

ESTUDO E REALIZAÇÃO DE UM SISTEMA PORTÁTIL DE CADASTROS

ORIENTADO PARA USUÁRIO LEIGO EM COMPUTAÇÃO

2a. PARTE - UTILIZAÇÃO E ATUALIZAÇÃO

Marinilza Bruno de Carvalho

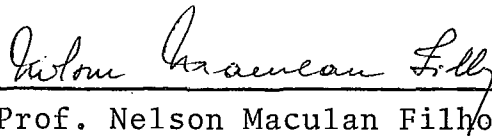
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.).

Aprovada por:

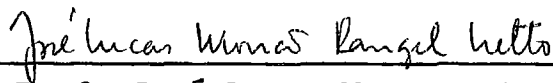


Prof. Pierre-Jean Lavelle

(Presidente)



Prof. Nelson Maculan Filho



Prof. José Lucas M. Rangel Netto

RIO DE JANEIRO , RJ - BRASIL

OUTUBRO DE 1977

BRUNO DE CARVALHO, MARINILZA

Estudo e Realização de um Sistema Portátil de
Cadastros Orientado para Usuário Leigo em Computa-
ção - 2a. parte - Utilização e Atualização
Rio de Janeiro 1976.

VIII, 88p. 29,7cm (COPPE-UFRJ, M.Sc., Engenharia
de Sistemas e Computação, 1977)

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

1. Assunto: Sistema de Cadastro I. COPPE/UFRJ

II. Estudo e Realização de um Sistema Portátil de
Cadastros Orientado para Usuário Leigo em Computa-
ção.

D E D I C A T Ó R I A

Ao

José Narciso

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

Ao Professor Pierre-Jean Lavelle, pela ori
entação.

Ao Professor Nelson Maculan Filho, pelo a
poio, incentivo e amizade.

Ao CNPq, pela bolsa concedida.

A Suely e Angela pelo trabalho de datilo -
grafia.

Aos Amigos da COPPE e do NCE que de alguma
forma contribuíram para a construção dessa tese e cuja citação
nominal me é tão difícil.

R E S U M O

Muito se tem feito no estudo de bancos de dados, visando a sistemas de grande complexidade e de grande porte.

A presente tese (segunda parte de um conjunto) apresenta uma outra visão do problema: a da maior simplificação sem perda de performance alcançável quando o problema pode se restringir a gestão de cadastros relativamente simples feita diretamente por pessoas leigas em computação, usando máquinas de pequeno porte.

Os enfoques são na independência do programa em relação à máquina usada, na facilidade de transportar esse programa de uma para outra máquina de estrutura diferente, e na modularidade desse programa.

S U M M A R Y

The study of data banks is made aiming at large and complex systems generally.

This thesis (second part of a two-part set) tries another look at this problem: the large simplification without loss of performance when the problem can be restricted to direct control of relatively simple cadasters by people not knowledgeable in computer science using mini computers.

The main parameters are machine independency, easy program transport to differing machines, and modularity.

Í N D I C E

	Páginas
AGRADECIMENTOS	iii
RESUMO	iv
ABSTRACT	v
INTRODUÇÃO	1
 CAPÍTULO I	 3
1. O Projeto - Características	3
1.1. Definição	3
1.2. Objetivos	7
1.3. Características	7
1.4. Acesso	10
 CAPÍTULO II	 14
2. Análise do Pedido de Pesquisa	14
2.1. Análise da Ordem de Pesquisa	17
2.1.1. Ordenar.....	19
2.1.2. Listar	20
2.1.3. Contar	22
2.2. Análise do Dado a Ser Pesquisado	22
2.2.1. Definição	22
2.2.2. Objetivos	23
2.2.3. Características	24
2.2.4. Sequencia de Operações	24

	Páginas
2.3. Análise da Condição de Pesquisa	25
2.3.1. Definição	25
2.3.2. Objetivos	25
2.3.3. Características	26
2.3.4. Etapas	26
CAPÍTULO III	30
3. Execução da Pesquisa	30
3.1. Procura dos Dados	32
3.2. Testes	36
3.3. Contagem	40
3.4. Impressão	42
3.5. Ordenação	51
3.6. Fim de Pesquisa	51
CAPÍTULO IV	55
4. Atualização	55
4.1. Características	58
4.2. Sequência de Operações	59
4.3. Fase Atua 1	60
4.4. Fase Atua 2	63
4.5. Fase Atua 3	66
4.6. Tipos de Atualização	70
4.6.1. Inserção	71
4.6.2. Retirar	72
4.6.3. Trocar	72

	Páginas
4.7. Observações	73
CONCLUSÕES	75
BIBLIOGRAFIA	76
ANEXO I - Lista de Erros	77
ANEXO II - Manual do Usuário	79
ANEXO III - Lista das Rotinas	86

I N T R O D U Ç Ã O

Muito se tem feito na área de computação sobre arquivos em disco e fita, e cada vez maior a necessidade de mão de obra especializada para a 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 os preparam para um novo tipo de trabalho. Tudo isto, quando não são feitos contratos de especialistas no ramo.

Muitas empresas relutam em implantar um novo sistema devido a tal 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 as necessidades de um usuário leigo em computação. Seguindo apenas as instruções de um manual, o usuário poderá utilizar e atualizar o sistema. Possui grande portabilidade, podendo ser implantado em vários tipos de máquina.

Dois arquivos lineares, gravados em fita , são elaborados, contendo cada um, uma tabela índice, que é sua própria descrição e a massa de dados codificados. Os dados são lidos em cartão, depurados e só então a montagem dos arquivos é feita.

A comunicação ao sistema pelo usuário é feita por cartão em formato livre e em linguagem corrente, seguindo apenas algumas instruções de um manual.

Os pedidos de informação podem ser uma listagem, uma contagem ou uma ordenação de qualquer ou quaisquer dados cadastrados e até mesmo sob condições estipuladas pelo usuário.

Na atualização o usuário pode inserir dados, retirar ou trocar outros dados.

Na Ordenação, depende da máquina em que estiver implantado, o sistema permite utilizar qualquer tipo de sort.

C A P Í T U L O II. O PROJETO - CARACTERÍSTICAS1.1. Definição

Através de "Montagem e Análise Crítica dos Cadastros", primeira parte de, Estudo e Realização de um Sistema Portátil de Cadastros Orientado para Usuário Leigo em computação um sistema de cadastros elaborados de forma tal que permita a montagem de qualquer tipo de cadastro, isto é, uma montagem que depende de como o cadastro é apresentado e não de quais são os seus dados.

Cadastros de diferentes tipos de dados podem ser processados desde que se constituam se INDICE + ARQUIVO, partes distintas e que juntas formam o cadastro (Fig. 2).

No índice, encontram-se todos os tipos de dados que constituem o cadastro. Se o dado é uma palavra ou um número, se essa palavra é um nome, um cargo ou um endereço; se o número é um código, telefone ou matrícula, todas estas especificações dos dados gravados estão no índice (Fig. 3). No index estão também as listas de códigos dos dados codificados (Fig. 4).

No arquivo tem-se os dados propriamente ditos codificados ou não (Fig. 4).

O projeto é justamente utilizar e atualizar um sistema de arquivos, elaborados na forma descrita, independente de seu conteúdo (Fig. 1).

A utilização consta de vários tipos de pesquisa. De uma forma geral o sistema de utilização projetado lista, conta e ordena dados do sistema de cadastros sob quaisquer condições (Fig. 9)-(Fig.1).

Quanto a atualização, permite inserir, retirar ou trocar dados do sistema de cadastros também sob quaisquer condições (Fig. 9)-(Fig.1).

O PROJETO

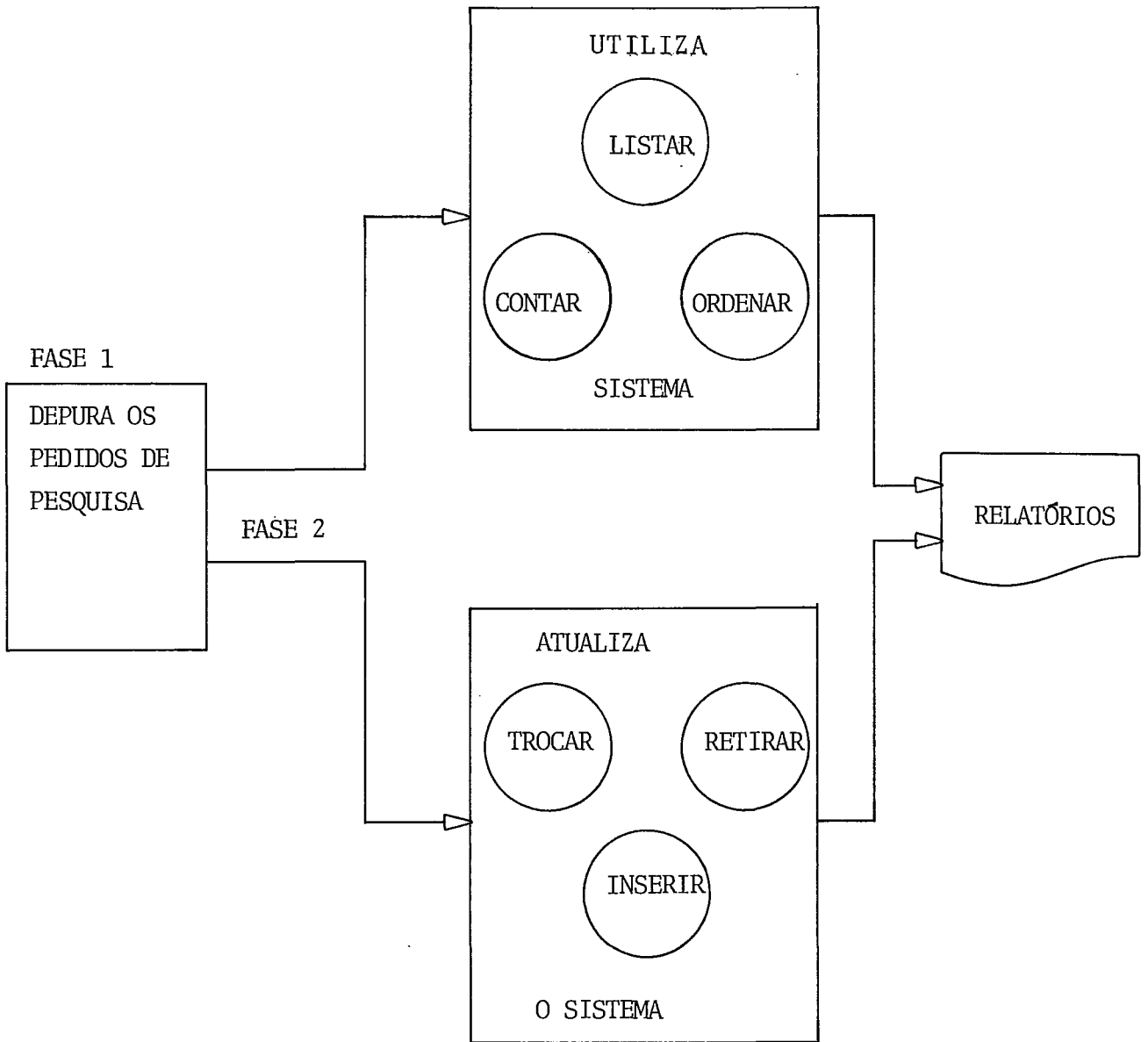


FIGURA 1

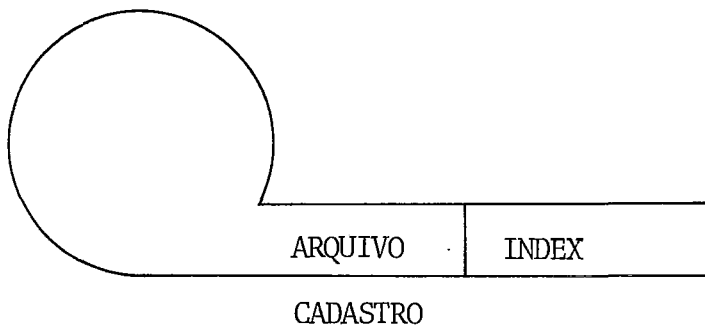


FIGURA 2

INDEX

Representa um registro lógico (RL) do arquivo codificado e cada RL possui um número de campos, todos neste formato.

NOME * A B C D E F G H I J K L M

NOME → NOME DO TIPO DE DADO
 * → DELIMITADOR
 A → BIT PARA A ORDEM LISTAR
 = 0 → NÃO LISTA
 = 1 → LISTA
 B → BIT PARA TESTAR
 = 0 → NÃO TESTA
 = 1 → TESTA
 C → INDICA SE O DADO DO ARQUIVO É
 ALFANUMÉRICO(=A) OU NUMÉRICO(=N)
 D E → INDICA SE O DADO É LISTA EXCLUSIVA
 (=LE), LISTA INCLUSIVA (=LI) OU NÃO É
 LISTA (=NÃO)
 F G → INDICA SE O DADO TEM TAMANHO FIXO(=TF)
 OU TAMANHO VARIÁVEL (=TV)
 H I → SE É FIXO, INDICA O TAMANHO
 (= 08,04,22,etc...) SE É VARIÁVEL (=00)
 JKLM → INDICA O ENDEREÇO DA LISTA DE CÓDIGOS.
 SE O DADO NÃO É CODIFICADO (=0000).

1.2. Objetivo

A utilização e atualização de cadastros é assunto bastante explorado atualmente, entretanto, apesar de conhecido e muito específico, toda firma que implanta um novo sistema, precisa de meses, por vezes, anos para formar pessoal especializado para operá-lo. Tal fato começou a ocorrer por demais os sistemas computacionais. A solução seria suprir as deficiências do usuário em computação com um sistema de fácil manipulação por parte deste. Com esse objetivo, o sistema foi projetado. Feito para um usuário leigo em computação, sua operação não exige linguagem específica, nem códigos tabelados, os pedidos de pesquisa são através de cartões perfurados pelo próprio usuário em linguagem corrente. É um sistema bem amplo que resolve o problema de operar qualquer cadastro por qualquer usuário.

1.3. Características

1.3.1. É um projeto elaborado para operar cadastros em geral, qualquer que seja seu conteúdo ou mesmo tamanho. O sistema é de certa maneira "cego" para o que se tem gravado no arquivo ou para o número de dados gravados.

1.3.2. Portabilidade é outra característica de grande importância. Não é um sistema projetado para um determinado tipo de computador. Possui condições próprias de ser implantado

INDEX DO CADASTRO

NOME*00A TV^CC000MATRICULA*00N TF080000FUNCAO*00
 NLETF040460CATEGORIA*00NLETF040530TEMPO. DE. SERVICIO
 *00N TF0400^OLOTACAO*CONLETF040565QUALIFICACAO*00
 NLITV000235D^DISCIPLINA.NOMEADD*00NLETF040375DISCIPL
 INA. OUTRAS*0^NLI TVCC0375= \$PRIMEIRO-GRAU
 -COMPLETO/SE^UNDO-GRAU-COMPLETO/LICENCIATURA-CURTA
 /LICENCIATUR^A-PLENA/ESPECIALIZACAO/MESTRADO/DOUTOR
 ADO/CURSO-DE-DIRETORIA/ \$MATEMATICA/PROCESSAMEN
 TO/PORTUGUES/GEOGRAFIA/HISTORIA/CIENCIAS/INGLES/FR
 ANCES/= \$DIRETOR/PROFESSOR/CHEFEDESERVICIO/CO
 ORDENADOR/ASSESSOR/INSPELOR/ \$SUBSTITUTO/EFETIV
 O/CONTRATADO/ \$SECRETARIADEEDUCACAO/FERREIRAVIA
 NA/PAULODEFR^ONTIN/INSTITUTODEEDUCA3AO//

ARQUIVO CODIFICADO

20SANDRA-DA-SILVA-REI S08046339-6040003040002040005
 040001120001^002000304000004000026RITA-DE-CASCIA-S
 ERR-JOGAIB08^35519-4040002040002040007040001120001
 000200030400^604000625LENIR-FERNANDES-DE--SOUZA080
 49077-604000^0400020400020400011200010002000404000
 104000117ANG^LA-MARA-SILVA080012057704000104000104
 001104000316^00100020004000804000304000312MARCOS
 SOARES08000136^8040004040002040015040002160001000200
 040008040007^4000719ADERBAL-DA-SILVEIRA08001264830
 4000304000304000504000412000100020003040000040000///

em vários tipos de máquina; devido em grande parte a linguagem Fortran na qual foi implementado e a sua forma modular, possibilitando uma estrutura de overlay.

1.3.3. A característica mais importante do projeto é justamente a facilidade de operação do sistema por parte do usuário. A utilização e atualização são feitas através de pedidos de pesquisa preparados pelo usuário em linguagem corrente seguindo determinadas regras de pontuação bem explícitas no manual do usuário. O campo de pesquisa é grande, exigindo do operador apenas o conhecimento do tipo do arquivo gravado, ortografia e linguagem.

1.3.4. O espaço de memória é outra característica de grande relevância. O fato do sistema poder operar cadastros de qualquer tamanho, lhe dá um campo de aplicação, tanto no ramo científico como comercial, bastante amplo, pois o espaço de memória muitas vezes se torna um enorme problema. Quanto a execução de um pedido de pesquisa, o espaço de memória necessário ao sistema é para o index do arquivo e o vetor janela.

Um cadastro é formado por registros lógicos, que é a sua unidade básica. Um arquivo de "n" funcionários, possui "n" registros lógicos, pois cada funcionário é um registro lógico. O index representa na memória, um registro lógico do cadastro, é portanto n vezes menor que o cadastro.

Quanto ao vetor janela é um vetor de tamanho fixo escolhido pelo usuário (por exemplo 1000 caracteres),

residente na memória do computador. Funciona como uma área de trabalho, recebendo os dados do arquivo por partes, a proporção que são requisitados na execução. Isto significa que todo o cadastro passa pela memória, mas de tantos em tantos caracteres por vez, tempo em que estará sendo pesquisado. Desse modo o espaço de memória utilizado é pequeno e não há problema quanto ao tamanho do cadastro.

Outro problema de memória, que surge muitas vezes, é ao implantar o sistema numa determinada máquina.

Em um mini-computador, provavelmente haveria problema de espaço, se o sistema não fosse modular, o que permite uma estrutura de overlay, possibilitando sua implantação tanto num computador de grande porte como num de médio ou um mini-computador.

1.4. Acesso ao Cadastro

Na utilização e atualização o acesso para toda e qualquer pesquisa é feito através do index, sem o qual o arquivo codificado seria total e completamente indecifrável. Ao receber um pedido de pesquisa, o sistema consulta o index do cadastro, para verificar a validade da pesquisa, quais os dados a serem pesquisados, onde se encontram e se estão codificados ou não. O arquivo codificado só é acessado durante a fase de execução.

O processo usado para acessar um dado do arquivo, consiste em dois apontadores que percorrem simulta -

neamente, o index e o arquivo, isto é, a cada instante existe um dado do index e um dado do arquivo sendo determinado, cada qual por um apontador (Fig. 5). Sendo o dado do index exatamente a descrição do dado apontado no arquivo.

Ao início de cada pesquisa os apontadores indicam os primeiros dados, tanto do index como do arquivo codificado.

Para se conseguir um dado do arquivo, inicialmente, ele deverá ser bem especificado no pedido de pesquisa. Tal especificação é, então, comparada com o primeiro dado do index, indicado. Caso a igualdade não se verifique, os apontadores são atualizados, passando a indicar os dados seguintes. Repete-se o processo até que a comparação com o index seja verdadeira, neste ponto o dado codificado indicado pelo apontador será o dado desejado.

A consulta ao índice é feita percorrendo - o totalmente e comparando cada um de seus dados com o dado sobre o qual se deseja a pesquisa. No caso de todo o índice ter sido comparado e nenhum dado encontrado, mensagens de erro são enviadas pelo sistema.

BITS - TESTAR - LISTAR

INDEX DO CADASTRO

N O M E * 0 0 ¢ ¢ T V 0 0 0 0 0 0 M A T R I C U L A * 0 0 N ¢ ¢
 LISTAR ↵ ↵ TESTAR LISTAR ↵ ↵ TESTAR

T F 0 8 0 0 0 0 F U N Ç Ã O * 0 0 N L E T F 0 4 0 4 6 0
 LISTAR ↵ ↵ TESTAR

C A T E G O R I A * 0 0 N L E T F 0 4 0 5 3 0 T E M P O .
 LISTAR ↵ ↵ TESTAR

D E . S E R V I Ç O * 0 0 N ¢ ¢ T F 0 4 0 0 0 0
 LISTAR ↵ ↵ TESTAR

FIGURA 6

C A P I T U L O IIII. ANÁLISE DO PEDIDO DE PESQUISA

A análise do pedido de pesquisa feito através de um ou mais cartões perfurados é a fase principal da pesquisa. Nesta fase, onde os cartões são lidos e analisados um de cada vez, todo e qualquer erro é anotado e mensagens específicas são enviadas ao usuário, possibilitando e facilitando a correção dos erros.

O pedido de pesquisa tem três partes que o caracterizam:

(i) ordem de pesquisa - são os sete primeiros caracteres não brancos.

(ii) tipo ou tipos de dados a serem pesquisados - todos os caracteres não brancos entre a ordem de pesquisa e o delimitador dois pontos (;).

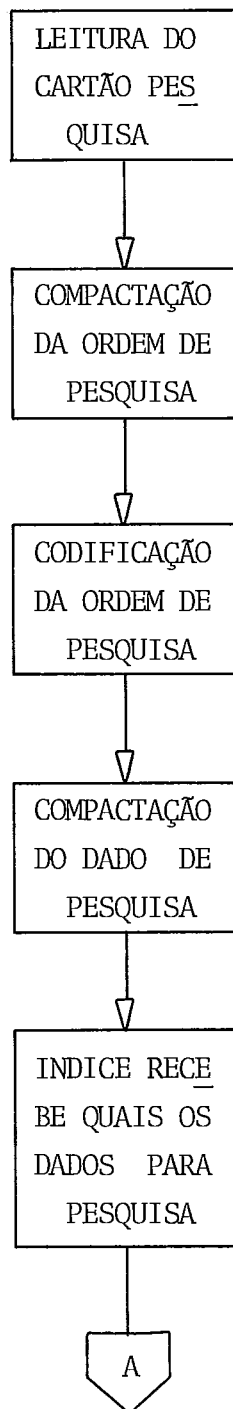
(iii) condição ou condições de pesquisa - todos os caracteres não brancos entre os delimitadores dois pontos (;) e ponto e vírgula (;).

O motivo do sistema utilizar esses delimitadores é, justamente, para facilitar o trabalho do usuário. Com esses símbolos, não há o problema de formato do cartão, para o pedido de pesquisa. O usuário pode perfurar o cartão em qualquer coluna e usar quantos cartões desejar.

Na leitura e depuração do pedido de pesquisa, três fases são distintamente analisadas (Fig. 7):

- (i) análise da ordem de pesquisa
- (ii) análise do dado a ser pesquisado
- (iii) análise da condição de pesquisa.

Supondo que não tenha ocorrido erros, ao fim desta fase o sistema terá anotado, qual a ordem de pesquisa, qual o dado a ser pesquisado, sob que condição ou condições de

DIAGRAMA DE BLOCO - FASE 1 -

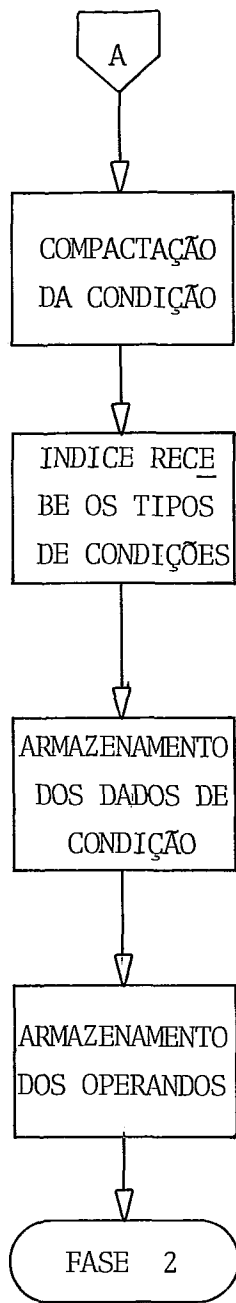


FIGURA 7

ve ser efetuada a pesquisa, etc.... também as codificações e decodificações necessárias terão sido feitas. Em suma o sistema estará totalmente preparado para a execução imediata da pesquisa.

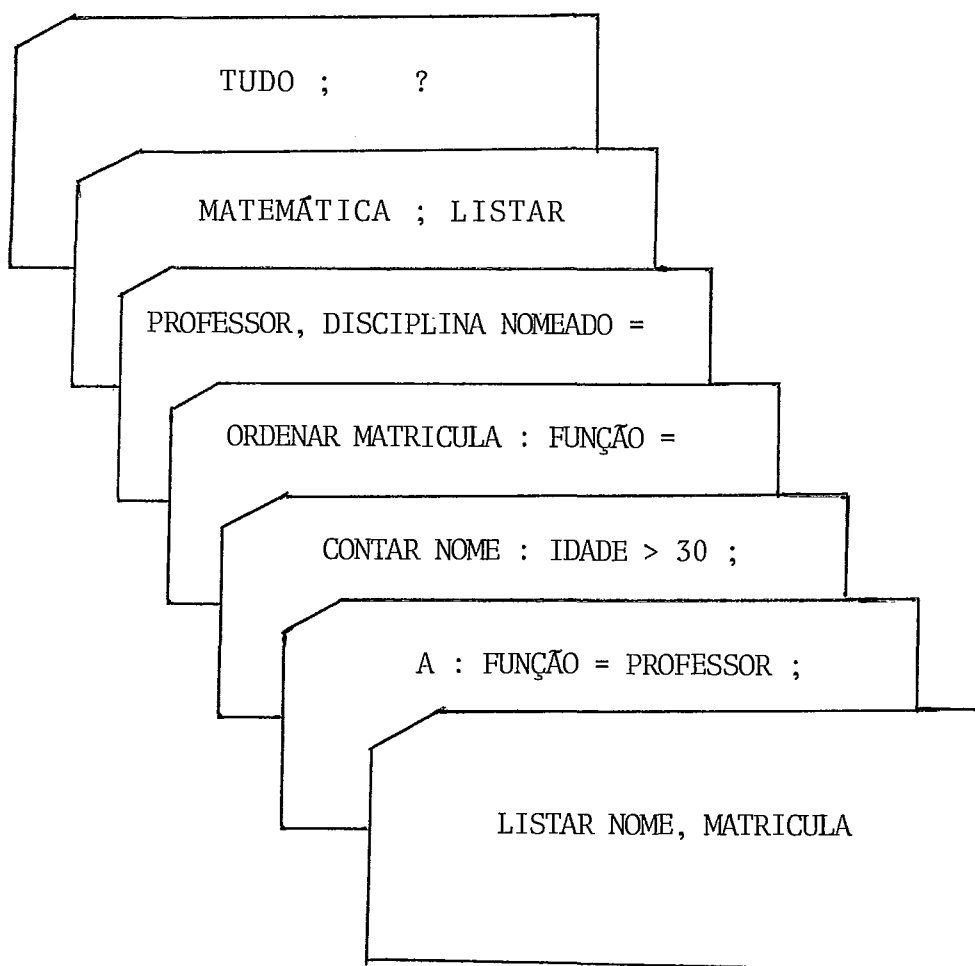
2.1. Análise da Ordem de Pesquisa

É o estudo dos sete primeiros caracteres não brancos, que representam a ordem de pesquisa.

O pedido de pesquisa recebido pelo sistema passa por uma fase de compactação dos sete primeiros caracteres não brancos. Isto se deve ao fato do usuário usar uma linguagem corrente e formato livre, desse modo, ele pode inserir uma quantidade incontável de brancos entre as letras do pedido de pesquisa. Com o auxílio da rotina de compactação, a facilidade permitida aos usuários, não dificulta o processamento. A determinação de sete caracteres não brancos é o número máximo de caracteres entre as ordens existentes.

Feita a compactação esses caracteres são comparados com as possíveis ordens de pesquisa. A ordem sendo aceita pelo sistema, seu código é armazenado e a análise do cartão continua, desta feita passando a analisar o dado a ser pesquisado.

Caso a ordem de pesquisa não confira com nenhuma das ordens existentes, a mensagem "ORDEM ERRADA" é enviada, o contador de erros é acrescido e a depuração do

EXEMPLO DE VÁRIOS PEDIDOS DE PESQUISAFIGURA 8

cartão continua. Evidentemente, no caso de uma ordem certa , perfurada errada, será uma ordem errada, não constando do sistema.

2.1.1. Ordenar

Se a pesquisa pedida for uma ordenação, existem algumas diferenças no processamento, durante a análise da ordem de pesquisa.

Uma vez que a ordem foi aceita pelo sistema, seu código é comparado com o código de "ordenar". Caso não confira, o sistema procede como descrito no item anterior, mas se o código é de ordenação, então o sistema troca esse código para o código da ordem LISTAR e liga uma chave de ordenação. Isto se deve ao fato de que uma pesquisa de ordenar dados sob alguma condição, significa, em primeiro lugar separar os dados que satisfazem a condição, e em segundo, ordená-los. Desse modo, toda ordenação, implica numa pesquisa de listagem e posteriormente numa ordenação dos dados a serem listados. Assim sendo, o sistema transforma a pesquisa ordenar em listar, ao fim da qual, a chave de ordenação é testada. Se estiver ligada, a ordenação é processada e em seguida a listagem dos dados é feita. Se não estiver ligada, simplesmente a listagem é feita.

A pesquisa ordenar pode-se apresentar como (Fig. 9):

a. Ordenar um tipo de dado sob uma ou mais condições. No caso de várias condições, o dado é consi-

derado somente se todas as condições forem satisfeitas.

b. Ordenar um tipo de dado, sempre. Neste caso, não há condição e todos os dados do tipo referido na pesquisa são ordenados.

2.1.2. Listar

Para a ordem de pesquisa listar, haverá um bit aceso em cada campo do index que se queira listar (Fig. 6). Desse modo no momento da execução as rotinas de teste saberão qual o dado a ser listado.

Um pedido para listar pode se apresentar como:

a. listar um ou vários tipos de dados sob determinada ou determinadas condições (Fig. 9). Em caso de várias condições, o dado é listado somente quando todas as condições forem satisfeitas.

b. listar um ou mais tipos de dados, sempre. Neste caso não há condição a ser testada, de modo que a listagem é feita para todos os dados do tipo referido na pesquisa.

c. listar tudo. Este é o caso de um dump de memória, onde o arquivo é inteiramente decodificado e listado.

A listagem de dump, não tem o formato das listagens comuns. Trata-se de um arquivo completo, uma listagem de grandes proporções, muitos dados num mesmo item, etc...

EXEMPLOS DE PEDIDOS DE PESQUISA

LISTAR NOME : FUNÇÃO = PROFESSOR ;

LISTAR NOME , MATRICULA : IDADE > 30 ,
FUNÇÃO = SERVENTE ;

LISTAR NOME : SEMPRE ;

LISTAR MATRICULA , FUNÇÃO : SEMPRE ;

LISTAR TUDO ;

CONTAR NOME : FUNÇÃO = PROFESSOR ;

CONTAR MATRICULA : FUNÇÃO = DIRETOR ,
TEMPO . DE . SERVIÇO > 10 ;

CONTAR NOME : SEMPRE ;

ORDENAR MATRICULA : FUNÇÃO = PROFESSOR ;

ORDENAR IDADE : TEMPO . DE . SERVIÇO > 15 ,
FUNÇÃO = DIRETOR ;

ORDENAR NOME : SEMPRE ;

FIGURA 9

Desse modo o relatório de dump é feito por registros, isto é, são listados sequencialmente os dados de cada registro do arquivo (Fig. 17).

2.1.3. Contar

Dos tres tipos de pesquisa, "contar" é a mais rápida. Tanto no tempo de processamento, pois não precisa armazenar nenhum dado, como no tempo de E/S pois não tem relatório impresso. A pesquisa contar, depois de executada, apenas imprime o título da pesquisa e o valor correspondente ao número de dados que satisfazem as condições dadas.

Um pedido de contar pode-se apresentar como (Fig. 9):

a. contar um ou mais tipos de dados sob uma ou mais condições. Neste caso, tal como, na pesquisa listar, para várias condições, o dado é contado quando todas as condições forem satisfeitas.

b. contar um ou mais tipos de dados, sempre. Esta pesquisa, não possui condição alguma. Todos os dados do tipo referido são contados.

2.2. Análise do Dado a ser Pesquisado

2.2.1. É o estudo dos caracteres do pedido de pesquisa contidos entre a ordem de pesquisa e o delimitador dois pontos(Fig.

8). Esses caracteres, que, representam o dado a ser pesquisado, são analisados quando-a sintaxe, com relação a pontuação, ortografia e quanto a lógica do sistema, com relação a existência ou não dos dados de pedidos no sistema. Também a falta do delimitador é analisada e neste caso, um erro é detetado . A presença de qualquer erro, origina a impressão imediata da mensagem de erro respectiva.

2.2.2. Os objetivos deste estudo são para facilitar tanto o processamento por parte do sistema como a operação do mesmo por parte do usuário.

Para o sistema o objetivo é evitar que o pedido de pesquisa chegue até a fase de execução com erros no próprio pedido, que impedirá, evidentemente a execução completa da pesquisa. Quando a fase de execução tem início, o cadastro de dados gravados começa a ser manipulado pelo sistema. Isto significa dizer que a fita ou o disco no qual o arquivo está gravado começará a ser lida ou lido. O caso do sistema de tetar erros que impessam a execução completa, depois desta ter sido iniciada, significa que o cadastro em fase de leitura será interrômido e a execução cancelada. Se é um arquivo em fita, esta deverá ser reemrolada para dar início a nova pesquisa. Se é em disco também o processo de leitura volta ao início.

Desnecessário se torna dizer, que o fato dos erros serem detetados antes do início da execução proporciona maior segurança e rapidez no processamento.

Há também erros de execução, erros que a parecem somente durante tal fase. Contudo esses erros são, em sua grande maioria, erros de sistema, e cuja existência é pouquíssimo provável. São erros que constam da lista de erros, tem suas mensagens específicas e toda a sequência normal de um erro ao ser detetado. A idéia do erro de sistema, por mais improvável que seja sua existência, se deve ao fato, de que este projeto foi feito de forma que nenhuma pergunta ficasse sem resposta por mais absurda ou inocente que fosse. Qualquer hipótese foi prevista, mesmo que nunca venham a acontecer.

Para o usuário, o objetivo da análise do dado a ser pesquisado é o mais simples e prático. Tem como finalidade indicar ao usuário, porque a pesquisa não foi efetuada e qual o erro por ele cometido, dando-lhe possibilidades de corrigi-lo e em menos tempo, uma vez que as mensagens específicas traduzem qual o erro detetado.

2.2.3. A característica dessa análise confunde-se com um de seus objetivos que é justamente permitir ao usuário, uma correção rápida de seus erros.

2.2.4. A depuração do dado a ser pesquisado começa com a compactação dos caracteres a serem estudados. Todos os brancos são retirados formando uma sequência de caracteres não brancos que compõem o tipo ou os tipos de dados a serem pesquisados. Numa etapa seguinte, esta sequência é comparada com todos os dados do index do arquivo. Uma vez que no index encontram-se todos os tipos de dados existentes.

Na hipótese de vários dados a serem pesquisados, são testadas também, as vírgulas entre esses dados, cuja falta ocasionam erros, impedindo a execução. Cada dado do pedido de pesquisa é comparado com todos os dados do index.

Durante a fase de comparação os dados do index que conferem com os dados de pesquisa são assinalados com o código da ordem de pesquisa pedida. Desse modo ao término desta etapa, no index estarão notados os dados que devem ser listados, contados ou ordenados.

Se nenhum dado do index for assinalado, significa erro no pedido de pesquisa, o que acarreta, acréscimo no contador de erros e o envio da mensagem específica.

2.3. Análise da Condição de Pesquisa

2.3.1. É o estudo dos caracteres não brancos, do pedido de pesquisa, contidos entre os delimitadores dois pontos (:) e ponto e vírgula . Esses caracteres representam a ou as condições de pesquisa (Fig. 8).

É uma análise sintática, ortográfica e de conteúdo, com relação ao cadastro gravado.

2.3.2. Os objetivos são vários, o mais prático inclusive, é a deteteção de erros do pedido de pesquisa, antes de se iniciar a execução da mesma, dando ao usuário uma avaliação completa do seu pedido, pois esta é a última fase de depuração dos

cartões de pesquisa.

A coerência dos dados de condição com o cadastro, é outro objetivo deste estudo bem como armazenar o dado condicionado, para futuros testes, marcar no index que tipo de dado será testado e fazer as devidas decodificações e codificações.

2.3.3. A característica básica é o fato desta análise de condições ser feita, sem testar a condição. É uma análise quanto as possibilidades de execução da pesquisa. Os dados de condição são testados para verificar se satisfazem as exigências operacionais do sistema e não para verificar as condições ditadas pelo usuário. As condições do usuário uma vez, satisfazendo a todas as condições do sistema, serão então pesquisadas e processadas.

Há também o número de condições. O sistema permite no máximo dez (10) condições para cada pesquisa. O que supos ser um limite bastante razoável. Um número de condições superior acarretará erro, mensagem de erro e interrupção na execução.

2.3.4. A fase de análise das condições possui tres (3) etapas (Fig.7):

- i) marcação no index
- ii) reconhecimento e armazenamento do operador
- iii) armazenamento do dado de condição.

i) Marcação do Index

Iniciando a análise da condição de pesquisa, o sistema faz a compactação dos caracteres não brancos contidos entre os delimitadores dois pontos e ponto e vírgula, caracteres estes que representam as condições da pesquisa. Uma vez que cada condição é constituída de tipos de dados, operadores e dados reais (Fig. 9), a sequência de caracteres compactados de uma ou mais condições apresenta-se na mesma forma. Assim sendo o sistema passa a comparar os primeiros caracteres (que representam o tipo de dado) com os dados do index a fim de reconhecer qual o tipo de dado da condição. Feitas as comparações sem conferir com nenhum dado do index, a mensagem de "dado não existente" é indicada, o contador de erro é acrescido e a depuração continua. No caso do tipo de dado conferir com algum dado do index, o bit de testar do dado do index em questão é assinalado (Fig. 6). Essa marcação é feita para facilitar a execução, pois quando esta tiver inicio bastará ao sistema verificar qual o dado do index que possui a marcação para teste. Somente estes dados deverão ser testados e pesquisados, o que evita maior perda de tempo, tanto de execução como de entrada e saída.

ii) Reconhecimento e Armazenamento do Operador

O sistema permite a utilização de tres operadores a saber: maior (>), menor (<) ou igual (=).

Uma vez que o tipo de dado já foi analisado, o sistema pesquisa se o caracter seguinte é um dos tres

operadores. Em caso afirmativo, o operador é armazenado e o sistema passa a etapa seguinte. No caso de não conferir, a mensagem adequada é enunciada, o contador de erros é acrescido e a depuração prossegue.

iii) Armazenamento do Dado de Condição

Nesta etapa o sistema armazena o dado de pesquisa num vetor determinado.

O processo de armazenamento desse dado utiliza um conjunto de tres vetores indexados em cadeia, isto é, uma composição de funções. O motivo desse processo, a princípio confuso, é devido ao fato do sistema permitir várias condições de pesquisa, desse modo, podem haver até 10 (dez) dados de pesquisa a serem armazenados, e que precisam ser indicados, que dado corresponde a tal condição, uma vez que a execução será processada num tempo posterior.

Os vetores indexados funcionam da seguinte maneira:

Sejam V1, V2 e V3 - vetores internos

Sendo V1 e V2 numéricos e V3 alfanumérico. No vetor V1 são armazenados sequencialmente todos os índices dos bits testar dos campos marcados no index. No vetor V3 são armazenados, também sequencialmente, os dados reais de condição, separados por (/) barra no caso de mais de uma condição. No vetor V2 são armazenados, também sequencialmente, os índices dos primeiros caracteres de cada dado, armazenado em V3.

Os vetores V1 e V2 variam juntos, pois

seus contadores representam o número de condições. Dessa forma uma condição como:

..... : NOME = MARIA LUCIA, FUNÇÃO = PROFESSOR ;

Sabendo que os índices dos bits testar dos dados NOME e FUNÇÃO do index são respectivamente 'a' e 'b', teríamos os vetores.

V1 (1) = a	V1 (2) = b
1a. condição	2a. condição
V2 (1) = 1	V2 (2) = 1 2
1a. condição	2a. condição
V3 (1) = M	V3 (12) = P
V3 (2) = A	V3 (13) = R
V3 (3) = R	V3 (14) = 0
V3 (4) = I	V3 (15) = F
V3 (5) = A	V3 (16) = E
V3 (6) = L	V3 (17) = S
V3 (7) = U	V3 (18) = S
V3 (8) = C	V3 (19) = 0
V3 (9) = I	V3 (20) = R
V3 (10) = A	V3 (21) = /
V3 (11) = /	

Com esta apresentação o sistema permite o armazenamento de inúmeros dados. Ao fim do qual verifica-se a existência de mais alguma condição de pesquisa e se não ultrapassa o limite de 10 condição, o que acarretaria erro para o sistema. Também o falta de vírgula, separando as condições a carretaria erro, impedindo a execução da pesquisa. Feitas to das essas verificações o sistema repete o processo de análise, no caso de haver mais condições, ou passa a execução propriamente dita.

C A P I T U L O I I IIII. EXECUÇÃO DA PESQUISA

Uma vez que a fase de leitura e depuração tenha sido completada, e nenhum erro detetado, tem início então, a execução da pesquisa, fase em que os dados do arquivo são procurados, as condições são testadas e os relatórios elaborados (Fig. 10).

As características principais residem no fato do sistema só entrar nesta fase, se nenhum erro anterior houver sido detetado e também o fato de que somente neste ponto é que o arquivo codificado é manipulado. Até então, o sistema trabalha somente com os cartões que dão entrada com os pedidos de pesquisa. Depois de verificar que não existe erro, o sistema passa então a ler o arquivo codificado, no qual procura os dados para as pesquisas, faz os testes e mais uma vez testa a presença de erros. Nenhum sendo encontrado, o relatório então será elaborado dando por terminada a pesquisa. Esses fatos concorrem para um processamento mais

UTILIZAÇÃO

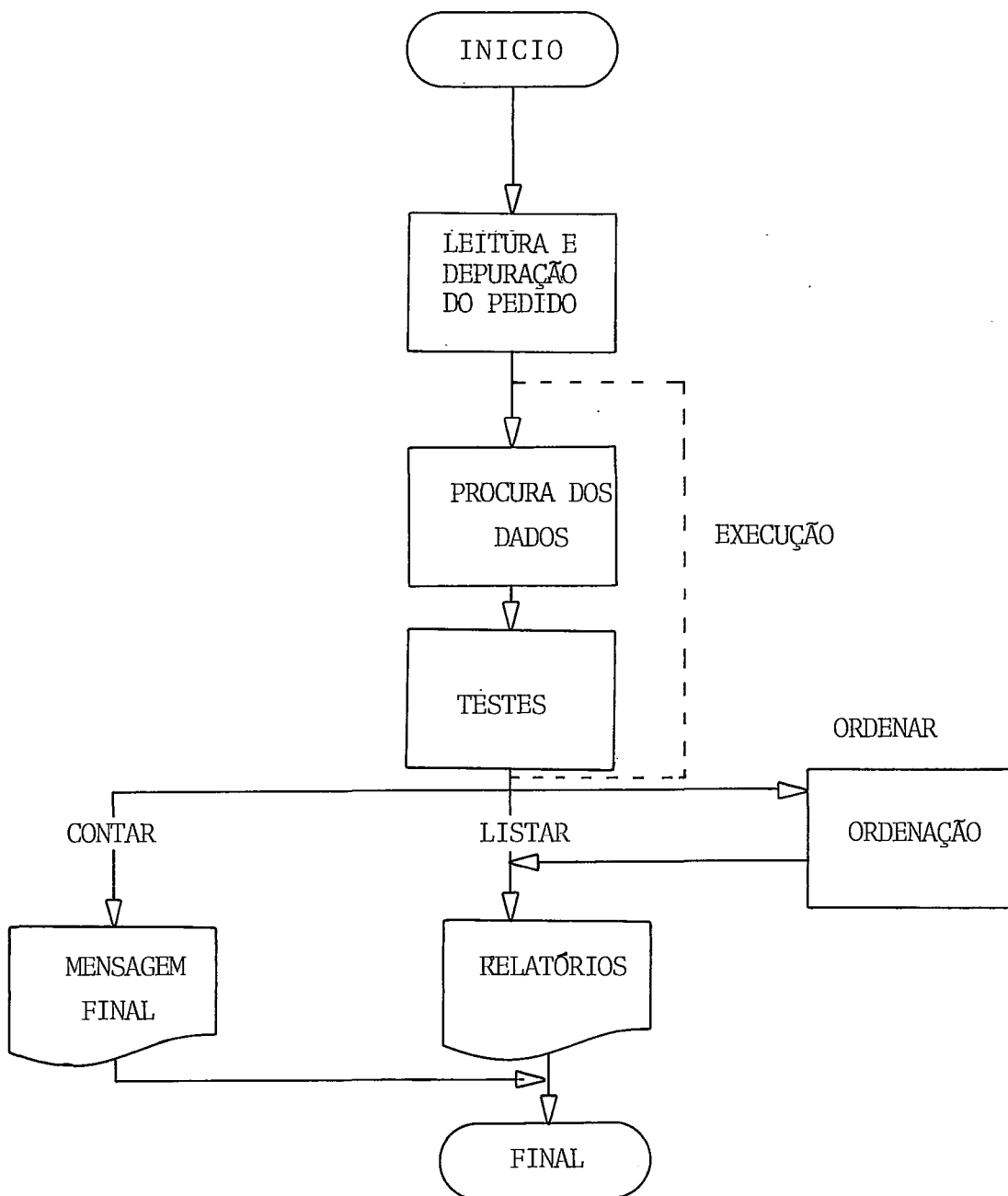


FIGURA 10

rápido e mais eficiente pois não se perde tempo em ler um arquivo, caso a pesquisa não possa ser efetuada e o mesmo não corre o risco de ser lido e relido sem necessidade, principalmente porque na maioria dos casos os arquivos são em fitas, cuja inconfiabilidade é bastante conhecida.

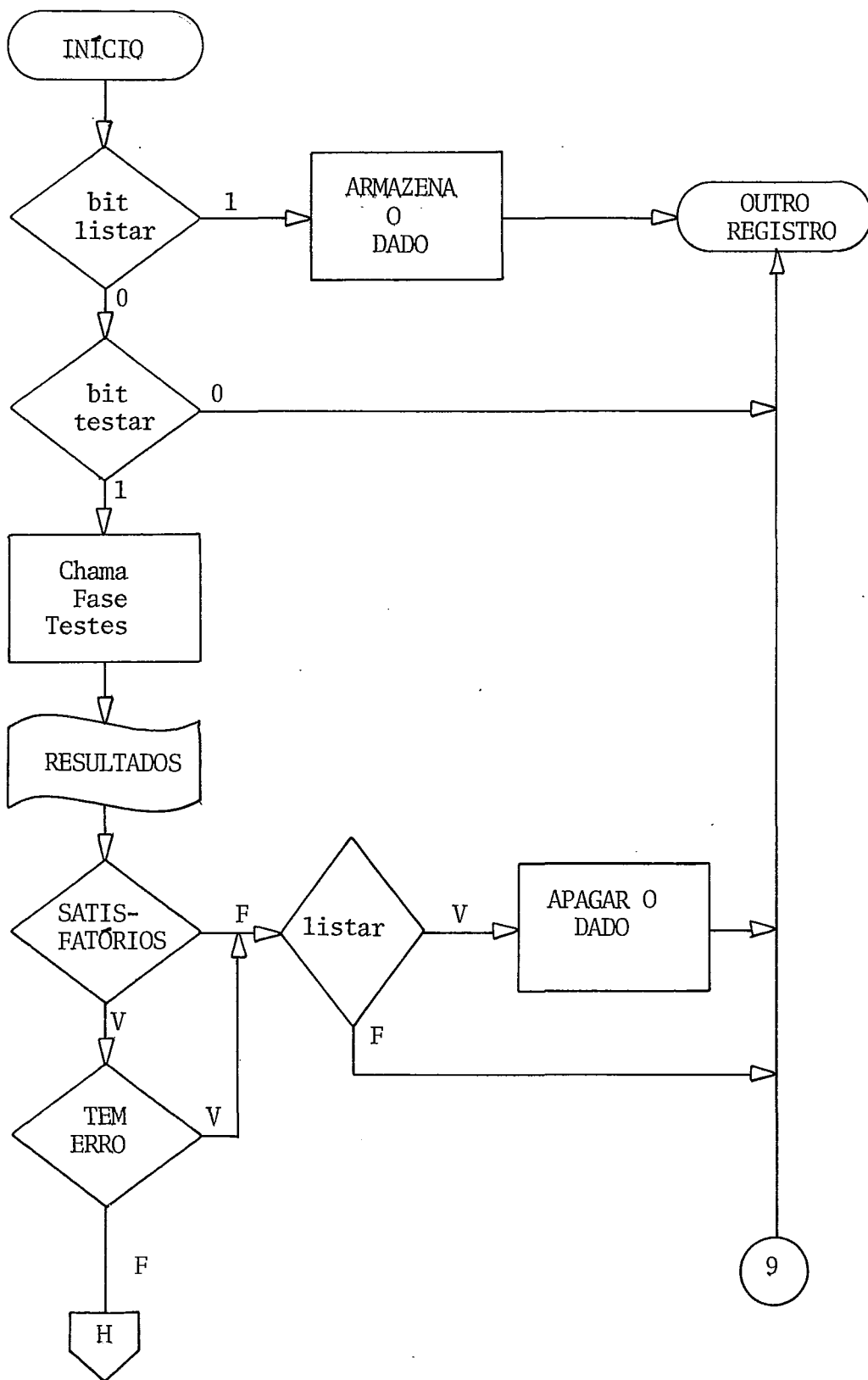
3.1. Procura dos Dados

Fase em que o arquivo codificado é lido , verificando-se qual o campo ou campos a serem listados, contados, ordenados, atualizados ou testados com outros dados (Fig. 11).

Tem por finalidade apanhar o dado do arquivo. Este é o ponto base da procura dos dados, sendo também a característica mais importante, pois é a única fase , exceto a atualização, que tem acesso ao arquivo codificado .

A procura dos dados é feita através do index, uma vez que este, possui assinalado cada campo (item 2) que deverá ser testado, listado ou contado. É conveniente lembrar que a pesquisa de ordenar é transformada em listar e somente na elaboração do relatório é que a ordenação é feita . Quanto a atualização o index assinala todos os campos a serem testados, somente, e a atualização propriamente é feita posterior aos testes, caso sejam satisfatórios.

Desse modo o sistema percorre simultaneamente o index e o arquivo codificado, onde cada qual possui um apontador que a cada instante indica os dados correspon-

FASE DE PROCURA DE DADOS

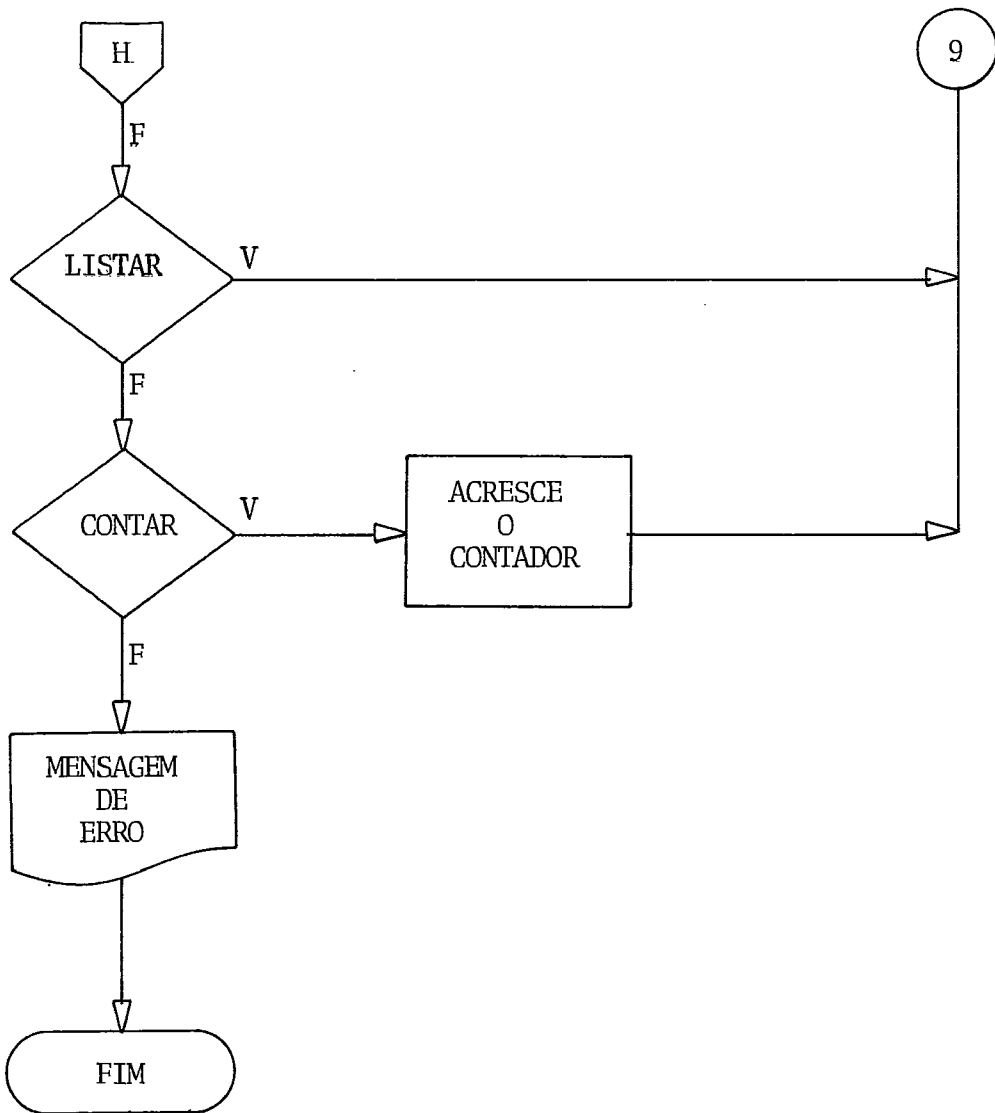


FIGURA 11

- se for listar, o sistema passa para testar o registro seguinte, uma vez que o dado já foi armazenado e o relatório só é elaborado quando todos os registros do arquivo tiverem sido pesquisados.

Se o resultado não for satisfatório e a ordem de pesquisa for listar, aquele dado armazenado é então apagado do vetor, o mesmo acontecendo quando os resultados forem satisfatórios mas for constatada a presença de erros.

Desse modo a "procura dos dados" é a fase mais importante da execução, pois é quem manuseia com os arquivos codificados, apanhando os dados e armazenando-os quando necessário. É também a partir desta fase que se tem início a fase de testes. Sendo ainda responsável pelo recebimento dos resultados e sua posterior análise.

3.2. Fase de Testes

São rotinas que testam os dados arquivados codificados ou não, numéricos ou alfanuméricos com as condições da pesquisa.

É uma fase cuja característica primordial consiste na rapidez e eficiência dos testes. É essencialmente prática, uma vez que toda a preparação já foi feita. Recebendo os dados, o sistema nesta fase, decodifica-os ou não, conforme o caso, e os compara com as condições já armazenadas e preparadas (item 2.3).

FASE DE TESTES

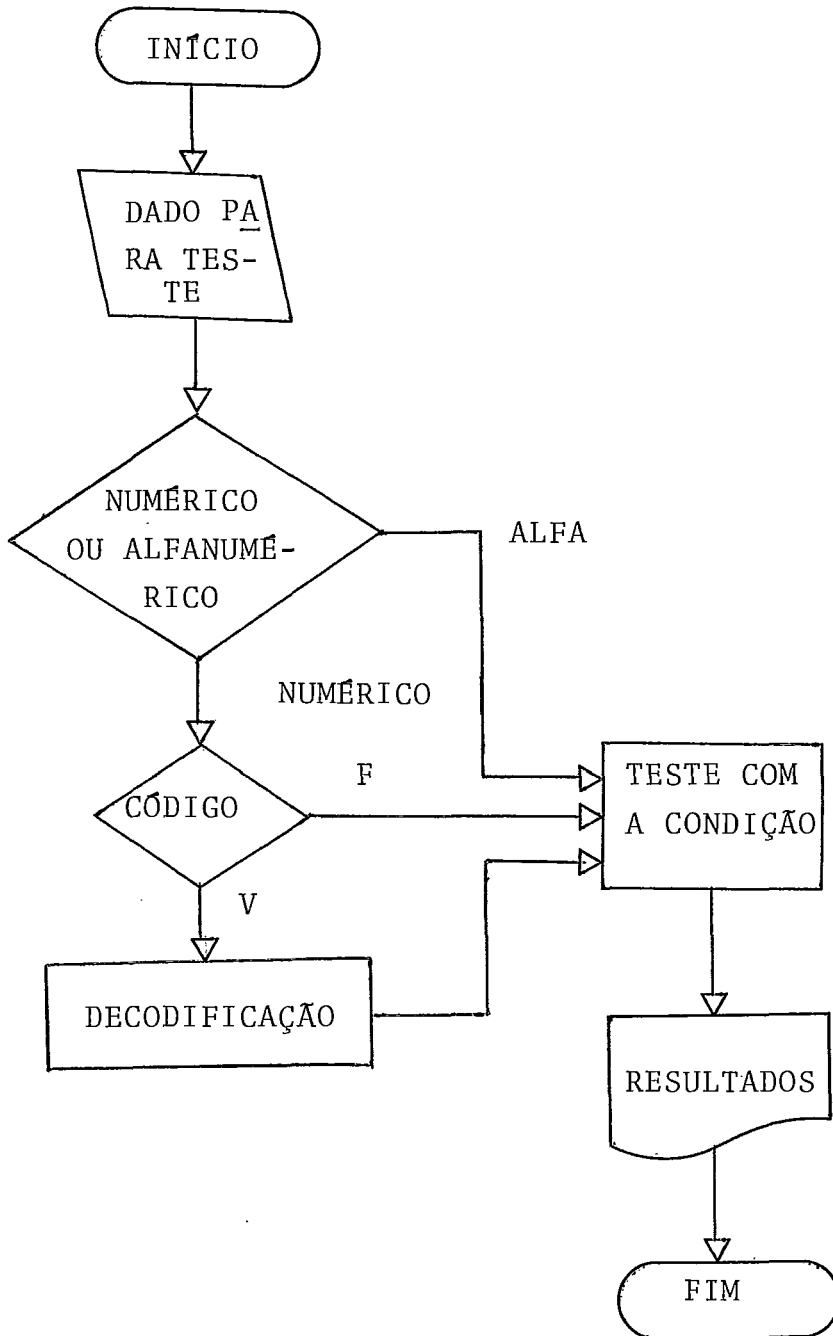


FIGURA 12

Esta fase tem início quando chamada pela fase de procura dos dados, da qual recebe o dado a ser testado e para a qual envia seus resultados (Fig. 12).

O dado a ser testado tem tres variações, podendo ser um dado alfanumérico, um número comum ou um número codificado. Nos dois primeiros, dizemos que o dado é real e a comparação com as condições processa-se imediatamente. No último caso, temos um dado codificado que é decodificado e então comparado com as condições. Desse modo vemos que os testes começam verificando se o dado é numérico ou alfanumérico, depois se é codificado ou não e então a comparação com a condição de pesquisa é efetuada. Uma vez que o dado está pronto para a comparação, o sistema apanha a condição. Para tanto, é utilizado o processo dos tres vetores, já exposto no item 2.3, que situam qual a condição para cada dado. O vetor V1 (Fig.13) contém o índice do bit testar, o vetor V2 contém o índice do 1º caracter da condição no vetor V3. Os índices de V1 e V2 são iguais para cada condição. Desse modo o sistema compara o vetor V1 com o bit testar do campo referente ao dado. Quando conferirem teremos o ponto exato onde começa a condição para ser testada.

No exemplo da Fig. 13 o sistema deve testar os campos "FUNÇÃO" e "TEMPO.DE.SERVIÇO", assim a fase de testes recebe o dado relativo a FUNÇÃO verifica que é código, decodifica-o e prepara-se para apanhar a condição. Então, testa V1 até encontrar o índice do bit testar de FUNÇÃO. Como o índice de V1 é o mesmo para V2 em cada condição, ao encontrar em V1(I), o bit de testar e usando o mesmo índice I, pa-

INDEX DO CADASTRO

NOME*00A TV^00000MATRICULA*00N TF080000FUNCAO*00
 NLETF040460CATEGORIA*00NLETF040530TEMPO.DE.SERVICO
 *00N TF0400^0LOTACAO*00NLETF040565QUALIFICACAO*00
 NLITV000235D^DISCIPLINA.NOMEADO*00NLETF040375DISCIPL
 INA.OUTRAS*G^NLITV000375= \$PRIMEIRO-GRAU
 -COMPLETO/SEGUNDO-GRAU-COMPLETO/LICENCIATURA-CURTA
 /LICENCIATURA-PLENA/ESPECIALIZACAO/MESTRADO/DOCTOR
 ADO/CURSO-DE-DIRETORIA/ \$MATEMATICA/PROCESSAMEN
 TO/PORTUGUES/GEOGRAFIA/HISTORIA/CIENCIAS/INGLES/FR
 ANCES/= \$DIRETOR/PROFESSOR/CHEFEDESERVICO/CO
 ORDENADOR/ASSESSOR/INSPECTOR/ \$SUBSTITUTO/EFETIV
 O/CONTRATADO/ \$SECRETARIADEEDUCACAO/FERREIRAVIA
 NA/PAULODEFRONTIN/INSTITUTODEEDUCA3AO//

PESQUISA → LISTAR NOME, MATRÍCULA:
 TEMPO.DE.SERVIÇO>15,FUNÇÃO=DIRETOR;

CAMPOS A SEREM TESTADOS:

TEMPO DE SERVIÇO - BIT TESTAR =103

FUNÇÃO - BIT TESTAR = 50

VETOR V1 CONTEM OS ÍNDICES DOS BITS TESTAR

V1(1) = 103 V1(2) = 50

VETOR V2 CONTEM OS ÍNDICES DO 1º CARACTER DE CADA CONDIÇÃO NO VETOR V3.

V2(1) = 1 V2(2) = 4

OS ÍNDICES DE V1 e V2 SÃO IGUAIS PARA CADA CONDIÇÃO.

VETOR V3 CONTEM AS CONDIÇÕES

V3(1) = 1	V3(4) = D	V3(7) = E	V3(10) = R
V3(2) = 5	V3(5) = I	V3(8) = T	V3(11) = /
V3(3) = /	V3(6) = R	V3(9) = 0	

ra V2, temos em V2(I), o índice de V3 que contém o primeiro caracter da condição referente ao dado.

Se V1(I) = 50 (bit testar) \rightarrow I = 2 logo
 V2(2) = 4 é o índice de V3 onde começa a condição. Então
 V3(4) = D é o primeiro caracter da condição.

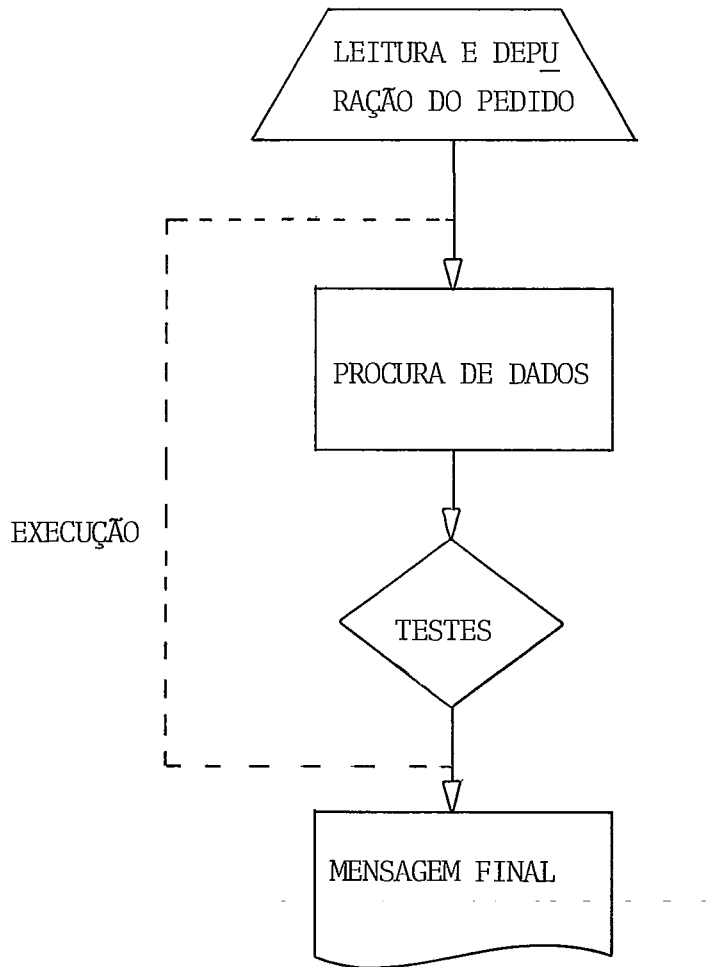
Encontrada a condição, a comparação com o dado tem início. É bom lembrar, que ambos devem ser do mesmo tipo, isto é, alfanumérico ou numérico. No caso de serem numéricos, procura-se no vetor V4 o elemento de índice I, o mesmo usado para V1 e V2. Este vetor contém os operandos das condições usando os mesmos índices que V1 e V2. Conhecendo o operando da condição, o sistema testa se o dado é maior, menor ou igual a condição, de acordo com o operando encontrado.

Para o caso dos dados serem alfanuméricos a comparação é feita testando-se caracter por caracter, uma vez que neste caso o resultado satisfatório, e evidentemente, é que a condição confira com o dado. Terminados os testes, sejam quais forem os resultados o sistema retorna a fase de procura dos dados, onde é feita análise dos resultados.

3.3. Contagem

A pesquisa de contar é a mais simples e rápida, uma vez que não existe relatório a ser elaborado, nem dados para armazenar. As codificações ou decodificações

CONTAGEM

FIGURA 14

feitas são exclusivamente para os testes.

A contagem consiste em ler e depurar o pedido de pesquisa, procurar os dados, testar e contar aqueles que satisfazem as condições (Fig. 14).

O resultado da pesquisa é dado numa única mensagem do tipo "EXISTEM X DADOS QUE SATISFAZEM AS EXIGÊNCIAS", onde X é o número encontrado.

Essa pesquisa segue a linha do projeto com fácil operação, rapidez e eficiência na execução. Os tipos de contagem são vários, dependendo do cadastro arquivado (veja manual).

3.4. Impressão

A parte mais prática do projeto, é a resposta aos pedidos de pesquisa com a solução dos problemas do usuário.

A impressão elabora os relatórios dos pedidos de pesquisa listar e ordenar construindo relatórios ordenados ou não de acordo com a pesquisa.

No caso da pesquisa ser ordenar, uma chave é ligada e transforma-se a pesquisa em listar. Daí o sistema procede na execução da mesma, depurando o pedido de pesquisa, procurando e armazenando os dados a serem listados, testando-os e finalmente os relatórios. Antes porém de elaborá-los, o sistema verifica a chave de ordenação se esta

IMPRESSÃO

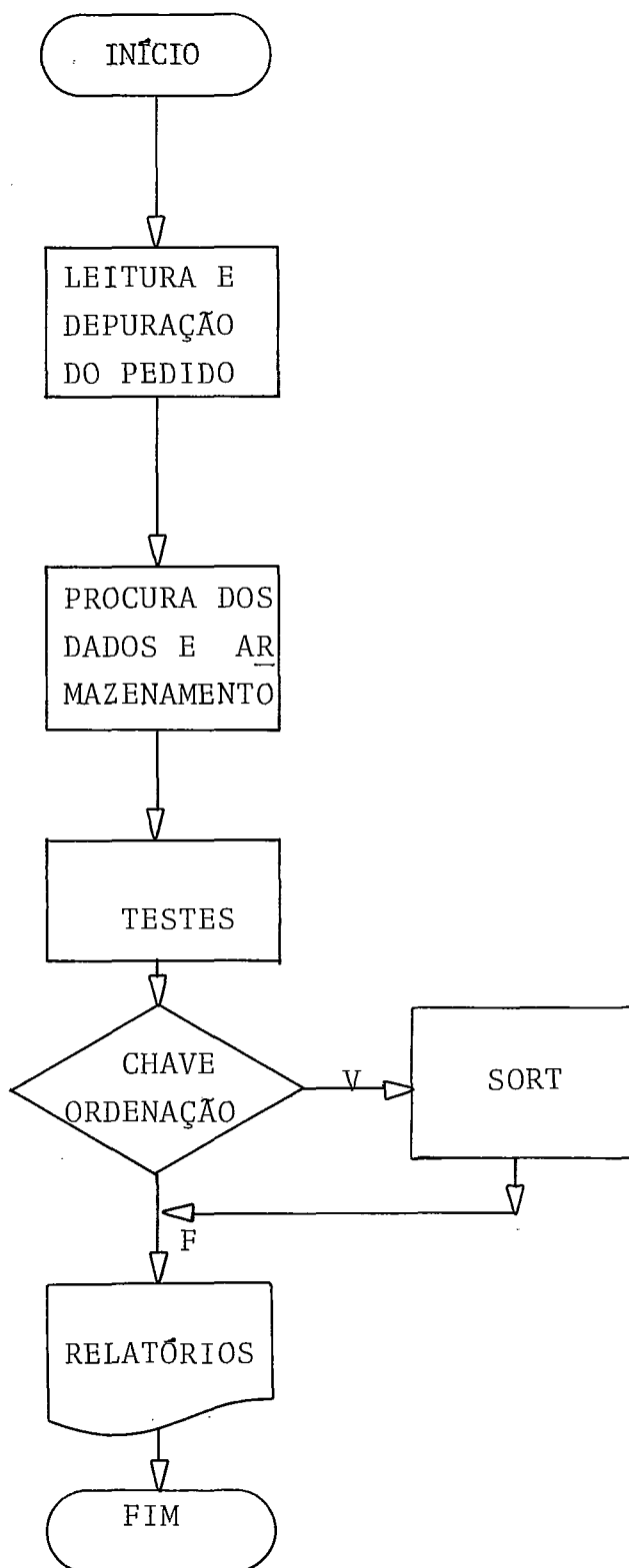


FIGURA 15

ligada . Em caso positivo, o sort é acionado enviando para a impressão um arquivo ordenado. Caso a chave não esteja ligada, o sistema inicia a elaboração dos relatórios com o arquivo na forma em que estiver (Fig. 15).

Pelo modo como está preparada a impressão tem como objetivo, facilitar a sua compreensão. São relatórios simples, práticos, sem muitas linhas e colunas que comumente confundem o usuário. Com estas características, os relatórios foram divididos em 4 tipos diversos podendo cada um deles estar ordenado ou não conforme o pedido de pesquisa.

Os relatórios elaborados são dos tipos:

i. Relatório número 1 (Fig. 16) - é o relatório impresso mais simples e comum. Cada linha corresponde a um registro lógico e cada coluna, os diferentes campos deste registro que tiverem sido pesquisados. As colunas são simetricamente centradas, cada qual com um título referente ao campo do registro.

ii. Relatório número 2 (Fig. 16) - este relatório satisfaz a necessidade de mais uma linha para cada campo do registro. Neste caso cada registro pode ter duas ou mais linhas e as colunas da mesma forma que no relatório número 1, são os campos dos registros que foram pesquisados.

iii. Relatório número 3 (Fig. 17) - este é

NOME	MATRICULA	FUNCAO
SANDRA, DA, SILVA, REIS	046339-6	CHEFE, DE, SERVICO
RITA, DE, CASCIA, SERR, JOGAIB	035519,4	PROFESSOR
LENIR, FERNANDES, DE, FREIRE	049077,6	PROFESSOR
ANGELA, MARA, SILVA	00120577	DIRETOR
MARCO, SOARES	00013698	COORDENADOR
ADERBAL, DA, SILVEIRA	00126483	CHEFF, DE, SERVICO

FIN DE PESQUISA

FIGURA 16.A

NOME	MATRICULA	FUNCAO	QUALIFICACAO	DISCIPLINA, NOMEADO
SANDRA. DA. SILVA. REIS	046339-6	CHEFE. DE. SERVICO	PRIMEIRO. GRAU. COMPLETO SEGUNDO. GRAU. COMPLETO LICENCIATURA. CURTA	
RITA. DE. CASCIA. SERR. JOGAIB	035519.4	PROFESSOR	PRIMEIRO. GRAU. COMPLETO SEGUNDO. GRAU. COMPLETO LICENCIATURA. CURTA	CIENCIAS
LENIR. FERNANDES. DE. FREIRE	049077.6	PROFESSOR	PRIMEIRO. GRAU. COMPLETO SEGUNDO. GRAU. COMPLETO LICENCIATURA. PLENA	MATEMATICA
ANGELA. MARA. SILVA	00120577	DIRETOR	PRIMEIRO. GRAU. COMPLETO SEGUNDO. GRAU. COMPLETO LICENCIATURA. PLENA CURSO. DE. DIRETORIA	PORTUGUES
MARCO. SOARES	00013698	COORDENADOR	PRIMEIRO. GRAU. COMPLETO	INGLES
ADERBAL. DA. SILVEIRA	00126483	CHEFE. DE. SERVICO	PRIMEIRO. GRAU. COMPLETO SEGUNDO. GRAU. COMPLETO LICENCIATURA. CURTA	

FIM DE PESQUISA

DUMP DE ARQUIVO REGISTRO LOGICO NUMERO 1

NOME: SANDRA.DA.SILVA.REIS
 MATRICULA: 046339-6
 FUNCAO: CHEFE.DE.SERVICO
 CATEGORIA: EFETIVO
 TEMPO.DE.SERVICO: 0005
 LOTACAO: SECRETARIA.DE.EDUCACAO
 QUALIFICACAO: PRIMEIRO.GRAU.COMPLETO;SEGUNDO.GRAU.COMPLETO;LICENCIATURA.CURTA
 DISCIPLINA.NOMEADO: -
 DISCIPLINA.OUTRAS: -

DUMP DE ARQUIVO REGISTRO LOGICO NUMERO 2

NOME: RITA.DE.CASCIA.SERR.JOGAIB
 MATRICULA: 035519.4
 FUNCAO: PROFESSOR
 CATEGORIA: EFETIVO
 TEMPO.DE.SERVICO: 0007
 LOTACAO: SECRETARIA.DE.EDUCACAO
 QUALIFICACAO: PRIMEIRO.GRAU.COMPLETO;SEGUNDO.GRAU.COMPLETO;LICENCIATURA.CURTA
 DISCIPLINA.NOMEADO: CIENCIAS
 DISCIPLINA.OUTRAS: CIENCIAS

FIGURA 17.A

NOME	MATRICULA	FUNCAO	CATEGORIA	TEMPO DE SERVICO	LOTACAO
SANDRA DA SILVA REIS	046339-6	CHEFE DE SERVICO	EFETIVO	6005	SECRETARIA DE EDUCACAO
RITA DE CASCIA SERRA JOGAIS	035519.4	PROFESSOR	EFETIVO	6007	SECRETARIA DE EDUCACAO
LENIR FERNANDES DE FREIRE	049077.6	PROFESSOR	EFETIVO	6002	SECRETARIA DE EDUCACAO
ANGELA MARA SILVA	00120577	DIRETOR	SUBSTITUTO	0011	PAULO DE FRONTIM
MARCO SOARES	00013698	COORDENADOR	EFETIVO	6015	FERREIRA VIAUA
ADERBAL DA SILVEIRA	00126483	CHEFE DE SERVICO	CONTRATADO	6005	INSTITUTO DE EDUCACAO

FIGURA 17.B - 1a. PARTE

DISCIPLINA. NOMEADO

QUALIFICACAO

NOME

SANDRA. DA. SILVA. REIS

PRIMEIRO. GRAU. COMPLETO
SEGUNDO. GRAU. COMPLETO
LICENCIATURA. CURTA

RITA. DE. CASCIA. SERR. JOGAIS

PRIMEIRO. GRAU. COMPLETO
SEGUNDO. GRAU. COMPLETO
LICENCIATURA. CURTA

LENIR. FERNANDES. DE. FREIRE

PRIMEIRO. GRAU. COMPLETO
SEGUNDO. GRAU. COMPLETO
LICENCIATURA. PLENA

ANGELA. MARA. SILVA

PRIMEIRO. GRAU. COMPLETO
SEGUNDO. GRAU. COMPLETO
LICENCIATURA. PLENA
CURSO. DE. DIRETORIA

MARCO. SCARES

PRIMEIRO. GRAU. COMPLETO

ADERBAL. DA. SILVEIRA

PRIMEIRO. GRAU. COMPLETO
SEGUNDO. GRAU. COMPLETO
LICENCIATURA. CURTA

FIM DE PESQUISA

CIEENCIAS

MATEMATICA

PORTUGUES

INGLES

o caso da pesquisa especial para "listar tudo", é o "dump de memória", onde todo o arquivo é impresso em relatório. Esta pesquisa é muito comum e até mesmo conveniente, seguindo uma pesquisa de atualização. No caso do dump, o relatório não se faz em linhas e colunas. É mais simples, é feito por registro, isto é, cada registro é totalmente impresso, na ordem em que está arquivado com um campo por linha, ou mais se necessário. Estes registros são numerados. Ao final dessa pesquisa o usuário terá todo o arquivo e com o número de quantos registros.

iv. Relatório número 4 (Fig. 17) - este relatório tem como característica a continuação em outra página. Isto significa, um relatório do tipo número 1 ou 2, mas que nem todas as colunas podem constar na mesma página. Neste caso, o sistema repete sempre a primeira coluna e a seguir aqueles campos que ainda não foram impressos.

Por exemplo, na fig. 17, temos um relatório com 5 campos, dos quais somente tres somam todo o espaço de uma página. Desse modo, a primeira página é elaborada com os tres primeiros e repetindo-se a primeira coluna e mais os campos restantes compõe-se a segunda página. Tal processo deve se repetir quantas vezes for necessário, para a impressão de todos os dados, e sempre repetindo em cada página a primeira coluna, visto ser esta, o ponto de conexão entre os dados impressos. Este tipo de relatório é chamado de paginação múltipla.

3.5. Ordenação

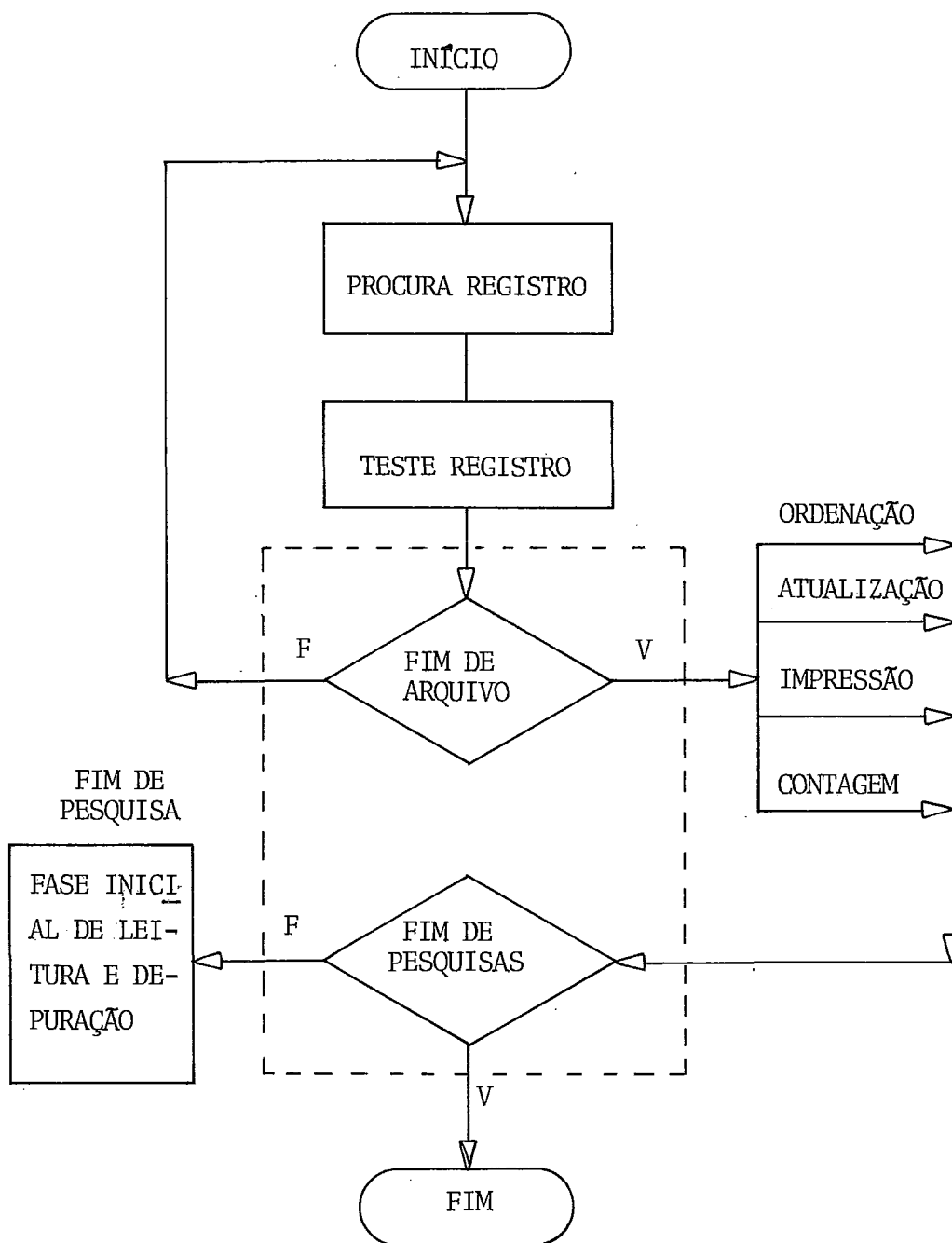
Ordenar é a pesquisa mais completa da Utilização (Fig. 15), dando como resultado um relatório ordenado de todos os dados pesquisados que satisfazem as condições exigidas. É uma pesquisa composta pela pesquisa listar e o sort implantado. Isto significa que quando um pedido de ordenar é enviado, o sistema dá início ao processamento de uma pesquisa de listar os dados desejados, interrompendo-a no momento de elaborar os relatórios. Neste ponto o arquivo a ser impresso é ordenado seguindo-se a elaboração dos relatórios.

Com relação ao tipo de sort, o sistema está preparado a operar com qualquer tipo, devido a portabilidade de de que é caracterizado e mesmo porque cada máquina possui a sua rotina de ordenação. Desse modo o arquivo resultante da pesquisa está preparado para ser ordenado por qualquer tipo de sort implantado ao sistema.

3.6. Fim de Pesquisa

A fase de Fim de Pesquisa tem dois objetivos diretos. Verificar o fim do arquivo de dados e o fim das pesquisas, isto é, se não há mais pesquisas a processar.

O motivo de existência desta etapa no sistema é devido as facilidades permitidas ao usuário no que se

FIM DE PESQUISAFIGURA 18

refere a tamanho do arquivo e número de pesquisas processadas. Uma vez que o arquivo é de tamanho variável e o número de pesquisas ilimitado, fêz-se necessário a presença de uma rotina específica que pesquisasse o fim do arquivo, qualquer que seja seu tamanho e o fim das pesquisas, qualquer que seja o número de pesquisas processadas.

Desse modo "FIM DE PESQUISA" é uma fase de enorme importância no projeto, pois é a unidade de controle do sistema dando os pontos de início e fim de pesquisa.

A sistemática do projeto, lê e testa registro por registro, armazenando ou contando os que satisfazem as condições. Desse modo, o arquivo é totalmente lido e testado para somente depois serem impressos os resultados. Para tanto é necessário saber quando termina o arquivo. Como o tamanho não é considerado nem fixado usa-se uma notação para fim de arquivo. Assim sendo, fim de pesquisa, é a fase que verifica se símbolo de fim de arquivo já foi lido.

Composta de duas rotinas, a primeira testa se o último caracter lido é o símbolo de fim de arquivo. Caso não seja, a rotina devolve o controle a execução que continua a ler e testar mais registros (Fig. 18).

Ao fim do arquivo, isto é, quando o caracter testado pela rotina for o símbolo de fim de arquivo, a rotina envia ao sistema uma mensagem que o fará ciente de que a fase de testes terminou dando início a elaboração dos relatórios, ou ordenação ou mensagem final de contagem, ou atualização segundo a pesquisa iniciada.

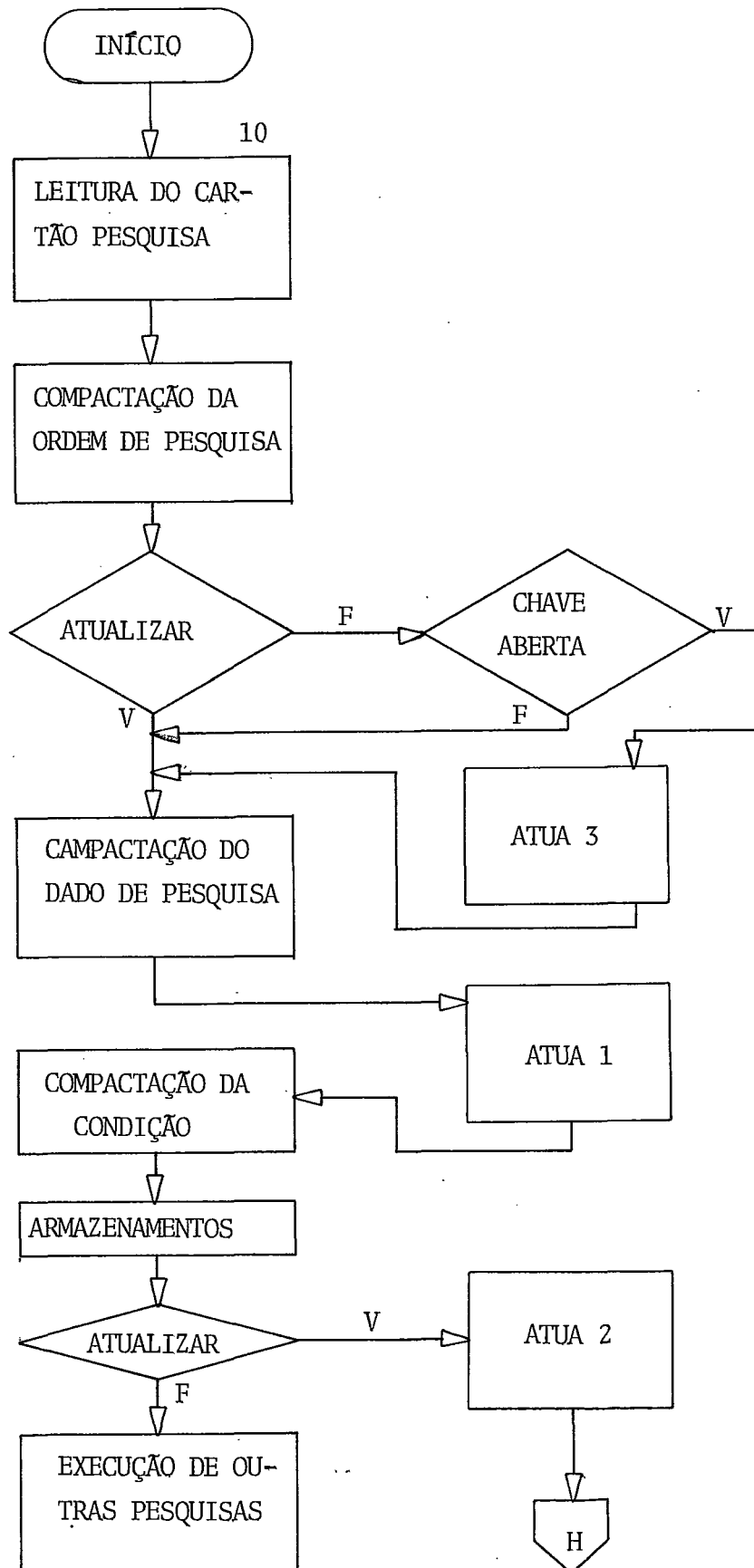
Uma vez completadas e corretas todas estas etapas, o sistema envia a mensagem "FIM DE PESQUISA" e passa o controle para a segunda rotina desta fase, que verifica se há mais pesquisas a serem efetuadas. Neste caso novos testes são processados no intuito de verificar se há mais cartões para leitura. Em caso negativo o processamento termina. Em caso positivo, os bits de testar e listar do index são zerados, juntamente com o contador de erros, e o contador de dados.

O controle é transferido para a fase de leitura e depuração e nova pesquisa começa a ser processada.

C A P Í T U L O I VIV. ATUALIZAÇÃO

É a possibilidade dada ao usuário de modificar os cadastros arquivados sem maiores problemas. Isto significa que o mesmo tipo de operação simples e fácil usado para as pesquisas de listar, contar ou ordenar é usado para a atualização dos cadastros. Através de cartões perfurados pelo próprio usuário, em linguagem corrente e seguindo as regras constantes do manual, os pedidos de atualização podem ser processados pelo sistema.

As pesquisas de atualização consistem em retirar, trocar ou inserir dados nos cadastros elaborados, sob quaisquer condições. Os objetivos dessa atualização, e principalmente pelo modo como é feita, são inúmeros. Em primeiro lu

ATUALIZAÇÃO

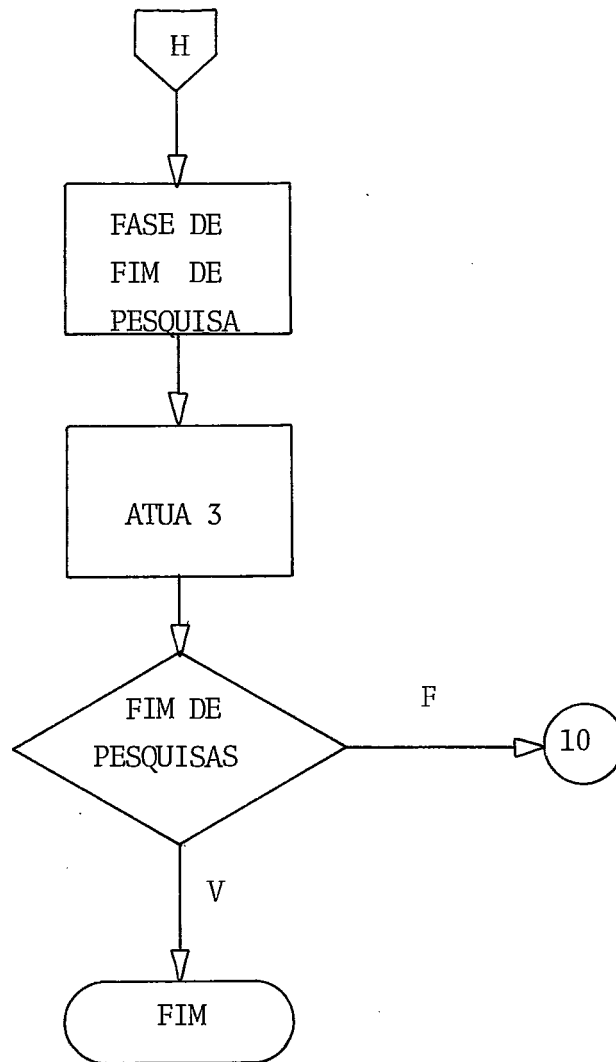


FIGURA 19

gar, não seria viável um sistema de cadastros que somente pesquisasse esses cadastros, sem atualizá-los, por mais complexa e interessante que fosse essa pesquisa. Atualmente quando grupos de pesquisa e estudo aprofundam-se nos mais variados assuntos, não teria a menor aplicação um sistema que manipulasse um cadastro, que dentro de anos, meses, até dias, estaria completamente ultrapassado. Desse modo o sistema de cadastro projetado consta de Utilização e Atualização.

4.1. Características

As características mais importantes são a facilidade de operação por parte do usuário e a confiabilidade, uma vez que o processo usado é bem simples e eficiente.

Este processo consiste em duas fitas de modo que a cada instante existam dois estados do cadastro arquivado, o atual e o anterior. Tal fato diminui grandemente o risco de perda de arquivo, outra vantagem real da atualização. Há também a possibilidade, de usar fitas novas em lugar das anteriores, construindo-se assim, uma biblioteca dos diferentes estágios em que esteve o cadastro.

O método de atualização permite várias pesquisas simultaneas. Isto é, vários pedidos consecutivos, consistem numa só pesquisa de atualização. Neste caso os pedidos são lidos e depurados um por um, e os dados referentes à pesquisa vão formando um arquivo auxiliar. Quando todos os pedi-

dos tiverem sido lidos, o arquivo auxiliar estará formado. A leitura do caracter de fim de pesquisa, ou um pedido diferente de atualizar, aciona o processo de atualização. A partir daí, e de posse do arquivo auxiliar, o cadastro atual é gerado, dando por encerrada a pesquisa de atualização.

A possibilidade de várias atualizações ao mesmo tempo, evitando-se com que o arquivo seja lido muitas vezes, acarreta um problema de máquina. Dependendo da quantidade de pedidos de pesquisa, o arquivo auxiliar, pode criar problema de espaço de memória, para o caso de computadores que não possuam memória virtual. Para estes casos, os pedidos de pesquisas devem ser divididos em grupos. Isto é, um grupo de 500 pedidos, por exemplo, que estourasse o espaço de memória, deveria ser lido em grupos de 250 pedidos. Desse modo, as 500 atualizações seriam feitas em duas pesquisas e o arquivo seria lido duas vezes apenas.

Este processo, do arquivo auxiliar, foi idealizado para evitar que 500 pedidos, por exemplo, significasse 500 pesquisas e 500 leituras no arquivo, o que acarretaria um aumento enorme de tempo de execução e risco de perda de arquivo, ao ser lido tantas vezes.

4.2. Sequência de Operações

Apesar de ser um ítem a parte, não é totalmente independente. A atualização opera acoplada às fases da Utilização (Fig. 19).

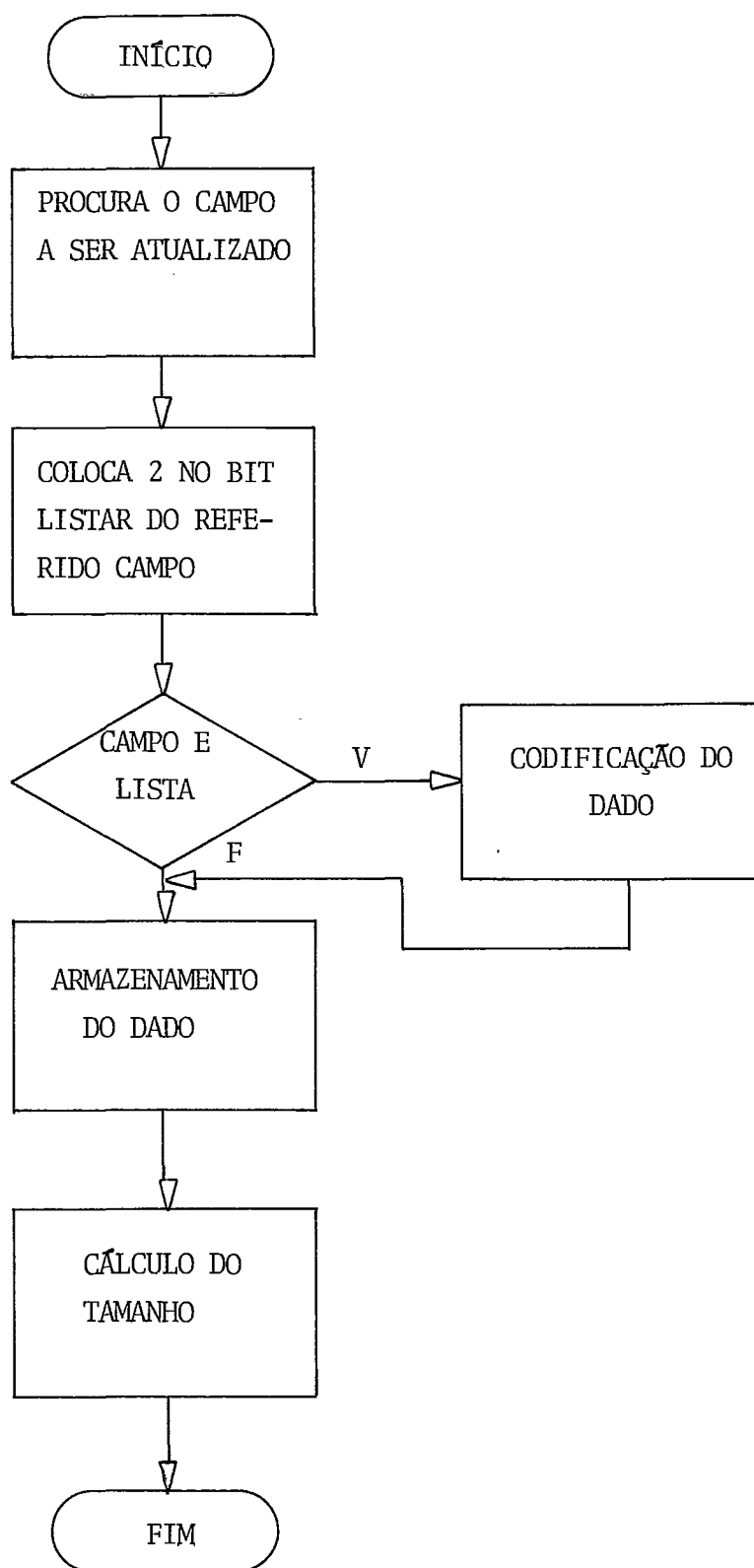
Composta de três etapas principais, essa pesquisa inicia sua FASE 1 ao ser chamada pela etapa de leitura

ra e depuração. Neste ponto a ATUA 1, que é a primeira etapa, faz a preparação do campo e do dado a ser atualizado e devolve o controle a etapa de leitura e depuração. A partir daí, o processamento é o mesmo da Utilização, onde a condição é compactada, os dados e operandos armazenados. Após cada leitura e depuração de um pedido, a FASE 2 tem início, formando o arquivo auxiliar. Esse arquivo é elaborado na forma de um vetor armazenando os dados sequencialmente. Desse modo são formados os campos de condição, dados atuais e campos atuais, para cada pedido de pesquisa.

Ao terminar a ATUA 2, que é a FASE 2, novo pedido é lido e depurado e a ATUA 2 é mais uma vez chamada. Este processo se repete até que todo o grupo de pesquisa esteja preparado no arquivo auxiliar. Neste ponto então, há duas ramificações diferentes do sistema. Se o caracter fim de pesquisa for lido, o comando é dado a fase de fim de pesquisa, na qual a ATUA 3 (FASE 3) é chamada, montando o arquivo atual, baseado no arquivo auxiliar, e devolve o comando para a fase de fim de pesquisa, que dá por encerrada a pesquisa. Ao contrário disto, o caracter fim de pesquisas não é lido mas outro pedido diferente de atualizar, então a própria fase de leitura e depuração chama a ATUA 3, criando o cadastro atual e a seguir executa a nova pesquisa lida.

4.3. Fase ATUA 1

É a fase de preparação da pesquisa de atualizar dados. O processo usado, utiliza o bit listar (Fig.6) ce

FASE ATUA 1FIGURA 20

mo auxiliar.

As funções da fase ATUA 1 são:

- (i) testar qual o campo que deve ser atualizado
- (ii) colocar o código "2" no bit listar
- (iii) codificar ou não, conforme a necesidade, o dado atual
- (iv) armazenamento de dado atual
- (v) cálculo do tamanho deste dado.

Todas estas operações são executadas pela ATUA 1 após ser chamada pela fase de leitura e depuração , tendo como dados iniciais o index do cadastro e as compactações da ordem de pesquisa, do campo e do dado atual (Fig.19).

A procura do campo a ser pesquisada é sequencial, comparando todos os campos do index com o campo compactado do cartão pesquisa.

Na hipótese deste campo não ser encontrado, isto significa que o campo a ser pesquisado não confere com nenhum daqueles existentes no index do cadastro, assim sendo, o sistema envia as devidas mensagens de erro e a pesquisa é interrompida.

No caso do campo ser encontrado, o processamento continua, desta feita, colocando o código "2" do

referido campo.

A codificação do dado atual, quando necessária, é feita por chamadas a rotinas específicas. Essa necessidade é definida pelas informações relativas ao dado, contidas no index do cadastro (Fig. 20).

O armazenamento e o cálculo do tamanho do dado atual são operações simples que determinam o fim da etapa de preparação à pesquisa de atualização.

Evidentemente, essas duas últimas operações, no caso do dado ser codificado, são feitas com relação ao da do atual já codificado.

4.4. Fase ATUA 2

É a fase de montagem do arquivo auxiliar . A Atua 2 recebe os dados atuais codificados ou não, na ordem em que aparecem no index do arquivo, com os respectivos tamanhos. Recebe também os vetores oper (contendo o operador da condição), IND (contendo o índice do bit testar), o cond 1 (contendo o índice do primeiro caracter da condição), o cond 2 (contendo a condição) e o index com o código 2 no bit listar do campo a ser atualizado. De posse de todos estes dados a Atua 2 monta um vetor, onde cada registro possui três campos, o das condições, dos dados atuais e dos campos a serem atualizados. Cada registro representa um pedido de pesquisa . Todos os dados são armazenados sequencialmente separados pelo

FASE ATUA2

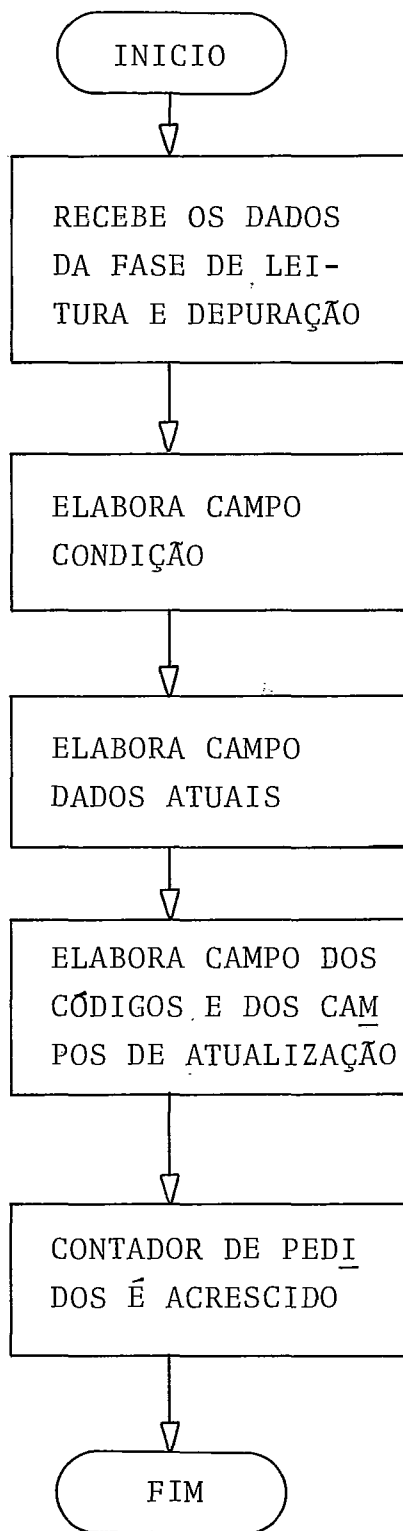


FIGURA 21

símbolo (\$) e o símbolo de fim do vetor auxiliar são duas barras (//). Os registros e seus campos possuem tamanho variável, o que representa economia de memória, pois serão ocupadas o número exato de posições necessárias, sem desperdício de espaço útil. Além do tamanho requerido para cada registro serão ocupadas duas posições a mais, no início do mesmo, que identificam o tamanho do registro (Fig. 22).

Assim sendo cada pedido de pesquisa terá um registro equivalente no vetor auxiliar composto da seguinte maneira:

Nas duas primeiras posições, o tamanho do registro. Sequencialmente os dados da condição que é um campo alfanumérico. Separado por um símbolo \$, segue-se o campo dos dados atuais, codificados ou não. Também é um campo alfanumérico e pode conter vários dados para atualização. O último campo é numérico, composto dos números dos campos a serem atualizados, na ordem em que se apresentarem no index. Podem ser vários, e o último número significativo deste campo representa o código da ordem de pesquisa (Fig. 22).

O campo que contém vários dados, estes estarão separados por barras:

A cada chamada da Atua 2, um registro é montado, o contador de pedidos é acrescido e uma chave de atualização é aberta. A Atua 2 é chamada pelo programa principal do sistema, sempre que a leitura e depuração de um pedido de pesquisa tiver terminada e se verificar que tal pedido é para atualizar.

O objetivo da chave de atualização é para

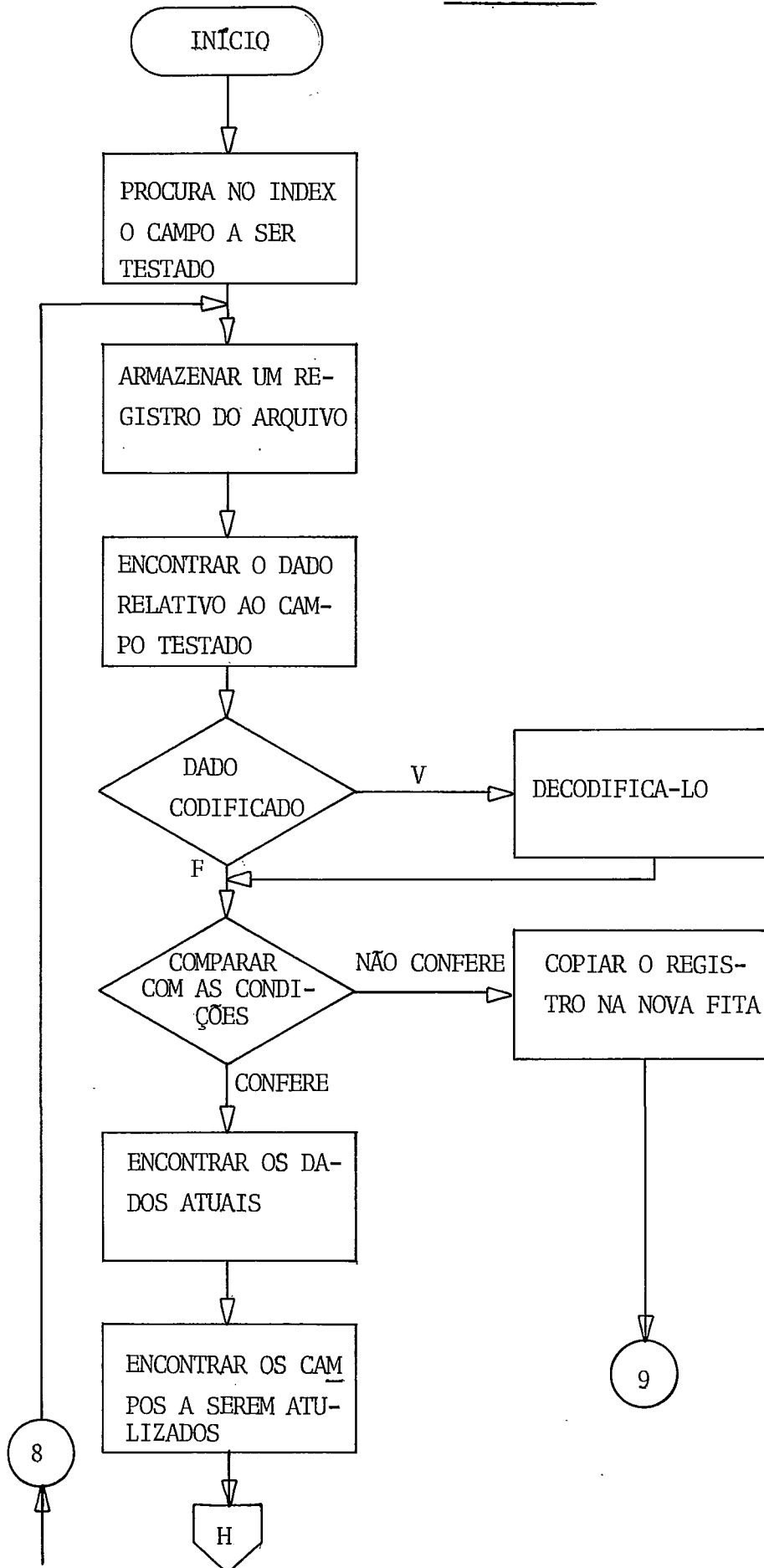
que o sistema ao ler um pedido diferente de atualizar, tenha o conhecimento de que existem pesquisas de atualizações que ainda não foram executadas e que estão preparadas no arquivo auxiliar. Desse modo, o sistema interrompe a execução da pesquisa lida, passa para a Atua 3, ao fim da qual recomeça a pesquisa interrompida.

Desse modo a divisão de grupos de pedidos de atualização pode ser feita através de pedidos diferentes entre estes grupos.

4.5. Fase ATUA 3

É a fase da atualização propriamente dita. A Atua 3 recebe o arquivo auxiliar baseado no qual elabora o novo cadastro (Fig. 23). O método consiste em armazenar um registro de cada vez do arquivo, compará-lo com todos os campos condição. Caso não confira, então, copiá-lo na nova fita. Caso confira, procurar no campo do registro, quais os campos a serem atualizados, ler os dados atuais do vetor auxiliar, atualizar o registro e então copiá-lo na nova fita. Este processo se repete até terminar o arquivo. Neste ponto todas as atualizações lidas terão sido processadas, o contador de pedidos de pesquisa é inicializado e a chave de atualização é fechada.

A Fase Atua 3 possui dois pontos de chamada:

FASE ATUA 3

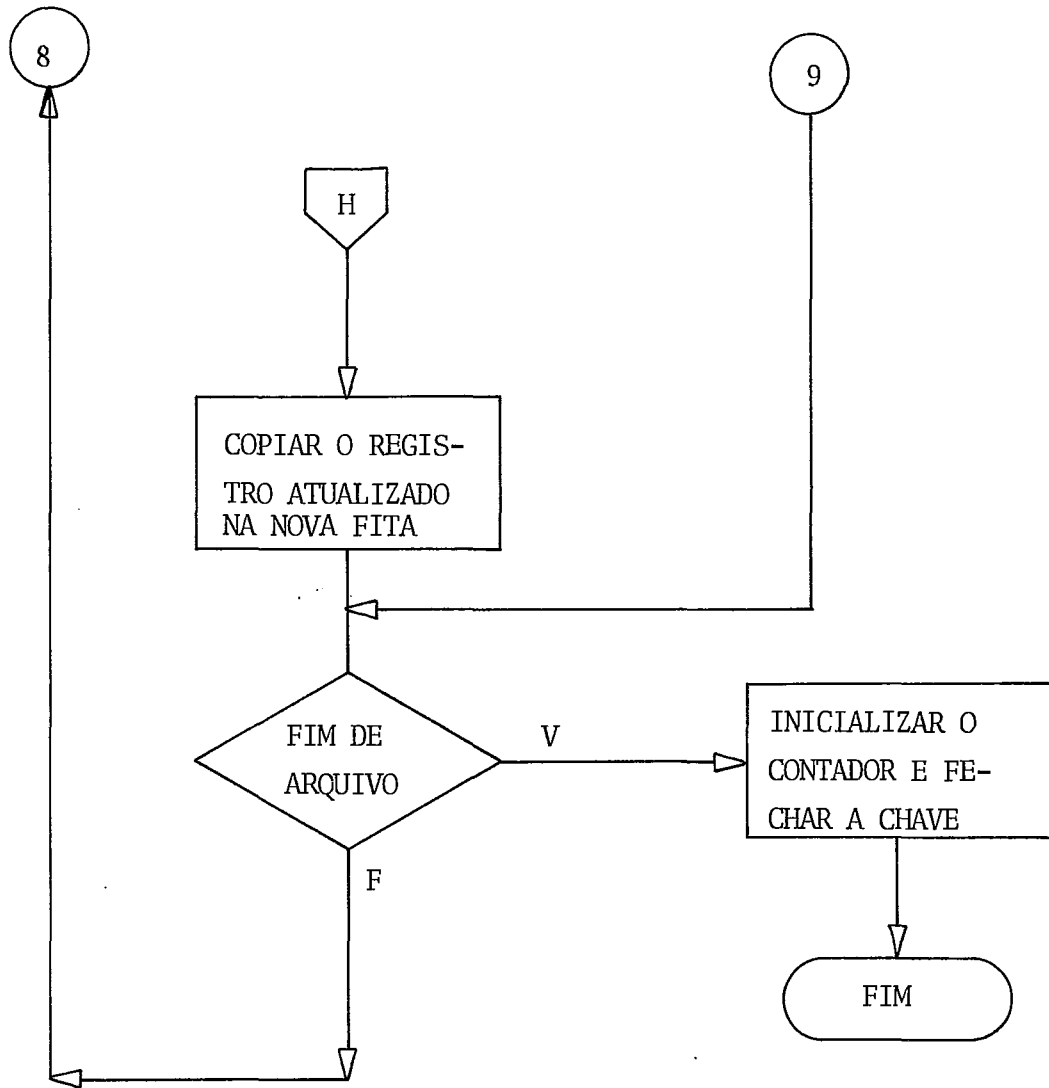


FIGURA 23

(i) Caso de haver outras pesquisas seguindo as atualizações - nesta hipótese, os pedidos são lidos e o arquivo auxiliar vai sendo formado até terminar os pedidos de atualização. Um novo pedido diferente de atualizar ao ser lido em seguida, levaria o sistema para outra pesquisa, sem contudo haver executado todas as atualizações preparadas. Desse modo , e neste caso, antes de prosseguir com a nova pesquisa, a chave de atualização é testada e uma vez que a pesquisa ainda não foi executada, ela estará aberta. Assim sendo a Atua 3 é chamada. Isto se verifica no programa principal do sistema após ter sido executada a fase de leitura e depuração.

(ii) Caso em que não há mais pesquisas após as atualizações - nesta hipótese, após o arquivo auxiliar estar elaborado, o caracter de fim de pesquisas é lido. Isto significa uma chamada a fase de fim de pesquisas, que por sua vez, antes de dar por encerrada a pesquisa, verifica a chave de atualização. Esta por conseguinte, estará aberta, posto que a pesquisa ainda não foi executada. Assim sendo a fase de fim de pesquisa chama a ATUA3, ao fim da qual, dá a pesquisa por en-cerrada.

4.6. Tipos de Atualização

A pesquisa de atualização permite a alteração no cadastro de três maneiras diferentes: Pode-se inserir, retirar ou trocar dados do cadastro, contudo cada uma dessas

operações possui características e restrições próprias.

4.6.1. No caso da inserção, só é permitido ao sistema inserir dados no cadastro existente. Isto é, aumentar o cadastro com mais registro, compete ao sistema de elaboração dos cadastros e não ao sistema de Utilização e Atualização. Entretanto, devido a forma modular deste sistema, é possível um acoplamento com fases de elaboração de cadastros.

Assim sendo, ao sistema é só permitido inserir dados do tipo lista inclusiva, isto é, um campo que possua mais de uma informação codificada. Neste caso, pode-se inserir mais informações.

Na figura 4, tem-se um exemplo de index e arquivo codificado. Note o campo do index "QUALIFICAÇÃO", é um campo do tipo lista inclusiva pois permite uma série de informações. A lista referente a este campo, também está no index e contém todas as hipóteses possíveis.

Observe na segunda linha do arquivo codificado. A funcionária de nome Sandra possui no campo "Qualificação" os dados "12000100020003" - este código desmembrado significa: 12 → é o tamanho do campo codificado. 0001 → a informação é o primeiro dado da lista (primeiro grau completo). 0002 → a informação é o segundo dado da lista (segundo grau completo). 0003 → a informação é o terceiro dado da lista (licenciatura curta).

Suponhamos, agora, que o dito funcionário fizesse mais cursos e o cadastro fosse atualizado. Se os cur -

os fossem licenciatura plena e mestrado, os dados inseridos seriam 0004 e 0006. Evidentemente o tamanho seria corrigido, e o campo codificado atual seria "20 00010002000300040006" . Assim sendo a inserção só é feita em campos com lista inclusiva e para dados pertencentes a esta lista.

4.6.2. A pesquisa retirar não oferece maiores problemas. Após encontrar o dado a ser retirado, é feita uma pequena análise com relação apenas ao tamanho do dado. Se o campo possui tamanho fixo, este permanece, e a retirada do dado é feita apenas zerando as posições em que se encontra a informação . Caso o campo seja de tamanho variável, então o dado é retirado e em seu lugar ficam quatro posições zeradas, que é o tamanho mínimo de uma informação. Por exemplo

(i) o campo codificado de tamanho fixo - "072657431" - ao ser retirado passa - "070000000" -

(ii) o campo codificado de tamanho variável - "0823744942" - passa para - "040000" -

4.6.3. A troca é a operação mais usada e pode ser processada para qualquer tipo de dado, isto é, dado codificado ou não, ou qualquer tipo de lista, inclusiva ou exclusiva. O problema maior que surge na pesquisa de troca de informações é com relação ao tamanho e a existência do tipo de dado. No

caso de uma lista, por exemplo, só se pode trocar o dado por outro existente na lista, uma vez que este será codificado e quem determina o código é a lista. Com relação ao tamanho temos:

(i) campos de Tamanho Fixo - só permite a troca com dados do mesmo tamanho ou menores que serão completados por zeros. Dados maiores, determinam mensagens de erro, e a pesquisa é interrompida.

(ii) campos de Tamanho Variável - este campo permite a troca com dados de qualquer tamanho. Exemplo:

1. campo de tamanho fixo

- "076302783" trocado para "072853753"
ou para "070003214".

2. campo de tamanho variável

"09MARIA.REI" - trocado para "15MARIA.REI.
SILVA" ou para "05MARIA" ou para "09MARIA.
SSA".

4.7. Observações

A atualização é uma pesquisa ampla, que permite grande manuseio do cadastro. Contudo é uma execução sem relatório, isto é, ao ser solicitada uma atualização, es

ta é processada e a única linha impressa, é "FIM DE PESQUISA". Evidentemente, isto acontece, no caso de não haver erros, pois neste caso as mensagens apropriadas são impressas. O sistema não se responsabiliza pelo sistema operacional, e como o arquivo de entrada nunca é alterado, a retomada depois de um erro de sistema ou de máquina envolve na pior das hipóteses, uma repetição da operação.

Outra observação importante é que a atualização é feita um registro de cada vez. Não existe a possibilidade de se atualizar mais de um registro na mesma pesquisa. Vários registros para atualização, significam vários pedidos de pesquisa, que se forem consecutivos poderão ser processados ao mesmo tempo, contudo cada pedido de pesquisa está relacionado a apenas um registro.

O que se pode fazer é atualizar vários campos no mesmo registro, isto é totalmente aceitável pelo sistema, contudo convém notar que cada registro só pode ser pesquisado uma vez em cada grupo de atualizações.

CONCLUSÃO

Conforme anteriormente citamos, o sistema aqui descrito, visa solucionar os problemas atuais de empresas de pequeno porte, com relação a custo e mão-de-obra especializada, oferecendo-lhes um sistema de utilização e atualização de cadastros, qualquer que seja seu conteúdo, podendo ser operado por usuários leigos em computação e implantado em vários tipos de máquina.

A linguagem na qual foi implementado é o FORTRAN básico, tendo sido testado nas máquinas BURROUGHS 6700, IBM/360 e MITRA-15. Evidentemente, que no caso de uma implementação, em qualquer que seja a máquina, exigirá algumas adaptações no sistema, principalmente no que diz respeito aos dispositivos de E/S.

Para minimizar o tempo de E/S técnicas de compressão de dados (Ver tese PIETRACCI) poderiam ser usadas.

Várias adaptações podem ser feitas, inclusive rotinas de ordenação, para as quais o sistema está preparado, com arquivos já pesquisados, prontos para ordenação.

Com tais características, este projeto pretende responder as necessidades sempre crescentes, de usuários leigos em computação.

BIBLIOGRAFIA

1. INPUT - OUTPUT SUBSYSTEM - INFORMATION MANUAL - BURROUGHS
2. KNUTH - The Art of Computer Programming
3. FORTRAN - BURROUGHS
4. FORTRAN - IBM
5. FORTRAN - MITRA 15
6. PIETRACCI - Compressão de Dados (Tese)

LISTA DE ERROS

<u>NÚMEROS</u>	<u>MENSAGENS</u>
1001.....	Ordem de pesquisa errada
1002.....	Dado para pesquisa não existente
1003.....	Caracter inválido depois do nome de um dado
1004.....	Erro do sistema (E.S)-o campo da pergunta não é LI, LE ou ¸ ¸
1005.....	Falta caracter delimitador (;)-(,)ou(?)
1006.....	Falta caracter (:)
1007.....	E.S -código de campo errado, não é pergunta nem condição
1008.....	Número de condições excede ao limite máximo de 10 condições
1009.....	Execução interrompida - pesquisa com erros
1010.....	E.S - o índice da condição não confere com o índice do campo testado
1011.....	Informação arquivada não confere com o que se pede
1012.....	E.S - o campo não é alfa nem numérico
1013.....	E.S - para a rotina cod, o campo não é LI nem LE
1014.....	E.S - o sistema detetou um erro que não existe, provavelmente número errado

- 1015..... Dado para troca ou inserção maior que o espaço reservado pelo índice do cadastro
- 1016..... Dado alfanumérico para troca ou inserção menor que o espaço fixo reservado pelo índice do cadastro
- 1017..... Erro de montagem do cadastro - caracter de multiplicidade não confere com os códigos existentes.
- 1018..... Dado para troca ou inserção não existente.

MANUAL DO USUÁRIO

O usuário envia ao sistema um pedido de pesquisa na forma de um ou mais cartões perfurados. Várias pesquisas podem ser enviadas consecutivamente e após a última tem-se um flag de fim de pesquisas, caracterizado pelo símbolo " ? ", podendo ser perfurado em qualquer coluna do mesmo cartão que contiver a última pesquisa ou de um novo cartão (Fig. 18). Não há limite para o número de cartões, nem para o número de pesquisas. Há limite somente para o número de condições, cada pesquisa deve ter no máximo 10 condições. O sistema recebe uma série de pesquisas que são lidas e testadas uma por uma. No caso de algum erro, o sistema envia mensagens específicas de erro segundo lista anexa. A fase de execução só é iniciada se não houver nenhum erro no pedido de pesquisa.

O sistema permite dois tipos de pesquisas:

Utilização e Atualização

a. A utilização do arquivo é feita através de três pesquisas diversas, sob quaisquer condições:

- i. listagem de dados
- ii. contagem de dados
- iii. ordenação de dados

3. CONDIÇÃO - possui o seguinte formato:

(TIPO DE DADO) (OPERADOR) (DADO REAL)

3.1. TIPO DO DADO - como no tipo de dado a ser pesquisado este ítem também depende do cadastro, e para o cadastro de funcionários pode ser:

NOME, MATRICULA, IDADE, etc...

3.2. OPERADOR - pode ser =
<
>

3.3. DADO REAL - é o dado propriamente dito para o cadastro - exemplo pode ser: MARIA
158745
15
DIRETOR, etc...

Havendo mais de uma condição, devem estar separadas por vírgulas (Fig. 9).

Exemplos:

LISTAR NOME, MATRICULA : FUNÇÃO = DIRETOR ;

CONTAR NOME : IDADE > 30, FUNÇÃO = PROFESSOR ;

ORDENAR MATRICULA : FUNÇÃO = SERVENTE, TEMPO DE SERVIÇO > 15 ;

Há também a hipótese de "DUMP DE MEMÓRIA", quando o usuário de seja uma listagem completa do arquivo e o caso de uma pesquisa sem condição. Para esses casos especiais os pedidos teem o seguinte formato:

- i. LISTAR TUDO; - para o dump
- ii. (PESQUISA DESEJADA):SEMPRE; - para a pesquisa sem condição: exemplo:
-LISTAR NOME, MATRICULA:SEMPRE;

b. Atualização

- i. Formato:

ATUALX (DADO ATUAL) : (CONDIÇÃO);

- 1. X - pode ser a letra - T - para trocar dados
 - I - para inserir dados
 - R - retirar dados

2. DADO ATUAL - é o dado que se deseja atualizar, possui a forma: (TIPO DE DADO) = (DADO REAL)

exemplos: NOME = MARIA RITA
 IDADE = 25
 FUNÇÃO = PROFESSOR, etc...

3. CONDIÇÃO - possui o mesmo formato que o dado atual:

(TIPO DE DADO) = (DADO REAL)

exemplo: FUNÇÃO = PROFESSOR
 TELEFONE = 2616534
 LOTAÇÃO = FERREIRA VIANA

Exemplos de pedidos de atualização:

ATUALR TELEFONE = 2588306 : NOME = MARIA ;

Isto significa retirar o número 2588306 de telefone do registro cujo nome é Maria.

ATUALT FUNÇÃO = DIRETOR : NOME = MARCOS;

Significa trocar a função para diretor quando o nome for Marcos.

Pesquisa Especial:

ATUALR : NOME = DENISE ;

Esta pesquisa significa retirar todo o registro cujo nome é Denise.

ii. Restrições

1. Uma vez que a condição para a pesquisa de atualização, tem a função de encontrar o registro a ser atualizado, não há necessidade, de mais de uma condição. Por este motivo o sistema aceita a pesquisa somente com uma condição.

2. Para o caso de uma sequência de atualizações, só serão pro -

cessadas aquela que tiverem o mesmo tipo de condição, isto é, o campo que identifica o registro deve ser o mesmo para todas as atualizações (Fig. 22).

3. Para o caso de vários campos no mesmo registro, estes campos devem estar na ordem em que se apresentam no index do arquivo.

iii. Observações:

1. Para inserir dados - Veja - Tese - Estudo e Realização de um Sistema Portátil de Cadastros orientado para Usuário Leigo em Computação - 1ª parte -
2. Para dividir grupos de atualização, basta intercalar uma outra pesquisa que não seja atualizar.
3. Posto que cada registro só pode ser chamado uma vez em cada grupo de pesquisas, o caso de se desejar pesquisas diferentes no mesmo registro, significa pedidos de pesquisas diferentes para o mesmo registro e que teriam de ser processados em grupos separados de pedidos.

Em alguns casos, poderiam pesquisas diferentes transformarem-se numa mesma pesquisa. Note o exemplo Retirar um dado e trocar outro. São duas pesquisas diversas contudo, poderia-se em lugar de retirar um dado, trocar para 0000 que é um dado não significativo. Desse modo as duas pesquisas poderiam ser feitas num mesmo pedido (VEJA 4.6.2.)

Exemplo prático: ATUALR TELEFONE = 2583411:

NOME = MARIA ; ATUALT END = RUA QUINZE 18 : NOME = MARIA ;

Poderia ser:

ATUALT END = RUA QUINZE 18, TELEFONE = 00000000 : NOME = MARIA;

ROTINAS QUE COMPÕEM O SISTEMA

1. Function libt (M) - testa se o campo é lista inclusiva , exclusiva ou não é lista.
2. Function Ibit (car) - testa se a ordem é listar (1) , contar (2), ordenar (3), atuali (6), atualR (5), atualt (4).
3. Function Lar (J) - transforma o número J em caracter
4. Icop (M,I,car,J,perg) - rotina que compacta as perguntas.
5. Tescan - prepara o índice do arquivo para a execução, liga os bits listar ou testar, prepara e armazena as condições, operadores, etc...
6. Erro (N) - imprime a mensagem de erro
7. Function Num (J) - transforma o caracter J num número
8. Cod - decodifica os dados
9. Nark - apanha o dado do arquivo
10. Arma - depura o cartão pesquisa, testa qual a ordem, faz as compactações, prepara o index.
11. Exec 1 - calcula o índice da condição, armazena o dado a

ser testado.

12. Exec 2 - testa os dados não codificados.
13. Exec 3 - decodifica e testa os dados.
14. Exec 4 - fase de fim de pesquisas, verifica se há mais pesquisas, zera os bits listar e testar.
15. Function Lend (J) - testa se é fim do arquivo de dados.
16. Exec 5 - procura do campo a ser testado, procura e testes dos dados, início dos registros lógicos.
17. Aread - faz as leituras e inicialização da pesquisa.
18. Lefita - lê o arquivo em vetores de N caracteres por vez.
19. Prima - faz as impressões simples.
20. Prima 2 - dump de memória.
21. Prima 3 - faz a impressão de listas inclusivas.
22. Prima 4 - imprime os dados da lista inclusive, é chamada pelo Prima 3.
23. Prima 5 - listagem múltipla.
24. Decod - codifica os dados.

25. Li - calcula os espaços e margens do relatório impresso.
26. Carac - transforma o número X em um vetor de caracteres.
27. Atua 1 - prepara os dados para a atualização.
28. Atua 2 - elabora os arquivos auxiliares.
29. Atua 3 - elabora o cadastro atual.