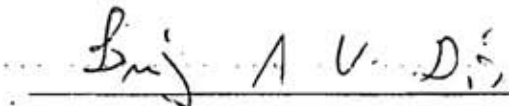


1. Classificação <i>INPE-COM.1/TPT</i> <i>CDU: 681.3.06:025.5</i>		2. Período	4. Critério de Distribuição: interna ☐ externa ☒
3. Palavras Chaves (selecionadas pelo autor) <i>Disseminação Seletiva da Informação</i> <i>Computação</i>			
5. Relatório nº <i>INPE-1169-TPT/075</i>	6. Data <i>Dezembro, 1977</i>	7. Revisado por <i>Carlos H. Boto Gois</i>	
8. Título e Sub-Título <i>UM SISTEMA DE DISSEMINAÇÃO</i> <i>SELETIVA DE INFORMAÇÃO</i>		9. Autorizado por <i>Nelson de Jesus Parada</i> <i>Diretor</i>	
10. Setor <i>DSE</i>	Código	11. Nº de cópias <i>16</i>	
12. Autoria <i>Ioris Ramalho Cortez</i> <i>José Iguelmar Miranda</i>		14. Nº de páginas <i>119</i>	
13. Assinatura Responsável <i>Ioris Ramalho Cortez</i>		15. Preço	
16. Sumário/Notas <i>Um sistema de Disseminação Seletiva de Informação visa oferecer aos pesquisadores, das mais diversas áreas, um acompanhamento atualizado do que se vem publicando nas suas áreas de pesquisa, evitando informações irrelevantes. Este tipo de serviço tem se proliferado em todo o mundo, nas bibliotecas especializadas. A idéia inicial foi de H.P. Luhn, em 1959, e logo ganhou corpo, porque visava o uso de computadores, na época também emergido, e isto solidificou o desejo de um Centro de Documentação, que é servir a um número maior possível de leitores. A idéia fundamental de Luhn, e as modificações já feitas ao longo do tempo, incorporam-se aqui novas pesquisas no sentido de tornar o sistema adaptativo, uma idéia nova em relação a todos os sistemas de DSI, e o uso de uma nova função de disseminação coloca o nosso sistema na vanguarda dos sistemas ora existentes.</i>			
17. Observações <i>Tese de Mestrado em Computação Aplicada, aprovada em 22 de novembro de 1977.</i>			



Aprovada pela Banca Examinadora
em cumprimento dos requisitos exigidos
para a obtenção do Título de Mestre em
Computação Aplicada

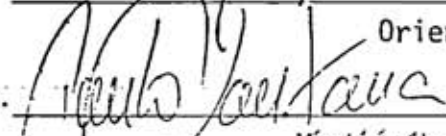
Dr. Luiz Alberto Vieira Dias


Presidente da Banca

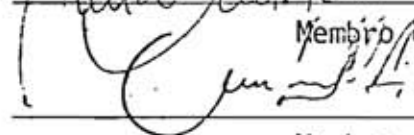
Eng. Carlos Henrique Bôto Gois, MSc.


Orientador

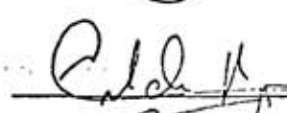
Eng. Paulo Henrique de Assis Santana, MSc.


Membro da Banca

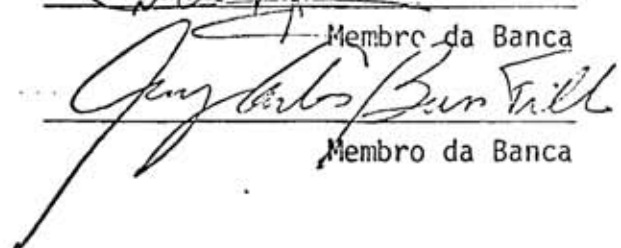
Eng. Newton Marcos Vasconcellos, MSc.


Membro da Banca

Dr. Celso de Renna e Souza


Membro da Banca

Eng. Arry Carlos Buss Filho, MSc.


Membro da Banca

José Iguelmar Miranda


Candidato

Iôris Ramalho Cortez


Candidato

São José dos Campos, 22 de novembro de 1977

A Nossos Pais
e Esposas

ÍNDICE

	Pág.
ABSTRACT	v
LISTA DE FIGURAS	vi
CAPÍTULO I - INTRODUÇÃO	1
1.1 - DISSEMINAÇÃO SELETIVA DE INFORMAÇÃO	1
1.2 - TENDÊNCIAS DOS SISTEMAS DE DSI	3
1.3 - SISTEMA DE DSI PARA O INPE	9
1.4 - PROJETO DO SISTEMA	10
1.4.1 - DEFINIÇÃO DO SISTEMA	10
1.4.2 - MODULARIDADE	11
1.5 - ENTRADA/SAÍDA	13
1.5.1 - ENTRADA	13
1.5.2 - SAÍDA	16
1.6 - RESUMO	17
CAPÍTULO II - DESCRIÇÃO DO SISTEMA	19
2.1 - MÓDULO GERAÇÃO	19
2.1.1 - MODELO DE INDEXAÇÃO	19
2.2 - MÓDULO CADASTRA/ATUALIZA	22
2.3 - MÓDULO REFINA	22
2.4 - MÓDULO BUSCA	24
2.5 - MÓDULO IMPRIME	26
CAPÍTULO III - ESTRUTURAS DOS ARQUIVOS	29
3.1 - ORGANIZAÇÃO DO PROCESSAMENTO	29
3.2 - ORGANIZAÇÃO DOS ARQUIVOS	29
3.3 - ARQUIVOS SEQUENCIAIS	30

	Pág.
3.4 - O ARQUIVO DE DESCRITORES	32
3.5 - O ARQUIVO USUCOD	37
3.6 - CONTROLES	41
3.7 - RESUMO	42
CAPÍTULO IV - FUNÇÃO DE DISSEMINAÇÃO	43
4.1 - INTRODUÇÃO	43
4.2 - ESTRATÉGIAS DE BUSCA	43
4.3 - FUNÇÃO DE DISSEMINAÇÃO	47
4.4 - FUNÇÃO DE INCREMENTO-DECREMENTO	51
4.5 - IMPLEMENTAÇÃO DA FUNÇÃO INCREMENTO-DECREMENTO	53
4.6 - FUNÇÃO DE BAIXO-INCREMENTO	55
CAPÍTULO V - CONCLUSÃO	57
AGRADECIMENTOS	59
BIBLIOGRAFIA	61
APÊNDICE A - NOTIFICAÇÃO DO SISTEMA AO USUÁRIO	A.1
APÊNDICE B - PROGRAMA DO MÓDULO GERAÇÃO	B.1
APÊNDICE C - PROGRAMA DO MÓDULO CADASTRA/ATUALIZA	C.1
APÊNDICE D - PROGRAMA DO MÓDULO REFINA	D.1
APÊNDICE E - PROGRAMA DO MÓDULO BUSCA	E.1
APÊNDICE F - TABELAS DE DIV	F.1
APÊNDICE G - VERIFICAÇÃO DA FUNÇÃO INCREMENTO-DECREMENTO E BAIXO INCREMENTO	G.1

ABSTRACT

The aim of a Seletive Dissemination Information System is to offer to distinct areas researchers an up to date accompaniment of that which is being published on their area, trying to avoid irrelevant information. This such kind of service is being proliferated on specialized libraries all over the world. The initial idea, from H.P. Luhn, has grown rapidly, since it aimed the utilization of computers, also emerging on that period. This has consolidated the Documentation Center purpose, that of being of use to as many readers as possible. To Luhn's fundamental ideas and to the modifications which have been effected throughout the years, we have incorporated new researchs, which aim to convert the system into an adaptive one, a new idea in relation to all SDI systems. This idea and the utilization of a new dissemination function places our system on the forefront of those which exist today.

LISTA DE FIGURAS

	Pág.
FIG. I.1 - FLUXO DE INFORMAÇÕES NO SISTEMA DE DSI	4
FIG. I.2 - NOTIFICAÇÃO AO USUÁRIO	5
FIG. I.3 - NOTIFICAÇÃO DO CNEN	7
FIG. I.4 - NOTIFICAÇÃO DO PASCAL	8
FIG. I.5 - GRÁFICO DE COMUNICAÇÃO DOS MÓDULOS	12
FIG. I.6 - FOLHA DE CADASTRO	14
FIG. I.7 - DESCRITORES	15
FIG. I.8 - REGISTRO LÓGICO DO ARQUIVO FONTE	16
FIG. II.1 - REGISTRO DO ARQUIVO FONTE	20
FIG. II.2 - FLUXOGRAMA DO MÓDULO GERAÇÃO	21
FIG. II.3 - REGISTRO DO ARQBIB	22
FIG. II.4 - REGISTRO DO DESIDEN	22
FIG. II.5 - MÓDULO CADASTRA/ATUALIZA	23
FIG. II.6 - FLUXO DO MÓDULO REFINA	25
FIG. II.7 - MÓDULO BUSCA	27
FIG. II.8 - FLUXO DO MÓDULO IMPRIME	28
FIG. III.1 - FORMATO DO REGISTRO DE RECUPERADO	31
FIG. III.2 - REGISTRO DO USUÁRIOS	31
FIG. III.3 - RELAÇÃO ENTRE DESCRITORES E USUCOD	33
FIG. III.4 - REGISTRO LÓGICO DO ARQUIVO PERFIL	34
FIG. III.5 - NÓ TERMINAL	36
FIG. III.6 - NÓ NÃO-TERMINAL	36
FIG. III.7 - EXEMPLO DA ESTRUTURA DO ARQUIVO DESCRITORES	38
FIG. III.8 - DESENHO DO NÓ TERMINAL	39
FIG. III.9 - DESENHO DO NÓ NÃO-TERMINAL	39

	Pág.
FIG. III.10 - ARQUIVO USUCOD E DETALHE DO SEU REGISTRO	40
FIG. IV.1 - LIMAR DE RELEVÂNCIA	50
FIG. IV.2 - GRÁFICO DA FUNÇÃO TRANSPONSTA	52
FIG. IV.3 - GRÁFICO DA FUNÇÃO INCREMENTO/DECREMENTO COMO FINALMEN TE USADA	54

CAPÍTULO I

INTRODUÇÃO

1.1 - DISSEMINAÇÃO SELETIVA DE INFORMAÇÃO

Uma disseminação efetiva de informações sempre foi o alvo principal de um Centro de Documentação. Assim é que sua preocupação é desenvolver técnicas que permitam levar ao leitor o que há de mais recente no seu campo de pesquisa.

Uma primeira solução adotada, ainda hoje em uso, é a de fornecer à comunidade leitora fac-símeles das publicações recentes reunidos numa só brochura. Este serviço denomina-se sumários ("current awareness") e existindo eventual interesse por alguma publicação, a pessoa recorre ao Centro.

O próximo passo, foi no sentido de fazer chegar ao usuário, de maneira individual, estas publicações. Isto era feito, inicialmente, de maneira manual (Susan, 1973). Sobre este passo podem destacar-se os seguintes dois fatos importantes:

- Primeiro, o número de documentos possíveis de "cadastrar" era pequeno, enquanto o crescimento da quantidade de informação era rápido.
- Segundo, eram poucas as pessoas atendidas - verdadeiros privilegiados na população de usuários - não havendo atendimento à comunidade em geral.

Justifica-se, então, buscar outros meios de solução do problema. De uma análise mais detalhada dos dois fatos citados anteriormente, surgiram duas constatações:

- DINAMIZAÇÃO - A alta taxa de geração de informação era o principal problema no sistema de comunicação. A disseminação seletiva

manual não podia nem acompanhar aquela taxa nem ter acesso à maior parte das informações. Fazia-se necessário um meio que pudesse registrar e disseminar, seletivamente, um universo muito maior que o até então processado, e acompanhar-lhe o crescimento, mantendo o serviço sempre num bom nível. Em outras palavras, era necessário uma dinamização do processo;

- OBJETIVIDADE - O emprego de sumários, como foi dito, visa atender a um grupo de leitores. Assim, recebidos as publicações das diversas áreas, envia-se aos leitores fac-símeles dos seus índices, sem nenhuma pré-seleção. Com isto, especialistas em Meteorologia, por exemplo, folheiam informações da área de Computação, Eletrônica etc., que normalmente não lhes são relevantes. Seria bem mais interessante tais pessoas receberem informações apenas do seu campo de pesquisa. Nota-se, então, uma falta de objetividade no material disseminado.

Hans Peter Luhn (1958), numa tentativa para superar os problemas citados, introduziu um novo elemento no processo, elemento este que foi o computador. Além de ser um pioneiro na aplicação de computadores a este serviço, foi ele também o primeiro a usar o nome DSI e a defini-lo do seguinte modo:

"DSI (Disseminação Seletiva de Informações) é um serviço que, dentro de uma organização, canaliza novos itens de informação, vindos de qualquer fonte, para aqueles pontos onde é alta a probabilidade de serem úteis em conexão com interesse ou trabalhos atuais."

Num outro trabalho (Luhn, 1961), mostrou que estava com uma visão do problema, bastante objetiva, lançando duas idéias básicas, classificando-as como:

ENTRADA DOS DOCUMENTOS

Entenda-se, pela locução, a entrada ("INPUT") do sistema. Os documentos a serem disseminados eram colocados em arquivos legíveis

por computador, geralmente fitas magnéticas. Dava-se então, nesta fase, a separação dos documentos e dos seus descritores (descritores são palavras retiradas do título ou do resumo da publicação, caracterizando-a). Este conjunto de descritores, chamado perfil do documento, passava então a identificá-lo;

DISSEMINAÇÃO SELETIVA DE NOVAS INFORMAÇÕES

Assim como o conjunto de descritores do documento, chamado perfil, o identificava, também os usuários eram identificados no sistema por um conjunto de descritores, chamado perfil do usuário, definindo a área de interesse do mesmo.

Fazia-se a comparação dos perfis documento/usuário e de acordo com o grau de similaridade entre eles, aquele era selecionado e, o usuário notificado.

Na Figura I.1 tem-se uma visão do fluxo de informações no sistema e, na Figura I.2, uma notificação ao usuário como desenvolvida por Luhn (1961).

1.2 - TENDÊNCIAS DOS SISTEMAS DE DSI

Após esta primeira implantação feita por Luhn, seguiram-se outras desenvolvidas pela IBM, na forma de sistemas estanques. No ano de 1966, houve um crescimento bem significativo no número de sistemas de DSI. O fato, na época, foi devido aos resultados altamente positivos que os bons serviços de informação prestaram ao governo e indústria. Entretanto, fatores como custo e problemas práticos inibiram um maior alastramento de tais sistemas. Uma outra observação que pode ser feita, refere-se à não uniformidade do formato dos dados de entrada. Cada sistema geralmente preparava os seus próprios arquivos e cada fornecedor de arquivos de informação tinha também, geralmente, seu formato próprio. Isto levou a diferentes mecanismos de operação dentro dos sistemas de DSI, tornando necessário o estudo de cada um deles em separado, para se conseguir um perfeito entendimento.

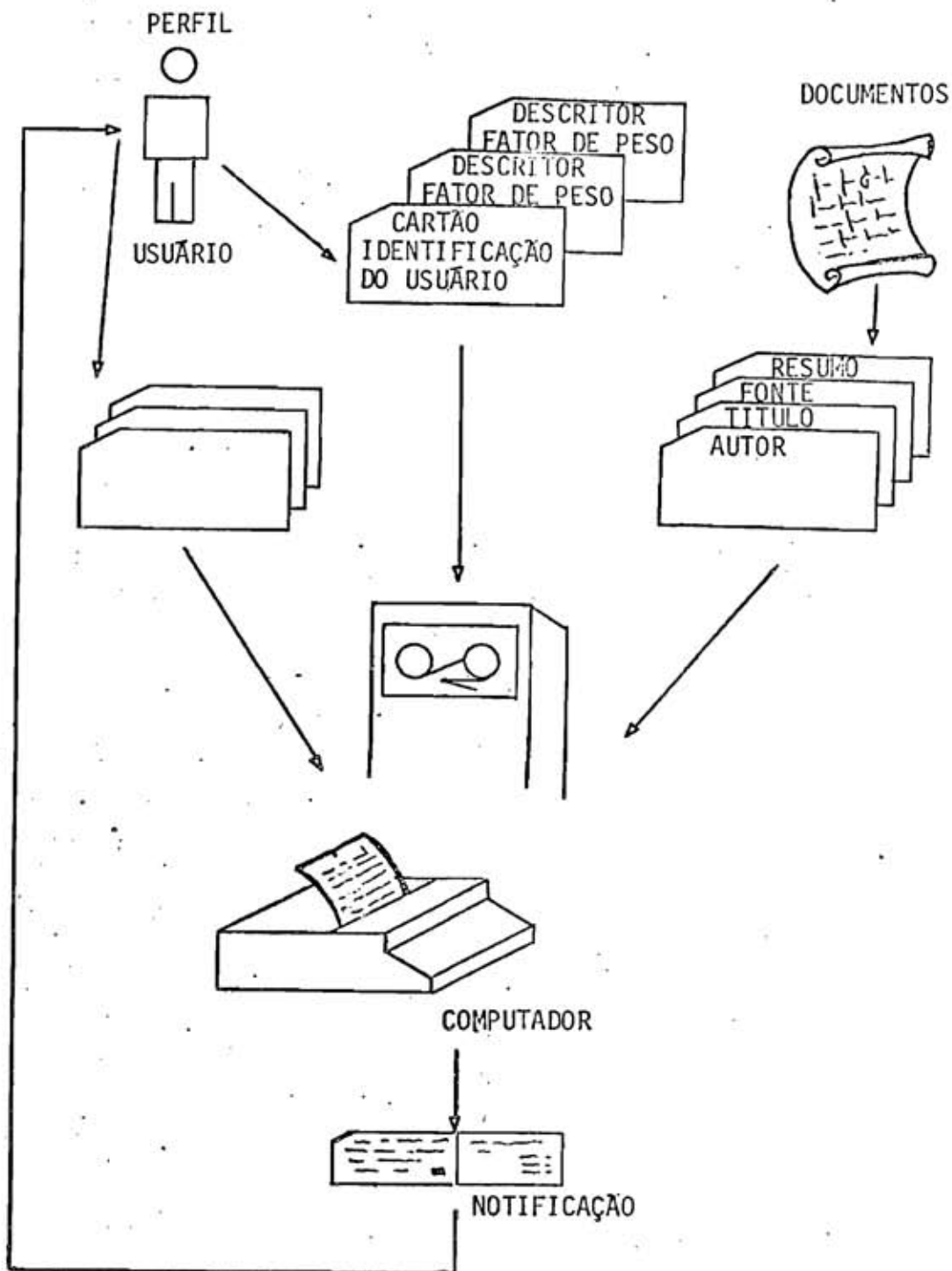


Fig. I.1 - Fluxo de Informações no Sistema de DSI

FRENTE

When responding, return this stub

SUB SERVICE
IBM RESEARCH CENTER
YORKTOWN HEIGHTS, N.Y.

In behalf of:
IBM ASDD YORKTWN

TO: C B HENSLEY
DEPT 496 BLDG 801

DATE 10/1/59

FROM: C B HENSLEY
DEPT 496 BLDG 801

Comments: [] [] [] []

If copy of document is wanted, check here: ☐ document requested

This information is sent to you in behalf of:
IBM ASDD YORKTWN

TO: C B HENSLEY
DEPT 496 BLDG 801

DATE 10/1/59

VERSO

Keyword-In-Context Index for Technical Literature (KWIC Index)

Luhn, H. P. Aug. 1959
IBM ASDD RC 127 16 pages 00018

ABSTRACT. A distinction is made between bibliographical indexes for new and past literature based on the willingness of the user to trade perfection for currency. Indexes giving keywords in their context are proposed as suitable for disseminating new information. These can be entirely machine-generated and hence kept up-to-date with the current literature. A compatible coding scheme to identify the indexed documents is also proposed. In it elements are automatically extracted from the usual identifiers of the document so that the coded identifier yields a maximum of information while remaining susceptible to normal methods of ordering.

Keyword-In-Context Index for Technical Literature (KWIC Index)

Luhn, H. P. Aug. 1959
IBM ASDD RC 127 00018

keyword	context	index
technical	literature	bibliography
disseminate	information	machine
generate	code	identify
document	order	retrieval
derive	readable	record
intellectual	complete	update
publication	journal	communication
print	text	concordance
title	list	service
notation	name	letter
word	punch	tape
card	dictionary	sort

Fig. I.2 - Notificação ao Usuário

Apesar do rápido crescimento dos anos 60, o total de sistemas de DSI até agora é relativamente pequeno. Dammers (1972) nos indica que apenas uma pequena percentagem ($\pm 2\%$) dos profissionais em ciência e tecnologia no mundo Ocidental usam serviços computarizados, para fazer disseminação seletiva de informação.

Os principais fatores apontados como responsáveis por esta pequena percentagem são:

1. A relutância natural em se aceitar novas técnicas;
2. A não propagação destes sistemas pelos centros de informações que os utilizavam;
3. O alto custo de coleta de grande quantidade de informação;
4. A variedade de campos de interesse de muitos cientistas, tornando necessária a busca regular em muitos arquivos de informação.

Na literatura afim, os arquivos com informação são chamados "Literatura-em-fitas magnéticas".

Felizmente, podemos dizer que, apesar dos problemas acima, todos os centros que ora têm tais sistemas em operação, estão satisfeitos com o desempenho dos mesmos, pois se consegue um altamente satisfatório atendimento aos usuários.

Em estatísticas levantadas por estes centros, observa-se que o índice de atendimento, quanto a material relevante, está numa faixa considerada boa quando em torno de 60% (Gluchowicz, 1973). Pode-se dizer que o sistema de DSI computarizado foi uma solução altamente satisfatória ao problema proposto.

Como exemplos de sistemas atualmente em operação, Figuras I.3 e I.4 mostram notificações ao usuários, feitas por dois deles. Trata-se do DSI do CNEN e do PASCAL, sistema desenvolvido na França.

058592 Rackley, R.I. Environment of Wyoming tertiary uranium deposits. Amer. Ass. Petrol. Geol. Bull. (1971). v. 56(4) p. 755-774.

B3 19

LUIZ A DE ALMEIDA DEM

Prezado pesquisador , para melhor selecao , enviamos neste cartao uma lista dos numeros das publicacoes recebidas acima. Assinale entre os asteriscos o seu interesse pelas publicacoes relativas a esses numeros , de acordo com o codigo abaixo. **** Devolva este cartao sem demora. ****

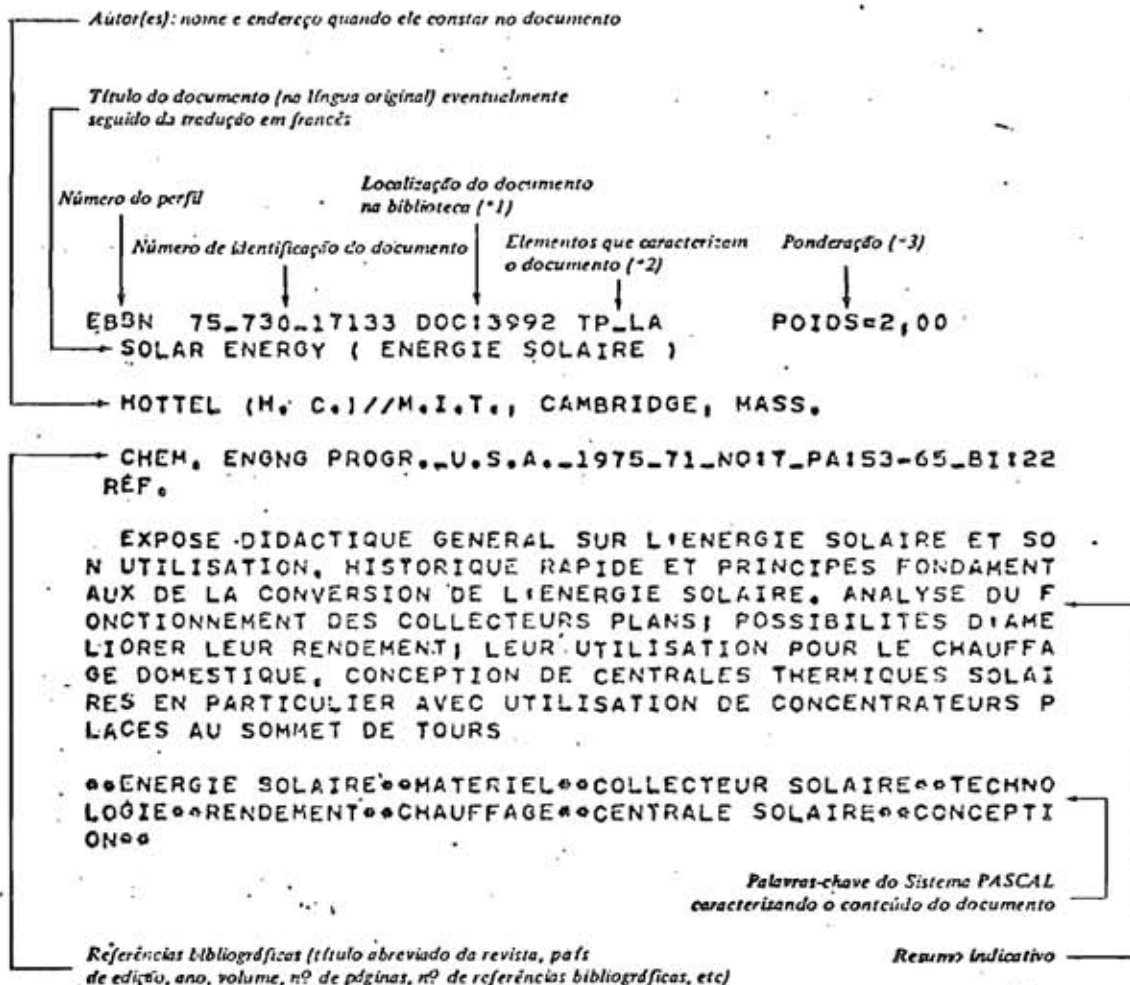
M Muito

Interesse *R* Razcavel

N Nenhum

0058592 * *

DESCRIÇÃO DE UMA FICHA DE REFERÊNCIA BIBLIOGRÁFICA



(*1) Se o documento não existe na biblioteca, lê-se o seguinte:

NA = documento não acessível

NC = documento não acessível ao CNRS mas disponível no lugar cuja sigla figura no fim das referências bibliográficas

(*2) Estes elementos podem ser os seguintes:

TP: o documento é um periódico

TL: o documento é um livro

TT: o documento é uma tese

TB: o documento é uma patente

TS: outros tipos de documentos

Estas informações podem ser completadas pelas seguintes:

SC: o documento é um congresso (num periódico por exemplo se o tipo é P)

ST: o documento é uma tese (num periódico por exemplo se o tipo é P)

SV: se o documento é uma tradução

SM: o documento é um mapa

DU: documento assinalado em várias seções do "Bulletin Signalétique"

(*3) A ponderação permite classificar as fichas seleccionadas em ordem decrescente de pertinências em relação à consulta.

Fig. I.4 - Notificação do PASCAL

1.3 - SISTEMA DE DSI PARA O INPE

O Instituto de Pesquisas Espaciais, atua em áreas de pesquisa e desenvolvimento as mais diversas, tais como, Meteorologia, Sensoriamento Remoto, Computação, Combustão, Eletrônica etc., gerando e recebendo enormes quantidades de informação, ficando o processo de armazenamento e recuperação desta informação a cargo da sua Divisão de Banco de Dados. Além disto, visando uma detenção ainda maior de informações pertinentes aos diversos projetos desta Instituição, a Divisão de Banco de Dados entrou em contato com o National Technical Information Service (NTIS) - USA, no sentido de adquirir, quinzenalmente, fitas magnéticas, contendo em média 2500 itens de informação sobre periódicos, revistas, artigos etc., publicados recentemente, nos quais o governo americano tem, de algum modo, participação. Esta seria a primeira etapa, seguindo-se posteriormente novos contatos com entidades tais como Chemical Abstracts Condensates - CAC, Scientific and Technical Aerospace Reports - STAR, Computerized Engineering INDEX-COMPENDEX etc.

Assim, a quantidade de informações chegando àquela divisão crescerá enormemente, e um processo efetivo de disseminação destas informações, mais do que interessante, é imperativo. Em consequência, um sistema automático de disseminação seletiva de informação foi desenvolvido e está sendo implantado e, neste trabalho, é apresentada a sua descrição.

Indica-se a seguir, o que foi introduzido neste DSI, suas entradas e saídas, bem como os módulos que o compõem. Nos outros capítulos tem-se:

- . Capítulo II: Descrição do Sistema
- . Capítulo III: Estrutura dos Arquivos
- . Capítulo IV: Função de Disseminação
- . Capítulo V: Conclusões

Fundamentalmente, este trabalho inspirou-se na idéia central estabelecida por Luhn (1958). As modificações feitas são baseadas em experiências de laboratório, onde se obtiveram resultados altamente positivos (Sage, 1965), sendo fundamentalmente as seguintes:

- LÓGICA DE BUSCA - As lógicas atualmente em uso são: percentagem, peso, booleana e peso-booleana. Usa-se aqui a de probabilidade. To das elas serão detalhadas no Capítulo IV.
- REFINAMENTO DOS PERFIS DOS USUÁRIOS - Aqui utilizar-se-á um modelo adaptativo de refinamento dos perfis dos usuários, uma das características que destaca o sistema do conjunto já existente.

1.4 - PROJETO DO SISTEMA

Apresenta-se a seguir, o projeto lógico do sistema, etapa por etapa, para facilitar o seu acompanhamento.

1.4.1 - DEFINIÇÃO DO SISTEMA

O sistema de DSI, a serviço do Centro de Documentação e Informação, tem por objetivo servir a uma população de leitores com o que de mais recente se publica em suas respectivas áreas de pesquisa, de uma maneira individual e a mais precisa possível, no sentido que oferece a cada indivíduo apenas material relevante para ele.

OBJETIVOS

Absorção do volume crescente de informações, gravadas nas fitas NTIS, ou chegadas de qualquer outra fonte de informação;

Fornecimento aos leitores, de informações mais precisas (mais relevantes), atualizadas e adequadas, de maneira a melhorar a qualidade da decisão acerca do material recebido;

Alívio da carga de trabalho do pessoal ativo na disseminação de informações por meio de sumários, podendo a biblioteca usar esse pessoal em outros serviços;

Atendimento a um número maior de usuários, com rapidez e muito mais informação.

1.4.2 - MODULARIDADE

Esta é uma técnica pela qual o sistema é dividido em seções lógicas independentes. Permite um desenvolvimento e manutenção eficiente das partes do sistema. Será dado antes, um resumo conjunto de conceitos e fundamentos para um melhor entendimento do que se seguirá. Tem-se então:

Módulo, um conjunto não vazio de programas que realiza uma função lógica definida, possuindo um e só um ponto de saída e um e só um ponto de entrada. Função lógica é o nome dado a qualquer algoritmo. Função do módulo é a transformação que ocorre quando este é chamado, i.e., o processo efetuado sobre os parâmetros de entrada. Pontos de entrada/saída são os pontos pelos quais se dá a passagem de parâmetros ou do que flui de um para outro módulo.

Equivalência de Módulos é a propriedade que dois módulos têm quando, submetendo-os a mesmos conjuntos específicos de entrada são obtidos os mesmos conjuntos de saída. Outro ponto importante a citar é o que se refere à independência dos módulos. Deve-se, entretanto, observar que, às vezes, esta independência deixa de existir em situações como, por exemplo: se um módulo M1 realiza uma função f e um módulo M2, uma função g que depende de f , i.e., $g = h(f)$, M2 passa a depender de M1.

Posto isto, o sistema se divide nas seguintes funções lógicas:

- . Manipulação dos arquivos-fonte
- . Cadastramento de usuários
- . Refinamento dos perfis
- . Manutenção dos arquivos existentes

- . Processamento da lógica de busca
- . Relatórios e notificações

Tais funções devem ser relacionadas de maneira a determinar como será o fluxo de informação. Este interrelacionamento está explicitado através do Gráfico de Comunicação dos Módulos (Figura 1.5), no qual

- O bloco DSI não realiza nenhum processamento. Pode ser enquadrado como bloco de controle, segundo a linguagem de Maynard (1976).
- O módulo de um nível mais alto (numericamente) pode ser processado a partir do momento que tiver recebido seus parâmetros de um nível mais baixo.
- O processamento paralelo, de dois dos três primeiros módulos, permite uma otimização do tempo de execução do sistema.

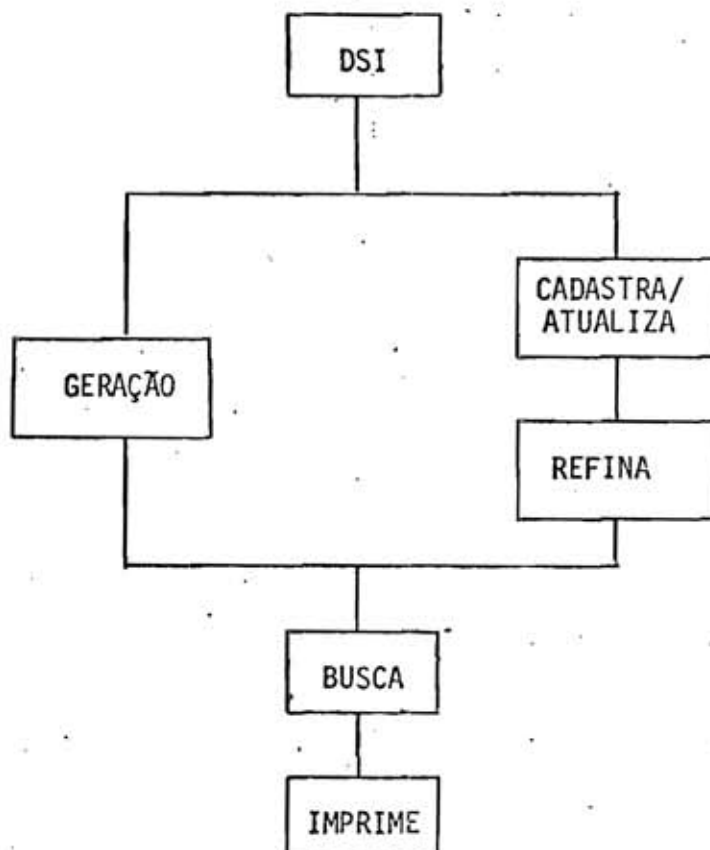


Fig. 1.5 - Gráfico de Comunicação dos Módulos

Algumas das vantagens obtidas com este tipo de abordagem são:

1. A lógica do sistema fica relacionada à lógica do problema;
2. Problemas complexos podem ser divididos em partes relativamente simples;
3. Cada módulo pode ser projetado, codificado e testado "independentemente" de outros módulos;
4. A programação do sistema torna-se mais acessível, simplificando a manutenção, que é feita por módulo;
5. Fica grandemente facilitado o uso de técnicas de superposição ("OVERLAY").

1.5 - ENTRADA/SAÍDA

Descrição resumida de cada uma destas fases.

1.5.1 - ENTRADA

A entrada consiste de duas fontes de dados, a saber:

- i. Dados do usuário - está é a parte que inclui cadastramento e atualização. Na Figura I.6 tem-se a imagem do que se necessita para montar o cadastro e, quando necessário, a atualizar os dados do leitor. Os campos se auto definem.

Na Figura I.7 tem-se o formulário de definição dos descritores a serem usados pelo leitor na definição do seu perfil. O campo de valor é um número real, que representa a probabilidade por ele associada ao descritor, variando entre 0 e 1.

- ii. Documentação a ser disseminada - trata-se do arquivo fonte de documentos, contendo as informações que serão disseminadas.

Atualmente, estes arquivos são as fitas NTIS reformatadas para o padrão do INPE; futuramente ter-se-á a biblioteca produzindo

DADOS PARA CADASTRO															
NÚMERO DO PERFIL <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; text-align: center;">1</td> <td style="width: 10%; border: 1px solid black; height: 15px;"></td> <td style="width: 10%; border: 1px solid black; height: 15px;"></td> <td style="width: 10%; border: 1px solid black; height: 15px;"></td> <td style="width: 10%; border: 1px solid black; height: 15px;"></td> <td style="width: 10%; border: 1px solid black; height: 15px;"></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	1						1	2					NOTA DO CPD: DUPLICAR AS COL. 2 A 6 EM TODOS OS CARTÕES	INSERÇÃO (I) 1 ALTERAÇÃO (A) 1 REMOÇÃO (R) 1 7	
1															
1	2														
DADOS PESSOAIS															
NOME COMPLETO															
ENDEREÇO DA INSTITUIÇÃO <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border: 1px solid black; padding: 2px;">INSTITUIÇÃO (NOME)</td> <td style="width: 50%; border: 1px solid black; padding: 2px;">RUA (NOME, NÚMERO)</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">CIDADE</td> <td style="border: 1px solid black; padding: 2px;">ESTADO CEP</td> </tr> </table>				INSTITUIÇÃO (NOME)	RUA (NOME, NÚMERO)	CIDADE	ESTADO CEP								
INSTITUIÇÃO (NOME)	RUA (NOME, NÚMERO)														
CIDADE	ESTADO CEP														
RESERVADO PARA O SETOR DE CADASTRO															
NOME 9															
INSTITUIÇÃO 39															
RUA 60															
CARTÃO 1															
DUPLICAR DO CARTÃO 1 2 1 7 NÚMERO 8 CIDADE 14 ESTADO 34 CEP 36 USUARIO INTERNO (I) 1 41 EXTERNO (E) 1 42 DATA CADASTRAMENTO															
CARTÃO 2															

Fig. I.6 - Folha de Cadastro

DESCRITORES

[illegible]

(*) INSERÇÃO (1)

ALTERAÇÃO NO VALOR (A)

RETIRADA (R)

* FORMATO DE ATRIBUIÇÃO :

●	X	X	X	X	X
---	---	---	---	---	---

Fig. 1.7 - Descritores

também seus próprios arquivos, além dos provenientes de outras fontes externas tais como NASA, Chemical Abstract etc.

Na Figura I.8, tem-se a representação em blocos de um registro lógico de fitas NTIS, usada como base para o desenvolvimento deste trabalho.

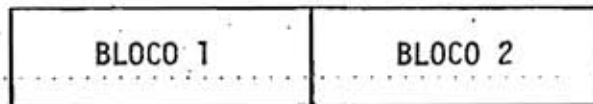


Fig. I.8 - Registro Lógico do Arquivo Fonte

No Bloco 1 tem-se o Diretório contendo especificação do tamanho do registro e posição, com o respectivo tamanho de cada campo que se encontra no Bloco 2.

No Bloco 2 tem-se as especificações da publicação: nome do(s) autor(es), título, ano de publicação, descritores, resumo, número de identificação etc.

Estas partes definem, então, a entrada do sistema.

1.5.2 - SAÍDA

Consiste basicamente das notificações aos indivíduos cadastrados, colocados em fichas de pequeno tamanho, permitindo um fácil manuseio pelo usuário, além de formar um arquivo próprio, contendo informações suficientes sobre o documento selecionado.

No Apêndice A tem-se uma mostra de como a notificação é apresentada.

Vamos dar agora uma breve explicação dos módulos apresentados na Figura I.5.

O Módulo Cadastra/Atualiza tem por função fazer a manutenção dos arquivos existentes e cadastrar usuários, deixando sempre atualizados os arquivos permanentes do sistema.

Com o Refina o sistema é tornado adaptativo, sendo um dos mais importantes módulos e o que mais atenção deve merecer.

O Busca, realiza o processamento da lógica de busca. O último dos módulos - Imprime - dá a saída do sistema imprimindo notificações aos leitores.

1.6 - RESUMO

Teve-se neste capítulo introdutório, uma visão geral dos sistemas de DSI em sua fase atual e tendências, concluindo que ainda é a maneira mais econômica, objetiva e rápida de atender a uma grande população de usuários.

Após isto, passou-se por uma fase de concepção e projeto do sistema, a qual foi subdividida em vários itens que levaram a uma visão geral do mesmo, e de suas necessidades.

Ressaltou-se, no item Modularidade a respectiva importância e se apresentou uma visualização em termos de fluxograma do funcionamento do sistema. Assim é que se tem o Módulo Geração manipulando os arquivos fonte, a fim de tornar as informações a disseminar assimiláveis pelo Módulo Busca.

CAPÍTULO II

DESCRIÇÃO DO SISTEMA

Neste capítulo detalha-se cada módulo citado no capítulo anterior.

2.1 - MÓDULO GERAÇÃO

Antes de sua análise falar-se-á sobre o modelo de indexação utilizado.

2.1.1 - MODELO DE INDEXAÇÃO

O problema era determinar como os indexadores identificariam uma obra, com um número não muito grande de informação. Essa é uma área de estudos em Ciência de Informação (Lancaster et al, 1964; Brandhorst et al, 1972), com muitos problemas ainda em aberto existindo, entretanto, algumas idéias que, se não aceitas pela totalidade dos estudiosos, são entretanto usadas atualmente. Uma delas é que, dada uma obra, tira-se do seu título e/ou resumo ("abstract") um conjunto de palavras-chave chamadas DESCRITORES, que a identificam.

Convém, entretanto, entender que o problema de indexação de um documento é, ainda, um dos pontos mais críticos em qualquer sistema - seja de recuperação, de cadastramento etc. - (merecendo, inclusive, estudos à parte, como já foi dito), tendo em vista que através dele o documento será individualizado. O que se fez, então, foi escolher um determinado processo, e a partir dele desenvolver o presente sistema.

Visto como os DESCRITORES tanto servirão para identificar documentos, como para caracterizar áreas de interesse de usuários, é necessário que, do arquivo fonte, se faça a separação entre os descritores e as informações restantes da obra, às quais chamaremos INFORMAÇÕES BIBLIOGRÁFICAS. Pode-se já entrever o processo de busca como sendo uma comparação DESCRITOR DO DOCUMENTO com DESCRITOR DO USUÁRIO.

O arquivo fonte de documentos, uma das entradas do sistema, terá, então, de ser particionado em dois, cujos conteúdos serão: AS INFORMAÇÕES BIBLIOGRÁFICAS e os DESCRITORES. O módulo GERAÇÃO terá por tarefa fazer tal particionamento.

Na Figura II.1 está um desenho do registro do arquivo fonte.

NÚMERO DE IDENTIFICAÇÃO DO DOCUMENTO	DESCRITORES	INFORMAÇÕES BIBLIOGRÁFICAS
---	-------------	-------------------------------

Fig. II.1 - Registro do Arquivo Fonte

As informações bibliográficas consistem dos seguintes itens identificadores da obra, segundo critério universal:

- Título, autor, edição, lugar de publicação, publicador, data, série, tradução, volume, título da revista, número do fascículo, páginas, mês e ano.

Na Figura II.2 vê-se o fluxograma do módulo.

O arquivo ARQUIBIB conterà as informações bibliográficas referidas acima e terá registros no formato visto na Figura II.3.

Já o DESIDEN conterà apenas os descritores e o número de identificação da obra. Vê-se, agora, a importância deste número, pois ele faz com que exista uma correspondência biunívoca entre os dois arquivos. Na Figura II.4 está representado seu registro.

No Apêndice B, está o programa em ALGOL que executa a tarefa do Módulo GERAÇÃO.

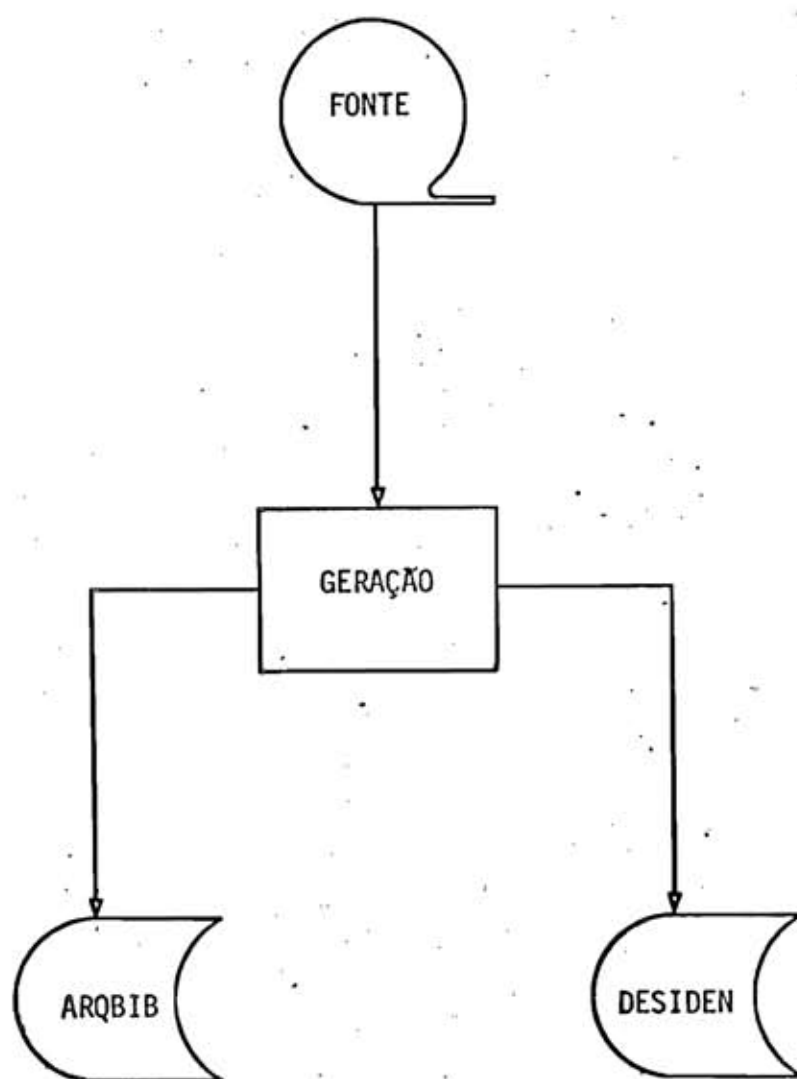


Fig. II.2 - Fluxograma do Módulo GERAÇÃO

NÚMERO DE IDENTIFICAÇÃO DO DOCUMENTO	INFORMAÇÕES BIBLIOGRÁFICA
---	------------------------------

Fig. II.3 - Registro do ARQBIB

NÚMERO DE IDENTIFICAÇÃO DO DOCUMENTO	DESCRITORES
---	-------------

Fig. II.4 - Registro do DESIDEN

2.2 - MÓDULO CADASTRA/ATUALIZA

Já foi mostrado, no Capítulo I, o modelo da ficha de ca dastramento/atualização. O leitor a preencherá de acordo com sua necesidade de se comunicar com o sistema, e este, tomando conhecimento do pedido, o executará.

Se for o caso de inicialização do sistema, obviamente sõ o cadastramento será necessário e nesta altura é montada a primeira versão de um arquivo chamado PERFIL, que conterà as informações sobre os usuários e os seus perfis, sendo este um arquivo permanente do sistema. Detalhes deste arquivo podem ser vistos no Cap. III (Implantação do Sistema) e detalhes do programa, no Apêndice C.

Na Figura II.5 vê-se um fluxograma do módulo.

2.3 - MÓDULO REFINA

Como foi dito no começo, trata-se de uma aplicação recente em sistemas de DSI. A tentativa aqui é tornar o sistema adaptativo.

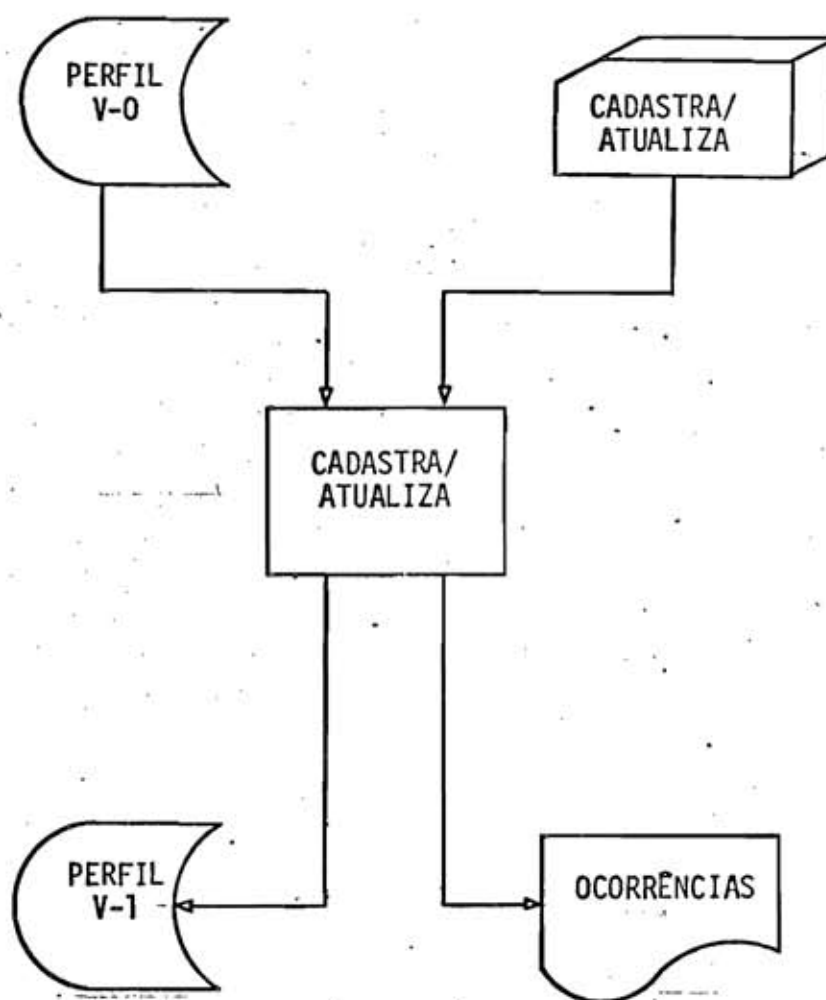


Fig. II.5 - Módulo CADAstra/ATUALIZA

Esta foi uma experiência feita por Sage et al (1965) no Laboratório AMES da Comissão de Energia Nuclear dos EEUU e que apresentou resultados positivos.

O usuário, ao receber a notificação (conforme Figura I.2), vai responder ao sistema, classificando o documento como RELEVANTE ou IRRELEVANTE. Esta informação, é codificada e introduzida no módulo, que a processará.

Como este assunto é minuciosamente estudado no Capítulo IV, serão mencionados apenas três pontos principais:

- i. Se a resposta ao documento selecionado é positiva, os valores dados aos descritores, que estavam naquele documento e no perfil do usuário, são incrementados de um certo valor, dado pela função de incremento.
- ii. Se a resposta ao documento selecionado for negativa, os valores dos descritores, que estavam naquele documento e no perfil do usuário, têm seus valores diminuídos de um valor, calculado pela função de decremento.
- iii. Todos os descritores dos perfis que coincidem com os descritores de documentos não selecionados sofrem um acréscimo, mas a taxa de incremento é menor que no caso i. A idéia é que descritores que aparecem em documentos têm algum significado para o usuário, mesmo quando o documento correspondente não é selecionado.

Na Figura II.6 se tem um fluxo do módulo e, no Apêndice D, o programa em ALGOL.

2.4 - MÓDULO BUSCA

As informações a serem disseminadas, já tratadas pelo Módulo GERAÇÃO e disponíveis no arquivo DESIDEN juntamente com a versão

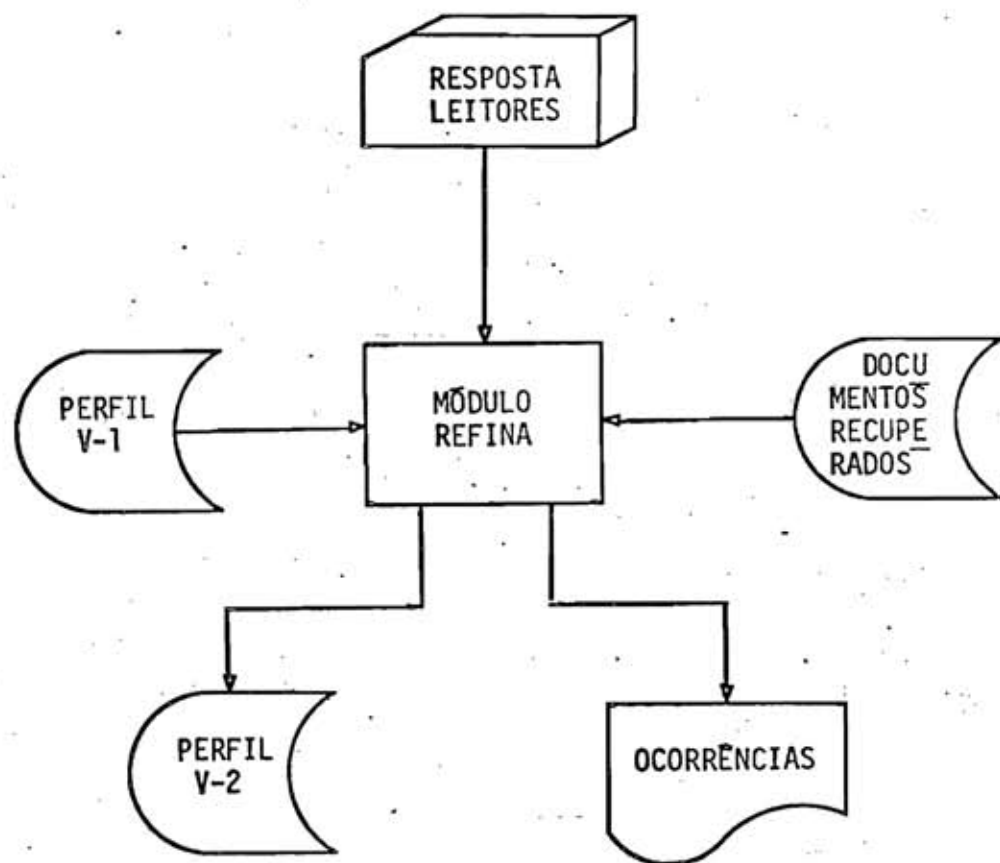


Fig. II.6 - Fluxo do Módulo REFINA

atualizada do arquivo PERFIL, compõem os parâmetros de entrada deste módulo.

De posse destes parâmetros é executado o seguinte algoritmo:

- B1. Leia um documento do DESIDEN; Se fim do arquivo, acaba execução do módulo; Execute módulo IMPRIME;
- B2. Selecione os usuários que têm os mesmos descritores que os constantes no documento;
- B3. Chame a FUNÇÃO DE DISSEMINAÇÃO, selecionando os números de perfis para os quais o documento foi considerado relevante;
- B4. Retorne ao passo B1.

No Capítulo IV estão os pormenores do passo B3, na Figura II.7 está um fluxograma do módulo e, no Apêndice E, o programa em ALGOL.

2.5 - MÓDULO IMPRIME

É o responsável pela saída do sistema e tem como parâmetros de entrada, ARQBIB, PERFIL e RECUPERADO.

RECUPERADO contém registros constituídos pela dupla: Número de identificação do documento, Número do perfil. De posse disto, o módulo faz uma pesquisa no ARQBIB, colhendo todas as informações bibliográficas que vão constar na ficha de notificação (Apêndice A) imprimindo-as e também pesquisando o arquivo PERFIL para colher os dados sobre o usuário cujo número do PERFIL consta da dupla. O produto é a notificação a ser enviada ao usuário (Apêndice A).

Na Figura II.8 vê-se o diagrama deste módulo.

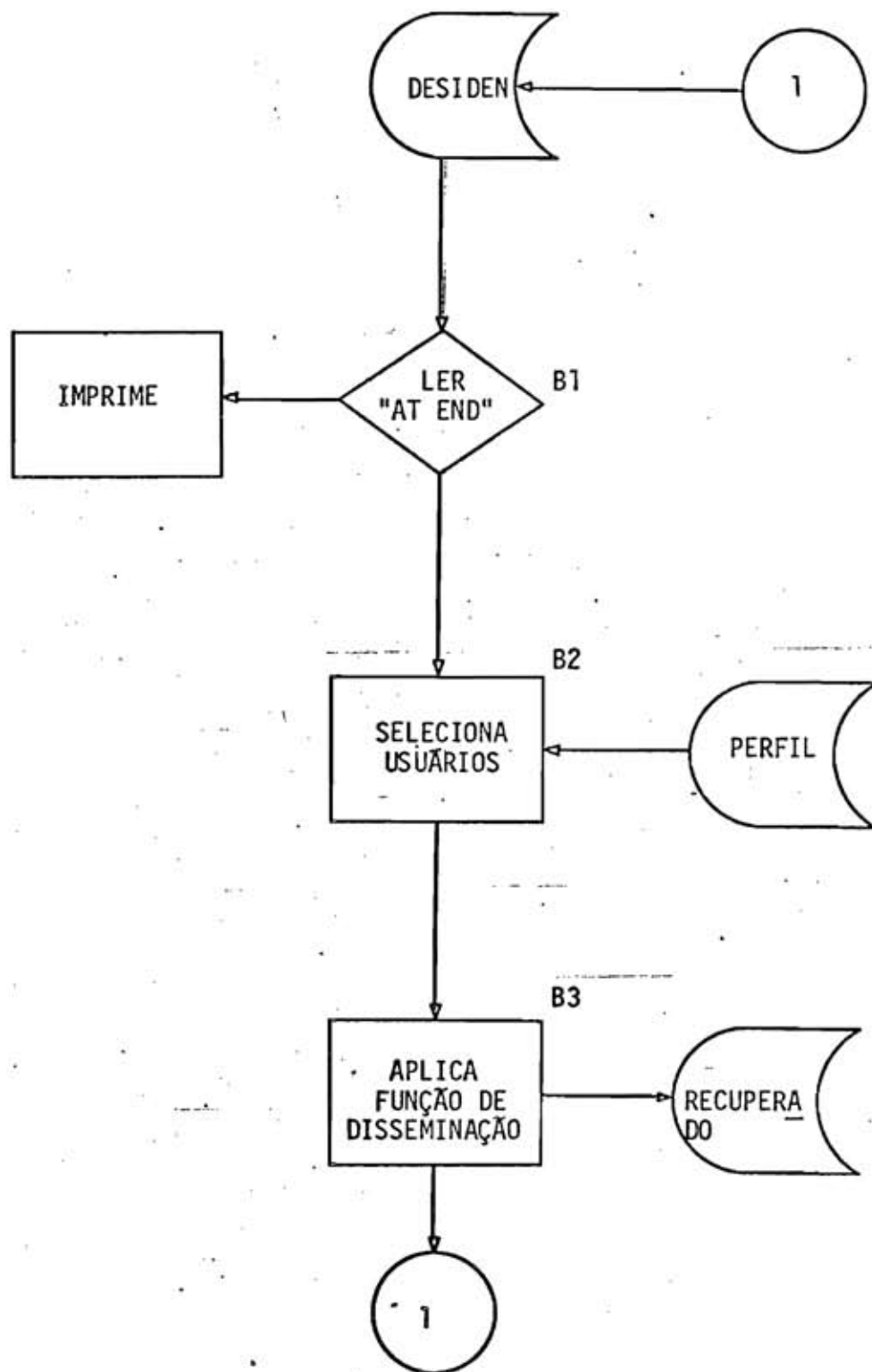


Fig. II.7 - Módulo BUSCA

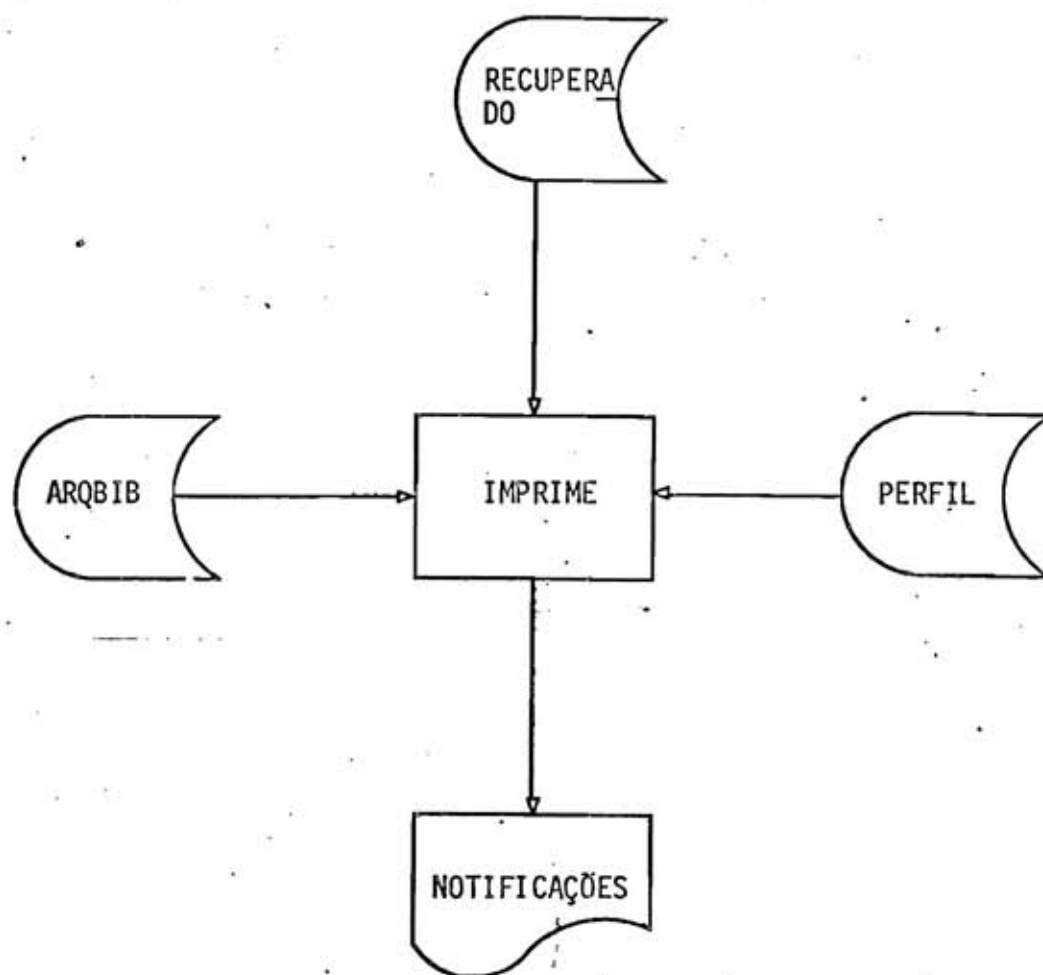


Fig. II.8 - Fluxo do Módulo IMPRIME

CAPÍTULO III

ESTRUTURAS DOS ARQUIVOS

Apresentados os pormenores dos módulos, será agora dada a especificação dos arquivos citados no capítulo anterior, justificando-a.

3.1 - ORGANIZAÇÃO DO PROCESSAMENTO

Como as execuções do sistema vão estar limitadas a períodos predeterminados, pela prioridade de chegada dos documentos a serem disseminados, e pela própria filosofia de servir ao usuário por notificações, justifica-se o uso de processamento em lotes ("BATCH").

3.2 - ORGANIZAÇÃO DOS ARQUIVOS

Nesta parte detalham-se as organizações de dados utilizadas nos diversos arquivos.

É sabido que em sistemas de recuperação de informação, a preocupação maior é uma rápida pesquisa nos dados armazenados, sendo as atualizações processadas mais lentamente. Mas adiante, usa-se este fato para explicar a escolha de uma estrutura chamada ÁRVORE COM PALAVRAS-CHAVE DE COMPRIMENTO FIXO (Lefkovitz, 1969).

Os arquivos utilizarão as organizações conhecidas como:

- SEQUENCIAL
- LISTAS
- ÁRVORE

Como alguns dos arquivos têm a mesma organização, após explicadas as estruturas usadas serão indicados os arquivos que as utilizam.

3.3 - ARQUIVOS SEQUENCIAIS

Como se sabe, este é o método mais conhecido de organização de arquivos de informações, os registros sendo colocados em localizações consecutivas. Na ordenação dos registros, um atributo comum é escolhido, aqui chamado de CHAVE DE CLASSIFICAÇÃO. Como os arquivos serão armazenados em dispositivos de acesso direto ("DASD"), no caso, disco magnético, a desvantagem citada para este método, ou seja, a pesquisa de registros isolados, não existe aqui, visto como é utilizado o acesso aleatório, fazendo uma pesquisa binária (Knuth, 1975). A outra desvantagem apresentada, a atualização, não causará problemas, pois dos arquivos que utilizam esta organização, 3 (três) são temporários e um quarto, como se verá, embora permanente, tem as atualizações colocadas sempre no fim.

Neste tipo de organização há os seguintes arquivos:

i. ARQBIB E DESIDEN

Como se sabe, estes arquivos são gerados pelo módulo GERAÇÃO e têm registros no formato apresentado no capítulo anterior, Figuras II.3 e II.4, respectivamente.

A ordenação está por conta do "NÚMERO DE IDENTIFICAÇÃO DO DOCUMENTO", que é a chave de classificação dos arquivos.

Eles são temporários, existindo apenas quando da execução do sistema.

ii. RECUPERADO

Montado no módulo BUSCA, ele tem como função armazenar os documentos recuperados, individualizados por seus números de identificação, armazenando também os números de perfis dos usuários, para os quais tais documentos serão recuperados para cada usuário, é preciso ter os registros com formato de tamanho variável.

Terminado o Módulo BUSCA, este arquivo - contendo os números de perfis ordenados de maneira crescente e, associado a cada um, os números de identificação dos documentos selecionados - será utilizado pelo módulo IMPRIME, que lerá sequencialmente os números de perfis, buscando maiores informações sobre o usuário no arquivo "USUÁRIOS", abaixo explicado, imprimindo-as e, de modo semelhante, utilizando o arquivo "ARQBIB", para dele retirar informações sobre os documentos, montando assim, o produto final do sistema.

A Figura III.1 vê-se o formato dos registros. O campo "NÚMERO DO PERFIL" é autodefinido. "NÚMERO DE DOCUMENTOS" é um contador que indica quantos documentos se encontram a partir do próximo campo ("NÚMERO DO DOCUMENTO 1", "NÚMERO DO DOCUMENTO 2" etc., são os números de identificação de tais documentos).

iii. USUÁRIOS

Arquivo permanente cujo propósito é guardar informações sobre o usuário cadastrado. Não exige muitas explicações. Basta acrescentar que ele é ordenado pelo número do perfil e, na Figura III.2, está o formato dos registros.

NÚMERO DO PERFIL	NÚMERO DE DOCUMENTOS	NÚMERO DO DOCUMENTO 1	NÚMERO DO DOCUMENTO 2	
------------------	----------------------	-----------------------	-----------------------	-------

Fig. III.1 - Formato do Registro do RECUPERADO

NÚMERO DO PERFIL	NOME	ENDEREÇO	DATA DE CADASTRAMENTO
------------------	------	----------	-----------------------

Fig. III.2 - Registro do USUÁRIOS

Apresentados, portanto, os arquivos seqüenciais, falar-se-á agora sobre o outro arquivo permanente. Trata-se do arquivo PERFIL. O USUÁRIO e este são os dois únicos permanentes. Na realidade, ele é composto por dois arquivos:

- Um contendo todos os descritores dos usuários inscritos, que é chamado de DESCRITORES e outro,
- contendo os números dos usuários inscritos que usam aqueles descritores, e respectivos pesos, chamado USUCOD. Na figura III.3, está uma esquematização dos mesmos:

3.4 - O ARQUIVO DE DESCRITORES

De acordo com o fluxograma do módulo BUSCA, apresentado na Figura II.7, sabe-se que após ser obtido um documento, com seus descritores, devem ser recuperados os usuários que têm interesse naquele documento. De acordo com o mesmo fluxograma, isto é feito utilizando-se o arquivo PERFIL, determinante do desempenho do módulo, no que diz respeito ao tempo de execução, velocidade de recuperação e otimização de espaço ocupado. A estrutura adotada satisfaz a estes requisitos, conforme será mostrado.

Pelo modelo de indexação, a recuperação está a cargo de DESCRITORES, ou, em outros termos, o arquivo (PERFIL) é passível de recuperação lógica através desta chave. Funcionalmente, pode-se dizer que a "CHAVE" representa uma partição simples do arquivo, ou de uma lista; estruturalmente, é um subcampo de um registro (Lefkovitz, 1969). Então, cada registro, no arquivo que contém o subcampo chave, é dito estar na partição do arquivo caracterizado pela mesma.

No caso em pauta, entra-se no arquivo PERFIL com a chave DESCRITOR e, a partir daí, recupera-se toda uma partição do arquivo ou lista, que traz os usuários que utilizam aquele descritor, com os "pesos" correntes associados aos mesmos.

ARQUIVO
DESCRITORES

DESCRITOR 1	...	DESCRITOR k
-------------	-----	-------------

DESCRITOR k + 1	...	DESCRITOR j
--------------------	-----	-------------

DESCRITOR p	...	DESCRITOR n
-------------	-----	-------------

NÚMERO DO PERFIL	DADOS

ARQUIVO
USUCOD

Fig. III.3 - Relação entre DESCRITORES E USUCOD

Sabendo isto, é possível adotar uma das seguintes soluções:

- a) Colocar num registro físico contíguo o descritor e todos os usuários;
- b) Separar, fisicamente, os descritores dos usuários.

As desvantagens do primeiro método são:

- 1) Não se sabe de antemão quantos indivíduos escolherão cada descritor, ficando assim impossível o controle do tamanho do registro.
- 2) O "manuseio" dos registros na fase de operação para recuperação traria problemas. É uma consequência direta da primeira desvantagem. A operação de tais registros, de tamanho variável, tornaria o módulo BUSCA mais lento, quando o contrário deve acontecer.
- 3) Repetição desnecessária do número do perfil da pessoa a cada novo descritor acarretando assim, no pior caso (todos usando todos os descritores) uma má utilização dos espaços de armazenamento.

Já a segunda solução elimina inconvenientes, além de que a separação física existe, por ser o arquivo PERFIL composto de dois arquivos. (Deve ficar claro, que há um registro com todas as informações armazenadas no PERFIL). Na Figura III.4 está o seu esquema.

DESCRITOR	Nº DO PERFIL	PESO	Nº DO PERFIL	PESO	
-----------	--------------	------	--------------	------	-------

Fig. III.4 - Registro Lógico do Arquivo PERFIL

(O campo "peso" representa a probabilidade que será usada pela função de disseminação (Cap. IV)).

Após decidir pela separação, o problema foi optar por uma organização que permitisse as vantagens inicialmente mencionadas (item 3.1), para ambos os arquivos, levando em conta o desempenho de cada um. Veja-se a análise abaixo:

- a - No caso do DESCRITORES, o seu crescimento não pode ser preciso. Vai depender da diversidade de assuntos a disseminar, uma grande expansão sendo inevitável, pois os indivíduos escolherão seus descritores, não existindo, como na maioria dos sistemas de DSI, um número pré-fixado daqueles, nos quais o usuário tem que se enquadrar.
- b - Claramente o caso em pauta se enquadra no problema de busca externa, desde que se deseja recuperar informações de arquivos muito grandes, que estão armazenados em unidades de armazenamento secundário, no presente caso, discos magnéticos.

Para tais casos as estruturas em árvores são apoiadas por Knuth (1975) como a melhor solução, desde que se escolha uma maneira apropriada de representá-las. Lefkovitz (1969) apresenta um modelo de organização de arquivos, baseado em árvores de múltiplas ramificações, que são as árvores com palavra chave de comprimento fixo, sendo esta a organização escolhida para o arquivo DESCRITORES.

Existem dois tipos bem definidos de nós nesta árvore: Terminais e Não-Terminais. Nos nós terminais, ou "folhas", encontram-se os campos do valor da chave e da informação referente a esta, conforme Figura III.5.

Nos nós não-terminais, estão campos dos valores das chave e os ponteiros para outros nós da árvore, conforme Figura III.6.

CHAVE	INFORMAÇÃO	CHAVE	INFORMAÇÃO	
-------	------------	-------	------------	-------

Fig. III.5 - NÓ Terminal

PONTEIRO	CHAVE	PONTEIRO	CHAVE	
----------	-------	----------	-------	-------

Fig. III.6 - NÓ Não-Terminal

O campo "informação" será o endereço para o arquivo USUCOD, e as "chaves", são os descritores.

Define-se a árvore em função de dois parâmetros (Vasconcelos et al, 1976) "m" e "p" com relação aos quais ela deve satisfazer as seguintes propriedades:

1. Todas as folhas estão num mesmo nível, e contêm p valores de chave, com exceção do nó mais à direita deste nível, que contém de 1 a p valores de chave.
2. Num nível que não o das folhas, todos os nós possuem m filhos, com exceção do nó mais à direita deste nível, que possui de 1 a m filhos.
3. Num nó não-terminal, o número de valores de chaves que ele contém é igual ao número de filhos que possui.
4. O único nível no qual existe apenas um nó é o nível da raiz.
5. Num nó com k valores de chave, ocorre sempre:

$$K_1 < K_2 < K_3 < \dots < K_{n-1} < K_n .$$

6. Dado um par (p_i, k_i) num nó não-terminal, p_i aponta para a raiz de uma Sub-árvore onde, para todo valor de chave K da Sub-árvore, se tem $K < K_i$.
7. Dado um nível qualquer da árvore, um valor de chave K pertence a, no máximo, um nó deste nível.
8. Todo valor de chave K_i do conjunto dado, pertence a uma folha da árvore.

Estas condições são necessárias mas não suficientes para a definição da árvore (m, p) . As condições de suficiência, são encontradas no trabalho de Vasconcelos e Bersano (1976) e se referem a um algoritmo, o CMP, que gera a árvore única.

Na Figura III.7 é apresentada a árvore $(3,3)$, correspondente a um conjunto de nomes próprios. Estão omitidos os campos de "informação" das folhas.

As Figuras III.8 e III.9 seguintes, apresentam respectivamente, desenhos dos nós terminais e não-terminais do arquivo.

3.5 - O ARQUIVO USUCOD

Trata-se do segundo arquivo que compõe o de nome PERFIL. Adotou-se uma estrutura de lista chamada "MULTILISTA", também conhecida como "LISTA COM MÚLTIPLAS COSTURAS" (Lefkovitz, 1969). Deve-se frisar que estas estruturas, juntamente com a árvore de palavra chave com comprimento fixo, são grandemente usadas em recuperações em tempo-real ("ON LINE") devido à facilidade de manuseio.

Para facilitar o entendimento, a Figura III.10 apresenta uma folha da árvore e o arquivo USUCOD. Na folha está a tripla: número de usuários que usam o descritor (o comprimento da lista); endereço do primeiro usuário (cabeça de lista); descritor (chave). É mostrado o detalhe de um registro do USUCOD, representado por um ponto; outra

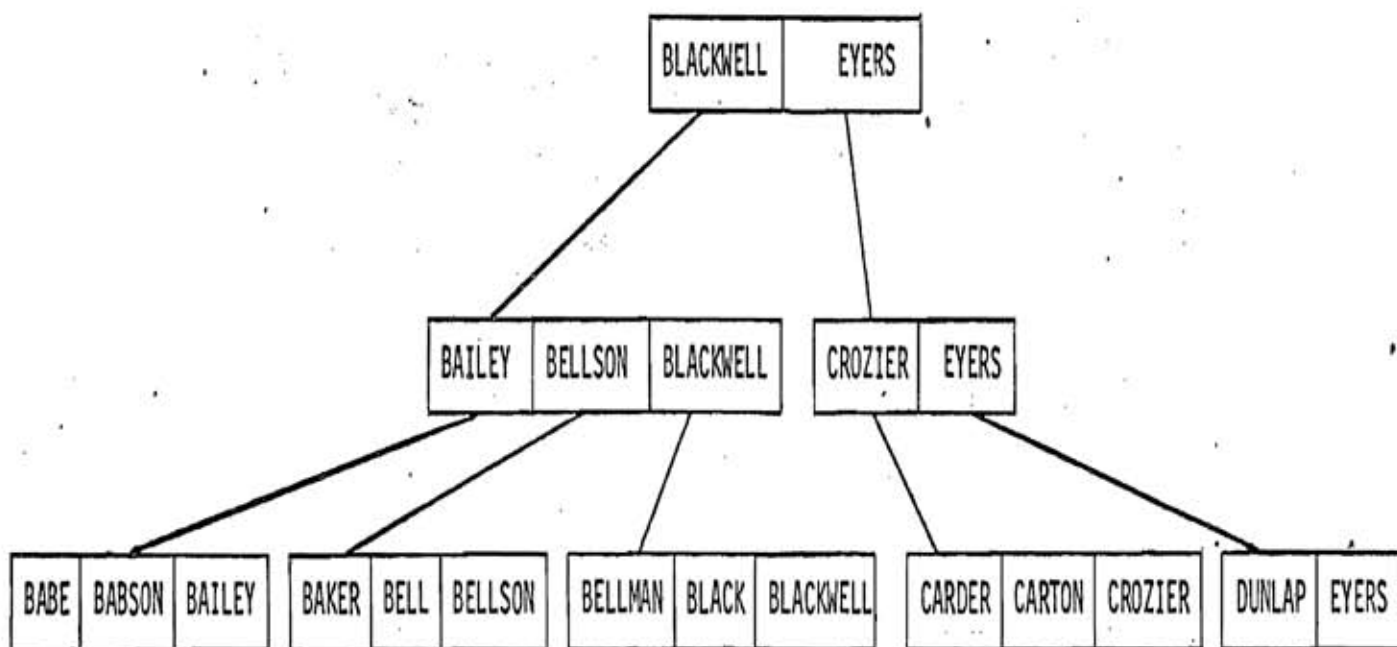


Fig. III.7 - Exemplo da Estrutura do Arquivo DESCRITORES

Nº DE USUÁRIOS	PONTEIRO P/ USUCOD	DESCRIPTOR	Nº DE USUÁRIOS	PONTEIRO P/ USUCOD	DESCRIPTOR	
-------------------	-----------------------	------------	-------------------	-----------------------	------------	-------

Fig. III.8 - Desenho do Nó Terminal

PONTEIRO	DESCRIPTOR	PONTEIRO	DESCRIPTOR	
----------	------------	----------	------------	-------

Fig. III.9 - Desenho do Nó Não-Terminal

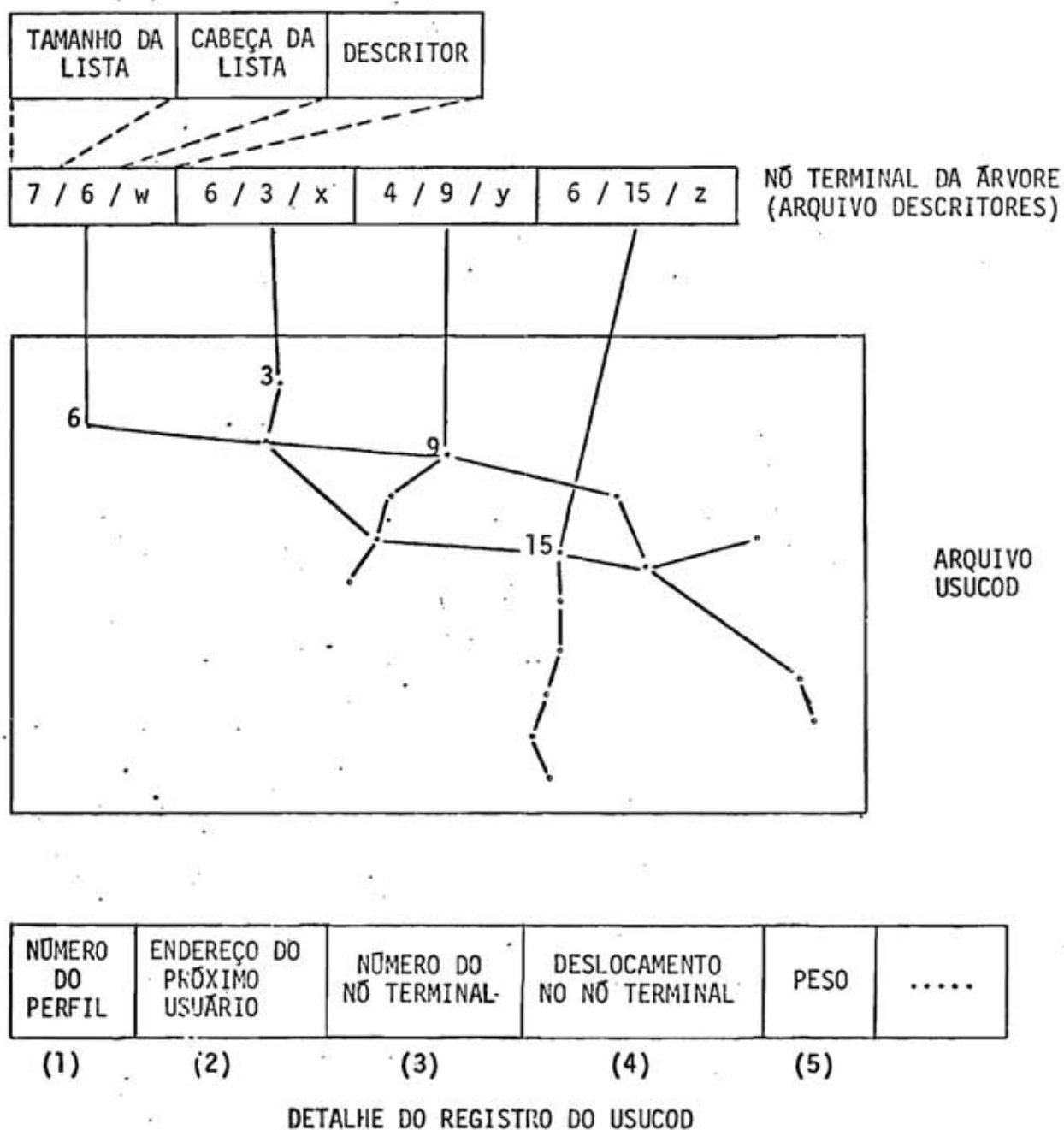


Fig. III.10 - Arquivo USUCOD e Detalhe do seu Registro

observação a fazer é que, embora o desenho apresente os registros (na figura com pontos) e suas comunicações com linhas simples, unidirecionais, na realidade, o que existe são listas estruturadas multidirecionais; isto permite que se percorra ("SCAN") todo o arquivo de uma maneira geral. As listas multidirecionais são uma generalização das multilistas uni e bidirecionais apresentadas por Lefkovitz (1969). Ele cita também, e se constata, que as multilistas são de fácil programação, permitindo ainda a otimização de espaços e velocidade de atualização (mesmo não necessários neste ponto), tornando, assim, o sistema bastante flexível no que se refere a tempo de busca.

O número de repetições dos campos (2) a (5) é função do número de descritores que cada usuário escolheu. No presente trabalho, este número será no máximo 20 (vinte). Se se desejar aumentar este número, não haverá problema com os programas, sendo suficiente alterar as especificações referentes a tais registros.

No campo (2) é um ponteiro para o próximo usuário que utiliza o mesmo descritor que este. A identificação do descritor está por conta dos campos (3) que situa o interessado num dos nós terminais e o (4) que o posiciona neste descritor. Claro está que num nó, há vários descritores, (são as células do nó); "peso" é um campo que será usado pela função de disseminação e função de incremento-decremento, baixo incremento (Capítulo IV) e que, na realidade, contém a probabilidade que representa um valor de significância para o usuário.

3.6 - CONTROLES

Este é um pequeno item, relacionado com a matrícula do usuário no sistema. A cada indivíduo inscrito, é atribuído um número, num sistema semelhante ao utilizado no INPE, que é o módulo 11. Surge então um código numérico, acrescido de valores também numéricos, chamados dígitos verificadores. Com isto, o código torna-se autoverificável. O trabalho de geração de tais números foi também realizado no

INPE, por Buss e Teixeira (1975), do qual foram tirados os números utilizados no trabalho de matrícula. A precisão do método, já detetada por um sistema foi de 1000/1.

3.7 - RESUMO

Em sistemas de recuperação de informação a preocupação maior é uma rápida pesquisa nos dados armazenados, sendo as atualizações processadas mais lentamente. Isto, aliado ao fato de se querer recuperar informações em arquivos são comportados em memórias secundárias (discos magnéticos), levou à adoção de uma estrutura em árvore, apontada por Knuth (1975) como a melhor solução, desde que se escolha uma maneira apropriada de representá-la. Lefkovitz (1969) apresenta um modelo de organização de arquivos, baseado em árvores de múltiplas ramificações, que são as árvores com palavra chave de comprimento fixo. Esta foi utilizada para nosso arquivo DESCRITORES, que preenche os primeiros requisitos citados.

A lista com estruturas chamadas "MULTILISTAS", também conhecida como "LISTA COM MÚLTIPLAS COSTURAS" (Lefkovitz, 1969), juntamente com a árvore de palavra chave de comprimento fixo são grandemente usadas em recuperações em tempo-real ("ON LINE"), devido à facilidade de manuseio. Nesta estrutura caiu a escolha para o arquivo USUCOD.

Com isto, se conseguiu uma fácil programação, permitindo ainda otimização de espaços e velocidade de atualização ficando, assim, o sistema bastante otimizado.

CAPÍTULO IV

FUNÇÃO DE DISSEMINAÇÃO

4.1 - INTRODUÇÃO

Um sistema de Disseminação Seletiva de Informações é um sistema que retém, em unidades de armazenamento (memória de massa) um grupo de usuários, indexados por descritores. Periodicamente, o sistema recebe uma coleção de documentos, também indexados por descritores, e faz a disseminação seletiva para os usuários, isto é, envia para cada usuário, os documentos selecionados para ele conforme um critério de BUSCA (ou, simplesmente, notifica-os da existência dos documentos).

Mais especificamente, um sistema de disseminação seletiva de documentos é aqui definido como consistindo de:

- . um conjunto de perfis;
- . um conjunto de documentos a serem disseminados;
- . uma função de disseminação, f .

f pode ser:

- . uma das quatro funções apresentadas a seguir; ou
- . uma função de probabilidade, usando um limiar ("threshold").

4.2 - ESTRATÉGIAS DE BUSCA

Embora o Grupo de Interesse Especial em DSI, da Sociedade Americana para Ciência da Informação, tenha listado seis possíveis estratégias de BUSCA no documento por ele preparado (Housman, 1969), quatro técnicas básicas podem ser reveladas como mais importantes:

1. MUST, MAY, NOT

Strain Aging
Tensile Properties
Time Dependence
Vacancies
Voids

Irradiation
Lithium Fluorides
Plasticity
Radiation Effects
Tensile Properties
Thermionic Emission
Time Dependence
X Radiation

Um batimento ("match") de percentagem é determinado pelo usuário. Se ele escolhe 30%, o fator P deve ser $P = 30\%$. No exemplo, o número de descritores iguais para usuário/documento é de 6. O número total de descritores do documento é 18. Então,

$$P = \frac{6}{18} \times 100 = 33 \frac{1}{3} \%$$

Esta técnica, porém, não recebeu muito apoio. Em 1969, num trabalho de Housman, ele mostra que apenas quatro instituições usavam esta técnica. Um dos principais problemas é a quantidade de descritores atribuídos a um documento visto que ser esta quantidade relevante na seleção do documento. Um número mínimo de descritores por documento teria que ser especificado, a fim de que, garantidamente, P alcance um limiar predeterminado.

3. LÓGICA BOLEANA

Os DESCRITORES do PERFIL do usuário são combinados numa função Booleana cujos operadores lógicos são OU, E, NÃO. O operador OU expressa a condição ser aceitável qualquer um dos predicados registrados ou qualquer um dos termos presentes numa sentença. O operador E é usado para expressar a condição de presença de qualquer termo de um dado grupo, junto com qualquer um dos termos de outro grupo dado. O uso

Strain Aging
Tensile Properties
Time Dependence
Vacancies
Voids

Irradiation
Lithium Fluorides
Plasticity
Radiation Effects
Tensile Properties
Thermionic Emission
Time Dependence
X Radiation

Um batimento ("match") de percentagem é determinado pelo usuário. Se ele escolhe 30%, o fator P deve ser $P = 30\%$. No exemplo, o número de descritores iguais para usuário/documento é de 6. O número total de descritores do documento é 18. Então,

$$P = \frac{6}{18} \times 100 = 33 \frac{1}{3} \%$$

Esta técnica, porém, não recebeu muito apoio. Em 1969, num trabalho de Housman, ele mostra que apenas quatro instituições usavam esta técnica. Um dos principais problemas é a quantidade de descritores atribuídos a um documento visto que ser esta quantidade relevante na seleção do documento. Um número mínimo de descritores por documento teria que ser especificado, a fim de que, garantidamente, P alcance um limiar predeterminado.

3. LÓGICA BOLEANA

Os DESCRITORES do PERFIL do usuário são combinados numa função Booleana cujos operadores lógicos são OU, E, NÃO. O operador OU expressa a condição ser aceitável qualquer um dos predicados registrados ou qualquer um dos termos presentes numa sentença. O operador E é usado para expressar a condição de presença de qualquer termo de um dado grupo, junto com qualquer um dos termos de outro grupo dado. O uso

de NÃO causa a eliminação de todos os itens contendo qualquer um dos descritores no grupo ao qual o operador se refere. Um exemplo de um comando Boleano é:

$$(A + B + C) . (D + E + F) . (- G)$$

onde os símbolos ., +, -, são usados respectivamente para E, OU e NÃO e A, B, C, ... são descritores do perfil do usuário. Aninhamentos são possíveis. Limitação no número de níveis (parênteses), como do número de descritores por perfil e das combinações possíveis, dependem da implementação.

4. TERMOS COM PESO

No caso de termos com peso, a cada descritor no perfil do usuário é atribuído um peso, de acordo com a importância relativa atribuída ao descritor. Os pesos são somados algebricamente e comparados com um limiar. Um batimento ("match") para verificar é feito se a soma dos pesos dos descritores de um documento se iguala ou excede este limiar. É permitido o uso de peso negativo. Exemplo:

valor do limiar 5

termo A peso 3

termo B peso 6

termo C peso -2

Isto significa que os documentos cujos perfis contenham termo A sozinho não serão selecionados. Os que contiverem o termo B serão selecionados, observando que eles não contenham termo C também.

Um levantamento dos sistemas de DSI (Bivona et al, 1966), revelou que a estratégia de busca mais usada era a de termos com peso, embora ambas, Boleana e de Peso, dêem resultados de busca satisfatórios, considerando precisão e atendimento ("recall"). Entretanto, existe

uma otimização considerável de tempo de UCP quando a busca por termos com peso é usada em vez da Lógica Boleana. Testes feitos no Centro de Informações Técnicas e Científicas da NASA revelaram que existia uma relação de 4 para 1 de vantagem de tempo de UCP quando operando num modo aritmético ou com peso.

4.3 - FUNÇÃO DE DISSEMINAÇÃO

Será, agora, apresentada a função de probabilidade, uma das citadas na Introdução, já mencionada em vários capítulos anteriores.

Pode-se dizer que existem, genericamente, duas categorias de funções de disseminação, quais sejam, booleana e aritmética. A disseminação para cada uma delas dá-se da seguinte maneira:

A cada documento do sistema está associado um vetor de índices (descritores), y , o qual indica quais os termos usados para indexar o documento. Analogamente, um perfil pode ser representado por um vetor de termos, x .

Num sistema do tipo booleano cada vetor y é examinado para ver se satisfaz às condições especificadas pelo perfil, x , e como resultado, haverá ou não, recuperação.

Os sistemas de disseminação do tipo aritmético, nos quais está o de probabilidade usando limiar, envolvem o processamento dos vetores x e y como quantidades. Mais comumente, o processo de disseminação é realizado através da computação de uma função de disseminação $f(x, y)$, sendo o documento recuperado quando $f(x, y)$ excede um determinado valor, o do limiar, pré-estabelecido.

Atualmente, muito esforço vem sendo aplicado neste método de disseminação, visto que ele oferece simplicidade na computação e é usado como base para algoritmos de agrupamento ("cluster").

Problema importante num sistema tipo aritmético é a escolha da função de disseminação, e este é um fator determinante para o desempenho do sistema.

Sage et al. (1965), propõem uma função de disseminação do tipo de probabilidade, usando limiar. Será descrita a técnica, apresentando primeiro a função de probabilidade e logo após será explicado como funciona o limiar. Para um melhor entendimento do que vem a seguir, observe-se que P_i representa o valor de probabilidade que um usuário dá a um descritor D_i do seu perfil. T_i é o valor de probabilidade obtido pelo sistema para um documento. Observe-se, também, que $0 < P_i \leq 1$, e $\sum_i P_i = 1$.

Se, para um documento, existe um só descritor D_1 que também pertence ao perfil de um usuário, com valor de probabilidade de P_1 , o sistema atribuirá ao documento um valor T_1 igual ao dado pelo leitor, ou seja, $T_1 = P_1$. Neste caso, a relevância dada ao documento pelo sistema coincide com a que o usuário está dando a ele.

Suponha-se, agora, que um segundo descritor D_2 pertence tanto ao documento como ao perfil. Neste caso, o sistema, fará a "soma" dos dois valores de probabilidade, atribuindo, assim, um valor a T_2 , ou seja:

$$\begin{aligned} T_2 &= P(D_1 \cup D_2) = P(D_1) + P(D_2) - P(D_1) \cdot P(D_2) \\ &= P_1 + P_2 - P_1 \cdot P_2 \end{aligned}$$

considerando que os descritores sejam independentes. Substituindo $P_1 = T_1$, vem:

$$T_2 = T_1 + P_2 - T_1 \cdot P_2$$

Supondo agora que um terceiro descritor D_3 , também faça parte do documento e do perfil do usuário então virá:

$$\begin{aligned} P(D_1 \cup D_2 \cup D_3) &= P([D_1 \cup D_2] \cup D_3) = P(D_1 \cup D_2) + \\ &+ P(D_3) - P(D_1 \cup D_2) \cdot P(D_3) \end{aligned}$$

substituindo $P(D_1 \cup D_2)$ por T_2 e $P(D_2)$ por P_2 tem-se:

$$P(D_1 \cup D_2 \cup D_3) = T_2 + P_3 - T_2 \cdot P_3 ,$$

fazendo $T_3 = P(D_1 \cup D_2 \cup D_3)$ vem:

$$T_3 = T_2 + P_3 - T_2 \cdot P_3 .$$

Analogamente, se D_4 também faz parte do documento e do perfil do usuário

$$T_4 = P(D_1 \cup D_2 \cup D_3 \cup D_4) = P([D_1 \cup D_2 \cup D_3] \cup D_4) = P(D_1 \cup D_2 \cup D_3) + P(D_4) - P(D_1 \cup D_2 \cup D_3) \cdot P(D_4) ,$$

ou

$$T_4 = T_3 + P_4 - T_3 \cdot P_4 .$$

Uma fórmula geral seria então expressa como:

$$T_k = T_{k-1} + P_k - T_{k-1} \cdot P_k ,$$

com k variando no conjunto interseção de descritores documentos/perfil.

Será visto, agora, o que vem a ser o limiar ("threshold") aplicado à função anterior.

Como foi visto, o sistema calcula um valor para T_k , que é a probabilidade que o mesmo associa ao documento. Entra em pauta agora a questão de quando será disseminado o documento ao usuário, ou seja, quando o sistema considera este documento relevante.

Um valor médio para o subconjunto de documentos relevantes e não-relevantes pode ser determinado. A Figura IV.1 mostra graficamente um exemplo disto

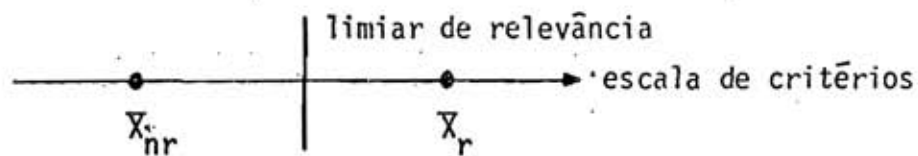


Fig. IV.1 - Limiar de Relevância

O ponto $\bar{X}_{nr}$ seria o valor médio para documentos identifi-
cados como não-relevantes. O ponto $\bar{X}_r$, o valor médio para documentos
identificados como relevantes. Um dos procedimentos usados para a deter-
minação do critério de relevância emprega a análise discriminante. Esta
técnica estatística calcula a função discriminante que maximiza a sepa-
ração dos centróides (médias) das variáveis de critério $\bar{X}_r$ e $\bar{X}_{nr}$. (Cook,
1975). Considerando as distribuições para os documentos relevantes e ir-
relevantes aproximadamente iguais, pelo critério acima, 0.5 seria um
valor para o limiar, e este será o adotado no nosso sistema.

No algoritmo de BUSCA, aqui proposto, isto se refletiria
no passo B3, onde, se aplica a função

$$T_k = P_k + T_{k-1} (1 - P_k) ,$$

para k variando de 1 ao número de descritores que constam no perfil e
no documento. O valor P_k é obtido no registro do arquivo PERFIL (Figura
III.4). Inicialmente, $T_0 = 0$, e, por isso,

$$T_1 = P_1 ,$$

seria o caso de só um descritor. Em seguida, aplica-se recursivamente
 T_k até o limite máximo de k.

Após isto, faz-se o teste:

$$T_k > 0.5? ,$$

caso a resposta seja positiva, o documento é recuperado e em caso contrário, ele é considerado irrelevante.

4.4 - FUNÇÃO DE INCREMENTO-DECREMENTO

Feita a disseminação dos documentos, usando a função de probabilidade, enviam-se aos usuários as fichas contendo o material selecionado (ver Apêndice A).

O usuário, de posse das fichas, verá quais documentos lhe parecem relevantes, ou não, assinalará isto na ficha e a devolverá ao sistema.

(Uma observação: quando foi feita a seleção dos documentos, armazenaram-se os descritores destes que estavam nos perfis dos usuários. Com as respostas em mão e com este arquivo, monta-se um arquivo novo, contendo os números de perfis dos usuários com respectivos valores de probabilidade a serem incrementados ou decrementados, dependendo do tipo de resposta).

Existem dois tipos de mecanismos de realimentação. A função de realimentação normal, que incrementa ou decrementa, conforme a resposta do usuário, e a função de realimentação anormal, que é uma função de baixo-incremento aplicada a descritores de documentos selecionados mas não disseminados, devido ao valor da função de disseminação ter sido menor ou igual a 0.5. Corrige-se, assim, suavemente, as probabilidades inicialmente atribuídas àqueles descritores.

A função de incremento-decremento é tal que, para valores na vizinhança de 0.5 (limiar), há incrementos ou decrementos mais significativos, de acordo com a relevância ou não, dada pelo usuário ao documento, na sua resposta, visando isto a se alcançar rapidamente um valor de equilíbrio. Isto é possível devido à natureza da curva - seno - usada pela função (ver Figura IV.2).

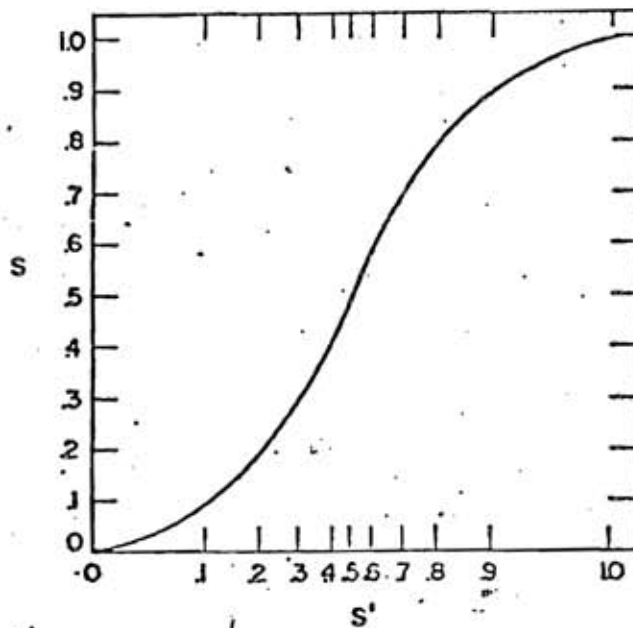


Fig. IV.2 - Gráfico da Função Transposta

À medida que os valores se aproximam de 0 e 1, os incrementos e decrementos tornam-se menores.

Os incrementos e decrementos dados aos descritores dos perfis são determinados da seguinte maneira:

Sejam S = valor de probabilidade do descritor

S'_1 = valor do descritor transposto ao primeiro quadrante da curva seno

S'_4 = valor do descritor transposto ao quarto quadrante

S^* = novo valor do descritor, incremento ou decremento

DIV = variável de divisão

$$\text{se } S \geq 0.5, \quad S'_1 = \arcseno(2S-1) \quad (\text{IV.1})$$

$$S < 0.5, \quad S'_4 = \arcseno(1-2S) \quad (\text{IV.2})$$

para respostas positivas, a função de incremento é:

$$\text{se } S \geq 0.5, S^* = (\text{sen}(S_1' + (1.5708 - S_1')/\text{DIV}))/2 \quad (\text{IV.3})$$

$$S < 0.5, S^* = (1 - \text{sen}(S_4' - (1.5708 - S_4')/\text{DIV}))/2 \quad (\text{IV.4})$$

para resposta negativas, a função de decremento é:

$$\text{se } S \geq 0.5, S^* = (\text{sen}(S_1' - (1.5708 - S_1')/\text{DIV}))/2 \quad (\text{IV.5})$$

$$S < 0.5, S^* = (1 - \text{sen}(S_4' + (1.5708 - S_4')/\text{DIV}))/2 \quad (\text{IV.6})$$

Foram feitas várias experiências para diversos valores de DIV, conforme tabelas do Apêndice F, pode-se observar que o valor de DIV=8 foi o que produziu melhores resultados.

Nas equações acima, de incremento-decremento, o primeiro quadrante do seno é usado no cálculo de S^* para todos $S \geq 0.5$ (ver Figura IV.2) e o quarto quadrante é usado para calcular S^* para todos $S < 0.5$. Os dois quadrantes da curva são ilustrados na Figura IV.2.

4.5 - IMPLEMENTAÇÃO DA FUNÇÃO INCREMENTO-DECREMENTO

No item anterior foi apresentada a fórmula usada por Sage et al (1965), no DSI do laboratório AMES.

Visando obter respostas mais rápidas quando da avaliação da função, algumas alterações foram introduzidas, no sentido de utilizar-se as intrínsecas SIN e ARCSIN, permanecendo o comportamento da função inalterado.

Tais alterações foram:

- 1 - Translação de eixos.
- 2 - Mudança de escala dos valores de probabilidade, o intervalo de variação sendo agora (-1, 1) e não mais (0, 1).

Isto fica explicitado com o novo gráfico agora obtido para a função (Figura IV.3).

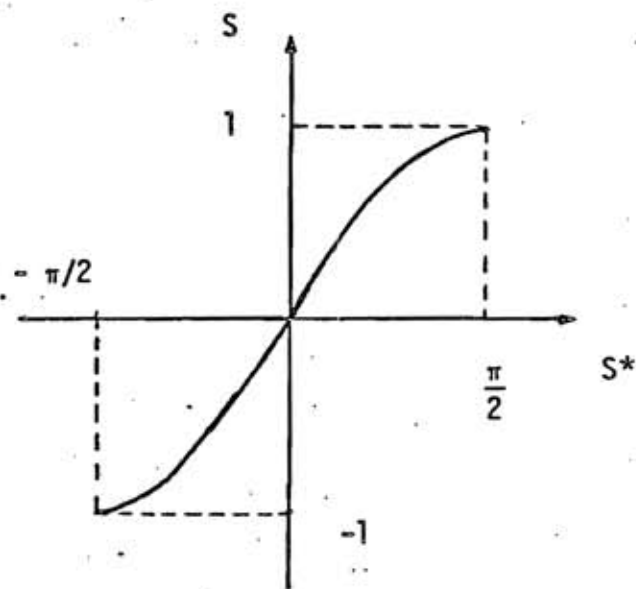


Fig. IV.3 - Gráfico da Função Incremento/Decremento como Finalmente Usada

Deste modo pode-se descrever a implementação do seguinte modo:

Sendo S um valor de probabilidade,

1 - calcula-se o valor correspondente na nova escala, fazendo

$$S_1 = 2S - 1$$

2 - calcula-se a função inversa, ou seja

$$S^* = \text{ARCSIN}(S_1)$$

3 - acha-se o deslocamento

$$\Delta = (1.5708 - \text{ABS}(S^*)) / \text{DIY}$$

ou seja, Δ é uma fração do intervalo

$$(\text{ABS}(S^*), 1.5708)$$

4 - se a resposta for positiva (relevante) faz-se

$$S = (\text{SIN}(S^* + \Delta) + 1) / 2$$

se a resposta for negativa (irrelevante) faz-se

$$S = (\text{SIN}(S^* - \Delta) + 1) / 2$$

4.6 - FUNÇÃO DE BAIXO-INCREMENTO

Conforme já foi visto, esta função tem a finalidade de corrigir probabilidades inicialmente atribuídas a determinados descritores que não permitiam fosse alcançado o valor de equilíbrio.

O processo de realimentação permite que os valores dados aos descritores dos perfis alcancem, eventualmente, um ponto de equilíbrio. Observa-se que a precisão do perfil aumenta à medida que os valores dos descritores tendem ao ponto referenciado. Assim, quanto maior o número de descritores do perfil que alcançam aquele valor, melhores são os resultados apresentados aos usuários.

Todos os descritores dos usuários que coincidiram com descritores dos documentos, mas para os quais tais documentos não foram selecionados para notificação devido à função de disseminação não ter alcançado o limiar, são incrementados de maneira semelhante àquela de uma resposta positiva (relevante), mas por um valor menor, através do baixo-incremento.

A introdução desta função de baixo-incremento num sistema de DSI está baseada na suposição de que descritores do perfil que aparecem em documento, mesmo não selecionados, possuem algum significado para o leitor.

Observa-se uma situação interessante nos sistemas de DSI: existem descritores do perfil, com valores abaixo do limiar, os quais aparecem frequentemente em documentos notificados. Estes são definidos como "gerais" em relação ao interesse do usuário. Os valores de tais descritores ficam oscilando em torno de baixos valores de equilíbrio, e assim devem permanecer, até que eventualmente respostas relevantes ou irrelevantes venham alterar tal situação.

O baixo-incremento dado aos valores de descritores do perfil são computados da seguinte maneira:

Seja S = valor dum descritor do perfil

S'_4 = valor dum descritor do perfil transposto ao quarto quadrante

S^* = valor do descritor após o incremento

DIV = variável de divisão

$$S'_4 = \arccos(1-2S) \quad (IV.7)$$

$$S^* = (1 - \sin(S - (1.5708 - S)/DIV))/2 \quad (IV.8)$$

As equações (7) e (8) são similares às (2) e (4), mostradas anteriormente. Vê-se que o baixo-incremento, dado a um descritor do perfil, é computado usando a função de incremento-decremento, e aplica-se apenas a respostas positivas e descritores de perfis com valores menores que 0.5.

A única diferença é o valor de DIV que, na função de baixo-incremento, é maior, sendo neste trabalho igual a 16.

CAPÍTULO V

CONCLUSÃO

Um sistema de Disseminação Seletiva de Informações (DSI) é aquele que retém, em unidades de armazenamento, um grupo de usuários, indexados por descritores.

Mais especificamente, um sistema de disseminação seletiva de informação foi aqui definido como:

- . um conjunto de perfis
- . um conjunto de documentos a serem disseminados
- . uma função de disseminação f

O problema, aqui abordado, foi determinar como os indexadores identificariam tais conjuntos com um número não muito grande de informações.

Uma idéia, usada atualmente para este fim, é que, dada uma obra, tira-se do seu título e/ou resumo, um conjunto de palavras-chave, chamadas descritores, que a identificam.

Sobre o armazenamento dos conjuntos citados, parte-se do princípio que, em sistemas de recuperação de informação, a preocupação maior é uma rápida pesquisa nos dados armazenados, sendo as atualizações processadas mais lentamente. Isto, aliado ao fato que se quer recuperar informações em arquivos contidos sô em memórias secundárias (discos magnéticos), levou a adotar uma estrutura em árvore, apontada por Knuth (1975), como a melhor solução. Lefkovitz (1974), apresenta um modelo de organização de arquivos baseados em árvores de múltiplas ramificações, que são as árvores com palavras-chave de comprimento fixo. Utiliza-se esta para o arquivo DESCRITORES, que preenche os primeiros requisitos citados.

Com isto, conseguiu-se uma fácil programação, permitindo otimização de espaços e velocidade de atualização, tornando, assim, o sistema bastante otimizado.

Sobre a função de disseminação, f , ela pode ser:

- . uma das quatro apresentadas no Capítulo IV ou
- . uma função de probabilidade, usando limiar

A função de disseminação, usada no sistema, foi a de probabilidade usando limiar. Isto porque experiências de laboratórios demonstraram a superioridade desta abordagem sobre as demais.

Entretanto, o desempenho específico, no caso de implementação, só poderá ser julgado definitivamente mediante uma análise de desenvolvimento dos valores de probabilidade dos descritores dos diversos perfis, após algum tempo de uso. Tendo isto em vista, um arquivo contendo dados referentes a pesos anteriores e atuais, bem como informações sobre os documentos recuperados, é mantido acumulativamente de tal modo que uma análise estatística possa vir a ser feita. Com base nos resultados desta análise, alterações na função poderão vir a serem feitas, ou novas funções poderão ser desenvolvidas e implementadas, como continuidade deste trabalho.

Um primeiro teste foi feito, aplicando-se a função de disseminação, a uma mesma massa de dados, sucessivamente (Apêndice G), demonstrando os resultados bastante coerência no que se refere a incrementos/decrementos em descritores de documentos considerados relevantes/irrelevantes, coerência esta indicadora do acerto da abordagem adotada.

AGRADECIMENTOS

Este trabalho foi realizado com auxílio financeiro das seguintes instituições, às quais externamos nossos agradecimentos:

- EMPRESA BRASILEIRA DE PESQUISAS AGROPECUÁRIA (EMBRAPA)
- UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE (UFRN)

Agradecemos, também:

- Ao Instituto de Pesquisas Espaciais (INPE);
- Ao professor Carlos Henrique Bôto Gois, pela orientação, incentivo e acompanhamento atencioso de todo o desenvolvimento desta tese;
- Ao mestre Miguel José Bersano, pelo apoio e sugestões oportunas apresentadas no início deste trabalho;
- Aos professores do INPE pelos ensinamentos transmitidos durante toda a realização do curso;
- Aos colegas e funcionários do INPE pela atenção dispensada;
- À Aline Zagallo da Silva pela dedicação na datilografia deste trabalho.

BIBLIOGRAFIA

- BIVONA, W.A.; SPIECER, N. SDI: review of selected systems and a design for Army Technical Libraries. AD-636816, 24-25, Ago., 1966.
- BRANDHORST, W.T.; CUADRA, C.A. Document Retrieval and Dissemination Systems. *Annual Review of Information Science and Technology*, 7: 384, 1972.
- BUSS FILHO, A.C.; TEIXEIRA, I.L.R. *Geração de Séries Numéricas com Dígitos de Controle*. São José dos Campos, INPE, Jan., 1975. (576-LAFE).
- COOK, KENNETH H. A threshold Model of Relevance Decisions. *Information Processing & Management*, 11(5/7): 125-135, 1975.
- DAMMERS, H.F. SDI: Some Economic and Organizational Aspects. *Aslib Proceedings*, 23(10): 517, 1972.
- GLUCHOWICZ, Z. *SDI and Retrospectives Searches Computer Based Information Service from RIT*. Trita-Lib-4023, Set., 1973.
- HOUSMAN, E.M. Survey of Current Systems for Selective Dissemination of Information. *Special Interest Group in SDI*, AD-692792, Jun., 1969.
- KNUTH, D.E. *The Art of Computer Programming*. Vol. 3 Reading, Massachusetts, Addison Wesley Publishing Co., 1975.
- LANCASTER, F.W.; BRANDHORST, W.T. Testing Indexes and Index Language Devices: the ASLIB Cranfield Project. *American Documentation*, 4 Jan., 1964.
- LEFKOVITZ, D. *File Structures for On-line Systems*. New York, Spartan Books, 1969.
- LUHN, H.P. A Business Intelligence System. *IBM Journal*, 314-319, Out., 1958.
- . Selective Dissemination of New Scientific Information with Aid of Electronic Processing Equipment. *American Documentation*, 131-138, Abr., 1961.

MAYNARD, J. *Programação Modular*. Rio de Janeiro, Livros Técnicos e Científicos Editora S.A., 1976.

SAGE, C.R.; ANDERSON, R.R.; FITZWATER, D.R. Adaptive Information Dissemination. *American Documentation*, 16(3): 185-200, Jul., 1965.

SUSAN, S.Y. *The Implementation, Evaluation and Refinement of a Manual SDI Service*. Bull. Med. Libr. Assoc., 61(1), Jan., 1973.

VASCONCELOS, N.M.; BERSANO, M.J. *Uma Sistemática de Armazenamento e Recuperação de Informações Bibliográficas*. Tese de Mestrado. São José dos Campos, INPE, Jun., 1976. (INPE-913-TPT/031).

APÊNDICE A

NOTIFICAÇÃO DO SISTEMA AO USUÁRIO

Seguindo um modelo mais generalizado de notificação constante das seguintes informações:

- . Título e demais identificações da obra
- . Resumo
- . Identificação do leitor
- . Cartão-resposta ao sistema

O nosso serviço de notificação fica apto a fornecer ao leitor tudo que ele necessita para identificar a obra, num interesse por ela. O uso do resumo, nem por todos adotado, facilita o usuário decidir pela relevância/irrelevância do material disseminado pelo sistema.

Esta tarefa é realizada no cartão-resposta ao sistema, onde, à obra correspondente, identificada por um número, ele coloca *I* para as irrelevantes e *R*, caso contrário.

00027

280/1

OXFORD UNIV (ENGLAND) DEPT OF AGRICULTURAL SCIEN
CE[*] RESEARCH AND DEVELOPMENT GROUP (EUROPE),
AND NEW YORK 00510. THE AIM OF THIS PROJECT IS TO ANALY
SE THE INFLUENCE OF ALL RELEVANT CHOICES BETWEEN PROCEDURES
FOR SAMPLING, CLASSIFYING, AND MAPPING SOILS OR SOIL PROPERT
IES, AND FOR EVALUATING THE SOILS INFORMATION. THE RESULTS
ARE EXPECTED ALSO TO BE APPLICABLE TO MAPS, ETC., OF OTHER
TERRAIN PROPERTIES. #

NUMERO DO PERFIL 00027

IORIS RAMALHO CORTEZ

* PREZADO PESQUISADOR, PARA MELHOR SELECAO,
INFORME QUANTO A RELEVANCIA PARA OS DOCUMENTOS
ABAIXO, USANDO OS CODIGOS:

R PARA RELEVANTE

I PARA IRRELEVANTE

280/1 * *

ILMO(A). SR(A).

IORIS RAMALHO CORTEZ.

R. RUBENS C. CAMARGO 400

NATAL

59000 RN

APÊNDICE B

PROGRAMA DO MÓDULO GERAÇÃO

Como vimos, este programa tem por função ler e dividir a Fita Fonte, ou entrada do sistema, contendo informações bibliográficas, descritores e número de identificação.

A divisão serve para colocar em arquivos distintos as informações bibliográficas e os descritores, ambos com o número de identicação da obra.

O arquivo com descritores é usado no próximo módulo, enquando o das informações bibliográficas são no fim, para impressão.

GERACAO

```

BEGIN
    00000000 000:0000:0
    0.0000 IS SEGMENT 00003
    1 00000010 003:0000:1
    00000020 003:0000:1
    00000030 003:0000:1
    DATA IS 0007 LONG
    00000040 003:0000:1
    00000050 003:0000:1
    DATA IS 0007 LONG
    00000060 003:0000:1
    00000070 003:0000:1
    DATA IS 0007 LONG
    00000080 003:0000:1
    00000090 003:0000:1
    00000100 003:0000:1
    00000110 003:0000:1
    00000120 003:0000:1
    00000130 003:0000:1
    00000140 003:0000:1
    00000150 003:0000:1
    00000160 003:0000:1
    00000170 003:0000:1
    00000180 003:0000:1
    00000190 003:0000:1
    00000200 003:0000:1
    00000210 003:0000:1
    00000220 003:0010:2
    00000230 003:0010:4
    2 00000240 003:0012:5
    00000250 003:0016:0
    00000260 003:0016:1
    3 00000270 003:0016:4
    00000280 003:001C:2
    00000290 003:001E:0
    00000300 003:001E:4
    00000310 003:001F:0
    4 00000320 003:001F:3
    00000330 003:0021:4
    00000340 003:0021:5
    00000350 003:0025:1
    5 00000360 003:0026:2
    00000370 003:0027:5
    00000380 003:0028:1
    5 00000390 003:0029:1
    00000400 003:0029:4
    4 00000410 003:002A:5
    00000420 003:002E:5
    00000430 003:0030:3
    00000440 003:0033:3
    00000450 003:0033:4
    00000460 003:0034:2
    00000470 003:003F:2
    3 00000480 003:003F:2

```

FILE

FRONTE(KIND=DISK, MAXRECSIZE=313, BLOCKSIZE=3130,
AREASIZE=313, AREAS= 3),

DESIDEN(KIND=DISK, MAXRECSIZE=57, BLOCKSIZE=570,
AREASIZE=57, AREAS= 5),

ARGDIR(KIND=DISK, MAXRECSIZE=290, BLOCKSIZE=2900,
AREASIZE=290, AREAS=5))

DEFINE

LINPADDUFFERS=

REPLACE FRONTE:PRINTFR(BUFFONTE) BY " " FOR 313 WORDS;

REPLACE DESIDEN:PRINTFR(BUFFDESIDEN) BY " " FOR 57 WORDS;

REPLACE ARGDIR:PRINTFR(BUFFARGDIR) BY " " FOR 290 WORDS; #

PRINTFR FRONTE, DESIDEN, ARGDIR

LOOP=1

INTEGER CONT;

APPAY

 BUFFONTE(0:312);

 BUFFDESIDEN(0:56);

 BUFFARGDIR(0:289);

LINPADDUFFERS

WHILE NOT READ(FRONT, 313, BUFFONTE) DO

 BEGIN

 REPLACE PARO:PARO BY FRONTE FOR 5;

 REPLACE DESIDEN BY FRONTE:FRONTE FOR CONT:302 UNTIL = " ";

 IF NOT TOGGLE THEN

 BEGIN

 WRITE(DESIDEN, 57, BUFFDESIDEN);

 CONT:=+1530;

 LOOP:=TRUE;

 WHILE LOOP DO

 BEGIN

 SCAN FRONTE FOR CONT:CONT UNTIL = " ";

 IF NOT TOGGLE THEN

 IF FRONTE = "21" FOR 3 THEN LOOP:=FALSE

 ELSE BEGIN

 FRONTE := FRONTE+1;

 CONT:=+1

 END

 ELSE LOOP:=FALSE

 END;

 CONT:=+1740 THEN 1740 ELSE CONT = 2;

 FRONTE:=FRONTE+2;

 REPLACE PARO:PARO BY FRONTE FOR CONT:CONT UNTIL = " ";

 IF NOT TOGGLE THEN

 REPLACE PARO BY " " FOR 1 ELSE REPLACE PARO #1 BY " " FOR 1;

 WRITE(ARGDIR, 290, BUFFARGDIR);

 END;

 LINPADDUFFERS

END;
 LOCK(FONTE,*);
 LOCK(DESTINCH,*);
 LOCK(ARQRIP,*)
 END.

2 00000490 003:004A:4
 00000500 003:004B:1
 00000510 003:004C:4
 00000520 003:004E:1
 00000530 003:004F:3
 R,0000(003) IS 0063 LONG

 NUMBER OF FRAMES DETECTED = 0.
 NUMBER OF SEGMENTS = 1. TOTAL SEGMENT SIZE = 120 WORDS. CORE ESTIMATE = 7226 WORDS. STACK ESTIMATE = 14
 PROGRAM SIZE = 54 CARDS, 349 SYNTACTIC ITEMS, 19 DISK SEGMENTS.
 PROGRAM FILE NAME: GEPACAN.
 COMPILATION TIME = 17.405 SECONDS ELAPSED; 0.864 SECONDS PROCESSING; 1.387 SECONDS I/O.
 #####

APÊNDICE C

PROGRAMA DO MÓDULO CADAстра/ATUALIZA

Na folha de cadastro, os usuários contam com três opções:

- INSERÇÃO: refere-se ao período de cadastramento do leitor, i.e., pela primeira vez ele está se comunicando com o sistema.
- ALTERAÇÃO: aqui ou ele quer mudar o endereço ou o valor de probabilidade de um ou mais descritores ou mesmo mudar estes.
- RETIRADA: é a sua opção de ser retirado do sistema, ou apenas descritores.

Obviamente, Cadastra funciona para a primeira opção, enquanto Atualiza tem por função executar as outras duas.

C R I T I C A

```

BEGIN
    00000000 000:0000:0
    8,0000 IS SEGMENT 00003
    1 00000010 003:0000:1
    00000020 003:0000:1
    DATA IS 0007 LONG
    00000030 003:0000:1
    DATA IS 0006 LONG
    00000040 003:0000:1
    00000050 003:0000:1
    DATA IS 0007 LONG
    00000060 003:0000:1
    00000070 003:0000:1
    DATA IS 0007 LONG
    00000080 003:0000:1
    00000090 003:0000:1
    DATA IS 0007 LONG
    00000100 003:0000:1
    DATA IS 0004 LONG
    00000110 003:0000:1
    DATA IS 0005 LONG
    00000120 003:0000:1
    DATA IS 0004 LONG
    00000130 003:0000:1
    00000140 003:0000:1
    00000150 003:0000:1
    00000160 003:0000:1
    00000170 003:0000:1
    00000180 003:0000:1
    00000190 003:0000:1
    00000200 003:0000:1
    00000210 003:0000:1
    00000220 003:0000:1
    00000230 003:0000:1
    00000240 003:0000:1
    00000250 003:0000:1
    00000260 003:0000:1
    00000270 003:0000:1
    00000280 003:0000:1
    00000290 003:0000:1
    00000300 003:0000:1
    00000310 003:0000:1
    00000320 003:0000:1
    00000330 003:0000:1
    00000340 003:0001:3
    00000350 003:0005:3
    00000360 003:0005:3
    00000370 003:0005:3
    CONTIN IS SEGMENT 00005
    2 00000380 005:0000:1
    00000390 005:0002:2
    00000400 005:0004:4
    00000410 005:0007:0
    00000420 005:0009:2

FILE CARTUES(KIND=DISK,MAXRECSIZE=14,BLOCKSIZE=210,
    AREASIZE=15,AREAS=10),
    CARDOUT(KIND=DISK,MAXRECSIZE=14,BLOCKSIZE=210),
    OFSCIN (KIND=DISK,MAXRECSIZE=10,BLOCKSIZE=210,
    AREASIZE=21,AREAS=10),
    OFSCINT(KIND=DISK,MAXRECSIZE=10,BLOCKSIZE=210,
    AREASIZE=21,AREAS=10),
    HSUIN (KIND=DISK,MAXRECSIZE=70,BLOCKSIZE=60,
    AREASIZE=3,AREAS=4),
    RD (KIND=READER, MAXRECSIZE=14),
    LP2(KIND=PRINTER),
    LP(KIND=PRINTER);

ARRAY IN(0:13),BUFOFSCIN(0:9),BUFUSUIN(0:19),AR(0:10);
POINTER P1,PCN,PRN,PAR,PRD,PIW;
INTEGER CARD2;
BOOLEAN TIPD,TAGDESC;
LABEL END1, END2;
DEFINE LINPAUFUSUIN = REPLACE POINTER(BUFUSUIN) BY * * FOR
    20 WORDS #,
    MUVECARD1 =
    BEGIN
        REPLACE PRN:PRN+BUFUSUIN BY P1+1 FOR 5; XND, NO PERFIL
        REPLACE PUL+112 BY P1+6 FOR 1; STIPO TRANSACAO
        REPLACE PRN:PRN BY P1+8 FOR 72; * NO, INSTITUICAO E RUA
        END #,
        LINPAUESCIN = REPLACE POINTER(BUFUESCIN) BY * * FOR
            10 WORDS #,
        LIMPAIN = REPLACE P1 BY * * FOR 14 WORDS #,
        EVITA3 = BEGIN
            REPLACE PAR BY P1 + 1 FOR 5;
            TAGDESC:=TRUE;
        END #;
    SWITCH FURPAT FT1=<X4*~X2~NUMERO DO PERFIL CODIFICADO ERRADO"/>,
        <X5*~X2~CODIGO DE TRANSACAO ERRADO"/>;
MODULE PROCEDURE COUTCO;
BEGIN
    INTEGER COUT1,COUT2,COUT3,COUT4,COUT5,Y;
    COUT1:=INTEGER(P1+1,1);
    COUT2:=INTEGER(P1+2,1);
    COUT3:=INTEGER(P1+3,1);
    COUT4:=INTEGER(P1+4,1);
    COUT5:=INTEGER(P1+5,1);

```

C.2

```

CONTIG:=IF Y=((C061+5 + C062+4 + C063+3 + C064+2 + C065) MOD 11) =0
THEN TRUE ELSE FALSE)
END DA CODIGO )

```

```

RMNLEAH PROCEDURE NUMERO)
RFGIN
  TRUTHSET/DIGITO(* 0123456789*)

```

```

  INTEGER I;
  SCAN P1+8 FOR I:5 WHILE IN DIGITO;
  NUMERO:= TOGGLE;
END DA NUMERO;

```

```

RMNLEAH PROCEDURE SIGLA)
RFGIN
  VALUE ARRAY SIGLA(*AC *AL *AM *
    *AP *AR *AS *AT *AU *AV *
    *BA *BB *BC *BD *BE *BF *BG *BH *BI *BJ *BK *BL *
    *BM *BN *BO *BP *BQ *BR *BS *BT *BU *BV *BW *BX *BY *
    *CA *CB *CC *CD *CE *CF *CG *CH *CI *CJ *CK *CL *
    *CM *CN *CO *CP *CQ *CR *CS *CT *CU *CV *CW *CX *CY *
    *DA *DB *DC *DD *DE *DF *DG *DH *DI *DJ *DK *DL *
    *DM *DN *DO *DP *DQ *DR *DS *DT *DU *DV *DW *DX *DY *
    *EA *EB *EC *ED *EE *EF *EG *EH *EI *EJ *EK *EL *
    *EM *EN *EO *EP *EQ *ER *ES *ET *EU *EV *EW *EX *EY *
    *FA *FB *FC *FD *FE *FF *FG *FH *FI *FJ *FK *FL *
    *FM *FN *FO *FP *FQ *FR *FS *FT *FU *FV *FW *FX *FY *
    *GA *GB *GC *GD *GE *GF *GG *GH *GI *GJ *GK *GL *
    *GM *GN *GO *GP *GQ *GR *GS *GT *GU *GV *GW *GX *GY *
    *HA *HB *HC *HD *HE *HF *HG *HH *HI *HJ *HK *HL *
    *HM *HN *HO *HP *HQ *HR *HS *HT *HU *HV *HW *HX *HY *
    *IA *IB *IC *ID *IE *IF *IG *IH *II *IJ *IK *IL *
    *IM *IN *IO *IP *IQ *IR *IS *IT *IU *IV *IW *IX *IY *
    *JA *JB *JC *JD *JE *JF *JG *JH *JI *JJ *JK *JL *
    *JM *JN *JO *JP *JQ *JR *JS *JT *JU *JV *JW *JX *JY *
    *KA *KB *KC *KD *KE *KF *KG *KH *KI *KJ *KK *KL *
    *KM *KN *KO *KP *KQ *KR *KS *KT *KU *KV *KW *KX *KY *
    *LA *LB *LC *LD *LE *LF *LG *LH *LI *LJ *LK *LL *
    *LM *LN *LO *LP *LQ *LR *LS *LT *LU *LV *LW *LX *LY *
    *MA *MB *MC *MD *ME *MF *MG *MH *MI *MJ *MK *ML *
    *MM *MN *MO *MP *MQ *MR *MS *MT *MU *MV *MW *MX *MY *
    *NA *NB *NC *ND *NE *NF *NG *NH *NI *NJ *NK *NL *
    *NM *NN *NO *NP *NQ *NR *NS *NT *NU *NV *NW *NX *NY *
    *OA *OB *OC *OD *OE *OF *OG *OH *OI *OJ *OK *OL *
    *OM *ON *OO *OP *OQ *OR *OS *OT *OU *OV *OW *OX *OY *
    *PA *PB *PC *PD *PE *PF *PG *PH *PI *PJ *PK *PL *
    *PM *PN *PO *PP *PQ *PR *PS *PT *PU *PV *PW *PX *PY *
    *QA *QB *QC *QD *QE *QF *QG *QH *QI *QJ *QK *QL *
    *QM *QN *QO *QP *QQ *QR *QS *QT *QU *QV *QW *QX *QY *
    *RA *RB *RC *RD *RE *RF *RG *RH *RI *RJ *RK *RL *
    *RM *RN *RO *RP *RQ *RR *RS *RT *RU *RV *RW *RX *RY *
    *SA *SB *SC *SD *SE *SF *SG *SH *SI *SJ *SK *SL *
    *SM *SN *SO *SP *SQ *SR *SS *ST *SU *SV *SW *SX *SY *
    *TA *TB *TC *TD *TE *TF *TG *TH *TI *TJ *TK *TL *
    *TM *TN *TO *TP *TQ *TR *TS *TT *TU *TV *TW *TX *TY *
    *UA *UB *UC *UD *UE *UF *UG *UH *UI *UJ *UK *UL *
    *UM *UN *UO *UP *UQ *UR *US *UT *UU *UV *UW *UX *UY *
    *VA *VB *VC *VD *VE *VF *VG *VH *VI *VJ *VK *VL *
    *VM *VN *VO *VP *VQ *VR *VS *VT *VU *VV *VW *VX *VY *
    *WA *WB *WC *WD *WE *WF *WG *WH *WI *WJ *WK *WL *
    *WM *WN *WO *WP *WQ *WR *WS *WT *WU *WV *WW *WX *WY *
    *XA *XB *XC *XD *XE *XF *XG *XH *XI *XJ *XK *XL *
    *XM *XN *XO *XP *XQ *XR *XS *XT *XU *XV *XW *XX *XY *
    *YA *YB *YC *YD *YE *YF *YG *YH *YI *YJ *YK *YL *
    *YM *YN *YO *YP *YQ *YR *YS *YT *YU *YV *YW *YX *YY *
    *ZA *ZB *ZC *ZD *ZE *ZF *ZG *ZH *ZI *ZJ *ZK *ZL *
    *ZM *ZN *ZO *ZP *ZQ *ZR *ZS *ZT *ZU *ZV *ZW *ZX *ZY *)

```

```

  INTEGER I;
  SIGLA:=FALSE;
  FOR I:=0 STEP 1 UNTIL 25 DO
    IF P1+33 = POINTER(SIGLA)+(I*6) FOR 2
    THEN
      BEGIN
        I:=27;
        SIGLA:=TRUE;
      END;
    END;
  END DA SIGLA;

```

```

RMNLEAH PROCEDURE CEP)
RFGIN
  TRUTHSET DIGITO(*0123456789*)

```

```

  INTEGER I;
  SCAN P1+35 FOR I:5 WHILE IN DIGITO;
  CEP:= TOGGLE;
END DA CEP;

```

```

RMNLEAH PROCEDURE DATA)
RFGIN
  TRUTHSET DIGITO(* 0123456789*)

```

```

  INTEGER I;
  SCAN P1+41 FOR I:6 WHILE IN DIGITO;
  DATA:= TOGGLE;
END DA DATA;

```

```

RMNLEAH PROCEDURE PESO)
RFGIN

```

```

00000430 005:0008:4
00000440 005:0011:1
00000450 005:0013:2
CONTIGO(005) IS 0015 LONG
2 00000460 003:0005:3
00000470 003:0005:3
00000480 003:0005:3
NUMERO IS SEGMENT 00006
  DATA IS 0018 LONG
  DATA IS 0008 LONG
2 00000490 006:0000:1
00000500 006:0000:1
00000510 006:0003:2
00000520 006:0004:0
NUMERO(006) IS 0005 LONG
2 00000530 003:0005:3
00000540 003:0005:3
00000550 003:0005:3
SIGLA IS SEGMENT 00008
2 00000560 008:0000:1
00000570 008:0000:1
00000580 008:0000:1
00000590 008:0000:1
00000600 008:0000:1
00000610 008:0000:1
00000620 008:0000:1
00000630 009:0000:1
  DATA IS 0014 LONG
00000640 008:0000:1
00000650 008:0000:1
00000660 008:0000:5
00000670 008:0003:1
00000680 008:0006:3
00000690 008:0007:1
3 00000700 008:0007:4
00000710 008:0008:3
00000720 008:0009:1
3 00000730 008:0000:0
SIGLA(009) IS 000F LONG
2 00000740 003:0005:3
00000750 003:0005:3
00000760 003:0005:3
  CEP IS SEGMENT 0000A
  DATA IS 0008 LONG
2 00000770 004:0000:1
00000780 004:0000:1
00000790 004:0003:2
00000800 004:0004:0
  CEP(00A) IS 0005 LONG
2 00000810 003:0005:3
00000820 003:0005:3
00000830 003:0005:3
  DATA IS SEGMENT 0000C
  DATA IS 0008 LONG
2 00000840 00C:0000:1
00000850 00C:0000:1
00000860 00C:0003:2
00000870 00C:0004:0
  DATA(00C) IS 0005 LONG
2 00000880 003:0005:3
00000890 003:0005:3

```

```

TRUTHSET DIGITO("0123456789")

INTEGER I;
SCAN P1+50 FOR I; WHILE IN DIGITO;
PESO:= TOGGLE;
END DA PESO;

PROCEDURE TESTA(L);
INTEGER L;
BEGIN
  LARFL LSIGLA, LCEP, LDESIG, LDATA;

  SWITCH DESV:= LSIGLA, LCEP, LDESIG, LDATA;
  GO TO DESV(L);
  IF NOT NUMERO
    THEN
      WRITE(LP, <X2, " NUMERO DEVE CONSTAR SO DIGITO"/>);
  LSIGLA:
  IF NOT SIGLA
    THEN
      WRITE(LP, <X3, " SIGLA DE ESTADO ERROADA"/>);
  LCEP:
  IF NOT CEP THEN
      WRITE(LP, <X35, " CEP DEVE CONSTAR SO DIGITO"/>);
  LDESIG:
  IF P1+40 = "E" OR P1+40 = "I" THEN
      ELSE
          WRITE(LP, <X40, " ERRO NA DESIGNACAO DO USUARIO"/>);
  LDATA:
  IF ACT DATA THEN
      WRITE(LP, <X41, " ERRO NA DATA"/>);
END;

MODULE PROCEDURE COMPA(A, N);
ARRAY A: N(0);
BEGIN
  POINTCH PA, PB;

  COMPA:= IF PA1=A + 1 < PB1=B + 1 FOR 5
    THEN TRUE
    ELSE
      IF PA1=A + 1 > PB1=B + 1 FOR 5
        THEN FALSE
        ELSE
          IF PA1=A < PB1=B FOR 1
            THEN TRUE
            ELSE FALSE;
END DA COMPA;

MODULE PROCEDURE COMPANESC(A, B);
ARRAY A, B(0);
BEGIN
  POINTER PA, PB;

  COMPANESC:= IF PA1=A < PB1=B FOR 42
    THEN TRUE
    ELSE
      IF PA = PB FOR 42
        THEN IF PA1=A+43 < PB1=B + 43 FOR 5

```

```

00000900 003:0005:3
PESO IS SEGMENT 0000E
DATA IS 0008 LONG
2 00000910 00E:0000:1
00000920 00E:0000:1
00000930 00E:0003:2
00000940 00E:0004:0
PESQ(40E) IS 0005 LONG
2 00000950 003:0005:3
00000960 003:0005:3
00000970 003:0005:3
00000980 003:0005:3
TESTA IS SEGMENT 00010
2 00000990 010:0000:1
00001000 010:0000:1
00001010 010:0006:0
00001020 010:0006:0
00001030 010:0006:4
00001040 010:000C:2
00001050 010:000C:2
00001060 010:000C:2
00001070 010:0000:0
00001080 010:0012:2
00001090 010:0012:2
00001100 010:0013:0
00001110 010:0018:2
00001120 010:0018:2
00001130 010:0010:3
00001140 010:001E:0
DATA IS 001E LONG
00001150 010:0022:2
00001160 010:0022:2
00001170 010:0023:0
00001180 010:0026:3
TESTA(010) IS 0029 LONG
2 00001190 003:0005:3
00001200 003:0005:3
00001210 003:0005:3
00001220 003:0005:3
COMPA IS SEGMENT 00014
2 00001230 014:0030:1
00001240 014:0003:3
00001250 014:0004:4
00001260 014:0005:2
00001270 014:0008:4
00001280 014:0009:5
00001290 014:000A:3
00001300 014:000C:5
00001310 014:0000:5
00001320 014:000F:1
COMPANESC(014) IS 0012 LONG
2 00001330 003:0005:3
00001340 003:0005:3
00001350 003:0005:3
00001360 003:0005:3
COMPANESC IS SEGMENT 00015
2 00001370 015:0000:1
00001380 015:0002:3
00001390 015:0003:4
00001400 015:0004:2
00001410 015:0005:2

```

```

                                THEN TRUE
                                ELSE FALSE
                                ELSE FALSE
FIM;

P1:=PONTER(IN);
PAR:=PONTER(AR);
P2:=PONTER(IN);
WHILE NOT HEAD(RO,14,1) DO
BEGIN
    IF CODIGO
    THEN
        BEGIN
            WRITE(CARTOES,14,IN)
        END
    ELSE
        BEGIN
            WRITE(LP,14,PAR,P1);
            WRITE(LP,FTC(0));
        END
END;

FIM;
LNC(CARTOES,14);
SHORT(CARDOUT,CARTOES,C,CMPA,14); CLOSE(CARTOES, PURGE);
CARDP:=0; P1:=PONTER(IN); PCN:=PONTER(RUFUSIN);
LTPADESCIN;
TAGDESC:=FALSE;
WHILE NOT HEAD(CARDOUT,14,IN) DO
BEGIN
    IF P1*7 = "1" 1 CODIGO = 1 ?
    THEN
        IF CARDP NEU 2
        THEN ORD1 1
        IF P1 = "1"
        THEN
            IF P1 + 0 = "R"
            OR P1 + 4 = "I"
            OR P1 + 6 = "A"
            THEN
                BEGIN
                    BEGIN
                        LIMP(RUFUSIN);
                        MOVE(CARTOES);
                        CARD2:=2;
                        TAGDESC:=FALSE;
                        TIPO:=FALSE;
                    END
                END
            ELSE
                BEGIN
                    MOVE(CARTOES);
                    WRITE(LP,14,PAR,P1);
                    WRITE(LP,FTC(1));
                    CARD2:=2;
                    TIPO:=TRUE;
                END
            END
        ELSE
            BEGIN
                WRITE(LP,14,PAR,P1);
                WRITE(LP,14,ERR DE URGEN, CARTAO 1 ESPERADO);
                IF P1 = "2" THEN TESTA(0)
                ELSE IF P1 = "3" THEN
                    IF NOT PESO THEN
                        WRITE(LP,14,PESO COM ATRIBUICAO);

```

```

00001420 015:0004:1
00001430 015:0004:2
00001440 015:0004:0
00001450 015:0004:2
COMPARESC(015) IS 0010 LONG
2 00001460 003:0005:3
00001470 003:0007:0
00001480 003:0003:3
00001490 003:0004:0
00001500 003:0006:2
2 00001510 003:0006:4
00001520 003:0006:4
00001530 003:0007:2
3 00001540 003:0007:5
00001550 003:0011:2
3 00001560 003:0016:2
00001570 003:0016:2
3 00001580 003:0016:5
00001590 003:0016:2
00001600 003:0027:2
3 00001610 003:0027:2
2 00001620 003:0027:5
00001630 003:0029:2
00001640 003:0031:2
00001650 003:0035:0
00001660 003:0038:3
00001670 003:0039:1
00001680 003:0030:2
2 00001690 003:0030:4
00001700 003:0031:5
00001710 003:0040:3
00001720 003:0041:2
00001730 003:0042:2
00001740 003:0042:5
00001750 003:0044:3
00001760 003:0046:1
00001770 003:0048:4
00001780 003:0048:2
00001790 003:0048:5
3 00001800 003:0050:2
00001810 003:0050:3
4 00001820 003:0054:5
00001830 003:0059:4
00001840 003:0059:2
00001850 003:0059:0
3 00001860 003:0058:0
00001870 003:0058:0
3 00001880 003:0058:3
4 00001890 003:0063:5
00001900 003:0068:2
00001910 003:0074:2
00001920 003:0075:1
00001930 003:0075:5
3 00001940 003:0075:5
00001950 003:0075:5
3 00001960 003:0076:2
00001970 003:0076:2
00001980 003:0083:2
00001990 003:0084:3
00020000 003:0084:2
00020010 003:0088:3

```

```

                                * ERRADA*>);
TAGDESC:=TRUE;
REPLACE PAR:PAR BY P1 + 1 FOR 5;
END
ELSE
    * ESTAVIS LENDO CARTAO DE ONDEM 2
    IF P1 + 1 EOL PCN FOR 5:
        * COMPARA SE NUMERO DO PERFIL COINCIDE
        THEN
            IF NOT T1P0
                * NAO HOUVE ERRO NO PRIMEIRO CARTAO
            THEN
                IF P1 = "2"
                    * VERIFICA ORDEM
                THEN
                    IF NUMERO
                        * VERIFICA NUMERO
                    THEN
                        IF SIGLA
                            THEN
                                IF CEP
                                    THEN
                                        IF P1+00 = "E" OR P1+00 = "I"
                                            THEN
                                                IF DATA
                                                    THEN
                                                        BEGIN
                                                            REPLACE PHA BY P1+8 FOR 40;
                                                            CARD2:=0;
                                                            WRITE(USUIN,20,BUFUSUIN);
                                                            WRITE(LP2,20,BUFUSUIN);
                                                            TAGDESC:=FALSE;
                                                        END
                                                    ELSE
                                                        BEGIN
                                                            WRITE(LP,<X00,>P1);
                                                            WRITE(LP,<X01,>"* ERRO NA DATA*>);
                                                            EVITA3;
                                                            CARD2:=0;
                                                        END
                                                    ELSE
                                                        BEGIN
                                                            WRITE(LP,<X00,>P1);
                                                            WRITE(LP,<X00,>"* ERRO NA DESIGNACAO DO USUARIO
                                                        END
                                                    TESTA(4);
                                                    EVITA3;
                                                    CARD2:=0;
                                                END
                                            ELSE
                                                BEGIN
                                                    WRITE(LP,<X00,>P1);
                                                    WRITE(LP,<X35,>"* CEP DEVE CONSTAR SO DIGITOS*>);
                                                    TESTA(3);
                                                    EVITA3;
                                                    CARD2:=0;
                                                END
                                            ELSE
                                                BEGIN
                                                    WRITE(LP,<X00,>P1);
                                                    WRITE(LP,<X33,>"* SIGLA DE ESTADO ERRADA*>);
                                                    TESTA(2);
                                                    EVITA3;
                                                    CARD2:=0;
                                                END
                                            END
                                        END
                                    END
                                END
                            END
                        END
                    END
                END
            END
        END
    END
END

```

```

00002020 003:008015
00002030 003:009112
00002040 003:009210
00002050 003:009414
3 00002060 003:009414
00002070 003:009414
00002080 003:009614
00002090 003:009712
00002100 003:009715
00002110 003:009811
00002120 003:009911
00002130 003:009413
00002140 003:009810
00002150 003:009814
00002160 003:009811
00002170 003:009815
00002180 003:009812
00002190 003:009810
00002200 003:009211
00002210 003:009314
00002220 003:009411
00002230 003:009415
3 00002240 003:009512
00002250 003:009713
00002260 003:009811
00002270 003:009812
00002280 003:009812
00002290 003:009810
3 00002300 003:009810
00002310 003:009810
3 00002320 003:009813
00002330 003:009812
00002340 003:009812
4 00002350 003:009810
00002360 003:009814
3 00002370 003:009814
00002380 003:009814
3 00002390 003:009811
00002400 003:009812
00002410 003:009811
DATA IS 0043 LONG
00002420 003:009812
00002430 003:009813
4 00002440 003:009811
00002450 003:009815
3 00002460 003:009815
00002470 003:009815
3 00002480 003:009812
00002490 003:009812
00002500 003:009811
00002510 003:009813
4 00002520 003:009811
00002530 003:009815
3 00002540 003:009815
00002550 003:009815
3 00002560 003:009812
00002570 003:009812
00002580 003:009813
00002590 003:009813
4 00002600 003:009811
00002610 003:009815

```

```

ELSE
  BEGIN
    WRITE(LP, <ABO>, />, P1);
    WRITE(LP, <X8>, "NÚMERO DEVE CONSTAR SO DIGITOS"/>);

    TESTA(1);
    EVITA3;
    CARD2:=0;
  END
ELSE
  BEGIN
    WRITE(LP, <ABO>, />, P1);
    WRITE(LP, <"ERRO DE ORDEN, DEVE SER CARTAO 2"/>);
    EVITA3;
    CARD2:=0;
  END
ELSE
  BEGIN
    WRITE(LP, <ABO>, />, P1);
    WRITE(LP, <"CARTAO 2 RECUSADO, CARTAO 1 COM ERRO"/>);

    TESTA(0);
    TIPO:=FALSE;
    EVITA3;
    CARD2:=0;
  END
ELSE
  BEGIN
    IF P1 = "2"
    THEN
      BEGIN
        WRITE(LP, <ABO>, />, PBN);
        WRITE(LP, <ABO>, />, P1);
        WRITE(LP, <"DOIS CARTOES DE ORDEN 2"/>);
        TESTA(0);
        EVITA3;
        CARD2:=0;
      END
    ELSE
      BEGIN
        WRITE(LP, <ABO>, />, PCN);
        WRITE(LP, <"FALTA CARTAO 2 /TESTE"/>);
        IF P1 = "1"
        THEN
          GO ORD1
        ELSE
          BEGIN
            WRITE(LP, <ABO>, />, P1);
            WRITE(LP, <"CARTAO IGNORADO"/>);
          END
        END
      END
    END
  END
  FLSE: X CODIGO EM 2 (DESCRITOR)
  IF CARD2 NEQ 2
  THEN
    IF "3" TAGDESC
    THEN COD2 :
    IF P1 = "2"
    THEN
      IF P1 = "3"
      THEN

```

```

3 00002620 003:00F0:5
00002630 003:00F0:5
3 00002640 003:00FE:2
00002650 003:0106:2
DATA IS 0029 LONG
00002660 003:0108:2
00002670 003:010U:2
4 00002680 003:0110:0
00002690 003:0110:4
3 00002700 003:0110:4
00002710 003:0110:4
3 00002720 003:0111:1
00002730 003:0118:2
00002740 003:0110:2
4 00002750 003:0120:0
00002760 003:0120:4
3 00002770 003:0120:4
00002780 003:0120:4
3 00002790 003:0121:1
00002800 003:0128:2
DATA IS 0010 LONG
00002810 003:0120:2
00002820 003:012F:2
00002830 003:0130:0
4 00002840 003:0132:4
00002850 003:0133:2
3 00002860 003:0133:2
00002870 003:0133:2
3 00002880 003:0133:5
00002890 003:0134:2
00002900 003:0135:4
4 00002910 003:0136:1
00002920 003:0130:2
00002930 003:0145:2
00002940 003:0144:2
00002950 003:014C:2
5 00002960 003:014F:0
00002970 003:014F:4
4 00002980 003:014F:4
00002990 003:014F:4
4 00003000 003:0150:1
00003010 003:0157:2
00003020 003:015C:2
00003030 003:015C:5
00003040 003:015E:3
00003050 003:015F:0
00003060 003:015F:0
5 00003070 003:015F:0
00003080 003:0166:2
00003090 003:0168:2
5 00003100 003:0168:2
4 00003110 003:0168:2
3 00003120 003:0168:2
00003130 003:0168:2
00003140 003:016C:1
00003150 003:016C:4
00003160 003:016U:1
00003170 003:016E:0
00003180 003:016F:1
00003190 003:0170:3
00003200 003:0171:3

```

```

IF P1+6 = "R"
CR P1+6 = "I"
CR P1+6 = "A"
THEN

```

```

IF PESO
THEN

```

```

BEGIN

```

```

- ARRAY CONVERTER(0:1),

```

```

INTEGER I;

```

```

REPLACE PD:PRD:=RUFDESCIN

```

```

BY P1*4 FOR 42;

```

```

REPLACE PD:PRD BY P1+1 FOR 5;

```

```

REPLACE POINTER(CONVERTER) BY

```

```

P1 + 50 FOR 6;

```

```

REPLACE POINTER(CONVERTER(1))

```

```

BY "R" FOR 1;

```

```

READ(CONVERTER, /, RUFDESCIN(8));

```

```

REPLACE PD:PRD + 7 BY P1 + 6

```

```

FOR 1;

```

```

WRITE(LP2, <R46,F7,5>, 16);

```

```

FOR I=0 STEP 1 UNTIL 9 DO

```

```

RUFDESCIN(I));

```

```

WRITE(DFSCIN, 10, RUFDESCIN);

```

```

LIMPAUESCIN;

```

```

END

```

```

ELSE

```

```

BEGIN

```

```

WRITE(LP, <A80, />, P1);

```

```

WRITE(LP, <

```

```

" PESO COM ATRIBUICAO INCORRETA" /

```

```

/>);

```

```

END

```

```

ELSE

```

```

BEGIN

```

```

WRITE(LP, <A80, />, P1);

```

```

WRITE(LP, FT(1));

```

```

IF NOT PESO THEN

```

```

WRITE(LP, <" PESO COM ATRIBUICAO ERRA

```

```

END

```

```

ELSE

```

```

BEGIN

```

```

WRITE(LP, <A80, />, P1);

```

```

WRITE(LP, <"ERRO NO NUMERO DE ORDEN, SO 1,2,3"

```

```

/>);

```

```

END

```

```

ELSE

```

```

BEGIN

```

```

WRITE(LP, <A80, />, P1);

```

```

WRITE(LP, <"ERRO NO CODIGO" SO 1,2"/>);

```

```

END

```

```

ELSE

```

```

IF P1 + 1 = PAR FOR 5

```

```

THEN

```

```

BEGIN

```

```

WRITE(LP, <A80, />, P1);

```

```

WRITE(LP, <"RECUASADO POR ERRO NOS CARTOES 1,2"

```

```

/>);

```

```

00003210 003:0172:5

```

```

00003220 003:0174:3

```

```

00003230 003:0177:0

```

```

00003240 003:0179:4

```

```

00003250 003:0178:1

```

```

00003260 003:0178:4

```

```

00003270 003:017C:2

```

```

3 00003280 003:017C:5

```

```

R,0001 IS SEGMENT 00019

```

```

00003290 019:0009:1

```

```

00003300 019:0000:1

```

```

00003310 019:0000:1

```

```

00003320 019:0003:4

```

```

00003330 019:0005:5

```

```

00003340 019:0006:5

```

```

00003350 019:0008:3

```

```

00003360 019:0008:4

```

```

00003370 019:0008:2

```

```

00003380 019:0019:3

```

```

00003390 019:0018:1

```

```

00003400 019:001C:5

```

```

00003410 019:001F:4

```

```

00003420 019:0023:1

```

```

00003430 019:002C:2

```

```

00003440 019:0032:2

```

```

00003450 019:0035:3

```

```

3 00003460 019:0035:3

```

```

R,0001(019) IS 003E LONG

```

```

00003470 003:0170:3

```

```

3 00003480 003:017E:0

```

```

00003490 003:0185:2

```

```

00003500 003:0186:3

```

```

00003510 003:0187:1

```

```

DATA IS 003A LONG

```

```

00003520 003:0184:2

```

```

3 00003530 003:0184:2

```

```

00003540 003:0184:2

```

```

3 00003550 003:0184:5

```

```

00003560 003:0192:2

```

```

00003570 003:0198:2

```

```

00003580 003:019C:0

```

```

00003590 003:019E:2

```

```

00003600 003:019F:3

```

```

3 00003610 003:01A1:2

```

```

00003620 003:01A1:2

```

```

3 00003630 003:01A1:5

```

```

00003640 003:01A9:2

```

```

00003650 003:01A8:1

```

```

00003660 003:01A8:2

```

```

3 00003670 003:01A8:2

```

```

00003680 003:01A8:2

```

```

3 00003690 003:01A8:5

```

```

00003700 003:01B6:2

```

```

00003710 003:01B8:2

```

```

3 00003720 003:01B8:2

```

```

00003730 003:01B8:2

```

```

00003740 003:01B0:2

```

```

00003750 003:01B8:0

```

```

3 00003760 003:01B8:3

```

```

00003770 003:01C6:2

```

```

00003780 003:01C6:1

```

```

      END
    ELSE
      BEGIN
        TAGDESCI=FALSE;
        GO COD2;
      END
    ELSE
      BEGIN
        WRITE(LP, <ABO, />, PCN);
        WRITE(LP, <"FALTA CARTA 2 DEST" />);
        WRITE(LP, <ABO, />, P1);
        WRITE(LP, <"RECUSADO, FORA DE ORDEM" />);
        EVITA3;
        CARD2:=C1
      END;
    END;
  LIMPATM;
  END ON WHILE;
  LOCK(DESCINT, *);
  SORT(DESCIN, DESCINT, C, COMPADESC, 10);
  CLOSE(DESCINT, PURGE);
  CLOSE(CARDINT, PURGE);
  LOCK(USUIN, *); LOCK(DESCIN, *);
  WRITE(LP, <///, X10, "CASC OS DADOS ESTEJAM CERTOS FAÇA EXECUTAR OS PROGRA
  MAS", /, X10, "DESCRIÇÕES E -USUÁRIOS- PARA FAZEM AS ALTERAÇÕES",
  /, X10, "NOS ARQUIVOS -DESCIN- E -USU- RESPECTIVAMENTE">);
  END.

```

```

      DATA IS 0034 LONG
00003790 003:01C8:2
3 00003800 003:01C8:2
00003810 003:01C8:2
3 00003820 003:01C8:5
00003830 003:01CC:3
00003840 003:01C0:0
3 00003850 003:01C0:0
00003860 003:01C0:0
3 00003870 003:01C0:3
00003880 003:0105:2
00003890 003:010A:2
00003900 003:01E2:2
00003910 003:01E7:2
4 00003920 003:01EA:0
00003930 003:01EA:4
3 00003940 003:01EA:4
00003950 003:01ED:3
2 00003960 003:01EE:0
00003970 003:01EF:3
00003980 003:01F5:5
00003990 003:01F7:2
00004000 003:01F8:5
00004010 003:01F8:5
00004020 003:01F0:4
00004030 003:01F0:4
      DATA IS 0022 LONG
00004040 003:01FE:4
      DATA IS 0202 LONG
      STACKCODE IS SEGMENT 0001E
      STACKCODE(01E) IS 006F LONG
      DATA IS 0027 LONG

```

```

*****
NUMBER OF ERRORS DETECTED = 3.
NUMBER OF SEGMENTS = 28, TOTAL SEGMENT SIZE = 1300 WORDS, CODE ESTIMATE = 14726 WORDS, STACK ESTIMATE = 73
PROGRAM SIZE = 405 CARDS, 1851 SYNTACTIC ITEMS, 121 DISK SEGMENTS.
PROGRAM FILE NAME: CHITICA.
COMPILATION TIME = 89.940 SECONDS ELAPSED; 4.706 SECONDS PROCESSING; 10.199 SECONDS I/O.
*****

```


00000470	003:0021:1
00000480	003:002A:1
00000490	003:002A:1
00000500	003:002A:3
2 00000510	003:002B:0
00000520	003:002B:4
00000530	003:002C:1
2 00000540	003:002C:1
00000550	003:002C:5
00000560	003:002F:1
00000570	003:002F:1
00000580	003:0031:2
00000590	003:0032:0
2 00000600	003:0032:3
00000610	003:0035:2
00000620	003:003A:2
00000630	003:003C:0
00000640	003:003F:3
00000650	003:0041:0
00000660	003:0044:3
2 00000670	003:0044:3
00000680	003:0044:3
2 00000690	003:0045:0
00000700	003:0046:1
00000710	003:004B:4
00000720	003:004A:1
3 00000730	003:004A:2
00000740	003:004D:3
00000750	003:0050:3
3 00000760	003:0050:3
00000770	003:0050:3
00000780	003:0052:1
00000790	003:0053:3
3 00000800	003:0054:0
00000810	003:0057:3
00000820	003:0059:0
00000830	003:005A:4
00000840	003:0060:2
00000850	003:0063:3
00000860	003:0066:3
3 00000870	003:0066:3
00000880	003:0066:3
00000890	003:0068:1
00000900	003:0069:3
3 00000910	003:006A:0
00000920	003:006D:3
00000930	003:006F:0
00000940	003:0070:4
00000950	003:0076:2
00000960	002:0079:3
00000970	003:007C:3
3 00000980	003:007C:3
00000990	003:007C:3
3 00001000	003:007D:0
00001010	003:0083:2
4 00001020	003:0084:2
00001030	003:008C:3
00001040	003:008F:3
3 00001050	003:008F:3
2 00001060	003:008F:3
00001070	003:008F:3

```

WRITE(DESCENT,9,REG1)
LIMPA REG1
REPLACE P1 BY P2 FOR 351
LIMPA REG2
PRIMEIRO:=FALSE

```

```

END;
GO L1;
EOPD: IF P1 NEQ " " FOR 42
  THEN
    BEGIN
      REPLACE P1 + 54 BY " " FOR 11
      WRITE(DESCENT,9,REG1)
    END;

```

```

CLOSE(DESCRGE,PURGE)
LOCK(DESCENT,*)
TERMINADA PRIMEIRA FASE DE MODIFICACAO DOS ARQUIVOS
"USU" E "DESC". PASSAMOS A MONTAR A B-ARVORE A PAR-
TIR DO ARQUIVO "DESCENT", GERANDO O NOVO "DESC" NA
SUA FORMA FINAL. A CIA MENCIONADA,
O ARQUIVO "USU" FICARA NA FORMA SEQUENCIAL-INDEXADA

```

```
END,
```

```

2 00001080 003:0090:0
00001090 003:0096:2
00001100 003:0099:3
00001110 003:0098:0
00001120 003:009E:3
00001130 003:009F:1
2 00001140 003:009F:1
00001150 003:009F:4
00001160 003:00A2:0
00001170 003:00A2:4
2 00001180 003:00A3:1
00001190 003:00A6:2
00001200 003:00AC:2
2 00001210 003:00AC:2
00001220 003:00AD:5
00001230 OMIT
00001240 OMIT
00001250 OMIT
00001260 OMIT
00001270 OMIT
00001280 003:00AF:2
0,0000(003) IS 000E LONG

```

```

=====
NUMBER OF FRAMES DETECTED = 0,
NUMBER OF SEGMENTS = 2, TOTAL SEGMENT SIZE = 226 WORDS, CORE ESTIMATE = 13574 WORDS, STACK ESTIMATE = 19
PROGRAM SIZE = 129 CARDS(5 OMITTED CARDS), 619 SYNTACTIC ITEMS, 29 DISK SEGMENTS,
PROGRAM FILE NAME: DESCRITURES,
COMPIATION TIME = 21.672 SECONDS ELAPSED; 1.394 SECONDS PROCESSING; 2.937 SECONDS I/O.
=====

```

U S U A R I O S

```

PERIN
      00000000 000:0000:0
      0.0000 IS SEGMENT 00003
      1 00000010 003:0000:1
      00000020 003:0000:1
      00000030 003:0000:1
      DATA IS 0007 LONG
      00000040 003:0000:1
      00000050 003:0000:1
      00000060 003:0000:1
      DATA IS 0008 LONG
      00000070 003:0000:1
      DATA IS 0006 LONG
      00000080 003:0000:1
      DATA IS 0004 LONG
      00000090 003:0000:1
      00000100 003:0000:1
      DATA IS 0007 LONG
      00000110 003:0000:1
      00000120 003:0000:1
      DATA IS 0008 LONG
      00000130 003:0000:1
      00000140 003:0002:4
      00000150 003:0002:4
      00000160 003:0002:4
      00000170 003:0002:4
      00000180 003:0002:4
      00000190 003:0002:4
      00000200 003:0002:4
      00000210 003:0002:4
      00000220 003:0002:4
      00000230 003:0002:4
      00000240 003:0002:4
      00000250 003:0002:4
      ORGHRGE IS SEGMENT 00004
      2 00000260 004:0000:1
      00000270 004:0003:4
      ORGHRGE(004) IS 0006 LONG
      2 00000280 003:0002:4
      00000290 003:0004:1
      00000300 003:0005:4
      00000310 003:0008:3
      00000320 003:0000:0
      00000330 003:0008:3
      00000340 003:0010:0
      00000350 003:0011:3
      00000360 003:0014:3
      00000370 003:0016:2
      00000380 003:0017:0
      00000390 003:0017:0
      00000400 003:0017:2
      2 00000410 003:0017:5
      00000420 003:0020:1
      00000430 003:0020:1

X PROGRAMA MODIFICACAO PARA ARQUIVO "USU"
FILE USUIN(KIND=DISK, MAXRECSIZE=20, BLOCKSIZE=60,
      AREASIZE=03, AREAS=4),
      USUVELHO(KIND=DISK, MAXRECSIZE=20, BLOCKSIZE=60,
      AREASIZE=03, AREAS=4, TITLE="USU.",
      OPTIONAL=TRUE),
      USURGE(KIND=DISK, MAXRECSIZE=20, BLOCKSIZE=60),
      CT(KIND=PRINTER),
      USUCNO(KIND=DISK, MAXRECSIZE=41, BLOCKSIZE=246,
      AREASIZE=12, AREAS=1),
      USUNOVO(KIND=DISK, MAXRECSIZE=20, BLOCKSIZE=60,
      AREASIZE=03, AREAS=4, TITLE="USU.N"),
      ARRAY REG1, REG2 (0:19), ATR(0:40)
OFFINE
      MOVUSUCNO= WRITE(USUCNO,41,ATR)14,
      LIMPAREG1 = REPLACE POINTER(REG1) BY " " FOR 20 WORDS #,
      LIMPAREG2 = REPLACE POINTER(REG2) BY " " FOR 20 WORDS #)
PRINTER P1, P2;
INTERSECT, K;
DOFILEN PRIMEIRO;
LABEL L1, ENF;
DOFILEN PROCEDURE ORGHRGE(4,8);
ARRAY A, ATR(1);
BEGIN
      PRINTER PA, PB;

      ORGHRGE:= PA:= A (:= PB:= 8 FOR 5)
      END DA ORGHRGE;

LOCK(USUIN,1); X SE AJINDA NAO "LOCKED"
LOCK(USUVELHO,1); IDEN IDEN
MFRGE(USURGE,ORGHRGE,20,USUIN=USUVELHO);
CLOSE(USUIN,PURGE);
CLOSE(USUVELHO,PURGE);
P1:=PRINTER(REG1);
P2:=PRINTER(REG2);
REPLACE POINTER(ATR) BY " " FOR 41 WORDS;
ATR(0),ATR(1)=0;
PRIMEIRO:=TRUE;
L1 IF PRIMEIRO
THEN
      BEGIN
      READ(USURGE=20,REG1)(EOF); XEN ENF TESTAR REG, EM REG1
      PRIMEIRO:=FALSE
      END

```

U S U A R I O S

```

PERIN 00000000 000:0000:0
      8.0000 IS SEGMENT 00003
X PROGRAMA MODIFICACAO PARA ARQUIVO "USU" 1 00000010 003:0000:1
FILE USUIN(KIND=DISK, MAXRECSIZE=20, BLOCKSIZE=60,
      00000020 003:0000:1
      AREASIZE=03, AREAS=4),
      00000030 003:0000:1
      DATA IS 0007 LONG
      00000040 003:0000:1
      00000050 003:0000:1
      00000060 003:0000:1
      DATA IS 0008 LONG
      00000070 003:0000:1
      DATA IS 0006 LONG
      00000080 003:0000:1
      DATA IS 0004 LONG
      00000090 003:0000:1
      00000100 003:0000:1
      DATA IS 0007 LONG
      00000110 003:0000:1
      00000120 003:0000:1
      DATA IS 0008 LONG
      00000130 003:0000:1
      00000140 003:0002:4
      00000150 003:0002:4
      00000160 003:0002:4
      00000170 003:0002:4
      00000180 003:0002:4
      00000190 003:0002:4
      00000200 003:0002:4
      00000210 003:0002:4
      00000220 003:0002:4
      00000230 003:0002:4
      00000240 003:0002:4
      00000250 003:0002:4
      ORGHRGE IS SEGMENT 00004
      2 00000260 004:0000:1
      00000270 004:0003:4
      ORGHRGE(1004) IS 0006 LONG
      2 00000280 003:0002:4
      00000290 003:0004:1
      00000300 003:0005:4
      00000310 003:0008:3
      00000320 003:0000:0
      00000330 003:0000:3
      00000340 003:0010:0
      00000350 003:0011:3
      00000360 003:0014:3
      00000370 003:0016:2
      00000380 003:0017:0
      00000390 003:0017:0
      00000400 003:0017:2
      00000410 003:0017:5
      00000420 003:0020:1
      00000430 003:0020:1

USUVELHO(KIND=DISK, MAXRECSIZE=20, BLOCKSIZE=60,
      AREASIZE=03, AREAS=4, TITLE="USU.",
      OPTIONAL=TRUE),

USURGE(KIND=DISK, MAXRECSIZE=20, BLOCKSIZE=60),

CT(KIND=PRINTER),

USUCNO(KIND=DISK, MAXRECSIZE=41, BLOCKSIZE=246,
      AREASIZE=12, AREAS=1),

USUNOVN(KIND=DISK, MAXRECSIZE=20, BLOCKSIZE=60,
      AREASIZE=03, AREAS=4, TITLE="USU.N"),

ARRAY REG1, REG2 (0:19), ATR(0:40)
OFFINE
  NOVINSICOD= WRITE(USUCNO,41,ATR)14,
  LIMPAREG1 = REPLACE POINTER(REG1) BY " " FOR 20 WORDS #,
  LIMPAREG2 = REPLACE POINTER(REG2) BY " " FOR 20 WORDS #)
PRINTER P1, P2;
  INTER 1, K;
  PRIMEIRO;
  LABEL L1, ENF;
  PRIMEIRO PROCEDURE ORGHRGE(4,R);
  ARRAY A,R(0);
  BEGIN
    PRINTER PA, PB;

    ORGHRGE:= PA# A # PB# B FOR 5;
  END DA ORGHRGE;

LOCK(USUIN,*) : 1 SE AJINDA NAO "LOCKED"
LOCK(USUVELHO,*) : IDEN IDEN
MFRGE(USUHRGE,ORGHRGE,20,USUIN,USUVELHO);
  CLOSE(USUIN,PURGE);
  CLOSE(USUVELHO,PURGE);
P1:=PRINTER(REG1);
P2:=PRINTER(REG2);
REPLACE POINTER(ATR) BY " " FOR 41 WORDS;
ATR(0),776611=0;
PRIMEIRO:=TRUE;
L1 IF PRIMEIRO
  THEN
    BEGIN
      READ(USUHRGE+20,REG1)(EOF); 1EM ENF TESTAR REG. EM REG1
      PRIMEIRO:=FALSE
    END
  END

```

```

WRITE(C1,20,REG1);
WRITE(C1,20,REG2);
LIMPARC1;
LIMPARC2;
END;
GO L1;
END; IF P1 NEW * * FOR 5
THEN
  BEGIN
    REPLACE P1 + 117 BY * * FOR 1;
    USUCOD
    WRITE(USUCOD,20,REG1);
  END;
  CLOSE(USUCOD,PURGE);
  LOCK(USUCOD,);
  LOCK(USUCOD,);
END.

```

```

00001040 003:000F:2
00001050 003:0005:2
00001060 003:0008:2
00001070 003:000E:3
00001080 003:0001:3
2 00001090 003:0001:3
00001100 003:0002:0
00001110 003:0004:0
00001120 003:0004:4
2 00001130 003:0005:1
00001140 003:0008:2
00001150 003:000E:2
00001160 003:0004:2
2 00001170 003:0004:2
00001180 003:0005:5
00001190 003:0007:2
00001200 003:0008:5
R,0000(003) IS 00FF LONG
DATA IS 0009 LONG

```

```

#####
NUMBER OF ERRORS DETECTED = 0.
NUMBER OF SEGMENTS = 5, TOTAL SEGMENT SIZE = 335 WORDS, CORE ESTIMATE = 15356 WORDS, STACK ESTIMATE = 25
PROGRAM SIZE = 121 CAMDS, 690 SYNTACTIC ITEMS, 37 DISK SEGMENTS.
PROGRAM FILE NAME: USUCOD.
COMPILED TIME = 143.879 SECONDS ELAPSED; 1.988 SECONDS PROCESSING; 2.724 SECONDS I/O.
#####

```

MONTA / U E S C

REGIM

FILE DESCENT(KIND=DISK, MAXRECSIZE=9, BLOCKSIZE=270,
AREASIZE=30, AREAS=8),

DESCNEW(KIND=DISK, MAXRECSIZE=56, BLOCKSIZE=2128,
AREASIZE=2128, AREAS=1,
TITLE="DESC."),

USUCON (KIND=DISK, MAXRECSIZE=41, BLOCKSIZE=246,
AREASIZE=12, AREAS=1),

LP (KIND=PRINTER),

ARRAY ATR, ATHAUX(C14C),

ATR1(0:5),

ATR2(0:8),

ATR3(0:28),

R, H(0:9),

POINTER PAT0, PAT01, PAT02, PAT03, PATHAUX, PDESCRITOR, PSIMUSU, PCON3,
PAUX;

INTEGER J, K, X, L, V, P,

DISPNIVEL, GUARDEND, DISP, CONTA, IND3;

DEFINE

NUMDESCN(I)=ATR1(A+1), (A7:20),

ENDDESCN(I)=ATR1(A+1), (27:22),

DESCN(I)=ATR1(A+1), (5: 6),

PSN(I)=ATR(2+1+1), (47:14),

CAMP0100USU(I)=ATR(2+1+1), (47:14),

CAMP0200USU(I)=ATR(2+1+1), (29: 3),

CAMP0300USU(I)=ATR(2+1+1), (26:21),

CAMP0400USU(I)=ATR(2+1+1), (5: 6),

CAMP05AUX(I)=ATR(11X(2+1+1), (26:21),

CAMP06AUX(I)=ATR(11X(2+1+1), (5: 6),

ACUM = ATR(0), (7:8),

ACUMPAUX = ATR(11X(0), (7:8),

PSN(I)=ATR(8+2+1),

IF (H(1+1) NEQ 0) AND (R(L) NEQ (P+1)) THEN 0,

GUARDUSU =

HFGIN

REPLACE PCINTER(ATHAUX) BY PCINTER(ATR) FOR 41 WORDS;

GUARDEND=DISP

DISP:=0

ENDIS

LIMPAATR=

BEGIN

REPLACE PCINTER(ATR) BY " " FOR 51

ACUM:=0;

FOR X=1 STEP 1 UNTIL 19 DO

BEGIN

CAMP0100LS(X)=0;

CAMP0200LS(X)=0;

00000000 00000000

8.0000 IS SEGMENT 00003

1 00000010 003:0000:1

00000020 003:0000:1

DATA IS 0007 LONG

00000030 003:0000:1

00000040 003:0000:1

00000050 003:0000:1

DATA IS 0008 LONG

00000060 003:0000:1

00000070 003:0000:1

DATA IS 0007 LONG

00000080 003:0000:1

DATA IS 0004 LONG

00000090 003:0000:1

00000100 003:0002:4

00000110 003:0002:4

00000120 003:0002:4

00000130 003:0002:4

00000140 003:0005:1

00000150 003:0005:1

00000160 003:0005:1

00000170 003:0005:1

00000180 003:0005:1

00000190 003:0005:1

00000200 003:0005:1

00000210 003:0005:1

00000220 003:0005:1

00000230 003:0005:1

00000240 003:0005:1

00000250 003:0005:1

00000260 003:0005:1

00000270 003:0005:1

00000280 003:0005:1

00000290 003:0005:1

00000300 003:0005:1

00000310 003:0005:1

00000320 003:0005:1

00000330 003:0005:1

00000340 003:0005:1

00000350 003:0005:1

00000360 003:0005:1

00000370 003:0005:1

00000380 003:0005:1

00000390 003:0005:1

00000400 003:0005:1

00000410 003:0005:1

00000420 003:0005:1

00000430 003:0005:1

00000440 003:0005:1

00000450 003:0005:1

00000460 003:0005:1

00000470 003:0005:1

```

        CAMPO3UCUSL(X)=0;
        CAMPO4UCUSL(X)=0;
        PESO(X)=0;
    END
    END; #
    LIMPAAH2=
        REPLACE POINTER(ATB2) BY * * FOR 9 WORDS;
    LIMPAAH3=
        REPLACE POINTER(ATB3) BY * * FOR 29 WORDS;
    MOVESCHITR=
        BEGIN
            LIMPAAH3;
            REPLACE PATB3:PTRB3=POINTER(ATB3) BY PATB2 FOR 40;
            PESO[CONTA]=ATH2[R];
            CONTA:=++1;
            LIMPAAH2
        END; #
    INSETHAISUM =
        BEGIN
            REPLACE PATB3:PTRB3 + 6 BY PATB2 + 42 FOR 5;
            PATH3:=PATB3 + 1;
            PESO[CONTA]=ATH2[R];
            CONTA:=++1;
            LIMPAAH2
        END; #
    PROCURASU =
        BEGIN
            INTEGER I;
            WHILE PESO[CONTA] + (6*I+1) NEO PAUX FOR 5 DO
                BEGIN
                    I:=I+1;
                    OJSP:=++1
                END
            END; #
    LEUSU =
        BEGIN
            LIMPAAH3;
            READ(USLCOC[ISPI], 41, ATB);
            IF ACUM = C THEN
                REPLACE POINTER(ATB) BY PAUX FOR 5;
            CAMPO1UCUSL(ACUM)=C[L];
            CAMPO2UCUSL(ACUM)=R[L];
            PESO(ACUM)=PTRB3[INO3]
        END; #
    MOVESCHITRPARAF(LHA=
        BEGIN
            REPLACE PATH1 BY PUESCHITR FOR 42;
            IF P[L] NEC (P-1) THEN
                PATH1:=PATH1 + 48
            END; #
    PROCURASUARIAS =
        BEGIN
            MOVESCHITRPARAF(LHA
            PROCURASU
            ENCFRCC(R[L])=OJSP;
            AMDEUSL(R[L])=CONTA;
            LEUSU
            DESLOUSU(R[L])=2*ACUM+1;
            GUARDIASU
            FOR I=2 STEP 1 UNTIL CONTA 00
            BEGIN

```

```

00000480 003:0005:1
00000490 003:0005:1
00000500 003:0005:1
00000510 003:0005:1
00000520 003:0005:1
00000530 003:0005:1
00000540 003:0005:1
00000550 003:0005:1
00000560 003:0005:1
00000570 003:0005:1
00000580 003:0005:1
00000590 003:0005:1
00000600 003:0005:1
00000610 003:0005:1
00000620 003:0005:1
00000630 003:0005:1
00000640 003:0005:1
00000650 003:0005:1
00000660 003:0005:1
00000670 003:0005:1
00000680 003:0005:1
00000690 003:0005:1
00000700 003:0005:1
00000710 003:0005:1
00000720 003:0005:1
00000730 003:0005:1
00000740 003:0005:1
00000750 003:0005:1
00000760 003:0005:1
00000770 003:0005:1
00000780 003:0005:1
00000790 003:0005:1
00000800 003:0005:1
00000810 003:0005:1
00000820 003:0005:1
00000830 003:0005:1
00000840 003:0005:1
00000850 003:0005:1
00000860 003:0005:1
00000870 003:0005:1
00000880 003:0005:1
00000890 003:0005:1
00000900 003:0005:1
00000910 003:0005:1
00000920 003:0005:1
00000930 003:0005:1
00000940 003:0005:1
00000950 003:0005:1
00000960 003:0005:1
00000970 003:0005:1
00000980 003:0005:1
00000990 003:0005:1
00001000 003:0005:1
00001010 003:0005:1
00001020 003:0005:1
00001030 003:0005:1
00001040 003:0005:1
00001050 003:0005:1
00001060 003:0005:1
00001070 003:0005:1
00001080 003:0005:1

```

```

PAUX:=PALX+12;
IND3:=+2;
PROCURAPAI
LEUSU
CAMPOJAI:=LACUMAUX:=DTSP;
CAMPOJAI:=LACUMAUX:=2*ACUM+1;
ACUMAUX:=ACUMAUX+1;
WRITE(USLCCO(GUARDEND),41,ATPAUX);
GUARDALS;
LTPAATR
END;
CAMPOJAI:=LACUMAUX:=+;
ACUMAUX:=LTPAUX+1;
WRITE(USLCCO(GUARDEND),41,ATPAUX);
LTPAATR
CONT:=0;
PAUX:=PC(L)TER(ATR3)+42;
IND3:=4;
IF R(L) = (P-1) THEN
  BEGIN
    WRITE(DFSCHEW(R(L)),56,ATB1);
    PROCURAPAI(UE(L));
    PATN:=POINTER(ATB1(1));
    FOR X:=0 STEP 1 UNTIL 6 DO
      BEGIN
        SUPREUS(X)=0;
        FNDREUG(X)=0;
        CESUS(X)=0;
        REPLACE PATN + (48*X) BY * FOR 42
      END;
    M(L)=0;
    R(L)=ADVANCH
  END
  ELSE R(L)=R(L)+1;
  NOVENESCHITR
END; #1
INTEGER PROCEDURE NOVENCH;
BEGIN
  NOVENCH:=DISPCHVEL;
  DISPCHVEL:=+1;
END;
PROCEDURE PROCURAPAI(1);
VALUE 1;
INTEGER 1;
BEGIN
  ARRAY PAT(15);
  INTEGER K;
  POINTER PFILHC;
  DEFINE LINKPAI(1)PAI(8+1);
  K:=1;
  L:=+1;
  IF R(L) = (P-1) THEN
    PFILHC:=PATB1
  ELSE
    PFILHC:=PATB1+48;
    IF R(L) = C THEN
      BEGIN
        FOR X:=0 STEP 1 UNTIL 6 DO
          BEGIN

```

```

00001090 003:0005:1
00001100 003:0005:1
00001110 003:0005:1
00001120 003:0005:1
00001130 003:0005:1
00001140 003:0005:1
00001150 003:0005:1
00001160 003:0005:1
00001170 003:0005:1
00001180 003:0005:1
00001190 003:0005:1
00001200 003:0005:1
00001210 003:0005:1
00001220 003:0005:1
00001230 003:0005:1
00001240 003:0005:1
00001250 003:0005:1
00001260 003:0005:1
00001270 003:0005:1
00001280 003:0005:1
00001290 003:0005:1
00001300 003:0005:1
00001310 003:0005:1
00001320 003:0005:1
00001330 003:0005:1
00001340 003:0005:1
00001350 003:0005:1
00001360 003:0005:1
00001370 003:0005:1
00001380 003:0005:1
00001390 003:0005:1
00001400 003:0005:1
00001410 003:0005:1
00001420 003:0005:1
00001430 003:0005:1
00001440 003:0005:1
00001450 003:0005:1
00001460 003:0005:1
00001470 003:0005:1
00001480 003:0005:1
2 00001490 003:0006:4
00001500 003:0007:0
2 00001510 003:0009:2
00001520 003:0009:2
00001530 003:0009:2
00001540 003:0007:2
00001550 003:0009:2
PROCURAPAI IS SEGMENT 00004
2 00001560 004:0000:1
00001570 004:0000:1
00001580 004:0000:1
00001590 004:0000:1
00001600 004:0001:1
00001610 004:0002:3
00001620 004:0004:0
00001630 004:0004:3
00001640 004:0005:3
00001650 004:0007:4
00001660 004:0008:4
3 00001670 004:0009:1
00001680 004:0000:1

```

```

LINKPAT(X):=01
REPLACE POINTER(PAT(1)) + (48*X)
BY * * FOR 42
END
Q(1):=NOVONONH
FND
ELSE
  READ(DESCENT(0(1)), 56, PAT)
  REPLACE POINTER(PAT(1)) + (48*RE(1)) BY PFILND FOR 42
  LINKPAT(R(1)):0(K)
  WRITE(DESCENT(0(1)), 56, PAT)
  IF R(1) = (N-1) THEN
    BEGIN
      PROCURAPAT(0(1))
      RE(1)=0
    END
  ELSE
    RE(1)=R(1)+1
  ENULTIMONC
  PROCURAPAT(0(1))
END
END

VALUE ARRAY SIMUSCOO
(* 00019*, " 00027", " 00035", " 00043", " 00051", " 00059", " 00067",
* 00094", " 00102", " 00114", " 00124", " 00132", " 00140", " 00159",
* 00167", " 00175", " 00183", " 00191", " 00205", " 00213", " 00221",
* 00248", " 00256", " 00264", " 00272", " 00280", " 00299", " 00307",
* 00310", " 00325", " 00337", " 00345", " 00353", " 00361", " 00388")

DATA IS 002 LONG

L1=01
DISPON(VCL=1)
Q(1):=NOVONONH
[NOI:=0]
PATR:=POINTER(ATR)
PATR1:=POINTER(ATR1)+41
PATH2:=POINTER(ATR2)
PATR3:=POINTER(ATR3)
PATRAUX:=POINTER(ATRAUX)
PATR4:=POINTER(ATR4)+42
PDESCR10H:=POINTER(ATR3)
PSIMUSCOO:=POINTER(SIMUSCOO)
MIP:=7
PRON3:=POINTER(ATR3)
READ(DESCENT, 9, ATR2)
WRITE(DESCENT
WHILE NOT READ(DESCENT, 9, ATR2) DO
  IF PATR2 = PCV3 FOR 42
  THEN
    INSERTAISLM
  ELSE
    PROCURASUARIOS
  END

PROCURASUARIOS

TIML+1

```

```

4 00001690 004:000C:1
00001700 004:000D:5
00001710 004:000F:1
00001720 004:0012:0
4 00001730 004:0016:2
00001740 004:0017:1
3 00001750 004:0018:0
00001760 004:0018:0
00001770 004:001F:2
00001780 004:0022:5
00001790 004:0025:2
00001800 004:002C:2
00001810 004:002D:5
3 00001820 004:002E:2
00001830 004:002F:3
00001840 004:0030:1
3 00001850 004:0030:4
00001860 004:0030:4
00001870 004:0033:1
00001880 004:0037:3
00001890 004:0038:3
PROCURAPAT(004) IS 003F LONG
2 00001900 003:0009:2
00001910 003:0009:2
00001920 003:0009:2
00001930 003:0009:2
00001940 003:0009:2
00001950 003:0009:2
DATA IS 002 LONG
00001960 003:0009:2
00001970 003:0004:0
00001980 003:0004:4
00001990 003:000C:2
00002000 003:000D:1
00002010 003:000E:4
00002020 003:0010:1
00002030 003:0011:4
00002040 003:0013:1
00002050 003:0014:4
00002060 003:0016:2
00002070 003:0017:5
00002080 003:0019:2
00002090 003:0014:5
00002100 003:001C:2
2 00002110 003:0022:2
00002120 003:002F:3
00002130 003:0034:2
00002140 003:0035:4
00002150 003:0036:2
2 00002160 003:0042:3
00002170 003:0042:3
R.0001 IS SEGMENT 00006
R.0001(006) IS 0008 LONG
R.0002 IS SEGMENT 00007
R.0002(007) IS 0008 LONG
2 00002180 003:0149:0
R.0003 IS SEGMENT 00008
R.0003(008) IS 0008 LONG
R.0004 IS SEGMENT 00009
R.0004(009) IS 0008 LONG
2 00002190 003:024E:3

```

```

IF NLT( LLO (P=1) ) THEN
  BEGIN
    PHUCUSAPA(CE(L))
    WRITE(DESCHEW(CE(L)),56,ATTR)
  END;
FOR I=0 STEP 1 UNTIL 9 DO
  IF Q(I) = 0 THEN
    BEGIN
      ATTR(I)=3;
      ATTR(1)=Q(I-1);
      ATTR(2)=Q(I-1);
      WRITE(DESCHEW(P), 56,ATTR);
      I=10
    END;
LOCK(USUCON,.);
LOCK(DESCHEW,.);
CLOSE(DESCHEW, PURGE)
END;

```

```

00002200 003:024f15
00002210 003:025112
2 00002220 003:025115
00002230 003:025310
2 00002240 003:025514
00002250 003:025412
00002260 003:025011
00002270 003:025E11
2 00002280 003:025E14
00002290 003:026011
00002300 003:026210
00002310 003:026410
00002320 003:026412
00002330 003:026412
2 00002340 003:026F10
00002350 003:027013
00002360 003:027210
00002370 003:027312
P,0000(003) IS 0298 LONG

```

```

*****
NUMBER OF ERRORS DETECTED = 0.
NUMBER OF SEGMENTS = 7, TOTAL SEGMENT SIZE = 823 WORDS, CORE ESTIMATE = 4406 WORDS, STACK ESTIMATE = 47
PROGRAM SIZE = 238 CARDS, 3109 SYNTACTIC ITEMS, 63 DISK SEGMENTS.
PROGRAM FILE NAME: HOUTA/DESC,
COMPILED TIME = 100.948 SECONDS ELAPSED; 5.402 SECONDS PROCESSING; 5.054 SECONDS I/O.
*****

```

RENDONTA / DESCENT

```

PRG:IN
FILE
DESC (KIND=DISK, MAXRECSIZE=56, BLOCKSIZE=2128,
      AREA SIZE=2128, AREAS= 1),
HISUCOD (KIND=DISK, MAXRECSIZE=41, BLOCKSIZE=246),
DESCENT(KIND=DISK, MAXRECSIZE= 9, BLOCKSIZE=270,
        AREA SIZE= 30, AREAS*7))

ARRAY
  RUFUSUCOD(0:55),
  RUFDESC (0:55),
  RUFDESCENT(0: 8);
POINTER RUFUSUCOD, RUFDESC;
INTEGER I, K, L, C, AREA12,
  CHARECO, DESLOCAMENTU, TANTOSUSU;
DEFINE
  KNUVHOLDREC=RUFUSUCOD[DESLOCAMENTU].[26:21]#,
  KNUVDESLOCAMENTU=RUFUSUCOD[DESLOCAMENTU].[15:61]#,
  LINKP12(1)=RUFDESC[A+1]#;
PROCEDURE PROCURAFILHOS(I,Q);
  VALUE I,Q;
  INTEGER I, Q;
  BEGIN
    ARRAY FILHOS(0:55);

    INTEGER JJ;
    DEFINE
      LINKFILHOS(J)=FILHOS[A+1]#;
    READ(DESC(I), 55, FILHOS);
    IF I < L
    THEN
      FOR JJ=0 STEP 1 UNTIL 6 DO
        IF LINKFILHOS(J) NEQ 0
        THEN
          PROCURAFILHOS(I+1, LINKFILHOS(J))
        ELSE
          JJ=7
    ELSE
      BEGIN
        POINTER PFILHOS;
        DEFINE
          KNUVHOLDUSU(J)=FILHOS[A+1].[47:20]#,
          CHARECOUSU(J)=FILHOS[A+1].[27:22]#,
          CFSLOCCHUSU(J)=FILHOS[A+1].[51:6]#;
        PFILHOS:=POINTER(FILHOS) + 6;
        FOR JJ=0 STEP 1 UNTIL 6 DO
          IF KNUVHOLDUSU(J) NEQ 0
          THEN
            BEGIN
              00000000 000:0000:0
              8.0000 IS SEGMENT 00003
              1 00000010 003:0000:1
              00000020 003:0000:1
              00000030 003:0000:1
              DATA IS 0007 LONG
              00000040 003:0000:1
              DATA IS 0006 LONG
              00000050 003:0000:1
              00000060 003:0000:1
              DATA IS 0007 LONG
              00000070 003:0000:1
              00000080 003:0000:1
              00000090 003:0000:1
              00000100 003:0000:1
              00000110 003:0000:1
              00000120 003:0000:1
              00000130 003:0000:1
              00000140 003:0000:1
              00000150 003:0000:1
              00000160 003:0000:1
              00000170 003:0000:1
              00000180 003:0000:1
              00000190 003:0000:1
              00000200 003:0000:1
              00000210 003:0000:1
              00000220 003:0000:1
              PROCURAFILHOS IS SEGMENT 00004
              2 00000230 004:0000:1
              00000240 004:0000:1
              00000250 004:0000:1
              00000260 004:0000:1
              00000270 004:0007:2
              00000280 004:0007:4
              00000290 004:0008:1
              00000300 004:0008:1
              00000310 004:000C:2
              00000320 004:000C:4
              00000330 004:000F:2
              00000340 004:0010:0
              00000350 004:0010:3
              00000360 004:0015:1
              3 00000370 004:0015:4
              8.0001 IS SEGMENT 00005
              00000380 005:0000:1
              00000390 005:0000:1
              00000400 005:0000:1
              00000410 005:0000:1
              00000420 005:0000:1
              00000430 005:0001:4
              00000440 005:0004:1
              00000450 005:0005:5
              00000460 005:0006:1
            END
          END
        END
      END
    END
  END

```

```

TANTUSUSU := NUME(NDUSU(J))
ENDERECD := ENDERECDUSU(J)
DESLOCAMENTO := NDUSU(J)
REPLACE PDSCENT PDSCENT := RUFDESCEN BY
  PFILHO + (44*J) FOR 42;
THRU TANTUSUSU UN
  BEGIN
    READ(USUCON(NDERECD), 41, RUFUSUCON);
    REPLACE PDSCENT BY PUSUCON FOR 5;
    RUFDESCEN(1) := RUFUSUCON(DESLOCAMENTO+1);
    WRITE(DESCEN, 9, RUFDESCEN);
    ENDERECD := NDUSUCON(NDERECD);
    DESLOCAMENTO := NDUSUCON(DESLOCAMENTO);
  END
END

```

```

4 00000470 005:0006:4
00000480 005:0009:0
00000490 005:0008:2
00000500 005:0000:4
00000510 005:000E:5
00000520 005:0011:4
00000530 005:0012:1
00000540 005:0015:0
00000550 005:001C:2
00000560 005:001D:5
00000570 005:001F:0
00000580 005:001F:5
00000590 005:0026:2
00000600 005:0028:1
00000610 005:0028:5
5 00000620 005:002A:0
4 00000630 005:002C:0
3 00000640 005:002F:5

```

END

```

PUSUCON := POINTER(RUFUSUCON);
RUFDESC(0) := 56, RUFDESC(1) := RUFUSUCON(1);
NLRA12 := RUFDESC(2);
N1 := RUFDESC(1);
I1 := 1;
RUFDESC(0) := 56, RUFDESC(1) := RUFUSUCON(1);
FOR N1=0 STEP 1 UNTIL NLRA12 = 1 DO
  PROCURAFILHOS(1, LINKRA12(K));
  (RUFDESCENT,);
  CLFSE(USUCON, PUSUCON);
  CLFSE(DESC, PUSUCON);
END

```

```

R,0001(005) IS 0032 LONG
PROCURAFILHOS(004) IS 0010 LONG
2 00000650 003:0000:1
00000660 003:0001:4
00000670 003:0008:2
00000680 003:0009:3
00000690 003:000A:5
00000700 003:000C:0
00000710 003:000C:4
00000720 003:0013:2
00000730 003:0018:1
00000740 003:0018:1
00000750 003:001C:4
00000760 003:001E:1
00000770 003:001F:4
R,0000(003) IS 0034 LONG

```

```

*****
NUMBER OF FRAGS DETECTED = 0.
NUMBER OF SEGMENTS = 3. TOTAL SEGMENT SIZE = 151 WORDS. CORE ESTIMATE = 5110 WORDS. STACK ESTIMATE = 31
PROGRAM SIZE = 74 PARMS, 461 SYNTACTIC ITEMS, 25 DISK SEGMENTS.
PROCESSOR FILE NAME: RFNONTA/DESCENT.
COMPILED TIME = 12.511 SECONDS ELAPSED; 1.138 SECONDS PROCESSING; 1.986 SECONDS I/O.
*****

```

APÊNDICE D

PROGRAMA DO MÓDULO REFINA

É o programa responsável pela adaptabilidade do sistema de DSI.

O leitor, ao responder ao sistema através dos cartões-resposta, ativa este programa que irá fazer as alterações nos valores atribuídos aos descritores por ele. Estas alterações estão a cargo das funções matemáticas apresentadas no Capítulo IV.

REFINA

REFIN

FILE CARLOS (KIND=READER),

INCREMENTAR(KIND=DISK, MAXRECSIZE=5, BLCKSIZE=100,
AREASIZE=100, AREAS=1,
TITLE="INCREMENTOS"),

MOVIMOR (KIND=DISK, MAXRECSIZE=5, BLCKSIZE=100,
AREASIZE=100, AREAS=1,
TITLE="INCREMENTOS"),

LINHA (KIND=PRINTER),

USUCNO (KIND=DISK, MAXRECSIZE=41, BLCKSIZE=246),

CARTASORT (KIND=DISK, MAXRECSIZE=2, BLCKSIZE=210),

ARRAY

CARTAO[0:13],

AUX[0:1],

INCR[0:4],

USUCNO[0:19],

MUSUCNO[0:40],

MODULE PRIME[INCR],

PRINTER PCARTAO, PAUX, MUSUCNO, POST, PCODIGO,

INTEGER I, NUMEROUSU, CIVISOM,

LABEL FINE

OFFINE

MUSUCNOUSUCNO=INCR[3],

RELEVANCIA *PAUX + 10 = *R*,

RESLONCUSU[1]=INCR[1-1] DIV R, (47-(1-1) MOD R)*616],

SANFUSUCNO[CFELONCUSU[1]-1],

MODULE PRIME[INCR],

REFIN

INTEGER CND1, CND2, CND3, CND4, CND5, Y

CND1:=INTEGER(PCND1G,1)

CND2:=INTEGER(PCND1G+1,1)

CND3:=INTEGER(PCND1G+2,1)

CND4:=INTEGER(PCND1G+3,1)

CND5:=INTEGER(PCND1G+4,1)

Y:=(CND1*5 + CND2*4 + CND3*3 + CND4*2 + CND5) MOD 11

CND1G:= IF Y = 0 THEN TRUE ELSE FALSE

END

MODULE PROCEDURE COMPRA(A, B)

ARRAY A, B[0]

REFIN

PRINTER PA, PB

COMPARA:= IF PA[0] < PB[0] FOR 5
THEN

00000000 000:0000:0

0.0000 IS SEGMENT 00003

1 00000010 003:0000:1

DATA IS 0005 LONG

00000020 003:0000:1

00000030 003:0000:1

00000040 003:0000:1

DATA IS 0004 LONG

00000050 003:0000:1

00000060 003:0000:1

00000070 003:0000:1

DATA IS 0009 LONG

00000080 003:0000:1

DATA IS 0005 LONG

00000090 003:0000:1

DATA IS 0006 LONG

00000100 003:0000:1

DATA IS 0007 LONG

00000110 003:0000:1

00000120 003:0000:1

00000130 003:0000:1

00000140 003:0000:1

00000150 003:0000:1

00000160 003:0000:1

00000170 003:0000:1

00000180 003:0000:1

00000190 003:0000:1

00000200 003:0000:1

00000210 003:0000:1

00000220 003:0000:1

00000230 003:0000:1

00000240 003:0000:1

00000250 003:0000:1

00000260 003:0000:1

00000270 003:0000:1

00000280 003:0000:1

00000290 004:0000:1

2 00000300 004:0001:5

00000310 004:0002:0

00000320 004:0003:2

00000330 004:0004:4

00000340 004:0005:0

00000350 004:0006:3

00000360 004:0007:2

00000370 004:0008:1

2 00000380 004:0009:1

00000390 004:0010:1

00000400 004:0011:1

00000410 004:0012:1

00000420 004:0013:1

00000430 004:0014:1

00000440 004:0015:1

00000450 004:0016:1

00000460 004:0017:1

00000470 004:0018:1

00000480 004:0019:1

00000490 004:0020:1

```

      TRUE
    ELSE
      IF PA = PR FOR 5
      THEN
        IF PA49 <= PR 45 FOR 5
        THEN
          TRUE
        ELSE
          FALSE
        ELSE
          FALSE
        ELSE
          FALSE
    END;

REAL PROCEDURE NOVOPESQ
  BEGIN
    REAL DELTA, S1;

    S1:=ARCSIN(2*5-1);
    DELTA:=(1.5708- ARS(S1))/DIVISOR;
    NOVOPESQ := IF RELEVANCIA THEN
      (SIN(S1*DELTA)+1)/2
    ELSE
      (SIN(S1*DELTA) + 1)/2
    END;

  DIVISOR:= 8;
  PAUX:=PRINTER(BU)USUCOD;
  PAUX:=PRINTER(AUX);
  PCARTAC:=PRINTER(CARTAC)+6;
  PRES:=PRINTER(INCR(4));
  WHILE NOT READ(CARTES, 14, CARTAO) DO
    IF PCARTAO :=PRINTER(CARTAO) = "PERFIL"
    THEN
      IF CODIGO
      THEN
        BEGIN
          PRINTER(CARTAO)=TRUE;
          REPLACE PAUX BY PCARTAO + 6 FOR 5
        END
      ELSE
        BEGIN
          WRITE(LINHA, < ARO, //, "ERRO NO NUMERO DO PERF"
            , "IL">);
          PCARTAO FOR 6);
          PRINTER(CARTAO)=FALSE
        END
      ELSE
        IF PRINTER(CARTAO) THEN
          FOR J:=1 STEP 1 UNTIL 13 DO
            IF PCARTAC NEQ " " FOR 6
            THEN
              BEGIN
                IF PCARTAO + 5 = "R" OR PCARTAO + 5 = "I" THEN
                  BEGIN
                    REPLACE PAUX + 5 BY PCARTAO+PCARTAO FOR 6;
                    WRITE(CARTAO SORT, 2, AUX);
                  END
                ELSE
                  BEGIN
                    WRITE(LINHA, < X5, X2, A6, //, X10, "ERRO TE"
                      , "QUE SER (R) OU (I)">);

```

```

00000430 005:0003:1
00000440 005:0003:4
00000450 005:0004:2
00000460 005:0005:2
00000470 005:0006:0
00000480 005:0008:5
00000490 005:0009:3
00000500 005:0014:0
00000510 005:0014:4
00000520 005:0014:4
00000530 005:0018:2
00000540 005:0018:2
COMPARA(005) IS 000F LONG
2 00000550 003:0000:1
00000560 003:0000:1
00000570 003:0000:1
NOVOPESQ IS SEGMENT 00006
2 00000580 006:0000:1
00000590 006:0007:1
00000600 006:0004:5
00000610 006:0000:2
00000620 006:000F:4
00000630 006:0010:4
00000640 006:0012:3
NOVOPESQ(006) IS 0015 LONG
2 00000650 003:0000:1
00000660 003:0011:0
00000670 003:0002:3
00000680 003:0004:0
00000690 003:0005:3
00000700 003:0007:1
00000710 003:0008:2
00000720 003:0000:1
00000730 003:000F:4
00000740 003:0010:1
00000750 003:0010:5
2 00000760 003:0011:2
00000770 003:0012:0
00000780 003:0013:4
2 00000790 003:0014:1
00000800 003:0014:1
2 00000810 003:0014:4
00000820 003:0016:3
00000830 003:0017:2
00000840 003:001E:2
00000850 003:001E:2
2 00000860 003:001F:0
00000870 003:001F:0
00000880 003:001F:5
00000890 003:0023:1
00000900 003:0025:0
00000910 003:0025:4
00000920 003:0026:1
00000930 003:0029:5
3 00000940 003:002C:1
00000950 003:002F:0
00000960 003:0030:3
3 00000970 003:0035:2
00000980 003:0035:2
3 00000990 003:0035:5
00010000 003:0037:4

```

```

      PAUX, PCARTAU))
      PCARTAU:=PCARTAU+6
    END
  ELSE
    IF I=14
      ELSE WRITE(LINHA, CARO,/, "***** CARTAO RECUSADO*")
      PCARTAU))
    LOCK(CARTASORT,*)
    SORT(CARTASORT, CARTASCRT, 0, COMPARA, 2))
    READ(INCREMENTAR, 5, INCR))
    READ(USUCOD, NUMEROEMUSU(1), 41, BUFUSUCOD))
    WHILE NOT HEAD(CARTASCRT, 2, AUX) DO
      BEGIN
        WHILE PAUX NEQ PUSUARIO FOR 5 DO
          READ(USUCOD, NUMEROEMUSU(1), 41, BUFUSUCOD))
          WHILE NUMEROUSUARIO < NUMEROEMUSU DO
            BEGIN
              WRITE(MOVINCR, 5, INCR))
              READ(INCREMENTAR, 5, INCR) (FIN)
            END
          WHILE PDESI < PAUX + 5 FOR 5
            AND
            NUMEROUSUARIO = NUMEROEMUSU DO
            BEGIN
              WRITE(MOVINCR, 5, INCR))
              READ(INCREMENTAR, 5, INCR) (FIN)
            END
          IF PDESI = PAUX + 5 FOR 5
            AND
            NUMEROUSUARIO = NUMEROEMUSU
          THEN
            BEGIN
              FOR I= 1 STEP 1 UNTIL 20 DO
                IF DESLOCOSU(1) NEQ 63
                  THEN
                    SINOVOPESO
                  ELSE
                    I= 21
                    READ(INCREMENTAR, 5, INCR) (FIN)
                    WRITE(USUCOD, NUMEROEMUSU(1), 41, BUFUSUCOD)
                END
              END
            BEGIN
              WRITE(MOVINCR, 5, INCR))
              WHILE NOT HEAD(INCREMENTAR, 5, INCR) DO
                WRITE(MOVINCR, 5, INCR))
              END
            BEGIN
              LOCK(USUCOD,*)
              LOCK(MOVINCR,*)
              CLOSE(INCREMENTAR, PURGE)
            END

```

	DATA	IS	0008	LONG
	00001010	003:0038:3		
	00001020	003:003F:2		
	00001030	003:003F:5		
3	00001040	003:0041:0		
2	00001050	003:0041:0		
	00001060	003:0041:0		
	00001070	003:0041:3		
	00001080	003:0049:3		
	00001090	003:004E:5		
	00001100	003:0050:2		
	00001110	003:0056:5		
	00001120	003:0050:2		
	00001130	003:0064:2		
	00001140	003:0068:2		
2	00001150	003:0068:4		
	00001160	003:0064:2		
	00001170	003:0072:5		
	00001180	003:0074:0		
3	00001190	003:0074:3		
	00001200	003:0074:2		
	00001210	003:0080:4		
3	00001220	003:0082:4		
	00001230	003:0084:2		
	00001240	003:0085:0		
	00001250	003:0086:2		
3	00001260	003:0086:5		
	00001270	003:0080:2		
	00001280	003:0093:4		
3	00001290	003:0095:4		
	00001300	003:0097:2		
	00001310	003:0098:0		
	00001320	003:0098:4		
	00001330	003:0099:2		
3	00001340	003:0099:5		
	00001350	003:009C:1		
	00001360	003:0040:3		
	00001370	003:0041:0		
	00001380	003:0046:4		
	00001390	003:0047:3		
	00001400	003:004C:4		
	00001410	003:0065:1		
	00001420	003:00H7:3		
3	00001430	003:008C:2		
2	00001440	003:008C:5		
	00001450	003:00C3:2		
	00001460	003:00C7:2		
	00001470	003:00C0:5		
	00001480	003:00C0:5		
	00001490	003:00CF:2		
	00001500	003:00D0:5		
	00001510	003:00D2:1		
	R,0000(003)	IS 00FC LONG		
		DATA IS 0018 LONG		

```

*****
NUMBER OF ERRORS DETECTED = 0.
NUMBER OF SEGMENTS = 0. TOTAL SEGMENT SIZE = 386 WORDS. CORE ESTIMATE = 14243 WORDS. STACK ESTIMATE = 38
PROGRAM SIZE = 157 CARDS. 849 SYNTACTIC ITEMS, 47 DISK SEGMENTS.
PROGRAM FILE NAME: RFFINA.
COMPILATION TIME = 20.731 SECONDS ELAPSED: 1.907 SEC. NOS PROCESSING: 3.778 SECONDS I/O.
*****

```

APÊNDICE E

PROGRAMA DO MÓDULO BUSCA

A tarefa aqui é ler o arquivo que contém as publicações a serem disseminadas, uma por uma, e "buscar" no arquivo dos usuários aqueles que têm descritores em seus perfis coincidindo com descritores dos perfis dos documentos.

A função aqui usada para se fazer a disseminação é uma função de probabilidade, chamada no programa TDEK.

Os usuários para os quais selecionamos documentos são colocados no arquivo RECUPERADO, para ser usado no programa de Impressão.

R U S C A

REATH

FILE RECUPEA00(KIND=DISK, MAXRECSIZE=8, BLOCKSIZE=240),

RESC (KIND=DISK, MAXRECSIZE=56, BLOCKSIZE=2128,
AREASIZE=2128, AREAS=1),

DESJOLU (KIND=DISK, MAXRECSIZE=57, BLOCKSIZE=570),

ARQOTU (KIND=DISK, MAXRECSIZE=290, BLOCKSIZE=2900),

INPRIMA (KIND=PRINTER),

INPRIMAER (KIND=PRINTER),

INCREMENTAR(KIND=DISK, MAXRECSIZE=5, BLOCKSIZE=100,
AREASIZE=100, AREAS=1),

INCRFHMT (KIND=DISK, MAXRECSIZE=5, BLOCKSIZE=100,
AREASIZE=100, AREAS=1,
OPTICHA[TRUE+TITLE="INCREMENTHS,"],

INCREMENTGE (KIND=DISK, MAXRECSIZE=5, BLOCKSIZE=100,
AREASIZE=100, AREAS=1,
TITLE="INCREMENTHS,"),

RAIXOINCR (KIND=DISK, MAXRECSIZE=5, BLOCKSIZE=30),

USU (KIND=DISK, MAXRECSIZE=20, BLOCKSIZE=60),

USUCOD (KIND=DISK, MAXRECSIZE=41, BLOCKSIZE=246),

INTEGER L, ILRATZ,

DIYSON,

NIYFL, HEDEF, INC, PROXEND, CONT,

J, JJ, JS, ALMERGFMDESI,

R, I, A, C, C), P,

ENDUSU, TANTCSUSU, DESLOCAMENTO,

PFAL TURK,

ARRAY RUFDES(0156),

RUFDES(0155),

RUFH(0101230),

RAIZ (0155),

RUFREUP(017),

RUFHATCM(011),

SPATEIA(011),

MR, WID12),

SPH(0120),

INC(014),

RUFDES(01016),

RECUPEA(0110),

RUFH(010120),

PRINTER PPECUP,POESC, PRATZ,

00000000 000:0000:0

R,0000 IS SEGMENT 00003

00000010 003:0000:1

DATA IS 0007 LONG

00000020 003:0000:1

00000030 003:0000:1

DATA IS 0007 LONG

00000040 003:0000:1

DATA IS 0006 LONG

00000050 003:0000:1

DATA IS 0006 LONG

00000060 003:0000:1

DATA IS 0005 LONG

00000070 003:0000:1

DATA IS 0006 LONG

00000080 003:0000:1

00000090 003:0000:1

DATA IS 0008 LONG

00000100 003:0000:1

00000110 003:0000:1

00000120 003:0000:1

DATA IS 0008 LONG

00000130 003:0000:1

00000140 003:0000:1

00000150 003:0000:1

DATA IS 000A LONG

00000160 003:0000:1

DATA IS 0007 LONG

00000170 003:0000:1

DATA IS 0006 LONG

00000180 003:0000:1

DATA IS 0006 LONG

00000190 003:0000:1

00000200 003:0000:1

00000210 003:0000:1

00000220 003:0000:1

00000230 003:0000:1

00000240 003:0000:1

00000250 003:0000:1

00000260 003:0000:1

00000270 003:0000:1

00000280 003:0000:1

00000290 003:0000:1

00000300 003:0000:1

00000310 003:0000:1

00000320 003:0000:1

00000330 003:0000:1

00000340 003:0002:4

00000350 003:0002:4

00000360 003:0002:4

00000370 003:0002:4

00000380 003:0002:4

00000390 003:0002:4


```

REPLACE PDES1 BY * FOR 7 WORDS)
SCAN PDES1:PDES1 FOR JJJJ UNTIL *[*]
IF NOT TOGGLE THEN
IF PDES1 + 2 = * THEN
BEGIN
REPLACE PDES1 BY PDES1 + 1 FOR 1;
JJJ:=3;
PDES1:=PDES1 + 3;
REPLACE PDES1 BY PDES1:PDES1 FOR JJJJ WHILE NEQ *[*]
NOVODESCRIPTOR:=IF TOGGLE THEN FALSE ELSE TRUE;
REPLACE PDES1 BY PDES1 FOR 42 WITH TRANS;
WRITE(IMPRTAER, < A42>, PDES1);
ELSE NOVODESCRIPTOR:=FALSE
ELSE
NOVODESCRIPTOR:=FALSE
END;

PROGRAM PROCEDURE BUSCA1
IF PDES1 > PRAIZ + (ALRAIZ - 1)*48 + 6) FOR 42
THEN
BUSCA1:=FALSE
ELSE
BEGIN
JJ:=0;
PESQUISA N1 PRAIZ
WHILE PDES1 > PRAIZ + (6*JJ+6) FOR 42 DO JJ:=JJ+1;
READ(PDES1(PRAIZ+JJ)*56, BUFDESC);
PESQUISA EN NON NAO TERMINAL
THRU L=1 DO
BEGIN
R:=0;
WHILE PDES1 > PDESC + (6*R+6) FOR 42 DO R:=R+1;
READ(PDESC(ENDEHEC(R))*56, BUFDESC);
END;
PESQUISA EN NON TERMINAL
R:=0;
WHILE PDES1 > PDESC + (6*R+6) FOR 42 DO R:=R+1;
IF PDES1 = PDESC + (6*R+6) FOR 42
THEN
BEGIN
BUSCA1:=TRUE;
ENCLSU:=ENDEHECENUSU(R);
DESLCAMEIN:=DESLCENUSUL(R);
TATEUSUSU:=NUMENDEUSUARIUS(R);
END
ELSE
BUSCA1:=FALSE
END;

PROGRAM PRINCIPAL
PRAIZ:=PRAIZ(RAIZ);
PDESC:=PDESC(RDESC);
P1:=P1(R1);
P1M2:=P1M2(R1M2);
R1:=P1(R1);
L1:=P1(R1);
G1:=P1(R1);

```

NOVODESCRIPTOR IS SEGMENT 00006
DATA IS 0040 LONG

2	00000970	006:0000:1
	00000980	006:0002:3
	00000990	006:0005:0
	00001000	006:0005:1
	00001010	006:0008:3
3	00001020	006:0009:0
	00001030	006:0004:5
	00001040	006:0002:2
	00001050	006:0002:0
	00001060	006:0011:3
	00001070	006:0013:3
	00001080	006:0016:1
	00001090	006:0010:2
3	00001100	006:0010:2
	00001110	006:0010:5
	00001120	006:0011:3
	00001130	006:0011:0
NOVODESCRIPTOR(006) IS 0023 LONG		
2	00001140	003:0022:4
	00001150	003:0022:4
	00001160	003:0006:1
	00001170	003:0006:5
	00001180	003:0007:2
	00001190	003:0008:0
2	00001200	003:0008:3
	00001210	003:0009:1
	00001220	003:0009:1
	00001230	003:0009:0
	00001240	003:0016:2
	00001250	003:0016:2
	00001260	003:0017:1
3	00001270	003:0014:0
	00001280	003:0014:4
	00001290	003:0020:3
	00001300	003:0023:1
3	00001310	003:0029:2
	00001320	003:0029:2
	00001330	003:0024:0
	00001340	003:0027:5
	00001350	003:0032:3
	00001360	003:0033:1
3	00001370	003:0033:4
	00001380	003:0034:2
	00001390	003:0036:1
	00001400	003:0038:0
	00001410	003:0039:5
3	00001420	003:0039:5
	00001430	003:0039:5
	00001440	003:0039:2
2	00001450	003:0039:3
	00001460	003:0039:3
	00001470	003:0039:3
	00001480	003:0039:3
	00001490	003:0039:0
	00001500	003:0039:3
	00001510	003:0041:1
	00001520	003:0042:4
	00001530	003:0049:2
	00001540	003:0044:3

```

MUNA[2]=HUFDESC(2);
MUNIF402MUESTI=1;
PDEST:=POINTER(HUFDESTIIX);
READ(DESC(40)), 56, RATZ);
WHILE NOT HEAD(CESTOFN+57, BUFDST) DO
  BEGIN
    WRITE(IMPRI=ACR, <"/> DOCUMENTO NUMERO 1=, AS>>POINTER(HUFDESTI));
    PDEST1:=POINTER(HUFDESTI);
    JJ:=J+2;
    KUNEPCEDESI:=+1;
    REDEF:=IND:=0;
    PROXEND:=0;
    CONT:=0;
    WHILE NOVODESCRIPTOR DO
      IF BUSCA
      THEN
        BEGIN
          REDEF:=+2+TANTOSUSU;
          RESIZE(NUFMATCH, REDEF, RETAIN);
          TNR:=TANTOSUSU DO
            BEGIN
              READ(USUCOD(ENDFUSU), 61, BUFSUCOD);
              CONT:=+1;
              ENDFMATCH(PROXEND)=ENDEUSU;
              DESLOCMATCH(PROXEND)=DESLCAMENTO;
              BUFMATCH(PROXEND+1)=BUFSUCOD(DESLOCAMENTO + 1);
              ENDFUSU:=NOVUSUARIO;
              DESLOCAMENTO:=NOVODESLOCAMENTO;
              PROXEND:=+2;
            END;
          END;
        END;
      ELSE
        WRITE(IMPRI=ACR, <" **** NAO EXISTE USUARIO COM ESTE",
          " DESCRI=O">>PDEST);
        DATA IS 000R LONG
        00001550 003:0048:5
        00001560 003:0040:1
        00001570 003:004E:0
        00001580 003:004F:3
        00001590 003:0056:2
        00001600 003:005A:2
        2 00001610 003:005A:4
        00001620 003:0062:2
        00001630 003:0063:5
        00001640 003:0064:5
        00001650 003:0066:1
        00001660 003:0067:3
        00001670 003:0068:1
        00001680 003:0068:5
        00001690 003:0069:3
        00001700 003:006A:0
        00001710 003:006A:4
        3 00001720 003:006B:1
        00001730 003:006B:1
        00001740 003:006F:1
        00001750 003:006F:4
        4 00001760 003:0072:3
        00001770 003:0079:2
        00001780 003:007A:4
        00001790 003:007C:5
        00001800 003:007F:0
        00001810 003:0081:2
        00001820 003:0083:1
        00001830 003:0085:0
        00001840 003:0085:2
        4 00001850 003:0086:3
        3 00001860 003:0088:3
        00001870 003:0088:3
        00001880 003:008A:5
        DATA IS 000R LONG
        00001890 003:0090:5
        00001900 003:0096:3
        3 00001910 003:0096:3
        00001920 003:0096:0
        00001930 003:0099:0
        00001940 003:009B:3
        00001950 003:009D:5
        00001960 003:009F:4
        00001970 003:00A0:2
        00001980 003:00A2:3
        00001990 003:00A3:1
        4 00002000 003:00A3:4
        00002010 003:00A5:5
        00002020 003:00A9:5
        00002030 003:00AC:2
        00002040 003:00AD:2
        00002050 003:00AD:4
        4 00002060 003:00AE:5
        00002070 003:00B0:4
        00002080 003:00B1:0
        00002090 003:00B1:3
        4 00002100 003:00B2:0
        00002110 003:00B4:3
        00002120 003:00B6:5
        00002130 003:00B8:1
        4 00002140 003:00B8:4

```

```

PSOMA:=SCPA + 1
C1:=0;
C11:=1;
PCOMP:=RUFHATCH;
WHILE C<REDEF=1 DO
  BEGIN
    FOR I:=1 STEP 1 UNTIL 20 DO
      DESLOCODUSU(I):=63;
    P:=1;
    ENDSOMA:=FINDERHATCH(C);
    REPLACE POINTER(INCR(4)) BY POINTER(RUFDES) FOR
      6;
    INCR(3):=ENDSOMA;
    DESLOCODUSU(P):=DESLOCMATCH(C);
    SCMA(P):=RUFHATCH(C1);
    C:=C+2;
    C1:=C+2;
    IF C<REDEF THEN
      PCOMP:=PCOMP + 12;
    WHILE C<REDEF AND PSOMA = PCOMP FOR 5 DO
      BEGIN
        P:=P+1;
        SCMA(P):=RUFHATCH(C1);
        DESLOCODUSU(P):=DESLOCMATCH(C);
        C:=C+2;
        C1:=C+2;
        IF C < REDEF THEN
          PCOMP:=PCOMP+12;
        END;
      TUEK:=0;
      FOR K:=1 STEP 1 UNTIL P DO
        TUEK:=SCMA(K) - SCMA(K)+TUEK;
      IF TUEK > 5
        THEN
          BEGIN
            ENDESCODUSU:=ENDSOMA;
            RUFRECUP(1):=NUMERENDESU;
            RUFRECUP(2):=TUEK;
            REPLACE PR1 BY PINCR FOR 5 WORDS
            WRITE(RECUPERADO, R, RUFRECUP);
          END
        ELSE
          EQUAD=NUMERENDESU DO DOCUMENTO E
          WRITE(RAIXO/OCR, S, INCR);
      END;
    REPLACE POINTER(RUFDES) BY " " FOR 57 WORDS
  END;
  LOCK(RECUPERADO, 1);
  SORT(RECUPERADO, RECUPRAUD, 0, ORD, R);
  *****
  RAIXO
  *****
  MODULO INRRIMIR
  *****
  ARRAY
  *****
  RUFDESU(0119);
  POINTER PUSU;
  LEAF L1;
  FORMAT F1(
    A30,///,
    "
    PRELADO PFSQUISADOM, PARA MELHOR SELECAO,
    /, "INFORME QUANTO A RELEVANCIA PARA OS DOCUMENTOS", /,
    "RAIXO, USANDO OS CODIGOS:", /,
    /, X25, "RA PARA RELEVANTE", /,

```

```

3 00002150 003:0089:1
00002160 003:008A:5
00002170 003:008B:3
00002180 003:008C:1
00002190 003:008D:2
00002200 003:008E:3
3 00002210 003:008F:0
00002220 003:008G:1
00002230 003:008H:2
00002240 003:008I:0
00002250 003:008J:3
00002260 003:008K:4
00002270 003:008L:1
00002280 003:008M:1
00002290 003:008N:2
00002300 003:008O:0
00002310 003:008P:3
00002320 003:008Q:0
00002330 003:008R:5
00002340 003:008S:0
00002350 003:008T:4
4 00002360 003:008U:1
00002370 003:008V:3
00002380 003:008W:5
00002390 003:008X:0
00002400 003:008Y:3
00002410 003:0090:0
00002420 003:0091:5
00002430 003:0092:0
4 00002440 003:0093:3
00002450 003:0094:1
00002460 003:0095:4
00002470 003:0096:1
00002480 003:0097:3
00002490 003:0098:3
4 00002500 003:0099:0
00002510 003:0100:1
00002520 003:0101:2
00002530 003:0102:4
00002540 003:0103:4
00002550 003:0104:4
4 00002560 003:0105:2
00002570 003:0106:2
00002580 003:0107:2
3 00002590 003:0108:5
00002600 003:0109:2
2 00002610 003:0110:0
00002620 003:0111:3
00002630 003:0112:5
00002640 003:0113:5
2 00002650 003:0114:5
00002660 003:0115:5
8,0001 15 SEGMENT 00008
00002670 008:0000:1
00002680 008:0000:1
00002690 008:0000:1
00002700 008:0000:1
00002710 008:0000:1
00002720 008:0000:1
00002730 008:0000:1
00002740 008:0000:1

```

```

X25, "A) PARA IRRELEVANTE",//);

DEFINIR
  IMPRIMIRUSUARIO
    WRITE(IMPRIHA, <///X25, "NUMERO DO PERFIL ", A5, //>, PUSU);
    WRITE(IMPRIHA, F1, PUSU + 5 FOR 30);
    / WRITE(IMPRIHA, < 3(A6, " * " )//>, FOR I=1 STEP 1
      UNTIL CONT=1 DO OSMAIORES(I));
    WRITE(IMPRIHA, <///X10, "L40(A), SR(A),", //X10,
      A30, //X10, A27, //X10, A19, //X3, A5, X2, A2>,
      FOR I=5, 56, A3, 104, 102 DO PUSU + 1);
    ENDEUSU:=ENDEECUSU;
PROCEDER IMPRIMIRRECUPERADON;
REGIM
  PONERAN LOOP;

  READ(AMORHU[BUFRUCUP(1)], 250, RUFRIB);
  WRITE(IMPRIHA, <///X5A, A5, //A5>, PUSU, POINTER(BUFRIB));
  PUES1:=POINTER(RUFRIB);
  REPLACE POINTER(OSMAIORES(CONT)) BY PUES1 FOR 5;
  REPLACE PUSU BY PUSU FOR 5 WORDS;
  WRITE(IMPRIHA, < 5, "INCH);
  CONT:=++1;
  JJ:=1740;
  SCAN PUES1:PUES1 FOR JJ:JJ UNTIL "A";
  IF NOT TOGGLE THEN
    REGIM
      LOOP:=IF JJ < 3 THEN FALSE ELSE
        IF PUES1 = "A2" FOR 3 THEN TRUE ELSE FALSE;
      PUES1:=PUES1+1;
    WHILE LOOP DO
      IF JJ > 60 THEN
        REGIM
          SCAN PUES1 FOR K1:60 UNTIL "A";
          I:=K1;
          JJ:=JJ+1;
          WRITE(IMPRIHA, < A>+I+PUES1 FOR I);
          PUES1:=PUES1+1;
          IF JJ>0 THEN
            IF PUES1 = "A" THEN LOOP:=FALSE;
            ELSE
              ELSE
                LOOP:=FALSE;
          END
        ELSE
          REGIM
            I:=JJ;
            SCAN PUES1 FOR JJ:JJ UNTIL "A";
            I:=JJ;
            WRITE(IMPRIHA, < A>+I, PUES1);
            LOOP:=FALSE;
          END
        END
      END;
    END;
  END;
  PUES1:=POINTER(RUFUSU);
  CONT:=1;
  REPLACE POINTER(OSMAIORES) BY "A" FOR 10 WORDS;
  READ(AMORHUFRADON, A, BUFRUCUP(1));
  PUESU:=ENDEECUSU;
  READ(AMORHU[ENDEUSU], 20, RUFUSU);

```

00002750	008:0000:1
DATA IS 000C LONG	
00002760	008:0000:1
00002770	008:0000:1
00002780	008:0000:1
00002790	008:0000:1
00002800	008:0000:1
00002810	008:0000:1
00002820	008:0000:1
00002830	008:0000:1
00002840	008:0000:1
00002850	008:0000:1
00002860	008:0000:1
00002870	008:0000:1
00002880	008:0000:1
IMPRIMIRRECUPERADON IS SEGMENT 00000	
3 00002890	008:0000:1
00002900	008:0007:2
00002910	008:0011:2
00002920	008:0012:5
00002930	008:0015:2
00002940	008:0016:5
00002950	008:0018:2
00002960	008:001E:4
00002970	008:001F:4
00002980	008:0022:1
00002990	008:0022:2
00003000	008:0022:5
00003010	008:0024:5
00003020	008:0029:3
00003030	008:0028:1
00003040	008:0028:3
00003050	008:002C:5
00003060	008:002D:2
00003070	008:002F:3
00003080	008:0031:0
00003090	008:0032:3
00003100	008:0030:2
00003110	008:003F:0
00003120	008:003F:4
00003130	008:0042:3
00003140	008:0043:1
00003150	008:0043:1
00003160	008:0044:2
00003170	008:0044:2
00003180	008:0044:2
00003190	008:0044:5
00003200	008:0045:5
00003210	008:0048:0
00003220	008:0049:3
00003230	008:0052:2
00003240	008:0052:2
5 00003250	008:0053:0
4 00003260	008:0053:3
IMPRIMIRRECUPERADON IS 005A LONG	
3 00003270	008:0000:1
00003280	008:0001:4
00003290	008:0002:2
00003300	008:0005:3
00003310	008:000E:1
00003320	008:000F:5

```

IMPRINIRRECUPERADO ;
WHILE NOT READ(RECUPERADO, B, BUFRECUP, 0)
  BEGIN
    IF ENDERECOUSU NEQ ENDEUSU OR CONT > 10 THEN
      BEGIN
        IMPRINIRUSUARIO
        WRITE(IMPRIMA (SKIP 1))
        IF CONT > 10 THEN
          WHILE ENDERECOUSU = ENDEUSU DO
            BEGIN
              REPLACE PIACR BY PRI FOR 5 WORDS;
              WRITE(BAIXOINCR, 5, INCR);
              READ(RECUPERADO, B, BUFRECUP) (L1);
              ENDEUSU = ENDERECOUSU;
            END;
            READ(LSU(ENDEUSU), 20, RUFUSU);
            CONT = 1;
          END;
        IMPRINIRRECUPERADO ;
      END;
    IMPRINIRUSUARIO
  L1:
    CLOSE(RECUPERADO, PURGE);
    LOCK(INCREMENTAR, *);
    SORT(INCREMENTAR, INCREMENTAR, 0, ORDENAINCR, 5);
    MERGE(INCREMENTAR, ORDENAINCR, 5, INCREMENT, INCREMENTAR);
    CLOSE(INCREMENT, PURGE);
    CLOSE(INCREMENTAR, PURGE);
    LOCK(BAIXOINCR, *);
    SORT(BAIXOINCR, BAIXOINCR, 0, ORDENAINCR, 5);
    LOCK(DESC, *);
    * =====
    * = MODULO BAIXO INCREMENTO
    * =====
  BEGIN
    DEFINE
      S=BUFUSUCOD(DESLOCOUSU[1]*114)
    ENDEUSU:=1;
    DIVISOR:=16;
    WHILE NOT READ(BAIXOINCR, 5, INCR) DO
      BEGIN
        IF ENDEUSU NEQ INCR(3) THEN
          BEGIN
            READ(USUCOD[INCR(3)], 41, BUFUSUCOD);
            ENDEUSU:=INCR(3);
          END;
          FOR I:=1 STEP 1 UNTIL 20 DO
            IF DESLOCOUSU[I] = 63
              THEN
                I:=21
              ELSE
                BEGIN
                  REAL S4;
                  S4=ARCSIN(2*S+1);
                  S:=(S)*N(S4+(1.5708*AR5(S4))/DIVISOR)*11/2
                END;
              END;
      END;

```

```

00003330 008:0016:2
00003340 008:0017:0
00003350 008:0018:2
3 00003360 008:0018:4
00003370 008:001E:1
4 00003380 008:001E:4
DATA IS 003A LONG
DATA IS 0012 LONG
00003390 008:0058:0
00003400 008:005C:2
00003410 008:005D:1
00003420 008:005F:1
5 00003430 008:005F:4
00003440 008:0061:1
00003450 008:0067:2
00003460 008:006F:1
00003470 008:0070:5
3 00003480 008:0071:2
00003490 008:0078:2
00003500 008:0078:2
4 00003510 008:0079:0
00003520 008:0079:4
3 00003530 008:007A:1
DATA IS 0028 LONG
00003540 008:0083:0
00003550 008:0083:0
00003560 008:0084:3
00003570 008:0086:0
00003580 008:008C:5
00003590 008:00C2:4
00003600 008:00C4:1
00003610 008:00C5:4
00003620 008:00C7:1
00003630 008:00C0:2
00003640 008:00CE:5
00003650 008:00CE:5
00003660 008:00CE:5
00003670 008:00CE:5
3 00003680 008:00CE:5
B,0007 IS SEGMENT 00011
00003690 011:0000:1
00003700 011:0000:1
00003710 011:0001:0
00003720 011:0001:5
00003730 011:0006:2
4 00003740 011:0006:4
00003750 011:0007:5
5 00003760 011:0008:2
00003770 011:000F:2
00003780 011:0010:4
3 00003790 011:0010:4
00003800 011:0013:1
00003810 011:0017:3
00003820 011:0018:0
00003830 011:0018:3
00003840 011:0019:2
5 00003850 011:0019:5
R,0003 IS SEGMENT 00012
00003860 012:0000:1
00003870 012:0007:1
00003880 012:0011:0

```

```

WRITE(USUCOD(INCRE3)), 41, DUFUSUCOD)
END
END
CLOSE(OUTXOINCR, PURGE)
LOCK(PRESJELN,*)
LOCK(USU,*)
LOCK(USUCOD,*)
END
LOCK(INCREMERGE,*)
END

```

```

N.0003(012) IS 0013 LONG
5 00003490 011:001E:2
00003900 011:0021:0
4 00003910 011:0025:5
N.0002(011) IS 0029 LONG
3 00003920 008:00CF:3
00003930 008:00D1:0
00003940 008:00D2:3
00003950 008:00D4:0
00003960 008:00D5:2
N.0001(008) IS 00CF LONG
2 00003970 003:0110:3
00003980 003:011E:5
N.0000(003) IS 0120 LONG
STACKCODE IS SEGMENT 00015
STACKCODE(015) IS 0044 LONG
DATA IS 0016 LONG

```

```

*****
NUMBER OF EXPROS DETECTED = 0,
NUMBER OF SEGMENTS = 19, TOTAL SEGMENT SIZE = 1126 WORDS, COME ESTIMATE = 26100 WORDS, STACK ESTIMATE = 89
PROGRAM SIZE = 300 CARDS, 2520 SYNTACTIC ITEMS, 111 DISK SEGMENTS,
PROGRAM FILE NAME: FUSCA,
COMPILATION TIME = 84.523 SECONDS ELAPSED; 5.233 SECONDS PROCESSING; 8.915 SECONDS I/O.
*****

```

APÊNDICE F

DIV é uma variável usada pelas funções responsáveis pelo refinamento dos perfis dos leitores.

Esta variável aparece, como vimos, tanto nas equações de incremento-decremento, como na equação de baixo incremento. Para cada caso, temos um valor específico.

A escolha do valor 8 (na figura sublinhado) para a função de incremento-decremento, deve-se ao fato de termos, com este valor, observado o comportamento desejado, ou seja, os incrementos ou decrementos sofrem as alterações mais suaves como desejamos que aconteçam.

Quanto à escolha do valor $DIV=16$ para a função de baixo incremento, prende-se ao fato de com ele observarmos um acréscimo suave aos valores iniciais.

p/ DIV=3

VALOR DE PROBABILIDADE	VALOR DE PROBABILIDADE OBTIDO PELA FUNÇÃO INCREMENTO/DECREMENTO	
	RESPOSTA RELEVANTE	RESPOSTA IRRELEVANTE
0.04	0.0703651	0.0179119
0.40	0.6261821	0.1942960
0.50	0.7500005	0.2499995
0.70	0.8579613	0.5125432
0.90	0.9546912	0.8269749

p/ DIV=5

VALOR DE PROBABILIDADE	VALOR DE PROBABILIDADE OBTIDO PELA FUNÇÃO INCREMENTO/DECREMENTO	
	RESPOSTA RELEVANTE	RESPOSTA IRRELEVANTE
0.04	0.0572575	0.0257250
0.40	0.5362335	0.2712213
0.50	0.6545088	0.3454912
0.70	0.7999489	0.5893476
0.90	0.9351956	0.8581880

p/ DIV=8

VALOR DE PROBABILIDADE	VALOR DE PROBABILIDADE OBTIDO PELA FUNÇÃO INCREMENTO/DECREMENTO	
	RESPOSTA RELEVANTE	RESPOSTA IRRELEVANTE
0.04	0.0504431	0.0307223
0.40	0.4849135	0.3180097
0.50	0.5975454	0.4024546
0.70	0.7640779	0.6317297
0.90	0.9228121	0.8746012

p/ DIV=10

VALOR DE PROBABILIDADE	VALOR DE PROBABILIDADE OBTIDO PELA FUNÇÃO INCREMENTO/DECREMENTO	
	RESPOSTA RELEVANTE	RESPOSTA IRRELEVANTE
0.04	0.0482625	0.0324834
0.40	0.4678154	0.3340570
0.50	0.5782174	0.4217826
0.70	0.75116637	0.6456515
0.90	0.9184639	0.8798803

p/ DIV=12

VALOR DE PROBABILIDADE	VALOR DE PROBABILIDADE OBTIDO PELA FUNÇÃO INCREMENTO/DECREMENTO	
	RESPOSTA RELEVANTE	RESPOSTA IRRELEVANTE
0.04	0.0468342	0.0336839
0.40	0.4564364	0.3448645
0.50	0.5652632	0.4347368
0.70	0.7432695	0.6548654
0.90	0.9155049	0.8833451

p/ DIV=16

VALOR DE PROBABILIDADE	VALOR DE PROBABILIDADE OBTIDO PELA FUNÇÃO INCREMENTO/DECREMENTO	
	RESPOSTA RELEVANTE	RESPOSTA IRRELEVANTE
0.03	0.0338231	0.0263995
0.42	0.4637524	0.3768686
0.51	0.5583300	0.4615761
0.72	0.7507358	0.6881959
0.90	0.9117390	0.8876141

APÊNDICE G

VERIFICAÇÃO DA FUNÇÃO INCREMENTO-DECREMENTO

E BAIXO INCREMENTO

A seguir, mostramos sucessivamente quatro descrições dos usuários, seus descritores e valores respectivos. Sublinhamos cinco leitores para mostrar a ação da função incremento-decremento e também sublinhamos os descritores que tiveram seus valores alterados.

Observar que para os usuários cujos números de perfis são:

- 00019
- 00108

os valores sofreram incrementos, devido a terem dado respostas relevantes.

Os de números:

- 00035
- 00078
- 00094

sofreram decrementos, devido a respostas irrelevantes.

A ação da função de baixo incremento faz-se observar no leitor de número

- 00035

que apesar de apresentar decrementos em alguns descritores existe um descritor:

. NAVAL PERSONNEL .

que apresenta o valor inicial de

- .457820

e na primeira iteração

- .504218

Isto se deve ao fato da ação da função citada, uma vez que o usuário considerou todos os documentos para ele recuperados irrelevantes.

Ao todo, temos quatro amostras, com ciclos de 5 (cinco) páginas, correspondendo a dados de cadastramento, atualização da primeira resposta, atualização da segunda resposta etc.

00019 JOSÉ IGUELMO MIRANDA

1ª execução 2ª execução 3ª execução 4ª execução

PAGINA : 1

ABYSSAL ZONES	,658740	,658740	,658740	,658740
AIRPLANE NOSES	,874500	,874500	,874500	,874500
COASTAL REGIONS	,452300	,498329	,547242	,637768
CONTROL THEORY	,658700	,658700	,658700	,658700
DOPPLER EFFECT	,452300	,452300	,452300	,452300
FLOW SEPARATION	,567800	,567800	,567800	,567800
FUSELAGES	,321450	,321450	,321450	,321450
GAS FLOW	,654320	,832944	,922522	,964714
INCOMPRESSIBLE FLOW	,785600	,785600	,785600	,785600
INJECTION	,365000	,365000	,365000	,365000
ION IMPLANTATION	,785640	,832841	,870214	,899564
LAMINAR BOUNDARY LAYER	,875600	,875600	,875600	,875600
MARINE BIOLOGY	,547800	,547800	,547800	,547800
OCEANOGRAPHIC DATA	,523400	,523400	,523400	,523400
SEMICONDUCTOR DOPING	,784560	,784560	,784560	,784560
SHOCK WAVES	,234500	,234500	,234500	,234500
SUPERSONIC AIRCRAFT	,754800	,754800	,754800	,754800
SUPERSONIC CHARACTERISTICS	,475800	,475800	,475800	,475800
WALLS	,876500	,904481	,926299	,943237
WIND TUNNEL TESTS	,568400	,568400	,568400	,568400

00035 AURELIO GOMES FERREIRA

1ª 2ª 3ª 4ª
execução execução execução execução

PAGINA : 2

ABYSSAL ZONES	,658740	,658740	,658740	,658740
COASTAL REGIONS	,856740	,821117	,777447	,724397
COMBAT READINESS	,857400	,857400	,857400	,857400
DEPARTMENT OF DEFENSE	,800000	,800000	,800000	,800000
DESTROYERS	,856320	,856320	,856320	,856320
MAGNETOHYDRODYNAMIC GENERATORS	,875420	,875420	,875420	,875420
MARINE BIOLOGY	,457860	,457860	,457860	,457860
MARINE PROPULSION	,874520	,874520	,874520	,874520
MEASURING INSTRUMENTS	,658740	,658740	,658740	,658740
MILITARY PROCUREMENT	,875460	,844202	,805707	,758665
NAVAL EQUIPMENT	,567400	,567400	,567400	,567400
NAVAL PERSONNEL	,457820	,504218	,407113	,449993
OCEAN SURFACE	,875420	,875420	,875420	,875420
OCEANOGRAPHIC DATA	,875420	,875420	,875420	,875420
PORTABLE EQUIPMENT	,574200	,574200	,574200	,574200
PROPELLERS (MARINES)	,475820	,475820	,475820	,475820
SHIP MOTIONS	,567840	,567840	,567840	,567840
SHIP STRUCTURAL COMPONENTS	,142570	,142570	,142570	,142570
THERMODYNAMICS	,600000	,515087	,419156	,462847
TRAINING	,500000	,500000	,500000	,500000

00078 CELIA LEMOS DE ALMEIDA

1ª 2ª 3ª 4ª
execução execução execução execução

COMMERCE	,875400	,875400	,875400	,875400
COMPUTER PROGRAMS	,689570	,689570	,689570	,689570
CORRELATION TECHNIQUES	,685740	,685740	,685740	,685740
DATA BASES	,568740	,568740	,568740	,568740
DECISION MAKING	,685700	,685700	,685700	,685700
DECISION THEORY	,685700	,685700	,685700	,685700
ECONOMICS	,500000	,500000	,500000	,500000
ECONOMICS MODELS	,658700	,658700	,658700	,658700
FINANCE	,658740	,583164	,495821	,544588
GAME THEORY	,685700	,685700	,685700	,685700
INDUSTRIAL EQUIPMENT	,657800	,657800	,657800	,657800
INFORMATION PROCESSING	,568740	,568740	,568740	,568740
INFORMATION RETRIEVAL	,600000	,600000	,600000	,600000
INVENTORY CONTROL	,475800	,381772	,302886	,238263
MAINTAINABILITY	,685300	,614360	,531607	,437570
MAINTENANCE MANAGEMENT	,653240	,653240	,653240	,653240
MARKETING	,685740	,685740	,685740	,685740
MAROV PROCESSES	,587460	,587460	,587460	,587460
STATISTICAL ANALYSIS	,685320	,685320	,685320	,685320
STOCHASTICS PROCESSES	,560000	,560000	,560000	,560000

ARMS CONTROL	.574230	.574230	.574230	.574230
DEPARTMENT OF DEFENSE	.874560	.874560	.874560	.874560
DOPPLER SYSTEMS	.685470	.685470	.685470	.685470
FOURIER TRANSFORMATION	.685470	.685470	.685470	.685470
GAS FLOW	.800000	.800000	.623937	.359880
LAND USE	.658740	.658740	.658740	.658740
LASERS	.587460	.587460	.587460	.587460
MAGNETOHYDRODYNAMIC GENERATORS	.587460	.587460	.587460	.587460
MARINE BIOLOGY	.700000	.700000	.700000	.700000
MILITARY PROCUREMENT	.685470	.685470	.614560	.531839
NAVAL EQUIPMENT	.875400	.875400	.875400	.875400
OCEANOGRAPHIC DATA	.500000	.500000	.500000	.500000
PARTIAL DIFFERENTIAL EQUATIONS	.854700	.854700	.854700	.854700
PARTICLE TRAJECTORIES	.658740	.658740	.658740	.658740
RELIABILITY	.854700	.854700	.854700	.818607
SHIP MOTION	.856740	.856740	.856740	.856740
SHIP STRUCTURAL COMPONENTS	.658740	.658740	.658740	.658740
SOILS	.685470	.685470	.614560	.531839
WASTE DISPOSAL	.500000	.549009	.457101	.503451
WIND TUNNEL TESTS	.800000	.800000	.800000	.800000

PAGINA : 5

ANTENNA ARRAYS	.658740	.658740	.658740	.658740
COMMUNICATIONS NETWORKS	.500000	.500000	.500000	.500000
DIGITAL SYSTEMS	.874560	.874560	.874560	.874560
ELECTRON BEAMS	.500000	.549009	.714471	.864027
GATES	.658740	.658740	.658740	.658740
INTEGRATED CIRCUITS	.658740	.658740	.730462	.835264
LIGHT EMITTING DIODES	.658740	.658740	.658740	.658740
LOGIC CIRCUITS	.685740	.685740	.752498	.849308
MESSAGE PROCESSING	.600000	.600000	.681990	.749446
MICROPHONE CIRCUITS	.500000	.500000	.500000	.500000
MICROWAVES	.658740	.658740	.658740	.658740
MONOPOLE ANTENNAS	.700000	.700000	.764078	.815648
PHASED ARRAYS	.658740	.658740	.658740	.658740
PULSED LASERS	.658740	.658740	.658740	.658740
SEMICONDUCTOR DEVICES	.658740	.658740	.730462	.835264
TACTICAL COMMUNICATION	.700000	.700000	.700000	.700000
THIN FILMS	.658740	.658740	.658740	.658740
TUNNELING (ELECTRONICS)	.658740	.658740	.658740	.658740
ULTRASONIC WELDING	.875460	.875460	.875460	.875460
WAVEGUIDES	.658740	.658740	.658740	.658740