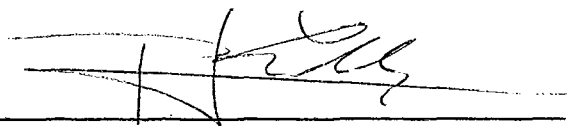


ESTUDO E REALIZAÇÃO DE UM SISTEMA DE CADASTROS
PORTÁTIL ORIENTADO PARA USUÁRIO LEIGO EM COMPUTAÇÃO
(PARTE I - MONTAGEM E ANÁLISE CRÍTICA
DOS CADASTROS)

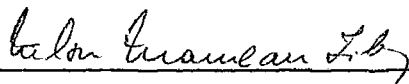
Maria Celia Arruda Grillo

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

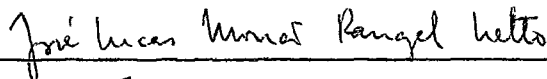
Aprovada por:



Prof. Pierre-Jean Lavelle
(Presidente)



Prof. Nelson Maculan Filho



Prof. José Lucas M. Rangel Netto

RIO DE JANEIRO - BRASIL

OUTUBRO DE 1977

GRILLO, MARIA CÉLIA ARRUDA

Estudo e Realização de um Sistema de Cadastros Portátil Orientado para Usuário Leigo em Computação (Parte I - Montagem e Análise Crítica) |Rio de Janeiro| 1977.

XI, 190p. 29,7cm (COPPE-UFRJ, M.Sc., Engenharia de Sistemas e Computação, 1977)

Tese - Univ. Fed. Rio de Janeiro. Fac. Engenharia

1. Banco de Dados I. COPPE/UFRJ II. Estudo e Realização de um Sistema de Cadastros Portátil Orientado para Usuário Leigo em Computação (Parte I - Montagem e Análise Crítica).

A · ANTONIO J. C. GRILLO

e ao meu filho

· MARCOS

A G R A D E C I M E N T O S

A minha admirável mãe CELINA DE MORAES ARRUDA, que me incutiu o espírito de LUTA PELOS GRANDES IDEAIS.

Ao meu marido DR. ANTONIO JOSÉ DA COSTA GRILLO, que com seu entusiasmo contagiante, muito me incentivou e soube contribuir para a publicação desta obra.

Ao meu filho MARCOS ARRUDA GRILLO, que antes mesmo de nascer me acompanhou em toda fase experimental deste trabalho, e que com sua graça e formosura tudo transforma em flores e alegria.

Ao DR. JEAN-PIERRE LAVELLE pela GRANDE IMAGEM que representa para a COPPE e pelas magníficas sugestões na orientação desta tese.

Ao DR. NELSON MACULAN FILHO pelo seu valor e sábias orientações dedicadas à COORDENAÇÃO DOS PROGRAMAS DE PÓS GRADUAÇÃO EM ENGENHARIA DE SISTEMAS.

Ao DR. JOSÉ LUCAS MOURÃO RANGEL NETTO pela sua participação na BANCA DE TESE.

Aos meus irmãos, de modo especial ao DR. ALLYRIO ÁQUILAS DE MORAES ARRUDA; antigos mestres, colegas de estudo, e pessoas amigas, pelo que de alguma maneira contribuíram para minha formação intelectual especializada.

À memória de meu pai TRAVASSO DOS SANTOS ARRUDA.

R E S U M O

Esta obra objetiva a elaboração de um sistema de cadastros em fita magnética que permite implantação em vários computadores de diferentes fabricantes sem alterações significativas. O projeto de montagem e análise crítica dos arquivos inclui não só o estudo da viabilidade das técnicas empregadas como também visa à simplificação das atividades do Sistema Operacional. Destina-se a uma clientela leiga em processamento de dados e preocupa-se de modo especial com a fase de consistência dos dados .

A B S T R A C T

This work examines the design of a file system in magnetic tape that allows implementation in several different computer manufacturers, without significant amount of modification. The study of file organization and file consistent analysis emphasizes not only the techniques which were used, but also tries to Simplify the Operating System interface. The system is designed to occasional users and special treatment is given to data consistence.

Í N D I C E

	Páginas
INTRODUÇÃO	1
CAPÍTULO I - CONSIDERAÇÕES INICIAIS	3
1.1. - OBJETIVOS	3
1.2. - ARQUIVOS UTILIZADOS	4
1.3. - DESCRIÇÃO DO BLOCO FÍSICO	6
1.4. - MODO DE GRAVAÇÃO	7
1.5. - FASES DO SISTEMA	8
1.5.1. - GERAÇÃO E ANÁLISE CRÍTICA DA TABELA DE SÍMBOLOS	8
1.5.2. - VALIDAÇÃO DE DADOS	9
1.5.3. - UTILIZAÇÃO DO SISTEMA	10
CAPÍTULO II - APRESENTAÇÃO DOS DOIS MÓDULOS	12
2.1. - GERAÇÃO DA TABELA DE SÍMBOLOS	12
2.1.1. - DESCRIÇÃO GERAL	12
2.1.2. - CAMPOS GERADOS NOS REGISTROS LÓGICOS	14
2.1.3. - CARACTERÍSTICAS DO LISTÃO	18
2.1.4. - ESTRUTURAÇÃO DA ENTRADA DE DADOS E O EMPREGO DE "FORMAT" LIVRE	20
2.1.5. - AUTOCONTROLE "VERSUS" INDEPENDENTI- ZAÇÃO DO USUÁRIO	22
2.1.6. - FORMATO INTERNO DA TABELA DE SÍMBO- LOS	23

	Páginas
2.2. - VALIDAÇÃO DE DADOS	25
2.2.1. - DESCRIÇÃO GERAL	25
2.2.2. - ESTRUTURAÇÃO DA ENTRADA DE DADOS..	25
2.2.3. - TIPOS DE ENTRADA	26
2.2.4. - DECODIFICAÇÃO DOS ELEMENTOS DAS LISTAS	27
2.2.5. - TESSITURA DOS "NÓS" QUE COMPÕEM OS CADASTROS EM FITA MAGNÉTICA	28
2.2.6. - FORMATO INTERNO DOS CADASTROS	28
2.2.7. - "WORKING-STORAGES" E O CADASTRAMEN <u>T</u> <u>T</u> O PROVISÓRIO	30
2.2.8. - ESTABELECIMENTO DE "DEFAULTS" DO SISTEMA	30
 CAPÍTULO III - GERAÇÃO DA TABELA DE SÍMBOLOS (1º MÓDU <u>L</u> LO)	 33
3.1. - GERAÇÃO EM BLOCOS	33
3.2. - DESCRIÇÃO GLOBAL DA ESTRUTURA DE DADOS	36
3.3. - FASES DA GERAÇÃO	39
3.3.1. - "IERRON(N)" - IMPRESSÃO DE MENSA- GENS DE ERRO	39
3.3.2. - "INDTAM" - DETERMINAÇÃO DO ENDERE- ÇO DA 1ª. LISTA DA TABELA DE SÍMBO <u>L</u> LOS	41
3.3.3. - "LELIST" - GERAÇÃO DO LISTÃO DA TA <u>B</u> <u>B</u> ELA DE SÍMBOLOS	44

	Páginas
3.3.4. - "IENDER" - GERAÇÃO DE ENDEREÇOS DAS LISTAS	46
3.3.5. - "ISAI(L)" - GRAVAÇÃO DE CARACTE <u>R</u> RES INDIVIDUAIS EM FITA MAGNÉT <u>I</u> CA	48
3.3.6. - "MLTIMP(N)" - GRAVAÇÃO EM FITA ADAPTADA PARA QUANTIDADES MAIO- RES DE DADOS	50
3.3.7. - "KONVER" - DESCOMPACTAÇÃO DOS ENDEREÇOS DAS LISTAS	52
3.3.8. - "KONALE" - CONVERSÃO DE NUMÉRI- COS EM ALFANUMÉRICOS	53
3.3.9. - PROGRAMAS DE PERCURSO E ANÁLISE DO ARQUIVO DE ENTRADA	55
3.3.9.1-"NAMEL"-GERAÇÃO DE NOME DO DADO NA TABELA DE SÍMBOLOS	55
3.3.9.2-"KTIPO"-GERAÇÃO DE TIPO DE DADO	58
3.3.9.3-"ICOGI"-GERAÇÃO DA ORGA <u>N</u> NIZAÇÃO DO DADO	63
3.3.9.4-"LGRILL"-GERAÇÃO DO MO- DO DE GRAVAÇÃO E TAMA- NHO DO DADO	67
CAPÍTULO IV - VALIDAÇÃO DE DADOS (2º MÓDULO)	72

	Páginas
4.1. - DIAGRAMA DE BLOCOS DA VALIDAÇÃO.....	72
4.2. - ANÁLISE DA VALIDAÇÃO DE DADOS	75
4.2.1. - ESTRUTURA GERAL	75
4.2.2. - TIPOS DE ENTRADA	76
4.2.3. - CORRELAÇÃO NO PROCESSAMENTO	77
4.3. - FASES DA VALIDAÇÃO	80
4.3.0. - TABELA DE SÍMBOLOS-MODELO	80
4.3.1. - "KLAUDI" - PERCURSO E VALIDAÇÃO DO NOME DE DADO DESCRITO PELO U- SUÁRIO	81
4.3.2. - "NAIEFV" - PESQUISA DO TIPO, OR- GANIZAÇÃO E MODO DE GRAVAÇÃO DOS DADOS ATRAVÉS DE PERCURSO DA TA- BELA DE SÍMBOLOS	87
4.3.3. - "KALFCK" - PERCURSO E VALIDAÇÃO DE DADOS ALFANUMÉRICOS	94
4.3.4. - "NUMCHK" - PERCURSO E VALIDAÇÃO DE DADOS NUMÉRICOS	96
4.3.5. - "LISTCK" - PERCURSO E VALIDAÇÃO DE DADOS EM LISTA	98
4.3.6. - "LAUXER" - TESSITURA DE DADOS INDEPENDENTES NO MINICADASTRO "LYAUX(M)" VISANDO A POSTERIOR TRANSLADAÇÃO DO REGISTRO PARA O DISPOSITIVO DE MEMÓRIA AUXI- LIAR	103

	Páginas
4.3.7. - "LISCOD" - DECODIFICAÇÃO DE DADOS COM ORGANIZAÇÃO DE LISTA EXCLUSI- VA E TESSITURA DOS MESMOS NO MINI CADASTRO "LYAUX (M)"	111
4.3.8. - "LEXCOD" - DECODIFICAÇÃO DE DADOS COM ORGANIZAÇÃO DE LISTA INCLUSI- VA E TESSITURA DOS MESMOS NO MINI CADASTRO "LYAUX (M)"	118
4.3.9. - "KONVTA" - DECOMPACTAÇÃO DE NÚME- ROS COMO FASE PREPARATÓRIA PARA GRAVAÇÃO DE TAMANHO DOS DADOS NO MINICADASTRO "LYAUX (M)"	123
4.3.10.- "KALFTA" - ALOCAÇÃO DE TAMANHO DO DADO NO MINICADASTRO "LYAUX (M)"..	125
4.3.11.- "KTURN" - TRANSFORMAÇÃO DOS ENDE- REÇOS DAS LISTAS GRAVADAS EM FITA COM A CONFIGURAÇÃO DE ALFANUMÉRI- COS PARA NÚMEROS DECIMAIS	126
4.3.12.- "MLTLIS" - SINDICÂNCIA E CADASTRA MENTO DE DADOS COM ORGANIZAÇÃO DE LISTA INCLUSIVA, NOS CASOS DE EN- TRADA MÚLTIPLA.....	128
4.3.13.- "KDEFAU" - ESTABELECIMENTO DE "DEFAULTS" DO SISTEMA, PARA ATEN- DER ÀS NECESSIDADES DE CADASTRA- MENTO, NOS CASOS EM QUE O USUÁRIO NÃO DISPÕE DE TODAS AS INFORMAÇÕES DE ENTRADA	131

	Páginas
CAPÍTULO V - CONCLUSÕES	140
BIBLIOGRAFIA	142
ANEXO I - MANUAL DO USUÁRIO	143
ANEXO II - GLOSSÁRIO DE ERROS	147
ANEXO III - LISTAGENS	151

INTRODUÇÃO

INTRODUÇÃO

Muito se tem feito na área de computação sobre arquivos em disco e fita e é cada vez maior a necessidade de MÃO-DE-OBRA ESPECIALIZADA para operação desses sistemas.

Sempre que um novo sistema é implantado, tem-se um período de adaptação e aprendizagem por parte do usuário, com cursos especiais que o preparam para o novo tipo de trabalho. Tudo isso, quando não são feitos contratos a especialistas no ramo .

Muitas empresas relutam em implantar um novo sistema, devido a essa consequência. Quem poderá utilizá-lo? Somente as empresas de grande porte estão preparadas para tal empreendimento.

Esta consequência, drástica para alguns, é a característica principal deste trabalho, cuja manipulação independe de programador. É um sistema feito para atender às necessidades de um usuário leigo em computação. Seguindo apenas as instruções de um manual, o usuário pode utilizar e atualizar o sistema .

Dois ARQUIVOS LINEARES, gravados em fita são elaborados, contendo cada um a Tabela de Símbolos, que é a sua própria descrição e a massa de dados codificados.

O usuário COMUNICA-SE COM O SISTEMA, através de cartões em "FORMAT" livre e em LINGUAGEM CORRENTE. Lê os

dados e, finalmente, faz a montagem dos arquivos.

Os pedidos de informação podem ser em forma de listagem, contagem ou ordenação de quaisquer dados cadastrados e até mesmo sob condições estipuladas.

Na utilização, o usuário pode manipular os cadastros para deles extrair as informações que desejar. Já na atualização, pode retirar ou trocar dados, modificando-lhes parcial ou totalmente a estrutura.

CAPÍTULO I

CONSIDERAÇÕES INICIAIS

CAPÍTULO I

CONSIDERAÇÕES INICIAIS

1.1. - OBJETIVOS

Esta é tentativa de idealizar um sistema otimizado que inclua não somente a ANÁLISE DA VIABILIDADE das técnicas que serão empregadas, como também a obtenção de um produto final, que funcione na prática.

Constitui objetivo principal, apresentar uma contribuição ao Departamento de Ensino de 2º grau, da Secretaria de Educação e Cultura, para automação de seus processos de controle de arquivos, através da transformação e manipulação controlada da informação de maneira simbólica.

Convém esclarecer, desde o início, alguns detalhes importantes:

- 1 - Este trabalho não envolve nenhum detalhe de "HARDWARE", de qualquer computador específico.
- 2 - A linguagem usada é "FORTRAN", para permitir a implantação em vários computadores de diferentes fabricantes sem alterações significativas.

3 - A tentativa de simplificação das tarefas do SISTEMA OPERACIONAL é uma constante no projeto do sistema. Baseado em estudos já feitos em UNIVERSIDADES NORTE-AMERICANAS (8) (10) com esse objetivo, visa-se apenas a maior eficiência no processamento, com resultados compensadores.

A geração do sistema está dividida em duas TESES DE MESTRADO, na COPPE:

- . MONTAGEM E ANÁLISE CRÍTICA DOS CADASTROS (VOLUME I).
- . UTILIZAÇÃO E ATUALIZAÇÃO DOS CADASTROS (VOLUME II).

1.2. - ARQUIVOS UTILIZADOS

Os dois cadastros:

- . DE FUNCIONÁRIOS
- . DE ESTABELECIMENTOS

estão gravados em FITA MAGNÉTICA e têm ORGANIZAÇÃO SEQUENCIAL.

A esta escolha estão aliadas várias razões, como por exemplo:

- . ECONOMIA EM ESPAÇO DE ARMAZENAGEM

A CAPACIDADE DE COMPACTAÇÃO DA FITA possibilita gravação de dados a uma densidade consideravelmente superior. No sistema em pauta, onde a quantidade de dados aumenta gradativamente em períodos de tempo variáveis, esta análise é extremamente importante.

. VANTAGEM EM CUSTO

A DENSIDADE DE GRAVAÇÃO DA FITA faz com que ela se constitua num meio de arquivo de baixo custo.

. ALTA VELOCIDADE

A fita magnética é um dos MEIOS MAIS VELOZES DE ENTRADA E SAÍDA de dados. Por isso, embora o tempo de acesso aos registros seja um pouco demorado, há uma compensação, o que justifica a escolha.

. MAIOR EFICIÊNCIA QUANDO A MOVIMENTAÇÃO DO ARQUIVO CRESCE

Afora seu valor para grandes cadastros, a fita é mais bem usada se existir MUITA MOVIMENTAÇÃO no arquivo. Este é, justamente, o caso considerado, pois NUMEROSAS ATUALIZAÇÕES são necessárias constantemente.

. POSSIBILIDADE DE USO DE MINI E ATÉ MICROCOMPUTADOR COM ARQUIVOS EM SUPORTES SEQUENCIAIS

Este fato descortina perspectivas de alto valor, pois permite GENERALIZAR O TIPO DE EQUIPAMENTO a ser utilizado pelo sistema.

. "SOFTWARE" SIMPLIFICADO

Entre outros fatores positivos da fita magnética, pode-se aceder como irrefutável sua simplicidade de programação.

1.3. - DESCRIÇÃO DO BLOCO FÍSICO

O "BUFFER" FÍSICO representado por BLOCOS FÍSICOS DE TAMANHO FIXO contém vários REGISTROS LÓGICOS OU FRAÇÕES DESTES.

REGISTRO LÓGICO - é um conjunto de dados agrupados que tem um significado completo e determinado.

É comum em processamento de dados a necessidade de se reunir diversas informações, a fim de facilitar e dar maior velocidade às operações efetuadas sobre os dados. Chama-se REGISTRO FÍSICO ou BLOCO FÍSICO qualquer um desses conjuntos.

Considera-se aqui um tamanho de bloco igual ao LIMITE MÁXIMO DAS RODAGENS do computador utilizado, analisando:

- . espaço disponível na memória central, para acomodar as memórias intermediárias
- . obtenção de um bom equilíbrio ENTRADA/SAÍDA para os programas.

- . minimização da quantidade de fita magnética exigida para armazenar o arquivo; em consequência REDUÇÃO DO TEMPO necessário ao seu processamento.
- . necessidade de sujeição às normas do "SOFTWARE" padrão, que vai ser usado no processamento dos cadastros.

Para fins de estudo, as listagens do ANEXO 3 consideram um TAMANHO DE BLOCO igual a 60 CARACTERES.

1.4. - MODO DE GRAVAÇÃO

Um fator que tem em mira a otimização da quantidade de fita magnética é a escolha do MODO DE GRAVAÇÃO. Como nos diferentes blocos físicos existem grupos de dados que, em relação ao seu tamanho, podem ou não sofrer alterações, utilizam-se os dois modos:

- . TAMANHO FIXO
- . TAMANHO VARIÁVEL

de maneira que somente a área efetivamente ocupada por informação é registrada, evitando desperdícios.

A presença de registros de TAMANHO VARIÁVEL reduz a quantidade de armazenamento e, portanto, minimiza o tempo de I/O, embora sua utilização seja mais complexa.

No caso em que só se pretendesse o emprego de registros de TAMANHO FIXO, as variações de tamanho dos dados levariam a concluir que a introdução de pequenas complexidades e problemas de processamento aliados à utilização de registros variáveis constitui melhor solução.

1.5. - FASES DO SISTEMA

1.5.1. - GERAÇÃO E ANÁLISE CRÍTICA DA TABELA DE SÍMBOLOS

Os cadastros se apresentam com uma estrutura inicial (TABELA DE SÍMBOLOS), NECESSÁRIA E SUFICIENTE PARA TODAS AS MANIPULAÇÕES. Esta constitui o conteúdo dos primeiros registros físicos do arquivo, permitindo que os PROGRAMAS desenvolvidos sejam INDEPENDENTES DOS DADOS.

Sua composição é uma descrição simbólica dos elementos codificados e, por sua fundamental importância, RESIDE EM MEMÓRIA, durante todo tempo de execução.

É formada por um conjunto de informações que descrevem:

a) OS DADOS DE ENTRADA, através de:

- . nome do campo
- . delimitador

- . "bit" de listar ou contar
- . "bit" de testar
- . tipo de dado
- . organização
- . modo de gravação
- . tamanho
- . endereço das listas

b) AS LISTAS ACOPLADAS AO SISTEMA - que têm objetivo de minimizar o tamanho dos cadastros, através de certas convenções, que referenciam seus elementos através de simples decodificações.

1.5.2. - VALIDAÇÃO DOS DADOS

A precisão nos dados cadastrados é fator de máxima importância, por isso os registros errados são rejeitados antes de entrarem no sistema.

Trata-se, pois, de um sistema AUTOCONTROLÁVEL, e qualquer ocorrência estranha, capaz de prejudicar o processamento, é automaticamente detectada e eliminada.

Para evitar a utilização de dados errados, foram galvanizados diversos controles que nos asseguram de tal. Os erros revelam-se os mais variados, e sua ocorrência é normalmente atribuída a possíveis incúrias na utilização do MANU

AL DO USUÁRIO.

Nestes casos, o sistema envia mensagem, assinalando a espécie de erro, e prossegue suas tarefas automaticamente.

Uma vez aceitos como válidos, os registros são organizados no formato exigido para arquivos em fita magnética e então editados. É a fase de MONTAGEM DOS CADASTROS.

1.5.3. - UTILIZAÇÃO DO SISTEMA

Estando os dois arquivos montados em fita o usuário pode utilizá-los para obter qualquer dado cadastrado.

Os três tipos básicos de PEDIDOS DE INFORMAÇÃO: LISTAR, CONTAR e ORDENAR admitem duas possibilidades distintas:

- 1 - SEM CONDIÇÃO
- 2 - COM UMA OU MAIS CONDIÇÕES

1 - SEM CONDIÇÃO :

- . LISTAR estabelecimentos : SEMPRE ;
- . CONTAR salas especiais : SEMPRE ;
- . ORDENAR matrículas : SEMPRE ;

2 - COM UMA OU MAIS CONDIÇÕES :

- . LISTAR nome i matrícula : tempo . de .
serviço > 25 anos i
- . CONTAR professor : função = coordenador i
colégio = Escola . modelo i
- . ORDENAR classificação . de . alunos : média
> 7 i

Os elementos não sublinhados representam as informações objetivadas pelo usuário (dados ou condições) no formato convencionado pelo sistema. Maiores detalhes ver CARVALHO (13) .

CAPÍTULO II

APRESENTAÇÃO DOS DOIS MÓDULOS

CAPÍTULO II

APRESENTAÇÃO DOS DOIS MÓDULOS

2.1. - GERAÇÃO DA TABELA DE SÍMBOLOS

2.1.1. - DESCRIÇÃO GERAL

A TABELA DE SÍMBOLOS compreende uma descrição simbólica dos dados de entrada. Estes são gerados de acordo com o tipo de informação a ser manipulada e podem ser alterados, conforme necessidade do usuário a respeito de maior ou menor quantidade de informação, o que caracteriza a INSERÇÃO ou RETIRADA de elementos dos cadastros.

Essa tabela é composta de:

- . um conjunto de REGISTROS LÓGICOS, que descrevem as características essenciais de cada dado, tais como: tipo, organização, tamanho, modo de gravação e endereço das LISTAS
- . um LISTÃO que é erigido com o escopo de descrever, para alguns dados, todas as possibilidades de entrada aventadas pelo usuário, visando a simplificação do cadastro em fitas, através de melhor emprego do espaço dis

ponível com a gravação de dados decodificados. A utilização desse artifício adstringe-se a elementos cujo campo de variação é conhecido "a priori" e, o total de listas depende em nº e conteúdo da informação a ser registrada.

Assim sendo, tem-se:

TABELA		REGISTROS LÓGICOS		
DE	=	DE	+	LISTÃO
SÍMBOLOS		DESCRIÇÃO DE DADOS		

A figura 2.1.A, abaixo, contém um esquema da disposição física de cada registro lógico da Tabela de Símbolos. Seu espectro mostra a separação dos diversos campos existentes e o tipo básico de informação que eles contêm.

NOME DO DADO	DELIMI- TADOR	"BIT" DE LISTAR OU CONTAR	"BIT" DE TESTAR	TIPO	ORGA- NIZA- ÇÃO	MODO DE GRAVAÇÃO	TAMANHO	ENDEREÇO DA LISTA
--------------------	------------------	---------------------------------	-----------------------	------	-----------------------	------------------------	---------	-------------------------

Fig. 2.1.A - ESQUEMA DE UM REGISTRO LÓGICO DA TABELA DE SÍMBOLOS.

2.1.2. - CAMPOS GERADOS NOS REGISTROS LÓGICOS

A conexão entre cada campo do registro lógico e o papel que ele desempenha pode ser sintetizada da seguinte maneira:

I - NOME DO DADO

É o primeiro campo do registro e tem, como objetivo a IDENTIFICAÇÃO DO DADO. Entre todos, é o único dotado de comprimento VARIÁVEL, pois não se pode estatuir previamente quantos caracteres ele contém.

II - DELIMITADOR DO DADO

Este campo é composto de um único caractere , cujo objetivo é marcar o término da descrição de nome do dado. O sistema convencionou sua designação através de um asterisco (*).

III - "BITS" DE LISTAR OU CONTAR E TESTAR

Parte do registro representada por 2 caracteres que na fase de utilização do sistema desempenham função de CHAVE DE CONTROLE. Durante a geração dos cadastros, eles são representados no registro lógico por dois zeros que caracterizam seu estado inicial: DESLIGADA.

IV - TIPO DE DADO

Este campo também ocupa uma posição do regis-

tro lógico e sua função é aferir de modo sistemático uma indicação para explicitar a presença de dados:

. ALFANUMÉRICOS

. NUMÉRICOS

Sua representação no registro lógico utiliza o caractere "A" para alfanuméricos e "N" para numéricos.

V - ORGANIZAÇÃO DO DADO

A parte de organização do dado é representada por um campo de comprimento fixo, que compreende 2 caracteres utilizados para simbolizar:

. DADO INDEPENDENTE

. DADO EM LISTA

. INDEPENDENTE - é a designação escolhida para indicar dados isolados que assumem valores diferentes para cada elemento do cadastro.

Sua representação no registro lógico da Tabela de Símbolos é composta por uma seqüência de dois zeros (0 0).

. EM LISTA - é a designação escolhida para representar dados cuja gama de variação é em número limitado .

Cada elemento do cadastro pode assumir NENHUM ou PELO MENOS UM dos valores explicitados no listão que reside na Tabela de Símbolos.

Para aceder certa OTIMIZAÇÃO NO CADASTRAMENTO, o sistema convencionou dois tipos fundamentais de listas:

- . LISTAS EXCLUSIVAS
- . LISTAS INCLUSIVAS

Sua representação no registro lógico utiliza "LE" ou "LI".

VI - MODO DE GRAVAÇÃO - é a parte do registro lógico que faz referência ao tipo de gravação utilizado para armazenar os dados nos respectivos cadastros.

Subdivide-se em:

- . TAMANHO FIXO
- . TAMANHO VARIÁVEL

Sua representação no registro lógico consta de 2 caracteres, que são: "TF" ou "TV".

TAMANHO FIXO - modo de gravação utilizado para dados, cujo total de caracteres envolvidos na descrição do usuário PODE SER PREVIAMENTE AVALIADO.

TAMANHO VARIÁVEL - modo de gravação empregado para dados cujo total de caracteres envolvidos na descrição do usuário NÃO PODE SER PREVIAMENTE AVALIADO.

VII - TAMANHO

O penúltimo campo do registro lógico é o de TAMANHO. Sua função é referenciar o total de caracteres que compõem o dado.

A representação na Tabela de Símbolos utiliza apenas 2 dígitos, porque, de acordo com as estimativas iniciais, nenhum dado tem comprimento superior a 99 caracteres. O cálculo feito liminarmente visa, em essência, a convencionar o LIMITE DO SISTEMA.

Nos casos de gravação em tamanho variável, o comprimento do dado é desconhecido "a priori", por isso sua representação no registro lógico é feita através de uma sequência de dois zeros (00) e o valor real é obtido através dos programas de processamento do 2º módulo.

VIII - ENDEREÇO DAS LISTAS

Ao considerar as possíveis inserções de dados feitas pelo usuário, na Tabela de Símbolos, verifica-se que o total de caracteres das listas não ultrapassa o valor 9999,

por isso a representação do campo de endereço nos registros lógicos abrange 4 dígitos apenas, constituindo assim o limite máximo convencionado.

A montagem do endereço das LISTAS feita nesse 1º módulo considera duas hipóteses:

. O DADO TEM ORGANIZAÇÃO DE LISTA

Neste caso, a geração é feita através de um conjunto de subprogramas, tal como se vê no Capítulo III deste volume.

. O DADO TEM ORGANIZAÇÃO INDEPENDENTE

Já aqui, como sua estrutura prescinde do LISTÃO da Tabela de Símbolos, é suficiente gravar a seqüência de caracteres "0000", que denota ausência de lista acoplada ao dado.

2.1.3. - CARACTERÍSTICAS DO LISTÃO

1 - CONCEITUAÇÃO

Como lista considera-se uma seqüência de dados de entrada, organizados num formato padrão e alocados na parte final da Tabela de Símbolos, com objetivo de minimizar os arquivos gerados, através de códigos que representam seus elementos.

2 - ORGANIZAÇÃO

As listas figuram no sistema na mesma ordem em que se apresentam os respectivos dados de entrada e possuem DEMARCADORES preestabelecidos, de modo a facilitar o percurso das mesmas.

São eles:

~~\$\$\$~~ início e fim de lista
/ separação dos elementos de uma lista
// fim de todas as listas.

O início e o fim de cada lista contêm 4 caracteres brancos, com objetivo de prover certa margem de segurança indispensável para possíveis FRAÇÕES DE ERRO NO PERCURSO DOS CADASTROS.

3 - CLASSIFICAÇÃO

Convencionou-se, para maior flexibilidade na movimentação dos dados, estabelecer dois tipos fundamentais de lista.

- . EXCLUSIVA
- . INCLUSIVA

LISTA EXCLUSIVA - é a lista composta de vários elementos, dos quais um e somente um representa a situação do dado cadastrado.

LISTA INCLUSIVA - é a lista composta por vários elementos, dos quais NENHUM, UM ou VÁRIOS podem representar a situação do dado cadastrado.

4 - DECODIFICAÇÃO

Convencionou-se utilizar 4 dígitos na representação de um dado com organização de "LI" ou "LE" depois de decodificado. Esse limite visa a atender possíveis mudanças no conteúdo das LISTAS.

2.1.4. - ESTRUTURAÇÃO DA ENTRADA DE DADOS E O EMPREGO DE "FORMAT" LIVRE

Como esse projeto atende às necessidades de uma clientela leiga em processamento de dados, o sistema apresenta um modo especial de introduzir as informações no equipamento de entrada. Trata-se do emprego de "FORMAT" LIVRE, que prescinde de medidas padrões no espaçamento dos campos gerados.

A entrada de dados informa ao computador características referentes a: nome do campo, tipo, organização, modo de gravação e tamanho do dado (Fig. 2.1.B).

NOME		TIPO	ORGANIZAÇÃO	MODO	TAMANHO
DO	*	DO	DO	DE	DO
CAMPO		DADO	DADO	GRAVAÇÃO	DADO

Fig. 2.1.B - ESBOÇO DE UM CARTÃO DE ENTRADA,
CONTENDO AS INFORMAÇÕES NECESSÁ
RIAS PARA A GERAÇÃO DE UM DADO
NA TABELA DE SÍMBOLOS.

Os exemplos que seguem ilustram várias entra
das mostrando uma classificação das diferentes possibilidades
lógicas de cada campo.

1 - QUANTO AO TIPO DE DADO ("A" ou "N")

ÓRGÃO.EXPEDIDOR*ALFA TAMANHO VARIÁVEL

MATRÍCULA*NUMÉRICO TAMANHO FIXO 06

2 - QUANTO A ORGANIZAÇÃO DO DADO ("LI" ou
"LE")

REGISTRO*NUMÉRICO TAMANHO FIXO 07

QUALIFICAÇÃO*NUMÉRICO LISTA INCLUSIVA TAMANHO VARIÁVEL

SITUAÇÃO*NUMÉRICO LISTA EXCLUSIVA TAMANHO FIXO 04

3 - QUANTO AO MODO DE GRAVAÇÃO ("TF" ou "TV")

CATEGORIA*NUMÉRICO LISTA EXCLUSIVA TAMANHO FIXO 04
 ENDEREÇO*ALFA TAMANHO VARIÁVEL

Para melhor compreensão é importante lembrar que nos casos de omitir-se a organização, trata-se de dado INDEPENDENTE cuja representação traduz-se por caracteres brancos. O mesmo ocorre com o tamanho do dado no caso de modo de gravação em dimensão VARIÁVEL.

2.1.5. - AUTOCONTROLE "VERSUS" INDEPENDENTIZAÇÃO DO USUÁRIO

Como se trata de um sistema AUTOCONTROLÁVEL, qualquer descuido na descrição dos dados, que revele não-observância às instruções do manual, ocasiona:

- . envio de MENSAGEM AO USUÁRIO , especificando o tipo de erro.
- . acionamento da CHAVE "KIERRO", para informar ao computador que o dado NÃO DEVE SER CADASTRADO.
- . prosseguimento AUTOMÁTICO do fluxo com a leitura do próximo dado.

Para efeitos elucidativos, apresenta-se, abaixo, pequena amostra de erros suscetíveis de ser cometidos pelo usuário.

- . TAMANHOS FIXOS ao invés de TAMANHO FIXO
- . ALFANUMÉRICO ao invés de ALFA
- . LISTA ESCLUSIVA ao invés de LISTA EXCLUSIVA
- . 2 ~~≠~~ ao invés de 23
- . ENDER,CO ao invés de ENDERECO

Incumbe recordar que espaços em branco no início, no meio ou no fim das especificações de entrada descritas em cartões são perfeitamente lícitos. Essas e outras instruções, estão detalhadas no ANEXO 1 deste volume (MANUAL DO USUÁRIO).

2.1.6. - FORMATO INTERNO DA TABELA DE SÍMBOLOS

Para que o leitor concatene uma idéia mais objetiva do que foi exposto apresenta-se abaixo, a Tabela de Símbolos correspondente a um CADASTRO-MIRIM, contendo, apenas, 6 dados: NOME, IDADE, TELEFONE, MATRÍCULA, ESTADO CIVIL e CATEGORIA.

```
NOME*00A00TV000000IDADE*00N00T
F020000TELEFONE*00N00TF070000M
ATRÍCULA*00N00TF060000ESTADO.C
IVIL*00NLETf040000CATEGORIA*00
NLETf040000ppp$SOLTEIRO/CASAD
```

0/VIUVO/DESQUITADO/~~VIUVO~~\$SUBSTITUTO/CONTRATADO/EFETIVO//

O esquema das entradas necessárias para sua geração é composto de:

1 - DESCRIÇÃO DOS DADOS:

NOME * ALFA TAMANHO VARIÁVEL
 IDADE * NUMÉRICO TAMANHO FIXO 02
 TELEFONE * NUMÉRICO TAMANHO FIXO 07
 MATRÍCULA * NUMÉRICO TAMANHO FIXO 06
 ESTADO.CIVIL * NUMÉRICO LISTA EXCLUSIVA TAMANHO FIXO 04
 CATEGORIA * NUMÉRICO LISTA EXCLUSIVA TAMANHO FIXO 04

2 - DESCRIÇÃO DAS LISTAS:

~~VIUVO~~\$SOLTEIRO/CASADO/VIUVO/DESQUITADO/~~VIUVO~~SUBSTITUTO/CONTRATADO/EFETIVO//

Impede ter em mira que, para efeito de transcrição neste manual, a Tabela de Símbolos supra-citada exhibe um tamanho de bloco igual a 30 caracteres. Os testes feitos no BURROUGHS/6700 apresentados no ANEXO 3 mantêm 60 caracteres, mas, na prática esse número pode ser alterado, conforme instruções contidas no item 1.3 deste volume.

2.2. - VALIDAÇÃO DE DADOS.

2.2.1 - DESCRIÇÃO GERAL.

Como foi visto no Capítulo 2, item 2.1, a Tabela de Símbolos reside em memória durante todo tempo de execução. Fato idêntico não ocorre no 2º módulo de validação, em que os dados são armazenados na memória de núcleos somente enquanto se processam os testes que vão estatuir:

- . a gravação dos mesmos no dispositivo de memória auxiliar previsto (fita magnética) ou,
- . seu egresso devido a discordância no confronto da descrição do usuário com os paradigmas do sistema.

O cadastramento de um dado só é possível quando este faz parte da Tabela de Símbolos. Dito fato pode corroborar a necessidade prevista no 1º módulo, de aceder grande flexibilidade e autonomia ao usuário no que tange à inserção ou retirada de elementos.

2.2.2. - ESTRUTURAÇÃO DA ENTRADA DE DADOS

A alimentação de um dado no equipamento de entrada efetiva-se através de um esquema geral que compreende a

explicitação do nome do campo e a descrição do dado no meio de entrada.

A utilização de linguagem corrente é mantida, no 2º módulo, podendo, ainda, o usuário valer-se de um ou mais cartões perfurados em sua descrição de dado.

O sistema convencionou para nomes compostos de mais de uma palavra a utilização do ponto (.) como elemento de ligação. É o que se vê, abaixo:

NOME FRIS.SOBEL.PETECK

2.2.3. - TIPOS DE ENTRADA.

- . "STANDARD"
- . "MÚLTIPLA"
- . "DEFAULT"

2.2.3.1. - ENTRADA "STANDARD" - A hipótese da entrada vista em 2.2.2., cuja utilização é a mais frequente no sistema. Ela compreende descrições que envolvem apenas UM CARTÃO DE DADOS e sua compleição descortina o nome do campo e a descrição do dado, tal como os padrões convencionados.

2.2.3.2. - ENTRADA MÚLTIPLA - Abrange casos das descrições que envolvem MAIS DE UM CARTÃO PERFURADO. Sua presença é reconhecida através de um código de identificação alocado na coluna 1 de cada cartão. A "input" neste caso, compreende:

- . código de identificação, nome do campo e o início da descrição do dado, nas 80 primeiras colunas
- . código de identificação e a continuação do dado em cada grupo de 80 colunas seguintes.

Geralmente é utilizado no caso de dados com ORGANIZAÇÃO DE "LI", cuja descrição envolve muitos elementos do LISTÃO.

2.2.3.3. - ENTRADA COM "DEFAULT" - Contempla as descrições que envolvem um cartão compreendendo APENAS O NOME DO CAMPO. Atende aos casos em que o usuário desconhece determinados dados. A OMISSÃO pode ser detectada pela presença de caracteres brancos em todas as colunas que seguem o nome do campo.

2.2.4. - DECODIFICAÇÃO DOS ELEMENTOS DAS LISTAS

A decodificação de um dado com organização de

LISTA adstringe-se a EXPLICITAR O Nº LÓGICO que equivale à posição do elemento no LISTÃO da Tabela de Símbolos. Utiliza um decimal composto de 4 DÍGITOS, que corresponde ao MAIOR TAMANHO de LISTA, previsto pelo sistema.

2.2.5. - TESSITURA DOS "NÓS" QUE COMPÕEM OS CADASTROS EM FITA MAGNÉTICA

Cada "NÓ" é composto de uma área de memória que contém:

- . TAMANHO DO DADO
- . DESCRIÇÃO DO MESMO

Assim sendo, para a entrada: "NOME LEIVAS.PORHCE", tem-se um esquema de "NÓ" referido por:

13	LEIVAS.PORHCE
----	---------------

2.2.6. - FORMATO INTERNO DOS CADASTROS.

A Fig. 2.2.6.A, abaixo, contém um esboço da ESTRUTURA GLOBAL DOS CADASTROS em fita magnética.

Através dela, pode-se observar:

- . a configuração final dos arquivos, composta de: Tabela de Símbolos + dados.
- . os blocos físicos de TAMANHO FIXO formados: por registros lógicos de TAMANHO VARIÁVEL.
- . o listão alocado na parte final da Tabela de Símbolos

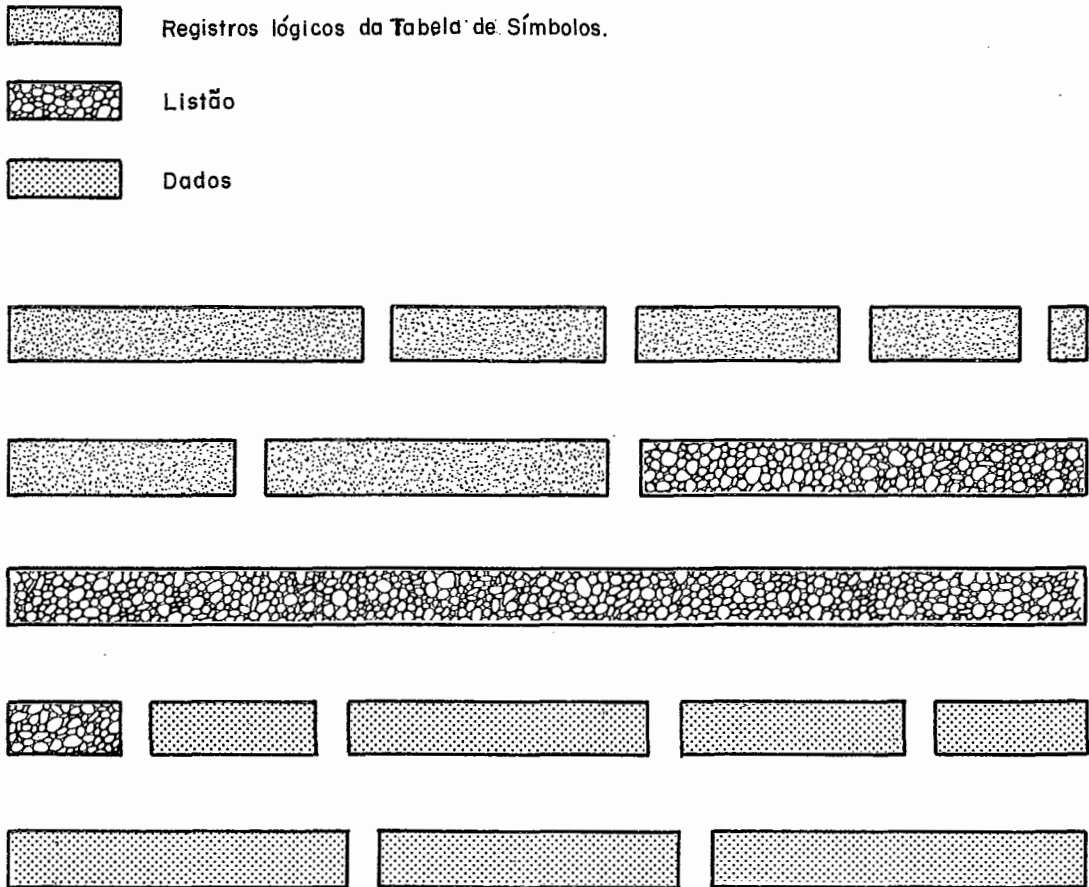


Fig. 2.2.6.A

2.2.7 - "WORKING-STORAGES" E O CADASTRAMENTO PROVISÓ- RIO

A longânime tarefa de DEPURAR DADOS, cadastran-
do-os somente em caso de DESCRIÇÃO VÁLIDA, requer a utiliza-
ção de uma área de trabalho à parte para organização PROVISÓ-
RIA do arquivo.

O principal escopo desta é atuar com as fun-
ções de um MINICADASTRO, onde a sequênci^a de caracteres cor-
respondente a cada "NÔ" é armazenada separadamente.

Ao fim do processamento de cada elemento, cujo
total de dados é equivalente ao previsto na Tabela de Símbo-
los, se nenhum deles vislumbra sintoma de erro, transferem-se
os blocos para o arquivo em fita magnética.

Em presença de erro na descrição do usuário, as
CHAVES DE ERRO são ACIONADAS, indicando que a transferência pa-
ra fita magnética não deve ser realizada e o fluxo continua
normalmente, com a leitura do próximo dado.

2.2.8. - ESTABELECIMENTO DE "DEFAULTS" DO SISTEMA.

A tarefa de suprir a ausência de dados em ca-
sos de descrições de entrada omissas, através de códigos pré-

fixados, constitui uma das funções do sistema.

O cadastro em fita não pode ser organizado de modo incompleto. Por isso, utiliza-se como meio auxiliar a gravação de valores previamente convencionados para DERRISCAR A AUSÊNCIA DE INFORMAÇÕES.

As normas explicitadas abaixo visam, sobretudo, a otimização do espaço disponível em memória auxiliar e constituem o embasamento adotado na criação dos "defaults" do sistema.

a) "DEFAULT" DE CÓDIGO "00" - traduz um dado de organização independente, que possui modo de gravação em tamanho variável. É aplicado tanto para dados numéricos como alfanuméricos. Sua estrutura no cadastro em fita é "0200", onde os 2 primeiros caracteres "02" compõem o tamanho do dado e os dois últimos "00" o "default" adotado.

b) "DEFAULT" DE CÓDIGO IGUAL AO NÚMERO DE ZEROS CORRESPONDENTE AO TAMANHO DO DADO - Representa dado de organização independente, que possui modo de gravação em tamanho fixo. Sua estrutura no cadastro em fita, para um dado cujo comprimento é referido por "07", tem a configuração de "070000000", onde "07" representa o tamanho do dado e "0000000" o "DEFAULT" implícito.

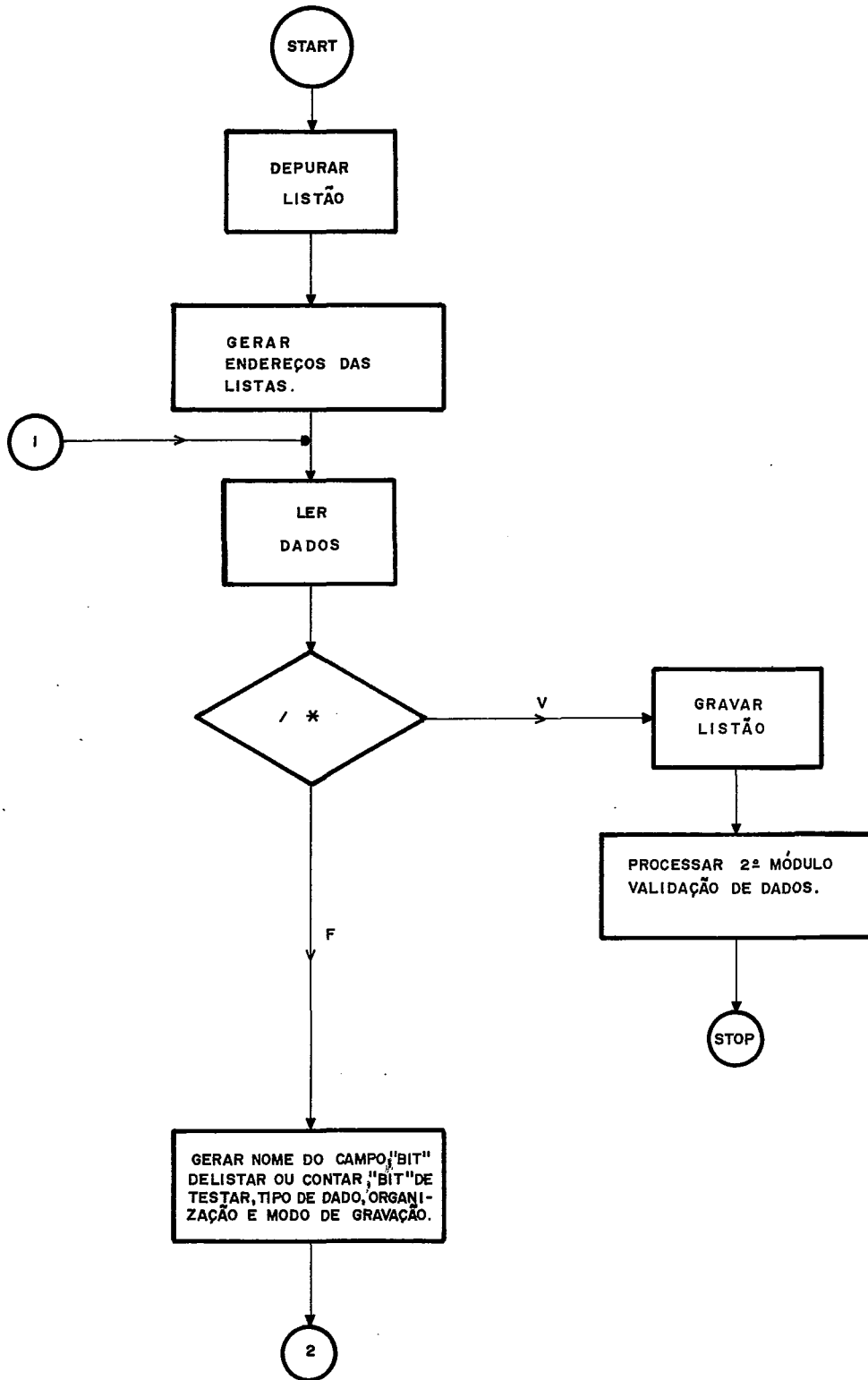
c) "DEFAULT" DE CÓDIGO "0000" - traduz um dado com organização de lista, cuja configuração no cadastro em fita é "040000". Os dois primeiros caracteres "04" refletem

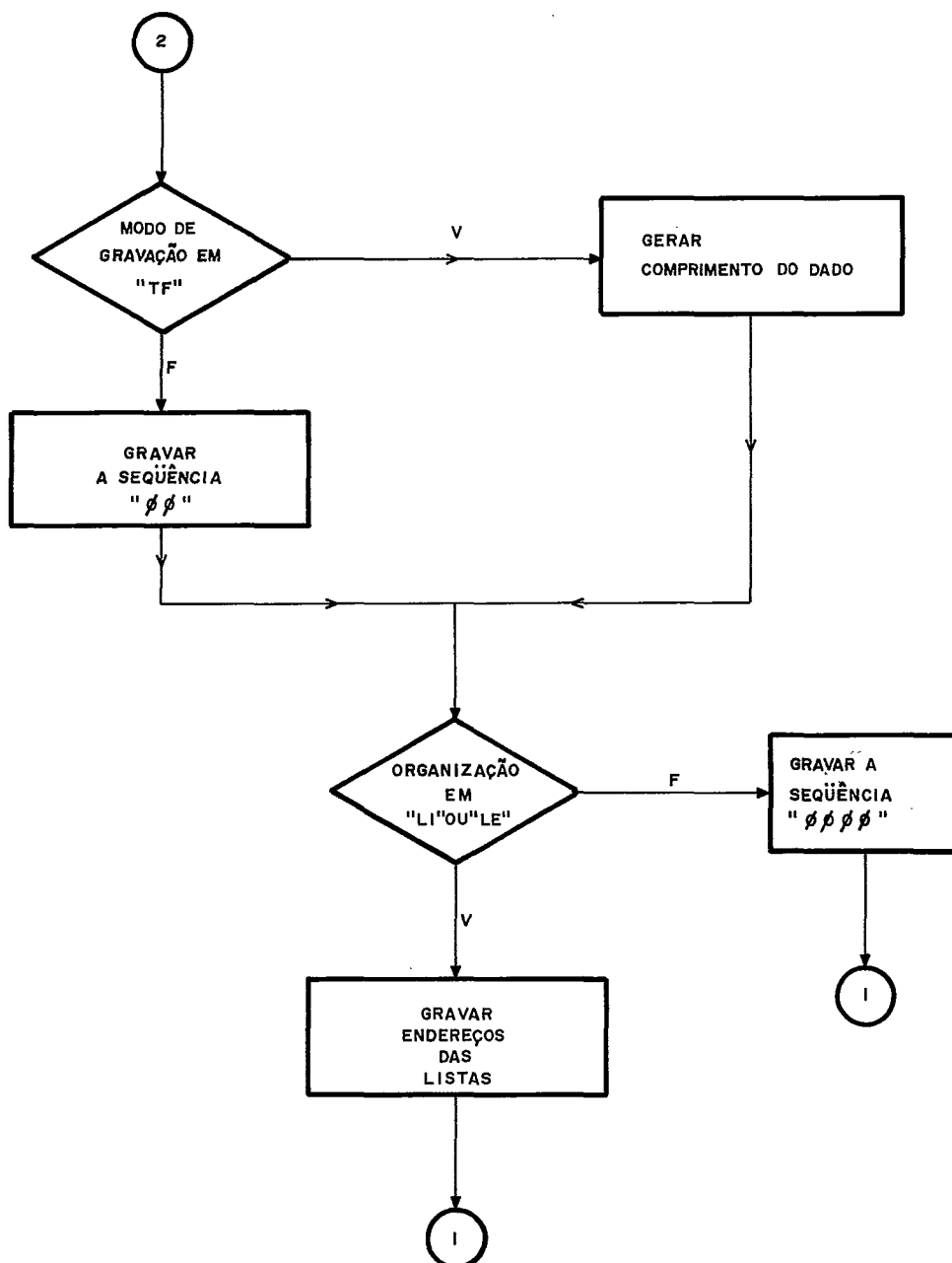
o tamanho do dado e, os quatro últimos "0000", o "default" adotado.

CAPÍTULO III

GERAÇÃO DA TABELA DE SÍMBOLOS (1º MÓDULO)

3.1. - GERAÇÃO EM BLOCOS





3.2. - DESCRIÇÃO GLOBAL DA ESTRUTURA DE DADOS

"I" - endereço dos cadastros em fita magnética.

"IAL(M)" - vetor do "DATA", que aloca a seqüência de caracteres da palavra "ALFA".

"IALFA(K)" - vetor do "DATA" que aloca caracteres numéricos , alfa e especiais.

"IBLOC" - comprimento do bloco físico.

"ICOMP(K)" - vetor auxiliar na alocação de dados validados.

"ICONDO" - variável que referencia a localização do elemento descrito pelo usuário no listão da Tabela de Símbolos.

"ICONT" - contador de blocos físicos.

"IENDCA" - endereço da primeira lista da Tabela de Símbolos.

"IFIM" - endereço do último caractere dos registros físicos de saída.

"IGERAL" - chave de erros.

"IMARC" - contador de caracteres que compõem o nome de dado.

"IMPRES(M)" - área de memória utilizada para armazenar dígitos numéricos.

"IMPRI(K)" - cadastro em fita magnética.

- "INIC" - endereço do 1º caractere dos registros físicos de saída.
- "INUM" - vetor do "DATA" que aloca a sequência de caracteres da palavra "NUMÉRICO".
- "IOG" - apontador do minicadastro auxiliar "LYAUX(M)".
- "IORG" - variável indicadora da organização do dado ("LI", "LE" ou independente).
- "IPEG" - chave de controle na gravação dos blocos físicos.
- "ITOT" - apontador auxiliar no percurso das listas da Tabela de Símbolos.
- "IVET(K)" - vetor auxiliar na alimentação dos dados de entrada.
- "JUCK" - apontador utilizado para percorrer o cadastro em fita magnética.
- "K" - apontador auxiliar no percurso do arquivo de entrada.
- "KARTY" - contador de caracteres dos dados.
- "KEMBY" - variável auxiliar na execução de "KONVTA" e "KALFTA", utilizada para guardar o tamanho do dado.
- "KFINDO" - contador de delimitadores encontrados na descrição de dados com organização de lista.

"KFITA(M)" - vetor do "DATA" que aloca a seqüência de caracteres da expressão "TAMANHO FIXO".

"KIERRO" - chave de erros.

"KIMERA" - chave de erros.

"KLI" - apontador utilizado no percurso do arquivo "NIDADO(K)" de entrada.

"KNIC" - marcador da posição inicial de cada elemento de entrada, descrito com organização de "LI".

"KOFIM" - delimitador de fim de dado.

"KOMEÇO" - delimitador de início de dado.

"KVATA(M)" - vetor do "DATA" que aloca a seqüência de caracteres da expressão "TAMANHO VARIÁVEL".

"LEVA" - total de caracteres que compõem o dado descrito pelo usuário.

"LISEX(M)" - vetor do "DATA" que aloca a seqüência de caracteres da expressão "LISTA EXCLUSIVA".

"LISIN(M)" - vetor do "DATA" que aloca a seqüência de caracteres da expressão "LISTA INCLUSIVA"

"LYAUX(M)" - minicadastro que encadeia a seqüência de caracteres a ser gravada em fita magnética.

"MIGRA" - variável indicadora do modo de gravação do dado ("TF" ou "TV").

"MONTE(K)" - vetor que armazena a fileira de caracteres que compõem as listas.

"NA" - variável indicadora do tipo de dado ("N" ou "A").

"NIDADO(K)" - vetor de entrada que contém o nome do dado e o próprio dado, tal como descritos pelo usuário.

"NUMALF(K)" - vetor do "DATA" que aloca os dígitos alfanuméricos.

"NUMERO(N)" - vetor do "DATA" que aloca os dígitos numéricos.

3.3. - FASES DA GERAÇÃO

3.3.1. - "IERRO(N)" - IMPRESSÃO DE MENSAGENS DE ERRO.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"KART" - posição inicial do enésimo erro.

"ITER" - posição final do enésimo erro.

"INTERG" - contador de frases percorridas.

"IFRAS(IFLY)" - vetor do "DATA" que aloca as MENSAGENS em linguagem corrente.

"KARLA" - apontador utilizado no percurso de "IFRAS(IFLY)".

"LETUR" - delimitador.

"N" - parâmetro que indica o número do item a ser impresso.

II - OBJETIVO.

. enviar mensagem ao usuário, no caso de descrição inválida, especificando o NÚMERO E TIPO DE ERRO.

III - DESCRIÇÃO GERAL.

"SUBROUTINE" caracterizada por um único parâmetro "N", cujo objetivo é referenciar o número do erro solicitado. A cada comando do tipo "CALL IERRO(N)", associa-se o pedido de impressão da enésima mensagem ao usuário.

Todas as frases são geradas pelo sistema em um grande comando "DATA", através das diferentes posições do vetor "IFRAS(IFLY)", que é lido em "FORMAT A4" e tem como delimitador a variável "LETUR" representada por "~~xxx~~".

IV - ESQUEMA DE "PROCEDURES".

Inicialmente, fixa-se a primeira coluna do vetor do "DATA" através do comando "KARLA = KARLA + 1" e admite-se que o erro "N" a ser impresso inicie na primeira posição de "IFRAS(IFLY)". Em seguida, atribui-se ao contador "INTERG"

seu valor inicial, fazendo: "INTERG = 0".

O rumo posterior consiste em percorrer "IFRAS (IFLY)", através de seu apontador "KARLA", até encontrar o delimitador "LETUR". Ao interceptá-lo, incrementa-se automaticamente "INTERG", indicando que mais um erro foi atingido.

Neste ponto, testa-se o valor de "INTERG" que, agora, contém a indicação do último erro alcançado, verificando se coincide com o número "N" solicitado.

Havendo dissidênciã, atualiza-se o apontador "KART", assinalando a posição inicial da próxima frase a ser percorrida, fazendo "KART = KARLA + 1" e procede-se como anteriormente.

Ao atingir o erro que se pretende (caso de "INTERG = N"), decrementa-se uma unidade no apontador "KARLA" para obter a localização final da mensagem, fazendo "ITER = KARLA - 1". Isto, porque, ao encontrar o erro pretendido, o apontador "KARLA" está assinalando o delimitador "LETUR".

Após a determinação das posições inicial e final da mensagem desejada, o erro solicitado é impresso e o comando das ações devolvido ao programa de origem.

3.3.2. - "INDTAM" - DETERMINAÇÃO DO ENDEREÇO DA PRIMEIRA LISTA DA TABELA DE SÍMBOLOS.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"KEY" - chave para controle de erros.

"MAGNA" - comprimento da gravação em fita magnética.

"ISCO" - acumulador de registro lógicos.

"LETTER" - contador de caracteres descritos em cada nome de dado.

"ICART(M)" - vetor que armazena a descrição do usuário.

II - OBJETIVO.

. Avaliar o total de caracteres que antecedem às LISTAS, no arquivo em fita magnética.

III - DESCRIÇÃO GERAL.

"INDTAM" é o primeiro subprograma executado pelo sistema. Sua função é galvanizar a contagem dos caracteres gerados na formação da Tabela de Símbolos desde a primeira posição por ela ocupada até o endereço de memória que antecede ao listão. Destaca-se por sua fundamental importância no cálculo posterior de endereços de todas as listas.

Sua filosofia baseia-se na própria estrutura do registro lógico, cujo espaço físico ocupado em fita envolve duas considerações:

1 - O total de caracteres utilizado para descrever o

nome do dado é representado por uma grandeza VARIÁVEL.

2 - O total de caracteres utilizados para representar delimitador, "bit" de listar ou contar, "bit" de testar, tipo de dado, organização, modo de gravação, tamanho e endereço da lista representa-se por uma constante, cujo valor é, por definição, igual a 14. (Fig. 3.3.2.A)

NOME				N	LI	TF		
DE	*	Ø	Ø		LE		ØØ	ØØØØ
DADO				A	ØØ	TV		

TOTAL DE

CARACTERES

VARIÁVEL

TOTAL DE CARACTERES

FIXO (K = 14)

Fig. 3.3.2-A - AVALIAÇÃO DO COMPRIMENTO DE UM REGISTRO LÓGICO DA TABELA DE SÍMBOLOS

IV - ESQUEMA DE "PROCEDURES".

Pelo exposto acima, depreende-se facilmente o descarte utilizado por "INDTAM", que pode ser resumido em 3 itens:

1 - Criação de um acumulador, representado pela variável "ISCO", com objetivo de explicitar o total de registros lógicos da Tabela de Símbolos, através da contagem do número de dados a serem gerados em cada cadastro.

2 - Utilização de uma variável aleatória "MAGNA", para informar ao sistema quantos são os caracteres necessários para a gravação de todos os nomes de dados.

3 - Cálculo do endereço da primeira lista da Tabela de Símbolos, através da adição do total de caracteres da parte fixa dos registros lógicos ("ISCO*14"), com o da parte variável representado por "MAGNA". O referido endereço é armazenado na variável "IENDCA", através de um comando único, tal como se vê abaixo:

"IENDCA = MAGNA + ISCO * 14".

3.3.3. - "LELIST" - GERAÇÃO DO LISTÃO DA TABELA DE SÍMBOLOS.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"LISTA(M)" - vetor auxiliar na leitura das listas.

"KPRI" - chave para controle de chamadas a "LELIST".

II - OBJETIVOS.

- . depurar as listas que vão ser acopladas ao sistema.
- . enviar mensagens de erro ao usuário, através de chamadas a "IERRO(N)".

III - DESCRIÇÃO GERAL.

O controle de chamadas a "LELIST" é feito pela chave "KPRI", que coordena:

- . a leitura e validação das listas, na primeira execução
- . a gravação das mesmas, através de transferência do fluxo para "MLTIMP(N)", na segunda execução.

A cada dado com organização de "LI" ou "LE" alocado na Tabela de Símbolos corresponde uma lista que é obtida do equipamento de entrada.

A inserção ou retirada de dados nos cadastros em fita supõe uma análise prévia da organização dos mesmos, o que pode ocasionar:

- . adjeção ou egresso da lista alusiva ao campo em movimento, no caso de "LI" ou "LE"
- . nenhuma alteração nas listas da Tabela de

Símbolos, na hipótese de dado independente.

IV - ESQUEMA DE "PROCEDURES".

A filosofia de "LELIST" baseia-se na análise individual dos caracteres integrantes das listas.

Em caso de elementos estranhos, verifica-se a chamada de "IERRO(N)", que envia mensagem ao usuário, correspondente ao ERRO 1.

Ao final, o comando das ações é transferido para "MLTIMP(N)", com o objetivo de gravar as listas nos arquivos em fita magnética.

3.3.4. - "IENDER" - GERAÇÃO DE ENDEREÇOS DAS LISTAS.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"KAM" - variável utilizada como contador.

"IENDLI(KAM)" - área de memória reservada para alocar endereços.

II - OBJETIVO.

. acessar as listas da Tabela de Símbolos.

III - DESCRIÇÃO GERAL.

"IENDER" é chamada pelo programa principal logo após a execução de "LELIST".

Obtém a localização da enésima lista somando o endereço da primeira, anteriormente calculado por "INDTAM" , com o total de caracteres gravados até a anterior.

IV - ESQUEMA DE "PROCEDURES".

O roteiro de ações de "IENDER" comporta resumo nas seguintes etapas:

- . posicionar devidamente "ITOT" e "KAM", de modo que representem respectivamente o primeiro caractere das listas e o início da contagem das mesmas

- . percorrer a área de memória organizada por "LELIST", até encontrar as seqüências de caracteres "~~xxx~~", que assinalam fim de uma lista e início da próxima.

- . gerar endereços somando o conteúdo de "IENDCA" (ver 3.3.2) com o número que representa o total de caracteres utilizados na gravação de todas as listas até a precedente

- . atualizar o contador representado pela variável "KAM"

- . efetuar teste para detectar a presença da

seqüência de duas barras (//), indicativa de final do listão, o que ocasiona o retorno do comando ao programa principal.

3.3.5. - "ISAI(L)" - GRAVAÇÃO DE CARACTERES INDIVIDUAIS EM FITA MAGNÉTICA

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

(VER DESCRIÇÃO GLOBAL - item 3.2)

II - OBJETIVO.

. organização dos arquivos gerados.

III - DESCRIÇÃO GERAL.

O intento de "ISAI(L)" é armazenar caracteres individuais no vetor "IMPRI(K)" de gravação em fita, de modo a possibilitar a geração paulatina de elementos isolados nos cadastros.

"ISAI(L)" considera um tamanho de bloco VARIÁVEL, de sorte a oferecer maior flexibilidade ao sistema. A definição deste comprimento faz-se logo no início do programa principal, através de um comando aritmético que atribui à variável "IBLOC" o valor desejado.

Em todos os testes feitos no "BURROUGHS/6700", para geração dos cadastros contidos no ANEXO 3 consideram-se blocos de 60 caracteres, ou seja: "IBLOC = 60".

IV - ESQUEMA DE "PROCEDURES".

O roteiro de ações de "ISAI(L)" pode ser resumido no seguinte esquema:

1 - Zerar o CONTADOR DE BLOCOS FÍSICOS, fazendo, "ICONT" = 0.

2 - ATUALIZAR o valor de "I", de modo que ele passe a representar o PRÓXIMO ENDEREÇO A SER OCUPADO nos cadastros em fita, através do comando "I = I + 1".

Duas situações podem sobrevir:

3 - ENDEREÇO SITUADO É O ÚLTIMO DO BLOCO FÍSICO

Delimitar, então, as posições de início e fim do bloco e proceder à gravação do mesmo no cadastro em fita.

Tratando-se de último bloco físico, retornar o comando ao programa principal e, em caso adverso, atualizar o contador de blocos físicos, agindo tal como no caso abaixo:

4 - ENDEREÇO SITUADO NÃO É O ÚLTIMO DO BLOCO FÍSICO

Testar se a gravação finda sem completar o último bloco. Neste caso, delimitar as respectivas posições de início e fim e proceder como anteriormente.

Caso contrário, armazenar o caractere e retornar o

comando ao programa que chamou "ISAI(L)".

3.3.6. - "MLTIMP(N)" - GRAVAÇÃO EM FITA ADAPTADA PARA
QUANTIDADES MAIORES DE DADOS.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"KEPE" - apontador auxiliar no percurso do vetor "MONTE(K)".

"N" - comprimento do dado a ser cadastrado.

II - OBJETIVOS.

GRAVAR em fita magnética uma SEQUÊNCIA DE CARACTERES, cujo total de elementos é referenciado pelo PARÂMETRO "N".

III - DESCRIÇÃO GERAL.

O intento de "MLTIMP(N)" é a gravação de "N" caracteres em fita magnética, através do DESLOCAMENTO SIMULTÂNEO DOS APONTADORES "I" E "KEPE", que posicionam respectivamente:

- . o próximo endereço da fita a ser alocado
- . a posição atual do vetor a ser gravado.

Tal como "ISAI(L)", esta "SUBROUTINE" também considera um tamanho de bloco VARIÁVEL, que é definido no programa principal através da variável "IBLOC". Em todos os testes feitos no "BURROUGHS/6700, anexos a este Tese, convencionou-se o valor "IBLOC = 60".

IV - ESQUEMA DE PROCEDURES".

O roteiro de ações de "MLTIMP(N)" pode ser sintetizado no seguinte esquema:

I - Zerar o contador de blocos físicos e situar os dois apontadores

"I" - no próximo endereço da fita a ser ocupado

"KEPE" - na primeira posição do vetor que contém a seqüência de caracteres a ser gravada

Duas hipóteses podem então instalar-se:

II - ENDEREÇO SITUADO NA FITA CORRESPONDE AO ÚLTIMO DO BLOCO FÍSICO

Delimitar as posições inicial e final do bloco e proceder à gravação do mesmo.

No caso de ÚLTIMO BLOCO FÍSICO, retornar o comando ao programa principal. Caso contrário, atualizar o contador de blocos físicos e agir tal como no caso de:

III - ENDEREÇO SITUADO NA FITA NÃO CORRESPONDE AO ÚLTIMO BLOCO FÍSICO

Testar se a gravação finda sem completar o último bloco físico. Neste caso, delimitar as posições inicial e final do bloco e proceder como anteriormente.

Caso contrário, atualizar, o apontador "KEPE", do vetor a ser gravado e alocar o respectivo caractere no endereço do cadastro REFERENCIADO POR "I".

IV - Repetir o mesmo esquema até o fim do cadastramento através de atualização dos apontadores "I" e "KEPE" e, ao final, retornar ao programa de origem.

3.3.7. - "KONVER" - DESCOMPACTAÇÃO DOS ENDEREÇOS DAS LISTAS.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"KERO" - variável que armazena o conteúdo dos endereços.

II - OBJETIVO:

. separar dígitos componentes da descrição de números decimais.

III - DESCRIÇÃO GERAL.

Ao solicitar a "SUBROUTINE KONVER" o sistema

dispõe de uma área de memória representada pela variável "KERO", cujo conteúdo é o número que se pretende descompactar.

IV - ESQUEMA DE "PROCEDURES".

Os procedimentos inerentes a este subprograma têm como alicerce a análise da posição relativa dos dígitos integrantes dos endereços das listas. A explicitação destes, através de comandos aritméticos utiliza princípios de numeração decimal e abrange como fase complementar a organização do vetor "KONT(K)" que reúne os 4 dígitos do endereço descompactado.

3.3.8. - "KONALF" - CONVERSÃO DE NUMÉRICOS EM ALFANUMÉRICOS.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

(VER DESCRIÇÃO GLOBAL - ITEM 3.2).

II - OBJETIVO:

. Transformar, em alfanuméricos, os endereços das listas geradas na Tabela de Símbolos.

III - DESCRIÇÃO GERAL.

Os vetores "NUMERO(N)" e "NUMALF(N)" são inicialmente armazenados em área de "COMMON".

Os quatro caracteres formadores dos endereços das listas, residentes no vetor "KONT(N)", são comparados com os elementos de "NUMERO(N)" e os dígitos alfanuméricos correspondentes alocados no vetor "MONTE(M)", de gravação em fita.

IV - ESQUEMA DE "PROCEDURES".

As conversões em alfa efetivam-se na medida em que os endereços são gerados, pois cada chamada à "KONALF" implica a mudança de um endereço apenas: aquele que estiver no vetor "KONT(M)".

Os testes para verificar a composição dos endereços concretizam-se através de duas iterações:

- . a primeira, que controla os dígitos dos endereços, utiliza o parâmetro "M", variando de 1 até 4;
- . a segunda, que controla as posições dos vetores "NUMERO(N)" e "NUMALF(N)", emprega o parâmetro "N", variando de 0 até 9.

A iteração mais interna é a segunda e seu percurso realiza-se para cada valor de "M" no 1º laço. Após a determinação dos dígitos do endereço, procede-se à alocação

dos correspondentes em alfanuméricos, no vetor "MONTE(M)", o que possibilita a chamada de "MLTIMP(N)" para gravação em fita magnética.

3.3.9. - PROGRAMAS DE PERCURSO E ANÁLISE DO ARQUIVO DE ENTRADA.

3.3.9.1 - "NAMEL" - GERAÇÃO DE NOME DO DADO NA TABELA DE SÍMBOLOS.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"IBRAN" - chave para controle de cartões em branco inseridos na massa de dados.

"ISUPR" - variável auxiliar na supressão de espaços em branco acoplados à descrição do campo.

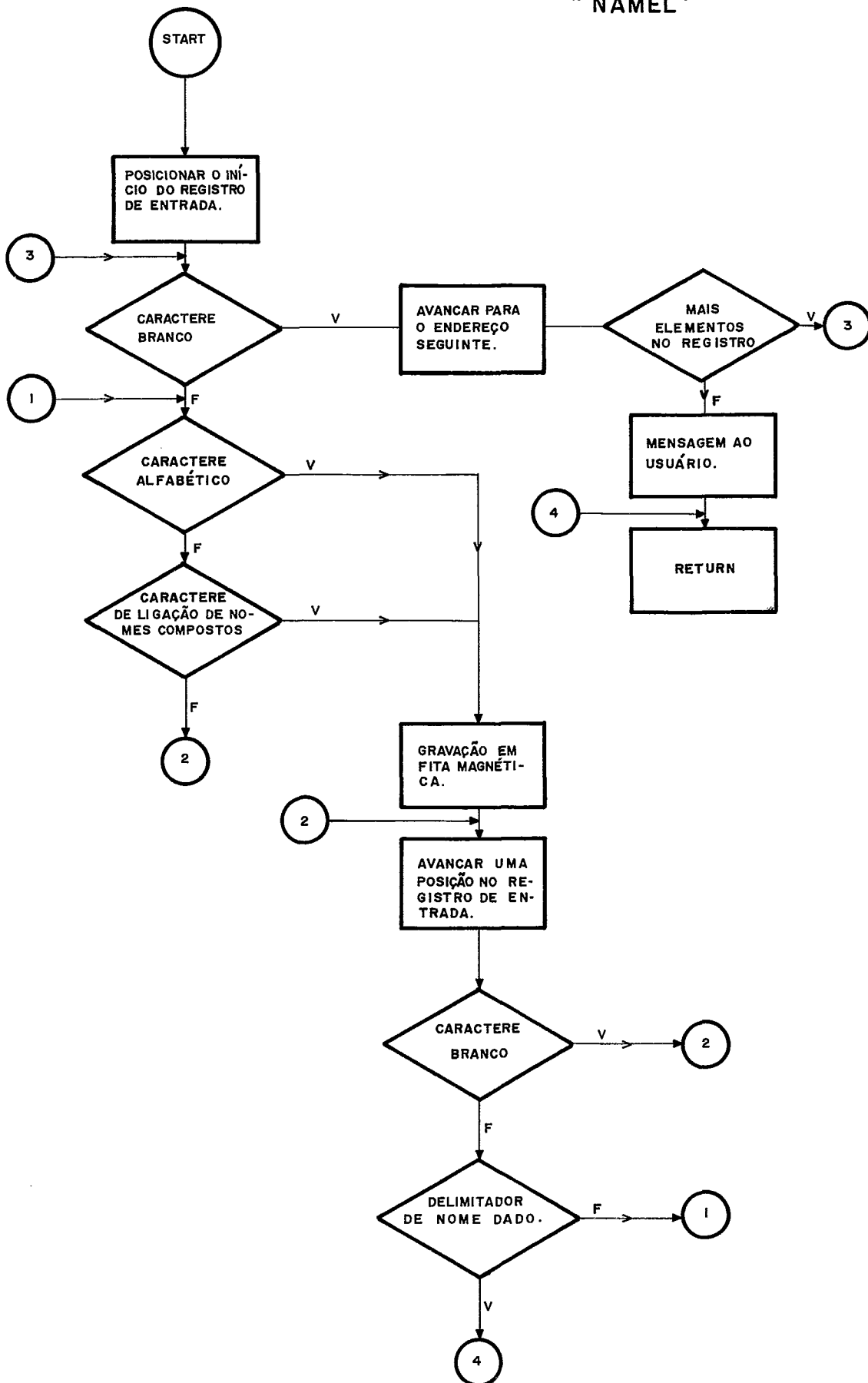
"ICHAV" - chave de controle na fase de validação.

"MARCF" - variável que contém referência ao final da gravação em fita magnética.

II - OBJETIVO:

. validar o nome do campo e proceder a geração do mesmo na Tabela de Símbolos.

" NAMEL "



. enviar mensagens de erro ao usuário, através de chamadas a "IERRO(N)".

. informar ao sistema que após o egresso dos nomes INVÁLIDOS, o fluxo prossegue automaticamente.

III - DESCRIÇÃO GERAL.

A fase de VALIDAÇÃO encontrada neste subprograma abrange uma análise minuciosa de cada caractere que compõe a identificação de NOME DO CAMPO, registrada pelo usuário.

A variável "IMARC", que atua como contador de caracteres dos NOMES DE DADOS constitui um meio auxiliar na atualização dos endereços em fita, nos casos de entrada inválida.

"NAMEL" tem, também, a incumbência de atender aos casos em que se observa a presença de cartões em branco, envolvidos no "deck" do usuário.

IV - ESQUEMA DE "PROCEDURES".

O modo de proceder da "SUBROUTINE NAMEL" é extremamente simples e, pode numa sistematização objetiva ser resumido no seguinte roteiro:

. o apontador "K", desloca-se ao longo do vetor "NIDADO(K)" de entrada até encontrar o primeiro caractere não branco

. o campo é submetido a uma série de testes que vão corroborar sua validade

. a presença de cartões em branco na massa é analisada e o sistema passa a ignorá-los

. o apontador "I" retorna tantas posições quantos são os endereços já ocupados por campos com diagnóstico estranho, utilizando: "I = I - IMARC" e a mensagem de erro correspondente é enviada ao usuário.

. os caracteres válidos são incorporados ao vetor "IMPRI(K)" de gravação em fita, através de "ISAI(L)".

3.3.9.2 - "KTIPO" - GERAÇÃO DE TIPO DE DADO.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

(VER DESCRIÇÃO GLOBAL - ITEM 3.2)

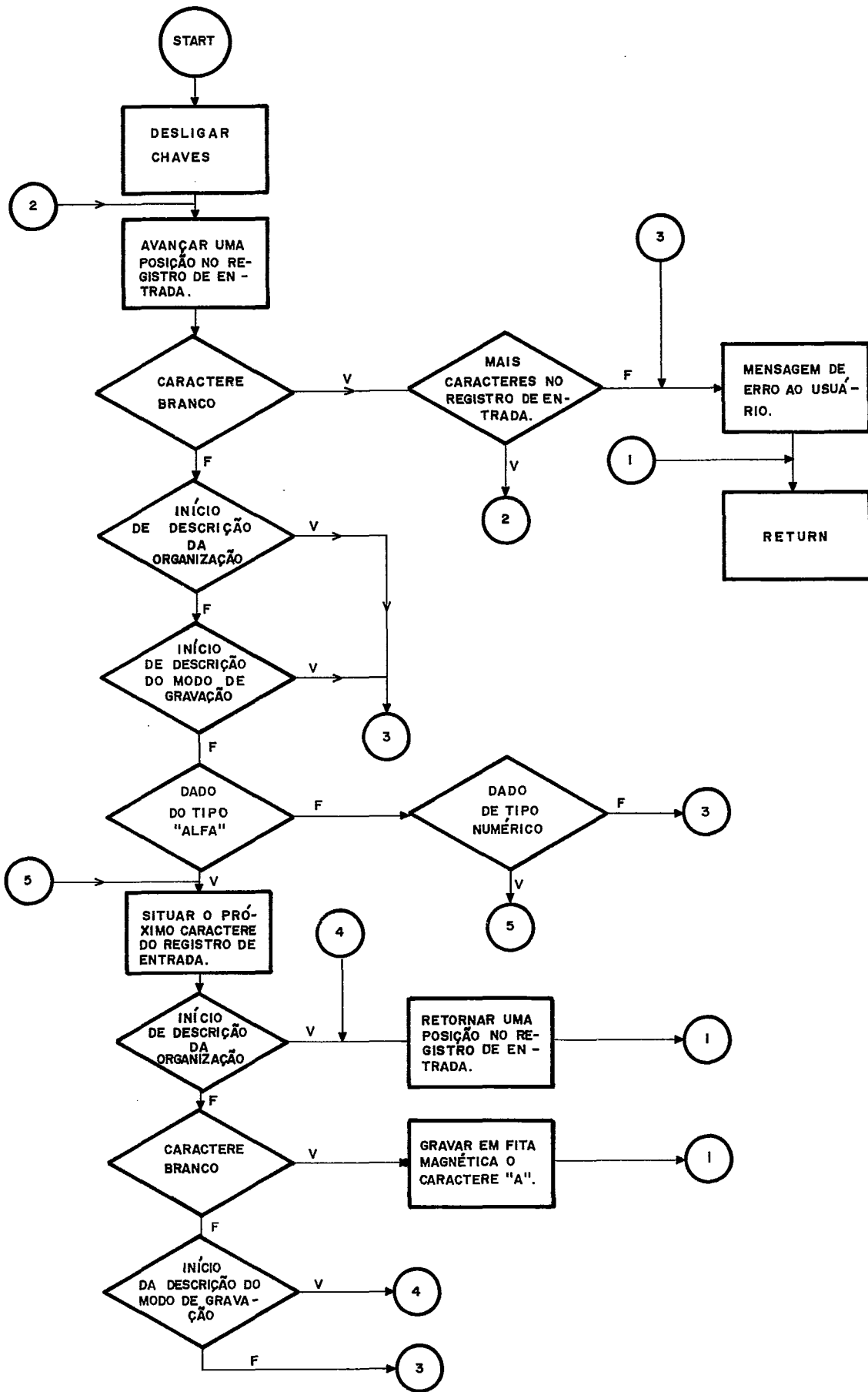
II - OBJETIVOS:

. validar o tipo de dado e proceder a geração do mesmo na Tabela de Símbolos.

. enviar MENSAGENS DE ERRO ao usuário, através de chamadas a "IERRO(N)"

. informar ao sistema que após o egresso das descrições INVÁLIDAS o fluxo prossegue automaticamente.

"KTIPO"



III - DESCRIÇÃO GERAL.

"KTIPO" é solicitada pelo sistema logo após a execução de "NAMEL", exatamente no instante em que surge, no registro de entrada, o asterisco, delimitador de nome do dado.

Sua tarefa consiste em continuar, através de deslocamento do apontador "K", o percurso da descrição do usuário ao longo de "IVET(K)", de modo a obter um espectro da descrição de tipo do dado.

Em caso de descrição inválida a chave "KIERRO" é ligada para indicar ao sistema que o dado não deve ser cadastrado. Há, então, um retorno no endereço "I" dos arquivos em fita, que promove a supressão da parte já gravada. A operação de volta abrange as posições correspondentes ao número de caracteres de NOME DO DADO, referenciado por "IMARC", mais as três que envolvem: asterisco, "BIT" de listar ou contar e "BIT" de testar.

O comando de retorno assume, portanto, a configuração:

"I = I - (IMARC + 3)"

IV - ESQUEMA DE "PROCEDURES".

Posicionar em "IVET(K)" o endereço seguinte ao asterisco e testar se o conteúdo deste é representado por caractere branco. Em caso afirmativo avançar mais uma posição e repetir o procedimento sucessivas vezes.

Ao vislumbrar o primeiro elemento dissidente verificar a possível omissão de TIPO do dado, testando se aquele denota:

- . início da ORGANIZAÇÃO
- . início do MODO DE GRAVAÇÃO

Qualquer das duas situações caracteriza a presença do ERRO 5.

Em caso adverso desviar o fluxo do sistema para a iteração que vai detectar se a descrição é de campo ALFA. A confirmação deste desencadêia dois passos indispensáveis:

- . gravação em fita magnética do "A" indicativo de tipo ALFA, através de chamada a "ISAI(L)"
- . avanço do apontador "K" para teste do endereço seguinte.

Quatro casos podem, então, eclodir:

- 1 - ENDEREÇO SEGUINTE CONTÉM INÍCIO DA ORGANIZAÇÃO DO DADO.

Ocorre sempre que não há colunas em branco, separando a descrição de tipo e a organização do dado. A função do sistema neste caso é retornar à antiga posição, decrementando uma unidade no valor de "K" e passar o comando das ações para "ICOGI", que se encarrega de gerar a ORGANIZAÇÃO do dado.

2 - ENDEREÇO SEGUINTE CONTÉM UM CARACTERE BRANCO.

Caracteriza situações em que há pelo menos uma coluna em branco depois da especificação de tipo ALFA. O procedimento é semelhante ao anterior onde o comando é devolvido ao programa principal, que se encarrega de chamar "ICOGI".

3 - ENDEREÇO SEGUINTE CONTÉM INÍCIO DO MODO DE GRAVAÇÃO.

Surge em casos de dados independentes onde o modo de gravação situa-se imediatamente após a descrição de tipo. Ocasiona transferência das ações para "LGRILL".

4 - ENDEREÇO SEGUINTE CONTÉM UM CARACTERE NÃO ESPECIFICADO NAS 3 HIPÓTESES ACIMA

Este caso corresponde ao ERRO 6 descrito no GLOSSÁRIO final e se a coincidência de caracteres não revelar a presença de dado ALFA, testa-se a "CHAVE DE CONFRONTO" para verificar se houve interferência de elementos estranhos, fato que caracteriza o ERRO 7.

Com CHAVE DE CONFRONTO DESLIGADA verifica-se , por extrapolação, se o campo é NUMÉRICO.

Ao fim da execução de "KTIPO", se a pesquisa revelar que o dado não é ALFA nem NUMÉRICO torna-se patente a presença do ERRO 10.

3.3.9.3. - "ICOGI" - GERAÇÃO DA ORGANIZAÇÃO DO
DADO

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"KDUCCO" - variável auxiliar na indicação de "LI", "LE" ou dado independente.

"KING" - apontador auxiliar no percurso de "LISIN(M)" e "LISEX(M)".

II - OBJETIVOS:

. validar a organização do dado de modo a gerar sua descrição na Tabela de Símbolos

. enviar MENSAGENS DE ERRO ao usuário, através de chamadas a "IERRO(N)"

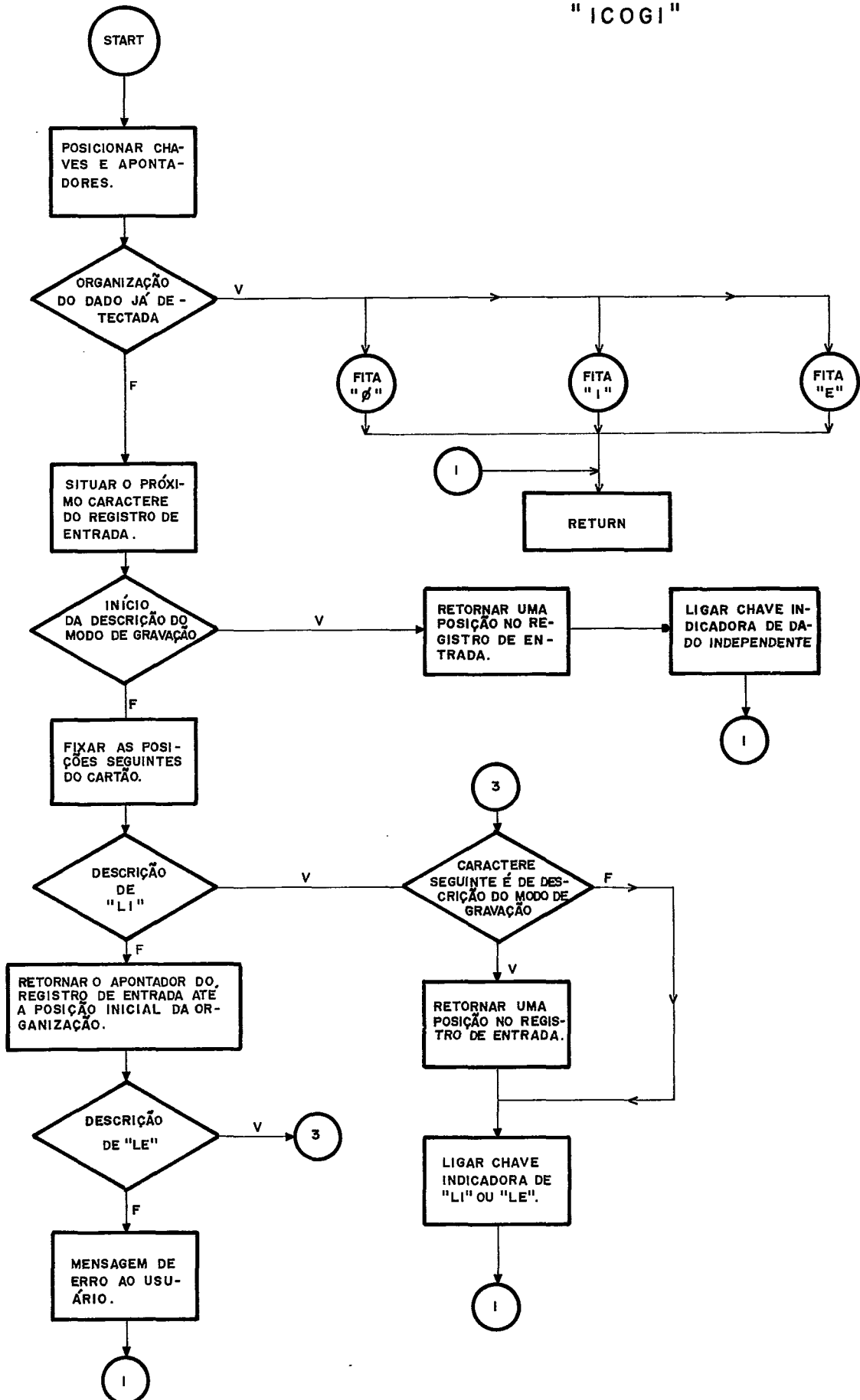
. informar ao sistema que após egresso das descrições INVÁLIDAS o fluxo prosegue automaticamente

III - DESCRIÇÃO GERAL.

"ICOGI" é requisitada pelo programa principal duas vezes para cada registro de entrada.

A organização do dado é alocada em fita magnética através de 2 caracteres: "LI", "LE" ou "∅∅".

"ICOGI"



A gravação do primeiro caractere ("L", "L" ou "Ø") é feita logo no início do subprograma, ficando ao encargo da segunda execução o cadastramento de ("I", "E" ou "Ø").

As atualizações nos arquivos de saída em caso de dados INVÁLIDOS utilizam o comando:

"I = I - (IMARC + 4)"

que ocasiona um retorno no endereço da fita correspondente ao total de caracteres de nome do dado ("IMARC"), mais as 4 posições referentes a: delimitador, "BIT" de listar ou contar, "BIT" de testar e tipo de dado.

IV - ESQUEMA DE "PROCEDURES".

O objetivo da recorrência a "ICOGI" é detectado através da chave "KDUCO", uma vez que:

. "KDUCO ≠ 0" (CHAVE LIGADA) - caracteriza a segunda chamada a "ICOGI" e conduz o fluxo para a gravação do segundo caractere ("I", "E" ou "Ø") da organização do dado, através de "ISAI(L)", conforme a posição referenciada:

"KDUCO = 1" - DADO INDEPENDENTE

"KDUCO = 2" - LISTA INCLUSIVA

"KDUCO = 3" - LISTA EXCLUSIVA

. "KDUCO = Ø" (CHAVE DESLIGADA) - referencia a primeira execução de "ICOGI", que vai percorrer a massa de dados codificados para informar ao sistema qual o caso consi-

derado.

Após a execução de "KTIPO" o apontador "K" desloca-se ao longo do arquivo de entrada e ao encontrar o primeiro caractere não branco verifica se este corresponde ao início do modo de gravação.

Confirmada a hipótese decrementa uma unidade no valor de "K", de modo a preparar a execução de "LGRILL" , que cuida do próximo campo referente a modo de gravação. Isto, porque, nos registros alimentados pelo usuário o branco representa ORGANIZAÇÃO INDEPENDENTE e, neste caso, implica em:

. recorrência a "ISAI(L)" para gravação do primeiro zero da designação através do comando:

```
"CALL ISAI(IALFA(29))"
```

. acionamento da chave de controle que passa a assinalar: "KDUCO = 1".

A presença de "LI" ou "LE" é detectada através de acareação dos vetores "LISIN(M)" e LISEX(M)" que contêm as seqüências alocadas no "DATA" com valores respectivamente iguais a : LISTA INCLUSIVA e LISTA EXCLUSIVA.

Ambos os casos desencadeiam:

. gravação do "L", através do comando

```
"CALL ISAI(IALFA(12))"
```

. acionamento da chave de controle isto é:

"KDUCO = 2" para "LI" e "KDUCO = 3" para "LE" .

. teste do caractere descrito imediatamente após a organização.

"ICOGI" preocupa-se em dissecar os erros mais comuns que podem invalidar a descrição do usuário, sendo que a especificação destes figura no GLOSSÁRIO do ANEXO 2 compreendendo a numeração de 11 a 17.

3.3.9.4. - "LGRILL" - GERAÇÃO DO MODO DE GRAVAÇÃO E TAMANHO DO DADO.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"ISPTO" - chave de controle para gravação em TAMANHO VARIÁVEL.

"KASE" - variável auxiliar no percurso dos vetores "KFITA(M)" e "KVATA(M)".

"IBIT" - chave de controle na pesquisa de tamanho do dado.

"INGRID" - variável auxiliar na indicação de "TF" ou "TV".

II - OBJETIVOS:

. validar o modo de gravação do dado e proceder a geração do mesmo na Tabela de Símbolos.

. gerar o comprimento do campo, no caso de dimensão fixa

. enviar MENSAGENS DE ERRO ao usuário, através de chamadas a "IERRO(N)".

III - DESCRIÇÃO GERAL.

"LGRILL" utiliza a chave "INGRID" para referenciar o MODO DE GRAVAÇÃO dos dados, cuja alocação nos cartões é feita através dos caracteres "TF" ou "TV".

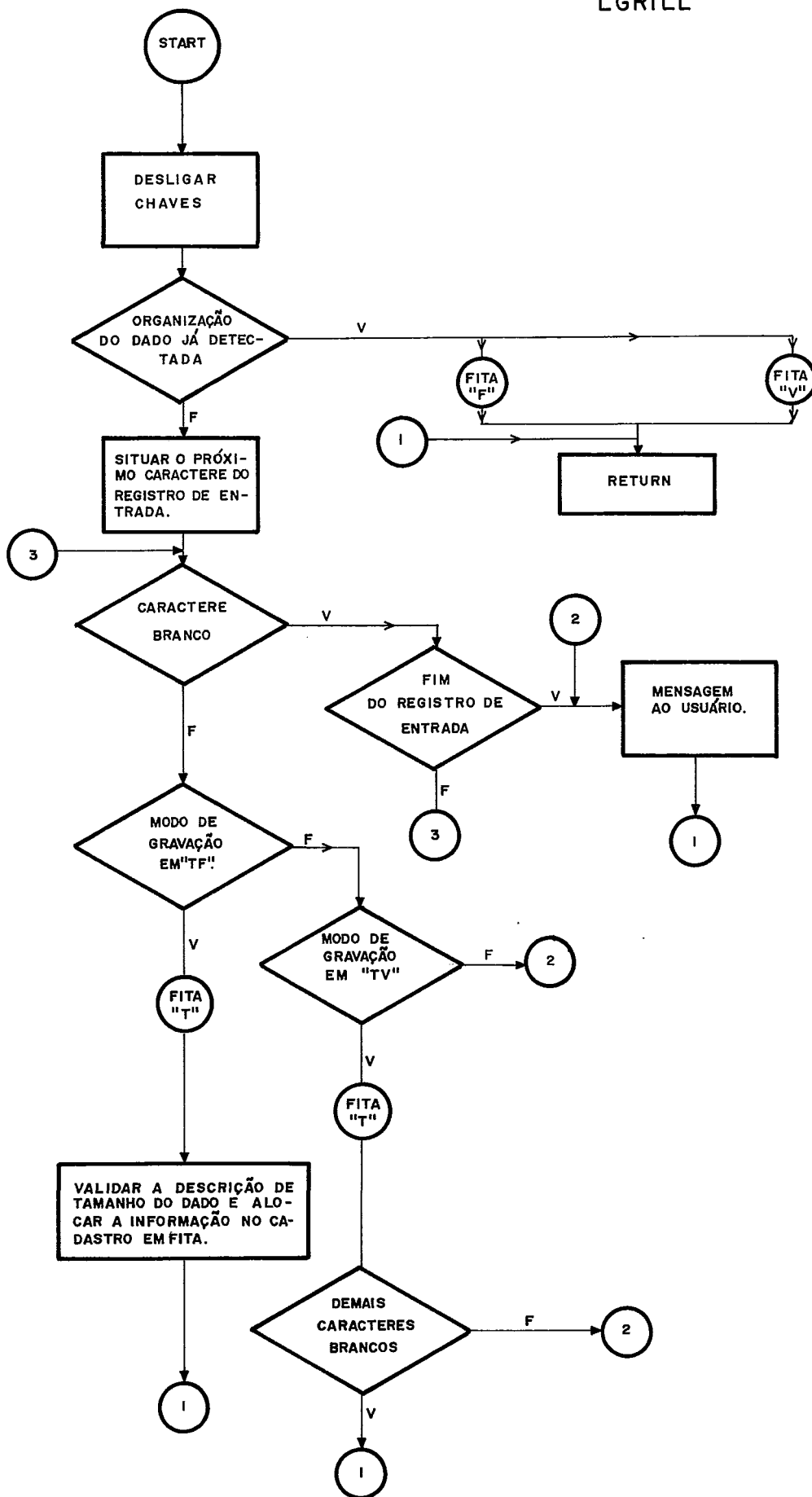
Sua meta principal está ligada ao percurso da parte final do registro de entrada, através do apontador "K", que desloca-se desde a coluna onde cessou a pesquisa de "ICOGI" até o fim. Este campo abrange tanto o modo de gravação do dado como também a ESPECIFICAÇÃO DE SEU TAMANHO, que no caso de comprimento VARIÁVEL é representado por caracteres brancos.

As atualizações nos arquivos, de saída, em caso de dados INVÁLIDOS utilizam o comando

"I = I - (IMARC + 6)"

que ocasiona um retorno no endereço da fita correspondente ao total de caracteres de nome do dado ("IMARC"), mais as 6 posições referentes a: delimitador, "BIT" de listar ou contar, "BIT" de testar, tipo e organização do dado.

"LGRILL"



IV - ESQUEMA DE "PROCEDURES".

Ao testar a chave de controle, no início da "SUBROUTINE LGRILL" visa-se obter informação a respeito do modo de gravação considerado.

Abaixo, encontram-se as 3 possíveis ocorrências, acompanhadas dos procedimentos necessários em cada caso.

1 - "INGRID = 1" (CHAVE LIGADA) - DADO DE TAMANHO FIXO.

- . executar "ISAI(L)" para gravação do 2º caractere "F" do modo de gravação, através do comando, "CALL ISAI (IALFA(6))".

- . atualizar o apontador "K" do arquivo de entrada, de modo a posicionar a descrição de tamanho do dado.

- . gerar o campo de TAMANHO nos cadastros em fita.

2 - "INGRID = 2" (CHAVE LIGADA) - DADO DE TAMANHO VARIÁVEL.

- . executar "ISAI(L)" para gravação do 2º caractere "V" do modo de gravação, através do comando "CALL ISAI (IALFA(22))".

- . verificar se no arquivo de entrada todos os caracteres posteriores a descrição de TAMANHO VARIÁVEL são brancos.

- . reservar espaço em fita magnética correspondente a 2 caracteres, para acondicionar o comprimento do dado

a ser calculado no 2º MÓDULO.

3 - "INGRID = 0" (CHAVE DESLIGADA) - MODO DE GRAVAÇÃO
DESCONHECIDO.

. atualizar o apontador "K" do arquivo de entrada, de modo a situar o 1º caractere não branco localizado após a organização do dado

. detectar a presença de TAMANHO FIXO ou TAMANHO VARIÁVEL através de acareação dos vetores "KFITA(M)" e "KVATA(M)".

. gravar em fita magnética o 1º caractere "T" do modo de gravação.

. acionar a chave "KIERRO" em caso de descrição inválida, de modo a informar ao sistema que o dado não deve ser cadastrado

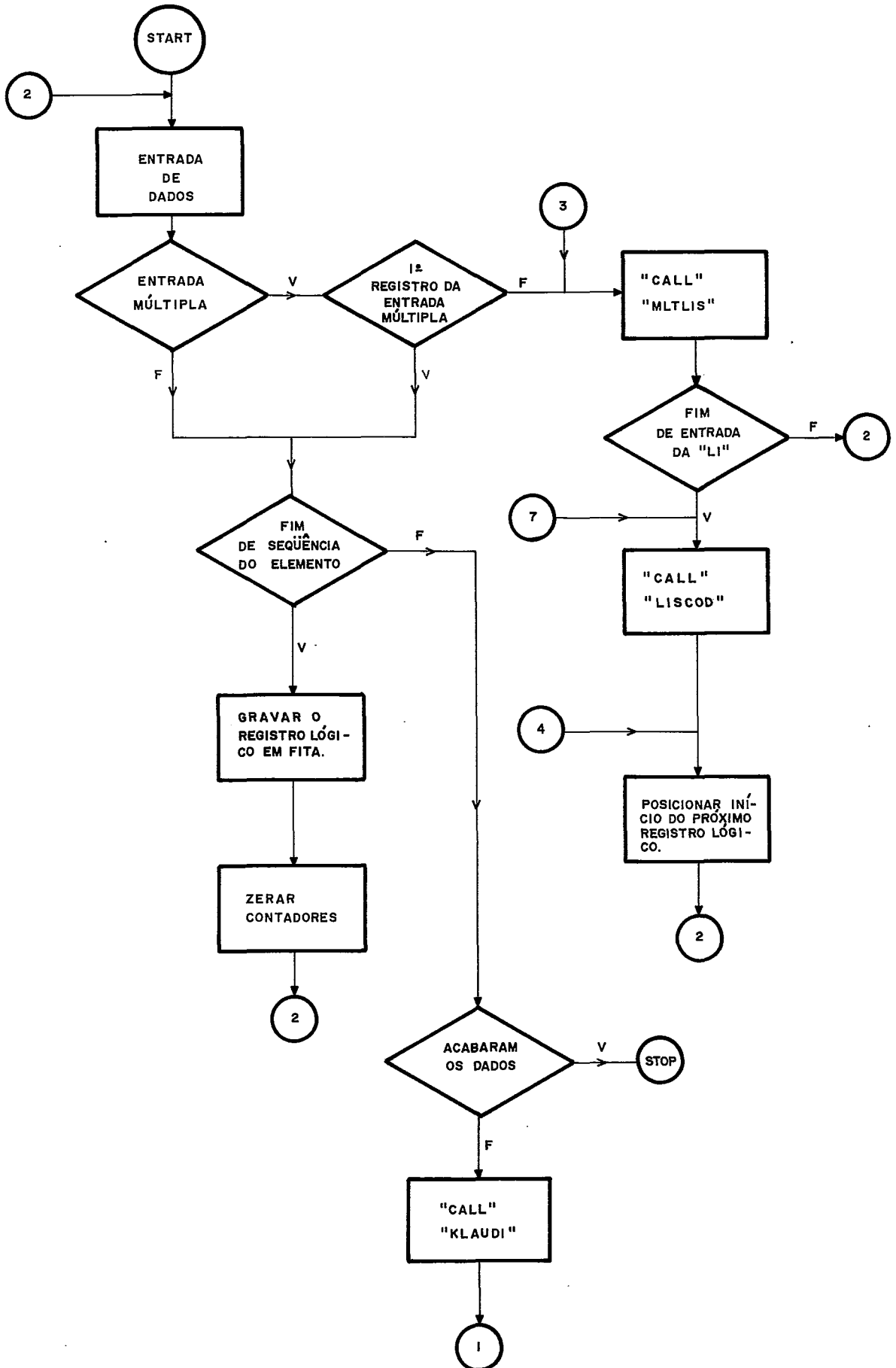
. atualizar o apontador dos arquivos de saída

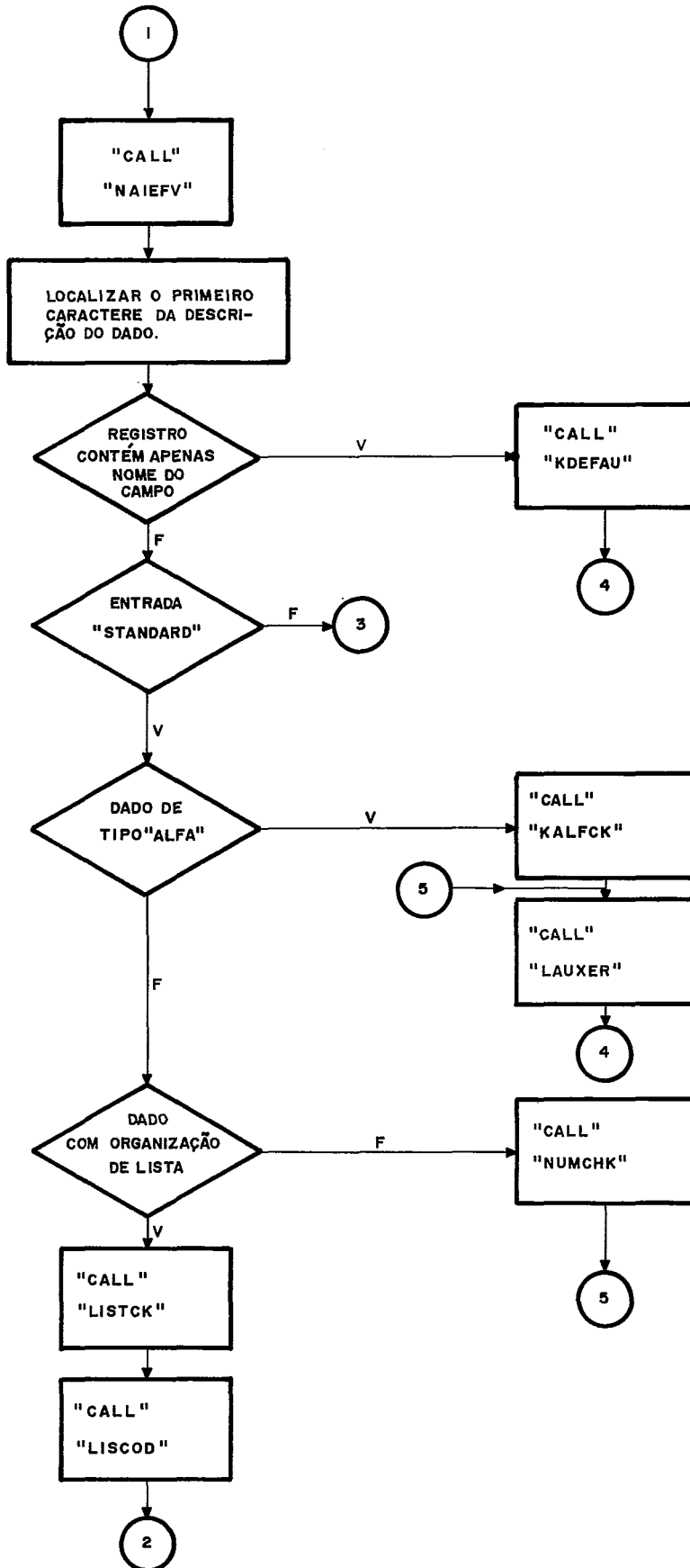
. aventar a possibilidade de ocorrência dos ERROS de 18 a 27 descritos no GLOSSÁRIO do ANEXO 2.

CAPÍTULO IV

VALIDAÇÃO DE DADOS (2º MÓDULO)

4.1. - DIAGRAMA DE BLOCOS DA VALIDAÇÃO





4.2. - ANÁLISE DA VALIDAÇÃO DE DADOS.

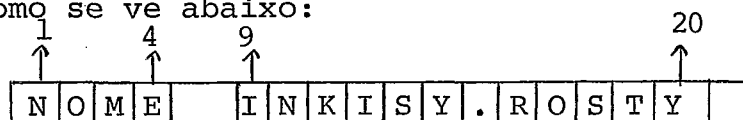
4.2.1. - ESTRUTURA GERAL.

Como foi visto no capítulo 2.2, o 2º MÓDULO DO SISTEMA compreende a parte de VALIDAÇÃO DOS DADOS.

A abrangência desta fase é manifestamente, uma característica de grande importância, pois sua atuação envolve um campo prolixo que vai desde a ALIMENTAÇÃO DOS DADOS no equipamento de entrada até a GRAVAÇÃO DOS CADASTROS em fita magnética.

A ENTRADA aventada pelo usuário é perfurada em cartões de 80 colunas e contém o NOME DO CAMPO e a DESCRIÇÃO DO DADO.

A leitura para a MEMÓRIA é feita através do vetor NIDADO(K), cujas posições demarcam cada caractere explicitado, tal como se vê abaixo:



Em síntese, pode-se aceder um esquema geral, assim constituído:

- os REGISTROS são lidos para a MEMÓRIA
- o NOME DO CAMPO é testado para checar sua INCLUSÃO na Tabela de Símbolos;
- a descrição do dado propriamente dito é dis-

secada, a fim de comprovar acordo com os PARADIGMAS DO SISTEMA, expostos no MANUAL DO USUÁRIO.

- em PRESENÇA DE ERRO, o elemento correspondente não é cadastrado e a SEQUÊNCIA DE OPERAÇÕES prossegue AUTOMATICAMENTE;
- no caso de ENTRADA VÁLIDA, o dado é CADASTRADO (Fig. 4.2.1-A) através de uma configuração que envolve o TAMANHO e a ESPECIFICAÇÃO DO DADO.

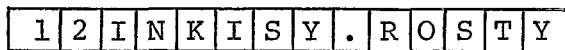


Fig. 4.2.1-A - ESQUEMA DE UM NÓ DO CADASTRO EM FITA MAGNÉTICA.

4.2.2. - TIPOS DE ENTRADA.

Considerando a PATENTE FLEXIBILIDADE DO SISTEMA e a GRANDE AUTONOMIA do usuário na descrição dos dados, podem-se caracterizar 3 tipos de ocorrência:

- I - ENTRADA "STANDARD" - quando a descrição do dado fica inteiramente contida num cartão único, isto é, o total de caracteres é menor ou igual a 80. É o caso mais comum,

no processamento habitual. Contém o NOME DO CAMPO e a DESCRIÇÃO DO DADO.

II - ENTRADA "MÚLTIPLA" - caso a descrição do dado ocupe mais de um cartão, isto é, o total de caracteres é superior a 80. Ocorre nos casos de organização em "LI", que contém muitos elementos do LISTÃO da Tabela de Símbolos. Utiliza como código de identificação o dígito alfanumérico '1' alocado na coluna 'UM' e contém: o CÓDIGO DE IDENTIFICAÇÃO, O NOME DO CAMPO E A DESCRIÇÃO DO DADO.

III - ENTRADA COM "DEFAULT" - hipótese em que a descrição do usuário é omissa, possibilitando derriscar o dado, sempre que aquele não dispõe de meios para obter este. Contém apenas referência ao NOME DO CAMPO.

4.2.3. - CORRELAÇÃO NO PROCESSAMENTO

O DIAGRAMA DE BLOCOS apresentado em 4.1 elucidada o ESQUEMA GERAL DA VALIDAÇÃO DOS DADOS (2º MÓDULO).

Através de seu esquema, depreende-se:

I - A entrada "MÚLTIPLA" pode cingir duas hipóteses:

- a) TRATAR-SE DO PRIMEIRO CARTÃO DA MESMA - isto é, o vetor "NIDADO(K)" contém o código de identificação, o nome do campo e o início da descrição do dado. Para atender este caso, de modo a SIMPLIFICAR AS TAREFAS DO SISTEMA OPERACIONAL, recorre-se à "KLAUDI" para localização de nome do campo e procede-se do mesmo modo que para ENTRADA "STANDARD", tal como se verá mais adiante.
- b) TRATAR-SE DE UM QUALQUER DOS CARTÕES DE CONTINUAÇÃO DA MESMA - isto é: vetor "NIDADO (K)" contém o código de identificação e a continuação do dado. Em tal caso, acondiciona-se a EXECUÇÃO DE "MLTLIS" para validação e armazenamento do dado em ÁREA DE TRABALHO PROVISÓRIA, cabendo a "LISCOD" proceder à DECODIFICAÇÃO do mesmo, como fase preparatória para posterior cadastramento.

II - O teste de FIM DE SEQUÊNCIA para detectar a presença do último dado que compõe um mesmo elemento é realizado AO FIM DE CADA ENTRADA, excetuando, é evidente, o caso dos itens acima.

III - A TRANSFERÊNCIA PARA FITA MAGNÉTICA, dos caracteres correspondentes à gravação total de um elemento do cadastro é feita sempre que adergar a entrada do último dado considerado. A transladação é baseada no uso de RESIDÊNCIAS PROVISÓRIAS em áreas de trabalho a parte, que possibilitam a gravação, em fita, somente nos casos em que não há ocorrência de erros.

IV - A CADA DADO de entrada corresponde uma SÉRIE DE PROCEDIMENTOS que vão corroborar ou não sua anuência como integrante do cadastro, observando o seguinte roteiro:

V - "KLAUDI" localiza o nome do campo na Tabela de Símbolos.

VI - "NAIEFV" fornece o espectro do tipo, organização e modo de gravação do dado.

VII - O teste de entrada com "DEFAULT" é realizado para analisar se todos os caracteres que seguem o nome do campo são brancos.

VIII - O subprograma "KDEFAU" é executado a fim de cadastrar uma referência à omissão do usuário na descrição do dado.

IX - O teste para detectar presença de entrada "STANDARD" é promovido com o escopo de separar o caso que retrata o 1º cartão de entrada "MÚLTIPLA", onde o comando é movido para "MLTLIS".

A ENTRADA "STANDARD" pode circundar:

X - A chamada de "KALFCK" ou "NUMCHK" para validação do dado, sempre que se tratar de descrição alfanumérica ou numérica, respectivamente.

Em qualquer dos dois casos segue-se a execução de "LAUXER", que aloca a referida descrição do usuário no minicadastro provisório.

XI - A chamada de "LISTCK", também para validação do dado, sempre que emergir organização de LISTA.

A DECODIFICAÇÃO INERENTE a este caso tem, então, sua plataforma assegurada, quando o fluxo assume a direção de "LISCOD" que faz o cadastramento do dado, na área de trabalho anteriormente definida.

4.3. - FASES DA VALIDAÇÃO

4.3.0. - TABELA DE SÍMBOLOS - MODELO

Os próximos itens do capítulo IV (4.3.1. a 4.3.13) são precedidos de uma TABELA DE SÍMBOLOS - MODELO (Fig. 4.3.0-A) cujo objetivo é facilitar a compreensão do leitor nesta fase mais complexa do sistema, que é a validação e cadastramento dos dados.

As explanações encontradas nos subprogramas seguintes referem-se, portanto, à criação de um CADASTRO-MIRIM, cujos DADOS e ENDEREÇOS têm em vista a TABELA GERADA,

abaixo:

```

N O M E * 0 0 A 0 0 T V 0 0 0 0 0 0 M A T R Í C U L A * 0 0
N 0 0 T F 0 6 0 0 0 0 T E M P O . D E . S E R V I Ç O * 0 0
N 0 0 T F 0 2 0 0 0 0 E S T A D O . C I V I L * 0 0 N L E T
F 0 4 0 1 9 2 T E L E F O N E * 0 0 N 0 0 T F 0 7 0 0 0 0 C
A T E G O R I A * 0 0 N L E T F 0 4 0 2 3 8 Q U A D R O * 0
O N L E T F 0 4 0 2 7 3 M A T É R I A S . Q U E . L E C I O
N A * 0 0 L I T V 0 0 0 2 8 4           $ S O L T E I R O / C
A S A D O / V I Ú V O / D E S Q U I T A D O /           $ S U
B S T I T U T O / C O N T R A T A D O / E F E T I V O /
           $ 1 / 2 / 3 /           $ P R O C E S S A M E N T O . D E
. D A D O S / C Á L C U L O S . I M O B I L I Á R I O S / E
S T A T Í S T I C A / G E O M E T R I A . A N A L Í T I C A
/ E Q U A Ç Õ E S . D I F E R E N C I A I S / L I N G U A G
E M . C O B O L / F O R T R A N / M A T E M Á T I C A / /

```

Fig. 4.3.0.A - TABELA DE SÍMBOLOS - MODELO

4.3.1. - "KLAUDI"

PERCURSO E VALIDAÇÃO DO NOME DE DADO, DESCRITO PELO USUÁRIO.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS

"KAND" - Chave para controle na seqüenciação dos dados.

"INCRE" - Contador de caracteres válidos encontrados na descrição do usuário.

II - OBJETIVOS:

- . analisar a descrição do nome de dado, feita pelo usuário, verificando se este consta da Tabela de Símbolos
- . examinar a seqüência dos dados alimentados no equipamento de entrada
- . enviar MENSAGEM DE ERRO, através de chamadas à "IERRO(N)".

III - DESCRIÇÃO GERAL.

"KLAUDI" é solicitada para cada elemento a ser cadastrado, tantas vezes quantos são os registros lógicos que compõem o cabeçalho dos arquivos.

Utiliza, na conquista de seu intento, uma conjugação no movimento de seus APONTADORES:

"JUCK" - que percorre os cadastros em fita.

"KLI" - que caminha sobre o arquivo em cartões.

O exemplo apresentado na Fig. 4.3.1.A mostra

que:

- a) no início da "SUBROUTINE"; "JUCK" assinala os endereços 1 e 19, respectivamente, conforme se trate da 1a. ou 2a. execução de "KLAUDI", isto é, aponta para o primeiro caractere do nome de dado em cada registro lógico.
- b) no final, "JUCK" contém os endereços 5 e 28, respectivamente, isto é, assinala os delimitadores.

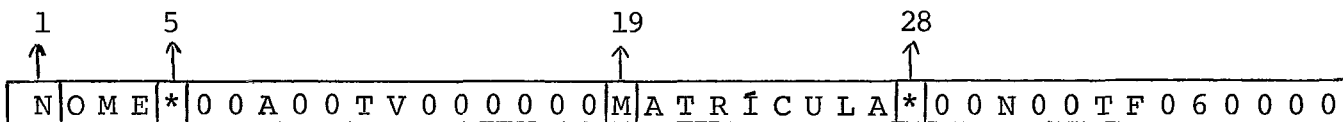


Fig. 4.3.1.A - TRECHO DA TABELA DE SÍMBOLOS COMPOSTO DE 2 REGISTROS LÓGICOS

O leitor pode estabelecer um paralelo na sincronia de movimentos desses 2 apontadores, ao observar a Fig. 4.3.1.B, em que:

- a) no início da execução, "KLI" assinala o 1º caractere da descrição do nome de dado: "N" ou "M", conforme se considere, respectivamente, a 1a. ou 2a. entrada.
- b) no fim, "KLI" aponta para o caractere situado imediatamente após o nome de dado (colunas 5 e 10, respectivamente).

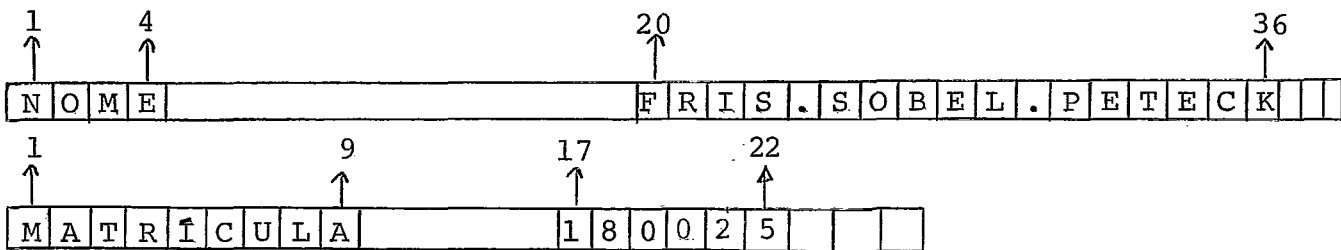


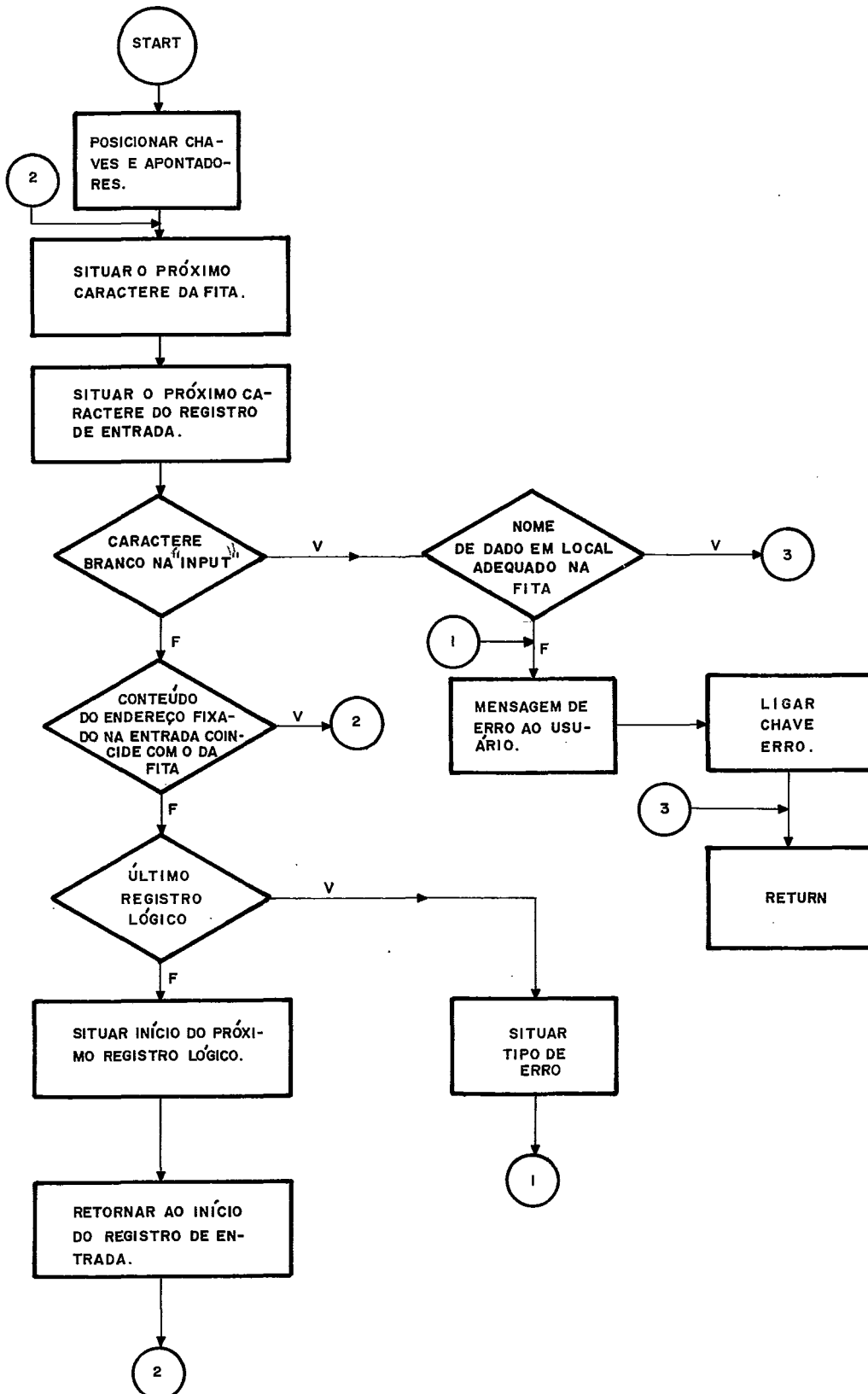
Fig. 4.3.1.B - ESQUEMA DE UMA ENTRADA VÁLIDA NA
GERAÇÃO DA TABELA DE SÍMBOLOS DA
Fig. 4.3.1.A

IV - ESQUEMA DE "PROCEDURES".

Liminarmente, o trâmite de "KLAUDI" consiste em deslocar o apontador "KLI" ao longo do vetor "NIDADO(K)". Ao emergir o primeiro caractere não branco, compara com seu correspondente na cabeça da Tabela de Símbolos. Não havendo dissidência, desloca "KLI" e "JUCK" para o endereço seguinte e repete o procedimento sucessivas vezes até encontrar o delimitador de dados, no arquivo de entrada.

Justamente nesse ponto analisa o conteúdo do endereço fixado na Tabela de Símbolos, para detectar a presença do asterisco.

Em caso afirmativo e se, também a chave "KAND" relatar que a entrada corresponde, em termos de disposição sequencial, ao cadastro em fita, fica explícita a validade da descrição do usuário e o controle é devolvido ao programa principal.



Se, porém, a chave "KAND" estiver ligada, evidenciando que a ordem dos dados descritos não corresponde à da Tabela de Símbolos, ocorre o ERRO 32.

Ao sobrevir discrepância entre posições correspondentes do vetor "NIDADO(K)" e o arquivo em fita, testa-se o conteúdo de "IMPRI(K)". Caso seja ocupado por um asterisco, o erro fica evidenciado, pois, se houve coincidência até esse elemento e ainda existem caracteres não brancos no vetor "NIDADO(K)", fazendo parte da descrição, o nome possui pelo menos um caractere além do que deveria. Trata-se do ERRO 31.

Caso não seja ocupada por asterisco, testa-se a inserção de caracteres brancos, numéricos ou especiais, que podem suscitar os ERROS 28, 29 ou 30, respectivamente.

No caso em que a discrepância ocorre desde o início avança-se o apontador "JUCK" até o próximo registro lógico, através de duas etapas:

- . a primeira, que salta os caracteres correspondentes ao tamanho do campo, fazendo estação no delimitador de dado.
- . a segunda, que vai até o limiar do próximo registro, transpondo as 13 posições correspondentes aos caracteres fixos: "bit" de listar ou contar, "bit" de testar, tipo de dado, organização, modo de gravação, tamanho e endereço da lista.

4.3.2. - "NAIEFV"

PESQUISA DO TIPO, ORGANIZAÇÃO E MODO DE GRAVAÇÃO DOS DADOS, ATRAVÉS DE PERCURSO DA TABELA DE SÍMBOLOS.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

(VER DESCRIÇÃO GLOBAL - CAPÍTULO 3.2).

II - OBJETIVOS:

Percorrer os arquivos em fita magnética, de modo a espartir informações geradas durante a execução do 1º MÓDULO.

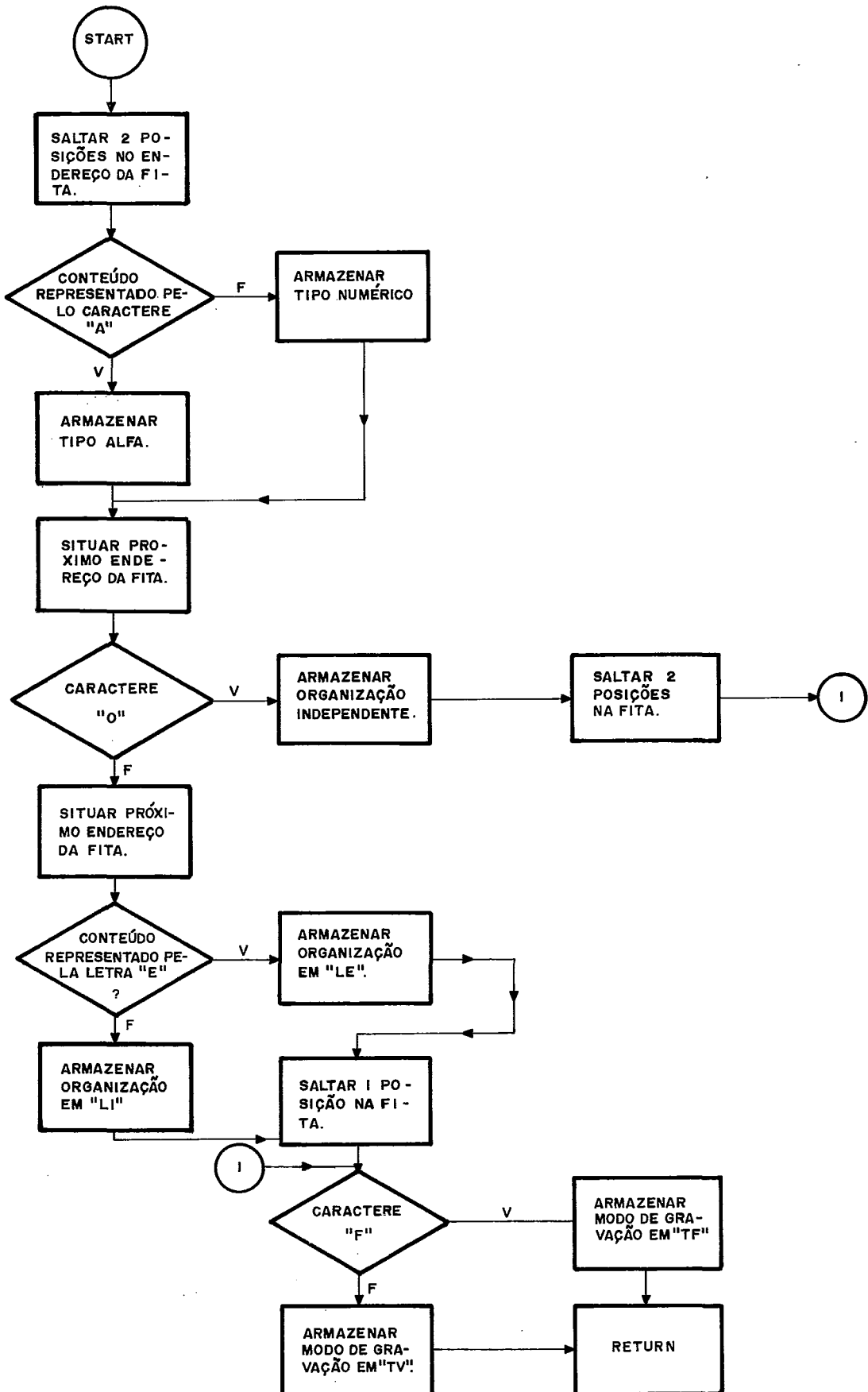
III - DESCRIÇÃO GERAL.

"NAIEFV" trabalha com o apontador "JUCK", que caminha sobre uma trajetória fixa e bem determinada.

O exemplo apresentado abaixo refere-se ao 1º dado do cadastro modelo (Fig. 4.3.0.A) e, por extrapolação, conclui-se o procedimento realizado para todos os demais.

A Fig. 4.3.2.A, que para maior comodidade do leitor transpõe o registro correspondente ao elemento supra-citado, evidencia o seguinte:

"NAIEFV"



- . no início da "SUBROUTINE"; "JUCK" assinala o delimitador de dado, posição mantida desde o final de "KLAUDI", isto é: "JUCK = 5"
- . "JUCK" faz avanços progressivos, ao longo da Tabela de Símbolos, assinalando as referências contidas nas posições 8, 9-10 e 11-12, correspondentes, respectivamente, às descrições de tipo de dado, organização e modo de gravação.
- . Ao final da execução de "NAIEFV", o apontador "JUCK" interrompe seu percurso no endereço 12, que contém o 2º caractere do modo de gravação do dado.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
N	O	M	E	*	0	0	A	0	0	T	V	0	0	0	0	0	0

Fig. 4.3.2.A - TRECHO DA TABELA DE SÍMBOLOS-
 MODELO DESCRITA NA Fig. 4.3.
 0.A

IV - ESQUEMA DE "PROCEDURES".

Imaginando um REGISTRO LÓGICO da Tabela de Símbolos - modelo, tal como o da Fig. 4.3.2.B, que supõe 12 como endereço do primeiro caractere gravado, torna-se fácil visualizar o procedimento inicial de deslocar "JUCK" ao longo

da fita magnética até atingir o tipo de dado. Isto se processa através de um INCREMENTO NO APONTADOR dos cadastros em fita, representado por $"JUCK = JUCK + 3"$

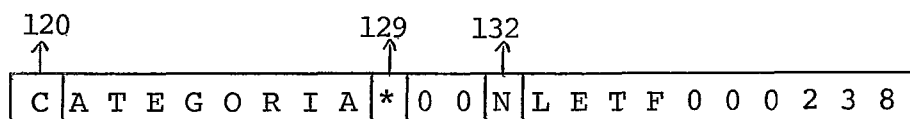


Fig. 4.3.2.B - POSIÇÃO INICIAL E FINAL DE
"JUCK" (129 e 132) NA PESQUISA
DO TIPO DE DADO

O caractere que denota tipo de dado possui, apenas, duas possibilidades lógicas: "A" ou "N", uma vez que na Tabela de Símbolos todos os erros detectados são eliminados antes de entrarem no sistema.

Sendo assim, depreende-se o tipo do dado utilizando, como PROCEDIMENTO ÚNICO, um teste feito no caractere assinalado pelo endereço atual do apontador ($"JUCK = 132"$).

Caso este mantenha similitude como o "A" (Fig. 4.3.2.B), armazena-se o campo como alfa ($"NA = 1"$) e, em caso adverso, como numérico ($"NA = 2"$).

Quanto a ORGANIZAÇÃO DO DADO, como são dois os caracteres que a representam, situa-se cada um deles, separadamente. Para galgar o primeiro, é suficiente DESLOCAR "JUCK" até o PRÓXIMO ENDEREÇO, fazendo:

$"JUCK = JUCK + 1"$

A Fig. 4.3.2.C, abaixo, que supõe 19 como endereço inicial do registro lógico, ilustra essa atualização:

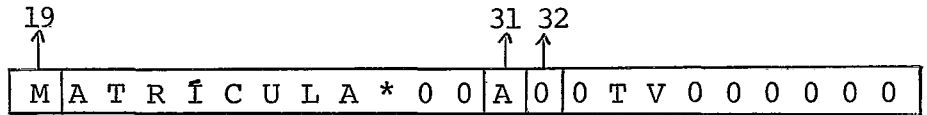


Fig. 4.3.2.C - POSIÇÃO INICIAL E FINAL DE "JUCK" (31 e 32) NA PESQUISA DE ORGANIZAÇÃO DO DADO.

O endereço atual do cadastro em fita ("JUCK = 32") pode conter um ("0"), ou um "L", pois as três possibilidades GERADAS NO 1º MÓDULO são:

"00" → DADO INDEPENDENTE

"LI" → LISTA INCLUSIVA

"LE" → LISTA EXCLUSIVA

Se o teste revelar a presença de um zero, fica automaticamente confirmado que o dado é independente e, como corolário, o ENDEREÇO SEGUINTE ("JUCK = 33") contém o outro "0" da designação. Neste caso, armazena-se a informação, alocando "IORG = 1".

Se o teste patentear a existência de um "L", pode-se desembuçar um dado em LISTA. Neste caso, só o conteúdo do próximo endereço, obtido através da atribuição "JUCK = JUCK + 1", pode fornecer um espectro completo da organização do dado.

Uma vez posicionado o referido caractere, isto é, fixado o apontador "JUCK" no endereço 89 (Fig. 4.3. 2. D), testa-se o seu conteúdo. Isto porque, como se vê no cadastro - exemplo da Fig. 4.3.0.A, o endereço inicial do registro lógico é 72.

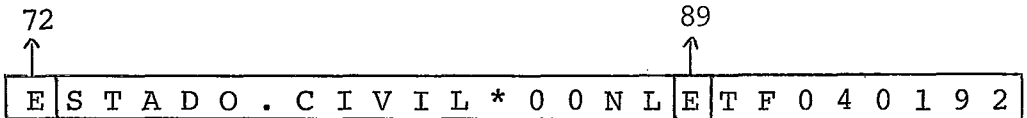


Fig. 4.3.2.D - LOCALIZAÇÃO DO APONTADOR "JUCK" AO FIXAR O 2º CARACTERE DA ORGANIZAÇÃO DO DADO ("JUCK = 89").

Se o espectro assinala um "E", sabe-se estar diante de um dado em lista exclusiva ou, se descortinado um "I", o diagnóstico é de lista inclusiva. Assim sendo, armazena-se a informação na memória, fazendo "IORG = 3", ou "IORG = 2", respectivamente.

Finalmente, ao ENCETAR CONSIDERAÇÕES sobre PROCEDIMENTOS utilizados na busca do MODO DE GRAVAÇÃO fornecida pela Tabela de Símbolos, tem-se dois aspectos a ponderar:

I - O DADO É DE ORGANIZAÇÃO INDEPENDENTE

Admitindo, como exemplo, o valor 42 para endereço do 1º caractere do registro lógico, tem-se que a posição inicial do apontador é: "JUCK = 62" (Fig. 4.3.2.E). Is-

to porque, como foi visto no caso acima, tratando-se de dado independente, só o 1º caractere da organização é testado, sendo o outro assumido, implicitamente.

Para posicionar o caractere que vai indicar modo de gravação em tamanho fixo ou variável, é suficiente fazer:

"JUCK = JUCK + 3"

O endereço atual assume o valor 65 e o caractere nele contido é testado para verificar se corresponde ao "F".

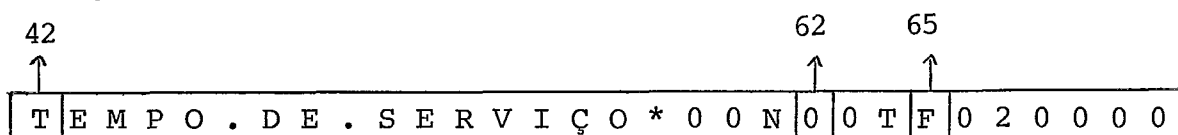


Fig. 4.3.2.E - LOCALIZAÇÃO DO APONTADOR "JUCK", AO FIXAR O 2º CARACTERE DO MODO DE GRAVAÇÃO DO DADO ("JUCK = 65").

Aloca-se a informação na memória fazendo "MIGRA = 1" ou "MIGRA = 2", conforme o modo de gravação em tamanho fixo ou variável, respectivamente.

II - O DADO É DE ORGANIZAÇÃO EM LISTA.

Neste caso, apresentando o exemplo da Fig. 4.3.2.F, que inicia no endereço 142 pode-se aceder ao esquema abaixo, onde "JUCK" aponta para o endereço 153.

II - OBJETIVOS:

- . aferir a quitação de dados do tipo ALFA;
- . enviar MENSAGENS DE ERRO, através de chamadas à "IERRO(N)"
- . armazenar em ÁREA DE TRABALHO a parte, através do vetor "ICOMP(K)", os dados alfanuméricos lidos, para posterior cadastramento.
- . detectar o comprimento da descrição do usuário.

III - DESCRIÇÃO GERAL.

Manifestamente, o espectro de "KALFCK" mostra que a utilização do apontador "KLI" tem como escopo o percurso dos dados de entrada.

A "KARTY" é conferida a preeminência das ações na contagem de caracteres descritos, sendo que, no fim da execução, a referida variável funciona como meio auxiliar na organização da área de trabalho que seleciona os dados.

IV - ESQUEMA DE "PROCEDURES".

A filosofia de "KALFCK" baseia-se na análise do registro alimentado no equipamento de entrada, aventando de modo enexaurível todas as possibilidades de INTERVENIÊNCIA DE ERROS por distração do usuário.

A avaliação de tamanho do dado descrito é feita através da localização do 1º e último caracteres que o compõem, armazenados, respectivamente, em "KOMEÇO" e "KOFIM". O procedimento baseia-se na atribuição:

$$\text{"LEVA"} = \text{KOFIM} - \text{KOMEÇO} + 1$$

O resultado é armazenado em dígitos numéricos, através da variável "LEVA", que serve mais adiante como base para transformação desse número em alfanumérico, afinal possibilitando sua gravação em fita.

4.3.4. - "NUMCHK"

PERCURSO E VALIDAÇÃO DE DADOS NUMÉRICOS.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"KARTY" - contador de caracteres.

"KABRAN" - chave para controle de espaços em branco.

II - OBJETIVOS:

- . analisar a descrição de item numérico;
- . enviar MENSAGENS DE ERRO, através de chamadas a "IERRO(N)"

- . armazenar em área de trabalho à parte, através do vetor "ICOMP(K)", os dados numéricos lidos, para posterior cadastramento
- . detectar o comprimento da descrição do usuário.

III - DESCRIÇÃO GERAL.

"NUMCHK" trabalha somente com o apontador "KLI", cujo objetivo é percorrer o arquivo em cartões. Este, logo no início, assinala o 1º caractere não branco que surge após a descrição do nome de dado. Isto porque, após o retorno de "KLAUDI", observa-se seu deslocamento até este endereço, através de avanços progressivos ao longo do cadastro em fita.

Esta "SUBROUTINE", cujo objetivo é também avaliar dimensões, utiliza duas variáveis: "KOMEÇO" e "KOFIM", que armazenam, respectivamente, as posições inicial e final da descrição do usuário.

O comando aritmético que atribui à variável "LEVA" o tamanho do dado é:

"LEVA = KOFIM - KOMEÇO + 1"

IV - ESQUEMA DE "PROCEDURES".

Os testes feitos para aferir a validade de um

dado numérico visam a detectar a presença de caracteres brancos, especiais ou alfabéticos, que suscitam os ERROS 36, 37 e 38, respectivamente.

Há sempre duas hipóteses a considerar:

. DESCRIÇÃO VÁLIDA:

Caso em que se armazena o dado no vetor "ICOMP(K)", que o torna disponível para posterior cadastramento.

. DESCRIÇÃO INVÁLIDA:

Onde ocorre o acionamento das chaves de erro e retorno do fluxo para o programa principal, que lê o próximo registro, eliminando automaticamente o anterior.

4.3.5. - "LISTCK"

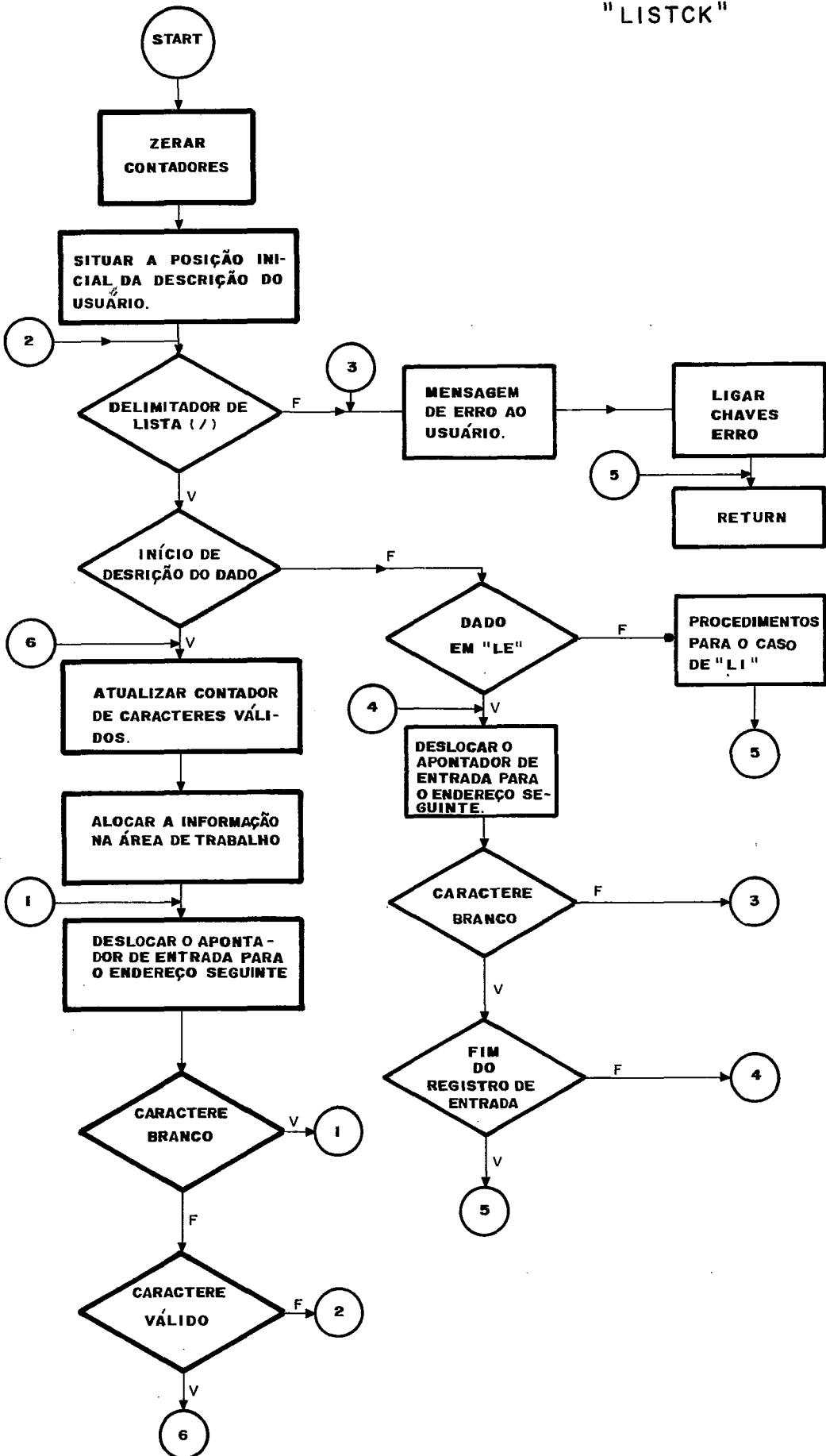
PERCURSO E VALIDAÇÃO DE DADOS EM LISTA.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

(VER DESCRIÇÃO GLOBAL - CAPÍTULO 3.2)

II - OBJETIVOS:

"LISTCK"



- . analisar a descrição de dados com organização de lista
- . armazenar em área de trabalho à parte, através do vetor "ICOMP(K)", os dados de entrada que envolvem "LI" ou "LE", para posterior decodificação
- . enviar MENSAGENS DE ERRO, através de chamadas a "IERRO(N)"
- . detectar o comprimento da descrição do usuário.

III - DESCRIÇÃO GERAL.

Como foi visto no capítulo 4.2. de Análise da Validação, um dado em lista, descrito pelo usuário, descortina um ou mais elementos da mesma e utiliza uma barra (/) como veículo de separação.

Pode-se aqui decantar certa abrangência no que se refere às funções de "LISTCK" referentes a contagem de caracteres e deliberação do dado através de "ICOMP(K)". Com isso, vislumbra-se um melhor fluxo nas operações posteriores que constam de:

- . PESQUISA DO DADO, nos listões da Tabela de Símbolos
- . DECODIFICAÇÃO DO MESMO, isto é, conversão para dígitos numéricos, que o representam, segundo certas convenções.

"LISTCK" também utiliza a serventia do apontador "KLI", inculcando, inicialmente, o 1º caractere da descrição do dado e conta com auxílio de "NAIEFV", que, ao percorrer a Tabela de Símbolos, aciona a chave "IORG", fornecendo informação a respeito de tartar-se de "LI" ou "LE".

Eventualmente, a descrição de dados em lista inclusiva não fica de todo acondicionada em apenas um registro de entrada, o que provoca requisição de "MLTLIS", ao invés de "LISTCK", por ser aquela inteiramente adaptada para esses casos.

IV - ESQUEMA DE "PROCEDURES".

Para corroborar uma plataforma viável na descrição de entrada, o primeiro teste feito é o que vai detectar a presença do delimitador inicial. Caso este se faça presente, trata-se de um bom começo, o que vai estatuir sua alocação na área de trabalho definida como "ICOMP(K)". Com isso, GALVANIZA-SE também a CONTAGEM DE CARACTERES VÁLIDOS através de "LÉVA", que assume o valor "UM". Caso contrário, cuida-se de um hiato extremamente significativo, que caracteriza o ERRO 40.

Ao ter como irrefutável a validade do 1º, promove-se uma investida que abrange os próximos caracteres, aventando, agora, também a possibilidade de existirem colunas em branco após a barra. Estas, no caso, são eclipsadas e o apontador "KLI" continua deslocando-se ao longo do registro,

até que surja um elemento não branco.

Neste ponto, o roteiro adstringe-se a dissecar todos esses caracteres localizados desde a posição atual até o delimitador de fim de dado.

A ocorrência de:

- . caracteres estranhos ao listão ou brancos suscita os ERROS 39 e 41, respectivamente, que implicam o ACIONAMENTO DAS CHAVES DE ERRO E RETORNO AO PROGRAMA PRINCIPAL para leitura do próximo dado.
- . caracteres alfabéticos, numéricos ou pontos promovem o INCREMENTO DO CONTADOR "LEVA" de caracteres válidos e cadastramento provisório no vetor "ICOMP(K)", que, em presença de erro posterior, pode ser deletado.

"LISTCK" tem seu alcantil assegurado quando atinge o caractere apontado por "KLI", que contém a 2a. barra (/). Este é o exato momento de testar o valor armazenado em "IORG" pela "SUBROUTINE NAIEFV", para verificar qual o tipo de lista descrita, se INCLUSIVA ou EXCLUSIVA, o que pode ocasionar a divisão abaixo:

. LISTA EXCLUSIVA ("IORG = 3")

Este caso vai cingir um teste de todos os caracteres situados após o espectro da 2a. barra, que, por defi

nição, só podem ser brancos, uma vez que a descrição de "LE" contém apenas um elemento.

Ficando patente essa característica, nada mais resta a fazer, senão armazenar a 2a. barra no vetor "ICOMP(K)" e incrementar o contador "LEVA".

Se, no entanto, eclodir algum caractere não branco, imediatamente após o delimitador de dado em "LE" des_u fecha-se um agravo de conteúdo lógico. Isto porque, havendo mais de um elemento na descrição do dado, trata-se de uma "LI", e "NAIEFV" revela que na Tabela de Símbolos consta uma "LE". O fato ocasiona o ERRO 42.

. LISTA INCLUSIVA ("IORG = 2")

Aqui, a 2a. barra tem o objetivo de separar elementos e não delimitar o dado, como no caso anterior. A alocação desta vai encantar a validação do 1º elemento, que já agora deve estar completo na área de trabalho e o procedimento é análogo para os demais integrantes da lista descritos pelo usuário.

4.3.6. - "LAUXER"

TESSITURA DE DADOS INDEPENDENTES, NO MINICASTRO "LYAUX(M)", VISANDO A POSTERIOR TRANSLADAÇÃO DO REGISU

TRO PARA O DESPOSITIVO DE MEMÓRIA AUXILIAR.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"LUZ" - chave de controle para acirrar chamadas à "SUBROUTINE LISCOD" ou "LAUXER", conforme a organização do dado.

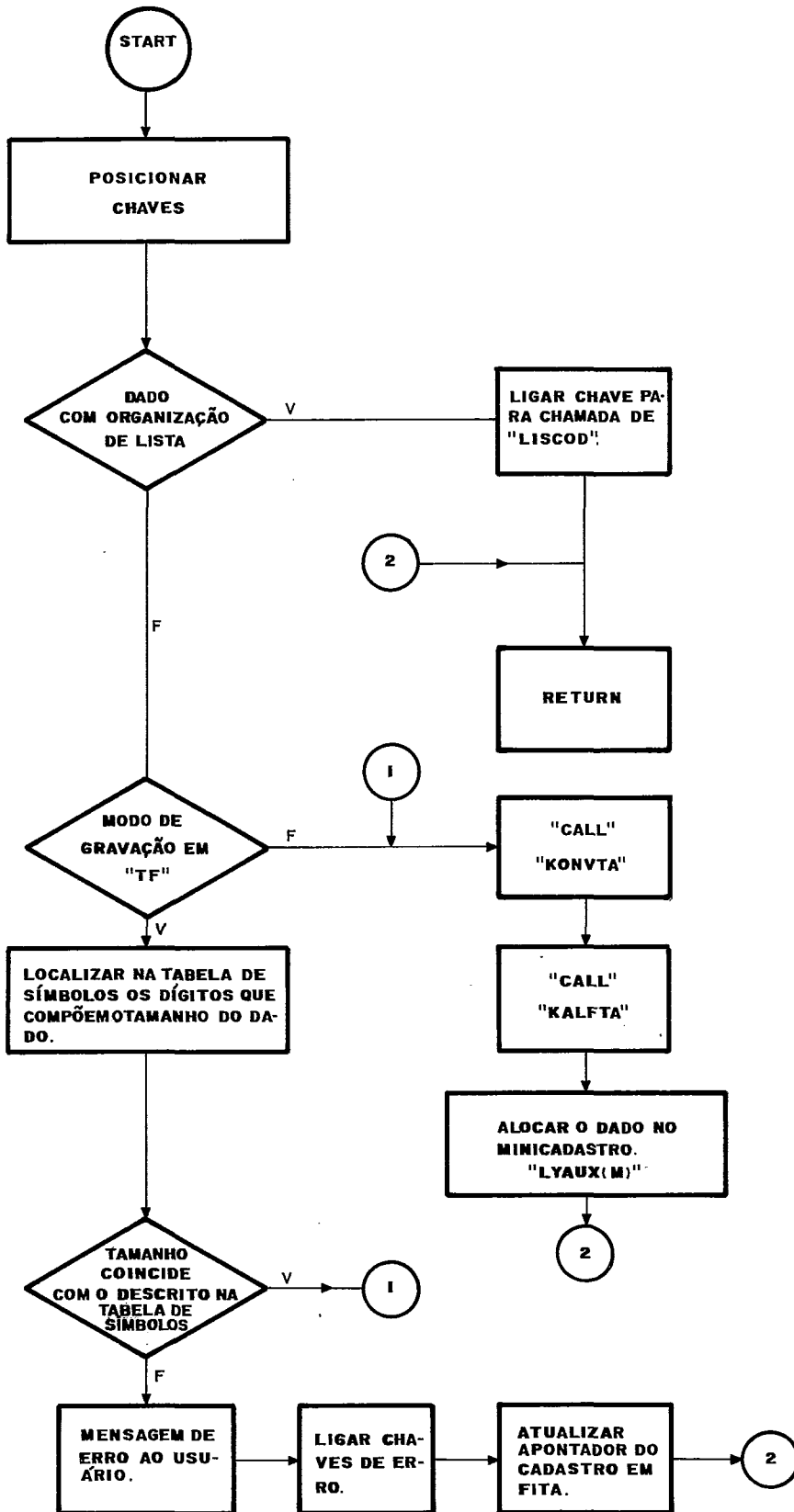
"KAPONT" - chave de controle para dados de TAMANHO FIXO, cujo total de caracteres gerado na Tabela de Símbolos mantém dissidência com a descrição do usuário.

II - OBJETIVOS:

- . alocar dados com ORGANIZAÇÃO INDEPENDENTE, acompanhados dos respectivos tamanhos, no minicadastro "LYAUX(M)", que contém a sequência de caracteres a ser gravada em fita.
- . asseverar que o comprimento da descrição do usuário, no caso de TAMANHO FIXO, coincide com o da Tabela de Símbolos.
- . enviar MENSAGENS DE ERRO, através de chamadas a "IERRO(N)".

III - DESCRIÇÃO GERAL.

"LAUXER"



Ao criar "LYAUX(M)", "LAUXER" tem como escopo utilizar-se de uma área de trabalho provisória para alocar os registros a serem gravados. Essa medida visa a aquiescer o cadastramento dos dados, somente nos casos em que não há ocorrência de erros.

A tarefa deste subprograma é essencialmente ligada à tessitura dos "NÓS" que compõem os arquivos.

Um "NÓ" é, justamente, a menor unidade do registro lógico (Fig. 4.3.6.A). Ele é composto basicamente de:

- . um "LINK", que contém o tamanho do dado
- . uma área de memória, que abrange a descrição do mesmo.

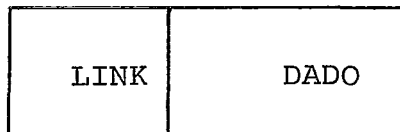


Fig. 4.3.6.A - ESQUEMA DE UM "NÓ" DO MINI-CADASTRO "LYAUX(M)".

"LAUXER" trata de dados que podem burlar qualquer um dos modos de gravação, isto é, TAMANHO FIXO ou TAMANHO VARIÁVEL. Assim sendo, logo no início, processa-se um teste que vai estatuir a presença de um ou de outro caso.

Tratando-se de tamanho fixo, há um desvio na seqüência lógica das operações, que vai ambargar a possível

ambigüidade de tamanho do dado gerado na Tabela de Símbolos e na descrição do usuário.

Dita verificação é extremamente simples e emprega:

- . o apontador "JUCK", para localizar, no arquivo em fita magnética, o endereço que contém o tamanho do dado.
- . o valor armazenado na variável "LEVA", que representa o comprimento obtido, através de "KALFCK" ou "NUMCHK".

IV - ESQUEMA DE "PROCEDURES".

"LAUXER" tem como medida liminar preeminente a sindicância da organização e modo de gravação do dado, através das posições mantidas pelas chaves "IORG" e "MIGRA", respectivamente.

Uma vez assegurada como plataforma a presença de um dado independente, pode-se coligir a seguinte divisão:

1 - DADOS DE TAMANHO FIXO:

Neste caso, antes de pensar na composição do "NÓ" a ser acrescentado em "LYAUX(M)", é exequível o processamento do teste, que vai comprovar se a descrição do usuário tem tamanho coerente com as informações da Tabela de Símbolos.

Como, no início de "LAUXER", "JUCK" assinala o segundo caractere "F" do modo de gravação do dado, o problema adstringe-se à localização das posições correspondentes no minicadastro "LYAUX(M)", onde está armazenado o comprimento da descrição feita pelo usuário. Deve-se a "KONVTA" e a "KALFTA" o encadeamento desses dígitos. (Ver 4.3.9 e 4.3.10).

Ao prover informações desse tipo, ressalta-se certa anuência do sistema no que se refere ao modo de proceder do apontador "IOG".

Na "SUBROUTINE KALFTA", por exemplo, antes da gravação de um caractere qualquer, procede-se à atualização de "IOG", que é, então, deliberado para o próximo endereço a ser preenchido.

Com essa inferência, pode-se dizer por indução que, após emergir o tamanho do dado em "LYAUX(M)", "IOG" assinala o segundo e último dígito componente da descrição (Fig. 4.3.6.B). Para verificar a coincidência de caracteres deve-se recuar o apontador "IOG" até o endereço imediatamente anterior.

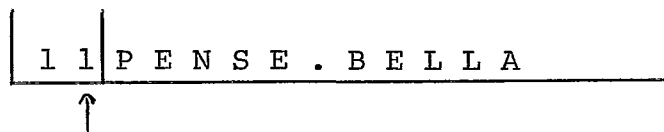


Fig. 4.3.6.B - ESQUEMA DE UM "NÓ" DO MINICADASTRO "LYAUX(M)", MOSTRANDO A POSIÇÃO DE "IOG" AO FINDAR A "SUBROUTINE KONVTA".

Ao promover esse estratagema, qualquer antítese na representação de um ou de outro suscita o ERRO 44 e, uma vez quitada a existência desse problema, procede-se do mesmo modo que para:

2 - DADOS DE TAMANHO VARIÁVEL:

Neste caso o prélio consiste, apenas, em fazer a adjeção dos "NÓS" que contêm dados independentes, no minicadastro "LYAUX(M)". Isto, de acordo com o que foi visto no preâmbulo, limita-se a duas etapas:

- 1) . gravação de tamanho do dado
- 2) . gravação do dado propriamente dito.

1) GRAVAÇÃO DE TAMANHO DO DADO:

O envolvimento desta tarefa está a cargo de "KONVTA" e "KALFTA". Aqui cabe apenas lembrar a filosofia utilizada no endereçamento do minicadastro, o que é de grande importância para a conjugação com "LAUXER".

A pertinência nessas operações de assentamento de tamanho do dado faz com que se recorra a certa maneira de proceder, bastante pragmática, cujo roteiro se resume em:

- . guardar o tamanho do dado, já calculado em subprogramas anteriores, através da variável "KEMBY"

- . descompactar o valor de "KEMBY"
- . converter os dígitos numéricos para alfanuméricos
- . incrementar o apontador "IOG", que passa a assinalar o próximo endereço a ser ocupado
- . alocar o respectivo caractere na posição correspondente do minicadastro.

Como se pode observar, a atualização do endereço opera-se antes da alocação do caractere, o que permite concluir que, ao final da execução de "KALFTA", o apontador "IOG" permanece indicando o último dígito gravado.

2) GRAVAÇÃO DO DADO PROPRIAMENTE DITO:

Cabe a "LAUXER" burlar o comando das operações para prover uma visão sistêmica da gravação em "LYAUX(M)". Cada dado auferido através do vetor "ICOMP(K)" é validado por "KALFCK" ou "NUMCHK", que têm o objetivo de auxiliar o cadastramento através da depuração prévia de cada caractere.

Assim sendo, a indefectível tarefa de "LAUXER" consiste em transferir a informação de "ICOMP(K)" para "LYAUX(M)". Esse movimento realiza-se através de uma iteração que recorre a formas de endereçamento compatíveis com a de "KALFTA" na alocação do tamanho do dado.

A Fig. 4.3.6.C mostra a posição final de "IOG", que, após alocar o dado, permanece assinalando o último caractere

tere gravado.

13	CARRIJO . ALVES	07	2567322
----	-----------------	----	---------

Fig. 4.3.6.C - ESQUEMA DE "NÓS" DO MINICADASTRO "LYAUX(M)", MOSTRANDO A POSIÇÃO DE "IOG" APÓS O CADASTRAMENTO DO 2º DADO (IOG = 24).

4.3.7. - "LISCOD"

DECODIFICAÇÃO DE DADOS COM ORGANIZAÇÃO DE LISTA EXCLUSIVA E TESSITURA DOS MESMOS NO MINICADASTRO "LYAUX(M)".

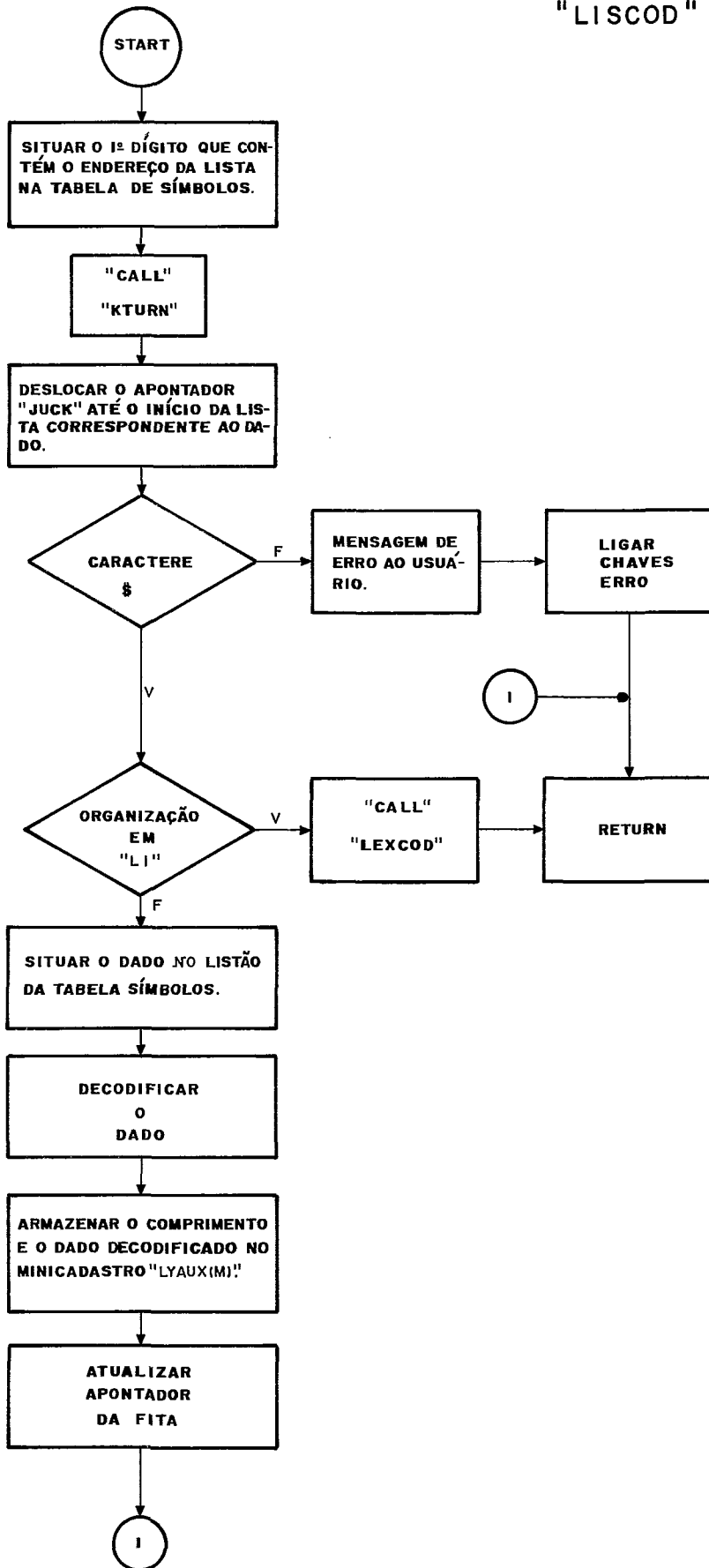
I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"JLEMBR" - variável utilizada para guardar a localização inicial do apontador "JUCK", em cada registro lógico.

"ICONDO" - indicador da ordem do elemento descrito pelo usuário na respectiva lista.

II - OBJETIVOS:

"LISCOD"



Cadastrar dados com organização de LISTA EXCLUSIVA, através de:

- . percurso do listão da Tabela de Símbolos, com o escopo de demarcar a localização dos mesmos
- . decodificação dos dados descritos pelo usuário, depois de coligir os respectivos endereços nas listas gravadas em fita
- . tessitura dos "NÓS" de "LYAUX(M)".

III - DESCRIÇÃO GERAL.

Com as explanações feitas no primeiro módulo, facilmente o leitor pode concluir que a estrutura de LISTA visa a facultar ao sistema maior flexibilidade, conferindo-lhe melhores condições ao manipular arquivos seqüenciais.

Em processamento de dados, sabe-se que o tamanho dos problemas e dados dos clientes muitas vezes são desconhecidos "a priori", e o usuário pode, a todo momento, pretender acrescentar ou retirar grupos de informações.

O único inconveniente das LISTAS é a limitação que fazem às "ESTRUTURAS ARBORESCENTES", o que no presente estudo é consideração sem nenhuma importância.

"LISCOD" retrata o manejo das LISTAS EXCLUSIVAS alocadas na Tabela de Símbolos, de acordo com a seqüência

lógica dos dados de entrada. Utiliza um ponteiro, cujo conteúdo é um endereço de memória, através do qual percorre os cadastros em fita, de modo a obter informações tais como:

a) ENDEREÇO DE INÍCIO DA LISTA.

Cada registro lógico que compõe os primeiros blocos físicos da fita contém informações simbólicas sobre um dado descrito pelo usuário, sendo que os 4 últimos caracteres formam o endereço da lista na Tabela de Símbolos e têm a configuração de alfanuméricos.

"LISCOD" transfere para "KTURN" o encargo de obter o número decimal correspondente, pois, com a simples execução deste MÓDULO DE CONVERSÕES, obtem-se a LOCALIZAÇÃO DA LISTA onde está SITUADO O DADO a ser cadastrado.

Existe, no sistema, interveniência de fatores que visam a OTIMIZAR A BUSCA DE INFORMAÇÕES. Sabe-se, por exemplo, que na estrutura inicial da fita o "\$", quando precedido de 4 brancos, denota início de lista. Essas colunas vazias funcionam como uma margem de segurança para possíveis erros de "SOFTWARE" no endereçamento.

b) DECODIFICAÇÃO DO DADO COM ORGANIZAÇÃO DE LISTA EXCLUSIVA.

Ao adergar um dado com organização de "LE", procede-se à localização do mesmo nas listas da Tabela de Símbolos.

los. A decodificação correspondente consiste em atribuir-lhe um valor numérico cuja estrutura lógica seja baseada na sua posição dentro da lista.

O cadastramento desse tipo de dado opera-se através de números de 4 algarismos. Isto porque as avaliações feitas liminarmente não impõem necessidade de maior ordem de grandeza.

Após a decodificação do dado, a tarefa de "LISCOD" adstringe-se à formação de "NÓ" que vai aloca-lo no minicadastro "LYAUX(M)", de acordo com a seqüenciação de endereços já iniciada por entradas anteriores.

IV - ESQUEMA DE "PROCEDURES".

"LISCOD" é solicitada, logo após a execução de "LISTCK", que fornece, através de "ICOMP(K)", um mapa geral do dado descrito pelo usuário, depois de validado (Fig. 4.3.7.A).

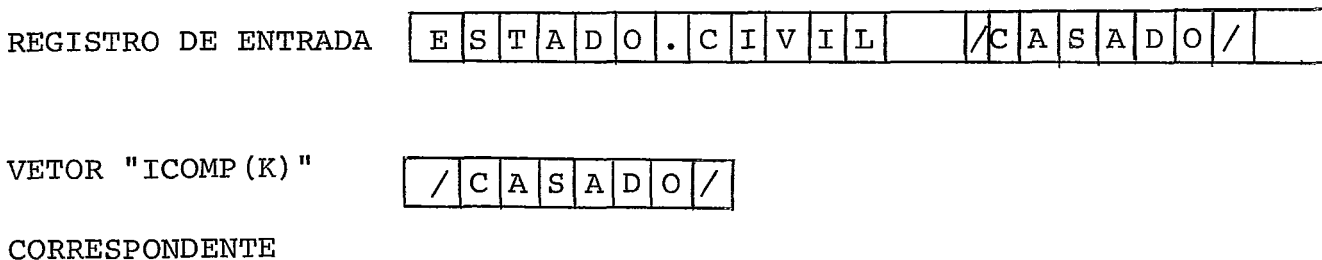


Fig. 4.3.7.A

Assim, sendo a tarefa deste subprograma reduzir-se a:

1) DETERMINAÇÃO DO ENDEREÇO DA LISTA QUE CON-
TÉM O DADO DE ENTRADA.

O percurso do listão da Tabela de Símbolos é feito através do apontador "JUCK", que no início de "LISCOD" inculca o 2º caractere da descrição do modo de gravação. Assim sendo, para localizar o primeiro dígito do endereço da lista correspondente a cada dado, basta saltar duas posições alusivas ao tamanho do campo.

Cabe à "SUBROUTINE KTURN" a transformação desse endereço, através da interpretação do número decimal formado pelos dígitos alfanuméricos explicitados.

Dada a prolixidade do sistema, deve-se checar o grau de precisão do endereço para corrigir pequenas defasagens que possam ocorrer na gravação em fita. Essa verificação é composta de um teste para caractere branco e um teste para "\$" e nos garante, é evidente, a presença imediata do 1º elemento da lista.

2) DECODIFICAÇÃO DO DADO:

Antes de galvanizar o início da decodificação do dado, há uma efêmera etapa, que consta da análise do tipo de lista considerada.

Caso esta tenha características de "LI", verifica-se o egresso do fluxo para "LEXCOD", que contém um tratamento especializado para tal hipótese.

Tratando-se, no entanto, de uma "LE", a seqüência lógica encaminha suas ações para a busca de localização do dado, nas listas da Tabela de Símbolos. Essa pesquisa emprega como descarte a variável "ICONDO", que representa uma área de memória criada para acondicionar a posição relativo do elemento no interior da lista.

Com isso, a decodificação limita-se, apenas, a explicitar esse número, através de uma compleição que envolve 4 dígitos e supõe o completamento das casas vazias com zeros.

3) GERAÇÃO DO "NÓ" QUE CONTÉM O DADO, NO MINI-CADASTRO "LYAUX (M)".

A estrutura do "NÓ", tal como foi convencionalmente anteriormente, envolve dois tipos de informação: tamanho e descrição do dado.

Com relação ao tamanho, sabe-se que no caso de "LE" o resultado da decodificação é um número de 4 dígitos, pois este tipo de organização só pode englobar um elemento da lista. Assim sendo, é suficiente que se transfira essa informação para "LYAUX (M)", através do encadeamento da seqüência 04 nos dois endereços consecutivos ao último caractere já alocado. Para isso, depois de enquadrar o primeiro zero, faz-se

a adjeção de uma unidade no apontador "IOG" do minicadastro, para, em seguida, armazenar o quatro.

Com relação ao dado propriamente dito, a gravação depende de uma simples transferência de sua representação decodificada para "LYAUX(M)".

A Fig. 4.3.7.B, abaixo, que tem em vista a Tabela de Símbolos gerada na figura 4.3.0.A apresenta uma entrada com organização de "LE" e o trecho correspondente do minicadastro a ser gravado em fita.

C	A	T	E	G	O	R	I	A	/	S	U	B	S	T	I	T	U	T	O	/	
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--

REGISTRO DE
ENTRADA

0	4	0	0	0	1
---	---	---	---	---	---

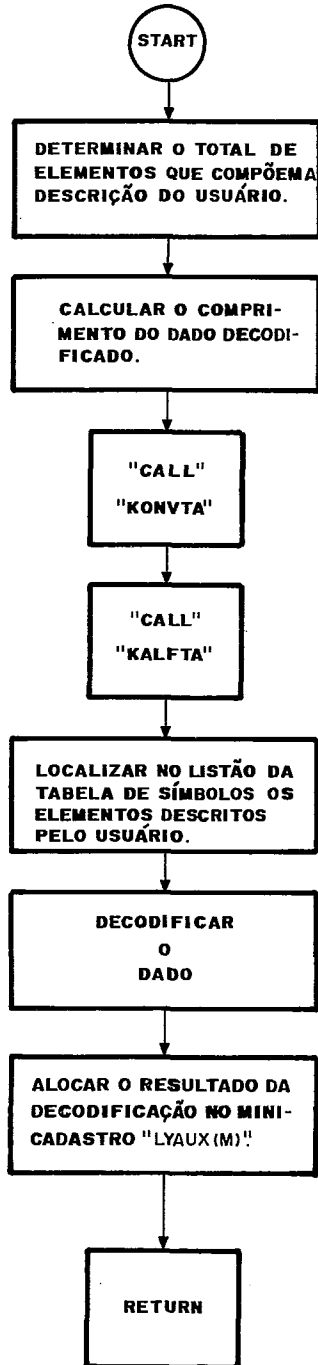
TRECHO CORRESPONDENTE DO MINICADASTRO
"LYAUX(M) "

Fig. 4.3.7.B

4.3.8. -"LEXCOD"

DECODIFICAÇÃO DE DADOS COM ORGANIZAÇÃO DE LISTA INCLUSIVA E, TESSITURA DOS MESMOS, NO MINICADASTRO "LYAUX(M) " .

"LEXCOD"



I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"IBARRA" - área de memória reservada para alo-
car o total de barras (/) utiliza-
das na descrição dos dados.

"IFIBAR(K)" - vetor que contém as localizações
dos delimitadores

"INSTY" - variável utilizada para acondicionar
a posição final de cada elemento des-
crito no registro de entrada.

II - OBJETIVOS:

Cadastrar os dados com organização de LISTA IN-
CLUSIVA, através de:

- . percurso do listão da Tabela de Símbolos, com
o escopo de demarcar a localização dos mes-
mos
- . decodificação dos dados descritos pelo usuá-
rio, depois de coligir os respectivos endere-
ços nas listas gravadas em fita
- . tessitura dos "NÓS" de "LYAUX(M)".

III - DESCRIÇÃO GERAL.

A recessão de "LISCOD" na presença de uma en-

trada com organização de LISTA INCLUSIVA vai persuadir a chamada de "LEXCOD", que envolve as etapas necessárias ao cadastramento do dado decodificado em "LYAUX (M)".

A sinopse de alguns adidos do sistema pode reerigir as características inerentes à descrição de cada dado com organização de "LI", que por definição, podem cingir nenhum, um ou vários elementos do listão residente na Tabela de Símbolos.

Destarte, é justamente através de sindicância da área de trabalho conhecida como "ICOMP(K)", gerada desde a execução de "LISTCK", que se pode granjear informações abrangentes no que tange a compleição da entrada, depois de validada.

Este subprograma retrata o manejo das listas inclusivas alocadas no cabeçalho da fita. De início tem como bônus uma área de memória gerada por "LISCOD", que contém informação referente ao endereço da LISTA solicitada.

Em cada entrada com organização de "LI", depois de localizar seus elementos no listão correspondente, procede à decodificação dos mesmos.

IV - ESQUEMA DE "PROCEDURES".

"LEXCOD" é solicitada pelo sistema após a anuência de "LISCOD" que, ao divisar uma entrada em "LI", modifica seu roteiro de ações. A delonga nessa transferência de e-

xecução perdura somente enquanto termina a avaliação do endereço da LISTA referenciada.

O sistema só pode aceder ao trabalho de "LEXCOD" após a chamada de "LISTCK", que fornece um mapa geral do dado descrito pelo usuário depois de validado, através de "ICOMP(K)".

Ao dissecar o referido vetor, não se pode refugar que a contagem do número de barras delimitadoras, através da variável "IBARRA" e a determinação de seus endereços podem aferir informações importantes para a demarcação de início e fim do elemento que se pretende localizar no listão da Tabela de Símbolos.

Cabe a variável "ICONDO" a tarefa de armazenar o número lógico que representa a posição do elemento procurado na LISTA, o que traduz consideração importante para a decodificação do dado.

Tal como em "LISCOD" a geração do "NÔ" integrante do minicadastro "LYAUX(M)", envolve dois tipos de informação: tamanho e descrição do dado.

Com relação ao tamanho sabe-se que o resultado da decodificação é sempre o múltiplo de 4, que corresponde ao total de elementos envolvidos na descrição de entrada. Assim sendo, é suficiente que se transfira esse número para "LYAUX(M)", através do encadeamento de seus dígitos, nos dois endereços consecutivos ao último caractere já alocado.

Quanto ao dado propriamente dito, a tessitura depende de uma simples transferência de sua representação de de codificada para "LYAUX(M)".

A Fig. 4.3.8.A, que tem em vista a Tabela de Símbolos gerada na Fig. 4.3.0.A apresenta o trecho do minica dastro LYAUX(M)" a ser gravado..

0	8	0	0	0	6	0	0	0	7
---	---	---	---	---	---	---	---	---	---

Fig. 4.3.8.Ā - TRECHO DE UM DOS CADASTROS EM FITA MAGNÉTICA

4.3.9. - "KONVTA"

DESCOMPACTAÇÃO DE NÚMEROS COMO FASE PREPARATÓÓ RIA PARA GRAVAÇÃO DE TAMANHO DOS DADOS NO MINICADASTRO "LYAUX(M)".

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"KAMEL(K)" - vetor que armazena os dígitos nuu méricos que compõem o tamanho do dado.

II - OBJETIVO:

Propiciar a execução da "SUBROUTINE" "KALFTA".

III - ESQUEMA DE "PROCEDURES".

De acordo com as convenções estabelecidas na Tabela de Símbolos, tanto no caso de modo de gravação em "TF" como em "TV", o comprimento do dado ocupa duas posições do cadastro.

Quando "KONVTA" é solicitada, já se encontra armazenado na variável "KEMBY" o total de caracteres do dado. Cabe a este subprograma descompactar seus dígitos e armazená-los no vetor "KAMEL(K)", o que é feito através de dois comandos, apenas:

$$\text{"KAMEL(1) = KEMBY / 10"}$$

$$\text{"KAMEL(2) = KEMBY - KAMEL(1) * 10"}$$

Supondo como exemplo o dado de entrada abaixo:

N	O	M	E		M	A	R	C	O	S	.	A	R	R	U	D	A	.	G	R	I	L	L	O
---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

tem-se:

$$\text{KEMBY = 20}$$

$$\text{KAMEL(1) = 20/10} \quad . \quad . \quad \text{KAMEL(1) = 2}$$

$$\text{KAMEL(2) = 20 - 2 * 10} \quad . \quad . \quad \text{KAMEL(2) = 0}$$

4.3.10. - "KALFTA"

ALOCAÇÃO DE TAMANHO DO DADO NO MINICADASTRO
"LYAUX (M)".

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"IBSE" - variável auxiliar na alocação de tamanho do dado em "LYAUX (M)".

"KAMEL (K)" - vetor que armazena os dígitos numéricos que compõem o tamanho do dado.

II - OBJETIVO:

. selecionar os dígitos alfanuméricos que representam o tamanho do dado.

III - ESQUEMA DE "PROCEDURES".

Existe uma interdependência funcional entre "KONVTA" e "KALFTA", sendo que esta última complementa a execução da primeira, através da alocação de tamanho do dado no minicadastro "LYAUX (M)".

Tudo isso se processa através de uma iteração que compara os elementos de "KAMEL (K)", que contém o tamanho

do dado, com as posições de 1 até 10 do vetor "NUMERO(K)", que retrata os numéricos de 0 até 9.

Após a determinação dos dígitos que representam o comprimento do dado, armazenam-se em "LYAUX(M)" os alfanuméricos correspondentes, através do vetor "NUMALF(K)", explicitado no "DATA".

"LYAUX(I0G) = NUMALF(LECLAI) "

onde "I0G" representa o endereço do caractere que está sendo gravado e "LECLAI", a posição do algarismo no vetor "NUMALF(K)".

4.3.11. - "KTURN"

TRANSFORMAÇÃO DOS ENDEREÇOS GRAVADOS EM FITA
COM A CONFIGURAÇÃO DE ALFANUMÉRICOS PARA NÚMEROS DECIMAIS.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"IMPRES(M)" - área de memória utilizada para armazenar os dígitos do endereço.

"ILHA" - contador de caracteres numéricos.

II - OBJETIVOS:

Burlar o NÚMERO DECIMAL correspondente à seqüên

cia de 4 dígitos alfanuméricos que compõe os endereços dos listões, nos arquivos em fita magnética.

III - DESCRIÇÃO GERAL.

"KTURN" é solicitada pelo sistema sempre que alimentado no equipamento de entrada um dado com organização de "LI" ou "LE",

Sua execução visa em essência a EXPLICITAR O ENDEREÇO do CARACTERE INICIAL ("\$"), das listas da Tabela de Símbolos.

IV - ESQUEMA DE PROCEDURES.

Liminarmente, a ação de "KTURN" limita-se a esgrimir a iteração que vai IDENTIFICAR cada um dos 4 CARACTERES ALFANUMÉRICOS que compõem o endereço a ser manipulado. O primeiro a eclodir é o que retrata o ALGARISMO DE MAIOR ORDEM DE GRANDEZA, devido a posição de "JUCK" no egresso de "LISCOD".

Este é então comparado com os DÍGITOS ALFANUMÉRICOS, alocados no vetor "NUMALF(K)" e, como resultado dos testes, tem-se um diagnóstico completo, que permite armazenar em determinada área de memória o dígito numérico correspondente.

Ao fim da iteração, a ÁREA ÚTIL, representada por "IMPRES(K)", contém os 4 CARACTERES DO ENDEREÇO, expressos em valor absoluto. O número decimal resultante é obtido atra-

vês de "JUCK=IMPRES(1)*10**3 + IMPRES(2)*10**2 +
+ IMPRES(3)*10 + IMPRES(4) "

e passa a ser, então, o conteúdo da variável "JUCK", que durante a execução de "KTURN" percorre os cadastros em fita desde a LOCALIZAÇÃO DO ENDEREÇO DAS LISTAS até o CARACTERE INICIAL DAS MESMAS.

4.3.12. - "MLTLIS"

SINDICÂNCIA E CADASTRAMENTO DE DADOS COM ORGANIZAÇÃO DE LISTA INCLUSIVA, NOS CASOS DE ENTRADA MÚLTIPLA.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"KTITO" - variável que referencia o final da descrição do usuário.

"IMPRIM" - chave de controle empregada para destacar o registro que contém nome do campo.

"LEVY" - variável auxiliar na avaliação de tamanho do dado.

II - OBJETIVOS:

. organizar a ÁREA DE TRABALHO, definida como

- . "ICOMP(K)", sempre que a descrição do dado em "LI" apresentar MAIS DE UM REGISTRO.
- . encaminhar o fluxo para a execução de "LISCOD", com o objetivo de DECODIFICAR o dado e organizar o "NÓ" de "LYAUX(M)".
- . avaliar o TAMANHO DO DADO descrito.

III - DESCRIÇÃO GERAL.

A utilização de "MLTLIS" visa a exterminar os PROBLEMAS DE PROCESSAMENTO criados por dados de entrada com organização de LISTA INCLUSIVA, sempre que adergar um comprimento SUPERIOR A 80 CARACTERES.

Esta ocorrência pode instalar-se quando o elemento cadastrado possui muitos atributos da "LI", o que ocasiona também, como corolário, a presença de vários DELIMITADORES.

A análise deste fato é uma das primeiras preocupações do sistema. Por isso, torna-se exequível a existência de UM MEIO APROPRIADO para corroborar a presença deste caso, antes que qualquer outra execução venha DESNORTEAR o roteiro de ações.

Utiliza-se como paradigma o código caracterizado por um dígito numérico perfurado na 1a. coluna de cada cartão.

A preeminência das atividades, quando confirmada a presença deste caractere, no local correspondente da entrada, é conferida a "MLTLIS", que contém TRATAMENTO ESPECIALIZADO para análise dos elementos da "LI".

IV - ESQUEMA DE "PROCEDURES".

"MLTLIS" seleciona as informações de acordo com a ordem dos registros de entrada. Para isto, testa em primeiro lugar a POSIÇÃO DA CHAVE "IMPRIM", que pode assumir dois estados:

"DESLIGADA" - retratando a presença do 1º registro da SEQUÊNCIA MÚLTIPLA, que contém o código de identificação, o nome do campo e o início da descrição do mesmo.

"LIGADA" - inculcando um qualquer dos registros seguintes que contém a CONTINUAÇÃO DOS ELEMENTOS DA "LI".

No caso de "IMPRIM = 0", que caracteriza o estado "DESLIGADA", o 1º caractere da descrição do dado é fornecido pelo conteúdo do vetor "NIDADO(K)" referente a posição delineada por "KLI". Isto, porque, no programa principal, o referido apontador desloca-se sobre o vetor de entrada, desde o 1º caractere, que representa o código de identificação até o último branco que precede o dado propriamente dito.

Deste modo, para VALIDAR A DESCRIÇÃO DA 1ª. SE

QUÊNCIA, basta percorrer o registro, desde a localização assinalada por "KLI", até o fim.

A alocação na área de trabalho conhecida como "ICOMP(K)" efetiva-se através de uma iteração sincrônica, logo após o acionamento da chave "IMPRIM". Esta, ao assumir o novo estado que corresponde a "IMPRIM = 1" (LIGADA), torna patente a presença de CARTÕES DE CONTINUAÇÃO. Tais registros no caso de uma descrição cingindo muitos elementos da "LI", são validados, através de percurso das colunas 2 até 80, pois a coluna 1 é reservada para o CÓDIGO DE IDENTIFICAÇÃO DA MULTIPLICIDADE.

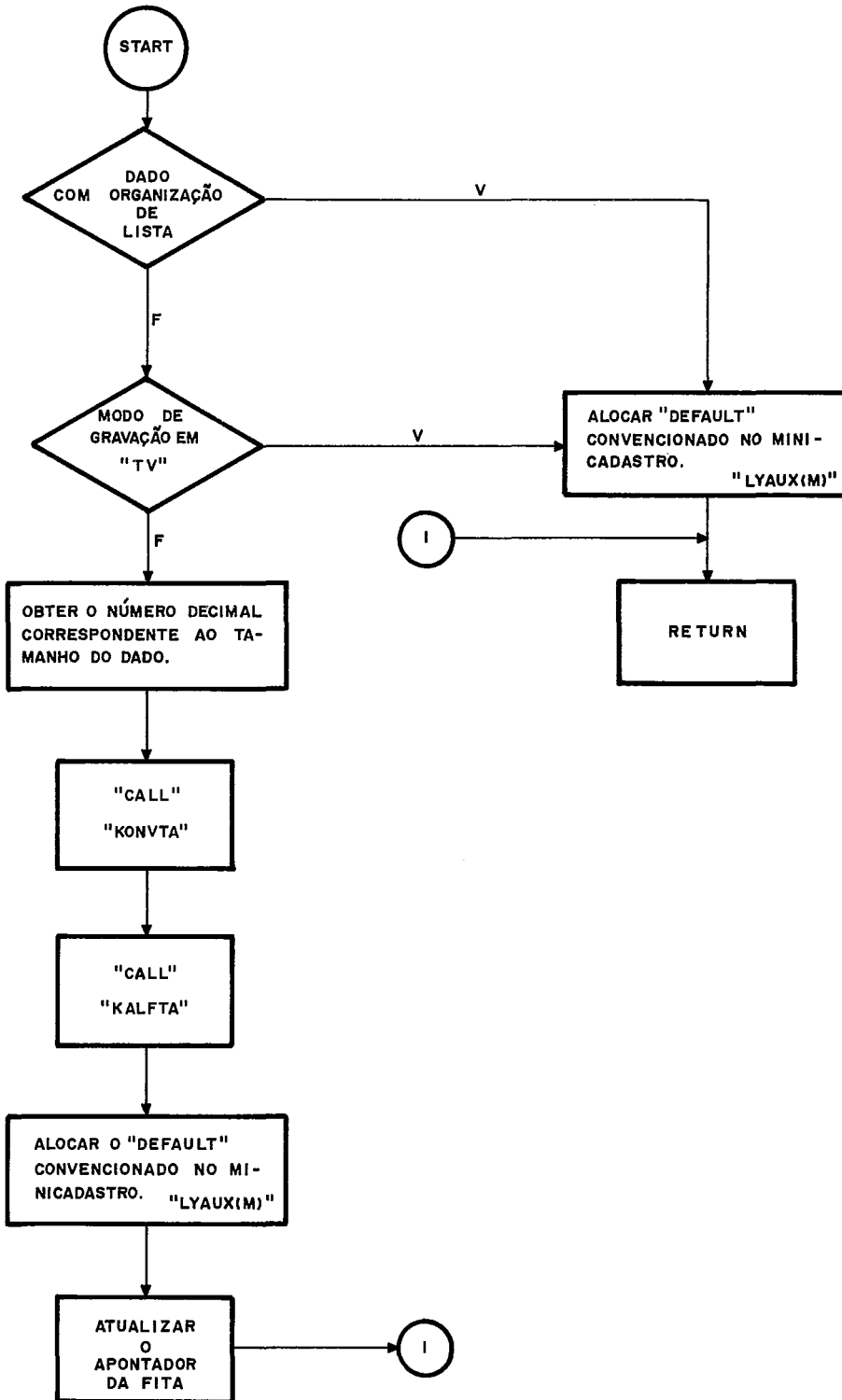
O procedimento que aloca os elementos do 2º, 3º--- enésimo cartão no vetor "ICOMP(K)" é análogo ao anterior e o total de caracteres do dado é fornecido pelo contador "LEVA", que permanece em atividade durante todo o procedimento.

4.3.13. - "KDEFAU"

ESTABELECIMENTO DE "DEFAULT" DO SISTEMA, PARA ATENDER ÀS NECESSIDADES DE CADASTRAMENTO NOS CASOS EM QUE O USUÁRIO NÃO DIPÕE DE TODAS AS INFORMAÇÕES DE ENTRADA.

I - DESCRIÇÃO DA ESTRUTURA DE DADOS.

"KDEFAU"



"JLEMBR" - variável utilizada para assinalar a posição inicial do apontador "JUCK" em cada registro lógico da Tabela de Símbolos.

II - OBJETIVO:

. SUPRIR A AUSÊNCIA DE DADOS, em casos de DESCRISÇÕES DE ENTRADA OMISSAS, através do cadastramento de CÓDIGOS PREESTABELECIDOS.

III - DESCRIÇÃO GERAL.

Considerando o grande número de aplicações deste sistema, pode-se circundar o caso em que determinados elementos de entrada prescindem de informações.

Isto ocorre sempre que a organização do BANCO DE DADOS descortina dificuldades de âmbito geral, em estado latente, que vêm intrinsecar uma abordagem completa dos componentes.

Os BLOCOS FÍSICOS que compõem os cadastros em fita não podem ser organizados de modo incompleto, por isso utiliza-se como paradigma a gravação de VALORES CONVENCIONADOS para DERRISCAR a AUSÊNCIA DE INFORMAÇÕES.

As normas explicitadas abaixo visam sobretudo a OTIMIZAÇÃO DO ESPAÇO DISPONÍVEL EM MEMÓRIA AUXILIAR E cons

tituem o embasamento adotado na geração de dados cuja descrição é omitida pelo usuário.

a) "DEFAULT" DE CÓDIGO "00":

Traduz um dado de ORGANIZAÇÃO INDEPENDENTE, que possui modo de gravação em TAMANHO VARIÁVEL. É aplicado tanto para dados NUMÉRICOS como ALFANUMÉRICOS. Sua estrutura no cadastro em fita é 0200, onde os dois primeiros caracteres (02) compõem o tamanho do dado e os dois últimos (00), o "default" adotado.

b) "DEFAULT" DE CÓDIGO IGUAL AO NÚMERO DE ZEROS CORRESPONDENTE AO TAMANHO DO DADO.

Traduz um dado de ORGANIZAÇÃO INDEPENDENTE, que possui modo de gravação em TAMANHO FIXO. Sua estrutura no cadastro em fita, para um dado cujo comprimento é referido por 07, tem a configuração de 0700000000, onde 07 representa o tamanho do dado e 00000000 o "default" implícito.

c) "DEFAULT" DE CÓDIGO "0000".

Representa um dado com ORGANIZAÇÃO DE LISTA, cuja configuração no cadastro em fita é 040000.

Os dois primeiros caracteres (04) refletem o tamanho do dado e os quatro últimos 0000 o "default" adotado.

IV - ESQUEMA DE "PROCEDURES".

"KDEFAU" é solicitada pelo sistema sempre que a descrição do usuário é constituída apenas pelo nome do dado.

Sua execução não pode prescindir de:

- "KLAUDI" - para validar o nome do campo
- "NAIEFV" - para explicitar o tipo, organização e modo de gravação do dado.

Caracteriza-se a necessidade de recorrência a esse subprograma sempre que, ao deslocar o apontador "KLI" ao longo do REGISTRO DE ENTRADA, ele assinala, apenas, CARACTERES BRANCOS em todas as colunas que seguem o nome do dado.

As ações de "KDEFAU" admitem ser subdivididas em 3 casos:

- 1) DADO INDEPENDENTE COM MODO DE GRAVAÇÃO EM TAMANHO VARIÁVEL.

Neste caso, para sistematizar a informação, no minicadastro "LYAUX(M)", é necessário, apenas, alocar o código Ø2ØØ. Como o comprimento Ø2 é conhecido "a priori", não há necessidade de recorrer a "KONVTA" e "KALFTA", para cadastramento desse tamanho, sendo suficiente fazer:

"LYAUX(IOG) = NUMALF(1) "

"LYAUX(IOG) = NUMALF(3) "

onde "IOG" é o apontador de "LYAUX(M)" e o endereço por ele assinalado é incrementado antes da tessitura de seu conteúdo

no registro lógico.

O "default" $\emptyset\emptyset$ é gravado através da execução de "LYAUX (IOG) = NUMALF (1)", duas vezes sucessivas intercaladas pela atualização de "IOG".

2) DADO INDEPENDENTE COM MODO DE GRAVAÇÃO EM TAMANHO FIXO.

Neste caso deve-se recorrer às informações da Tabela de Símbolos para obter o TAMANHO DO DADO. Este é explicitado através de dois dígitos e pode ser localizado através do deslocamento de "JUCK", duas vezes sucessivas, ao longo dos CADASTROS EM FITA. Isso porque, após a execução de "NAIEFV", sua posição permanece assinalando o 2º caractere da descrição do modo de gravação.

Como a tessitura do cadastro é feita em alfanuéricos, torna-se necessária uma conversão que possibilite obter o número decimal correspondente ao tamanho do dado. Isto é feito através de 2 procedimentos:

a) CONVERSÃO DE CADA DÍGITO ALFANUMÉRICO PARA NUMÉRICO.

Para tanto basta acarear cada um dos dois dígitos através do vetor "NUMALF (K)". Nos casos de similitude, armazena-se o elemento numérico correspondente, fazendo:

"IMPRES (KL) = NUMERO (LE) "

sendo que:

- "KL" pode assumir os valores 1 ou 2 e indica, respectivamente, o 1º ou 2º dígito da representação
- "LE" pode adquirir os valores de 1 até 10 e especifica qual a localização do dígito considerado no vetor "NUMERO(LE)"

Assim sendo, supondo que na Tabela de Símbolos o tamanho do dado descrito seja 7, tem-se a seguinte configuração para "IMPRES(LE)":

"IMPRES(1) = 0"

"IMPRES(2) = 7"

Sendo que sua função é armazenar o número que representa o tamanho do dado em dígitos numéricos.

b) OBTENÇÃO DO NÚMERO DECIMAL QUE REPRESENTA O TAMANHO DO DADO.

De acordo com o que ficou visto, sabe-se que, para suscitar o número decimal correspondente ao tamanho do dado, basta atribuir a cada dígito seu valor relativo isto é:

"KEMBY = IMPRES(1)*10 + IMPRES(2)"

Assim sendo, para gravar em fita o "DEFAULT" correspondente a um dado envolvendo tamanho fixo, aloca-se na fita o valor de "KEMBY" (que representa o tamanho do dado), acompanhado de uma FILEIRA DE ZEROS cujo comprimento é igual ao conteúdo dessa variável.

Para um dado cujo modo de gravação e tamanho foram gerados no 1º módulo como "TF" e "Ø7" respectivamente, verifica-se a seguinte configuração de "LYAUX(M)":

"Ø7ØØØØØØØØ"

Para sistematizar a tessitura dos 2 primeiros dígitos recorre-se a "KONVTA" e "KALFTA", que contêm tratamento especializado para essa função (ver 4.3.9 e 4.3.10).

O "DEFAULT" propriamente dito ("ØØØØØØØØ") é cadastrado através de um laço de "DO" que envolve como PARÂMETROS o ENDEREÇO ATUAL DE "IOG" em "LYAUX(M)" e a posição do último zero da seqüência representada por "IOG + KEMBY - 1".

III - DADO COM ORGANIZAÇÃO DE LISTA.

A gravação em fita no caso de um dado em "LE" utiliza 4 posições do vetor de impressão e, no caso de "LI", qualquer um de seus múltiplos.

Destarte, convencionou-se utilizar como paradigma geral o "DEFAULT" Ø4ØØØØØØ em todos os casos que cingem organização de LISTA.

Aqui também não se faz necessária a recorrência à "KONVTA" e "KALFTA" para montagem dos dois primeiros caracteres (Ø4) representativos de tamanho do dado, pois é suficiente utilizar as 2 atribuições abaixo, que se processam após a atualização de "IOG".

```
"LYAUX(IOG) = NUMALF(1) "
```

```
"LYAUX(IOG) = NUMALF(.5) "
```

O código "0000" é alocado em "LYAUX(M)", através de uma iteração, que utiliza o comando:

```
"LYAUX(IOG) = NUMALF(1) "
```

CAPÍTULO V

CONCLUSÕES

CAPÍTULO VCONCLUSÕES

Esta obra demonstrou que o sistema em estudo realmente funciona na prática.

A análise da viabilidade das técnicas empregadas tornou evidentes as otimizações procuradas que visaram em essência:

- a diminuição da quantidade de memória indispensável para alocar os programas.
- redução do tempo necessário para processamento.
- simplificação das atividades do sistema operacional.

A escolha de arquivos em fita magnética provou constituir a melhor solução. A grande facilidade na movimentação dos cadastros deixou patente a presença de métodos adaptados a uma clientela leiga em processamento de dados.

A fim de continuar este trabalho apresentaria aos estudiosos no assunto sugestões como as seguintes:

I - sem alterar a lógica já estruturada codificar os programas em linguagem "ASSEMBLER", de modo a propiciar uma economia ainda maior no espaço de armazenagem.

II - tratar o arquivo a nível de "BIT" e não de caractere ou valer-se de outros métodos mais avançados tal como o emprego do código de "HUFFMAN", em que o número de "BITS" necessários para codificar cada caractere é inversamente proporcional à frequência do mesmo.

III - Usar técnica de OVERLAY.

IV - admitir como expansão do sistema a presença de organização VARIÁVEL, caracterizada por dados em lista, cujos elementos são desconhecidos "a priori".

B I B L I O G R A F I A

- (1) STONE - Introduction to Computer Organization and Data Structures - Computer Science Series - McGraw-Hill Book Company - 1963.
- (2) KNUTH - The art of computer programming - volume 1 / Fundamental algorithms - Addison - Wesley publishing company - 1969.
- (3) WALNUT, K. Francis - Introduction to computer programming and coding - Englewood Cliffs, New Jersey, 1970.
- (4) McCARTHY X HUMES - Integrated Data Processing Systems - University of Florida - 1970.
- (5) HEHL, Maximilian Emil - Técnicas práticas e eficientes em programação Fortran - Série Ciência de Computação - ed. USP - 1974.
- (6) HARRIS, L. Dale - Fortran Programming - Library of Congress Catalog Card Number 64 - 20503 - 1969.
- (7) GOLDE, Hellmut - Fortran II and IV for Engineers and Scientists - The Macmillan Company, New York - 1967.
- (8) NOLAN, Richard L. - Fortran IV Computing and applications - American Elsevier - 1963.

- (9) McCracken, Daniel D. - Fortran with engineering applications - John Wiley & Sons, INC., New York - 1967.
- (10) Moon, B.A.M. - Computer Programming for Science and Engineering - University of Canterbury, Christchurch - London - 1966.
- (11) Davison, J.F. - Programming for Digital Computers - New York, St. Louis - 1969.
- (12) Hellerman - Digital Computer System Principles - J.P. Nash, Consulting Editor - 1968.
- (13) Carvalho, M. Bruno - Tese de Mestrado - Estudo e Realização de um Sistema de Cadastros Portátil Orientado Para Usuário Leigo em Computação - Parte II - COPPE - 1977.

ANEXO I

MANUAL DO USUÁRIO

ANEXO IMANUAL DO USUÁRIO

Impende ter em mira que de acordo com os objetivos previstos pelo usuário, a fase de MONTAGEM DOS CADASTROS abrange:

1º CASO: GERAÇÃO DE DADOS JÁ INCORPORADOS À TABELA DE SÍMBOLOS.

2º CASO: MODIFICAÇÃO DOS CADASTROS ATRAVÉS DE INSERÇÃO OU RETIRADA DE DADOS.

Este MANUAL visa a fornecer a seqüência de instruções necessária para obtenção de um bom equilíbrio na COMUNICAÇÃO COM O SISTEMA.

Para MAIOR COMODIDADE convencionou-se que uma DESCRIÇÃO DE ENTRADA VÁLIDA deve:

CONDIÇÃO 1 : usar linguagem corrente.

CONDIÇÃO 2 : iniciar ou finalizar por qualquer coluna do cartão.

CONDIÇÃO 3 : prescindir de eventuais preocupações de espaçamento, pois, colunas em branco são liberadas, a não ser no interior de um dado.

CONDIÇÃO 4: utilizar o ponto(.) como veículo de separação nos casos de nomes compostos de mais de uma palavra.

1º CASO : GERAÇÃO DE DADOS JÁ INCORPORADOS AO SISTEMA.

Envolve duas hipóteses suscetíveis de ocorrer:

1a. HIPÓTESE : O DADO NÃO ESTÁ ASSOCIADO ÀS LISTAS JÁ CRIADAS

O usuário deve na sua descrição valer-se das condições 1, 2, 3 e 4.

EXEMPLOS:

BAIR RO	MUDA
ORG AO.EXPEDIDOR	INSTITUTO.FELIX.PACHECO
TEL EF ONE	2563216

2a. HIPÓTESE : O DADO ESTÁ INCORPORADO A UMA DAS LISTAS GERADAS.

Já aqui, além das condições 1, 2, 3 e 4 é necessário observar:

CONDIÇÃO 5 : admitir somente elementos integrantes do LISTÃO, considerados na mesma ordem estabelecida pelo sistema.

CONDIÇÃO 6 : utilizar a barra como delimitador de início e fim de cada elemento.

EXEMPLOS:

CATEGORIA	/SUBSTITUTO/
QUALIFICAÇÃO	/LICENCIATURA.PLENA/CETEG/

Nos casos em que a descrição não fica inteiramente acondicionada em um único cartão deve-se observar além do exposto acima, o seguinte:

CONDIÇÃO 7 : situar o caractere 1 na 1ª. coluna de cada cartão utilizado.

CONDIÇÃO 8 : descrever o nome do campo apenas no 1º cartão.

EXEMPLO:

1MATÉRIAS.QUE.LECIONA /LINGUAGEM.FORTRAN/M
 1ATEMÁTICA/PROCESSAMENTO.DE.DADOS/TELEPROCES
 1SAMENTO/

2º CASO : MODIFICAÇÃO DOS CADASTROS ATRAVÉS DE INSERÇÃO OU RETIRADA DE DADOS.

É de emprego mais raro na prática pois trata de casos em que o usuário pretende aumentar ou diminuir a quantidade de dados cadastrados. Deve atender:

CONDIÇÃO 9 : descrever nome e delimitador (*) do campo, acompanhados de: tipo, organização, modo de gravação e tamanho.

EXEMPLO:

CPF*NUMÉRICO TAMANHO FIXO 11

CATEGORIA*NUMÉRICO LISTA EXCLUSIVA TAMANHO FIXO 04

CONDIÇÃO 10 : utilizar caracteres brancos na representação de ORGANIZAÇÃO e TAMANHO dos dados, sempre que tratar-se de estrutura independente ou modo de gravação variável, respectivamente.

EXEMPLO:

TELEFONE * NUMÉRICO TAMANHO FIXO 07

ENDEREÇO * ALFA TAMANHO VARIÁVEL

CONDIÇÃO 11 : promover a inserção ou retirada das listas acopladas aos dados : em movimento no caso de organização em "LI" ou "LE".

ANEXO II

GLOSSÁRIO DE ERROS

ANEXO IIGLOSSÁRIO DE ERROS

I - ERROS SUSCETÍVEIS DE OCORRER NA GERAÇÃO DA TABELA DE SÍMBOLOS.

ERRO 1 : Caractere estranho no meio das listas.

ERRO 2 : dados inválidos, asterisco na coluna 1.

ERRO 3 : caractere estranho no nome do campo.

ERRO 4 : cartão contendo apenas nome e delimitador de dado.

ERRO 5 : especificação de tipo de dado ausente.

ERRO 6 : especificação de tipo de dado alfa com caracteres a mais no fim ou especificação da organização iniciando com caracteres estranhos.

ERRO 7 : caractere estranho na especificação de tipo de dado alfa.

ERRO 8 : especificação de tipo de dado numérico com caracteres a mais no fim ou especificação da organização iniciando com caracteres estranhos.

ERRO 9 : caractere estranho na especificação de tipo de dado numérico.

ERRO 10 : informação estranha onde era esperada a especificação de tipo de dado.

- ERRO 11 : especificação da organização do dado incompleta e declarações de modo de gravação e tamanho do dado não referenciadas.
- ERRO 12 : especificação da organização do dado em lista inclusiva com caracteres a mais no fim ou declaração do modo de gravação iniciando com caracteres estranhos.
- ERRO 13 : caractere estranho na especificação da organização do dado em lista inclusiva.
- ERRO 14 : especificação da organização do dado em lista exclusiva com caracteres a mais no fim ou declaração do modo de gravação iniciando com caracteres estranhos.
- ERRO 15 : informação estranha onde era esperada a organização do dado.
- ERRO 16 : caractere estranho na especificação de organização do dado em lista exclusiva.
- ERRO 17 : caractere estranho na especificação da organização do dado.
- ERRO 18 : especificações de modo de gravação e tamanho do dado ausentes.
- ERRO 19 : especificação de modo de gravação incompleta e declaração de tamanho do dado não referenciada.
- ERRO 20 : especificação de modo de gravação com caracteres a mais no fim.

- ERRO 21 : caractere estranho na especificação do modo de gravação em tamanho fixo.
- ERRO 22 : especificação de modo de gravação em tamanho variável com caracteres a mais no fim.
- ERRO 23 : caractere estranho na especificação de modo de gravação em tamanho variável.
- ERRO 24 : caractere estranho na especificação do tipo de gravação.
- ERRO 25 : informação estranha onde era esperado o modo de gravação do dado.
- ERRO 26 : dado de tamanho fixo sem especificação do próprio tamanho.
- ERRO 27 : caractere estranho na especificação de tamanho do dado.

II - ERROS SUSCETÍVEIS DE OCORRER NA CRIAÇÃO DOS DADOS.

- ERRO 28 : espaços em branco inseridos no nome do campo.
- ERRO 29 : caractere numérico no nome do campo.
- ERRO 30 : caracteres estranhos inseridos no nome do campo.
- ERRO 31 : nome do campo com caracteres a mais no fim.
- ERRO 32 : dados descritos pelo usuário fora de seqüência.
- ERRO 33 : caractere estranho na descrição de dado alfa.

- ERRO 34 : descrição de dado alfabético incompleta ou contendo espaços em branco.
- ERRO 35 : caractere numérico inserido na descrição de dado alfabético.
- ERRO 36 : descrição de dado numérico incompleta ou contendo espaços em branco.
- ERRO 37 : caracteres estranhos na descrição de dado numérico.
- ERRO 38 : caractere alfabético inserido na descrição de dado numérico.
- ERRO 39 : dado com organização de LISTA com elementos estranhos ao conteúdo da mesma na Tabela de Símbolos.
- ERRO 40 : delimitador ausente na descrição de elementos das listas.
- ERRO 41 : espaços em branco inseridos na descrição dos elementos das listas.
- ERRO 42 : descrição de dado com organização de lista exclusiva sem conteúdo lógico, ou seja; descrevendo mais de um elemento da mesma.
- ERRO 43 : descrição de elementos em lista inclusiva com caractere estranho ao conteúdo da mesma na Tabela de Símbolos.
- ERRO 44 : número de caracteres utilizados pelo usuário na descrição de dado de tamanho fixo não coincide com o descrito na Tabela de Símbolos.

ANEXO III

LISTAGENS

DADOS LIDOS PARA GERAÇÃO DA TABELA DE SÍMBOLOS
DO CADASTRO DE FUNCIONÁRIOS.

NOME* ALFA TAMANHO VARIAVEL
 ENDERECO * ALFA TAMANHO VARIAVEL
 NUMERO* NUMERICO TAMANHO VARIAVEL
 APARTAMENTO* NUMERICO TAMANHO FIXO 05
 BAIRRO * ALFA TAMANHO VARIAVEL
 TELEFONE* NUMERICO TAMANHO FIXO 07
 ZONA.POSTAL * NUMERICO TAMANHO FIXO 07
 M A T R I C U L A * NUMERICO TAMANHO FIXO 06
 IDENTIDADE* NUMERICO TAMANHO FIXO 10
 ORGAO.EXPEDIDOR * ALFA TAMANHO VARIAVEL
 R E G I S T R O * NUMERICO TAMANHO FIXO 07
 TITULO. DE .ELEITOR *NUMERICO T A M A N H O F I X O 07
 Z O N A * N U M E R I C O TAMANHO FIXO 02
 SECCAO* N U M E R I C O TAMANHO FIXO 04
 CARTEI RA. PRO FISSIONAL * NUMERICO TAMANHO FIXO 07
 SERIE* NUMERICO TAMANHO FIXO 04
 C P F * NUMERICO TAMANHO FIXO 11
 ESTADO.CIVIL * NUMERICO LISTA EXCLUSIVA TAMANHO FIXO 04
 DEPENDENTES * NUMERICO TAMANHO FIXO 02
 TEMPO.DE.SERVICO* NUMERICO TAMANHO FIXO 02
 CATEGORIA* NUMERICO LISTA EXCLUSIVA TAMANHO FIXO 04
 SITUACAO * NUMERICO LISTA EXCLUSIVA TAMANHO FIXO 04
 FUNCAO* NUMERICO LISTA EXCLUSIVA TAMANHO FIXO 04
 CARGO* N U M E R I C O L I S T A E X C L U S I V A TAMANHO FIXO 04
 QUADRO * NUMERICO LISTA EXCLUSIVA TAMANHO F I X O 04
 ACUMULACAO* NUMERICO TA MA NH O F I X O 06
 QUALIFICACAO * NUMERICO LISTA INCLUSIVA TAMANHO VARIAVEL
 A R E A * NUMERICO LISTA EXCLUSIVA TAMANHO FIXO 04
 MATERIAS.QUE.LECIONA * NUMERICO LISTA INCLUSIVA TAMANHO VARIAVEL
 LOTACAO* NUMERICO LISTA EXCLUSIVA TAMANHO FIXO 04

TABELA DE SÍMBOLOS DO CADASTRO DE FUNCIONÁRIOS

NOME*00A00TV000000ENDERECO*00A00TV000000NUMERO*00N00TV000000
 APARTAMENTO*00N00TF050000BAIRRO*00A00TV000000TELEFONE*00N00T
 F070000ZONA.POSTAL*00N00TF070000MATRICULA*00N00TF060000IDENT
 IDADE*00N00TF100000ORGAO.EXPEDIDOR*00A00TV000000REGISTRO*00N
 00TF070000TITULO.DE.ELEITOR*00N00TF070000ZONA*00N00TF020000S
 ECCAO*00N00TF040000CARTEIRA.PROFISSIONAL*00N00TF070000SERIE*
 00N00TF040000CPF*00N00TF110000ESTADO.CIVIL*00NLETF040700\$DEPE
 NENTES*00N00TF020000TEMPO.DE.SERVICO*00N00TF020000CATEGORIA
 *00NLETF040741SITUACAO*00NLETF040776FUNCAO*00NLETF040883CARG
 O*00NLETF041978QUADRO*00NLETF042010ACUMULACAO*00N00TF060000Q
 UALIFICACAO*00NLITV002021AREA*00NLETF042183MATERIAS.QUE.LECI
 DNA*00NLITV002226LOTACAO*00NLETF047037 \$SOLTEIRO/CASADO/V
 IUVO/DESQUITADO/ \$SUBSTITUTO/CONTRATADO/EFETIVO/ \$LICE
 NCA.MEDICA/LICENCA.DE.GESTANTE/LICENCA.PREMIO/LICENCA.SEM.VE
 NCIMENTO/APOSENTADO/ATIVO/READAPTADO/ \$DIRETOR/ASSESSOR/
 COORDENADOR.GERAL/COORDENADOR.DE.TURNO/COORDENADOR.DE.DISCIP
 LINA/PROFESSOR/AGENTE.DE.PESSOAL/ORIENTADOR.PEDAGOGICO/ORIEN
 TADOR.EDUCACIONAL/CHEFE.DE.SECRETARIA/INSPETOR/SECRETARIO/SE
 RVENTE/OFICIAL.ADMINISTRATIVO/ESCRITURARIO/DATILOGRAFO/MEREN
 DEIRA/PORTEIRO/ZELADOR/CONTINUO/ALMOXERIFE/PEDEIRO/BOMBEIRO
 /PINTOR/TRABALHADOR/ASCENSOR/TECNICO.ADMINISTRATIVO/ATENDENT
 E/COZINHEIRA/FUNDIDOR/MESTRE/AUXILIAR.DE.ESTATISTICA/MARROEI
 RO/VIGIA/OPERARIO.DE.MAQUINAS.PESADAS/ENFERMEIRO.AUXILIAR/AR
 QUIVISTA/ENFERMEIRO/CARPINTEIRO/COPEIRO/MOTORISTA/FERREIRO/S
 ERVICAL/IMPRESSOR/FISCAL.DE.OBRA/TECNICO.DE.LABORATORIO/ALFA

IATE/JARDINEIRO/ELETRICISTA/MEDICO/TECNICO.DE.TV/TELEFONISTA
 /PROTETICO/DENTISTA/OBSTETRIZ/CHEFE.DE.SERVICO.DE.SAUDE.ESCO
 LAR/CHEFE.DE.COMUNICACAO.DE.SECRETARIA/AUXILIAR.DE.PORTARIA/
 ZELADOR/ESCREVENTE/REVISOR/CALCULISTA/ASSESSOR.ADMINISTRATIV
 O/AUXILIAR.GRAFICO/PREPARADOR.DE.ORIGINAIS/PEDREIRO/ELETRICI
 STA/BIBLIOTECARIO/CARPINTEIRO/MARCINEIRO/MECANICO/SOLDADOR/A
 RMADOR/MAQUINISTA/AJUOANTE.DE.MAQUINAS/ENSAGEIRO/ASSESSOR.D
 E.PESQUISAS/ESCRITURARIO/FISCAL/SUPERVISOR.DE.ENSINO/ §CO
 MISSAO/EXTRA.CLASSE/NULO/ §1/2/3/ §LICENCIATURA.PLENA/
 LICENCIATURA.CURTA/CETEG/PROJETO.3/MESTRADO/DOCTORADO/CURSO.
 DE.DIRETOR/NIVEL.SUPERIOR/PRIMEIRO.GRAU.INCOMPLETO/PRIMEIRO.
 GRAU/SEGUNDO.GRAU/ §EDUCACAO.GERAL/FORMACAO.ESPECIAL/NULO
 / §ARTES.PLASTICAS/ATUALIDADES/ANALISES.DE.BALANCOS/ANALI
 SE.DE.CIRCUITOS/ANATOMIA.E.FISIOLOGIA/ARTES.APLICADAS/AGRICU
 LTURA/ADMINISTRACAO.ESCOLAR/ARTES.INDUSTRIAIS/ARTES.DRAMATIC
 AS/APARELHOS.ORTODONICOS/ADMINISTRACAO/ANALISES.QUIMICAS/ECO
 NOMIA.RURAL/ARTES.GRAFICAS/ARTESANATO/ARTES.FEMININAS/ANATOM
 IA.E.ESCULTURA.DENTAL/ARTES/BIOLOGIA/BIOQUIMICA/BIOTECNICA/B
 IBLIGTECONOMIA.E.ARQUIVISTICA/BIOLOGIA.E.PROGRAMA.DE.SAUDE/B
 IOLOGIA.EDUCACIONAL.E.HIGIENE/BASES.ARQUITETONICAS.DE.CONSTR
 UCAO/BIOLOGIA.APLICADA/BASES.DE.ESTRUTURA.DA.CONSTRUCAO/CIEN
 CIAS.FISICAS.E.BIOLOGICAS/CONTABILIDADE.GERAL/CONTABILIDADE.
 PUBLICA/CONTABILIDADE.E.CUSTOS/CONTABILIDADE.INDUSTRIAL/CIEN
 CIAS.NATURAIS/COMPLEMENTOS.DE.MATEMATICA/CRIACOES/CULTURAS/C
 ONTABILIDADE.E.ECONOMIA.DO.MERCADO/CONTABILIDADE.BANCARIA/CI

ENCIAS. E. PROGRAMA. DE. SAUDE/CONSTRUCOES. E. INSTALACOES. RURAIS/
 CORTE. E. COSTURA/CONSTRUCAO/COMUNICACAO. EM. LINGUA. PORTUGUESA/
 CURRICULOS. E. PROGRAMAS/CONTABILIDADE. COMERCIAL/DESENHO/DIREI
 TO. USUAL/DIDATICA/DIDATICA. GERAL/DIDATICA. ESPECIAL/DESCRITIV
 A/DATILOGRAFIA/DIREITO. E. LEGISLACAO/DESENHO. TECNICO/DESENHO.
 DE. ESQUADRIAS/DESENHO. DE. INSTALACOES. HIDRAULICAS/DESENHO. E. T
 OPOGRAFIA/DESENHO. PROJETIVO/DESENHO. ARQUITETONICO/DETALHE. DE
 . CONCRETO/DIDATICA. E. PRATICA. DE. ENSINO/DIDATICA. GERAL. E. ESPE
 CIAL/DINAMICA. DE. GRUPO/ESTUDOS. SOCIAIS/EDUCACAO. MORAL. E. CIVI
 CA/EDUCACAO. FISICA/ECONOMIA. DOMESTICA/EDUCACAO. ARTISTICA/EST
 ATISTICA/EDUCACAO. PARA. O. LAR/ESTUDOS. REGIONAIS/ENFERMAGEM. CI
 RURGICA/ESTUDO. DO. MEIO/EDUCACAO. MUSICAL/ENSINO. RELIGIOSO/EST
 RUTURA. E. ANALISE. DE. BALANCO/ECONOMIA. E. MERCADO/ESTENOGRAFIA/
 ENFERMAGEM. MEDICA/ENFERMAGEM. NEURO. PSIQUIATRICA/ENFERMAGEM. M
 ATERNO. INFANTIL/ELEMENTOS. DE. FISIOTERAPIA/ELEMENTOS. DE. LOGOP
 EDIA/ELEMENTOS. DE. MUSICOTERAPIA/ELEMENTOS. DE. TERAPIA. OCUPACI
 ONAL/ESTUDO. DOS. MEIOS. DE. COMUNICACAO/ELETRICIDADE/ELETROTECN
 ICA/EXPRESSAO. ARTISTICA/ESTRUTURA/ESTABILIDADE. DE. CONSTRUCAO
 /ESTAGIO. SUPERVISIONADO/EXPRESSAO. ORAL. E. ESCRITA/ECONOMIA/ES
 TRUTURA. E. FUNCIONAMENTO. DO. ENSINO. . DE. PRIMEIRO. E. SEGUNDO. GRA
 U/FISICA/FISICA. APLICADA/FUNDAMENTOS. DA. EDUCACAO/FISICOQUIMI
 CA/FILOSOFIA/FUNDAMENTOS. BIOLOGICOS/FUNDAMENTOS. SOCIOLOGICOS
 /FUNDAMENTOS. FILOSOFICOS/FUNDAMENTOS. PSICOLOGICOS/FUNDAMENTOS
 S. HISTORICOS/FUNDAMENTOS. DE. ENFERMAGEM/FUNDAMENTOS. DA. REABIL
 ITACAO/FOLCLORE/GEOGRAFIA/GEOCIENCIAS/HISTORIA/HIGIENE/HISTO

RIA. ADMINISTRATIVA/HIGIENE. E. PUERICULTURA/HIGIENE. DE. ALIMENTOS/HISTORIA. DA. ARTE/HISTORIA. DA. EDUCACAO. BRASILEIRA/HIGIENE. E. SEGURANCA. NO. TRABALHO. E. PROGRAMA. DE. SAUDE/INGLES/INSTALACOES. ELETRICAS/INTEGRACAO. SOCIAL/INTEGRACAO. PROFISSIONAL/IRRIGACAO. E. DRENAGEM/INFORMACAO. OCUPACIONAL/INGLES. TECNICO/INTRODUCAO. A. TEORIA. DOS. CONJUNTOS. E. NOCOES. DE. ESTRUTURA. MATEMATICA/INICIACAO. AS. CIENCIAS/LATIM/LEGISLACAO. APLICADA/LINHA. DE. TRANSMISSAO/LITERATURA. INFANTIL/LEGISLACAO/LOGOPEDIA/MATEMATIC A/MEDIDAS. E. AVALIACOES/MATERIAIS. E. REVESTIMENTOS/MATEMATICA. E. DESENHO/MEDIDAS/METODOLOGIA. DA. LINGUAGEM/MATEMATICA. E. ESTADISTICA/METODOLOGIA. DA. REDACAO. OFICIAL/MATEMATICA. COMERCIAL/MATEMATICA. FINANCEIRA/MATEMATICA. COMERCIAL. E. FINANCEIRA/MECANOGRAFIA/MECANOGRAFIA. E. PROCESSAMENTO. DE. DADOS/MICROBIOLOGIA/MATEMATICA. APLICADA/MATEMATICA. DE. CONSTRUCAO/MATERIAIS. E. COMPONENTES/MECANICA/MAQUINAS. E. INSTALACOES/MEDIDAS. ELETRICAS/MUSEOLOGIA/MARZENARIA/METODOLOGIA. DO. ENSINO. DE. PRIMEIRO. GRAU/MERCADOLOGIA/NOCOES. DE. COMERCIO/NAVEGACAO/NOCOES. DE. TEORIA. LITERARIA. E. ESTADISTICA/ORGANIZACAO. SOCIAL. E. POLITICA. BRASILEIRA/ORGANIZACAO. E. NORMAS/OFICINAS/ORGANIZACAO. VOCACIONAL/ORGANIZACAO. E. TECNICAS. COMERCIAIS/ORGANIZACAO. E. TECNICA. PROFISSIONAL/PORTUGUES/PUBLICIDADE/PROGRAMA. DE. SAUDE/PROCESSAMENTO. DE. DADOS/PROCESSOS. INDUSTRIAIS/PRODUCAO. MECANICA/PRATICA. DE. ENSINO/PRATICAS. AGRICOLAS/PROTESE/PROJETO. DE. DECORACAO/PSICOLOGIA/PSICOPATOLOGIA/PETROQUIMICA/PRINCIPIOS. E. METODOS. DE. ADMINISTRACAO/PROTECAO. DO. SISTEMA. ELETRICO/PESQUISAS. E. MEDIDA

S. EM. EDUCACAO/PRINCIPIOS. E. METODOS. EM. ORIENTACAO. EDUCACIONAL
 /PRINCIPIOS. E. TECNICAS/PRINCIPIOS. E. METODOS. DE. SUPERVISAO. ES
 COLAR/PERSPECTIVA/QUIMICA. GERAL/QUIMICA. ORGANICA/QUIMICA. INO
 RGANICA/QUIMICA. APLICADA/REDACAO. E. EXPRESSAO/REABILITACAO. NE
 UROLOGICA/REABILITACAO. PSIQUIATRICA/REABILITACAO. CIRURGICA/R
 ESISTENCIA. DOS. MATERIAIS/RECREACAO. INFANTIL/REDE. DE. DISTRIBU
 ICAD/REDACAO. COMERCIAL/REDACAO. E. EXPRESSAO/SAUDE. PUBLICA/SOC
 ORROS. URGENTES/SISTEMA. DE. COMUTACAO/SOLOS/SUPERVISAO. ESCOLAR
 /SERVICO. DE. ORIENTACAO. EDUCACIONAL/SUBESTACAO/SOCIOLOGIA. DA.
 EDUCACAO/TRABALHOS. MANUAIS/TECNICAS. DE. SECRETARIADO/TECNICAS
 .PROFISSIONAIS/TECNICAS. COMERCIAIS/TECNICAS. ORCAMENTARIAS/TE
 CNICAS. GERAIS/TECNICAS. DE. TURISMO/TECNICAS. INDUSTRIAIS/TECNI
 CAS. ORCAMENTARIAS. E. CONTABILIDADE. PUBLICA/TECNICAS. DE. SECRET
 ARIADO. E. ESTENOGRAFIA/TECNICAS. MEDICAS. DE. LABORATORIO/TECNIC
 AS. E. EXAMEPSICOLOGICO/TECNICAS. GERAIS. DE. LABORATORIO/TECNICA
 .DOS. MATERIAIS/TRABALHOS. EM. COURO/TECNOLOGIA. DO. NAVIO/TERMOD
 INAMICA/TRANSFORMADOR/TOPOGRAFIA/TEORIA. E. PRATICA. DA. EDUCACA
 O/ZOOTECNICA SAMARO. CAVALCANTE/ANDRE. MAUROIS/ANTONIO. PRAD
 O. JUNIOR/BAHIA/BARAO. DO. RIO. BRANCO/BERNARDO. SAYAO/BRIGADEIRO
 .SCHORCHT/CARMELA. OUTRA/CELESTINO. SILVA/CENTRAL. DO. BRASIL/PR
 OFESSOR. CLOVIS. MONTEIRO/COLLECCHIO/PROFESSOR. DAL TRO. SANTOS/E
 CA. DE. QUEIROZ/FERREIRA. VIANA/FRANCISCO. JOBIM/FREIRE. ALEMAC/G
 OMES. FREIRE. DE. ANDRADA/HEITOR. LIRA/IGNACIO. AZEVEDO. AMARAL/IN
 FANTE. DON. HENRIQUE/INSTITUTO. DE. EDUCACAO. DE. CAMPO. GRANDE/INS
 TITUTO. DE. EDUCACAO. DO. RIO. DE. JANEIRO/JOAO. ALFREDO/MARECHAL

. JOAO. BATISTA. DE. MATOS/PROFESSOR. JOSE. ACIOLI/JULIA. KUBITSCH
 C/PREFEITO. MENDES. DE. MORAES/MEXICO/OLAVO. BILAC/PAULO. DE. FRON
 TIM/PEDRO. ALVARES. CABRAL/PROFESSOR. RAJA. GABAGLIA/REPUBLICA. D
 O. PERU/SOUZA. AGUIAR/VISCONDE. DE. CAIRU/VISCONDE. DE. MAUA/ANEXO
 . DO. VISCONDE. DE. MAUA/WASHINGTON. LUIZ/HENCESLAU. BELLO/JOAO. CA
 NDIDO. BRASIL/BARRA. MANSA/NILDO. PECANHA/PADRE. MELLO/ESTADUA. DE
 . CABO. FRIO/ANTONIO. SARLO/LICEU. DE. HUMANIDADES. DE. CAMPOS/PROF
 ESSOR. ALDO. MUYLAERT/MARIA. ZULPIRA. TORRES/TOBIAS. TOSTES. MACHA
 DO/BARAO. DE. MAUA/ROBERTO. SILVEIRA/MARECHAL. DEODORO. DA. FONSEC
 A/LUIZ. REID/ESTADUAL. DE. MAGE/DEODATO. LINHARES/FLAVIO. RIBEIRO
 . DE. REZENDE/BRIGADEIRO. JOAO. NEPOMUCENO. CASTRICO/PROFESSOR. A
 RMANDO. GONCALVES/HENRIQUE. LAGE/AURELINO. LEAL/PROFESSOR. ISMAE
 L. COUTINHO/ESTADUAL. DE. NOVA. FRIBURGO/INSTITUTO. DE. EDUCACAO. D
 E. NOVA. IGUACU/ENGENHEIRO. MARIO. MOURA. BRASIL/ESTADUAL. DE. PETR
 OPOLIS/ESTADUAL. DA. CASCATINHA/JOSE. DE. LANES. DANTAS. BRANDAO/M
 ARECHAL. SOUZA. DANTAS/RUI. GUIMARAES. DE. ALMEIDA/ESTADUAL. DE. SA
 O. FIDELIS/CLELIA. NANJI/SAPUCAIA/EDMUNDO. BITTENCOURT/THEODORI
 CO. FONSECA/DEPUTADO. LUIZ. PINTO/PROFESSOR. MANUEL. MARINHO/

DADOS LIDOS PARA GERAÇÃO DO PRIMEIRO FUNCIONÁRIO

NOME FRIS.SOBEL.PETEK
 ENDEREÇO RUA.GAY.LUSSAC
 NUMERO 249
 APARTAMENTO 00602
 BAIRRO JOA
 TELEFONE 2573512
 ZONA.POSTAL 0020000
 MATRICULA 130456
 IDENTIDADE 0001302288
 ORGAO.EXPEDIDOR INSTITUTO.FELIX.PACHECO
 REGISTRO 0002024
 TITULO.DE.ELEITOR 0120231
 ZONA 14
 SECCAO 0131
 CARTEIRA.PROFISSIONAL 0025015
 SERIE 0315
 CPF 02640078704
 ESTADO.CIVIL /CASADO/
 DEPENDENTES 01
 TEMPO.DE.SERVICO 10
 CATEGORIA /EFETIVO/
 SITUACAO /LICENCA.DE.GESTANTE/
 FUNCAO /COORDENADOR.GERAL/
 CARGO /COMISSAO/
 QUADRO /2/
 ACUMULACAO 125365
 QUALIFICACAO /LICENCIATURA.PLENA/MESTRADO/
 AREA /EDUCACAO.GERAL/
 MATERIAS.QUE.LECIONA /ADMINISTRACAO.ESCOLAR/ARTES.INDUSTRIAIS/
 LDTACAO /ANDRE.MAUROIS/

DADOS LIDOS PARA GERAÇÃO DO SEGUNDO FUNCIONÁRIO

NOME JORO.SERE.BELEL
 ENDEREÇO
 NUMERO 8026
 APARTAMENTO 00103
 BAIRRO BARRA.DA.TIJUCA
 TELEFONE 3992215
 ZONA.POSTAL 0019000
 MATRICULA 123235
 IDENTIDADE 0001139804
 ORGAO.EXPEDIDOR INSTITUTO.FELIX.PACHECO
 REGISTRO 0001032
 TITULO.DE.ELEITOR 0002050
 ZONA 26
 SECCAO 0121
 CARTEIRA.PROFISSIONAL
 SERIE 0420
 CPF 24002563204
 ESTADO.CIVIL /VIUVO/
 DEPENDENTES 05
 TEMPO.DE.SERVICO 12
 CATEGORIA /CONTRATADO/
 SITUACAO /ATIVO/
 FUNCAO /PROFESSOR/
 CARGO
 QUADRO /1/
 ACUMULACAO 732546
 QUALIFICACAO /LICENCIATURA.PLENA/DOCTORADO/
 AREA /FORMACAO.ESPECIAL/
 MATERIAS.QUE.LECIONA
 LDTACAO /AMARO.CAVALCANTE/

DADOS LIDOS PARA GERAÇÃO DO TERCEIRO FUNCIONÁRIO

NOME WANKI, BERLAU
 ENDEREÇO RUA, FROTA, BERGMANN
 NUMERO 126
 APARTAMENTO 00302
 BAIRRO GAVEA
 TELEFONE 2389797
 ZONA, POSTAL 0010000
 MATRICULA 295138
 IDENTIDADE 0000496184
 ORGAO, EXPEDIDOR INSTITUTO, FELIX, PACHECO
 REGISTRO 0002043
 TITULO, DE, ELEITOR 0003863
 ZONA 13
 SECCAO 0181
 CARTEIRA, PROFISSIONAL 0032145
 SERIE 0211
 CPF 03214263203
 ESTADO, CIVIL /SOLTEIRO/
 DEPENDENTES 03
 TEMPO, DE, SERVICO 12
 CATEGORIA /SUBSTITUTO/
 SITUACAO /LICENCA, PREMIO/
 FUNCAO /DIRETOR/
 CARGO /EXTRA, CLASSE/
 QUADRO /3/
 ACUMULACAO 281357
 QUALIFICACAO /CETEG/PROJETO, 3/
 AREA /EDUCACAO, GERAL/
 IMATERIAS, QUE, LECIONA /ANALISES, DE, BALANCOS/ANALISE, DE, CIRCUIT
 IDS/ARTES, APLICADAS/AGRICULTURA/
 #
 LOTACAO /BRIGADEIRO, SCHORCHT/
 ?

GRAVAÇÃO EM FITA MAGNÉTICA CORRESPONDENTE AO
CADASTRAMENTO DOS TRÊS FUNCIONÁRIOS

/ 16F

RIS. SOBEL. PETEK | 14 RUA. GAY. LUSSAC | 03249 | 0500602 | 03JON | 072573512 | 070
020000 | 06130456 | 100001302288 | 23 INSTITUT C. FELIX. PACHECO | 070002024
07012023 | 10214 | 040131 | 070025015 | 040315 | 102640078704 | 04002020 | 1021
040003 | 040002 | 040003 | 040001 | 040002 | 06125365 | 0800010005 | 040001 | 08000
80009 | 040002 | 15 JORO. SERE. BELEL | 0200 | 048026 | 0500103 | 15 BARRA. DA. TIJU
CA | 073992215 | 070019000 | 06123235 | 100001139804 | 23 INSTITUTO. FELIX. PA
CHECO | 070001032 | 070002050 | 0226 | 04012 | 070000000 | 040420 | 112400256320
4040003 | 0205 | 0212 | 040002 | 040006 | 040006 | 040000 | 040001 | 06732546 | 0800010
006 | 040002 | 040000 | 040001 | 12 WANKI. BERLAU | 18 RUA. FROTA. BERGMANN | 03126
0500302 | 05 GAVEA | 072389797 | 070010000 | 06295138 | 100000496184 | 23 INSTIT
UTO. FELIX. PACHECO | 070002043 | 070003863 | 0213 | 040181 | 070032145 | 040211 |
1103214263203 | 040001 | 0203 | 0212 | 040001 | 040003 | 040001 | 040002 | 040003 | 062
81357 | 0800030004 | 040001 | 160003000400060007 | 040007

PROGRAMA FONTE

```

SUBROUTINE ISAI(L)
C  SUBROTINA PARA GRAVACAO DE CARACTERES INDIVIDUAIS EM FITA
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDOC
?O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),NU
*MLIS,LISEND(100),KAM,ICART(80),KLOST,KAD,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMEÇO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
I=I+1
IF(I.LE.ICONT*IBLOC)GO TO 1
2 INIC=ICONT*IBLOC-(IBLOC-1)
IFIM=ICONT*IBLOC
IF(IPEG.NE.0) GO TO 44
45 ICONT=ICONT+1
1 IF(IPEG.NE.0) GO TO 2
40 IMPRI(I)=L
44 RETURN
END

```

```

SUBROUTINE MLTIMP(N)
C  GRAVACAO EM FITA ADAPTADA PARA QUANTIDADES MAIORES DE DADOS
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDOC
*O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMEÇO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
KEPE=0
1000 I=I+1
IF(I.LE.ICONT*IBLOC)GO TO 1001
1002 INIC=ICONT*IBLOC-(IBLOC-1)
IFIM=ICONT*IBLOC
IF(IPEG.NE.0) GO TO 1004
1005 ICONT=ICONT+1
1001 IF(IPEG.NE.0) GO TO 1002
1006 KEPE=KEPE+1
IMPRI(I)=MONTE(KEPE)
IF(KEPE.LT.N) GO TO 1000
1004 RETURN
END

```

```

SUBROUTINE IERRO(N)
C  SUBROTINA PARA IMPRESSAO DE ERROS
DIMENSION IFRAS(2000)
DATA LETUR/' ?'/
DATA IFRAS/'ERRO',' 1 =',' CAR','ACTE','RE E','STRA','NHO '
*, 'NO M','EIO ','DAS ','LIST','AS ',' ?','ERRO',' 2 =','
*DAD','OS I','NVAL','IDOS','AST','ERIS','CO N','A CO','LUNA

```


* 1 ? , ERRO , 3 = , CAR , ACTE , RE E , STRA ,
 * NHO , NO N , OME , DO C , AMPO , ? , ERRO , 4 = , CA
 * R , TAO , CONT , ENDO , APE , NAS , NOME , E D , FLIM ,
 * ITAD , OR D , E DA , DO ? , ERRO , 5 = , ESP , EC
 * IF , ICAC , AO D , E TI , PO D , E DA , DO A , USEN , TE
 * ? , ERRO , 6 = , ESP , ECIF , ICAC , AO D , E TI , P
 * O D , E DA , DO A , LFA , COM , CARA , CTER , ES A , MAI
 * , S NO , FIM , OU , ESPE , CIFI , CACA , O DA , ORG ,
 * ANIZ , ACAO , INI , CIAN , DO C , OM C , ARAC , TERE , S E
 * S , TRAN , HOS ? , ERRO , 7 = , CAR , ACTE , RE E ,
 * STRA , NHO , NA E , SPEC , IFIC , ACAO , DE , TIPO , D
 * E , DAD , O AL , FA ? , ERRO , 8 = , ESP , ECIF
 * , ICAC , AO D , E TI , PO D , E DA , DO N , UMER , ICO , C
 * OM , CARA , CTER , ES A , MAI , S NO , FIM , OU , ESPE
 * , CIFI , CACA , O DA , ORG , ANIZ , ACAO , INI , CIAN ,
 * DO C , OM C , ARAC , TERE , S ES , TRAN , HOS ? , ERR
 * O , 9 = , CAR , ACTE , RE E , STRA , NHO , NA E , SPEC ,
 * IFIC , ACAO , DE , TIPO , DE , DADO , NUM , ERIC , O
 * ? , ERRO , 10 = , IN , FORM , ACAO , EST , RANH
 * , A ONI , DE E , RA E , SPER , ADA , A ES , PECI , FICA , C
 * AO , DE T , IPO , DE D , ADO ? , ERRO , 11 = , ES
 * , PECI , FICA , CAO , DA O , RGAN , IZAC , AO D , O DA ,
 * DO I , INCOM , PLET , IA E , DECL , ARAC , OES , DE M , ODO
 * , DE G , RAVA , CAO , E TA , MANH , O DO , DAD , O NA ,
 * O RE , FERE , INCIA , DAS ? , ERRO , 12 = , ES , PE
 * CI , FICA , CAO , DA O , RGAN , IZAC , AO D , O DA , DO E
 * , M LI , STA , INCL , USIV , A CO , M CA , RACT , ERES ,
 * A M , AIS , NO F , IM O , U DE , CLAR , ACAO , DO , MODO
 * , DE , GRAV , ACAO , INI , CIAN , DO C , OM C , ARAC ,
 * TERE , S ES , TRAN , HOS ? , ERRO , 13 = , CA , RAC
 * T , ERE , ESTR , ANHO , NA , ESPE , CIFI , CACA , O DA ,
 * ORG , ANIZ , ACAO , DO , DADO , EM , LIST , A INI , CL
 * US , IVA ? , ERRO , 14 = , ES , PECI , FICA , CAO
 * , DA O , RGAN , IZAC , AO D , O DA , DO E , M LI , STA , E
 * XCL , USIV , A CO , M CA , RACT , ERES , A M , AIS , NO
 * , FIM , OU D , ECLA , RACA , O DO , MOD , O DE , GRA ,
 * VACA , O IN , ICIA , INDO , COM , CARA , CTER , ES E , STR
 * A , NHOS ? , ERRO , 15 = , IN , FORM , ACAO , EST ,
 * RANH , A ONI , DE E , RA E , SPER , ADA , A OR , GANI , IZA
 * CA , O DO , DAD , O ? , ERRO , 16 = , CA , RACT
 * , ERE , ESTR , ANHO , NA , ESPE , CIFI , CACA , O DE ,
 * ORG , ANIZ , ACAO , DO , DADO , EM , LIST , A EX , CLUS
 * , IVA ? , ERRO , 17 = , CA , RACT , ERE , ESTR ,
 * ANHO , NA , ESPE , CIFI , CACA , O DA , ORG , ANIZ , ACA
 * O , DO , DADO , ? , ERRO , 18 = , ES , PECI , FICA ,
 * COES , DE , MODO , DE , GRAV , ACAO , E T , AMAN , HO
 * D , O DA , DO A , USEN , TES ? , ERRO , 19 = , ES
 * , PECI , FICA , CAO , DE M , ODO , DE G , RAVA , CAO , I
 * NCO , IMPL , ITA E , DEC , LARA , CAO , DE T , AMAN , HO D
 * , O DA , DO N , AD R , EFER , ENCI , ADA ? , ERRO ,
 * 20 = , ES , PECI , FICA , CAO , DE M , ODO , DE G , RAV
 * A , CAO , COM , CARA , CTER , ES A , MAI , S NO , FIM ,
 * ? , ERRO , 21 = , CA , RACT , ERE , ESTR , ANHO , N

*A 'ESPE', 'CIFI', 'CACA', 'O DO', 'MOD', 'O DE', 'GRA', 'VACA'
 *,'O EM', 'TAM', 'ANHO', 'FIX', 'O', 'ERRO', '22', 'ES', 'PECI', 'FICA', 'CAO', 'DE M', 'ODO', 'DE G', 'RAVA', 'CAO'
 *,'EM T', 'AMAN', 'HO V', 'ARIA', 'VEL', 'COM', 'CARA', 'CTER', 'ES A', 'MAI', 'S NO', 'FIM', 'ERRO', '23', 'CA', 'RAC'
 *T', 'ERE', 'ESTR', 'ANHO', 'NA', 'ESPE', 'CIFI', 'CACA', 'O DE', 'MOD', 'O DE', 'GRA', 'VACA', 'O EM', 'TAM', 'ANHO', 'VAR', 'IA'
 *VE', 'L', 'ERRO', '24', 'CA', 'RACT', 'ERE', 'ESTR', 'ANHO', 'NA', 'ESPE', 'CIFI', 'CACA', 'O DO', 'TIPI', 'O DE', 'GRA', 'VACA', 'O', 'ERRO', '25', 'IN', 'FORM', 'ACAO'
 *,'EST', 'RANH', 'A ON', 'DE E', 'RA E', 'SPER', 'ADO', 'O MO', 'DO D', 'E GR', 'AVAC', 'AO D', 'O DA', 'DO', 'ERRO', '26', 'DA', 'DO D', 'E TA', 'MANH', 'O FI', 'XO S', 'EM E', 'SPEC', 'IFIC', 'ACAO', 'DO', 'PROP', 'RIO', 'TAMA', 'NHO', 'ERRO', '27', 'CA', 'RACT', 'ERE', 'EST', 'RANH', 'O NA', 'ESPI', 'ECIF', 'ICAC', 'AO D', 'E TA', 'MANH', 'O DO', 'DAD', 'O', 'ERRO', '28', 'ES', 'PACO', 'S EM', 'BRA', 'NCO', 'INSE', 'RIDO', 'S NO', 'NOM', 'E DO', 'CAM', 'PO', 'ERRO', '29', 'CA', 'RACT', 'ERE', 'NUME', 'RICO', 'NO', 'NOME', 'DO', 'CAMP', 'O', 'ERRO', '30', 'CA', 'RACT', 'ERES', 'EST', 'RANH', 'OS I', 'NSER', 'IDOS', 'NO', 'NOME', 'DO', 'CA', 'MP', 'O', 'ERRO', '31', 'NO', 'ME D', 'O CA', 'MPO', 'COM', 'CARA', 'CTER', 'ES A', 'MAI', 'S NO', 'FIM', 'ERRO', '32', 'DA', 'DOS', 'DESC', 'RITO', 'S PE', 'LO U', 'SUAR', 'IO F', 'ORA', 'DE S', 'EQUE', 'NCIA', 'ERRO', '33', 'CA', 'RACT', 'ERE', 'ESTR', 'ANHO', 'NA', 'DESC', 'RICA', 'O D', 'E', 'DAD', 'O AL', 'FA', 'ERRO', '34', 'DE', 'SCRI', 'CAO', 'DE D', 'ADO', 'ALFA', 'BETI', 'CO I', 'NCOM', 'PLET', 'A', 'OU', 'CON', 'TEND', 'O ES', 'PACO', 'S EM', 'BRA', 'NCO', 'ERRO', '35', 'CA', 'RACT', 'ERE', 'NUME', 'RICO', 'INS', 'E', 'RID', 'O NA', 'DES', 'CRIC', 'AO D', 'E DA', 'DO A', 'LFAB', 'ETIC', 'O', 'ERRO', '36', 'DE', 'SCRI', 'CAO', 'DE D', 'ADO', 'NUME', 'RICO', 'INC', 'OMPL', 'ETA', 'OU C', 'ONTE', 'INDO', 'ESPA', 'COS', 'EM B', 'RANC', 'O', 'ERRO', '37', 'CA', 'RACT', 'ERES', 'EST', 'RANH', 'OS N', 'A DE', 'SCRI', 'CA', 'O', 'DE D', 'ADO', 'NUME', 'RICO', 'ERRO', '38', 'CA', 'RACT', 'ERE', 'ALFA', 'BETI', 'CO I', 'NSER', 'IDO', 'NA D', 'E', 'SCRI', 'CAO', 'DE', 'DADO', 'NUM', 'ERIC', 'O', 'ERRO', '39', 'DA', 'DO C', 'OM O', 'IRGAN', 'IZAC', 'AO D', 'E LI', 'STA', 'COM', 'ELEM', 'ENTO', 'S ES', 'TRAN', 'HOS', 'AO C', 'ONT', 'E', 'UDO', 'DA M', 'ESMA', 'NA', 'TABE', 'LA D', 'E SI', 'MBOL', 'OS', 'ERRO', '40', 'DE', 'LIMI', 'TADO', 'R AU', 'SE', 'NT', 'E NA', 'DES', 'CRIC', 'AO D', 'E EL', 'EMEN', 'TOS', 'DAS', 'LIST', 'AS', 'ERRO', '41', 'ES', 'PACO', 'S EM', 'BRA', 'NCO', 'INSE', 'RIDO', 'S NA', 'DES', 'CRIC', 'AO D', 'OS E', 'LEME', 'NTOS', 'DAS', 'LIS', 'TAS', 'ERRO', '42', 'DE', 'SCRI', 'CAO', 'DE D', 'ADO', 'COM', 'ORGA', 'NIZA', 'CAO', 'DE L', 'ISTA', 'EXC', 'LUSI', 'VA S', 'EM C', 'ONTE', 'UDO', 'LOGI', 'CO', 'ERRO', '43', 'DE', 'SCRI', 'CAO', 'DE', 'LEME', 'NTOS', 'EM', 'LIST', 'A IN', 'CLUS', 'IVA', 'COM', 'CARA', 'CTER', 'E ES', 'TRAN', 'HO A', 'O CO', 'NTEU', 'DO D', 'A', 'ME', 'SMA', 'NA T', 'ABEL', 'A DE', 'SIM', 'BOLO', 'S', 'ERRO', '44', 'NU', 'MERO', 'DE', 'CARA', 'CTER', 'ES U', '

```
*TILI','ZADO','S PE','LO U','SUAR','IO N','A DE','SCRI','CAO
* ','DE D','ADO ','DE T','AMAN','HO F','IXO ','NAO ','COIN',
*'CIDE',' COM',' O D','ES R','ITO ','NA T','ABEL','A DE',' S
*IM','BOLO','S ',' '?/
```

```
KART=1
INTERG=0
KARLA=0
1010 KARLA=KARLA+1
IF(IFRAS(KARLA).EQ.LETUR) GO TO 1012
GO TO 1010
1012 INTERG=INTERG+1
IF(INTERG.EQ.N) GO TO 1013
KART=KARLA+1
GO TO 1010
1013 ICOM=KART
ITER=KARLA+1
RETURN
END
```

SUBROUTINE INDTAM

```
C SUBROTINA PARA DETERMINAÇÃO DO ENDERECO DA PRIMEIRA LISTA
C DA TABELA DE SIMBOLOS
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDOC
*O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),NU
*MLIS,LISEND(100),KAM,ICART(80),KLOST,KAD,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMECCO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IQG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
KEY=0
MAGNA=0
ISCO=0
LETTER=0
857 READ(8,850)(ICART(M),M=1,80)
850 FORMAT(80A1)
DO 852 J=1,80
IF(ICART(J).EQ.IALFA(28)) GO TO 853
IF(ICART(J).EQ.IALFA(42)) GO TO 854
IF(ICART(J).EQ.IALFA(27)) GO TO 852
DO 855 L=1,26
IF(ICART(J).EQ.IALFA(L)) GO TO 856
IF(ICART(J).EQ.IALFA(49)) GO TO 856
855 CONTINUE
MAGNA=MAGNA+LETTER
KEY=1
852 CONTINUE
GO TO 857
856 LETTER=LETTER+1
MAGNA=MAGNA+1
GO TO 852
853 LETTER=0
IF(KEY.EQ.0) GO TO 858
```

```

KEY=0
GO TO 852
858 ISCO=ISCO+1
GO TO 852
854 IENDCA=MAGNA+ISCO*14
RETURN
END

```

```

SUBROUTINE LELIST
C SUBROTINA PARA LEITURA E GRAVACAO DAS LISTAS
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDUC
*O,INGRID,IDA,IA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMECD,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
IENDLI(1)=0
ITOT=0
800 ITOT=ITOT+1
MONTE(ITOT)=IALFA(27)
NOTE(ITOT)=IALFA(27)
IF(ITOT.NE.4) GO TO 800
ITOT=ITOT+1
MONTE(ITOT)=IALFA(39)
NOTE(ITOT)=IALFA(39)
ITOT=ITOT+1
704 READ(8,700)(LISTA(M),M=1,80)
700 FORMAT(80A1)
IF(LISTA(1).EQ.IALFA(42)) GO TO 702
DO 703 M=1,80
KUBA=M
IF(LISTA(M).EQ.IALFA(27)) GO TO 703
IF(LISTA(M).EQ.IALFA(39)) GO TO 900
IF(LISTA(M).EQ.IALFA(49)) GO TO 1050
IF(LISTA(M).EQ.IALFA(51)) GO TO 1050
DO 1071 IVV=29,38
IF(LISTA(M).EQ.IALFA(IVV)) GO TO 1050
1071 CONTINUE
DO 1051 IPODA=1,26
IF(LISTA(M).EQ.IALFA(IPODA)) GO TO 1050
1051 CONTINUE
CALL IERRO(1)
GO TO 703
1050 MONTE(ITOT)=LISTA(M)
NOTE(ITOT)=LISTA(M)
ITOT=ITOT+1
703 CONTINUE
GO TO 704
702 KLAST=ITOT-1
ITOT=ITOT-1
GO TO 902

```

```

900 KLAB=ITOT
    KLIB=ITOT+3
904 MONTE(KLAB)=IALFA(27)
    NOTE(KLAB)=IALFA(27)
    KLAB=KLAB+1
    IF(KLAB.LE.KLIB) GO TO 904
    ITOT=ITOT+4
    MONTE(ITOT)=LISTA(KUBA)
    NOTE(ITOT)=LISTA(KUBA)
    ITOT=ITOT+1
    GO TO 703
902 RETURN
    END

```

SUBROUTINE IENDER

```

C   SUBROTINA PARA CALCULAR OS ENDEREÇOS DE TODAS AS LISTAS
    COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
    *M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDUC
    *O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
    *0),ITOT,IFNDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
    *M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
    COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMECO,KOFI
    *M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
    *UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
    ITOT=6
    KAM=1
    IENDLI(KAM)=IENDCA+5
712 IF(MONTE(ITOT).EQ.IALFA(39)) GO TO 711
    IF(MONTE(ITOT).NE.IALFA(51)) GO TO 805
    ITOT=ITOT+1
    IF(MONTE(ITOT).EQ.IALFA(51)) GO TO 714
    GO TO 712
805 ITOT=ITOT+1
    GO TO 712
711 KAM=KAM+1
    IENDLI(KAM)=IFNDCA+ITOT
    ITOT=ITOT+1
    GO TO 712
714 KLAST=ITOT
    RETURN
    END

```

SUBROUTINE KONVER

```

C   SUBROTINA PARA DESCOMPACTAR OS ENDEREÇOS DAS LISTAS=
    COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
    *M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDUC
    *O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
    *0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),NU
    *MLIS,KAM,ICART(80),KERO,KEVEDO,KLOST,NOTE(9999),KLAST
    COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMECO,KOFI
    *M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
    *UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
    IF(KERO.LT.0) GO TO 906

```

```

IF(KERO.GT.9999) GO TO 906
KONT(1)=KERO/1000
KONT(2)=(KERO-KONT(1)*1000)/100
KONT(3)=(KERO-(KONT(1)*1000+KONT(2)*100))/10
KONT(4)=KERO-(KONT(1)*1000+KONT(2)*100+KONT(3)*10)
GO TO 905
906 KEVEDO=1
905 RETURN
END

```

```

C SUBROUTINE KONALF
SUBROTINA PARA CONVERSAO DE ENDEREÇOS NUMERICOS EM ALFA
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDOC
*O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),KAM,ICART(80),KLOST,KAD,NOTE(9999),KLAS
*T
COMMON NIDADO(80),KIMFRA,JUCK,NA,MIGRA,IORG,KLI,KOMEÇO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
DO 914 M=1,4
DO 915 N=1,10
IF(KONT(M).EQ.NUMERO(N)) GO TO 917
915 CONTINUE
917 MONTE(M)=NUMALF(N)
914 CONTINUE
RETURN
END

```

```

C SUBROUTINE NAMEL
SUBROTINA PARA DEPURAR O NOME DO DADO
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDOC
*O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
COMMON NIDADO(80),KIMFRA,JUCK,NA,MIGRA,IORG,KLI,KOMEÇO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
KID=0
MARCF=0
IF(IVET(K).NE.IALFA(28)) GO TO 17
CALL IERRO(2)
GO TO 5002
17 IF(IVET(K).NE.IALFA(27)) GO TO 21
K=K+1
IF(K.LE.80) GO TO 17
23 IF(IBRAN)5002,24,5002
24 IBRAN=1
GO TO 5002
21 IF(IBRAN.EQ.0) GO TO 22
IBRAN=0
22 DO 27 J=1,27

```

```

IF(ICHAV,NE,0) GO TO 28
29 IF(IVET(K),EQ,IALFA(27)) GO TO 47
46 IF(IVET(K),EQ,IALFA(J)) GO TO 31
IF(IVET(K),EQ,IALFA(49)) GO TO 31
IF(IVET(K),NE,IALFA(27)) GO TO 27
K=K+1
GO TO 46
27 CONTINUE
CALL IERRO(3)
I=I+IMARC
GO TO 5002
31 IMARC=IMARC+1
CALL ISAI(IVET(K))
GO TO 48
47 ISUPR=1
48 K=K+1
614 IF(IVET(K),NE,IALFA(27)) GO TO 613
K=K+1
GO TO 614
613 IF(IVET(K),EQ,IALFA(28)) GO TO 34
ICHAV=1
GO TO 21
28 ICHAV=0
37 IF(IVET(K),NE,IALFA(27)) GO TO 35
ISUPR=1
K=K+1
IF(K,LE,80) GO TO 37
I=I+IMARC
KTCHE=IMARC
MARC=1
GO TO 5002
35 IF(IVET(K),NE,IALFA(28)) GO TO 46
34 MELHOR=K
252 K=K+1
IF(IVET(K),NE,IALFA(27)) GO TO 251
IF(K,LT,80) GO TO 252
I=I+IMARC
CALL IERRO(4)
GO TO 5002
251 K=MELHOR
ISALV=IMARC
CALL ISAI(IVET(K))
GO TO 5003
5002 KID=1
5003 RETURN
END

```

SUBROUTINE KTIPO

```

C SUBROTINA PARA DETETAR O TIPO DO DADO : NUMERICO OU ALFA
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDUC
*O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),NU

```

```

*MLIS,LISEND(100),KAM,ICART(80),KLOST,KAD,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMECO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,I0G,INY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
LIMO=0
62 K=K+1
   IF(IVET(K).EQ.IALFA(27)) GO TO 62
   IF(IVET(K).NE.LISIN(1)) GO TO 64
66 CALL IERRO(5)
   GO TO 561
64 IF(IVET(K).EQ.KFITA(1)) GO TO 66
   DO 67 IME=1,4
559 IF(IVET(K).NE.IALFA(27)) GO TO 558
   K=K+1
   GO TO 559
558 IF(IVET(K).NE.IAL(IME)) GO TO 71
   K=K+1
67 LIMO=1
   IF(IVET(K).NE.LISIN(1)) GO TO 996
995 K=K+1
994 CALL ISAI(IALFA(1))
   GO TO 72
996 IF(IVET(K).EQ.KFITA(1)) GO TO 995
   IF(IVET(K).EQ.IALFA(27)) GO TO 994
   CALL IERRO(6)
   GO TO 561
71 IF(LIMO.EQ.0) GO TO 76
   CALL IERRO(7)
   GO TO 561
76 DO 300 ITA=1,8
562 IF(IVET(K).NE.IALFA(27)) GO TO 563
   K=K+1
   GO TO 562
563 IF(IVET(K).NE.INUM(ITA)) GO TO 301
   K=K+1
300 LIMO=1
   IF(IVET(K).NE.LISIN(1)) GO TO 992
991 K=K+1
900 CALL ISAI(IALFA(14))
   GO TO 72
992 IF(IVET(K).EQ.KFITA(1)) GO TO 991
   IF(IVET(K).EQ.IALFA(27)) GO TO 900
   CALL IERRO(8)
   GO TO 561
301 IF(LIMO.EQ.0) GO TO 302
   CALL IERRO(9)
   GO TO 561
302 CALL IERRO(10)
561 KIERRO=1
   I=I+(IMARC+3)
72 RETURN
END

```



```

SUBROUTINE ICOGI
C  SUBROTINA PARA DETECTAR A ORGANIZACAO DO DADO
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDU
*O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),NU
*MLIS,KAM,ICART(80),KLOST,KAD,ISTEIM,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMECO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
KIERRO=0
KING=0
IF(KDUCCO) 93,94,93
94 K=K+1
81 IF(IVET(K),NE,IALFA(27)) GO TO 77
K=K+1
IF(IVET(K),NE,KFITA(1)) GO TO 81
K=K+1
CALL ISAI(IALFA(29))
KDUCCO=1
GO TO 84
77 DO 85 IANCE=1,14
576 IF(IVET(K),NE,IALFA(27)) GO TO 575
K=K+1
KING=KING+1
IF(K,LE,80) GO TO 576
88 CALL IERRO(11)
GO TO 552
575 IF(IVET(K),NE,LISIN(IANCE)) GO TO 86
KING=KING+1
85 K=K+1
IF(IVET(K),NE,KFITA(1)) GO TO 1020
K=K+1
1021 CALL ISAI(IALFA(12))
KDUCCO=2
GO TO 84
1020 IF(IVET(K),EQ,IALFA(27)) GO TO 1021
CALL IERRO(12)
GO TO 552
86 IF(IANCE,LT,7) GO TO 1023
CALL IERRO(13)
GO TO 552
1023 K=K-KING
KING=0
DO 87 NAMA=1,14
578 IF(IVET(K),NE,IALFA(27)) GO TO 577
K=K+1
KING=KING+1
IF(K,LE,80) GO TO 578
GO TO 88
577 IF(IVET(K),NE,LISEX(NAMA)) GO TO 550
KING=KING+1
87 K=K+1

```

```

      IF(IVET(K),NE,KFITA(1)) GO TO 1025
      K=K+1
1026 CALL ISAI(IALFA(12))
      KDUCC=3
      GO TO 84
1025 IF(IVET(K),EQ,IALFA(27)) GO TO 1026
      CALL IERRO(14)
      GO TO 552
550 IF(KING,553,554,553)
554 CALL IERRO(15)
      GO TO 552
553 IF(NAMA.LT,7) GO TO 1029
      CALL IERRO(16)
      GO TO 552
1029 CALL IERRO(17)
552 KIERRO=1
      I=I+(IMARC+4)
      GO TO 84
93 GO TO (90,91,92),KDUCC
90 CALL ISAI(IALFA(29))
      GO TO 84
91 CALL ISAI(IALFA(9))
      GO TO 84
92 CALL ISAI(IALFA(5))
84 RETURN
      END

```

```

SUBROUTINE LGRILL
C SUBROTINA PARA DETECTAR O MODO DE GRAVACAO E O TAMANHO DO
C CAMPO
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDUCC
*O,INGRID,IDALIA,ISPTO,KEPE,MONTF(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),KAM,ICART(80),KLOST,KAD,NOTE(9999),KLAS
*T
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMECO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
ISPTO=0
KASE=0
KIERRO=0
IF(INGRID,NE,0) GO TO 103
K=K+1
107 IF(IVET(K),NE,IALFA(27)) GO TO 105
      K=K+1
      IF(K,LE,80) GO TO 107
      CALL IERRO(18)
      IDALIA=1
      GO TO 585
105 DO 113 LALA=1,11
581 IF(IVET(K),NE,IALFA(27)) GO TO 580
      K=K+1
      KASE=KASE+1

```

```
      IF(K,LE,80) GO TO 581
591  CALL IERRO(19)
      GO TO 585
580  IF(IVET(K),NE,KFITA(LALA)) GO TO 114
      KASE=KASE+1
113  K=K+1
      DO 1030 KFLY=29,38
      IF(IVET(K),EQ,IALFA(KFLY)) GO TO 1031
1030  CONTINUE
      IF(IVET(K),EQ,IALFA(27)) GO TO 1033
      CALL IERRO(20)
      GO TO 585
1031  K=K-1
1033  CALL ISAI(IALFA(20))
      INGRID=1
      GO TO 110
114  IF(LALA,LT,9) GO TO 1034
      CALL IERRO(21)
      GO TO 585
1034  K=K-KASE
      KASE=0
      DO 115 LULA=1,15
583  IF(IVET(K),NE,IALFA(27)) GO TO 582
      K=K+1
      KASE=KASE+1
      IF(K,LE,80) GO TO 583
      GO TO 591
582  IF(IVET(K),NE,KVATA(LULA)) GO TO 116
      KASE=KASE+1
115  K=K+1
      DO 1036 JLIPI=29,38
      IF(IVET(K),EQ,IALFA(JLIPI)) GO TO 1037
1036  CONTINUE
      IF(IVET(K),EQ,IALFA(27)) GO TO 1038
      CALL IERRO(22)
      GO TO 585
1037  K=K-1
1038  CALL ISAI(IALFA(20))
      INGRID=2
      GO TO 110
116  IF(KASE,EQ,0) GO TO 588
      IF(LULA,LT,9) GO TO 1040
      CALL IERRO(23)
      GO TO 585
1040  CALL IERRO(24)
      GO TO 585
588  CALL IERRO(25)
      GO TO 585
103  GO TO (101,102),INGRID
101  CALL ISAI(IALFA(6))
      K=K+1
120  IF(IVET(K),NE,IALFA(27)) GO TO 118
      K=K+1
```

```

IF(K.LE.80) GO TO 120
CALL IERRO(26)
KIERRO=1
I=I+(IMARC+8)
GO TO 110
118 IBIT=0
127 DO 123 NACO=29,39
611 IF(IVET(K).NE.IALFA(NACO)) GO TO 610
CALL ISAI(IALFA(NACO))
K=K+1
IBIT=IBIT+1
IF(IBIT=1) 609,127,609
610 IF(IVET(K).NE.IALFA(27)) GO TO 123
K=K+1
GO TO 611
123 CONTINUE
CALL IERRO(27)
126 INGRID=1
IF(IBIT.EQ.0) GO TO 615
I=I+(IMARC+9)
616 KIERRO=1
GO TO 110
615 I=I+(IMARC+8)
GO TO 616
102 CALL ISAI(IALFA(22))
CALL ISAI(IALFA(29))
ISPTO=1
609 INGRID=1
GO TO 110
585 KIERRO=1
I=I+(IMARC+6)
110 RETURN
END

```

```

SUBROUTINE KLAUDI
C SUBROTINA PARA VALIDACAO DE NOME DO CAMPO
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDOC
*O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIFERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
COMMON NIDADO(80),KIMFRA,JUCK,NA,MIGRA,IORG,KLI,KOMECO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IQG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGFRAL,LIN
KAND=0
KLI=0
INCRE=0
KIMFRA=0
IF(NIDADO(1).NE.IALFA(30)) GO TO 3019
KLI=KLI+1
3019 KLI=KLI+1
JUCK=JUCK+1

```

```

3012 IF(NIDADO(KLI),NE,IALFA(27)) GO TO 3010
      IF(INCRE,NE,0) GO TO 3011
      KLI=KLI+1
      GO TO 3012
3011 IF(IMPRI(JUCK),EQ,IALFA(28)) GO TO 3013
      CALL IERRO(28)
      GO TO 3015
3013 IF(KAND,NE,1) GO TO 3016
      CALL IERRO(32)
      GO TO 3015
3010 IF(NIDADO(KLI),NE,IMPRI(JUCK)) GO TO 3018
      INCRE=INCRE+1
      GO TO 3019
3018 IF(IMPRI(JUCK),NE,IALFA(28)) GO TO 3020
      CALL IERRO(31)
      GO TO 3015
3020 IF(INCRE,NE,0) GO TO 3022
      KAND=1
3023 JUCK=JUCK+1
      IF(IMPRI(JUCK),NE,IALFA(28)) GO TO 3023
      JUCK=JUCK+14
      KLI=1
      GO TO 3012
3022 DO 3024 JIJI=29,38
      IF(NIDADO(KLI),EQ,IALFA(JIJI)) GO TO 3025
3024 CONTINUE
      DO 3026 JIJI=39,52
      IF(NIDADO(KLI),EQ,IALFA(JIJI)) GO TO 3027
3026 CONTINUE
      INCRE=0
      KLI=1
      KAND=1
3028 JUCK=JUCK+1
      IF(IMPRI(JUCK),NE,IALFA(28)) GO TO 3028
      JUCK=JUCK+14
      GO TO 3012
3025 CALL IERRO(29)
      GO TO 3015
3027 CALL IERRO(30)
3015 KIMERA=1
      IGERAL=1
      JUCK=0
3016 RETURN
      END

```

SUBROUTINE NAIEFV

```

C SUBPROGRAMA PARA DETECTAR A ORGANIZACAO, TIPO E MODO DE GRA-
C VAÇAO DO DADO
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDOC
*O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIFERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEYEDO,KID,NOTE(9999),KLAST

```

```

COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMEÇO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
JUCK=JUCK+3
IF(IMPRI(JUCK),EQ,IALFA(1)) GO TO 10
NA=2
GO TO 11
10 NA=1
11 JUCK=JUCK+1
IF(IMPRI(JUCK),EQ,IALFA(29)) GO TO 12
JUCK=JUCK+1
IF(IMPRI(JUCK),EQ,IALFA(5)) GO TO 13
IORG=2
16 JUCK=JUCK+2
17 IF(IMPRI(JUCK),EQ,IALFA(6)) GO TO 14
MIGRA=2
GO TO 15
14 MIGRA=1
GO TO 15
13 IORG=3
GO TO 16
12 IORG=1
JUCK=JUCK+3
GO TO 17
15 RETURN
END

```

```

SUBROUTINE KALFCK
C SUBROTINA PARA VALIDACAO DA DESCRICAO DE DADO ALFA
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDOC
*O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMEÇO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
KARTY=0
KABRAN=0
KOMEÇO=KLI
203 DO 201 KIKI=29,38
IF(NIDADO(KLI),EQ,IALFA(KIKI)) GO TO 202
201 CONTINUE
KLI=KLI+1
IF(KLI,LE,80) GO TO 203
KLI=KOMEÇO
215 DO 204 MIDI=1,26
IF(NIDADO(KLI),EQ,IALFA(MIDI)) GO TO 205
204 CONTINUE
IF(NIDADO(KLI),EQ,IALFA(49)) GO TO 205
208 IF(NIDADO(KLI),NE,IALFA(27)) GO TO 207
KABRAN=KABRAN+1
KLI=KLI+1

```

```

      IF(KLI,LE,80) GO TO 208
      LEVA=KOFIM=KOMECCO+1
      DO 315 KAB=KOMECCO,KOFIM
      KARTY=KARTY+1
315  ICOMP(KARTY)=NIDADO(KAB)
      GO TO 210
207  IF(KABRAN,EQ,0) GO TO 211
      CALL IERRO(34)
      GO TO 213
211  CALL IERRO(33)
      GO TO 213
205  KLI=KLI+1
      KOFIM=KLI-1
      GO TO 215
202  CALL IERRO(35)
213  KIMERA=1
      IGERAL=1
      JUCK=0
210  RETURN
      END

```

```

C      SUBROUTINE NUMCHK
      SUBROTINA PARA VALIDACAO DE DADOS NUMERICOS
      COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
      *M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDOC
      *O,INGRID,IDADIA,ISPTO,KEPF,MONTE(9999),KIERRO,IMARC,LISTA(8
      *0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
      *M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
      COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMECCO,KOFI
      *M,ICOMP(80),KARTY,KEMBY,LEVA,IQG,IMY,LYAUX(500),KAMEL(10),L
      *UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
      KARTY=0
      KABRAN=0
      KOMECCO=KLI
105  DO 103 MYNY=1,26
      IF(NIDADO(KLI),EQ,IALFA(MYNY)) GO TO 104
103  CONTINUE
      KLI=KLI+1
      IF(KLI,LE,80) GO TO 105
      KLI=KOMECCO
116  DO 106 MAXI=29,38
      IF(NIDADO(KLI),EQ,IALFA(MAXI)) GO TO 107
106  CONTINUE
108  IF(NIDADO(KLI),NE,IALFA(27)) GO TO 111
      KABRAN=KABRAN+1
      KLI=KLI+1
      IF(KLI,LE,80) GO TO 108
      LEVA=KOFIM=KOMECCO+1
      DO 130 KUT=KOMECCO,KOFIM
      KARTY=KARTY+1
130  ICOMP(KARTY)=NIDADO(KUT)
      GO TO 110
111  IF(KABRAN,EQ,0) GO TO 112

```

```

    CALL IERRO(36)
    GO TO 114
112 CALL IERRO(37)
    GO TO 114
107 KLI=KLI+1
    KOFIM=KLI-1
    GO TO 116
104 CALL IERRO(38)
114 KIMERA=1
    IGERAL=1
    JUCK=0
110 RETURN
    END

```

```

SUBROUTINE LISTCK
C  SUBROTINA PARA VALIDACAO DE DADOS EM LISTA
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDUÇ
*O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMEÇO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
LEVA=0
317 IF(NIDADO(KLI),EQ,IALFA(51)) GO TO 300
    IF(LEVA,EQ,0) GO TO 301
    IF(IORG,EQ,2) GO TO 340
    CALL IERRO(39)
    GO TO 303
301 CALL IERRO(40)
    GO TO 303
300 IF(LEVA,EQ,0) GO TO 305
    IF(IORG,NE,3) GO TO 306
308 KLI=KLI+1
    IF(KLI,GT,80) GO TO 307
    IF(NIDADO(KLI),EQ,IALFA(27)) GO TO 308
    CALL IERRO(42)
    GO TO 303
307 LEVA=LEVA+1
    ICOMP(LEVA)=IALFA(51)
    GO TO 311
305 LEVA=LEVA+1
    ICOMP(LEVA)=IALFA(51)
312 KLI=KLI+1
    IF(NIDADO(KLI),EQ,IALFA(27)) GO TO 312
318 DO 313 MEL=1,26
    IF(NIDADO(KLI),EQ,IALFA(MEL)) GO TO 314
313 CONTINUE
    DO 315 MITO=29,38
    IF(NIDADO(KLI),EQ,IALFA(MITO)) GO TO 314
315 CONTINUE
    IF(NIDADO(KLI),NE,IALFA(49)) GO TO 317

```



```

314 LEVA=LEVA+1
    ICOMP(LEVA)=NIDADO(KLI)
    KLI=KLI+1
    IF(NIDADO(KLI),NE,IALFA(27)) GO TO 318
    IF(IORG,NE,3) GO TO 319
    CALL IERRO(41)
    GO TO 303
306 LEVA=LEVA+1
    ICOMP(LEVA)=IALFA(51)
322 KLI=KLI+1
    IF(KLI,GT,80) GO TO 311
    IF(NIDADO(KLI),EQ,IALFA(27)) GO TO 322
327 DO 323 KELY=1,26
    IF(NIDADO(KLI),EQ,IALFA(KELY)) GO TO 324
323 CONTINUE
    DO 325 NEY=29,38
    IF(NIDADO(KLI),EQ,IALFA(NEY)) GO TO 324
325 CONTINUE
    IF(NIDADO(KLI),NE,IALFA(49)) GO TO 326
324 LEVA=LEVA+1
    ICOMP(LEVA)=NIDADO(KLI)
    KLI=KLI+1
    IF(NIDADO(KLI),NE,IALFA(27)) GO TO 327
319 CALL IERRO(41)
    GO TO 303
326 IF(NIDADO(KLI),EQ,IALFA(51)) GO TO 306
340 CALL IERRO(43)
303 KIMERA=1
    IGERAL=1
    JUCK=0
311 RETURN
    END

```

SUBROUTINE LAUXER

```

C SUBROTINA PARA GRAVACAO DE DADOS INDEPENDENTES NO MINICADAS
C TRD LYAUX(M)
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDUÇ
?O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDÇA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMEÇO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
LUZ=0
KAPONT=0
KARTY=0
IF(IORG,EQ,1) GO TO 4030
LUZ=1
GO TO 4020
4030 IF(MIGRA,NE,2) GO TO 4023
    IF(KIMERA,EQ,1) GO TO 4020
    KEMBY=LEVA

```

```

CALL KONVTA
CALL KALFTA
4025 IMY=IOG+LEVA
4022 IOG=IOG+1
IF(IOG.GT.IMY) GO TO 4021
KARTY=KARTY+1
LYAUX(IOG)=ICOMP(KARTY)
GO TO 4022
4021 IOG=IOG-1
GO TO 4020
4023 KEMBY=LEVA
CALL KONVTA
CALL KALFTA
JUCK=JUCK+1
IF(IMPRI(JUCK),NE,LYAUX(IOG=1)) GO TO 4024
JUCK=JUCK+1
IF(IMPRI(JUCK),NE,LYAUX(IOG)) GO TO 4024
JUCK=JUCK-2
GO TO 4025
4024 KAPONT=KAPONT+1
CALL IERRO(44)
GO TO (4028,4029),KAPONT
4028 JUCK=JUCK-1
GO TO 4051
4029 JUCK=JUCK-2
4051 KIMERA=1
JUCK=0
IGERAL=1
4020 RETURN
END

```

```

C SUBROUTINE KONVTA
SUBROTINA PARA DESCOMPACTACAO DE NUMEROS
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDOC
?O,INGRID,IDALIA,ISPTD,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMECC,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
IF(KEMBY.GT.99) GO TO 180
KAMEL(1)=KEMBY/10
KAMEL(2)=KEMBY-KAMEL(1)*10
GO TO 181
180 CALL IERRO(46)
181 RETURN
END

```

```

C SUBROUTINE KALFTA
SUBROTINA PARA ALOCACAO DE TAMANHO DO DADO
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDOC

```

```

?O, INGRID, IDALIA, ISPTO, KEPE, MONTE(9999), KIERRO, IMARC, LISTA(8
*0), ITOT, IENDLI(100), IENDCA, KONT(4), NUMALF(10), NUMERO(10), KA
*M, ICART(80), KPRI, KERO, KEVEDO, KID, NOTE(9999), KLAST
COMMON NIDADO(80), KIMERA, JUCK, NA, MIGRA, IORG, KLI, KOMECO, KOFI
*M, ICOMP(80), KARTY, KEMBY, LEVA, IOG, IMY, LYAUX(500), KAMEL(10), L
*UZ, IMPRES(5), JLEMBR, IFIBAR(50), IMPRIM, KTITO, LEVY, IGERAL, LIN
IBSE=1
188 DO 186 LECLAI=1,10
    IF(KAMEL(IBSE).EQ.NUMERO(LECLAI)) GO TO 187
186 CONTINUE
187 IOG=IOG+1
    LYAUX(IOG)=NUMALF(LECLAI)
    IBSE=IBSE+1
    IF(IBSE.GT.2) GO TO 189
    GO TO 188
189 RETURN
END

```

```

C SUBROUTINE LISCOD
SUBROTINA PARA DECODIFICACAO DE DADOS EM "LE"
COMMON IBLOC, I, ICONT, IPEG, K, IMPRI(9999), IVET(80), IAL(4), INU
*M(8), IALFA(51), KFITA(12), KVATA(16), LISIN(14), LISEX(14), KDOC
?O, INGRID, IDALIA, ISPTO, KEPE, MONTE(9999), KIERRO, IMARC, LISTA(8
*0), ITOT, IENDLI(100), IENDCA, KONT(4), NUMALF(10), NUMERO(10), KA
*M, ICART(80), KPRI, KERO, KEVEDO, KID, NOTE(9999), KLAST
COMMON NIDADO(80), KIMERA, JUCK, NA, MIGRA, IORG, KLI, KOMECO, KOFI
*M, ICOMP(80), KARTY, KEMBY, LEVA, IOG, IMY, LYAUX(500), KAMEL(10), L
*UZ, IMPRES(5), JLEMBR, IFIBAR(50), IMPRIM, KTITO, LEVY, IGERAL, LIN
ICONDO=0
JLEMBR=JUCK
JUCK=JUCK+3
CALL KTURN
IF(IMPRI(JUCK=1).NE.IALFA(27)) GO TO 4010
IF(IMPRI(JUCK).NE.IALFA(39)) GO TO 4010
IF(IORG.EQ.2) GO TO 4050
4018 DO 4012 MAR=2,LEVA
    JUCK=JUCK+1
    IF(ICOMP(MAR).NE.IMPRI(JUCK)) GO TO 4013
4012 CONTINUE
    ICONDO=ICONDO+1
    LEVA=4
    IOG=IOG+1
    LYAUX(IOG)=NUMALF(1)
    IOG=IOG+1
    LYAUX(IOG)=NUMALF(5)
    IOG=IOG+1
    DO 4015 MARC=IOG, IOG+2
    IF(ICONDO.GE.10) GO TO 4014
    LYAUX(MARC)=NUMALF(1)
4015 CONTINUE
    LYAUX(IOG+3)=NUMALF(ICONDO+1)
4017 IOG=IOG+3
    JUCK=JLEMBR

```

```

      GO TO 4011
4014  LYAUX(IOG)=NUMALF(1)
      LYAUX(IOG+1)=NUMALF(1)
      IREVER=ICONDO/10
      IRE=ICONDO=10*IREVER
      LYAUX(IOG+2)=NUMALF(IREVER+1)
      LYAUX(IOG+3)=NUMALF(IRE+1)
      GO TO 4017
4013  JUCK=JUCK+1
      IF(IMPRI(JUCK),NE,IALFA(51)) GO TO 4013
      ICONDO=ICONDO+1
      GO TO 4018
4010  KIMERA=1
      IGERAL=1
      JUCK=ILEMBR
      GO TO 4011
4050  CALL LEXCOD
4011  RETURN
      END

```

```

C      SUBROUTINE LEXCOD
      SUBROTINA PARA DECODIFICACAO DE DADOS EM "LI"
      COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
      *M(8),IALFA(51),KPITA(12),KVATA(16),LISIN(14),LISEX(14),KDU
      ?O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
      *0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
      *M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
      COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMECCO,KOFI
      *M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
      *UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
      IVES=0
      IBARRA=0
      ICONDO=0
      MAR=2
5061  IF(ICOMP(MAR),EQ,IALFA(51)) GO TO 5060
5072  MAR=MAR+1
      IF(MAR,LE,LEVA) GO TO 5061
      KNIC=2
      KFINDO=IBARRA
      KTAMA=4*KFINDO
      KEMBY=KTAMA
      CALL KONVTA
      CALL KALFTA
      IOG=IOG+1
      IBARRA=1
5067  INSTY=IFIBAR(IBARRA)
5071  DO 5062 MAR=KNIC,INSTY
      JUCK=JUCK+1
      IF(ICOMP(MAR),NE,IMPRI(JUCK)) GO TO 5063
5062  CONTINUE
      ICONDO=ICONDO+1
      DO 5064 MARCOD=IOG,IOG+2
      IF(ICONDO,GE,10) GO TO 5065

```

```

5064 LYAUX(MARCOD)=NUMALF(1)
      IVES=IVES+1
      IREPE=IVES*4+1
      IOG=IOG+3
      LYAUX(IOG)=NUMALF(ICONDO+1)
      IOG=IOG+1
5070 IF(IBARRA.GE.KFINDO) GO TO 5066
      KNIC=INSTY+1
      IBARRA=IBARRA+1
      GO TO 5067
5066 JUCK=JLEMBR
      LEVA=4*KFINDO
      GO TO 5069
5065 LYAUX(IOG)=NUMALF(1)
      LYAUX(IOG+1)=NUMALF(1)
      ICONDO=ICONDO+1
      IREVER=ICONDO/10
      IRE=ICONDO-IREVER*10
      LYAUX(IOG+2)=NUMALF(IREVER+1)
      LYAUX(IOG+3)=NUMALF(IRE+1)
      IOG=IOG+4
      GO TO 5070
5063 JUCK=JUCK+1
      IF(IMPRI(JUCK),NE,IALFA(51)) GO TO 5063
      ICONDO=ICONDO+1
      GO TO 5071
5060 IBARRA=IBARRA+1
      IFIBAR(IBARRA)=MAR
      GO TO 5072
5069 IOG=IOG-1
      RETURN
      END

```

SUBROUTINE KTURN

```

C SUBROTINA DE CONVERSÃO PARA NÚMEROS DECIMAIS
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDUÇ
*O,INGRID,IDALIA,ISPTQ,KEPE,MONTE(9999),KIERRO,IMARÇ,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMEÇO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
ILHA=1
4055 DO 4050 LA=1,10
      IF(IMPRI(JUCK).EQ.NUMALF(LA)) GO TO 4051
4050 CONTINUE
      GO TO 4053
4051 IMPRES(ILHA)=NUMERO(LA)
      ILHA=ILHA+1
      JUCK=JUCK+1
      IF(ILHA.NE.5) GO TO 4055
      JUCK=IMPRES(1)*1000+IMPRES(2)*100+IMPRES(3)*10+IMPRES(4)

```

4053 RETURN
END

```

SUBROUTINE MLTLIS
C  SUBROTINA PARA PERCURSO DE ENTRADA "MULTIPLA"
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDU
?O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IFNDCA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMEÇO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IQG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGFRAL,LIN
IF(NIDADO(1),NE,IALFA(40)) GO TO 80
LEVY=0
IMPRIM=0
KTITO=1
GO TO 81
80 KTITO=0
IF(IMPRIM,NE,0) GO TO 83
KOB=KLI
GO TO 84
83 KOB=2
84 IMPRIM=1
DO 85 KLIST=KOB,80
IF(NIDADO(KLIST),EQ,IALFA(27)) GO TO 86
LEVY=LEVY+1
85 ICOMP(LEVY)=NIDADO(KLIST)
GO TO 81
86 LEVA=LEVY
81 RETURN
END

```

```

SUBROUTINE KDEF AU
C  SUBROTINA PARA ESTABELECIMENTO DE DEFAULTS DO SISTEMA
COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDU
?O,INGRID,IDALIA,ISPTO,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMEÇO,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IQG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
IF(IORG,NE,1) GO TO 210
IF(MIGRA,EQ,1) GO TO 201
IQG=IQG+1
LYAUX(IQG)=NUMALF(1)
LYAUX(IQG+1)=NUMALF(3)
LYAUX(IQG+2)=NUMALF(1)
LYAUX(IQG+3)=NUMALF(1)
IQG=IQG+3
GO TO 202
201 JLEMBR=JUCK

```

```

      JUCK=JUCK+1
      KL=1
207 DO 204 LE=1,10
      IF(IMPRI(JUCK),EQ,NUMALF(LE)) GO TO 203
204 CONTINUE
      GO TO 206
203 IMPRES(KL)=NUMERO(LE)
      KL=KL+1
      JUCK=JUCK+1
      IF(KL.NE.3) GO TO 207
      KEMBY=IMPRES(1)*10+IMPRES(2)
      CALL KONVTA
      CALL KALFTA
      IOG=IOG+1
      DO 208 LUST=IOG,IOG+KEMBY-1
208 LYAUX(LUST)=NUMALF(1)
      IOG=IOG+KEMBY-1
      JUCK=JLEMBR
      GO TO 202
210 IOG=IOG+1
      LYAUX(IOG)=NUMALF(1)
      LYAUX(IOG+1)=NUMALF(5)
      IOG=IOG+2
      DO 209 LIVRO=IOG,IOG+3
209 LYAUX(LIVRO)=NUMALF(1)
      IOG=IOG+3
      GO TO 202
206 JUCK=0
      KIMERA=1
      IGERAL=1
202 RETURN
      END

```

C*****PROGRAMA PRINCIPAL

C PROGRAMA PARA CONTROLE DE SUBPROGRAMAS

```

COMMON IBLOC,I,ICONT,IPEG,K,IMPRI(9999),IVET(80),IAL(4),INU
*M(8),IALFA(51),KFITA(12),KVATA(16),LISIN(14),LISEX(14),KDOC
*O,INGRID,IDALIA,ISPTD,KEPE,MONTE(9999),KIERRO,IMARC,LISTA(8
*0),ITOT,IENDLI(100),IENDCA,KONT(4),NUMALF(10),NUMERO(10),KA
*M,ICART(80),KPRI,KERO,KEVEDO,KID,NOTE(9999),KLAST
COMMON NIDADO(80),KIMERA,JUCK,NA,MIGRA,IORG,KLI,KOMECC,KOFI
*M,ICOMP(80),KARTY,KEMBY,LEVA,IOG,IMY,LYAUX(500),KAMEL(10),L
*UZ,IMPRES(5),JLEMBR,IFIBAR(50),IMPRIM,KTITO,LEVY,IGERAL,LIN
DATA IAL/'A','L','F','A'/,INUM/'N','U','M','E','R','I','C',
* 'O'/,LISIN/'L','I','S','T','A','I','N','C','L','U','S','I',
* 'V','A'/,LISEX/'L','I','S','T','A','E','X','C','L','U','S',
* 'I','V','A'/,NUMERO/0,1,2,3,4,5,6,7,8,9/,NUMALF/'0','1','2',
* '3','4','5','6','7','8','9'/,KFITA/'T','A','M','A','N','H',
* 'O','F','I','X','O'/,KVATA/'T','A','M','A','N','H','O','V',
* 'A','R','I','A','V','E','L'/,IALFA/'A','B','C','D','E','F',
* 'G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U',
* 'V','X','Y','Z',' ','*','0','1','2','3','4','5','6','7',
* '8','9','$','#','@','?','!','<','>','+','%','=','.'','/','

```

```

*1/1/
KONFER=0
DO 42 I=1,9999
42 IMPRI(I)=IALFA(27)
I=0
CALL INDTAM
KAPA=0
MAYS=0
NAYS=0
IPEG=0
IDALIA=0
IBRAN=0
ITAM=0
LIN=0
IBLOC=60
ICONT=1
KPRI=0
CALL LELIST
CALL IENDER
IF(KONFER.EQ.0) GO TO 16
80 IF(KDUCC.EQ.1) GO TO 16
KONFER=1
16 READ(8,11)(IVET(K),K=1,80)
11 FORMAT(80A1)
KIERRO=0
ICHAV=0
K=1
KPRI=0
ISUPR=0
IMARC=0
IF(IVET(K).NE.IALFA(29)) GO TO 13
DO 301 MINA=1,KLAST
301 MONTE(MINA)=NOTE(MINA)
CALL MLTIMP(KLAST)
GO TO 100
13 CALL NAMEL
IF(KID.EQ.1) GO TO 16
CALL ISAI(IALFA(29))
CALL ISAI(IALFA(29))
CALL KTIPO
IF(KIERRO.EQ.1) GO TO 16
KDUCC=0
CALL ICOGI
IF(KIERRO.EQ.1) GO TO 80
CALL ICOGI
IF(KIERRO.EQ.1) GO TO 80
INGRID=0
IDALIA=0
CALL LGRILL
IF(KIERRO.EQ.1) GO TO 16
IF(IDALIA.NE.0) GO TO 111
112 CALL LGRILL
IF(KIERRO.EQ.1) GO TO 16

```



```

        IF(ISPTO)130,111,130
130 CALL ISAI(IALFA(29))
111 IDALIA=0
        IF(KDUCC.NE.1) GO TO 940
        DO 240 L=1,4
240 MONTE(L)=IALFA(29)
941 CALL MLTIMP(4)
        GO TO 16
940 KAPA=KAPA+1
        KERO=IENDLI(KAPA)
        KEVEDO=0
        CALL KONVER
        IF(KEVEDO.EQ.0) GO TO 908
        CALL IERRO(46)
        GO TO 100
908 CALL KONALF
        GO TO 941
100 IGERAL=0
        LEVY=0
        IMPRIM=0
        IOG=0
        KLI=0
        KOMECO=0
        KOFIM=0
        MIGRA=0
        IORG=0
        NA=0
        LY=0
        JUCK=0
3090 KIMERA=0
3036 READ(8,3033)(NIDADO(INTER),INTER=1,80)
3033 FORMAT(80A1)
        IF(ITAM.EQ.0) GO TO 8020
        WRITE(5,8025)
8025 FORMAT(1H1)
        ITAM=0
8020 WRITE(5,3047)(NIDADO(INTER),INTER=1,80)
3047 FORMAT(1X,80A1,/)
        IF(IMPRIM.NE.0) GO TO 5010
4065 IF(NIDADO(1).NE.IALFA(42)) GO TO 3035
        IF(IGERAL.EQ.1) GO TO 3091
        IF(KIMERA.EQ.1) GO TO 3090
        GO TO 5080
3091 IOG=IOG+1
        IGERAL=0
        GO TO 5082
5080 DO 3071 ILA=1,IOG
3071 MONTE(ILA)=LYAUX(ILA)
        CALL MLTIMP(IOG)
5082 IOG=0
        KIMERA=0
        JUCK=0
        ITAM=1

```

```
GO TO 3036
3035 IF(NIDADO(1),EQ,IALFA(39)) GO TO 3040
      IF(KIMERA,EQ,1)GO TO 3036
      CALL KLAUDI
      IF(KIMERA,NE,0)GO TO 3036
      CALL NATIEFV
3061 KLI=KLI+1
      IF(NIDADO(KLI),NE,IALFA(27)) GO TO 3060
      IF(KLI,LT,80) GO TO 3061
      CALL KDEFU
      IF(KIMERA,EQ,1) GO TO 3036
      GO TO 3064
3060 IF(NIDADO(1),EQ,IALFA(30)) GO TO 5010
      GO TO (3062,3063),NA
3062 CALL KALFCK
      IF(KIMERA,EQ,1) GO TO 3036
      CALL LAUXER
      IF(KIMERA,EQ,1) GO TO 3036
      GO TO 3064
3063 IF(IORG,NE,1) GO TO 3065
      CALL NUMCHK
      IF(KIMERA,EQ,1) GO TO 3036
      CALL LAUXER
      IF(KIMERA,EQ,1) GO TO 3036
      GO TO 3064
3065 CALL LISTCK
      IF(KIMERA,EQ,1) GO TO 3036
      CALL LAUXER
      IF(KIMERA,EQ,1) GO TO 3036
      IF(LUZ,NE,1) GO TO 3064
5060 CALL LISCOD
      IF(KIMERA,EQ,1) GO TO 3036
3064 JUCK=JUCK+6
      GO TO 3036
5010 CALL MLTLIS
      IF(KTITO,EQ,1) GO TO 5060
      GO TO 3036
3040 IPEG=1
      I=I+1
      CALL ISAI(IALFA(27))
3037 CALL EXIT
      END
```