

*** SET PRINT ON/OFF liga ou desliga logicamente a impressora às operações do dBASE. Sua sintaxe é:

SET PRINT ON ou OFF

*** CONTROL/P é uma função de controle por caractere do MSX DOS válida quando o dBASE está atuando. Ela ativa o sistema "eco" do DOS, ligando a impressora logicamente ao sistema e gerando a impressão em papel de tudo o que é digitado no console e exibido no vídeo.

*** Tendo-se uma impressora acoplada ao microcomputador, os dados obtidos do arquivo pelo comando DISPLAY podem ser impressos em papel, no mesmo formato em que são exibidos no vídeo, sem a limitação de 40 colunas que ocorre quando o conjunto opera sem cartucho apropriado para expansão.

Para tanto, basta ligar antes a fonte de alimentação da impressora e, em seguida, ligá-la logicamente ao sistema, o que pode ser feito através de duas maneiras:

1. Pressionando-se simultaneamente as teclas
CONTROL e P.

2. Comandando através do dBASE:

SET PRINT ON <RETURN>

Para desligar logicamente a impressora do sistema:

1. Pressionar simultaneamente as teclas
CONTROL e P.

2. Comandar através do dBASE:

SET PRINT OFF <RETURN>

ATENÇÃO! Deve-se cuidar de não tentar ligar logicamente a impressora se a mesma não estiver fisicamente conectada ao microcompu-

tador, sob pena de travar o sistema e ter que reiniciá-lo, com perda eventual de dados digitados e ainda não transferidos para o disquete.

*** Pode-se comandar a exibição de dados de um registro determinado:

```
D          DISPLAY RECORD 5  <RETURN>
          00005  Julio Cesar      Gerente      30000  04/87
```

*** Pode-se também comandar a exibição de dados selecionados de apenas um registro, digitando-se antes o seu número:

```
D          3  <RETURN>
D          DISPLAY SALARIO NOME  <RETURN>
          00003  20000  Maria Cecilia
```

Ou, de outra forma:

```
D          DISPLAY RECORD 4 NOME CARGO  <RETURN>
          00004  Maria Beatris  Psicologa
```

*** Os comandos DISPLAY ALL e LIST iniciam sua ação sempre a partir do registro número 1.

Para se iniciar a exibição de registros ou de seus dados a partir de um número determinado, deve-se antes digitar esse número e teclar RETURN e, em seguida, dar o comando LIST ou DISPLAY com complementação de NEXT seguido do número que corresponda à quantidade de registros a serem exibidos.

```
D          4  <RETURN>
D          DISPLAY NEXT 2 NOME SALARIO  <RETURN>
          00004  Maria Beatris      19000
          00005  Julio Cesar        30000
```

*** Para se obter a exibição de todos os registros, ou de seus dados,

a partir de um número determinado e sem saber a quantidade de registros existentes no arquivo, adota-se procedimento igual, atribuindo a NEXT um número bem alto (até 65535):

```
D          5  <RETURN>
D          LIST NEXT 10000 NOME CARGO  <RETURN>

          00005  Julio Cesar      Gerente
          00006  Paulo Sergio    Engenheiro
```

PONTEIRO DE REGISTROS

*** A fim de localizar os registros num arquivo, o dBASE utiliza um indicador cujo controle permite fazê-lo apontar para o(s) registro(s) a ser(em) acessado(s). É o PONTEIRO DE REGISTROS (POINTER) ou simplesmente PONTEIRO.

Como vimos acima, desloca-se o ponteiro de registros para um registro determinado, digitando-se apenas o seu número.

Pode-se também deslocar o ponteiro de registros através dos comandos GOTO e SKIP:

Comando GO ou GOTO

*** GO desloca o ponteiro de registros para o registro indicado. Sua sintaxe é:

```
GO [<número de registro>] [BOTTOM] [TOP]
```

```
D          GO 4  <RETURN>
D          DISPLAY NOME SALARIO OFF  <RETURN>
```

```
          Maria Beatris    19000
```

Comandando-se:

```
D          GO TOP  <RETURN>
```

desloca-se o ponteiro para o registro do topo (início) do arquivo.

Comandando-se:

```
D          GO BOTTOM  <RETURN>
```

desloca-se o ponteiro para o fim do arquivo.

Função # (REGISTRO ATUAL)

*** A função # indica a posição atual do ponteiro de registros, sendo sua sintaxe:

#

Exemplo:

```
? #  <RETURN>
```

3

(No momento, o ponteiro está apontando para o registro número 3.)

? - Comando de IMPRESSÃO

No exemplo acima usamos o caractere ? não como um sinal de interrogação, mas sim como um comando. Em analogia com o MSX BASIC, também no dBASE o sinal de interrogação assume a condição do comando PRINT quando antecede uma expressão e é complementado pela função RETURN.

O produto do comando ? é sempre exibido no início da linha seguinte à posição do cursor no momento do comando.

Quando usado em duplicata (??), o produto de sua execução é exibido no início da linha em que se encontra o cursor.

Sua sintaxe é:

? ou ?? (expressão)

Comando SKIP

*** SKIP desloca o ponteiro de registros em relação à sua última posição. Sua sintaxe é:

SKIP [+ou-] [<quantidade de registros>]

D ? # <RETURN>

3

D SKIP <RETURN>

Registro =00004

São válidos comandos como

SKIP+n ou SKIP-n

em que n é um número qualquer entre 1 e 64534.

*** Para se obter exibição de dados de registros com os números destes deslocados para outra posição, usar no local desejado o símbolo ‡ entre vírgulas:

D 1 <RETURN>

D DISPLAY SALARIO,NOME,‡,OFF <RETURN>

23000 Marco Antonio 1

*** Pode-se obter apenas a lista de salários e números dos registros correspondentes na segunda coluna:

D LIST ALL SALARIO,‡,OFF <RETURN>

23000 1

21000	2
20000	3
19000	4
30000	5
25000	6

Ou, inversamente:

D DISPLAY ALL SALARIO <RETURN>

00001	23000
00002	21000
00003	20000
00004	19000
00005	30000
00006	25000

10.2 - PESQUISAS DIVERSAS EM CAMPOS "C"

*** As pesquisas em campos "C" devem ser feitas com o dado a ser pesquisado indicado entre aspas.

D DISPLAY FOR CARGO = "Gerente" <RETURN>

00005	Julio Cesar	Gerente	30000	04/87
-------	-------------	---------	-------	-------

Experimentar agora:

D DISPLAY FOR CARGO = "gerente". <RETURN>

D DISPLAY FOR CARGO = "GERENTE" <RETURN>

*** Pesquisa comandada com digitação em forma diferente da que consta no registro redonda infrutífera e o dBASE retorna apenas seu ponto de prontidão.

Notar que os dados indicados para as pesquisas têm formatos diferentes do dado correspondente contido no registro número 5, que tem a

letra inicial maiúscula, sendo as demais minúsculas (Gerente).

Tal situação pode ser contornada através da

FUNÇÃO ! (LETRAS MAIÚSCULAS)

*** A função ! (letras maiúsculas) converte as letras de expressões ou de conteúdos de campos em LETRAS MAIÚSCULAS. É representada pelo sinal de exclamação (!) e sua sintaxe é:

!(expressão ou campo)

*** Para pesquisar um dado cujo formato de caracteres desconhecemos, utiliza-se a FUNÇÃO "LETRAS MAIÚSCULAS".

```
D          DISPLAY FOR !(CARGO) =
                                "GERENTE"  (RETURN)
00005  Julio Cesar      Gerente      30000 04/87
```

Em tal caso, o dBASE converte os conteúdos dos campos indicados em letras maiúsculas apenas para efeito de comparação com o dado indicado no comando, devendo este, portanto, ser digitado em letras maiúsculas, como no exemplo citado.

Notar que o campo indicado no comando está entre parênteses e antecedido pelo símbolo da função.

Para alterar definitivamente para o formato de letras maiúsculas os conteúdos dos registros, é utilizado o comando REPLACE juntamente com a função LETRAS MAIÚSCULAS, como pode ser visto no item 11.1.

10.3 - BUSCAS DE DADOS ACENTUADOS GRAFICAMENTE

Vimos no item 9.1 que os conteúdos de campos (dados) com acentos gráficos e com cedilha são impressos com truncamento ou troca de caracteres quando liberados para a impressora, embora sejam exibidos sem problemas no vídeo.

Com relação à busca ou pesquisa de dados em tais condições, cumpre-nos ressaltar os seguintes pontos:

1. Buscas ou pesquisas de dados com acentuação gráfica e cedilha são efetuadas normalmente pelo dBASE, DESDE QUE O DADO INDICADO E O DADO PESQUISADO COINCIDAM EXATAMENTE.
2. A função "LETRAS MAIUSCULAS", antes descrita, não é válida para tais dados, isto é, se o dado constante do registro tiver acento(s) gráfico(s) e o indicado para pesquisa tiver ou não, a função não compensa a diferença e a busca redundante nula.
3. Dados acentuados existentes e localizados como tais através de indicações precisas serão exibidos normalmente no vídeo, mas impressos com truncamento ou troca de caracteres pela impressora, com sensível prejuízo da parte estética.

10.4 - PESQUISAS DE DADOS

COM VERIFICAÇÕES EXATAS

Se comandarmos uma busca de dados do tipo:

```
DISPLAY FOR !(NOME) = "JULIO"
```

e houver no arquivo diversos registros com o campo NOME com nomes como Julio Cesar, Julio Campos, Julio Santos etc., todos eles serão exibidos. Se, por exemplo, houver 200 nomes nessas condições, todos serão indistintamente apresentados.

Pode ocorrer, entretanto, que se queira obter informações sobre um único e determinado nome, como, por exemplo, Julio C. S. Sarica. Em tal caso, deverá ser feita no arquivo uma busca que compare com exatidão o dado indicado para pesquisa com os existentes no arquivo e apresente aquele cuja igualdade é absoluta, se existir. Isto é feito através do comando apresentado a seguir:

Comando SET EXACT ON/OFF

*** O comando SET EXACT ON aciona a modalidade de busca de dados pelo método de comparações exatas entre o dado indicado e o dado pesquisado, apresentando apenas o dado que satisfaça plenamente tal condição. Sua sintaxe é:

SET EXACT ON ou OFF

Ao ser ativado o dBASE, SET EXACT está normalmente em OFF.

10.5 - PESQUISAS DIVERSAS EM CAMPOS "N"

*** As pesquisas em campos "N" devem ser feitas com indicação sem aspas do dado a ser pesquisado.

D DISPLAY FOR SALARIO = 19000 <RETURN>

00004 Maria Beatris Psicologa 19000 10/82

D DISPLAY FOR SALARIO >= 25000 <RETURN>

00005 Julio Cesar Gerente 30000 04/87

00006 Paulo Sergio Engenheiro 25000 09/82

*** O dBASE efetua cálculos com os dados de campos "N".

D DISPLAY ALL NOME SALARIO * 1.2 <RETURN>

00001 Marco Antonio 27600.0

00002 Luis Carlos 25200.0

00003 Maria Cecilia 24000.0

00004 Maria Beatris 22800.0

00005 Julio Cesar 36000.0

00006 Paulo Sergio 30000.0

Caso não se queira que nos produtos das operações de cálculos sejam exibidas frações decimais, usa-se a função STR.

Função STR

*** A função STR transforma números ou variáveis numéricas em simples cadeias de caracteres ou "strings". Sua sintaxe é:

```
STR ((expressão numérica, comprimento,  
      quantidade de casas decimais))
```

```
D      DISPLAY FOR SALARIO (<= 21000 NOME SALARIO  
      STR (SALARIO * 1.33456,8,2) <RETURN>
```

```
00002 Luis Carlos      21000 28025.76  
00003 Maria Cecilia    20000 26691.20  
00004 Maria Beatris    19000 25356.64
```

No exemplo acima fez-se uma projeção de aumento de 33,456% sobre salários inferiores ou iguais a 21000. As frações decimais não foram desprezadas em vista de o comando especificar a conversão baseada num produto de 8 dígitos, incluindo o ponto, sendo 2 decimais, como se deduz facilmente, observando o comando.

```
D      DISPLAY FOR SALARIO > 21000 NOME SALARIO  
      STR (SALARIO * 1.25678,5,0) <RETURN>
```

```
00001 Marco Antonio    23000 28905  
00005 Julio Cesar      30000 37703  
00006 Paulo Sergio     25000 31419
```

Neste exemplo, a projeção de aumento é de 25,678% sobre salários maiores do que 21000, sendo desprezadas as frações decimais em vista de o comando especificar a conversão baseada num produto de 5 dígitos, sem decimais e, conseqüentemente, sem ponto (observar o comando).

Função \$ (SUBSTRING)

*** A função \$ extrai uma parte específica de uma "string" (cadeia de caracteres alfanuméricos). Sua sintaxe é:

\$((string, posição inicial,
quantidade de caracteres))

D DISPLAY ALL \$(NOME,1,3) <RETURN>

```
00001  Mar
00002  Lui
00003  Mar
00004  Mar
00005  Jul
00006  Pau
```

A função \$ corresponde à função MID\$ do BASIC MSX e oferece inúmeras possibilidades de uso. Novos exemplos de seu uso serão apresentados com outros comandos.

Comandos LOCATE e CONTINUE

Éis um poderoso comando do dBASE para pesquisa ou busca de dados.

*** Através de LOCATE e seu comando complementar CONTINUE pode-se localizar dados semelhantes existentes em mais de um registro. Sintaxe:

LOCATE [(escopo)] FOR (condição)

Se não for limitado pelo escopo, LOCATE efetuará busca em todo o arquivo, através de CONTINUE.

Se o dado pesquisado não existir no arquivo, ao ser atingido seu fim, será exibida no vídeo a mensagem:

Fim de arquivo encontrado.

Vamos testá-lo:

D LOCATE FOR CARGO = "Contador" <RETURN>

Ao localizar o dado procurado, o dBASE apenas indica o número do

Veremos agora como ser informados rapidamente sobre a quantidade de registros existentes num arquivo e como somar os conteúdos de campos numéricos do mesmo.

Comando COUNT

*** COUNT conta e informa o número total de registros do arquivo ou o número de registros que satisfaçam uma condição especificada. O resultado pode também ser atribuído a uma variável de memória do dBASE. Sua sintaxe é:

```
COUNT [<escopo>] [FOR <condição>]
      [TO<variável>]
```

```
D          COUNT  <RETURN>
```

```
Contagem =00006
```

```
D          COUNT FOR !(CARGO) =
                        "DESENHISTA"  <RETURN>
```

```
Contagem =00001
```

```
D          COUNT FOR SALARIO
                        < 21000 TO FOLHA  <RETURN>
```

```
Contagem =00002
```

(O comando acima significa para o dBASE: "conte os registros cujos salários sejam inferiores a 21000 e guarde o resultado na variável de memória FOLHA".)

Ao executar o comando, o dBASE cria a variável FOLHA para armazenar nela o resultado da contagem que efetuar. O conteúdo da variável pode ser exibido através do comando ?:

```
D          ? FOLHA  <RETURN>
```

NOTA: O assunto referente a variáveis de memória do dBASE é tratado no Capítulo 22.

Comando SUM

*** SUM soma os conteúdos de um ou mais campos dos registros, podendo somar todos ou apenas parte. Sua sintaxe é:

```
SUM [ <escopo> ] <nome(s) de campo(s)>
      [ TO <variável(is)> ] [ FOR <condição> ]
```

```
D          SUM SALARIO    <RETURN>
```

```
138000
```

```
D          SUM SALARIO * 1.3  <RETURN>
```

```
179400.00
```

```
D          SUM SALARIO * 1.43 FOR SALARIO
          <= 21000 TO AUMENTO1  <RETURN>
```

```
85800.00
```

```
D          SUM SALARIO * 1.37 FOR SALARIO
          > 21000 TO AUMENTO2  <RETURN>
```

```
106860.00
```

```
D          ? AUMENTO1, AUMENTO2  <RETURN>
```

```
85800.00  106860.00
```

Pode-se obter a soma dos conteúdos das duas variáveis de memória digitando-se:

192660.00

(Consultar o Capítulo 22 sobre variáveis de memória.)

10.6 - PESQUISA DE DADOS ORDENADOS

É comum ter-se que pesquisar ou extrair dados de um arquivo segundo uma determinada ordem - alfabética, numérica, cronológica etc.

Todavia, de maneira quase que geral, os registros são inseridos aleatoriamente nos arquivos magnéticos, por diversas razões, mas principalmente porque têm que ser introduzidos à medida que vão surgindo. Seria impraticável tentar tê-los todos de uma só vez e encaixá-los já ordenados no arquivo.

Face a tal situação, para obtermos informações do arquivo sob uma determinada ordem, o mesmo tem que ser previamente ordenado e, eventualmente, mantido nessa condição.

Por outro lado, será pouco útil - a não ser em casos especiais - manter um arquivo ordenado de modo convencional, alfabeticamente e por um único campo, NOME, por exemplo, sabendo-se que o mesmo terá que ser pesquisado ou fornecer informações de outros campos, ordenados numérica ou cronologicamente ou por outra ordem.

Acrescem-se a tais fatos outros pontos importantes:

1. O processo de ordenação é lento, principalmente em arquivos de elevado número de registros.
2. Não é possível ordenar um arquivo em função de mais de um campo ou chave ao mesmo tempo.
3. Não é possível ordenar apenas registros que satisfaçam a uma condição determinada.
4. Manter diversas ordenações de um mesmo arquivo apresenta inconvenientes diversos: dificuldade de manutenção (correções,

atualizações, deleções etc.) e consumo excessivo de espaço em disquetes, pois cada modalidade de ordenação gera um novo arquivo que ocupa espaço equivalente ao do arquivo principal.

Por tais razões, a solução mais prática que se impõe com relação à pesquisa e extração de dados ordenados do arquivo é a modalidade de

INDEXAÇÃO DE ARQUIVOS,

que apresenta as seguintes vantagens sobre a ordenação convencional:

1. Maior rapidez de processamento.
2. Possibilidade de indexação por mais de um campo ou chave concomitantemente.
3. Economia de espaço em disquetes.

Em suas versões comuns, o dBASE inclui dois sistemas de ordenação de arquivos:

1. Ordenação convencional (direta ou plena) - comando SORT.
2. Indexação - comando INDEX.

Na versão dBASE II PLUS, o comando SORT não foi incluído.

10.6.1 - INDEXAÇÃO DE ARQUIVO

O processo de indexação de um arquivo consiste em criar um índice para um determinado campo, em ordem crescente.

Ao INDEXAR um campo de um arquivo de dados, o dBASE cria automaticamente outro arquivo, denominado ARQUIVO DE ÍNDICE ou ARQUIVO-ÍNDICE, o qual mantém em ordem crescente todos os dados do campo especificado, sendo cada dado acompanhado de um indicador do lugar ocupado por ele no arquivo principal. Os arquivos de índice recebem automaticamente a extensão .NDX, se outra não for especificada pelo operador.

Um arquivo de dados pode ser indexado por qualquer de seus campos

e dispor paralelamente de tantos ARQUIVOS DE ÍNDICE quantos forem necessários, embora seja possível usar conjuntamente apenas um por vez.

Ao ser indexado, o arquivo de dados não sofre nenhuma alteração na ordem em que foi constituído, mas o processo de pesquisa ou extração de dados do mesmo passa a ser incomparavelmente mais rápido.

Suponha-se, como exemplo, que se queira obter informações sobre determinado cliente constante de arquivo ainda não indexado. Utiliza-se o comando DISPLAY e o seguinte processo se desenvolve no microcomputador:

Cada registro do arquivo, a partir do número um, é lido no disquete e colocado na memória do computador inteiramente. Aí, o dado indicado no comando de pesquisa é comparado com o dado existente no campo correspondente. A pesquisa se estende desse modo, até que ambos os dados coincidam exatamente ou até que o fim do arquivo seja atingido.

Assim, se os registros forem extensos e o arquivo muito grande, a pesquisa poderá consumir longos e cansativos minutos.

Estando o arquivo indexado, a mesma pesquisa será efetuada em alguns segundos apenas, através do comando FIND, porque, em vez de processar todo o arquivo, apenas o índice do campo específico é pesquisado e, ao ser localizado o dado procurado, o sistema capta no mesmo o número do registro que lhe corresponde no arquivo principal e passa a apontar para ele, bastando nesse ponto dar o comando DISPLAY para que seus dados sejam exibidos.

10.6.2 - CRIAÇÃO DE ARQUIVO DE ÍNDICE

Comando INDEX ON

*** INDEX ON coleta os conteúdos de um campo especificado de arquivo de dados, indexa-os em ordem crescente e cria para os mesmos um arquivo próprio, onde os armazena. Sua sintaxe é:

```
INDEX ON <campo 1 + campo 2 + ...>
```

TO <nome do arquivo de índice>

Ao ser executado o comando, é criado e automaticamente gravado no disquete o arquivo de índice com o nome especificado no comando. Não sendo indicada uma extensão para o nome do arquivo, o dBASE acrescenta automaticamente a específica do tipo de arquivo, que é, no caso: .NDX.

Vamos criar agora um índice para nosso arquivo PESSOAL e examinar outros pontos relativos à indexação de arquivos e seus recursos.

Depois de ativado o dBASE, comandar:

```
D          USE PESSOAL    <RETURN>
D          INDEX ON NOME TO PESSOAL    <RETURN>
```

Após alguns instantes, surgirá no vídeo a mensagem:

```
00006 Registros indexados.
```

Certifiquemo-nos de que foi gravado no disquete um ARQUIVO DE ÍNDICE com o nome PESSOAL.NDX:

```
D          DISPLAY FILES LIKE *.*    <RETURN>
```

Além de outros e do próprio PESSOAL.DBF, notaremos o

```
PESSOAL.NDX
```

Ao ser criado, o arquivo de índice entra automaticamente em uso, senão vejamos:

```
D          DISPLAY ALL NOME    <RETURN>
```

```
00005 Julio Cesar
00002 Luis Carlos
00001 Marco Antonio
00004 Maria Beatris
00005 Maria Cecilia
00006 Paulo Sergio
```

Notar que os registros foram exibidos em ORDEM ALFABÉTICA e não pela ordem de entrada no arquivo.

*** Quando um arquivo de dados está em uso juntamente com um arquivo de índice, os dados do mesmo são processados pelo dBASE segundo a ordem de indexação.

Vamos agora abandonar o arquivo de índice e voltar ao principal, comandando a exibição dos nomes:

```
D          USE PESSOAL  <RETURN>
D          DISPLAY ALL NOME  <RETURN>
```

```
00001  Marco Antonio
00002  Luis Carlos
00003  Maria Cecilia
00004  Maria Beatris
00005  Julio Cesar
00006  Paulo Sergio
```

Notar que, agora, os registros foram exibidos pela ordem em que foram inseridos no arquivo.

*** Quando se retorna ao uso do arquivo principal ou se passa para outro arquivo (de dados ou de índice), o arquivo de índice em uso é automaticamente fechado.

Vimos que, ao ser criado, um arquivo de índice é concomitantemente aberto, entrando automaticamente em uso. Vamos ver agora como colocá-lo em uso quando ele está fechado.

Comando SET INDEX TO

*** SET INDEX TO ativa arquivo(s) de índice quando o arquivo principal correspondente estiver em uso. Sua sintaxe é:

```
SET INDEX TO <nome arq índice 1,
```

nome arq índice 2, ...>

Pode-se também ativar um arquivo de índice juntamente com a ativação do arquivo principal a que pertencer:

Comando INDEX

*** Quando utilizado em conjunto com o comando USE, o comando INDEX ativa um ou mais arquivos de índice.

Sua sintaxe de conjunto é:

```
USE <nome arq dados> INDEX <nome arq índice 1,  
nome arq índice 2,nome arq índice 3, ...>
```

Estávamos usando somente o arquivo PESSOAL.DBF. Ativemos o arquivo de índice PESSOAL.NDX:

```
D          SET INDEX TO PESSOAL  <RETURN>
```

(Lembremos que as extensões de nomes de arquivo próprias do dBASE não precisam ser indicadas no comando, mesmo quando o nome principal é o mesmo.)

Se comandarmos agora:

```
D          LIST ALL NOME  <RETURN>
```

comprovaremos que o arquivo de índice está ativado, porque os nomes serão listados em ordem alfabética e não pela ordem de registros.

A título de experiência, desativemos ambos os arquivos para, em seguida, colocá-los novamente em uso através de um único comando:

```
USE  <RETURN>
```

Os arquivos PESSOAL.DBF e PESSOAL.NDX foram fechados.

```
D          USE PESSOAL INDEX. PESSOAL  <RETURN>
D          DISPLAY ALL NOME    <RETURN>
```

```
00005  Julio Cesar
00002  Luis Carlos
00001  Marco Antonio
00004  Maria Beatris
00005  Maria Cecilia
00006  Paulo Sergio
```

Pesquisas em arquivos de índice são efetuadas com rapidez através do comando descrito a seguir.

Comando FIND

*** FIND efetua pesquisas em arquivos de índice até localizar um dado especificado ou atingir o fim do arquivo. Sua sintaxe é:

```
FIND <conteúdo do campo>
```

FIND só é válido para arquivos de índice.

Ao localizar o dado especificado no comando, a pesquisa simplesmente cessa, retornando ao vídeo o ponto do dBASE, sendo então necessário utilizar o comando DISPLAY para exibição dos dados do registro:

```
D          FIND Maria  <RETURN>
```

```
D          DISPLAY  <RETURN>
```

```
00004  Maria Beatris  Psicologa  19000  10/82
```

```
D          FIND Maria Cecilia  <RETURN>
```

```
D          DISPLAY  <RETURN>
```

Quando o dado especificado não é encontrado, o dBASE retorna uma mensagem:

Não encontrado.

```
D          FIND MARCO ANTONIO  <RETURN>
```

Não encontrado.

Notar que o formato dos caracteres do nome indicado no comando de pesquisa é diferente do que consta no registro, tendo a busca redundado nula por essa razão.

*** A função "LETRAS MAIÚSCULAS" não é válida para arquivo indexado.

Se não houver coincidência entre os formatos dos dados indicados para pesquisa e os existentes no arquivo, a pesquisa não será bem sucedida.

10.6.3 - INDEXAÇÃO POR VÁRIOS CAMPOS OU CHAVES

É possível indexar um arquivo por vários campos ao mesmo tempo. A indexação múltipla, porém, só funciona em campos "C". Campos numéricos devem ser convertidos previamente para campos de caracteres através da função STR, para poderem ser indexados nessa condição. Exemplo:

```
D          INDEX ON NOME + STR(SALARIO,5)
                                     TO NOMESAL  <RETURN>
```

00006 Registros indexados.

Nessa condição, o arquivo PESSOAL.DBF é indexado pelo campo NOME, sendo os nomes iguais ordenados por ordem crescente de salários.

10.6.4 - REINDEXAÇÃO DE ARQUIVOS DE ÍNDICE

Quando um arquivo de dados sofre qualquer alteração - atualização de dados, novas inserções, eliminação de registros etc. - seus arquivos de índice devem ser reindexados.

*** Se, durante o uso do arquivo de dados, estiverem abertos até sete arquivos de índice a ele pertencentes, serão os mesmos atualizados ou reindexados automaticamente quando forem feitas alterações nos registros.

Pondera-se, todavia, que, quanto maior for a quantidade de arquivos de índice a serem reindexados nessas condições, menor será a velocidade de processamento geral dos dados, podendo tornar-se sensivelmente demorado se o arquivo tiver elevado número de registros.

Se existirem mais do que sete arquivos de índice para um arquivo de dados, os demais deverão ser abertos posteriormente, em outro lote, e reindexados pelo comando REINDEX, devendo ser feito o mesmo com arquivos eventualmente não incluídos para atualização por esquecimento.

Comando REINDEX

*** REINDEX reindexa ou atualiza arquivos de índice que estejam abertos ou em uso. Sua sintaxe é, apenas:

REINDEX

*** Até sete arquivos de índice podem ser atualizados de uma só vez.

Para que os arquivos de índice sejam atualizados, é preciso que estejam abertos. Pode-se abri-los ao mesmo tempo que se abre o arquivo de dados, através do comando:

USE PESSOAL INDEX ARQIN1, ARQIN2, ...

ou, se o arquivo de dados já estiver aberto, com o comando:

SET INDEX TO ARQIN1, ARQIN2, ... <RETURN>

em que ARQINnOs representam os nomes dos arquivos de índice.

*** Até sete arquivos de índice podem estar abertos ao mesmo tempo, embora um por vez apenas possa ser usado.

*** Quando diversos arquivos de índice são abertos ao mesmo tempo, o que figurar em primeiro lugar no comando é que estará automaticamente em uso com o arquivo principal, sendo os dados processados por ordem crescente dos conteúdos do campo indexado por ele.

*** Troca-se o arquivo de índice em uso através do comando SET INDEX TO, especificando-se no mesmo o nome do arquivo a ser usado.

Depois de abertos os arquivos, comanda-se:

REINDEX <RETURN>

A medida que o comando é executado, o operador toma conhecimento através de mensagens exibidas no vídeo, para cada arquivo indexado:

Reindexando o arquivo - A=ARQINnO.NDX
0000n Registros indexados.

10.6.5 - USANDO ALTERNATIVAMENTE

DIVERSOS ARQUIVOS DE ÍNDICE

Já vimos que apenas um arquivo de índice pode ser usado por vez com o arquivo de dados, embora possam estar abertos até sete ao mesmo tempo e existir quaisquer quantidades deles.

A título de experiência, vamos agora indexar o arquivo PESSOAL por todos os seus campos e usá-los alternativamente.

Vamos ver também que, no formato em que foram inseridos no arquivo, os dados do campo DATADM não se prestam a uma ordenação perfeita, sendo conveniente modificá-los adequadamente para tal fim, o que nos

obrigará a reindexar os arquivos de índice existentes.

Iniciaremos nossa prática com o arquivo PESSOAL fechado e já indexado pelo campo NOME:

```
D          USE PESSOAL    <RETURN>
D          INDEX ON CARGO TO CARGO    <RETURN>
```

00006 Registros indexados.

```
D          INDEX ON SALARIO TO SALARIO    <RETURN>
```

00006 Registros indexados.

```
D          INDEX ON DATADM TO DATA    <RETURN>
```

00006 Registros indexados.

```
D          DISPLAY FILES LIKE *.*    <RETURN>
```

```
...
PESSOAL .DBF          PESSOAL .NDX
CARGO   .NDX          SALARIO .NDX
DATA   .NDX
```

```
D          SET INDEX TO CARGO    <RETURN>
D          LIST ALL NOME CARGO    <RETURN>
```

```
00002 Luis Carlos      Contador
00001 Marco Antonio    Desenhista
00006 Paulo Sergio     Engenheiro
00005 Julio Cesar      Gerente
00003 Maria Cecilia    Pedagoga
00004 Maria Beatris    Psicologa
```

Notar que os registros foram apresentados seguindo a ordem alfabética dos cargos.

```
D          SET INDEX TO SALARIO    <RETURN>
```

D DISPLAY ALL NOME,#,SALARIO,OFF <RETURN>

Maria Beatris	4	19000
Maria Cecilia	3	20000
Luis Carlos	2	21000
Marco Antonio	1	23000
Paulo Sergio	6	25000
Julio Cesar	5	30000

Notar que os registros foram exibidos seguindo a ordem crescente dos salários.

D SET INDEX TO DATA <RETURN>

D LIST ALL NOME DATADM OFF <RETURN>

Julio Cesar	04/87
Maria Cecilia	06/81
Marco Antonio	07/77
Paulo Sergio	09/82
Maria Beatris	10/82
Luis Carlos	12/78

Observar que os registros foram mostrados seguindo a ordem crescente APENAS dos MESES, não sendo, portanto, ORDEM CRONOLÓGICA PERFEITA, já que os dados englobam também ANOS.

*** Para que a ordenação cronológica no arquivo seja exata, as datas devem ser digitadas na ordem ANO/MES/DIA, ou apenas ANO/MES.

Assim, temos que modificar o formato das datas nos registros. Para fazê-lo, utilizaremos o comando BROWSE, que será explicado em detalhes mais à frente. Por ora utilizá-lo-emos apenas para alterar as datas do arquivo PESSOAL, a fim de que seus arquivos de índice sejam reindexados.

Vamos fazê-lo pelo modo mais racional: abrimos o arquivo PESSOAL e todos os seus arquivos de índice juntamente. Em seguida, alteramos as datas do arquivo principal. Como todos os seus arquivos de índice estão abertos, eles serão automaticamente reindexados.

```
D          USE PESSOAL INDEX PESSOAL,CARGO,  
          SALARIO,DATA <RETURN>  
D          BROWSE FIELDS NOME,DATADM <RETURN>
```

O dBASE colocará no vídeo todos os registros do arquivo, na disposição reproduzida a seguir:

```
RECORD # :00005  
NOME----- DATAD  
Julio Cesar   04/87  
Luis Carlos   12/78  
Marco Antonio 07/77  
Maria Beatris 10/82  
Maria Cecilia 08/81  
Paulo Sergio  09/82
```

O cursor estará posicionado no primeiro caractere do primeiro registro exibido no vídeo. Mantendo pressionada a tecla CONTROL e pressionando uma vez a tecla F, deslocaremos o cursor até o primeiro caractere do campo DATADM. Então, sem nos preocuparmos em apagar a data exibida, digitamos sobre ela:

```
D          87/04
```

Digitada a data no novo formato, o cursor saltará automaticamente para a data seguinte, não sendo necessário teclar RETURN. Com o cursor já na posição certa, digitamos:

```
D          78/12
```

Procedemos da mesma forma com as datas restantes. Depois de alterada a última, o cursor se manterá posicionado nela.

(Se alguma data for alterada incorretamente, o cursor poderá ser deslocado até ela com auxílio do INDICADOR VISUAL DAS TECLAS DE CONTROLE DO COMANDO BROWSE, que se encontra no ITEM 7.7.)

Findas as alterações das datas, pressionamos simultaneamente as teclas:

CONTROL e W

e as alterações serão automaticamente passadas para o arquivo de dados no disquete, sendo os arquivos de índice automaticamente reindexados.

Nesse caso, o sistema não emite mensagens sobre a execução da reindexação. Mas, é certo que ela é efetuada.

Outro modo de se proceder à reindexação dos arquivos de índice é indicado a seguir:

Abrimos o arquivo de dados e efetuamos as alterações através do comando BROWSE, como foi exemplificado pouco atrás. Em seguida, abrimos os arquivos de índice com o comando SET INDEX TO:

```
D          SET INDEX TO PESSOAL,CARGO,  
                                SALARIO,DATA  (RETURN)
```

e comandamos:

```
D          REINDEX  (RETURN)
```

A partir desse comando, a reindexação é efetuada com emissão das mensagens reproduzidas a seguir:

```
Reindexando o arquivo - A:PESSOAL .NDX  
00006 Registros indexados.
```

```
Reindexando o arquivo - A:CARGO .NDX  
00006 Registros indexados.
```

```
Reindexando o arquivo - A:SALARIO .NDX  
00006 Registros indexados.
```

```
Reindexando o arquivo - A:DATA .NDX  
00006 Registros indexados.
```

Vamos agora verificar se o arquivo de índice DATA apresenta os dados efetivamente em ORDEM CRONOLÓGICA:

D SET INDEX TO DATA <RETURN>
D LIST ALL NOME,DATADM,#,OFF <RETURN>

Marco Antonio	77/07	1
Luis Carlos	78/12	2
Maria Cecilia	81/06	3
Paulo Sergio	82/09	6
Maria Beatris	82/10	4
Julio Cesar	87/04	5

D FIND 81/06 <RETURN>

D DISPLAY <RETURN>

00006 Maria Cecilia Pedagoga 20000 81/06

Tudo certo! Passemos para outro capítulo.

11 - ALTERANDO DADOS

DO ARQUIVO

Geralmente, dados de arquivo são elementos dinâmicos que carecem de alterações ou atualizações.

Os dados de arquivos dBASE podem ser alterados diretamente através dos comandos EDIT, BROWSE, CHANGE e REPLACE e, indiretamente, por meio de um arquivo de movimento, usando o comando UPDATE. Neste capítulo são estudados os comandos do modo direto de alteração. UPDATE é estudado no Capítulo 18.

Comando EDIT

*** EDIT exibe no vídeo um registro especificado, no modo de edição, a fim de que sejam efetuadas alterações em seus dados.

Sua sintaxe é:

```
EDIT [⟨número de registro⟩]
```

Se não for especificado o número do registro no comando, o dBASE o solicitará em seguida, através da mensagem:

Entre o número do registro :

Digitado o número do registro, o seu conteúdo é exibido no vídeo, no mesmo formato gerado pelo comando APPEND, permitindo que, com auxílio das teclas de controle por caractere, sejam efetuadas alterações

no mesmo.

Todos os recursos das teclas de controle por caractere dos diversos comandos são descritos e detalhados nos INDICADORES VISUAIS DAS TECLAS DE CONTROLE, no Capítulo 7, de modo a facilitar consultas rápidas. Para o comando EDIT, funcionam as seguintes teclas:

TECLAS DE CONTROLE DO COMANDO EDIT

- CONTROL+D - Movem o cursor um espaço à direita.
- CONTROL+S - Movem o cursor um espaço à esquerda.
- CONTROL+E - Movem o cursor para o campo anterior.
- *1 CONTROL+X - Movem o cursor para o próximo campo.
- *2 CONTROL+C - Movem o cursor para o próximo registro.
- CONTROL+R - Movem o cursor para o registro anterior.
- CONTROL+G - Apagam o caractere sob o cursor.
- CONTROL+Y - Apagam o conteúdo do campo.
- CONTROL+V - Ligam/desligam o modo de inserção.
- CONTROL+U - Marcam/desmarcam o registro a deletar.
- CONTROL+Q - Encerram EDIT sem regravar o registro.
- CONTROL+W - Encerram EDIT e regravam o registro.
- RETURN - Move o cursor para o próximo campo.
- DELETE - Apaga o caractere à esquerda do cursor.

Observações:

- *1 - Se o cursor estiver no último campo do registro, será deslocado para o próximo registro. Se estiver no último campo do último registro, o modo EDIT será encerrado, retornando o ponto do dBASE.
- *2 - Se o cursor estiver no primeiro campo do registro, será deslocado para o registro anterior. Se estiver no primeiro campo do primeiro registro, o comando EDIT será encerrado, retornando ao vídeo o ponto de prontidão do dBASE para comando.

Vamos fazer duas experiências com EDIT. Na primeira, será especificado o número do registro no comando; na segunda, não será. Assim, ficará demonstrada a diferença existente entre ambos os modos. O dBASE deverá estar ativado. Comandamos:

```
D          USE PESSOAL  <RETURN>
D          EDIT 1      <RETURN>
```

O dBASE colocará no vídeo:

```
Registro 00001
NOME      :Marco Antonio  :
CARGO     :Desenhista:
SALARIO   :23000:
DATADM    :77/07:
```

O cursor estará colocado no primeiro caractere do primeiro campo.

A título de experiência, vamos alterar o dado SALARIO:

- mantendo a tecla CONTROL pressionada, teclamos duas vezes X e o cursor se deslocará para o primeiro dígito do campo SALARIO.

Vamos alterar o valor de 23000 para 25500:

- teclamos CONTROL + D e, com o cursor sobre o dígito 3, digitamos 5. O cursor pula para o dígito seguinte. Então, digitamos novamente 5.

Feita a alteração pretendida, teclamos simultaneamente

```
CONTROL e W
```

e o dBASE gravará a alteração efetuada, retornando ao modo de comando.

O registro número um está agora com o dado SALARIO alterado para 25500, senão vejamos:

```
D          DISPLAY RECORD 1 NOME SALARIO  <RETURN>

00001  Marco Antonio    25500
```

Vamos experimentar agora a outra modalidade de EDIT (sem especi-

ficar inicialmente o número de registro) para retornar o dado SALARIO do registro 1 ao seu valor normal:

D EDIT <RETURN>

Neste caso, o dBASE solicita:

Entre o número do registro :

D 1 <RETURN>

O registro será exibido na forma de edição, como mostrado acima.

Adotando o procedimento já descrito, alteramos o valor de 25500 para 23000 e, em seguida, teclamos simultaneamente:

CONTROL e W

e o dBASE regravará o registro, como fez no exemplo anterior, mas, em vez de retornar ao modo de comando direto, emitirá a seguinte mensagem através do vídeo:

Entre o número do registro :

Nessa condição, podemos ou não indicar outro registro para alteração, devendo o procedimento descrito repetir-se.

Caso se deseje apenas encerrar o modo EDIT, bastará teclar

<RETURN>

e o dBASE retornará ao modo de comando direto.

Do exposto, deduz-se que o comando EDIT seguido do número de registro é mais conveniente quando se quer alterar apenas dados de um só registro.

Para alterar dados de diversos registros, mantendo o comando de edição ativo, comanda-se EDIT inicialmente sem número de registro.

Comando SET SCREEN ON/OFF

D
D

*** SET SCREEN ON/OFF ativa ou desativa o modo FULL SCREEN (ou TELA CHEIA) do dBASE. Sua sintaxe é:

SET SCREEN ON ou OFF

Tanto em um como em outro exemplo acima, notamos que o registro foi exibido no vídeo com todos os seus campos, permitindo visualização completa dos dados e alteração de qualquer um deles, com auxílio das teclas de controle por caractere.

D

Em tal situação, isto é, podendo deslocar o cursor para qualquer ponto do registro e sendo possível mudar um ou todos os caracteres de cada campo e mesmo suprimir todo o seu conteúdo, sendo válidas essas alterações para regravação do registro, o comando EDIT está operando no modo

D

FULL SCREEN ON

que é o modo padrão - ou default - quando o dBASE é ativado.

Porém, havendo conveniência de se editar para alterações apenas campos específicos de registros, pode-se fazê-lo, comandando-se:

SET SCREEN OFF <RETURN>

antes de se comandar EDIT.

Também nesse modo, EDIT poderá ser comandado inicialmente com ou sem número do registro a alterar.

Vamos experimentar as duas versões:

D SET SCREEN OFF <RETURN>
D EDIT 6 <RETURN>

Em vez de o registro todo, o dBASE exibe:

NOME: Paulo Sergio
Trocar?

Apenas a título de experiência, colocaremos acento agudo na letra "e" (depois o removeremos). Para fazê-lo, devemos digitar a letra que desejamos alterar:

D e <RETURN>

Para :

D é <RETURN>

NOME: Paulo Sérgio
Trocar?

Se quisermos retirar agora o acento da letra "é", basta repetir a digitação citada, mas inversamente. Se quisermos manter o acento, basta teclar RETURN para encerrar o comando.

Nesse caso, o dBASE exibiu o conteúdo do campo NOME para alteração, porque não especificamos nenhum campo no comando. Quando quisermos alterar um dado específico de um registro, é preciso indicar o seu campo no comando.

Supondo que pretendemos alterar o conteúdo do campo SALARIO do registro 6, comandamos:

D EDIT 6,SALARIO <RETURN>

O dBASE responde:

SALARIO: 25000
TO

Vamos alterar esse valor para 27500:

D 27500 <RETURN>

Nessa condição, o registro será regravado com o conteúdo numérico 27500 no campo SALARIO.

Supondo-se que se edite qualquer campo e, ao examinar o seu conteúdo, decida-se não alterá-lo, bastará teclar RETURN para encerrar o comando.

A título de prática, retornemos o salário do registro número seis a seu valor normal:

```
D          EDIT 6,SALARIO  <RETURN>
```

```
SALARIO= 27500  
TO
```

```
D          25000  <RETURN>
```

Ainda no modo FULL SCREEN OFF, vamos testar EDIT sem especificar o número do registro a alterar:

```
D          EDIT  <RETURN>
```

O dBASE responderá com a seguinte mensagem:

```
Entre No. do registro, No. ou nome do  
campo, novo valor.  
COORD:
```

Experimentemos alterar o salário do registro número 4:

```
D          4,SALARIO  <RETURN>
```

```
SALARIO= 19000  
TO
```

```
D          21000  <RETURN>
```

O dBASE ainda continuará a solicitar coordenadas para alterações:

COORD:

A título de prática, retornemos a seu valor normal o conteúdo do campo SALARIO do registro número 4:

D 4,SALARIO <RETURN>

SALARIO: 21000
TO

D 19000 <RETURN>

COORD:

Nesse ponto, se não quisermos efetuar mais alterações, digitamos:

D <RETURN>

para encerrar o comando EDIT e retornar ao modo de comando direto.

Com o comando EDIT no modo FULL SCREEN OFF, há ainda outro recurso a considerar quando o campo a alterar é de caracteres (C):

D EDIT 4,CARGO <RETURN>

CARGO: PSICOLOGA
Trocar?

Nessa situação, em vez de digitar o caractere ou a palavra a alterar, pode-se saltar para a etapa seguinte, pressionando-se simultaneamente as teclas:

CONTROL e Y

e, em seguida:

<RETURN>

Surgirá no vídeo:

Para:

Digita-se então o novo conteúdo do campo e, em seguida, tecla-se apenas RETURN.

Neste caso, não vamos alterar nada. Então, teclamos RETURN.

Antes de passarmos para o próximo comando, retornemos à modalidade normal de tela:

```
D          SET SCREEN ON  (RETURN)
```

Comando BROWSE

*** BROWSE exhibe no modo de edição todos os registros do arquivo, interrompendo a exibição a cada 17 registros, a fim de que alterações possam ser efetuadas nos seus conteúdos. Sua sintaxe é:

```
BROWSE [FIELD <nome do campo,  
        nome do campo, nome do campo, ...>]
```

Se a opção FIELD não for usada, serão exibidos todos os campos.

A exibição gerada por BROWSE se inicia a partir do registro para o qual o ponteiro aponta no momento do comando. Assim, antes de se dar o comando, deve-se posicionar o ponteiro no registro desejado, digitando-o e teclando RETURN em seguida, ou através do comando GO, como exemplificado a seguir:

```
GO TOP ou GOTO 10 ou 33  (RETURN)
```

Se o arquivo tiver mais do que 17 registros, apenas essa quantidade será exibida por vez, COM TANTOS CAMPOS QUANTOS COUBEREM NA TELA.

Quando a largura total dos campos dos registros é maior do que 40 ou 80 colunas - dependendo de estar ou não instalado o cartucho de expansão de colunas do vídeo -, o comando BROWSE permite o deslizamento

horizontal dos dados no vídeo, a fim de que possam ser visualizados os campos inicialmente situados fora do mesmo.

As alterações nos dados dos registros, bem como os deslocamentos dos mesmos na tela de vídeo, nos sentidos horizontal e vertical, são feitos com auxílio das teclas de controle por caractere. (Consultar o INDICADOR VISUAL DAS TECLAS DE CONTROLE DO COMANDO BROWSE, item 7.7.) As seguintes teclas funcionam com o comando BROWSE:

TECLAS DE CONTROLE DO COMANDO BROWSE

CONTROL+B - Deslizam os dados horizontalmente para a direita, isto é, deslocam as colunas "escondidas" no lado esquerdo, colocando-as no campo visual.

CONTROL+Z - Ação invertida de CONTROL+B.

CONTROL+A - Movem o cursor para o campo à esquerda. Encontrando-se o cursor no primeiro campo, deslocam-no para o primeiro campo do registro anterior.

CONTROL+E - O mesmo que CONTROL+A.

CONTROL+F - Movem o cursor para o campo à direita. Se o cursor encontrar-se no último campo, será deslocado para o último campo do registro seguinte.

CONTROL+X - O mesmo que CONTROL+F.

CONTROL+R - Movem o cursor para o mesmo campo do registro anterior.

CONTROL+C - Movem o cursor para o mesmo campo do registro seguinte.

CONTROL+S - Movem o cursor um espaço à esquerda.

CONTROL+D - Movem o cursor um espaço à direita.

CONTROL+G - Apagam o caractere sob o cursor.

CONTROL+Y - Apagam o conteúdo do campo.

CONTROL+V - Ligam/desligam o modo de inserção.

CONTROL+U - Marcam/desmarcam registro para deleção.

*1 CONTROL+Q - Encerram o modo BROWSE regravando apenas parcialmente alterações efetuadas.

CONTROL+W - Encerram o modo BROWSE regravando todas as alterações efetuadas nos registros.

RETURN - Move o cursor para o próximo campo.

DELETE - Apaga caractere à esquerda do cursor.

*1 - IMPORTANTE:

É preciso muito cuidado ao comandar CONTROL+Q no modo BROWSE. Embora alguns manuais do dBASE informem que, desse modo, é encerrado o comando BROWSE sem que as alterações efetuadas nos registros em tela sejam consideradas e gravadas no disquete, ISSO NÃO ACONTECE. Em verdade, todas as alterações efetuadas em tela, COM EXCEÇÃO DAQUELAS FEITAS NO REGISTRO EM QUE O CURSOR ESTIVER NO MOMENTO DO ENCERRAMENTO DO COMANDO, SERÃO AUTOMATICAMENTE CONSIDERADAS E GRAVADAS NO DISQUETE.

Por tal razão, recomenda-se que, ao encerrar-se o comando BROWSE através de CONTROL-Q, sejam conferidas alterações eventualmente efetuadas nos registros e que o cursor seja previamente colocado em registro que não sofreu nenhuma alteração.

O comando BROWSE é muito poderoso. Nosso pequeno arquivo PESSOAL não nos permitirá sentir toda a sua potencialidade, mas propiciará uma boa idéia sobre o mesmo. Vamos experimentá-lo, considerando o arquivo PESSOAL ainda aberto, isto é, em uso:

```
D          GO TOP    <RETURN>
D          BROWSE   <RETURN>
```

Surgirá no vídeo:

```
RECORD # =00001
NOME----- CARGO----- SALAR DATAD
```

Marco Antonio	Desenhista	23000	77/07
Luis Carlos	Contador	21000	78/12
Maria Cecilia	Pedagoga	20000	81/06
Maria Beatris	Psicologa	19000	82/10
Julio Cesar	Gerente	30000	87/04
Paulo Sergio	Engenheiro	25000	82/09

Antes de experimentarmos alterar os dados, vamos deslocá-los da direita para a esquerda e vice-versa:

- Mantendo pressionada a tecla CONTROL, apertar e soltar alternativamente a tecla B, até não haver mais deslocamento das colunas de dados.
- Em seguida, mantendo do mesmo modo a tecla CONTROL pressionada, apertar e soltar a tecla Z até parar o deslocamento das colunas de dados da esquerda para a direita.

No caso do arquivo PESSOAL, o deslocamento é relativamente insignificante, em razão de a largura total dos campos dos registros ser pequena e de caber completamente na tela de vídeo. Torna-se bastante significativo, porém, quando se trabalha com arquivos cujos campos totalizam 150, 200 ou mais colunas.

Pode-se dizer o mesmo com relação ao deslizamento vertical de registros através de CONTROL+C e CONTROL+R.

Vimos que, ao ser executado, BROWSE exibiu no vídeo o quadro acima reproduzido, com o cursor posicionado no primeiro caractere do primeiro registro.

Se observarmos a primeira linha do quadro, notaremos a inscrição:

RECORD # :00001

indicando a posição atual do ponteiro de registros.

Com auxílio de CONTROL+C e CONTROL+F, vamos deslocar o cursor para o registro 3, coluna SALARIO. Feito isso, digitamos simultaneamente

CONTROL e Y

a fim de apagar o conteúdo do campo.

Surgirá um 0 apenas em seu lugar.

Digitemos agora no mesmo local o número 11111.

Em seguida, desloquemos o cursor para o mesmo campo do registro seguinte, com auxílio das teclas CONTROL+A e CONTROL+C.

Digitemos o número 99999 ali, por cima do que já existia.

Em seguida, teclemos simultaneamente

CONTROL e W

Ato contínuo, a tela de vídeo será totalmente apagada e será ouvido o ruído do DISK DRIVE, significando que as alterações efetuadas nos dados em tela no modo FULL SCREEN ON estarão sendo repassadas para os registros respectivos, isto é, os registros estarão sendo regravados com as alterações que efetuamos nos dados exibidos no vídeo.

Terminada a gravação, o dBASE retornará à condição de comando direto. O modo BROWSE foi encerrado.

Vamos verificar se as alterações efetuadas em vídeo foram repassadas para os registros:

```
D          LIST ALL NOME SALARIO OFF  <RETURN>
```

```
Marco Antonio      23000
Luis Carlos        21000
Maria Cecilia      11111
Maria Beatris      99999
Julio Cesar        30000
Paulo Sergio       25000
```

Tudo certo! Assim, a título de exercício, vamos reconstituir nos-

so arquivo PESSOAL, através do mesmo comando BROWSE.

Se, em vez de CONTROL+W, tivéssemos teclado CONTROL+Q, o comando BROWSE seria da mesma forma encerrado, mas as alterações feitas em vídeo poderiam não ser completamente repassadas para os registros, como foi observado anteriormente.

Com o comando BROWSE, pode-se também especificar a edição apenas de campos específicos:

```
D          1  <RETURN>
D          BROWSE FIELD NOME,SALARIO  <RETURN>
```

```
RECORD # =00001
NOME----- SALAR
Marco Antonio 23000
Luis Carlos   21000
Maria Cecilia 20000
Maria Beatris 19000
Julio Cesar   30000
Paulo Sergio  25000
```

Da mesma forma que no modo pleno, são igualmente válidas para este modo as teclas de controle por caractere.

Comando CHANGE

*** CHANGE permite alterar qualquer conteúdo de campo de registro de arquivo de dados. Sua sintaxe é:

```
CHANGE [<escopo>] FIELD <nome(s) de
      campo(s)>] [FOR <condição>
```

Podem ser alterados os conteúdos de um apenas ou de vários campos e registros por comando.

Se não for especificado o escopo, o comando CHANGE exibirá no vídeo, para alteração, o registro apontado pelo ponteiro do sistema no momento do comando, com o(s) campo(s) especificado(s) pelo operador.

Vamos testá-lo com o arquivo PESSOAL:

D CHANGE RECORD 3 FIELD NOME <RETURN>

Registro :00003

NOME: Maria Cecilia
Trocar?

Vamos experimentar trocar CECILIA por CRISTINA:

D Cecilia <RETURN>

Para :

D Cristina <RETURN>

NOME: Maria Cristina
Trocar?

Nesse ponto, se quisermos manter a alteração feita, basta teclar RETURN e o campo NOME do registro número 3 terá seu conteúdo efetivamente alterado.

Em nosso caso, vamos efetuar nova troca, a fim de manter o arquivo na sua forma original:

D Cristina <RETURN>

Para :

D Cecilia <RETURN>

NOME: Maria Cecilia
Trocar?

D <RETURN>

Quando é comandada a troca de conteúdo de vários registros e diversos campos, como, por exemplo:

```
CHANGE NEXT 3 FIELD NOME,CARGO,SALARIO
```

o dBASE exibe sucessivamente: em primeiro lugar, os campos e conteúdos respectivos do registro correspondente à primeira posição do comando, e depois os campos e conteúdos de cada registro seguinte. Exemplo:

```
Registro =00003
```

```
NOME: Maria Cecilia
```

```
Trocar?
```

```
CARGO: Pedagoga
```

```
Trocar?
```

```
SALARIO: 20000
```

```
TO
```

```
Registro =00004
```

```
---
```

até o último registro especificado no comando.

Comando REPLACE

*** REPLACE permite substituir o conteúdo de um ou mais campos do registro corrente ou de um conjunto de registros. Sua sintaxe é:

```
REPLACE [<escopo>] <nome do campo>  
WITH <expressão>, <nome do campo>  
WITH <expressão>... [FOR <condição>]
```

Em vez de **<expressão>**, pode-se também especificar um campo.

REPLACE é usado também para armazenar conteúdos de variáveis em campos de arquivos.

Quando REPLACE é utilizado em campo indexado, o arquivo de índice será automaticamente atualizado. Em se tratando de SUBSTITUIÇÕES MÚLTIPLAS EM CAMPO INDEXADO, os registros mudam de lugar no arquivo de índice. Por esta razão, devem ser evitadas, sendo aconselhável efetuá-las com o arquivo não indexado e reindexá-lo em seguida.

Exemplos de comandos REPLACE válidos:

- No registro corrente:

```
REPLACE SALARIO WITH SALARIO * 1.3
REPLACE CARGO WITH "DIRETOR",
SALARIO WITH 40000
```

- Em diversos registros:

```
REPLACE NEXT 3 SALARIO WITH SALARIO * 1.18
REPLACE NEXT 5 SALARIO WITH
SALARIO * 1.15 FOR SALARIO > 25000
```

- No arquivo todo:

```
REPLACE ALL SALARIO WITH SALARIO * 1.25
REPLACE ALL NOME WITH "SR(A)" + NOME
```

Observar que as substituições em campos de caracteres (campos C) devem ser indicadas entre aspas.

Depois de efetuada(s) a(s) alteração(ões), o dBASE informa quantas substituições foram efetuadas:

```
0000X Substituições.
```

11.1 - CONVERTENDO LETRAS MINÚSCULAS DE DADOS PARA LETRAS MAIÚSCULAS

O comando REPLACE conjugado com a função LETRAS MAIÚSCULAS permi-

te converter letras minúsculas dos conteúdos de campos de registros em letras maiúsculas.

O comando a ser dado nesse sentido deve ater-se à sintaxe indicada a seguir:

```
REPLACE [<escopo>] <nome do campo>  
WITH !(<nome do campo>), <nome do  
campo> WITH !(<nome do campo>),...
```

Supondo-se que se deseje passar para letras maiúsculas os conteúdos dos campos NOME e CARGO do arquivo PESSOAL, dá-se o seguinte comando, estando o arquivo em uso:

```
REPLACE ALL NOME WITH !(NOME),  
CARGO WITH !(CARGO) <RETURN>
```

12 - DELETANDO REGISTROS DO ARQUIVO

Comando DELETE

*** DELETE assinala registros para posterior eliminação do arquivo.
É usado também para eliminar arquivos do disquete.

Com relação a registros, sua sintaxe é:

```
DELETE [<escopo>] [FOR <condição>]
```

Com relação a arquivos, sua sintaxe é:

```
DELETE <nome do arquivo>
```

Neste capítulo é discutida apenas a deleção - ou eliminação - de registros de arquivos. A eliminação de arquivos de disquetes é discutida no Capítulo 21, item 21.3.

No dBASE, o processo para eliminação de registros é executado em duas etapas:

1. Os registros são inicialmente apenas "marcados" para eliminação através da função * (asterisco). Quando acionado, DELETE coloca um asterisco (*) à frente da primeira posição do registro a ser eliminado.
2. Os registros "marcados" são efetivamente eliminados do arquivo pelo comando PACK, ou "desmarcados" para eliminação através do

comando RECALL.

Função * (ASTERISCO)

*** A função * tem por finalidade identificar registros marcados para deleção.

Comando PACK

*** PACK elimina do arquivo registros marcados pelo comando DELETE.

Sua sintaxe é:

PACK

Comando RECALL

*** RECALL retira dos registros as marcas para eliminação colocadas através do comando DELETE, evitando sua deleção.

Sua sintaxe é:

RECALL [<escopo>] [FOR <condição>]

Em arquivos de grande quantidade de registros, o processo de eliminação é geralmente demorado, podendo sua execução não ser conveniente em determinadas circunstâncias de uso do arquivo. Assim, os registros a eliminar podem ser apenas "marcados", a fim de que sejam efetivamente eliminados oportunamente.

Por outro lado, esse recurso evita que sejam eliminados por engano registros que devam permanecer no arquivo.

Comando SET DELETED ON/OFF

*** Quando SET DELETED está em ON, registros marcados para eliminação não são processados por comandos como DISPLAY, LIST, COPY e APPEND FROM, no seu modo coletivo. Sua sintaxe é:

SET DELETED ON ou OFF

Vamos testar os comandos descritos por último:

D USE PESSOAL <RETURN>
D DISPLAY ALL NOME <RETURN>

00001 Marco Antonio
00002 Luis Carlos
00003 Maria Cecilia
00004 Maria Beatris
00005 Julio Cesar
00006 Paulo Sergio

D DELETE RECORD 1 <RETURN>

00001 Apagado(s)

D DISPLAY ALL <RETURN>

00001	*Marco Antonio	Desenhista	23000	07/77
00002	Luis Carlos	Contador	21000	12/78
00003	Maria Cecilia	Pedagoga	20000	06/81
00004	Maria Beatris	Psicologa	19000	04/87
00005	Julio Cesar	Gerente	30000	04/87
00006	Paulo Sergio	Engenheiro	25000	09/82

Notar que o registro 1 é exibido com um asterisco antes do nome, significando que está marcado para eliminação.

Vamos agora ativar SET DELETED:

D SET DELETED ON <RETURN>
D LIST ALL NOME <RETURN>

00002 Luis Carlos
00003 Maria Cecilia
00004 Maria Beatris

00005 Julio Cesar
00006 Paulo Sergio

O registro marcado para deleção já não é exibido porque foi ativado o comando SET DELETED ON. Vamos agora recuperá-lo:

D RECALL RECORD 1 <RETURN>

00001 Recuperado(s)

Dando-se agora o comando LIST ou DISPLAY ALL, veremos que o registro número 1 aparece sem a marca para eliminação.

Retornemos agora à condição normal - ou default - do comando SET DELETED e façamos outra experiência:

D SET DELETED OFF <RETURN>
D 5 <RETURN>
D DELETE NEXT 2 <RETURN>

00002 Apagado(s)

D LIST ALL NOME <RETURN>

00001 Marco Antonio
00002 Luis Carlos
00003 Maria Cecilia
00004 Maria Beatris
00005 *Julio Cesar
00006 *Paulo Sergio

D PACK <RETURN>

Compactação completada. 0004 Registros Copiados.

Essa mensagem do dBASE significa que os registros que estavam assinalados para supressão foram efetivamente eliminados do arquivo.

D LIST ALL NOME <RETURN>

00001 Marco Antonio
00002 Luis Carlos
00003 Maria Cecilia
00004 Maria Beatris

Os dois últimos registros do arquivo PESSOAL foram eliminados.

Mas, como necessitamos deles para continuar nossas experiências, vamos recolocá-los no arquivo com ajuda do comando APPEND:

D APPEND <RETURN>

O dBASE apresentará no vídeo uma ficha em branco de número 5, já conhecida nossa. Devemos preenchê-la com os mesmos dados utilizados anteriormente para Julio Cesar, fazendo o mesmo com relação a Paulo Sergio quando for apresentada a ficha em branco de número 6. Utilizando os mesmos dados, identificaremos mais prontamente as respostas dadas aos comandos exercitados.

Em seguida, comandamos:

D LIST ALL <RETURN>

e, depois de comprovar que nosso arquivo está recomposto, continuamos com nossas experiências:

D DELETE FOR SALARIO < 21000 <RETURN>

00002 Apagado(s)

D LIST ALL NOME SALARIO <RETURN>

00001	Marco Antonio	23000
00002	Luis Carlos	21000
00003	*Maria Cecilia	20000
00004	*Maria Beatris	19000
00005	Julio Cesar	30000
00006	Paulo Sergio	25000

Lembrar que o operador lógico AND deve ser usado entre pontos.

*** Cancelamentos de marcas de deleção de registros podem ser feitos também através de seleção de conteúdo de campos.

D RECALL FOR SALARIO < 25000 <RETURN>

00001 Recuperado(s)

Verifiquemos se o registro número 4, que se enquadra na condição especificada pelo comando, foi desmarcado para supressão:

D LIST ALL NOME FOR * <RETURN>

00005 *Julio Cesar

00006 *Paulo Sergio

Vamos agora retornar o arquivo PESSOAL à sua condição normal, retirando dos registros marcados a assinalação para eliminação:

D RECALL ALL <RETURN>

00002 Recuperado(s)

13 - EMITINDO RELATÓRIOS

Os dados armazenados em arquivos dBASE podem ser utilizados para gerar relatórios. Os recursos oferecidos pelo dBASE II PLUS nesse sentido estão reunidos no comando REPORT.

Comando REPORT

*** REPORT gera relatórios baseados em arquivos de dados. Os relatórios podem ser gerados para a tela de vídeo e/ou para a impressora. Sua sintaxe é:

```
REPORT [FORM <nome arq. de formatação>]  
      [<escopo>] [FOR <condição>] [TO PRINT]
```

Ao ser executado pela primeira vez em relação a um arquivo de dados, REPORT solicita ao usuário as definições necessárias para a organização e emissão do relatório, armazena e grava automaticamente tais definições em um arquivo com o nome especificado no comando e extensão .FRM, e emite o relatório na forma especificada.

Ao ser novamente comandado com o mesmo nome de arquivo, o comando REPORT pesquisa no diretório do disquete sua existência e, encontrando-o, emite novo relatório sem depender de novas definições.

O comando REPORT pode ser usado indiferentemente em arquivos indexados ou não. Quando usado em arquivos indexados, os registros são

listados segundo a ordem do índice em uso.

Se for comandado sem parâmetros, REPORT solicitará o nome do relatório:

```
D          REPORT  <RETURN>
          Entre o nome do relatório.
```

13.1 - EMISSÃO DE RELATÓRIOS ATRAVÉS DO VÍDEO

Utilizando nosso arquivo PESSOAL.DBF, vamos emitir alguns relatórios sobre o mesmo:

```
D          USE PESSOAL  <RETURN>
D          REPORT FORM PESSOAL  <RETURN>
```

Poderia ser usado outro nome em lugar de PESSOAL. Mas, como o arquivo de formatação a ser gerado receberá automaticamente a extensão .FRM, será conveniente adotar o mesmo nome do arquivo principal, a fim de identificá-lo com maior facilidade no diretório do disquete.

Dado o comando, o dBASE coloca a seguinte mensagem no vídeo:

```
Entre opções, M=margem esq.
L=linhas/pág. , W=tamanho pág.
```

Teclando apenas RETURN, prevalecerá o formato padrão do dBASE II PLUS, com os seguintes parâmetros:

```
M = margem esquerda: 8 colunas.
L = linhas por página: 57.
W = tamanho da página (largura): 80 colunas.
```

Outros parâmetros podem ser adotados, em conformidade com as características da impressora e formato do papel.

Vamos experimentar o formato padrão, que se presta bem para nosso

arquivo PESSOAL:

D <RETURN>

Em seguida, o dBASE colocará no vídeo a pergunta:

Cabeçalho (Y/N) ?

Como desejamos colocar cabeçalho no relatório, respondemos:

D Y <RETURN>

E o processo de definição da formatação do relatório continuará a se desenvolver com as mensagens do dBASE e as respostas nossas transcritas a seguir:

Entre com o cabeçalho da página :

D <RELATÓRIO DE EMPREGADOS <RETURN>

(O sinal < alinha o cabeçalho pela margem esquerda. Se for usado o sinal > o cabeçalho será alinhado pela margem direita.)

Relatório com espaço duplo (Y/N) ?

D N <RETURN>

Totais requeridos (Y/N) ?

D Y <RETURN>

Subtotais no relatório (Y/N) ?

N <RETURN>

A partir deste ponto, o usuário deve definir os campos que figurarão no relatório e os espaços e nomes para as colunas do mesmo, em conformidade com as solicitações do dBASE. No nosso caso, vamos fazer figurar no relatório todos os campos do arquivo PESSOAL e na mesma or-

dem em que se encontram em cada registro.

Coluna, Tamanho, Conteúdo
001

D 16,NOME <RETURN>

Entre com o cabeçalho =

D NOME <RETURN>

002

D 11,CARGO <RETURN>

Entre com o cabeçalho =

D CARGO <RETURN>

003

D 6,SALARIO <RETURN>

Entre com o cabeçalho =

D SAL. <RETURN>

Como o campo é numérico, o dBASE pergunta:

Totais requeridos <Y/N> ?

D Y <RETURN>

004

D 6,DATADM <RETURN>

Entre com o cabeçalho =

D ADMIS <RETURN>

005

Como temos apenas quatro campos de dados no arquivo PESSOAL, encerramos a definição de formatação do relatório, apenas digitando:

D <RETURN>

*** A definição de formato de relatórios através do comando REPORT é encerrada quando se pressiona a tecla RETURN ao ser apresentado o número para uma nova coluna.

Nesse ponto, o dBASE emite o seguinte relatório através do vídeo:

Página Nº 00001

25/10/87

RELATORIO DE EMPREGADOS

NOME	CARGO	SAL.	ADMIS
Marco Antonio	Desenhista	23000	77/07
Luis Carlos	Contador	21000	78/12
Maria Cecilia	Pedagoga	20000	81/06
Maria Beatris	Psicologa	19000	87/04
Julio Cesar	Gerente	30000	87/04
Paulo Sergio	Engenheiro	25000	82/09
** TOTAL **		138000	

Além do cabeçalho especificado durante a definição de formatação do relatório, um segundo cabeçalho pode ser adicionado a cada página do mesmo, através do comando descrito a seguir:

Comando SET HEADING TO

*** SET HEADING TO permite que um cabeçalho adicional seja incluído

em cada página de um relatório. Sua sintaxe é:

```
SET HEADING TO <cabeçalho adicional>
```

Exemplo:

```
SET HEADING TO RELATORIO PARCIAL PESSOAL
```

SET HEADING TO deve ser comandado antes do comando REPORT.

O cabeçalho adicional será exibido ou impresso em seguida ao número de cada página do relatório.

Durante a definição de formatação do relatório, se for respondido Y (= sim) à pergunta:

```
Subtotais no relatório (Y/N) ?
```

serão formuladas mais as seguintes questões pelo dBASE:

```
Entre com campos de subtotais :
```

```
Relatório resumido (Y/N) ?
```

```
Alimenta página após subtotais (Y/N) ?
```

```
Entre cabeçalho do subtotal :
```

Ainda com relação à definição da formatação de relatórios, os seguintes pontos devem ser observados atentamente:

- Até 254 caracteres são permitidos no cabeçalho do relatório.
- Até 24 campos podem figurar num relatório.
- A largura da coluna destinada a cada dado não deve ser menor do que a definida para o campo respectivo na estrutura do arquivo. Deve ter no mínimo um espaço a mais, mas pode ser tão mais larga quanto desejado pelo usuário. Se for menor, os dados serão truncados.

- Os nomes das colunas devem ocupar espaço menor do que o destinado às mesmas ou, no máximo, o mesmo espaço. Quando menores do que as colunas, serão automaticamente centralizados pelo dBASE.
- Para definir o formato de relatório de um arquivo, deve-se recorrer à pequena ficha adotada para definir a estrutura do arquivo (Figura B.1) ou, na falta desta, à exibição da estrutura do arquivo (comandando-se DISPLAY STRUCTURE), com a finalidade de informar-se sobre os campos de cada registro e respectivos tamanhos.

Antes de passarmos para o próximo item, verifiquemos o diretório do disquete, a fim de constatar nele a presença do arquivo de formatação PESSOAL.FRM, gravado automaticamente pelo dBASE:

```
D          DISPLAY FILES LIKE *.*  <RETURN>
```

```
...
PESSOAL .FRM
```

13.2 - EMISSÃO DE RELATÓRIOS

ATRAVÉS DE IMPRESSORA

Da mesma forma que no vídeo, relatórios de dados podem ser emitidos por uma impressora acoplada ao microcomputador.

Se quisermos emitir o relatório acima através de impressora, isto é, impresso em papel, comandamos:

```
D          REPORT FORM PESSOAL TO PRINT  <RETURN>
```

Nesse caso, como as definições para formatação do relatório já se encontram armazenadas no arquivo PESSOAL.FRM, o relatório é imediatamente emitido, desde que a impressora esteja pronta para o trabalho.

Normalmente, antes de dar início à impressão, a impressora recebe diretamente do dBASE um comando que faz com que o papel da mesma avan-

ce automaticamente até o início da próxima página lógica, que, em condições usuais, deverá ser também o início de uma página do formulário contínuo. Este recurso do dBASE existe para fazer com que a impressão do relatório se inicie no topo da página. Se o relatório se estender por diversas páginas, a impressão sempre será iniciada no topo de cada uma, com avanço automático de algumas linhas antes e depois do picote divisor das mesmas.

Caso o papel já tenha sido previamente ajustado pelo usuário, esse avanço poderá ser evitado se, antes de comandar a emissão do relatório, for feita a desativação do comando indicado a seguir, cuja ativação se dá automaticamente quando o dBASE é inicializado:

Comando EJECT

*** EJECT provoca o avanço do papel da impressora para posição correspondente ao início de uma nova página lógica.

O comando EJECT pode ser ativado ou desativado através do comando SET ON/OFF:

SET EJECT ON ativa EJECT.

SET EJECT OFF desativa EJECT.

Assim, para evitar o avanço automático do papel da impressora ao ser dado o comando para emissão do relatório, deve-se comandar antes:

D SET EJECT OFF <RETURN>

13.3 - EMISSÃO DE RELATÓRIOS PARCIAIS

Uma vez definida a formatação de relatório para o arquivo, torna-se fácil e rápido emitir relatórios parciais sobre o mesmo, já que, ao ser dado o comando competente, o dBASE sempre utiliza o respectivo arquivo de formatação para a tarefa. A busca do mesmo no disquete é feita automaticamente pelo dBASE.

Assim, vamos praticar um pouco com o mesmo, a fim de conhecer outros recursos do comando REPORT:

D REPORT FORM PESSOAL FOR
SALARIO < 21000 <RETURN>

Página Nº 00001
25/10/87

RELATORIO DE EMPREGADOS

NOME	CARGO	SAL.	ADMIS
Maria Cecilia	Pedagoga	20000	06/81
Maria Beatris	Psicologa	19000	10/82
** TOTAL **		39000	

D REPORT FORM PESSOAL FOR
!(CARGO) = "DESENHISTA" <RETURN>

Página Nº 00001
25/10/87

RELATORIO DE EMPREGADOS

NOME	CARGO	SAL.	ADMIS
Marco Antonio	Desenhista	23000	07/77
** TOTAL **		23000	

Pode-se também gerar relatórios na tela de vídeo e na impressora, simultaneamente, comandando-se previamente:

SET PRINT ON <RETURN>

ou pressionando-se concomitantemente, antes de comandar a emissão do relatório, as teclas:

CONTROL e P.

Tais comandos devem ser dados com a impressora já ligada.

Antes de passarmos para o próximo item, recomendamos experimentar os seguintes comandos, como exercício para aquisição de experiência:

```
D          REPORT FORM PESSOAL FOR SALARIO .AND.  
          !(CARGO) = "GERENTE"    <RETURN>  
  
D          REPORT FORM PESSOAL FOR !(CARGO) = "CONTA-  
          DOR" .AND. SALARIO > 19000    <RETURN>  
  
D          REPORT FORM PESSOAL FOR  
          DATADM > "81/06"    <RETURN>  
  
D          REPORT FORM PESSOAL NEXT 3  
          FOR SALARIO >= 20000    <RETURN>
```

13.4 - INSTRUÇÕES PARA A IMPRESSORA

Estando o DBASE ativado e desejando-se enviar instruções para a impressora, a fim de alterar a modalidade de impressão, entrelinhamento etc., usa-se a instrução ? (ponto de interrogação), que é a mesma do BASIC MSX e equivalente à instrução PRINT, seguida dos códigos significativos para tal finalidade.

A impressora deverá estar ligada e pronta para receber comandos. Nessa condição, deve-se ligá-la logicamente ao dBASE, comandando-se:

```
D          SET PRINT ON    <RETURN>
```

Em seguida, poderão ser enviadas as instruções desejadas.

Geralmente, tais instruções são transmitidas na forma de uma sequência de códigos decimais precedidos por uma das instruções PRINT ou

LPRINT. Os códigos que representam tais instruções podem variar para cada tipo de impressora, devendo sua utilização basear-se no manual de uso respectivo.

No caso da impressora MÓNICA EI 6011, por exemplo, para se ativar a impressão em QUALIDADE CARTA (ou QC), através do BASIC MSX, é preciso enviar-lhe apenas a seguinte sequência de códigos: CHR\$(27);"G". E, para desativar a impressão em QC e voltar ao modo padrão, em QUALIDADE DADOS (ou QD), envia-se a sequência de códigos CHR\$(27);"H".

Através do dBASE, tais instruções são, respectivamente:

```
? CHR(27) + CHR(71) <RETURN>
```

```
? CHR(27) + CHR(72) <RETURN>
```

Se verificarmos a tabela ASCII, constataremos que o código decimal de G é 71 e que o de H é 72.

Por exemplo, para produzir relatórios impressos na modalidade de IMPRESSÃO CONDENSADA SIMPLES na impressora MÓNICA EI 6011, é preciso enviar à mesma apenas o código decimal 15.

Através do dBASE, a instrução correspondente é:

```
? CHR(15) <RETURN>
```

Para desativar a impressão em modalidade CONDENSADA e voltar para a normal:

```
? CHR(18) <RETURN>
```

Notar que a função CHR do dBASE é escrita sem \$. No BASIC MSX, a mesma tem a grafia CHR\$.

A função CHR do dBASE II PLUS é estudada no Capítulo 23.

13.4.1 - ALTERAÇÃO DA MARGEM

ESQUERDA DE IMPRESSÃO

Normalmente, a impressão de relatórios dBASE é iniciada na coluna 8. Esse padrão pode ser mudado através do comando descrito a seguir.

Comando SET MARGIN TO

Sua sintaxe é:

SET MARGIN TO <n>

sendo n o número da coluna onde deve ser iniciada a impressão.

1.4 - UM NOVO ARQUIVO PARA NOVAS EXPERIÊNCIAS

Até esta parte de nosso curso, todas as experiências que fizemos com o dBASE basearam-se em um único arquivo, o PESSOAL, a fim de facilitar a análise e o entendimento dos resultados gerados pela execução dos diversos comandos introduzidos e descritos.

Com a mesma finalidade, mas ampliando o elenco de experiências, vamos agora criar um novo tipo de arquivo, com diversos campos numéricos, a fim de exercitar determinados comandos do dBASE II PLUS que manipulam dados dessa natureza.

Por outro lado, por possuir maior número de campos e utilizar espaço horizontal maior do que as 40 colunas habituais da tela de vídeo, o novo arquivo ensejará oportunidade para uso da versão de oitenta colunas do dBASE II PLUS, bem como do cartucho de expansão de colunas da tela de vídeo. Se este não estiver disponível, porém, o novo arquivo poderá ser usado indiferentemente com a versão de 40 ou de 80 colunas. Obviamente, a visualização dos dados com o uso do cartucho de expansão para 80 colunas será mais cômoda e estética.

Introduziremos também a utilização de um segundo disk drive, visando diversificar nossas experiências no trato com o dBASE.

A utilização de dois disk drives no sistema se torna desejável - e mesmo imprescindível - se um arquivo a ser processado não couber no mesmo disquete que contém os programas do dBASE. Em tal caso, isto é, se houver apenas um disk drive disponível no conjunto, será tão fre-

quente a necessidade de intercambiar os disquetes no disk drive - aumentando consideravelmente a possibilidade de engano na troca dos disquetes, com sério risco de destruir o arquivo - que o seu processamento se torna impraticável.

No caso de nosso novo arquivo - que terá finalidade puramente didática - não ocorrerá tal problema, já que ele será bem pequeno. Poderá ser criado em disquete à parte ou no mesmo do dBASE, como foi feito com o arquivo PESSOAL.

A fim de enquadrá-lo nas experiências que realizaremos, recomendamos:

- A - CRIAR O NOVO ARQUIVO EM DISQUETE INDEPENDENTE, SE HOUVER DOIS DISK DRIVES FUNCIONANDO COM O MICROCOMPUTADOR.
- B - CRIAR O NOVO ARQUIVO NO MESMO DISQUETE DE PROGRAMAS DO dBASE, SE HOUVER APENAS UM DISK DRIVE OPERANDO NO CONJUNTO.

No caso A:

- Alojarmos o disquete que contém o dBASE no disk drive A e um novo disquete, já formatado e sem o DOS, no disk drive B.

No caso B:

- Colocar normalmente o disquete contendo o dBASE no único disk drive físico disponível, que, todavia, é reconhecido e tratado pelo sistema operacional para disco como dois disk drives lógicos, A e B, como sabemos.

Neste caso, como nossas experiências vão ser doravante baseadas em dois disk drives, A e B, toda vez que o sistema solicitar troca de disquetes através da mensagem:

Insira disco no drive B: (ou A:)
e pressione qualquer tecla

ou

Insert diskette for drive b: (ou a:)
and strike a key when ready

bastará teclar RETURN, SEM EFETUAR A TROCA. Nessa condição, o sistema mudará internamente a configuração do disk drive lógico, de A para B ou vice-versa, equivalendo esta mudança à troca de disquetes solicitada, e tratará o mesmo disquete como se fossem dois diversos. Este pequeno truque nos poupará o trabalho de manusear e processar dois diferentes disquetes no disk drive - no caso de o novo arquivo ser criado em disquete à parte do de programas do dBASE - e eliminará a possibilidade de enganos durante o processamento do arquivo. Devemos lembrar, todavia, que pode ser utilizado apenas quando um arquivo principal, de extensão DBF, e auxiliares (NDX, FRM etc.) couberem no mesmo disquete de sistema do dBASE.

Nosso novo arquivo será criado com base nas informações contidas na pequena ficha reproduzida a seguir:

Nome do arq.: CTRLSTOQ.DBF Disco: 123				
Assunto: CONTROLE DE ESTOQUE DE PRODUTOS				
Campo	T	E	D	Referência
1 - CODIGO	C	4		CÓDIGO DO PRODUTO
2 - DENOM	C	20		NOME DO PRODUTO
3 - DEP	C	3		DEPTº DE VENDA
4 - QTMIN	N	4		QTDE. MÍNIMA P/EST.
5 - QTAT	N	5		QTDE. ATUAL EM EST.
6 - PRC	N	8	2	PREÇO DE CUSTO
7 - PRV	N	8	2	PREÇO DE VENDA
8 - VALC	N	10	2	VALOR DE CUSTO
9 - VALV	N	10	2	VALOR DE VENDA
T=Tipo / E=Espaço / D=Decimais				

FIGURA 14.1

Depois de alojado(s) o(s) disquete(s) no(s) disk drive(s) conforme descrição acima, ativamos o dBASE.

Em seguida, e como fizemos anteriormente, usamos o comando CREATE para criar o novo arquivo:

D

CREATE B: CTRLSTOQ <RETURN>

O dBASE solicitará entrar com a estrutura do registro, a partir do campo de número 1. Digitaremos os dados correspondentes, de acordo com a ficha acima. Quando for apresentado o número de campo 10, teclaremos apenas RETURN para acabar a definição da estrutura dos registros do arquivo. O dBASE perguntará se vão ser incluídos dados "agora". Digitaremos apenas N (= não) porque, antes, vamos verificar se a estrutura foi gravada no disquete em conformidade com o planejado.

Devemos lembrar que o arquivo foi criado no disquete alojado no disk drive B e que é lá que ele se encontra. Por essa razão, devemos instruir o dBASE para que o acesse lá. Em vez de digitarmos a indicação apropriada para o caso (B:) a cada comando dado, podemos adotar um comando que tem validade indeterminada (até que seja cancelado por outro comando ou até o computador ser desligado), a seguir introduzido:

Comando SET DEFAULT TO

*** SET DEFAULT TO muda o disk drive corrente para o disk drive especificado.

O disk drive corrente é normalmente o A. Quando não especificado outro, o dBASE sempre tenta executar um comando recebido no disk drive corrente.

Assim, para que todos os comandos que dermos sobre o novo arquivo sejam dirigidos para o disk drive B, onde o localizamos, comandamos:

D

SET DEFAULT TO B <RETURN>

Tal comando só precisa ser dado no início do processamento do arquivo, e apenas uma vez, como dissemos. O disquete que contém o dBASE deve permanecer normalmente no disk drive A, sendo acessado automaticamente pelo sistema quando for necessário.

Em seguida, comandamos:

D

USE CTRLSTOQ <RETURN>

D

LIST STRUCTURE <RETURN>

Em resposta, o dBASE exibirá as seguintes informações no vídeo:

Estrutura para o arquivo :B:CTRLST00.DBF

Número de registros :00000

Data da última atualização :30/11/87

Uso primário do banco de dados

Cpo	Nome	Tipo	Tam.	Dec.
001	CODIGO	C	004	
002	DENOM	C	020	
003	DEP	C	003	
004	QTMIN	N	004	
005	QTAT	N	005	
006	PRC	N	008	002
007	PRV	N	008	002
008	VALC	N	010	002
009	VALV	N	010	002
** TOTAL **			00069	

Visto estar certa e gravada a estrutura do arquivo, começamos em seguida a dar entrada de dados no arquivo:

D

APPEND <RETURN>

Ato contínuo, o dBASE apresentará no vídeo uma "ficha em branco" de número 1, com a seguinte configuração:

Registro 00001

CODIGO	:	:	
DENOM	:		:
DEP	:	:	
QTMIN	:	:	
QTAT	:	:	
PRC	:	:	
PRV	:	:	
VALC	:	:	
VALV	:	:	

Como vimos anteriormente, depois de preenchida uma ficha, outra será apresentada e, assim, sucessivamente. Para ser encerrado o modo de entrada de dados, deve-se apenas teclar RETURN quando uma nova ficha em branco for introduzida pelo programa no vídeo, estando o cursor posicionado no primeiro espaço do primeiro campo da mesma.

Lembrar que o modo SET CONFIRM pode ser ativado, se houver conveniência para o operador.

Nosso novo arquivo será uma réplica miniaturizada de um arquivo de controle de estoque de uma loja de departamentos. Para que possam ser melhor analisados e compreendidos os resultados de comandos executados do dBASE, consideraremos apenas três departamentos de vendas e dois produtos diferentes de cada um.

Os dados dos produtos que devem ser digitados em cada "ficha em branco" apresentada no vídeo são reproduzidos abaixo.

Lembrar que o dBase adota o sistema americano de separação de casas decimais, que usa o ponto (.) em vez da vírgula (,), e não aceita a separação da parte inteira a cada três dígitos, de forma que os valores numéricos devem ser digitados com a formatação 0000000.00.

Se houver erro durante a digitação dos dados, efetuar as correções com auxílio das TECLAS DE CONTROLE POR CARACTERE (ver seu indicador visual no Capítulo 7). Correções posteriores deverão ser feitas através dos comandos EDIT, BROWSE, CHANGE ou REPLACE, com auxílio das teclas de controle por caractere.

0033	0043
CAMISA TERGAL BRANCA	GRAVATA CROCHE AZUL
CAM	CAM
5	5
20	30
450.00	123.50
652.50	179.00
9000.00	3705.00
13050.00	5370.00

0127
 JOGO CAFE CERAMICA
 BAZ
 3
 12
 355.00
 514.75
 4260.00
 6177.00

0345
 JOGO CHA CERAMICA
 BAZ
 3
 10
 423.50
 614.00
 4235.00
 6140.00

1234
 SAPATO SOCIAL CROMO
 SAP
 3
 15
 1234.45
 1790.00
 18516.75
 26850.00

1427
 SAPATO ESPORTE CAMUR
 SAP
 3
 18
 987.00
 1431.00
 17766.00
 25758.00

Depois de completada a entrada de dados, fazemos a primeira consulta ao arquivo:

```
D          LIST  <RETURN>

00001  0033 CAMISA TERGAL BRANCA CAM    5    20    450.00    652.50    9000.00    13050.00
00002  0043 GRAVATA CROCHE AZUL CAM     5    30    123.50    179.00    3705.00    5370.00
00003  0127 JOGO CAFE CERAMICA BAZ      3    12    355.00    514.75    4260.00    6177.00
00004  0345 JOGO CHA CERAMICA BAZ       3    10    423.50    614.00    4235.00    6140.00
00005  1234 SAPATO SOCIAL CROMO SAP      3    15    1234.45    1790.00    18516.75    26850.00
00006  1427 SAPATO ESPORTE CAMUR SAP     3    18    987.00    1431.00    17766.00    25758.00
```

Constatando estar tudo certo com o novo arquivo, encerramos esta primeira sessão com o mesmo, para prosseguir com novos ensaios no próximo capítulo:

```
D          QUIT  <RETURN>
```

15 - ARQUIVOS DE SUBTOTAIIS

Um dos recursos de grande utilidade do dBASE para gerenciamento de arquivos de dados predominantemente numéricos é o de gerar arquivos e relatórios de subtotais, muito utilizados na administração de empresas em geral.

No dBASE, o encarregado dessa tarefa é o

Comando TOTAL ON

*** TOTAL ON subtotaliza campos numéricos de registros cujos conteúdos do campo chave indicado são iguais, armazenando os resultados em arquivo-resumo ou de subtotais. Sua sintaxe é:

```
TOTAL ON <nome do campo chave> TO  
      <nome do arquivo> [FIELD <campo 1,  
      campo 2, ...>] [FOR <condição>]
```

Quando a opção FIELD não é adotada, todos os campos numéricos são automaticamente subtotalizados.

TOTAL ON é válido apenas para arquivos indexados.

Como primeiro passo, vamos indexar o novo arquivo CTRLSTOQ pelo campo DEP (= Departamento), que será o campo chave, a fim de obter determinados subtotais. Depois de ativado o dBASE, comandar:

```
D      SET DEFAULT TO B   <RETURN>
D      USE CTRLSTOP      <RETURN>
D      INDEX ON DEP TO CTRLDEP   <RETURN>
```

Após surgir no vídeo a mensagem:

```
00006 Indexados.
```

podemos fazer o primeiro teste com o comando TOTAL ON:

```
D      TOTAL ON DEP TO TOTDEP
                                FIELD VALC,VALV   <RETURN>
```

Após uns poucos segundos, o dBASE responderá:

```
00006 Registros copiados.
```

significando que os dados dos campos VALC e VALV foram subtotalizados em função do campo chave DEP e armazenados no arquivo TOTDEP, que também recebe automaticamente do dBASE a extensão DBF. Examinemos TOTDEP:

```
D      USE TOTDEP   <RETURN>
D      LIST DEP,VALC,VALV   <RETURN>
```

00001	CAM	12705.00	18420.00
00002	BAZ	8495.00	12317.00
00003	SAP	36282.75	52608.00

Como vemos, o comando TOTAL ON criou um novo arquivo com três registros, nos quais resumiu os valores dos campos especificados.

Se comandarmos agora:

```
D      LIST FILES LIKE *.*   <RETURN>
```

constataremos no diretório do disquete a presença dos novos arquivos, criados a partir do CTRLSTOQ:

```
...  
CTRLSTOQ DBF  
CTRLDEP  NDX  
TOTDEP   DBF
```

Um ponto a observar com relação ao arquivo de subtotais é que ele sempre usa a denominação de um item do grupo subtotalizado como título desse grupo. Isto pode ser constatado se, em vez do comando dado anteriormente (LIST DEP, VALC, VALV), for comandado:

```
LIST ALL <RETURN>
```

Podemos também obter informações selecionadas do arquivo de subtotais. Por exemplo, para conhecermos os valores totais de custo e de venda das mercadorias do departamento CAM (= Camisaria), comandamos:

```
D          DISP DEP,VALC,VALV  
                                FOR DEP = "CAM" <RETURN>  
  
00001  CAM      12705.00      18420.00
```

Para obtermos o valor total de custo das mercadorias do departamento SAP (= SAPATOS), comandamos:

```
D          DISP DEP,VALC FOR DEP = "SAP" <RETURN>  
  
00003  SAP      36282.75
```

Para obtermos apenas o valor total de venda de sapatos estocados:

```
D          DISP VALV FOR DEP = "SAP" OFF <RETURN>  
  
00003  52608.00
```

Se, por outro lado, quisermos ter uma projeção desse valor com um aumento de 30% a ser adotado, comandamos:

```
D          DISP STR (VALV * 1.3,10,2)  
                                FOR DEP = "SAP" <RETURN>
```

00003 68390.40

Os dados do arquivo de subtotais TOTDEP.DBF podem ser reproduzidos através da impressora, ativando-se a mesma com o comando

SET PRINT ON <RETURN>

ou pressionando-se simultaneamente as teclas

CONTROL e P

como visto anteriormente.

Relatórios impressos totais ou parciais também podem ser gerados pelo arquivo TOTDEP, como exemplificado com o arquivo PESSOAL, através do comando:

REPORT FORM.

16 - TRABALHANDO COM DOIS ARQUIVOS A UM SÓ TEMPO

O dBASE possibilita trabalhar com dois arquivos concomitantemente, recurso que pode ser muito útil em determinadas circunstâncias. Por exemplo, para tomar decisões sobre reposição e renovação dos estoques mercantis, o administrador deve levar em conta os itens existentes em cada departamento, seus preços, valores parciais e totais, verba disponível etc.

Nessa situação, operar dois arquivos complementares, para extrair deles as informações necessárias, pode ser a solução ideal.

Uma simulação desse fato pode ser feita com os arquivos criados nos Capítulos 14 e 15, CTRLSTOQ.DBF e TOTDEP.DBF.

O comando do dBASE para tal finalidade é descrito a seguir:

Comando SELECT PRIMARY/SECONDARY

*** SELECT PRIMARY/SECONDARY permite que dois arquivos de dados sejam usados ao mesmo tempo sem a necessidade de abri-los e fechá-los a cada operação, alternativamente.

Sua utilização é exemplificada a seguir. Vamos usar os dois citados arquivos. A operação é absolutamente normal, isto é, no caso de os programas estarem todos, inclusive os do dBASE, num só disquete, deve-se alojá-lo normalmente no disk drive único. No caso de o sistema contar com dois disk drives, alojar o disquete do dBASE no A e o que con-

tém os arquivos no B.

Depois de ativado o dBASE, comandar:

```
D          SET DEFAULT TO B   <RETURN>
D          USE CTRLSTOQ      <RETURN>
D          SELECT SECONDARY  <RETURN>
D          USE TOTDEP        <RETURN>
```

Dados tais comandos, os dois arquivos especificados estão abertos para uso concomitante, sendo o CTRLSTOQ considerado como de uso primário pelo dBASE e o TOTDEP como de uso secundário.

Após a execução dos comandos, em função de sua ordem, o arquivo SELECIONADO automaticamente para uso é o SECUNDÁRIO, como se pode verificar através do comando:

```
D          DISPLAY STATUS    <RETURN>
```

ao qual o dBASE responde:

```
Banco selecionado -B:TOTDEP .DBF
Uso secundário do banco de dados.
```

```
Banco não selecionado -B:CTRLSTOQ.DBF
Uso primário do banco de dados.
```

...

Para selecionarmos para uso o arquivo PRIMÁRIO, comandamos:

```
D          SELECT PRIMARY    <RETURN>
```

Para retornar ao arquivo SECUNDÁRIO:

```
D          SELECT SECONDARY  <RETURN>
```

*** Quando dois arquivos de dados estão abertos para uso simultâneo, o dBASE precisa ser informado através do comando SELECT para qual deles deve direcionar os comandos recebidos do operador.

Tal seleção não precisa ser feita para cada comando dado. Um arquivo selecionado para uso por SELECT continua nessa condição até que o outro o substitua por força de um novo comando SELECT.

Arquivos de índice podem ser usados juntamente com o arquivo selecionado por SELECT e todos os comandos do dBASE são válidos na condição de uso duplo de arquivos.

16.1 - EXIBINDO DADOS DE DOIS

ARQUIVOS AO MESMO TEMPO

Vimos que, estando em uso simultâneo dois arquivos de dados, os comandos do dBASE devem ser direcionados para um ou outro arquivo, que se torna o selecionado para recebê-los, sendo esse direcionamento feito através do comando SELECT.

Veremos agora que é possível processar dados de dois arquivos em uso, podendo ser de campos afins ou não.

*** Para processar dados de dois arquivos em uso simultâneo, deve-se acrescentar aos campos a serem acessados os prefixos:

P. (de Primário) e

S. (de Secundário).

Vamos simular tal situação, fazendo uso dos arquivos CTRLSTOQ e TOTDEP e dos comandos estudados até este ponto de nosso curso.

Depois de ativado o dBASE, comandar:

```
D      SET DEFAULT TO B   <RETURN>
D      USE CTRLSTOQ      <RETURN>
D      SELECT SECONDARY  <RETURN>
D      USE TOTDEP        <RETURN>
D      DISPLAY P.DENOM,P.QTAT,P.VALC,
                                     S.DEP,S.VALC <RETURN>
```

A resposta pelo vídeo será:

```
00001 CAMISA TERGAL BRANCA      20      9000.00 CAM      12705.00
```

Através de DISPLAY, comandamos a exibição do conteúdo dos campos DENOM, QTAT e VALC do arquivo selecionado como PRIMÁRIO, e dos campos DEP e VALC do arquivo selecionado como SECUNDÁRIO, a fim de verificarmos a relação entre os valores dos campos VALC dos dois arquivos. Neste caso, em vista de que, ao serem ativados os arquivos, inicialmente seus ponteiros sempre apontam para o registro número 1, os dados dos dois campos VALC são correspondentes porque o registro número 1 do arquivo CTRLSTOQ está subtotalizado no registro número 1 do TOTDEP.

Mas devemos lembrar que TOTDEP é um arquivo de SUBTOTAIS, conforme visto no Capítulo 15, e que, nessa condição, subtotalizou em apenas três registros os dados dos seis registros do arquivo CTRLSTOQ. Assim, se desejarmos conhecer a mesma relação, isto é, custo de um item estocado e custo total de todos os itens do mesmo departamento, quando os registros que os contêm não são os primeiros de cada arquivo, devemos direcionar antes os ponteiros para os registros devidos. Para fazê-lo, temos de verificar, se não soubermos de memória, quais são os números de registros correspondentes em ambos os arquivos. Com relação ao item SAPATO SOCIAL CROMO, por exemplo, os registros correspondentes são:

- Registro 5 do arquivo PRIMÁRIO.
- Registro 3 do arquivo SECUNDÁRIO.

Assim, para conhecermos a citada relação, adotamos a seguinte sequência de comandos:

```
D      SELECT PRIMARY   <RETURN>
D      GOTO 5           <RETURN>
D      SELECT SECONDARY <RETURN>
D      GOTO 3           <RETURN>
D      DISPLAY P.DENOM,P.QTAT,
D      P.VALC,DEP,VALC <RETURN>
```

e teremos como resposta no vídeo:

```
00003  SAPATO SOCIAL CROMO      15    18516.75 SAP    36282.75
```

*** Os prefixos P. e S. podem ser dispensados quando destinados a designar campos do arquivo que está selecionado para uso.

Podemos notar no último comando que os campos DEP e VALC não foram especificados com prefixos, porque o arquivo a que pertencem é o selecionado para uso no momento.

Façamos mais uma experiência, partindo dessa mesma situação:

```
D          GOTO 2    <RETURN>
D          SELECT PRIMARY  <RETURN>
D          4    <RETURN>
D          DISPLAY CODIGO DENOM S.DEP    <RETURN>
```

(Observar que não estamos usando vírgulas para separar os campos, o que é perfeitamente possível em tais comandos.)

Teremos como resposta:

```
00004  0345 JOGO CHA CERAMICA    BAZ
```

14.2 - DESLOCANDO OS PONTEIROS DE REGISTROS SIMULTANEAMENTE EM DOIS ARQUIVOS EM USO

Na operação conjunta de dois arquivos de dados, pode ser necessário deslocar os ponteiros de registros simultaneamente nos dois arquivos. Suponha-se, por exemplo, que seja necessário comparar o arquivo CTRLSTOQ atual com uma cópia de segurança do mesmo, feita algum tempo atrás em outro disquete. Entre outras, uma das finalidades da comparação seria estudar a evolução dos preços, a movimentação dos estoques e outros dados de interesse da administração da empresa.

Nessas condições, se os ponteiros apontarem automática e sistema-

ticamente para os registros equivalentes (de mesmo número) em ambos os arquivos, a comparação se torna muito mais rápida, prática e eficaz.

Tal possibilidade é propiciada pelo dBASE através do comando descrito a seguir:

Comando SET LINKAGE ON/OFF

*** SET LINKAGE ON provoca o deslocamento simultâneo dos ponteiros de registros de dois arquivos em uso, operados através do comando SELECT. SET LINKAGE OFF cessa tal condição.

Podemos simular uma situação de uso como a descrita acima, fazendo uma cópia do arquivo CTRLSTOQ e, em seguida, alterando os dados da cópia, para fazê-los corresponder "grosso modo" a alguns dias atrás.

Faremos essa cópia usando o comando COPY do próprio dBASE, o qual será estudado em detalhes no Capítulo 21. Para que possa coexistir no mesmo disquete em que mantemos os outros programas e também ser utilizado para o fim proposto por nós, devemos dar ao arquivo-cópia um nome diferente; digamos, CTRLSTCP.

Assim, estando o dBASE ativado e o sistema funcionando como indicado no Capítulo 14, comandar:

```
D          SET DEFAULT TO B   <RETURN>
D          USE CTRLSTOQ      <RETURN>
D          COPY TO B:CTRLSTCP <RETURN>
```

Após retornar o ponto de prontidão para comando do dBASE, conferimos a presença do novo arquivo de dados no disquete:

```
D          LIST FILES LIKE *.DBF ON B   <RETURN>
```

Confirmada a verificação, procederemos à alteração dos conteúdos dos campos QTAT, PRV e VALV de todos os registros:

```
D          USE CTRLSTCP      <RETURN>
D          REPLACE NEXT 3 QTAT
```

WITH QTAT + 5 <RETURN>

00003 Substituições.

```
D      GOTO 4 <RETURN>
D      REPLACE NEXT 3 QTAT
      WITH QTAT + 7 <RETURN>
```

00003 Substituições.

Com os três últimos comandos, aumentamos as quantidades de todos os itens do arquivo, em 5 unidades nos três primeiros e 7 nos demais. Em seguida, vamos alterar os valores dos preços de venda correspondentes, tornando-os menores:

```
D      REPLACE ALL PRV WITH PRV * 0.9 <RETURN>
```

00006 Substituições.

Agora vamos atualizar os dados do campo VALV de todos os registros, em função das quantidades e preços alterados:

```
D      REPLACE ALL VALV WITH QTAT * PRV <RETURN>
```

Ao final, apenas para verificar quais "eram" os dados dos campos alterados na época simulada como de alguns dias atrás, comandamos:

```
D      DISPLAY ALL QTAT,PRV,VALV <RETURN>
```

Teremos então o seguinte quadro:

00001	25	587.25	14681.25
00002	35	161.10	5638.50
00003	17	463.27	7875.59
00004	17	552.60	9394.20
00005	22	1611.00	35442.00
00006	25	1287.90	32197.50

Temos agora duas versões do mesmo arquivo e vamos usá-las simul-

taneamente, para fins de comparação, com auxílio dos comandos SELECT e SET LINKAGE ON:

(SET LINKAGE ON deve sempre preceder os demais comandos e os arquivos devem ser colocados em uso ao mesmo tempo, a fim de que a posição inicial dos ponteiros seja no primeiro registro de cada. Se um dos arquivos já estiver em uso, como é o caso do CTRLSTCP no momento, convirá fechá-lo antes de dar os comandos.)

```
D      USE      <RETURN>
D      SET LINKAGE ON  <RETURN>
D      USE CTRLSTOQ   <RETURN>
D      SELECT SECONDARY <RETURN>
D      USE CTRLSTCP   <RETURN>
```

Em seguida, podemos comandar a exibição dos dados que nos interessam comparar, lembrando que podemos obtê-los também através da impressora com auxílio do comando SET PRINT ON.

Vamos comparar as quantidades em estoque e os valores de venda de todos os itens em estoque:

```
D      LIST S.DENOM S.QTAT
      S.VALV P.QTAT P.VALV OFF  <RETURN>
```

Após tal comando, teremos os seguintes dados exibidos:

CAMISA TERGAL BRANCA	25	14681.25	20	13050.00
GRAVATA CROCHE AZUL	35	5638.50	30	5370.00
JOGO CAFE CERAMICA	17	7875.59	12	6177.00
JOGO CHA CERAMICA	17	9394.20	10	6140.00
SAPATO SOCIAL CROMO	22	35442.00	15	26850.00
SAPATO ESPORTE CAMUR	25	32197.50	18	25758.00

No quadro exibido podem ser comparados facilmente os estoques de mercadorias e respectivos valores de venda em duas datas diferentes, já que os arquivos utilizados supostamente refletem tal situação, permitindo analisar, neste caso, a evolução dos estoques e valores envolvidos. Outras análises poderão ser feitas, obviamente. Se se desejar

comparar os preços de venda, por exemplo, o último comando deverá ser:

```
LIST S.PRV P.PRV <RETURN>
```

Dois pontos importantes devem ser salientados sobre SET LINKAGE:

- A ativação de SET LINKAGE só é válida para uma passagem sobre dois arquivos em uso simultâneo. No caso de haver necessidade de outra passagem pelos arquivos, todos os comandos já exemplificados devem ser repetidos.
- Para que funcione a contento, SET LINKAGE deve ser usado em arquivos com idêntico número de registros correspondentes, ainda que não com campos comuns ou afins.

Um dos aspectos interessantes do comando SET LINKAGE é que, com o recurso de que dispõe de possibilitar a exibição simultânea de campos iguais ou diferentes de dois arquivos, o limite de 32 campos por registro imposto pelo dBASE pode ser "superado", em condições especiais e através de soluções imaginosas, elevando esse número até 64. Seria o caso, por exemplo, de criar um arquivo com 64 campos, operando 32 sob um nome e os outros 32 sob outro nome e juntá-los para emissão de relatórios, quando preciso, através do comando SET LINKAGE.

16.3 - FECHAMENTO DE DOIS ARQUIVOS

DBF EM USO SIMULTÂNEO

Vimos anteriormente que o comando USE sem parâmetros fecha arquivos em uso.

Deve-se observar, todavia, que a competência de USE nesse sentido se restringe a APENAS UM arquivo tipo DBF e a TODOS OS SECUNDÁRIOS que se relacionem a ele (NDX, FRM etc.) e que estejam em uso no momento do comando.

O COMANDO USE NÃO TEM PODER PARA FECHAR DOIS ARQUIVOS TIPO DBF EM USO SIMULTÂNEO ATRAVÉS DO COMANDO SELECT.

Se for usado nessa situação, apenas o arquivo selecionado para uso será fechado, permanecendo o não selecionado aberto, o que poderá causar alguma confusão para um operador desavisado.

Assim, DOIS ARQUIVOS TIPO DBF EM USO SIMULTÂNEO ATRAVÉS DO COMANDO SELECT DEVEM SER FECHADOS PELO COMANDO CLEAR, usado sem parâmetros.
Exemplo:

```
CLEAR <RETURN>
```

Arquivos secundários em uso no momento do comando serão igualmente fechados, bem como serão anuladas as variáveis de memória ativas.

O comando CLEAR é detalhado no Capítulo 22.

17 - JUNCÇÃO DE DADOS DE DOIS ARQUIVOS

Um importante aspecto da característica RELACIONAL do dBASE é o que se refere ao comando JOIN.

Comando JOIN

*** JOIN junta - ou concatena - dados de arquivos quando estes possuem campos-chaves comuns ou afins. Sua sintaxe é:

```
JOIN TO <nome do arquivo> FOR <condição>  
[FIELD <nome(s) do(s) campo(s) afim(ns)>]
```

Ao concatenar dois arquivos, o comando JOIN gera um terceiro arquivo, o qual reúne os dados concatenados por intermédio de um campo-chave afim ou de mais de um.

Para que JOIN seja válido, é preciso que os arquivos em junção tenham pelo menos um campo-chave comum.

Tal recurso poderoso do dBASE é muito útil em diversas situações.

Aproveitando a existência do arquivo CTRLSTOQ, vamos simular uma dessas situações.

Admitamos que a loja de departamentos que utilize tal arquivo tenha uma filial que comercialize quase a mesma linha de produtos, com itens diversos e outros idênticos e que, em dado momento, precise efe-

tuar um levantamento de todos os produtos iguais em estoque em ambas as lojas, a fim de igualar os seus preços ou fazer um remanejamento.

Presumiremos que tal filial gerencie seu estoque com um banco de dados (arquivo) idêntico ao da Matriz, o que é muito comum em verdade. Para serem concatenados pelo comando JOIN, não é preciso que os arquivos tenham a mesma estrutura ou que tenham os mesmos campos. Como dissemos antes, é preciso apenas que tenham pelo menos um campo comum ou afim, isto é, idêntico.

Em tal situação, bastará dispor dos disquetes que contenham esses arquivos e concatená-los através do comando JOIN. A título de exemplo, vamos fazê-lo. Em nosso caso, porém, será preciso "produzir" o arquivo da filial referida, o que faremos com facilidade, inicialmente gerando uma cópia integral do próprio arquivo CTRLSTOQ, com o nome CTRLSTFL e, em seguida, alterando alguns itens do mesmo. A cópia poderá ser feita no disquete do sistema ou no de arquivos.

Como fizemos no Capítulo 16, efetuaremos a cópia através do comando COPY do próprio dBASE.

Estando o dBASE ativado e o sistema funcionando conforme indicado no Capítulo 14, comandar:

```
D          SET DEFAULT TO B   <RETURN>
D          USE CTRLSTOQ      <RETURN>
D          COPY TO B:CTRLSTFL <RETURN>
```

Quando retornar o ponto do dBASE, conferir a presença do novo arquivo de dados no disquete, comandando:

```
D          LIST FILES LIKE *.DBF ON B   <RETURN>
```

Uma vez confirmada a verificação, vamos proceder à alteração de alguns itens do novo arquivo:

```
D          USE CTRLSTFL      <RETURN>
D          BROWSE FIELD CODIGO,DENOM   <RETURN>
```

Utilizando os recursos já conhecidos do comando BROWSE, efetuemos as seguintes alterações:

- Registro 2: alteramos seu número de código para 34 e trocamos o AZUL por CINZA.
- Registro 3: alteramos seu número de código para 128 e trocamos CERAMICA por PORCELANA.
- Registro 6: alteramos seu número de código para 1428 e trocamos ESPORTE CAMUR por SOCIAL PRETO.

Os demais dados não precisam ser alterados, já que, para a finalidade pretendida, são suficientes os citados.

Feitas as alterações, pressionamos CONTROL e W simultaneamente, a fim de que as mesmas sejam repassadas efetivamente para o arquivo.

Agora, temos dois arquivos idênticos, mas com dados diferentes, e podemos efetuar sua junção. Para tanto, devemos antes colocar ambos em uso, lembrando que o CTRLSTFL já se encontra nessa condição:

```
D          SELECT SECONDARY  <RETURN>
D          USE CTRLSTO@   <RETURN>
D          JOIN TO CTRLJUNT FOR P.CODIGO = S.CODIGO
          FIELD CODIGO,DEP,DENOM  <RETURN>
```

(P. e S. no comando referem-se a arquivo PRIMARIO e a arquivo SECUNDARIO, como vimos no Capítulo 16.)

Após tais comandos, apenas os produtos de códigos iguais de ambas as lojas estarão reunidos no arquivo CTRLJUNT, com os dados dos campos CODIGO, DEP e DENOM e poderão ser analisados nos dois arquivos de origem, para os fins almejados. Para verificar seu conteúdo, comandar:

```
D          USE CTRLJUNT  <RETURN>
D          LIST  <RETURN>
```

O dBASE responderá:

```
00001 0033 CAM CAMISA TERGAL BRANCA
00002 0345 BAZ JOGO CHA CERAMICA
00003 1234 SAP SAPATO SOCIAL CROMO
```

Este é um dos exemplos mais simples e elucidativos do uso do comando JOIN, assim elaborado propositadamente a fim de facilitar a sua compreensão.

Todavia, sendo um comando complexo e poderoso do dBASE, JOIN permite gerar os mais variados arquivos e, conseqüentemente, os mais diversos resultados e relatórios, devendo ser bastante exercitado a fim de ser dominado.

18 - ARQUIVO DE MOVIMENTO

De modo geral, arquivos de dados são elementos vivos que carecem de atualizações. Dependendo da natureza do arquivo e de sua aplicação, o processo de atualização demanda alterações mais ou menos frequentes, mais ou menos extensas, mais ou menos complexas.

Arquivos de contas a pagar ou a receber e de controles de estoques, por exemplo, são tipos de arquivos de dados que estão mais sujeitos a alterações frequentes, extensas e complexas. Em consequência, operar alterações diretamente nesses arquivos pode apresentar alguns inconvenientes e maximizar os riscos próprios do sistema, em decorrência da acentuada frequência de uso. O principal inconveniente é que, nessa condição de uso, os fatos geradores das alterações de dados são assimilados e diluídos pelo arquivo, sem deixar pistas de rastreamento para eventuais auditorias ou reconstituições dos dados.

Para contornar tal situação, o dBASE oferece um interessante recurso, que é o de efetuar alterações em um arquivo de dados através de um arquivo paralelo. Os fatos geradores de alterações são armazenados no arquivo paralelo, a fim de que as alterações sejam efetuadas no arquivo principal através dele.

Tal recurso apresenta as seguintes conveniências:

- as alterações são efetuadas em lotes no arquivo principal, minimizando riscos de erros e riscos gerais inerentes ao sistema;

- os fatos geradores das alterações ficam armazenados no arquivo paralelo, podendo este ser preservado e servir para eventuais reconstituições do arquivo principal ou para auditorias.

Nessa condição de uso, o arquivo principal de dados é denominado

ARQUIVO MESTRE

e o que contém as alterações a serem efetuadas nele é chamado de

ARQUIVO DE MOVIMENTO ou DE TRANSAÇÕES.

Há dois modos para atualização de um ARQUIVO MESTRE através de um ARQUIVO DE MOVIMENTO:

MODO SEQUENCIAL;

MODO ALEATÓRIO.

No modo SEQUENCIAL, é necessário que OS DOIS ARQUIVOS estejam INDEXADOS através de um CAMPO CHAVE COMUM.

No modo ALEATÓRIO, é preciso que APENAS o ARQUIVO MESTRE esteja INDEXADO pelo CAMPO CHAVE COMUM e que no fim do comando seja especificada a cláusula RANDOM.

Em ambos os casos, os dois arquivos devem ser usados simultaneamente através do comando SELECT, devendo o arquivo MESTRE ser o selecionado.

A atualização de um arquivo mestre através de um arquivo de movimento é feita pelo comando descrito a seguir:

Comando UPDATE

*** UPDATE atualiza arquivo de dados (MESTRE) a partir de um arquivo paralelo (DE MOVIMENTO), estando os dois em uso através do comando SELECT e o arquivo MESTRE sendo o selecionado. Sua sintaxe é:

```

UPDATE FROM <nome do arquivo de movimento>
ON <nome do campo chave comum>
[ADD <nome(s) do(s) campo(s)>]
[REPLACE <nome(s) do(s) campo(s)>]
[WITH <dado(s) ou nome(s) do(s) campo(s)
do arquivo de movimento>]] [RANDOM]

```

Como se depreende de sua sintaxe, UPDATE pode atualizar um arquivo mestre adicionando dados (ADD) ou substituindo-os (REPLACE).

Vamos testar UPDATE atualizando o arquivo CTRLSTOQ através do modo aleatório.

Nossa primeira providência será indexar o arquivo CTRLSTOQ. Vamos fazê-lo pelo campo CODIGO.

Depois de ativado o dBASE conforme Capítulo 14, comandar:

```

D          SET DEFAULT TO B   <RETURN>
D          USE CTRLSTOQ      <RETURN>
D          INDEX ON CODIGO TO CODSTOQ  <RETURN>

```

00006 Registros indexados.

Em seguida, criamos o arquivo de movimento, definindo sua estrutura de modo que possua apenas os campos sujeitos a mudanças mais frequentes, quais sejam: quantidade atual (QTAT), preço de custo (PRC), preço de venda (PRV) e um CAMPO CHAVE COMUM, que será, no caso, o campo CODIGO. Salienta-se ser conveniente dimensionar os campos do arquivo de movimento de acordo com os correspondentes do arquivo mestre.

```

D          CREATE STOGMOV    <RETURN>

```

Ao serem solicitados pelo dBASE os dados para definição de sua estrutura, devemos digitar para os campos de números 1 a 4, respectivamente:

```

D          CODIGO,C,4   <RETURN>
D          QTAT,N,5    <RETURN>
D          PRC,N,8,2   <RETURN>
D          PRV,N,8,2   <RETURN>

```

Ao ser exibido o número 05, teclamos apenas RETURN para encerrar a definição da estrutura do arquivo. Em seguida, poderemos iniciar a entrada de dados.

O campo CODIGO funcionará como CAMPO CHAVE COMUM para casamento dos registros, enquanto que os outros funcionarão como campos afins, por cujo intermédio serão feitos os acréscimos e decréscimos (alterações ou atualizações) nos campos correspondentes do arquivo mestre. Os movimentos de decréscimos deverão ser indicados pelo sinal - (menos) precedendo as quantidades registradas no arquivo de movimento.

A mensagem-pergunta do dBASE:

Incluir dados agora ?

devemos responder Y (= sim).

Inicialmente, vamos utilizar os últimos dados "disponíveis" sobre as vendas efetuadas em um determinado período em que não houve alterações nos preços. Os dados transcritos a seguir devem ser inseridos nas "fichas em branco" que o dBASE apresentará no vídeo. Os campos PRC e PRV não devem ser preenchidos nesta oportunidade.

FICHA	CODIGO	QTAT	PRC	PRV
001	0345	-2		
002	1234	-2		
003	1427	-1		
004	0043	-5		
005	0033	-2		
006	0127	-3		

Encerrada a entrada de dados no arquivo STOQMOV (depois de pres-

sionada a tecla RETURN ao ser exibida a "ficha em branco" número 7), podemos fazer a primeira atualização no arquivo CTRLSTOQ:

```
D          USE CTRLSTOQ INDEX CODSTOQ   <RETURN>
D          SELECT SECONDARY   <RETURN>
D          USE STOQMOV   <RETURN>
D          SELECT PRIMARY   <RETURN>
D          UPDATE FROM STOQMOV ON
D          CODIGO ADD QTAT RANDOM   <RETURN>
```

Após efetuada a atualização, retornará apenas o ponto de prontidão do dBASE. Para verificar as alterações feitas, comandamos:

```
D          LIST CODIGO QTAT   <RETURN>
```

00001	0033	18
00002	0043	25
00003	0127	9
00004	0345	8
00005	1234	13
00006	1427	17

Em seguida, vamos fazer nova atualização no arquivo CTRLSTOQ, baseada em novos preços de custo e de venda para os itens estocados.

Grosso modo, todos os preços serão alterados em 10% a mais, como se deduz da tabela reproduzida a seguir:

0033	495.00	717.75
0043	135.85	196.90
0127	390.50	566.20
0345	465.85	675.40
1234	1357.90	1969.00
1427	1085.70	1574.00

Podemos repassar esses dados para o arquivo STOQMOV, selecionando-o para uso e, em seguida, comandando sua edição através de BROWSE.

Mas, se o editarmos, verificaremos que os dados foram inseridos

nele aleatoriamente, não correspondendo a ordem em que se encontram os códigos com a da tabela acima, ficando assim ligeiramente dificultado o repasse de dados.

O arquivo poderia ser indexado para essa finalidade apenas. Mas, considerada a possibilidade de ter que ser mantido como um "documento" temporário para reconstituição eventual do arquivo mestre ou como uma pista para auditoria, convirá preservá-lo em sua forma original. O nome e a data com que foi gravado servirão de referência para esse fim. (Percebe-se nesse fato uma das razões de atualizar a data de operação do sistema quando o dBASE é ativado.)

Assim, vamos aproveitar apenas a sua estrutura para criar outro arquivo, no qual registraremos os novos preços, utilizando-o em seguida para atualizar o arquivo mestre CTRLSTOQ.

Para aproveitar a estrutura do STOQMOV, comandaremos sua seleção (no momento ele está em uso como secundário, mas não selecionado). Em seguida, faremos uma cópia apenas de sua estrutura através do comando COPY (este é detalhado no Capítulo 21). Ao destinar-se a cópia da estrutura a um novo arquivo, este estará automaticamente criado. Assim, devemos dar ao novo arquivo um nome que seja diferente de seu precedente, mas que mantenha uma lembrança de relação com o mesmo, a fim de facilitar sua identificação e sugerir a seqüência existente:

```
D          SELECT SECONDARY  <RETURN>
D          COPY STRUCTURE TO STMOV-1  <RETURN>
```

Efetuada a cópia, o novo arquivo estará disponível. Vamos usá-lo como arquivo de movimento e inserir nele os dados referentes aos novos preços. Neste caso, deixaremos o campo QTAT em branco:

```
D          USE STMOV-1  <RETURN>
D          APPEND  <RETURN>
```

Depois de digitados todos os dados da tabela acima, comandamos:

```
D          SELECT PRIMARY  <RETURN>
```

para que o arquivo mestre seja selecionado em lugar do STMOV-1.

Em seguida, atualizamos CTRLSTOQ:

```
D          UPDATE FROM STMOV-1 ON CODIGO
          REPLACE PRC,PRV RANDOM   <RETURN>
```

Efetuada a atualização, o ponto de prontidão do dBASE retornará ao vídeo. Verificamos então a exatidão das alterações feitas:

```
D          LIST CODIGO QTAT PRC PRV   <RETURN>
```

00001	0033	18	495.00	717.75
00002	0043	25	135.85	196.90
00003	0127	9	390.50	566.20
00004	0345	8	465.85	675.40
00005	1234	13	1357.90	1969.00
00006	1427	17	1085.70	1574.00

Devemos agora atualizar os valores globais de custo e de venda. A atualização desses dois campos poderia ter sido feita também através do arquivo de movimento, numa só operação, se o tivéssemos criado com a estrutura adequada para esse fim. Porém, com o propósito de facilitar a compreensão de seu mecanismo de ação, preferimos criá-lo simplificado. Mesmo porque a atualização dos campos em questão pode ser feita com facilidade através do comando REPLACE, como veremos a seguir:

```
D          REPLACE ALL VALC WITH PRC * QTAT,
          VALV WITH PRV * QTAT   <RETURN>
```

00006 Substituições.

Os dois testes realizados por nós com o comando UPDATE utilizaram separadamente as cláusulas ADD e REPLACE. Efetuaremos agora um terceiro teste, utilizando as duas cláusulas no mesmo comando.

As últimas versões dos arquivos CTRLSTOQ e STMOV-1 serão usadas, a fim de facilitar as verificações. As alterações a efetuar são as seguintes:

- acréscimo de 5 unidades nos itens de códigos 0127 e 0345;
- aumento geral dos preços de venda em 5%;
- acerto conseqüente do valor total de venda de cada item.

Indicamos a seguir apenas a seqüência de comandos e providências a adotar. O sistema deverá estar funcionando de acordo com as indicações feitas no Capítulo 14. Depois de ativado o dBASE, comandar:

```
D      SET DEFAULT TO B    <RETURN>
D      USE STMOV-1        <RETURN>
D      BROWSE             <RETURN>
      (Nesse ponto, fazer as alterações no campo QTAT.)
      (Feitas as alterações, pressionar CONTROL + W.)
D      REPLACE ALL PRV WITH PRV * 1.05  <RETURN>
D      USE CTRLSTOQ INDEX CODSTOQ      <RETURN>
D      SELECT SECONDARY                 <RETURN>
D      USE STMOV-1                       <RETURN>
D      SELECT PRIMARY                     <RETURN>
D      UPDATE FROM STMOV-1 ON CODIGO
      ADD QTAT REPLACE PRV RANDOM        <RETURN>
D      REPLACE ALL VALV WITH QTAT*PRV   <RETURN>
D      LIST QTAT PRV VALV                <RETURN>
```

00001	18	753.63	13565.34
00002	25	206.74	5168.50
00003	14	594.51	8323.14
00004	13	709.17	9219.21
00005	13	2067.45	26876.85
00006	17	1652.70	28095.90

Se compararmos o conjunto de comandos e providências acima descrito com o tamanho do arquivo atualizado, obviamente parecerá esdrúxula a utilização da informática em tal situação.

Cumpramos ressaltar, todavia, a finalidade puramente didática do arquivo CTRLSTOQ, sendo essa a razão de seu tamanho reduzido.

Por outro lado, devemos lembrar que os últimos sete comandos descritos acima podem ser reunidos em um simples ARQUIVO DE COMANDOS, de modo que sua execução total e automática se efetue através de um único comando, como veremos no capítulo seguinte.

19 - ARQUIVO DE COMANDOS (COMANDO COLETIVO)

Na parte final do Capítulo 18, tomamos contato com um fato freqüente na operação de arquivos de dados: a necessidade de formular uma série de comandos para realizar um determinado processamento.

No caso enfocado, sete comandos diretos foram necessários à execução do processamento desejado.

Dependendo do tipo de atividade e das circunstâncias em que são processados certos arquivos de dados, a necessidade de repetir um processamento que exija muitos comandos pode ser uma necessidade constante e muito freqüente.

Para tais situações, o dBASE oferece a possibilidade de reunir os comandos necessários em um tipo de arquivo especial, de tal sorte que, para sua execução automática, total e seqüencial, basta dar um único comando direto.

Trata-se do ARQUIVO DE COMANDOS, também denominado COMANDO COLETIVO, que recebe a extensão .CMD do dBASE.

Como o nome define, o ARQUIVO DE COMANDOS contém comandos do próprio dBASE, reunidos na ordem em que devem ser executados.

19.1 - CRIAÇÃO DE ARQUIVO DE COMANDOS

Um arquivo de comandos pode ser criado de maneira fácil através do comando MODIFY do dBASE ou do comando COPY CON do MSX DOS. Pode ser criado também através de um processador de texto tipo WORDSTAR. Neste caso e no do comando COPY CON, todavia, o arquivo de comandos deve ser criado com o dBASE desativado, enquanto que o comando MODIFY do dBASE permite que seja criado mesmo com o dBASE ativado, isto é, em uso.

Comando MODIFY

*** MODIFY ativa o EDITOR DE TEXTO embutido no dBASE, o qual permite criar e modificar ARQUIVOS DE COMANDOS, bem como modificar ESTRUTURAS DE ARQUIVOS. Suas sintaxes são:

```
MODIFY COMMAND <nome do arquivo de comandos>  
MODIFY STRUCTURE
```

(A alteração de estruturas de arquivos é tratada no Capítulo 21.)

A título de experiência, criaremos um ARQUIVO DE COMANDOS baseado na série de comandos diretos usados na atualização do arquivo CTRLSTOQ através do arquivo de movimento STMOV-1, conforme descrição feita no final do Capítulo 18. Daremos a esse arquivo o nome CTRLSTAT.

Depois de ativado o dBASE conforme Capítulo 14, comandar:

```
D          SET DEFAULT TO B   (RETURN)  
D          MODIFY COMMAND CTRLSTAT  (RETURN)
```

Dado o comando, o dBASE pesquisará no disquete a existência de um arquivo de nome CTRLSTAT.CMD. Se o encontrar, seu conteúdo será exposto na tela de vídeo, no modo FULL SCREEN ON, podendo o mesmo ser alterado com auxílio das teclas de controle por caractere. Se não encontrar no disquete em uso um arquivo com o nome indicado, emitirá a men-

sagem:

Arquivo novo.

e limpará a tela de vídeo, posicionando o cursor no canto superior esquerdo do vídeo, a fim de que sejam digitados os comandos a serem armazenados no novo arquivo. Nessa condição, a tela de vídeo estará sendo operada também no modo FULL SCREEN ON, com ativação das teclas de controle por caractere do comando MODIFY.

A digitação dos comandos desejados deverá ser feita normalmente, isto é, como se os comandos fossem dados no modo direto: um por vez, teclando-se RETURN ao fim de cada um e seguindo o posicionamento automático do cursor. Nessa condição, isto é, com o comando MODIFY ativado, obviamente os comandos não serão executados. Digitemo-los:

```
D      USE CTRLSTO@ INDEX CODSTO@  <RETURN>
D      SELECT SECONDARY  <RETURN>
D      USE STMOV-1  <RETURN>
D      SELECT PRIMARY  <RETURN>
D      UPDATE FROM STMOV-1 ON CODIGO ADD @TAT
          REPLACE PRC, PRV RANDOM  <RETURN>
D      REPLACE ALL VALC WITH @TAT * PRC,
          VALV WITH @TAT * PRV  <RETURN>
D      LIST CODIGO @TAT PRC
          VALC PRV VALV  <RETURN>
D      CLEAR  <RETURN>
```

Terminada a digitação dos comandos, pressionar simultaneamente as teclas

CONTROL e W

a fim de que os comandos digitados sejam transferidos para o arquivo CTRLSTAT e este gravado no disquete. A verificação da presença dele no disquete poderá ser feita através do comando DISPLAY FILES LIKE *.*.

Podemos observar que, em relação aos comandos utilizados antes no modo direto, introduzimos pequenas modificações nos 50, 60 e 70 coman-

dos, visando torná-los mais abrangentes e definitivos. Acrescentamos também o 8º comando, a fim de fechar os dois arquivos em uso ao final do processamento.

19.2 - EXECUÇÃO DE ARQUIVO DE COMANDOS

Depois de reunidos em ARQUIVO DE COMANDOS, comandos do dBASE são executáveis automática e seqüencialmente, através do comando DO.

Comando DO

*** DO executa comandos ou programas do dBASE agrupados em arquivos específicos, com extensão .CMD. Sua sintaxe é:

```
DO <nome do arquivo de comandos  
ou do arquivo de programa>
```

Assim, sempre que se desejar executar automaticamente os comandos todos reunidos no arquivo CTRLSTAT, bastará comandar:

```
D          DO CTRLSTAT  <RETURN>
```

19.3 - ALTERAÇÃO OU CORREÇÃO

DE ARQUIVO DE COMANDOS

O conteúdo de um arquivo de comandos pode ser alterado ou corrigido a qualquer momento através do comando MODIFY usado com a cláusula COMMAND.

Como afirmamos antes, MODIFY opera no modo FULL SCREEN ON, sendo válidas durante sua ativação as teclas de controle por caractere indicadas a seguir:

TECLAS DE CONTROLE DO COMANDO MODIFY

- CONTROL+D - Movem o cursor um espaço à direita.
- CONTROL+S - Movem o cursor um espaço à esquerda.
- CONTROL+E - Movem o cursor para a linha ou para o campo anteriores.
- CONTROL+X - Movem o cursor para a linha ou para o campo seguintes.
- CONTROL+Y - Apagam a linha ou campo na posição do cursor.
- CONTROL+N - Inserem uma linha em branco na posição do cursor.
- CONTROL+T - Apagam a linha na posição do cursor, movendo as seguintes para cima.
- CONTROL+V - Ligam/desligam o modo de inserção.
- CONTROL+G - Apagam o caractere sob o cursor.
- CONTROL+Q - Encerram o comando sem gravar alterações eventualmente feitas.
- CONTROL+W - Encerram o comando e regravam o arquivo com as alterações feitas.
- RETURN - Desloca o cursor para a próxima linha ou para o próximo campo.

Para alterar o conteúdo de um arquivo de comandos, deve-se editá-lo através do comando MODIFY COMMAND, digitando-se após este o nome do arquivo e teclando-se RETURN. Exemplo:

```
MODIFY COMMAND NOMEARQ <RETURN>
```

Dado o comando, o conteúdo do arquivo de comandos será exposto na tela de vídeo, no modo FULL SCREEN ON, a fim de receber alterações ou correções.

A título de experiência, vamos introduzir o comando ERASE no início do arquivo CTRLSTAT, a fim de que a execução dos comandos contidos no mesmo seja precedida de uma "limpeza" na tela de vídeo:

```
D MODIFY COMMAND CTRLSTAT <RETURN>
```

O dBASE exporá no vídeo todo o conteúdo do arquivo, com o cursor posicionado no primeiro caractere da primeira linha. Nessa condição:

- pressionar simultaneamente as teclas

CONTROL e N;

- digitar na linha em branco criada:

ERASE

- pressionar simultaneamente as teclas

CONTROL e W.

Feito isso, o arquivo CTRLSTAT será regravado com o novo comando acrescido.

Caso seja preciso acrescentar um novo comando após o 3º já existente, por exemplo, deverá ser adotado o seguinte procedimento:

- deslocar o cursor até a posição do 4º comando, com auxílio das teclas CONTROL+X;
- pressionar em seguida a tecla CONTROL+N;
- digitar o comando desejado na linha em branco criada;
- pressionar as teclas CONTROL+W para confirmar a alteração.

As teclas de controle por caractere do comando MODIFY podem ser visualizadas com rapidez no seu INDICADOR VISUAL, no item 7.8.

20 - ARQUIVO ALTERNATIVO OU ARQUIVO DE FATOS

Vimos antes que é possível reproduzir através da impressora tudo o que é exibido no vídeo pelo dBASE, bastando para tanto pressionar as teclas CONTROL e P simultaneamente ou comandar-se SET PRINT ON.

Além disso, pode-se também armazenar em arquivo gravado em disquete tudo o que é exibido no vídeo pelo dBASE durante sua ativação, inclusive os comandos digitados pelo operador.

Trata-se do ARQUIVO ALTERNATIVO ou ARQUIVO DE FATOS do dBASE, que pode ser utilizado para diversas finalidades. Uma delas, por exemplo, é registrar toda uma sessão com o dBASE, ou partes dela, para fins de reprodução impressa posterior de certos dados, ou para sua repetição.

O arquivo alternativo é um arquivo considerado como de TEXTO, recebendo automaticamente a extensão .TXT ao ser gravado, se outra não lhe for dada pelo operador.

Como arquivo de texto, é armazenado em códigos ASCII e pode ser exibido no vídeo - fora do dBASE - pelo comando TYPE do MSX DOS e concomitantemente reproduzido pela impressora através do sistema eco ativado, pressionando-se as teclas CONTROL + P.

Um arquivo alternativo do dBASE pode também ser processado pelo WORDSTAR ou outro processador de texto similar. Em tal caso, por exemplo, um relatório de dados de um arquivo DBF dBASE pode ser facilmente incluído num texto de carta ou de reportagem elaborado através do pro-

cessador de texto.

Cria-se um arquivo alternativo pelo comando descrito a seguir:

Comando SET ALTERNATE TO

*** SET ALTERNATE TO cria e grava arquivo em disco para armazenamento dos fatos exibidos em vídeo durante a ativação do dBASE. Sua sintaxe é:

SET ALTERNATE TO <nome do arquivo>

Depois de comandada a criação do arquivo, o armazenamento dos fatos nele é controlado pelo comando descrito a seguir:

Comando SET ALTERNATE ON/OFF

*** SET ALTERNATE ON ativa o armazenamento de fatos em ARQUIVO ALTERNATIVO, enquanto que

*** SET ALTERNATE OFF desativa tal armazenamento.

Deduz-se da descrição dos comandos ser fácil selecionar os fatos que se deseja armazenar em um arquivo alternativo durante uma sessão com o dBASE, pois, estando SET ALTERNATE em ON depois de o arquivo ter sido criado, tudo o que for exibido na tela de vídeo será armazenado nele, cessando tal condição ao dar-se o comando SET ALTERNATE OFF.

Deve-se observar que apenas as telas - ou fatos - produzidos no vídeo pelos comandos APPEND, BROWSE e EDIT não são armazenáveis em arquivo alternativo.

21 - LIDANDO COM ARQUIVOS

Até este ponto de nosso curso estudamos comandos do dBASE destinados a manipular registros de arquivos.

Neste capítulo, estudaremos os comandos do dBASE que permitem manipular arquivos como unidades globais: cada arquivo como um todo.

21.1 - COPIAÇÃO DE ARQUIVO

E DE ESTRUTURA DE ARQUIVO

Com o dBASE ativado, é possível efetuar cópias apenas de arquivos específicos de dados, isto é, arquivos que tenham a extensão .DBF nos nomes. Outros tipos de arquivos podem ser copiados fora do dBASE apenas, isto é, com o dBASE desativado.

Comando COPY

*** COPY efetua cópia global ou parcial de arquivo de dados (DBF). Só é válido para arquivo em uso.

Embora homônimo de comandos do MSX DISC BASIC e do MSX DOS, o comando COPY do dBASE tem funções diferentes dos mesmos.

COPY do dBASE possibilita:

- copiar todo o arquivo ou parte dele apenas, na sua forma usual;
- copiar todo o arquivo ou parte dele apenas, convertendo-o em formato que possa ser utilizado por outras linguagens de programação como BASIC, COBOL etc., ou outros programas aplicativos como WORDSTAR, por exemplo;
- copiar apenas campos determinados de registros;
- copiar apenas registros que atendam condições especificadas;
- copiar somente a estrutura do arquivo ou parte dela.

A sintaxe genérica do comando COPY é:

```
COPY [(escopo)] TO <nome do novo
      arquivo> [FIELD <nome(s) do(s)
               campo(s)>] [FOR <condição>]
```

Para copiar estrutura de arquivo ou parte dela, sua sintaxe é:

```
COPY STRUCTURE TO <nome do novo ar-
                  quivo> [FIELD <nome(s) do(s) campo(s)>]
```

Quando diversos campos são especificados no comando, seus nomes devem ser separados por vírgulas.

Exemplos de comandos COPY:

```
USE CTRLSTO@ <RETURN>
COPY TO CTRLSTFL <RETURN>
```

Quando se deseja copiar arquivo de disquete do disk drive A para disquete do disk drive B, sendo este o drive "default", comanda-se:

```
USE CTRLSTO@ <RETURN>
COPY A=CTRLSTO@ TO B=CTRLSTO@ <RETURN>
```

*** Para copiar arquivo no mesmo disquete, o nome do arquivo-cópia

deve ser diferente:

```
COPY CTRLSTOQ TO CTRLNOVO <RETURN>

USE CTRLSTOQ <RETURN>
COPY TO CTRLSTPT FOR DEP = "CAM" <RETURN>
```

Nesse caso, todos os registros do arquivo CTRLSTOQ cujos campos DEP tenham o conteúdo "CAM" serão copiados ao novo arquivo CTRLSTPT.

```
USE CTRLSTOP <RETURN>
COPY TO CTRLSTAT FIELD
        CODIGO,QTAT,PRC,PRV <RETURN>
```

Nesse exemplo, os campos CODIGO, QTAT, PRC e PRV, com os conteúdos respectivos, serão copiados do arquivo CTRLSTOQ para o novo arquivo CTRLSTAT.

```
USE CTRLSTOQ <RETURN>
COPY TO CTRLNOVO FIELD CODIGO,DENOM,
        QTMIN,QTAT FOR QTAT < QTMIN <RETURN>
```

Com este comando COPY, os campos especificados e respectivos conteúdos de todos os registros do arquivo CTRLSTOQ cujo conteúdo do campo QTAT seja menor do que o conteúdo do campo QTMIN serão copiados para o novo arquivo CTRLNOVO.

```
USE CTRLTOT <RETURN>
GO 100 <RETURN>
COPY NEXT 100 TO CTRLTOT2 <RETURN>
```

Através destes comandos, 100 registros do arquivo CTRLTOT, a partir do registro de número 100, serão copiados para o arquivo CTRLTOT2.

CÓPIA DE ESTRUTURA DE ARQUIVO OU DE PARTE DELA:

```
USE CTRLSTOQ <RETURN>
COPY STRUCTURE TO CTRLSTFL <RETURN>
```

Nesse caso, a estrutura completa do arquivo CTRLSTOQ será copiada para o novo arquivo CTRLSTFL.

```
USE CTRLSTOQ <RETURN>
COPY STRUCTURE TO CTRLSTES FIELD CODIGO,
          QTAT,DENOM,VALC,VALV <RETURN>
```

Nesse exemplo será copiada do arquivo CTRLSTOQ para o novo arquivo CTRLSTES apenas a estrutura dos campos especificados.

21.2 - MUDANÇA DE NOMES DE ARQUIVOS

RENAME é o comando do dBASE destinado a mudar nomes de arquivos.

Comando RENAME

*** RENAME muda ou altera nome de arquivo gravado em disquete.

O comando RENAME é válido para qualquer arquivo do dBASE que não esteja em uso ou para qualquer outro do disquete. Sua sintaxe é:

```
RENAME <nome atual do arquivo> TO
          <novo nome do arquivo>
```

As extensões dos nomes (atual e novo) devem ser especificadas no comando. Exemplo:

```
RENAME CTRLSTOQ.NDX TO
          CTRLINDX.NDX <RETURN>
```

Se o arquivo estiver em uso no momento do comando, o dBASE emitirá a mensagem:

Arquivo no momento está aberto.

Em tal situação, o arquivo deverá ser fechado com o comando USE:

USE <RETURN>

e o comando RENAME deverá ser repetido.

21.3 - ELIMINAÇÃO DE ARQUIVOS DO DISQUETE

O comando do dBASE para eliminar arquivos do disquete é o

Comando DELETE FILE

*** DELETE FILE elimina o arquivo especificado do disquete. Sua sintaxe é:

DELETE FILE <nome do arquivo>

No dBASE II PLUS, o arquivo a ser eliminado deverá estar fechado. Um comando dado sobre arquivo aberto gerará a mensagem:

Arquivo no momento está aberto.

Em tal caso, o arquivo deverá ser fechado através do comando USE e o comando DELETE deverá ser repetido.

Depois de eliminado o arquivo, é exposta no vídeo a mensagem:

Arquivo foi eliminado.

21.4 - LISTAGEM DE ARQUIVOS DO DISQUETE

Os nomes de arquivos existentes no disquete em uso podem ser listados no vídeo ou pela impressora. Os comandos do dBASE utilizados para esse fim são:

Comandos DISPLAY e LIST

Para se obter a listagem dos nomes dos arquivos com as extensões

.DBF, dá-se um dos comandos seguintes:

```
                DISPLAY FILES  <RETURN>
ou              LIST FILES    <RETURN>
```

Para se obter a listagem de todos os nomes de arquivos existentes no disquete, comanda-se:

```
                DISPLAY FILES LIKE *.*  <RETURN>
ou              LIST FILES LIKE *.*    <RETURN>
```

Pode-se obter a listagem de nomes de arquivos de um tipo específico, como de arquivos de índice, por exemplo, especificando-se o asterisco no lugar do nome e a extensão característica do tipo depois do do ponto. Exemplo:

```
                LIST FILES LIKE *.NDX  <RETURN>
```

A listagem de nomes de arquivos de disquete alojado em disk drive não corrente é obtida indicando-se o drive no fim do comando. No caso de o disk drive corrente ou "default" ser o B e desejar-se a listagem de todos os nomes de arquivos do disquete do drive A, o comando a ser dado é:

```
                LIST FILES LIKE *.* ON A  <RETURN>
```

21.5 - MODIFICAÇÃO DE ESTRUTURA DE ARQUIVO

Há situações em que, por razões diversas, um arquivo de dados deve ter sua estrutura modificada, a fim de atender necessidades ou conveniências como:

- aumentar ou diminuir tamanho de campo(s);
- eliminar ou inserir campo(s);
- etc.

No dBASE, tais alterações podem ser feitas através do comando indicado a seguir:

Comando MODIFY STRUCTURE

*** MODIFY STRUCTURE possibilita modificar a estrutura de arquivo de dados. Sua sintaxe é, simplesmente:

MODIFY STRUCTURE

Seu comando só é válido sobre um arquivo em uso.

A possibilidade de alterar a estrutura de arquivos de dados é um dos recursos de grande valia do dBASE. Todavia, deve ser utilizado com atenção, já que, ao ser executado, MODIFY STRUCTURE esvazia todo o arquivo em uso, fazendo-o, todavia, após obter do operador consentimento à seguinte mensagem que emite através do vídeo:

MODIFY apaga todos os registros de dados. Prossegue (Y/N) ?

Assim, para que sua utilização atinja seus objetivos de maneira acertada, um procedimento adequado deve ser adotado, como exemplificaremos em seguida:

A título de experiência, vamos modificar a estrutura de nosso arquivo PESSOAL, de modo a acrescentar-lhe dois novos campos e aumentar o tamanho do campo SALARIO - já existente - de 5 para 10 espaços, com duas casas decimais. Os novos campos serão:

ESTCIV,C,8
DEPEND,N,2

destinando-se a armazenar informações sobre o estado civil e dependentes dos cadastrados.

Depois de ativado o dBASE, comandamos:

USE PESSOAL <RETURN>

Em seguida, e como primeira providência visando resguardar o conteúdo do arquivo PESSOAL, copiamos sua estrutura para um novo arquivo:

```
COPY STRUCTURE TO PESNOVO <RETURN>
```

Feito isso, colocamos o novo arquivo em uso, a fim de poder modificar a estrutura deste, em vez da do PESSOAL:

```
USE PESNOVO <RETURN>  
MODIFY STRUCTURE <RETURN>
```

Nesse estágio, o dBASE adverte:

```
MODIFY apaga todos os registros de  
dados. Prossegue (Y/N) ?
```

(Tal mensagem nos faz compreender a razão de não estarmos trabalhando diretamente com a estrutura do arquivo PESSOAL.)

Como o arquivo PESNOVO é um arquivo "vazio", respondemos:

Y

(Nesse ponto, se o arquivo em uso fosse o PESSOAL, todos os seus registros teriam sido destruídos.)

Em seguida, o dBASE expõe no vídeo:

	Nome	Tip.	Tam.	Dec.	
Campo 01	=NOME	C	015	000	:
CAMPO 02	=CARGO	C	010	000	:
Campo 03	=SALARIO	N	005	000	:
Campo 04	=DATADM	C	005	000	:
Campo 05	:				:
Campo 06	:				:
Campo 07	:				:
Campo 08	:				:
...					

...

Campo 32 :

:

Como o comando MODIFY opera no modo FULL SCREEN ON, as alterações desejadas poderão ser efetuadas diretamente no vídeo, com auxílio das teclas de controle por caractere (Consultar itens 7.8 e 19.3).

Depois de feitas as alterações, pressionar

CONTROL e W

simultaneamente, a fim de que sejam efetivadas e a estrutura do arquivo seja regravada no disquete.

Em seguida, verificamos a nova estrutura, comandando:

DISPLAY STRUCTURE <RETURN>

O quadro a ser exibido no vídeo é reproduzido a seguir:

Estrutura para o arquivo :B:PESNOVO .DBF
Data da última atualização :18/11/87
Uso primário do banco de dados.

Cpo	Nome	Tipo	Tam.	Dec.
001	NOME	C	015	
002	CARGO	C	010	
003	SALARIO	N	010	002
004	DATADM	C	005	
005	ESTCIV	C	008	
006	DEPEND	N	002	
** TOTAL **			00051	

Conferida a estrutura, as seguintes providências devem ser tomadas, a fim de completar a operação:

- transferir os dados do arquivo PESSOAL para o arquivo PESNOVO;
- eliminar em seguida o arquivo PESSOAL do disquete;
- mudar o nome do arquivo PESNOVO.DBF para PESSOAL.DBF;

- atualizar os registros do arquivo PESSOAL, inserindo dados nos novos campos criados;
- atualizar o arquivo PESSOAL.FRM em função dos novos campos.

No elenco de providências acima citadas, devemos destacar a primeira delas, que exige a introdução de um novo comando:

Comando APPEND FROM

*** APPEND FROM permite transferir registros de um arquivo de dados do disquete para outro que esteja em uso. Sua sintaxe é:

```
APPEND FROM <nome do arquivo-fonte>  
[FOR <condição>]
```

Como se depreende de sua sintaxe, o arquivo-fonte todo pode ser considerado para transferência de dados para o arquivo em uso ao ser dado o comando, ou apenas os dados de registros que satisfaçam a uma condição determinada.

Os arquivos envolvidos na operação de transferência podem ter estruturas diferentes. Apenas os conteúdos dos campos com nomes idênticos serão transferidos do arquivo-fonte para o arquivo em uso, sendo conveniente que os tamanhos dos campos respectivos sejam equivalentes. Se os dados em transferência ocuparem espaço maior do que o campo que lhes é destinado no arquivo em uso, serão passados para a(s) linha(s) seguinte(s), com prejuízo da estética de reprodução.

Deve-se salientar que o arquivo-fonte não sofre quaisquer alterações durante o processo, já que os dados envolvidos são apenas copiados dele para o novo arquivo.

Descrito o novo comando, podemos usá-lo e começar a adotar as citadas providências:

```
D          APPEND FROM PESSOAL  <RETURN>
```

```
00006 Registros adicionados.
```

D DELETE FILE PESSOAL.DBF <RETURN>

Arquivo foi eliminado.

D USE <RETURN>

D RENAME PESNOVO.DBF TO
PESSOAL.DBF <RETURN>

D USE PESSOAL <RETURN>

Em seguida, inserir os dados transcritos abaixo nos campos novos dos registros, respectivamente, usando um dos comandos

EDIT ou BROWSE

e as teclas de controle por caractere correspondentes.

Dados para complementação do arquivo PESSOAL:

REGISTRO	ESTCIV	DEPEND
00001	Casado	2
00002	Solteiro	-
00003	Casada	-
00004	Solteira	-
00005	Casado	3
00006	Casado	-

Completada a operação, os dados dos novos campos poderão ser verificados com o comando:

D LIST NOME, ESTCIV, DEPEND <RETURN>

00001	Marco Antonio	Casado	2
00002	Luis Carlos	Solteiro	0
00003	Maria Cecilia	Casada	0
00004	Maria Beatris	Solteira	0
00005	Julio Cesar	Casado	3
00006	Paulo Sergio	Casado	0

Resta agora adotar a última providência, qual seja a atualização do arquivo de formato de relatório, PESSOAL.FRM, a fim de que os novos campos adicionados ao arquivo PESSOAL.DBF sejam reproduzidos devidamente em relatórios emitidos sobre o arquivo.

A modificação de um arquivo de formato de relatório pode parecer um pouco mais complicada do que a de um arquivo de comandos, em consequência de ser constituído essencialmente pelas respostas dadas pelo operador ao dBASE quando foi organizado.

Assim, ao comandarmos sua edição para modificação:

```
D          MODIFY COMMAND PESSOAL.FRM  (RETURN)
```

veremos seu conteúdo exposto no vídeo na seguinte forma:

```
Y
RELATORIO DE EMPREGADOS
N
Y
N
16,NOME
NOME
11,CARGO
CARGO
6,SALARIO
SAL.
Y
6,DATADM
ADMIS
```

Podemos ver que aí estão exatamente as respostas e elementos de informação que digitamos quando, depois de comandada a emissão de um relatório do arquivo PESSOAL pela primeira vez, o dBASE formulou através do vídeo as questões necessárias à organização do arquivo de formato criado automaticamente por ele para essa finalidade.

Nessa situação, isto é, editado no modo de tela FULL SCREEN ON, o arquivo de formato PESSOAL.FRM poderá ser modificado facilmente, bas-

tando que se tenha em conta que as alterações a serem feitas deverão ater-se às normas de sua organização e formatação, como se fossem respostas às questões normalmente formuladas pelo dBASE para esse fim.

Em nosso caso, devemos apenas alterar o tamanho do campo SALARIO e acrescentar no final do arquivo os dois novos campos criados, o que deverá ser feito de acordo com as indicações dadas a seguir e com auxílio das teclas de controle por caractere do comando MODIFY (consultar o item 7.8, se houver dúvida):

- inicialmente, usando as teclas CONTROL + X, deslocar o cursor para a linha

6,SALARIO

- com o cursor posicionado sobre o número 6, pressionar as teclas

CONTROL + G

para deletá-lo, isto é, eliminá-lo.

- em seguida, pressionar as teclas

CONTROL + V

para ativar o modo de inserção do dBASE;

- digitar então

11 <RETURN>

- com as teclas CONTROL + X, deslocar o cursor para a primeira linha após a última do arquivo (ADMIS);

- digitar a partir dessa posição do cursor:

9,ESTCIV <RETURN>

EST CIV <RETURN>

3,DEPEND <RETURN>

DEP <RETURN>

- pressionar finalmente as teclas

CONTROL + W

para regravar o arquivo de formato com as alterações feitas.

Após essas providências, o arquivo PESSOAL.FRM estará atualizado em função da nova estrutura do arquivo PESSOAL.DBF e poderá ser usado normalmente para emitir relatórios.

21.6 - CONVERSÃO DE ARQUIVOS dBASE

PARA ARQUIVOS NÃO dBASE E VICE-VERSA

A fim de possibilitar o processamento de seus arquivos por outros programas gerenciadores e também poder processar arquivos gerados por tais programas, o dBASE dispõe de recursos que lhe permitem gerar cópias de arquivos com formatos convertidos.

Normalmente, arquivos de dados criados por programas ou linguagens não dBASE têm formato e características definidos pelo SDF (SYSTEM DATA FORMAT) dos sistemas operacionais CP/M, MS DOS e MSX DOS.

No padrão SDF observa-se que:

- os registros de ARQUIVOS SEQUENCIAIS não têm obrigatoriamente tamanhos fixos e são separados pelos códigos 0DH e 0AH, correspondentes a RETORNO DE CURSOR (CARRIAGE RETURN) e AVANÇO DE LINHA (LINE FEED), sendo os campos de cada registro SEPARADOS por vírgulas. Os campos alfanuméricos são DELIMITADOS com aspas duplas (") - em alguns casos com aspas simples (') - e os campos numéricos não são delimitados;
- os registros de ARQUIVOS DE ACESSO DIRETO têm obrigatoriamente tamanho fixo. Registros e campos não são separados nem delimitados por quaisquer códigos ou caracteres. O controle sobre os mesmos é exercido pelos sistemas operacionais em função de suas extensões, estabelecidas na declaração LEN da instrução OPEN e na instrução FIELD de programas em MSX DISC BASIC ou por instruções equivalentes de outras linguagens;
- em ambas as modalidades de arquivos, o fim do arquivo é assinalado com o código 1AH (EOF - end of file).

21.6.1 - CONVERSÃO DE ARQUIVOS dBASE

EM ARQUIVOS NÃO dBASE

Ao ser efetuada a conversão, o arquivo original não sofre nenhuma alteração, já que o mesmo é apenas lido no disquete pelo microcomputador, que o processa e converte em sua memória, transferindo-o em seguida - já convertido - para o novo arquivo designado no comando dado para conversão e gravando-o automaticamente no disquete ao encerrar a operação.

Para obter arquivos de dados convertidos do dBASE nas duas modalidades acima descritas, é usado o comando COPY com as cláusulas específicas para a finalidade:

```
DELIMITED [WITH] <caractere do limitador>  
SDF
```

e com a sintaxe descrita a seguir:

```
COPY TO <nome do arquivo> [(escopo)]  
[FIELD <nome(s) do(s) campo(s)>] [FOR  
<condição>] [DELIMITED [WITH  
<caractere do delimitador>]] [SDF]
```

Por exemplo, desejando-se obter do arquivo CTRLSTOQ.DBF uma cópia convertida em formato de arquivo de ACESSO SEQUENCIAL que possa ser utilizado por um programa gerenciador de arquivos em BASIC, através do MSX DISC BASIC, comanda-se:

```
USE CTRLSTOQ <RETURN>  
COPY TO nome DELIMITED WITH " <RETURN>
```

Deve ser indicado como delimitador de campos alfanuméricos o sinal de aspas duplas ("), em razão de ser o delimitador usado pelo sistema MSX DISC BASIC.

Outro delimitador poderá ser usado, em conformidade com o adotado pelo sistema operacional ou programa aplicativo a que se destina o arquivo em conversão.

Se a cláusula DELIMITED for usada sem parâmetros, o arquivo convertido terá os campos normalmente separados por vírgulas, mas os campos alfanuméricos serão delimitados apenas com aspas simples (').

Para obter do mesmo arquivo CTRLSTOQ.DBF uma cópia convertida em formato de arquivo de ACESSO ALEATÓRIO que possa ser processado através do MSX DISC BASIC ou outro sistema, comanda-se:

```
USE CTRLSTOQ   <RETURN>
COPY TO nome SDF   <RETURN>
```

Em ambos os casos, os arquivos convertidos através desse processo recebem a extensão .TXT e são considerados também como arquivos dessa modalidade, podendo ser lidos e processados por quaisquer processadores de textos que manipulem os registros como seqüências de caracteres ASCII.

21.6.2 - CONVERSÃO DE OUTROS ARQUIVOS

EM ARQUIVOS dBASE

A conversão de arquivos de outros formatos em arquivos dBASE é um pouco mais complicada, pois a conversão direta só é possível se o arquivo estiver na modalidade SEQUENCIAL.

Assim, para converter um arquivo de ACESSO DIRETO em arquivo padrão dBASE, o mesmo deverá ser antes convertido em arquivo SEQUENCIAL, o que terá que ser feito fora do dBASE. Um pequeno programa em BASIC poderá ser escrito para esse fim, se não estiver disponível outro programa conversor.

Estando o arquivo de ACESSO ALEATÓRIO convertido em SEQUENCIAL, ele estará apto a ser convertido em arquivo padrão dBASE.

O comando utilizado para converter um arquivo SEQUENCIAL em arquivo padrão dBASE é o APPEND com uma das cláusulas SDF ou DELIMITED.

Todavia, antes de ser dado tal comando, deverá ser criado um arquivo dBASE com a estrutura adequada para receber os dados convertidos do arquivo de outro formato. Usar-se-á para esse fim o comando CREATE.

Criado e posto em uso o arquivo para receber os dados convertidos do arquivo sequencial, deverá ser dado o comando

```
APPEND FROM <nome do arquivo sequencial>  
DELIMITED WITH <RETURN>
```

se o arquivo SEQUENCIAL a ser convertido em arquivo dBASE já era originalmente dessa modalidade, isto é, que não tenha sido convertido da modalidade de ACESSO DIRETO para SEQUENCIAL.

No caso de o arquivo a ser convertido em arquivo dBASE ter sido previamente convertido da modalidade de ACESSO DIRETO em modalidade SEQUENCIAL através de um programa em BASIC, o comando a ser dado é:

```
APPEND FROM <nome do arquivo  
sequencial> SDF <RETURN>
```

Com relação à cláusula DELIMITED, em se tratando de converter arquivos de outros formatos em arquivos dBASE, frisamos que sua sintaxe se restringe ao exemplo utilizado, isto é, não podem ser especificados delimitadores como parâmetros.

Se ocorrer que os campos de caracteres forem delimitados por símbolos que não sejam aspas, os delimitadores serão transferidos para o novo arquivo como partes integrantes dos dados.

Duas observações finais:

- as conversões de arquivos de ACESSO DIRETO em arquivos dBASE (depois de previamente convertidos em SEQUENCIAIS) geralmente não apresentam problemas porque as dimensões fixas de seus campos - e conseqüentemente de seus registros - permitem criar uma

estrutura adequada para o arquivo dBASE que receberá os dados em formato convertido;

- as conversões de arquivos originalmente SEQUENCIAIS em arquivos dBASE podem apresentar alguns problemas de truncamento de dados, se os campos dos registros do novo arquivo não forem dimensionados com folga suficiente para absorver as variações de extensão que normalmente se verificam nos dados de arquivos sequenciais.

22 - CONSTANTES E

VARIÁVEIS DE MEMÓRIA

No dBASE, como em computação em geral, números e conjuntos de caracteres definidos são considerados CONSTANTES e podem ser representados por nomes criados pelo usuário. Por exemplo: 123, -11, 219.75 são CONSTANTES NUMÉRICAS. CASA, LÁPIS Nº 1, RELÓGIO são CONSTANTES ALFANUMÉRICAS.

Uma constante alfanumérica como "Brasília é a moderna Capital do Brasil e foi inaugurada em 1960." pode ser representada por nome como CAP, por exemplo, e ser armazenada na memória do microcomputador por intermédio desse nome, de modo que, quando se deseja reproduzi-la pelo vídeo ou pela impressora, não seja preciso digitá-la novamente, bastando dar ao computador o comando PRINT CAP.

Do mesmo modo, números ou expressões numéricas podem ser armazenados e processados pelo computador através de simples nomes. Se o número 2345.87 for representado pelo nome SM e o número 12 pelo nome M, podemos comandar ao computador que efetue operações aritméticas com os mesmos sem precisar digitá-los, bastando indicar os seus nomes no comando de operação. Por exemplo, para obter a multiplicação dos números exemplificados, comanda-se PRINT SM * M.

Supondo-se que o número 2345.87 retrate o Piso Nacional de Salário num período e que o nome SM seja usado em programas de computador para representá-lo, todas as vezes que o referido Piso sofrer alterações, SM passará a representar sucessivamente valores diferentes.

Nessas condições, o nome SM passa a ser um nome-símbolo representante de CONSTANTES numéricas ou VALORES mutáveis.

No dBASE, em analogia com a Matemática e com a Computação, quando nomes-símbolos representam números ou conjuntos de caracteres mutáveis (palavras, frases etc.), tais nomes-símbolos são denominados VARIÁVEIS e classificados como tais.

VARIÁVEIS, pois, são elementos que assumem sucessivamente valores ou expressões alfanuméricas diferentes no decurso de um cálculo ou de um programa de computação.

No dBASE, as variáveis são classificadas em três tipos:

NUMÉRICAS

ALFANUMÉRICAS (ou "STRINGS")

LÓGICAS

Os NOMES das variáveis podem ser constituídos por até 10 caracteres, devendo começar obrigatoriamente por uma letra.

Variáveis NUMÉRICAS podem representar números com até 10 dígitos e podem ser processadas aritmeticamente pelo computador através apenas de seus nomes.

Variáveis ALFANUMÉRICAS podem representar expressões alfanuméricas constituídas por até 254 caracteres e podem ser concatenadas pelo computador também apenas através de seus nomes.

As variáveis LÓGICAS representam apenas as duas letras designativas de suas funções: F (= FALSE = falso) e T (= TRUE = verdadeiro).

No dBASE, variáveis podem ter nomes idênticos a campos de registros, mas estes terão prioridade nas execuções de comandos.

O dBASE II PLUS permite armazenar até 64 variáveis na memória do computador. Nessa condição, isto é, quando armazenadas no computador, são denominadas VARIÁVEIS DE MEMÓRIA.

22.1 - CRIAÇÃO E INICIALIZAÇÃO DE VARIÁVEIS

Através do dBASE, e também em analogia com outras linguagens de computação, variáveis de memória são criadas pelos mesmos comandos que permitem fazer-lhes atribuições.

Variáveis podem ser criadas ou inicializadas em modo direto ou em modo indireto.

22.1.1 - CRIAÇÃO E ATRIBUIÇÃO DE VARIÁVEIS

EM MODO DIRETO

Comando STORE

*** STORE cria uma variável ou mais de uma e permite atribuir-lhe(s) uma expressão numérica ou alfanumérica. Sua sintaxe é:

```
STORE <atribuição> TO <nome(s)>  
                                da(s) variável(is))
```

Depreende-se que uma mesma atribuição pode ser feita a diversas variáveis com um só comando.

Exemplos de comandos:

```
STORE 219 TO N <RETURN>  
STORE "BRASIL GRANDE" TO BG <RETURN>
```

*** Números são atribuídos a variáveis sem aspas como delimitadores.

*** Expressões alfanuméricas devem ser atribuídas a variáveis com delimitadores, podendo ser aspas simples (') ou duplas (").

O comando STORE permite também que se armazene o(s) conteúdo(s) de campo(s) de registro em variável(is) de memória. Exemplo:

STORE <nome(s) do(s) campo(s)>
TO <variável(is) de memória>

Quando mais de um campo for designado no comando, os mesmos devem ser separados pelo sinal de adição (+).

22.1.2 - CRIAÇÃO E ATRIBUIÇÃO DE VARIÁVEIS EM MODO INDIRETO

Atribuições indiretas a variáveis podem ser feitas através de comandos que as solicitem apenas após sua execução. Tais comandos permitem que comentários elucidativos sejam incluídos em sua execução. São eles: INPUT e ACCEPT.

Comando INPUT

*** INPUT cria variável e permite atribuir-lhe dado numérico ou alfanumérico digitado após sua execução. INPUT permite a inclusão de comentário a ser exibido no vídeo no momento que precede a digitação dos dados, de modo a informar ao operador a finalidade dos dados esperados. Sua sintaxe é:

```
INPUT [ <comentário> ] TO <nome da variável>
```

Exemplos:

```
INPUT "CUSTO DO PRODUTO" TO CP  
INPUT "NOME DO FORNECEDOR" TO NOMFOR
```

Após a execução de INPUT, o comentário incluído no seu comando é exibido no vídeo seguido de dois-pontos (:), significando que a atribuição a ser feita à variável designada no comando deverá ser digitada em seguida aos mesmos. Após a digitação, deverá ser pressionada a tecla RETURN, como de costume.

Se não for incluído nenhum comentário no comando INPUT, após sua execução serão exibidos apenas dois-pontos (:), indicando estar o sistema aguardando entrada de dados.

No primeiro exemplo, a atribuição à variável CP deverá ser digitada sem aspas e apenas com números dígitos, a fim de que ela seja automaticamente classificada pelo dBASE como uma VARIÁVEL NUMÉRICA, para ser processada como tal, isto é, aritmeticamente.

No segundo exemplo, a atribuição à variável NOMFOR deverá ser digitada com aspas. Nessa condição, ela será classificada como VARIÁVEL alfanumérica, automaticamente.

Comando ACCEPT

*** ACCEPT cria variável e permite atribuir-lhe dado apenas ALFANUMÉRICO digitado após sua execução. ACCEPT permite a inclusão de comentário, que é exibido no vídeo precedendo a digitação da atribuição a ser-lhe feita.

A atribuição feita através de ACCEPT não deve ser delimitada por aspas, já que será sempre considerada como expressão alfanumérica. Se for feita com aspas, estas figurarão como parte integrante da cadeia alfanumérica.

Até 64 caracteres são aceitos por variáveis criadas por ACCEPT.

Exemplos de comandos:

```
ACCEPT NOME DO CLIENTE TO NMCLI  
ACCEPT TO END
```

A execução de ACCEPT segue as mesmas normas de INPUT.

Deve-se observar que INPUT e ACCEPT são empregadas em arquivos de comando no modo interativo do dBASE, sendo sua principal utilização em programas no modo programável.

22.1.3 - ATRIBUIÇÃO A VARIÁVEIS COMO RESULTADO DA EXECUÇÃO DOS COMANDOS COUNT E SUM

Os comandos COUNT e SUM foram estudados no Capítulo 10, tendo sido exemplificadas atribuições de seus resultados a variáveis. Revendo-os após a leitura deste capítulo, sua compreensão será reforçada.

22.1.4 - ALTERAÇÃO DO CONTEÚDO DE VARIÁVEIS

No modo direto, altera-se o conteúdo de qualquer variável simplesmente atribuindo-se à mesma o novo conteúdo. Por exemplo, seja uma variável de nome A existente com o valor 10. Altera-se o seu valor para 200, por modelo, comandando-se:

```
STORE 200 TO A <RETURN>
```

No modo indireto, altera-se o conteúdo de qualquer variável utilizando-a em novo comando ou através de um novo comando.

22.1.5 - ELIMINAÇÃO DE VARIÁVEIS DE MEMÓRIA

Em determinadas situações, o uso de variáveis de memória pode ser bastante acentuado, de modo que algumas ou muitas tenham que ser apagadas ou eliminadas da memória do computador, para dar lugar a outras, já que o dBASE II pode controlar no máximo 64 delas.

É possível também eliminar de vez todas as variáveis de memória.

O comando do dBASE que permite eliminar seletivamente variáveis é descrito a seguir:

Comando RELEASE

*** RELEASE permite eliminar seletivamente variáveis de memória. Suas

sintaxes são:

```
RELEASE <nome(s) da(s) variável(is)>  
RELEASE ALL LIKE <nome da variável>  
RELEASE ALL EXCEPT <nome da variável>
```

Com referência à primeira sintaxe, quando forem designadas diversas variáveis, seus nomes deverão ser separados por vírgulas.

Exemplos:

```
RELEASE A,B,NM <RETURN>  
RELEASE ALL EXCEPT NMCLI <RETURN>
```

O uso dos caracteres-chaves ? e * é permitido, a fim de provocar a eliminação de um lote específico de variáveis. Um comando do tipo

```
RELEASE ALL LIKE N* <RETURN>
```

elimina da memória do computador todas as variáveis cujos nomes comecem com a letra N. O comando

```
RELEASE ALL EXCEPT AB? <RETURN>
```

elimina todas as variáveis menos aquelas cujas duas primeiras letras de seus nomes sejam AB, sendo a terceira letra qualquer.

Quando se deseja eliminar de uma só vez todas as variáveis de memória, pode ser usado o comando descrito a seguir:

Comando CLEAR

*** CLEAR elimina todas as variáveis de memória e fecha concomitantemente todos os arquivos em uso ou abertos. Sua sintaxe é:

```
CLEAR
```

22.1.6 - GRAVAÇÃO DE VARIÁVEIS

DE MEMÓRIA EM DISQUETE

Determinados tipos de arquivos de dados requerem que os conteúdos das variáveis processadas durante o seu uso sejam armazenados em arquivos específicos gravados em disquete, para usos sucessivos em ocasiões diversas, já que, ao encerrar-se o dBASE, todas as variáveis de memória são perdidas. O comando para tal fim é descrito a seguir:

Comando SAVE

*** SAVE armazena variáveis e respectivos conteúdos em arquivo, gravando-o automaticamente em disquete. Suas sintaxes são:

```
SAVE TO <nome do arquivo>  
SAVE TO <nome do arquivo> ALL LIKE  
      <nome(s) da(s) variável(is)>
```

As duas sintaxes admissíveis no comando SAVE permitem gravar todas as variáveis de memória em uso, ou seletivamente apenas algumas.

Um arquivo criado e gravado por SAVE recebe automaticamente a extensão .MEM, se outra não lhe for dada pelo operador. O dBASE reconhece a extensão .MEM como de arquivo específico de variáveis de memória.

22.1.7 - RECUPERAÇÃO DE VARIÁVEIS DE

MEMÓRIA DE ARQUIVO EM DISQUETE

Comando RESTORE

*** RESTORE recupera variáveis de memória de arquivos em disquete com extensão .MEM. Sua sintaxe é:

```
RESTORE FROM <nome do arquivo  
             de variáveis> [ADDITIVE]
```

A cláusula ADDITIVE permite que as variáveis existentes na memória do computador no momento do comando sejam preservadas, sendo recuperadas do disco tantas outras quantas não excederem o total de 64 admitido pelo dBASE II.

22.1.8 - LISTAGEM DE VARIÁVEIS DE MEMÓRIA

Comando LIST MEMORY ou DISPLAY MEMORY

Através do comando LIST MEMORY obtém-se a listagem das variáveis de memória existentes com o dBASE em uso.

A listagem inclui as seguintes informações:

- nomes, tipos e conteúdos das variáveis;
- quantidade de variáveis em uso;
- quantidade de memória usada pelas variáveis.

A sintaxe do comando é:

```
LIST MEMORY [TO PRINT]
```

23 - OUTRAS FUNÇÕES

DO dBASE II

Em capítulos anteriores foram estudadas funções do dBASE que se coadunam com as matérias tratadas nos mesmos. Para fins de referência, indicamos em seguida as funções já estudadas e os capítulos em que se encontram:

FUNÇÃO #

Número de registro atual: Capítulo 10.

FUNÇÃO !

Letras maiúsculas - Capítulo 10.

FUNÇÃO STR

Conversão de número para cadeia alfanumérica: Capítulo 10.

FUNÇÃO \$

Seleção de subcadeia alfanumérica: Capítulo 10.

FUNÇÃO *

Marcação de registro para deleção: Capítulo 12.

Neste capítulo serão estudadas as demais funções do dBASE II:

23.1 - FUNÇÃO @ - PESQUISA DE SUBCADEIAS

Permite pesquisar a posição inicial de uma seqüência de caracteres dentro de outra principal. Sua sintaxe é:

```
@ (<"cadeia1">, <"cadeia2">)
```

"Cadeial" refere-se à seqüência a ser pesquisada e "cadeia2" refere-se à cadeia alfanumérica principal. Quando são usadas variáveis em lugar das cadeias, não devem ser usadas aspas. Exemplos:

```
D          STORE "PROGRAMANDO" TO PR   <RETURN>
          ? @("AMANDO",PR)   <RETURN>
```

6

```
D          STORE "GRAMA" DO GR   <RETURN>
D          ? @ (GR,PR)   <RETURN>
```

4

A função @ permite pesquisar subcadeias alfanuméricas em campos de registros de arquivos. Por exemplo, seja utilizado o campo NOME do arquivo PESSOAL em uso:

```
D          DISPLAY NOME CARGO FOR
          @("cilia",NOME)   <RETURN>
```

```
00003  Maria Cecilia  Pedagoga
```

23.2 - FUNÇÃO & - MACRO-SUBSTITUIÇÃO

É usada para representar o conteúdo de uma variável alfanumérica. Sua sintaxe é:

```
&<nome da variável>
```

Um comando de uso frequente pode ser armazenado como conteúdo de variável e ser executado com maior facilidade e rapidez através desta função. Exemplo:

```
D          STORE "LIST FILES LIKE *.*" TO L  <RETURN>
D          &L  <RETURN>
```

Após o comando &L todos os arquivos e programas do disquete corrente serão listados.

A função & pode também ser usada com o comando FIND em arquivos indexados. Exemplo:

```
FIND &(nome da variável)
```

23.3 - FUNÇÃO CHR

Retorna o caractere correspondente a um número de código da tabela ASCII. Sua sintaxe é:

```
CHR (<número>)
```

Exemplo:

```
D          ? CHR (67)  <RETURN>
```

C

A função CHR é usada também para produzir na tela de vídeo ou por intermédio da impressora caracteres para os quais não há tecla no console. Exemplo:

```
D          ? CHR (129) <RETURN>
```

ü

Com o dBASE em uso, instruções para impressora são enviadas através da função CHR, conforme foi explicado no item 13.4.

23.4 - FUNÇÃO DATE

Indica ou altera a data corrente no dBASE em uso. Sua sintaxe é:

DATE ()

Exemplo:

```
D          ? DATE ( )    <RETURN>
```

06-01-88

Para mudar a data corrente no dBASE para 23/12/87, por exemplo:

```
D          SET DATE TO 23/12/87    <RETURN>
```

Ao alterar-se a data por esse meio, todavia, deve-se fazê-lo com atenção, pois a validade da mesma não é conferida pelo sistema, de modo que, se for indicada uma data absurda como 45/21/87, por exemplo, ela passará a ser corrente para todos os efeitos do dBASE.

23.5 - FUNÇÃO EOF (END OF FILE)

A finalidade da função EOF é controlar fim de arquivo. Sua função é lógica, assumindo a condição de F (FALSE = falso) em condições normais de uso do arquivo. Assume a condição T (TRUE = verdade) se for tentada ultrapassagem do fim do arquivo, caso em que o ponteiro de registros é forçado a voltar para o último registro pela função EOF. Sua sintaxe é o próprio nome, sem parâmetros:

EOF

Exemplo:

```
D          USE PESSOAL    <RETURN>
D          GOTO BOTTOM    <RETURN>
D          ? EOF        <RETURN>
```

F

D SKIP <RETURN>

Registro :00006

D ? EOF <RETURN>

T

23.6 - FUNÇÃO FILE

A função FILE pesquisa a existência de arquivo em disquete. Sua sintaxe é:

FILE (<<"Cd:Jnome do arquivo">>)

Sua função é lógica, retornando um F (FALSE) se não existir o arquivo no disquete alojado no disk drive corrente ou indicado. Retorna um T (TRUE) em caso afirmativo. Exemplo:

D ? FILE (A:PESSOAL.NDX) <RETURN>

T

23.7 - FUNÇÃO INT

A função INT converte um número fracionado em inteiro, desprezando todos os dígitos à direita do ponto. Sua sintaxe é:

INT (<número>)

Exemplo:

D ? INT (123.77) <RETURN>

123

INT não arredonda os números. Por exemplo:

```
D          ? INT (10.68) * 100  <RETURN>
```

1000

O resultado correto é 1068.

Arredondamentos são obtidos apenas através de artifícios de cálculo.

23.8 - FUNÇÃO LEN

A finalidade da função LEN é verificar o tamanho de cadeias alfanuméricas, que é expresso em quantidade de caracteres. Sua sintaxe é:

```
LEN (<"cadeia", ou <variável de  
      memória>, ou <nome de campo>)
```

Exemplo:

```
D          ? LEN ("INCONSTITUCIONALMENTE")  <RETURN>
```

21

```
D          STORE "PINDAMONHANGABA" TO PI  <RETURN>
```

```
D          ? LEN (PI)  <RETURN>
```

15

```
D          USE PESSOAL  <RETURN>
```

```
D          GOTO 5  <RETURN>
```

```
D          ? LEN (NOME)  <RETURN>
```

15

Neste caso, o número retornado refere-se ao tamanho do campo. Para obter-se o tamanho do CONTEÚDO do campo, usa-se a função TRIM (descrita mais abaixo) juntamente com LEN. Exemplo:

```
D          ? LEN (TRIM(NOME))  <RETURN>
```

11

23.9 - FUNÇÃO RANK

É utilizada para se obter o número de código ASCII de um caractere, do caractere inicial de uma cadeia alfanumérica ou do conteúdo de campo de caracteres. Sua sintaxe é:

```
RANK (<"caractere">, ou <"cadeia">, ou <campo>)
```

Exemplos:

```
D          ? RANK ("BRASIL")  <RETURN>
```

66

Em se tratando de conteúdo de um campo denominado CARGO de arquivo em uso:

```
D          ? RANK (CARGO)  <RETURN>
```

68

23.10 - FUNÇÃO TRIM

É utilizada para eliminar espaços em branco existentes à direita de campos, de variáveis ou de cadeias de caracteres. Sua sintaxe é:

TRIM (<campo>, ou <variável>, ou <"cadeia">)

Exemplo:

```
D      USE PESSOAL   <RETURN>
D      GOTO 5       <RETURN>
D      DISPLAY NOME, CARGO   <RETURN>
```

00005 Julio Cesar Gerente

```
D      DISPLAY TRIM(NOME), CARGO   <RETURN>
```

00005 Julio Cesar Gerente

23.11 - FUNÇÃO TYPE

A finalidade da função TYPE é identificar campos de registros, variáveis ou expressões quanto à sua classificação no dBASE. Sua sintaxe é:

TYPE (<campo> ou <variável>
ou <número> ou <"cadeia">)

Exemplos:

```
D      ? TYPE ("123")   <RETURN>
```

C

```
D      ? TYPE (123)    <RETURN>
```

N

```
D      STORE "CAJU" TO C   <RETURN>
```

```
D      ? TYPE (C)       <RETURN>
```

C

23.12 - FUNÇÃO VAL

Através da função VAL, uma cadeia de caracteres numéricos originalmente constituída como cadeia alfanumérica (sem valor aritmético, conseqüentemente) é convertida em valor aritmético. Sua sintaxe é:

VAL (<<"cadeia de caracteres numéricos">>)

Exemplos:

D ? VAL ("123") <RETURN>

123

D ? VAL ("12") * VAL ("12") <RETURN>

144

D ? VAL ("ABC") <RETURN>

0

24 - OUTROS COMANDOS

DO GRUPO "SET"

Ao longo deste curso foram descritos diversos comandos do grupo "SET", os quais permitem alterar determinadas condições de operação do dBASE, ativando ou desativando os parâmetros estabelecidos para elas. Sua introdução foi feita em função de seu vínculo com a matéria tratada em cada capítulo.

De tal grupo, restaram alguns comandos de menor importância, que são descritos a seguir:

SET BELL ON/OFF

*** Liga/desliga o sinal audível de advertência em operações indevidas de digitação, como digitar dados inválidos para um determinado tipo de campo ou quando este já foi preenchido totalmente.

Estado normal: ON. Não funciona no dBASE II PLUS.

SET CONSOLE ON/OFF

*** Liga/desliga a exibição normal em vídeo, sem afetar a saída para a impressora e as entradas pelo teclado, que deixam de ser monitoradas pelo vídeo.

Estado normal: ON.

SET ESCAPE ON/OFF

*** Liga/desliga o modo de interrupção de execução de comandos através da tecla ESC.

Estado normal: ON.

SET INTENSITY ON/OFF

*** Liga/desliga o modo de inversão do vídeo na exibição de campos de registros.

Estado normal: ON. Não funciona no dBASE II PLUS.

SET RAW ON/OFF

*** Liga/desliga o modo de exibição de DISPLAY e LIST sem espaços em branco entre os campos.

25 - "PROGRAMANDO" COM AUXÍLIO DOS UTILITÁRIOS DO dBASE II PLUS

De forma quase que geral, os comandos estudados neste volume são próprios do MODO INTERATIVO do dBASE II PLUS. Apenas alguns deles são usados também no MODO PROGRAMÁVEL.

As funções estudadas se aplicam quase que indistintamente em ambos os modos. Algumas encontram maior aplicação no MODO INTERATIVO enquanto que outras são mais usadas no MODO PROGRAMÁVEL.

Já as chamadas variáveis de memória encontram sua principal aplicação no MODO PROGRAMÁVEL do dBASE, mas seu estudo completo não podia deixar de ser feito neste volume, não só porque encontram certas aplicações no MODO INTERATIVO, mas também porque seu conhecimento é fundamental para a compreensão do MODO PROGRAMÁVEL, que será a próxima etapa de estudos para quem pretende se aprimorar no domínio do dBASE. Foi também considerado o fato de que seria pouco prático estudá-las apenas parcialmente agora e reencetar seu estudo depois.

O conhecimento e domínio do MODO INTERATIVO do dBASE possibilitam atender quase que ilimitadamente a todos os quesitos para criação, organização, uso e manutenção de BANCOS DE DADOS.

Por outro lado, o MODO PROGRAMÁVEL permite "flexibilizar", personalizar e otimizar o seu uso. Em outras palavras, o MODO PROGRAMÁVEL torna o uso do dBASE sofisticado, enriquecendo-o com recursos que permitem a qualquer usuário imaginoso fazer de um BANCO DE DADOS não apenas um serviço organizado e eficiente de informações programadas, mas também uma obra de arte quanto a seu funcionamento e apresentação geral em vídeo ou em relatórios produzidos através de impressora.

A nosso ver, tentar programar em dBASE sem dominar absolutamente o seu uso no MODO INTERATIVO não só torna o aprendizado mais difícil, mas - pior! - pode "bitolar" o programador por falta de conhecimento de detalhes de recursos e de "nuanças" de funcionamento do sistema que só o uso prático e contínuo do mesmo pode propiciar.

Assim, nossa recomendação a todos aqueles que pretendem se tornar bons profissionais em programação dBASE é que adquiram antes bastante prática no MODO INTERATIVO, para depois começar a estudar o MODO PROGRAMÁVEL.

Aos mais afoitos, todavia, o dBASE II PLUS oferece um recurso intermediário: a programação em dBASE II PLUS através dos programas utilitários incluídos em seu disco-mestre. São eles:

DGEN

GERAARQ

GERAREL

GERAETIQ

GERAMENU

Vamos descrevê-los em seguida, a fim de que possam prestar algum serviço àqueles que têm necessidade de começar a elaborar imediatamente alguns programas em dBASE. Devemos esclarecer, todavia, que os programas possíveis de serem obtidos através de tais utilitários são limitados quanto a alcance e recursos, como veremos.

DGEN

O utilitário DGEN serve apenas para gerar no vídeo um menu através do qual é permitido acesso a qualquer dos outros quatro. Dispensável, a nosso ver.

Sua descrição no manual do produto causa inicialmente alguma con-

fusão quanto a seu uso e finalidade. Acreditamos que o melhor serviço que pode prestar ao programador principiante em dBASE é ser analisado através do comando MODIFY do dBASE ou do TYPE do MSX DOS, com a finalidade de propiciar uma idéia de como pode ser programada a geração de um menu em dBASE.

O DGEN pode ser ativado diretamente através do MSX DOS, comandando-se:

```
DBASE DGEN <RETURN>
```

ou depois de o dBASE estar ativo no computador, através do comando:

```
DO DGEN <RETURN>
```

Após a execução de um desses comandos, será exibido no vídeo um menu através do qual qualquer dos outros quatro utilitários poderá ser acessado.

No caso de se desejar acesso direto a qualquer deles, comanda-se:

```
DBASE <nome do utilitário> <RETURN>
```

diretamente do MSX DOS, ou:

```
DO <nome do utilitário> <RETURN>
```

se o dBASE já estiver ativo no computador.

Vamos agora à descrição sinóptica dos outros utilitários:

GERAARQ

Desde que exista um arquivo de dados no disquete corrente no momento em que é ativado, GERAARQ gera e grava automaticamente um conjunto de programas relacionados a tal arquivo, os quais permitem fazer consultas, inserções, correções e deleções no mesmo através de um menu padrão, ao qual são subordinados referidos programas.

Ao ser ativado, GERAARQ solicita o nome do arquivo de dados sobre o qual vai operar, o nome do arquivo de índice relacionado ao mesmo e o nome do campo chave deste último.

Se não encontrar o arquivo de índice determinado, informa tal situação através do vídeo, juntamente com a pergunta "Cria? (S/N)", sendo conveniente responder S (= sim).

Em seguida, o GERAARQ informa sua ação através do vídeo e, ao finalizá-la, emite a seguinte mensagem:

```
PARA INICIAR O SISTEMA "nome do arquivo", DIGI-
TE O SEGUINTE:
      DO NO-PRIN  <RETURN>
FIM
GERAARQ
```

(As letras NO na terceira linha representam as duas primeiras letras do nome do arquivo DBF indicado ao GERAARQ ao ser ativado.)

Para testar GERAARQ com o arquivo PESSOAL.DBF, por exemplo, partindo da situação de estar apenas o MSX DOS ativado e encontrando-se no disk drive corrente o disquete que o contém, deve ser comandado:

```
DBASE GERAARQ  <RETURN>
```

Nessa situação, o processo passa a se desenvolver em conformidade com as mensagens de vídeo e respostas digitadas que são transcritas a seguir:

```
GERADOR                                     16/01/87
Nome do arq de BCO DE DADOS? :
      PESSOAL  <RETURN>
Digite o nome do arq de índice:
      PESSOAL  <RETURN>
Digite o campo chave de índice :
```

NOME <RETURN>

Buscando os nomes dos campos...

Checando os nomes de arquivos...

A partir deste ponto é listado no vídeo todo o conjunto de programas gerados por GERAARQ, sendo o processo finalizado com a mensagem reproduzida a seguir:

PARA INICIAR O SISTEMA "PESSOAL" , DIGI-
TE O SEGUINTE:

DO PE-PRIN

Nesse ponto, se comandarmos:

DO PE-PRIN <RETURN>

veremos exibido no vídeo o seguinte menu:

PESSOAL MENU PRINCIPAL

- 0. fim
- 1. consultar
- 2. adicionar
- 3. editar
- 4. compactar

selecione : :

o qual poderá ser usado de acordo com as opções oferecidas, dispensando o uso dos comandos específicos do dBASE para as mesmas finalidades.

Os programas gerados por GERAARQ recebem automaticamente nomes formados pelos dois caracteres iniciais do nome do arquivo de dados e, separados por hífen, sufixos próprios do utilitário. Todos eles recebem a extensão .CMD.

Para se ter uma idéia dos programas gerados na experiência feita com o arquivo PESSOAL, listamos a seguir seus nomes completos e espaços ocupados pelos mesmos no disquete:

PE-PRIN.CMD	3584 BYTES
PE-TELAS.CMD	512 BYTES
PE-GETS.CMD	512 BYTES
PE-QUALQ.CMD	512 BYTES
PE-EDIT.CMD	2048 BYTES
PE-PACK.CMD	1536 BYTES
PE-PESQ.CMD	2560 BYTES
PE-LOCAT.CMD	5120 BYTES

Todos esses programas podem ser facilmente analisados através do comando TYPE do MSX DOS.

GERAREL

Da mesma forma que o utilitário anterior, GERAREL opera sobre um arquivo de dados já existente no disquete corrente no momento em que é ativado. Também exibe mensagens no vídeo e requer entrada de informações via teclado.

Sua ação e o que resulta dela quase não diferem em relação ao comando REPORT FORM, a não ser quanto ao fato de que gera e grava automaticamente no disquete um programa muito mais longo do que o arquivo de formato, também gerado e gravado automaticamente por REPORT FORM. Para se ter uma idéia, enquanto que o arquivo de formato criado pelo comando REPORT FORM para o arquivo PESSOAL ocupa apenas 512 bytes no disquete, o programa gerado por GERAREL para a mesma finalidade ocupa 3072 bytes!

O programa gerado por GERAREL terá o nome que lhe for destinado pelo usuário como resposta a uma das mensagens-pergunta exibida no vídeo logo ao ser ativado. A extensão do nome será .PRG, compulsoriamente acrescentada pelo utilitário.

Um bom uso que talvez possa ser feito de um programa gerado pelo utilitário em foco é analisá-lo através do comando TYPE do MSX DOS, a fim de estudar a sua estruturação.

O programa gerado por GERAREL é executável com o dBASE ativado, por meio do comando:

DO <nome do programa>.PRG <RETURN>

e permite ao usuário determinar a saída do relatório através do vídeo ou da impressora.

GERAETIQ

A finalidade do utilitário GERAETIQ é gerar programa que gerencie a impressão de etiquetas com dados extraídos de arquivo tipo DBF.

Do mesmo modo que os outros utilitários já descritos, opera apenas sobre arquivo de dados já existente no disquete corrente no momento em que é comandado. Também exibe mensagens no vídeo e requer entrada de informações, sobre as quais monta o programa para gerenciamento da impressão de etiquetas.

Deve ser esclarecido que o programa gerado é bastante simples e limitado quanto à sua ação, possibilitando somente a impressão "empilhada" de etiquetas, ou seja: uma etiqueta apenas em toda a largura da tela de vídeo ou da impressora, neste caso originando grande desperdício de papel ou de etiquetas, se se conseguir utilizá-las com o programa, que não prevê nenhum ajuste de impressora.

Contudo, um programa gerado pelo utilitário GERAETIQ poderá ser usado para produzir relatórios de dados em que estes sejam dispostos apenas em seqüência perpendicular. Todos ou apenas alguns campos podem ser incluídos no relatório.

O programa gerado por GERAETIQ recebe o nome que lhe for destinado pelo usuário e a extensão compulsória .PRG. é executável apenas com o dBASE ativado, através do comando

DO <nome do programa>.PRG <RETURN>

e permite ao usuário determinar a saída através do vídeo ou da impressora.

GERAMENU

A finalidade deste utilitário está implícita em seu nome: gerar menus que gerenciem a execução de programas ou de módulos independentes de programas.

Ao contrário dos outros três utilitários, opera independente da existência ou não dos programas que serão administrados pelo menu que gera. Obviamente, se tais programas não existirem, terão que ser elaborados ou supridos a fim de que o menu se torne válido.

Há uma ressalva, todavia, com relação aos menus gerados pelo utilitário em foco: eles não funcionarão se não forem "modificados" após terem sido gerados, o que poderá ser feito através do MODIFY COMMAND do dBASE ou de um editor de texto.

A explicação para esse fato é que deve haver um comando de execução para cada programa do menu, mas sua inclusão deve ser feita juntamente com os nomes dos programas. Conseqüentemente, como estes serão determinados de acordo com o desejo de cada usuário, os menus gerados devem receber posteriormente tais comandos.

Essa é uma tarefa que exige algum conhecimento de programação em dBASE. Caso o leitor não possua experiência no assunto, poderá guiar-se pela instrução a respeito publicada no manual oficial do dBASE II PLUS, página 9-22, e que transcrevemos a seguir:

"SOMENTE PARA PROGRAMADORES

Criando

Os programas de menu principal que você criou com o GERAMENU não funcionarão até você modificá-los. Use o MODIFY COMMAND do dBASE II para chamar os programas.

1. Vá à instrução "DO CASE". Observe que cada opção do menu principal está descrita em um "CASE".

2. Agora, insira uma nova linha entre cada comando "CASE" e acrescente o comando "DO <nome do arquivo de programa>". Por exemplo:

```
DO PESQEMPR
```

3. Enquanto estiver no MODIFY COMMAND, você também pode:

- (a) colocar o seu nome no início do programa, na opção AUTOR e
- (b) fazer qualquer modificação desejada no texto do programa. Por exemplo, no texto da opção 0 trocar "fim" por "retornar ao prompt do dBASE II".

Como dissemos antes, para que um menu gerado pelo utilitário GERAMENU seja válido, é preciso que existam os programas incluídos nele, não se justificando, por outro lado, criar um menu para menos do que três itens.

Caso o usuário não disponha de programas em dBASE ou módulos que possam ser reunidos em menu, poderá incluir os gerados pelos outros utilitários através de experiências que realizar com os mesmos.

Depois de testar ou usar tais utilitários, o usuário constatará que os programas gerados por eles não utilizam recursos de grande alcance, quais sejam os propiciados por comandos como APPEND FROM, JOIN, REPLACE, UPDATE e outros ainda não estudados, que tornam um banco de dados não só um serviço otimizado de informações programadas, mas também uma obra de arte quanto a seu funcionamento e apresentação geral, incluindo estética de telas e impressos em vídeo e relatórios.

Para tanto, em verdade, são necessários programas mais sofisticados, que exigem do programador conhecimentos amplos do MODO PROGRAMÁVEL do dBASE, os quais demandam estudo especializado.

Os utilitários descritos neste capítulo não ensinam a programar, mas servem como introdução à programação em dBASE, principalmente se forem estudados analiticamente quanto à sua estrutura.

Como comentado no Capítulo 5, este primeiro volume de nosso curso é dedicado ao estudo exclusivo do MODO INTERATIVO do dBASE II PLUS, de sorte que o estudo completo do MODO PROGRAMÁVEL será feito no segundo

volume, a ser publicado brevemente.

Mas, a fim de que o leitor ou estudante tenha um panorama completo do dBASE II PLUS, listamos a seguir os comandos que encontram aplicação exclusiva no MODO PROGRAMÁVEL:

25.1 - COMANDOS PRÓPRIOS DO MODO PROGRAMÁVEL

? (*)	?? (*)
@ (*)	ACCEPT (*)
CANCEL	CLEAR GETS
DO (*)	DO CASE... ENDCASE
DO WHILE... ENDDO	ELSE
ERASE (*)	GET
IF... ENDIF	INPUT (*)
LOOP	NOTE (ou *)
OTHERWISE	READ
REMARK	RETURN
SAY	SET COLON (*)
SET CONSOLE (*)	SET ECHO
SET DEBUG	SET ESCAPE (*)
SET FORMAT	SET INTENSITY (*)
SET STEP	SET TALK
TEST	TEXT... ENDTEXT
WAIT	

Obs.: Os comandos assinalados com (*) são descritos neste volume. Consultar índices.

COMANDOS "PLUS" (Incluídos recentemente no dBASE II)

PLUS ADATE	PLUS CDATE
PLUS CGC	PLUS CHANGE
PLUS CODE	PLUS CPF
PLUS DDATE	PLUS FORMAT
PLUS LZERO	PLUS MOD11
PLUS TDATE	

BIBLIOGRAFIA

BARNES, LAN. dBASE II completo-total. São Paulo, McGraw Hill, Ltda., 1986.

CASARI, N. MSX - prática e domínio. São Paulo, Atlas, 1986.

CASARI, N. MSX com disk drive. São Paulo, McGraw Hill, Ltda, 1987.

DATALOGICA - dBASE II plus msx; versão 1.0. São Paulo, 1987.

MAGRI, J.A. dBASE II. São Paulo, Atlas, 1985.

YONG, C.S. Banco de dados. São Paulo, Atlas, 1985.

dBASE II PLUS PARA MSX

Curso Prático

Graças ao recente lançamento do dBASE II Plus para MSX, os microcomputadores deste padrão passaram a integrar o rol de máquinas profissionais de grande utilidade, já que a modalidade de serviço prestada pelo dBASE é a mais utilizada atualmente na área administrativa, empresarial e pública.

O dBASE II Plus para MSX propicia aos proprietários de microcomputadores de padrão MSX a possibilidade de organizarem rapidamente Bancos de Dados para uso pessoal ou empresarial, com as finalidades mais diversas, ampliando ilimitadamente o universo do uso prático dessas máquinas, que atualmente somam mais de duas centenas de milhar em uso no Brasil.

Este livro foi organizado com o objetivo de oferecer aos proprietários de microcomputadores do padrão MSX os conhecimentos necessários para assimilação das técnicas envolvidas na criação, organização e manutenção de Bancos de Dados elaborados através do dBASE. Foi compilado nos moldes de um Curso Prático, através do qual a matéria é ministrada de maneira gradativa e objetiva, por meio de experiências detalhadas que simulam o uso real de Bancos de Dados e que induzem o leitor a efetuá-las com facilidade no microcomputador, propiciando-lhe a prática necessária para o domínio do sistema.

Nos capítulos iniciais são ministrados conhecimentos básicos indispensáveis sobre a definição e tipos de Bancos de Dados. No Capítulo 7 são reunidas todas as informações úteis para apoio ao prosseguimento do curso prático ministrado, estendendo-se este até o Capítulo 25 e cobrindo todos os comandos, funções e instruções do Modo Interativo do dBASE II Plus para MSX.

APLICAÇÃO

Manual para aprendizagem e manipulação do software dBASE II Plus. Recomendado para cursos de reciclagem profissional. Indicado para profissionais e iniciantes que desejem conhecer este poderoso software.

publicação atlas