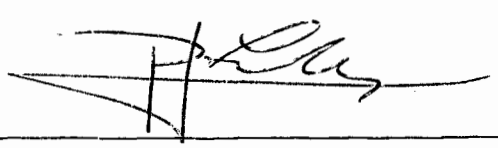
UM SISTEMA AUTOMÁTICO PARA COMPRESSÃO DE DADOS

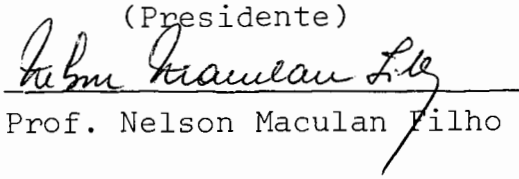
Maurilio Gonçalves dos Santos Filho

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

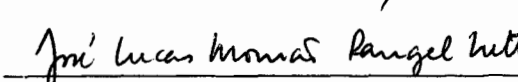
Aprovada por:



Prof. Pierre Jean Lavelle
(Presidente)



Prof. Nelson Maculan Filho



Prof. José Lucas M. Rangel Netto

RIO DE JANEIRO, BRASIL
JUNHO DE 1978

GONÇALVES DOS SANTOS FILHO, MAURILIO

Um Sistema Automático para Compressão de Dados |Rio de Janeiro| 1978

xii, 398 p. 29,7cm (COPPE-UFRJ, M.Sc., Engenharia de Sistemas e Computação, 1978)

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

1. Compressão de Dados I. COPPE/UFRJ II. Um Sistema Automático para Compressão de Dados.

A minha família

AGRADECIMENTOS

Ao professor Pierre Jean Lavelle pela orientação paciente e precisa deste trabalho.

A meus pais Maurilio Gonçalves dos Santos e Elita Rios dos Santos que com seus esforços sempre me incentivaram a continuar.

A Companhia de Eletricidade do Ceará nas pessoas do professor Jesamar Leão de Oliveira e Cleto Prata Crisóstomo que tão bem souberam entender esta oportunidade profissional.

Aos amigos do Departamento de Processamento de Dados da COELCE pelo incentivo e apoio prestados no transcorrer de todo o trabalho.

RESUMO

Como sabemos, um dos fatos mais comuns em todo Centro de Processamento de Dados, é o grande volume de informações que certas aplicações requerem para seu funcionamento. Estas informações agrupadas em um ou mais arquivos, necessitam normalmente de uma grande disponibilidade de meios de armazenamento, trazendo como consequências imediatas, aumento nos custos do sistema e maior complexidade no processamento.

O trabalho aqui apresentado trata de um Sistema Automático para Compressão de Dados aplicado a arquivos organizados sequencialmente, independente do meio físico de armazenamento.

Este sistema, que utiliza-se das técnicas de segmentação e códigos de tamanho variável, em particular o código HSF (Huffman-Shannon-Fano), está constituído basicamente por um grupo de programas utilitários e uma subrotina de compressão e descompressão.

A finalidade principal dos programas utilitários é fornecer informações sobre os dados a serem trabalhados, de modo a permitir que o usuário com o seu manuseio adequado venha a obter um alto índice de compressão.

Já à subrotina caberá a responsabilidade da compressão e descompressão efetiva destes dados, o que se obtem pela substituição dos comandos de leitura e gravação nos programas envolvidos por chamadas a estas rotinas.

ABSTRACT

It is a very common fact, in every Data Processing Center, that some applications require a high volume of informations for a well development.

These informations, groupped into one or more files, need, usually, many storage device and because of this costs and complexity of processing became higher.

This work presents an Automatic System for Data Compression applied for sequential files, independent of the storage device used.

The system using technic of segmentation and variable lenght fields and particulary the HSF code (Huffman-Shannon-Fano), is a combination of a group of utilities programs and a pair of subroutines of compression and descompression.

The main finality of the utilities programs is to give information about the data to be elaborated so as to alow that the user using them correctly obtain a high degries of compression.

The subroutines will do the compression and descompression of these data by substitution of the read-write statements in the programs involved, by calls and these routines.

ÍNDICE

Páginas

INTRODUÇÃO

1. MOTIVOS DA PESQUISA

A idéia de desenvolver um trabalho desta natureza, remonta a algum tempo, desde que em nosso Centro de Processamento de Dados sempre enfrentamos sérias dificuldades com a falta de área para armazenamento de nossos arquivos, os quais apresentam como característica o fato de sempre aumentarem o seu volume, embora que à uma taxa de crescimento relativamente pequena.

Em vista destes fatos, várias foram as soluções adotadas, que com o passar do tempo demonstraram ser ineficientes, pois as reduções de volume conseguidas, eram rapidamente absorvidas pelo constante aumento dos arquivos, o que nos levava a novas alterações. A repetição deste ciclo, como não poderia deixar de ser, ia aos poucos repercutindo no sistema como um todo, uma vez que a cada alteração:

- novos programas eram adicionados;
- programas existentes eram modificados, acarretando muitas vezes degradação de performance;
- novos procedimentos de operação eram criados, contribuindo assim para o aumento da taxa de erros dos operadores.

Toda esta problemática era agravada, quando éramos obrigados a classificar estes arquivos, pois as necessidades de áreas de trabalho nem sempre eram atendidas completamente, o que acarretava, além do que foi anteriormente citado, o aumento no tempo de execução destes programas.

Diante dos fatos, a única alternativa que nos restou foi a de se estudar o conjunto de técnicas que se oferecia através da Compressão de Dados.

2. O QUE É A COMPRESSÃO DE DADOS

Por Compressão de Dados entendemos uma ou um conjunto de técnicas, que se aplicadas a uma dada informação nos

permite reduzir o espaço útil necessário para o seu armazenamento.

Podemos agrupá-las em 2 categorias:

a) aquelas que dependem da estrutura do registro ou conteúdo dos dados e que por esta razão são geralmente escritas para uma aplicação específica;

b) aquelas de aplicação geral, que podem ser implantadas em software, hardware ou micro-código.

Da mesma forma duas são as áreas de aplicação existentes:

a) aplicação das técnicas de compressão para reduzir o volume de informações nos arquivos e possivelmente obter um maior rendimento dos programas de aplicação, visto que a informação compressa pela sua densidade, permite uma maior taxa de transferência do canal assim como um maior volume de informações nas áreas de entrada/saída. Para os programas de classificação, consegue-se uma maior eficiência, uma vez que com os dados compressos as áreas de trabalho passam a ter maior capacidade de armazenamento;

b) aplicação das técnicas à transmissão de dados, pois transmitindo-se dados compressos consegue-se dobrar em muitos casos a velocidade efetiva da linha de comunicação.

Por tudo que foi exposto, podemos deduzir, que haverá uma redução razoável nos custos dos sistemas, uma vez que a Compressão de Dados, pelo possível aumento na performance dos programas, diminuição no volume dos arquivos, otimização das linhas de comunicação, etc., implicará numa redução no tempo de processamento, utilização de memória auxiliar e utilização das linhas de comunicação.

Finalmente, um arquivo compresso é um arquivo ilegível aos estranhos, ou seja, ninguém poderá ter facilmente acesso ao seu conteúdo, uma vez que será impossível identificar qualquer informação no mesmo sem a(s) chave(s) necessária(s), assegurando desta maneira uma melhor privacidade dos dados.

Por outro lado alguns aspectos devem ser destacados, por se constituirem decisivos no emprego da Compressão de Dados em qualquer instalação:

- Aumento na utilização da CPU - talvez o ponto mais crítico em um empreendimento desta natureza, deve ser cuidadosamente avaliado, para que surpresas desagradáveis não venham a ocorrer, pois como sabemos, durante o processo de compressão e descompressão, tempo de CPU suplementar é consumido, podendo, dependendo dos métodos empregados, degradar a performance de um programa a tal ponto que desaconselharia totalmente a sua utilização, se esse fosse já o ponto de estrangulamento do sistema.

- Utilitários - uma prática muito comum em toda instalação é o emprego de utilitários para retirar, alterar ou incluir dados em arquivos e uma vez que pela compressão o conteúdo destes arquivos passa a ser ilegível para procedimentos clássicos, podemos deduzir ser impossível a utilização de tais práticas, a não ser que novos utilitários fossem desenvolvidos com as respectivas rotinas de compressão e descompressão.

Assim o emprego da Compressão de Dados em uma instalação se justificará, desde que todas as implicações de sua utilização sejam exaustivamente avaliadas, evitando desta maneira os dissabores de maus resultados.

3. OBJETIVOS DA PESQUISA

O nosso objetivo básico, quando do desenvolvimento deste trabalho, foi o de mostrar ser possível construir um Sistema de Compressão de Dados, que pudesse ser aplicado de forma generalizada a arquivos organizados sequencialmente.

Para isto, nos propomos aqui, apresentar um roteiro, em que todas as etapas do trabalho são minuciosamente detalhadas, como os estudos preliminares, tarefas realizadas, estruturação dos programas, técnicas desenvolvidas, etc. Esperamos assim, contribuir de forma definitiva para que novos trabalhos sejam desenvolvidos nesta área, que julgamos ser de relevada importância para o processamento de dados de modo geral.

CAPÍTULO I

DEFINIÇÕES

1. ESTRUTURA DOS CAMPOS

Definimos por Estrutura dos Campos, para um dado registro, cada uma das combinações possíveis de preenchimento destes campos. Convencionamos ainda que a presença de um campo na Estrutura será indicado pelo dígito 1 e sua ausência pelo 0.

Para melhor compreensão, seja um registro conforme o lay-out abaixo:

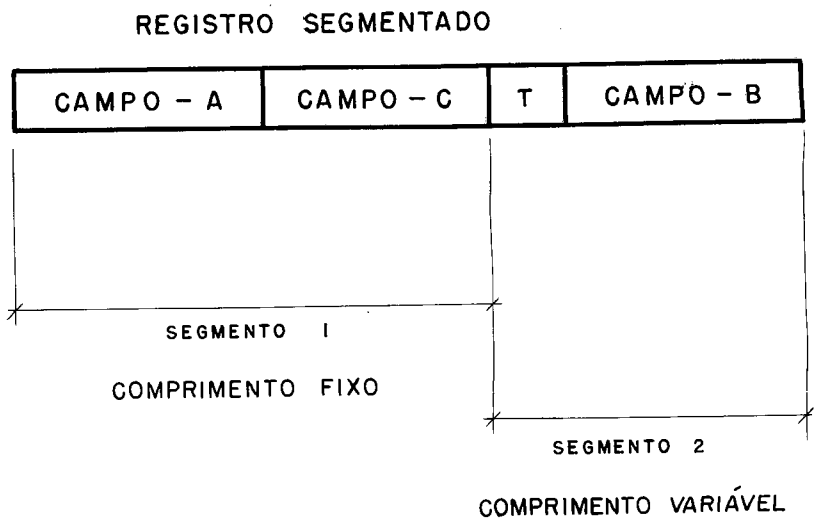
AS ESTRUTURAS POSSÍVEIS PARA O EXEMPLO SÃO:

EST.	CAMPO-A	CAMPO-B	CAMPO-C
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

2. SEGMENTO

É o conjunto de um ou mais campos adjacentes em um registro, que gozam de certas relações lógicas e possuem características comuns como formato, ocorrências no registro, etc. Estes Segmentos podem ser de comprimento fixo ou variável. Quando variável devem ser precedidos de um campo (1 ou mais bytes) indicativo de seu tamanho (T).

Segmentando o registro do exemplo anterior, poderíamos ter:



I-2

3. GRUPO

É toda e qualquer configuração binária utilizada para representar um dado caráter, que dependendo do código escolhido pode ser de tamanho (número de bits) fixo ou variável.

Por exemplo o EBCDIC (Extended Binary Coded Decimal Interchange Code) é um código onde os grupos são de tamanho fixo de 8 bits:

A - 1100 0000
0 - 1111 0000

Por outro lado, Shannon-Fano, Huffman, e HSF são códigos onde o número de bits das configurações variam de acordo com a Probabilidade de Ocorrência do carater na amostra analisada.

4. FREQUÊNCIA ABSOLUTA

É o número de ocorrências de um determinado fato na amostra.

No decorrer deste trabalho analisaremos as Frequências Absolutas de estruturas, campos, segmentos e caracteres.

Por exemplo, seja determinar as Frequências Absolutas de ocorrência dos campos de um registro em uma dada amostra de um arquivo:

- Lay-out do registro

CAMPO - A	CAMPO - B	CAMPO - C

1 - 3

- Amostra analisada - 100 registros
- Ocorrências do campo A-50
- Ocorrências do campo B-60
- Ocorrências do campo C-100

Podemos então afirmar que as Frequências Absolutas de ocorrência dos campos A, B e C são respectivamente 50, 60 e 100.

5. PROBABILIDADE

E a possibilidade de que um certo fato aconteça, determinada pela razão entre o número de casos favoráveis e o número total de casos possíveis. Se N é o número total de casos possíveis de um evento sob determinadas condições e M o número de casos conhecidos como o evento A, a probabilidade do evento A sob estas condições M/N.

No nosso exemplo a Probabilidade de na amostra

- ocorrer o campo A é 50/100 = 0,5
- ocorrer o campo B é 60/100 = 0,6
- ocorrer o campo C é 100/100 = 1,0

onde 100 é o número de registros analisados.

6. CUSTO

Definimos por Custo de um código, o número médio de dígitos binários necessários para representar um caráter nesse código.

Suponhamos que ao se analisar os dados de um arquivo, obtivemos:

- $C_1, C_2, C_3, \ldots, C_k$ - caracteres encontrados na amostra, também chamado de Alfabeto.
- $T_1, T_2, T_3, \ldots, T_k$ - tamanho em bits das configurações binárias de cada carater na amostra.
- $P_1, P_2, P_3, \ldots, P_k$ - probabilidade de ocorrência de cada caráter na amostra.

Assim,

$$(\text{CUSTO MÉDIO})_{\text{Código}} = P_1 \times T_1 + P_2 \times T_2 + P_3 \times T_3 + \ldots + P_k \times T_k = \sum_{j=1}^{k} P_j \times T_j$$

Por exemplo, seja o Alfabeto abaixo:

ALF.	PROBAB. (Pj)	TAM. COD. (Tj)	CUSTO (PJ x TJ)	COD. HSF
2	0.226	2	0.452	00
9	0.165	3	0.495	010
4	0.135	3	0.405	011
0	0.120	3	0.360	100
A	0.079	4	0.316	1010
8	0.063	4	0.252	1011
3	0.054	4	0.216	1100
7	0.041	4	0.164	1101
1	0.038	5	0.190	11100
5	0.034	5	0.170	11101
B	0.030	5	0.150	11110
6	0.015	5	0.075	11111

$(\text{CUSTO MÉDIO})_{\text{HSF}} = 3{,}245$ bits/caráter.

Observando-se a tabela acima, temos:

- Alfabeto (ALF)
- Probabilidade de ocorrência de cada caráter do alfabeto na amostra (Pj)
- Tamanho em bits do código HSF, representado na última coluna (Tj)
- Custo médio de cada carater (Pj x Tj)
- Código HSF, que será empregado na representação dos caracteres. (COD. HSF)

Como é do nosso conhecimento, qualquer símbolo do alfabeto acima codificado em EBCDIC, necessitaria de 8 bits, enquanto pelo código HSF apenas 3,245 bits/caráter (custo médio do código) seriam precisos, ou seja, uma compressão de

$$\frac{8 - 3{,}245}{8} \times 100 = \frac{4{,}755}{8} \times 100 = 59{,}43\%.$$

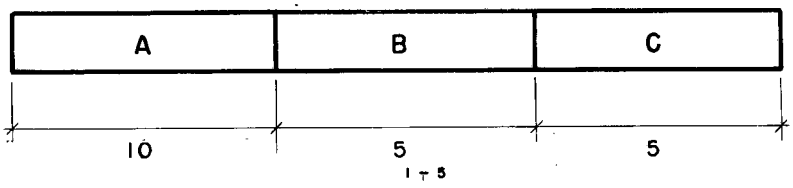
7. COMPRIMENTO ORIGINAL

Definimos por Comprimento Original, o número que expresso em bytes ou bits, traduz o tamanho inicial de uma estrutura, campo, arquivo (amostra) ou caráter.

Assim, no exemplo teríamos:

- campo A - Comprimento Original - 10 bytes
- campo B - Comprimento Original - 5 bytes
- campo C - Comprimento Original - 5 bytes

E uma vez que nossa amostra é de 100 registros e que cada registro tem 20 bytes de tamanho, o Comprimento Original da amostra é de 2000 bytes.



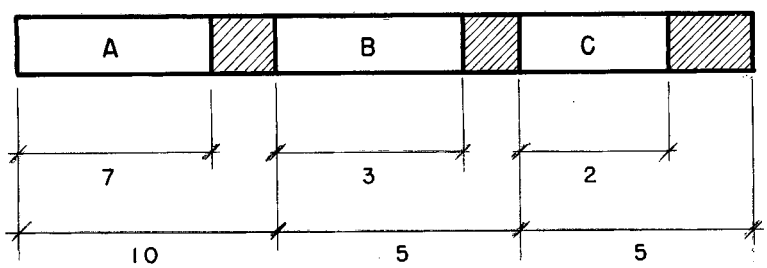
8. COMPRIMENTO COMPRESSO

Definimos por Comprimento Compresso, o número que expresso em bytes ou bits, traduz o tamanho final (após a compressão) de uma estrutura, campo, arquivo (amostra) ou caráter.

Suponhamos, para exemplificar, que após a compressão do arquivo os campos A, B e C passaram a ter os comprimentos abaixo em todos os registros:

- campo A - 7 bytes
- campo B - 3 bytes
- campo C - 2 bytes

Assim 7, 3, 2 bytes correspondem exatamente aos Comprimentos Compressos dos campos A, B, C.



I - 6

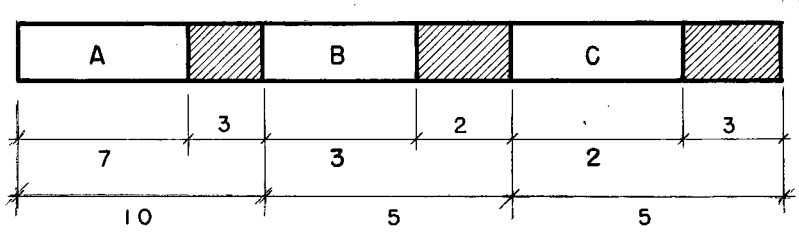
9. COMPRESSÃO ABSOLUTA

Medida expressa em bytes ou bits da redução alcançada pela compressão de uma estrutura, campo, arquivo ou caráter.

(COMPRESSÃO ABSOLUTA) = (COMPRIMENTO ORIGINAL) - (COMPRIMENTO COMPRESSO)
(CA) (CO) (CC)

Onde CA $\geq$ 0

No exemplo:



I - 7

- Compressão Absoluta do campo A = 10 - 7 = 3 bytes
- Compressão Absoluta do campo B = 5 - 3 = 2 bytes
- Compressão Absoluta do campo C = 5 - 2 = 3 bytes

10. COMPRESSÃO RELATIVA

A Compressão Absoluta não expressa de uma maneira real a redução obtida, pois o mesmo resultado pode ser conseguido a partir de amostras que possuam Comprimento Original e Compressos diferentes. Por exemplo, a Compressão Absoluta do campo A = 10 - 7 = 3 bytes poderia ser igual a Compressão Absoluta de um campo X = 100 - 97 = 3 bytes, embora a redução de 3 bytes em 10 no campo A seja muito mais significativa. Por esta razão, ficou estabelecido o conceito de Compressão Relativa, que relaciona a Compressão Absoluta de uma certa amostra ao Comprimento Original da mesma ou de outra amostra com a qual desejamos comparar.

$$(\text{COMPRESSÃO RELATIVA})_{(CR)} = \frac{(\text{COMP.ORIGINAL})_{(CO)} - (\text{COMP.COMPRESSO})_{(CC)}}{(\text{COMPRIMENTO ORIGINAL})_{(CO)}}$$

onde

$0 \leqslant CR < 1 \qquad CC > 0$

Exemplos:

a) A Compressão Relativa do campo A seria:

$$\frac{10 - 7}{10} = 0,3$$

ou seja, obteríamos uma redução de 0,3 bytes por byte do campo A.

b) A Compressão Relativa do campo A na amostra em relação a amostra do arquivo (100 registros) seria:

- Comprimento Original do campo A na amostra 10x100 = 1000 bytes.
- Comprimento Compresso do campo A na amostra 7x100 = 700 bytes.
- Compressão Relativa = $\frac{1000 - 700}{100}$ = 3 bytes/registro.

11. COMPRESSÃO PERCENTUAL

A Compressão Percentual expressa em percentagem a Compressão Relativa, desde que haja homogeneidade de unidade (bits/bit, bytes/byte, etc.).

$$\underset{(CP)}{(\text{COMPRESSÃO PERCENTUAL})} = \underset{(CR)}{(\text{COMPRESSÃO RELATIVA})} \times 100$$

onde

$$0 \leq CP < 100$$

12. RAIO DE COMPRESSÃO

É o número que revela de quantas vezes a informação original foi reduzida.

$$\underset{(RC)}{(\text{RAIO DE COMPRESSÃO})}_{\text{amostra}} = \frac{\overset{(CO)}{(\text{COMPRIMENTO ORIGINAL})}}{\underset{(CC)}{(\text{COMPRIMENTO COMPRESSO})}}_{\text{amostra}}^{\text{amostra}}$$

onde $CC > 0$

Tomando-se os campos A, B e C do nosso exemplo, teríamos os seguintes Raios de Compressão:

- Campo A - 10/7 = 1,428
- Campo B - 5/3 = 1,666
- Campo C - 5/2 = 2,500

ou seja, os campos A, B e C tiveram os seus comprimentos reduzidos de 1,428, 1,666 e 2,500 respectivamente.

CAPÍTULO II

MÉTODOS PARA COMPRESSÃO DE DADOS

1. APRESENTAÇÃO

Neste capítulo, de conformidade com nossos objetivos, apresentaremos todos os métodos estudados, de uma forma a permitir que o leitor venha ter uma visão global da compressão de dados com suas técnicas, nomenclatura, vantagens e desvantagens.

Para a exposição, os métodos foram agrupados segundo a classificação abaixo:

- Métodos de aplicação específica
  . Aqueles que dependem do conteúdo dos dados
  . Aqueles que dependem da estrutura do registro
- Métodos de aplicação geral

Para cada um deles foi preparado:

- Uma descrição sumária
- Sistemática de implantação
  . Algoritmos e/ou exemplos
- Vantagens
- Desvantagens.

2. MÉTODOS DE APLICAÇÃO ESPECÍFICA QUE DEPENDEM DO CONTEÚDO DOS DADOS

2.1. COMPRESSÃO PELA ELIMINAÇÃO DE ÍTENS REDUNDANTES

DESCRIÇÃO

Um importante método para a redução do volume de dados em um arquivo é a eliminação da redundância que existe devido ao múltiplo armazenamento de dados e/ou seus derivados em registros ou arquivos de um sistema.

SISTEMÁTICA DE IMPLANTAÇÃO

a) Da análise de um sistema de pagamento, observou-se que o salário do empregado encontrava-se armazenado em 4 arquivos distintos, onde:

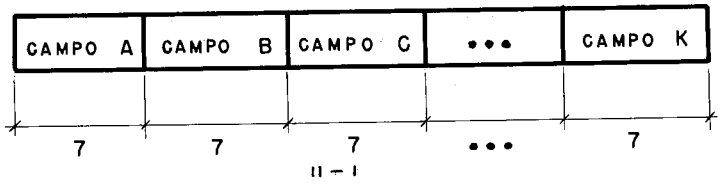
- O arquivo A continha 3000 registros
- O arquivo B continha 2500 registros
- O arquivo C continha 4000 registros.

Se o campo de salário que ocupa 7 bytes fosse eliminado:

- Do arquivo A, redução obtida 21000 bytes.
- Do arquivo B, redução obtida 17500 bytes.
- Do arquivo C, redução obtida 28000 bytes.

Redução total 66500 bytes.

b) Seja o registro, cujo lay-out se encontra abaixo:



Onde os conteúdos dos campos A, B e C obedecem as seguintes relações:

$B = 0,7 \times A$

$C = 0,4 \times A$

Ora, os campos B e C podem ser eliminados dos registros, resultando daí uma economia de 14 bytes/registro e tudo isto por um pequeno acréscimo de processamento nos programas para cálculo dos campos "B" e "C".

VANTAGENS

a) Proporciona uma redução razoável no volume de informações dos arquivos envolvidos.

b) Reduz as possibilidades de erros, pois a multiplicidade dos dados, normalmente implica que estes mesmos dados cheguem ao sistema por vias diferentes.

c) Aumenta a performance dos programas que atualizam estes arquivos, uma vez que a eliminação de certos itens implica numa redução das atualizações.

DESVANTAGENS

a) Diminui a segurança do sistema, pela limitação de cópias de certos dados.

b) Aumento da complexidade do sistema, uma vez que nem sempre teremos uma cópia do dado à nossa disposição na hora e no arquivo oportuno.

2.2. COMPRESSÃO DE DADOS ESPECIAIS

DESCRIÇÃO

Sempre que os dados são armazenados de uma forma facilmente legível, é frequente eles conterem mais caracteres que o necessário para o seu armazenamento de uma forma mais complexa.

Por esta razão, para certos tipos de dados como datas, endereços, campos numéricos, etc., existem maneiras compactas de representá-los.

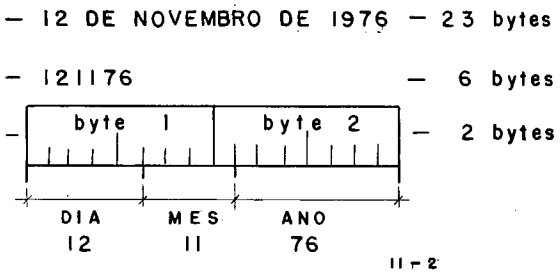
SISTEMÁTICA DE IMPLANTAÇÃO

a) Um campo numérico armazenado no formato decimal zonado, necessita de 1 byte para cada dígito, enquanto no formato decimal compactado, apenas 1 byte representa dois dígitos exceto o byte de sinal que comporta apenas um dígito. Se por outro lado utilizamos o formato binário, uma maior redução poderá ser conseguida. Para melhor exemplificar, se armazenarmos o número 60501 no:

   - formato decimal zonado necessitaremos de 5 bytes;
   - formato decimal compactado necessitaremos de 3 bytes;
   - formato binário necessitaremos de 2 bytes.

b) Como dissemos anteriormente, existem certos tipos de dados que podem ser expressos de uma maneira mais compacta. Pa-

ra isto, vejamos as diversas formas de representar uma DATA.



VANTAGENS

a) Redução razoável nas necessidades de espaço para a informação.

b) Redução nas necessidades de tempo de CPU, pela redução ou eliminação das conversões de formato quando operando com campos numéricos, uma vez que para reduzir os seus tamanhos, temos que defini-los com formato binário ou decimal compactado.

DESVANTAGENS

a) Tornar menos aparente o conteúdo dos campos.

2.3. SUPRESSÃO DE CARACTERES REPETIDOS

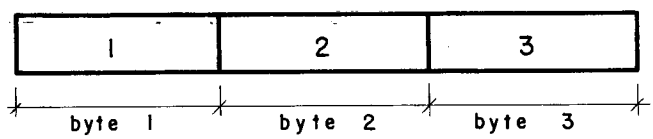
DESCRIÇÃO

Este método tem por objetivo eliminar ou reduzir sequências de zeros, brancos ou de outros caracteres, substituindo-os por um grupo de símbolos (2 ou 3 bytes), que permitem posteriormente a obtenção da sequência original.

Um primeiro esquema para supressão de caracteres repetidos, consiste em eliminar brancos à direita de campos alfabéticos ou alfanuméricos e zeros a esquerda de campos numéricos, tornando o campo em questão menor e variável em tamanho.

II-3

Como segundo esquema, a sequência de caracteres é substituída por um grupo de 3 bytes com as seguintes funções:



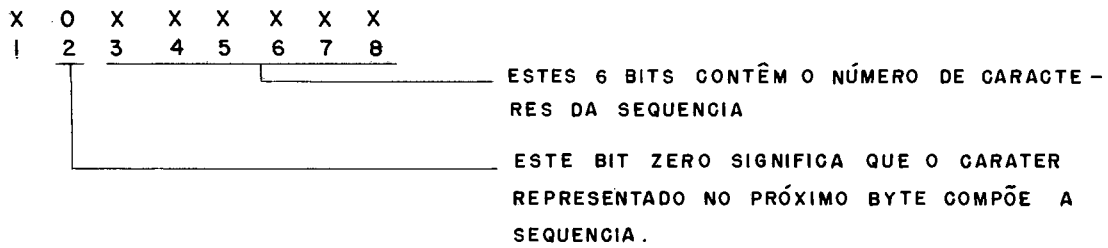
II-4

- Byte 1 - indica a compressão de uma sequência de caracteres.
- Byte 2 - indica o caráter que compõe a sequência.
- Byte 3 - contém o número de caracteres da sequência, no máximo 256, uma vez que aqui somente um byte foi empregado para esta informação.

O terceiro esquema é basicamente uma evolução do segundo e agora a sequência é substituída por um grupo de 2 bytes com as seguintes funções:

- Byte 1 - Chamado CARATER SENTINELA, indica a compressão de uma sequência e que esta é composta de um dado caráter. Para cada sequência diferente, devemos observar que, o Caráter Sentinela deve ser criteriosamente escolhido de modo a evitar que o mesmo participe dos dados.
- Byte 2 - Contém o número de caracteres da sequência, no máximo 256,por motivos já assinalados.

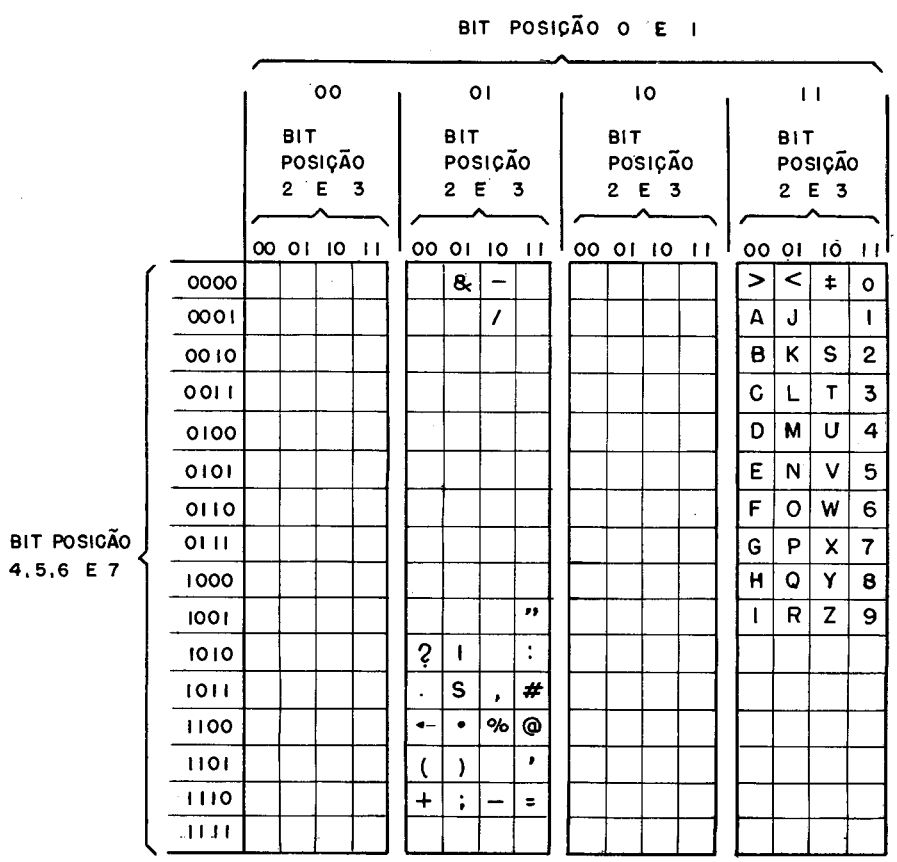
Se olharmos atentamente o código EBCDIC na tabela a seguir, vamos ver que a maioria das combinações não são usualmente empregadas na representação de dados, principalmente aquelas com o bit 2 zerado.



II-5

Do exposto uma alternativa se apresenta para o terceiro esquema:

- Byte 1 - (Caráter Sentinela) - É uma combinação de bits onde o 2º bit zero (X0XX XXXX), indica a compressão de uma sequência e os 6 bits restantes (bit 3 ao bit 8)contém o número de caracteres, no máximo 64.
- Byte 2 - Fornece o caráter para a reconstituição da sequência original.



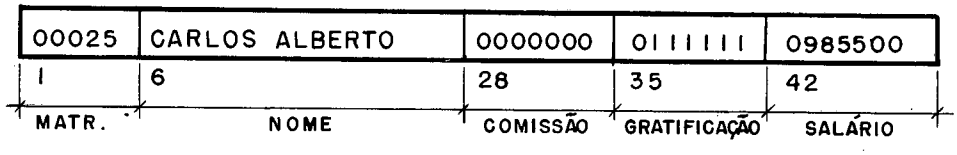
II-6

Para certas aplicações, onde as sequências são compostas de um único caráter pré-determinado o byte 2 na alternativa acima poderá ser suprimido.

SISTEMÁTICA DE IMPLANTAÇÃO

De um cadastro de pessoal de uma empresa se obteve o registro cujo lay-out e conteúdo estão abaixo:

POSIÇÃO	DESCRIÇÃO	COMPRIMENTO
1 - 5	MATRICULA DO EMPREGADO	= 5 bytes
6 - 27	NOME DO EMPREGADO	= 22 bytes
28 - 34	COMISSÃO	= 7 bytes
35 - 41	GRATIFICAÇÃO	= 7 bytes
42 - 48	SALÁRIO	= 7 bytes

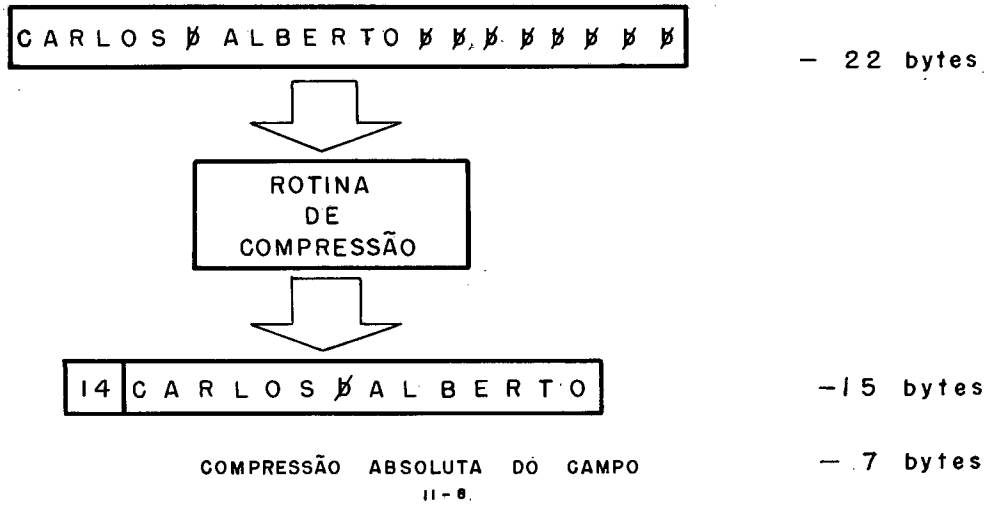


II-7

a) Primeiro esquema:

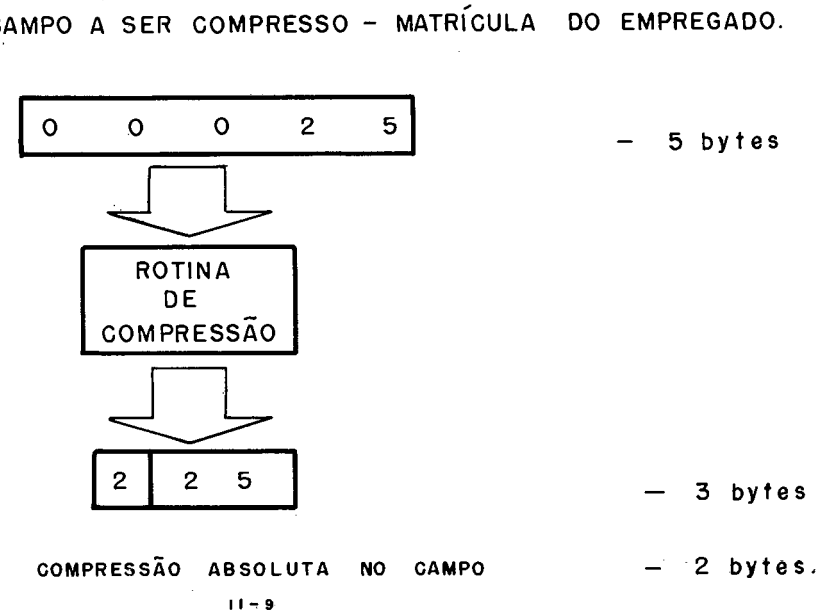
Eliminação de campos à direita de campos alfabéticos ou alnuméricos.

CAMPO A SER COMPRESSO - NOME DO EMPREGADO



II-8

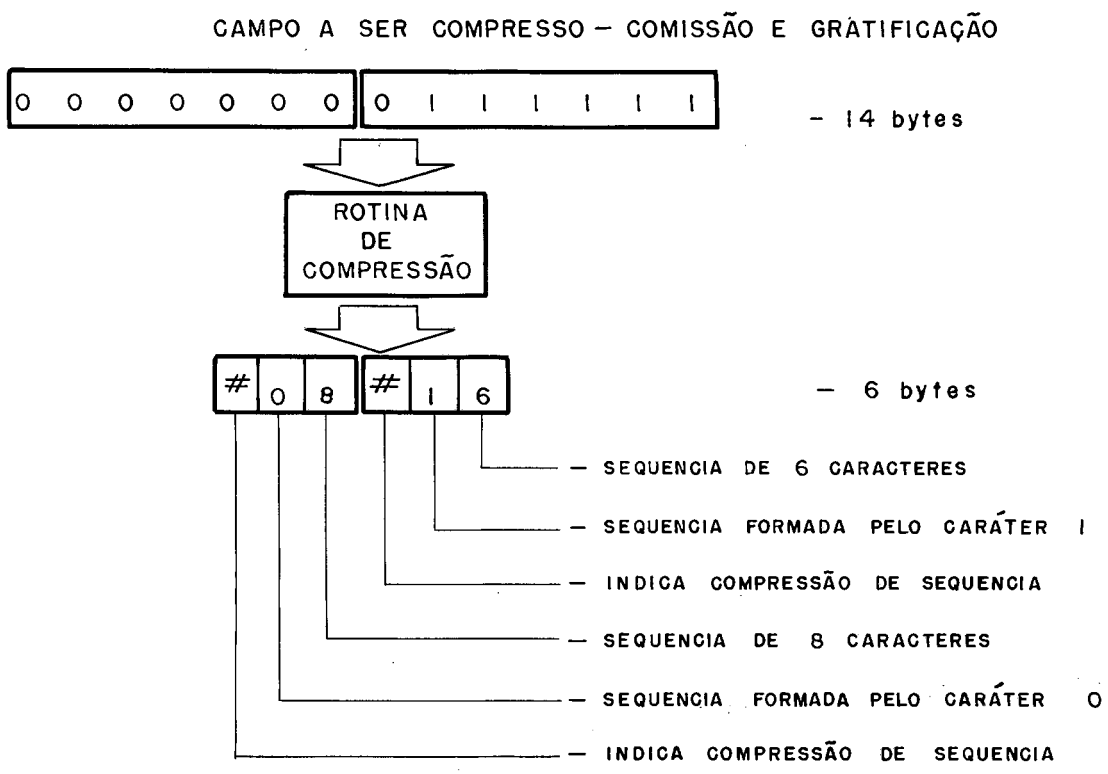
Eliminação de zeros a esquerda de campos numéricos.



II-9

b) Segundo esquema:

Compressão de uma sequência de caracteres identicos pela sua substituição por 3 símbolos.

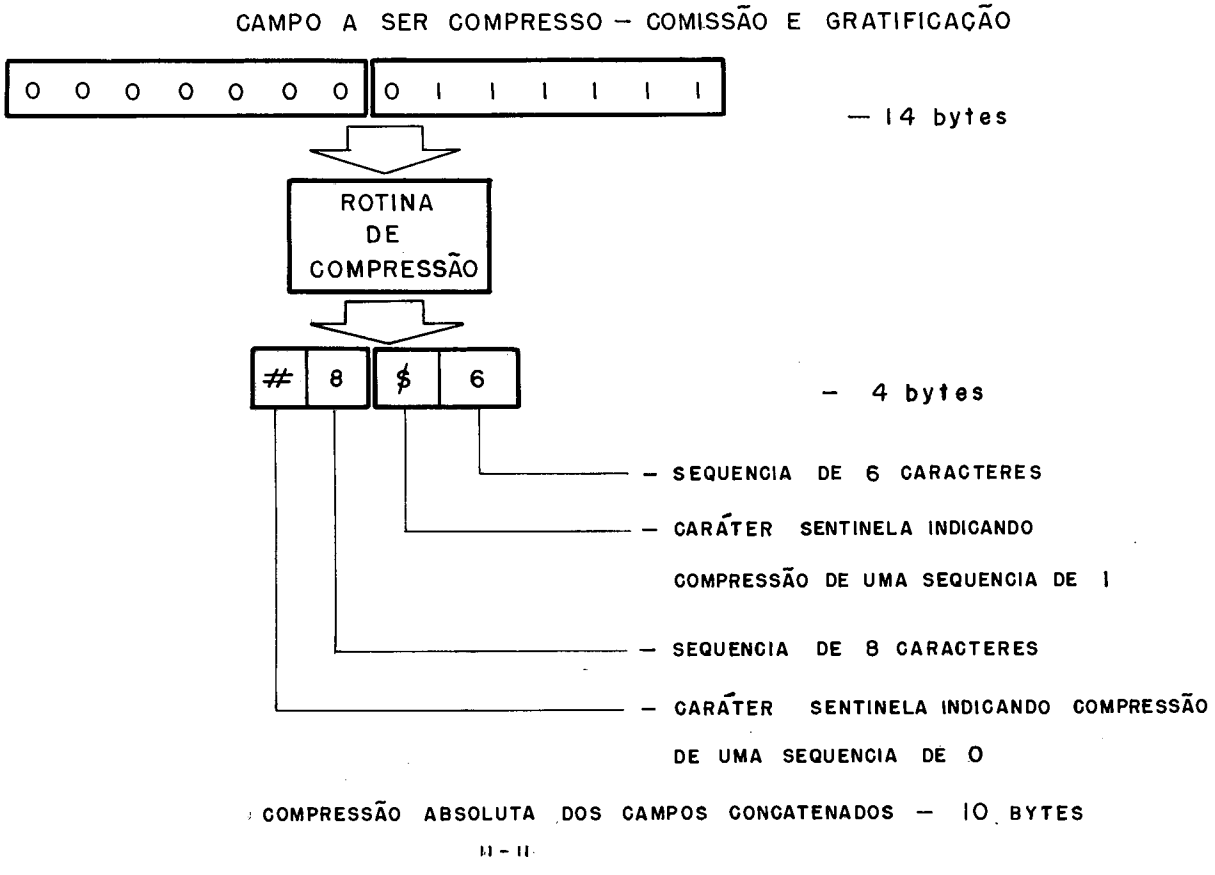


II-10

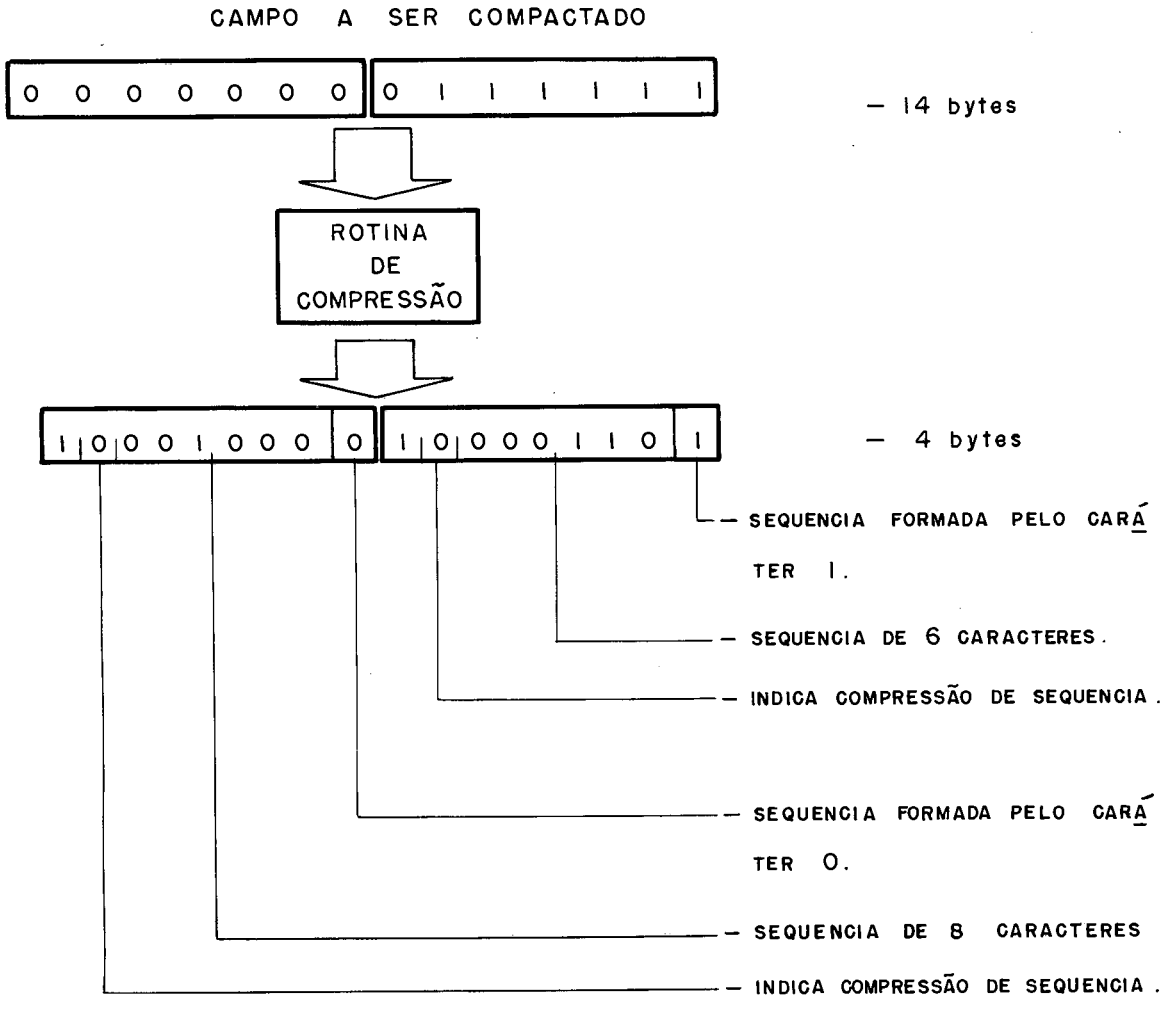
c) Terceiro esquema:

Compressão de uma sequência de caracteres idênticos pela sua substituição por 2 símbolos.

Alternativa 1



Alternativa 2



COMPRESSÃO ABSOLUTA DOS CAMPOS CONCATENADOS — 10 bytes.

II-12

VANTAGENS

a) Simples na sua conceituação e implantação.

b) Dependendo das características dos dados, altas compressões podem ser obtidas.

c) Baixa utilização em tempo de CPU.

DESVANTAGENS

a) As rotinas e/ou programas que acessam diretamente estes arquivos passarão a processar registros de tamanho variável, tornando-se assim um pouco mais complexos.

2.4. COMPRESSÃO DE DADOS CLASSIFICADOS

DESCRIÇÃO

Esta é uma técnica bastante simples e se fundamenta basicamente no fato de que ao se ter um conjunto de itens de dados classificados, haverá uma tendência natural da repetição de caracteres item a item. Aproveitando-se desta característica o método aqui proposto reduz o comprimento destes dados, substituindo no item em questão, os elementos repetidos, por uma indicação de quantos caracteres se repetiram do item anterior.

Esta indicação poderá ser representada por um número ou por um grupo de bits, como os abaixo, indicando o tipo de compressão executada.

- 1000 - item atual idêntico ao item anterior menos o último caráter.
- 1001 - item atual idêntico ao item anterior menos os 2 últimos caracteres.
- 1010 - item atual idêntico ao item anterior menos os 3 últimos caracteres.
- 1011 - item atual idêntico ao item anterior menos os 4 últimos caracteres.

SISTEMÁTICA DE IMPLANTAÇÃO

a) Consideremos a relação de nomes abaixo, classificados em ordem ascendente:

ANTONIO LOPES PINTO
ANTONIO LUCAS PINTO
ANTONIO LÚCIO SANTOS
ANTONIO LUIZ FILHO
ANTONIO LUIZ NETO

Aplicando-se agora o método aqui descrito, substituiremos os caracteres repetidos de cada item em relação ao anterior, por um dígito indicativo do número de elementos eliminados.

NOME ORIGINAL	NOME COMPRESSO	COMPRIMENTO ORIGINAL	COMPRIMENTO COMPRESSO	COMPRESSÃO ABSOLUTA
ANTONIO LOPES PINTO	ANTONIO LOPES PINTO	19	19	0
ANTONIO LUCAS PINTO	(9)UCAS PINTO	19	11	8
ANTONIO LÚCIO SANTOS	(11)IO SANTOS	20	10	10
ANTONIO LUIZ FILHO	(10)IZ FILHO	18	9	9
ANTONIO LUIZ NETO	(13) NETO	17	5	12
	TOTAL EM BYTES	93	54	39

II-14

VANTAGENS

a) Ser um método de conceituação e implantação simples.

b) Baixo consumo em tempo de CPU.

DESVANTAGENS

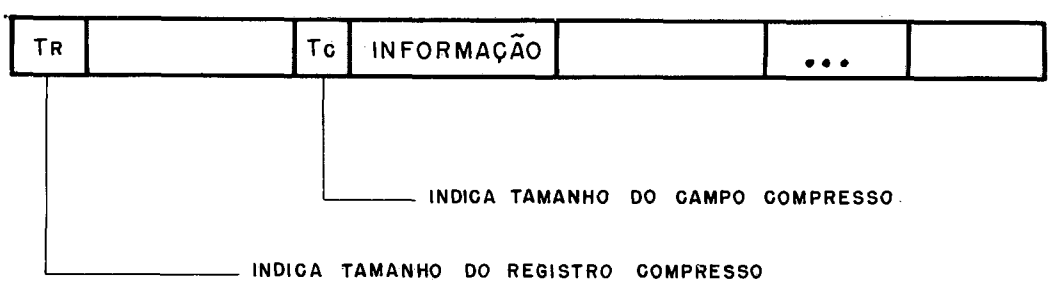
a) Os dados a serem compressos deverão estar classificados.

b) Método de aplicação a campos específicos do registro.

c) Método de resultados imprevisíveis uma vez que depende da frequência com que as sequências de caracteres se repetem item a item.

d) Os programas que acessam estas informações serão mais complexos uma vez que terão que processar campos e consequentemente registros de tamanho variável.

REGISTRO COMPRESSO



II-15

2.5. COMPRESSÃO POR SUBSTITUIÇÃO DE PALAVRAS

DESCRIÇÃO

Através deste processo, procura-se reduzir o volume de informações no arquivo, pela substituição das palavras mais frequentes e que proporcionam maior compressão, por um código de 1 ou mais bytes que identifique perfeitamente cada palavra. Este processo poderá ser aplicado a um ou mais campos do registro.

Quando um item de dados assume um conjunto limitado de valores, o procedimento mais racional é codificar cada valor, armazenando-o em uma tabela (código e valor), permitindo assim, sempre que necessário, a codificação e decodificação da informação desejada.

Este procedimento é bastante difundido, quando se utiliza dados, como:

- nome do material em um código de material
- nome do banco em um código de bancos
- nome de cargos em um código de cargos.

Por outro lado, campos como nome, endereço e outros, de comprimento razoável e que assumem uma quase infinidade de valores, tornam praticamente impossível qualquer tipo de codificação, sendo por esta razão, sempre registrados na sua forma total. Porém se analisarmos os componentes de cada campo separadamente (1º nome, 2º nome, etc.), vamos verificar que cada um deles, toma um conjunto de valores bastante reduzido, o que torna possível o emprego de códigos para representá-los.

De outro modo, caso não seja possível codificar todos os valores assumidos por um dado componente, elegeremos aqueles que proporcionam uma maior compressão ao arquivo.

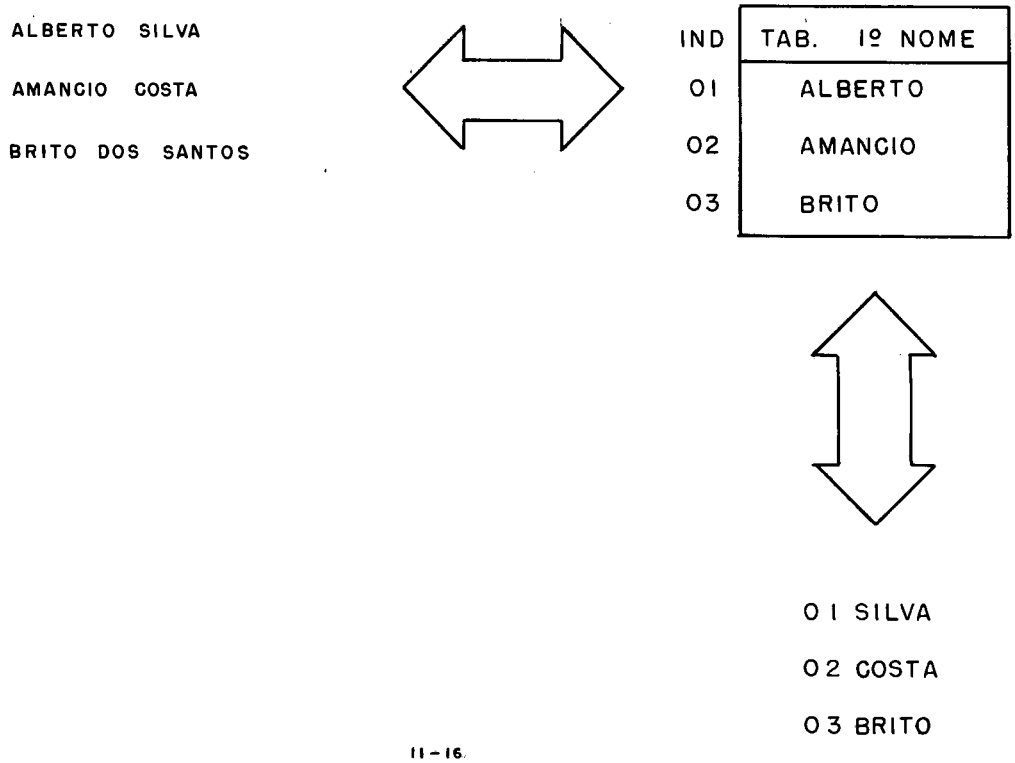
SISTEMÁTICA DE IMPLANTAÇÃO

Uma vez que várias alternativas se apresentam para desenvolvimento do método, escolhemos como sugestão aquelas

mais comumente encontradas.

a) Um processo para codificar um determinado componente de um campo, consiste em se utilizar 1 byte e fazer corresponder cada uma das suas 256 combinações possíveis a cada valor assumido pelo componente em questão, armazenando respectivamente combinação (se necessário, pois em alguns casos a combinação poderá ser o próprio índice da tabela) e valor em uma tabela para posterior decodificação. Nos casos em que a quantidade de valores exceda 256, poderíamos nos utilizar de 2 bytes aumentando o número de combinações possíveis para 65.536 ou de um caráter especial que na decodificação indicaria que o valor a seguir não está codificado.

Suponhamos que a relação de nomes abaixo, constitua o conjunto de valores assumido por um campo e que desejamos codificar o primeiro nome desta relação (componente) com as combinações (Hexadecimal) 01, 02, 03 (utilizadas como índice).



b) Um outro processo muito difundido, consiste em se obter para cada componente do campo ou campos em questão, os respectivos valores com as correspondentes frequências de ocorrência e com estas 2 informações determinar aqueles que permitem uma maior compressão, codificando-os utilizando-se das combinações do código EBCDIC não empregadas na representação de caracteres pertencentes do texto.

Suponhamos que a relação abaixo se constitua no conjunto de valores de um dado campo do registro:

ANTONIO LOPES PINTO

ANTONIO LUCAS PINTO

ANTONIO LÚCIO SANTOS

ANTONIO LUIZ FILHO

ANTONIO LUIZ NETO

II-17

Em nosso próximo passo, obtemos as frequências de ocorrências de cada um dos valores assumidos por cada um dos componentes do nosso campo, representando-os na tabela abaixo.

COMPONENTES	FREQUENCIA	C. ORIGINAL (bytes)	C. COMPRESSO (bytes)	COMP. ABSOL (bytes)
ANTONIO ∅	5	40	5	35
LUIZ ∅	2	10	10	0
PINTO ∅	2	12	2	10
LOPES ∅	1	6	6	0
LUCAS ∅	1	6	6	0
LÚCIO ∅	1	6	6	0
SANTOS ∅	1	7	7	0
FILHO ∅	1	6	6	0
NETO ∅	1	5	5	0
	TOTAL (bytes)	98	53	45

II-18

Uma vez supondo-se que temos disponíveis do código EBCDIC (não utilizados no texto) apenas 2 códigos (≠, $), tomaremos da tabela acima os 2 valores que nos ofereçam a maior redução possível e codificamos.

ANTONIO ↓ - ≠

PINTO ↓ - $

Assim após a compressão a nossa relação de nomes se transformará na relação abaixo, e conseguiremos reduzir 45 bytes do total de 98.

≠ LOPES ↓ $

≠ LUCAS ↓ $

≠ LÚCIO ↓ SANTOS ↓

≠ LUIZ ↓ FILHO ↓

≠ LUIZ ↓ NETO ↓

II-19

VANTAGENS

a) Como o método anterior este também apresenta fácil conceituação, permitindo uma simples implantação.

b) Dependendo do dado consegue-se boas compressões.

DESVANTAGENS

a) Para execução do método necessitamos de tabelas que, dependendo do conjunto de valores escolhido, podem absorver grandes quantidades de memória, comprometendo também o sistema pelo uso exagerado de CPU advindo das pesquisas a estas tabelas.

b) Mais uma vez como no método anterior, comprometeríamos a simplicidade de nossos programas, uma vez que teríamos que trabalhar com campos e registros de tamanho variável.

2.6. COMPRESSÃO POR AGRUPAMENTOS DE CARACTERES

DESCRIÇÃO

Considero este método uma generalização do método anterior - Compressão por Substituição de Palavras onde não se procura as palavras e sim todo e qualquer grupo de caracteres cuja a compressão obtida em função da respectiva probabilidade de ocorrência justifique a sua substituição por um código.

Entretanto, devemos advertir, que, para o sucesso deste método é necessário que as informações sejam estatisticamente estáveis, ou seja, que as suas propriedades estatísticas permaneçam constantes em relação ao tempo, sob pena de termos periodicamente de reavaliar os grupos e recodificalos.

Dependendo do número de grupos eleitos, nosso código poderá ser composto de 1 ou mais bytes. Porém devemos sempre ter em mente que as combinações utilizadas na codificação não devem fazer parte do texto a ser compresso.

Neste ponto, podemos sentir a importância da determinação dos grupos neste processo, e para isto devemos levar em consideração 2 pontos fundamentais:

a) A economia de espaço obtida, quando se substitui o grupo por seu código, fato conhecido como Poder de Compressão (PC).
   PC = FREQUÊNCIA GRUPO * (Comp. Grupo - Comp. Código)

b) O quanto a frequência de um dado grupo de K elementos afetará as frequências de outros grupos de K elementos que possuam elementos comuns ao grupo em análise.

SISTEMÁTICA DE IMPLANTAÇÃO

a) Determinação dos grupos - Como já sabemos, este método se fundamenta na codificação de grupos de caracteres que por sua frequência e comprimento resultam numa maior compressão. A escolha destes grupos é feita através de um processo interativo fundamentado nas propriedades a e b defi

nidas anteriormente,levando em consideração características do arquivo e comprimento dos grupos.

Os procedimentos a seguir, determinam os grupos de 2 e 3 elementos que satisfazem as nossas condições.

a.1. Determinar na amostra o grupo de 3 elementos mais frequente.

a.2. Substituir na amostra o grupo obtido no item a.1,por um caráter indicador de supressão.

a.3. Determinar as frequências dos grupos de 2 e 3 elementos na amostra resultante do item a.2.

a.4. Considerando a propriedade "b", ou seja, que a frequência de um grupo de K elementos influencia a frequência de outros grupos de K elementos, escolher os "N" grupos de 2 elementos mais frequentes, onde "N" representa o número de códigos disponíveis para substituição dos grupos.

a.5. Determinar o Poder de Compressão dos grupos escolhidos no item "a.4".

a.6. Voltar ao item "a.2".

Estes procedimentos serão repetidos até haver uma redução sensível no Poder de Compressão, quando então já teremos determinado os nossos grupos. Assim, para o nome abaixo

FRANCISCO FRANCIRAN

Supondo que os códigos disponíveis sejam em número de 2

\# $

poderíamos ter obtido os seguintes grupos

GRUPO	FREQUENCIA	CÓDIGO
FRA	2	#
NCI	2	$

b) Rotinas de Compressão/Descompressão - Tanto a compressão como a descompressão podem ser desenvolvidas de uma forma bastante simples, uma vez que basicamente o processo compreende:

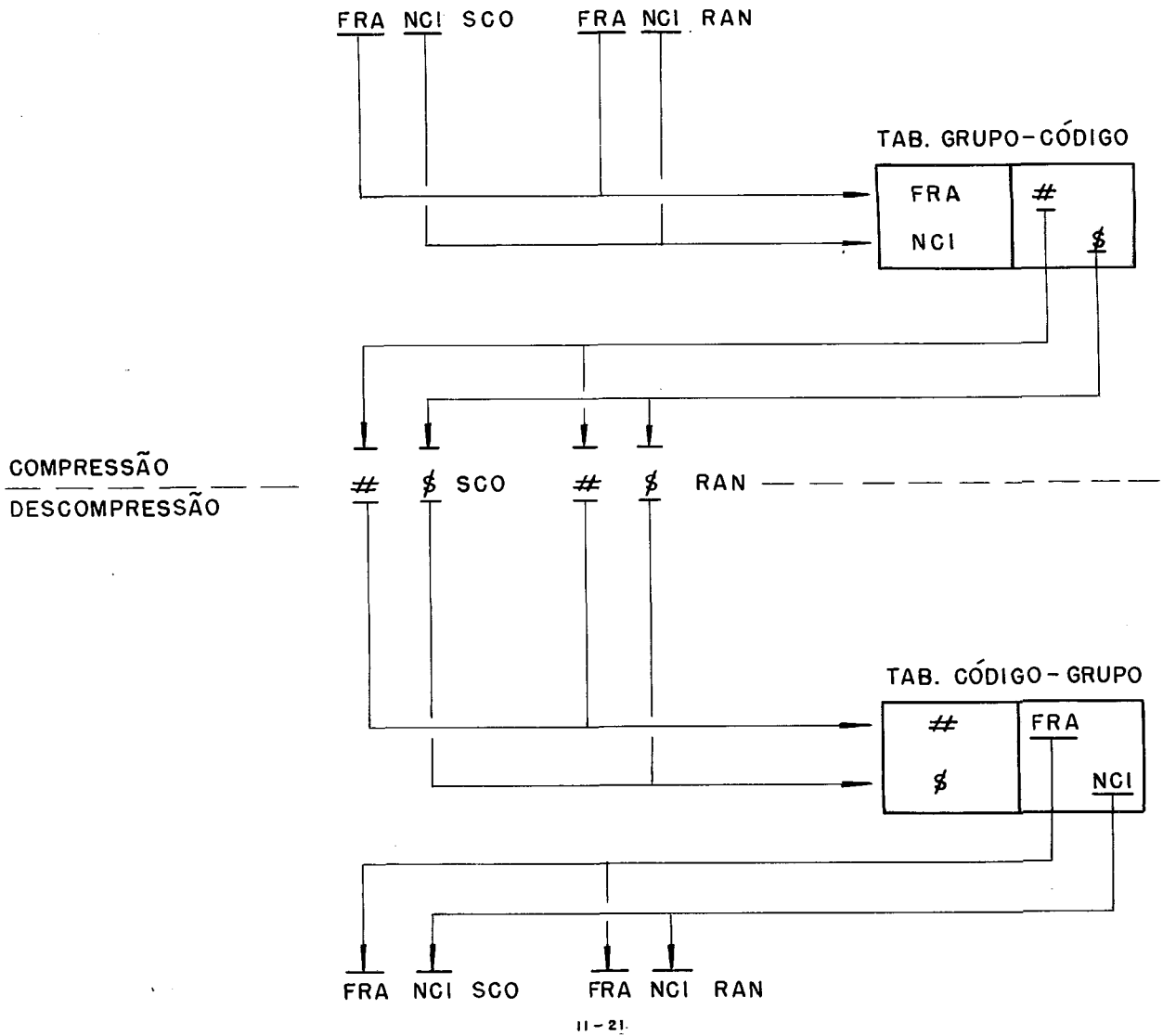
b.1. Obter da amostra o grupo ou código pré-selecionado.

b.2. Com a informação recebida no item anterior pesquisar a tabela desejada, obtendo-se o código ou grupo correspondente.

b.3. Enviar para a saída o dado obtido no item anterior.

Como acabamos de mostrar os procedimentos acima são comuns às rotinas de compressão/descompressão, residindo a única diferença no fato de que durante a compressão obtemos o grupo da amostra e através deste e de uma tabela grupo-código retiramos o código desejado, enquanto na descompressão obtemos o código da amostra e através deste e de outra tabela código - grupo retiramos o grupo desejado. Convém salientar que o método de pesquisa a tabela deve ser cuidadosamente escolhido de modo a dar maior rapidez ao processo.

Seja, por exemplo comprimir e descomprimir o nome abaixo,utilizando os grupos e códigos determinados anteriormente.



VANTAGENS

a) Este método, apresenta bons resultados em arquivos densos de informação, e a redução obtida pode ser estimada por volta de 50% sobre a parte útil dos campos.

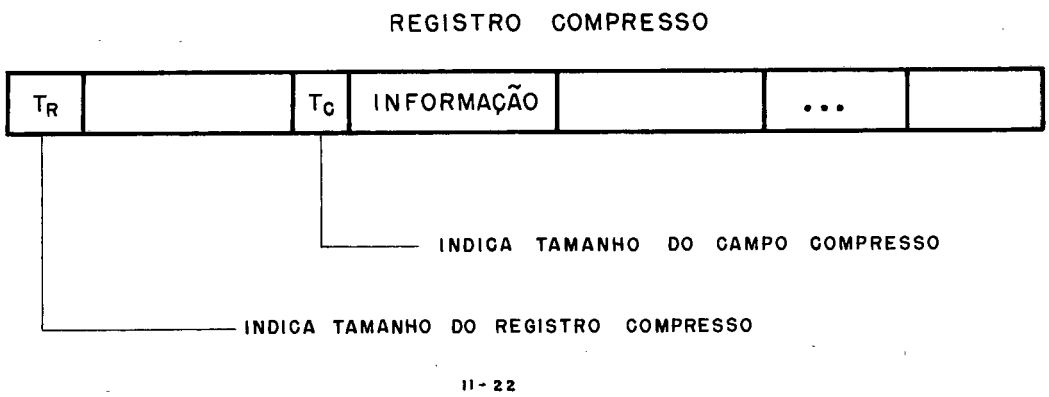
DESVANTAGENS

a) O processamento deste método, apresenta normalmente uma baixa performance, pelas incessantes pesquisas a tabelas, quer na determinação dos grupos quer na execução das rotinas de compressão.

b) A fim de aumentar a eficiência do método os arquivos devem ter alta densidade de informação e estas por sua vez devem ser estatisticamente estáveis.

c) Método, no que se refere a determinação dos grupos, bastante complexo em sua conceituação e implantação.

d) Os programas que acessam estas informações serão mais complexos uma vez que terão que processar campos e consequentemente registros de tamanho variável.



e) Tamanho memória (real) necessário para as tabelas.

3. MÉTODOS DEPENDENTES DA ESTRUTURA DO REGISTRO

3.1. COMPRESSÃO DE DADOS USANDO MAPA DE BITS

DESCRIÇÃO

Na maioria dos centros de processamento de dados, uma prática muito comum é se trabalhar com arquivos cujos registros são de tamanho fixo, uma vez que o processamento de registros com tamanho variável é normalmente mais complexa. Por outro lado, dificilmente encontramos um arquivo onde todos os itens de dados sempre estão presentes, ou seja, existem campos que por certas circunstâncias num dado instante estarão sem conteúdo. Assim, a partir deste fato, um esquema para compressão de dados, pode ser idealizado, tal que itens de dados sem conteúdo sejam eliminados do registro.

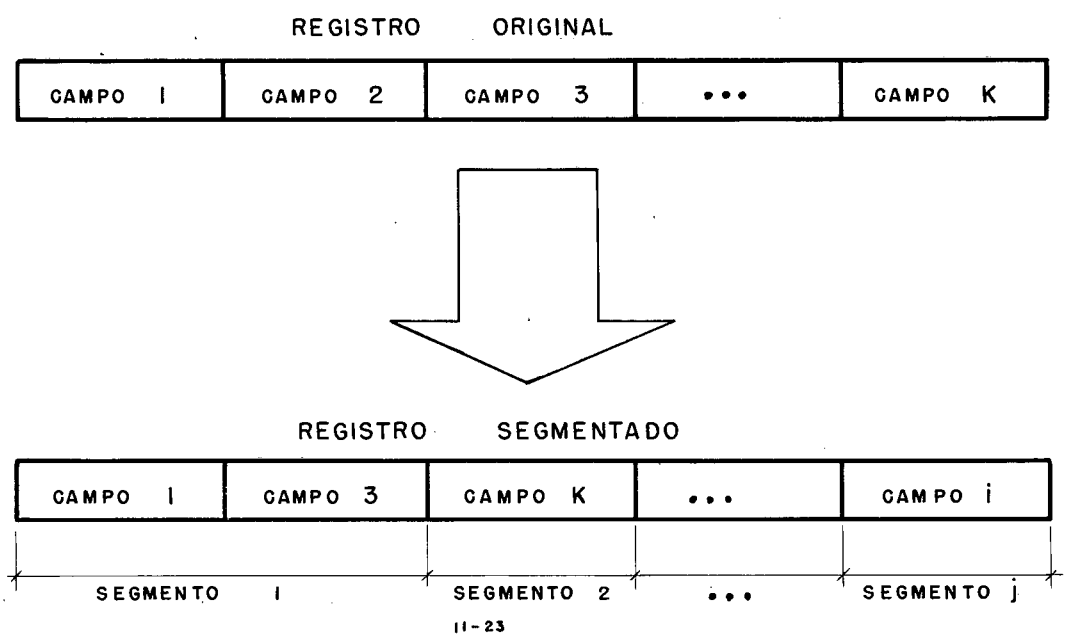
Este esquema, impropriamente denominado compressão de dados, usando mapa de bits, associa um identificador a cada unidade de informação a ser processada, de modo a formar um mapa, onde a presença ou ausência de informação estará registrada através do identificador correspondente.

Esta maneira de registrar a presença ou ausência de uma informação, possibilita a eliminação e posterior recuperação desta mesma informação.

Diversas são as formas de se representar as unidades de informação - bytes, meias palavras, palavras inteiras, campos, segmentos, etc. Porém aconselhamos sejam escolhidas unidades como campos e segmentos uma vez que além de reduzir o custo em tempo de CPU, proporcionam um caráter mais geral as rotinas desenvolvidas.

Por esta razão, deste ponto em diante, trabalharemos com segmentos, pois, como já foi dito, permitem uma maior generalização do método e consequentemente da exposição.

Nós já sabemos, que segmento é um conjunto de campos adjacentes gozando de certas propriedades comuns. Assim uma vez obtida a definição do registro, devemos proceder a definição de sua estrutura (segmentos do registro) e de cada segmento (campos do segmento) resultando daí o que chamamos de registro segmentado.



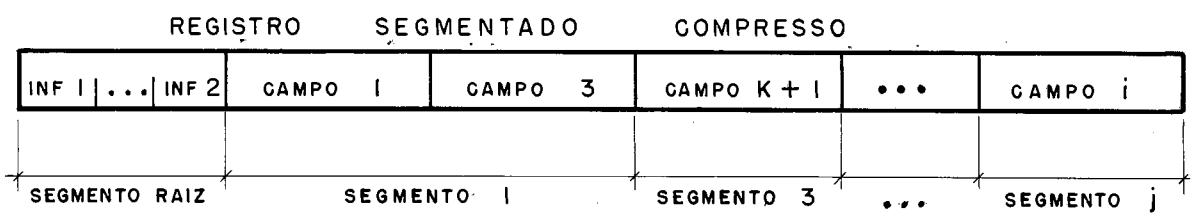
Completando a definição acima, acrescentamos algumas regras básicas, que deverão ser seguidas, de modo a possibilitar a implantação do método.

a) Todo e qualquer segmento possuirá características próprias, independentes dos demais.

b) Uma vez determinado os segmentos e sua sequência no registro lógico, estas permanecerão fixas até uma nova reestruturação.

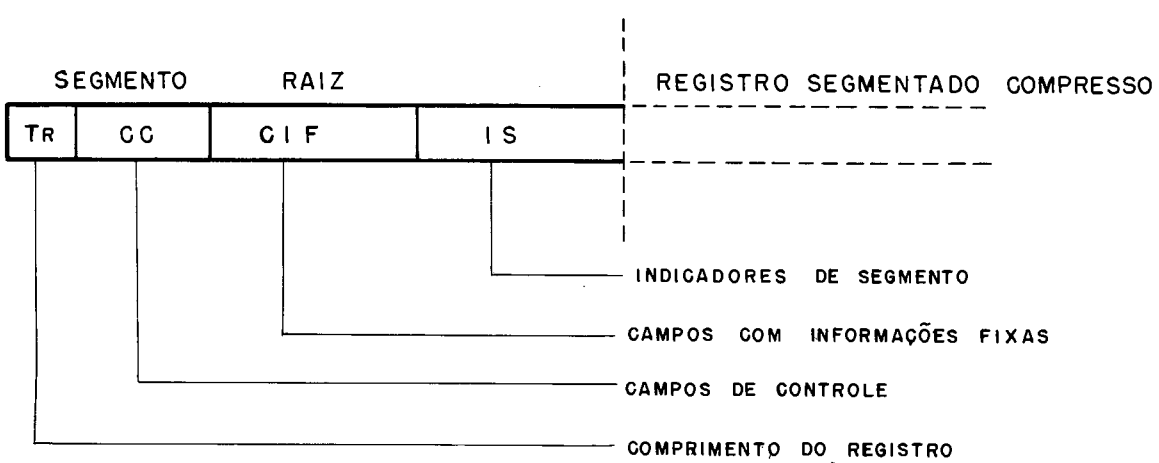
c) Todas as informações concernentes aos segmentos como composição, formato, tamanho, posição relativa, etc., serão registradas em tabelas e armazenadas em bibliotecas do sistema, programas ou no início do arquivo compresso.

Uma vez obtido o registro segmentado, devemos partir para a construção da raiz, ou seja, de um segmento, usualmente localizado no início do registro compresso contendo um conjunto de informações utilizáveis pelas rotinas de compressão e descompressão.



II - 24

Normalmente ao se projetar o segmento raiz, necessitamos dos seguintes campos:



II - 25

a) Tamanho do Registro (T_R) - Uma vez, que os registros,após a eliminaçãodos segmentos sem valor (compressão),serão de tamanhos diferentes, necessário se faz, criar um campo on de se encontre registrado o comprimento real de cada registro compresso. Este campo terá o seu comprimento determinado em função do tamanho do maior registro compresso no arquivo. Por exemplo - 1 byte comporta um registro de no máximo 256 bytes.

b) Campos de Controle (CC) - São campos que dependendo do modo como foram implantadas as rotinas de compressão e descompressão podem conter informações do Sistema Operacional ou das próprias rotinas.

c) Campos de Informações Fixas (CIF) - A critério do usuário, são campos que por sua frequência de utilização(chaves)ou por outras características, justificam sua inclusão na raiz. Observamos que estes campos não estão compactados, podendo se desejado os mesmos fazerem parte de forma compactada do registro compresso.

Este tipo de informação é de grande utilidade, quando além da compressão por segmentação, utiliza-se de outras técnicas para comprimir os segmentos restantes, pois uma vez que o dado desejado se encontra na raiz na sua forma original, não necessitaremos descomprimir um segmento ou mesmo o registro para obter este dado, resultando daí uma redução no tempo de execução dos programas.

d) Indicadores de Segmento (IS) - São identificadores, que associados a cada segmento,indicam a presença ou ausência de conteúdo neste segmento do registro que está sendo processado. Para tornar isto possível, é necessário:

d.1. Que os indicadores se encontrem no segmento raiz.

d.2. Que exista 1 indicador para cada segmento definido no registro.

d.3. Que estes indicadores para maior facilidade de processamento se encontrem na mesma sequência dos segmentos no registro segmentado.

d.4. Que ao usuário seja permitido de alguma forma reestru

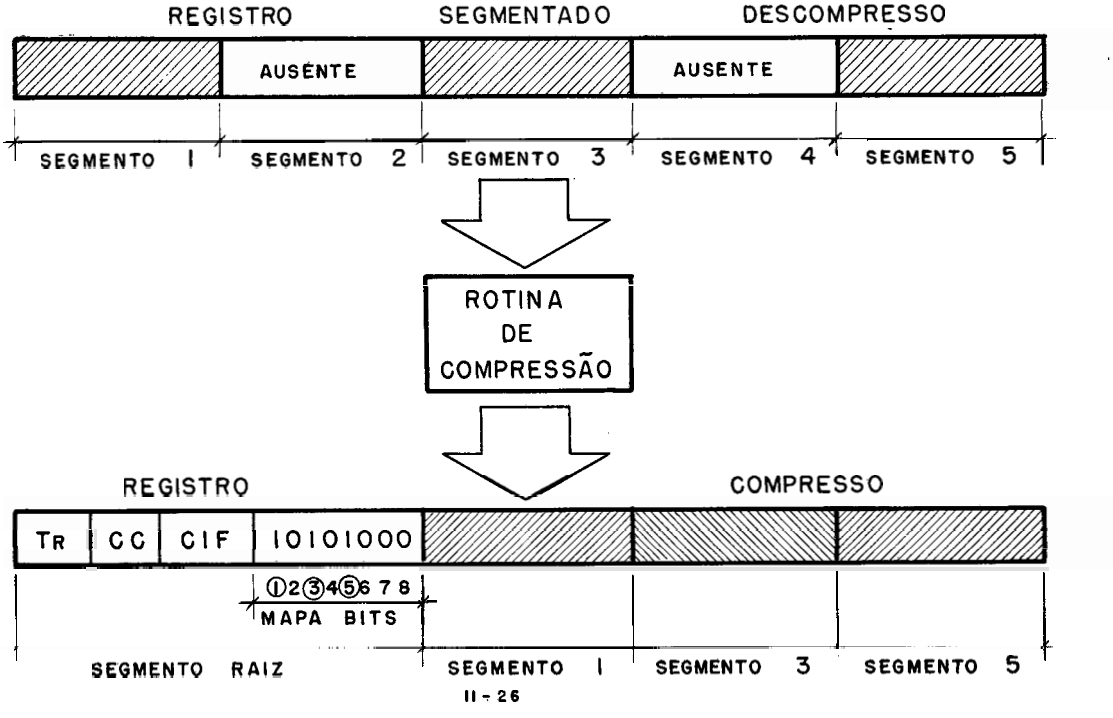
turar os segmentos no registro e consequentemente os indicadores.

A partir da definição apresentada, uma grande variedade de indicadores podem ser idealizados. Porém os indicadores tipo Bit e tipo Deslocamento são os mais frequentemente utilizados.

a) Indicadores tipo Bit (mapa de bits).

Com este tipo de indicadores, cada segmento é representado por 1 bit, onde:

- bit igual a 0 representa segmento sem conteúdo
- bit igual a 1 representa segmento com conteúdo



II - 26

Como podemos verificar no exemplo acima, os segmentos 2 e 4 estão sem conteúdo e no caso os indicadores correspondentes, bits 2 e 4 estão zerados. O fato de no mapa de bits existir os bits 6, 7 e 8 embora não exista os segmentos 6, 7 e 8 é porque a unidade mínima que podemos processar é o byte (8 bits).

b) Indicadores tipo Deslocamento

Para indicadores tipo Deslocamento, cada segmento é representado por 1 ou mais bytes cujo valor represen

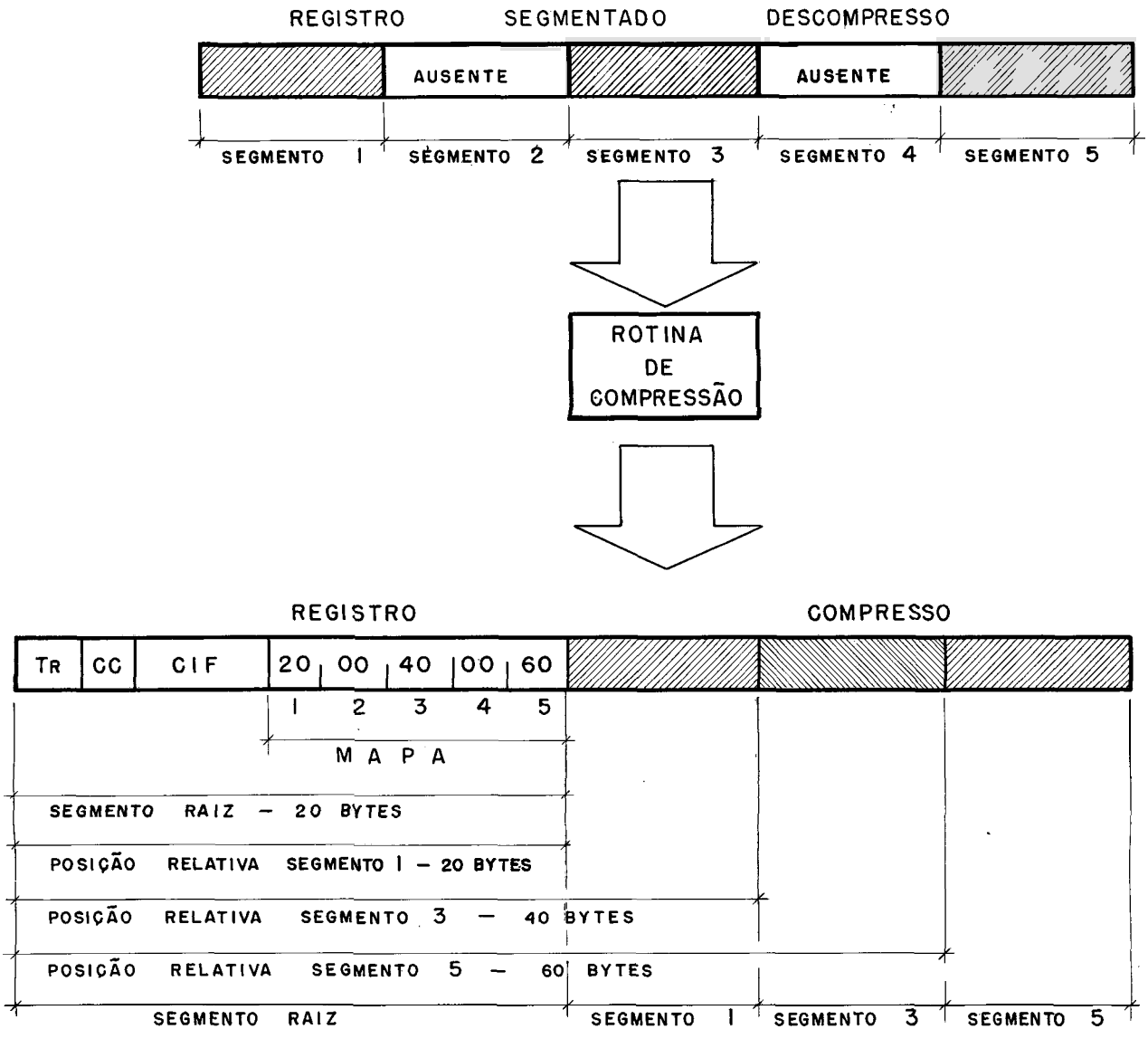
ta o tamanho do segmento correspondente ou sua posição relativa no registro.

Assim,

- indicador igual a 0 representa segmento sem conteúdo
- indicador diferente de 0 representa segmento com conteúdo.

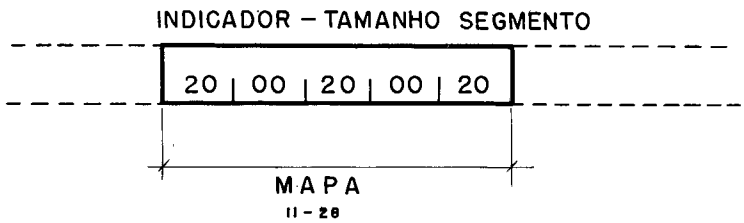
Exemplo:

utilizando indicadores tipo Deslocamento - com indicação no mapa da posição relativa do segmento no registro compresso. Observe-se, que consideràmos todos segmentos com 20 bytes de comprimento.



II-27

Pelo exemplo acima podemos constatar,que os segmentos 2 e 4 estão sem conteúdo e por isto os indicadores correspondentes (bytes 2 e 4) estão zerados. Caso optarmos pela indicação no mapa do tamanho do segmento sempre que este se encontrar presente, a única alteração será no próprio mapa, cujo conteúdo será alterado.

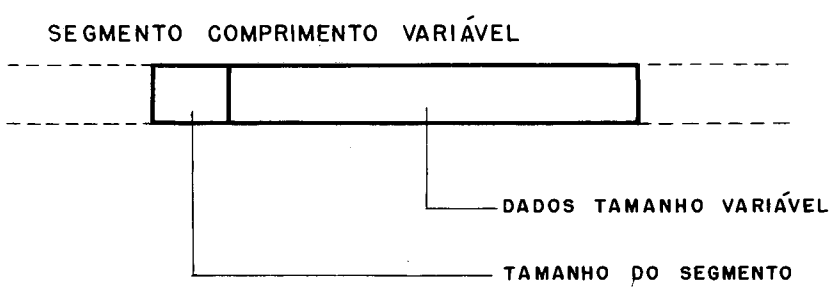


II-28

Finalizando, devemos esclarecer 2 pontos:

- Toda e qualquer informação complementar a respeito dos registros, segmentos e campos, será encontrada em tabelas de descrição (TDR - Tabela de Descrição do Registro e/ou TDS - Tabela de Descrição dos Segmentos), que deverão de alguma forma estar disponíveis para as rotinas de compressão e descompressão.
- Por ser este método um dos que melhor se adapta ao acoplamento com outras técnicas, como supressão de Caracteres Repetidos, Substituição de Palavras, Compressão utilizando Códigos de Tamanho Variável, etc., queremos deixar claro que os segmentos após a compressão, pelas razões que acabamos de expor poderão ser de tamanho variável.

Por este motivo cada segmento variável terá que ser precedido por um campo indicativo do seu tamanho e se necessário, dependendo de como for implantado o método, um código poderá ser acrescentado indicando ser o segmento em questão de tamanho variável.



II-29

Por fim os esclarecimentos aqui prestados em nada alteram os conceitos e exemplos emitidos anteriormente, servindo apenas como complementação destes.

SISTEMÁTICA DE IMPLANTAÇÃO

Aqui, procuraremos mostrar através de um exemplo, todos os procedimentos necessários, para que uma vez determinado o conjunto de informações de um arquivo, possamos definir o registro segmentado e todos os dados imprescindíveis ao bom funcionamento das rotinas de compressão e descompressão.

Suponhamos então, que o conjunto de informações abaixo foi projetado para compor um dado arquivo.

INFORMAÇÃO	DESCRIÇÃO	COMPRIMENTO (bytes)
MATRICULA	IDENTIFICAÇÃO DO CLIENTE	5
NOME	NOME DO CLIENTE	22
ENDEREÇO	ENDEREÇO DO CLIENTE	22
VALOR 1	VALOR 1º DÍVIDA DO CLIENTE	7
VENCIMENTO 1	DATA VENCIMENTO 1º DÍVIDA	6
VALOR 2	VALOR 2º DIVIDA DO CLIENTE	7
VENCIMENTO 2	DATA DE VENCIMENTO 2º DÍVIDA	6
VALOR 3	VALOR 3º DÍVIDA DO CLIENTE	7
VENCIMENTO 3	DATA VENCIMENTO 3º DÍVIDA	6

II-30

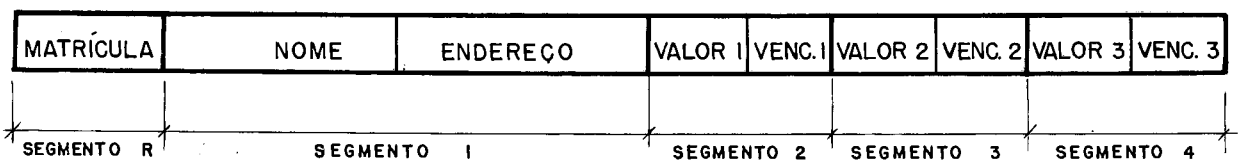
Como primeiro passo para definir o registro segmentado, devemos proceder a uma análise cuidadosa de cada uma das informações, procurando agrupá-las em segmentos. Para este agrupamento de dados, certas informações, como formato, ocorrência, comprimento, tipo informação, etc., devem ser levadas em consideração.

No nosso exemplo o registro segmentado tomou a forma a seguir:

SEGMENTO	CAMPO	POSIÇÃO RELATIVA	COMPRIMENTO SEGMENTO	DESCRIÇÃO
FIXO RAIZ	MATRÍCULA	—	5	NUMERO DO CLIENTE
1	NOME ENDEREÇO	6	44	IDENTIFICAÇÃO DO CLIENTE
2	VALOR 1 VENCIMENTO 1	50	13	DADO RELATIVOS A 1a. DIVIDA DO CLIENTE.
3	VALOR 2 VENCIMENTO 2	63	13	DADOS RELATIVOS A 2a. DIVIDA DO CLIENTE.
4	VALOR 3 VENCIMENTO 3	76	13	DADOS RELATIVOS A 3a. DIVIDA DO CLIENTE.

II-31

LAY-OUT DO REGISTRO SEGMENTADO



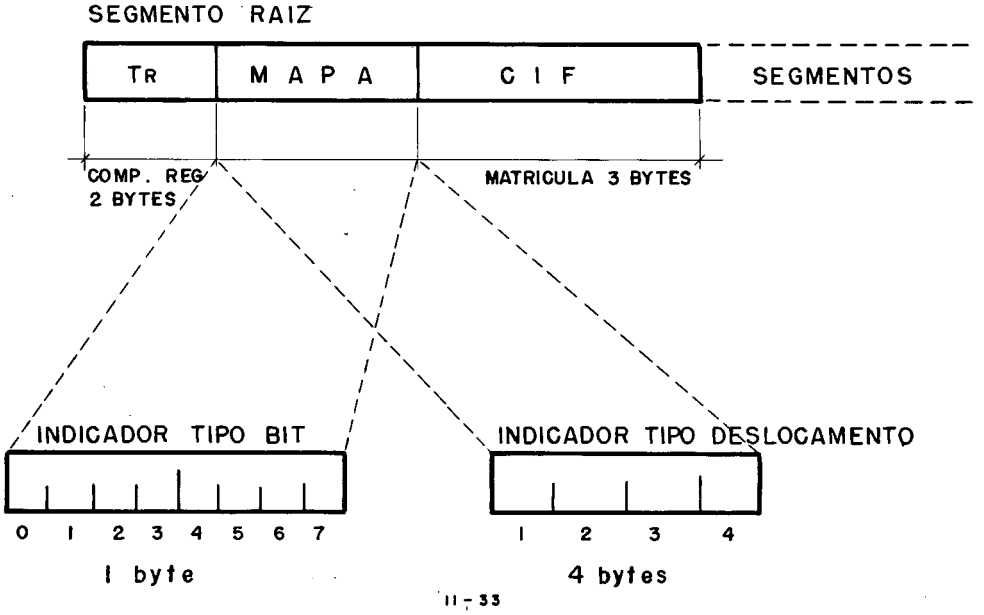
II-32

Uma vez determinado o registro e seus segmentos, passamos ao projeto da raiz que trata da definição e obtenção dos campos de controle, campos de informação fixa, indicadores dos segmentos, etc.

Em nosso exemplo, a composição final ficou estabelecida como se segue:

- T_R - tamanho do registro compresso - 2 bytes.
- Campos de Controle - inexistentes.
- Campos de Informação Fixa - a fim de se evitar compressões e descompressões desnecessárias, decidiu-se colocar a matrícula no segmento raiz, possibilitando desta forma um aumento de eficiência nos programas que processam apenas alguns registros do arquivo.

- Indicadores de Segmento - para uma melhor compreensão resolvemos apresentar o exemplo utilizando os 2 tipos.
  - Mapa de Bits - definido em 1 byte, sendo de 8 o número máximo de segmentos permitidos (8 bits).
  - Deslocamento - definido para controlar 4 segmentos. Considerando que nenhum deslocamento excederá 255 bytes, reservamos 1 byte para cada segmento num Total de 4.



II-33

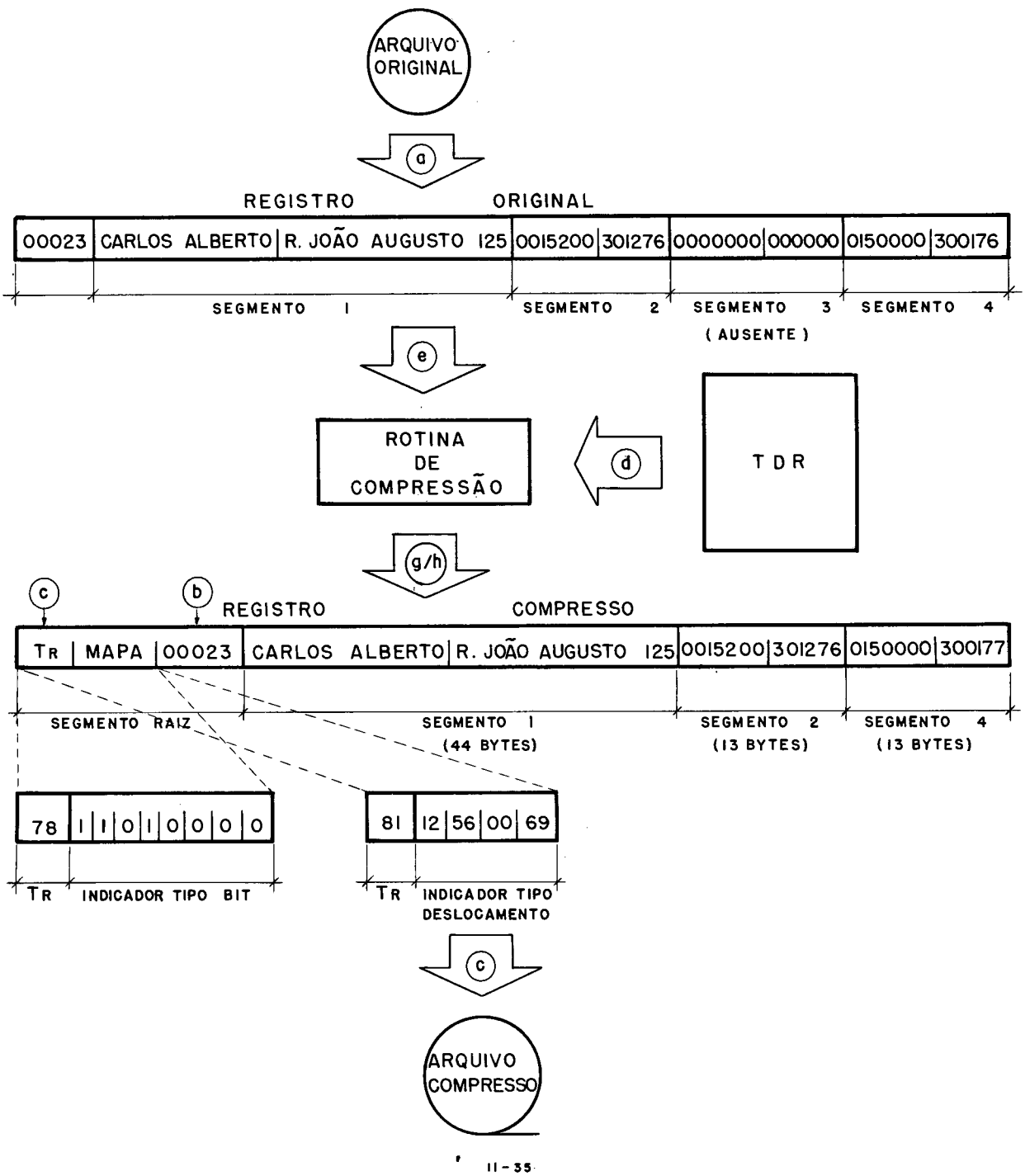
Para o processamento conveniente de cada registro, acrescentam-se as informações já definidas, outras como formato de cada segmento, campos componentes de cada segmento, comprimento dos campos e segmentos e tantas informações quantas sejam necessárias dependendo de como tenham sido projetados os registros e as respectivas rotinas. Por serem estas informações de caráter geral e de utilização por todo programa que processa estes registros, armazenamos estes dados em uma tabela conhecida como Tabela de Descrição do Registro (TDR).

SEGMENTO,	FORMATO	POSIÇÃO RELATIVA	COMPRIMENTO
1	1	6	44
2	2	50	13
3	2	63	13
4	2	76	13

onde: 1 - indica formato alfanumérico
2 - indica formato decimal zonado.

Visto que, o grau de automaticidade das rotinas será tanto maior quanto desejado pelo usuário, procuramos em nosso exemplo apresentar as etapas principais das rotinas de compressão e descompressão, de modo a permitir uma maior automatização no manuseio da raiz e demais segmentos.

Rotina de Compressão:



II-35

Procedimentos de Compressão:

a) Rotina de compressão obtem registro original da entrada (registro descompresso).

b) Armazena Matrícula no segmento Raiz.

c) Se completou o processamento do registro:
   - armazenar comprimento do registro compresso em T_R;
   - armazenar registro compresso na saída; voltar item "a".

d) Da Tabela de Descrição do Registro (TDR), obter dados do próximo segmento:
   - formato
   - posição relativa
   - tamanho

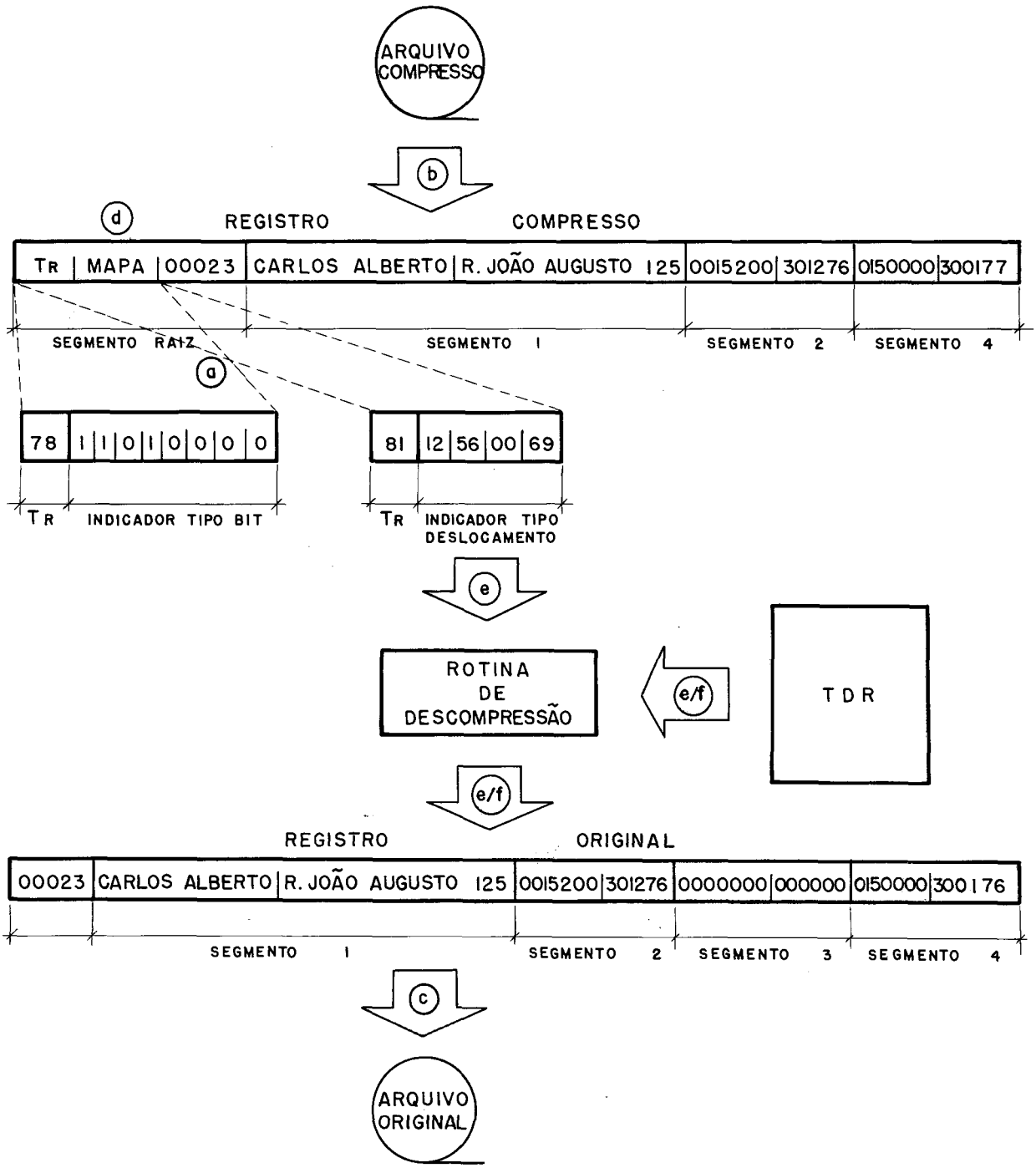
e) Com a posição relativa e tamanho obtidos da Tabela de Descrição do Registro (TDR) retirar do registro descompresso o próximo segmento.

f) De acordo com o formato, verificar se o segmento está com conteúdo:

g) Se o segmento esta com conteúdo:
   - atualizar no mapa a posição correspondente ao segmento, que dependendo do tipo do indicador será ligar um bit ou armazenar a posição relativa do segmento no registro compresso;
   - armazenar segmento no registro compresso;
   - atualizar comprimento do registro compresso;
   - voltar ao item "c".

h) Se o segmento está sem conteúdo:
   - atualizar o mapa zerando o bit ou posição relativa correspondente ao segmento no mapa;
   - voltar ao item "c".

Rotina de Descompressão:



Procedimentos de Descompressão:

a) Rotina de descompressão, obtem da área de entrada a raiz do próximo registro de onde retira o comprimento do registro compresso, mapa e demais informações.

b) Com o comprimento obtem o próximo registro compresso.

c) Se completou o processamento do registro:
   - armazena registro descompresso na área de saída;
   - voltar ao item "a".

d) Pesquisa no mapa de indicadores, o indicador correspondente ao próximo segmento, o qual quando indicador tipo Bit é representado por 1 bit e quando indicar tipo Deslocamento é representado por 1 ou mais bytes.

e) Se indicador presente (bit igual a 1 ou diferente de zero para indicador tipo Deslocamento):
   - obter da Tabela de Descrição do Registro (TDR) o tamanho do segmento;
   - armazenar o segmento no registro descompresso;
   - voltar ao item "a".

f) Se indicador ausente:
   - obter da Tabela de Descrição do Registro (TDR) formato e comprimento do segmento;
   - de conformidade com o formato preencher o referido segmento com zeros ou brancos no registro descompresso;
   - voltar ao item "a".

VANTAGENS

a) Método de fácil conceituação e implantação, sendo ainda bastante flexível, de modo a permitir que as rotinas sejam projetadas e desenvolvidas de uma forma que melhor se adapte às necessidades do usuário.

   Assim as rotinas podem, se desejado, serem mais sofisticadas e possuirem características como transparência ao programador, segmentos variáveis, atualização automática do mapa de indicadores, etc.

b) Aplicável a qualquer tipo de informação desde que seja pos

sível subdividi-la em alguma das unidades de informação anteriormente definidas.

c) Permite uma razoável redução no volume de informações do arquivo, da ordem 20% a 40% dependendo da densidade de informação.

d) Permite compressão sobre compressão, ou seja, após aplicar-se a compressão por segmentação, tomamos os segmentos restantes (com conteúdo) e aplicamos um outro método, conseguindo-se assim um melhor índice de compactação.

DESVANTAGENS

a) Depende da estrutura dos dados, ou seja, os itens de dados no registro devem estar estruturados de modo a permitir uma boa segmentação do registro.

b) A compressão resultante depende da densidade de informação no arquivo, pois quanto mais denso for um arquivo menor será a compactação obtida.

c) Redução no índice de compressão pela inclusão da raiz no registro compactado.

4. MÉTODOS DE APLICAÇÃO GERAL

4.1. COMPRESSÃO ATRAVÉS DE CÓDIGOS DE TAMANHO VARIÁVEL

DESCRIÇÃO

Os esquemas para codificação de caracteres normalmente em uso, empregam um número fixo de bits por caráter. Porém, métodos existem, onde o número de bits por caráter é variável, permitindo assim que os dados sejam armazenados ou transmitidos de uma forma bem mais compacta.

Para tanto o tamanho das configurações de bits(grupo) de um código é inversamente proporcional à probabilidade de ocorrência dos caracteres na amostra analisada, ou seja, quanto mais frequente se apresentar o caráter na amostra menor será o grupo atribuído, até o mínimo de 1 bit.

Temos a acrescentar que, por serem os códigos de tamanho variável estruturados à partir de propriedades estatísticas da informação, poderão perder sua eficiência caso, por algum motivo, haja alteração nestas propriedades.

Para maior clareza, suponhamos uma amostra constituída dos caracteres A, B, C e D (alfabeto), cujo comportamento estatístico apresentamos abaixo:

CARÁTER	PROBABILIDADE DE OCORRENCIA
D	0.60
B	0.25
A	0.10
C	0.05

II-37

Assim a cada 100 caracteres observados nesta amostra é provável se encontrar:

60 CARACTERES D
25 CARACTERES B
10 CARACTERES A
5 CARACTERES C

II-38

Desta forma, caso desejassemos utilizar um código de tamanho fixo, necessitaríamos de 2 bits por caráter e uma maneira de representá-los, seria:

CARÁTER	CODIGO
D	00
B	01
A	10
C	11

II-39

Porém, se agora, tomamos os mesmos caracteres e utilizamos um código de tamanho variável, teremos:

CARÁTER	CÓDIGO	COMPRIMENTO (BITS)	PROB. OCORRENCIA	CUSTO
D	0	1	0.60	0.60
B	10	2	0.25	0.50
A	110	3	0.10	0.30
C	111	3	0.05	0.15

CUSTO MÉDIO - 1.55 BITS/CARÁCTER

II-40

Observando-se a tabela acima, o caráter "D" por ser o mais frequente recebe o menor grupo - "0", seguindo-se os demais grupos, que deverão iniciar com o dígito binário "1", evitando assim, que por ocasião da decodificação, estes se confundam com o grupo "0" atribuído ao caráter "D", desta forma o carater "B" será codificado pelo par de bits "10", o "A" por "110" e o "C" por "111", sempre atento para que nenhum grupo coincida com o início de outro.

Da comparação dos 2 métodos de codificação, acusamos uma redução média de 0,45 bits/caráter, visto que o primeiro processo para codificar os 4 caracteres necessitou de 2 bits/caráter enquanto o segundo utilizando-se de códigos de tamanho variável, requereu em média apenas 1,55 bits/caráter.

No entanto convem-nos salientar que, se a probabilidade de ocorrência dos caracteres fosse a mesma, o custo médio do código de tamanho variável passaria a 2,25 bits/caráter (1X0,25 + 2X0,25 + 3X0,25 + 3X0,25), ou seja, em média haveria um aumento de 0,25 bits/caráter pela utilização do código de tamanho variável em relação ao código de tamanho fixo.

Diante do que foi exposto, podemos concluir que quanto mais diversificada for a probabilidade de ocorrência dos caracteres na amostra, menor será o custo médio apresentado pelo código.

Pelos fatos até aqui apresentados, podemos agora enunciar 2 importantes propriedades relacionadas com os códigos de tamanho variável.

a) A probabilidade de ocorrência (Pi) de um dado caráter esta intrinsecamente relacionada com o comprimento (Ci = número de bits) do grupo a ele atribuído.

Assim

$$P_1 \geq P_2 \geq P_3 \ldots \geq P_n \quad \text{então}$$

$$C_1 \leq C_2 \leq C_3 \ldots \leq C_n$$

onde "n" representa o número de elementos do alfabeto.

Em nosso exemplo:

$(P_1\ 0{,}60) > (P_2 = 0{,}25) > (P_3 = 0{,}10) > (P_4 = 0{,}50)$ então

$(C_1 = 1) < (C_2 = 2) < (C_3 = 3) = (C_4 = 3)$

b) Como vimos anteriormente em nosso exemplo, nos códigos de tamanho variável qualquer configuração de bits atribuída a um caráter (grupo) não pode ser início de outro grupo do mesmo código.

Conhecida como a propriedade do Prefixo, somente ela torna possível a decodificação de uma mensagem.

Por fim, temos a acrescentar que existem 2 métodos que se utilizam dos códigos de tamanho variável, cuja diferença fundamental se constitui na forma de determinação das probabilidade de ocorrência dos diversos caracteres na amostra.

Assim:

a) O método denominado Compressão por codificação Condicional, determina a probabilidade de ocorrência dos caracteres na amostra condiconando-os ao caráter precedente.

b) Nos demais métodos a probabilidade de ocorrência é determinada em função da frequência absoluta dos caracteres na

amostra, ou seja, em função da ocorrência total de cada caráter.

Como sabemos os processos aqui tratados se utilizam dos códigos de tamanho variável e por esta razão apresentamos a seguir 3 códigos que julgamos, sejam os mais importantes.

SISTEMÁTICA DE IMPLANTAÇÃO

. OBTENÇÃO DO CÓDIGO DE SHANNON-FANO

Este é um código de tamanho variável, que além de satisfazer a propriedade do prefixo, possui o atributo de sequência numérica (os grupos do código ocorrem numa sequência ascendente de números inteiros), que permite reduzir o tamanho das tabelas de tradução como também o tempo necessário a codificação e decodificação das informações.

Para a sua determinação necessitamos apenas executar os procedimentos abaixo:

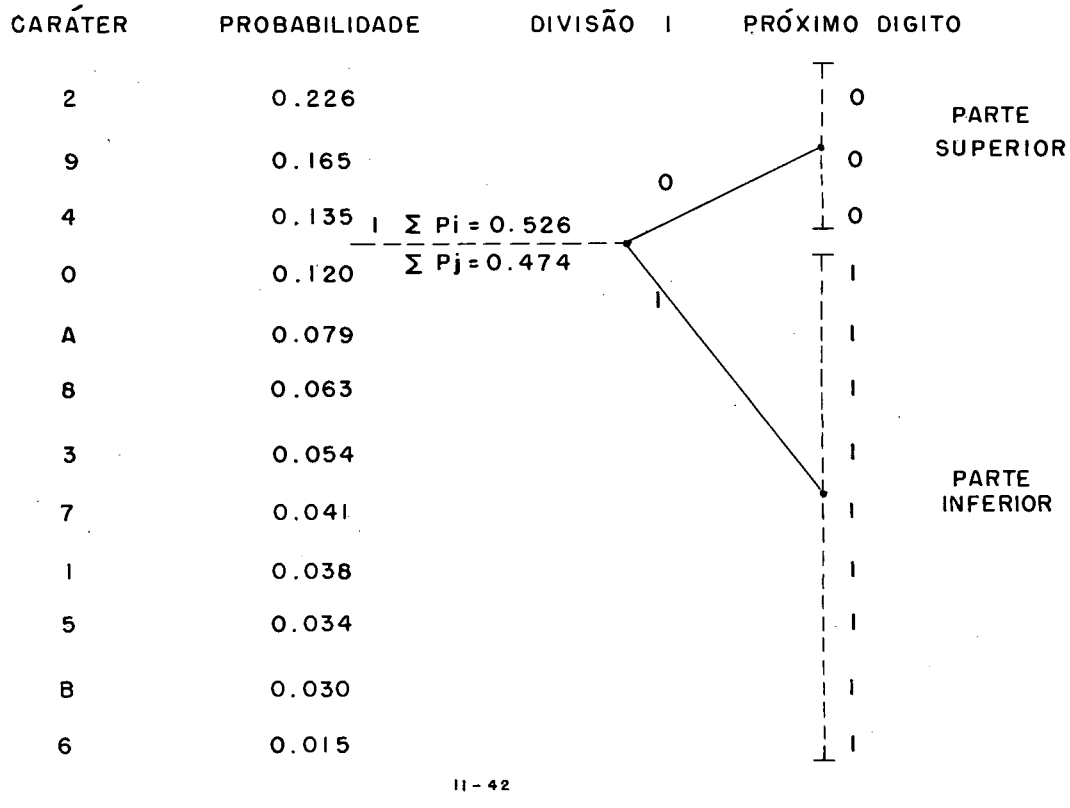
a) Definir um alfabeto de acordo com as características do dado a ser compresso. Assim, em nosso exemplo o alfabeto se constituirá dos elementos 2, 9, 4, 0, A, 8, 3, 7, 1, 5, B, 6, onde qualquer caráter na amostra não previsto no alfabeto será considerado como sendo o elemento "0".

b) Para cada elemento do alfabeto escolhido, determinar na amostra a frequência e a correspondente probabilidade de ocorrência, apresentando o resultado sob forma de tabela, organizada em ordem decrescente de probabilidade.

CARÁTER	PROBABILIDADE
2	0.226
9	0.165
4	0.135
0	0.120
A	0.079
8	0.063
3	0.054
7	0.041
1	0.038
5	0.034
B	0.030
6	0.015
	1,000

c) Dividir a tabela ou Sub-tabela em 2 partes, de tal forma que a soma das probabilidades da parte superior seja a mais próxima possível da soma das probabilidades da parte inferior.

Atribuir "0" e "1" como o próximo dígito binário dos grupos pertencentes respectivamente as partes superior e inferior da tabela.



II-42

d) Tomando-se agora as partes superior e inferior da tabela e considerando cada sub-tabela como uma tabela independente, voltamos ao item "c".

Este processo se repetirá até que a tabela original fique sub-dividida em tantas sub-tabelas quantos forem os elementos do alfabeto, ou seja, quando cada sub-tabela contiver 1 só elemento.

CARÁTER	PROBABILIDADE	CÓDIGO	CÓDIGO	CÓDIGO	CÓDIGO	CÓDIGO	VALOR	COMP	CUSTO
2	0.226	0	00				0	2	0.452
9	0.165	0	01	010			2	3	0.495
4	0.135	0	01	011			3	3	0.405
0	0.120	1	10	100			4	3	0.360
A	0.079	1	10	101	1010		10	4	0.316
8	0.063	1	10	101	1011		11	4	0.252
3	0.054	1	11	110	1100		12	4	0.216
7	0.041	1	11	110	1101		13	4	0.164
1	0.038	1	11	111	1110	11100	28	5	0.190
5	0.034	1	11	111	1110	11101	29	5	0.170
B	0.030	1	11	111	1111	11110	30	5	0.150
6	0.015	1	11	111	1111	11111	31	5	0.075

CUSTO MEDIO = 3,245 bits / car

Observando-se a construção do código demonstrado na tabela acima, vamos verificar a formação de uma árvore binária, onde a raiz é a tabela de probabilidades original, os nós as sub-tabelas e as folhas (nós terminais) os grupos. Se percorrermos esta árvore em todos os seus caminhos iremos obter todos os grupos do código.

. OBTENÇÃO DO CÓDIGO DE HUFFMAN

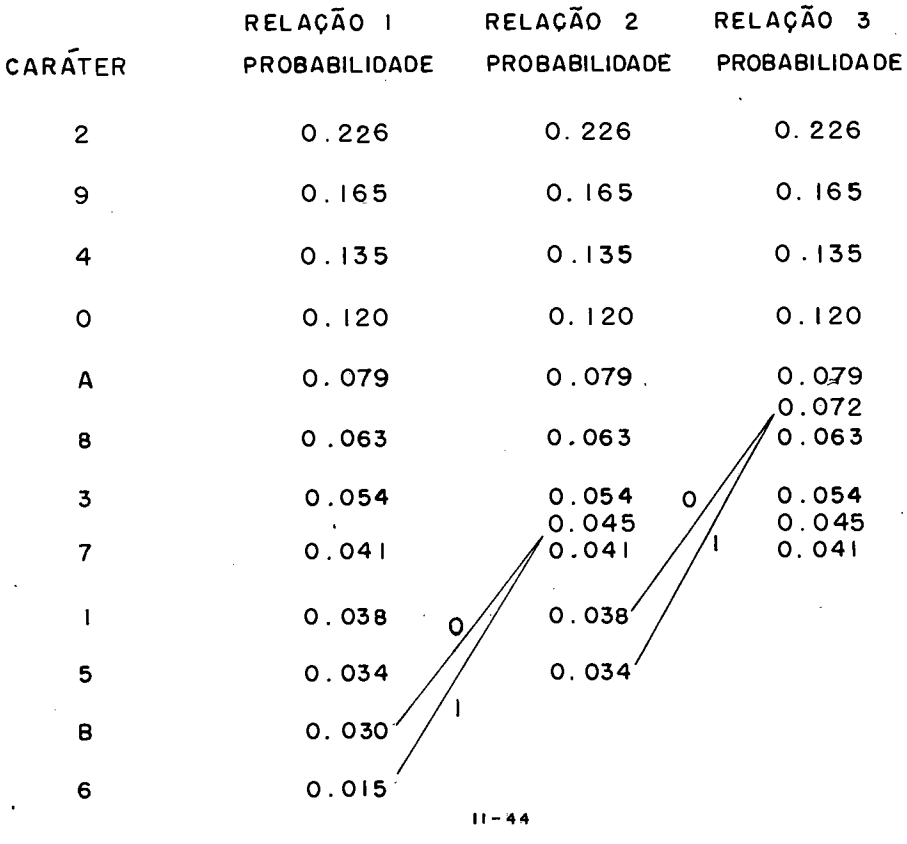
Sendo mais um código de tamanho variável, o código de Huffman tem a vantagem de possuir o menor custo, satisfazendo a propriedade do prefixo, além de nos oferecer um melhor rendimento na transmissão de dados e necessidades de memória.

Da mesma forma que no código de Shannon-Fano, necessitaremos determinar os grupos do código, o que será feito pelos procedimentos a seguir:

a) Definir o alfabeto de acordo com as características do dado a ser compresso. Uma vez que vamos utilizar o mesmo exemplo, o alfabeto será composto dos elementos 2, 9, 4, 0, A, 8, 3, 7, 1, 5, B, 6, onde qualquer caráter na amostra não previsto no alfabeto será considerado como sendo o elemento "0".

b) Para cada elemento do alfabeto escolhido, determinar na amostra a frequência e a correspondente probabilidade de ocorrência, apresentando o resultado sob forma de tabela, organizada em ordem decrescente de probabilidade, como mostrado no procedimento "b" para determinação do código de Shannon-Fano.

c) Tomando-se agora as 2 menores probabilidades de ocorrência da relação de probabilidades apresentada e atribuindo se "0" e "1" como próximo dígito binário, vamos compondo os grupos dos caracteres correspondentes às probabilidades escolhidas. Em seguida construimos uma nova relação de probabilidades pela eliminação das 2 menores probabilidades e a inclusão de uma nova resultante da soma destas 2.



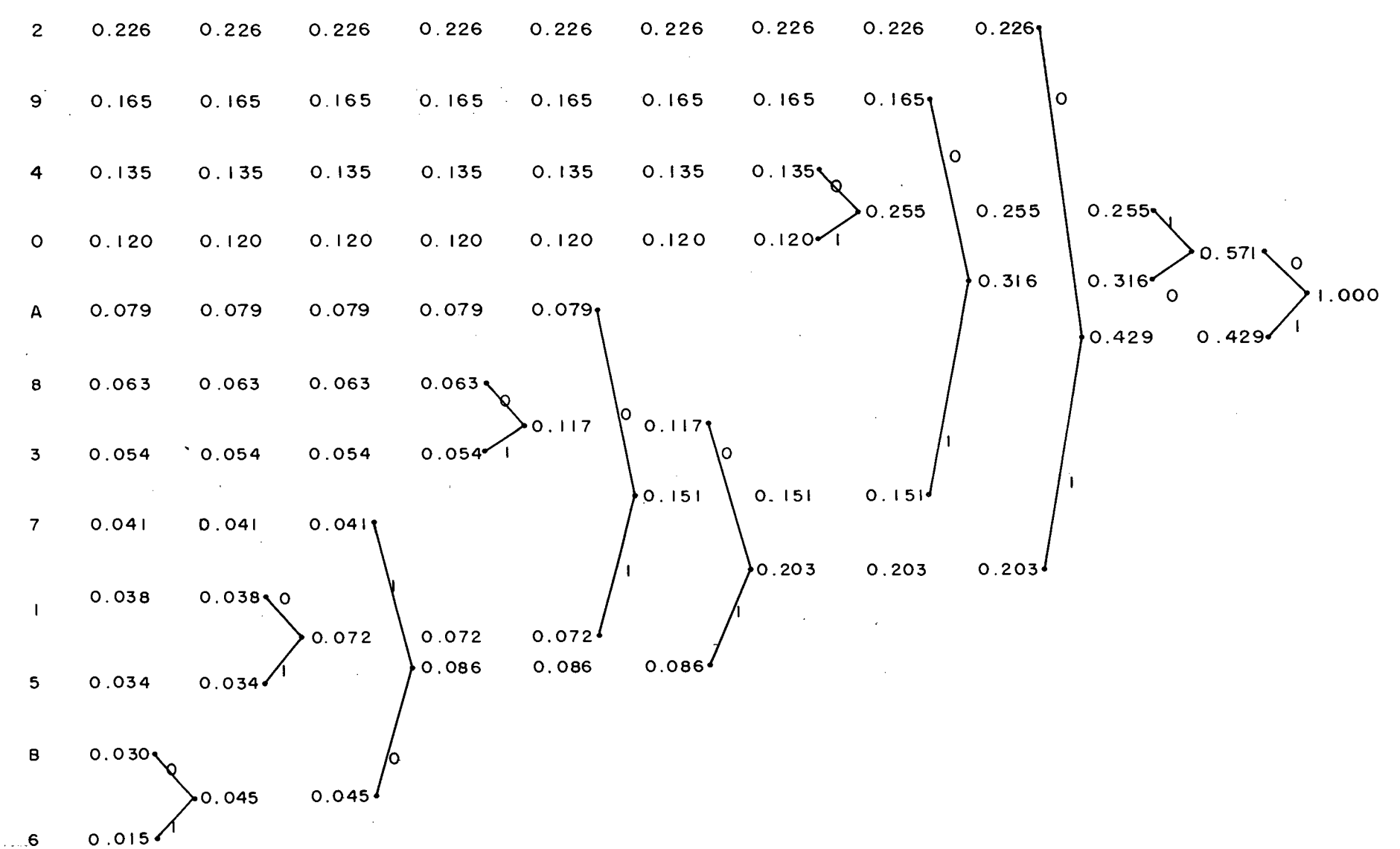
Este procedimento corresponde a uma unificação dos caracteres em questão, de modo a se obter um novo alfabeto e consequentemente uma nova relação de probabilidades. No exemplo acima, é como se os caracteres "B" e "6" fossem considerados um só na amostra e desta forma ao se calcular a sua probabilidade obteríamos 0,045.

d) Com a nova relação de probabilidades, voltamos ao procedimento "c".

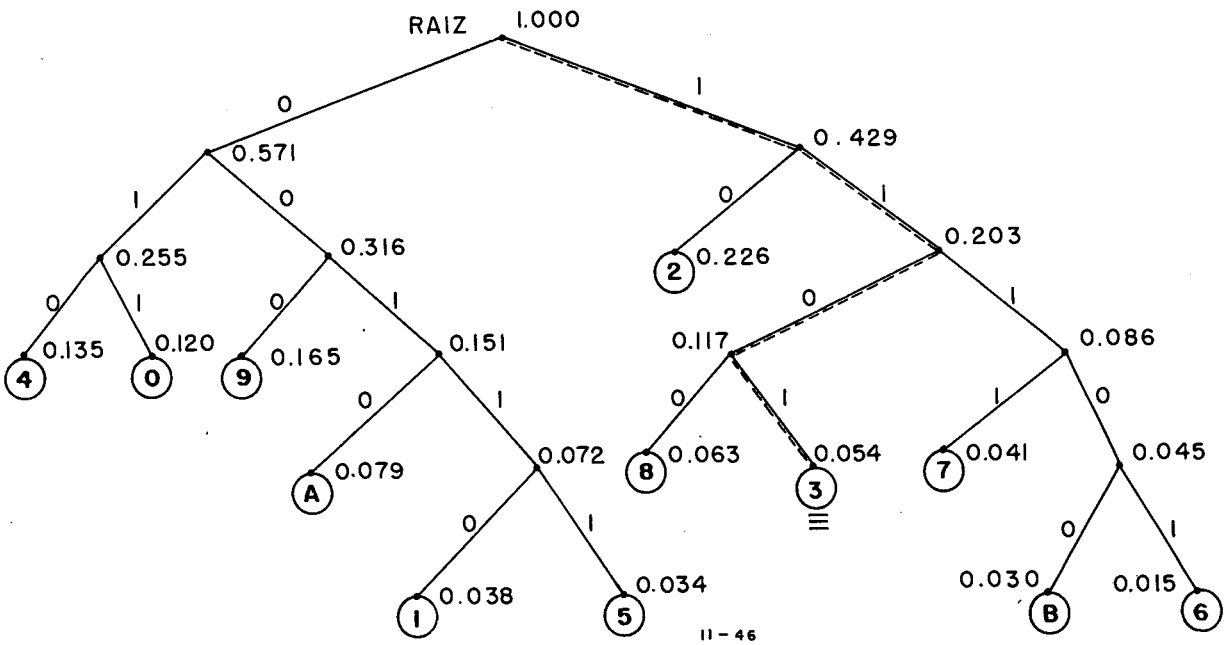
Estes procedimentos serão repetidos, até que se obtenha uma relação de probabilidades com 1 único elemento de valor igual a 1,000.

Mais uma vez se analisarmos o desenvolvimento do processo, vamos sentir a formação de uma árvore binária,onde as folhas (nós terminais) correspondem as probabilidades de ocorrência de cada elemento do alfabeto e logo ao próprio elemento, os nós às probabilidades unificadas, a raiz à probabilidade final de valor 1,000 como se toda amostra fosse composta de 1 único caráter e

RELAÇÕES DE PROBABILIDADES DE OCORRENCIA



os ramos a dígitos binários "0" e "1" que comporão os grupos do código quando caminharmos pela árvore. Observamos entretanto que a atribuição destes dígitos deve ser ser padronizada por todo o processo. No nosso exemplo atribuímos "0" ao ramo de maior probabilidade e "1" ao de menor.



Se agora percorremos a árvore binária acima, da raiz para as folhas em todos os seus caminhos, considerando os valores de cada ramo da árvore por onde passamos ("0" e "1"), obteremos todos os grupos do código.

CARÁTER	PROBABILIDADE	CÓDIGO	COMPRIMENTO	CUSTO
2	0.226	10	2	0.452
9	0.165	000	3	0.495
4	0.135	010	3	0.405
0	0.120	011	3	0.360
A	0.079	0000	4	0.316
8	0.063	1100	4	0.252
3	0.054	1101	4	0.216
7	0.041	1111	4	0.164
1	0.038	00110	5	0.190
5	0.034	00111	5	0.170
B	0.030	11100	5	0.150
6	0.015	11101	5	0.075

II-47

Custo médio - 3,245 bits/caráter.

Embora neste exemplo o custo seja o mesmo que no caso Shannon-Fano, este método da resultados demonstrados como iguais ou ligeiramente superiores, dependendo dos dados.

. OBTENÇÃO DO CÓDIGO DE
HUFFMAN - SHANNON - FANO (HSF)

É um código de tamanho variável, que apresenta um mínimo de redundância, satisfazendo a propriedade do prefixo.

Através da utilização deste código, obtemos:

- Mensagens de tamanho médio mínimo.
- Tabelas de tradução mais reduzidas.
- Maior eficiência nos processos de codificação/decodificação.

O código HSF foi idealizado de forma a reunir as vantagens do código de Huffman, que implica numa otimização do tempo de transmissão, redução das necessidades de memória e do código de Shannon-Fano pela redução nas tabelas de tradução e tempo de codificação/decodificação, pois, também o código HSF possui o atributo de sequência numérica.

Para a construção deste código, é necessário introduzir um novo conceito conhecido como Code Index (I), definido como uma concatenação de números decimais

T_1, T_2 ... T_M, assim:

$I = T_1\ T_2\ T_3 \ldots T_M$

Onde T_i - representa o número de grupos em um código de tamanho variável que possuem comprimento "i", sendo "M" o comprimento do maior grupo.

A utilização do Code Index na construção do código HSF, nos permite obter em uma amostra, com o mesmo alfabeto, o mesmo custo médio (número médio de bits/caráter) de qualquer outro código de tamanho variável. Porém como desejamos

que o código HSF nos ofereça mensagens de tamanho médio mínimo, devemos então, uma vez fixado o alfabeto, forçar que o Code Index referente ao futuro código HSF seja igual ao Code Index, resultante da aplicação do código de Huffman a mesma amostra, pois, como sabemos, este é o que nos oferece o menor custo médio.

CARÁTER	PROBABILIDADE	CÓDIGO HUFFMAN	COMPRIMENTO	CODE INDEX
-	-	-	1	$T_1 = 0$
2	0.226	10	2	$T_2 = 1$
9	0.165	000	3	
4	0.135	010	3	$T_3 = 3$
0	0.120	011	3	
A	0.079	0010	4	
8	0.063	1100	4	$T_4 = 4$
3	0.051	1101	4	
7	0.041	1111	4	
1	0.038	00110	5	
5	0.034	00111	5	$T_5 = 4$
B	0.030	11100	5	
6	0.015	11101	5	

II-48

Da tabela acima, utilizando-se do nosso exemplo, obtivemos como code index:

$$I = 01344$$

que é um valor conseguido por intermédio dos grupos de mesmo comprimento. Desta forma para se calcular o code index desejado deveremos empregar todos os procedimentos necessários a determinação do código, porém obtendo-se como resultado final o comprimento de cada grupo e consequentemente o número de grupos de comprimento 1, 2, 3 ..., M bits.

Uma vez conseguido o Code Index, passamos a construção propriamente do código, que se compõe dos procedimen-

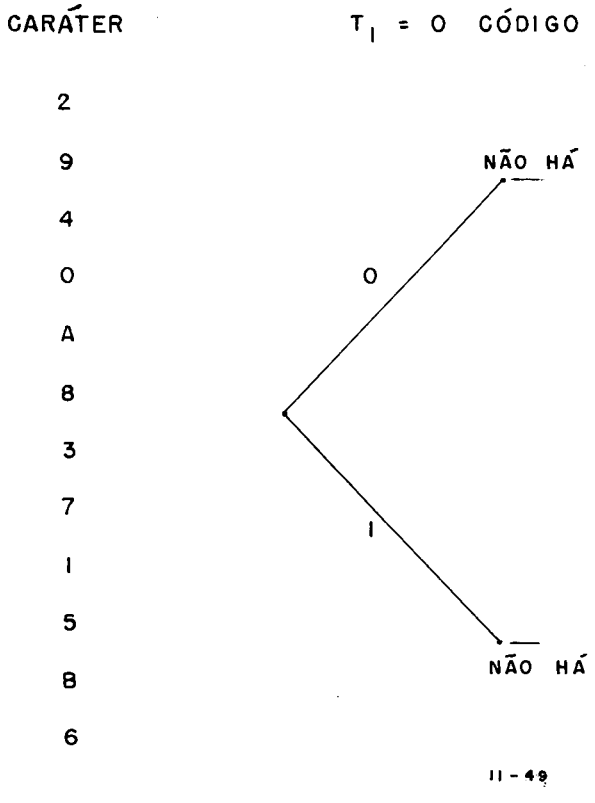
tos a seguir:

a) Retirar do Code Index da esquerda para a direita o valor correspondente ao próximo T_i (este procedimento inicia com T_1).

Se $T_i = 0$, significa que no código de Huffman não existe nenhum grupo de comprimento "i". Assim também em nosso código HSF não deverá existir nenhum grupo com este comprimento. Voltamos ao procedimento "a".

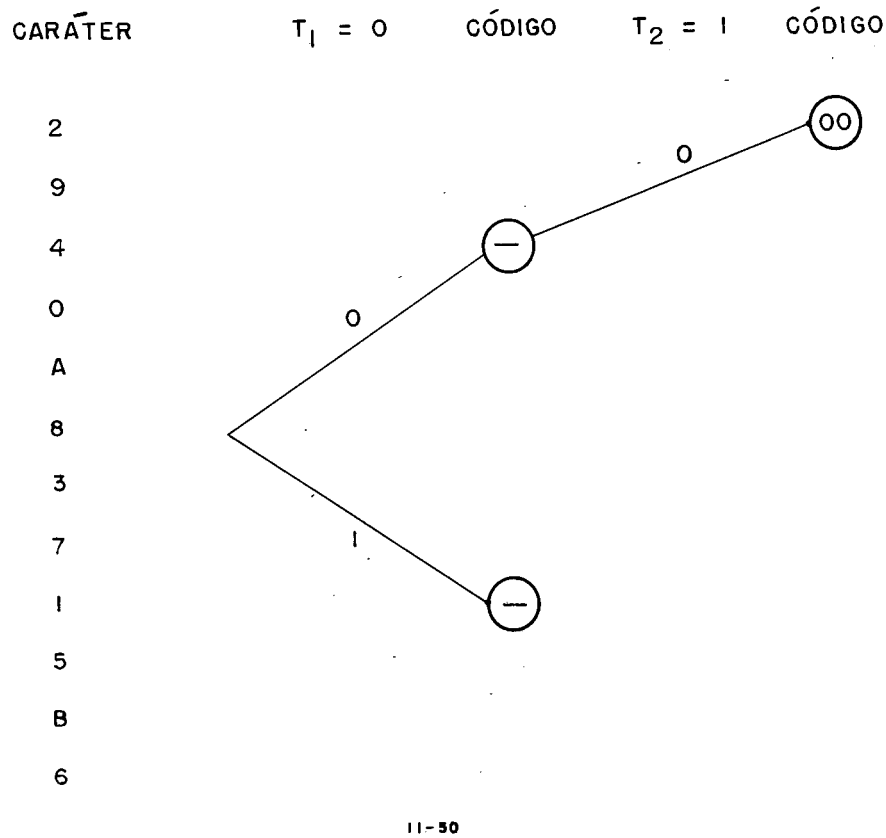
Se $T_i \neq 0$, vamos para o procedimento "b".

	T_1	T_2	T_3	T_4	T_5
I =	0	1	3	4	4



b) Nesta etapa estamos de posse do primeiro $T_i \neq 0$ e ao primeiro grupo deste conjunto de comprimento "i" atribuimos o valor zero, seguindo-se a obtenção dos grupos restantes (T_i - 1) pela simples adição de 1 ao grupo anterior, visto que nosso código HSF possui o atributo de sequência numérica.

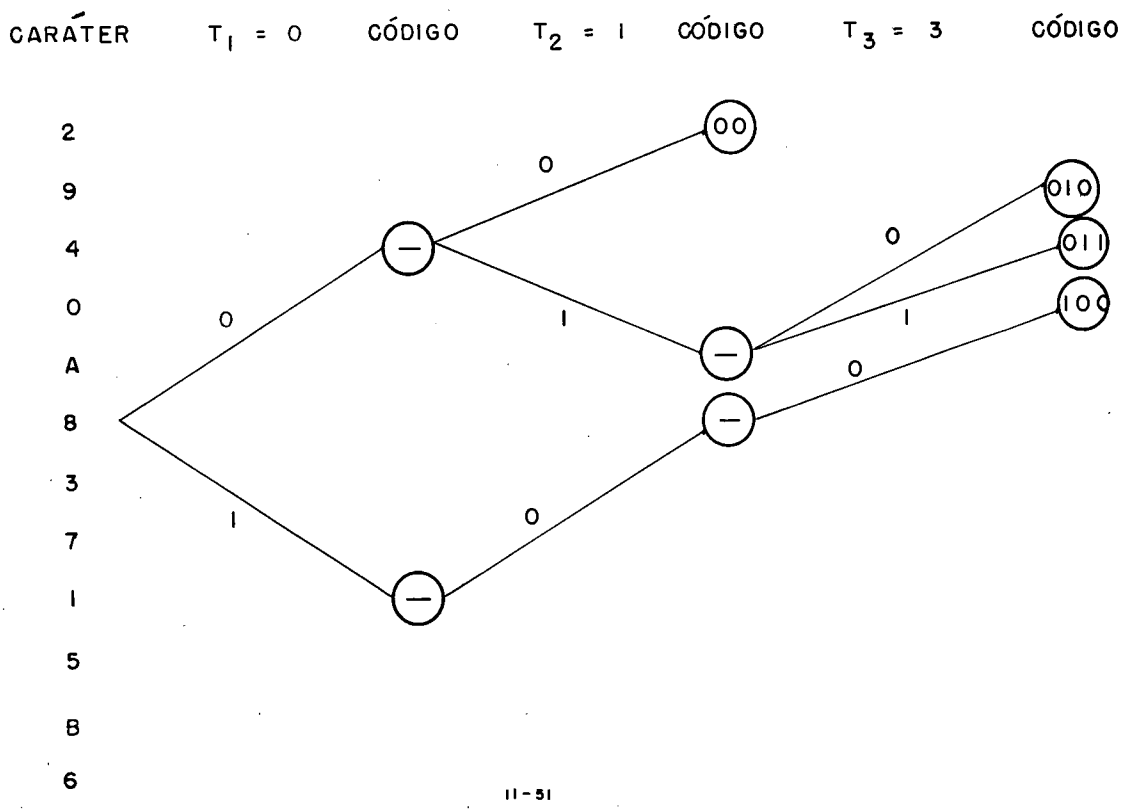
	T_1	T_2	T_3	T_4	T_5
I =	0	1	3	4	4



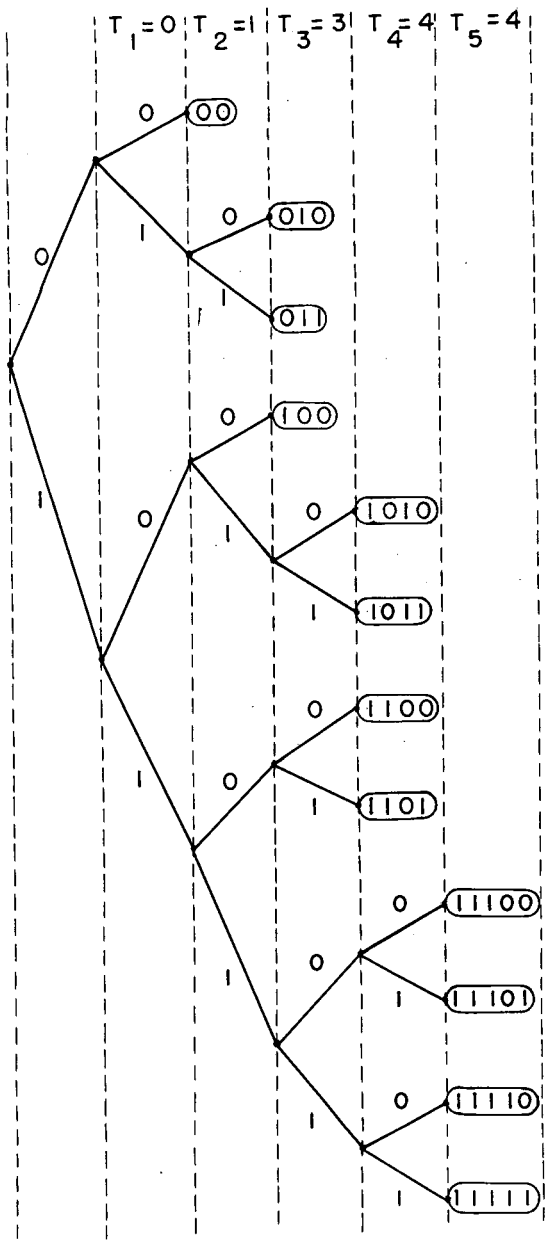
II-50

c) Retirar do Code Index o próximo T_i, se $T_i \neq 0$, gerar todos os grupos (T_i) de comprimento "i", iniciando com o valor do grupo anterior mais 1 preenchidos à direita com tantos zeros quantos sejam necessários para o grupo atingir o comprimento desejado ("i"), seguindo-se a obtenção dos demais grupos pelo incremento de 1 ao grupo anterior. Caso $T_i = 0$, não existe nenhum grupo no código de comprimento ("i") e assim voltamos ao procedimento "c".

	T_1	T_2	T_3	T_4	T_5
I =	0	1	3	4	4



Estes procedimentos se repetirão até termos processado todos os T_i do nosso Code Index, quando então teremos obtido como último grupo, uma configuração totalmente formada de dígitos binários "1" de valor igual a $2^M - 1$, onde "M" é o número de bits do maior grupo no código.



CRC	PROB.	CODIGO HUFFMAN	CÓDIGO HSF	VALOR	CMP.	CUSTO
2	0.226	10	00	0	2	0.452
9	0.165	000	010	2	3	0.495
4	0.135	010	011	3	3	0.405
0	0.120	011	100	4	3	0.360
A	0.079	0010	1010	10	4	0.316
8	0.063	1100	1011	11	4	0.252
3	0.054	1101	1100	12	4	0.216
7	0.041	1111	1101	13	4	0.164
1	0.038	00110	11100	28	5	0.190
5	0.034	00111	11101	29	5	0.170
B	0.030	11100	11110	30	5	0.150
6	0.015	11101	11111	31	5	0.075

CUSTO MÉDIO – 3.245 bits/carater

II – 52

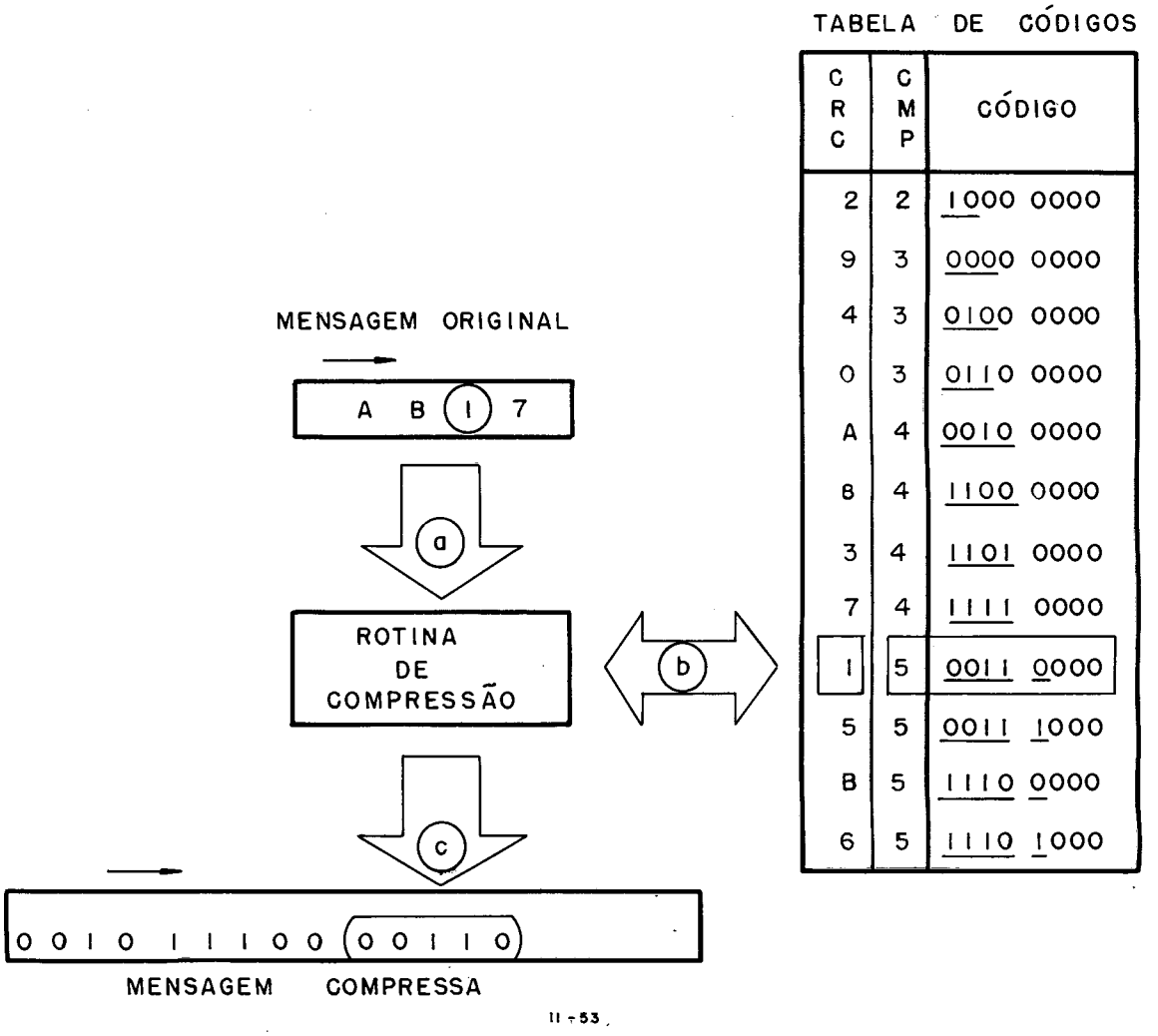
Como nos códigos anteriores, mais uma vez podemos observar a formação de uma árvore binária de onde podemos retirar todos os grupos do código HSF.

Comparando-se, agora o código de Huffman com o HSF, vamos verificar que:

a) Ambos possuem o mesmo custo médio, já que o Code Index é igual.

b) Os grupos no código HSF ocorrem numa sequência ascendente de números inteiros, propriedade do código Shannon-Fano, que muito facilitará as rotinas de compressão/descompressão.

Rotina de Compressão:



Procedimentos de compressão:

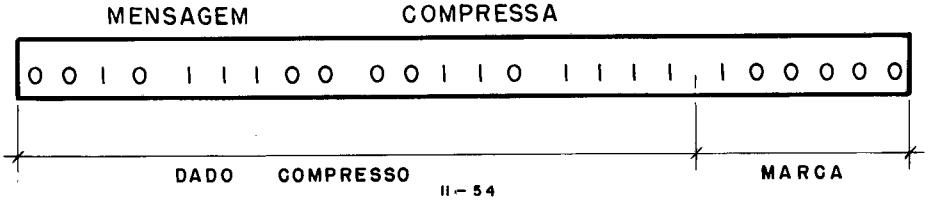
a) Obter da mensagem original o próximo caráter da esquerda para a direita.

b) Tomando-se este caráter como argumento de pesquisa retirar da tabela de códigos, a configuração binária e seu comprimento (número de bits). Convém observar que o método de pesquisa à tabela deve ser desenvolvido tendo em vista um equilíbrio entre velocidade e memória gasta, conforme as necessidades do usuário.

c) Armazenar na área destinada a mensagem compressa o grupo de bits obtido em "b", conforme estipulado pelo comprimento.

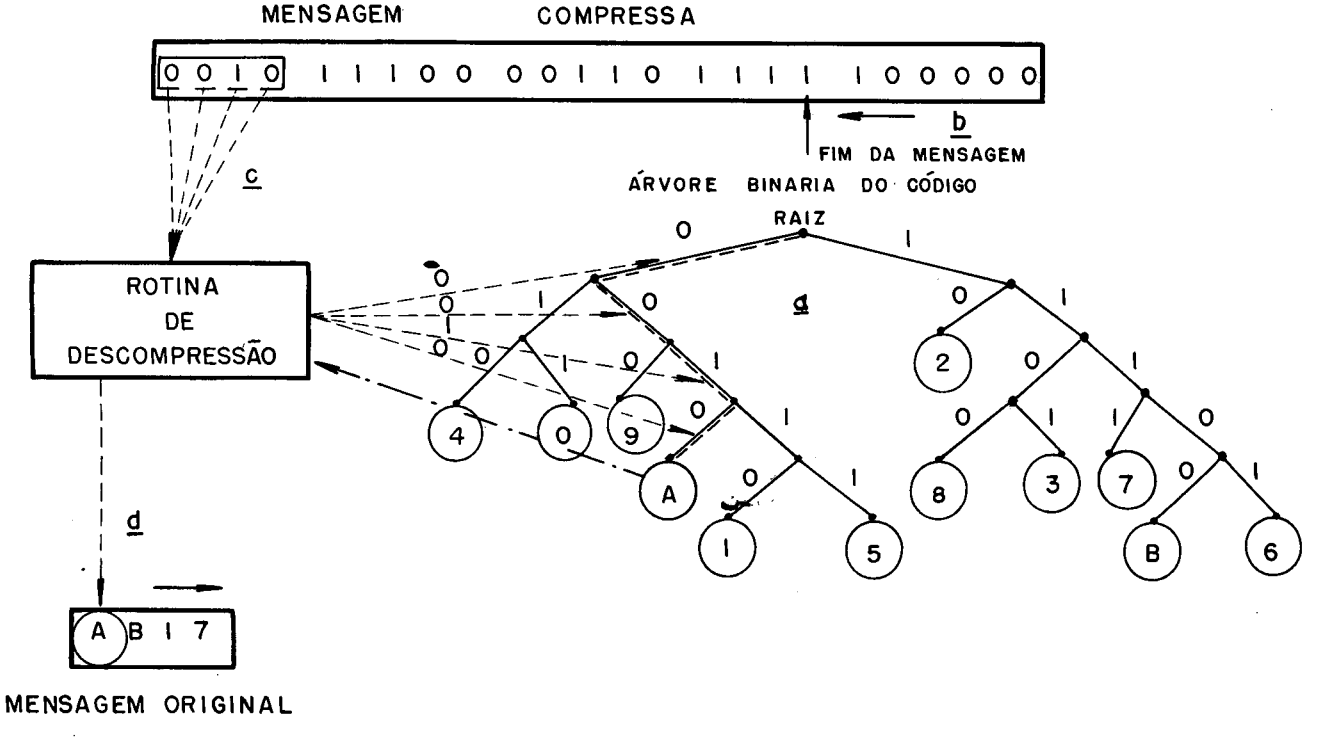
d) Retornar ao procedimento "a".

Este ciclo se repetirá tantas vezes quantos sejam os elementos da mensagem original, quando então para indicar o fim da mensagem compressa e completar um número inteiro de bytes, gravamos uma marca. Das várias maneiras existentes de se marcar uma mensagem, escolhemos uma onde se atribui o valor "1" ao próximo dígito binário, preechendo-se com "0" os demais bits até se completar um número inteiro de bytes para a mensagem compressa.



II-54

Rotina de Descompressão:



II-55

Procedimentos de descompressão:

a) Como vimos anteriormente, quando da determinação dos códigos de tamanho variável, sempre obtínhamos ao final do processo uma árvore binária, que nos permitia através dos seus vários caminhos obter todos os grupos de um código. Esta característica é agora utilizada de modo a possibilitar a decodificação de mensagens compressas.

Por esta razão este procedimento se constitui na obtenção desta árvore, o que se torna possível pela propriedade do prefixo pertinente a estes códigos.

b) Determinar o fim da mensagem compressa, pelo reconhecimento do primeiro dígito binário de valor "1" ao se processar a mensagem bit a bit da direita para a esquerda.

c) A partir do início e da esquerda para a direita, processar a sequência de bits modo que a cada bit obtido, dependendo do seu valor ("1" ou "0"), vamos percorrendo a árvore binária do código pelo ramo esquerdo ou direito até ser atingido um nó terminal (folha).

d) Uma vez detectada uma folha, retiramos o caráter aí contido e o movemos para a mensagem descompressa. Retornar ao procedimento "c".

Este processo será repetido até ser encontrado o fim da mensagem compressa.

Os procedimentos de compressão e descompressão que acabamos de apresentar se aplicam de uma forma geral a todo e qualquer método, que se utilize dos códigos de tamanho variável. Porém se analisarmos mais detidamente, vamos observar que estes processos pela dimensão de suas tabelas, necessitam para sua implantação de uma quantidade razoável de memória, o que nem sempre está disponível na quantidade desejada. Por esta razão, o código HSF, por meio de suas propriedades, nos oferece uma alternativa que permite o desenvolvimento de rotinas mais otimizadas quanto à utilização de memória. Esta otimização se deve ao fato de se ter idealizado para os procedimentos de compressão e descompressão, um conjunto de tabelas de tamanho mais reduzido como veremos a seguir:

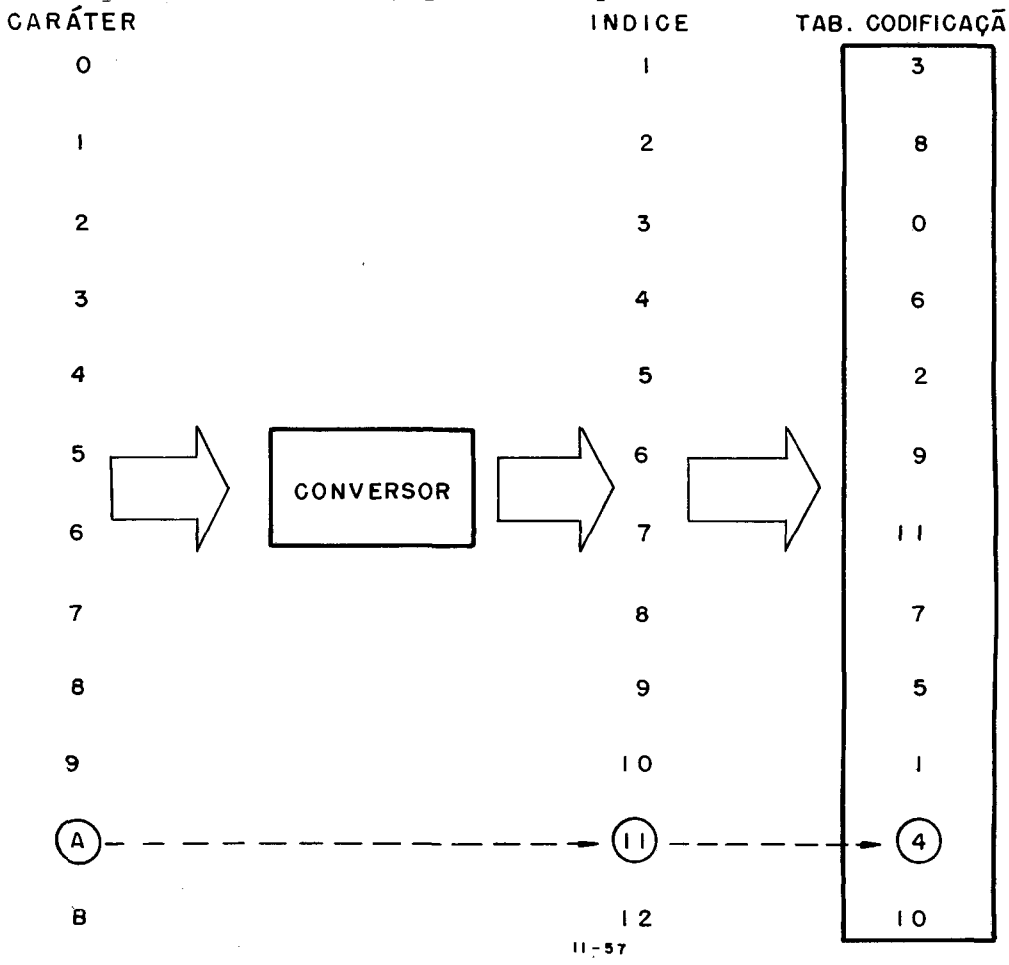
a) Tabela de Codificação

Esta tabela associa um número de ordem a cada caráter obtido da mensagem original. Este número de ordem é obtido através da atribuição de um número sequencial a partir de zero a todos os elementos do alfabeto organizados em ordem decrescente de probabilidade.

CARÁTER	PROBABILIDADE	NÚMERO ORDEM
2	0.226	0
9	0,165	1
4	0.135	2
0	0.120	3
A	0.079	4
8	0.063	5
3	0.054	6
7	0.041	7
1	0.038	8
5	0.034	9
B	0.030	10
6	0.015	11

II-56

Supondo-se que o usuário já tenha definido uma forma de traduzir o carater fonte no índice de entrada na Tabela de Codificação (conversor), podemos apresentá-la como se segue:

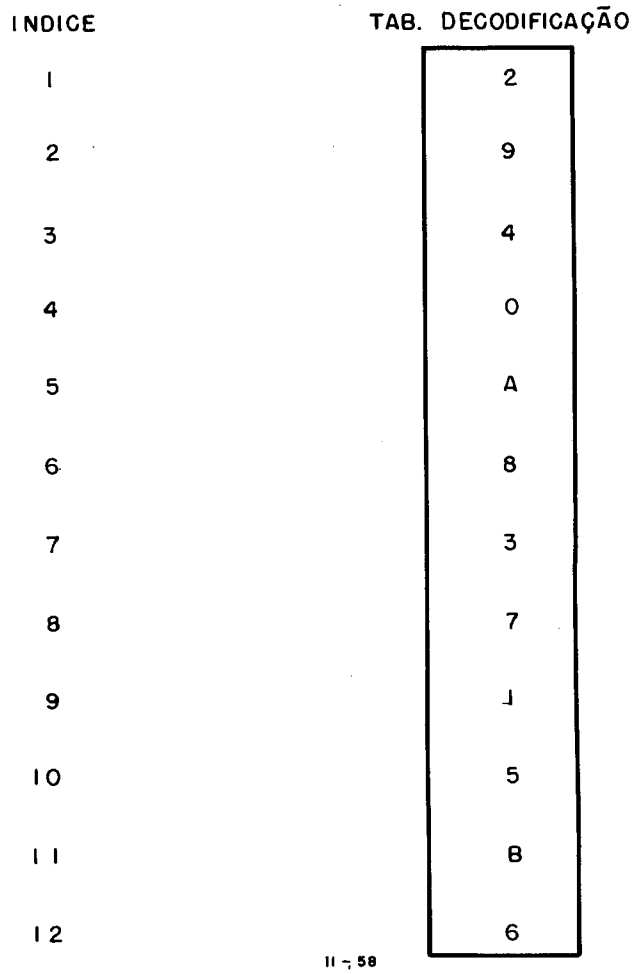


II-57

Assim no exemplo acima, ao caráter "A" corresponderia o índice "11" e o número de ordem 4, indicando que o "A" é o quinto caráter mais frequente na amostra.

b) Tabela de Decodificação

Esta tabela apenas relaciona os componentes do alfabeto em ordem decrescente de probabilidade.



c) Tabela de Codificação/Decodificação

Esta é uma tabela bidimensional utilizada pelas rotinas de compressão e descompressão onde o número de linhas corresponde ao comprimento do maior grupo (número de bits) e as colunas, em número de 3, apresentam os valores abaixo:

c.1. Valores Limite - Esta coluna consta dos valores decimais correspondentes aos grupos de maior valor numérico em cada conjunto de grupos de mesmo comprimento no código. A sua determinação é feita através das fór

mulas abaixo:

$$L_n = T_n - 1$$

$$L_i = 2 \times L_{i-1} + T_i + 1$$

onde i - N + 1, N+ 2, N + 3

L_i - Traduz o Valor Limite dos grupos de comprimento "i".

L_{i-1} - Traduz o Valor Limite dos grupos de comprimento "i - i".

T_i - Representa o número de grupos no código de comprimento "i".

c.2. Valores Base - Uma constante para cada conjunto de grupos do mesmo comprimento. O Valor Base nos fornece o grupo desejado quando adicionado ao Número de Ordem representativo do caráter fonte.

c.3. Número de Ordem - É a relação dos números de ordem atribuídos aos caracteres correspondentes aos Valores Limites de cada conjunto de grupos de mesmo comprimento.

Assim, podemos escrever

$$R_i = R_{i-1} + T_i$$

onde i - N + 1, N + 2, ..., M

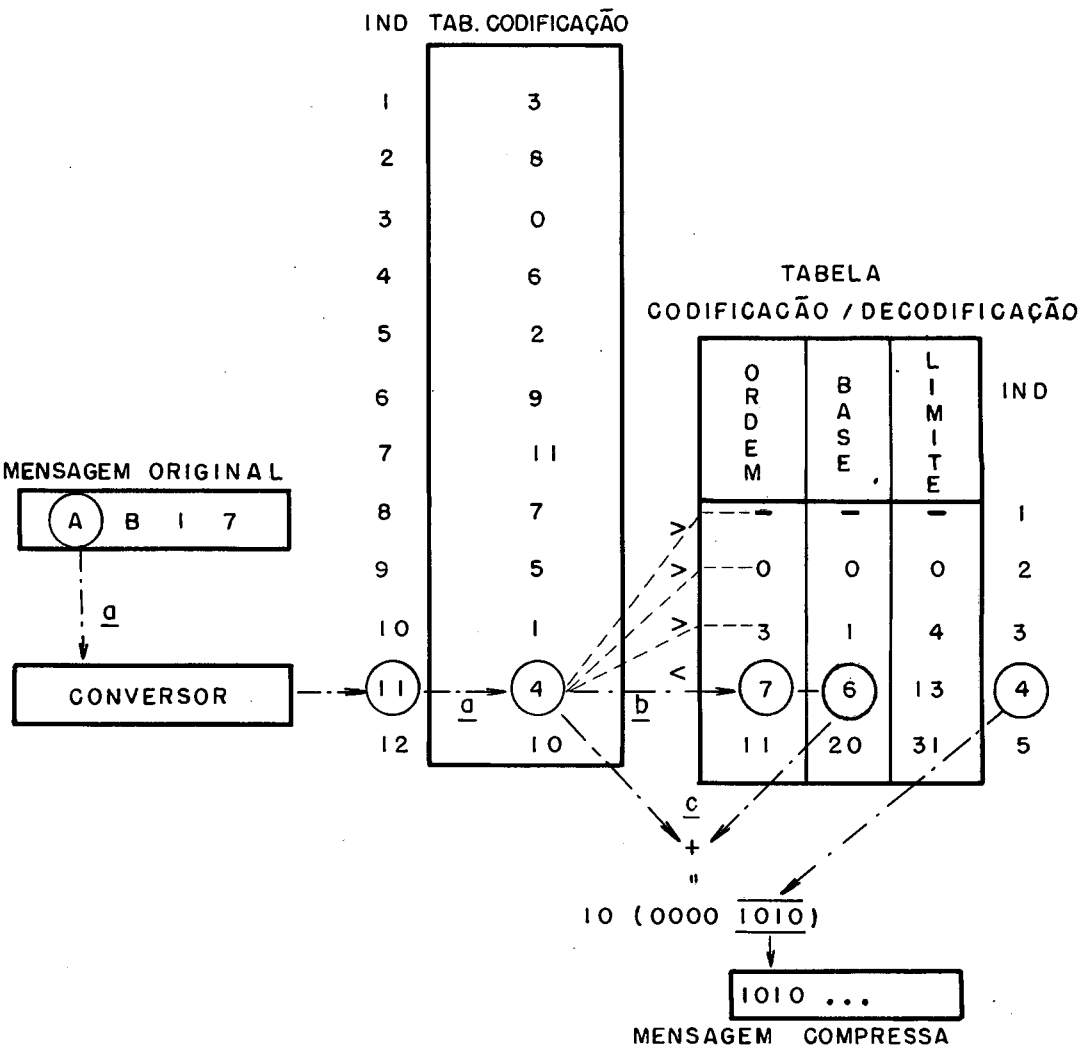
R_i - Número de Ordem atribuído ao caráter correspondente ao Valor Limite dos grupos de comprimento "i".

R_{i-1} - Número de Ordem atribuído ao caráter correspondente ao Valor Limite dos grupos de comprimento "i - 1".

Tomando-se nosso exemplo, teríamos para Limite, Base e Número de Ordem os valores a seguir:

CARÁTER	CÓDIGO HSF	VALOR DECIMAL	NÚMERO DE ORDEM	CODE INDEX	VALOR LIMITE	VALOR BASE	NÚMERO DE ORDEM
—	—	—	—	0	—	—	—
2	00	0	0	1	0	0	0
9	010	2	1				
4	011	3	2				
0	100	4	3	3	4	1	3
A	1010	10	4				
8	1011	11	5				
3	1100	12	6				
7	1101	13	7	4	13	6	7
1	11100	28	8				
5	11101	29	9				
B	11110	30	10				
6	11111	31	11	4	31	20	11

Rotina de Compressão Utilizando o Código HSF:



II-60

Procedimentos de Compressão utilizando-se o Código HSF:

a) Obter o próximo caráter fonte da mensagem original, transformando-o em seguida através do conversor em um índice. Com este valor acessamos a Tabela de Codificação, retirando-se então o Número de Ordem correspondente ao caráter fonte.

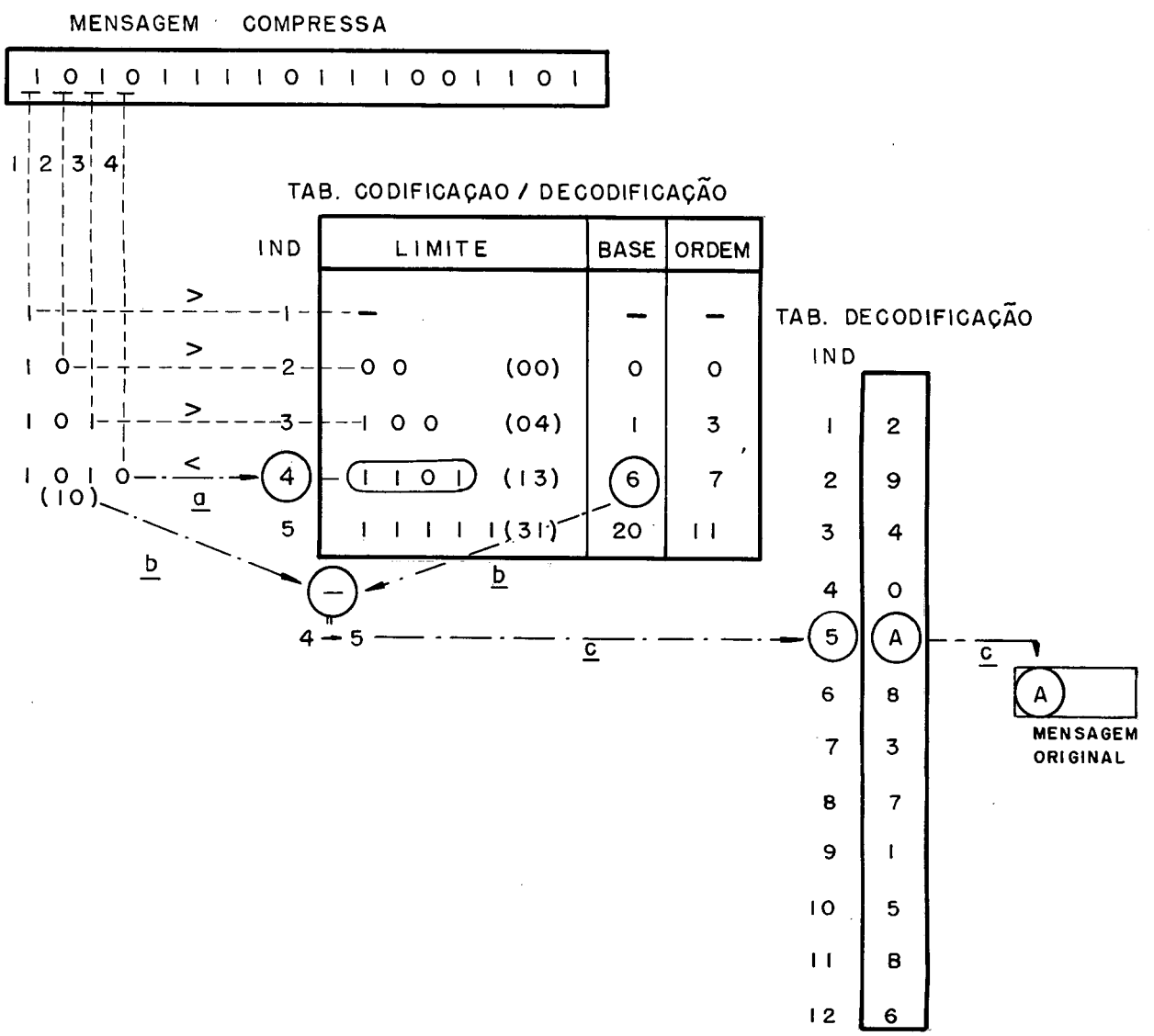
b) Pesquisar sequencialmente a Tabela de Codificação/Decodificação, até que o Número de Ordem obtido do item "a" seja menor ou igual ao Número de Ordem da tabela. Neste ponto temos o Valor Base e no índice da tabela o comprimento do grupo.

c) Uma vez que temos o Número de Ordem obtido em "a" e o Va-

lor Base obtido no item "b", bastará adicionarmos estas 2 quantidades para termos o valor decimal do grupo desejado. Para a retirada do grupo basta tomarmos o índice obtido em "b", cujo conteúdo traduz o comprimento do grupo, e extrairmos da direita para a esquerda tantos bits quanto seja o conteúdo do índice.

Este processo se repetirá até serem processados todos os elementos da mensagem original.

Rotina de Descompressão utilizando-se o Código HSF:



Procedimentos de Descompressão Utilizando-se o Código HSF:

a) Agrupar um a um os bits da sequência de entrada, até que se encontre uma configuração cujo valor seja menor ou igual ao Valor Limite para os grupos deste comprimento. Neste ponto tomamos o Valor Base correspondente ao Valor Limite encontrado.

b) Visto que temos o Valor Base e o valor decimal da configuração de bits obtidos da sequência de entrada, basta então subtrair o primeiro do segundo e obteremos o Número de Ordem do caráter original.

c) Através do Número de Ordem indexamos a Tabela de Decodificação e retiramos o carater desejado.

Este processo será repetido até ser encontrado o fim da mensagem compressa.

VANTAGENS

a) Os códigos de tamanho variável, possibilitam de maneira geral compressões drásticas aos arquivos, pela redução do número médio de bits para representar cada caráter.

b) Este é um método independente da estrutura e do conteúdo dos dados e por esta razão as rotinas podem ser desenvolvidas para satisfazer qualquer aplicação.

c) Maior segurança nas informações pertencentes ao arquivo, uma vez que estas se encontram registradas de uma forma ilegível, só tendo acesso ao seu conteúdo pessoas que conheçam o código empregado com seus alfabetos e grupos.

d) A utilização deste método nos permite com vantagens o acoplamento a outras técnicas como eliminação de caracteres repetidos, compressão por mapa de bits (segmentação),etc, resultando daí uma possível redução no consumo de tempo de CPU, pela diminuição da parte útil dos dados.

e) Com relação às vantagens particulares de cada código, preferimos apresentá-las anteriormente, quando da descrição de cada um deles.

DESVANTAGENS

a) Por ser este método fundamentado nas propriedades estatísticas da informação, poderá, com o passar do tempo, ocorrer alterações nestas propriedades, trazendo como consequência uma redução na eficiência do código. Por esta razão é interessante se prever programas utilitários, que possam quando desejado reavaliar estas propriedades, gerando novo código.

b) Se por um lado é conveniente se ter um arquivo cujo conteúdo é ilegível para estranhos, por outro é inconveniente quando se está acostumado a fazer uso de utilitários para processar estes arquivos.

c) Maior complexidade das rotinas que manuseiam o arquivo compresso pois estas terão de trabalhar com registros e campos de comprimento variável.

d) Normalmente as necessidades mínimas de memória e tempo de CPU destes métodos não são desprezíveis, o que nos leva a uma série de restrições para a sua implantação.

4.2. COMPRESSÃO POR CODIFICAÇÃO CONDICIONAL

DESCRIÇÃO

Como vimos no método anterior, cada caráter era codificado por um grupo de bits, cujo o comprimento variava em função da sua probabilidade de ocorrência na amostra.

Neste método, através da utilização dos mesmos códigos de tamanho variável definidos anteriormente, cada caráter também é codificado por um grupo de bits de comprimento variável, considerando-se porém a probabilidade de ocorrência do respectivo caráter condicionada ao caráter precedente.

Assim, suponhamos uma amostra composta apenas da palavra

B A R B A R I D A D E

onde, ao tomarmos a letra "A" isoladamente e calcularmos a sua probabilidade de ocorrência, obteremos o resultado de 3/11 ▫ 0,27 e se considerarmos a mesma letra "A" agora condicionada ao caráter "D" sua nova probabilidade de ocorrência será 1/2 ▫ 0,50, pois como podemos constatar o caráter "D" aparece 2 vezes na amostra, seguido uma vez pela letra "A" e outra pela letra "E".

Pelo o que acabamos de expor, podemos concluir,que provavelmente o comprimento do grupo a ser atribuído ao caráter "A" considerando-o isoladamente será maior que o comprimento do grupo a ser atribuído ao caráter "A" condicionando-o ao carater "D".

SISTEMÁTICA DE IMPLANTAÇÃO

Para se estudar a implantação do método, vamos apresentar separadamente todas as fases componentes do mesmo, de modo a dar ao leitor uma visão clara do que seja o processo, tanto na sua estrutura como no seu funcionamento.

a) Determinação do alfabeto

Tendo em vista o tipo do dado, o formato e sua utilização, determinamos o conjunto de caracteres que formarão o nosso alfabeto. Para o nosso exemplo vamos supor um alfabeto constituído dos caracteres A, B,C e E.

b) Determinação das probabilidades de ocorrência

Uma vez fixado o alfabeto, o nosso próximo passo se constituirá na obtenção das probabilidades de ocorrência dos caracteres condicionados aos seus precedentes.

Primeiramente, procedemos a determinação das probabilidades de ocorrência para o caráter inicial do dado, pois por ser o primeiro não possui precedente, para em seguida calcularmos as demais probabilidades condicionadas a cada caráter do alfabeto. Desta forma ao final deste procedimento, teremos tantas tabelas quantas sejam os elementos do alfabeto mais 1.

TAB. PROB. OCORR.
1º CARÁTER

CARAT.	PROB.
B	0.60
A	0.30
C	0.05
E	0.05

TAB. PROB. OCORR.
CONDICIONADO "A"

CARAT.	PROB.
B	0.65
C	0.30
E	0.05

TAB. PROB. OCORR.
CONDICIONADO "B"

CARAT.	PROB.
A	0.40
E	0.35
C	0.25

TAB. PROB. OCORR.
CONDICIONADO "C"

CARAT.	PROB.
A	0.60
E	0.30
B	0.10

TAB. PROB. OCORR.
CONDICIONADO "E"

CARAT.	PROB.
C	0.70
B	0.25
A	0.05

II-62

c) Aplicação de um código de comprimento variável

Uma vez de posse de todas as tabelas, com os respectivos alfabetos e probabilidades de ocorrência, aplicamos a cada uma delas o código de tamanho variável escolhido.

Deste procedimento, obtemos um código para cada tabela de probabilidade.

Para o nosso exemplo utilizamos o código de Huffman.

1º CARÁT.

CRC	CÓDIGO	CMP
B	0	1
A	10	2
C	110	3
E	111	3

CONDICIONADO "A"

CRC	CÓDIGO	CMP
B	0	1
C	10	2
E	11	2

CONDICIONADO "B"

CRC	CÓDIGO	CMP
A	0	1
E	10	2
C	11	2

CONDICIONADO "C"

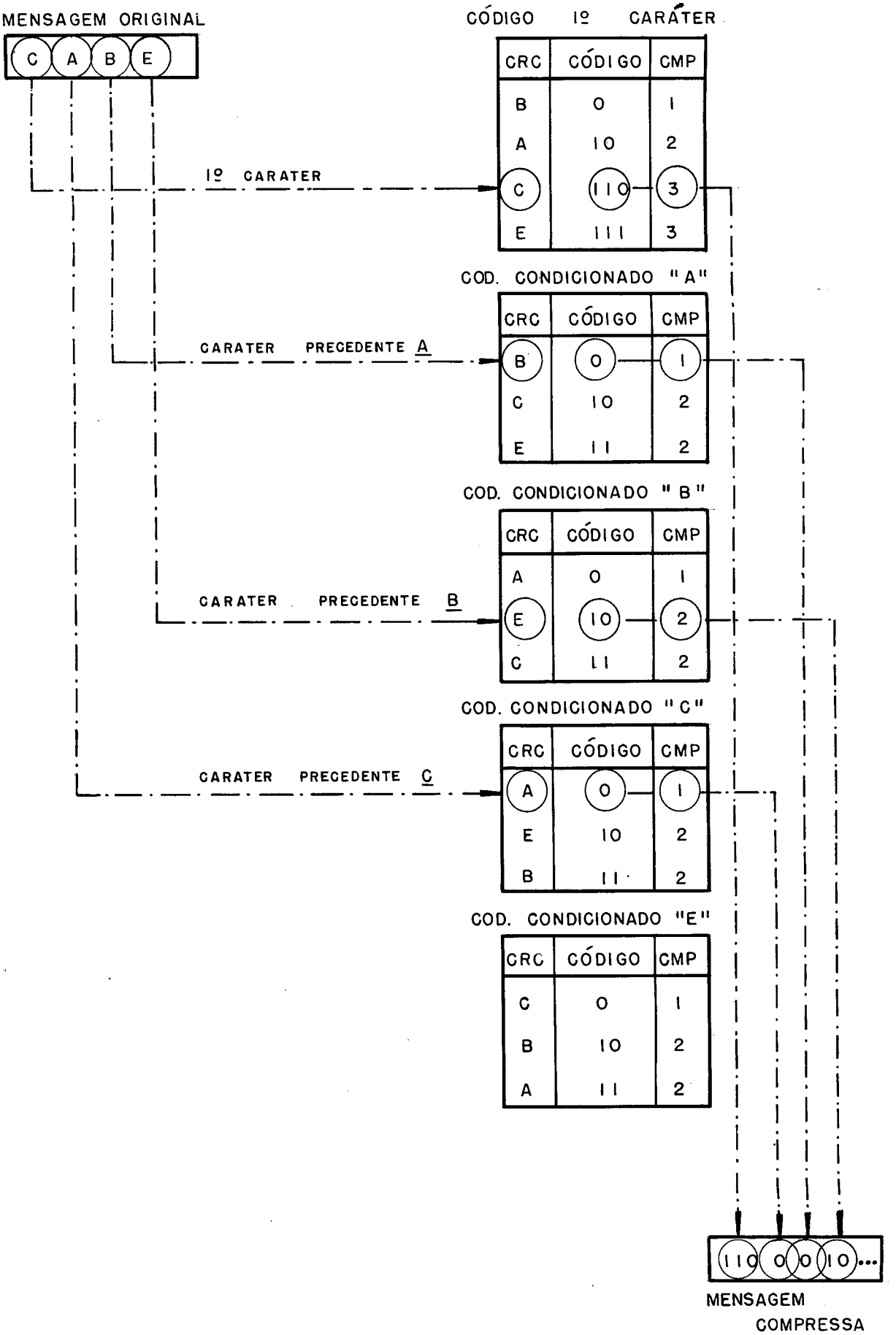
CRC	CÓDIGO	CMP
A	0	1
E	10	2
B	11	2

CONDICIONADO "E"

CRC	CODIGO	CMP
C	0	1
B	10	2
A	11	2

II-63

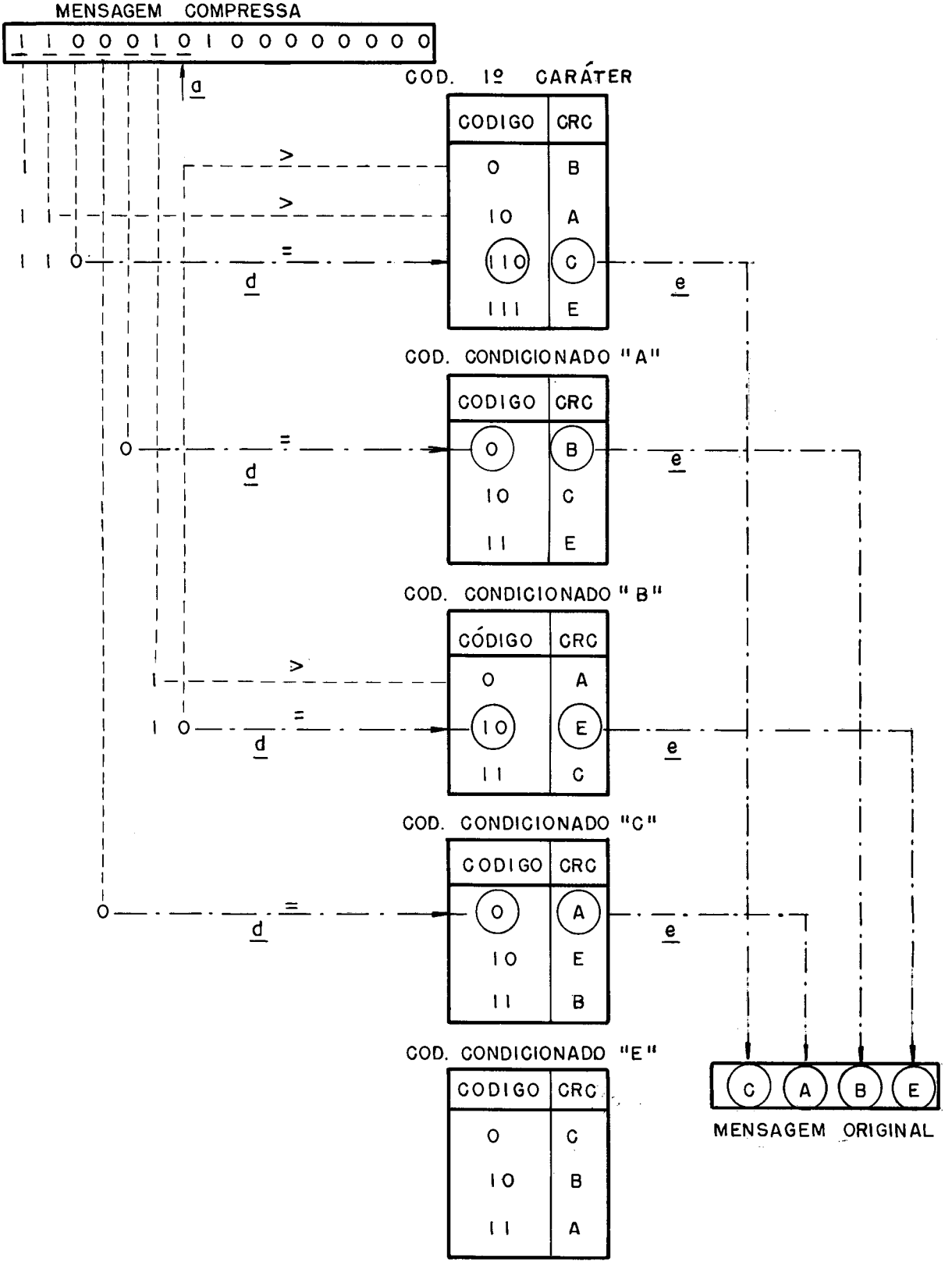
Rotina de Compressão:



Procedimentos de Compressão:

A rotina de compressão, de alguma forma, deverá conter todas as tabelas obtidas no item "c" com os respectivos códigos e tamanho, para que durante a execução,a cada dado fornecido, possamos codificar todos os caracteres conforme o caráter precedente, que apontará a tabela de onde devemos retirar o grupo desejado.

Rotina de Descompressão:



Procedimentos de Descompressão:

Como na rotina de compressão, a rotina de descompressão, deverá conter também todas as tabelas necessárias para sua execução.

Para minimizar as necessidades de memória é conveniente que as rotinas de compressão e descompressão componham um único módulo, pois desta forma ambas utilizarão o mesmo conjunto de tabelas.

A rotina de descompressão pode ser desenvolvida de 2 maneiras diferentes: a primeira consiste no mesmo processo empregado para os códigos de tamanho variável, porém, devido trabalharmos com vários códigos simultaneamente, seria necessário uma quantidade de memória razoável para armazenar todas as árvores binárias e/ou tabelas referentes aos códigos existentes, sob pena de degradar consideravelmente a performance do sistema; a segunda, embora menos eficiente, necessita de menos memória para a sua execução e basicamente compõe se dos procedimentos abaixo:

a) Determinar o fim da mensagem compressa e marcá-la.

   Em nosso exemplo usamos a mesma marca utilizada por ocasião da apresentação dos códigos de tamanho variável.

b) Selecionar a tabela de códigos em função do último caráter decodificado. Caso o caráter seja o primeiro, tomaremos a tabela correspondente ao primeiro caráter.

c) Obter da esquerda para a direita o próximo bit da mensagem compressa.

d) Pesquisar sequencialmente na tabela selecionada em "b", todos os grupos de 1 bit, se não existir igual, obter o próximo bit da mensagem compressa e anexá-lo ao bit obtido anteriormente, pesquisar sequencialmente na mesma tabela todos os grupos de 2 bits, se não encontrar anexar o próximo bit da mensagem e proceder a pesquisa sequencial dos grupos de 3 bits. Este procedimento se repetirá até encontrarmos o grupo desejado.

e) Uma vez identificado o grupo na tabela, obtemos o caráter correspondente e o movemos para a mensagem descompressa.

f) Voltar ao procedimento "b".

Este ciclo se repetirá até encontrar a marca que indica o fim da sequência de entrada.

VANTAGENS/DESVANTAGENS

As vantagens e desvantagens deste método, coincidem com as apresentadas, por ocasião da exposição do método anterior, pois ambos trabalham com códigos de tamanho variável.

Porém devemos acrescentar aqui a grande necessidade de memória para armazenamento das tabelas necessárias à execução do método, o que poderá se tornar um ponto crítico, caso se tente a generalização, condiconando cada caráter da amostra a 2 ou mais caracteres precedentes.

CAPÍTULO III

O SISTEMA

1. APRESENTAÇÃO DO SISTEMA

1.1. INTRODUÇÃO

Após o estudo que acabamos de realizar, onde os principais métodos de compressão foram abordados e analisados e uma vez que estamos plenamente conscientes das nossas dificuldades com relação ao armazenamento de dados, podemos agora, de forma definitiva, partir para o desenvolvimento de um Sistema para Compressão de Dados.

Neste capítulo, vamos procurar dar ao leitor uma visão geral do sistema, quando apresentaremos em todos os seus aspectos a definição, implantação e operação do mesmo.

1.2. OBJETIVOS

Neste sistema procuramos de uma forma eficiente comprimir e descomprimir dados organizados em arquivos sequenciais independentemente de suas estruturas e/ou conteúdos, com um mínimo de alteração nos programas de aplicação e um máximo de transparência para os programadores da instalação.

Assim, sempre que as necessidades de uma aplicação justificarem a utilização da Compressão de Dados, poderemos através de um esforço mínimo de pessoal e num curto espaço de tempo implantar este sistema.

1.3. FACILIDADES

Para o cumprimento dos objetivos estabelecidos anteriormente, o sistema dispõe das seguintes facilidades:

a) Estatística para Segmentação

Aqui por meio de 3 relatórios, fornecemos para cada estrutura encontrada na amostra e para cada campo do registro, um conjunto de informações, contendo dados esta

tísticos e de compressão, os quais permitem ao usuário, através de uma redistribuição dos campos, segmentar o registro de uma forma mais eficiente e objetiva.

b) Código de Tamanho Variável - HSF

A utilização desta técnica, já apresentada anteriormente, torna o sistema de aplicação geral, dotando-o de uma alta eficiência, embora comprometida com a estabilidade estatística dos dados.

Para sanar este problema o sistema foi desenvolvido de forma a possibilitar ao usuário a reavaliação destas propriedades e consequentemente dos códigos, alfabetos, segmentos, etc, sempre que necessário.

c) Subrotinas de Compressão e Descompressão

O fato de que a compressão e descompressão efetiva dos dados seja feita pela utilização de subrotinas, as quais efetuam a leitura e gravação dos arquivos compressos, facilita em muito, a implantação do sistema, uma vez que ao usuário caberá basicamente, substituir nos programas de aplicação os comandos de leitura e gravação do arquivo agora compresso, por chamadas a estas subrotinas.

d) Automatização da Execução

A automatização da execução permite a utilização da Compressão de Dados em qualquer aplicação de um modo totalmente transparente ao usuário, pois, como iremos ver, o sistema é o único responsável pela geração, armazenamento e posterior recuperação de todos os dados necessários a sua execução.

1.4. CARACTERÍSTICAS GERAIS

O sistema apresenta características básicas que devem ser rigorosamente observadas no decorrer de toda a implantação, como se verá a seguir.

a) Os arquivos a serem compressos deverão:

- Possuir organização sequencial.

- Utilizar qualquer suporte de armazenamento com algumas alterações que serão mencionadas no momento oportuno.
- Conter apenas registros de tamanho fixo e todos do mesmo tipo (lay-out).

b) O sistema projetado deverá:

- Utilizar como linguagens, dependendo das características do problema a ser solucionado, Cobol ou Assembler.
- Comprimir ou descomprimir o registro integralmente.
- Ser totalmente transparente aos usuários, quer seja ao nível de análise, programação ou operação.
- Não comprimir segmentos binários.
- Ser limitado aos valores abaixo:
  - Registro lógico descompresso máximo - 512 bytes
  - Número máximo de segmentos - 64
  - Número máximo de campos - 120
  - Bloco no arquivo compresso - 1024 bytes
  - Comprimento máximo de um campo - 256 bytes.

c) Nos arquivos compressos deveremos encontrar:

- Inicialmente todas as tabelas, códigos e constantes necessários a compressão/descompressão efetiva dos dados.
- Dados do arquivo, compressos.

2. DEFINIÇÃO DO SISTEMA

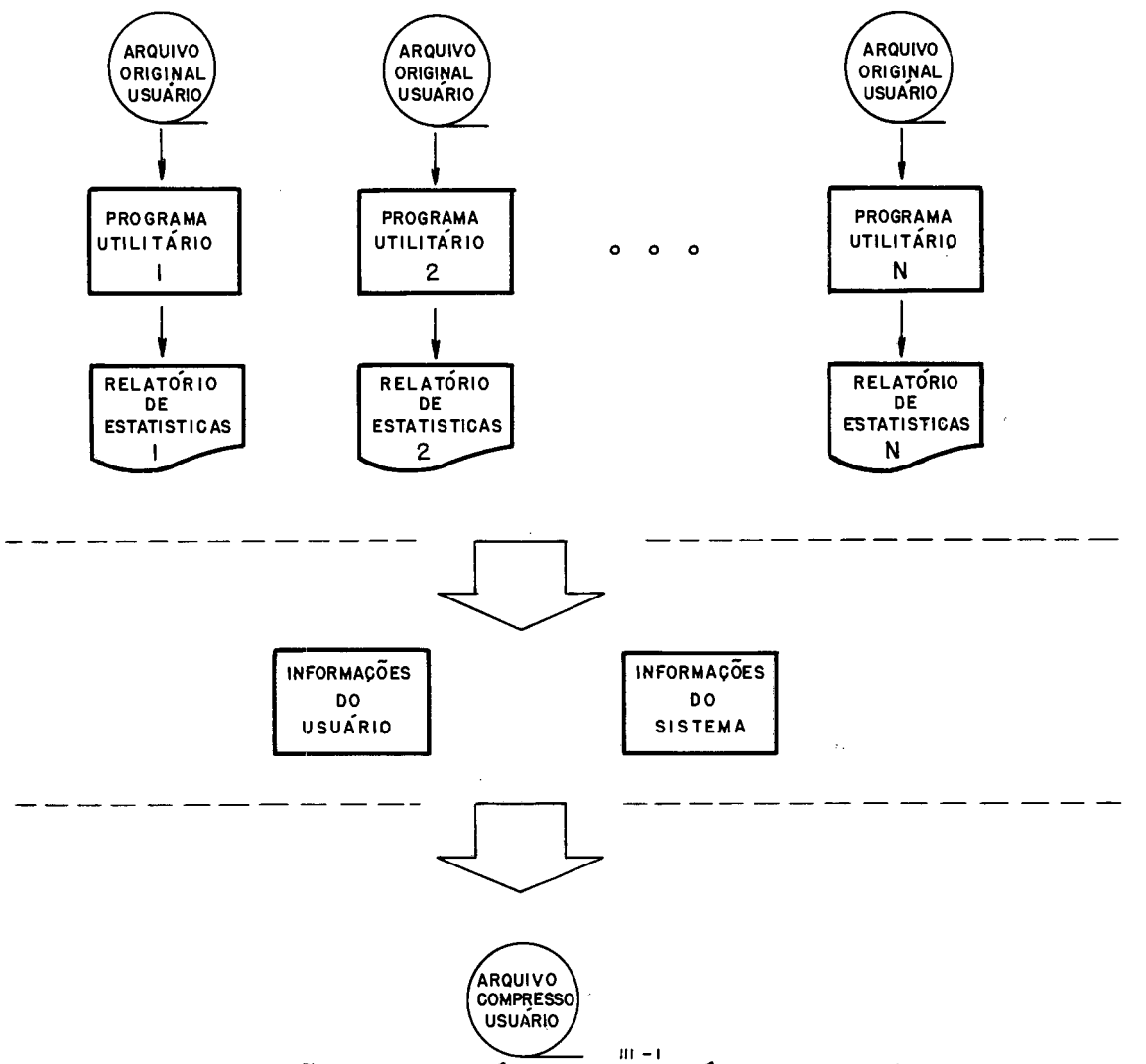
2.1. ESTRUTURA DO SISTEMA

Um sistema desta natureza, a nosso modo de ver, compreende 2 etapas distintas, que convencionamos chamar de "MÓDULO DE PREPARAÇÃO" e "MÓDULO DE EXECUÇÃO".

O Módulo de Preparação, que reúne uma série de programas utilitários, procura através da manipulação estatísti

ca dos dados na sua forma original, fornecer informações ao usuário e ao próprio sistema. Estas informações permitem eleger uma melhor taxa de compressão e um maior índice de automatização e tanto orientam na definição como definem os parâmetros necessários ao funcionamento do sistema.

MÓDULO DE PREPARAÇÃO

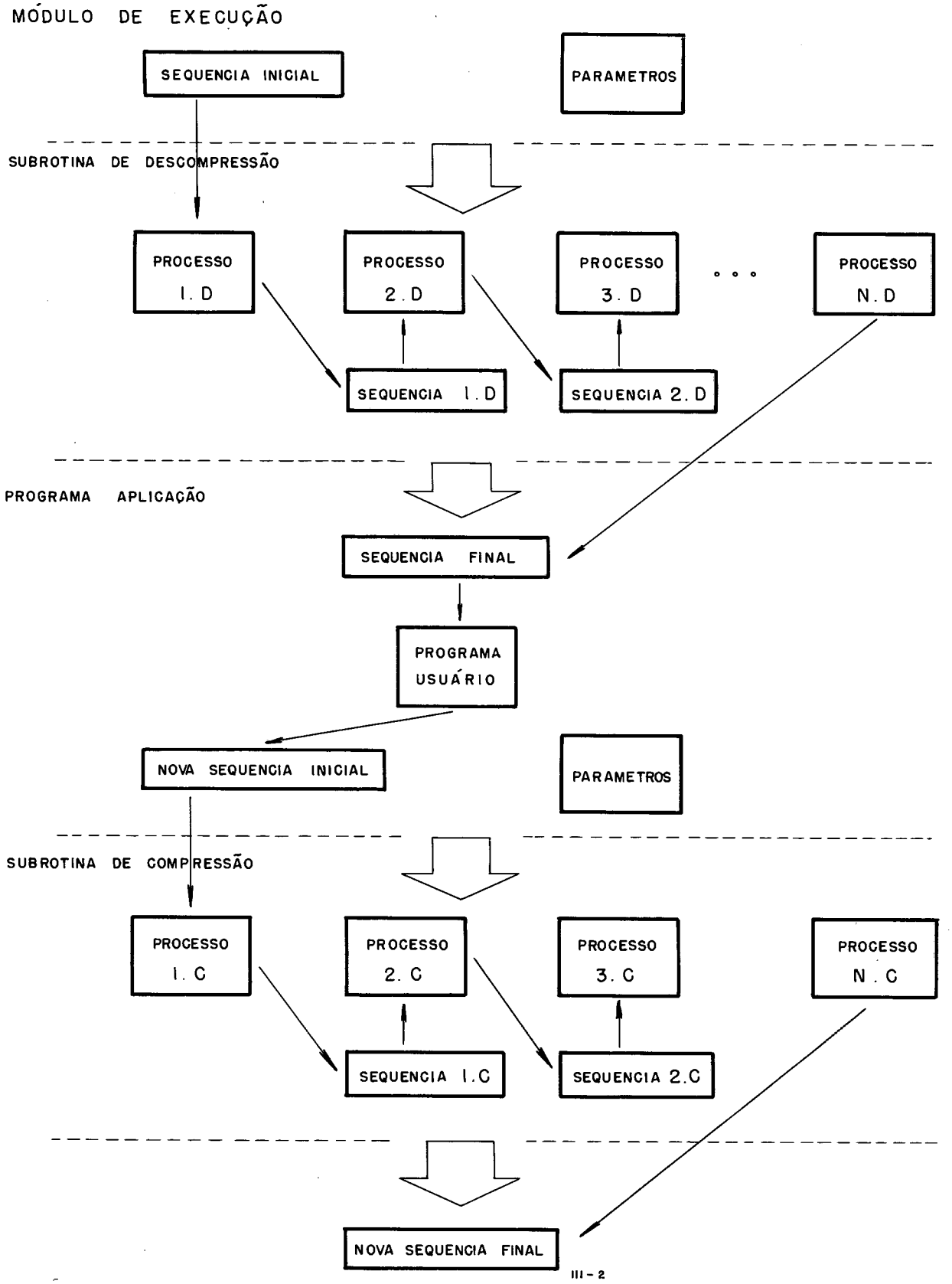


III-1

A execução deste módulo deverá ocorrer integral ou parcialmente, sempre que por alguma razão (implantação ou reestruturação do sistema), se tenha que avaliar as propriedades estatísticas dos dados e consequentemente todos os parâmetros necessários a execução.

O Módulo de Execução, que reúne as subrotinas de Compressão e Descompressão, é o responsável pela compressão e/ou descompressão efetiva dos dados para os programas do usuário. Aqui encontramos uma sequência de processos, onde a partir de parâmetros definidos no Módulo de Preparação, cada processo toma como entrada uma sequência de símbolos em um

alfabeto resultante do processo anterior e a transforma em uma nova sequência em outro alfabeto, que será utilizado como entrada no processo seguinte, caso não seja a sequência final.



III - 2

2.2. MÓDULO DE PREPARAÇÃO DO SISTEMA

Para o nosso sistema, o Módulo de Preparação ficou constituído de 3 programas utilitários, que são responsáveis diretamente por toda a transição dos dados de sua forma original para a forma compressa, incluindo a determinação de todas as propriedades e parâmetros necessários ao funcionamento do Módulo de Execução.

A cada programa atribuimos os procedimentos abaixo relacionados:

a) Programa Gerador de Estatísticas para Segmentação (COMPØ3Ø)

Este programa tem por finalidade, a partir da definição dos campos no registro, processar uma amostra representativa destes registros, apresentando:

- Todas as estruturas presentes.
- Comportamento estatístico e de compressão para cada uma das estruturas encontradas, onde dados como frequência, probabilidade de ocorrência, compressão absoluta, compressão relativa, compressão percentual, etc, são apresentados.
- Comportamento estatístico e de compressão para cada um dos campos definidos no registro, onde dados como frequência, probabilidade de ocorrência, participação total na amostra, participação compressa na amostra, etc, são apresentados.

Todas as informações acima são registradas em 3 relatórios, os quais, emitidos ao final e a intervalos regulares (especificados pelo usuário), durante a execução do programa, fornecem ao usuário todos os dados necessários para se alcançar a combinação de campos ideal para cada segmento, ou seja, aquela que produzirá a maior taxa de compressão.

b) Programa Gerador de Códigos (COMPØ4Ø)

Principal utilitário do sistema, este programa,

a partir de dados fornecidos pelo usuário referentes ao registro e seus segmentos, projetados com o auxílio dos relatórios do programa anterior, passa a processar o arquivo desejado, reagrupando, a cada registro, os campos em seus respectivos segmentos e emitindo ao final e a intervalos regulares durante a execução, relatórios que demonstram:

- Comportamento estatístico e de compressão dos segmentos, onde dados como probabilidade de ocorrência, compressão absoluta, compressão relativa, raio de compressão, etc, são apresentados.
- Código HSF gerado para cada segmento, acrescido de dados como custo, compressão absoluta, compressão relativa, compressão percentual, etc, que revelam a compressão resultante da aplicação do código.

Com estas informações, o usuário fica em condições de avaliar com bastante precisão, a compressão resultante tanto da eliminação dos segmentos sem conteúdo no arquivo, como da aplicação do código HSF aos segmentos restantes.

Uma vez que podem ser desenvolvidos diferentes projetos para o registro e seus segmentos é possível, através de uma análise comparativa dos relatórios emitidos para cada caso, escolher o projeto mais eficiente ficando, desta forma, assegurada uma grande flexibilidade ao sistema.

O programa termina sua execução gravando um arquivo onde são encontradas todas as informações necessárias a execução das Subrotinas de Compressão/Descompressão como Tabelas, alfabetos, códigos, etc.

É conveniente observar, ser este arquivo um dos principais responsáveis pelo alto índice de automatização e transparência do sistema, uma vez que é gerado e utilizado pelo próprio sistema sem nenhuma interferência dos responsáveis pela aplicação.

c) Programa de Conversão (COMPØ5Ø)

A finalidade deste programa é a de comprimir ou descomprimir arquivos, utilizando-se para tanto do Módulo de Execução, ou seja, das Subrotinas de Compressão/Descompressão, e do arquivo de tabelas e códigos gerados pelo programa anterior (COMPØ4Ø).

2.3. MÓDULO DE EXECUÇÃO DO SISTEMA

2.3.1. Rotinas de Compressão/Descompressão

Como já vimos, o Módulo de Execução, que engloba um ou mais processos de compressão, é o responsável direto pela compressão/descompressão efetiva dos dados para os programas do usuário.

Para o nosso sistema, este módulo está constituído de uma rotina de compressão e uma de descompressão, ambas compondo uma única subrotina com 2 entradas o que possibilita, executar um ou outro procedimento. Assim, com esta estrutura, podemos tirar proveito dos seguintes pontos:

a) Economia de memória pelo armazenamento de apenas uma versão de todas as tabelas comuns a ambas as rotinas.

b) Simplicidade de comunicação entre as rotinas, possibilitando que, tanto no sistema atual como em um sistema futuro, possamos obter um alto grau de automatização com uma maior eficiência na execução.

c) Maior facilidade de implantação e manutenção uma vez que teremos de manusear, apenas uma subrotina, que congrega todos os procedimentos necessários à execução das rotinas de compressão e descompressão.

Como todo o sistema, estas rotinas foram desenvolvidas de modo a atender qualquer aplicação que necessite reduzir o volume de informação de alguns de seus arquivos.

Visto isto e mais os objetivos que temos de atender, escolhemos para o nosso sistema as Técnicas de Compressão por Segmentação (mapa de bits) e Códigos de Tamanho Va

riável (HSF).

Assim da combinação destas 2 técnicas se desenvolveu um processo em que, a partir do registro segmentado, aplicamos o Método de Compressão por Segmentação, eliminando todos os segmentos sem conteúdo deste registro, para, em seguida, aplicar, aos segmentos restantes, o Código de Tamanho Variável - HSF, correspondente a cada um.

Desta forma, conseguimos alcançar um maior índice de compressão e uma melhor performance do sistema, uma vez que apenas os segmentos com conteúdo serão compressos utilizando o Código de Tamanho Variável - HSF.

2.3.2. Compressão por Segmentação

DESCRIÇÃO

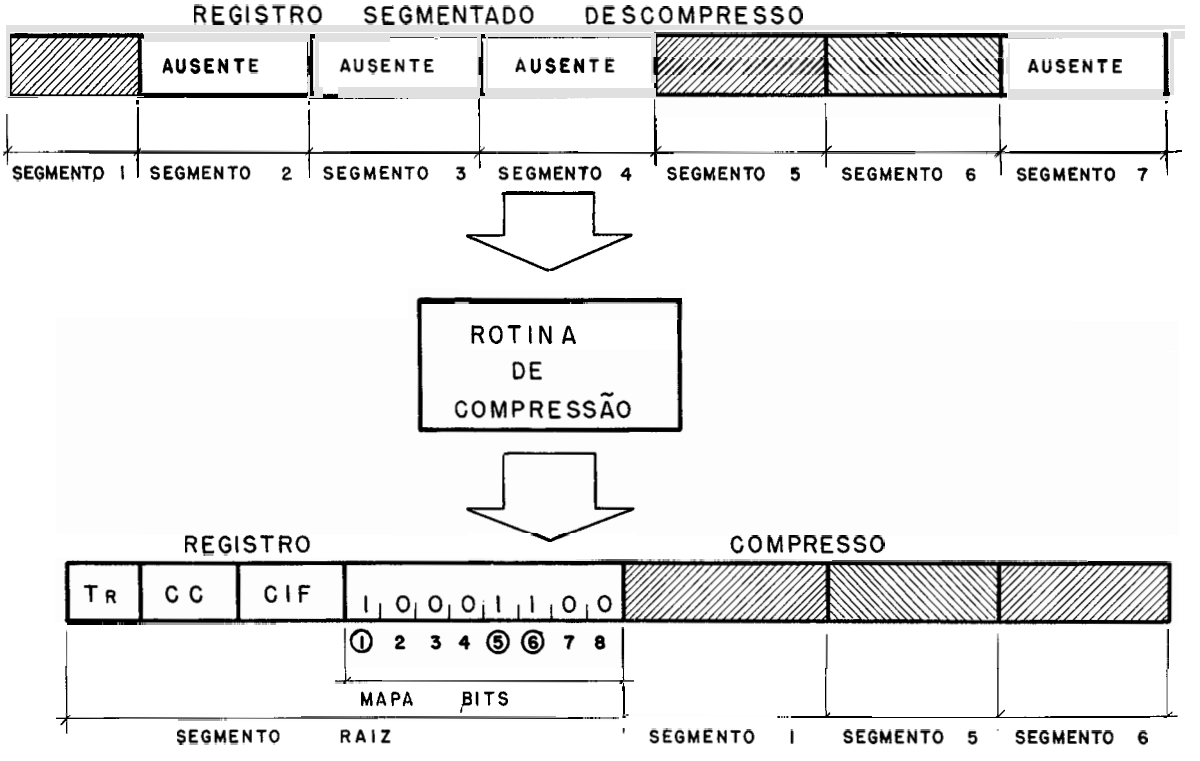
Por ser este um método de fácil conceituação e desenvolvimento, de apresentar, à exceção de alguns detalhes, um caráter geral quanto a sua aplicação, de oferecer uma boa compressão e, principalmente, de permitir combinação com outras técnicas, contribuiu, de forma definitiva, na sua escolha para compor o nosso sistema.

Este método, a partir do registro original segmentado, elimina os segmentos sem conteúdo registrando, na saída, os segmentos restantes, concatenados a um segmento raiz. Este segmento como já vimos, reune um conjunto de informações, que facilitam e otimizam a execução das rotinas de compressão e descompressão e permitem, sempre que desejado, a recuperação dos segmentos eliminados.

Este conjunto de informações do segmento raiz consiste, basicamente, de dados como Comprimento do Registro Compresso, Campos de Controle, Campos de Informações Fixas e Indicadores de Segmento. Neste último campo encontramos através de um bit, por exemplo, a indicação da presença ou ausência de conteúdo em um dado segmento.

A seguir, para uma melhor fixação de idéias, representamos este processo por um diagrama, onde mostramos um registro segmentado descompresso, a rotina de compressão e, fi

nalmente, o registro compresso com o segmento raiz e os indicadores, no caso, tipo bit.



III-3

No exemplo acima verifica-se que os bits 1, 5 e 6 estão ligados indicando a presença no registro, dos segmentos 1, 5 e 6, respectivamente.

Neste ponto iniciamos a apresentação de um outro processo, mencionado anteriormente, que trata da Supressão de Caracteres Repetidos.

Aqui, uma sequência de caracteres idênticos é substituída por um grupo de 2 bytes, onde o primeiro, denominado "Caráter Sentinela", indica a supressão de uma sequência, podendo, ainda, indicar, se desejado, o caráter componente da sequência ou a quantidade de seus elementos. O segundo byte indicará, dependendo do esquema escolhido, o número de elementos ou o caráter componente da sequência. Observamos que o segundo byte poderá ser eliminado caso o caráter componente da sequência esteja pré-estabelecido.

Lembrando que o nosso registro segmentado consiste de sequências de segmentos com e sem conteúdo, [illegible]e

fazendo-se uma analogia da sequência de segmentos com uma sequência de caracteres, da qual se conhece antecipadamente o caráter componente, veremos que ambas possuem as mesmas características.

Assim, por extrapolação, podemos concluir, ser possível aplicar, para representar uma sequência de segmentos sem conteúdo em um registro compresso pelo Método de Segmentação, a mesma filosofia empregada pelo Método de Compressão por Supressão de Caracteres Repetidos, em que o caráter componente da sequência é antecipadamente conhecido. Isto é, podemos substituir uma série de segmentos sem conteúdo por um único byte - (Byte Sentinela - BS), que indicará o número de segmentos eliminados naquele local do registro. Da mesma forma que o Caráter Sentinela, o Byte Sentinela deverá possuir uma marca de modo a identificá-lo entre os caracteres do texto compresso.

Do exposto e considerando os fatores:

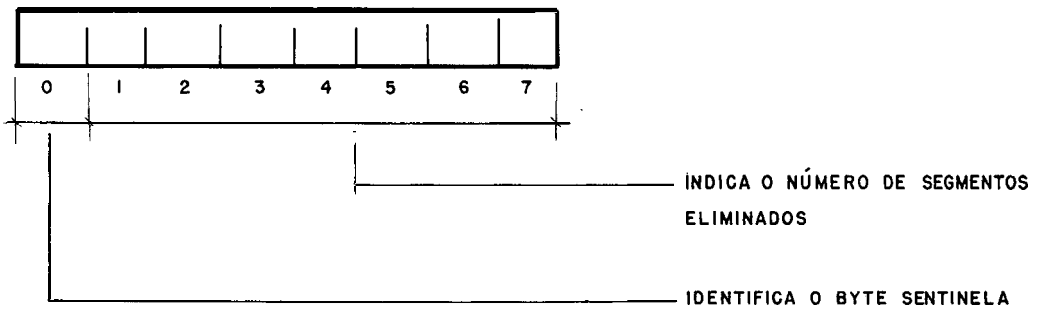
a) Número máximo de 64 segmentos para um registro, e

b) Comprimento máximo de 128 bytes para um segmento compresso,

projetamos o nosso Byte Sentinela, como segue:

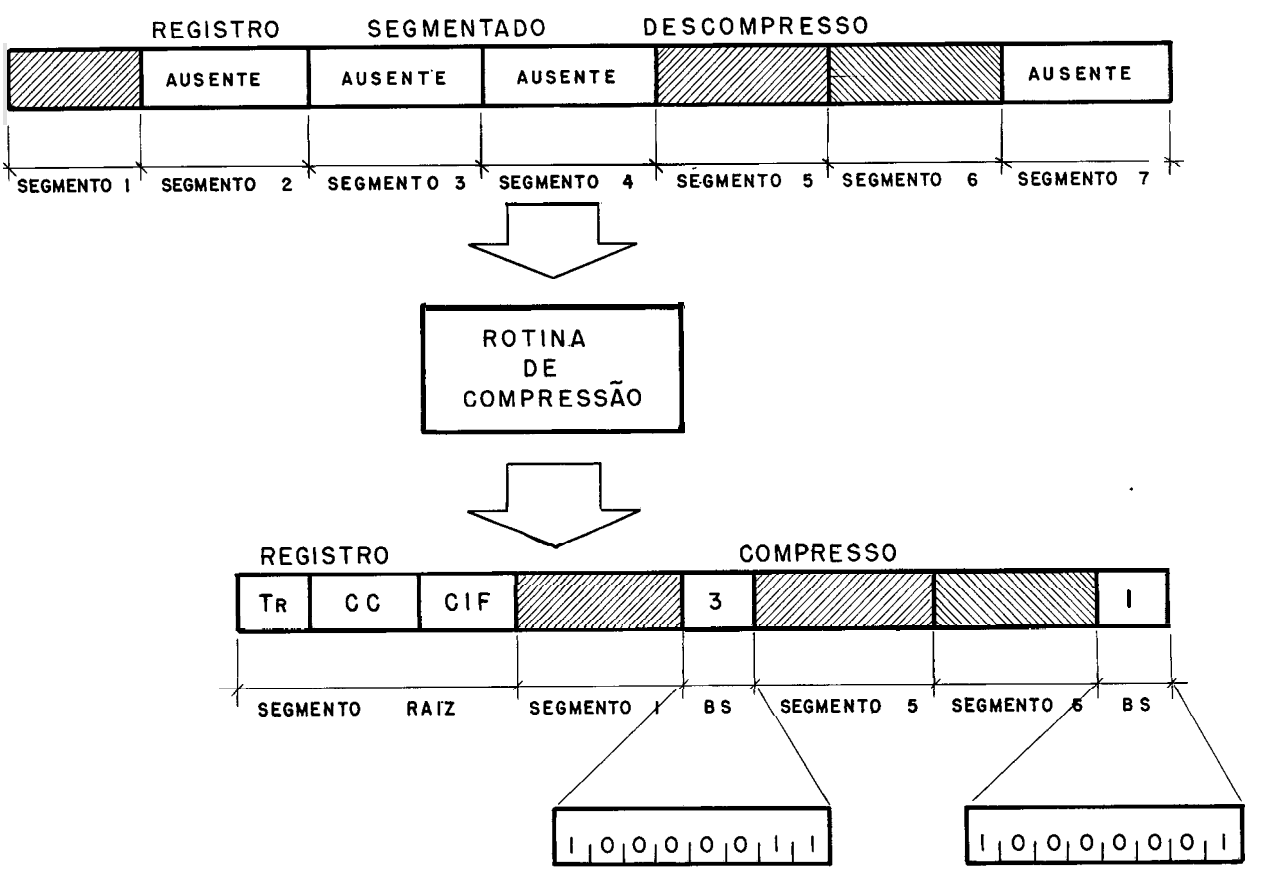
a) Bit 1 a 7: indica o número de segmentos eliminados para um máximo de 128 segmentos (considerados uma futura expansão do número de segmentos por registro).

b) Bit Ø, ligado - 1: com esta marca conseguimos identificar perfeitamente o Byte Sentinela uma vez que a unidade de informação para compressão é o segmento, e que os segmentos com conteúdo compressos, por serem de tamanho variável, estarão sempre precedidos por um byte indicador de tamanho, o qual terá o bit Ø desligado - Ø, visto que o comprimento máximo do segmento compresso é de 128 bytes.



III.-4

Tomando-se o exemplo anterior e representando-se os segmentos eliminados dentro desta nova sistemática, teríamos para o registro compresso o formato abaixo:

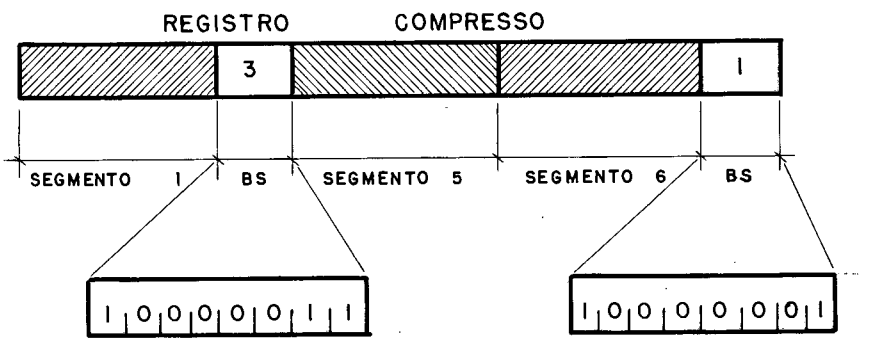


III-5

Uma vez alterada a forma de representar os segmentos ausentes no registro compresso e que o sistema proposto procederá a compressão e/ou descompressão total do registro, passam a ser dispensáveis para o sistema atual os Campos de Controle e Campos de Informação Fixa. Com isto, do segmento raiz, resta somente o campo corresponden-

te ao comprimento compresso do registro. Este campo também será eliminado uma vez que foi encontrada uma outra forma de determiná-lo, conforme veremos adiante.

Assim, com a eliminação do segmento raiz, o registro compresso passou a ser representado da seguinte forma:



III-6

Finalmente devemos observar que caso o número de segmentos ausentes, seja maior que a capacidade do Byte Sentinela, poderemos à semelhança da técnica de Supressão de Caracteres Repetidos, utilizar:

- uma sentinela com mais de 2 bytes, ou
- tantos Bytes Sentinela quantos sejam necessários para registrar todos os segmentos ausentes.

Para que o leitor sinta de forma prática o método apresentamos a seguir parte do nosso arquivo de testes original e compresso por segmentação, onde ressaltamos 2 registros para análise.

Inicialmente vamos observar nos registros compressos, que cada segmento se encontra precedido por 1 byte indicador de comprimento, visto que aos mesmos será aplicado, posteriormente, um outro método de compressão que os tornará de comprimento variável. Como se pode observar no exemplo apresentado a seguir, o primeiro segmento do registro 1 tem por conteúdo do seu primeiro byte, em hexadecimal, X' 08', indicando que o segmento a seguir possui o comprimento de 8 bytes.

Da mesma forma podemos observar também os Bytes

Sentinelas, os quais retratam o número de segmentos eliminados do registro original por estarem sem conteúdo. Assim para o registro 1 temos o Byte Sentinela X' 84', o qual indica que neste local foram eliminados, por não terem conteúdo, 4 segmentos conforme podemos constatar observando o registro original. Do mesmo modo, no registro 2, encontramos o Byte Sentinela X '82' indicando a eliminação de 2 segmentos.

Por fim, queremos mais uma vez chamar a atenção para o fato, de que todo segmento é iniciado por um byte indicador de comprimento cujo bit Ø é zero diferindo do Byte Sentinela que tem o bit Ø igual a 1. Assim, ao descomprimir os registros, podemos perfeitamente distinguir um segmento de um Byte Sentinela, porém devemos deixar claro que a formatação do Byte Sentinela não é fixa, ficando a critério do projetista determinar a configuração que melhor se adapte ao seu sistema.

```
51   DATA   80      000250      0000000000      1111155555CCCCCCCCCC12345
52   DATA   80      000250      0C00000000      1111155555CCCCCCCCCC12345
53   DATA   80      000250      000000C000      1111155555CCCCCCCCCC12345
54   DATA   80      000250      C000000000      1111155555CCCCCCCCCC12345
55   DATA   30      000250      0C00000000      1111155555CCCCCCCCCC12345
56   DATA   80      000250      0000000000      1111155555CCCCCCCCCC12345
57   DATA   80      000250      0000000000      1111155555CCCCCCCCCC12345
53   DATA   30      000250      0C00000000      1111155555CCCCCCCCCC12345
59   DATA   80      000250      0000000000      1111155555CCCCCCCCCC12345
50   DATA   80      000250      0C0C000000      1111155555CCCCCCCCCC12345

* * DEVICE  280  SYS018,  MODE IS  9 TRACK  1600 BPI      BLOCK    7   DATA   800

51   DATA   90      000300      1111100000      1111155555CCCCCCCCCC00000
62   DATA   80      000300      11111000C0      1111155555CCCCCCCCCC00000
63   DATA   80      000300      1111100000      1111155555CCCCCCCCCC00000
54   DATA   30      000300      1111100000      1111155555CCCCCCCCCC00000
65   DATA   80      000300      1111100000      1111155555CCCCCCCCCC00000
55   DATA   30      000300      1111100000      1111155555CCCCCCCCCC00000
57   DATA   80      000300      1111100000      1111155555CCCCCCCCCC00000
58   DATA   90      000300      1111100000      1111155555CCCCCCCCCC00000
59   DATA   80      000300      1111100000      1111155555CCCCCCCCCC00000
70   DATA   80      000300      1111100000      1111155555CCCCCCCCCC00000

            12345 25011111 55555 CCCCCCCCCC  000                    12345 25011111 55555 CCCCCCCCCC  000
4444444444FFFFF0FFFFFFFFF0FFFFFF0CCCCCCCCCC81FFF44444444444444444444FFFFF0FFFFFFFFF0FFFFFF0CCCCCCCCCC81FFF
00000C00012345825011111555555A33333333334800C00000C00000000000000123458250111115555554333333333348000
1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....
            (1)
                        12345 25011111 55555 CCCCCCCCCC  000                    12345 25011111 55555 CCCCC
444444444444444444FFFFF0FFFFFFFFF0FFFFFF0CCCCCCCCCC81FFF444444444444444444FFFFF0FFFFFFFFF0FFFFFF0CCCCC
00000000000000000012345825011111555555A3333333333480C0000000000000C00C000012345825011111555555A33333
101...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

CCCCC  000                    12345 30011111 55555 CCCCCCCCCC  11111  30011111 55555 CCCCCCCCCC  1111
CCCCC81FFF44444444444444444444FFFFF0FFFFFFFFF0FFFFFF0CCCCCCCCCC80FFFFF80FFFFFFFFF0FFFFFF0CCCCCCCCCC80FFFF
333334800000000000000000000000123458300111115555555A33333333332511111283001111155555A3333333333251111
201...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

1  30011111 55555 CCCCCCCCCC  11111  30011111 55555 CCCCCCCCCC  11111  30011111 55555 CCCCCCCCCC  11
F80FFFFFFFFF0FFFFFF0CCCCCCCCCC80FFFFF80FFFFFFFFF0FFFFFF0CCCCCCCCCC80FFFFF80FFFFFFFFF0FFFFFF0CCCCCCCCCC80FF
128300111115555555A33333333332511111283001111155555A33333333332511111283001111155555A3333333333332511
301....5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....
(2)                                               III - 7
```

SISTEMÁTICA DE IMPLANTAÇÃO

Tomaremos aqui o mesmo exemplo desenvolvido por o casião da apresentação do método tradicional (página 39), pois assim damos ao leitor a oportunidade de comparação entre os esquemas, permitindo uma avaliação mais apurada das qualidades do atual processo.

No exemplo escolhido já estão definidos o registro segmentado e a Tabela de Descrição do Registros, (quadros que se seguem).

Observe-se que no registro foi criado mais 1 segmento para a matrícula, pelo fato de, anteriormente, no processo tradicional, ela constar do segmento raiz.

Isto posto, passaremos a apresentar os procedimentos relativos às rotinas de compressão e descompressão.

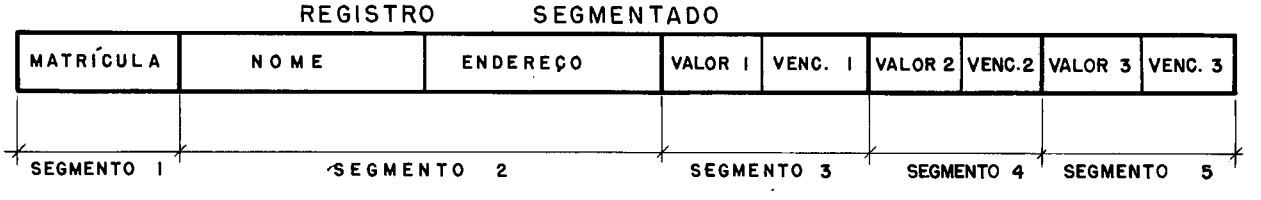


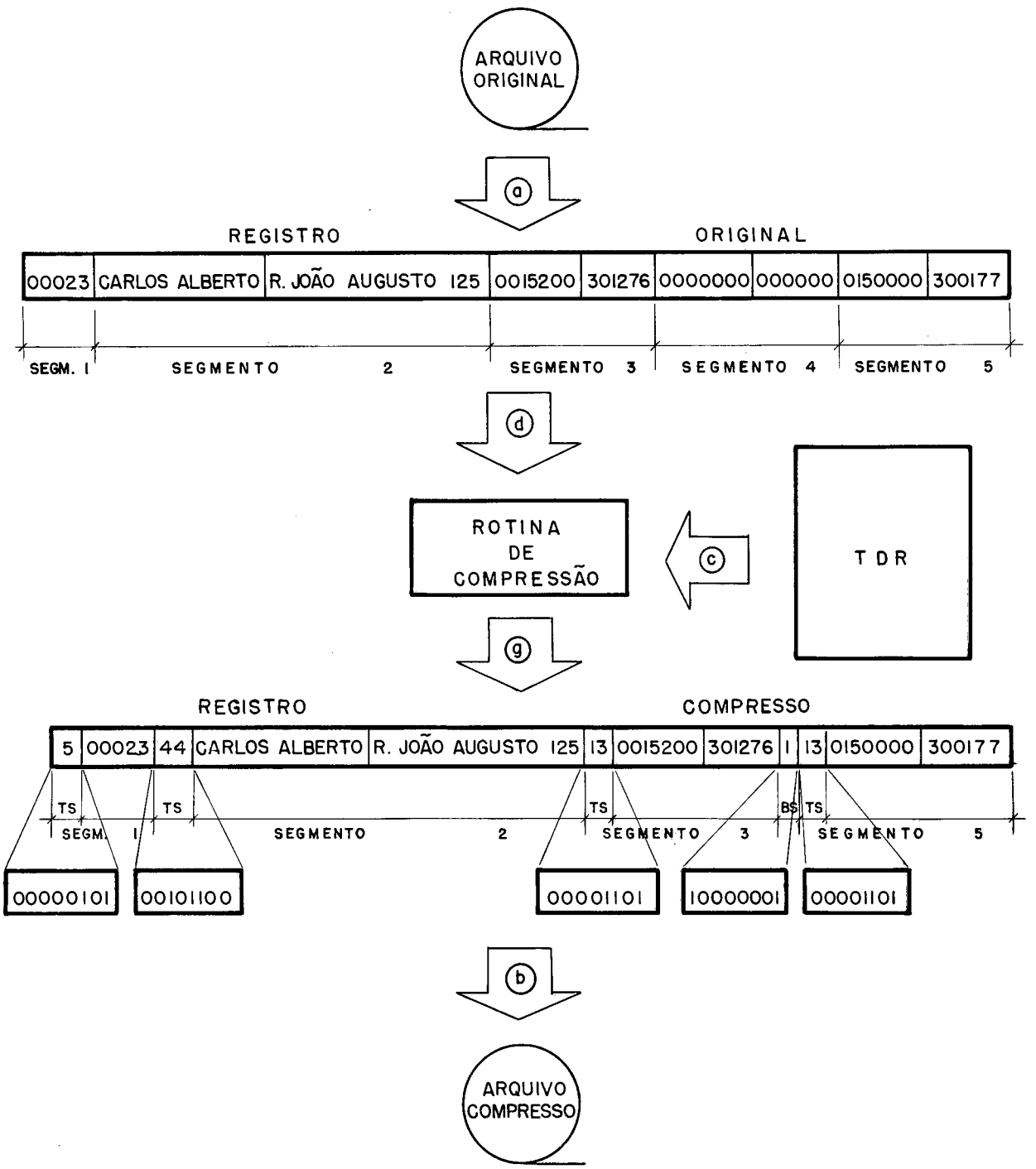
TABELA DE DESCRIÇÃO DO REGISTRO (TDR)

Nº SEGMENTO	FORMATO	POSIÇÃO RELATIVA	COMPRIMENTO
1	2	1	5
2	1	6	44
3	2	50	13
4	2	63	13
5	2	76	13

1 - INDICA FORMATO ALFANUMÉRICO

2 - INDICA FORMATO DECIMAL ZONADO

Rotina de Compressão:



Procedimentos de Compressão:

a) Rotina de Compressão recebe o registro original do arquivo (registro descompresso).

b) Se completou processamento do registro:

- Contador de segmentos sem conteúdo diferente de zero; formatar Byte Sentinela (BS) e armazená-lo no registro de saída, zerando, em seguida, o contador.
- Armazenar registro compresso na saída.
- Voltar ao item "a".

c) Da Tabela de Descrição do Registro (TDR), obter formato e comprimento do próximo segmento.

d) Retirar a partir do comprimento o segmento seguinte do registro descompresso.

e) Verificar, de acordo com o formato, se o segmento possui conteúdo.

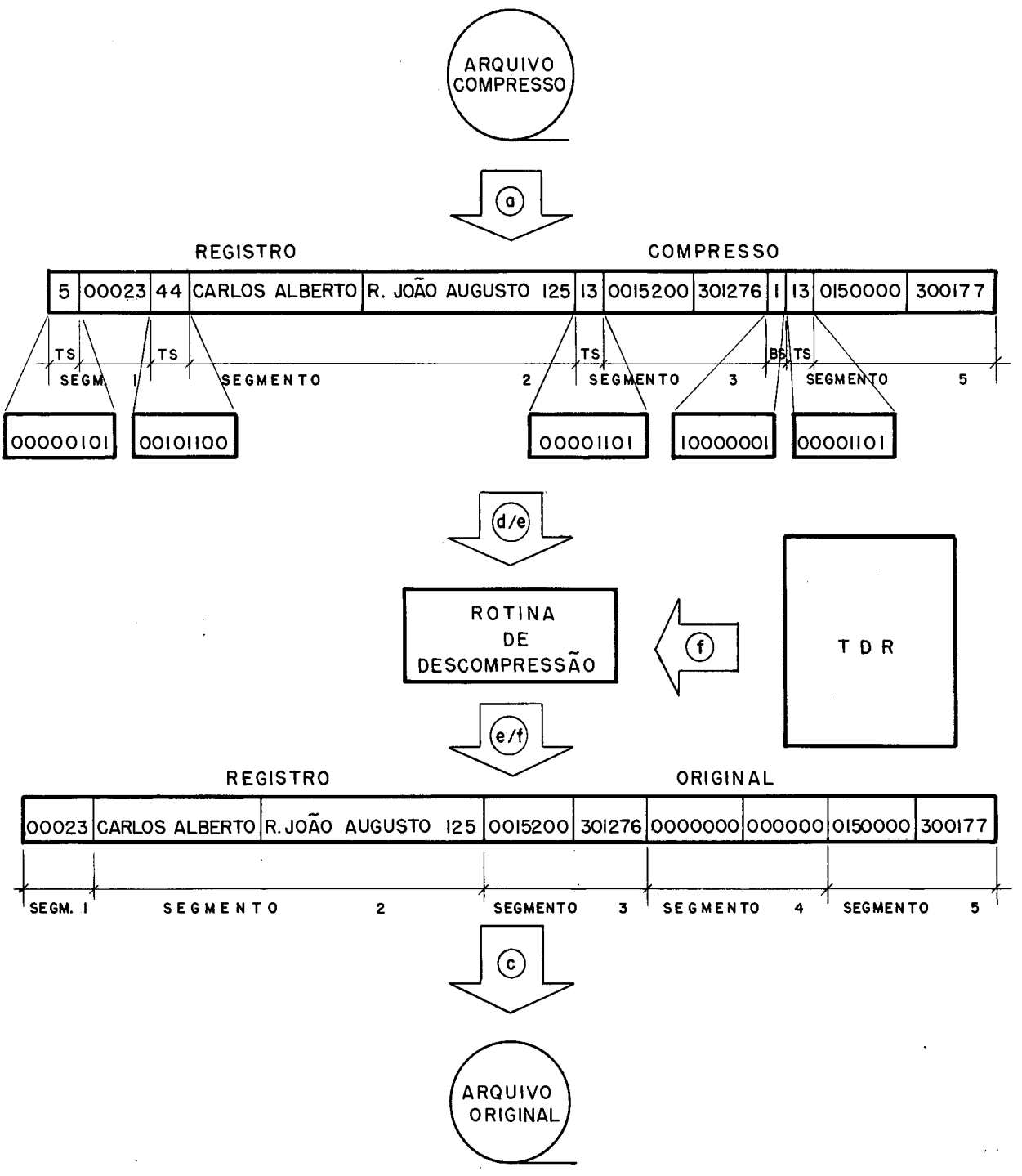
f) Se o segmento não tem conteúdo:

- Incrementar contador de segmentos sem conteúdo.
- Voltar ao item "b".

g) Se o segmento tem conteúdo:

- Contador de segmentos sem conteúdo diferente de zero; formatar Byte Sentinela (BS) e armazená-lo no registro compresso, zerando em seguida o contador.
- Formatar Byte de Tamanho do Segmento (TS), armazenando-o juntamente com o segmento no registro compresso.
- Voltar ao item "b".

Rotina de Descompressão:



III-10

Procedimentos de Descompressão:

a) Rotina de descompressão carrega bloco na área de entrada.

b) Se completou processamento do bloco, voltar ao item "a".

c) Se registro descompresso esta completo, armazená-lo na área de saída.

d) Obter próximo byte da área de entrada e verificar se é um Byte Sentinela ou de Comprimento do segmento através do bit Ø.

e) Se bit Ø igual a Ø, estamos com o Byte de Comprimento do segmento e então, devemos:

   - Obter do byte o Tamanho do Segmento (TS).
   - Transferir o segmento para o registro descompresso.
   - Voltar ao item "b".

f) Se bit Ø igual a 1, estamos com o Byte Sentinela e os procedimentos serão:

   - Obter da Tabela de Descrição de Registros (TDR) o formato e o comprimento do segmento.
   - De acordo com o formato e o comprimento do segmento expandir zeros ou brancos no registro descompresso.
   - Se o número de segmentos gerados confere com o valor indicado no Byte Sentinela (BS) voltar ao item "b", caso contrário posicionar Tabela de Descrição de Registros para o próximo segmento e repetir os 3 procedimentos deste item.

VANTAGENS/DESVANTAGENS

Com a esquematização apresentada o método proposto mantem as mesmas vantagens do método original, além de:

a) Aumentar a taxa de compressão, resultante do arquivo pela eliminação do segmento raiz e inclusão do Byte Sentinela.

b) Tornar o método ainda mais fácil na sua conceituação e implantação, uma vez que a eliminação da raiz com os Campos de Controle, Campos de Informação Fixa e Mapa de Indicadores traz como consequência a eliminação das rotinas e to-

dos os procedimentos necessários ao manuseio destas informações.

c) Não prefixar o número máximo de segmentos do registro segmentado.

d) Oferecer uma maior velocidade de compressão e descompressão se comparada ao esquema original, pois as rotinas para o esquema atual são muito mais simples.

Resta-nos observar ainda, que este método em ambos os esquemas, depende da estrutura dos dados, ou seja, da estrutura dos campos no registro, o que, em determinados casos, pode comprometer bastante a sua eficiência. Para sanar este problema, o sistema através do Módulo de Preparação, oferece o Programa Gerador de Estatísticas para Segmentação (COMP030), por meio do qual um conjunto de informações é fornecido ao usuário, possibilitando que os campos do registro sejam reestruturados em segmentos de forma a obter a melhor compressão possível.

2.3.3. Compressão Utilizando o Código de Tamanho Variável Huffman - Shannon - Fano (HSF)

DESCRIÇÃO

Uma vez que as reduções obtidas com o Método de Compressão por Segmentação não nos satisfizeram, partimos a procura de um outro processo que acoplado ao primeiro nos oferecesse uma compressão condizente com as nossas necessidades.

Assim, estudando os vários métodos existentes, decidimos pela utilização dos Códigos de Tamanho Variável, por serem eles, aqueles que nos oferecem maiores vantagens como, acoplamento com outras técnicas (Segmentação), altos índices de compressão, etc. Entre os vários códigos existentes, escolhemos o HSF por ser aquele que nos oferece:

- Mensagens de tamanho médio mínimo.
- As menores tabelas de tradução.
- Maior eficiência nos processos de codificação/decodificação.

- Atributo de sequência numérica, que permite o desenvolvimento de alternativas para os esquemas de codificação/decodificação que melhor se adaptem às nossas pretensões.

Desta forma, consideramos solucionado o problema da compressão efetiva dos dados, uma vez que o acoplamento da técnica de Compressão por Segmentação ao Código HSF nos fornecerão, dependendo da densidade de informação no arquivo, reduções em torno de 60%, o que achamos satisfatório para as nossas necessidades.

No capítulo anterior, tivemos oportunidade de expor detalhadamente o código HSF, quando apresentamos a forma convencional de determinação do código e dos procedimentos de compressão/descompressão.

Contudo a metodologia sugerida para a codificação/decodificação de informações não nos satisfez, uma vez que a mesma não oferecia um certo equilíbrio entre a memória utilizada e a velocidade de execução.

Por esta razão desenvolvemos uma nova alternativa para os procedimentos de compressão/descompressão utilizando o código HSF, onde procuramos balancear a memória utilizada com a velocidade de execução tendo em vista atender as necessidades básicas do sistema.

A nova alternativa proposta está fundamentada basicamente nos fatos abaixo:

a) Fixação de Alfabetos

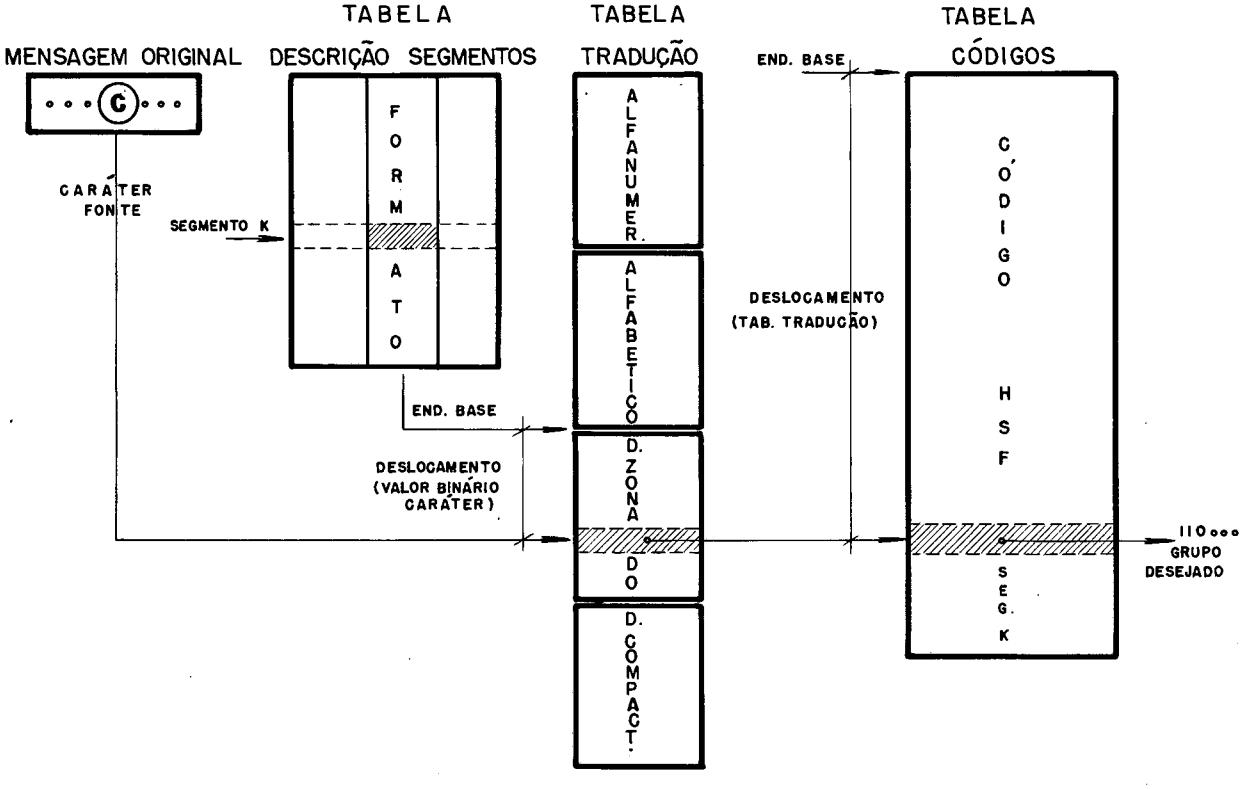
Como se sabe, ao trabalharmos com Códigos de Tamanho Variável, atribuímos novas configurações de bits para representar cada um dos caracteres encontrados na amostra. Ao conjunto dos diferentes caracteres encontrados chamamos de Alfabeto.

Há várias maneiras de se obter um Alfabeto. Por ser a forma mais simples e segura e aquela que nos permitirá obter um melhor balanceamento entre a velocidade de codificação e memória real utilizada, nos decidimos pela fixação de um Alfabeto para cada formato e neste, de um

"caráter substituto", cuja função é substituir todo e qualquer carater não definido no Alfabeto.

Como decorrência da fixação dos alfabetos reduzimos substancialmente a memória requerida, para armazenamento de todos os grupos pertencentes aos diversos códigos HSF gerados. Também conseguimos melhorar sensivelmente a performance global da rotina de compressão, já que foi possível fixar para cada formato a localização relativa do grupo correspondente a cada carater na Tabela de Códigos do segmento.

Fixados estes pontos, desenvolvemos uma rotina de compressão em que, a partir do formato do segmento e do caráter fonte é indexada uma Tabela de Tradução, a qual define os alfabetos fixados para cada formato, obtendo-se daí o deslocamento para a Tabela de Códigos que nos dará para o carater em questão o grupo do código HSF gerado.



III-11

Escolhemos, para o nosso sistema, os alfabetos e caracteres substitutos (sublinhados), a seguir:

ALFABETOS

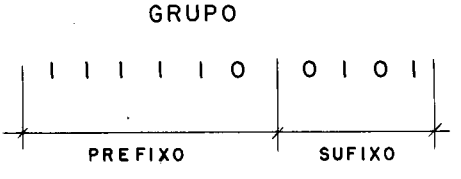
ALFANUMÉRICO		ALFABÉTICO		DECIMAL ZONADO		DECIMAL COMPACTADO	
HEXADECIMAL	CARÁTER	HEXADECIMAL	CARÁTER	HEXADECIMAL	CARÁTER	HEXADECIMAL	CARÁTER
40	∅	40	∅	C0	0	F0	0
4A	¢	C1	A	C1	1	F1	1
4B	.	C2	B	C2	2	F2	2
4D	(	C3	C	C3	3	F3	3
4E	+	C4	D	C4	4	F4	4
4F	|	C5	E	C5	5	F5	5
50	&	C6	F	C6	6	F6	6
5B	$	C7	G	C7	7	F7	7
5C	*	C8	H	C8	8	F8	8
5D	)	C9	I	C9	9	F9	9
5E	;	D1	J	D0	0	FC	+
60	-	D2	K	D1	-1	FD	-
61	/	D3	L	D2	-2	FF	
		D4	M	D3	-3		
6A	¦	D5	N	D4	-4		
6B	,	D6	O	D5	-5		
6C	%	D7	P	D6	-6		
6D	_	D8	Q	D7	-7		
7A	:	D9	R	D8	-8		
7B	#	E2	S	D9	-9		
7C	@	E3	T	F0	0		
7D	'	E4	U	F1	1		
7E	=	E5	V	F2	2		
7F	"	E6	W	F3	3		
C1	A	E7	X	F4	4		
C2	B	E8	Y	F5	5		
C3	C	E9	Z	F6	6		
C4	D			F7	7		
C5	E			F8	8		
C6	F			F9	9		
C7	G						
C8	H						
C9	I						
D1	J						
D2	K						
D3	L						
D4	M						
D5	N						
D6	O						
D7	P						
D8	Q						
D9	R						
E2	S						
E3	T						
E4	U						
E5	V						
E6	W						
E7	X						
E8	Y						
E9	Z						
F0	0						
F1	1						
F2	2						
F3	3						
F4	4						
F5	5						
F6	6						
F7	7						
F8	8						
F9	9						

Para efeito de determinação do código HSF, através do Programa Utilitário Gerador de Códigos - COMPØ4Ø, consideramos uma frequência de ocorrência inicial de 1 para cada elemento do alfabeto.

b) Atributo de Sequência Numérica do Código HSF

A utilização do Atributo de Sequência Numérica principal característica do código HSF, permite, para a rotina de descompressão, um melhor balanceamento entre a velocidade de decodificação e a memória real utilizada.

Para tanto, consideramos o grupo de um código HSF como sendo formado de 2 partes, a primeira das quais, denominada de prefixo, constitui-se, da esquerda para a direita do grupo, dos bits 1 até o primeiro bit Ø inclusive, enquanto a segunda, denominada de sufixo, ficam agrupados os bits restantes.



III-13

Tendo em mente nosso problema principal, observemos, como exemplo, a tabela abaixo, onde apresentamos para um dado alfabeto, o código HSF gerado, os prefixos e sufixos.

ALFABETO	CÓDIGO HSF	PREFIXO	SUFIXO
2	0 0	0	0
9	0 1 0		1 0
4	0 1 1		1 1
Ø	1 0 0	1 0	0
A	1 0 1 0		1 0
8	1 0 1 1		1 1
3	1 1 0 0	1 1 0	0
7	1 1 0 1		1
1	1 1 1 0 0	1 1 1 0	0
5	1 1 1 0 1		1
B	1 1 1 1 0	1 1 1 1 0	-
(ÚLTIMO)	1 1 1 1 1	-	-

III-14

Após estudarmos detalhadamente o código HSF e as implicações de seu emprego conseguimos extrair 3 pontos que consideramos básicos para o novo processo de decodificação a ser desenvolvido e que podem ser observados no código registrado na Tabela mostrada como exemplo, a saber:

- Pelo Atributo de Sequência Numérica, os grupos do código HSF gerado se apresentam numa sequência ascendente de números inteiros, o que também é verdade para os prefixos e para os sufixos correspondentes a cada prefixo isoladamente.
- Para haver uma mudança de prefixo é necessário que o sufixo do grupo anterior seja composto apenas de bits 1.
- O "ÚLTIMO" grupo de qualquer código HSF (111...1) não segue a lei de formação por nós proposta (prefixo e sufixo), e por isso será oportunamente eliminado. De forma que será válido afirmar que qualquer grupo de um código HSF deve, no mínimo conter prefixo.

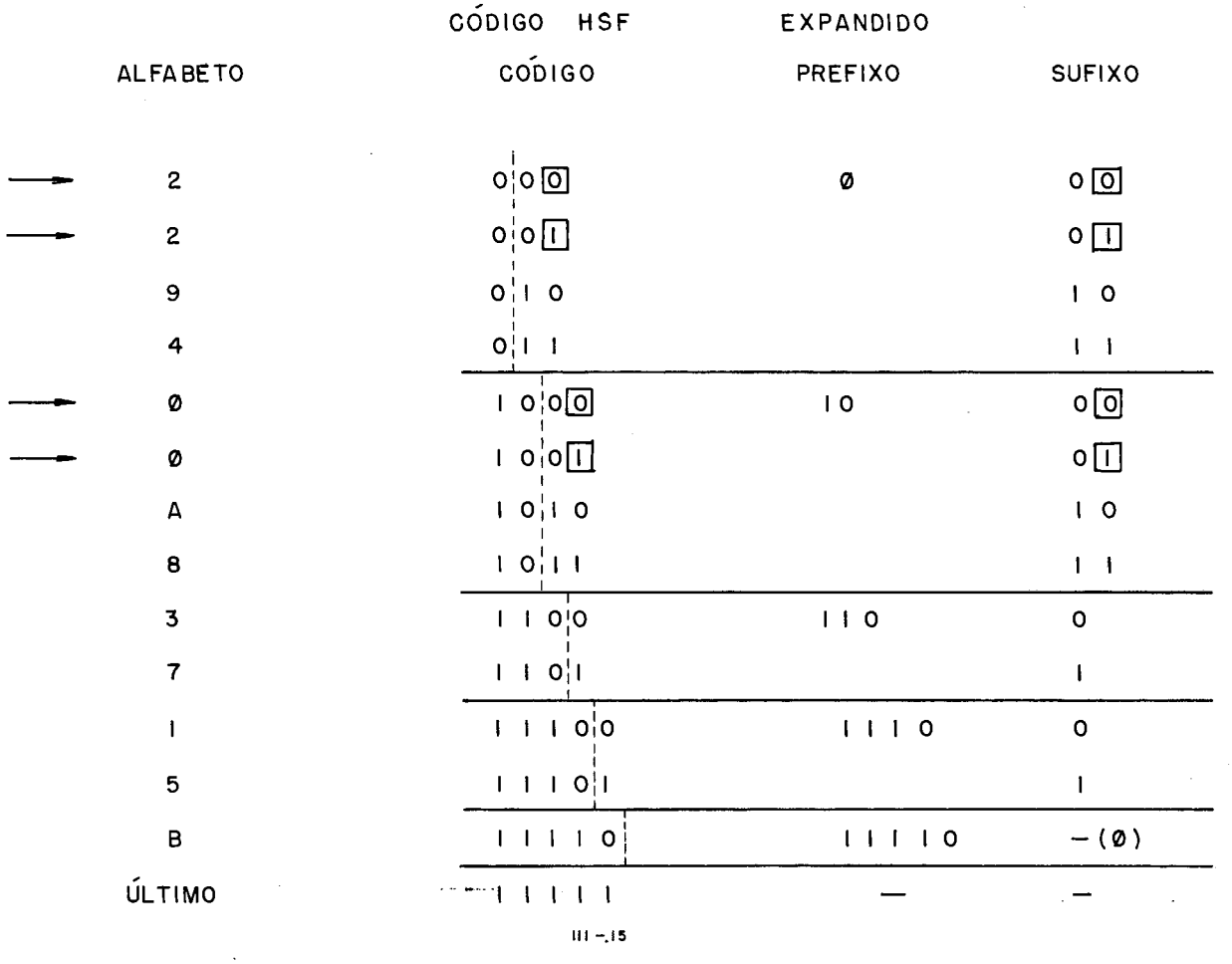
De acordo com o que foi exposto, a metodologia desenvolvida apresenta as seguintes características:

b.1) Expansão do Alfabeto - Código

Aqui, pela utilização do Atributo de Sequência Numérica e da propriedade para mudança de prefixo, desenvolvemos, para a descompressão, um novo alfabeto e um novo código HSF que denominamos de código HSF Expandido.

Esta expansão procura simplesmente completar para um dado prefixo a sequência numérica do sufixo, atribuindo para estes casos o mesmo elemento do alfabeto.

Assim ao expandirmos o código HSF da tabela anterior, teremos:



Podemos verificar, observando-se a tabela acima, que ao se completar a sequência numérica do sufixo para os prefixos "Ø" e "1Ø", tivemos de repetir os elementos "2" e "Ø".

Complementando, apresentamos as seguintes informações:

1. Para os códigos sem sufixo, não haverá expansão, considerando-se para o mesmo o valor "Ø".

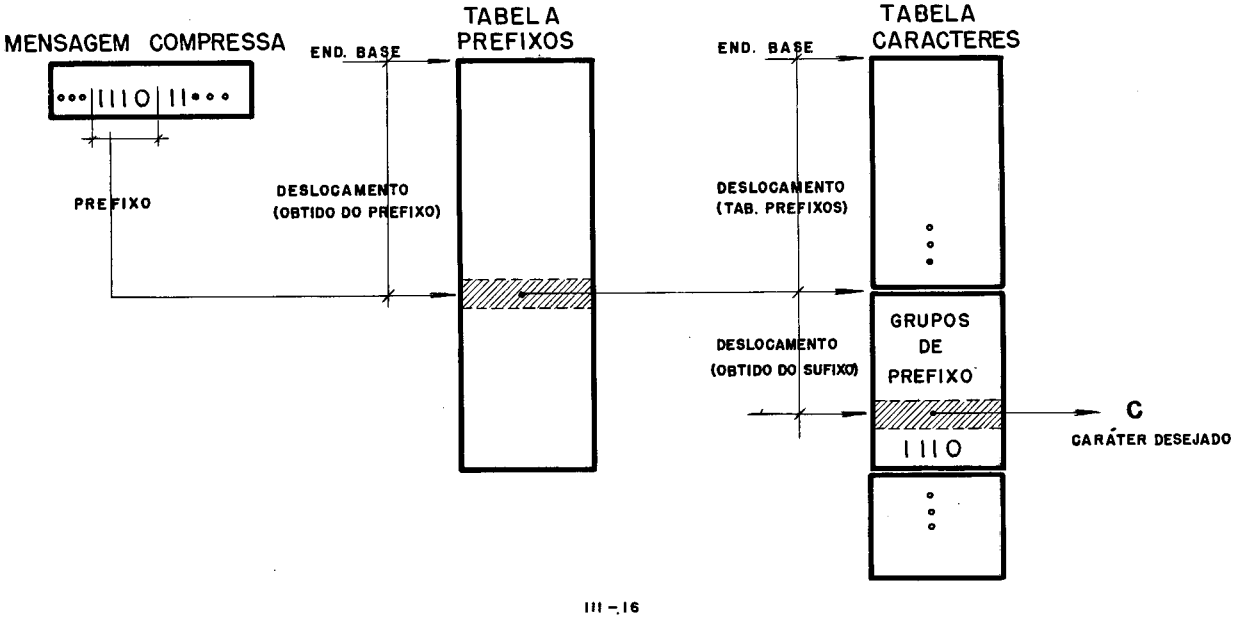
2. Para o "ÚLTIMO" grupo do código, uma vez que ele não satisfaz a lei de formação (prefixo-sufixo), criamos um "Caráter Fantasma", o qual não se encontra nos dados e cuja finalidade é compor o alfabeto para construção do código HSF não participando das tabelas de compressão e descompressão.

b.2) Acesso pelo Prefixo

Aqui utilizamos o prefixo do grupo na mensagem compressa para determinar a localização,em uma dada tabela, dos caracteres correspondentes aos grupos expandidos com o mesmo prefixo.

Podemos concluir que o novo processo pela utilização das características acima terá uma melhor performance do que o processo convencional apesar do pequeno acréscimo de memória real para armazenamento das tabelas.

Desta forma, foi simples, como poderemos ver posteriormente, desenvolver uma rotina de descompressão onde a partir do prefixo do grupo que esta sendo decodificado, indexa-se uma Tabela de Prefixo obtendo-se o endereço relativo do conjunto de caracteres correspondente aos grupos com este prefixo na Tabela de Caracteres para posteriormente através do sufixo selecionar deste conjunto o caráter desejado.



III-16

Vistos os fundamentos do processo proposto, passamos a apresentação das tabelas requeridas para o seu funcionamento que são:

a) Tabela de Tradução (TRAD1)

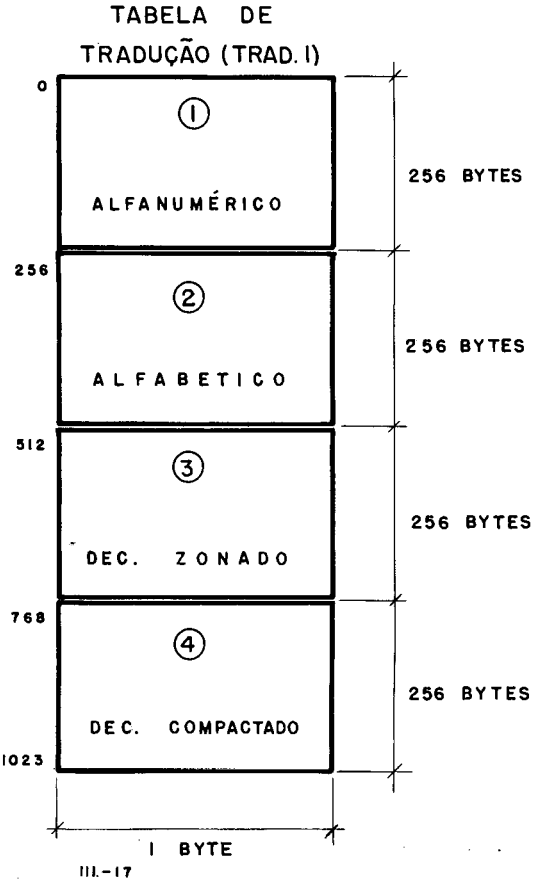
Tabela fixa, através da qual a rotina de compres

são é informada a respeito de cada um dos alfabetos definidos para cada um dos formatos (alfanumérico, alfabético, decimal zonado, decimal compactado) e do deslocamento fixo na Tabela de Códigos de onde se encontra o grupo atribuído a um dado caráter fonte.

Para uma melhor compreensão por parte do leitor, apresentaremos a seguir as principais características desta tabela:

a.1) Estrutura

A tabela está estruturada segundo um vetor de tamanho igual a 1024 bytes dividida em 4 subtabelas de 256 bytes cada uma. Observamos que cada subtabela contem a definição do alfabeto de um dado formato.



Observamos que cada entrada nesta tabela possui o tamanho de 1 byte.

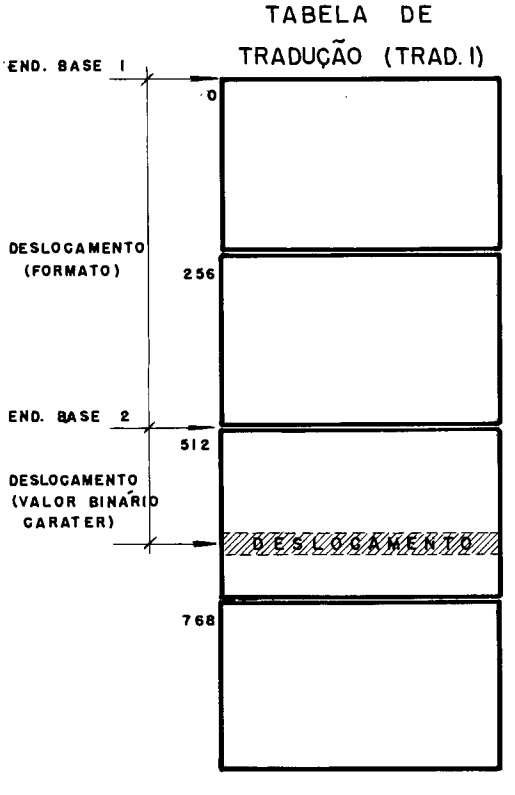
a.2) Acesso

1. A cada novo segmento a ser processado tomamos o seu formato da Tabela de Descrição dos Segmentos (TDS) e por seu intermédio determinamos o endereço base da subtabela desejada.

FORMATO	ENDEREÇO BASE
ALFANUMÉRICO	0
ALFABÉTICO	256
DEC. ZONADO	512
DEC. COMPACTADO	768

III-18

2. A cada caráter fonte com o valor correspondente à sua configuração binária, temos o deslocamento que juntamente com o endereço base, determinado anteriormente, nos fornecerá o endereço efetivo de entrada na Tabela de Tradução.



III-19

a.3) Conteúdo

Desta tabela se obtem o deslocamento fixo indicativo do local onde, na Tabela de Códigos, se encontra o grupo correspondente ao carater fonte em questão.

É oportuno lembrar, que este deslocamento deverá ser combinado ao endereço base correspondente aos grupos do código HSF gerado para o segmento em análise. Esse procedimento será visto ao se descrever a Tabela de Códigos.

Concluindo, apresentamos abaixo a Tabela de Tradução, onde os deslocamentos estão expressos em hexadecimal.

DESLOCAMENTO	FORMATO			
	ALFANUMER. END. BASE = Ø	ALFABETICO END. BASE = 256	D. ZONADO END. BASE = 512	D.COMPACTADO END. BASE = 1024
0	Ø Ø	Ø Ø	5 Ø	Ø Ø
1	Ø Ø	Ø Ø	5 Ø	Ø Ø
2	Ø Ø	Ø Ø	5 Ø	Ø Ø
3	Ø Ø	Ø Ø	5 Ø	Ø Ø
4	Ø Ø	Ø Ø	5 Ø	Ø Ø
5	Ø Ø	Ø Ø	5 Ø	Ø Ø
6	Ø Ø	Ø Ø	5 Ø	Ø Ø
7	Ø Ø	Ø Ø	5 Ø	Ø Ø
8	Ø Ø	Ø Ø	5 Ø	Ø Ø
9	Ø Ø	Ø Ø	5 Ø	Ø Ø
10	Ø Ø	Ø Ø	5 Ø	Ø Ø
11	Ø Ø	Ø Ø	5 Ø	Ø Ø
12	Ø Ø	Ø Ø	5 Ø	Ø Ø
13	Ø Ø	Ø Ø	5 Ø	Ø Ø
14	Ø Ø	Ø Ø	5 Ø	Ø Ø
15	Ø Ø	Ø Ø	5 Ø	Ø Ø
16	Ø Ø	Ø Ø	5 Ø	Ø Ø
17	Ø Ø	Ø Ø	5 Ø	Ø Ø
18	Ø Ø	Ø Ø	5 Ø	Ø Ø
19	Ø Ø	Ø Ø	5 Ø	Ø Ø
20	Ø Ø	Ø Ø	5 Ø	Ø Ø
21	Ø Ø	Ø Ø	5 Ø	Ø Ø
22	Ø Ø	Ø Ø	5 Ø	Ø Ø
23	Ø Ø	Ø Ø	5 Ø	Ø Ø
24	Ø Ø	Ø Ø	5 Ø	Ø Ø
25	Ø Ø	Ø Ø	5 Ø	Ø Ø
26	Ø Ø	Ø Ø	5 Ø	Ø Ø
27	Ø Ø	Ø Ø	5 Ø	Ø Ø
28	Ø Ø	Ø Ø	5 Ø	Ø Ø
29	Ø Ø	Ø Ø	5 Ø	Ø Ø
30	Ø Ø	Ø Ø	5 Ø	Ø Ø
31	Ø Ø	Ø Ø	5 Ø	Ø Ø
32	Ø Ø	Ø Ø	5 Ø	Ø Ø
33	Ø Ø	Ø Ø	5 Ø	Ø Ø
34	Ø Ø	Ø Ø	5 Ø	Ø Ø
35	Ø Ø	Ø Ø	5 Ø	Ø Ø
36	Ø Ø	Ø Ø	5 Ø	Ø Ø
37	Ø Ø	Ø Ø	5 Ø	Ø Ø
38	Ø Ø	Ø Ø	5 Ø	Ø Ø
39	Ø Ø	Ø Ø	5 Ø	Ø Ø
40	Ø Ø	Ø Ø	5 Ø	Ø Ø
41	Ø Ø	Ø Ø	5 Ø	Ø Ø
42	Ø Ø	Ø Ø	5 Ø	Ø Ø
43	Ø Ø	Ø Ø	5 Ø	Ø Ø
44	Ø Ø	Ø Ø	5 Ø	Ø Ø
45	Ø Ø	Ø Ø	5 Ø	Ø Ø
46	Ø Ø	Ø Ø	5 Ø	Ø Ø
47	Ø Ø	Ø Ø	5 Ø	Ø Ø
48	Ø Ø	Ø Ø	5 Ø	Ø Ø
49	Ø Ø	Ø Ø	5 Ø	Ø Ø
50	Ø Ø	Ø Ø	5 Ø	Ø Ø
51	Ø Ø	Ø Ø	5 Ø	Ø Ø
52	Ø Ø	Ø Ø	5 Ø	Ø Ø
53	Ø Ø	Ø Ø	5 Ø	Ø Ø
54	Ø Ø	Ø Ø	5 Ø	Ø Ø
55	Ø Ø	Ø Ø	5 Ø	Ø Ø
56	Ø Ø	Ø Ø	5 Ø	Ø Ø
57	Ø Ø	Ø Ø	5 Ø	Ø Ø
58	Ø Ø	Ø Ø	5 Ø	Ø Ø
59	Ø Ø	Ø Ø	5 Ø	Ø Ø
60	Ø Ø	Ø Ø	5 Ø	Ø Ø
61	Ø Ø	Ø Ø	5 Ø	Ø Ø
62	Ø Ø	Ø Ø	5 Ø	Ø Ø
63	Ø Ø	Ø Ø	5 Ø	Ø Ø

DESLOCAMENTO	FORMATO			
	ALFANUMER.	ALFABETICO	D. ZONADO	D.COMPACTADO
64	Ø Ø	Ø Ø	5 Ø	Ø Ø
65	Ø Ø	Ø Ø	5 Ø	Ø Ø
66	Ø Ø	Ø Ø	5 Ø	Ø Ø
67	Ø Ø	Ø Ø	5 Ø	Ø Ø
68	Ø Ø	Ø Ø	5 Ø	Ø Ø
69	Ø Ø	Ø Ø	5 Ø	Ø Ø
70	Ø Ø	Ø Ø	5 Ø	Ø Ø
71	Ø Ø	Ø Ø	5 Ø	Ø Ø
72	Ø Ø	Ø Ø	5 Ø	Ø Ø
73	Ø Ø	Ø Ø	5 Ø	Ø Ø
74	Ø 4	Ø Ø	5 Ø	Ø Ø
75	Ø 8	Ø Ø	5 Ø	Ø Ø
76	Ø Ø	Ø Ø	5 Ø	Ø Ø
77	Ø C	Ø Ø	5 Ø	Ø Ø
78	1 Ø	Ø Ø	5 Ø	Ø Ø
79	1 4	Ø Ø	5 Ø	Ø Ø
80	1 8	Ø Ø	5 Ø	Ø Ø
81	Ø Ø	Ø Ø	5 Ø	Ø Ø
82	Ø Ø	Ø Ø	5 Ø	Ø Ø
83	Ø Ø	Ø Ø	5 Ø	Ø Ø
84	Ø Ø	Ø Ø	5 Ø	Ø Ø
85	Ø Ø	Ø Ø	5 Ø	Ø Ø
86	Ø Ø	Ø Ø	5 Ø	Ø Ø
87	Ø Ø	Ø Ø	5 Ø	Ø Ø
88	Ø Ø	Ø Ø	5 Ø	Ø Ø
89	Ø Ø	Ø Ø	5 Ø	Ø Ø
90	Ø Ø	Ø Ø	5 Ø	Ø Ø
91	1 C	Ø Ø	5 Ø	Ø Ø
92	2 Ø	Ø Ø	5 Ø	Ø Ø
93	2 4	Ø Ø	5 Ø	Ø Ø
94	2 8	Ø Ø	5 Ø	Ø Ø
95	Ø Ø	Ø Ø	5 Ø	Ø Ø
96	2 C	Ø Ø	5 Ø	Ø Ø
97	3 Ø	Ø Ø	5 Ø	Ø Ø
98	Ø Ø	Ø Ø	5 Ø	Ø Ø
99	Ø Ø	Ø Ø	5 Ø	Ø Ø
100	Ø Ø	Ø Ø	5 Ø	Ø Ø
101	Ø Ø	Ø Ø	5 Ø	Ø Ø
102	Ø Ø	Ø Ø	5 Ø	Ø Ø
103	Ø Ø	Ø Ø	5 Ø	Ø Ø
104	Ø Ø	Ø Ø	5 Ø	Ø Ø
105	Ø Ø	Ø Ø	5 Ø	Ø Ø
106	3 4	Ø Ø	5 Ø	Ø Ø
107	3 8	Ø Ø	5 Ø	Ø Ø
108	3 C	Ø Ø	5 Ø	Ø Ø
109	4 Ø	Ø Ø	5 Ø	Ø Ø
110	Ø Ø	Ø Ø	5 Ø	Ø Ø
111	Ø Ø	Ø Ø	5 Ø	Ø Ø
112	Ø Ø	Ø Ø	5 Ø	Ø Ø
113	Ø Ø	Ø Ø	5 Ø	Ø Ø
114	Ø Ø	Ø Ø	5 Ø	Ø Ø
115	Ø Ø	Ø Ø	5 Ø	Ø Ø
116	Ø Ø	Ø Ø	5 Ø	Ø Ø
117	Ø Ø	Ø Ø	5 Ø	Ø Ø
118	Ø Ø	Ø Ø	5 Ø	Ø Ø
119	Ø Ø	Ø Ø	5 Ø	Ø Ø
120	Ø Ø	Ø Ø	5 Ø	Ø Ø
121	Ø Ø	Ø Ø	5 Ø	Ø Ø
122	4 4	Ø Ø	5 Ø	Ø Ø
123	4 8	Ø Ø	5 Ø	Ø Ø
124	4 C	Ø Ø	5 Ø	Ø Ø
125	5 Ø	Ø Ø	5 Ø	Ø Ø
126	5 4	Ø Ø	5 Ø	Ø Ø
127	5 8	Ø Ø	5 Ø	Ø Ø

DESLOCAMENTO	FORMATO			
	ALFANUMER.	ALFABETICO	D. ZONADO	D.COMPACTADO
128	00	00	50	00
129	00	00	50	00
130	00	00	50	00
131	00	00	50	00
132	00	00	50	00
133	00	00	50	00
134	00	00	50	00
135	00	00	50	00
136	00	00	50	00
137	00	00	50	00
138	00	00	50	00
139	00	00	50	00
140	00	00	50	00
141	00	00	50	00
142	00	00	50	00
143	00	00	50	00
144	00	00	50	00
145	00	00	50	00
146	00	00	50	00
147	00	00	50	00
148	00	00	50	00
149	00	00	50	00
150	00	00	50	00
151	00	00	50	00
152	00	00	50	00
153	00	00	50	00
154	00	00	50	00
155	00	00	50	00
156	00	00	50	00
157	00	00	50	00
158	00	00	50	00
159	00	00	50	00
160	00	00	50	00
161	00	00	50	00
162	00	00	50	00
163	00	00	50	00
164	00	00	50	00
165	00	00	50	00
166	00	00	50	00
167	00	00	50	00
168	00	00	50	00
169	00	00	50	00
170	00	00	50	00
171	00	00	50	00
172	00	00	50	00
173	00	00	50	00
174	00	00	50	00
175	00	00	50	00
176	00	00	50	00
177	00	00	50	00
178	00	00	50	00
179	00	00	50	00
180	00	00	50	00
181	00	00	50	00
182	00	00	50	00
183	00	00	50	00
184	00	00	50	00
185	00	00	50	00
186	00	00	50	00
187	00	00	50	00
188	00	00	50	00
189	00	00	50	00
190	00	00	50	00
191	00	00	50	00

DESLOCAMENTO	FORMATO			
	ALFANUMER.	ALFABETICO	D. ZONADO	D.COMPACTADO
192	00	00	00	00
193	5C	04	04	00
194	60	08	08	00
195	64	0C	0C	00
196	68	10	10	00
197	6C	14	14	00
198	70	18	18	00
199	74	1C	1C	00
200	78	20	20	00
201	7C	24	24	00
202	00	00	50	00
203	00	00	50	00
204	00	00	50	00
205	00	00	50	00
206	00	00	50	00
207	00	00	50	00
208	00	00	28	00
209	80	28	2C	00
210	84	2C	30	00
211	88	30	34	00
212	8C	34	38	00
213	90	38	3C	00
214	94	3C	40	00
215	98	40	44	00
216	9C	44	48	00
217	A0	48	4C	00
218	00	00	50	00
219	00	00	50	00
220	00	00	50	00
221	00	00	50	00
222	00	00	50	00
223	00	00	50	00
224	00	00	50	00
225	00	00	50	00
226	A4	4C	50	00
227	A8	50	50	00
228	AC	54	50	00
229	B0	58	50	00
230	B4	5C	50	00
231	B8	60	50	00
232	BC	64	50	00
233	C0	68	50	00
234	00	00	50	00
235	00	00	50	00
236	00	00	50	00
237	00	00	50	00
238	00	00	50	00
239	00	00	50	00
240	C4	00	50	00
241	C8	00	54	04
242	CC	00	58	08
243	D0	00	5C	0C
244	D4	00	60	10
245	D8	00	64	14
246	DC	00	68	18
247	E0	00	6C	1C
248	E4	00	70	20
249	E8	00	74	24
250	EC	00	50	00
251	00	00	50	00
252	00	00	50	28
253	00	00	50	2C
254	00	00	50	00
255	00	00	50	30

b) Tabela de Códigos (CODS)

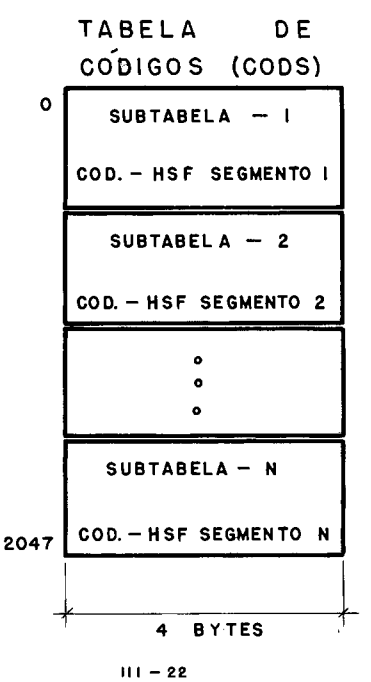
Esta é uma tabela gerada pelo Programa Utilitário Gerador de Códigos (COMPØ4Ø) e utilizada pela rotina de compressão.

Fornece, a cada caráter fonte obtido da mensagem original por intermédio da Tabela de Tradução, o grupo do código HSF que lhe é correspondente.

Para uma melhor compreensão, mostramos em seguida, as principais características desta tabela:

b.1) Estrutura

Esta tabela esta estruturada segundo um vetor de tamanho igual a 2048 bytes dividida em subtabelas, uma para cada segmento projetado para o registro, visto que para cada segmento será gerado um código HSF.

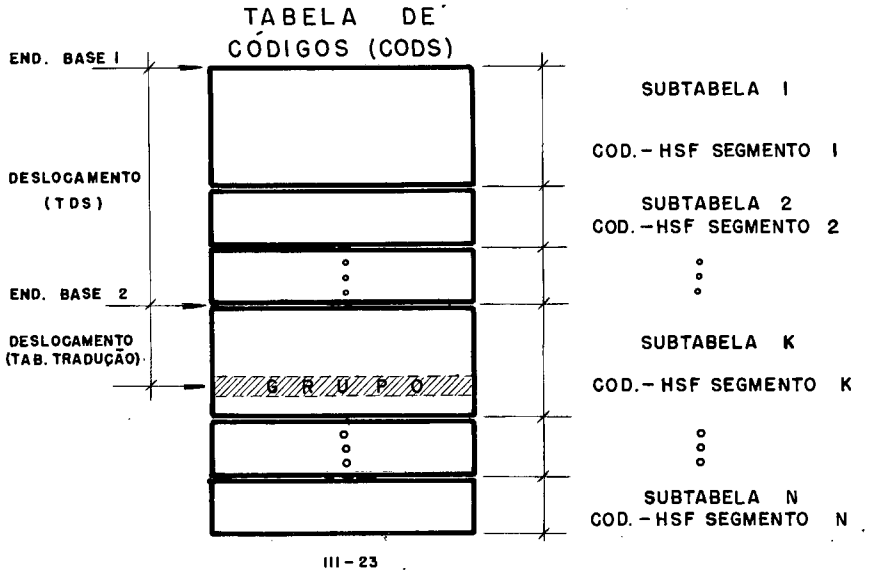


Observamos que cada entrada nesta tabela possui tamanho de 4 bytes.

b.2) Acesso

1. A cada novo segmento a ser processado, obtemos através da Tabela de Descrição dos Segmentos (TDS) o deslocamento que nos permitirá obter o endereço base correspondente à subtabela que contem os grupos do código HSF gerados para o segmento em questão.

2. A cada caráter fonte de posse do valor obtido da Tabela de Tradução, temos o deslocamento, que combinado ao endereço base da subtabela já determinado nos fornecerá o endereço efetivo, na Tabela de Códigos, do grupo desejado.



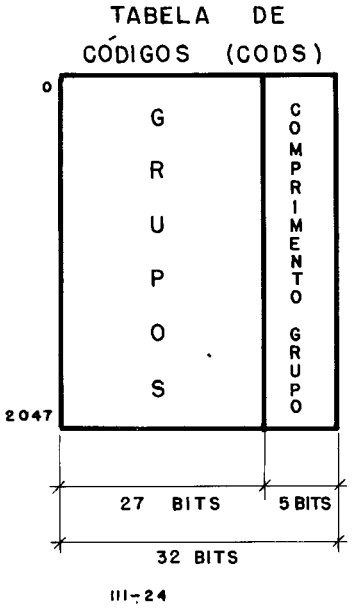
b.3) Conteúdo

A tabela está formada por todos os grupos pertencentes aos diversos códigos HSF gerados (1 código HSF para cada segmento) concatenados aos seus respectivos comprimentos que se encontram nos últimos 5 bits de cada entrada na tabela.

Os grupos componentes do código HSF relativos a cada segmento, para cada formato, possuem as suas localizações fixas na Tabela de Códigos.

Apresentamos a seguir, para cada formato, os alfabetos (Hexadecimal) com os respectivos endereços relativos fixos de cada um de seus elementos pa-

ra as Tabelas de Códigos e Tradução.



III-24

SEQUENCIA	END. RELATIVO	ALFABETOS			
		ALFANUMERICO	ALFABETICO	D. ZONADO	D. COMPACTADO
1	0 0	4 0	4 0	C 0	F 0
2	0 4	4 A	C 1	C 1	F 1
3	0 8	4 B	C 2	C 2	F 2
4	0 C	4 D	C 3	C 3	F 3
5	1 0	4 E	C 4	C 4	F 4
6	1 4	4 F	C 5	C 5	F 5
7	1 8	5 0	C 6	C 6	F 6
8	1 C	5 B	C 7	C 7	F 7
9	2 0	5 C	C 8	C 8	F 8
10	2 4	5 D	C 9	C 9	F 9
11	2 8	5 E	D 1	D 0	F C
12	2 C	6 0	D 2	D 1	F D
13	3 0	6 1	D 3	D 2	F F
14	3 4	6 A	D 4	D 3	ÚLTIMO
15	3 8	6 B	D 5	D 4	
16	3 C	6 C	D 6	D 5	
17	4 0	6 D	D 7	D 6	
18	4 4	7 A	D 8	D 7	
19	4 8	7 B	D 9	D 8	
20	4 C	7 C	E 2	D 9	
21	5 0	7 D	E 3	F 0	
22	5 4	7 E	E 4	F 1	
23	5 8	7 F	E 5	F 2	
24	5 C	C 1	E 6	F 3	
25	6 0	C 2	E 7	F 4	
26	6 4	C 3	E 8	F 5	
27	6 8	C 4	E 9	F 6	
28	6 C	C 5	ÚLTIMO	F 7	
29	7 0	C 6		F 8	
30	7 4	C 7		F 9	
31	7 8	C 8		ÚLTIMO	
32	7 C	C 9			
33	8 0	D 1			
34	8 4	D 2			
35	8 8	D 3			
36	8 C	D 4			
37	9 0	D 5			
38	9 4	D 6			
39	9 8	D 7			
40	9 C	D 8			
41	A 0	D 9			
42	A 4	E 2			
43	A 8	E 3			
44	A C	E 4			
45	B 0	E 5			
46	B 4	E 6			
47	B 8	E 7			
48	B C	E 8			
49	C 0	E 9			
50	C 4	F 0			
51	C 8	F 1			
52	C C	F 2			
53	D 0	F 3			
54	D 4	F 4			
55	D 8	F 5			
56	D C	F 6			
57	E 0	F 7			
58	E 4	F 8			
59	E 8	F 9			
60	E C	ÚLTIMO			

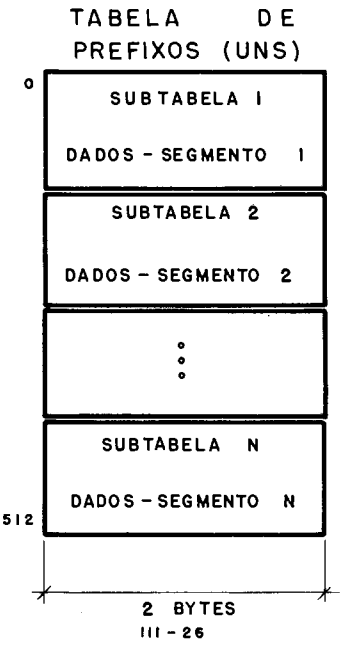
c) Tabela de Prefixos (UNS)

Esta Tabela é gerada pelo Programa Utilitário Gerador de Códigos (COMPØ4Ø) e utilizada pela rotina de descompressão. Ela fornece, para cada prefixo do código HSF gerado, todos os dados necessários à determinação do caráter correspondente ao grupo em decodificação.

Como principais características desta tabela, temos:

c.1) Estrutura

A tabela está estruturada segundo um array bidimensional de tamanho igual a 512 bytes, dividida em subtabelas, uma para cada segmento projetado para o registro, visto existir para cada segmento, dependendo do seu formato, um código HSF.

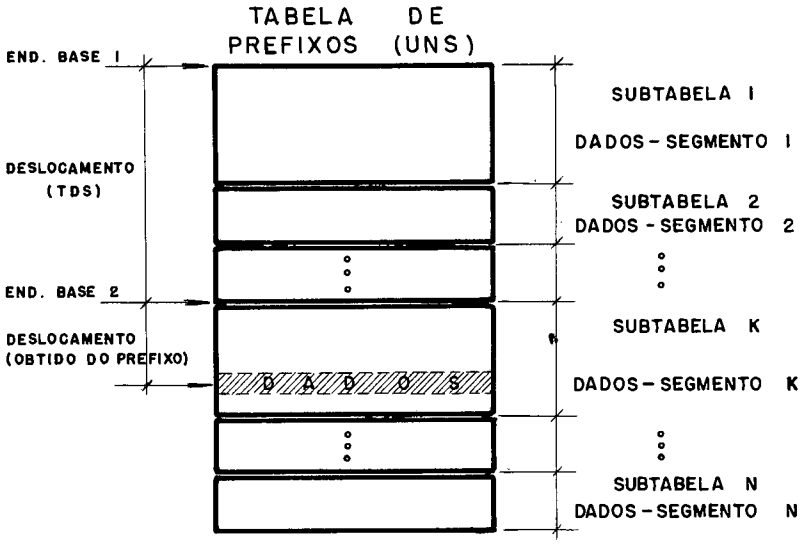


Observamos que cada entrada nesta tabela, possui o tamanho de 2 bytes.

c.2) Acesso

1. A cada novo segmento a ser processado, através da Tabela de Descrição dos Segmentos (TDS), obtemos o deslocamento que nos permitirá obter o endereço base da subtabela correspondente ao segmento em questão.

2. A cada prefixo obtido da mensagem compressa, considerando que para cada prefixo do código temos uma entrada na tabela de tamanho igual a 2 bytes, determinamos o deslocamento (número de bits do prefixo * 2), que combinado ao endereço base da subtabela, determinado anteriormente, nos dará o endereço efetivo dos dados referentes aos grupos que possuem o prefixo em questão.



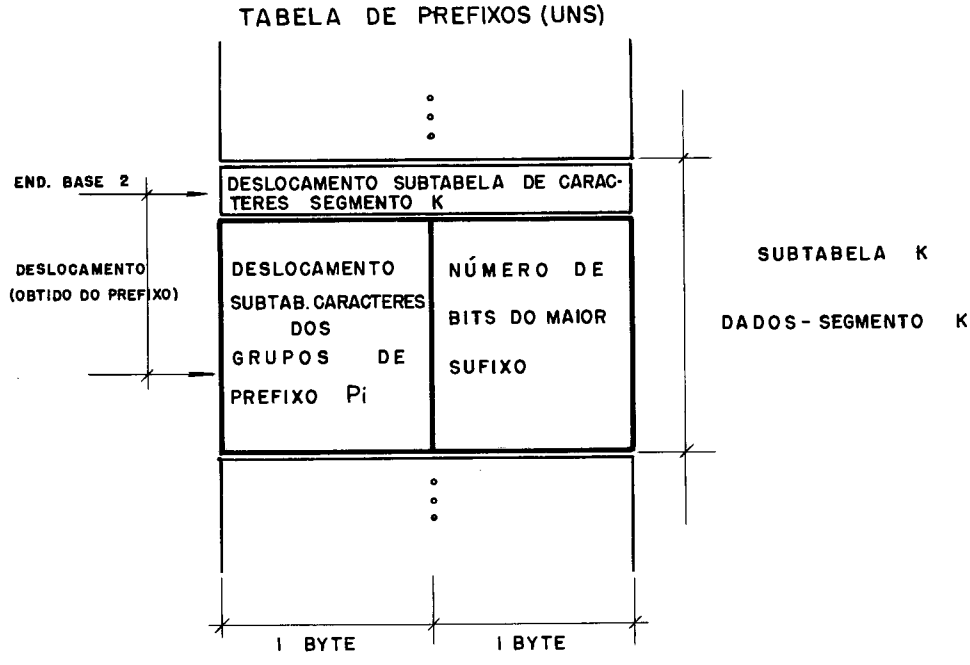
III-27

c.3) Conteúdo

1. Ocupando a primeira entrada de cada subtabela temos o Endereço Base da Subtabela de Caracteres relativa ao segmento em análise (próxima tabela a ser apresentada).

2. Para cada prefixo do código HSF de cada segmento, o deslocamento que indicará nas Subtabelas de Caracteres do segmento o local onde se encontra o conjunto de dados relativos aos grupos do código HSF expandido com o mesmo prefixo do grupo em decodificação.

3. Número de bits do maior sufixo pertencente aos grupos com o prefixo do grupo em decodificação.



III - 28

Para uma melhor compreensão, apresentamos abaixo, a Tabela de Prefixos referente ao nosso exemplo (página):

TABELA DE PREFIXOS (UNS)

DESLOCAMENTO	COMPRIMENTO	
SUBTAB - CARACTERES	MAIOR	SUFIXO
Ø		2
8		2
16		1
18		1
2Ø		Ø

III.- 29

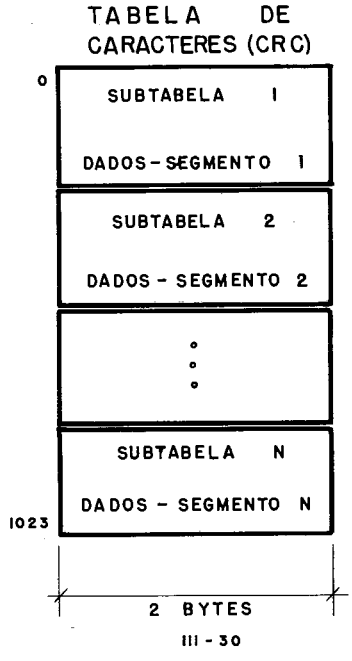
d) Tabela de Caracteres (CRC)

Esta Tabela é gerada pelo Programa Utilitário Gerador de Códigos (COMPØ4Ø) e utilizada pela rotina de descompressão. Sua finalidade é fornecer, a cada grupo obtido da mensagem compressa por intermédio da Tabela de Prefixos, o caráter fonte correspondente ao mesmo.

Como principais características desta tabela, temos:

d.1) Estrutura

Esta tabela esta estruturada segundo um array bidimensional de tamanho igual a 1024 bytes, dividida em subtabelas, uma para cada segmento projetado para o registro, visto existir para cada segmento um alfabeto.



Observamos que cada entrada nesta tabela possui o tamanho de 2 bytes.

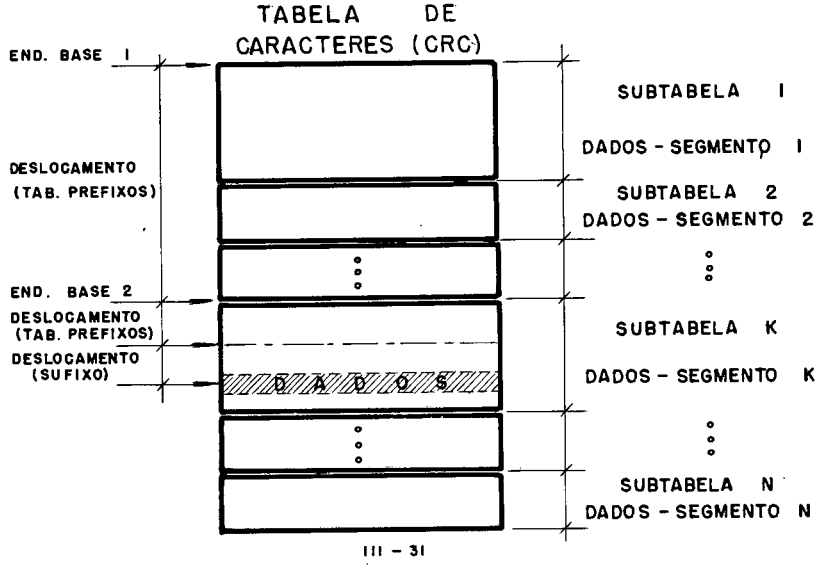
d.2) Acesso

1. A cada novo segmento a ser processado através da Tabela de Prefixos, obtemos o deslocamento que nos permitirá obter o endereço base da subtabela correspondente ao segmento em questão.

2. A cada prefixo obtido da mensagem compressa, através da Tabela de Prefixos, obtemos o endereço relativo, o qual combinado ao endereço base da subtabela, determinado anteriormente, nos fornecerá o endereço efetivo do conjunto de dados referentes aos grupos do código HSF expandido com o mesmo prefixo do grupo em decodificação.

3. A cada prefixo obtido da mensagem compressa, através da Tabela de Prefixos, obtemos o comprimento do maior sufixo ("N") para os grupos com este prefixo e a partir deste dado retiramos da mensagem compressa os próximos "N" bits.

Uma vez, que na Tabela de Caracteres, trabalhamos com o código HSF expandido, isto é, para cada prefixo os dados foram expandidos de forma a atender todas as configurações binárias do maior sufixo, e que cada entrada na tabela possui o comprimento de 2 bytes, bastará multiplicarmos por 2 o valor correspondente aos "N" bits obtidos anteriormente para termos o deslocamento que combinado ao endereço base obtido nos permitirá acessar os dados relativos ao grupo em decodificação.

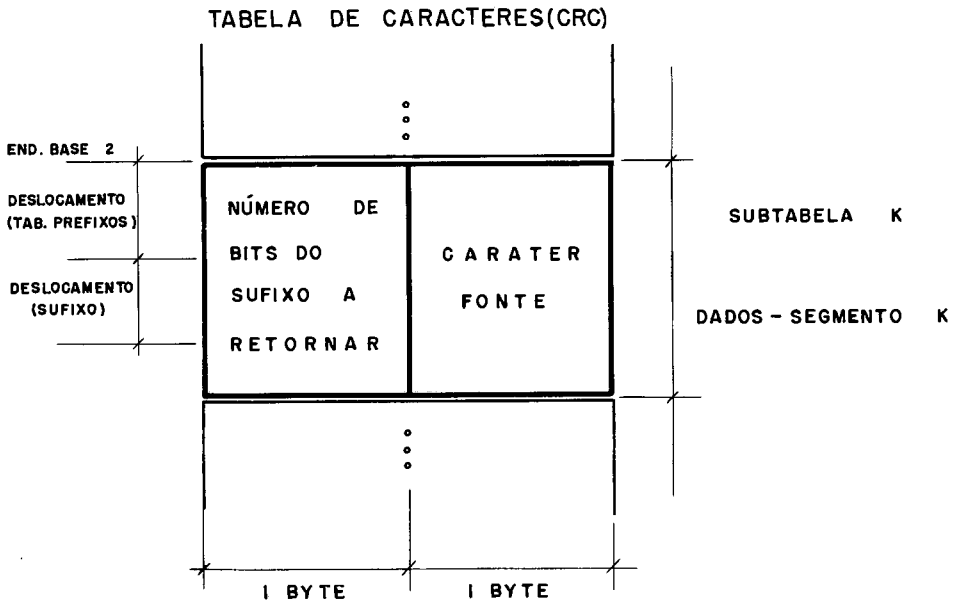


d.3) Conteúdo

Para cada grupo do código HSF expandido, temos uma entrada na Tabela de Caracteres, que nos fornece:

1. O caráter fonte correspondente ao grupo expandido obtido da mensagem compressa.

2. O número de bits do grupo expandido que deve retornar à mensagem compressa, já que o grupo expandido, por definição pode incorporar bits que não lhe pertencem.



III - 32

Para uma melhor compreensão, apresentamos abaixo, a Tabela de Caracteres referente ao nosso exemplo.

TABELA DE CARACTERES

BITS RETORNO	CARÁCTER FONTE	
1	2	
1	2	
0	9	PREFIXO Ø
0	4	
1	Ø	
1	Ø	
0	A	PREFIXO 10
0	8	
0	3	
0	7	PREFIXO 110
0	1	
0	5	PREFIXO 1110
0	B	PREFIXO 11110

III - 33

SISTEMÁTICA DE IMPLANTAÇÃO

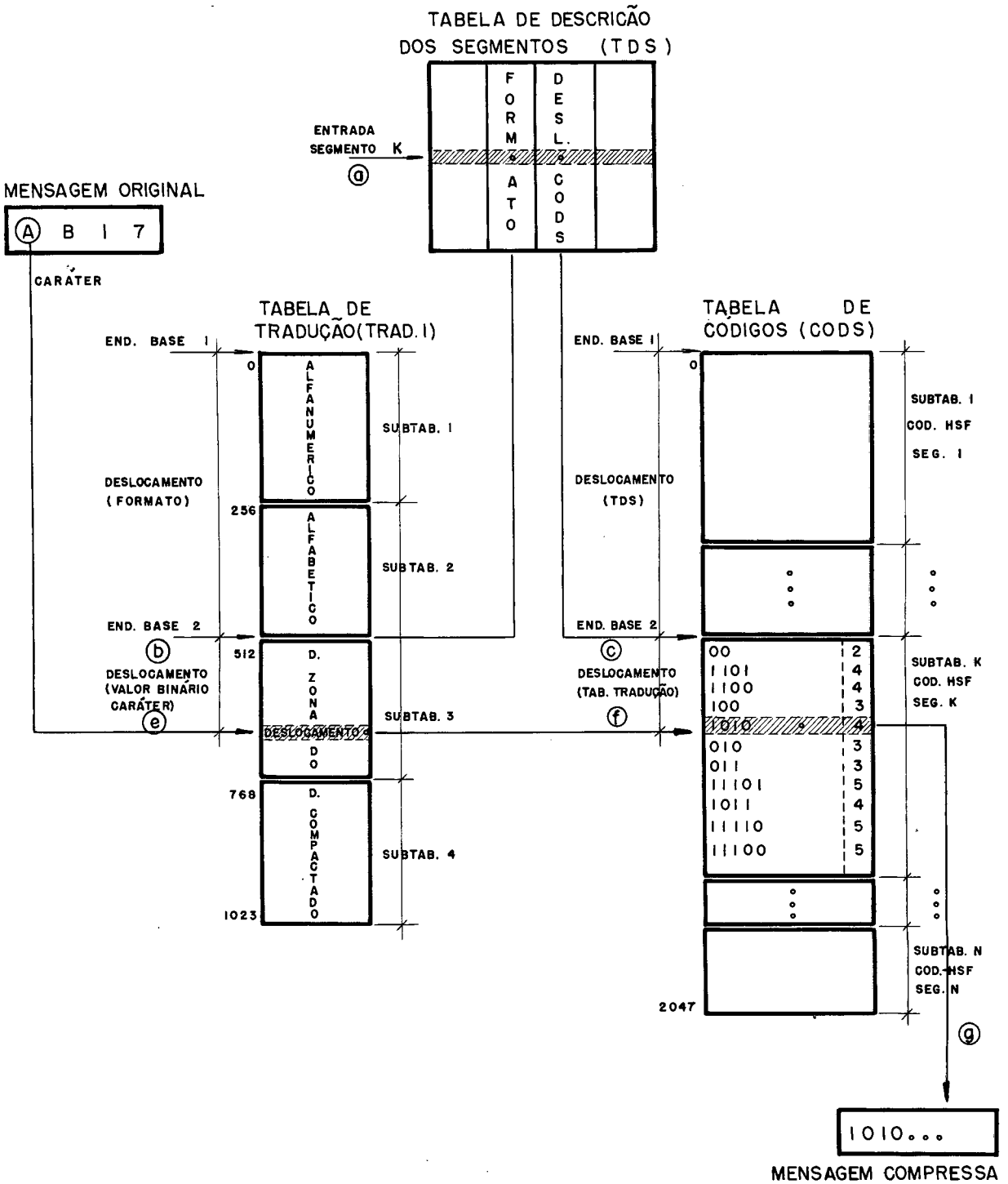
. OBTENÇÃO DO CÓDIGO DE HUFFMAN - SHANNON - FANO (HSF)

Com relação a esta etapa, pouco temos a acrescentar permanecendo aqui todos os conceitos e procedimentos emitidos anteriormente quando da apresentação do código HSF.

Devemos lembrar que para o nosso sistema os alfabetos são pré-fixados de acordo com o formato.

Para efeito de determinação do código HSF é dado a cada elemento uma frequência inicial de 1, como veremos de forma prática detalhdamente, no capítulo IV, na apresentação do Programa Gerador de Códigos (COMPØ4Ø).

Rotina de Compressão:



Procedimentos de Compressão:

a) Posicionar a Tabela de Descrição de Segmentos (TDS) na entrada referente ao segmento a ser processado.

b) Obter o endereço base da subtabela de Tradução relativa ao segmento, pela adição do endereço base da Tabela de Tradução ao formato obtido da Tabela de Descrição de Segmentos.

c) Obter o endereço base da Subtabela de Códigos relativa ao segmento, pela combinação do endereço base da Tabela de Códigos ao deslocamento, obtido da Tabela de Descrição dos Segmentos, indicativo da posição relativa da subtabela na tabela.

d) Se todos caracteres do segmento foram processados, terminou compressão do segmento, e se não for o último segmento voltar ao item "a". Se último,encerrou compressão do registro.

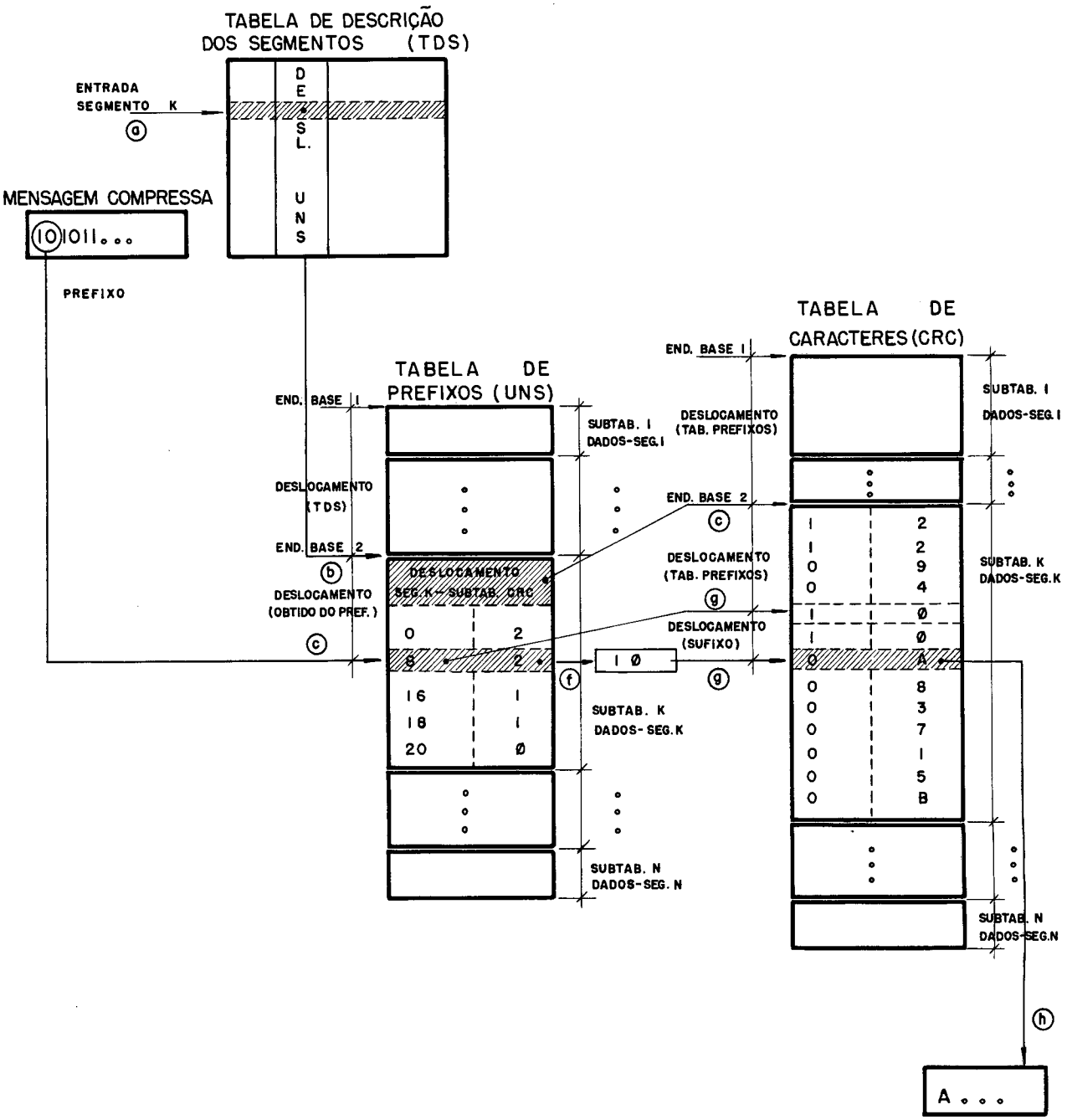
e) Obter o próximo caráter fonte da mensagem original. Utilizando o valor correspondente à sua configuração binária como deslocamento e o endereço base da Subtabela de Tradução (item b), determinar o endereço efetivo de entrada na tabela. Desta tabela colhemos o deslocamento que permitirá retirar da Tabela de Códigos o grupo referente ao caráter fonte selecionado.

f) Com o deslocamento obtido no item anterior, determinamos o endereço efetivo para acesso à Tabela de Códigos, pela combinação deste deslocamento ao endereço base da subtabela de códigos (item "c"). Assim se obtem uma configuração binária de 27 bits complementada com o comprimento do grupo nos últimos 5 bits.

g) Com a configuração binária e comprimento obtidos do item anterior e sabendo-se queo grupo está alinhado à esquerda retiramos o grupo do código HSF correspondente ao caráter fonte e colocamos na mensagem compressa.

h) Voltar ao item "d".

Rotina de Descompressão:



III – 35

Procedimentos de Descompressão:

a) Posicionar a Tabela de Descrição de Segmentos (TDS) na entrada referente ao segmento a ser processado.

b) Obter o endereço base da Subtabela de Prefixos relativa ao segmento, pela combinação do endereço base da Tabela de Prefixos ao deslocamento obtido da Tabela de Descrição de Segmentos, indicativo da posição relativa da subtabela

na tabela.

c) Obter o endereço base da Subtabela de Caracteres, relativa ao segmento, pela combinação do endereço base da Tabela de Caracteres ao deslocamento obtido da primeira entrada da Tabela de Prefixos, indicativo da posição relativa da subtabela na tabela.

d) Se o segmento compresso foi totalmente processado, terminou a descompressão do segmento e se não for o último segmento voltar ao item "a". Se o último encerrou descompressão do registro.

e) Obter próximo prefixo da mensagem compressa e uma vez que a cada bit do prefixo corresponde um deslocamento de 2 bytes (1 entrada da tabela), teremos ao final o deslocamento desejado. Este deslocamento combinado ao endereço base da Subtabela de Prefixos (item "b") nos dá o endereço efetivo de entrada na Tabela. Desta tabela retiramos o comprimento do maior sufixo e o deslocamento que indica a localização, na Subtabela de Caracteres, dos dados referentes ao grupo com o prefixo em questão.

f) Com o comprimento do maior sufixo, obtemos os bits da mensagem compressa.

g) Com o deslocamento obtido no item "e" e o sufixo obtido no item "f", determinamos o endereço efetivo de acesso à Tabela de Caracteres, pela combinação destes 2 elementos ao endereço base da Subtabela de Caracteres (item "c"). Desta forma obtemos o caráter fonte desejado e o número de bits em excesso do sufixo que deverá retornar à mensagem compressa.

h) Armazenar o caráter fonte na mensagem original.

i) Retornar os bits excedentes do sufixo do grupo à mensagem compressa.

j) Voltar ao item "d".

VANTAGENS/DESVANTAGENS

Com esta esquematização o método proposto mantem as mesmas vantagens/desvantagens do método original, uma vez que o nosso objetivo principal aqui, foi o de desenvolver uma metodologia que desse maior velocidade aos processos de codificação/decodificação, embora, aumentando um pouco mais o consumo de memória real para armazenamento das tabelas necessárias.

3. IMPLANTAÇÃO DO SISTEMA

3.1. SUGESTÃO PROPOSTA

Por Implantação do Sistema entendemos as alterações que devem ser processadas nos programas do usuário, de modo que os mesmos, através da subrotina de compressão/descompressão, possam ter acesso a registros do arquivo compresso.

Lembrando que cabe à subrotina de compressão/descompressão toda e qualquer manipulação do arquivo compresso, decidimos entre as várias opções de alteração dos programas, por aquela que, embora um pouco mais trabalhosa, torna o programa simples e mais compreensivo.

A sugestão proposta, consiste basicamente em se eliminar do programa do usuário toda e qualquer referência ao arquivo agora compresso, substituindo os comandos de leitura e gravação por chamadas à subrotina de compressão/descompressão, a qual se utiliza de uma área na "WORKING-STORAGE SECTION" para passar ou receber o registro descompresso. Por este processo, além das vantagens anteriormente citadas, conseguimos, pela eliminação das áreas de entrada/saída, definição do arquivo, etc, reduzir substancialmente a memória real requerida pelo programa original.

Observamos, que a subrotina de compressão/descompressão desenvolvida aceita somente chamadas de programas escritos em COBOL.

3.2. ALTERAÇÕES PARA IMPLANTAÇÃO DA SUGESTÃO PROPOSTA

3.2.1. Comando de Leitura

READ nome-arquivo RECORD (1) INTO identificador (2) AT END comando-imperativo (3)

a) A parte (1) do comando de leitura será introduzida por meio de uma chamada à subrotina de descompressão, a qual passará o registro descompresso ao programa principal através de um parâmetro definido na "WORKING STORAGE SECTION".

Este parâmetro, caso a parte (2) do comando não exista, poderá ser a própria descrição do registro, transferida da "FILE SECTION".

b) Para introduzir a parte (2) do comando de leitura bastará apenas que o parâmetro da subrotina seja o identificador já definido originalmente na "WORKING-STORAGE SECTION".

Neste caso uma vez que o registro obtido do arquivo é colocado e processado numa área de trabalho (identificador), não será necessário transferir, da "FILE SECTION" para a "WORKING-STORAGE SECTION", as entradas relativas à descrição original do registro.

c) Quanto à parte (3) do comando de leitura, a implantação foi feita pela introdução de um comando "IF",o qual testando o conteúdo da primeira posição do registro recebido através do parâmetro da subrotina, determina se a condição de fim do arquivo foi atingida, o que em caso afirmativo implicará na execução dos comandos sob seu controle,que são os mesmos transferidos da cláusula "AT END" (comando-imperativo).

Convencionamos que ao ser detectado o fim do arquivo pela subrotina de descompressão, a mesma passará ao programa principal o carater "*" na primeira posição do parâmetro.

d) Visto isto, são os seguintes os comandos substitutos do comando de leitura:

```
CALL 'DCOMP' USING parâmetro.
IF posição - 1 - parâmetro EQUAL TO '*'
     comando-imperativo.
```

3.2.2. Comando de Gravação

WRITE nome-registro (1) FROM identificador (2) INVALID KEY comando-imperativo (3)

a) A parte (1) do comando de gravação será introduzida, através de uma chamada à subrotina de compressão, a qual receberá o registro descompresso do programa principal através de um parâmetro definido na "WORKING-STORAGE SECTION".

   Este parâmetro, caso a parte (2) do comando não exista, poderá ser a própria descrição do registro transferida da "FILE SECTION".

b) Para introduzir a parte (2) do comando de gravação, bastará apenas que o parâmetro da subrotina seja o identificador já definido originalmente na "WORKING-STORAGE SECTION".

   Neste caso, uma vez que o registro descompresso a ser processado, se encontra numa área de trabalho,(identificador), não será necessário transferir da "FILE SECTION" para a "WORKING-STORAGE SECTION" as entradas relativas à descrição original do registro.

c) A parte (3) do comando não será implantada uma vez que o arquivo é manipulado e definido na subrotina de compressão. Desta forma deixamos a cargo do próprio sistema o controle de chave invalida.

d) Visto isto, é o seguinte o comando substituto do comando de gravação:

```
CALL 'COMP' USING parâmetro.
```

3.2.3. Comando de Abertura

Este comando deverá ser eliminado do programa do usuário, uma vez que esta função seja na compressão e/ou descompressão será executada pela subrotina de compressão/descompressão.

3.2.4. Comando de Fechamento

Pelas mesmas razões apresentadas para o caso do comando de abertura, este comando também deverá ser eliminado do programa do usuário.

Devemos observar, contudo, que para fechar o arquivo compresso por ocasião de sua geração, devemos chamar e passar para a subrotina de compressão (COMP), através da primeira posição do parâmetro o carater '*', que indicará para esta subrotina a situação de fim-de-arquivo.

3.2.5. Comandos Declarativos

Pelo que vimos anteriormente, a sugestão proposta requer a eliminação do programa de todas as referências diretas ou indiretas ao arquivo em questão, a saber:

- Eliminar toda e qualquer declaração referente ao arquivo encontrada na "INPUT-OUTPUT SECTION" nos parágrafos "FILE-CONTROL" e "I-O-CONTROL".
- Eliminar da "FILE SECTION" todas as entradas relativas à descrição do arquivo, ("FILE DESCRIPTION ENTRY").
- Transferir quando necessário e com as devidas adaptações, as entradas relativas à descrição do registro("RECORD DESCRIPTION ENTRY") da "FILE SECTION" para a "WORKING-STORAGE SECTION".

3.2.6. Restrições Gerais

a) O arquivo a ser compresso deverá ser apenas de entrada ou saída, nunca de entrada e saída (I-O).

b) Para efeito de compressão o usuário deverá, de acordo com a 'USAGE', dar para campos e segmentos do registro os seguintes formatos:

b.1) Serão considerados de formato Alfanuméricos e manipulados byte a byte, os campos:

- Alfanuméricos - Display
- Decimal zonado com sinal no início - Display sign leading

- Decimal zonado com sinal no fim - Display sing trailing
- Ponto flutuante externo - Display

b.2) Serão considerados de formato Alfabético e manipulados byte a byte os campos:

- Alfabéticos - Display

b.3) Serão considerados de formato Decimal Zonado e manipulados byte a byte os campos:

- Numéricos com ou sem sinal - Display

b.4) Serão considerados de formato Binário e não compressos pelo código HSF os campos:

- Numéricos com ou sem sinal
  - Computational
  - Computational - 1
  - Computational - 2
  - Computational - 4.

b.5) Serão considerados de formato Decimal Compactado e manipulados de 4 em 4 bits, os campos:

- Numéricos com ou sem sinal - Computational - 3.

3.2.7. Exemplo

Para que se possa sentir de forma prática, a implantação da sugestão proposta, apresentamos, a seguir, um programa exemplo codificado da forma convencional e da forma modificada.

O programa apresentado é bastante simples, executando apenas a leitura e a gravação de um arquivo compresso.

Para melhor clareza do exemplo, grifamos todas as alterações procedidas em cada um dos programas, como poderemos ver a seguir:

Programa Convencional

```
ID DIVISION.
PROGRAM-ID.
    'ORIGINAL'.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT ENTRADA ASSIGN TO SYS010-UT-2400-S.
    SELECT SAIDA   ASSIGN TO SYS011-UT-2400-S.
DATA DIVISION.
FILE SECTION.
FD  ENTRADA
    RECORDING MODE IS F
    LABEL RECORD IS STANDARD
    BLOCK CONTAINS 10 RECORDS
    DATA RECORD IS ENT.
01  ENT   PIC X(80).
FD  SAIDA
    RECORDING MODE IS F
    LABEL RECORD IS STANDARD
    BLOCK CONTAINS 10 RECORDS
    DATA RECORD IS SAI.
01  SAI   PIC X(80).
PROCEDURE DIVISION.
ABERTURA.
    OPEN INPUT ENTRADA
         OUTPUT SAIDA.
TRANSFERE.
    READ ENTRADA AT END
        DISPLAY 'FIM' UPON CONSOLE
        GO TO FECHAMENTO.
    MOVE ENT TO SAI
    WRITE SAI
    GO TO TRANSFERE.
FECHAMENTO.
    CLOSE ENTRADA
          SAIDA.
FIM.
    STOP RUN.
```

Programa Modificado

```
ID DIVISION.
PROGRAM-ID.
    'SUGESTAO-1'.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01  TRABALHO.
    05  ENT.
        10  POS1   PIC X.
        10  RESTO  PIC X(79).
    05  SAI        PIC X(80).
PROCEDURE DIVISION.
TRANSFERE.
    CALL 'DCOMP' USING ENT.
    IF POS1 EQUAL '*'
        DISPLAY 'FIM' UPON CONSOLE
        GO TO FECHAMENTO.
    MOVE ENT TO SAI
    CALL 'COMP' USING SAI.
    GO TO TRANSFERE.
FECHAMENTO.
    STOP RUN.
```

III-36

4. OPERAÇÃO DO SISTEMA

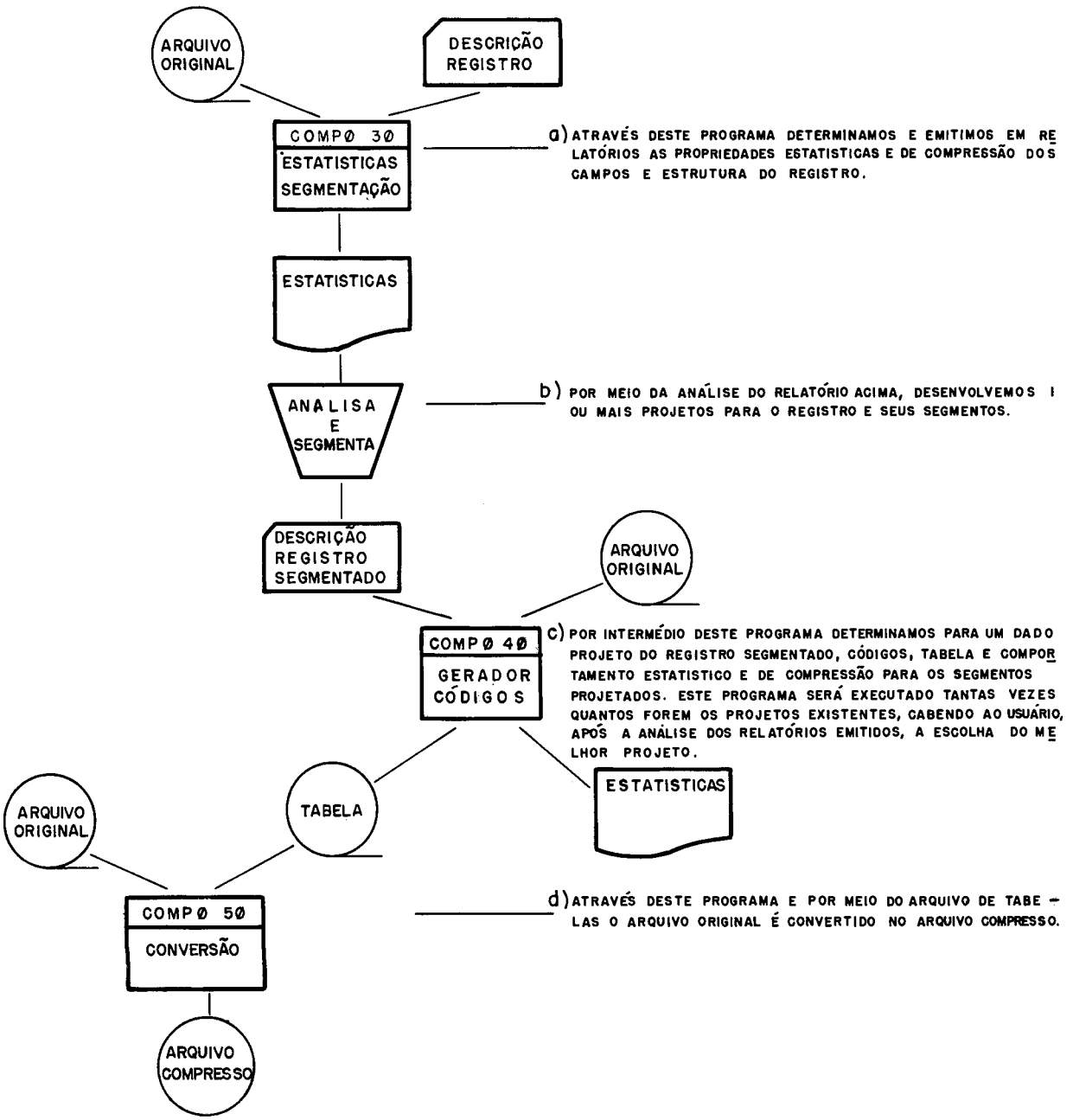
Concluindo, devemos observar que a execução parcial ou integral dos Módulos de Preparação e Execução, dependem das necessidades do usuário para executar uma dada tarefa.

4.1. UTILIZAÇÃO DO MÓDULO DE PREPARAÇÃO

4.1.1. Implantação do Sistema para Operação

Para a implantação do sistema de compressão, devemos executar o Módulo de Preparação completo, já que, por ser a primeira vez, se faz necessário determinar todas as propriedades, constantes, tabelas e códigos referentes ao arquivo a ser compresso.

Assim o usuário deverá proceder, como indicado a seguir:



III - 37

4.1.2. Reavaliação dos Segmentos e suas Propriedades

Sabemos que o processo de segmentar registros está fundamentado na estabilidade estatística de ocorrência das estruturas e campos do registro no arquivo. Torna-se, portanto, conveniente que o usuário proceda periodicamente a uma reavaliação destas propriedades, para que, caso ocorra alguma alteração sensível, se reimplante o sistema.

Assim, para esta reavaliação o usuário deverá:

a) Descomprimir o arquivo, utilizando o programa COMPØ5Ø.

b) Executar os itens "a" e "b" do item 4.1.1.

c) Complementar a reimplantação do sistema, caso se julgue necessário, pela execução dos itens "c" e "d" do item 4.1.1.

4.1.3. Reavaliação dos Códigos e suas Propriedades

Da mesma forma, que no item anterior, métodos de compressão utilizando códigos de tamanho variável são aplicados à informações estatisticamente estáveis, fato normal para grandes volumes de informação. Mesmo assim, é aconselhável reavaliações periódicas destes códigos e suas propriedades, pois somente assim poderemos detectar alterações que venham prejudicar a eficiência global do sistema.

Assim, para esta reavaliação o usuário deverá:

a) Descomprimir o arquivo, utilizando o programa COMPØ5Ø.

b) Executar o item "c" do item 4.1.1.

c) Caso constatado alteração nas propriedades estatísticas das informações no arquivo, executar o item "d" do item 4.1.1.

4.2. UTILIZAÇÃO DO MÓDULO DE EXECUÇÃO

Para utilização deste módulo nada há de especial a destacar.

Devemos apenas observar que o programa deverá usar a rotina de compressão ou de descompressão de acordo se o arquivo processado esteja sendo gravado ou lido do modo apresentado no item 3 "Implantação do Sistema".

CAPÍTULO IV

MÓDULO DE PREPARAÇÃO — PROGRAMAS UTILITÁRIOS

1. APRESENTAÇÃO

Este módulo é a parte do sistema responsável diretamente pela preparação e criação, em uma dada aplicação, de um ambiente propício à implantação efetiva das rotinas de compressão e descompressão.

A "preparação", reúne, todo um esforço para fazer com que a informação a ser compressa satisfaça, tanto quanto possível, às condições ideais exigidas pelo sistema de compressão. No nosso caso, particularmente, a preparação se resume em, dado um registro, tentar segmentá-lo objetivando uma maior taxa de compressão e eficiência do Módulo de Execução.

Com relação à "criação", sabemos que o Módulo de Execução necessita de um conjunto de informações, tais como tabelas, códigos, alfabetos, etc. Este conjunto de informações deverá ser gerado no Módulo de Preparação, por não ser possível, para alguns dados, a sua determinação durante a corrida do Módulo de Execução, além de não ser interessante sobrecarregar este módulo de tarefas, sob pena de comprometer sua eficiência.

Desta forma, para atender aos fins a que se propõe o Módulo de Preparação, foram desenvolvidos 3 programas utilitários, que deverão ser executados periodicamente, a fim de possibilitar uma reavaliação da eficiência e performance do sistema.

Para cada programa, apresentaremos a sua finalidade e os procedimentos de operação, enfocando ainda as entradas e saídas, os procedimentos de manutenção e os roteiros que permitirão ao usuário sempre que necessário modificar partes do mesmo. Informamos também para facilitar referências aos programas por parte do usuário, que serão anexados sempre que desejável os meneumonicos de tabelas, variáveis etc.

Paralelamente à descrição de cada programa e de

seus componentes, apresentaremos de forma prática, todos os procedimentos envolvidos para a implantação do sistema em um arquivo de testes especialmente construído para este fim.

2. SUBROTINA DE LEITURA

2.1. FINALIDADE

Esta subrotina tem por finalidade, efetuar a leitura de qualquer arquivo sequencial, independente da blocagem e comprimento do registro, sendo possível, ainda, por meio de uma pequena alteração, a utilização de qualquer suporte de armazenamento.

Assim, conseguimos dar ao nosso sistema,um caráter mais geral, uma vez que os programas utilitários, através da sua utilização, ficam em condições de ler qualquer arquivo de qualquer aplicação do usuário.

2.2. ENTRADAS

a) Arquivo Aplicação

Arquivo do usuário, de organização sequencial e registros lógicos do mesmo formato e comprimento, a ser processado pela Subrotina de Leitura, cuja versão atual apresenta as seguintes características:

- Endereço lógico - SYSØ1Ø
- Nome interno - ENTRADA 1
- Comprimento máximo do registro lógico - 200 bytes
- Fator de bloco máximo - 60
- Meio físico - FITA (para outros suportes, ver item 2.5. "MODIFICAÇÕES").

b) Arquivo Console

Através da Console, esta subrotina solicitará do usuário as seguintes informações:

b.1) "DE FATOR DE BLOCO - TAMANHO REGISTRO XX-XXX"

Como resposta o usuário fornecerá, o fator de bloco (2 dígitos) e o comprimento em bytes do registro (3 dígitos), que permitirão a deblocagem do arquivo desejado.

De forma prática, para um arquivo cujos registros possuem o comprimento de 52 bytes.com fator de bloco 8, a resposta seria Ø8-Ø52.

Complementando devemos observar mais uma vez que, na atual versão da subrotina ficou estabelecido um máximo de 60 para o fator de bloco e, de 200 bytes para o comprimento do registro lógico.

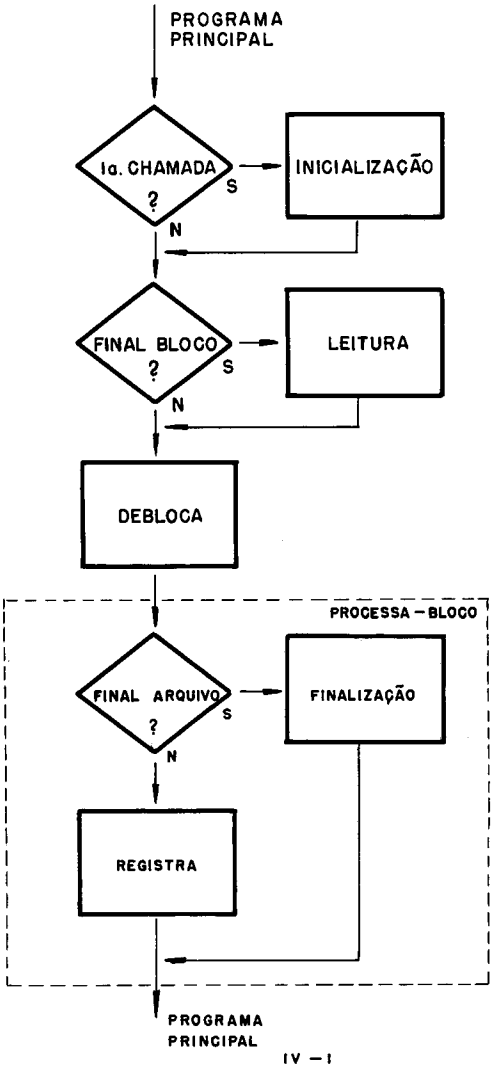
2.3. SAÍDAS

a) Parâmetros

Como resultado, esta subrotina passa ao programa principal, por meio do parâmetro REGISTRO, o próximo registro lógico do arquivo a ser processado e do parâmetro FIM-LEIT um "*" quando o fim de arquivo for detectado.

2.4. ESTRUTURA DO PROGRAMA

2.4.1. Fluxograma Geral



IV - 1

2.4.2. Procedimentos

Como podemos observar no fluxograma apresentado esta subrotina é bastante simples e sua finalidade é proceder a leitura de arquivos para qualquer comprimento de registro e fator de bloco (respeitados os limites pré-estabelecidos). Isto é feito através da utilização de registros de tamanho indefinido sendo, a leitura dos registros físicos (blocos), executada em uma área estruturada segundo um array bidimensional, onde, o número de linhas corresponde ao fator de bloco e o número de colunas ao comprimento do registro, ambas fixa

das pelo usuário no início do processamento.

Para sua execução, os procedimentos abaixo devem ser executados:

a) Chamada, pelo programa principal, por meio do comando abaixo

CALL 'LERFITA' USING parâmetro 1 parâmetro 2.

onde no parâmetro 1 passamos o registro lógico e no parâmetro 2 a indicação, por meio de um '*', de que o final do arquivo foi atingido.

Observamos que o nome 'LERFITA' é devido a versão da subrotina por nós utilizada processar arquivos em fita. Mais adiante, mostraremos o que deve ser feito para que a mesma processe arquivos em outro meio-físico.

b) Módulo de 'Inicialização'

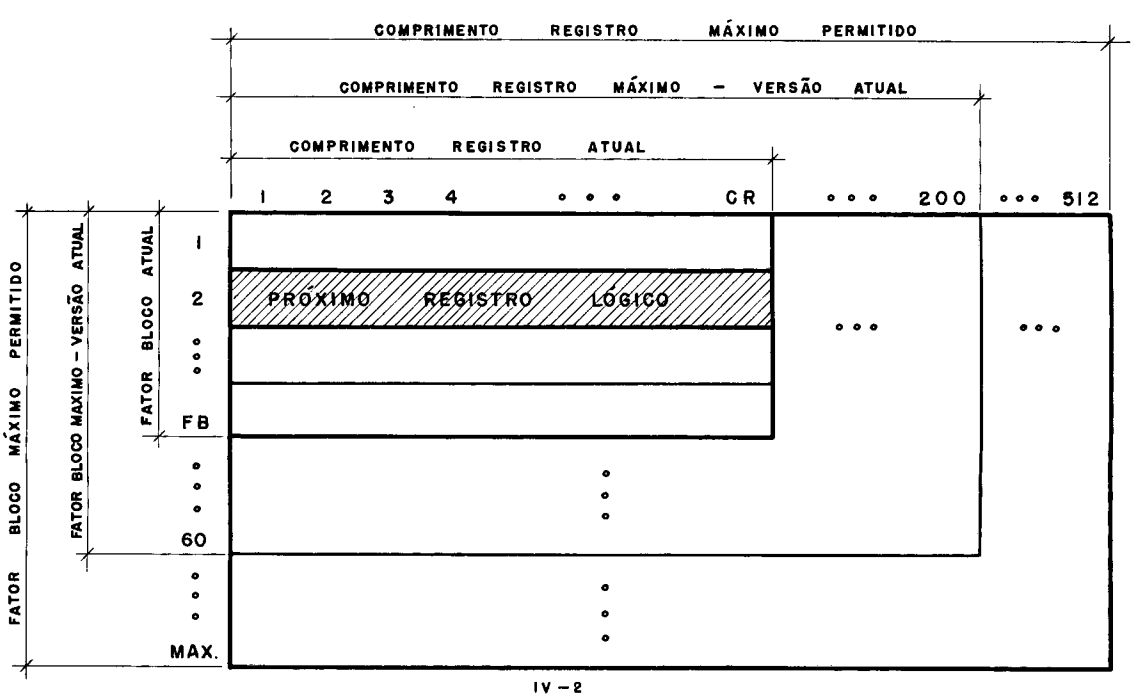
Se esta é a primeira chamada da subrotina, o MÓDULO DE INICIALIZAÇÃO é executado, quando o arquivo é aberto e as constantes são definidas, (fator de bloco e tamanho do registro).

c) Módulo de 'Leitura'

Se nenhum bloco foi lido ou se o bloco anterior foi totalmente processado, proceder a leitura do próximo bloco.

d) Módulo 'Processa Bloco'

Obter o próximo registro lógico do bloco que se encontra em uma linha do array.



Se a primeira posição do registro lógico contém um '*', atingimos o final do arquivo e neste caso procedemos a finalização, fechando o arquivo do usuário e movendo um '*' para FIM-LEIT (parâmetro 2). Caso contrário, movemos o registro lógico obtido para REGISTRO (parâmetro 1) e colocamos um '*' na primeira posição da linha correspondente ao registro, indicando que o mesmo já foi processado. Desta forma evita-se o reprocessamento por ocasião do último bloco que normalmente contém um número de registros menor do que o especificado pelo fator de bloco.

e) Retornar ao programa principal.

2.5. MODIFICAÇÕES

a) Caso os limites máximos estabelecidos para comprimento do registro lógico (200 bytes) e fator de bloco (60), não satisfaçam ao usuário, este poderá alterá-los pela mudança dos valores fixados:

- na cláusula RECORD CONTAINS - 1 a 12000
- nas cláusulas OCCURS, quando definindo o registro de entrada - 60 e 200
- na área de ligação, quando definindo o parâmetro 1 - REGISTRO - 200

Ressaltamos que ao se alterar estes limites, devemos observar o bloco máximo, em bytes, permitido para o meio físico onde se encontra o arquivo e, por motivos do projeto do sistema, o comprimento máximo de 512 bytes para o registro lógico do arquivo.

b) Como a versão apresentada processa apenas arquivos armazenados em fita, caso o usuário deseje utilizar arquivos em outro meio físico, bastará alterar a cláusula SELECT no que se refere à descrição do sistema.

c) Para cada versão desenvolvida, se desejado, poderá se alterar o nome da subrotina encontrada no PROGRAM-ID.

3. SUBROTINA DE IMPRESSÃO

3.1. FINALIDADE

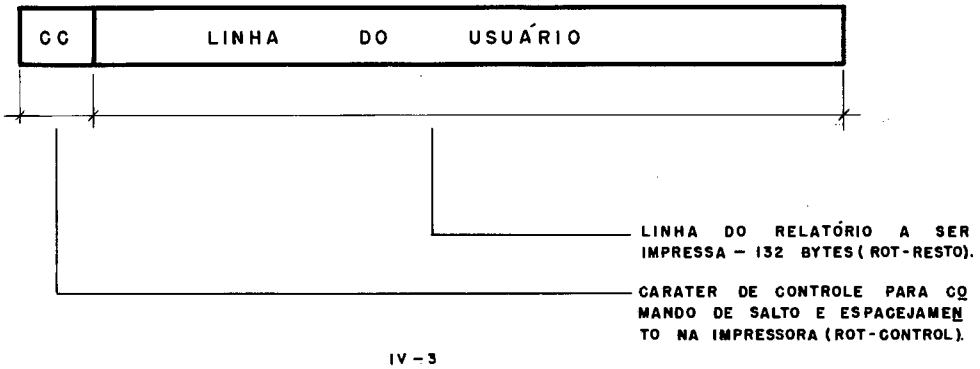
Por ser a impressão, uma função encontrada em mais de um programa do nosso sistema é objetivando sempre a simplificação modular, decidimos pelo desenvolvimento de uma subrotina para imprimir nossos relatórios.

3.2. ENTRADAS

a) Parâmetros

A única entrada para esta subrotina é feita através de 2 parâmetros que são passados à ela pelo programa principal.

O parâmetro 1, denominado ROTLINE, conterá a linha a ser impressa no formato abaixo:



IV-3

Em nossos programas utilizamos o Caráter de Controle ASA:

CÓDIGO	AÇÃO ANTES DE IMPRIMIR O REGISTRO
b	ESPACEJA 1 LINHA
0	ESPACEJA 2 LINHAS
—	ESPACEJA 3 LINHAS
+	NÃO ESPACEJA
1	SALTA PARA LINHA 1 DA PRÓXIMA PÁGINA

IV-4

O parâmetro 2, denominado ROT-CHAVE 1, conterá o caráter "T", indicando a subrotina que o relatório foi concluído.

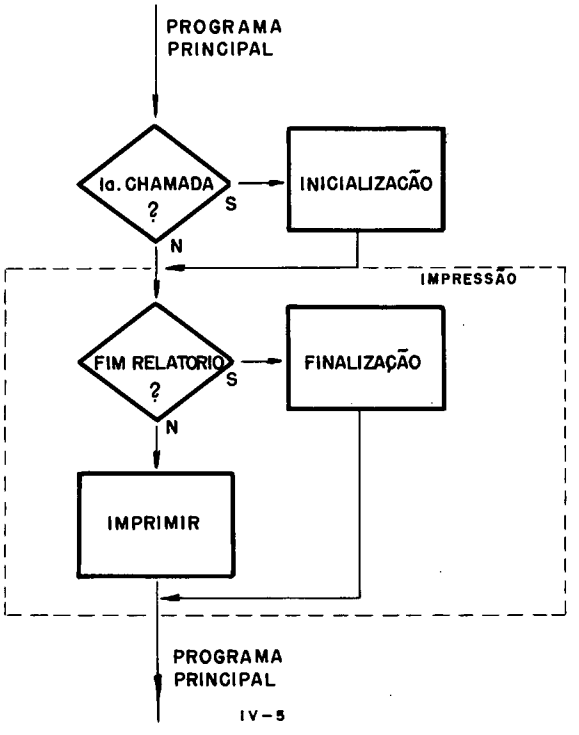
3.3. SAÍDAS

a) Relatório

A única saída é executada na impressora escolhida pelo usuário para impressão do seu relatório, cujo endereço lógico é SYSØØ6 e nome interno - PRINTER.

3.4. ESTRUTURA DO PROGRAMA

3.4.1. Fluxograma Geral



3.4.2. Procedimentos

Esta subrotina não apresenta nenhuma complexidade, necessitando, para sua execução, apenas dos procedimentos abaixo:

a) Chamada pelo programa principal através do comando abaixo:

CALL 'RPRINT' USING parâmetro 1 parâmetro 2.

onde, no parâmetro 1, passamos a linha a ser impressa e no parâmetro 2 passamos o caráter "T" indicando à subrotina que a impressão do relatório foi concluída.

b) Módulo de 'Inicialização'

Se esta for a primeira chamada da subrotina, o MÓDULO DE INICIALIZAÇÃO é executado. Nele o arquivo de impressão (PRINTER) é aberto.

c) Módulo de 'Impressão'

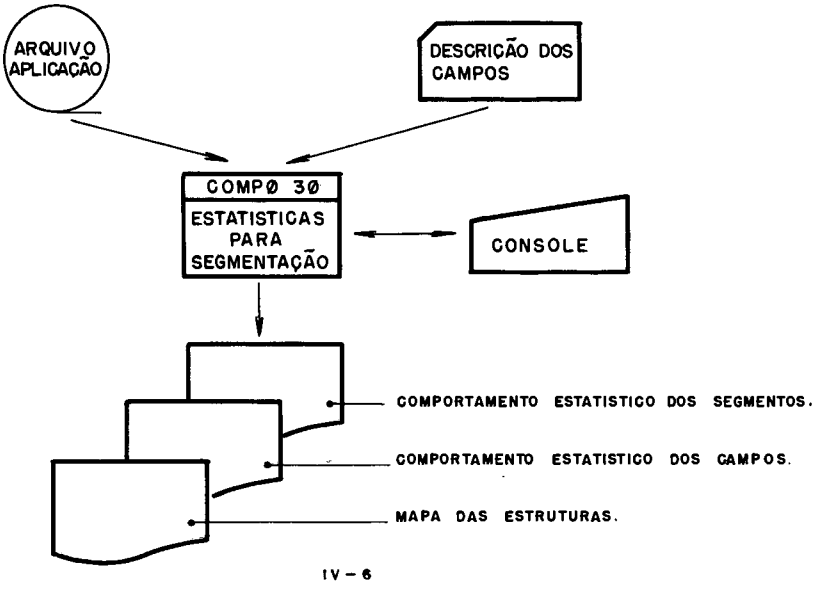
Se o parâmetro 2 (ROT-CHAVE1) contiver o caráter "T" o relatório foi concluído e o módulo de FINALIZAÇÃO é executado procedendo o fechamento do arquivo de impressão. Caso contrário, o conteúdo do parâmetro 1 (ROT-LINE) é impresso de acordo com o caráter de controle encontrado na primeira posição (ROT-CONTROL).

d) Retornar ao programa principal

3.5. MODIFICAÇÕES

a) A única modificação possível é feita na impressora utilizada (1403) se caso o usuário deseje usar um outro modelo, devendo para isto, apenas alterar a cláusula SELECT no que se refere a descrição do sistema.

4. PROGRAMA GERADOR DE ESTATÍSTICAS PARA SEGMENTAÇÃO(COMP030)



IV-6

4.1. FINALIDADE

Este é o primeiro programa utilitário do sistema e também o primeiro a ser executado quando se decide pelo uso da Compressão de Dados.

Um sistema desta natureza, deve ser desenvolvido tendo em vista a sua utilização, de forma generalizada, para toda e qualquer aplicação em arquivos sobrecarregados de informação.

Assim ao se estudar as técnicas de compressão consideramos também este fator e decidimos pelo uso da Segmentação acoplada aos Códigos de Tamanho Variável.

Contudo, esta estrutura tem a sua eficiência comprometida caso o registro com os seus segmentos e campos não apresentem uma boa distribuição, pois, desta forma, maior será o número de segmentos com conteúdo a serem processados pelo código HSF, o que implicará em uma redução na taxa de compressão e em um aumento no tempo de execução.

Procurando minimizar este problema, desenvolvemos este utilitário cuja finalidade é a de fornecer ao usuário todos os subsídios necessários a uma melhor reestruturação dos campos nos segmentos e destes no registro. Estes dados são apresentados sob forma de 3 relatórios que mostram o comportamento estatístico e de compressão das estruturas e campos componentes do arquivo.

4.2. ENTRADAS

a) Arquivo-aplicação

Arquivo do usuário a ser processado pela Subrotina de Leitura que efetuará a deblocagem remetendo a este programa um registro lógico à cada chamada.

Por esta razão tanto a definição como as restrições impostas a este arquivo podem ser encontradas na descrição desta subrotina (item 2).

Características do arquivo de testes:

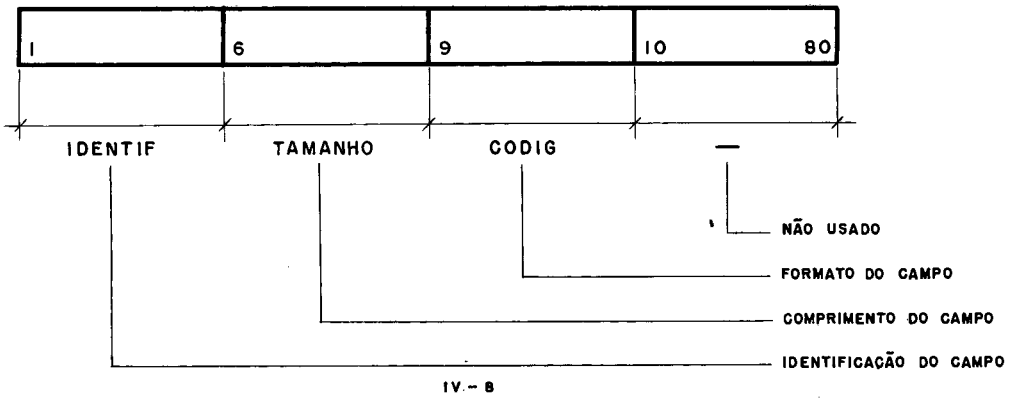
- Comprimento do registro - 80 bytes
- Fator de bloco - 10
- Total de registros no arquivo - 310 registros

```
                                                   A  B  C       D       E    F          G         H    I      J       K     L    M
REC      39    DATA    80                        000150ABCDE            0000000000                1111155555CCCCCCCCCC00000
REC      40    DATA    80                        000150ABCDE            0000000000                1111155555CCCCCCCCCC00000

* * * * DEVICE  280  SYS018,  MODE IS  9 TRACK  1600 BPI       BLOCK    5    DATA   800                                    * * * *

REC      41    DATA    80                        000200                 0000000000                1111155555CCCCCCCCCC00000H IS K
REC      42    DATA    80                        000200                 0000000000                1111155555CCCCCCCCCC00000H IS K
REC      43    DATA    80                        000200                 0000000000                1111155555CCCCCCCCCC00000H IS K
REC      44    DATA    80                        000200                 0000000000                1111155555CCCCCCCCCC00000H IS K
REC      45    DATA    80                        000200                 0000000000                1111155555CCCCCCCCCC00000H IS K
REC      46    DATA    80                        000200                 0000000000                1111155555CCCCCCCCCC00000H IS K
REC      47    DATA    80                        000200                 0000000000                1111155555CCCCCCCCCC00000H IS K
REC      48    DATA    80                        000200                 0000000000                1111155555CCCCCCCCCC00000H IS K
REC      49    DATA    80                        000200                 0000000000                1111155555CCCCCCCCCC00000H IS K
REC      50    DATA    80                        000200                 0000000000                1111155555CCCCCCCCCC00000H IS K

* * * * DEVICE  280  SYS018,  MODE IS  9 TRACK  1600 BPI       BLOCK    6    DATA   800                                    * * * *

REC      51    DATA    80                        000250                 0000000000                1111155555CCCCCCCCCC12345
REC      52    DATA    80                        000250                 0000000000                1111155555CCCCCCCCCC12345
REC      53    DATA    80                        000250                 0000000000                1111155555CCCCCCCCCC12345
REC      54    DATA    80                        000250                 0000000000                1111155555CCCCCCCCCC12345
REC      55    DATA    80                        000250                 0000000000                1111155555CCCCCCCCCC12345
REC      56    DATA    80                        000250                 0000000000                1111155555CCCCCCCCCC12345
REC      57    DATA    80                        000250                 0000000000                1111155555CCCCCCCCCC12345
REC      58    DATA    80                        000250                 0000000000                1111155555CCCCCCCCCC12345
REC      59    DATA    80                        000250                 0000000000                1111155555CCCCCCCCCC12345
REC      60    DATA    80                        000250                 0000000000                1111155555CCCCCCCCCC12345

* * * * DEVICE  280  SYS018,  MODE IS  9 TRACK  1600 BPI       BLOCK    7    DATA   800                                    * * * *

REC      61    DATA    80                        000300                 1111100000                1111155555CCCCCCCCCC00000
REC      62    DATA    80                        000300                 1111100000                1111155555CCCCCCCCCC00000
REC      63    DATA    80                        000300                 1111100000                1111155555CCCCCCCCCC00000
REC      64    DATA    80                        000300                 1111100000                1111155555CCCCCCCCCC00000
REC      65    DATA    80                        000300                 1111100000                1111155555CCCCCCCCCC00000
REC      66    DATA    80                        000300                 1111100000                1111155555CCCCCCCCCC00000
REC      67    DATA    80                        000300                 1111100000                1111155555CCCCCCCCCC00000
REC      68    DATA    80                        000300                 1111100000                1111155555CCCCCCCCCC00000
REC      69    DATA    80                        000300                 1111100000                1111155555CCCCCCCCCC00000
REC      70    DATA    80                        000300                 1111100000                1111155555CCCCCCCCCC00000

* * * * DEVICE  280  SYS018,  MODE IS  9 TRACK  1600 BPI       BLOCK    8    DATA   800                                    * * * *

REC      71    DATA    80                        000350ABCDE            1111100000                1111155555CCCCCCCCCC00000 L
REC      72    DATA    80                        000350ABCDE            1111100000                1111155555CCCCCCCCCC00000 L
REC      73    DATA    80                        000350ABCDE            1111100000                1111155555CCCCCCCCCC00000 L
REC      74    DATA    80                        000350ABCDE            1111100000                1111155555CCCCCCCCCC00000 L
```

b) Arquivo - Descrição dos Campos

Arquivo em cartão, de nome interno 'CARTAO' e endereço lógico - SYSØØ9, cujo conteúdo fornece ao programa uma descrição completa de todos os campos componentes do registro da aplicação a ser processado.

Apresentamos a seguir o formato deste registro com todas as normas necessárias ao preenchimento de seus campos.



IV.-8

b.1) Identificação do Campo (IDENTIF)

Campo numérico de 5 posições cujo conteúdo apresenta um número de 5 dígitos, atribuído pelo usuário de forma sequencial aos campos do registro (esquerda para direita).

Este número tem por finalidade identificar os campos do registro do arquivo de aplicação tanto para uso interno no programa como para os relatórios.

b.2) Comprimento do Campo (TAMANH)

Campo numérico de 3 posições, cujo conteúdo exprime, em bytes, o comprimento de cada campo no registro.

b.3) Formato do Campo (CODIG)

Campo numérico de 1 posição, cujo conteúdo apresenta um código indicativo do formato de cada campo no registro .

Convencionamos para cada formato, os códigos a seguir:

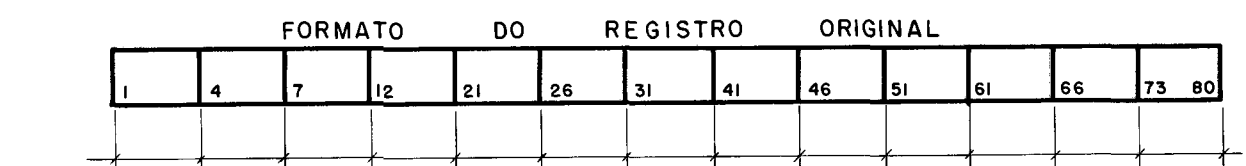
FORMATO	CÓDIGO
ALFANUMÉRICO	1
DECIMA ZONADO	2
DECIMAL COMPACTADO	3
ALFABÉTICO	4
BINÁRIO	5

IV-9

Observamos, para concluir, que a cada campo do registro a ser analisado corresponderá um cartão de descrição, para um máximo de 120 campos, além de um cartão especial localizado no início da massa, no qual informamos, para uso interno do programa:

- fator de bloco - 5 dígitos - IDENTIF
- número de campos do registro - 3 dígitos - TAMANH

Dados de descrição dos campos para o arquivo de testes



CAMPO –	A	B	C	D	E	F	G	H	I	J	K	L	M
IDENTIF. –	00001	00002	00003	00004	00005	00006	00007	00008	00009	00010	00011	00012	00013
TAMANHO –	003	003	005	009	005	005	010	005	005	010	005	007	008
CODIG. –	2	2	1	1	2	2	1	2	2	1	2	1	1

IV.-10

Massa final obtida

00010	0130	
00001	0032	
00002	0032	
00003	0051	
00004	0091	
00005	0052	
00006	0052	
00007	0101	
00008	0052	
00009	0052	
00010	0101	
00011	0052	
00012	0071	
00013	0031	

IV.-11

c) Arquivo Console

Através da console o programa solicitará do usuário as seguintes informações:

c.1) "DE A DATA XX/XX/XX" (DD/MM/AA)

Como resposta,o usuário fornecerá a data desejada no formato pedido, para apresentação nos relatórios.

c.2) "DE O LIMITE - XXXXX"

Para permitir um acompanhamento do comportamento estatístico das informações que estão sendo processadas, o programa está previsto para gerar todos os seus relatórios a intervalos regulares, determinado pelo número de registros processados. Esse número deve ser aqui especificado através de 5 dígitos, pelo usuário.

c.3) "DE FATOR DE BLOCO - TAMANHO REGISTRO XX-XXX"

Esta mensagem é dada pela Subrotina de Leitura que já foi convenientemente explicada.

Diálogo usuário - programa quando da execução dos testes:

dados fornecidos - DATA - 10/01/78
LIMITE - 00200
FATOR DE BLOCO - 10
COMPRIMENTO REGISTRO- 080

```
BG  DE FATOR DE BLOCO-TAMANHO REGISTRO   XX-XXX
BG  C111A AWAITING REPLY
BG  10-080
BG  DE A DATA XX/XX/XX
BG  C111A AWAITING REPLY
BG  10/01/78
BG  DE 0 LIMITE - XXXXX
BG  C111A AWAITING REPLY
BG  00200
```

IV-12

4.3. SAÍDAS

Os resultados oferecidos por este programa, consistem de 3 relatórios, os quais por meio da Subrotina de Impressão são emitidos a intervalos regulares(especificado pelo usuário). Tal fato possibilita um acompanhamento para detectar o ponto em que as informações se tornam estatisticamente estáveis, ou seja, quando as propriedades estatísticas não mais se alteram com o processamento de novos dados. Neste ponto a execução do programa pode e deve ser encerrada.

De um modo geral as informações prestadas por estes relatórios refletem, a cada intervalo, as probabilidades de ocorrência e as compressões obtidas tanto para cada campo do registro como para cada estrutura encontrada. É, por esta razão, uma ferramenta indispensável ao projeto do novo registro.

MAPA DE SEGMENTACAO 10/01/78 PAG - 4

NUM. EST.	ESTRUTURA DOS CAMPOS 0 1 2 3 4 5 6 7 8 9 0 1 2
	123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
1	01000000100000
2	01000001111000
→ 3	01000001110100
4	01000001111100
5	01001001101100
6	01001001111000
7	01001101111000
8	01100001111000
9	01100001110100
10	01100001111100
11	01100101110100
12	01101001101000
13	01101001110100
14	01101101110100

Como podemos observar, este relatório fornece ao usuário toda a relação de estruturas encontradas até o momento de sua emissão.

Componentes do relatório:

a.1) Número da Estrutura (NUM.EST.)

Número atribuído pelo programa a cada estrutura, utilizado para sua identificação nos relatórios.

Por exemplo, estrutura número 2.

a.2) Estrutura dos Campos

Aqui apresentamos as estruturas onde cada campo está identificado por um número sequencial(1 a 120) atribuído pelo usuário e representado dependendo do seu conteúdo, pelo dígito 1 ou Ø.

Uma vez que a cada estrutura corresponde um número binário e visando facilitar ao usuário uma análise das mesmas, achamos conveniente que a relação fosse apresentada em ordem ascendente do valor correspondente a configuração binária da estrutura.

Por exemplo, a estrutura 2, para os 13 campos do registro, apresenta a configuração binária abaixo, onde os campos 2, 8, 9 e 1Ø se acham com conteúdo:

	0	1	0	0	0	0	1	1	1	0	0	0
Identificação campos	1	2	3	4	5	6	7	8	9	10	11	12	13

MAPA DE SEGMENTAÇÃO - 10/01/78 - PAG - 5

NUM. EST.	FREQ. ESTR.	PROBAL. ESTRUT.	COMP. ORIGINAL EST.	COMP. ORIGINAL ARQ.	COMP. COMPRES. EST.	COMP. COMPRES. ARQ.	COMPR. ABSOLUTA EST.	COMPR. ABSOLUTA ARQ.	COMPR. RELATIVA EST.	COMPR. RELATIVA ARQ.	COMPR. PER. EST.	COMPR. PER. ARQ.	RAIO COMP.
1	10	0,032258	80	800	8	80	72	720	0,900000	0,029032	90	2	10,000000
2	60	0,193548	80	4800	23	1380	57	3420	0,712500	0,137903	71	13	3,478260
3	50	0,161290	80	4000	30	1500	50	2500	0,625000	0,100806	62	10	2,666666
4	20	0,064516	80	1600	28	560	52	1040	0,650000	0,041935	65	4	2,857142
5	10	0,032258	80	800	30	300	50	500	0,625000	0,020161	62	2	2,666666
6	40	0,129032	80	3200	28	1120	52	2080	0,650000	0,083870	65	8	2,857142
7	10	0,032258	80	800	33	330	47	470	0,587500	0,018951	58	1	2,424242
8	40	0,129032	80	3200	28	1120	52	2080	0,650000	0,083870	65	8	2,857142
9	20	0,064516	80	1600	35	700	45	900	0,562500	0,036290	56	3	2,285714
10	10	0,032258	80	800	33	330	47	470	0,587500	0,018951	58	1	2,424242
11	10	0,032258	80	800	40	400	40	400	0,500000	0,016129	50	1	2,000000
12	10	0,032258	80	800	28	280	52	520	0,650000	0,020967	65	2	2,857142
13	10	0,032258	80	800	40	400	40	400	0,500000	0,016129	50	1	2,000000
14	10	0,032258	80	800	45	450	35	350	0,437500	0,014112	43	1	1,777777
	310	0,999998		24800		8950		15850					

Este relatório é uma continuação do anterior. Nele, para cada estrutura e através de sua identificação, são fornecidos os dados estatísticos e de compressão alcançados.

Componentes do relatório:

b.1) Número da Estrutura (NUM.EST.)

Número que permite identificar de maneira única uma determinada estrutura.

Por exemplo, estrutura número 2.

b.2) Frequência de Ocorrência das Estruturas(FREQ.ESTR.)

Indica o número de vezes, que cada estrutura foi observada na amostra. Ao final do relatório é apresentado o total de estruturas observadas que deverá coincidir com o total de registros processados.

Por exemplo, a estrutura número 2 foi observada 60 vezes para um total de 310 registros processados.

b.3) Probabilidade de Ocorrência das Estruturas (PROBAL.ESTRUT.)

Calculada com base na frequência de ocorrência de cada estrutura, a probabilidade de ocorrência nos mostra, na amostra analisada, quais as possibilidades de se encontrar uma determinada configuração de campos.

Por exemplo, a configuração de campos atribuída a estrutura número 2 tem a probabilidade de ocorrência de 0,193548, ou seja, a cada 100 estruturas analisadas há a possibilidade de que 19 sejam a número 2 (19,35%).

b.4) Comprimento Original (COMP.ORIGINAL)

Sob este título, apresentamos para cada estrutura o comprimento inicial expresso em bytes que deverá ser o mesmo do registro (EST.), e as correspondentes participações na composição final da amos-

tra analisada (ARQ.), que é obtida do produto da frequência pelo comprimento inicial da estrutura. Estas participações são acumuladas à cada estrutura. para impressão ao final do relatório, refletindo o tamanho original da amostra.

Para o nosso arquivo de testes, temos:

- comprimento original da estrutura 2 - 80 bytes
- comprimento original da estrutura 2 no arquivo - 80*60 = 4800 bytes
- comprimento original da amostra - 24800 bytes.

b.5) Comprimento Compresso (COMP.COMPRES.)

Sob este título, apresentamos expresso em bytes, para cada estrutura, o comprimento compresso (EST.), que é a quantidade resultante da soma dos tamanhos dos campos presentes na estrutura e a contribuição de cada uma na forma compressa (ARQ.) para a constituição final da amostra. Este valor é obtido do produto da frequência pelo comprimento compresso da estrutura.

Do mesmo modo que no item anterior, esta contribuição é acumulada à cada estrutura para impressão ao final do relatório refletindo o tamanho compresso da amostra.

Para o nosso arquivo de testes, temos:

- comprimento compresso da estrutura 2

campos presentes	tamanho
2	3
8	5
9	5
10	10
	- 23 bytes

- comprimento compresso da estrutura 2 no arquivo - 23*60 = 1380 bytes
- comprimento compresso da amostra - 8950 bytes.

b.6) Compressão Absoluta (COMPR.ABSOLUTA)

Para cada estrutura, expressa em bytes, apresentamos o valor da economia obtida pelo uso da estrutura compressa em substituição à estrutura original, tanto no que se refere à própria estrutura (EST.) como na amostra (ARQ.).

A economia obtida na amostra, para cada estrutura, é acumulada e impressa ao final do relatório, refletindo a redução total conseguida.

Para o nosso arquivo de testes, temos:

- compressão absoluta da estrutura 2
  80 - 23 = 57 bytes
- compressão absoluta da estrutura 2 no arquivo
  4800 - 1380 = 3420 bytes
- compressão absoluta da amostra
  24800 - 8950 = 15850 bytes

b.7) Compressão Relativa (COMP.RELATIVA)

Para que se tenha uma idéia mais exata da redução obtida, apresentamos a compressão da estrutura comparado com a própria estrutura (EST.) e a compressão da estrutura no arquivo comparada com a amostra (ARQ.).

Para a estrutura número 2, temos:

- compressão relativa da estrutura
  $$\frac{80 - 23}{80} = \frac{57}{80} = 0,7125$$
- compressão relativa da estrutura no arquivo
  $$\frac{4800 - 1380}{24800} = \frac{3420}{24800} = 0,1379$$

b.8) Compressão Percentual (COMPR.PER.).

Aqui expressamos, em percentagem, as compressões relativas obtidas no item anterior.

Para a estrutura número 2, temos:

- compressão percentual da estrutura

  0,7125*100 = 71%

- compressão percentual da estrutura no arquivo

  0,1379*100 = 13%

b.9) Raio de Compressão (RAIO COMP.)

Aqui, apresentamos uma outra forma de se visualizar a compressão conseguida, registrando quantas vezes o comprimento original é superior ao comprimento compresso da estrutura.

Observamos que o fator obtido é o mesmo tanto para a estrutura como para a estrutura do arquivo.

Para a estrutura número 2, temos:

- Raio de Compressão da estrutura

  $\frac{80}{23}$ = 3,47826

- Raio de Compressão da estrutura no arquivo

  $\frac{4800}{1380}$ = 3,47826

MAPA DE SEGMENTACAO - 10/01/78 - PAG - 6

NUM. CMP.	POS. INI.	FREQ. CMP.	PROBAB. CAMPO	COMP. ORIG.	PARTIC. TOTAL	PARTIC. COMPR.	COMPR. ABSOLUTA	COMPR. RELATIVA	COMP. PERC.	COMP. REL. ARQUIVO	FORMATO CAMPO
1	1	0	0,000000	3	930	0	930	1,000000	100	0,037500	DECIMAL ZONADO
2	4	310	1,000000	3	930	930	0	0,000000	0	0,000000	DECIMAL ZONADO
3	7	110	0,354838	5	1550	550	1000	0,645161	64	0,040322	ALFANUMERICO
4	12	0	0,000000	9	2790	0	2790	1,000000	100	0,112500	ALFANUMERICO
5	21	90	0,290322	5	1550	450	1100	0,709677	70	0,044354	DECIMAL ZONADO
6	26	30	0,096774	5	1550	150	1400	0,903225	90	0,056451	DECIMAL ZONADO
7	31	0	0,000000	10	3100	0	3100	1,000000	100	0,125000	ALFANUMERICO
8	41	310	1,000000	5	1550	1550	0	0,000000	0	0,000000	DECIMAL ZONADO
9	46	300	0,967741	5	1550	1500	50	0,032258	3	0,002016	DECIMAL ZONADO
10	51	280	0,903225	10	3100	2800	300	0,096774	9	0,012096	ALFANUMERICO
11	61	50	0,161290	5	1550	250	1300	0,838709	83	0,052419	DECIMAL ZONADO
12	66	110	0,354838	7	2170	770	1400	0,645161	54	0,056451	ALFANUMERICO
13	73	0	0,000000	8	2480	0	2480	1,000000	100	0,100000	ALFANUMERICO
		1590		80	24800	8950	15850				

Este relatório, expõe, para cada campo definido no registro da aplicação, todos os dados estatísticos e de compressão resultantes da amostra analisada.

Sua principal finalidade é auxiliar o usuário durante o projeto dos segmentos, ou seja, quando se está definindo os campos que irão compor os vários segmentos.

Componentes do relatório:

c.1) Número do Campo (NUM.CMP.)

Identificação fornecida, na entrada, pelo usuário através da descrição dos campos.

Por exemplo, campo número 3.

c.2) Posição Inicial (POS.INI.)

Para facilitar a localização, fornecemos para cada campo a sua posição inicial(em bytes) no registro (posição do primeiro byte do campo relativo ao início do registro).

Por exemplo, o campo número 3, se encontra na posição 7 do registro, ou seja, o primeiro byte do campo se encontra a 7 bytes do início do registro.

c.3) Frequência dos Campos com Conteúdo(FREQ.CMP.)

Indica o número de vezes que cada campo do registro foi observado na amostra, com conteúdo, apresentando ao final do relatório o número total de campos preenchidos.

Por exemplo, o campo número 3 apresentou-se com conteúdo, 110 vezes nos 310 registros analisados e o número total de campos com conteúdo na amostra atingiu a 1590.

c.4) Probabilidade de Ocorrência dos Campos com Conteúdo (PROBAB.CAMPO)

Com base na frequência dos campos com conteúdo, calculamos para cada campo do registro a probabilidade de encontra-lo com conteúdo em um dado re-

gistro da amostra.

Para o campo número 3, por exemplo, temos a probabilidade 0,354838 (35,48%) de encontrá-lo preenchido quando analisando os campos de um registro da amostra.

c.5) Comprimento Original (COMP.ORIG.)

Aqui, para cada campo, apresentamos, expresso em bytes, seu comprimento inicial, o qual será acumulado para emissão ao final do relatório. Observamos que este total deverá ser igual ao comprimento do registro da aplicação. Em caso contrário, erros devem ter sido cometidos quando da descrição dos campos do registro.

Para o nosso arquivo de testes, temos:

- comprimento original do campo 3 - 5 bytes
- soma dos comprimentos originais dos campos - 80 bytes

c.6) Participação Total (PARTIC.TOTAL)

Este dado expressa, em bytes,o quanto cada campo contribuiu para a composição final da amostra original, considerando o fato de que o arquivo na sua forma original armazena sempre todos os campos do registro, estejam eles com ou sem conteúdo.

Estas contribuições são acumuladas campo a campo para emissão ao final do relatório representando o tamanho original da amostra.

Para o nosso arquivo de testes, temos:

- Participação total do campo 3

  Comprimento do campo - 5 bytes

  Número de registros da amostra - 310 registros

  Participação total - 310*5 = 1550 bytes.

- Comprimento original da amostra - 24800 bytes.

c.7) Participação Compressa (PARTIC. COMPR.)

Este dado expressa, em bytes, a contribuição de cada campo para a composição final da amostra compressa, considerando o fato de que o arquivo na sua forma compressa só armazena os campos do registro que estejam com conteúdo.

Da mesma forma que no item anterior as contribuições são acumuladas campo a campo para emissão ao final do relatório, refletindo o tamanho compresso da amostra.

Para o nosso arquivo de tetes, temos:

- Participação compressa do campo 3
  Comprimento do campo - 5 bytes
  Frequência de preenchimento do campo - 110
  Participação compressa - 110*5 = 550 bytes
- Comprimento compresso da amostra - 8950 bytes.

c.8) Compressão Absoluta (COMPR.ABSOLUTA)

Para cada campo, expresso em bytes, apresentamos a economia resultante da eliminação da amostra, dos campos sem conteúdo.

A economia obtida é acumulada campo a campo e impressa ao final do relatório, refletindo a redução total conseguida na amostra.

Para o nosso arquivo de testes, temos:

- Compressão absoluta do campo 3
  1550 - 550 ▫ 1000 bytes
- Compressão absoluta da amostra
  24800 - 8950 = 15850 bytes

c.9) Compressão Relativa (COMPR.RELATIVA)

Para cada campo apresentamos a compressão obtida pela eliminação dos campos sem conteúdo em relação à participação total deste campo na amostra.

Para o campo número 3, temos:

- Compressão relativa

$$\frac{1550 - 550}{1550} = \frac{1000}{1550} = 0,645161$$

c.10) Compressão Percentual (COMP.PERC.)

Expressamos, em percentagem, as compressões relativas obtidas no item anterior.

Para o campo número 3, temos:

- Compressão percentual

0,645161 x 100 = 64%

c.11) Compressão Relativa ao Arquivo (COMP. REL.ARQUIVO)

Para cada campo apresentamos a compressão obtida pela eliminação dos campos sem conteúdo em relação ao tamanho original da amostra.

Para o campo número 3, temos:

- Compressão relativa ao arquivo

$$\frac{1550 - 550}{24800} = \frac{1000}{24800} = 0,040322 \ (4\%)$$

c.12) Formato do Campo (FORMATO CAMPO)

Para esclarecimentos complementares apresentamos por extenso o formato do campo, que foi fornecido pelo usuário quando da descrição dos campos do registro.

O campo número 3, por exemplo, tem o formato 'ALFANUMÉRICO'.

4.4. UTILIZAÇÃO DOS RELATÓRIOS

A utilização destes relatórios, permitirá ao usuário segmentar registros tendo em vista tanto uma maior taxa de compressão (pela eliminação de segmentos sem conteúdo), como também uma distribuição mais racional e lógica para os campos nos segmentos e destes no registro.

Para atingir estes objetivos, apresentamos a seguir algumas regras básicas, que deverão ser observadas du-

rante todo o projeto do registro e seus segmentos:

a) Reunir em cada segmento:

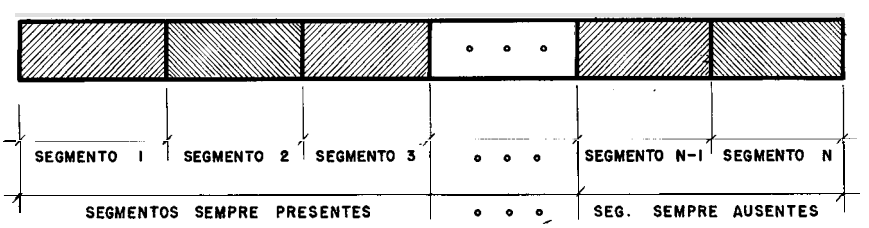
   a.1) Campos cuja presença ou ausência de conteúdo ocorra simultaneamente.

   a.2) Campos cuja compressão resultante seja a maior possível.

   a.3) Campos de mesmas características como formato e outras.

   a.4) Campos relacionados logicamente como "nome" e "endereço" em um segmento de identificação.

b) Organizar os segmentos no registro, visando obter, do começo para o fim, uma passagem gradativa dos segmentos sempre presentes aos segmentos sempre ausentes.



IV-16

Agora, considerando as regras expostas e utilizando os relatórios obtidos com o nosso arquivo de testes, vamos desenvolver um roteiro para segmentação de registros que sirva de orientação para o usuário.

Procedimentos para o projeto de um registro segmentado:

a) Retirar do Relatório "Comportamento Estatístico das Estruturas", as estruturas que ofereçam maior compressão no arquivo.

ESTRUTURA	COMPRESSÃO PERCENTUAL
2	13
3	10
4	4
6	8
8	8

IV-17

b) Selecionar do "Mapa das Estruturas" as configurações correspondentes à cada uma das estruturas.

ESTRUTURA	CONFIGURAÇÃO												
	1	2	3	4	5	6	7	8	9	10	11	12	13
2	0	1	0	0	0	0	0	1	1	1	0	0	0
3	0	1	0	0	0	0	0	1	1	1	0	1	0
4	0	1	0	0	0	0	0	1	1	1	1	0	0
6	0	1	0	0	1	0	0	1	1	1	0	0	0
8	0	1	1	0	0	0	0	1	1	1	0	0	0

IV - 18

c) Analisar as configurações escolhidas e selecionar os campos que possuem as mesmas características quanto a se encontrarem sempre com ou sem conteúdo.

1	2	3	4	5	6	7	8	9	10	11	12	13
0	1	X	0	X	0	0	1	1	1	X	X	0

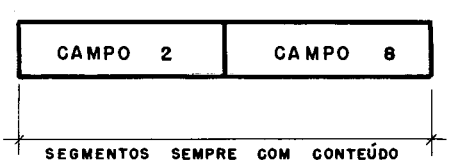
IV - 19

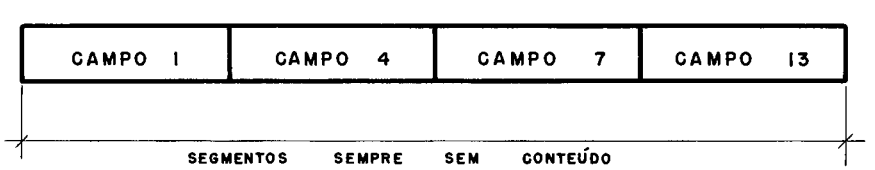
O 'X' na configuração acima representa campos indefinidos quanto às suas características.

d) Considerando a configuração acima e o "Mapa das Estruturas", determinar os campos sempre com e sem conteúdo.

- campos sempre com conteúdo - 2 e 8
- campos sempre sem conteúdo - 1, 4, 7 e 13.

e) Considerando as regras "a.3" e "a.4" agrupar em 2 ou mais segmentos os campos sempre com e sem conteúdo.





IV-20

f) Analisar os campos restantes da configuração do item "c" que não estejam marcados com "x" e, com base nos relatórios "Comportamento Estatístico dos Campos e Estruturas", e nas regras do item "a", tentar agrupá-los em segmentos.

IDENTIFICAÇÃO SEGMENTO	IDENTIFICAÇÃO CAMPOS
A	6
B	9
C	10

IV-21

g) Proceder da mesma forma que na etapa anterior para os campos marcados com "x".

IDENTIFICAÇÃO SEGMENTO	IDENTIFICAÇÃO CAMPOS
D	3
E	5
F	11
G	12

IV-22

h) Utilizando-se agora das regras do item "a" e consultando sempre que necessário todos os relatórios, tentar formar novos segmentos pela concatenação dos anteriormente determinados.

No arquivo de testes o segmento "F" foi anexado ao segmento sempre ausente pelos motivos anteriormente assinalados.

i) Uma vez que temos os segmentos definidos, formatar o registro segmentado considerando a regra "b"

REGISTRO SEGMENTADO

CAMPO 2	CAMPO 8	CAMPO 9	CAMPO 10	CAMPO 3	CAMPO 12	CAMPO 5	CAMPO 6	CAMPO 1	CAMPO 4	CAMPO 7	CAMPO 11	CAMPO 13
SEGMENTO 1		SEG. 2	SEG. 3	SEG. 4	SEG. 5	SEG. 6	SEG. 7	SEGMENTO 8				

IV-23

Como acabamos de ver, não é possível definir um roteiro fixo para um trabalho desta natureza, uma vez que partes dele são bastante subjetivas. Sugerimos porém para compensar essa deficiência, que o usuário projete vários registros segmentados e, pela aplicação do COMPØ4Ø (programa utilitário a ser apresentado), determine o projeto que reune maiores vantagens.

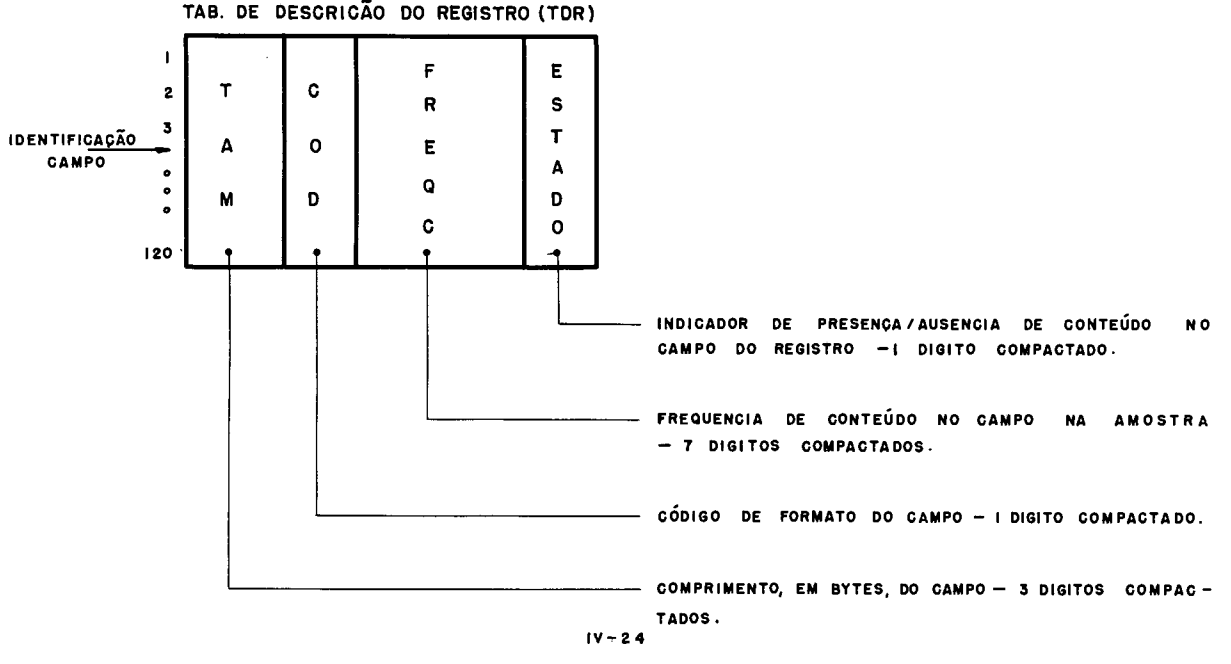
4.5. ESTRUTURA DO PROGRAMA

4.5.1. Tabelas do Programa

a) Tabela de Descrição do Registro (TDR)

Tabela de utilização interna do programa, que por meio da identificação do campo, obtem ou registra informações relativas a campos do registro.

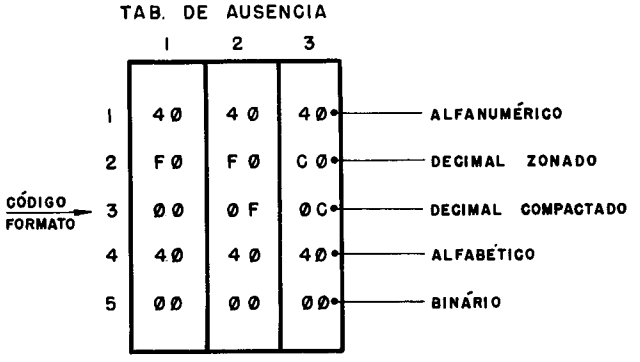
Projetada para um máximo de 120 campos, esta tabela reune em cada entrada os seguintes dados:



IV-24

b) Tabela de Ausência (TABEST2)

Tabela que fornece, para cada formato, os caracteres correspondentes à ausência de conteúdo.

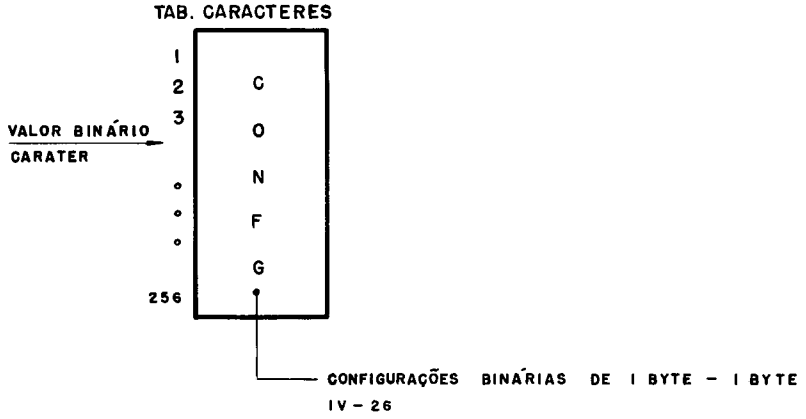


IV - 25

Os caracteres estão representados em hexadecimal, onde as colunas 2 e 3 representam as combinações possíveis de sinal para campos sem conteúdo.

c) Tabela de Caracteres (TABCARAC2)

Tabela auxiliar, utilizada para a composição da configuração binária das estruturas, contendo todas as combinações de bits para 1 byte. Desta forma, é possível armazenar em um programa COBOL qualquer número binário de Ø a 255 em um byte.

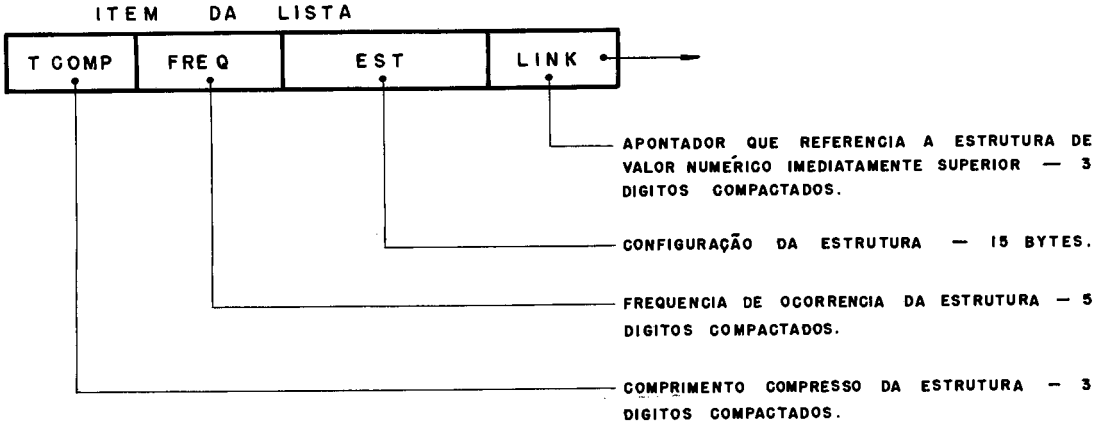


IV - 26

d) Tabela de Descrição das Estruturas (LISTA)

Tabela utilizada pelo programa para registro de todos os dados relativos à cada estrutura encontrada na a mostra.

Projetada para um máximo de 400 estruturas esta tabela reúne os seguintes dados em cada entrada.

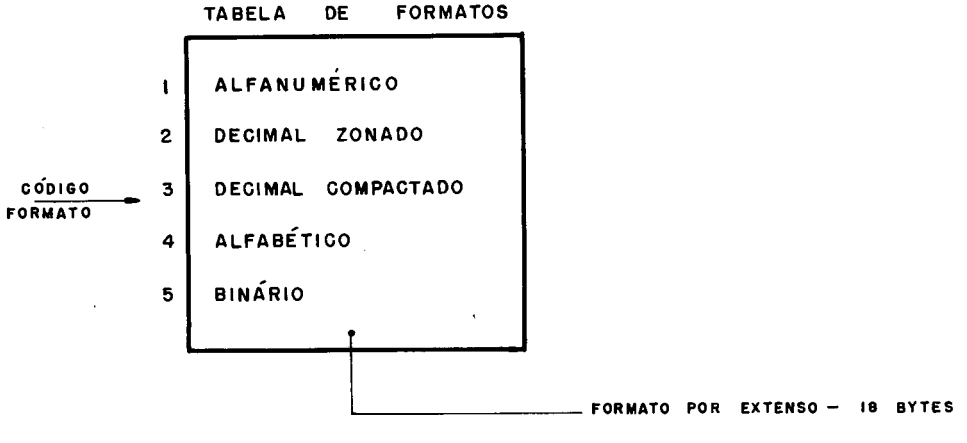


IV-27

Podemos observar que esta tabela está estruturada segundo uma lista a qual mantem as estruturas com os seus dados em ordem ascendente de valor correspondente à configuração binária de cada uma, pois cada estrutura é constituída de dígitos binários "1" e "Ø".

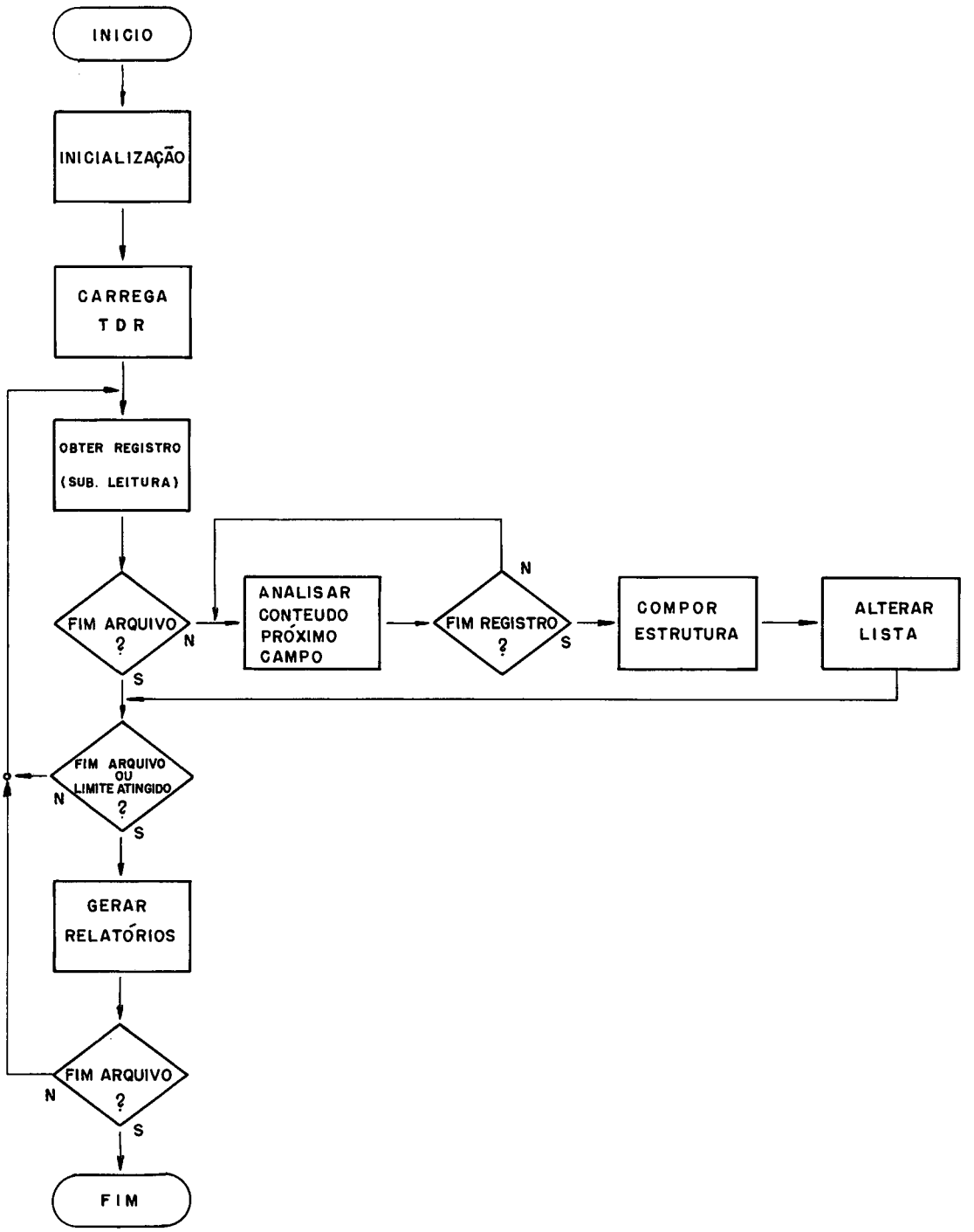
e) Tabela de Formatos (TABFOR)

Tabela, a partir da qual são fornecidos aos relatórios os valores, por extenso, correspondentes aos códigos de formato (FORM).



IV-28

4.5.2. Fluxograma Geral.



IV – 29

4.5.3. Procedimentos

Com relação ao fluxograma chamamos a atenção para o fato de que cada bloco definido poderá abranger 1 ou mais parágrafos no programa, como teremos oportunidade de observar por meio da listagem fonte, em anexo.

Assim para a execução do programa, os seguintes procedimentos devem ser executados:

a) Módulo de 'Inicialização'

Ao iniciar sua execução o programa através deste módulo, requisita do usuário, por meio da console, a DATA e o LIMITE.

b) Módulo 'Carrega TDR'

A execução deste módulo consiste além de algumas inicializações na transferência dos dados da massa de "Descrição dos Campos" para a "Tabela de Descrição do Registro" na memória.

Os dados transferidos são "Formato" e "Tamanho" uma vez que o índice da tabela será a própria Identificação do Campo.

Este módulo, caso, ocorra algum problema com relação ao número de campos definidos, dará saída na console da mensagem:

"ERRO NOS CARTÕES"

c) Módulo 'Obter Registro'

A obtenção do próximo registro lógico do arquivo do usuário é feita pela Subrotina de Leitura através do parâmetro RG.

No nosso caso, como o arquivo do usuário se encontra em fita, denominamos esta versão da Subrotina de Leitura de "LERFITA".

```
CALL 'LERFITA' USING RG FIM-LEIT.
```

Se o fim de arquivo foi detectado desviar para executar o item "h".

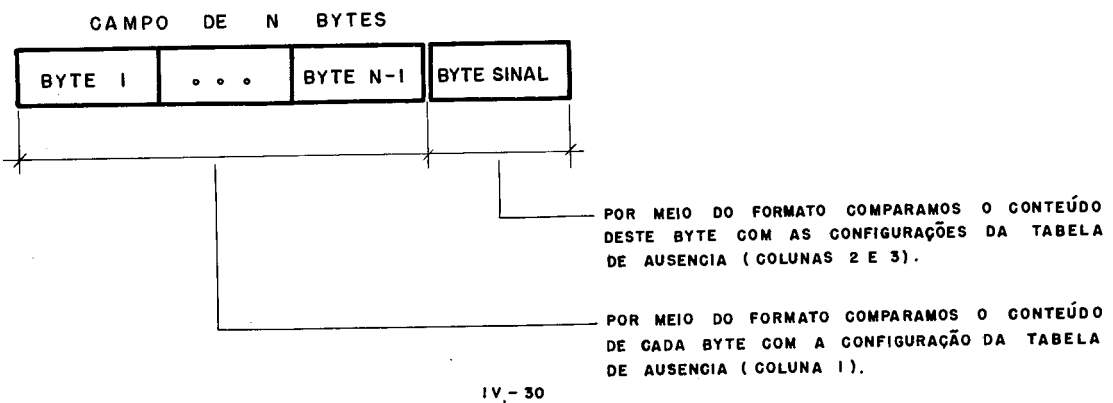
d) Módulo 'Analisar Conteúdo Próximo Campo'

Tomar o próximo campo do registro e verificar a existência ou não de conteúdo.

Esta verificação é feita analisando-se cada campo, cujo comprimento se encontra na Tabela de Descrição do

Registro (TDR). Determina-se que o mesmo está sem conteúdo quando, dependendo do formato obtido da TDR, encontramos zeros ou brancos por valor.

Esta análise é feita para cada campo, byte a byte, da esquerda para a direita, utilizando-se a Tabela de Ausência, a qual contém para cada formato a configuração binária indicativa da ausência de conteúdo para o byte de sinal (coluna 2 e 3) e para os bytes restantes (coluna 1). O campo será considerado com conteúdo no momento em que houver discordância entre o byte do campo e o da tabela. Para padronização do processamento, convencionamos ser o byte mais à direita de todo campo o byte de sinal, independente de seu formato.



IV.- 30

Uma vez determinado o estado do campo (com ou sem conteúdo), o programa através de sua identificação acessa a Tabela de Descrição do Registro, onde atualiza a frequência de preenchimento do campo (FREQC) e o indicador de presença ou ausência de conteúdo (ESTADO). Esta indicação é feita pela colocação, neste campo, do dígito 1 (presente) ou dígito Ø (ausente).

e) Voltar ao item "d", caso ainda exista algum campo do registro a ser analisado.

f) Módulo 'Compor Estrutura'

Uma vez que todos os campos do registro tiveram seus conteúdos verificados e os estados correspondentes registrados na Tabela de Descrição do Registro, passamos a este módulo que, por meio da Tabela de Descrição de Registro e através do indicador de conteúdo (ESTADO), promove a composição final da estrutura obtida.

Esta composição, por motivos de economia de memória, consiste numa representação mais compacta da estrutura (EST), na "LISTA", pois a indicação de presença ou ausência de conteúdo no campo, que anteriormente ocupava 1 byte (TDR-ESTADO) passa agora a ocupar 1 bit (LISTA-EST).

g) Módulo 'Alterar Lista'

De posse da estrutura compacta, este módulo por meio da Tabela de Descrição das Estruturas-(LISTA), e através do apontador "LINK" percorre a LISTA até determinar a presença ou ausência da estrutura.

- Caso a estrutura seja encontrada na lista, deve ser atualizado o campo de frequência de ocorrência(FREQ).
- Caso a estrutura não seja encontrada na lista, deve ser inicializado e adicionado um novo elemento à lista, procedendo a todas as alterações no apontador "LINK" do item a ser adicionado e do item vizinho de modo que os mesmos permaneçam organizados em ordem crescente do valor correspondente à configuração binária da estrutura.

h) Se fim de arquivo (caráter '*' no parâmetro FIM-LEIT) e/ou limite não foi atingido voltar ao item "c".

i) Módulo 'Gerar Relatórios'

Geração dos 3 relatórios do programa.

j) Se fim de arquivo encerrar a execução do programa, caso contrário voltar ao item "c".

4.6. MODIFICAÇÕES

a) Qualquer modificação relacionada com a forma de leitura do Arquivo - Aplicação, deveremos alterar a Subrotina de Leitura. Caso o comprimento máximo do registro tenha sido modificado, deverão ser alteradas neste programas as definições das variáveis RG e REGISTRO.

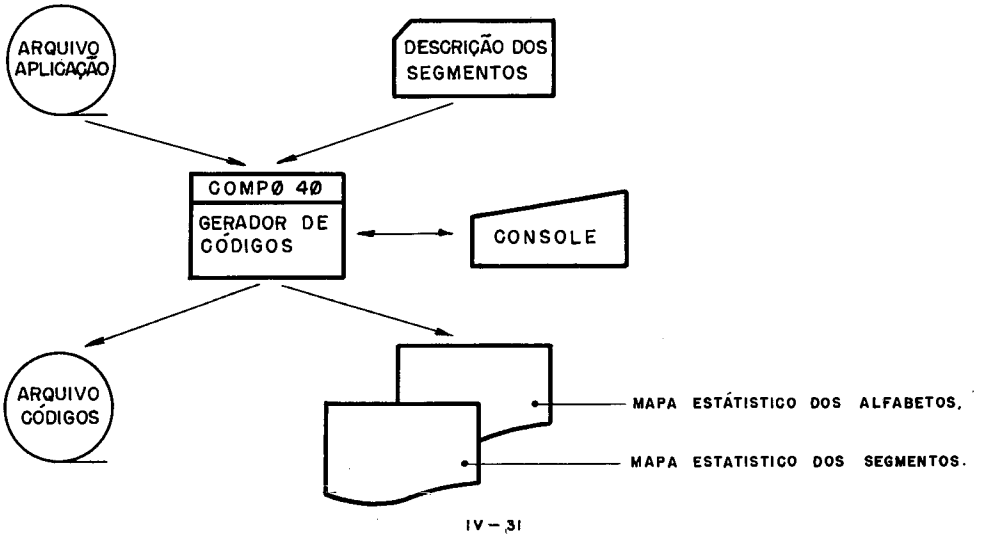
b) Caso o usuário deseje mudar o endereço lógico (SYSØØ9) ou o meio físico do arquivo - Descrição dos Campos, atualmente em cartão, bastará alterar convenientemente a cláusula SELECT no programa.

c) Qualquer modificação com relação a impressão dos relatórios, alterar a Subrotina de Impressão.

d) O número de campos no registro foi limitado ao máximo de 120. Este limite não deve ser alterado uma vez que este dado tem implicações profundas em outros programas.

e) Caso o usuário deseje alterar o limite máximo para a "LISTA" de estruturas, bastará mudar o número de ocorrências na cláusula OCCURS.

5. PROGRAMA GERADOR DE CÓDIGOS (COMPØ4Ø)



IV – 31

5.1. FINALIDADE

Este é o segundo programa utilitário do sistema e o mais importante deste módulo, uma vez que de seus resultados depende o bom desempenho das etapas seguintes.

O usuário, através dos relatórios gerados neste programa, pode, com toda precisão, avaliar, para os segmentos projetados, a compressão resultante da aplicação das Técnicas de Segmentação e Código de Tamanho Variável - HSF ao arquivo. A partir desta avaliação é possível decidir,se para esta formatação do registro, a taxa de compressão alcançada é satisfatória ou não.

Tendo em vista tal possibilidade, o usuário pode, e deve, submeter à este programa um conjunto de diferentes projetos do registro e seus segmentos, de modo que ao final, através de uma análise comparativa dos relatórios emitidos e correspondentes a cada projeto, possa ser escolhido o mais eficiente.

Ao final de sua execução o programa gera um arquivo, utilizado pelas etapas seguintes, e cujo conteúdo consiste dos códigos, tabelas e parâmetros criados a partir do arquivo do usuário e de conformidade com o projeto em análise.

5.2. ENTRADAS

a) Arquivo - Aplicação

Da mesma forma que no programa COMPØ3Ø este arquivo será processado pela Subrotina de Leitura que efetuará a deblocagem remetendo a este programa um registro lógico à cada chamada.

Por esta razão tanto a definição como as restrições impostas à este arquivo podem ser encontradas na descrição desta subrotina (item 2).

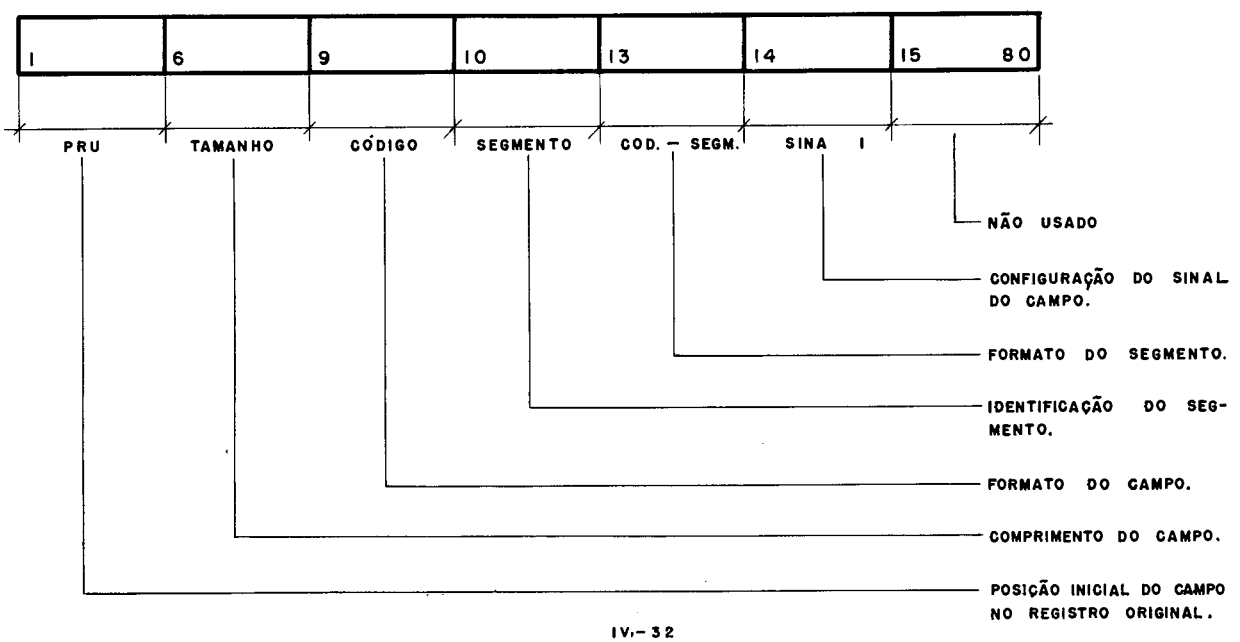
Para melhor exemplificar, utilizamos nosso arquivo de testes, cujas características são:

- comprimento do registro - 80 bytes
- fator de bloco - 10
- total de registros no arquivo - 310 registros

b) Arquivo - Descrição dos Segmentos

Arquivo em cartão, de nome interno 'CARTAO' e endereço lógico - SYSØØ9, cujo conteúdo fornece ao programa uma descrição completa de todos os segmentos definidos pelo usuário, para compor o registro da aplicação a ser processado.

O formato deste registro com todas as normas necessárias ao preenchimento de seus campos é apresentada em seguida.



IV.-32

b.1) Posição Inicial do Campo no Registro Original (PRV)

Campo numérico de 5 posições, cujo conteúdo é um número de 5 dígitos representativo da posição inicial do campo no registro original.

b.2) Comprimento do Campo (TAMANH)

Campo numérico de 3 posições, cujo conteúdo exprime, em bytes, o comprimento de cada campo no registro.

b.3) Formato do Campo (CODIG)

Campo numérico de 1 posição, cujo conteúdo apresenta um código indicativo do formato de cada campo no registro.

Convencionamos para cada formato, os códigos abaixo:

FORMATO	CÓDIGO
ALFANUMÉRICO	1
DECIMAL ZONADO	2
DECIMAL COMPACTADO	3
ALFABÉTICO	4
BINÁRIO	5

IV-33

b.4) Identificação do Segmento (SEGMENTO)

Campo numérico de 3 posições, cujo conteúdo é um número de 3 dígitos, atribuído pelo usuário de forma sequencial aos segmentos do registro(esquerda para direita).

Este número tem por finalidade identificar os segmentos do registro do arquivo de aplicação tanto para uso interno do programa como para os relatórios.

b.5) Formato do Segmento (COD-SEGM)

Campo numérico de 1 posição, cujo conteúdo apresenta um código indicativo do formato de cada segmento no registro.

Utilizar para cada formato os mesmos códigos do item "b.3".

b.6) Configuração de Sinal do Campo (SINAl)

Campo numérico de 1 posição, cujo conteúdo apresenta um código indicativo da configuração binária a ser considerado pelo programa como a configuração do byte de sinal a ser obtido da Tabela de Ausência.

Para padronização de processamento convencionamos o byte mais à direita de todo campo, independente do seu formato, como byte de sinal.

Os códigos a serem considerados de acordo com o formato do campo são:

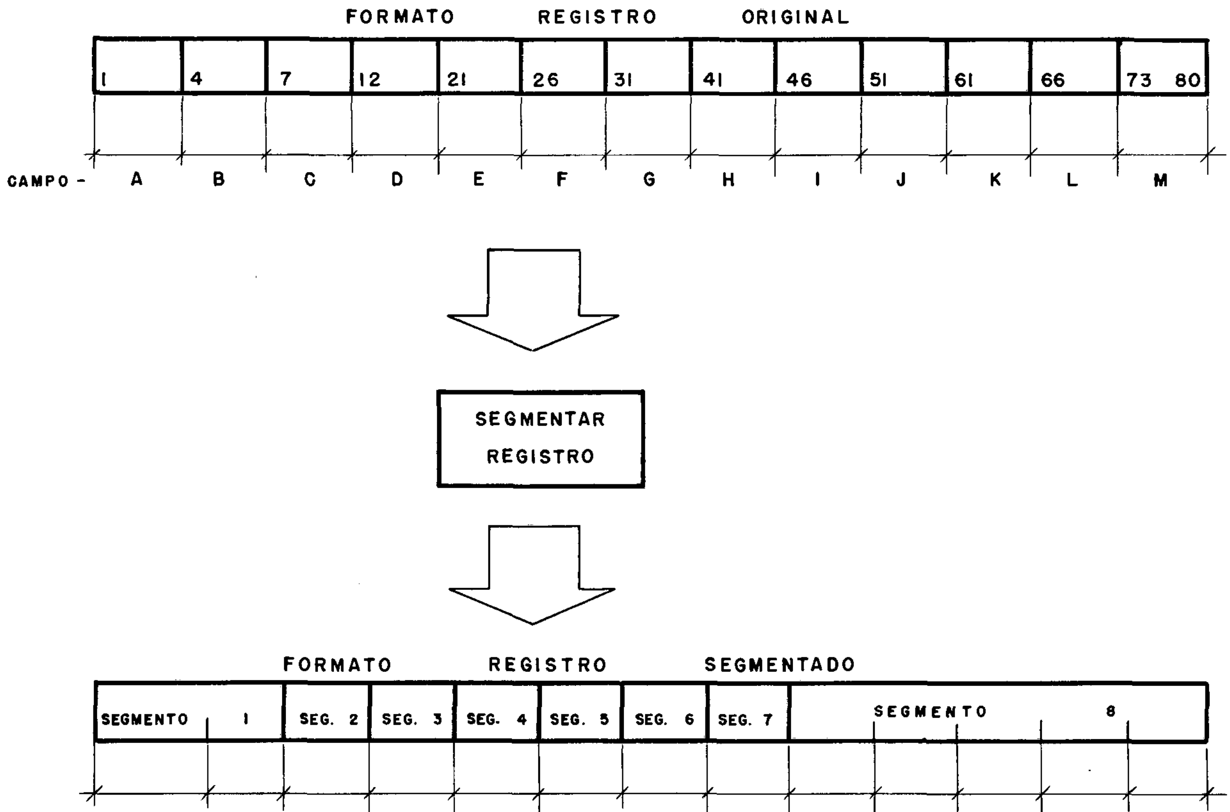
Config. Sinal - 4Ø - FØ - CØ - ØF - ØC - ØØ
Código - 2 - 2 - 3 - 2 - 3 - 2

Observe-se que para cada campo do registro a ser analisado haverá um cartão de descrição, para um máximo de 120 campos, além de um especial posto no início da massa através do qual informamos, para uso interno do programa.

- fator de bloco - 5 dígitos - PRV
- número de campos do registro - 3 dígitos - TAMANH
- número de segmentos do registro - 3 dígitos - SEGMENTO

Concluindo, chamamos a atenção para o fato de que a massa, na entrada, deverá estar organizada de modo que, para cada segmento, os registros de descrição dos campos se apresentem na mesma sequência dos campos no segmento, ou seja, a massa deverá estar classificada de forma a retratar a mesma sequência dos campos no registro segmentado.

Dados de descrição dos segmentos para o arquivo de testes.



CAMPO -	B	H	I	J	C	L	E	F	A	D	G	K	M
PRU -	00004	00041	00046	00051	00007	00066	00021	00026	00001	00012	00031	00061	00073
TAMANH -	003	005	005	010	005	007	005	005	003	009	010	005	008
CODIG -	2	2	2	1	1	1	2	2	1	1	1	1	1
SEGM. -	001	001	002	003	004	005	006	007	008	008	008	008	008
COD-SEG. -	2	2	2	1	1	1	2	2	1	1	1	1	1
SINAL -	2	2	2	2	2	2	2	2	2	2	2	2	2

IV - 34

Massa final obtida

```
00010013000802
00004003200122
00041005200122
00046005200222
00051010100312
00007005100412
00066007100512
00021005200622
00026005200722
00001003100812
00012009100812
00031010100812
00061005100812
00073008100812
```

IV - 35

c) Arquivo Console

Através da console o programa solicitará do usuário as seguintes informações:

c.1) "DE A DATA XX/XX/XX" (DD/MM/AA)

Como resposta, o usuário fornecerá a data desejada no formato pedido, para apresentação nos relatórios.

c.2) "DE O LIMITE - XXXXX"

Para permitir um acompanhamento do comportamento estatístico das informações que estão sendo processadas, o programa está previsto para gerar todos os seus relatórios a intervalos regulares, determinado pelo número de registros processados. Este número deve ser especificado, neste ponto, pelo usuário, de acordo com o formato pedido (5 dígitos).

c.3) "DE O NUM. MAX. DE REGS A SER PROCESSADO - XXXXXX"

Neste ponto o usuário determina o número máximo de registros a ser processado pelo programa (6 dígitos).

c.4) "DE FATOR DE BLOCO - TAMANHO REGISTRO XX-XXX"

Esta mensagem é dada pela Subrotina de Leitura já convenientemente explicada.

Diálogo usuário - programa quando da execução dos testes:

dados fornecidos - DATA - 10/Ø1/78
LIMITE - 00200
NUM.MAX.REGS - 000310
FATOR DE BLOCO - 10
COMPRIMENTO REGISTRO - 080

```
BG  DE A DATA XX/XX/XX
BG  C111A AWAITING REPLY
BG  10/01/78
BG  DE O LIMITE - XXXXX
BG  C111A AWAITING REPLY
BG  00200
BG   DE NUM. MAX. DE REGS A SER PROCESSADO-XXXXXX
BG  C111A AWAITING REPLY
BG  000310
BG  DE FATOR DE BLOCO-TAMANHO REGISTRO    XX-XXX
BG  C111A AWAITING REPLY
BG  10-080                          IV - 36
```

5.3. SAÍDAS

Da mesma forma que no programa anterior (COMPØ3Ø), este programa emite, a intervalos regulares de registros processados e através da Subrotina de Impressão, os seus 2 relatórios, até ser atingido o fim do arquivo ou o número máximo de registros a ser processado.

No primeiro relatório são demonstrados, a cada intervalo, as probabilidades de preenchimento e as compressões resultantes da eliminação dos segmentos sem conteúdo. No segundo relatório são considerados apenas os segmentos com conteúdo sendo apresentado para cada um o comportamento estatístico dos caracteres encontrados, os grupos gerados pela aplicação do código HSF e a compressão final obtida.

Observa-se que a emissão regular destes relatórios permitirá ao usuário detectar o ponto a partir do qual as informações permanecerão estatisticamente estáveis.

MAPA ESTATISTICO DOS SEGMENTOS - 10/01/78 - PAG - 14

NUM SEG	FREQ. SEG.	PROBAB. SEG.	TAM SEG	PART.TOT. ARQ.	PART.COMP ARQ.	COMPR.ABS ARQ.	COMPR.REL ARQ.	COM PER	RAIO COMP ARQ.
1	310	0,201298	8	2480	2480				1,000000
2	300	0,194805	5	1550	1500	50	0,002016		1,033333
3	290	0,181818	10	3100	2800	300	0,012096	1	1,107142
4	110	0,071428	5	1550	550	1000	0,040322	4	2,818181
5	110	0,071428	7	2170	770	1400	0,056451	5	2,818181
6	90	0,058441	5	1550	450	1100	0,044354	4	3,444444
7	30	0,019480	5	1550	150	1400	0,056451	5	10,333333
8	310	0,201298	35	10850	10850				1,000000
	1540	0,999996	80	24800	19550	5250			

DADOS NUM	DOS TAM	DOS PRV	CAMPOS PRN	FORMATO
1	3	4	1	DECIMAL ZONADO
2	5	41	4	DECIMAL ZONADO
1	5	46	9	DECIMAL ZONADO
1	10	51	14	ALFANUMERICO
1	5	7	24	ALFANUMERICO
1	7	66	29	ALFANUMERICO
1	5	21	36	DECIMAL ZONADO
1	5	26	41	DECIMAL ZONADO
1	3	1	46	ALFANUMERICO
2	9	12	49	ALFANUMERICO
3	10	31	58	ALFANUMERICO
4	5	61	68	ALFANUMERICO
5	8	73	73	ALFANUMERICO

Como podemos observar, este relatório nos oferece para cada segmento, segundo sua identificação, todos os dados necessários a uma avaliação da eficiência do método para o registro projetado, permitindo assim que entre vários projetos de registros segmentados se possa escolher o melhor.

Componentes do relatório:

a.1) Número do Segmento (NUM SEG)

Número fornecido pelo usuário, na entrada (descrição dos segmentos), que permite identificar de maneira única um dado segmento.

Por exemplo o segmento 2.

a.2) Frequência dos Segmentos com Conteúdo (FREQ. SEG.)

Indica o número de vezes que cada segmento do registro foi observado na amostra, com conteúdo, apresentando ao final do relatório o número total de segmentos preenchidos.

Por exemplo, o segmento número 2, apresentou-se com conteúdo 300 vezes dos 310 registros analisados e o número total de segmentos com conteúdo na amostra atingiu 1540.

a.3) Probabilidade de Ocorrência dos Segmentos com Conteúdo (PROBAB. SEG.)

Com base na frequência dos segmentos com conteúdo, calculamos para cada segmento do registro, a probabilidade de encontrá-lo com conteúdo entre todos os segmentos preenchidos.

Para o segmento número 2 temos a probabilidade de 0,194805 (19,40%) de encontrá-lo preenchido quando analisando os segmentos com conteúdo de um registro da amostra.

a.4) Tamanho do Segmento (TAM SEG)

Para cada segmento, apresentamos, expresso

em bytes, seu comprimento inicial, o qual será acumulado para emissão ao final do relatório. Observamos, que este total deverá ser igual ao comprimento do registro da aplicação. Em caso contrário erros devem ter sido cometidos quando da descrição dos segmentos do registro.

Para o nosso arquivo de testes, temos:

- Tamanho do segmento 2 - 5 bytes
- Total dos comprimentos dos segmentos - 80 bytes

a.5) Participação Total (PART.TOT.ARQ.)

Este dado expressa, em bytes, o quanto cada segmento contribuiu para a composição final da amostra original, considerando o fato de que o arquivo, na sua forma original, armazena sempre todos os segmentos do registro, estejam eles com ou sem conteúdo.

Estas contribuições são acumuladas segmento a segmento para impressão ao final do relatório, representando o tamanho original da amostra.

Para o nosso arquivo de testes, temos:

- Participação total do segmento 2
  Comprimento do segmento - 5 bytes
  Número de registros da amostra - 310 registros
  Participação total - 310x5 = 1550 bytes
- Comprimento original da amostra - 24800 bytes.

a.6) Participação Compressa (PART.COMP.ARQ.)

Este dado expressa, em bytes, a contribuição de cada segmento para a composição final da amostra compressa, considerando o fato de que o arquivo, na sua forma compressa, só armazena os segmentos do registro que estejam com conteúdo.

Da mesma forma que no item anterior as contribuições são acumuladas segmento a segmento para impressão ao final do relatório, representando o tamanho compresso da amostra.

Para o nosso arquivo de tetes, temos:

- Participação compressa do segmento 2
  Comprimento do segmento - 5 bytes
  Frequência de preenchimento do segmento - 300
  Participação compressa - 300x5 ▫ 1500 bytes
- Comprimento compresso da amostra - 19550 bytes

a.7) Compressão Absoluta (COMPR. ABS. ARQ.)

Para cada segmento, e expresso em bytes, apresentamos a economia resultante da eliminação da amostra dos segmentos sem conteúdo.

A economia obtida é acumulada segmento a segmento e impressa ao final do relatório, refletindo a redução total conseguida na amostra.

Para o nosso arquivo de testes, temos:

- Compressão absoluta do segmento 2
  1550 - 1500 = 50 bytes
- Compressão absoluta da amostra
  24800 - 19550 = 5250 bytes

a.8) Compressão Relativa ao Arquivo (COMPR. REL. ARQ.)

Para cada segmento apresentamos a compressão obtida pela eliminação dos segmentos sem conteúdo em relação ao tamanho original da amostra (Imprime somente se o resultado > Ø).

Para o segmento número 2, temos:

- Compressão relativa ao arquivo

$$\frac{1550 - 1500}{24800} = \frac{50}{24800} = 0{,}002016$$

a.9) Compressão Percentual (COM PER)

Aqui, expressamos em percentagem as compressões relativas obtidas do item anterior (Imprime somente a parte inteira e se o resultado > Ø).

Para o segmento número 2, temos:

- Compressão percentual
  0,002016 x 100 = 0%(não impresso)

a.10) Raio de Compressão (RAIO COMP ARQ.)

Aqui, registramos, para cada segmento, quantas vezes a participação total é superior à participação compressa do segmento na amostra analisada.

Por exemplo, para o segmento número 2, temos:

- Raio de Compressão

$$\frac{1550}{1500} = 1,033333$$

a.11) Dados dos Campos (DADOS DOS CAMPOS)

Para qualquer esclarecimento sobre a definição dos segmentos, apresentamos:

- Número sequencial do campo no segmento (NUM)
- Comprimento de cada campo do segmento (TAM)
- Posição inicial do campo no registro original (PRV)
- Posição inicial do campo no registro segmentado (PRN)
- Formato por extenso de cada campo do segmento (FORMATO)

Por exemplo, para estrutura número 2, temos:

- Número sequencial - 1
- Comprimento do único campo - 5 bytes
- Posição inicial registro original - 46
- Posição inicial registro segmentado - 9
- Formato - Decimal Zonado

b) Mapa Estatístico dos Alfabetos

MAPA ESTATISTICO DOS ALFABETOS - 10/01/78 - PAG.- 15

ALF.		FREQ.	PROB.	GRUPO	CUSTO	COMPR. ORIG. CAR	COMPR. ORIG. ARQ	COMPR. COMP. CAR	COMPR. COMP. ARQ	COMP. CAR	ABSOL. ARQ	COMP. CAR	RELATIVA ARQ	COM.PER CAR	ARQ	RAIO COMP.
SEGMENTO NUMERO - 1				FORMATO - DECIMAL ZONADO												
1	F1	1591	0,63361210	0	0,6336	8	12723	1	1591	7	11137	0,875000	0,056134	87	5	8,000
0	F0	321	0,12783751	100	0,3835	8	2568	3	963	5	1605	0,625000	0,008089	62		2,666
5	F5	151	0,06013540	101	0,1804	8	1208	3	453	5	755	0,625000	0,003805	62		2,666
9	F9	141	0,05615292	1100	0,2246	8	1128	4	564	4	564	0,500000	0,002842	50		2,000
6	F6	131	0,05217045	1101	0,2086	8	1048	4	524	4	524	0,500000	0,002641	50		2,000
2	F2	31	0,01234567	11100	0,0617	8	248	5	155	3	93	0,375000	0,000468	37		1,500
3	F3	31	0,01234567	11101	0,0617	8	248	5	155	3	93	0,375000	0,000468	37		1,500
4	F4	31	0,01234567	111100	0,0740	8	248	6	186	2	62	0,250000	0,000312	25		1,333
7	F7	31	0,01234567	111101	0,0740	8	248	6	186	2	62	0,250000	0,000312	25		1,333
8	F8	31	0,01234567	111110	0,0740	8	248	6	186	2	62	0,250000	0,000312	25		1,333
	C0	1	0,00039824	1111110000	0,0039	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
A	C1	1	0,00039824	1111110001	0,0039	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
B	C2	1	0,00039824	1111110010	0,0039	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
C	C3	1	0,00039824	1111110011	0,0039	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
D	C4	1	0,00039824	1111110100	0,0039	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
E	C5	1	0,00039824	1111110101	0,0039	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
F	C6	1	0,00039824	1111110110	0,0039	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
G	C7	1	0,00039824	1111110111	0,0039	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
H	C8	1	0,00039824	1111111000	0,0039	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
I	C9	1	0,00039824	1111111001	0,0039	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
	D0	1	0,00039824	1111111010	0,0039	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
J	D1	1	0,00039824	11111110110	0,0043	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
K	D2	1	0,00039824	11111110111	0,0043	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
L	D3	1	0,00039824	11111111000	0,0043	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
M	D4	1	0,00039824	11111111001	0,0043	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
N	D5	1	0,00039824	11111111010	0,0043	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
O	D6	1	0,00039824	11111111011	0,0043	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
P	D7	1	0,00039824	11111111100	0,0043	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
Q	D8	1	0,00039824	11111111101	0,0043	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
R	D9	1	0,00039824	11111111110	0,0043	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
	FL	1	0,00039824	11111111111	0,0043	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
		2511	0,99999977		2,0620		20033		5183		14905		0,075126			

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. ORIG. CAR	COMPR. ORIG. ARQ	COMPR. CAR	COMPR. ARQ	COMP. ABSOL. CAR	COMP. ABSOL. ARQ	COMP. RELATIVA CAR	COMP. RELATIVA ARQ	COM.PER CAR	COM.PER ARQ	RAIO COMP.
SEGMENTO NUMERO = 2			FORMATO = DECIMAL ZONADO												
5 F5	1501	0,98040496	0	0,9804 8	12008	1	1501	7	10507		0,875000	0,052958	87	5	8,000
CO	1	0,00065316	10000	0,0032 8	3	5	5	3	3		0,375000	0,000015	37		1,600
A C1	1	0,00065316	10001	0,0032 8	3	5	5	3	3		0,375000	0,000015	37		1,600
B C2	1	0,00065316	100100	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
C C3	1	0,00065316	100101	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
D C4	1	0,00065316	100110	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
E C5	1	0,00065316	100111	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
F C6	1	0,00065316	101000	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
G C7	1	0,00065316	101001	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
H C8	1	0,00065316	101010	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
I C9	1	0,00065316	101011	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
DO	1	0,00065316	101100	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
J D1	1	0,00065316	101101	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
K D2	1	0,00065316	101110	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
L D3	1	0,00065316	101111	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
M D4	1	0,00065316	110000	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
N D5	1	0,00065316	110001	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
O D6	1	0,00065316	110010	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
P D7	1	0,00065316	110011	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
Q D8	1	0,00065316	110100	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
R D9	1	0,00065316	110101	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
0 F0	1	0,00065316	110110	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
1 F1	1	0,00065316	110111	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
2 F2	1	0,00065316	111000	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
3 F3	1	0,00065316	111001	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
4 F4	1	0,00065316	111010	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
6 F6	1	0,00065316	111011	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
7 F7	1	0,00065316	111100	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
8 F8	1	0,00065316	111101	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
9 F9	1	0,00065316	111110	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
D FL	1	0,00065316	111111	0,0039 8	8	6	6	2	2		0,250000	0,000010	25		1,333
	1531	0,99999976		1,0960	12248		1579		10569			0,053271			

MAPA ESTATISTICO DOS ALFABETOS -:10/01/78 -:PAG -: 17

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. ORIG. CAR	ARQ	COMPR. COMPR. CAR	ARQ	COMP. CAR	ABSOL. ARQ	COMP. CAR	RELATIVA ARQ	COM.PER CAR	RAIO ARQ	COMP.
SEGMENTO NUMERO - 3			FORMATO - :ALFANUMERICO												
C C3	2801	0,97937062	0	0,9793	8	22408	1	2801	7	19607	0,875000	0,098825	87	9	8,000
40	1	0,00034965	100000	0,0020	8	8	6	6	2	2	0,250000	0,000010	25		1,333
4A	1	0,00034965	100001	0,0020	8	8	6	6	2	2	0,250000	0,000010	25		1,333
. 4B	1	0,00034965	100010	0,0020	8	8	6	6	2	2	0,250000	0,000010	25		1,333
(4D	1	0,00034965	100011	0,0020	8	8	6	6	2	2	0,250000	0,000010	25		1,333
+ 4E	1	0,00034965	100100	0,0020	8	8	6	6	2	2	0,250000	0,000010	25		1,333
| 4F	1	0,00034965	1001010	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
& 50	1	0,00034965	1001011	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
$ 5B	1	0,00034965	1001100	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
* 5C	1	0,00034965	1001101	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
) 5D	1	0,00034965	1001110	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
; 5E	1	0,00034965	1001111	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
:60	1	0,00034965	1010000	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
| 61	1	0,00034965	1010001	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
6A	1	0,00034965	1010010	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
, 6B	1	0,00034965	1010011	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
% 6C	1	0,00034965	1010100	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
6D	1	0,00034965	1010101	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
: 7A	1	0,00034965	1010110	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
# 7B	1	0,00034965	1010111	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
@ 7C	1	0,00034965	1011000	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
7D	1	0,00034965	1011001	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
= 7E	1	0,00034965	1011010	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
7F	1	0,00034965	1011011	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
A C1	1	0,00034965	1011100	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
B C2	1	0,00034965	1011101	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
D C4	1	0,00034965	1011110	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
E C5	1	0,00034965	1011111	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
F C6	1	0,00034965	1100000	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
G C7	1	0,00034965	1100001	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
H C8	1	0,00034965	1100010	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
I C9	1	0,00034965	1100011	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
J D1	1	0,00034965	1100100	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
K D2	1	0,00034965	1100101	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
L D3	1	0,00034965	1100110	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
M D4	1	0,00034965	1100111	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
N D5	1	0,00034965	1101000	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
O D6	1	0,00034965	1101001	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
P D7	1	0,00034965	1101010	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
Q D8	1	0,00034965	1101011	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
R D9	1	0,00034965	1101100	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
S E2	1	0,00034965	1101101	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
T E3	1	0,00034965	1101110	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
U E4	1	0,00034965	1101111	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
V E5	1	0,00034965	1110000	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
W E6	1	0,00034965	1110001	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
X E7	1	0,00034965	1110010	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142
Y E8	1	0,00034965	1110011	0,0024	8	8	7	7	1	1	0,125000	0,000005	12		1,142

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. ORIG. CAR	COMPR. ORIG. ARQ	COMPR. COMP. CAR	COMPR. COMP. ARQ	COMP. ABSOL. CAR	COMP. ABSOL. ARQ	COMP. RELATIVA CAR	COMP. RELATIVA ARQ	CON.PER CAR	RAID ARQ COMP.
SEGMENTO NUMERO - 3 FORMATO - ALFANUMERICO														
Z E9	1	0,00034965	1110100	0,0024	8	8	7	7	1	1	0,125000	0,000005	12	1,142
0 F0	1	0,00034965	1110101	0,0024	8	8	7	7	1	1	0,125000	0,000005	12	1,142
1 F1	1	0,00034965	1110110	0,0024	8	8	7	7	1	1	0,125000	0,000005	12	1,142
2 F2	1	0,00034965	1110111	0,0024	8	8	7	7	1	1	0,125000	0,000005	12	1,142
3 F3	1	0,00034965	1111000	0,0024	8	8	7	7	1	1	0,125000	0,000005	12	1,142
4 F4	1	0,00034965	1111001	0,0024	8	8	7	7	1	1	0,125000	0,000005	12	1,142
5 F5	1	0,00034965	1111010	0,0024	8	8	7	7	1	1	0,125000	0,000005	12	1,142
6 F6	1	0,00034965	1111011	0,0024	8	8	7	7	1	1	0,125000	0,000005	12	1,142
7 F7	1	0,00034965	1111100	0,0024	8	8	7	7	1	1	0,125000	0,000005	12	1,142
8 F8	1	0,00034965	1111101	0,0024	8	8	7	7	1	1	0,125000	0,000005	12	1,142
9 F9	1	0,00034965	1111110	0,0024	8	8	7	7	1	1	0,125000	0,000005	12	1,142
D FL	1	0,00034965	1111111	0,0024	8	8	7	7	1	1	0,125000	0,000005	12	1,142
	2860	0,99999997		1,1189		22880		3209		19671		0,099148		

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. ORIG. CAR	COMPR. ORIG. ARQ	COMPR. CAR	COMPR. ARQ	COMP. CAR	ABSOL. ARQ	COMP. RELATIVA CAR	COMP. RELATIVA ARQ	COM.PER CAR	RAIO ARQ	COMP.
SEGMENTO NUMERO - 4			FORMATO - ALFANUMERICO												
A C1	111	0,18196721	00	0,3639	8	333	2	222	6	666	0,750000	0,003356	75		4,000
B C2	111	0,18196721	01	0,3639	8	888	2	222	6	666	0,750000	0,003356	75		4,000
C C3	111	0,18196721	100	0,5459	8	833	3	333	5	555	0,625000	0,002797	62		2,666
D C4	111	0,18196721	101	0,5459	8	883	3	333	5	555	0,625000	0,002797	62		2,666
E C5	111	0,18196721	110	0,5459	8	888	3	333	5	555	0,625000	0,002797	62		2,666
40	1	0,00163934	11100000	0,0131	8	3	8	8		0					1,000
4A	1	0,00163934	11100001	0,0131	8	3	8	8		0					1,000
. 4B	1	0,00163934	11100010	0,0131	8	3	8	8		0					1,000
(4D	1	0,00163934	11100011	0,0131	8	8	8	8		0					1,000
+ 4E	1	0,00163934	11100100	0,0131	8	3	3	8		0					1,000
| 4F	1	0,00163934	11100101	0,0131	8	8	8	8		0					1,000
& 50	1	0,00163934	11100110	0,0131	8	8	8	8		0					1,000
$ 5B	1	0,00163934	11100111	0,0131	8	8	8	8		0					1,000
* 5C	1	0,00163934	11101000	0,0131	8	3	3	8		0					1,000
) 5D	1	0,00163934	111010010	0,0147	8	3	9	9	-1	-1	0,125000	0,000005	12		0,888
; 5E	1	0,00163934	111010011	0,0147	8	3	9	9	-1	-1	0,125000	0,000005	12		0,888
- 60	1	0,00163934	111010100	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
/ 61	1	0,00163934	111010101	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
6A	1	0,00163934	111010110	0,0147	8	3	9	9	-1	-1	0,125000	0,000005	12		0,888
, 6B	1	0,00163934	111010111	0,0147	8	3	9	9	1	1	0,125000	0,000005	12		0,888
= 6C	1	0,00163934	111011000	0,0147	8	3	9	9	-1	-1	0,125000	0,000005	12		0,888
6D	1	0,00163934	111011001	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
: 7A	1	0,00163934	111011010	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
# 7B	1	0,00163934	111011011	0,0147	8	3	9	9	-1	-1	0,125000	0,000005	12		0,888
@ 7C	1	0,00163934	111011100	0,0147	8	3	9	9	-1	1	0,125000	0,000005	12		0,888
7D	1	0,00163934	111011101	0,0147	8	3	9	9	-1	-1	0,125000	0,000005	12		0,888
= 7E	1	0,00163934	111011110	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
7F	1	0,00163934	111011111	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
F C6	1	0,00163934	111100000	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
G C7	1	0,00163934	111100001	0,0147	8	3	9	9	1	1	0,125000	0,000005	12		0,888
H C8	1	0,00163934	111100010	0,0147	8	3	9	9	-1	1	0,125000	0,000005	12		0,888
I C9	1	0,00163934	111100011	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
J D1	1	0,00163934	111100100	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
K D2	1	0,00163934	111100101	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
L D3	1	0,00163934	111100110	0,0147	8	8	9	9	-1	1	0,125000	0,000005	12		0,888
M D4	1	0,00163934	111100111	0,0147	8	3	9	9	-1	-1	0,125000	0,000005	12		0,888
N D5	1	0,00163934	111101000	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
O D6	1	0,00163934	111101001	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
P D7	1	0,00163934	111101010	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
Q D8	1	0,00163934	111101011	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
R D9	1	0,00163934	111101100	0,0147	8	3	9	9	1	1	0,125000	0,000005	12		0,888
S E2	1	0,00163934	111101101	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
T E3	1	0,00163934	111101110	0,0147	8	3	9	9	-1	-1	0,125000	0,000005	12		0,888
U E4	1	0,00163934	111101111	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
V E5	1	0,00163934	111110000	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
W E6	1	0,00163934	111110001	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
X E7	1	0,00163934	111110010	0,0147	8	3	9	9	-1	-1	0,125000	0,000005	12		0,888
Y E8	1	0,00163934	111110011	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. ORIG. CAR	COMPR. ORIG. ARQ	COMPR. COMPR. CAR	COMPR. COMPR. ARQ	COMP. ABSOL. CAR	COMP. ABSOL. ARQ	COMP. RELATIVA CAR	COMP. RELATIVA ARQ	COM.PER CAR	RAIO ARQ COMP.
SEGMENTO NUMERO - 4 FORMATO - ALFANUMERICO														
Z E9	1	0,00163934	111110100	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
0 F0	1	0,00163934	111110101	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
1 F1	1	0,00163934	111110110	0,0147	8	8	9	9	-1	1	0,125000	0,000005	12	0,888
2 F2	1	0,00163934	111110111	0,0147	8	8	9	9	-1	1	0,125000	0,000005	12	0,888
3 F3	1	0,00163934	111111000	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
4 F4	1	0,00163934	111111001	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
5 F5	1	0,00163934	111111010	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
6 F6	1	0,00163934	111111011	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
7 F7	1	0,00163934	111111100	0,0147	8	8	9	9	1	1	0,125000	0,000005	12	0,888
8 F8	1	0,00163934	111111101	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
9 F9	1	0,00163934	111111110	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
D FL	1	0,00163934	111111111	0,0147	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
	610	0,99999975		3,1595		4880		1929		2951		0,014873		

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. ORIG. CAR	ARQ	COMPR. COMPR. CAR	ARQ	COMP. ABSOL. CAR	ARQ	COMP. RELATIVA CAR	ARQ	COM.PER CAR	RAID ARQ	COMP.
SEGMENTO NUMERO - 5 FORMATO - ALFANUMERICO															
40	511	0,61566265 0		0,6156 8	4033	1	511	7	3577	0,875000	0,018029	87	1	8,000	
0 D6	21	0,02530120 10000		0,1265 8	163	5	105	3	63	0,375000	0,000317	37		1,500	
S E2	21	0,02530120 10001		0,1265 8	163	5	105	3	63	0,375000	0,000317	37		1,500	
A C1	11	0,01325301 10010		0,0662 8	88	5	55	3	33	0,375000	0,000166	37		1,500	
B C2	11	0,01325301 10011		0,0662 8	88	5	55	3	33	0,375000	0,000166	37		1,600	
C C3	11	0,01325301 101000		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
D C4	11	0,01325301 101001		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
E C5	11	0,01325301 101010		0,0795 8	88	5	66	2	22	0,250000	0,000110	25		1,333	
F C6	11	0,01325301 101011		0,0795 8	88	5	66	2	22	0,250000	0,000110	25		1,333	
G C7	11	0,01325301 101100		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
H C8	11	0,01325301 101101		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
I C9	11	0,01325301 101110		0,0795 8	88	5	66	2	22	0,250000	0,000110	25		1,333	
K D2	11	0,01325301 101111		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
L D3	11	0,01325301 110000		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
M D4	11	0,01325301 110001		0,0795 8	88	5	66	2	22	0,250000	0,000110	25		1,333	
N D5	11	0,01325301 110010		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
P D7	11	0,01325301 110011		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
Q D8	11	0,01325301 110100		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
R D9	11	0,01325301 110101		0,0795 8	88	5	66	2	22	0,250000	0,000110	25		1,333	
T E3	11	0,01325301 110110		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
V E5	11	0,01325301 110111		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
W E6	11	0,01325301 111000		0,0795 8	88	5	66	2	22	0,250000	0,000110	25		1,333	
X E7	11	0,01325301 111001		0,0795 8	88	5	66	2	22	0,250000	0,000110	25		1,333	
Y E8	11	0,01325301 111010		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
Z E9	11	0,01325301 111011		0,0795 8	88	6	66	2	22	0,250000	0,000110	25		1,333	
4A	1	0,00120481 111100000		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
. 4B	1	0,00120481 111100001		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
(4D	1	0,00120481 111100010		0,0108 8	8	9	9	-1	1	0,125000	0,000005	12		0,888	
+ 4E	1	0,00120481 111100011		0,0108 8	8	9	9	1	1	0,125000	0,000005	12		0,888	
| 4F	1	0,00120481 111100100		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
& 50	1	0,00120481 111100101		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
$ 5B	1	0,00120481 111100110		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
* 5C	1	0,00120481 111100111		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
) 5D	1	0,00120481 111101000		0,0108 8	8	9	9	-1	1	0,125000	0,000005	12		0,888	
; 5E	1	0,00120481 111101001		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
- 60	1	0,00120481 111101010		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
| 61	1	0,00120481 111101011		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
6A	1	0,00120481 111101100		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
, 6B	1	0,00120481 111101101		0,0108 8	8	9	9	-1	1	0,125000	0,000005	12		0,888	
= 6C	1	0,00120481 111101110		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
6D	1	0,00120481 111101111		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
: 7A	1	0,00120481 111110000		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
# 7B	1	0,00120481 111110001		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
@ 7C	1	0,00120481 111110010		0,0108 8	8	9	9	1	1	0,125000	0,000005	12		0,888	
7D	1	0,00120481 111110011		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
= 7E	1	0,00120481 111110100		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
7F	1	0,00120481 111110101		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	
J D1	1	0,00120481 111110110		0,0108 8	8	9	9	-1	-1	0,125000	0,000005	12		0,888	

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. ORIG. CAR	COMPR. ORIG. ARQ	COMPR. COMPR. CAR	COMPR. COMPR. ARQ	COMP. ABSOL. CAR	COMP. ABSOL. ARQ	COMP. RELATIVA CAR	COMP. RELATIVA ARQ	COM.PER CAR	RAID ARQ COMP.
SEGMENTO NUMERO - 5			FORMATO - ALFANUMERICO											
U E4	1	0,00120481	111110111	0,0108	8	8	9	9	1	1	0,125000	0,000005	12	0,888
0 F0	1	0,00120481	111111000	0,0108	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
1 F1	1	0,00120481	111111001	0,0108	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
2 F2	1	0,00120481	111111010	0,0108	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
3 F3	1	0,00120481	111111011	0,0108	8	8	9	9	-1	-1	0,125000	0,000005	12	0,888
4 F4	1	0,00120481	111111100	0,0108	8	8	9	9	1	-1	0,125000	0,000005	12	0,888
5 F5	1	0,00120481	1111111010	0,0120	8	8	10	10	-2	-2	0,250000	0,000010	25	0,800
6 F6	1	0,00120481	1111111011	0,0120	8	8	10	10	-2	-2	0,250000	0,000010	25	0,800
7 F7	1	0,00120481	1111111100	0,0120	8	8	10	10	-2	-2	0,250000	0,000010	25	0,800
8 F8	1	0,00120481	1111111101	0,0120	8	8	10	10	-2	-2	0,250000	0,000010	25	0,800
9 F9	1	0,00120481	1111111110	0,0120	8	8	10	10	-2	-2	0,250000	0,000010	25	0,800
D FL	1	0,00120481	1111111111	0,0120	8	8	10	10	2	2	0,250000	0,000010	25	0,800
	830	0,99999962		2,9762		6640		2472		4168		0,021008		

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. ORIG. CAR	COMPR. ORIG. ARQ	COMPR. COMPR. CAR	COMPR. COMPR. ARQ	COMP. ABSOL. CAR	COMP. ABSOL. ARQ	COMP. RELATIVA CAR	COMP. RELATIVA ARQ	COM.PER CAR	RAIO ARQ	RAIO COMP.
SEGMENTO NUMERO - 6			FORMATO DECIMAL ZONADO												
1 F1	281	0,58419958	0	0,5841	8	2248	1	281	7	1967	0,875000	0,009914	87		8,000
6 F6	71	0,14760914	100	0,4428	8	568	3	213	5	355	0,625000	0,001789	62		2,666
5 F5	51	0,10602910	101	0,3180	8	408	3	153	5	255	0,625000	0,001285	62		2,666
0 F0	31	0,06444906	110	0,1933	8	248	3	93	5	155	0,625000	0,000781	62		2,666
2 F2	21	0,04365904	1110	0,1745	8	168	4	84	4	84	0,500000	0,000423	50		2,000
C0	1	0,00207900	11110000	0,0166	8	8	8	8		0					1,000
A C1	1	0,00207900	11110001	0,0166	8	8	8	8		0					1,000
8 C2	1	0,00207900	11110010	0,0166	8	8	8	8		0					1,000
C C3	1	0,00207900	11110011	0,0166	8	8	8	8		0					1,000
D C4	1	0,00207900	11110100	0,0166	8	8	8	8		0					1,000
E C5	1	0,00207900	11110101	0,0166	8	8	8	8		0					1,000
F C6	1	0,00207900	111101100	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
G C7	1	0,00207900	111101101	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
H C8	1	0,00207900	111101110	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
I C9	1	0,00207900	111101111	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
D0	1	0,00207900	111110000	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
J D1	1	0,00207900	111110001	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
K D2	1	0,00207900	111110010	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
L D3	1	0,00207900	111110011	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
M D4	1	0,00207900	111110100	0,0187	8	8	9	9	1	-1	0,125000	0,000005	12		0,888
N D5	1	0,00207900	111110101	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
O D6	1	0,00207900	111110110	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
P D7	1	0,00207900	111110111	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
Q D8	1	0,00207900	111111000	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
R D9	1	0,00207900	111111001	0,0187	8	8	9	9	1	-1	0,125000	0,000005	12		0,888
3 F3	1	0,00207900	111111010	0,0187	8	8	9	9	-1	1	0,125000	0,000005	12		0,888
4 F4	1	0,00207900	111111011	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
7 F7	1	0,00207900	111111100	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
8 F8	1	0,00207900	111111101	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
9 F9	1	0,00207900	111111110	0,0187	8	8	9	9	-1	-1	0,125000	0,000005	12		0,888
D FL	1	0,00207900	111111111	0,0187	8	8	9	9	1	1	0,125000	0,000005	12		0,888
	481	0,99999992		2,1864		3848		1052		2796		0,014092			

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR.ORIG. CAR	COMPR.ORIG. ARQ	COMPR. COMP. CAR	COMPR. COMP. ARQ	ABSOL. COMP. CAR	ABSOL. COMP. ARQ	RELATIVA COMP. CAR	RELATIVA COMP. ARQ	COM.PER RAIO CAR	COM.PER RAIO ARQ	RAIO COMP.
SEGMENTO NUMERO - 7			FORMATO DECIMAL ZONADO												
1 F1	151	0,83425414	0	0,8342	8	1208	1	151	7	1057	0,875000	0,005327	87		8,000
CO	1	0,00552486	10000	0,0276	8	8	5	5	3	3	0,375000	0,000015	37		1,600
A C1	1	0,00552486	10001	0,0275	8	8	5	5	3	3	0,375000	0,000015	37		1,600
B C2	1	0,00552486	100100	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
C C3	1	0,00552486	100101	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
D C4	1	0,00552486	100110	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
E C5	1	0,00552486	100111	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
F C6	1	0,00552486	101000	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
G C7	1	0,00552486	101001	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
H C8	1	0,00552486	101010	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
I C9	1	0,00552486	101011	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
DO	1	0,00552486	101100	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
J D1	1	0,00552486	101101	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
K D2	1	0,00552486	101110	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
L D3	1	0,00552486	101111	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
M D4	1	0,00552486	110000	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
N D5	1	0,00552486	110001	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
O D6	1	0,00552486	110010	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
P D7	1	0,00552486	110011	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
Q D8	1	0,00552486	110100	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
R D9	1	0,00552486	110101	0,0331	8	8	6	5	2	2	0,250000	0,000010	25		1,333
0 F0	1	0,00552486	110110	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
2 F2	1	0,00552486	110111	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
3 F3	1	0,00552486	111000	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
4 F4	1	0,00552486	111001	0,0331	8	8	6	5	2	2	0,250000	0,000010	25		1,333
5 F5	1	0,00552486	111010	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
6 F6	1	0,00552486	111011	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
7 F7	1	0,00552486	111100	0,0331	8	8	6	5	2	2	0,250000	0,000010	25		1,333
8 F8	1	0,00552486	111101	0,0331	8	8	6	5	2	2	0,250000	0,000010	25		1,333
9 F9	1	0,00552486	111110	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
D FL	1	0,00552486	111111	0,0331	8	8	6	6	2	2	0,250000	0,000010	25		1,333
	181	0,99999994		1,8162		1443		329		1119		0,005640			

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. ORIG. CAR	COMPR. ORIG. ARQ	COMPR. CAR	COMPR. ARQ	COMP. CAR	ABSOL. ARQ	COMP. CAR	RELATIVA ARQ	COM-PER CAR	COM-PER ARQ	RAIO COMP.
SEGMENTO NUMERO - 8			FORMATO - ALFANUMERICO												
40	8371	0,76727772	0	0,7672	8	66968	1	8371	7	58597	0,875000	0,295347	87	29	8,000
0 F0	2241	0,20540788	10	0,4108	8	17923	2	4482	6	13446	0,750000	0,067772	75	6	4,000
9 F9	61	0,00559120	11000	0,0279	8	488	5	305	3	183	0,375000	0,000922	37		1,500
1 F1	31	0,00284142	11001	0,0142	8	248	5	155	3	93	0,375000	0,000468	37		1,600
3 F3	31	0,00284142	11010	0,0142	8	248	5	155	3	93	0,375000	0,000468	37		1,600
4 F4	31	0,00284142	11011	0,0142	8	243	5	155	3	93	0,375000	0,000468	37		1,500
6 F6	31	0,00284142	11100	0,0142	8	243	5	155	3	93	0,375000	0,000468	37		1,500
7 F7	31	0,00284142	11101	0,0142	8	248	5	155	3	93	0,375000	0,000468	37		1,500
2 F2	11	0,00100824	1111000	0,0070	8	88	7	77	1	11	0,125000	0,000055	12		1,142
5 F5	11	0,00100824	1111001	0,0070	8	88	7	77	1	11	0,125000	0,000055	12		1,142
8 F8	11	0,00100824	1111010	0,0070	8	88	7	77	1	11	0,125000	0,000055	12		1,142
4A	1	0,00009165	1111011000	0,0009	8	8	10	10	2	-2	0,250000	0,000010	25		0,800
. 4B	1	0,00009165	1111011001	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
(4D	1	0,00009165	1111011010	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
+ 4E	1	0,00009165	1111011011	0,0009	8	8	10	10	2	2	0,250000	0,000010	25		0,800
| 4F	1	0,00009165	1111011100	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
& 50	1	0,00009165	1111011101	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
$ 5B	1	0,00009165	1111011110	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
* 5C	1	0,00009165	1111011111	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
) 5D	1	0,00009165	1111100000	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
; 5E	1	0,00009165	1111100001	0,0009	8	8	10	10	2	2	0,250000	0,000010	25		0,800
60	1	0,00009165	1111100010	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
| 61	1	0,00009165	1111100011	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
6A	1	0,00009165	1111100100	0,0009	8	8	10	10	2	2	0,250000	0,000010	25		0,800
, 6B	1	0,00009165	1111100101	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
% 6C	1	0,00009165	1111100110	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
6D	1	0,00009165	1111100111	0,0009	8	8	10	10	2	2	0,250000	0,000010	25		0,800
: 7A	1	0,00009165	1111101000	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
# 7B	1	0,00009165	1111101001	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
@ 7C	1	0,00009165	1111101010	0,0009	8	8	10	10	2	2	0,250000	0,000010	25		0,800
7D	1	0,00009165	1111101011	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
= 7E	1	0,00009165	1111101100	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
7F	1	0,00009165	1111101101	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
A C1	1	0,00009165	1111101110	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
B C2	1	0,00009165	1111101111	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
C C3	1	0,00009165	1111110000	0,0009	8	8	10	10	-2	2	0,250000	0,000010	25		0,800
D C4	1	0,00009165	1111110001	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
E C5	1	0,00009165	1111110010	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
F C6	1	0,00009165	1111110011	0,0009	8	8	10	10	-2	2	0,250000	0,000010	25		0,800
G C7	1	0,00009165	1111110100	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
H C8	1	0,00009165	1111110101	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
I C9	1	0,00009165	1111110110	0,0009	8	8	10	10	-2	-2	0,250000	0,000010	25		0,800
J D1	1	0,00009165	11111101110	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
K D2	1	0,00009165	11111101111	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
L D3	1	0,00009165	11111110000	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
M D4	1	0,00009165	11111110001	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
N D5	1	0,00009165	11111110010	0,0010	8	8	11	11	-3	3	0,375000	0,000015	37		0,727
O D6	1	0,00009165	11111110011	0,0010	8	8	11	11	-3	3	0,375000	0,000015	37		0,727

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. ORIG. CAR	COMPR. ORIG. ARQ	COMPR. CAR	COMPR. ARQ	COMP. ABSOL. CAR	COMP. ABSOL. ARQ	COMP. RELATIVA CAR	COMP. RELATIVA ARQ	COM.PER CAR	COM.PER ARQ	RAIO COMP.
SEGMENTO NUMERO 8			FORMATO - ALFANUMERICO												
P D7	1	0,00009165	11111110100	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
Q D8	1	0,00009165	11111110101	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
R D9	1	0,00009165	11111110110	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
S E2	1	0,00009165	11111110111	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
T E3	1	0,00009165	11111111000	0,0010	8	8	11	11	3	-3	0,375000	0,000015	37		0,727
U E4	1	0,00009165	11111111001	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
V E5	1	0,00009165	11111111010	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
W E6	1	0,00009165	11111111011	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
X E7	1	0,00009165	11111111100	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
Y E8	1	0,00009165	11111111101	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
Z E9	1	0,00009165	11111111110	0,0010	8	8	11	11	-3	-3	0,375000	0,000015	37		0,727
D FL	1	0,00009165	11111111111	0,0010	8	8	11	11	3	3	0,375000	0,000015	37		0,727
	10910	0,99999947		1,3438		87280		14672		72608		0,365967			

Já foi visto anteriormente que o sistema em desenvolvimento acopla 2 técnicas de compressão, ou seja, ao se analisar cada segmento do registro eliminamos pelo método de Compressão por Segmentação aqueles sem conteúdo, aplicando para os demais o código HSF.

O Mapa Estatístico dos Alfabetos apresenta para cada segmento de acordo com seu formato a distribuição estatística, o código HSF gerado e a compressão obtida para cada caráter do alfabeto correspondente ao formato do segmento.

Para o nosso arquivo de testes,(para o registro foram definidos 8 segmentos), podemos observar que existem exatamente 8 relatórios, onde por motivos de construção do código HSF, cada um apresenta os caracteres do alfabeto correspondente ao segmento organizado em ordem decrescente de frequência de ocorrência.

Assim, através destes relatórios, o usuário pode avaliar rigorosamente a compressão final obtida para cada segmento no arquivo pela aplicação do código HSF e determinar se os resultados obtidos com os segmentos projetados são satisfatórios.

Componentes do relatório:

b.1) Identificação e Formato do Segmento

Uma vez que para cada segmento será gerado 1 relatório e que os caracteres possíveis de serem encontrados neste segmento são determinados pelo formato, torna-se conveniente, para facilitar a análise, ressaltar estas 2 informações através da mensagem:

"SEGMENTO NÚMERO - XXX FORMATO - XXX$\underbrace{\quad\quad}_{16}$X"

onde apresentamos a identificação e o formato do segmento atribuídos pelo usuário.

Por exemplo:

SEGMENTO NUMERO - 2 FORMATO - DECIMAL ZONADO

b.2) Alfabeto (ALF.)

Consiste na representação caráter e hexadecimal do alfabeto correspondente ao segmento.

Por exemplo, para o segmento numero 2, temos a representação caráter e hexadecimal do alfabeto, correspondente ao formato Decimal Zonado.

b.3) Frequência de Ocorrência dos Caracteres (FREQ.)

Indica o número de vezes que cada caráter do alfabeto, para um dado segmento, foi observado na amostra, apresentando ao final o número total de caracteres observados para o segmento na amostra. Lembramos que, para efeito de obtenção do código HSF, consideramos para todos os elementos do alfabeto uma frequência inicial de 1.

Por exemplo, para o segmento número 2, o caráter 5 foi observado 1501 vezes para 300 segmentos analisados.

b.4) Probabilidade de Ocorrência dos Caracteres (PROB.)

Calculada a partir da frequência de ocorrência dos caracteres para um determinado segmento, a probabilidade de ocorrência nos mostra quais as possibilidades de se encontrar um dado caráter no segmento em análise.

O caráter 5 no segmento 2 por exemplo, tem a probabilidade de ocorrência de 0,98040496, ou seja, a cada 100 caracteres analisados é provável que 98 sejam o número 5 (98%).

b.5) Código HSF gerado (GRUPO)

A partir da probabilidade de ocorrência de cada caráter do alfabeto correspondente a cada segmento do registro e através da utilização do código HSF determinamos a configuração binária (grupo) que irá representá-lo no arquivo compresso.

O caráter 5 do segmento 2 por exemplo, tem por grupo a configuração binária 'Ø'.

b.6) Custo do Código (CUSTO)

Aqui para cada caráter do alfabeto relativo a cada segmento do registro apresentamos o custo, que é acumulado para impressão no final deste relatório. Este valor acumulado indica o custo total do código, ou seja, quantos bits, em média, serão necessários para representar cada caráter do alfabeto em questão.

Para o caráter 5 do segmento 2 por exemplo, o custo é de 0,98040496 x 1 ◻ 0,9804 bits/caráter e o custo do código é 1,096 bits/caráter.

b.7) Comprimento Original (COMP. ORIG.)

Sob este título, apresentamos expresso em bits, para cada caráter do alfabeto relativo a cada segmento, o comprimento inicial, que dependendo do formato, deverá ser de 4 ou 8 bits (CAR.). São apresentadas ainda as correspondentes participações para a composição final da amostra analisada (ARQ.), a qual é o produto da frequência pelo comprimento inicial do caráter. Estas participações para cada segmento são acumuladas caráter a caráter para impressão ao final deste relatório e expressam o tamanho original do segmento na amostra.

Para o nosso arquivo de testes, temos:

- Comprimento original do caráter 5 no segmento 2 - 8 bits
- Comprimento original do caráter 5 na amostra - 8 x 1501 = 12008 bits (frequência acrescida de 1)
- Comprimento original do segmento 2 na amostra - 12248 bits (frequências acrescidas de 1)

b.8) Comprimento Compresso (COMPR. COMPR.)

Sob este título, apresentamos expresso em

bits, o comprimento compresso (CAR.) para cada cárater do alfabeto relativo a cada segmento. Este comprimento é o número de bits do grupo correspondente ao caráter. É apresentada também a contribuição de cada caráter na forma compressa (ARQ.) para a constituição final da amostra. A contribuição é o produto da frequência do caráter pelo seu comprimento compresso.

Do mesmo modo que no item anterior, esta contribuição é acumulada a cada caráter para impressão ao final deste relatório e expressam o tamanho compresso do segmento na amostra.

Para o nosso arquivo de testes, temos:

- Comprimento compresso do caráter 5 no segmento 2 - 1 bit
- Comprimento compresso do caráter 5 na amostra - 1 x 1501 = 1501 bits
- Comprimento compresso do segmento 2 na amostra - 1679 bits.

b.9) Compressão Absoluta (COMP. ABSOL.)

Para cada caráter do alfabeto relativo à cada segmento apresentamos, expresso em bits, o valor resultante do uso da configuração compressa, em substituição à configuração original, tanto no que se refere ao próprio caráter (CAR.) como do caráter no segmento (ARQ.).

Este valor é obtido caráter a caráter para cada segmento e acumulado para impressão ao final do relatório.

Observe-se que este valor poderá também ser negativo caso a nova configuração de bits atribuída ao caráter, seja maior em tamanho.

Para o nosso arquivo de testes, temos:

- Compressão absoluta do cárater 5 no segmento 2 - 8 - 1 = 7 bits

- Compressão absoluta do caráter 5 na amostra -
  12008 - 1501 = 10507 bits

- Compressão absoluta do segmento 2 na amostra -
  12248 - 1679 = 10569 bits

b.10) Compressão Relativa (COMP. RELATIVA)

Para cada caráter do alfabeto relativo à cada segmento apresentamos a compressão relativa ao próprio caráter (CAR) e a compressão relativa à amostra (ARQ), sendo esta última acumuladas para impressão ao final deste relatório (Impresso somente se valor $\neq 0$).

Para o caráter 5 do segmento 2, temos:

- Compressão relativa ao caráter

$$\frac{7}{8} = \frac{10507}{12008} = 0,875$$

- Compressão relativa ao arquivo

$$\frac{10507}{310x80x8} = 0,052958$$

b.11) Compressão Percentual (COM. PER)

Neste ponto são apresentadas em percentagem, as compressões relativas obtidas no item anterior (Imprime somente a parte inteira e se o valor $\neq 0$).

Para o caráter 5 do segmento 2, temos:

- Compressão percentual do carater
  0,875 x 100 = 87%

- Compressão percentual do caráter no arquivo
  0,052958 x 100 = 5%

b.12) Raio de Compressão (RAIO COMP.)

Aqui registramos para cada caráter do alfabeto relativo a cada segmento quantas vezes o comprimento original é superior ao comprimento compresso do caráter ou do caráter na amostra.

Para o caráter 5 do segmento 2, temos:

- Raio de compressão do caráter

$$\frac{8}{7} = 8,000$$

- Raio de Compressão do caráter na amostra

$$\frac{12008}{1501} = 8,000$$

c) Arquivo de Códigos

Gerado neste programa e com base em informações fornecidas pelo usuário o Arquivo de Códigos se constitui numa peça fundamental do sistema pois é por seu intermédio que as rotinas de compressão/descompressão codificam e decodificam informações.

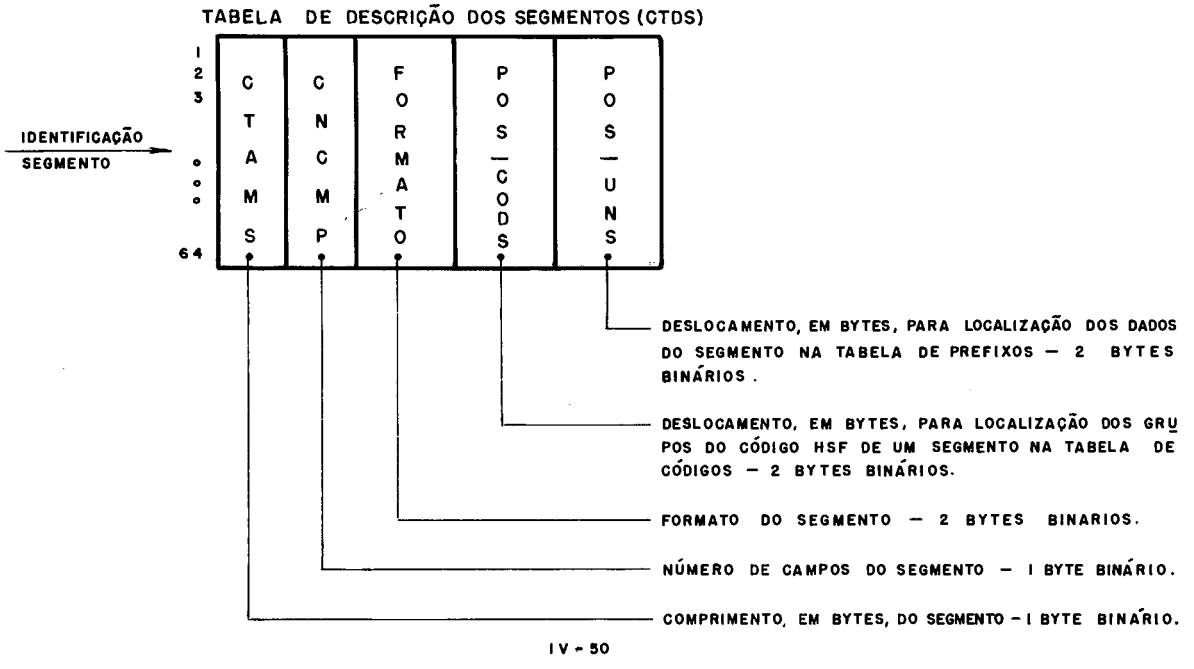
De nome interno FITA e endereço lógico - SYSØ11 o Arquivo de Códigos possui registros lógicos de 1024 bytes de comprimento não blocados e deve utilizar o mesmo meio-físico do Arquivo da Aplicação. Assim, uma vez que o nosso arquivo se encontra armazenado em fita, a versão atual deste programa gera o Arquivo de Códigos também em fita. (Veja o item de MODIFICAÇÕES - 5.6 para outros suportes de armazenamento).

A seguir, apresentaremos os dados componentes do arquivo:

c.1) Tabela de Descrição dos Segmentos (CTDS)

Esta tabela reune, para as rotinas de compressão e descompressão, todas as informações necessárias à composição e processamento de cada segmento do registro.

Projetada para um máximo de 64 segmentos e apresentando as suas entradas dispostas na mesma sequência dos segmentos no registro, esta tabela reune em cada entrada os dados a seguir.

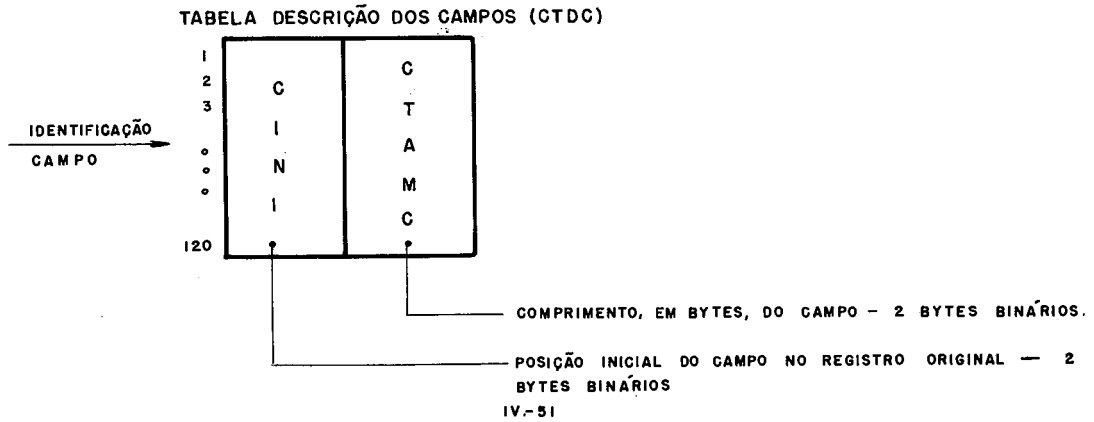


IV - 50

c.2) Tabela de Descrição dos Campos (CTDC)

Esta tabela reune, para as rotinas de compressão e descompressão, todos os dados necessários à seleção dos campos relativos a cada segmento.

Para um máximo de 120 campos e apresentando as suas entradas dispostas na mesma sequência dos campos no registro segmentado, esta Tabela reune, em cada entrada, os dados abaixo.



IV-51

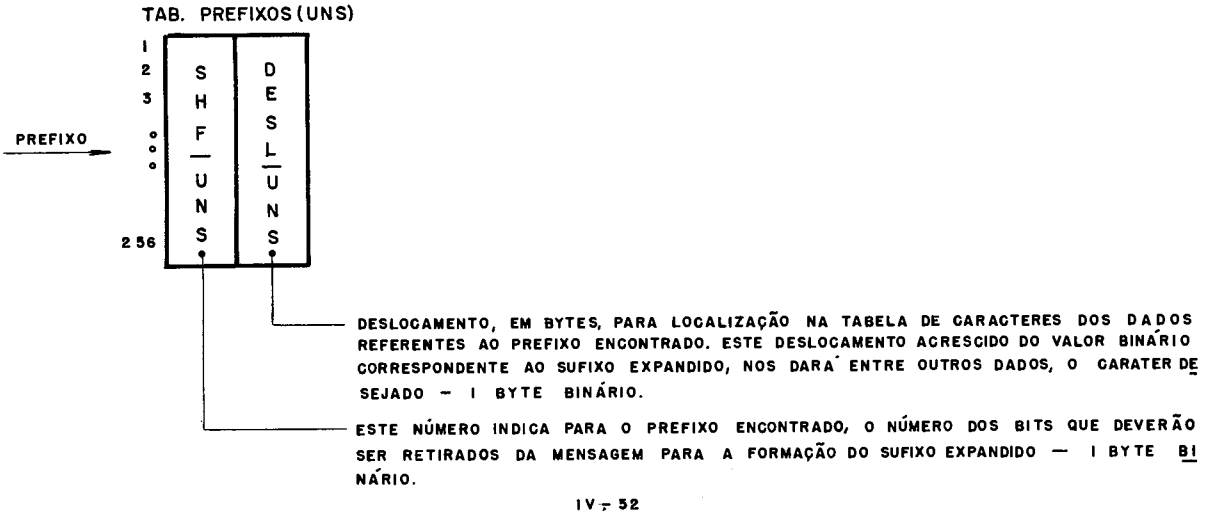
c.3) Padrão de Ausência (PADRAO)

Para um registro máximo de 512 bytes, o Padrão de Ausência reflete o registro segmentado com todos os campos componentes de cada segmento, se apresentando sem conteúdo.

c.4) Tabela de Prefixos (TABUNS)

Tabela indexada através do prefixo do grupo em decodificação e utilizada pela rotina de descompressão, tem a finalidade de auxiliar a decodificação de mensagens no código HSF. (Este aspecto já foi abordado quando da apresentação das tabelas desenvolvidas para este código).

Para um máximo de 256 entradas, a Tabela de Prefixos reúne para cada uma delas os dados abaixo.



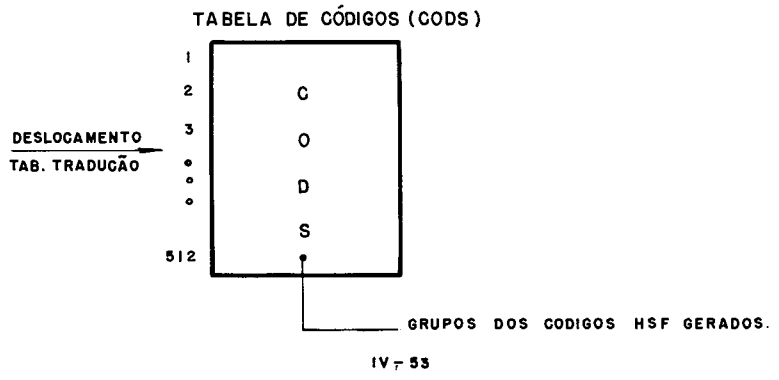
IV - 52

Observamos que para cada segmento poderemos utilizar uma ou mais entradas desta tabela.

c.5) Tabela de Códigos (CODS)

Tabela utilizada pela rotina de compressão cuja finalidade é fornecer, para todo e qualquer caráter de qualquer segmento, o grupo do código HSF correspondente.

Para a seleção do grupo desejado, utilizamos uma tabela fixa residente na rotina de compressão, a qual (conforme vimos anteriormente), por meio do caráter obtido, fornece o índice de entrada para esta tabela, que está projetada para um máximo de 512 grupos.

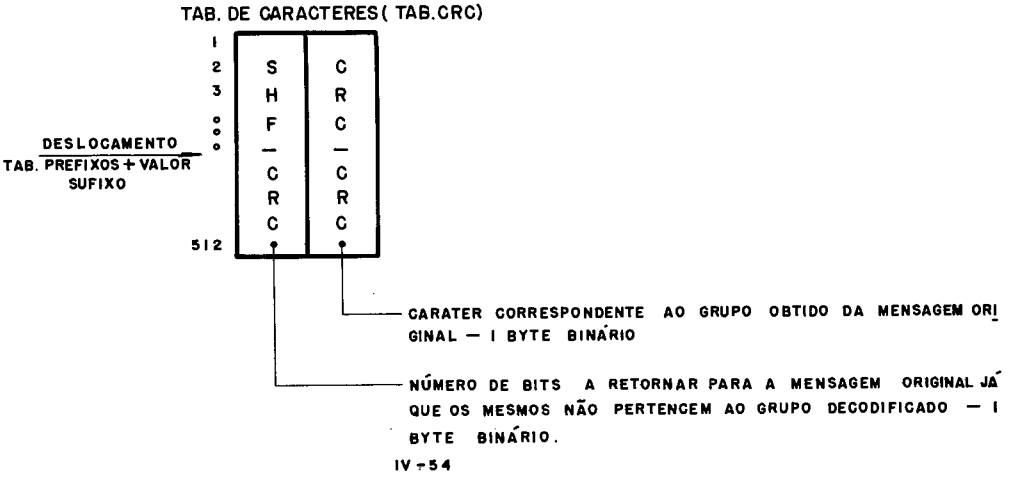


IV-53

c.6) Tabela de Caracteres (TAB-CRC)

Utilizada pela rotina de descompressão, e indexada por meio de dados obtidos da Tabela de Prefixos esta tabela fornece o caráter correspondente ao grupo decodificado, para todo e qualquer código HSF gerado.

Para um máximo de 512 entradas esta tabela reúne, em cada entrada, os dados abaixo.



IV-54

É interessante observar que para cada segmento utilizaremos tantas entradas quantas forem as necessárias à rotina de descompressão para recuperação de elementos do alfabeto correspondente.

```
* * * * DEVICE 281  SYS018,  MODE IS  9 TRACK  1600 BPI        BLOCK    1   DATA  1024                                   * * * *
                                     TABELA DE DESCRIÇÃO DOS SEGMENTOS
                                   | SEG.1  | SEG.2  | SEG.3  | SEG.4  | SEG.5  | SEG.6  | SEG.7  | SEG.8  |F|
REC      1   DATA  1024   CHAR     |        |   9    |   8    |   Y    |   Q    |   H -  |        |        |M|
                          ZONE      0000000000000701 00000F02 00000E03 00000D04 00000C05 00000407 20000C08 00444444 0000000000000000000000000000
                          NUMR      7220000041200C08 91000806 41001806 6100280A 41203800 41204404 25004002 00000000 0000000000000000000000000000
                                    1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                  101...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                  201...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                  301...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                  401...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....
                                                 TABELA DE DESCRIÇÃO DOS CAMPOS
                          CHAR                   |CMP 1|CMP 2|CMP 3|CMP 4|CMP5|CMP 6|CMP 7|CMP 8|CMP 9|CMP10|CMP 11|CMP12|CMP13| FIM |
                          ZONE      0000000000000 0000 0200 0200 0300 0000 4000 1000 1300 0000 0000 0100 3000 4000 0000 000000000000000000000000000000000
                          NUMR      0000000000000 0302 0804 0D04 0209 0604 0106 0404 0904 0002 0B03 0E09 0C04 0807 0000 000000000000000000000000000000000
                                  501...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                  601...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                  701...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                  801...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR      0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                  901...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE      000000000000000000000000
                          NUMR      000000000000000000000000
                                 1001...5...10...15...20...2
```

```
* * * * DEVICE  281  SYS018,  MODE IS  9 TRACK  1600 BPI      BLOCK  2  DATA  1024                                   * * * *
                                     PADRÃO DE AUSENCIA
                               SEG.1   SEG.2      SEG.3     SEG.4   SEG.5    SEG.6  SEG.7    SEG.8
REC      2    DATA   1024   CHAR  000000000000000                      000000000000                                 00  (
                            ZONE  FFFFFFFFFFFFFF44444444444444444444444FFFFFFFFF4444444444444444444444444444444FC43004FFB44009141181
                            NUMR  000000000000000000000000000000000000000000000000000000000000000000000000000000013017000100FB38490
                                  1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                            CHAR    0   0Q        2        0 N  R     03B  -   00    2    1       ;       2   1         Q      Q'K  -  P
                            ZONE  009F3048F0410051F641DD4DFEDODDE042FFFFD6D94FF844DD114BF0111545001151F64DF6141141DD1141D0004D1D6D80009
                            NUMR  0110007009110150282008700E50090170088F000070008008347014B1B53E01A55028701385B13003B40008602F00007F00
                               101...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                            CHAR               1+K          1      1  02*      8      K  -    00          &  01         +    & Q       1  -1  8
                            ZONE  D990D990D91448F4D0DAD9FFD9DA44F45FF50E9EDDFFD9D3D0D6DB4FF81444DD45014FF15EDD14440045001511 48FD46FFFF
                            NUMR  00210E2C0F24701E21000EC10000601480 2C5F8F0C3F00002F000070008430081 00E7014800CB43E01800884057010501C8F
                               201...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                            CHAR   -  8        N  R     1  8     -P  - -K   R    02*       8        01  K  R     02-          2+P  - - -10 02
                            ZONE  D6D7FFD7D9D0D0E044FCFFD9D6D0D6D6D0DDE05FF50E9EDDFFD9DB4FFDDODDE05FF60E9EDD104BF4D1D6D646FF4FF3114100
                            NUMR  00008F00005309017 01C8F00007F0000200901802C5F8F0C8F0000701C2C0901802 05F8F0C20702E7F00005010702AB13E01
                               301...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                            CHAR      Q      "P          2    2        6                   "         2        6     6 02*        8  -   02 3  -   W
                            ZONE  42DD1190D7D0D7D751F64DF191D70F41004 1DD81004307139 0D041F390300F91300F5F=50E9EDDFFD6DB4FF5FFD6D99EDD4F
                            NUMR  8008A22C0F7E000050287024600076110120 0AA0011D0FB1110A7024610076600076802C5F8F0C8F000070248F00008600 7E
                               401...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....
                                                              T                A          B            E            L          A
                            CHAR                          SEG. 1                    SEG. 2            SEG. 3   : =          SEG. 4    :           SEG. 5
                            ZONE  0000E800E300000000000000001010203030404000002030303030000040607070700000000000406070808090000020303 05
                            NUMR  0201C001B9000000121 61A1E02343424 1C000400423222 1A0E050052423222 1A0E1810140 85A4A3A2A1206120042322 24A3A
                               501...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....
                                D  E          P          R      E        F        I          X          O            S
                            CHAR         : =         SEG. 6                  ;  SEG. 7                  SEG. 8  &                 0            00   0        )  0
                            ZONE  06070707010000000000020304040500000203030 30A000000000010507090A0A0A90E047FA91D49FE04FFA9CF39ED01599FB
                            NUMR  2A221A0E24001206084A3A2A120 62E004232222 1A0E2000022 41C50404030201 80C50007002600 82F00700 8E0A0C0C8D8F0A
                               601...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                            CHAR              6    0    0                    B     C0BF3000LERFITA         M20/12/7616.50.33        0
                            ZONE  91D4010F00004F00F100F100400048004C004DCDCCFFFD0D0CEC4004DFF6FF6FFFF4FF4FF0000F00000000000000000002000
                            NUMR  1008797F7002960100100026802380262029C36263003359693100254201121761685 09330000000000000000000011940 0
                               701...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                            CHAR                                          1                 SYS010       >                                  1
                            ZONE  0000000000000000000000C08000D000F70000104812EEEFFF44058000200000000000000000521400FA0020001B001B4EE0202000
                            NUMR  000000000000000000000020001A011000 0002A048282010002E0000100000000000707080042118009402A002A01001949370
                               801...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                            CHAR       (    (
                            ZONE  0000420047000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                            NUMR  0002D802D000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                               901...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                            CHAR
                            ZONE  000000000000000000000000
                            NUMR  000000000000000000000000
                              1001...5...10...15...20...2
```

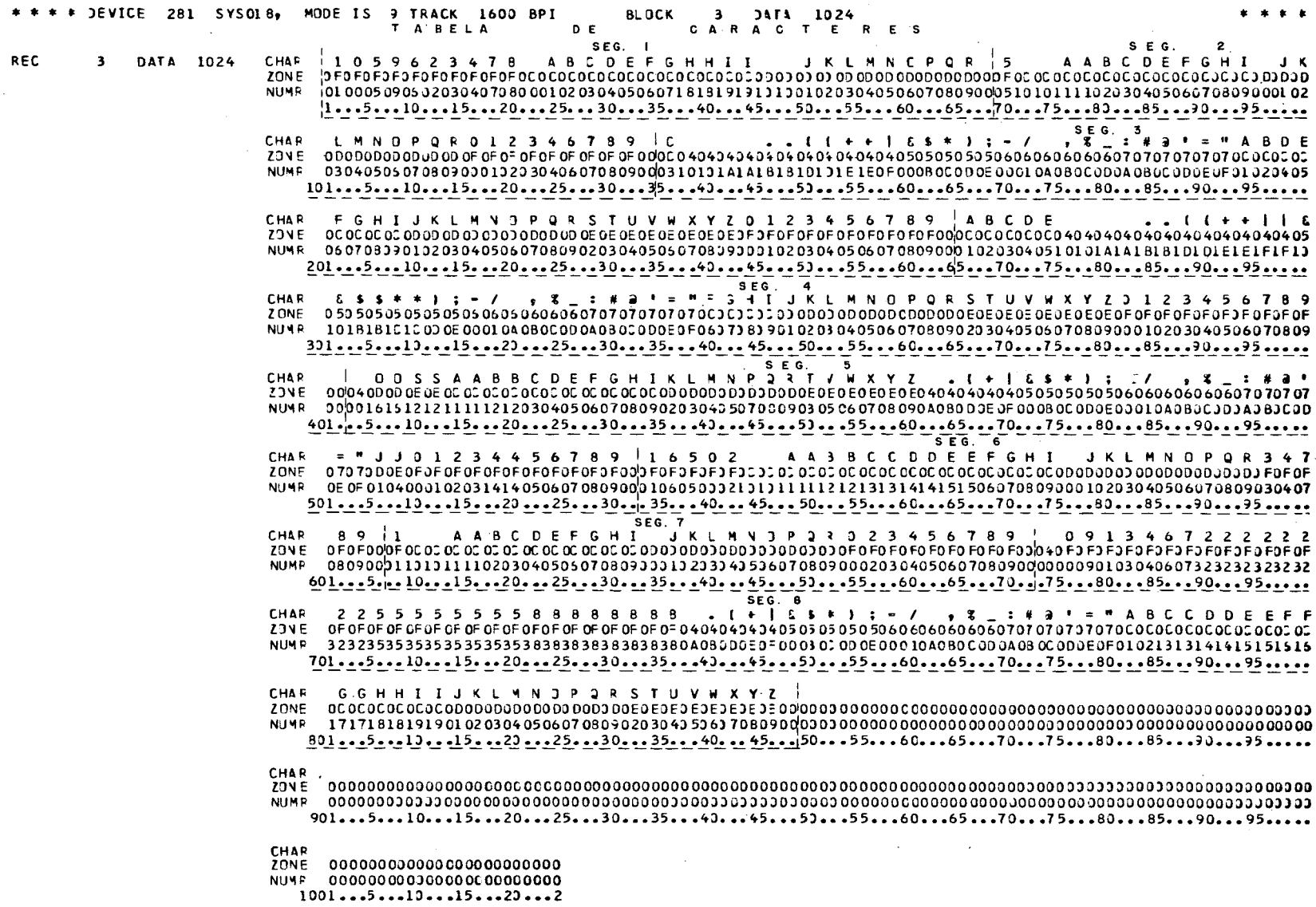
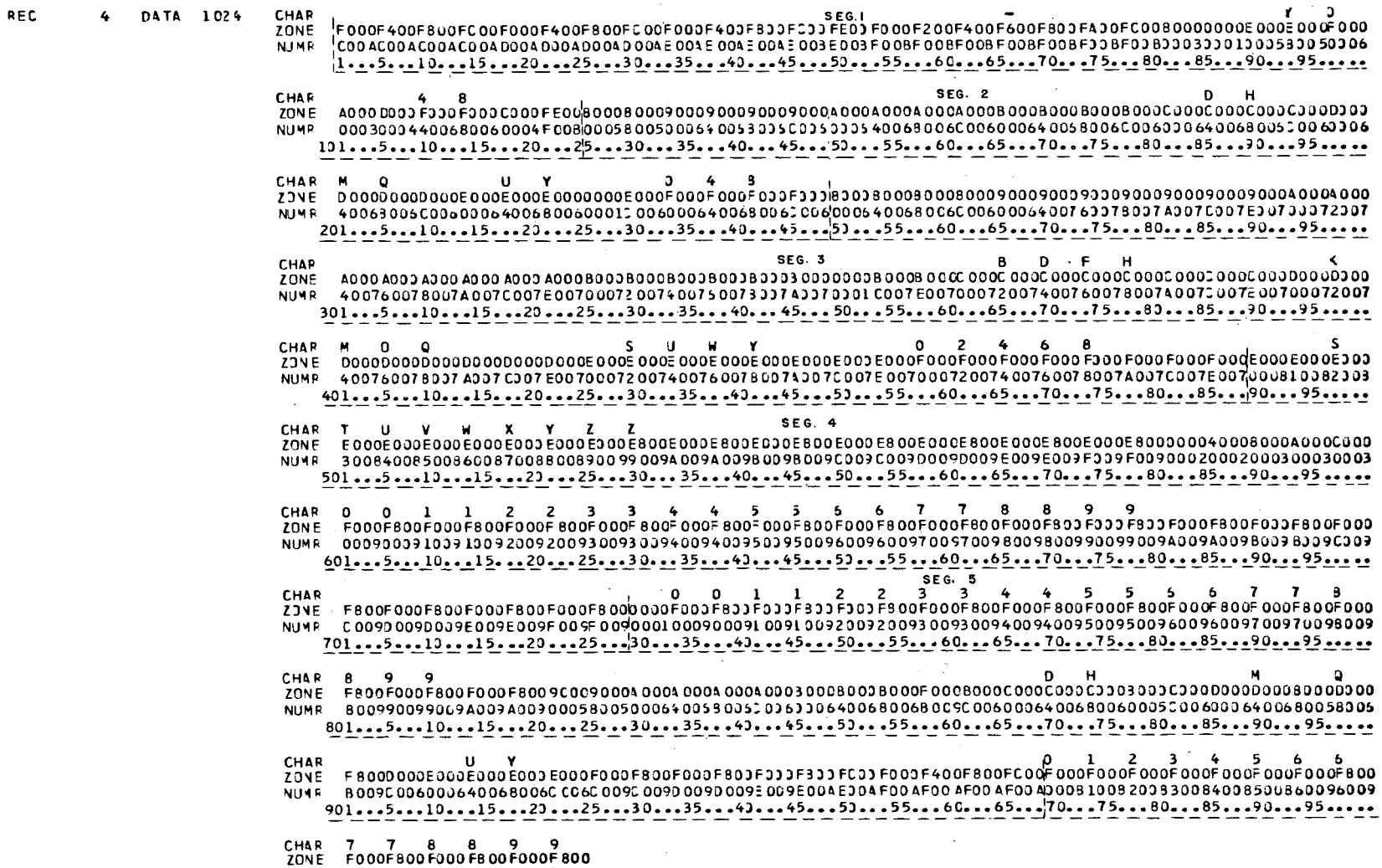

**** DEVICE 281 SYS018, MODE IS 9 TRACK 1600 BPI BLOCK 3 DATA 1024 ****
TABELA DE CARACTERES
REC 3 DATA 1024
CHAR
ZONE
NUMR
SEG. 1
SEG. 2
SEG. 3
SEG. 4
SEG. 5
SEG. 6
SEG. 7
SEG. 8

TABELA DE CÓDIGOS

REC 4 DATA 1024



```
* * * * DEVICE  281  SYS018,  MODE IS  9 TRACK  1600 BPI        BLOCK     5    DATA  1024                       * * * *

REC     5    DATA  1024   CHAR                        SEG. 6
                          ZONE   F000F800F000F800F000F800C0000000E000F000F800A0008000F00CF80CF000F800B0008000900090009000900A000A000
                          NUMR   A009A009B009B009C009C0090003000100040009000000030003E009E009F009F00900058005000640068006006000640064006
                                 1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR                                   D    H      SEG. 7       M    Q                     U    Y         0    4    8              6
                          ZONE   A000A000B000B000B000B000C000C000C000C0000000000000D00000000D000E000E000E000E000F000F000F000F00000000F000
                          NUMR   8006C00500640068006C00600064006800 6C00600640068005000100060006400680 06C0060006400680 06C00500015000A
                                 101...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   6    6    6    7    7    7    7    8    8    8    8    9    9    9    9
                          ZONE   F400F800FC00F000F400F800FC00F000F400F800FC00F000F400F800FC00F000F400F800FC00F000F400F800FC00F000F400
                          NUMR   600A600A600A700A700A700A700A800A800A800A800A900A900A900A900AA00AA00AA00AA00AB00AB00AB00AB00AC00AC00A
                                 201...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR                                    SEG. 8                                                           +    0
                          ZONE   F800FC00F000F400F800FC00FE00F000F200F400F600F800FA00FC00FE00F000F200F400F600F800FA00FC00800C000F000
                          NUMR   C00AC00AD00AD00AD00AD00BD00BE00BE00BE00BE00BE00BE00BE00BE00BF00BF00BF00BF00BF00BF00BF00B0002800050007
                                 301...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR        Q    2              Y    4
                          ZONE   D000D000F000E000E000F000C000FE00000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR   00058005200700058005400700 05F00B000000000000000000000000000000000000000000000000000000000000000000000
                                 401...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE   0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR   0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                 501...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE   0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR   0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                 601...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE   0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR   0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                 701...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE   0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR   0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                 801...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE   0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                          NUMR   0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                                 901...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR
                          ZONE   000000000000000000000000
                          NUMR   000000000000000000000000
                                 1001...5...10...15...20...2
```

Desta forma, para o nosso arquivo de testes, obtivemos os resultados acima, a partir dos quais podemos observar, para cada um dos blocos de 1024 bytes encontrados e a nível de segmento (8 segmentos) os dados abaixo:

- bloco 1 - Tabela de Descrição de Segmentos (TDS) e a Tabela de Descrição de Campos.
- bloco 2 - Padrão de Ausência e a Tabela de Prefixos.
- bloco 3 - Tabela de Caracteres.
- blocos 4 e 5 - Tabela de Códigos.

5.4. UTILIZAÇÃO DOS RELATÓRIOS

Como já foi dito anteriormente, quando projetamos estes relatórios, tínhamos em mente oferecer ao usuário uma ferramenta pela qual fosse possível a determinação, através de um modo racional e preciso, da compressão resultante da aplicação, em um dado arquivo, das técnicas de compressão por Segmentação e Código de Tamanho Variável - Huffman - Shannon - Fano.

Com este objetivo, foram desenvolvidos o Mapa Estatístico dos Segmentos e o Mapa Estatístico dos Alfabetos, os quais, fornecem ao usuário todo o suporte necessário a um perfeito acompanhamento do processo.

Finalmente, insistimos na observação de que a compressão final e a performance das rotinas são função direta de um bom projeto dos segmentos uma vez que, quanto mais bem projetados forem os segmentos, maior será a compressão por Segmentação, menor será o número de segmentos sem conteúdo, e, consequentemente, menor será a utilização do código HSF. De acordo com esta lógica, aconselhamos, mais uma vez, que, ao segmentar um registro, sejam desenvolvidos vários projetos sendo, a escolha do mais eficiente, feita pelo estudo comparativo dos relatórios resultantes deste programa.

Por tais razões, não existem procedimentos de utilização destes relatórios, desenvolvidos, uma vez que esta

utilização se prende mais a uma análise subjetiva e minuciosa dos dados ali apresentados, feita pelo usuário, buscando:

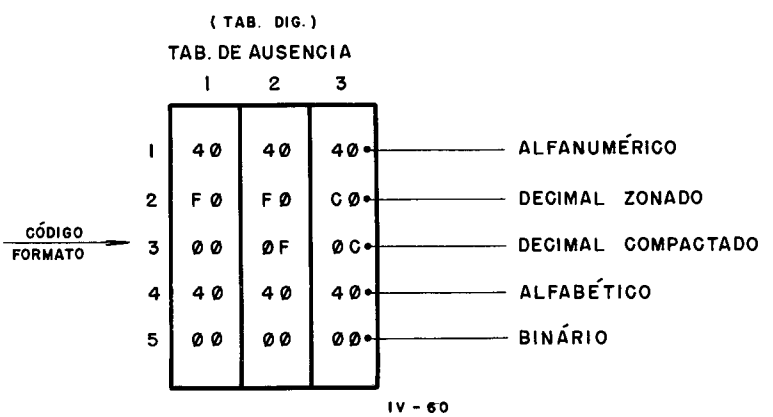
- Uma maior taxa de compressão.
- Uma maior eficiência das rotinas.
- Um acompanhamento preciso de todo o processo.
- O projeto de segmentos mais eficiente.

5.5. ESTRUTURA DO PROGRAMA

5.5.1. Tabelas

a) Tabela de Ausência (TAB DIG)

Tabela que fornece, para cada formato, os caracteres correspondentes à ausência de conteúdo.



Os caracteres estão representados em hexadecimal e as colunas 2 e 3 representam as combinações possíveis de sinal para campos sem conteúdo.

b) Tabela de Formatos (TABFOR)

Tabela em que, a partir do código de formato, são fornecidos os dados a seguir:

TABELA DE FORMATOS (TAB. FOR)

CÓDIGO FORMATO			
1	ALFANUMÉRICO	60	000
2	DECIMAL ZONADO	31	512
3	DECIMAL COMPACTADO	14	768
4	ALFABÉTICO	28	256
5	BINÁRIO	00	999

CÓDIGO DE FORMATO A SER UTILIZADO PELA ROTINA DE COMPRESSÃO E DESCOMPRESSÃO (FORMAT) – 3 DIGITOS ZONADO.

NÚMERO DE ELEMENTOS DO ALFABETO FIXADO PARA CADA FORMATO (QUANT.-CARAC.) – 2 DIGITOS ZONADO.

FORMATO, POR EXTENSO, PARA UTILIZAÇÃO NOS RELATÓRIOS (FORM) – 16 CARACTERES.

IV – 61

c) Tabela de Descrição dos Segmentos (TDS)

Tabela de utilização interna do programa, que, por meio da identificação do segmento, obtem ou registra informações relativas a segmentos do registro.

Projetada para um máximo de 64 segmentos, esta Tabela reúne em cada entrada, os seguintes dados.

TAB. DE DESCRIÇÃO DOS SEGMENTOS (TDS)

IDENTIFICAÇÃO SEGMENTO	TAM_SEG	FREQ_SEG	COD_SEG	FREQ_CRCS	APT_TDC
1					
2					
3					
...					
64					

APONTADOR PARA A TABELA DE DESCRIÇÃO DE CAMPOS, INDICANDO ONDE SE ENCONTRA, NESTA TABELA, O ÚLTIMO CAMPO DO SEGMENTO EM ANÁLISE – 3 DIGITOS COMPACTADOS.

PARA OS SEGMENTOS COM CONTEÚDO, INDICA O NÚMERO TOTAL DE CARACTERES OBSERVADOS ACRESCIDO DO NÚMERO DE ELEMENTOS DO ALFABETO CORRESPONDENTE AO SEGMENTO – 3 DIGITOS COMPACTADOS.

FORMATO DO SEGMENTO – 1 DIGITO COMPACTADO.

NÚMERO DE OCORRENCIAS DO SEGMENTO COM CONTEÚDO NA AMOSTRA – 7 DIGITOS COMPACTADOS.

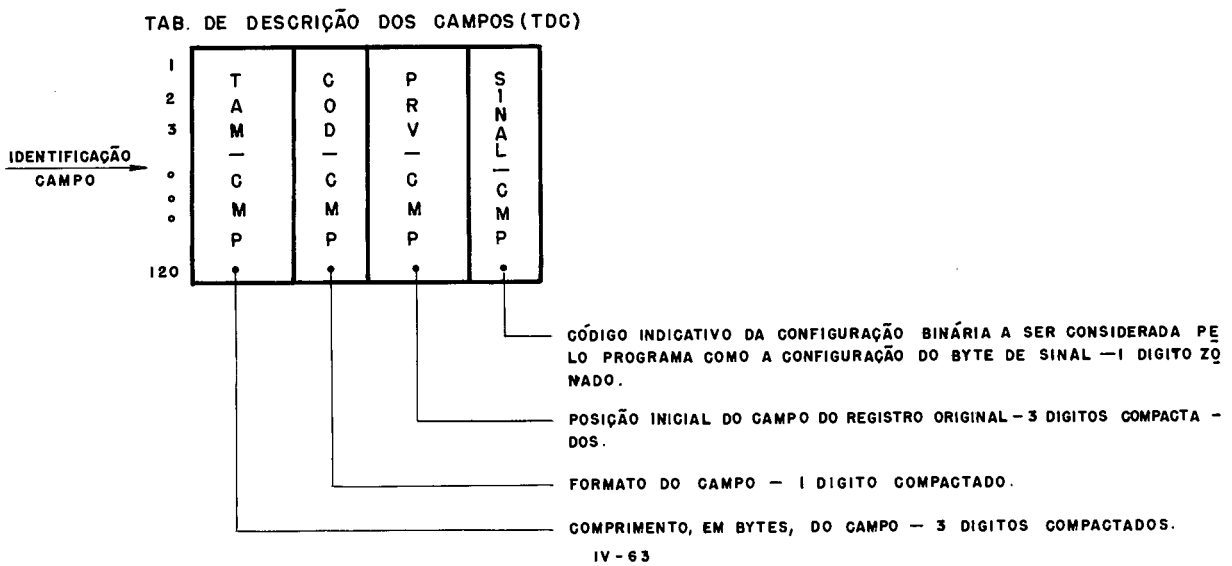
COMPRIMENTO, EM BYTES, DO SEGMENTO – 3 DIGITOS COMPACTADOS.

IV – 62

d) Tabela de Descrição dos Campos (TDC)

Esta tabela complementa a Tabela de Descrição Segmentos (TDS) uma vez que ela reúne em cada entrada dados descritivos de cada campo no registro. As entradas nesta tabela se apresentam na mesma sequência dos campos no registro segmentado.

Projetada para um máximo de 120 campos, esta tabela reúne, em cada entrada, os seguintes dados.



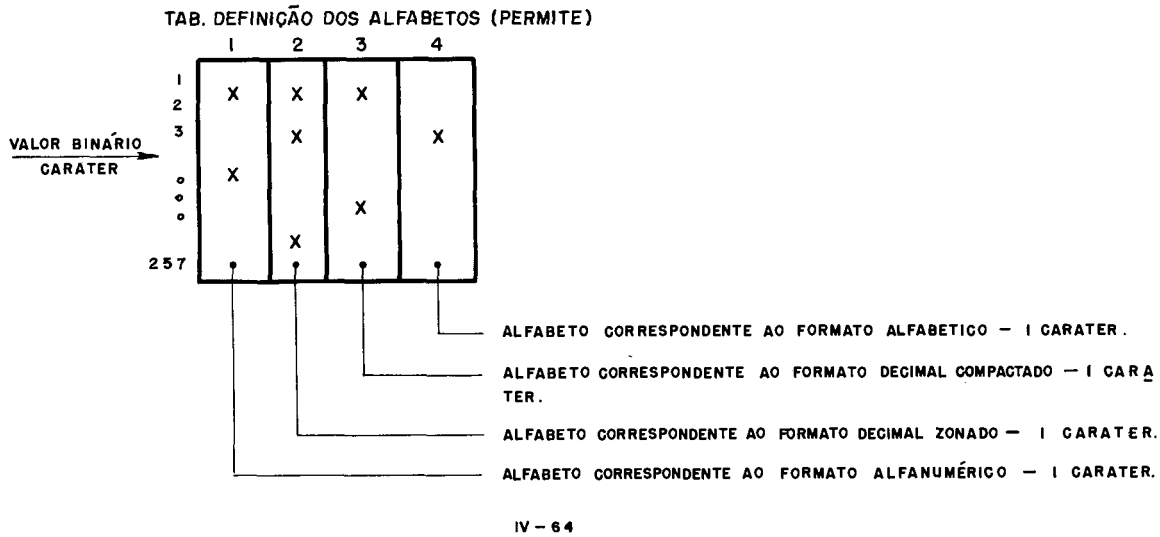
e) Tabela de Definição dos Alfabetos (PERMITE)

Através desta tabela definimos os caracteres componentes dos alfabetos para cada formato.

Assim, uma vez que segmentos no formato Binário não serão compressos, teremos de definir apenas 4 alfabetos para os formatos Alfanumérico, Alfabético, Decimal Zonado e Decimal Compactado.

Desta forma a tabela desenvolvida é bidimensional e nela as 257 linhas representam todas as configurações binárias para um byte (mais uma marca na linha 257), e as 4 colunas representam os 4 formatos para os quais serão definidos os alfabetos. Esta definição é feita através de um 'X' na tabela indicando que aquela configuração

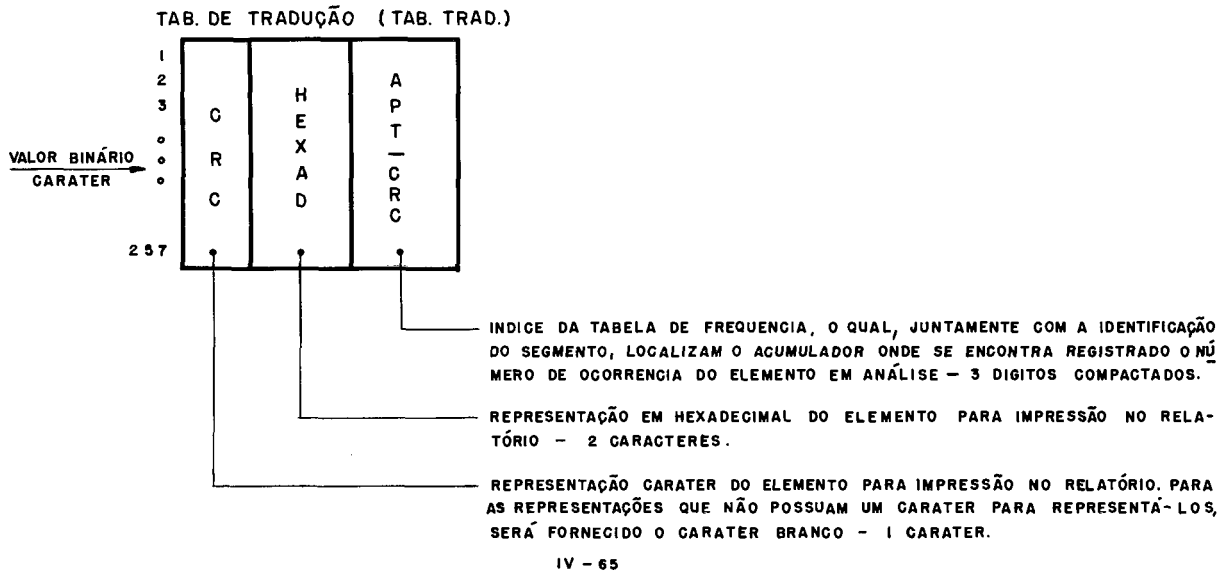
binária se constitui em um elemento do alfabeto para o formato em evidência.



IV - 64

f) Tabela de Tradução (TAB TRAD)

Esta tabela é de utilização interna do programa e sua finalidade básica é uma vez acessada pelo valor binário de um elemento de um alfabeto, fornecer para este elemento os dados abaixo:



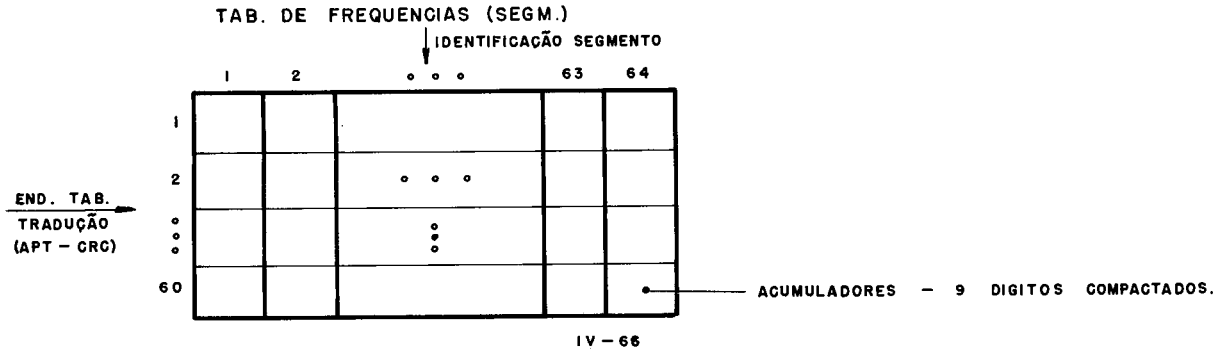
IV - 65

g) Tabela de Frequências (SEGM)

Esta é uma tabela bidimensional onde são registrados, para cada segmento analisado, o número de ocorrências dos caracteres encontrados.

Para sua utilização ela foi estruturada segundo

uma pilha de espaço disponível onde cada item no formato abaixo, fica liberado sempre que um dado caráter se apresenta pela primeira vez na amostra, ficando registrado na Tabela de Tradução (APT-CRC) o endereço do item, para registro das próximas ocorrências do caráter.



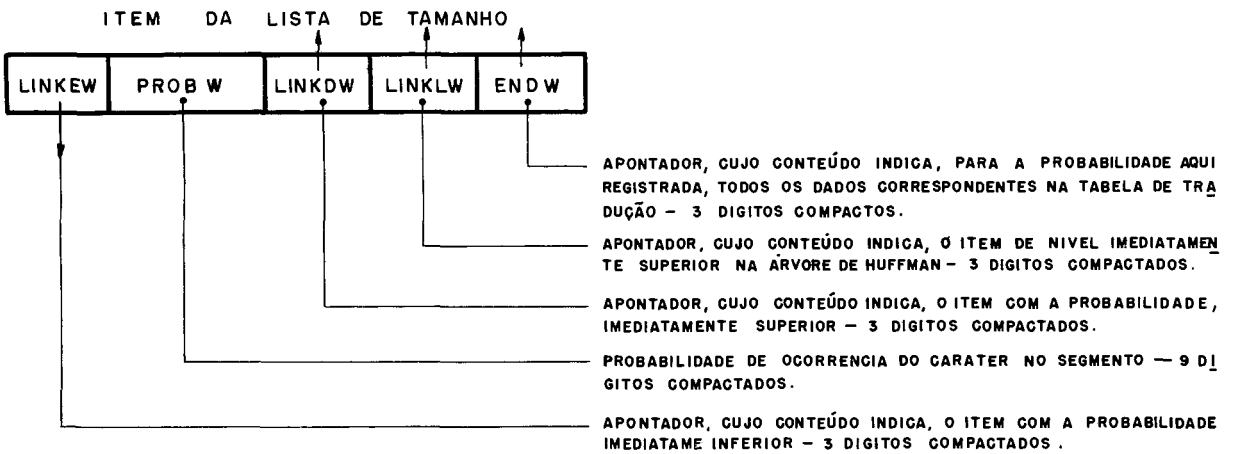
Podemos observar na figura acima que, cada item possui 64 acumuladores, referentes ao máximo de 64 segmentos, nos quais registramos as ocorrências do caráter, dependendo do segmento em que o mesmo apareceu.

Chamamos a atenção, ao final, para o fato de que a pilha tem 60 itens isto por ter o nosso maior alfabeto 60 elementos (alfanumérico).

h) Lista de Trabalho (LISTA-WORK)

Como o próprio nome sugere, esta é uma estrutura de trabalho utilizada na determinação do código HSF referente a cada segmento. Nesta estrutura a cada segmento do registro carregamos as probabilidades de ocorrência relativas a cada caráter do alfabeto. Ao final a árvore representativa do código Huffman correspondente ao segmento em questão, é obtida através do manuseio adequado dos itens desta estrutura.

Projetada para um máximo de 520 itens, esta estrutura reúne a cada item os dados a seguir:

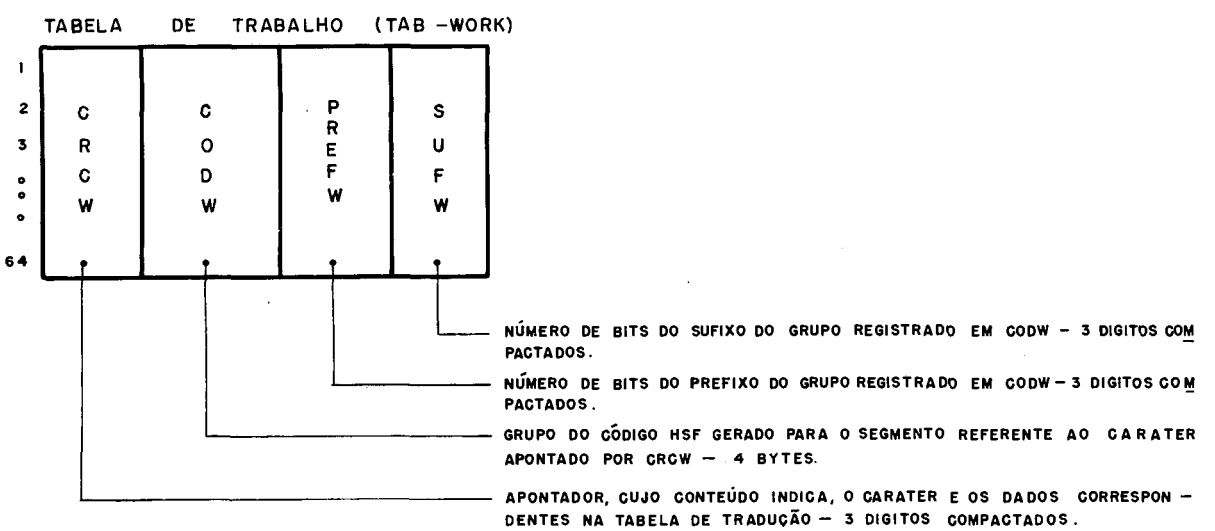


IV-67

i) Tabela de Trabalho (TAB-WORK)

Tabela de Trabalho utilizada pelo programa para auxiliar a determinação de alguns dados relativos ao Arquivo de Códigos.

Projetada para um máximo de 64 entradas, esta tabela é utilizada a cada segmento processado e nela são registrados, para cada elemento do alfabeto, os dados abaixo:



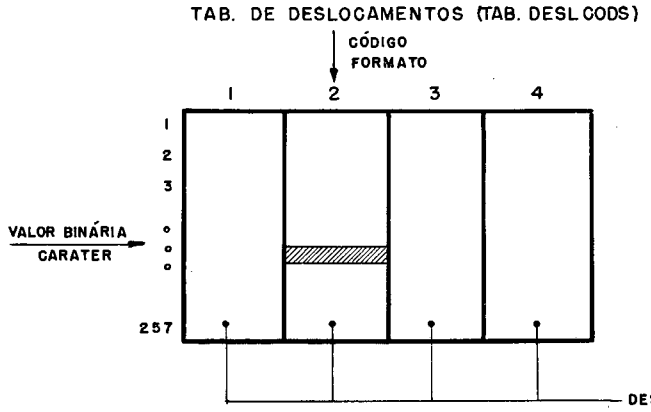
IV-68

j) Tabela de Deslocamentos (TABDESLCODS)

No capítulo III, quando descrevemos o método desenvolvido para a implantação do código HSF, vimos que os grupos correspondentes a cada código gerado eram armazenados, dependendo do segmento e de seu formato, em posições fixas na Tabela de Código. Para obtenção destes grupos utilizavamos uma tabela fixa, residente na rotina de compressão, a partir da qual, através do caráter fonte e do formato do segmento selecionavamos o deslocamento em bytes que compunha o endereço efetivo que nos daria, na Tabela de Códigos, o grupo desejado.

Esta tabela, contem para cada elemento dos 4 alfabetos (1 alfabeto para cada formato), um índice que indica, para um dado segmento e formato o deslocamento que permitirá armazenar na Tabela de Códigos cada um dos grupos do código HSF gerado. Como informações complementares observamos que tanto o índice como o deslocamento são dados em termos relativos.

Assim, para registro ou obtenção de um grupo na Tabela de Códigos devemos, primeiramente, determinar o índice ou endereço base da subtabela correspondente aos grupos do segmento e adicioná-lo ao índice ou deslocamento obtido para o grupo desejado.



DESLOCAMENTOS QUE INDICAM NA TABELA DE CÓDIGOS, A LOCALIZAÇÃO FIXA DE CADA UM DOS GRUPOS DO CÓDIGO HSF CORRESPONDENTE A CADA SEGMENTO - 2 DIGITOS ZONADO.

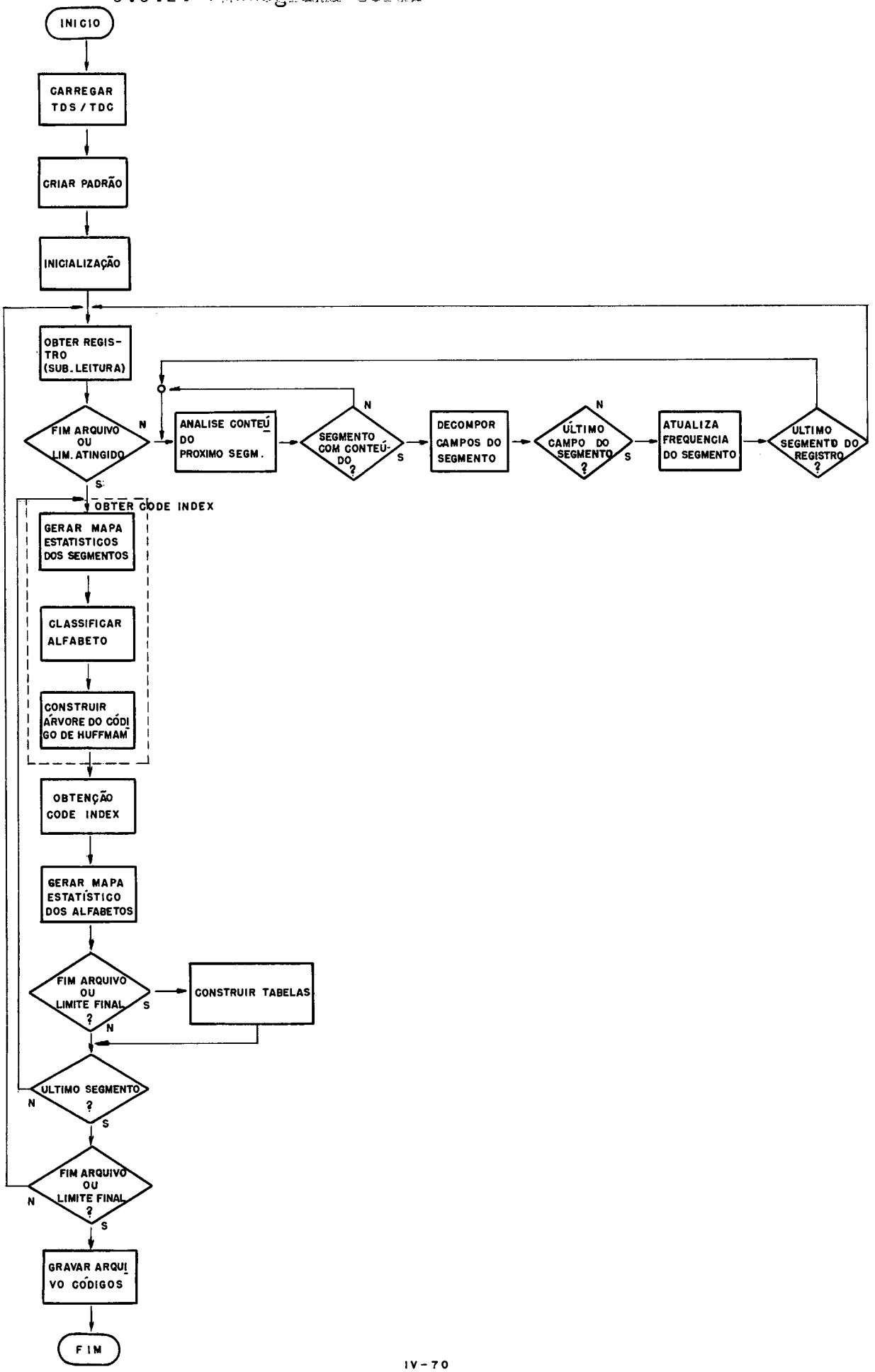
IV-69

k) Vetor de Comprimento do Código (INDIC)

Neste vetor, para auxiliar a determinação do código HSF, registramos, a cada segmento processado, o número de grupos de comprimento 1, 2, 3..., 27 bits. O comprimento de cada grupo é determinado ao se percorrer a árvore de Huffman e usado para indexar o Vetor de Comprimento.

Observamos, que aqui, embora participem do conjunto de tabelas do programa não fizemos qualquer referência as Tabelas de Código (CODS), Prefixos (UNS), Caracteres(TAB-CRC) e Padrão de Ausência, por já terem sido exaustivamente expostas quando da apresentação do conteúdo do Arquivo de Códigos.

5.5.2. Fluxograma Geral



5.5.3. Procedimentos

Da mesma forma como foi mostrado para o programa "Gerador de Estatísticas para Segmentação", (COMPØ3Ø), cada bloco do fluxograma poderá compreender um ou mais parágrafos do programa.

Visto isto, passamos a execução do programa, a qual reúne os procedimentos abaixo:

a) Módulo 'Carregar TDS/TDC'

Ao iniciar sua execução o programa, através deste módulo, processa a massa de "Descrição dos Segmentos" e por meio dela e de algumas inicializações, preenche as Tabelas de Descrição dos Segmentos e Descrição dos Campos do programa (TDS e TDC) e do Arquivo de Códigos (CTDS e CTDC).

Este preenchimento consiste na obtenção de dados tais como formato do segmento, número de campos por segmento, tamanho do segmento, posição inicial do campo no registro original, tamanho do campo, código de sinal do campo, etc, os quais podem ser apenas transferidos da massa de entrada ou calculados a partir desta.

b) Módulo 'Criar Padrão'

Uma vez que a Tabela de Descrição dos Campos(TDC) se apresenta convenientemente preenchida, passamos a geração do Padrão de Ausência, onde, a partir do proceessamento sequencial da TDC, criamos um registro segmentado totalmente sem conteúdo.

Como sabemos, as entradas na Tabela de Descrição dos Campos apresentam-se na mesma sequência dos campos no registro segmentado. Desta forma, ao processarmos entrada a entrada da tabela, vamos formatando campo a campo do registro, usando para isto o código de formato,o código de sinal e o tamanho do campo.

c) Módulo de 'Inicialização'

Através da console, o programa requisita do

usuário a data, limite para geração dos relatórios e o limite máximo de registros a processar.

d) Módulo 'Obter Registro'

Obter próximo registro lógico do arquivo do usuário, através da Subrotina de Leitura usando o parâmetro RG.

Para o nosso caso, como o arquivo do usuário se encontra em fita, denominamos esta versão da Subrotina de Leitura de "LERFITA".

CALL 'LERFITA' USING RG FIM-LEIT.

e) Se for atingido o fim de arquivo (indicado pelo caráter '*' no parâmetro FIM-LEIT), e/ou limite de geração dos relatórios, passar ao procedimento "1".

f) Módulo 'Análise Conteúdo Próximo Segmento'

Tomar o próximo segmento e verificar se o mesmo está sem conteúdo.

Para esta verificação, analisamos sequencialmente campo a campo de cada segmento do registro, determinando a ausência de conteúdo, quando para cada campo do segmento, dependendo do seu formato, encontramos por valor zeros ou brancos.

Esta análise é feita byte a byte, para cada campo do segmento, utilizando-se para isto da Tabela de Ausência, a qual contém, para cada formato, a configuração binária indicativa da ausência de conteúdo para o byte de sinal (colunas 2 e 3) e para os bytes restantes(coluna 1). Lembramos que o byte mais à direita de qualquer campo é convencionado como byte de sinal.

g) Se o segmento não tem conteúdo, voltamos ao item "f".

h) Módulo 'Decompor Campos do Segmento'

Uma vez comprovado que o segmento tem conteúdo, passamos à execução deste módulo o qual, de acordo com o formato do segmento,vai decompondo cada campo deste segmen

to em seus elementos básicos, atualizando, a cada elemento extraído, a sua frequência e a frequência total de elementos no segmento.

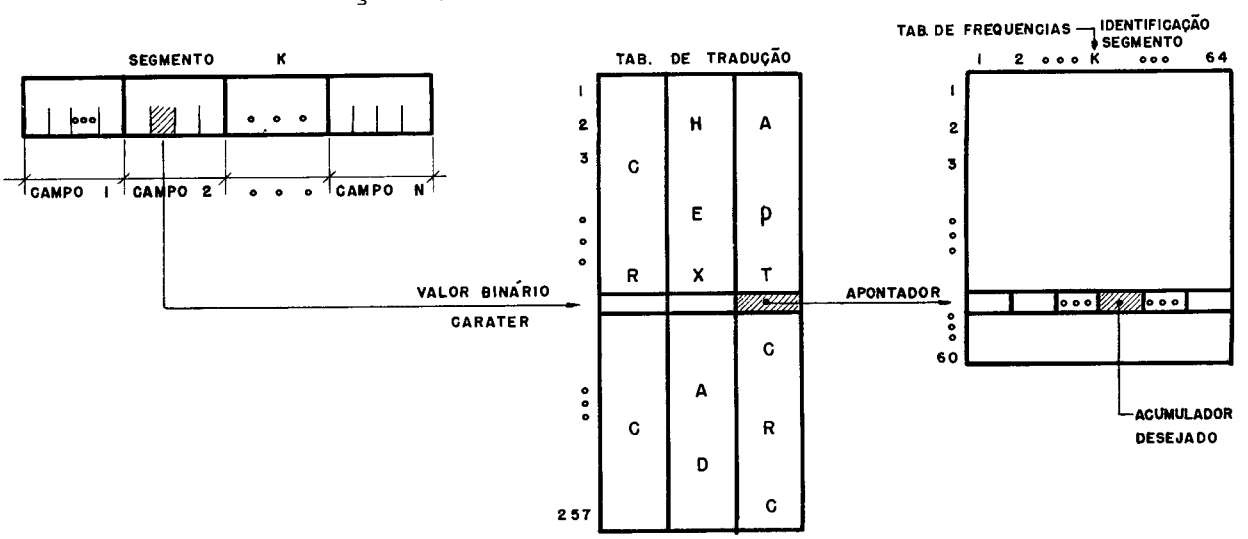
Dependendo do formato, os elementos básicos,(caracteres fonte), podem-se encontrar armazenados em:

- 4 bits - formato Decimal Compactado
- 8 bits - formatos Decimal Zonado, Alfabético e Alfanumérico.

Desta forma, da esquerda para a direita, vamos processando cada campo do segmento, de modo que a cada carater obtido, atualizamos a frequência do acumulador correspondente ao elemento e segmento.

A localização deste acumulador é feita tomando-se o valor binário correspondente ao elemento selecionado indexando a Tabela de Tradução de onde retiramos um índice (APT-CRC), o qual, juntamente com a identificação do segmento que está sendo processado, determinam, na Tabela de Frequências, o acumulador desejado.

Convém lembrar que a Tabela de Frequências, está estruturada segundo uma pilha de espaço disponível,composta de 60 itens, cada item com 64 acumuladores (1 para cada segmento). Assim, se um dado elemento aparece pela primeira vez na amostra, o próximo item da pilha é solicitado e a ligação do elemento com o item, é estabelecida através da colocação do índice referente ao item na Tabela de Tradução (APT-CRC).



Caso o número de itens projetados não seja suficiente, a mensagem abaixo será impressa na console: "ESTOUROU TABELA DE FREQUÊNCIA".

A mensagem chama a atenção do usuário para o fato de ter sido usado um alfabeto de número de itens superior ao máximo previsto. O processamento continuará e a partir deste ponto os novos itens serão ignorados pelo programa.

i) Se não for o último campo do segmento voltar ao item "h".

j) Incrementar, na Tabela de Descrição dos Segmentos (TDS), a frequência de ocorrência do segmento.

k) Se último segmento do registro, voltar ao item "d", caso contrário voltar ao item "f".

l) Módulo 'Gerar Mapa Estatístico dos Segmentos'

Calcular e imprimir o "Mapa Estatístico dos Segmentos".

m) Módulo 'Obter Code Index'

Neste módulo, a cada segmento do registro, é gerado, para o alfabeto correspondente ao mesmo, o Code Index.

Sabemos que o Code Index é definido como uma concatenação de números decimais T_1, T_2, T_3 ..., T_M

$$I = T_1\ T_2\ T_3 \cdots T_M$$

onde T_i - representa o número de grupos, em um código de tamanho variável, que possuem comprimento "i", sendo "M" o comprimento do maior grupo.

Por outro lado, para que o Código HSF apresente o mesmo custo médio do código de Huffman (mínimo) gerado para amostra, ou seja, o mesmo Code Index, deveremos executar as seguintes etapas:

m.1) Classificar Alfabeto

Organizar os elementos do alfabeto corres-

pondente ao segmento em ordem decrescente de probabilidade de ocorrência.

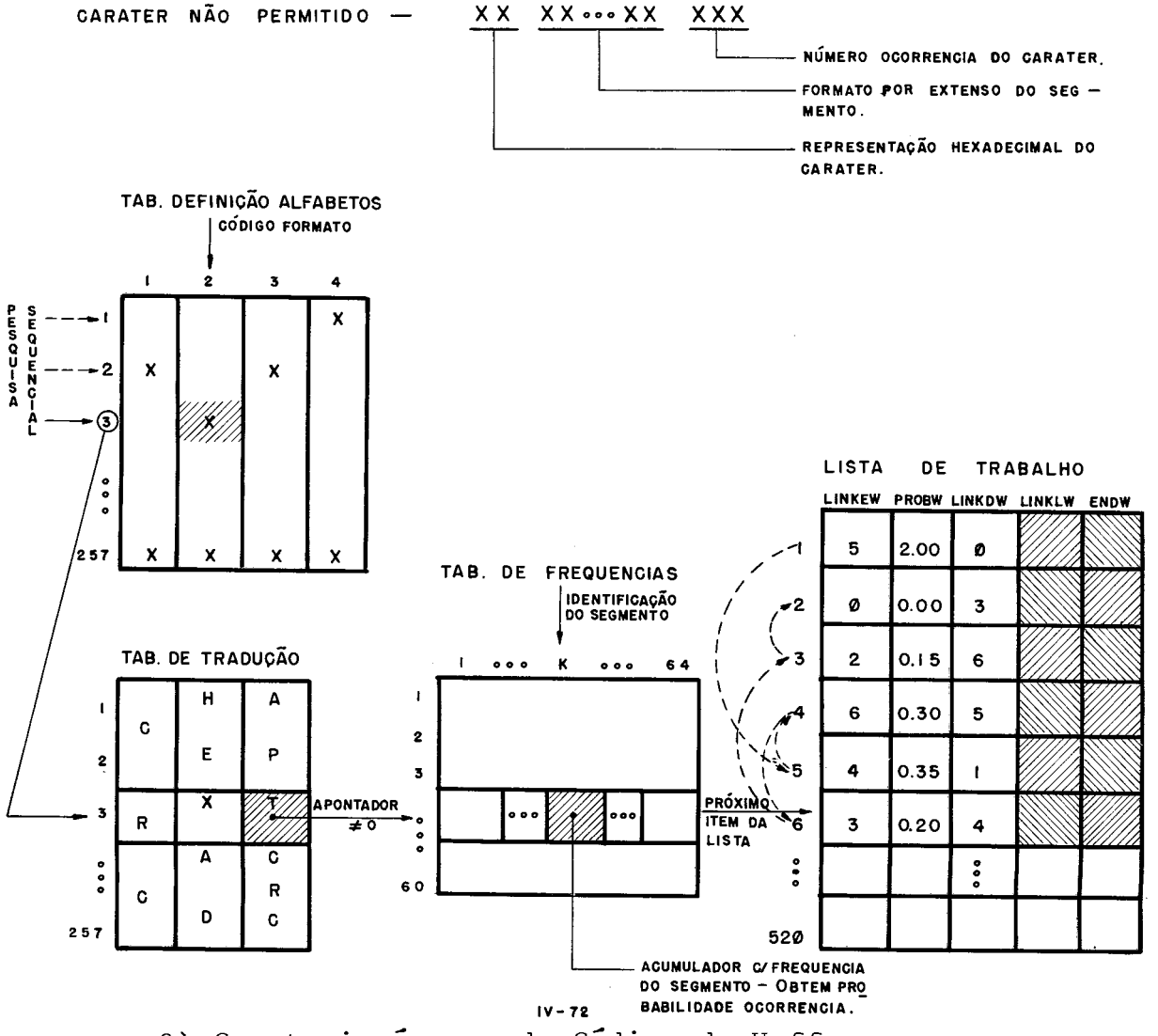
Esta etapa se encontra no programa como se segue:

- Pesquisar sequencialmente a Tabela de Definição dos Alfabetos - (PERMITE), selecionando os elementos do alfabeto correspondente ao formato do segmento.
- Para cada elemento selecionado:
  - Acessar a Tabela de Tradução (TABTRAD) analisando o conteúdo do apontador da Tabela de Frequência (APT-CRC).
  - Se APT-CRC = Ø, tomar para frequência de ocorrência do elemento o valor 1, uma vez que o mesmo ainda não apareceu na amostra.
  - Se APT-CRC ≠ Ø, tomar para frequência de ocorrência do elemento o conteúdo do acumulador indexado por APT-CRC, (obtem o item da pilha), e a identificação do segmento, (obtem o acumulador no item), na Tabela de Frequência.
  - Calcular a probabilidade de ocorrência.
  - Obter o próximo item disponível da lista de trabalho (LISTA-WORK), inicializando convenientemente os seus campos.
  - Pesquisar a lista de probabilidades existente, incluindo o item de modo que a lista permaneça em ordem decrescente de probabilidades se considerarmos o apontador LINKEW.

Estes procedimentos serão executados até que toda a Tabela de Definição dos Alfabetos (PERMITE), tenha sido pesquisada.

Observamos, para o caso de ser encontrado no segmento caracteres não pertencentes ao alfabeto, que o programa não os levará em consideração, infor-

mando apenas ao usuário do fato, por mensagem nos relatórios.



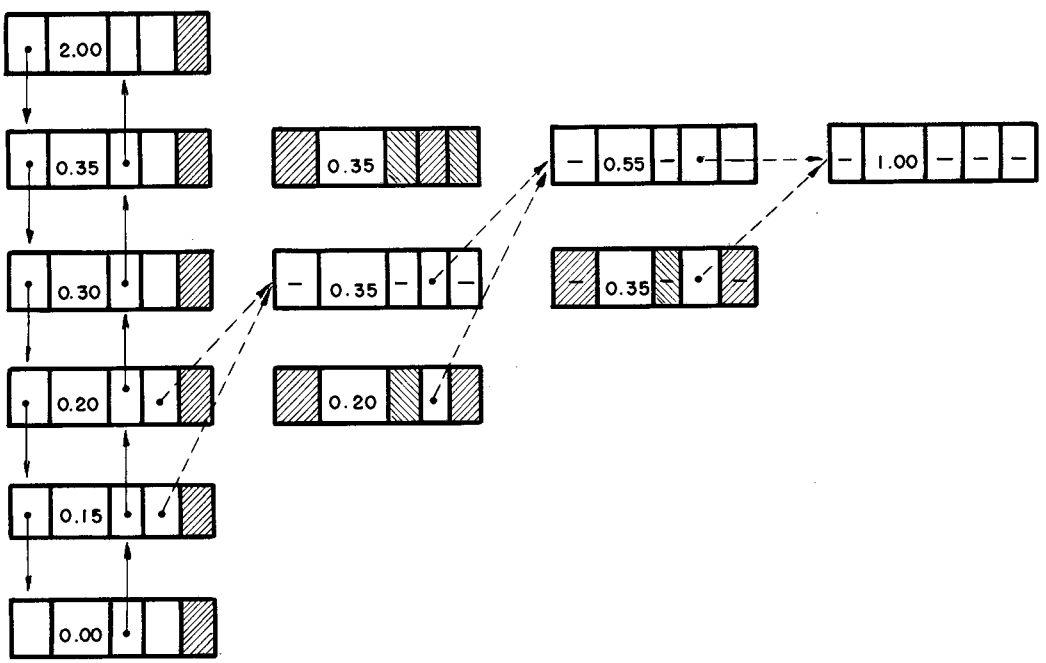
m.2) Construir árvore do Código de Huffman

Com a lista resultante do procedimento anterior onde se encontram registrados os elementos do alfabeto em ordem decrescente de probabilidade de ocorrência, partimos para a construção da árvore binária correspondente ao código de Huffman para o segmento em questão.

Lembrando a forma de construção do código de Huffman e tomando-se o campo cujo conteúdo endereça o item de probabilidade imediatamente superior, (LINKDW) passamos a perrer a lista, criando, a cada par de itens visitados, um novo item onde a probabilidade é a soma das probabilidades de cada cada item do par.

A cada novo item criado, passamos à forma-

tação da árvore, que compreende a adição do item à lista e o estabelecimento da ligação deste item aos itens geradores. Esta ligação é feita pela colocação do endereço - (índice), do item criado em campo apropriado dos itens geradores (LINKLW). Este procedimento será executado até que a probabilidade de ocorrência do item criado seja 1.



IV - 73

m.3) Obtenção do Code Index

Uma vez de posse da árvore binária correspondente ao código de Huffman para o alfabeto em questão, resta-nos, somente, obter o Code Index.

Para armazenar o Code Index projetamos um vetor, onde o índice indica o comprimento do grupo em número de bits e o conteúdo indica o número de grupos de um dado comprimento (INDIC). Observamos porém que, por razões de projeto do sistema, o comprimento máximo de um grupo será de 27 bits.

Isto posto, para a determinação do Code Index, processamos a lista obtida do item "m.2", como se segue:

- Percorrer, item a item, toda a lista de elementos do alfabeto em ordem crescente de probabili

dade de ocorrência. Para isto utilizamos o conteúdo do campo LINKDW o qual representa o endereço do item com o caráter de probabilidade de ocorrência imediatamente superior.

- A cada item da lista que for visitado, tomamos o LINKLW e, através dele, percorremos a árvore correspondente ao código de Huffman até a sua raiz. Desta forma, determinamos, pela contagem dos ramos percorridos, o número de bits de cada grupo do código de Huffman atribuído a cada elemento do alfabeto correspondente ao segmento.
- Através do número de bits do grupo indexamos o vetor do Code Index (INDIC) e incrementamos o conteúdo do respectivo acumulador.

n) Módulo 'Gerar Mapa Estatístico dos Alfabetos'

Observamos que por ocasião da geração deste relatório serão construídos todos os códigos HSF correspondentes a cada um dos segmentos.

Ainda nesta etapa, caso o limite-final e/ou fim de arquivo tenha sido atingido, a Tabela de Trabalho(TAB-WORK) é preenchida. Nela, para cada caráter do alfabeto atribuído ao segmento em análise, armazenamos os seguintes dados:

- Endereço do caráter na Tabela de Tradução (TABTRAD).
- O grupo com o comprimento atribuído ao caráter.
- Comprimento do prefixo do grupo.
- Comprimento do sufixo do grupo.

Se a quantidade de caracteres do alfabeto ultrapassar 64, a tabela não comportará mais esta entrada e será impressa na console a mensagem:

"ESTOUROU TAB-WORK",

após o que a execução do programa será encerrada.

o) Módulo 'Construir Tabelas'

Quando o fim de arquivo ou o limite máximo de

registros processados for atingido,este módulo em que a maioria das tabelas do Arquivo de Códigos são construídas ou complementadas, deve ser executado.

Aqui, a partir da Tabela de Trabalho, são construídas ou complementadas as seguintes Tabelas:

- Tabela de Descrição dos Segmentos (CTDS).
- Tabela de Prefixos (TABUNS).
- Tabela de Códigos (CODS).
- Tabela de Caracteres (TAB-CRC).

Caso o número máximo de segmentos ultrapasse 64 uma mensagem será impressa, na console para dar conhecimento do fato ao usuário e o programa será encerrado.

"ESTOUROU TDS"

Caso, ainda, o número de entradas na Tabela de Prefixos ultrapasse o limite de 256,uma mensagem será impressa na console para que o usuário tome conhecimento e o programa será encerrado.

"ESTOUROU TABUNS"

5.6. MODIFICAÇÕES

a) Para qualquer modificação relacionada com a forma de leitura do Arquivo-Aplicação, deveremos nos reportar à Subrotina de Leitura. Observar que no caso do comprimento máximo do registro ter sido modificado deveremos alterar, nes programa, as definições das variáveis RG e REGISTRO.

b) Caso o usuário deseje mudar o endereço lógico (SYSØØ9) ou o meio físico do "Arquivo - Descrição dos Segmentos", atualmente em cartão, bastará alterar, convenientemente,a cláusula SELECT no programa.

c) Se o usuário desejar mudar o endereço lógico (SYSØ11), ou necessite mudar o meio físico do "Arquivo-Códigos", atualmente em fita, bastará alterar, convenientemente, a cláusula SELECT no programa.

d) Para qualquer modificação com relação à impressão dos relatórios, reportar-se, à Subrotina de Impressão.

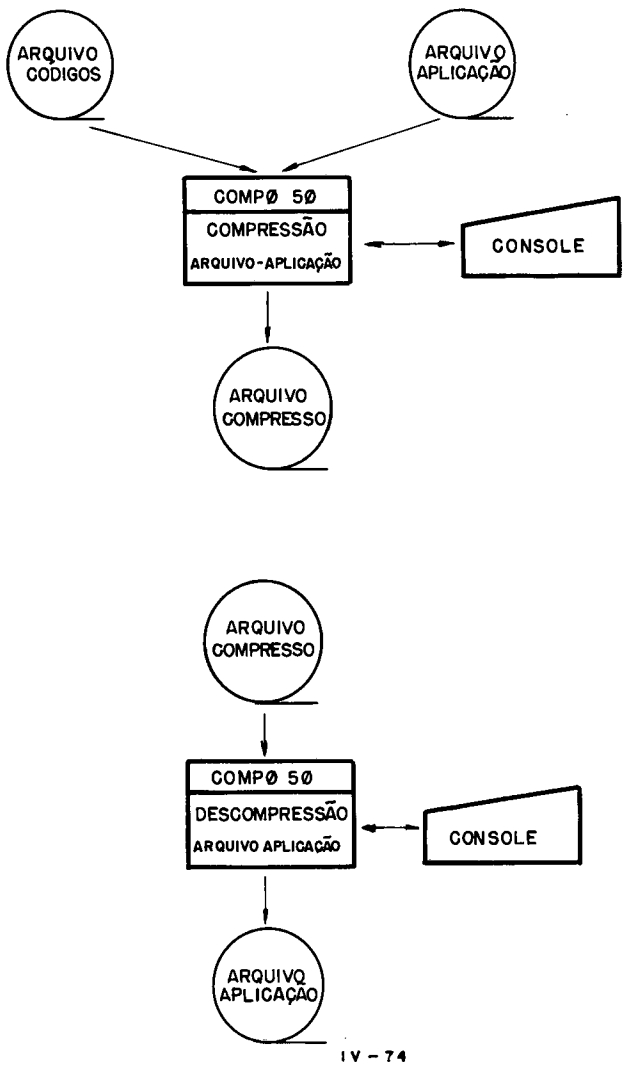
e) Aconselhamos que os limites relacionados abaixo não sejam alterados uma vez que os mesmos, por motivos de projeto têm implicações profundas em todo o sistema.

- Número máximo de segmentos no registro - 64.
- Número máximo de campos no registro - 120.
- Comprimento máximo do Padrão de Ausência e do registro lógico no Arquivo-Aplicação - 512 bytes.
- Número máximo de entradas na Tabela de Prefixos - 256.
- Número máximo de entradas na Tabela de Códigos - 512.
- Número máximo de entradas na Tabela de Caracteres - 512.
- Número máximo de bits em 1 grupo do código HSF - 27 bits.

f) Caso o número de entradas das tabelas relacionadas a seguir, não seja suficiente, o usuário poderá fazer a alteração no número de ocorrências na clausula OCCURS por ocasião da definição das tabelas.

- Lista de Trabalho (LISTA-WORK).
- Tabela de Trabalho (TAB-WORK).

6. PROGRAMA DE CONVERSÃO (COMPØ5Ø)



6.1. FINALIDADE

Este é o último programa utilitário do sistema e através dele é efetuada a conversão do arquivo do usuário (ARQUIVO APLICAÇÃO).

Esta conversão, como podemos ver acima, consiste, dependendo das necessidades do momento, da compressão ou descompressão do arquivo do usuário.

Observar que, o programa COMPØ5Ø, para efetuar a conversão utiliza o "ARQUIVO CÓDIGOS" gerado no programa COMPØ4Ø e as rotinas de compressão/descompressão, além de proceder a leitura (compressão) ou gravação (descompressão) do Arquivo Aplicação, seja qual for o comprimento do registro e o fator de bloco.

6.2. ENTRADAS

a) Arquivo Aplicação

Na compressão, é o arquivo do usuário na sua forma descompressa, o qual, tem como endereço lógico SYS012 e nome interno ARQUIVO.

Da mesma forma que para a Subrotina de Leitura, este arquivo deverá ter organização sequencial, registros de um mesmo tipo e de comprimento fixo. Para a atual versão do programa o comprimento máximo é de 200 bytes e o fator de bloco é 60. Se necessário, através de alterações no programa, o comprimento máximo pode ser ampliado até 512 bytes e o fator de bloco a um valor compatível com o maior bloco permitido pelo suporte de armazenamento do arquivo.

A versão apresentada por este programa processa arquivos em fita. Para outro meio físico ver item "6.5" (MODIFICAÇÕES).

b) Console

Através da console o programa solicitará do usuário as seguintes informações:

b.1) "DE FATOR DE BLOCO - TAMANHO REGISTRO XXX-XXX"

Em resposta, o usuário fornecerá o fator de bloco (3 dígitos) e o comprimento, em bytes, do registro lógico (3 dígitos) o que possibilitará a deblocagem do arquivo a ser trabalhado.

b.2) "BATA C/D SE COMPRESSÃO/DESCOMPRESSÃO

Já foi visto que este programa pode executar tanto a compressão como a descompressão do arquivo de aplicação. Assim, dependendo da necessidade do usuário no momento, a resposta deve ser dada teclando-se 'C' ou 'D' (maiúsculo).

b.3) "FITA LIBERADA 280"

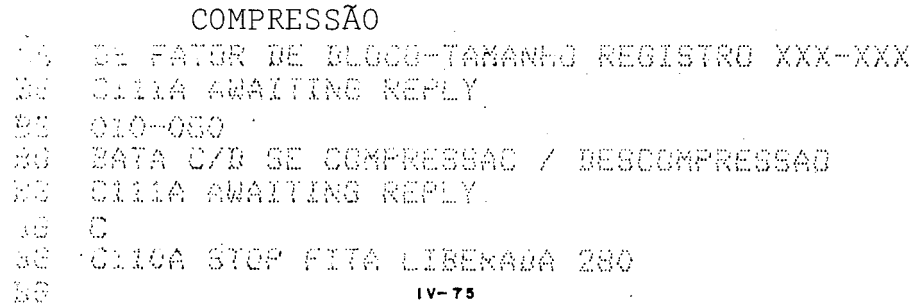
Esta mensagem indica que o operador deve colocar uma fita liberada na unidade cujo o endereço fí

sico é 280, durante este tempo o programa permanece no estado de espera, até que o operador, através da console, mande prosseguir a execução.

Diálogo usuário-programa quando da execução dos testes:

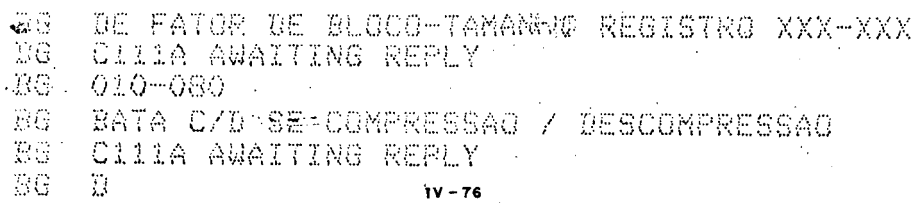
dados fornecidos - FATOR DE BLOCO - 010
COMPRIMENTO DO REGISTRO - 080
COMPRESSÃO DO ARQUIVO - C
DESCOMPRESSÃO DO ARQUIVO - D

COMPRESSÃO



IV-75

DESCOMPRESSÃO



IV-76

c) Arquivo Código

De nome interno ARQ e endereço lógico SYSØ1Ø, este arquivo, apresentado anteriormente quando descrevemos o programa COMPØ4Ø (item 5.3 - SAIDAS - item "C"), na versão atual das rotinas, se encontra armazenado em fita e possui todas as informações necessárias à execução das rotinas de compressão/descompressão.

Uma vez que já foi feita, uma apresentação de seu conteúdo é que o mesmo é processado somente pelas rotinas de compressão/descompressão, achamos conveniente deixar para o capítulo seguinte, onde descreveremos estas rotinas, com todo e qualquer comentário complementar.

d) Arquivo Compresso

Processado somente pela rotina de descompres-

são, este arquivo de nome interno ARQE e endereço lógico SYSØ1Ø, encontra-se na versão atual das rotinas, armazenado em fita, podendo, no entanto, serem utilizados outros suportes do armazenamento, conforme mostraremos no próximo capítulo, onde também apresentaremos todos os detalhes relativos a este arquivo.

6.3. SAÍDAS

a) Arquivo Compresso

Gerado pela rotina de compressão, este arquivo de nome interno ARQS e endereço lógico SYS011, se encontra, na versão atual das rotinas, armazenado em fita.

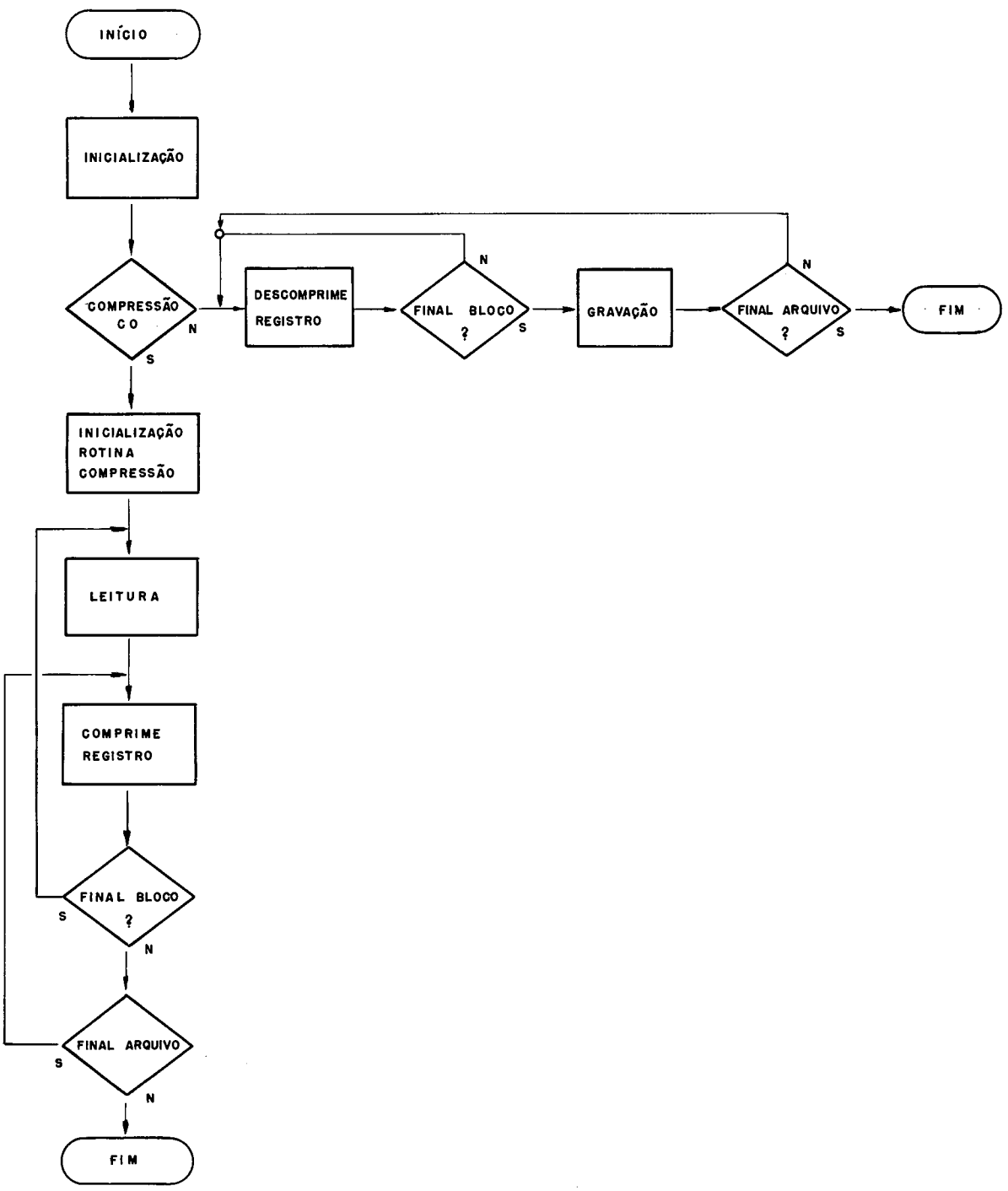
Maiores informações a respeito deste arquivo, serão dadas no próximo capítulo, quando descreveremos a rotina de compressão.

b) Arquivo Aplicação

Este é o mesmo arquivo do usuário apresentado no item "6.2 - ENTRADAS". A diferença reside no fato de naquele item era o arquivo do usuário a ser compresso, enquanto que em SAIDAS é um arquivo produto (saída) de uma descompressão.

6.4. ESTRUTURA DO PROGRAMA

6.4.1. Fluxograma Geral



6.4.2. Procedimentos

Do mesmo modo que na Subrotina de Leitura,este programa, dependendo da função a ser executada (compressão ou descompressão), procede a leitura ou gravação do Arquivo Aplicação para qualquer comprimento de registro e fator de bloco, desde que os mesmos atendam aos limites pré-estabelecidos para o sistema.

Isto é possível, uma vez que através da utilização de registros de tamanho indefinido, são lidos ou são gravados registros físicos (blocos) para ou de uma área estruturada segundo um array bidimensional. Neste array, o número de linhas corresponde ao fator de bloco e o número de colunas ao comprimento do registro, ambos fixados pelo usuário no início do processamento.

Para a execução deste programa, os procedimentos abaixo devem ser executados:

a) Módulo de 'Inicialização'

Ao iniciar sua execução o programa, através deste módulo requisita do usuário, por meio da console, o FATOR DE BLOCO, O COMPRIMENTO DO REGISTRO e o carater 'C' ou 'D' indicativo de uma compressão ou descompressão de arquivos.

b) Se os procedimentos a serem executados são de descompressão passar ao item "h".

c) Módulo 'Inicialização Rotina Compressão'

Para o caso de compressão este módulo será executado e através dele é chamada a rotina de descompressão (DCOMP) para proceder a leitura do Arquivo de Códigos e consequentemente carregar na memória todas as informações referentes ao arquivo a ser compresso.

Para isto, utilizamos o comando a seguir, onde REG é um parâmetro cuja única finalidade é conter uma marca na sua primeira posição ('/') indicando que esta chamada foi feita apenas para o carregamento das Tabelas e dos

dados relativos ao arquivo:

CALL 'DCOMP' USING REG.

d) Módulo de 'Leitura'

Através do módulo de 'Leitura', obter do arquivo o próximo registro físico (bloco).

e) Módulo 'Comprime Registro'

Utilizando-se da mesma filosofia empregada na Subrotina de Leitura, obter o próximo registro lógico do arquivo e pelo parâmetro REG e de uma chamada à rotina de compressão (COMP), comprimir o registro selecionado.

Para isto utilizamos o seguinte comando:

CALL 'COMP' USING REG.

f) Se o último registro do bloco, foi processado, voltar ao item "d".

g) Se fim de arquivo o programa deverá ser encerrado, caso contrário voltar ao item "e".

h) Módulo 'Descomprime Registro'

Obter, através do parâmetro REG e de uma chamada à rotina de descompressão, o próximo registro lógico do arquivo descompresso.

Para isto, utilizamos o comando abaixo:

CALL 'DCOMP' USING REG.

Empregando ainda, a mesma filosofia da Subrotina de Leitura, apenas (com a diferença que, aqui, vamos blocar ao invés de deblocar registros), anexamos o registro lógico obtido ao bloco em montagem.

i) Se o registro processado não é o último do bloco, voltar ao item "h".

j) Módulo de 'Gravação'

Gravar bloco no arquivo.

k) Se foi alcançado o fim de arquivo, indicado por um "*" na primeira posição do parâmetro REG gravar bloco e encerrar a execução do programa. Caso contrário voltar ao item "h".

6.5. MODIFICAÇÕES

a) Caso os limites máximos estabelecidos para comprimento do registro lógico (200 bytes), e fator de bloco (60),não satisfaçam ao usuário, estes limites poderão ser alterados pela mudança dos valores fixados.

   - Na cláusula RECORD CONTAINS - 1 a 12000.
   - Na cláusula OCCURS ao se definir o registro - 60 e 200.
   - No parâmetro REG - 200.

   Devemos frisar, contudo, que estas alterações devem estar contidas nos limites máximos estabelecidos.

   - Para o tamanho do bloco, o qual depende do suporte de armazenamento onde se encontra o arquivo.
   - Para o comprimento do registro - 512 bytes, o qual foi assim fixado por motivos de projeto do sistema.

b) Como a versão atual processa somente arquivos armazenados em fita, caso o usuário deseje utilizar outro meio físico ou outro endereço lógico, bastará alterar os campos devidos na cláusula SELECT relativa ao arquivo (ARQUIVO).

c) Com relação ao Arquivo Compresso e Arquivo Códigos,por serem eles manuseados pelas rotinas de compressão, achamos conveniente mencionar as modificações possíveis quando da apresentação destas rotinas, no capítulo seguinte.

CAPÍTULO V

MÓDULO DE EXECUÇÃO — SUBROTINA DE COMPRESSÃO/DESCOMPRESSÃO

1. APRESENTAÇÃO

Este módulo trata da parte do sistema, responsável pela compressão e/ou descompressão efetiva dos dados para os programas do usuário, que processam informações do arquivo compresso.

O módulo de execução, necessita de um conjunto de informações como tabelas, códigos, alfabetos, entre outras, que geradas pelo módulo de preparação e armazenadas no início do arquivo compresso permitem a sua execução.

O módulo consiste, estruturalmente, de uma rotina de compressão e uma rotina de descompressão, ambas compondo uma única subrotina com 2 pontos de entrada.

Neste capítulo, vamos apresentar detalhadamente cada uma das rotinas, mostrando as entradas, saídas, procedimentos de manutenção, estrutura interna, alterações possíveis, enfim, todos os pontos que lhes sejam relativos.

2. ROTINA DE DESCOMPRESSÃO

2.1. FINALIDADE

Esta parte da subrotina, denominada de 'DCOMP', é a responsável pela descompressão efetiva dos dados.

Através dela, o programa do usuário tem acesso aos registros do arquivo compresso na sua forma descompressa(conforme tivemos oportunidade de mostrar no Capítulo III - item 3 - "Implantação do Sistema"). Em outras palavras, a cada chamada a subrotina obtem o registro compresso por meio,de seus segmentos, descomprime segmento a segmento, executa todos os controles, formatando ao final para o programa chamador o registro original.

De modo geral ao se descomprimir segmento a segmen

to de um registro, temos que:

a) Para segmentos com conteúdo, decodificar utilizando os procedimentos de descompressão do código HSF.

b) Para segmentos sem conteúdo, decodificar, utilizando os procedimentos de descompressão do método de Segmentação.

c) A cada segmento decodificado, reagrupar os campos conforme formato do registro original.

2.2. ENTRADAS

a) Arquivo Aplicação Compresso

Este é o arquivo do usuário, compresso, a ser descompresso pela subrotina de compressão/descompressão.

Apresenta as seguintes características:

- Nome interno - ARQE.
- Endereço lógico - SYSØ1Ø.
- Label Standard - IBM.
- 2 áreas de entrada.
- Registros não blocados de tamanho fixo igual a 1024 bytes.

A versão atual do sistema, por imposição dos arquivos do usuário, foi desenvolvida, para o meio físico fita. Portanto, a descrição do arquivo apresentada na sub rotina, considerou também o meio físico fita. Como veremos, porém, no item de "MODIFICAÇÕES" é fácil a adaptação para outros suportes de armazenamento.

É interessante informar, que no conteúdo deste arquivo encontramos os seguintes dados:

a.1) Tabelas e Constantes

Ocupando os 5 primeiros blocos do arquivo, temos todas as tabelas e constantes necessárias as rotinas de compressão e descompressão para a codificação e/ou decodificação de informações.

Carregados para a memória pela rotina de descompressão, estes dados estão distribuídos no ar-

quivo como segue:

- bloco 1 - Tabela de Descrição de Segmentos (TDS) - 512 bytes
  Tabela de Descrição de Campos (TDS) - 512 bytes
- bloco 2 - Padrão de Ausência (PAD) - 512 bytes
  Tabela de Prefixos (UNS) - 512 bytes
- bloco 3 - Tabela de Caracteres (CRC) - 1024 bytes
- bloco 4 e 5 - Tabela de Códigos (CODS) - 2048 bytes

Estas tabelas já foram exaustivamente discutidas em capítulos anteriores (Capítulo III - item 2.3.3, Capítulo IV - item 5.3 - c).

a.2) Dados Compressos

Ocupando os blocos restantes do arquivo, temos todos os dados, componentes do arquivo do usuário, compressos.

2.3. SAÍDAS

a) Como resultado, esta subrotina passa ao programa principal a cada chamada e através de seu único parâmetro, o registro lógico descompresso do arquivo compresso que está sendo processado, até o final do arquivo ser atingido, quando, através deste mesmo parâmetro, passamos um "*" na primeira posição indicando ao programa principal que o fim do arquivo foi atingido.

2.4. ESTRUTURA DO PROGRAMA

2.4.1. Tabelas

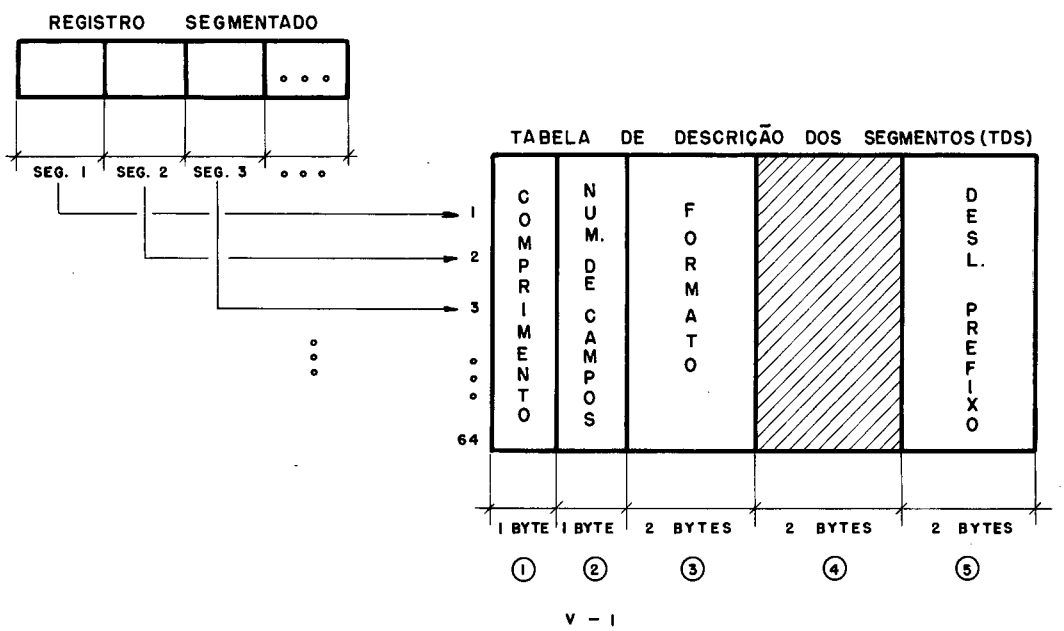
a) Tabela de Descrição dos Segmentos (TDS)

Para um máximo de 64 segmentos, esta tabela reúne, em cada entrada, para a rotina de descompressão todos

os dados descritivos referentes a cada segmento do registro.

Por motivos de projeto do sistema, a organização e acesso à esta tabela deve ser sequencial (na mesma ordem dos segmentos no registro segmentado), visto ser o processamento destes segmentos, quando da compressão e/ou descompressão do registro, feita de forma sequencial.

É interessante relembrar, aqui, as informações, que compõem a Tabela de Descrição dos Segmentos:



V - 1

1. Comprimento do Segmento - 1

Com esta informação é feita toda e qualquer comparação ou movimento envolvendo o segmento descompresso.

2. Número de Campos do Segmento

Com base neste dado controlamos o número de entradas que deverão ser pesquisadas na Tabela de Descrição de Campos (TDC), de modo a permitir a extração e o posterior armazenamento no registro original de cada um dos campos componentes do segmento em questão.

3. Formato do Segmento

Para a rotina de descompressão, este dado tem

por finalidade fornecer o formato do segmento, para determinar a necessidade ou não de compactá-lo.

4. Não utilizado na rotina de descompressão.

5. Deslocamento para a Tabela de Prefixos

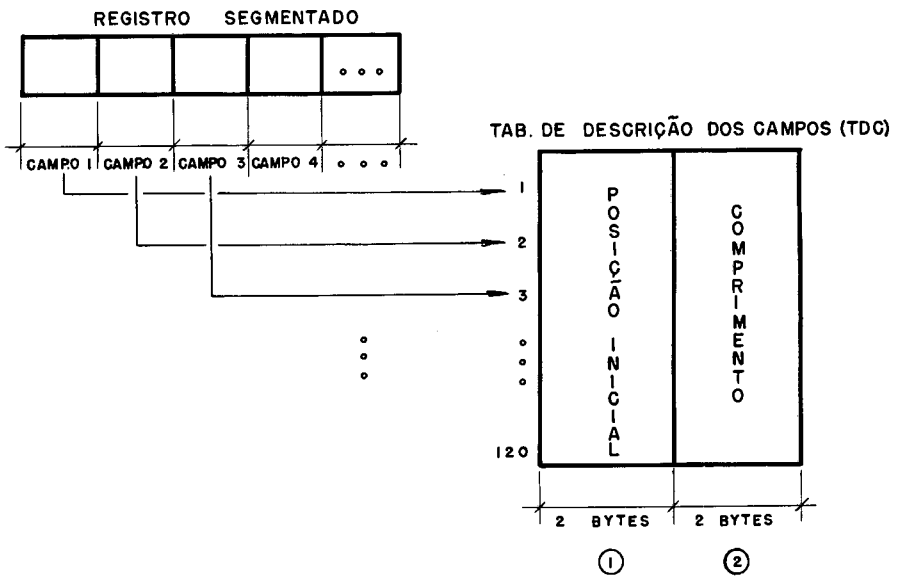
Aqui encontramos o deslocamento, em bytes, que, adicionado ao endereço base da Tabela de Prefixos, nos fornecerá o endereço efetivo, nesta tabela, do conjunto de dados referente ao segmento que está sendo descompresso.

b) Tabela de Descrição dos Campos (TDC)

Para um máximo de 120 campos, esta tabela reúne, em cada entrada, para a rotina de descompressão, todos os dados descritivos referentes a cada campo do registro.

Com base nos dados desta tabela, a rotina de descompressão, extrai campo a campo de cada segmento descompresso, armazenando-os, posteriormente, em suas devidas localizações no registro original. Para isto, esta tabela apresenta as suas entradas organizadas de forma sequencial e na mesma ordem dos campos no registro segmentado. Estas entradas, durante o processamento, deverão ser acessadas sequencialmente tantas vezes quantos sejam os campos no segmento, obtido da Tabela de Descrição dos Segmentos.

Podemos apresentar a Tabela de Descrição dos Campos, como segue:



V - 2

1. Endereço Relativo do Campo

Aqui obtemos o endereço relativo do campo no registro original. Com este dado, e uma vez obtido, o campo do segmento descompresso armazenamos o mesmo no registro.

2. Comprimento do Campo - 1

Com esta informação é feita toda e qualquer comparação ou movimento envolvendo o campo descompresso.

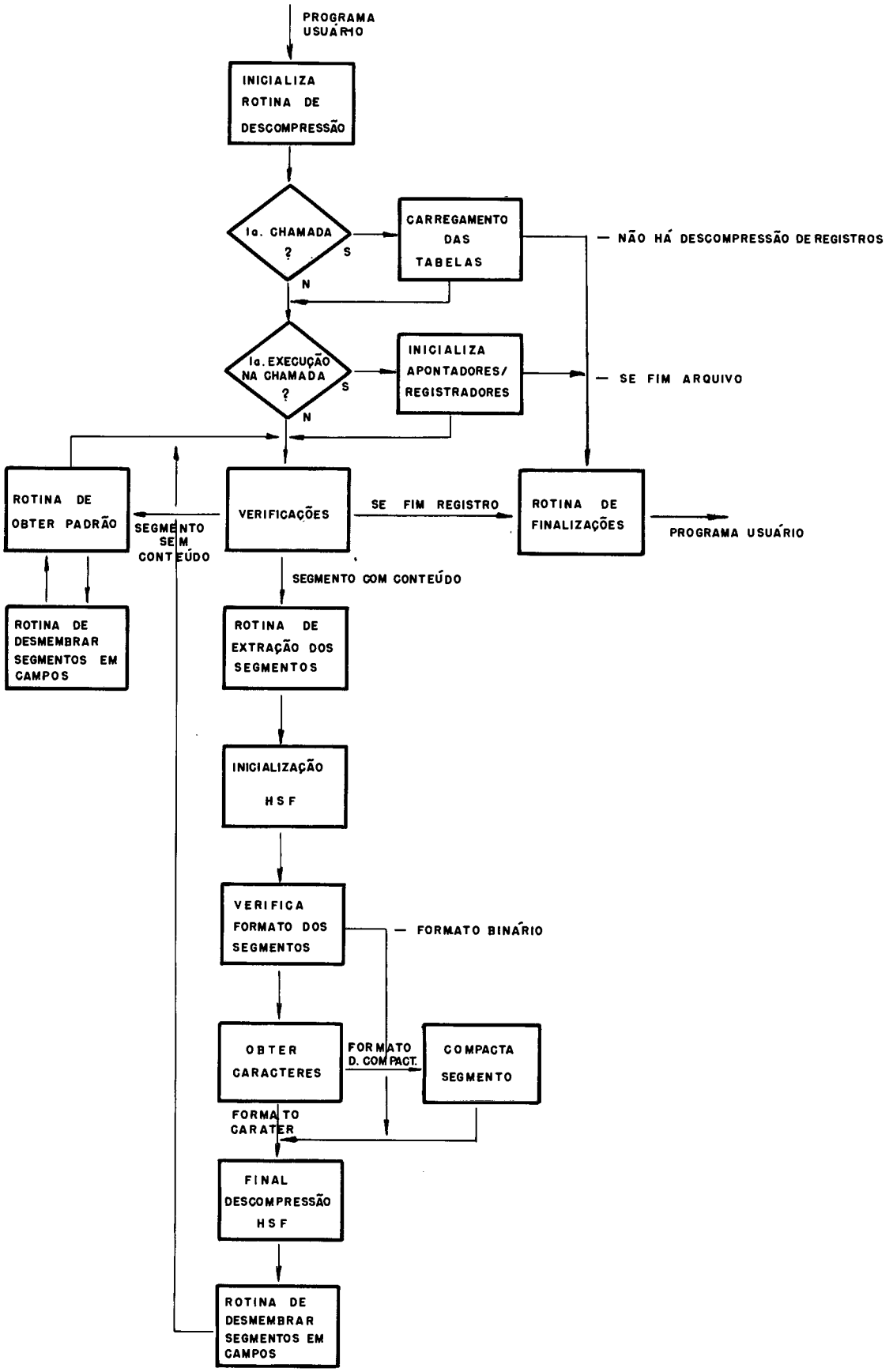
c) Padrão de Ausência(PAD)

Para um máximo de 512 bytes o Padrão de Ausência representa um registro segmentado sem conteúdo. Tem por finalidade, através de comparações, determinar se um dado segmento está ou não sem conteúdo, assim como, quando da descompressão de registros, restaurar segmentos eliminados.

d) Tabela de Prefixos/Tabela de Caracteres

Já discutidas (Capítulo III - itens 2.3.3 - c e 2.3.3 - d).

2.4.2. Fluxograma Geral



2.4.3. Procedimentos

Objetivando um maior aproveitamento para o leitor deste trabalho, passamos a descrever, detalhadamente, cada um dos módulos definidos na subrotina (representados no fluxograma anterior) assim como suas conexões.

Mostramos, a seguir os procedimentos que compõem a descompressão.

a) Módulo 'Inicializa Rotina de Descompressão - DCOMP'

Chamada para cada registro a ser descompresso pelo programa do usuário, através do comando

CALL 'DCOMP' USING parâmetro.

a subrotina de descompressão inicia a sua execução:

- Salvando os registradores do programa principal.
- Carregando os registradores base.
- Restaurando o conteúdo dos registradores de controle da área de entrada (PIO, OVFL).
- Carregando, em um registrador (PREG), o endereço da posição inicial do único parâmetro da subrotina.

b) Módulo'Carregamento das Tabelas'

Como o próprio nome sugere, este módulo é responsável pelo carregamento, para a memória, das Tabelas que se encontram nos primeiros 5 blocos do arquivo.

Pelas suas características este módulo deve ser executado apenas 1 vez, quando da primeira chamada da subrotina de descompressão, o que deverá sempre ocorrer, tendo em vista que a execução de qualquer uma das rotinas está condicionada à utilização das tabelas aqui carregadas. Observamos, que no caso da simples compressão de um arquivo, o programa do usuário deverá chamar esta rotina e passar, através do primeiro byte do parâmetro, uma marca('/'), indicando não haver dados a descomprimir e que esta chamada foi feita unica e exclusivamente para carregamento das tabelas. Este é exatamente o caso da conversão do arquivo

do usuário, conforme podemos observar no programa de conversão (COMPØ5Ø - anexo).

c) Módulo 'Inicializa Apontadores/Registradores'

Este módulo, executado 1 vez a cada chamada desta rotina, tem por função:

- Verificar o fim de arquivo, o que é feito pela análise do conteúdo do próximo byte da área de entrada. Se for encontrado o valor X'ØØ' fim de arquivo foi atingido e, o módulo "ROTINA DE FINALIZAÇÕES" deverá ser executado.
- Se fim do arquivo não foi detectado, carregar os apontadores PTDC, PTDS, PPAD com o endereço das Tabelas de Descrição dos Campos (TDC), Descrição dos Segmentos (TDS), Padrão de Ausência (PAD), zerando os registradores de trabalho NCMP e W3.

d) Módulo de 'Verificações'

Este módulo procede 2 verificações, que são:

- Se o número de campos obtido da Tabela de Descrição dos Segmentos para o próximo segmento a ser analisado for zero (X'ØØ'), o fim de registro foi atingido e neste caso desviamos para executar o módulo "ROTINA DE FINALIZAÇÕES".
- Se no entanto, o fim de registro não foi detectado, e o bit Ø, do próximo byte na área de entrada, não está ligado, estamos diante de um byte com tamanho do segmento. Esta situação indica que o próximo segmento do registro apresenta conteúdo e desta forma devemos executar o módulo "ROTINA DE EXTRAÇÃO DOS SEGMENTOS. Caso contrário, o byte em análise é o Sentinela, e deste modo devemos executar o módulo "ROTINA DE OBTER PADRÃO" o qual é responsável pela recuperação dos segmentos sem conteúdo.

e) Módulo 'Rotina de Obter Padrão'

Antes de entrarmos na descrição deste módulo, devemos lembrar que todo processamento na compressão/des-

compressão de informações está fundamentado no fato de que, a cada registro, os segmentos devem ser processados sequencialmente da esquerda para a direita.

Este módulo tem por finalidade recuperar todos os segmentos sem conteúdo eliminados quando da Compressão por Segmentação.

Assim ao ser executado, este módulo:

e.1) Obtem, a partir do Byte Sentinela, o número de segmentos eliminados, ou seja, quais os próximos segmentos a serem recuperados.

e.2) Com o tamanho obtido da Tabela de Descrição de Segmentos e o Padrão de Ausência, o qual representa um registro com todos os segmentos sem conteúdo, recuperamos em uma área reservada (SEGD) o segmento desejado.

e.3) Atualiza o apontador PPAD com o endereço referente ao início do Padrão de Ausência do próximo segmento.

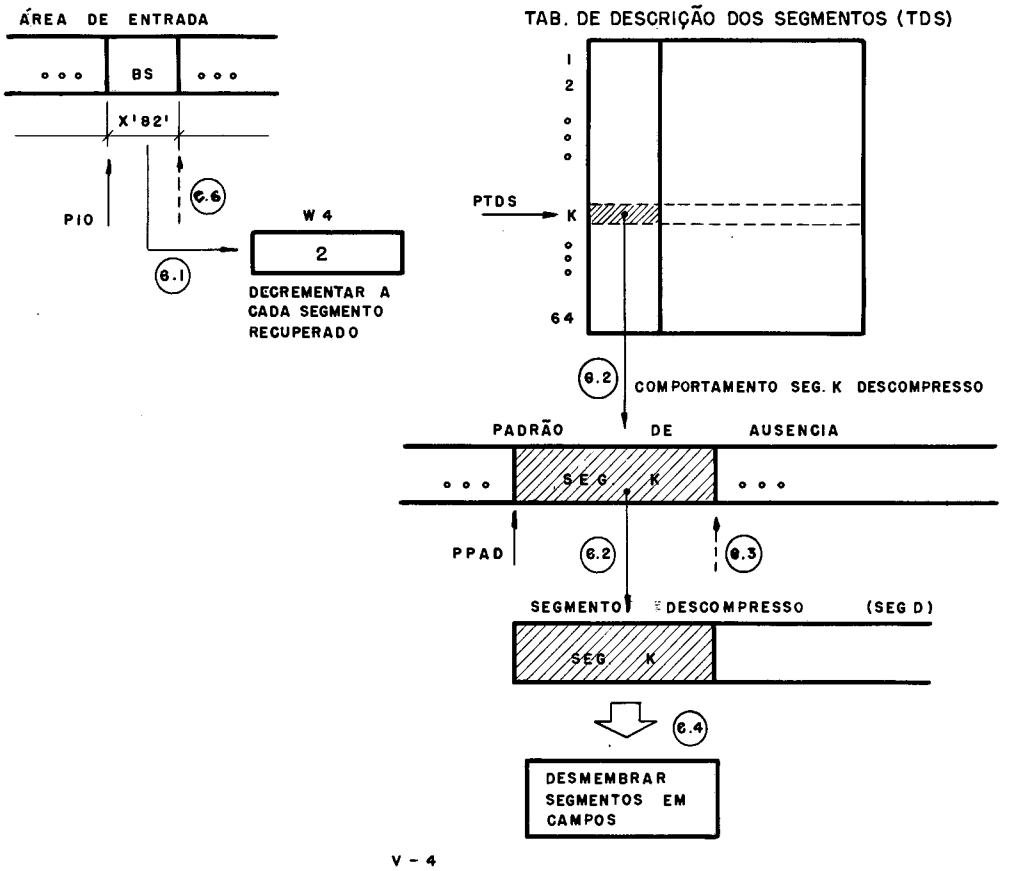
e.4) Executa o módulo "ROTINA DE DESMEMBRAR SEGMENTOS EM CAMPOS", o qual decompõe o segmento em campos armazenando-os em suas localizações originais no registro descompresso.

e.5) Se todos os segmentos indicados no Byte Sentinela não foram recuperados voltar ao procedimento "e.2".

e.6) Caso contrário, incrementa de 1 os registradores de controle da área de entrada (PIO, OVFL).

e.7) Verifica se a área de entrada foi toda processada; Se afirmativo obtem o próximo bloco do arquivo compresso, inicializa o registrador de controle da área de entrada (OVFL) e o registrador de trabalho W3 com zeros.

e.8) Desviar para o módulo de "VERIFICAÇÕES" (item "d").



f) Módulo'Rotina de Extração dos Segmentos'

Uma vez constatado que o segmento a ser descompresso possui conteúdo, passamos a execução deste módulo, que é o responsável pela obtenção do segmento compresso, da área de entrada, para uma área reservada (SEGC).

Ao ser executado, este módulo:

f.1) Obtem do próximo byte na área de entrada o tamanho do segmento compresso.

f.2) Obtem da Tabela de Descrição dos Segmentos o tamanho do segmento descompresso e atualiza o apontador do Padrão de Ausência com o endereço do próximo segmento.

f.3) Incrementa de 1 o registrador de controle da área de entrada OVFL.

f.4) Inicializa umregistrador de trabalho W2 e o apontador PSEGC com, respectivamente, o tamanho da área de entrada (1024 bytes) e o endereço da área reservada

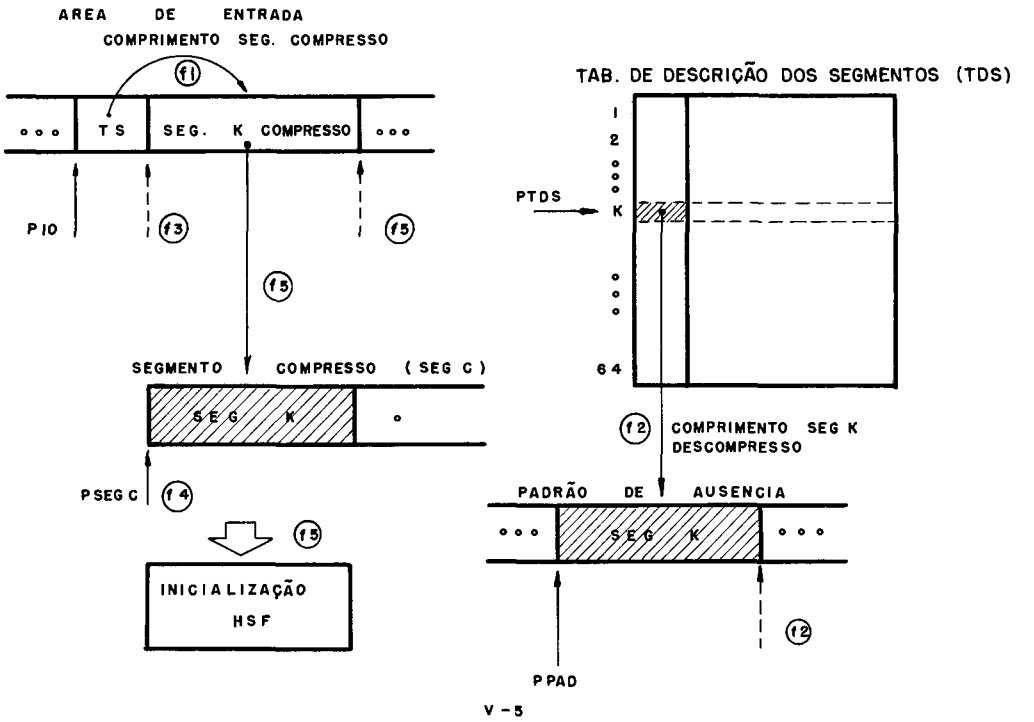
para o segmento compresso (SEGC).

f.5) Se o segmento compresso está totalmente contido na área de entrada:

- Com o tamanho compresso, obtido do Byte de Tamanho (TS), transferir, da área de entrada para a área reservada, o segmento compresso.
- Com o comprimento do segmento compresso, atualizar o apontador de controle da área de entrada PIO, com o endereço do próximo segmento compresso.
- Verificar se a área de entrada, foi completamente processada e, caso afirmativo, obter o próximo bloco do arquivo compresso inicializando os apontadores de controle.
- Desviar para o módulo "INICIALIZAÇÃO HSF" (item "g").

f.6) Se o segmento compresso esta parcialmente contido na área de entrada:

- Determinar quantos bytes do segmento excederam a capacidade da área de entrada (1024 bytes).
- Com o tamanho obtido do item anterior, transferir, da área de entrada para a área reservada (SEGC), a primeira parte do segmento compresso.
- Obter próximo bloco do arquivo compresso.
- Com o comprimento da primeira parte do segmento compresso. Atualizar o apontador da área reservada PSEGC, com o endereço inicial para carregamento da 2a. parte do segmento compresso.
- Executar o item "f.5", considerando como segmento compresso a 2a. parte a ser transferida do atual segmento.



g) Módulo de 'Inicialização HSF'

Este módulo prepara todo o ambiente para a decodificação do segmento, pela utilização do código HSF.

Este módulo ao ser executado:

g.1) Salva registradores de utilização do restante da rotina.

g.2) Carrega os apontadores PSEGC e PSEGD com os endereços das áreas reservadas aos segmentos compresso (SEGC) e descompresso (SEGD), zerando CENT e W1.

g.3) Carrega os registradores base BUNS e BCRC, a partir das Tabelas de Descrição dos Segmentos e de Prefixos, com os endereços base das Tabelas de Prefixo(UNS) e Caracteres (CRC) referentes aos dados do segmento a ser decodificado.

h) Módulo'Verifica Formato dos Segmentos'

A função deste módulo é determinar o formato do segmento, pois como sabemos o tratamento dispensado a cada segmento, por ocasião de sua decodificação, depende deste dado, como veremos a seguir:

- Segmentos de formato alfanumérico, alfabético e deci

cimal zonado a codificação/decodificação, utilizando o código de Tamanho variável HSF, é feita tomando o conteúdo byte a byte.

- Segmentos de formato Decimal Compactado, a codificação/decodificação, utilizando o código HSF, é feita com os caracteres encontrados de 4 em 4 bits.
- Segmentos de formato binário não são compressos utilizando o código de tamanho variável HSF.

Ao ser executado, este módulo:

h.1) Compara o código de formato obtido da Tabela de Descrição de Segmentos, correspondente ao segmento em questão, com o código de formato - Decimal Compactado - 768.

h.2) Se o código de formato do segmento for menor que 768 o segmento que está sendo decodificado tem o formato carater e neste caso, após a decodificação, não precisa ser compactado. Executar o módulo "OBTER CARACTERES" (item "i").

h.3) Se o código de formato do segmento for igual a 768, o segmento que está sendo decodificado tem o formato Decimal Compactado e neste caso, após a decodificação, necessita ser compactado para obtenção do segmento original. Executar o módulo "OBTER CARACTERES" (item "i").

h.4) Se o código de formato do segmento for maior que 768 o segmento em questão tem o formato binário e neste caso não esta codificado. Assim, com o comprimento do segmento armazenado no seu 1º byte, transferimos o segmento da área reservada ao segmento compresso para a área reservada ao segmento descompresso, desviando, em seguida, para o módulo "FINAL DESCOMPRESSÃO HSF" (item "l").

i) Módulo 'Obter Caracteres'

É o principal módulo de descompressão. Tem por finalidade utilizando a metodologia de descompressão de-

senvolvida para a utilização do código HSF, (apresentada no capítulo III), determinar, para cada grupo do código HSF encontrado no segmento compresso, o caráter fonte correspondente ao mesmo. Ao final, como resultado da execução deste módulo, obtemos o segmento descompresso.

Ao ser executado, este módulo:

i.1) Se o formato do segmento é decimal compactado:

- Altera controles de desvio, no final deste módulo para permitir a execução do módulo seguinte "COMPACTA SEGMENTO" (item "j").
- A partir da Tabela de Descrição dos Segmentos obtem e duplica o comprimento descompresso do segmento a ser decodificado, armazenando-o em um registrador de trabalho (W1).
- Desvia para o item "i.3".

i.2) Se formato do segmento é caráter:

- Coloca no registrador de trabalho W1 o comprimento descompresso do segmento a ser decodificado.

i.3) A partir do comprimento descompresso do segmento(W1) posiciona o apontador de controle de fim do segmento descompresso TMSEG com o endereço do último byte deste.

i.4) Inicializa os registradores incremento INSEGD e de Trabalho W1 com as constantes 1 e Ø.

i.5) Carrega no registrador de entrada ENT os primeiros 4 bytes do segmento compresso.

i.6) Inicializa os registradores de trabalho W3 e SAI com Ø.

i.7) A filosofia aqui empregada, para controle de fim da decodificação tem por base o endereço final do segmento descompresso, ou seja a decodificação será feita até que o segmento descompresso esteja completo. Em função disto decidimos utilizar um comando (BXH) para controle do ciclo e que, a cada caráter fonte de-

codificado, incrementa o apontador PSEGD de armazenamento do próximo caráter fonte na área reservada ao segmento descompresso SEGD.

Em seguida este comando verifica se o endereço no apontador PSEGD é superior ao endereço final do segmento descompresso no apontador de fim TMSEG.

Caso afirmativo é feito um desvio para os controles no final deste módulo quando, dependendo do formato do segmento, é feito um desvio para o módulo "COMPACTA SEGMENTO" ou "FINAL DESCOMPRESSÃO HSF".

i.8) Eliminar o prefixo do grupo em decodificação, deslocando para a esquerda bit a bit do registrador de entrada ENT, adicionando por cada bit deslocado, 2 ao registrador de trabalho W3 e 1 ao registrador de trabalho W1 (controla número de bits processados no registrador de entrada), até não ocorrer overflow.

Com a execução deste procedimento processamos todo o prefixo do grupo, determinando no registrador de trabalho W3 o deslocamento para acessar,na Tabela de Prefixos, os dados referentes a este prefixo para o segmento em decodificação.

i.9) No procedimento anterior, consideramos o não overflow no registrador de entrada ENT, como o fim de prefixo ou fim de dados no registrador, ou seja, quando todos os 32 bits já foram deslocados. Neste último caso alteramos o endereço no apontador dos dados no segmento compresso (PSEGC), 4 bytes à frente, inicializamos o registrador de trabalho W1 com 1, carregamos no registrador de entrada ENT os próximos 4 bytes, voltando, em seguida, ao procedimento anterior, para dar continuidade ao processamento do prefixo. Porém se o não overflow foi proveniente do fim de prefixo, executar o procedimento "i.10".

i.10) Caso o prefixo do grupo tenha sido processado e o registrador de entrada ENT ainda contenha dados,pas

samos a determinação do caráter fonte.

Para isto:

1. Obtemos da Tabela de Prefixos o comprimento do maior sufixo dos grupos para o prefixo decodificado.

2. Verificamos, através do registrador de trabalho W1,se o registrador de entrada contem todos os bits do maior sufixo.

3. Se não desviar para o procedimento 8, onde através de 2 passagens obtemos o sufixo.

4. Caso contrário, deslocar, para o registrador de saída SAI, estes bits. Daí, obtemos o deslocamento que determina na Tabela de Caracteres,para os grupos expandidos com este prefixo, os dados desejados.

5. Obter da Tabela de Prefixos o 2º deslocamento que determina, para os grupos expandidos do segmento em decodificação, o início dos grupos com o prefixo em questão.

6. Adicionando os deslocamentos, obtidos, nos procedimentos 4 e 5, alcançamos o deslocamento total que juntamente com o endereço base da Subtabela de Caracteres, determinado no item "g.3" nos dá o endereço efetivo dos dados desejados.

7. De posse do endereço efetivo conseguido no procedimento 6, obtemos o caráter fonte, armazenando o na área reservada ao segmento descompresso(SEGD), obtemos o número de bits a retornar para o registrador de entrada, por não pertencerem ao grupo decodificado, retornamos estes bits; atualizamos o registrador de controle W1 e voltamos ao procedimento "i.6" para processamento do grupo seguinte.

8. Como o registrador de entrada ENT não contem todos os bits do maior sufixo do grupo procedemos

a 2 passagens (etapas) para que se complete sua obtenção.

9. A partir do registrador de controle W1 e do tamanho do sufixo, determinar e deslocar todos os bits que restam do registrador de entrada ENT para o registrador de saída SAI.

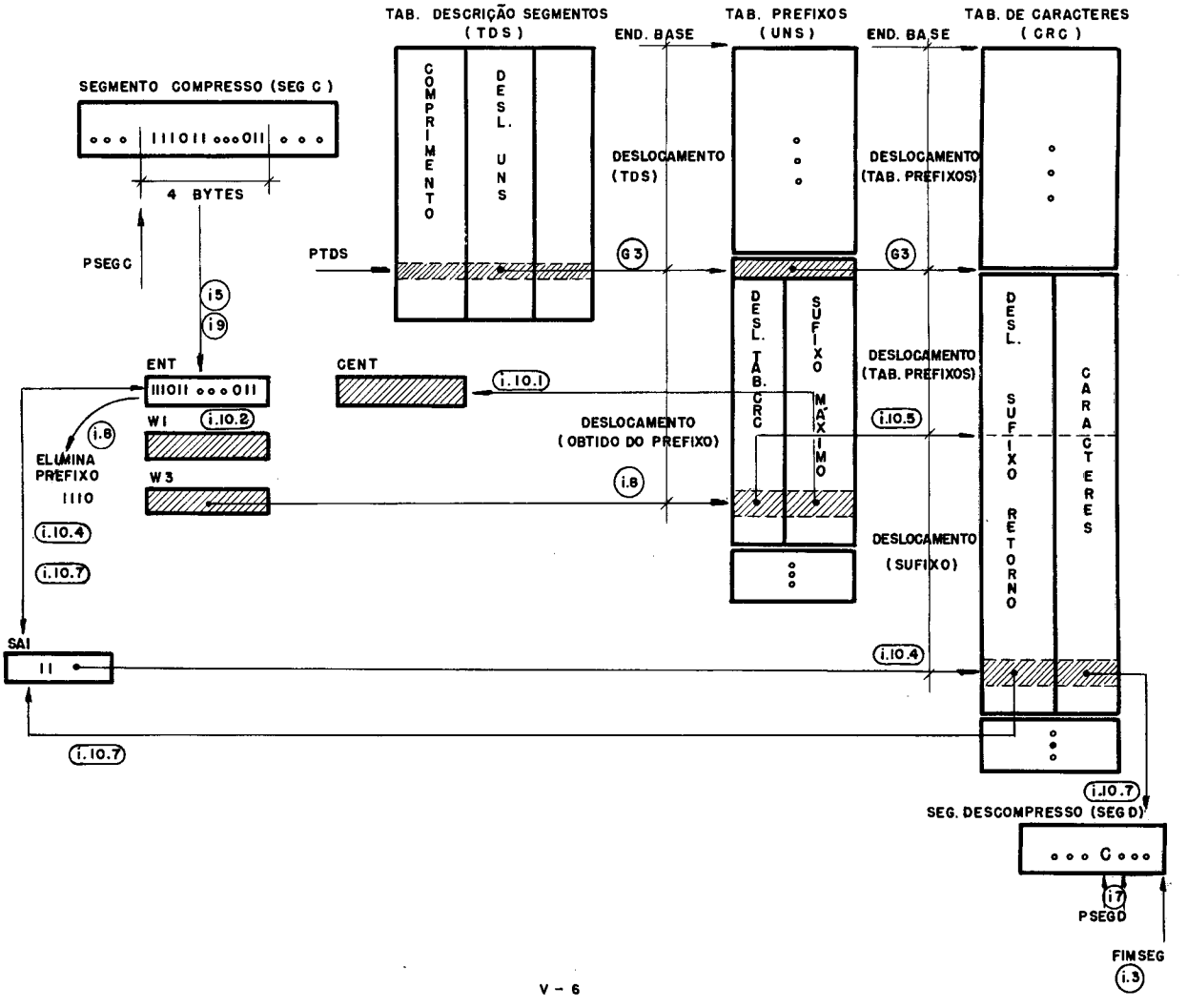
10. Carregar no registrador de entrada os próximos 4 bytes do segmento compresso (SEGC).

11. A partir do registrador de controle W1 determinar e deslocar os bits restantes do sufixo para o registrador de saída SAI.

12. Neste ponto, uma vez já obtido o sufixo, procedemos da mesma forma que nos procedimentos 4, 5, 6 e 7, para obtenção do caráter fonte e do deslocamento de retorno(bits do sufixo a retornar do registrador de saída SAI para o registrador de entrada ENT).

13. Se o deslocamento de retorno for tal, que implique no retorno de algum bit dos 4 bytes anteriores do registrador de entrada ENT, desviar para o procedimento 14. Caso contrário, simplesmente, fazemos retornar de SAI para ENT, os bits desejados, atualizamos o apontador de dados no segmento compresso PSEGC, atualizamos o registrador de controle W1 e voltamos ao procedimento "i.6" para processamento do grupo seguinte.

14. Neste caso como o conteúdo anterior do registrador de entrada ENT não foi totalmente processado (a indicação é dada pelos bits de retorno) determinamos e eliminamos através de deslocamento para a direita, todos os bits pertencentes aos últimos 4 bytes obtidos de SEGC, por não ter sido ainda completado o processamento dos 4 bytes anteriores; retornamos de SAI para ENT os bits desejados; atualizamos o registro de controle W1 e voltamos ao procedimento "i.6" para processamento do grupo seguinte.



j) Módulo 'Compacta Segmento'

Para segmentos de formato Decimal Compactado este módulo deve ser executado, uma vez que o resultado proveniente do módulo de decodificação "OBTER CARACTERES", está sempre no formato caráter, ou seja um byte para cada caracter do segmento compactado, inclusive o sinal.

Como sabemos, a instrução de compactação PACK, compacta de cada vez apenas um máximo de 15 bytes e como os nossos segmentos terão comprimentos superiores a esse máximo, desenvolvemosuma metodologia que compacta qualquer segmento em etapas de 15 bytes. Observamos que a compactação é feita na própria área em que se encontra o segmento compresso (SEGC).

Assim, ao ser executado, este módulo:

j.1) Carrega os apontadores PSEGD e PSEGC com o endereço da área reservada ao segmento decodificado anteriormente pelo método HSF - SEGD e SEGC.

j.2) Obtem, da Tabela de Descrição dos Segmentos, o comprimento real do segmento compactado - 1 incrementando-o em seguida de 2. Com isto forçamos o aumento de um Byte no comprimento do segmento compactado o que nos permitirá obter o último byte de sinal do segmento, os quais se encontram em bytes separados (dígito - sinal).

j.3) Inicializa o registrador incremento NCMP com -7 e o registrador de controle TMSEG com Ø.

j.4) Executa o comando de controle do ciclo (BXLE) o qual, a cada ciclo de compactação, decrementa o comprimento real do segmento compactado no registrador de trabalho W1 de 7. Verifica em seguida, se o valor em W1 ficou ≤ Ø, o que, em caso afirmativo, indicará termos alcançado a última parte do segmento a ser compactado, momento em que devemos desviar para o procedimento "j.6".

j.5) Para os ciclos normais, de compactação como vimos anteriormente, compactamos a informação de 15 em 15 bytes, a menos da parte final do segmento.

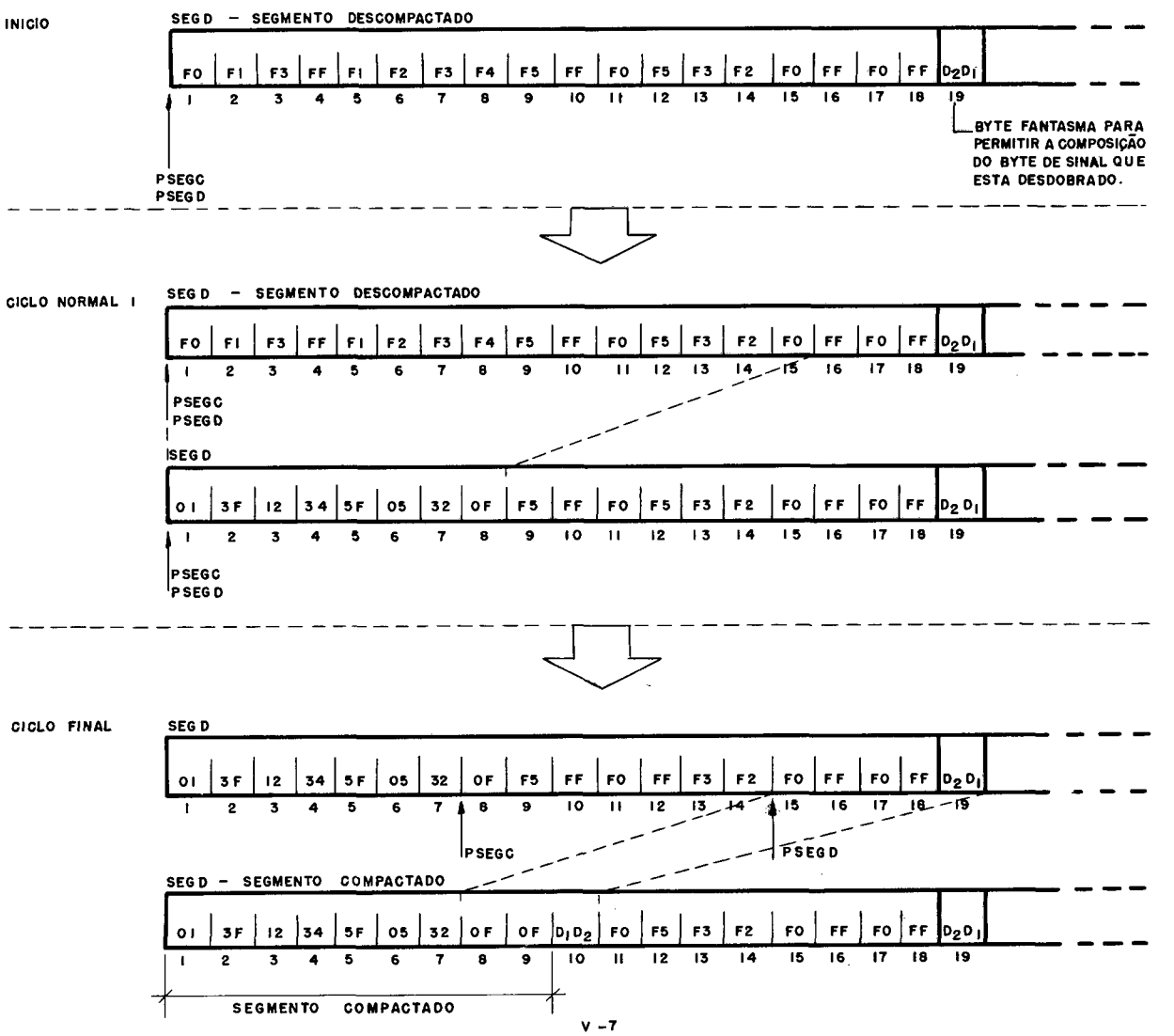
Assim, este procedimento compacta os próximos 15 bytes do segmento original, atualiza os apontadores PSEGC e PSEGD com os endereços iniciais dos próximos 8 bytes compactados e dos próximos 15 bytes a serem compactados.

Como sabemos, ao se compactar um dado, o último byte, sofre apenas a troca de posição dígito-sinal. Assim, para as etapas intermediárias de compactação devemos ter o cuidado de eliminar a cada etapa de compactação o último byte e novamente submetê-lo à etapa seguinte, tanto do segmento descompactado, como do compactado.

j.6) Para compactar a última parte do segmento:

- Determinar a partir do registrador de trabalho W1 o comprimento compactado da parte final.
- Determinar a partir do registrador de trabalho W1 o comprimento descompactado da parte final.
- Formatar e alterar o byte de comprimento na instrução de compactação (PACK).
- Compactar parte final do segmento.

Para melhor compreensão do processo, ilustramos a seguir, a compactação de um segmento com 19 bytes de comprimento(incluindo 1 Byte Fantasma para permitir a composição do Byte de Sinal).



l) Módulo 'Final Descompressão - HSF'

Com este módulo encerramos a parte da rotina responsável pela descompressão utilizando o Código de Tamanho Variável - HSF.

Ao ser executado, este módulo:

- Restaura o conteúdo dos registradores, salvos na inicialização, para utilização no restante da rotina.
- Carrega, no registrador de trabalho W1, o endereço de entrada no módulo de "VERIFICAÇÕES".

m) Módulo 'Rotina de Desmembrar Segmentos em Campos'

Uma vez que estamos com o nosso segmento descompresso e que o mesmo pode reunir um ou mais campos do registro, resta-nos, para encerrar a rotina, armazenar cada um destes campos em suas localizações originais no registro.

Assim, ao ser executado, este módulo:

m.1) Carrega no apontador PSEGD o endereço da área onde se encontra o segmento descompresso não compactado.

m.2) Carrega no registrador NCMP, a partir da Tabela de Descrição dos Segmentos (TDS), o número de campos do segmento.

m.3) Devido ao fato de que na Tabela de Descrição de Campos, cada entrada correspondente à descrição de um campo, e que estas por sua vez se encontram organizadas na mesma sequência dos campos no registro segmentado, para se processar campo a campo de cada segmento do registro, bastará processar esta tabela sequencialmente.

Desta forma, uma vez que o apontador da Tabela de Descrição de Campos (TDC) já está posicionado na entrada referente ao primeiro campo do segmento em questão (posicionado no processamento do segmento anterior), obter da TDC o endereço relativo do campo no registro original carregando-o no registra-

dor de trabalho W2.

m.4) Obtem o endereço efetivo do campo no registro original, pela adição do endereço relativo deste campo ao endereço base da área reservada ao registro, colocando-o no registrador W2.

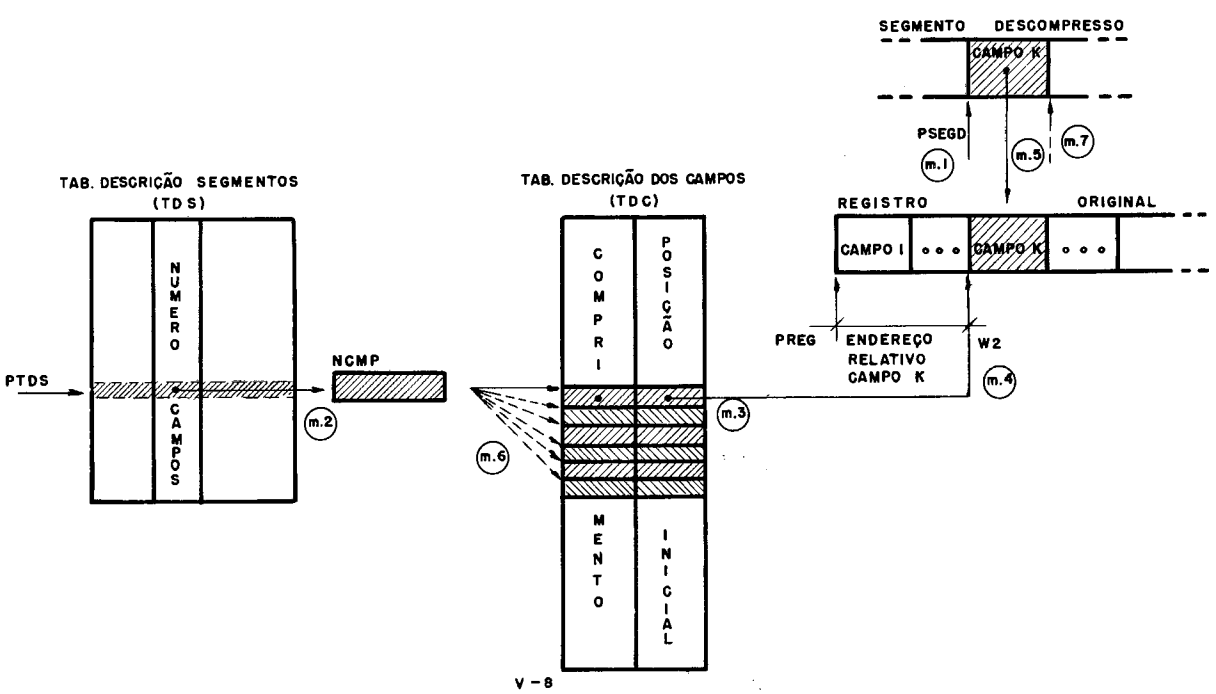
m.5) Obtem, da TDC, o comprimento original do campo armazenando-o em W3, e a partir dele e do endereço efetivo obtido no item anterior, mover o campo do segmento descompresso (SEGD), para o registro original.

m.6) Posiciona o apontador PTDC, na próxima entrada na Tabela de Descrição de Campos (TDC).

m.7) Posiciona o apontador PSEGD no próximo campo do segmento descompresso (SEGD).

m.8) Se todos os campos do segmento não foram processados voltar ao procedimento "m.3".

m.9) Caso contrário, posiciona PTDS na entrada da TDS referente ao próximo segmento a ser descompresso e desvia para o endereço colocado anteriormente em W1, que pode ser dependendo da ocasião, o do módulo "VERIFICAÇÕES" ou o da "ROTINA OBTER PADRÃO".



n) Módulo Rotina de Finalizações

Este módulo encerra a rotina de descompressão e para isto:

n.1) Fecha o arquivo de entrada ARQE se fim de arquivo.

n.2) Move para a la. posição do parâmetro a marca '*' indicando no programa do usuário se fim de arquivo.

n.3) Salva os apontadores de controle da área de entrada (PIO, OVFL) na área denominada SAVENT.

n.4) Restaura o conteúdo dos registradores para o programa principal.

n.5) Retorna ao programa principal.

2.5. MODIFICAÇÕES

a) Caso o usuário deseje mudar o suporte de armazenamento deverá criar para a subrotina uma nova descrição do arquivo (DTF), tendo cuidado, porém, de não mudar os parâmetros abaixo:

- Blocos de tamanho fixo igual a 1024 bytes.
- Label standard IBM.
- 2 áreas de entrada.
- Processamento na própria área de entrada.

b) Para qualquer alteração no suporte armazenamento do arquivo, o usuário deverá ter o cuidado de verificar nesta rotina a compatibilidade dos comandos de leitura, abertura e fechamento do arquivo.

c) Para alterações envolvendo tamanho de tabelas, registros, bloco do arquivo compresso, etc, o usuário deverá ter muito cuidado e efetuar um estudo global no sistema, uma vez que estes dados influenciam de forma direta todo o sistema.

3. ROTINA DE COMPRESSÃO

3.1. FINALIDADE

Esta parte da subrotina, denominada de 'COMP', é responsável pela compressão efetiva dos dados.

Através dela o programa do usuário, a partir de registros descompressos, gera o arquivo compresso, ou seja, a cada chamada a subrotina recebe do programa chamador o registro original,comprime segmento a segmento, executa todos os controles, gravando-os por meio de segmentos.

De um modo geral, ao se comprimir segmento a segmento de um registro, temos de:

a) Construir o segmento, reagrupando campos do registro orinal.

b) Para segmentos sem conteúdo, eliminá-los utilizando o método de Segmentação.

c) Para segmentos com conteúdo, codificá-los utilizando o código HSF.

3.2. ENTRADAS

a) Como entrada, esta subrotina recebe do programa principal a cada chamada, através de seu único parâmetro o registro lógico que deverá ser compresso.

Para que a subrotina de compressão possa efetuar os procedimentos finais ao ser atingido o fim de arquivo, o programa principal passa, através da primeira posição do parâmetro, o caráter '*'.

3.3. SAÍDAS

a) Arquivo Aplicação Compresso

Este é o arquivo do usuário compresso, a ser ge

rado pela subrotina de compressão/descompressão e que apresenta as seguintes características:

- Nome interno - ARQS
- Endereço - SYSØ11
- Label standard IBM
- 2 áreas de saída
- Registros não blocados de tamanho fixo igual a 1024 bytes.

Este arquivo já foi apresentado anteriormente quando da apresentação da rotina de descompressão (item "2.2 - a"). Torna-se, pois, desnecessário repetir todas as suas características.

3.4. ESTRUTURA DO PROGRAMA

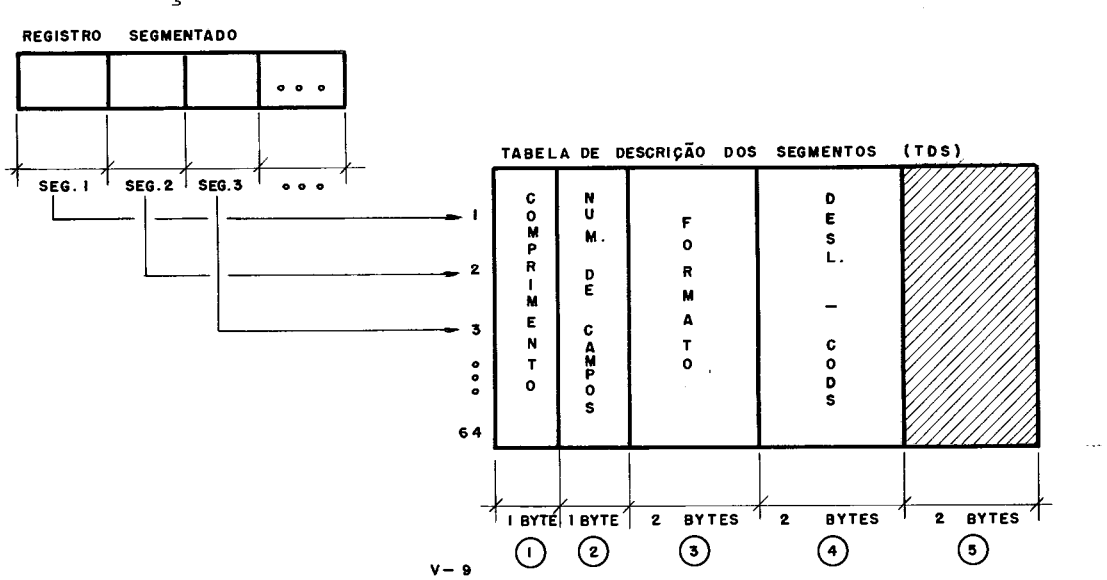
3.4.1. Tabelas

a) Tabela de Descrição dos Segmentos (TDS)

Para um máximo de 64 segmentos, esta tabela reúne em cada entrada, para a rotina de compressão, todos os dados descritivos referentes a cada segmento do registro.

Como já vimos, a organização e acesso à esta tabela é feita de forma sequencial.

Para esta rotina a tabela dispõe das seguintes informações:



1. Comprimento do Segmento - l

Com esta informação, é feita toda e qualquer comparação ou movimento envolvendo o segmento descompresso.

2. Número de Campos do Segmento

Com base neste dado, controla-se o número das entradas que deverão ser pesquisadas na Tabela de Descrição de Campos (TDC), de modo a permitir a extração do registro original de cada um dos campos que irão compor o segmento em questão.

3. Formato do Segmento

Para a rotina de compressão, este dado, além de fornecer o formato do segmento para se determinar a necessidade ou não de se descompactar este segmento antes da codificação, serve também como deslocamento em bytes que, adicionado ao endereço base da Tabela de Tradução, determinará o endereço efetivo da subtabela desejada.

4. Deslocamento para a Tabela de Códigos

Aqui encontramos o deslocamento em bytes que, adicionado ao endereço base da Tabela de Códigos, nos fornecerá o endereço efetivo, nesta tabela, dos grupos do código HSF gerados para o segmento a ser compresso.

5. Não utilizado na rotina de compressão.

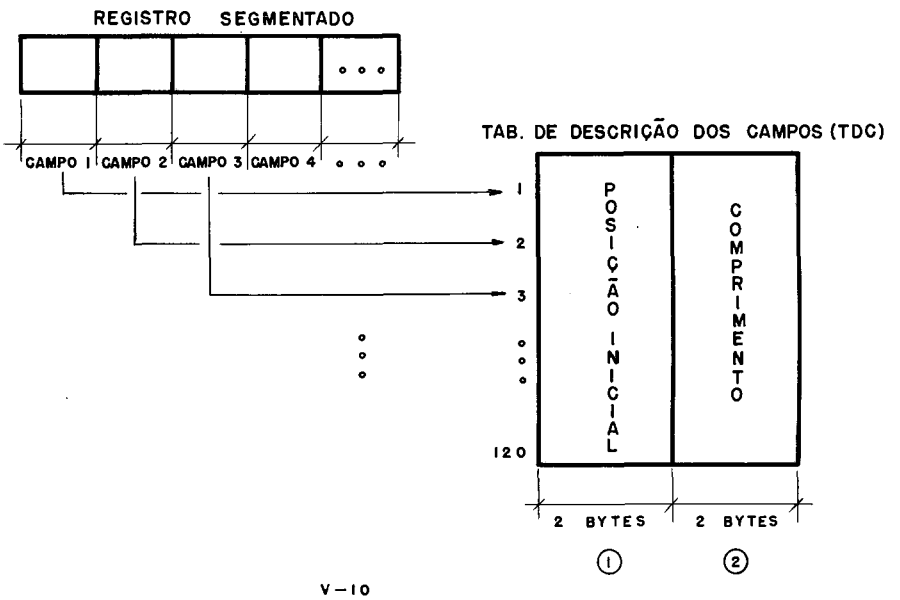
b) Tabela de Descrição dos Campos (TDC)

Para um máximo de 120 campos, esta tabela reúne em cada entrada, para a rotina de compressão, todos os dados descritivos referentes a cada campo.

Com base nos dados desta tabela a rotina de compressão extrai cada campo do registro armazenando-o posteriormente de forma sequencial na área reservada ao segmento descompresso. Para isto, como já foi visto, esta tabela apresenta as suas entradas de forma sequencial,na mes

ma ordem dos campos no registro segmentado.

Podemos apresentar esta tabela, como se segue:



V-10

1. Endereço Relativo do Campo

Aqui obtemos o endereço relativo do campo no registro original. Com este dado adicionado ao endereço base do registro original obtemos o endereço efetivo do campo, que, por meio do comprimento permite a sua extração para o segmento descompresso em composição.

2. Comprimento do Campo - 1

Com esta informação é feita toda e qualquer comparação ou movimento envolvendo o campo descompresso.

c) Padrão de Ausência (PAD)

Para a rotina de compressão, o Padrão de Ausência, por meio de comparações, tem a finalidade de determinar se um dado segmento possui ou não conteúdo.

d) Tabela de Tradução/Tabela de Códigos

Todos os comentários já foram feitos (Capítulo III - itens 2.3.3 - a e 2.3.3 - b).

e) Tabela de Mascaras (MASC)

Esta é uma tabela de utilização exclusiva da rotina de compressão e tem por finalidade fornecer os dados necessários para o armazenamento na área reservada ao segmento compresso dos últimos bytes codificados no registrador de saída.

Como nós iremos ver adiante, ao se comprimir um segmento, o fazemos caráter a caráter, armazenando os grupos correspondentes em um registrador de saída (SAI). Sempre que o registrador de saída fica completo, descarregamos o seu conteúdo na área do segmento compresso (4 em 4 bytes). Porém ao se terminar a codificação, normalmente não teremos todos os 4 bytes do registrador de saída, preenchidos.

Do exposto, esta tabela acessada pelo número de bits usados no registrador de saída fornece os dados necessários para que a rotina armazene no segmento compresso apenas os bytes com conteúdo.

Podemos então, apresentar a tabela como se segue, onde os conteúdos se encontram em Hexadecimal.

TAB. DE MASCARAS (MASC)

MÁSCARA	BYTES TRANSFERIDOS	BITS ALINHAMENTO
0 0	0 0	0 0
8 1	0 1	0 7
8 1	0 1	0 6
8 1	0 1	0 5
8 1	0 1	0 4
8 1	0 1	0 3
8 1	0 1	0 2
8 1	0 1	0 1
8 1	0 1	0 0
8 3	0 2	0 7
8 3	0 2	0 6
8 3	0 2	0 5
8 3	0 2	0 4
8 3	0 2	0 3
8 3	0 2	0 2
8 3	0 2	0 1
8 3	0 2	0 0
8 7	0 3	0 7
8 7	0 3	0 6
8 7	0 3	0 5
8 7	0 3	0 4
8 7	0 3	0 3
8 7	0 3	0 2
8 7	0 3	0 1
8 7	0 3	0 0
8 F	0 4	0 7
8 F	0 4	0 6
8 F	0 4	0 5
8 F	0 4	0 4
8 F	0 4	0 3
8 F	0 4	0 2
8 F	0 4	0 1
	1 BYTE ①	1 BYTE ② 1 BYTE ③

1. Mascara

Este byte, da forma como se apresenta, é utilizado pela rotina, para alterar a instrução STCM, de modo que esta ao ser executada armazene somente os bytes desejados.

2. Número de Bytes Transferidos

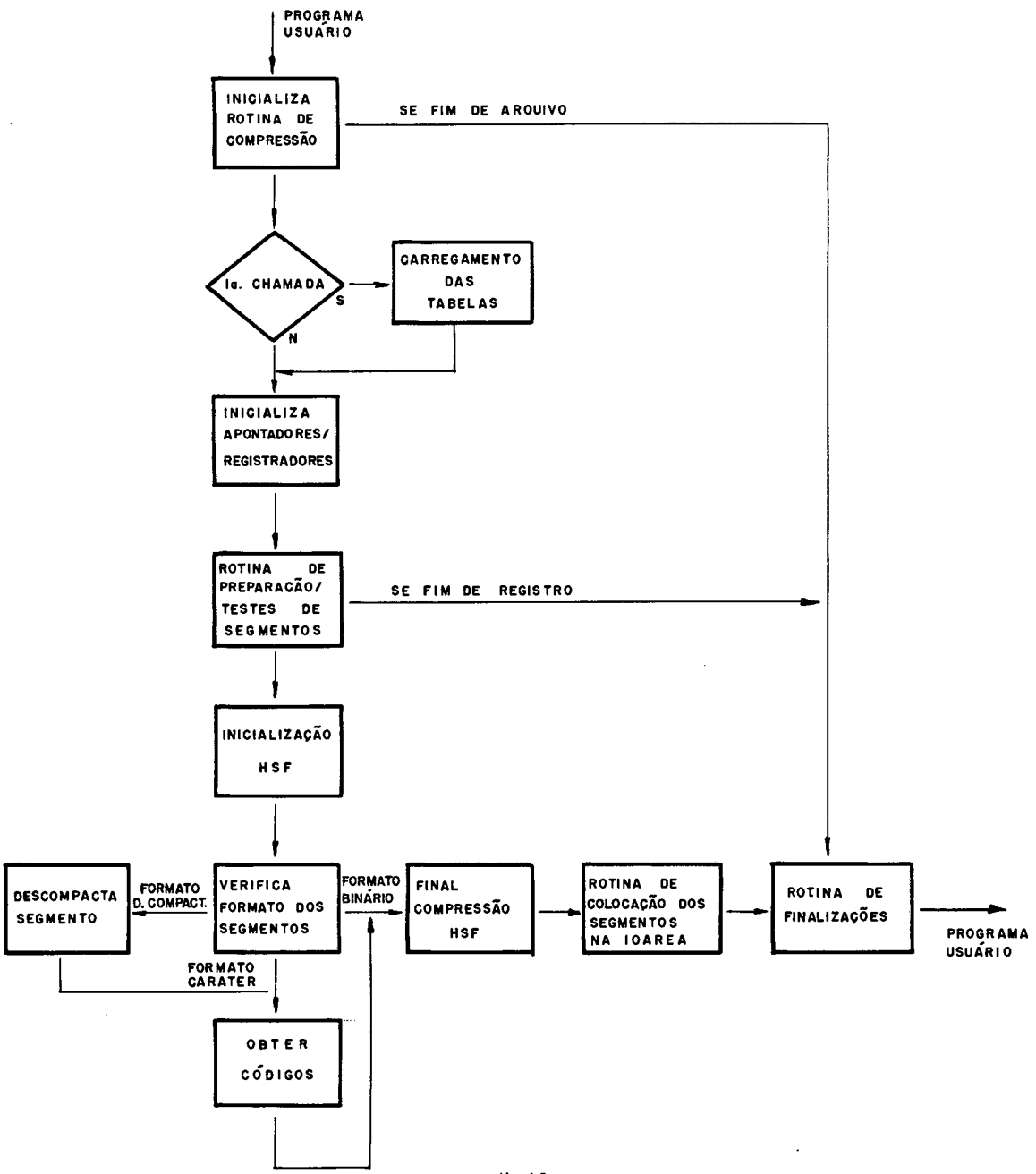
Para computar o comprimento compresso do segmento, fornecemos aqui o número de bytes transferidos para a mascara utilizada.

3. Alinhamento em Byte

Devido a forma de processamento dos grupos, os bits processados se encontram ajustados a direita do registrador de saída (SAI). Porém, como trabalhamos a nível de byte, deveremos alinhar em byte e a esquerda os nossos bits para posterior transferência.

Assim, o dado aqui fornecido, nos dá o número de bits de conformidade com a máscara usada, para alinhamento dos nossos bits para o byte mais próximo a esquerda.

3.4.2. Fluxograma Geral



3.4.3. Procedimentos

Neste item, detalharemos cada um dos módulos componentes da rotina de compressão, assim como todas as suas conexões.

Ao ser executada, a subrotina de compressão reúne os seguintes procedimentos:

a) Módulo 'Inicializa Rotina de Compressão - COMP'

Chamada pelo programa do usuário para cada registro a ser compresso, através do comando

CALL 'COMP' USING parametro.

a subrotina de compressão inicia a sua execução:

- Salvando os registradores do programa principal.
- Carregando os registradores base.
- Restaurando o conteúdo dos apontadores de controle da área de saída (PIO, OVFL).
- Carregando no registrador PREG o endereço da posição inicial do único parâmetro da subrotina.
- Verificando a condição de fim de arquivo (carater '*' na la. posição do parametro).
- Se constatado fim de arquivo, desviar para a execução do módulo "ROTINA DE FINALIZAÇÕES" (item "k").

b) Módulo 'Carregamento das Tabelas'

Como o próprio nome sugere, este é o módulo responsável pela gravação, nos primeiros 5 blocos do arquivo, das tabelas necessárias à execução da subrotina de compressão/descompressão.

Pelas suas características, este módulo deverá ser executado apenas 1 vez, quando da primeira chamada da subrotina compressão.

c) Módulo 'Inicializa Apontadores/Registradores'

Este módulo, executado 1 vez a cada chamada da subrotina, tem por função carregar os apontadores PTDC,

PTDS e PPAD com o endereço inicial das Tabelas de Descrição dos Campos (TDC), Descrição dos Segmentos (TDS) e Padrão de Ausência (PAD), zerando o registrador de trabalho NCMP.

d) Módulo 'Rotina de Preparação/Testes de Segmentos'

Um dos principais desta rotina, este módulo resume o processo de compressão por segmentação através das seguintes funções:

- Obter os segmentos a partir do registro original(passado pelo programa do usuário através do parametro da subrotina).
- Verificar se o segmento construído possui ou não conteúdo.
- Criar e armazenar o Byte Sentinela.

Ao ser executado, este módulo:

d.1) Inicializa o registrador do Byte Sentinela(RBS), com a constante 127.

d.2) Incrementa de 1 o registrador referente ao Byte Sentinela (RBS).

d.3) Carrega o apontador PSEGD com o endereço inicial da área reservada ao segmento descompresso (SEGD) a ser construído.

d.4) Posiciona o apontador PTDS na entrada da Tabela de Descrição de Segmentos (TDS), referente ao segmento a ser construído.

d.5) Obtem da Tabela de Descrição dos Segmentos (TDS) o número de campos componentes do segmento, carregando-o no registrador NCMP.

d.6) Verifica, através do registrador NCMP, se o número de campos é zero.

Se afirmativo, significa que o fim de registro foi atingido e neste caso desviamos para execução do módulo "ROTINA DE FINALIZAÇÕES" (item "k").

d.7) Uma vez que o apontador da Tabela de Descrição de Campos (TDC) está posicionado na entrada referente ao primeiro campo deste segmento, obtem endereço relativo do campo no registro original, armazenando-o no registrador de trabalho W2.

d.8) Obtem o endereço efetivo do campo pela adição do endereço relativo deste campo ao endereço base do registro original, armazenando-o no registrador de trabalho W2.

d.9) Obtem, da Tabela de Descrição de Campos (TDC), o comprimento original do campo armazenando-o no registrador de trabalho W1 e, a partir dele, e do endereço efetivo do item anterior move o campo do registro original para a área reservada ao segmento descompresso (SEGD).

d.10) Posiciona o apontador PTDC na próxima entrada na Tabela de Descrição de Campos (TDC).

d.11) Posiciona o apontador PSEGD para endereçar o próximo campo a ser colocado no segmento (SEGD).

d.12) Se todos os campos do segmento não foram processados (controlado pelo número de campos do segmento), volta ao procedimento "d.7".

d.13) Se o segmento descompresso está completo, obtem, da Tabela de Descrição dos Segmentos (TDS), o comprimento descompresso do segmento construído, armazenando-o no registrador NCMP e, a partir dele, verifica se o segmento descompresso está sem conteúdo comparando-o com o segmento correspondente no Padrão de Ausência.

d.14) Posiciona o apontador PPAD no próximo segmento no Padrão de Ausência.

d.15) Se o segmento construído está sem conteúdo, voltar ao procedimento "d.2".

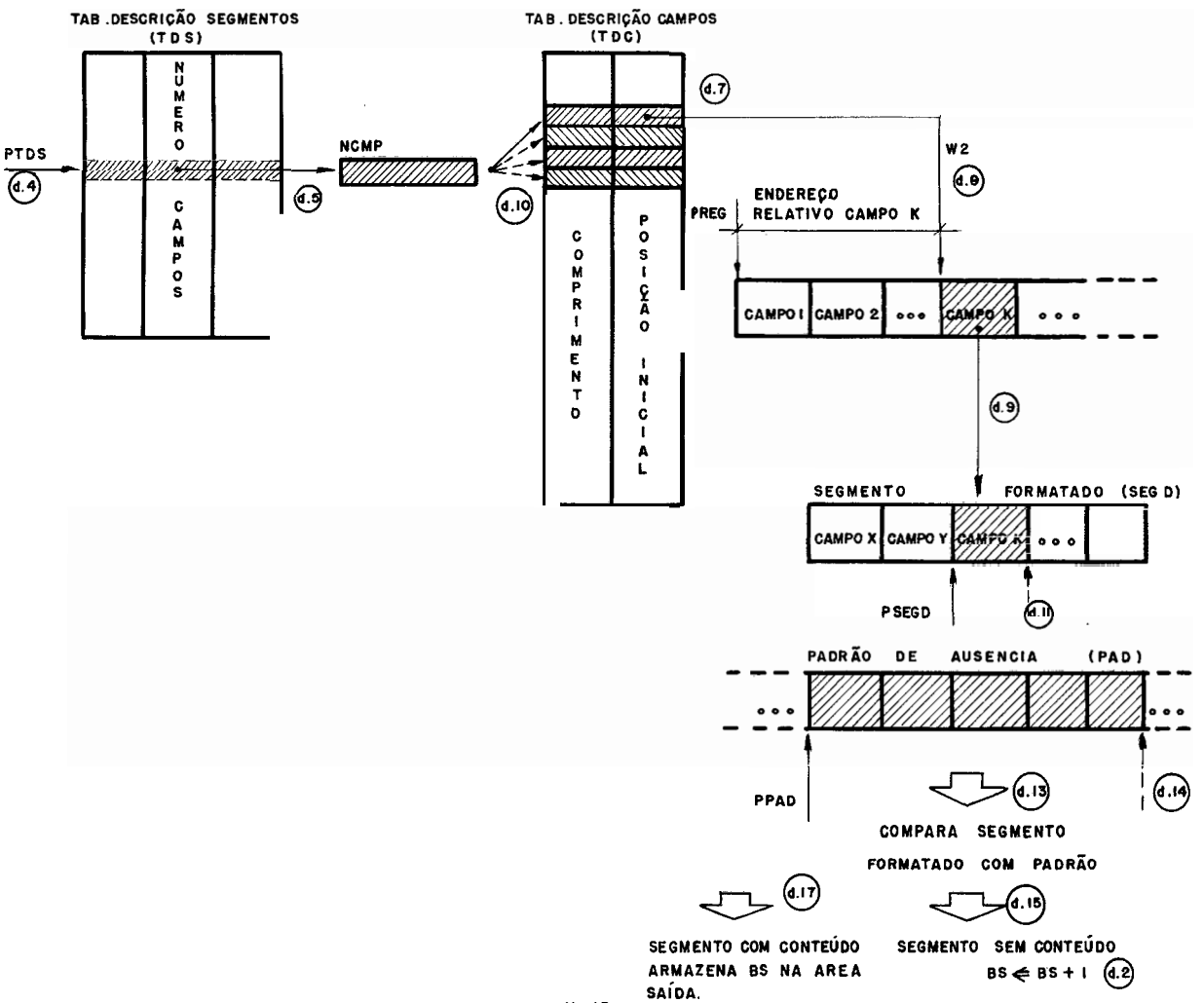
d.16) Caso contrário, verificar se o Byte Sentinela, que se encontra no registrador RBS indica a presença de segmentos sem conteúdo. Em caso negativo, desviar

para execução do módulo "INICIALIZAÇÃO HSF" (item "e").

d.17) Caso contrário, armazena o Byte Sentinela (BS) na área de saída.

d.18) Incrementa de 1 os apontadores de controle da área de saída (PIO, OVFL).

d.19) Verifica se a área de saída está cheia. Caso afirmativo, grava bloco, inicializa apontadores de controle da área de saída (OVFL). Em caso contrário desvia para o módulo "INICIALIZAÇÃO HSF" (item "e").



e) Módulo 'Inicialização HSF'

Este módulo prepara todo o ambiente para a codificação do segmento, pela utilização do código HSF.

Ao ser executado, este módulo:

- Salva registradores de utilização do restante da rotina.
- Carrega os apontadores PSEGC e PSEGD com os endereços das áreas reservadas aos segmentos compresso(SEGC) e descompresso (SEGD).
- Carrega o registrador base BTRAD1 a partir da Tabela de Descrição de Segmentos com o endereço base referente aos dados do segmento a ser codificado das Subtabelas de Tradução (TRAD1).
- Carrega o registrador de contagem de bits (CSAI) com a constante - 32.

f) Módulo 'Verifica Formato dos Segmentos'

A função deste módulo é determinar o formato do segmento, pois, como sabemos, o tratamento dispensado a cada segmento, por ocasião de sua codificação, depende deste dado. Desta forma:

- Segmentos de formato alfanumérico, alfabético e decimal zonado, a codificação/decodificação utilizando o código de tamanho variável HSF, é feita tomando o conteúdo byte a byte.
- Segmentos de formato Decimal Compactado, a codificação/decodificação utilizando o código HSF, é feita com os caracteres de 4 em 4 bits.
- Segmentos de formato Binário não são compressos utilizando o código de tamanho variável HSF.

Este módulo, ao ser executado:

f.1) Compara o código de formato, obtido da Tabela de Descrição de Segmentos correspondente ao segmento em questão, com o código de formato - Decimal Compacta-

do - 768.

f.2) Se o código de formato do segmento for menor que 768, o segmento que está sendo codificado tem o formato caráter e neste caso não há necessidade de descompactá-lo antes da codificação. Passar portanto a execução do módulo "OBTER CÓDIGOS" (item "h").

f.3) Se o código de formato do segmento for igual a 768 o segmento que está sendo codificado tem o formato Decimal Compactado. Neste caso, antes da codificação ele deve ser descompactado. Executa, portanto, módulo "DESCOMPACTA SEGMENTO" (item "g").

f.4) Se o código de formato do segmento for maior que 768 o segmento em questão tem o formato binário e neste caso não será codificado utilizando o código HSF. Assim, uma vez que o segmento já se encontra na área reservada ao segmento descompresso SEGD, carregamos, no apontador PSEGC, o endereço desta área, carregamos o byte imediatamente anterior a SEGD, o comprimento do segmento que se encontra em NCMP e desviamos para executar o módulo "FINAL COMPRESSÃO HSF" (item "i").

g) Módulo 'Descompacta Segmento'

Para segmentos de formato Decimal Compactado este módulo deve ser executado, uma vez que o módulo que codifica os segmentos usando o código HSF, trabalha sempre com segmentos no formato caráter ou seja, um byte para cada carater do segmento compactado inclusive sinal.

Como sabemos, a instrução de compactação(UNPACK), descompacta de cada vez apenas o máximo de 8 bytes. Como nossos segmentos terão comprimento superior àquele máximo, desenvolvemos uma metodologia que descompacta segmento de qualquer tamanho, em etapas de 8 bytes. Observamos que, como na compactação, o segmento é descompactado em outra área.

Isto posto, mostramos as etapas de execução deste módulo:

g.1) Carregar nos registradores de trabalho W1 e W3, respectivamente, o comprimento real + 1 (colocação de um Byte fantasma) e o comprimento real - 1 do segmento compactado.

g.2) Inicializa o registrador incremento NCMP, com - 7 e o registrador de controle TMSEG, com Ø.

g.3) Executar o comando de controle do ciclo (BXLE). Este comando, a cada ciclo de descompactação, decrementa o comprimento real do segmento compactado em W1, de 7. Verifica, em seguida, se o valor em W1 ficou ≤ Ø, o que em caso afirmativo, indica ter sido alcançada a última parte do segmento a ser descompresso e para isto é feito um desvio para execução do procedimento "g.5".

g.4) Para os ciclos normais de descompactação, como vimos anteriormente, descompactamos a informação de 8 em 8 bytes.

Assim este procedimento descompacta os próximos 8 bytes do segmento original, atualiza os apontadores PSEGC e PSEGD com os endereços iniciais dos próximos 15 bytes descompactados e dos próximos 8 bytes a serem descompactados.

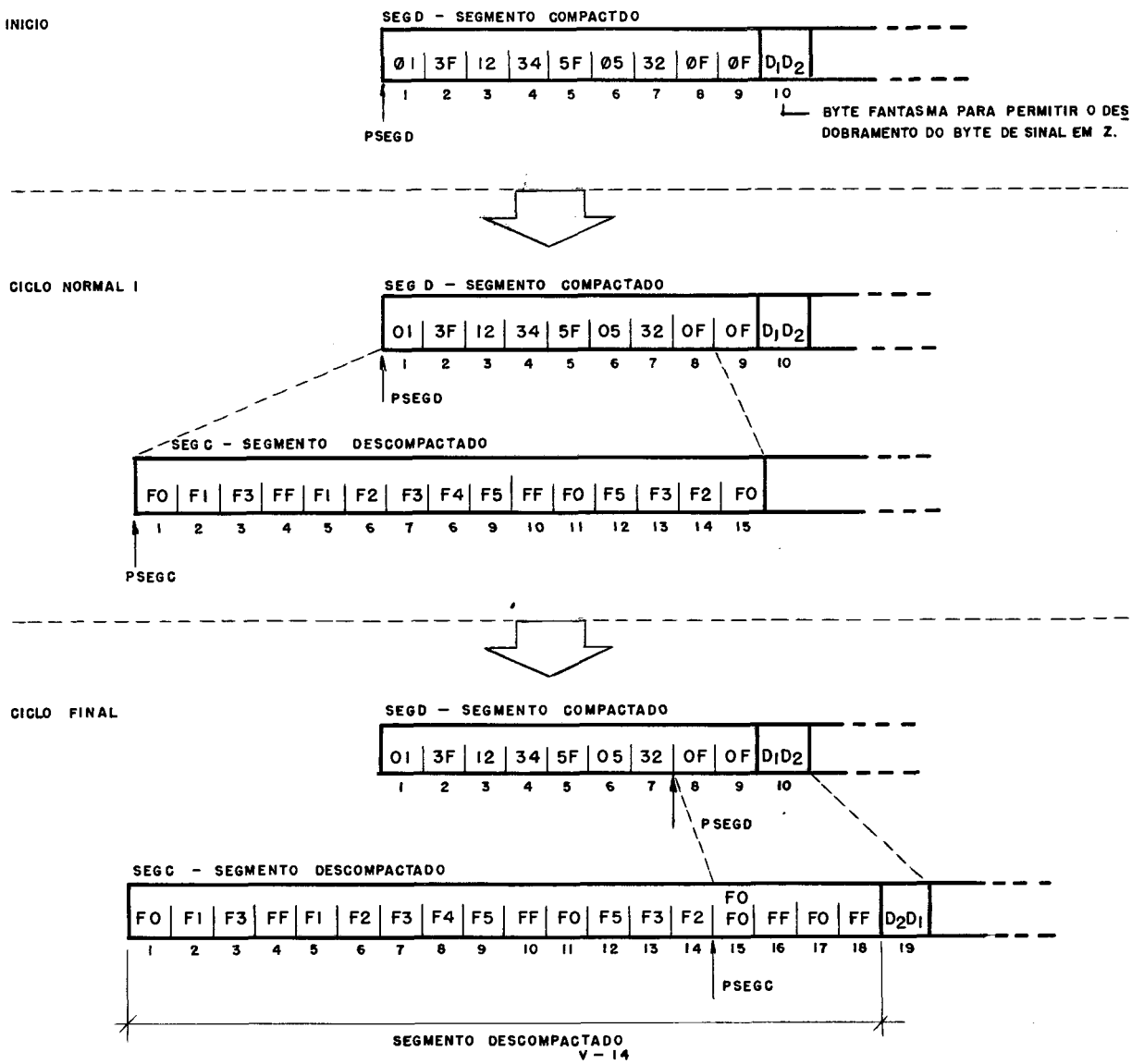
Como sabemos, ao se descompactar um dado,o último byte sofre apenas a troca de posição dígito - sinal. Assim, para as etapas intermediárias de descompactação, devemos ter o cuidado de resubmeter a descompactação o último byte de cada 8 do segmento. Desta forma armazenamos cada carater do segmento compactado em 1 byte.

g.5) Para descompactar a última parte do segmento:

- Determinar, a partir do registrador de trabalho W1, o comprimento compactado da parte final. Lembramos aqui que este comprimento está aumentado de 1, pelo fato do desdobramento também do Byte de Sinal.

- Determinar, a partir do Registrador de Trabalho W1, o comprimento descompactado da parte final.
- Formatar e alterar o byte de comprimento na instrução de descompactação (UNPAK).
- Descompactar a parte final do segmento.
- Carregar os apontadores PSEGC e PSEGD, com o endereço da área reservada ao segmento descompresso, e da área reservada ao segmento compresso. Estas áreas contem, respectivamente, o segmento compactado e o segmento descompactado.
- Carregar o registrador NCMP, com o comprimento descompactado do segmento + 1, pois uma vez que o byte de sinal também é desagregado em 2, o comprimento do nosso segmento descompactado fica aumentado de 1 (Byte Fantasma).

Para melhor compressão do processo, apresentamos de forma ilustrada, a descompactação de um campo com 9 bytes de comprimento.



h) Módulo 'Obter Códigos'

Principal módulo da rotina de compressão. Tem por finalidade, utilizando a metodologia de compressão desenvolvida para a utilização do código HSF (apresentada no capítulo III),determinar, para cada caráter fonte encontrado no segmento descompresso, o grupo do código HSF que lhe é correspondente. Ao final, como resultado da execução deste módulo, obtemos o segmento compresso. Para tanto, ao ser executado, este módulo deve:

h.1) Inicializar o contador CONTA, e o registrador base BCODS.

h.2) Posicionar o apontador de controle TMSEG, com o ende

reço do último byte do segmento descompresso.

h.3) Inicializar o registrador de incremento INSEGD, com o valor 1.

h.4) Salvar o conteúdo do apontador da área referente ao segmento compresso PSEGC em WPSEGC.

h.5) Determinar o fim da codificação. Para isto, empregamos uma filosofia a qual se baseia no endereço do último byte do segmento descompresso, para concluir a codificação.

Decidimos então, utilizar o comando (BXH) para controlar o ciclo o qual, a cada caráter fonte codificado, incrementa de 1 o apontador PSEGD para obtenção do próximo caracter fonte no segmento descompresso, verificando sempre se o endereço no apontador PSEGD, ultrapassou o endereço final do segmento, que se encontra em TMSEG. Caso afirmativo, um desvio será feito para executar o módulo "FINAL COMPRESSÃO HSF" (item "i").

h.6) Carregar no registrador de trabalho W1 o caráter fonte do segmento apontado por PSEGD.

h.7) Determinar através da Tabela de Tradução (TRAD1), o deslocamento que indicará na Tabela de Códigos(CODS), para o caráter fonte obtido, o local de armazenamento do grupo do código HSF gerado para o segmento que lhe é correspondente.

Para acessar a Tabela de Tradução, consideramos o valor relativo à configuração binária do carater fonte como deslocamento. Este deslocamento adicionado ao endereço base da subtabela de tradução(determinado no item "e") nos dá o endereço efetivo.

h.8) Obter da Tabela de Códigos (CODS) o grupo correspondente ao caráter carregando-o no registrador de entrada ENT. Para isto, consideramos o deslocamento obtido da Tabela de Tradução (item "h.7") o qual, adicionado ao endereço base da subtabela de códigos (de

terminado no item "e"), nos dá o endereço efetivo de acesso ao grupo desejado.

h.9) Selecionar do registrador de entrada ENT o comprimento do grupo obtido, colocando-o no registrador de deslocamento CENT.

h.10) Verificar se o registrador de saída (SAI), suporta o grupo obtido. Para isto tomamos o registrador CSAI que expressa na forma negativa, o número de bits disponíveis no registrador de saída SAI e adicionamos o comprimento do grupo que se encontra no registrador CENT. Se o resultado for positivo, significa que o registrador de saída não suporta todos os bits do grupo em questão, e neste caso desviamos para executar o procedimento "h.14". Se, no entanto, o resultado for menor ou igual a zero, o registrador de saída suporta todo o grupo e o procedimento seguinte deve ser executado.

d.11) Deslocar para o registrador de saída SAI, a partir do comprimento do grupo em CENT, todos os bits do grupo que está armazenado no registrador de entrada ENT.

d.12) Se o resultado da adição procedida no item h.10 é menor que zero, significa que o registrador de saída SAI, após a transferência do grupo, ainda possui bits disponíveis.

Por esta razão, voltamos ao procedimento "h.5" para obtenção do próximo caráter fonte e consequentemente do próximo grupo.

d.13) Se o resultado da adição procedida no item "h.10" é igual a zero, significa que, após a transferência do grupo, o número de bits disponíveis no registrador de saída SAI, é zero, ou seja, que o mesmo esta totalmente cheio e neste caso as seguintes funções são executadas.

- Armazenar os 4 bytes do registrador de saída na área reservada ao segmento compresso (para endereçar utilizamos o apontador PSEGC).

- Adicionar 4 ao registrador que computa o comprimento do segmento compresso (CONTA).
- Adicionar 4 ao apontador que endereça os bytes a serem armazenados no segmento compresso(PSEGC).
- Inicializar o contador de bits disponíveis(CSAI), com o valor '-32'.
- Voltar ao procedimento "h.5" para obtenção do próximo caráter fonte, e, consequentemente, do próximo grupo.

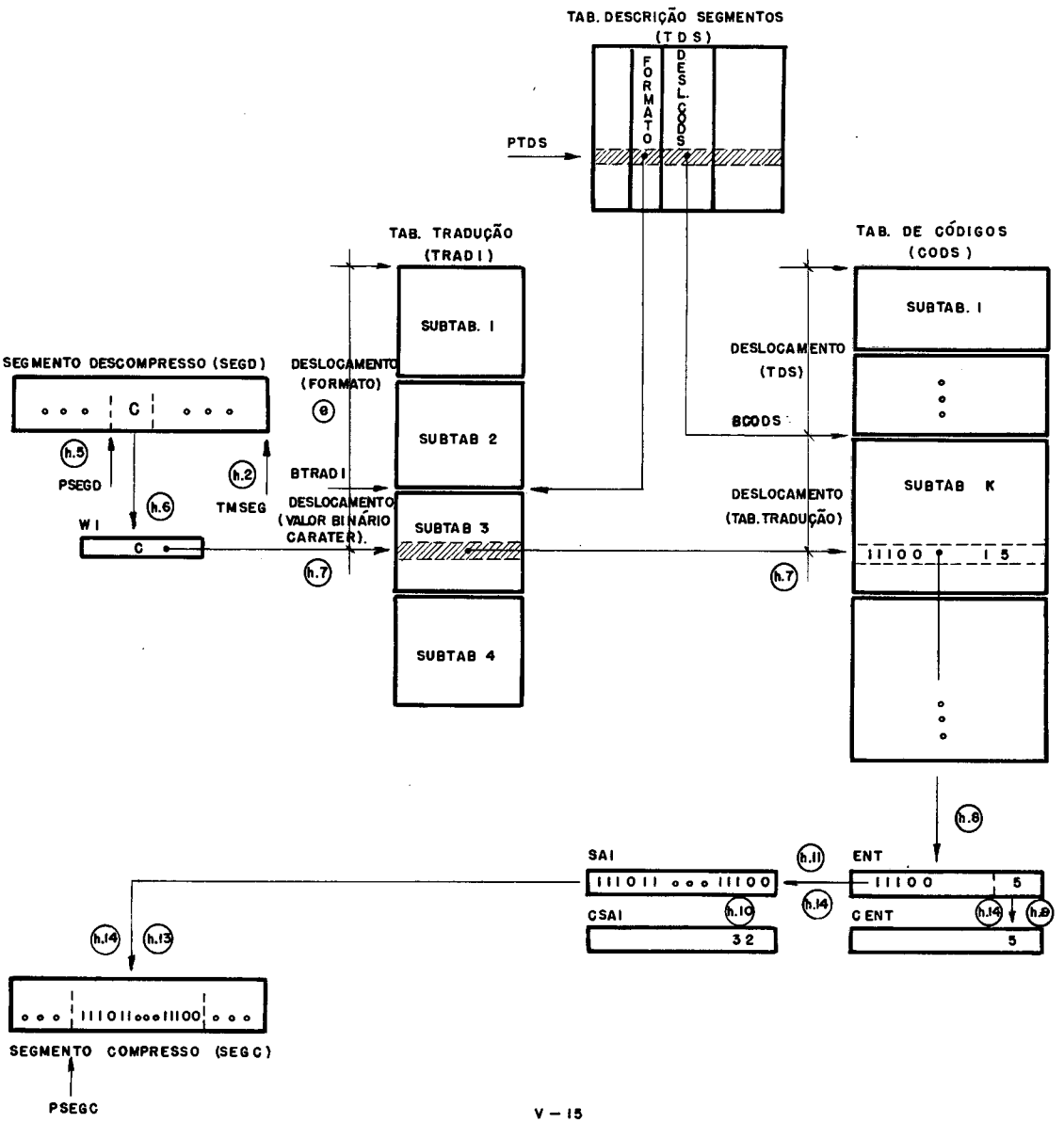
h.14) Por ter sido o resultado da adição procedida no item "h.10" negativa, significando que o registrador de saída (SAI) não possui bits disponíveis em número suficiente para suportar todo o grupo que se encontra no registrador de entrada ENT, teremos que transferir o grupo em 2 etapas.

Desta forma, este procedimento:

- Determina, no registrador CENT, o número de bits disponíveis no registrador de saída SAI.
- Desloca do registrador de entrada ENT, para o registrador de saída SAI, tantos bits do grupo quantos sejam os bits disponíveis registrados em CENT.
- Armazena os 4 bytes do registrador de saída, na área reservada ao segmento compresso, utilizando para endereçamento o apontador PSEGC.
- Desloca do registrador de entrada ENT, para o registrador de saída SAI, os bits restantes do grupo. O número de bits a ser deslocado se encontra registrado em CSAI.
- Atualiza o contador de bits disponíveis CSAI adicionando a constante '-32'.
- Adiciona 4 ao registrador que computa o comprimento do segmento compresso (CONTA).
- Adiciona 4 ao apontador que endereça os bytes

serem armazenados no segmento compresso(PSEGC).

- Volta ao procedimento "h.5" para obtenção do próximo caráter fonte e, consequentemente, do próximo grupo.



V – 15

i) Módulo 'Final Compressão HSF'

Através deste módulo, transferimos do registrador de saída SAI para a área reservada ao segmento compresso, os últimos bytes codificados mas não transferidos, encerrando, em seguida, a parte da rotina responsável pela compressão utilizando o código de tamanho variável HSF.

Assim ao ser executado, este módulo deve:

i.1) Caso o contador de bits disponíveis CSAI seja igual a -32, nenhum byte do registrador de saída foi processado, e neste caso, deve ser feito um desvio para execução do procedimento "i.10"; caso contrário deve ser executado o procedimento que se segue.

i.2) A partir do contador CSAI, determinar o número de bits utilizados no registrador de saída SAI.

i.3) Triplicar o valor encontrado no procedimento anterior, colocando-o no registrador SAI.

i.4) Carregar no registrador CENT o endereço da Tabela de Mascara (MASC).

i.5) Tomando o valor determinado no procedimento "i.3" como deslocamento e o endereço base da Tabela de Mascara, acessamos esta tabela, carregando-se, a partir dela, no registrador de trabalho W1, o número de bits que deverá ser deslocado para a esquerda no registrador de saída SAI, de forma que os bits já processados fiquem alinhados no limite de byte.

i.6) Com o valor determinado no procedimento anterior e registrado em W1, deslocar para a esquerda os bits do registrador de saída SAI.

i.7) Considerando a mesma entrada na tabela, determinada no procedimento "i.5", obter o byte de mascara que, armazenado na instrução STCM, indicará que bytes do registrador de saída SAI que, possuindo dados ainda não transferidos, deverão ser armazenados na área do segmento compresso.

i.8) Transferir os últimos bytes do registrador de saída para o segmento compresso.

i.9) Mais uma vez, da mesma entrada na Tabela de Mascaras, obtemos, no registrador de trabalho W1, o número de bytes transferidos, adicionando, em seguida, este valor ao contador de tamanho do segmento compresso (CONTA).

i.10) Restaurar o conteúdo de PSEGC.

i.11) Armazenar, a partir de CONTA, o comprimento do segmento - 1 no 1º byte do segmento compresso.

i.12) Restaurar o conteúdo dos registradores de utilização do restante da rotina.

j) Módulo 'Rotina de Colocação dos Segmentos na Ioarea'

Uma vez que temos o segmento compresso, passamos a execução deste módulo, que é o responsável pela transferência deste segmento, de uma área reservada para a área de saída.

Ao ser executado, este módulo:

j.1) Atualiza os registradores de controle da área de saída (OVFL, W2).

j.2) Se a área de saída comporta o segmento compresso integralmente:

- Com o tamanho compresso, que se encontra no registrador de trabalho NCMP, transferir da área reservada, para a área de saída, o segmento compresso.
- Com o comprimento do segmento compresso, atualizar o apontador de controle da área de saída PIO, com o endereço, para armazenamento do próximo segmento.
- Verificar se a área de saída esta completamente preenchida, caso afirmativo, gravar o bloco no arquivo compresso inicializando os apontadores de controle.
- Desviar para o módulo "ROTINA DE PREPARAÇÃO/TESTES DE SEGMENTOS" (item "d").

j.3) Se a área de saída não comporta o segmento integralmente:

- Determinar quantos bytes do segmento compresso excederam a capacidade da área de saída.
- Determinar quantos bytes do segmento compresso comporta a área de saída.

- Com o número de bytes obtido do item anterior, transferir da área reservada (SEGC), para a área de saída, a primeira parte do segmento compresso.
- Gravar este bloco no arquivo compresso.
- Com o comprimento da primeira parte do segmento compresso, atualizar o apontador da área reservada PSEGC, com o endereço inicial da 2a. parte do segmento compresso.
- Executar o item "j.2", considerando como segmento compresso a 2a. parte a ser transferida do atual segmento.

k) Módulo 'Rotina de Finalizações'

Este módulo encerra a rotina de compressão e para isto:

k.1) Se fim de arquivo ('*' no 1º byte do parametro):

- Coloca marca de fim de arquivo, na área de saída (X'ØØ').
- Grava último bloco do arquivo compresso.
- Fecha o arquivo de saída (ARQS).
- Desvia para o item "k.3".

k.2) Se fim de registro (X'ØØ' - número de campos na Tabela Descrição de Segmentos):

- Verifica se o Byte Sentinela possui conteúdo.
- Desvia para o item "k.3", se o Byte Sentinela não tem conteúdo.
- Caso contrário, armazena o Byte Sentinela na área de saída incrementando os respectivos apontadores de controle (PIO, OVFL). Continuando, verifica se a área de saída está cheia, o que em caso afirmativo procede a gravação do bloco, atualizando os apontadores de controle da área de saída.

k.3) Salva os apontadores de controle da área de saída (PIO, OVFL).

k.4) Restaura o conteúdo dos registradores para o programa principal.

k.5) Retorna ao programa principal.

3.5. MODIFICAÇÕES

As modificações permitidas nesta rotina são as mesmas expostas para a rotina de descompressão, pois como já vimos ambas se utilizam dos mesmos fundamentos.

CAPÍTULO VI

RESULTADOS

1. APRESENTAÇÃO

Neste capítulo mostraremos de forma prática uma aplicação do nosso sistema de compressão ao Arquivo de Débitos dos Consumidores de uma empresa de energia elétrica.

Para esta apresentação, como já mostramos em capítulos anteriores, analisaremos em detalhes todas as etapas, programa a programa, desde a descrição do registro original, até a avaliação da compressão final obtida.

2. APLICAÇÃO DO PROGRAMA GERADOR DE ESTATÍSTICAS PARA SEGMENTAÇÃO (COMPØ3Ø)

2.1. DESCRIÇÃO DAS ENTRADAS

2.1.1. Arquivo de Aplicação

Como já dissemos, o arquivo a ser trabalhado é o de Débitos dos Consumidores de uma empresa de energia elétrica, o qual apresenta as características abaixo:

- Número total de registros no arquivo - 150.000
- Número total de registros processados - 80.000
- Suporte de armazenamento - Fita
- Label Standard IBM.

2.1.2. Arquivo Descrição dos Campos

Para a definição destes dados que fornecem ao programa todos os subsídios necessários ao processamento dos registros do Arquivo - Aplicação, consideremos a tabela a seguir, onde encontramos para cada campo do registro sua descrição acompanhada da entrada correspondente neste arquivo.

DESCRIÇÃO DOS CAMPOS					ARQUIVO		
NOME	DESCRIÇÃO	POSIÇÃO INICIAL	TAMANHO		IDENTIFICAÇÃO	TAMANHO	CÓDIGO FORMATO
CAMPO 1	SEQUENCIAL	1	7	DECIMAL COMPACTADO	00001	007	3
CAMPO 2	CODIGO REGISTRO	8	1	DECIMAL ZONADO	00002	001	2
CAMPO 3	NÚMERO CONTA	9	9	DECIMAL ZONADO	00003	009	2
CAMPO 4	TARIFA	18	3	ALFANUMÉRICO	00004	003	1
CAMPO 5	DEMANDA	21	3	DECIMAL COMPACTADO	00005	003	3
CAMPO 6	CONSUMO	24	4	DECIMAL COMPACTADO	00006	004	3
CAMPO 7	IMPORTE	28	5	DECIMAL COMPACTADO	00007	005	3
CAMPO 8	IMPOSTO ÚNICO	33	4	DECIMAL COMPACTADO	00008	004	3
CAMPO 9	EMPRESTIMO COMPULSÓRIO	37	5	DECIMAL COMPACTADO	00009	005	3
CAMPO 10	QUOTA DE PREVIDENCIA	42	4	DECIMAL COMPACTADO	00010	004	3
CAMPO 11	ILUMINAÇÃO PÚBLICA	46	4	DECIMAL COMPACTADO	00011	004	3
CAMPO 12	AJUSTE ANTERIOR	50	1	DECIMAL COMPACTADO	00012	001	3
CAMPO 13	AJUSTE ATUAL	51	1	DECIMAL COMPACTADO	00013	001	3
CAMPO 14	MULTA	52	4	DECIMAL COMPACTADO	00014	004	3
CAMPO 15	TOTAL	56	5	DECIMAL COMPACTADO	00015	005	3
CAMPO 16	MES	61	2	DECIMAL ZONADO	00016	002	2
CAMPO 17	ANO	63	2	DECIMAL ZONADO	00017	002	2
CAMPO 18	GRUPO	65	2	DECIMAL COMPACTADO	00018	002	3
CAMPO 19	SUBGRUPO	67	2	DECIMAL COMPACTADO	00019	002	3
CAMPO 20	NOME CONSUMIDOR	69	22	ALFANUMÉRICO	00020	022	1
CAMPO 21	ENDEREÇO CONSUMIDOR	91	22	ALFANUMÉRICO	00021	022	1

VI - 1

Com base na tabela apresentada como podemos observar a seguir, foi preparada a massa real a ser utilizada pelo programa, com a descrição dos campos no registro.

Chamamos a atenção para o fato de que o primeiro cartão informa o fator de bloco (60) e o numero de campos definidos no registro (21).

00053	0021	0
00001	0007	3
00002	0001	2
00003	0009	2
00004	0003	1
00005	0003	3
00006	0004	3
00007	0005	3
00008	0004	3
00009	0005	3
00010	0004	3
00011	0004	3
00012	0001	3
00013	0001	3
00014	0004	3
00015	0005	3
00016	0002	2
00017	0002	2
00018	0002	3
00019	0002	3
00020	0022	1
00021	0022	1

VI - 2

2.1.3. Arquivo Console

Em resposta às solicitações do programa foram fornecidas as seguintes informações:

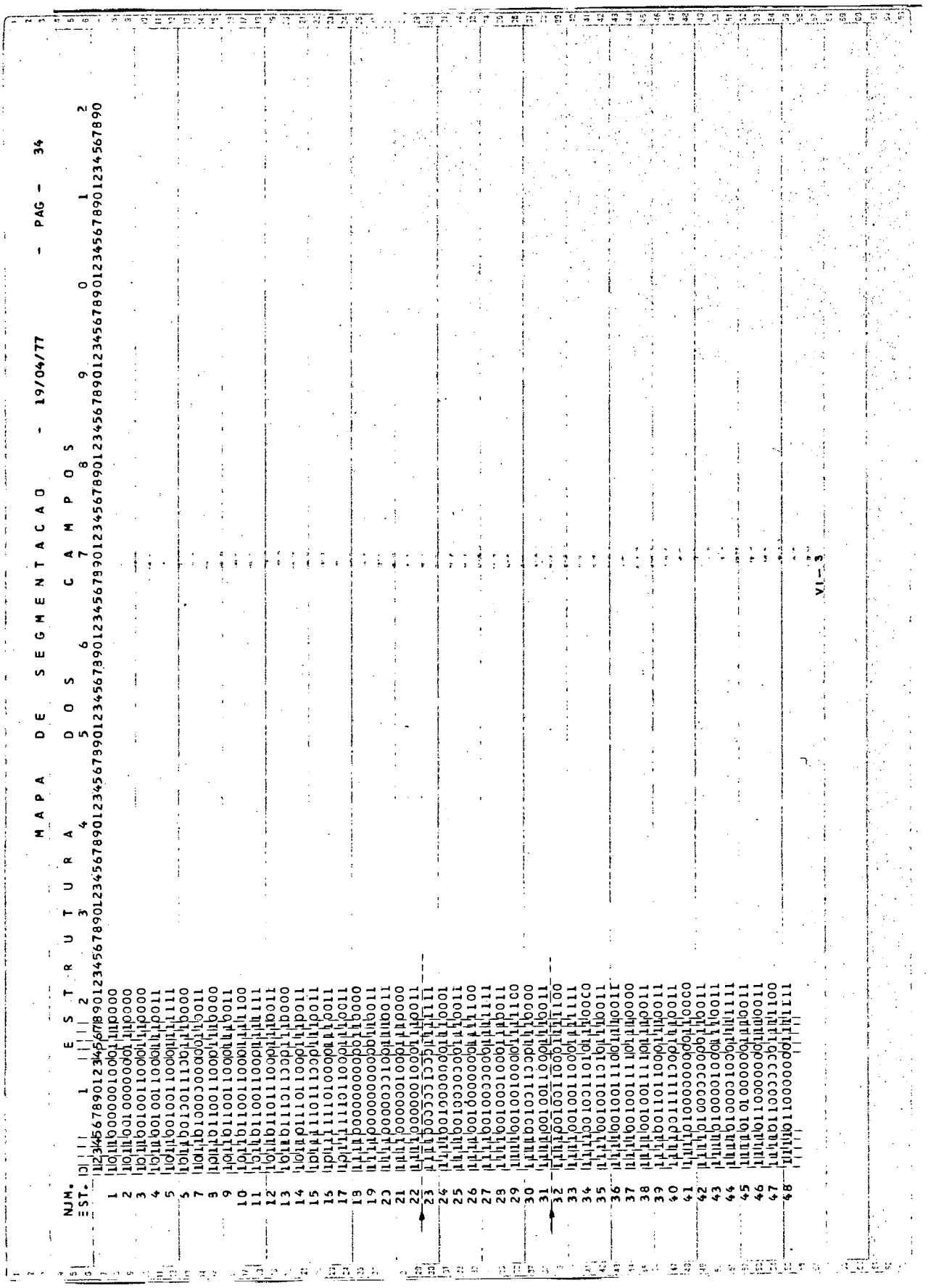
- Data - 10/04/77
- Limite de impressão dos relatórios - 10000
- Fator de bloco - 60
- Comprimento do registro - 112

2.2. DESCRIÇÃO DAS SAÍDAS

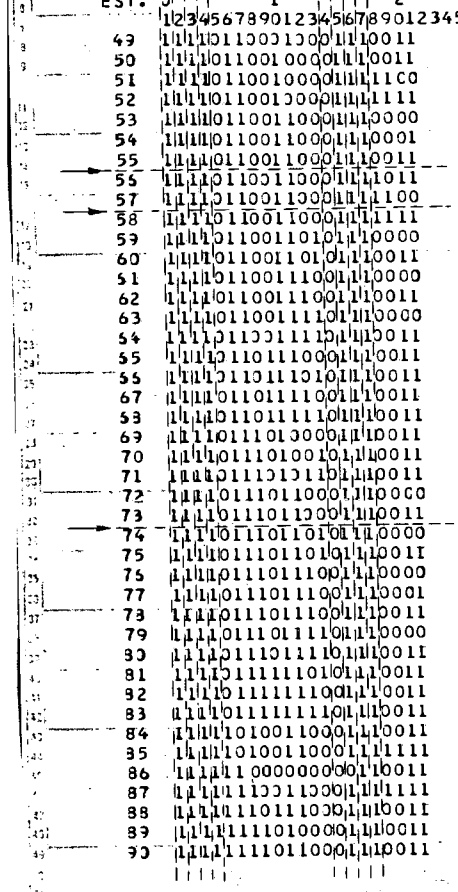
Para este arquivo, por meio da impressão sistemática dos relatórios a cada 10000 registros processados, pudemos constatar, após o processamento de 80000 registros, a estabilidade estatística das informações manipuladas.

Por esta razão, no decorrer da exposição apresentamos para efeito de análise, apenas os últimos relatórios gerados.

2.2.1. Mapa das Estruturas



NUM. EST. — ESTRUTURA DOS CAMPOS



Deste relatório, podemos, de imediato, fazer as seguintes observações:

- Em 80.000 registros analisados foram encontradas 90 estruturas diferentes.
- Campos 1, 3, 4, 16 e 17 se encontram sempre com conteúdo (dígito '1' nas posições 1, 3, 4, 16 e 17).
- Campo 14 se encontra sempre sem conteúdo - (dígito '0' na posição 14).

MAPA DE SEGMENTACAO - 19/04/77 - PAG - 36

NUM. EST.	FREQ. ESTR.	PROBAL. ESTRUT.	COMP.ORIGINAL EST.	COMP.ORIGINAL ARQ.	COMP.COMPRES. EST.	COMP.COMPRES. ARQ.	COMPR.ABSOLUTA EST.	COMPR.ABSOLUTA ARQ.	COMPR.RELATIVA EST.	COMPR.RELATIVA ARQ.	COMPR.PER. EST.	COMPR.PER. ARQ.	RAIO COMP.
1	8	0,000100	112	896	32	256	80	640	0,714285	0,000071	71		3,500000
2	1	0,000012	112	112	33	33	79	79	0,705357	0,000008	70		3,393939
3	2	0,000025	112	224	41	82	71	142	0,633928	0,000015	63		2,731707
4	99	0,001237	112	11088	85	8415	27	2673	0,241071	0,000298	24		1,317647
5	55	0,000687	112	6160	89	4895	23	1265	0,205357	0,000141	20		1,258426
6	1	0,000012	112	112	42	42	70	70	0,625000	0,000007	62		2,666666
7	2	0,000025	112	224	71	142	41	82	0,366071	0,000009	36		1,577464
8	5	0,000062	112	560	45	225	67	335	0,598214	0,000037	59		2,488888
9	357	0,004462	112	39984	89	31773	23	8211	0,205357	0,000916	20		1,258426
10	1	0,000012	112	112	49	49	63	63	0,562500	0,000007	56		2,285714
11	378	0,004725	112	42336	93	35154	19	7182	0,169642	0,000801	16		1,204301
12	1	0,000012	112	112	94	94	18	18	0,160714	0,000002	16		1,191489
13	4	0,000050	112	448	49	196	63	252	0,562500	0,000023	56		2,285714
14	554	0,006925	112	62048	93	51522	19	10526	0,169642	0,001174	16		1,204301
15	2	0,000025	112	224	97	194	15	30	0,133928	0,000003	13		1,154639
16	1	0,000012	112	112	92	92	20	20	0,178571	0,000002	17		1,217391
17	6	0,000075	112	672	96	576	16	96	0,142857	0,000010	14		1,166666
18	3	0,000037	112	336	24	72	88	264	0,785714	0,000029	78		4,666666
19	973	0,012162	112	108976	68	66164	44	42812	0,392857	0,004778	39		1,547058
20	404	0,005050	112	45248	74	29896	38	15352	0,339285	0,001713	33		1,513513
21	14	0,000175	112	1568	33	462	79	1106	0,705357	0,000123	70		3,393939
22	4983	0,062287	112	558096	77	383691	35	174405	0,312500	0,019464	31	1	1,454545
23	214	0,002675	112	23968	81	17334	31	6634	0,276785	0,000740	27		1,382716
24	2	0,000025	112	224	56	112	56	112	0,500000	0,000012	50		2,000000
25	169	0,002112	112	18928	78	13182	34	5746	0,303571	0,000641	30		1,435897
26	86	0,001075	112	9632	38	3268	74	6364	0,660714	0,000710	66		2,947368
27	78	0,000975	112	8736	82	6396	30	2340	0,267857	0,000261	26		1,365853
28	1	0,000012	112	112	82	82	30	30	0,267857	0,000003	26		1,365853
29	9	0,000112	112	1008	42	378	70	630	0,625000	0,000070	62		2,666666
30	15	0,000187	112	1680	42	630	70	1050	0,625000	0,000117	62		2,666666
31	8330	0,110375	112	988960	86	759380	26	229580	0,232142	0,025622	23	2	1,302325
32	703	0,008787	112	78736	46	32338	66	46398	0,589285	0,005178	58		2,434782
33	2491	0,031137	112	278992	90	224190	22	54802	0,196428	0,006116	19		1,244444
34	2	0,000025	112	224	43	86	69	138	0,616071	0,000015	61		2,604651
35	157	0,001962	112	17584	87	13659	25	3925	0,223214	0,000438	22		1,287356
36	103	0,001287	112	11536	87	8961	25	2575	0,223214	0,000287	22		1,287356
37	1	0,000012	112	112	44	44	68	68	0,607142	0,000007	60		2,545454
38	1101	0,013762	112	123312	88	96888	24	26424	0,214285	0,002949	21		1,272727
39	2	0,000025	112	224	90	180	22	44	0,196428	0,000004	19		1,244444
40	1	0,000012	112	112	95	95	17	17	0,151785	0,000001	15		1,178947
41	1	0,000012	112	112	28	28	84	84	0,750000	0,000009	75		4,000000
42	12	0,000150	112	1344	72	864	40	480	0,357142	0,000053	35		1,555555
43	47	0,000587	112	5264	81	3807	31	1457	0,276785	0,000162	27		1,382716
44	1	0,000012	112	112	85	85	27	27	0,241071	0,000003	24		1,317647
45	7	0,000087	112	784	81	567	31	217	0,276785	0,000024	27		1,382716
46	375	0,004687	112	42000	82	30750	30	11250	0,267857	0,001255	26		1,365853
47	1	0,000012	112	112	42	42	70	70	0,625000	0,000007	62		2,666666
48	2	0,000025	112	224	86	172	26	52	0,232142	0,000005	23		1,302325

NUM. EST.	FREQ. ESTR.	PROBAL. ESTRUT.	COMP.ORIGINAL EST.	COMP.ORIGINAL ARQ.	COMP.COMPRES. EST.	COMP.COMPRES. ARQ.	COMPR.ABSOLUTA EST.	COMPR.ABSOLUTA ARQ.	COMPR.RELATIVA EST.	COMPR.RELATIVA ARQ.	COMPR.PER. EST.	COMPR.PER. ARQ.	RAIO COMP.
49	12	0,000150	112	1344	86	1032	26	312	0,232142	0,000034	23		1,302325
50	5	0,000062	112	560	86	430	26	130	0,232142	0,000014	23		1,302325
51	3	0,000037	112	336	46	138	66	198	0,589285	0,000022	58		2,434782
52	24	0,000300	112	2688	90	2160	22	528	0,196428	0,000058	19		1,244444
53	18	0,000225	112	2016	46	828	66	1188	0,589285	0,000132	58		2,434782
54	1	0,000012	112	112	68	68	44	44	0,392857	0,000004	39		1,647058
55	14722	0,184025	112	1648864	90	1324980	22	323884	0,196428	0,036147	19	3	1,244444
56	2	0,000025	112	224	92	184	20	40	0,178571	0,000004	17		1,217391
57	1775	0,022187	112	198800	50	88750	62	110050	0,553571	0,012282	55	1	2,240000
58	14154	0,176925	112	1585248	94	1330476	18	254772	0,160714	0,028434	16	2	1,191489
59	1	0,000012	112	112	47	47	65	65	0,580357	0,000007	58		2,382978
60	125	0,001562	112	14000	91	11375	21	2625	0,187500	0,000292	18		1,230769
61	1	0,000012	112	112	47	47	65	65	0,580357	0,000007	58		2,382978
62	113	0,001412	112	12656	91	10283	21	2373	0,187500	0,000264	18		1,230769
63	8	0,000100	112	896	48	384	64	512	0,571428	0,000057	57		2,333333
64	1016	0,012700	112	113792	92	93472	20	20320	0,178571	0,002267	17		1,217391
65	86	0,001075	112	9632	95	8170	17	1462	0,151785	0,000163	15		1,178947
66	3	0,000037	112	336	96	288	16	48	0,142857	0,000005	14		1,166666
67	1	0,000012	112	112	96	96	16	16	0,142857	0,000001	14		1,166666
68	9	0,000112	112	1008	97	873	15	135	0,133928	0,000015	13		1,154639
69	55	0,000687	112	6160	90	4950	22	1210	0,196428	0,000135	19		1,244444
70	1	0,000012	112	112	91	91	21	21	0,187500	0,000002	18		1,230769
71	2	0,000025	112	224	92	184	20	40	0,178571	0,000004	17		1,217391
72	17	0,000212	112	1904	50	850	62	1054	0,553571	0,000117	55		2,240000
73	22910	0,286375	112	2565920	94	2153540	18	412380	0,160714	0,046024	16	4	1,191489
74	8	0,000100	112	896	51	408	61	488	0,544642	0,000054	54		2,195078
75	369	0,004612	112	41328	95	35055	17	6273	0,151785	0,000700	15		1,178947
76	3	0,000037	112	336	51	153	61	183	0,544642	0,000020	54		2,196078
77	1	0,000012	112	112	73	73	39	39	0,348214	0,000004	34		1,534246
78	195	0,002437	112	21840	95	18525	17	3315	0,151785	0,000369	15		1,178947
79	10	0,000125	112	1120	52	520	60	600	0,535714	0,000066	53		2,153846
80	1812	0,022650	112	202944	96	173952	16	28992	0,142857	0,003235	14		1,166666
81	1	0,000012	112	112	100	100	12	12	0,107142	0,000001	10		1,120000
82	1	0,000012	112	112	100	100	12	12	0,107142	0,000001	10		1,120000
83	16	0,000200	112	1792	101	1616	11	176	0,098214	0,000019	9		1,108910
84	2	0,000025	112	224	89	178	23	46	0,205357	0,000005	20		1,258426
85	50	0,000625	112	5600	93	4650	19	950	0,169642	0,000106	16		1,204301
86	24	0,000300	112	2688	75	1800	37	888	0,330357	0,000099	33		1,493333
87	61	0,000762	112	6832	97	5917	15	915	0,133928	0,000102	13		1,154639
88	36	0,000450	112	4032	98	3528	14	504	0,125000	0,000055	12		1,142857
89	12	0,000150	112	1344	93	1116	19	228	0,169642	0,000025	16		1,204301
90	70	0,000875	112	7840	97	6790	15	1050	0,133928	0,000117	13		1,154639
	80000	0,999975	10080	8960000	6701	7115720	3379	1844280					

Para 80000 estruturas observadas, correspondentes aos 80000 registros analisados, apresentamos o comportamento estatístico e de compressão para cada estrutura tanto individualmente como na amostra. Entre as observadas destacamos a estrutura 73, como sendo a que apresentou maior frequência de ocorrência (22910), contribuindo com 2.565.920 bytes para a formação da amostra e possuindo uma compressão absoluta em potencial de 412.380 bytes.

MAPA DE SEGMENTACAO - 19/04/77 - PAG - 38

NUM. CMP.	POS. INI.	FREQ. CMP.	PROBAB. CAMPO	COMP. ORIG.	PARTIC. TOTAL	PARTIC. COMPR.	COMPR. ABSOLUTA	COMPR. RELATIVA	COMP. PERC.	COMP. REL. ARQUIVO	FORMATO CAMPO
1	1	80000	1,000000	7	560000	560000	0				DECIMAL COMPACTADO
2	8	78523	0,981537	1	80000	78523	1477	0,018462	1	0,000164	DECIMAL ZONADO
3	9	80000	1,000000	9	720000	720000	0				DECIMAL ZONADO
4	18	80000	1,000000	3	240000	240000	0				ALFANUMERICO
5	21	264	0,003300	3	240000	792	239208	0,996700	99	0,026697	DECIMAL COMPACTADO
6	24	59440	0,743000	4	320000	237760	82240	0,257000	25	0,009178	DECIMAL COMPACTADO
7	28	73307	0,916337	5	400000	366535	33465	0,083662	8	0,003734	DECIMAL COMPACTADO
8	33	26058	0,325725	4	320000	104232	215768	0,674275	67	0,024081	DECIMAL COMPACTADO
9	37	157	0,001962	5	400000	785	399215	0,998037	99	0,044555	DECIMAL COMPACTADO
10	42	72580	0,907250	4	320000	290320	29680	0,092750	9	0,003312	DECIMAL COMPACTADO
11	46	77748	0,971850	4	320000	310992	9008	0,028150	2	0,001005	DECIMAL COMPACTADO
12	50	4798	0,059975	1	80000	4798	75202	0,940025	94	0,008393	DECIMAL COMPACTADO
13	51	4642	0,058025	1	80000	4642	75358	0,941975	94	0,008410	DECIMAL COMPACTADO
14	52	0	0,000000	4	320000	0	320000	1,000000	100	0,035714	DECIMAL COMPACTADO
15	56	78985	0,987312	5	400000	394925	5075	0,012687	1	0,000566	DECIMAL COMPACTADO
16	61	80000	1,000000	2	160000	160000	0				DECIMAL ZONADO
17	63	80000	1,000000	2	160000	160000	0				DECIMAL ZONADO
18	65	20088	0,251100	2	160000	40176	119824	0,748900	74	0,013373	DECIMAL COMPACTADO
19	67	20086	0,251075	2	160000	40172	119828	0,748925	74	0,013373	DECIMAL COMPACTADO
20	69	77295	0,966187	22	1760000	1700490	59510	0,033812	3	0,006641	ALFANUMERICO
21	91	77299	0,966237	22	1760000	1700578	59422	0,033762	3	0,006631	ALFANUMERICO
		1071270	3,390872	112	8960000	7115720	1844280				

VI - 7

Para este relatório, que mostra o comportamento estatístico e de compressão para cada um dos campos no registro, foram analisados 1.680.000 campos (21x80000), dos quais 1.071.270 se apresentaram com conteúdo.

Dos 21 campos destacamos o campo 9, como sendo aquele que maior compressão pode nos oferecer, conforme os dados abaixo:

- Participação Total na amostra - 5*80000 □ 400000 bytes.
- Participação Compressa na amostra = 5*157 □ 785 bytes.
- Compressão Absoluta □ 399215 bytes.
- Compressão Relativa a amostra = 0,044555 (4,45%).

Isto é, a eliminação do campo 9 na amostra corresponde a 4,45% do tamanho total da amostra.

Enfim, de uma forma geral, observando as linhas de totais dos Relatórios Comportamento Estatístico das Estruturas e dos Campos e considerando a eliminação de todos os campos sem conteúdo da amostra, temos:

- Comprimento Original na amostra - 112*800000 - 89600000 bytes.
- Comprimento Compresso Mínimo da amostra - 7115720 bytes.
- Compressão Absoluta Máxima na amostra - 1.844.280 bytes.
- Compressão Relativa Máxima na amostra - $\frac{1844280}{8960000} = 0,2058$ ou 20,58%.

Pelo que acabamos de apresentar podemos concluir ser, o nosso arquivo, denso de informações.

2.3. PROJETO DO REGISTRO SEGMENTADO

De posse dos relatórios anteriormente apresentados, passamos à fase de projeto do registro segmentado, quando então definiremos que campos comporão quais segmentos.

Para isto, seguiremos os procedimentos sugeridos para esta seleção:

a) Selecionar do relatório Comportamento Estatístico das Estruturas aquelas que oferecam maior compressão no arquivo.

IDENTIFICAÇÃO ESTRUTURA	COMP. PERCENTUAL ARQUIVO	COMP. RELATIVA ARQUIVO
22	1	0,019464
31	2	0.025622
55	3	0.036147
57	1	0.012282
58	2	0.028434
73	4	0.046024

VI-8

b) Selecionar a configuração correspondente à cada estrutura do Mapa de Estruturas:

IDENTIFICAÇÃO ESTRUTURA	CONFIGURAÇÃO
22	111100000010001110011
31	111100010010001110011
55	111101100100011100011
57	111101100100011111100
58	111101100100011111111
73	111101101100011100011

VI - 9

c) Analisando as estruturas acima quanto à presença ou ausência de conteúdo nos seus campos, construimos, a partir dela, a estrutura padrão abaixo:

11110XXX0X1000XXX XXXX, onde

o 'X' representa campos indefinidos quanto à presença ou ausência de conteúdo.

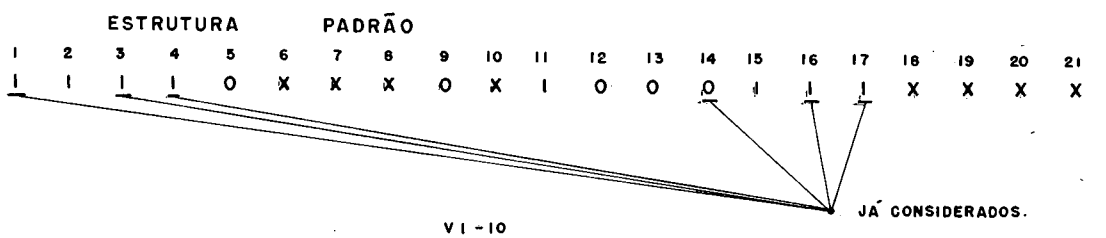
d) Considerando a estrutura padrão e o Mapa das Estruturas determinar os campos sempre com ou sem conteúdo.

- Campos sempre com conteúdo - 1, 3, 4, 16, 17.
- Campos sempre sem conteúdo - 14.

e) Agrupar em 2 ou mais segmentos os campos sempre com e sem conteúdo, considerando os de mesmas características,os relacionados logicamente, etc.

- 1º segmento - sempre com conteúdo - campos 1, 16, 17 .
- 2º segmento - sempre com conteúdo - campos 3, 4.
- 3º segmento - sempre sem conteúdo - campo 14.

f) Tomando-se agora a estrutura padrão e com base nos relatórios, analisar quanto à presença ou ausência de conteúdo nos campos não marcados com 'X' e não considerados no item anterior. Após esta análise formamos novos segmentos e/ou aumentamos os segmentos já construídos.



Campos a considerar:

f.1) Campo 2 - por ter uma probabilidade de ocorrência com conteúdo de 0,981537 e por possuir características em comum com os campos 3 e 4, consideramos este campo sempre com conteúdo e o anexamos ao 2º segmento.

f.2) Campos 11 e 15 - por terem as probabilidades de ocorrência com conteúdo de 0,971850 e 0,987312 e por possuirem características em comum com os campos 1, 16, 17, consideramos estes campos sempre com conteúdo e os anexamos ao 1º segmento.

f.3) Campos 5, 9, 12 e 13 - com as probabilidades de ocorrerem com conteúdo de 0,003300, 0,001967, 0,059975 e 0,058025 respectivamente, estes campos foram considerados sempre sem conteúdo e anexados ao 3º segmento.

Nossos segmentos, assim, passaram a ter as seguintes constituições:

- 1º segmento - sempre com conteúdo - campos 1, 11, 15, 16 e 17.
- 2º segmento - sempre com conteúdo - campos 2, 3 e 4.
- 3º segmento - sempre com conteúdo - campos 5, 9, 12, 13 e 14.

g) Proceder de forma idêntica ao item "f" para os campos indefinidos quanto ao seu conteúdo (marcados com "x" na estrutura padrão).

Considerando, para os campos a serem analisados, o relacionamento lógico, características comuns e simultaneidade de ocorrência nas estruturas construimos, a partir dos três relatórios, os seguintes segmentos:

- 4º segmento - campos 7 e 10.
- 5º segmento - campo 6.
- 6º segmento - campo 8.
- 7º segmento - campos 18 e 19.

h) Como não há possibilidade de se concatenar qualquer um dos segmentos projetados, resta-nos organizá-los de forma que, da esquerda para a direita, haja uma passagem gradativa de segmentos sempre presentes aos segmentos sempre ausentes.

Posto isto, podemos apresentar o nosso registro segmentado como se segue:

IDENTIFICAÇÃO SEGMENTO	FORMATO	COMPRIMENTO (BYTES)	CAMPOS COMPONENTES	POSIÇÃO INICIAL
1	DECIMAL COMPACTADO	20	1	1
			11	8
			15	12
			16	17
			17	19
2	ALFANUMÉRICO	13	2	21
			3	22
			4	31
3	DECIMAL COMPACTADO	9	7	34
			10	39
4	ALFANUMÉRICO		20	43
			21	65
5	DECIMAL COMPACTADO	4	6	87
6	DECIMAL COMPACTADO	4	8	91
7	DECIMAL COMPACTADO	4	18	95
			19	97
8	DECIMAL COMPACTADO	14	5	99
			9	102
			12	107
			13	108
			14	109

VI-11

3. APLICAÇÃO DO PROGRAMA GERADOR DE CÓDIGOS (COMPØ4Ø)

3.1. DESCRIÇÃO DAS ENTRADAS

3.1.1. Arquivo Aplicação

O arquivo a ser trabalhado é o de Débitos dos Consumidores de uma empresa de energia elétrica, o qual apresenta as características abaixo:

- Número Total de registros no arquivo - 150000.
- Número Total de registros processados - 80000.
- Suporte do armazenamento - Fita.
- Label Standard IBM.

3.1.2. Arquivo Descrição dos Segmentos

Para a definição destes dados, que fornecem a este programa todos os subsídios necessários ao processamento dos segmentos nos registros do Arquivo Aplicação, consideremos a

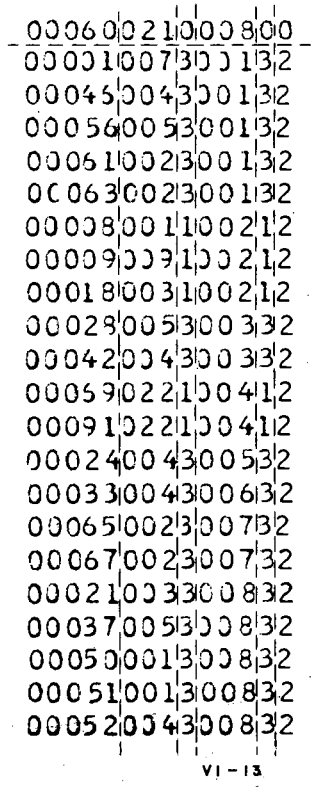
tabela a seguir, onde encontramos, para cada campo de cada segmento do registro, a sua descrição acompanhada da entrada correspondente neste arquivo.

DESCRIÇÃO DOS SEGMENTOS							ARQUIVOS					
NOME	CAMPOS COMPONENTES	FORMATO	POSIÇÃO REGISTRO ORIGINAL	TAM.	SINAL		POSIÇÃO REGISTRO ORIGINAL	TAMAN.	FORM.	IDENTIF. SEGM.	FORM. SEGM.	CÓDIGO SINAL
SEGMENTO 1	CAMPO 1	DECIMAL COMPACTADO	1	7	0 F		00001	007	3	001	3	2
	CAMPO 11	DECIMAL COMPACTADO	46	4	0 F		00046	004	3	001	3	2
	CAMPO 15	DECIMAL COMPACTADO	56	5	0 F		00056	005	3	001	3	2
	CAMPO 16	DECIMAL COMPACTADO	61	2	0 F		00061	002	3	001	3	2
	CAMPO 17	DECIMAL COMPACTADO	63	2	0 F		00063	002	3	001	3	2
SEGMENTO 2	CAMPO 2	ALFANUMÉRICO	8	1	40		00008	001	1	002	1	2
	CAMPO 3	ALFANUMÉRICO	9	9	40		00009	009	1	002	1	2
	CAMPO 4	ALFANUMÉRICO	18	3	40		00018	003	1	002	1	2
SEGMENTO 3	CAMPO 7	DECIMAL COMPACTADO	28	5	0 F		00028	005	3	003	3	2
	CAMPO 10	DECIMAL COMPACTADO	42	4	0 F		00042	004	3	003	3	2
SEGMENTO 4	CAMPO 20	ALFANUMÉRICO	69	22	40		00069	022	1	004	1	2
	CAMPO 21	ALFANUMÉRICO	91	22	40		00091	022	1	004	1	2
SEGMENTO 5	CAMPO 6	DECIMAL COMPACTADO	24	4	0 F		00024	004	3	005	3	2
SEGMENTO 6	CAMPO 4	DECIMAL COMPACTADO	33	4	0 F		00033	004	3	006	3	2
SEGMENTO 7	CAMPO 18	DECIMAL COMPACTADO	65	2	0 F		00065	002	3	007	3	2
	CAMPO 19	DECIMAL COMPACTADO	67	2	0 F		00067	002	3	007	3	2
SEGMENTO 8	CAMPO 5	DECIMAL COMPACTADO	21	3	0 F		00021	003	3	008	3	2
	CAMPO 9	DECIMAL COMPACTADO	37	5	0 F		00037	005	3	008	3	2
	CAMPO 12	DECIMAL COMPACTADO	50	1	0 F		00050	001	3	008	3	2
	CAMPO 13	DECIMAL COMPACTADO	51	1	0 F		00051	001	3	008	3	2
	CAMPO 14	DECIMAL COMPACTADO	52	4	0 F		00052	004	3	008	3	2

VI-12

Com base na tabela anterior, como podemos observar a seguir, foi preparada a massa real a ser utilizado pelo programa com a descrição dos segmentos no registro.

Chamamos a atenção, para o fato de que o primeiro cartão informa o fator de bloco (60), o número de campos no registro (21) e o número de segmentos definidos (8).



3.1.3. Arquivo Console

Em resposta às solicitações do programa foram fornecidas as seguintes informações:

- Data - 09/01/78.
- Limite de impressão dos relatórios - 10.000.
- Número de registros a processar - 80.000.
- Fator de bloco - 60.
- Comprimento do registro - 112.

3.2. DESCRIÇÃO DAS SAÍDAS

Pelo fato de se ter atingido aos 80.000 registros processados, como no programa anterior, a estabilidade estatística das informações manipuladas foi alcançada. Aqui apresentaremos apenas os últimos relatórios gerados.

M A P A E S T A T I S T I C O D O S S E G M E N T O S - 09/01/78 - PAG - 89

NUM SEG	FREQ. SEG.	PROBAB. SEG.	TAM SEG	PART.TOT. ARQ.	PART.COMP ARQ.	COMPR.ABS ARQ.	COMPR.REL ARQ.	COM PER	RAIO COMP ARQ.	DADOS NUM	TAM	DOS PRV	CAMPOS PRN	F O R M A T O
1	80000	0,189569	20	1600000	1600000				1,000000					
										1	7	1	1	DECIMAL COMPACT.
										2	4	46	8	DECIMAL COMPACT.
										3	5	56	12	DECIMAL COMPACT.
										4	2	61	17	DECIMAL COMPACT.
										5	2	63	19	DECIMAL COMPACT.
2	80000	0,189569	13	1040000	1040000				1,000000					
										1	1	8	21	ALFANUMERICO
										2	9	9	22	ALFANUMERICO
										3	3	18	31	ALFANUMERICO
3	73307	0,173709	9	720000	659763	60237	0,006722		1,091300					
										1	5	28	34	DECIMAL COMPACT.
										2	4	42	39	DECIMAL COMPACT.
4	77239	0,183169	44	3520000	3401156	118844	0,013263	1	1,034942					
										1	22	69	43	ALFANUMERICO
										2	22	91	65	ALFANUMERICO
5	59440	0,140850	4	320000	237760	82240	0,009178		1,345895					
										1	4	24	87	DECIMAL COMPACT.
6	25058	0,061747	4	320000	104232	215768	0,024081	2	3,070074					
										1	4	33	91	DECIMAL COMPACT.
7	20088	0,047600	4	320000	80352	239648	0,026746	2	3,982477					
										1	2	65	95	DECIMAL COMPACT.
										2	2	67	97	DECIMAL COMPACT.
8	5817	0,013784	14	1120000	81438	1038562	0,115910	11	13,752793					
										1	3	21	99	DECIMAL COMPACT.
										2	5	37	102	DECIMAL COMPACT.
										3	1	50	107	DECIMAL COMPACT.
										4	1	51	108	DECIMAL COMPACT.
										5	4	52	109	DECIMAL COMPACT.
	422039	0,999997	112	8960000	7204701	1755299								

CARACTER NAO PERMITIDO = FEDECIMAL COMPACT.000000002

Uma vez de posse da definição do registro e seus segmentos, este programa, através deste relatório, fornecerá ao usuário, todos os dados estatísticos e de compressão para cada segmento do projeto em análise, resultante da eliminação dos segmentos sem conteúdo da amostra analisada.

Esclarecidos os pontos acima, passamos a apresentar os resultados obtidos:

- Número Total de segmentos analisados = 8 * 80000 = 640.000.
- Número Total de segmentos com conteúdo = 422.009.
- Número Total de segmentos sem conteúdo = 217.991.

- Participação Total dos segmentos na amostra(tamanho original da amostra) = 8.960.000 bytes.
- Participação Compressa dos segmentos na amostra (tamanho compresso da amostra) = 7.204.701 bytes.
- Compressão Absoluta na amostra = 1.755.299 bytes.
- Compressão Relativa na amostra = 0,1959 (19,59%).

Comparando-se agora, para a amostra com os segmentos projetados, a Compressão Absoluta obtida - 1.755.299 bytes com a Compressão Absoluta máxima - 1.844.280 bytes e a Compressão Relativa obtida - 19,59% com a Compressão Relativa máxima - 20,58, encontramos por diferença 88.981 bytes e 0,99%.

Com estas diferenças, podemos concluir, ser bom o projeto do registro e seus segmentos, uma vez que o mesmo permite uma eliminação quase que total dos campos sem conteúdo, quando da aplicação do Método de Compressão por Segmentação à amostra.

3.2.2. Mapa Estatístico dos Alfabetos

Os relatórios que se seguem apresentam uma análise estatística do conteúdo dos segmentos não eliminados pelo processo de Segmentação e a compressão obtida pela aplicação a estes mesmos conteúdos, separadamente para cada segmento

do Código de Tamanho Variável HSF.

Por ser este relatório emitido para cada segmento, vamos abordá-los um a um, separadamente, apresentando ao final a compressão total obtida pela aplicação do código HSF.

Convém lembrar que para a obtenção das compressões Absolutas e Relativas, os valores a serem manipulados podem ser positivos ou negativos, caso a configuração de bits atribuída pelo código HSF seja superior à configuração original do carater.

Devemos ainda observar, que para todos os cálculos a seguir, consideramos a frequência de cada um dos elementos dos vários alfabetos existentes acrescida de 1. Isto por motivos de obtenção do código HSF o qual exige no nosso sistema uma frequência mínima de 1 e pela pouca representatitidade deste acréscimo no computo geral da compressão.

MAPA ESTATISTICO DOS ALFABETOS -:09/01/78 -:PAG -: 90

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. CAR	ORIG. ARQ	COMPR. CAR	COMPR. ARQ	COMP. CAR	ABSOL. ARQ	COMP. CAR	RELATIVA ARQ	COM. CAR	PER ARQ	RAIO COMP.
SEGMENTO NUMERO - 1			FORMATO - DECIMAL COMPACT.												
0 F0	1335859	0,41745424	0	0,4174	4	5343436	1	1335859	3	4007577	0,750000	0,055909	75	5	4,000
FF	399565	0,12486355	100	0,3745	4	1598260	3	1198595	1	399565	0,250000	0,005574	25		1,333
1 F1	249415	0,07794187	1010	0,3117	4	997660	4	997660		0					1,000
7 F7	219304	0,06853222	1011	0,2741	4	877216	4	877216		0					1,000
6 F6	167578	0,05236791	1100	0,2094	4	670312	4	670312		0					1,000
5 F5	159298	0,04978042	11010	0,2489	4	637192	5	796490	-1	-159298	0,250000	0,002222	25		0,800
8 F8	146725	0,04585137	11011	0,2292	4	586900	5	733625	-1	-146725	0,250000	0,002046	25		0,800
2 F2	141630	0,04425919	11100	0,2212	4	566520	5	708150	-1	-141630	0,250000	0,001975	25		0,800
4 F4	135382	0,04230670	11101	0,2115	4	541523	5	676910	-1	-135382	0,250000	0,001888	25		0,800
3 F3	123671	0,03864703	11110	0,1932	4	494684	5	618355	-1	-123671	0,250000	0,001725	25		0,800
9 F9	121148	0,03785859	111110	0,2271	4	434592	6	726388	-2	-242296	0,500000	0,003380	50		0,666
FC	436	0,00013624	1111110	0,0009	4	1744	7	3052	-3	-1308	0,750000	0,000018	75		0,571
FD	1	0,00000031	11111110	0,0000	4	4	8	8	-4	-4	1,000000		100		0,500
0 FL	1	0,00000031	11111111	0,0000	4	4	8	8	-4	-4	1,000000		100		0,500
	3200013	0,99999995		2,9191		12800052		9343228		3456824		0,048225			

CARACTER NAO PERMITIDO - DO ALFANUMERICO 000000002

a) Segmento 1 - formato Decimal Compactado

a.1) Caracteres não permitidos no alfabeto

2 X 'FE' foram encontrados, ou seja, 1 número com sinal "E" (uma vez que este valor é acrescido de 1 por razões de obtenção do código HSF).

a.2) Número de caracteres analisados

- Número de segmentos com conteúdo	-	(80000)
		*
- Tamanho em bytes do segmento	-	(20)
		*
- Número de caracteres por byte	-	(2)
		+
- Número de elementos do alfabeto	-	(14)
		-
- Número de elementos não permitidos no alfabeto	-	(1)
		=
	Total -	3.200.013 caracteres

a.3) Economia em bits por caráter

- Número de bits por caráter formato Decimal Compactado	-	(4 bits/caráter)
		—
- Custo médio do código HSF gerado	-	(2,9191 bits/caráter)
		=
	Economia Total -	1,0809 bits/caráter

a.4) Economia em bits e bytes para o segmento na amostra

- Comprimento Original do segmento na amostra	-	(12.800.052 bits)
		—
- Comprimento Compresso do segmento na amostra	-	(9.343.228 bits)
		=
- Compressão Absoluta do segmento na amostra	-	3.456.824 bits ou 432.103 bytes.

a.5) Compressão Relativa do segmento na amostra

0,048225 ou 4,82%

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. CAR	ORIG. ARQ	COMPR. CAR	COMPR. ARQ	COMP. CAR	ABSOL. ARQ	COMP. CAR	RELATIVA ARQ	COM.PER CAR	RAID ARQ	COMP.
SEGMENTO NUMERO - 2			FORMATO - ALFANUMERICO												
0 F0	233033	0,22405748	00	0,4481	8	1864264	2	466066	6	1398198	0,750000	0,019506	75	1	4,000
1 F1	180200	0,17325940	010	0,5197	8	1441600	3	540500	5	901000	0,625000	0,012569	62	1	2,666
3 F3	139309	0,13394336	011	0,4018	8	1114472	3	417927	5	696545	0,625000	0,009717	62		2,666
2 F2	121944	0,11724719	100	0,3517	8	975552	3	365832	5	609720	0,625000	0,008506	62		2,666
8 F8	72940	0,07013063	1010	0,2805	8	583520	4	291760	4	291760	0,500000	0,004070	50		2,000
9 F9	67349	0,06475498	1011	0,2590	8	538792	4	269396	4	269396	0,500000	0,003758	50		2,000
40	51287	0,04931162	1100	0,1972	8	410296	4	205148	4	205148	0,500000	0,002861	50		2,000
5 F5	47167	0,04535031	1101	0,1814	8	377336	4	188668	4	188668	0,500000	0,002632	50		2,000
6 F6	44264	0,04255912	1110	0,1702	8	354112	4	177056	4	177056	0,500000	0,002470	50		2,000
4 F4	42587	0,04094671	11110	0,2047	8	340696	5	212935	3	127761	0,375000	0,001782	37		1,500
7 F7	39930	0,03839205	111110	0,2303	8	319440	6	239580	2	79860	0,250000	0,001114	25		1,333
4A	1	0,00000096	111111100000	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
. 4B	1	0,00000096	111111100001	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
(4D	1	0,00000096	111111100010	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
+ 4E	1	0,00000096	111111100011	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
| 4F	1	0,00000096	111111100100	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
& 50	1	0,00000096	111111100101	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
$ 5B	1	0,00000096	111111100110	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
* 5C	1	0,00000096	111111100111	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
) 5D	1	0,00000096	111111101000	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
; 5E	1	0,00000096	111111101001	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
- 60	1	0,00000096	111111101010	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
| 61	1	0,00000096	111111101011	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
6A	1	0,00000096	111111101100	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
, 6B	1	0,00000096	111111101101	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
% 6C	1	0,00000096	111111101110	0,0000	8	8	11	11	-3	-3	0,375000		37		0,727
6D	1	0,00000096	111111011110	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
: 7A	1	0,00000096	111111011111	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
# 7B	1	0,00000096	111111100000	0,0000	8	8	12	12	-4	-4	0,500000		50		0,565
@ 7C	1	0,00000096	111111100001	0,0000	8	8	12	12	-4	-4	0,500000		50		0,565
7D	1	0,00000096	111111100010	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
= 7E	1	0,00000096	111111100011	0,0000	8	8	12	12	-4	-4	0,500000		50		0,565
7F	1	0,00000096	111111100100	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
A C1	1	0,00000096	111111100101	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
B C2	1	0,00000096	111111100110	0,0000	8	8	12	12	-4	-4	0,500000		50		0,565
C C3	1	0,00000096	111111100111	0,0000	8	8	12	12	-4	-4	0,500000		50		0,565
D C4	1	0,00000096	111111101000	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
E C5	1	0,00000096	111111101001	0,0000	8	8	12	12	-4	-4	0,500000		50		0,565
F C6	1	0,00000096	111111101010	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
G C7	1	0,00000096	111111101011	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
H C8	1	0,00000096	111111101100	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
I C9	1	0,00000096	111111101101	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
J D1	1	0,00000096	111111101110	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
K D2	1	0,00000096	111111101111	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
L D3	1	0,00000096	111111110000	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
M D4	1	0,00000096	111111110001	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
N D5	1	0,00000096	111111110010	0,0000	8	8	12	12	-4	-4	0,500000		50		0,565
O D6	1	0,00000096	111111110011	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666

b) Segmento 2 - formato alfanumérico

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. CAR	ORIG. ARQ	COMPR. CAR	COMPR. ARQ	COMP. CAR	ABSOL. ARQ	COMP. CAR	RELATIVA ARQ	COM.PER CAR	RAID ARQ	COMP.
SEGMENTO NUMERO - 2			FORMATO - ALFANUMERICO												
P D7	1	0,00000096	111111110100	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
Q D8	1	0,00000096	111111110101	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
R D9	1	0,00000096	111111110110	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
S E2	1	0,00000096	111111110111	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
T E3	1	0,00000096	111111111000	0,0000	8	8	12	12	-4	-4	0,500000		50		0,565
U E4	1	0,00000096	111111111001	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
V E5	1	0,00000096	111111111010	0,0000	8	8	12	12	-4	-4	0,500000		50		0,565
W E6	1	0,00000096	111111111011	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
X E7	1	0,00000096	111111111100	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
Y E8	1	0,00000096	111111111101	0,0000	8	8	12	12	-4	-4	0,500000		50		0,566
Z E9	1	0,00000096	111111111110	0,0000	8	8	12	12	-4	-4	0,500000		50		0,565
D FL	1	0,00000096	111111111111	0,0000	8	8	12	12	-4	-4	0,500000		50		0,666
	1040059	0,99999989		3,2446		8320472		3375541		4944931		0,068985			

b.1) Caracteres não permitidos no alfabeto

2 X '00' foram encontrados, ou seja, 1 caráter (uma vez que esta quantidade é acrescida de 1 por razões da obtenção do código HSF).

b.2) Número de caracteres analisados

- Número de segmentos com conteúdo	-	(80000)
		*
- Tamanho em bytes do segmento	-	(13)
		*
- Número de caracteres por byte	-	(1)
		+
- Número de elementos do alfabeto	-	(60)
		-
- Número de elementos não permitidos no alfabeto	-	(1)
		=
Total	-	1.040.059 caracteres

b.3) Economia em bits por caráter

- Número de bits por caráter formato Alfanumérico	-	(8 bits/caráter)
		—
- Custo médio do código HSF gerado	-	(3,2446 bits/caráter)
		=
Economia Total	-	4,7554 bits/caráter

b.4) Economia em bits e bytes para o segmento na amostra

- Comprimento Original do segmento na amostra	-	(8.320.472 bits)
		—
- Comprimento Compresso do segmento na amostra	-	(3.375.541 bits)
		=
- Compressão Absoluta do segmento na amostra	-	4.944.931 bits ou 618.116 bytes.

b.5) Compressão Relativa do segmento na amostra

0,068986 ou 6,89%

c) Segmento 3 - formato Decimal Compactado

MAPA ESTATISTICO DOS ALFABETOS - 09/01/78 - PAG - 93

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. CAR	ORIG. ARQ	COMPR. CAR	COMPR. ARQ	COMP. CAR	ABSOL. ARQ	COMP. CAR	RELATIVA ARQ	COM. CAR	PER ARQ	RAIO COMP.
SEGMENTO NUMERO - 3			FORMATO - DECIMAL COMPACT.												
0 F0	745683	0,56510829	0	0,5651	4	2982732	1	745583	3	2237049	0,750000	0,031208	75	3	4,000
FF	145745	0,11045136	100	0,3313	4	582980	3	437235	1	145745	0,250000	0,002033	25		1,333
1 F1	98871	0,07492838	1010	0,2997	4	395484	4	395484		0					1,000
4 F4	53930	0,04087030	1011	0,1634	4	215720	4	215720		0					1,000
2 F2	52295	0,03963123	1100	0,1585	4	209180	4	209180		0					1,000
3 F3	50668	0,03839822	11010	0,1919	4	202672	5	253340	-1	-50668	0,250000	0,000706	25		0,800
8 F8	46914	0,03555329	11011	0,1777	4	187656	5	234570	-1	-46914	0,250000	0,000654	25		0,800
5 F5	42516	0,03222031	11100	0,1611	4	170064	5	212580	-1	-42516	0,250000	0,000593	25		0,800
6 F6	31480	0,02385679	11101	0,1192	4	125920	5	157400	-1	-31480	0,250000	0,000439	25		0,800
7 F7	26550	0,02012064	11110	0,1006	4	106200	5	132750	-1	-26550	0,250000	0,000370	25		0,800
9 F9	24015	0,01819952	111110	0,1091	4	96060	6	144090	-2	-48030	0,500000	0,000670	50		0,666
FC	871	0,00065007	1111110	0,0046	4	3484	7	6097	-3	-2613	0,750000	0,000036	75		0,571
FD	1	0,00000075	11111110	0,0000	4	4	8	8	-4	-4	1,000000		100		0,500
D FL	1	0,00000075	11111111	0,0000	4	4	8	8	-4	-4	1,000000		100		0,500
	1319540	0,99999990		2,3822		5278160		3144145		2134015		0,029771			

CARACTER NAO PERMITIDO - 44ALFANUMERICO 000000002
CARACTER NAO PERMITIDO - 4CALFANUMERICO 000000015
CARACTER NAO PERMITIDO - 63ALFANUMERICO 000000003
CARACTER NAO PERMITIDO - 96ALFANUMERICO 000000007

c.1) Caracteres não permitidos no alfabeto.

Zero.

c.2) Número de caracteres analisados

- Número de segmentos com conteúdo	-	(73.307)
		*
- Tamanho em bytes do segmento	-	(9)
		*
- Número de caracteres por byte	-	(2)
		+
- Número de elementos do alfabeto	-	(14)
		-
- Número de elementos não permitidos no alfabeto	-	(Ø)
		=
Total	-	1.319.540 caracteres

c.3) Economia em bits por caráter

- Número de bits por caráter formato Decimal Compactado	-	(4 bits/caráter)
		—
- Custo médio do código HSF gerado	-	(2,3822 bits/caráter)
		=
Economia Total	-	1,6178 bits/caráter

c.4) Economia em bits e bytes para o segmento na amostra

- Comprimento Original do segmento na amostra	-	(5.278.160 bits)
		—
- Comprimento Compresso do segmento na amostra	-	(3.144.145 bits)
		=
- Compressão Absoluta do segmento na amostra	-	2.134.015 bits ou 266.751 bytes.

c.5) Compressão Relativa do segmento na amostra

0,029771 ou 2,97%

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR.ORIG. CAR	COMPR.ORIG. ARQ	COMPR.COMPR. CAR	COMPR.COMPR. ARQ	COMP. CAR	ABSOL. ARQ	COMP. CAR	RELATIVA ARQ	COM.PER CAR	ARQ	RAIO COMP.
SEGMENTO NUMERO - 4			FORMATO - ALFANUMERICO												
40	1013094	0,29786430	00	0,5957	8	8104752	2	2026188	6	6078564	0,750000	0,084801	75	8	4,000
A C1	300839	0,08845102	0100	0,3538	8	2406712	4	1203356	4	1203356	0,500000	0,016787	50	1	2,000
O D6	200018	0,05880818	0101	0,2352	8	1600144	4	800072	4	800072	0,500000	0,011161	50	1	2,000
E C5	178853	0,05258537	0110	0,2103	8	1430824	4	715412	4	715412	0,500000	0,009980	50		2,000
R D9	169645	0,04987808	0111	0,1995	8	1357160	4	678580	4	678580	0,500000	0,009466	50		2,000
I C9	168376	0,04950498	1000	0,1980	8	1347003	4	673504	4	673504	0,500000	0,009395	50		2,000
S E2	147676	0,04341888	10010	0,2170	8	1181408	5	738380	3	443028	0,375000	0,006180	37		1,500
N D5	103300	0,03037169	10011	0,1518	8	826400	5	516500	3	309900	0,375000	0,004323	37		1,500
C C3	99491	0,02925179	10100	0,1462	8	795928	5	497455	3	298473	0,375000	0,004163	37		1,600
M D4	88088	0,02589914	10101	0,1294	8	704704	5	440440	3	264264	0,375000	0,003686	37		1,600
L D3	87372	0,02568863	10110	0,1284	8	698976	5	436860	3	262116	0,375000	0,003656	37		1,500
D C4	85777	0,02521968	10111	0,1260	8	686216	5	428885	3	257331	0,375000	0,003589	37		1,500
O F0	82985	0,02439879	11000	0,1219	8	663880	5	414925	3	248955	0,375000	0,003473	37		1,600
T E3	68299	0,02008089	110010	0,1204	8	546392	6	409794	2	136598	0,250000	0,001905	25		1,333
U E4	55968	0,01645540	110011	0,0987	8	447744	6	335808	2	111936	0,250000	0,001561	25		1,333
P D7	49517	0,01455871	110100	0,0873	8	396136	6	297102	2	99034	0,250000	0,001381	25		1,333
F C6	47190	0,01387454	110101	0,0832	8	377520	6	283140	2	94380	0,250000	0,001316	25		1,333
1 F1	46626	0,01370871	110110	0,0822	8	373008	6	279756	2	93252	0,250000	0,001300	25		1,333
V E5	43618	0,01282432	110111	0,0769	8	348944	6	261708	2	87236	0,250000	0,001217	25		1,333
B C2	37526	0,01103318	111000	0,0661	8	300208	6	225156	2	75052	0,250000	0,001047	25		1,333
G C7	33184	0,00975657	1110010	0,0682	8	265472	7	232288	1	33184	0,125000	0,000462	12		1,142
2 F2	27662	0,00813302	1110011	0,0569	8	221296	7	193634	1	27662	0,125000	0,000385	12		1,142
J D1	26649	0,00783519	1110100	0,0548	8	213192	7	186543	1	26649	0,125000	0,000371	12		1,142
5 F5	24439	0,00718541	1110101	0,0502	8	195512	7	171073	1	24439	0,125000	0,000340	12		1,142
H C8	24438	0,00718512	1110110	0,0502	8	195504	7	171066	1	24438	0,125000	0,000340	12		1,142
4 F4	24178	0,00710868	1110111	0,0497	8	193424	7	169246	1	24178	0,125000	0,000337	12		1,142
3 F3	23969	0,00704723	1111000	0,0493	8	191752	7	167783	1	23969	0,125000	0,000334	12		1,142
6 F6	21157	0,00622046	1111001	0,0435	8	169256	7	148099	1	21157	0,125000	0,000295	12		1,142
. 4B	20623	0,00606346	1111010	0,0424	8	164984	7	144361	1	20623	0,125000	0,000287	12		1,142
7 F7	20477	0,00602053	1111011	0,0421	8	163815	7	143339	1	20477	0,125000	0,000285	12		1,142
8 F8	19366	0,00569388	1111100	0,0398	8	154923	7	135562	1	19366	0,125000	0,000270	12		1,142
9 F9	17628	0,00518288	1111101	0,0362	8	141024	7	123396	1	17628	0,125000	0,000245	12		1,142
Z E9	13991	0,00411355	11111100	0,0329	8	111928	8	111928		0					1,000
60	11685	0,00343555	11111101	0,0274	8	93480	8	93480		0					1,000
Q D8	5687	0,00167206	111111100	0,0150	8	45495	9	51183	-1	-5687	0,125000	0,000079	12		0,888
X E7	3235	0,00095113	111111101	0,0085	8	25880	9	29115	-1	-3235	0,125000	0,000045	12		0,888
(61	3154	0,00092732	111111110	0,0083	8	25232	9	28386	-1	-3154	0,125000	0,000044	12		0,888
Y E8	2609	0,00076708	1111111110	0,0075	8	20872	10	25090	-2	-5218	0,250000	0,000072	25		0,800
W E6	1605	0,00047189	11111111110	0,0051	8	12840	11	17555	-3	-4815	0,375000	0,000067	37		0,727
K D2	1040	0,00030577	111111111110	0,0036	8	8320	12	12480	-4	-4160	0,500000	0,000058	50		0,666
& 5D	80	0,00002352	1111111111110	0,0003	8	640	13	1340	-5	-400	0,625000	0,000005	62		0,615
* 5C	46	0,00001352	11111111111110	0,0001	8	368	14	644	-6	-276	0,750000	0,000003	75		0,571
, 63	10	0,00000294	1111111111111100	0,0000	8	80	15	160	-8	-80	1,000000	0,000001	100		0,500
7D	4	0,00000117	11111111111111010	0,0000	8	32	17	68	-9	-36	1,125000		112		0,470
$ 5B	2	0,00000058	11111111111111110110	0,0000	8	16	18	36		-20	1,250000		125		0,444
% 6C	2	0,00000058	11111111111111110111	0,0000	8	16	18	36		-20	1,250000		125		0,444
# 78	2	0,00000058	11111111111111111000	0,0000	8	16	18	36		-20	1,250000		125		0,444
4A	1	0,00000029	11111111111111111001	0,0000	8	8	18	18		-10	1,250000		125		0,444

ALF.	FREQ.	PROB.	G R U P O	CUSTO.	COMPR. ORIG. CAR	ARQ	COMPR. COMPR. CAR	ARQ	COMP. CAR	ABSOL. ARQ	COMP. CAR	RELATIVA ARQ	COM.PER CAR	ARQ	RAIO COMP.
SEGMENTO NUMERO - 4			FORMATO - ALFANUMERICO												
{ 4D	1	0,00000029	11111111111111110100	0,0000	8	8	19	19	-1	-11	1,375000		137		0,421
+ 4E	1	0,00000029	11111111111111110101	0,0000	8	8	19	19	-1	-11	1,375000		137		0,421
| 4F	1	0,00000029	11111111111111110110	0,0000	8	8	19	19	-1	-11	1,375000		137		0,421
} 5D	1	0,00000029	11111111111111110111	0,0000	8	8	19	19	-1	-11	1,375000		137		0,421
; 5E	1	0,00000029	11111111111111111000	0,0000	8	8	19	19	-1	-11	1,375000		137		0,421
6A	1	0,00000029	11111111111111111001	0,0000	8	8	19	19	-1	-11	1,375000		137		0,421
6D	1	0,00000029	11111111111111111010	0,0000	8	8	19	19	-1	-11	1,375000		137		0,421
: 7A	1	0,00000029	11111111111111111011	0,0000	8	8	19	19	-1	-11	1,375000		137		0,421
@ 7C	1	0,00000029	11111111111111111100	0,0000	8	8	19	19	-1	-11	1,375000		137		0,421
= 7E	1	0,00000029	11111111111111111101	0,0000	8	8	19	19	-1	-11	1,375000		137		0,421
7F	1	0,00000029	11111111111111111110	0,0000	8	8	19	19	-1	-11	1,375000		137		0,421
D FL	1	0,00000029	11111111111111111111	0,0000	8	8	19	19	-1	-11	1,375000		137		0,421
	3401193	0,99999972		4,1201		27209544		14021994		13187550		0,183978			

d.1) Caracteres não permitidos no alfabeto

- 2 x '44' - 1
- 15 x '4C' - 14
- 3 x '63' - 2
- 7 x '96' - 6

Total - 23 caracteres.

d.2) Número de caracteres analisados

- Número de segmentos com conteúdo - (77.299)
  *
- Tamanho em bytes do segmento - (44)
  *
- Número de caracteres por byte - (1)
  +
- Número de elementos do alfabeto - (60)
  -
- Número de caracteres não permitidos no alfabeto- (23)
  =

Total - 3.401.193 caracteres

d.3) Economia em bits por caráter

- Número de bits por caráter formato alfanumérico - (8 bits/caráter)
  —
- Custo médio do código HSF gerado - (4,1201 bits/caráter)
  =

Economia Total - 3,8799 bits/caráter

d.4) Economia em bits e bytes para o segmento na amostra

- Comprimento Original do segmento na amostra -(27.209.544 bits)
  —
- Comprimento Compresso do segmento na amostra -(14.021.994 bits)
  =
- Compressão Absoluta do segmento na amostra - 13.187.550 bits ou 1.648.443 bytes.

d.5) Compressão Relativa do segmento na amostra

0,183978 ou 18,39%

e) Segmento 5 - formato Decimal Compactado

MAPA ESTATISTICO DOS ALFABETOS - 09/01/78 - PAG - 96

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. CAR	ORIG. ARQ	COMPR. CAR	COMPR. ARQ	COMP. CAR	ABSOL. ARQ	COMP. CAR	RELATIVA ARQ	COM. CAR	PER ARQ	RAIO COMP.
SEGMENTO NUMERO - 5			FORMATO - DECIMAL COMPACT.												
0 F0	289437	0,60865679	0	0,6086	4	1157748	1	289437	3	868311	0,750000	0,012113	75	1	4,000
FF	59061	0,12419932	100	0,3725	4	236244	3	177183	1	59061	0,250000	0,000823	25		1,333
1 F1	26492	0,05571000	1010	0,2228	4	105963	4	105968		0					1,000
2 F2	17923	0,03769025	1011	0,1507	4	71692	4	71592		0					1,000
3 F3	17154	0,03607313	1100	0,1442	4	68616	4	68616		0					1,000
4 F4	12914	0,02715683	11010	0,1357	4	51656	5	54570	-1	-12914	0,250000	0,000180	25		0,800
5 F5	11898	0,02502029	11011	0,1251	4	47592	5	59490	-1	-11898	0,250000	0,000165	25		0,800
6 F6	10906	0,02293421	11100	0,1146	4	43624	5	54530	-1	-10906	0,250000	0,000152	25		0,800
8 F8	10045	0,02112362	11101	0,1056	4	40180	5	50225	-1	-10045	0,250000	0,000140	25		0,800
7 F7	9942	0,02090702	11110	0,1045	4	39763	5	49710	-1	-9942	0,250000	0,000138	25		0,800
9 F9	9379	0,01972309	111110	0,1183	4	37516	6	56274	-2	-18758	0,500000	0,000261	50		0,565
FC	381	0,00080120	1111110	0,0056	4	1524	7	2567	-3	-1143	0,750000	0,000015	75		0,571
FD	1	0,00000210	11111110	0,0000	4	4	8	8	-4	-4	1,000000		100		0,500
D FL	1	0,00000210	11111111	0,0000	4	4	8	8	-4	-4	1,000000		100		0,500
	475534	0,99999995		2,2082		1902136		1050378		851758		0,011882			

e.1) Caracteres não permitidos no alfabeto

Zero.

e.2) Número de caracteres analisados

- Número de segmentos com conteúdo - (59.440)
*
- Tamanho em bytes do segmento - (4)
*
- Número de caracteres por byte - (2)
+
- Número de elementos do alfabeto - (14)
-
- Número de caracteres não permitidos no alfabeto- (Ø)
=

Total - 475.534 caracteres

e.3) Economia em bits por caráter

- Número de bits por caráter formato Decimal Compactado - (4 bits/caráter)
—
- Custo médio do código HSF gerado - (2,2082 bits/caráter)
=

Economia Total - 1,7918 bits/caráter

e.4) Economia em bits e bytes para o segmento na amostra

- Comprimento Original do segmento na amostra - (1.902.136 bits)
—
- Comprimento Compresso do segmento na amostra - (1.050.378 bits)
=
- Compressão Absoluta do segmento na amostra - 851.758 bits ou 106.469 bytes.

e.5) Compressão Relativa do segmento na amostra

0,011882 ou 1,18%

M A P A E S T A T I S T I C O D O S A L F A B E T O S - 09/01/78 - PAG - 97

ALF.	FREQ.	PROB.	G R U P O	CUSTO	COMPR. CAR	ORIG. ARQ	COMPR. CAR	COMPR. ARQ	COMP. CAR	ABSOL. ARQ	COMP. CAR	RELATIVA ARQ	COM. CAR	PER ARQ	RAIO COMP.
SEGMENTO NUMERO - 6			FORMATO - DECIMAL COMPACT.												
0 F0	95484	0,45800516	0	0,4580	4	381936	1	95484	3	286452	0,750000	0,003996	75		4,000
FF	26059	0,12499640	100	0,3749	4	104236	3	78177	1	26059	0,250000	0,000363	25		1,333
1 F1	16952	0,08131313	1010	0,3252	4	67808	4	67808		0					1,000
2 F2	11120	0,05333896	1011	0,2133	4	44480	4	44480		0					1,000
4 F4	8796	0,04219150	1100	0,1687	4	35184	4	35184		0					1,000
6 F6	8737	0,04214833	11010	0,2107	4	35143	5	43935	-1	-8787	0,250000	0,000122	25		0,800
5 F5	8784	0,04213394	11011	0,2106	4	35135	5	43920	-1	-8784	0,250000	0,000122	25		0,800
7 F7	8458	0,04057022	11100	0,2028	4	33832	5	42290	-1	-8458	0,250000	0,000117	25		0,800
3 F3	8347	0,04003779	11101	0,2001	4	33388	5	41735	-1	-8347	0,250000	0,000116	25		0,800
8 F8	8151	0,03909765	11110	0,1954	4	32604	5	40755	-1	-8151	0,250000	0,000113	25		0,800
9 F9	7537	0,03615249	111110	0,2169	4	30148	6	45222	-2	-15074	0,500000	0,000210	50		0,666
FC	1	0,00000479	1111110	0,0000	4	4	7	7	-3	-3	0,750000		75		0,571
FD	1	0,00000479	11111110	0,0000	4	4	8	8	-4	-4	1,000000		100		0,500
0 FL	1	0,00000479	11111111	0,0000	4	4	8	8	-4	-4	1,000000		100		0,500
	208478	0,99999994		2,7766		833912		579013		254899		0,003556			

f.1) Caracteres não permitidos no alfabeto

Zero.

f.2) Número de caracteres analisados

- Número de segmentos com conteúdo - (26.058)
*
- Tamanho em bytes do segmento (4)
*
- Número de caracteres por byte (2)
+
- Número de elementos do alfabeto (14)
-
- Número de caracteres não permitidos no alfabeto- (Ø)
=

Total - 208.478 caracteres

f.3) Economia em bits por caráter

- Número de bits por caráter formato Decimal Compactado - (4 bits/caráter)
—
- Custo médio do código HSF gerado - (2,7766 bits/caráter)
=

Economia Total- 1,2234 bits/caráter

f.4) Economia em bits e bytes para o segmento na amostra

- Comprimento Original do segmento na amostra - (833.912 bits)
—
- Comprimento Compresso do segmento na amostra - (579.013 bits)
=
- Compressão Absoluta do segmento na amostra - 254.899 bits

ou 31.862 bytes.

f.5) Compressão Relativa do segmento na amostra

0,003556 ou 0,35% (0%)

g) Segmento 7 - formato Decimal Compactado

MAPA ESTATISTICO DOS ALFABETOS - 09/01/78 - PAG - 98

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. CAR	ORIG. ARQ	COMPR. CAR	COMPR. ARQ	COMP. CAR	ABSOL. ARQ	COMP. CAR	RELATIVA ARQ	COM.PER CAR ARQ	RAIO COMP.
SEGMENTO NUMERO - 7			FORMATO - DECIMAL COMPACT.											
0 F0	72095	0,44858074	0	0,4485	4	288380	1	72095	3	216285	0,750000	0,003017	75	4,000
FF	39307	0,24457123	10	0,4891	4	157228	2	78614	2	78614	0,500000	0,001096	50	2,000
2 F2	13874	0,08632511	1100	0,3453	4	55495	4	55496		0				1,000
1 F1	10663	0,06634602	1101	0,2653	4	42652	4	42552		0				1,000
7 F7	9314	0,05795243	11100	0,2897	4	37255	5	46570	-1	-9314	0,250000	0,000129	25	0,800
8 F8	5855	0,03643026	11101	0,1821	4	23420	5	29275	-1	-5855	0,250000	0,000081	25	0,800
5 F5	3871	0,02408566	11110	0,1204	4	15484	5	19355	-1	-3871	0,250000	0,000054	25	0,800
4 F4	2601	0,01618362	111110	0,0971	4	10404	6	15606	-2	-5202	0,500000	0,000072	50	0,565
6 F6	899	0,00559364	11111100	0,0447	4	3595	8	7192	-4	-3596	1,000000	0,000050	100	0,500
3 F3	884	0,00550031	11111101	0,0440	4	3536	8	7072	-4	-3536	1,000000	0,000049	100	0,500
FC	871	0,00541943	11111110	0,0433	4	3484	8	6968	-4	-3484	1,000000	0,000048	100	0,500
9 F9	482	0,00299904	111111110	0,0269	4	1928	9	4338	-5	-2410	1,250000	0,000033	125	0,444
FD	1	0,00000622	1111111110	0,0000	4	4	10	10	-6	-6	1,500000		150	0,400
D FL	1	0,00000622	1111111111	0,0000	4	4	10	10	-6	-6	1,500000		150	0,400
	160718	0,99999993		2,3964		642872		385253		257619		0,003594		

g.1) Caracteres não permitidos no alfabeto

Zero.

g.2) Número de caracteres analisados

- Número de segmentos com conteúdo - (20.088)
  *
- Tamanho em bytes do segmento - (4)
  *
- Número de caracteres por byte - (2)
  +
- Número de elementos do alfabeto - (14)
  -
- Número de caracteres não permitidos no alfabeto- (Ø)
  =

Total - 160.718 caracteres

g.3) Economia em bits por caráter

- Número de bits por caráter formato Decimal Compactado - (4 bits/caráter)
  —
- Custo médio do código HSF gerado - (2,3964 bits/caráter)
  =

Economia Total - 1,6036 bits/caráter

g.4) Economia de bits e bytes para o segmento na amostra

- Comprimento Original do segmento na amostra - (642.872 bits)
  —
- Comprimento Compresso do segmento na amostra - (382.253 bits)
  =
- Compressão Absoluta do segmento na amostra - 257.619 bits
  ou 32.202 bytes.

g.5) Compressão Relativa do segmento na amostra

0,003594 ou 0,35% (0%)

h) Segmento 8 - formato Decimal Compactado

MAPA ESTATISTICO DOS ALFABETOS - 09/01/78 - PAG - 99

ALF.	FREQ.	PROB.	GRUPO	CUSTO	COMPR. ORIG. CAR	COMPR. ORIG. ARQ	COMPR. COMPR. CAR	COMPR. COMPR. ARQ	COMP. ABSOL. CAR	COMP. ABSOL. ARQ	COMP. RELATIVA CAR	COMP. RELATIVA ARQ	COM.PER CAR	COM.PER ARQ	RAIO COMP.
SEGMENTO NUMERO - 8			FORMATO - DECIMAL COMPACT.												
0 F0	123134	0,75593345	0	0,7559	4	492536	1	123134	3	369402	0,750000	0,005153	75		4,000
FF	23269	0,14285100	10	0,2857	4	93075	2	46539	2	46538	0,500000	0,000649	50		2,000
FC	5818	0,03571735	1100	0,1428	4	23272	4	23272		0					1,000
2 F2	1445	0,00887101	11010	0,0443	4	5780	5	7225	-1	-1445	0,250000	0,000020	25		0,800
4 F4	1293	0,00793787	11011	0,0396	4	5172	5	6465	-1	-1293	0,250000	0,000018	25		0,800
1 F1	1238	0,00760022	11100	0,0380	4	4952	5	6190	-1	-1238	0,250000	0,000017	25		0,800
6 F6	1186	0,00728098	111010	0,0436	4	4744	6	7116	-2	-2372	0,500000	0,000033	50		0,565
5 F5	1157	0,00710295	111011	0,0426	4	4628	6	6942	-2	-2314	0,500000	0,000032	50		0,666
8 F8	1148	0,00704770	111100	0,0422	4	4592	6	6888	-2	-2296	0,500000	0,000032	50		0,666
3 F3	1137	0,00698017	111101	0,0418	4	4548	6	6822	-2	-2274	0,500000	0,000031	50		0,565
9 F9	1036	0,00636012	111110	0,0381	4	4144	6	6216	-2	-2072	0,500000	0,000028	50		0,565
7 F7	1027	0,00630486	1111110	0,0441	4	4108	7	7189	-3	-3081	0,750000	0,000042	75		0,571
FD	1	0,00000613	11111110	0,0000	4	4	8	8	-4	-4	1,000000		100		0,500
D FL	1	0,00000613	11111111	0,0000	4	4	8	8	-4	-4	1,000000		100		0,500
	162890	0,99999994		1,5587		651560		254013		397547		0,005545			

h.1) Caracteres não permitidos no alfabeto

Zero.

h.2) Número de caracteres analisados

- Número de segmentos com conteúdo - (5.817)
*
- Tamanho em bytes do segmento - (14)
*
- Número de caracteres por byte - (2)
+
- Número de elementos do alfabeto - (14)
-
- Número de caracteres não permitidos no alfabeto- Ø
=

Total - 162.890 caracteres

h.3) Economia em bits por caráter

- Número de bits por caráter formato Decimal Compactado - (4 bits/caráter)
—
- Custo médio do código HSF gerado - (1,5587 bits/caráter)
=

Economia Total - 2,4413 bits/caráter

h.4) Economia em bits e bytes para o segmento na amostra

- Comprimento Original do segmento na amostra - (651.560 bits)
—
- Comprimento Compresso do segmento na amostra - (254.013 bits)
=
- Compressão Absoluta do segmento na amostra - 397.547 bits

ou 49.693 bytes.

h.5) Compressão Relativa do segmento na amostra

0,005546 ou 0,55%.(0%)

3.2.3. Arquivo de Códigos

Complementando a sua execução, este programa gera o Arquivo de Códigos a ser utilizado pelo programa de conversão.

Este arquivo, como já vimos, contem os seguintes dados:

- Tabela de Descrição dos Segmentos.
- Tabela de Descrição dos Campos.
- Padrão de Ausência.
- Tabela de Prefixos.
- Tabela de Códigos.
- Tabela de Caracteres,

os quais podem ser observados a seguir.

```
* * * * DEVICE  281  SYS019,  MODE IS  9 TRACK  1600 BPI        BLOCK    1   DATA  1024                                  * * * *

REC      1   DATA  1024   CHAR                          -          &                    8                       476   2 K - 9 < D  OA
                          ZONE   10000000000000301000002022000060300000506000008070000 0C08000C0F0A00444444010040FFF441F7D266F0D2C75FCA
                          NUMR   35300000C30008028230180CB200100E31300200631302808323020 0AD5302800000000005F01044760042028089 02F498010
                                 1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR         12 K - 9 K D  0A  " 0  K - D:K  @D   L & K   A & L   A  1  K & L   @ 1K - D:   & 1 0
                          ZONE   0851C50FF7D266F1D2C85FCA079F30D066C7D067385002500550CA50D251C00F50D650D251C70FD066C751C50F5FC0010040
                          NUMR   7F800871212309912E4980147F2000210944213C478030002480180030801471802400308 00C7121094A80007130045F0104
                               101...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   487   2 K - 9 K D  0A          1 0  2 K - 9 < D& D * " 0-    @ 1 0         492   2 K -   K D&3 - K  0-
                          ZONE   FFF440F7D266F0D2C65FCA0851C30F9F80F7D261F0D2C55FC5079F6151C70F5FC0010040FFF440F7D260F1D2C5F160D29F61
                          NUMR   4870002028089 02F46801C7F800712000202801902F408007F2001800C7180045F01044920002123 0FA02E40310F2E6000
                               201...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   2 K - 9 K D  0A        Q2 K - | K < D    $ D & K   K       K  OK K   O    * 1K - D:   D 1 0          499
                          ZONE   F7D260F1D2C95FCB0A446DF7D26042D242C81454C954D35ED394E05ED35FD3D0E0F051C50FD060C751CC0F5FC0010040FFF4
                          NUMR   21280F912E4380107F105821280FF028C04CA2B040002C802C200008 02C802C200100300C71210F4A80047180045F01044990
                               301...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR      K  DA            0A K   0  0    0 & J4   J4 0A     1-     K          1  A N             1K     2 K -
                          ZONE   4051D05FCB921101144FC8D014F05F114EF052DF57DF5FC80F9F609F70D0707051CF0F52C8D06A700751C60FD06A70F7D260
                          NUMR   00802C801410007F811014220101800050080014801480187F21082F012D020180 0C71801C5E580172800371 2E5801202800
                               401...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   9 K D& 0   "                                                                                       - 9
                          ZONE   F0D2C55FC607000002000300030003000000000001000100020030040105010100020004000400010002000300030003006 0F0
                          NUMR   902F4080047F00060030 7040C010E0107000808010203040903040 50A05070300030001020104020404010002000303 0190
                               501...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   K D& OAH " O-        1   < 1 0            512   42 K    9 K D   0A       % 12 K    9 K D& D % "2 K -   K D&3 -
                          ZONE   D2C55FCC079F6051C40F51C40F5FC001004 0FFF44FF7D270F0D2C75FCCC851C60FF7D271F1D2C55FC607F7D266F1D2C5F165
                          NUMR   2F4080187F2001800071800C71800 45F0104512004202800902F49801C7F800C71212801902E408003 7F2128 09A02E403109
                               601...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   K  O-      Q2 K    | K < D    $ D & K    K K     <      2 K      O    L & K   A & L     1  K & L    - 1
                          ZONE   D29F66446DF7D27142D242C81454C954D35ED30DE07100807 09F80D081719F8150D250D550C050D251C00F50D650D251C60F
                          NUMR   2E600A10532128 01F028C04CA2B040002C802C21000 12E01012200 2E0003600E803400288010003480 0C7180280034800071
                               701...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   O            523  O2 K    9 K D& OAM  2 K    9 K D   OAM          12 K -    K D&3 - K   O- 2 K    9 K D  DAQ "
                          ZONE   5FC0010040FFF44FF7D270F0D2C55FCD08F7D270F0D2C75FCD0351C70FF7D266F1D2C5F166D29F66F7D271F1D2C75FCD0744
                          NUMR   80045F010452300020280090 2F4080147F202800902F4980147F800471212809A02E4031092E600A212801902E4980187F10
                               801...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR    Q2 K    | K < D    $ D & K    K K      2 K   3    K       K          O        - 12 K    9 K D  0A  "   Q2 K   | K
                          ZONE   6DF7D27142D242C81454C954D35ED3D0E071F7D271F331D29081D06B719F6B51C60FF7D271F1D2C95FCD07446DF7D27142D2
                          NUMR   58212801F023C04CA2B04000 2C802C21000126280383 0A2C6F0D265803605E8000712128 01912E45801C7F105821 2801F028
                               901...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   < D    $ D & K    K K
                          ZONE   42C81454C954D35ED3D0E071
                          NUMR   C04CA2B04000 2C802C210001
                               1001...5...10...15...20...2
```

```
* * * * DEVICE 281 SYS019, MODE IS 9 TRACK 1600 BPI      BLOCK   2   DATA  1024                  * * * *

REC     2   DATA  1024   CHAR
                         ZONE  0000000000000000000044444444444400000000044444444444444444444444444444444444444440000000000000
                         NUMR  000000F000F0000F0F0F000000000000000000F000F0000000000000000000000000000000000000000000F000F0F0F00
                               1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR                        2S 4      02@  . 24 02      "           2M      02M          Z      J  0              3
                         ZONE  000000000000404400047FE0F48004FF71949FF4FF902978090114 1FD93804FFD000119263E0011BD44F75E20902048F45920
                         NUMR  F0000FFF000F01140270227418087 02CA8B024702 6004F041405 70244F04702486D022991 9E5EA1150048 00C110470 3A8004
                            101...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR     1  05             3    3   00 K   ,           3M  W            &3:   4    3K        3   04 N           2  . . / %
                         ZONE  44F04FF19F20982048F444F14FF8D01617901148FD1E5D214A1545F744F048FD4E144AF64FF0D0D0E044F784046616109232
                         NUMR  501070504E041004703A103E7002240B0213057034A68000800240 3A704A7032A00A6032704A50000050240B68411C332004
                            201...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR           0            3      :   4 K         03                      &            W  W&   &               3 K        S  1
                         ZONE  9F1744F75E209F1748FB911748F0D016174FF99212901150254C005C254E001E1E5E25101A592147FFD024161E92FF40045E
                         NUMR  2F025004800C5F027038500A704A27080270325008510E80041C0100041E0AA6B600008E8E80045034270808828D1C4E0680
                            301...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR  2                    1   7        00U   0                 2   /  ,J# % / ,        2*N          4&N           4&          00U
                         ZONE  F20F981208431444FD0F91114FFE44F39C12911244F77505D75576061044F5D0141247F5D0141248F591114FFE5D21902044
                         NUMR  2C7E100877300A501077600E700450004D0860035024410B133C41481C502C54020C704052070170406 00E7004800061 0450
                            401...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR  0             4                                         :                                          < &
                         ZONE  F71D902048F80000000001010101010200000101010101050708090908000000010101010 10D0001020304040405050505050 5
                         NUMR  0480110470460000222A1206080A0C0020281004060 85A4A3A2A12060A00222A1206080A0C0A30303030201 81C100406030A
                            501...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR    * ; -                                                                           F                     J# % / .
                         ZONE  0505060708080806000000010101010101080000000101010101 0A0000000C00000101010 10C00000000010101010746660410
                         NUMR  0C0E3030201 80C1A00222A1206080A0C1A00222A1206180A0C1A000214180C0E10040608160002242C14080A0C1B1C414332
                            601...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR                3          03          ?          ?   2    . . % / .             3          03                0U          0 K                 &
                         ZONE  9020901148F196324FF11444160444164 4F7840446560410901148F196324FF190129D1144FE901144F3D016245A21592547
                         NUMR  61041805703A2908703AB4300F60200F50240B631C41433218057 03A2900703A620B100E700440 0E5000270808004800050
                            701...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR  3    2    . . C . .                      5       +                 0            5             1   04            00U            0U   5   00  $$BERRTN 0
                         ZONE  FF44F784048C8404139F20921241FB441444314433 44F7902043FD592044F04FF19F204FFE982048FE44FB4FF855CCDDED4F
                         NUMR  3A502408633308 68334E041008705 0800E000E00065004110470 50800C501070424E04700410047004105470028825993570
                            801...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR  0   00      0 P            1    2                                0     2    3    2   -            0< -             1    1           0           2
                         ZONE  F34FF343F2D0151543F992F89011010090114 8F292F80F92F355109F1148F44614901241F944F2941248F69B1243F5540004
                         NUMR  087008500471020250148620140 07EA9560070020620730620800 85F0E700CA00A130B70147018100C70084F0C502586 0060
                            901...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR            0   -   K           W
                         ZONE  981248FA9613D013149E4048
                         NUMR  100A70006004200001560070
                           1001...5...10...15...20...2
```

```
* * * * DEVICE  281  SYS019,  MODE IS  9 TRACK  1600 BPI       BLOCK    3   DATA  1024                                   * * * *

REC     3   DATA   1024   CHAR  0     1 7 6 6 5 8 2 4 3 9         0 0 1 3 2 2 8 9    5 6 4 7        . . ( ( + + | | & & $ $ * * ) ) ; ; -
                          ZONE  0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F000F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0404040404040404040405050505050505050505050506
                          NUMR  001F1F01071516050302040309000D001010010312120809000506040 71A1A1B1B1D1D1E1E1F1F1010181B1C1C1D1D1E1E10
                                1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR  - / /    . , , % % _ : # @ ' = " A B C D E F G H I J K L M N O P Q R S T U V W X Y Z   0     1 4 2 2
                          ZONE  06060606060606060505050507070707070707070C0C0C0C0C0C0C0C0C0D0D0D0D0D0D0D0D0D0E0E0E0E0E0E0E0E000F0F0F0F0F0F0F
                          NUMR  1011111A1A1B1B1C1C0D0A0B0C000E0F0102030405060703090102030405060708090203040506070809000001F1F01041212
                            101...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   3 8 5 6 7 9                    A O E R I I S N C M L D O O T U P F 1 V B B G 2 J 5 H 4 3 6 . 7 8 9 Z . Q
                          ZONE  0F0F0F0F0F0F0F0F0F0F0004040404040C0D0C0D0C0C0E0D0C0D0D0C0F0F0E0E0C0C0F0E0C0C0C0F0D0F0C0F0F0F0F040F0F0F0E0600
                          NUMR  03080506070900C0D0002020202001060509191902050304030410100304070601051212070201050804030608070809090303
                            201...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   X / Y W K & * , , , , ' ' $ % # #       ( + | ) ;    _ : @ = "   0       1 2 3 3 4 5 6 8 7 9        0
                          ZONE  0E060E0E0305050606060607070506070704040404050505050707070700F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F000F0F0F
                          NUMR  0701080502000C2B232B2B1D1D0B0C1B1B1A1A0D0E0F0D0E0A0D0A0C0E0F00001F1F0102131304050608070900D00001F1F
                            301...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   1 2 4 4 6 5 7 3 8 9        0   2 1 7 8 5 4 6 3     9       0         2 4 1 1 6 5 8 3 9 7      1       0
                          ZONE  0F0F0F0F0F0F0F0F0F0F0F0F0F000F0F0F0F0F0F0F0F0F0F0F0F000F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F041004EFB1748
                          NUMR  01021414060507030809000D000000F0201070805040603000900000000F1C1C0204111106050803090700012110A500C2770
                            401...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR  3Y  -          03Y       03Y 03    J    0      3      3   1  03      03          -   3   1     3K 03    J   0   &
                          ZONE  FE94604600074FFE41004FFE4FF7900D04EF3154C=B1747FB4EF34FF9184FFB1808946043F04EF91548FD4FF990D04F005900
                          NUMR  38200016016070381100A703870326215500977038277032501870 3E87703A856020004 03A501425703270 3E41157 0020802
                            501...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   Y  &          H      ¬                                                                         B
                          ZONE  4E005300510040057005600880080008800880091003100800084008C008A00800088008200 8A0088008F0084008C008400
                          NUMR  4802000288020802280 2F8002E003A003A003E00420056006400 6A006A007200 8600 84009600900 0A400AE00860082 00CA00
                            601...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR          -          6          @                                                     4        *        -
                          ZONE  8B0086008300 9F009A009100 97009200 96009A004400 4700AA00A700A2CCA000A400A800AC00AF00A200A500A800A600AE00
                          NUMR  C200D000F4009600C400D600EC00F600F500F00002000000C00120020004600440 06C00600064007800 7007A0090009E00
                            701...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR          O          B                 :           &                            D               :
                          ZONE  A900AF00A400A500AC00A100A200AE00A000A400A700AD00A500BD00B100BF00B100BD00BE00B200 8C0087008F0085008700
                          NUMR  A400A000C400CA00C200D200D800D000EE00E400EA00EE00F000 0E001A001C002200 2C002C00300014003800 3E0045004A00
                            801...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR          B   W                                        H                                    +
                          ZONE  8900BC008E008000830 0BF0084008B008A008C003000320083008C008F0083008500870088008F008E00810084008900 8800
                          NUMR  4E0042004600 5A005E005C0064006A006200600078007000 74007800 7A0088008A00 860086008C008E009300 9E00940 09600
                            901...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                          CHAR   U     <              2     U
                          ZONE  8E00840084008A008F008E00
                          NUMR  9400AC00A400AC00A200A400
                           1001...5...10...15...20...2
```

```
* * * * DEVICE  281  SYS019,  MODE IS  9 TRACK  1600 BPI        BLOCK    4    DATA  1024                                          * * * *

REC     4   DATA  1024   CHAR            0   Y                         Q   8                                    -
                         ZONE  0000A000E000F000E000D000C0003000D000F000F000F0008000F000C000F000F200F400F500F800FA00FC00FE00F000F200
                         NUMR  0001000400050005800500050004000480058006C007E0080003F0080004C00BC00BC00BC00BC00BC00BC00BC00BD00BD00B
                               1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR       .:                 0                          &  -                               0
                         ZONE  F400F600F800FA00FC00FE00FF00F000F100F200F300F400F500F600F700F800F900FA00FB00FC00FD00FE00FF00F000F100
                         NUMR  D00BD00BD00BD00BD00BD00CD00CD00CE00CE00CE00CE00CE00CE00CE00CE00CE00CE00CE00CE00CE00CE00CE00CF00CF00C
                            101...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR              &  -                                               -  0         8          0
                         ZONE  F200F300F400F500F600F700F800F900FA00FB00FC00FD00FE000000400080006000F000D000E000F000A000B000F000000
                         NUMR  F00CF00CF00CF00CF00CF00CF00CF00CF00CF00CF00CF00C000200030003000300050004000430006000400004F00C0001
                            201...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR                    Y  0  Q   8                      4                 0      8
                         ZONE  A000C000D000B000E000E000F000D000F000F000F0008000F0000000FF41F000FF81FFA1FFC1FF00FF81FF00FFE1FF01F000
                         NUMR  000400040005000400058005000580058005C007E0030003F0080002FE024007FE03FE03FE03F000FD02F80EFE03FF030008
                            301...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR                         -                                 -   M  U          Y
                         ZONE  F000FF21FF01FFC1FF41FF61FF01FF81FF01FFA1FFC14000E000A0008000 6000D000E000E0008000E000FE00B000A0009000
                         NUMR  F009FF03FC00FD02FF03FF03FE02FF03FD01FF03FF0300040006000580050004400640 07C0070004800 7F00C0005800 5300 5
                            401...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR  &                H                                 Q   W   0         2   6   8
                         ZONE  5000D000F0007000900 0C000C000D000F C00F800F800F000C000D000E0CCF000E000E000F000F000F000F000FFE10000A000
                         NUMR  00040006E0090004000580 06C006C006F008E009F00AC008000580066007 0007E007A0072007600780 07A007FF0300010004
                            501...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR                  Q      0  Y  8                                      Y      Q         0   8
                         ZONE  B000C000D000D000E000F000E000F000F000F0008000F0000000A000B000E000C000D000D000E000F000F000F000F0008000
                         NUMR  0004000430035803 05000500058005800 6C007E0080003F0080001000400048005000480050005000500058006C007E0080003
                            601...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR                   8   0                   Y                                    4   Q      Y     0   8
                         ZONE  F0000000D000C000F000F000F000F000E000E000F000F000F000F3008000FC000000E000D000F000D000E000E000F000F000F000
                         NUMR  F0080001000400040008 8C060005C00800058005F009E008F00A0002F00A0001000500054006800 5C0068006C00700068006
                            701...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR                    00   0S 00    3    0    3    3 &&3    3    3 & 2Y   2          1     3 & 2Y        1
                         ZONE  C000F0008000F0004FF348FE4FF344F14DFE44F144F055F090F041F051FE41FE0098104 1F10040F150FE0098104 1F2009010
                         NUMR  0004E0080002F008700010027000903A700E803A003E00382938103000281020A010027012A710300028A010027016A71104
                            801...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR    1  00          0    00    3    1<  3    2 &&28   28   20& 2Y   2          1     28& 2Y   2        1
                         ZONE  41F14FFC15164DFA154FFA44F14DF444F144FF55FF90FF41FF51FE41FE0098104 1F70040FF50FE41FE0098104 1F800901041
                         NUMR  701A700AA694700284700290387 01C803800 2E00282A28102C00281020A01002701 0A7102300281020A010027018A7110470
                            901...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                         CHAR   1a  1  00        &
                         ZONE  F74EFB4FFC14159450547048
                         NUMR  1C501C700AB6A620009 00470
                           1001...5...10...15...20...2
```

```
* * * * DEVICE  281  SYS019,  MODE IS  9 TRACK  1600 BPI        BLOCK    5   DATA  1024                                  * * * *

REC      5   DATA  1024    CHAR  1        2K         00      ' 1F     & 1
                           ZONE  FB040444FD4400151 44FFA130343FCJFD050FC0000000000000000011111111111111111222222222222222223333333333 33333
                           NUMR  14606040221402B6A67002846040167EC0001C0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789A3C0
                                 1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                           CHAR                 .<(+|&          $*);¬-/              ,%_>?          :#@'="  ABCDEFGHI        JKLMNOPQR
                           ZONE  3344444444444444445555555555555555666665556565666667777777777777778CCCCCCCCC888889DDDDDDDDD999999AA
                           NUMR  EF0123456789ABCDEF0123456789ABCDEF0123456789A3CDEF0123456789ABCDEF0123456789ABCDEF0123455739ABCDEF01
                             101...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                           CHAR  STUVWXYZ                            ABCDEFGHI        JKLMNOPQR           STUVWXYZ       0123456789      K &
                           ZONE  EEEEEEEEAAAAAABBBBBBBBBBBBBBBBCCCCCCCCCCCCCCCCDDDDDDDDDDDDDDDEEEEEEEEEEEEEEEEFFFFFFFFFFFFFFFFD05010
                           NUMR  23456789A3CDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789A3CDEF0123456789ABCDEF200000
                             201...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                           CHAR  K & &       RARE                       G                                        C110A STOP C111A AWAITING
                           ZONE  D0505000 5BDCDC000000000005900000 05C200100592000005BA000005A2000005B30000F07CFFFC4EEDD4CFFFC4CECCECDC4
                           NUMR  200100024091950000000402300000923700 04A23800009230 008923800 00A23000010F08311010236703111101619395 70
                             301...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                           CHAR  REPLYENTE      30090PARAMBU           30100PIRANGA           30110TAMBORIL          30111
                           ZONE  DCDDECDEC44444FFFFFDCDCDCE4444444444FFFFFDDDCDCC4444444444FFFFFECDCDDCD444444444FFFFF0000000080000000
                           NUMR  957385535000003009071914240000000000301007591571 0000000000301103142699300000000003011107070004080 0102
                             401...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                           CHAR             (    IJSYSIN               3                                            &             %
                           ZONE  580000004C30CDEEECD4000000000000022F805D800000000000000F0005600001001000000054000030564000305640000 05920
                           NUMR  480000020022912829500000000030000302380000000000000 0F0021A0000A109000010007000724A0006124C0005824000
                             501...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                           CHAR            &$$8OPEN                          00 I_300S?0220012/13/74K J-5    J   5         0      0    5 K J
                           ZONE  000050200555CDDC040000000000100044444FF1CDCCCEDFFFFFFF6FF6FFD0D6F19ED05BF194104EF3940448F69FF0000 51090
                           NUMR  01623B0000BB26755000000001800000007008932442702200121113174231 0500C1880541002700610087002 6F5F2118 0061
                             601...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                           CHAR       _ '&-48   - K - -     1U      00     J             24   J          0                         0      &          1Q
                           ZONE  BB460755FF9460D7606048FE40004FF690D09810410041FF9000941047FB1E4E001E4E009010 41F94E005E0194D448FD1545
                           NUMR  09100D0048200027010050147004700240141000110270244015500070 0CBE3101A11E03110170061E0100041008701 8B531
                             701...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                           CHAR            &1    J  01    J          OK   OU 05       0   05          J     .                                      J
                           ZONE  004400170745FA90D04FF39ED6921048FD4EFE5FF20F4EFD5FF20F9ED6010400000000000000000000BB4500450090D00115
                           NUMR  011102B56040186114701 80F14100070021004805 87F10080587F8F14000B0000000 000000000000006A10235 0012157E97
                             801...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                           CHAR    1          1   J    1        1  01   &5    1    5    1         1U       01;&   *   J  05       J      *  _  '  - K - -
                           ZONE  4CF8901041F2900541F21748F34FF355F042F19FF041F191DF43FE40004FF551059ED65FF20E9ED6510546079460D7606057
                           NUMR  70161F00701311 1A701827701870109058701E1F5F701E0205014700 2701E0D1C0F1480505F8F148D1C1D0020 0027010080
                             901...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80...85...90...95.....

                           CHAR  5    J    1Q        OI      01
                           ZONE  F090D041FD91DF4FF0184FF8
                           NUMR  5811147018820C7012877018
                            1001...5...10...15...20...2
```

4. APLICAÇÃO DO PROGRAMA DE CONVERSÃO (COMP050)

Por ser este um programa cuja finalidade exclusiva é a conversão, ou seja, compressão ou descompressão de arquivos do usuário, não vimos necessidade de descrever sua execução, uma vez que, além de ser muito simples, já foi convenientemente exposta em capítulos anteriores.

5. COMPRESSÃO FINAL

Finalmente, podemos, através do quadro que se segue, demonstrar, de uma forma geral, os resultados obtidos da aplicação, ao Arquivo de Débitos dos Consumidores, dos Métodos de Compressão por Segmentação e do Código de Tamanho Variável HSF.

DEMONSTRATIVO DE COMPRESSÃO

METODO	IDENT.	SEGM.	CARACT.	NAO PER.	CARACTERES ANALISADOS (CARACTERES)	CUSTO MÉDIO CÓDIGO (BITS/CARATER)	COMPRESSÃO ABSOLUTA (BITS)	COMPRESSÃO RELATIVA
H	1		1		3 200 013	1.0809	3 456 824	0.048 225
	2		1		1 040 059	3.2446	4 944 931	0.068 986
S	3		0		1 319 540	1.6178	2 134 015	0.029 771
	4		23		3 401 193	4.1201	13 187 550	0.183 978
F	5		0		475 534	1.7918	851 758	0.011 882
	6		0		208 478	1.2234	254 899	0.003 556
	7		0		160 718	1.6036	257 619	0.003 594
	8		0		162 890	2.4413	397 547	0.005 546
TOTAL					9 968 425		25 485 143	0.355 538
SEGMENTAÇÃO	1				1 600 000	0	0	0
	2				1 040 000	0	0	0
	3				720 000	0	481 896	0.006 722
	4				3 520 000	0	950 752	0.013 263
	5				320 000	0	657 920	0.009 178
	6				320 000	0	1 726 144	0.024 081
	7				320 000	0	1 917 184	0.026 746
	8				1 120 000	0	8 308 496	0.115 910
TOTAL					8 960 000	0	14 042 392	0.195 900
TOTAL GERAL							39 527 535	0.551 438

Antes de apresentarmos o resultado final, devemos esclarecer 3 pontos:

1. Por serem os segmentos compressos de Tamanho Variável, devemos quando da sua gravação, anexar 1 byte contendo o seu comprimento (TS), conforme ficou estabelecido quando projetamos o sistema. Desta forma o nosso arquivo compresso ficou aumentado de 422.009 bytes que é o número de segmentos com conteúdo.

2. Para 1 ou mais segmentos adjacentes sem conteúdo, gravamos no arquivo compresso um Byte Sentinela (BS). Por não termos condições de determinar, de quantos bytes foi acrescido o arquivo compresso pela inclusão do Byte Sentinela, não levaremos em consideração este valor, para computo da compressão final.

3. Visto trabalharmos a nível de byte, teremos de anexar, ao final de todo segmento compresso, pelo código HSF, tantos bits quantos sejam necessários para completar um número inteiro de bytes. Considerando que temos 422.009 segmentos compressos pelo Código HSF com uma média de 4 bits anexados por segmento, aumentaremos o nosso arquivo compresso de 211.004 bytes.

Esclarecidos os pontos acima utilizando-se o quadro demonstrativo de Compressão, passamos à obtenção da compressão final, que se resume nos dados abaixo:

a) Tamanho da amostra

- Número de registros - (80000)
  *
- Comprimento do registro em bytes - (112)
  =

Total - 8.960.000 bytes
ou 71.680.000 bits.

b) Compressões Totais na amostra sem considerar os acréscimos advindos da complementação do byte e do byte de Tamanho do Segmento (TS):

- Compressão Absoluta - 39.527.535 bits ou 4.940.941 bytes.

- Compressão Relativa - 0,551438.
- Compressão Percentual - 55,14%.
- Número médio de bits por caráter - 3,5884 bits/caráter.

c) Compressões totais na amostra considerando os acréscimos advindos da complementação do byte e do byte de Tamanho do Segmento (TS):

- Compressão Absoluta - 39.527.535 - 5.064.108 = 34.463.427 bits ou 4.307.928 bytes.
- Compressão Relativa - 0,480795.
- Compressão Percentual - 48,07%.
- Número médio de bits por carater - 4,1536 bits/caráter.

6. DADOS COMPLEMENTARES

Em complementação aos dados de compressão anteriormente fornecidos, temos a acrescentar os seguintes pontos:

1. Memória - A subrotina de compressão/descompressão necessita para sua execução de 12.568 bytes de memória.
2. Tempo de execução - Para estes testes, utilizamos um computador /370 modelo 145 e uma amostra do arquivo do usuário de 10.000 registros.

Antes porém, de apresentarmos os resultados observados, devemos chamar a atenção para o fato de que para estes testes foram desenvolvidos 4 programas em cobol, com as seguintes finalidades:

- Leitura do arquivo compresso, para obtenção do tempo de descompressão.
- Leitura e gravação do arquivo compresso, para obtenção do tempo de descompressão e compressão.
- Leitura do arquivo descompresso, para obtenção do tempo normal de leitura.

- Leitura e gravação do arquivo descompresso para obtenção do tempo normal de leitura e gravação.

Com relação aos tempos de compressão e gravação esclarecemos que os mesmos foram obtidos dos tempos acima citados.

Finalmente devemos observar que nos resultados apresentados estão incluídos os seguintes tempos:

- Abertura e fechamento dos arquivos.
- Verificação de labels.
- Transferência para ou das unidades de entrada e saída.

Assim, uma vez esclarecidos os pontos acima, foram os seguintes os resultados obtidos para a amostra de 10000 registros:

- Leitura do arquivo compresso (tempo de descompressão) - 96 seg.
- Leitura e gravação do arquivo compresso (tempo de descompressão/compressão) - 152 seg.
- Gravação do arquivo compresso (tempo de compressão) - 56 seg.

- Leitura do arquivo normal - 16 seg.
- Leitura e gravação do arquivo normal - 30 seg.
- Gravação do arquivo normal - 14 seg.

Dos dados acima, podemos tirar as seguintes conclusões:

a) Considerando-se os tempos de forma absoluta, ou seja, com os tempos de abertura, fechamento, verificação de labels, etc, temos:

REGISTROS	TEMPOS		
	COMPRESSÃO	DESCOMPRESSÃO	COMP / DESCOMP.
1	5,6 μSeg	9.6 μSeg	15.2 μSeg
10000	5.6 Seg	96 Seg	152 Seg
80000	446 Seg (7.46 min)	768 Seg (12.8 min)	1216 Seg (20.26 min)

VI - 31

b) Considerando-se os tempos de forma relativa, ou seja, tomando-se os tempos de leitura e gravação do arquivo normal, como sendo aqueles correspondentes a abertura, fechamento, verificação de labels, etc, quando da compressão e/ou descompressão do arquivo, temos:

- Leitura do arquivo compresso
  (tempo de descompressão) - 96 - 16 = 80 seg.
- Leitura e gravação do arquivo compresso
  (tempo de descompressão/compressão) - 152-30=122 seg.
- Gravação do arquivo compresso
  (tempo de compressão) - 56 - 14 = 42 seg.

REGISTROS	TEMPOS		
	COMPRESSÃO	DESCOMPRESSÃO	COMP. / DESCOMP.
1	4.2 μSeg	8 μSeg	12.2 μSeg.
10000	42 Seg	80 Seg	122 Seg
80000	336 Seg (5.6 min)	640 Seg (10.6 min)	976 Seg (16.26 min)

VI - 32

7. CONCLUSÕES

Pelos resultados expostos, concluimos, ser o sistema apresentado, perfeitamente viável para compressão de dados.

Considerando que o arquivo escolhido possui uma alta densidade de informação (79,41% com conteúdo) e que 38,35% dos caracteres a serem compressos pelo código HSF estão registrados no formato Decimal Compactado (4 bits por carater), conseguimos atingir os objetivos do sistema, que são:

- Comprimir/descomprimir informações de modo eficiente independente de sua estrutura e/ou conteúdo.
- Minimizar as alterações nos programas do usuário quando da implantação do sistema.
- Maximizar a transparência do sistema aos programadores da instalação.

CAPÍTULO VII

SUGESTÕES

1. APRESENTAÇÃO

Encerramos nosso trabalho apresentando algumas sugestões que permitam a generalização do uso, otimização da eficiência e facilidade da implantação de futuros sistemas de compressão.

Estas sugestões, reunidas no decorrer de todo o trabalho, estão apresentadas a seguir.

2. SUGESTÕES

2.1. PADRONIZAÇÃO DE INFORMAÇÃO

Já se encontram definidas para o atual sistema, 2 programas (COMPØ1Ø e COMPØ2Ø) cujas finalidades são a padronização do conteúdo dos campos. Assim, campos como endereço, nome de pessoas, etc, possuem características de terem alguns dos seus componentes repetidos diversas vezes por todo arquivo. Normalmente deveria se esperar, que o valor atribuído a um dado componente que se repetisse fosse o mesmo para todo o arquivo. Observa-se porém que o que se encontra são valores iguais registrados de formas diferentes.

Assim, o que se pretende aqui, através destes 2 programas, é que valores iguais sejam registrados do mesmo modo.

Com esta padronização conseguimos aumentar a eficiência do sistema, pelo aumento da taxa de compressão.

2.2. COMPATIBILIDADE COM OUTRAS LINGUAGENS DE PROGRAMAÇÃO

Como já vimos, o nosso sistema, através das suas rotinas de compressão/descompressão, só pode ser implantado em programas codificados utilizando a Linguagem de Programação COBOL.

Esta restrição pode facilmente ser contornada, desde que seja desenvolvida na subrotina para cada linguagem de programação que se deseje compatibilidade do sistema, um módulo de ligação com o seu respectivo entry-point.

2.3. COMPRESSÃO PARCIAL DO REGISTRO

Um dos fatos da perda de performance do nosso sistema está na compressão/descompressão integral do registro.

Do exposto, um sistema poderia ser desenvolvido,em que seriam descompressos e compressos apenas os campos do registro desejados pelo programa do usuário.

2.4. REGISTROS DO USUÁRIO

O sistema por nós desenvolvido, trabalha com arquivos que possuem apenas 1 tipo de registro.

Uma forma de generalização do sistema se resume em fazê-lo aceitar arquivos com mais de 1 tipo de registro.

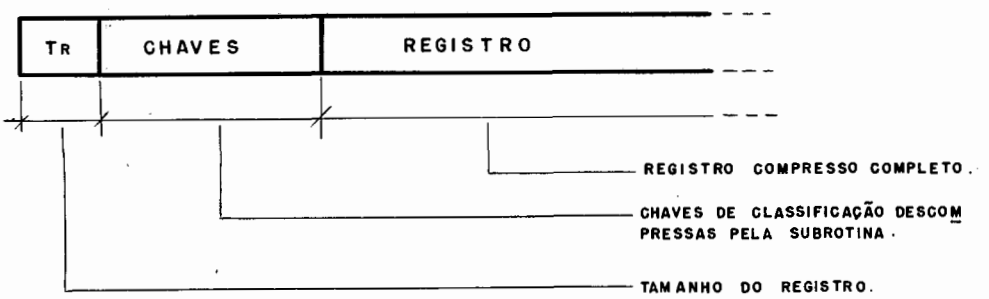
2.5. PROGRAMAS UTILITÁRIOS

Afim de melhorar a performance dos programas utilitários (COMPØ3Ø e COMPØ4Ø), desenvolver as rotinas de determinação de ausência ou presença de conteúdo nos segmentos (byte a byte) à semelhança da rotina implantada na subrotina de compressão/descompressão com Padrão de Ausência (segmento a segmento).

2.6. CLASSIFICAÇÃO DOS ARQUIVOS COMPRESSOS

Através do EXIT 15 e 35 adaptar subrotinas ao SORT. Estas teriam por função entregar e/ou receber do SORT os registros do arquivo do usuário na forma compressa.

Para isto, estas subrotinas fariam apenas a descompressão e/ou compressão das chaves de classificação, entregando e/ou recebendo do SORT o registro de tamanho variável no formato a seguir:



VII - 1

Observamos que desta maneira além de economizarmos áreas nos arquivos do usuário, também economizaríamos área de trabalho e talvez tempo de execução.

2.7. COMPRESSÃO UTILIZANDO MICRO PROGRAMAS

Uma sugestão bastante interessante, para quem possa interessar, será aproveitando o caráter geral de aplicação das rotinas de compressão/descompressão, implantá-las utilizando micro códigos.

APÊNDICES

APÊNDICE A

LISTAGEM FONTE

SUBROTINA DE LEITURA

```
ID DIVISION.
PROGRAM-ID.
    'LEFITA'.
AUTHOR.
    MAURILIO G FILHO.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT ENTRADA1 ASSIGN TO SYS010-UT-2400-S.
DATA DIVISION.
FILE SECTION.
FD  ENTRADA1
    RECORDING MODE IS U
    LABEL RECORD IS STANDARD
    RECORD CONTAINS 1 TO 12000 CHARACTERS
    DATA RECORD IS ENTRADA1A.
01  ENTRADA1A.
    05  REG OCCURS 60 TIMES DEPENDING ON FB.
        10  ITEM          PIC X OCCURS 200 TIMES DEPENDING ON
            RL.
WORKING-STORAGE SECTION.
01  TRABALHO.
    05  VEZ           PIC 9          VALUE ZEROS.
        88 VEZ1       VALUE 1.
    05  CONTA         PIC 99   COMP  VALUE ZEROS.
    05  CONT  PIC 999 COMP-3.
    05  DADOS.
        10  FB  PIC 99.
        10  FILLER PIC X.
        10  RL  PIC 999.
LINKAGE SECTION.
77      FIM-LEIT       PIC X.
        88  FIM        VALUE '*'.
77  REGISTRO PIC X(200).
 PROCEDURE DIVISION USING                REGISTRO FIM-LEIT.
    PERFORM PROC INICIAL UNTIL VEZ1
    PERFORM LER-ARQ UNTIL CONTA NOT = FB.
    PERFORM PROC BLOCO.
    GOBACK.
    STOP RUN.
-------------------------------------------------------------
PROC INICIAL. OPEN INPUT ENTRADA1          MÓDULO DE "INICIALIZAÇÃO"
    DISPLAY 'DE FATOR DE BLOCO-TAMANHO REGISTRO   XX-XXX'
    UPON CONSOLE.
    ACCEPT DADOS FROM CONSOLE
    MOVE FB TO CONTA
    MOVE 1 TO VEZ.
-------------------------------------------------------------
LER-ARQ.                                   MÓDULO DE "LEITURA"
    READ ENTRADA1 AT END
      CLOSE ENTRADA1.
    MOVE ZERO TO CONTA.
-------------------------------------------------------------
PROC BLOCO.                          MODULO "PROCESSA BLOCO"
    ADD 1 TO CONTA
```

```
IF ITEM (CONTA,  1) = '*'
  MOVE '*' TO FIM-LEIT
ELSE
MOVE REG (CONTA) TO REGISTRO
  MOVE '*' TO ITEM (CONTA,  1).
```

APÊNDICE B

LISTAGEM FONTE

SUBROTINA DE IMPRESSÃO

```
IDENTIFICATION DIVISION.
PROGRAM-ID.
    'ROTPRES'.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
INPUT-OUTPUT SECTION.
FILE CONTROL.
    SELECT PRINTER ASSIGN TO SYS006-UR-1403-S.
DATA DIVISION.
FILE SECTION.
FD  PRINTER   RECORDING MODE IS F
              RECORD CONTAINS 133 CHARACTERS
              LABEL RECORD IS OMITTED
                DATA RECORD IS ROT-LINHA.
01  ROT-LINHA      PICTURE X(133).
WORKING-STORAGE SECTION.
77      ROT CHAVE2     PICTURE X VALUE SPACE.
LINKAGE SECTION.
01  ROTLINE.
    05  ROT CONTROL PICTURE X.
    05  ROT-RESTO  PICTURE X(132).
01  ROT-CHAVE1      PICTURE X.
PROCEDURE DIVISION.
    ENTER LINKAGE.
    ENTRY 'RPRINT'  USING ROTLINE ROT-CHAVE1.
    ENTER COBOL.
RPRINT1.
    IF ROT CHAVE2 IS EQUAL TO '1'
       THEN GO TO  RPRINT2
       ELSE OPEN OUTPUT  PRINTER                    MÓDULO DE "INICIALIZAÇÃO"
            MOVE '1' TO ROT CHAVE2.
RPRINT2.                                            MÓDULO DE "IMPRESSÃO"
    IF ROT-CHAVE1 IS EQUAL TO 'T'
       THEN CLOSE PRINTER
    ELSE WRITE ROT-LINHA FROM ROTLINE
        AFTER POSITIONING ROT-CONTROL LINES
          MOVE SPACES TO ROTLINE.
FIM. EXIT PROGRAM.
```

APÊNDICE C

LISTAGEM FONTE

PROGRAMA GERADOR DE ESTATÍSTICA PARA SEGMENTAÇÃO (COMPØ3Ø)

```
ID DIVISION.
PROGRAM-ID.
    'COMP030'.
AUTHOR.
    MAURILIO G. FILHO.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
    DECIMAL-POINT IS COMMA.
INPUT-OUTPUT SECTION.
FILE CONTROL.
    SELECT CARTAO ASSIGN TO SYS009-UR-3505-S.
DATA DIVISION.
FILE SECTION.
FD  CARTAO
    LABEL RECORD IS OMITTED
    DATA  RECORD IS CARD.
01  CARD.
    05  IDENTIF        PIC 9(5).
    05  TAMANH         PIC 9(3).
    05  CODIG          PIC 9.
    05  FILLER         PIC X(71).
WORKING-STORAGE SECTION.
01  TRABALHO.
    05  FIM LEIT       PIC X      VALUE ZEROS.
        88  FIM        VALUE '*'.
    05  CONTA          PIC 9(5)  COMP 3.
    05  LIMITE         PIC 9(5).
    05  NCAMP          PIC 9(3)  COMP-3.
    05  PL             PIC 999    VALUE 0      COMP-3.
    05  FB             PIC 9(5)  COMP-3.
    05  CONTAC         PIC 9(3)  VALUE 1      COMP 3.
    05  CODIGO         PIC 9      COMP-3.
    05  CONTAB         PIC 9(3)  COMP-3.
    05  APTB1.
        10 FILLER  PIC X  VALUE ' '.
        10 APTBS  PIC X.
    05 APTB REDEFINES APTB1 PIC 999 COMP.
    05  APTC           PIC 9(3)  COMP 3.
    05  PAG1           PIC 9(4)  VALUE ZEROS COMP-3.
    05  CONTA-LINHA    PIC 99     VALUE 57.
    05  LINHA.
        10  CONTR01    PIC X.
        10  DADOS      PIC X(132).
    05  SOBRA          PIC S999 COMP.
    05  FIM-IMP        PIC X      VALUE ' '.
    05  ACUM1          PIC 9(6)  VALUE ZEROS COMP-3.
    05  PROB           PIC 9V999999 COMP-3.
    05  ACUM2          PIC 9V999999 COMP-3  VALUE ZEROS.
    05  ACUM4          PIC 9(8)  VALUE ZEROS COMP 3.
    05  ACUM5          PIC 9(5)  VALUE ZEROS COMP-3.
    05  ACUM6          PIC 9(8)  VALUE ZEROS COMP-3.
    05  ACUM8  PIC 9(8) VALUE ZEROS COMP-3.
```

```
05  ACUM9          PIC 9V999999 VALUE ZEROS COMP-3.
05  TOT-ARQ  PIC 9(9) COMP-3.
05  COMPARQ1       PIC 9(7)  COMP 3.
05  COMPARQ2       PIC 9(7)  COMP-3.
05  COMPABS1       PIC 9(4)  COMP 3.
05  COMPABS2       PIC 9(7)  COMP-3.
05  COMPREL1       PIC 9V999999 COMP-3.
05  COMPREL2       PIC 9V999999 COMP-3.
05  COMPPER1       PIC 9(3)  COMP-3.
05  COMPPER2       PIC 9(3)  COMP-3.
05  RAIOCOM1       PIC 99V999999 COMP-3.
05  RAIOCOM2       PIC 99V999999 COMP-3.
05  POSINI         PIC 9(4)  VALUE 1   COMP-3.
05  ACUM15         PIC 9(8)  VALUE ZEROS COMP-3.
05  INICIO         PIC 9(3)  VALUE ZEROS COMP-3.
05  AVAIL          PIC 9(3)  VALUE 1     COMP-3.
05  APTA           PIC 9(3)              COMP-3.
05  APTN           PIC 9(3)              COMP 3.
05  DAT1  PIC X(8).
05  CABEC1.
    10  FILLER     PIC X(43) VALUE SPACE.
    10  FILLER     PIC X(48) VALUE
        'M A P A    D E    S E G M E N T A C A O
    10  DAT2       PIC X(8).
    10  FILLER     PIC X(18) VALUE
        '              PAG   '.
    10  PAG        PIC ZZZZ.
    10  FILLER     PIC X(11) VALUE SPACE.
05  CABEC2.
    10  FILLER     PIC X(50) VALUE
        'NUM. POS.   FREQ.    PROBAB.   COMP. PARTIC.   PAR'.
    10  FILLER     PIC X(50) VALUE
        'TIC.   COMPR.    COMPR.     COMP. COMP. REL.
    10  FILLER     PIC X(32) VALUE
        'F O R M A T O
05  CABEC3.
    10  FILLER     PIC X(50) VALUE
        'CMP. INI.    CMP.    CAMPO     ORIG.  TOTAL    COM'.
    10  FILLER     PIC X(50) VALUE
        'PR.    ABSOLUTA  RELATIVA   PERC.  ARQUIVO
    10  FILLER     PIC X(32) VALUE
        '  C A M P O                    '
05  DET1.
    10  CAMPO1     PIC Z(3).
    10  FILLER     PIC XX    VALUE SPACE.
    10  CAMPO2     PIC Z(4).
    10  FILLER     PIC X(2)  VALUE SPACE.
    10  CAMPO3     PIC ZZZZZZ9.
    10  FILLER     PIC X(2)  VALUE SPACE.
    10  CAMPO4     PIC 9,999999.
    10  CAMPO4R REDEFINES CAMPO4 PIC X(8).
    10  FILLER     PIC X(3)  VALUE SPACE.
    10  CAMPO5     PIC Z(4).
```

```
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO6      PIC ZZZZZZZ9.
    10  FILLER      PIC X(2)   VALUE SPACE.
    10  CAMPO7      PIC ZZZZZZZ9.
    10  FILLER      PIC X(2)   VALUE SPACE.
    10  CAMPO8      PIC ZZZZZZZ9.
    10  FILLER      PIC X(3)  VALUE SPACE.
    10  CAMPO9      PIC 9,999999.
    10  CAMPO9R REDEFINES CAMPO9 PIC X(8).
    10  FILLER      PIC X(3)   VALUE SPACE.
    10  CAMPO10     PIC ZZ9.
    10  CAMPO10R REDEFINES CAMPO10 PIC X(3).
    10  FILLER      PIC X(3)   VALUE SPACE.
    10  CAMPO11     PIC Z9,999999.
    10  CAMPO11R REDEFINES CAMPO11 PIC X(9).
    10  FILLER      PIC X(4)   VALUE SPACE.
    10  CAMPO12     PIC X(20).
    10  FILLER      PIC X(15) VALUE SPACE.
05  CABEC4.
    10  FILLER      PIC X(50) VALUE
        'NUM.    FREQ.    PROBAL.   COMP.ORIGINAL   COMP.COMP'.
    10  FILLER      PIC X(50) VALUE
        'RES.    COMPR.ABSOLUTA  COMPR.RELATIVA        COMPR.'.
    10  FILLER      PIC X(32) VALUE
        'PER. RAIO COMP.
05  CABEC5.
    10  FILLER      PIC X(50) VALUE
        'EST.    ESTR.    ESTRUT.    EST.     ARQ.     EST.     A'.
    10  FILLER      PIC X(50) VALUE
        'RQ.     EST.     ARQ.       EST.       ARQ.      EST. A'.
    10  FILLER      PIC X(32) VALUE
        'RQ.
05  DET2.
    10  CAMPO13     PIC Z(3).
    10  FILLER      PIC X(3)   VALUE SPACE.
    10  CAMPO14     PIC Z(6).
    10  FILLER      PIC XX     VALUE SPACE.
    10  CAMPO15     PIC 9,999999.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO16     PIC Z(6).
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO17     PIC Z(8).
    10  FILLER      PIC XX     VALUE SPACE.
    10  CAMPO18     PIC Z(5).
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO19     PIC Z(3).
    10  FILLER      PIC XX     VALUE SPACE.
    10  CAMPO20     PIC ZZZZZ.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO21     PIC ZZZZZZZ9.
    10  FILLER      PIC X(3)   VALUE SPACE.
    10  CAMPO22     PIC 9,999999 BLANK WHEN ZERO.
    10  FILLER      PIC XX     VALUE SPACE.
```

```
    10  CAMPO23    PIC 9,999999 BLANK WHEN ZERO.
    10  FILLER     PIC XXX    VALUE SPACE.
    10  CAMPO24    PIC ZZ9 BLANK WHEN ZERO.
    10  FILLER     PIC XX     VALUE SPACE.
    10  CAMPO25    PIC ZZ9 BLANK WHEN ZERO.
    10  FILLER     PIC XXX    VALUE SPACE.
    10  CAMPO26    PIC Z9,999999 BLANK WHEN ZERO.
    10  FILLER     PIC X(18) VALUE SPACE.
05  CABEC6.
    10  FILLER     PIC X(50) VALUE
        'NUM.                  E  S  T  R  U  T  U  R  A
    10  FILLER     PIC X(37) VALUE
        '    D  O  S           C  A  M  P  O  S
    10  FILLER     PIC X(45) VALUE SPACE.
05  CABEC7.
    10  FILLER     PIC X(50) VALUE
        'EST. 0         1          2          3          4
    10  FILLER     PIC X(50) VALUE
        '    5          6          7          8          9
    10  FILLER     PIC X(32) VALUE
        '    0          1          2
05  CABEC8.
    10  FILLER     PIC X(50) VALUE
        '     12345678901234567890123456789012345678901234567890123'.
    10  FILLER     PIC X(50) VALUE
        '4567890123456789012345678901234567890123456789012345678'.
    10  FILLER     PIC X(32) VALUE
        '90123456789012345678901234567890
05  DET3.
    10  CAMPO28    PIC Z(3).
    10  FILLER     PIC XX     VALUE SPACE.
    10  CAMPO30    PIC X      OCCURS 120 TIMES.
    10  FILLER     PIC X(7)   VALUE SPACE.
05  TOR OCCURS 120 TIMES.
    10  TAM        PIC 9(3)   COMP 3.
    10  COD        PIC 9      COMP-3.
    10  FREQC      PIC 9(7)   COMP-3.
    10  ESTADO     PIC 9      COMP-3.
05  TABEST1.
    10  FILLER     PIC X(15) VALUE '   00
05  TABEST2 REDEFINES TABEST1 OCCURS 5 TIMES.
    10  VAL        PIC X OCCURS 3 TIMES.
05  TABCARAC1.
    10  FILLER     PIC X(50) VALUE
        '
    10  FILLER     PIC X(51) VALUE
        '                                .<(+|&          $*);¬-/
    10  FILLER     PIC X(24) VALUE
        '      ,%_>?          :#@'.
    10  FILLER PIC X VALUE QUOTE.
    10  FILLER  PIC X(25) VALUE
                                '="
    10  FILLER     PIC X(50) VALUE
```

```
           '                                                    ABCDEFGH'
       10  FILLER     PIC X(50) VALUE
           'I       JKLMNOPQR          STUVWXYZ      0123456789 '
       10  FILLER     PIC X(6)  VALUE '      '.
   05  TABCARAC2 REDEFINES TABCARAC1 OCCURS 256 TIMES.
   10  CONFG          PIC X.
   05  LISTA OCCURS 400 TIMES.
       10  TCOMP      PIC 9(3)  COMP-3.
       10  FREQ       PIC 9(5)  COMP-3.
       10  EST.
           15  BYTE   PIC X OCCURS 15 TIMES.
       10  LINK       PIC 9(3)  COMP-3.
   05  TABFORM.
       10  FILLER     PIC X(18) VALUE 'ALFANUMERICO      '.
       10  FILLER     PIC X(18) VALUE 'DECIMAL ZONADO    '.
       10  FILLER     PIC X(18) VALUE 'DECIMAL COMPACTADO'.
       10  FILLER     PIC X(18) VALUE 'ALFABETICO        '.
       10  FILLER     PIC X(18) VALUE 'BINARIO           '.
   05  TABFOR REDEFINES TABFORM OCCURS 5 TIMES.
       10  FORM       PIC X(18).
   05  RG  PIC X(200).
   05  REGISTRO REDEFINES RG.
       10  ITEM PIC X OCCURS 200 TIMES.
PROCEDURE DIVISION.
    DISPLAY 'DE A DATA XX/XX/XX' UPON CONSOLE      MODULO DE "INICIALIZAÇÃO"
    ACCEPT DAT1  FROM CONSOLE
    MOVE DAT1  TO DAT2
    DISPLAY 'DE O LIMITE = XXXXX' UPON CONSOLE
    ACCEPT LIMITE FROM CONSOLE
    PERFORM LER-CARTAO
    PERFORM SEGMENTACAO
    PERFORM SEGMENTACAO UNTIL FIM
    MOVE 'T' TO FIM-IMP
    PERFORM RPRINT
    CLOSE CARTAO
    STOP RUN.
LER-CARTAO.                                        MODULO "CARREGA TDR"
    OPEN INPUT CARTAO
    PERFORM LEITURA
    MOVE IDENTIF TO FB
    MOVE TAMANH TO NCAMP
    PERFORM CARREGA-TDR VARYING CONTA FROM 1 BY 1
     UNTIL CONTA > NCAMP.
CARREGA-TDR.
    PERFORM LEITURA
    MOVE TAMANH TO TAM (IDENTIF)
    MOVE CODIG  TO COD (IDENTIF)
    MOVE ZEROS  TO FREQC (IDENTIF) ESTADO (IDENTIF)
    ADD TAMANH TO RL.
LEITURA.
    READ CARTAO AT END
      DISPLAY 'ERRO NOS CARTOS' UPON CONSOLE.
SEGMENTACAO.
```

```
    PERFORM ENTRADA VARYING CONTA FROM 1 BY 1
      UNTIL CONTA > LIMITE OR FIM.
    MOVE INICIO TO APTN
    MOVE 57 TO CONTA-LINHA
    PERFORM RELATORIO1 VARYING CONTA FROM 1 BY 1
      UNTIL APTN = 0.
    MOVE INICIO TO APTN
    MOVE 57 TO CONTA-LINHA
    MULTIPLY ACUM1 BY RL GIVING TOT-ARQ
    PERFORM RELATORIO2 VARYING CONTA FROM 1 BY 1
      UNTIL APTN = 0.
    MOVE ZEROS TO CAMPO13
    MOVE ACUM1 TO CAMPO14
    MOVE ACUM2 TO CAMPO15
    MOVE ACUM4 TO CAMPO17
    MOVE ACUM6 TO CAMPO19
    MOVE ACUM8 TO CAMPO21
    MOVE ZEROS TO CAMPO22 CAMPO23 CAMPO24 CAMPO25 CAMPO26
    CAMPO16 CAMPO18 CAMPO20
    MOVE DET2 TO DADOS
    PERFORM IMPRIME
    MOVE ZEROS TO        ACUM2        ACUM4 ACUM5 ACUM6        AC
    MOVE 57 TO CONTA-LINHA.
    PERFORM RELATORIO3 VARYING CONTA FROM 1 BY 1
      UNTIL CONTA > NCAMP.
    MOVE ZEROS TO CAMPO1 CAMPO2
    MOVE ACUM4 TO CAMPO3
    MOVE ACUM5 TO CAMPO5
    MOVE ACUM8 TO CAMPO6
    MOVE ACUM6 TO CAMPO7
    MOVE ACUM15 TO CAMPO8
    MOVE SPACE TO CAMPO9R CAMPO10R CAMPO11R CAMPO4R
    MOVE SPACE  TO CAMPO12
    MOVE DET1 TO DADOS
    PERFORM IMPRIME
    MOVE ZEROS TO ACUM1 ACUM4 ACUM9 ACUM5 ACUM8 ACUM6 ACUM15
    MOVE 1 TO POSINI.
-------------------------------------------------------------------
ENTRADA.                                  MÓDULO "OBTER REGISTRO"
    MOVE 1 TO CONTAB
    ENTER LINKAGE
    CALL 'LERFITA' USING RG FIM-LEIT.
    ENTER COBOL
    IF NOT FIM
      PERFORM CONSTRUIR-ESTRUTURA VARYING CONTAC FROM 1 BY 1
        UNTIL CONTAC > NCAMP
      PERFORM CONSTRUIR-LISTA.
-------------------------------------------------------------------
CONSTRUIR-ESTRUTURA.      MÓDULO "ANALISAR CONTEÚDO PRÓXIMO CAMPO"
    MOVE COD (CONTAC) TO CODIGO
    PERFORM PROCESSA-BYTE VARYING APTB FROM CONTAB BY 1
      UNTIL APTB = CONTAB + TAM (CONTAC) - 1 OR
      ESTADO (CONTAC) = 1.
    IF ESTADO (CONTAC) = 0
      PERFORM PROCESSA-SINAL.
```

```
    ADD TAM (CONTAC) TO CONTAB.
PROCESSA BYTE.
    IF ITEM (APTB) NOT = VAL (CODIGO, 1)
      MOVE 1 TO ESTADO (CONTAC)
      ADD 1 TO FREQC (CONTAC).
PROCESSA-SINAL.
    IF ITEM (APTB) NOT = VAL (CODIGO, 2) AND
       ITEM (APTB) NOT = VAL (CODIGO, 3)
         MOVE 1 TO ESTADO (CONTAC)
         ADD 1 TO FREQC (CONTAC).
```

```
CONSTRUIR-LISTA.
    MOVE 8 TO CONTAB
    MOVE ZEROS TO APTB APTC EST (AVAIL)
    PERFORM FORMATA-LISTA VARYING CONTAC FROM 1 BY 1
      UNTIL CONTAC > NCAMP.
    IF CONTAB NOT = 8
      ADD 1 TO APTC
    ADD 1 TO APTB
      MOVE CONFG (APTB) TO BYTE (AVAIL, APTC).
    PERFORM INIC LISTA        UNTIL INICIO NOT = 0.
    MOVE INICIO TO APTN
    PERFORM PESQUISA-LISTA UNTIL
      APTN = 0 OR
      EST (AVAIL) NOT > EST (APTN).
    IF APTN = 0
      PERFORM INCLUIR-LISTA
    ELSE
      IF EST (AVAIL) < EST (APTN)
        PERFORM INCLUIR-LISTA
      ELSE
        ADD 1 TO FREQ (APTN).
    IF INICIO = AVAIL
      ADD 1 TO AVAIL.
```

```
FORMATA-LISTA.                                  MODULO "COMPOR ESTRUTURA"
    SUBTRACT 1 FROM CONTAB
    COMPUTE APTB = APTB + (2 ** CONTAB) * ESTADO (CONTAC)
    PERFORM INIC-VAR UNTIL CONTAB NOT = 0.
    MOVE ZEROS TO ESTADO (CONTAC).
INIC-VAR.
    ADD 1 TO APTC
    ADD 1 TO APTB
    MOVE CONFG (APTB) TO BYTE (AVAIL, APTC)
    MOVE 8 TO CONTAB
    MOVE ZEROS TO APTB.
```

```
INIC-LISTA.                                     MÓDULO "ALTERAR LISTA "
    MOVE AVAIL TO INICIO
    MOVE ZEROS TO TCOMP (INICIO) FREQ (INICIO) LINK (INICIO).
PESQUISA-LISTA.
    MOVE APTN TO APTA
    MOVE LINK (APTN) TO APTN.
INCLUIR-LISTA.
    MOVE ZEROS TO TCOMP (AVAIL)
    MOVE 1 TO FREQ (AVAIL)
```

```
     IF INICIO = APTN
       MOVE AVAIL TO INICIO
     ELSE
       MOVE AVAIL TO LINK (APTA).
     MOVE APTN TO LINK (AVAIL)
     ADD 1 TO AVAIL.
     IF AVAIL > 400
       MOVE '*' TO FIM-LEIT
       DISPLAY 'LISTA ESTOROU' UPON CONSOLE.
```

MODULO "GERAR RELATÓRIOS"

```
RELATORIO1.
     MOVE CONTA TO CAMPO28
     MOVE 0 TO APTC CONTAB
     PERFORM DECOD-LISTA VARYING CONTAC FROM 1 BY 1
       UNTIL CONTAC > NCAMP.
     PERFORM LIMPEZA VARYING CONTAC FROM CONTAC BY 1
       UNTIL CONTAC > 120
     IF CONTA-LINHA > 56
       PERFORM CABECALHO1.
     MOVE DET3 TO DADOS
     PERFORM IMPRIME
     ADD FREQ (APTN) TO ACUM1
     MOVE LINK (APTN) TO APTN.
DECOD-LISTA.
     PERFORM INIC-VAR-R UNTIL CONTAB NOT = 0.
     SUBTRACT 1 FROM CONTAB
     COMPUTE SOBRA = APTB - 2 ** CONTAB
     IF SOBRA NOT < 0
       MOVE SOBRA TO APTB
       ADD TAM (CONTAC) TO TCOMP (APTN)
       MOVE 1 TO CAMPO30 (CONTAC)
     ELSE
       MOVE 0 TO CAMPO30 (CONTAC).
INIC-VAR-R.
     MOVE 8 TO CONTAB
     ADD 1 TO APTC
     MOVE  BYTE (APTN, APTC) TO APTBS.
LIMPEZA.
     MOVE ' ' TO CAMPO30 (CONTAC).
CABECALHO1.
     ADD 1 TO PAG1
     MOVE PAG1 TO PAG
     MOVE 9 TO CONTA-LINHA
     MOVE 1 TO CONTRO1
     MOVE CABEC1 TO DADOS
     PERFORM RPRINT
     MOVE 0 TO CONTRO1
     MOVE CABEC6 TO DADOS
     PERFORM RPRINT
     MOVE ' ' TO CONTRO1
     MOVE CABEC7 TO DADOS
     PERFORM RPRINT
     MOVE ' ' TO CONTRO1
     MOVE CABEC8 TO DADOS
```

```
     PERFORM RPRINT.
IMPRIME.
     ADD 1 TO CONTA-LINHA
     MOVE ' ' TO CONTROL
     PERFORM RPRINT.
RPRINT.
     ENTER LINKAGE.
     CALL 'RPRINT' USING LINHA FIM-IMP.
     ENTER COBOL.
RELATORIO2.
     MOVE CONTA TO CAMPO13
     MOVE FREQ (APTN) TO CAMPO14
     DIVIDE FREQ (APTN) BY ACUM1 GIVING PROB
     MOVE PROB TO CAMPO15
     ADD PROB TO ACUM2.
     MOVE RL TO CAMPO16
     MULTIPLY RL BY FREQ (APTN) GIVING COMPARQ1
     MOVE COMPARQ1 TO CAMPO17
     ADD COMPARQ1 TO ACUM4
     MOVE TCOMP (APTN) TO CAMPO18
     ADD TCOMP (APTN) TO ACUM5
     MULTIPLY TCOMP (APTN) BY FREQ (APTN) GIVING COMPARQ2
     MOVE COMPARQ2 TO CAMPO19
     ADD COMPARQ2 TO ACUM6
     SUBTRACT TCOMP (APTN) FROM RL GIVING COMPABS1
     MOVE COMPABS1 TO CAMPO20
     SUBTRACT COMPARQ2 FROM COMPARQ1 GIVING COMPABS2
     MOVE COMPABS2 TO CAMPO21
     ADD COMPABS2 TO ACUM8
     DIVIDE COMPABS1 BY RL GIVING COMPREL1
     MOVE COMPREL1 TO CAMPO22
     DIVIDE COMPABS2 BY TOT-ARQ  GIVING COMPREL2
     MOVE COMPREL2 TO CAMPO23
     MULTIPLY COMPREL1 BY 100 GIVING COMPPER1
     MOVE COMPPER1 TO CAMPO24
     MULTIPLY COMPREL2 BY 100 GIVING COMPPER2
     MOVE COMPPER2 TO CAMPO25
     DIVIDE RL BY TCOMP (APTN) GIVING RAIOCOM1
     MOVE RAIOCOM1 TO CAMPO26
     IF CONTA-LINHA > 56
       PERFORM CABECALHO2.
     MOVE DET2 TO DADOS
     PERFORM IMPRIME
     MOVE ZEROS TO TCOMP (APTN)
     MOVE LINK (APTN) TO APTN.
CABECALHO2.
     ADD 1 TO PAG1
     MOVE PAG1 TO PAG
     MOVE 9 TO CONTA-LINHA
     MOVE 1 TO CONTROL
     MOVE CABEC1 TO DADOS
     PERFORM RPRINT
     MOVE 0 TO CONTROL
```

```
     MOVE CABEC4 TO DADOS
     PERFORM RPRINT
     MOVE ' ' TO CONTROL
     MOVE CABEC5 TO DADOS
     PERFORM RPRINT.
RELATORIO3.
     MOVE CONTA TO CAMPO1
     MOVE POSINI TO CAMPO2
     ADD TAM (CONTA) TO POSINI
     MOVE FREQC (CONTA) TO CAMPO3
     ADD FREQC (CONTA) TO ACUM4
     DIVIDE FREQC (CONTA) BY ACUM1 GIVING PROB
     MOVE PROB TO CAMPO4
     ADD PROB TO ACUM9
     MOVE TAM (CONTA) TO CAMPO5
     ADD TAM (CONTA) TO ACUM5
     MULTIPLY TAM (CONTA) BY ACUM1 GIVING COMPARQ1
     MOVE COMPARQ1 TO CAMPO6
     ADD COMPARQ1 TO ACUM8
     MULTIPLY TAM (CONTA) BY FREQC (CONTA) GIVING COMPARQ2
     MOVE COMPARQ2 TO CAMPO7
     ADD COMPARQ2 TO ACUM6
     SUBTRACT COMPARQ2 FROM COMPARQ1 GIVING COMPABS2
     MOVE COMPABS2 TO CAMPO8
     ADD COMPABS2 TO ACUM15
     DIVIDE COMPABS2 BY COMPARQ1 GIVING COMPREL2
     MOVE COMPREL2 TO CAMPO9
     MULTIPLY COMPREL2 BY 100 GIVING COMPPER2
     MOVE COMPPER2 TO CAMPO10
     DIVIDE COMPABS2 BY TOT-ARQ  GIVING RAIOCOM2
     MOVE RAIOCOM2 TO CAMPO11
     MOVE COD (CONTA) TO CODIGO
     MOVE FORM (CODIGO)  TO CAMPO12
     IF CONTA-LINHA > 56
       PERFORM CABECALHO3.
     MOVE DET1 TO DADOS
     PERFORM IMPRIME.
CABECALHO3.
     ADD 1 TO PAG1
     MOVE PAG1 TO PAG
     MOVE 9 TO CONTA-LINHA
     MOVE 1 TO CONTROL
     MOVE CABEC1 TO DADOS
     PERFORM RPRINT
     MOVE 0 TO CONTROL
     MOVE CABEC2 TO DADOS
     PERFORM RPRINT
     MOVE ' ' TO CONTROL
     MOVE CABEC3 TO DADOS
     PERFORM RPRINT.
```

APÊNDICE D

LISTAGEM FONTE

PROGRAMA GERADOR DE CÓDIGOS
(COMPØ4Ø)

```
ID DIVISION.
PROGRAM-ID.
    'COMP040'.
AUTHOR.
    MAURILIO G FILHO.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
    DECIMAL POINT IS COMMA.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT CARTAO ASSIGN TO SYS009-UR-3505-S.
    SELECT FITA ASSIGN TO SYS011-UT-3420-S
      RESERVE NO ALTERNATE AREA.
DATA DIVISION.
FILE SECTION.
FD  CARTAO
    LABEL RECORD IS OMITTED.
01  CARD.
    05  PRV            PIC 9(5).
    05  TAMANH         PIC 9(3).
    05  CODIG          PIC 9.
    05  SEGMENTO       PIC 9(3).
    05  COD SEGM       PIC 9.
    05  SINAL  PIC 9.
    05  FILLER  PIC X(66).
FD  FITA
    LABEL RECORD IS STANDARD.
01  REG.
    05  CTDS OCCURS 64  TIMES.
        10  CTAMS  PIC X.
        10  CNCMP  PIC X.
        10  FORMATO  PIC 999 COMP.
        10  POS-CODS PIC 9(4) COMP.
        10  POS-UNS PIC 9(4) COMP.
    05  CTDC OCCURS 120 TIMES.
      10  CINI PIC 999 COMP.
      10  CTAMC PIC 999 COMP.
      05  FILLER  PIC X(32).
WORKING-STORAGE SECTION.
01  TRABALHO.
    05  DAT1           PIC X(8).
    05  FIM-LEIT       PIC X     VALUE ZEROS.
        88  FIM        VALUE '*'.
    05  LINHA.
        10  CONTRO1    PIC X.
        10  DADOS      PIC X(132).
    05  FIM IMP        PIC X     VALUE SPACE.
    05  FB             PIC 9(3)  COMP-3.
    05  NCAMP          PIC 9(3)  COMP-3.
    05  NSEG           PIC 9(3)  COMP-3.
    05  CONT1          PIC 9(5)  COMP-3.
    05  CONT2          PIC 9(5)  COMP      VALUE 1.
```

```
 5  WORK1           PIC 9(5)  COMP-3    VALUE 0.
 5  LIMITE          PIC 9(5).
05  SINAL           PIC X     VALUE SPACE.
05  CONT3           PIC 9(5)  COMP-3.
05  CONTA           PIC 9(7)  COMP-3 VALUE ZEROS.
05  CODIGO          PIC 9     COMP-3.
05  COMPAC2.
    10  FILLER PIC X VALUE ' '.
     10  COMPAC1 PIC X.
 05  COMPAC REDEFINES COMPAC2 PIC 999 COMP.
05 NUM1  PIC 999 COMP.
05 NUM2  PIC 999 COMP.
05  WORK2W.
    10  FILLER     PIC X     VALUE ' '.
    10  WORK2      PIC X.
05  IDC REDEFINES WORK2W PIC 999 COMP.
05  AVAIL           PIC 9(5)  COMP-3    VALUE 1.
05  APTN            PIC 9(5)  COMP-3.
05  APTA            PIC 9(5)  COMP-3.
05  APTK            PIC 9(5) COMP-3.
05  WORK3           PIC 9(5)  COMP 3.
05  PAGS            PIC 9(4)  COMP-3 VALUE ZEROS.
05  CONT-LINHA      PIC 99    COMP-3 VALUE 57.
05  WORK10          PIC 99V99999999  COMP-3.
05  WORK4           PIC 9(9)  COMP-3.
05  WORK5           PIC 9(9)  COMP 3.
05  WORK6           PIC 9(9)  COMP-3.
05  WORK7           PIC 9V999999 COMP-3.
05  AVAILW          PIC 9(3)  COMP-3.
05  APTAW           PIC 9(3)  COMP-3.
05  APTNW           PIC 9(3)  COMP 3.
05  APTIND          PIC 9(3)  COMP-3.
05  WORK8           PIC 9(8)  COMP VALUE ZEROS.
05  CONT4           PIC 9(3)  COMP-3.
05  WORK9           PIC 9.
05  WORK11          PIC S99   COMP 3.
05  WORK12          PIC S9(8) COMP-3.
05  WORK13  PIC 99V9999 COMP 3.
05  ACUM1           PIC 9(6)  COMP-3 VALUE 0.
05  ACUM2           PIC 9V99999999 COMP-3 VALUE 0.
05  ACUM3           PIC 9(4)  COMP-3 VALUE 0.
05  ACUM4           PIC 9(9)  COMP-3 VALUE 0.
05  ACUM5           PIC 9(9)  COMP 3 VALUE 0.
05  ACUM6           PIC 9(9)  COMP-3 VALUE 0.
05  ACUM7           PIC 9(9)  COMP-3 VALUE 0.
05  ACUM8           PIC 99V999999 COMP-3 VALUE ZEROS.
05  TAM-REG PIC 9(3) COMP-3 VALUE ZEROS.
05  B               PIC 9.
05  W  PIC 999 COMP-3 VALUE 1.
05  W1  PIC 9 COMP-3.
05  LIMITE-TOTAL PIC 9(6).
05  CONTA-TOTAL PIC 9(6) COMP-3 VALUE 0.
05  COD WORK.
```

```
    10  FILLER      PIC X(4).
    10  CODY        PIC X(4).
05  WORK COD REDEFINES COD-WORK PIC 9(18) COMP.
05  CONTA-PREF     PIC 9(3)  COMP-3  VALUE 0.
05  CONTA-SUF      PIC 9(3)  COMP-3  VALUE 0.
05  CONT7          PIC 9(4)  COMP VALUE 0.
05  FIM-PREF       PIC X             VALUE ' '.
05  APT COD        PIC 9(3)  COMP-3  VALUE 0.
05  APT-COD1       PIC 9(3)  COMP-3  VALUE 0.
05  APT-COD2       PIC 9(3)  COMP-3  VALUE 0.
05  APT CODS1      PIC 9(3)  COMP-3  VALUE 1.
05  APT-CODS2      PIC 9(3)  COMP-3  VALUE 0.
05  APT-CR1        PIC 9(3)  COMP-3  VALUE 1.
05  APT-UNS        PIC 9(3)  COMP-3  VALUE 1.
05  APT-CODS3      PIC 9(3)  COMP-3.
05  CONT8          PIC 9(4)  COMP VALUE 0.
05  CONT9          PIC 9(4)  COMP VALUE 0.
05  CONT10         PIC 9(3)  COMP-3.
05  FMT            PIC 9.
05  APT-UNS1       PIC 9(3)  COMP-3.
05  APT CRC1 PIC 9(3) COMP-3 VALUE 1.
05  COMPACS2.
    10  FILLER  PIC X VALUE ' '.
    10  COMPACS1 PIC X.
05  COMPACS REDEFINES COMPACS2 PIC 999 COMP.
05  CONT11  PIC 9(3) COMP-3.
05  V241 PIC XX VALUE ' 0'.
05  V240 REDEFINES V241 PIC 999 COMP.
05  CAB1.
    10  FILLER     PIC X(50) VALUE
        '                                M A P A   E S T A T I'.
    10  FILLER     PIC X(50) VALUE
        ' S T I C O   D O S   S E G M E N T O S              '.
    10  DAT2       PIC X(8).
    10  FILLER     PIC X(18) VALUE
        '              PAG   '.
    10  PAG1       PIC Z(4).
    10  FILLER     PIC XX    VALUE SPACE.
05  CAB2.
    10  FILLER     PIC X(50) VALUE
        'NUM  FREQ.  PROBAB. TAM PART.TOT. PART.COMP COMPR.'.
    10  FILLER     PIC X(30) VALUE
        'ABS COMPR.REL COM  RAIO COMP  '.
    10  FILLER     PIC X(32) VALUE
        'DADOS  DOS   CAMPOS              '.
    10 FILLER PIC X(20) VALUE SPACE.
05  CAB3.
    10  FILLER     PIC X(50) VALUE
        'SEG  SEG.    SEG.   SEG   ARQ.      ARQ.      ARQ.'.
    10  FILLER     PIC X(30) VALUE
        '      ARQ.    PER     ARQ.    '.
    10  FILLER     PIC X(32) VALUE
        'NUM TAM PRV PRN   F O R M A T O '.
```

```
    10 FILLER PIC X(20) VALUE SPACE.
05  DET1.
    10  CAMPO1     PIC Z(3)  BLANK WHEN ZERO.
    10  FILLER     PIC X     VALUE SPACE.
    10  CAMPO2     PIC ZZZ.ZZ9 BLANK WHEN ZERO.
    10  FILLER     PIC X     VALUE SPACE.
    10  CAMPO3     PIC 9,999999 BLANK WHEN ZERO.
    10  FILLER     PIC X     VALUE SPACE.
    10  CAMPO4     PIC ZZ9   BLANK WHEN ZERO.
    10  FILLER     PIC X     VALUE SPACE.
    10  CAMPO5     PIC ZZZZZZZZ9 BLANK WHEN ZERO.
    10  FILLER     PIC X     VALUE SPACE.
    10  CAMPO6     PIC ZZZZZZZZ9 BLANK WHEN ZERO.
    10  FILLER     PIC X     VALUE SPACE.
    10  CAMPO7     PIC ZZZZZZZZ9 BLANK WHEN ZERO.
    10  FILLER     PIC X     VALUE SPACE.
    10  CAMPO8     PIC 9,999999  BLANK WHEN ZERO.
    10  FILLER     PIC XX    VALUE SPACE.
    10  CAMPO9     PIC ZZ9   BLANK WHEN ZERO.
    10  FILLER     PIC XX    VALUE SPACE.
    10  CAMPO10    PIC Z9,999999 BLANK WHEN ZERO.
    10  FILLER     PIC X     VALUE SPACE.
    10  FILLER     PIC X     VALUE SPACE.
    10  CAMPO12    PIC ZZ9   BLANK WHEN ZERO.
    10  FILLER     PIC X     VALUE SPACE.
    10  CAMPO13    PIC ZZ9   BLANK WHEN ZERO.
    10  FILLER     PIC X     VALUE SPACE.
    10  CAMPO14    PIC ZZ9   BLANK WHEN ZERO.
    10  FILLER     PIC X     VALUE SPACE.
    10  CAMPO15    PIC ZZ9   BLANK WHEN ZERO.
    10  FILLER     PIC X     VALUE SPACE.
    10  CAMPO16    PIC X(16).
    10 FILLER PIC X(20) VALUE SPACE.
05  CAB4.
    10  FILLER     PIC X(50) VALUE
        '                                   M A P A   E S T A T I'.
    10  FILLER     PIC X(50) VALUE
        ' S T I C O   D O S   A L F A B E T O S                  '.
    10  DAT3       PIC X(8).
    10  FILLER     PIC X(18) VALUE
        '          - PAG - '.
    10  PAG2       PIC Z(4).
    10  FILLER     PIC XX    VALUE SPACE.
05  CAB5.
    10  FILLER     PIC X(50) VALUE
        'ALF.   FREQ.     PROB.            G R U P O       '.
    10  FILLER     PIC X(50) VALUE
        '    CUSTO  COMPR.ORIG.  COMPR.COMPR. COMP. ABSOL. '.
    10  FILLER     PIC X(32) VALUE
        'COMP.    RELATIVA COM.PER RAIO  '.
05  CAB6.
    10  FILLER     PIC X(50) VALUE SPACE.
    10  FILLER     PIC X(50) VALUE
```

```
                    CAR    ARQ     CAR     ARQ    CAR    ARQ
    10  FILLER      PIC X(32) VALUE
       '  CAR       ARQ    .CAR ARQ COMP. '.
05  DET2.
    10  FILLER      PIC X(18) VALUE
        'SEGMENTO NUMERO = '.
    10  CAMPO17     PIC ZZZ.
    10  FILLER      PIC X(11)  VALUE SPACE.
    10  FILLER      PIC X(10)  VALUE 'FORMATO = '.
    10  CAMPO35     PIC X(16).
    10  FILLER    PIC X(74) VALUE SPACE.
05  DET3.
    10  CAMPO18     PIC X.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO34     PIC XX.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO19     PIC Z(9).
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO20     PIC 9,99999999.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPOS.
        15  CAMPO21  PIC X OCCURS 27 TIMES.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO22     PIC Z9,9999.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO23     PIC 9 BLANK WHEN ZERO.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO24     PIC ZZZZZZZZ9.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO25     PIC Z9 BLANK WHEN ZERO.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO26     PIC ZZZZZZZZ9.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO27     PIC -9 BLANK WHEN ZERO.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO28     PIC - ------9.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO29     PIC 9,999999 BLANK WHEN ZERO.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO30     PIC 9,999999 BLANK WHEN ZERO.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO31     PIC ZZ9    BLANK WHEN ZERO.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO32     PIC ZZ9    BLANK WHEN ZERO.
    10  FILLER      PIC X      VALUE SPACE.
    10  CAMPO33     PIC Z9,999 BLANK WHEN ZERO.
05  TAB-DIG.
    10  FILLER      PIC X(15) VALUE '   00          '.
05  TABDIG REDEFINES TAB-DIG OCCURS 5 TIMES.
    10  VAL         PIC X OCCURS 3 TIMES.
05  TABFORM.
    10 FILLER PIC X(21) VALUE
       'ALFANUMERICO    60000'.
```

```
    10 FILLER PIC X(21) VALUE
       'DECIMAL ZONADO  31512'.
    10 FILLER PIC X(21) VALUE
       'DECIMAL COMPACT.14768'.
    10 FILLER PIC X(21) VALUE
       'ALFABETICO      28256'.
    10 FILLER PIC X(21) VALUE
   'BINARIO          00999'.
05  TABFOR REDEFINES TABFORM OCCURS 5 TIMES.
    10  FORM      PIC X(16).
    10  QUANT-CARAC PIC 99.
    10 FORMAT PIC 999.
05  TDS OCCURS 64  TIMES.
    10  TAM-SEG   PIC 9(3)  COMP-3.
    10  FREQ SEG  PIC 9(7)  COMP-3.
    10 COD-SEG PIC 9 COMP-3.
10  FREQ-ORC-S  PIC 9(9) COMP-3.
    10  APT-TDC   PIC 9(3)  COMP-3.
05  TDC OCCURS 120 TIMES.
    10  TAM CMP   PIC 9(3)  COMP 3.
    10  COD-CMP   PIC 9     COMP-3.
    10  PRV-CMP   PIC 9(3)  COMP-3.
    10  SINAL CMP PIC 9.
05  PERMIT.
    10  FILLER    PIC X(256)      VALUE SPACE.
    10  FILLER    PIC X(32)       VALUE
       'X  X                              '.
    10  FILLER    PIC X(32)       VALUE
       '        X   X       X   X   X     '.
    10  FILLER    PIC X(32)       VALUE
       'X                                 '.
    10  FILLER    PIC X(32)       VALUE
       '          X   X   X   X           '.
    10  FILLER    PIC X(32)       VALUE
       'X   X                             '.
    10  FILLER    PIC X(32)       VALUE
       '        X   X   X   X             '.
    10  FILLER    PIC X(32)       VALUE
       '                                  '.
    10  FILLER    PIC X(32)       VALUE
       '        X   X   X   X   X   X     '.
    10  FILLER    PIC X(256)      VALUE SPACE.
    10  FILLER    PIC X(32)       VALUE
       ' X  XX XXX XXX XXX XXX XXX XXX X'.
    10  FILLER    PIC X(32)       VALUE
       'XX XXX X                          '.
    10  FILLER    PIC X(32)       VALUE
       ' X  XX XXX XXX XXX XXX XXX XXX X'.
    10  FILLER    PIC X(32)       VALUE
       'XX XXX X                          '.
    10  FILLER    PIC X(32)       VALUE
       '        X  XX  XX  XX  XX  XX  X'.
    10  FILLER    PIC X(32)       VALUE
```

```
           'X  XX  X                                  '.
       10  FILLER     PIC X(32)       VALUE
           'XXX XXX XXX XXX XXX XXX XXX XXX '.
       10  FILLER     PIC X(32)       VALUE
           'XXX XXX            X   X        X '.
       10  FILLER     PIC X(4)        VALUE
           'XXXX'.
   05  PERMITE REDEFINES PERMIT OCCURS 257 TIMES.
       10  PMR-CRC  PIC X OCCURS 4 TIMES.
   05  TAB TRAD.
       10  FILLER     PIC X(50) VALUE
           ' 00   01   02   03   04   05   06   07   08   09  '.
       10  FILLER     PIC X(50) VALUE
           ' 0A   0B   0C   0D   0E   0F   10   11   12   13  '.
       10  FILLER     PIC X(50) VALUE
           ' 14   15   16   17   18   19   1A   1B   1C   1D  '.
       10  FILLER     PIC X(50) VALUE
           ' 1E   1F   20   21   22   23   24   25   26   27  '.
       10  FILLER     PIC X(50) VALUE
           ' 28   29   2A   2B   2C   2D   2E   2F   30   31  '.
       10  FILLER     PIC X(50) VALUE
           ' 32   33   34   35   36   37   38   39   3A   3B  '.
       10  FILLER     PIC X(50) VALUE
           ' 3C   3D   3E   3F   40   41   42   43   44   45  '.
       10  FILLER     PIC X(50) VALUE
           ' 46   47   48   49   4A  .4B  <4C  (4D  +4E  |4F  '.
       10  FILLER     PIC X(50) VALUE
           '&50   51   52   53   54   55   56   57   58   59  '.
       10  FILLER     PIC X(50) VALUE
           ' 5A  $5B  *5C  )5D  ;5E  ¬5F  =60  |61   62   63  '.
       10  FILLER     PIC X(50) VALUE
           ' 64   65   66   67   68   69   6A  ,6B  %6C  6D  '.
       10  FILLER     PIC X(50) VALUE
           '>6E  ?6F   70   71   72   73   74   75   76   77  '.
       10  FILLER     PIC X(50) VALUE
           ' 78   79  :7A  #7B  @7C   7D  =7E   7F   80   81  '.
       10  FILLER     PIC X(50) VALUE
           ' 82   83   84   85   86   87   88   89   8A   8B  '.
       10  FILLER     PIC X(50) VALUE
           ' 8C   8D   8E   8F   90   91   92   93   94   95  '.
       10  FILLER     PIC X(50) VALUE
           ' 96   97   98   99   9A   9B   9C   9D   9E   9F  '.
       10  FILLER     PIC X(50) VALUE
           ' A0   A1   A2   A3   A4   A5   A6   A7   A8   A9  '.
       10  FILLER     PIC X(50) VALUE
           ' AA   AB   AC   AD   AE   AF   B0   B1   B2   B3  '.
       10  FILLER     PIC X(50) VALUE
           ' B4   B5   B6   B7   B8   B9   BA   BB   BC   BD  '.
       10  FILLER     PIC X(50) VALUE
           ' BE   BF   C0  AC1  BC2  CC3  DC4  EC5  FC6  GC7  '.
       10  FILLER     PIC X(50) VALUE
           'HC8  IC9   CA   CB   CC   CD   CE   CF   D0  JD1  '.
       10  FILLER     PIC X(50) VALUE
```

```
           'KD2  LD3  MD4  ND5  OD6  PD7  QD8  RD9   DA   DB  '.
       10  FILLER     PIC X(50) VALUE
           ' DC   DD   DE   DF   E0   E1  SE2  TE3  UE4  VE5  '.
       10  FILLER     PIC X(50) VALUE
           'WE6  XE7  YE8  ZE9   EA   EB   EC   ED   EE   EF  '.
       10  FILLER     PIC X(50) VALUE
           '0F0  1F1  2F2  3F3  4F4  5F5  6F6  7F7  8F8  9F9  '.
       10  FILLER     PIC X(35) VALUE
           ' FA   FB   FC   FD   FE   FF  DFL  '.
   05  TABTRAD REDEFINES TAB-TRAD OCCURS 257 TIMES.
       10  CRC        PIC X.
       10  HEXAD      PIC XX.
       10  APT-CRC    PIC 9(3) COMP-3.
   05  SEGM OCCURS  60 TIMES.
       10  FREQ CRC   PIC 9(9) COMP OCCURS 64 TIMES.
   05  LISTA-WORK OCCURS 520 TIMES.
       10  LINKEW     PIC 9(3) COMP-3.
       10  PROBW      PIC 9V99999999 COMP-3.
       10  LINKDW     PIC 9(3)  COMP-3.
       10  LINKLW     PIC 9(3)  COMP 3.
       10  ENDW       PIC 9(3) COMP-3.
   05  REG1.
   10  PADRAD  PIC X OCCURS 512 TIMES.
   10  TABUNS OCCURS 256 TIMES.
       15  SHF-UNS PIC X.
       15  DESL-UNS PIC X.
   05  COD-S.
       10  COD S1 PIC X(1024).
       10  COD-S2 PIC X(1024).
   05  CODS REDEFINES COD-S PIC X(4) OCCURS 512 TIMES.
   05  TAB CRC.
       10  CRT OCCURS 512 TIMES.
           15  SHF-CRC    PIC X.
           15  CRC-CRC    PIC X.
   05  TAB-WORK.
       10  ITW OCCURS 64 TIMES.
           15  CRCW       PIC 9(3) COMP-3.
           15  CODW       PIC X(4).
           15  PREFW      PIC 9(3) COMP-3.
           15  SUFW       PIC 9(3) COMP-3.
   05  TAB DESL-CODS.
       10  FILLER     PIC X(148)      VALUE ZEROS.
       10  FILLER     PIC X(14)       VALUE
           '01020003040506'.
       10  FILLER     PIC X(20)       VALUE ZEROS.
       10  FILLER     PIC X(14)       VALUE
           '07080910001112'.
       10  FILLER     PIC X(16)       VALUE ZEROS.
       10  FILLER     PIC X(8)        VALUE
           '13141516'.
       10  FILLER     PIC X(24)       VALUE ZEROS.
       10  FILLER     PIC X(12)       VALUE
           '171819202122'.
```

```
10  FILLER     PIC X(130)      VALUE ZEROS.
10  FILLER     PIC X(18)       VALUE
    '2324252627282930 31'.
10  FILLER     PIC X(14)       VALUE ZEROS.
10  FILLER     PIC X(18)       VALUE
    '3233343536373839 40'.
10  FILLER     PIC X(16)       VALUE ZEROS.
10  FILLER     PIC X(16)       VALUE
    '4142434445464748'.
10  FILLER     PIC X(12)       VALUE ZEROS.
10  FILLER     PIC X(20)       VALUE
    '4950515253545556 5758'.
10  FILLER     PIC X(14)       VALUE
    '00000000000059'.
10  FILLER     PIC X(50)       VALUE
    '20202020202020202020202020202020202020202020202020'.
10  FILLER PIC X(50) VALUE
    '20202020202020202020202020202020202020202020202020'.
10  FILLER     PIC X(50)       VALUE
    '20202020202020202020202020202020202020202020202020'.
10  FILLER     PIC X(50)       VALUE
    '20202020202020202020202020202020202020202020202020'.
10  FILLER     PIC X(50)       VALUE
    '20202020202020202020202020202020202020202020202020'.
10  FILLER     PIC X(50)       VALUE
    '20202020202020202020202020202020202020202020202020'.
10  FILLER     PIC X(50)       VALUE
    '20202020202020202020202020202020202020202020202020'.
10  FILLER     PIC X(34)       VALUE
    '2020202020202020202020202020202020'.
10  FILLER     PIC X(20)       VALUE
    '0001020304050607 0809'.
10  FILLER     PIC X(12)       VALUE
    '202020202020'.
10  FILLER     PIC X(20)       VALUE
    '1011121314151617 1819'.
10  FILLER     PIC X(50)       VALUE
    '20202020202020202020202020202020202020202020202122'.
10  FILLER     PIC X(28)       VALUE
    '2324252627282920202020202030'.
10  FILLER     PIC X(482)      VALUE ZEROS.
10  FILLER     PIC X(32)       VALUE
    '01020304050607080900001011001213'.
10  FILLER     PIC X(386)      VALUE ZEROS.
10  FILLER     PIC X(18)       VALUE
    '0102030405060708 09'.
10  FILLER     PIC X(14)       VALUE ZEROS.
10  FILLER     PIC X(18)       VALUE
    '1011121314151617 18'.
10  FILLER     PIC X(16)       VALUE ZEROS.
10  FILLER     PIC X(16)       VALUE
    '1920212223242526'.
10  FILLER     PIC X(44)       VALUE ZEROS.
```

```
        10  FILLER      PIC XX          VALUE '27'.
    05  TABDESLCODS    REDEFINES TAB-DESL CODS OCCURS 4 TIMES.
        10  DESL-CODS  PIC 99 OCCUPS 257 TIMES.
    05  INDIC          PIC 999   COMP-3 OCCURS 27 TIMES.
    05  RG             PIC X(200).
    05  REGISTRO REDEFINES RG.
        10  ITEM       PIC X   OCCURS 200 TIMES.
PROCEDURE DIVISION.
    MOVE ZEROS TO WORK-COD
    PERFORM INICIALIZAR
    MOVE ZEROS TO W
    PERFORM CRIAR-PADRAO VARYING CONT1 FROM 1 BY 1
      UNTIL CONT1 > NCAMP.
    DISPLAY 'DE A DATA XX/XX/XX' UPON CONSOLE.   MÓDULO DE "INICIALIZAÇÃO"
    ACCEPT DAT1 FROM CONSOLE
    MOVE DAT1 TO DAT2 DAT3
    DISPLAY 'DE O LIMITE - XXXXX' UPON CONSOLE
    ACCEPT LIMITE FROM CONSOLE
    DISPLAY ' DE NUM. MAX. DE REGS A SER PROCESSADO-XXXXXX'
      UPON CONSOLE
    ACCEPT LIMITE-TOTAL FROM CONSOLE
    PERFORM PROCESSAR UNTIL FIM
    MOVE 'T' TO FIM-IMP
    PERFORM RPRINT
    OPEN OUTPUT FITA
    WRITE REG
    WRITE REG FROM REG1
    WRITE REG FROM TAB-CRC
    WRITE REG FROM COD S1
    WRITE REG FROM COD-S2
    CLOSE FITA
    STOP RUN.
INICIALIZAR.                                   MODULO " CARREGAR TDS/TDC "
    OPEN INPUT CARTAO
    PERFORM LER-CARTAO
    MOVE PRV      TO FB
    MOVE TAMANH   TO NCAMP
    MOVE SEGMENTO TO NSEG
    PERFORM LER-CARTAO
    PERFORM CARREGA-TDS VARYING CONT1 FROM 1 BY 1
      UNTIL CONT1 > NSEG OR CONT1 > 64.
    MOVE '  '  TO CTDS (CONT1).
CARREGA-TDS.
    MOVE FORMAT (COD-SEGM) TO FORMATO (CONT1)
    MOVE COD-SEGM TO COD-SEG (CONT1)
    MOVE QUANT CARAC (COD-SEGM) TO FREQ-CRC-S (CONT1)
    PERFORM CARREGA-TDC VARYING CONT2 FROM CONT2 BY 1
      UNTIL SEGMENTO NOT = CONT1
    SUBTRACT W FROM CONT2 GIVING COMPAC
    MOVE COMPAC1 TO CNCMP (CONT1)
    MOVE CONT2 TO W
    SUBTRACT 1 FROM CONT2 GIVING APT-TDC (CONT1)
    MOVE WORK1 TO TAM-SEG (CONT1)
```

```
     SUBTRACT 1 FROM WORK1 GIVING COMPAC
     MOVE COMPAC1 TO CTAMS (CONT1)
     ADD WORK1 TO TAM-REG
     MOVE ZEROS TO WORK1 FREQ-SEG (CONT1)
     PERFORM ZERA-SEGM VARYING CONT3 FROM 1 BY 1
       UNTIL CONT3 >  60.
CARREGA-TDC.
     SUBTRACT 1 FROM PRV GIVING CINI (CONT2)
     SUBTRACT 1 FROM TAMANH GIVING CTAMC (CONT2)
     MOVE SINA1 TO SINAL-CMP (CONT2)
     MOVE TAMANH TO TAM-CMP (CONT2)
     MOVE CODIG TO COD-CMP (CONT2)
     MOVE PRV    TO PRV-CMP (CONT2)
     ADD TAMANH  TO WORK1
     PERFORM LER-CARTAO.
LER-CARTAO.
     READ CAPTAO AT END
       MOVE 999 TO SEGMENTO
       CLOSE CARTAO.
ZERA-SEGM.
     MOVE 1 TO FREQ-CRC (CONT3, CONT1).
-------------------------------------------------------------
CRIAR-PADRAO.                          MÓDULO "CRIAR PADRÃO"
     MOVE COD-CMP (CONT1) TO CODIGO
     PERFORM MOVE PAD VARYING CONT2 FROM 1 BY 1 UNTIL
       CONT2 > TAM-CMP (CONT1) - 1
     MOVE SINAL-CMP (CONT1) TO W1
     ADD 1 TO W
     MOVE VAL (CODIGO, W1) TO PADRAO (W).
MOVE PAD.
     ADD 1 TO W
     MOVE VAL (CODIGO, 1) TO PADRAO (W).
-------------------------------------------------------------
PROCESSAR.
     PERFORM CRIAR-LISTA VARYING CONT1 FROM 1 BY 1
       UNTIL CONT1 > LIMITE OR
             FIM.
     MOVE 1 TO CONT2 WORK3
     MOVE 57 TO CONT-LINHA
     PERFORM CRIAR-EST-SEG VARYING CONT1 FROM 1 BY 1
       UNTIL CONT1 > NSEG
     MOVE ACUM1 TO CAMPO2
     MOVE ACUM2 TO CAMPO3
     MOVE ACUM3 TO CAMPO4
     MOVE ACUM4 TO CAMPO5
     MOVE ACUM5 TO CAMPO6
     MOVE ACUM6 TO CAMPO7
     MOVE ZEROS TO CAMPO12 CAMPO13 CAMPO14 CAMPO15
                   ACUM2 ACUM3 ACUM4 ACUM5 ACUM6
     MOVE SPACE TO CAMPO16
     MOVE DET1 TO DADOS
     PERFORM IMPRIME
     PERFORM CRIAR-ALFAB VARYING CONT1 FROM 1 BY 1
       UNTIL CONT1 > NSEG.
-------------------------------------------------------------
CRIAR-LISTA.                           MÓDULO "OBTER REGISTRO"
```

```
    ENTER LINKAGE
    CALL 'LERFITA' USING RG FIM-LEIT.
    ENTER COBOL
    ADD 1 TO CONTA-TOTAL
    IF CONTA-TOTAL > LIMITE-TOTAL
      MOVE '*' TO FIM-LEIT.
    IF NOT FIM
    ADD 1 TO CONTA
    MOVE 1 TO CONT3
    PERFORM PROC-SEG VARYING CONT2 FROM 1 BY 1
      UNTIL CONT2 > NSEG.
PROC-SEG.
    PERFORM INEXISTE-SEG VARYING CONT3 FROM CONT3 BY 1
      UNTIL SINAL = 1 OR
            CONT3 > APT-TDC (CONT2)
    IF SINAL = 1
      IF CONT2 = 1
      MOVE 1 TO CONT3
    ELSE
    SUBTRACT 1 FROM CONT2 GIVING CONT4
      ADD 1 APT-TDC (CONT4)      GIVING CONT3.
    IF SINAL = 1
      PERFORM EXISTE SEG VARYING CONT3 FROM CONT3 BY 1
      UNTIL CONT3 > APT-TDC (CONT2)
      ADD 1 TO FREQ-SEG (CONT2)
      MOVE SPACE TO SINAL
      ADD 1 TO ACUM1.
    MOVE SPACE TO SINAL.
INEXISTE-SEG.                    MÓDULO "ANÁLISE CONTEÚDO PRÓXIMO SEGMENTO"
    MOVE COD-CMP (CONT3) TO CODIGO
    PERFORM TESTA DIG VARYING CONT4 FROM PRV-CMP (CONT3) BY 1
      UNTIL CONT4 = TAM-CMP (CONT3) + PRV-CMP (CONT3) - 1 OR
            SINAL = 1.
    IF SINAL = SPACE
      PERFORM TESTA SINAL.
TESTA-DIG.
    IF ITEM (CONT4) NOT = VAL (CODIGO, 1)
      MOVE 1 TO SINAL.
TESTA-SINAL.
    IF ITEM (CONT4) NOT = VAL (CODIGO, 2) AND
      ITEM (CONT4) NOT =  VAL (CODIGO, 3)
      MOVE 1 TO SINAL.
EXISTE SEG.                      MÓDULO "DECOMPOR CAMPOS DO SEGMENTO"
    IF COD-CMP (CONT3) NOT = 5
    IF COD CMP (CONT3) = 3
        PERFORM PRC-COMP VARYING CONT4 FROM PRV-CMP (CONT3) BY 1
          UNTIL CONT4 = TAM-CMP (CONT3) + PRV-CMP (CONT3)
    MOVE ZEROS TO IDC
      ELSE
        PERFORM PRC-ZONA VARYING CONT4 FROM PRV-CMP (CONT3) BY 1
          UNTIL CONT4 = TAM-CMP (CONT3) + PRV-CMP (CONT3).
PRC-COMP.
    MOVE ITEM (CONT4) TO COMPAC1
```

```
     DIVIDE COMPAC  BY 16 GIVING NUM1 REMAINDER NUM2
     ADD NUM1 V240 GIVING IDC
     PERFORM CONST LIST
     ADD NUM2 V240 GIVING IDC
     PERFORM CONST-LIST.
PRC-ZONA.
     MOVE ITEM (CONT4) TO WORK2
     PERFORM CONST LIST.
CONST-LIST.
     ADD 1 TO IDC
     IF APT CRC (IDC) = ZEROS
       MOVE AVAIL TO APT-CRC (IDC)
       IF AVAIL = 60
         DISPLAY 'ESTOUROU TABELA DE FREQUENCIA' UPON CONSOLE
       ELSE
         ADD 1 TO AVAIL.
     MOVE APT-CRC (IDC) TO APTN
     ADD 1 TO FREQ-CRC (APTN, CONT2)
     IF PMP CRC (IDC, CODIGO) = 'X'
     ADD 1 TO FREQ-CRC-S (CONT2).
```

MODULO "GERAR MAPA ESTATISTICO DO SEGMENTO "

```
CRIAR-EST SEG.
     MOVE CONT1 TO CAMPO1
     MOVE FREQ-SEG (CONT1) TO CAMPO2
     DIVIDE FREQ-SEG (CONT1) BY ACUM1 GIVING WORK7
     MOVE WORK7 TO CAMPO3
     ADD WORK7 TO ACUM2
     MOVE TAM-SEG (CONT1) TO CAMPO4
     ADD TAM-SEG (CONT1) TO ACUM3
     MULTIPLY CONTA BY TAM-SEG (CONT1) GIVING WORK4
     MOVE WORK4 TO CAMPO5
     ADD WORK4 TO ACUM4
     MULTIPLY TAM-SEG (CONT1) BY FREQ-SEG (CONT1) GIVING WORK5
     MOVE WORK5 TO CAMPO6
     ADD WORK5 TO ACUM5
     SUBTRACT WORK5 FROM WORK4 GIVING WORK6
     MOVE WORK6 TO CAMPO7
     ADD WORK6 TO ACUM6
     MULTIPLY CONTA BY TAM-REG GIVING WORK4
     DIVIDE WORK6 BY WORK4 GIVING WORK7
     MOVE WORK7 TO CAMPO8
     MULTIPLY WORK7 BY 100 GIVING WORK1
     MOVE WORK1 TO CAMPO9
     IF FREQ SEG (CONT1) NOT = 0
     DIVIDE CONTA BY FREQ-SEG (CONT1) GIVING WORK10
     MOVE WORK10 TO CAMPO10
     ELSE
       MOVE ZEROS TO CAMPO10.
     MOVE ZEROS TO CAMPO12 CAMPO13 CAMPO14 CAMPO15
     MOVE SPACE TO CAMPO16
     IF CONT LINHA > 56
       PERFORM CABECALHO1.
     MOVE DET1 TO DADOS
     PERFORM IMPRIME
```

```
     MOVE ZEROS TO CAMPO1 CAMPO2 CAMPO3 CAMPO4 CAMPO5 CAMPO6
                   CAMPO7 CAMPO8 CAMPO9 CAMPO10 WORK1
     PERFORM DADOS-CAMPO VARYING CONT2 FROM CONT2 BY 1
       UNTIL CONT2 > APT-TDC (CONT1).
DADOS CAMPO.
     ADD 1 TO WORK1
     MOVE WORK1 TO CAMPO12
     MOVE TAM-CMP (CONT2) TO CAMPO13
     MOVE PRV-CMP (CONT2) TO CAMPO14
     MOVE WORK3 TO CAMPO15
     MOVE COD-CMP (CONT2) TO CODIGO
     MOVE FORM (CODIGO) TO CAMPO16
     IF CONT LINHA > 56
       PERFORM CABECALHO1.
     MOVE DET1 TO DADOS
     PERFORM IMPRIME.
     ADD  TAM-CMP (CONT2) TO WORK3.
CABECALHO1.
     ADD 1 TO PAGS
     MOVE PAGS TO PAG1
     MOVE 9 TO CONT LINHA
     MOVE 1 TO CONTRO1
     MOVE CAB1 TO DADOS
     PERFORM RPRINT
     MOVE 0 TO CONTRO1
     MOVE CAB2 TO DADOS
     PERFORM RPRINT
     MOVE ' ' TO CONTRO1
     MOVE CAB3 TO DADOS
     PERFORM RPRINT.
IMPRIME.
     ADD 1 TO CONT-LINHA
     MOVE ' ' TO CONTRO1
     PERFORM RPRINT.
-------------------------------------------------------------
RPRINT.
     ENTER LINKAGE
     CALL 'RPRINT' USING LINHA FIM-IMP.
     ENTER COBOL.
CRIAR-ALFAB.
     MOVE 3 TO AVAILW
     MOVE 2 TO PROBW (2) LINKDW (1)
     MOVE 0 TO PROBW (1) LINKDW (2) LINKLW (1) LINKLW (2)
     ENDW (1) ENDW (2) LINKEW (1)
     MOVE 1 TO LINKEW (2)
     MOVE FREQ CRC S (CONT1) TO ACUM7
     PERFORM CRIA-LISTW VARYING CONT2 FROM 1 BY 1
       JNTIL CONT2 > 257
     MOVE LINKDW (1) TO APTA
     MOVE LINKDW (APTA) TO APTN
     PERFORM CRIAR-APVORE
       UNTIL APTN = 2
     PERFORM ZERA INDEX VARYING CONT2 FROM 1 BY 1
       UNTIL CONT2 > 27
```

```
      MOVE LINKEW (2) TO APTN
      MOVE 99999 TO APTA
      MOVE 0 TO APTIND
      PERFORM CALC-CODE-INDEX
        UNTIL APTN = 1
      MOVE LINKEW (2) TO APTN
      MOVE ZERO TO                WORK8 APT-COD
      MOVE 57 TO CONT-LINHA
      PERFORM GERA-REL-ALF VARYING CONT2 FROM 1 BY 1
        UNTIL CONT2 > APTIND
      MOVE SPACE TO CAMPO18
      MOVE ACUM7 TO CAMPO19
      MOVE ACUM2 TO CAMPO20
      MOVE SPACE TO CAMPO5
      MOVE ACUM8 TO CAMPO22
      MOVE ACUM5 TO CAMPO24
      MOVE ACUM6 TO CAMPO26
      MOVE ACUM4 TO CAMPO28
      MOVE SPACE TO CAMPO34
      COMPUTE WORK10 = ACUM4 / (CONTA * TAM-REG * 8)
      MOVE WORK10 TO CAMPO30
      MOVE ZEROS TO CAMPO29          CAMPO31 CAMPO32 CAMPO33
                    ACUM2 ACUM8 ACUM3 ACUM5        ACUM6
                    ACUM4
                    CAMPO23 CAMPO25 CAMPO27
      MOVE DET3 TO DADOS
      PERFORM IMPRIME
      ADD 1 TO APT-COD
      MOVE 0 TO PREFW (APT-COD)
      SUBTRACT 1 FROM APT-COD
      IF FIM PERFORM CONSTRUIR-TABELAS.
-------------------------------------------------------------------
CRIA-LISTW.                                    "OBTER CODE INDEX"
                                    MÓDULO "CLASSIFICAR ALFABETO"
      MOVE COD SEG (CONT1) TO APTK
      MOVE APT-CRC (CONT2) TO APTN
      IF PMF CRC (CONT2, APTK) = 'X'
        IF APTN = 0
        DIVIDE 1 BY ACUM7 GIVING PROBW (AVAILW)
          PERFORM CRIA-CELULA
      ELSE
        DIVIDE FREQ-CRC (APTN, CONT1) BY ACUM7 GIVING
          PROBW (AVAILW)
          PERFORM CRIA-CELULA
      ELSE
        IF APTN NOT = 0
          IF FREQ-CRC (APTN, CONT1) > 1
          DISPLAY 'CARACTER NAO PERMITIDO = '
          HEXAD (CONT2)  FORM (APTK) FREQ-CRC (APTN, CONT1).
CRIA-CELULA.
      MOVE CONT2 TO ENDW (AVAILW)
      MOVE 0 TO LINKLW (AVAILW)
      MOVE 1 TO APTA
      MOVE LINKDW (1) TO APTN
      PERFORM SORT-LISTW UNTIL PROBW (AVAILW) NOT > PROBW (APTN)
```

```
    MOVE LINKDW (APTA) TO LINKDW (AVAILW)
    MOVE LINKEW (APTN) TO LINKEW (AVAILW)
    MOVE AVAILW TO LINKDW (APTA) LINKEW (APTN)
    ADD 1 TO AVAILW.
SORT-LISTW.
    MOVE APTN TO APTA
    MOVE LINKDW (APTA) TO APTN.
```

```
CRIAR-ARVORE.                          MÓDULO "CONSTRUIR ÁRVORE DO CÓDIGO DE HUFFMAM"
    ADD PROBW (APTA) PROBW (APTN) GIVING PROBW (AVAILW)
    MOVE ZEROS TO LINKLW (AVAILW) ENDW (AVAILW) LINKEW (AVAILW)
    MOVE AVAILW TO LINKLW (APTA) LINKLW (APTN)
    MOVE LINKDW (APTN) TO APTNW
      MOVE APTN TO APTAW.
    PERFORM PESQ-LISTW
      UNTIL PROBW (AVAILW) NOT > PROBW (APTNW)
    MOVE LINKDW (APTAW) TO LINKDW (AVAILW)
    MOVE AVAILW TO LINKDW (APTAW)
    ADD 1 TO AVAILW
    MOVE LINKDW (APTN) TO APTA
    MOVE LINKDW (APTA) TO APTN.
PESQ-LISTW.
    MOVE APTNW TO APTAW
    MOVE LINKDW (APTNW) TO APTNW.
```

```
ZERA-INDEX.                            MÓDULO "OBTENÇÃO DO CODE INDEX"
    MOVE ZEROS TO INDIC (CONT2).
CALC-CODE-INDEX.
    IF LINKLW (APTN)  NOT = APTA
      MOVE LINKLW (APTN)  TO APTNW APTA
    PERFORM CONTA-BIT VARYING CONT3 FROM 0 BY 1
      UNTIL APTNW = 0
    IF CONT3 > APTIND
      MOVE CONT3 TO APTIND.
    MOVE LINKEW (APTN) TO APTN
      ADD 1 TO INDIC (CONT3).
CONTA-BIT.
    MOVE LINKLW (APTNW) TO APTNW.
```

```
GERA REL ALF.                          MODULO "GERAR MAPA ESTATISTICO DOS ALFABETOS"
    PERFORM GERA-LINHA-ALF VARYING CONT3 FROM 1 BY 1
      UNTIL CONT3 > INDIC (CONT2)
    COMPUTE WORK8 = WORK8 * 2.
GERA-LINHA-ALF.
    MOVE ENDW (APTN) TO APTA.
    IF COD-SEG (CONT1) = 3
    MOVE 4 TO B
    ELSE
    MOVE 8 TO B.
    MOVE CRC (APTA) TO CAMPO18
    MOVE HEXAD (APTA) TO CAMPO34
    MOVE APT-CRC (APTA) TO APTA
    IF APTA NOT = 0
      MOVE FREQ-CRC (APTA, CONT1) TO CAMPO19
      DIVIDE FREQ-CRC (APTA, CONT1) BY ACUM7 GIVING WORK10
      MULTIPLY B BY FREQ-CRC (APTA, CONT1) GIVING WORK4
```

```
   MULTIPLY CONT2 BY FREQ-CRC (APTA, CONT1) GIVING WORK5
ELSE
   MOVE 1 TO CAMPO19
   DIVIDE 1 BY ACUM7 GIVING WORK10
   MOVE B TO WORK4
   MOVE CONT2 TO WORK5.
MOVE WORK10 TO CAMPO20
ADD WORK10 TO ACUM2
MULTIPLY WORK10 BY CONT2 GIVING WORK13
MOVE WORK13 TO CAMPO22
ADD WORK13 TO ACUM8
MOVE B TO CAMPO23
MOVE WORK4 TO CAMPO24
ADD WORK4 TO ACUM5
MOVE CONT2 TO CAMPO25
MOVE WORK5 TO CAMPO26
ADD WORK5 TO ACUM6
SUBTRACT CONT2  FROM B      GIVING WORK11
MOVE WORK11 TO CAMPO27
SUBTRACT WORK5 FROM WORK4 GIVING WORK12
MOVE WORK12 TO CAMPO28
ADD WORK12 TO ACUM4
DIVIDE WORK11 BY B GIVING WORK7
MOVE WORK7 TO CAMPO29
COMPUTE WORK10 = WORK12 / (CONTA * TAM-REG * 8)
MOVE WORK10 TO CAMPO30
MULTIPLY WORK7 BY 100 GIVING WORK1
MOVE WORK1 TO CAMPO31
MULTIPLY WORK10 BY 100 GIVING WORK1
MOVE WORK1 TO CAMPO32
DIVIDE WORK4 BY WORK5  GIVING WORK10
MOVE WORK10 TO CAMPO33
MOVE WORK8 TO WORK5
PERFORM MOVE CAMPO21 VARYING CONT4 FROM 1 BY 1
   UNTIL CONT4 > CONT2
IF FIM
   ADD 1 TO APT-COD
   MOVE ENDW (APTN) TO CRCW (APT-COD)
   ADD CONT2 TO WORK-COD
   MOVE CODY TO CODW (APT-COD)
   MOVE CONTA-PREF TO PREFW (APT-COD)
   MOVE CONTA-SUF TO SUFW (APT-COD)
   MOVE ZEROS TO WORK-COD CONTA-PREF CONTA-SUF
MOVE ' ' TO FIM-PREF
   IF APT-COD = 64
      DISPLAY 'ESTOROU TAB-WORK' UPON CONSOLE
      STOP RUN.
PERFORM BRANQUEAR VARYING CONT4 FROM CONT4 BY 1
   UNTIL CONT4 > 27.
IF CONT-LINHA > 56
   PERFORM CABECALHO2.
MOVE DET3 TO DADOS
PERFORM IMPRIME.
```

```
    ADD 1 TO WORK8
    MOVE LINKEW (APTN) TO APTN.
MOVE-CAMPO21.
    COMPUTE WORK6 = 2 ** (CONT2 - CONT4)
    DIVIDE  WORK5 BY WORK6 GIVING WORK9 REMAINDER WORK5
    MOVE WORK9 TO CAMPO21 (CONT4).
    IF FIM
      COMPUTE WORK-COD = WORK-COD + 2 ** (32 - CONT4) * WORK9
      IF FIM PREF = ' '
        ADD 1 TO CONTA-PREF
        IF WORK9 NOT = 1
          MOVE '*' TO FIM-PREF
        ELSE
          NEXT SENTENCE
      ELSE
        ADD 1 TO CONTA-SUF.
BRANQUEAR.
    MOVE SPACE TO CAMPO21 (CONT4).
CABECALHO2.
    ADD 1 TO PAGS
    MOVE PAGS TO PAG2
    MOVE 9 TO CONT LINHA
    MOVE 1 TO CONTRO1
    MOVE CAB4 TO DADOS
    PERFORM RPRINT
    MOVE 0 TO CONTRO1
    MOVE CAB5 TO DADOS
    PERFORM RPRINT
    MOVE ' ' TO CONTRO1
    MOVE CAB6 TO DADOS
    PERFORM RPRINT
    MOVE CONT1 TO CAMPO17
    MOVE FORM (APTK) TO CAMPO35
    MOVE 0 TO CONTRO1
    MOVE DET2 TO DADOS
    PERFORM RPRINT.
-------------------------------------------------------------------
CONSTRUIR-TABELAS.                          MÓDULO "CONSTRUIR TABELAS"
    IF CONT1 > 64
      DISPLAY 'ESTOUROU TDS' UPON CONSOLE
      STOP RUN
    ELSE
      MOVE CONT9 TO POS-CODS (CONT1)
      MOVE CONT8 TO POS-UNS  (CONT1)
      MOVE CONT7 TO TABUNS (APT-UNS)
      ADD 2 TO CONT8
    MOVE 0 TO CONT11
      PERFORM CONSTRUIR-TABS VARYING APT-COD1 FROM 1 BY 1
        UNTIL APT-COD1 > APT-COD
      ADD QUANT-CARAC (FMT) TO APT-CODS1
      ADD PREFW (APT-COD2) 1 TO APT-UNS.
CONSTRUIR-TABS.
    ADD 1    APT-COD1 GIVING CONT10
    PERFORM QUEBRA PREF VARYING APT-COD2 FROM APT-COD1 BY 1
```

```
      UNTIL PREFW (APT-COD2) NOT = PREFW (CONT10) OR
            CONT10 > APT-COD
    ADD PREFW (APT COD2) APT-UNS GIVING APT UNS1
    IF APT-UNS1 > 256
      DISPLAY 'ESTOUROU TABUNS' UPON CONSOLE STOP RUN.
    MOVE CONT11 TO COMPAC
    MOVE COMPAC1 TO DESL-UNS (APT-UNS1)
    MOVE SUFW (APT-COD2) TO COMPAC
    MOVE COMPAC1 TO SHF-UNS (APT-UNS1)
    ADD 2 TO CONT8
    PERFORM MOVE TABS VARYING APT-COD1 FROM APT-COD1 BY 1
      UNTIL APT-COD1 > APT-COD2
    SUBTRACT 1 FROM APT-COD1.
QUEBRA PREF.
    ADD 1 TO CONT10.
MOVE TABS.
    MOVE COD-SEG (CONT1) TO FMT
    MOVE CRCW (APT-COD1) TO APT-CODS2
    ADD  DESL CODS (FMT, APT CODS2) APT-CODS1 GIVING APT-CODS3
    MOVE CODW (APT-COD1) TO CODS (APT-CODS3)
    ADD 4 TO CONT9
    SUBTRACT SUFW (APT-COD1) FROM SUFW (APT-COD2) GIVING COMPAC
    PERFORM EXPANDE-CRC VARYING APT-CRC1 FROM APT-CRC1 BY 1
      UNTIL APT CRC1 = APT CR1 + 2 ** COMPAC OR
      APT-CRC1 > 512
    MOVE APT-CRC1 TO APT-CR1.
EXPANDE-CRC.
    SUBTRACT 1 FROM APT-CODS2 GIVING COMPACS
    MOVE COMPACS1 TO CRC-CRC (APT-CRC1)
    MOVE COMPAC1 TO SHF-CRC (APT-CRC1)
    ADD 2 TO CONT11
    ADD 2 TO CONT7.
```

```
        SOURCE RECORDS =    986      DATA ITEMS =    373      F
        PARTITION SIZE = 227208      LINE COUNT =     56      F
ECT*    PMAP RELOC ADR =   NONE      SPACING    =      1      F
ECT*     NOLISTX       APOST      NOSYM    NOCATALR        LIS
ECT*     NOCLIST       FLAGW        ZWB    NOSUPMAP      NOXR
ECT*     NOSTATE       TRUNC        SEQ    NOSYMDMP      NODE
ECT*                NOCOUNT                            NOVERBSI
*          NONE
```

APÊNDICE E

LISTAGEM FONTE

PROGRAMA DE CONVERSÃO
(COMPØ5Ø)

```
ID DIVISION.
PROGRAM ID.
    'COMP050'.
AUTHOR.
    MAJR ILIO G FILHO.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT ARQUIVO ASSIGN TO SYS012-UT-3420-S.
DATA DIVISION.
FILE SECTION.
FD  ARQUIVO
    RECORDING MODE IS U
    LABEL RECORD IS STANDARD
    RECORD CONTAINS 1 TO 12000 CHARACTERS
    DATA RECORD IS REGIST.
01  REGIST.
    05  REGISTRO OCCURS 60 TIMES DEPENDING ON FBLK.
        10  BYTE PIC X OCCURS 200 TIMES DEPENDING ON RL.
WORKING-STORAGE SECTION.
    01  WORK.
        05  DADOS.
            10  FB      PIC 999.
            10  FILLER PIC X.
            10  RL      PIC 999.
    05  FBLK PIC 999 COMP.
        05  TIPO PIC X.
        05  REG.
            10  POS1  PIC X.
            10  RESTO PIC X(199).
        05  CONTA PIC 999 COMP VALUE 1.
    05  CHAVE PIC X VALUE SPACE.
PROCEDURE DIVISION.
    DISPLAY 'DE FATOR DE BLOCO-TAMANHO REGISTRO XXX-XXX'
      UPON CONSOLE                                    MODULO "INICIALIZAÇÃO"
    ACCEPT DADOS FROM CONSOLE
    DISPLAY 'DATA C/D SE COMPRESSAO / DESCOMPRESSAO'
      UPON CONSOLE
    ACCEPT TIPO FROM CONSOLE
    IF TIPO = 'C'
      PERFORM COMPRESSAO
    ELSE
    OPEN OUTPUT ARQUIVO
    PERFORM DESCOMPRESSAO UNTIL POS1 = '*'.
      DISPLAY 'FIM2' UPON CONSOLE.
    CLOSE ARQUIVO
    PERFORM FIM.
COMPRESSAO.
    MOVE FB TO FBLK
    OPEN INPUT ARQUIVO
    MOVE '/' TO POS1
    ENTER LINKAGE
    CALL 'DCOMP' USING REG
```

```
      ENTER COBOL
      STOP 'FITA LIBERADA 230'
      PERFORM PROC-COMP UNTIL CHAVE = '1'.
PROC COMP.                                        MODULO " LEITURA "
      READ ARQUIVO AT END
        DISPLAY 'FIM1' UPON CONSOLE.
      PERFORM GRAVA-COMP VARYING CONTA FROM 1 BY 1
        UNTIL CONTA > FB OR CHAVE = '1'.
GRAVA COMP.                                  MODULO " COMPRIME   REGISTRO "
      IF BYTE (CONTA, 1) = '*'
      MOVE '1' TO CHAVE.
      MOVE REGISTRO (CONTA) TO REG
      MOVE '*' TO BYTE (CONTA, 1)
      ENTER LINKAGE
      CALL 'COMP' USING REG
      ENTER COBOL.
DESCOMPRESSAO.                                MODULO DE " GRAVAÇÃO "
      PERFORM PROC-DCOMP VARYING CONTA FROM 1 BY 1
        UNTIL CONTA > FB OR
              POS1 = '*'
      SUBTRACT 1 FROM CONTA GIVING FBLK.
      WRITE REGIST.
PROC-DCOMP.                          MODULO " DESCOMPRIME  REGISTRO "
      ENTER LINKAGE
      CALL 'DCOMP' USING REG.
      ENTER COBOL
      IF POS1 NOT = '*'
      MOVE REG TO REGISTRO (CONTA)
      ELSE
       SUBTRACT 1 FROM CONTA.
FIM.
      STOP RUN.
```

```
$*          SOURCE RECORDS =     85      DATA ITEMS =      16
$*          PARTITION SIZE = 227208      LINE COUNT =      56
$ EFFECT*   PMAP RELOC ADR =   NONE      SPACING    =       1
$ EFFECT*    NOLISTX       APOST       NOSYM    NOCATALR
$ EFFECT*    NOCLIST       FLAGW         ZWB    NOSUPMAP
$ EFFECT*    NOSTATE       TRUNC         SEQ    NOSYMDMP
$ EFFECT*                  NOCOUNT                        NOV
IONS*          NONE
```

APÊNDICE F

LISTAGEM FONTE

SUBROTINA DE COMPRESSÃO/DESCOMPRESSÃO

```
         PRINT  NOGEN
COMPAC   START  0
         ENTRY  COMP,DCOMP


***
*  DEFINE NOMES PARA OS REGISTRADORES ***
***

W1       EQU    1    REGISTRADOR DE TRABALHO N. 1
BASE     EQU    2    REGISTRADOR BASE
BASE2    EQU    3    REGISTRADOR BASE COMPLEMENTAR
PSEGC    EQU    4    APONTADOR PARA AREA ONDE ESTA SEG.DESCOMPRESSO
RBS      EQU    4    REGISTRADOR QUE CONTEM N. SEGMENTOS AUSENTES
W4       EQU    4    REGISTRADOR DE TRABALHO N. 4
PSEGD    EQU    5    APONTADOR PARA AREA ONDE ESTA SEG.COMPRESSO
PTDS     EQU    6    APONTADOR DA TABELA DE SEGMENTOS
BCODS    EQU    6
CENT     EQU    7    REGISTRADOR QUE RECEBE CODIGO DE CARATER = HSF
PTDC     EQU    7    APONTADOR DA TABELA DE CAMPOS
PPAD     EQU    8    APONTADOR DOS PADROES DE AUSENCIA
SAI      EQU    8    REGISTRADOR  CONTENDO OS GRUPOS DE SAIDA HSF
ENT      EQU    9    REGISTRADOR QUE RECEBE GRUPOS HSF DE SEGC
PREG     EQU    9    APONTADOR DO PARAMETRO (REGISTRO DESCOMPRESSO)
PIO      EQU    10   APONTADOR DOS SEGMENTOS NAS IOAREAS
CSAI     EQU    10   CONTEM ESPACO DISPONIVEL EM SAI
BCRC     EQU    10   REGISTRADOR BASE DA TABELA DE CARACTERES
BTRAD1   EQU    11   REGISTRADOR BASE DA TABELA DE TRADUCAO
BUNS     EQU    11   REGISTRADOR BASE DA TABELA DE PREFIXOS
OVFL     EQU    11   REGISTRADOR QUE CONTROLA OVERFLOW NAS IOAREAS
NCMP     EQU    12   REGISTRADOR QUE CONTEM N./TAMANHO DOS SEGMENTOS
INSEGD   EQU    12   REGISTRADOR DE INCREMENTO
W2       EQU    13   REGISTRADOR DE TRABALHO N. 2
TMSEG    EQU    13   REGISTRADOR DE COMPARACAO
CONTA    EQU    15   CONTEM TAMANHO DO SEGMENTO COMPRESSO
W3       EQU    15   REGISTRADOR DE TRABALHO N. 3


***
*    D.T.F. DO ARQUIVO A SER DESCOMPRESSO ***
***

ARQE     DTFMT  BLKSIZE=1024,
                DEVADDR=SYS010,
                EOFADDR=FIMENT,
                FILABL=STD,
                IOAREA1=ENT1,
                IOAREA2=ENT2,
                IOREG=(10),
```

```
                RECFORM=FIXJNB,
                REWIND=UNLOAD,
                TYPEFILE=INPUT


***
*  INICIALIZACAO ROTINA DE DESCOMPRESSAO - DCOMP ***
***

         USING    *,BASE,BASE2      INDICA REGISTRADORES BASE
DCOMP    STM      14,12,12(13)      SALVA REGISTRADORES DO PROG. PRINCIPAL
         LR       BASE,15           CARREGA REGISTRADOR BASE
         L        BASE2,ADDR        CARREGA REGIST. BASE2 COM DCOMP+4096
         ST       13,SAV13          SALVA REGISTRADOR 13 EM SAV13
         LM       PIO,OVFL,SAVENT   RESTAURA REGISTRADOR PIO E OVFL
         L        PREG,0(,1)        CARREGA EM PREG ENDERECO DO PARAMETRO


***
*   CARREGAMENTO DAS TABELAS ***
***

         BC       0,RTSEG           SE NAO E 1- EXECUCAO DESVIAR RTSEG
         OI       *+3,X'F0'         MUDA INST. ANT. P/ DESVIO INCONDICIONAL
         OPEN     ARQE              ABRE ARQUIVO COMPRESSO
         LA       PTDS,TDS          CARREGA END. INICIO  TABELAS EM PTDS
         LA       W4,5              INICIALIZA REGISTRADOR W4 COM 5
LACO1    LA       W2,4            INICIALIZA REGISTRADOR W2 COM 4
         GET      ARQE              OBTER PROXIMO BLOCO
LACO2    MVC      0(256,PTDS),0(PIO) TRANSFERE BLOCO
         LA       PIO,256(,PIO)     DA AREA DE ENTRADA
         LA       PTDS,256(,PTDS)   PARA TABELA NA
         BCT      W2,LACO2          MEMORIA
         BCT      W4,LACO1          VERIFICA SE TABELAS COMPLETAS
         CLI      0(PREG),X'61'     VERIFICA 1-. POSICAO PARAMETRO COM '/'
         BE       FIMENT            AFIRMATIVO DESVIAR FINAL DA ROTINA
         GET      ARQE              OBTER PROXIMO BLOCO-DADOS COMPRESSOS


***
*   INICIAL. A APONTADORES / REGISTRADORES ***
***

RTSEG    CLI      0(PIO),X'00'      VERIFICA FIM DE ARQUIVO
         BE       FIMENT            DESVIA PARA ROTINA DE FIM SE FIM ARQ.
         LA       PTDC,TDC          CARREGA END.TAB.DESCRICAO DE CAMPOS
         LA       PTDS,TDS          CARREGA END.TAB.DESCRICAO DE SEGMENTOS
         LA       PPAD,PAD          CARREGA END. DO PADRAO DE AUSENCIA
```

```
         LA       NCMP,0              ZERA REGIST. NCMP
         LA       W3,0                ZERA REGIST. R3


***
*    VERIFICACOES ***
***

RTSEG1   CLI      1(PTDS),X'00'       VERIFICA SE FIM DE REGISTRO
         BE       FIMPG               DESVIA P/ ROT.FIM REGISTRO SE FIM
         TM       0(PID),X'80'        VERIFICA SE PROX.SEG. PRESENTE/AUSENTE
         BZ       DBLOCK              DESVIA SE SEG. PRESENTE


***
*    ROTINA DE OBTER PADRAO ***
***

         IC       W4,0(,PID)          OBTEM BYTE SENTINELA DA IOAREA
         N        W4,=F'127'          RETIRA MARCA DO BYTE SENTINELA
OBPAD    IC       W3,0(,PTDS)         OBTEM TAMANHO SEG. DA TDS
         STC      W3,*+5              COLOCA NA M/C TAMANHO DO SEGMENTO
         MVC      SEGD,0(PPAD)        MOVE PADRAO PARA SEG
         LA       PPAD,1(W3,PPAD)     POSICIONA PPAD PARA O PROXIMO PADRAO
         BAL      W1,DMENB            DESVIA PARA A ROTINA DE DESMEMBRAR
         BCT      W4,OBPAD            TESTA SE BYTE SENTINELA ZEROU

         LA       PIO,1(,PIO)         AVANCO APONTADOR DA IOAREA 1 BYTE
         LA       OVFL,1(,OVFL)       INCREMENTO REGISTRADOR OVERFLOW
         C        OVFL,=F'1024'       COMPARO OVERFLOW COM TAMANHO IOAREA
         BNE      RTSEG1              DESVIO PARA RTSEG1 SE DIFERENTE
         GET      ARQE                SE = OBTER PROXIMO BLOCO
         LA       OVFL,0                   ZERAR OVERFLOW
         LA       W3,0
         B        RTSEG1                   DESVIAR PARA RTSEG1


***
*    ROTINA DE EXTRACAO DOS SEGMENTOS ***
***

DBLOCK   IC       NCMP,0(,PID)        COLOCA EM NCMP TAMANHO REAL DO SEG. - 1
         IC       W3,0(PTDS)          COLOCA EM W3 TAMANHO DESCOMPRESSO - 1
         LA       PPAD,1(W3,PPAD)     POSICIONA PPAD NO PROX.PADRAO DO SEG.
         LA       W2,1024             CARREGA W2 COM TAMANHO DA IOAREA
         LA       OVFL,1(,OVFL)       INCREMENTA OVFL
         LA       PSEGC,SEGC          CARREGA ENDERECO DE SEGC EM PSEGC
```

```
         BXH      OVFL,NCMP,OVERFL DESVIO P/ OVERFL SE IOAREA ESTOUROU
REST     STC      NCMP,*+5         COLOCA NA MVC TAMANHO DO SEGMENTO
         MVC      0(,PSEGC),0(PIO) MOVE SEG.COMPRESSO DA IOAREA P/ SEGC.
         LA       PIO,1(NCMP,PIO)  POSICIONO APONTADOR IOAREA PROX.SEG.
         C        OVFL,=F'1024'    COMPARO OVFL COM TAMANHO IOAREA
         BNE      RTHSF            DESVIO PARA RTHSF SE DIFERENTE
         GET      ARQE             SE = OBTER PROXIMO BLOCO
         LA       OVFL,0                ZERAR OVERFLOW (OVFL)
         B        RTHSF                 DESVIAR PARA RTHSF

OVERFL   SR       OVFL,W2          DETERMINA DE QUANTO E O OVERFLOW
         SR       NCMP,OVFL        DETERMINA QUANTO DO SEG.FICA NA IOAREA
         STC      NCMP,*+5         COLOCA NA MVC TAMANHO DA PARTE DO SEG.
         MVC      0(,PSEGC),0(PIO) MOVE PARTE DO SEGMENTO COMPRESSO PDSEG
         GET      ARQE             OBTER PROXIMO BLOCO
         LA       PSEGC,1(NCMP,PSEGC) POSICIONA SEGC P/RECEBER RESTANTE
         LR       NCMP,OVFL        COLOCA EM NCMP SOBRA DO SEGMENTO
         S        NCMP,=F'1'       DECREMENTA NCMP
         B        REST             DESVIA PARA REST PARA COMPLEMENTACAO


***
*    ROTINA DE HSF ***
***



***
*    INICIALIZACAO HSF ***
***

RTHSF    STM      PTDC,OVFL,SAVSG SALVA DE PTDC A OVFL
         LA       PSEGC,SEGC+1    CARREGA ENDERECO DE SEGC+1 EM PSEGC
         LA       PSEGD,SEGD-1    CARREGA ENDERECO DESEGD EM PSEGD
         LA       CENT,0
         LA       W1,0
         LA       BUNS,UNS        CARREGA END.TAB.PREFIXOS EM BUNS
         AH       BUNS,6(,PTDS)   POSICIONA BUNS NA TAB.PREFIXOS P/SEG.
         LA       BCRC,CRC        CARREGA END.TAB.CARACTERES EM BCRC
         AH       BCRC,0(,BUNS)   POSICIONA BCRC NA TAB.CARACTERES P/SEG.



***
*    VERIFICA FORMATO DOS SEGMENTOS ***
***

         CLC      2(2,PTDS),=H'768' VERIFICA FORMATO SEG.E DEC.COMPAC.
         BL       ZONADO           DESVIA P/ZONADO SE FORMATO CARACTER
```

```
         BE        COMPAT              DESVIA P/COMPAT SE FORMATO COMPACTADO
         MVC       *+7(1),SEGC         SE BINARIO
         MVC       SEGD,SEGC+1         MOVE SEG.COMPRESSO P/ SEG.DESCOMPRESSO
         B         LIGA                DESVIA PARA LIGA


***
*    OBTER CARACTERES ****
***

COMPAT   NI        DESVIO+1,X'00'      FORCAR EXECUCAO 'COMPACTA SEGMENTO'
         IC        W1,0(,PTOS)         COLOCA EM W1 COMP. SEG.DESCOMPRESSO
         LA        W1,0(W1,W1)         DUPLICA CONTEUDO EM W1
         B         SEGUE               DESVIA PARA SEGUE
ZONADO   IC        W1,0(,PTOS)         COLOCA EM W1 COMP. DESCOMPRESSO SEG.
SEGUE    LA        TMSEG,2(W1,PSEGD)   COLOCA EM TMSEG END.FINAL SEG.DESCOM.
         LA        INSEGD,1            INICIALIZA REG.INCREMENTO COM 1
         LA        W1,0                INICIALIZA REGISTRADOR W1 COM 0
         L         ENT,0(,PSEGC)       CARREGA PROXIMOS 4 BYTES DE SEGC EM ENT
PROX     LA        W3,0                INICIALIZA REGISTRADOR W3 COM 0
       LA        SAI,0
         BXH       PSEGD,INSEGD,DESVIO DESVIA PARA DESVIO SE PSEGD > TMSEG
CONTA1   LA        W3,2(,W3)           ADICIONA 2 AO CONTEUDO DE W3
         LA        W1,1(,W1)           ADICIONA 1 AO CONTEUDO DE W1
CONTA2   ALR       ENT,ENT             DESLOCA P/ESQUERDA 1 BIT DE ENT
         BC        3,CONTA1            DESVIA P/CONTA1 SE CARRY
         BC        4,CODIF             DESVIA P/CODIF SE NAO CARRY E ENT#0
         C         W1,=F'33'           VERIFICA SE TODOS BITS ENT DESLOCADOS
         BNE       CODIF               SE NAO = DESVIA P/ CODIF
         LA        PSEGC,4(,PSEGC)     POSICIONA PSEGC PROX.4 BYTES DE SEGC
         LA        W1,1                INICIALIZA REGISTRADOR W1 COM 1
         L         ENT,0(,PSEGC)       CARREGA PROXIMOS 4 BYTES DE SEGC EM ENT
         B         CONTA2              DESVIA P/CONTA2
CODIF    IC        CENT,0(W3,BUNS)     CARREGA CENT COM TAMANHO MAIOR SUFIXO
         AR        W1,CENT             ADICIONA TAMANHO SUF.A BITS DESLOCADOS
         C         W1,=F'32'           VERIFICA SE SUFIXO ESTA NO REG. ENT
         BH        ESTOUR              SE NAO =DESVIA ESTOUR
         SLDL      SAI,0(CENT)         SHIFTA SUFIXO PARA SAI
         IC        CENT,1(W3,BUNS)     CARREGA CENT COM DESLOCAMENTO TAB.CRC
         LA        W3,0(SAI,SAI)
         LA        W3,0(W3,CENT)       DETERMINA ENDERECO RELATIVO
         IC        CENT,1(W3,BCRC)     OBTEM CARATER FONTE DE TAB.CRC
         STC       CENT,0(,PSEGD)      ARMAZENA CARACTER FONTE EM SEGD
         IC        CENT,0(W3,BCRC)     CARREGA CENT COM SHIF RETORNO
         SRDL      SAI,0(CENT)         RETORNA BITS A ENT
         SR        W1,CENT             POSICIONA CONTADOR DE BITS W1 DE ENT
         B         PROX                DESVIA P/PROX
ESTOUR   S         W1,=F'32'           DETERMINA DE QUANTO E O OVERFLOW
         SR        CENT,W1             DETERMINA QUANTOS BITS RESTAM EM ENT
         SLDL      SAI,0(CENT)         DESLOCA BITS RESTANTES DE ENT
```

```
         L        ENT,4(,PSEGC)    CARREGA PROXIMOS 4 BYTES DE SEGC EM ENT
         SLDL     SAI,0(W1)        DES_DCA BITS COMPLEMENTARES DO SUFIXO
         IC       CENT,1(W3,BJNS)  CARREGA CENT COM DESLOCAMENTO TAB.CRC
         LA       W3,0(SAI,SAI)
         LA       W3,0(W3,CENT)     ETERMINA ENDERECO RELATIVO
         IC       CENT,1(W3,BCRC)  OBTEM CARACTER FONTE DE TAB.CRC
         STC      CENT,0(,PSEGD)   ARMAZENA CARACTER FONTE EM SEGD
         IC       CENT,0(W3,BCRC)   CARREGA CENT COM SHIFT RETORNO
         CR       CENT,W1          VERIFICA SE SHIF RETORNO>DESL.COMPLENT.
         BH       RECOMP           SE MAIOR=DESVIA PORECOMP
         SRDL     SAI,0(CENT)      RETORNA BITS A ENT
         LA       PSEGC,4(,PSEGC)  POSICIONA PSEGC PROX. 4 BYTES DE SEGC
         SR       W1,CENT          POSICIONA CONTADOR DE BITS W1 DE ENT
         B        PROX             DESVIA P/ PROX
RECOMP   LA       ENT,0            INICIALIZA REGISTRADOR ENT COM 0
         SR       CENT,W1          DETERMINA BITS A RETORNAR
         SRA      SAI,0(W1)
         SRDL     SAI,0(CENT)      RETORNA BITS A ENT
         LA       W1,32            INICIALIZA REGISTRADOR W1 COM 32
         SR       W1,CENT          POSICIONA CONTADOR DE BITS W1 DE ENT
         B        PROX             DESVIA P/ PROX
DESVIO   B        LIGA             DESVIA LIGA
         OI       *-3,X'F0'        ALTERA INST.ANTERIOR PARA DESVIO NULO


***
*    COMPACTA SEGMENTO ***
***

         LA       PSEGC,SEGC       CARREGA ENDERECO DE SEGC EM PSEGC
         LA       PSEGD,SEGD       CARREGA ENDERECO DE SEGD EM PSEGD
         IC       W1,0(,PTOS)      COLOCA EM W1 COMP.DESCOMPRESSO DO SEG.
         LA       W1,2(,W1)        INCREMENTA W1 DE  2
         L        NCMP,=F' 7'      INICIALIZA REGISTRADOR NCMP COM  7
         LA       TMSEG,0          INICIALIZA REGISTRADOR TMSEG COM 0
RETORN   BXLE     W1,NCMP,FIMCOM   DESVIA PARA FIMCOM SE W1<= 0
         PACK     0(8,PSEGC),0(15,PSEGD) COMPACTA PROX. 16 BYTES
         LA       PSEGC,7(,PSEGC)  ADICIONA 7 A PSEGC
         LA       PSEGD,14(,PSEGD) ADICIONA 1 A PSEGD
         B        RETORN           DESVIA PARA RETORN
FIMCOM   LA       W1,6(,W1)        ADICIONA 6 A W1
         LA       W2,0(W1,W1)      DUPLICA W1 INCREMENTANDO
         SLL      W1,4             DESLOCA W1 4 A ESQUERDA
         LA       W1,0(W2,W1)      ADICIONA W1 A W2
         STC      W1,*+5           ALTERA INST. PACK PO COMPACTAR FIM
         PACK     0(0,PSEGC),0(0,PSEGD) COMPACTA ULTIMOS BYTES
         LA       NCMP,0
         LA       PSEGD,SEGC
         LM       PTOC,OVFL,SAVSG  RESTAURAR REGS PTOC A OVFL
         LA       W1,RTSEG1        CARREGA EM W1 ENDERECO DE RTSEG1
```

```
         B         DMENB1


***
*    FINAL DESCOMPRESSAP HSF ***
***

LIGA     LM        PTDC,OVFL,SAVSG RESTAURAR REGS PTDC A OVFL
         LA        W1,RTSEG1       CARREGA EM W1 ENDERECO DE RTSEG1


***
*    ROTINA DE DESMEMBRAR SEGMENTOS EM CAMPOS ***
***

DMENB    LA        PSEGD,SEGD      CARREGA END.SEG.DESCOMPRESSO EM PSEGD
DMENB1   IC        NCMP,1(,PTDS)   COLOCA EM NCMP N. CAMPOS DO SEGMENTO

VOLTA    LH        W2,0(,PTDC)     COLOCA EM W2 POSICAO INICIO CAMPO REG.
         LA        W2,0(W2,PREG)   COLOCA EM W2 ENDERECO DESTA LOCALIZACAO
         LH        W3,002(,PTDC)   COLOCA EM W3 TAMANHO DO CAMPO
         STC       W3,*+5          COLOCA NA MVC TAMANHO DO CAMPO
         MVC       0(,W2),0(PSEGD) MOVE CAMPO DE SEGD PARA REGISTRO
         LA        PTDC,4(,PTDC)   POSICIONA PTDC PROXIMO CAMPO
         LA        PSEGD,1(W3,PSEGD) POSICIONA PSEGD PROXIMO CAMPO EM SEGD
         BCT       NCMP,VOLTA      SE N. CAMPOS NAO ZERO DESVIA PARA VOLTA
         LA        PTDS,8(,PTDS)   POSICIONA PTDS PROXIMO SEGMENTO
         BR        W1  DESVIA PARA ENDERECO EM W1.


***
*    ROTINA DE FINALIZACOES ***
***

FIMENT   CLOSE     ARQE            FECHA ARQUIVO COMPRESSO ARQE
         MVI       0(PREG),C'*'    INDICA AO PROG. PRINCIPAL FIM ARQUIVO

FIMRG    STM       PIO,OVFL,SAVENT SALVA PIO E OVFL
         L         13,SAV13        RESTAURA REGIST. 13
         LM        14,12,12(13)    RESTAURA REGIST. 12 A 14
         BR        14              DESVIA PARA O PROGRAMA PRINCIPAL


***
*    AREAS / CONTANTES DE TRABALHO ***
***
```

```
ADDR     DC      A(DCOMP+4096)   ENDERECO PARA REGIST.BASE2
SAVENT   DC      2F'000'         AREA P/SALVAR PIO E OVFL
SAVSG    DS      5F
ENT1     DS      CL1024          AREA DE ENTRADA1 (IOAREA1)
ENT2     DS      CL1024          AREA DE ENTRADA2 (IOAREA2)
         LTORG
                 =C'$$BOPEN '
                 =C'$$BCLOSE'
                 =A(ARQE)
                 =F'127'
                 =F'1024'
                 =F'1'
                 =F'33'
                 =F'32'
                 =F'-7'
                 =H'768'
```

```
***
*    DTF DO ARQUIVO A SER COMPRESSO ***
***

ARQS     DTFMT  BLKSIZE=1024,
                DEVADDR=SYS011,
                FILABL=STD,
                IOAREA1=SAI1,
                IOAREA2=SAI2,
                IOREG=(10),
                RECFORM=FIXUNB,
                REWIND=UNLOAD,
                TYPEFLE=OUTPUT


*    INICIALIZA ROTINA DE COMPRESSAO = COMP ***
***
***

         USING  *,BASE,BASE2      INDICA REGISTRADOR BASE
COMP     STM    14,12,12(13)      SALVA REGISTRADORES DO PROG.PRINCIPAL
         LR     BASE,15           CARREGA REGISTRADOR BASE
         L      BASE2,ADR
         ST     13,SAV13          SALVA REGISTRADOR 13 EM SAV 13
         LM     PIO,OVFL,SAVSAI   RESTAURA REGISTRADORES PIO E OVFL
         L      PREG,0(,1)        CARREGA EM PREG ENDERECO DO PARAMETRO
         CLI    0(PREG),C'*'      VERIFICA SE FIM DE ARQUIVO
         BE     FIMSAI            DESVIA PARA FINALIZACOES


*    CARREGAMENTO DAS TABELAS ***
***
***

         BC     0,ROTSEG          SE NAO E 1-EXECUCAO DESVIAR ROTSEG
         OI     *-3,X'F0'         ALTERAR INST.ANTERIOR P/DESVIO INCOND.
         OPEN   ARQS              ABRIR ARQUIVO COMPRESSO
         LA     PTDS,TDS          CARREGAR END.INICIO TABELAS
         LA     W4,5              INICIALIZA REGISTRADOR W4 COM 5
LACO1C   LA     W2,4              INICIALIZA REGISTRADOR W2 COM 4
LACO2C   MVC    0(256,PIO),0(PTDS) TRANSFERE TABELAS
         LA     PIO,256(,PIO)     DA MEMORIA
         LA     PTDS,256(,PTDS)   PARA AREA DE
         BCT    W2,LACO2C         SAIDA
         PUT    ARQS              GRAVAR PROXIMO BLOCO
         BCT    W4,LACO1C         VERIFICA SE TODAS TABELAS GRAVADAS
```

```
***
*    INICIALIZA APONTADORES / REGISTRADORES ***
***

ROTSEG   LA      PTDS,TDS+3        CARREGA END.TAB.DEFINICAO DE SEGMENTOS
         LA      PTDC,TDC          CARREGA END.TAB.DEFINICAO DE CAMPOS
         LA      PPAD,PAD          CARREGA END. DE PADRAO DE AUSENCIA
         LA      NCMP,0            ZERA REGIST. NCMP


***
*    ROTINA DE PREPARACAO/TESTES DE SEGMENTOS ***
***

VOLTA1   LA      RBS,127           MARCA REGISTRADOR COM BYTE SENTINELA
VOLTA2   LA      RBS,1(,RBS)       INCREMENTA RBS
         LA      PSEGD,SEGD        CARREGA END.AREA PARA SEG.DESCOMPRESSO
         LA      PTDS,8(,PTDS)     POSICIONA PTDS NA PROX.ENTRADA TDS
         IC      NCMP,1(,PTDS)     COLOCA EM NCMP N.CAMPOS DO SEGMENTO
         C       NCMP,=F'0'        COMPARA NCMP COM ZERO
         BE      FIMREG            SE = DESVIA PARA FIMREG (FIM REGISTRO)

EXPAND   LH      W2,0(,PTDC)       COLOCA EM W2 POSICAO INICIO CAMPO REG.
         LA      W2,0(W2,PREG)     COLOCA EM W2 ENDERECO DESTA LOCALIZACAO
         LH      W1,002(,PTDC)     COLOCA EM W1 TAMANHO DO CAMPO
         STC     W1,*+5            COLOCA NA MVC TAMANHO DO CAMPO
         MVC     0(,PSEGD),0(W2)   MOVE CAMPO DO REGISTRO PARA SEGD
         LA      PTDC,4(,PTDC)     POSICIONA PTDC PROXIMO CAMPO
         LA      PSEGD,1(W1,PSEGD) POSICIONA PSEG PROXIMO CAMPO EM SEGD
         BCT     NCMP,EXPAND       SE N.CAMPOS NAO ZERO DESVIA EXPAND

         IC      NCMP,0(,PTDS)     COLOCA EM NCMP TAMANHO DO SEGMENTO
         STC     NCMP,*+5          COLOCA NA CLC TAMANHO DO SEGMENTO
         CLC     0(,PPAD),SEGD     COMPARA SEGMENTO DESCOMPRESSO C/PADRAO
         LA      PPAD,1(NCMP,PPAD) POSICIONA PPAD P/PROXIMO PADRAO
         BE      VOLTA2            SE = DESVIA PARA VOLTA2

         C       RBS,=F'128'       VERIFICA SE HA SEGMENTO AUSENTE
         BE      ROTHSF            SE = DESVIA PARA ROTHSF

         STC     RBS,0(,PIO)       COLOCA BYTE SENTINELA NA IOAREA
         LA      PIO,1(,PIO)       INCREMENTA PIO
         LA      OVFL,1(,OVFL)     INCREMENTA OVFL
         C       OVFL,=F'1024'     COMPARA OVFL COM TAMANHO IOAREA
         BNE     ROTHSF            SE DIFERENTES DESVIA PARA ROTHSF
         PUT     ARQS              SE = GRAVA BLOCO
         LA      OVFL,0                 ZERA OVFL
```

```
***
***
*    ROTINA DE HSF ***



***
*    INICIALIZACAO HSF ***
***

ROTHSF   STM      PTDS,OVFL,SAVSEG SALVA REGISTRADORES DA ROT. SEGMENT.
         LA       PSEGC,SEGC+1     CARREGA END.AREA PARA SEG.COMPRESSO
         LA       PSEGD,SEGD+1     CARREGA END.AREA PARA SEG.DESCOMPRESSO
         LA       BTRAD1,TRAD1     CARREGA END.TAB.TRADUTORA
         AH       BTRAD1,2(,PTDS)  POSICIONA BTRAD1 NO LOCAL DESEJADO
         L        CSAI,=F'-32'



***
*    VERIFICA FORMATO DOS SEGMENTOS ***
***

FORMAT   CLC    2(2,PTDS),=H'758' VERIFICA SE FORMATO SEG.E DEC.COMPACT
         BL       ALFA             SE ALFA/DEC.ZONADO DESVIA PARA ALFA
         BE       DECCOM           SE DEC.COMPACT. DESVIA P/ DECCOM
         LA       PSEGC,SEGD+1     SE BINARIO CARREGA EM PSEGC END.SEGD+1
         LA       NCMP,1(,NCMP)    COLOCA EM NCMP TAM.TOTAL SEGMENTO+1
         STC      NCMP,0(,PSEGC)   ARMAZENA TOT.ANTERIOR NO BYTE ANTES SEG
         B        FINAL1           DESVIA PARA FIM ROTINA HSF



***
*    DESCOMPACTA SEGMENTO ***
***

DECCOM   LA       W1,2(,NCMP)      INICIALIZA W1 COM TAM.SEG.COMPACTADO
         LR       W3,NCMP          SALVA EM W3 TAM.SEG.COMPACTADO
         LA       TMSEG,0
         L        NCMP,=F'-7'      CARREGA NCMP COM -7
VOLTA3   BXLE     W1,NCMP,FIMDES   SE W1 <= ZERO DESVIA FIM DESCOMPATACAO
         UNPK     0(15,PSEGC),1(8,PSEGD) DESCOMP.PARTES DE 8 BYTES DO SEG
         LA       PSEGC,14(,PSEGC) POSICIONA PSEGC PROX. PARTE A RECEBER
         LA       PSEGD,7(,PSEGD)  POSICIONA PSEGD PROX. PARTE A RETIRAR
         B        VOLTA3           DESVIA PARA VOLTA3
FIMDES   LA       W1,6(,W1)        COLOCA EM W1 TAM-1 DA PARTE FINAL
         LA       W2,0(W1,W1)      COLOCA EM W2 2*W1 - 1
         SLL      W2,4             SHIFT W2 4 BYTES A ESQUERDA
         LA       W2,0(W2,W1)      PREPARA BYTE DE TAMANHO DO UNPK
```

```
         STC     W2,**+5           ARMAZENA W2 NA INSTRUCAO UNPK
         UNPK    0(0,PSEGC),1(0,PSEGD) DESCOMPACTACAO
         LA      PSEGC,SEGD        CARREGA EM PSEGC ENDERECO SEGD
         LA      PSEGD,SEGC        CARREGA EM PSEGD ENDERECO SEGC+1
         LA      NCMP,1(W3,W3)     COLOCA EM NCMP TAM.SEG.DESCOMPAC


***
*    OBTER CODIGOS ***
***

ALFA     LA      CONTA,1           INICIALIZA CONTADOR TAMANHO SEGC
         LA      W2,CODS
         AH      W2,4(,PTDS)
         LR      BCODS,W2
         LA      TMSEG,1(NCMP,PSEGD) POSICIONA APONTADOR FIM SEGD
         LA      INSEGD,1          INICIALIZA REGISTRADOR PARA INCREMENT=1
         ST      PSEGC,WPSEGC      SALVO REGISTRADOR PSEGC
VOLTA5   BXH     PSEGD,INSEGD,FIMHSF VERIFICA FIM SEGD
         IC      W1,0(,PSEGD)      COLOCA CARATER EM W1
         IC      W1,0(BTRAD1,W1)   COLOCA LOCALIZACAO DO CODIGO EM W1
         L       ENT,0(W1,BCODS)   OBTEM DE TAB.CODS O CODIGO EM ENT
         LA      CENT,15           COLOCA MASCARA EM CENT
         NR      CENT,ENT          COLOCA EM CENT O N= DE SHIFTS
         AR      CSAI,CENT         ADICIONA A CSAI N= DE SHIFTS DO CODIGO
         BP      ARMAZ             SE RESULTADO POSITIVO = ESTOURO
         SLDL    SAI,0(CENT)       SHIFT DO CODIGO PARA SAI
         BM      VOLTA5            SE RESULTADO NEGATIVO OK
         ST       SAI,0(,PSEGC)    ARMAZENO  SAI EM PSEGC
         LA      CONTA,4(,CONTA)   ADICIONA 4 A CONTA TAMANHO
         LA      PSEGC,4(,PSEGC)   ADICIONA 4 A PSEGC
         L       CSAI,=F'32'       CARREGO EM CSAI =32
         B       VOLTA5            VOLTO A PROCESSAR
ARMAZ    SR      CENT,CSAI         OBTEM EM CENT O ESPACO VAGO EM SAI
         SLDL    SAI,0(CENT)       SHIFT DOS BITS POSSIVEIS
         ST      SAI,0(,PSEGC)     ARMAZENO SAI EM PSEGC
         SLDL    SAI,0(CSAI)       SHIFT DOS BITS RESTANTES
         A       CSAI,=F'-32'      POSICIONO CSAI COM ESPACO VAGO EM SAI
         LA      CONTA,4(,CONTA)   ADICIONA 4 A CONTA TAMANHO
         LA      PSEGC,4(,PSEGC)   ADICIONA 4 A PSEGC
         B       VOLTA5            VOLTO A PROCESSAR


***
*    FINAL COMPRESSAO HSF ***
***

FIMHSF   C       CSAI,=F'0'        VERIFICA SE CONTADOR DE BITS E ZERO
         BE      FINAL             SE = 0 DESVIO PARA FINAL
```

```
         LA       CSAI,32(,CSAI)  OBTER NUMERO BITS USADOS EM SAI
         LA       CENT,0(CSAI,CSAI) DUPLICAR VALOR EM CSAI
         LA       CSAI,0(CSAI,CENT) TRIPLICAR VALOR EM CSAI
         LA       CENT,MASC       CARREGAR EM CENT ENDERECO TAB.MASCARA
         IC       W1,2(CENT,CSAI) OBTER NUM. BITS P/ALINHAR EM BYTE
         SLL      SAI,0(W1)       ALINHAR SAI
         IC       W1,0(CSAI,CENT) OBTENHO MASCARA
         STC      W1,*+5          ARMAZENAR MASCARA INST.SEGUINTE
         STCM     SAI,0,0(PSEGC)  ARMAZENO EM SEGC OS BYTES COMPLETOS
         IC       W1,1(CSAI,CENT) OBTENHO NUM.BYTES COMPLETOS DE SAI
         LA       CONTA,0(CONTA,W1) ADICIONO A CONTA ESTE NUMERO
FINAL    A        CONTA,=F'-1'    SUBTRAIR 1 DE CONTA
         L        PSEGC,WPSEGC    RESTAURAR PSEGC
         A       PSEGC,=F'-1'
         STC     CONTA,0(,PSEGC)
         LR       NCMP,CONTA
FINALL   LM       PTDS,OVFL,SAVSEG RESTAURO REGS DA ROTINA DE SEGMENTACAO


***
*    ROTINA DE COLOCACAO DOS SEGMENTOS DA IOAREA ***
***

BLOCK    LA       OVFL,1(,OVFL)
         LA       W2,1024

         BXH      OVFL,NCMP,DIV   DESVIO P/ DIV SE IOAREA ESTOROU
RESTO    STC      NCMP,*+5        COLOCA NA MVC TAMANHO DO SEGMENTO
         MVC      0(,PIO),0(PSEGC) MOVE SEG.COMPRESSO DE SEGC P/ IOAREA
         LA       PIO,1(NCMP,PIO) POSICIONA APONTADOR IOAREA PROX.SEG.
         CR       OVFL,W2         COMPARO OVFL COM TAMANHO IOAREA
         BNE      VOLTA1          DESVIO PARA VOLTA1 SE DIFERENTE
         PUT      ARQS            SE = GRAVAR BLOCO
         LA       OVFL,0               ZERAR OVFL
         B        VOLTA1               DESVIA PARA VOLTA1

DIV      SR       OVFL,W2         DETERMINA DE QUANTO E O OVERFLOW
         SR       NCMP,OVFL       DETERMINA QUANTO DO SEG.FICA NA IOAREA
         STC      NCMP,*+5        COLOCA NA MVC TAMANHO DA PARTE IOAREA
         MVC      0(,PIO),0(PSEGC) MOVE PARTE DO SEG.EM SEGC PARA IOAREA
         PUT      ARQS            GRAVA BLOCO
         LA       PSEGC,1(NCMP,PSEGC) POSICIONA PSEGC PARA OBTER RESTANTE
         LR       NCMP,OVFL       COLOCA EM NCMP EXCEDENTE
         S        NCMP,=F'1'      DECREMENTA NCMP
         B        RESTO           DESVIA PARA RESTO


***
*    ROTINA DE FINALIZACOES ***
```

```
***

FIMSAI   MVI     C(PIO),X'00'     GRAVA SINAL FIM DE ARQUIVO NO IOAREA
         PUT     ARQS             GRAVA BLOCO
         CLOSE   ARQS             FECHA ARQS
         B       FIMREG1          DESVIA PARA FIMREG1

FIMREG   C       RBS,=F'128'      VERIFICA SE HA SEGMENTOS AUSENTES
         BE      FIMREG1          SE NAO HA DESVIA PARA FIMREG1
         STC     RBS,0(,PIO)      GRAVA BYTE SENTINELA NA IOAREA
         LA      PIO,1(,PIO)      INCREMENTA PIO
         LA      OVFL,1(,OVFL)    INCREMENTA OVFL
         C       OVFL,=F'1024'    COMPARO OVFL COM TAMANHO IOAREA
         BNE     FIMREG1          DESVIO PARA FIMREG1 SE DIFERENTE
         PUT     ARQS             GRVA BLOCO
         LA      OVFL,0           ZERA OVFL

FIMREG1  STM     PIO,OVFL,SAVSAI  SALVA PIO E OVFL
         L       13,SAV13         RESTAURA REGIST.13
         LM      14,12,12(13)     RESTAURA REGIST. 12 A 14
         BR      14               DESVIA PARA O PROGRAMA PRINCIPAL
         EOJ


***
*    AREAS / CONSTANTES DE TRABALHO ***
***

         LTORG
                 =C'$$BOPEN '
                 =C'$$BCLOSE'
                 =A(ARQS)
                 =F'0'
                 =F'128'
                 =F'1024'
                 =F'-32'
                 =F'-7'
                 =F'-1'
                 =F'1'
                 =H'768'

ADR      DC       A(COMP+4096)    ENDERECO REGISTRADOR BASE2
WTESTE   DS       F
WPSEGC    DS      F               AREA SALVAMENTO PSEGC
SAVSEG   DS       7F              AREA SALVAMENTO REGS PRA ROT.HSF
SAV13    DS       F               AREA DE SALVAMENTO REGIST. 13
SAVSAI   DC       2F'000'         AREA DE SALVAMENTO DE PIO/OVFL
         DS       CL1
SEGD     DS       CL127           AREA DE SEGMENTO DESCOMPRESSO
SEGC     DS       CL128           AREA DE SEGMENTO COMPRESSO
```

```
TDS      DS      CL512           TABELA DE SEGMENTOS
TDC      DS      CL512           TABELA DE CAMPOS
PAD      DS      CL512           PADRAO DE AUSENCIA
UNS      DS      CL512           TABELA DE PREFIXOS
CRC      DS      CL1024          TABELA DE CARACTERES
CODS     DS      CL2048          TABELA DE CODIGOS
TRAD1    DC      74XL01'00'      TABEL DE TRADUCAO
         DC        XL37'040800C1D1418'
         DC      10XL01'00'
         DC        XL07'1C20242800203D'
         DC      08XL01'00'
         DC        XL04'34383C40'
         DC      12XL01'00'
         DC        XL06'44484C505453'
         DC      65XL01'00'
         DC        XL39'5C6D64685C7074787C'
         DC       7XL01'00'
         DC        XL09'808488C9094989CA0'
         DC       8XL01'00'
         DC        XL08'A4A8ACB0B4B8BCC0'
         DC       6XL01'00'
         DC        XL10'C4C8CCD0D4D8DCE0E4E8'
         DC       6XL01'00'
         DC      193XL01'00'
         DC        XL09'040800C101413C2024'
         DC       7XL01'00'
         DC        XL09'282C3034383C404448'
         DC       8XL01'00'
         DC        XL08'4C5054585C6064'
         DC      22XL01'00'
         DC      192XL01'50'
         DC        XL10'0004080C101413C2024'
         DC       6XL01'50'
         DC        XL10'282C3034383C4044484C'
         DC      23XL01'50'
         DC        XL09'54585C6064686C7074'
         DC       6XL01'50'
         DC      241XL01'00'
         DC        XL15'040800C101413C2024000282C0030'
MASC     DC       XL20'00000081010781010631D1058101048101038101'
         DC       XL20'028101018101006302078302068302C583020483'
         DC       XL20'02038302028302018302003703078703068703D5'
         DC       XL20'87030487030387030287030187030C8F04078F04'
         DC       XL16'068F04058F04048F04038F04028F04C1'
SAI1     DS       CL1024          AREA DE SAIDA1 (IOAREA1)
SAI2     DS       CL1024          AREA DE SAIDA2 (IOAREA2)
         END
```

REFERÊNCIAS BIBLIOGRÁFICAS

1. Bremer, R.W., Do it by the numbers - digital shorthand, Communication of the ACM, vol.3 nº 10, pp: 530-536, oct. 1960.

2. Connel, J. Brian, A Huffman-Shannon-Fano code, Proceeding of the IEEE, pp: 1046-1047, july 1973.

3. Gilbert, E.N. & Moore, E.F., Variable - length binary encodings, The Bell System Technical Journal, vol. 38, pp: 933-967, july 1959.

4. Houston, George B., Generalized data compression, Boeign Computer Services, Inc, Seattle, Washington, feb. 1977.

5. Houston, George B., Aplication of data compaction to computer networks, Boeing Computer Services, Inc, Seattle, Washington.

6. Huffman, D.A., Method for construction of minimun - redundancy codes, Proceedings of the I.R.E., vol. 40 nº 9, pp: 1093-1101, sept. 1952.

7. Lavelle, Pierre J. & Cunha, Caio M., Data compression through microprocessors for data transmission, Communication of the ACM, feb. 1975.

8. Marron, B.A. & Maine P.A.D., Automatic data compression, Communication of the ACM, vol. 10 nº 11, pp: 711-715, nov. 1967.

9. Martin James, Computer Data-Base Organization, Englewood Cliffs, N. Jersey, Prentice-Hall, p: 558, 1975.

10. Pietracci, Luciano, M.Sc., Compressão de Dados, Tese, COPPE-Universidade Federal do Rio de Janeiro, outubro 1973.

11. Shannon, C.E., A mathematical theory of communication, The Bell System Thechnical Journal, vol. 37, pp: 373-423, july 1948.

12. Snyderman, Martin & Hunt, Bernard, The myriad virtues of text compaction, Datamation, pp: 36-40, dec. 1970.

13. Stephen, S. R. & Paul, J. Kreutzer, Data compression for large business files, Datamation, pp: 62-66, sept. 1972.

14. Wells, M., File compression using variable length encodings, The Computer Journal, vol. 15 nº 4, pp: 308-313.