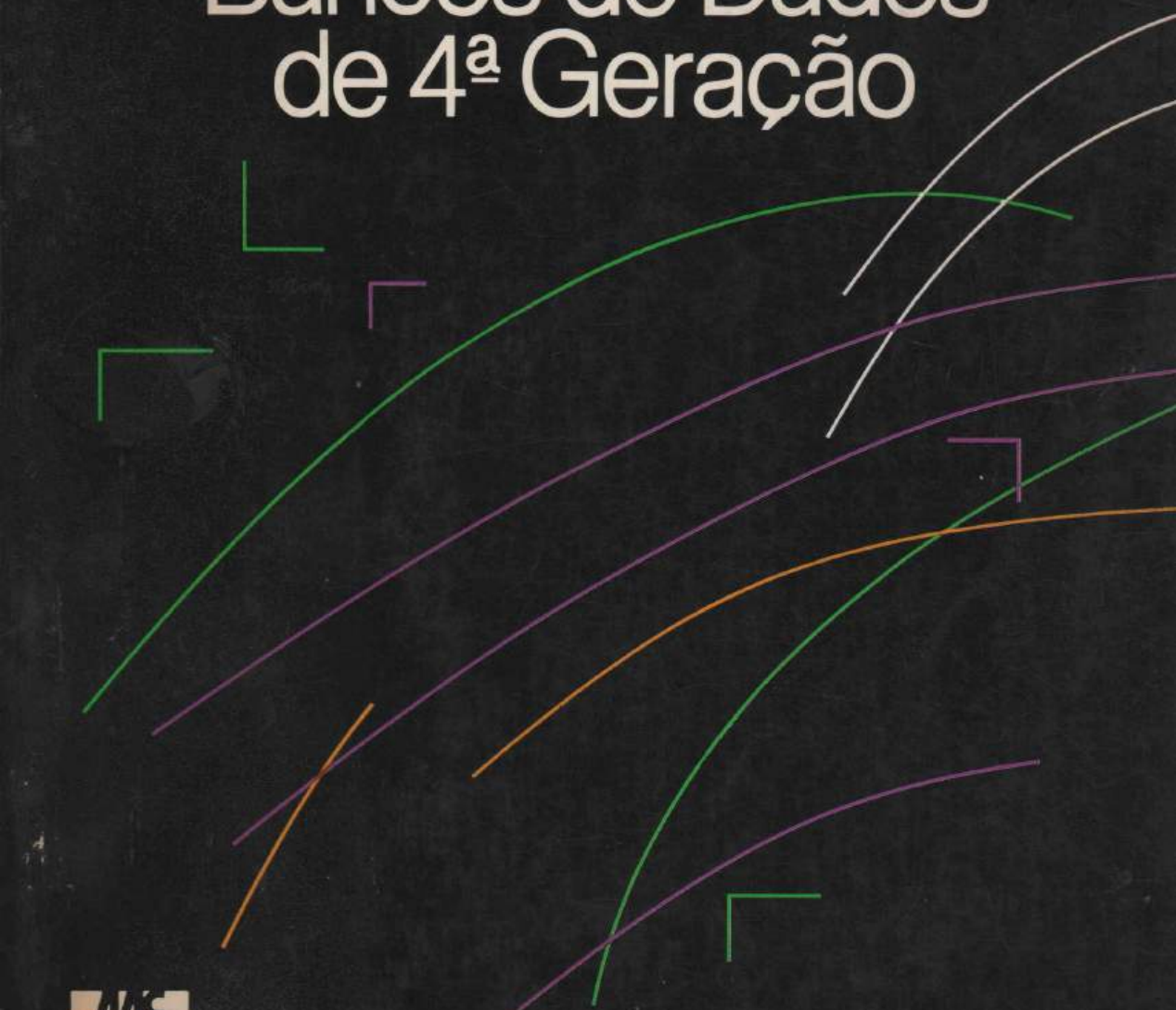
DATAFLEX

Bancos de Dados de 4ª Geração



McGraw-Hill

BURD INFORMÁTICA

DATAFLEX

O Banco de Dados de Quarta Geração

Valorize sua formação profissional,
seu futuro, sua consciência

DATAFLEX

O Banco de Dados de Quarta Geração

FOTOCÓPIA
XEROX - JAPROTUA
Valorize sua formação profissional,
seu futuro, sua consciência

DATAFLEX

O Banco de Dados de Quarta Geração

BURD INFORMÁTICA

MAKRON *Books* do Brasil Editora Ltda.
Editora McGraw-Hill Ltda.
São Paulo
Rua Tabapuã, 1105, Itaim-Bibi
CEP 04533
(011) 829-8604 e (011) 820-8528

Rio de Janeiro • *Lisboa* • *Porto* • *Bogotá* • *Buenos Aires* • *Guatemala* • *Madrid* • *México* • *New York* • *Panamá* • *San Juan* • *Santiago*

Auckland • Hamburg • Kuala • Lumpur • London • Milan • Montreal • New Delhi • Paris • Singapore • Sydney • Tokyo • Toronto

DATAFLEX. O Banco de Dados de Quarta Geração
Copyright © 1988 da Editora Mc Graw-Hill, Ltda.

Todos os direitos para a língua portuguesa reservados pela Editora McGraw-Hill, Ltda.

Nenhuma parte desta publicação poderá ser reproduzida, guardada pelo sistema "retrieval" ou transmitida de qualquer modo ou por qualquer outro meio, seja este eletrônico, mecânico, de fotocópia, de gravação, ou outros, sem prévia autorização, por escrito, da Editora.

EDITOR: MILTON MIRA DE ASSUMPÇÃO FILHO

Coordenadora de Revisão: Daisy Pereira Daniel
Supervisor de Produção: José Rodrigues
Composição e Arte-Final: Databecker Brasil

BURD SISTEMAS E TREINAMENTO
Equipe responsável:
Coordenação :José Ricardo Céspedes de Souza
Grupo de auxilio :Ana Claudia de Carvalho Fernandes
Daniel José Burd
Eliana Céspedes
Romero Numes de Moura

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

D257 Dataflex, o banco de dados de quarta geração / Burd Sistemas e Treinamento; [coordenação da equipe responsável] José Ricardo Céspedes de Souza. -- São Paulo : McGraw-Hill, 1988.

1. Banco de dados - Gerência 2. Dataflex I. Burd Sistemas e Treinamento. II. Souza, José Ricardo Céspedes de.

88-0663 CDD-001.6442

Índice para catálogo sistemático:

1. Banco de dados : Sistemas de gerência 001.6442
2. Dataflex : Banco de dados : Sistema gerenciador 001.6442

AGRADECIMENTOS

Gostaríamos de expressar os nossos sinceros agradecimentos a Ricardo, Carlos Eduardo e Guarizo todos da INTERCOMP pelo grande apoio no fornecimento de literatura sobre o Dataflex.

O auxílio prestado pela editora McGraw-Hill tornou viável o nosso anseio. Queremos agradecer em especial ao editor Mílton Mira de Assumpção Filho.

A Reginaldo Quintella de Carvalho, analista e programador de sistemas em dBASE III, pelo auxílio durante o desenvolvimento do Apêndice I - Dataflex X outros SGBDs.

AGRADECIMENTOS

Gostaríamos de expressar os nossos sinceros agradecimentos a Ricardo, Carlos Eduardo e Guarico todos da INTERCOMP pelo grande apoio no fornecimento de literatura sobre o Dataflex.

O auxílio prestado pela editora McGraw-Hill tornou viável o nosso ansseio. Queremos agradecer em especial ao editor Milton Mira de Assumpção Filho.

A Reginaldo Quintella de Carvalho, analista e programador de sistemas em dBASE III, pelo auxílio durante o desenvolvimento do Apêndice I - Dataflex X outros SGBDs.

A ESTRUTURA DO LIVRO

"DATAFLEX o banco de dados de quarta geração" foi dividido em seis grandes partes visando a um bom aproveitamento do leitor:

PARTE I: Discutiremos conceitos gerais de banco de dados como também já serão apresentados alguns termos exclusivos do Dataflex. Esta parte é fundamental para a compreensão das demais.

PARTE II: A criação de um programa em Dataflex será analisada em detalhes. Os utilitários que nos auxiliam tanto no desenvolvimento quanto na execução dos programas também serão apresentados. A PARTE II é a principal em termos de ressaltar as potencialidades da linguagem Dataflex.

PARTE III: Os comandos básicos da linguagem aqui serão apresentados. Esta parte pode ser pulada por aqueles que já possuem um contato com linguagens estruturadas.

PARTE IV: Como criar relatórios, e como extrair dados dos arquivos, fazendo seleções sem ter que se criar um programa serão tópicos tratados nesta parte.

PARTE V: Os comandos que tornam o Dataflex uma linguagem muito peculiar serão aqui apresentados; como você verá são comandos muito poderosos para o tratamento de bancos de dados.

PARTE VI: Esta parte é composta por VI interessantes apêndices:

APÊNDICE I: Dataflex X Outros SGBDs
APÊNDICE II: A versão 2.3 - vale a pena conhecê-la!
APÊNDICE III: Resumo de comandos e sintaxes
APÊNDICE IV: Resumo de Flex-Keys e funções
APÊNDICE V: Palavras reservadas do Dataflex
APÊNDICE VI: Glossário

A ESTRUTURA DO LIVRO

"DATAFLEX o banco de dados de quarta geração" foi dividido em seis grandes partes visando a um bom aproveitamento do leitor:

PARTE I: Discutiremos conceitos gerais de banco de dados como também já serão apresentados alguns termos exclusivos do Dataflex. Esta parte é fundamental para a compreensão das demais.

PARTE II: A criação de um programa em Dataflex será analisada em detalhes. Os utilitários que nos auxiliam tanto no desenvolvimento quanto na execução dos programas também serão apresentados. A PARTE II é a principal em termos de ressaltar as potencialidades da linguagem Dataflex.

PARTE III: Os comandos básicos da linguagem aqui serão apresentados. Esta parte pode ser pulada por aqueles que já possuem um contato com linguagens estruturadas.

PARTE IV: Como criar relatórios, e como extrair dados dos arquivos, lazendo seleções sem ter que se criar um programa serão tópicos tratados nesta parte.

PARTE V: Os comandos que tornam o Dataflex uma linguagem muito peculiar serão aqui apresentados; como você verá são comandos muito poderosos para o tratamento de bancos de dados.

PARTE VI: Esta parte é composta por VI interessantes apêndices:

APÊNDICE I: Dataflex X Outros SGBDs
APÊNDICE II: A versão 2.3 - vale a pena conhecê-la!
APÊNDICE III: Resumo de comandos e sintaxes
APÊNDICE IV: Resumo de Flex-Keys e funções
APÊNDICE V: Palavras reservadas do Dataflex
APÊNDICE VI: Glossário

SUMÁRIO

PARTE I - Introdução ao DATAFLEX

1. O que é Dataflex 3

2. Por que Dataflex 5

3. Conceitos Básicos 7
- 3.1 Convenções 7
- 3.2 Rotinas 9
- 3.3 Labels 10
- 3.4 Arquivos 11
- 3.5 Registros 13
- 3.6 Campos 14
- 3.7 Índices 14

4. Relacionamento Entre Arquivos 17

5. Conceitos Básicos em Dataflex 21
- 5.1 Organização dos dados 22
- 5.2 Imagem 23
- 5.3 Janelas 25
- 5.4 Buffer 27
- 5.5 Criando Variáveis 28

PARTE II - Criando, alterando e usando um Banco de Dados

6. Autodef 37
- 6.1 Criando uma imagem 37
- 6.2 Usando o Autodef 40
- 6.3 Alterando a aplicação 44
- 6.4 Usando a aplicação 50

7. Macro Enter 58
7.1 O que é Macro Enter 58
7.2 Estrutura da Macro 58

8. Macro Entergroup 70
8.1 O que é Macro Entergroup 70
8.2 O que faz a Macro Entergroup 71

9.Utilitários Dataflex 83
9.1 Filedef 84
9.2 Reindex 105
9.3 Enterdef 108
9.4 Menudef 111

PARTE III - Comandos Básicos

10. Comando de Decisão IF/ELSE 121

11. Indicadores 123

12. Comandos de Desvio 125
12.1 Goto 126
12.2 Gosub 127
12.3 On Goto 127
12.4 On Gosub 128
12.5 Return 129
12.6 Pop 130

13. Comandos de Laço (Loop) 131
13.1 Begin - End 131
13.2 Case / Caseelse - Caseend 132
13.3 For - From - To - Step - Loop 134
13.4 Repeat - Until 135
13.5 Repeat - Loop 137
13.6 While - End 137

14. Comandos de Processamento Numérico 139
14.1 Calc ou Calculate 139
14.2 Increment e Decrement 140
14.3 Move 141

15. Comandos de Manipulação de strings 143
15.1 Left 143
15.2 Right 145
15.3 Mid 146
15.4 Append 148

16. Comandos de Manipulação de Arquivos 151
16.1 Open 151
16.2 Clear 153
16.3 Save 153
16.4 Saverecord 154
16.5 Delete 155
16.6 Find 155

PARTE IV - Selecionando e obtendo informações do Banco de Dados

17. Usando o Query 163

18. Macro Report 186
18.1 O que é Macro Report 186
18.2 A estrutura da Macro 186
18.3 Seções da Macro Report 189
18.4 Seleções com o Report 202

PARTE V - Comandos avançados e Programação

19. Comandos de Manipulação de arquivos 207
19.1 Findkey 207
19.2 Searchkey 209
19.3 Sum 211

20. Relacionamento de Múltiplos Arquivos 213
20.1 Relate 214
20.2 Attach 214

21. Usando o Report 218
21.1 Por quê? 218
21.2 Não só em relatórios 219

22. Passando e Recebendo Parâmetros 225
22.1 Chain 225
22.2 Cmdline 227

PARTE VI - Apêndices

I - Dataflex X Outros SGBDs 231
II - A versão 2.3 - Vale a pena conhecê-la! 239
III - Resumo de Comandos e Sintaxes 245
IV - Resumo de Flexkeys e Funções 254
V - Palavras Reservadas do Dataflex 255
VI - Glossário 257

PARTE I

Introdução ao Dataflex

Muitas preciosidades em termos de sistemas e aplicativos vivem no anonimato. Para ser popular um programa deve, além de ser muito bom, ser extremamente bem divulgado. Deve ser usado, criticado e testado.

Com o intuito de tirar do anonimato o excelente Sistema Gerenciador de Banco de Dados Dataflex, empreendemos a hercúlea tarefa de escrever este texto, o qual é o primeiro livro sobre Dataflex escrito e publicado no Brasil.

O "DATAFLEX o banco de dados de quarta geração" destina-se a analistas, programadores, usuários e curiosos em geral que queiram penetrar neste novo conceito de Banco de Dados, e será especialmente útil àqueles que já conhecem algum outro sistema de Banco de Dados, os quais poderão efetuar uma interessante comparação entre os sistemas.

Todo o texto foi escrito através de um diálogo agradável, no qual procuramos introduzir lentamente os novos conceitos. Não poupamos exemplos para ilustrar os tópicos mais significativos, prova disso é que desenvolvemos um sistema completo para vídeo locadoras.

PARTE I

Introdução ao Dataflex

CAPÍTULO 1
O QUE É DATAFLEX

Dataflex é o mais versátil e compreensível Sistema de Gerenciamento de Banco de Dados disponível hoje no mercado.

Ele é uma poderosa ferramenta para geração de aplicações tanto para um programador profissional como para um usuário final, pois através de um sistema de menus, qualquer um deles poderá desenvolver aplicações sem necessitar ter um prévio conhecimento de programação ou a respeito de sistemas operacionais.

Dataflex é a solução ideal para problemas de manipulação de dados numa ampla variedade de sistemas tanto monousuários como multiusuários. Com rotinas pré-desenvolvidas, um vasto número de comandos e facilidades fornecidas pela sua estrutura, Dataflex livra você do tédio da programação e possibilita que o seu êxito seja a finalização da aplicação e não o alívio de terminar mais um programa com sucesso.

Comparando Dataflex, passo a passo, suas facilidades de uso, com outros sistemas de desenvolvimento, veremos rapidamente por que Dataflex é a melhor escolha em termos de Sistemas Gerenciadores de Bancos de Dados e Sistemas Geradores de Aplicação. Com Dataflex você não necessita de módulos extras, pois conta com um sistema completo em mãos. Todas as ferramentas e utilitários estão em um único e compacto bloco de programas e utilitários. Dataflex é formado por:

- Gerenciador de Banco de Dados Relacional;
- Linguagem de 4a geração, expandível com macrocomandos;
- Gerador de relatórios e entradas de dados automático;
- Query para geração imediata de relatórios individualizados com nove níveis de seleção;
- Compilador de programas ;
- Teclas de função redefiníveis em tempo de programação;
- Editor de textos;
- Indexação de acesso seqüencial.

Em Dataflex pode-se criar conjuntos de dados independentes de programas de aplicação, pois os itens de conjuntos de dados podem ser usados em múltiplas aplicações, sem o uso desnecessário de grandes esforços normalmente usados na maior parte nas aplicações geradas por outros sistemas geradores de aplicações.

CAPÍTULO 2
POR QUE DATAFLEX

A cada dia fica mais fácil conseguirmos que o computador faça aquilo que desejamos. Esta situação é bem diferente da vivida pelos pioneiros da informática, que na maioria das vezes eram super técnicos em eletrônica e suavam muito para efetuar uma multiplicação nos seus gigantescos computadores cheios de válvulas.

Hoje praticamente "conversamos" com a máquina, pedimos-lhe que execute uma dada tarefa sem qualquer dificuldade. Vários programas permitiram esta evolução e no que tange a banco de dados o Dataflex é um grande responsável pela facilidade encontrada nesta comunicação.

Se você quiser implementar uma aplicação que envolva banco de dados com certeza o Dataflex poderá ajudá-lo muito na obtenção do máximo de seu computador exigindo o mínimo de você.

Uma linguagem que faz tudo para o seu "patrão" costuma ser limitada. "Como o Dataflex equaciona este problema?"

Os programas geradores de programas oferecem, na maioria das vezes, soluções bastante limitadas, sem muita imaginação. Em contrapartida as linguagens que permitem a você fazer o que quiser, normalmente, exigem profundo conhecimento técnico, como é o caso das linguagens C e Pascal. Sendo o Dataflex desenvolvido nestas linguagens, ele permite ao

programador mais minucioso usufruir das potencialidades da mesma sem qualquer restrição (da versão 2.3 em diante os fontes foram escritos em C e as versões anteriores foram desenvolvidas em PASCAL).

FLEXIBILIDADE:

É disponível para um grande número de computadores desde os famosos 8 bits até os modernos 32, opera em vários sistemas operacionais. Estas características asseguram um crescimento tranqüilo, permitindo que se mude de equipamento sem perder todo o trabalho já executado.

MULTIUSUÁRIO: ...

Enfim, por estas e muitas outras razões: DATAFLEX.

CAPÍTULO 3
CONCEITOS BÁSICOS

Neste capítulo veremos alguns itens e conceitos básicos em termos gerais de processamento de dados, para um fácil aprendizado tanto de Dataflex como de outros bancos de dados.

3.1 Convenções

Este item trata das convenções de sintaxes de comandos que serão estudados nos próximos capítulos. Isto se deve ao fato de que alguns comandos necessitam de determinados argumentos obrigatoriamente, enquanto outros comandos são mais liberais quanto ao uso deste ou daquele argumento. Portanto usaremos os símbolos abaixo declarados para indicar obrigatoriedade ou opcionabilidade de argumentos de comandos:

( ) - parênteses: indica obrigatoriedade, isto é, os argumentos que estiverem dentro dos parênteses terão de serobrigatoriamente usados. Caso haja mais de um argumento obrigatório, vários conjuntos de "()" um embaixo do outro, devemos entender que se pode escolher qualquer uma daquelas formas.

** Em virtude da clareza do texto alguns comandos não virão acompanhados de "()", sendo a parte obrigatória aquela que estiver escrita em letras minúsculas à direita dos comandos.

< > - sinal de maior e menor: indica opcionabilidade, isto é, os comandos ou opções internas aos "< >" poderão ser dispensados. Caso haja mais de um argumento opcional, poderá ser feita uma escolha daquele que melhor se adaptar à sua vontade.

Exemplos:

1) ENTRY arquivo.campo < { opções de formatação } >

Neste caso, "arquivo.campo" é um argumento obrigatório e "{opções de formatação}" é opcional (as chaves "{}" presentes na área opcional fazem parte obrigatória do argumento).

2) OPEN nome_arquivo < INDEX.n >

3) MOVE (const1) TO (var_int1 < var_int2 ... var_int n >)
(expr1) (var_num1 < var_num2 ... var_num n >) ...

Neste caso, temos diversos argumentos obrigatórios. Deve-se primeiro escolher um dos argumentos obrigatórios da direita. Segundo, devemos escolher um dos argumentos obrigatórios da esquerda. Terceiro, no caso de querermos escolher os argumentos opcionais, do argumento da esquerda previamente selecionado, deveremos usar tantos argumentos opcionais quantos forem dados ou necessários para seus propósitos.

3.2 ROTINAS

São conjuntos delimitados de instruções que serão executadas seqüencialmente até que seja encontrado um comando que indique o fim da mesma. Após a execução da rotina, o Dataflex devolverá o fluxo do programa ao ponto de onde esta foi chamada, isto é, em certo ponto do programa Dataflex um determinado comando desviou o fluxo de execução para uma área diferente da que ele estava executando; nesta área encontram-se as rotinas e sub-rotinas dos programas, que serão executadas e, ao seu final, retornarão ao ponto de onde foram chamadas para executar o comando imediatamente abaixo do comando de desvio.

As normas para a nomenclatura das rotinas são as seguintes:

- são válidas quaisquer letras (A a Z), maiúsculas ou minúsculas (para Dataflex é indiferente, mas em programação ajuda na visualização geral do programa);

- é válido também o underline "_" como componente do nome de rotinas (não deve ser o primeiro caractere do nome da rotina);

- ter no máximo 80 caracteres de tamanho, isto é, o somatório geral do número de letras e underlines do nome não deve ultrapassar 80 posições;

- obrigatoriamente, o nome da rotina deve terminar com ":" (dois pontos), indicando ao Dataflex que não se trata de um comando, mas sim de uma rotina.

Exemplo:

```
Rotina_exemplo:
   CLEARXY 24 0
   GOTOXY  24 0
   SHOW " Confirma (S/N) ?"
```

```
INKEY TECLA
CLEARXY 24 0
IF TECLA EQ "N" RETURN INICIO
RETURN
```

3.3 LABELS

Basicamente, Labels são rotinas, um conjunto de instruções que serão executadas seqüencialmente, mas estes não voltarão ao ponto de onde foram chamados, continuando a execução do programa até que seja encontrado o fim do programa ou uma das instruções for um comando de desvio que redirecionará o fluxo de execução do programa.

As regras de formação de nomes de Labels são as mesmas da formação de nomes de rotinas.

Exemplo:

```
Label_exemplo:
  CLEAR LOCACAO
  CLEAR CADCLI
  CLEAR CADFITAS
Inicio:
  ACCEPT tela.1
  ACCEPT tela.2
  ACCEPT tela.3
  IF tela.3 EQ "5 GOTO inicio

  MOVE                tela.1  TO locacao.numero
  MOVE                tela.2  TO locacao.cliente

  SAVE locacao
  GOTO Label_exemplo
```

3.4 ARQUIVOS

São agrupamentos de conjuntos de dados, ordenados ou não, que podem sofrer algum tipo de processamento gerando informações. Os processamentos que um arquivo de dados pode sofrer são: inclusões de novos dados, alterações nos dados já existentes, exclusão dos dados residentes no arquivo, e consultas podem ser feitas com os dados ali armazenados.

Arquivos também podem ser vistos como fontes de programas, isto é, programas que são escritos e que depois são compilados para uma linguagem que o computador entenda. Arquivos podem também ser os programas gerados por geradores de aplicações que futuramente serão compilados (programas de aplicação), listados em impressoras ou vídeo (relatórios gerados pelos módulos automáticos de aplicações ou pelos programas de aplicação). Como saber diferenciar estes arquivos é simples; basta sabermos suas terminações, isto é, sabermos que extensões dos nomes devemos dar aos arquivos gerados ou sabermos reconhecer quais são.

As seguintes extensões são geradas pelo Dataflex:

- **.FRM** : arquivo-fonte de programas de aplicação. Este arquivo deve ser compilado para que possa ser executado.
- **.FLX** : arquivo-destino do programa-fonte após compilação. Já está pronto para ser executado.
- **.DAT** : arquivo de dados. Este arquivo guarda todos os dados gravados durante os processos básicos de manipulação de dados (inclusão, alteração e exclusão).
- **.Kn** : arquivo de índices. Este arquivo tem a relação de registros organizados seqüencialmente de acordo com sua definição. Você pode

criar até 10 (n entre 1 e 10) arquivos de índices, isto é, pode-se ter até 10 índices para cada arquivo de dados.

- **.TAG** : arquivo que guarda os nomes dos campos do arquivo de dados. Cada registro deste arquivo é um nome de campo do arquivo.

- **.FD** : arquivo que guarda os "*apelidos*" dados aos campos do arquivo Dataflex. Isto é necessário para a agilização do processo de compilação e acesso aos nomes de arquivos durante um dos módulos de criação de aplicações.

- **.RPT** : arquivo-fonte de programas de aplicações de geração de relatórios. Tem as mesmas características que o de terminação .FRM devendo também ser compilado para que possa ser executado (sua terminação após a compilação também será .FLX).

- **.DEF** : arquivo que é a imagem da definição do arquivo. É usado para conferência da estrutura do arquivo de dados e pode ser usado para geração de programas de aplicação através de um módulo chamado ENTERDEF, que a partir da definição de um arquivo gera um programa-fonte pronto para ser compilado e usado.

Das terminações descritas acima, as únicas que não devem ser alteradas são: **.DAT, .FLX, .Kn, .TAG, .FD, .DEF**; pois serão usadas pelo compilador para checagem do seu programa-fonte. As demais terminações (**.FRM** e **.RPT**) podem ser alteradas à sua vontade ou, como se trata de programas-fonte, podem ser criados com qualquer terminação, não sendo nenhuma daquelas já citadas.

Exemplo:

CADFITAS DAT arquivo de dados
CADFITAS TAG arquivo que guarda os nomes dos campos do aquivo de dados
CADFITAS FD arquivo que guarda os nomes atribuidos ao arquivo e aos campos
CADFITAS K1 arquivo de dados do indice 1
CADFITAS K2 arquivo de dados do indice 2
CADFITAS DEF arquivo que guarda a imagem da definicao do arquivo de dados
CADFITAS K3 arquivo de dados do indice 3
CADFITAS K4 arquivo de dados do indice 4

3.5 REGISTROS

São conjuntos de dados agrupados logicamente organizados por um identificador comum. Este identificador comum normalmente é um número seqüencial denominado RECNUM ou RECord NUMber, isto é, número do registro. Este RECNUM é exclusivo, não pode ser duplicado. Um conjunto de registros forma um arquivo de dados.

Exemplo:

Registro do arquivo FICHA:LOCACAO CODIGO_FITA VALOR_LOCACAO DESCONTO

Registro do arquivo LOCACAO :

NUMERO DATA RESPONSAVEL CODIGO_CLIENTE NOME_CLIENTE NUM_FITAS_LOC VALOR BAIXA

3.6 CAMPOS

São pedaços de informação. São itens de dados que se agrupam para formar conjuntos de dados e formar uma informação. Um conjunto de campos forma um registro.

Exemplo:

Campos do arquivo LOCACAO:

NUMERO
DATA
RESPONSAVEL
CODIGO_CLIENTE

3.7 ÍNDICES

São chaves pelas quais ordenamos ou acessamos um conjunto de dados de um arquivo. A formação de índices é sempre feita, com relação aos campos do arquivo, em qualquer ordem escolhida pelo usuário. Os índices ordenam os campos em ordem alfabética e em ordem numérica crescente. Em Dataflex podemos ter índices de múltiplos segmentos, isto é, uma chave formada por diversos campos. Os índices podem ser exclusivos (a chave só aparece uma única vez no arquivo de dados) ou não-exclusivos (a chave pode ser encontrada repetida, em diversos registros, dentro do arquivo de dados).

Exemplo:

Supondo um conjunto de dados apresentados da seguinte forma:

NOME	IDADE
1 ANTONIO	20
2 JOAO	21
3 JOSE	23
4 ROBERTO	10
5 ANA MARIA	19
6 ANA CLAUDIA	45
7 ANA PAULA	21
8 JOSE ROBERTO	34
9 JOSE RICARDO	17

Poderíamos criar uma chave por NOME para indexar o conjunto de dados. Ficaríamos então com o conjunto de dados ordenados pelo índice NOME da seguinte forma:

NOME	IDADE
6 ANA CLAUDIA	45
5 ANA MARIA	19
7 ANA PAULA	21
1 ANTONIO	20
2 JOAO	21
3 JOSE	23
9 JOSE RICARDO	17
8 JOSE ROBERTO	34
4 ROBERTO	10

Poderíamos, ainda, organizar estes dados por IDADE:

NOME	IDADE
4 ROBERTO	10
9 JOSE RICARDO	17
5 ANA MARIA	19
1 ANTONIO	20
7 ANA PAULA	21
2 JOAO	21
3 JOSE	23
8 JOSE ROBERTO	34
6 ANA CLAUDIA	45

CAPÍTULO 4
RELACIONAMENTO ENTRE ARQUIVOS

Relacionamento entre arquivos é uma função que liga dois ou mais arquivos de dados através de seus campos em comum. Isto significa acessar dados de um arquivo a partir de dados de outro arquivo. Isto ocorre devido ao fato de que os dois arquivos compartilham da mesma informação, isto é, os dois arquivos possuem os mesmos campos e conteúdos.

O fato de arquivos diferentes armazenarem a mesma informação deveria indicar diminuição de espaço de armazenamento, redundância de dados e perda de tempo de acesso aos dados. Isto não ocorre porque os dados relacionados são mais confiáveis por não serem repetidos dentro do mesmo registro do arquivo de dados, reduzem o tempo de acesso pois a quantidade de registros a serem lidos será menor, facilidade de atualização dos dados, pois não será necessário alterar uma quantidade enorme de registros, bastando alterar um.

Vejamos o exemplo abaixo:

Devemos cadastrar todos os funcionários que trabalham nas seções de uma loja de departamentos. Teríamos basicamente estas informações a cadastrar:

- NOME DA SECAO
- NUMERO DA SECAO
- CHEFE DA SECAO
- NOME FUNCIONARIO
- ENDERECO
- BAIRRO
- CIDADE
- ESTADO
- CEP
- TELEFONE

Caso fosse criado um cadastro nesta estrutura teríamos uma perda de espaço de armazenamento enorme, pois estaríamos armazenando dados repetidos no arquivo de dados (NOME, NÚMERO e CHEFE DA SEÇÃO). Como se não bastasse, o que aconteceria ao arquivo de dados se mudasse o chefe da seção? Teríamos de alterar em todos os registros com aquela informação.

Mas e se separássemos este arquivo de dados em outros dois: um contendo os dados da seção e o outro os dados dos funcionários.

Secoes :
- NOME DA SECAO
- NUMERO DA SECAO
- CHEFE DA SECAO

Funcionarios :
- NOME DA SECAO
- NOME FUNCIONARIO
- ENDERECO
- BAIRRO
- CIDADE
- ESTADO
- CEP
- TELEFONE

Agora já não teríamos de alterar todos os registros, para trocar nomedo chefe da seção. Bastaria alterar, no arquivo de seções, o campo.

CHEFE DA SEÇÃO. Perceba que no cadastro de funcionários o campo NOME DA SEÇÃO foi mantido, pois deseja-se manter o controle de onde os funcionários trabalham.

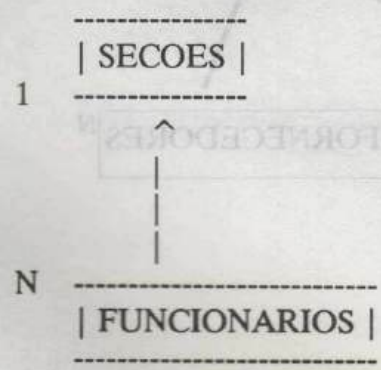
Veja que se quisermos saber quais funcionários trabalham na seção de roupas infantis, torna-se fácil com esta nova estrutura, bastando vasculhar o cadastro de funcionários, pelo campo nome da seção e mostrar aqueles cujo conteúdo do campo for igual ao desejado.

A este tipo de ligação entre os arquivos de dados damos o nome de *Relacionamento.*

Repare que neste exemplo ainda perdemos espaço de armazenamento pois usamos o campo NOME DA SEÇÃO quando poderíamos usar o NÚMERO DA SEÇÃO, uma vez que um número é mais fácil de pesquisar do que um nome. Neste caso o arquivo de funcionários seria mudado para:

Funcionarios :
- NUMERO DA SECAO
- NOME FUNCIONARIO
- ENDERECO
- BAIRRO
- CIDADE
- ESTADO
- CEP
- TELEFONE

Graficamente representamos um relacionamento da seguinte maneira :



Esta representação gráfica significa dizer que o arquivo funcionários está relacionado ao arquivo de seções, na proporção de 1 seção para N funcionários.

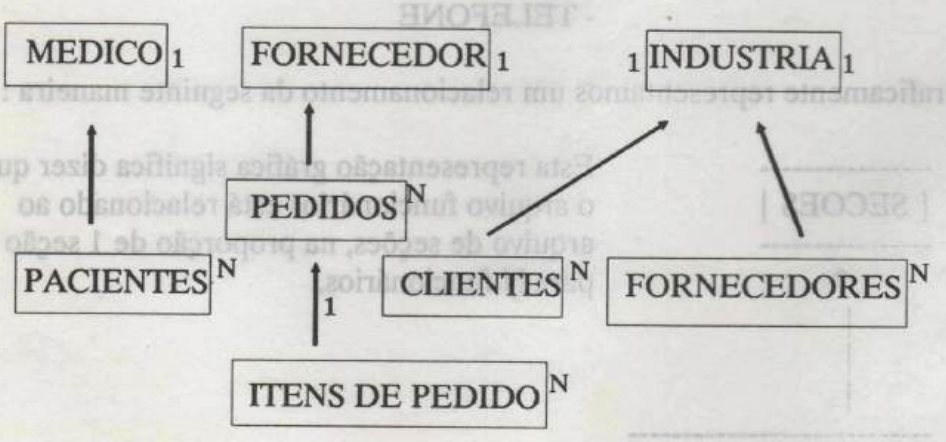
Através deste exemplo podemos notar mais detalhes quanto ao relacionamento de arquivos. Enquanto em uma seção podemos ter dezenas de funcionários, cada funcionário poderá estar trabalhando em apenas uma seção. Este tipo de situação, onde para cada seção existem "N" funcionários e para cada funcionário existe apenas uma seção, chamamos *classe de relacionamento* 1:N.

Relacionamentos entre arquivos em Dataflex podem ocorrer somente quando a classe de relacionamento for de 1:N (1 para N).

Os relacionamentos entre arquivos em Dataflex devem ser olhados sempre do arquivo que está mais abaixo para o arquivo que está mais acima, isto é, do arquivo N para o arquivo 1 (Funcionários está relacionado com arquivo de Seções).

Como foi dito, para que haja um relacionamento entre arquivos devemos ter determinadas informações presentes em todos os arquivos alvos do relaciona- mento. Para que isto aconteça é preciso que estes dados não se repitam. Portanto, os campos que serão alvo de relacionamentos devem pertencer a índices exclusivos dos arquivos de dados, para evitar a duplicação de informações.

Vejamos outros exemplos:



CAPÍTULO 5
CONCEITOS BÁSICOS EM DATAFLEX

Neste capítulo veremos alguns conceitos necessários para uma perfeita comunicação entre você e Dataflex. Os conceitos descritos a seguir são primordiais para o entendimento de como Dataflex trata os dados, arquivos e variáveis; são conceitos diferentes ou inexistentes em outros bancos de dados. Antes, alguns conhecimentos são necessários para um bom entendimento do que será discutido a seguir:

- **BIT** : é a menor unidade de armazenamento de dados de um computador. Para o computador um bit pode assumir apenas dois valores: 0 ou 1 (verdadeiro ou falso).

- **BYTE** : conjunto de oito bits, que recebe, também, o nome de PALAVRA.Como um byte contém oito bits e cada bit pode assumir o valor 0 ou 1, portanto um byte pode assumir um total de 256 (2^8) combinações de valores. Um byte como conjunto de oito bits, não nos diz nada, mas um byte como uma parte de informação (uma letra, um número, um caractere gráfico) já nos interessa.

- **ASCII** : código padrão de transferência de informação. Este código assinala combinações de 7 bits para codificar 128 caracteres usados nos computadores. Estes 7 bits, combinados adequadamente, podem representar letras, números, caracteres gráficos ou símbolos no monitor de vídeo de seu computador ou na sua impressora. O oitavo

bit é normalmente usado para assinalação do sinal; quando se trata da representação de um número, este sinal pode ser positivo ou negativo, de acordo com o arranjo do oitavo bit. As outras 128 combinações são usadas por alguns computadores para a codificação de caracteres gráficos ou simplesmente a repetição dos caracteres já existentes no *set* de caracteres.

5.1 ORGANIZAÇÃO DOS DADOS

Neste item do capítulo trataremos de como Dataflex armazena os dados, sejam eles letras, números, datas ou caracteres especiais.

- Formato numérico: campos numéricos podem conter apenas números, isto é, Dataflex permite apenas a entrada de números, e pontos decimais (vírgulas se assim estiver formatada sua versão) e caracteres de sinalização numérica (+ -). Basicamente, como um byte é uma *PALAVRA*, ela pode guardar um número, mas Dataflex dá outro tratamento aos campos numéricos; um número é armazenado compactado, isto é , um número de dois dígitos será armazenado em um byte inteiro, enquanto que outro número de apenas um dígito será armazenado em meio byte; isto não é totalmente verdade pois Dataflex reserva um byte inteiro para guardar o número. Vejamos um exemplo, talvez seja mais fácil:

Seja um campo numérico com quatro dígitos (1234): ocupará dois bytes no arquivo de dados, sendo que um campo numérico de cinco dígitos (12345), ocupará um espaço de três bytes. Isto é, divide-se o número de dígitos do número por dois, se der fracionário, arredonda-se para o inteiro superior mais próximo.

- Formato data: Dataflex trata as datas de forma diferente do usual. As datas para Dataflex são números inteiros mas transformados para o formato juliano de datas, isto é, as datas são convertidas em números

que indicam o número de dias transcorridos a partir de um ponto predeterminado. Isto ajuda muito quando se deseja realizar contas com datas futuras ou passadas. Como são números as datas também sofrem o processo de compactação ficando com 3 bytes de tamanho (DDMMAA são seis números inteiros). Em Dataflex pode-se ter três formatos para datas: Americano MMDDAA, Europeu DDMMAA e Militar AAMMDD.

- Formato ASCII : são todas as letras, números e caracteres especiais, seja do alfabeto ou do *set* de caracteres, disponíveis. Todos os caracteres ASCII ocupam, invariavelmente, um byte independente de seu tipo.

5.2 IMAGEM

É a representação de seu banco de dados. Uma imagem pode conter dados de entrada de diversos arquivos, como também pode conter mensagens para uma outra imagem de entrada de dados. Dataflex pode gerenciar até 255 imagens ao mesmo tempo por programa. São como páginas de livros que podemos acessar em qualquer ordem, mas têm um início e um fim bem definidos.

Para iniciar uma imagem, coloca-se uma barra "/" na primeira linha e coluna seguida do nome da imagem. Desenha-se a imagem desejada com auxílio de um editor de textos que gere arquivos ASCII. Para indicar o fim de uma imagem, coloca-se uma barra "/" seguida de um asterisco "*". Caso se deseje desenhar mais de uma imagem, basta ao final da primeira iniciar a nova imagem colocando uma barra seguida do nome da segunda imagem, o fim das duas imagens ocorrerá quando for encontrado um "/*" (as duas imagens criadas, uma seguida da outra, são independentes entre elas).

Dataflex fornece também a possibilidade de telas (imagens) de auxílio (HELP). Para isto basta colocá-la após a imagem que receberá a tela de HELP. A tela de HELP deve conter em seu nome a palavra HELP, para que o compilador possa endereçá-la como tal.

Dentro de cada imagem podemos ter desde letras, números, caracteres gráficos e locais para as entradas de dados (janelas).

Exemplo:

```
/tela

     AGENDA                TELEFONICA

NOME : < _____________________________ >

ENDERECO : _____________ No : ___. APTO : ___.

CIDADE : _____ ESTADO : __ CEP : ____

BAIRRO : _____

TELEFONE : (___) _____ DATA DE NASCIMENTO : <__/__/__>

   Use os campos com < > para procura de dados.

/*
```

5.3 JANELAS

Janela é uma área do Dataflex usada para impressão e entrada de informações no banco de dados. Dependendo da instalação de seu Dataflex, o caractere definidor de janelas poderá ser diferente. No nosso caso usaremos o underline "_" para defini-las. O número de underlines usados dará o tamanho da janela criada.

Dataflex permite três tipos de janelas válidas:

- janela numérica: sua formação é feita com underlines e ponto decimal.Neste caso usaremos o número de underlines do tamanho da janela menos um, mais um ponto decimal, isto é, se o tamanho da janela for quatro casas inteiras e nenhuma decimal então usaremos três underlines e um ponto decimal (___.). No caso de números com casas decimais representaremos (___.__).

- janela ASCII: sua formação é feita totalmente com underlines. O tamanho da janela ASCII é definido com o número de underlines usados. Supondo que tivéssemos uma janela ASCII com tamanho de 10 caracteres, teríamos 10 underlines para sua definição (_____).

- janela data: estas janelas obedecem ao padrão americano, europeu ou notação militar de representação de datas, MM/DD/AA, DD/MM/AA, AA/MM/DD respectivamente. Nota-se que o formato básico é dois underlines, uma barra, dois underlines, uma barra e dois underlines (__/__/__), podendo-se usar quatro underlines para notação de ano por completo (__/__/____ notação européia ou americana). No ato da entrada de dados a janela aparecerá com o número total de caracteres que compõem a data, 8 underlines: 6 dos números mais dois das barras que, ao término, pressionando a tecla ENTER, serão preenchidos com as barras.

Como já foi dito as janelas são áreas de uma imagem usada tanto para impressão como para entrada de dados. Numa imagem podemos ter quantas janelas esta comportar. O processamento numa entrada de dados ou numa impressão de dados nas janelas é sempre feita de cima para baixo e da esquerda para a direita, da primeira janela à última.

Como se referenciar a uma janela específica?

Fácil. Já que elas são seqüenciais de cima para baixo e da esquerda para a direita, podemos dar números a elas. Como as janelas fazem parte de imagens, por que não darmos os nomes de acordo com a imagem a que pertencem? Usaremos então o nome da imagem mais o número seqüencial da mesma. Isto nos ajudará muito quando estivermos usando diversas imagens. Para as janelas podemos mover números, datas, letras ou o conteúdo de outras janelas, campos de arquivos, etc. Todo este trabalho de nomenclatura de janelas o próprio Dataflex reconhece, portanto não se faz necessária esta declaração. Adiante veremos outra forma de acesso a estas janelas, como dar nomes a elas e como usá-las como variáveis de memória.

Exemplo:

```
/tela
     AGENDA          TELEFONICA

NOME : <                              >

ENDERECO :                    No : ___.APTO : ___.

CIDADE :                      ESTADO : __ CEP : ____

BAIRRO :

TELEFONE : (___) _____DATA DE NASCIMENTO : <__/__/__>

     Use os campos com < > para procura de dados.
/*
```

Veja que as janelas NOME (tela.1) e ENDEREÇO (tela.2) são janelas ASCII, as janelas No (tela.3) e APTO (tela.4) são numéricas e a janela DATA DE NASCIMENTO (tela. 11) tem o formato data europeu com apenas dois dígitos para o ano.

5.4 BUFFER

É um espaço da memória reservado para um arquivo de dados, no ato de sua abertura, isto é, quando este arquivo é deixado pronto para ser acessado e manipulado. O buffer de registros de arquivos do Dataflex tem o mesmo tamanho do registro do arquivo aberto, pois servirá de ponte para as transações entre a memória e os arquivos em disco.

Como Dataflex pode gerenciar até 255 arquivos de dados ao mesmo tempo, temos também possibilidade de abrir 255 buffers de registro, um para cada arquivo de dados aberto. No exato momento em que Dataflex abre um arquivo de dados, ele deixa disponível, no buffer do arquivo, o primeiro registro encontrado. Um registro só pode ser lido ou gravado se seus dados estiveremresidentes no buffer de registro. Dentro da programação Dataflex, existem comandos que podem limpar e encher o buffer de registros de dados (tais comandos serão estudados posteriormente).

Exemplo:

Supondo que tivéssemos dois arquivos a serem utilizados em uma aplicação, sendo os dois formados por: ARQ1 - NOME (20), ENDEREÇO(20), TELEFONE (7) e ARQ2 - RG (9), CIC (11).

Quando abríssemos os dois arquivos, o Dataflex reservaria espaço na memória, para os buffers do arquivos de dados, da seguinte maneira, colocando imediatamente nestes buffers o primeiro registro disponível do arquivo de dados, se este o tiver; caso contrário, ficará limpo até que algum dado venha a ser movido para o mesmo:

```
ARQ1                    ARQ2
********************    *****
********************    ******
****
```

Veja que os campos numéricos tiveram seu tamanho reduzido à metade, devido à compactação dos dados numéricos. Portanto, o buffer para o arquivo ARQ1 ficaria com espaço reservado de 24 bytes e o espaço destinado ao arquivo ARQ2 seria de 11 bytes. Durante as operações básicas de manipulação de dados dos arquivos (inclusão, alteração, exclusão e consulta), dados serão movidos e limpos destes buffers de registro até que a aplicação tenha seu fim.

5.5 CRIANDO VARIÁVEIS

Em Dataflex, como em qualquer outra linguagem existente, temos a ocorrência de variáveis.

Variáveis são basicamente locais na memória do computador reservadas para o armazenamento de informações numéricas, datas e letras. Algumas linguagens não fazem questão de identificar suas variáveis antes delas serem usadas (é o caso do BASIC onde as variáveis vão sendo criadas à medida que são necessárias), mas Dataflex necessita desta declaração pois, para o compilador, é importante saber a quantidade de espaço da memória que deve ser reservada para a área de trabalho do programa ou aplicação.

Temos em Dataflex, basicamente, quatro tipos de variáveis disponíveis:

- variáveis inteiras: são variáveis numéricas que aceitam números inteiros no limite +- *2,147,483,647* . Números inteiros são aqueles que não possuem parte fracionária. Com estes números podemos realizar as quatro operações básicas (+, -, *, /), mas no caso das divisões os resultados poderão ser reais.
- variáveis reais: são variáveis numéricas que englobam os números inteiros e os fracionários, portanto um grande espectro de números. As variáveis reais fornecidas por Dataflex permitem uma precisão de +- *99,999,999,999,999.9999*. Neste caso também podemos realizar as quatro operações básicas, sendo que todos os resultados serão reais.

- variáveis string: são variáveis que podem conter todas as letras, números, datas e caracteres especiais. Variáveis *string* são conjuntos de caracteres agrupados seqüencialmente. Na definição de variáveis *string* deve-se especificar o tamanho da variável (o número de caracteres máximo que ela terá); caso contrário, será adotado o tamanho 80 caracteres. Podem ser movidos quaisquer tipos de dados para as variáveis *string* e, caso o comando seja adequado, haverá uma transformação de tipos, isto é, o dado movido será convertido para o formato de uma *string*, independentemente de seu tipo. As variáveis *string* também podem sofrer operações, porém são operações diferentes das realizadas com números, pois agora são letras, informações indivisíveis por números. No entanto, como são letras, podem ser manipuladas; podemos inserir, deletar, copiar, juntar e truncar conjuntos de caracteres.

- variáveis data: são variáveis onde serão armazenados dados em formato de uma data. Estas variáveis podem ser somadas e subtraídas de outras variáveis do mesmo tipo, nunca de outro, a não ser que tenham sido movidas para variáveis numéricas ou variáveis string.

Sabendo como Dataflex trata as variáveis em seus programas e aplicações, podemos aprender como criá-las. Segue-se abaixo as sintaxes e explicações necessárias para a criação de variáveis de memória em Dataflex:

- Variáveis Inteiras:

INTEGER (var_int1 < var_int2 ... var_int9 >)

Neste caso podemos criar até nove variáveis inteiras ao mesmo tempo. As variáveis 2 a 9 são opcionais, pois nem sempre você terá 9 variáveis a serem criadas. Nada nos impede de usar uma linha para a definição de cada variável, mas recomenda-se o formato compacto por uma questão de economia do número de linhas do programa-fonte.

- Variáveis Reais:

NUMBER (var_real1 < var_real2 ... var_real9 >)

Como no caso anterior, podemos criar até 9 variáveis numéricas reais.

- Variáveis Data:

DATE (var_data1 < var_data2 ... var_data9 >)

- Variáveis String:

STRING (var_str1 < tam_var1 > < var_str2 tam_var2...var_str9 tam_var9 >)

ou

STRING (var_str1 < var_str2 tam_var2 ... var_str9 >)

No primeiro caso todas as variáveis foram definidas com seus nomes e tamanhos. No segundo caso somente algumas variáveis foram definidas completamente, isto é, com nomes e tamanhos; aquelas onde não foi declarado seu tamanho assumirão o tamanho de 80 caracteres.

Exemplo:

```
INTEGER     diferenca
NUMBER      valor desconto total_a_pagar
STRING      tecla 1
DATE        dat1 dat2 data_sys
```

Dataflex permite também que usemos as janelas como variáveis de memória. Como vimos anteriormente, as janelas recebem seu nome de acordo com a imagem a que pertencem, mais o número a ela atribuído pelo Dataflex. Antes de falar a respeito deste uso de janelas, faz-se necessário o conhecimento de dois comandos para controle de imagens em Dataflex (PAGE, AUTOPAGE e NAME).

- **PAGE**: mostra uma imagem no vídeo ou armazena os dados referentes àquela imagem para quando esta for impressa no vídeo. Os dados referentes à imagem podem ser a posição onde ela será impressa, a cor de impressão da imagem ou as duas situações no mesmo comando.

Sintaxe:

```
PAGE imagem

PAGE SET imagem < AT lin col  COLORS  cor_jan cor_imag  >
```

No primeiro caso o comando imprime a imagem na posição comum: linha 0, coluna 0 (HOME) a não ser que já tenha sido dado algum PAGE SET anteriormente. Com este formato, a imagem será impressa na posição assinalada, podendo haver superposição de imagens.

No segundo formato do comando indica-se a posição onde deverá ser impressa a imagem, podendo-se também escolher a cor da janela e da imagem (monitores cromáticos). Este efeito não é tão visível em monitores monocromáticos, caso as cores forem de matizes próximos, mas com cores contrastantes o efeito poderá ser bem razoável.

Com este comando recomenda-se o endereçamento de cada janela usada, pois ele poderá não fazer o endereçamento automático das mesmas.

- **AUTOPAGE:** faz o endereçamento automático de todas as janelas ou de parte delas. Com este comando não precisamos endereçar janela a janela as entradas de dados ou as impressões das mesmas.

Sintaxe:

AUTOPAGE imagem < num_jan >

Usando-se apenas AUTOPAGE e o nome da imagem, estaremos imprimindo a imagem no vídeo e endereçando automaticamente as janelas da mesma. Usando-se o num_jan indica-se a partir de qual janela desejamos ter o endereçamento automático.

O comando AUTOPAGE usado como meio para impressão de uma imagem pode ser alterado usando-se um comando PAGE SET com argumentos de posição de vídeo e cor de janelas e imagem, isto é, através de um comando PAGE SET imagem, anterior ao AUTOPAGE imagem, pode-se formatar o AUTOPAGE para imprimir a imagem na linha, coluna, cor de janela e cor de imagem desejadas.

Com o uso dos dois comandos juntos podemos gerenciar várias imagens ao mesmo tempo, podemos usar imagens como janelas (windows) para telas de entrada de dados, usar telas de HELP paralelas ao processamento normal das janelas de uma determinada imagem, etc.

- **NAME:** sua função é a de mudar o nome da janela de imagem.num_jan para um nome usual, qualquer.

Sintaxe:

1) NAME imagem.num_jan nome
2) NAME imagem nome1 < nome2 ... nome7 >
3) AUTOPAGE imagem num_jan
NAME nome1 < nome2 ... nome 9 >

Na primeira sintaxe dá-se um nome a uma janela específica.

Na segunda sintaxe, Dataflex dará nome seqüencialmente às janelas da imagem referida (o endereçamento das janelas será de cima para baixo e da esquerda para a direita).

O terceiro caso é o mais comum, pois com ele conseguimos dar nome seqüencialmente para as janelas a partir de uma janela em específico de uma determinada imagem. Basicamente é uma combinação das duas anteriores.

Exemplo:

```
NAME            tela.1 nome
NAME            tela.2 endereco
NAME            tela.1 nome endereco numero apto

AUTOPAGE        tela 1
NAME            nome endereco numero apto
AUTOPAGE        tela 5
NAME            cidade estado cep bairro
AUTOPAGE        tela 9
NAME            ddd fone data_de_nascimento
```

Como dizíamos, podemos usar janelas como variáveis de memória, com o conhecimento destes comandos citados anteriormente podemos entender como isto ocorre.

Após o " /* " (indicando o término da área de imagens dos programas e aplicações Dataflex), temos a região usada normalmente para definição de variáveis, abertura de arquivos e um pré-processamento antes do programa principal. Nesta região, podemos definir, ou melhor, dar nomes às janelas e assim usá-las como variáveis de memória.

Como isto ocorre?

Usaremos o terceiro formato do comando NAME:

AUTOPAGE imagem num_jan
NAME nome1 < nome2 ... nome9 >

Dá-se um AUTOPAGE na tela selecionada mais o número da janela pela qual se deseja começar o endereçamento e nomenclatura (no caso de ser a primeira janela usa-se o número 1 para num_jan).

Inicia-se uma série de comandos NAME, até que complete as janelas ou até que não se queira mais nomear as janelas. Podemos também truncar a nomenclatura das janelas mas precisaremos de outro comando AUTOPAGE para a janela seguinte de onde se parou.

Exemplo:

```
/tela
     AGENDA              TELEFONICA

NOME : < _____ >

ENDERECO : _____ No : ____.APTO : ____.

CIDADE : _____ ESTADO : __ CEP : _____

BAIRRO : _____
TELEFONE : (____) _____DATA DE NASCIMENTO : <__/__/__>

     Use os campos com < > para procura de dados.
/*
  AUTOPAGE tela 1
  NAME nome endereco numero apto
  AUTOPAGE tela 5
  NAME cidade estado cep bairro
  AUTOPAGE tela 9
  NAME ddd fone data_de_nascimento
```

PARTE II
CRIANDO, ALTERANDO E USANDO UM BANCO DE DADOS

Com os conhecimentos até aqui adquiridos já temos condições para, lentamente, irmos criando e usando uma aplicação, gerada pelo Dataflex. Veremos a seguir os passos básicos e necessários para esta criação.

- Criar um arquivo usando uma imagem.

Usando um editor de textos qualquer que gere um arquivo-texto em código ASCII puro, desenharemos a imagem do arquivo, onde definiremos os tamanhos e tipos de campos do arquivo de dados.

- Criando a estrutura do arquivo de dados através de uma imagem.

Usando o módulo AUTODEF e a imagem anteriormente criada, definiremos o arquivo de dados e os índices pelos quais este arquivo e seus registros serão acessados. Paralelamente a isto será criado um programa-fonte (programa de aplicação), através das informações fornecidas durante o uso deste módulo.

- Compilando o programa-fonte.

Nesta fase o compilador Dataflex transformará automaticamente o programa-fonte em um reduzido e executável programa-objeto, que ocupará menos espaço; será executado mais rapidamente e garantirá maior segurança dos dados a serem manipulados.

- Entrada de dados, consultas e manipulação.

Usando a aplicação recém-criada, poderemos realizar inclusões, alterações, deleções e consultas de dados no arquivo recém-criado.

Todos os passos citados acima serão discutidos e exemplificados no próximo capítulo, como usar AUTODEF.

PARTE II
CRIANDO, ALTERANDO E USANDO UM BANCO DE DADOS

CAPÍTULO 6

AUTODEF

Discutiremos, neste capítulo, os passos acima descritos para a criação de uma aplicação a partir de uma imagem. Iniciaremos com a criação da imagem, seguida do uso do AUTODEF e por final o uso da aplicação gerada.

6.1 CRIANDO UMA IMAGEM

Como foi dito, para criar uma imagem precisamos de um editor de textos que gere arquivos ASCII puros, isto é, arquivos somente com caracteres do padrão ASCII. Dataflex fornece, juntamente com outros módulos do sistema, um editor de textos. Neste livro não falaremos a respeito deste editor, porém seus comandos constarão do Apêndice III, portanto ficará a seu critério a escolha do editor a ser usado.

Uma vez selecionado o editor de textos passaremos à criação da imagem. Primeiramente devemos dar um nome ao arquivo-texto que guardará a imagem; este nome é de extrema importância, pois será o nome do arquivo de dados a ser criado a partir da imagem. Inicia-se o processo de criação da imagem colocando-se uma barra "/" na posição 0 0 (HOME, coluna 0 e

linha 0) de seu monitor de vídeo, seguido do nome da imagem. O nome da imagem poderá ter até 80 caracteres. Em seguida, desenha-se a imagem e ao final colocamos, na primeira coluna da última linha da imagem uma barra seguida de um asterisco "/*", indicando fim da mesma.

Passemos ao desenho da imagem do nosso arquivo de dados.

Para um melhor entendimento do funcionamento deste módulo, usaremos um exemplo simples, e à medida que formos progredindo o exemplo irá crescendo, até tornar-se uma aplicação completa.

O exemplo será um cadastro pessoal composto por:

- nome completo
- endereço (número, apto)
- cidade
- estado
- bairro
- cep
- telefone (DDD e numero)
- data de nascimento

Segue-se então a tela de exemplificação (ela poderá ser digitada em qualquer editor de textos como mencionado acima. Preste atenção no número de caracteres underline "_" usados na construção de cada janela ASCII, no caso de janelas numéricas, a quantidade de underlines e o ponto decimal e, em janelas data, o formato em que é escrita).

/tela

AGENDA TELEFONICA

NOME : < _____ >

ENDERECO : _____ No : ___. APTO : ___.

CIDADE : _____ ESTADO : __ CEP : _____

BAIRRO : _____

TELEFONE : (___) _____ DATA DE NASCIMENTO : <__/__/__>

Use os campos com < > para procura de dados.

/*

Obs1: A configuração de cada janela:

Janela	número de underlines	número de pontos decimais
- NOME	30	0
- ENDERECO	30	0
- NUMERO (No)	04	1
- APTO	04	1
- CIDADE	20	0
- ESTADO	03	0
- CEP	02	0
- BAIRRO	20	0
- TELEFONE (DDD + numero)	13	4 (DDD) + 9 (numero)0
- DATA DE NASCIMENTO (DD/MM/AA)		0

Obs2: Como você pode ver, em uma imagem podemos colocar qualquer caractere disponível, com exceção de conjuntos de underlines "__" e underlines combinados com pontos "_._" ou barras "_/_", pois serão reconhecidos como janelas.

Obs3: Usaremos a convenção de colocar os sinais "< >" para indicar campos indexados, isto é, campos onde podem ser realizadas procuras sequenciais ou randômicas pelos índices.

Terminada a criação da imagem, grave-a em disco e então passaremos para o próximo passo. Devido ao fato de que o nome atribuído a esta imagem seja um nome de arquivo de dados, devemos dar o nome de acordo com o tipo de dados que este arquivo irá armazenar. Sugestão: " CADTEL.SCR ", por se tratar de um cadastro de telefones. A extensão ".SCR" é individual, mas serve para indicar que este arquivo guarda a imagem de um arquivo de dados. Não é obrigatória sua aceitação, pois trata-se apenas de uma convenção particular.

6.2 USANDO O AUTODEF

AUTODEF é um módulo, um programa, que será usado para criar facilmente uma definição de arquivo (incluindo a definição de índices), juntamente à criação de um programa-fonte de aplicação de entrada de dados para o arquivo. A definição do arquivo consta: do tamanho e tipo dos campos e informações a respeito dos índices. A nível de Bancos de Dados será formado um Dicionário de Dados, isto é, uma relação de nomes de arquivos, campos e índices.

Existem dois meios de acessar o módulo AUTODEF:

1) Se você estiver usando o Sistema de Menus:

- vá para o menu principal, escolha a opção Utilitários Dataflex e aparecerá um novo Menu;
- escolha a opção Criando uma Aplicação (AUTODEF) e aparecerá na última linha uma janela para entrada do nome da imagem da qual será gerada uma aplicação;

- digite o nome da imagem (CADPES.SCR no nosso caso).

2) Se você estiver no sistema operacional:

- digite AUTODEF ou AUTODEF nome_imagem.

Independente do modo como foi acessado o módulo AUTODEF, uma tela parecida com esta aparecerá:

DATAFLEX Auto File Definition Program

AUTODEF irá criar os arquivos de dados e o programa-fonte baseado na imagem fornecida pelo Usuário. Esta imagem normalmente foi criada com o editor de textos. O nome deste arquivo não deve conter extensões. AUTODEF irá criar arquivos com extensões que conterão dados, índices e programas. Digite o nome da imagem ou pressione <ENTER> para sair.

QUAL O NOME DO ARQUIVO DE IMAGEM? CADTEL.SCR

Em seguida à digitação do nome da imagem, outra tela será apresentada:

DATAFLEX Auto File Definition Program

SUMARIO ARQUIVO:

NOME DA IMAGEM	CADTEL.SCR
NOME BASE	CADTEL
NOME PARA O DATAFLEX	CADTEL
NOME DO PROGRAMA	CADTEL.FRM

ARQUIVO "CADTEL" ASSINALADO C/ NUMERO 2

QUAL O NUMERO MAXIMO DE REGISTROS QUE "CADTEL" PODE TER ? 100

Esta tela mostra os nomes com que o arquivo CADTEL poderá ser acessado através deste e de outros módulos Dataflex. O NOME BASE é o nome que será usado como referência a nível de sistema operacional e o

NOME PARA O DATAFLEX constará no Dicionário de Dados, na divisão reservada a nomes de arquivos de dados Dataflex. A numeração dada, também será usada pelo Dicionário para futuras referências ao arquivo de dados. Esta numeração poderá substituir o nome do arquivo em determinadas condições.

Pede-se, então, uma estimativa do número máximo de registros que o arquivo poderá ter. Esta estimativa poderá ser aleatória ou baseada em cálculos anteriores, o número máximo de registros possíveis para o arquivo de dados poderá ser alterado via programa (ver manual Dataflex volume 2, comando ZEROFILE nome_arquivo).

Após a digitação deste número, outra tela será apresentada:

AGENDA TELEFONICA

NOME : < ********** >

ENDERECO : _____ No : ___. APTO : ___.

CIDADE : _____ ESTADO : __ CEP : _____

BAIRRO : _____

TELEFONE : (___) _____ DATA DE NASCIMENTO : <__/__/__>

Use os campos com < > para procura de dados.

NOME DO CAMPO PARA CAMPO_1 : NOME

Neste ponto do AUTODEF estamos dando nome aos campos do arquivo CADTEL e formando o Dicionário de Dados. Este processo ocorrerá para todas as janelas, uma a uma, de cima para baixo, da esquerda para a direita até o fim das janelas,sendo pedido o nome dos campos seqüencialmente desde o campo 1, NOME, até o último campo, de número 11: DATA_NASCIMENTO. Os nomes atribuídos aos campos neste exemplo foram dados de acordo com os conteúdos que guardarão, podendo ser alterados se desejar. Com a nomenclatura e numeração dos campos do arquivo terminada poderemos passar à próxima tela : definição dos índices.

Pressione "C" p/ Continuar ou "R" p/ Refazer

Esta mensagem dá a você a chance de corrigir algum nome de campo escrito errado; caso tudo esteja certo tecle "C".

Na próxima tela o AUTODEF entra no processo de formação dos índices. Serão dados os nomes de campos, devidamente numerados em ordem seqüencial crescente, e será feita uma pergunta a respeito da criação do primeiro índice:

1: NOME	2: ENDERECO	3: NUMERO
4: APTO	5: CIDADE	6: ESTADO
7: CEP	8: BAIRRO	9: DDD
10: FONE	11: DATA_NASCIMENTO	

DEFINA O INDICE 1
QUAL(IS) CAMPO(S) SERA(O) INDEXADO(S)? .....: 1
OS DADOS DESTE INDICE SERAO UNICOS? <N> N

DEFINA O INDICE 2
QUAL(IS) CAMPO(S) SERA(O) INDEXADO(S)? .....: 11
OS DADOS DESTE INDICE SERAO UNICOS? <N> N

Com estas duas perguntas e respostas criamos dois índices, um segmento cada um, não exclusivos (podendo haver repetição destes dados em outros registros). O primeiro índice ordena o arquivo de dados pelo nome em ordem alfabética. O segundo índice ordena o arquivo por data de

nascimento em ordem cronológica crescente. Dataflex criará então dois arquivos de índices : CADTEL.K1 e CADTEL.K2 contendo informações da ordenação do arquivo CADTEL.

Para sairmos deste processo de formação de índices basta teclarmos ENTER; a partir daí, como todos os dados a respeito do arquivo de dados já foram fornecidos, o AUTODEF criará uma definição de arquivo automaticamente, juntamente com um programa-fonte com o nome da imagem ("CADTEL") e terminação ".FRM".

Surgirá, então, a seguinte mensagem:

O PROGRAMA DEVE SER COMPILADO ANTES DE USAR.
COMPILA CADTEL.FRM JA ? (S ou N):

Se respondermos "S" a esta pergunta, estaremos certos de que mais nenhuma modificação será necessária; portanto, a aplicação estará pronta para ser compilada, transformada em um código executável pelo Dataflex. Mas se a resposta for "N", significa que talvez queiramos modificar a aplicação, lay-out ou determinadas áreas da aplicação. As modificações a serem relizadas em lay-outs de tela não afetam a futura compilação e execução da aplicação. No entanto, as modificações a nível de programa: instruções, estruturas, arquivos, opções de formatação de janelas, opções de controle de buffer e comandos suplementares, poderão trazer erros de compilação se não forem colocadas corretamente.

6.3 ALTERANDO A APLICAÇÃO

Como falamos anteriormente, pode-se alterar a aplicação gerada com o uso do AUTODEF. As modificações inseridas nesta aplicação são para que a entrada de dados funcione perfeitamente, isto é, funcione como uma aplicação de inclusão, de alteração, de deleção e consulta de dados. A

aplicação gerada é criada basicamente sem estes recursos, portanto, faz-se necessário esta implementação.

Dentro destas implementações podemos citar algumas opções de formatação de janelas, de processos de procura e de checagem dos dados. As opções aqui referenciadas serão detalhadamente explicadas no próximo capítulo Macro Enter:

- Autofind : opção de procura automática no índice principal.

- Capslock : opção de formatação de janela ASCII - transforma as letras minúsculas em maiúsculas.

- Range = n1,n2 : opção de checagem de janela numérica - verifica se o conteúdo da janela está entre n1 e n2, inclusive (válida também para datas).

- Check = "str1%str2*str3#str4!...|str n" : tem a mesma função do Range = só que verifica se o conteúdo de uma janela ASCII está contida na string do comando (entre aspas "").

- Displayonly : imprime o conteúdo de um campo de arquivo na janela e impede os dados das janelas de irem para o buffer (impede a entrada de dados naquela janela).

Todas estas opções devem vir entre chaves "{ }" após os comandos de entrada de dados (comandos ENTRY seguidos do nome do arquivo e do nome do campo). Podemos ter mais de uma opção por comando ENTRY, basta separá-las por vírgulas ",". Todas as implementações usadas devem ser acrescentadas no programa gerado pelo AUTODEF. Use o editor de textos de sua preferência para as implementações; ao final grave o programa no disco (não se esqueça de que o programa se chama " CADTEL.FRM").

A seguir, a listagem da aplicação gerada e a listagem da aplicação implementada das opções de procura e formatação de janelas.

/tela

AGENDA TELEFONICA

NOME : < _____ >

ENDERECO : _____ No : ___. APTO : ___.

CIDADE : _____ ESTADO : __ CEP : _____

BAIRRO : _____

TELEFONE : (___) _____ DATA DE NASCIMENTO : <__/__/__>

Use os campos com < > para procura de dados.

```
/*
  PAGE tela
  OPEN CADTEL
  ENTER CADTEL
  AUTOPAGE tela
    ENTRY CADTEL.NOME
    ENTRY CADTEL.ENDERECO
    ENTRY CADTEL.NUMERO
    ENTRY CADTEL.APTO
    ENTRY CADTEL.CIDADE
    ENTRY CADTEL.ESTADO
    ENTRY CADTEL.CEP
    ENTRY CADTEL.BAIRRO
    ENTRY CADTEL.DDD
    ENTRY CADTEL.FONE
    ENTRY CADTEL.DATA_NASCIMENTO
    RETURN
  ENTEREND
  ABORT
```

Dataflex reconhece dentro de seus programas-fonte as duas barras "//" como indicação de um comentário. O comentário deve ser colocado à direita das duas barras e pode ser de qualquer tipo (letras, números, datas, outras barras, enfim qualquer caractere disponível), com exceção de caracteres de controle.

/tela

AGENDA TELEFONICA

NOME : < _____ >

ENDERECO : _____ No : ____. APTO : ____.

CIDADE : _____ ESTADO : __ CEP : _____

BAIRRO : _____

TELEFONE : (____) _____ DATA DE NASCIMENTO : <__/__/__>

Use os campos com < > para procura de dados.

```
/*                          // indica fim da imagem
PAGE tela                   // imprime a tela
OPEN CADTEL                 // deixa o arquivo CADTERL pronto para uso
ENTER CADTEL
AUTOPAGE tela               // enderecamento automatico das janelas
   -ENTRY CADTEL.NOME {AUTOFIND,CAPSLOCK}// opção de
   -ENTRY CADTEL.ENDERECO {CAPSLOCK}// procura no indice 1
    -ENTRY CADTEL.NUMERO            // e entrada so em
    -ENTRY CADTEL.APTO              // letras maiusculas
    -ENTRY CADTEL.CIDADE            {CAPSLOCK}
    -ENTRY CADTEL.ESTADO            {CAPSLOCK}
    -ENTRY CADTEL.CEP
    -ENTRY CADTEL.BAIRRO            {CAPSLOCK}
    -ENTRY CADTEL.DDD
    -ENTRY CADTEL.FONE
    -ENTRY CADTEL.DATA_NASCIMENTO {AUTOFIND} // procura
```

```
-RETURN                    // pelo indice 2
ENTEREND
ABORT                      // fim
```

Não estranhe os comandos não conhecidos, pois eles serão devidamente explicados nos capítulos seguintes.

Pronto. Agora basta compilarmos esta aplicação modificada e poderemos usá-la à vontade.

Como compilar a aplicação?

Simples. Se você está usando o Sistema de Menus vá para o Menu Principal e escolha a opção: Utilitários Dataflex. Neste menu escolha: Compilando uma Aplicação. Surgirá, então, uma janela para a entrada do nome da aplicação a ser compilada. Digite o nome da aplicação ("CADTEL.FRM"). Caso esteja no sistema operacional, digite COMP e tecle ENTER. Surgirá a seguinte tela:

```
DATAFLEX COMPILER (c) 1985 DATA ACCESS CORPORATION, All
Rights Reserved.
NOME ARQUIVO: CADTEL.FRM
```

Após digitarmos o nome do arquivo a ser compilado, o compilador Dataflex iniciará a compilação do programa. Abaixo está a impressão da compilação do arquivo CADTEL.FRM:

```
FORM>/TELA
Carregando1, TELA
```

```
    AGENDA              TELEFONICA

NOME : < _____ >

ENDERECO : _____ No : ___. APTO : ___.

CIDADE : _____ ESTADO : __ CEP : ____

BAIRRO : _____

TELEFONE : (___) _____ DATA DE NASCIMENTO : <__/__/__>

Use os campos com < > para procura de dados.

/*                          // indica fim da imagem
FORM>/*                     // INDICA FIM DA IMAGEM
MEMORIA DISPONIV: 178674
1>PAGE tela                 // imprime a tela
2>OPEN CADTEL               // deixa o arquivo CADTERL pronto para uso
INCLUINDO ARQ.: CADTEL.FD
3>ENTER CADTEL
75>AUTOPAGE tela            // endereçamento automático das janelas
75> ENTRY CADTEL.NOME {AUTOFIND,CAPSLOCK} // opção de
76> ENTRY CADTEL.ENDERECO {CAPSLOCK} // procura no índice 1
77> ENTRY CADTEL.NUMERO                 // e entrada só em letras
78> ENTRY CADTEL.APTO                   // maiúsculas
79> ENTRY CADTEL.CIDADE          {CAPSLOCK}
80> ENTRY CADTEL.ESTADO          {CAPSLOCK}
81> ENTRY CADTEL.CEP
82> ENTRY CADTEL.BAIRRO          {CAPSLOCK}
83> ENTRY CADTEL.DDD
84> ENTRY CADTEL.FONE
85> ENTRY CADTEL.DATA_NASCIMENTO {AUTOFIND}// procura
86> RETURN                              // pelo índice 2
87>ENTEREND
89>ABORT                    // fim
90>
MACRO:
```

MEMORIA DISPONIV: 174241
COMANDOS GERADOS....: 89
ERROS.: 0
TOTAL SIMBOLOS:156

6.4 USANDO A APLICAÇÃO

A partir do momento em que já temos a aplicação devidamente gerada e compilada, podemos passar para o próximo passo: usar a aplicação gerada.

Como fazer para executar esta aplicação?

Simples (até parece que tudo é tão simples!).

Se você estiver usando o Sistema de Menus: vá ao Menu Principal e e escolha a opção: Utilitários Dataflex. Escolha, deste menu, a opção: Executando uma Aplicação. Aparecerá uma janela para a entrada do nome da aplicação a ser executada ("CADTEL" sem a terminação, pois, como vimos, um arquivo compilado possui terminação ".FLX", não sendo necessária sua declaração).

Caso esteja no sistema operacional, digite FLEX e tecle ENTER ou digite FLEX nome arquivo compilado ("FLEX CADTEL"). Irá aparecer a tela criada no primeiro passo de criação de uma aplicação, somente com a diferença que os pontos decimais das janelas numéricas foram substituídos por underlines, idem as barras da data de nascimento. Não se preocupe, pois, aquelas janelas continuam as mesmas, com a diferença que o programa reconhecerá uma entrada errada assim que ela ocorrer.

6.4.1 Flex-Keys

Dentro de um enfoque de facilidade de uso, muitos computadores possuem em seus teclados aquilo que se denomina Teclas de Função. Teclas de Função podem ser alocadas em qualquer parte ou região do teclado, como também podem requerer um chaveamento de mais de uma tecla, tudo depende de qual computador você está usando.

No nosso caso usamos um equipamento compatível com IBM PC, com 10 teclas de função (F1 a F10) e um teclado numérico reduzido.

Muitos softwares utilizam-se destas teclas de função para processos de manipulação de dados (procuras, deleção, etc). As teclas de função do Dataflex estão um passo adiante de alguns softwares, são flexíveis e programáveis, de onde recebem seu nome FLEX-KEYs.

Para sua conveniência a configuração que será usada neste livro é a configuração padrão de Flex-Keys transcrita no Manual Dataflex:

função	Flex-Key	função
HELP	- F1 F2	- BACK WIND
SUPERFIND	- F3 F4	- PRINT SCR
CALCULATE	- F5 F6	- DEL RECORD
USER1	- F7 F8	- USER2
CLEAR WIND	- F9 F10	- SAVE RECORD

tecla	função
TAB	Find (procura)
Pg Dn	Previous Rec (registro ant)
Pg Up	Next Rec (prox registro)
Ins	Insert (insere caractere)
Del	Delete (deletar caractere)
ESC	Escape (fuga)
ENTER	Return (confirmação)

Na instalação do Dataflex, você pode alterar a configuração destas teclas, mas basicamente elas se apresentam da seguinte forma:

função	Flex-Key	função
HELP	ctrl Q ctrl J	BACK WIND
SUPERFIND	ctrl F ctrl O	PRINT SCR
CALCULATE	ctrl C ctrl D	DEL RECORD
USER1	ctrl U ctrl Y	USER2
CLEAR WIND	ctrl A ctrl S	SAVE RECORD

tecla	função
TAB	Find
ctrl P	Reg anterior
ctrl N	Prox registro
ctrl Z	Ins caractere
ctrl X	Del caractere
ESC	Escape
ENTER	ENTER

6.4.2 Processos Básicos de Manipulação de Dados

Dataflex nos dá, a partir da tela de entrada de dados, diversos métodos de manipulação e consulta aos dados de um ou mais arquivos. Podemos incluir dados, procurar registros, achá-los, alterá-los e gravá-los novamente no arquivo de dados, deletá-los ou ignorá-los. Sempre que a tela de entrada de dados for mostrada, novos registros poderão ser incluídos, alterados, consultados ou excluídos.

Para as operações básicas de manipulação de dados, necessitamos conhecer como Dataflex executa cada um destes processos. Abaixo estão descritos os quatro processos básicos para manipulação de dados (inclusão, procura, edição ou alteração e exclusão). Esses quatro processos são independentes, isto e', você pode executá-los em qualquer ordem ou ao mesmo tempo. Preste atenção nas operações de edição e exclusão, pois será necessária uma operação de procura anterior.

- *Operação de inclusão de registros*: É a primeira operação a ser realizada com um arquivo de dados vazio (não se pode alterar, consultar ou deletar dados que não existam). Irá aparecer a imagem criada, totalmente limpa com as janelas de entrada de dados identificadas por

underlines. Podemos começar as inclusões com dados aleatórios, como por exemplo:

1) ANA CLAUDIA G PEREIRA, RUA PASSOS DA PATRIA, 125, SAO PAULO, SP, 01234, VILA MARIANA, 011, 841-3467, 19/02/67.

2) ROMERO MARIA SOARES, RUA SAO CARLOS DO PINHAL, 36, apto 2, ALTAMIRA, SP, 01654, CENTRO, 07/07/68.

3) RODINEI CARLOS DA SILVA, ALAMEDA DOS AICAS, 165, SAO PAULO, SP, 01232, IBIRAPUERA, 011, 62-2365, 18/11/64.

Antes de iniciar a digitação dos dados fictícios são necessários alguns esclarecimentos:

- Se os dados digitados não preencherem completamente todos os underlines da janela, deve-se teclar um ENTER ao final destes dados, para que se possa passar para a próxima janela.

- Caso os dados da janela preencherem completamente a janela, Dataflex passará automaticamente para a próxima janela (este efeito dependerá se o seu Dataflex tiver sido instalado com Autoreturn, isto é, passagem para a próxima janela de entrada de dados sem que o operador ou o digitador tenha de confirmar a entrada com um ENTER).

- Se os dados forem numéricos, não digite letras como entrada de dados, pois um BEEP será ouvido indicando tipo de dados inválidos para aquela janela.

- Se a janela for uma data, não coloque as barras, Dataflex ao final (ENTER) as colocará. Uma janela data apresenta dois underlines a mais, reservados para as barras, por isso ao final da entrada dos seis (6) números tecle ENTER.

- Se a janela for ASCII, todos os caracteres são válidos, inclusive números, pontos decimais, hífen, barras, etc.

- Caso você tenha acabado de digitar os dados da janela, teclado ENTER e descoberto que os dados digitados estavam errados, use a Flex-Key Back-Field (F2) (ou use a seta para cima de seu teclado numérico), para retornar ao campo anterior. Este procedimento é válido para qualquer campo em que você esteja, com exceção do primeiro.

- Ao final da digitação da última janela da imagem, confira todos os dados digitados. Caso encontre algum errado, use F2 para chegar até ele, corrija-o, tecle ENTER para passar para a próxima janela e proceda deste modo até a última janela da imagem. Tecle ENTER e o Dataflex irá limpar a tela automaticamente para você, gravará os dados das janelas no arquivo de dados, atualizará os arquivos de índices, limpará os buffers de registro e estará pronto para receber outra entrada de dados.

- Caso, em qualquer parte da digitação dos dados, você desejar reentrá-los, isto é, perder os dados já digitados nas janelas anteriores, por motivo de erros ou engano quanto aos dados, bastará teclar a Flex-Key Clear Windows (F9) para que as janelas sejam limpas juntamente com os buffers de registro, deixando a imagem pronta para receber os dados corretos.

Após estes simples mas não menos importantes esclarecimentos, podemos iniciar a digitação dos dados.

Segue-se agora as telas reduzidas de duas entradas de dados:

NOME : < ANA CLAUDIA G PEREIRA >
ENDERECO : RUA PASSOS DA PATRIA No : 125APTO :0
CIDADE : SAO PAULO ESTADO : SP CEP : 01234
BAIRRO : VILA MARIANA
TELEFONE : (011) 841-3467 DATA DE NASCIMENTO : < 19/02/67 >

NOME : < ROMERO MARIA SOARES >
ENDERECO : RUA SAO CARLOS DO PINHAL No : 36APTO :2
CIDADE : ALTAMIRA ESTADO : SPCEP : 01654
BAIRRO : CENTRO
TELEFONE : ( ) DATA DE NASCIMENTO : < 07/07/68 >

- *Operação de Procura de registro*: Você pode procurar um registro por qualquer campo indexado do arquivo de dados. Um campo indexado é representado por uma janela entre os sinais "< >". Para uma procura de dados eficiente, Dataflex fornece diversos métodos, como o uso das Flex-Keys SuperFind, Find, Next Record e Previous Record, ou ainda comandos de procura de dados (Findkey, Searchkey e Find), que serão estudados no capítulo 19 (Comandos de Manipulação em arquivos). Para que possamos realizar procuras e consultas a arquivos de dados, necessitamos, também, de alguns esclarecimentos de como Dataflex realiza estes processos:

- Como foi dito, pode-se procurar registros através das informações contidas nas janelas associadas a campos indexados. Para isso deveríamos entrar toda a informação na janela e fazer uso de uma Flex-Key como Find (TAB) ou SuperFind (F3). Dataflex permite também uma procura por pedaços de informação, isto é, usando-se a primeira letra do nome, um conjunto de letras ou números, o dia da data de nascimento, etc. (seguidas de uma das duas Flex-Keys TAB ou F3). Nestes casos Dataflex irá procurar a primeira ocorrência do pedaço de informação fornecida, no campo pesquisado. Caso seja encontrado um registro que satisfaça, em parte, os dados usados na operação de procura, Dataflex mostrará todos os dados do registro nas janelas a eles associadas. Estes dados achados poderão ser maiores ou iguais aos dados fornecidos a procura, portanto não se assuste se os dados achados forem diferentes daqueles fornecidos, pois Dataflex fará uma procura por um registro maior, igual as informações fornecidas. Caso não ache nenhum registro que satisfaça as informações dadas, Dataflex irá mostrar o último registro do arquivo, ordenado pelo índice.

- Pode-se, também, fazer procuras de registros de forma seqüencial pelo índice. Usam-se as Flex-Keys Previous Record (Pg Up) e Next Record (Pg Dn), para uma varredura de registro a registro do arquivo, ordenado pelo índice do campo, por onde estamos procurando. Este processo pode ser feito antes ou após o método anterior (pesquisa aleatória através de chaves parciais). Este processo tem limites: o

número de registros do arquivo de dados. Em se tratando de uma procura seqüencial, esbarra-se sempre no início ou no fim de arquivo, dependendo do tamanho de seu arquivo. Quando isto ocorrer haverá uma mensagem de erro, seguida de um BEEP, indicando procura após fim de arquivo ou procura antes do início do arquivo. Estes erros são comuns e não interferem em nada no processamento usual.

A operação de procura de registros, seja ela seqüencial ou aleatória, é operação primordial e básica para que as operações de edição e exclusão possam ser executadas corretamente, pois somente após achado o registro que deve ser manipulado, poderemos alterá-lo ou excluí-lo.

Operações de procura de registro têm uma segunda finalidade: consultas aos arquivos de dados. Estas operações não indicam que estes dados devam ser alterados ou excluídos, mas apenas uma consulta a uma relação de informações ordenadas alfabeticamente ou numericamente.

- *Operação de Alteração*: Editar um registro significa achá-lo, mudar as informações contidas em seus campos e reescrevê-lo novamente no arquivo de dados. O método de procura de registro pode ser qualquer um daqueles anteriormente mencionados. Tenha certeza absoluta quanto aos dados a serem alterados, verifique se o registro mostrado é exatamente aquele que será modificado e tenha certeza dos dados que foram introduzidos durante esta operação. Pode-se realizar quantas operações de edição forem necessárias, não há diferença entre um registro alterado e um corretamente incluído.

- *Operação de exclusão*: Excluir um registro de um arquivo significa remover, irrevogavelmente, aquela informação de um meio de armazenamento de dados, juntamente com informações do arquivo de índices sobre a mesma. Não se pode excluir de um campo somente um registro completo. Deve-se achar o registro a ser deletado, verificar se realmente é o registro desejado e usar a Flex-Key Delete Record (F6) em qualquer ponto da imagem. A exclusão deste registro atualizará o arquivo de dados, de onde serão removidas todas as informações a respeito daquele arquivo, e atualizará também os arquivos de índices.

Uma vez excluído um registro, ele não poderá nunca ser recuperado. Após a exclusão de um registro, todas as janelas serão limpas deixando a imagem disponível para qualquer tipo de operação.

Como sair da aplicação?

Se você já usou sua aplicação, já incluiu, consultou, alterou ou deletou os dados necessários, tecle ESC (escape). Após a teclagem, a tela será limpa e será devolvido o controle ou para o sistema operacional (se você tiver executado a aplicação com o comando FLEX nome aplicação) ou para o Sistema de Menus, no caso o Menu Utilitários Dataflex.

CAPÍTULO 7
MACRO ENTER

7.1 O que é Macro Enter

A Macro Enter é um conjunto de rotinas e sub-rotinas usadas basicamente para a manipulação de um único registro por arquivo. Como Dataflex pode gerenciar até 250 arquivos de dados ao mesmo tempo, você terá acesso a 1 registro de cada arquivo de todos os arquivos de dados devidamente abertos.

A Macro Enter controla para você desde o cursor até os processos de manipulação de arquivos (inclusão, alteração, deleção e procura de registros dentro dos arquivos abertos).

7.2 Estrutura da Macro

Como foi dito anteriormente, a Macro Enter é um grupo, um conjunto de rotinas e sub-rotinas, que se encontram arranjadas da seguinte forma:

- Seção Pré-Enter
- Seção Enter arquivos
- Seção Entry
- Enterend
- Seção Post-Entry

O núcleo central da Macro Enter encontra-se entre as seções ENTER e ENTEREND, pois nesta área acontecerão todos os processos de manipulação de arquivos. A seção restante ficará reservada para as outras rotinas do programa juntamente com as sub-rotinas da Macro Enter.

Seção Pré-Enter:

É uma área livre, não necessariamente pertencente à Macro, pois aqui poderão ser colocadas as imagens, as definições de variáveis, os primeiros controles de telas, abertura de arquivos, enfim, qualquer procedimento desejado (qualquer comando usado em Dataflex é válido nesta área). Exemplo:

```
/tela1                                  // indica inicio da imagem

      Nome : _____
      Endereco : _____
      Cidade: _____                     Estado : __
      Cep : _____
      Data de Nascimento: __/__/__

/*                                      // indica fim das imagens
STRING tecla 1 resposta 10              // definindo variavel string
INTEGER idade                           // definindo variavel inteira
DATE dia_de_hoje                        // definindo variavel tipo data
OPEN arq1                               // deixa o arq1 pronto para uso
OPEN arq2
...
etc...
```

Seção Enter arquivos:

Sintaxe:

ENTER arquivo1 < arquivo2 arquivo3 .. arquivo n >

É o indicador de relacionamentos entre os arquivos citados, previamente abertos, que serão manipulados durante as seções seguintes. Estes arquivos relacionados devem ser colocados na ordem de relacionamento: do que está no nível mais baixo (N) para o que está no mais alto (1); sendo que o "arquivo1" será o que sofrerá algum tipo de manipulação, enquanto que os à sua direita (arquivo2 .. arquivo n) servirão para consultas, podendo também ser manipulados.

Exemplo:

```
/*                                  // indica fim das imagens
OPEN arq1                           // abrindo arquivos
OPEN arq2
OPEN arq3
ENTER arq3 arq1 arq2
   .
   .
RETURN
etc ...
```

Seção ENTRY:

Sintaxe:

ENTRY arquivo.campo {opcao1,opcao2,...,opcao n}

Esta seção é a responsável pela ligação do banco de dados com o usuário. Dentro desta seção ocorre o entrelaçamento do que está nas janelas de uma imagem com os arquivos do banco de dados. No caso de haver um procedimento de procura com sucesso, esta seção moverá, para todas as janelas refenciadas com um comando ENTRY, todos os dados daquele arquivo e campos selecionados, automaticamente, isto é, no caso de ser encontrado um registro no arquivo, todos os comandos ENTRY serão preenchidos com os dados do arquivo. Como

a seção ENTRY é mais uma rotina da Macro Enter, deve ser indicado o seu fim com um comando RETURN. Para a entrada de dados, a seção Entry fornece diversas opções de formatação, de procura e de checagens dos dados das janelas:

- Autofind: opção de procura automática no índice principal. Caso a procura seja por um índice de mais de um segmento, a opção Autofind deverá estar no último campo do índice, isto é, na última opção de linha da entrada de dados dos segmentos do índice, para termos certeza de que todos os dados necessários para a procura já se encontram disponíveis.
- Autofind_ge: opção idêntica à anterior, só que irá procurar um registro diferente do contido na janela, procurando por um registro maior ou igual ao constante na janela.
- Skipfound: é uma opção que aparece, normalmente, junto a um Autofind ou a um Autofind_ge. Sua função é, caso seja achado um registro, mostrar os dados da janela e pular para a próxima, sem deixar que ela possa ser modificada.
- Findreq: mais uma opção de procura, só que esta exige que um registro seja encontrado, isto é, exige que os dados da janela estejam obrigatoriamente no arquivo de dados, permanecendo nesta janela até que seja achado o registro ou seja usada alguma Flex-Key.
- Capslock: opção de formatação de janela ASCII, que não permite a entrada de dados em letras minúsculas, transformando-as imediatamente em letras maiúsculas.
- Range = n1,n2: opção de checagem de janela numérica. Verifica se o conteúdo da janela é maior ou igual a n1 e menor ou igual a n2, isto é, se o conteúdo está entre n1 e n2, inclusive. Esta opção é válida também para datas (Range = dat1,dat2).
- Check = "str1%str2*str3#str4!...|str n": tem a mesma função do Range = só que verifica se o conteúdo de uma janela ASCII está contida na string do comando (entre aspas ""). Esta string do comando é formada por diversas outras strings. Usa-se separar estas strings, dentro da

opção, com caracteres difíceis de serem digitados, isto porque a opção Check = verifica se o conteúdo da janela se encontra dentro da string da opção. Ex: se sua janela possui três caracteres, a opção Check = irá procurar aquele conjunto de caracteres dentro da string da opção, três a três caracteres (o tamanho da janela); se não achá-la, ocorrerá um erro, caso contrário, se não for achada passará para o próximo comando ENTRY, se houver.

- Required: opção que normalmente acompanha um Autofind ou um Autofind_ge, podendo aparecer sozinho. Sua função é só passar para a próxima janela caso a janela, a que a opção está ligada, for preenchida, parcialmente ou não.
- Retain: mantém os dados na janela. Quando for pressionada a Flex-Key limpa (F9) uma vez, todas as janelas serão limpas, menos as que possuírem a opção que possuir Retain; estas poderão ser limpas teclando-se a Flex-Key limpa (F9) duas vezes.
- Retainall: mesma função da opção acima, sendo que os dados das janelas jamais serão apagados, mesmo teclando F9 diversas vezes.
- Forceput: forçar a entrada dos dados das janelas para o buffer de registro sem verificar se alguma modificação foi realizada.
- Noput: impede que os dados das janelas sejam movidos para o buffer.
- Noenter: impede a entrada de dados naquela janela, pois esta janela será preenchida e pulada, indepen-dentemente de se ter encontrado um registro com Autofind ou Autofind_ge (diferindo do Skipfound que avançará o cursor para a próxima janela somente se for encontrado o registro).
- Displayonly: é uma combinação do Noput e Noenter, pois impede os dados das janelas de ir para o buffer e impede a entrada de dados naquela janela, sendo preenchida e pulada.

Exemplo:

```
/*
PAGE tela                    // imprime a tela
OPEN CADTEL                  // deixa o arquivo CADTERL pronto para uso
ENTER CADTEL
AUTOPAGE tela                // enderecamento automatico das janelas
 ENTRY CADTEL.NOME {AUTOFIND,CAPSLOCK} // opcao de
 ENTRY CADTEL.ENDERECO {CAPSLOCK}// procura no indice 1
 ENTRY CADTEL.NUMERO         // e entrada so em letras
 ENTRY CADTEL.APTO           // maiusculas
 ENTRY CADTEL.CIDADE         {CAPSLOCK}
 ENTRY CADTEL.ESTADO         {CAPSLOCK}
 ENTRY CADTEL.CEP
 ENTRY CADTEL.BAIRRO         {CAPSLOCK}
 ENTRY CADTEL.DDD
 ENTRY CADTEL.FONE
 ENTRY CADTEL.DATA_NASCIMENTO {AUTOFIND} // procura pelo
                                         indice 2
 RETURN                      // marca fim da rotina da entrada de dados
ENTEREND                     // delimita o fim da macro
                             // neste ponto devem ser inseridas as outras
                             // rotinas do programa
ABORT                        // fim
```

A seção ENTRY apresenta algumas peculiaridades:

- no caso de uma operação de procura de um registro com sucesso, os campos do arquivo que tiverem um ENTRY relacionado serão exibidos na respectiva janela associada ao comando;

- como o processamento na seção ENTRY é de cima para baixo, todos os comandos ENTRY serão executados seqüencialmente, juntamente com todos os comandos que se encontrarem entre os ENTRY, caso um comando Autopage esteja antecedendo a série de comandos ENTRY;

- caso ocorra uma procura, com sucesso, os comandos internos aos ENTRY não serão executados, pois esta seção preencherá somente os comandos ENTRY relacionados com os campos do arquivo;

- certas Flex-Keys, quando usadas, poderão não processar os comandos internos aos ENTRY, tais como Flex-Key back-field (F2), find (TAB), super-find (F3), onde somente os ENTRY serão executados;

- comandos relacionados a operações com buffer devem ser evitados, pois os dados ainda não foram movidos para lá, podendo ocorrer erros ou geração de registros em branco; estes comandos devem ser colocados na seção POST-ENTRY, onde terão local apropriado.

Dentro da seção ENTRY podemos ainda usar outros dois comandos paralelamente ao ENTRY, mas com cuidado redobrado devido às peculiaridades da seção:

- ACCEPT: comando para entrada de dados em uma janela.

Sintaxe:

ACCEPT < janela > < TO var_destino < {opcoes} > >

Funciona basicamente como um ENTRY mas esta entrada de dados será para uma variável de memória, para um janela em específico ou para a janela que vier na seqüência, isto é, entre comandos ENTRY. O comando ACCEPT não possui todas as opções de que dispõe o comando ENTRY, mas os comandos de formatação de janelas são válidos (POINTS,RANGE =, CHECK =).

Exemplo:

```
ACCEPT {check="SN", capslock}
ACCEPT tela.4
ACCEPT tela.3 TO nome_do_mes {capslock}

ACCEPT TO valor_pagamento {range=10000,99000}
```

- DISPLAY : move os dados de um campo, variável, constante para uma janela específica ou para a janela corrente.

Sintaxe:

DISPLAY arquivo.campo < TO janela < {opcoes} > >

O comando display possui opções de formatação de janelas diferentes do ACCEPT e do ENTRY (SIGNRT, FILL, FLOAT$, etc).

Exemplo:

DISPLAY cadtel.data_nascimento TO tela.2

DISPLAY duplicatas.valor_pagamento { float$ }

Seção Post_entry:

É uma seção formada por 4 (quatro) sub-rotinas, responsáveis pelas operações com os arquivos de dados. Estas sub-rotinas deverão estar após o comando RETURN, da seção Entry, e antes do Enterend que delimita o fim da macro.

Para que as operações com os arquivos de dados sejam realizadas, fazem-se necessários alguns pré-processamentos com os conteúdos das janelas de entradas de dados. Para cada uma das operações básicas com arquivos de dados usa-se uma seqüência de uma a duas sub-rotinas que compõem a seção Post-Entry. Como na seção Entry estas sub-rotinas são delimitadas pelo comando RETURN que indicará seu fim.

A seção Post-Entry é formada pelas sub-rotinas:

- ENTER.EDIT
- ENTER.SAVE
- ENTER.DELETE
- ENTER.CLEAR
- ENTER.EXIT

Dentro de cada uma destas sub-rotinas podem ser usados quais-quer comandos Dataflex válidos, com exceção daqueles que venham de encontro às funções básicas de cada sub-rotina (não se deve salvar dados durante uma operação de exclusão). Mas pode-se, através destas, bloquear as operações de manipulação, bastando que os comandos internos à sub-rotina não permitam a operação que está sendo realizada.

Estas sub-rotinas devem ser declaradas da seguinte maneira:

```
ENTER.nome_sub-rotina :
      comando1
      comando2
      ...
      comando n
RETURN
```

Para que sejam entendidas as funções básicas de cada uma das sub-rotinas, torna-se necessário saber como elas agem de acordo com o tipo de operação que está sendo realizada:

- Inclusão: no ato da inclusão de um registro a Macro Enter executará a sub-rotina: ENTER.SAVE.

Ela irá limpar o buffer de registro do arquivo de dados e em seguida moverá o conteúdo das janelas para o buffer, gravando-o em seguida.

- Alteração: ao se alterar um registro a Macro Enter executará as seguintes sub-rotinas: ENTER.EDIT e ENTER.SAVE. A operação de alteração deve ser precedida de uma operação de procura, para colocar o registro a ser modificado no buffer de registro do arquivo de dados. A execução da Enter.Edit servirá para um pré-processamento, antes dos dados das janelas serem movidos para o buffer para serem gravados. Após isto executa-se a Enter.Save para gravar os dados alterados das janelas para o arquivo de dados.

- Exclusão: para se deletar um registro é necessário que ele esteja disponível no buffer de registro, isto é, após uma operação de procura realizada com sucesso. A operação de exclusão é realizada com a teclagem da Flex-Key Delete (F6). A Macro Enter executará então as sub-rotinas ENTER.DELETE e ENTER.EDIT na ordem. Haverá a exclusão do registro residente no buffer de registro juntamente com a atualização do arquivo de índices. O fluxo do programa passará então para a sub-rotina Enter. Delete que fará um pós-processamento nos dados.

As outras duas sub-rotinas têm um papel diferente das demais:

- ENTER.CLEAR: sua função é, basicamente, limpar as janelas da imagem juntamente com os buffers de registro. Esta sub-rotina é executada, sempre, depois de operações de inclusão, alteração e deleção. Ela pode ser executada em qualquer parte da execução do programa, bastando apenas teclar a Flex-Key CLear Windows (F9). Caso algum comando seja inserido dentro desta sub-rotina, ela os executará normalmente, não importando que estes comandos façam-na perder suas características.

- ENTER.EXIT: sua função é sair do programa em execução executando um desvio para o fim do programa, onde normalmente se encontra um comando ABORT.

Exemplo:

Supondo um programa para cadastro de pessoas e telefones:

```
/tela
      AGENDA      TELEFONICA
NOME : <                                  >
ENDERECO :                    No :    . APTO :    .
CIDADE :                    ESTADO :   CEP :
BAIRRO :

TELEFONE : (   )                DATA DE NASCIMENTO : <  /  /  >

      Numero de registros cadastrados :

      Use os campos com < > para procura de dados.
/*                          // indica fim da imagem
PAGE tela                   // imprime a tela
OPEN CADTEL                 // deixa o arquivo CADTEL pronto para uso
OPEN NUMREG // arquivo onde esta o numero de registro do arquivo
CADTEL
ENTER CADTEL                // inicializa a Macro Enter
AUTOPAGE tela               // enderecamento automatico das janelas
 ENTRY CADTEL.NOME {AUTOFIND,CAPSLOCK}// opcao de procura
no
 ENTRY CADTEL.ENDERECO {CAPSLOCK}      // indice 1 e entrada
 ENTRY CADTEL.NUMERO                   // so em letras
 ENTRY CADTEL.APTO                     // maiusculas
 ENTRY CADTEL.CIDADE {CAPSLOCK}
 ENTRY CADTEL.ESTADO{CAPSLOCK}
 ENTRY CADTEL.CEP
 ENTRY CADTEL.BAIRRO {CAPSLOCK}
 ENTRY CADTEL.DDD
 ENTRY CADTEL.FONE
 ENTRY CADTEL.DATA_NASCIMENTO {AUTOFIND}// procura pelo
                                        // indice 2
```

```
ENTRY NUMREG.COUNTER {DISPLAYONLY} // mostra campo
 RETURN                            // delimita fim da secao ENTRY
ENTER.SAVE: // sub-rotina para inclusao ou alteracao de registros
     calc (NUMREG.COUNTER + 1 ) to NUMREG.COUNTER
                                   // soma 1 no contador
                                   // de registros
SAVE NUMREG                        // Salva o arquivo auxiliar
RETURN                             // delimita fim da sub-rotina

ENTER.EDIT:                        // sub-rotina executada em
                                   //alteracoes e/ou delecoes
     calc ( NUMREG.COUNTER - 1 ) to NUMREG.COUNTER
                                   // subtrai um do
                                   // contador de registros
     SAVE NUMREG                   // Atualiza arquivo auxiliar
     RETURN                        // delimita fim da sub-rotina
ENTEREND                           // delimita fim da Macro Enter
ABORT                              // fim
```

CAPÍTULO 8
MACRO ENTERGROUP

8.1 O que é Macro Entergroup

É uma pequena MACRO ENTER, realiza as mesmas operações, tem as mesmas características, porém podemos manipular mais de um registro por arquivo, de cada vez.

Ela não tem tantas facilidades como o controle das operações de manipulação de arquivos, mas oferece ao usuário outras possibilidades e opções de programação, exatamente por ser mais liberal e resumida.

Nesta Macro diversas Flex-Keys perdem suas funções, sendo possível redefini-las (F3 SuperFind, F6 Delete, F9 Clear Windows, F10 Save, ESC Escape). A redefinição destas Flex-Keys pode ser feita usando-se os comandos disponíveis do Dataflex. As demais Flex-Keys mantêm suas funções normais (F1 Help, F2 Back-Field, F4 Print Screen, F5 Calculate, F7 User, F8 User2, TAB Find, Pg Up Next Rec, Pg Dn Previous Rec, Enter), não querendo dizer que estas também não possam ser redefinidas pelo usuário.

A principal diferença entre as duas macros é que a Macro ENTER só pode existir uma única vez na aplicação, enquanto a Macro ENTERGROUP pode existir quantas vezes forem necessárias. Todas as Macro ENTERGROUP devem, obrigatoriamente, terminar antes que uma outra Macro tenha início.

8.2 O que faz a Macro Entergroup

Devido à sua semelhança com a Macro ENTER, a Macro Entergroup tem as mesmas funções básicas da anterior. Funciona como base para entrada e consulta de dados.

A Macro ENTERGROUP é delimitada pelos comandos ENTERGROUP e ENDGROUP. Dentro deste bloco podem ser usados quaisquer comandos usados em Dataflex, principalmente os comandos ENTRY, responsáveis pelo acesso aos dados dos arquivos de dados, exatamente como na Macro ENTER.

Sua estrutura é:

```
ENTERGROUP
    ENTRY arquivo.campo1 { opcoes }
    ENTRY arquivo.campo2 { opcoes }
    comando1
    comando2
    ENTRY arquivo.campo3 { opcoes }
    .
    .
    ENTRY arquivo.campo n { opcoes }
    comando n
ENDGROUP
```

Como você pode verificar, o conjunto de comandos ENTRY não termina com um comando RETURN como a Macro ENTER, pois este bloco de comandos (interno aos comandos ENTERGROUP e ENDGROUP) só será executado uma vez, a não ser que algum comando desvie o fluxo do programa para o início da Macro.

Outra diferença a ser notada é que esta Macro não apresenta comandos que indiquem os relacionamentos entre arquivos, portanto torna-se necessário que se usem comandos para fazer trazer os dados de arquivos secundários para o principal ou comandos que levem os dados do arquivo principal aos secundários. Como a Macro ENTERGROUP não apresenta esta peculiaridade, indicar o relacionamento de arquivos, os processos de procura, inclusão, alteração e exclusão de dados devem ser feitos separadamente. Todos estes processos básicos devem ser construídos instrução a instrução.

Com a falta de algumas Flex-Keys, o processo de criação de uma aplicação torna-se mais trabalhoso, porém mais gratificante.

Para que possamos trabalhar adequadamente com esta Macro, devemos aprender, primeiramente, como redefinir algumas das Flex-Keys que perderam a função.

8.2.1 Redefinindo as Flex-Keys

Antes de iniciar este tópico devemos esclarecer alguns pontos:

- As Flex-Keys são basicamente rotinas executadas assim que é pressionada determinada tecla. Sua localização é ao final do programa, após todas as rotinas e sub-rotinas.

- Sendo rotinas devem ter um início e um fim, podendo, seus comandos, desviarem o fluxo do programa para outras rotinas ou sub-rotinas.

- Todas as Flex-Keys têm um nome próprio para o Dataflex, portanto qualquer nome, além destes usados por ele, ocasionará um erro durante a compilação.

- Os nomes usados pelo Dataflex são os seguintes:

KEY.HELP	-F1	F2 - KEY.FIELD	| TAB	KEY.FIND
KEY.SFIND	-F3	F4 - KEY.PRINT	| Pg Dn	KEY.NEXT
	-F5	F6 - KEY.DELETE	| Pg Up	KEY.PREVIOUS
KEY.USER	-F7	F8 - KEY.USER2	| ESC	KEY.ESCAPE
KEY.CLEAR	-F9	F10 - KEY.SAVE	|	

(F5)

A Flex-Key CALC não pode ser redefinida.

Como as Flex-Keys são rotinas, sua delimitação obedece as mesmas regras, somente com uma diferença, deve-se usar o comando KEYPROC, seguido do nome da Flex-Key mais dois pontos ":", para indicar que se trata de uma redefinição de Flex-Key. Ao final do bloco de comandos, deve-se usar o comando RETURN, delimitando seu fim e indicando que o processamento deve retornar ao ponto de onde foi desviado.

A sintaxe para que uma Flex-Key possa ser redefinida é:

```
KEYPROC KEY.nome da Flex-Key:
        comando1
        comando2
        ...
        comando n
    RETURN
```

Exemplo:

```
KEYPROC KEY.ESCAPE:
    SHOW " DESEJA SAIR DO PROGRAMA S/N ? "
    ACCEPT TELA.12 TO RESPOSTA {CHECK="SN", CAPSLOCK}
    IF RESPOSTA EQ "S" ABORT
  RETURN

KEYPROC KEY.DELETE
    SHOW "ESTE PROGRAMA NAO PERMITE EXCLUSAO DE DADOS"
    ENTAGAIN
  RETURN
```

Nestes dois exemplos foi usado o comando SHOW. Este comando serve para a impressão de mensagens, variáveis e campos de arquivos na posição do cursor; as mensagens devem vir entre aspas ("") com no máximo 250 caracteres. Se a mensagem não couber na tela de seu monitor de vídeo, divida-a em duas, utilizando dois comandos SHOW. Este comando imprime a mensagem e mantém o cursor na mesma linha de impressão. Existe uma forma variante do comando SHOW e SHOWLN que tem as mesmas características porém, ao final da impressão, passa para a próxima linha (imprime um line-feed).

O outro comando, ENTAGAIN, evita em Keyprocs que o cursor avance para a próxima janela da imagem, permanecendo na janela onde foi teclada a Flex-Key.

Vejamos a seguir, um exemplo de um programa usando Macros ENTERGROUP e redefinições de Flex-Keys.

Exemplo:

/TELA resident Programa:LOCACAO.FRM

LOCACAO DE FITAS Data : __/__/__

LOCACAO : ____. Data da locacao : __/__/__

Codigo : _____. Cliente : _____

Numero de fitas locadas : _ (1-5) Responsavel : _____

FITA : _ Codigo: _____. Titulo: _____

Valor : _____.__ Desconto : _____.__

Total: _____.__ _____.__

TOTAL A PAGAR: _____.__

[HELP Disponivel]

/help resident

TELA DE AJUDA AO CADASTRO DE LOCACAO

Esta tela pode ser usada somente para operacoes de inclusao de locacoes de fitas de video da Locadora. Onde :

Locacao : numero atribuido a locacao.
Data da Locacao : data em que foi realizada a locacao, usa-se a data do sistema neste caso.
Codigo do Cliente : codigo atribuido ao cliente no ato de seu cadastramento.
Responsavel : funcionario que realizou a locacao.

Numero de fitas locadas : numero total de fitas levadas pelo cliente, maximo de cinco fitas.
Codigo, Titulo, Valor e Desconto: dados a respeito das fitas.
Total e TOTAL : valores totalizados de valor do aluguel da fita e valor de desconto da mesma e o total a pagar: (valor a pagar - desconto)

< qualquer tecla volta ao programa ... >

/HEADER resident

VIDEO LOCADORA - HIPOTETICA

LOCACAO DE FITAS Data : __/__/__

LOCACAO : ____. Data da locacao : __/__/__

Codigo : _____. Cliente : _____

Numero de fitas locadas : _

Fita Codigo Titulo Valor Desconto

/BODY resident

_ _____. _____ ___,___,___.__ ___,___.__

/TOTAL resident

Total: ___,___,___.___,___.__

TOTAL A PAGAR: ___,___,___.__

Responsavel : _____

Cliente : _____ Ass: _____

A Pagar

```
/*
// abrindo arquivos

   OPEN locacao
   OPEN ficha
   OPEN cadfitas
   OPEN cadcli

// criando variaveis
   INTEGER num_locacao counter num_fitas
   NUMBER  valor desconto tot
   STRING tecla 1
   DATE data_sys
   SYSDATE data_sys
   MOVE data_sys TO tela.1

// inicio do programa

Comeco:
      CLEAR locacao
      CLEAR ficha
      CLEAR cadfitas
      CLEAR cadcli
      clearform tela.2 thru tela.15
      clearxy 24 0

      MOVE 0          TO valor
      MOVE 0          TO desconto
      MOVE 0          TO tot
      MOVE 0          TO num_locacao
```

```
Inicio:

    MOVE data_sys TO tela.3
    ENTERGROUP
        AUTOPAGE tela 2
        ENTRY locacao.numero          {autofind,required}
        ENTRY locacao.data            {displayonly}
        ENTRY cadcli.codigo           {autofind,findreq}
        ENTRY locacao.nome_cliente    {capslock}
        ENTRY locacao.num_fitas_loc   {points=0,range=1,5}
        ENTRY locacao.responsavel     {capslock}
    ENDGROUP

    INDICATE achei STATUS locacao
    GOSUB Verifica // verifica algumas condicoes basicas para a locacao
    GOSUB Confirma       // confirma locacao
    GOSUB Grava_locacao // efetua locacao
    GOSUB Monta_ficha // looping para o cadastramento das fitas locadas

    gotoxy 24 0
    show " Uma tecla para continuar !"
    inkey tecla
    clearxy 24 0

GOSUB imprime_locacao
ABORT
```

```
//////////////////////////// sub-rotinas ////////////////////////////


Verifica:
    [achei] begin
        clearxy 24 0
        gotoxy 24 0
        show " Esta locacao ja existe !!!."
        show " Use outro numero para uma nova locacao !"
        inkey tecla
        clearxy 24 0
        ABORT
      end
    [not achei] findkey eq cadcli by index.1 for cadcli.codigo eq tela.4
                                [not found] begin
                                clearxy 24 0
                                gotoxy 24 0
                                show " Este cliente nao esta cadastrado !!!."
                                show " Entre um codigo valido !."
                                inkey tecla
                                clearxy 24 0
                                return Inicio
                            end
    RETURN


Grava_locacao:
    MOVE locacao.numero         TO num_locacao
    MOVE tela.6                 TO num_fitas
    MOVE "N"                    TO locacao.BAIXA
    MOVE cadcli.codigo          TO locacao.codigo_cliente
    MOVE data_sys               TO locacao.data
    SAVE locacao
    RETURN
```

```
Monta_ficha:

 MOVE 1 TO counter
REPEAT
     CLEAR ficha
      CLEARFORM TELA.8 THRU TELA.12
      ENTERGROUP
          AUTOPAGE      TELA 8
          DISPLAY COUNTER
          ENTRY cadfitas.codigo  TELA.9 {AUTOFIND,FINDREQ}
          ENTRY cadfitas.titulo          TELA.10 {DISPLAYONLY}
          ENTRY ficha.valor_locacao      TELA.11
          ENTRY ficha.desconto           TELA.12
          CALC (tela.11 + valor)         TO valor
          MOVE valor                     TO tela.13
          CALC (tela.12 + desconto)      TO desconto
          MOVE desconto                  TO tela.14
          MOVE (valor - desconto )       TO tela.15
     ENDGROUP

     MOVE num_locacao                    TO ficha.locacao
     MOVE cadfitas.codigo                TO ficha.codigo_fita
     SAVE ficha
     INCREMENT counter
 UNTIL counter GT num_fitas
 MOVE tela.15 TO tot
 RETURN

Confirma:
     CLEARXY 24 0
     GOTOXY 24 0
     SHOW " Confirma locacao (S/N) ?"
     INKEY TECLA
     CLEARXY 24 0
     IF tecla IN "Nn" RETURN Inicio
     RETURN
```

```
Imprime_locacao:
     MOVE 1 TO counter
     OUTFILE
     REPORT FICHA BY INDEX.1
          INDICATE SELECT AS FICHA.LOCACAO EQ num_locacao
[NOT SELECT] RETURN END.OF.REPORT  //SAI FORA INDICE

SECTION HEADER
     PRINT data_sys
     PRINT LOCACAO.NUMERO
     PRINT LOCACAO.DATA
     PRINT LOCACAO.CODIGO_CLIENTE
     PRINT cadcli.nome
     PRINT LOCACAO.NUM_FITAS_LOC
     OUTPUT HEADER

SECTION BODY
     PRINT counter
     PRINT FICHA.CODIGO_FITA
     PRINT CADFITAS.TITULO
     PRINT FICHA.VALOR_LOCACAO
     PRINT FICHA.DESCONTO
     increment counter
     OUTPUT BODY

SECTION TOTAL
     SUBTOTAL BODY.4                    TO TOTAL.1
     SUBTOTAL BODY.5                    TO TOTAL.2
     PRINT tot                          TO TOTAL.3
     PRINT LOCACAO.NOME_CLIENTE         TO TOTAL.5
     PRINT LOCACAO.RESPONSAVEL          TO TOTAL.4
     OUTPUT TOTAL

     REPORTEND
     FORMFEED
 RETURN
```

```
////////////////////////// Redefinicao de Flex-Keys ////////////////////

KEYPROC key.clear
    CLEARFORM TELA.2 THRU TELA.15
    RETURN COMECO
    RETURN

KEYPROC key.escape
    CLEARXY 24 0
    GOTOXY 24 0
    SHOW "Deseja sair (S/N) ? "
    INKEY tecla
    clearxy 24 0
    IF tecla IN "Ss" abort
    ENTAGAIN
    RETURN

KEYPROC key.delete
    clearxy 24 0
    gotoxy 24 0
    show "TECLA DESATIVADA !!!"
    inkey tecla
    clearxy 24 0
    entagain
    RETURN
```

CAPÍTULO 9
UTILITÁRIOS DATAFLEX

Este capítulo tem por objetivo dar uma visão geral das possibilidades de alguns módulos do Sistema de Gerenciamento de Bancos de Dados e do Sistema Gerador de Aplicações do Dataflex.

Como já foi dito Dataflex é formado por diversos módulos, cada um com uma função bem específica. Veremos como podem ser aproveitados cada um deles e quais vantagens estes módulos, juntos ou separados, podem fornecer ao usuário Dataflex.

Iniciaremos com o utilitário FILEDEF, como lembra o nome Definidor de arquivos. Veremos também o utilitário REINDEX, responsável pela organização dos arquivos de índices, quando estes já não cumprem, satisfatoriamente, suas funções. Usaremos ENTERDEF e MENUDEF para gerar respectivamente programas de entrada de dados a partir de uma definição de arquivo gerada pelo FILEDEF e como criar menus, fácil e seguramente, apenas digitando suas opções.

9.1 FILEDEF

FILEDEF possibilita a você criar definições de arquivos, através da definição de suas estruturas e relacionamentos, que serão usados nas aplicações Dataflex.

A definição de dados, uma vez criada, contém informações a respeito do nome do arquivo, dos índices e dos campos. Através do módulo FILEDEF podemos criar aplicações, partindo das definições dos arquivos de dados ou ainda, usando um método parecido com o do AUTODEF, criar aplicações, a partir de uma imagem.

Para executarmos o módulo FILEDEF devemos estar no Menu Principal e escolher a opção: Definir um Banco de Dados (FILEDEF). Ou então digitar FILEDEF, no Sistema Operacional.

Irá aparecer uma tela parecida com esta:

```
        DATAFLEX 2.2 FILE DEFINITION PROGRAM
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

    ARQUIVO
    NUMERO      NOME
    ----------  --------

    2   CADTEL
    49  @DataFlex Menu File
    50  @DataFlex error file
DIGITE NUMERO DO ARQUIVO OU <ENTER> P/ MAIS:
```

Escolha o número de arquivo desejado. Este número de arquivo não deverá ser nem o 49 nem o 50, pois são arquivos usados pelo Dataflex, portanto qualquer alteração na sua estrutura poderá acarretar sérios prejuízos à sua execução e integridade.

Quando usamos o AUTODEF criamos uma aplicação com o nome CADTEL juntamente com o arquivo CADTEL. Durante o AUTODEF foi-nos pedido que entrássemos os nomes dos campos para a formação do Dicionário de Dados. Parte deste Dicionário nos é mostrado agora na forma Número e Nome do Arquivo de Dados.

Se quiser tecle o número 2, caso contrário digite um número qualquer para a criação de outro arquivo de dados.

Vejamos, primeiramente, o uso do FILEDEF em um arquivo já conhecido. Após aprendermos como usá-lo, poderemos criar a estrutura de um outro arquivo juntamente com outra aplicação.

A seguinte tela irá aparecer:

```
        DATAFLEX 2.2 FILE DEFINITION PROGRAM
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

    ARQUIVO.....: CADTEL (2)

        1: CRIA/ALTERA ESPECIFICACAO CAMPOS
        2: CRIA/ALTERA INDICES
        3: MOSTRA ESTRUTURA DO ARQUIVO
        4: ALTERA PARAM/NOMES DE ARQUIVO
        5: EXCLUI ARQUIVO
        6: POE ARQ INATIVO
        7: CRIA ESTRUTURA DE ARQUIVO USANDO IMAGENS
        8: CRIA ESTRUTURA ARQUIVO USANDO .DEF
        9: GRAVA ESTRUTURA E SAI DO FILEDEF
        10: SAI DO FILEDEF, SEM ATUALIZAR ESTRUTURA


        QUAL A SUA OPCAO.... (1 A 10):
```

Neste ponto temos a nossa disposição uma série de pequenos utilitários que podem ser usados quase separadamente. Iniciaremos com o mais comum, em termos de uso:

Item 3: MOSTRA ESTRUTURA DO ARQUIVO

A seguinte mensagem irá aparecer:

Envia saida p/ <S>creen, <P>rinter, ou <F>ile S

Cada uma destas opções é auto-explicativa, mas de vital importância para os passos futuros. A resposta comum a esta pergunta é "S", saída para o monitor de vídeo. As demais opções enviam a definição do arquivo, respectivamente, para a impressora e para um arquivo extensão ".DEF", que será usado futuramente para a definição e criação de uma aplicação através do utilitário ENTERDEF.

Em seguida esta ou uma seqüência destas aparecerá:

```
        FILE DEFINITION - ESTRUTURA ARQ. #2
************************************************************
        NOME BASE DO ARQUIVO = CADTEL
        NOME PARA USUARIO = CADTEL
        NOME PARA DATAFLEX = CADTEL
************************************************************
        TAMANHO REGST = 170 (USADO = 129)
        NUMERO MAXIMO DE REGS = 100 (USADO = 0)
        ESPACO EXCLUIDO E REUTL
        MULTI-USER RE-READ ATIVO
************************************************************
```

FIELD NMBR	FIELD OFFSET	FIELD LEN	FIELD TSPE	DEC PTS	MAIN INDEX	RELATES--TO FILE	FIELD	
1	1	30	ASCII		3	0	0	NOME
2	31	30	ASCII		0	0	0	ENDERECO
3	61	3	NUMERIC	0	0	0	0	NUMERO
4	64	3	NUMERIC	0	0	0	0	APTO
5	67	20	ASCII		0	0	0	CIDADE
6	87	2	ASCII		0	0	0	ESTADO
7	89	5	ASCII		0	0	0	CEP
8	94	20	ASCII		0	0	0	BAIRRO
9	114	4	ASCII		0	0	0	DDD
10	118	9	ASCII		0	0	0	FONE
11	127	3	DATE		2	0	0	DATA_NASCIMENTO

INDEX 1: FIELD SEGMENTS: <1> <0>
INDEX 2: FIELD SEGMENTS: <11> <0>

Esta tela acima indica como é a estrutura do arquivo de dados CADTEL. Constam desta tela informações a respeito dos nomes atribuídos aos arquivos CADTEL, como também informações a respeito dos campos, tipo, tamanho, numeração e seu nome.

Cada uma das colunas da definição de arquivos dá uma informação específica e concisa quanto aos dados do arquivo.

FIELD NMBR : numeração do campo atribuída a ele durante a definição do arquivo.

FIELD OFFSET : posição do campo, em número de bytes, com relação ao início do registro.

FIELD LEN : tamanho do campo do arquivo em bytes (campos numéricos têm seu tamanho reduzido à metade; o cálculo para saber o tamanho de campos numéricos, com número ímpar de caracteres, é tomar a parte inteira de: [tamanho do campo / 2] + 0.5). Para campos data o tamanho é sempre de três bytes.

FIELD TSPE : tipo do campo : ASCII, numérico, data e overlap.

DEC PTS : número de casas decimais de um campo numérico. É zero para campos ASCII, datas e overlap.

MAIN INDEX : coluna onde são colocados os números dos índices criados durante a definição do arquivo de dados. Pode-se criar apenas 9 índices por arquivo de dados.

RELATES FILE : número de arquivo a que este está relacionado.

TO FIELD : campo de relacionamento entre os dois arquivos segundo o número de arquivo na coluna anterior.

Os outros dados fornecidos pelo FILEDEF quanto à estrutura do arquivo de dados são a respeito dos índices criados e os campos que o compõe.

Item 1: CRIA/ALTERA ESPECIFICACAO CAMPOS

Responsável pela criação ou alteração dos campos dos arquivos de dados.

A seguinte tela aparecerá:

```
CAMPO          :1 <NOME>
OFFSET         :1
TAMANHO        :30 BYTES
TIPO           :ASCII
INDICE ....    :1
RELACIONA
   ARQUIVO     : 0
     CAMPO     : 0
```

OPCOES :
G - Vai p/ campo especifico
I - Insere antes deste campo
D - Exclui este campo
A - Inclui campo no final
C - Altera este campo
R - Reajusta deslocamento do campo
+ - Mostra campo posterior
- - Mostra campo anterior
E - Volta ao Menu do FILEDEF
Sua Opcao...:

Para que haja um bom entendimento não só desta como de outras partes do FILEDEF, precisamos aqui de alguns esclarecimentos:

- Número e Nome do campo : o número é dado automaticamente ao campo no ato de sua criação. O nome deve ter no máximo 15 caracteres, sendo válidas todas as letras de A a Z, os números de 0 a 9 e os caracteres @, #, e o underline "_" . O nome do campo aqui declarado será acessado por outros módulos do Dataflex e será atualizado no Dicionário de Dados.

- Offset : posição em bytes do campo em relação ao início do registro.

- Tamanho : o tamanho em bytes, do campo.

- Tipo : o tipo do campo pode ser: ASCII (formado por letras, números, enfim, todos os caracteres disponíveis no *set* de caracteres do computador, com exceção de caracteres especiais; cada caractere ocupa um byte no meio de armazenamento); Numeric (números inteiros ou reais; números reais devem ser declarados com o número de casas decimais após a vírgula); Date (campos numéricos com tamanho fixo de 3 bytes, que são guardados na forma do calendário juliano); Overlap (são campos que englobam diversos outros, são usados normalmente para facilitar a criação de índices multi-segmento; o campo Overlap engloba os demais campos,

guardando suas informações, o que facilita nos processos de acesso às informações contidas no registro).

- Índice : mostra o número do índice a que o campo está ligado.

- Arquivo : mostra o número do arquivo a que o campo está relacionado.

- Campo : mostra o campo do arquivo relacionado a que este campo está ligado.

Seguem-se as telas bases para entrada de dados a respeito de novos campos, explicações a respeito do menu de opções fornecido neste item do FILEDEF.

Campos ASCII:

```
************ CAMPO = XX ************

NOME CAMPO: TESTE
TIPO CAMPO: ASCII
NUMERO DE CARACTERES:30

RELACIONA COM ARQUIVO:  0
RELACIONA COM CAMPO :   0
```

Campos Numericos:

```
************ CAMPO = XX ************

NOME CAMPO: TESTE
TIPO CAMPO: NUMERIC
NUMERO DE CARACTERES A ESQUERDA DO PONTO DECIMAL.....: 2
NUMERO DE CARACTERES A DIREITA DO PONTO DECIMAL.....:5

RELACIONA COM ARQUIVO:  0
RELACIONA COM CAMPO :   0
```

Campos Data:

```
************* CAMPO = XX *************
NOME CAMPO: TESTE
TIPO CAMPO: DATE

RELACIONA COM ARQUIVO:  0
RELACIONA COM CAMPO :   0
```

Campos Overlap:

```
************* CAMPO = XX *************
NOME CAMPO: TESTE
TIPO CAMPO: OVERLAP
PROX OFFSET = nnn
OFFSET CAMPO: yyy
TAMANHO DO CAMPO OVERLAP: Z

RELACIONA COM ARQUIVO:  0
RELACIONA COM CAMPO :   0
```

Obs.:

1 Próximo Offset : é a posição do campo, com relação ao início do registro.

2 Offset Campo : é a posição, em bytes, de onde terá início o campo overlap.

3 Tamanho do Campo Overlap : é o número de bytes que serão abrangidos pelo campo Overlap.

Menu de opções de manipulação de campos do item 1:

G - Vai p/ campo específico : pede-se o número do campo para onde se deseja ir. Após escolher o número do campo, irá aparecer uma tela idêntica à que está acima, somente mudando os dados referentes ao campo.

I - Insere antes deste campo : insere um novo campo antes da posição do campo mostrado. Irá aparecer a tela-base para a entrada dos dados do campo.

D - Exclui este campo : exclusão do campo residente na tela-base de entrada de dados. Ao final será perguntado se desejamos realinhar os demais campos do arquivo de acordo com a nova estrutura.

A - Inclui campo no final : inclusão de campo ao final do último campo do arquivo de dados.

C - Altera este campo : permite que sejam feitas mudanças no campo do arquivo, isto é, alterações de nome, tipo, tamanho, relacionamento e índices. Para alterar os dados do campo basta digitar os dados novos ou alterados nas janelas destinadas a estes dados. Caso se queira manter as informações anteriores, basta teclar ENTER.

R - Reajusta deslocamento do campo : atualização das posições dos campos, com relação ao início do registro, quando um campo é incluído ou excluído. Este processamento pode ser deixado para o FILEDEF, pois ao final deste item ele perguntará se desejamos reajustar os campos do registro.

+ - Mostra campo posterior : mostra na tela-base os dados do próximo campo.

- - Mostra campo anterior : mostra na tela-base os dados do campo anterior.

E - Volta ao Menu do FILEDEF.

Item 2: CRIA/ALTERA INDICES.

```
DATAFLEX 2.2 FILE DEFINITION PROGRAM
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

                    DEFINICAO INDICE

      INDEX     CAMPOS                STATUS     TAMANHO      NIVEIS
      -----     ------------------    ---------  -----------  ---------
1         1     0                     ONLINE          33           2
2        11     0                     ONLINE           6           2
         3
         4
         5
         6
         7
         8
         9
        10

Add, Batch, Change, Delete, Online, Exit E
```

Este item do FILEDEF mostra-nos a estrutura, tipo e tamanho dos índices criados durante a fase de definição do arquivo de dados.

Podemos incluir, alterar e excluir qualquer índice. Na última linha da tela, são fornecidas todas as opções para manipulações de índices em Dataflex:

- Add : adicionar um índice ao final da relação. Será perguntado pelo número de segmentos seguido pelos respectivos segmentos. A escolha dos segmentos poderá ser feita através da digitação do número do campo ou do uso das teclas *Pg Dn* e *Pg Up*, para uma varredura seqüencial dos nomes de campo do arquivo.

- Batch : altera a característica do índice Online para Batch (índices que são atualizados apenas com o uso do utilitário REINDEX).

- Change : altera a definição de um índice específico. Deve-se dar o número do índice a ser alterado e, em seguida, dar o número de segmentos que este índice terá, indicando a alteração procedendo do mesmo modo que a adição de um índice.

- Delete : exclui um índice, bastando para isso a indicação de seu número.

- Online : muda a característica de índice Batch para Online (índices que são atualizados paralelamente às inclusões, alterações e exclusões dos registros do arquivo).

- Exit : saída deste item para o menu principal de opções do FILEDEF.

Item 4 : ALTERA PARAM/NOMES DE ARQUIVO

Permite a mudança dos nomes dos arquivos, o tamanho dos registros e o número máximo de registros dos arquivos de dados.

DATAFLEX 2.2 FILE DEFINITION PROGRAM

= = = =

PARAMETROS	
TAMANHO REGS	[170]
NUMERO MAX REGISTROS	[100]
ESPACO REUTILIZADO	[S]
MULTI-USER RE-READ	[S]

Altera informacao acima?

Agora, vejamos alguns esclarecimentos a respeito desses dados.

- NOME BASE : é o nome com que foi criado originalmente o arquivo de dados.
- NOME USUÁRIO : é o nome que será usado em outros módulos Dataflex; seu tamanho não deve exceder 30 caracteres.
- NOME DATAFLEX : é o nome usado internamente pelo Dataflex, isto é, o nome com que devem ser abertos e referidos dentro dos programas Dataflex.
- TAMANHO REGISTROS : é o tamanho máximo atingido pelos registros do arquivo de dados. Caso este valor seja alterado, Dataflex, independentemente do tamanho indicado, estará sempre tentando otimizar, reduzir os espaços inúteis, portanto, este número poderá variar.
- NÚMERO MÁXIMO DOS REGISTROS : é a estimativa do maior número de registros que poderão ser utilizados para armazenamento de dados. No caso da modificação da quantidade inicial do número máximo de registros, uma reindexação será obrigatória para manter a confiabilidade dos dados.
- ESPAÇO EXCLUÍDO REUTILIZADO : utilização do espaço de registros excluídos do arquivo de dados. Dataflex normalmente aloca seus novos registros em espaço ainda livre, mas com esta opção, Dataflex passará a utilizar os registros que antes pertenceram a

registros ativos. Os registros, uma vez excluídos, nunca mais poderão ser acessados.

- MULTI-USER RE-READ : caso o Dataflex esteja em ambiente multiusuário ele passará a fazer um REREAD (releitura) automático, atualizando sempre os dados do buffer de registro.

Item 5: EXCLUI ARQUIVO

Este item pode ser usado para todos os arquivos Dataflex quando se deseja excluir todos os registros do arquivos de dados de uma única vez. Existe um comando de programação Dataflex que tem o mesmo efeito: ZEROFILE seguido do nome de arquivo.

Item 6 : POE ARQUIVO INATIVO

Realiza a mesma função que o item anterior, porém também retira o nome do arquivo do Dicionário de Dados, isto é, da relação de nomes de arquivos. Se realmente foi a opção escolhida, confirma todas as perguntas que se seguem.

Item 7 : CRIA ESTRUTURA DE ARQUIVO USANDO IMAGENS

Este item é um pequeno AUTODEF, pois, a partir de uma imagem podemos criar uma definição de arquivo. Passando basicamente pelos

mesmos passos do AUTODEF, criamos uma definição de arquivo com a diferença que os passos individuais podem parar sempre que necessário.

Item 8 : CRIA ESTRUTURA ARQUIVO USANDO .DEF

Este item utiliza o arquivo gerado através do item 3 (arquivo que contém a estrutura do arquivo), com terminação ".DEF". Este item é de extrema importância quando desejamos transportar dados de um equipamento para outro, independente de sua compatibilidade em software ou hardware.

Item 9 : GRAVA ESTRUTURA E SAI DO FILEDEF

Talvez o item de maior importância, em termos do trabalho que tivemos até aqui. Este item atualiza as informações alteradas durante a execução deste módulo nos respectivos lugares (Dicionário de Dados, arquivos de dados e de índices), tornando-as efetivas.

Item 10 : SAI DO FILEDEF, SEM ATUALIZAR ESTRUTURA

Este item é usado normalmente quando executamos uma simples operação de consulta à estrutura do arquivo de dados selecionado, ou quando alteramos alguma informação relevante da estrutura ou da definição dos dados do arquivo. Este item não atualiza os dados alterados nos arquivos de dados nem dos índices e não atualiza o Dicionário de Dados.

Como foi dito no início da aprendizagem do FILEDEF poderíamoscriar uma aplicação através do FILEDEF. Podemos sugerir um sistema decontrole de locações de uma locadora de vídeo, constando de um cadastro de clientes, um cadastro de fitas, um cadastro de fornecedores e um cadastro das fichas de locação. Para isso precisamos responder à primeira pergunta do FILEDEF com um número diferente daqueles mostrados:

```
DIGITE NUMERO DO ARQUIVO OU <ENTER> P/ MAIS: 3
```

Aparecerá a seguinte mensagem:

```
ARQUIVO NOVO : S
```

Poderemos neste ponto definir o arquivo de cadastro de fitas: CADFITAS.

Seguida da tela idêntica à mostrada no item 4:

```
DATAFLEX 2.2 FILE DEFINITION PROGRAM
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
NOMES DATAFLEX
-------------          ------------------------------------
NOME BASE              [CADFITAS                ]
NOME USUARIO           [CADFITAS                ]
NOME DATAFLEX          [CADFITAS          ]

PARAMETROS
--------------------          ------------
TAMANHO REGS                  [512 ]
NUMERO MAX REGISTROS          [100     ]
ESPACO REUTILIZADO            [S]
MULTI-USER RE-READ            [S]
```

Em seguida teremos:

Selecionaremos o item 1, para a definição dos campos do arquivo.

```
CAMPO    :1    <CODIGO>          |  CAMPO    :2    <TITULO>
OFFSET   :1                      |  OFFSET   :4
TAMANHO  :3 BYTES                |  TAMANHO  :30 BYTES
TIPO     :NUMERICO, DECIM: 0     |  TIPO     :ASCII
INDICE .... :1                   |  INDICE .... :2
RELACIONA                        |  RELACIONA
   ARQUIVO : 0                   |     ARQUIVO    : 0
      CAMPO : 0                  |        CAMPO   : 0
                                 |
=================================|=================================
                                 |
CAMPO  :3  <FORNECEDOR>          |  CAMPO    :4 <@PRECO_PAGO>
OFFSET :34                       |  OFFSET   :36
TAMANHO :3 BYTES                 |  TAMANHO  :6 BYTES
TIPO     :NUMERICO, DECIM: 0     |  TIPO     :NUMERICO, DECIM: 2
INDICE .... :3                   |  INDICE .... :0
RELACIONA                        |  RELACIONA
   ARQUIVO : 4                   |     ARQUIVO    : 0
      CAMPO : 1                  |        CAMPO   : 0
                                 |
=================================|=================================
                                 |
CAMPO    :5    <TIPO>            |  CAMPO    :6 <DATA_COMPRA>
OFFSET   :43                     |  OFFSET   :34
TAMANHO  :1 BYTES                |  TAMANHO  :3 BYTES
TIPO     :NUMERICO, DECIM: 0     |  TIPO     :DATE
INDICE .... :4                   |  INDICE .... :0
RELACIONA                        |  RELACIONA
   ARQUIVO : 8                   |     ARQUIVO    : 0
      CAMPO : 1                  |        CAMPO   : 0
                                 |
=================================|=================================
```

Passaremos então à definição dos índices para o arquivo.

INDEX	CAMPOS	STATUS	TAMANHO	NIVEIS
1 *	1	ONLINE	3	2
2 *	2	ONLINE	30	2
3 *	3	ONLINE	3	2
4 *	5	ONLINE	1	1

Vejamos como ficou a estrutura do arquivo criado :

```
      FILE DEFINITION - ESTRUTURA ARQ. #3
**************************************************
      NOME BASE DO ARQUIVO  = CADFITAS
      NOME PARA USUARIO     = CADFITAS
      NOME PARA DATAFLEX    = CADFITAS
**************************************************
      TAMANHO REGST = 512 (USADO= 46)
      NUMERO MAXIMO DE REGS = 100 (USADO= 2)
      ESPACO EXCLUIDO E REUTL
      MULTI-USER RE-READ ATIVO
**************************************************
```

FIELD NMBR	FIELD OFFSET	FIELD LEN	FIELD TSPE	DEC PTS	MAIN INDEX	RELATES--TO FILE	FIELD	
1	1	3	NUMERIC	0	1	0	0	CODIGO
2	4	30	ASCII		2	0	0	TITULO
3	34	3	NUMERIC	0	3	4	1	FORNECEDOR
4	37	6	NUMERIC	2	0	0	0	@PRECO_PAGO
5	43	1	NUMERIC	0	4	8	1	TIPO
6	44	3	DATE		0	0	0	DATA_COMPRA

```
INDEX 1: FIELD SEGMENTS:  <1>
INDEX 2: FIELD SEGMENTS:  <2>
INDEX 3: FIELD SEGMENTS:  <3> <0>
INDEX 4: FIELD SEGMENTS:  <5> <0>
```

Proceda da mesma maneira para o cadastro de fornecedores e clientes. Segue a listagem da definição do arquivo FORNECEDORES, CADCLI,

LOCAÇÃO e FICHAS.

```
FILE DEFINITION - ESTRUTURA ARQ. #4
**************************************************
NOME BASE DO ARQUIVO  = FORNECEDORES
NOME PARA USUARIO     = FORNECEDORES
NOME PARA DATAFLEX    = FORNECEDOR
**************************************************
TAMANHO REGST = 512 (USADO = 169)
NUMERO MAXIMO DE REGS = 100 (USADO = 4)
ESPACO EXCLUIDO E REUTL
MULTI-USER RE-READ ATIVO
**************************************************
```

FIELD NMBR	FIELD OFFSET	FIELD LEN	FIELD TSPE	DEC PTS	MAIN INDEX	RELATES--TO FILE	FIELD	
1	1	3	NUMERIC	0	1	0	0	CODIGO
2	4	30	ASCII		2	0	0	NOME
3	34	13	ASCII		0	0	0	CGC
4	47	30	ASCII		0	0	0	ENDERECO
5	77	3	NUMERIC	0	0	0	0	NUMERO
6	80	10	ASCII		0	0	0	COMPLEMENTO
7	90	20	ASCII		0	0	0	CIDADE
8	110	20	ASCII		0	0	0	BAIRRO
9	130	2	ASCII		0	0	0	ESTADO
10	132	5	ASCII		0	0	0	CEP
11	137	4	ASCII		0	0	0	DDD
12	141	9	ASCII		0	0	0	FONE
13	150	20	ASCII		0	0	0	CONTATO

INDEX 1: FIELD SEGMENTS: <1>
INDEX 2: FIELD SEGMENTS: <2>

FILE DEFINITION - ESTRUTURA ARQ. #5
****
NOME BASE DO ARQUIVO = CADCLI
NOME PARA USUARIO = CADCLI
NOME PARA DATAFLEX = CADCLI
****
TAMANHO REGST = 512 (USADO = 196)
NUMERO MAXIMO DE REGS = 100 (USADO = 3)
ESPACO EXCLUIDO E REUTL
MULTI-USER RE-READ ATIVO
****

FIELD NMBR	FIELD OFFSET	FIELD LEN	FIELD TSPE	DEC PTS	MAIN INDEX	RELATES--TO FILE	FIELD	
1	1	3	NUMERIC	0	1	0	0	CODIGO
2	4	30	ASCII		2	0	0	NOME
3	34	30	ASCII		0	0	0	ENDERECO
4	64	3	NUMERIC	0	0	0	0	NUMERO
5	67	10	ASCII		0	0	0	COMPLEMENTO
6	77	20	ASCII		0	0	0	CIDADE
7	97	20	ASCII		0	0	0	BAIRRO
8	117	2	ASCII		0	0	0	ESTADO
9	119	5	ASCII		0	0	0	CEP
10	124	10	ASCII		0	0	0	RG
11	134	3	DATE		3	0	0	DATA_NASCIMENTO
12	137	20	ASCII		0	0	0	ASSOCIADO1
13	157	20	ASCII		0	0	0	ASSOCIADO2
14	177	20	ASCII		0	0	0	ASSOCIADO3

INDEX 1: FIELD SEGMENTS: <1>
INDEX 2: FIELD SEGMENTS: <2> <0>
INDEX 3: FIELD SEGMENTS: <11> <0>

FILE DEFINITION - ESTRUTURA ARQ. #6

****

NOME BASE DO ARQUIVO = LOCACAO
NOME PARA USUARIO = LOCACAO
NOME PARA DATAFLEX = LOCACAO

****

TAMANHO REGST = 512 (USADO = 48)
NUMERO MAXIMO DE REGS = 100 (USADO = 11)
ESPACO EXCLUIDO E REUTL
MULTI-USER RE-READ ATIVO

****

FIELD NMBR	FIELD OFFSET	FIELD LEN	FIELD TSPE	DEC PTS	MAIN INDEX	RELATES--TO FILE	FIELD	
1	1	3	NUMERIC	0	1	7	1	NUMERO
2	4	3	DATE		2	0	0	DATA
3	7	10	ASCII		0	0	0	RESPONSAVEL
4	17	3	NUMERIC	0	3	5	1	CODIGO_CLIENTE
5	20	20	ASCII		0	0	0	NOME_CLIENTE
6	40	2	NUMERIC	0	0	0	0	NUM_FITAS_LOC
7	42	6	NUMERIC	2	0	0	0	VALOR
8	48	1	ASCII		0	0	0	BAIXA

INDEX 1: FIELD SEGMENTS: <1>
INDEX 2: FIELD SEGMENTS: <2> <0>
INDEX 3: FIELD SEGMENTS: <4> <0>

FILE DEFINITION - ESTRUTURA ARQ. #7

****

NOME BASE DO ARQUIVO = FICHA
NOME PARA USUARIO = FICHA
NOME PARA DATAFLEX = FICHA

****

```
TAMANHO REGST = 512 (USADO = 16)
NUMERO MAXIMO DE REGS = 100 (USADO = 5)
ESPACO EXCLUIDO E REUTL
MULTI-USER RE-READ ATIVO
*********************************************************
```

FIELD NMBR	FIELD OFFSET	FIELD LEN	FIELD TSPE	DEC PTS	MAIN INDEX	RELATES--TO FILE	FIELD	
1	1	3	NUMERIC	0	1	6	1	LOCACAO
2	4	3	NUMERIC	0	2	3	1	CODIGO_FITA
3	7	5	NUMERIC	2	0	0	0	VALOR_LOCACAO
4	12	5	NUMERIC	2	0	0	0	DESCONTO

```
INDEX 1: FIELD SEGMENTS: <1> <0>
INDEX 2: FIELD SEGMENTS: <2> <0>
```

Uma vez que já temos as definições dos arquivos de dados, podemos passar para o próximo utilitário Dataflex.

Não esqueça estas definições, pois voltaremos a falar delas mais adiante, quando tivermos adquirido um pouco mais de conhecimentos.

9.2 REINDEX

Sua função básica é reindexar um ou mais arquivos de índices ou somente arquivos de índice Batch.

Dataflex pode manter acima de 9 arquivos de índice online por arquivo de dados. Um décimo arquivo de índice pode ser criado, este será obrigatoriamente um índice batch. Este tipo de índice recebe o nome de índice AD HOC.

Cada índice é mantido em um arquivo separado, nomeado com o mesmo nome do arquivo de dados, mas com terminação ".Kn", onde o "n" significa o número do índice daquele arquivo de dados.

O utilitário REINDEX deve ser executado quando, durante operações de manipulação de arquivos, ocorrerem os seguintes problemas:

- Status dos arquivos iguais a 1, 2, 3, 4, 5, 6, 20, 21, 22, 26, 78, 81 ou 82.

Obs.: Status 1 a 6 : disco cheio;
20, 22, 26 : arquivo de índice estragado;
21 : erro de gravação no arquivo de índices;
78 : diretório danificado;
81 : número de registro fora das limitações;
82 : registro editado não gravado.

- Registros parecendo estar fora de ordem durante uma operação de consulta seqüencial indexada (registros ordenados).

- Procurando por um registro e encontrando outro.

- Registros desaparecidos dos arquivos mas tem-se certeza que estão lá.

Para executar o utilitário REINDEX via Menu Principal escolha a alternativa: Utilitários Dataflex e depois RE-INDEXAÇÃO de arquivos de Dados. Ou do sistema operacional: REINDEX. Irá aparecer uma tela como no FILEDEF, contendo todos os nomes dos arquivos de dados com seus respectivos números. Escolha um deles, teclando seu número e veja em seguida a seguinte tela:

DATAFLEX 2.2
REINDEX
CADTEL

1.... RE-CRIA TODOS OS INDICES/CHECA OS DADOS

2.... RE-CRIA INDICES BATCH

3.... RE-CRIA UM INDICE

4.... DEFINE & CRIA INDICE AD HOC

5.... EXCLUI INDICE AD HOC

- 1.... RE-CRIA TODOS OS INDICES/CHECA OS DADOS :

- reconstrói todos os índices
- checa os dados do arquivo e se achar dados estragados, ele dará três opções a respeito do destino destes dados:

1) deixa o arquivo do mesmo modo (sem alterações);
2) lista os dados ruins;
3) envia os registros estragados para um arquivo de terminação ".BAD".

- 2.... RE-CRIA INDICES BATCH :
- cria todos os índices Batch (atualização dos índices).

- 3.... RE-CRIA UM INDICE:
- reconstrói um único índice. Esta opção é normalmente usada quando acabamos de criar um novo índice pelo FILEDEF ou quando algum índice, em específico, apresenta problemas.

- 4.... DEFINE & CRIA INDICE AD HOC :
- Usa-se para criar um índice Batch (AD HOC). Só deve ser usado se o arquivo de dados já possui 9 índices anteriores.

- 5.... EXCLUI INDICE AD HOC :
- remove o arquivo de índice AD HOC criado com a opção anterior.

Existem diversas maneiras de executar o utilitário REINDEX:

1) REINDEX número arquivo- seleciona o arquivo.

2) REINDEX número arquivo MENU - seleciona arquivo e mostra o menu.

3) REINDEX número arquivo BATCH - seleciona arquivo e cria Batch.

4) REINDEX número arquivo número índice - seleciona arquivo e índice.

5) REINDEX ALL - reindexa todos os arquivos.

9.3 ENTERDEF

Sua função principal é, a partir de uma definição de arquivo de dados (.DEF), criar um programa-fonte para entrada de dados. Este programa-fonte é criado com a estrutura de uma Macro ENTER.

Há um único meio de acessar este utilitário: pelo sistema operacional do computador. Digite FLEX ENTERDEF e a seguinte tela aparecerá:

```
    INTERCOMP Interamericana de Computação Ltda
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

    ESTE UTILITARIO GERA UMA CONFIGURACAO
    COM MACRO 'ENTER' A PARTIR DE UM ARQUIVO '.DEF'

  Entre o nome do arquivo .DEF : FORNECED.DEF

  Entre o nome da configuracao a ser criada : FORNECED.FRM
```

O ENTERDEF criará então um programa baseado na Macro ENTER, que poderá ser usado para entrada de dados no arquivo selecionado.

Segue a listagem do programa gerado.

```
    FILE DEFINITION - ESTRUTURA ARQ. #4
**************************************************
    NOME BASE DO ARQUIVO   = FORNECEDORES
    NOME PARA USUARIO      = FORNECEDORES
    NOME PARA DATAFLEX     = FORNECEDOR
**************************************************
    TAMANHO REGST = 512 (USADO = 169)
    NUMERO MAXIMO DE REGS = 100 (USADO = 4)
    ESPACO EXCLUIDO E REUTL
    MULTI-USER RE-READ ATIVO
**************************************************
```

FIELD NMBR	FIELD OFFSET	FIELD LEN	FIELD TSPE	DEC PTS	MAIN INDEX	RELATES--TO FILE	FIELD	
1	1	3	NUMERIC	0	1	0	0	CODIGO
2	4	30	ASCII		2	0	0	NOME
3	34	13	ASCII		0	0	0	CGC
4	47	30	ASCII		0	0	0	ENDERECO
5	77	3	NUMERIC	0	0	0	0	NUMERO
6	80	10	ASCII		0	0	0	COMPLEMENTO
7	90	20	ASCII		0	0	0	CIDADE
8	110	20	ASCII		0	0	0	BAIRRO
9	130	2	ASCII		0	0	0	ESTADO
10	132	5	ASCII		0	0	0	CEP
11	137	4	ASCII		0	0	0	DDD
12	141	9	ASCII		0	0	0	FONE
13	150	20	ASCII		0	0	0	CONTATO

INDEX 1: FIELD SEGMENTS:

INDEX 2: FIELD SEGMENTS:

```
/TELA Programa: FORNeced.frm, Macro ENTER gerada pelo
'ENTERDEF' (c)INTERCOMP
Arquivo:     FORNECEDOR
REGISTRO No  <____>.
CODIGO       |____.|
NOME         |____________________________|
CGC          |_____________|
ENDERECO     |____________________________|
NUMERO       |____.|
COMPLEMENTO  |_____|
CIDADE       |____________________|
BAIRRO       |____________________|
ESTADO       |__|
CEP          |_____|
DDD          |___|
FONE         |_________|
CONTATO      |____________________|
```

```
/*                          // indica final da imagem (tela)

OPEN FORNECEDOR             // abre arquivo

ENTER FORNECEDOR            // macro ENTER

    AUTOPAGE TELA           // enderecamento automatico de janelas
    ENTRY   FORNECEDOR.RECNUM {AUTOFIND} // se existe,
    ENTRY   FORNECEDOR.CODIGO                // mostra
    ENTRY   FORNECEDOR.NOME
    ENTRY   FORNECEDOR.CGC
    ENTRY   FORNECEDOR.ENDERECO
    ENTRY   FORNECEDOR.NUMERO
    ENTRY   FORNECEDOR.COMPLEMENTO
    ENTRY   FORNECEDOR.CIDADE
    ENTRY   FORNECEDOR.BAIRRO
    ENTRY   FORNECEDOR.ESTADO
    ENTRY   FORNECEDOR.CEP
    ENTRY   FORNECEDOR.DDD
    ENTRY   FORNECEDOR.FONE
    ENTRY   FORNECEDOR.CONTATO
    RETURN                  // final da secao ENTRY
ENTEREND                    // finaliza macro ENTER
ABORT                       // abandona esta configuracao
```

Para executar esta aplicação, você deve primeiro compilá-la.

Para compilar esta aplicação, do Menu principal vá ao Utilitários Dataflex e escolha a opção Compilar um Programa. Do sistema operacional digite COMP FORNECED.FRM .

Uma vez compilado execute: do Menu Principal escolha Executar um Programa. Do sistema operacional digite FLEX FORNECED . E boa sorte.

9.4 MENUDEF

É sistema de criação e manipulação de menus, que possibilita a execução de qualquer programa ou aplicação Dataflex ou comando do sistema operacional.

A melhor maneira de entender isto é executando o utilitário do seguinte modo: do Menu Principal vá ao Utilitários Dataflex e escolha o item Definindo Menus. Do sistema Operacional digite FLEX MENUDEF . A seguinte tela aparecerá:

Sistema Menu DataFlex Selecao Menu
= = = =

As teclas Registro Posterior e Anterior podem ser usadas para mostrar outros menus. Pressione <RETURN> para começar novo menu (HELP disponível).

Temos aqui uma tela auto-explicativa, mas escolha um dos menus já existentes, para consulta ou alteração, ou tecle ENTER para a inclusão de um novo menu. Escolhamos o Menu Principal, número 1:

```
Sistema Menu DataFlex                              Configuracao
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

Numero do Menu : 1      Cabecalho 1:    ---- DataFlex ----
                        Cabecalho 2:    Menu Principal

                        Menu Default:   1 (no return)

     DESCRICAO                          ACAO              SENHA
 1. Menu Exemplos de Aplicacao          MENU 2
 2. Query Banco de Dados                ychange QUERY
 3. Menu Utilitarios DataFlex           MENU 3
 4. Menu de Utilitarios de Sistema      MENU 4
 5. Executar um Programa DataFlex       CHAIN $1
 6. Executar um Comando do Sistema      $2
 7. Sair p/ o Sistema Operacional       SYSTEM

(N)ovo Menu  (C)abecalho  (Q)uestoes  (G)ravar      Im(P)rimir  {HELP
(I)nserir    (E)xcluir    (A)lterar   Comple(T)ar  (S)aida _ DISPONIVEL}
```

Esta é a tela de configuração de um Menu já criado. A seguir temos a tela de HELP para tela anterior (para acessar a tela de HELP digite F1, caso existam mais telas de HELP digite novamente F1 para a próxima tela ou qualquer tecla volta para a tela de onde saiu). Estas telas de HELP foram acrescidas de comentários para uma melhor clareza da finalidade das informações:

Sistema Menu DataFlex HELP MENU (tela1)

= = = =

Sistema Menu DataFlex HELP MENU (Tela 2)
= = = =

Sistema Menu DataFlex HELP MENU (Tela 3)

= = = =

Sistema Menu DataFlex HELP MENU (tela 4)

Estes sao os comandos possiveis de serem usados no campo ACAO para podermos executar os seguintes tipos de programas e aplicacoes:

Podem ser executados todos os programas ".COM" ou ".CMD" (estando incluidos os argumentos necessarios para sua perfeita execucao).

Seguem os comandos para a opção ACAO (os itens dentro dos parenteses sao os paramentros que devem ser fornecidos):

CHAIN <nome do programa> : Chain para outro programa FLX.
MENU <numero do menu> : Ira para outro menu.
SYSTEM : Retornara para o sistema operacional.
ERASEFILE <nome-do-arquivo> : Excluira um arquivo do disco.
TYPE <nome-do-arquivo> : Imprimira, no monitor de video, o arquivo selecionado um arquivo. Mostrara um arquivo sequencial.
RENAMEFILE <de> <para> : Trocara o nome do arquivo.
COPYFILE <de> <para> : Copiara um arquivo em disco (os sinais "*" e "?", usados para especificao de arquivos, sao permitidos).
DIRECTORY <espec.drive>: Mostrara o diretorio selecionado.

$ <questao numero>: Mostrara a questao criada anteriormente. As questoes nao estao ligadas, diretamente, a uma ou outra opcao, portanto, cada uma delas pode ser usada separadamente para qualquer opcao. As questoes podem ser usadas como entradas de parametros para os diversos comandos citados acima. Tais parametros podem ser : nomes de arquivos, nome do drive, nomes de programas a serem executados, deletados, etc.

Poderemos utilizar o MENUDEF para a criacao do Menu Principal do sistema de locadoras de video.

Para iniciar, va ao menu principal do MENUDEF e tecle ENTER, para a inclusao de um novo Menu. A seguinte tela ira aparecer e poderemos preenche-la com os seguintes dados.

Sistema Menu DataFlex Configuracao

= = = =

Veja que na opção 4 do menu de locadoras há a indicação de um outro menu, MENU 6, o qual estará encarregado dos processamentos da locação das fitas de vídeo. Este menu deverá ser criado teclando-se ENTER, quando da apresentação da tela principal deste utilitário ou escolhendo a opção "(N)ovo Menu". Devemos entrar então com os seguintes dados:

Cabecalho 1: --- SISTEMAS DE LOCADORAS ---
Cabecalho 2: Menu de Processos com Locacao
Menu Default: 5 (no return)

LOCACAO	chain locacao
BAIXA DA LOCACAO	chain baixaloc
PESQUISA DE LOCACOES	chain pesqloc

Em seguida devemos gravar este menu, escolhendo a opção "(G)ravar". Com resultado teremos a seguinte tela :

Sistema Menu DataFlex Configuracao
= = = =

PARTE III
COMANDOS BÁSICOS

Nos capítulos seguintes veremos os comandos básicos para programação em Dataflex. Iniciaremos com os comandos de Decisão IF - ELSE, Indicadores, depois veremos comandos de desvio de fluxo de execução de programas (GOTO, GOSUB, ON GOTO, ON GOSUB, RETURN e POP), seguindo veremos os comandos de laço, looping (FOR - FROM - TO - LOOP, BEGIN - END, CASE / CASEELSE - CASEEND, REPEAT - UNTIL e WHILE - END), veremos também os comandos para manipilação numérica e de *strings* e por final os comandos de manipulação de arquivos (OPEN, CLEAR, SAVE, DELETE e FIND).

Antes que iniciemos o primeiro capítulo desta nova parte do livro devemos conhecer como Dataflex faz comparações, isto é, quais os subcomandos utilizados por ele para fazer estas comparações.

Igual	: EQ	Menor	: LT
Maior	: GT	Maior Igual	: GE
Menor Igual	: LE	Diferente	: NE
Dentro	: IN		

Juntamente com estes modos de comparação pode-se usar a partícula NOT, como negação a qualquer um destes modos.

PARTE III
COMANDOS BÁSICOS

CAPÍTULO 10
COMANDO DE DECISÃO
IF - ELSE

O comando IF / ELSE tem como função checar se determinada condição é ou não verdadeira; no caso de ser, será executado um comando ou bloco de comandos, caso contrário, um outro comando ou outro bloco de comandos será executado por sua vez.

Graficamente teríamos:

```
SE argumento1 MODO argumento2
      verdade inicie
         execute comando1
         execute comando2
        fim
      caso contrario inicie
                  comando3
                  comando4
                  fim
fim do SE
```

Sua sintaxe é:

1) IF condição comando <ou bloco de comandos>

2) IF condição comando1 <ou bloco de comandos1>
ELSE comando2 <ou bloco de comandos2>

Condição neste caso significa:

argumento1 MODO argumento2

No caso 1 teremos: se a condição for verdadeira, então, será executado o comando ou o bloco de comandos.

No caso 2 teremos: se a condição for verdadeira, será executado o comando1 ou um bloco de comandos1, caso contrário (se a condição for falsa), será executado o comando2 ou um bloco de comandos2.

Exemplo:

```
IF VALOR_VENDA GT 10000
      BEGIN
         CALC ( VALOR_VENDA / NUMERO_PARCELAS )TO VALOR_PARCELAS
         DISPLAY VALOR_PARCELAS TO TELA.4
      END
ELSE
      BEGIN
         CALC ( VALOR_VENDA * 0.12 ) TO VALOR_PARCELAS
         DISPLAY VALOR_VENDA - VALOR_PARCELAS TO TELA.6
      END
```

CAPÍTULO 11
INDICADORES

Indicadores funcionam como comandos IF, mas são mais rápidos e ocupam menos espaço, pois geram menos linhas intermediárias, durante a compilação do comando.

Indicadores são variáveis de memória de tamanho um bit e conteúdo verdadeiro ou falso, que servem para checagem de linhas de comando ou blocos de comando.

Um indicador é criado através do comando: INDICATOR ind1 < ind2 ... ind9 >, que indicará ao compilador que deverá ser reservado espaço, em memória, para esta nova variável criada. Este indicador criado deve ser representado entre colchetes "[ ]".

Indicadores, não necessariamente, precisam ser declarados para podermos usá-los. Durante um programa Dataflex pode-se criar indicadores sem que estes tenham sido declarados anteriormente. Isto é feito usando-se o comando INDICATE ind1 < ind2 ... ind 9 > .

Quando é criado um indicador, não se sabe o valor inicial que lhe é atribuído, portanto, devemos inicializá-lo com um valor conhecido (TRUE ou FALSE , verdadeiro ou falso). Esta inicialização pode ocorrer
dos seguintes modos:

1) INDICATE ind1 TRUE ou FALSE
2) INDICATE ind1 AS argumento1 modo argumento2
3) INDICATE ind1 AS [ind2]

4) INDICATE ind1 GROUP ANY/ALL [ind2 ind3 ind4] AND/OR [ind5 ind6 ind7]
5) INDICATE ind1 STATUS nome_arquivo

No caso 1: atribuímos valor verdadeiro (TRUE) ou falso (FALSE) ao indicador ind1.

No caso 2: atribui-se ao indicador ind1 um valor (TRUE ou FALSE) de acordo com a condição que se segue. O comando INDICATE verificará se argumento1 modo argumento2 é verdadeiro; se for, setará o indicador ind1 como TRUE, caso contrário, se argumento1 modo argumento2 for falso, o indicador ind1 ficará com status FALSE.

No caso 3: atribui-se o valor do indicador ind1 de acordo com o status de outro indicador ind2.

No caso 4: seta-se o status de um indicador (ind1) de acordo com os status de todos ou algum (ANY / ALL) dos indicadores do primeiro argumento (ind2, ind3 e ind4) e/ou (AND / OR) dos indicadores do segundo argumento (ind5, ind6 e ind7). Os argumentos e partes de comando a partir do final do primeiro grupo de indicadores é opcional.

No caso 5: seta-se o status de ind1 como verdadeiro se houver algum registro residente no buffer de registro do arquivo declarado, caso contrário será setado como falso.

Um indicador deve vir obrigatoriamente à esquerda do comando ou bloco de comandos que se deseja executar se o seu status for verdadeiro; caso não seja, o comando ou bloco de comandos não será executado.

Exemplo:

```
INDICATE achei TRUE
INDICATE e_hoje AS data_contrato EQ data_sistema
INDICATE cidade AS [ achei ]
INDICATE found GROUP ALL [ achei e_hoje ] AND [ pago_em_dia ]
INDICATE faco_locacao STATUS locacao
```

CAPÍTULO 12
COMANDOS DE DESVIO

São aqueles que desviam a execução do programa para uma rotina, um label ou uma sub-rotina, não sendo necessariamente obrigatória a volta da execução ao ponto de onde saiu a ordem de desvio.

Graficamente teríamos:

```
Comeco:
  ...
  comando 1
  desvie rotina2
  comando 8
  va para comeco
...

Rotina2:
  Se condicao inicie
       comando 3
       desvie para rotina1
       comando 7
     fim
fim Se
retorne
```

```
Rotina1:
   comando 4
   comando 5
   comando 6
retorne
```

12.1 GOTO

Desvia a execução do programa, incondicionalmente, para uma rotina, continuando a execução a partir deste ponto, quebrando a seqüência natural de execução.

Sua sintaxe é:

```
GOTO rotina
```

Neste caso teremos: o programa será executado normalmente até que se encontre um comando "GOTO rotina", o Dataflex desviará o fluxo do programa para a rotina chamada, continuando a execução do programa a partir da rotina até que se encontre outro comando de desvio ou o fim do programa.

Exemplo:

```
Inicio:
   MOVE 0 TO diferenca
   MOVE 0 TO contador
   MOVE 1 TO pagina
   ACCEPT tela.2
   IF tela.2 EQ "S" goto inicio
   CALC ( tela.1 * 1000 ) to contador
   CALC ( contador + 0.1 * contador ) TO diferenca
   DISPLAY contador TO tela.3
   GOTO fim
```

12.2 GOSUB

Transfere a execução do programa para uma rotina, executa-a até que encontre o comando "RETURN", que indicará o fim da rotina e o retorno da execução ao ponto de partida da ordem de transferência.

Sua sintaxe é:

GOSUB rotina

Teremos: a execução normal do programa, até que seja encontrado o comando "GOSUB rotina", onde o fluxo do programa será desviado para uma rotina declarada e, ao seu término (comando RETURN), retornará a execução para a linha seguinte ao comando Gosub.

Exemplo:

```
INDICATE achei STATUS locacao
    Comeco:
        GOSUB Verifica
        GOSUB Confirma
        GOSUB Grava_locacao
        MOVE 0 TO diferenca
        GOTO Comeco
```

12.3 ON GOTO

Este comando tem por função o desvio incondicional do fluxo do programa para um label e continuar a execução a partir daí, isto é, de acordo com o valor de uma variável inteira, o fluxo será desviado para cada um dos labels da lista, contidos no comando.

Sua sintaxe é:

ON var_inteira GOTO label1, label2, ..., labeln

Neste caso teremos: conforme o valor alcançado pela variável inteira (1, 2, 3, 4, .., n), o fluxo será desviado para o label na posição correspondente na lista de labels.

Exemplo:

```
ACCEPT tela.5 {range = 1,4}
MOVE tela.5 TO opcao
ON opcao GOTO opcao1, opcao2, opcao3, opcao4
```

12.4 ON GOSUB

Este comando tem as mesmas características do anterior, só que o desvio será feito para rotinas, tendo o retorno, ao final desta rotina, com o comando RETURN, devolvendo o controle para a linha seguinte ao comando de desvio.

Sua sintaxe é:

ON var_inteira GOSUB rotina1, rotina2, rotina3,.., rotina n

No caso teremos: execução das rotinas de acordo com o valor da variável inteira (vide ON GOTO).

Exemplo:

```
ACCEPT tela.5 {range = 1,4}
MOVE tela.5 TO opcao
ON opção GOSUB opcao1, opcao2, opcao3, opcao2
GOTO Inicio
```

12.5 RETURN

Tem como função indicar o fim de uma rotina e retornar o controle do programa para a linha logo abaixo do comando Gosub que chamou a rotina ou indicar o fim da rotina e desviar o fluxo para uma outra rotina.

Sua sintaxe é:

1) RETURN < rotina >

Nestes casos: sem especificar a rotina, ocorre o retorno normal para o Gosub que chamou a rotina. Especificando a rotina, indica fim da rotina em que se encontra e desvia o fluxo de execução para uma rotina específica, funcionando como um GOTO.

Exemplo:

```
Grava_locacao:
    MOVE locacao.numero TO num_locacao
    MOVE tela.6 TO num_fitas
    MOVE "N" TO locacao.baixa
    SAVE locacao
    RETURN

Verifica:

    SHOW " Confirma locacao (S/N) ?"
    INKEY tecla
    IF tela EQ "N" RETURN Inicio
    RETURN
```

12.6 POP

Sua função é executar um comando RETURN sem desviar o fluxo de execução para o Gosub que chamou a rotina em questão.

Sua sintaxe é:

```
POP             (sem argumentos)
```

Exemplo:

```
    GOSUB rotina1
    GOSUB rotina2
    GOTO Inicio
Rotina1:
    SHOW "Rotina 1"
    CALC ( valor - diferenca ) TO desconto
    DISPLAY desconto TO tela.7
    GOSUB Rotina3
    SHOW " voltando Rotina3"
    RETURN

Rotina2:
    SHOW "Rotina 2"
    DISPLAY desconto TO tela.4
    SHOW "fim"
    RETURN

Rotina3:
    SHOW " Rotina 3"
    CALC ( valor + total_a_pagar    ) TO total_a_pagar
    CALC ( total_a_pagar - desconto ) TO total_a_pagar
    DISPLAY total_a_pagar TO tela.9
    POP
    CLEARSCREEN
    ...
```

CAPÍTULO 13
ESTRUTURAS DE LAÇO (LOOP)

Estruturas de Laço são basicamente blocos de instruções, que dependendo de sua função podem ser de repetição, de checagem ou somente para definição de blocos de comandos.

Obs.: Não use nenhum desses comandos dentro das Macro Enter, Entergroup, pois durante a execução dessas macros, o Dataflex, em uma operação de procura (Find, Autofind, etc) não executa os comandos entre os ENTRY.

13.1 BEGIN - END

Sua função é definir um bloco de comandos, uma vez que alguns comandos Dataflex só aceitam múltiplos comandos em blocos. Begin - End definem, respectivamente, o início e o fim do bloco de comandos (um conjunto de dois ou mais comandos podem ser definidos como um bloco de instruções).

Sua sintaxe é:

```
BEGIN
   comando1
   comando2
   ...
   comando n
END
```

Exemplo:

```
[achei] begin
        SHOW " Esta locacao ja existe !!"
        INKEY tecla
        RETURN comeco
      end

IF opcao EQ "V" begin
                MOVE -1 TO pagefeed
                MOVE 20 TO pageend
                MOVE "CON:" TO saida
             end
```

13.2 CASE / CASEELSE - CASEEND

Tem como função iniciar e terminar um bloco de instruções que serão executadas se a condição determinante for verdadeira, até que o comando CASEEND seja encontrado. Podem haver diversos blocos Case Caseend encadeados.

Um grupo de comandos Case deve ser precedido de um comando CONDITIONAL, que inicia o bloco e deve ser sucedido por um comando CONDITIONALEND, que o encerra. Opcionalmente o comando CASEELSE pode ser usado para o caso de todos os Case anteriores do

bloco serem falsos, funcionando normalmente como um Case, sem a checagem de condição, terminando obrigatoriamente com um Caseend, seguido de um Conditionalend.

Sua sintaxe é:

```
CONDITIONAL
  CASE condicao determinante1
    comando1
    comando2
    ...
    comando n
  CASEEND
  CASE condicao determinante2
    comando1
    comando2
    ...
    comando n
  CASEEND
  CASEELSE
    comando1
    comando2
    ...
    comando n
  CASEEND
CONDITIONALEND
```

Obs.: Neste caso não se faz necessária a estrutura BEGIN - END, pois o bloco já o engloba. (Válido para as demais estruturas de bloco.)

Exemplo:

```
CONDITIONAL
  CASE  opcao eq "V"
    MOVE -1 TO pageend
    MOVE 20 TO pagefeed
    MOVE "CON:" TO saida
  CASEEND
```

```
    CASE opcao EQ "I"
      MOVE 0 TO pageend
      MOVE 55 TO pageend
      MOVE "" TO saida
    CASEEND
    CASE opcao EQ "A"
    ACCEPT tela.5
      MOVE 0 TO pageend
      MOVE 55 TO pagefeed
      MOVE tela.5 to saida
    CASEEND
    CONDITIONALEND
```

13.3 FOR - FROM - TO - STEP - LOOP

Basicamente este é um comando de repetição, executando um comando, ou bloco de comandos, um determinado número de vezes.

Sua sintaxe é:

```
FOR var_int FROM {var1_int } TO/DOWN {var2_int } [ STEP {var3_int } ]
      {const1_int}  {const2_int} [  {const3_int} ]
      comando1
      comando2
      ...
      comando n
    LOOP
```

Exemplo:

```
INTEGER contador quadrado

MOVE 10 TO contador
move 0 to quadrado

FOR contador FROM 2 TO 100 STEP 2
     SHOW " numero  par : " contador
     CALC ( contador * contador ) TO quadrado
     SHOW " quadrado  : " quadrado
LOOP
```

13.4 REPEAT - UNTIL

É uma estrutura de repetição e checagem, na qual um grupo de comandos ou um único comando é executado até que uma condição determinante seja verdadeira. A checagem da condição é feita ao final do bloco de comandos, sendo que, obrigatoriamente, este bloco será totalmente executado antes que a condição seja checada. A condição determinante pode ser tanto uma expressão lógica como um indicador ou um grupo deles (devem estar entre colchetes "[ind1 ind2 ind3]", sendo somente obrigatório o uso do primeiro indicador); neste caso a condição determinante só será verdadeira se o status de todos os indicadores, ou do grupo de indicadores, for verdadeiro.

Sua sintaxe é:

```
REPEAT
   comando1
   comando2
   comando3
   ...
   comando n
UNTIL condicao determinante
```

Obs.: condição determinante para indicadores:

1) [ind1 ind2 ind3]
Neste caso a condição determinante só será verdadeira se o ind1, o ind2 e o ind3 tiverem status verdadeiro.

2) [not ind1 ind2 not ind3]
Neste caso a condição determinante só será verdadeira se o ind1 for falso (não-verdadeiro), se o ind2 for verdadeiro e o ind3 for não-verdadeiro (falso).

Exemplo:

```
REPEAT
   ACCEPT tela.2
   ACCEPT tela.3
   ACCEPT tela.4
   ACCEPT tela.5
   CALC ( tela.4 - tela.5 ) TO soma
UNTIL tela.2 EQ 0

MOVE num_locacao TO locacao.numero
REPEAT
   FIND GT locacao BY index.1
   [found] CALC (locacao.valor + somatorio) TO somatorio
UNTIL [not found]
```

13.5 REPEAT - LOOP

Sua função é idêntica ao Repeat - Until, só que a checagem não vai ser feita pelo Until mas por um If ou um Indicador.

Sua sintaxe é:

```
REPEAT
   comando1
   comando2
   ...
   condicao determinante LOOP
```

Exemplo:

```
MOVE num_locacao TO locacao.numero
REPEAT
   FIND GT locacao BY index.1
   [found] CALC (locacao.valor + somatorio) TO somatorio
[not found] LOOP
```

13.6 WHILE - END

O WHILE - END checa uma determinada condição e, caso ela seja verdadeira, executa um bloco de instruções até o fim e volta ao início do bloco para nova checagem da condição. Caso a condição seja falsa o bloco de comandos não será executado, passando o fluxo do programa para a próxima instrução. A condição determinante pode ser tanto uma expressão lógica como um indicador ou um grupo deles (devem estar entre colchetes "[ind1 ind2 ind3]", sendo somente obrigatório o uso do primeiro indicador);

neste caso a condição determinante só será verdadeira se o status de todos os indicadores, ou do grupo de indicadores, for verdadeiro.

Sua sintaxe é:

```
WHILE condicao determinante
  comando1
  comando2
  ...
  comando n
END
```

Obs.: condição determinante para indicadores:

1) [ind1 ind2 ind3]
Neste caso a condição determinante só será verdadeira se o ind1, o ind2 e o ind3 tiverem status verdadeiro.

2) [not ind1 ind2 not ind3]
Neste caso a condição determinante só será verdadeira se o ind1 for falso (não-verdadeiro), se o ind2 for verdadeiro e o ind3 for não-verdadeiro (falso).

Exemplo:

```
MOVE num_locacao TO locacao.numero
INDICATE found TRUE

WHILE [found]
  FIND GT locacao BY index.1
  [found] CALC (locacao.valor + somatorio) TO somatorio
END
```

CAPÍTULO 14
COMANDOS DE PROCESSAMENTO NUMÉRICO

Os comandos de processamento numérico têm como finalidade realizar cálculos, conversões de tipos de variáveis e servir de incrementadores de contadores.

14.1 CALC ou CALCULATE

Realiza todas as quatro operações básicas em matemática: (soma, subtração, divisão e multiplicação) guardando os resultados em variáveis, que podem ser inteiras, numéricas, datas, campos de arquivos e janelas.

Sua sintaxe é:

CALC (expressao matematica) TO var1
ou
CALCULATE (expressao matematica) TO var1

Exemplo:

```
CALC ( 1000 * 1000 ) TO quadrado_de_mil
CALC ( valor - desconto ) TO total_a_pagar
CALC ( total_a_pagar + valor ) TO total_a_pagar
CALC ( total_a_pagar / num_parcelas ) TO parcela
```

14.2 INCREMENT / DECREMENT

Sua função é somente de somar um ou diminuir um em uma variável inteira. Seu principal uso é o de servir de somador ou decrementador de um contador para estruturas de repetição de comandos.

Suas sintaxes são:

```
INCREMENT var_int1
DECREMENT var_int1
```

Exemplo:

```
INTEGER contador num_fitas

MOVE 1 TO contador
MOVE 0 TO num_fitas

ACCEPT num_fitas TO tela.6
REPEAT
    DISPLAY contador TO tela.7
    ACCEPT codigo TO tela.8
    ACCEPT titulo TO tela.9
    ACCEPT valor TO tela.10
    ACCEPT desconto TO tela.11
    INCREMENT contador
UNTIL contador GT num_fitas
```

14.3 MOVE

Sua função é a de mover dados de uma posição para outra. Estes dados podem ser valores de expressões matemáticas, de variáveis inteiras, numéricas, datas, de campos de arquivos, de janelas e de status de indicadores para outras janelas, campos de arquivos, indicadores e variáveis. No caso da "variável-destino" não ser do mesmo tipo da "variável-fonte" o comando MOVE converterá automaticamente o tipo da "variável-fonte" para o tipo da "variável-destino". Se a "variável-fonte" que você está movendo tiver de ir para mais de uma "variável-destino" você não precisará de "n" MOVEs para isto, bastando apenas um único comando com no máximo 8 "variáveis-destino" (todas estas variáveis terão o mesmo conteúdo da "variável-fonte").

Suas sintaxes são:

MOVE { const1 }	TO { var_int1	[var_int2 ... var_int n	] }
{ expr1 }	{ var_num1	[var_num2 ... var_num n	] }
{ var_int1 }	{ var_data	[var_data ... var_data n	] }
{ var_num1 }	{ ind1	[ind2 ... ind3	] }
{ var_data }	{ campo1	[campo2 ... campo n	] }
{ ind1 }	{ janela1	[janela2 ... janela n	] }
{ campo1 }			
{ janela1 }			

Exemplo:

```
DATE   data1 data2
STRING string_data string_num
INTEGER diferenca
MOVE 10/01/88      TO data1
MOVE 11/01/88      TO data2
MOVE ""            TO string_data
MOVE   data1 TO string_data
MOVE ( data2 - data1 ) TO diferenca
CALC ( diferenca * 6 ) TO diferenca
MOVE diferenca TO string_num
```

CAPÍTULO 15
COMANDOS DE MANIPULAÇÃO DE STRINGS

Os comandos de manipulação de strings servem basicamente para um processamento de separação ou concatenação de variáveis e campos string. Neste capítulo serão explicados os seguintes comandos de manipulação : RIGHT, LEFT MID E APPEND. Vejamos a seguir as explicações a respeito de cada um destes comandos. Usaremos a notação de data européia (DD/MM/AA).

15.1 LEFT

Seleciona um determinado conjunto de caracteres, à esquerda, de uma variável, janela ou campo ASCII, guardando-o numa variável ou janela do tipo ASCII.

Sintaxe:

LEFT str1 TO str2 (const_int1)
(var_int1)

Este comando irá retirar determinado número (Const_int1, Var_int1) de caracteres mais à esquerda da Str1 e colocá-los em Str2. Se o número especificado for menor que um (1), não haverá efeito algum, e se o número especificado for maior que o tamanho da Str1, toda esta será movida para Str2. Tanto Str1 como Str2 podem ser variáveis, janelas ou constantes ASCII (representadas entre aspas "").

Exemplos:

1) LEFT "Sistemas Integrados" TO texto 5

   Como resultado na variável TEXTO : Siste

2) Sendo o conteúdo da variável NOME = "CARLOS AUGUSTO DE OLIVEIRA"

   LEFT nome TO parte_nome 8

   Teríamos a resposta na variável PARTE_NOME : CARLOS A

3) Para datas o processo é um pouco diferente. Deve-se mover primeiramente a variável, constante, ou campo data, para uma variável string de tamanho 8 caracteres. Aplica-se então o comando:

   LEFT data TO dia 2

Deste modo a variável DIA conteria os dois primeiros caracteres da variável DATA, isto é, conteria o dia da data. Vejamos um exemplo de parte de um programa Dataflex demonstrando como poderíamos fazer para extrair o dia de uma data:

```
...
    MOVE 20/02/88 TO data
    LEFT data TO dia 2
    showln dia
...
```

Como resposta teríamos: "20"

15.2 RIGHT

É um comando parecido com o LEFT, mas retira de uma string-fonte (constante ou variável string, janela ou campo ASCII) um certo número de caracteres à direita desta, guardando-os numa outra variável string, janela ou campo ASCII.

Sintaxe:

RIGHT str1 TO str2 (const_int1)
(var_int1)

Este comando irá retirar determinado número (Const_int1, Var_int1) de caracteres mais à direita da Str1 e colocá-los em Str2.

Se o número especificado for menor que um (1), não haverá efeito algum, e se o número especificado for maior que o tamanho da Str1, toda esta será movida para Str2. Tanto Str1 como Str2 podem ser variáveis, janelas ou constantes ASCII (representadas entre aspas "").

Exemplo:

1) RIGHT "Sistemas Integrados" TO texto 7

Como resultado teríamos a variável TEXTO : egrados

2) Sendo o conteúdo da variável NOME = "CARLOS AUGUSTO DE OLIVEIRA"

RIGHT nome TO parte_nome 11

Teríamos a resposta na variável PARTE_NOME : E OLIVEIRA

3) Para datas o processo é o mesmo do comando LEFT. Deve-se mover a variável, constante, ou campo data, para uma variável string de tamanho 8 caracteres. E aplica-se então o comando:

```
RIGHT data TO ano 2
```

Deste modo a variável ANO conteria os dois últimos caracteres da variável DATA, isto é, conteria o ano da data. Vejamos um exemplo, parte de um programa Dataflex demostrando como poderíamos extrair o ano de uma data:

```
...
MOVE 20/02/88 TO data
RIGHT data TO ano 2
showln ano
...
```

Como resposta teríamos: "88"

15.3 MID

Sua função é retirar de uma string-fonte um pedaço, um conjunto de caracteres, de determinado tamanho, partindo de uma determinada posição dentro da string-fonte.

Sintaxe:

```
MID str1 TO str2    (const_int1   < const_int2   >)
                    (const_int1   <var_int2      >)
                    (var_int1     <const_int2    >)
                    (var_int1     <var_int2      >)
```

Este comando retira de Str1 um conjunto de caracteres, de tamanho (Const_int1, Var_int1), a partir da posição (Const_int2, Var_int2) da Str1, guardando o resultado em Str2.

Const_int2 e Var_int2 devem, obrigatoriamente, ser maiores que zero e menores ou iguais ao tamanho da string-fonte.

Se o argumento tamanho (Const_int1, Var_int1) for menor que um (1), não haverá efeito algum e se for maior que o tamanho da string-fonte1, será movida toda a string à direita do argumento posição para a string-destino. Tanto Str1 como Str2 podem ser variáveis, janelas ou constantes ASCII (representadas entre aspas "").

Exemplo:

1) MID "Sistemas Integrados" TO texto 5 4

Como resultado teríamos a variável TEXTO : temas

2) Sendo o conteúdo da variável NOME = "CARLOS AUGUSTO DE OLIVEIRA"

MID nome TO parte_nome 5 8
Teríamos a resposta na variável PARTE_NOME : AUGUS

3) Para datas o processo é um tanto complicado. Variáveis string em formato data contém as barras de separação dia - mês e mês - ano. Portanto devemos nos desviar desses empecilhos. Usaremos então o comando MID da seguinte maneira:

MID data TO mes 4 2

Deste modo a variável MS conteria os dois caracteres centrais da variável DATA, isto é, conteria o mês da data. Vejamos um exemplo, parte de um programa Dataflex demonstrando como poderíamos extrair o mês de uma data:

```
...
MOVE 20/02/88 TO data
MID data TO mes 4 2
showln mes
...
```

Como resposta teríamos : "02"

Um exemplo completo para separar dia, mês e ano de uma data seria :

```
/*
STRING data 8 dia 2 mes 2 ano 2
DATE  data_de_hoje

MOVE 12/02/88 TO data_de_hoje

LEFT  data_de_hoje TO dia 2
MID   data_de_hoje TO mes 4 2
RIGHT data_de_hoje TO ano 2

SHOW dia "-" mes "-" ano
ABORT
```

15.4 APPEND

Sua função é concatenar duas ou mais strings em uma única.

Sintaxe:

```
APPEND str_dest str1 str2 ... str n
```

Este comando irá concatenar as strings Str1, Str2 até Strn e guardará o resultado em Str_dest. Tome cuidado, pois a concatenação pode gerar uma string maior do que aquela que a vai receber. Se isto acontecer a Str_dest será preenchida somente com a quantidade de caracteres suficientes para completá-la, ignorando o resto da string resultante da concatenação. Este comando pode ser usado para inversão de datas, para a junção de diversos conteúdos string de variáveis de memória, janelas e campos de arquivos.

Deve-se tomar cuidado quando Str_dest for uma janela ou campo de arquivo de dados, pois se ocorrer uma concatenação falha, seus dados estarão arruinados. Para evitar erros use sempre que possível variáveis de memória.

Para que o comando APPEND seja executado sem problemas de concatenação, recomenda-se que a Str_dest esteja zerada, isto é, que esteja vazia. Isto é conseguido com o comando MOVE : MOVE "" TO Str_dest. Deste modo nada será movido para a Str_dest.

Exemplo:

Vejamos as definições das variáveis que serão usadas nos exemplos a seguir. Estas definições serão a base para o bom entendimento do que acontece com as variáveis-destino e componentes da concatenação.

```
/*
STRING destino 25 fonte1 3
STRING fonte2 11
STRING nome 20

MOVE "Burd Sistemas" TO nome
MOVE "e" TO fonte1
MOVE "Treinamento" TO fonte2
MOVE "" TO destino
```

1) APPEND destino fonte2 nome

Como resultado teríamos DESTINO : TreinamentoBurd Sistemas

Veja que o resultado final não ficou muito bom. Para melhorar podíamos inserir um espaço entre Fonte2 e Nome (não devemos nos esquecer de zerar a variável-DESTINO):

```
APPEND destino fonte2 " " nome
```

O resultado seria o seguinte : Treinamento Burd Sistemas

2)
```
MOVE "" to destino
APPEND destino nome " " fonte1
APPEND destino " " fonte2
SHOWLN destino
```

Como conteúdo da variável-DESTINO : Burd Sistemas e Treinamento

CAPÍTULO 16
COMANDOS DE MANIPULAÇÃO DE ARQUIVOS

Comandos de manipulação de arquivos são aqueles que permitem as operações com bancos de dados, isto é, permitem que sejam feitas inclusões, alterações, deleções e procuras dentro dos arquivos dos bancos de dados.

16.1 OPEN

Sua função é de abrir um arquivo de dados e deixá-lo disponível para operações de inclusão, alteração, deleção e procura de registros dentro deste arquivo, além de reservar espaço na memória do micro para o buffer do registro do arquivo, deixando-o com status verdadeiro. Após o comando OPEN, o primeiro registro do arquivo já está disponível no buffer para todas as operações citadas acima.

Sintaxes:

1) OPEN nome_arquivo

2) OPEN nome_arquivo INDEX.n

Nestes casos:

1) abre o arquivo desejado e os arquivos de índices, se existirem, e aloca espaço para buffer de registro.

2) abre o arquivo desejado e o arquivo de índice especificado, bufferizando-o, tornando o processamento mais rápido, porém somente para este índice de pesquisa (é usado geralmente durante operações de relatórios, onde o acesso será seqüencial aos registros).

Obs.: Há mais uma sintaxe do comando OPEN:

OPEN nome_arquivo AS path_nome_arquivo [INDEX.n]

Obs.: path_nome_arquivo significa o diretório onde se encontra o arquivo além do arquivo em si.

Neste caso ocorre que será aberto um arquivo que não consta do FILELIST.CFG (arquivo que guarda os nomes dos arquivos de dados, sendo o arquivo do Dicionário de Dados). Isto é possível porque esta abertura será durante a compilação e não durante a execução do mesmo.

Exemplo:

```
OPEN locacao INDEX.1
OPEN cadcli INDEX.2
OPEN cadfitas
OPEN fornecedores
```

16.2 CLEAR

Este comando tem por finalidade deixar vazio o buffer do arquivo especificado, previamente aberto com o comando OPEN, deixando-o com status falso.

Sua sintaxe é:

CLEAR nome_arquivo

Exemplo:

CLEAR locacao
CLEAR cadcli
CLEAR cadfitas
CLEAR fornecedores

16.3 SAVE

Sua função é gravar os dados do buffer de registro selecionado no arquivo de dados, isto é, passar os dados do buffer para o meio de armazenamento. Em arquivos relacionados, o comando move os dados do arquivo principal para os relacionados secundários, gravando-os em seguida.

É usado para efetuar uma alteração ou inclusão em um arquivo de dados.

Sintaxe:

SAVE nome_arquivo

Obs.: Não use o comando Save dentro das Macros ENTER e ENTERGROUP, pois os dados ainda não foram movidos para o buffer de registro, portanto o comando Save gravará somente registros em branco, logo este comando deverá ser colocado após um Enterend ou um Entergroup, para que possa realizar sua função plenamente.

Exemplo:

```
SAVE locacao
SAVE cadfitas
SAVE ficha
```

16.4 SAVERECORD

Tem a mesma função que o comando SAVE, e no caso de arquivos relacionados, não move os dados do arquivo principal para os arquivos relacionados secundários.

Sua sintaxe é:

```
SAVERECORD nome_arquivo
```

Obs.: Não use o comando Saverecord dentro das Macros ENTER e ENTERGROUP pois os dados ainda não foram movidos para o buffer de registro, portanto o comando Saverecord gravará somente registros em branco, logo, este comando deverá ser colocado após um Enterend ou um Entergroup, para que possa realizar sua função plenamente.

Exemplo:

```
SAVERECORD locacao
SAVERECORD cadfitas
SAVERECORD fornecedores
```

16.5 DELETE

Sua função é retirar do arquivo de dados selecionado o registro residente no buffer de registro do arquivo, seguindo-se incondicionalmente um comando Clear no arquivo, intrínseco ao comando DELETE. Tome cuidado especial para não deletar (excluir) um registro que seja alvo de um relacionamento, pois isto poderá estragar sua organização de dados, onde uma série de dados poderá perder parte de suas informações ou passará a não ter mais sentido (ex.: relação de pacientes sem o médico responsável, funcionário sem os respectivos cargos).

Sintaxe:

```
DELETE nome_arquivo
```

Exemplo:

```
DELETE locacao
DELETE cadfitas
DELETE ficha
```

16.6 FIND

Sua função é de procurar um registro em específico no arquivo de dados e colocá-lo no buffer de registro do arquivo, estabelecendo o status do buffer como verdadeiro e setar dois indicadores como verdadeiro ou falso.

Sintaxe:

1) FIND modo nome_arquivo BY INDEX.n

2) FIND modo arquivo.campo

No primeiro caso: *modo* pode ser um dos componentes de comando de comparação ou uma variável inteira (podendo ser também uma constante inteira) com conteúdo de acordo com a seguinte tabela:

Var_int	modo comparação
0	LT
1	LE
2	EQ
3	GE
4	GT

O nome_arquivo deve ser um nome de arquivo de dados, previamente aberto com o comando OPEN. O restante do comando "BY INDEX.n" indica por qual índice deve ser procurado um determinado registro. Lembre-se de que para uma operação de procura, por índices, ser executada todos os dados referentes àquele índice devem estar residentes no buffer de registro do arquivo.

No segundo caso ocorre uma operação de procura, não pelo índice, mas pelo conteúdo do campo do arquivo declarado.

Após realizada a procura nos arquivos de dados, o comando FIND seta dois indicadores que conterão o resultado da operação: se for realizada com sucesso, isto é, foi achado um registro com as especificações requeridas, os indicadores FOUND e FINDERR serão setados respectivamente como verdadeiro (TRUE) e falso (FALSE); caso a operação tenha sido um fracasso, isto é, nenhum registro foi encontrado, de acordo com as especificações do comando, os indicadores FOUND e FINDERR serão setados, respectivamente como falso (FALSE) e verdadeiro (TRUE).

Os indicadores FOUND e FINDERR são indicadores preexistentes, não sendo necessário defini-los. Como são setados automaticamente, pelo comando FIND ou uma opção de procura AUTOFIND, não se faz necessário setá-los antes das operações de procura. Isto não quer dizer que seus status não possam ser alterados dentro dos programas Dataflex.

Como uma convenção própria, neste livro o indicador FINDERR será "*substituído*" pelo NOT FOUND, isto é, pela negação do status do indicador FOUND. Esta convenção serve para uma melhor visualização do programa como também serve para um melhor entendimento dos procedimentos adotados dentro dele.

Exemplo:

```
/TELA
        CADASTRO DE CLIENTES DA VIDEO LOCADORA

CODIGO : <____. >

NOME : < _____ >
ENDERECO : _____ No: ____.COMPL: _____
CIDADE : _____ BAIRRO : _____
ESTADO : __ CEP : ____
RG : _____ DATA-NASCIMENTO : __/__/__

        ASSOCIADO1 : _____
        ASSOCIADO2 : _____
        ASSOCIADO3 : _____

use os campos com < > para processos de procura de dados.
```

```
/*
STRING tecla 1
OPEN cadcli
INICIO:                          // loop principal
 ENTERGROUP                      // inicio da Macro
    CLEAR cadcli                 // limpando buffer de registro
    clearform tela.1 thru tela.14 // limpa as janelas de tela.1 a tela.14
    LOOP:                        // loop secundario
     AUTOPAGE tela
       ENTRY cadcli.codigo {required}  // entrada obrigatoria
       FIND cadcli BY INDEX.1    // procura no arquivo cadcli pelo
       [ FOUND ] begin           // campo codigo
          gotoxy 24 0
          show " Este cliente ja existe !!!"
          inkey tecla            // pega uma resposta do teclado e guarda
          goto loop              // na variavel tecla
          end
    [ NOT FOUND ] begin     // inicia cadastramento se nao existir
                // nenhum outro com o mesmo codigo
            ENTRY cadcli.nome          {capslock}
            ENTRY cadcli.endereco      {capslock}
            ENTRY cadcli.numero
            ENTRY cadcli.complemento
            ENTRY cadcli.cidade        {capslock}
            ENTRY cadcli.bairro        {capslock}
            ENTRY cadcli.estado        {capslock}
            ENTRY cadcli.cep
            ENTRY cadcli.rg
            ENTRY cadcli.data_nascimento
            ENTRY cadcli.associado1    {capslock}
            ENTRY cadcli.associado2    {capslock}
            ENTRY cadcli.associado3    {capslock}
            end
 ENDGROUP
```

```
   gosub confirma// rotina para confirmar os dados digitados
   SAVE cadcli// salva os dados digitados no arquivo de dados cadcli
   goto inicio// volta para loop principal


CONFIRMA:
      clearxy 24 0 // limpa a linha da posicao declarada ate o fim da mesma
      gotoxy 24 0 // leva o cursor ate a posicao declarada
      show "Confirma (S/N) "          // imprime mensagem
      inkey tecla                     // pega resposta
      if not tecla in "SN" goto confirma // se tecla nao estiver dentro da
      if tecla eq "N" goto loop       // string declarada, repete a rotina
RETURN

KEYPROC KEY.ESCAPE:
      clearxy 24 0
      gotoxy 24 0
      show " deseja deixar este programa ? (S/N) "
      inkey tecla
      if tecla in "Ss" ABORT // se desejamos sair da aplicacao o
      clearxy 24 0     // comando ABORT ira terminar o programa
                       // para o sistema operacional ou um menu
        entagain       // nao deixa que se passe para a proxima janela
RETURN

KEYPROC KEY.CLEAR:
      clearform tela1 thru tela.14 // limpa as janelas declaradas se for
RETURN                             // teclada a Flex-Key limpa F9
```

PARTE IV
SELECIONANDO E OBTENDO INFORMAÇÕES DO BANCO DE DADOS

Veremos nos dois próximos capítulos como retirar e classificar as informações de um banco de dados, como também usar as ferramentas que Dataflex nos fornece, para tirar o melhor proveito destes dados no formato de relatórios e aplicações que gerem relatórios. Aprenderemos, no final,. como criar aplicações voltadas à emissão de relatórios a partir de um arquivo de dados.

PARTE IV
SELECIONANDO E OBTENDO INFORMAÇÕES DO BANCO DE DADOS

Veremos nos dois próximos capítulos como retirar e classificar as informações de um banco de dados, como também usar as ferramentas que Datafiex nos fornece, para tirar o melhor proveito destes dados no formato de relatórios e aplicações que gerem relatórios. Aprenderemos, no final, como criar aplicações voltadas à emissão de relatórios a partir de um arquivo de dados.

CAPÍTULO 17
USANDO O QUERY

QUERY é um módulo que possibilita a você extrair rapidamente e facilmente informações dos arquivos de dados em forma de relatórios. Estes relatórios poderão ser armazenados em discos, impressos ou ainda poderá ser criado um programa-fonte, que gere este tipo de relatório, sempre que você desejar.

Todos os arquivos de dados estão disponíveis para consultas. Todos poderão sofrer seleções, dos dados a serem impressos, sem que isto cause algum dano à confiabilidade do arquivo de dados.

QUERY permite dez condições de seleções de dados para os relatórios gerados. Campos numéricos poderão ser totalizados.

As aplicações geradas pelo QUERY são baseadas na Macro REPORT, tema do próximo capítulo.

Antes de podermos executar e usar adequadamente o QUERY, devemos ter uma base de dados adequada ao seu uso. Durante a explicação do uso do FILEDEF mostramos como seriam, basicamente, as estruturas dos arquivos que compõem um Sistema de Locadoras de Fitas de Vídeo Cassete. Naquele capítulo falamos apenas a respeito da estrutura do arquivo e como esta poderia ser usada para gerar uma aplicação de entrada, consulta de dados usando o utilitário ENTERDEF. Como vimos, a

aplicação gerada não tem aquilo que podemos chamar de uma beleza estética, porém servirá aos nossos propósitos com algumas modificações.

Segue-se a listagem dos programas-fontes gerados pelo ENTERDEF modificados, referentes aos arquivos de clientes (CADCLI.FRM), de fitas (CADFITAS.FRM) e de fornecedores (FORNECED.FRM) e o arquivo TIPOS, que conterá os tipos de filmes da locadora (aventura, comédia, etc).

/TELA Programa: CADCLI.FRM

CADASTRO DE CLIENTES DA VIDEO LOCADORA

CODIGO : < ____. >

NOME : < _____ >

ENDERECO : _____ No: ____. COMPL: _____

CIDADE : _____ BAIRRO : _____

ESTADO : _ CEP : ____

RG : _____ DATA-NASCIMENTO : __/__/__

ASSOCIADO1 : _____

ASSOCIADO2 : _____

ASSOCIADO3 : _____

use os campos com < > para processos de procura de dados.

```
/*                          // indica final das imagens (telas)
OPEN CADCLI                 // abre arquivo
ENTER CADCLI                // macro ENTER
    AUTOPAGE TELA // enderecamento automatico das janelas
    ENTRY CADCLI.CODIGO              {autofind}
    ENTRY CADCLI.NOME                {capslock,autofind}
    ENTRY CADCLI.ENDERECO            {capslock}
    ENTRY CADCLI.NUMERO
    ENTRY CADCLI.COMPLEMENTO         {capslock}
    ENTRY CADCLI.CIDADE              {capslock}
    ENTRY CADCLI.BAIRRO              {capslock}
    ENTRY CADCLI.ESTADO              {capslock}
    ENTRY CADCLI.CEP
    ENTRY CADCLI.RG
    ENTRY CADCLI.DATA_NASCIMENTO
    ENTRY CADCLI.ASSOCIADO1          {capslock}
    ENTRY CADCLI.ASSOCIADO2          {capslock}
    ENTRY CADCLI.ASSOCIADO3          {capslock}
    RETURN                  // final da secao ENTRY
ENTEREND                    // finaliza macro ENTER
ABORT                       // abandona esta configuracao
/TELA      Programa: FORNECED.FRM

        CADASTRO DOS FORNECEDORES DE FITAS

CODIGO : < ____. >
NOME : < _____ >
CGC : _____
ENDERECO : _____ No: ____. COMPL: _____
CIDADE : _____ BAIRRO : _____
ESTADO : __     CEP : ____

TELEFONE : (___) _____ CONTATO : _____

   use os campos com < > para processos de procura de dados.
```

```
/*                  // indica final das imagens (telas)
OPEN  FORNECEDOR            // abre arquivo
ENTER FORNECEDOR            // macro ENTER
      AUTOPAGE TELA  // enderecamento automatico das janelas
      ENTRY FORNECEDOR.CODIGO          {autofind}
      ENTRY FORNECEDOR.NOME            {capslock,autofind}
      ENTRY FORNECEDOR.CGC
      ENTRY FORNECEDOR.ENDERECO        {capslock}
      ENTRY FORNECEDOR.NUMERO
      ENTRY FORNECEDOR.COMPLEMENTO     {capslock}
      ENTRY FORNECEDOR.CIDADE          {capslock}
      ENTRY FORNECEDOR.BAIRRO          {capslock}
      ENTRY FORNECEDOR.ESTADO          {capslock}
      ENTRY FORNECEDOR.CEP
      ENTRY FORNECEDOR.DDD
      ENTRY FORNECEDOR.FONE
      ENTRY FORNECEDOR.CONTATO         {capslock}
      RETURN      // final da secao ENTRY
ENTEREND          // finaliza macro ENTER
ABORT             // abandona esta configuracao
```

Antes de mostrarmos como ficaria a aplicação de cadastro de fitas, devemos conhecer tanto a estrutura, o programa que foi criado a partir desta e a relação dos dados.

```
        FILE DEFINITION - ESTRUTURA ARQ. #8
*****************************************************
        NOME BASE DO ARQUIVO      = TIPOS
        NOME PARA USUARIO         = TIPOS
        NOME PARA DATAFLEX        = TIPOS
*****************************************************
        TAMANHO REGST = 512 (USADO = 13)
        NUMERO MAXIMO DE REGS = 11 (USADO = 0)
        ESPACO EXCLUIDO E REUTL
        MULTI-USER RE-READ ATIVO
*****************************************************
```

FIELD NMBR	FIELD OFFSET	FIELD LEN	FIELD TSPE	DEC PTS	MAIN INDEX	RELATES--TO FILE	FIELD	
1	1	1	NUMERIC	0	1	0	0	CODIGO
2	2	12	ASCII		0	0	0	NOME

INDEX 1: FIELD SEGMENTS: <1>

```
/TELA        Programa:TIPOS.FRM

     REGISTRO No.    <_____.>
     CODIGO          |_.|
     NOME            |__________|
/*                   // indica final das imagens (telas)
OPEN TIPOS           // abre arquivo
                     //
ENTER TIPOS          // macro ENTER

     AUTOPAGE TELA   // enderecamento automatico das janelas
ENTRY TIPOS.RECNUM {AUTOFIND} // se existe, mostra
     ENTRY TIPOS.CODIGO
     ENTRY TIPOS.NOME
     RETURN          // final da secao ENTRY
ENTEREND             // finaliza macro ENTER
ABORT                // abandona esta configuracao
```

Dados do arquivo TIPOS usado como consulta para o cadastro de fitas

Listagem dos campos CODIGO e NOME

1 AVENTURA	2 COMEDIA	3 DOCUMENTARIO	4 DRAMA
5 EROTICO	6 FICCAO	7 INFANTIL	8 MUSICAL
9 POLICIAL	10 ROMANCE	11 SUSPENSE	

```
/TELA       Programa: CADFITAS.FRM

            CADASTRO DAS FITAS

      CODIGO : < ____. >

      TITULO : < _____________________________ >

      TIPO : < _ > _____________

      FORNECEDOR : < ___. > - _____________________________

      PRECO PAGO : _________.__

      DATA DA COMPRA : __/__/__

use os campos com  < >  para processos de procura de dados.
/*                        // indica final das imagens (telas)
OPEN  CADFITAS                    // abre arquivo secundario
open  fornecedor                  // abre arquivo principal
open tipos          // abre arquivo terciario
ENTER CADFITAS tipos fornecedor // macro ENTER
      AUTOPAGE TELA         // enderecamento automatico das janelas
      ENTRY CADFITAS.CODIGO  {autofind}
      ENTRY CADFITAS.TITULO  {autofind,capslock}
      entry tipos.codigo     {autofind}// procura no arq terciario
      entry tipos.nome             {displayonly} // imprime campo do arq
      entry fornecedor.codigo      {autofind} // procura no arq secundario
      entry fornecedor.nome        {displayonly} // imprime campo do arq
      ENTRY CADFITAS.@PRECO_PAGO
      ENTRY CADFITAS.DATA_COMPRA {range = 01/01/70,31/12/99}
      move tipos.codigo to cadfitas.tipo       // move dados p/ buffer
      move fornecedor.codigo to cadfitas.fornecedor // move dados p/ buffer
      RETURN          // final da secao ENTRY
ENTEREND              // finaliza macro ENTER
ABORT                 // abandona esta configuracao
```

Após as entradas de dados nos arquivos anteriormente citados podemos finalmente aprender a usar e desfrutar deste módulo Dataflex.

Para ter acesso ao QUERY pelo Sistema de Menus devemos escolher a opção: Query Banco de Dados ou através do sistema operacional digitar QUERY seguido do número do arquivo que será usado. Este número de arquivo não é obrigatório, bastando digitar QUERY seguido de um ENTER. A seguinte tela irá aparecer:

DataFlex QUERY
BANCO DE DADOS DISPONIVEIS
PAG. 1 DE 1

Estes sao os bancos de dados disponiveis. "Aponte e Atire" naquele que voce deseja pesquisar. Pressione <HELP> para assistencia

[CADTEL]	CADFITAS
FORNECEDORES	CADCLI
LOCACAO	FICHA
TIPOS	

Aqui podemos selecionar o arquivo que será alvo do uso do QUERY. Isto não quer dizer que os outros arquivos não estarão disponíveis para o uso, porém isto veremos conforme formos aprendendo a usá-lo.

As seguintes telas de HELP são disponíveis, com algumas modificações para um melhor entendimento:

BANCO DE DADOS DISPONIVEIS HELP Tela 1

Os nomes dos bancos de dados disponiveis sao mostrados nesta tela. Estes nao sao os nomes verdadeiros, eles sao nomes ilustrativos de banco de dados definidos para operacao com Dataflex. Selecionando um deles, o acesso aos outros bancos de dados relacionados será automático.

Se voce nao encontrar o nome do banco de dados que voce esta procurando, estas poderao ser algumas das razoes:

a. O banco de dados nao esta no disquete, no drive ou no "pathname" corrente. Mude o disquete, o nome logico do drive ou o "pathname".

b. Em sistemas multiusuario, voce pode estar em area de outro usuário. Procure saber qual a area que o seu banco de dados esta.

c. O banco de dados que voce procura esta "inativo". Se este for o caso encontre a pessoa que fez isto e saiba por que, e peca a ela que devolva o status "ativo" de seu banco de dados para voce fazer o QUERY.

DATAFLEX QUERY - HELP GERAL Tela 2

Legendas entre < > indicam o uso de Flex-Keys (verifique nos capitulos anteriores do livro para verificar a acao de cada Flex-Key).

ACAO	DESCRICAO
APONTE	Use setas indicadoras para envolver os colchetes "[ ]" na opcao desejada.
ATIRE	Pressione ENTER para selecionar a opcao desejada.
DESFAZ	Para cancelar escolha, "Aponte" para o campo e pressione a Flex-Key Delete (F6).
VA EM FRENTE	Para finalizar opcoes, pressione a Flex-Key Save (F10).
PROXIMO	Para examinar a proxima tela de banco de dados ou campos, pressione a Flex-Key Next Record (Pg Dn).
ANTERIOR	Para voltar uma tela, pressione a Flex-Key Previous Record (Pg Up).
RELACAO	Para ver o relacionamento dos arquivos do banco de dados, pressione a Flex-Key Super Find (F3).
SAIDA	Para recomecar ou sair do QUERY, pressione a Flex-Key ESC.

Escolha, então, um dos arquivos disponíveis da relação acima. Poderemos escolher o cadastro de clientes por ser de fácil entendimento. Qualquer um destes arquivos poderia ser escolhido, mas por exemplo o arquivo de fitas é um tanto complicado para se explicar o uso do QUERY. O arquivo de fornecedores fica no mesmo nível do arquivo de clientes, mas não apresenta as mesmas características deste.

Aponte e Atire no CADCLI. A seguinte tela aparecerá:

A analise de (T)odos os X regs do CADCLI
sao possiveis, como tambem (S)elecionar
apenas uma serie deles. Aponte e Atire na opcao desejada.

[S] (T)

Para esta tela a imagem de Help disponível é a imagem 2 mostrada acima. Neste ponto podemos escolher o que fazer: consultar todos os registros (X registros) do arquivo selecionado (CADCLI) ou consultar estes mesmos X registros, só que através de uma seleção, uma filtragem destes dados. Para início devemos escolher a opção Todos os X registros. Aponte e Atire "T". Na seqüência a tela a seguir aparecerá:

Dataflex QUERY
SEQUENCIA SAIDA
CADCLI

QUERY pode mostrar registros classificados por numero do registro ou atraves dos indices mostrados abaixo. "Aponte e Atire" na sequencia desejada. (Para assistencia <HELP>.)

Classificado por:

[RECORD NUMBER]
CODIGO
NOME
DATA_NASCIMENTO

A seguinte tela de HELP é disponível :

SEQUENCIA DE SAIDA

Os registros selecionados podem ser listados na sequencia em que seu banco de dados foi indexado. Os indices existentes sao listados nesta tela, como tambem o numero do registro. "APONTE E ATIRE" na opcao desejada.

Tome cuidado quando selecionar indices com a palava "(BATCH)". Esses indices nao sao automaticamente mantidos pelo Dataflex. Todos os registros criados ou interados desde a ultima atualizacao do indice, nao foram atualizados (para atualiza-los utilize o utilitario REINDEX para a atualizacao destes). Sabendo que o indice foi atualizado com as ultimas alteracoes e estando confiante disto, va em frente.

Você deve escolher por qual destas opções deseja fazer a consulta aos dados do arquivo CADCLI. Note que a primeira opção é a procura seqüencial por RECNUM, isto é, pela seqüência com que foram gravados os dados. As outras opções referem-se à procura seqüencial pelos índices do arquivo. Podemos escolher qualquer um destes índices, por exemplo o código do cliente. Aponte e Atire no índice CÓDIGO.

A seguinte tela será mostrada:

```
Dataflex QUERY
    CAMPOS
CADCLI                                  Pag. 1 de 1

Voce pode escolher os campos a serem mostrados de acordo com a selecao
    feita. "Aponte e Atire" nos campos desejados.
        (Para assistencia, <HELP>).

[   RECORD #  ]     CODIGO          NOME
    ENDERECO        NUMERO          COMPLEMENTO
    CIDADE          BAIRRO          ESTADO
    CEP             RG              DATA_NASCIMENTO
    ASSOCIADO1      ASSOCIADO2      ASSOCIADO3
```

Use : + para Totalizar e Mostrar
< para Mostrar e Iniciar uma Nova Linha

A largura da linha já é de : N caracteres

Pressione ENTER para Atirar, F6 Delete para refazer e F10 Save para finalizar.

Aqui você tem a opçãq de escolher quais campos serão mostrados no relatório que será gerado a partir dos dados do arquivo de dados. A seqüência com que forem escolhidos os campos será a mesma com que serão impressos no relatório.

Podemos também totalizar campos numéricos, isto é, os campos numéricos selecionados terão seus conteúdos acumulados e ao final do relatório será mostrado o resultado do somatório.

Conforme forem selecionados os campos, será atualizado o número de caracteres da linha de impressão.

A seguinte tela de HELP é disponível:

CAMPOS DISPONIVEIS

Nesta tela, voce "Aponta e Atira" nos campos cujos registros serao mostrados pelo Query. A ordem que voce "Atirar" sera a ordem que os campos irao aparecer. Voce pode escolher um campo mais de uma vez "APONTANDO E ATIRANDO", mais de uma vez no mesmo campo.

Para imprimir um campo e saltar para a proxima linha "Atire" com um "<".

Para imprimir um campo e totaliza-lo "Atire" no campo usando um "T".

Se voce selecionou um campo normalmente e quer "quebrar" a linha, digitando um "<", voce nao imprimira o campo duas vezes.

Se voce selecionou um campo normalmente e quer "totalizar" a linha, digite um "T", e o campo nao sera impresso duas vezes.

Se o banco de dados que voce selecionou tem muitos campos e eles nao cabem em uma tela, pressione a Flex-Key Next Record (Pg Dn) para ve-los. Para voltar a tela anterior basta pressionar (Pg Up) Previous Record. Note que se voce tiver arquivos relacionados ao arquivo selecionado, voce podera acessar seus campos pressionando <REGISTRO POSTERIOR> (Pg Dn) ou pressione (F3) <PROCURA REGISTRO> e selecionando o arquivo desejado.

Poderemos selecionar os campos CODIGO, NOME e DATA de NASCIMENTO.

	RECORD #	*CODIGO	*NOME
	ENDERECO	NUMERO	COMPLEMENTO
	CIDADE	BAIRRO	ESTADO
	CEP	RG	*DATA_NASCIMENTO
[	ASSOCIADO1]	ASSOCIADO2	ASSOCIADO3

A largura da linha já é de : 55 caracteres

Tecle a Flex-Key F10 Save se estiver tudo correto; caso contrário. Aponte o campo errado e tecle F6 Delete para deixar o campo como não-selecionado, Aponte o campo correto e Atire. Após a teclagem da Flex-Key F10 será pedido, na última linha a digitação de um cabeçalho para o relatório que será gerado:

Digite a descrição do cabeçalho para este QUERY: [QUERY TESTE - IMPRESSAO DOS CAMPOS : CODIGO, NOME E DATA DE NASCIMENTO]

Digite a largura do formulario se for maior que 79 caracteres : [ ]

Após a digitação dos dados pedidos acima (não precisam ser necessariamente preenchidos), a seguinte tela será mostrada:

```
                    Dataflex QUERY
                    COMANDOS QUERY
    CADCLI

Cada secao do QUERY pode ser usada para definir o tipo de informacao
    que o QUERY precisa preparar. As secoes podem ser usadas para
    alterar o criterio de selecao, ordem de saida, campos a serem
    mostrados, etc. "Aponte e Atire" nas secoes que voce deseja alterar.
    GERACAO ira gerar um programa para repetir este Query a
    qualquer hora. Para assistencia, <HELP>.

    (S)elecao    (O)rdem C(A)mpos    Cabe(C)alho    (L)arg
      [E]xecuta    (G)eracao    (R)ecomeca    (F)im

"QUERY TESTE - IMPRESSAO DOS CAMPOS : CODIGO, NOME E
DATA DE NASCIMENTO"
Impressao:
  CADCLI.CODIGO CADCLI.NOME CADCLI.DATA_NASCIMENTO
Ordem Saida: CODIGO
Em 79 colunas.
```

Nesta tela temos o menu principal de opções do QUERY. Poderemos entrar no item Seleção e selecionar os dados que serão mostrados; poderemos escolher uma nova ordem, um novo índice para a operação de procura e seleção dos dados; poderemos escolher e alterar a ordem e os campos a serem impressos, poderemos modificar a largura de impressão dos dados na impressora; poderemos executar o relatório com os dados até agora recolhidos; poderemos gerar uma aplicação que executará um relatório com os dados anteriormente selecionados; poderemos recomeçar o QUERY a partir do ponto da escolha do arquivo-alvo ou poderemos sair do módulo QUERY.

A opção selecionada estará sempre dentro de colchetes "[ ]", enquanto as outras estarão dentro de parênteses "( )".

Para continuação do aprendizado poderemos executar o relatório que até aqui foi configurado (cabeçalho, largura da linha de detalhe, campos a serem impressos e ordem de seleção).

A seguinte tela aparecerá:

Dataflex QUERY
FORMA DE SAIDA
CADCLI

Este Query pode ser armazenado nas formas mostradas abaixo,
"Aponte e Atire" na forma desejada.

[V]ideo Im(P)ressora (D)isco

Se você tiver uma impressora ligada ao seu computador, pode escolher qualquer uma das opções; caso contrário selecione ou Vídeo ou Arquivo. Nestas duas escolhas não haverá grandes diferenças a não ser o número de linhas que serão mostradas para cada folha de impressão: Vídeo 15 linhas e Arquivo 55 linhas.

A seguinte tela de HELP é disponível:

FORMAS DE SAIDA

Qualquer das opcoes abaixo, apos serem executadas, retornarao a esta output, opcao do QUERY, portanto voce podera: a) mostrar o resultado na tela para uma verificacao, em seguida imprimi-lo e finalmente gerar um programa-fonte para nao precisar repetir a operacao.

Aqui voce pode selecionar as formas de saida do Query.

(V)ideo ira mostrar a saida na tela.
(I)mpressora ira mostrar a saida na impressora.
(D)isco ira gravar seus dados em um arquivo ASCII (texto), o qual pode ser processado, anexado a um documento ou transferido para outro programa ou sistema.

(R)ecomeca abandona o query e permite que voce recomece tudo novamente.
(F)im abandona o QUERY inteiramente.

Após ter selecionado o dispositivo de saída (vídeo, disco ou impressora) o QUERY executará a aplicação como segue:

```
QUERY TESTE - IMPRESSAO DOS CAMPOS : CODIGO, NOME E
DATA DE NASCIMENTO

CODIGO NOME                          DATA NASCIMENTO

    1 JOSE RICARDO                   14/12/67
    2 ELIANA FERNANDES               12/03/62
    3 NILSON DOS SANTOS PEREIRA      31/01/72
    4 ROMERO DE MOURA SAMPAIO        24/12/68
    5 ANTONIO CARLOS MAGALHAES       23/09/52
    6 JOAO DA COSTA RIBEIRO          01/01/73
    7 CLAUDIA ROSEMBERG              01/10/43

    QUERY COMPLETO. 7 registros mostrados. Pressione ENTER.
```

Após ter pressionado ENTER o fluxo de execução retornará ao menu principal das opções do QUERY.

Poderíamos, então, fazer uma seleção dos dados que serão mostrados. Para isso devemos Apontar na opção Seleção e Atirar.

```
                    Dataflex QUERY
                 DEFINICAO DE SELECAO
CADCLI                                          Pag. 1 de 1

QUERY pode selecionar registros da base de dados de acordo com o
conteudo dos campos. "Aponte e Atire" no(s) campo(s) pelos quais a
selecao e desejada. (Para assistencia, <HELP>).
```

```
[   RECORD #  ]   *CODIGO          *NOME
    ENDERECO      NUMERO           COMPLEMENTO
    CIDADE        BAIRRO           ESTADO
    CEP           RG               *DATA_NASCIMENTO
    ASSOCIADO1    ASSOCIADO2       ASSOCIADO3

    = IGUAL A       >  MAIOR QUE                  <  MENOR QUE
    X DIFERENTE     }  MAIOR QUE OU IGUAL A       {  MENOR QUE OU IGUAL A

                                   Selecoes Disponiveis: 10
```

Pressione ENTER para Atirar, F6 Delete para refazer e F10 Save para finalizar.

Os itens indicados com asteriscos são aqueles que serão impressos na execução do relatório. O menu de opções de seleção abaixo dos campos do arquivo de dados são referentes aos tipos de seleção que podem ser realizados com estes campos.

Para esta tela de Seleção é fornecida a seguinte tela de HELP:

SELECAO - DEFINICAO DO PROCESSO DE SELECAO HELP

Selecao e o processo de fazer com que o QUERY selecione os registros no banco de dados para satisfazer sua selecao. "A Selecao" e composta de campos de arquivos de dados e a maneira com que podemos compara-los com valores.

Query ira guia-lo para que voce possa construir ate 10 selecoes.

Neste ponto do Query, voce deve APONTAR para o campo que voce quer fazer uma seleção. Existem HELPS específicos para determinar a maneira com que as informações de seu banco de dados sejam comparadas com os valores digitados e estão disponíveis depois que você seleciona o campo.

Depois que voce estabeleceu sua selecao pressione <GRAVA> F10, para continuar.

Se seu banco de dados tem mais campos do que e possivel mostrar em uma tela, pressione <REGISTRO POSTERIOR>, para que mais campos sejam mostrados. Se o banco de dados que voce selecionou esta relacionado com outros, seus campos tambem poderao ser selecionados. Para saber quais os arquivos de dados estao relacionados com este que voce selecionou pressione <PROCURA>.

Poderemos selecionar os clientes com data de nascimento maior que 01/01/60, para isso, Aponte no campo DATA_NASCIMENTO e Atire. Aparecerá uma mensagem para que você selecione o campo data de nascimento de acordo com as opções fornecidas. No nosso caso a opção é MAIOR QUE. Aponte e Atire nesta opção. Será perguntado, então, uma data "maior que" pela qual deve ser procurado e selecionado o campo DATA_NASCIMENTO, digite 01/01/60. Note que junto ao campo DATA_NASCIMENTO aparecerá um símbolo maior ">" indicando que aquele campo está selecionado. Tecle F10 e execute o relatório.

Obs.: Verifique os dados gerados com os dados anteriormente listados.

CODIGO	NOME	DATA NASCIMENTO
1	JOSE RICARDO	14/12/67
2	ELIANA FERNANDES	12/03/62
3	NILSON DOS SANTOS PEREIRA	31/01/72
4	ROMERO DE MOURA SAMPAIO	24/12/68
6	JOAO DA COSTA RIBEIRO	01/01/73

Vejamos agora o que acontece se mudarmos a ordem de procura de CODIGO para NOME. Aponte para a opção Ordem e Atire, modifique de código para nome, tecle F10 e execute o relatório.

CODIGO	NOME	DATA NASCIMENTO
2	ELIANA FERNANDES	12/03/62
6	JOAO DA COSTA RIBEIRO	01/01/73
1	JOSE RICARDO	14/12/67
3	NILSON DOS SANTOS PEREIRA	31/01/72
4	ROMERO DE MOURA SAMPAIO	24/12/68

Verifique que a ordem de apresentação dos registros é diferente apresentando-se em ordem alfabética por nome. Agora execute o relatório para o índice DATA de NASCIMENTO.

CODIGO	NOME	DATA NASCIMENTO
2	ELIANA FERNANDES	12/03/62
1	JOSE RICARDO	14/12/67
4	ROMERO DE MOURA SAMPAIO	24/12/68
3	NILSON DOS SANTOS PEREIRA	31/01/72
6	JOAO DA COSTA RIBEIRO	01/01/73

Poderíamos selecionar os dados para apenas aqueles clientes que nasceram na década de 60. Vá à opção de seleção e elimine a seleção já feita (teclando F6 no campo selecionado), crie uma nova seleção "maior gual" a 01/01/60 e crie uma nova seleção "menor igual" a 31/12/69, tecle F10 e execute.

CODIGO	NOME	DATA NASCIMENTO
2	ELIANA FERNANDES	12/03/62
1	JOSE RICARDO	14/12/67
4	ROMERO DE MOURA SAMPAIO	24/12/68

Já chega de ficarmos brincando com seleções de arquivos de dados, passemos agora para a opção Geração de uma aplicação: será mostrada a seguinte tela:

Gera QUERY formatado ou Programa-fonte?

[Q]uery ou (F)onte

A seguinte tela de HELP tem as explicações necessárias para um bom uso desta fase do QUERY:

GERANDO PROGRAMA-FONTE

O QUERY que voce acabou de fazer pode ser guardado de duas maneiras diferentes.

Ambas criarao um arquivo em disco com o nome dado por voce e a extensao padrao provida pelo QUERY.

Query cria um conjunto de instrucoes utilizaveis e modificaveis somente pelo QUERY. Esta opcao nao requer nenhum outro processo adicional e representa a unica maneira de armazenar relatorios para futuro uso, sem a necessidade de compilacao. Porem, a limitacao das coisas possiveis de serem feitas esta naquilo que o QUERY faz, bem como na lista de arquivos autorizados que é mostrada no inicio do query.

Fonte cria um programa-fonte em Dataflex, que pode ser usado e modificado da mesma maneira que um programa convencional em Dataflex e escrito. Este programa gerado deve ser compilado antes de ser usado. Note que uma vez o programa gerado o usuario podera altera-lo para usar todo o escopo da linguagem, bem como saber que tem em suas maos um programa que funciona, livre de erros e a partir dai implementa-lo de acordo com sua necessidade. Este programa tem a extensao padrao ".RPT".

Selecione Fonte para a geração da aplicação que irá gerar um idêntico a este que acabou de ser gerado e executado.

A seguinte tela de Help é disponível:

FORMAS DE SAIDA- (G)ERACAO

Esta opcao do Query grava um programa-fonte em Dataflex, que depois de compilado pode ser usado para gerar o mesmo resultado que o query o faria, sem ter de passar por todo o processo seletivo novamente. A compilacao é um passo necessario pois programas escritos em Dataflex nao podem ser executados sem antes serem compilados.

As quatro opcoes de formato oferecidas sao mostradas na proxima tela de help.

Editando o codigo fonte gerado por esta opcao, com seu processador de textos ou com o EDITOR de textos do Dataflex, voce pode adicionar funcoes e detalhes que nao podem ser providos pelo QUERY. As possibilidades de expansao sao tao grandes que seria impossivel para o QUERY coloca-las ou serem mostradas neste espaco. Sempre use o QUERY para gerar o codigo-fonte para todos os relatorios que voce necessitar e depois molde-o para sua necessidade.

Antes de compilar um programa gerado pelo QUERY voce pode muda-lo, adicionar novos comandos, etc. Isto pode ser feito com o Editor do Dataflex ou qualquer outro processador de textos. Para compilar um programa gerado pelo QUERY, use o compilador. Se voce nao tem o compilador e porque voce tem somente a licenca de RUNTIME do Dataflex. Para adquirir o compilador e os outros modulos de desenvolvimento, consulte seu representante ou a INTERCOMP.

Existe uma segunda tela de HELP para a explicação de uso e finalidade das outras opções de geração de programas-fontes:

FORMAS DE SAIDA - OPCAO DE FORMATACAO DE ARQUIVOS

QUERY oferece quatro opcoes de formato de saida em arquivo de disco:

(R)ELATORIO e (M)ALA DIRETA : tem o mesmo significado, respectivamente ao (R)elatorio e (M)ala Direta da opcao (I)mpressao, com a diferenca de que, neste caso, os dados serao jogados no disco em vez de na impressora.

(L)INHA (delim) : grava um arquivo ASCII com um campo em cada linha, ou seja, um <CR> <LF> (carriage return e um line feed) separando cada campo. Esta e uma forma que Dataflex pode usar para importar dados de outros sistemas. Muitos outros softwares podem importar dados da mesma maneira.

(V)IRGULA (delim) : grava um arquivo ASCII, mas com todos os campos de um registro organizado em uma linha. Campos ASCII sao marcados com (") e todos os outros campos sao separados por virgula (,). Esta forma de saida e compativel com muitas aplicacoes tais como BASIC, muitos processadores de textos e suas funcoes de "merge" como tambem planilhas eletronicas.

Para direcionar o relatorio de um programa gerado pelo Dataflex para um disco em vez da impressora, você podera colocar o nome do arquivo no comando que ira executar o programa.

Exemplo:

FLEX CLIREL MEUDADO.PRN

Selecione a opção Relatório, pois o que foi gerado no nosso caso foi um relatório e não uma mala direta. A seguinte questão será apresentada:

Digite o nome a ser dado (d:arquivo) ao programa
[RELCLI]

Seu novo programa RELCLI.RPT, esta pronto!
Pressione qualquer tecla !

Neste ponto foi gerado um programa-fonte para executar um relatório com os dados de formatação idêntico ao que foi gerado durante a execução do QUERY.

Temos aqui a listagem do programa-fonte gerado :

```
/HEADER                    // tela de cabecalho
    CODIGO NOME                    DATA NASCIMENTO

/BODY RESIDENT                     // tela da linha detalhe
_____. _____ __/__/____
/TOTAL                     // tela de totalizacao de registros impressos
REGS IMPRESSOS: _____.
/SELECTION                 // tela de selecao dos dados
Selecione o campo DATA NASCIMENTO maior que ou igual a __/__/____
Selecione o campo DATA NASCIMENTO menor que ou igual a__/__/____
/*                         // fim das telas
OPEN CADCLI INDEX.3        // abre arquivo de dados
OUTFILE                    // indica dispositivo de saida
      PAGE SELECTION       // imprime tela de selecao no video
      ACCEPT SELECTION.1 // aceita entrada do primeiro dado de selecao
      MOVE SELECTION.1 TO CADCLI.DATA_NASCIMENTO    //sai fora indice
      ACCEPT SELECTION.2 // aceita segundo dados para selecao

REPORT CADCLI BY INDEX.3 // inicializa a Macro REPORT
 INDICATE SELECT AS CADCLI.DATA_NASCIMENTO GE SELECTION.1 // seta
[NOT SELECT] RETURN                                        // indicador
 INDICATE SELECT AS CADCLI.DATA_NASCIMENTO LE SELECTION.2 // de
[NOT SELECT] RETURN END.OF.REPORT //SAI FORA INDICE        // selecao
```

```
SECTION HEADER                          // secao do cabecalho
     OUTPUT HEADER                      // imprime cabecalho
SECTION BODY                            // secao linha de detalhe
     PRINT CADCLI.CODIGO                // move os dados do arquivo
     PRINT CADCLI.NOME                  // para linha de detalhe, mas
     PRINT CADCLI.DATA_NASCIMENTO       // nao imprime
     OUTPUT BODY                        // impressao da linha de detalhe
SECTION TOTAL                           // secao de totalizacao
PRINT RECCOUNT TO TOTAL.1               // move variavel predefinida para
     OUTPUT TOTAL                       // campo de impressao. Impressao
REPORTEND                               // fim da Macro REPORT

FORMFEED                                // pula pagina
ABORT                                   // sai do programa
```

Não estranhe a diferença deste programa-fonte com os outros gerados anteriormente, pois este se utiliza de uma outra estrutura de programa, a Macro REPORT. A Macro Report será explicada no próximo capítulo.

Para sair do módulo QUERY basta Apontar para a opção Fim e teclar ENTER.

CAPÍTULO 18
MACRO REPORT

18.1 O QUE É MACRO REPORT

É um conjunto de rotinas direcionadas, basicamente, em emissão derelatórios e seleção de dados. A Macro REPORT pode ser usada uma única vez dentro de um programa ou aplicação Dataflex. Sua função geral é a emissão de relatórios, formatados ao gosto do usuário. É gerada normalmente durante o uso do QUERY ou do MAIL (um utilitário usado para a geração de programas de emissão de etiquetas). Isto não quer dizer que não possamos criar relatórios sem a Macro REPORT ou sem a ajuda destes utilitários, mas é recomendável, devido à inexistência de geração de erros, por parte destes utilitários.

18.2 A ESTRUTURA DA MACRO

A Macro Report é basicamente formada por três áreas distintas e individuais : a área pré-report, o comando Report e as seções da macro.

A área pré-report consta, basicamente, das imagens que farão parte dos cabeçalhos, linhas de detalhe, linhas de subtotais, linhas de totais e linhas de rodapé, consta das operações iniciais com os arquivos (abertura de arquivos, criação de variáveis de memória e outros processamentos que forem necessários). O comando Report inicializa Macro e indica por qual índice será gerado o relatório e quais serão os pontos de quebra do mesmo. As seções da Macro estão assim distribuídas:

- Seção HEADER
- Seção subHeader # (de 1 a 9)
- Seção BODY
- Seção Footer
- Seção subTotal # (de 1 a 9)
- Seção Total

Usaremos o exemplo criado pelo QUERY para uma exemplificação da Macro Report:

```
/HEADER

CODIGO NOME                    DATA NASCIMENTO

/BODY RESIDENT
_____.____________________________ __/__/____
/TOTAL
REGS IMPRESSOS: _______.
/SELECTION

Selecione o campo DATA NASCIMENTO maior que ou igual a __/__/____
Selecione o campo DATA NASCIMENTO menor que ou igual a __/__/____
/*
```

```
//
//  Esclarecimentos a respeito das imagens na Macro REPORT
//
```

As imagens HEADER, BODY, TOTAL e SELECTION são as imagens que serão usadas durante o uso do REPORT: a imagem Header será usada sempre como cabeçalho, isto é, será usada sempre que se iniciar uma nova página (por quebra ou início de uma nova); a imagem Body é a linha de detalhe do relatório (pode haver mais de uma linha de detalhe formada por diversas janelas e texto). Esta imagem será apresentada e impressa sempre que um registro passar pelo processo de seleção (a palavra RESIDENT informa ao compilador que esta imagem deverá ficar residente na memória, evitando assim que tenha de ser acessada sempre do disco, economizando tempo de acesso mas gastando espaço em memória); a imagem Total contém as janelas que serão totalizadas (estas janelas podem estar associadas a campos do arquivo de dados, a variáveis de memória que sofrem processos matemáticos durante a execução do programa ou campos que foram subtotalizados). A imagem Selection contém as janelas de entradas de dados para os parâmetros das seleções que serão realizadas.

```
//

OPEN CADCLI INDEX.3      // abertura do arquivo que sera alvo do relatorio
OUTFILE                  // indica o dispositivo de saida do relatorio:
                         // pode ser um nome de arquivo, "con:" indicando video,
                         // "lst:" ou nada, indicando impressora

    PAGE SELECTION                  // imprime no video a imagem de selecao
    ACCEPT SELECTION.1              // para entrada de parametro 1
    MOVE SELECTION.1 TO CADCLI.DATA_NASCIMENTO    //cai fora indice
    ACCEPT SELECTION.2      //pega entrada do parametro 2
REPORT CADCLI BY INDEX.3    // inicializa a Macro e indica o arquivo e o
                            // pelo qual vai ser gerado o relatorio
```

```
INDICATE SELECT AS CADCLI.DATA_NASCIMENTO GE SELECTION.1 // seta
[NOT SELECT] RETURN                    // indicador de selecao
  INDICATE SELECT AS CADCLI.DATA_NASCIMENTO LE SELECTION.2
[NOT SELECT] RETURN END.OF.REPORT //SAI FORA INDICE
SECTION HEADER                         // inicializa a secao de cabecalho
    OUTPUT HEADER                      // imprime cabecalho
SECTION BODY                           // inicializa secao linha de detalhe
    PRINT CADCLI.CODIGO                // move os conteudos dos campos para
    PRINT CADCLI.NOME                  // as janelas da linha de detalhe
    PRINT CADCLI.DATA_NASCIMENTO
    OUTPUT BODY                        // imprime a linha de detalhe
SECTION TOTAL                          // inicializa a secao de totalizacao
    PRINT RECCOUNT TO TOTAL.1          // move os dados (variavel predefinida,
    OUTPUT TOTAL                       // contador de numero de registros impressos)
                                       // impressao do total
REPORTEND                              // finaliza a Macro
FORMFEED                               // pula pagina
ABORT                                  // sai do programa
```

18.3 SEÇÕES DA MACRO REPORT

Como você pode perceber, para cada seção corresponde uma imagem com o mesmo nome. Podemos ter seções sem que tenhamos as imagens correspondentes. Neste caso, o comando OUTFILE não deve estar presente, pois não haverá imagem a ser impressa, o que ocasionaria um erro. Qualquer comando Dataflex pode estar presente nestas seções com exceção daqueles que venham de encontro à função da seção. Note que dentro destas seções foram usados os comandos PRINT. Este comando não tem como função a impressão dos dados (função atribuída ao comando OUTFILE), mas o de mover os dados para as janelas das imagens e atualizar os acumuladores de totais e subtotais.

Seção Header: é usada basicamente sempre que ocorre quebra de página. Nesta seção constam normalmente alguns dados que devem ser impressos somente em ocasiões especiais (número de página, data da emissão do relatório, nome da empresa, dados quanto ao relatório, etc).

Seções SubHeader e SubTotal: são em número de nove cada uma (os números possíveis de quebra). São usadas paralelamente um ao outro, isto é, ao subheader7 corresponde o subtotal7, ao subheader6 corresponde o subtotal6, e assim por diante. Estas seções delimitam os pontos de quebra (para o nível mais freqüente de quebra corresponde o subheader e subtotal de maior número e para o nível de quebra menos freqüente corresponde o subheader e subtotal de número mais alto). Todos os nomes das imagens destas seções podem vir com a "palavra" RESIDENT. Estas seções não precisam, necessariamente, de imagens a elas associadas, contanto que não sejam usados os comandos OUTFILE entre seus comandos. Um subtotal pode totalizar somente o subtotal anterior ou subtotalizar as janelas ou campos de arquivos (o subtotal3 subtotaliza o subtotal4 e assim por diante).

Seção Body: nesta seção constam os dados da linha de detalhe que serão impressos. Nesta seção encontramos o comando PRINT, cujas principais funções são: mover os dados dos arquivos para as janelas correspondentes na imagem Body e atualizar qualquer acumulador de subtotais ou totais que porventura existirem. O comando PRINT é equivalente, nas outras Macros, aos comandos Move e Display, os quais não atualizam os acumuladores de totais. Para a impressão o comando OUTFILE é obrigatório.

Seção Footer: é a seção encarregada da impressão de rodapés de páginas e que, normalmente, não apresentam nenhum tipo de processamento específico ou que acabe modificando os dados dos relatórios gerados. Esta seção é como as demais: pode ter sua imagem residente na memória, pode ter mais de uma linha de impressão, pode conter comandos PRINT ou apenas o comando OUTFILE ou comandos de controle.

Seção Total: esta seção é responsável pelas totalizações dos subtotais, se existirem ou, em caso contrário, responsável pelas totalizações de campos de arquivo ou janelas.

Não é necessário usarmos todas as seções em um relatório ou aplicação, somente aquelas que forem necessárias à sua perfeita execução.

O comando Report: inicializa a Macro REPORT e indica por qual índice será pesquisado o arquivo de dados e quais serão os campos de quebra e a ordem de pesquisa (ascendente ou descendente pelo índice). O comando REPORT pode ser usado uma única vez dentro da aplicação.

Sintaxe:

```
REPORT arquivo BY/DOWN index.n < BREAK arq.campo1 .. arq.campo n >
                < arquivo.recnum BREAK arq.campo1 .. arq.campo n >
```

No primeiro caso selecionamos o arquivo-alvo do report, escolhemos a ordem da pesquisa, se ascendente ou descendente, o índice pelo qual será pesquisado o arquivo e os campos que serão usados para comparação, ocasionando quebras sempre que seus conteúdos mudarem.

Obs.: Quando se desejar executar um relatório com quebras, devemos ter os campos implicados dentro de um índice multissegmento, isto é, a seqüência de quebras será realizada de acordo com os campos indexados no índice referente a este relatório.

No segundo caso não usaremos a ordenação de um índice para selecionarmos a seqüência de pesquisa, mas a ordem de entrada dos dados (ordem do número de registro do arquivo-alvo do relatório).

Vejamos agora como ficaria um esquema de uma aplicação que gera um relatório com uma quebra, um subtotal e uma totalização.

```
/header                 // imagem de cabecalho
...
/body                   // imagem da linha de detalhe
...
/total                  // imagem da totalizacao
...
/*                      // fim das imagens
open arq1
outfile
report arq1 by index.n break arq1.campo1 arq2.campo3    // report  arq1
                                                         //o arquivo pelo indice n quebrando
section header                                           // pelo campo1 e campo3
     outfile header                                      // imprime header
section body
     print arq1.campo2
     print arq1.campo3
     print arq1.campo5
     outfile body                                        // imprime linha de detalhe
section subtotal.1
     subtotal body.2                                     // subtotaliza imagem 2 do body

section total
     subtotal subtotal.1 to total.1                      // totaliza a subtotalizacao da
     outfile total                                       // imagem 2 do body e imprime
repotend                                                 // finaliza a Macro
formfeed
abort
```

De acordo com o Sistema de locadoras, poderíamos criar o Relatório de Faturamento. Devemos criar uma aplicação através do QUERY e em seguida fazer modificações no programa-fonte. O índice a ser usado deve ser DATA, isto é, o índice número dois (2), (quebra por dia de locação e subtotal por valor pago/dia) e ao final um total arrecadado no período. Não

devemos nos esquecer de que para que tenhamos um faturamento, devemos selecionar apenas as locações baixadas, isto é, locação.baixa igual a "S".

A seguir temos a tela do menu principal do QUERY sobre o relatório a ser gerado e o programa gerado sem as modificações:

LOCAÇÃO

Cada seção do QUERY pode ser usada para definir o tipo de informação que o QUERY precisa preparar. As seções podem ser usadas para alterar o critério de seleção, ordem de saída, campos a serem mostrados, etc.

"Aponte e Atire" nas seções que você deseja alterar.
GERAÇÃO irá gerar um programa para repetir este Query a qualquer hora.
Para assistência, <HELP>.

(S)elecao (O)rdem C(A)mpos Cabe(C)alho (L)arg
[E]xecuta (G)eracao (R)ecomeca (F)im

Impressao:
LOCACAO.DATA LOCACAO.NUM_FITAS_LOC LOCACAO.VALOR
Selecao:

Selecione o campo BAIXA casando S
Selecione o campo DATA maior que ou igual a 010185
Selecione o campo DATA menor que ou igual a 311288
Ordem Saida DATA
Em 79 colunas.

```
/HEADER
DATA        NUM FITAS LOC        VALOR

/BODY RESIDENT
__/__/____      ____. _________.__
/TOTAL
NUM FITAS LOC _________.__
VALOR         _________.__
REGS IMPRESSOS:     _______.
/SELECTION
Selecione o campo BAIXA casando _
Selecione o campo DATA maior que ou igual a  __/__/____
Selecione o campo DATA menor que ou igual a  __/__/____
/*
OPEN LOCACAO INDEX.2
OUTFILE
      PAGE SELECTION
      ACCEPT SELECTION.1
      ACCEPT SELECTION.2
      MOVE SELECTION.2 TO LOCACAO.DATA //SAI FORA INDICE
      ACCEPT SELECTION.3

REPORT LOCACAO BY INDEX.2
         INDICATE SELECT AS SELECTION.1 MATCH LOCACAO.BAIXA
[   SELECT] INDICATE SELECT AS LOCACAO.DATA GE SELECTION.2
[NOT SELECT] RETURN
         INDICATE SELECT AS LOCACAO.DATA LE SELECTION.3
[NOT SELECT] RETURN END.OF.REPORT      //SAI FORA INDICE
SECTION HEADER
      OUTPUT HEADER
SECTION BODY
      PRINT LOCACAO.DATA
      PRINT LOCACAO.NUM_FITAS_LOC
      PRINT LOCACAO.VALOR
      OUTPUT BODY
```

```
SECTION TOTAL
        SUBTOTAL BODY.2 TO TOTAL.1
        SUBTOTAL BODY.3 TO TOTAL.2
        PRINT RECCOUNT TO TOTAL.3
        OUTPUT TOTAL
REPORTEND
FORMFEED
ABORT
```

Passemos agora para as modificações a serem feitas:

- Mudar o Layout do relatório: mudar o HEADER. Acrescentar uma numeração de páginas, a data do sistema e imprimir as datas do período do relatório.
- Colocar instruções de quebra no comando REPORT, por locação.data.

- Criar uma imagem de subtotal para as totalizações de valor por data. Este subtotal deverá ser uma espécie de linha de detalhe, pois no nosso caso interessa imprimir a data somente quando ela é quebrada. Para isto devemos guardar esta data numa variável, dentro da seção BODY, pois na seção SUBTOTAL ela será perdida, portanto nesta seção devemos imprimir a variável data e não o campo data.
- Modificar o Body: Ele servirá apenas para as totalizações do número de fitas locadas e do valor arrecadado. Nela deve-se mover a data da locação do arquivo para uma variável, que guardará a data para ser exibida assim que ela sofrer quebra (a data lida será diferente daquela à qual os dados estão associados).

Antes de modificarmos o programa gerado, devemos ter esclarecidos alguns pontos cruciais para um bom aproveitamento desta macro. Estes pontos referem-se ao fato de podermos configurar tanto a saída do relatório como o formato final do mesmo (escolha do número de linhas por página, escolha do tamanho físico da página e a escolha dos tipos de formatação de janelas numéricas ou ASCII).

18.3.1 Variáveis Predefinidas

As variáveis predefinidas do Dataflex são usadas basicamente para controle de paginação dos relatórios por ela gerados ou para a a formatação das saídas das aplicações. São elas:

LINECOUNT: conta o número de linhas impressas pela Macro Report por página. É uma variável inteira e pode conter qualquer número inteiro. É raramente usada, a não ser que se deseje saber o número de linhas restantes em uma página.

PAGEFEED: mantém o controle do dispositivo de saída (vídeo, impressora ou arquivo-texto). Esta variável é inteira e pode assumir qualquer um destes valores (-1, 0, N):

-1: pede confirmação para que possamos passar para a próxima página de impressão (quando usando o monitor de vídeo como saída, recomenda-se principalmente para permitir uma boa visualização dos dados impressos).

0 : usa o padrão de impressão normal, isto é, ao se atingir o número máximo de linhas de um página, uma folha ou uma página será avançada.

N : é um número qualquer, inteiro que definirá o tamanho físico do papel de impressão. Deve ser maior que zero.

PAGEEND: estabelece o número de linhas que serão impressas por página. Também é uma variável inteira e pode assumir o valor 0, quando se deseja um relatório contínuo ou um valor maior do que zero para se estabelecer um número máximo de linhas/página. Pageend menos Linecount nos dá o número de linhas que ainda faltam a serem impressas em uma página.

PAGECOUNT: armazena o número da página corrente no relatório.

Juntamente com estas variáveis existe um comando que é responsável pelo avanço da folha ou página de impressão de um relatório:

FORMFEED: provoca um avanço de página na impressora ou limpa a tela do monitor de vídeo.

18.3.2 Tipos de formatação de janelas ASCII e numéricas

Os tipos de formatação aqui apresentados devem ser usados ou pelo comando PRINT ou então devem ser colocados diretamente nas janelas-alvo destas formatações.

Para o comando PRINT temos as seguintes opções de formatação de janelas:

SUPRESS : elimina os espaços em branco na impressão apenas em janelas ASCII. Não tem argumentos.

FILL="?" : preenche os espaços em branco restantes de uma janela após o dado ter sido impresso ou digitado com o símbolo que está entre as aspas da opção.

SIGNRT : coloca um hífen à direita de valores negativos.

FLOAT$: coloca o caractere equivalente àquele da configuração, à esquerda de valores numéricos (normalmente "$").

CAPSLOCK : imprime os dados do arquivo em letras maiúsculas.

POINTS= X : especifica o número de casas decimais à direita do ponto decimal em uma janela numérica (pode ser de 0 a 4).

Para as janelas especificamente temos estes tipos de formatação:

-Uso de vírgula marcando milhar: ___,___,___.__ Esta janela, de acordo com a instalação de seu Dataflex, seria preenchida da seguinte maneira : *999.999.999,99* . Note que o ponto decimal foi substituído por uma vírgula e que as vírgulas de milhar foram substituídas por pontos de milhar.

-Preenche campo à esquerda com zeros: _____0__.__ Esta janela ao ser preenchida parcialmente apresentará as casas à esquerda do último número cheias de zeros: 00000012345.67

-Preenche os espaços à esquerda com brancos: _____Z__.__

A seguir a listagem de como ficaria este relatório de faturamento quebrando por data de locação, com a seleção do período de emissão do relatório.

```
/HEADER     RELATORIO DE FATURAMENTO/DIA

            RELATORIO DE FATURAMENTO                    Pag: __.

      Video Locadora - Faturamento por dia              Data : __/__/__

            Periodo de : __/__/__ a __/__/__

DATA NUMERO DE FITAS LOCADAS/DIA VALOR ARRECADADO/DIA
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
/BODY RESIDENT
      _____.      _____.__
/SUBTOTAL1
      __/__/__    _____.      _____.__
-----------------------------------------------------------------------------------------
TOTAL
```

= = = =

Caso tenha escolhido a opcao IMPRESSORA : o relatorio sera gerado com 55 linhas/pagina, automaticamente.

Caso tenha escohido a opcao ARQUIVO : sera pedido o nome do arquivo em letras maiusculas com tamanho maximo de 8 letras ou numeros, tendo a terminacao igual a ".REL".

```
/*
STRING    tecla 1 saida 12
INTEGER   dias pagina
DATE      data_sys data_locacao

SYSDATE   data_sys

MOVE ""TO saida
MOVE 0 TO pagina
MOVE 0 TO dias

OPEN LOCACAO INDEX.2


    PAGE SELECTION
    blankform selection.4
    ACCEPT SELECTION.1
    MOVE SELECTION.1 TO LOCACAO.DATA
    ACCEPT SELECTION.2

    ACCEPT selection.3 {capslock,check = "VIA"}
// verificacao do tipo de saida do relatorio

    IF selection.3 EQ "V" begin                    // VIDEO
          MOVE    -1      TO pagefeed
          MOVE    20      TO pageend
          MOVE    "CON:"  TO saida
       end
```

```
    IF selection.3 EQ "I" begin                    // IMPRESSORA
            MOVE  0 TO pagefeed
            MOVE  55 TO pageend
            MOVE  "" TO saida
          end

    IF selection.3 EQ "A" begin                    // ARQUIVO
            ACCEPT selection.3  {capslock}
            MOVE  55 TO pageend
            MOVE  0 TO pagefeed
            APPEND saida selection.3 ".REL"
          end
show "saida para " selection.3 saida
inkey tecla

OUTFILE saida          // "CON:" : Seta saida para video
                       // Nome de arquivo : saida para um relatorio completo em
                       //                   disco
                       //Impressora : nao precisa de parametros

    clearscreen

    REPORT locacao BY INDEX.2 BREAK locacao.data

    INDICATE SELECT AS locacao.baixa EQ "S"
[   SELECT] INDICATE SELECT AS LOCACAO.DATA GE SELECTION.1
[NOT SELECT] RETURN
    INDICATE SELECT AS LOCACAO.DATA LE SELECTION.2
[NOT SELECT] RETURN END.OF.REPORT

SECTION HEADER
    INCREMENT pagina
    PRINT pagina      TO header.1
    PRINT data_sys    TO header.2
    PRINT selection.1 TO header.3
    PRINT selection.2 TO header.4
    OUTPUT HEADER
```

```
SECTION BODY
    PRINT LOCACAO.NUM_FITAS_LOC
    PRINT LOCACAO.VALOR
    MOVE locacao.data TO data_locacao

SECTION SUBTOTAL1
    PRINT data_locacao      TO subtotal1.1
    SUBTOTAL BODY.1         TO subtotal1.2
    SUBTOTAL BODY.2         TO subtotal1.3
    INCREMENT dias
    OUTPUT SUBTOTAL1

SECTION TOTAL
    SUBTOTAL subtotal1.2    TO total.1
    SUBTOTAL subtotal1.3    TO total.2
    PRINT dias              TO total.3
    OUTPUT TOTAL

REPORTEND

FORMFEED
ABORT
```

Verifique que foi incluso um procedimento de escolha do direcionamento da saída do relatório. Este procedimento poderia ter sido feito de outros modos, porém, este parece ser um dos mais simples.

18.4 SELEÇÕES COM O REPORT

Como vimos acima, podemos selecionar dados de arquivos para uma emissão de relatórios ou uma seleção destes dados para outros fins. Basicamente as seleções usando o REPORT centralizam-se em dois pontos principais: o indicador SELECT e o comando INDICATE. Como vimos

anteriormente, o comando INDICATE serve para atribuir um valor verdadeiro ou falso a um indicador, de acordo com algumas condições. Este comando é usado para as seleções dos dados do arquivo-alvo, do relatório, para filtrar aqueles que interessam ou não ao processamento que se dará durante a execução do mesmo. O indicador SELECT servirá para checagem de novas condições de seleção. A estrutura de uma seleção de dados dentro do REPORT é a seguinte:

```
        INDICATE SELECT AS CADCLI.CODIGO GT SELECTION.1
[       SELECT] INDICATE SELECT AS SELECTION.2 MATCH CADCLI.CIDADE
```

Neste caso será selecionado o cliente com código maior ao indicado na janela de entrada de dados da imagem de seleção. Se for achado um registro que satisfaça a primeira condição, será checado se este satisfaz a segunda condição (cidade igual à indicada na segunda janela da imagem de seleção). Caso satisfaça este registro, será processado e a Macro REPORT passará à leitura do próximo registro do arquivo de dados.

PARTE V
COMANDOS AVANÇADOS E PROGRAMAÇÃO

Nesta parte do livro veremos alguns comandos parecidos com aqueles já apresentados, porém, estes serão usados em casos onde o processamento dos arquivos necessita de maiores cuidados e recursos de programação como, por exemplo, pesquisas seqüenciais por índices, seleções de dados, consulta de dados em arquivos relacionados, importação e exportação de parâmetros e dados.

PARTE V
COMANDOS AVANÇADOS E PROGRAMAÇÃO

Nesta parte do livro veremos alguns comandos parecidos com aqueles já apresentados, porém, estes serão usados em casos onde o processamento dos arquivos necessita de maiores cuidados e recursos de programação como, por exemplo, pesquisas sequenciais por índices, seleções de dados, consulta de dados em arquivos relacionados, importação e exportação de parâmetros e dados.

CAPÍTULO 19
COMANDOS DE MANIPULAÇÃO DE ARQUIVOS

Os comandos mostrados aqui têm certa semelhança com o comando FIND, anteriormente apresentado, porém com a diferença de que estes comandos possuem outros recursos além de efetuar uma simples procura em um arquivo de dados.

19.1 FINDKEY

Basicamente é um comando FIND. Realiza uma pesquisa no arquivo de dados selecionado; se encontrar um registro que satisfaça as condições de procura, coloca-o no buffer de registro de arquivo e seta automaticamente os indicadores FOUND e FINDERR como True e False respectivamente.

Sintaxe:

FINDKEY modo arquivo BY index.n < FOR param1 EQ param2 < param3 EQ param4 > >

Onde:

1- modo : LT, LE, EQ, GE, GT
(1, 2, 3, 4, 5 respectivamente).

2- arquivo : nome do arquivo que será alvo da operação de pesquisa.

3- index.n : número do índice pelo qual será realizada a operação de pesquisa no arquivo selecionado.

4- param1 : é um campo do arquivo selecionado (deve pertencer ao índice declarado), que servirá para a operação de procura.

5- param2 : pode ser uma janela, uma variável (inteira, real, data ou string), uma constante (inteira, real, data ou string) ou um campo de um arquivo diferente ou não daquele que será alvo da pesquisa.

6- param3 e param4 : idem aos itens 4 e 5, não são obrigatórios.

Este comando só deve ser usado se você possuir todos os dados a respeito do índice pelo qual será executada a pesquisa, isto é, você deve ter todos os dados, todos os conteúdos dos campos que pertencem ao índice.

Exemplo:

```
FINDKEY eq cadtel by index.1 FOR cadtel.codigo EQ tela.1 ;
                                 cadtel.cidade EQ "SAO PAULO"
```

Obs.: os dois pontos ":" usados na primeira linha do comando indicam que o comando continuará na próxima linha. Este recurso, de colocação de pedaços de comandos em outras linhas, deve ser usado com cautela, pois é aconselhado somente quando o comando completo não cabe em uma única linha. Outras opções para que isto ocorra devem ser verificadas e se possível, evitadas, devido a erros que possivelmente possam ocorrer.

Como você pode perceber, este comando tem a estrutura básica do comando FIND, com a ressalva que a pesquisa deve ocorrer quando tivermos todos os dados a respeito do índice de procura. Vejamos agora um paralelo entre os dois comandos: FIND e FINDKEY.

```
CLEAR CADTEL
MOVE TELA.1 TO CADTEL.CODIGO
FIND EQ CADTEL BY INDEX.1
  e
FINDKEY eq cadtel BY index.1 FOR cadtel.codigo eq tela.1
```

19.2 SEARCHKEY

É muito parecido com os comandos FINDKEY e FIND, porém a pesquisa pode ocorrer sem que saibamos exatamente todos os conteúdos dos campos que formam o índice pelo qual será realizada a procura, além do que, poderemos realizar pesquisas em índices não-exclusivos, isto é, em índices que aceitam a repetição do mesmo conteúdo dos dados do arquivo.

Sintaxe:

```
SEARCHKEY modo arquivo BY index.n < COMPARING param1 EQ param2;
                                  < param3 EQ param4 > >
```

Obs.: o sinal de dois pontos ":" foi usado para indicar que a sintaxe do comando continua na próxima linha.

Onde:

1- modo: LT, LE, EQ, GE, GT
(1, 2, 3, 4, 5 respectivamente).

2- arquivo: nome do arquivo que será alvo da operação de pesquisa.

3- index.n: número do índice pelo qual será realizada a operação de pesquisa no arquivo selecionado.

4- param1: é um campo do arquivo selecionado (deve pertencer ao índice declarado), que servirá para a operação de procura.

5- param2: pode ser uma janela, uma variável (inteira, real, data ou string), uma constante (inteira, real, data ou string) ou um campo de um arquivo diferente ou não daquele que será alvo da pesquisa.

6- param3 e param4: idem aos itens 4 e 5, não são obrigatórios.

Como o FINDKEY, o comando SEARCHKEY também tem um paralelo com o comando FIND:

```
CLEAR CADTEL
MOVE TELA.2 TO CADTEL.NOME
FIND GE CADTEL BY INDEX.2
[FOUND] begin
   IF CADTEL.NOME EQ TELA.2 INDICARTE FOUND TRUE
   INDICATE FOUND AS FALSE // [NOT FOUND] = TRUE
end
```

e

```
SEARCHKEY eq cadtel BY index.2 COMPARING cadtel.nome eq tela.2
```

19.3 SUM

Sua função básica é de totalizar um campo de um arquivo de dados selecionado e guardar o resultado deste somatório em uma variável numérica. Este comando realizará uma pesquisa seqüencial no arquivo de dados e, segundo condições predefinidas, acumulará em uma variável o total apurado. Este comando é parecido com os dois anteriores, somente com a diferença que este percorrerá um arquivo de dados, do início ao fim, antes de devolver o controle do fluxo do programa para o usuário.

Sintaxe:

```
SUM arquivo < USING campo1 TO arg1 < COMPARING campo2 modo arg2 ;
                                   < campo3 modo arg3 > > >
```

Onde :

1- arquivo: é o arquivo alvo da pesquisa e totalização.

2- campo1: é o campo do arquivo de dados que será totalizado, e que pode ser um campo ASCII, numérico ou data.

3- arg1: é uma variável (inteira, real) ou janela que guardará o resultado da totalização.

4- campo2: é o campo do arquivo selecionado que servirá para a conferência da condição de seleção.

5- arg2: é o valor (conteúdo) que servirá para a comparação com o argumento anterior (campo2).

6- campo3 e arg3 : idem aos itens 4 e 5.

Este comando também tem seu paralelo com o comando FIND:

```
INTEGER SOMA
DATE DAT_INICIO DAT_FIM
STRING TECLA 1
OPEN LOCACAO
CLEAR LOCACAO
MOVE 0 TO SOMA
ACCEPT DAT_INICIO
ACCEPT DAT_FIM
LOOP:
    FIND GT LOCACAO BY RECNUM
    [FOUND] begin
      IF LOCACAO.DATA GE DAT_INICIO // estrutura: E logico (AND)
           IF LOCACAO.DATA LE DAT_FIM
             MOVE (LOCACAO.VALOR + SOMA) TO SOMA
      GOTO LOOP
    end
    [NOT FOUND] begin          // final do arquivo
        showln "Valor arrecadado no periodo : " soma
        inkey tecla
      end
ABORT
```

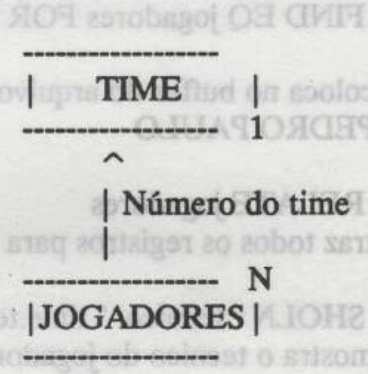
e

```
INTEGER SOMA
DATE DAT_INICIO DAT_FIM
STRING TECLA 1
OPEN LOCACAO
CLEAR LOCACAO
MOVE 0 TO SOMA
ACCEPT DAT_INICIO
ACCEPT DAT_FIM

SUM locacao USING locacao.valor TO soma ;
          COMPARING locacao.data ge dat_inicio ;
               locacao.data le dat_fim

    showln "Valor arrecadado no periodo : " soma
    inkey tecla
    ABORT
```

CAPÍTULO 20
RELACIONAMENTO DE MÚLTIPLOS ARQUIVOS

Supondo que temos uma aplicação com apenas dois arquivos de dados, quais sejam:

TIME	JOGADORES	Graficamente teríamos:
Número do Time	Nome	
Técnico	Idade	
Diretor	Número do time	
Endereço	Posição	



E temos duas situações:

1) Dado um jogador qualquer, quem é seu técnico?

2) Dado um diretor, quem compõe o seu elenco de jogadores?

Os comandos RELATE e ATTACH são os melhores para se solucionar estas questões.

20.1 RELATE

O comando RELATE traz para o buffer todos os registros de um arquivo a a que um dado registro aponta. Vejamos um exemplo:

Temos no buffer do arquivo de jogadores um registro que contém o jogador PEDRO PAULO e queremos saber o endereço do clube onde ele treina. Para isto basta executarmos:

RELATE jogadores

Automaticamente será colocado no buffer time, o registro do time onde joga PEDRO PAULO. Veja o fragmento de programa a seguir:

```
FIND EQ jogadores FOR nome EQ "PEDRO PAULO"

// coloca no buffer do arquivo jogadores o registro que contem o jogador
// PEDRO PAULO

RELATE jogadores
// traz todos os registros para os quais o arquivo jogadores aponta: time

SHOLN " tecnico: " time.tecnico
// mostra o tecnico do jogador PEDRO PAULO
```

20.2 ATTACH

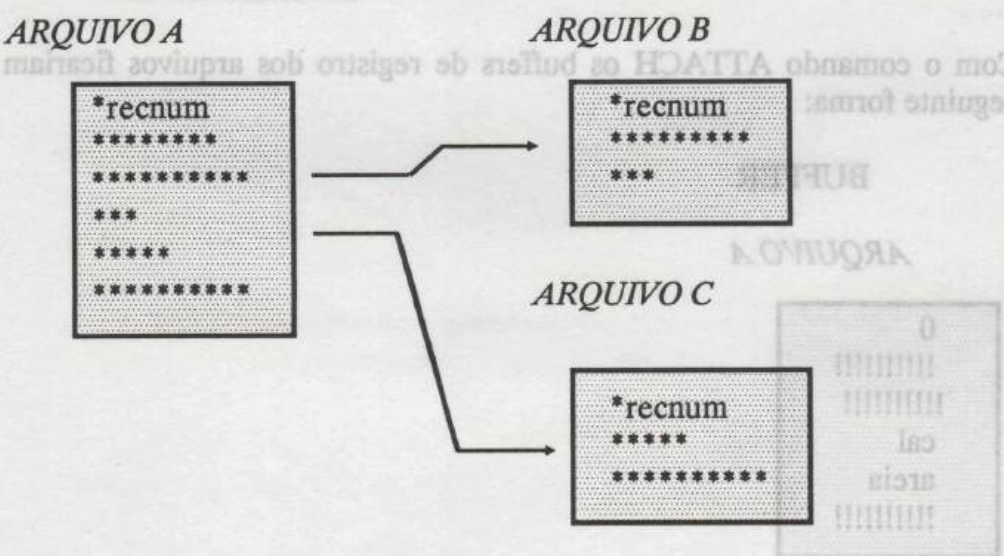
O comando ATTACH faz o contrário do comando RELATE, no caso deste, a partir de registros *filhos* (aqueles que apontam para um dado registro de outro arquivo de dados) ele traz o registro *pai*. No caso do ATTACH a partir do *pai* ele traz o *filho*.

Você deve estar se perguntando:

"Um pai pode ter vários filhos?"

"E se eu der um técnico qual jogador aparecerá (será chamado)?"

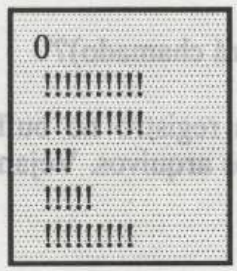
Na realidade, o comando ATTACH não coloca um registro no buffer, mas apenas prenche os campos que apontam para outros arquivos. Vejamos um exemplo:



Supondo que o buffer do ARQUIVO A estivesse vazio e que os demais buffers estivessem com os seguintes dados:

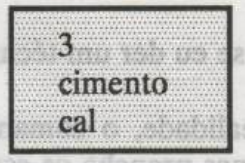
BUFFERs

ARQUIVO A

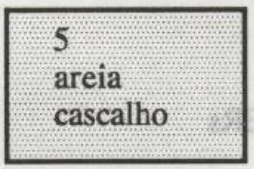


Vazio

ARQUIVO B



ARQUIVO C



Com o comando ATTACH os buffers de registro dos arquivos ficariam da seguinte forma:

BUFFER

ARQUIVO A

Note que o buffer do ARQUIVO A não possui um registro ativo, isto é, diferente de zero, o que nos diz que não temos um registro completo e ativo no buffer do ARQUIVO A. Imagine que os campos preenchidos façam parte de um índice, por exemplo o índice um. Execute portanto o seguinte comando e veja os resultados:

FIND EQ ARQUIVO_A BY INDEX.1

BUFFERs

ARQUIVO A

3
tijolos
pedras
cal
areia
vigas

ARQUIVO B

3
cimento
cal

ARQUIVO C

5
areia
cascalho

Note que agora, após o comando FIND já temos um registro ativo no buffer.

Obs.: - O comando SAVE executa automaticamente um comando ATTACH. O comando RELATE só terá efeito quando houver ao menos um registro no arquivo *filho*.

- Se o arquivo *pai* não estiver aberto, o comando RELATE não dará qualquer mensagem de erro.

CAPÍTULO 21
USANDO O REPORT

21.1 POR QUÊ?

Como vimos anteriormente podemos usar a Macro REPORT para fazermos seleções de dados, quando queremos emitir um relatório. Isto não é totalmente verdade. Podemos usar o REPORT para selecionar determinados dados de determinados arquivos sem que seja necessário gerar um relatório específico para estes dados ou arquivo. Como a Macro Report pode estar presente uma única vez no programa ou aplicação Dataflex, devemos escolher corretamente como iremos usá-la. Este fato não indica que esta Macro não possa ser acessada diversas vezes, portanto podemos deixar a Macro Report em uma parte separada do programa, como uma rotina a ser executada somente quando for necessário. A Macro Report controla sozinha todo o processo de leitura, seleção e impressão dos registros lidos e selecionados.

21.2 NÃO SÓ EM RELATÓRIOS

Como foi dito o uso do Report não é exclusivo de relatórios, portanto poderíamos usá-lo no programa de baixa de locações para selecionar os dados do arquivo de ficha e totalizá-los.

/TELA resident Programa:BAIXALOC.FRM

BAIXA DE LOCACAO DE FITAS Data : __/__/__

LOCACAO : ____. Data da locacao : __/__/__

Codigo : _____. Cliente : _____

Numero de fitas locadas : _ Responsavel : _____

BAIXA : ____. Data da baixa : __/__/__

Dias de locacao : __.

Valor a pagar em __/__/__ : _____.__

Total a pagar em __/__/__ : _____.__

[HELP Disponivel]

/HELP resident

TELA DE AJUDA A BAIXA DE LOCACAO DE FITAS

Esta tela pode ser usada somente para operacoes de baixas de locacoes de fitas de video da Locadora. Onde :

Locacao : Numero da locacao ainda nao-baixada. Todos os dados referentes a locacao serao mostrados, menos seu total a pagar, pois deveremos confirmar se a locacao e a correta.

Baixa : Mesmo numero da locacao. Nao teremos acesso a ele.

Data da Baixa : E a data de entrega das fitas locadas. Caso seja o mesmo dia ou o dia seguinte, sera cobrada uma diaria completa, caso contrario, serao calculados os dias de locacao pela formula : (data_locacao - data_baixa) = numero_de_dias_de_locacao.

Dias de Locacao : Resultado do calculo acima mencionado.

Valor a pagar : Valor a pagar por uma unica diaria.

Total a pagar : Valor a pagar apos o calculo dos dias de locacao.

< qualquer tecla volta ao programa ... >

```
/BODY resident
_____.__ _____.__
/*

// abrindo arquivos

OPEN locacao index.1
OPEN ficha   index.1

// criando variaveis

INTEGER  diferenca
NUMBER     valor desconto total_a_pagar
STRING   tecla 1
DATE       data_sys  data_locacao data_baixa
SYSDATE  data_sys
MOVE data_sys TO tela.1
```

```
// inicio do programa
Comeco:
      CLEAR  locacao
      CLEAR  ficha

      CLEARFORM tela.2 THRU tela.14
      CLEARXY 24 0

      MOVE 0 TO valor
      MOVE 0 TO desconto
      MOVE 0 TO total_a_pagar
      MOVE 0 TO diferenca
      ENTERGROUP

         AUTOPAGE tela 2
         ENTRY locacao.numero              {autofind,findreq}
         ENTRY locacao.data                {displayonly}
         ENTRY locacao.codigo_cliente      {displayonly}
         ENTRY locacao.nome_cliente        {displayonly}
         ENTRY locacao.num_fitas_loc       {displayonly}
         ENTRY locacao.responsavel         {displayonly}
      ENDGROUP

      GOSUB Verifica          // Verifica se a locacao ja nao foi baixada
      GOSUB Confirma          // Confirma baixa daquela locacao
      GOSUB Calcula_valor     // Calculo do valor que deveria ser pago
                              // por uma unica  diaria e depois
                              // o total a pagar pela locacao
      GOSUB Baixa             // Efetiva a baixa naquela locacao

goto Comeco
```

```
////////////////////// sub-rotinas ////////////////////////////////

Verifica:
      IF locacao.baixa EQ "S" begin
                              gotoxy 24 0
                              show " Esta locacao ja foi baixada !!!."
                              inkey tecla
                              clearxy 24 0
                              return comeco
                        end

      RETURN

Calcula_valor:

      MOVE tela.2          TO tela.8
      MOVE data_sys        TO tela.9
      MOVE data_sys        TO data_baixa
      MOVE tela.3          TO data_locacao
      CALC ( data_baixa - data_locacao ) TO diferenca // calcula o
                                                      // numero de  dias de locacao
IF diferenca EQ 0 MOVE 1 TO diferenca    // se foi baixada no mesmo dia
                                                      // cobra uma diaria inteira
      MOVE diferenca TO tela.10
      MOVE data_locacao TO tela.11
      GOSUB Totaliza_ficha
      MOVE data_baixa TO tela.13
      CALC ( diferenca * total_a_pagar ) TO total_a_pagar
      MOVE total_a_pagar to tela.14
      RETURN

Baixa:

      MOVE "S"             TO locacao.baixa
      MOVE data_baixa      TO locacao.data
      MOVE total_a_pagar   TO locacao.valor
```

```
        CLEARXY 24 0
        GOTOXY 24 0

        SHOW    " Efetuando a baixa na locacao : " locacao.numero
        SAVE    locacao
        SHOW    " .    Uma tecla para continuar !"
        INKEY   tecla
        CLEARXY 24 0
        RETURN

Confirma:

        CLEARXY 24 0
        GOTOXY 24 0
        SHOW " Confirma baixa desta locacao (S/N) ?"
        INKEY TECLA
        CLEARXY 24 0
        IF tecla IN "Nn" RETURN Comeco
        RETURN

Totaliza_ficha:

        REPORT FICHA BY INDEX.1
           INDICATE SELECT AS FICHA.LOCACAO EQ LOCACAO.NUMERO
        [SELECT] INDICATE SELECT AS LOCACAO.BAIXA EQ "N"

SECTION BODY
        PRINT FICHA.VALOR_LOCACAO  TO BODY.1
        PRINT FICHA.DESCONTO       TO BODY.2
CALC (FICHA.VALOR_LOCACAO - FICHA.DESCONTO + total_a_pagar) TO
TOTAL_A_PAGAR
REPORTEND

MOVE total_a_pagar TO tela.12

RETURN
```

```
////////////////////////// Redefinicao de Flex-Keys //////////////////////

KEYPROC key.clear
    CLEARFORM TELA.2 THRU TELA.14
    RETURN COMECO
    RETURN

KEYPROC key.escape
    CLEARXY 24 0
    GOTOXY 24 0
    SHOW "Deseja sair (S/N) ? "
    INKEY tecla
    clearxy 24 0
    IF tecla IN "Ss" abort
    ENTAGAIN
    RETURN

KEYPROC key.delete
    clearxy 24 0
    gotoxy 24 0
    show "TECLA DESATIVADA !!!"
    inkey tecla
    clearxy 24 0
    entagain
    RETURN
```

CAPÍTULO 22
PASSANDO E RECEBENDO PARÂMETROS

Passar e receber parâmetros é um ótimo negócio quando precisamos agilizar ou reduzir passos de programação de uma aplicação, diminuindo, portanto, o tempo gasto com simples entradas de dados, às vezes somente letras ou números.

Estes parâmetros podem ser usados de muitas maneiras diferentes, tais como: selecionar a saída de um relatório para o vídeo, para a impressora ou um arquivo, com a simples emissão de uma única letra ou número.

Veremos, a seguir, os comandos básicos para emissão e recebimento de parâmetros, tanto do sistema operacional como de dentro de aplicações Dataflex.

22.1 CHAIN

Sua função básica é de executar um outro programa abandonando aquele em que se encontra. Isto é, de dentro de um programa, aplicação Dataflex, podemos executar um outro programa, aplicação também Dataflex ou não. Quando o comando CHAIN desvia a execução de um programa para outro

ele automaticamente fecha todos os arquivos abertos no programa e inicia uma série de procedimentos para que o programa indicado possa ser executado.

Variáveis inteiras poderão ser reaproveitadas no programa que será executado, bastando, para isto, que neste programa elas sejam declaradas na mesma ordem com que foram no programa que continha o comando CHAIN.

Sintaxe:

1) CHAIN "nome_programa_destino" ["param1" .. "param n"]

2) CHAIN str1

No primeiro caso: o comando Chain procurará um programa, com terminação ".FLX", com o mesmo nome do primeiro conjunto de caracteres (palavra) da string e o executará caso o ache. Caso contrário ocorrerá um erro por não ter sido achado o arquivo declarado. As "palavras" seguintes são os parâmetros que serão passados ao programa-destino, que podem ser números, conjuntos de letras ou nomes de dispositivos de direcionamento de saída (não devendo estar entre aspas). Os outros devem obrigatoriamente estar entre aspas ("), com tamanho máximo de 255 caracteres.

No segundo caso: a string declarada deve conter o nome do programa a ser executado. Basicamente é parecido com o formato anterior, porém, neste formato não poderemos passar parâmetros para o programa-destino.

Exemplo:

```
KEYPROC Key.user
    GOTOXY 24 0
    SHOW " Executando o Programa Cadastro de Clientes !."
    CHAIN "cadcli" "45"
    RETURN
```

```
KEYPROC Key.user2

    ACCEPT      tela.15     {capslock}
    MOVE        tela.15     TO arquivo
    CHAIN       arquivo
    RETURN
```

22.2 CMDLINE

Sua função é recuperar os parâmetros passados pelo comando CHAIN. Estes parâmetros serão guardados em variáveis que devem ser definidas anteriormente. A recuperação dos parâmetros deve obedecer a mesma ordem com que foram enviados, caso contrário um parâmetro será recebido no lugar de um outro.

Sintaxe:

```
CMDLINE var1 .. var n
```

Neste caso as variáveis Var1 até Varn deverão ter sido previamente definidas como strings. Portanto Var1 receberá o primeiro parâmetro enviado pelo comando CHAIN assim como Var3 receberá o terceiro e assim por diante.

Exemplo:
1) Programa1:

```
KEYPROC Key.user
   GOTOXY 24 0
   SHOW " Executando o Programa Cadastro de Clientes !."
   CHAIN "cadcli" "45"
   RETURN
```

```
Programa_destino:

STRING codigo_cliente 5
CMDLINE codigo_cliente

FINDKEY EQ Cadcli BY index.1 FOR Cadcli.codigo EQ codigo_cliente
[found] begin
    ...
```

2) Programa1:

```
...
...
IF tela.2 EQ "V" CHAIN "RELFAT"  "11"  CON:       // video
IF tela.2 EQ "I" CHAIN "RELFAT"  "11"  ""         // impressora
IF tela.2 EQ "A" CHAIN "RELFAT" "11" "DESTINO.REL"// arquivo
GOTO Comeco
...

Programa_destino:

...
STRING locacao 5 destino 12
CMDLINE locacao destino

OUTFILE destino

REPORT ...
```

PARTE VI
APÊNDICES

Nesta parte do livro teremos acesso à informações que não foram dadas durante o desenrolar do texto devido ao fato de que eram assuntos secundários mas não de memor importância. Esta parte do livro está dividida em seis apêndices independentes.

PARTE VI
APÊNDICES

Nesta parte do livro teremos acesso à informações que não foram dadas durante o desenrolar do texto devido ao fato de que eram assuntos secundários mas não de menor importância. Esta parte do livro está dividida em seis apêndices independentes.

APÊNDICE I

DATAFLEX X Outros SGBDs

Neste apêndice veremos a comparação entre duas aplicações, uma gerada pelo Dataflex e um programa escrito em dBASEIII.

Não faremos qualquer comentário quanto ao desempenho das duas aplicações. Isto de deve ao fato de que desejamos que o julgamento destas fiquese a seu cargo, uma vez que somos suspeitos para opinar quanto a este assunto.

Primeiro veremos a aplicação gerada pelo Dataflex:

Estrutura do arquivo de dados:

```
    FILE DEFINITION - ESTRUTURA ARQ. #10
**************************************************************
    NOME BASE DO ARQUIVO   = LIVRO
    NOME PARA USUARIO      = LIVRO
    NOME PARA DATAFLEX     = LIVRO
**************************************************************
```

TAMANHO REGST = 170 (USADO= 132)
NUMERO MAXIMO DE REGS = 100 (USADO= 2)
ESPACO EXCLUIDO E REUTL
MULTI-USER RE-READ ATIVO

****

FIELD NMBR	FIELD OFFSET	FIELD LEN	FIELD TSPE	DEC PTS	MAIN INDEX	RELATES--TO FILE	FIELD	
1	1	3	NUMERIC	0	1	0	0	CODIGO
2	4	30	ASCII		0	0	0	NOME
3	34	30	ASCII		0	0	0	ENDERECO
4	64	3	NUMERIC	0	0	0	0	NUMERO
5	67	3	NUMERIC	0	0	0	0	APTO
6	70	20	ASCII		0	0	0	CIDADE
7	90	2	ASCII		0	0	0	ESTADO
8	92	5	ASCII		0	0	0	CEP
9	97	20	ASCII		0	0	0	BAIRRO
10	117	4	ASCII		0	0	0	DDD
11	121	9	ASCII		0	0	0	FONE
12	130	3	DATE		0	0	0	DATA_NASCIMENTO

INDEX 1: FIELD SEGMENTS: <1>

Listagem da aplicação:

/tela Programa: LIVRO.FRM

AGENDA TELEFONICA

CODIGO : < ___. >
NOME : _____
ENDERECO : _____ No : ____. APTO : ____.
CIDADE : _____ ESTADO : __ CEP : _____
BAIRRO : _____
TELEFONE : (____) _____DATA DE NASCIMENTO : __/__/__

[Use o campo CODIGO para pesquisa] [Pressione ESC para sair do programa]

```
                [ HELP Disponivel - tecle (F1) ]
/HELP
                        AGENDATELEFONICA
                          HELP GERAL
```

- Use o campo CODIGO para processos de pesquisa e procura : Use as Flex-Keys : F3 (SFIND), TAB (FIND), Pg Up (Previous Record) e Pg Dn (Next Record), alem de poder apenas digitar o numero do registro (acesso randomico direto), seguido de um ENTER.

- Para processos de inclusao : digite um novo codigo no campo codigo e a seguir entre com os dados referentes 'aquele registro.

- Para processos de alteracao : pesquise o arquivo de dados, ache o registro a ser alterado e modifique a informacao desejada.

- Para processos de exclusao : pesquise o arquivo de dados, ache o registro desejado e tecle a Flex-Key F6 (DELETE).

- Para limpar a tela de dados ou para cancelar uma entrada de dados incompleta use a Flex-Key F9 (CLEAR WINDOWs).

- Para gravar os dados das janelas no arquivo de dados, digite a Flex-Key F10 (SAVE), em qualquer parte da imagem.

- Para retroceder uma janela, use a Flex-Key F2 (BACK FIELD)

```
                [ Uma tecla volta para o programa ]
/*
        // indica fim da imagem
```

```
PAGE  tela              // imprime a tela
OPEN  LIVRO             // deixa o arquivo pronto para uso
ENTER LIVRO
AUTOPAGE tela           // enderecamento automatico das janelas
     ENTRY   LIVRO.CODIGO          {AUTOFIND}
     ENTRY   LIVRO.NOME            {CAPSLOCK}
     ENTRY   LIVRO.ENDERECO        {CAPSLOCK}
     ENTRY   LIVRO.NUMERO
     ENTRY   LIVRO.APTO
     ENTRY   LIVRO.CIDADE          {CAPSLOCK}
     ENTRY   LIVRO.ESTADO          {CAPSLOCK}
     ENTRY   LIVRO.CEP
     ENTRY   LIVRO.BAIRRO          {CAPSLOCK}
     ENTRY   LIVRO.DDD
     ENTRY   LIVRO.FONE
     ENTRY   LIVRO.DATA_NASCIMENTO
     RETURN
ENTEREND
ABORT        // fim
```

Vejamos agora a aplicação criada atraves do dBASEIII:

Estrutura do arquivo de dados:

```
Structure for database     : C:livro.dbf
Number of data records     :      5
Date of last update        : 03/16/88
Field     Field Name       Type          Width     Dec
    1     CODIGO           Numeric       5
    2     NOME             Character     30
    3     ENDERECO         Character     30
    4     NUM              Numeric       5
    5     APTO             Numeric       5
    6     CIDADE           Character     20
    7     ESTADO           Character     2
    8     CEP              Character     5
```

```
     9    BAIRRO          Character     20
     10   DDD             Character     4
     11   FONE            Character     9
     12   DATA_NASC       Date          8
** Total **                             144
```

Listagem do programa criado:

- Listagem do arquivo de rotinas do programa:

```
PROCEDURE mostra2
@ 09,15 SAY codigo        PICT "99999"
@ 11,15 GET nome          PICT "@!"
@ 13,15 GET endereco      PICT "@!"
@ 13,50 GET num           PICT "99999"
@ 13,62 GET apto          PICT "99999"
@ 15,15 GET cidade        PICT "@!"
@ 15,50 GET estado        PICT "!!"
@ 17,15 GET cep           PICT "99999"
@ 17,32 GET bairro        PICT "@!"
@ 19,17 GET ddd           PICT "9999"
@ 19,23 GET fone          PICT "999-99-99"
@ 19,60 GET data_nasc
return
procedure le
     READ
     CLEAR gets
RETURN
procedure mostra1
@ 08,01 TO 20,78
@ 09,05 SAY "Codigo...:"
@ 11,05 SAY "Nome.....: "
@ 13,05 SAY "Endereco.: "
@ 13,46 SAY "No.:"
```

```
@ 13,56 SAY "Apto.:"
@ 15,05 SAY "Cidade....: "
@ 15,42 SAY "Estado.: "
@ 17,05 SAY "C.E.P....: "
@ 17,24 SAY "Bairro.: "
@ 19,05 SAY "Telefone.: (____) _________"
@ 19,40 SAY "Data de nascimento.: "
return
procedure confirma
publ xresp2
xresp2=" "
@ 21,02 say "[M]antem registro ou [A]paga ? [_]"
   do while xresp2 # "M" .and. xresp2 # "A"
      @TAB3 = @ 21,35 get xresp2 pict "!"
      do le
enddo
return
```

- Listagem do programa principal:

```
****************** CADASTRO TELEFONICO ******************
***
***      REGINALDO QUINTELA DE CARVALHO
***      15/03/88 SAO PAULO SP
***
*************************************************************

USE livro index livro
SET talk off
SET echo off

CLEAR
```

```
SET PROCEDURE TO ROTINAS

@ 01,01 to 07,78

@ 01,30 SAY   " CADASTRO TELEFONICO "
@ 02,09 SAY   "[+] Reg.posterior"
@ 03,09 SAY   "[-] Reg.anterior "
@ 04,09 SAY   "[X] Inclui/ Altera/ Exclui/ Consulta"
@ 05,09 SAY   "[S] Sai"
@ 06,09 SAY   "[_] Opcao"

DO WHILE .T.
      do mostra1
      xresp1 = " "

DO WHILE xresp1 # "S" .AND. xresp1 # "+" .AND. xresp1 # "-" .AND.
xresp1 # "X"
      @ 06,10 GET xresp1 PICT "!"
      do le
ENDdo
@ 21,01 SAY spac(78)
DO CASE
CASE xresp1 = "S"
      EXIT
CASE xresp1 = "+"
  if eof()
      @ 21,01 say " Nao ha registro posterior. Barra de espaco para continuar"
      wait " "
      @ 21,01 say space(78)
      loop
  endif

CASE xresp1 = "-"
      if bof()
         @ 21,01 say " Nao ha registro anterior. Barra de espaco
                                                          para continuar"
         wait " "
         @ 21,01 say space(78)
```

```
        loop
      endif
      skip -1
CASE xresp1 = "X"
      GO BOTTOM
      xnumero = 1 + codigo
      @ 09,15 get xnumero pict "99999"
      do le
      GO CODIGO
      seek xnumero
      if eof()
         APPEND BLANK
         do mostra2
         do le
         REPLACE codigo with xnumero
         @ 09,02 CLEAR TO 20,77
         loop
      ENDif
  ENDcase
  do mostra2
  do le
  do confirma
  @ 21,01 say space(78)
  if xresp2 = "A"
      delete
  endif
  if xresp1 = "+"
      skip
   endif
  @ 09,02 CLEAR TO 20,77
enddo
clear
close data
return
```

APÊNDICE II
A VERSÃO 2.3
VALE A PENA CONHECÊ-LA!

Programando algum tempo com Dataflex, descobrimos alguns pontos que poderiam ser aperfeiçoados, sem dúvida necessários: Funções matemáticas e Gráficos.

Ao que tudo indica o pessoal da *DATACCESS* ouviu nossas preces e nos presenteou com a versão 2.3, que além de dar ótimas soluções para os problemas citados anteriormente, incorpora muitas outras melhorias, sendo a mais significativa, em termos de linguagem, a passagem do *PASCAL* para a linguagem *C* como linguagem-fonte na qual foi desenvolvida a nova versão. Desta forma, nós apaixonados pelo *C*, agora poderemos usá-la juntamente com o Dataflex 2.3. O que mais poderíamos querer?

Veja abaixo, de forma resumida os principais comandos gráficos e funções matemáticas da nossa nova versão 2.3. A seguir teremos alguns exemplos dos programas e dos gráficos gerados pelos mesmos.

- Funções matemáticas e trigonométricas

ACOS expressão
ASIN expressão
ATAN expressão
COS expressão
EXP expressão
INTEGER expressão
LENGTH variável
LOG expressão
REAL expressão
ROUND expressão
SIN expressão
SQRT expressão
TAN expressão

- Comandos gráficos

ARC raio ângulo ângulo cor_da_linha
BAR_CHART posição_horizontal valor_vertical < label > cor_de_preenchimento
BOX linha1 coluna1 linha2 coluna2 < cor_da_linha >
CHART BAR/ LINE/ PIE
CHART_INIT
CIRCLE linha coluna raio cor_da_linha < cor_de_preenchimento >
DISPLAY_GRAPHIC arquivo
FLOOD cor
GET_GRXY linha coluna
GR_PRINT < dispositivo_de_saída >
GRAPH_GETFIRST label valor
GRAPH_GETNEXT label valor
GRAPH_STAT mínimo máximo
GRAPH_SUM total contador
GRAPHIC ON/ OFF modo (LT, LE, EQ, GE, GT) < dispositivo_de_saída >
GRXY linha coluna
LINE linha coluna
PALETTE cor_primeiro_plano cor_de_fundo
PIE ângulo ângulo cor_da_linha cor_de_preenchimento
PLOT linha coluna

```
SAVE_GRAPH arquivo
SET_LINE_STYLE estilo tamanho
SET_TEXT orientação tamanho
SET_WORLD x1 y1 x2 y2
```

Programa-exemplo que gera três tipos de gráficos : barras, torta, linha.

```
/*
//
// Programa-exemplo da utilizacao do DataFlex para se gerar graficos.
//

string temp 1
CHART_INIT
GRAPH_VALUE "JAN" 700 // Estes valores serao usados nos graficos gerados
GRAPH_VALUE "FEV" 900 // pelos comandos CHART.
GRAPH_VALUE "MAR" 800
GRAPH_VALUE "ABR" 600
GRAPH_VALUE "MAI" 500
GRAPH_VALUE "JUN" 100

GRAPHIC ON

CLEARSCREEN 0

CHART BAR "PRODUTO X" "Vendas" "Meses"   // Faz o grafico em forma de barras.
inkey temp                               // Ver grafico I.

CHART PIE "PRODUTO X" "Vendas" "Meses"   // Faz o grafico em forma de torta.
inkey temp                               // Ver grafico II.

CHART LINE "PRODUTO X" "Vendas" "Meses"  // Faz o grafico em forma de linhas.
inkey temp                               // Ver grafico III.
ABORT
```

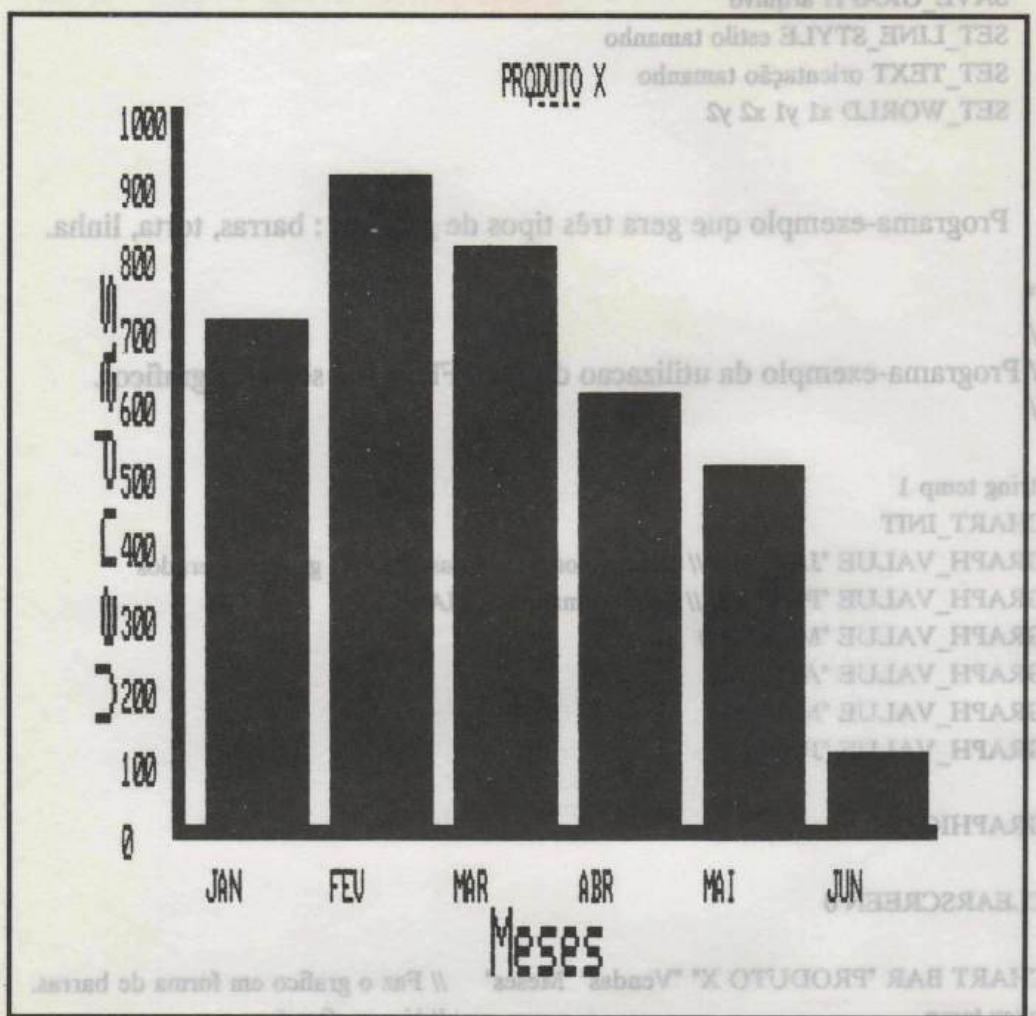

PRODUTO X
1000
900
800
700
600
500
400
300
200
100
0
JAN
FEV
MAR
ABR
MAI
JUN
Meses

Gráfico I

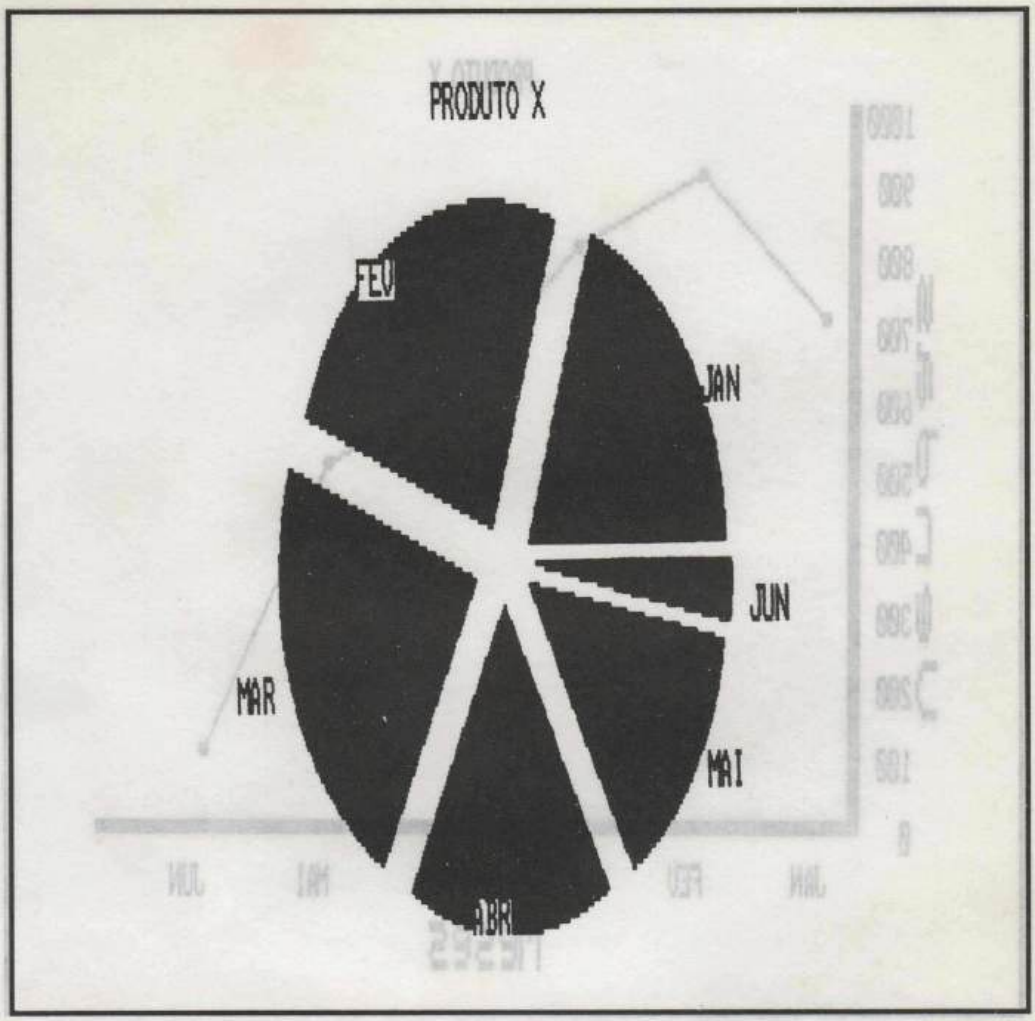


Gráfico II

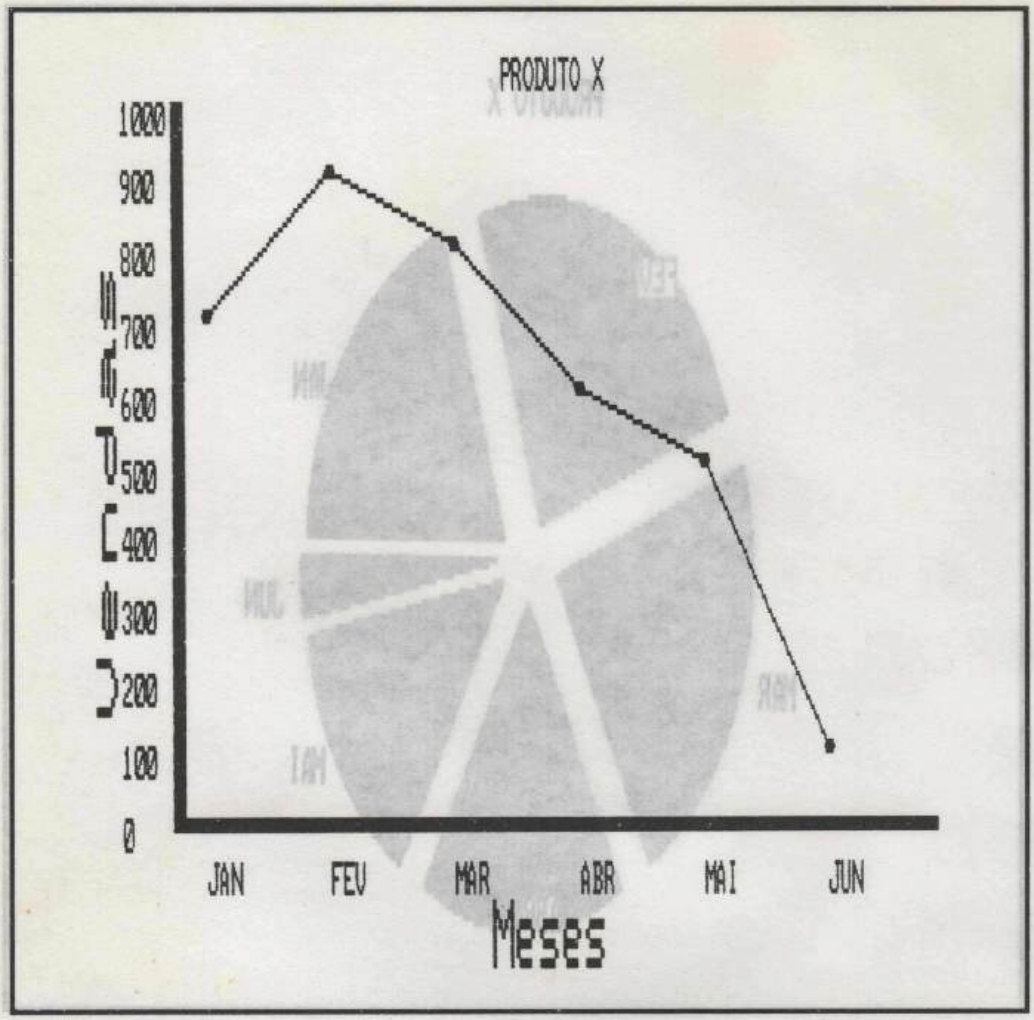

PRODUTO X
1000
900
800
700
600
500
400
300
200
100
0
JAN
FEV
MAR
ABR
MAI
JUN
Meses

Gráfico III

APÊNDICE III
RESUMO DE COMANDOS E SINTAXES

Neste apêndice, serão mostrados os comandos Dataflex, agrupados por função. Muitos deles não apareceram neste livro devido ao fato de que não seria possível o uso e explicação de todos estes comandos e também porque muitos deles são usados em tipos de processamento especiais e, portanto, aparecem raramente.

Aqueles que apareceram no livro tiveram uma rápida explicação de sua função e sintaxe. Veja aqui os resumos de suas sintaxes e em alguns casos algumas palavras a respeito de alguns comandos. Para maiores esclarecimentos consulte o Manual Dataflex, Volume II - Enciclopédia.

Antes de iniciarmos a relação de comandos e sintaxes, devemos ver primeiramente os comandos do módulo EDITOR, ao qual nos referimos no início do livro. Segue-se a lista de comandos e funções de cada um deles:

EDITOR

Tecla	Flex-Key	Função
F1	HELP	nenhuma
F2	BACK WIND	Load arquivo
F3	SFIND	Find texto (procura por palavra)
F4	PRINT	nenhuma
F5	CALCULATE	Home (cursor linha0 coluna 0)
F6	DELETE REC	Deleta linha do cursor
F7	USER	Save arquivo
F8	USER2	nenhuma
F9	CLEAR WINDs	nenhuma
F10	SAVE	Insere linha na posição do cursor

Tecla	Flex-Key	Função
TAB	FIND	Tabulação de 8 em 8 casas (dir, esq)
ENTER	ENTER	Confirma linha e passa para a próxima
Pg Up	PREV REC	Uma página acima
Pg Dn	NEXT REC	Uma página abaixo
UP	UP	Cursor sobe uma linha
DOWN	DOWN	Cursor desce uma linha
LEFT	LEFT	Cursor uma casa à esquerda
RIGHT	RIGHT	Cursor uma casa à direita
DEL	DELETE CHAR	Exclui caractere abaixo do cursor
INS	INSERT CHAR	Insere um caractere na posição do cursor
BS	BACKSPACE	Deleta o caractere abaixo do cursor e apaga até o fim da linha
ESC	ESCAPE	Abandona o texto e salva

PALAVRAS RESERVADAS DO DATAFLEX E RESUMO DE SINTAXES

Convenções:

< >	: uso obrigatório
[]	: uso opcional
argumento	: janela, variáveis inteiras, reais, string, data e campos
argumento_int	: argumento inteiro
var_int	: variável inteira
expressão	: expressão numérica (+ - * /)
dispositivo	: dispositivo de saída (CON: tela, disco, impressora)
modo	: LT (1),LE(2),EQ(3),GE(4),GT(5)
seção	: HEADER, SUBHEADER### (1-9), BODY, (1-9), SUBTOTAL### TOTAL
###	: número
##	: número do arquivo de índices
#	: número da janela ou imagem

- COMANDOS DE PROCESSAMENTO NUMÉRICO:

MOVE	argumento TO argumento
CALC	expressão TO argumento
INCREMENT	var_int
DECREMEMT	var_int

- COMANDOS DE TELA

```
CLEARSCREEN
CLEARXY          linha coluna
GOTOXY           linha coluna
INKEY            argumento
INKEY$           argumento
INPUT            argumento
KEYCHECK         comando
SHOW             argumento1 [argumento2 ... argumento n]
SHOWLN           argumento1 [argumento2 ... argumento n]
```

- COMANDOS DE CONTROLE

```
ABORT
CHAIN            nome_programa [param1... param n]
DEBUG
ERROR            ### [mensagem]
GOSUB            rotina
GOTO             label
ON ERROR GOSUB  rotina
ON argumento_int GOSUB rotina1 [ rotina2...rotina n ]
ON argumento_int GOTO label1 [ label2...label n ]
RETURN           rotina
RETURN           label
```

- COMANDOS DE MANIPULAÇÃO DE ARQUIVOS

ATTACH	nome_arquivo
CLEAR	nome_arquivo
DELETE	nome_arquivo
FIND	modo nome_arquivo BY index.##
FINDKEY	nome_arquivo BY index.## FOR nome_arquivo.campo modo argumento
OPEN	nome_arquivo
RELATE	nome_arquivo
SAVE	nome_arquivo
SAVERECORD	nome_arquivo
SEARCHKEY	modo nome_arquivo BY index.## COMPARING nome_arquivo.campo modo argumento
ZEROFILE	nome_arquivo

- CRIAÇÃO DE VARIÁVEIS

DATE	data1 [data2...data n]
INDICATOR	ind1 [ind2...ind n]
INTEGER	var_int1 [var_int2...var_int n]
NUMBER	var_real1 [var_real2...var_real n]
STRING	str1 [str2...str n]

- COMANDOS DAS MACROS ENTER E ENTERGROUP

```
ENDGROUP
ENTDISPLAY
ENTER          nome_arquivo1 [ nome_arquivo2...nome_arquivo n ]
ENTEREND
ENTERGROUP
ENTERMODE
ENTRY          nome_arquivo.campo [ [ janela.# ] [opcoes] ]
```

- COMANDOS DE CONTROLE DE IMAGEM

```
ACCEPT          janela.# [ TO argumento [opcoes] ]
AUTOPAGE        imagem [#]
BLANKFORM       janela.# [ THRU janela.# ]
CLEARFORM       imagem
CLEARFORM       janela.# THRU janela.#
DISPLAY         argumento TO janela.#
FORMAT          janela.# [ THRU janela.# [opcoes] ]
NAME [ imagem.# [ nome_janela [ nome_janela... nome_janela ] ] ]
OUTFILE         dispositivo
OUTPUT          imagem
PAGE            imagem
PRINT           argumento [ TO janela.# [opcoes] ]
```

- COMANDOS DE DECISÃO

```
ELSE comandos
IF < [ ind1 [ ind2 ind3 ] ] > comandos
IF argumento modo argumento comandos
INDICATE ind1 AS argumento modo argumento
INDICATE ind1 GROUP ANY/ALL [ [not] ind2 ind3 ind4 ]
        [ AND/OR [not] ind5 ind6 ind7 ]
INDICATE ind1 STATUS nome_arquivo
INDICATE ind1 AS [not] ind2 [ ind3 ind4 ]
INDICATE ind1 [ <TRUE> ] [ <FALSE> ]
```

- COMANDOS DE CONTROLE DE FLEX-KEYs

```
BACKFIELD
ENTAGAIN
HELP          imagem
KEYPROC       flex-key comandos RETURN
```

- COMANDOS DE CONTROLE DA MACRO REPORT

```
    FORMFEED
    OUTPUT section

PAGECHECK numero_de_linhas
REPORT nome_arquivo [ BY/DOWN index.##
    [BREAK nome_arquivo.campo1 [nome_arquivo.campo2...
                                nome_arquivo.campo n] ]
REPORTEND
SECTION seção
```

- COMANDOS DE CONTROLE DE ARQUIVOS SEQÜENCIAIS

CLOSE_INPUT	
CLOSE_OUTPUT	
DIRECT_INPUT	dispositivo
OUTFILE	dispositivo
READ	argumento1 [argumento2.. argumento n]
READLN	argumento1 [argumento2.. argumento n]
WRITE	argumento1 [argumento2.. argumento n]
WRITELN	argumento1 [argumento2.. argumento n]

- COMANDOS DE CONTROLE DE STRINGs

APPEND	var_destino str1 [str2...str n]
ASCII	str1 TO var_int
CHARACTER	var_int TO str1
CMDLINE	variavel_string
INSERT	str1 IN str2 AT posicao
LEFT	str1 TO str2 tamanho
LENGTH	str1 TO var_int
MID	str1 TO str2 tamanho inicio
PAD	str1 TO str2 tamanho
POS	str1 IN str2 TO var_int
REPLACE	str1 IN str2 WITH str3
RIGHT	str1 TO str2 tamanho
TRIM	str1 TO str2
UPPERCASE	str1

- COMANDOS DE CONTROLE DE ESTRUTURAS DE PROGRAMAS

```
BEGIN              comandos END
FOR var_int FROM início TO fim [ STEP passo ] comandos LOOP
REPEAT             comandos LOOP
REPEAT             comandos UNTIL    argumento modo argumento
WHILE              comandos END
```

- COMANDOS DE ACESSO AO SISTEMA OPERACIONAL

```
COPYFILE    nome_arquivo_fonte TO nome_arquivo_destino
DIRECTORY   parâmetros
ERASEFILE   nome_arquivo
FIELD_DEF   nome_arquivo ###_campo TO tipo_dado indice_principal
FILE_SIZE   nome_arquivo TO tam_corrente tam_maximo num_campos
FILELIST    num_arquivo TO nome_usuário nome_arquivo
INDEX_DEF   nome_arquivo INDEX.## TO str1
MAKE_FILE   [drive] nome_arquivo_.DEF [drive] ##_arquivo
MEMORY      argumento
REGISTRATION
RENAMEFILE      nome_arquivo1 TO nome_arquivo2
RUNPROGRAM      nome_arquivo_.COM parametros
SYSDATE         var_data [ var_int_horas var_int_minutos ]
SYSTEM
```

APÊNDICE IV
RESUMO DE FLEX-KEYS E FUNÇÕES

FLEX-KEY			FUNÇÃO
F1	-	HELP	Mostra uma tela de HELP caso ela exista
F2	-	BACK WIND	Retrocede uma janela
F3	-	SUPERFIND	Procura por um registro através de uma chave
F4	-	PRINT SCR	Imprime a imagem na impressora
F5	-	CALCULATE	Realiza um cálculo com resultado na janela
F6	-	DEL RECORD	Exclui o registro residente no buffer
F7	-	USER1	Tecla definível1
F8	-	USER2	Tecla definível2
F9	-	CLEAR WIND	Limpa as janela e os buffers
F10	-	SAVE RECORD	Salva os dados das janelas para os arquivos

TAB	-	Find	Procura um registro idêntico ao da janela
Pg Dn	-	Previous Rec	Vai para o registro anterior
Pg Up	-	Next Record	Vai para o próximo registro
Ins	-	Insert	Insere caractere branco na posição do cursor
Del	-	Delete	Exclui o caractere na posição do cursor
ESC	-	Escape	Abandona o programa
ENTER	-	Return	Confirma a entrada de dados

APÊNDICE V
PALAVRAS RESERVADAS DO DATAFLEX

ABORT	ENTDISPLAY	JOIN	PAGE	THROUGH
ACCEPT	ENTER	JOINEND	PAGE.LINES	THRU
ALL	ENTER.CLEAR	JOINMOVE	PAGEBREAK	TO
AND	ENTER.DELETE	KEY.CLEAR	PAGECHECK	TOTAL
ANY	ENTER.EDIT	KEY.DELETE	PAGECOUNT	TRIM
APPEND	ENTER.EXIT	KEY.DOWN	PAGEEND	TRUE
AS	ENTER.SAVE	KEY.ESCAPE	PAGEFEED	UNLOCK
ASCII	ENTEREND	KEY.FIELD	PAGE1...255	UNTIL
ATTACH	ENTERGROUP	KEY.FIND	PATHNAME	UPPERCASE
AUTOFIND	ENTERMODE	KEY.HELP	PAUSE	USING
AUTOFIND_GE	ENTFIND	KEY.NEXT	POINTS	WAIT
AUTOPAGE	ENTRY	KEY.PREVIOUS	POP	WHILE
BACKFIELD	ENTSFIND	KEY.RETURN	POS	WINDOWINDEX
BEGIN	EQ	KEY.SAVE	PRINT	WITH
BLANKFORM	ERASEFILE	KEY.SFIND	QUIET	WRITE
BREAK	ERR	KEY.UP	RANGE	WRITELN
BREAKINIT	ERRLINE	KEY.USER	READ	ZEROFILE
BREAKPOINT	ERROR	KEYCHECK	READ_BLOCK	
BY	FALSE	KEYPRESS	READLN	
CALC	FIELD_DEF	KEYPROC	RECCOUNT	
CALCULATE	FIELDINDEX	LASTERR	RECNUM	
CAPSLOCK	FILE_SIZE	LE	REGISTRATION	
CASE	FILELIST	LEFT	RELATE	
CASEELSE	FILELNUMBER	LENGTH	REMOVE	
CASEEND	FILL	LINECOUNT	RENAMEFILE	

CHAIN
CHARACTER
CHECK
CLEAR
CLEARFORM
CLEARSCREEN
CLEARWARNING
CLEARXY
CLOSE_INPUT
CLOSE_OUTPUT
CMDLINE
COMPARING
CONDITITONAL
CONDITIONALEND
COPY
COUNT
CURRENT_PAGE
CURRENT_WINDOW
CURRENT_IMAGE
DATE
DEBUG
DELETE
DECREMENT
DESPOOL
DIRECT_INPUT
DIRECT_OUTPUT
DIRECTORY
DISPLAY
DISPLAYONLY
ELSE
END
ENDOFREPORT
END.OF.REPORT
ENDGROUP
ENT$PERMISSIVE
ENTAGAIN
FIND
FINDERR
FINDKEY
FINDREQ
FLEXKEY
FLOAT$
FOOTER
FOOTER_COMPLETE
FOR
FORCEPUT
FORMAT
FORMFEED
FOUND
FROM
GE
GOSUB
GOTO
GOTOXY
GROUP
HEADER
HELP
IF
IFCHANGE
IFNOT
IN
INCREMENT
INDEX.BATCH
INDEX.1...10
INDEX_DEF
INDICATE
INDICATOR
INKEY
INKEY
INPUT
INSERT
INTEGER
LOCK
LOOP
LT
MAKE_FILE
MATCH
MEMORY
MERGE
MID
MOD
MOVE
MOVEINT
MOVENUM
MOVE
STR
MULTIUSER
NAME
NE
NEWPAGE
NEXT
NOENTER
NOISY
NOMATCH
NOPUT
NOT
NUMBER
NUMPAGE
NUMWINDOW
ODD
OFF
ON
OPEN
OR
OUTCLOSE
OUTFILE
OUTPUT
PAD
REPEAT
REPLACE
REPLACEALL
REPORT
REPORTEND
REQUIRED
REREAD
RETAIN
RETAINALL
RETURN
RIGTH
RPT.anything
RUNPROGRAM
SAVE
SAVERECORD
SCREENMODE
SEARCHKEY
SECTION
SELECT
SEQEOF
SEOFL
SETCHANGE
SHOW
SHOWLN
SKIPFOUND
STATUS
STRING
STRLEN
STRMARK
SUBHEADER1...9
SUBTOTAL
SUM
SUPRESS
SYSDATE
SYSTEM
TERMCHAR

APÊNDICE VI
GLOSSÁRIO

APLICAÇÃO: um trabalho ou conjunto de tarefas a serem realizadas pelo computador, através de configurações também chamadas de programas.

ARQUIVO: conjunto de registros com a mesma estrutura.

ASCII: American Standart Code for Information Interchange. Este código assinala combinações de 7 dígitos binários que codificam os 128 caracteres usados pelo computador. Estes sete bits codificados transcrevem letras, números e símbolos especiais.

BANCO DE DADOS: uma coleção de informações, estruturadas de forma a facilitar o acesso aos dados pelo computador.

BYTE: conjunto de oito bits, que correspondem a uma palavra; é a menor unidade de informação de um SGBD.

CAMPO: um tipo simples de dado que faz parte de um registro.

CAMPO CHAVE: campo(s) do arquivo de dados ao qual está associada a chave de ordenação.

CHAVE: uma combinação de letras e/ou números que servirão para organizar um arquivo de dados.

DADOS: é a verdadeira informação contida em arquivos de bancos de dados. A menor unidade em um banco de dados é um campo de um registro em um arquivo.

EDITOR: programa que manipula informações não-formatadas, normalmente um documento (cartas, relatórios) ou programas-fonte.

ENTRADA DE DADOS: processo de colocar novas informações no computador através de um teclado ou de algum outro dispositivo de entrada.

EXTENSÃO: convenção de nomenclatura de programas ou arquivos adotada por diversos sistemas operacionais, formado basicamente por um ponto (.) seguido de 3 caracteres (letras, números ou espaços em branco). Em Dataflex:

.DAT - arquivo de dados
.Kn - arquivo de índices
.RPT - programa-fonte de geração de relatórios
.FLX - programa final após compilação (pronto para uso)
.FRM - programa-fonte de aplicações de entrada de dados
.TAG - arquivo do Dicionário de dados que guarda os nomes dos campos do arquivo
.CFG - arquivo de informação para o sistema de aplicações
.FD - arquivo que guarda as definições dos arquivos de dados
.DEF - arquivo que guarda a estrutura do arquivo na forma de um texto

FLEX-KEYs: teclas de função programáveis que podem ser instaladas para o auxílio do uso do computador tarefas.

ÍNDICE: uma lista organizada de chaves associada aos registros de um arquivo de dados. Estes dados podem ser acessados seqüencialmente dentro desta lista, porém podendo também serem acessados randomicamente pela mesma.

JANELA: área do monitor de vídeo usada para processos de entrada de dados ou de impressão de dados.

MANIPULAÇÃO DE ARQUIVOS DE DADOS: operações de alteração, inclusão, exclusão e consulta dos dados do arquivo.

MENU: conjunto de opções que indicam as aplicações possíveis e disponíveis de um sistema de aplicações.

PROGRAMA: um conjunto específico de instruções que guiará o computador à realização de uma determinada tarefa.

QUERY: uma aplicação que permite a você extrair dados de arquivos de dados em forma de relatórios de forma individualizada.

REGISTRO: conjunto de campos, portanto de informações, que agrupadas formam um arquivo de dados.

RELATÓRIO: resultado final de um procedimento de extração de dados de um ou mais arquivos de dados.

SELEÇÕES: processo de escolha, filtragem dos dados de um ou mais arquivos, normalmente usado durante processo de emissão de relatórios.

TECLA DE FUNÇÃO: uma tecla do teclado de um computador, configurada para, assim que for teclada, realizar uma determinada rotina predefinida pelo programador ou uma rotina predefinida pela aplicação Dataflex.

Foto: Leo Burd

DANIEL JOSÉ BURD *iniciou sua jornada no ramo editorial de informática com a publicação de jogos em linguagem de máquina para o TK 85; ministrou cursos da linguagem Assembler para TK 90X na revista* Micro Hobby *e curso sobre MSX na revista* Micro Sistemas.

Sua experiência com o DATAFLEX provém da atuação como analista durante 2 anos na empresa Banco de Idéias, da qual foi sócio fundador.

Hoje é sócio da BURD Informática, *atuando na área industrial com o sistema operacional PICK.*

Este livro tem por objetivo permitir a aprimoração daqueles que já conhecem esta poderosa linguagem e também fornecer subsídios a uma análise crítica do DATAFLEX por aqueles que já tiveram algum contato com outro sistema ou querem escolher o seu gerenciador de banco de dados.

0-07-450110-0