

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

ESTÁGIO SUPERVISIONADO  
IMPLANTAÇÃO DO SISTEMA DE BANCO DE  
DADOS HYADES

RELATÓRIO FINAL

ESTAGIÁRIA: JOSELUCE DE FARIAS CUNHA  
ORIENTADORES: LIKISO HATTORI  
HAROLDO CATUNDA

Campina Grande, 10/07/79

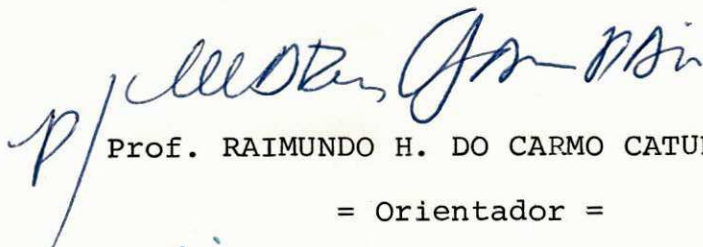


SERVIÇO PÚBLICO FEDERAL  
Ministério da Educação e Cultura  
Universidade Federal da Paraíba  
Centro de Ciências e Tecnologia  
Departamento de Sistemas e Computação  
Av. Aprígio Veloso, 882 - Cx. Postal 518  
Campina Grande - Paraíba

D E C L A R A Ç Ã O

Declaramos para os devidos fins e efeitos, que a aluna JOSELUCE DE FARIAS CUNHA, do Curso de Formação de Tecnólogos em Processamento de Dados, concluiu estágio neste Departamento, desenvolvendo um trabalho de "Im<sup>plantação</sup> do Sistema de Banco de Dados HYADES", duran<sup>te</sup> o período de 15.01.79 a 15.06.79, sob minha orienta<sup>ção</sup>.

Campina Grande, 16 de julho de 1979.

  
Prof. RAIMUNDO H. DO CARMO CATUNDA  
= Orientador =

Visto:

  
Prof. ERATÓSTENES E.R. DE ARAÚJO

= Chefe do DSC =



Biblioteca Setorial do CDSA. Março de 2021.

Sumé - PB

## 1. APRESENTAÇÃO

Este relatório destina-se a coordenação do Curso de Formação de Tecnólogos em Processamento de Dados. Descrevendo as atividades exercidas durante o Estágio Supervisionado de JOSELUCE DE FARIAS CUNHA, realizado no período de 15 de janeiro à 15 de junho de 1979 no Departamento de Sistemas e Computação da Universidade Federal da Paraíba, Campus de Campina Grande.

O Estágio foi orientado pelos professores LIKISO HATTORI e HAROLDO CATUNDA.



## 2. INTRODUÇÃO

O Estágio teve como objetivo a implantação do Sistema de Bancos de Dados HYADES no nosso computador IBM-370 modelo 145, OS/VSl. Este sistema foi desenvolvido na PUC/Rj durante o ano de 77 e início de 78 como Tese de Mestrado "Desenvolvimento, Implementação, Descrição Lógica e Documentação de um Sistema de Banco de Dados" por Roberto Pires Vasques, tendo como orientador Antonio Luz Furtado.

Para esta implantação foi preciso um aprendizado ã parte envolvendo Job Control Language (incluindo procedures catalogadas), Programas Utilitários, noções sobre o OS/VSl e Linguagem de Programação PL/1, e, o estudo do funcionamento de HYADES.

## ÍNDICE

1. Apresentação
2. Introdução
3. Descrição Detalhada das Atividades Preliminares
  - 3.1 - Estudo de Job Control Language (JCL)
    - 3.1.1 - Declaração Job
    - 3.1.2 - Declaração EXEC
    - 3.1.3 - Declaração DD
    - 3.1.4 - Procedures Catalogadas
  - 3.2 - Estudo de Programas Utilitários
  - 3.3 - Noções sobre o OS/VSl.
  - 3.4 - Estudo de Programming Language ONE (PL/1)
4. Idéia Geral do Sistema de Banco de Dados HYADES
  - 4.1 - Concepção do Sistema
  - 4.2 - Arquivos
  - 4.3 - Primitivos e Subrotinas
  - 4.4 - Fluxo de Execução do Sistema
  - 4.5 - Manuseio do Sistema
  - 4.6 - Necessidade de Hardware e Software
5. Implantação do Sistema

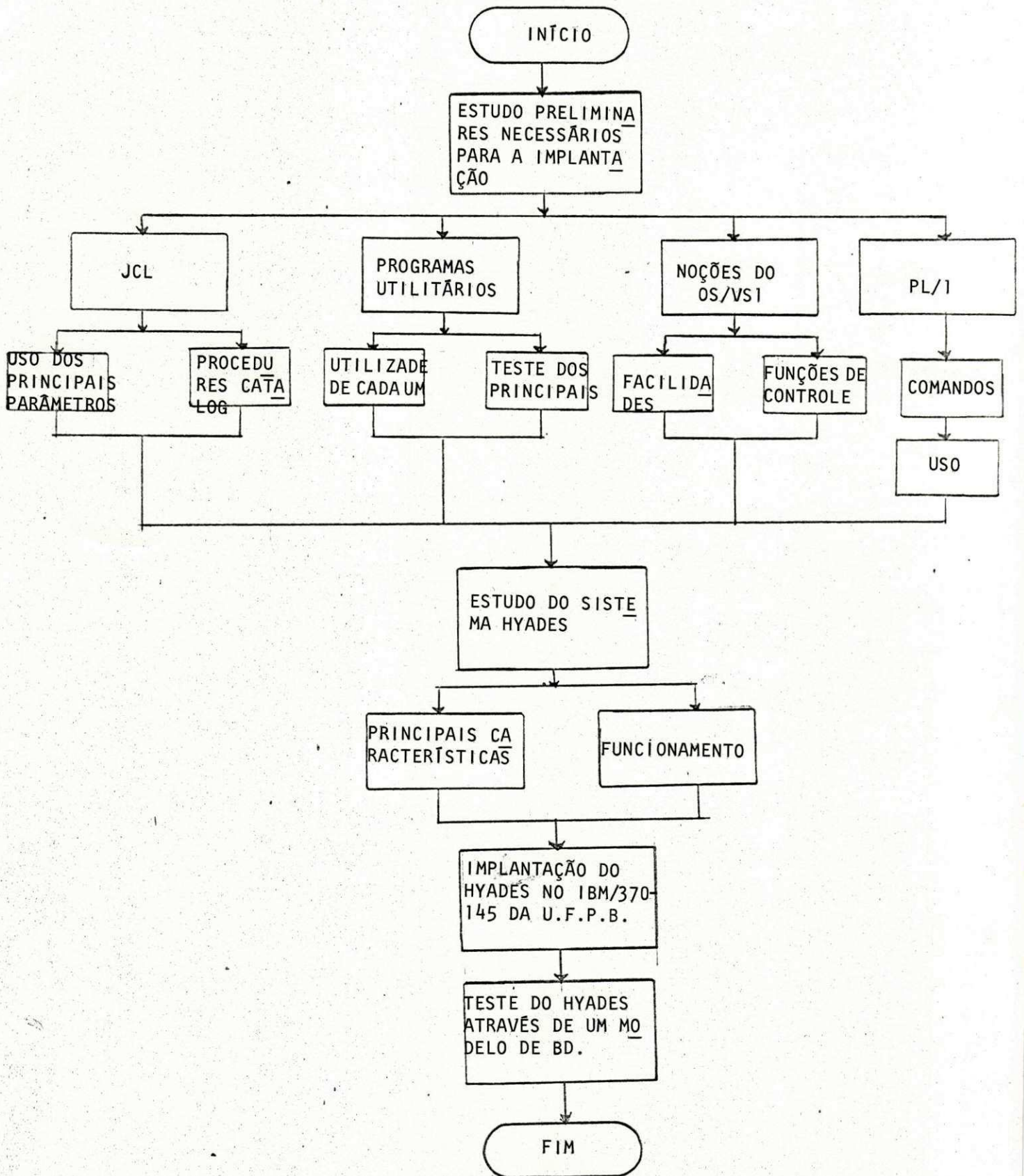
6. Conclusão

7. Considerações Gerais

8. Bibliografia

9. Anexos

Diagrama das Atividades do Estágio





### 3. DESCRIÇÃO DETALHADA DAS ATIVIDADES PRELIMINARES

Durante o estágio foi necessário um estudo preliminar referente a assuntos de importância fundamental para a implantação do Sistema de Banco de Dados HYADES. Estes assuntos são os estudos descritos a seguir.

#### 3.1 - Estudo de Job Control Language (JCL)

Nesta fase foram estudados os principais parâmetros da Linguagem de Controle de Jobs. E, noções de Procedures Catalogadas.

#### Sumário dos Parametros Estudados

##### 3.1.1 - Declaração JOB

Indica o início de um JOB e, quando os Jobs encontram-se empilhados num input stream, indica o fim de controle do JOB anterior.

Forma Geral.

```
// Jobnome JOB Operandos Comentários
```

O Jobnome é a identificação do JOB.

Os operandos são:

- Os parametros posicionais - indicam informações de contabilização e o nome do programador.

- Os parametros de palavras chave - são os seguintes:

ADDRSPC - usado para indicar que um programa tem seu endereço virtual igual ao real;

CLASS - atribui uma classe ao JOB dependendo das características do mesmo;

COND - permite que o JOB seja processado até atingir determinada condição, a qual é testada usando o return code de um step;

MSGCLASS - especifica a classe de saída das mensagens do sistema;

MSGLEVEL - especifica que parte da saída do JOB referente a declarações e mensagens deve sair na listagem;

PRTY - especifica a prioridade do JOB dentro de uma classe;

- RD - especifica que a facilidade de restart de step são usadas para suprimir o restart automático;
- REGION - determina a quantidade de memória real que deve ser alocada para um JOB;
- RESTART - especifica que o JOB será reexecutado através das facilidades de restart;
- TIME - limita o tempo máximo que um JOB pode usar a CPU;
- TYPRUN - especifica se um JOB é para ficar retido na fila de entrada até que um evento ocorra; se é para ser apenas verificado a sintaxe das declarações de JCL ou para ficar normalmente nas fitas.

### 3.1.2 - Declaração EXEC

O statement EXEC é o primeiro de cada JOB step e de um step numa procedure catalogada.

Forma Geral.

```
.....  
| // Stepnome EXEC Operandos Comentários |
```



Os operandos tem como parametros posicionais:

PGM - especifica um programa a ser executado, o qual deve ser membro de uma biblioteca do sistema ou privada;

PROC - especifica que uma procedure catalogada ou in-stream é para ser executada.

Os parametros de palavra chave dos operandos são:

ACCT - especifica um ou mais sub- parametros de informações de contabilização para ser passado para as rotinas de contabilização da instalação;

COND - testa através do completion code de um ou mais steps anteriores se um JOB deve ou não ser executado;

Os parametros ADDRSPC, RD, REGION, PARM, TIME, etc., também podem ser usados na declaração EXEC.

### 3.1.3 - Declaração DD

Data Definition descreve um data set a ser usado num JOB step e especifica as facilidades de entrada e saída requeridos pelo data set.

Tem como parametros posicionais:

\* - o asterisco usa como delimitador o //  
ou /\*;

DATA - sua função é semelhante a do \*, sendo que este permite o uso de dados com // nas colunas 1 e 2, o delimitador será o /\* ou outro especificado pelo usuário;

DUMMY - especifica que: não é para alocar memória para o data set, nenhuma disposição de processamento é para ser executada no data set;

Tem como principais parametros de palavra chave:

AFF - permite a separação de canal para um JOB step;

COPIES - permite a impressão de mais de uma cópia do data set de saída;

DCB - utilizado para completar as informações no DCB (Data Control Block) acerca de um data set no tempo de execução através dos subparametros: BLKSIZE, BUFNO, LRECL, DSORG, RECFM, BUFL, DEN;

- DISP - especifica o status do data set ao sistema e o que deve ser feito com o data set depois do fim do JOB set ou do processamento do JOB;
- DSNAME - atribui um nome ao data set;
- LABEL - descreve o label associado com o data set;
- OUTLIM - especifica o limite de registros lógicos a serem incluídos no data set de saída;
- SYSOUT - especifica a classe de saída de um data set;
- UNIT - especifica o tipo e a quantidade de dispositivos atribuídos a um data set;
- VOLUME - identifica o volume que o data set está ou será armazenado;
- PRIVATE - indica que nenhum data set de saída pode ser alocado neste volume, sem que seja especificamente requisitado;
- RETAIN - indica que o volume não é para ser desmontado após o step.

#### 3.1.4 - Procedures Catalogadas

Uma Procedure Catalogada é um conjunto de de clarações de JCL colocadas como um membro de um data set parti cionado num dispositivo de acesso direto. As procedures cata logadas são mantidas na biblioteca do sistema, denominada SYS1. PROCLIB. Dependendo do sistema é possível que outras bibliote cas sejam concatenadas a SYS1.PROCLIB.

#### 3.2 - Estudo de Programas Utilitários

Nesta parte foram estudados os principais pro gramas utilitários e executados alguns testes, isto incluiu também a prática de JCL. (Vide Anexo Biblioteca de Programas).

Alguns dos utilitários estudados foram:

IEBGENER - pode copiar um data set sequencial ou um membro de data set particio nado de um data set sequencial ou de um membro de um data set parti cionado usado como input. O progra ma pode expandir um data set parti cionado existente, criando membros particionados e intercalando-os no data set que deve ser expandido;

IEBCOPY - copia, comprime ou intercala data sets particionados; seleciona ou exclui membros específicos em uma operação de cópia por step; renomeia e/ou substitui membros de um data set particionado.

IEBUPDTE - usado para criar e atualizar uma biblioteca simbólica; incorporar troca de membros de um data set sequencial ou particionado; trocar a organização de um data set sequencial para particionado ou vice versa;

IEBUPDAT - semelhante ao IEBUPDTE, porém só se aplica a data sets particionados;

IEBPTPCH - imprime ou perfura todo, ou partes selecionadas de um data set sequencial ou particionado;

IEHMOVE - move ou copia conjuntos lógicos de dados;

IEHLIST - lista os dados de controle do sistema que são: entradas em um catálogo; entradas num diretório de data sets particionados e VTOC.

- IEHPROGM - constroi e mantém dados de controle do sistema. Pode ser usado para:
- "scratch" - em um data set ou um membro.
  - "renome" - em um data set ou um membro.
  - catalogar ou descarregar um data set.
  - construir ou deletar um índice de apelido.
  - conectar ou liberar dois volumes.
  - construir e manter um "generation data group index".

### 3.3 - NOÇÕES SOBRE O OS/VS1

A definição mais geral de Sistema Operacional é a seguinte "O Sistema Operacional é uma coleção de programas escritos e supridos pelos fabricantes para uma instalação de Processamento de Dados usar a fim de automatizar o processamento".

Alguns destes programas são:

- Programas para manipular as operações de entrada/saída para todos os tipos de organização de data sets;

- Programa para manter as trilhas com espaço disponível nos dispositivos de acesso direto;
- Programas para diagnosticar e analisar os dispositivos de entrada/saída de erros de operação;
- Um Programa "MESTRE", que, quando na memória é chamado de SUPERVISOR. (O supervisor será detalhada mais adiante);
- Programa para manter as trilhas de localização sobre o disco de todos os programas de aplicação e os citados acima;
- Programas que escrevem mensagens para assessorar a operação do computador quando for necessário ou quando alguma coisa está errada;
- Programa para manter catálogo de data sets, notificando o número do volume que o data set está arquivado.

O SUPERVISOR é um programa que controla a CPU para que esta seja utilizada por programas concorrentes.

O SUPERVISOR controla a CPU no início do processamento do dia. Assim como após o início do processamento ele controla os programas que entram na CPU, informando a cada



um dos programas de aplicação que pode utilizar a CPU. Outra função do SUPERVISOR é o controle de entrada e saída.

### Facilidades do OS/VSl

O Sistema Operacional OS/VSl trabalha com organização de memória virtual. A memória possui duas áreas distintas.

- Área não paginável - corresponde a memória central quando mapeada sobre virtual = Real;
- Área paginável - corresponde a área alocada num dispositivo de acesso direto, que funciona na modalidade de memória virtual, e poderá assumir dimensões acima da capacidade real.

Quando a prioridade está classificada em duas:

- Prioridade de Seleção - é a que é ditada pela classe;
- Prioridade de Execução - é a prioridade existente dentro da classe.

O usuário poderá ter acesso aos programas das diversas bibliotecas, tanto para incorporar novos programas como para utilizar os já existentes.

Permite ainda a manipulação de arquivos em fita magnética, em disco e em cartão. Os arquivos podem ser permanentes, temporários com data de expiração, com password, ou com diversas gerações com um mesmo nome.

### Funções Gerais de Controle

- FUNÇÕES DE ADMINISTRAÇÃO DE JOB.
    - . analisa o input stream
    - . aloca os dispositivos de entrada e saída
    - . seleciona jobs para execução
    - . provê comunicações entre o operador e o sistema.
  
  - ADMINISTRAÇÃO DE TASK
- Executa algumas funções de:
- . supervisão de task
  - . fetch de programa
  - . supervisão de conteúdos
  - . supervisão de timers
  - . supervisão de entrada e saída
  - . supervisão de overlay
  - . manipulação de condições de execução.

- ADMINISTRAÇÃO DE PÁGINAS

- . adcionamento de novas rotinas funcionais
- . composto de:
  - supervisor de memória real
  - supervisor de página de memória externa.
- . execução de vai e vem de:
  - páginas
  - manutenção de performance do sistema
  - suporte de memória virtual = real.
- . páginas de data set (SYS1.PAGE).

- ADMINISTRAÇÃO DE DADOS

- . supervisor de entrada e saída
- . métodos de acesso
- . administração de catálogos
- . administração de dados de acesso direto
- . open/close/OEV.

- ADMINISTRAÇÃO DE RECUPERAÇÃO (RECOVERY)

- . função principal é a recuperação e documentação dos erros do sistema
- . recuperação:
  - recuperação de função
  - recuperação de sistema
  - restart do suporte do sistema
  - reparo do sistema.
- . as informações são escritas na SYS1.LOGREC e impressas por utilitários.

### 3.4 - ESTUDO DE PROGRAMMING LANGUAGE ONE (PL/1)

Esta aprendizagem foi necessária porque a DML (linguagem de manipulação dos dados) do sistema HYADES é implementada por meio de subrotinas escritas em PL/1. E para manipularmos o sistema este é restrito a uma única linguagem hospedeira que é o PL/1, esta prevê expressões aritméticas poderosas com funções embutidas, o que é útil para cálculos estatísticos sobre os dados.

Esta linguagem apresenta varias facilidades, além de ser uma linguagem de formato livre, oferece a possibilidade de trabalhar-se com bit e com strings, e facilita a economia de memória, pois, a alocação das variáveis podem ocorrer estaticamente, ou seja, antes da execução do bloco, ou dinamicamente, durante a execução do bloco a que os identificadores pertencem.

O PL/1 é uma linguagem superpoderosa, daí sua complexidade como um todo; muitas vezes a dificuldade reside na simples escolha do recurso adequado para se programar uma determinada função, haja visto o grande número de recursos existentes.

Para aprendizagem desta linguagem foi feito um estudo teórico dos principais comandos e a execução de programas simples com a finalidade de por em prática estes comandos.

#### 4. IDÉIA GERAL DO SISTEMA DE BANCO DE DADOS HYADES

Este estudo foi feito para se conhecer as principais características e o funcionamento do sistema a ser implantado, o HYADES.

##### 4.1 - CONCEPÇÃO DO SISTEMA

O suporte foi desenvolvido como implementação de um projeto envolvendo Banco de Dados. Inclui as necessidades básicas para uma implementação seja qual for o modelo adotado: hierárquico, relacional e ou rede.

Conta com três naturezas fundamentais de arquivos denominados: arquivos de dados, arquivos de elos, e arquivos de inversões que são disponíveis para o usuário, e com o arquivo de estrutura contendo o dicionário de dados que é o arquivo suporte do sistema. O método de acesso utilizado é o sequencial indexado.

A cada arquivo do usuário é tido como um segmento dos arquivos do sistema; Então, quando nos referimos a segmentos são os arquivos dos usuários, e quando falamos em arquivos são os do sistema.

O sistema permite que os arquivos sejam criados de dois tipos: permanentes, que são a base do sistema, e

temporários, que realizam as operações transitórias durante o processamento sendo depois liberados.

#### 4.2 - ARQUIVOS

Os segmentos de dados dos arquivos de dados contém as informações dos usuários. Os registros são identificados por: um sequencial gerado internamente denominado identificador. (vê Figura - 1).

Os segmentos de elos são constituídos de registros que representam ligações entre dois arquivos de dados, podendo estas ligações ser da ordem 1:n ou m:n. (vê Figura - 2).

Os arquivos de inversões tem seus segmentos constituídos de registros contendo ligações entre registros de um mesmo segmento de dados baseado em um conjunto de tipos de itens previamente classificados. (vê Figura - 3).

Os segmentos dos arquivos de estruturas, contém o dicionário de dados dos segmentos de dados criados.

#### 4.3 - PRIMITIVOS E SUBROTINAS

O sistema para atender as funções solicitadas pelo usuário, dispõe de vinte e sete módulos de programas, sendo vinte e quatro consideradas primitivos por operarem independentes, e três denominados subrotinas auxiliares utilizadas pe

FIGURA 1 - GRAFICO DA FORMAÇÃO DOS ARQUIVOS DE DADOS

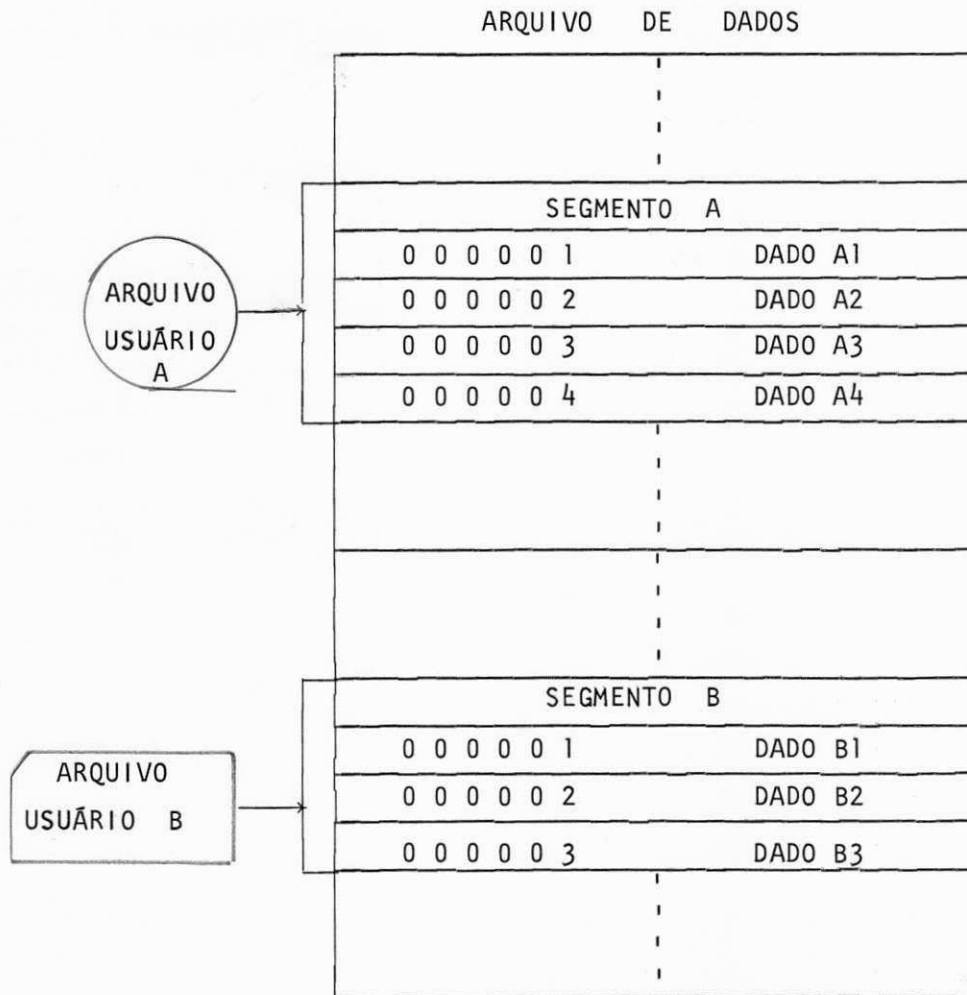




FIGURA 2 - GRAFICO DA FORMAÇÃO DOS ARQUIVOS DE ELOS

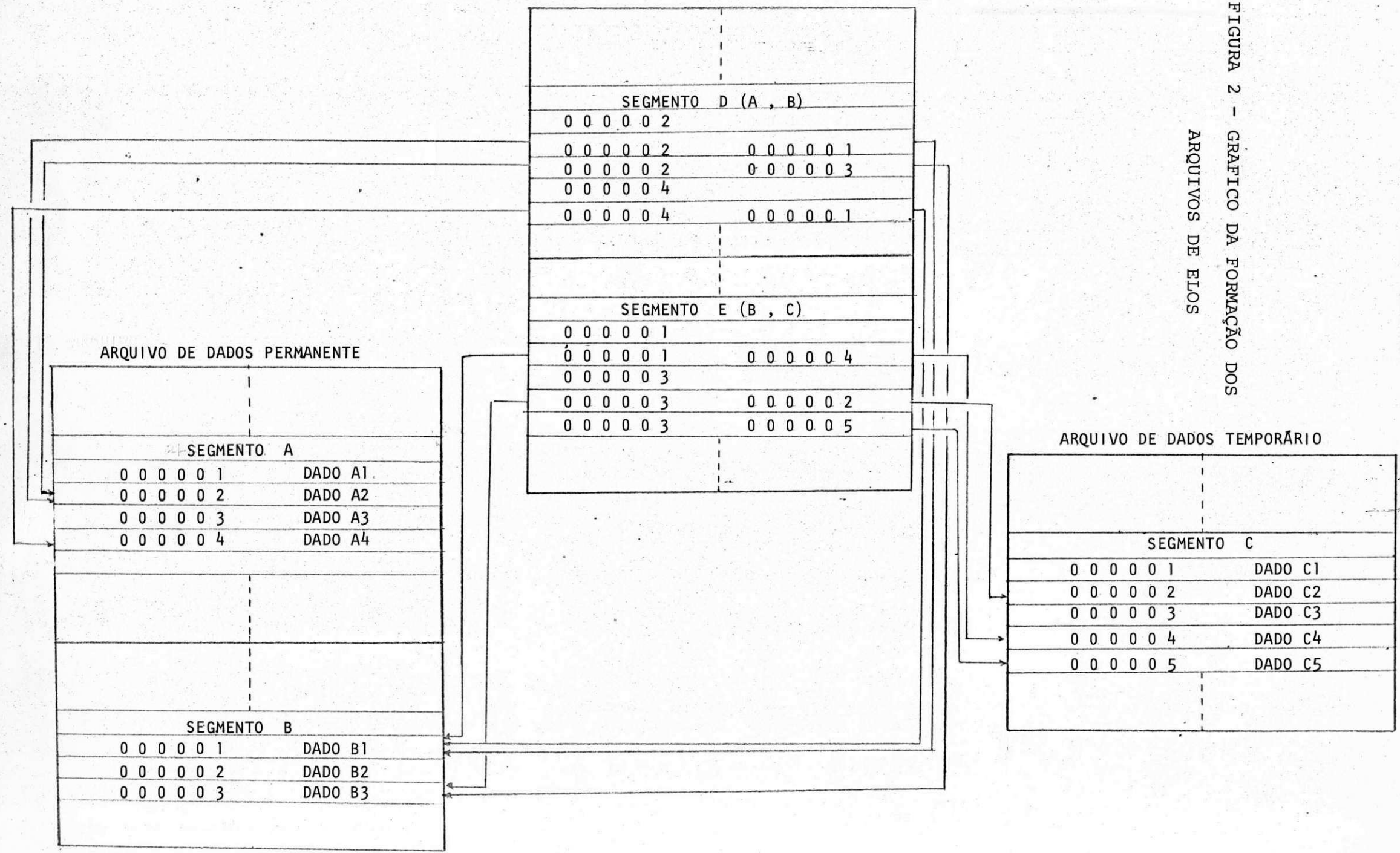
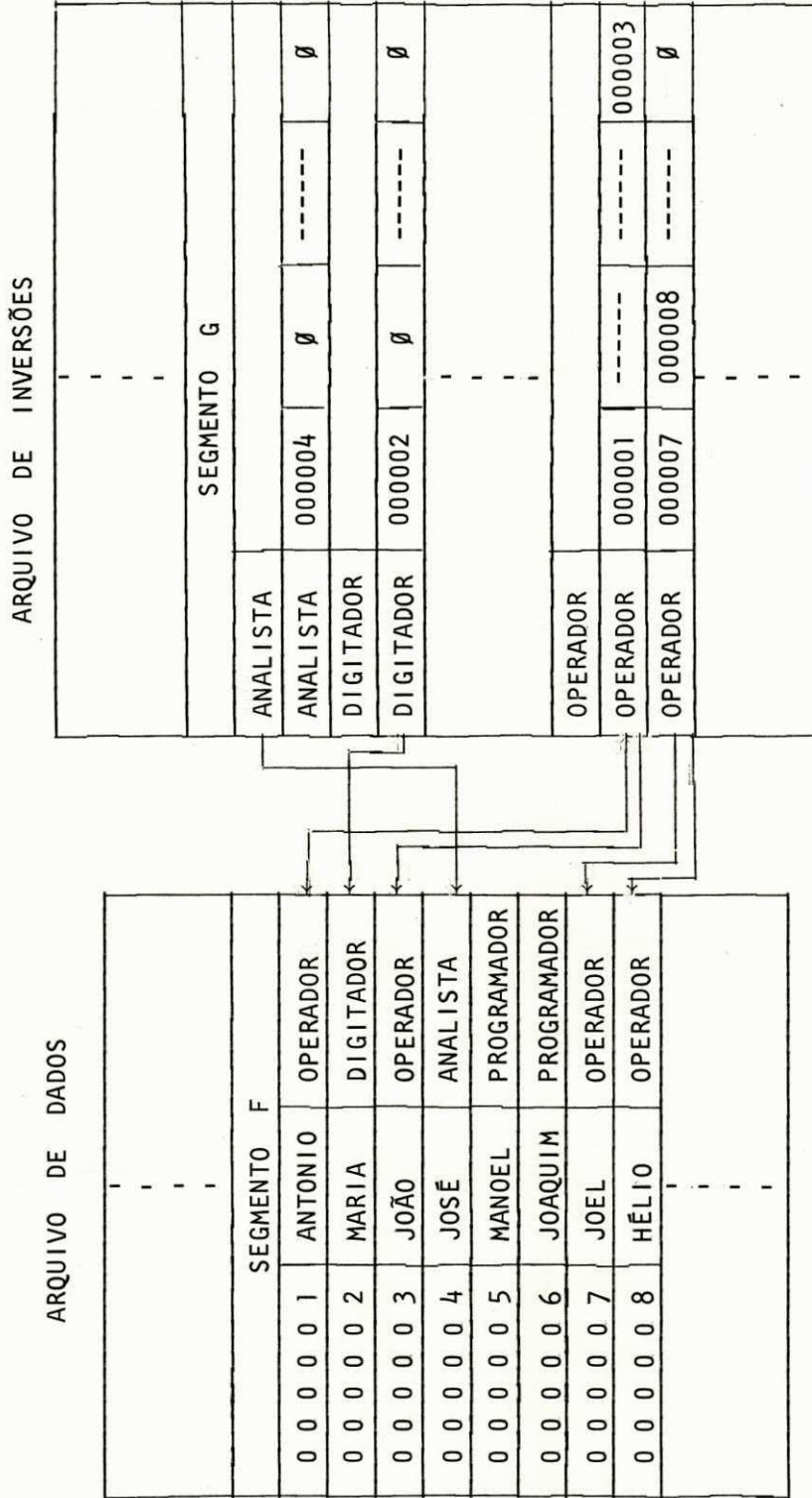


FIGURA 3 - GRAFICO DA FORMAÇÃO DOS ARQUIVOS DE INVERSÃO



la maioria dos primitivos, estes são ativados pelos usuários para realização de determinada tarefa contida em sua estrutura.

Os primitivos estão classificados de acordo com sua utilização em:

. PRIMITIVOS PARA MANUSEIO DE SEGMENTO:

- CRIDAD - cria segmento de dados;
- CRIELO - cria segmento de elos;
- CRIINV - cria segmento de inversões;
- CRICLS - cria segmento de dados classifi  
cados;
- CRIPROJ - cria segmento de dados a partir  
de segmento de inversão. Cria  
uma projeção;
- CRIDUP - duplica um segmento de dados;
- CANDAD - cancela segmento de dados;
- CANELO - cancela segmento de elos;
- CANINV - cancela segmento de inversão;
- INFARQ - fornece informações sobre deter  
minado segmento;
- TRIARQ - troca identificação externa de  
segmento;

. PRIMITIVOS PARA MANUSEIO DE REGISTROS:

- SELDAD - seleção de registro de dados;
- SELELO - seleção de registro de elos;
- SELINV - seleção de registro de inversão;
- INSDAD - inserção de registro de dados;
- INSELO - inserção de registro de elos;
- REMDAD - remoção de registro de dados;
- MODDAD - modifica valor em registros de dados;

. PRIMITIVOS DE APOIO:

- ATIVDB - ativa o sistema tornando disponível o Banco de Dados;
- DATVDB - desativa o sistema;
- COPIDB - rearrumação e cópia dos arquivos de Banco de Dados, obtendo um conjunto de dados sequenciais. Imagem do Banco de Dados;
- RESTDB - restaura os arquivos do Banco de Dados;
- RESMDB - emite relatório resumo sobre o Banco de Dados.



As subrotinas auxiliares que fazem parte do sistema são:

- AUX1 - verifica a situação de um segmento no Banco de Dados com vistas a inclusão, exclusão e consulta;
- AUX2 - atualiza a situação de um elemento com vistas a inclusão ou exclusão do mesmo no sistema;
- AUX3 - manuseio de abertura e fechamento dos arquivos do sistema.

Acopladas ao sistema por necessidades de manter informações durante o processamento dos primitivos, existem cinco módulos do sistema de avaliação de eficiência de:

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

nuseio de arquivos.

#### 4.4 - FLUXO DE EXECUÇÃO DO SISTEMA

O suporte foi desenvolvido de modo que o Banco de Dados possa ser manuseado em uma só execução e em mais de uma execuções, podendo ocorrer a eliminação e restauração do Banco de Dados entre estas execuções.

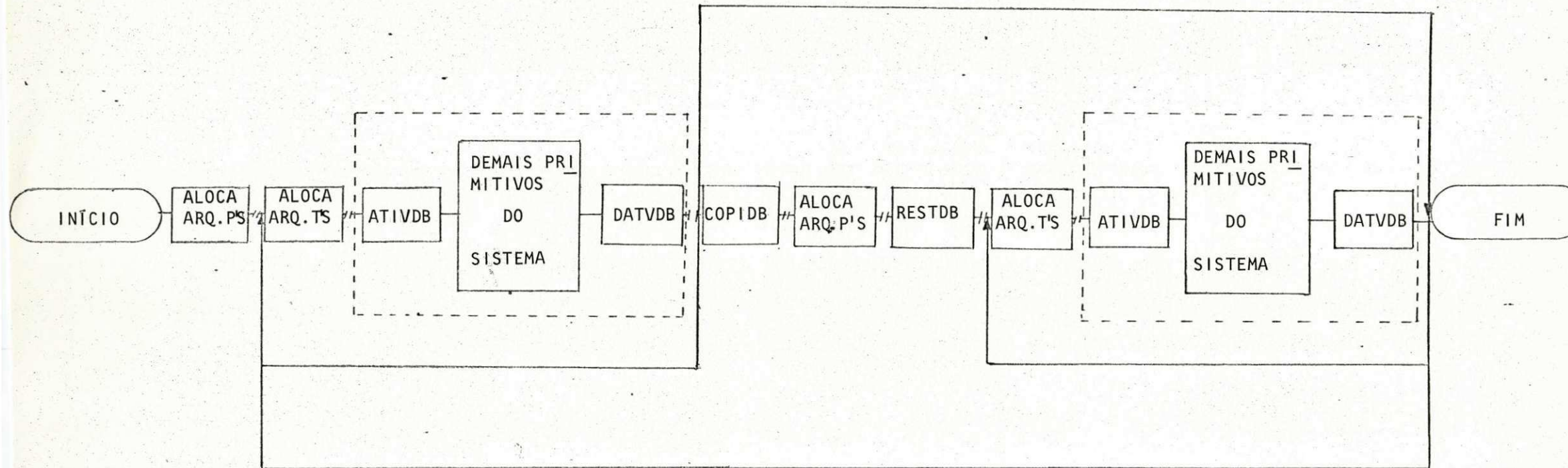
A Figura - 4 indica o fluxo operacional do sistema em todas as suas fazes. Notamos que a primeira etapa é a alocação dos dados permanentes do suporte, referentes aos dados, elos, inversões e estruturas. Esta etapa só deve ser utilizada através da primeira execução do suporte e antes da restauração do Banco de Dados.

A etapa seguinte aloca conjunto de dados temporários deixando-os disponíveis ao suporte. É utilizado antes da execução ou reexecução do suporte.

Os primitivos de apoio COPIDB e RESTDB executam suas tarefas independente dos outros primitivos.

Os primitivos ATIVDB e DATVDB sempre iniciam e terminam uma execução do suporte e entre a execução do suporte e entre a execução dos dois são executados tantos primitivos quanto os necessários.

FIGURA 4 - FLUXO DE EXECUÇÃO DO SISTEMA





#### 4.5 - MANUSEIO DO SISTEMA

O sistema permite que vários usuários executem ao mesmo tempo operações de seleção, embora, para operações de atualização (inserção, alteração ou remoção) fique o Banco do Dados reservado só para um usuário.

#### 4.6 - NECESSIDADES DE HARDWARE E SOFTWARE

O suporte foi desenvolvido para uso em equipamentos IBM/370 modelo 145 da PUC. Podendo 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.

Os arquivos do sistema podem ser armazenados em disco IBM 2314, IBM 3330 ou outros compatíveis com a organização utilizada.

O sistema é operável no Sistema Operacional OS, podendo ser OS/MVT, OS/VS1, OS/VS2.

O suporte foi desenvolvido na linguagem PL/I-F, utiliza em seus arquivos a organização sequencial indexada.

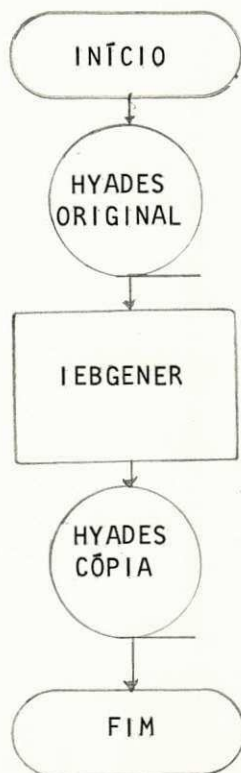
É necessário que o Sistema Operacional dis  
ponha do programa utilitário SORT compatível ao compilador  
PL/1-F.

## 5. IMPLANTAÇÃO DO SISTEMA

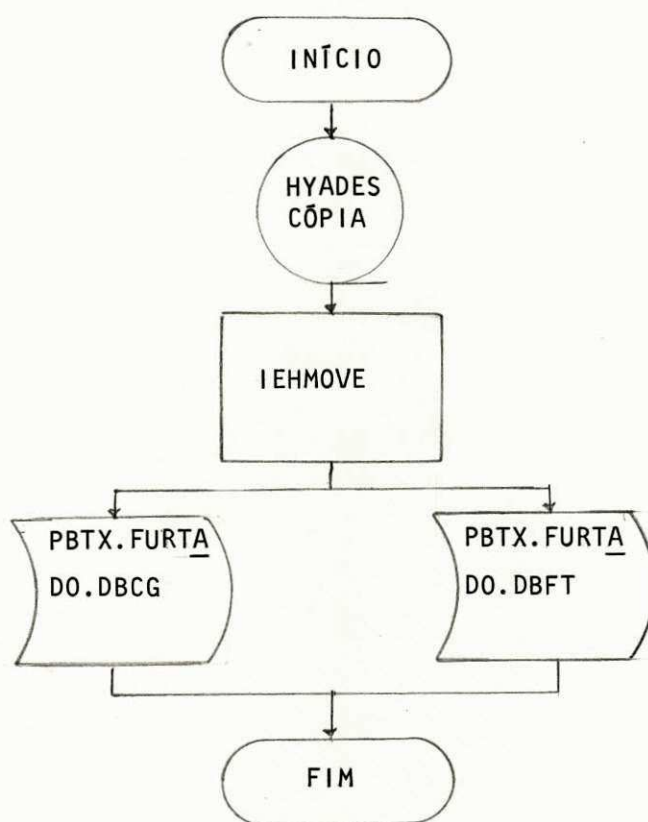
Durante a implantação do sistema executamos as seguintes atividades:

- . Foi feita uma cópia da fita que recebemos da PUC contendo todos os arquivos, os primitivos e as subrotinas do sistema de Banco de Dados HYADES, para obtermos segurança das informações, evitando assim que no caso de danos na fita original houvesse perda de informações. E também porque a fita não era nossa, veio na condição de ser devolvida a PUC logo que não fosse mais necessária para nosso uso.

Para isto utilizamos o programa utilitário IEBGENER.



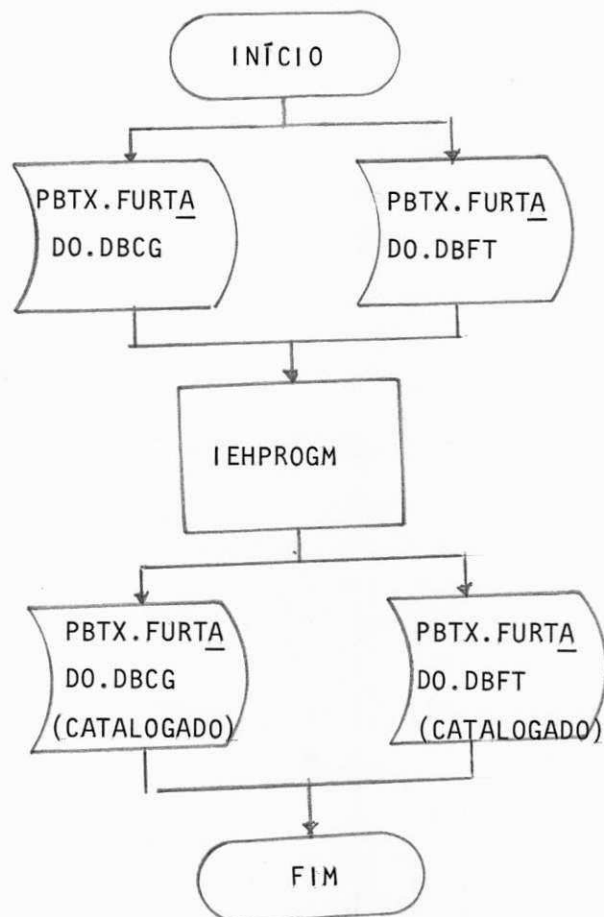
. O segundo passo foi descarregar a cópia da fita no disco UFPVS2 através do programa utilitário IEHMOVE. Escolhemos este pela facilidade que o IEHMOVE oferece pois, o próprio programa é que aloca o espaço necessário para o data set e automaticamente separa seus membros e principalmente pelo o fato da fita original ter sido carregada pelo o IEHMOVE, isto faz com que ela só possa ser descarregada com este.



PBTX.FURTADO.DBCG → é o nome do data set que contém as rotinas referentes ao sistema HYADES.

PBTX.FURTADO.DBFT → é o nome do data set que contém as rotinas para avaliação de eficiência.

- Depois utilizando o utilitário IEHPROGM catalogamos o da ta set para facilitar o acesso ao mesmo.



- Finalmente para testarmos o sistema utilizamos um modelo de Banco de Dados para uso do suporte, apresentado na Te se de Roberto Pires, descrito a seguir. Mas, não chegou a funcionar porque houve problemas de interação do sistema HYADES com o Sistema Operacional OS/VS1. Este erro pode ser verificado nas listagens em anexos.

### Descrição do Modelo:

Este é para uma empresa hipotética 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 habilitação. 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 admissão só pode ser realizada se o empregado não estiver envolvido em nenhum projeto como componente líder.

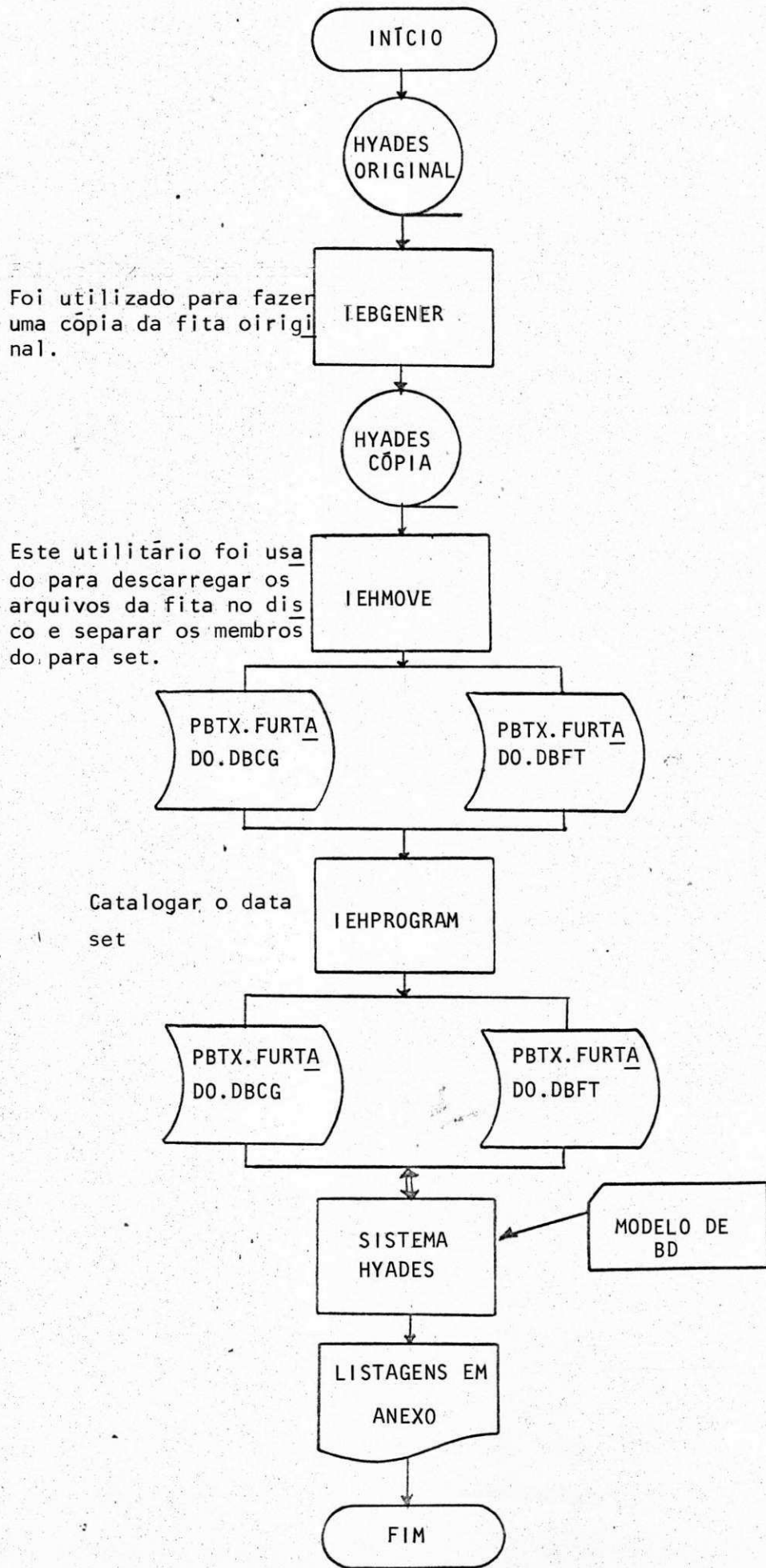
A criação de grupos de projeto, designação de componentes e de líder é tarefa do Gerente de Desenvolvimento de Sistemas.

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

Logo, o suporte deve ser usado de modo que atenda as necessidades de cada Departamento.



DIAGRAMA DA IMPLANTAÇÃO DO SISTEMA



## 6. CONCLUSÃO

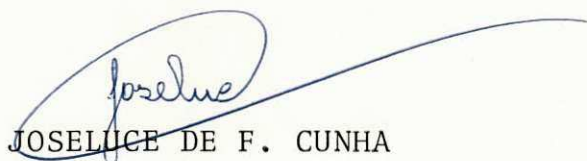
O Estágio foi de grande validade porque tive a oportunidade de ampliar meus conhecimentos com assuntos que não foi possível aprender durante o curso e ver na prática as dificuldades que podem surgir na implantação de um sistema.

No nosso caso, isto é: na implantação do sistema de Banco de Dados HYADES, surgiram muitos problemas, pois, a UFPb está sendo a primeira a implantar este sistema através da Tese de Mestrado "Desenvolvimento, Implementação, Descrição Lógica e Documentação de um sistema de Banco de Dados" de Roberto Pires. Em nosso computador, IBM/370 modelo 145, tentamos fazer funcionar no sistema HYADES um modelo de Banco de Dados que funcionou no computador da PUC, IBM/370 modelo 165 e ainda não conseguimos, depois de algumas tentativas, procuramos nos comunicar com a PUC. Para isto foi preciso algumas semanas de espera e quando obtivemos resposta não tinha sido possível solucionarem nosso problema. Logo depois, JOSÉ LEONIDAS, um dos analistas do nosso NPD (Núcleo de Processamento de Dados), superou este erro. Mas, logo surgiram outros na parte de interação do HYADES com o Sistema Operacional OS/VS1.

Outra dificuldade que surgiu, talvez a maior, foi o problema de alocação de espaço permanente em disco, necessário para os módulos do sistema.



Por estes empecílios ainda não foi possível concluir a implantação do HYADES, pois para isto é necessário dedicar uma boa parte de tempo para o sistema, e o orientador responsável por esta parte, Likiso Hattori, não está dispondo de tempo suficiente, pois encontra-se supercarregado com suas funções. Então, em comum acôrdo resolvemos fazer uma pausa no trabalho, para no próximo mês continuarmos, ficando o estágio encerrado neste ponto e eu comprometida de prosseguir o trabalho independente de estágio, pois o mesmo se for bem aproveitado poderá ser bastante útil para a UFPb.



JOSELICE DE F. CUNHA

= ESTAGIÁRIA =



ORIENTADORES: LIKISO HATTORI



HAROLDO CATUNDA

## BIBLIOGRAFIA

### - MANUAIS

- . IBM - OS/VS1 PLANNING AND USER GUIDE
- . IBM - OS/VS1 JCL REFERENCE
- . IBM - SYSTEMS REFERENCE LIBRARY OS UTILITIES

### - APOSTÍLAS

- . INTRODUÇÃO AO JOB CONTROL LANGUAGE (JCL) DO SISTEMA OS/VS1
- . INTRODUÇÃO AO JCL E MANIPULAÇÃO DE ARQUIVOS EM FORTRAN IV

LIKISO HATTORI

### - SISTEM/360 JOB CONTROL LANGUAGE

GARY DEWARD BROWN

### - DISSERTAÇÃO DE MESTRADO

DESENVOLVIMENTO, IMPLEMENTAÇÃO, DESCRIÇÃO LÓGICA E DOCUMENTAÇÃO DE UM SISTEMA DE BANCO DE DADOS.

ROBERTO PIRES VASQUES

### - INTRODUÇÃO A PROGRAMAÇÃO COM PL/1

ANTONIO LUZ FURTADO

### - PROGRAMAÇÃO PL/1

FRANCISCO SÃ

9. ANEXOS



```

//JOSELUCE JOB MSGLEVEL=1,PRTY=4
*** *****
***
***  FORMATO
*** // EXEC HYADES,USUARIO='XXXXXXXX',CDT='20,LT'
***  FUNCAO
***  USO DO SISTEMA HYADES.
***  DELETA E CATALOGA OS ARQUIVOS PERMANENTES E TEMPORARIOS
***  QUE CONSTITUEM O BANCO DE DADOS DO SISTEMA.
***  COMPTIA. LINK E EXECUTA UM PROGRAMA ESCRITO EM PL/I-F
***  PELO USUARIO, COM USO DAS PORTAS PRIMITIVAS.
***  APRESENTA ATRAVES DO PROGRAMA EXAMINE UM RESUMO DOS
***  ARQUIVOS UTILIZADOS PELO USUARIO, COM INFORMACOES RELA-
***  TIVAS A NOMF, ESPACO, FORMATO, VOLUME, USO.
***
***  DATA
***  27.03.79
***
***  RESPONSAVEIS
***  I K I S O . C A T U N D A
***
*** *****
***
//HYADES PROC CDP='20,GT',CCP='20,GT',CDT='20,GT',CCT='20,LT',
// REGCOMP=120K,REGKED=150K,REGGO=160K,PDISCO=UEPVS3,
// JDISCO=UEPVS2,DISPP='SHR',DISPT='MOD,KEEP',
// USUARIO='JOSELUCE,ROBERTO'
//EXAMINEP EXEC PGM=EXAMINE,PARM='&USUARIO'
//* *****
//* RESUMO DOS ARQUIVOS PERMANENTES
//* *****
//SYSPRINT DD SYSOUT=A
//DELETAP EXEC PGM=IEFBR14,COND=(&CCP)
//* *****
//* DELECAO DOS ARQUIVOS PERMANEN
//* *****
//ARQDP DD DSN=&USUARIO,ARQDP(INDEX),
// DISP=(OLD,DELETE),VOL=SER=&PDISCO,UNIT=SYSDA
// DD DSN=&USUARIO,ARQDP(INDEX),
// DISP=(OLD,UNCATLG),VOL=SER=&PDISCO,UNIT=SYSDA
//AROSP DD DSN=&USUARIO,AROSP(INDEX),
// DISP=(OLD,DELETE),VOL=SER=&PDISCO,UNIT=SYSDA
// DD DSN=&USUARIO,AROSP(INDEX),
// DISP=(OLD,UNCATLG),VOL=SER=&PDISCO,UNIT=SYSDA
//ARQEP DD DSN=&USUARIO,ARQEP(INDEX),
// DISP=(OLD,DELETE),VOL=SER=&PDISCO,UNIT=SYSDA
// DD DSN=&USUARIO,ARQEP(INDEX),
// DISP=(OLD,UNCATLG),VOL=SER=&PDISCO,UNIT=SYSDA
//ARQIP DD DSN=&USUARIO,ARQIP(INDEX),
// DISP=(OLD,DELETE),VOL=SER=&PDISCO,UNIT=SYSDA
// DD DSN=&USUARIO,ARQIP(INDEX),
// DISP=(OLD,UNCATLG),VOL=SER=&PDISCO,UNIT=SYSDA
//CATALOGP EXEC PGM=IEFBR14,COND=(&CCP)
//* *****
//* CRIACAO DOS ARQUIVOS PERMANENTES
//* *****
//ARQDP DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=516,
// BLKSTZE=5164,DSORG=IS,RKP=5,KEYLEN=9,OPTCD=LIM,
// NTM=2),DSN=&USUARIO,ARQDP(INDEX),
// DISP=(NEW,CATLG),VOL=SER=&PDISCO

```



IFF287I	VOI SER NOS= IEPVS2.	
IFF287I	AROTT	NOT CATLGD 2
IFF287I	VOI SER NOS= IEPVS2.	
IFF287I	AROTT	NOT CATLGD 2
IFF287I	VOI SER NOS= IEPVS2.	
IFF287I	AROTT	NOT CATLGD 2
IFF287I	VOI SER NOS= IEPVS2.	
IFF287I	AROST	NOT CATLGD 2
IFF287I	VOI SER NOS= IEPVS2.	
IFF287I	AROST	NOT CATLGD 2
IFF287I	VOI SER NOS= IEPVS2.	
IFF287I	AROST	NOT CATLGD 2
IFF287I	VOI SER NOS= IEPVS2.	



```

//          DD  UNIT=SYSDA,SPACE=(CYL,1),DCB=(DSORG=IS,
//          DSN=&USUARIP,..ARQDP(PRIME),
//          DISP=(NEW,KEEP),VOL=SER=&PDISCO
//          DD  UNIT=SYSDA,SPACE=(CYL,1),DCB=(DSORG=IS,
//          DSN=&USUARIP,..ARQDP(OVERFLOW),
//          DISP=(NEW,KEEP),VOL=SER=&PDISCO
//ARQSP      DD  UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=FB,LRECL=43,
//          BKSIZE=860,DSORG=IS,RKP=1,KEYLEN=5,OPTCD=LIM,
//          NTM=2),DSN=&USUARIP,..ARQSP(INDEX),
//          DISP=(NEW,CATLG),VOL=SER=&PDISCO
//          DD  UNIT=SYSDA,SPACE=(CYL,1),DCB=(DSORG=IS,
//          DSN=&USUARIP,..ARQSP(PRIME),
//          DISP=(NEW,KEEP),VOL=SER=&PDISCO
//          DD  UNIT=SYSDA,SPACE=(CYL,1),DCB=(DSORG=IS,
//          DSN=&USUARIP,..ARQSP(OVERFLOW),
//          DISP=(NEW,KEEP),VOL=SER=&PDISCO
//ARQEP      DD  UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=60,
//          BKSIZE=1204,DSORG=IS,RKP=5,KEYLEN=16,OPTCD=LIM,
//          NTM=2),DSN=&USUARIP,..ARQEP(INDEX),
//          DISP=(NEW,CATLG),VOL=SER=&PDISCO
//          DD  UNIT=SYSDA,SPACE=(CYL,1),DCB=(DSORG=IS,
//          DSN=&USUARIP,..ARQEP(PRIME),
//          DISP=(NEW,KEEP),VOL=SER=&PDISCO
//          DD  UNIT=SYSDA,SPACE=(CYL,1),DCB=(DSORG=IS,
//          DSN=&USUARIP,..ARQEP(OVERFLOW),
//          DISP=(NEW,KEEP),VOL=SER=&PDISCO
//ARQIP      DD  UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=FB,LRECL=336,
//          BKSIZE=672,DSORG=IS,RKP=1,KEYLEN=255,OPTCD=LIM,
//          NTM=2),DSN=&USUARIP,..ARQIP(INDEX),
//          DISP=(NEW,CATLG),VOL=SER=&PDISCO
//          DD  UNIT=SYSDA,SPACE=(CYL,1),DCB=(DSORG=IS,
//          DSN=&USUARIP,..ARQIP(PRIME),
//          DISP=(NEW,KEEP),VOL=SER=&PDISCO
//          DD  UNIT=SYSDA,SPACE=(CYL,1),DCB=(DSORG=IS,
//          DSN=&USUARIP,..ARQIP(OVERFLOW),
//          DISP=(NEW,KEEP),VOL=SER=&PDISCO
//DEFSTAT   EXEC   PGM=IEFBR14,COND=(&CCT)
//* *****
//*      D E F I C A O   D O S   A R Q U I V O   T E M P O R A R I O S
//* *****
//ARQDT      DD  DSN=&USUARIP,..ARQDT(INDEX),
//          DISP=(OLD,DELETE),VOL=SER=&TDISCO,UNIT=SYSDA
//          DD  DSN=&USUARIP,..ARQDT(INDEX),
//          DISP=(OLD,UNCATLG),VOL=SER=&TDISCO,UNIT=SYSDA
//ARQST      DD  DSN=&USUARIP,..ARQST(INDEX),
//          DISP=(OLD,DELETE),VOL=SER=&TDISCO,UNIT=SYSDA
//          DD  DSN=&USUARIP,..ARQST(INDEX),
//          DISP=(OLD,UNCATLG),VOL=SER=&TDISCO,UNIT=SYSDA
//ARQFT      DD  DSN=&USUARIP,..ARQFT(INDEX),
//          DISP=(OLD,DELETE),VOL=SER=&TDISCO,UNIT=SYSDA
//          DD  DSN=&USUARIP,..ARQFT(INDEX),
//          DISP=(OLD,UNCATLG),VOL=SER=&TDISCO,UNIT=SYSDA
//ARQIT      DD  DSN=&USUARIP,..ARQIT(INDEX),
//          DISP=(OLD,DELETE),VOL=SER=&TDISCO,UNIT=SYSDA
//          DD  DSN=&USUARIP,..ARQIT(INDEX),
//          DISP=(OLD,UNCATLG),VOL=SER=&TDISCO,UNIT=SYSDA
//CATALOGT  EXEC   PGM=IEFBR14,COND=(&CCT)
//* *****
//*      C R I A C A O   D O S   A R Q U I V O S   T E M P O R A R I O S
//* *****

```



```

//ARQDT DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=516,
//      RLKST7F=5164,DSORG=IS,RKP=5,KEYLEN=9,OPTCD=LIM,
//      NTM=2),DSN=&USUARID,ARQDT(INDEX),
//      DISP=(NEW,CATLG),VOL=SER=&TDISCD
//      DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,
//      DISP=(NEW,KEEP),VOL=SER=&TDISCD
//      DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,
//      DSN=&USUARID,ARQDT(OVERFLOW),
//      DISP=(NEW,KEEP),VOL=SER=&TDISCD
//AROST DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=FB,LRECL=43,
//      RLKST7F=860,DSORG=IS,RKP=1,KEYLEN=5,OPTCD=LIM,
//      NTM=2),DSN=&USUARID,AROST(INDEX),
//      DISP=(NEW,CATLG),VOL=SER=&TDISCD
//      DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,
//      DSN=&USUARID,AROST(PRIME),
//      DISP=(NEW,KEEP),VOL=SER=&TDISCD
//      DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,
//      DSN=&USUARID,AROST(OVERFLOW),
//      DISP=(NEW,KEEP),VOL=SER=&TDISCD
//ARQTT DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=FB,LRECL=336,
//      RLKST7F=672,DSORG=IS,RKP=1,KEYLEN=255,OPTCD=LIM,
//      NTM=2),DSN=&USUARID,ARQTT(INDEX),
//      DISP=(NEW,CATLG),VOL=SER=&TDISCD
//      DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,
//      DSN=&USUARID,ARQTT(PRIME),
//      DISP=(NEW,KEEP),VOL=SER=&TDISCD
//      DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,
//      DSN=&USUARID,ARQTT(OVERFLOW),
//      DISP=(NEW,KEEP),VOL=SER=&TDISCD
//ARQFT DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=60,
//      RLKST7F=1204,DSORG=IS,RKP=5,KEYLEN=16,OPTCD=LIM,
//      NTM=2),DSN=&USUARID,ARQFT(INDEX),
//      DISP=(NEW,CATLG),VOL=SER=&TDISCD
//      DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,
//      DSN=&USUARID,ARQFT(PRIME),
//      DISP=(NEW,KEEP),VOL=SER=&TDISCD
//      DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,
//      DSN=&USUARID,ARQFT(OVERFLOW),
//      DISP=(NEW,KEEP),VOL=SER=&TDISCD
//PI11 EXEC PGM=TFMAA,PARM='LOAD.M,ITNFCNT=80,A,X',REGION=&REGCOMP
/** *****
/**      C O M P I L E R   D O   P R O G R A M   U S U A R I D
/** *****
//SYSPRINT DD SYSOUT=A
//SYS1IN DD DSNNAME=&&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,
//      SPACE=(80,(250,1(0)))
//SYSUT3 DD DSNNAME=&&SYSUT3,UNIT=SYSDA,SPACE=(80,(250,250)),
//      DCB=RLKST7F=80
//SYSUT1 DD DSNNAME=&&SYSUT1,UNIT=SYSDA,SPACE=(1024,(60,60)),CONTIG),
//      SEP=(SYSUT3,SYS1IN),DCB=RLKST7F=1024
//SYS1IB DD DSN=PBTX,FURTADD,DBET,DISP=SHR
//IKED EXEC PGM=TFWI,PARM='ITST,SIZE=200K',COND=(9,LT,PL11),
/** *****
/**      I T N K E D   T C A D   D O   P R O G R A M   U S U A R I D
/** *****
//      REGION=&REGIKED
//SYS1IB DD DSNNAME=SYS1.PI11IB,DISP=SHR
//SYS1MOD DD DSNNAME=&&G0SET(G0),DISP=(MOD,PASS),
//      UNIT=SYSDA,SPACE=(1024(50,20,1),RLSE)
//SYSUT1 DD DSNNAME=&&SYSUT1,UNIT=SYSDA,SPACE=(1024,(200,20)),

```



```

//          SFP=(SYSDMOD,SYSDTB),DCB=BIKSI7F=1024
//SYSPRINT DD  SYSOUT=A
//SYSDTN   DD  DSN=ESLLOADSET,DISP=(OLD,DELETE)
//          DD  DDNAME=SYSDTN
//DRMO     DD  DSN=PRTX_FURTADD,DBRG,DISP=SHR
//SYSDTN   DD  DSN=PRTX_FURTADD,DBRT(HYINCLUD),DISP=SHR
//GO       EXEC PGM=*,IKED,SYSDMOD,COND=((9,IT,IKED),(9,IT,PLIL)),
//* *****
//*          EXECUCAO DO PROGRAMA USUARIO
//* *****
//          REGION=EREGGO
//SYSPRINT DD  SYSOUT=A
//ARODP    DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARODP(INDEX)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=ESUARTP..ARODP(PRIME)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=(EUUSHARTP..ARODP(OVERFLOW))
//ARQEP    DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQEP(INDEX)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=(EUUSHARTP..ARQEP(PRIME))
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQEP(OVERFLOW)
//ARQTP    DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQTP(INDEX)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQTP(PRIME)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQTP(OVERFLOW)
//ARQSP    DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQSP(INDEX)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQSP(PRIME)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQSP(OVERFLOW)
//ARQDTP   DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARODP(INDEX),
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARODP(PRIME)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARODP(OVERFLOW)
//ARQETP   DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQEP(INDEX)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQEP(PRIME)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQEP(OVERFLOW)
//ARQDTP   DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQTP(INDEX)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQTP(PRIME)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQTP(OVERFLOW)
//ARQSTP   DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQSP(INDEX)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQSP(PRIME)
//          DD  DISP=(&DISPP),DCB=DSORG=IS,
//          DSN=EUUSHARTP..ARQSP(OVERFLOW)

```



```

//ARQDT DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQDT(INDEX)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQDT(PRIME)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQDT(OVERFLOW)
//ARQET DD DTSP=(&DTSP),DCB=DSORG=IS,
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQET(INDEX)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQET(PRIME)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQET(OVERFLOW)
//ARQIT DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQIT(INDEX)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQIT(PRIME)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQIT(OVERFLOW)
//ARQST DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQST(INDEX)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQST(PRIME)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQST(OVERFLOW)
//ARQDIT DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQDIT(INDEX)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQDIT(PRIME)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQDIT(OVERFLOW)
//ARQETT DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQETT(INDEX)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQETT(PRIME)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQETT(OVERFLOW)
//ARQITT DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQITT(INDEX)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQITT(PRIME)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQITT(OVERFLOW)
//ARQSTIT DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQSTIT(INDEX)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQSTIT(PRIME)
// DD DTSP=(&DTSP),DCB=DSORG=IS,
// DSN=&USUARID,ARQSTIT(OVERFLOW)
//SORTLIB DD DSN=SYS1.SORTLIB,DIS=SHR
//SORTLIST DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,3,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,3,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,3,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,3,CONTIG)
//ARQSAFT DD DUMMY,DCB=(RECFM=VB,LRECL=40,BLKSIZE=404)
//ARQSAFD DD DUMMY,DCB=(RECFM=VB,LRECL=40,BLKSIZE=404)
//EXAMINET EXEC PGM=EXAMINE,PARM='&USUARID',COND=(EVEN)
//* *****
//* RESUMO DOS ARQUIVOS TEMPORARIOS
//* *****

```



```
//SYSPRINT DD SYSOUT=A
//DELETEAT2 EXEC PGM=IEFBR14,COND=(EVEN)
//* *****
//* D E L E T E A T 2   D D S   A R O H I V O S   T E M P O R A R I O S
//* *****
//ARQDT DD DSN=EUUSUAR10..ARQDT(INDEX),
//      DISP=(OLD,DELETE),VOL=SER=&TDISCD,UNIT=SYSDA
//      DD DSN=EUUSUAR10..ARQDT(INDEX),
//      DISP=(OLD,UNCATLG),VOL=SER=&TDISCD,UNIT=SYSDA
//AROST DD DSN=EUUSUAR10..AROST(INDEX),
//      DISP=(OLD,DELETE),VOL=SER=&TDISCD,UNIT=SYSDA
//      DD DSN=EUUSUAR10..AROST(INDEX),
//      DISP=(OLD,UNCATLG),VOL=SER=&TDISCD,UNIT=SYSDA
//ARQFT DD DSN=EUUSUAR10..ARQFT(INDEX),
//      DISP=(OLD,DELETE),VOL=SER=&TDISCD,UNIT=SYSDA
//      DD DSN=EUUSUAR10..ARQFT(INDEX),
//      DISP=(OLD,UNCATLG),VOL=SER=&TDISCD,UNIT=SYSDA
//ARQIT DD DSN=EUUSUAR10..ARQIT(INDEX),
//      DISP=(OLD,DELETE),VOL=SER=&TDISCD,UNIT=SYSDA
//      DD DSN=EUUSUAR10..ARQIT(INDEX),
//      DISP=(OLD,UNCATLG),VOL=SER=&TDISCD,UNIT=SYSDA
//SYSIN DD * GENERATED STMT
IEF041I MISSING PEND STATEMENT
IEF607I JOB HAS NO STEPS
```



```

//ALDCA JOB (TRP,CDD..10),RDP,MSGLEVEL=(1,1),CLASS=A,PRTY=3
//STDD EXEC PGM=IEFBR14
//ARQDP DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=516,
//          BLKSIZE=5164,DSORG=IS,RKP=5,KEYLEN=9,OPTCD=LIM,NTM=2),
//          DSN=ARQP(INDEX),DISP=(NEW,CATLG),VOL=SER=UEPVS3
//          DD UNIT=SYSDA,SPACE=(CYL,3),DCB=DSORG=IS,DSN=ARQP(PRIME),
//          DISP=(NEW,CATLG),VOL=SER=UEPVS3
//          DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQP(OVERFLOW),
//          DISP=(NEW,CATLG),VOL=SER=UEPVS3
//ARQEP DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRECL=60,
//          BLKSIZE=1204,DSORG=IS,RKP=5,KEYLEN=16,OPTCD=LIM,NTM=2),
//          DSN=ARQEP(INDEX),DISP=(NEW,CATLG),VOL=SER=UEPVS3
//          DD UNIT=SYSDA,SPACE=(CYL,2),DCB=DSORG=IS,DSN=ARQEP(PRIME),
//          DISP=(NEW,CATLG),VOL=SER=UEPVS3
//          DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQEP(OVERFLOW),
//          DISP=(NEW,CATLG),VOL=SER=UEPVS3
//ARQIP DD UNIT=SYSDA,SPACE=(CYL,3),DCB=(RECFM=FB,LRECL=336,
//          BLKSIZE=672,DSORG=IS,RKP=1,KEYLEN=255,OPTCD=LIM,NTM=2),
//          DSN=ARQIP(INDEX),DISP=(NEW,CATLG),VOL=SER=UEPVS3
//          DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQIP(PRIME),
//          DISP=(NEW,CATLG),VOL=SER=UEPVS3
//          DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQIP(OVERFLOW),
//          DISP=(NEW,CATLG),VOL=SER=UEPVS3
//ARQSP DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=FB,LRECL=43,
//          BLKSIZE=860,DSORG=IS,RKP=1,KEYLEN=5,OPTCD=LIM,NTM=2),
//          DSN=ARQSP(INDEX),DISP=(NEW,CATLG),VOL=SER=UEPVS3
//          DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQSP(PRIME),
//          DISP=(NEW,CATLG),VOL=SER=UEPVS3
//          DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQSP(OVERFLOW),
//          DISP=(NEW,CATLG),VOL=SER=UEPVS3
//SYSIN DD * GENERATED STMT
IEF236I ALLOC. FOR ALDCA STDD
IEF237I 151 ALLOCATED TO ARQDP
IEF237I 151 ALLOCATED TO
IEF237I 151 ALLOCATED TO
IEF237I 151 ALLOCATED TO ARQEP
IEF237I 151 ALLOCATED TO
IEF237I 151 ALLOCATED TO
IEF237I 151 ALLOCATED TO ARQIP
IEF237I 151 ALLOCATED TO
IEF237I 151 ALLOCATED TO
IEF237I 151 ALLOCATED TO ARQSP
IEF237I 151 ALLOCATED TO
IEF237I 151 ALLOCATED TO
IEF142I - STEP WAS EXECUTED - COND CODE 0000
IEF287I ARQP NOT CATLGD 2
IEF287I VOL SER NOS= UEPVS3.
IEF287I ARQP NOT CATLGD 2
IEF287I VOL SER NOS= UEPVS3.
IEF287I ARQP NOT CATLGD 2
IEF287I VOL SER NOS= UEPVS3.
IEF287I ARQEP NOT CATLGD 2
IEF287I VOL SER NOS= UEPVS3.
IEF287I ARQEP NOT CATLGD 2
IEF287I VOL SER NOS= UEPVS3.
IEF287I ARQEP NOT CATLGD 2
IEF287I VOL SER NOS= UEPVS3.
IEF287I ARQIP NOT CATLGD 2
IEF287I VOL SER NOS= UEPVS3.
IEF287I ARQIP NOT CATLGD 2

```



IFF287T	V01 SER NOS= UEPVS3.	
IFF287T	ARQIP	NOT CATLGD 2
IFF287T	V01 SER NOS= UEPVS3.	
IFF287T	ARQSP	NOT CATLGD 2
IFF287T	V01 SER NOS= UEPVS3.	
IFF287T	ARQSP	NOT CATLGD 2
IFF287T	V01 SER NOS= UEPVS3.	
IFF287T	ARQSP	NOT CATLGD 2
IFF287T	V01 SER NOS= UEPVS3.	



```

//EXECUTAI JOB (TRP.CPD..10),RDP,MSGLEVEL=(1,1),CLASS=A,PRTY=2
//STOO      EXFC PGM=IEFBR14
//ARQT      DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRFCI=516,
//           BKSIZF=5164,DSORG=IS,RKP=5,KEYLEN=9,OPTCD=LIM,
//           NTM=2),
//           DSN=ARQT(INDEX),DISP=(NEW,CATLG),VOL=SER=UEPVS2
//           DD UNIT=SYSDA,SPACE=(CYL,3),DCB=DSORG=IS,DSN=ARQT(PRIME),
//           DISP=(NEW,CATLG),VOL=SER=UEPVS2
//           DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQT(OVFLOW),
//           DISP=(NEW,CATLG),VOL=SER=UEPVS2
//ARQET     DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=VB,LRFCI=60,
//           BKSIZF=1204,DSORG=IS,RKP=5,KEYLEN=16,OPTCD=LIM,
//           NTM=2),
//           DSN=ARQET(INDEX),DISP=(NEW,CATLG),VOL=SER=UEPVS2
//           DD UNIT=SYSDA,SPACE=(CYL,2),DCB=DSORG=IS,DSN=ARQET(PRIME),
//           DISP=(NEW,CATLG),VOL=SER=UEPVS2
//           DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQET(OVFLOW),
//           DISP=(NEW,CATLG),VOL=SER=UEPVS2
//ARQIT     DD UNIT=SYSDA,SPACE=(CYL,6),DCB=(RECFM=FB,LRFCI=336,
//           BKSIZF=672,DSORG=IS,RKP=1,KEYLEN=255,OPTCD=LIM,
//           NTM=2),
//           DSN=ARQIT(INDEX),DISP=(NEW,CATLG),VOL=SER=UEPVS2
//           DD UNIT=SYSDA,SPACE=(CYL,2),DCB=DSORG=IS,DSN=ARQIT(PRIME),
//           DISP=(NEW,CATLG),VOL=SER=UEPVS2
//           DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=ARQIT(OVFLOW),
//           DISP=(NEW,CATLG),VOL=SER=UEPVS2
//AROST     DD UNIT=SYSDA,SPACE=(CYL,1),DCB=(RECFM=FB,LRFCI=43,
//           BKSIZF=860,DSORG=IS,RKP=1,KEYLEN=5,OPTCD=LIM,
//           NTM=2),
//           DSN=AROST(INDEX),DISP=(NEW,CATLG),VOL=SER=UEPVS2
//           DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=AROST(PRIME),
//           DISP=(NEW,CATLG),VOL=SER=UEPVS2
//           DD UNIT=SYSDA,SPACE=(CYL,1),DCB=DSORG=IS,DSN=AROST(OVFLOW),
//           DISP=(NEW,CATLG),VOL=SER=UEPVS2

```

```

//SYSIN DD * GENERATED SJMT
IEF236I ALLOC. FOR EXECUTAI STOO
IEF237I 152 ALLOCATED TO ARQT
IEF237I 152 ALLOCATED TO
IEF237I 152 ALLOCATED TO
IEF237I 152 ALLOCATED TO ARQET
IEF237I 152 ALLOCATED TO
IEF237I 152 ALLOCATED TO
IEF237I 152 ALLOCATED TO ARQIT
IEF237I 152 ALLOCATED TO
IEF237I 152 ALLOCATED TO
IEF237I 152 ALLOCATED TO AROST
IEF237I 152 ALLOCATED TO
IEF237I 152 ALLOCATED TO

```

```
IEF142I - STEP WAS EXECUTED - COND CODE 0000
```

```

IEF287I ARQT NOT CATLGD 2 ✓
IEF287I VOL SER NOS= UEPVS2.
IEF287I ARQT NOT CATLGD 2
IEF287I VOL SER NOS= UEPVS2.
IEF287I ARQT NOT CATLGD 2
IEF287I VOL SER NOS= UEPVS2.
IEF287I ARQET NOT CATLGD 2
IEF287I VOL SER NOS= UEPVS2.
IEF287I ARQET NOT CATLGD 2
IEF287I VOL SER NOS= UEPVS2.
IEF287I AROET NOT CATLGD 2

```



```

//JOSELUCE JOB MSGLEVEL=1,CLASS=A
//ST1 EXEC PILLFCIG,PARM,PL11=(A,X),REGION,PL11=128K,
// PARM,IKFD='SI7F=200000',TIME,GO=15,REGION,GO=300K
XXPL11 EXEC PGM=TEMAA,PARM='LOAD,NOCHECK',REGION=52K 00000010
XXSYSPRINT DD SYSOUT=A,OUTIM=1800 00000020
XXSYSIN DD DSNAME=88LOADSET,DISP=(MOD,PASS),UNIT=SYSSG, 00000030
XX SPACE=(80,(250,100)) 00000040
XXSYSUT3 DD DSNAME=88SYSUT3,UNIT=SYSDA,SPACE=(80,(250,250)), 00000050
XX DCB=BLKSTZF=80 00000060
XXSYSUT1 DD DSNAME=88SYSUT1,UNIT=SYSDA,SPACE=(1024,(60,60),,CONTIG), 00000070
XX SEP=(SYSUT3,SYSIN),DCB=BLKSTZF=1024 00000080
//PL11.SYSIN DD *
IEF236T ALLLOC FOR JOSELUCE PL11 ST1
IEF237T 151 ALLOCATED TO SYSIN
IEF237T 152 ALLOCATED TO SYSUT3
IEF237T 151 ALLOCATED TO SYSUT1
IEF142T - STEP WAS EXECUTED - COND CODE 0004
IEF285T SYS9143.T104625.RE102.JOSELUCE.LOADSET PASSED
IEF285T VOL SER NOS= JEPVS3.
IEF285T SYS9143.T104625.RE102.JOSELUCE.SYSUT3 DELETED
IEF285T VOL SER NOS= JEPVS2.
IEF285T SYS9143.T104625.RE102.JOSELUCE.SYSUT1 DELETED
IEF285T VOL SER NOS= JEPVS3.
XXIKFD EXEC PGM=IEF1,PARM='XREF,IST',COND=(9,LT,PL11),REGION=96K 00000090
XXSYSITB DD DSNAME=SYST,PL11TB,DISP=SHR 00000100
//IKFD.SYSIMOD DD DSN=PBTX,FURTADD,DCB,DISP=SHR
X/SYSIMOD DD DSNAME=88GSET(GO),DISP=(MOD,PASS),UNIT=SYSDA, 00000110
XX SPACE=(1024,(50,20,1),RLSE) 00000120
XXSYSUT1 DD DSNAME=88SYSUT1,UNIT=SYSDA,SPACE=(1024,(200,20)), 00000130
XX SEP=(SYSIMOD,SYSITB),DCB=BLKSTZF=1024 00000140
XXSYSPRINT DD SYSOUT=A 00000150
XXSYSIN DD DSNAME=88LOADSET,DISP=(OLD,DELETE) 00000160
XX DD DSNAME=SYSIN 00000170
//IKFD.SYSIN DD *
IEF212T - SYSIMOD - DATA SET NOT FOUND.
IEF236T ALLLOC FOR JOSELUCE IKFD ST1
IEF272T - STEP WAS NOT EXECUTED.
XXGO EXEC PGM=*,IKFD.SYSIMOD,COND=((9,LT,IKFD),(9,LT,PL11)) 00000180
XXSYSPRINT DD SYSOUT=A 00000190
//GO.ARQDP DD DSN=ARQDP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQDP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQDP(OVERFLOW),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(OVERFLOW),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(OVERFLOW),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(OVERFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQITP DD DSN=ARQIP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIP(OVERFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS
//GO.ARQITP DD DSN=ARQIP(INDEX),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIP(PRIME),DISP=(MOD,KEEP),DCB=DSORG=IS
// DD DSN=ARQIP(OVERFLOW),DISP=(MOD,KEEP),DCB=DSORG=IS

```



```

//GD.ARCSTP DD DSN=ARCSP(INDEX),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCSP(PRIME),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCSP(OVERFLOW),DISP=(MOD,KFEP),DCB=DSORG=IS
//GD.ARCOT DD DSN=AROT(INDEX),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=AROT(PRIME),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=AROT(OVERFLOW),DISP=(MOD,KFEP),DCB=DSORG=IS
//GD.ARCFT DD DSN=ARCF(INDEX),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCF(PRIME),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCF(OVERFLOW),DISP=(MOD,KFEP),DCB=DSORG=IS
//GD.ARCIT DD DSN=ARCIT(INDEX),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCIT(PRIME),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCIT(OVERFLOW),DISP=(MOD,KFEP),DCB=DSORG=IS
//GD.ARCST DD DSN=ARCST(INDEX),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCST(PRIME),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCST(OVERFLOW),DISP=(MOD,KFEP),DCB=DSORG=IS
//GD.ARCOT DD DSN=AROT(INDEX),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=AROT(PRIME),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=AROT(OVERFLOW),DISP=(MOD,KFEP),DCB=DSORG=IS
//GD.ARCFIT DD DSN=ARCF(INDEX),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCF(PRIME),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCF(OVERFLOW),DISP=(MOD,KFEP),DCB=DSORG=IS
//GD.ARCIIT DD DSN=ARCIT(INDEX),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCIT(PRIME),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCIT(OVERFLOW),DISP=(MOD,KFEP),DCB=DSORG=IS
//GD.ARCST DD DSN=ARCST(INDEX),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCST(PRIME),DISP=(MOD,KFEP),DCB=DSORG=IS
//          DD DSN=ARCST(OVERFLOW),DISP=(MOD,KFEP),DCB=DSORG=IS
//GD.SORTIBR DD DSN=SYS1.SORTIBR,DISP=SHR
//GD.SORTIST DD SYSOUT=A
//GD.SYSOUT DD SYSOUT=A
//GD.SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)
//GD.SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)
//GD.SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)
//GD.SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)
//GD.ARCSAFE DD DUMMY
//GD.ARCSAFE DD DSN=FFTC(1),DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//          VOL=SER=UEPVS3,SPACE=(TRK,(1)),
//          DCB=(RECFM=VB,1RECL=40,BLKSI7F=404)
//GD.EMPT DD *
//GD.PRJLEFT DD *
//GD.TARFFAT DD *
//GD.HARTIIT DD *
//GD.PRJPST DD *
//

```

```

IEF236T ALL DD FOR JOSELUCE GD ST1
IEF272I - STEP WAS NOT EXECUTED.
IEF285I SYS79143.T104625.RE102.JOSELUCE.LOADSET DELETED
IEF285I VOL SER NOS=UEPVS3.
IEF298I JOSELUCE SYSOUT=A.

```



EXEMPLE C PROC OPTIONS (MAIN)

```

35      COD = INSELD ('REQ','TAREFA',.000001,'HABILIT',.00004)
36      COD = INSELD ('REQ','TAREFA',.000001,'HABILIT',.00010)
37      COD = INSELD ('REQ','TAREFA',.000002,'HABILIT',.00004)
38      COD = INSELD ('REQ','TAREFA',.000002,'HABILIT',.00010)
39      COD = INSELD ('REQ','TAREFA',.000003,'HABILIT',.00004)
40      COD = INSELD ('REQ','TAREFA',.000003,'HABILIT',.00010)
41      COD = INSELD ('REQ','TAREFA',.000004,'HABILIT',.00002)
42      COD = INSELD ('REQ','TAREFA',.000004,'HABILIT',.00007)
43      COD = INSELD ('REQ','TAREFA',.000004,'HABILIT',.00008)
44      COD = INSELD ('REQ','TAREFA',.000005,'HABILIT',.00007)
45      COD = INSELD ('REQ','TAREFA',.000005,'HABILIT',.000008)
46      COD = INSELD ('REQ','TAREFA',.000005,'HABILIT',.000011)
47      COD = INSELD ('REQ','TAREFA',.000005,'HABILIT',.000002)
48      COD = INSELD ('CAP','EMP',.000001,'HABILIT',.000001)
49      COD = INSELD ('CAP','EMP',.000002,'HABILIT',.000001)
50      COD = INSELD ('CAP','EMP',.000002,'HABILIT',.000005)
51      COD = INSELD ('CAP','EMP',.000003,'HABILIT',.000001)
52      COD = INSELD ('CAP','EMP',.000003,'HABILIT',.000005)
53      COD = INSELD ('CAP','EMP',.000004,'HABILIT',.000001)
54      COD = INSELD ('CAP','EMP',.000004,'HABILIT',.000005)
55      COD = INSELD ('CAP','EMP',.000023,'HABILIT',.000001)
56      COD = INSELD ('CAP','EMP',.000023,'HABILIT',.000005)
57      COD = INSELD ('CAP','EMP',.000005,'HABILIT',.000002)
58      COD = INSELD ('CAP','EMP',.000005,'HABILIT',.000012)
59      COD = INSELD ('CAP','EMP',.000005,'HABILIT',.000011)
60      COD = INSELD ('CAP','EMP',.000005,'HABILIT',.000007)
61      COD = INSELD ('CAP','EMP',.000005,'HABILIT',.000008)
62      COD = INSELD ('CAP','EMP',.000006,'HABILIT',.000002)
63      COD = INSELD ('CAP','EMP',.000006,'HABILIT',.000012)
64      COD = INSELD ('CAP','EMP',.000006,'HABILIT',.000011)
65      COD = INSELD ('CAP','EMP',.000006,'HABILIT',.000007)
66      COD = INSELD ('CAP','EMP',.000006,'HABILIT',.000008)
67      COD = INSELD ('CAP','EMP',.000010,'HABILIT',.000002)
68      COD = INSELD ('CAP','EMP',.000010,'HABILIT',.000007)
69      COD = INSELD ('CAP','EMP',.000010,'HABILIT',.000008)
70      COD = INSELD ('CAP','EMP',.000010,'HABILIT',.000011)
71      COD = INSELD ('CAP','EMP',.000010,'HABILIT',.000012)
72      COD = INSELD ('CAP','EMP',.000011,'HABILIT',.000002)
73      COD = INSELD ('CAP','EMP',.000011,'HABILIT',.000007)
74      COD = INSELD ('CAP','EMP',.000011,'HABILIT',.000008)
75      COD = INSELD ('CAP','EMP',.000011,'HABILIT',.000011)
76      COD = INSELD ('CAP','EMP',.000011,'HABILIT',.000012)
77      COD = INSELD ('CAP','EMP',.000012,'HABILIT',.000002)
78      COD = INSELD ('CAP','EMP',.000012,'HABILIT',.000007)
79      COD = INSELD ('CAP','EMP',.000012,'HABILIT',.000008)
80      COD = INSELD ('CAP','EMP',.000012,'HABILIT',.000011)
81      COD = INSELD ('CAP','EMP',.000012,'HABILIT',.000012)
82      COD = INSELD ('CAP','EMP',.000013,'HABILIT',.000002)
83      COD = INSELD ('CAP','EMP',.000013,'HABILIT',.000007)
84      COD = INSELD ('CAP','EMP',.000013,'HABILIT',.000008)
85      COD = INSELD ('CAP','EMP',.000013,'HABILIT',.000011)
86      COD = INSELD ('CAP','EMP',.000013,'HABILIT',.000012)
87      COD = INSELD ('CAP','EMP',.000019,'HABILIT',.000002)
88      COD = INSELD ('CAP','EMP',.000019,'HABILIT',.000007)
89      COD = INSELD ('CAP','EMP',.000019,'HABILIT',.000008)
90      COD = INSELD ('CAP','EMP',.000019,'HABILIT',.000011)
91      COD = INSELD ('CAP','EMP',.000019,'HABILIT',.000012)

```



EXEMPLED PROC OPTIONS (MAIN)

```

92      COD = INSELD ('CAP','EMP',.000020,'HABILIT',.000002)
93      COD = INSELD ('CAP','EMP',.000020,'HABILIT',.000007)
94      COD = INSELD ('CAP','EMP',.000020,'HABILIT',.000008)
95      COD = INSELD ('CAP','EMP',.000020,'HABILIT',.000011)
96      COD = INSELD ('CAP','EMP',.000020,'HABILIT',.000012)
97      COD = INSELD ('CAP','EMP',.000021,'HABILIT',.000002)
98      COD = INSELD ('CAP','EMP',.000021,'HABILIT',.000007)
99      COD = INSELD ('CAP','EMP',.000021,'HABILIT',.000008)
100     COD = INSELD ('CAP','EMP',.000021,'HABILIT',.000011)
101     COD = INSELD ('CAP','EMP',.000021,'HABILIT',.000012)
102     COD = INSELD ('CAP','EMP',.000021,'HABILIT',.000009)
103     COD = INSELD ('CAP','EMP',.000007,'HABILIT',.000004)
104     COD = INSELD ('CAP','EMP',.000007,'HABILIT',.000010)
105     COD = INSELD ('CAP','EMP',.000008,'HABILIT',.000014)
106     COD = INSELD ('CAP','EMP',.000008,'HABILIT',.000010)
107     COD = INSELD ('CAP','EMP',.000008,'HABILIT',.000013)
108     COD = INSELD ('CAP','EMP',.000009,'HABILIT',.000014)
109     COD = INSELD ('CAP','EMP',.000009,'HABILIT',.000010)
110     COD = INSELD ('CAP','EMP',.000009,'HABILIT',.000007)
111     COD = INSELD ('CAP','EMP',.000014,'HABILIT',.000014)
112     COD = INSELD ('CAP','EMP',.000014,'HABILIT',.000010)
113     COD = INSELD ('CAP','EMP',.000014,'HABILIT',.000013)
114     COD = INSELD ('CAP','EMP',.000015,'HABILIT',.000014)
115     COD = INSELD ('CAP','EMP',.000015,'HABILIT',.000010)
116     COD = INSELD ('CAP','EMP',.000016,'HABILIT',.000014)
117     COD = INSELD ('CAP','EMP',.000016,'HABILIT',.000010)
118     COD = INSELD ('CAP','EMP',.000016,'HABILIT',.000013)
119     COD = INSELD ('CAP','EMP',.000017,'HABILIT',.000014)
120     COD = INSELD ('CAP','EMP',.000017,'HABILIT',.000010)
121     COD = INSELD ('CAP','EMP',.000018,'HABILIT',.000014)
122     COD = INSELD ('CAP','EMP',.000018,'HABILIT',.000010)
123     COD = INSELD ('CAP','EMP',.000018,'HABILIT',.000007)
124     COD = INSELD ('CAP','EMP',.000022,'HABILIT',.000002)
125     COD = INSELD ('CAP','EMP',.000022,'HABILIT',.000006)
126     COD = INSELD ('CAP','EMP',.000024,'HABILIT',.000002)
127     COD = INSELD ('CAP','EMP',.000024,'HABILIT',.000006)
128     COD = INSELD ('CAP','EMP',.000024,'HABILIT',.000011)
129     COD = INSELD ('ASN','PROJPFES',.000001,'TARFFA',.000001)
130     COD = INSELD ('ASN','PROJPFES',.000001,'TARFFA',.000002)
131     COD = INSELD ('ASN','PROJPFES',.000002,'TARFFA',.000003)
132     COD = INSELD ('ASN','PROJPFES',.000003,'TARFFA',.000005)
133     COD = INSELD ('ASN','PROJPFES',.000004,'TARFFA',.000005)
134     COD = INSELD ('ASN','PROJPFES',.000005,'TARFFA',.000001)
135     COD = INSELD ('ASN','PROJPFES',.000006,'TARFFA',.000001)
136     COD = INSELD ('ASN','PROJPFES',.000007,'TARFFA',.000002)
137     COD = INSELD ('ASN','PROJPFES',.000007,'TARFFA',.000003)
138     COD = INSELD ('ASN','PROJPFES',.000008,'TARFFA',.000004)
139     COD = INSELD ('ASN','PROJPFES',.000009,'TARFFA',.000005)
140     COD = INSELD ('ASN','PROJPFES',.000010,'TARFFA',.000003)
141     COD = INSELD ('ASN','PROJPFES',.000011,'TARFFA',.000004)
142     COD = INSELD ('ASN','PROJPFES',.000012,'TARFFA',.000004)
143     COD = INSELD ('ASN','PROJPFES',.000013,'TARFFA',.000005)
144     COD = INSELD ('ASN','PROJPFES',.000014,'TARFFA',.000005)
145     COD = INSELD ('ASN','PROJPFES',.000015,'TARFFA',.000001)
146     COD = INSELD ('ASN','PROJPFES',.000016,'TARFFA',.000002)
147     COD = INSELD ('ASN','PROJPFES',.000016,'TARFFA',.000003)
148     COD = INSELD ('ASN','PROJPFES',.000017,'TARFFA',.000004)

```



VERSION 5.5

OS/360 PL/I COMPILER (F)

PAGE 1  
DATE 79.143

PL/I COMPILER OPTIONS SPECIFIED ARE AS FOLLOWS--

A.X

THE COMPLETE LIST OF OPTIONS USED DURING THIS COMPILATION IS--

FRCDIC  
CHAR60  
NOMACRO  
SOURCE2  
NOMACDCK  
COMP  
SOURCE  
ATR  
XREF  
NOEXTRF  
NOLIST  
LOAD  
NODCK  
FLAGW  
NOSTMT  
SIZE=0061440  
LINECNT=060  
OPT=00  
SORMGIN=(002,072,001)  
NOEXTDIC  
NONEST  
OPLIST  
SYNCHKT

\*OPTIONS IN EFFECT\* FRCDIC,CHAR60,NOMACRO,SOURCE2,NOMACDCK,COMP,SOURCE,ATR,XREF,NOEXTRF,NOLIST,LOAD,  
\*OPTIONS IN EFFECT\* NODCK,FLAGW,NOSTMT,SIZE=0061440,LINECNT=060,OPT=00,SORMGIN=(002,072,001),NOEXTDIC,  
\*OPTIONS IN EFFECT\* NONEST,OPLIST,SYNCHKT



```

1      EXEMPLO PROC OPTIONS (MAIN)
2
3
4      1      EXEMPLO PROC OPTIONS (MAIN)
5              /* PROCEDURES ATIVADAS */
6      2      DCI ATIVDR          ENTRY (FIXED(1),FIXED(1))
7                              RETURNS (BIN FIXED)
8      3      DCI CRIDAC          ENTRY (CHAR(7),CHAR(1),CHAR(7),
9                              PIC'(6)9',PIC'99',(4)CHAR(39))
10                             RETURNS (BIN FIXED)
11     4      DCI CRTINV          ENTRY (CHAR(7),CHAR(1),CHAR(7),
12                             PIC'99',PIC'99')
13                             RETURNS (BIN FIXED)
14     5      DCI CRTFLC          ENTRY (CHAR(7),CHAR(1),
15                             PIC'(6)9',CHAR(7),CHAR(7))
16                             RETURNS (BIN FIXED)
17     6      DCI INSELD          ENTRY (CHAR(7),CHAR(7),PIC'(6)9',
18                             CHAR(7),PIC'(6)9')
19                             RETURNS (BIN FIXED)
20     7      DCI RESMDF          ENTRY
21     8      DCI DATVDR          RETURNS (BIN FIXED)
22     9      DCI EMPS (4)        CHAR (39) INIT
23                             (*1)ESTRUTURA ARQUIVO EMP          001080CH',
24                             (*2)NOME EMPREGADO                001040CH',
25                             (*3)SALARIO                       041047CH',
26                             (*4)PROFISSAO                    048080CH')
27    10     DCI PROJETS (4)      CHAR (39) INIT
28                             (*1)ESTRUTURA ARQUIVO PROJETO      001020CH',
29                             (*2)NOME DO PROJETO                001020CH')
30    11     DCI TAREFAS (4)      CHAR (39) INIT
31                             (*1)ESTRUTURA ARQUIVO TAREFA      001015CH',
32                             (*2)NOME DA TAREFA                001015CH')
33    12     DCI HABILTS (4)      CHAR (39) INIT
34                             (*1)ESTRUTURA ARQUIVO HABILITACAO 001030CH',
35                             (*2)HABILITACAO                   001030CH')
36    13     DCI PROJPESS (4)     CHAR (39) INIT
37                             (*1)ESTRUTURA ARQUIVO PROJPESS    001060CH',
38                             (*2)NOME DO PROJETO                001020CH',
39                             (*3)NOME DO EMPREGADO              021060CH')
40    14     DCI DOMEMP          PIC '99' INIT (02)
41    15     DCI DOMPRJ          PIC '99' INIT (03)
42    16     DCI COD              BIN FIXED /* CODIGO DE RETORNO */
43    17     OPEN FILE (SYSPRINT) LINESIZE (132)
44    18     COD = ATIVDR (1,1)
45    19     COD = CRIDAC ('EMP', 'P', 'EMP', 50, 4, EMPS)
46    20     COD = CRIDAC ('PROJETO', 'P', 'PROJETO', 10, 2, PROJETS)
47    21     COD = CRIDAC ('TAREFA', 'P', 'TAREFA', 20, 2, TAREFAS)
48    22     COD = CRIDAC ('HABILIT', 'P', 'HABILIT', 15, 2, HABILTS)
49    23     COD = CRIDAC ('PROJPESS', 'P', 'PROJPESS', 30, 3, PROJPESS)
50    24     COD = CRTFLC ('MNG', 'P', 5, 'PROJETO', 'EMP')
51    25     COD = CRTFLC ('REQ', 'P', 13, 'TAREFA', 'HABILIT')
52    26     COD = CRTFLC ('CAP', 'P', 83, 'EMP', 'HABILIT')
53    27     COD = CRTFLC ('ASN', 'P', 25, 'PROJPESS', 'TAREFA')
54    28     COD = CRTINV ('EMPINV', 'P', 'EMP', 01, DOMEMP)
55    29     COD = CRTINV ('PRJIN', 'P', 'PROJPESS', 01, DOMPRJ)
56    30     COD = INSELD ('MNG', 'PROJETO', 000001, 'EMP', 000008)
57    31     COD = INSELD ('MNG', 'PROJETO', 000002, 'EMP', 000009)
58    32     COD = INSELD ('MNG', 'PROJETO', 000003, 'EMP', 000014)
59    33     COD = INSELD ('MNG', 'PROJETO', 000004, 'EMP', 000016)
60    34     COD = INSELD ('MNG', 'PROJETO', 000005, 'EMP', 000017)

```



ATTRIBUTE AND CROSS-REFERENCE TABLE

DCI NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
2	***** ATIVDR	EXTERNAL,ENTRY,BINARY,FIXED(15,0) 18
16	***** COD	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40 41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63 64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86 87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106 107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123 124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140 141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,157
3	***** CRIDAD	EXTERNAL,ENTRY,BINARY,FIXED(15,0) 19,20,21,22,23
5	***** CRIFID	EXTERNAL,ENTRY,BINARY,FIXED(15,0) 24,25,26,27
4	***** CRINV	EXTERNAL,ENTRY,BINARY,FIXED(15,0) 28,29
8	***** DATVDR	EXTERNAL,ENTRY,BINARY,FIXED(15,0) 157
14	DIEMEP	AUTOMATIC,UNALIGNED,INITIAL,DECIMAL,PICTURE(99) 28
15	DIEMPRJ	AUTOMATIC,UNALIGNED,INITIAL,DECIMAL,PICTURE(99) 29
	FMPS	(4)AUTOMATIC,UNALIGNED,INITIAL,STRING(39),CHARACTER 19
1	EXEMPLE	ENTRY,DECIMAL,FICAT(SINGLE)
17	FABILIS	(4)AUTOMATIC,UNALIGNED,INITIAL,STRING(39),CHARACTER 22
6	***** INSFID	EXTERNAL,ENTRY,BINARY,FIXED(15,0) 30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52 53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75 76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98 99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115 116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132 133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149 150,151,152,153,154,155
13	PREPSS	(4)AUTOMATIC,UNALIGNED,INITIAL,STRING(39),CHARACTER 23
10	PROJETS	(4)AUTOMATIC,UNALIGNED,INITIAL,STRING(39),CHARACTER



EXEMPLO PROC OPTIONS (MAIN)

PAGE 9

STORAGE REQUIREMENTS.  
-----

THE STORAGE AREA FOR THE PROCEDURE LABELLED EXEMPLO IS 1188 BYTES LONG.

THE PROGRAM OBJECT IS NAMED EXEMPLO AND IS 14260 BYTES LONG.

THE STATIC OBJECT IS NAMED EXEMPLICA AND IS 1143 BYTES LONG.

\*STATISTICS\*      SOURCE RECORDS =      181, PROG TEXT STMTS =      158, OBJECT BYTES =      14260







TABLE OF OFFSETS AND STATEMENT NUMBERS WITHIN PROCEDURE EXEMPLC

OFFSET (HEX)	0000	0200	0206	030E	0376	03DE	0446	04AF	0516	0576	05D6	0636	0696	06F0	074A	07AC	080E	0870	08D2	0934	0996
STATEMENT NO	1	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
OFFSET (HEX)	09FF	0A5A	0A8C	0B1E	0B8C	0BF2	0C44	0C36	0D08	0D6A	0DCC	0E2E	0E90	0FF2	0F54	0FB6	1018	107A	10DC	113E	11A0
STATEMENT NO	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
OFFSET (HEX)	1202	1264	12C6	1328	138A	13FC	144E	1480	1512	1574	15D6	1638	169A	16FC	175E	17C0	1822	1884	18E6	1948	19AA
STATEMENT NO	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
OFFSET (HEX)	1A0C	1A6E	1AD0	1B22	1B84	1BF6	1C58	1CBA	1D1C	1D7E	1DF0	1E52	1EA4	1F06	1F68	1FCA	202C	208E	20F0	2152	21B4
STATEMENT NO	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
OFFSET (HEX)	2216	2278	22DA	233C	239E	2400	2462	24C4	2526	2588	25FA	266C	26AE	2710	2772	27D4	2836	2898	28FA	295C	29BE
STATEMENT NO	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
OFFSET (HEX)	2A20	2AF2	2AF4	2B4E	2B88	2CCA	2C6C	2CCF	2D30	2D92	2DF4	2E56	2EB8	2F1A	2F7C	2FDF	3040	30A2	3104	3166	31C8
STATEMENT NO	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141
OFFSET (HEX)	322A	328C	32EE	3350	33B2	3414	3476	34D8	353A	359C	35FE	3660	36C2	3724	3786	3790	37AC				
STATEMENT NO	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158				



COMPILER DIAGNOSTICS.

WARNINGS.

IFM0764I ONE OR MORE FIXED BINARY ITEMS OF PRECISION 15 OR LESS HAVE BEEN GIVEN HALFWORD STORAGE. THEY ARE FLAGGED '\*\*\*\*\*' IN THE XREF/ATR LIST.

IFM3898I COMPILER CORE REQUIREMENT EXCEEDED SIZE GIVEN. AUXILIARY STORAGE USED.

END OF DIAGNOSTICS.

AUXILIARY STORAGE WILL NOT BE USED FOR DICTIONARY WHEN SIZE = 84K

COMPILE TIME 1.43 MINS

ELAPSED TIME 10.16 MINS



EXEMPLE PROC OPTIONS (MAIN)

```
149          COD = INSELD ('ASN','PROJPES',000018,'TARFFA',000004)
150          COD = INSELD ('ASN','PROJPES',000019,'TARFFA',000005)
151          COD = INSELD ('ASN','PROJPES',000020,'TARFFA',000005)
152          COD = INSELD ('ASN','PROJPES',000021,'TARFFA',000005)
153          COD = INSELD ('ASN','PROJPES',000022,'TARFFA',000005)
154          COD = INSELD ('ASN','PROJPES',000019,'TARFFA',000004)
155          COD = INSELD ('ASN','PROJPES',000016,'TARFFA',000005)
156          CALL RESMCR
157          COD = DATVDB
158          END EXEMPLE
```