UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

RELATÓRIO FINAL

ALUNA: MARIA DE FÁTIMA BRAGA

...a, 10 de dezembro de 1.976.

DA - Estagiária Maria de Fátima Braga, nº de matrícula 7611156-2, aluna do Centro de Ciências e Tecnologia -UFPB -Departamento de Sistemas e Computação - Coordenação do Curso de Processamento de Dados.

AO - Centro de Ciências e Tecnologia -Departamento de Sistemas e Computação - Supervisão do Estágio Supervisionado.
Prof. Raimundo Haroldo Catunda.

Senhor Supervisor,

Em obediência às normas estabelecidas pelo programa Universidade/Empresa apresento a Vossa Senhoria o RELATÓRIO FINAL DO ESTÁGIO SUPERVISIONADO, realizado junto a CODATA - Companhia de Processamento de Dados da Paraiba.

Mª de Fátima Braga.
ESTAGIÁRIA.

Francisco Machado.
ANALISTA -SUPERVISOR.

Biblioteca Setorial do CDSA. Março de 2021.

Sumé - PB

P R O J E T O

PROGRAMAÇÃO PADRONIZADA USANDO TÉCNICAS AVANÇADAS

EM LINGUAGEM COBOL

DATA

APRESENTAÇÃO

Este projeto tem como finalidade de uniformizar as regras básicas da programação utilizando a linguagem COBOL, tendo assim um melhor efeito na depuração de erros, alterações, etc.

O projeto em definição está dividido em:

- PRIMEIRA PARTE - definição da estruturação das três primeiras divisões. Foi elaborada por:
  - Maria de Fátima Braga e Walmar Batista Rodrigues.
- SEGUNDA PARTE - estruturação da PROCEDURE DIVISION. Foi elaborada por:
  - Ricardo José Fernandes Aragão e Arnóbio Ferreira da Nóbrega.

DATA

PRIMEIRA PARTE

1. DIVISÃO

1.1 - Um EJECT para cada divisão

1.2 - Codificada na coluna 8

1.3 - Entre o nome e outro da divisão 2 espaços em branco

1.4 - Abaixo do nome da divisão comentários em linha de " * "

1.5 - De uma divisão para uma seção ou parágrafo um SKIP3

2. SEÇÃO

2.1 - Codificada na coluna 9

2.2 - Entre o nome e outro da seção 2 espaços em branco

2.3 - De uma seção para um parágrafo ou para outra seção um SKIP3

3. PARÁGRAFO

3.1 - Codificado na coluna 10

3.2 - Entre o nome e outro do parágrafo e de seus conteúdos 2 espaços em branco

3.3 - O conteúdo do parágrafo será codificado na coluna 16

3.4 - Cada função do parágrafo em linhas diferentes

ATA

4. CONSTRUÇÃO DE IDENTIFICADORES USADOS NOS PROGRAMAS

4.1 - Nome de Arquivos

4.1.1 - Com 8 caracteres

4.1.2 - 5 posições para o código do sistema

4.1.3 - 1 posição para o tipo do arquivo, que pode ser:

X - PERMANENTE

Y - TRANSITÓRIO

Z - TEMPORÁRIO

4.1.4 - 2 últimas posições para o código do arquivo

4.2 - Nome de Registros e Campos

4.2.1 - No máximo de 8 caracteres

4.2.2 - Devem ser significativos

4.3 - Chaves, Acumuladores e áreas de salvamento de registros

4.3.1 - No máximo de 8 caracteres

4.3.2 - Devem ser significativos e constar um prefixo do nível Ø1, como mostra o exemplo abaixo:

- ACUMULADORES	- AC
- CHAVES	- CH
- ÁREAS DE SALVAMENTO DE REG.	- AS

5. COMENTÁRIOS

5.1 - Fica a critério do programador

A SEGUIR A DESCRIMINAÇÃO E EXEMPLOS DE CADA

DIVISÃO

1. IDENTIFICATION DIVISION

1.1 - Codificação ID DIVISION

1.2 - Constará apenas de um parágrafo 'PROGRAM-ID'

1.3 - Comentários como mostra o quadro abaixo; O ítem observação do comentário constará no máximo dez linhas.

```
 ID  DIVISION.
***  *********

    PROGRAM-ID.
         ZZ99804.

*********************************C O D A T A*********************************
*
*
*
*     CLIENTE       - DEPTO DE VENDAS INDUSTRIAIS
*
*     SISTEMA       - SADOIA - SISTEMA INTEGRADO DE PESSOAL
*
*     PROGRAMA      - SADOIA05 - ATUALIZA CADASTRO DO DPJ
*
*     ANALISTA      - TARCISIO
*
*     PROGRAMADOR   - FATIMA E WALMAR
*
*     DATA          - 27 DE NOVEMBRO DE 197[illegible]
*
*     OBSERVACAO    - ESTE PROGRAMA LE UM ARQUIVO CADASTRO EM FITA E UM
*                   - ARQUIVO EM CARTAO, EMITINDO UM RELATORIO MENSAL
*                   - COM A SITUACAO DE VENDAS REFERENTE AOS PRODUTOS
*                   - INDUSTRIAIS DE CADA ESTADO.
*
*
*********************************C O D A T A*********************************
```

2. ENVIRONMENT DIVISION

2.1 - Exceto as clausulas SELECT e ASSIGN do FILE-CONTROL as outras clausulas restantes são co codificados na coluna 20, e em linhas dife - rentes.

```
 ENVIRONMENT  DIVISION.
************  **********

 CONFIGURATION  SECTION.

  SOURCE-COMPUTER.
       IBM-360-G40.
  OBJECT-COMPUTER.
       IBM-360-G40.
  SPECIAL-NAMES.
       C01  IS  NOVA-PAGINA
       DECIMAL-POINT  IS  COMMA.

 INPUT-OUTPUT  SECTION.

  FILE-CONTROL.

       SELECT  ZZZAT01  ASSIGN  TO  SYS006-UR-2540R-S.
       SELECT  ZZZAT03  ASSIGN  TO  SYS008-UR-1403-S.
       SELECT  ZZZAX02  ASSIGN  TO  SYS020-DA-2314-A
               ACCESS  MODE  IS  RANDOM
               ACTUAL  KEY  IS  CHAVE.
```

3. DATA DIVISION

3.1 - Para o nível de grupo usar o 01

3.2 - Para a quebra de níveis usar de 05 em 05

3.3 - Evitar a utilização do nível 77 e usar o 01

3.4 - Clausula Picture

3.4.1 - Codificada na coluna 32

3.4.2 - O tamanho da Picture deverá ser reservado três dígitos dentro do Parêntese.

3.5 - A clausula Value, Comp. será codificada na coluna 44.

```
DATA DIVISION.
**************

FILE SECTION.

FD    ZZZAT01
01    CART-01.
      05  CAMPO-01      PIC  X(005).
      05  NOME-01       PIC  X(030).
      05  FILLER        PIC  X(045).

FD    ZZZAX02
01    DISK-02.
      05  CAMPO-02      PIC  X(005).
      05  NOME-02       PIC  X(030).
      05  SAL-02.
          10  SAL-IN    PIC  9(006).
          10  SAL-AT    PIC  9(006).

FD    ZZZAT03
      05  LINHA         PIC  X(132).
```

```
      WORKING-STORAGE  SECTION.

       01     CAB.
              05  FILLER          PIC  X(050) VALUE 'RELACAO DAS SITUACOES
      -                           'DE VENDAS'.
              05  FILLER          PIC  X(010) VALUE SPACES.
              05  DATAS.
                  10  DIA.       PIC  X(002).
                  10  BARRA      PIC  X.
                  10  MES        PIC  X(002).
                  10  ANO        PIC  X(003).

       01     ACUMULA.
              05  AC-SOMA         PIC  9(003) VALUE ZEROES.


      LINKAGE  SECTION.
       01     AREA                PIC  X(592).
```

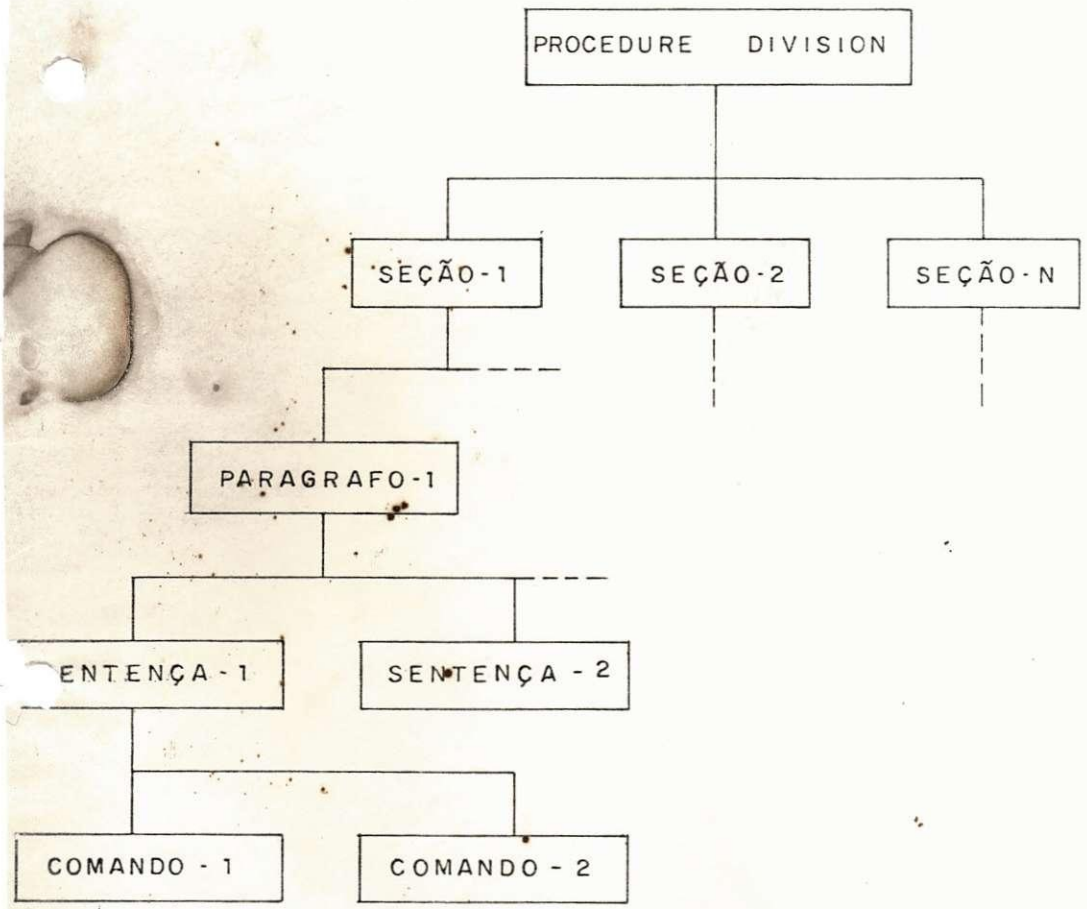
```
      WORKING-STORAGE  SECTION.

       01     CAB.
              05  FILLER          PIC  X(050) VALUE 'RELACAO DAS SITUACOES
      -                           'DE VENDAS'.
              05  FILLER          PIC  X(010) VALUE SPACES.
              05  DATAS.
                  10  DIA.       PIC  X(002).
                  10  BARRA      PIC  X.
                  10  MES        PIC  X(002).
                  10  ANO        PIC  X(003).

       01     ACUMULA.
              05  AC-SOMA         PIC  9(003) VALUE ZEROES.


      LINKAGE  SECTION.
       01     AREA                PIC  X(592).
```

Nesta primeira parte não foi utilizada nenhuma bibliografia, apenas procuramos nos informar das regras básicas utilizadas na emprêsa.

SEGUNDA PARTE

1 . PROCEDURE DIVISION.

1.1 - É nesta divisão que o programador vai codificar passo a passo o caminho que o programa deve seguir, ou seja a lógica do programa. O caminho faz parte de uma hierarquia organizacional como se vê no diagrama abaixo:



OBS: Como se vê, os comandos são agrupados em sentenças, as sentenças em parágrafos e os parágrafos em seções.

1.2 - A procedure division, deve ser codificada como as demais divisões, a partir da magem " A ",deixando dois espaços entre a palavra " PROCEDURE " e " DIVISION "

2.2. SEÇÕES:

2.1 - Uma seção é composta de um ou mais parágrafos sucessivos. O nome da seção deve ser codificado da seguinte maneira:

X99 - IDENTIFICAÇÃO SECTION

X - Este caracter especifica a sequência da section codifica-se em ordem alfabética.

99 - Caracter numérico, especificando a sequência dos parágrafos, inicia-se com ØØ (zero, zero).

IDENTIFICAÇÃO - Identifica o que será feito na seção.

OBS: A codificação é feita a partir da magem "A", deixando ' dois espaços entre a IDENTIFICAÇÃO e a SEÇÃO.

EXEMPLO:

```
PROCEDURE DIVISION.
*****************
      SKIP2
AØØ-ABERTURA-ARQUIVOS SECTION.
      OPEN INPUT CARTAO
           ITA
           OUTPUT IMPRESSORA.
AØ1-FIM-ABERTURA.
      SKIP2
BØØ-LEITURA SECTION.
BØ1-LER-CARTAO.
      READ CARTAO AT END
      GO TO BØ2-LER-FITA
      MOVE REG-CARTAO TO REG-IMPRESSORA
      GO TO BØ1-LER-CARTAO.
```

Continuação do assunto 2.

OBS: Os parágrafos são subordinados as seções com suas sentenças (grupo de comandos)., respectivos.
Note que, os parágrafos obedecem uma ordem sequencial.

3 - PARÁGRAFOS:

3.1 - A codificação dos nomes-de-procedimentos é feita como os nomes-seções, omitindo-se a palavra SECTION.

3.2 - Os parágrafos devem ter no máximo 30 comandos.

EXEMPLO :

OBS: Qualquer comentário a critério do programador, deve ser feito através da cláusula " NOTE ".
Os parágrafos são codificados a partir da coluna 9.

4 - SENTENÇAS:

4.1 - Cada sentença é composta de um ou mais comandos, que pode ser separada por uma vírgula (,), mas devem ser separadas no mínimo por dois espaços. Uma sentença deve ter no máximo trinta comandos. A codificação é feita pelo método de escada, isto é, um comando para cada, sendo o próximo comando codificado pulando uma coluna (LINHA) à direita.

EXEMPLO:

```
NOI-ESCREVE-DETALHES.
    MOVE SPACE TO RATFID.
    ADD 1 TO PAGINA.
    MOVE PAGINA TO PAGINA-E.
    WRITE REGLATORIO FROM DET1 AFTER PAGE.
    MOVE ZERO TO AL.
NOR-FIM-DETALHE.
```

FIG 1

OBS: Este método é usado, devido a boa visiualização e de fácil acesso ao comando desejado.

5 - COMANDOS:.

5.1 - Os comandos estão divididos em três categorias:

5.1.1 - Comando de Decisão
5.1.2 - Comandos Aritmèticos
5.1.3 - Comandos Especiais para o compilador

5.1.1 - COMANDO IF.

5.1.1.1 - Teste de Classe
5.1.1.2 - Teste de Relação
5.1.1.3 - Teste de Condição
5.1.1.4 - Teste de Sinal
5.1.1.5 - Condições Compostas

5.1.1.1 - Codificação deve ser feita clara; usando os OPERADORES-DE-RELAÇÃO, e nunca os caracteres correspondentes.
Inicia na margem " B ", deixando dois espaços em branco entre o comando, nome-de-dado, operador-de-relação etc.

EXEMPLO:

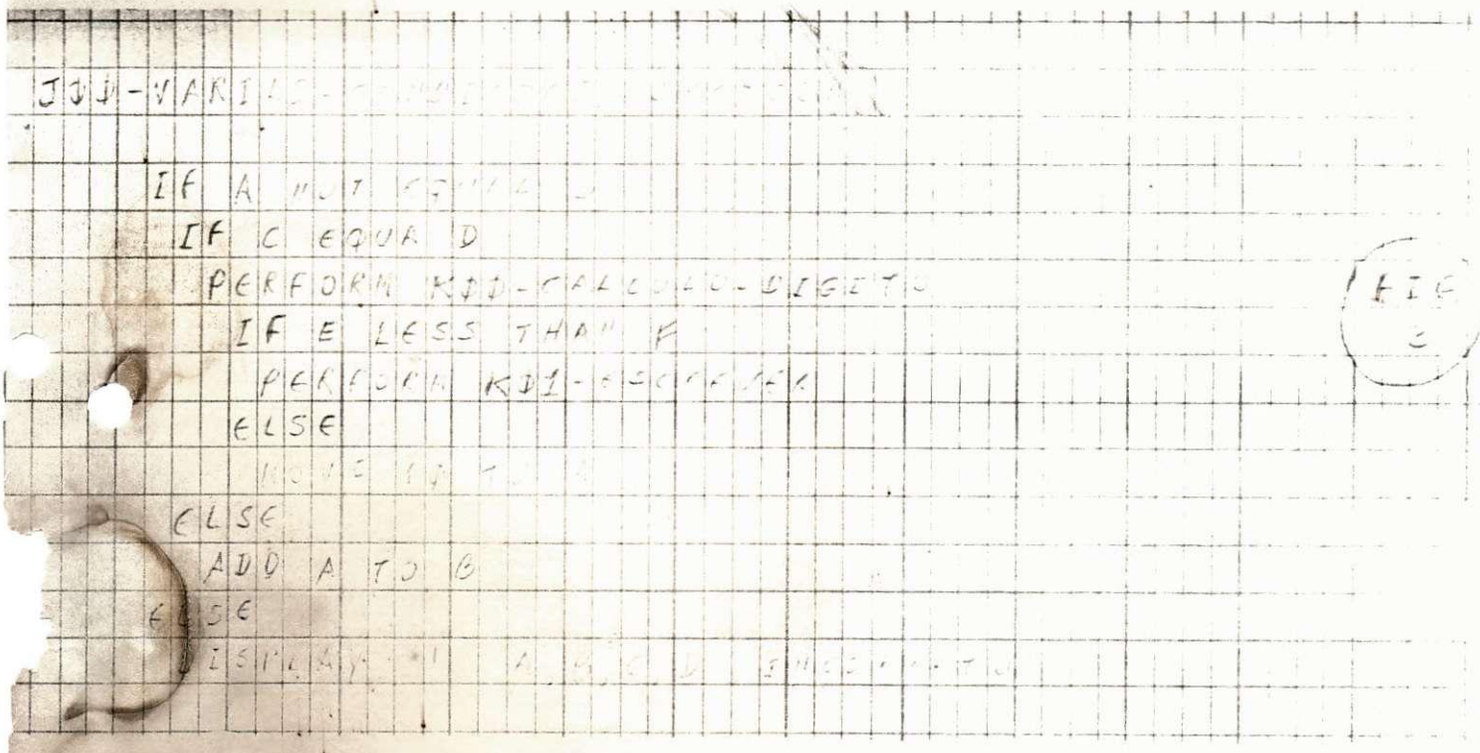
```
IF NOME-DE-DADO NOT NUMERIC
PERFORM 1302

1302.
```

5.1.1.2 - O teste de relação deve ser codificado igual ao item 5.1.1.1. Neste caso podemos ter em parte o IF aninhado. O IF aninhado, deverá ser codificado obedecendo os seguintes critè_

Continuação do ítem 5.

1º - Usar a cláusula ELSE ou OTHERWISE para relacionar com as próximas sentenças, em vez do delimitador.

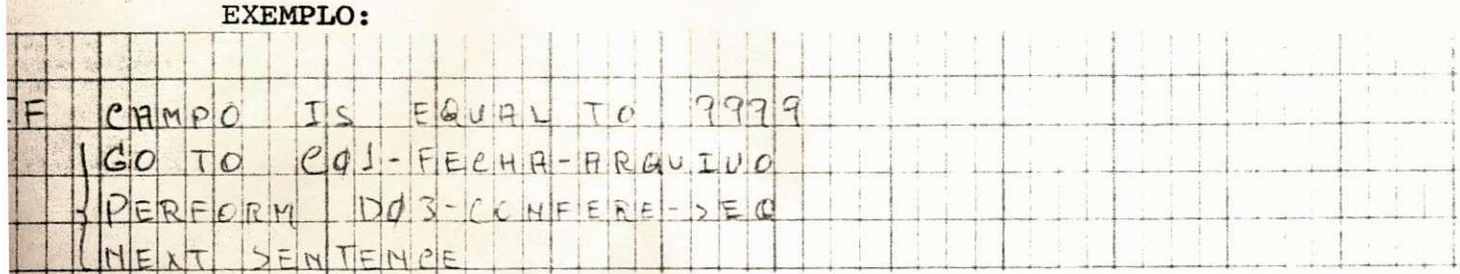
EXEMPLO:



OBS: Observe que o IF mais interno está relacionado ao ELSE mais interno, o mesmo acontecendo com os mais externos, ambos deverão ser codificados na mesma ' coluna.

2º - Quando um teste é provido de apenas um IF, codifica-se com um dos seguintes comandos de desvio:

EXEMPLO:



OBS: Neste tipo de teste é aconselhável usar como fim de condição o delimitador, omitindo o ELSE/OTHERWISE.

5.1.1.3 - TESTE DE CONDIÇÃO

Este tipo de teste está combinado com NOME-DE-CONDIÇÃO definido no nível 88.

EX: Supônhamos a seguinte determinação na WORKING-STORAGE ou LINKAGE SECTION.

```
03  GRAU-ESCOLARIDADE  PIC 9.

    88  PRIMARIO  VALUE 1.
    88  SECUNDARIO VALUE 2.
    88  SUPERIOR  VALUE 3.
```

TEMOS A SEGUINTE CODIFICAÇÃO:

```
    IF  PRIMARIO
        ADD 1 TO CONT-PRIMARIO
          PERFORM 100-FORA
    IF  SECUNDARIO.
        ADD 1 TO CONT-SECUNDARIO
          PERFORM 200-FORA.
          PERFORM M01 THRU M02 UNTIL SUPERIOR.
```

OBS:

A codificação permanece com as mesmas regras anteriores.

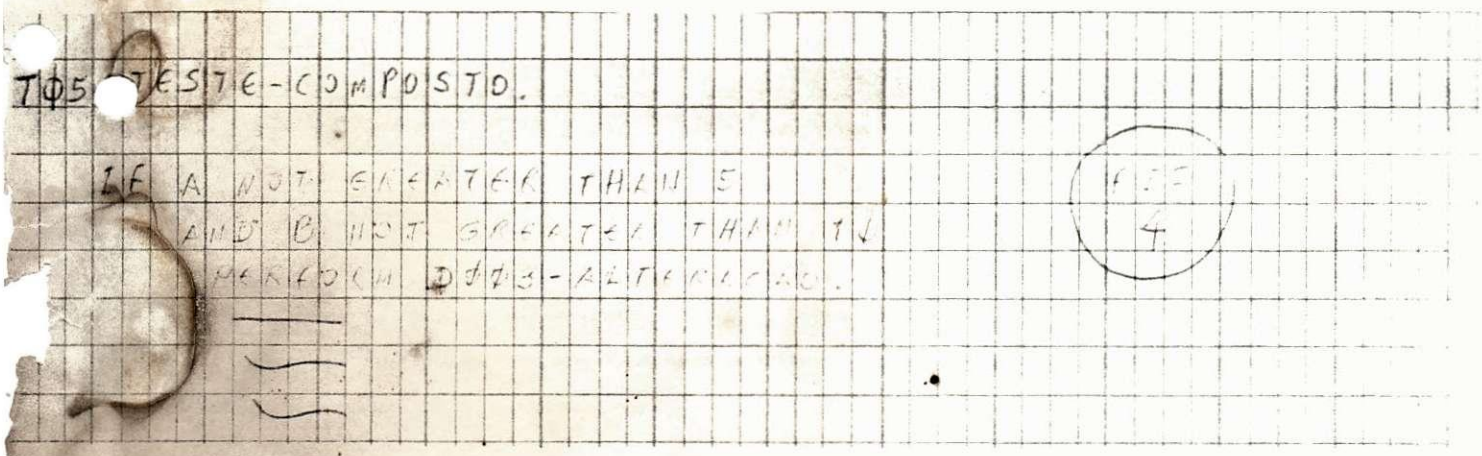
5.1.1.4- TESTE SINAL.

Permanece a mesma codificação do item 5.1.1.1 .

5.1.1.5- TESTE COMPOSTO.

Usar este tipo de teste quando ocorrer mais de uma condição. Neste caso, o programador **deve** limitar uma condição para cada linha, iniciando na margem "B", a primeira condição e às demais condições abaixo do primeiro nome-de-dado.

EXEMPLO:



5.1.2 - COMANDOS ARITMÉTICOS.

O comando " COMPUTE " substitui os demais comandos aritméticos, devido a rapidez de processamento tornando-se a interpretação e a codificação da operação aritmética direta.

EXEMPLO:

```
C00-CALCULOS SECTION.
C01-CALCULA-IDADE.
    COMPUTE IDADE = DATA-ATUAL - DATA-NASC.
C02-CALCULA-HORAS.
    COMPUTE HORAS = IDADE * 365
C03-VALOR-SALARIO.
    COMPUTE VAL-SAL = HORAS / 1...
C04-SAIDA-CALCULOS.
```

FIG. 5

- Para se fazer um incremento ou decremento, o programador deve usar a cláusula SET , para manusear tabelas.

EXEMPLO:

```
01  TABELA
    05  TAB-1 OCCURS 100 TIMES
        INDEXED BY IND.

PROCEDURE DIVISION.
    SET IND TO 1
    SET IND UP BY 1
```

- A codificação para estes comandos, obedecerá as seguintes regras:

1- Inicia na margem "B".

2- O máximo de três caracteres para cada indexador.

3- Usá-los sempre definidos na cláusula indexed by.

5.1.3- COMANDOS ESPECIAIS.

Dividimos os comandos especiais em duas classes, para uma melhor explanação:

5.1.3.1 - Comandos de abertura e fechamento.

EXEMPLO

FIG. 7

```
OPEN   INPUT   ARQ-1,
               ARQ-2,
       OUTPUT  ARQ-3,
               ARQ-4.
.
.
CLOSE  ARQ-1,
       ARQ-2.
```

Estes comandos têm uma codificação diferente das anteriores, devido a quantidade de nome-de-arquivo que aparecem para um só comando, "OPEN ou CLOSE". Usamos este método para dar melhor clareza ao programador.

REGRAS:

1- Os verbos iniciam-se na margem "B".

3- Se hover mais de um nome-de-arquivo,deve codificar um abaixo do outro, da esquerda para direita.
4- A cláusula OUTPUT, codifica-se abaixo da INPUT.
5- Os nomes-de-arquivos seguirão o mesmo critério do item 3.

5.1.4- COMANDOS DE ENTRADA E SAIDA.

Os comandos deverão ser codificados a partir da margem "B", usando como os outros o método de escada.

EXEMPLO:

```
A00-LEITURA-IMPRESSAO SECTION.
A01-LEITURA.
    READ CARTAO AT END
      MOVE 'X' TO FIM
      MOVE CORR REG01 TO SAIDA
      MOVE SPACE TO IMPRESSAO
      ADD 1 TO CONT-CARTAO.
P02-IMPRESSAO.
    ADD 1 TO LINHA.
      WRITE IMPRESSAO FROM SAIDA AFTER N.C.
A03-FIM
```

******
****
**
*

BIBLIOGRAFIA:

1- McCRACKEN,DANIEL D. - A GUIDETO COBOL PROGRAMING

2- GUSMAN/VASCONCELLOS: Fluxogramas e P. Cobol

3- H. CATUNDA - PROGRAMAÇÃO ESTRUTURADA

4- ALEX BATOS - PROGRAMAÇÃO COBOL