

PUC

Série: Monografias em Ciência da Computação
Nº 16/78

O SUPORTE BÁSICO DO SISTEMA HYADES DE GERÊNCIA
DE BANCO DE DADOS

- VOLUME 1 -

O PACOTE DE GERÊNCIA DE BANCO DE DADOS

PARTE 2

por

Sônia M. M. Passos
Roberto P. Vasques

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

Rua Marquês de São Vicente 225 — ZC 19

Rio de Janeiro — Brasil

Série: Monografias em Ciência da Computação
Nº 16/78

Editor da Série: Michael F. Challis

Agosto, 1978

O SUPORTE BÁSICO DO SISTEMA HYADES DE GERÊNCIA
DE BANCO DE DADOS *

- VOLUME 1 -

O PACOTE DE GERÊNCIA DE BANCO DE DADOS

PARTE 2

por

Sônia M. M. Passos

Roberto P. Vasques

* Trabalho patrocinado pelo CNPq TC2222.0060/77

Para obter cópias, dirija-se a:

Rosane T. L. Castilho
Chefe, Setor de Documentação e Informação
Deptº de Informática - PUC/RJ
Rua Marquês de São Vicente, 209 - Gávea
22.453 - Rio de Janeiro - RJ - Brasil.

SUMÁRIO

APRESENTAÇÃO..... iii

VOLUME 1 - O PACOTE DE GERÊNCIA DE BANCO DE DADOS

PARTE 1

1. INTRODUÇÃO.....	1
1.1. Apresentação do Problema.....	1
1.2. Objetivo do Trabalho.....	2
1.3. Desenvolvimento.....	3
1.4. Organização.....	3
2. MANUAL DO SISTEMA.....	5
2.1. Concepção do Sistema.....	5
2.2. Arquivos do Sistema.....	9
2.3. Procedimentos do Sistema.....	33
2.4. Fluxo do Sistema.....	41
2.5. Padronização dos Módulos.....	46
2.6. Especificação dos Primitivos e Subrotinas do Sistema.....	53

PARTE 2

3. MANUAL DO USUÁRIO.....	108
3.1. Funções do Sistema.....	108
3.2. Necessidades de Hardware e Software.....	111
3.3. Arquivos do Sistema.....	112
3.4. Fluxo do Sistema.....	120
3.5. Especificação dos Primitivos a Nível de Usuário.....	121
3.6. Códigos de Retorno dos Primitivos.....	148
3.7. Restrições de Implementação.....	149
4. MANUAL DE OPERAÇÃO.....	151
4.1. Execução do Sistema.....	151
4.2. Exemplo de Uso do Suporte.....	154
4.3. Manutenção do Suporte.....	157
REFERÊNCIAS BIBLIOGRÁFICAS.....	187

VOLUME 2 - O PACOTE DE AVALIAÇÃO DE EFICIÊNCIA

1. INTRODUÇÃO.....	1
1.1. Apresentação do Problema.....	1
1.2. Desenvolvimento do Trabalho.....	1
1.3. Objetivo do Trabalho.....	2
1.4. Organização.....	4
2. DESENVOLVIMENTO E IMPLANTAÇÃO DO SISTEMA DE BANCO DE DADOS.....	6
2.1. Considerações Gerais.....	6
2.2. Arquivos.....	6
2.3. Procedimentos.....	9
3. DESENVOLVIMENTO DO SISTEMA DE AVALIAÇÃO DE EFICIÊNCIA...14	
3.1. O que entendemos por Avaliação de Eficiência de Sistema de Banco de Dados.....	14
3.2. Concepção.....	16
3.3. Considerações Gerais.....	16
3.4. Parâmetros a Analisar.....	18
3.5. Arquivos.....	19
3.6. Programas.....	29
3.7. Subrotinas.....	41
4. MANUAL DE UTILIZAÇÃO DO SISTEMA DE AVALIAÇÃO DE EFICIÊNCIA.....	53
4.1. Programa de Distribuições.....	53
4.2. Programa de Frequência.....	61
4.3. Programa de Alocação de Área.....	64
5. ESTUDO DE UM CASO.....	67
5.1. Relatórios de Distribuições.....	67
5.2. Relatórios de Frequência.....	74
5.3. Relatórios de Alocação de Espaço.....	77
REFERÊNCIAS BIBLIOGRÁFICAS.....	82

O SUPORTE BÁSICO DO SISTEMA HYADES DE GERÊNCIA
DE BANCO DE DADOS

- VOLUME 1 -

O PACOTE DE GERÊNCIA DE BANCO DE DADOS

PARTE 2

Este volume foi elaborado, como tese de
Mestrado, por Roberto Pires Vasques

CAPÍTULO 3 - MANUAL DO USUÁRIO

3.1 - Funções do Sistema

O suporte foi desenvolvido como implementação inicial em um projeto envolvendo Banco de Dados. Inclui os mecanismos básicos para uma implementação seja qual for o modelo adotado: hierárquico, de redes ou relacional. Manuseia três estruturas de arquivo fundamentais: arquivo de dados, elos, inversões além de dicionário de dados, denominado arquivo de estrutura. Os procedimentos que operam sobre estas estruturas são chamados primitivos. O sistema possui 24 primitivos que conforme a Figura 1 manuseiam:

- Arquivos do usuário, considerados no sistema como segmentos;
- Registros destes segmentos; e
- Arquivos do sistema.

Utilizadas pelos primitivos temos 8 subrotinas sendo:

- 3 subrotinas do sistema, e
- 5 subrotinas do sistema de avaliação de eficiência.

As principais funções do sistema especificadas por primitivos são:

PRIMITIVO	FUNÇÃO
CRIDAD	- cria segmento de dados nos arquivos de dados do sistema, permanente ou temporário, a partir de um arquivo sequencial do usuário;
CRIELO	- cria um segmento de elos em um dos arquivos de elos do sistema permanente ou temporário a partir de dois segmentos de dados;
CRIINV	- cria um segmento de inversões baseado em valores de um conjunto de cam

pos de registro de um segmento de dados ordenado;

- CRIDUP - cria cópia de um segmento de dados, elos ou inversões. Permite criar cópia de um segmento temporário sob a forma de um segmento permanente;
- CRICLS - cria cópia de segmento de dados clas_sificado;
- CRIPRJ - cria segmento de dados com base em segmento de inversão. Cria uma projeção;
- CANDAD - cancela um segmento de dados, retirando-o do sistema;
- CANELO - cancela um segmento de elos, retirando-o do sistema;
- CANINV - cancela um segmento de inversões, retirando-o do sistema;
- INFARQ - fornece informações em tempo de execução sobre determinado segmento do sistema;
- TRIARQ - troca identificador de segmento de dados para o usuário;
- INSDAD - insere registro de dados em um segmento de dados permanente ou temporário;
- INSELO - insere registro de elo em um segmento de elos permanente ou temporário;

- REMDAD - remove registro de dados de um segmento de dados;
- REMELO - remove registro de elos de um segmento de elos;
- SELDAD - seleciona de acordo com o tipo de seleção um registro de dados de um segmento de dados;
- SELELO - seleciona de acordo com o tipo de seleção, um registro de elos de um segmento de elos;
- SELINV - seleciona de acordo com o tipo de seleção, um registro de inversões de um segmento de inversões;
- MODDAD - altera valor de item em registro de segmento de dados;
- ATIVDB - torna disponível a utilização do Banco de Dados;
- DATVDB - desativa o suporte, guardando informações necessárias para nova utilização;
- RESMDB - emite relatório resumo sobre o Banco de Dados;
- COPIDE - rearrumação e cópia do Banco de Dados;
- RESTDB - restaura Banco de Dados;

As subrotinas do sistema são:

- AUX1 - verifica a situação de um segmento no Banco de Dados;
- AUX2 - atualiza situação de um segmento no Banco de Dados;
- AUX3 - abre os arquivos do Banco de Dados.

As subrotinas do sistema de avaliação de eficiência acopladas no sistema são:

- SUB1 - aloca tabelas de frequência de uso dos primitivos e manuseio de arquivos;
- SUB2 - armazena hora de início;
- SUB3 - adiciona à tabela, tempo de utilização de primitivo;
- SUB4 - atualiza tabela de uso dos arquivos;
- SUB5 - armazena em arquivo as tabelas de uso dos primitivos e de manuseio de arquivos.

3,2 - Necessidades de Hardware e Software

O suporte foi desenvolvido para uso em equipamento IBM/370 modelo 165 da PUC. Pode, porém, ser executado em outros modelos que disponham de:

- memória suficiente real ou virtual;
- dispositivo de acesso direto compatível.

A memória utilizada pelo suporte depende dos primitivos e

subrotinas ativas durante o processamento.

Por restrição de tamanho de memória, pode o suporte ter seus primitivos executados sob a forma de overlay, porém com perda substancial de tempo de processamento.

Os arquivos do suporte podem ser armazenados em disco IBM2314, IBM3330 ou outros compatíveis com a organização utilizada.

O suporte é operável no sistema operacional OS; podendo ser: OS/MVT, que é o utilizado na PUC; OS/VS2, ou outros. O suporte foi desenvolvido em PL1/F, por restrição ao compilador disponível na PUC. Utiliza em seus arquivos a organização sequencial indexada.

É utilizada pelo suporte, a opção de 'JCL' que permite o ingresso de novos registros ao final de um arquivo já criado sem a necessidade de recriá-lo.

O sistema operacional da instalação deverá dispor do programa utilitário SORT compatível com o compilador PL1/F.

3. 3 - Arquivos do sistema

Em seus primitivos, o sistema manuseia 20 definições de dados (DDNAME), além das definições de dados dos usuários e das áreas de trabalho do programa utilitário SORT. Porém o sistema atua sobre 10 a 12 conjuntos de dados (DATA-SET'S), além das áreas de trabalho do SORT e dos conjuntos de dados dos usuários; sendo que 4 conjuntos de dados representam a base do sistema:

- dados permanente;
- elos permanente;
- inversões permanente;
- estruturas permanente.

As figuras 5, 6 e 7 indicam graficamente a formação dos arquivos de dados, elos e inversões.

A relação entre os arquivos definidos nos módulos (FILENAME), as definições de dados (DDNAME) e os respectivos conjunto de dados (DSNAME) está indicada no Quadro 7.

O nome dos arquivos e das definições de dados são padrões, sendo indicados internamente nos primitivos e subrotinas do sistema. O nome dos conjuntos de dados é livre, devendo, porém, obedecer as especificações da instalação onde o sistema é processado.

As definições de dados devem estar sempre presentes mesmo que os conjuntos de dados não existam, quando então deve-se usar a opção "DUMMY". Este pode ser o caso das definições de dados ARQSAEI e ARQSAEO.

No arquivo relativo a posição do Banco de Dados não se faz obrigatório o uso de um conjunto de dados próprio, pois os dados podem ser armazenados em uma classe de saída de OS.

Os arquivos de trabalho são necessários para o programa utilitário SORT, ativado pelos primitivos que necessitam de dados ordenados. O número de definições de dados para áreas de trabalho é dependente do tipo de opção de classificação utilizada e do número de registros a classificar.

Os arquivos dos usuários devem conter como definição de dados o mesmo nome informado como parâmetro no primitivo CRIDAD. Podem ter ou não identificação do conjunto de dados.

O Quadro 8 indica a existência das definições de dados nos primitivos e subrotinas do sistema.

Em qualquer condição de uso do sistema, os conjuntos de dados conhecidos como dados permanente, elos permanente, inversões permanente e estruturas permanentes são sempre manuseados.

A utilização de uma definição de dados permanente ou temporária só é definida em tempo de execução. Por este motivo os primitivos de manuseio de segmentos e de registros tem assinados no Quadro 8 o manuseio dos arquivos permanente e temporário de uma mesma natureza. Na realidade, dependendo da opção solicitada em parâmetro, podem ser manuseadas as definições de dados permanente e/ou temporária de uma mesma natureza.

FIGURA 5 - ARQUIVO DE DADOS

FIGURA 5 - ARQUIVO DE DADOS

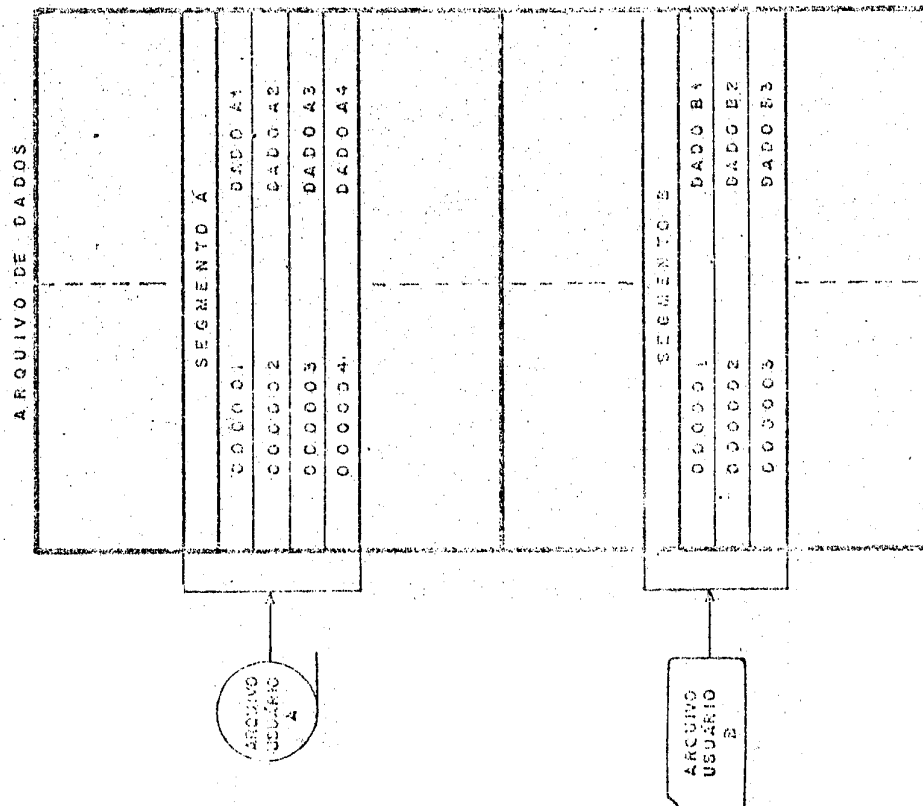


FIGURA 6 - ARQUIVO DE ELOS

FIGURA 6 - COMPOSIÇÃO ARQUIVO DE ELOS

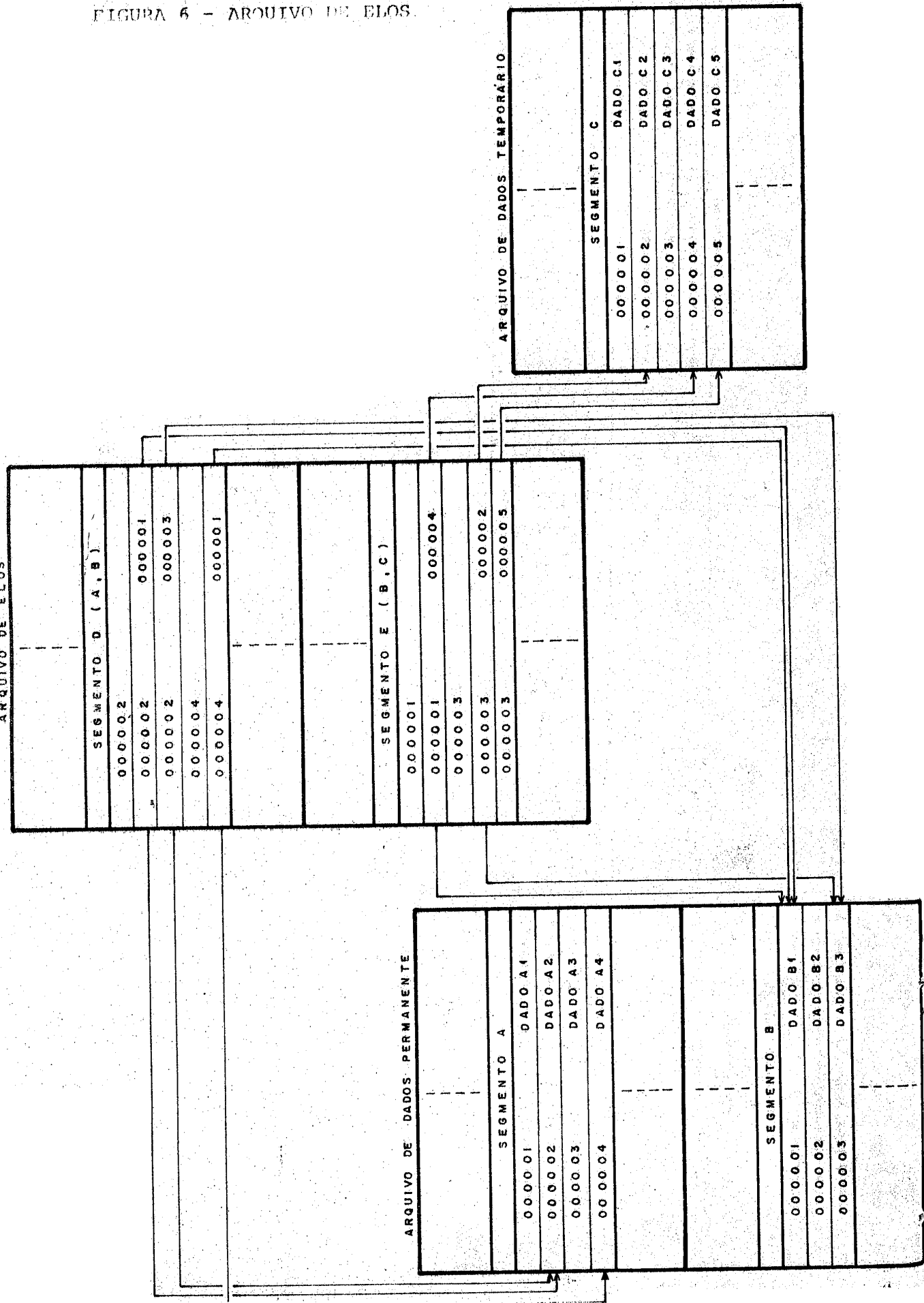
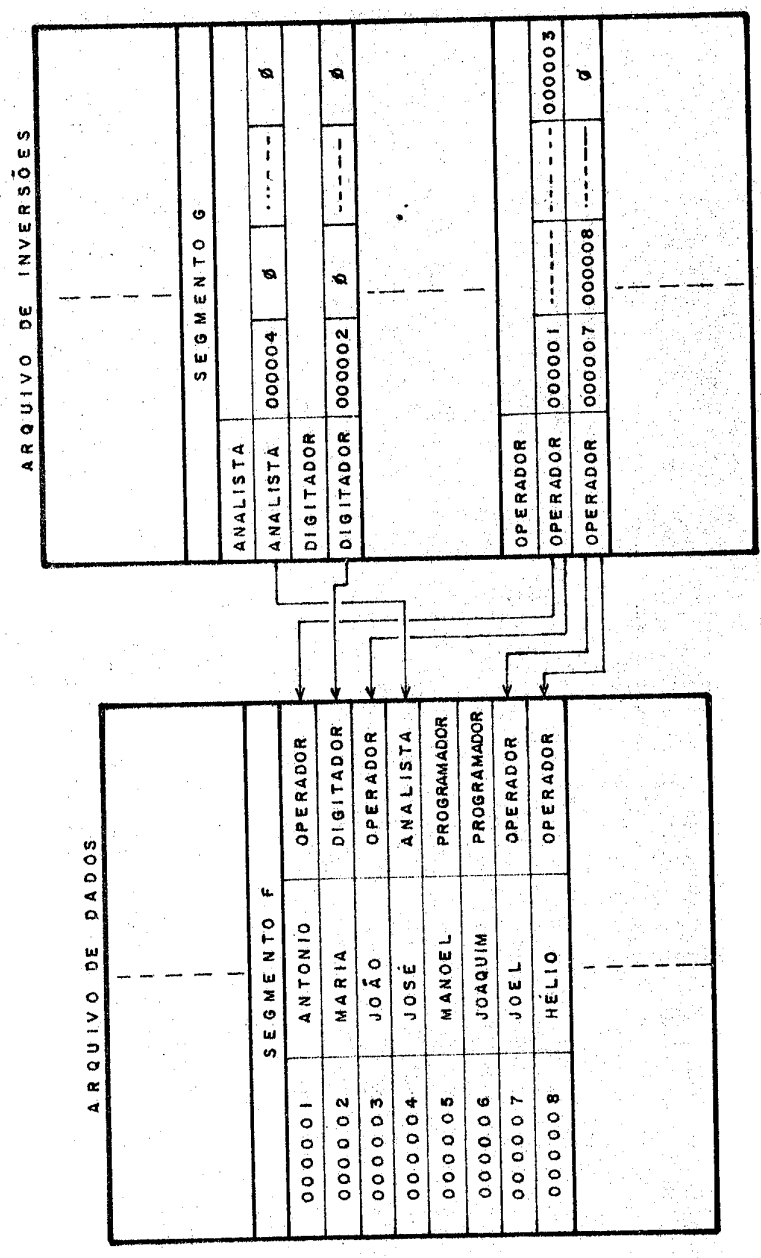


FIGURA 7 - ARQUIVO DE INVERSÕES

FIGURA 7 - ARQUIVO DE INVERSÕES



QUADRO 7 - Relação entre arquivos, Definições de Dados e Conjuntos de dados do Sistema.

IDENTIFICAÇÃO	ARQUIVO (FILENAME)	DEFINIÇÃO DE DADOS (DDNAME)	CONJUNTO DE DADOS (DSNAME)
Dados permanente	ARQD	ARQDP	ARQDP
Dados permanente	ARQDI	ARQDIP	ARQDP
Dados temporários	ARQD	ARQDT	ARQDT
Dados temporários	ARQDI	ARQDIT	ARQDT
Elos permanente	ARQE	ARQEP	ARQEP
Elos permanente	ARQEI	ARQEIP	ARQEP
Elos temporários	ARQE	ARQET	ARQET
Elos temporários	ARQEI	ARQEIT	ARQET
Inversões permanente	ARQI	ARQIP	ARQIP
Inversões permanente	ARQII	ARQIIP	ARQIP
Inversões temporários	ARQI	ARQIT	ARQIT
Inversões temporários	ARQII	ARQIIT	ARQIT
Estruturas permanen tes	ARQS	ARQSP	ARQSP
Estruturas permanen tes	ARQSI	ARQSIP	ARQSP
Estruturas temporá rios	ARQS	ARQST	ARQST
Estruturas temporá rios	AQOSI	ARQSIT	ARQST
Avaliação Eficiên - cia Entrada	ARQSAE2	ARQSAEI	ARQSAE2ANT
Avaliação Eficiên - cia Saída	ARQSAE2	ARQSAEO	ARQSAE2AT
Posição Sistema	SYSPRINT	SYSPRINT	-
Imagem Banco de Da dos	ARQB	ARQB	ARQB
Arquivo Usuários	ARQUS	*	-
Arquivos trabalho	SORTWKnn	SORTWKnn	-

* DDNAME indicado em tempo de execução.

QUADRO 8 - Uso das definições de dados (DDNAME) nos primitivos e subrotinas do sistema

DDNAMES	ARQDP	ARQDIP	ARQDT	ARQDIT	ARQEP	ARQEIP	ARQET	ARQEIT	ARQIP	ARQIIP	ARQIT	ARQIIT	AROSP	AROSIP	AROST	AROSIT	AROSAI	AROSAO	SYSPRINT	AROB	ARQDO	ARQDO	USRARIO	ARQDO	SORTWKIN
PRIMITIVOS																									
CRIDAD	X		X										X		X								X		
CRIELO	X		X		X		X																		
CRINV	X		X						X		X	X		X										X	
CRIDUP	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X									
CRICLS	X		X										X	X	X	X									X
CRIPRJ	X		X						X		X		X	X	X	X									
CANDAD	X		X		X		X		X		X		X		X										
CANELO	X		X		X		X																		
CANINV	X		X						X		X														
INFARO	X		X		X		X		X		X		X		X										
TRIARO	X		X		X		X		X		X														
INSDAD	X		X						X		X		X		X										
INSELO	X		X		X		X																		
REMDAD	X		X		X	X	X	X	X		X		X		X										
REMELO					X	X	X	X																	
SELDAD	X		X																						
SELELO					X		X																		
SELINV										X		X													
MODDAD	X		X						X		X		X		X										
ATIVDB	X				X				X				X												
DSTVDB	X				X				X				X												
RESMDB	X				X				X				X							X					
COPIDB	X				X				X				X										X		
RESTDB	X				X				X				X										X		
Subrotinas																									
AUX1																									
AUX2																									
AUX3	X		X		X		X		X		X		X		X										
SUB1																									X
SUB2																									
SUB3																									
SUB4																									
SUB5																		X							

* DDNAME só indicado em tempo de execução.

3.4 - FLUXO DO SISTEMA

O suporte foi desenvolvido de maneira que o Banco de Dados possa ser manuseado em uma só execução e em mais de uma execução, inclusive com a ocorrência de eliminação e restauração do Banco de Dados entre estas execuções.

A Figura 8 indica o Fluxo Operacional do sistema em todas as suas fases. Notamos que a primeira etapa do procedimento a ser realizada para o uso do suporte é a alocação dos conjuntos de dados permanentes do suporte, referentes a dados, elos, inversões e estruturas. Este procedimento faz-se necessário tendo em vista a opção de "JCL" "MOD" utilizada nos conjuntos de dados do suporte. Esta etapa pode ser realizada através da utilização do programa utilitário da IBM, IEFBR14. Para a utilização desta etapa faz-se necessário que os conjuntos de dados que constituem a base do sistema não estejam disponíveis. Esta etapa só deve ser utilizada antes da primeira execução do suporte e antes da restauração do Banco de Dados.

A etapa seguinte que aloca conjuntos de dados temporários destina-se, a tornar disponível ao suporte estes conjuntos de dados. É utilizada sempre antes da execução ou reexecução do suporte.

Os primitivos de apoio COPIDB e RESTDB executam suas funções independentes dos demais primitivos. O primitivo COPIDB além de rearrumar o Banco de Dados obtém um conjunto de dados sequencial, imagem do Banco de Dados. O primitivo RESTDB, após as alocações necessárias restaura o Banco de Dados rearrumado.

Os primitivos de apoio ATIVDB e DATVDB sempre iniciam e terminam uma execução do suporte. Entre a execução destes dois primitivos podem ser executados os demais primitivos do suporte que manuseiam com segmentos e registros. Porém estes primitivos possuem relações de dependência entre si para tarefas determinadas. A Figura 2 indica esta dependência através da precedência de uso dos primitivos. Esta dependência é em re

lação a tarefas interrelacionadas ou seja, só posso criar um segmento de inversão se o segmento de dados base desta inversão já tenha sido criado anteriormente na mesma execução ou em execuções anteriores do suporte. Podemos, entretanto, em uma mesma execução criar um segmento de inversões com base em um segmento de dados previamente criado e logo após criar um segmento de dados. O número de execuções do suporte é livre assim como o número de utilização de seus primitivos em uma execução do suporte.

3.5 - Especificação dos primitivos a nível de usuário:

São indicados graficamente os primitivos e suas relações com os arquivos envolvidos e subrotinas ativadas.

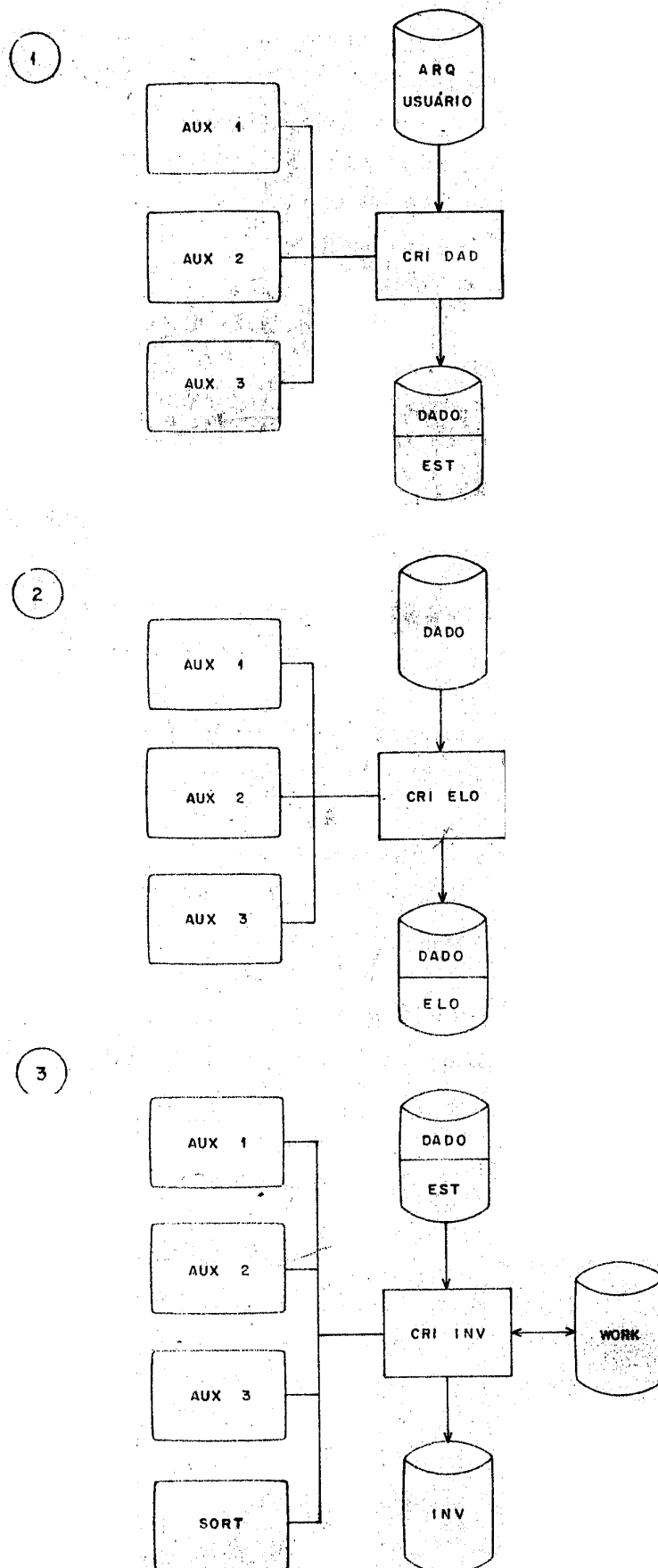
Na segunda parte são descritos os parâmetros dos primitivos, sendo utilizada a seguinte representação dos parâmetros;

Ei - Iésimo parâmetro de entrada;

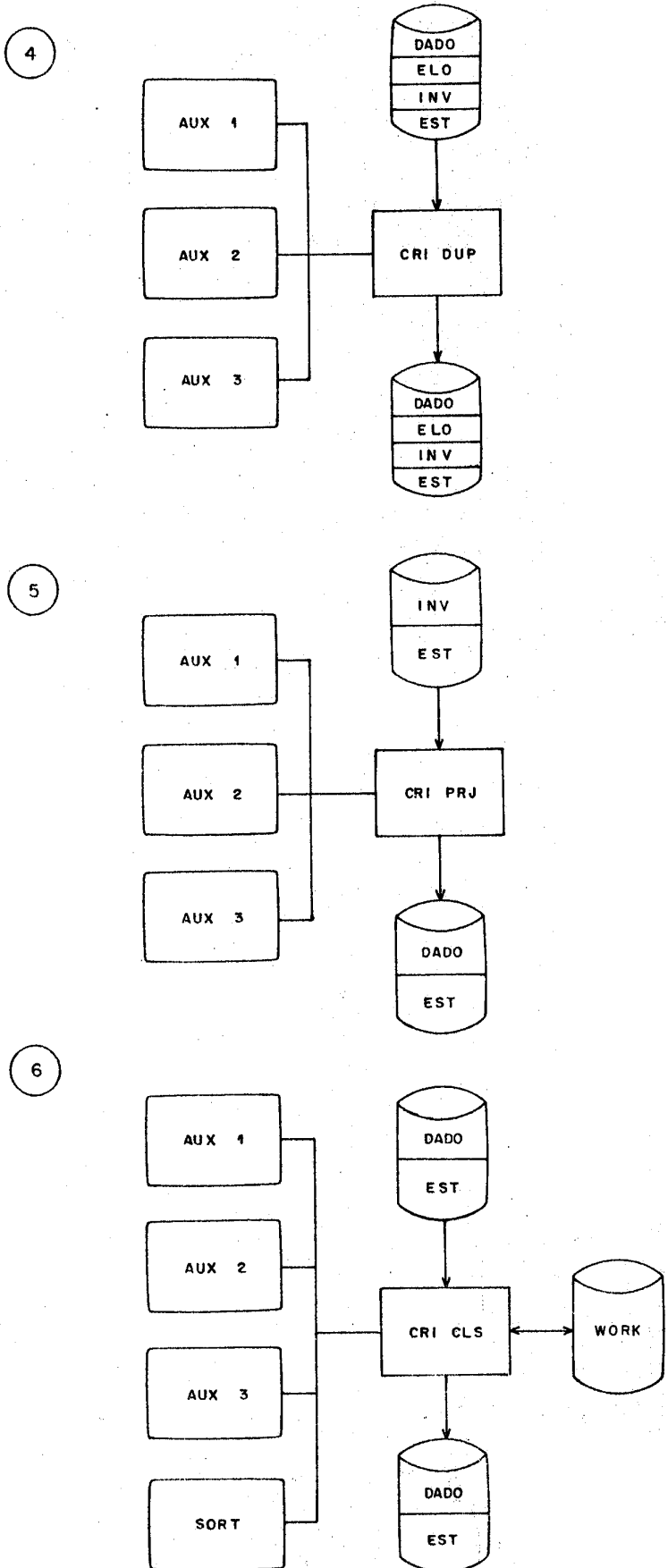
Si - Iésimo parâmetro de saída;

Di - Iésimo parâmetro que pode ser de entrada ou de saída.

PRIMITIVOS PARA MANUSEIO DE ARQUIVOS

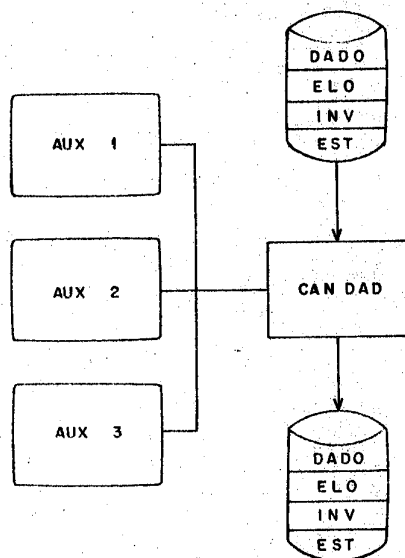


PRIMITIVOS PARA MANUSEIO DE ARQUIVOS

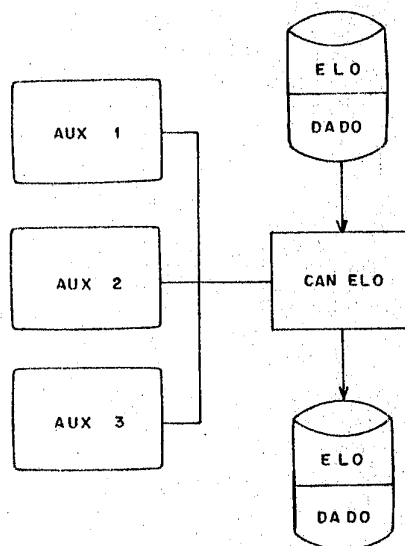


PRIMITIVOS PARA MANUSEIO DE ARQUIVOS

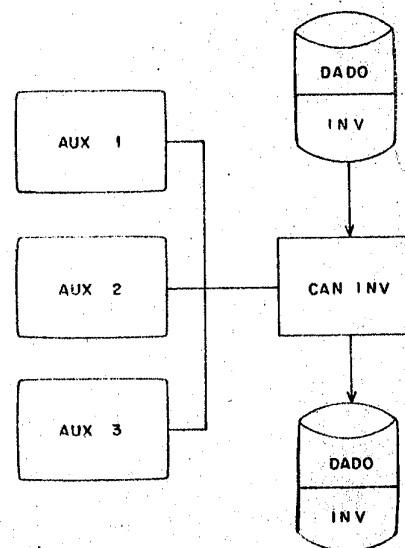
7



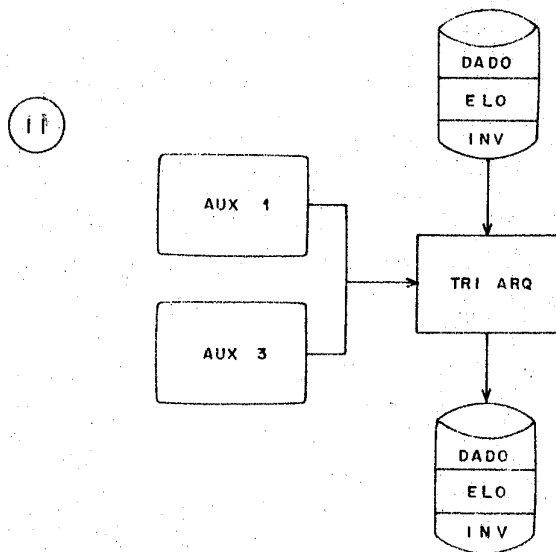
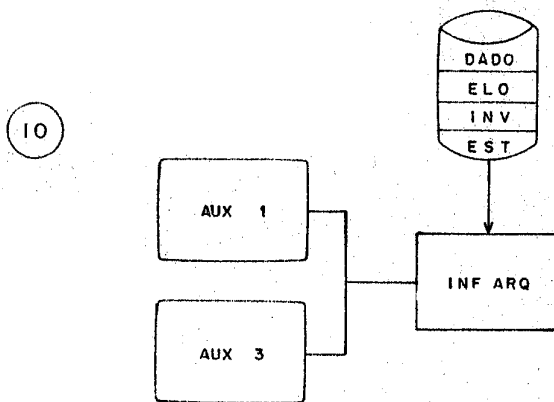
8



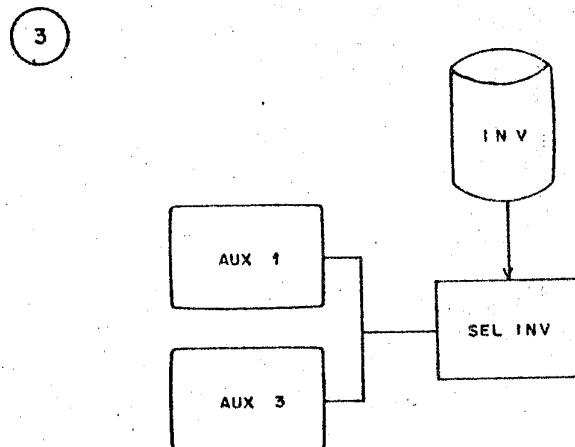
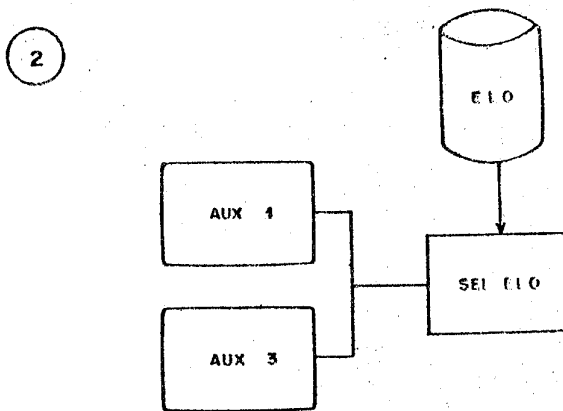
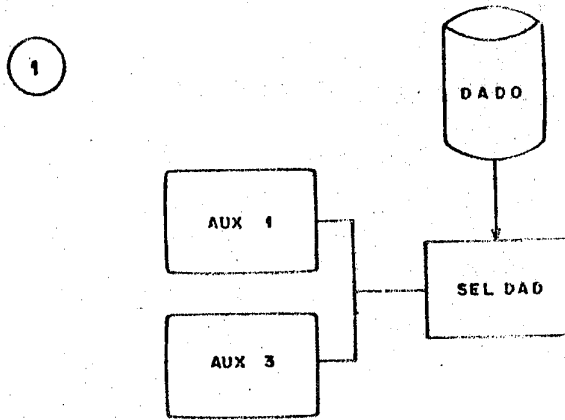
9



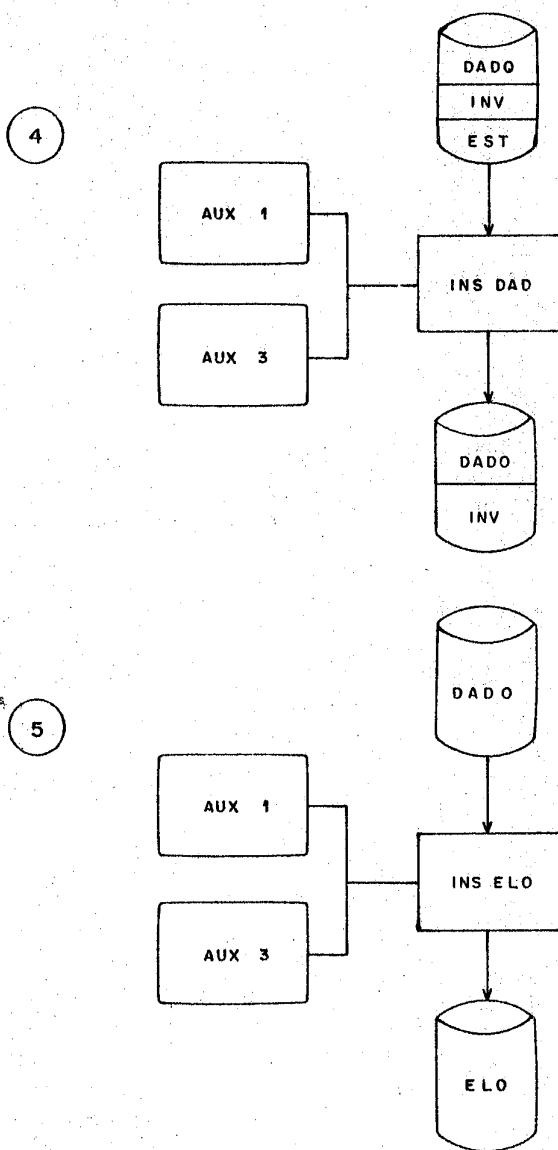
PRIMITIVOS PARA MANUSEIO DE ARQUIVOS



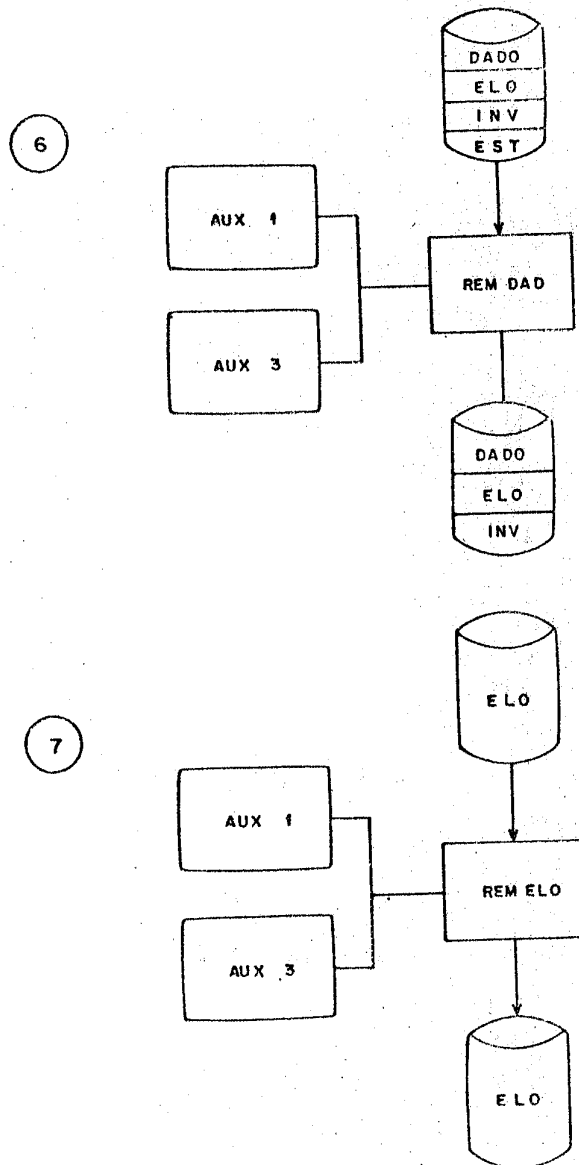
PRIMITIVOS PARA MANUSEIO DE REGISTROS



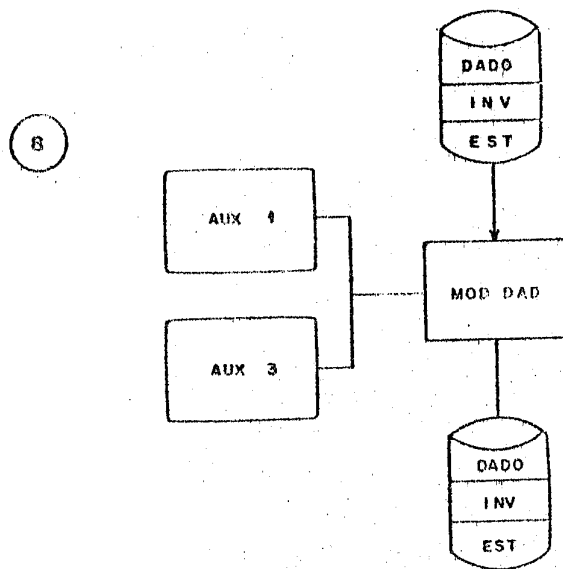
PRIMITIVOS PARA MANUSEIO DE REGISTROS



PRIMITIVOS PARA MANUSEIO DE REGISTROS

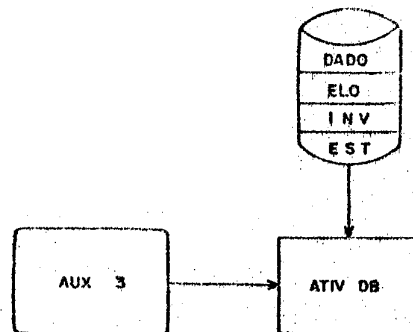


PRIMITIVOS PARA MANUSEIO DE REGISTROS

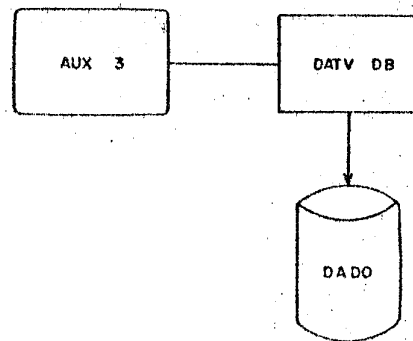


PRIMITIVOS DE APOIO

1

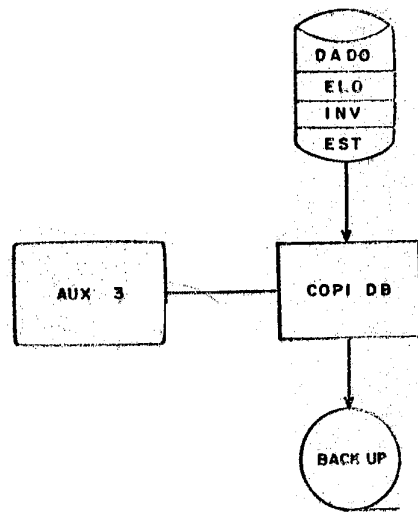


2

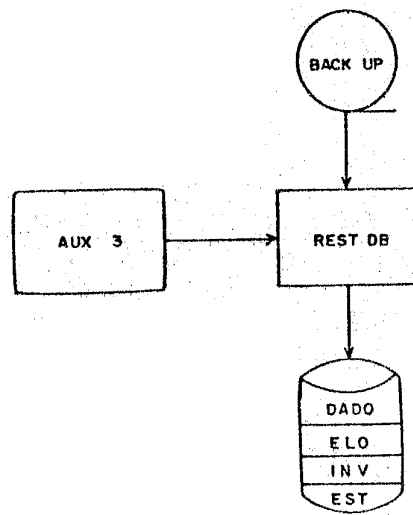


PRIMITIVOS DE APOIO

3

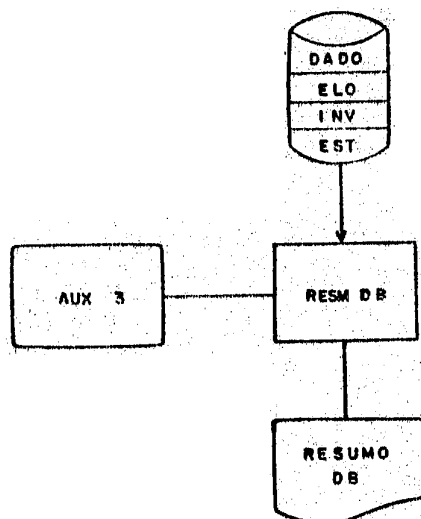


4



PRIMITIVOS DE APOIO

5



DESCRICÃO DOS PARÂMETROS DOS PRIMITIVOS DO SISTEMA:

- Cria arquivo (segmento) de dados:

CRIDAD (E1,E2,E3,E4,E5)

- E1 - identificação do arquivo de dados a ser criado no Banco de Dados. Alfanumérico com até 7 posições;
- E2 - Indica o tipo do arquivo a ser criado. Pode ser 'P' para arquivo permanente ou 'T' para arquivo temporário. Alfanumérico com 1 posição.
- E3 - Identificação do arquivo do usuário com base no qual será gravado o arquivo de dados. Deve ser idêntico ao nome da definição de dados onde estão disponíveis os dados. Alfanumérico com até 7 posições;
- E4 - Número previsto de registros com que o arquivo deve ser criado. Deve ser igual ou maior que o número de registros do arquivo do usuário. Se maior o sistema colocará registros DUMMY's ao final do arquivo guardando espaço para futuras inclusões sem utilização da área de "overflow". Numérico zonado com até 6 posições.

- E5 - Número de campos de registros que serão indi
cados no parâmetro seguinte. Numérico zonado
com até 2 posições.

- E6 - Descrição dos campos dos registros. Tabela com
cada elemento contendo:

- Item do campo, 1 posição numérica;
- identificação do campo, alfanumérico com 30
posições;
- posição início do campo, numérico zonado com
3 posições;
- posição de fim do campo, numérico com 3 posi
ções;
- tipo do campo, alfanumérico com 2 posições.
Pode ser:

CH - alfanumérico

ZD - numérico zonado

FI - binario fixo

PD - decimal fixo

FL - ponto flutuante

- Cria arquivo (segmento) de elos,

CRIELO (E1,E2,E3,E4,E5)

- E1 - Identificação para o sistema do arquivo de elos a ser criado. Alfanumérico com até 7 posições;
- E2 - Tipo de arquivo (segmento) a ser criado. Pode ser 'P' para arquivo permanente ou 'T' para arquivo temporário. Alfanumérico com 1 posição;
- E3 - Número previsto de registros considerados "DUMMYS" para o sistema que reservam lugar para futuras indicações de elos. Numérico zonado com até 6 posições;
- E4 - Identificação do arquivo de dados envolvido no elo como principal. Alfanumérico com até 7 posições;
- E5 - Identificação do arquivo de dados envolvido no elo como arquivo secundário. Alfanumérico com até 7 posições.

- Cria arquivo (segmento) de inversões:

CRIINV (E1,E2,E3,E4,E5)

- E1 - Identificação do arquivo de inversões a ser criado. Alfanumérico com até 7 posições.
- E2 - Tipo do arquivo a ser criado. Pode ser 'P' para arquivo permanente ou 'T' para arquivo temporário. Alfanumérico com 1 posição;
- E3 - Identificação do arquivo de dados com base no qual será criada a inversão. Alfanumérico com até 7 posições;

- E4 - Número de domínios da inversão que são indicados no parâmetro seguinte. Numérico zonado, com até 2 posições;
- E5 - Domínios da inversão. Indica os índices dos campos do arquivo de estrutura correspondente ao arquivo de dados a partir dos quais será formatada a chave do arquivo de inversão. Tabela numérica, com até 20 elementos com 2 posições numéricas zonadas cada.

- Duplica um arquivo de dados, elos ou inversões.

CRIDUP (E1,E2,E3,E4)

- E1 - Identificação do arquivo a ser criado. Alfanumérico com até 7 posições;
- E2 - Tipo de arquivo a ser criado. Pode ser 'P' para arquivo permanente ou 'T' para arquivo temporário. Alfanumérico com 1 posição;
- E3 - Identificação do arquivo a ser duplicado. Pode ser arquivo de dados, de elos ou de inversões. Alfanumérico com até 7 posições.
- E4 - Identifica o tipo de cópia. Numérico zonado com 1 posição, podendo ser:
 - 0 - cópia do arquivo integralmente, inclusive com os registros considerados DUMMY's para o usuário.
 - 1 - cópia dos registros ativos do arquivo.

- Cria arquivo (segmento) de dados classificado.

CRICLS (E1,E2,E3,E4,E5)

E1 - Identificação do arquivo de dados a ser criado. Alfanumérico com até 7 posições;

E2 - tipo do arquivo a ser criado. Pode ser 'P' para arquivo permanente ou 'T' para arquivo temporário. Alfanumérico com 1 posição.

E3 - Identificação do arquivo de dados a ser classificado. Alfanumérico com até 7 posições;

E4 - numero de dominios, para a classificação, que são indicados no parâmetro seguinte. Numérico zonado com até 2 posições;

E5 - indica os indices dos campos do arquivo de estrutura correspondente ao arquivo de dados origem, a partir dos quais será formatada a chave para classificação. Tabela numérica com até 20 elementos com 2 posições numéricas zonadas cada.

- Cria arquivo (segmento) de dados representando uma projeção.

CRIPRJ (E1,E2,E3)

E1 - Identificação do arquivo de dados a ser criado. Alfanumérico com até 7 posições;

E2 - Tipo do arquivo a ser criado. Pode ser 'P' para arquivo permanente ou 'T' para arquivo temporário. Alfanumérico com 1 posição.

E3 - Identificação do arquivo de inversões que servirá de base para o arquivo a ser criado. Alfanumérico, com até 7 posições.

- Cancela arquivo (segmento) de dados

CANDAD (E1)

E1 - Identificação do arquivo de dados a ser cancelado .
Alfanumérico com até 7 posições.

- Cancela arquivo (segmento) de elos

CANELO (E1)

E1 - Identificação do arquivo de elos a ser cancelado.
Alfanumérico com até 7 posições.

- Cancela arquivo (segmento) de inversões.

CANINV (E1)

E1 - Identificação do arquivo de inversões a ser cancelado.
Alfanumérico com até 7 posições.

- Informações sobre um arquivo (segmento)

INFARQ (E1,E2,S1, S2)

E1 - Identificação do arquivo sobre o qual se deseja informações.
Alfanumérico com até 7 posições;

E2 - Identificação do campo do registro sobre o qual se deseja informações. Alfanumérico com 2 posições.
Para arquivo de estruturas, indica o campo sobre o qual se deseja informações. Para arquivos de dados, elos ou inversões, deve ser codificado com brancos.

S1 - Informa a natureza e o tipo do arquivo cujo parâmetro seguinte conterà as informações. Alfanumérico, com 2 posições contendo:

'DP'- arquivo de dados permanente;

'DT'- arquivo de dados temporário;

- 'EP' - arquivo de elos permanente;
- 'ET' - arquivo de elos temporários;
- 'IP' - arquivo de inversões permanente;
- 'IT' - arquivo de inversões temporário;
- 'SP' - arquivo de estruturas permanente;
- 'ST' - arquivo de estruturas temporário.

S2 - Fornece informações sobre o arquivo ou sobre o campo solicitado. Se o arquivo for de dados as informações retornadas são:

- tamanho dos registros, numérico, compactado com 3 posições;
- dimensão do segmento, numérico compactado com 6 posições;
- número inicial de registros, numérico compactado, com 6 posições;
- número de inserções de registros, numérico compactado com 6 posições;
- número de cancelamentos de registros, numérico compactado com 6 posições;
- número de domínios, numérico compactado com 3 posições;

Se o arquivo for de elos as informações retornadas são:

- identificação do arquivo de dados principal, alfanumérico com 7 posições;
- identificação do arquivo de dados secundário, alfanumérico com 7 posições;
- dimensão do arquivo, decimal compactado com 6 posições;
- número de registros inseridos, decimal compactado com 6 posições;
- número de registros removidos, decimal compactado com 6 posições;

Se o arquivo for de inversões as informações retornadas são:

- identificação do arquivo de dados base da inversão, alfanumérico com 7 posições;

- número inicial de registros, decimal compactado com 6 posições;
- número de registros inseridos, numérico compactado com 6 posições;
- número de registros removidos, numérico compactado com 6 posições;
- número de domínios de inversões, numérico compactado com 2 posições;
- domínio da inversão, tabela com 20 elementos, cada elemento, numérico compactado com 2 posições;

Se for relativa a campos de registro, as informações retornadas são:

- nível do campo, numérico compactado com 1 posição;
- identificação do campo, alfanumérico com 30 posições;
- posição inicial do campo, numérico compactado com 3 posições;
- posição final do campo, numérico compactado com 3 posições;
- tipo de campo, alfanumérico com 2 posições.

Estas informações são retornadas em forma de um campo alfanumérico de tamanho variável. O usuário deve preparar as estruturas correspondentes para o manuseio das informações.

- Troca de nome de arquivo (segmento)

TRIARO (E1, E2)

E1 - Identificação do arquivo a ser alterado; alfanumérico com até 7 posições;

E2 - nova identificação do arquivo, alfanumérico com até 7 posições.

- Insere registro em arquivo (segmento) de dados

INSDAD (E1,E2,S1)

E1 - Identificação do arquivo de dados no qual o registro deve ser incluído, alfanumérico com até 7 posições;

E2 - registro a ser inserido, devendo estar com os dados no formato e tipo do arquivo correspondente, alfanumérico com até 502 posições;

S1 - identificador do registro inserido no arquivo de dados, numérico zonado com 6 posições.

- Insere registro em arquivo (segmentos) de elos

INSELO (E1,E2,E3,E4,E5,)

E1 - Identifica o arquivo no qual o elo deve ser inserido, alfanumérico com até 7 posições;

E2 - identificação do arquivo de dados principal que compõe o elo, alfanumérico com até 7 posições;

E3 - identificador do registro no arquivo de dados principal, numérico zonado com 6 posições;

E4 - identificação do arquivo de dados secundário que compõe o elo, alfanumérico com até 7 posições;

E5 - identificador do registro no arquivo de dados secundário, numérico zonado com 6 posições:

- Remove registro do arquivo (segmento) de dados

REMDAD (E1,E2)

E1 - identificação do arquivo de dados de onde será eliminado o registro, alfanumérico com até 7 posições;

E2 - identificador do registro a ser eliminado, numérico zonado com 6 posições;

- Remove registro do arquivo (segmento) de elos

REMELO - (E1,E2,E3)

E1 - identificação do arquivo de elos de onde será eliminado o registro, alfanumérico com até 7 posições;

E2 - identificador principal do registro de elo a ser eliminado, numérico zonado com 6 posições;

E3 - identificador secundário do registro de elo a ser eliminado, numérico zonado com 6 posições;

- Seleciona registro do arquivo (segmento) de dados

SELDAD - (E1,E2,D1,S1)

E1 - identificação do arquivo de dados no qual o registro será selecionado, alfanumérico com até 7 posições;

E2 - identifica o tipo de seleção. Numérico zonado de 1 posição que indica o tipo de seleção:

Ø - seleciona o registro com base no identificador informado no parametro seguinte. Neste caso o parametro seguinte é de entrada.

1 - seleciona o primeiro registro do arquivo;

2 - seleciona o registro seguinte ao anteriormente lido para o arquivo.

3 - seleciona o registro anterior ao último lido no arquivo:

9 - libera o arquivo de dados de pesquisas posteriores. Neste caso os demais parâmetros não contêm informações válidas:

D1 - Identificador do registro selecionado. É parâmetro de entrada se E2 for \emptyset . É de saída se E2 for, 1, 2 ou 3. Se E2 for 9, este parâmetro não é manuseado;

S1 - registro selecionado, alfanumérico com até 502 posições;

- Seleciona registro do arquivo (segmento) de elos

SELELO (E1,E2,D1,D2,D3,D4)

E1 - identificação do arquivo de elos no qual o registro será selecionado; alfanumérico com até 7 posições;

E2 - numérico zonado com 1 posição. Identifica o tipo de seleção que pode ser:

0 - verifica se o registro com os parâmetros D1,D2, D3, D4, existe.

1 - seleciona o primeiro registro do arquivo de elos. Neste caso D1,D2,D3,D4 são parâmetros de saída;

2 - seleciona o registro seguinte ao último selecionado. Os parâmetros D1,D2,D3,D4 são de saída;

4 - seleciona o primeiro registro de uma determinada classe de elo. Neste caso os parâmetros D1, D2 são de entrada e os parâmetros D3 e D4 de saída;

5 - seleciona o registro de elo seguinte da mesma classe. Neste caso os parâmetros D1 e D2 não contêm informações válidas e os parâmetros D3 e D4 são de saída;

9 - Libera o arquivo de elos de pesquisas poste
riores. Neste caso os demais parâmetros não
contem informações válidas;

D1 - Identificação do arquivo de dados principal que
compõe o elo; alfanumérico com até 7 posições. É
parâmetro de entrada quando E2 for \emptyset ou 4; é de saí
da quando E2 for 1, 2 ou 5.

D2 - Identificador principal do registro de elos, numé
rico zonado com 6 posições. É parâmetro de entrada
quando E2 for \emptyset ou 4. É de saída quando E2 for 1,
2 ou 5.

D3 - Identificação do arquivo de dados secundário que
compõe o elo, alfanumérico com até 7 posições. É
parâmetro de entrada quando E2 for \emptyset . É de saída
quando E2 for 1,2,4 ou 5.

D4 - Identificador secundário do registro de elos, numéri
co zonado com 6 posições. É parâmetro de entrada
quando E2 for \emptyset . É de saída quando E2 for 1,2,4 ou
5.

- Seleciona registro do arquivo (segmento) de inversões

SELINV (E1,E2,D1,S1,S2)

E1 - Identificação do arquivo de inversão de onde o re
gistro será selecionado, alfanumérico com até 7 po
sições.

E2 - Numérico zonado com 1 posição. Identifica o tipo
de seleção:

1 - seleciona primeiro identificador de dados do
registro do arquivo de inversões;

2 - seleciona identificador de dados seguinte ao
último selecionado:

4 - seleciona primeiro identificador de uma determi
nada classe de inversão;

- 5 - seleciona identificador de dados na classe, se guinte ao último selecionado;
- 6 - seleciona primeiro identificador da classe se guinte à última selecionada;
- 9 - libera o arquivo de inversões de pesquisas pos teriores. Neste caso os demais parâmetros D1, S1, S2 não contêm informações válidas;

D1 - Identificação do registro de inversão a ser selecionado. Alfanumérico com 249 posições. É parâmetro de entrada quando E2 for 4. É de saída quando E2 for 1, 2, 5 ou 6.

S1 - Identificação do arquivo de dados que deu origem a inversão. Alfanumérico com até 7 posições;

S2 - Identificador do registro de dados selecionado, numérico zonado com 6 posições.

- Modifica registro de arquivo (segmento) de dados:

MODDAD (E1, E2, E3, E4)

E1 - Identificação do arquivo de dados que contém o registro no qual deve ser feita a alteração, alfanumérico com até 7 posições;

E2 - Identificador do registro a ser modificado, numérico zonado com 6 posições;

E3 - Identifica no arquivo (segmento) de estrutura o campo a ser alterado, numérico zonado com 2 posições;

E4 - novo valor do campo, alfanumérico com até 502 posições;

- Ativa Banco de Dados

ATIVDB (E1, E2)

E1 - Numérico compactado 1 posição; indica situação do Banco de Dados. Pode ser:

1 - O Banco de dados não existe esta é sua primeira ativação.

∅ - o Banco de Dados já existe, não sendo a primeira execução sobre o mesmo.

E2 - Numérico compactado 1 posição, indica se vão ser ativados os procedimentos para a avaliação de eficiência e se já existe informações no arquivo. Pode ser:

∅ - Não ativa os procedimentos;

1 - Ativa os procedimentos, não tendo informações anteriores;

2 - Ativa os procedimentos atualizando informações anteriores;

- Desativa Banco de Dados

DATVDB

Sem parâmetros.

- Posição do Banco de Dados

RESMDB

Sem parâmetros.

- Cópia e rearruma o Banco de Dados

COPIDB

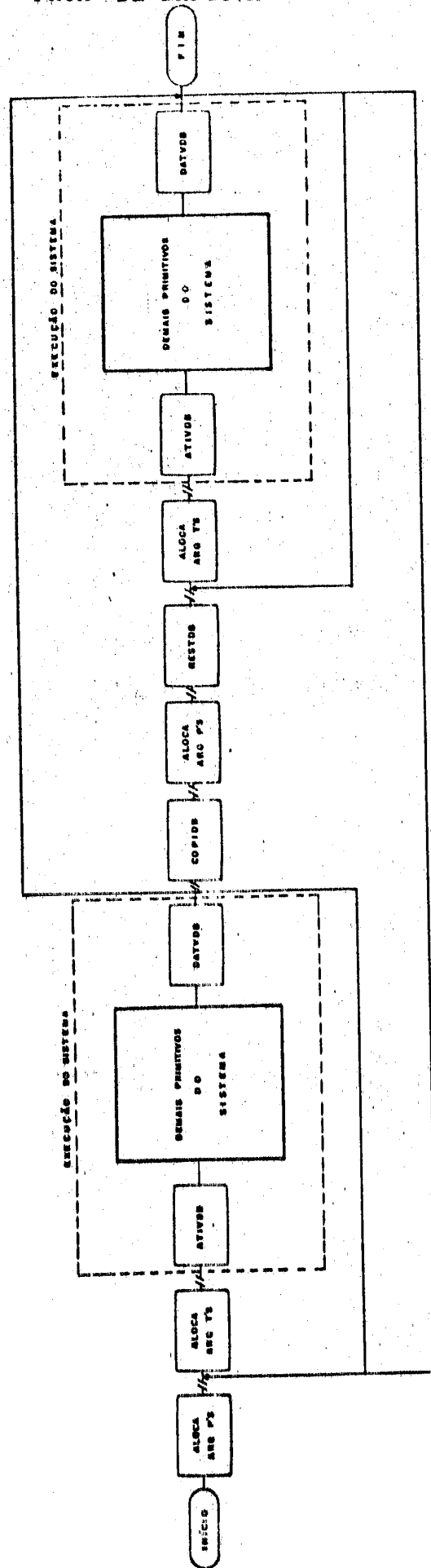
Sem parâmetros.

- Restaura o Banco de Dados

RESTDB

Sem parâmetros.

FIGURA 8 - FLUXO DE EXECUÇÃO DO SISTEMA



3.6 - CÓDIGOS DE RETORNO DOS PRIMITIVOS

Após cada execução de primitivo do sistema, um código de retorno é enviado ao ponto de chamada do primitivo. Este código de retorno indica se o processamento do primitivo foi normal ou, caso contrário, indica o erro detectado.

Relacionamos, a seguir, os códigos de retorno utilizados pelo sistema e a consequente identificação.

CÓDIGO	IDENTIFICAÇÃO
Ø	- primitivo executado sem problema;
1	- segmento (arquivo para o usuário) não consta do Banco de Dados;
2	- segmento já existe no Banco de Dados;
3	- Elemento da estrutura inválido;
4	- programa SORT detetou erro;
5	- parâmetro da procedure fora do formato previsto;
6	- parâmetro inválido;
7	- operação não permitida;
8	- registro não encontrado;
9	- registro ele não existe;
10	- erro no Banco de Dados. Por medida de segurança é conveniente restaurar o Banco de Dados.
11	- fim de arquivo;
12	- fim de classe;
13	- alteração na identificação de um segmento;
14	- chave inválida;
15	- estouro da tabela de dados do usuário;
16	- estouro da tabela de arquivos do usuário do Banco de Dados. Não mais podem ser inseridos arquivos

- vos dos usuários. Faz-se necessário rearrumar o Banco de Dados;
- 17 - falta de área para gravação no arquivo de dados que está sendo utilizado: permanente ou temporário;
 - 18 - falta de área para gravação no arquivo de elos que está sendo utilizado: permanente ou temporário;
 - 19 - falta de área para gravação no arquivo de inversões que está sendo utilizado: permanente ou temporário;
 - 20 - falta de área para gravação no arquivo de estruturas ou dicionário de dados que está sendo utilizado: permanente ou temporário;
 - 21 - registro de elo já existe.

3.7 - RESTRIÇÕES DE IMPLEMENTAÇÃO

No desenvolvimento e implementação do suporte foram assumidas restrições, algumas impostas pelo sistema operacional e outras por necessidade do suporte, visando padronização ou racionalização de seu uso.

As restrições impostas pelo sistema operacional foram:

- Tamanho de chave para arquivos de organização sequencial indexado limitado a 255 bytes.
- Não utilização da primeira posição dos registros dos arquivos sequencial indexado pois o sistema utiliza esta posição para marcar o registro quando o mesmo é "removido", obrigando a chave que é embutida a ter início na 2a. posição dos registros.

- não utilização de acesso simultâneo aos arquivos de modo "sequencial output" e "direct update".

As restrições impostas pelo suporte são:

- o número de arquivos de usuários permitidos no suporte é de 255 incluído arquivos de dados, elos e inversões.
- o número máximo de registros permitidos por arquivo de dados (segmento para o sistema) é de 999.999.
- um arquivo de inversão pode ter a chave constituída de até 20 campos do registro do arquivo de dados base da inversão.
- possibilidade de acesso compartilhado a até 10 arquivos (segmentos) de cada natureza: dados, elos e inversões através dos primitivos de seleção.
- os dados do usuário para a criação de um arquivo (segmento) de dados devem estar sob a forma de arquivo, a estrutura relativa a este arquivo deve estar disponível para o primitivo do suporte sob a forma de tabela em memória.
- Não é permitida a criação de arquivos de elos ou inversões permanentes se os arquivos de dados respectivos forem temporários.

CAPÍTULO 4 - MANUAL DE OPERAÇÃO

4.1 - Execução do sistema

O suporte está constituído operacionalmente por 4 (quatro) fases operativas que se relacionam de acordo com o fluxo indicado na Figura 9. As fases operativas, cujo ciclo de execução é variável, são:

- JOB ALOCA, aloca área necessária para ser utilizada pelos conjuntos de dados permanentes do suporte;
- JOB EXECUTA, onde os primitivos do suporte são manuseados;
- JOB COPIA, copia o Banco de Dados, rearrumando-o, e obtendo cópia sequencial do mesmo;
- JOB RESTAURA, restaura o Banco de Dados a partir da cópia obtida no JOB anterior.

- JOB ALOCA

Sempre utilizado antes da primeira execução do suporte ou antes do JOB RESTAURA. Constituído por programa utilitário IBM, tem por finalidade reservar espaço para os conjuntos de dados considerados permanentes do suporte: dados, elos, inversões e estruturas. A previsão de área para cada conjunto de dados pode ser obtida através de uso de programa do sistema Avaliação de Eficiência de Suporte para Banco de Dados, objeto de tese de mestrado.

- JOB EXECUTA

Utilizado o número de vezes necessárias ao usuário. Deve-se levar em conta as opções básicas no primitivo ATIVDB sobre se o suporte está sendo utilizado pela primeira vez ou se está sendo reexecutado. Ainda neste primitivo é indicada a criação, utilização ou não de conjunto de dados contendo in

formações para avaliação de eficiência. O JOB é constituído de dois Steps. No primeiro é utilizado um programa utilitário IBM com finalidade idêntica ao JOB ALOCA, porém para os conjuntos de dados temporários do suporte que são: dados, elos, inversões e estruturas. No segundo step são executados os primitivos do suporte, em qualquer número e ordem desde que obedecidas as precedências específicas para cada tarefa.

Deve conter obrigatoriamente os conjuntos de dados permanentes e com opção MOD no parâmetro Disp. É importante que o JOB esteja sendo executado corretamente, baseando-se tal fato nos códigos do primeiro parâmetro do primitivo ATIVDB:

Código 1 - o Banco de Dados está sendo criado, portanto, antes da execução deste JOB, foi executado o JOB ALOCA;

0 - o Banco de Dados já existe, portanto, antes da execução deste JOB foi executado o mesmo JOB EXECUTA ou o JOB RESTAURA.

Deve conter os conjuntos de dados temporários que serão utilizados. Todos os conjuntos de dados temporários criados no primeiro step podem ser utilizados.

Para cada conjunto de dados permanente e temporário são necessárias duas definições de dados (DDNAME) a saber:

Definição de dados	Conjunto de dados
ARQDP e ARQDIP	Dados Permanentes
ARQEP e ARQEIP	Elos permanentes
ARQIP e ARQIIP	Inversões permanentes
ARQSP e ARQSIP	Estruturas permanentes
ARQDT e ARQDIT	Dados temporários
ARQET e ARQEIT	Elos temporários
ARQIT e ARQIIT	Inversões temporárias
ARQST e ARQSIT	Estruturas temporárias.

Dependendo do segundo parâmetro indicado no primitivo ATIVDB, o step poderá dar o seguinte tratamento aos conjuntos de

dados para avaliação de eficiência.

Se código 0 - não é necessária apropriação de informações para avaliação de eficiência. Neste caso, as definições de dados ARQSAEI e ARQSAEO deverão ser Dummy's.

Se código 1 - é necessária apropriação de informações para avaliação de eficiência, porém não existe conjunto de dados anterior com tais informações. Neste caso temos a definição de dados (DDNAME) ARQSAEI com a opção Dummy e ARQSAEO ativo representando conjunto de dados a ser criado;

Se código 2 - Indica que existe conjunto de dados anterior de avaliação de eficiência, ao qual deverão ser acrescentadas as informações do presente JOB. Neste caso, ARQSAEI e ARQSAEO representam conjuntos de dados; o primeiro ativo e o segundo a ser criado.

Caso seja utilizado os primitivos CRIINV ou CRICLS é necessário que seja indicada as definições de dados relativas ao programa utilitário IBM SORT, notadamente as áreas de trabalho. O número de áreas e dimensão para estes conjuntos de dados dependerão da quantidade de registros a classificar.

Devem ainda, serem indicados os conjuntos de dados dos usuários que irão dar origem aos segmentos de dados do suporte através do primitivo CRIDAD.

A utilização de SYSPRINT é necessária para a utilização do primitivo RESMDB.

- JOB COPIA

Utilizado sempre que se faz necessária a rearrumação do Banco de Dados ou no intuito apenas de obter cópia de posição do Banco de Dados para efeito de "Back up". Manuseia os

conjuntos de dados permanentes do sistema: dados, elos, inversões e estruturas e um conjunto de dados de saída de forma sequencial.

- JOB RESTAURA

Utilizado para restaurar o Banco de Dados a partir de uma cópia extraída no JOB COPIA. Entre este JOB e o JOB COPIA faz-se necessário o processamento do JOB ALOCA para a nova criação dos conjuntos de dados permanentes do suporte: dados, elos, inversões e estruturas.

Os comandos do JCL referentes aos JOB's citados encontram-se indicados no exemplo da seção seguinte:

4.2 Exemplo de uso do suporte

A seguir indicaremos através de exemplo, a utilização do suporte.

Suponhamos uma empresa com Departamentos de: pessoal, desenvolvimento de sistemas e treinamento que se utilizam do suporte para criar e manter um Banco de Dados contendo informações sobre o pessoal da empresa, projetos em desenvolvimento e habilitações. Cada projeto é relacionado com as tarefas a serem cumpridas. Para a execução de um projeto é criado um grupo de projeto composto de um líder e componentes. A admissão e demissão de um empregado é realizada através do departamento de pessoal verificadas as necessidades da empresa. A demissão só pode ser realizada se o empregado não estiver envolvido em nenhum projeto como componente ou líder.

A criação de grupos de projeto, designação de componentes e de líder é tarefa do Gerente de Desenvolvimento de Sistemas.

Cabe ao Departamento de treinamento formar e aperfeiçoar o pessoal da empresa de acordo com as solicitações dos outros

departamentos.

A primeira tarefa a ser realizada, de acordo com a Figura 9, é a alocação dos conjuntos de dados permanentes do sistema; mostrada no Quadro 9.

A primeira utilização de JOB EXECUTA será para a criação dos conjuntos de dados básicos da empresa. É criado:

- O segmento de dados 'EMP' pela gerência de pessoal com os empregados da empresa contendo nome, salário e profissão;
- O segmento de dados 'TAREFA' contendo as tarefas desenvolvidas, pela gerência de desenvolvimento de sistemas;
- O segmento de dados 'PROJETO', contendo os projetos manuseados pela gerência de desenvolvimento de sistemas;
- O segmento de dados 'HABILIT' contendo as habilitações de forma geral dos empregados da empresa. Criado e manuseado pela gerência de treinamento;
- O segmento de dados 'PROJPES' contendo os componentes de cada projeto. Criado e manuseado pela gerência de desenvolvimento de sistemas;
- O segmento de elos 'MNG' indicando o líder de cada projeto. Vai conter ligações entre os segmentos de dados 'EMP' e 'PROJETO'. É criado e manuseado pela gerência de desenvolvimento de sistemas;
- O segmento de elos 'REQ' indicando para cada tarefa as habilitações necessárias. Vai conter ligações entre os segmentos de dados 'TAREFA' e 'HABILIT'. É criado pela gerência de desenvolvimento de sistema;
- O segmento de elos 'CAP' indicando as habilitações por empregado. Vai conter ligações entre os segmentos de dados 'EMP' e 'HABILIT', é criado e manuseado pela gerência

de treinamento;

- O segmento de elos 'ASN' indicando a cada componente de um projeto, as tarefas a serem desempenhadas. Vai conter ligações entre os segmentos de dados 'PROJPES' e 'TAREFA'. É criado e manuseado pelo líder do Projeto.

Para melhor entendimento do exemplo, foi ainda criado:

- o segmento de inversões 'EMPINV' com base no segmento de dados 'EMP' contendo inversão por nome de empregado;
- O segmento de inversões 'PRJINV' com base no segmento de dados 'PROJPES' contendo inversão por nome de empregado por projeto.

O Quadro 10 ilustra o JOB denominado EXECUTAL utilizado para a criação dos segmentos descritos. No Quadro 11 é indicada a posição do Banco de Dados ao fim da execução do JOB EXECUTAL.

No JOB EXECUTA2, indicado no Quadro 12, a gerência do pessoal efetua a contratação de pessoal. Para isso é necessária a inclusão dos novos empregados no segmento de dados 'EMP'.

Isto é feito através do primitivo INSAD. Porém, antes, deve ser testado se o empregado a ser admitido já pertence a firma. Tal teste é feito através de pesquisa, utilizando o primitivo SELINV, no segmento de inversões 'EMPINV'. Se o código de retorno do primitivo for zero, indica que o empregado já existe. Caso contrário o empregado não existe e pode ser incluído no segmento de dados 'EMP'. Não se faz necessária a inclusão de seu nome no segmento de inversões 'EMPINV' pois a mesma é feita automaticamente pelo primitivo INSDAD. O Quadro 13 indica a posição do Banco de Dados ao fim do JOB EXECUTA2.

Notamos a inclusão de registros de empregados no segmento de dados 'EMP'.

No Quadro 14 é indicado o JOB EXECUTA3. Nele, o gerente de pessoal comanda a demissão de empregados. Porém o empregado só pode ser demitido se não estiver participando de projeto como líder ou componente. Por este motivo, antes da demissão, é utilizado o primitivo SELINV para verificar se o empregado consta do segmento de dados 'EMP'. Caso conste é então pesquisado se o empregado participa de algum projeto, como líder, através do primitivo SELELO no segmento 'MNG', ou como componente, através do primitivo SELINV no segmento "PRJINV".

Caso o código de retorno de ambas as pesquisas indiquem que o empregado não está alocado a projeto, pode então o mesmo ser demitido, através de primitivo REMDAD.

Este primitivo atualiza todos os segmentos nos quais o empregado estiver indicado. O Quadro 15 indica a posição do Banco de Dados ao final do JOB EXECUTA3.

O Quadro 16 mostra os comandos necessários para a execução do JOB COPIA destinado a obtenção de cópia do Banco de Dados.

O Quadro 17 indica os comandos necessários para a execução do JOB RESTAURA, destinado a recomposição do Banco de Dados, tornando-o operativo para os primitivos do sistema.

4.3 - Manutenção do suporte

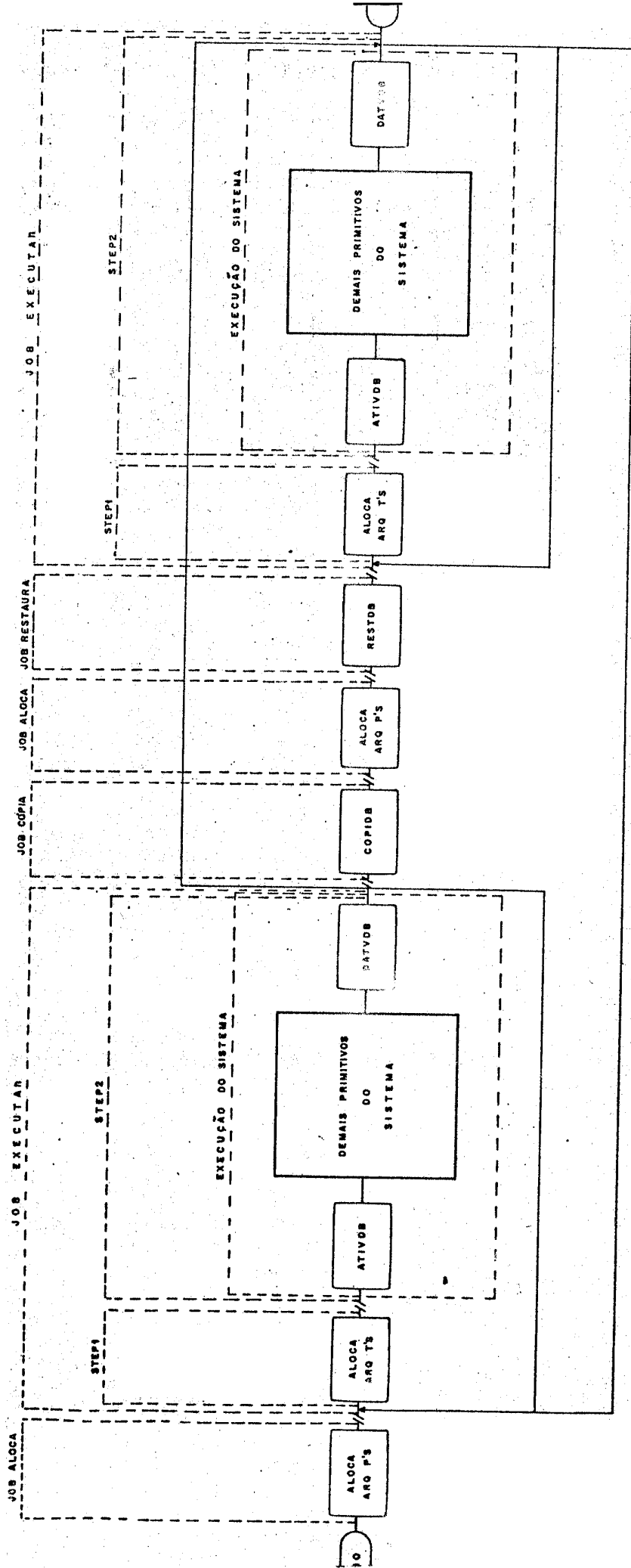
Os módulos do suporte estão armazenados em módulo fonte, em um conjunto de dados particionado denominado PBTX.FURTADO.DBFT. Paralelamente, no conjunto de dados particionado denominado PBTX.FURTADO.DBCG, estão armazenados os módulos do suporte em linguagem objeto, que são utilizados para execução do suporte.

Através de utilitários IBM é benéfico que se tenha sempre em fita magnética cópia destes dois conjuntos de dados como 'Back up'.

Qualquer alteração nos módulos do sistema pode ser feita a través de utilitário IBM IEBUPDTE, podendo ser uma alteração inclusão ou exclusão de comandos.

Novos módulos podem ser incluídos e outros excluídos do sistema. Sugerimos nestes casos que seja criado um item nos manuais indicando os impactos destas alterações no sistema. A exclusão de módulos deve ser realizada de modo criterioso, a fim de evitar a permanência de módulos sem finalidade ou impossibilitar o uso de módulos pela retirada de outros essenciais.

FIGURA 9 - FLUXO OPERACIONAL DO SISTEMA



```

//ALOCA JOB (TRP,CPD,,10),RDP,MSGLEVEL=(1,1),CLASS=A
//STOO EXEC PGM=IFFER14
//ARQDP DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=516,
//      BLKSIZE=5164,DSORG=IS,RKP=5,KEYLEN=9,OPTCD=LIM,
//      NTM=2),LABEL=EXPDT=99365,
//      DSN=ARQP(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,3),DCB=DSORG=IS,DSN=ARQP(PRIME),
//      LABEL=EXPDT=99365,
//      DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQP(OVFLOW),
//      LABEL=EXPDT=99365,
//      DISP=(NEW,CATLG),VOL=SER=DK1002
//ARQEP DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=60,
//      BLKSIZE=1204,DSORG=IS,RKP=5,KEYLEN=16,OPTCD=LIM,
//      NTM=2),LABEL=EXPDT=99365,
//      DSN=ARQEP(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,2),DCB=DSORG=IS,DSN=ARQEP(PRIME),
//      LABEL=EXPDT=99365,DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQEP(OVFLOW),
//      LABEL=EXPDT=99365,DISP=(NEW,CATLG),VOL=SER=DK1002
//ARQIP DD UNIT=SYSDA,SPACE=(CYL,3),DCB=(RECFM=FB,LRECL=336,
//      BLKSIZE=672,DSORG=IS,RKP=1,KEYLEN=255,OPTCD=LIM,
//      NTM=2),LABEL=EXPDT=99365,
//      DSN=ARQIP(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQIP(PRIME),
//      LABEL=EXPDT=99365,DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQIP(OVFLOW),
//      LABEL=EXPDT=99365,DISP=(NEW,CATLG),VOL=SER=DK1002
//ARQSP DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=FB,LRECL=43,
//      BLKSIZE=860,DSORG=IS,RKP=1,KEYLEN=5,OPTCD=LIM,
//      NTM=2),LABEL=EXPDT=99365,
//      DSN=ARQSP(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQSP(PRIME),
//      LABEL=EXPDT=99365,DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQSP(OVFLOW),
//      LABEL=EXPDT=99365,DISP=(NEW,CATLG),VOL=SER=DK1002

```

QUADRO 9 - JOB ALOCA - Aloca área para o Banco de Dados

```

//EXECUTAI JOB (TRP,CPD,,10),ROP,MSGLEVEL=(1,1),CLASS=A
//ST00 EXEC PGM=IFFBR14
//ARQDT DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=516,
//      BLKSIZE=5164,DSORG=IS,RKP=5,KEYLEN=9,OPTCD=LIM,
//      NTM=2),
//      DSN=ARQDT(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,3),DCB=DSORG=IS,DSN=ARQDT(PRIME),
//      DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQDT(OVFLOW),
//      DISP=(NEW,CATLG),VOL=SER=DK1002
//ARQET DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=60,
//      BLKSIZE=1204,DSORG=IS,RKP=5,KEYLEN=16,OPTCD=LIM,
//      NTM=2),
//      DSN=ARQET(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,2),DCB=DSORG=IS,DSN=ARQET(PRIME),
//      DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQET(OVFLOW),
//      DISP=(NEW,CATLG),VOL=SER=DK1002
//ARQIT DD UNIT=SYSDA,SPACE=(CYL,6),DCB=(RECFM=FB,LRECL=336,
//      BLKSIZE=672,DSORG=IS,RKP=1,KEYLEN=255,OPTCD=LIM,
//      NTM=2),
//      DSN=ARQIT(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,2),DCB=DSORG=IS,DSN=ARQIT(PRIME),
//      DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQIT(OVFLOW),
//      DISP=(NEW,CATLG),VOL=SER=DK1002
//ARQST DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=FB,LRECL=43,
//      BLKSIZE=860,DSORG=IS,RKP=1,KEYLEN=5,OPTCD=LIM,
//      NTM=2),
//      DSN=ARQST(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQST(PRIME),
//      DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQST(OVFLOW),
//      DISP=(NEW,CATLG),VOL=SER=DK1002
//ST1 EXEC PPL1LFCLG,PARM.PL1L=(A,X),REGION.PL1L=128K,
//      PARM.LKED='SIZE=200000',TIME.GO=15,REGION.GO=300K
//PL1L.SYSIN DD *
EXEMPLO1: PROC OPTIONS (MAIN);
/* PROCEDURES ATIVADAS
DCL ATIVDB ENTRY (FIXED(1),FIXED(1))
/* ATIVDB RETURNS (BIN FIXED);
DCL CRIDAD ENTRY (CHAR(7),CHAR(1),CHAR(7),
/* CRIDAD PIC'(6)9',PIC'99',CHAR(*))
/* CRIDAD RETURNS (BIN FIXED);
DCL CRIINV ENTRY (CHAR(7),CHAR(1),CHAR(7),
/* CRIINV PIC'99',PIC'99')
/* CRIINV RETURNS (BIN FIXED);
DCL CRIELO ENTRY (CHAR(7),CHAR(1),
/* CRIELO PIC'(6)9',CHAR(7),CHAR(7))
/* CRIELO RETURNS (BIN FIXED);
DCL INSELO ENTRY (CHAR(7),CHAR(7),PIC'(6)9',
/* INSELO CHAR(7),PIC'(6)9')
/* INSELO RETURNS (BIN FIXED);
DCL RESMDB ENTRY:
DCL DATVDB RETURNS (BIN FIXED);
DCL FMPS (4) CHAR (39) INIT
/* ESTRUTURA ARQUIVO EMP 001000CH',
/* 2NOME EMPREGADO 001000CH',
/* 2SALARIO 041007CH',

```

```

DCL PROJETS (2)          '2PROFISSAO          048080CH');
                          CHAR (39) INIT
                          ('1ESTRUTURA ARQUIVO PROJETO 001020CH',
                          '2NOME DO PROJETO           001020CH');
DCL TAREFAS (2)         CHAR (39) INIT
                          ('1ESTRUTURA ARQUIVO TAREFA  001015CH',
                          '2NOME DA TAREFA             001015CH');
DCL HABILIS (2)        CHAR (39) INIT
                          ('1ESTRUTURA ARQUIVO HABILITACAO 001030CH',
                          '2HABILITACAO                001030CH');
DCL PRUPESS (3)        CHAR (39) INIT
                          ('1ESTRUTURA ARQUIVO PROJES  001060CH',
                          '2NOME DO PROJETO           001020CH',
                          '2NOME DO EMPREGADO         021060CH');
DCL DOMEMP (1)         PIC '99' INIT (02);
DCL DOMPRJ (1)         PIC '99' INIT (03);
DCL COD                 BIN FIXED; /* CODIGO DE RETORNO */
OPEN FILE (SYSPRINT) LINESIZE (132);
COD = ATIVDB (1,1);
COD = CRIDAD ('EMP','P','EMPI',50,4,FMPS);
COD = CRIDAD ('PROJETO','P','PROJETI',10,2,PROJETS);
COD = CRIDAD ('TAREFA','P','TAREFAI',20,2,TAREFAS);
COD = CRIDAD ('HABILIT','P','HABILII',15,2,HABILIS);
COD = CRIDAD ('PROJPES','P','PRJPEST',30,3,PRJPESS);
COD = CRIELO ('MNG','P',5,'PROJETO','EMP');
COD = CRIELO ('REQ','P',13,'TAREFA','HABILIT');
COD = CRIELO ('CAP','P',83,'EMP','HABILIT');
COD = CRIELO ('ASN','P',25,'PROJPES','TAREFA');
COD = CRIINV ('EMPINV','P','EMP',01,DOMEMP);
COD = CRIINV ('PRJINV','P','PROJPES',01,DOMPRJ);
COD = INSELO ('MNG','PROJETO',000001,'EMP',000008);
COD = INSELO ('MNG','PROJETO',000002,'EMP',000009);
COD = INSELO ('MNG','PROJETO',000003,'EMP',000014);
COD = INSELO ('MNG','PROJETO',000004,'EMP',000016);
COD = INSELO ('MNG','PROJETO',000005,'EMP',000017);
COD = INSELO ('REQ','TAREFA',000001,'HABILIT',000004);
COD = INSELO ('REQ','TAREFA',000001,'HABILIT',000010);
COD = INSELO ('REQ','TAREFA',000002,'HABILIT',000004);
COD = INSELO ('REQ','TAREFA',000002,'HABILIT',000010);
COD = INSELO ('REQ','TAREFA',000003,'HABILIT',000004);
COD = INSELO ('REQ','TAREFA',000003,'HABILIT',000010);
COD = INSELO ('REQ','TAREFA',000004,'HABILIT',000002);
COD = INSELO ('REQ','TAREFA',000004,'HABILIT',000007);
COD = INSELO ('REQ','TAREFA',000004,'HABILIT',000008);
COD = INSELO ('REQ','TAREFA',000005,'HABILIT',000007);
COD = INSELO ('REQ','TAREFA',000005,'HABILIT',000008);
COD = INSELO ('REQ','TAREFA',000005,'HABILIT',000011);
COD = INSELO ('REQ','TAREFA',000005,'HABILIT',000002);
COD = INSELO ('CAP','EMP',000001,'HABILIT',000001);
COD = INSELO ('CAP','EMP',000002,'HABILIT',000001);
COD = INSELO ('CAP','EMP',000002,'HABILIT',000005);
COD = INSELO ('CAP','EMP',000003,'HABILIT',000001);
COD = INSELO ('CAP','EMP',000003,'HABILIT',000005);
COD = INSELO ('CAP','EMP',000004,'HABILIT',000001);
COD = INSELO ('CAP','EMP',000004,'HABILIT',000005);
COD = INSELO ('CAP','EMP',000023,'HABILIT',000001);
COD = INSELO ('CAP','EMP',000023,'HABILIT',000005);
COD = INSELO ('CAP','EMP',000005,'HABILIT',000002);
COD = INSELO ('CAP','EMP',000005,'HABILIT',000012);

```



```

COD = INSELO ('CAP', 'EMP', 000016, 'HABILIT', 000013);
COD = INSELO ('CAP', 'EMP', 000017, 'HABILIT', 000014);
COD = INSELO ('CAP', 'EMP', 000017, 'HABILIT', 000010);
COD = INSELO ('CAP', 'EMP', 000018, 'HABILIT', 000014);
COD = INSELO ('CAP', 'EMP', 000018, 'HABILIT', 000010);
COD = INSELO ('CAP', 'EMP', 000018, 'HABILIT', 000007);
COD = INSELO ('CAP', 'EMP', 000022, 'HABILIT', 000002);
COD = INSELO ('CAP', 'EMP', 000022, 'HABILIT', 000006);
COD = INSELO ('CAP', 'EMP', 000024, 'HABILIT', 000002);
COD = INSELO ('CAP', 'EMP', 000024, 'HABILIT', 000006);
COD = INSELO ('CAP', 'EMP', 000024, 'HABILIT', 000011);
COD = INSELO ('ASN', 'PROJPES', 000001, 'TAREFA', 000001);
COD = INSELO ('ASN', 'PROJPES', 000001, 'TAREFA', 000002);
COD = INSELO ('ASN', 'PROJPES', 000002, 'TAREFA', 000003);
COD = INSELO ('ASN', 'PROJPES', 000003, 'TAREFA', 000005);
COD = INSELO ('ASN', 'PROJPES', 000004, 'TAREFA', 000005);
COD = INSELO ('ASN', 'PROJPES', 000005, 'TAREFA', 000001);
COD = INSELO ('ASN', 'PROJPES', 000006, 'TAREFA', 000001);
COD = INSELO ('ASN', 'PROJPES', 000007, 'TAREFA', 000002);
COD = INSELO ('ASN', 'PROJPES', 000007, 'TAREFA', 000003);
COD = INSELO ('ASN', 'PROJPES', 000008, 'TAREFA', 000004);
COD = INSELO ('ASN', 'PROJPES', 000009, 'TAREFA', 000005);
COD = INSELO ('ASN', 'PROJPES', 000010, 'TAREFA', 000003);
COD = INSELO ('ASN', 'PROJPES', 000011, 'TAREFA', 000004);
COD = INSELO ('ASN', 'PROJPES', 000012, 'TAREFA', 000004);
COD = INSELO ('ASN', 'PROJPES', 000013, 'TAREFA', 000005);
COD = INSELO ('ASN', 'PROJPES', 000014, 'TAREFA', 000005);
COD = INSELO ('ASN', 'PROJPES', 000015, 'TAREFA', 000001);
COD = INSELO ('ASN', 'PROJPES', 000016, 'TAREFA', 000002);
COD = INSELO ('ASN', 'PROJPES', 000016, 'TAREFA', 000003);
COD = INSELO ('ASN', 'PROJPES', 000017, 'TAREFA', 000004);
COD = INSELO ('ASN', 'PROJPES', 000018, 'TAREFA', 000004);
COD = INSELO ('ASN', 'PROJPES', 000019, 'TAREFA', 000005);
COD = INSELO ('ASN', 'PROJPES', 000020, 'TAREFA', 000005);
COD = INSELO ('ASN', 'PROJPES', 000021, 'TAREFA', 000005);
COD = INSELO ('ASN', 'PROJPES', 000022, 'TAREFA', 000005);
COD = INSELO ('ASN', 'PROJPES', 000019, 'TAREFA', 000004);
COD = INSELO ('ASN', 'PROJPES', 000016, 'TAREFA', 000005);
CALL RESMDB;
COD = DATVDB;
END EXEMPLO1;
//LKED.DBMO DD DSN=DBMO,DISP=OLD
//LKED.SYSIN DD *
INCLUDE DBMO(ATIVCB)
INCLUDE DBMO(AUX1)
INCLUDE DBMO(AUX2)
INCLUDE DBMO(AUX3)
INCLUDE DBMO(CRIDAD)
INCLUDE DBMO(CRIELO)
INCLUDE DBMO(CRIINV)
INCLUDE DBMO(CRIDUP)
INCLUDE DBMO(DATVDB)
INCLUDE DBMO(INSELO)
INCLUDE DBMO(RESMDB)
INCLUDE DBMO(SUB1)
INCLUDE DBMO(SUB2)
INCLUDE DBMO(SUB3)
INCLUDE DBMO(SUB4)
INCLUDE DBMO(SUB5)

```

```

//GO.ARQDP DD DSN=ARQP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQEP DD DSN=ARQEP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// CD DSN=ARQEP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQEP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQIP DD DSN=ARQIP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// CD DSN=ARQIP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQSP CD DSN=ARQSP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// CD DSN=ARQSP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQSP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQDIP DD DSN=ARQP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQEIP DD DSN=ARQEP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQEP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQEP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQIIP DD DSN=ARQIP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQSIIP DD DSN=ARQSP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQSP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQSP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQDT CD DSN=ARQT(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQT(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// CD DSN=ARQT(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQET DD DSN=ARQET(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQET(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// CD DSN=ARQET(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQIT DD DSN=ARQIT(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// CD DSN=ARQIT(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQST CD DSN=ARQST(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQST(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQST(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQBIT DD DSN=ARQT(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQT(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQT(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQEIT DD DSN=ARQET(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQET(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQET(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQIIT DD DSN=ARQIT(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQSIIT DD DSN=ARQST(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQST(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQST(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//GO.SORTLST DD SYSOUT=A
//GO.SYSOUT DD SYSOUT=A
//GO.SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)
//GO.SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)
//GO.SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)
//GO.SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)
//GO.ARQSAEI DD DUMMY
//GO.ARQSAEO DD DSN=EFIC(+1),DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
// LABEL=RETPD=3,VOL=SER=DK1002,SPACE=(TRK,(1)),
// DCB=(RECFM=VB,LRECL=40,BLKSIZE=404)

```


//GO.EMPI DD *	
MARIA LUCIA	280000DIGITADOR
MARIA ISaura	280000DIGITADOR
MARIA JOSE	300000DIGITADOR
MARIA CLARA	310000DIGITADOR
VALERIA REGINA	987500PROGRAMADOR
REGINA LUCIA	1450300PROGRAMADOR
AFRANIO SILVA	200000CANALISTA
ROBERVAL SANTOS	255610CANALISTA
LUCIA MARIA	221140CANALISTA
LUCIA FERREIRA	130000PROGRAMADOR
CRISTINA DA SILVA	130000PROGRAMADOR
RENATO SILVA	130000PROGRAMADOR
ARNALDO CARLOS	130000PROGRAMADOR
ANTONIO FERNANDES	200000CANALISTA
CARLOS MEDEIROS	200000CANALISTA
DANUZA CAMARGO	200000CANALISTA
ELIAS AAGOSTINHO	200000CANALISTA
FRANCISCO JOSE	200000CANALISTA
FERNANDO ANTUNES	1258300PROGRAMADOR
FREDERICO ASSIS	1258300PROGRAMADOR
JORGE PACHECO	1258300PROGRAMADOR
MOACIR SANTOS	0900000OPERADOR
MIGUEL ASSUNCAO	300000DIGITADOR
MAURC SANTOS	900000OPERADOR
//GO.PROJETI DD *	
PAGAMENTO	
ESTOQUE	
CONTABILIDADE	
EMPRESTIMOS	
FINANCIAMENTOS	
CONTA CORRENTE	
EMISSAO DE CHEQUES	
CONTROLE DE ACOES	
INVESTIMENTOS	
APLICACOES	
//GO.TAREFAI DD *	
LEVANTAMENTO	
CONCEPCAO	
ANALISE	
PROGRAMACAO	
IMPLANTACAO	
//GO.HABILII DD *	
PRIMEIRO GRAU COMPLETO	
SEGUNDO GRAU COMPLETO	
SUPERIOR INCOMPLETO	
SUPERIOR COMPLETO	
OPERACAO DE TECLADO	
OPERACAO DE COMPUTADOR	
PROGRAMACAO COBOL	
PROGRAMACAO PL/I	
PROGRAMACAO ASSEMBLER	
ANALISE DE SISTEMA	
SISTEMA OPERACIONAL	
FUNDAMENTOS	
EXTENSAO EM ANALISE	
POS-GRADUACAO	
//GO.PRJPESI DD *	
PAGAMENTO	AFRANIO SILVA

PAGAMENTO	LUCIA FERREIRA
PAGAMENTO	MOARCIR SANTOS
PAGAMENTO	MARIA LUCIA
ESTOQUE	CARLOS MEDEIROS
CONTABILIDADE	DANUZA CAMARGO
CONTABILIDADE	ELIAS AGOSTINHO
CONTABILIDADE	CRISTINA DA SILVA
CONTABILIDADE	FERNANDO ANTUNES
EMPRESTIMOS	FRANCISCO JOSE
EMPRESTIMOS	REGINA LUCIA
EMPRESTIMOS	JORGE PACHECO
EMPRESTIMOS	MARIA CLARA
EMPRESTIMOS	MARIA JOSE
FINANCIAMENTOS	LUCIA MARIA
FINANCIAMENTOS	ROBERVAL SANTOS
FINANCIAMENTOS	JORGE PACHECO
FINANCIAMENTOS	FERNANDO ANTUNES
FINANCIAMENTOS	MAURA SANTOS
FINANCIAMENTOS	MOACIR SANTOS
FINANCIAMENTOS	MIGUEL ASSUNCAO
FINANCIAMENTOS	MARIA LUCIA

//

QUADRO 11 - Posição do Banco de Dados após processamento do JOB EXECUTAL

PROJTO HYDES
 POSICAO DA BASE DE DADOS EM 24/01/78

PAGINA 1

ARQUIVOS DE DADOS

IDENTIFICACAO	TAMANHO REGISTRO	DIMENSAO ARQUIVO	NUM INICIAL REGISTROS	NUMERO INSERCOES	NUMERO DELECOES	NUM ELCS PERMANENT	NUM INVERSJES PERMANENTES
EMP	90	50	24	0	0	2	1
PROJETO	30	10	10	0	0	1	0
TAREFA	25	20	5	0	0	2	0
HABILIT	40	15	14	0	0	2	0
PROJRES	70	30	22	0	0	1	1

IDENTIFICACAO	ARQUIVO DE EL0S				NUMERO DELECOES
	ARQUIVOS ENVOLVIDOS PRINCIPAL	SECUNDARIO	DIMENSAO ARQUIVO	NUMERO INSERCOES	
MNG	PROJFC	EMP	5	20	0
REQ	TAREFA	HABILIT	13	37	0
CAP	EMP	HABILIT	83	199	0
ASN	PROJPE	TAREFA	25	81	0

Cont.

IDENTIFICACAO	ARQUIVO BASE	ARQUIVOS DE INVERSOES	NUMERO DOS CAMPOS CLASSIFICADOS	NUM INICIAL INVERSOES	NUMERO INVERSOES	NUMERO DELECOES
EMPINV	EMP	2		48	0	0
PRJINV	PROJPES	3		38	0	0

QUADRO 11

Cont.

```

//EXECUTA2 JOB (TRP,CPD,,10),RDP,MSGLEVEL=(1,1),CLASS=A
//ST00 EXEC PGM=IFFBR14
//ARQDT DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=516,
// BLKSIZE=5164,DSORG=IS,RKP=5,KEYLEN=9,OPTCD=LIM,
// NTM=2),
// DSN=ARQDT(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,2),DCB=DSORG=IS,DSN=ARQDT(PRIME),
// DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQDT(OVFLOW),
// DISP=(NEW,CATLG),VOL=SER=DK1002
//ARQST DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=FB,LRECL=43,
// BLKSIZE=860,DSORG=IS,RKP=1,KEYLEN=5,OPTCD=LIM,
// NTM=2),
// DSN=ARQST(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQST(PRIME),
// DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQST(OVFLOW),
// DISP=(NEW,CATLG),VOL=SER=DK1002
//ARQIT DD UNIT=SYSDA,SPACE=(CYL,6),DCB=(RECFM=FB,LRECL=336,
// BLKSIZE=672,DSORG=IS,RKP=1,KEYLEN=255,OPTCD=LIM,
// NTM=2),
// DSN=ARQIT(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,2),DCB=DSORG=IS,DSN=ARQIT(PRIME),
// DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQIT(OVFLOW),
// DISP=(NEW,CATLG),VOL=SER=DK1002
//ARQET DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=60,
// BLKSIZE=1204,DSORG=IS,RKP=5,KEYLEN=16,OPTCD=LIM,
// NTM=2),
// DSN=ARQET(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQET(PRIME),
// DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQET(OVFLOW),
// DISP=(NEW,CATLG),VOL=SER=DK1002
//ST1 EXEC PL1LFC LG,PARM,PL1L=(A,X),REGION,PL1L=128K,
// PARM.LKED='SIZE=200000',TIME,GO=15,REGION.GO=200K

```

```

//PL1L,SYSIN DD *

```

```

EXEMPLO2: PROC OPTIONS (MAIN);

```

```

/* PROCEDURES ATIVADAS

```

```

DCL ATIVDB ENTRY (FIXED(1),FIXED(1))
RETURNS (BIN FIXED);
DCL INSDAD ENTRY (CHAR(7),CHAR(502),PIC'(6)9')
RETURNS (BIN FIXED);
DCL SELINV ENTRY (CHAR(7),PIC'9',CHAP(249),
CHAR(7),PIC'(6)9')
RETURNS (BIN FIXED);
DCL RES4DB ENTRY;
DCL DATVDB RETURNS (BIN FIXED);
DCL COD BIN FIXED INIT (0);
DCL 1 EMPREGADO,
2 NOME CHAR (40),
2 SAL CHAR (7),
2 PROF CHAP (23),
2 FILLER CHAR (10);
DCL ARQ_DAD CHAR (7), /* IDENTIFICACAO ARQ DAD
IDENT PIC '(6)9', /* IDENTIFICADOR REG DAD
IDENTS PIC '(6)9'; /* IDENTIF SECUNDARIO
ON ENDFILE (SYSIN)
BEGIN;

```

```

CALL RESMDB;
COD = DATVDB;
EXIT;
END;
OPEN FILE (SYSPRINT) LINESIZE (132);
COD = ATIVDB (0,2);
DO WHILE ('1'B);
  READ FILE (SYSIN) INTO (EMPREGADO);
  /* VE EMPREGADO EXISTE */
  COD = SELINV ('EMPINV',4,NOME,ARQ_DAD,IDENT);
  IF COD = 0
    /* INCLUI EMPREGADO */
  THEN COD = INSDAD ('EMP',EMPREGADO,IDENT);
  ELSE PUT SKIP(1) EDIT ('EMPREGADO JAH CONSTA DO CADASTRO')
    (A);

```

```
END;
```

```
END EXEMPLO2;
```

```
//LKED.DBMO DD DSN=DBMC,DISP=OLD
```

```
//LKED.SYSIN DD *
```

```
  INCLUDE DBMO(ATIVDB)
```

```
  INCLUDE DBMO(AUX1)
```

```
  INCLUDE DBMO(AUX2)
```

```
  INCLUDE DBMO(AUX3)
```

```
  INCLUDE DBMO(DATVDB)
```

```
  INCLUDE DBMO(INSDAD)
```

```
  INCLUDE DBMO(RESMDB)
```

```
  INCLUDE DBMO(SELINV)
```

```
  INCLUDE DBMO(SUB1)
```

```
  INCLUDE DBMO(SUB2)
```

```
  INCLUDE DBMO(SUB3)
```

```
  INCLUDE DBMO(SUB4)
```

```
  INCLUDE DBMO(SUB5)
```

```
//GO.ARQCP DD DSN=ARQP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
//GO.ARQEP DD DSN=ARQEP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQEP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQEP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
//GO.ARQIP DD DSN=ARQIP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQIP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQIP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
//GO.ARQSP DD DSN=ARQSP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQSP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQSP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
//GO.ARQDIP DD DSN=ARQP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
//GO.ARQEIP DD DSN=ARQEP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQEP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQEP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
//GO.ARQIIP DD DSN=ARQIP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQIP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQIP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
//GO.ARQSTIP DD DSN=ARQSP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQSP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=ARQSP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
//GO.ARQOT DD DSN=AROT(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=AROT(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```
// DD DSN=AROT(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
```

```

//GO.ARQET CD DSN=ARQET(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// CD DSN=ARQET(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// CD DSN=ARQET(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQIT DD DSN=ARQIT(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQST DD DSN=ARQST(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQST(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQST(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQIT DD DSN=ARQIT(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQIT DD DSN=ARQIT(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQST DD DSN=ARQST(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQST(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQST(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//GO.SORTLST DD SYSOUT=A
//GO.SYSOUT DD SYSOUT=A
//GO.SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10),,CCNTIG)
//GO.SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(10),,CCNTIG)
//GO.SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(10),,CCNTIG)
//GO.SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(10),,CCNTIG)
//GO.ARQSAEI DD DSN=EFIC(0),DISP=CLD
//GO.ARQSAED DD DSN=EFIC(+1),DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
// LABEL=RETPD=3,VOL=SER=0K1002,SPACE=(TRK,(1)),
// DCB=(RECFM=VB,LRECL=40,BLKSIZE=404)
//GO.SYSIN DD *
FERNANDO ANTUNES
LUCIA HELENA
ROBERVAL SANTOS
MARIA CLARA
//

```


PROJETO HYDES
 POSICAO DA BASE DE DADOS EM 24/01/78

ARQUIVOS DE DADOS

PAGINA 1

IDENTIFICACAO	TAMANHO REGISTRO	DIMENSAO ARQUIVO	NUM INICIAL REGISTROS	NUMERO INSERCOES	NUMERO DELECOES	VJM ELOS PERMANENT	NUM INVERSSES PERMANENTES
EMP	50	50	24	1	0	2	1
PROJETO	30	10	10	0	0	1	0
TAREFA	25	20	5	0	0	2	0
HABILIT	40	15	14	0	0	2	0
PROJPES	70	30	22	0	0	1	1

QUADRO 13 - Resultado do processamento do JOB EXECUTA2 com posicao de Banco de Dados.

ARQUIVO DE ELDS

IDENTIFICACAO	ARQUIVOS ENVOLVIDOS PRINCIPAL	SECUNDARIOS	DIMENSÃO ARQUIVO	NÚMERO INSERÇÕES	NÚMERO DELEÇÕES
MNG	PROJETS	EMP	5	20	0
REQ	TAREFA	HABILIT	13	37	0
CAP	EMP	HABILIT	83	199	0
ASN	PROJETS	TAREFA	25	81	0

QUADRO 13

Cont:

ARQUIVOS DE INVERSOES

PAGINA 3

IDENTIFICACAO	ARQUIVO BASE	NUMERO DOS CAMPOS CLASSIFICADOS	NUM INICIAL INVERSOES	NUMERO INSERCOES	NUMERO DELECOES
EMPINV	EMP	2	48	2	0
PRJINV	PROJPES	3	38	0	0

QUADRO 13

Cont.

QUADRO 13

Cont.

EMPREGADO JAH CONSTA DO CADASTRO
EMPREGADO JAH CONSTA DO CADASTRO
EMPREGADO JAH CONSTA DO CADASTRO

```

//EXECUTA3 JOB (TRP,CPD,,10),ROP,MSGLEVEL=(1,1),CLASS=A
//ST00 EXEC PGM=IEFBR14
//ARQDT DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=516,
// BLKSIZE=5164,DSORG=IS,RKP=5,KEYLEN=9,OPTCD=LIM,
// NTM=2),
// DSN=ARQT(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,3),DCB=DSCRG=IS,DSN=ARQT(PRIME),
// DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSCRG=IS,DSN=ARQT(OVFLOW),
// DISP=(NEW,CATLG),VOL=SER=DK1002
//ARQST DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=FB,LRECL=43,
// BLKSIZE=860,DSORG=IS,RKP=1,KEYLEN=5,OPTCD=LIM,
// NTM=2),
// DSN=ARQST(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSCRG=IS,DSN=ARQST(PRIME),
// DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSCRG=IS,DSN=ARQST(OVFLOW),
// DISP=(NEW,CATLG),VOL=SER=DK1002
//ARQIT DD UNIT=SYSDA,SPACE=(CYL,6),DCB=(RECFM=FB,LRECL=336,
// BLKSIZE=672,DSORG=IS,RKP=1,KEYLEN=255,OPTCD=LIM,
// NTM=2),
// DSN=ARQIT(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,2),DCB=DSCRG=IS,DSN=ARQIT(PRIME),
// DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSCRG=IS,DSN=ARQIT(OVFLOW),
// DISP=(NEW,CATLG),VOL=SER=DK1002
//ARQET DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=60,
// BLKSIZE=1204,DSORG=IS,RKP=5,KEYLEN=16,OPTCD=LIM,
// NTM=2),
// DSN=ARQET(INDEX),DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSCRG=IS,DSN=ARQET(PRIME),
// DISP=(NEW,CATLG),VOL=SER=DK1002
// DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSCRG=IS,DSN=ARQET(OVFLOW),
// DISP=(NEW,CATLG),VOL=SER=DK1002
//ST1 EXEC PL1LFC LG,PARM.PL1L=(A,X),REGION.PL1L=128K,
// PARM.LKED='SIZE=200000',TIME.GO=15,REGION.GO=200K
//PL1L.SYSIN DD *
EXEMPL03: PROC OPTIONS (MAIN);
/* PROCEDURES ATIVADAS */
DCL ATIVDB ENTRY (FIXED(1),FIXED(1))
RETURNS (BIN FIXED);
DCL REMDAD ENTRY (CHAR(7),PIC'(6)9')
RETURNS (BIN FIXED);
DCL SELELO ENTRY (CHAR(7),PIC'9',CHAR(7),
PIC'(6)9',CHAR(7),PIC'(6)9')
RETURNS (BIN FIXED);
DCL SELINV ENTRY (CHAR(7),PIC'9',CHAR(249),
CHAR(7),PIC'(6)9')
RETURNS (BIN FIXED);
DCL RESMDB ENTRY;
DCL DATVDB RETURNS (BIN FIXED);
DCL GOO BIN FIXED INIT (0);
DCL 1 EMPREGADO,
2 NOME CHAR (40),
2 SAL CHAR (7),
2 PROF CHAR (23),
2 FILLER CHAR (10);
DCL ARQ_DAD CHAR (7), /* IDENTIFICACAO ARQ_DAD */
IDENT PIC '(6)9', /* IDENTIFICADOR REG_DAD */

```

```

IDENTS          PIC '(019)'; /* IDENTIF SECUNDARIO */
ON ENDFILE (SYSIN)
  BEGIN;
  CALL RESMDB;
  COD = DATVDB;
  EXIT;
  END;
OPEN FILE (SYSPRINT) LINESIZE (132);
COD = ATIVDB (0,2);
DO WHILE ('1'B);
  READ FILE (SYSIN) INTO (EMPREGADO);
  /* VE EMPREGADO EXISTE */
COD = SELINV ('EMPINV',4,NOME,ARQ_DAD,IDENT);
  IF COD = 0
  THEN PUT SKIP(1) EDIT ('EMPREGADO NAO CONSTA DO CADASTRO')
      (A);
  ELSE DO;
    COD = SELELO ('MNG',4,'EMP',IDENT,ARQ_DAD,IDENTS);
    IF COD = 0
    THEN PUT SKIP(1) EDIT ('EMPREGADO EH LIDER DE PROJETO
        'NAO PODE SER DISPENSADO') (A);
    ELSE DO;
      COD = SELINV ('PRJINV',4,NOME,ARQ_DAD,IDENTS);
      IF COD = 0
      THEN PUT SKIP(1) EDIT ('EMPREGADO ALOCADO A',
          ' PROJETO NAO PODE SER DISPENSADO') (A);
      ELSE COD = REMDAD ('EMP',IDENT);
      END;
    END;
  END;
END EXEMPLO3;
//LKED,DBMO DD DSN=DBMO,DISP=OLD
//LKED,SYSIN DD *
  INCLUDE DBMO(ATIVDB)
  INCLUDE DBMO(AUX1)
  INCLUDE DBMO(AUX2)
  INCLUDE DBMO(AUX3)
  INCLUDE DBMO(DATVDB)
  INCLUDE DBMO(REMDAD)
  INCLUDE DBMO(RESMDB)
  INCLUDE DBMO(SELELO)
  INCLUDE DBMO(SELINV)
  INCLUDE DBMO(SUB1)
  INCLUDE DBMO(SUB2)
  INCLUDE DBMO(SUB3)
  INCLUDE DBMO(SUB4)
  INCLUDE DBMO(SUB5)
//GO,ARQDP DD DSN=ARQP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
//          DD DSN=ARQP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
//          DD DSN=ARQP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO,ARQEP DD DSN=ARQEP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
//          DD DSN=ARQEP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
//          DD DSN=ARQEP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO,ARQIP DD DSN=ARQIP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
//          DD DSN=ARQIP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
//          DD DSN=ARQIP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO,ARQSP DD DSN=ARQSP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
//          DD DSN=ARQSP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
//          DD DSN=ARQSP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS

```

```

//GO.ARQCIP DD DSN=ARQP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQEP DD DSN=ARQEP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQEP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQEP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQIP DD DSN=ARQIP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQSP DD DSN=ARQSP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQSP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQSP(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQDT DD DSN=ARQT(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQT(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQT(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQET DD DSN=ARQET(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQET(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQET(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQIT DD DSN=ARQIT(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQST DD DSN=ARQST(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQST(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQST(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQDIT DD DSN=ARQT(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQT(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQT(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQEIT DD DSN=ARQET(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQET(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQET(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQIIT DD DSN=ARQIT(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIT(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQST DD DSN=ARQST(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQST(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQST(OVFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//GO.SORTLST DD SYSOLT=A
//GO.SYSOUT DD SYSOUT=A
//GO.SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10),,CNTIG)
//GO.SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(10),,CNTIG)
//GO.SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(10),,CNTIG)
//GO.SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(10),,CNTIG)
//GO.ARQSAEI DD DSN=EFIC(0),DISP=CLD
//GO.ARQSAEO DD DSN=EFIC(+1),DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
// LABEL=RETPD=3,VOL=SER=DK1002,SPACE=(TRK,(1)),
// DCB=(RECFM=VB,LRECL=40,BLKSIZE=404)
//GO.SYSIN DD *
ANTONIC FERNANDES 200000CANALISTA
LUIS AUGUSTO 120000PROGRAMADOR
//

```

QUADRO 15 - Resultado do processamento do JOB EXECUTA2 com posição do Banco de Dados

EMPREGADC NAD CONSTA DO CADASTRO

PROJETO HYDES
 POSICAO DA BASE DE DADOS EM 24/01/78

PAGINA 1

ARQUIVOS DE DADOS

IDENTIFICACAO	TAMANHO REGISTRO	DIMENSAO ARQUIVO	NUM INICIAL REGISTROS	NUMERO INSERCOES	NUMERO DELECOES	NUM ELOS PERMANENT	NUM INVERSOES PERMANENTES
EMP	90	50	24	1	1	2	1
PROJETO	30	10	10	0	0	1	0
TAREFA	25	20	5	0	0	2	0
HABILIT	40	15	14	0	0	2	0
PROJES	70	30	22	0	0	1	1

QUADRO 15

Cont.

QUADRO 15

Cont.

ARQUIVO DE ELDS

IDENTIFICACAO	ARQUIVOS ENVOLVIDOS PRINCIPAL SECUNDARIO	DIMENSAO ARQUIVO	NUMERO INSERCOES	NUMERO DELECOES
MNG	PROJETS EMP	5	20	4
REQ	TAREFA HABILIT	13	37	0
CAP	EMP HABILIT	83	199	7
ASN	PROJPES TAREFA	25	81	0

ARQUIVOS DE INVERSOES

IDENTIFICACAO	ARQUIVO BASE	NUMERO DOS CAMPOS CLASSIFICADOS	NUM INICIAL INVERSOES	NUMERO INVERSOES	NUMERO DELECOES
EMPINV	EMP	2	48	2	0
PRJINV	PROJES	3	38	0	0

QUADRO 15

Cont.

```

//COPIA: JOB (TRP,CPD,,10),RQP,MSQL=VOL=(1,1),CLASS=A
//ST1 EXEC PGM=PL1,FCLG,PAPH=PL1L=CA,X),REGION=PL1L=128K,
// PARM,LKED='SIZE=200000',TIME,CO=15,REGION,GO=100K
//PL1L.SYSIN DD *
COPIA: PROC OPTIONS (MAIN);
/* BACK UP DA DATA BASE
DCL COPIDB ENTRY RETURNS(BIN FIXED);
DCL COD BIN FIXED INIT (0);
PUT SKIP (1) EDIT ('XXX TESTE DO COPIDB XXX') (A);
COD = COPIDB;
PUT SKIP (1) DATA (COD);
END COPIA;
//LKED,DBM0 DD DSN=DBM0,DISP=OLD
//LKED,SYSIN DD *
INCLUDE DBM0(COPIDB)
//GO.ARODP DD DSN=ARQP,DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQEP DD DSN=ARQEP,DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQIP DD DSN=ARQIP,DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQSP DD DSN=ARQSP,DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.AROB DD DSN=BACKUP.HYADES,DISP=(NEW,KEEP),
// UNIT=SYSDA,VOL=SER=CK1002,LABEL=RETPD=5,
// SPACE=(TRK,(80)),
// DCB=(RECFM=VB,LAECL=516,BLKSIZE=1036)
//

```

```

//RESTAURA JOB (TRP,CPD,,10),POB,MSGLEVEL=(1,1),CLASS=A
//ST1 EXEC PL11FC1G,PARM,PL11=(A,X),REGION,PL11=128K,
// PARM.LKED='SIZE=200000',TIME,GM=15,REGION.GM=100K
//PL11.SYSIN DD *
RESTAURA; PROC OPTIONS (MAIN);
/* RESTAURA DATA BASE
DCL RESTDB ENTRY RETURNS(BIN FIXED);
DCL COD BIN FIXED INIT (0);
PUT SKIP (1) EDIT ('XXX TESTE DO RESTDB XXX') (A);
COD = RESTDB;
PUT SKIP (1) DATA (COD);
END RESTAURA;
//LKED.DBMO DD DSN=CBMO,DISP=OLD
//LKED.SYSIN DD *
INCLUDE DBMO(RESTDB)
//GO.ARQEP DD DSN=ARQEP,DISP=(MOD,KEEP),DCB=DSCRG=IS
//GO.ARQEP DD DSN=ARQEP,DISP=(MOD,KEEP),DCB=DSCRG=IS
//GO.ARQIP DD DSN=ARQIP,DISP=(MOD,KEEP),DCB=DSCRG=IS
//GO.ARQSP DD DSN=ARQSP,DISP=(MOD,KEEP),DCB=DSCRG=IS
//GO.ARQB DD DSN=BACKUP.HYADES,DISP=OLD,UNIT=SYS0A,VOL=SER=DK1002
//

```

REFERÊNCIAS BIBLIOGRÁFICAS

- MARTIN, J.T., - Computer Database Organization - Prentice Hall - 1975
- DATE, C.J.; - An Introduction To Database Systems Addison Wesley - 1975
- KATZAN, H. - Systems Design And Documentation- Computer Science Series - 1976
- WALSH, D.A.. - A Guide For Software Documentation - Mc Graw Hill - 1969
- FURTADO, A.L., PASSOS, S. & VASQUES, R. - Um suporte para Implementação de Gerência de Banco de Dados; PUC/RJ - 1977
- LONDON, K. - Documentation Standards - Petrocelli Books - 1974
- FURTADO, A. L. - Uma introdução a Banco de Dados - PUC/RJ 1976
- FURTADO, A.L. - Towards the design of Database interfaces for non - Programmers - PUC/RJ - 1976
- CHALLIS, M.F. - The Jackdaw Database Package - University of Cambridge - computer laboratory - 1974
- STONEBRAKER, M. - The design and implementation of ingres - ACM transactions on Database systems - vol. 1 num.3 - 1976
- FARLEY, J - Query execution and index selection for relational Databases - technical report CSRG. - 53 - 1975.

- THE ANSI/SPARE DEMS MODEL - D.A. JARDINE - NORTH HOLLAND
1977
- TSICHRITZIS, D.C. & LOCHOVSKY, F.H. - Database management
systems - Academic Press - 1977,