

BR 780

1501



IMPLEMENTAÇÃO DO SISTEMA ERP NO COMPUTADOR

IBM 1620 - MOD. 2 DE 40 K

Cíbar Cáceres Aguilera

DISSERTAÇÃO E TESE - IEA 006

FEVEREIRO/1976

BR780150L

DISSERTAÇÃO E TESE - IEA 006

FEVEREIRO/1976

**IMPLEMENTAÇÃO DO SISTEMA ERP NO COMPUTADOR
IBM 1620 – MOD. 2 DE 40 K**

Cíbar Cáceres Aguilera

Dissertação para obtenção do Título de "Mestre em Engenharia" – Orientador Prof. Ivan de Queiroz Barros. Apresentada e defendida em 17 de junho de 1970, na Escola Politécnica da Universidade de São Paulo.

APROVADA PARA PUBLICAÇÃO EM JANEIRO/1976.

CONSELHO DELIBERATIVO

MEMBROS

Klaus Reinach — Presidente
Roberto D'Utra Vaz
Helcio Modesto da Costa
Ivano Humbert Marchesi
Admar Cervellini

PARTICIPANTES

Regina Elisabete Azevedo Baretta
Flávio Gori

SUPERINTENDENTE

Rômulo Ribeiro Pieroni

INSTITUTO DE ENERGIA ATÔMICA
Caixa Postal 11.049 (Pinheiros)
Cidade Universitária "Armando de Gallet Oliveira"
SÃO PAULO — BRASIL

NOTA: Este trabalho foi conferido pelo autor depois de composto e sua redação está conforme o original, sem qualquer correção ou mudança.

ÍNDICE

	Pág.
INTRODUÇÃO	
CAPÍTULO I – Transcrição de Partes do ERP Original	4
Características do ERP Original	4
Manual de Referências	6
Modos do Sistema	6
Formato das Instruções	7
Comprimento das Instruções	8
Instruções Continuação	8
Organização da Memória	9
Tabelas e Áreas	9
Programas	12
Organização do Disco	13
Comandos do Usuário	15
Área de Parâmetros	17
CAPÍTULO II – Nova Distribuição da Área de Memórias e do Disco	18
Trabalho de Adaptação	18
Facilidades Disponíveis no Computador IBM 1620 Mod 2 de 40 K	19
Verificação da Compatibilidade	19
Análise das Limitações Apresentadas pela Versão Original	19
Áreas de Programas	20
Áreas e Tabelas	21
Organização do Disco	23
Adaptação dos Subprogramas do Sistema	24
CAPÍTULO III – Comandos do Sistema Implementados	25
Comando *MANIPULE	27
Descrição do ERP5	28
Comando *CALCULE	29
Operadores, Símbolos e Funções	30
Função .LOG	30
Função .LN	30
Função .DX	30
Função .EXP	30

	Pág.
Função RQ	30
Função ATAN	30
Função SEN	30
Função COS	30
Função TAN	30
Função ABS	30
Função **	30
Descrição do Método Utilizado	34
Conversão da Notação Normal em Polonesa	36
Aplicação do T-CALC	37
Descrição dos Subprogramas ERP 6 e ERP 7	39
Diagrama Simplificado ERP 5	42
Diagrama Simplificado ERP 6	44
Diagrama Simplificado ERP 7	45
CAPÍTULO IV – Comandos do Usuário Implementados	46
Comando SEJA (ERP 8)	46
Comando TRAP (ERP 9)	49
Comando LOCR (ERP 10)	50
Comando PLLOT (ERP 11)	54
Comando MATRIC (ERP 12)	54
Comando SOLUÇÃO (ERP 13)	55
Comando AJUSTE (ERP 14)	56
CAPÍTULO V – Exemplos de Aplicações	58
CAPÍTULO VI – Considerações Finais	94
Anexo A – Organização da Memória – Programas	97
Anexo B – Tabelas e Áreas	98
Anexo C – Organização da Área de Trabalho do Disco	99
Anexo D – Área de Comunicação	100
Anexo E – Usos do IND4	102
Anexo 1 – Distribuição da Memória – Área de Programas	105
Anexo 2 – Distribuição da Memória – Área de Tabelas	106
Anexo 3 – Distribuição das Áreas do Disco	107
Anexo 4 – Conteúdo do X'in	108
Anexo 5 – Descrição Formal dos Comandos do Sistema	109
Descrição Formal dos Comandos do Usuário	112
Cartão Intervenção do Operador	113
Anexo 6 – Relação das Mensagens de Erro	114
Referências Bibliográficas	119

IMPLEMENTAÇÃO DO SISTEMA ERP NO COMPUTADOR

IBM 1620 – MOD. 2 DE 40 K

Cíbar Cáceres Aguilera

INTRODUÇÃO

O INSTITUTO DE ENERGIA ATÔMICA de São Paulo, que é um centro de Pesquisas Nucleares, dispõe de um Computador IBM 1620, mod. 2, que se encontra à disposição dos pesquisadores, funcionários do Instituto de Energia Atômica, para a realização dos cálculos de que necessitam para as pesquisas em andamento.

Num momento dado da pesquisa, o pesquisador sente a necessidade de realizar alguns cálculos, ainda não definitivos. Dedicar-se-á, então, a preparar um modelo matemático, também não definitivo que lhe proporcione os resultados que lhe permitirão conhecer até que ponto está seguindo o caminho certo.

O pesquisador que tem experiência no uso do Computador, tratará de aplicá-lo na solução do seu modelo, mas sabe que deve percorrer várias etapas até chegar ao resultado: elaboração do programa, compilação, correção de erros formais, recompilação, correção dos erros lógicos, testes numéricos, etc.

Todos esses passos requerem um investimento de precioso tempo e esforço do pesquisador. Encontra-se este ainda na situação de ter que realizar um programa perfeito, mais ou menos sofisticado para uma pesquisa ainda com alguma incerteza no caminho escolhido e modelos idealizados

Seria muito conveniente, para ele, dispor de um sistema no qual seja possível:

- a) escrever instruções que sejam imediatamente executadas sem necessidade de compilação.
- b) descrever seus modelos mediante macro-instruções.
- c) fazer as correções dos erros à medida que são encontrados sem perder o trabalho já executado.
- d) controlar a execução e interferir de forma não programada.
- e) verificar as diversas alternativas, que porventura se apresentem, sem necessidade de retornar ao princípio.
- f) interromper a execução em qualquer instante e retomá-la após transcorrido qualquer lapso de tempo, no ponto em que parou, sem ter perdido nada do que havia sido feito até esse momento.

Em última análise, um sistema em que seja possível comunicar-se com a máquina numa forma similar àquela em que se comunica com outro pesquisador.

Este problema induziu-nos a procurar uma solução adequada. Essa solução já existia para uma máquina do mesmo tipo, mas, de menor capacidade que a disponível no Instituto de Energia Atômica,

mediante o ERP UM SISTEMA PARA INTERAÇÃO HOMEM COMPUTADOR, feito pelo Prof. IVAN DE QUEIRÓZ BARROS⁽²⁾, da Escola Politécnica da Universidade de São Paulo.

Decidimos portanto implementá-lo para o Computador IBM 1620 mod. 2 com 40.000 posições de memória. Para realizar isto foi necessário redefinir áreas, modificar os programas que já formavam parte do sistema e criar outros. Com isto o sistema torna-se mais flexível, com um melhor aproveitamento das facilidades disponíveis no Computador.

O sistema ERP original foi preparado para um Computador de 20.000 posições de memória e como tal, as diferentes áreas utilizadas eram relativamente pequenas; conseqüentemente os programas destinados à execução de um comando, na maioria dos casos, necessitavam ser divididos em duas ou mais partes. Isto acarretava uma velocidade de operação reduzida, devido ao tempo que se perde para trazer do disco um programa a ser processado na memória de núcleos.

Com a nova capacidade foi possível dispor de mais áreas que permitissem deixar residentes na memória os programas de uso mais frequente, e reunir as partes de um programa dividido até completar a capacidade disponível. Tudo isto, unido a um menor ciclo de memória que permite uma execução mais rápida, e mais a possibilidade de saída de dados pela impressora, aumenta sensivelmente a eficiência do sistema.

Salientamos que a idéia de criar sistemas conversacionais não é nova. Ela nasceu paralelamente à idéia de computação em partição de tempo, e multiprogramação aplicada em computadores de grande porte e que podem atender a vários usuários que dispõem de terminais remotos.

Um desses sistemas foi desenvolvido no MIT para um computador IBM 360 com 65 Kbytes de memória e duas unidades de discos magnéticos e com facilidade de entrada e saída por cartões. É o sistema conhecido com o nome de ICES (Integrated Civil Engineering System)⁽¹⁵⁾. Posteriormente foi implementado no modelo 40 da IBM 360, do Civil Engineering System Laboratory, com dispositivos gráficos de entrada e saída e facilidades de computação remota.

O sistema ICES é um sistema desenvolvido para resolver problemas de Engenharia Civil, dispondo, basicamente de uma linguagem orientada para Engenheiros possibilitando solução de problemas que comportam mais de uma disciplina da engenharia. Possui uma estrutura modular ou de blocos, chamados subsistemas, com um esquema de organização de dados que possibilita a transferência dos mesmos de um sistema à outro.

Cada subsistema corresponde a um comando que é identificado em geral por um termo técnico.

Inclui uma linguagem de programação ICETRAM que permite criar os subsistemas, e um sistema operacional que coordena e supervisiona a ação desses subsistemas.

A linguagem ICETRAM é uma extensão do FORTRAN comportando todas as instruções desta linguagem mais alguns recursos adicionais, sendo o principal a alocação dinâmica de memória.

No mesmo MIT foi desenvolvido o sistema OPS-3⁽⁵⁾ (On-Line computation and simulation) para um Computador de grande porte.

OPS-3 dispõe de facilidades para computação "On-Line", programação e construção de modelos. É um sistema aberto e modular, sendo portanto possível diminuí-lo ou aumentá-lo de acordo com as necessidades. O sistema é compatível com várias linguagens de programação.

Basicamente possui uma estrutura em blocos, sendo cada bloco formado por um subprograma pré-compilado e conhecido como **operador**.

Um operador executa uma determinada função, especificada pelo seu nome. Cada operador possui um conjunto de parâmetros que identificam os dados a serem processados.

Um conjunto de operadores relacionados, constitui um operador composto que é uma verdadeira sub-rotina do sistema.

Um operador pode ser executado imediatamente, ou armazenado para posterior execução, ou armazenado e ao mesmo tempo executado.

Neste sistema foi inspirado o nosso sistema ERP.

De acordo com as fontes de informações a nosso alcance, podemos referenciar vários outros sistemas, praticamente baseados nos mesmos princípios dos já mencionados, mas na sua maioria desenvolvidos para computadores de grande porte.

Estes são: (20) e (21)

– OMNITAB é um programa interpretativo para IBM 7090/7094 que permite utilização direta da máquina por pesquisadores sem conhecimentos prévios de programação.

Basicamente está constituído por instruções dadas em forma de sentenças em inglês, que controlam o fluxo do cálculo em uma forma muito análoga à lógica que se segue quando se executa um cálculo numa máquina de calcular. Também simula tabelas matemáticas e folhas de trabalhos multi-colunas.

Cada sentença é perfurada num cartão. O programa simula uma folha de trabalho de 46 colunas e 101 linhas ficando os argumentos em colunas específicas. As operações são realizadas nestas colunas e os resultados são armazenados nas colunas previamente designadas.

OMNITAB é um sistema que contém aproximadamente 100 subrotinas e dispõe de mais de 100 comandos.

– O sistema JOSS (Jhoniac Open-Shop System) desenvolvido pela RAND CORPORATION, para computo numérico através de máquina de escrever em localizações remotas. Foi desenvolvido para o computador JHONIAC construído em Princeton no ano 1953.

Utiliza uma linguagem muito próxima à linguagem natural da álgebra. Realiza as operações elementares com números.

– O sistema CULLER-FRIEND que além de tratar com cálculos elementares, manipula funções. Estas funções são programadas como sub-programas e representadas nas teclas de uma máquina de escrever por um símbolo especial, existindo uma correspondência direta entre a tecla e a função. Basta apertar a tecla para que o programa correspondente à função entre em ação. Os parâmetros necessários são representados por símbolos alfanuméricos que podem ser constantes; ou funções que podem ser quantidades aritméticas, vetores reais ou complexos, ou matrizes.

– O sistema FFL (First Financial Language) que realiza cálculos relacionados com a análise financeira utilizando uma linguagem muito similar à linguagem financeira.

Basicamente é formado por um conjunto de subprogramas relacionados com a solução de problemas estatísticos e funções especiais, que entram em ação quando o usuário introduz pela máquina de escrever o nome do subprograma requerido seguindo à palavra CALL. Cada comando possui várias opções que são especificadas pelo usuário uma vez que o comando esteja em operação.

– Além desses sistemas citados existem outros sistemas conversacionais para computadores grandes, tais como o CAI (Computer Assisted Instruction) utilizado na Universidade da Califórnia, o sistema SYNTEX, o sistema DEACON, etc. Todos possuem uma estrutura similar, i.é. o usuário escreve numa linguagem próxima ao natural e o sistema a interpreta e põe em ação o adequado subprograma previamente compilado, para dar a resposta conveniente.

— Para computadores de porte médio existe uma linguagem desenvolvida no Departamento de Cálculo Eletrônico de Saclay, França, aplicada ao computador CAE 90-40. O sistema chama-se SPRINT/1 e tem por finalidade facilitar a utilização do computador, permitindo ao usuário se exprimir numa linguagem muito próxima ao natural, e está destinado à avaliação de fórmulas matemáticas. A conversação se estabelece através de uma máquina de escrever.

A linguagem é formada por um Tradutor, um Interpretador Global e um Modificador de parâmetros que, atuando em conjunto, executam a instrução dada pelo usuário⁽³⁾.

Para computador de pequeno porte a única linguagem que encontramos em uso atualmente é o APL para o computador IBM 1130.

Salientamos que o ERP não constitui, como a maioria dos citados, uma linguagem; é um sistema que serve de suporte à linguagem que o próprio usuário cria de acordo com sua necessidade.

Encontramos assim que em todos os campos de atividade humana realizam-se esforços para o desenvolvimento de sistemas ou linguagens de interação homem-computador.

Esperamos que o esforço realizado na implementação do Sistema ERP possa servir de ajuda eficaz ao pesquisador. O sistema é simples, flexível e fácil de se utilizar.

CAPÍTULO I

TRANSCRIÇÃO DE PARTES DO ERP ORIGINAL

Para facilitar a compreensão do leitor, das modificações introduzidas e a implementação realizada, transcrevemos neste capítulo, literalmente, as características do sistema ERP e uma parte do manual de referência da versão original.

1.1 — Características do Sistema ERP

1.1.1 — No sistema ERP pode-se optar pela execução das instruções à medida que estas vão sendo fornecidas, através da máquina de escrever ou cartões, ou pelo processamento automático de um programa previamente registrado.

O nome do sistema é formado pelas iniciais dos modos (EXECUTE, REGISTRE, PROCESSE) nos quais este opera.

Qualquer combinação de execução imediata, registro de instruções e processamento de instruções registradas é possível.

1.1.2 — O sistema ERP é interpretativo

Isto significa que as instruções não são compiladas produzindo um programa objeto, mas são analisadas por uma rotina do sistema (INTERPRETADOR) que reconhece o comando e chama a subrotina apropriada transmitindo também a esta última os parâmetros presentes na instrução.

Isto permite a execução ou processamento imediatos de uma instrução ou programa registrado.

1.1.3 – O sistema ERP é modular e aberto.

Os comandos são executados por subrotinas pré-compiladas ou pré-montadas (módulos) de tamanho limitado construídas segundo um mesmo modelo e chamadas para execução de modo uniforme. Um número indefinido de módulos pode ser incorporado ao sistema, isto é, o sistema é aberto.

1.1.4 – O sistema ERP é adaptável

De fato, o usuário pode acrescentar ao sistema um comando cuja especificação tenha sido totalmente dada por ele, com apenas pequenas restrições de formato. Em outras palavras, o usuário pode construir uma linguagem adaptada ao tipo particular de problemas, dentro de sua especialidade.

Estes verdadeiros sub-sistemas não são estanques mas podem ser considerados como partes de um único estoque de comandos.

1.1.5 – O sistema ERP permite a interação entre o operador e o sistema.

Esta intercomunicação pode assumir os seguintes aspectos:

1.1.5.1 – Ao detectar qualquer erro, o sistema comunica ao operador o seu diagnóstico bem como todas as informações relevantes e passa a ele o controle para a correção.

1.1.5.2 – A qualquer instante o operador pode chamar a si o controle e/ou interferir ativamente alterando o estado do sistema ou pedindo informações sobre o mesmo. Qualquer informação solicitada não prejudica o processamento interrompido que poderá ser reiniciado no ponto onde parou.

1.1.6 – O sistema ERP possibilita ao operador corrigir erros durante a execução.

Isto é possível na quase totalidade dos erros detectados pelo sistema (cerca de 40) sem perda dos cálculos já efetuados até o momento.

1.1.7 – O sistema ERP facilita a programação

As instruções incorporadas ao sistema pelos usuários podem representar operações complexas (macro-instruções), possibilitando vencer grandes etapas por instrução. Isto reduz a ocorrência de erros e reduz o tempo de programação.

1.1.8 – O sistema ERP facilita a depuração do programa.

Entendemos por depuração do programa o conjunto de providências destinadas a eliminar erros formais e lógicos(*).

Ela é facilitada por estar o sistema ERP dotado de recursos análogos aos que se acham a disposição do programador quando utilizando linguagem de máquina.

São exemplos a possibilidade de inspecionar qualquer campo significativo da memória e a possibilidade de processar-se um programa registrado, instrução por instrução.

(*) Corresponde ao termo "debug" usado pelos americanos

1.1.9 – No sistema ERP a atribuição de espaço na medida é dinâmica.

Isto significa que uma determinada variável vetor ou matriz, não ocupa uma posição fixa na memória. Caso haja necessidade de espaço pode ser temporariamente retirada da memória e arquivada no disco. Quando trazida de volta ocupará provavelmente posição diferente.

O espaço ocupado é o estritamente necessário. No FORTRAN por exemplo é preciso reservar espaço de acordo com a previsão mais desfavorável.

1.1.10 – O sistema ERP permite arquivar cópias (FOTOS) do estado do sistema em momentos escolhidos e reproduzi-los posteriormente.

A utilização deste recurso pode ser útil em diversas situações:

1.1.10.1 – Numa verificação de um algoritmo no qual diversas alternativas devem ser experimentadas em diversos pontos, convém arquivar FOTOS do sistema tirados nas bifurcações. A volta a qualquer desses pontos pode ser feita reproduzindo-se o estado em que então se achava o sistema (ativando-se a FOTO).

Estas FOTOS são numeradas para identificação.

1.1.10.2 – Suponhamos um algoritmo em que tenham que ser tomadas decisões baseadas em critérios heurísticos e sujeitas a arrependimento conforme resultados posteriores. FOTOS do sistema nos momentos de decisão podem possibilitar a reforma destas e isto pode ser feito automaticamente num programa registrado.

1.1.10.3 – O recurso de arquivamento e ativação de FOTOS permite interromper um processamento, executar nova série de instruções não diretamente relacionadas com as anteriores, e continuar o processamento interrompido no ponto em que parou.

1.1.10.4 – Permite também interromper um processamento e reiniciá-lo em outra data.

1.1.11 – O sistema ERP foi feito para operar preferencialmente com blocos de dados, isto é, vetores e matrizes. De fato, todas as variáveis são consideradas pelo sistema como matrizes.

1.2 – Manual de Referência

1.2.1 – Modos do Sistema.

1.2.1.1 – Quando em operação o sistema encontra-se em um de três modos: EXECUTE (abreviado por EX) REGISTRE (abreviado por RG) ou PROCESSE (abreviado por PR).

1.2.1.2 – MODO EX: neste modo as instruções lidas são imediatamente interpretadas e executadas. Lida nova instrução a anterior é destruída.

1.2.1.3 – MODO RG: neste modo, as instruções lidas não são interpretadas ou executadas mas registradas na memória numa tabela de instruções.

1.2.1.4 – MODO PR: neste modo as instruções registradas na tabela de instruções vão sendo sequencialmente interpretadas e executadas, de forma automática. A próxima instrução a ser executada pode não ser a seguinte se uma opção de desvio for encontrada na instrução presente.

1.3 – Formato das Instruções

1.3.1 – Uma instrução é formada por: IDENTIFICAÇÃO, COMANDO e PARÂMETROS

1.3.1.1 – **IDENTIFICAÇÃO**: as duas primeiras posições da instrução são interpretadas pelo sistema como identificação da instrução. No modo EXECUTE podem estar em branco pois não são utilizadas. Toda instrução registrada porém deve ter uma identificação formada por dois caracteres quaisquer, sendo pelo menos um deles não branco. As instruções são registradas pelo sistema respeitando automaticamente o ordenamento crescente das identificações.

Os caracteres(*) empregados no IBM 1620 em ordem crescente são: b) + \$ * / , (= 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 2 3 4 5 6 7 8 9.

1.3.1.2 – **COMANDO**: é formado por uma seqüência de caracteres quaisquer (exceto: b _), em número qualquer começando não antes da terceira posição. Toda instrução contém obrigatoriamente um comando.

O comando identifica a subrotina ou grupo de subrotinas que executarão a instrução.

Apesar do número de caracteres num comando ser arbitrário, apenas os primeiros quatro são significativos, podendo portanto o comando ser abreviado. Os caracteres que faltarem para completar 4 caracteres serão considerados definidos implicitamente como brancos.

Um comando pode ser escrito começando em qualquer posição da terceira em diante.

O primeiro caráter diferente de b encontrado a partir da terceira posição não poderá ser _ ou _ e será interpretado como primeiro caráter do comando.

1.3.1.3 – **PARÂMETROS**: os parâmetros vêm após o comando e podem ser de três naturezas.

1.3.1.3.1 – **LITERAIS**: são caracterizados por virem imediatamente precedidos pelo caráter _ .

São formados por uma seqüência de caracteres em número qualquer, excetuados b _ .

Apenas os dois primeiros caracteres são significativos. Os caracteres que faltarem para completar dois, são implicitamente definidos como brancos. Por exemplo os três literais do grupo 3 ALFA são interpretados respectivamente com bb 3b e AL.

Os literais se distinguem dos outros tipos de parâmetros por não sofrerem decodificação. São usados pelas subrotinas chamadas pelos comandos, como se apresentam.

1.3.1.3.2 – **VARIÁVEIS**: são caracterizadas por virem imediatamente precedidas pelo caráter b e por ser seu primeiro caráter não numérico. São formadas por uma seqüência de caracteres excetuados b _ . Apenas os dois primeiros são significativos. Se faltar um caráter para completar dois, será considerado definido implicitamente como branco.

1.3.1.3.3 – **NÚMEROS**: são caracterizados por virem imediatamente precedidos pelo caráter b e por ser seu primeiro caráter numérico. São formados por uma seqüência de caracteres numéricos. Apenas os quatro primeiros são considerados. Se faltarem caracteres numéricos para completar quatro, zeros a esquerda serão considerados implicitamente definidos.

(*) Indicamos por b o caráter em branco.

1.3.2 – COMPRIMENTO DA INSTRUÇÃO

No modo EX uma instrução pode ter no máximo 80 caracteres. Se a instrução entra por cartão, ela termina com uma marca de registro ⚡ ou na falta desta na coluna 80. Se a entrada é por máquina de escrever, termina com ⚡ ou na falta desta, com o último caráter batido. Após bater a instrução deve-se usar a tecla RELEASE/START.

Após uma marca de registro ⚡ as posições seguintes até a de nº 80 podem ser utilizadas para comentários.

No modo RG uma instrução pode ter no máximo 29 caracteres. Se a entrada é por cartão uma marca de registro é obrigatória na coluna 30. Após uma marca de registro em qualquer caso, as posições seguintes até a de número 80 podem ser usadas para comentário.

1.3.3 – INSTRUÇÕES CONTINUAÇÃO

Se uma instrução excede o número máximo de caracteres permitido ela pode ser truncada escrevendo-se o caráter _ após o último parâmetro que cabe na instrução e continuada escrevendo-se os parâmetros restantes numa nova instrução continuação.

Esta instrução continuação é caracterizada por ter no lugar do comando os caracteres // .

Uma instrução pode ser seguida de um número qualquer de instruções continuação. Uma instrução continuação que por sua vez é continuada deve terminar também por _ .

1.3.4 – NÚMERO DE PARÂMETROS

O número de parâmetros de uma instrução incluindo suas continuações não pode exceder 24.

1.3.5 – CARACTERES EM BRANCO

Respeitadas as restrições já vistas quanto ao uso de caracteres em branco, estes podem ser removidos ou acrescentados a vontade.

1.3.6 – EXEMPLOS DE INSTRUÇÕES(*)

Exemplo I

```
03 *CTL .SE A 2 .NÃO MAIOR B 3 4 .ALTERE .MOD . RG
Se o modo for RG esta instrução pode ser fracionada como segue:
03 *CTL .SE A 2 .NÃO .MAIOR B 3 , ⚡
04 // 4 .ALT .MOD . PG
```

Exemplo II

```
*ADMINISTRE .SUBROTINA 10.
Neste exemplo o último literal é um literal em branco. A instrução acima poderia ser
abreviada como:
*ADM.SU 10. ⚡ COMENT INSTR ESCRITA ABREVIADA
```

(*) Nestes exemplos o caráter _ faz parte dos comandos.

Exemplo III

05 *ADM.DEFINA .A .B 2 .C .D , ± DEFINIÇÃO DE VARIÁVEIS,
 06 // .H 10 3 .F 5 .DELTA , ± VETORES E MATRIZES
 07 // .RAIO .MATRIZ 2 20.

1.4 – ORGANIZAÇÃO DA MEMÓRIA – TABELAS E ÁREAS

1.4.1 – A presente versão do ERP é destinada ao computador IBM-1620 com 20K de memória (20.000 posições de memória).

A memória está dividida, como se pode observar nos esquemas 1 e 2 (ANEXOS A e B), em duas partes. A primeira ocupando aproximadamente o primeiro terço ocupada por programas. A segunda formada por áreas de dados, áreas de trabalho e tabelas, ocupando os restantes dois terços.

Passaremos a descrever as subdivisões da segunda parte (vide esquema 2 no ANEXO B).

1.4.2 – ÁREA S (SIGLA AS): esta é uma área auxiliar de 8 posições^(*) para onde são transmitidos, um de cada vez, os diversos elementos de uma instrução lida na área de leitura, para serem decodificados.

1.4.3 – ÁREA DE LEITURA (SIGLA AL): nesta área com 162 posições são lidas as instruções via máquina de escrever ou cartão. A sua capacidade é de 80 caracteres sendo as duas últimas posições ocupadas por 0 ±.

1.4.3 – ÁREA DE PARÂMETROS (SIGLA AP): é uma área formada por 50 campos de 4 posições. Nestes campos são colocados normalizados^(**) o comando e os parâmetros da instrução, já interpretados. As subrotinas que executam os comandos consultam esta área para obtenção dos parâmetros.

1.4.5 – ÁREA DE COMUNICAÇÕES (SIGLA AK): é formada por 28 campos de 4 posições. Estes campos contém informações que o sistema consulta e que são por sua vez alterados e atualizados pelo sistema. Uma relação completa do conteúdo dos 28 campos encontra-se no ANEXO D.

1.4.6 – ÁREA DE PARÂMETROS TRANSMITIDOS (SIGLA PT): formada por 10 campos de 4 posições.

Nestes campos são colocados os parâmetros transmitidos para uma subrotina construída com instruções incorporadas ao sistema ERP^(***).

1.4.7 – TABELA DE INSTRUÇÕES (SIGLA TI): formada por 24 campos de 60 posições. Cada um destes campos chamaremos de linha. Teremos portanto 24 linhas, cada uma comportando o registro de uma instrução (até 29 caracteres mais marca de registro).

1.4.8 – PÁGINA: é formada pelos cinco últimos campos da área de comunicações, pela área de parâmetros transmitidos e pela tabela de instruções. Nos últimos cinco campos de AK temos:

INDTI (indicador da tabela de instruções) contém o número da próxima linha vaga da tabela de instruções.

NPG contém o número da página.

(*) Recordemos que no IBM-1620 cada caráter é representado na memória por 2 dígitos.

(**) No § 8 encontra-se uma descrição detalhada da área de parâmetros. § 8 = 1 7.2.5.

(***) Não confundir com as subrotinas pré-copiladas ou pré-montadas que executam os comandos.

KX1 (uso explicado posteriormente)

KX2 ..

KX3 ..

A razão de termos essa composição da página é a seguinte: Uma página pode ser arquivada no disco e posteriormente ativada (lida de volta na memória).

O INDTI e o NPG que se referem à particular tabela de instruções que foi arquivada e depois ativada, acompanham-na nestas operações, pois que pertencem a mesma página.

Além disso, como veremos mais tarde, uma subrotina escrita em instruções ERP está associada a uma determinada página. Portanto é natural que os parâmetros transmitidos a ela, acompanhem a página respectiva aonde quer que ela vá.

1.4.9 – TABELA DE RETORNO (SIGLA TR): esta tabela compõe-se de 7 pares de campos de 4 posições mais um campo não utilizado. Cada vez que há um desvio para uma subrotina, toda a tabela sofre um deslocamento de dois campos para a direita.

O par tornado vago a esquerda é preenchido com o número da página na qual se encontrava a instrução contendo a chamada para a subrotina e com a identificação da instrução seguinte. Esta tabela chama-se tabela de retorno, pois que à saída de uma subrotina, o sistema encontra no primeiro par da tabela as informações para executar o desvio de retorno, após o qual a tabela é deslocada dois campos para a esquerda.

Com esta explicação vê-se pela capacidade da tabela que sete chamadas de subrotinas são possíveis antes que aconteça o primeiro retorno.

1.4.10 – ÁREA COMUM DE DADOS (SIGLA AC): esta área ocupa 3.000 posições de memória e está dividida em 300 campos de 10 posições.

Cada campo pode armazenar um número em "ponto flutuante" no formato padrão utilizado pelo FORTRAN. Para isso os campos se decompõem ainda em dois sub-campos; o primeiro com 8 posições reservado à mantissa que é sempre um número do intervalo $0.1 \leq m < 1.0$ e o segundo com duas posições para a característica.

A característica indica a potência de 10 que deve multiplicar a mantissa.

1.4.11 – TABELA DE SÍMBOLOS (SIGLA TS): é formada por 50 áreas de 20 posições. Cada uma destas áreas chamaremos linha pois que correspondem às linhas da tabela esquematizada abaixo (onde algumas linhas de exemplo estão preenchidas).

Nome	Posição na AC	Número Linhas	Número Colunas	Posição do Disco
BT	1	3	7	0
CI	22	4	1	13
AF	0	1	15	14

Figura 1

Antes de prosseguirmos na explicação, tornemos preciso o significado de alguns termos, como empregados neste trabalho:

1.4.11.1 – DEFINIR uma variável, significa preencher uma linha na tabela de símbolos correspondente à variável e reservar espaço na área comum de dados. Toda variável é considerada pelo sistema ERP como uma matriz.

A primeira linha exemplificada na tabela da figura 1 indica que a variável BT foi definida como uma matriz de 3 linhas e 7 colunas, e que os campos de número 1 a 21 na AC foram reservados para seus valores.

1.4.11.2 – APAGAR uma variável significa cancelá-la da TS e da AC. Tanto a TS como a AC sofrem acomodação para eliminar os vazios resultantes da eliminação da variável.

1.4.11.3 – ARQUIVAR uma variável significa guardar seus valores em setores consecutivos do disco sem alterar a AC. Na última coluna da tabela, na linha correspondente à variável, o número de ordem do primeiro setor utilizado pela variável é registrado. Na figura 1 a segunda linha da tabela indica que a variável CI foi arquivada a partir do setor 13 do disco.

A numeração dos setores é relativa ao início da área do disco reservada para arquivo de variáveis e vai de 1 a 2.000. Se uma variável já arquivada o for novamente, ocupará a mesma área do disco.

1.4.11.4 – LIBERAR uma variável significa arquivá-la e ao mesmo tempo retirá-la da AC.

Os valores na AC são deslocados preenchendo o espaço que era ocupado pela variável liberada. A terceira linha da figura 1 indica que a variável AF foi liberada o que é caracterizado pelo zero na segunda coluna e que se encontra arquivada no disco a partir do setor nº 14.

1.4.11.5 – ATIVAR uma variável significa trazê-la do disco para a AC. Se a variável estava simplesmente arquivada, os valores guardados no disco, substituirão os valores registrados na AC. Se a variável estava liberada, nova área será reservada, na AC e o número do primeiro campo utilizado, registrado na coluna dois da tabela.

Cada linha (20 posições) está dividida em 5 campos de quatro posições que servem para registrar respectivamente: o nome da variável (dois caracteres), a posição na AC, o número de linhas da variável, o número de colunas e a posição no disco.

O "estado" de uma variável é caracterizado pelo conteúdo das colunas 2 e 5 da TS.

Conteúdo da col. 2 diferente de zero: ativa
 Conteúdo da col. 2 igual a zero : liberada
 Conteúdo da col. 5 diferente de zero: arquivada
 Conteúdo da col. 5 igual a zero : não arquivada

1.4.12 – TABELA DE COMANDOS (SIGLA TC): é formada por 35 campos de 10 posições. Cada campo correspondente a um comando, e está dividido em dois sub-campos: o primeiro com 8 posições recebe o nome do comando (4 caracteres) e o segundo um número de dois dígitos igual ao número de subrotinas ERPn utilizadas pelo comando, multiplicado por 3. O seu uso será explicado no § 6.

Os cinco primeiros campos reservados para os comandos do sistema são carregados automaticamente com o sistema. Os restantes são carregados a escolha do usuário.

1.5 – ORGANIZAÇÃO DA MEMÓRIA – PROGRAMAS

1.5.1 – Descrevemos a área de programas acompanhando o esquema do ANEXO A.

1.5.2 – As primeiras 7500 posições são ocupadas por: área de multiplicação, tabuada, rotinas do MONITOR, rotinas aritméticas e rotinas de entrada e saída.

As rotinas de entrada e saída são carregadas, quando chamadas, numa área comum desta parte da memória.

1.5.3 – ERP (80 posições)

É escrito em FORTRAN e apesar do seu tamanho desempenha funções importantíssimas.

1.5.3.1 – Sendo escrito em FORTRAN e sendo o programa principal do sistema^(*), permite que sejam ativadas funções do MONITOR por meio de cartões de controle FORTRAN do tipo *LOCAL, bem como sejam ativadas rotinas usadas pelo FORTRAN. Isto permite ainda que as subrotinas que se destinam a executar comandos do ERP possam ser escritas em FORTRAN ou em SPS.

1.5.3.2 – As instruções em FORTRAN do programa ERP são:

```
10 CALL ERP2
    CALL INTER
    GO TO 10
15 CALL PEGUE
    END
```

Como vemos a instrução nº 15 não é executada. Sua presença serve o propósito de provocar a carga da subrotina PEGUE descrita mais adiante.

1.5.3.3 – Na primeira vez que a instrução 10 é executada, a subrotina ERP2 (LOCAL) é chamada e uma de suas funções que é inicializar o sistema é utilizada, isto é áreas não limpas, tabelas são inicializadas, os nomes dos comandos selecionados pelo usuário são lidos na TC.

1.5.3.4 – Nas vezes seguintes em que a instrução 10 é executada, outras subrotinas ERPn são executadas pois que esta instrução é modificada pelo INTERPRETADOR (INTER).

Este programa passa então, indefinidamente, a alterar o controle entre o INTERPRETADOR e uma particular subrotina ERPn.

1.5.4 – PEGUE (242 posições)

Este programa usa ainda 25 posições não utilizadas dentre as 80 posições do ERPn.

É escrito em SPS como subprograma de um programa FORTRAN (ERP), e sua finalidade é simplificar a obtenção dos dados relevantes sobre uma variável, facilitando a codificação de uma subrotina ERPn que deve executar um comando.

Se o comando for codificado em FORTRAN; PEGUE será chamado como no exemplo abaixo:

```
CALL PEGUE (LTS, N, L, K, M)
```

(*) Referido por "Main Program" no manual do MONITOR. I

onde

LTS	(dado)	nº da linha na TS
N	(recebido)	primeiro campo ocupado na AC
L	"	nº de linhas da matriz
K	"	nº de colunas da matriz
M	"	último campo ocupado na AC

Se o comando for codificado em SPS a seqüência de chamada deverá ser:

```
BTM 7514, * +11, 67
DSA LTS, N, L, K, M
```

1.5.5 – SUBROTINA 03 (510 posições)

Esta subrotina pertence ao conjunto de subrotinas relocáveis utilizáveis pelo FORTRAN e é executada quando num programa em FORTRAN aparecem variáveis indexadas.

É a única subrotina relocável do FORTRAN utilizada pelo sistema ERP.

Sua carga é provocada por ser solicitada no "HEADER"^(*) da rotina PEGUE.

1.5.6 – FLIPER (301 posições)

A carga desta rotina é provocada pelos cartões *LOCAL os quais são permitidos porque o programa principal ERP é escrito em FORTRAN. Esta rotina pertence ao sistema MONITOR I e sua finalidade é carregar a subrotina ERPn (LOCAL) solicitada pelo programa principal ERP.

1.5.7 – TABELA DO FLIPER (até 1351 posições)

FLIPER tem como parte integrante uma tabela a qual é construída durante a carga das subrotinas do tipo *LOCAL e por meio da qual estas são achadas nos primeiros cilindros do disco e trazidas para a memória.

1.5.8 – ÁREA ENTRE A TABELA DO FLIPER E A ÁREA DE TABELAS (a partir de 4.144 posições)

Nesta área se revezam o INTERPRETADOR que é do tipo *LOCAL, e as subrotinas ERPn que executam comandos e que são também do tipo *LOCAL.

São chamadas pelo programa principal ERP com o auxílio do FLIPER e de sua tabela.

Esquemáticamente; o INTERPRETADOR altera a instrução 10 do programa principal ERP condicionando-o a chamar ERPn, devolve o controle a ERP, o qual chama ERPn o qual após execução devolve o controle a ERP o qual chama novamente o interpretador e assim por diante.

1.6 – ORGANIZAÇÃO DO DISCO

1.6.1 – Para benefício do leitor não familiarizado com a unidade 1311 de discos magnéticos apresentamos aqui as noções estritamente necessárias para acompanhar este parágrafo.

(*) Informações que precedem um subprograma de um programa FORTRAN no disco, e que são utilizadas na carga do subprograma.

Um conjunto de discos ao qual em geral nos referimos neste manual abreviadamente como "disco", é formado por 6 discos, montados sobre um eixo comum.

Como a face superior do disco superior e a face inferior do disco inferior não são utilizadas temos 10 faces no total.

Cada face é dividida em 100 "trilhas", cada trilha em 20 "setores" e cada setor contém 100 "posições".

As trilhas das diversas faces que se correspondem no sentido vertical formam um "cilindro". Portanto um cilindro contém 10 trilhas, ou 200 setores ou 20.000 posições.

Os setores são numerados de 0 a 19.999 sendo que os setores acham-se ordenados dentro das trilhas, as trilhas ordenadas dentro dos cilindros e os cilindros ordenados no conjunto de discos.

No sistema MONITOR I os 100 cilindros (0 a 99) estão divididos em várias partes com funções específicas. Como padrão, os 24 primeiros cilindros (0 a 23) são considerados de trabalho e estão a disposição do programador. O que segue é uma discriminação do uso que faz o sistema ERP destes 24 cilindros (Vide ANEXO C)

1.6.2 – CILINDROS 0 a 9

Nestes cilindros são guardados durante a carga do sistema as rotinas que são carregadas na memória apenas quando chamadas. São elas, rotinas aritméticas e de entrada e saída do FORTRAN (218 setores), o INTERPRETADOR, e o ERPn. Considerando a capacidade do MONITOR I poderão ser carregados no máximo 49 ERPn. Considerando o espaço reservado no disco pelo sistema ERP e o tamanho de cada ERPn poderão ser carregados no mínimo 42 ERPn.

1.6.3 – CILINDROS 10 a 11

Esta área é reservada para o arquivamento de até 20 páginas (vide noção de página no § 4 e o comando *ADM no § 7).

Como cada página tem 1500 posições, ocupam 15 setores. Cada um dos cilindros 10 e 11 foi dividido em 10 partes de 20 setores e cada uma dessas partes reservadas para uma página.

A razão é que o número de setores de um cilindro (200) não é divisível por 15 e se usadas partes de 15 setores em lugar de 20 isto traria dificuldades adicionais na programação do arquivamento e ativação de páginas com perda de eficiência, e problemas de espaço na memória.

1.6.4 – CILINDROS 22 e 23

Esta área é reservada para o arquivamento de até 4 "fotos" do sistema. Arquivar uma foto do sistema significa armazenar no disco a área da memória que vai da posição 13.998 (início da AK) até a posição 19.999 (fim da AC). Esta área contém 6.002 posições. Os cilindros 22 e 23 foram divididos em duas partes cada um e cada meio cilindro reservado a uma foto.

A razão de usarmos 10.000 posições do disco para guardar 6.002 da memória é análoga a explicada no número anterior com referência ao arquivamento de páginas.

1.6.5 – CILINDROS 12 a 21

Esta área, o restante dos 24 cilindros, é reservada para o arquivamento de variáveis (escalares, vetores ou matrizes).

O número de setores utilizados por uma variável nesta área é dado pela fórmula

$$\text{PARTE INTEIRA DE } \frac{L \times K + 9}{10}$$

onde

L = número de linhas

K = número de colunas

Portanto na pior das hipóteses (todas as variáveis sendo escalares) poderemos arquivar até 2.000 dados e na melhor das hipóteses (todas as variáveis tendo L x K múltiplos de 10) poderemos arquivar até 20.000 dados.

Estes números comparam-se favoravelmente com a capacidade de 300 dados da AC.

Nesta área do disco, para garantir um melhor aproveitamento do espaço não são respeitadas as fronteiras dos cilindros.

Uma variável pode ter parte de seus elementos arquivados num cilindro e o restante no cilindro seguinte.

1.7 – COMANDOS DO USUÁRIO

1.7.1 – Para o usuário incorporar um comando de seu interesse na biblioteca de comando do sistema deverá preliminarmente:

1.7.1.1 – Dar um nome ao comando evitando duplicação dos quatro primeiros caracteres com comandos já existentes.

1.7.1.2 – Definir perfeitamente todas as opções e sub-opções deste comando, isto é as regras para se escreverem os parâmetros respeitando o formato das instruções do sistema.

1.7.1.3 – Definir qual será a execução de cada opção e sub-opção.

1.7.1.4 – Determinar quais todas as situações que serão detectadas como erros e interromperão a execução da instrução.

1.7.1.5 – Escolher entre o FORTRAN e o SPS, qual a linguagem melhor adaptada (em termos de facilidade de codificação, eficiência e espaço) para a codificação do comando.

1.7.1.6 – Verificar na biblioteca de comandos qual o último número usado para numeração das subrotinas que executam comandos.

Se este número for por exemplo 47 e o comando a ser codificado exigir por exemplo duas subrotinas, estas serão chamadas ERP48 e ERP49.

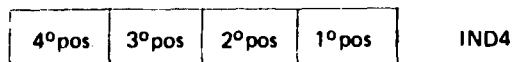
O motivo desta padronização está em evitar a duplicação de nomes com os de outras subrotinas que podem estar armazenadas no disco, e que nada tem a ver com o sistema ERP.

1.7.2 – Para programar a execução da instrução referente ao comando projetado é necessário que o usuário conheça como o sistema usa os campos IND4, ERRO (KERRO na codificação em

FORTTRAN) KX1 e KX2 da área de comunicação e como os parâmetros de uma instrução se apresentam decodificados na AP (area de parâmetros)

1.7.2.1 – IND4

É um campo de 4 posições tendo cada uma das quatro posições uma significação propria e independente.



As posições que nos interessam agora são a 2º e a 4º.

Estas posições serão ditas ligadas se diferentes de zero e desligadas se contiverem zeros

A posição nº 2 é consultada pelo INTERPRETADOR

Se estiver ligada é desligada e o INTERPRETADOR chama imediatamente a subrotina ERPn cujo código se encontrar no momento no primeiro campo da AP

É utilizada portanto para permitir que uma subrotina tipo *LOCAL chame outra subrotina tipo *LOCAL.

A posição nº 4 é consultada pelo INTERPRETADOR

Seu uso será melhor explicado por meio de um exemplo. Consideremos o comando *E/S (vide diagrama de bloco e codificação no ANEXO H) o qual utiliza ERP5 e ERP6 para sua execução.

ERP5 logo à entrada consulta a posição 4 de IND4.

Se estiver desligada isto significa que é a primeira entrada na subrotina ERP5

Caso durante a execução da instrução a subrotina ERP6 deva ser chamada, ERP5 liga a posição 2 coloca o código de ERP6 no primeiro campo da AP e devolve o controle ao INTERPRETADOR Este consulta a posição 2 de IND4, desliga-a e transfere o controle para ERP6.

A saída, ERP6 liga a posição 4 de IND4 e chama ERP5 por intermédio do INTERPRETADOR com procedimento análogo ao já descrito.

Desta vez ao consultar a posição 4 de IND4, ERP5 encontra-a ligada recebendo portanto a informação que a execução já foi iniciada e deve ser continuada a partir do ponto em que ERP6 a deixou A posição 4 de IND4 é desligada por ERP5 logo após a consulta.

Deve-se fazer com que a devolução definitiva do controle ao INTERPRETADOR seja feita sempre pela primeira subrotina que executa o comando para que o esquema lógico descrito acima funcione.

1.7.2.2 – ERRO (KERRO em FORTRAN)

É um campo de 4 posições cuja missão é receber um código de erro e transmiti-lo para a subrotina ERP4 de recuperação de erro.

Ao detectar um erro na execução de uma instrução a subrotina ERPn em controle, coloca em ERRO o código do erro e transmite o controle à ERP4 cujo código é 120 através do

INTERPRETADOR, com procedimento já descrito acima. O campo ERRO é apagado pelo INTERPRETADOR quando este recebe o controle de ERP4.

1.7.2.3 – KX1

Este campo de 4 posições serve para transmitir o número de ordem do próximo parâmetro a ser processado na execução de uma instrução.

Este campo é atualizado toda vez que uma subrotina ERPn chama outra.

1.7.2.4 – KX2

Este campo de 4 posições é usado para transmitir para ERP4 alguma informação relativa ao erro que será processado por ERP4.

Por exemplo, no erro de código 23:ESTOURO DA ÁREA COMUM DE DADOS, serve para transmitir o total do "deficit" encontrado. (Para informações mais detalhadas sobre os diversos códigos de erro refira-se ao MANUAL DE OPERAÇÃO).

1.7.2.5 – ÁREA DE PARÂMETROS

Vamos explicar como se apresenta uma instrução após decodificada na área de parâmetros por meio de um exemplo.

Suponhamos que as variáveis A, C e H encontram-se definidas e os respectivos estados registrados nas linhas 3, 5 e 15 da TS.

Consideremos a instrução:

04 *E/S.LEIA A C H .EXIBA .AC 5 70

Após decodificação pelo INTERPRETADOR ela se apresenta na AP com o seguinte aspecto:

*E/S	LEIA			A	C	H			
0150	0300	5345	0000	0003	0000	0005	0000	0015	continua
EXIBA			AC	5	70				
0300	4567	0300	4143	1300	0005	1300	0070	3400	

No primeiro campo encontra-se o código do primeiro ERPn utilizado no comando *E/S, no caso ERP5 cujo código é 150.

Em seguida cada parâmetro utiliza dois campos.

No primeiro teremos 0300, 0000 ou 1300 conforme o parâmetro for um literal, uma variável ou um número.

No segundo campo teremos:

- a) no caso de variável o número de ordem da sua linha na TS.
- b) No caso de literal, os dois primeiros caracteres do literal escritos no código alfa-numérico do IBM-1620

Assim LE (IA) = 5345 pois 53 é o código de L e 45 o de E.

- c) no caso de número, o próprio número.

Finalmente indicando o fim da instrução comparece um último campo contendo 3400.

No código alfa-numérico de dois dígitos teremos:

0300	significa	<u>. b</u>
0000	"	<u>b b</u>
1300	"	<u>\$ b</u>
3400	"	<u>(a b</u>

Os caracteres de identificação da instrução não aparecem na AP.

No capítulo II, explanamos as limitações encontradas e a forma como as superamos para uma utilização mais eficiente do sistema ERP no Computador IBM 1620 modelo 2 de 40.000 posições de memória.

Em volumes a parte encontram-se colecionados os diagramas de blocos, tanto os originais com as modificações introduzidas quanto os correspondentes aos novos comandos implementados, e as listagens de todos os programas até aqui implementados. Essa documentação está formada por mais de 150 páginas de diagramas e 127 páginas de codificação.

CAPÍTULO II

NOVA DISTRIBUIÇÃO DE ÁREAS DA MEMÓRIA E DO DISCO

2.1 – TRABALHO DE ADAPTAÇÃO

A implementação do sistema ERP no Computador IBM 1620 mod. 2 requereu várias etapas de adaptação às novas disponibilidades:

- Verificação das novas facilidades disponíveis.
- Verificação da compatibilidade do sistema com o modelo da máquina.
- Análise das limitações apresentadas pela versão original.
- Redistribuição das áreas.

- Junção, nos casos possíveis, dos subprogramas ERPn, que executam um comando.
- E transformação em residente na memória de núcleos, do subprograma interpretador (INTER).

2.2 – FACILIDADES DISPONÍVEIS NO COMPUTADOR IBM 1620 MOD. 2 de 40 K

- Capacidade: 40.000 posições de memória.
- Ciclo de memória: 10 μ s.
- 2 unidades de discos IBM 1311 sendo uma modelo 3 e outra modelo 1.
- Discos magnéticos com 2 milhões de dígitos de capacidade.
- 1 impressora "CN LINE", IBM 1443. Velocidade de impressão de 240 linhas por minuto, quando usa a barra de impressão de 52 caracteres. Esta velocidade aumenta para 600 linhas por minuto quando se usa barra de 13 caracteres (todos numéricos). Cada linha comporta 144 caracteres.
- 1 máquina de escrever Selettric. Velocidade de impressão de 15,5 caracteres por segundo.
- 1 leitora-perfuradora de cartões. Velocidade de leitura 500 cartões por minuto. Velocidade de perfuração, 250 cartões por minuto.
- CPU IBM 1620 mod. 2
- Memória de núcleos: IBM 1625 mod. 2.
- Dispositivo de ponto flutuante automático.
- Divisão automática.
- Sistema Operacional Monitor I com compilador FORTRAN II-D e Montador SPS II.

2.3 – VERIFICAÇÃO DA COMPATIBILIDADE

Para verificar que o sistema ERP era compatível com o modelo do Computador disponível, utilizamos o programa de utilidade de disco que permite definir os parâmetros da máquina. Com isto reduzimos a capacidade de memória a 20.000 dígitos. Esta verificação foi simplesmente realizada para ter certeza que o sistema ERP, que ainda conhecíamos pouco na sua estrutura, não apresentava surpresa alguma. O resultado foi positivo.

2.4 – ANÁLISE DAS LIMITAÇÕES APRESENTADAS PELA VERSÃO ORIGINAL

As limitações que encontramos no sistema original devem-se exclusivamente à exiguidade da memória disponível. Apontamos principalmente os seguintes:

- a) A área de memória de núcleos reservada para os subprogramas tipo LOCAL está limitada a 4144 posições, conseqüentemente isto implica na segmentação dos subprogramas que executam alguns comandos.

Isto acarreta um aumento no tempo de execução do comando, pois sempre será necessário procurar cada parte, por vez, do disco e transferi-la para a memória de núcleos

- b) Área de parâmetros limitada a 50 campos consequentemente comprimento da instrução também limitado
- c) Área de registros das instruções no modo RG, limitada a 29 caracteres, requer a utilização de várias instruções continuação. Reflete-se também num maior investimento de tempo de execução
- d) Área comum com 300 dados. Isto acarreta uma limitação na dimensão das matrizes com as quais o sistema poderá operar. No caso de tratar-se de matriz quadrada, a dimensão máxima permissível é de 17 x 17 elementos
- e) Áreas reservadas no disco, para arquivar dados e fotos, encontram-se localizadas na área de trabalho no Monitor(*).

Consequentemente não serão possíveis interrupções entre uma execução e outra do sistema. Na nova versão, colocamos tanto a área de dados (10 cilindros) quanto a área destinada às fotos (4 cilindros) na área de programas permanentes do Monitor (ver ANEXO 3) Isto permite que um trabalho interrompido possa ser continuado em qualquer outro momento.

2.5 – NOVA DISTRIBUIÇÃO DE ÁREAS (Ver ANEXOS 1, 2 e 3)

2.5.1 – ÁREA DE PROGRAMAS

A área reservada para programas inicia-se, devido a utilização da impressora, na posição 8100 e estende-se até a posição 24939 num total de 16940 posições de memória. É assim constituída:

a) PROGRAMA PRINCIPAL ERP (104 posições)

Este programa, sem alterar as suas funções, foi modificado, ficando aumentado em 24 posições. Os comandos do mesmo seguem uma lógica modificada introduzindo-se uma chamada ao subprograma BASERP, nova base de referência.

A introdução deste subprograma deve-se ao fato de que na nova versão o Interpretador (INTER) fica residente na memória. O Interpretador, modificando o primeiro endereço do programa principal, define qual o ERPn que deve entrar em ação. Isto se consegue somando $30 \cdot n$ (sendo n o número relativo do ERPn) ao endereço da base de referência, cujas instruções estão constituídas pelas primeiras 30 posições da tabela FLIPER, que originalmente correspondia ao próprio INTER. Este era o primeiro subprograma referenciado no registro LOCAL. Agora corresponde ao subprograma BASERP que nunca é executado. No total perdem-se 54 posições de memória, mas em compensação não foram introduzidas modificações profundas na lógica dos subprogramas originais do sistema.

O programa principal ERP, portanto ficou constituído assim:

```

10 CALL ERP1
   GO TO 20
15 CALL PEGUE
20 CALL INTER
   GO TO 10
25 CALL BASERP
   END

```

(*) A área de trabalho que corresponde aos 24 primeiros cilindros é modificada cada vez que entra em operação um novo programa

b) SUBPROGRAMA PEGUE (242 posições)

Não sofreu modificação, além dos endereços reais necessários.

Endereço de carga 8204.

c) SUBPROGRAMA INTER (3792 posições)

Este subprograma que originalmente era do tipo LOCAL passou a ser residente. Isto é, ele é carregado na memória em continuação ao PEGUE, e permanece lá durante todo o tempo que o sistema esteja em ação. Consegue-se assim uma maior velocidade de execução, já que não é necessário carregá-lo do disco, todas as vezes que deve ser ativado.

Não foi necessário introduzir modificações sensíveis além dos endereços reais utilizados no subprograma.

Endereço de carga: 8446.

d) SUBPROGRAMA 03 (510 posições)

É carregado em continuação ao INTER no endereço: 12238.

e) SUBROTINA FLIPER (301 posições)

Endereço de carga 12748.

f) TABELA DO FLIPER (1500 posições)

A tabela é formada por informações que ocupam 30 posições por cada subprograma referenciado no registro LOCAL. Como o FORTRAN permite que sejam utilizados até 50 subprogramas deste tipo, evidentemente, quando forem chamados todos os subprogramas permissíveis, a tabela ocupará as 1500 posições previstas.

Endereço de carga: 13048.

g) ÁREA DE SUBPROGRAMAS TIPO LOCAL

Esta área é reservada para comportar subprogramas até 10389 posições de memória: estende-se até a posição 24939. O endereço de carga dependerá de quantos subprogramas LOCAL são referenciados no registro LOCAL correspondente para executar os comandos do sistema necessários à tarefa em andamento.

2.5.2 – ÁREAS E TABELAS

Seguindo a área destinada aos programas são definidas as áreas e tabelas que serão utilizadas pelo sistema.

a) ÁREA S (AS)

8 posições

Endereço: 24939

Não sofreu modificações

b) ÁREA DE LEITURA (AL)

162 posições
Endereço: 24947
Não sofreu modificações

c) ÁREA DE PARÂMETROS (AP)

392 posições
Endereço: 25109

Esta área foi aumentada para 98 campos de 4 posições. Como cada parâmetro utiliza 2 campos, ela tem capacidade para 48 parâmetros, pois o primeiro campo de 4 dígitos é ocupado pelo código ERPn e o último campo pelo indicador de fim de instrução (3400)

d) ÁREA DE COMUNICAÇÕES (AK)

112 posições
Endereço: 25501
Não sofreu modificações

e) ÁREA DE PARÂMETROS TRANSMITIDOS (PT)

40 posições
Endereço: 25613
Não sofreu modificações

f) TABELA DE INSTRUÇÕES (TI)

3936 posições
Endereço: 25653

O tamanho dos 24 campos foi aumentado de 60 para 134 posições. Desta forma as instruções registradas correspondem à capacidade de um cartão ou seja 80 caracteres. Não é mais necessário, como na versão original, quando a entrada é pela máquina de escrever, limitar a 29 caracteres. Diminui assim a necessidade de usar instruções continuação, aumentando conseqüentemente, a velocidade de processamento. O aumento desta tabela evidentemente, ocasiona também o aumento do tamanho da página, que incluindo os 5 últimos campos da área de comunicações e a área de parâmetros transmitidos, ocupa atualmente 3996 posições. Cada página, então, ao ser arquivada no disco ocupará 40 setores.

g) TABELA DE RETORNO (TR)

60 posições
Endereço: 25589
Não sofreu modificações

h) TABELA DE SÍMBOLOS (TS)

1000 posições
Endereço: 29649

As 50 áreas de 5 campos de 4 posições, continuam as mesmas que na versão original. Achamos que os 50 símbolos possíveis de serem definidos, são suficientes e estão em relação correta com as dimensões das demais áreas e tabelas.

i) TABELA DE COMANDOS (TC)

350 posições
Endereço: 30649
Não sofreu modificações

j) ÁREA COMUM (AC)

9000 posições
Endereço: 30999

A área comum de dados foi triplicada. Assim, pode comportar uma matriz de até 30 x 30, dimensão que se compara favoravelmente com o tamanho dos subprogramas que o sistema pode comportar para a execução de um determinado comando. Isto aumenta enormemente a flexibilidade de programação, aumentando a eficiência de todo sistema.

2.5.3 – ORGANIZAÇÃO DO DISCO (ver ANEXO 3)

De acordo com as modificações introduzidas na distribuição de áreas de memória do Computador e as maiores facilidades que oferece uma memória de maior capacidade, fizemos uma nova distribuição de memória de arquivo de disco, ficando assim organizada:

a) CILINDROS 0 a 19 (20 CILINDROS)

Nestes cilindros são carregadas as rotinas aritméticas e de entrada e saída do FORTRAN e os subprogramas tipo LOCAL (ERPn), 49 no total. Como o tamanho de cada subprograma pode ser de até 10.389 posições poderão ser carregados até 38 ERPn de tamanho máximo.

b) CILINDROS 20 a 23 (4 CILINDROS)

Reservados para arquivamento de 20 páginas, ocupando cada página 40 setores. O número de páginas é igual ao da versão original, mas com capacidade dupla.

c) CILINDROS 24 e 25 (2 CILINDROS)

São ocupados pela tabela DIM e a tabela de Equivalência do sistema Monitor I^(*).

d) CILINDROS 26 a 35 (10 CILINDROS)

Estes cilindros são reservados em forma permanente para o arquivamento dos valores das variáveis. No caso em que todas as variáveis fossem escalares, poderíamos arquivar até 2.000 dados e, se todas as variáveis fossem matrizes em que o produto linha por coluna fosse múltiplo de 10, poderíamos arquivar até 20.000 dados, como na versão original.

e) CILINDROS 36 a 39 (4 CILINDROS)

Nestes cilindros poderemos arquivar até 4 fotos, isto é, em cada cilindro uma foto. As razões que nos levaram a reservar um cilindro para cada foto, cujo tamanho é de 14.418 posições ficando mais de 50 setores sem utilização, são as mesmas que as especificadas na versão original (vide capítulo I).

Tanto a área para dados quanto a área para fotos foram reservadas mediante a utilização de um programa de utilidade de disco, possuindo portanto um número de entrada na tabela DIM e um

(*) Estas duas tabelas são próprias do Monitor e servem para registrar os dados referentes aos programas gravados no disco em forma permanente.

nome na tabela de Equivalência do Monitor. Assim não será ocupado por nenhum programa mesmo que este não faça parte do sistema ERP.

CILINDROS 40 EM DIANTE

Ficam disponíveis para serem utilizados pelo Sistema Monitor para armazenamento de programas ou dados administrados por ele, assim como, os programas e tabelas próprios desse sistema.

2.6 – ADAPTAÇÃO DOS SUBPROGRAMAS DO SISTEMA

As adaptações dos subprogramas que executam os comandos do sistema original, todos do tipo LOCAL, foram as seguintes:

a) INTER

O interpretador, escrito em SPS como subprograma FORTRAN, como já dissemos, foi deixado residente devido ao fato de que ele é um dos subprogramas que mais freqüentemente é ativado. Não houve necessidade de modificações sensíveis na sua lógica. Foram modificados somente os endereços reais que dependiam da distribuição das áreas e tabelas utilizadas pelo sistema. Para facilitar qualquer modificação que se deseja introduzir novamente, utilizamos como endereço base, o fim da memória que é também o fim da área comum de dados, como único endereço real. Baseado nisto definimos os outros endereços por ajustes que são funções das dimensões das áreas reservadas. Assim, só será necessária a troca de um ou dois cartões para refletir a nova distribuição pretendida. Adicionamos uma verificação do dispositivo de saída vigente, possibilitando as saídas necessárias, tanto pela máquina de escrever quanto pela impressora.

b) Os subprogramas ERP1 e ERP2 eram os que executavam o comando Administre (*ADM) (ver descrição formal e opções no ANEXO 5). Cada um deles ocupava quase que toda a área reservada para os subprogramas LOCAL na memória de núcleos. Como a capacidade atual é mais do que o dobro da original, foi possível condensá-los em um só subprograma. Consegue-se com isto uma velocidade de processamento maior.

A lógica geral não sofreu modificações. Só a opção SUBROTINA teve um adendo. Este consistiu numa nova facilidade, a de que uma SUBROTINA escrita com instruções do sistema possa ser referenciada com um nome e não somente com o número da página como era originalmente. Para isto, basta definir o nome como se fosse uma variável com a opção DEFINA do Comando *ADM e em seguida liberá-la para não ter parte da área comum de dados reservada inutilmente.

O número da página entra na definição da variável como se correspondesse ao valor da linha da mesma. A coluna será considerada igual 1. Quando a instrução é interpretada, o nome da subrotina será substituído pelo número da página correspondente, em forma automática. Portanto qualquer subrotina do sistema pode ser referenciada tanto pelo seu nome quanto pelo número da página, dando assim mais facilidade para a programação.

Os dois ERPn reunidos constituem atualmente o ERP1

c) O subprograma ERP3 que executa o comando CONTROLE (*CTL) permite que possa ser modificado o dispositivo de E/S, o MODO de operação ou o conteúdo de qualquer elemento individual da área de comunicações, da área de parâmetros transmitidos, da tabela de retorno ou da tabela de símbolos em forma programada ou não.

Além dos endereços reais necessários, uma alteração introduzida na lógica de programação, foi na opção SAÍDA para comportar a impressora disponível "ON-LINE" (ver ANEXO 5). Outro adendo consistiu na introdução da opção FIM que devolve automaticamente o controle ao sistema Monitor.

Não foi necessário uni-lo com outro subprograma, porque a sua ação é independente de qualquer outro, assim como o seu uso

d) O subprograma ERP4, destinado ao diagnóstico dos possíveis erros e emissão das mensagens correspondentes, conserva a sua lógica original. Foram introduzidas duas novas mensagens que a implementação de novos comandos, principalmente o CALCULE, exigiam. Estas novas mensagens são: 1) PARENTESES NÃO CASAM e 2) TENTANDO DIVIDIR POR ZERO. Ambos ocasionam a inibição da execução da instrução. No ANEXO 6 reproduzimos todas as mensagens que o sistema pode fornecer. Nos novos comandos adicionados, tratamos na média do possível de utilizar na medida das mensagens existentes, já que elas são suficientemente completas. Todas as mensagens de erros somente têm saída pela máquina de escrever, devido ao caráter conversacional do sistema. Qualquer erro deve ser corrigido imediatamente, assim não achamos prático a saída pela impressora^(*)

e) O comando de entrada e saída (*E/S) era executado pelos subprogramas ERP5 e ERP6. Na nova versão reunimos num único subprograma passando a ser o ERP2. Foram introduzidas substanciais modificações, mas sempre dentro da lógica original, para adicionar a possibilidade de saída pela impressora de todas as opções disponíveis. Modificamos algumas mensagens para adaptá-las às novas disponibilidades. Obviamente a velocidade de respostas do computador a qualquer requerimento do usuário é extraordinariamente mais rápida. A saída pela impressora é conveniente, num sistema conversacional, quando as respostas do computador não requerem uma imediata réplica de parte do usuário, isto é, quando se quer estudar os resultados obtidos com mais detalhes e mais tempo. Deste modo pode-se abandonar o processamento nesse ponto e só retomá-lo após uma análise completa, não prendendo inutilmente o computador.

CAPÍTULO III

COMANDOS DO SISTEMA IMPLEMENTADOS

3.1 – Originalmente, o ERP possuía 8 comandos, chamados COMANDOS DO SISTEMA e que seriam carregados automaticamente junto com o sistema. Os comandos *MANIPULE e *CALCULE estavam projetados naquela versão mas não implementados. Nesta nova versão completamos as suas definições e os implementamos possibilitando o cálculo de expressões aritméticas e manipulação de matrizes. Com isto fica aumentado extraordinariamente o poder do sistema.

Os disponíveis até o momento eram:

- *ADM (Administre)
- *CTL (Controle)
- *E/S (Entrada/Saída)
- *NOP (Não opere)
- *MSG (Mensagem)
- *MSP (Mensagem e Pausa)

Notemos que todos têm como primeiro caráter um asterisco. Isto os diferencia dos comandos construídos pelo usuário, que não devem levar este caráter inicialmente. Assim evita-se o perigo de duplicação.

Os três últimos comandos são parte integrante do INTERPRETADOR e não correspondem a um ERPn em particular.

(*) No ANEXO 6 damos uma relação de todas as mensagens de erro

3.1.1 – Na descrição formal dos comandos exceto no Comando *CAL seguiremos as convenções originais que são as seguintes:

- a) "As letras v , l , n devem ser substituídas por variáveis, literais e números, respectivamente, de acordo com o contexto da instrução.

Se várias variáveis possivelmente distintas devem aparecer em vários lugares são usados índices v_1, v_2, \dots

Analogamente para os literais e números."

- b) "Se várias opções são incluídas dentro de colchetes:

[]

Isto significa que na instrução pode não aparecer nenhuma delas ou uma das opções deve ser escolhida."

- c) "Se várias opções estão incluídas dentro de chaves:

{ }

isto significa que obrigatoriamente uma das opções deve comparecer na instrução."

- d) "A locução etc seguindo um parênteses direito significa que tudo entre este e o correspondente parênteses esquerdo pode ser repetido um número arbitrário de vezes."

3.1.2 – Na descrição formal do comando *CAL não seguimos a convenção anterior para evitar incoerências possíveis; porque a sua definição tem caráter recursivo e achamos mais conveniente adotar a notação de Backus usada para descrição formal da sintaxe de linguagens. Para uma expressão aritmética a convenção é a que segue:

$\langle ea \rangle ::= \langle t \rangle | \langle ea \rangle + \langle t \rangle | \langle ea \rangle - \langle t \rangle$

$\langle t \rangle ::= \langle f \rangle | \langle t \rangle * \langle f \rangle | \langle t \rangle / \langle f \rangle$

$\langle f \rangle ::= \langle pr \rangle | \langle f \rangle ** \langle pr \rangle$

$\langle pr \rangle ::= \langle u \rangle | \langle v \rangle | \langle df \rangle | (\langle ea \rangle)$

$\langle v \rangle ::= \langle l \rangle | \langle v \rangle \langle ct \rangle$

$\langle ct \rangle ::= \langle l \rangle | \langle d \rangle$

$\langle l \rangle ::= A | B | \dots | Z$

$\langle d \rangle ::= 0 | 1 | \dots | 9 |$

$\langle c \rangle ::= \langle d \rangle | \langle d \rangle \langle c \rangle$

onde

e a = expressão aritmética; t = termo; f = fator; pr = primário; v = variável; ct = caráter; l = letra; d = dígitos; df = designador de função; c = constante.

3.2 – COMANDO MANIPULE (*MAN)

Este comando realiza manipulação de vetores e matrizes, i.e., dados uma ou mais matrizes ou vetores ele executa operações de redistribuição dos elementos, para se obter novas matrizes e vetores, sem se perderem os originais.

3.2.1 – DESCRIÇÃO FORMAL DO COMANDO *MAN (ERP5)

Para a descrição formal do comando reproduzimos a descrição que originalmente fôra prevista e que na preparação do subprograma ERP5, que o executa, seguiu-se estritamente

*MAN	[. DECOMPONHA v .EM .{f n} etc. . COMONHA { v } etc .EM .l. . PARTA v .EM .{.l n} etc. . GRUPE { v } etc .EM .l . TRANSPONHA { v } etc.]	etc
------	---	--	---	-----

3.2.2 – DESCRIÇÃO DE CADA OPÇÃO E EXEMPLOS

a) "Opção .DECOMPONHA

Exemplo:

*MAN .DECOMPONHA A .EM B 2 .C 5 .E 10.

Execução

A deve ser um vetor com $2+5+10 = 17$ componentes. Os vetores B, C e E são definidos como formados pelos elementos de A na ordem em que se apresentam.

O vetor A é arquivado e na TS passa a constar como liberado. Pode-se portanto daqui em diante modificar os vetores B, C e E sem perder A que poderá a qualquer momento ser ativado vindo a ocupar lugar diferente na AC."

b) "Opção .COMONHA

Exemplo:

*MAN .COMONHA B C .EM .F.

Execução

B e C devem ser vetores. Estes são deslocados na AC de forma a ficarem justapostos na ordem em que se apresentam na instrução.

Em seguida um novo vetor F formado pelos componentes de B mais as de C, é definido.

Finalmente B e C são arquivados e passam a constar na TS como liberados."

c) "Opção .PARTA

Exemplo:

*MANIPULE PARTA H EM M 2 .N 3 P 5.

Execução

H deve ser uma matriz com $2+3+5 = 10$ colunas. Novas matrizes M, N, P são definidas com o número de linhas igual ao de H e o número de colunas respectivamente 2, 3 e 5, formadas repartindo as colunas de H.

A matriz H é arquivada e passa a constar como liberada na TS".

d) "Opção .GRUPE

Exemplo:

*MAN .GRUPE N P EM L.

Execução:

N e P devem ser matrizes com o mesmo número de linhas. P é deslocada na AC de forma a ficar justaposta a N. Uma nova matriz L é definida com o mesmo número de linhas e cujas colunas são as de N mais a de P.

N e P são arquivadas e passam a constar na TS como liberadas."

e) "Opção .TRANSPONHA

Exemplo:

*MAN .TRANSP L

Execução:

Os elementos de L são reagrupados na AC de maneira que as colunas se tornem linhas e as linhas colunas

A TS é atualizada para refletir esta nova situação, L não é arquivada. A transposição executa-se na mesma área ocupada pela matriz original."

A matriz pode ser qualquer, quadrada ou retangular

Nas páginas 65 a 70 reproduzimos a saída, pela impressora, dos exemplos anteriores, para mostrar como ficam as áreas e tabelas afetadas.

3.2.3 – DESCRIÇÃO DO ERP5

Para descrever em forma breve o subprograma ERP5 que executa o comando *MANIPULE vamos acompanhar o diagrama simplificado das páginas 42 e 43

Ele usa dois ponteiros ENDS e ENDI que inicialmente situam-se no 2º e 3º campo da AP. Verifica na área assinada pelo ponteiro ENDS se existe um indicador de fim de instrução (@ = 3400). Se existe, devolve o controle ao INTERPRETADOR, se não, verifica se é um ponto (0300); se não for ponto transfere o controle ao ERP4 que cuida dos erros e emite as mensagens necessárias. Se é um ponto prossegue com a análise, verificando qual das opções está sendo indicada pelo ponteiro ENDI.

Uma vez conhecida qual é a opção, modifica o valor do identificador TESTE de acordo com a opção. Avança os ponteiros para continuar o teste de outros dois campos.

Verifica se ENDS contém branco. Se não, transfere o controle ao ERP4 por considerar um erro a ausência de um campo em branco, que indicaria que o campo seguinte é ocupado por uma variável. Em caso positivo, chama o subprograma PEGUE para determinar onde se iniciam, dentro da AC, os

dados correspondentes à variável (matriz ou vetor). Conhecidos os elementos da variável (1º campo ocupado na AC, número de linhas, número de colunas e último campo ocupado na AC) verifica o identificador TESTE. Se este possui o valor 4 desvia para a subrotina que executa a transposição dos elementos de uma matriz. Caso seja diferente de 4, indica que deve desviar para a execução de uma das outras opções: DECOMPONHA E COMPONHA para manipular vetores; PARTA e GRUPE para manipular matrizes.

As subrotinas que executam cada uma dessas opções são identificadas pelo valor atual guardado em TESTE.

Neste ponto para o avanço dos ponteiros ENDS e ENDI e continua a fase de análise da instrução com outro ponteiro, para verificar a consistência de indexação da variável (matriz ou vetor) que de acordo com a opção a ser executada deve existir (vide parágrafo 3.2.2) Se é detectado alguma falta de consistência na indexação, o controle é transferido ao ERP4 para emissão da correspondente mensagem de erro, e a execução é abandonada retornando o controle ao INTERPRETADOR.

Se existe consistência, a variável é liberada (subrotina LIBERA) e conseqüentemente os valores correspondentes são arquivados no disco e os dados que permanecem na AC são deslocados para preencher as vagas deixadas pelo arquivamento referido. A tabela de símbolos é atualizada para refletir o novo estado da variável. A seguir é ativada a subrotina COPIA que copia novamente os valores arquivados em continuação ao último campo ocupado na AC. Esses valores serão atribuídos às novas variáveis que serão definidas durante a execução da instrução, na ordem em que nela aparecem.

Avança novamente os ponteiros ENDS e ENDI e analisa se o campo apontado por ENDS é branco. Se sim, é chamado outra vez o PEGUE para determinar os elementos da nova variável apontada por ENDI e assim continua o processo anterior (CONSIG,LIBERA,COPIA), até que o ponteiro ENDS não tenha branco, terminando a análise das variáveis. Os ponteiros tratam de localizar a seguir o literal .EM. Se ENDS acha uma indicação diferente de ponto, transfere o controle a ERP4 (ERRO), caso contrário avança e determina os nomes das novas variáveis e as define. Feito isto, procuram (os ponteiros) a indicação de fim de instrução, um literal em branco. Se não encontram é sinal de que existe ainda uma ou mais opções a serem executadas ou se não encontram o literal (sempre precedido pelo ponto) é erro e a execução é abandonada, transferindo o controle ao INTERPRETADOR. A falta de fim de instrução não invalida a execução realizada até esse ponto.

Dentre as diferentes opções do comando, achamos que a opção transponha é a que apresentou maior problema na definição da lógica a seguir. De fato, como o comando foi programado em SPS é já a disponibilidade da memória de núcleos era reduzida, fomos obrigados a criar um algoritmo que tratasse com poucas áreas auxiliares.

3.3 – COMANDO CALCULE *CAL

Este comando calcula as expressões aritméticas escritas numa forma muito similar a uma expressão em FORTRAN. A única diferença consiste no uso do ponto precedendo um literal. Os subprogramas que executam este comando são os ERP6 e 7.

3.3.1 – DESCRIÇÃO FORMAL DO COMANDO *CAL

```
*CAL <v> . =<ea>
<ea> ::= <t> | <ea> .+ <t> | <ea> .- <t>
<t> ::= <f> | <t> .* <f> | <t> ./ <f> |
<f> ::= <pr> | <f> .** <pr>
```

$\langle pr \rangle ::= \langle c \rangle | \langle v \rangle | \langle df \rangle | (\langle ea \rangle)$
 $\langle v \rangle ::= \langle \ell \rangle | \langle v \rangle \langle ct \rangle$
 $\langle ct \rangle ::= \langle \ell \rangle | \langle d \rangle$
 $\langle c \rangle ::= \langle d \rangle | \langle c \rangle \langle d \rangle$
 $\langle \ell \rangle ::= A | B | \dots | Z$
 $\langle d \rangle ::= 0 | 1 | \dots | 9$
 $\langle df \rangle ::= \langle LOG \rangle | \langle LN \rangle | \langle EXP \rangle | \langle DX \rangle | \langle RQ \rangle |$
 $\langle ARTAN \rangle | \langle SEN \rangle | \langle TAN \rangle | \langle ASB \rangle$

3.3.2 – OPERADORES SÍMBOLOS E FUNÇÕES

a) Os operadores que se podem usar no comando *CAL são:

Prefixo	.	C ⁻	
Infixo	.	+	para soma
"	.	-	para subtração
"	.	*	para multiplicação
"	.	/	para divisão
"	.	**	para potenciação (tratado como função)

As regras de utilização dos operadores são:

- I. Nenhum operador pode aparecer seguindo imediatamente outro operador e pelo fato de serem considerados literais, pelo sistema, sempre devem estar precedidos pelo ponto
- II. O operador de potenciação admite como expoente seja uma variável previamente definida seja um número inteiro.

b) O símbolo = corresponde ao sinal igual e os símbolos (e) correspondem ao parênteses esquerdo e direito respectivamente.

Podem-se usar um número ilimitado de pares de parênteses, numa expressão.

c) As funções implementadas são as seguintes:

.LOG	–	logarítmo base <u>10</u>
.LN	–	logarítmo base <u>e</u>
.EXP	–	Exponencial
.DX	–	Potencia de <u>10</u>
.RQ	–	Raiz quadrada
.SEN	–	Seno trigonométrico
.COS	–	Coseno "
.TANG	–	Tangente "
.ATAN	–	Arco tangente
.ABS	–	Valor absoluto

Observação: O operador **.** é tratado pelo subprograma que executa o comando como uma função.

I. A função LOG que calcula o logaritmo na base 10, utiliza a aproximação dada por Hastings⁽⁶⁾ (pag. 111).

$$\log_{10} X = \frac{1}{2} + C_1 \left(\frac{x - \sqrt{10}}{x + \sqrt{10}} \right) + C_3 \left(\frac{x - \sqrt{10}}{x + \sqrt{10}} \right)^3 + \dots + C_9 \left(\frac{x - \sqrt{10}}{x + \sqrt{10}} \right)^9$$

O argumento deve ser reduzido ao intervalo 1 a 10 i.e.

$$1 \leq x \leq 10$$

Os valores dos coeficientes utilizados são:

$$C_1 = .86859172$$

$$C_3 = .28933552$$

$$C_5 = .17752207$$

$$C_7 = .94376476$$

$$C_9 = .19133771$$

O valor do argumento pode ser uma variável previamente definida ou um número inteiro.

O argumento quando se trata de expressões deve ser colocado entre parênteses.

O erro E cometido na utilização desta aproximação é delimitado por $|E| \leq .000001$.

II. A função LN calcula o logaritmo natural de um número inteiro ou do valor de uma variável dada. Por razões de espaço de memória disponível não utilizamos um particular desenvolvimento em série ou aproximação polinomial; calculamos previamente o LOG e o resultado multiplicamos pelo valor de $\log_e 10$ ou seja:

$$LN(n) = LOG(n) * 2.3025851$$

III. A função DX calcula a potência de 10

$$DX(x) = 10^x$$

O argumento é o valor de uma variável previamente definida ou um número inteiro X sendo $0 \leq X \leq 9999$.

A função é calculada utilizando-se a aproximação de Hastings dada pela expressão:

$$10^x = [1 + a_1 x + a_2 x^2 + \dots + a_7 x^7]^2$$

$$0 \leq x \leq 1$$

Os valores dos coeficientes são:

$$\begin{aligned} a_1 &= 1.15129277603 \\ a_2 &= .66273088429 \\ a_3 &= .25434357484 \\ a_4 &= .07295173666 \\ a_5 &= .01742111988 \\ a_6 &= .00255491796 \\ a_7 &= .00093264267 \end{aligned}$$

O erro E cometido na utilização desta aproximação é delimitado por $|E| < .000000005$.

IV. A função EXP calcula a exponencial e^x .

Esta função é calculada através da expressão dada por Hastings (página 184)⁽⁶⁾

$$e^{-x} = \frac{1}{[1 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6]^4}$$

sendo

$$\begin{aligned} a_1 &= .2499986842 \\ a_2 &= .0312575832 \\ a_3 &= .0025913712 \\ a_4 &= .0001715620 \\ a_5 &= .0000054302 \\ a_6 &= .0000006906 \end{aligned}$$

$$0 \leq x \leq \infty$$

O erro E é delimitado por $|E| \leq .000002$.

O argumento X pode ser o valor de uma variável previamente definida ou um número inteiro qualquer ≤ 9999 .

V. A função RQ calcula a raiz quadrada de qualquer valor de uma variável previamente definida ou um número inteiro ≤ 9999 e positivo.

A raiz quadrada é calculada através de logaritmo e é dada pela expressão:

$$Y = DX (.5 * \text{LOG}(X))$$

VI. A função ATAN calcula o arco tangente, utilizando a aproximação de Hastings dada pela expressão:

$$Y = \arctg x = \frac{\pi}{4} + \sum_{i=0}^7 C_{2i+1} \left(\frac{x-1}{x+1} \right)^{2i+1}$$

$$0 \leq x < \infty$$

Os valores dos coeficientes utilizados são:

$$\begin{aligned} C_1 &= .99999993329 \\ C_3 &= .3332985605 \\ C_5 &= .1994653599 \\ C_7 &= .1390853351 \\ C_9 &= .0964200441 \\ C_{11} &= .0559098861 \\ C_{13} &= .0218612288 \\ C_{15} &= .0040540580 \end{aligned}$$

O erro cometido ao utilizar a aproximação é delimitado por $|E| < .00000004$.

VII. Funções Trigonômétricas

As funções trigonométricas que o comando calcula são o seno, coseno e tangente.

A função seno é calculada através da aproximação de Hastings dada pela expressão:

$$Y = \text{sen} \frac{\pi}{2} x = \sum_{i=0}^4 C_{2i+1} x^{2i+1}$$

$$-1 \leq x \leq 1$$

Com os seguintes valores dos coeficientes:

$$\begin{aligned} C_1 &= 1.57079631847 \\ C_3 &= .64596371106 \\ C_5 &= .07968967928 \\ C_7 &= .00467376557 \\ C_9 &= .00015148419 \end{aligned}$$

O erro E cometido na utilização desta aproximação é delimitado por $|E| < .000000005$.

Para calcular a função coseno utiliza-se a mesma expressão usando a relação de arco existente entre as duas funções trigonométricas

$$\cos(x) = \text{sen} \left(\frac{\pi}{2} - x \right)$$

De modo que dado o argumento em radianos o comando automaticamente acha o complemento do mesmo e calcula o coseno com a expressão dada.

O cálculo da tangente executa-se através da relação

$$\operatorname{tg} x = \operatorname{sen} x / \operatorname{cos} x$$

utilizando-se a expressão aproximada que calcula a função seno.

VIII A função ABS determina o valor absoluto de uma expressão qualquer

IX A função determinada pelo operador .** calcula a potência de um número dado ou do valor de uma variável previamente definida.

O expoente pode ser outra variável ou número inteiro. Portanto o expoente pode ser de ponto fixo ou flutuante.

Se o expoente é de ponto fixo e menor que 100 a potenciação é calculada por produto. Se é igual ou maior que 100 o cálculo efetua-se utilizando a relação seguinte:

$$Y = DX (B \cdot \operatorname{LOG} A)$$

para $Y = A^B$

Tomamos 100 como limite máximo para fazer por multiplicação, porque achamos que os valores comumente usados são inferiores a esse valor, e para valores superiores facilmente ocorrerá uma situação de "over-flow"; além disso o tempo de cálculo por iteração de produtos ainda é razoável com este número

3 3 3 – Descrição do Método Utilizado

Já dissemos que este comando realiza o cálculo de qualquer expressão aritmética que pode conter ou não uma ou várias das funções especificadas, seja isoladamente ou formando parte do argumento de outras funções.

Uma expressão aritmética, em ERP, é formada pela combinação de variáveis, números ou funções, símbolos e operadores, colocados à direita do símbolo de atribuição =. Do lado esquerdo deste símbolo deve sempre aparecer o nome de uma variável, previamente definida, indexada ou não.

Exemplo:

$$*CAL A 2 3 = A \cdot B 2 3 \cdot (C / F 2 \cdot A 3 \cdot \operatorname{LOG} (\operatorname{SEN} B))$$

A avaliação de uma expressão é realizada de acordo com as prioridades existentes na ordem de execução das operações. Existe, portanto, uma hierarquia de operações dada pelos símbolos de operações (operadores) e os parênteses.

Em geral, devem ser executadas em primeiro lugar as operações que estão encerradas entre parênteses e no caso de haver mais de um par de parênteses, as do mais interno. A seguir a potenciação, divisão, multiplicação, soma e subtração.

A programação para a execução das expressões aritméticas tal como se apresentam na notação de uso corrente, é complicada por causa dessas prioridades existentes.

Consegue-se uma grande simplificação transformando a notação normal em notação polonesa^(*), na qual as operações podem ser realizadas na ordem em que se apresentam da esquerda para a direita.

3.3.3.1 – Na notação polonesa os operadores são escritos a direita (ou esquerda) de um par de operandos e não no meio, formando uma cadeia.

Notação algébrica normal $(X - Y) * (A - B)$

Notação polonesa $XY - AB - *$

As regras para avaliação de uma cadeia polonesa são:

- 1) Percorre-se a cadeia da esquerda para a direita.
- 2) Guardam-se os operandos na ordem em que aparecem.
- 3) Quando aparece um operador procede-se assim:
 - a) Tomam-se os dois últimos operandos,
 - b) Opera-se sobre eles de acordo com o tipo de operador encontrado,
 - c) Guarda-se o resultado da operação eliminando-se ambos operandos.
 - d) Considera-se este resultado como o último operando da cadeia.
- 4) Continua-se o percurso até atingir o delimitador final.

Um exemplo da notação polonesa:

$$a = bc * e f ** d * +$$

Aplicando as regras anteriores obtém-se a correspondente notação algébrica comum na forma como detalhamos abaixo.

Os parênteses indicam que a operação é guardada provisoriamente, constituindo um operando.

a
 a =
 a = b
 a = bc
 a = bc *
 a = (b * c) (primeira operação)
 a = (b * c) e
 a = (b * c) ef
 a = (b * c) ef **
 a = (b * c) (e ** f)
 a = (b * c) (e ** f) d

(*) Notação introduzida em 1921 na Polónia por Jan Łukasiewicz.

$$\begin{aligned}
 a &= (b * c) (e ** f) d * \\
 a &= (b * c) ((e * f) * d) \\
 a &= (b * c) ((e * f) * d) + \\
 a &= ((b * c) + ((e ** f) + d) \\
 a &= b * c + e ** f + d
 \end{aligned}$$

que colocado em forma algébrica será:

$$a = b . c + e^f . d$$

3.3.3.2 – Conversão da Notação Normal em Polonesa

Para se passar da notação normal à notação polonesa direta com os operadores imediatos, deve-se estabelecer uma prioridade dos operadores. Em geral adota-se a seguinte:

Tabela 3.1

Prioridade	Operador
1	≠ (*)
2)
3	+ -
4	* /
5	**
6	Funções
7	(
8	b

Estabelecidas as prioridades segue-se o seguinte algoritmo:

Seja um comando composto dos símbolos S_1, S_2, \dots, S_n vamos produzir uma seqüência de símbolos P_1, P_2, \dots, P_k na notação polonesa.

Vamos empregar uma tabela TEO (Table of encountered operators) organizada no sistema LIFO (Last in first out) onde colocamos os operadores e os dois parênteses, chamando-os e_1, e_2, \dots, e_m .

Sejam S_j os símbolos a serem examinados; e_m o último operador da tabela TEO e P_k o próximo símbolo a ser gerado em notação polonesa.

- 1) Se S_j for um símbolo primário $\langle pr \rangle$ fazemos $P_k = S_j$.
- 2) Se S_j for um "(" , fazemos $e_{m+1} = S_j$ i.e. colocamos o abre parênteses na tabela TEO sem mais.
- 3) Se S_j for um operador comparamo-lo com e_m

(*) Este símbolo é utilizado como delimitador final.

- a) Se S_j tem prioridade maior ou igual à de e_m fazemos $P_k = e_m$ i.e. geramos P_k como último símbolo da tabela e comparamos S_j com e_{m-1} .
- b) Se S_j tem prioridade menor que e_m fazemos $e_{m+1} = S_j$.
- 4) Se S_j for "(" todos os símbolos da TEO são transferidos para a saída até que apareça um ")" i.e. $P_k = e_m, P_{k+1} = e_{m-1}, \dots, P_{k+q} = e_{m-q}$ sendo $e_{m-q} = "("$. Cancelam-se os "(" e ")" da TEO.
- 5) Se $S_j \neq \neq$ terminamos com os símbolos do comando e fazemos

$$P_k = e_m, P_{k+1} = e_{m-1}, \dots, P_{k+m-1} = e_1.$$

O operador unário "+" (quando segue ao = ou abre parênteses) não é considerado, enquanto que o operador unário "-" é considerado como uma função.

Assim a expressão algébrica

$$a (b + c + d) / (e + f)^2$$

que expressa em notação adequada de computação fica:

$$A * (B + C + D) / (E + F) **2$$

Aplicando as regras anteriores, transforma-se na seguinte notação polonesa:

$$A B ^ D + * E F + 2 ** / \sim$$

O símbolo \sim representa o operador unário ou prefixo "-".

As tabelas mencionadas ficam preenchidas assim: (página seguinte)

3.3.3.3 -- Aplicação no *CAL

No caso do comando calcule (*CAL) utilizamos um sistema similar à notação polonesa, um pouco modificada na sua fase de execução. Estabelecemos basicamente as prioridades especificadas na tabela 3.1., e para facilitar mais ainda a programação, damos diferentes prioridades aos operadores +, -, *, e /, ou seja utilizaremos a tabela 3.3.

Para o cálculo de uma expressão, dividiremos em duas fases: a fase de análise da instrução e a de execução.

a) Fase de Análise

Na fase de análise, verificamos os possíveis erros existentes seja na sintaxe, seja na consistência de indexação das variáveis, seja no balanceamento de parênteses, seja na validade dos parâmetros utilizados; determinamos a natureza dos parâmetros (variável, número ou função) e construímos duas

Tabela 3.2

j	S_j	TEO e_m	P_k
1	~	b	b
2	A	~	A
3	*	~	A
4	(~*	A
5	B	~* (AB
6	+	~* (AB
7	C	~* (+	ABC
8	+	~* (ABC+
9	D	~* (+	ABC + D
10)	~*	ABC + D +
11	/	~*	ABC + D + *
12	(~/	ABC + D + *
13	E	~/ (ABC + D + * E
14	+	~/ (ABC + D + * E
15	F	~/ (+	ABC + D + * EF
16)	~/	ABC + D + * EF +
17	**	~/	ABC + D + * EF +
18	2	~/**	ABC + D + * EF + 2
19	≠	b	ABC + D + * EF + 2 ** / ~

Tabela 3.3

Prioridade	Operador
1	≠
2)
3	-
4	+
5	*
6	/
7	**
8	Funções
9	(

tabelas. Uma tabela P_k onde colocamos os valores das variáveis e os números que aparecem na expressão, e outra tabela com as prioridades. Esta tabela de prioridades está formada por números de 4 dígitos dos quais os dois primeiros da esquerda especificam a ordem relativa que ocupa cada operador na expressão. Isto servirá para identificar, caso aconteça um erro durante a fase de execução, a posição do parâmetro onde se verificou tal erro. Os dois dígitos da direita dão a prioridade de todos os operadores e funções, e para estas últimas dão o endereço de entrada à sub-rotina que executa cada uma das funções. Este endereço de entrada é dado pela diferença entre os dois últimos dígitos e a prioridade correspondente. Todas as funções possuem igual prioridade.

b) Fase de Execução

Construídas as tabelas de valores P_k e de prioridades entra em função a fase de execução

Percorremos a tabela de prioridade dos operadores comparando um com outro. Quando achamos que a prioridade do 1º é maior que a do segundo executamos a operação correspondente utilizando os valores da tabela P_k . A seguir comprimimos a tabela deslocando todos os valores e prioridades, ficando em correspondência o valor calculado e o operador seguinte. Retrocedemos e comparamos novamente. Se a prioridade atual é menor que a seguinte continuamos em frente.

Se encontramos um "(" seguimos em frente e executamos as operações indicadas de acordo com as prioridades até atingir um ")". Isto é indicação de que devemos executar todas as operações indicadas desde o último "(" até o correspondente ")" e cancelamos esse par de parênteses comprimindo a tabela tantas vezes quantas forem necessárias. Retrocedemos um passo e continuamos com as comparações, execuções e compressão da tabela até atingir a prioridade 1 indicadora de fim de instrução.

Quando achamos uma prioridade correspondente a uma função (todas possuem a mesma prioridade) determinamos o endereço da mesma, na forma já indicada, e transferimos o controle da execução à subrotina correspondente. Terminada a execução da mesma devolvemos o controle ao programa de execução principal para continuar com as comparações até o fim.

Se acontece qualquer erro, como a tentativa de dividir por zero, imediatamente será determinado o lugar relativo ocupado pelo parâmetro na instrução, na forma já indicada. Esta informação é transferida à rotina de erro que imprimirá a mensagem correspondente.

3.3.4 – Descrição dos Subprogramas ERP6 e ERP7

A execução do comando "CALCULE" utiliza dois subprogramas escritos em FORTRAN II. Um subprograma, o ERP6, realiza a análise da instrução e outro subprograma, ERP7, se encarrega da fase de execução das funções. A execução das operações aritméticas, assim como das funções "-" monádico e a determinação do valor absoluto ficam dentro do ERP6. Desta forma qualquer outro ERPn que deseje utilizar os cálculos das outras funções pode fazê-lo simplesmente chamando o ERP7, dentro do seu próprio ERPn.

Para dar uma descrição resumida do funcionamento de ambos. (ERP6 e ERP7) seguiremos o diagrama de blocos simplificado das páginas 44 e 45.

3.3.4.1 – Fase da Análise

Inicia com a verificação da posição 4 de IND4 para ver se encontra ligada ou desligada. Caso estiver desligada quer dizer que deve executar a análise de uma nova instrução de cálculo, então zera a tabela P_k e posiciona o ponteiro I no terceiro parâmetro da instrução. Verifica se na posição I - 1 existe o indicador de fim de instrução (@ = 3400). Se existe devolve o controle ao Interpretador, se não, verifica se nessa posição tem um branco, caso não tenha é um erro, porque a seguir deve vir o nome de

uma variável previamente definida; se a resposta à verificação é positiva entra em ação a subrotina que analisa uma variável através da subrotina PEGUE determinando os valores dos seus índices e a região da área comum de dados, AC, ocupada por seus valores. Verifica a consistência de indexação e avança o ponteiro dois parâmetros adiante. Verifica se em I existe o sinal "=". Caso não existir será considerado erro, transferindo o controle ao ERP4 que emitirá a mensagem correspondente do diagnóstico realizado.

Avança dois parâmetros verificando se I-1 contém um ponto. Se é sim, entra em função a subrotina que analisa se I está apontando uma função ou um operador e preenche a tabela de prioridade. Se em I-1 não for ponto, verifica se é branco, em cujo caso chama a subrotina que analisa a variável e preenche a tabela P com o valor correpondente da variável. Se não é branco verifica se é um número ($\$=1300$), caso seja sim desvia para a subrotina que analisa número, preenchendo com o valor achado a tabela P. A seguir avança e verifica se já chegou ao final da instrução, se não, continua com a análise para preencher as tabelas P e de prioridade.

Se atingiu ao fim da instrução passa à fase de execução. No caso em que durante a análise verificou-se a não existência de um identificador de variável (b), número (\$), literal (.) (operador ou função) ou fim, transferirá o controle ao ERP4 para o diagnóstico correspondente.

3.3.4.2 – Fase de Execução

Durante a fase de execução percorre as tabelas P e de prioridade verificando que operação deve executar. Se deve executar operações aritméticas (exceto potenciação), valor absoluto ou prefixo (-) desvia para a subrotina que toma conta da execução no próprio ERP6. A seguir comprime ambas as tabelas e continua com a execução. Quando deve executar funções (incluindo potenciação) determina o endereço da função mediante os dois últimos dígitos da tabela de prioridade, o coloca na área comum do próprio programa e transfere o controle ao ERP7. Este subprograma com o endereço fornecido pelo ERP6 executa a função correspondente, modifica o estado de IND4 ligando a posição 2, e devolve o controle ao ERP6 onde se produz a compressão das tabelas, continuando com a execução até atingir o indicador de fim de instrução quando devolve o controle ao Interpretador, dando por terminada a sua ação.

Se ao entrar ERP6 o valor de IND4 fosse maior ou igual a 1000, i.e posição 4 ligada, significaria que não era a primeira entrada no subprograma e que esta nova chamada era para comprimir ambas as tabelas.

3.3.5 – Exemplo das Tabelas P_k e de Prioridade

CAL A = B .- 5 .+ LOG (2 . 3 .) .-(A 2 3 .** 2 .)+SEN .(PI ./ 6 .)

$P_{(k)}$	Prioridade
.35100000E+03	103
.50000000E+01	204
.00000000E-99	309
.00000000E-99	419
.20000000E+01	506
.30000000E+01	602
.00000000E-99	703
.00000000E-99	819
.35500000E+03	908
.20000000E+01	1002
.00000000E-99	1104
.00000000E-99	1216
.00000000E-99	1319
.31415927E+01	1405
.60000000E+01	1502
.00000000E-99	1

Percorre-se, então, a tabela de prioridade comparando os dois últimos dígitos de cada valor da tabela com os dois últimos dígitos seguintes. Se estes forem menores executa-se a operação correspondente à prioridade achada em primeiro lugar, coloca-se o resultado na tabela P_k na altura correspondente à prioridade mais alta e comprime-se a tabela; depois retrocede de uma linha e continua com a comparação até atingir a prioridade 1.

Na 4ª linha do quadro acima encontra-se com uma prioridade maior que a seguinte, mas não executa operação alguma por tratar-se de um abre parênteses. Segue-se com a comparação, como foi explicado no parágrafo 3.3.3.3 fase de execução.

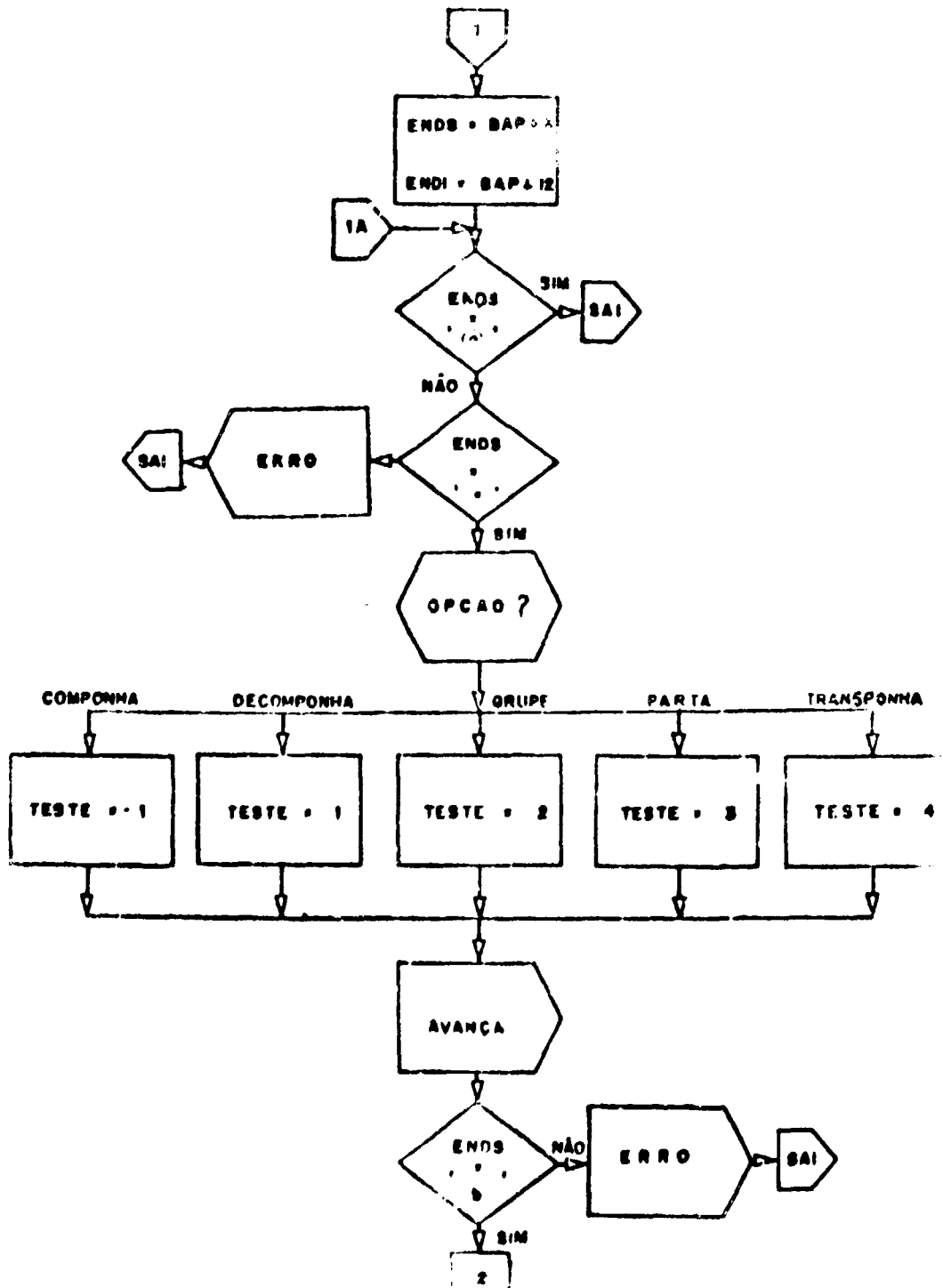
No exemplo apresentado acima, a primeira operação a ser executada corresponderá à prioridade 06 que ocupa o lugar 5 porque a seguir vem uma prioridade 2 ocupando o lugar 6, então multiplicar-se-á o valor 2 da quinta linha pelo valor 3 da 6ª linha e o resultado colocar-se-á na 5ª linha e a seguir comprimem-se as tabelas. Como a prioridade 02 da linha 6 corresponde a um ")" e da linha 4 a "(", conseqüentemente comprimir-se-á mais duas vezes as tabelas ficando a prioridade 03 da linha 7 na linha 5. Prossegue-se desse jeito até executar as operações indicadas.

Após a primeira execução e as compressões correspondentes a tabela ficará assim:

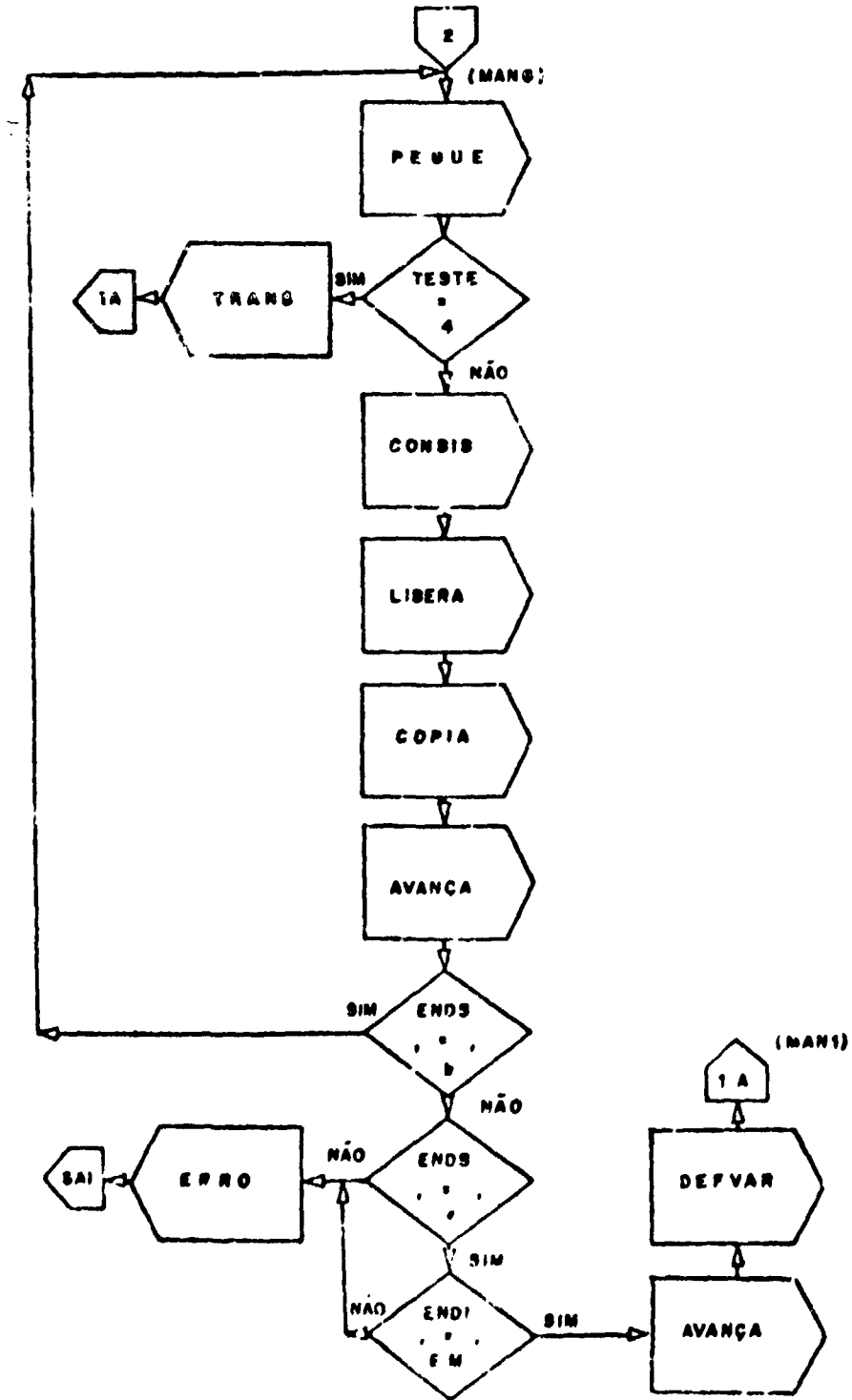
$P_{(k)}$	Prioridade
.35100000E+03	103
10000000E+01	204
.00000000E-99	309
.00000000E-99	419
.60000000E+01	703
.00000000E-99	819
.35900000E+03	908
.20000000E+01	1002
.00000000E-99	1104
.00000000E-99	1216
.00000000E-99	1319
.31415927E+01	1405
.60000000E+01	1502
.00000000E-99	1

Nas páginas 71 e 72 do capítulo V temos vários exemplos aplicativos do comando *CAL na resolução de expressões aritméticas.

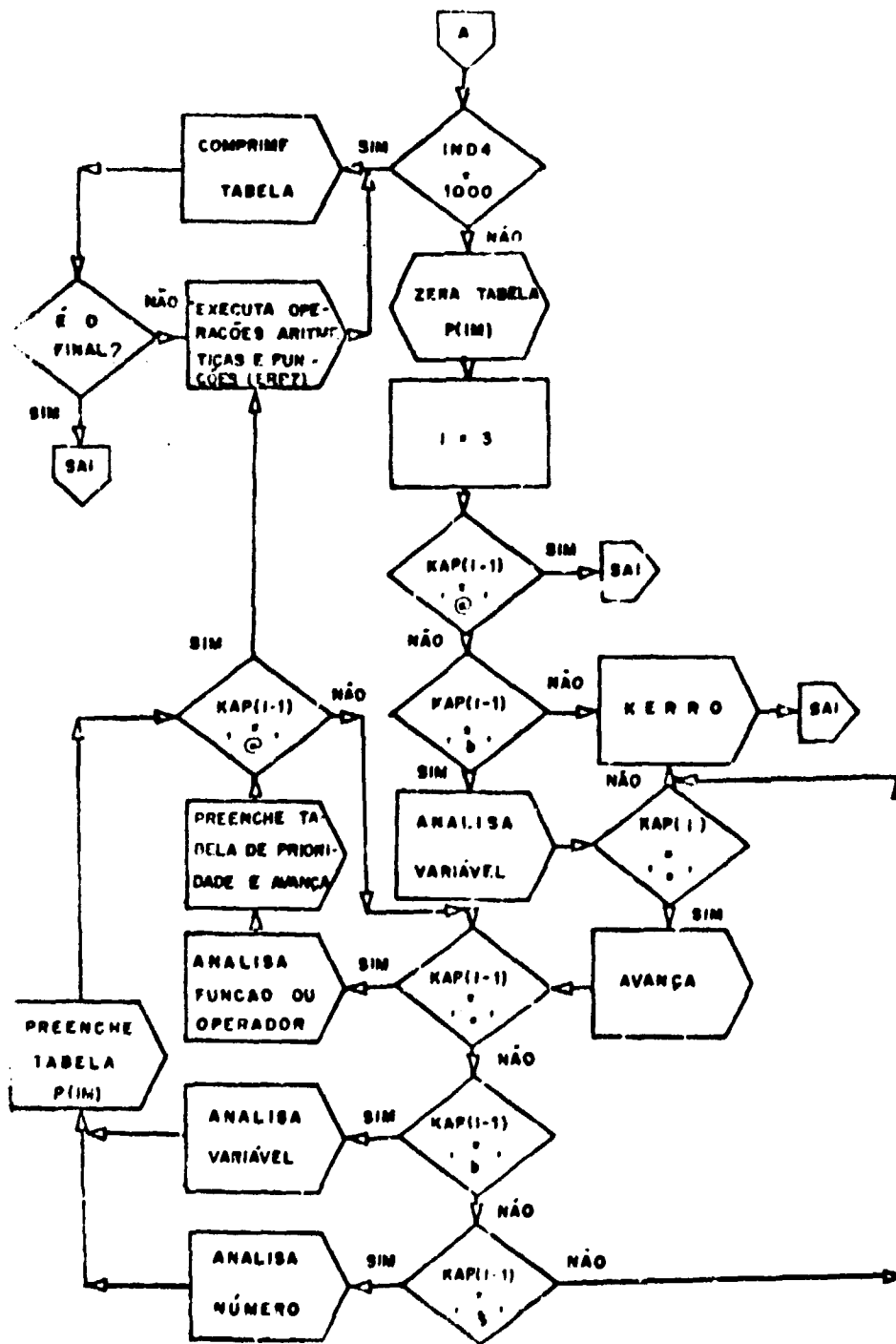
COMANDO *MAE - DIAGRAMA SIMPLIFICADO ERPE



COMANDO *MAN - DIAGRAMA SIMPLIFICADO ERPS (cont.)

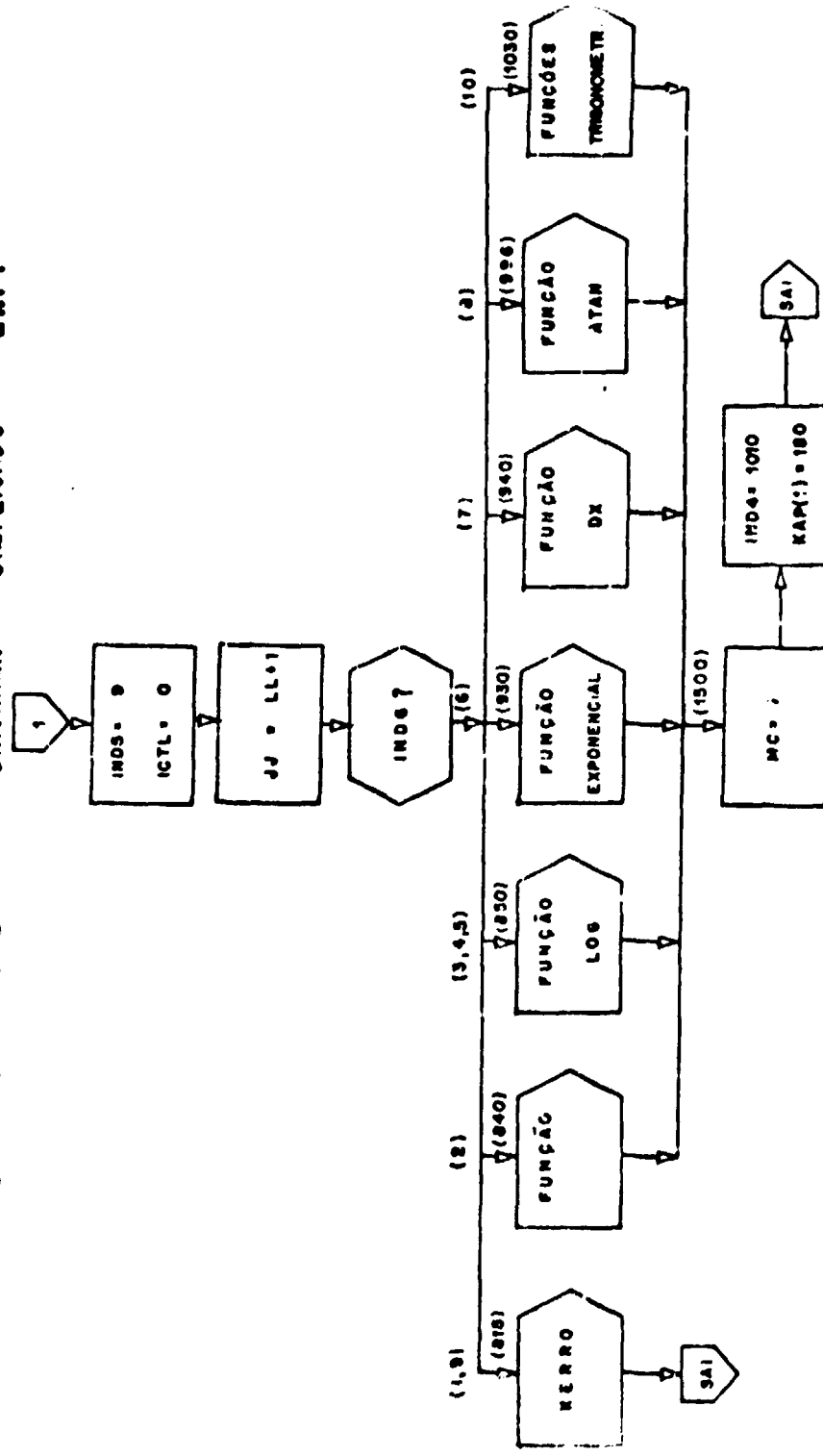


COMANDO *CAL - DIAGRAMA SIMPLIFICADO - ERP6



FASE DE ANALISE

COMANDO -CAL DIAGRAMA SIMPLIFICADO ERP7



FASE DE EXECUÇÃO

CAPÍTULO IV

COMANDOS DO USUÁRIO, IMPLEMENTADOS

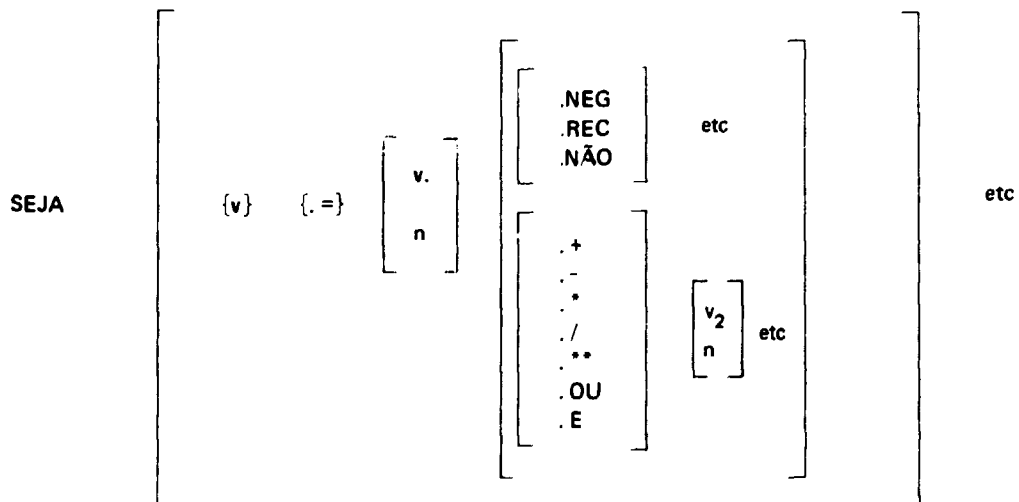
Os comandos do usuário, implementados, foram os seguintes:

SEJA
TRAP
PLOT
MATRIZ
SOLUÇÃO
AJUSTE
LOCR

4.1 – Comando SEJA (ERP8)

Este comando serve para realizar cálculo com matrizes, elemento a elemento.

4.1.1 – Descrição Formal



Obs.: Todas as matrizes em consideração devem ter as mesmas dimensões.

4.1.2 – OPÇÃO .NEG

Exemplo:

SEJA A .= B.NEG

Execução:

A matriz B é copiada na matriz A e são trocados de sinal todos os elementos de A. B permanece inalterada.

4.1.3 – OPÇÃO .REC

Exemplo:

SEJA A. = B .REC

Execução:

A matriz B é copiada na matriz A e os elementos de A são substituídos pelos elementos da matriz recíproca.

4.1.4 – OPÇÃO .+

Exemplo:

SEJA A. = B .+ C

Execução:

A matriz B é copiada em A e os elementos da matriz C são somados aos elementos da matriz A ficando a soma na matriz. As matrizes B e C permanecem inalteradas.

4.1.5 – OPÇÃO .-

Exemplo:

SEJA A. = B .- C

Execução:

A matriz B é copiada em A e os elementos da matriz C são subtraídos dos elementos da matriz A e os resultados substituem os elementos da A ficando B e C inalteradas.

4.1.6 – OPÇÃO .*

Exemplo:

SEJA A. = B .* C

Execução:

A matriz B é copiada na matriz A e os elementos da matriz A são multiplicados pelos respectivos elementos da matriz C ficando os produtos como novos elementos da matriz A. A matriz B e C permanecem inalteradas.

4.1.7 – OPÇÃO ./

Exemplo:

SEJA A. = B ./ C

Execução:

A matriz B é copiada na matriz A e os elementos da matriz A são divididos pelos respectivos elementos da matriz C ficando o quociente como novos elementos da matriz A. As matrizes B e C permanecem inalteradas.

4.1.8 – OPÇÃO .****Exemplo:**

SEJA A .= B.** C

Execução:

A matriz B é copiada na matriz A e os elementos da matriz A são elevados à potência dada pelo respectivo elemento da matriz C. A potência calculada de cada elemento é o novo elemento da matriz A. As matrizes B e C permanecem inalteradas.

4.1.9 – OPÇÃO .NÃO**Exemplo:**

SEJA A .= B.NÃO

Execução:

A matriz B é copiada em A e cada elemento da matriz A é comparada com zero. Se é igual é substituído por 1; caso contrário é substituído por zero. A matriz B permanece inalterada.

4.1.10 – OPÇÃO .OU**Exemplo:**

SEJA A .= B.OU C

Execução:

A matriz B é copiada em A e cada um dos elementos das matrizes A e C são comparados com zero. Se ambos são zeros então o valor do elemento de A é feito igual a 0 e se um deles é diferente de zero, o elemento correspondente de A é substituído por 1.

4.1.11 – OPÇÃO .E**Exemplo:**

SEJA A .= B.E C

Execução:

A matriz B é copiada em A e cada elementos de A e C é comparado com zero. Se ambos são diferentes, o valor do correspondente elemento de A é feito igual a 1 e se um deles for zero o elemento de A é substituído por zero.

4.1.12 – OPÇÃO n

Exemplo:

SEJA A .= 20

Execução:

Todos os elementos de A são feitos iguais a 20. Se A é um escalar, o seu valor será trocado pelo valor de n.

Em geral, em qualquer das opções mencionadas anteriormente o valor da operação executada com valor de n substitue os elementos da matriz.

4.1.13 – OPÇÃO v

Exemplo:

SEJA A .= B

Execução:

Os elementos da matriz B são copiados na matriz A. Esta opção serve para guardar o valor de uma variável (matriz, vetor ou escalar) que provavelmente será modificada durante uma operação, mas cujo valor original deseja-se conservar.

obs: Em todos dados mencionados anteriormente, as matrizes devem ser definidas previamente.

4.2 – COMANDO TRAP (ERP9)

Este comando produz a partir de um polinômio dado outro polinômio cujas raízes guardam com as raízes do polinômio original uma relação especificada.

4.2.1 – Descrição formal

TRAP	[.TRANSLADE v_1 v_2 .MULTIPLIQUE v_1 v_2 .NEGUE v .RECIPROQUE v .QUADRE v]	etc
------	---	--	---	-----

4.2.2 – OPÇÃO .TRANSLADE

Exemplo:

TRAP .TRANSL P ALFA

Execução:

Realiza-se a translação ALFA das raízes do polinômio P.

4.2.3 – OPÇÃO .MULTIPLIQUE

Exemplo:

TRAP .MULTIP P ALFA

Execução:

As raízes do polinômio P são multiplicadas por ALFA.

4.2.4 – OPÇÃO .NEGUE

Exemplo:

TRAP .NEG P

Execução:

As raízes do polinômio P são as raízes anteriores com os sinais trocados.

4.2.5 – OPÇÃO .RECIPROQUE

Exemplo:

TRAP .RECIPR P

Execução:

As raízes do polinômio P são transformadas de forma que as novas raízes resultantes são as recíprocas das raízes do polinômio original.

4.2.6 – OPÇÃO .QUADRE

Exemplo:

TRAP .QUADRAT P

Execução:

As raízes do polinômio P são transformadas de forma que as novas raízes são o quadrado das raízes do polinômio original.

4.2.7 – Exemplo:

TRAP .TRANSL P ALFA .QUADRAT P

Execução:

Em primeiro lugar as raízes do polinômio P são trasladadas de ALFA e a seguir estas novas raízes são substituídas pelos seus quadrados respectivos.

4.3 – COMANDO LOCR (ERP 10)

Este comando localiza as raízes de um polinômio.

4.3.1 – Descrição formal

LOCR { v } etc

Este comando possui uma única opção.

Exemplo:

LOCR P

Execução:

A execução deste comando fornece vários dados de interesse para achar as raízes do polinômio, que são:

- 1) Determinar quantas são as raízes que têm partes reais positivas e quantas têm partes reais negativas.
- 2) Determinar as raízes complexas.
- 3) Informar se o polinômio é constante.
- 4) Fornecer informação de divisor nulo durante o processo do algoritmo seguido.

A seguir damos uma descrição detalhada do algoritmo utilizado, por considerá-lo de interesse, devido à forma prática que é realizada a localização das raízes.

Seja o polinômio

$$P = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \dots + a_{n-2} x^2 + a_{n-1} x + a_n$$

que podemos escrever

$$P = P_n + P_{n-1}$$

onde P_n representa o polinômio formado pelos coeficientes de ordem par de P e P_{n-1} o formado pelos coeficientes de ordem ímpar de P .

Multiplicamos os coeficientes de ordem par pelo quociente $q_1 = -\frac{a_0}{a_1}$ e os resultados somamos aos coeficientes de ordem ímpar. Assim o primeiro coeficiente será nulo e obtemos os coeficientes de um novo polinômio de grau $(n - 1)$ ao qual aplicamos o mesmo processo.

Obtemos então:

$$\begin{aligned} P_{n-2} &= P_n - q_1 \times P_{n-1} \\ &\vdots \\ P_{n-3} &= P_{n-1} - q_2 \times P_{n-2} \\ &\vdots \\ P_{n-4} &= P_{n-2} - q_3 \times P_{n-3} \\ &\vdots \end{aligned}$$

$$\begin{aligned}
 & \vdots \\
 & \vdots \\
 P_4 &= P_6 - q_{n-5} \times P_5 \\
 P_3 &= P_5 - q_{n-4} \times P_4 \quad (1) \\
 P_2 &= P_4 - q_{n-3} \times P_3 \\
 P_1 &= P_3 - q_{n-2} \times P_2 \\
 P_0 &= P_2 - q_{n-1} \times P_1
 \end{aligned}$$

Depois de repetido n v\u00e9zes este processo, chega-se a uma constante e tem-se os n valores de q_j .

Pode-se demonstrar que: "os sinais dos quocientes q_j coincidem com os sinais das partes reais das ra\u00edzes".

Em particular ser\u00e3o negativos todas as partes reais se todos os q_j forem negativos. Esta determina\u00e7\u00e3o \u00e9 de grande import\u00e2ncia em quest\u00f5es de estabilidade de vibra\u00e7\u00f5es.

No processo descrito acima interessa especialmente os coeficientes do polin\u00f4mio de 2\u00b0 grau que aparece na redu\u00e7\u00e3o. Este polin\u00f4mio pode ser escrito:

$$k_2 x^2 + k_1 x + k_0 = 0$$

onde:

$$P_2 = k_2 x^2 + k_0$$

e

$$P_1 = k_1 x$$

Se $k_1 = 0$ ent\u00e3o $P_1 = 0$

Portanto em (1) teremos:

$$P_3 - q_{n-2} \times P_2 \equiv 0$$

isto implica que P_2 \u00e9 divisor de P_3

isto \u00e9:

$$P_2 \mid P_3 \quad \text{o que implica que}$$

$$P_2 \mid P_4$$

\vdots

$$P_2 \mid P_{n-1}$$

$$P_2 \mid P_n$$

mas $P = P_n + P_{n-1}$ então

P_2 é divisor de P

Portanto a equação de partida têm as raízes:

$$x = \pm i \sqrt{\frac{k_0}{k_2}}$$

Então o que procuramos é fazer $k_1 = 0$, podemos conseguir-lo fazendo uma translação do polinômio dado e aplicando em cada caso o processo de iteração anterior. Assim, podemos delimitar as raízes do polinômio, considerando os sinais dos q_j de cada translação, e construindo um gráfico para as diversas aproximações.

Para melhor ilustração do processo de redução descrito damos a seguir um exemplo prático;

$$P = x^4 - x^3 - 2x + 3$$

Escrevemos os coeficientes da expressão completa numerando-os da esquerda para a direita para estabelecer a ordem dos mesmos.

Determinamos $q_1 = -\frac{1}{-1} = 1$ que é multiplicado pelos coeficientes de ordem par e somado aos coeficientes de ordem ímpar.

Ordem	1	2	3	4	5	
Coeficientes	1	-1	0	-2	3	P_3
	-1		-2			$q_1 = 1$
Novos Coeficientes	0	-1	-2	-2	3	
Ordem	1	2	3	4		
Coeficientes	-1	-2	-2	3		P_2
	1		-1,5			$q_2 = -1/2$
	0	-2	-3,5	3		P_1
		2				$q_3 = -4/7$
		0	-3,5	3		P_0
			3,5			$q_4 = 7/6$
			0	3		constante

4.4 – COMANDO PLOT (ERP 11)

Este comando serve para se obter pela impressora um gráfico de uma função dada por pontos.

4.4.1 – Descrição formal

PLOT { v_1 v_2 } etc

Exemplo:

PLOT X Y

Execução:

Obtém-se um gráfico, pela impressora, na qual as abcissas são dadas pelos elementos do vetor X e as ordenadas pelos correspondentes elementos do vetor Y. Ambos vetores devem ser definidos previamente.

4.5 – COMANDO MATRIZ (ERP 12)

Este comando executa operações com matrizes. Exemplos das diferentes opções possíveis saídas pela impressora, estão dados nas pags 83 a 87 do cap. V.

4.5.1 – Descrição formal

MATRIZ	.SOME v_1 v_2 v_3 .SUBTRAI v_1 v_2 v_3 .MULTIPLIQUE v_1 v_2 v_3 .DETERMINANTE v_1 v_2 .TRANSPONHA [v v_2] etc .INVERTA v	etc
--------	--	-----

4.5.2 – OPÇÃO .SOME

Exemplo:

MATRIZ .SOMA A B C

Execução:

Soma a matriz A com a matriz B e o resultado é guardado em C. As três matrizes devem estar definidas previamente.

4.5.3 – OPÇÃO .SUBTRAI

Exemplo:

MATRIZ .SUBTRAI A B D**Execução:**

A matriz B é subtraída da matriz A e o resultado é guardado em D. As três matrizes devem estar definidas previamente.

4.5.4 – OPÇÃO .MULTIPLIQUE**Exemplo:****MATRIZ .MULT A B E****Execução:**

A matriz A é multiplicada pela matriz B e o resultado é guardado em E.

4.5.5 – OPÇÃO .DETERMINANTE**Exemplo:****MATRIZ .DETER A F****Execução:**

É calculado o valor do determinante da matriz A e o resultado é guardado no escalar F. Tanto A quanto F devem ser definidos previamente.

4.5.6 – OPÇÃO .TRANSPONHA**Exemplo:****MATRIZ .TRANSP B G****Execução:**

A matriz B é transposta e o resultado é guardado na matriz G. Ambas as matrizes devem ser definidas previamente.

4.5.7 – OPÇÃO .INVERTA**Exemplo:****MATRIZ .INVERTA A****Execução:**

É determinada a matriz inversa de A ficando o resultado em A. A deve ser uma matriz previamente definida.

4.6 – COMANDO SOLUÇÃO (ERP 13)

Este comando resolve um sistema de N equações lineares a M incógnitas verificando se o sistema é compatível ou não, pelo estudo do posto da matriz.

4.6.1 – Descrição formal

SOLUÇÃO [{ v₁ } { v₂ } { v₃ }] etc

4.6.2 – Exemplo:

SOLUÇÃO A X B

Execução:

Determina-se o vetor da solução X do sistema cuja matriz dos coeficientes é A e o vetor dos termos conhecidos é B. O comando testa a compatibilidade do sistema. Verifica a compatibilidade pelo estudo do posto da matriz e resolve o sistema pelo método de triangularização. Se é compatível e o posto é igual ao número de incógnitas há um único vetor solução. Se a característica é menor que o número de incógnitas o conjunto das soluções do sistema homogêneo formam um sub-espço Rⁿ de dimensão igual à diferença entre o número de incógnitas e o posto. O sistema não homogêneo admite pelo menos um vetor solução X₀. O conjunto das soluções do sistema não homogêneo é obtido somando X₀ às soluções do sistema homogêneo. Se B = 0 o vetor X₀ poderá ser tomado como X₀ = 0.

Durante o processamento das operações aritméticas, quando da subtração de duas quantidades muito próximas apresenta-se o problema de perda de precisão pela introdução de zeros a esquerda.

Na solução de sistemas lineares por triangularização aparece muitas vezes aquele problema, pela utilização da expressão $a_{ij} = a_{ij} - a_{ik} a_{kj}$. Para isto, o subprograma que executa o comando, testa o resultado da subtração e se a diferença é muito menor que o número que está sendo subtraído em valor absoluto, faz a diferença igual a zero. Considera suficiente a perda de mais de 4 algarismos significativos na subtração, i.e. o valor da diferença menor que .0001 vezes o valor do produto $a_{ik} a_{kj}$ para fazer $a_{ij} = 0$.

4.7 – COMANDO AJUSTE (ERP 14)

Este comando serve para ajustar curvas por polinômios ortogonais, utilizando o critério dos mínimos quadrados, para um conjunto de pontos. Os pontos são dados pelas suas coordenadas, (X, Y).

O método utilizado é o recomendado por Forsythe (que é válido para pontos não igualmente espaçados e leva em conta o peso de cada um).

O método de Forsythe pode ser resumido do modo seguinte:

Procura-se encontrar os coeficientes $a_i (i = 1, 2, \dots, N)$ e os polinômios $P_i(x)$ tais que

$$y = a_1 P_1(x) + a_2 P_2(x) + \dots + a_n P_n(x)$$

Para isso deve-se:

1º Fazer $P_1 = 1$

2º Fazer $P_2 = x - c$

sendo

$$c = \frac{\sum_{i=1}^M x_i w(x_i)}{\sum_{i=1}^M w(x_i)}$$

onde $M = N - 1$ é o grau do polinômio ajustado e $w(x_i)$ é o fator peso para x_i .

3º Calcular cada polinômio usando a relação

$$P_{N+1} = (x - B) P_N + G P_{N-1}$$

onde,

$$B = \frac{\sum_{i=1}^M x_i (P_N)^2 w(x_i)}{\sum_{i=1}^M (P_N)^2 w(x_i)}$$

$$G = \frac{\sum_{i=1}^M x_i P_N P_{N-1} w(x_i)}{\sum_{i=1}^M (P_{N-1})^2 w(x_i)}$$

4º Calcular os coeficientes a_N usando

$$a_N = \frac{\sum_{i=1}^M y_i P_N}{\sum_{i=1}^M (P_N)^2}$$

4.7.1 – Descrição formal

AJUSTE $\left[\begin{array}{l} \text{.COMPESO} \quad \{v_1 \ v_2 \ v_3 \ v_4\} \\ \text{.SEMPESO} \quad \{v_1 \ v_2 \ v_3\} \end{array} \right]$ etc

4.7.2 – OPÇÃO .COMPESO

Exemplo:

AJUSTE COMPESO A X Y W

Execução:

O conjunto de pontos dados pelas coordenadas (X , Y) e com peso W é ajustado por um polinômio. Os coeficientes do polinômio são guardados em A. O grau do polinômio é dado pelo número de componentes de A menos 1.

4.7.3 – OPÇÃO .SEMPESO

Exemplo:

AJUSTE .SEMPESO A X Y

Execução:

Similar a opção .COMPESO exceto que o peso W = 1.

Obs. O programa além de guardar os coeficientes, imprime pelo dispositivo de saída selecionado (Impressora ou máquina de escrever) o grau do polinômio que melhor se ajusta à curva, os coeficientes e as potências correspondentes.

Todas as variáveis da opção devem estar definidas previamente.

CAPÍTULO V

EXEMPLOS DE APLICAÇÕES

Neste capítulo damos alguns exemplos de aplicação dos comandos implementados; tanto dos comandos do sistema quanto dos comandos dos usuários.

5.1 – No caso do comando *MANIPULE utilizamos as instruções dadas no capítulo III quando da exemplificação das diferentes opções do comando. As instruções foram registradas e posteriormente executadas. Aproveitamos a execução dos exemplos para mostrarmos, mediante o comando *E/S, a situação das diferentes tabelas do sistema e da área comum de dados após a execução de cada comando *MAN.

As instruções iniciais foram introduzidas pela máquina de escrever e mediante um comando de controle modificamos o dispositivo de entrada para cartões e quando da execução destes modificamos o dispositivo de saída para impressora e o modo para RG e posteriormente para PR. Com a execução da última instrução registrada retornamos à situação inicial dos dispositivos de entrada e saída e do modo.

A execução do programa registrado foi realizada instrução por instrução, ligando a chave 2 de console. Desta forma cada instrução é impressa antes da execução.

5.2 – Para exemplificar o comando *CAL resolvemos casos práticos apresentados pelos usuários de Divisões do I.E.A.

Nas páginas 71 a 75 temos como exemplo o problema apresentado por um Pesquisador do Serviço de Produção de Material Radioativo, que não tinha suficiente prática em programação. Os cálculos são simples mas os casos apresentados são vários e resulta mais prático tentar resolvê-los pelo uso do computador.

5.2.1 – O problema é o seguinte:

Calcular a atividade de todos os isótopos do Telúrio que se formam por irradiação de 50 grs. de telúrio natural durante 2 semanas interruptas, sendo na 1ª semana irradiado 8 hrs/dia durante 5 dias e na 2ª semana 8 hrs/dia durante 4 dias, havendo entre as duas semanas um tempo de espera de 88 hrs. O intervalo de espera entre os dias de uma mesma semana é de 16 hrs.

A atividade é calculada pela expressão seguinte:

$$A_n = A_{sat} \frac{(1 - \text{Exp}(-\frac{.693}{T} t_i)) (1 - \text{Exp}(-\frac{.693}{T} n (t_i + t_e)))}{1 - \text{Exp}(-\frac{.693}{T} (t_i + t_e))}$$

onde:

$$A_{sat} = \frac{1.63 \cdot 10^{-8} \phi \sigma a}{M}$$

t_i = tempo de irradiação = 8 hrs/dia

t_e = tempo de espera = 16 hrs/dia

$t_i + t_e$ = 24 hrs = 86.400 seg.

n_1 = nº de dias da primeira irradiação = 4

n_2 = nº de dias da segunda irradiação = 5

ϕ = 10^{13} = $.1 \cdot 10^{14}$ nêutrons

σ = seção de choque microscópico

a = fração do isótopo por grama de T_e natural

T = meia vida de cada isótopo formado

M = massa atômica de cada isótopo formado

Os dados de entrada estão relacionados na tabela abaixo:

	M	σ	a	T	T em seg
T_e^{120}	T_e^{121m}	2,0	0,00089	154 dias	13.305.600
	T_e^{121}	0,3	0,00089	17 dias	1.468.800
T_e^{122}	T_e^{123m}	1,0	0,0246	104 dias	8.985.600
T_e^{124}	T_e^{125m}	5,0	0,0461	58 dias	5.111.200

	M	σ	a	T	T em seg
T_e^{126}	$T_e^{127 m}$	0,1	0,1871	105 dias	9.072.000
	T_e^{127}	0,9	0,1871	9,3 hrs	33.480
T_e^{128}	$T_e^{129 m}$	0,017	0,3179	33 dias	2.851.200
	T_e^{129}	0,14	0,3179	74 min.	4.440
T_e^{130}	$T_e^{131 m}$	0,008	0,3448	30 hrs	108.000
	T_e^{131}	0,22	0,2448	25 min.	1.500

O cálculo realizado em ERP está dado na pag 72 .

5.2.2 – Uma segunda aplicação corresponde a um problema de cálculo de reatores de potência, que é o seguinte:

Deseja-se construir um reator do tipo urânio natural, grafita e gás com uma potência térmica $W_{tm} = 1650$ Mw. Calcular a potência máxima que se pode tirar dum canal com as seguintes limitações tecnológicas.

$$T_{u_{máx}} = 610^\circ\text{C} \text{ temperatura do combustível}$$

$$T_{g_{máx}} = 475^\circ\text{C} \text{ temperatura máxima da camisa}$$

Dados geométricos:

$$l_g = 60 \text{ cm altura da camisa por cartucho}$$

$$l_u = 53,25 \text{ cm altura do urânio por cartucho}$$

$$l_e = 15 \text{ altura extrapolada do reator}$$

$$D_o = 4,3 \text{ cm diâmetro exterior do combustível}$$

$$D_i = 2,3 \text{ cm diâmetro interior do combustível}$$

$$S = 68,8 \text{ cm}^2 \text{ seção de passagem do gás}$$

$$\mu = 268 = 10^{-5} \text{ MTS viscosidade do gás}$$

$$C_p = 1090 \text{ MTS calor específico do gás}$$

$$h_e = 40 \text{ MTS coeficiente de intercâmbio urânio-camisa}$$

$$T_e = 220^\circ\text{C} \text{ temperatura de entrada do gás}$$

A potência é dada pela expressão

$$P = q C_p \Delta\theta$$

q = vazão do gás

$$\Delta \theta = \theta_s - \theta_e$$

C_p é dado, tem-se que calcular q e $\Delta\theta$ de modo que a potência seja máxima.

Inicialmente calcula-se o valor da relação

$$(1) \frac{1 + \frac{1}{\text{sen } \gamma \text{ sen } \beta}}{1 + \frac{1}{\text{sen } \gamma \text{ sen } \beta u}} \quad \text{que é igual}$$

a relação das diferenças de temperatura

$$(2) \frac{t_{g\max} - \theta_e}{t_{u\max} - \theta_e}$$

$$(1) = (2)$$

Os parâmetros de (2) são dados.

Toma-se um valor aproximado do q e com esse se determina um novo valor da relação (1). Se não for igual ao calculado através da relação (2) toma-se outro valor de q e segue-se o mesmo processo até achar o valor correto que dará a potência máxima.

O valor de $\Delta\theta$ é determinado através de uma das seguintes expressões:

$$t_{g\max} - \theta_e = \frac{\Delta\theta}{2} \left(1 + \frac{1}{\text{sen } \gamma \text{ sen } \beta} \right)$$

$$t_{u\max} - \theta_e = \frac{\Delta\theta}{2} \left(1 + \frac{1}{\text{sen } \gamma \text{ sen } \beta u} \right)$$

onde o produto dos senos são determinados através do valor de q pela expressão (1).

Com estes valores obtemos o valor máximo da potência por canal.

Nas páginas 76 a 81 temos o programa em ERP e os resultados das tentativas realizadas.

5.3 – Como exemplo de aplicação dos comandos SEJA, TRAP e LOCR nas páginas 88 e 89 damos a saída pela máquina de escrever do cálculo das raízes do seguinte polinômio:

$$P(x) = x^3 + 3x^2 + x - 5$$

O programa em ERP que calcula estas raízes é:

```

EX *ADM .DEF P 5 Q 5 .ALFA
EX *E/S LEIA P
EX
RG 10 SEJA Q . = P
RG 20 *E/S LEIA ALFA
RG 30 TRAP .TRANSL Q ALFA
RG 40 LOCR Q
RG 50 *E/S .ESCREVE Q
RG 60 *CTL .SALTE .10
RG
EX *CTL .ALT .MODO .PR

```

5 4 – Para exemplificar o comando PLOT utilizamos a instrução:

```
PLOT X Y
```

onde X e Y foram definidos com as instruções

```

*ADM .DEFINA .X 50 .Y 50
*E/S .LEIA CT X Y

```

o gráfico resultante o temos na pag. 71 .

5 5 – Nas pags 83 a 87 damos um exemplo sobre operações com matrizes escrevendo várias opções numa mesma instrução do comando MATRIZ e as correspondentes saídas pela impressora

5 6 – Na página 92 está a saída da solução do sistema linear seguinte:

$$\begin{bmatrix} 4 & 2 & 1 & 1 \\ 1 & -1 & 2 & -1 \\ 3 & 3 & -3 & 2 \\ 2 & -2 & 4 & -2 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \\ 5 \\ 2 \end{bmatrix}$$

Utilizamos as instruções seguintes:

```

*ADM DEFINA .A 4 4 .X 4 B4
*E/S .LEIA CT A B
SOLUÇÃO A X B
*E/S .ESCREVE X

```

57 – As instruções seguintes servem para exemplificar o uso do comando AJUSTE no ajuste de pontos experimentais, utilizando o método dos mínimos quadrados aplicado a polinômios ortogonais.

```
*ADM .DEFINA .X 9.Y 9.A 6.W 9.
```

```
*E/S .LEIA .CT X Y W
```

```
AJUSTE .COMPESO X Y W A
```

```
*E/S .ESC A
```

Os valores das constantes para um ajuste por polinômio de 5º grau estão dados no vetor A cujos valores estão na pag. 93 .

COMANDOS USADOS

```
01 *CTL.ALTERE,ENTRADA,MO,SAIDA,IM,ENTRADA,CT#
02 *CTL.ALTERE,SAIDA,IMPRESSORA,ENERADA,CT#
03 *ADM,DEFINIAO,REDEFINIR,VARIAVEL -EXEC,INTERR,NO,PAR,0
04 *ADM,DEFINA,A,17,H,2,10
05 *ADM,INITIALISE#
06 *CTL.ALTERE,SAIDA,IM,ENTRADA,CR#
07 *CTL.ALTERE,SAIDA,IM,ENTRADA,CI#
08 *ADM,INITIALISE#
09 *CTL.ALTERE,SAIDA,IM,ENTRADA,CT#
10 *PAG,01-ESPERADO,VAR, ENCONTRADO,NUM,-EXEC,INTERR,NO,PAR,0
11 *80*CTL,SE,CO,NAO,MAIOR,24,SALTE,10
12 *ADM,INI#
13 *CTL,AL,SA,IM,EN,CT#
14 *PAG,01-INTERVENCAO,DO,OPERADOR -EXEC,INTERR,NO,PAR,0
15 *10#F/S,LEIA,MA,SIG,FA,T
16 *ADM,INI#
17 *CTL.ALTERE,SAIDA#
```

A SEGUIR APARECEM OS COMANDOS UTILIZADOS PARA PRODUZIR OS EXEMPLOS
DE MENSAGENS SEQUENTES, FORAM OMITIDOS PROPOSITADAMENTE ALGUNS CARACTERES
DE CONTROLE PARA FACILITAR A LEITURA DAS MENSAGENS CORRESPONDENTES, COMO EXEMPLOS.

EXEMPLO DE UTILIZACAO DO COMANDO *MANIPULE

SAIDA OBTIDA LIGANDO A CHAVE 2 DA CONSOLE

01*ADM.ROL

LISTAGEM PAG 01

01*ADM.ROL
 02*ADM.DEFINA.A 17.H 2 10.
 03*E/S.LEIA.CT A H
 05*E/S.EXIBA.TS.AC 1 40.AK
 10*MAN.DECOMPNHA A.EM.B 2.C 5.E 10.
 15*E/S.EXIBA.TS.AK.ESCREVA B C E
 20*MAN.COMPNHA B C.EM.F.
 25*E/S.EXIBA.TS.AC 1 40.ESC F
 30*MANIPULE.PARTA H.EM.M 2.N 3.P 5.
 35*E/S.EXIBA.TS.ESC M N P
 40*MAN.GRUPE N P.EM.L.
 45*E/S.EXIBA.TS.ESC L
 46*ADM.APAGUE L.DEFINA.L 3 5.
 47*E/S.LEIA L.ESC.IM L
 50*MAN.TRANSPONHA L.
 55*E/S.EXIBA.TS.AC 1 40.ESCREVA L
 60*ADM.ATIVE A B C H N P
 65*E/S.EXIBA.TS.AC 1 80.AK
 70*CTL.ALTERE.MODO.EXECUTE.SAIDA.MQ.ENTRADA.MQ

02*ADM.DEFINA.A 17.H 2 10.

03*E/S.LEIA.CT A 1.

05*E/S.EXIBA.TS.AC 1 40.AK

TABELA DE SIMBOLOS

A	1	17	1	C
H	18	2	10	0

AREA COMUM 1 40

.10000000E+01	.20000000E+01	.30000000E+01	.40000000E+01
.50000000E+01	.60000000E+01	.70000000E+01	.80000000E+01
.90000000E+01	.10000000E+02	.11000000E+02	.12000000E+02
.13000000E+02	.14000000E+02	.15000000E+02	.16000000E+02
.17000000E+02	.11000000E+02	.21000000E+02	.12000000E+02
.22000000E+02	.13000000E+02	.23000000E+02	.14000000E+02
.24000000E+02	.15000000E+02	.25000000E+02	.16000000E+02
.26000000E+02	.17000000E+02	.27000000E+02	.18000000E+02
.28000000E+02	.19000000E+02	.29000000E+02	.11000000E+03
.21000000E+03	.00000000E-99	.00000000E-99	.00000000E-99

AREA COMUNICACOES

IND1	IND2	IND3	IND4	ERRO	K1	K2	K3	K4	INDAL
0	0	0	0	0	5	0	0	0	38
1	1	1	500	900	20	1	2	0	0
IND5	INDPG	INDSK	IDE	IDS	INDTI	NPG	KX1	KX2	KX3

COMPANHIA A.E.M.B 2.C 5.F 199

TADEIRA.TS.AK.ESCREVA B C E

TABELA DE SIMBOLOS

A	0	17	1	1
4	1	2	10	0
9	21	2	1	0
0	23	5	1	0
8	28	10	1	0

AREA COMUNICACOES

IND1	IND2	IND3	IND4	ERRO	K1	K2	K3	K4	INDAL
0	0	0	0	0	7	0	0	0	38
1	1	3	500	900	20	1	2	0	0
IND5	INDPG	INDSK	IDE	IDS	INDTI	NPG	KX1	KX2	KX3

B 2 VALOR(ES)

.10000000E+01

.20000000E+01

C 5 VALOR(ES)

.30000000E+01

.40000000E+01

.50000000E+01

.60000000E+01

.70000000E+01

E 10 VALOR(ES)

.80000000E+01

.90000000E+01

.10000000E+02

.11000000E+02

.12000000E+02

.14000000E+02

.14000000E+02

.15000000E+02

.16000000E+02

.17000000E+02

20*MAN.COMPONHA R C.I.M.F.

25*E/S.EXIRA.TS.AC 1 40.ESC F

TABELA DE SIMBOLOS

A	0	17	1	1
H	1	2	10	0
B	0	2	1	3
C	0	5	1	4
F	21	10	1	0
F	31	7	1	0

AREA COMUM 1 40

.11000000E+02	.21000000E+02	.12000000E+02	.22000000E+02
.13000000E+02	.23000000E+02	.14000000E+02	.24000000E+02
.15000000E+02	.25000000E+02	.16000000E+02	.26000000E+02
.17000000E+02	.27000000E+02	.18000000E+02	.28000000E+02
.19000000E+02	.29000000E+02	.11000000E+03	.21000000E+03
.80000000E+01	.90000000E+01	.10000000E+02	.11000000E+02
.12000000E+02	.13000000E+02	.14000000E+02	.15000000E+02
.16000000E+02	.17000000E+02	.10000000E+01	.20000000E+01
.30000000E+01	.40000000E+01	.50000000E+01	.60000000E+01
.70000000E+01	.00000000E-99	.00000000E-99	.00000000E-99

F 7 VALOR(ES)

.10000000E+01

.20000000E+01

.30000000E+01

.40000000E+01

.50000000E+01

.60000000E+01

.70000000E+01

30*MANIPULE.PARTA H.EM.M 2.N 3.P 5.

35*E/S.EXIBA.TS.ESC M N P

TABELA DE SIMBOLOS

A	0	17	1	1
H	0	2	10	5
B	0	2	1	3
C	0	5	1	4
E	1	10	1	0
F	11	7	1	0
M	18	2	2	0
N	22	2	3	0
P	28	2	5	0

M 2 X 2

.11000000E+02 .12000000E+02

.21000000E+02 .22000000E+02

N 2 X 3

.13000000E+02 .14000000E+02 .15000000E+02

.23000000E+02 .24000000E+02 .25000000E+02

P 2 X 5

.16000000E+02 .17000000E+02 .18000000E+02 .19000000E+02

.11000000E+03

.26000000E+02 .27000000E+02 .28000000E+02 .29000000E+02

.21000000E+03

40*MAN.GRUPE N P.E.M.L.

45*E/S.EXIBA.TS.ESC L

TABELA DE SIMBOLOS

A	0	17	1	1
H	0	2	10	5
B	0	2	1	3
C	0	5	1	4
E	1	10	1	0
F	11	7	1	0
M	18	2	2	0
N	0	2	3	7
P	0	2	5	8
L	22	2	8	0

L 2 X 8

.13000000E+02	.14000000E+02	.15000000E+02	.16000000E+02
.17000000E+02	.18000000E+02	.19000000E+02	.20000000E+02
.23000000E+02	.24000000E+02	.25000000E+02	.26000000E+02
.27000000E+02	.28000000E+02	.29000000E+02	.30000000E+02

46*ADM.AFAGUE L.DEFINA.L 3 5.

47*E/S.LETA L.ESC.IM L

L 3 X 5

.11000000E+02	.12000000E+02	.13000000E+02	.14000000E+02
.15000000E+02			
.21000000E+02	.22000000E+02	.23000000E+02	.24000000E+02
.25000000E+02			
.31000000E+02	.32000000E+02	.33000000E+02	.34000000E+02
.35000000E+02			

50*MAN.TRANSPONHA L.

55*E/S.EX16A.TS.AC 1 40.ESCREVA I

TABELA DE SIMBOLOS

A	0	17	1	1
B	0	2	10	5
R	0	2	1	3
C	0	5	1	4
E	1	10	1	0
F	11	7	1	0
M	18	2	2	0
N	0	2	3	7
P	0	2	5	8
L	22	5	3	0

ARFA COMUM 1 00

.80000000E+01	.90000000E+01	.10000000E+02	.11000000E+02
.12000000E+02	.13000000E+02	.14000000E+02	.15000000E+02
.16000000E+02	.17000000E+02	.18000000E+02	.19000000E+02
.20000000E+02	.21000000E+02	.22000000E+02	.23000000E+02
.24000000E+02	.25000000E+02	.26000000E+02	.27000000E+02
.28000000E+02	.29000000E+02	.30000000E+02	.31000000E+02
.32000000E+02	.33000000E+02	.34000000E+02	.35000000E+02
.00000000E-99	.00000000E-99	.00000000E-99	.00000000E-99

L 5 X 3

.11000000E+02	.21000000E+02	.31000000E+02
.12000000E+02	.22000000E+02	.32000000E+02
.13000000E+02	.23000000E+02	.33000000E+02
.14000000E+02	.24000000E+02	.34000000E+02
.15000000E+02	.25000000E+02	.35000000E+02

60*ADM.ATIVE A B C H N P

65*E/S.EXTRA.TS.AC 1 80.AK

TABELA DE SIMBOLOS

A	37	17	1	1
H	61	2	10	4
B	54	2	1	3
C	56	5	1	4
E	1	10	1	0
F	11	7	1	0
M	18	2	2	0
N	81	2	3	7
P	87	2	5	8
L	22	5	3	0

AREA COMUM 1 80

.80000000E+01	.90000000E+01	.10000000E+02	.11000000E+02
.12000000E+02	.13000000E+02	.14000000E+02	.15000000E+02
.16000000E+02	.17000000E+02	.18000000E+01	.20000000E+01
.30000000E+01	.40000000E+01	.50000000E+01	.60000000E+01
.70000000E+01	.11000000E+02	.21000000E+02	.12000000E+02
.22000000E+02	.11000000E+02	.12000000E+02	.13000000E+02
.14000000E+02	.15000000E+02	.21000000E+02	.22000000E+02
.23000000E+02	.24000000E+02	.25000000E+02	.31000000E+02
.32000000E+02	.33000000E+02	.34000000E+02	.35000000E+02
.10000000E+01	.20000000E+01	.30000000E+01	.40000000E+01
.50000000E+01	.60000000E+01	.70000000E+01	.80000000E+01
.90000000E+01	.10000000E+02	.11000000E+02	.12000000E+02
.13000000E+02	.14000000E+02	.15000000E+02	.16000000E+02
.17000000E+02	.18000000E+01	.20000000E+01	.30000000E+01
.40000000E+01	.50000000E+01	.60000000E+01	.70000000E+01
.11000000E+02	.21000000E+02	.12000000E+02	.22000000E+02
.13000000E+02	.23000000E+02	.14000000E+02	.24000000E+02
.15000000E+02	.25000000E+02	.16000000E+02	.26000000E+02
.17000000E+02	.27000000E+02	.18000000E+02	.28000000E+02
.19000000E+02	.29000000E+02	.11000000E+03	.21000000E+03

AREA COMUNICACOES

IND1	IND2	IND3	IND4	ERRO	K1	K2	K3	K4	INDAC
0	0	0	0	0	19	0	0	0	97
11	1	9	500	900	20	1	2	0	0
INDTS	INDPG	INDSK	IDE	IDS	INDFI	NPG	KY1	KX2	KX3

EXEMPLOS DO COMANDO *CALC APLICADO A PRODUCAO DE RADIOISOTOPOS

TABELA DE SIMBOLOS

C1	1	1	1	0
C2	2	1	1	0
TI	3	1	1	0
TE	4	1	1	0
FI	5	1	1	0
N	6	1	1	0
CO	7	1	1	0
CF	8	1	1	0
MA	9	1	1	0
SI	10	1	1	0
FA	11	1	1	0
T	12	1	1	0
A1	13	1	1	0
F1	14	1	1	0
F2	15	1	1	0
DE	16	1	1	0
AN	17	1	1	0
M	18	1	1	0

C1 1 VALOR(ES)
 .10000000E-07

C2 1 VALOR(ES)
 .69300000E+00

TI 1 VALOR(ES)
 .28800000E+05

TE 1 VALOR(ES)
 .57600000E+05

FI 1 VALOR(ES)
 .10000000E+14

N 1 VALOR(ES)
 .40000000E+01

CF 1 VALOR(ES)
 .10000000E+02

```

LISTAGEM PAG 01
10*E/S.LEIA MA SIG FA T
15*CAL CO.= CO.+ 1
20*CAL AINF.= C1.* F1.* SIG.* FA./ MA
30*CAL F1.= 1.-.EXP.(.-.( C2.* T1./ T.))
40*CAL F2.= 1.-.EXP.(.-.( C2.* N.*( T1.+ TE.)/ T.))
50*CAL DEN.= 1.-.EXP.(.-.( C2.*( T1.+ TE.)/ T.))
60*CAL AN.= AINF.* F1.* F2./ DEN
70*E/S.ESC MA AN
80*CTL.SE CO.NAO.MAIOR CF.SALTE.10
85*CTL.SE N.NAO.= M.SALTE.95
90SEJA N.= 5 CO.= 10
93*CTL.SALTE.10
95*CTL.ALTERE.MODO.EXECUTE

```

***LIGANDO A CHAVE 1**

```

10*E/S.LEIA MA SIG FA T

15*CAL CO.= CO.+ 1

20*CAL AINF.= C1.* F1.* SIG.* FA./ MA

30*CAL F1.= 1.-.EXP.(.-.( C2.* T1./ T.))

40*CAL F2.= 1.-.EXP.(.-.( C2.* N.*( T1.+ TE.)/ T.))

50*CAL DEN.= 1.-.EXP.(.-.( C2.*( T1.+ TE.)/ T.))

60*CAL AN.= AINF.* F1.* F2./ DEN

70*E/S.ESC MA AN

MA 1 VALOR(ES)
    .12100000E+03

AN 1 VALOR(ES)
    .87568970E-02

80*CTL.SE CO.NAO.MAIOR CF.SALTE.10

10*E/S.LEIA MA SIG FA T

```

MA 1 VALOR(ES)
 .12100000E+03
AN 1 VALOR(ES)
 .11217060E-01
80*CTL.SE CO.NAO.MAIOR CF.SALTE.10

10*E/S.LEIA MA SIG FA T

MA 1 VALOR(ES)
 .12300000E+03
AN 1 VALOR(ES)
 .17570239E+00
80*CTL.SE CO.NAO.MAIOR CF.SALTE.10

10*E/S.LEIA MA SIG FA T

MA 1 VALOR(ES)
 .12500000E+03
AN 1 VALOR(ES)
 .28245469E+01
80*CTL.SE CO.NAO.MAIOR CF.SALTE.10

10*E/S.LEIA MA SIG FA T

MA 1 VALOR(ES)
 .12700000E+03
AN 1 VALOR(ES)
 .12820559E+00
80*CTL.SE CO.NAO.MAIOR CF.SALTE.10

10*E/S.LEIA MA SIG FA T

MA 1 VALOR(ES)

.12700000E+03

AN 1 VALOR(ES)

.71441680E+02

80*CL.SE CO.NAO.MAIOR CF.SALTE.10

10*E/S.LEIA MA SIG FA T

MA 1 VALOR(ES)

.12900000E+03

AN 1 VALOR(ES)

.11329354E+00

80*CTL.SE CO.NAO.MAIOR CF.SALTE.10

10*E/S.LEIA MA SIG FA T

MA 1 VALOR(ES)

.12900000E+03

AN 1 VALOR(ES)

.35821463E+02

80*CTL.SE CO.NAO.MAIOR CF.SALTE.10

10*E/S.LEIA MA SIG FA T

MA 1 VALOR(ES)

.13100000E+03

AN 1 VALOR(ES)

.74392913E+00

80*CTL.SE CO.NAO.MAIOR CF.SALTE.10

10*E/S.LEIA MA SIG FA T

MA 1 VALOR(ES)

.13100000E+03

AN 1 VALOR(ES)

.43400806E+02

80*CTL.SE CO.NAO.MAIOR CF.SALTÉ.10

10*E/S.LEIA MA SIG FA T

EXEMPLO DO COMANDO *CALC APLICADO A CALCULO DE REATORES

TABELA DE SIMBOLOS

GA	1	1	1	0
A	2	1	1	0
Q	3	1	1	0
RF	4	1	1	0
CU	5	1	1	0
TA	6	1	1	0
BE	7	1	1	0
CO	8	1	1	0
DE	9	1	1	0
CM	10	1	1	0
BU	11	1	1	0
RL	12	1	1	0
RT	13	1	1	0
DT	14	1	1	0
PM	15	1	1	0
C3	16	1	1	0
PI	17	1	1	0
H	19	1	1	0
HF	19	1	1	0
DU	20	1	1	0
DZ	21	1	1	0
C4	22	1	1	0
S	23	1	1	0
VI	24	1	1	0
C1	25	1	1	0
C2	26	1	1	0
CL	27	1	1	0
CG	28	1	1	0
AL	29	1	1	0
AZ	30	1	1	0
CN	31	1	1	0
CP	32	1	1	0
TM	33	1	1	0
TE	34	1	1	0
TU	35	1	1	0

AREA COMUNICAOES

IND1	IND2	IND3	IND4	ERRO	K1	K2	K3	K4	INDAC
0	0	0	0	0	1	0	0	0	36
36	1	1	500	100	1	1	2	0	0
INDTS	INDPG	INDSK	IDE	IDS	INDTI	NPG	KX1	KX2	KX3

PI 1 VALOR(ES)

.31415920E+01

H 1 VALOR(ES)

.90000000E+01

HF 1 VALOR(ES)

.98500000E+01

DU 1 VALOR(ES)

.23000000E-01
DZ 1 VALOR(ES)
.43000000E-01
S 1 VALOR(ES)
.68800000E-02
V1 1 VALOR(ES)
.26800000E-07
C1 1 VALOR(ES)
.19600000E-01
C2 1 VALOR(ES)
.70000000E+06
CL 1 VALOR(ES)
.53250000E+00
CG 1 VALOR(ES)
.60000000E+00
AL 1 VALOR(ES)
.95000000E+00
H7 1 VALOR(ES)
.40000000E+02
CN 1 VALOR(ES)
.30000000E-01
CP 1 VALOR(ES)
.10900000E+04
TH 1 VALOR(ES)
.47500000E+03
TE 1 VALOR(ES)
.22000000E+03
TU 1 VALOR(ES)
.61000000E+03
VALOR(ES)
.1333333E+00

C4 1 VALOR(F5)

.1000000E-02

LISTAGEM PAG 01

```

05*CALCULE GAMA.= PI./ 2.* H./ HE
10*CALC A.= 1.- 2.* DUM.** 2./.( DZERO.** 2.- DUM.** 2.).*LN.( DZERO./ DUM.)
15*E/S.LEIA.MQ Q
20*CALC RE.= Q.* DZERO./.( S.* VISC.)
25*CALC GULIS.= C1.*.( C2./ RE.).** C3
30*CALC TANGB.=.( ( CLU./ CGL.).* GULIS.* DZERO.* HE.)/( ALFA.* S.)
35*CALC BETA.=.ATAN.( TANGB.)
40*CALC DENOM.= 1.+ 1./.( (.SEN.( GAMA.).)*.(.SEN.( BETA.).).)
45*CALC COTBU.= 1./ TANGB.+ ALFA./.( CLU./ CGL.).*(. ( A./.( 4.* CND.).).+ 1.
46//.( HZERO.* DZERO.).)* Q.* CP./ HE
50*CALC BU.=.ATAN.( 1./ COTBU.)
55*CALC CMB.= 1.+ 1./.( (.SEN.( GAMA.).)*.(.SEN.( BU.).).)
60*CALC RLAC.= CMB./ DENOM
65*CALC RTH.= ( TUM.- TE.)/ ( TMG.- TE.)
70*E/S.ESC.MQ RTH Q RLAC
72*CALC RE.=.( RL.- RT.)/ RL
75*CTL.SE RE.MAIOR C4.SALTE.15
80*CALC DT.= 2.*( TUM.- TE.)/ CMB
85*CALC PM.= Q.* CP.* DT
90*E/S.ESC.MQ Q DT CP PM
95*CTL.ALTERE.MODO.EXECUT

```

LIGANDO A CHAVE 2

0.*CALCULE GAMA.= PI./ 2.* H./ HE

10*CALC A.= 1.- 2.* DUM.** 2./.(DZERU.** 2.- DUM.** 2.).*LN.(DZERO./ DUM.)

15*E/S.LEIA.MQ Q

20*CALC RE.= Q.* DZERO./.(S.* VISC.)

25*CALC GULIS.= C1.*.(C2./ RE.).** C3

30*CALC TANGB.=.((CLU./ CGL.).* GULIS.* DZERO.* HE.)/(ALFA.* S.)

35*CALC BETA.=.ATAN.(TANGD.)

40*CALC DENOM.= 1.+ 1./.((.SEN.(GAMA.).)*.(.SEN.(BETA.).).)

45*CALC COTBU.= 1./ TANGB.+ ALFA./.(CLU./ CGL.).*(. (A./.(4.* CND.).).+ 1.

46//.(HZERO.* DZERO.).)* Q.* CP./ HE

```

50*CALC BU.=.ATAN.' 1./ COTBU.)
55*CALC CMB.= 1.+ 1./.(.SEN.( GAMA.).).*(.SEN.( BU.).).)
60*CALC RLAC.= CMB./ DENOM
65*CALC RTH.=.( TUM.- TE.)/.( TMC.- TE.)
70*E/S.ESC.MO RTH Q RLAC
72*CALC RE.=.( RL.- RT.)/ RL
75*CTL.SE RE.MAIOR C4.SALTE.15
15*E/S.LEIA.MO G
72*CALC RE.=.( RL.- RT.)/ RL
75*CTL.SE RE.MAIOR C4.SALTE.15
10*CALC DT.= 2.*( TUM.- TE.)/ CMB
85*CALC PM.= Q.* CP.* DT
90*E/S.ESC.MO Q DT CP PM
95*CTL.ALTERE.MODD.E>FUIFF

```

```

NAZ PAGINAS SEQUENTES ESTAO OS VALORES DA VAZAO Q INTRODUIZIDAS PELA MAQUIL-
*****
NA DE ESCREVER, E OS VALORES Q SAIDAM NA SAIDA.

```

EX *CTL.ALTERE,ENTRADA,CTB

Q 1 VALOR(ES)
ENTRE
.0028

RT 1 VALOR(ES)
.15294117E+01

Q 1 VALOR(ES)
.20000000E-02

RL 1 VALOR(ES)
.13896948E+01

Q 1 VALOR(ES)
ENTRE
.00218

RT 1 VALOR(ES)
.15294117E+01

Q 1 VALOR(ES)
.21000000E-02

RL 1 VALOR(ES)
.14118805E+01

Q 1 VALOR(ES)
ENTRE
.00238

RT 1 VALOR(ES)
.15294117E+01

Q 1 VALOR(ES)
.23000000E-02

RL 1 VALOR(ES)
.14562193E+01

Q 1 VALOR(ES)
 ENTRE
 .00258

RT 1 VALOR(ES)
 .15294117E+01

Q 1 VALOR(ES)
 .25000000E-02

RL 1 VALOR(ES)
 .15004673E+01

Q 1 VALOR(ES)
 ENTRE
 .002653

RT 1 VALOR(ES)
 .15294117E+01

Q 1 VALOR(ES)
 .26500000E-02

RL 1 VALOR(ES)
 .15335668E+01

Q 1 VALOR(ES)
 ENTRE
 .00262

RT 1 VALOR(ES)
 .15294117E+01

Q 1 VALOR(ES)
 .26000000E-02

RL 1 VALOR(ES)
 .15225429E+01

Q 1 VALOR(ES)
 .26000000E-02

CT 1 VALOR(ES)
 .22063277E+03

CP 1 VALOR(ES)
.10900003F+04
PM 1 VALOR(ES)
.62532995F+03

EXEMPLO DE COMANDO MATRIZ

05*ADM.ROL

LISTAGEM PAG 01

05*ADM.ROL

10*ADM.DEFINA.A 2 3.B 2 3.C 2 3.D 3 2.F 2 2.G 4 4.H.I 3 5.J 5 3.

15*E/S.LEIA.CT A B D G I.EXIBA.TS.AC 1 85.AK

20MATRIZ.SOMA A B C

25*E/S.ESC.IM A B C

30MATRIZ.SUBTRA! A B C

35*E/S.ESCREVA.IM A B C

40MATRIZ.MULTIPLIQUE C D F

45*E/S.ESC.IM C D F

50MATRIZ.DETERMINANTE G H

55*E/S.ESC.IM G H

60MATRIZ.TRANSPONHA I J

65*E/S.ESC I J.LEIA.CT F.ESC.IM F

70MATRIZ.INVERTA F

75*E/S.ESC.IM F.EXIBA.IM.TS.AC 1 85.AK

80*CTL.ALTERE.MODO.EXECUTE

10*ADM.DEFINA.A 2 3.B 2 3.C 2 3.D 3 2.F 2 2.G 4 4.H.I 3 5.J 5 3.

15*E/S.LEIA.CT A B D G I.EXIBA.TS.AC 1 85.AK

TABELA DE SIMBOLOS

A	1	2	3	0
B	7	2	3	0
C	13	2	3	0
D	19	3	2	0
F	25	2	2	0
G	29	4	4	0
H	45	1	1	0
I	46	3	5	0
J	61	3	3	0

AREA (LINE) 1 85

```

.11000000E+02 .21000000E+02 .12000000E+02 .27000000E+02
.13000000E+02 .23000000E+02 .11000000E+02 .21000000E+02
.12000000E+02 .22000000E+02 .13000000E+02 .23000000E+02
.00000000E-99 .00000000E-99 .00000000E-99 .00000000E-99
.00000000E-99 .00000000E-99 .11000000E+02 .21000000E+02
.31000000E+02 .12000000E+02 .22000000E+02 .32000000E+02
.00000000E-99 .00000000E-99 .00000000E-99 .00000000E-99
.10000000E+01 .20000000E+01 .30000000E+01 .80000000E+01
.30000000E+01 -.20000000E+01 -.10000000E+01 .20000000E+01
-.10000000E+01 .10000000E+01 .10000000E+01 -.20000000E+01
-.10000000E+01 .30000000E+01 -.40000000E+01 -.40000000E+01
.00000000E-99 .11000000E+02 .21000000E+02 .31000000E+02
.12000000E+02 .22000000E+02 .32000000E+02 .13000000E+02
.23000000E+02 .33000000E+02 .14000000E+02 .24000000E+02
.34000000E+02 .15000000E+02 .25000000E+02 .35000000E+02
.00000000E-99 .00000000E-99 .00000000E-99 .00000000E-99
.00000000E-99 .00000000E-99 .00000000E-99 .00000000E-99
.00000000E-99 .00000000E-99 .00000000E-99 .00000000E-99
.00000000E-99 .00000000E-99 .00000000E-99 .00000000E-99
.00000000E-99 .00000000E-99 .00000000E-99 .00000000E-99
.00000000E-99 .00000000E-99 .00000000E-99 .00000000E-99
.00000000E-99 .00000000E-99 .00000000E-99 .00000000E-99

```

AREA COMUNICAOES

IND1	IND2	IND3	IND4	ERRO	K1	K2	K3	K4	INDAC
0	0	0	0	0	4	0	0	0	76
10	1	1	500	900	17	1	2	0	0
INDTS	INDPG	INDSK	IDE	IOS	INDTI	NPG	Kx1	Kx2	Kx3

20MATRIZ.SOMA A B C

25+E/S.ESC.IM A B C

A 2 X 3

```

.11000000E+02 .12000000E+02 .13000000E+02
.21000000E+02 .22000000E+02 .23000000E+02

```

B 2 X 3

```

.11000000E+02 .12000000E+02 .13000000E+02
.21000000E+02 .22000000E+02 .23000000E+02

```

C 2 X 3

```

.22000000E+02 .24000000E+02 .26000000E+02
.42000000E+02 .44000000E+02 .46000000E+02

```

30MATRIZ.SUBTRAI A B C

35*E/S.ESCREVA.IM A B C

A 2 X 3

.11000000E+02	.12000000E+02	.13000000E+02
.21000000E+02	.22000000E+02	.23000000E+02

B 2 X 3

.11000000E+02	.12000000E+02	.13000000E+02
.21000000E+02	.22000000E+02	.23000000E+02

C 2 X 3

.00000000E-99	.00000000E-99	.00000000E-99
.00000000E-99	.00000000E-99	.00000000E-99

40MATRIZ.MULTIPLIQUE C D F

45*E/S.ESC.IM C D F

C 2 X 3

.00000000E-99	.00000000E-99	.00000000E-99
.00000000E-99	.00000000E-99	.00000000E-99

D 3 X 2

.11000000E+02	.12000000E+02
.21000000E+02	.22000000E+02
.31000000E+02	.32000000E+02

F 2 X 2

.00000000E-99	.00000000E-99
.00000000E-99	.00000000E-99

50MATRIZ.DETERMINANTE G H

55*E/S.ESC. JM G H

G 4 X 4

.10000000E+01	.30000000E+01	-.10000000E+01	-.10000000E+01
.20000000E+01	-.20000000E+01	.10000000E+01	.30000000E+01
.30000000E+01	-.10000000E+01	.10000000E+01	-.40000000E+01
.00000000E+01	.20000000E+01	-.20000000E+01	-.40000000E+01

H 1 VALOR(ES)

.15000000E+03

60MATRIZ. TRANSPUNHA I J

65*E/S.ESC I J. LEIA. CT F. ESC. JM F

I 3 X 5

.11000000E+02	.12000000E+02	.13000000E+02	.14000000E+02
.15000000E+02			
.21000000E+02	.22000000E+02	.23000000E+02	.24000000E+02
.25000000E+02			
.31000000E+02	.32000000E+02	.33000000E+02	.34000000E+02
.35000000E+02			

J 5 X 3

.11000000E+02	.21000000E+02	.31000000E+02
.12000000E+02	.22000000E+02	.32000000E+02
.13000000E+02	.23000000E+02	.33000000E+02
.14000000E+02	.24000000E+02	.34000000E+02
.15000000E+02	.25000000E+02	.35000000E+02

F 2 X 2

.10000000E+01	.20000000E+01
.30000000E+01	.40000000E+01

70. MATRIZ. INVERTA F

75*E/5.E.SC.IM F.EX...IM.75.AC 1 85.AK

F 2 X 2

-.2000000E+01 .1000000E+01
 .1500000E+01 -.5000000E+00

TABELA DE SIMBOLOS

A	1	2	3	0
B	7	2	3	0
C	13	2	3	0
D	19	3	2	0
F	25	2	2	0
G	29	4	4	0
H	45	1	1	0
I	46	3	5	0
J	61	5	3	0

AREA COMM 1 85

.1100000E+02	.2100000E+02	.1200000E+02	.2200000E+02
.1300000E+02	.2300000E+02	.1100000E+02	.2100000E+02
.1200000E+02	.2200000E+02	.1300000E+02	.2300000E+02
.0000000E-99	.0000000E-99	.0000000E-99	.0000000E-99
.0000000E-99	.0000000E-99	.1100000E+02	.2100000E+02
.3100000E+02	.1200000E+02	.2200000E+02	.3200000E+02
-.2000000E+01	.1500000E+01	.1000000E+01	-.5000000E+00
.1000000E+01	.2000000E+01	.3000000E+01	.8000000E+01
.3000000E+01	-.2000000E+01	-.1000000E+01	.2000000E+01
-.1000000E+01	.1000000E+01	.1000000E+01	-.2000000E+01
-.1000000E+01	.3000000E+01	-.4000000E+01	-.4000000E+01
.1500000E+03	.1100000E+02	.2100000E+02	.3100000E+02
.1200000E+02	.2700000E+02	.3200000E+02	.1300000E+02
.2300000E+02	.3300000E+02	.1400000E+02	.2400000E+02
.3400000E+02	.1500000E+02	.2500000E+02	.3500000E+02
.1100000E+02	.1200000E+02	.1300000E+02	.1400000E+02
.1500000E+02	.2100000E+02	.2200000E+02	.2300000E+02
.2400000E+02	.2500000E+02	.3100000E+02	.3200000E+02
.3300000E+02	.3400000E+02	.3100000E+02	.0000000E-99
.0000000E-99	.0000000E-99	.0000000E-99	.0000000E-99
.0000000E-99	.0000000E-99	.0000000E-99	.0000000E-99
.0000000E-99	.0000000E-99	.0000000E-99	.0000000E-99

AREA COMUNICACOES

IND1	IND2	IND3	IND4	ERRO	K1	K2	K3	K4	INDAC
0	0	0	0	0	16	0	0	0	76
10	1	1	50	90	17	1	2	0	0
INDTS	INDPG	INDSK	IDE	IDS	INDTI	MPG	KX1	KX2	KX3

80*CTL.ALTERE.MODO.EXECUTE

EXEMPLO DE UTILIZACAO DOS COMANDOS SFJA TRAP F LOCR

PR AS SUAS ORDENS

CX *CTL.ALTERE.ENTRADA.CTQ

PR 05*ADM.RUL

LISTAGEM PAG 01

- 05*ADM.RUL
- 10*ADM.DEFINA.P 5.Q 5.ALFA.
- 20*F/S.LEIA.CT P
- 30SEJA Q. = P
- 40*E/S.LEIA.MQ ALFA
- 50TRAP.TRANSLADE Q ALFA
- 60LOCR Q
- 65*E/S.ESC.MQ Q
- 70*CTL.SALTE.30

PR 10*ADM.DEFINA.P 5.Q 5.ALFA.

PR 20*E/S.LEIA.CT P

PR 30SEJA Q. = P

PR 40*E/S.LEIA.MQ ALFA.

AL 1 VALOR(FS)
ENTRE
0.02

PR 50TRAP.TRANSLADE Q ALFA

PR 60LOCR Q	.26666666E+01	.16666666E+01	2
-------------	---------------	---------------	---

PR 65*E/S.ESC.MQ Q

Q 5 VALOR(FS)

- .50000000E+01
- .20666666E+01
- .30000000E+01
- .10000000E+01
- .00000000E-99

PR 70*CTL.SALTE.30

PR 30SEJA Q. = P

PK 40*E/S..LEIA.MQ ALFA

AL 1 VALOR(ES)
ENTRE
1.05
DIVISOR NULO

Q 5 VALOR(ES)
-.40000000E+01
-.20000000E+01
.00000000E-99
.10000000E+01

.00000000E-99

AL 1 VALOR(ES)
ENTRE
2.5
 .10000000E-06 -.10000000E+01 0

Q 5 VALOR(ES)
-.30000000E+01
.10000000E-06
-.30000000E+01
.10000000E+01
.00000000E-99

AL 1 VALOR(ES)
ENTRE
2.5
 .57777778E+01 -.97222222E+00 0

Q 5 VALOR(ES)
-.43750000E+01
.57777778E+01
-.45000000E+01
.10000000E+01
.00000000E-99

K₁

K₀/K₂

Q

EXEMPLO DE APLICACAO DO COMANDO PLOT

*****:*****:*****:*****:*****:*****:*****:*****:*****:*****:

05*ADM.RCL.DEFINA.X 50.Y 50.

LISTAGEM PAG 01

05*ADM.RCL.DEFINA.X 50.Y 50.

10*E/S.LEIA.CT X Y.EXIBA.IM.AC 1 50.AC 51 100

20PLOT X Y

25*CTL.ALTERE.ENTRADA.MQ.SAIDA.MQ.MODD.EXECUTE

10*E/S.LEIA.CT X Y.EXIBA.IM.AC 1 50.AC 51 100

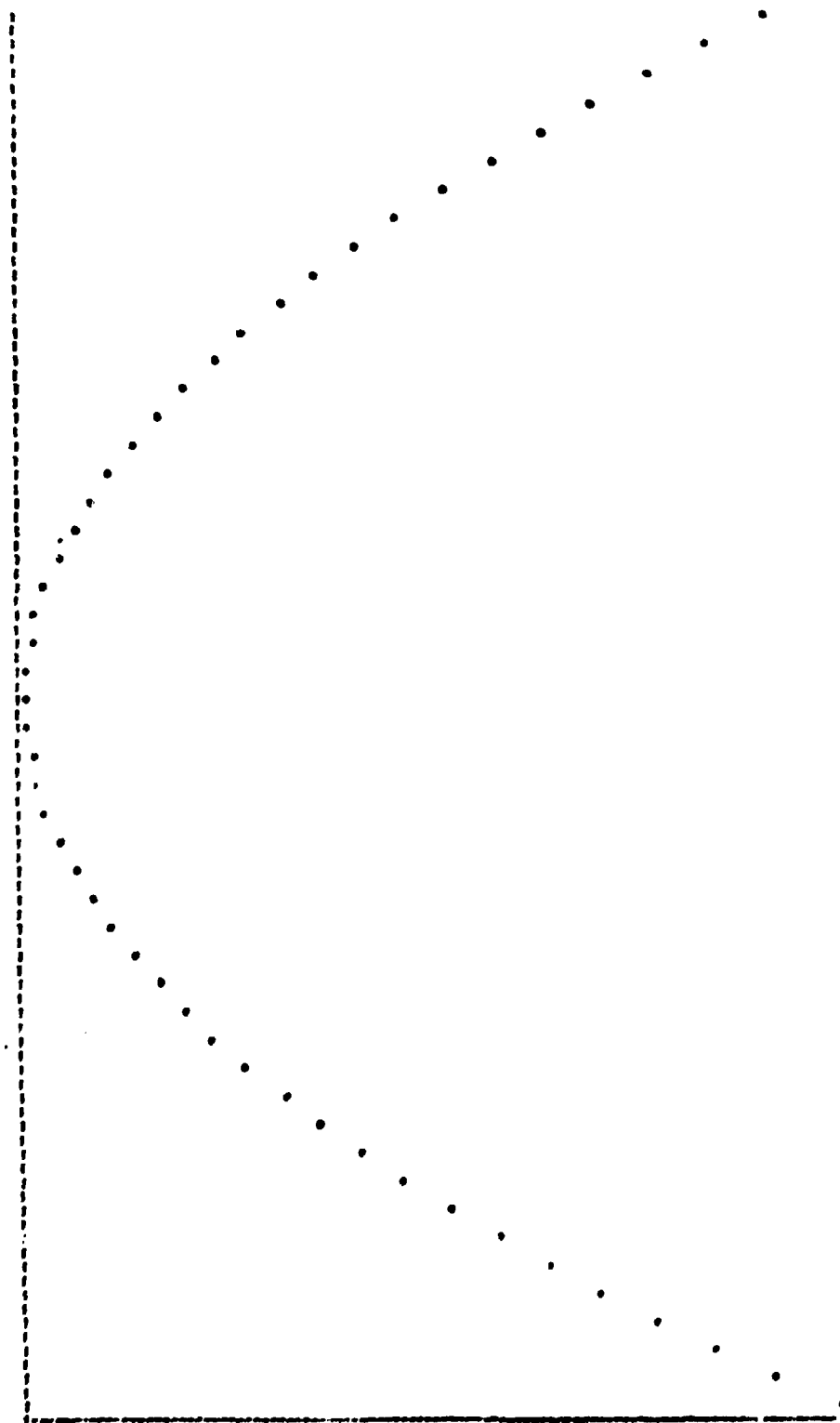
AREA COMUM 1 50

.13580000E+01	.12600000E+01	.11850000E+01	.11100000E+01
.10410000E+01	.98040000E+00	.92640000E+00	.87900000E+00
.30000000E+01	.28230000E+01	.26540000E+01	.24920000E+01
.23370000E+01	.21900000E+01	.20490000E+01	.19160000E+01
.17900000E+01	.16710000E+01	.15600000E+01	.14550000E+01
.84000000E+00	.80760000E+00	.78240000E+00	.76440000E+00
.75360000E+00	.75000000E+00	.75360000E+00	.76440000E+00
.78240000E+00	.80760000E+00	.84000000E+00	.87900000E+00
.92640000E+00	.98040000E+00	.10410000E+01	.11100000E+01
.11850000E+01	.12680000E+01	.13580000E+01	.14550000E+01
.15600000E+01	.16710000E+01	.17900000E+01	.19160000E+01
.20490000E+01	.21900000E+01	.23370000E+01	.24920000E+01
.26540000E+01	.28230000E+01		

AREA COMUM 51 100

-.12800000E+01	-.12200000E+01	-.11600000E+01	-.11000000E+01
-.10400000E+01	-.98000000E+00	-.92000000E+00	-.86000000E+00
-.20000000E+01	-.19400000E+01	-.18800000E+01	-.18200000E+01
-.17600000E+01	-.17000000E+01	-.16400000E+01	-.15800000E+01
-.15200000E+01	-.14600000E+01	-.14000000E+01	-.13400000E+01
-.80000000E+00	-.74000000E+00	-.68000000E+00	-.62000000E+00
-.56000000E+00	-.50000000E+00	-.44000000E+00	-.38000000E+00
-.32000000E+00	-.26000000E+00	-.20000000E+00	-.14000000E+00
-.80000000E-01	-.20000000E-01	.40000000E-01	.10000000E+00
.16000000E+00	.22000000E+00	.28000000E+00	.34000000E+00
.40000000E+00	.46000000E+00	.52000000E+00	.58000000E+00
.64000000E+00	.70000000E+00	.76000000E+00	.82000000E+00
.88000000E+00	.94000000E+00		

20PLOT X Y



75°C 1. ALTERNATIVE, 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15.

***EXEMPLO DE APLICACAO DO COMANDO SOLUCAO.SISTEMA**

FRP AS SUAS ORDENS

FX *CTL.ALTEE ,ENTRADA,CTB

PR 05*ADM,ROL,DEFINA,A 4 4.X 4.B 4.

LISTAGEM PAG 01

05*ADM,ROL,DEFINA,A 4 4.X 4.B 4.

10*E/S,LEIA,CT A B

15SOLUCAO,SISTEMA A X B

20*E/S,ESC A X B

25*CTL,ALTERE,MOD,EX,ENTRADA,MQ

PR 10*F/S,LEIA,CT A B

PR 15SOLUCAO,SISTEMA A X B

LIN. INDE 2

PR 20*E/S,ESC A X B

A 4 X 4

.33333334E+00	.11111112E+00	-.10000000E+01	.10000000E+01
-.10000000E+01	.00000000E-99	.20000000E+01	-.10000000E+01
-.66666666E+00	-.55555555E+00	-.30000000E+01	.20000000E+01
.00000000E-99	-.10000000E+01	.40000000E+01	-.20000000E+01

X 4 VALOR(ES)

.14444445E+01
.00000000E-99
-.22222222E+00
.00000000E-99

B 4 VALOR(ES)

.60000000E+01
.10000000E+01
.50000000E+01
.20000000E+01

PR 25*CTL,ALTERE,MOD,EX,ENTRADA,MQ

****EXEMPLO DE APLICACAO DO COMANDO AJUSTE, OPCAO .COMPESO****

FX *CTL.ALTERE.ENTRADA.CT
 PR 05*ADM.ROL.DEF.X 9.Y 9.W 9.A 6.

LISTAGEM PAG 01

05*ADM.ROL.DEF.X 9.Y 9.W 9.A 6.
 10*E/S.LEIA.CT X Y W
 15AJUSTE.COMPEO A X Y W
 20*CTL.ALTER.MODO.EX

PR 10*E/S.LEIA.CT X Y W

PR 15AJUSTE.COMPEO A X Y W

A POLINOMIAL DE ORDEM 5 QUE MELHOR AJUSTA A CURVA F

POTENCIA DE X

COEFICIENTE

0	.81333907E+03
1	.30330125E+04
2	-.27006387E+05
3	.11764067E+06
4	-.23656219E+06
5	.17629873E+06

PR 20*CTL.ALTER.MODO.EX

****PROGRAMA TESTE DE TEMPO DE PROCESSAMENTO****

10*CTL.ALTERE.X 1.AUMENTA #
 20*CTL.SE.X 1.ME 1001.SAL .10#
 30*MSG.TESTE DE TEMPO DE PROC#
 40*CTL.ALTERE.MODO.EXECUTE #

*CTI.ALTERE.ENTRADA.MQ

CAPÍTULO VI

CONSIDERAÇÕES FINAIS

Vantagens obtidas na presente versão.

6.1 – Conforme está relacionado na versão original (Cap. V), o sistema ERP possuía certo número de limitações devidas principalmente à pequena memória do equipamento utilizado que para aumentar a capacidade do sistema obrigou a um aumento de tempo de execução.

Na versão presente tratamos de superar aqueles inconvenientes apontados, aproveitando-nos da dupla capacidade de memória disponível, o menor ciclo de memória (10 μ s), se traduz numa velocidade de processamento muito maior (~4 vezes).

A eficiência E é dado por:

$$E = \frac{C}{C + I}$$

onde: C = tempo gasto na execução dos subprogramas précompilados e I = tempo gasto na interpretação das instruções.

Nesta expressão o valor de I diminui em relação ao de C pois que o tempo gasto na interpretação da instrução é menor pelos motivos seguintes:

- 1º INTER residente
- 2º Os subprogramas ERP: não estão muito segmentados.
- 3º Não existe necessidade de usar muitas instruções continuação, pelo grande aumento que houve no espaço reservado para cada instrução C.
- 4º Menor tempo de acesso ao disco.

6.2 – Para teste, processamos um programa (vide pag. 93.) registrado nas duas máquinas. Obtendo-se os seguintes resultados:

IBM 1620 mod 1 com 20 k deu 15m 16 seg de processamento.

IBM 1620 mod 2 com 40 k deu 2m 16 seg de processamento.

6.3 – O fato de manter o Interpretador residente na memória reflete-se numa sensível diminuição do tempo de análise da instrução porque não mais é necessária, todas as vezes que a presença do INTER seja requerida, procurá-lo no disco. O controle entre a execução de dois comandos é passado diretamente por transferência dentro da memória de núcleos. Evita-se assim a procura realizada pelo braço portador dos cabeçotes de leitura, do cilindro onde está localizado o INTER.

6.4 – Um dos comandos que mais é utilizado juntamente com o comando *ADM é aquele que executa as operações de entrada e saída (*E/S, o ERP2) constituído originalmente por dois subprogramas. Na nova versão estes foram reunidos em um único subprograma, diminuindo, portanto, o tempo total de processamento.

6.5 – Foram feitas todas as modificações apontadas na versão original, referentes às áreas e espaços reservados para parâmetros, tamanho das instruções, etc. Todas visam uma redução do tempo gasto na interpretação. A área AC foi triplicada o que implica numa nova redução do tempo de execução, pois não será necessário o arquivamento de dados não utilizados no momento para deixar espaço livre para novos dados.

6.6 – Achamos que um passo muito importante foi deixar a área de Dados e as Fotos numa área permanente do disco. Assim, é possível a interrupção do processamento em qualquer instante sem se perder os dados já calculados e a etapa de execução já alcançada, podendo ser continuado em qualquer outro instante.

6.7 – Com as transformações realizadas, o sistema ERP torna-se muito flexível, e com uma capacidade de operação elevada que permite ao pesquisador uma comunicação fácil com o computador.

6.8 – Para aumentar ainda mais sua capacidade foram criados os vários comandos do usuário que tratam com as operações de matrizes, resolução de sistemas, localização de raízes de polinômios etc. Assim o sistema já possui os meios necessários para que o usuário desde o início possa realizar as aplicações mais freqüentes.

6.9 – Comprovamos que o sistema constitui, alcançando a finalidade para o qual foi criado, um elemento poderoso nas mãos dos pesquisadores do INSTITUTO DE ENERGIA ATÔMICA para realizarem cálculos rápidos dentro das pesquisas em andamento, permitindo verificar a boa escolha do caminho seguido ou realizar alterações a tempo, quando comprovarem a existência de algum erro.

6.10 – O sistema é particularmente útil às pessoas que não conhecem nenhuma linguagem de programação, como o FORTRAN, pois rapidamente aprendem as poucas restrições que o sistema possui e praticamente escrevendo em português transmitem ao computador comandos que a máquina prontamente analisa fornecendo uma resposta adequada. Temos como exemplo, o caso de um pesquisador que devia calcular a atividade dos isótopos do Telúrio. A expressão a calcular, se bem que bastante simples, requer o cálculo de vários exponenciais, para 20 casos diferentes que utilizando uma máquina de calcular comum e consultando tabelas, consumiria muitas horas e com probabilidade de cometer algum erro. Ora, o pesquisador em questão não tinha conhecimento de programação, então foi convidado a se sentar perante a máquina de escrever do console do computador. Após receber algumas instruções sobre o ERP, em menos de uma hora tinha feito seus próprios cálculos e aprendido a trabalhar com o sistema.

Também comprovamos a grande utilidade nos casos em que sejam necessários realizar cálculos em que o valor dos parâmetros é praticamente desconhecido e deve-se proceder por tentativas, como é o caso do cálculo de um reator em que é necessário determinar a potência máxima de um canal. De fato, fornecido um valor inicial do parâmetro não conhecido, a vazão do refrigerante no caso, imediatamente comprovamos o valor da potência para aquele dado e decidimos no momento qual será o novo valor a fornecer. Assim por sucessivas tentativas com novos valores de entrada poderemos chegar ao valor da potência máxima, sem ter necessidade de preparar previamente um programa em linguagem científica de alto nível mais ou menos sofisticado nem passar pelas sucessivas etapas de compilação, execução e modificações requeridas.

6.11 – Achamos que atingimos o objetivo proposto, que foi de dotar os pesquisadores com um sistema ágil, flexível, modular, de fácil aprendizado, relativamente rápido e conversacional utilizando o

português corrente. O pesquisador inteligente e criativo pode aproveitar muito bem as possibilidades do sistema facilitando grandemente o seu trabalho de pesquisa.

Estamos de acordo com o criador do sistema, que a sua implementação no computador IBM 1130 é fácil e pode ser imediato. O sistema possui ainda, uma vantagem muito importante em relação a outras linguagens conversacionais como o APL, a de que não requer nenhuma máquina com tipos especiais para sua operação.

**ORGANIZAÇÃO DA
MEMÓRIA**

ANEXO A

PROGRAMAS

0000	SISTEMA	IBM
7000	ERP	(80)
7500	PEJUE	(242)
7822	SUBROTINA G3 DE INDEXAÇÃO (utilizada pelo FORTRAN) (810)	
8332	FLIPER	(601)
8633	TABELA DO FLIPER (48 x 80, + 1)	
9984	INTERPRETADOR (LOCAL) E SUBROTINAS (LOCAL'S) de produção de comandos (4144)	
		14127

1627	ÁREA DE LITURA	13797
	ÁREA DE PARÂMETROS (BOR-4)	13798
	ÁREA DE COMUNICAÇÃO (BOR-4)	13998
	INDIC. PARÂMETROS TRANSMITIDOS	14119
		14180
	TABELA DE INSTRUÇÕES - REGRAS (24 x 60)	14889
	TABELA DE RETRANSMISSÃO	15048
	TABELA DE SÍMBOLOS (60 x 20)	15649
	TABELA DE COMANDOS (28 x 10)	15999
	ÁREA COMUM DE DADOS (200 x 10)	17000
16950		

1800
PAGINAS

TABELAS
E
ÁREAS

ANEXO B

ANEXO C

ORGANIZAÇÃO DA ÁREA
DE TRABALHO DO
DISCO
(24 cilindros)

<p>CILINDROS C - 9</p> <p>ÁREA PARA LOCAL'S (10 CILINDROS)</p> <p>SETORES: 0 - 1000</p>
<p>CILINDROS 10-11 ARQUIVO PARA 20 PÁGINAS DE PROGRAMA</p>
<p>CILINDROS 12 - 21</p> <p>ARQUIVO DE DADOS (10 CILINDROS)</p> <p>SETORES: 2400 - 4300</p>
<p>CILINDROS 22 - 23</p> <p>ARQUIVO PARA 4 FOTOS</p>

ANEXO - D

ÁREA DE COMUNICAÇÃO

SÍMBOLO	CONTEÚDO DO CAMPO
IND1	Para uso do programador. Pode ser utilizado como indicador , indexador, contador, etc...
IND2	Idem
IND3	Idem
IND4	Para uso do sistema . Seu emprego é explicado nos §6 e §8 do Cap. II.
ERPO	Código de erro
K1	Para controle de seqüência no modo PR. Seu emprego é explicado no §6 do Cap. I. (versão original)
K2	Idem
K3	Idem
K4	Idem
L	Utilizado como campo auxiliar provisório. Pode ser usado pelo programador para uso temporário.
INDAC	Indica próximo campo vago na AC .
INDTS	Indica próxima linha da TS
INDPG	Número da página onde se encontra a instrução sendo executada.
INDSK	Indica próximo setor vago no disco para arquivar variáveis.
IDE	Indica qual o dispositivo vigente de entrada
IDS	Indica qual o dispositivo vigente de saída
MODO	Indica o modo vigente

ÁREA DE COMUNICAÇÃO - (Continuação)

SÍMBOLO	CONTEÚDO DO CAMPO	Antiga	Nova
MAXAC	Capacidade máxima da AC	(= 300)	(900)
MAXTS	" " " TS	(= 50)	(50)
MAXTI	" " " TI	(= 24)	(60)
MAXDK	Número de setores do disco <u>re</u> servados para arquivo de vari áveis	(=2000)	(2000)
MAXAP	Capacidade máxima do AP	(= 50)	(98)
MAXPG	Número máximo de páginas	(= 20)	(20)
INDTI	Indica próxima linha da TI		
NPG	Número da página na memória		
KX1	Usado para transmitir número de ordem do próximo parâmetro a ser analisado entre os ERPn.		
KX2	Usado para transmitir informação sobre o erro para o ERP4.		
KX3	Usado para comunicação entre os ERPn.		

ANEXO E

USOS DO IND4

- 1a. Posição Se chave 3-on , é ligada pelo interpretador e desligado pelo leitor após leitura.
Significa intervenção do operador.
Ligado pelo ERP3 para possibilitar intervenção do operador na correção de erros.
Desligado pelo leitor após leitura.
Ligado pelo conversor para indicar existência de erro e desligado à saída do conversor.
- 2a. Posição Ligado por uma subrotina ERPXX para pedir execução imediata de outra subrotina ERPTY.
Desligado pelo interpretador.
- 3a. Posição Ligado pelo interpretador ao encontrar uma vírgula na instrução.
Desligado pelo interpretador.
- 4a. Posição Ligado por um ERPXX subsidiário.
Desligado ao sair do ERPTY principal.

4a. pos.	3a. pos.	2a. pos.	1a. pos.
----------	----------	----------	----------

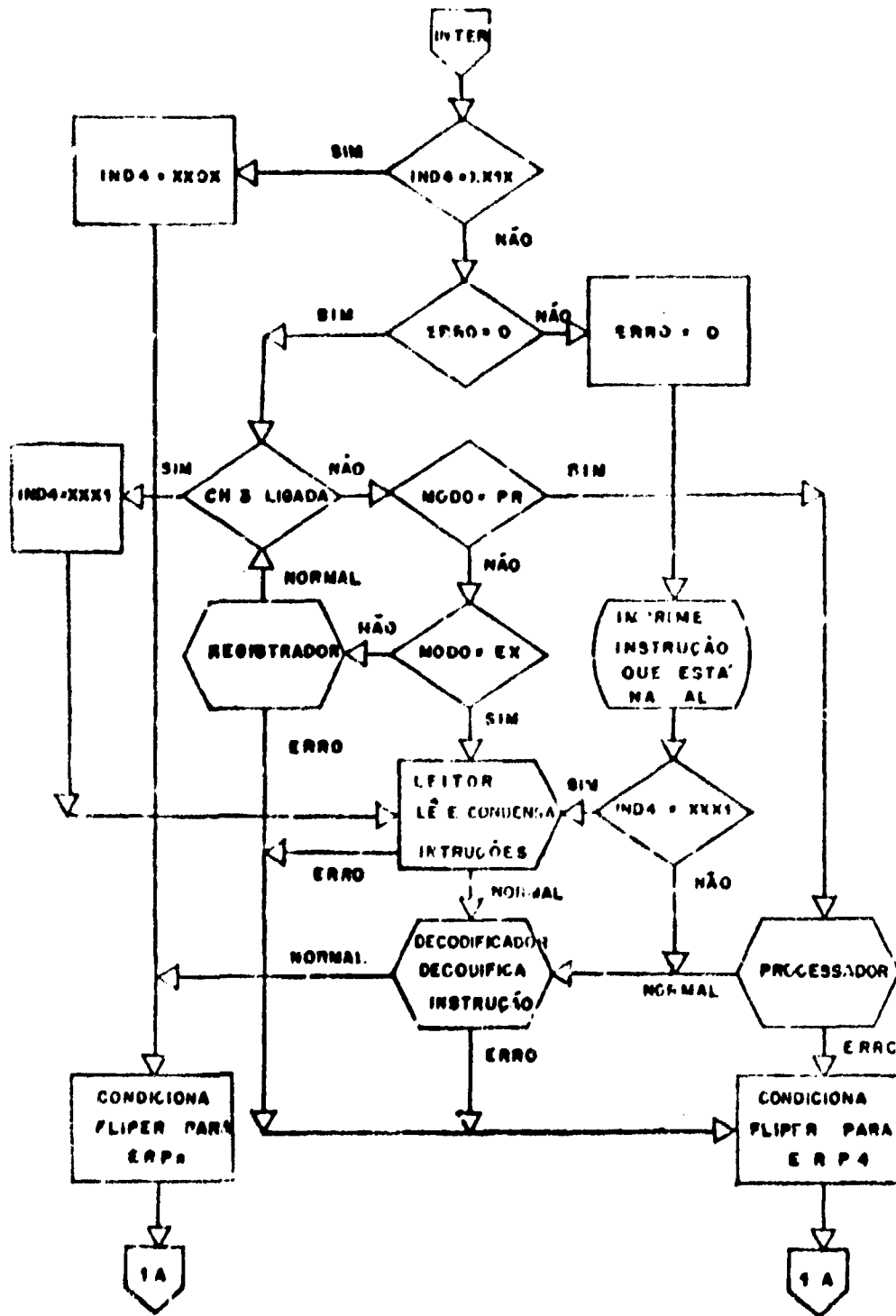
IND4

LIGADO = 1

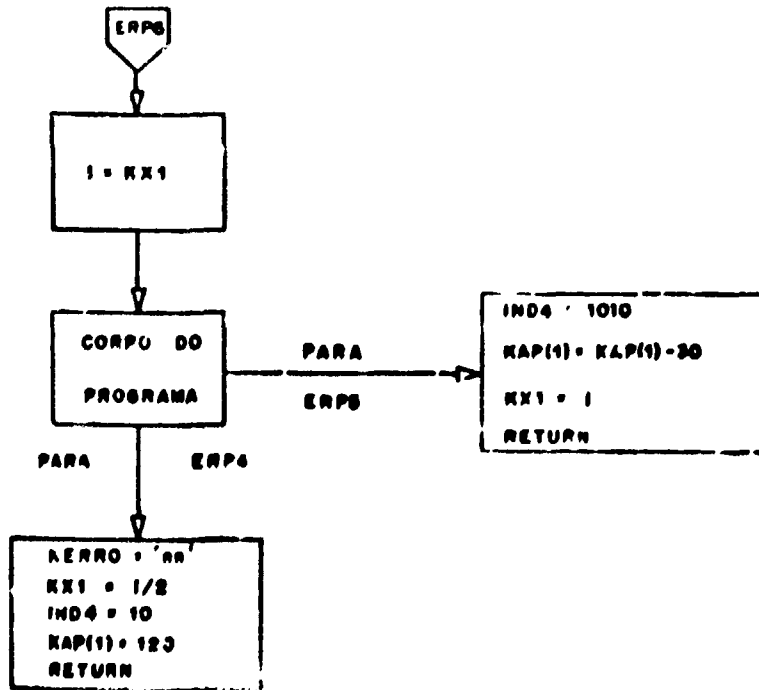
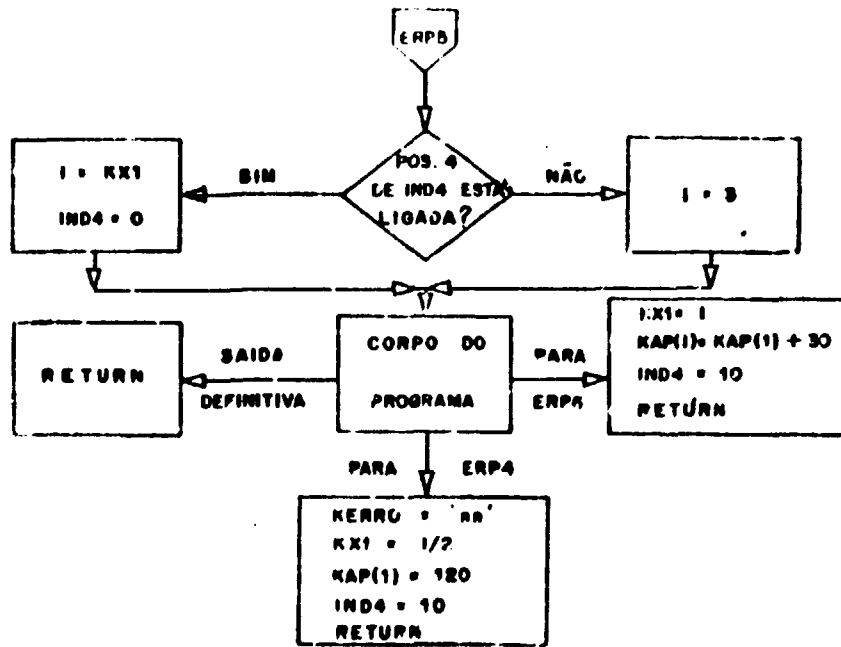
DESLIGADO = 0

ANEXO F

INTERPRETADOR DIAGRAMA RESUMO



ANEXO H



DISTRIBUIÇÃO
DA MEMÓRIA

ÁREA DE
PROGRAMAS

ANEXO 1

00000		
	MONITOR I COM IMPRESSORA	00000
00103	ERP (104)	00203
00204	PEQUE (242)	00410
00440	INTER (3702)	10237
10230	SUBROTINA DE INDEXAÇÃO (810)	10407
10740	FLIPER (300)	13047
13040	TABELAS DOS PROGRAMAS LOCAL (1000) + 8	14040
14000	ÁREA DOS PROGRAMAS TIPO LOCAL (10000)	24000

ANEXO 2

**DISTRIBUIÇÃO DA
MEMÓRIA**

**ÁREA DE
TABELAS**

PÁGINA

					24948	
	ÁREA 5:			ÁREA DE LEITURA	(102)	25109
	24947					
	25110			ÁREA DE PARÂMETROS		25501
				(98 x 4)		
	25702			ÁREA DE COMUNICAÇÃO:		25893
				(28 x 4)		
				25614		25617
				PARÂMETROS TRANSMITIDOS		
				(10 x 4)		
	25887					
				TABELA DE INSTRUÇÕES		
				REGISTRADAS		
				(164 x 24)		25889
	25650			TABELA DE RETORNO		25652
				(60)		
	29680			TABELA DE SIMBOLOS		30049
				(20 x 50)		
	30657			TABELA DE COMANDOS		30999
				(150)		
	31000					
				ÁREA COMUM		
				DE BANCOS		
				(900)		
						30999

DISTRIBUIÇÃO DAS

ÁREAS DO

DISCO

ANEXO 3

10	SUBROTINAS ARITMÉTICAS DE ENTRADA/SAÍDA E SUBPROGRAMAS TIPO LOCAL (20 CILINDROS)	14
20	PÁGINAS (4 CILINDROS)	23
24	TABELAS DIM E DE EQUA- LÊNCIA DO MONITOR (2 CILINDROS)	25
28	ÁREA DE DADOS (10 CILINDROS)	35
38	F C T O S (4 CILINDROS)	38
40	ÁREA DE ARQUIVO DOS PROGRAMAS (40 CILINDROS.)	79
80	Ocupado pelo sistema monitor (20 CILINDROS)	86

ANEXO 4

CONTEÚDO DE 'X' 'n'

108

X	Conteúdo	X	Conteúdo
1	IND1	29 a 38	Parâmetros Transmítidos
2	IND2	39 + 41 n	Identificação da instrução "n" ($0 \leq n \leq 23$)
3	IND3		
4	IND4	1022 - 1023	TAB. REF. 19 par
5	ERRO	1024 - 1025	29 par
6	K1	1026 - 1027	39 par
7	K2	1028 - 1029	49 par
8	K3	1030 - 1031	59 par
9	K4	1032 - 1033	69 par
10	L	1034 - 1035	79 par
11	INDAC	1035	livre
12	INDTS		TABELA DE SÍMBOLOS
13	INDPG		($0 \leq n \leq 23$)
14	INDSK		
15	IDF	1037 + 5 n	Nome da variável
16	IDS	1038 + 5 n	Localiz. na área comum
17	MODO	1039 + 5 n	Número de linhas
18	MAXAC	1040 + 5 n	Número de colunas
19	MAXTS		
20	MAXTI	1041 + 5 n	Localização no disco
21	MAXDK		
22	MAXAP		
23	MAXPG		
24	INDT1		
25	NPG		
26	KX1		
27	KX2		
28	KX3		

DESCRIÇÃO FORMAL DOS COMANDOS DO SISTEMA

ALM

.DEFINA	$\left[\cdot k_1 \left[n_1 \left[n_2 \right] \right] \right]$	etc.
.APAGUE	$\left[v \right]$	etc
.LIBERE	$\left[v \right]$	etc
.ARQUIVE	$\left[v_1 \left[v_2 \right] \text{ etc} \right]$	
	$\left\{ \begin{array}{l} \text{.PÁGINA} \\ \text{.FOTO } n \end{array} \right\}$	etc
.ATIVE	$\left[v_1 \left[v_2 \right] \text{ etc} \right]$	
	$\left\{ \begin{array}{l} \text{.PÁGINA } n \left[l \right] \\ \text{.POTO } n \left[l \right] \end{array} \right\}$	etc
.ABRA	.PÁGINA n	
.VIRE		
.SUBROTINA	$\left[\begin{array}{c} n \\ v_1 \end{array} \right] \left[\cdot l \right] \left[\begin{array}{c} n_2 \\ \cdot l_2 \\ v_2 \end{array} \right]$	etc .
.RETORNE		
.ROL		
.	$\left[\text{INICIALIZA} \left[\cdot \text{CMDOS} \right] \right]$	

etc

.CTL	.ALBRE	{	}	etc	etc
	.SR	{	}		
	.DESVIE	{	.l ₁ . . .l ₁ .l ₂ .l ₃		
	.SALTS .L				
	.FII:				

.ALBRE	.MODO	{	.BX .RG .PR	}	et .
	.ENTRADA	{	.MQ .CT	}	
	.SAÍDA	{	.MQ .CT .TM	}	
	.X n	{	.AUMENTA .DIMINUE	}	

.SE	v ₁ [n ₁ [n ₂]]	.NÃO	{	.MAIOR .MENOR . =	}	v ₂ [n ₃ [n ₄]]
	.X n ₁ [.NÃO]	{	.MAIOR .MENOR . = .DIVISÍVEL	}	[.X n ₂] [n]	

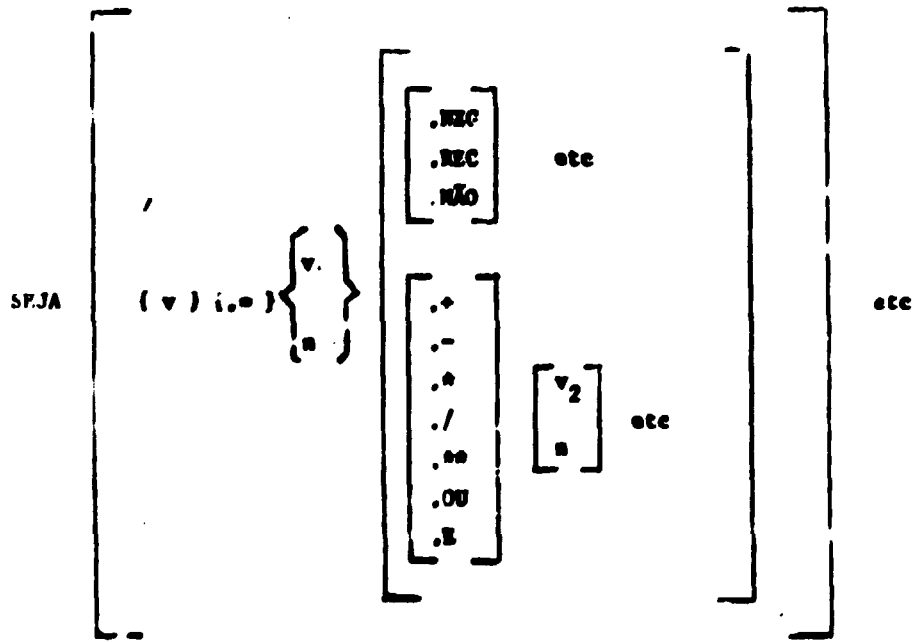
.LITA	$\{v\}$ etc	etc
	.CT $\{v\}$ etc	
	.HQ $\{v\}$ etc	
.PORZVA	$\{v\}$ etc	etc
	.CT $\{v\}$ etc	
	.HQ $\{v\}$ etc	
	.IM $\{v\}$ etc	
.KISA	.AK	etc
	.PT	
	.TR	
	.TS $\{v\}$ etc	
	.AC $u_1 u_2$	
	.X n	
.ALTI BE		

.CONTONHA	$\{v\}$ etc	.EM	$\{l\}$	
.DESCOMPONHA	v	.EM	$\{l\}$	$n\}$ etc .
.GRUPE	$\{v\}$ etc	.EM	$\{l\}$	
.PARTA	v	.EM	$\{l\}$	$n\}$ etc .
.TRANSPONHA	$\{v\}$ etc			

```

*CAL < v > , = < ea >
< ea > ::= < e > | < ea > . + < e > | < ea > . - < e >
< e > ::= < l > | < e > . * < l > | < e > . / < l > |
< f > ::= < pr > | < l > . ** < pr >
< pr > ::= < e > | < v > | < dl > | . ( < ea > . )
< v > ::= < l > | < v > < ct >
< ct > ::= < l > | < d >
< e > ::= < d > | < e > < d >
< l > ::= A | B | ... | Z
< d > ::= 0 | 1 | ... | 9
< dl > ::= < LOG > | < LN > | < EXP > | < DX > | < PQ > | < ARCTAN > |
          < SEN > | < TAN > | < ABS >
    
```

DESCRIÇÃO FORMAL DOS COMANDOS DO USUÁRIO



TRAP $\left[\begin{array}{l} \text{TRANSLADE } v_1 \ v_2 \\ \text{.MULTIPLIQUE } v_1 \ v_2 \\ \text{.NEGUE } v \\ \text{.RECIPROQUE } v \\ \text{.QUADRE } v \end{array} \right] \text{ etc}$

LOCR (v) etc

PLOT (v₁ v₂) etc

MATRIZ $\left[\begin{array}{l} \text{.SOME } v_1 \ v_2 \ v_3 \\ \text{.SUBTRAI } v_1 \ v_2 \ v_3 \\ \text{.MULTIPLIQUE } v_1 \ v_2 \ v_3 \\ \text{.DETERMINANTE } v_1 \ v_2 \\ \text{.TRANSPONHA } \left[\begin{array}{l} v \\ v_2 \end{array} \right] \text{ etc} \\ \text{.INVERTA } v \end{array} \right] \text{ etc}$

SOLUÇÃO $\left[\begin{array}{l} (v_1) \\ (v_2) \\ (v_3) \end{array} \right] \text{ etc}$

AJUSTE $\left[\begin{array}{l} \text{.COMPESC } (v_1 \ v_2 \ v_3 \ v_4) \\ \text{.SEMPESO } (v_1 \ v_2 \ v_3) \end{array} \right] \text{ etc}$

CARTÃO INTERVENÇÃO OPERADOR

1625521000501525516000011625113001204908172+

ANEXO 6

RELAÇÃO DAS MENSAGENS DO PRPO

01 - ESTouro DA ÁREA DE PARÂMETROS

Forma utilizado na instrução e suas continuações mais de 98 parâmetros.

02 - COMANDO IRRECONHECÍVEL

Comando inexistente na tabela de comandos; comando escrito errado;
comando começa antes da terceira posição; falta o comando.

03 - VARIÁVEL 'v' NÃO DEFINIDA

No lugar de 'v' aparece o nome da variável.
A variável não consta da tabela de símbolos; literal sem ponto foi interpretado como variável.

04 - VARIÁVEL 'v' LIBERADA

Tentativa de utilizar uma variável liberada numa instrução cujo comando não é ADM

05 - ESTOURC DA PÁGINA NO REGISTRO

Tentativa registrar além de 24 instruções na página

06 - FALTA PRIMEIRA IDENTIF. DE DESVIO

A rotina PROCESSADOR do INTERPRETADOR ao reconhecer uma situação de desvio (K1=0), não encontra indicação de para onde desviar (K2=0) .

07 - FALTA TERCEIRA IDENTIF. DE DESVIO

A rotina PROCESSADOR do INTERPRETADOR ao reconhecer uma situação de desvio (K3=0 é igual a identificação da última instrução executada), não encontra indicação de para onde desviar (K4=0)

08 - PRIM. IDENT. DESVIO INEXISTE NA PÁG

Não existe na página instrução com identificação igual a K2.

09 - SEGDA IDENT. DESVIO INEXISTE NA PAG

Não existe na página instrução com identificação igual a K3.

10 - TERC. IDENT. DESVIO INEXISTE NA PAG

Não existe na página instrução com identificação igual a K4.

11 - ESTOURO DA PÁGINA NO PROCESSAMENTO

A última instrução da página não é de desvio e não transfere o controle ao operador.

Tentativa de processar além da última instrução registrada na página.

12 - INSTRUÇÃO EXCEDE 80 CARACTERES

Só pode ocorrer por entrada pela máquina de escrever.

Verificar e tentar reparar os estragos causados às áreas via linhas.

13 - INSTR. A REGISTRAR EXCEDE 80 CARACT

Pela máquina de escrever: número excessivo de caracteres.

14 - INSTR. CONTINUAÇÃO NÃO COMEC. POR //

Auto-explicativa

15 - PARÊNTESES NÃO CASAM

Numa expressão aritmética o número de parênteses à esquerda não são iguais ao número de parênteses à direita.

16 - TENTANDO DIVIDIR POR ZERO

Tentativa de dividir por zero numa expressão aritmética do comando *CALCULE.

17 - CONSULTE TABELA - ERRO TIPO XXX

Nesta mensagem XXX é um código dado pelo usuário para um tipo de erro não catalogado nesta série de mensagens.

Tabela destes tipos de erro devem existir atualizadas.

18 - ESPERADO LIT. ENCONTRADO XXX.

Nesta mensagem XXX está no lugar de VAR, NUM ou FIN.

Provavelmente faltou um ponto final de uma lista de parâmetros que podem incluir literais ou faltou um ponto definidor de um literal.

- 22 - ESPERADO NUM. ENCONTRADO XXX.
Nesta mensagem XXX está em lugar de VAR, LIT ou FIM.
- 23 - ESPERADO VAR. ENCONTRADO XXX.
Nesta mensagem XXX está em lugar de LIT, NUM ou FIM.
- 24 - ESPERADO NUM./VAR. ENCONTRADO XXX.
Nesta mensagem XXX está em lugar de LIT ou FIM.
- 25 - ESPERADO NUM./LIT. ENCONTRADO XXX.
Nesta mensagem XXX está em lugar de VAR ou FIM.
- 26 - ESPERADO LIT./VAR. ENCONTRADO XXX.
Nesta mensagem XXX está em lugar de NUM ou FIM.
- 27 - LITERAL INVALIDO NESTE PONTO
Literal escrito errado; omitido algum literal intermediário;
transmissão errada de parâmetros para subrotina.
- 28 - FIM DE INSTRUÇÃO INESPERADO
A instrução exige ainda algum parâmetro. Por exemplo, uma lista de parâmetros que pode conter literais não termina com ponto mas com o fim da instrução.
- 29 - PARÂMETRO INVALIDO NESTE PONTO
O teste incompleto feito pela subrotina ERPN detectou que o parâmetro encontrado não coincide com nenhum admissível mas não verificou qual a natureza do parâmetro encontrado, se variável, literal ou número.
Examinar a instrução.
- 30 - TENTADO RE-EPINIR VARIÁVEL
A variável já se encontra na tabela de símbolos -
Verificar se a instrução não faz parte de um ciclo, ou se a lógica do programa não a chama para execução mais de uma vez, -
Verifique se duas variáveis não têm os dois primeiros caracteres em comum.
- 32 - ESTOURO DA TABELA DE SÍMBOLOS
Tentado definir além de 50 variáveis

Verifique se existem variáveis que podem ser apagadas.

33 - ESTOURO ÁREA DADOS. DEFICIT = XXX

Tenta mensagem XXX está em lugar do número de campos de área comum de dados que faltam, para definir ou ativar uma variável liberada.

No cálculo de Determinante a área auxiliar utilizada, não cabe na AC.

Verifique que variáveis podem ser liberadas.

34 - NÚMERO DE PÁGINA INVÁLIDO

Na instrução há referência a uma página cujo número está fora do intervalo 1 - 20

35 - ESTOURO TAB. DE PARAM. TRANSP. P/SBR

Tentado transmitir mais que 10 parâmetros; a lista de parâmetros a transmitir não termina com ponto.

36 - ESTOURO DA TABELA DE RETORNO

Mais que 7 chamadas de subrotinas foram executadas antes do primeiro retorno.

37 - NÚMERO DE FOTO INVÁLIDO

Na instrução há referência a uma foto cujo número está fora do intervalo 1 - 4.

38 - INCONSISTÊNCIA NA INDEXAÇÃO DE VAR

O primeiro ou segundo índice estoura o número de linhas ou colunas de variável.

39 - TENTADO ALTERAR ÁREA PROIBIDA

Tentado redefinir ou operar como ponto fixo um campo em ponto flutuante.

40 - TENTADO ARQUIVAR VARIÁVEL LIBERADA

Auto-explicativa.

41 - TENTADO ATIVAR VAR. NÃO ARQUIVADA

Auto-explicativa.

42 - ESTOURO ARQUIVO DE DADOS DO DISCO

Tentado utilizar para arquivo de dados mais de 2000 setores do disco.

43 - ERRO DEFINITIVO NA LEITURA DO DISCO

Cinco tentativas não conseguiram efetuar a leitura do disco. A informação é considerada perdida.

44 - ERRO DEFINITIVO - GRAVAÇÃO DO DISCO

Esta mensagem não ocorre no arquivamento de variáveis, pois que após 5 tentativas frustradas outros setores do disco são utilizados.

No caso de arquivamento de páginas ou fotos, o recurso é mudar o número da página ou da foto.

49 - CONSULTE TABELA - ERRO TIPO XXX

Nesta mensagem XXX é um código dado pelo usuário para um tipo de erro não catalogado nesta série de mensagens.

Tabelas destes tipos de erros devem existir atualizadas.

50 - INTERVENÇÃO DO OPERADOR

Esta mensagem é dada quando o operador intervir forçando a rotina de interrupção que é carregada nas primeiras posições da memória.

REFERÊNCIAS BIBLIOGRÁFICAS

1. Abramowitz, Milton and Stegun, Irene A. – Handbook of Mathematical Functions – V.S. Department of Commerce, National Bureau of Standard, 1964.
2. Barros, Ivan de Queiróz, E. R. P. um sistema para interação homem-Computador, Teses de Docência-livre da disciplina "Cálculo Numérico" da EPUSP, 1968.
3. Cavada, Izvoras Cristian, Le langage SPRINT/1 pour l'évolution dynamique des formules mathématiques par dialogue direct avec un ordinateur (CAE 90-50). These Rapport CEA-R-3596, 1968.
4. Davis, Ruth M., Programming Language Processors – Advances in Computer, volume 7, Academic Press, 1966.
5. Grenberg, Martins e outros. On Line computation and Simulation OPS-3 The M.I.T. Presse, 1965.
6. Hastings, Cecil Jr., Aproximation for Digital Computers Princeton University Press, fifth edition, 1966.
7. IBM – 1620 Monitor System – Reference Manual.
8. IBM Reference Manual – 1620 – Data Processing System.
9. IBM 1620/1710 – Symbolic Programming System.
10. IBM 1311 – Disk Storage Drive, Model 3.
11. IBM 1620 – Gotran Interpretative Programming System.
12. Nydegger, Adolph C., An Introduction to computer Programming Addison – Wesley Publishing Company, 1968.
13. Pennington, Ralph H., Introductory Computer methodes and numerical analysis, The Macmillan Company, New York, 1966.
14. Proceedings IBM Scientific Computing Symposium Digital Simulation of Continuous Systems. IBM Data Processing Division, White Plains, New York, 1967.
15. Ross, Daniel, An Integrated Computers System, separata AFIPS-Conference Proceedings - Vol. 27 Parte 1 pag. 423, 1965.
16. Sadosky, Manuel, Cálculo numérico y gráfico. Ediciones Libreria del Colegio, cuarta edición, 1962.
17. Kelly, Louis G., Handbook of Numerical Methods and Applications – Addison – Wesley Publishing Company, 1967.
18. Klaus Reinach e Collatz, L., Notas do Curso de Cálculo Numérico – Escola Politécnica da Universidade de São Paulo, 1954.
19. Programoteca do Serviço de Cálculo Analógico e Digital do I.E.A., foram consultados e adaptados os seguintes programas:
1 – POLOR – Ajuste de curvas por polinômios ortogonais Método dos mínimos quadrados, por M.E. Hehl

- 2 – PLOTTER – Gráfico pela impressora por Lúcia Faria Silva e Cíbar Cáceres Aguilera
- 3 – DETER – Cálculo de Determinantes por Lúcia Faria Silva.

20. Hilsenrath, Joseph, A computer Program For Statistical and Numerical Analysis – National Bureau of Standard, 1968.

21. Orr, William D., Conversational Computers – John Wiley and Sons, Inc, 1968.

