

Copyright (c) 2003, Luciano Rogério Perdigão Braga.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

A copy of the license is included in the section entitled "GNU Free Documentation License".

Copyright (c) 2003, Luciano Rogério Perdigão Braga.

E garantida a permissão para copiar, distribuir e/ou modificar este documento sob os termos da GNU Free Documentation License, versão 1.1 ou qualquer outra versão posterior publicada pela Free Software Foundation; sem obrigatoriedade de Seções Invariantes na abertura e ao final dos textos.

Uma copia da licença esta incluída na seção intitulada GNU Free Documentation License.

Apostila do Curso **Básico de Natural**

Luciano Perdigão

natural_2-owner@yahoo.com.br

Grupo no Yahoo da Linguagem Natural 2

Para mais informações: http://br.groups.yahoo.com/group/natural_2

Enviar mensagem.....: natural_2@yahoo.com.br

Assinar.....: natural_2-subscribe@yahoo.com.br

Cancelar Assinatura..: natural_2-unsubscribe@yahoo.com.br

Proprietário da Lista: natural_2-owner@yahoo.com.br

HISTÓRICO DE ATUALIZAÇÕES

DATA	AUTOR	ATUALIZAÇÃO
22/08/2004	Luciano Perdigão	Inclusão novos comandos (MASK, SUBSTRING, MOVE BY NAME, SET CONTROL, etc..)
20/04/2004	Luciano Perdigão	Reorganização
16/11/2003	Luciano Perdigão	Inclui CALLNAT, CALL, FETCH
23/05/2003	Luciano Perdigão	Acertos de Português

ÍNDICE

1. INTRODUÇÃO AO NATURAL.....	7
1.1. COMPONENTES	7
1.2. SYSTEM FILE	8
1.3. BIBLIOTECAS.....	8
1.4. PROGRAMAS.....	9
1.5. INTERFACE NATURAL.....	9
1.6. DDM.....	10
1.7. OBJETOS NATURAL	12
2. SESSÃO NATURAL.....	12
2.1. COMANDOS DE UMA SESSÃO NATURAL	12
2.1.1. <i>Comandos de Controle de Bibliotecas</i>	13
2.1.2. <i>Comandos de Manipulação de Programas Fontes</i>	15
2.1.3. <i>Comandos de Manipulação de Programas Objetos</i>	15
2.1.4. <i>Comandos de Manipulação Mista de Programas</i>	15
2.1.5. <i>Comandos de Transferência/Limpeza de Work-Area</i>	15
2.1.6. <i>Comandos de Execução de Programas</i>	16
2.1.7. <i>Comandos de Processos Especiais</i>	16
2.2. EDITORES PARA OBJETOS.....	17
2.2.1. <i>Editor de Áreas de Dados</i>	17
2.2.1.1. Define Data.....	18
2.2.2. <i>Editor de Mapas</i>	21
2.2.3. <i>Editor de Programas</i>	29
3. VARIÁVEIS DE SISTEMA	31
4. COMANDOS	31
4.1. SET CONTROL	31
4.2. SET KEY	32
4.3. MASK	33
4.4. FORMAT	34
4.5. INSTRUÇÕES DE ENTRADA/SAÍDA	34
4.5.1. <i>Input</i>	34
4.5.2. <i>Reinput</i>	36
4.5.3. <i>Display</i>	37
4.5.4. <i>Write</i>	38
4.5.5. <i>Write Title/Trailer</i>	39
4.5.6. <i>Newpage</i>	40
4.5.7. <i>Eject</i>	40
4.5.8. <i>Skip</i>	41
4.6. OPERAÇÕES ARITMÉTICAS	41
4.6.1. <i>Add</i>	41
4.6.2. <i>Subtract</i>	42
4.6.3. <i>Multiply</i>	42
4.6.4. <i>Divide</i>	43
4.6.5. <i>Compute</i>	44
4.7. MANIPULAÇÃO DE CAMPOS	44
4.7.1. <i>Compress</i>	44
4.7.2. <i>Move</i>	44
4.7.3. <i>Substring</i>	46
4.7.4. <i>Reset</i>	46

4.7.5.	<i>Separate</i>	47
4.7.6.	<i>Examine</i>	47
4.8.	INSTRUÇÕES CONDICIONAIS.....	48
4.8.1.	<i>If</i>	48
4.8.2.	<i>Decide On/Decide For</i>	49
4.9.	CONTROLE DE PROCESSAMENTO.....	51
4.9.1.	<i>For</i>	51
4.9.2.	<i>Repeat</i>	52
4.9.3.	<i>Accept / Reject</i>	52
4.9.4.	<i>Escape Top/Bottom/Routine</i>	53
4.9.5.	<i>End</i>	54
4.9.6.	<i>Terminate</i>	54
4.9.7.	<i>Stop</i>	55
4.9.8.	<i>Limit</i>	55
4.10.	PROCESSAMENTO DE QUEBRA AUTOMÁTICA.....	55
4.10.1.	<i>At Start Of Data</i>	55
4.10.2.	<i>At End Of Data</i>	55
4.10.3.	<i>At Top Of Page</i>	56
4.10.4.	<i>At End Of Page</i>	56
4.10.5.	<i>At Break</i>	57
4.11.	MODULARIZAÇÃO DO SISTEMA.....	58
4.11.1.	<i>Define Subroutine</i>	58
4.11.1.1.	<i>Perform</i>	59
4.11.1.2.	<i>Callnat</i>	60
4.11.1.3.	<i>Fetch</i>	60
4.11.1.4.	<i>Call</i>	61
4.12.	UTILIZANDO JANELAS.....	61
4.12.1.	<i>Define Window</i>	61
4.12.2.	<i>Set Window</i>	62
4.13.	UTILIZANDO BANCO DE DADOS ADABAS.....	63
4.13.1.	<i>Store</i>	63
4.13.2.	<i>Update</i>	63
4.13.3.	<i>Delete</i>	64
4.13.4.	<i>End Transaction</i>	65
4.13.5.	<i>Backout Transaction</i>	65
4.13.6.	<i>Read</i>	65
4.13.7.	<i>Find</i>	67
4.13.8.	<i>Get</i>	69
4.13.9.	<i>Histogram</i>	69
4.14.	UTILIZANDO BANCO DE DADOS DB2/SQL.....	71
4.14.1.	<i>Insert</i>	71
4.14.2.	<i>Select</i>	71
4.14.3.	<i>Update</i>	73
4.14.4.	<i>Delete</i>	74
4.14.5.	<i>Comando WHERE</i>	74
4.14.6.	<i>Cláusula GROUP BY</i>	74
4.14.7.	<i>Cláusula HAVING</i>	75
4.14.8.	<i>Cláusula ORDER BY</i>	75
4.14.9.	<i>Operador IN</i>	75
4.14.10.	<i>Operador BETWEEN .. AND</i>	75
4.14.11.	<i>Operador LIKE</i>	75
4.14.12.	<i>Operador IS NULL</i>	76
4.14.13.	<i>Função AVG</i>	76
4.14.14.	<i>Função COUNT</i>	76
4.14.15.	<i>Funções MIN, MAX</i>	76
4.14.16.	<i>Função SUM</i>	76

4.14.17. Comando COMMIT.....	76
4.14.18. Comando ROLLBACK.....	76
5. LISTA DE EXERCÍCIOS	77
5.1. EXERCÍCIOS DE FIXAÇÃO EM NATURAL.....	77
5.2. EXERCÍCIOS EXTRAS - ALGORITMO.....	103
5.3. OUTROS EXERCÍCIOS – LÓGICA (INTERNET).....	105
6. BIBLIOGRAFIA	106
7. LICENÇA PÚBLICA GERAL GNU.....	107
7.1. LICENÇA PÚBLICA GERAL GNU (TRADUÇÃO).....	107
7.2. GNU FREE DOCUMENTATION LICENSE (ORIGINAL EM INGLES)	113

1. Introdução ao Natural

O Natural é uma ferramenta de desenvolvimento de 4ª geração desenvolvida pela Software AG (Alemanha) e distribuída no Brasil pela Consist. Além de possuir a versão para MVS, objeto deste curso, possui versões para UNIX, OS/2 e Windows.

Algumas características do Natural são:

- Linguagem de programação com acesso a diversos bancos de dados: ADABAS (Hierárquico e relacional), Oracle, Db2, etc.;
- Utiliza diversos editores para criar programas, funções, telas pré-formatadas e áreas de dados;
- Permite uma programação modularizada;
- Utiliza funções e variáveis do sistema;
- Permite execução on-line e batch;
- Possui utilitário de testes de programação;
- Possui utilitário de criação/manutenção de mensagens de erros de aplicação;
- Qualquer aplicação pode ser facilmente portada para várias outras plataformas.

Os objetos Natural (programas, mapas, áreas de dados, etc.) são armazenados em bibliotecas ("Libraries"), com estrutura parecida com o diretório DOS e podem ter 8 caracteres como nome máximo. Mesmo sendo objetos de diferentes tipos, não podem possuir o mesmo nome.

As principais vantagens do NATURAL em relação às linguagens tradicionais são:

- Maior facilidade de acesso a Banco de Dados (ADABAS, DB2, ORACLE, etc...);
- Maior velocidade de programação;
- Maior dinamismo em testes e manutenção de programas; e
- Menor preocupação com os demais recursos da instalação.

1.1. Componentes

O Natural tem os seguintes componentes principais:

- Núcleo → executa os comandos e instruções e gerência os demais componentes;
- Compilador → de execução interativa;
- Módulos de interface:
 - Com o Sistema Operacional → executa todos os comandos de entrada/saída e demais comunicações;
 - Com o monitor de TP → gerência todos os recursos necessários à aplicação;

- Com o ADABAS → executa todos os comandos ADABAS e controla vários processos de acessos/atualizações.
- System File → arquivo ADABAS de suporte ao Natural.

1.2. System File

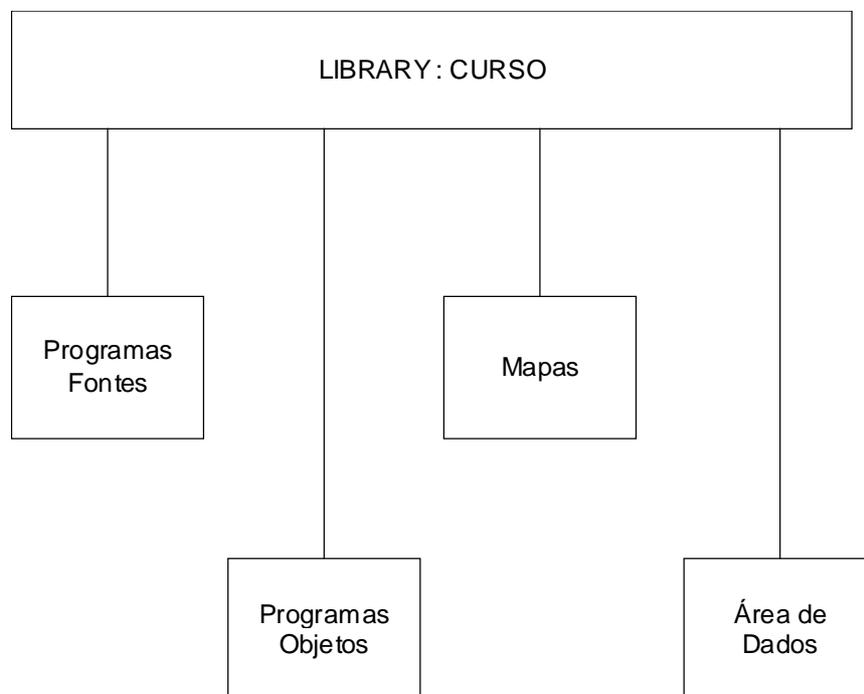
O System File é um arquivo ADABAS reservado para uso do Natural, que contém:

- Todos os programas Natural, tanto em formato fonte (programas) como em formato objeto (compilado), agrupados em bibliotecas;
- Os módulos de definição de arquivos, ou DDM's (Data Definition Modules), com a definição para o Natural dos arquivos ADABAS e suas userviews;
- As mensagens de erro do Natural;
- Os textos da função "Help".

1.3. Bibliotecas

São formadas por um conjunto de programas fontes e/ou carga, geralmente pertencentes ao mesmo sistema.

Em uma sessão Natural, as "LIBS" são estabelecidas pelo comando "LOGON". As chamadas de programas a programas são feitas dentro da mesma biblioteca.



1.4. Programas

Os programas Natural, fontes e objetos são identificados por nomes com formação igual aos nomes das bibliotecas, ou seja, até 8 caracteres, sendo o primeiro alfabético. Fontes e objetos podem (mas não devem) Ter nomes diferentes.

Os programas fontes são formados por linhas de até 72 posições, numeradas por 4 dígitos. Esta numeração é gerada pelo Natural durante a criação do programa, e é usada tanto pelo compilador e editores do Natural como também tem importantes funções lógicas dos programas.

Dentro das linhas, a disposição das instruções (statements ou comandos de programas) é livre, não havendo instruções nem parâmetros posicionais.

Comentários podem ser incluídos e duas formas:

- Toda linha será considerada um comentário se contiver “*” ou “**” nas duas primeiras posições;
- A parte de uma linha à direita de “/*” será considerada um comentário.

Exemplo:

```
0010 * Estas duas linhas (0010 e 0020)
0020 ** são comentários.
0030 FORMAT LS=80 /* Bem como esta parte da linha (0030)
.
.
0200 END
```

As instruções “END” ou “.” indicam fim de programa.

1.5. Interface Natural

Ao entrar no Natural, nos é apresentada a seguinte tela:

```
17:52:21          ***** NATURAL *****          03/01/2003
User X121212          - Main Menu -          Library CURSO

                Function
                _ Development Functions
                _ Development Environment Settings
                _ Maintenance and Transfer Utilities
                _ Debugging and Monitoring Utilities
                _ Example Libraries
                _ Other Products
                _ Help
                _ Exit NATURAL Session

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                Help                Exit                                Canc
```

Informações da tela:

- User → Chave do usuário ativo.
- Library → Biblioteca atualmente selecionada.

Entrando na *Development Functions*, temos acesso à área de programação do Natural, que permite as seguintes opções:

- C → Cria um objeto
- E → edita um objeto
- R → Permite alterar nome de um objeto
- D → Deleta um objeto
- X → Executa um programa (único objeto que permite execução direta)
- L – Lista fonte dos objetos
- S – Lista subrotinas utilizadas
- ? – Help
- . – Sair do menu *Development Functions*

```

17:52:50          ***** NATURAL *****          03/01/2003
User X028862      - Development Functions -          Library CURSO
                                                         Mode Structured
                                                         Program PERPGRAF

                Code  Function
                C    Create Object
                E    Edit Object
                R    Rename Object
                D    Delete Object
                X    Execute Program
                L    List Object(s)
                S    List Subroutines Used
                ?    Help
                .    Exit

                Code .. _   Type .. _
                Name .. _____

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit                                     Canc
    
```

1.6. DDM

DDM's são módulos que descrevem um arquivo ADABAS e seus campos, ou uma userview de um arquivo ADABAS. As DDM's são identificadas por um nome de até 32 caracteres, nome este também usado para identificar os arquivos nos programas Natural.

Em um programa Natural todas as características dos campos de banco de dados são retiradas das DDM's não havendo a necessidade de se definir esses campos nos programas.

O processo de obtenção das características dos campos é feito em tempo de compilação, e elas são agregadas aos programas quando estes são catalogados; portanto, os programas em formato objeto não utilizam as DDM's.

A estrutura de DDM's é mostrada a seguir:

```

10:53:12          ***** NATURAL LIST COMMAND *****          03/01/2003
User X121212          - List DDM VEHICLES -          Library CURSO

DDM DBID 0          DDM FNR 119          VSAM Name          Default Sequence          Page 1

T L DB Name          F Leng S D Remarks
- - - - -
1 AA REG-NUM          A 15 N D CAR'S REGISTRATION N
1 AB CHASSIS-NUM      I 4 F
1 AC PERSONNEL-ID     A 8 D CNNNNNNN
G 1 CD CAR-DETAILS    A 20 N D DESCRIPTION OF THE C
2 AD MAKE             A 20 N
2 AE MODEL            A 10 N D
1 AG YEAR             N 4,0 N YEAR OF THE CAR'S MA
1 AH CLASS            A 1 F D P=PRIVAT
1 AI LEASE-PUR        A 1 F L=LEASED
1 AJ DATE-ACQ         N 8,0 N DATE WHEN CAR WAS AC
1 AL CURR-CODE        A 3 N CURRENCY OF MAINTENA
M 1 AM MAINT-COST     P 7,0 N MAINTENANCE COST IN
1 AN DAT-ACQ-DESC     B 4 N S CAR AGE BASED ON DAT
* ----- SOURCE FIELD(S) -----
* DATE-ACQ(5-6)
* DATE-ACQ(3-4)
1 AO MODEL-YEAR-MAKE  A 22 N S MODEL YEAR + CAR MAK
* ----- SOURCE FIELD(S) -----
* YEAR(1-2)
* MAKE(1-20)
    
```

```

11:03:13          ***** NATURAL LIST COMMAND *****          03/01/2003
User X121212          - List DDM DB2TST-CADASTRO_CURSO -          Library CURSO

DDM DBID 250        DDM FNR 1          VSAM Name          Default Sequence ?          Page 1

T L DB Name          F Leng S D Remarks
- - - - -
1 OA CD_USU          A 11 D
1 AA COD_CPF_CGC     P 14,0 D
1 AB NOM             A 60 D
1 AC NR_SEQ_L_END     I 2 D
1 AD SEXO            A 1 D
1 AE NVL_ECLD        I 2 D
1 AF DT_ATL          A 10 D
    
```

1.7. Objetos Natural

Um objeto Natural nada mais é do que um componente de um sistema. Os diversos tipos de objetos Natural são:

- **COPYCODE** → Peçaço de código fonte, que pode ser aproveitado em vários programas. Único objeto Natural que não é compilado. Nos programas aonde o COPYCODE vai ser aproveitado, através do comando INCLUDE, ele será compilado, juntamente com o programa.
- **GLOBAL** → Objeto utilizado para definição de variáveis. Todos os objetos que se referenciam a uma GLOBAL, terão as mesmas variáveis, e estas manterão os valores após o término de um objeto, permitindo que um objeto altere valores e outro reconheça as alterações.
- **LOCAL** → Objeto utilizado para definição de variáveis. A diferença entre uma LOCAL e uma GLOBAL é que valores das variáveis locais somente serão reconhecidos pelo objeto onde foram declaradas, deixando de existir quando do término do mesmo.
- **PARAMETER** → Objeto utilizado para definição de variáveis que serão utilizadas entre programas e subprogramas.
- **MAP** → Utilizado para entrada e exibição de dados. Permite a definição de Regras de Validação para os campos, inclusive com acesso à Base de Dados.
- **Helproutine** → Tipo de objeto que permite ser definido como ajuda a um campo de entrada de dados. Quando o cursor estiver posicionado num campo e for teclado "?", será chamada a HELPROUTINE que estiver associada ao campo.
- **PROGRAM** → Principal objeto Natural. É a partir dele que são chamados os outros componentes.
- **SUBPROGRAM (**N**)** → Objeto chamado por outro, com passagem de parâmetros, que permite o retorno de dados ao objeto chamador.
- **SUBROUTINE** → Uma sub-rotina é um pedaço de código geralmente executado mais de uma vez ou de diversas partes de um programa. Pode ser interno, definido dentro de um programa ou fora dele (externo). Quando se escolhe a segunda opção, o tipo de objeto que conterà as sub-rotinas é o SUBROUTINE.

As letras sublinhadas nos nomes dos objetos podem ser utilizadas logo após o comando EDIT, evitando a digitação completa do tipo de objeto.

2. Sessão Natural

2.1. Comandos de uma Sessão Natural

Os comandos do sistema Natural são identificados pela mensagem na linha de comando direto do Natural. São utilizados para criação, modificação e execução de programas. Os comandos podem ser os seguintes:

- Comandos de controle de bibliotecas
- Comandos de manipulação de programas fontes
- Comandos de manipulação de programas objetos
- Comandos de manipulação mista de programas (fontes / objetos)
- Comandos de transferência / limpeza de Work-Area
- Comandos de execução de programas
- Comandos de processos especiais

2.1.1. Comandos de Controle de Bibliotecas

LOGON

As bibliotecas são estabelecidas pela instrução 'LOGON'

LOGON | *biblioteca* |

FIN

A instrução 'FIN' encerra uma sessão Natural, devolvendo o controle ao sistema de onde o Natural foi invocado.

LIST

A instrução 'LIST' invoca um utilitário para listagem de diretórios de bibliotecas (fontes/objetos), listagem de programas e de DDM's.

Pode-se chamar um 'MENU' das várias funções do LIST através do formato 'LIST *', ou chamar diretamente a função, como veremos a seguir:

<i>L(IST)</i>	<i>Object-Type</i>	<i>Object-Name</i>
LIST FILE	FILE-NAME	→ DDM
LIST P(ROGRAM)	PROGRAM-NAME	→ Programa
LIST M(AP)	MAP-NAME	→ Mapa
LIST S(UBROUTINE	SUBROUTINE-NAME	→ Subrotina
LIST SUBPROGRAM ou N	SUBPROGRAM-NAME	→ Subprograma
LIST H(ELPROUTINE)	HELPROUT-NAME	→ Rotina de Help
LIST G(LOBAL)	GLOBAL-NAME	→ Global Data Área
LIST L(OCAL)	LOCAL-NAME	→ Local Data Área
LIST PARAM(ETER)	PARAMETER-NAME	→ Parameter Data Área
LIST C(OPYCODE)	COPYCODE-NAME	→ Copycode

Exemplo: LIST * → Lista todos os tipos de objetos

```

11:06:06          ***** NATURAL LIST COMMAND *****          03/01/2003
User X121212      - LIST Objects in a Library -          Library CURSO

Cmd  Name          Type          S/C  SM  Version  User ID  Date          Time
---  *              *              *    *  *        *        *              *
___  A              Program       S    S   3.1.04   X117566  2002-12-16   10:45:34
___  AAA            Program       S    S   3.1.04   X087789  2002-12-18   20:01:45
___  AAA01          Local        S/C           3.1.04   I017641  2002-11-04   16:49:54
___  AAA02          Program      S/C  S   3.1.04   X117566  2002-12-13   14:03:16
___  AADS           Program      S/C  R   3.1.04   X116213  2002-12-18   09:43:44
___  AAFGDA         Global       S/C  R   3.1.04   A431601  2002-11-06   16:22:32
___  AAFLAY         Map          S/C  R   3.1.04   A431601  2002-11-07   11:38:49
                                           14 Objects found

Top of List.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Print Exit          --  -  +  ++          >  Canc
    
```

EDIT

A instrução *'EDIT'* exibe a cópia do programa (ou outro tipo de objeto) na *'work-area'*, ficando inalterado o original no system file até que seja dado um comando de atualização (*'SAVE'* ou *'STOW'*).

Pode-se chamar um *'MENU'* das várias funções do *LIST* através do formato *'EDIT *'*, ou chamar diretamente a função, como veremos a seguir.

```

E(DIT)  | Object-Type | | Object-Name |
E(DIT)  | Object-Type |  * → lista tudo referente ao 'object-type'
    
```

Se não for especificado o *'object-name'*, será editado o conteúdo existente na *'work-área'*; se não houver nenhum conteúdo na *'work-área'*, o modo de edição permitirá a criação de um programa (ou qualquer tipo dos objetos especificados) sem o controle do compilador.

```

EDIT P(ROGRAM)      | PROGRAM-NAME | → Programa
EDIT M(AP)          | MAP-NAME     | → Mapa
EDIT S(UBROUTINE)  | SUBROUTINE-NAME | → Subrotina
EDIT SUBPROGRAM ou N | SUBPROGRAM-NAME | → Subprograma
EDIT H(ELPROUTINE) | HELPROUT-NAME | → Rotina de Help
EDIT G(LOBAL)      | GLOBAL-NAME  | → Global Data Área
EDIT L(OCAL)       | LOCAL-NAME   | → Local Data Área
EDIT PARAM(ETER)  | PARAMETER-NAME | → Parameter Data Área
EDIT C(OPYCODE)    | COPYCODE-NAME | → Copycode
    
```

2.1.2. Comandos de Manipulação de Programas Fontes

CHECK

Checa a sintaxe do objeto atualmente na área de trabalho. Pode ser substituído por apenas 'C'.

SAVE

Salva o programa fonte. Pode-se especificar o nome desejado logo após o comando. Pode ser substituído por apenas "SA".

PURGE

O comando '*PURGE*' é usado para deletar um ou mais programas fontes do Natural System File. O conteúdo existente em work-área não é alterado.

Se o *program-name* não for especificado, será exibido um mapa com os programa fontes existentes no Natural System File, para especificação individual.

2.1.3. Comandos de Manipulação de Programas Objetos

CATALOG

Cataloga o programa fonte da área de trabalho. Este comando checa a sintaxe e grava a versão executável do programa. Não grava o fonte. Pode ser substituído por apenas 'CAT'.

UNCATALOG

Apaga apenas o módulo carga, mantendo o módulo fonte.

2.1.4. Comandos de Manipulação Mista de Programas

STOW

Checa a sintaxe do programa fonte da área de trabalho, salva e cataloga. Este comando executa internamente as funções dos comandos *CHECK*, *SAVE* e *Catalog*. Porém, ele somente salva o programa fonte se este não contiver nenhum erro.

SCRATCH

Apaga o fonte e o módulo carga informado logo após o comando. Pode ser substituído por 'SCR'.

2.1.5. Comandos de Transferência/Limpeza de Work-Area

READ

Lê e transfere para a área de trabalho o programa fonte definido após o comando.

CLEAR

Limpa a área de trabalho.

RENUMBER (n)

Renumeras as linhas do programa fonte que estiver na *work-area*, em incrementos de 10.

SCAN

Pesquisa um determinado 'STRING' dentro de um ou de todos os objetos de uma biblioteca.

2.1.6. Comandos de Execução de Programas

RUN

Executa o programa atualmente da área de trabalho. O comando compila o programa antes de executá-lo. Pode ser substituído por apenas "R".

2.1.7. Comandos de Processos Especiais

GLOBALS

Exibe menu que está definido com uma série de ações e atributos que afetam tanto o resultado dos comandos como o resultado das instruções.

As principais funções dos parâmetros são:

- Definir característica dos relatórios e mapas;
- Estabelecer limites p/quantidades de registros lidos, de páginas de relatórios, etc.;
- Definir caracteres de controle.

```

11:23:25          ***** NATURAL GLOBALS COMMAND *****          03/01/2003
                    - Session Parameters -

(CC) Cond.Prog.Execution ..... OFF  (MT) Max.CPU Time ..... _____0
(CF) Term.Control Character .... %  (NC) Nat.Sys.Commands ..... OFF
(DC) Dec. Character ..... ,  (OPF) Overw.Prot.Fields ..... OFF
(DFOUT) Date Format Output ..... I  (PD) Page Dataset ..... _50
(DFSTACK) Date Format Stack Cmd .. S  (PM) Print Mode ..... RP
(DFTITLE) Date Format Title ..... S  (PS) Page Size ..... _23
(DU) Dump Generation ..... OFF  (REINP) Reinput on Error ..... ON
(EJ) Page Eject ..... ON  (SA) Sound Alarm ..... OFF
(FS) Default Format ..... ON  (SF) Spacing Factor ..... _1
(FCDP) FC on Dyn.Prot.Fields ... ON  (SL) Source Line Length ..... 72
(IA) Input Assign ..... =  (SM) Structured Mode ..... OFF
(ID) Input Delimiter ..... ;  (TS) Translate Sys.Prog ..... OFF
(IM) Input Mode ..... F  (WH) Wait on Hold ..... OFF
(LE) Limit Error ..... OFF  (ZD) Zero Division ..... ON
(LS) Line Size ..... _80  (ZP) Zero Printing ..... ON
(LT) Limit ..... __99999999

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Print Exit          Updat                                Canc
    
```

HELP

Permite a exibição de mensagens de erro e comandos do Natural. Também pode ser acesso teclando "?". Ambas as opções permitem a passagem do código diretamente (? 3148, por exemplo, exibe o conteúdo do código de erro 3148).

PROFILE

Mostra a profile de segurança corrente para o usuário.

```

11:24:56          ***** NATURAL EDITORS *****          03/01/2003
                    - Editor Profile -

Profile Name .. X121212_

PF and PA Keys
PF1 ... HELP_____ PF2 ... SC=_____ PF3 ... EXIT_____
PF4 ... _____ PF5 ... _____ PF6 ... _____
PF7 ... -_____ PF8 ... +_____ PF9 ... _____
PF10 .. SC=_____ PF11 .. _____ PF12 .. CANCEL_____
PF13 .. _____ PF14 .. _____ PF15 .. MENU_____
PF16 .. _____ PF17 .. _____ PF18 .. _____
PF19 .. --_____ PF20 .. ++_____ PF21 .. _____
PF22 .. _____ PF23 .. _____ PF24 .. _____
PA1 ... _____ PA2 ... SCAN_____ PA3 ... _____

Automatic Functions
Auto Renumber .. Y   Auto Save Numbers .. 0__   Source Save into .. EDITWORK

Additional Options .. N

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit  AddOp Save Flip      Del  Canc
    
```

2.2. Editores para Objetos

2.2.1. Editor de Áreas de Dados

- **GLOBAL DATA AREA (GDA)**
 - Objeto utilizado para definição de variáveis. Todos os objetos que se referenciam a uma GLOBAL, terão as mesmas variáveis, e estas manterão os valores após o término de um objeto, permitindo que um objeto altere valores e outro reconheça as alterações.
 - Somente uma GDA pode ser usada ao mesmo tempo.
 - Se uma GDA for recatalogada, todos os objetos que a utilizam devem ser recatalogados.
 - A GDA já deve existir quando um programa for ser catalogado ou executado.

- **PARAMETER DATA AREA (PDA)**
 - Objeto utilizado para definição de variáveis que serão utilizadas entre programas e subprogramas.
 - Os dados na PDA devem Ter o mesmo tamanho/formato e seqüência que no objeto chamador, mas não precisam Ter o mesmo nome.
- **LOCAL DATA AREA (LDA)**
 - Objeto utilizado para definição de variáveis. A diferença entre uma LOCAL e uma GLOBAL é que nos valores das variáveis locais somente serão reconhecidos pelo objeto onde foram declaradas, deixando de existir quando do término do mesmo.
 - A LDA é obrigatória em modo estruturado e opcional em report.

A tela a seguir mostra o editor de área de dados:

Local	XXXL0001	Library	CURSO	DBID	29	FNR	4
Command							
I T L	Name	F	Leng	Index/Init/EM/Name/Comment			
All	-	-	-	-	-----		
V	1	EMPL		EMPLOYEES			
	2	PERSONNEL-ID	A	8	/*	CNNNNNNN	
	2	FIRST-NAME	A	20	/*	FIRST/CHRISTIAN NAME	
	2	NAME	A	20	/*	SURNAME/FAMILY NAME	
	2	CITY	A	20	/*	MAIN CITY/TOWN	
	2	JOB-TITLE	A	25			
	2	SALARY	P	9,0	(1:3)	/*	ANNUAL SALARY
M	2	BONUS	P	9,0	(1:3,1:2)	/*	BONUS PER YEAR

2.2.1.1. Define Data

Para a definição de variáveis, utilizamos a instrução DEFINE DATA. Deve ser o primeiro comando de um programa. Somente pode existir uma DEFINE DATA por objeto. As definições podem ser internas ou externas.

Definições internas:

```
INDEPENDENT
LOCAL
PARAMETER
```

Definições externas:

```
GLOBAL USING XXXG0001
```

```
LOCAL USING XXXL0001
LOCAL USING XXXA0001
```

Quando múltiplas áreas de dados são utilizadas pelo mesmo programa, elas devem ser codificadas na seguinte ordem:

- Global
- Parameter
- Local
- Independent

Após a definição da última variável, fechar o laço com END-DEFINE. Exemplo:

```
DEFINE DATA LOCAL
1 #CD-MATR-FUNC          (I4)
1 #NM-FUNC              (A35)
1 #TELEFONE
  2 #DDD                (N3)
  2 #NUMERO              (N8)
1 #VL-SAL-FUNC          (P11,2)
1 #TAB-MÊS              (A3/12)
END-DEFINE
```

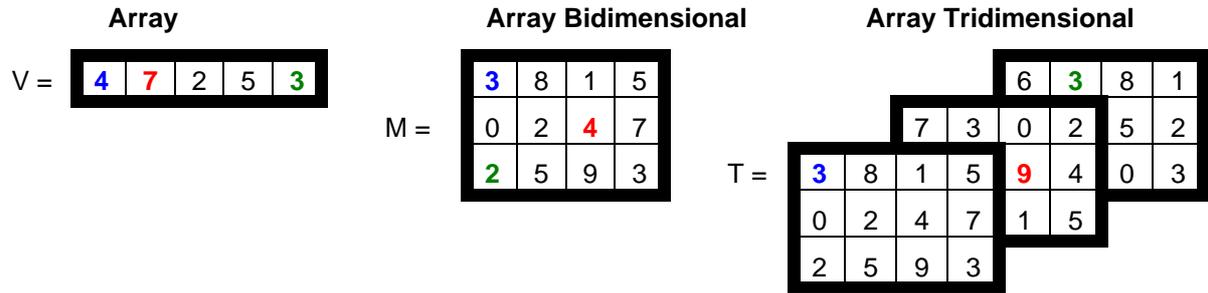
A estrutura para definição é composta de:

- Nível
- Nome
- Formato/tamanho
- Complemento

Podemos definir até 7 níveis de variáveis. O tamanho máximo para um nome de variável é 32 caracteres. Os caracteres válidos para formação do nome são os alfabéticos (A-Z), números (0-9) e os especiais (#, -, _, @, \$, &). O caracter + também pode ser utilizado, desde que apenas na primeira posição. Para efeito de padronização, utilizamos o caracter “#” como caracter inicial para designar variáveis definidas pelo usuário.

Formato	Código	Tamanho Máximo	Observação
Alfanumérico	A	253	
Numérico	N	27	Máximo de 7 decimais
Compactado	P	27	Máximo de 7 decimais
Inteiro	I	1, 2 ou 4	I1 - valores de 1 até 127 I2 - valores de 1 até 32767
Data	D		
Hora	T		
Lógico	L		TRUE ou FALSE
Binário	B	126	
Controle	C		Atributo de controle de mapas
Ponto Flutuante	F	4, 8	

Array



Cada elemento dos arrays pode ser referenciado através de índices. Exemplos:

V(L) -----	M(L,C) -----	T(L,C,T) -----
V(1) := 4	M(1,1) := 3	T(1,1,1) := 3
V(2) := 7	M(2,3) := 4	T(2,3,2) := 9
V(5) := 3	M(3,1) := 2	T(1,2,3) := 3

Onde: **L** = Linha, **C** = Coluna e **T** = Tabela.

Para definir a ocorrência de múltiplos elementos, basta colocar uma barra, seguida do número de vezes que o campo deve ser repetido. Opcionalmente, podemos definir 1:n como quantidade. Variáveis de qualquer tipo podem ser transformadas em matriz.

JAN	FEV	MAR	...	DEZ
-----	-----	-----	-----	-----

```

1 #MESES-ANO           (A3/12)
1 #DIAS-SEMANA        (A7,1:7)
    
```

Redefinição

Podemos redefinir um campo, ou parte dele, em campos menores, de acordo com a necessidade. Os campos de menor nível podem ser de formato diferente, desde que tenham o mesmo tamanho.

DATA		
DIA	MÊS	ANO

```

1 #DT-NASC           (A8)
1 REDEFINE #DT-NASC
2 #DD-NASC           (A2)
2 #MM-NASC           (A2)
2 #AA-NASC           (A4)
    
```

Valores iniciais

A atribuição de valores iniciais pode ser feita quando da definição da variável. Durante a execução do programa, pode-se atribuir novos valores às mesmas. Quando a variável for alfanumérica, o valor inicial deve vir entre aspas simples.

```

1 #DT-ADM          (A8)  INIT<'28121945'>
1 REDEFINE #DT-ADM
  2 #DD-ADM        (A2)
  2 #MM-ADM        (A2)
  2 #AA-ADM        (A4)
1 #IN-CPF          (L)   INIT<FALSE>
1 #CPF             (N14) INIT<75825641034>
    
```

Não se pode atribuir valores a uma variável que é redefinição de outra, mas apenas à variável de nível acima dela.

Registros

Um registro é uma coleção de variáveis, com características em comum. O Natural permite a definição de registros, bastando para isso criar variáveis com níveis diferentes.

```

1 #REG-SAIDA
  2 #CD-FUNC          (N9)
  2 #NM-FUNC          (A35)
  2 #DT-NASC
    3 #DD-NASC        (A2)
    3 #MM-NASC        (A2)
    3 #AA-NASC        (A4)
  2 #CD-SEXO          (A1)
    
```

2.2.2. Editor de Mapas

Após escolher a opção “D” no menu inicial, digitar EDIT MAP (ou simplesmente E M) e será exibido o submenu do editor de mapas:

```

08:41:21          ***** NATURAL MAP EDITOR *****          03/01/2003
User X121212          - Edit Map -          Library CURSO

          Code      Function
          -----
          D          Field and Variable Definitions
          E          Edit Map
          I          Initialize new Map
          H          Initialize a new Help Map
          M          Maintenance of Profiles & Devices
          S          Save Map
          T          Test Map
          W          Stow Map
          ?          Help
          .          Exit

          Code .. I      Name .. _____      Profile .. SYSPROF_

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Help          Exit Test Edit
    
```

O Natural pressupõe a criação de um mapa novo: por esse motivo, CODE já vem preenchido com "I". Se formos Editar, etc. um mapa, devemos trocar o código.

Opções:

- "D" → Exibe todos os campos definidos no mapa, com respectiva posição, formato, nome e outras informações.
- "E" → Edita um mapa já existente . Se for informado EDIT MAP NOME, o mapa já será carregado.
- "I" → Inicializa um novo mapa. Exibe uma tela de parâmetros para construção de um novo mapa. No caso de edição de um mapa já existente, a tela de parametrização não é exibida automaticamente.
- "H" → Inicializa um novo mapa de help.
- "M" → Manutenção de profiles. Profiles são mapas que podem ser usados como modelo de novos mapas.
- "S" → Salva o mapa que estiver na área de trabalho.
- "T" → Testa o mapa. Permite visualizar o mapa e simular seu preenchimento.
- "W" → Gera o módulo objeto e salva o módulo fonte do mapa (STOW).
- "?" → Help
- "." → Sai do sub-menu de mapas.

Se formos inicializar um novo mapa, será exibida a seguinte tela:

```

17:58:39          Define Map Settings for MAP          03/01/2003

Delimiters          Format          Context
-----
Cls Att CD  Del  Page Size ..... 23      Device Check .... _____
T  D      BLANK  Line Size ..... 79      WRITE Statement  _
T  I      ?      Column Shift ... 0 (0/1)    INPUT Statement  X
A  D      _      Layout ..... _____  Help _____
A  I      )      dynamic ..... N (Y/N)    as field default N (Y/N)
A  N      ^      Zero Print ..... N (Y/N)
M  D      &      Case Default ... UC (UC/LC)
M  I      :      Manual Skip .... N (Y/N)    Automatic Rule Rank 1
O  D      +      Decimal Char ... ,      Profile Name .... SYSPROF
O  I      (      Standard Keys .. N (Y/N)
                          Justification .. L (L/R)
                          Print Mode ..... _
                          Control Var .... _____

                          Filler Characters
                          -----
                          Optional, Partial ....
                          Required, Partial ....
                          Optional, Complete ...
                          Required, Complete ...

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help          Exit
    
```

Delimiters

Os delimitadores servem para informar ao Natural qual tipo de campo e seus atributos serão definidos a seguir:

Tipos de Campos	
T	Texto constante
A	Campo somente de entrada
M	Campo modificável. Serve para entrada e saída de dados
O	Campo somente de saída

Atributos de Campos			
B	Blinking (piscante)	V	Modo de vídeo reverso
C	Cursivo / Itálico	U	Sublinhado (Underline)
D	Default (cor normal)	Y	Dinâmico
I	Intensificado (high light)	N	Não exibido

Caracteres Default de Prefixo de Campos		
Campo	Atributo	Caracter Prefixo
T	D - Default	Espaço em branco
T	I - Intensificado	?
A	D - Default	_
A	I - Intensificado)
A	N – Neutro (Não exibido)	¬
M	D - Default	&
M	I - Intensificado	:
O	D - Default	+
O	I - Intensificado	(

Format

Especifica características de entrada/saída do mapa.

Page Size	Número de linhas do mapa. Pode variar de 1 a 250.
Line Size	Número de linhas do mapa. Pode variar de 5 a 249.
Column Shift	Se o conteúdo for 1, o mapa é deslocado uma coluna para a esquerda em tempo de execução.
Layout	Nome do mapa que contém o layout pré-definido para o mapa
Dynamic	Se especificado "Y", os mapas são combinados em tempo de execução. Senão, são combinados durante a inicialização.
Zero Print	Se especificado "Y", campos zerados serão exibidos como "0". Senão, campos zerados serão mostrados como espaços.
Case Default	"UC" indica que todos os campos terão as letras minúsculas transformadas em maiúsculas; "LC", as letras minúsculas permanecem em minúsculas.
Manual Skip	Se especificado "Y", o cursor não passará para o próximo

	campo Quando ele estiver todo preenchido. Com "N", o cursor passará para o campo seguinte assim que o campo atual estiver completo.
Decimal Character	Caracter a ser utilizado como separador das casas decimais.
Standard Keys	Se informado "Y", nas duas últimas linhas serão exibidas as teclas de função.
Justification	Se informado "L", todos os campos serão justificados a esquerda (LEFT). Se informado "R", à direita (RIGHT).
Print Mode	Modo de impressão do campo: "I" → Invertido "C" → Utiliza caracteres alternativos "N" → Campo não será impresso "D" → Mode de caracter duplo-byte.
Control Variable	Variável de controle para todos os campos que sejam controlados dinamicamente e que não tenham variável de controle definida.

Context

Especifica o contexto do mapa.

Device Check	Se um nome de um dispositivo é informado neste campo, todos os atributos do mapa serão checados para verificar compatibilidade com o dispositivo.
Write Statement	Marcando este campo, o mapa será usado apenas para saída de dados.
Input Statement	Marcando este campo, o mapa será usado para entrada/saída de dados.
Help	Nome da helproutine associada ao mapa
As field default	Se especificado "Y", o nome do campo onde for pressionada a tecla help será passado como parâmetro default para a helproutine.
Automatic Rule Rank	Define o número default de uma regra de validação automática.
Profile Name	Nome da profile ativa para inicialização do mapa.

Filler Characters

Especifica caracteres de preenchimento para os campos.

Optional, Partial	Opcional, não precisa ser preenchido completamente.
Required, Partial	Preenchimento requerido, não precisa ser preenchido completamente.
Optional, Complete	Opcional, deve ser preenchido completamente.
Required, Complete	Preenchimento requerido, deve ser preenchido completamente.

Após a alteração dos parâmetros desejados, aparece a tela de edição de mapas:

```

Ob _          Ob D CLS ATT DEL      CLS ATT DEL
.            .   T  D   Blnk    T  I   ?
.            .   A  D   _       A  I   )
.            .   A  N   -       M  D   &
.            .   M  I   :       O  D   +
.            .   O  I   (
.            .
001  --010---+-----+-----030---+-----+-----050---+-----+-----070---+-----

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Mset Exit Test Edit -- - + Full < > Let
    
```

Áreas do Editor:

1. Área de split. Permite visualizar e inserir variáveis de views e de outros objetos.
2. Áreas de prefixos. Mostram quais são os prefixos associados a cada tipo de campo.
3. Área de edição. Permite definir o layout do mapa.

Para elaborarmos um mapa, basta digitarmos os campos desejados, precedidos ou não do prefixo correspondente, de acordo com critérios pré-estabelecidos.

Para criarmos um campo alfanumérico modificável, por exemplo, basta digitarmos o código “&”, seguido do número de X correspondentes ao desejado. Outra forma é digitar &X(*qtde*).

Para criarmos um campo numérico, digitar o código, seguindo de 9. Exemplo: &99999.

A entrada de campos do tipo data não é possível diretamente da tela, sendo necessário digitarmos a data num campo alfa ou numérico e depois movermos para um campo DATE.

```

Ob _          Ob D CLS ATT DEL      CLS ATT DEL
.            .   T  D   Blnk    T  I   ?
.            .   A  D   _       A  I   )
.            .   A  N   -       M  D   &
.            .   M  I   :       O  D   +
.            .   O  I   (
.            .
001  --010---+-----+-----030---+-----+-----050---+-----+-----070---+-----

                ?Consórcio?de?Veículos

?Número:&9999999999

?Nome...:&XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

?Valor.:&999999999,99
    
```

Enter	PF1	PF2	PF3	PF4	PF5	PF6	PF7	PF8	PF9	PF10	PF11	PF12
Help	Mset	Exit	Test	Edit	--	-	+	Full	<	>	Let	

Teclas de função na tela de edição de mapas:

Teclas de Função			
PF1	Invoca o help do editor de mapa	PF7	Move metade da página para baixo
PF2	Permite alterar definições do mapa	PF8	Move metade da página para cima
PF3	Retorna ao menu do editor	PF9	Permite edição em tela cheia
PF4	Testa o mapa corrente	PF10	Move a janela para a esquerda
PF5	Edita o campo sob o cursor	PF11	Move a janela para a direita
PF6	Topo – move para o início do mapa	PF12	Ignora as mudanças após o ENTER

Para colocarmos um campo diretamente de uma view ou de um programa, basta na área de split digitar “V” e o nome da view para exibir os respectivos campos, ou “P” e o nome do programa.

Os caracteres + e – permitem movimentar a tela de split para cima ou para baixo.

Para inserirmos o campo desejado, colocar na tela o prefixo desejado e o número do campo que aparece na tela de split. Para desativar, colocar espaço em branco.

Comandos de Linha

Os comandos de linha devem ser digitados no início da linha. Alguns comandos de linha não podem ser digitados ao mesmo tempo, pois o Natural não interpreta todos os comandos ao mesmo tempo.

- ..C → Centraliza a linha corrente. Deve Ter sido dado <ENTER> previamente na linha a centralizar.
- ..D → Deleta a linha.
- ..D(n) → Apaga as “n” linhas, contando com a linha atual.
- ..E → Permite a edição de campos em tela cheia.
- ..Fc → Preenche a linha inteira com o caracter “c”.
- ..I → Insere uma linha.
- ..I(n) → Insere “n” linhas.
- ..J → Join. Concatena a linha corrente com a linha de baixo.
- ..M → Move a linha. Colocar o cursor na linha de destino e teclar <ENTER>.
- ..P → Abre o editor de regras de validação do mapa.
- ..Q → Sair do editor de mapas.
- ..R → Duplica a linha.

- ..S → Split. Permite dividir a linha. Posicionar o cursor no ponto de quebra e teclar <ENTER>.

Comandos de Campo

Os comandos de campo devem ser digitados em cima do prefixo do campo.

- .A → Definição de array (matriz). Permite definir o número de ocorrências, matrizes bidimensionais e qual elemento deverá ser o primeiro a ser apresentado.
- .C → Centraliza o campo, levando em consideração os campos adjacentes.
- .D → Deleta o campo.
- .E → Permite editar o campo no modo estendido.
- .M → Move o campo. Posicionar o cursor no novo lugar e teclar <ENTER>
- .R → Repete o campo marcado na posição atual do cursor.
- .P → Permite editar as regras de validação do campo.

Exemplo de edição estendida de campo:

```

Fld #002                                     Fmt A5
-----
AD= MDLT_____ ZP=          SG=          HE= _____ Rls 0
AL= _____ CD=  _      CV= _____ Mod Undef
PM=  _  DF=          DY= _____
EM= _____

001  --010---+-----+-----+---030---+-----+-----+---050---+-----+-----+---070---+-----

Código:&99999
Valor.:.EXXXX

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
HELP Mset Exit <--- ---> -- - + < > Let
    
```

Quando da criação do campo, o Natural estabelece #001 para o primeiro campo e vai nomeando sucessivamente os campos à medida que são criados , seqüencialmente. Para alterar o nome do campo, digitar .E no campo.

Outros campos na tela de edição estendida do campo:

- Fmt → Formato. Permite alterar o formato (N, A, P, D, etc.);
- AD → Definição de atributos do campo. Digitar o atributo diretamente no campo ou teclar “?” para preencher as opções disponíveis.
- ZP → Zero print. Disponível apenas se for campo numérico. Se informado “Y”, imprime 0 (zero) quando for valor zerado. Se informado “N”, o campo é mostrado como espaços em branco.
- SG → Signal. Especifica se o sinal do campo será mostrado ou não. Disponível apenas se for campo numérico.
- HE → Helproutine. Especifica uma Helproutine a ser executado caso seja pressionada a tecla de HELP no campo. Outra forma de ativar o Help é digitar “?” na primeira posição do campo.
- NL / AL → Numeric Length ou Alphabetic Length. Especifica o número de caracteres a serem exibidos. O campo pode ser A9, por exemplo, mas ser necessário mostrar apenas as 4 primeiras: AL=4.
- CD → Color Atributes. Permite atribuir uma cor específica para o campo.
- CV → Control Variable. Variável de controle. Estabelece a variável de controle (tipo C) do mapa.
- PM → Print Mode. I = Invertido, C = Conjunto de caracteres alternativos, N – Sem impressão.
- DY → Atributos Dinâmicos. Permite definir atributos para altera parte do campo.
- EM → Edit Mask. Permite especificar uma máscara para o campo.
 - Máscara para números:
 - 9 → apresenta o número, mesmo que seja zero;
 - Z → suprime os zeros não significativos. Ex.: ZZZ.ZZ9,99
 - S → sinal do campo. Ex.: ZZ0,00S
 - Máscara para campos data:
 - DD → Dia
 - MM → Mês
 - YYYY → Ano. Ex.: DD/MM/YYYY
 - Máscara para campos hora:
 - HH → horas
 - II → minutos
 - SS → segundos. Ex.: HH:II:SS
 - Máscara para campos alfa:
 - X → especifica um caracter alfa. Ex.: XXX-XX

Se for digitado .A num campo, será aberta a tela de edição de array:

Name #002	Upper Bnds 1_____ 1_____ 1_____		

Dimensions	Occurrences	Starting from	Spacing
0 . Index vertical	1__	_____	0 Lines
0 . Index horizontal	1__	_____	1 Columns
0 . Index (h/v) V	1__	_____	0 Cls/Ls

```

001  --010---+-----+-----+-----030---+-----+-----+-----050---+-----+-----+-----070---+-----
                                     &XXXXX
                                     .A9999

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit      --    -    +          <    >      Let
    
```

- Upper Boundaries → estabelece o tamanho máximo de ocorrências do array.
- Occurrences → a linha que for preenchida nesta coluna, estabelece o número de ocorrências verticais e/ou horizontais do array.
- Starting from → permite estabelecer uma variável para controlar qual o número do primeiro elemento da tela em relação ao array definido no programa.

Por exemplo, podemos definir um array de 100 ocorrências internamente e 10 ocorrências sendo exibidas na tela. Definimos então uma variável que terá o número do primeiro elemento. Se a variável tiver o conteúdo 70, serão mostradas as ocorrências de 70 a 80. É útil para controle de paginações.

2.2.3. Editor de Programas

Após escolher a opção “D” no menu inicial, digitar *EDIT PROGRAM* (ou simplesmente E P) e será exibido o editor de programas. O editor de programas possui o seguinte layout:

```

>
All  <-----+-----1.....+-----2.....+-----3.....+-----4.....+-----5.....+-----Mode Structured.
0010
0020      Linha de Comandos
0030
0040
0050
0060
0070
0080
0090
0100
0110
0120
0130
0140
0150
0160
0170
0180
0190
0200
      > + Program XXXP0001      Lib CURSO
      Tipo de Objeto
      Biblioteca      Nome do Objeto
      Modo de Programação
      Informações do fonte
      .....Current Source Size: 0 Char. Free: 70273 .+.....+... S 0      L 1
    
```

Na linha de comandos podemos digitar os seguintes comandos, exclusivos à área do editor de programas:

- NNNN → Traz a linha 'NNNN' para o início.
- ADD → Adiciona linhas no final do fonte.
- TOP ou -- → Posiciona a janela de edição na primeira linha do fonte sendo editado. Pode ser substituído por *T*.
- BOTTOM ou ++ → Posiciona a janela de edição no final do fonte sendo editado. Pode ser substituído por *B*.
- +H → Avança meia página (10 linhas).
- -H → retrocede meia página (10 linhas).
- CHANGE → São passados dois parâmetros: o valor a trocar e o novo valor. O Comando procura a primeira ocorrência do valor e efetua a troca. Caracteres especiais devem ser informados entre aspas simples ('). Na linha em que tiver sido efetuada a troca, na área destinada à numeração das linhas, aparecerá a informação da substituição. Pode ser substituído por *CH*.
- GLOBALS SM=[ON/OFF] → Muda o modo de programação entre estruturado e report.
- LET → Ignora todas as alterações feitas no fonte após o último <ENTER>.
- PROFILE → Exibe e permite alterar as informações de edição da seção atual.
- RESET → Limpa marcas de bloco e qualquer outra informação presente na área de numeração das linhas.
- SCAN → Procura a string no fonte.
- SPLIT <tipo de objeto> <nome do objeto> → Divide a tela em duas partes: na superior, mantém o editor ativo; na inferior, apresenta apenas para visualização o objeto solicitado.

A seguir, a relação dos principais comandos de movimentação de texto do editor de programas. Estes comandos devem ser digitados na primeira posição da linha:

- .C[n] → Copy. Copia a linha atual, duplicando-a. Se especificado um número para *n* entre parênteses, duplica a linha *n* vezes.
- .X e .Y → Marca o bloco.
- .CX-Y[n] → Copia o bloco marcando com .X e .Y. Se especificado um número para *n* entre parênteses, duplica a linha *n* vezes.
- .D[n] → Delete. Apaga a linha corrente. Se especificado um número para *n* entre parênteses, duplica a linha *n* vezes.
- .I[n] → Insert. Insere linhas após a linha corrente. Se especificado um número para *n* entre parênteses, duplica a linha *n* vezes. Se informado *.i[objeto]*, o fonte do objeto será inserido no fonte atual.

- .J → Join. Concatena a linha corrente com a linha seguinte, desde que o tamanho da linha resultante seja suportado pelo editor (127 caracteres por linha).
- .L → Ignora as mudanças da linha, desde que não pressionado <ENTER>.
- .MX-Y[n] → Move. Move o bloco marcado para a linha seguinte à atual.
- .P → Posiciona a linha como sendo a primeira da tela.
- .S → Split. Divide a linha corrente, quebrando-a em duas. Após colocar .S no início da linha, colocar o cursor sob o caracter que deverá ser o primeiro da nova linha e teclar <ENTER>.

Após a marcação de um bloco, aparecem as marcas correspondentes (X e Y) na primeira coluna do fonte. Quando definimos o .X e o .Y para a mesma linha, na primeira coluna é exibido Z.

Quando existir um bloco marcado, podem ser digitados os comando DX-Y na linha de comandos para apagar o bloco e Ex-y para editar apenas o bloco, ou seja, apaga o resto do fonte.

As teclas de função podem ser configuradas pelo usuário, através do comando PROFILE, para executar diversas funções.

3. Variáveis de Sistema

*APPLIC-ID... : XXXPRG	*APPLIC-NAME... : SYSTEM
*CURSOR..... : 1004	*CURS-COL..... : 4
*CURS-FIELD... : 0	*CURS-LINE..... : 1
*DATA..... : -1	*DEVICE..... : COLOR
*ERROR-LINE... : 0	*ERROR-NR..... : 0
*ERROR-TA.... :	*ETID..... : X?hÂ?ø?%
*GROUP..... :	*HARDCOPY..... :
*HARDWARE.... : 2064	*INIT-ID..... : B1D8
*INIT-PROGRAM: NTRN	*INIT-USER.... : X121212
*LANGUAGE.... : 38	*LEVEL..... : 1
*LIBRARY-ID.. : CURSO	*LINE-COUNT... : 10
*LINESIZE.... : 80	*LOG-LS..... : 80
*LOG-OS..... : 23	*MACHINE-CLASS: MAINFRAME
*OPSYS..... : MVS/ESA	*OS..... : MVS/ESA
*OSVERS..... : SP7.0.1	*PAGESIZE..... : 24
*PAGE-NUMBER. : 1	*PF-KEY..... : ENTR
*PF-NAME..... :	*PROGRAM..... : XXXP0001
*SCREEN-IO... : X	*SERVER-TYPE... :
*STARTUP..... :	*STEPLIB..... : SISTEM
*SUBROUTINE... :	*THIS-OBJECT... : 0000000000000000
*TPSYS..... : CICS	*UI..... : CHARACTER
*USER..... : X121212	*USER-NAME.... : SYSTEM
*WINDOW-LS... : 80	*WINDOW-POS... : 0
*WINDOW-OS... : 24	

4. Comandos

4.1. Set Control

Comando utilizado para executar comandos de controle de terminal de dentro de um programa. Exemplo:

```
SET CONTROL 'M-2' → Exibe mensagens de erro no rodapé da tela
SET CONTROL 'N' → Simula um ENTER
```

4.2. Set Key

Comando utilizado para ativar/desativar as teclas de função.

```
[ON ]
SET KEY [OFF]
[ALL]

[programa]
SET KEY PFn = [ON      ] [NAMED [OFF      ]
              [OFF     ]      [operando]
              [HELP    ]
```

SET KEY ALL → Ativa todas as PF's para uso.

SET KEY PF2=PGM → Ativa PF2 para executar o programa especificado.

SET KEY OFF → Desativa todas as PF's.

SET KEY ON → Reativa todas as PF's.

SET KEY PF4=HELP → Ativa a tecla F4 para servir de HELP de campo.

SET KEY PF7='PRGP0010' NAMED 'MENU' → Ativa a tecla PF7 para executar o programa PRGP0010 e coloca a string MENU na régua de PF's.

```
DEFINE DATA LOCAL
1 #PF4 (A56)
1 #FCT (A8)
END-DEFINE
...
SET KEY PF1
      PF2
      PF3 = `MENU`
      PF4 = #PF4
      PF5 = `LISTA EMPREGADO`
      PF6 = `XXXP0070` NAMED `IMPRIME`
...

```

4.3. Mask

Permite que o conteúdo das posições de um campo seja checado. MASK é justificado à esquerda.

Os sinais de "*" (asterisco) ou "%" (percentual), inseridos na definição do MASK indicam que qualquer número de posições podem ser ignoradas na verificação. O "." (ponto), "," (vírgula) ou o "_" (underscore), significam que esta posição não será verificada.

A checagem pode ser feita por constantes, variáveis, tipos, datas e intervalos.

Exemplo de constante:

IF #X = MASK(..'VA') → as posições 3 e 4 do campo #X serão verificadas para o valor 'VA'.

TIPO	TESTE DE VERIFICAÇÃO
A	Alfabético minúsculo ou maiúsculo
N	Numérico
C	Alfanumérico
H	Hexadecimal
L	Alfabético minúsculo
U	Alfabético maiúsculo

Exemplo:

IF #X = MASK(AA..NN) → as posições 1 e 2 serão checadas para valores alfabéticos, e as posições 5 e 6 serão checados para valores numéricos.

DATA	TESTE DE VERIFICAÇÃO
YYYY	Validação de ano com 4 posições
YY	Validação de ano com 2 posições
MM	Valida mês
DD	Valida dia

Exemplo:

IF #X = MASK(MM'/'DD) → as posições 1 e 2 serão checadas para Mês, a posição 3 para '/' e as posições 4 e 5 para validar a data no mês.

Exemplo de checagem de variáveis e constantes:

Permite a validação de dados constantes ou variáveis. "X" representa a posição dentro do campo. Exemplos:

```
#X    (A5) init<'ABC E'>
#Y    (A4) init<'23C1'>
#Z    (A4) INIT<'0230'>

#X ← MASK(...'E')
#Y ← MASK(NN.N)
#Z ← MASK(..DD)
#Z ← MASK(MMDD)
#Z ← MASK(AA.N)
```

O MASK pode também ser definido numa variável auxiliar, tornando o processamento mais flexível. Exemplo:

```
MOVE `*"R"*' TO #M
IF #CAMPO NE MASK #M
```

A Verificação feita na variável #CAMPO será verdadeira se for encontrada a letra "R" em qualquer ponto desta, uma vez que o sinal "*" ignora qualquer número de caracteres.

4.4. Format

A instrução 'FORMAT' especifica valores para os parâmetros globais que se aplicam à formatação de relatórios e mapas.

```
FORMAT {parâmetro}
```

```
FORMAT AL=7    /* Exibição de 7 bytes do campo
              LS=15 /* Torna um página com 15 linhas
              PS=50 /* Permite exibição em 50 colunas
```

4.5. Instruções de Entrada/Saída

4.5.1. Input

Instrução para obter dados para o programa ou para mostrar os dados de programa.

```

INPUT  parâmetros
      WITH TEXT-option
      MARK *operando
      ALARM-option

      nX  'texto' (attributes) operand1 (parâmetros) ...
      nT  'c'(n)
      x/y
      /
    
```

onde X significa espaços em branco, T posição de tabulação, X/Y linha e coluna da tela.

Exemplos:

```

INPUT #VAR-ALFA #VAR-NUM(SG=OFF)
INPUT ` ` #AGENCIA
INPUT //// 10T `AGENCIA.....` #AGENCIA
    
```

Parâmetros:

AD=[r][a][i/o][c][f]

r → apresentação: Blink, Default, Intensified, Non-display

a → alinhamento: Left, Right

i/o → modo: Input (A), Modifiable, Output

c → tratamento de caracteres: T minúsculas para maiúsculas, W aceita minúsculas

f → caracter de preenchimento (filler): definido pelo usuário.

```

INPUT (AD=IMLT'.') //// 10T `AGENCIA.....` #AGENCIA
INPUT /// `AGENCIA` (I) #AGENCIA(AD=M)
INPUT (AD=M) ` ` #AGENCIA ` ` #CONTA
INPUT MARK *#CONTA /// #AGENCIA #CONTA
INPUT MARK 2 /// #AGENCIA #CONTA
    
```

INPUT com mensagem na linha de mensagem:

```

INPUT WITH TEXT `Digite a agência. ` `AGENCIA:` #AGENCIA (AD=MI'_)
    
```

INPUT com utilização de mapa externo:

```

INPUT USING MAP `XXXM0001`
    
```

Apresentando mensagem na linha de mensagem:

```
INPUT WITH TEXT #MSG USING MAP 'XXXM0014'
```

Marcando um determinado campo:

```
INPUT WITH TEXT #MSG MARK #CODIGO USING MAP 'XXXM0002'
```

4.5.2. Reinput

Instrução para voltar a executar a última instrução INPUT.

```
REINPUT [FULL] operando1 MARK *operando2 [alarm]
```

Operando1: variável ou constante com a mensagem a ser apresentada no REINPUT.

Operando2: campo para posicionamento do cursor.

Entre uma instrução INPUT e o REINPUT correspondente, não pode haver nenhuma instrução WRITE ou DISPLAY, pois para a execução do REINPUT, a última comunicação com a tela tem obrigatoriamente que Ter sido com um INPUT.

A cláusula FULL atualiza todos os campos da tela que foram alterados entre o INPUT e o REINPUT. Se não for especificada, não atualiza a tela com o novo conteúdo, mesmo que haja alteração dos mesmos.

```
DEFINE DATA LOCAL
1 #USUARIO      (A8)
1 #SENHA        (A8)
END-DEFINE
INPUT (AD=M) WITH TEXT 'Preencher os campos.' //
  10T 'Usuário.:' (I) #USUARIO /
  10T 'Password:' (I) #SENHA (AD=N) /
IF #USUARIO = ' '
  REINPUT FULL 'Informe código de usuário.' MARK *#USUARIO ALARM
END-IF
IF #SENHA = ' '
  REINPUT FULL 'Informe a senha.' MARK *#SENHA ALARM
END-IF
END
```

Utilizando um mapa para a entrada de dados, o programa anterior ficaria da seguinte maneira:

```
DEFINE DATA LOCAL
1 #USUARIO      (A8)
1 #SENHA        (A8)
END-DEFINE
```

```

INPUT WITH TEXT 'Preencher os campos.' USING MAP 'XXXM0020'
IF #USUARIO = ' '
  REINPUT FULL 'Informe código de usuário.' MARK *#USUARIO ALARM
END-IF
IF #SENHA = ' '
  REINPUT FULL 'Informe a senha.' MARK *#SENHA ALARM
END-IF
END

```

4.5.3. Display

Instrução para imprimir dados do programa, em formato de colunas. A instrução DISPLAY automaticamente inclui cabeçalhos, com o mesmo nome da variável, caso não seja informado um cabeçalho diferente.

```

DISPLAY (rep) NOTITLE parâmetros
      '='
      Nx  'texto'  operando  parâmetros
      NT  'c' n

```

Opções:

NOTITLE → suprime a linha de cabeçalho da página.

Rep → número do relatório. O Natural pode controlar até 32 relatórios simultâneos, numerados de 0 a 31.

Parâmetros:

SF (Spacing Fator) → Fator de espaçamento das colunas. Valores: de 1 a 30.

HC (Heading Centering) → Ajuste do cabeçalho das colunas – CENTER, LLEFT, RRIGHT.

FC (Filler Character) → Caracter de preenchimento do cabeçalho. Default: brancos.

UC (Underlining Character) → Caracter utilizado para sublinhar os cabeçalhos.

Default: '-'

Exemplos:

- DISPLAY (SF=25 HC=L UC=*) NUM-MATRIC NOME-EMPREG
- DISPLAY 'MATRICULA EMPREGADO' NUM-MATRIC
- DISPLAY 'MATRICULA/DO/EMPREGADO' NUM-MATRIC

Parâmetros de variáveis:

AL (Alphanumeric Length) → Tamanho máximo para exibição de campo alfanumérico.

NL (Numeric Length → Tamanho máximo para exibição de campo numérico.

Exemplos:

- DISPLAY NUM-MATRIC NOME-EMPREG(AL=20)
- DISPLAY NUM-MATRIC(NL=3) NOME-EMPREG SIG-ORGAO(AL=4)

Vejam os seguintes programas como exemplo de utilização da instrução DISPLAY:

```

DEFINE DATA LOCAL
1 EMP VIEW OF AD-EMPREGADO
  2 NUM-MATRIC
  2 NOME-EMPREG
  2 SIG-ORGAO
END-DEFINE
FIND EMP WITH SIG-ORGAO = 'DEPTO-001'
  DISPLAY NUM-MATRIC NOME-EMPREG SIG-ORGAO
END-FIND
END
    
```

A execução gera o seguinte resultado:

Page 1		15:43:16
NUM-MATRIC	NOME-EMPREG	SIG-ORGAO
-----	-----	-----
001536	FULANO DE TAL	DEPTO-001
147685	LUCIANO PERDIGÃO	DEPTO-001
875923	MARIA APARECIDA DA SILVA	DEPTO-001

4.5.4. Write

A instrução 'WRITE' causa a impressão de uma ou mais linhas com formatação livre, e com as seguintes características:

- Se uma linha lógica excede a linha física, ela será dividida em duas, mas nenhum campo será truncado;
- Não há geração de cabeçalhos, e o tamanho de cada elemento determina as posições de impressão;

- Faixas de valores de campos múltiplos e grupos periódicos são dispostos horizontalmente na mesma linha.

```
WRITE (rep) NOTITLE parâmetros
  NX      '='
  NT      'Texto'   operando   parâmetros
          'c' n
```

1. WRITE NOTITLE `=` #PRIMEIRO-NOME `=` #SOBRENOME `=` #ULTIMO-NOME //
`L O C A L I Z A Ç Ã O` /
`CIDADE:` #CIDADE /
`PAÍS:` #PAIS //
- ...
2. WRITE NOTITLE 5X #NOME 50T #PROFISSAO
- ...
3. WRITE / `MÉDIA CIDADE:` C*SALARIO(1) AVER(SALARIO(1)) //
- ...
4. WRITE / `MÉDIA SALÁRIO` C*SALARIO(1) AVER(SALARIO(1)) //
- ...
5. WRITE NOTITLE (AL=16 NL=8)
`=` #CODIGO `=` #NOME `=` #TELEFONE(AL=10 EM=XXX-XXXXXXX)

4.5.5. Write Title/Trailer

A instrução '*WRITE TITLE*' define um título para um relatório, em substituição ao título padrão do Natural.

```
WRITE (rep) TITLE [LEFT] [parâmetros]
  NX      '='
  NT      'Texto'   operando   parâmetros
          'c' n
  [SKIP n]
```

A instrução '*WRITE TRAILER*' define uma ou mais linhas a serem impressas no rodapé de um relatório.

```
WRITE (rep) TRAILER [LEFT] [parâmetros]
  NX      '='
  NT      'Texto'   operando   parâmetros
          'c' n
  [SKIP n]
```

Exemplo:

```
DEFINE DATA LOCAL
1 EMP VIEW OF EMPREGADO
  2 NUM-MATRIC
  2 NOME-EMPREG
  2 SIG-ORGAO
END-DEFINE
FORMAT(1) LS=100 PS=50
WRITE(1) TITLE LEFT 'RELAÇÃO DE EMPREGADOS' (I)
WRITE(1) TRAILER LEFT 'PÁGINA' *PAGE-NUMBER(1)(EM=ZZ9)
      SKIP 1
FIND EMP WITH SIG-ORGAO = 'DEPTO-001'
  DISPLAY NUM-MATRIC NOME-EMPREG SIG-ORGAO
END-FIND
END
```

4.5.6. Newpage

Este comando provoca um avanço de página do relatório.

```
NEWPAGE (rep) IF LESS THAN operand1 LINES LEFT
      WHEN
```

O '*NEWPAGE*' avança a página e provoca a impressão do cabeçalho na página seguinte.

IF LESS THAN: se especificada a cláusula, ocorrerá quebra de página somente quando houver menos linhas do que o especificado no parâmetro.

```
...
NEWPAGE (1) IF LESS THAN 5 LEFT
...
```

4.5.7. Eject

Este comando provoca um avanço de página do relatório, sem gerar cabeçalho na próxima página.

```
EJECT (rep) IF LESS THAN operand1 LINES LEFT
      WHEN
```

IF LESS THAN: se especificada a cláusula, ocorrerá quebra de página somente quando houver menos linhas do que o especificado no parâmetro.

```

...
EJECT (1) IF LESS THAN 5 LEFT
...

```

4.5.8. Skip

Gera linhas em branco em um relatório.

```
SKIP (rep) [Lines]
```

```

...
WRITE 'Total de pratos:' #TOTAL-PRATOS
SKIP 2
...

```

4.6. Operações Aritméticas

4.6.1. Add

Comando utilizado para efetuar a operação de soma.

```
ADD [ROUNDED] operando-1 ... operando-n TO operando-x
```

```
ADD [ROUNDED] operando-1 ... operando-n GIVING operando-x
```

O comando *ADD* permite adicionar constantes ou variáveis a uma variável. Se for utilizada a opção *TO*, os valores do *operando-1* até o *operando-n* serão adicionados ao *operando-x*, inclusive.

Se for utilizada a opção *GIVING*, o resultado da soma dos valores do *operando-1* até o *operando-n* será armazenado no *operando-x*.

A cláusula *ROUNDED* permite que o resultado seja arredondado, ajustado-se ao tamanho do *operando-x*.

Exemplos:

INSTRUÇÃO	RESULTADO
ADD +5 -2 -1 GIVING #A	#A: 2

```

ADD 0,231 3,6 GIVING #B           #B: 3,8
ADD ROUNDED 2,9 3,8 GIVING #C     #C: 7
MOVE *DATX TO #DATE
ADD 7 TO #DATE                     current date +7
ADD #ARRAY1 (2,*) TO #ARRAY2 (4,*) #ARRAY2 value = #ARRAY1 + #ARRAY2
    
```

4.6.2. Subtract

Comando utilizado para efetuar a operação de subtração.

```
SUBTRACT [ROUNDED] operando-1 ... FROM operando-2
```

```
SUBTRACT [ROUNDED] operando-1 ... FROM operando-2 GIVING operando-3
```

Quando for utilizada somente a cláusula FROM, o valor de *operando-2* será utilizado com a subtração do *operando-1* do *operando-2*.

Quando for utilizada a cláusula GIVING, o resultado da subtração do *operando-1* do *operando-2* será armazenado no *operando-3*.

Com a cláusula ROUNDED, o resultado é arredondado de acordo com o campo receptor.

```

DEFINE DATA LOCAL
  1 #A (P2) INIT <50>
  1 #B (P2)
  1 #C (P1,1) INIT <2,4>
END-DEFINE
SUBTRACT 6 FROM #A                /** Resultado:  #A: 44
SUBTRACT 6 FROM 11 GIVING #A      /** Resultado:  #A: 5
SUBTRACT 3 4 FROM #A GIVING #B   /** Resultado:  #A: 5  #B: -2
SUBTRACT -3 -4 FROM #A GIVING #B /** Resultado:  #A: 5  #B: 12
SUBTRACT ROUNDED 2,06 FROM #C    /** Resultado:  #C: 0,3
    
```

4.6.3. Multiply

Comando utilizado para efetuar a operação de multiplicação.

```
MULTIPLY [ROUNDED] operando-1 BY operando-2
```

```
MULTIPLY [ROUNDED] operando-1 BY operando-2 GIVING operando-3
```

Quando for utilizada somente a cláusula BY, o resultado da operação será armazenado em *operando-2*.

Quando for utilizada a cláusula GIVING, o resultado será armazenado em *operando-3*.

Com a cláusula ROUNDED, o resultado é arredondado de acordo com o campo receptor.

```
MULTIPLY #A BY 3
MULTIPLY #A BY 3 GIVING #C
MULTIPLY ROUNDED 3 BY 3.5 GIVING #C
MULTIPLY #ARRAY1 (2,*) BY #ARRAY2 (4,*)
```

4.6.4. Divide

Comando utilizado para efetuar a operação de divisão.

```
DIVIDE [ROUNDED] operando-1 INTO operando-2 GIVING operando-3
```

```
DIVIDE operando-1 INTO operando-2 GIVING operando-3 REMAINDER operando-4
```

Se for utilizada a cláusula GIVING, o resultado da divisão será armazenado em *operando-3*, senão o resultado será armazenado em *operando-2*.

Se for utilizada a cláusula REMAINDER, o resto da divisão será armazenado em *operando-4*.

```
DEFINE DATA LOCAL
  1 #A (N7) INIT <20>
  1 #B (N7)
  1 #C (N3.2)
  1 #D (N1)
  1 #E (N1) INIT <3>
  1 #F (N1)

DIVIDE 5 INTO #A          /** Resultado: #A: 4
DIVIDE 5 INTO #A GIVING #B /** Resultado: #B: 4
DIVIDE 3 INTO 3,1 GIVING #C /** Resultado: #C: 1,03
DIVIDE 3 INTO 3,1 GIVING #D /** Resultado: #D: 1
DIVIDE 2 INTO #E REMAINDER #F /** Resultado: #E: 1 #F: 1
```

4.6.5. Compute

A instrução 'COMPUTE' executa operações aritméticas e permite a utilização de funções do Natural.

```
COMPUTE [ROUNDED] operando-1 = ... expressão aritmética
```

A cláusula ROUNDED faz com que o resultado seja arredondado.

A palavra COMPUTE é opcional, desde que o sinal de igualdade seja precedido de dois pontos: #VAR := 4 * #MULT

```
COMPUTE #A = 3 * 2 + 4 / 2 - 1      Resultado: #A = 0007
#A := SQRT (#B)                    Resultado: Raiz quadrada de #B
```

4.7. Manipulação de Campos

4.7.1. Compress

A instrução 'COMPRESS' é utilizada para combinar o conteúdo de dois ou mais operandos para um campo.

```
COMPRESS operando-1 ... operando-n [IN]TO operando-x
      [LEAVING NO [SPACE]]
      [WITH DELIMITER operando-3]
```

Operando-1 ... Operando-n → qualquer tipo de variável ou constante.

Operando-x → variável alfanumérica que receberá o resultado da concatenação.

LEAVING NO → concatena sem deixar espaço em branco entre os dados.

WITH DELIMITER → especifica um caracter para servir de separador entre os campos.

```
COMPRESS NOME SOBRENOME INTO #NOME-COMPLETO
COMPRESS #DIA #MÊS #ANO INTO #DATA LEAVING NO
COMPRESS #DIA #MÊS #ANO INTO #DATA WITH DELIMITER '/'
```

4.7.2. Move

A instrução 'MOVE' copia o conteúdo de um campo para outro ou atribui um valor a uma variável.

```

[ROUNDED]
[BY NAME]
[BY POSITION]
MOVE          operando-1 TO operando-2
[ALL]
[EDITED]

```

ROUNDED → faz a movimentação e arredonda, de acordo com o tamanho do campo receptor.

BY NAME → movimenta todos os campos de uma estrutura/registro para outro, desde que haja campos nas duas estruturas com o mesmo nome.

BY POSITION → movimenta todos os campos de uma estrutura/registro para outro, sendo que os nomes deste campos podem estar diferentes. As regras são as seguintes:

- O número de campos nos dois grupos precisa ser o mesmo;
- Os níveis dos campos precisam ser iguais;
- Se houver a presença de estruturas de ARRAY, eles precisam ter as mesmas dimensões.

ALL → efetua o preenchimento de uma variável *string* com parâmetro especificado.

EDITED → efetua a movimentação de um campo com uma máscara para outro campo.

```

MOVE #VAR TO #VAR-2
MOVE ROUNDED 1,95 TO #N1 /* (N1)
MOVE BY NAME #VIEW TO #VIEW-REG
    DEFINE DATA LOCAL
    1 #VIEW VIEW OF PESSOAL
    2 APELIDO
    2 PRIMEIRO-NOME
    2 IDADE
    2 SALARIO
    2 SEXO
    2 HOBBY
    1 #VIEW-REG
    2 #APELIDO (A30)
    2 #IDADE (N3)
    2 #SALARIO (N7)
    END-DEFINE
    READ #VIEW
    MOVE BY NAME #VIEW TO #VIEW-REG
    END-READ
    END

MOVE BY POSITION #VIEW TO #VIEW-REG
    DEFINE DATA LOCAL
    1 #VIEW VIEW OF PESSOAL

```

```

2 APELIDO
2 PRIMEIRO-NOME
2 IDADE
2 SALARIO
1 #VIEW-REG
2 #APELIDO-EMP      (A30)
2 #PRIMEIRO-NOME-EMP (A30)
2 #IDADE-EMP       (N3)
2 #SALARIO-EMP     (N7)
END-DEFINE
READ #VIEW
MOVE BY POSITION #VIEW TO #VIEW-REG
END-READ
END

MOVE ALL '*' TO #LINHA /* (A78)
MOVE EDITED DATX(EM=DD/MM/YYYY) TO #A /* (A10)

```

4.7.3. Substring

Permite mover a porção de um campo para outro campo.

```
MOVE SUBSTRING(#operando-1,i,q) TO operando-2
```

Onde i = Posição inicial e q = Quantidade de bytes.

4.7.4. Reset

A instrução 'RESET' estabelece valores nulos, de acordo com o tipo de variável, ou os valores especificados inicialmente.

```
RESET [INITIAL] operando-1 ... operando-n
```

A cláusula INITIAL restabelece o valor original definido no DEFINE DATA pela cláusula INIT.

DEFAULT DE INICIALIZAÇÃO		
A	Alfanumérico	Branco
B	Binário	0 (zero)
N	Numérico	0 (zero)
P	Numérico Compactado	0 (zero)
I	Inteiro	0 (zero)
L	Lógico	False
F	Ponto Flutuante	0 (zero)
D	Data	0 (zero)
T	Hora	00:00:00

```

DEFINE DATA LOCAL
1 #ALFA          (A10)
1 #PERCENTUAL   (N3,2)
1 #MAT-MÊS      (N2/12)
1 #REGISTRO
  2 #NOME        (A40)
  2 #MATRICULA   (N9)
1 #MAX-REG      (N4) INIT<100>
END-DEFINE
...
RESET #ALFA #PERCENTUAL #REGISTRO #MAT-MÊS(*) #MAX-REG
...
RESET INITIAL #MAX-REG
...

```

4.7.5. Separate

A instrução '*SEPARATE*' é usada para separar o conteúdo de um campo para dois ou mais campos (alfanuméricos, array ou grupos).

```
SEPARATE operando-1 [LEFT] INTO operando-2 WITH operando-3
```

Operando-1 → variável ou constante alfanumérica.

Operando-2 → variável alfanumérica (pode ser um array).

Operando-3 → variável ou constante. Se omitido, assume que o delimitador é qualquer carácter "menor" que a letra "a" (minúscula).

```

SEPARATE 'A B,C.D' INTO #A #B #C #D
SEPARATE '123.5,26.3' INTO #N1 #N2 WITH DELIMITER ','
SEPARATE '1 2 3 4' INTO #ARRAY(*) /* (A1/4)

```

4.7.6. Examine

A instrução '*EXAMINE*' pesquisa um campo (alfanumérico ou array) por uma opção e altera, deleta ou conta o número de ocorrências de um campo.

```

EXAMINE operando-1 FOR operando-2
  [DELETE [FIRST]]
  [REPLACE [FIRST] operando-3]
  [GIVING NUMBER operando-3]
  [GIVING POSITION operando-3]
  [GIVING INDEX operando-3]

```

Operando-1 → representa o campo a ser examinado.

Operando-2 → representa o valor a ser utilizado na operação de pesquisa.

DELETE → apaga a ocorrência especificada, concatenando o resto da string. Se FIRST for especificado, apaga apenas a primeira ocorrência.

REPLACE → troca a ocorrência especificada em *operando-2* pela especificada em *operando-3*. Se FIRST for especificado, apenas a primeira ocorrência será alterada.

GIVING NUMBER → usado para obter o número de vezes que ocorreu o *operando-2* no *operando-1*. Se o *operando-2* não for encontrado, retorna 0 (zero).

GIVING POSITION → usado para obter a partir de qual *byte* o *operando-2* inicia no *operando-1*. Se o *operando-2* não for encontrado, retorna 0 (zero).

GIVING INDEX → usado para obter o número da ocorrência (índice) em que o *operando-2* ocorre no *operando-1*, que deverá ser uma matriz. Caso não seja encontrado, retorno 0 zero).

```

DEFINE DATA LOCAL
1 #DEP1      (A18)
1 #DEP      (A18) INIT<'AAABBBCCDDDEEEFFF'>
1 REDEFINE #DEP
  2 #TAB-DEP (A3/6)
1 #I        (I1)
1 #J        (I1)
1 #Z        (I1)
END-DEFINE
MOVE #DEP TO #DEP1
*
EXAMINE #TAB-DEP(*) FOR 'BBB' GIVING INDEX #I
EXAMINE #TAB-DEP(*) FOR 'AAA' GIVING NUMBER #Z
EXAMINE #DEP FOR 'BBB' GIVING POSITION #J
EXAMINE #DEP FOR 'DE' REPLACE 'XY'
EXAMINE #DEP FOR 'C' REPLACE FIRST 'W'
EXAMINE #DEP FOR 'F' REPLACE 'Z'
EXAMINE #DEP FOR 'B' DELETE
*
DISPLAY #I #J #Z #DEP1 #DEP
END
    
```

A execução gera o seguinte resultado:

```

#I #J #Z      #DEP1      #DEP
-- -- -- -
  2  4  1 AAABBBCCDDDEEEFFF AAAWCCDDXYEEZZZ
    
```

4.8. Instruções Condicionais

4.8.1. If

A instrução *IF* é usada para controlar a execução de grupos de instruções com base em uma condição lógica.

```

IF condição [THEN]
    instrução
[ELSE]
    instrução
END-IF

```

```

IF SALARY (1) LT 40000
    WRITE NOTITLE `*****` NAME 30X `SALARY LT 40000`
ELSE
    IF BIRTH GT 300101
        FIND VEHIC-VIEW WITH PERSONNEL-ID = PERSONNEL-ID
        DISPLAY (IS=ON) NAME BIRTH SALARY (1) MAKE (AL=8)
    END-FIND
END-IF
END-IF

```

4.8.2. Decide On/Decide For

A instrução '*DECIDE*' é uma estrutura de múltipla escolha (case).

```

DECIDE ON FIRST VALUES OF operando-1
    EVERY
        VALUE operando-2 [,...]
        VALUE operando-2 : operando-3
        Instruções
    ANY VALUE
        Instruções
    ALL VALUE
        Instruções
    NONE VALUE
        Instruções
END-DECIDE

```

```

DECIDE FOR FIRST CONDITION
    EVERY
        WHEN condição-lógica
        Instruções
    WHEN ANY
        Instruções
    WHEN ALL
        Instruções
    WHEN NONE
        Instruções
END-DECIDE

```

Esta instrução é usada de duas formas, sendo:

- DECIDE ON → usada para executar uma ou mais ações, dependendo de um valor (ou valores) contido em uma variável.

- DECIDE FOR → usada para executar uma ou mais ações, dependendo de múltiplas condições lógicas.

As seguintes opções de controle podem ser usadas:

FIRST ou EVERY → para após a primeira condição verdadeira (FIRST), ou executa todas as condições descritas (EVERY).

ANY → é executado se alguma das condições for verdadeira.

ALL → é executado se todas as condições forem verdadeiras. (A opção EVERY tem de ser especificada).

NONE → é executado se nenhuma das condições for verdadeira. É obrigatório o uso desta opção e pode-se usar a cláusula IGNORE quando não se deseja executar nada.

Decide On First Value

```
DECIDE ON FIRST VALUE OF #COD
  VALUE 1
    PERFORM VERIFICA-COD-1
  VALUE 2:4
    PERFORM VERIFICA-COD-2-3-4
  VALUE 5 , 9 , 11
    PERFORM VERIFICA-COD-5-9-11
  VALUE 8
    STOP
  NONE
    IGNORE
END-DECIDE
```

Decide On Every Value

```
DECIDE ON EVERY VALUE OF #COD
  VALUE 1:6
    PERFORM VERIFICA-COD-1-6
  VALUE 1:3
    PERFORM VERIFICA-COD-1-3
  VALUE 4:6
    PERFORM VERIFICA-COD-4-6
  NONE
    IGNORE
END-DECIDE
```

Decide For First Condition

```
DECIDE FOR FIRST CONDITION
  WHEN #TIPO = 'A'
    PERFORM TRATA-CONTA-A
  WHEN #TIPO = 'C'
    PERFORM TRATA-CONTA-C
  WHEN NONE
    REINPUT 'Tipo de conta inválida.' MARK *#TIPO
END-DECIDE
```

Decide For Every Condition

```

DECIDE FOR EVERY CONDITION
  WHEN #TIPO = 'A'
    PERFORM TRATA-CONTA-A
  WHEN #NASC NE MASK(DDMMYYYY)
    REINPUT 'Data Nascimento inválida.'
  WHEN ANY
    MOVE TRUE TO #ERRO
  WHEN NONE
    IGNORE
END-DECIDE

```

Decide dentro de Decide

```

DECIDE ON FIRST VALUE OF #COD
VALUE 1
  DECIDE FOR FIRST CONDITION
  WHEN #TIPO > #TIPO-AUX
    REINPUT 'Tipo Inválido...' MARK ...
  WHEN #TIPO < #TIPO-AUX
    PERFORM GRAVA-TIPO
  WHEN NONE
    IGNORE
  END-DECIDE
VALUE 2
  WRITE 'Tipo: #TIPO'
  NONE
  IGNORE
END-DECIDE

```

4.9. Controle de Processamento

4.9.1. For

Instrução utilizada para iniciar um processamento de loop e controlar o número de vezes que o loop será executado.

```

FOR operando1 [=] operando2 [TO] operando3 [STEP operando4]
  ...
END-FOR

```

Operando1: pode ser uma variável de usuário ou um campo de arquivo, que terá o valor alterado a cada ocorrência do loop.

Operando2 e *Operando3*: variáveis ou constantes, de valor inicial e final do loop.

Operando4: incremento, podendo ser positivo ou negativo.

```

DEFINE DATA LOCAL
1 #I      (I2)

```

```

1 #NUM-PAR (I2)
END-DEFINE
FOR #I 2 100 2
  WRITE 'Número par:' #I
END-FOR
END

```

4.9.2. Repeat

A instrução *'REPEAT'* inicia um loop de processamento cujo final é controlado por uma condição lógica.

```

REPEAT
  Instruções
  [UNTIL]
  [WHILE] condição-lógica
END-REPEAT

```

```

REPEAT
  [UNTIL]
  [WHILE] condição-lógica
  Instruções
END-REPEAT

```

Se a condição lógica for omitida, o loop deve ser encerrado com instruções do tipo *'ESCAPE'* ou *'STOP'*.

WHILE → executa o loop enquanto a condição for verdadeira.

UNTIL → executa o loop enquanto a condição for falsa.

```

...
REPEAT
  INPUT USING MAP 'XXXM0005'
  IF #MATRICULA = 0
    ESCAPE BOTTOM
  END-IF
  FIND EMPREGADO WITH MATRICULA = #MATRICULA
  END-FIND
  INPUT USING MAP 'XXXM0008'
END-REPEAT
...

```

4.9.3. Accept / Reject

Estes comandos são usados para aceitar (ACCEPT) ou rejeitar (REJECT) registros, de acordo com condições lógicas em processamento de loop.

```
ACCEPT IF condição-lógica
```

```
REJECT IF condição-lógica
```

O Natural é quem faz o teste e não o banco de dados. Por essa razão, o registro é recuperado (lido), para que o teste seja efetuado.

Os campos utilizados para teste podem ser campos da view ou variáveis de usuário.

Pode haver combinação de vários ACCEPT / REJECT numa leitura, mas o comando REJECT deve ser codificado antes do comando ACCEPT.

```
READ EMP
  ACCEPT IF #SEXO = 'M' AND #ESTADO-CIVIL = 'S'
  DISPLAY #NOME #IDADE
END-READ
```

4.9.4. Escape Top/Bottom/Routine

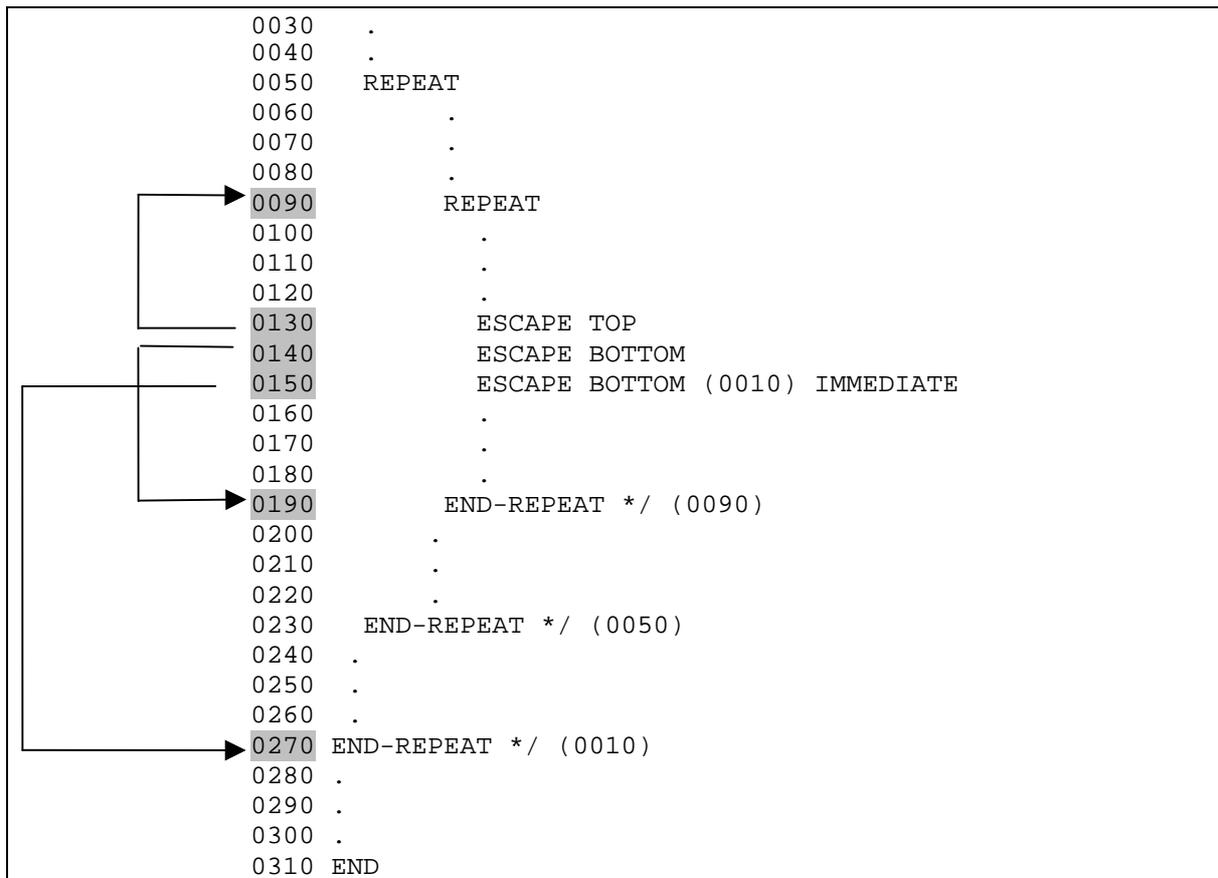
```
ESCAPE TOP
      BOTTOM (r) IMMEDIATE
      ROUTINE IMMEDIATE
```

ESCAPE TOP → causa o retorno para o início do processamento de loop. Dar-se-á nova repetição do processo de loop.

ESCAPE BOTTOM → encerra o processamento do loop, e continuará com o primeiro comando após o loop. Se a cláusula 'LABEL' for especificada, o processamento seguirá com o comando identificado pelo 'LABEL' ou pelo número de referência. Se a cláusula 'IMMEDIATE' for especificada, não se dará a evolução das funções 'STANDARD' (Aver, count, max) e as funções de quebras por fim de processamento.

ESCAPE ROUTINE → a opção indica que a rotina, que pode Ter sido invocada por um PERFORM, CALLNAT, FETCH RETURN ou pelo programa principal, abandonará o controle retornando ao nível superior. No caso do programa principal, o Natural receberá o controle. Pode-se usar também a cláusula 'IMMEDIATE'.

```
0010 REPEAT
0020 .
```



4.9.5. End

A instrução 'END' indica fim de programa. Todos os objetos têm de Ter o END como último statement, com exceção do COPYCODE.

```

FIND ...
  READ ...
  END-READ
END-FIND
END

```

4.9.6. Terminate

O comando termina a execução do programa e do Natural, retornando o controle ao monitor de teleprocessamento (CICS, TSO, etc.).

```

...
IF *PF-KEY = 'PF5' OR = 'PF17'
  TERMINATE
END-IF
...

```

4.9.7. Stop

Termina a execução de um programa e retorna o controle para o Natural.

```
...
IF #OPÇÃO = 9
  STOP
END-IF
...
```

4.9.8. Limit

A instrução 'LIMIT', limita o número de loop's para comandos de leitura (READ, FIND, HISTOGRAM) de acordo com o valor de 'n', retornando uma mensagem.

```
LIMIT n
```

```
...
LIMIT 4
*
READ EMPLOY-VIEW BY NAME STARTING FROM `BAKER`
...
END-READ
```

4.10. Processamento de Quebra Automática

4.10.1. At Start Of Data

Comando que executa uma série de comandos no início da leitura de um file.

```
[AT] START [OF] DATA
...
END-START
```

Deve ser codificado dentro do laço de leitura.

```
FIND EMPRESA WITH SIGLA-UF = `MG`
  AT START OF DATA
  WRITE `Início do Relatório.`
  END-START
END-FIND
```

4.10.2. At End Of Data

Comando que executa uma série de comandos ao término da leitura de um file.

```
[AT] END [OF] DATA
...
END-ENDDATA
```

Deve ser codificado dentro do laço de leitura.

```
FIND EMPRESA WITH SIGLA-UF = 'MG'
  AT START OF DATA
    WRITE 'Término do Relatório.'
  END-START
END-FIND
```

4.10.3. At Top Of Page

Comando que executa uma série de comandos toda vez que uma página for inicializada para impressão de um relatório.

```
[AT] TOP [OF] PAGE
...
END-TOPPAGE
```

Pode ser codificado em qualquer lugar do programa.

```
...
AT TOP OF PAGE
  WRITE 'Relatório de Funcionários em Férias.'
END-TOPPAGE
...
```

4.10.4. At End Of Page

Comando que executa uma série de comandos quando uma condição de fim de página for detectada.

```
[AT] END [OF] PAGE
...
END-ENDPAGE
```

Pode ser codificada em qualquer parte do programa.

```
...
AT END OF PAGE
  WRITE 'Página:' #PAG(EM=ZZ9)
END-ENDPAGE
...
```

4.10.5. At Break

A instrução 'AT BREAK' especifica os campo para cujos valores serão feitos controles de quebra automáticos ou manuais, e também os procedimentos a serem executados nestas quebras.

```
[AT] BREAK [OF] campo [/n/]
...
END-BREAK
```

Campo: atributo do arquivo, cujo conteúdo será utilizado para teste de quebra, ou seja, o *AT BREAK* terá os comandos internos executados toda vez que ocorrer uma alteração do valor do campo em relação ao conteúdo anterior.

/n/: pode-se especificar a quebra de um certo número de caracteres iniciais do campo, através da cláusula */n/*. Por exemplo, se quisermos uma quebra quando houver alteração da primeira letra de um nome, basta codificar *AT BREAK OF NAME /1/*.

Mais de um *AT BREAK* pode ser codificado numa operação de leitura, e quando acontece um quebra, as quebras de níveis anteriores também acontecem.

Funções disponíveis na quebra:

- COUNT → quantidade de registros lidos
- SUM → soma de valores
- OLD → conteúdo do campo de quebra antes de ocorrer a quebra.
- AVER → média de valores
- MAX → maior valor lido
- MIN → menor valor lido

Exemplo de quebra simples:

```
READ EMPREGADO BY CARGO
  WHERE DEPTO = 'VENDAS'
  AT BREAK OF CARGO
    WRITE '-' (70) /
      10T 'CARGO:' OLD(CARGO) (AD=I) /
      40T 'SALARIO MEDIO:' AVER(SALARIO) /
      40T 'QTDE. FUNCIONARIOS:' COUNT(SALARIO) /
      40T 'TOTAL SALARIO:' SUM(SALARIO)
  END-BREAK
  DISPLAY MATRICULA NOME CARGO DEPTO
END-READ
```

Exemplo de quebra composta:

```
FIND EMPREGADO WITH DEPTO = 'COMPRAS'
  SORTED BY SECAO SETOR SERVICIO
  DISPLAY MATRICULA NOME CARGO SALARIO
  AT BREAK OF SERVICIO
    WRITE 10T 'CARGO:' OLD(CARGO) /
    40T 'MENOR SALARIO:' MIN(SALARIO) /
    40T 'MAIOR SALARIO:' MAX(SALARIO) /
    40T 'SALARIO MEDIO:' AVER(SALARIO) /
    40T 'TOTAL SALARIO:' SUM(SALARIO) /
    40T 'QTDE. FUNCIONARIOS:' COUNT(SALARIO)
  END-BREAK
  AT BREAK OF SETOR
    WRITE 10T 'CARGO:' OLD(CARGO) /
    40T 'MENOR SALARIO:' MIN(SALARIO) /
    40T 'MAIOR SALARIO:' MAX(SALARIO) /
    40T 'SALARIO MEDIO:' AVER(SALARIO) /
    40T 'TOTAL SALARIO:' SUM(SALARIO) /
    40T 'QTDE. FUNCIONARIOS:' COUNT(SALARIO)
  END-BREAK
  AT BREAK OF SECAO
    WRITE 10T 'CARGO:' OLD(CARGO) /
    40T 'MENOR SALARIO:' MIN(SALARIO) /
    40T 'MAIOR SALARIO:' MAX(SALARIO) /
    40T 'SALARIO MEDIO:' AVER(SALARIO) /
    40T 'TOTAL SALARIO:' SUM(SALARIO) /
    40T 'QTDE. FUNCIONARIOS:' COUNT(SALARIO)
  END-BREAK
END-FIND
```

4.11. Modularização do Sistema

4.11.1. Define Subroutine

Este comando é utilizado para definir uma sub-rotina, interna ou externa.

```
DEFINE [SUBROUTINE] nome-da-rotina
  Instruções
END-SUBROUTINE
```

Nome-da-rotina → pode conter até 32 caracteres.

A sub-rotina é chamada através do comando PERFORM. Uma alteração na sub-rotina externa não condiciona à recompilação do objeto que a chama.

Subrotina Interna

```

...
END-READ
*
DEFINE SUBROUTINE VALIDA-CAMPO
IF #CAMPO <= 0
  WRITE 'Campo inválido.'
  ESCAPE ROUTINE
END-IF
IF #CAMPO-2 = ' '
  WRITE 'Campo obrigatório.'
END-IF
END-SUBROUTINE

```

Subrotina Externa

```

DEFINE DATA PARAMETER
1 #SENHA      (A5)
1 #RETORNO    (L)
END-DEFINE
DEFINE SUBROUTINE TESTA-SENHA-ENTRADA
IF #SENHA = '15488'
  #RETORNO := TRUE
ELSE
  #RETORNO := FALSE
END-IF
END-SUBROUTINE
END

```

4.11.1.1. Perform

Executa a chamada a uma rotina, interna ou externa. O controle é devolvido à instrução seguinte ao PERFORM, após a execução da sub-rotina.

```
PERFORM subrotina [parâmetro-1 ... parâmetro-n]
```

O comando ESCAPE ROUTINE permite interromper a execução de uma sub-rotina antes do seu término, devolvendo o controle para a instrução PERFORM.

```

DEFINE DATA LOCAL
1 FUNC VIEW OF FUNCIONARIO
  2 NOME-FUNC
1 #COD-FUNC      (N9)
END-DEFINE
INPUT USING MAP 'XXXM0013'
PERFORM VALIDA-MAPA
PERFORM ALTERA-REGISTRO
...
DEFINE SUBROUTINE VALIDA-MAPA
IF #COD-FUNC = 0
  REINPUT 'Código inválido.' MARK *#COD-FUNC
END-IF
...
END-SUBROUTINE
...
END

```

4.11.1.2. Callnat

O comando CALLNAT permite chamar uma sub-rotina passando alguns parâmetros. O sub-programa recebe estes parâmetros através de uma PARAMETER, onde devem ter a mesma seqüência, formato e tamanho. A DEFINE DATA PARAMETER deve ser a primeira instrução no sub-programa.

Os parâmetros podem ser variáveis ou constantes e podem ser modificados pelo sub-programa.

```
CALLNAT 'subrotina' [parâmetro-1 ... parâmetro-n]
```

```
0010 DEFINE DATA LOCAL
0020 1 #YEAR (A4/9)
0030 1 #MONTH (A10/12)
0040 END-DEFINE
0050 SET KEY PF5
0060 PERFORM ARRAY-DATA
0070 CALLNAT 'SPGM-A' #YEAR(*) #MONTH(*)
0080 INPUT USING MAP 'ARR-MAP1'
0090 IF *PF-KEY = 'PF5'
0100     PERFORM ARRAY-DATA
0110 END-IF
0120 WRITE 'FIM PROGRAMA'
0130 END
```

```
0010 ***** Subprograma SPGM-A *****
0020 DEFINE DATA PARAMETER
0030 1 #YEAR (A4/9)
0040 1 #MONTH (A10/12)
0050 END-DEFINE
0060 SEPARATE '1981 1982 1983 1984 1985 1986 1987 1988 1989'
0070     INTO #YEAR (1:9) WITH DELIMITERS ' '
0080 SEPARATE JAN. FEB. MAR. APR. MAY JUNE JULY AUG. SEPT. OCT. NOV. DEC.'
0090     INTO #MONTH (1:12) WITH DELEMITERS ' '
0100 END
```

4.11.1.3. Fetch

O comando FETCH é usado para executar um programa.

```
FETCH [RETURN] 'programa' [parâmetro-1 ... parâmetro-n]
```

Usando a opção *RETURN*, após o término do programa chamado pelo *FETCH*, o controle é retornado para a próxima linha de comando do programa principal.

Se a opção *RETURN* não for usada, o controle é retornado ao Natural 2, finalizando a execução do programa.

O programa chamado deve receber os parâmetros através da instrução INPUT, sendo que estes devem ter a mesma seqüência, formato e tamanho.

```
0010 DEFINE DATA
0030 LOCAL
0040 1 #YEAR (A4/9)
0050 1 #MONTH (A10/12)
0060 END-DEFINE
0070 SET KEY PF5
0080 PERFORM ARRAY-DATA
0090 FETCH RETURN 'PGM1' #YEAR(*) #MONTH(*)
0100 INPUT USING MAP 'ARR-MAP1'
0110 IF *PF-KEY = 'PF5'
0120     PERFORM ARRAY-DATA
0140 END-IF
0150 WRITE 'FIM PROGRAMA'
0160 END
```

```
0010 *** PGM1 ***
0020 DEFINE DATA LOCAL
0030 1 #ANO (A4/9)
0040 1 #MES (A10/12)
0050 END-DEFINE
0060 IF *DATA > 0
0070     INPUT #ANO(*) #MES(*)
0080 END-IF
0090 DISPLAY #ANO(*) #MES(*)
0100 END
```

4.11.1.4. Call

O comando CALL é usado para executar uma sub-rotina escrita em qualquer linguagem que suporte o comando ou instrução CALL padrão.

```
CALL 'sub-rotina' [parâmetro-1 ... parâmetro-n]
```

```
0010 DEFINE DATA LOCAL
0020 1 #DIA (N2)
0030 END-DEFINE
0040 CALL 'TESTE' #DIA
0050 *
0060 WRITE 'DIA:' #DIA
0070 WRITE 'FIM PROGRAMA'
0080 END
```

4.12. Utilizando Janelas

4.12.1. Define Window

Comando utilizado para especificar o tamanho, posição de uma janela.

```

DEFINE WINDOW nome-janela
    SIZE lin * col
    BASE lin / col
    [TITLE operando-1]
    FRAMED [ON]
            [OFF]
    [POSITION [ON]]
            [OFF]
    
```

SIZE → tamanho da janela, em linhas * colunas

BASE → posição de abertura da janela, em relação a uma janela normal (24 * 80), em linhas / colunas.

TITLE → título da janela. Pode ser uma variável alfanumérica ou uma constante.

FRAMED → se informado ON, a janela terá moldura. Se informado OFF, a janela não terá.

POSITION → deve ser informado apenas se a janela a ser exibida for maior do que a tela física. Se a janela for maior, ON mostrará na última linha uma informação de que existe mais janela a ser exibida. OFF desabilita a linha de posição.

O comando DEFINE WINDOW apenas define uma janela, sendo necessário o comando SET WINDOW para tornar a janela ativa.

```

DEFINE DATA LOCAL
1 #I      (N3)
END-DEFINE
DEFINE WINDOW JANELA1
    SIZE 13 * 10
    BASE 5 / 12
    FRAMED ON POSITION OFF
*
RESET #I
...
    
```

4.12.2.Set Window

Comando utilizado para ativar / desativar uma janela.

```

SET WINDOW nome-janela
    OFF
    
```

Com o comando *SET WINDOW nome-janela*, ativa-se a janela especificada, o que significa que as próximas instruções de entrada/saída utilizarão essa janela, até que a mesma seja desativada ou uma nova seja ativada.

OFF → desativa a janela lógica aberta atualmente.

```

DEFINE DATA LOCAL
1 #I      (N3)
END-DEFINE
DEFINE WINDOW JANELA1
  SIZE 13 * 10
  BASE 5 / 12
  FRAMED ON POSITION OFF
*
RESET #I
SET WINDOW 'JANELA1'
INPUT USING MAP 'XXXM0010'
SET WINDOW OFF
END

```

4.13. Utilizando Banco de Dados ADABAS

4.13.1.Store

A instrução 'STORE' cria um registro em um arquivo ADABAS, especificado pelo seu nome.

```
STORE [RECORD] [IN] [FILE] nome-view
```

Para que a gravação seja efetivada, é necessário que seja dado o comando END TRANSACTION.

```

DEFINE DATA LOCAL
1 EMP VIEW OF EMPREGADO
  2 MATRIC
  2 NOME
  2 DEPTO
1 #OPCAO      (A1)
END-DEFINE
INPUT USING MAP 'XXXM0011'
PERFORM CONSISTE-CAMPOS
INPUT USING MAP 'XXXM0012'
IF NOT(#OPCAO = 'S' OR = 'N')
  REINPUT 'Opção inválida.' MARK *#OPCAO
END-IF
IF #OPCAO = 'S'
  STORE EMP
  END TRANSACTION
END-IF
END

```

4.13.2.Update

A instrução 'UPDATE' atualiza valores de campos de um arquivo ADABAS. O registro atualizado é sempre o último lido anteriormente pelas instruções FIND, READ ou GET.

```
UPDATE [RECORD] [IN] [STATEMENT] [(R)]
```

O parâmetro (R) pode ser especificado para atualizar um registro de um FIND/READ de nível superior, informando o número da linha do primeiro FIND/READ entre parênteses. Se forem inseridas/apagadas linhas, o Natural automaticamente atualizará o número da linha especificada.

```
...
FIND EMP WITH MATRIC = #MATRIC
  #ISN := *ISN
END-FIND
...
GET EMP #ISN
MOVE 4012 TO EMP.COD-CARGO
UPDATE
END TRANSACTION
...
```

```
...
READ TAB-SALARIO BY NIVEL-SALARIAL
  PISO := PISO * 1,25
  UPDATE
  END TRANSACTION
END-READ
...
```

4.13.3.Delete

A instrução 'DELETE' exclui um registro do arquivo ADABAS. O registro excluído é sempre o último lido anteriormente pelas instruções FIND, READ ou GET.

```
DELETE [RECORD] [IN] [STATEMENT] [(R)]
```

O parâmetro (R) pode ser especificado para deletar um registro de um FIND/READ de nível superior, informando o número da linha do primeiro FIND/READ entre parênteses. Se forem inseridas/apagadas linhas, o Natural automaticamente atualizará o número da linha especificada.

```
...
DEP. READ DEPENDENTE BY MATRIC
  FIND EMPREGADO WITH MATRIC = DEPENDENTE.MATRIC
  IF NO RECORDS FOUND
    DELETE (DEP.)
  END TRANSACTION
  END-NOREC
  END-FIND
  END-READ
...
```

```
...  
READ ARQUIVO  
  DELETE  
END-READ  
END TRANSACTION  
...
```

4.13.4.End Transaction

Este comando atualiza os registros dos comandos *STORE*, *UPDATE* e *DELETE*.

```
END TRANSACTION
```

O '*END TRANSACTION*' encerra uma transação lógica, fechando todos os cursores abertos.

```
...  
IF #CONFIRMA = 'S'  
  UPDATE(0500)  
  END TRANSACTION  
END-IF  
...
```

4.13.5.Backout Transaction

Este comando é usado para cancelar atualizações em um arquivo.

```
BACKOUT TRANSACTION
```

O '*BACKOUT TRANSACTION*' desfaz as atualizações efetuadas no arquivo. As atualizações somente serão desfeitas desde que não seja dado um *END TRANSACTION*.

```
...  
IF #CONFIRMA = 'S'  
  UPDATE(0500)  
  END TRANSACTION  
ELSE  
  BACKOUT TRANSACTION  
END-IF  
...
```

4.13.6.Read

A instrução 'READ' inicia um loop de processamento para leitura seqüencial de um arquivo ADABAS.

```
READ [limite] nome-da-view
      IN PHYSICAL SEQUENCE
      BY ISN
          Descritor = operando-1 [thru operando-2]
              STARTING FROM operand1 [ENDING at operando-2]
      [WHERE condições]
```

Se não for especificada nenhuma cláusula, o READ executará um acesso físico à tabela (da mesma forma que o parâmetro IN PHYSICAL SEQUENCE).

A opção BY ISN executa uma leitura na tabela pelo número de ISN (Internal Sequence Number) do registro.

Quando o acesso é feito por um descritor, os registros serão recuperados na ordem do descritor, podendo ou não ser especificado um valor inicial para pesquisa e um valor final, para limitar a leitura.

A cláusula WHERE estabelece condições de recuperação de registros.

É importante frisar que o comando READ, quando utilizado com acesso pelo descritor, quando não encontra um registro que satisfaça à condição, recupera os registros a partir do próximo. Exemplo:

```
READ VIEW BY MATRICULA = 24912
```

Se não houver matrícula com valor 24912, o READ recuperará registros a partir da primeira matrícula acima da informada.

Outros exemplos:

```
READ CARGO
READ(5) CARGO
READ #LIMITE EMPREGADO
READ TAB-UF BY SIGLA-ESTADO
READ TAB-UF BY SIGLA-ESTADO = 'AL' THRU 'CE'
READ EMPREGADO BY MATRIC = 1000 THRU 2000
      WHERE LOCAL = 'BH'
```

```

DEFINE DATA LOCAL
1 FUNC VIEW OF EMPREGADO
  2 MATRICULA
  2 NOME
  2 LOTACAO
END-DEFINE
READ(10) FUNC BY MATRICULA
  WHERE LOTACAO = 'DEPTO-002'
  DISPLAY MATRICULA NOME LOTACAO
END-READ
END
    
```

```

DEFINE DATA LOCAL
1 CARGO VIEW OF TAB-CARGO
  2 COD-CARGO
  2 DESCR-CARGO
  2 NFR-CARGO
  2 DT-FIM-CARGO
END-DEFINE
READ CARGO BY COD-CARGO
  WHERE NFR-CARGO = 5 AND DT-FIM-CARGO = 999999
  PERFORM ALTERA-NFR
END-READ
END
    
```

4.13.7.Find

A instrução '*FIND*' seleciona um conjunto de registros de um arquivo ADABAS e inicia um loop de processamento, onde os registros selecionados são lidos um a um. É baseado em um critério de pesquisa com base em uma chave definida. Se não for satisfeita a condição, nenhum registro é retornado.

```

[ALL]
FIND [Limite] nome-da-view WITH critério-de-seleção
  [NUMBER]
  [WHERE condição-lógica]
  [SORTED BY campo-1 ... campo-3]
  [IF NO RECORDS FOUND]
  Instruções
[END-NOREC]
Instruções
END-FIND
    
```

ALL → processa todos os registros.

Limite → pode ser estabelecido um número de registros a serem lidos.

WITH → especifica um critério de seleção, aplicado a um campo descritor.

NUMBER → se for especificada a cláusula NUMBER, o FIND não abre um laço, não precisando do END-FIND, mas apenas conta o número de registros que atendam aos critérios de seleção. O número de registros será recuperado com a variável de sistema *NUMBER.

WHERE → pode-se especificar critérios condicionais para a recuperação de registros.

SORTED BY → especifica campos (até 3) para ordenação.

IF NO RECORDS FOUND → caso não haja nenhum registro a recuperar com as condições especificadas, a cláusula permite que se executem instruções.

Para cada 'FIND' estarão disponíveis as seguintes variáveis do sistema:

- *ISN → contém o ISN (Internal Sequence Number) do registro em processamento.
- *NUMBER → contém a quantidade de registros que foram selecionados pelo critério de pesquisa.
- *COUNTER → contém o número de vezes que o loop foi inicializado até o momento (antes da verificação da cláusula WHERE).

Exemplos:

```
FIND CARGO WITH COD-CARGO = 101
FIND CARGO WITH COD-CARGO > 100
  WHERE NOME-CARGO < 'C'
FIND NUMBER CARGO WITH SITUACAO = 001
FIND ESTADO WITH SIGLA-ESTADO = 'MG' AND DATA-NASC > #DATA-NASC
FIND(20) FUNCIONARIO WITH SITUACAO = 'ATIVO'
```

```
DEFINE DATA LOCAL
1 DEPTO VIEW OF DEPARTAMENTO
  2 COD-DEPTO
  2 NOME-DEPTO
1 FUNC VIEW OF FUNCIONARIO
  2 MATRICULA
  2 NOME
  2 COD-DEPTO
1 #MATR-FUNC          (N9)
1 #COD-DEPTO         (A6)
END-DEFINE
REPEAT
  INPUT USING MAP 'XXXM0050'
  IF *PF-KEY = 'PF3'
    ESCAPE BOTTOM
  END-IF
  FIND NUMBER FUNC WITH MATRICULA = #MATR-FUNC
  IF *NUMBER NE = 0
    REINPUT 'Funcionário já cadastrado.' MARK *#MATR-FUNC
  END-IF
```

```

FIND DEPTO WITH COD-DEPTO = #COD-DEPTO
  IF NO RECORDS FOUND
    REINPUT 'Departamento inexistente.' MARK *#COD-DEPTO
  END-NOREC
  MOVE NOME-DEPTO TO #NOME-DEPTO
END-FIND
DISPLAY NOME-DEPTO NOME MATRICULA
END-REPEAT
END

```

4.13.8. Get

Comando utilizado para recuperação de um registro através do seu ISN. O número do ISN de um registro pode ser conhecido através da variável de sistema *ISN, quando da sua recuperação através do FIND ou READ.

```

GET nome-da-view operando
      *ISN

```

O GET não inicia processo de loop. O operando pode ser uma constante ou uma variável numérica.

```

DEFINE DATA LOCAL
  1 CLIENTE VIEW OF CLIENTES
    2 COD-CLIENTE
    2 NOME-CLIENTE
  1 #NUMERO-ISN      (P10)
END-DEFINE
FIND CLIENTE WITH COD-CLIENTE = 2678
  #NUMERO-ISN := *ISN
END-FIND
INPUT USING MAP 'XXXM0099'
IF *PF-KEY = 'PF5'
  GET CLIENTE #NUMERO-ISN
  DELETE
  END TRANSACTION
END-IF
END

```

4.13.9. Histogram

A instrução *'HISTOGRAM'* inicia um loop de processamento para uma leitura seqüencial dos valores de um descritor de um arquivo ADABAS.

O HISTOGRAM faz a leitura diretamente das listas invertidas do descritor, portanto não torna disponível nenhum campo do arquivo usado.

```
HISTOGRAM [limite] nome-da-view
          [VALUE FOR FIELD] campo
          [STARTING FROM operando-1]
                [WITH]
          [THRU      operando-2]
          [ENDING AT]
Instruções
END-HISTOGRAM
```

Limite → pode ser especificado um número ou variável como limite.

Nome-view → nome da view, que deve Ter apenas os campos descritores declarados.

Campo → nome do campo a contar.

Operando-1 / operando-2 → especificação de critério de seleção.

Para a instrução '*HISTOGRAM*' estão disponíveis as seguintes variáveis do sistema:

- *ISN → para descritores contidos em grupos periódicos, contém a ocorrência em que o valor lido se encontra. Para os demais descritores contém zero.
- *NUMBER → para cada valor do descritor, contém a quantidade de registros em que este valor aparece.
- *COUNTER → contém a quantidade de valores lidos até o momento (antes da verificação da cláusula '*WHERE*').

```
DEFINE DATA LOCAL
1 FUNC VIEW OF EMPREGADO
  2 #DEPTO
END-DEFINE
HISTOGRAM FUNC DEPTO
  DISPLAY DEPTO *NUMBER
END-HISTOGRAM
END
```

A execução gera o seguinte resultado:

DEPTO	*NUMBER
-----	-----
DEP001	6
DEP002	10
DEP003	8
DEP004	25
DEP005	11

4.14. Utilizando Banco de Dados DB2/SQL

4.14.1. Insert

Adiciona um ou múltiplos registros em uma tabela.

```
INSERT INTO Tabela [(campo1[, campo2[, ...]])]
VALUES (valor1[, valor2[, ...])
```

Exemplos utilizando as tabelas definidas acima:

```
INSERT INTO dependente
  (Código_sócio, Código_dependente, nome, data_nascimento)
VALUES
  (1005 , 3 , 'Joana Filha de João' , '20.07.2001')
```

Notem que os campos definidos e os valores estão na mesma ordem. Havendo mais campos na tabela, não referidos na lista:

- Se aceitarem nulos, serão nulificados;
- Se não aceitam nulo e não tem cláusula *default*, não haverá a inclusão com retorno de erro;
- Se não aceitam nulo e tem cláusula *default*, o default será assumido.

Não somos obrigados a listar os campos. Se os valores estiverem na mesma ordem da criação da tabela e nenhum campo omitido (salvo no final), podemos informar somente os valores. Essa é uma prática imperdoável para um programador NATURAL ou COBOL. Qualquer alteração na tabela causará alteração no código.

Outro exemplo:

```
INSERT INTO Dependente
  (Código_sócio, Código_dependente, nome, data_nascimento)
VALUES
  (1213 , 3 , 'Joaquina' , '20.07.2001');
```

Neste caso, se estamos utilizando os corretos conceitos de restrição de integridade, não haverá a inclusão, pois na tabela sócio não existe registro com chave 1213.

4.14.2. Select

É o comando SQL que permite a consulta. Sua sintaxe é:

```
SELECT [DISTINCT | ALL] { * | tabela.* | [tabela.]campo1 [AS alias1]
[[tabela.]campo2 [AS alias2] [, ...]] }
FROM nome-da-tabela [, ...]
[WHERE critérios]
[GROUP BY ...]
[UNION ...]
[ORDER BY ...]
```

O comando seleciona os campos informados, no produto cartesiano das tabelas informadas, as tuplas que atendem aos critérios informados na cláusula WHERE, ordenados pelas colunas informadas na cláusula ORDER BY.

Vamos a exemplos:

Tabela Sócio:

Código sócio	Nome	Data associação	...
1005	João	25/05/1998	...
1006	José	26/05/2000	...
1100	Francisco	27/08/1999	...
1212	Maria	12/12/1997	...

Tabela Dependente:

Código sócio	Código dependente	Nome	Data nascimento	...
1005	1	João Filho	14/07/1986	...
1005	2	Maria Filha	19/05/1998	...
1006	1	José Filho	13/06/1960	...
1006	2	Maria José	14/07/1961	...
1006	3	José Caçula	15/08/1962	...
1212	1	Mariana	17/03/1966	...

```
SELECT Sócio.nome, Dependente.nome
FROM Sócio, dependente
WHERE Sócio.Código_sócio = Dependente.Código_sócio
ORDER BY Sócio.nome, Dependente.nome
```

Sócio.nome	Dependente.nome
João	João Filho
João	Maria Filha
José	José Caçula
José	José Filho
José	Maria José
Maria	Mariana

Vamos observar que:

- O nome do sócio foi repetido para cada um de seus dependentes;
- A ordem não está a mesma: o resultado foi ordenado pela ordem alfabética;
- O sócio Francisco não apareceu pois não tem dependente.

Vamos a outro exemplo utilizando as mesmas tabelas:

```
SELECT Sócio.nome, count(*) as QTD
FROM Sócio, Dependente
WHERE Sócio.Código_sócio = Dependente.Sócio_código
GROUP BY Sócio.nome
```

O resultado será:

Sócio.nome	QTD
João	2
José	3
Maria	1

Vamos observar que:

- O SGDB contou quantas tuplas foram encontradas agrupadas pela coluna Sócio.nome;
- A coluna com as quantidades foi nomeada para QTD, obedecendo ao alias informado;
- O sócio Francisco não apareceu novamente.

Vamos ao mesmo exemplo com a cláusula HAVING:

```
SELECT Sócio.nome, count(*) as QTD
FROM Sócio, Dependente
WHERE Sócio.Código_sócio = Dependente.Código_sócio
GROUP BY Sócio.nome
HAVING QTD > 1
```

O resultado será:

Sócio.nome	QTD
João	2
José	3

Vamos observar que:

- O SGDB retirou a última tupla;
- O sócio Francisco não apareceu novamente.

4.14.3.Update

Altera valores de campos em uma tabela especificada, com base em critérios especificados.

```
UPDATE tabela
SET novovalor
WHERE critérios
```

Exemplo:

```
UPDATE Dependente
SET data_nascimento = '22.07.2001',
    Nome = 'Joaquina Filha de João'
WHERE Código_sócio = 1005 and
    Código_dependente = 3
```

4.14.4.Delete

Remove os registros de uma ou mais tabelas que satisfaçam aos critérios.

```
DELETE
FROM Tabela
WHERE critérios
```

Exemplo:

```
DELETE
FROM dependente
WHERE Código_sócio = 1005 and Código_dependente = 3
```

Outro Exemplo:

```
DELETE
FROM Sócio
WHERE Código_sócio = 1005
```

Novamente, se estamos utilizando os corretos conceitos de restrição de integridade, não haverá a exclusão, pois na tabela Dependente existem registro com chave 1005 que se tornarão órfãos.

4.14.5.Comando WHERE

Especifica quais registros das tabelas listadas na cláusula FROM são afetados por uma instrução SELECT, UPDATE ou DELETE. Se você não especificar uma cláusula WHERE, a consulta retornará todas as linhas da tabela, ou pior, todas as combinações das linhas das tabelas em produto cartesiano.

4.14.6.Cláusula GROUP BY

Combina registros com valores idênticos na lista de campos especificada em um único registro. Um valor de resumo é criado para cada registro se você incluir uma função agregada SQL, como SUM ou COUNT, na instrução SELECT.

4.14.7. Cláusula HAVING

Especifica mais registros agrupados são exibidos na instrução SELECT com uma cláusula GROUP BY. Depois de GROUP BY combinar os registros, HAVING exibirá qualquer registro agrupado pela cláusula GROUP BY que satisfaça às condições da cláusula HAVING.

4.14.8. Cláusula ORDER BY

Classifica os registros resultantes de uma consulta em um campo ou campos especificados, em ordem crescente ou decrescente. Os registros são classificados pelo primeiro campo listado após ORDER BY. Os registros que tem valores iguais naquele campo serão então classificados pelo valor no segundo campo listado e assim por diante.

4.14.9. Operador IN

Determina se o valor de uma expressão é igual a algum dos vários valores em uma lista especificada. A sintaxe é:

WHERE ... <campo> IN (<lista de valores>)

Se <campo> for encontrado em <lista de valores>, o operador IN retornará TRUE; caso contrário, retornará FALSE. A <lista de valores> pode ser uma lista simples, do tipo valores separados por vírgula ou um SELECT.

4.14.10. Operador BETWEEN .. AND

Determina se o valor de uma expressão se situa dentro de um intervalo especificado de valores. Se o valor de *expr* estiver entre *valor1* e *valor2* (inclusive ambas as pontas), o operador BETWEEN ... AND retornará TRUE; caso contrário, retornará FALSE.

4.14.11. Operador LIKE

Compara uma expressão de seqüência com um padrão em uma expressão SQL. Para padrão, você pode utilizar caracteres curinga (por exemplo, LIKE 'MAR%', para 'MAKROBOOKS') ou utilizar caracteres isolados (por exemplo, LIKE '_OSE', para 'JOSE' ou 'ROSE').

4.14.12. Operador IS NULL

Verifica se o valor de um expressão é nulo.

4.14.13. Função AVG

Calcula a média aritmética de um conjunto de valores contido em um campo especificado em uma consulta.

4.14.14. Função COUNT

Calcula o número de registros retornado por uma consulta. Utilizando-se formato COUNT(*) todas as tuplas resultantes serão contadas. Se o formato COUNT(distinct <campo>), a função contará quantas ocorrências diferentes estão presentes nas tuplas resultantes, excetuando-se os nulos.

4.14.15. Funções MIN, MAX

Retornam o mínimo ou o máximo de um conjunto de valores contido em um campo especificado em uma consulta. Não considera nulo. Havendo somente nulos, resultará em nulo.

4.14.16. Função SUM

Retorna a soma de um conjunto de valores contido em um campo especificado em uma consulta. A função SUM ignora os registros que contenham campos nulos. Havendo somente nulos, retornará nulo.

4.14.17. Comando COMMIT

Tornam permanentes todas a operações que alteraram o Banco de Dados e libera todos os *locks* desde o início da transação, do último COMMIT ou ROLLBACK.

4.14.18. Comando ROLLBACK

Desfaz todas a alterações (como os comandos INSERT, DELETE e UPDATE) feitas desde o início da transação, ou do último comando COMMIT ou ROLLBACK. Libera também todos os *locks*.

5. Lista de Exercícios

5.1. Exercícios de fixação em Natural

1. Definir uma área de variáveis locais com as seguintes características:
 - Variável “#CONTADOR” para assumir valores de 0 a 10.000
 - Variável “CÓDIGO” para guardar a identificação do departamento com 4 caracteres e inicializar com “PP01”
 - Construir uma tabela para lançar os valores de salários de todo o exercício (13 meses)
 - Variável “#CPF”, redefinindo-a em “#CHAVE” e “#DÍGITO_VERIFICADOR”

3. Fazer o programa abaixo conforme as regras definidas.

```

NOMEMAPA                SIGLA / APLICATIVO                DD/MM/AAAA
CHAVEUSU                Folha de Pagamento                HH:MM:SS
----- Cálculo de Salário -----

Nome Funcionário.....: Fulano de Tal
Salário.....: 9999999,99
-----
                        Quantidade Dias/Horas Trabalhadas
                        -----
Qtde. de Dias Normais.....: 99 Dias
Qtde. de Horas-Extras Normais.....: 99 Horas

Qtde. de Horas-Extras Feriados/Fim de Semana: 99 Horas
-----
Total de Horas Trabalhadas: 9999 Horas

Valor da Hora Normal.....(R$): 99999,99
Valor da Hora Extra.....(R$): 99999,99 (Hora normal + 50%)
Valor da Hora Extra Feriado/Domingo.(R$): 99999,99 (Hora normal + 100%)
Valor a Ser Remunerado.....(R$): 9999999,99
-----
F1 Ajuda  F3 Sai  F5 Encerra
    
```

Variáveis utilizadas:

#NOME-FUNC	(A40)
#VLR-SALARIO	(N9,2)
#QTD-DIA-NORMAL	(N2)
#QTD-HR-EXTRA-NORMAL	(N2)
#QTD-HR-EXTRA-FERDOM	(N2)
#QTD-TOT-HR-TRAB	(N4)
#VLR-HR-NORMAL	(N5,2)
#VLR-HR-EXTRA	(N5,2)
#VLR-HR-FERDOM	(N5,2)
#VLR-REMUNERADO	(N8,2)

Recomendações:

- Permitir entrar com o nome, o valor do salário, a quantidade de dias normais, horas-extras e horas-extras Feriados/Fim de Semana. Os demais campos serão calculados e exibidos após teclar ENTER.
- Nenhum dos campos de valor poderá ser menor que zero.
- O nome do funcionário, salário e a quantidade de dias normais devem ser obrigatórios.
- Quando a tecla de função "F1 Ajuda" for acionada, exibir a mensagem "Ajuda não disponível".
- Quando a tecla de função "F3 Sai" for acionada, sair do programa.
- Quando a tecla de função "F5 Encerra" for acionada, sair da Sessão Natural.

g) O cálculo deverá ser feito a partir de uma chamada a uma sub-rotina interna. Segue abaixo as fórmulas que serão utilizadas para o cálculo:

- *Total de Horas Trabalhadas*
(Quantidade Dias Normais * 8) + Quantidade de Hora-Extra Normal +
Quantidade de Hora-Extra Feriados/Domingo
- *Valor Hora Normal*
(Valor do Salário / 220)
- *Valor Hora-Extra*
(Valor Hora Normal + 50%)
- *Valor Hora-Extra Feriado/Domingo*
(Valor Horas Normais + 100%)
- *Valor Remunerado*
(Valor Hora Normal * (Quantidade Dias Normais * 8)) +
(Valor Hora-Extra * Quantidade Hora-Extra Normal) +
(Valor Horas-Extra Feriado/Domingo * Quantidade Hora-Extra Feriado/
Domingo)

4. (V) Verdadeiro ou (F) Falso:

- Os comandos “ . ” são comandos de linha no editor de mapa.
- SCRATCH apaga somente o programa fonte.
- Um Sub-programa pode ter Parameter e/ou Local.
- Uma linha contendo “ */ ” pode ser considerada com um comentário.
- Uma biblioteca só existe a partir da criação de qualquer objeto nela.
- Um sub-programa recebe os dados via Parameter.
- DDM's são arquivos ADABAS e DB2.
- O System File contém todos os programas Natural, fonte e/ou carga.
- LIST * → Lista todos os objetos.
- Para inserir um registro na tabela do DB2 utiliza-se o INSERT.
- Um programa recebe os dados via Parameter.
- O comando LIST CURM0001 lista o fonte do objeto CURM0001.
- O CHECK checa a sintaxe do objeto da área de trabalho.
- Os comandos “ .. ” não são comandos de linha em um editor de mapa.

5. Dado o programa abaixo, qual será o resultado dele? Os valores de #NUMERO são: 25, 28, 76, 14, 12, 5, 20, 10, 8, 90 e 1.

```
DEFINE DATA LOCAL
1 #NUMERO-AUX      (I2)
1 #SOMA            (I4)
1 #NUMERO          (I2)
END-DEFINE
REPEAT
  RESET #NUMERO
  INPUT (AD=MDLT'_' ) 'ENTRE COM O NUMERO:' #NUMERO
  IF #NUMERO >= #NUMERO-AUX
    #SOMA := #SOMA + #NUMERO
  END-IF
  #NUMERO-AUX := #NUMERO-AUX + 2
  UNTIL #NUMERO-AUX > 20
END-REPEAT
IF #SOMA > 260
  WRITE '=' #SOMA(EM=999.999.999)
END-IF
END
```

6. Os comandos abaixo são usados em um editor de mapas. Descreva o que cada uma faz.

- ..I - _____
- .C - _____
- .A - _____
- ..C - _____
- .D - _____

7. Construir um programa para dar entrada em uma data e consistir se esta data é válida. Utilizar o comando REINPUT FULL para consistência.

8. O programa abaixo utiliza o mapa CURM0001 para dar entrada nos dados de Código, Nome e Telefone. O mapa possui as seguintes variáveis: #CODIGO(N5), #NOME(A40) e #TELEFONE(N11/3).

```
DEFINE DATA LOCAL
1 #CODIGO          (I2)
1 #NOME            (A40)
1 #TELEFONE        (N11/1:3)
END-DEFINE
INPUT USING MAP `CURM0001`
*
DISPLAY #CODIGO #NOME #TELEFONE(*)
END
```

Quando este programa for compilado/executado ele deverá abrir o mapa CURM0001 que irá permitir a entrada do Código, Nome e Telefone. Quando teclar o *ENTER* ele irá mostrar o conteúdo dos campos #CODIGO, #NOME e #TELEFONE através do comando *DISPLAY*. Pergunta-se: este programa irá compilar corretamente? Porquê?

9. Qual o resultado das expressões abaixo:

```
DEFINE DATA LOCAL

1 #A (N2) INIT <50>

1 #B (N2)

1 #C (N1,1) INIT <2,4>

END-DEFINE

SUBTRACT 6 FROM #A                /** #A : _____
SUBTRACT 6 FROM 11 GIVING #A      /** #A : _____
SUBTRACT 3 4 FROM #A GIVING #B   /** #B : _____
SUBTRACT -3 -4 FROM #A GIVING #B /** #B : _____

RESET #A #B

ADD +9 -2 -1 GIVING #A           /** #A : _____
ADD ,231 3,6 GIVING #B          /** #B : _____

RESET #B #C

DIVIDE 3 INTO #A                 /** #A : _____
DIVIDE 5 INTO 5 GIVING #B       /** #B : _____

RESET #B #C

MULTIPLY #A BY 3 GIVING #C       /** #C : _____
MULTIPLY #A BY 0 GIVING #C      /** #C : _____

END
```

10. Identifique os tipos de objetos existentes:

- P → _____
- N → _____
- C → _____
- G → _____
- L → _____
- A → _____
- M → _____

11. A alteração da característica de um campo no mapa é feita utilizando o comando ".E". A partir daí precisamos colocar o nome da variável que representa este campo, formato, tamanho e as características do *Define AD-Attributes*. Identifique abaixo cada característica.

- MULT → _____
- MDR → _____
- OI → _____
- ALT'_' → _____
- TI → _____

12. Qual é o comando utilizado para as situações abaixo?

- Um sub-programa recebe os dados de outro programa.
-

- Um *pgm A* chama um *pgm B* que processa e retorna para o *pgm A*.
-

- Um programa recebe os dados de outro programa.
-

- Um programa chama um sub-programa.
-

- Chamada de uma sub-rotina interna.
-

13. Fazer um programa para dar entrada a 2 números, subtrair o menor do maior e jogar o resultado numa variável chamada diferença.

14. Quantas vezes serão executadas as etapas “1” e “2” do algoritmo abaixo, e que valor será impresso? Os valores de “A” a serem lidos são respectivamente, 50, 14, 38, 7, 6, 65, 31, 11, 44, 81 e 19.

```
DCL  N(n2)
      S(n2)
      A(n3)
enquanto (N <= 20) faça
  leia(A)
  se A >= N faça
    S = S + A → {1}
  fim-se
  N = N + 2 → {2}
fim-enquanto
se S < 170 faça
  imprime(S)
fim-se
```

15. Escreva um programa para ler uma temperatura dada na escala Fahrenheit e exibir o equivalente em Celsius. Fórmula:

$$C = \frac{5}{9}(F - 32)$$

16. Construir um programa para elevar um número a uma potência. A base e o expoente serão lidos do teclado, sendo seus valores números inteiros, positivos ou negativos, somente exibir o resultado se o mesmo for positivo.

17. Construir um programa que permita efetuar uma divisão utilizando apenas soma e subtração, exibindo no final o quociente e o resto. (obs.: o dividendo deve ser maior que o divisor, e ambos devem ser inteiros positivos).

18. Deseja-se calcular a conta de consumo de energia elétrica de um consumidor. Para isto, escreva um programa que leia o código do consumidor, o preço do Kw e a quantidade de Kw consumido, e exiba o código do consumidor e o total a pagar.

- total a pagar = preço x quantidade
- total a pagar mínimo = R\$ 11,20

19. Fazer um programa para ler uma variável que contém 8 dígitos alfanuméricos no formato DD/MM/AA (Data). Consistir e exibir no final se é uma data válida ou não.

20. Escreva um programa que determine o grau de obesidade de uma pessoa, sendo fornecido o peso e a altura da pessoa. O grau de obesidade é determinado pelo índice da massa corpórea ($\text{Massa} = \text{Peso} / \text{Altura}^2$) através da tabela abaixo:

MASSA CORPÓREA	GRAU DE OBESIDADE
< 26	Normal
≥ 26 e < 30	Obeso
≥ 30	Obeso Mórbido

21. Fazer um programa para calcular o quadrado dos inteiros pares positivos até 100 (incluindo o 0), atribuindo cada valor calculado aos elementos de um array denominado *quadrados*.

22. Fazer um programa para achar o maior elemento de cada posição correspondente entre 2 arrays numéricos com 20 elementos, e colocá-lo num 3º array. Exibir no final o novo array.
Obs.: Tamanho máximo de cada elemento é 3 bytes.

23. Faça um programa que, dado as três notas de um aluno, determine e exiba a sua média final e o seu conceito, sabendo-se que:

- a média final é calculada pela média aritmética das 3 notas;
- o conceito é determinado de com base na tabela abaixo:

MÉDIA FINAL	CONCEITO
$\geq 8,0$	A
$\geq 5,0$ e $< 8,0$	B
$< 5,0$	C

24. Fazer um programa para dar entrada em um vetor de 10 ocorrências. Exibir na saída o mesmo vetor, porém com a ordem de seus elementos invertida.

25. Faça um programa que, para um número indeterminado de pessoas:

- leia a idade de cada pessoa, sendo que a leitura da idade 0 (zero) indica o fim dos dados (flag) e não deve ser considerada;
- calcule e escreva o número de pessoas;
- calcule e escreva a idade média do grupo;
- calcule e escreva a menor idade e a maior idade.

26. Dado uma matriz MAT(10,10) quadrada numérica de 2 bytes cada elemento, fazer um programa para:

- Somar os elementos da diagonal principal e acima dela;
- Somar os elementos da diagonal principal e abaixo dela;
- Exibir o maior somatório encontrado indicando a que condição ele pertence.

5.2. Exercícios extras - Algoritmo

1. Fazer um algoritmo ou um programa para ler 100 dados de entrada positivos ou negativos. Imagine que estes valores representam as jogadas de um jogo entre 2 pessoas. Se o número é positivo o jogador "A" ganha um ponto, se o número for negativo o jogador "B" ganha um ponto. Imagine a contagem de pontos da seguinte forma: o jogador "A" começa servindo. Se quem estiver servindo ganha o ponto, então soma-se um ao seu score, caso contrário o serviço passa para o outro jogador e a contagem não se modifica. Completar o score do jogo e ao final exibir o resultado da partida (placar e vencedor).

2. Suponhamos que você esteja a beira de um rio e tenha 2 baldes, 1 com capacidade máxima de 5 litros e outro de 9 litros sem marcas de litros nos seus lados, estando inicialmente ambos vazios. O balde de 5 litros só pode receber água do rio. Construa um algoritmo que mostre todos os números de litros que podem ser medidos com esta manobra alternada de apanhar água do rio com o balde de 9 litros e jogá-la fora com o balde de 5 litros. Exibir também a quantidade de medidas encontradas.

Obs.: Em hipótese alguma você poderá usar um balde auxiliar.

5.3. Outros exercícios – Lógica (Internet)

- a) <http://www.portalchapeco.com.br/~jackson/sapo.htm>
- b) <http://www.portalchapeco.com.br/~jackson/canibais.htm>
- c) <http://www.portalchapeco.com.br/~jackson/rio.htm>

6. BIBLIOGRAFIA

- Manual da Consist
- Apostila de Natural Avançado – Politec
- <http://www.consist.com>

7. LICENÇA PÚBLICA GERAL GNU

7.1. Licença Pública Geral GNU (Tradução)

Esta é uma tradução não oficial da Licença de Documentação Livre GNU em Português Brasileiro. Ela não é publicada pela Free Software Foundation, e não se aplica legalmente a distribuição de textos que usem a GFDL - apenas o texto original em Inglês da GNU FDL faz isso. Entretanto, nós esperamos que esta tradução ajude falantes de português a entenderem melhor a GFDL.

Licença de Documentação Livre GNU

Versão 1.1, Março de 2000

Copyright (C) 2000 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

É permitido a qualquer um copiar e distribuir cópias exatas deste documento de licença, mas não é permitido alterá-lo.

0. INTRODUÇÃO

O propósito desta Licença é deixar um manual, livro-texto ou outro documento escrito "livre" no sentido de liberdade: assegurar a qualquer um a efetiva liberdade de copiar ou redistribuí-lo, com ou sem modificações, comercialmente ou não. Secundariamente, esta Licença mantém para o autor e editor uma forma de ter crédito por seu trabalho, sem ser considerado responsável pelas modificações feitas por terceiros.

Esta licença é um tipo de "copyleft" ("direitos revertidos"), o que significa que derivações do documento precisam ser livres no mesmo sentido. Ela complementa a GNU Licença Pública Geral (GNU GPL), que é um copyleft para software livre.

Nós fizemos esta Licença para que seja usada em manuais de software livre, porque software livre precisa de documentação livre: um programa livre deve ser acompanhado de manuais que forneçam as mesmas liberdades que o software possui.

Mas esta Licença não está restrita a manuais de software; ela pode ser usada para qualquer trabalho em texto, independentemente do assunto ou se ele é publicado como um livro impresso. Nós recomendamos esta Licença principalmente para trabalhos cujo propósito seja de instrução ou referência.

1. APLICABILIDADE E DEFINIÇÕES

Esta Licença se aplica a qualquer manual ou outro texto que contenha uma nota colocada pelo detentor dos direitos autorais dizendo que ele pode ser distribuído sob os termos desta Licença. O "Documento", abaixo, se refere a qualquer tal manual ou texto. Qualquer pessoa do público é um licenciado e é referida como "você".

Uma "Versão Modificada" do Documento se refere a qualquer trabalho contendo o documento ou uma parte dele, quer copiada exatamente, quer com modificações e/ou traduzida em outra língua.

Uma "Seção Secundária" é um apêndice ou uma seção inicial do Documento que trata exclusivamente da relação dos editores ou dos autores do Documento com o assunto geral do Documento (ou assuntos relacionados) e não contém nada que poderia ser incluído diretamente nesse assunto geral. (Por exemplo, se o Documento é em parte um livro texto de matemática, a Seção Secundária pode não explicar nada de matemática). Essa relação poderia ser uma questão de ligação histórica com o

assunto, ou matérias relacionadas, ou de posições legais, comerciais, filosóficas, éticas ou políticas relacionadas ao mesmo.

As "Seções Invariantes" são certas Seções Secundárias cujos títulos são designados, como sendo de Seções Invariantes, na nota que diz que o Documento é publicado sob esta Licença.

Os "Textos de Capa" são certos trechos curtos de texto que são listados, como Textos de Capa Frontal ou Textos da Quarta Capa, na nota que diz que o texto é publicado sob esta Licença.

Uma cópia "Transparente" do Documento significa uma cópia que pode ser lida automaticamente, representada num formato cuja especificação esteja disponível ao público geral, cujos conteúdos possam ser vistos e editados diretamente e sem mecanismos especiais com editores de texto genéricos ou (para imagens compostas de pixels) programas de pintura genéricos ou (para desenhos) por algum editor de desenhos grandemente difundido, e que seja passível de servir como entrada a formatadores de texto ou para tradução automática para uma variedade de formatos que sirvam de entrada para formatadores de texto. Uma cópia feita em um formato de arquivo outrossim Transparente cuja constituição tenha sido projetada para atrapalhar ou desencorajar modificações subseqüentes pelos leitores não é Transparente. Uma cópia que não é "Transparente" é chamada de "Opaca".

Exemplos de formatos que podem ser usados para cópias Transparentes incluem ASCII simples sem marcações, formato de entrada do Texinfo, formato de entrada do LaTeX, SGML ou XML usando uma DTD disponibilizada publicamente, e HTML simples, compatível com os padrões, e projetado para ser modificado por pessoas. Formatos opacos incluem PostScript, PDF, formatos proprietários que podem ser lidos e editados apenas com processadores de texto proprietários, SGML ou XML para os quais a DTD e/ou ferramentas de processamento e edição não estejam disponíveis para o público, e HTML gerado automaticamente por alguns editores de texto com finalidade apenas de saída.

A "Página do Título" significa, para um livro impresso, a página do título propriamente dita, mais quaisquer páginas subseqüentes quantas forem necessárias para conter, de forma legível, o material que esta Licença requer que apareça na página do título. Para trabalhos que não tenham uma tal página do título, "Página do Título" significa o texto próximo da aparição mais proeminente do título do trabalho, precedendo o início do corpo do texto.

2. FAZENDO CÓPIAS EXATAS

Você pode copiar e distribuir o Documento em qualquer meio, de forma comercial ou não comercial, desde que esta Licença, as notas de copyright, e a nota de licença dizendo que esta Licença se aplica ao documento estejam reproduzidas em todas as cópias, e que você não acrescente nenhuma outra condição quaisquer que sejam às desta Licença.

Você não pode usar medidas técnicas para obstruir ou controlar a leitura ou confecção de cópias subseqüentes das cópias que você fizer ou distribuir. Entretanto, você pode aceitar compensação em troca de cópias. Se você distribuir uma quantidade grande o suficiente de cópias, você também precisa respeitar as condições da seção 3.

Você também pode emprestar cópias, sob as mesmas condições colocadas acima, e você também pode exibir cópias publicamente.

3. FAZENDO CÓPIAS EM QUANTIDADE

Se você publicar cópias do Documento em número maior que 100, e a nota de licença do Documento obrigar Textos de Capa, você precisa incluir as cópias em capas que tragam, clara e legivelmente, todos esses Textos de Capa: Textos de Capa da Frente na capa da frente, e Textos da Quarta Capa na capa de trás. Ambas as capas também precisam identificar clara e legivelmente você como o editor dessas cópias. A capa da frente precisa apresentar o título completo com todas as

palavras do título igualmente proeminentes e visíveis. Você pode adicionar outros materiais às capas. Fazer cópias com modificações limitadas às capas, tanto quanto estas preservem o título do documento e satisfaçam essas condições, pode tratado como cópia exata em outros aspectos.

Se os textos requeridos em qualquer das capas for muito volumoso para caber de forma legível, você deve colocar os primeiros (tantos quantos couberem de forma razoável) na capa verdadeira, e continuar os outros nas páginas adjacentes.

Se você publicar ou distribuir cópias Opacas do Documento em número maior que 100, você precisa ou incluir uma cópia Transparente que possa ser lida automaticamente com cada cópia Opaca, ou informar em ou com cada cópia Opaca a localização de uma cópia Transparente completa do Documento acessível publicamente em uma rede de computadores, à qual o público usuário de redes tenha acesso a download gratuito e anônimo utilizando padrões públicos de protocolos de rede. Se você utilizar o segundo método, você precisa tomar cuidados razoavelmente prudentes, quando iniciar a distribuição de cópias Opacas em quantidade, para assegurar que esta cópia Transparente vai permanecer acessível desta forma na localização especificada por pelo menos um ano depois da última vez em que você distribuir uma cópia Opaca (diretamente ou através de seus agentes ou distribuidores) daquela edição para o público.

É pedido, mas não é obrigatório, que você contate os autores do Documento bem antes de redistribuir qualquer grande número de cópias, para lhes dar uma oportunidade de prover você com uma versão atualizada do Documento.

4. MODIFICAÇÕES

Você pode copiar e distribuir uma Versão Modificada do Documento sob as condições das seções 2 e 3 acima, desde que você publique a Versão Modificada estritamente sob esta Licença, com a Versão Modificada tomando o papel do Documento, de forma a licenciar a distribuição e modificação da Versão Modificada para quem quer que possua uma cópia da mesma. Além disso, você precisa fazer o seguinte na versão modificada:

- A. Usar na Página de Título (e nas capas, se alguma) um título distinto daquele do Documento, e daqueles de versões anteriores (que deveriam, se houvesse algum, estarem listados na seção Históricos do Documento). Você pode usar o mesmo título de uma versão anterior se o editor original daquela versão lhe der permissão.
- B. Listar na Página de Título, como autores, uma ou mais das pessoas ou entidades responsáveis pela autoria das modificações na Versão Modificada, conjuntamente com pelo menos cinco dos autores principais do Documento (todos os seus autores principais, se ele tiver menos que cinco).
- C. Colocar na Página de Título o nome do editor da Versão Modificada, como o editor.
- D. Preservar todas as notas de copyright do Documento.
- E. Adicionar uma nota de copyright apropriada para suas próprias modificações adjacente às outras notas de copyright.
- F. Incluir, imediatamente depois das notas de copyright, uma nota de licença dando ao público o direito de usar a Versão Modificada sob os termos desta Licença, na forma mostrada no Adendo abaixo.
- G. Preservar nessa nota de licença as listas completas das Seções Invariantes e os Textos de Capa requeridos dados na nota de licença do Documento.
- H. Incluir uma cópia inalterada desta Licença.

- I. Preservar a seção intitulada "Histórico", e seu título, e adicionar à mesma um item dizendo pelo menos o título, ano, novos autores e editor da Versão Modificada como dados na Página de Título. Se não houver uma sessão denominada "Histórico"; no Documento, criar uma dizendo o título, ano, autores, e editor do Documento como dados em sua Página de Título, então adicionar um item descrevendo a Versão Modificada, tal como descrito na sentença anterior.
- J. Preservar o endereço de rede, se algum, dado no Documento para acesso público a uma cópia Transparente do Documento, e da mesma forma, as localizações de rede dadas no Documento para as versões anteriores em que ele foi baseado. Elas podem ser colocadas na seção "Histórico". Você pode omitir uma localização na rede para um trabalho que tenha sido publicado pelo menos quatro anos antes do Documento, ou se o editor original da versão a que ela se refira der sua permissão.
- K. Em qualquer seção intitulada "Agradecimentos"; ou "Dedicatórias";, preservar o título da seção e preservar a seção em toda substância e tim de cada um dos agradecimentos de contribuidores e/ou dedicatórias dados.
- L. Preservar todas as Seções Invariantes do Documento, inalteradas em seus textos ou em seus títulos. Números de seção ou equivalentes não são considerados parte dos títulos da seção.
- M. Apagar qualquer seção intitulada "Endossos";. Tal sessão não pode ser incluída na Versão Modificada.
- N. Não re-intitular qualquer seção existente com o título "Endossos"; ou com qualquer outro título dado a uma Seção Invariante.

Se a Versão Modificada incluir novas seções iniciais ou apêndices que se qualifiquem como Seções Secundárias e não contenham nenhum material copiado do Documento, você pode optar por designar alguma ou todas aquelas seções como invariantes. Para fazer isso, adicione seus títulos à lista de Seções Invariantes na nota de licença da Versão Modificada. Esses títulos precisam ser diferentes de qualquer outro título de seção.

Você pode adicionar uma seção intitulada "Endossos";, desde que ela não contenha qualquer coisa além de endossos da sua Versão Modificada por várias pessoas ou entidades - por exemplo, declarações de revisores ou de que o texto foi aprovado por uma organização como a definição oficial de um padrão.

Você pode adicionar uma passagem de até cinco palavras como um Texto de Capa da Frente, e uma passagem de até 25 palavras como um Texto de Quarta Capa, ao final da lista de Textos de Capa na Versão Modificada. Somente uma passagem de Texto da Capa da Frente e uma de Texto da Quarta Capa podem ser adicionados por (ou por acordos feitos por) qualquer entidade. Se o Documento já incluir um texto de capa para a mesma capa, adicionado previamente por você ou por acordo feito com alguma entidade para a qual você esteja agindo, você não pode adicionar um outro; mas você pode trocar o antigo, com permissão explícita do editor anterior que adicionou a passagem antiga.

O(s) autor(es) e editor(es) do Documento não dão permissão por esta Licença para que seus nomes sejam usados para publicidade ou para assegurar ou implicar endossamento de qualquer Versão Modificada.

5. COMBINANDO DOCUMENTOS

Você pode combinar o Documento com outros documentos publicados sob esta Licença, sob os termos definidos na seção 4 acima para versões modificadas, desde que você inclua na combinação todas as Seções Invariantes de todos os documentos

originais, sem modificações, e liste todas elas como Seções Invariantes de seu trabalho combinado em sua nota de licença.

O trabalho combinado precisa conter apenas uma cópia desta Licença, e Seções Invariantes Idênticas com múltiplas ocorrências podem ser substituídas por apenas uma cópia. Se houver múltiplas Seções Invariantes com o mesmo nome mas com conteúdos distintos, faça o título de cada seção único adicionando ao final do mesmo, em parênteses, o nome do autor ou editor original daquela seção, se for conhecido, ou um número que seja único. Faça o mesmo ajuste nos títulos de seção na lista de Seções Invariantes nota de licença do trabalho combinado.

Na combinação, você precisa combinar quaisquer seções intituladas "Históricas"; dos diversos documentos originais, formando uma seção intitulada "Histórico"; da mesma forma combine quaisquer seções intituladas "Agradecimentos", ou "Dedicatórias". Você precisa apagar todas as seções intituladas como "Endosso".

6. COLETÂNEAS DE DOCUMENTOS

Você pode fazer uma coletânea consistindo do Documento e outros documentos publicados sob esta Licença, e substituir as cópias individuais desta Licença nos vários documentos com uma única cópia incluída na coletânea, desde que você siga as regras desta Licença para cópia exata de cada um dos Documentos em todos os outros aspectos.

Você pode extrair um único documento de tal coletânea, e distribuí-lo individualmente sob esta Licença, desde que você insira uma cópia desta Licença no documento extraído, e siga esta Licença em todos os outros aspectos relacionados à cópia exata daquele documento.

7. AGREGAÇÃO COM TRABALHOS INDEPENDENTES

Uma compilação do Documento ou derivados dele com outros trabalhos ou documentos separados e independentes, em um volume ou mídia de distribuição, não conta como uma Versão Modificada do Documento, desde que não seja reclamado nenhum copyright de compilação seja reclamado pela compilação. Tal compilação é chamada um "agregado", e esta Licença não se aplica aos outros trabalhos auto-contidos compilados junto com o Documento, só por conta de terem sido assim compilados, e eles não são trabalhos derivados do Documento.

Se o requerido para o Texto de Capa na seção 3 for aplicável a essas cópias do Documento, então, se o Documento constituir menos de um quarto de todo o agregado, os Textos de Capa do Documento podem ser colocados em capas adjacentes ao Documento dentro do agregado. Senão eles precisam aparecer nas capas de todo o agregado.

8. TRADUÇÃO

A tradução é considerada como um tipo de modificação, então você pode distribuir traduções do Documento sob os termos da seção 4. A substituição de Seções Invariantes por traduções requer uma permissão especial dos detentores do copyright das mesmas, mas você pode incluir traduções de algumas ou de todas as Seções Invariantes em adição as versões originais dessas Seções Invariantes. Você pode incluir uma tradução desta Licença desde que você também inclua a versão original em Inglês desta Licença. No caso de discordância entre a tradução e a versão original em Inglês desta Licença, a versão original em Inglês prevalecerá.

9. TÉRMINO

Você não pode copiar, modificar, sublicenciar, ou distribuir o Documento exceto como expressamente especificado sob esta Licença. Qualquer outra tentativa de copiar, modificar, sublicenciar, ou distribuir o Documento é nulo, e resultará automaticamente no término de seus direitos sob esta Licença. Entretanto, terceiros que tenham recebido cópias, ou direitos, de você sob esta Licença não

terão suas licenças terminadas tanto quanto esses terceiros permaneçam em total acordo com esta Licença.

10. REVISÕES FUTURAS DESTA LICENÇA

A Free Software Foundation pode publicar novas versões revisadas da Licença de Documentação Livre GNU de tempos em tempos. Tais novas versões serão similares em espírito à versão presente, mas podem diferir em detalhes ao abordarem novos problemas e preocupações. Veja <http://www.gnu.org/copyleft/>.

A cada versão da Licença é dado um número de versão distinto. Se o Documento especificar que uma versão particular desta Licença "ou qualquer versão posterior" se aplica ao mesmo, você tem a opção de seguir os termos e condições daquela versão específica, ou de qualquer versão posterior que tenha sido publicada (não como rascunho) pela Free Software Foundation. Se o Documento não especificar um número de Versão desta Licença, você pode escolher qualquer versão já publicada (não como rascunho) pela Free Software Foundation.

ADENDO: Como usar esta Licença para seus documentos

Para usar esta Licença num documento que você escreveu, inclua uma cópia desta Licença no documento e ponha as seguintes notas de copyright e licenças logo após a página de título:

```
Copyright (c) ANO SEU NOME.  
É dada permissão para copiar, distribuir e/ou modificar este documento  
sob os termos da Licença de Documentação Livre GNU, Versão 1.1 ou qualquer  
versão posterior publicada pela Free Software Foundation;  
com as Seções Invariantes sendo LISTE SEUS TÍTULOS, com os Textos da  
Capa da Frente sendo LISTE, e com os Textos da Quarta-Capa sendo LISTE.  
Uma cópia da licença em está inclusa na seção intitulada  
"Licença de Documentação Livre GNU".
```

Se você não tiver nenhuma Seção Invariante, escreva "sem Seções Invariantes" ao invés de dizer quais são invariantes. Se você não tiver Textos de Capa da Frente, escreva "sem Textos de Capa da Frente" ao invés de "com os Textos da Capa da Frente sendo LISTE"; o mesmo para os Textos da Quarta Capa.

Se o seu documento contiver exemplos não triviais de código de programas, nós recomendamos a publicação desses exemplos em paralelo sob a sua escolha de licença de software livre, tal como a GNU General Public License, para permitir o seu uso em software livre.

7.2. GNU Free Documentation License (original em ingles)

GNU Free Documentation License Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant.

The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section

- Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a

unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
    Copyright (c) YEAR YOUR NAME.
    Permission is granted to copy, distribute and/or modify this document
    under the terms of the GNU Free Documentation License, Version 1.2 or any
    later version published by the Free Software Foundation; with no Invariant
    Sections, no Front-Cover Texts, and no Back-Cover Texts.
    A copy of the license is included in the section entitled "GNU Free
    Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
    with the Invariant Sections being LIST THEIR TITLES, with the
    Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

A tradução para o português encontra-se disponível em <http://www.cipsga.org.br>. A tradução não é oficial. Ela não é publicada pela Free Software Foundation, e não se aplica legalmente a distribuição de textos que usem a GFDL - apenas o texto original em Inglês da GNU FDL faz isso. Entretanto, nós esperamos que esta tradução ajude a entender melhor a GFDL.