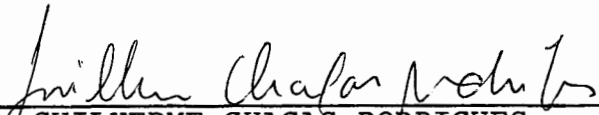


"EDIPRINT" SISTEMA DE EDIÇÃO E FORMATAÇÃO DE TEXTOS :  
DESENVOLVIMENTO E IMPLEMENTAÇÃO NO COMPUTADOR IBM - 1130

ANA MARIA MOURA MENDES

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE  
PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JA  
NEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU  
EM CIÊNCIAS (M.Sc.)

Aprovada por:

  
\_\_\_\_\_  
GUILHERME CHAGAS RODRIGUES  
(Presidente)

  
\_\_\_\_\_  
IVAN DA COSTA MARQUES

  
\_\_\_\_\_  
NELSON MACULAN FILHO

ÍNDICE

Índice .....	ii
Ficha catalográfica .....	iv
Dedicatória .....	v
Agradecimentos .....	vi
Resumo .....	vii
Abstract .....	viii
I - INTRODUÇÃO	
1.1 Objetivos do Trabalho .....	1
1.2 Etapas do Sistema .....	1
1.3 Fluxograma Geral .....	4
II - UTILIZAÇÃO DO SISTEMA	
2.1 Normas de utilização do sistema .....	5
2.2 Descrição dos procedimentos .....	7
2.2.1 Criação do Arquivo Fonte .....	7
2.2.2 Comandos para utilização do Editor de Texto .....	8
2.2.3 Comandos para utilização do Editor de Relatório..	16
2.2.4 Normas gerais para codificação dos comandos .....	20
2.2.5 Procedimento de Impressão .....	23
2.3 Observações Gerais .....	23
III - DETALHAMENTO DO SISTEMA	
3.1 Programa supervisor EPRIN .....	25
3.2 Criação do Arquivo Fonte .....	25
3.2.1 Arquivos Utilizados .....	25
3.2.2 Descrição dos Procedimentos de Criação do Arq.Fonte..	28
3.3 Editor de Texto .....	28
3.3.1 Descrição dos Procedimentos Utilizados .....	29
3.4 Editor de Relatório .....	35
3.4.1 Arquivos Utilizados .....	35
3.4.2 Descrição dos Procedimentos Utilizados .....	36
3.5 Impressão do Arquivo Fonte .....	38
IV - IMPLEMENTAÇÃO DO SISTEMA .....	40
V - CONCLUSÕES .....	43

VI - REFERÊNCIAS BIBLIOGRÁFICAS .....	46
VII-APÊNDICE	
A-I Rotinas de Criação do Arquivo .....	48
EPRIN, CRIA, INCLU, COPIA, LOCK, UNLOCK.	
A-II Rotinas do Editor de Texto .....	51
TEXTO, APAGA, ELIMI, DELET, INSER, LOCAL, SUBST, RENUM, BLOCO, STRIN, BUSCA, ACHAR, COMD1, COMD2, COMD3, SOMD4, CLEAN, APREG, TESTA, AGRUP, CRITI.	
A-III Rotinas do Editor de Relatório .....	62
RELAT, PESQ, ROUT, PULA, BRANC, COLUM, TITLE, NEWP, PAGE, TERMO, GRAVA, CABE, LEREG, CONCA, ORDEN, ESCR, PROC, PROCL, SEPAR, TRANS, COMPA, PONT, IMPL, INICL, FINAL, LISTA, TESTE.	
A-IV Rotinas de Impressão do Arquivo .....	79
IMPR	

FICHA CATALOGRÁFICA

MENDES, ANA MARIA MOURA

"EDIPRINT" Sistema de edição e formatação de Textos: Desenvolvi-  
mento e implementação no computador IBM - 1130 | Rio de Janeiro |  
1979.

viii, 79p 29.7cm (COPPE - UFRJ , M. Sc. Engenharia  
de Sistemas, 1979)

Tese - Univ. Fed. Rio de Janeiro - COPPE - UFRJ

1. EDIÇÃO DE TEXTOS VIA COMPUTADOR I COPPE/UFRJ  
II TÍTULO (série)

À minha filhinha Danielle,  
ao meu marido,  
aos meus pais,  
dedico este trabalho.

## AGRADECIMENTOS

Ao Prof. Guilherme Chagas Rodrigues, cujo interesse e dedicação aliados a uma orientação segura, contribuíram de maneira decisiva para os resultados aqui alcançados.

A Walter Lucio Barbosa do NPD - UFPb pelas valiosas sugestões e pelo apoio técnico recebido ao longo do trabalho.

A Hermes Pessoa Filho, chefe do NPD - UFPb por toda colaboração e ajuda material recebidas junto a UFPb.

Ao meu marido pela dedicação e apoio durante as horas de trabalho passadas ao meu lado.

A todos os colegas, operadores e perfuradores do Núcleo de Processamento de Dados da UFPb, por toda a colaboração direta ou indiretamente prestada.

Aos Profs. Ivan da Costa Marques e Nelson Maculan Filho pelas valiosas sugestões e por sua participação na banca examinadora.

A CAPES pelo suporte financeiro recebido.

## RESUMO

O presente trabalho visa o desenvolvimento de um sistema para edição de textos e relatórios dotado de versatilidade, rapidez e eficiência, para um computador de pequeno porte - IBM-1130. O sistema EDIPRINT opera OFFLINE e está constituído por 4 procedimentos: Criação e Impressão de arquivos, Editor de Texto e Editor de Relatório.

Através de comandos específicos do Editor de Texto é possível realizar atualizações de registros (alterar, inserir ou apagar) diretamente nos arquivos sem que para isto seja necessário regravar toda a massa de dados no sistema.

O Editor de Relatório permite que o texto gravado no arquivo seja emitido em um padrão pré-determinado pelo usuário, em uma ou mais cópias sucessivas, a partir de comandos contidos no texto que possibilitam a formatação desejada.

Os comandos utilizados são simples (mnemônicos de 1 ou 2 caracteres) e de fácil utilização, de modo a permitir que o sistema seja utilizado na elaboração de qualquer tipo de texto mesmo por aqueles que tenham poucos conhecimentos de computação.

ABSTRACT

This thesis has as its object the development of a system that edits text and reports and which brings versatility, quickness and efficiency, for a small configuration computer - IBM-1130. The EDPRINT system works offline and has 4 procedures: Generation and Print of data sets, Text and Report Editor.

It is possible through specific instructions of the Text Editor to change the registers (alter, insert or erase) directly in the data sets without being necessary to record all data again.

The Report Editor permits the text recorded in the data set to be printed in a standard way previously determined by the user, in one or more successive copies, from the commands in the text which give the desired format.

The commands used are simple (mnemonics of 1 or 2 characters) and easy to manipulate, making it possible for the system to be used in organization of any kind of report, even by those who have little computer experience.



## I - INTRODUÇÃO

### 1.1 - OBJETIVOS DO TRABALHO

O sistema EDIPRINT desenvolvido neste trabalho tem por objetivo a impressão de textos por computador em um padrão pré determinado pelo usuário, a partir de um texto qualquer fornecido em formato livre, através de cartões perfurados.

O EDIPRINT está constituído basicamente por dois subsistemas distintos, denominados respectivamente Editor de Texto e Editor de Relatório. O Editor de Texto permite alterar um texto gravado através de comandos específicos, que possibilitam inserir, modificar ou apagar trechos diretamente no arquivo, sem que para isto seja necessário regravar todo o texto novamente.

O Editor de Relatório permite imprimir textos em um padrão previamente escolhido, a partir de um texto fornecido inicialmente ao sistema em um formato qualquer. O EDIPRINT permite além disto criar e imprimir arquivos contendo textos a serem editados.

O EDIPRINT foi desenvolvido visando auxiliar o usuário na realização de trabalhos de documentação. Ele permite elaborar e emitir textos didáticos e científicos, relatórios e documentos com grande rapidez.

Durante o desenvolvimento do trabalho procurou-se implementar os comandos de maneira a tornar a utilização do sistema a mais simples e cômoda possível para o usuário, objetivando-se com isto torná-lo acessível mesmo àqueles que tenham reduzidos conhecimentos de computação.

O EDIPRINT embora tenha sido projetado para o IBM-1130 - sistema existente no Núcleo de Processamento de Dados (NPD) de J. Pessoa - UFPb, onde o trabalho foi desenvolvido - poderá com pequenas modificações ser estendido a qualquer outro computador, desde que este tenha o compilador Fortran. Considerações a respeito deste assunto são feitas na parte final do trabalho.

### 1.2 - ETAPAS DO SISTEMA

O sistema EDIPRINT está constituído por quatro blocos interligados entre si (Fig. I.1), cujas funções encontram-se es-

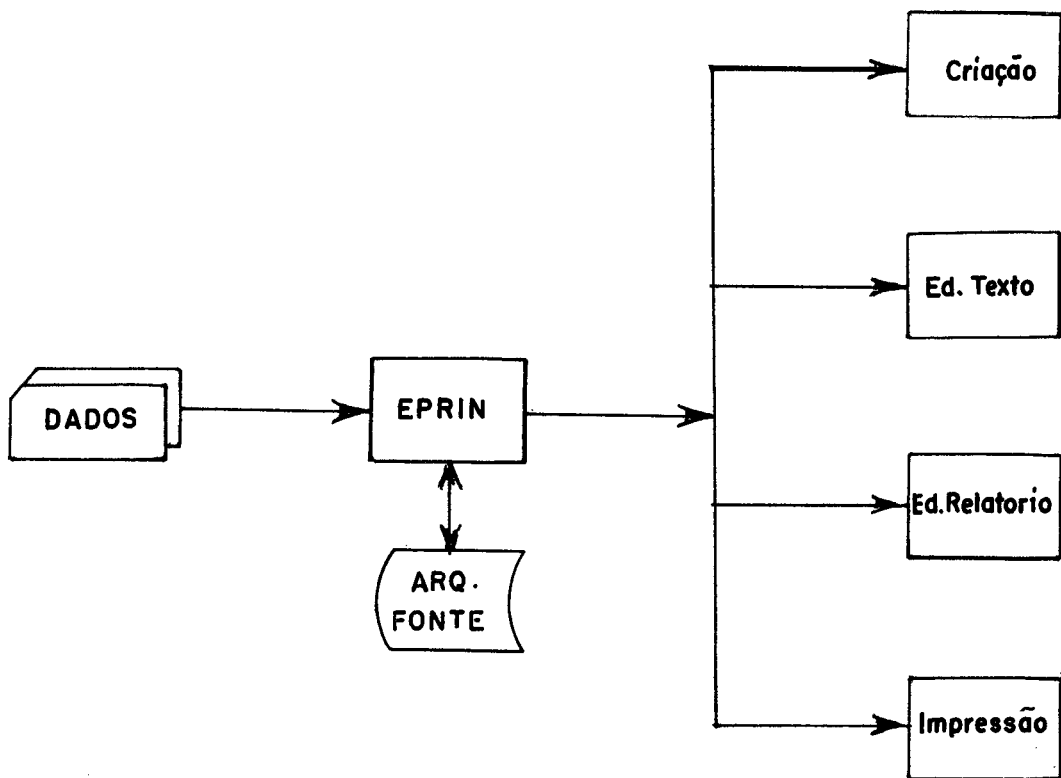


FIG. I.1 - FLUXOGRAMA GERAL DO EDIPRINT

pecificadas a seguir:

- 1 - Criação do Arquivo Fonte
- 2 - Editor de Texto
- 3 - Editor de Relatório
- 4 - Impressão do Arquivo Fonte

#### CRIAÇÃO DO ARQUIVO FONTE

Este bloco permite que o texto, inicialmente disposto em um padrão qualquer, seja fornecido ao sistema através de cartões perfurados e armazenado em um arquivo em disco. Tal arquivo é denominado Fonte, e contém as informações que deverão ser impressas no relatório segundo o padrão desejado.

#### EDITOR DE TEXTO

O Editor de Texto permite atualizar o arquivo Fonte a partir de modificações efetuadas no texto. Através de comandos de edição (seção 2.2.3.1) é possível inserir, apagar, substituir, renumerar registros e localizar sequências de caracteres incluídas nos mesmos.

#### EDITOR DE RELATÓRIO

O Editor de Relatório é responsável pela edição do texto no padrão de saída desejado. Os comandos de edição fazem parte das informações contidas no arquivo Fonte, e são analisados a cada registro lido. Conforme o comando de especificação encontrado no texto, o editor executa determinada sequência de operações relacionadas ao mesmo.

#### IMPRESSÃO DO ARQUIVO FONTE

Através do bloco de impressão do arquivo Fonte, o usuário certifica-se da exatidão das informações gravadas, e se após a atualização do arquivo, as alterações efetuadas foram implementadas de maneira correta. A impressão do arquivo Fonte fornece o número do registro (ou número da linha) em ordem crescen-

te e o texto contido naquele registro.

### 1.3 - FLUXOGRAMA GERAL DO EDIPRINT

O EDIPRINT é controlado por um programa supervisor denominado EPRIN, cuja função é analisar as opções escolhidas pelo usuário, e desviar o comando do sistema para um dos quatro procedimentos possíveis: Criação, Editor de Texto, Editor de Relatório e Impressão do arquivo. (Fig. I.1).

O sistema prevê situações em que o usuário poderá modificar o seu texto realizando correções, inserções e supressões de textos, sem que para isso seja necessário regravar toda a massa de dados inicial, com as devidas alterações.

O usuário ao utilizar o EDIPRINT poderá realizar as seguintes tarefas:

#### 1. OBTER UM TEXTO EM UM PADRÃO PRÉ-ESTABELECIDO:

Para isto ele deverá entrar com o texto disposto em um padrão qualquer através de cartões perfurados. Deverão ser incluídos no texto os comandos necessários para a formatação do relatório no padrão final de saída. Tais comandos de edição serão analisados posteriormente no parágrafo 2.2.3.

#### 2. ATUALIZAR O ARQUIVO FONTE

É possível manter um arquivo gravado no sistema e utilizá-lo com o único objetivo de realizar atualizações. Neste caso, o usuário deverá fornecer apenas os comandos necessários à correção do texto no arquivo, que serão analisados pelo Editor de Texto, e implementados pelo sistema.

#### 3. ATUALIZAR E EMITIR RELATÓRIO

É possível modificar sempre que necessário um texto inicial gravado no arquivo Fonte, e após cada alteração imprimí-lo em um formato de saída desejado.

## II - UTILIZAÇÃO DO SISTEMA

### 2.1 - NORMAS DE UTILIZAÇÃO DO SISTEMA

Um esquema mais detalhado da estrutura integral do EDI PRINT encontra-se disposto na figura II.1.

Para acessar o sistema EDIPRINT o usuário deverá entrar inicialmente com os seguintes cartões perfurados a partir da 1a. coluna:

1a. col.

//~~Ø~~JOB

11a. col.

~~Ø~~AA1~~Ø~~~~Ø~~AA2~~Ø~~~~Ø~~AA3~~Ø~~~~Ø~~AA4

//~~Ø~~XEQ~~Ø~~EPRIN

onde:

~~Ø~~ - zero

~~Ø~~ - espaço em branco

~~Ø~~AA1....~~Ø~~AA4 - são discos utilizados pelo sistema

Através do cartão // XEQ EPRIN o programa supervisor ' EPRIN é carregado na memória juntamente com todos os subprogramas a ele referenciados. De acordo com a opção fornecida no cartão seguinte, o programa supervisor desviará o comando para o bloco correspondente, onde a tarefa solicitada é executada.

Ao utilizar o EDIPRINT o usuário poderá optar por um ou mais dos seguintes procedimentos:

T - EDITOR DE TEXTO

R - EDITOR DE RELATÓRIO

C - CRIAÇÃO DO ARQUIVO FONTE

I - IMPRESSÃO DO ARQUIVO FONTE

É importante observar que os quatro procedimentos acima podem ser executados dentro de um mesmo processo - JOB - entretanto, é imprescindível que o usuário crie um arquivo Fonte ' (procedimento C) antes de optar por qualquer outro passo. Uma vez criado tal arquivo, os demais procedimentos poderão ser solicitados em qualquer ocasião bastando que o usuário referencie o nome

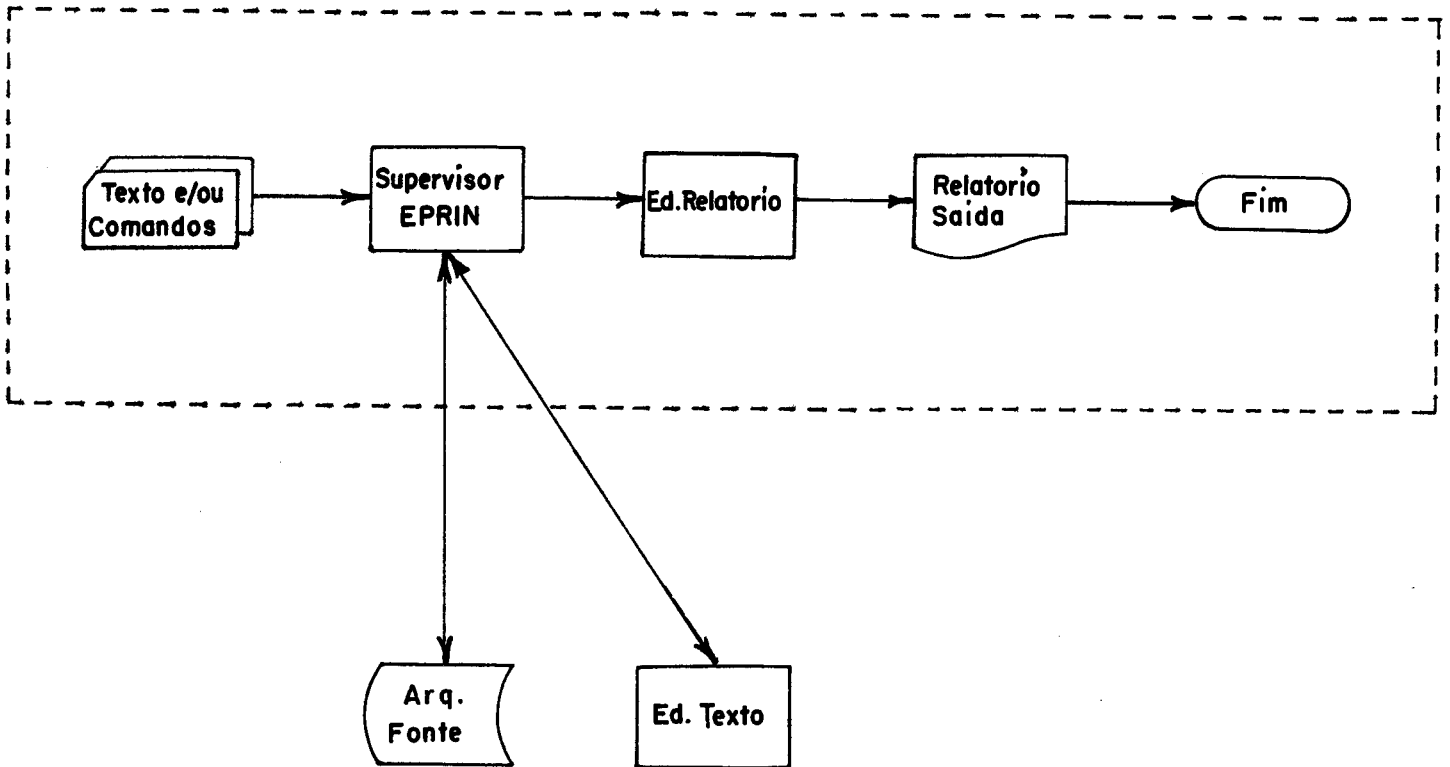


FIG. II . 1 - DETALHAMENTO DOS BLOCOS (PROCEDIMENTOS) QUE COMPOEM O SISTEMA EDIPRINT

do arquivo previamente gravado no sistema.

Para solicitar qualquer um dos procedimentos descritos, o usuário deverá utilizar cartões perfurados no seguinte formato:

\*n,NOME (II.1)

onde:

n - indica a opção escolhida T, R, C ou I (definido em 2.1)

NOME - nome designado para o arquivo, contendo no máximo 8 caracteres alfanuméricos.

Feita a escolha do procedimento desejado, o usuário deverá fornecer a seguir os cartões de dados relativos àquele item. Cada procedimento que o usuário deseje executar implicará a se entrar com um novo cartão do tipo (II.1).

A condição de término de procedimento para o sistema é dado através de um cartão // perfurado nas colunas 1 e 2 respectivamente.

O sistema emitirá mensagens explicativas de erro caso a sintaxe de qualquer comando seja diferente da estabelecida na definição do mesmo.

## 2.2 - DESCRIÇÃO DOS PROCEDIMENTOS

### 2.2.1 - CRIAÇÃO DO ARQUIVO FONTE

Ao escolher a opção C, o usuário estará recorrendo inicialmente ao EDIPRINT com o objetivo de criar um arquivo Fonte.

Os cartões contendo as informações que serão editadas deverão ser fornecidos logo após o cartão \*C, nome do arquivo, seguindo-se de um /\* nas colunas 1 e 2 respectivamente, para indicar ao sistema fim de entrada de dados.

EXEMPLO:

```
1a. col.                               11a. col.
//JOB                                   ØAA1ØØAA2ØØAA3ØØAA4

// XEQ EPRIN

*C, nome arq.                          (II..2)

.....
.....   cartões de dados contendo o texto em for
.....   mato livre
.....

/*                                       fim de entrada de dados

//                                       fim de procedimentos
```

No procediemnto de criação deve-se tomar o cuidado com a escolha do nome do arquivo. Como o espaço reservado para a criação dos mesmos é única no disco para todos os usuários, um mesmo NOME não poderá constar duas vezes no arquivo. Caso isto ocorra, o sistema emitirá mensagem de erro do tipo: "Já existe arquivo com esse nome utilize um outro".

## 2.2.2 - COMANDOS PARA UTILIZAÇÃO DO EDITOR DE TEXTO

Através do Editor de Texto o usuário poderá alterar in formações contidas no arquivo Fonte utilizando o procedimento T, fornecido ao sistema de forma semelhante ao item anterior. O usuáário deverá obrigatoriamente se referenciar a um arquivo previamente gravado de acordo com as especificações defendidas no paráágrafo 2.2.1.

### 2.2.2.1 - FACILIDADES DO EDITOR

O EDIPRINT permite efetuar as seguintes operações nos registros:

- 1 - Apagar o texto X no registro Y (ou mais registros)
- 2 - Eliminar no registro X a partir do texto A até o texto B no



registro Y

- 3 - Substituir | cadeia 1 | por | cadeia 2 | no registro X ( ou mais registros)  
CADEIA: sequência de caracteres
- 4 - Localizar cadeias de caracteres, apontando o número do registro em que se encontram
- 5 - Remover registro X (ou mais registros)
- 6 - Inserir registros utilizando a numeração padrão do sistema (de 100 em 100) ou outra especificada pelo usuário
- 7 - Renumerar registros

As facilidades descritas cobrem a maior parte das operações normalmente efetuadas pelos sistemas de edição de texto<sup>7,9</sup>. Estas opções deverão ser especificadas imediatamente após o cartão \*T, nome do arq, pois contem as alterações a serem efetuadas no arquivo. O cartão seguinte às alterações deverá ser um /\* nas colunas 1 e 2 respectivamente.

Durante a fase de implementação, levou-se em consideração a simplicidade dos comandos, de modo a facilitar a interação entre o usuário e o sistema. Para isso foram estabelecidas mnemônicos de um só caracter para a representação dos comandos de edição relacionados a seguir:

Comando	Mnemônico
Apagar	A
Eliminar	E
Substituir	S
Localizar	L (II.3)
Deletar (=remover)	D
Inserir	I
Renumerar	R

#### 2.2.2.2 - DESCRIÇÃO DOS COMANDOS

Os comandos anteriores são reconhecidos por comandos de

edição sempre que vierem precedidos do caracter "/" na 1a. coluna.

```
<INSTRUÇÃO> ::= /<COMANDO1>|/<COMANDO2>|/<COMANDO3>|/<COMANDO4>|
               /<COMANDO5>|/<COMANDO6>|/<COMANDO7>
<COMANDO1> ::= <MNEMÔNICO1><REGISTRO><SEQUÊNCIA>
<COMANDO2> ::= <MNEMÔNICO2><REGISTRO><CADEIA3>
<COMANDO3> ::= <MNEMÔNICO3><SEQUÊNCIA>
<COMANDO4> ::= <MNEMÔNICO4><REGISTRO>
<COMANDO5> ::= <MNEMÔNICO5><NÚMERO>|<MNEMÔNICO5><NÚMERO>$<NÚMERO>
               RO>
<COMANDO6> ::= <MNEMÔNICO6><TEXTO>
<COMANDO7> ::= <MNEMÔNICO7><CADEIA2>
<MNEMÔNICO1> ::= A
<MNEMÔNICO2> ::= S
<MNEMÔNICO3> ::= L
<MNEMÔNICO4> ::= D
<MNEMÔNICO5> ::= I
<MNEMÔNICO6> ::= R
<MNEMÔNICO7> ::= E
<REGISTRO> ::= <NÚMERO>-<NÚMERO>|<REGISTRO>,<NÚMERO>-<NÚMERO>|
               <REGISTRO>,<NÚMERO>|<NÚMERO>
<SEQUÊNCIA> ::= <CADEIA1>|<SEQUÊNCIA><CADEIA><DELIMITADOR>
<CADEIA2> ::= <NÚMERO><CADEIA1><NÚMERO><CADEIA>
<CADEIA3> ::= <CADEIA1><CADEIA><DELIMITADOR>
<NÚMERO> ::= <DÍGITO>|<NÚMERO><DÍGITO>
<TEXTO> ::= <NÚMERO>-<NÚMERO>,<NÚMERO>,<NÚMERO>
<CADEIA1> ::= <DELIMITADOR><CADEIA><DELIMITADOR>
<CADEIA> ::= <CARACTER>|<CADEIA><CARACTER>
<DELIMITADOR> ::= ·|<|(|+|&|*|)|/|%|#|@|·|=|:
<DÍGITO> ::= 0|1|2|3|4|5|6|7|8|9
<CARACTER> ::= <CONJUNTO DE CARACTERES EBCDIC>
<LETRA> ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|
           Y|Z
```

Obs: A impressão do conjunto de caracteres EBCDIC é restrita à cadeia de impressão utilizada pela impressora, e por este motivo nem todos os símbolos podem ser impressos.

## 2.2.2.3 - DEFINIÇÕES

## 1. BLOCO DE REGISTROS

São combinações das estruturas:

$$n_k^- n_{x+k} \quad (\text{II.4})$$

$$n_k, n_{x+k} \quad (\text{II.5})$$

O caráter '-' indica que uma determinada operação será efetuada a partir do registro  $n_k$  até o registro  $n_{x+k}$ . Esta estrutura deverá ser precedida e seguida de vírgula (,), sempre em ordem crescente dos números de registros. É vedada a utilização de  $n_x^- n_y^- n_z$ .

O carácter ',' indica que uma determinada operação será efetuada apenas nos registros  $n_k$  e  $n_{x+k}$ .

As operações efetuadas pela maioria dos comandos podem ser realizadas sobre diversas combinações entre si das estruturas II.4 e II.5, limitando-se apenas ao tamanho do cartão. A única exceção se faz ao comando renumerar, cuja estrutura  $n_k^- n_{x+k}$  obedece ao formato estabelecido na descrição do mesmo.

Exemplo:

/An<sub>1</sub><sup>-</sup> n<sub>10</sub>, n<sub>16</sub>, n<sub>23</sub>, n<sub>40</sub><sup>-</sup> n<sub>45</sub> DxxxxxD  
A

O texto A será apagado do registro  $n_1$  ao  $n_{10}$ , nos registros  $n_{16}$ ,  $n_{23}$  e do  $n_{40}$  ao  $n_{45}$ .

## 2. DELIMITADOR

Chama-se Delimitador (D) a um carácter do conjunto determinado por:

$$D = \{', <, (, +, \&, *, ), /, \%, \#, @, ', =, : \} \quad (\text{II.6})$$

Qualquer um dos caracteres do conjunto D poderá ser utilizado na estrutura dos comandos desde que obedeça, as seguin-

tes restrições:

- a - Entre cada texto o caracter utilizado deve ser o mesmo. Isto equivale a dizer que para a execução de um comando, o delimitador escolhido deve ser mantido sempre entre os textos daquele comando.
- b - Os caracteres &, <, %, #, @, : poderão ser utilizados como delimitadores, mas quando presentes em um texto não serão impressos devido a restrições da cadeia de impressão da IBM-1403, impressora utilizada no computador IBM-1130 disponível no Núcleo de Computação da UFPb.

### 3. CHARACTER ESPECIAL

Chama-se caracter especial a um caracter do conjunto ' (E) determinado por:

$$E = \{ , , - , \$ , \emptyset \} \quad (\text{II.7})$$

$\emptyset$  - caracter branco

Nenhum dos elementos de E pode ser utilizado como delimitador na sintaxe de um comando.

#### 2.2.2.4 - FORMATO DE CHAMADA DOS COMANDOS

Todos os comandos deverão ser perfurados a partir da coluna 1, limitando-se às 80 colunas correspondentes a um cartão. Se um único cartão não for suficiente para conter o comando, este poderá ser repetido no(s) cartão(ões) seguinte(s) com o restante das informações. A 1a. coluna deverá conter sempre o caracter '/', seguido do mnemônico correspondente ao comando solicitado.

$$1. \quad /An_k - n_{x+k} DxxxxxDxxxxxD... \quad (\text{II.8})$$

Este comando indica que os textos entre delimitadores serão apagados a partir do registro  $n_k$  até o registro  $n_{x+k}$ . É per

mitida a utilização de Blocos de Registros.

2. 
$$/En_k \underset{A}{Dxxxxxx} Dn_{x+k} \underset{B}{Dxxxxxx} D \quad (II.9)$$

Este comando indica que todo o conteúdo entre o texto A no registro  $n_k$  e o texto B (intervalo fechado) no registro  $n_{x+k}$  será eliminado. Não é permitida a utilização de Blocos de Registros.

3. 
$$/Sn_k - n_{x+k} \underset{A}{Dxxxxxx} \underset{B}{Dxxxxxx} D \quad (II.10)$$

O texto A será substituído pelo texto B em toda estrutura determinada pelos blocos de registro. Se o texto B for maior do que o texto A, a ponto de não caber no mesmo registro, um novo registro é incluído, e o número do mesmo será igual a média aritmética entre os registros anterior e próximo. Se não for possível inseri-lo devido a incompatibilidade na numeração do arquivo, o sistema emite mensagem de erro.

4. 
$$/LDxxxxDxxxxD... \quad (II.11)$$

Este comando localiza as diversas cadeias entre delimitadores, indicando o número do registro em que se encontram. Caso a(s) mesma(s) não seja(m) encontrada(s), o sistema envia mensagem avisando que determinada(s) cadeia(s) não foi(ram) localizada(s).

5. 
$$/Dn_k - n_{x+k} \quad (II.12)$$

O objetivo deste comando é fazer com que todos os registros contidos na estrutura de bloco sejam removidos do arquivo Fonte. Caso algum registro contido na estrutura de blocos não exista no arquivo, o programa acusa tal ocorrência e não efetua a operação.

6. O COMANDO /I APRESENTA 2 MODALIDADES:

a - 
$$/In \quad (II.13)$$

Este comando é responsável pela inserção de registros

adicionais ao arquivo.

O valor  $n$  indica o número a partir do qual os registros serão inseridos com o incremento padrão de 100. Logo após o comando devem ser colocados cartões com os registros a serem inseridos. O final dos registros é reconhecido por um cartão com /\* nas colunas 1 e 2.

b -            / $In_i \$n_j$    (II.14)

Este comando é semelhante a estrutura anterior com a diferença de que a numeração do 1º registro inserido recebe o valor  $n_i$  e os demais são incrementados de  $n$ .

Em ambos os casos (II.13 e II.14) o sistema emite mensagem de erro caso a numeração seja incompatível com a do arquivo. O registro que provocou tal ocorrência não é inserido.

7.            / $Rn_k^{-n_1, n_i, n_j}$    (II.15)

O comando /R renumera registros a partir de  $n_k$  até atingir o registro  $n_1$ . O registro  $n_k$  passará a ter o número de registro  $n_i$  (ou número de linha) e a partir daí os demais registros serão incrementados de  $n_j$  até que o registro  $n_1$  seja atingido. Caso a numeração se torne incompatível com a do arquivo, o sistema emite mensagem de erro. Se isto ocorrer, o registro que provocou tal incompatibilidade não é renumerado. O comando em questão só admite a estrutura de bloco  $n_k^{-n_1}$ , onde  $n_k < n_1$ .

#### 2.2.2.5 - EXEMPLO DE UM PROCEDIMENTO DE CHAMADA PARA O EDITOR DE TEXTO:

Considere o arquivo fonte denominado TESTE a ser alterado pelo Editor de Texto antes da emissão do relatório no padrão de saída solicitado.

```
col.1   col.11
//JOBØØAA1ØØAA2ØØAA3ØØAA4 ..... Início do processo
//ØXEQØEPRIN ..... Chama sistema EDI
                                           PRINT
*T, TESTE ..... Chama ed. texto pa
```

ra alterar o arq.  
TESTE

/A400-600,800,1000%COMPILADORZ ..... Apaga o texto 'COM  
PILADOR' nos regis-  
tros de 400 a 600,  
no registro 800 e  
no 1000

/L+BRANCO+CEU+ ..... Localiza os textos  
entre '+' no arqui-  
vo

/I650\$10 ..... Insere registros com  
incremento de 10 em  
10 a partir do reg.  
650

ABCDEFGH ..... Equivale ao reg. 650

IJKLMNOPQRSTU ..... Equivale ao reg. 660

VXZ ..... Equivale ao reg. 670

/\* ..... Fim de dados

/D1100-1300 ..... Registros entre 1100  
e 1300 serão remo-  
vidos do arquivo '  
TESTE

/S300=FORTRAN=COBOL=..... Substituí no reg.300  
FORTRAN por COBOL

/E800\*CARACTER\*1000\*VÍRGULA\* ..... Elimina todo o con-  
teúdo entre carac-  
ter (reg. 800)e vír-  
gula (reg. 1000)

/\* ..... Fim de correções

\*I,TESTE ..... Imprime arquivo

\*R,TESTE ..... Imprime relatório  
no formato solici-  
tado

// ..... Fim de processo

O EDIPRINT emitirá mensagens explicativas de erros caso estes ocorram na sintaxe dos comandos.

### 2.2.3 - COMANDOS PARA A UTILIZAÇÃO DO EDITOR DE RELATÓRIO

O Editor de Relatório permite que o usuário obtenha a impressão de um texto em um padrão de saída pré-estabelecido.

Os dados gravados inicialmente no arquivo Fonte (em formato livre) deverão conter todos os comandos necessários para a impressão do texto no padrão de saída desejado. Quando o usuário solicitar o procedimento R através do cartão \*R,nome do arq., o EDITOR de RELATÓRIO é chamado para analisar os comandos de edição contidos nos registros do arquivo solicitado.

#### 2.2.3.1 - COMANDOS DE EDIÇÃO

Os comandos de edição foram estabelecidos segundo um padrão simples, visando facilitar a sua utilização por parte dos usuários.

O formato do cartão de especificação é o seguinte:

\$XXn (II.16)

onde:

\$ - caracter especial que indica comando de edição

XX - caracteres literais que identificam um comando de edição

n - apresenta 4 especificações distintas:

␣ - caracter branco

x

x,y

x,y,z, onde x,y,z representam números

#### 2.2.3.2 - DESCRIÇÃO DOS COMANDOS

1 - \$IR . . . . . \$FR

IR e FR indicam respectivamente início e fim de rodapé.

O conteúdo entre estes 2 comandos será impresso no final da página



gina, no local determinado para a impressão do rodapé (comandos de controle-seção 2.2.3.4)

2 - \$ITx,y . . . . . \$FT

IT e FT indicam respectivamente início e fim de título.

O conteúdo entre estes 2 comandos será impresso no início de cada página, x linhas após a impressão da numeração da mesma, a partir da margem esquerda determinada por y.

3 - \$PE . . . . . \$RE

PE e RE indicam respectivamente parada de edição e reinício de edição.

O conteúdo entre os 2 comandos será impresso exatamente como se encontra no texto. Nenhum comando de edição existente entre eles será analisado pelo EDITOR de RELATÓRIO.

4 - \$PLx

Indica que serão saltadas x linhas a partir da linha atual impressa.

5 - \$PI . . . . . \$RI

PI e RI indicam respectivamente parada de impressão e reinício de impressão. Todo texto contido entre estes 2 comandos não será impresso.

6 - \$NCx,y,z

Este comando indica que as próximas x linhas serão impressas a partir da coluna y até a coluna z do relatório. Quando x=0 o comando redefine as margens direita e esquerda (item 4-seção 2.2.3.4).

7 - \$NPx

Este comando indica mudança de parágrafo a partir da coluna x. Quando x não vier especificado o parágrafo continua o mesmo definido pelo sistema ou pelo usuário (seção 2.2.4.1)

8 - \$IO . . . . . \$FO

IO e FO indicam respectivamente início e fim de "overprint" (impressão em negrito). O conteúdo existente entre os 2 co-

mandos será impresso em negrito.

9 - \$AP

Este comando indica que a letra seguinte virá acentuada ( acento grave ou agudo).

10 - \$CE

Este comando indica que a letra C a seguir será impressa com cedilha.

11 - \$II . . . . . \$FI

II e FI indicam respectivamente início e fim de índice. O conteúdo existente entre os 2 comandos constará apenas no índice por página, não saindo no texto editado. No final da edição uma página é saltada, porém sem ser numerada, e o índice é impresso.

12 - \$IX . . . . . \$FX

IX e FX indicam respectivamente início e fim de referência cruzada. Todo conteúdo existente entre estes 2 comandos constará também no índice de palavra chave, que será impresso no final da edição do relatório após a mudança de página, e em ordem alfabética. A página é numerada, e cada linha impressa do índice é acompanhada do número da página onde o texto se encontra.

13 - \$BRx

Este comando permite que sejam deixados x espaços em branco antes que a próxima sequência de caracteres seja impressa na linha.

14 - \$MP

Este comando permite que seja efetuada mudança de página.

### 2.2.3.3 - DADOS DE CONTROLE

É possível estabelecer parâmetros no EDIPRINT que permaneçam constantes durante toda a fase de edição de um relatório. Tais parâmetros, denominados dados de controle, quando for

precisos são poderão ser modificados atravẽs de comandos de ediçãõ contidos no texto. Caso não sejam fornecidos inicialmente, são assumidos valores padrões pelo sistema (seçãõ 2.2.4.1)

O formato de tais cartões de controle ẽ dado pela especificaçãõ (II.17) a seguir:

$n-n_1, n_2, n_3$  (II.17)

$n$ : Nũmero correspondente a entrada do cartãõ de controle, seguido do caracter '-'.  
 $n$  varia de 1 a 14 e ẽ perfurado a partir da coluna 1.

Os nũmeros de um sãõ caracter (de 1 a 9) são perfurados na 2a. coluna, ficando a 1a. em branco. Os demais (10 a 14) são perfurados a partir da 1a. coluna.

$n_1, n_2, n_3$ : Valores numẽricos que especificam os dados de controle.

Nem todos os cartões de controle necessitam dos 3 parãmetros (seçãõ 2.2.4.1), variando conforme a especificaçãõ de cada um.

#### 2.2.3.4 - DESCRIÇãõ DOS COMANDOS DE CONTROLE

1- $n_1$   $n_1$  determina o nũmero de cõpias do relatõrio final de saída.

2- $n_1$   $n_1$  especifica o nũmero de linhas em branco entre cada linha impressa.

3- $n_1, n_2, n_3$  numera pãginas em algarismos arãbicos de acordo com os parãmetros:

$n_1$ : nũmero da linha onde o nũmero da pãgina serã impresso.

$n_2$ : nũmero da coluna onde o nũmero da pãgina serã impresso.

$n_3$ : determina o nũmero da pãgina inicial a partir do qual a numeraçãõ irã começar.

- 4- $n_1, n_2$   $n_1$  e  $n_2$  determinam o posicionamento das margens esquerda e direita respectivamente.
- 5- $n_1, n_2$  determina o posicionamento da impressão do rodapé na linha  $n_1$  e margem esquerda na coluna  $n_2$ .
- 6- $n_1, n_2$  especifica a impressão do relatório somente da linha  $n_1$  até a linha  $n_2$ . Esta opção só é válida se o usuário solicitar cópia em disco do relatório editado (opção 11)
- 7- $n_1$   $n_1$  determina o número de linhas por página.
- 8- $n_1$   $n_1$  determina a linha topo, isto é, a linha em que será impressa a primeira frase do relatório.
- 9- $n_1$   $n_1$  especifica a coluna correspondente ao ponto parágrafo.
- 10- $n_1$  Se  $n_1=0$  a saída é feita pela impressora 1403.  
Se  $n_1=1$  a saída é feita pela console.
- 11- $n_1$  Se  $n_1=0$  a saída do relatório editado não é gravada em disco.  
Se  $n_1=1$  a saída do relatório editado é gravada em disco.
- 12- $n_1$  Se  $n_1=1$  o índice é listado no final da edição.  
Se  $n_1=0$  o índice não é listado.
- 13- $n_1$  Se  $n_1=1$  o índice de palavra chave é listado no final da edição.  
Se  $n_1=0$  o índice de palavra chave não é listado.
- 14- $n_1$  Se  $n_1=1$  não há separação de sílabas no relatório final.  
Se  $n_1=0$  há separação de sílabas no relatório sempre que necessário.

#### 2.2.4 - NORMAS GERAIS PARA CODIFICAÇÃO DOS COMANDOS

Os comandos de edição são gravados juntamente com tex-

to a ser formatado durante a fase de criação do arquivo Fonte.

EXEMPLO:

O COMPILADOR TRANSFORMA O PROGRAMA FONTE EM LINGUAGEM DE MÁQUINA. ESTA É A CHAMADA FASE DE COMPILAÇÃO.

Se desejarmos que o texto anterior seja impresso 2 linhas a partir da última linha impressa, em negrito e em um parágrafo diferente do padrão, deveremos entrar com o seguinte texto:

**\$PL2\$NP20\$IO O COMPILADOR TRANSFORMA O PROGRAMA FONTE EM LINGUAGEM DE M\$APAQUINA\$F0. ESTA \$APE A CHAMADA FASE DE \$IOCOMPILA\$CECAO\$F0.**

Os espaços em branco existentes no texto não serão con-servados, a menos que sejam solicitados através do comando \$BR.

#### 2.2.4.1 - COMANDO DE CONTROLE

Os cartões de controle seguem imediatamente o cartão de chamada do EDITOR de RELATÓRIO através do cartão \*R,nome arq.,mu ma ordem qualquer.

O usuário deverá especificar apenas os controles que de sejar, já que o EDIPRINT estabelece valores a todos os itens que não forem fornecidos.

O número máximo de linhas impressas em uma folha de relatório é 60, e cada linha permite a impressão de até no máximo 120 caracteres.

Os valores estabelecidos pelo sistema quando os cartões de controle correspondentes não são fornecidos pelo usuário atendem às especificações de uma folha impressa no formato A4. Os valores são relacionados abaixo:

- 1-1 (nenhuma outra cópia é emitida além do relatório final)
- 2-1 (espaço simples entre uma linha e outra)
- 3-3,93,1 (numeração na linha 3 coluna 93, a partir de 1)
- 4-30,93 (margem esquerda na coluna 30, direita na coluna 93)
- 5-42,30 (rodapé inicia na linha 42, coluna 30)

- 6-0 (todo texto é impresso)
- 7-40 (número máximo de linhas por página)
- 8-5 (tôpo da página na linha 5)
- 9-35 (parágrafo na coluna 35)
- 10-0 (saída pela impressora)
- 11-0 (a saída do relatório não é gravada)
- 12-0 (o índice por página não é listado)
- 13-0 (o índice por palavra chave não é listado)
- 14-0 (separa sílabas sempre que necessário)

O sistema emitirá mensagens sempre que o usuário cometer qualquer um dos seguintes tipos de incorreções:

#### 1. Erros de Sintaxe

a - n varia de 1 a 14 e deve ser seguido do caracter '-'. Os números de 1 são caracter (1 a 9) são perfurados a partir da coluna 2, ficando a 1a. em branco. Os demais (de 10 a 14) são perfurados a partir da 2a. coluna.

b - Não deve existir brancos entre os caracteres que especificam os dados no cartão de controle.

c - Somente o cartão de controle 3 possui 3 especificações numéricas.

d - Somente os cartões de controle 4, 5 e 6 possuem 2 especificações numéricas.

e - Os cartões de controle de 7 a 14 possuem apenas 1 especificação numérica, sendo que de 10 a 14 só podem ter informações 0 ou 1.

#### 2. Limite de Variação de Parâmetros

a - O número máximo de linhas permitido por página é 60. Se ultrapassar este limite através de cartões de controle o programa acusa erro. Se isto ocorrer através de um comando de edição contido no texto, salta o número de linhas necessárias até completar

o limite de linhas determinado por página.

b - O limite máximo permitido para margem esquerda é 120. Se ultrapassar este limite através de cartões de controle o programa acusa erro. Se isto ocorrer através de comandos de edição contidos no texto, limita a margem direita para 120.

O limite mínimo para a margem esquerda é 1, e o mesmo procedimento anterior é tomado: ao invés de 120 o valor final é 1.

#### 2.2.5 - PROCEDIMENTO DE IMPRESSÃO

Este procedimento, obtido a partir do cartão \*I, nome do arquivo, permite ao usuário certificar-se da exatidão das informações contidas no arquivo Fonte. Este procedimento poderá ser solicitado após a criação de um arquivo ou quando alterações forem efetuadas no mesmo através do EDITOR de TEXTO.

A impressão do arquivo deverá fornecer as seguintes informações:

- nome do arquivo
- número do registro em ordem crescente
- conteúdo do registro

Caso o nome do arquivo solicitado não seja encontrado, o programa acusará erro.

#### 2.3 - OBSERVAÇÕES GERAIS

O relatório final poderá ser obtido também através de um arquivo auxiliar denominado "BACK-UP". Tal arquivo é único para todos os usuários e será mantido no sistema até que, em outro processo, o mesmo seja solicitado por outro usuário.

O objetivo do "BACK-UP" é permitir a emissão de cópias sucessivas do relatório final, sem que para isto, o EDITOR de RELATÓRIO precise processar o texto a cada cópia desejada. Esta é a razão pela qual tal arquivo não deve ser solicitado quando o usuário desejar apenas uma cópia do texto editado.

O arquivo geral que armazena os arquivos Fonte de todos os usuários do sistema tem tamanho limitado. Quando este ponto é atingido, o EDIPRINT emite mensagem de tal ocorrência, e automa-

ticamente lista os nomes de todos os arquivos Fonte gravados no sistema. Cabe ao usuário a tarefa de determinar os que não são mais utilizados e removê-los do sistema através do Editor de Texto, com o objetivo de inserir o próprio arquivo no sistema. Isto é realizado através do comando Deletar, que permite remover todos os registros do arquivo.



### III - DETALHAMENTO DO SISTEMA

#### 3.1 - PROGRAMA SUPERVISOR EPRIN

O programa supervisor EPRIN tem por objetivo analisar as opções fornecidas através do cartão \*n,nome arq., e desviar o comando do sistema para executar cada tarefa correspondente. Conforme já foi visto anteriormente, para n= T,R,C,I os procedimentos solicitados são respectivamente: Editor de Texto, Editor de Relatório, Criação e Impressão do arquivo. (Vide Apêndice I.1)

#### 3.2 - CRIAÇÃO DO ARQUIVO FONTE

Arquivo Fonte é o resultado da gravação do conjunto de informações fornecidas ao sistema que poderá eventualmente ser editado se assim o usuário o desejar. No procedimento de criação os dados são lidos e automaticamente gravados em disco, para utilização posterior dos procedimentos relativos ao Editor de Texto, de Relatório ou Impressão.

O procedimento de criação do arquivo Fonte é imprescindível para que qualquer um dos procedimentos anteriormente citados possam ser executados. O arquivo Fonte recebe um nome fornecido através do cartão \*C,nome arq., que deverá ser referenciado sempre que o arquivo for impresso ou editado.

##### 3.2.1 - ARQUIVOS UTILIZADOS

###### 3.2.1.1 - RESERVATÓRIO DE NÓS

Denomina-se reservatório de nós a uma área fixa em disco formada de registros vazios, a partir dos quais os diversos arquivos Fonte dos usuários são formados. O reservatório de nós, também chamado de espaço disponível, é constituído de registros contendo 44 palavras cada um.

Inicialmente o reservatório de nós apresenta uma estrutura de lista linear, onde cada nó aponta sempre para o endereço do registro seguinte, e o último para zero. Este procedimento é realizado uma única vez em todo o sistema. DISP é o ponteiro que indica sempre o 1º registro disponível, denominado topo do reservatório.

Quando um registro é solicitado, o sistema fornece aquele cujo endereço é apontado por DISP. O ponteiro passará então a apontar o registro seguinte. Por outro lado, quando um registro for removido do arquivo ele será devolvido para o reservatório, e DISP passará a apontá-lo.

### 3.2.1.2 - ARQUIVO FONTE

Arquivo Fonte é todo arquivo criado pelo usuário com o nome fornecido através do cartão \*C, nome arq., que deverá ser referenciado sempre que o usuário solicitar os procedimentos de Impressão, Editor de Texto ou de Relatório.

Os registros que compõem o arquivo Fonte são retirados do reservatório de nós e encadeados em uma estrutura de lista circular duplamente ligada.

Inicialmente o 1º registro aponta sempre para ele mesmo, e à medida que novos registros são inseridos no arquivo, estes vão sendo encadeados circularmente na lista.

Cada registro é formado de 44 palavras distribuídas em 4 campos distintos:

campo 1: constituído de 2 palavras, este campo armazena a numeração lógica de cada registro, também chamado de número de linha. Inicialmente o arquivo é criado com uma numeração padrão, onde a diferença entre o número lógico de um registro e de seu registro anterior é 100.

campo 2: constituído de 1 palavra, indica o endereço do próximo registro no arquivo.

campo 3: constituído de 1 palavra, indica o endereço do registro anterior no arquivo.

campo 4: as 40 palavras reservadas para este campo armazenam o conteúdo de um cartão (80 colunas) com as informações inseridas no formato A2 (2 caracteres por palavra). O texto aqui armazenado deverá conter as especificações dos comandos

dos de edição, caso o usuário deseje utilizar o Editor de Relatório para formatá-lo.

### 3.2.1.3 - TABELA DE ÍNDICES

A Tabela de Índices tem por objetivo otimizar o processo de busca de registros no arquivo Fonte, tornando esta tarefa mais rápida e eficiente.

A tabela é constituída por registros de 44 palavras - retirados a partir do reservatório de nós - e dispostos em uma lista circular duplamente ligada. Cada registro é dividido em 2 campos distintos:

campo 1: constituído por 2 palavras, indica respectivamente o endereço do registro anterior e seguinte na tabela.

campo 2: as 42 palavras deste campo subdividem-se em 14 subcampos de 3 palavras cada um, distribuídos da seguinte forma:

a 1ª. palavra contém o endereço do registro no arquivo Fonte, cujo número de linha é fornecido pelas 2 palavras seguintes.

Durante a fase de criação do arquivo Fonte cada registro da tabela de Índices é preenchida até o 10º subcampo. Os 4 últimos ficam sempre vazias, reservados para futuras inserções de registros no arquivo Fonte. Inicialmente, o 1º registro da tabela contém os números de linha que variam entre 100 e 1000, o 2º entre 1100 e 2000, etc.

### 3.2.1.4 - DIRETÓRIO DE ARQUIVOS

O objetivo do Diretório é relacionar os nomes dos arquivos Fonte gravados no sistema. Cada registro é constituído de 2 campos num total de 5 palavras.

campo 1: constituído de 1 palavra, aponta o endereço do registro inicial da tabela de Índices corres-

pondente ao arquivo Fonte solicitado.

campo 2: constituído de 4 palavras, contem o nome do arquivo Fonte gravado no formato A2 (2 caracteres por palavra).

O Diretório de arquivos é gravado numa área separada em disco, e o método de procura utilizado é sequencial. Após a gravação de um registro no diretório, o registro seguinte é zerado para indicar o próximo registro disponível.

O 1º registro do arquivo é utilizado para conter o endereço do 1º registro disponível do reservatório de nós. Sempre que um registro é solicitado ou devolvido ao reservatório este campo é atualizado.

### 3.2.2 - DESCRIÇÃO DOS PROCEDIMENTOS DE CRIAÇÃO DO ARQUIVO FONTE

Quando a opção \*C,nome arq. é reconhecida pelo sistema, o programa supervisor EPRIN (Apêndice I.1) desvia o comando para a rotina responsável pela criação do arquivo Fonte. CRIA (Apêndice I.2) é a rotina capaz de efetuar este processo. A cada passo lê um cartão com as informações a serem inseridas, retira nó do reservatório e chama a rotina INCLU (Apêndice I.3) que efetua a inclusão do registro no arquivo Fonte e grava o endereço e número da linha correspondente no tabela de Índices. Quando o 1º subcampo é preenchido, um outro nó do reservatório é retirado, para a inclusão de um novo registro na tabela de Índices. Inicialmente a numeração do arquivo Fonte é feita de 100 em 100 podendo ser alterada posteriormente através do EDITOR DE TEXTO. Após a inclusão de registros, a rotina INCLU efetua o ajuste de ponteiros no arquivo Fonte e Tabela de Índices, retornando à rotina CRIA que realizará o mesmo procedimento para o próximo cartão lido.

### 3.3 - EDITOR DE TEXTO

Tendo sido analisada a opção \*T,nome arq., o programa supervisor EPRIN desvia o comando do sistema para o EDITOR DE TEXTO.

Denomina-se TEXTO (Apêndice II.1) à rotina principal do

EDITOR DE TEXTO cujo objetivo é analisar os comandos de edição solicitados (seção 2.2.2.4) e desviar o fluxo do sistema para as rotinas correspondentes (FIG. III.1)

### 3.3.1 - DESCRIÇÃO DOS PROCEDIMENTOS UTILIZADOS

#### 3.3.1.1 - COMANDO APAGAR ( $/A_{n-x+k} D_{xxxxxD}$ )

A rotina APAGA (Apêndice II.2) tem por objetivo apagar os textos entre delimitadores nos registros contidos nos blocos de registros. A 1a. etapa do procedimento é a execução das rotinas BLOCO (Apêndice II.9) e STRIN (Apêndice II.10) responsáveis pela separação dos blocos de registros e dos textos entre delimitadores respectivamente. A 2a. etapa, executada pela rotina COMD1 (Apêndice II.13), refere-se a localização dos registros ou números de linha, já separados por BLOCO, através das rotinas BUSCA (Apêndice II.11) e ACHAR (Apêndice II.12). A 1a. procura o registro na tabela de índices onde o número da linha está gravado. A 2a. rotina percorre os subcampos daquele registro na tabela, até encontrar o número desejado, localizando o seu endereço no arquivo Fonte. Finalmente, a 3a. etapa refere-se a remoção do(s) texto(s) guardado(s) por STRIN, no registro recém encontrado. Esta tarefa é executada pela rotina CLEAN (Apêndice II.17), cuja função é pesquisar o texto no registro - rotina TESTA (Apêndice II.19) - preenchendo-o com brancos, e rearrumar o registro, isto é, deslocar o conteúdo à direita do texto localizado para à esquerda, a partir do ponto inicial do mesmo.

#### 3.3.1.2 - COMANDO ELIMINAR ( $/E_{n_k} D_{xxxxx} : D_{n_{x+k}} D_{xxxxxD}$ ) A x+kB

A rotina ELIMI (Apêndice II.3) tem por objetivo eliminar o conteúdo existente entre os textos A e B a partir do registro  $n_k$  até o  $n_{x+k}$ . Inicialmente as rotinas BLOCO e STRIN são executadas 2 vezes consecutivamente, com o objetivo de separar o registro  $n_k$  e o texto A e o registro  $n_{x+k}$  e o texto B. A rotina COMD3 (Apêndice II.15) é responsável pela remoção dos textos nos registros. Esta operação é executada da seguinte forma: procura o registro  $n_k$  através das rotinas ACHAR e BUSCA, localizando o texto A no mesmo. Se o registro  $n_{x+k}$  for igual ao  $n_k$ , procura o tex

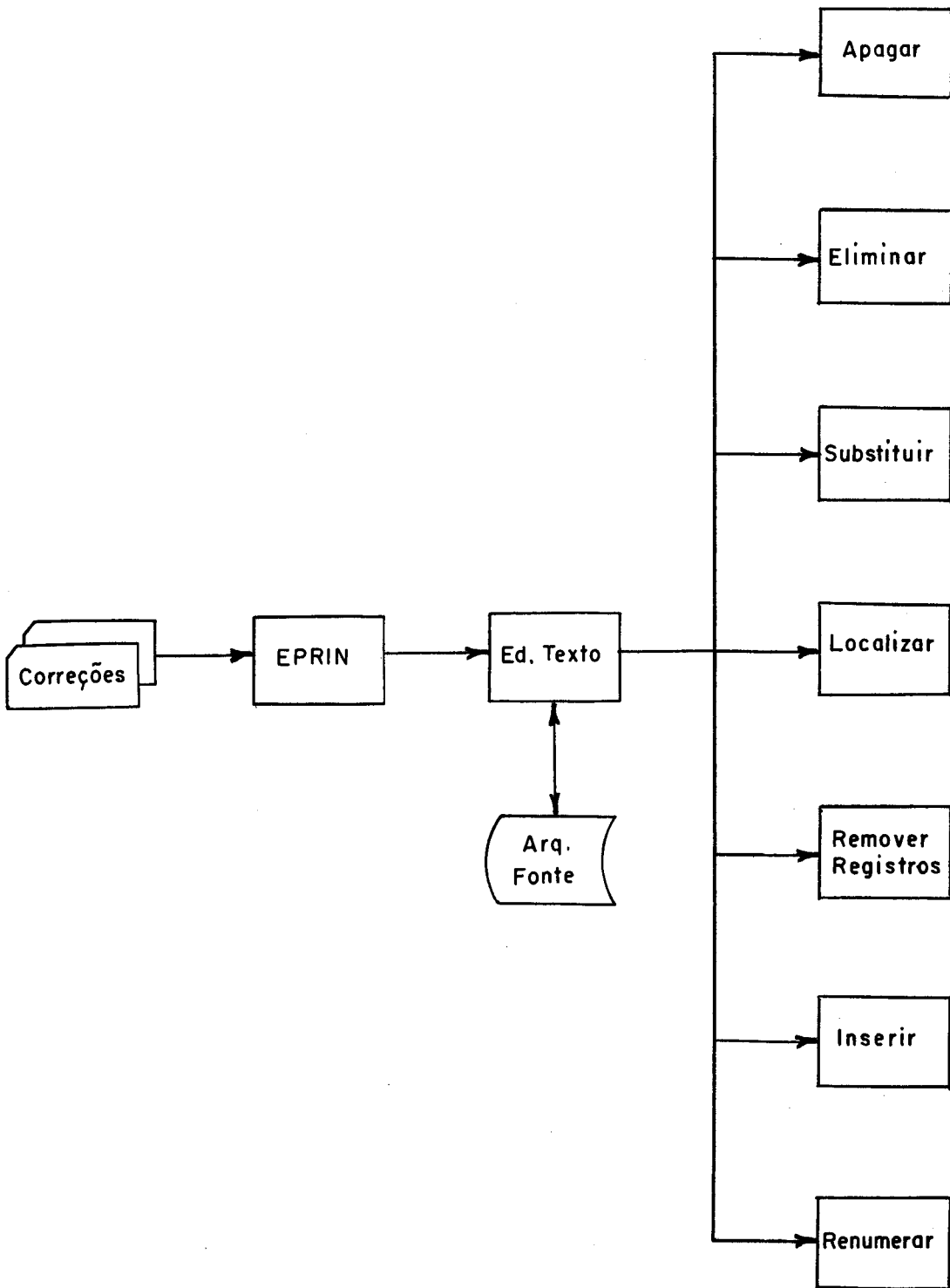


FIG. III. 1 - FLUXOGRAMA GERAL DO EDITOR DE TEXTO

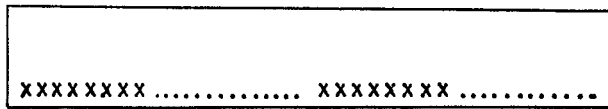
to B no mesmo registro, coloca brancos no conteúdo entre os textos A e B, rearrumando-o de maneira semelhante ao item anterior (FIG. III.2). Se  $n_{x+k}$  for diferente de  $n_k$ , procura o texto B no registro  $n_{x+k}$  e coloca brancos no registro  $n_k$  a partir do texto A até o final do mesmo e no registro  $n_{x+k}$  até atingir o final do texto B. Rearrumba os registros  $n_k$  e  $n_{x+k}$ , e se possível remove registros desnecessários do arquivo (FIG. III.3 e III.4).

### 3.3.1.3 - COMANDO DELETAR (/Dx<sub>n</sub> - x<sub>n+k</sub>)

A rotina DELET (Apêndice III.4) é responsável pela remoção dos registros agrupados nos blocos de registros. A 1a. etapa do procedimento é a separação dos blocos de registros através da rotina BLOCO. A 2a. etapa, efetuada pela rotina APREG (Apêndice II.18), tem por objetivo localizar cada registro (rotinas ACHAR e BUSCA), e removê-lo do arquivo Fonte. Para cada registro removido um nó é devolvido ao reservatório, e a entrada correspondente àquele registro no tabela de índices também é apagada. No caso de um arquivo inteiro ser removido do sistema, o seu nome também deixará de constar no diretório de arquivos.

### 3.3.1.4 - COMANDO INSERIR (TIPO 1: /In<sub>i</sub>; TIPO 2: /In<sub>i</sub>\$n<sub>j</sub>)

A rotina INSER (Apêndice II.5) tem por objetivo incluir novos registros em um arquivo já existente no sistema. Se o comando for do tipo 1, a numeração entre os novos registros terá um incremento de 100. Se for do tipo 2, o incremento será de  $n_j$ . A 1a. etapa do procedimento é a chamada da rotina BLOCO, que guarda os valores  $n_i$  (TIPO 1) ou  $n_i$  e  $n_j$  (TIPO 2). A 2a. etapa refere-se a localização da posição em que ficará situado o número de linha  $n_i$  na tabela de índices. Esta operação, efetuada pela rotina BUSCA (Apêndice II.11) procura o endereço da 1a. entrada superior a  $n_i$ , que corresponderá no arquivo Fonte ao seu próximo registro. A 3a. etapa é a solicitação de nó do reservatório para a inclusão da informação de cada cartão lido através da rotina INCLU já mencionada (seção 3.2.2). Aqueles registros cujos números de linha se tornem incompatíveis com a numeração do arquivo não serão inseridos.



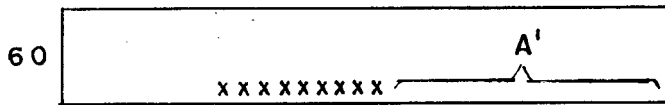
início 1º texto

final 2º texto

FIG. III. 2 - COMANDO ELIMINAR: TEXTOS SITUADOS NO MESMO REGISTRO



início 1º texto



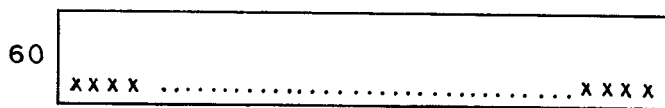
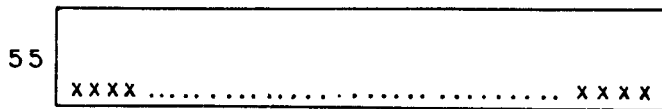
final 2º texto

Se  $A > A'$ ,  $A'$  ocupará o lugar de  $A$  e o reg. 60 será eliminado

FIG. III. 3 - TEXTOS EM REGISTROS CONSECUTIVOS



início 1º texto



final 2º texto

Os reg. 55 e 60 serão eliminados

FIG. III. 4 - TEXTOS EM REGISTROS NÃO CONSECUTIVOS



### 3.3.1.5 - COMANDO LOCALIZAR (/LDxxxxxDxxxxxD...)

A rotina LOCAL (Apêndice II.6) tem por objetivo localizar no arquivo Fonte o(s) texto(s) existente(s) no comando que está(ão) entre delimitadores. Inicialmente a rotina STRIN é executada para separar e guardar estes textos, que são em seguida comparados com cada registro lido do arquivo Fonte. Se algum texto for localizado, o sistema apontará o texto e o número do registro onde o mesmo foi encontrado.

### 3.3.1.6 - COMANDO SUBSTITUIR (/Sn<sub>x</sub>-n<sub>x+k</sub> D<sub>A</sub>xxxxxD<sub>B</sub>xxxxxD)

A rotina SUBST (Apêndice II.7) tem por objetivo substituir o texto A pelo texto B em todos os registros contidos nos blocos de registros. A 1a. etapa do procedimento é a chamada das rotinas BLOCO e STRIN, que guardam os números de linha e textos A e B respectivamente. A 2a. etapa é realizada pela rotina COMD2 (Apêndice II.14), que é responsável pela substituição dos registros e substituição dos textos A por B nos mesmos. Três são os casos considerados pela rotina:

- 1 - Se o texto A for do mesmo tamanho que o texto B (FIG. III.5):  
A substituição dos textos é feita em cada registro guardado por BLOCO, que é localizado no arquivo FONTE através das rotinas ACHAR e BUSCA.
- 2 - Se o texto A for maior do que o texto B (FIG. III.6):  
O procedimento anterior é efetuado, mas como etapa final o registro é rearrumado, deslocando-se todo o texto à direita de A para as posições mais à esquerda, a partir do final do texto B, preenchendo-se com brancos as posições que ficaram vagas.
- 3 - Se o texto A for menor do que o texto B (FIG. III.7):  
A substituição é feita de maneira semelhante aos procedimentos anteriores, mas quando o mesmo registro não é suficiente para conter todo o texto B, ou quando, após a substituição de A por B o restante do registro (guardado antes da substituição) a partir de A não couber todo naquela linha, torna-se '

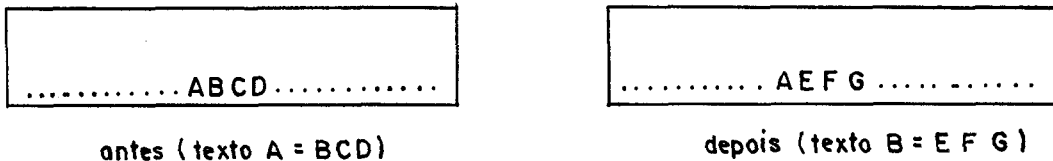


FIG. III. 5 - TAMANHO TEXTO A = TAMANHO TEXTO B

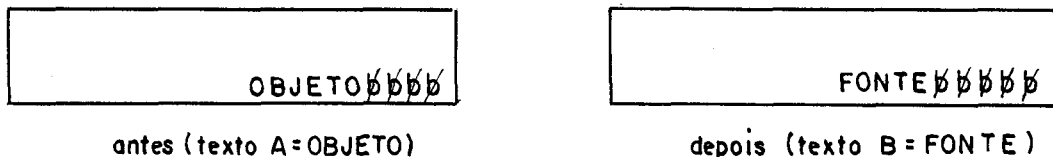


FIG. III. 6 - TAMANHO TEXTO A > TAMANHO TEXTO B

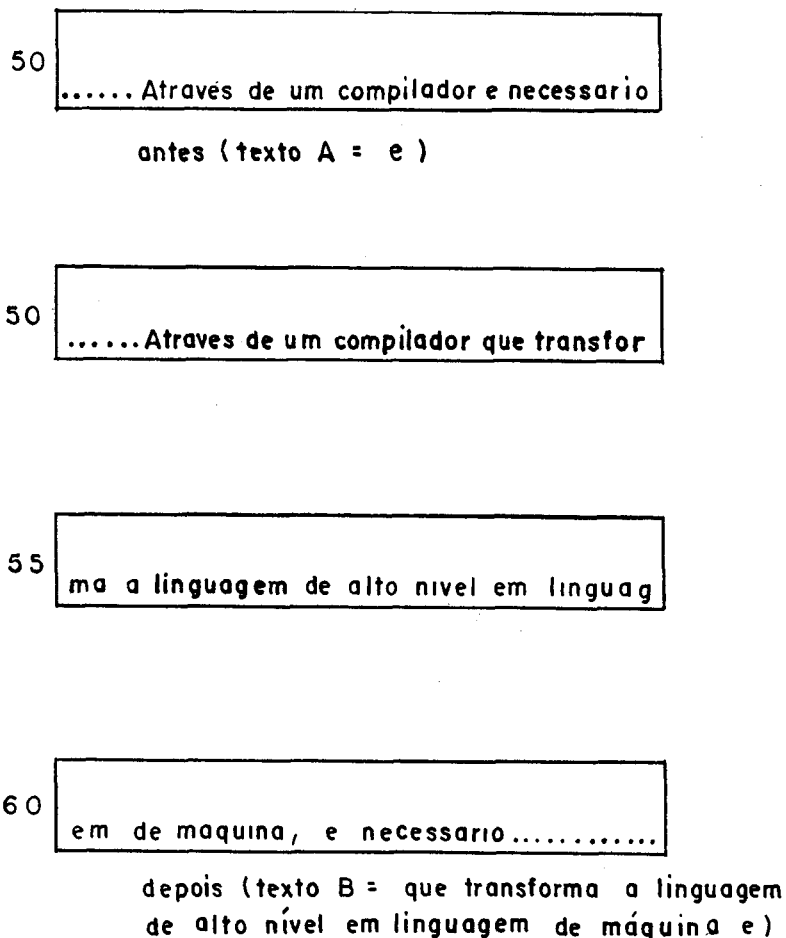


FIG. III. 7 - TAMANHO TEXTO A < TEXTO B: INCLUSÃO DE NOVO REGISTRO

necessário a inclusão de mais um registro no arquivo Fonte, que é feita através da rotina INSER já mencionada anteriormente (seção 3.3.1.4). O número deste novo registro será dado pela média aritmética entre os registros anterior e posterior ao mesmo.

### 3.3.1.7 - COMANDO RENUMERAR ( $/Rn_k^{-n_1, n_i, n_j}$ )

A rotina RENUM (Apêndice II.8) tem por objetivo renumerar os registros de  $n_k$  a  $n_1$ , passando  $n_k$  a receber o valor  $n_i$  e a partir daí os próximos registros são numerados com o acréscimo de  $n_j$  até que o valor  $n_1$  seja atingido.

Inicialmente a rotina BLOCO é executada a fim de guardar os números de linha fornecidos pelo comando. A rotina COMD4 (Apêndice II.16) estabelece a nova numeração para cada registro localizado (rotinas ACHAR e BUSCA), verifica a compatibilidade desta numeração no arquivo - caso esta não seja compatível, aquele registro não é renumerado - e altera a numeração do registro também na tabela de índices.

## 3.4 - EDITOR DE RELATÓRIO

### 3.4.1 - ARQUIVOS UTILIZADOS

#### 3.4.1.1 - ARQUIVO "BACK-UP"

Back-up é o nome do arquivo utilizado para conter a cópia do relatório formatado, e pode ser solicitado através de comandos de controle (seção 2.2.3.4).

Tal arquivo é único para todos os usuários e será mantido por tempo limitado, até que um outro usuário utilize o sistema para emitir um relatório. O objetivo principal da criação deste arquivo é possibilitar a edição de várias cópias sucessivas do relatório final, sem que para isto o sistema tenha que ser processado novamente a cada cópia.

Cada registro é constituído de 120 palavras correspondendo a um número máximo de 2 registros por setor, e o método de acesso ao arquivo é sequencial.

#### 3.4.1.2 - ARQUIVO DE RODAPÉS

O arquivo de rodapés tem por objetivo reunir os rodapés correspondentes a cada página - através dos comandos \$IR e \$FR - e ao final das mesmas imprimi-los na linha especificada pelo sistema ou pelos cartões de controle (seção 2.2.3.4). Este arquivo é único para todos os usuários, e é destruído sempre após a impressão de algum rodapé. Cada registro é constituído de 120 palavras correspondendo a um número máximo de 2 registros por setor. O método de acesso ao arquivo é sequencial.

#### 3.4.1.3 - ARQUIVO DE ÍNDICES

O usuário poderá organizar um índice por página se utilizar o comando \$II incluído no texto que fará parte daquele índice. Todo conteúdo existente entre \$II e \$FI será gravado em uma área em disco denominado arquivo de índices, que só poderá ser impresso no final da formatação do relatório. O arquivo é único para todos os usuários, e será mantido no sistema até que outro usuário utilize o Editor de Relatório. Cada registro do arquivo é constituído de 120 palavras, correspondendo a um número máximo de 2 registros por setor. O método de acesso ao arquivo é sequencial.

#### 3.4.1.4 - ARQUIVO DE REFERÊNCIA CRUZADA

O sistema permite ao usuário a elaboração de um índice de palavras chaves também denominado referência cruzada, que é impresso em ordem alfabética após a impressão do relatório no formato final. Toda a informação existente entre o comando \$IX e \$FX é gravada neste arquivo, juntamente com o número da página em que se encontra. O arquivo é único para todos os usuários, e é destruído sempre que o Editor de Relatório for utilizado. Cada registro é constituído de 120 palavras, correspondendo a um número máximo de 2 registros por setor. O método de acesso ao arquivo é direto.

#### 3.4.2 - DESCRIÇÃO DOS PROCEDIMENTOS UTILIZADOS

A chamada para o EDITOR DE RELATÓRIO é feita através do

cartão \*R, nome arq., analisado pelo programa supervisor EPRIN. Quando a opção R é solicitada, a rotina RELAT (Apêndice III.1) assume o controle do sistema. Ela tem por objetivo criticar e guardar as informações contidas nos cartões de controle que devem seguir imediatamente o cartão \*R, nome arq.. A partir daí tem início a formatação do texto através da análise das informações contidas em cada registro do arquivo Fonte (rotina PESQ - Apêndice III.2).

Ao atingir o final de registro, todo o conteúdo a partir da 1a. posição do registro ou a partir do último comando achado são jogadas para uma área auxiliar de 120 posições - número máximo de caracteres permitidos em uma linha de impressão para impressão posterior - e é feita a análise do próximo registro.

Sempre que um "\$" é reconhecido no texto, as posições anteriores a partir do último comando encontrado também são jogadas para a mesma área auxiliar, e a rotina ROUT (Apêndice III.3) assume o controle do sistema com o objetivo de reconhecer o possível (às vezes o "\$" pode ser apenas um caracter solto no texto) comando de edição correspondente àquele "\$" encontrado. A FIG. III.8) apresenta um esquema contendo todas as facilidades do EDITOR DE RELATÓRIO. Para cada comando de edição um procedimento diferente é executado, às vezes através de rotinas específicas - PULA (Apêndice III.4) no caso do comando \$PLx ser reconhecido; BRANC (Apêndice III.5), se o comando for \$BRx; COLUM (Apêndice III.6), se o comando for \$NCx,y,z; TITLE (Apêndice III.7), se o comando for \$ITx,y,z; NEWP (Apêndice III.8), se o comando for \$NP ( ou '\$NPx); PAGE (Apêndice III.9), se o comando for \$MP - outras vezes na própria rotina ROUT, quando variáveis (ou chaves) recebem valores 0 ou 1 para indicar que o sistema está sob determinado comando. Podemos citar por exemplo o caso dos comandos \$CE (cedilha), \$AP (acentuação), \$IO (início de "overprint"), \$PE (suspende a edição), \$PI (suspende a impressão), \$II (início de índice), \$IX (início de referência cruzada), \$IR (início de rodapé), \$FI (fim de índice), \$FO (fim de "overprint"), \$FR (fim de rodapé), \$FT (fim de título), \$FX (fim de referência cruzada), que se enquadram nesta categoria.

A área auxiliar de 120 posições contem todas as informações do registro já padronizadas no formato final de edição. Esta linha é impressa sempre que o limite máximo da margem direita

for atingido, ou quando algum dos comandos \$PLx, \$NCx,y,z, \$NPx, \$MP, \$ITx,y for reconhecido no registro que está sendo analisado. Por ocasião da impressão da linha, o sistema testa se existe condições para haver separação de sílabas, e em caso afirmativo, ele verifica se o usuário solicitou tal procedimento, pois do contrário esta operação não é realizada.

Ao final de cada página o sistema testa se nela existe algum rodapé. Em caso positivo o mesmo é impresso antes que seja efetuada a mudança de página. Quando isto ocorre a nova página é numerada, e se houve presença de títulos, estes são impressos antes que a 1ª. linha da página seja impressa.

Feita a análise do último registro do arquivo Fonte, o sistema entra na sua fase final de edição (TERMO - Apêndice III.10), através da impressão do índice por página e do índice de palavra chave (em ordem alfabética), caso o usuário os tenha solicitado. A impressão de cópias sucessivas do texto formatado também é feita nesta ocasião.

### 3.5 - IMPRESSÃO DO ARQUIVO FONTE

Este procedimento, solicitado através do cartão \*I, nome arq., permite que a rotina IMPR (Apêndice IV.1) assuma o comando do sistema, com o objetivo de imprimir o conteúdo do arquivo Fonte sequencialmente em ordem crescente por número de linha.

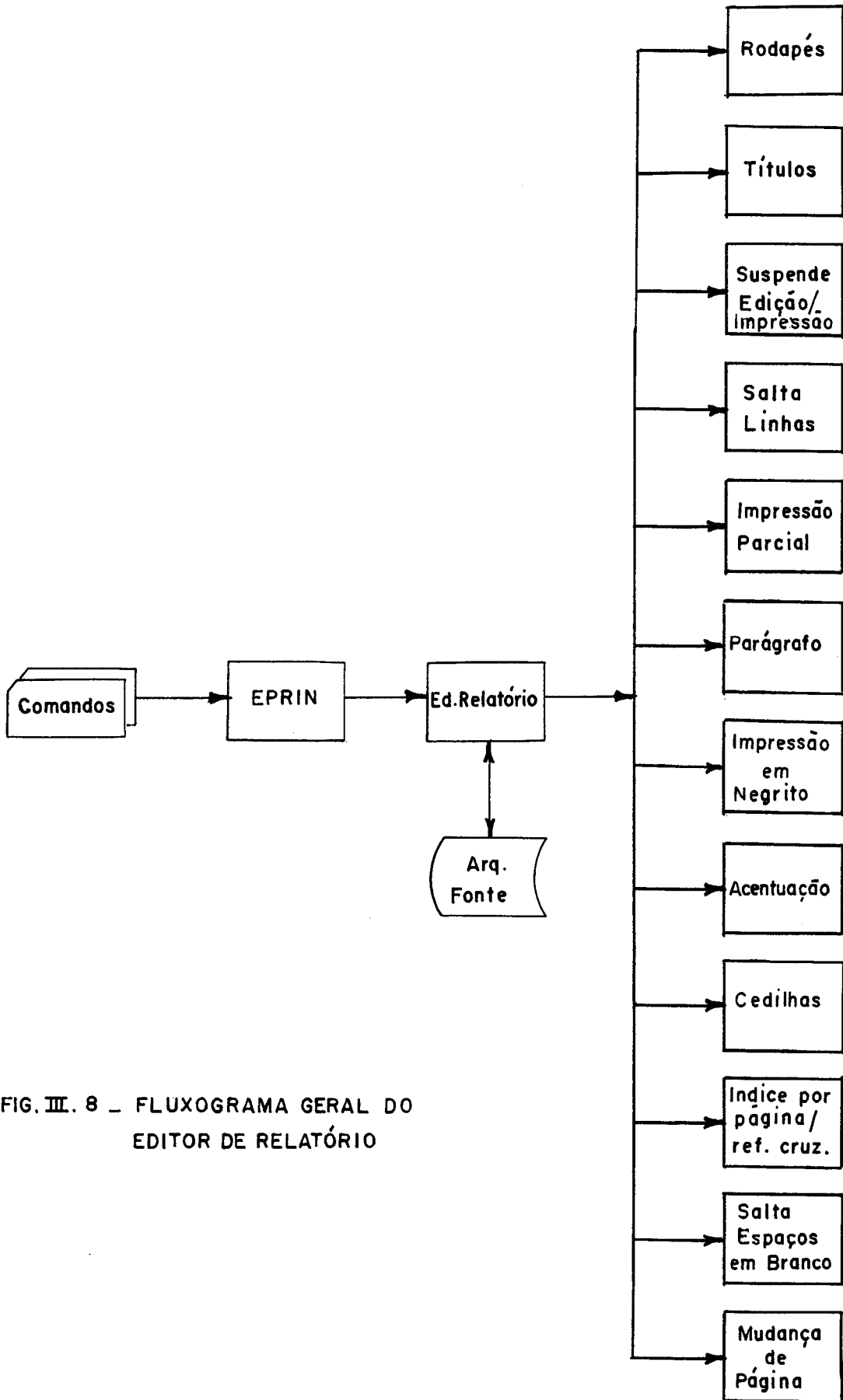


FIG. III. 8 - FLUXOGRAMA GERAL DO EDITOR DE RELATÓRIO

#### IV - IMPLEMENTAÇÃO DO SISTEMA

O EDIPRINT foi desenvolvido no Núcleo de Processamento de Dados - NPD - da UFPb, cujo equipamento disponível está constituído de um computador IBM-1130, com uma configuração de 16k palavras de memória, associado a uma console, unidade de entrada que serve como leitora e perfuradora de cartões modelo 1442, duas impressoras modelos 1403 e 1132 e uma unidade de disco modelo 2311. Das linguagens de programação disponíveis no IBM-1130 - COBOL (destinado a aplicação comercial), FORTRAN (subconjunto do FORTRAN 'IV) e ASSEMBLER - o FORTRAN foi a linguagem escolhida para a implementação do sistema, devido às facilidades que apresenta uma linguagem de alto nível (rapidez de programação, menor interação do usuário com a máquina) dirigida ao uso científico, em relação ao ASSEMBLER, linguagem considerada de baixo nível. Apesar da utilização do ASSEMBLER tornar a execução do sistema mais rápida, a tornaria bastante dependente da máquina, a elaboração e depuração dos programas se tornaria mais lenta, requerendo um volume ainda bem maior do que o atual, aproximadamente 5000 cartões.

A maior dificuldade encontrada durante a fase final de implementação do EDIPRINT foi com relação a capacidade de memória, que não era suficiente para conter todo o sistema. Em consequência disto, o EDIPRINT teve de sofrer várias partições, e somente através de encadeamentos de programas foi possível executar com êxito todas as etapas do sistema. O EDIPRINT está dividido em 6 módulos (partições) - EPRIN, TEXTO, RELAT, IMPR, PESQ, TERMO - interligados ao sistema através da estrutura de LINK da seguinte maneira:

EPRIN: 4800 palavras

Tem por objetivo reconhecer a opção fornecida pelo usuário através do cartão \*n,nome arq. e desviar o comando do sistema para o bloco correspondente àquele procedimento. Chama os módulos: TEXTO, RELAT e IMPR.

TEXTO: 11200 palavras

Tem por objetivo analisar os cartões contendo os comandos de edição de texto e chamar a rotina específica a cada comando.



RELAT: 4100 palavras.

Tem por objetivo analisar e guardar as informações fornecidas através dos cartões de controle. Chama o módulo PESQ.

IMPR: 3500 palavras.

Tem por objetivo imprimir o número de linha e o texto correspondente do arquivo Fonte.

PESQ: 16300 palavras.

Tem por objetivo a formatar o texto através da análise do arquivo Fonte, verificando a cada posição do registro a presença de algum \$ - característica de um comando de formatação de texto. Chama o módulo TERMO.

TERMO: 9900 palavras.

Tem por objetivo finalizar a formatação do texto através da impressão do índice por página e de referência cruzada.

Devido aos poucos recursos que a linguagem FORTRAN apresenta para processamento de cadeias em comparação a linguagens tais como o PL/1 e AIGOL, alguns trechos de algoritmos, que nestas linguagens seriam executadas com no máximo 2 comandos, precisaram com o FORTRAN, de mais comandos para executar o mesmo procedimento. Isto fez com que o sistema ficasse bem extenso. Alguns trechos de programas foram simplificados com a utilização de pacotes de subrotinas denominadas CSP - Commercial Subroutine Package - desenvolvidas pela IBM.

O EDIPRINT foi implementado em aproximadamente 9 meses. Dos 5 discos disponíveis, apenas 3 foram utilizados: o 1º contém o monitor e todas as rotinas do sistema, aproximadamente 60. O 2º é reservado para a gravação de todos os arquivos Fonte dos usuários e o 3º contém os arquivos auxiliares: "back-up", rodapés, referência cruzada, índices e diretório de arquivos. O 4º e 5º são reservados para futuras expansões do sistema.

A chamada para o sistema é feita através do programa supervisor EPRIN, que está gravado em imagem de memória (STORECI). A grande vantagem deste procedimento é a rapidez com que ele é

trazido para a memória no momento em que o usuário entra com o cartão // XEQ EPRIN. A transformação do programa para a imagem ' de memória é feita uma única vez durante a fase de montagem, e as sim guardado no momento da gravação do programa.

A área reservada para a gravação dos arquivos Fonte de todos os usuários, também denominado reservatório de nós, é montada uma única vez em todo o sistema. Nesta etapa, todos os registros (ou nós) são dispostos em uma estrutura de lista linearmente ligada e é feita também a inicialização do diretório de ar quivos.

O sistema prevê a situação em que a tecla INTREQ da con sole possa ser apertada no momento em que uma atualização está sendo processada pelo Editor de Texto. As rotinas LOCK e UNLOCK (Apêndices I.5, I.6) foram criadas com o objetivo de impedir que o programa seja interrompido nesta ocasião. Por outro lado, há pe rigo de um arquivo ser destruído nesta fase, em decorrência da falta de energia. Neste caso, o reservatório deverá ser re inicia lizado e o arquivo Fonte criado novamente. No capítulo V voltare mos a abordar o problema da segurança do arquivo Fonte.

## V - CONCLUSÕES

Até o momento os sistemas desenvolvidos e implementados para o IBM-1130, com a mesma configuração de memória - 16K palavras - são sistemas de Editores de Textos<sup>2,5</sup>. Estes foram também implementados em computadores maiores (IBM/360, /370) e são bastante utilizados através de sistemas ONLINE (TSO - TIME SHARING OPTION).

O EDIPRINT foi o 1º sistema criado para o IBM-1130, dentro de uma configuração de memória reduzida, capaz de criar, alterar e logo após formatar um determinado texto em um padrão pré-determinado, contido em um arquivo qualquer dentro do sistema. Seu funcionamento é OFFLINE, e além de ser de fácil utilização, apresenta grande versatilidade: o usuário poderá em um só passo (JOB) efetuar um ou mais procedimentos (criar, imprimir, alterar e formatar o arquivo) inúmeras vezes.

A grande preocupação foi obter um método de acesso rápido aos arquivos de modo a reduzir o tempo de execução do sistema. Outra característica importante é o baixo custo que este poderá proporcionar em relação a despesas de gráfica e secretaria com a impressão de textos didáticos, ou qualquer outro tipo de documentação.

Apesar do sistema ter sido desenvolvido para o IBM-1130, levou-se em consideração a sua independência em relação à máquina, de maneira a permitir a sua implementação em outros computadores. Isto será possível desde que o computador tenha o compilador FORTRAN e apresente uma configuração mínima compatível com o IBM-1130 utilizado. Algumas ligeiras modificações deverão ser feitas:

- implementação de rotinas que desempenhem as mesmas funções das rotinas MOVE, FILL, NCOMP, PACK e UNPACK (CSP-IBM) caso as mesmas não existam naquele computador.
- as rotinas LOCK e UNLOCK, desenvolvidas no ASSEMBLER do IBM-1130 como medida de segurança para o sistema deverão ser convertidas para o ASSEMBLER do computador utilizado.

Futuras expansões poderão ser feitas no EDIPRINT, e algumas sugestões são dadas a seguir:

- Tornar possível a sua implementação em um outro computador através de um sistema ONLINE. Isto é possível com a criação de mais um comando que permita a edição do texto através de um terminal de vídeo. Seria conveniente também a utilização de uma configuração maior de memória, o que acarretaria em um maior rendimento para o sistema.
- Permitir a impressão de textos em letras minúsculas, tendo em letras maiúsculas apenas nomes próprios ou a 1ª. letra após um símbolo característico de pontuação. Esta nova opção deverá requerer 3 novos comandos de edição do tipo: \$M1 - apenas o caracter seguinte ao comando deverá ser impresso em letra maiúscula; \$IM - início de maiúscula; \$FM - fim de maiúscula. É necessário também que a cadeia de impressão do periférico de saída utilizado permita a impressão de letras minúsculas.

Alguns procedimentos foram esquematizados como soluções viáveis que permitam oferecer maior segurança ao sistema:

- Criação de um arquivo denominado de histórico, inicialmente igual ao Fonte, que permita gravar numa área contígua dentro do próprio arquivo cada alteração efetuada pelo Editor de Texto. Em caso de falha no sistema (falta de energia, por exemplo) durante uma atualização os comandos ali gravados poderão ser reexecutados pelo editor diretamente sobre o texto contido neste arquivo, que eventualmente poderá ser atualizado.
- Criação de um utilitário que permita recuperar um arquivo Fonte em caso de falha no sistema. Como a estrutura do arquivo é circular duplamente ligada, seria razoavelmente simples reconstituí-lo quando ape-

nas um ponteiro fosse destruído. O procedimento utilizado pelo utilitário seria percorrer o arquivo nos 2 sentidos - do último registro para o 1º, e deste para o último - até localizar o ponteiro destruído.

- Entretanto, a mesma falha no sistema pode comprometer mais de 1 registro, e até mesmo um setor inteiro do arquivo. Naturalmente o utilitário anterior não permitirá reconstituir o arquivo da maneira em que os registros estão ligados inicialmente no reservatório, isto é, nós encadeados sequencialmente. Porém, se eles forem encadeados de 8 em 8 (1 setor = 320 palavras ~ 8 registros de 44 pal), de modo que 1 setor aponte sempre para o próximo, será possível reconstituir o setor inteiro através de um utilitário específico.

Uma etapa do EDIPRINT que pode comprometer o sistema está relacionado com o arquivo de referência cruzada: o desempenho do algoritmo de ordenação ficará muito lento se o arquivo for muito grande (mais de 250 referências). Como a solução para o problema não é imediata, e o sistema ainda se encontra em fase experimental, esta etapa poderá ser otimizada futuramente, se necessário, já que índices de referência cruzada não são comumente utilizados em cartas, relatórios e apostilas.

VI - REFERÊNCIAS BIBLIOGRÁFICAS

- 01 - Applied Data Research - ETC Extended Text Compositor - User Guide, Applied Data Research Inc. Nov, 1976.
- 02 - Benjamin, A.J. - An Extensible Editor for a Small Machine with Disk Storage Communications of the ACM 8 (15): 742-744 Aug, 1972
- 03 - Berns, G.M. - The FORMAT Program IEEE Trans on Eng. writing and Speech 2 (11): 85-91, Aug, 1968.
- 04 - Berns, G.M. - Description of the FORMAT, a text-processing program - Communications of the ACM 3 (12):141-146 Mar, 1969.
- 05 - Bourne, S.R. - A Design for a Text Editor Software - Practice and Experience (1): 73-81, 1971.
- 06 - Bratman, H. et all - Program Composition and Editing with an on-line display Fall Joint Computer Conference, 1968 pp 1349-1360.
- 07 - Deutsch, L.P., Lampson, B.W. - An Online Editor - Comm. ACM, 10 (12) Dec, 1967.
- 08 - Elliot, W.D. et all - Computer Assisted Tracin of Text Evolution Fall Joint Computer Conference, 1971 pp 533-540.
- 09 - Hazel, P. - A General-Purpose Text Editor for OS/360 Software - Practice and Experience (4): 389-399, 1974.
- 10 - IBM - System/360 Text Processor - Hyphenating/360 Application Description E20-0257, IBM Corp. Tech. Publications.
- 11 - IBM - Programmer's and Operator's Guide, Version 2 - IBM Corp. Tech. Publications
- 12 - IBM - IBM 1130 Subroutine Library - IBM Corp.Tech.Publications GC 26-5929-6.
- 13 - IBM - 1130 Commercial Subroutine Package - Program Reference Manual (1130) - SE - 25x), Version 3,Modification 1-1968.
- 14 - Knott G.D.,Anderson, W.J.- A Table-Marking Language Proc of the IEEE.
- 15 - Lampson, B.W., Lichtenberger, W.W. - A User Machine in a Time - Sharing System - Proc. IEEE, 54(12): 1766-1775 Dec, 1966.
- 16 - Madnick, S.E. - String Processing Techniques Communications of the ACM 2(10): 420-424 Jul, 1967.

- 17 - Madnick, S.E. e Moulton, A. - Script, An On-line Manuscript Processing System IEEE Trans.on Eng.Writing and Speech 2(11): 92-100 Aug, 1968.
- 18 - NCE - CANDE - Comande e Edite Publicação do NCE-UFRJ.
- 19 - Poole, M.D. - Implementation of an Editing Algorithm Allowing Repeating Corrections.  
Software - Prattice and Experience (1): 373-381, 1971.
- 20 - Rich, R.P., Stone, A.G. - Method for Hyphenating at the end of a Printed Line Communications of the ACM 7(8): 444-445 July, 1965.
- 21 - Van Dam, A. e Rice, D.E. - On-line Text Editing: A Survey Computing Surveys 3(3): 93-114 Sep, 1971.
- 22 - Walter, O.G. - Typesetting - Scientific American, 220(5): 61-69 May, 1969.

## APÊNDICE I

### I.1 - PROGRAMA SUPERVISOR EPRIN

Objetivo: Analisar as opções fornecidas através do cartão \*n, nome arq. e desviar o comando do sistema para o procedimento correspondente.

- 1 - Le cartão \*n, nome arq.
- 2 - Se for // fim de processo.
- 3 - Senão, critica a sintaxe do comando, emitindo mensagens caso ocorram erros de sintaxe.
- 4 - Se n = C
  - 4.1 - Verifica se o nome já consta no diretório de arquivos. Em caso afirmativo, emite mensagem de erro e vai para 1.
  - 4.2 - Chama rotina CRIA (Apêndice I.2) e vai para 1.
- 5 - Verifica se o nome consta no diretório. Em caso negativo o programa emite mensagem de erro e vai para 1.
- 6 - Se n = T, R ou I chama-se respectivamente TEXTO (Apêndice II.1), RELAT (Apêndice III.1), IMPR (Apêndice IV.1) e vai para 1.

### I.2 - ROTINA CRIA

Objetivo: Cria o arquivo Fonte e tabela de índices correspondente no sistema.

- 1 - Le cartão contendo as informações em formato livre e compacta-o para o formato A2.
- 2 - Se for /\* retorna.
- 3 - Le 1º registro disponível do reservatório. Se der overflow, emite mensagem de erro, chama rotina COPIA (Apêndice I.4) que lista os nomes dos arquivos do diretório e retorna. Senão, aponta próximo registro.
- 4 - Se precisar inserir novo registro na tabela de índices:
  - 4.1 - Aloca o registro para a tabela de índices.
  - 4.2 - Executar passo 3.



- 5 - Aloca registro para o arquivo Fonte.
- 6 - Se corresponder ao 1º cartão inserido, compacta e grava nome do arquivo no diretório.
- 7 - Incrementa o número de cartões inseridos (INCR), nº do subcampo preenchido na tabela e estabelece o número de linha' (=INCR\*100).
- 8 - Chama rotina INCLU (Apêndice I.3) para efetuar a inclusão dos registros e vai para 1.

### I.3 - ROTINA INCLU

Objetivo: Incluir registro no arquivo Fonte e tabela de Índices.

- 1 - Chama rotina LOCK (Apêndice I.5) para impedir que o sistema seja cortado.
- 2 - Se estiver preenchendo um subcampo diferente do 1º registro da tabela de Índices vai para 5.
- 3 - Se corresponder ao 1º cartão inserido:
  - 3.1 - Faz o 1º registro da tabela apontar para ele mesmo.
  - 3.2 - Vai para 5.
- 4 - Senão, atualiza ponteiros na tabela de Índices.
- 5 - Grava registro e atualiza ponteiros no arquivo Fonte.
- 6 - Localiza o 1º subcampo vago no registro referente a tabela de Índices e grava o nº da linha e o seu endereço correspondente ( no arquivo Fonte).
- 7 - Chama rotina UNLOCK (Apêndice I.6) e retorna.

### I.4 - ROTINA COPIA

Objetivo: Listar os nomes de todos os arquivos gravados no sistema.

- 1 - IP ← 1
- 2 - IP ← IP+1, le registro IP do diretório.
- 3 - Se for '0000' retorna.
- 4 - Descompacta o nome para formato A1, imprime-o e vai para 2.

### I.5 - ROTINA LOCK

Objetivo: Impedir que o programa seja cortado através da tecla '

INTREQ (da console) no momento em que são feitas atualizações no arquivo Fonte e tabela de índices.

- 1 - Salva endereço da rotina que corta o programa.
- 2 - Modifica o núcleo residente para retornar no caso da tecla INTREQ ser apertada.
- 3 - Retorna.

#### I.6 - ROTINA UNLOCK

Objetivo: Devolver a ILS04 (nível de interrupção) o endereço guardado pela rotina LOCK.

- 1 - Recupera endereço salvo pela rotina LOCK.
- 2 - Retorna.

## APÊNDICE II

### II.1 - ROTINA TEXTO

Objetivo: Analisar os cartões de atualização e transferir o comando do sistema para cada procedimento específico.

- 1 - Le cartão de atualização.
- 2 - Se por /\* retorna.
- 3 - Se a 1a. posição for / e a 2a. A, E, D, I, L, S ou R, chama respectivamente as rotinas APAGA (Apêndice II.2), ELIMI (Apêndice II.3), DELET (Apêndice II.4), INSER (Apêndice II.5), LOCAL (Apêndice II.6), SUBST (Apêndice II.7) e RENUM (Apêndice II.8) respectivamente. Vai para 1.
- 4 - Emite mensagem de erro de sintaxe e vai para 1.

### II.2 - ROTINA APAGA

Objetivo: Apagar textos contidos nos números de linha agrupados nos blocos de registros.

- 1 - Posiciona ponteiro para a 3a. posição do cartão correspondente ao 1º caracter depois do comando de edição (/A).
- 2 - Chama rotina BLOCO (Apêndice II.9) e se tiver havido algum erro de sintaxe retorna.
- 3 - Chama rotina STRIN (Apêndice II.10) e se tiver havido algum erro de sintaxe retorna.
- 4 - Chama rotina COMD1 (Apêndice II.13) e retorna.

### II.3 - ROTINA ELIMI

Objetivo: Remover a partir do 1º texto do comando no registro  $n_k$  até o 2º texto no registro  $n_{x+k}$ .

- 1 - Posiciona ponteiro para a 1a. posição no cartão depois do comando /E.
- 2 - LM ← 1
- 3 - Chama rotina BLOCO e se tiver havido algum erro de sintaxe retorna.
- 4 - Chama rotina STRIN e se tiver havido algum erro de sintaxe

retorna.

5 - LM ← LM+1; se LM < 3 vai para 3.

6 - Chama rotina COMD3 (Apêndice II.15) e retorna.

#### II.4 - ROTINA DELET

Objetivo: Remover registros de um arquivo Fonte.

1 - Posiciona ponteiro para a 1ª. posição do cartão depois do co mando /D.

2 - Chama rotina BLOCO e se tiver havido erro de sintaxe retorna.

3 - Chama rotina COMD1 (Apêndice II.13) e retorna.

#### II.5 - ROTINA INSER

Objetivo: Inserir registros em um arquivo Fonte.

1 - Se o registro a ser inserido é decorrência de uma substituição de textos (i.e, se um texto A foi substituído por B, e B não coube no registro) vai para 4.

2 - Posiciona ponteiro para a 1ª. posição do cartão depois do co mando /I.

3 - Chama rotina BLOCO; se tiver havido erro retorna.

4 - Chama rotina BUSCA (Apêndice II.11) para localizar o registro na tabela onde deverá constar o novo número de linha.

5 - Procura neste registro o subcampo adequado (de acordo com o nº de linha) para a nova entrada. Se este corresponder a um subcampo vago vai para 7.

6 - Testa compatibilidade do novo registro com a numeração do ar quivo, se este for incompatível, emite mensagem de erro e re torna.

7 - Salva conteúdo a partir do subcampo onde a nova linha será ' gravada..

8 - Se o sistema não estiver sob o comando /I vai para 12.

9 - Le cartão contendo o novo registro e compacta-o para o forma to A2.

10 - Se for /\*.

10.1 - Grava o conteúdo salvo (passo 7) após a última entrada na tabela. Se este registro não for suficiente pa-

ra conter toda a informação um novo registro na tabela é inserido com o objetivo de conter o restante que não coube no registro anterior.

10.2 - Retorna.

- 11 - Estabelece o número do novo registro; executa passo 6.
- 12 - Retira nã do reservatório e chama rotina INCLU para inclusão do novo registro.
- 13 - Se estiver sob ação do comando /I vai para 9, senão vai para 10.1.

## II.6 - ROTINA LOCAL

Objetivo: Localizar o número dos registros onde estão gravados os textos entre delimitadores.

- 1 - Posiciona ponteiro para a la. posição do cartão depois do comando /L.
- 2 - Chama rotina STRIN, e se tiver havido erro retorna.
- 3 - Aponta o 1º registro do arquivo Fonte.
- 4 - Le registro e descompacta-o para o formato A1.
- 5 - Para cada texto a ser localizado chama rotina TESTA (Apêndice II.19).
- 6 - Aponta próximo registro. Enquanto não houver percorrido todo arquivo vai para 4.
- 7 - Indica os textos não localizados e retorna.

## II.7 - ROTINA SUBST

Objetivo: Substituir um determinado texto por outro nos números' de linha agrupados pelos blocos de registros.

- 1 - Posiciona ponteiro para a la. posição do cartão depois do comando /S.
- 2 - Chama rotina BLOCO, e se tiver havido erro retorna.
- 3 - Chama rotina STRIN, e se tiver havido erro retorna.
- 4 - Chama rotina COMD1 e retorna.

## II.8 - ROTINA RENUM

Objetivo: Renumerar os registros de um arquivo.

- 1 - Posiciona ponteiro para a 1a. posição do cartão depois do comando /R.
- 2 - Chama rotina BLOCO, e se tiver havido erro retorna.
- 3 - Chama rotina COMD4 (Apêndice II.16) e retorna.

## II.9 - ROTINA BLOCO

Objetivo: Identificar e guardar os números de linha agrupados em cada bloco de registro.

- 1 - Testa se a posição pré-determinada do cartão que contem o comando de edição é um caracter numérico, se não for o programa emite erro de sintaxe e retorna.
- 2 - Chama rotina AGRUP (Apêndice II.20) que reconhece o nº do registro, e se tiver havido erro retorna.
- 3 - Testa o 1º caracter depois do número do último registro encontrado. Este número é guardado em uma tabela de acordo com o seguinte esquema:
  - Se o caracter for '-' um zero é colocado na 1a. coluna da tabela e o número encontrado antes do '-' e depois são guardados nas colunas 2 e 3 respectivamente. Vai para 1.  
Exemplo: /A100-500            0 100 500
  - Se o caracter for ',' e o último caracter também, o n-1 é colocado e o número anterior a ',' é guardado na 2a. coluna. Porém, se o caracter for ',' e o último for '-', o nº anterior ao '-' é guardado na próxima linha da tabela.  
Exemplo: 1. /A100,500            1 100            1 100  
          2. /A100,500-1000    1 500            0 500 1000  
                                  Ex.1            Ex. 2
  - Se for '\$' o nº 2 é colocado na 1a. coluna de TAB.  
Exemplo: /I50\$10                2 50 10
  - Se for 'Ø', verifica se na 1a. coluna existe um 2. Se existir coloca o número do registro encontrado na 3a. coluna e retorna. Se não, coloca-o na coluna 2 e retorna.
- 4 - Testa se aquela posição corresponde a algum delimitador. Se não for emite mensagem de erro e retorna.
- 5 - Coloca o último número encontrado na tabela de acordo com o passo 3 e retorna.

## II.10 - ROTINA STRIN

Objetivo: Guardar cada texto entre delimitadores.

- 1 - Incrementa de 1 posição o ponteiro que percorrer o cartão contendo o comando de atualização. Se posição > 80 e o 2º delimitador não tiver sido encontrado, emite mensagem de erro e retorna.
- 2 - Se a posição corresponder a um branco e não houver mais texto no cartão retorna.
- 3 - Se a posição corresponder a um delimitador, e se este for igual ao 1º já encontrado, guarda o texto em uma área auxiliar.
- 4 - Vai para 1.

## II.11 - ROTINA BUSCA

Objetivo: Localizar o registro na tabela de índices onde um determinado número de linha deverá ser encontrado, ou no caso de inserção, onde o mesmo deverá ser posicionado.

- 1 - Aponta registro inicial da tabela.
- 2 - Le registro.
- 3 - Enquanto o nº de linha do 1º subcampo for menor do que o valor pesquisado, aponta o próximo registro. Se este já corresponder ao registro inicial, retorna este valor ao programa principal. Senão, vai para 2.
- 4 - Aponta registro anterior e retorna.

## II.12 - ROTINA ACHAR

Objetivo: Determinar o endereço do registro desejado no arquivo Fonte.

- 1 - Chama rotina BUSCA (rotina anterior).
- 2 - Percorre cada subcampo do registro na tabela (determinado por BUSCA) até encontrar o número de linha desejado. Aponta o endereço que está gravado na 1ª palavra daquele subcampo.
- 3 - Se estiver sob ação dos comandos DELETAR ou RENUMERAR retorna.

- 4 - Le o registro no arquivo Fonte, descompacta o texto lá gravado para o formato A1 e retorna.

#### II.13 - ROTINA COMD1

Objetivo: Fazer com que o comando seja processado em cada bloco de registro contido no cartão de dados.

- 1 - Aponta 1a. linha da tabela guardado pela rotina BLOCO.
- 2 - Chama rotina CRITI (Apêndice II.21) para criticar a compatibilidade dos registros na tabela.
- 3 - Aponta registro a ser localizado e chama rotina ACHAR. Se não achar o registro, envia mensagem de erro e retorna.
- 4 - Se estiver sob ação do comando /D (DELETAR), chama rotina APREG e retorna.
- 5 - Para cada texto guardado pela rotina STRIN executa:
  - 5.1 - Chama rotina TESTA (Apêndice II.19) e se houver erro ' retorna.
  - 5.2 - Se estiver sob ação do comando /A (APAGAR) chama rotina CLEAN (Apêndice II.17), compacta e grava registro . Vai para 5.
  - 5.3 - Se estiver sob ação do comando /S (SUBSTITUIR) chama ' rotina COMD2 (Apêndice II.14) .
- 6 - Se a 1a. posição da linha correspondente na tabela for 1, aponta a próxima linha da tabela, e se ainda houver registros vai para 2. Senão, retorna.
- 7 - Se a 1a. posição da linha correspondente na tabela for 0, aponta o próximo registro no arquivo Fonte e enquanto o nº de linha não corresponder ao número guardado na 3a. posição da tabela:
  - 7.1 - Le registro e descompacta-o para o formato A1.
  - 7.2 - Executa passo 5.
- 8 - Aponta próxima linha da tabela e vai para 2.

#### II.14 - ROTINA COMD2

Objetivo: Efetuar a substituição do texto nos registros correspondentes fornecidos pelos blocos de registros.



- 1 - Compara o tamanho do 1º e 2º textos contidos no cartão de da dos.
- 2 - Se os 2 textos forem do mesmo tamanho:
  - 2.1 - Substituir um texto pelo outro na posição adequada do registro fornecida pela rotina TESTA.
  - 2.2 - Compacta registro, grava-o de volta no arquivo Fonte e retorna.
- 3 - Se o 1º texto for maior do que o 2º:
  - 3.1 - Substituir o 1º texto pelo 2º, e rearruma o registro ' deslocando as posições à direita do texto antigo (1º) para à esquerda do atual e preenchendo com brancos as posições que ficaram vagas.
  - 3.2 - Executa passo 2.2.
- 4 - Se o 1º texto for menor do que o 2º:
  - 4.1 - Substituir o 1º texto pelo 2º, mas se o 2º texto não ' couber todo naquele registro:
    - 4.1.1 - Compacta-o e grava-o de volta no arquivo, esta belece o nº de linha para a inclusão de um novo registro, chama a rotina INSER e retorna.
  - 4.2 - Substituí o 1º texto pelo 2º, mas se durante a rearrumação do registro o conteúdo do registro a partir do texto antigo não couber todo naquele registro executa passo 4.1.1.

## II-15 - ROTINA COMD3

Objetivo: Eliminar o texto contido no cartão de dados nos registros especificados pelo comando /E.

- 1 - Testa se o 1º registro contido no cartão é menor do que 2º . Se for envia mensagem de erro e retorna. Caso contrário, aponta registro  $n_k$ .
- 2 - Chama rotina ACHAR (Apêndice II.12) para localizar o registro e caso o mesmo não seja encontrado envia mensagem de erro e retorna.
- 3 - Chama rotina TESTA para localizar o 1º texto no registro  $n_k$ . Se houver erro retorna.
- 4 - Se o registro  $n_k = n_{x+k}$ , chama rotina TESTA para localizar o 2º texto no registro.

- 4.1 - Coloca brancos nas posições entre o 1º e 2º texto, rearruma registro, desviando as posições à direita do 2º texto para as posições a partir do 1º texto e coloca brancos nas posições que ficarem vagas.
- 4.2 - Compacta registro e grava-o no arquivo; retorna.
- 5 - Aponta registro  $n_{x+k}$ , executa passo 2 e chama rotina TESTA para localizar o 2º texto no registro  $n_{x+k}$ .
- 6 - Coloca brancos a partir do 1º texto até o final do registro  $n_k$  e também a partir da 1ª. posição do registro  $n_{x+k}$  até o final do 2º texto. Desloca as posições à direita do 2º texto para o registro  $n_k$ , a partir da 1ª. posição vaga à direita correspondente ao 1º texto.
- 7 - Se o conteúdo do registro  $n_{x+k}$  couber todo no registro  $n_k$ , compacta-o e grava-o no arquivo. Chama rotina APREG (Apêndice II.18) para remover todos os registros  $n$  entre:  $n_k < n \leq n_{x+k}$ ; retorna.
- 8 - Se o conteúdo do registro  $n_{x+k}$  não couber todo no registro  $n_k$ :
  - 8.1 - Compacta-o e grava-o no arquivo.
  - 8.2 - Coloca o restante no registro seguinte, executa passo 8.1 e chama a rotina APREG para remover os registros  $n$  entre:  $n_{k+1} < n \leq n_{x+k}$ ; retorna.

## II.16 - ROTINA COMD4

Objetivo: Efetuar a renumeração dos registros conforme as especificações contidas no comando.

- 1 - Chama rotina CRITI (Apêndice II.21) para verificar a compatibilidade dos números de linha fornecidos no cartão. Se houver erro envia mensagem e retorna.
- 2 - Chama rotina ACHAR (Apêndice II.12) para localizar o registro inicial a ser renumerado. Se não encontrar retorna.
- 3 - Estabelece a nova numeração e verifica se está é compatível com a numeração do arquivo. Se for o registro não é renumerado e retorna.
- 4 - Grava registro com a nova numeração e atualiza a sua antiga numeração na tabela de índices.
- 5 - Aponta o próximo registro. Se o nº de linha deste for maior do que

o último registro do comando a ser renumerado retorna. Caso contrário vai para 3.

## II.17 - ROTINA CLEAN

Objetivo: Apagar um determinado texto no arquivo.

- 1 - Desloca as posições à direita do texto eliminado para à esquerda a partir da sua posição inicial.
- 2 - Preenche as posições vagas com brancos e retorna.

## II.18 - ROTINA APREG

Objetivo: Remover registros do arquivo Fonte.

- 1 - Se a rotina estiver atuando sobre uma estrutura de bloco do tipo  $n_x, n_{x+k}$  vai para 4.
- 2 - Chama rotina ACHAR (Apêndice II.12) para localizar o registro  $n_{x+k}$  (estrutura  $n_x - n_{x+k}$ ).
- 3 - Se um arquivo inteiro estiver sendo removido, retira o seu nome do diretório.
- 4 - Remove registro(s) do arquivo Fonte devolvendo o(s) nó(s) correspondente(s) ao reservatório, atualiza ponteiros no arquivo e retira da tabela de índices o(s) número(s) do(s) registro(s) removido(s) no arquivo Fonte, isto é, coloca brancos nos subcampos correspondentes. Rearruma cada registro da tabela, e se o número de registros removidos do Fonte corresponder a 1 registro inteiro da tabela, devolve aquele nó ao reservatório, retira-o da tabela de índices e atualiza seus ponteiros, retorna.

## II.19 - ROTINA TESTA

Objetivo: Procurar um determinado texto em um registro do arquivo.

- 1 - Aponta 1ª. posição do texto (=Texto 1) no registro e do texto a ser localizado (texto 2).
- 2 - Se o comando em ação do /L (localizar):

- 2.1 - Se a posição do texto no registro corresponder a um 'branco':
  - 2.1.1 - Se o mesmo ocorrer o Texto 2 vai para 4.
  - 2.1.2 - Aponta próxima posição do texto 1 e vai para '2.1.
- 3 - Compara texto 1 com texto 2:
  - 3.1 - Se for igual vai para 4.
  - 3.2 - Senão, aponta próxima posição do texto 1. Se for maior do que 80:
    - 3.1.1 - Se estiver sob a ação do comando /L retorna.
    - 3.2.2 - Envia mensagem de que o texto não foi encontrado e retorna.
- 4 - Aponta próxima posição do texto 1. Se for menor ou igual a 80:
  - 4.1 - Aponta a próxima posição do texto 2. Se todo o texto 2 já tiver sido localizado retorna, fornecendo caso este já sob a ação do comando /L, o nº do registro em que este se encontra.
  - 4.2 - Vai para 2.
- 5 - Se não estiver sob ação do comando /L executa 3.2.2. Caso contrário, verifica se o restante do texto continua no registro seguinte.

## II.20 - ROTINA AGRUP

Objetivo: Identificar cada caracter numérico contido nos blocos de registro até encontrar um delimitador ou um caracter especial ('', '-', '\$', '¢').

- 1 - Se a posição correspondente no cartão for um caracter especial o programa acusa erro e retorna.
- 2 - Se for um caracter numérico, guarda-o em um vetor auxiliar, aponta a próxima posição no cartão e testa de novo.
- 3 - Forma o número a partir dos caracteres armazenados no vetor, e retorna.

## II.21 - ROTINA CRITI

Objetivo: Criticar os registros armazenados pela rotina BLOCO.

- 1 - Se a estrutura do bloco de registro analisado for do tipo  $n_k, n_{x+k}$ , retorna.
- 2 - Senão, testa se  $n_x < n_{x+k}$ . Se isto não ocorrer acusa erro.
- 3 - Retorna.

## APÊNDICE III

### III.1 - ROTINA RELAT

Objetivo: Criticar e guardar as informações contidas nos cartões de controle.

- 1 - Le cartão de controle.
- 2 - Se for /\*:
  - 2.1 - Testa se as informações contidas nos cartões de controle de 10 a 14 são exclusivamente 0 ou 1. Se não forem, envia mensagem de erro e retorna.
  - 2.2 - Se o nº de linhas por página for menor do que o número da linha determinado para início de rodapé, envia mensagem de erro e retorna.
  - 2.3 - Chama rotina PESQ (Apêndice III.2) para formatação do texto e retorna.
- 3 - Identifica número de controle. Se for diferente das 14 entradas permitidas (sabendo-se que os cartões de controle de 1 a 9 começam na 2a. coluna e de 10 a 14 a partir da 1a.) envia mensagem de erro e retorna.
- 4 - Se o caracter seguinte ao nº de controle for diferente de '-' acusa erro e vai para 1.
- 5 - Aponta próxima posição do cartão. Enquanto esta posição corresponder a um caracter numérico executa 5.
- 6 - Se for diferente de branco ou vírgula, acusa erro de sintaxe e retorna. Chama rotina CONCA (Apêndice III.14) para formar o número contido (informação) naquele cartão de controle
- 7 - Se for branco e o nº de controle permitir apenas 1 informação:
  - 7.1 - Guarda informação e testa se é adequada ao nº de controle correspondente. Caso não seja envia mensagem de erro e vai para 1.
  - 7.2 - Vai para 1.
- 8 - Se for ',' chama rotina CONCA. Testa se o nº de controle permite mais de uma informação. Caso contrário, acusa erro e vai para 1.  
Executa 7.1 e vai para 5.
- 9 - Se for branco e permitir mais de 2 informações, testa se o nº de controle é 3. Senão for acusa erro e vai para 1.

Guarda informação e vai para 1.

### III.2 - ROTINA PESQ

Objetivo: Reconhecer o caracter '\$' no registro referente ao comando de formatação do texto e comandar a sua execução.

- 1 - Atribui valores padrões aos comandos de controle caso estes não tenham sido fornecidos através de cartões. Chama rotina INIC1 (Apêndice III.24) para inicializar a página. Aponta o 1º registro do arquivo.
- 2 - Le registro e descompacta-o para formato A1. Aponta la. posição do registro.
- 3 - Marca esta posição.
- 4 - Se for igual a \$:
  - 4.1 - Se estiver sob ação do comando \$PI (para impressão) ou se o '\$' corresponder a la. posição do registro vai para 8.1.
  - 4.2 - Senão, vai para 6.
- 5 - Aponta próxima posição. Se for menor que 81 vai para 4. Caso contrário:
  - 5.1 - Se estiver com a impressão suprimida:
    - 5.1.1 - Chama rotina LEREG (Apêndice III.13) para ler o próximo registro e se já houver percorrido todo o arquivo chama TERMO (Apêndice III.10) e retorna.
    - 5.1.2 - Vai para 3.
- 6 - Se o texto com a edição suprimida, redefine margem esquerda.
- 7 - Verifica se as posições a partir da posição marcada até a posição atual cabem na área auxiliar de saída que contém o texto formatado. Se couber, transfere as posições para esta área a partir da margem esquerda determinada. Se houver impressão em negrito, estes caracteres também são transferidos para uma área determinada. Acerta margem esquerda.
- 8 - Se a posição for maior que 81 executa 5.1.1. Se a posição corresponder a um \$:
  - 8.1 - Chama rotina ROUT (Apêndice III.3).

Se já estiver percorrido todo o arquivo chama TERMO (Apêndice III.10) e retorna.

- 9 - Aponta próxima posição e executa 5.1.1. Se esta posição for \$ vai para 8.1.
- 10 - Vai para 3.

### III.3 - ROTINA ROUT

Objetivo: Reconhecer cada comando de edição e chamar o procedimento adequado para execução do mesmo.

- 1 - Aponta a próxima posição no registro e chama rotina LEREG (Apêndice III.13) para ver se atingiu final de registro.
- 2 - Se o comando começar por I, M, N, P, F e R, executa passo 1 e a seguir os passos 2.1, 2.2, 2.3, 2.4, 2.5 e 2.6 respectivamente.
  - 2.1 - Se o 2º caracter do comando for X, I, R, O, T executa passos 2.1.1, 2.1.2, 2.1.3, 2.1.4, 2.1.5 respectivamente. Caso contrário, chama rotina COMPA (Apêndice III.21). Se daquela posição até o final do registro só existir caracter branco vai para 2.1, senão retorna (nesse caso o \$ achado não faz parte de um comando).
    - 2.1.1 - Marca a posição que o próximo caracter vai ter na área auxiliar de saída e retorna.
    - 2.1.2 - Se o conteúdo da área auxiliar ainda não foi impresso, chama rotina IMPL (Apêndice III.23).
      - 2.1.2.1 - Retorna.
    - 2.1.3 - Executa procedimento 2.1.2, redefine margem esquerda para a posição correspondente a impressão do rodapé e retorna.
    - 2.1.4 - Liga uma chave para indicar que existe impressão em negrito e retorna.
    - 2.1.5 - Inicializa arquivo de títulos, chama rotina TITLE (Apêndice III.7) e retorna.
  - 2.2 - Se o 2º caracter do comando for P: executa 2.1.2; chama rotina PAGE e retorna. Senão, chama rotina COMPA (Apêndice III.21). Se daquela posição até o final do registro só tiver caracter branco vai para 2.2. Caso contrário, retorna.
  - 2.3 - Se o 2º caracter do comando começar por C ou P, chama respectivamente rotinas COLUM (Apêndice III.6) ou NEWP



(Apêndice III.8) e retorna.

2.4 - Se o 2º caracter começar por L, I ou E executa 2.4.1 , 2.4.2 e 2.4.3 respectivamente. Se não for nenhum dos 3 símbolos chama rotina COMPA. Se daquela posição até o final do registro só existir caracter branco testa 2.4 de novo. Caso contrário, retorna.

2.4.1 - Chama rotina PULA (Apêndice III.4) e retorna.

2.4.2 - Indica que a impressão do texto será suspensa e retorna.

2.4.3 - Executa 2.1.2; aponta ponteiro do registro para a última posição chama rotina LEREG, redefine margem esquerda e retorna.

2.5 - Se o 2º caracter começar por O, T, I, X, R executa 2.5.1, 2.5.2, 2.5.3, 2.5.4 e 2.5.5 respectivamente. Se não for, chama rotina COMPA. Se daquela posição até o final do registro só existir caracter branco testa 2.5 novamente. Caso contrário, retorna.

2.5.1 - Indica fim de overprint e retorna.

2.5.2 - Chama rotina IMPL (Apêndice III.23), marca fim de título e retorna.

2.5.3 - Chama rotina IMPL, marca fim de índice e retorna.

2.5.4 - Guarda informação relativa ao índice de referência cruzada, chama rotina GRAVA (Apêndice III.11) e retorna.

2.5.5 - Chama rotina IMPL, marca fim de rodapé e retorna.

2.6 - Se o 2º caracter for diferente de I ou E :

2.6.1 - Chama rotina COMPA. Se daquela posição até o final do registro só tiver caracter branco volta a testar 2.6. Senão, retorna.

2.6.2 - Caso contrário, reinicia impressão e edição respectivamente e retorna.

3 - Se o comando não começar por A, B ou C, chama COMPA. Se daquela posição até o final do registro só existir caracter branco executa passo 3 novamente. Senão, retorna.

Caso contrário, executa passos 3.1, 3.2 e 3.3 respectivamente.

3.1 - Se o 2º caracter começar por P coloca em uma área auxi

- liar o caracter 'i' relativo a acentuação e retorna.
- 3.2 - Se o 2º caracter começar por R chama rotina BRANC ( Apêndice III.5) e retorna.
- 3.3 - Se o 2º caracter começar por E coloca em uma área auxiliar o caracter ', ' relativo a cedilha e retorna.

#### III.4 - ROTINA PULA

Objetivo: Saltar o número de linha(s) estabelecido pelo comando \$PLn.

- 1 - Imprime o conteúdo da área auxiliar - rotina IMPL (Apêndice III.23) - caso esta ainda não tenha sido impressa.
- 2 - Aponta próxima posição no registro e chama rotina LEREG ( Apêndice III.13). Se já houver percorrido todo arquivo retorna.
- 3 - Testa se aquela posição corresponde a um caracter numérico . Em caso afirmativo guarda-o e vai para 2. Caso contrário:
  - 3.1 - Chama rotina COMPA (Apêndice III.21). Se daquela posição até o final do registro só existir brancos:
    - 3.1.1 - Se ainda não houver reconhecido nenhum caracter numérico vai para 2.
    - 3.1.2 - Transfere os caracteres numéricos posteriores' ao comando no registro anterior para uma área' auxiliar e vai para 2, continuando a pesquisa.
- 4 - Se a etapa 3.1.2 foi executada, agrupa todos os caracteres ' numéricos encontrados e vai para 5. Caso contrário, se o 1º caracter diferente de branco, posterior ao comando não for numérico, é porque não foi encontrado o nº de linhas a ser saltada e retorna.
- 5 - Chama rotina CONCA (Apêndice III.14) para agrupar os caracteres encontrados e formar o número na base 10.
- 6 - Se o número de linhas a serem saltadas na página ultrapassarem o limite máximo de linha por página:
  - 6.1 - Se existir rodapé naquela página chama rotina ESCR (Apêndice III.16) para imprimí-lo.
- 7 - Salta número de linhas. Para cada linha saltada executa (7.1 e 7.2):
  - 7.1 - Se ultrapassar o nº máximo de linhas por página chama

rotina PAGE (Apêndice III.9) e vai para 8.

7.2 - Chama rotina PROC1 (Apêndice III.18).

8 - Aponta posição anterior no registro e retorna.

### III.5 - ROTINA BRANC

Objetivo: Deixar n espaços em branco entre um texto e outro em u ma linha conforme a especificação dada pelo comando ' \$BRn.

- 1 - Aponta próxima posição no registro e chama rotina LEREG ( Apêndice III.13). Se já houver percorrido todo o arquivo retorna.
- 2 - Testa se aquela posição corresponde a um caracter numérico . Em caso afirmativo guarda-o e vai para 1. Senão chama rotina COMPA (Apêndice III.21). Se daquela posição até o final do registro sô existir brancos:
  - 2.1 - Se ainda não houver reconhecido nenhum número vai para 1.
  - 2.2 - Transfere os caracteres numéricos seguintes ao comando contidos no registro anterior para uma área auxiliar e vai para 1.
- 3 - Se a etapa 2.2 foi executada, agrupa todos os caracteres numéricos encontrados e vai para 4. Caso contrário, se o 1º ' caracter diferente de branco, posterior ao comando não for ' numérico retorna.
- 4 - Chama rotina CONCA (Apêndice III.14).
- 5 - Chama rotina CABE (Apêndice III.12) para testar se o número de espaços a ser deixados em branco cabem naquela linha, se não couberem vai para 7.
- 6 - Caso contrário, acerta margem esquerda a partir de onde o pró ximo caracter será movido na área auxiliar de saída.
- 7 - Aponta posição anterior no registro e retorna.

### III.6 - ROTINA COLUM

Objetivo: Alterar as margens direita e esquerda estabelecido pelo comando \$NCx,y,z.

- 1 - Imprime o conteúdo da área auxiliar - rotina IMPL (Apêndice III.23) - caso esta ainda não tenha sido impressa.
- 2 - Aponta próxima posição no registro e chama rotina LEREG ( Apêndice III.13). Se já houver percorrido todo o arquivo retorna.
- 3 - Testa se aquela posição corresponde a um carácter numérico . Em caso positivo vai para 2. Caso contrário:
  - 3.1 - Chama rotina COMPA (Apêndice III.21). Se daquela posição até o final do registro só existir brancos:
    - 3.1.1 - Se a 1a. posição do registro seguinte for vírgula:
      - 3.1.1.1 - Reune os caracteres numéricos posteriores ao comando contidos no registro anterior e vai para 2.
      - 3.1.2 - Se ainda não houver reconhecido nenhum número, vai para 2. Senão executa 3.1.1.1.
- 4 - Se a etapa 3.1.1.1 foi executada agrupa todos os caracteres numéricos contidos no comando e vai para 6.
- 5 - Se o 1º carácter diferente de branco, seguinte ao comando ou a uma vírgula contida na sintaxe do mesmo não for numérico ' retorna.
- 6 - Chama rotina CONCA (Apêndice III.14). Se o número formado cor responder a x ou y, estabelece o número de linhas (=x) e a margem esquerda (=y). Se a posição corresponder a uma vírgula vai para 2.Caso contrário retorna.
- 7 - Se corresponder ao z, estabelece a margem direita, aponta po sição anterior no registro e retorna.

### III.7 - ROTINA TITLE

Objetivo: Imprimir o título no t<sup>o</sup>po de um p<sup>a</sup>gina a partir das es pecificações fornecidas a partir do comando \$ITx,y.

Passos 1, 2, 3, 4 e 5 idênticos aos da rotina anterior.

- 6 - Chama rotina CONCA. Se o número formado corresponder ao x, es tabelece o número de linhas a partir do qual o título será ' impresso. Se a posição no registro corresponder a vírgula, vai para 1. Senão, retorna.
- 7 - Se o número formado corresponder a y, estabelece a margem es

querda para impressão do título, aponta posição anterior no registro e retorna.

### III.8 - ROTINA NEWP

Objetivo: Estabelecer um novo parágrafo para a próxima linha a ser impressa (\$NPn).

- 1 - Imprime o conteúdo da área auxiliar (rotina IMPL) caso esta ainda não tenha sido impressa.
- 2 - Aponta próxima posição no registro. Se já houver percorrido todo o arquivo retorna.
- 3 - Testa se aquela posição corresponde a um caracter numérico . Em caso positivo guarda-o e vai para 2. Caso contrário:
  - 3.1 - Chama rotina COMPA (Apêndice III.21), Se daquela posição até o final do registro só existir brancos:
    - 3.1.1 - Se ainda não tiver formado nenhum número vai para 2.
    - 3.1.2 - Senão, transfere para uma área auxiliar as posições numéricas contidas no registro anterior que seguem o comando \$NP e vai para 2.
- 4 - Se o passo 3.1.1 tiver sido executado agrupa todos os caracteres encontrados após o comando e vai para 6.
- 5 - Se nenhum número tiver sido encontrado após o comando, o parágrafo continua sendo o mesmo já definido anteriormente (através dos cartões de controle ou pelo próprio sistema) e vai para 7.
- 6 - Chama rotina CONCA. Se estiver sob ação do comando \$NCx,y,z, a margem esquerda continua sendo a mesma por ele estipulada. Caso contrário ela é alterada para n.
- 7 - Aponta posição anterior do registro e retorna.

### III.9 - ROTINA PAGE

Objetivo: Efetuar mudança de página.

- 1 - Se existir rodapé naquela página, posiciona linha para a posição que marca o início de rodapé. Para cada linha que for saltada, chama rotina PROC1 (Apêndice III.18) para testar se

o usuário solicitou arquivo back up. Chama rotina ESCR (Apêndice III.16) para imprimir o rodapé.

- 2 - Chama rotina INIC1 (Apêndice III.24) para inicializar nova página. Se houver impressão do título, posiciona linha para impressão do mesmo. Para cada linha saltada, chama PROC1. Chama ESCR para imprimir o título.
- 3 - Estabelece linha topo (1a. linha da página) e para cada linha saltada chama rotina PROC1.

### III.10 - ROTINA TERMO

Objetivo: Efetuar a impressão do índice por página e de referência cruzada.

- 1 - Imprime o conteúdo da área auxiliar (rotina IMPL) caso esta ainda não tenha sido impressa.
- 2 - Caso exista rodapé na última página impressa, chama rotina ESCR para efetuar a impressão.
- 3 - Chama rotina FINAL (Apêndice III.25) para imprimir os índices
- 4 - Se o usuário houver solicitado mais de 1 cópia do arquivo "back up", chama as rotinas ESCR e FINAL respectivamente para cada cópia editada.
- 5 - Retorna.

### III.11 - ROTINA GRAVA

Objetivo: Gravar as informações relativas aos arquivos de índice, rodapês, referência cruzada e títulos.

- 1 - Procura no arquivo solicitado o 1º registro disponível para gravação.
- 2 - Chama rotina PROC (Apêndice III.17) para gravação de acentos, cedilhas ou overprint.
- 3 - Coloca flag após o último registro gravado no arquivo.

### III.12 - ROTINA CABE

Objetivo: Verificar se é possível transferir todas as posições para a área auxiliar de saída.

- 1 - Se a posição atual acrescida do número de posições a serem transferidas não ultrapassar a margem direita estabelecida, retorna.
- 2 - Caso contrário, transfere para a área de saída apenas o número de posições que faltam para completar até a sua margem direita.
- 3 - Chama rotina IMPL (Apêndice III.23) para efetuar a impressão.
- 4 - Transfere para a área de saída o restante das posições que não couberam no passo 2, retirando posições em branco desnecessárias.
- 5 - Se durante a transferência a área de saída ainda não for suficiente para conter todo o texto, os passos 2, 3 e 4 são executados novamente.
- 6 - Atualiza o ponteiro que marca a posição na área de saída a partir do qual o próximo caracter será movido e retorna.

### III.13 - ROTINA LEREG

Objetivo: Testar se já atingiu final de registro e avisar quando o arquivo Fonte já foi totalmente percorrido.

- 1 - Se o ponteiro que percorre cada registro for maior do que 80, aponta próximo registro. Se este for igual ao registro inicial do arquivo, é porque o mesmo já foi todo percorrido. Retorna.
- 2 - Le registro, descompacta-o para o formato A1 e retorna.

### III.14 - ROTINA CONCA

Objetivo: Transformar o conjunto de caracteres numéricos para um número na base 10.

- 1 - NUM ← 0
- 2 - Para cada dígito: formato A1  
NUM ← NUM + (dígito + 4032/256) \* 10<sup>n</sup>  
n: potência de 10 equivalente a casa decimal que ocupa.

### III.15 - ROTINA ORDEN

Objetivo: Imprimir em ordem alfabética o arquivo de referência

cruzada.

- 1 - Conta número de registros gravados (MAX) no arquivo (sem contar com o FLAG);  $I \leftarrow 0$ .
- 2 -  $I \leftarrow I+1$ ; se  $I > \text{MAX}$  vai para 7.  
Senão le registro: Se os 2 primeiros caracteres corresponderem a \$1, \$2 ou \$3 vai para 2.
- 3 -  $J2 \leftarrow I+1$ .
- 4 - Le registro J2. Se os 2 primeiros caracteres corresponderem a \$1, \$2 ou \$3, vai para 6.
- 5 - Compara o conteúdo do registro I com o do registro J2. Se a chave do 1º (as 5 primeiras letras são comparadas da seguinte forma: se a 1a. letra do registro I for igual a 1a. do registro J2, a 2a. letra é comparada, e assim sucessivamente) for superior a do 2º, os 2 registros trocam de posição entre si no arquivo. (Obs: Sempre que o registro seguinte a I ou J2 começar por \$1, \$2 ou \$3, estes acompanham também os respectivos registros I e J2 durante a troca, isto é, continuam sendo os registros seguintes a I e J2 no arquivo).
- 6 -  $J2 \leftarrow J2+1$ ; se  $J2 > \text{MAX}-1$  vai para 2, senão vai para 4.
- 7 - Aponta 1º registro do arquivo.
- 8 - Le registro, se começar por \$ chama rotina PONT (Apêndice III.22) e vai para 9. Senão, imprime registro (textos iguais são impressos apenas 1 vez acrescidos dos diversos números de página). Se atingir o número máximo de linhas por página, chama rotina PAGE (Apêndice III.9).
- 9 - Aponta próximo registro. Se este for superior a MAX retorna. Caso contrário vai para 8.

### III.16 - ROTINA ESCR

Objetivo: Imprimir os arquivos "back up", índice, títulos e rodapés quando estes forem solicitados.

- 1 -  $I \leftarrow 0$
- 2 -  $I \leftarrow I+1$ ; le registro I; se for igual ao FLAG:
  - 2.1 - Se o arquivo impresso for o de rodapés, inicializa arquivo.
  - 2.2 - Retorna.



- 3 - Se o registro começar por \$ vai para 7.
  - 3.1 - Senão, se começar por 'PPPP' muda de página.
  - 3.2 - Se o arquivo solicitado for o back up vai para 2.
  - 3.3 - Senão, chama rotina INIC1 (Apêndice III.24) e vai para 2.
- 4 - Se a linha atual impressa ultrapassar o limite de linhas por página:
  - 4.1 - Se o arquivo for de rodapês vai para 8.
  - 4.2 - Se o arquivo for o "back up": muda de página e vai para 6.
  - 4.3 - Chama rotina INIC1 e vai para 6.
- 5 - Se o arquivo impresso for o "back-up" e o usuário tiver solicitado impressão parcial, verifica se a linha atual deve ser impressa. Em caso negativo, vai para 2.
- 6 - Imprime informação, aponta a próxima linha da página e vai para 2.
- 7 - Chama rotina PONT (Apêndice III.22) para imprimir cedilhas, "overprint" e acentos. Se o \$ achado fizer parte do texto, vai para 6. Caso contrário:
  - 7.1 - Se a última linha impressa for a última da página.  
Se o arquivo for "back-up", muda de página e vai para 2.  
Senão, chama rotina INIC1 e vai para 2.
  - 7.2 - Caso contrário, vai para 2.
- 8 - Tratamento para impressão do rodapé:  
Conta o número de linhas que ainda faltam para a impressão do rodapé. Inicializa nova página (rotina INIC1), e aponta a posição da linha necessária para completar a impressão do rodapé (contando de baixo para cima, a partir da última linha determinada para a impressão do rodapé, o número de linhas que ainda faltam para completar a sua impressão). Para cada linha saltada na página, a rotina PROCl é executada. Vai para 2.

### III.17 - ROTINA PROC

Objetivo: Determinar quando a informação de um determinado registro contém cedilhas, acentos ou overprint. o código estabelecido é o seguinte:

\$1n... (para cedilhas)

\$2n... (para a acentuação)

\$3n... (overprint)

n: indica as posições onde isto se verifica no registro.

- 1 - Se não existir acentos, cedilhas e informação em "overprint" retorna.
- 2 - Testa cada posição da área reservada para cedilha (é a mesma para acento). Sempre que encontrar vírgula ou apóstrofe guarda o número da posição em áreas separadas (área 1 para cedilhas e área 2 para acentos).
- 3 - Executa o mesmo procedimento na área de "overprint" (área 3).
- 4 - Grava no registro seguinte do arquivo \$1 seguido da área 1, \$2 seguido da área 2 e \$3 seguido da área 3. Retorna.

### III.18 - ROTINA PROC1

Objetivo: Testar se existe ou não gravação no arquivo back-up.

- 1 - Se o usuário não houver solicitado arquivo back-up retorna.
- 2 - Caso contrário chama rotina GRAVA (Apêndice III.11) e retorna.

### III.19 - ROTINA SEPAR

Objetivo: Efetuar a separação de sílabas de acordo com a seguinte regra: qualquer 2 consoantes no final da linha deverão ser separados, exceto os grupos - BR, BL, BS, CR, CL, CH, DR, FR, FL, GR, GL, GU, LH, MN, NS, NH, PS, PR, PL, QU, TR.

- 1 - Marca última posição da margem direita pré-estabelecida.
- 2 - Se for consoante:
  - 2.1 - Se a posição anterior não for consoante vai para 3. Senão:
    - 2.1.1 - Se as 2 posições pertencerem ao grupo acima especificado, transfere a partir da posição atual até o final da linha para uma área auxiliar, pre

enchendo com brancos as posições anteriores.

Vai para 2.2.

2.1.2 - Senão, transfere a partir da próxima posição até o final da linha para uma área auxiliar, preenchendo as posições anteriores com brancos.

2.2 - Conta o número de posições transferidas e o número de intervalos da linha.

2.3 - Chama rotina TRANS (Apêndice III.20) para inserir posições em branco em cada intervalo da linha referentes ao número de posições transferidas. Se o número de posições transferidas for superior ao número de intervalos executa 2.3 novamente.

2.4 - Chama rotina LISTA (Apêndice III.26) e rotina TESTE (Apêndice III.27). Transfere as posições guardadas por 2.1.1 ou 2.1.2 para a área auxiliar de saída (corresponde a próxima linha), atualizando o ponteiro que indica a margem esquerda, a partir de onde os próximos caracteres serão transferidos nesta área em outros procedimentos. Retorna.

3 - Se a posição corresponder a um branco, vai para 2.1.2.

Senão, se o usuário não desejar separação de sílabas:

3.1 - Se a posição atual já corresponder a 1ª. posição da linha (margem esquerda), chama rotinas LISTA, TESTE, e retorna (a linha será impressa do jeito que se encontra).

4 - Marca posição anterior e vai para 2.

### III.20 - ROTINA TRANS

Objetivo: Distribuir o número de posições em branco referente às posições transferidas entre os intervalos da linha.

1 - Conta o número de intervalos a serem alterados a partir do último intervalo da linha. Marca esta posição.

2 - A partir da posição marcada, para cada intervalo que encontrar até o final da linha, acrescenta o número de brancos necessários (mesma quantidade para cada intervalo) para preencher a linha até margem direita estabelecida. Retorna.

### III.21 - ROTINA COMPA

Objetivo: Testar se existe a possibilidade de parte de um comando estar presente em um registro, e após várias posições em branco, o restante do comando só se encontrar no próximo registro.

- 1 - Se a posição do registro não for branco retorna.
- 2 - Senão, se o conteúdo a partir da posição atual até o final do registro for branco, chama rotina LEREG (Apêndice III.13). Se não houver percorrido todo o arquivo marca posição inicial do registro.
- 3 - Retorna.

### III.22 - ROTINA PONT

Objetivo: Imprimir as cedilhas, acentos e overprint na linha já impressa.

- 1 - Aponta próxima posição no registro.
- 2 - Se for diferente de 1, 2 ou 3, retorna.
- 3 - Caso contrário executa 3.1, 3.2 e 3.3 respectivamente.
  - 3.1 - Guarda numa área auxiliar (área 1) na posição referenciada por \$1n... a vírgula referente a cedilha. Vai para 4.
  - 3.2 - Guarda numa área auxiliar (área 2) na posição referenciada por \$2n... o apóstrofe referente ao acento. Vai para 4.
  - 3.3 - Guarda numa área auxiliar (área 3) o conteúdo existente na posição referenciada por \$3n... dentro do registro.
- 4 - Aponta próxima posição. Se for \$ vai para 2. Se for branco imprime áreas 1, 2 e 3 e retorna. Senão vai para 3.

### III.23 - ROTINA IMPL

Objetivo: Comandar a impressão da linha do relatório já formatada.

- 1 - Se estiver sob ação dos comandos início de título ou início de rodapé vai para 4.
- 2 - Aponta próxima linha da página. Se ultrapassar o limite de linhas por página, chama rotina PAGE (Apêndice III.9).
- 3 - Se estiver com a edição suprimida, chama rotina LISTA (Apêndice III.26), TESTE (Apêndice III.27) e retorna.
- 4 - Se o usuário não desejar separação de sílabas vai para 5.1.
- 5 - Se a última posição relativa a margem direita estabelecida for um algarismo, ou um símbolo qualquer (exceto letras):
  - 5.1 - Chama rotina SEPAR (Apêndice III.19) para transferir toda a palavra para a linha seguinte a partir do 1º branco à esquerda encontrado. Retorna.
- 6 - Chama rotina SEPAR e retorna.

### III.24 - ROTINA INIC1

Objetivo: Inicializar uma nova página.

- 1 - Muda de página.
- 2 - Se o arquivo back-up tiver sido solicitado chama rotina GRAVA (Apêndice III.11).
- 3 - Posiciona linha para numeração:
  - 3.1 - Para cada linha saltada chama rotina PROC1 (Apêndice III.18).
- 4 - Numera página e executa 2.
- 5 - Aponta próxima linha da página e retorna.

### III.25 - ROTINA FINAL

Objetivo: Comandar a impressão do índice por página e de referência cruzada.

- 1 - Se a impressão do índice tiver sido solicitada, chama rotina PAGE (Apêndice III.9) e ESCR (Apêndice III.16).
- 2 - Se a impressão do índice de referência cruzada tiver sido solicitada chama rotina PAGE e ORDEN (Apêndice III.15)
- 3 - Retorna.

### III.26 - ROTINA LISTA

Objetivo: Imprimir a linha do relatório já no formato final de edição.

- 1 - Se estiver com a edição suprimida vai para 4.
- 2 - Corta brancos em excesso da linha.
- 3 - Se estiver sob ação do comando de início de título, início ' de rodapé ou início de índice, chama rotina GRAVA ( Apêndice III.11) e vai para 7.
- 4 - Imprime a linha formatada, cedilhas e overprint. Se o arquivo back-up tiver sido solicitado chama rotina GRAVA.
- 5 - Se estiver sob ação do comando \$NCx,y,z, incrementa um contador de linhas específico. Se este for superior a x, o comando não fica fora de ação.
- 6 - Estabelece espaço entre cada linha, caso este seja diferente de 1. Para cada linha saltada: testa se já completou o limite da página. Em caso afirmativo, chama rotina PAGE; chama ' rotina PROCl.
- 7 - Se estiver sob ação do comando de fim de referência cruzada:  
7.1 - Chama rotina GRAVA para gravar as posições relativas ' ao índice de referência cruzada na área auxiliar.
- 8 - Retorna.

### III.27 - ROTINA TESTE

Objetivo: Redefinir a margem esquerda após a impressão de uma linha.

- 1 - Se estiver sob ação do comando de início de rodapé, define a margem esquerda para a posição fornecida pelo cartão de controle ou pelo sistema. Retorna.
- 2 - Se estiver sob ação do comando \$NCx,y,z, a margem esquerda ' recebe o valor y e retorna.
- 3 - Se estiver sob ação do comando \$ITx,y a margem esquerda recebe o valor x e retorna.
- 4 - A margem esquerda recebe o valor fornecido pelo sistema ou através dos cartões de controle. Retorna.

## APÊNDICE IV

### IV.1 - ROTINA IMPR

Objetivo: Efetuar a impressão do arquivo Fonte solicitado.

- 1 - Muda de página.
- 2 - Imprime nome do arquivo e aponta registro inicial.
- 3 - Le registro, descompacta-o para o formato A1 e imprime o número da linha e o seu conteúdo.
- 4 - Aponta próximo registro. Se já houver impresso todo arquivo retorna. Senão, vai para 3.

NO. REG.

T E X T O

100. \$IT2,20 7.2-GENERALIDADES \$FT \$NP0 COMANDO \$IRDIVERSAS VEZES. NESTA PUBLICA\$CECAO

200. USAMOS O TERMO COMANDO EM VEZ DE DECLARA\$CECAO.COMO DISSEMOS ANTERIORMENTE POSS

300. UEM O MESMO SIGNIFICADO. \$FR \$I000 \$FD E SEUS INTERVALOS SAO,INDICADOS ABAIXO. \$PL2 \$

400. NP25.....\$NP25..... \$NP25..... INTERVALO DO 'DO' \$NP25N.....\$N

500. PONDE I=\$APINDICE DA DECLARA\$CECAO \$I000 \$FC, SENDO DISPON\$APIVEL DURANTE A EXECUS

600. CECAO DO INTERVALO PODENDO SER USADO COMO SUBSCRITO DE ALGUMA VARI\$APIVEL OU \$AP

700. A DIRFITA DO SINAL E NUNCA \$APA ESQUERDA

800. \$NP9 \$IT2,20 \$APINDICES \$FT \$PL1 \$NP15 1-CONCEITOS GERAIS \$PL1 1.1-INTRODU\$CECAO..... \$

900. NP9

1000. \$PL1 1.2-ANALOGIA E CONCEITUA\$CECAO.....2 \$PL1 1.2.1ANALOGIA COM A M\$APAQUIN

1100. A DE CALCULAR DE MESA.....2 \$PL1 1.3-UNIDADES B\$APASICAS DE UM COMPUTADOR DI

1200. GITAL.VE\$APICULOS MAIS COMUNS DE ENTRADA E SA\$APIIDA.....5 \$PL1 3.1.1-VE\$APIC

1300. ULOS.....6 \$PL1 1.3.2-UNIDADES DE ENTRADAS E SAIDA.....6 \$FI

ACUT1  
ACUT1

TEXTC NAO ENCONTRADO NO REGISTRO 800.TEXTO EQUIVALENTE A -

\$PL1 \$II



NOME DO ARQUIVO-TESTIO

NL.REG.

T E X T O

100.            \$IT2,70    7.2-GENERALIDADES\$FT\$NPO COMANDO\$IRDIVERSAS VEZES NESTA PUBLICA\$CECAO  
 200.            USAMOS O TERMO COMANDO EM VEZ DE DECLARA\$CECAO.COMO DISSEMOS ANTERIORMENTE POSS  
 300.            UEM O MESMO SIGNIFICADO.\$FR\$IODO\$FO E SEUS INTERVALOS SAO ,INDICADOS ABAIXO.\$PL2\$  
 400.            NP25.....\$NP25.....\$NP25..... INTERVALO DO 'DO'\$NP25N.....\$N  
 500.            PONDE I=\$APINDICE DA DECLARA\$CECAO \$IODO\$FC, SENDO DISPON\$APIVEL DURANTE A EXECUS  
 600.            CECAO DO INTERVALO PODENDO SER USADO COMO SUBSCRITO DE ALGUMA VARI\$APAVEL OU \$AP  
 700.            A DIRFITA DO SINAL E NUNCA \$APA ESCUFROA  
 800.            \$NP9\$IT2,40\$APINDICES\$FT\$PL1\$NP15 1-CONCEITCS GERAIS\$PL1 1.1-INTRODU\$CECAO.....\$  
 900.            NPS  
 1000.           \$PL1 1.2-ANALOGIA E CONCEITUA\$CECAO.....2\$PL1 1.2.1ANALOGIA COM A M\$APAQUIN  
 1100.           A DE CALCULAR DE MESA.....2\$PL1 1.3-UNIDADES B\$APASICAS DE UM COMPUTADOR DI  
 1200.           GITAL.VE\$APICULCS MAIS COMUNS DE ENTRADA E SA\$APIDA.....5\$PL1 3.1.1-VE\$APIC  
 1300.           ULOS.....6\$PL1 1.3.2-UNIDADES DE ENTRADAS E SAIDA.....6\$FI

NOME DO ARQUIVO-ARQ1

NO.REG.

T E X T O

100. \$IT2,20 7.2-GENERALIDADES\$FT\$NPO COMANDO\$IRDIVERSAS VEZES NESTA PUBLICA\$CECAO

200. USAMOS O TERMO COMANDO EM VEZ DE DECLARA\$CECAO.COMO DISSEMOS ANTERIORMENTE POSS

300. UEM O MESMO SIGNIFICADO.\$FR\$IODO\$FO E SEUS INTERVALOS SAO INDICADOS ABAIXO.\$PL2\$

400. NP35.....\$NP35.....\$NP35..... INTERVALO DO 'DO'\$NP35N.....\$N

500. PONDE I=\$APINDICE DA DECLARA\$CECAO \$IODO\$FC, SENDO DISPON\$APIVEL DURANTE A EXECU\$

600. CECAO DO INTERVALO PODENDO SER USADO COMO SUBSCRITO DE ALGUMA VARI\$APAVEL OU \$AP

700. A DIREITA DO SINAL E NUNCA \$APA ESQUERDA.\$IT2,40\$APINDICE\$FT\$II\$NP30 1-CONCEITOS

800. GERAIS\$NP30 1.1-INTRODU\$CECAO.....1\$NP30 1.2

900. -ANALOGIA E CONCEITUA\$CECAO.....2\$NP30 1.2-1-ANALOGIA COM A

1000. M\$APAQUINA DE CALCULAR DE MESA.....4\$NP30 1.3-UNIDADES B\$APASICAS DE UM COMPUTA

1100. DOR DIGITAL.....5\$NP30 1.3.1-VE\$APICULOS MAIS COMUNS DE ENTRADA E SA\$APIDA...

1200. ....8\$NP30 1.3.2-UNIDADES DE ENTRADA E SA\$APIDA.....10\$FI

C COMANDO  
DO E SEUS INTERVALOS SAO INDICADOS ABAIXO.

.....  
.....  
..... INTERVALO DO 'DO'  
N.....  
CNDE I=INDICE DA DECLARAÇÃO DO, SENDO DISPONIVEL DURANTE A  
EXECUÇÃO DO INTERVALO PODENDO SER USADO COMO SUBSCRITO DE AL-  
GUMA VARIÁVEL OU A DIREITA DO SINAL E NUNCA A ESQUERDA.

DIVERSAS VEZES NESTA PUBLICAÇÃO USAMOS O TERMO COMANDO EM VEZ  
DE DECLARAÇÃO. COMO DISSEMOS ANTERIORMENTE POSSUEM O MESMO SIG-  
NIFICADO.

INDICE

1-CONCEITOS GERAIS	
1.1-INTRODUÇÃO.....	1
1.2-ANALOGIA E CONCEITUAÇÃO.....	2
1.2.1-ANALOGIA COM A MAQUINA DE CALCULAR DE MESA.....	4
1.3-UNIDADES BASICAS DE UM COMPUTADOR DIGITAL.....	5
1.3.1-VEICULOS MAIS COMUNS DE ENTRADA E SAIDA.....	8
1.3.2-UNIDADES DE ENTRADA E SAIDA.....	10

NOME DO ARQUIVO-TEST9

NC.REG.

T E X T O

100.           \$NP30I-ESPECIFICAS\$CECOES PROPRIAMENTE DITAS.CONVERSAO DE DADOS NUM\$APERICOS

200.           \$PL2\$NPOS VALORES DOS DADOS DE ENTRADA OU DOS RESULTADOS NA SA\$API DA DE UM

300.           PROC\$SSAMENTO,EM SUAS FORMAS EXTERNAS,PODERAO SER DO \$IXTIPO REAL\$FX OU

400.           \$IXINTEIROS\$FX COMO J\$APA ESTUDAMOS.ESTAS INFORMA\$CECOES DEVEM SER DEVIDAMENTE

500.           CODIFICADAS E AGRUPADAS EM UM CERIO CAMPO DE COMPRIMENTO DETERMINADO.AL\$AP

600.           EM DISTO ,O CASO DAS QUANTIDADES REAIS QUE NAO TRAZEM O PONTO,\$APE PRECISO

700.           SABER ONDE EST\$APA O PONTO DECIMAL E SE A QUANTIDADE FOR MUITO GRANDE UMA

800.           MANEIRA COMPACTA E ELEGANTE DE REPRESENT\$APA-LA.TODA ESTA INFORMA\$CECAO

900.           DEVE SER ENTREGUE AO \$IXCOMPILADOR\$FX,USANDO-SE PARA ISSO OS C\$APODIGOS DE

1000.           CONVERSAO,OU TIPO DE ESPECIFICAS\$CECAO,CONTIDOS NA DECLARA\$CECAO \$IXFORMA

1100.           T\$FX.LEMBRAMOS QUE PARA CADA \$IXVARI\$APAVEL\$FX NA LISTA DE UMA DECLARA\$C

1200.           ECAO \$IXREAD\$FX E \$IXWRITE\$FX DEVE CORRESPONDER A UM DESTES \$IXC\$APODIGOS

1300.           \$FX NA DECLARA\$CECAO FORMAT CORRESPONDENTE.OS C\$APODIGOS OU ESPECIFICAS

1400.           CECOES SAO OS SEGUINTES \$NP40IW\$NP40EW.D\$NP40FW.D\$NP30I.1.ESPE

1500.           CIFICAS\$CECAO IW\$NPO 'I' SIGNIFICA QUE A \$IXVARI\$APAVEL\$FX DA LISTA A SER LI

1600.           DA,IMPRESSA CU PERFURADA \$APE DO TIPO INTEIRO.\$NP30I.2-FORTRAN B\$APASICO

1700.           \$NPAT\$APE AQUI FOI ESTUDADO O CONJUNTO DE DECLARA\$CECOES QUE SE DIZ CONS

1800.           TITUIR O \$IX\$I OFORTRAN B\$APASICO\$FO\$FX.COM O QUE FOI APRES\$NTADO AT\$APE ES

1900.

TE CAP\$APITULO,O LEITOR J\$APA TEM FERRAMENTA NECESS\$APARIA PARA RESOLVER

2000.

CERTOS TIPOS DE PROBLEMAS.\$NP\$IT2,30\$APINCICE ALFAB\$APETICO\$FT

## I-ESPECIFICAÇÕES PROPRIAMENTE DITAS. CONVERSÃO DE DADOS NUMÉRICOS

OS VALORES DOS DADOS DE ENTRADA OU DOS RESULTADOS NA SAÍDA DE UM PROCESSAMENTO, EM SUAS FORMAS EXTERNAS, PODERÃO SER DO TIPO REAL QUINTEIRO COMO JÁ ESTUDAMOS. ESTAS INFORMAÇÕES DEVEM SER DEVIDAMENTE CODIFICADAS E AGRUPADAS EM UM CERTO CAMPO DE COMPRIMENTO DETERMINADO. ALÉM DISTO, O CASO DAS QUANTIDADES REAIS QUE NÃO TRAZEM O PONTO, É PRECISO SABER ONDE ESTÁ O PONTO DECIMAL E SE A QUANTIDADE FOR MUITO GRANDE UMA MANEIRA COMPACTA E ELEGANTE DE REPRESENTÁ-LA. TOCA ESTA INFORMAÇÃO DEVE SER ENTREGUE AO COMPILADOR, USANDO-SE PARA ISSO OS CÓDIGOS DE CONVERSÃO, OU TIPO DE ESPECIFICAÇÃO, CONTIDOS NA DECLARAÇÃO FORMAT. LEMBRAMOS QUE PARA CADA VARIÁVEL NA LISTA DE UMA DECLARAÇÃO READ E WRITE DEVE CORRESPONDER A UM DESTES CÓDIGOS NA DECLARAÇÃO FORMAT CORRESPONDENTE. OS CÓDIGOS OU ESPECIFICAÇÕES SÃO OS SEGUINTE

IW

EW.D

FW.D

### I.1. ESPECIFICAÇÃO IW

C 'I' SIGNIFICA QUE A VARIÁVEL DA LISTA A SER LIDA, IMPRESSA OU PERFORADA É DO TIPO INTEIRO.

### I.2-FORTRAN BÁSICO

ATE AQUI FOI ESTUDADO O CONJUNTO DE DECLARAÇÕES QUE SE DIZ CONSTITUIR O FORTRAN BÁSICO. COM O QUE FOI APRESENTADO ATE ESTE CAPÍTULO, O LEITOR JÁ TEM FERRAMENTA NECESSÁRIA PARA RESOLVER CERTOS TIPOS DE PROBLEMAS.

## INDICE ALFABETICO

## C

CODIGOS 1  
COMPILADOR 1

## F

FORMAT 1  
FORTRAN BASICO 1

## I

INTEIRO 1

## R

READ 1

## T

TIPO REAL 1

## V

VARIAVEL 1

## W

WRITE 1