

ÍNDICE

CONSIDERAÇÕES INICIAIS	6
PRINCIPAIS FUNÇÕES DE GERENCIAMENTO	7
CONECTIVIDADE DO CICS	8
O CICS NO AMBIENTE MVS	9
PROGRAMAÇÃO SOB CICS	
TECNICAS GERAIS DE PROGRAMAÇÃO	10
MULTITHREADING	11
VIRTUAL-STORAGE PAGING	11
PAGE-FAULT	11
TRANSAÇÕES E TASKS	
EXECUÇÃO DE UMA TRANSAÇÃO	12
PARTICULARIDADES DO PROGRAMA EM COBOL	
NÃO CODIFICAR	13
NÃO UTILIZAR	13
NÃO ESQUECER	14
INÍCIO TÍPICO DE UM PROGRAMA	15
TIPOS DE TRANSAÇÕES	
SÍNCRONAS	16
ASSÍNCRONAS	16
NÃO CONVERSACIONAL	16
CONVERSACIONAL	17
PSEUDO CONVERSACIONAL	17
PREPARAÇÃO DE UM PROGRAMA	
TRANSLATOR	20
LINKAGE EDITOR	20
O COMANDO EXEC CICS	
ARGUMENTOS POSSÍVEIS PARA PROGRAMA COBOL	21
CONSIDERAÇÕES	22
TRATAMENTO DE ERROS	23
COMANDOS DE TERMINAL CONTROL	
EXEC CICS RECEIVE	24
EXEC CICS SEND	25
EXEC CICS ISSUE ERASEAUP	27
COMANDOS DE INTERVAL CONTROL	
EXEC CICS ASKTIME	28
EXEC CICS FORMATTIME	28
COMANDOS DE PROGRAM CONTROL	
EXEC CICS LINK	30
EXEC CICS XCTL	30
EXEC CICS RETURN	32
USO DA COMMAREA	33
EXEMPLOS DE USO DA COMMAREA	34
BMS - BASIC MAPPING SUPPORT	
MAPAS	36
MAPAS FÍSICOS:	36
MAPAS LÓGICOS:	36
MAPSET	38

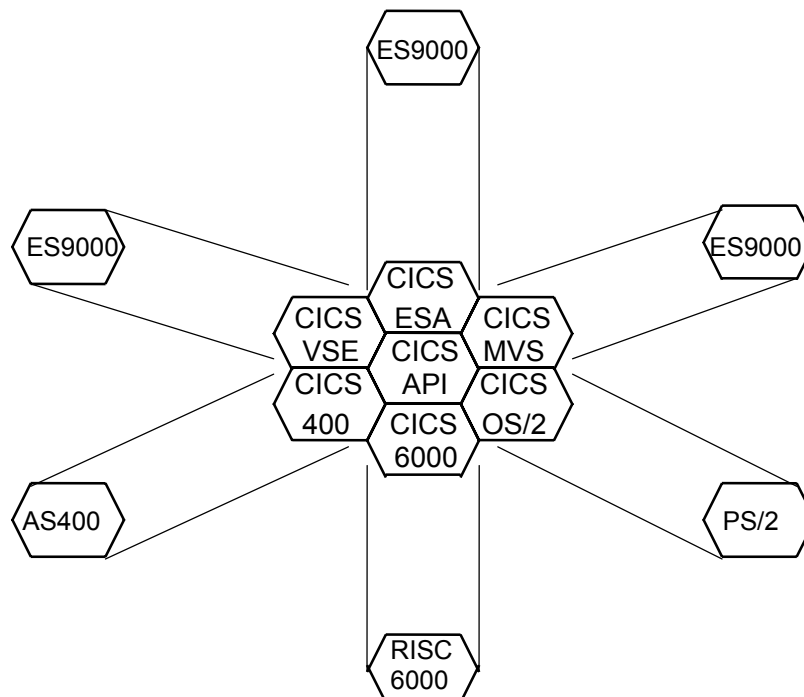
EXEMPLO	39
DEFINIÇÃO DO MAPA FÍSICO	40
DEFINIÇÃO DO MAPA LÓGICO	40
EXEC CICS SEND MAP	41
EXEC CICS RECEIVE MAP	41
EXEMPLOS DE COMANDOS DE BMS	41
ALTERANDO DINAMICAMENTE OS BYTES DE ATRIBUTO	43
TESTE DE TECLAS FUNCIONAIS	44
FILE CONTROL	
ACESSO A ARQUIVOS	45
PESQUISA A ARQUIVOS VSAM	46
ESDS	46
KSDS	46
RRDS	47
EXEC CICS READ	48
PESQUISA A ARQUIVOS VSAM	49
EXEMPLO DE PESQUISA A ARQUIVO	50
CONDIÇÕES EXCEÇÃO	51
BROWSE - PRINCIPAIS COMANDOS	
EXEC CICS STARTBR/RESETBR	53
EXEC CICS READNEXT/READPREV	53
EXEC CICS ENDBR	53
EXEMPLO DE "BROWSE"	54
ATUALIZAÇÃO DE REGISTROS VSAM	
EXEC CICS REWRITE	55
EXEC CICS READ	55
EXEC CICS DELETE	55
EXEC CICS UNLOCK	55
CONDIÇÕES DE EXCEÇÃO	57
EXEMPLO DE ATUALIZAÇÃO DE ARQUIVO	58
ATUALIZAÇÃO EM MODO CONVERSACIONAL	58
ATUALIZAÇÃO EM MODO PSEUDO CONVERSACIONAL	59
CONDIÇÃO DE "DEADLOCK"	60
PARA EVITAR DEADLOCKS:	60
ADIÇÃO DE UM NOVO REGISTRO	61
EXEC CICS WRITE	61
EXEMPLO DE ADIÇÃO	62
ACESSANDO ARQUIVOS SQL/DS ou DB2	63
ACESSANDO TABELAS DB2/SQL	65
ACESSO A INFORMAÇÕES ESPECIAIS	
EXEC CICS ASSIGN	66
ENDEREÇAMENTO DE ÁREAS EXTERNAS	
EXEC CICS ADDRESS	68
EXEMPLOS DE ENDEREÇAMENTO DE ÁREAS EXTERNAS	69
PESQUISA EM TABELA	
EXEC CICS LOAD	71
EXEC CICS RELEASE	71
CARGA DA TABELA	72
AQUISIÇÃO E LIBERAÇÃO DINÂMICA DE ÁREAS DE MEMÓRIA	
EXEC CICS GETMAIN	73

EXEC CICS FREEMAIN	73
EXEMPLO	74
MANIPULAÇÃO DE FILAS	
FILA TRANSIENT DATA	76
EXTRAPARTITION	76
INTRAPARTITION	76
GRAVAÇÃO DE MENSAGEM EM UMA FILA TD	77
EXEC CICS WRITEQ TD	77
LEITURA DE MENSAGENS DE UMA FILA TD	78
EXEC CICS READQ TD	78
DELEÇÃO DE UMA FILA TD	79
EXEC CICS DELETEQ TD	79
CONDIÇÕES DE EXCEÇÃO EM ACESSO A FILA TD	79
FILAS DE TEMPORARY STORAGE	80
GRAVAÇÃO DE MENSAGEM EM UMA FILA TS	81
EXEC CICS WRITEQ TS	81
EXEMPLOS DE GRAVAÇÃO EM UMA FILA TS	82
CRIAÇÃO DA FILA	82
ATUALIZAÇÃO DA FILA	82
LEITURA DE UMA FILA TS	83
EXEC CICS READQ TS	83
LEITURA CONCORRENTE DE UMA FILA TS (EM SEQUÊNCIA)	84
DELEÇÃO DE UMA FILA TS	85
EXEC CICS DELETEQ TS	85
CONDIÇÕES DE EXCEÇÃO EM COMANDOS DE TS	85
COMO COMPARTILHAR DADOS ENTRE TRANSAÇÕES	86
PROCESSOS DISPONÍVEIS PARA TRATAMENTO DE CONDIÇÕES DE EXCEÇÃO/ERRO	87
EXEC CICS HANDLE CONDITION	89
COMO DESATIVAR AÇÕES INDICADAS VIA HANDLE CONDITION	
EXEC CICS IGNORE	90
COMO RESTAURAR AS CONDIÇÕES DEFAULTS DE TRATAMENTO DE EXCEÇÕES	
PUSH HANDLE	91
POP HANDLE	91
TRATAMENTO DE TECLAS FUNCIONAIS	
EXEC CICS HANDLE AID	92
EXEMPLO DE HANDLE AID	93
OBSERVAÇÕES GERAIS SOBRE HANDLE E CONDITION/HANDLE AID	94
TRANSAÇÕES ÚTEIS A INSTALAÇÃO	
CEDF	
CMAC	95
CEBR	98
CECI / CECS	98
ANEXO 1 ATRIBUTOS E CÓDIGOS DE RETORNO	100
MANUAIS PARA REFERÊNCIA	101
LABORATÓRIO	
EXERCÍCIO DE FIXAÇÃO 1	102
EXERCÍCIO DE FIXAÇÃO 2 - Transações TC1A à TC1J	103
EXERCÍCIO DE FIXAÇÃO 3 - Transações TC2A à TC2J	103

EXERCÍCIO DE FIXAÇÃO 4 - Transações TC4A à TC4J	104
EXERCÍCIO DE FIXAÇÃO 5	
Mapset MPCUR1A à MPCUR1J	105
Arquivo VSAM CADCURSA à CADCRUSF	107
Transações TC5A à TC5J	108
Transações TC6A à TC6J	108
Transações TC7A à TC8J	108
Transações TC9A à TC9J	108
Apêndice A – Uma Solução para os exercícios propostos	109

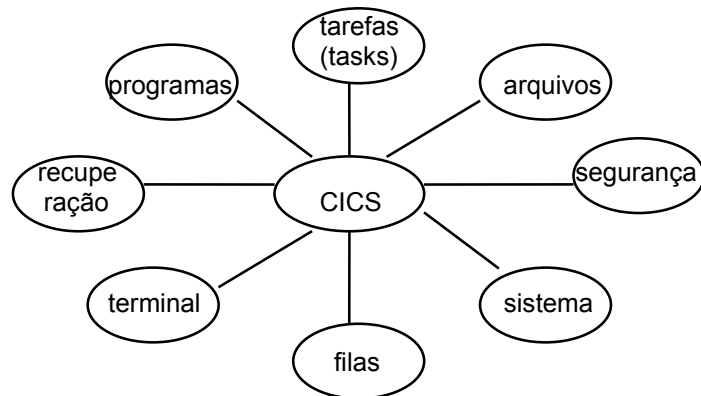
CONSIDERAÇÕES INICIAIS

Customer Information Control System



- O CICS pode ser executado em todas grandes arquiteturas e ambientes IBM.
- Embora os módulos e interfaces com o hardware/sistema operacional possam ser distintos, todas as versões do CICS compartilham um API (Application Programming Interface) comum.

PRINCIPAIS FUNÇÕES DE GERENCIAMENTO



- O CICS se apresenta ao sistema operacional como uma única tarefa (TASK) executada em um ADDRESS SPACE.
- Internamente o CICS é dividido em diversos módulos de gerenciamento de recursos e serviços.
- O analista de suporte seleciona e mantém os módulos apropriados a sua instalação, além de criar também tabelas para definir os terminais, arquivos, transações e programas.

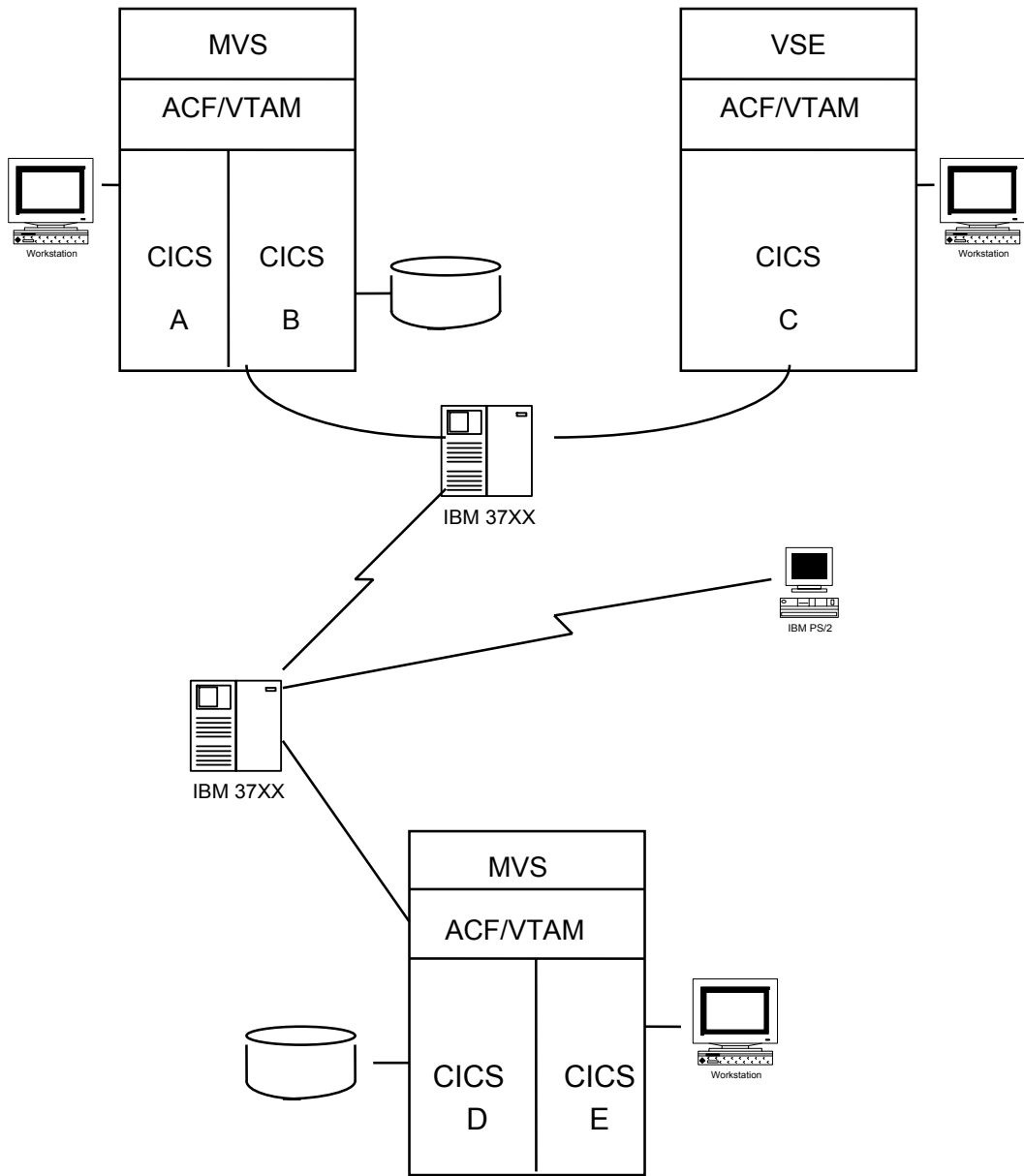
NOTAS:

TOR – Terminal only Region

AOR - Application only Region

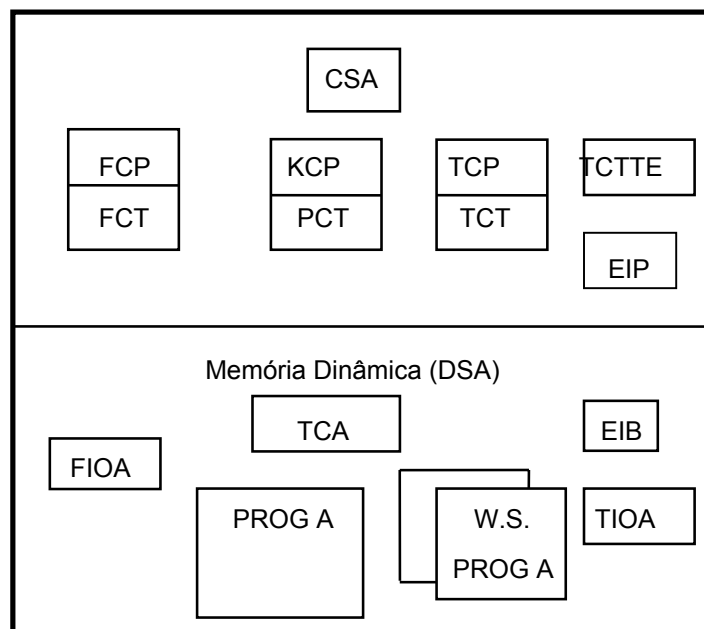
FOR - File only Region

CONECTIVIDADE DO CICS



O CICS NO AMBIENTE MVS

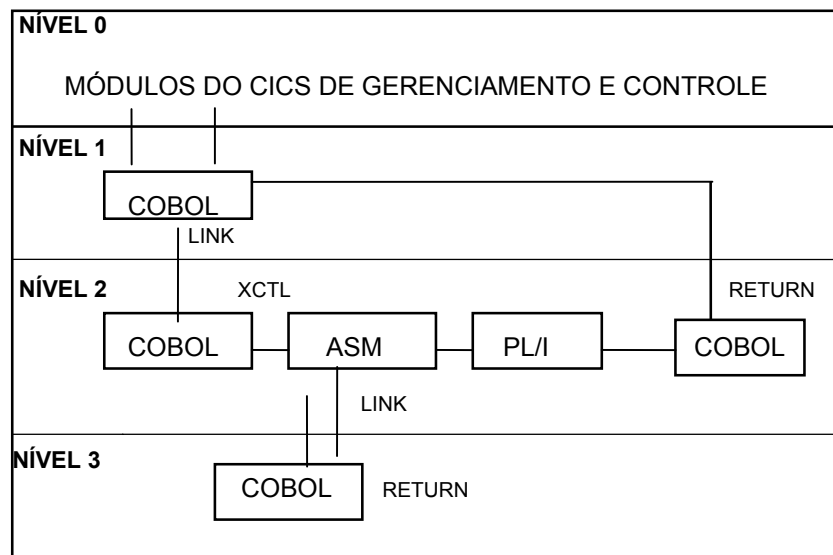
- O CICS se apresenta ao sistema operacional como uma única tarefa (TASK) executada em um ADDRESS SPACE.
- Todas as requisições de entrada/saída (I/O) feitas pelos programas de aplicação são executadas pelo CICS.
- Gerenciamentos de memória, de tasks e as solicitações dos programas de aplicação são executadas pelo CICS.



NOTAS:

PROGRAMAÇÃO SOB CICS

- Linguagens suportadas: Assembler, COBOL , C++, JAVA e PL/I.
- Níveis de programação: Macro Level (descontinuada) e Command Level.
- Programas devem ser feitos de forma modular, isto é, divididos em programas menores que passam o controle entre si, acompanhado de dados ou não.



OBS. Macro Level não é mais suportado no CICS/ESA.

- A passagem do controle de um programa para outro pode ser feita de forma temporária (**LINK**) ou permanente (**XCTL**).
- Quando a passagem é feita via LINK, o programa que recebe o controle é dito de mais baixo nível. Quando este terminar, o controle volta para a instrução seguinte ao LINK do programa chamador.
- Quando a passagem é feita via XCTL, o programa que recebe o controle é dito do mesmo nível. Quando ele terminar, o controle retorna ao programa chamador de nível mais alto, que pode ser o próprio CICS.

TECNICAS GERAIS DE PROGRAMAÇÃO

MULTITHREADING - técnica que permite que uma única cópia de programa de aplicação processe diversas transações concorrentemente. Por exemplo:

a primeira sessão, (SECTION) de um programa pode estar processando uma transação e quando terminar ou tiver sua execução suspensa, (geralmente quando se executa um comando CICS que causa WAIT), outra SECTION do programa que processe outra transação pode ser executada.

- Portanto, *multithreading* requer que programas CICS sejam quasi-reentrantes, isto é, serem serialmente reusáveis entre o ponto de entrada e de saída e que quaisquer dados ou instruções alterados sejam restaurados.
- Programa COBOL em Command-Level está dentro da regra, isto é, uma nova cópia da Working-Storage é obtida a cada vez que o programa é invocado.

VIRTUAL-STORAGE PAGING - processo que objetiva reduzir o PAGE-FAULT.

PAGE-FAULT - ocorre quando o programa refere-se a um dado ou instrução que não se encontra na memória real.

- Quando ocorre o PAGE-FAULT o Sistema Operacional é obrigado a carregar a página de memória que contém o dado ou instrução requerido. Quanto maior o fluxo, menor a performance do ONLINE como um todo.
- Técnicas geralmente utilizadas:
 1. programa ideal é aquele executado seqüencialmente, sem desvio incondicional. Exemplo: GO TO;
 2. subrotinas devem ser colocadas próximas de onde serão chamadas;
 3. dar preferência ao uso do XCTL evitando o uso do LINK;
 4. inicializar variáveis próxima da primeira referência;
 5. definir tabelas ou outras estruturas de dados na ordem que serão referenciadas;
 6. escrever programas modulares e estruturar os módulos de acordo com a freqüência de uso;
 7. usar subprogramas separados se o fluxo do programa não for sequencial;
 8. liberar memória quando o programa estiver esperando resposta do terminal, (usuário).

TRANSAÇÕES E TASKS

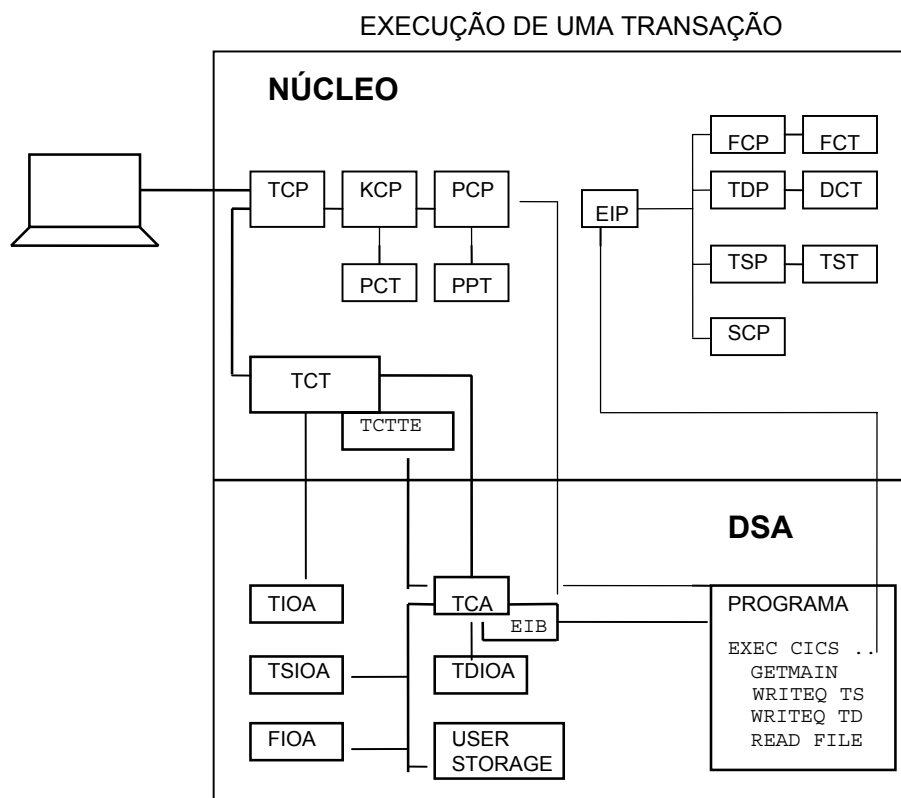
Transação é um serviço que foi previamente definido para o CICS através de um código escolhido pelo usuário. Uma transação corresponde a execução de um ou mais programas.

- Por exemplo:

TR01 - Consulta ao saldo de uma conta
TR02 - Emissão de um pedido de compra

Cada execução de uma transação define uma TASK no CICS. Uma transação pode ser executada concorrentemente por vários usuários.

- Cada execução é uma TASK. O KCP é o módulo do CICS que tem como função principal **controlar** as múltiplas TASKs em execução.



PARTICULARIDADES DO PROGRAMA CICS

COBOL

- Não há definições de arquivos: **FDs**, **SELECTs**, no programa e sim na tabela do CICS **File Control Table**;
- Programa não emite **OPEN/CLOSE**. É função do CICS;
- Programa não pode conter **STOP RUN**, utilizar **GOBACK** ou **RETURN**;
- Não codificar:

CONFIGURATION SECTION.

INPUT-OUTPUT SECTION.

FILE SECTION da DATA DIVISION

- Não utilizar:

Report Writer, Segmentation, Sort;
Count, Endjob, Flow, Dynam, State, Sysdump, Syst, Test;
Statements Cobol: ACCEPT, CURRENT-DATE, DATE, DAY, READ,
WRITE, OPEN, CLOSE, DISPLAY, EXHIBIT, SIGN IS, UNSTRING,
SEPARATE, TIME.

ASSEMBLER

- Utilizar a macro DFHEIENT para estabelecer os registradores base do programa. Cuidado com os registradores utilizados automaticamente pelo CICS.
- Não utilizar macros do Sistema Operacional e de métodos de acesso. Exemplo: OPEN, CLOSE, GET, PUT, PUTX, WRITE, READ, RETURN, etc...
- Definir áreas variáveis na Dummy Section DFHEISTG, área externa ao programa.
- Não definir arquivos através da macro DCB (Data Control Block).

NOTAS:

- Não esquecer:
 1. campo de tamanho no comando EXEC CICS, deve ser definido como **S9(4) COMP** no COBOL e, como Half Word no Assembler;
 2. a **DFHCOMMAREA** deve ser o primeiro campo da **LINKAGE SECTION**;
 3. os serviços são solicitados ao CICS via **EXEC CICS**. O programa **não se comunica direto** com o método de acesso;
 4. o **EXEC CICS HANDLE ABEND** ativa, reativa ou cancela um **EXIT** dentro do programa de aplicação. Com isso a aplicação pode tratar determinados eventos antes de “abendar”;
 5. o **EIBAID**, (campo do bloco de controle EIB, criado no início da task), é utilizado para identificar qual a tecla de função/comando que o operador pressionou;
 6. o teste do **EIBRESP** associado à opção **NOHANDLE** em substituição ao uso do **HANDLE CONDITION**, facilita a leitura do programa: melhor estruturado e mais coerente com o uso do **EIBAID**;
 7. de emitir copy para os books **DFHAID** e **DFHBMSCA**;
 8. sempre que possível, gravar informações de erro na fila **CSMT**;
 9. evite “abends” controlados.

NOTAS:

INÍCIO TÍPICO DE UM PROGRAMA CICS

COBOL

```

IDENTIFICATION DIVISION.
PROGRAM-ID. MEUPROG.
AUTHOR.
*
* ** OBSERVACOES SINTETICAS
*
ENVIRONMENT DIVISION.

DATA DIVISION.
WORKING-STORAGE SECTION.
01  FILLER          PIC  X(19)  '*** INICIO-WORKING' .
*
* ** AREAS E CAMPOS DA WORKING
*
      _____
      _____
*
* ** COPY DOS BOOKS DFH E MAPAS BMS
*
      COPY          DFHAID.
      COPY          DFHBMSCA.
      COPY          PG010MS.

LINKAGE SECTION.
01  DFHCOMMAREA PIC  X(xx) .
EJECT
PROCEDURE DIVISION.

```

ASSEMBLER

```

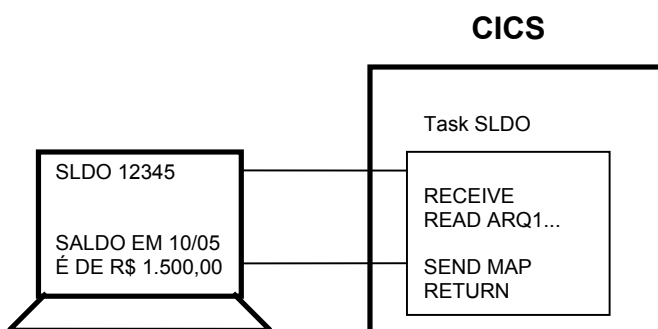
      PRINT NOGEN

R0      EQU      0
R1      EQU      1
      .
      .
      .
      EJECT
*****
DFHEISTG DSECT
      DFHEISTG
      DS      0D
USERI    DS      CL2           - CODIGO DO USUARIO
USERID   DS      CL4           - CODIGO DO USUARIO
      DS      CL4
      EJECT
      COPY    SFT010M
      DFHEIEND
      EJECT
PROG0010 DFHEIENT CODEREG=R8 ,DATAREG=R10 ,EIBREG=R3

```

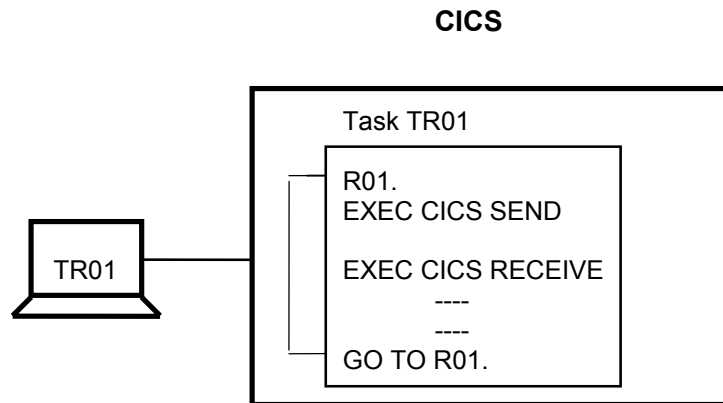
TIPOS DE TRANSAÇÕES

- Síncronas
 - Assíncronas
 - Não Conversacional
 - Conversacional
 - Pseudo Conversacional
-
- **SÍNCRONA:** associada a um terminal.
 - **ASSÍNCRONA:** não está associada a um terminal. O processamento assíncrono independe das sessões onde os pedidos iniciam e de como as repostas serão devolvidas ao operador. Não há correlação direta entre pedidos e respostas.
 - **NÃO CONVERSACIONAL:** processa um único input (que deu origem a TASK), responde e encerra. Ela nunca recebe um segundo input do terminal, portanto não há uma conversação real.
Exemplo:



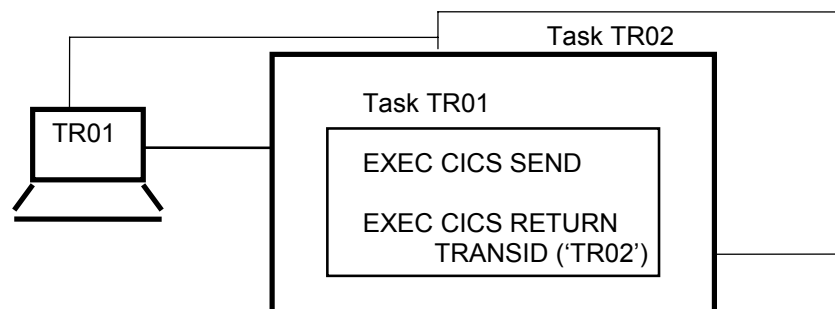
- **CONVERSACIONAL:** esse tipo de transação aguarda que o operador responda a suas solicitações sem liberar nenhum tipo de recurso até o seu término.

Exemplo:



- **PSEUDO CONVERSACIONAL:** ao terminar, a transação pseudo-conversacional, libera recursos e memória, devolvendo o controle ao CICS. Porém, associa o terminal em uso ao código de uma transação, que será inicializada quando o próximo input ocorrer no terminal.

Exemplo:

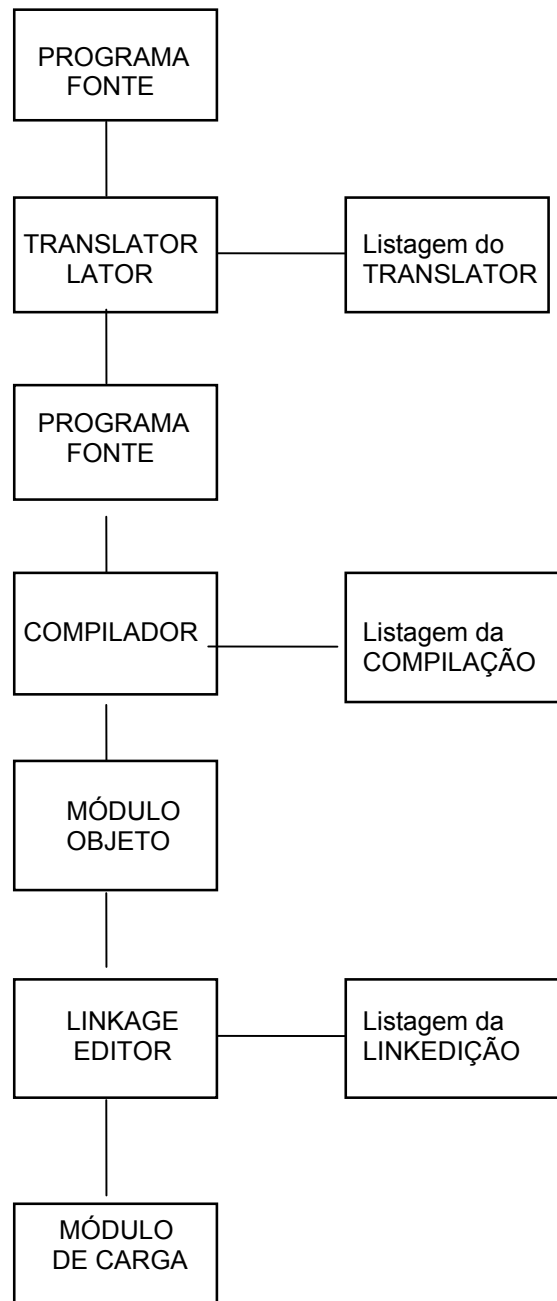


OBSERVAÇÕES

- *Operator Think Time*: tempo que o operador leva para pensar ou se preparar (digitar dados) para executar o próximo passo.
- A transação Conversacional, pelo fato de prender os recursos (memória, arquivos) durante toda sua existência e de existir uma conversação com o operador deve ser evitada.
- Na transação Pseudo-Conversacional, cada RETURN TRANSID equivale a um **sync point**, (liberação total dos recursos).

NOTAS:

PREPARAÇÃO DE UM PROGRAMA



TRANSLATOR

- Traduz comandos EXEC CICS em *MOVES* e *CALLS*

EXEC CICS RETURN END-EXEC.

gera:

```
*EXEC      CICS RETURN END-EXEC.  
  
MOVE '00160' TO DFHEIV0  
  
CALL 'DFHEI1' USING DFHEIV0
```

LINKAGE EDITOR

- Acopla o módulo DFHEI1 ao Programa Objeto.

OBSERVAÇÕES IMPORTANTES:

1. A cada TASK criada cuja aplicação é codificada em COMMAND LEVEL o DFHEIP cria uma WORKING STORAGE, devidamente inicializada para uso exclusivo da TASK.

2. Para cada TASK o DFHEIB cria um bloco de controle 'READ ONLY' chamado DFHEIB e o endereça para uso da TASK.

O EIB contém informações importantes para a programação.

3. A cada nova versão de uma aplicação catalogada na LOADLIB, é importante emitir o comando:

CEMT SET PROGRAM (programa) NEWCOPY

O COMANDO EXEC CICS

- Sintaxe:

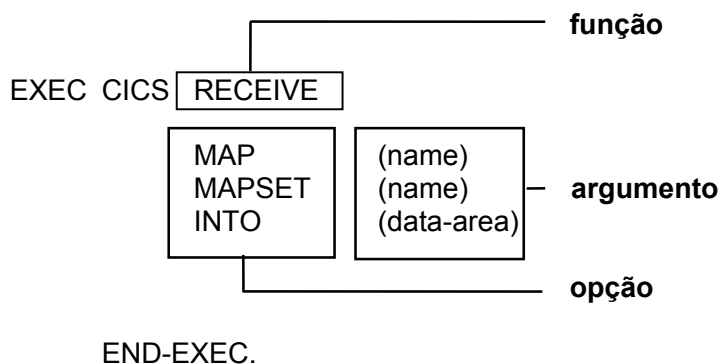
EXEC CICS *função opção argumento* END-EXEC.

função: serviço que está sendo solicitado ao CICS.

opção: descreve uma das várias opções existentes para a função desejada.

argumento: é o argumento passado para o programa "data-value" ou "data-name".

- Exemplo:



TRATAMENTO DE ERROS

- existem diversas soluções
- a solução preferencial (de acordo com a técnica de programação estruturada), é a inclusão da opção RESP dentro do comando, (*ver Anexo 1*).

- EXEMPLO:

```
WORKING-STORAGE SECTION.
```

```
01  TRABALHO.  
  
    02  TAMANHO      PIC S9(4)  COMP.  
  
    02  ERRO        PIC S9(8)  COMP.  
  
01  INAREA.  
  
    02  FILLER      PIC X(5) .  
  
    02  ACAO        PIC X(3) .  
  
    02  RESTO      PIC X(30) .  
  
    _____  
    _____
```

```
PROCEDURE DIVISION.
```

```
    _____  
    EXEC CICS RECEIVE INTO(INAREA) RESP(ERRO)
```

```
    LENGTH(TAMANHO) END-EXEC.  
  
    _____  
    _____  
    _____
```

```
    IF ERRO = DFHRESP(NORMAL) THEN
```

```
    ACAO = 'FIM' THEN  
  
    _____  
    _____
```

COMANDOS DE TERMINAL CONTROL

EXEC CICS RECEIVE INTO (data-area)

LENGTH (data-area)

END-EXEC

- **LENGTH** campo binário com 2 bytes de tamanho que especifica o maior tamanho de entrada esperado.
- **INTO** área de memória onde os dados serão colocados a disposição da aplicação.
- **LENGERR** ocorre se a mensagem for maior que o tamanho esperado.

NOTAS:

COMANDOS DE TERMINAL CONTROL, (continuação)

EXEC CICS SEND FROM (data-area)

LENGTH (data-area) ERASE WAIT

END-EXEC

- **LENGTH** número de bytes que serão enviados.
- **FROM** endereço da área que contém os dados a serem enviados.
- **ERASE** limpa a tela antes de exibir a mensagem enviada.
- **WAIT** aguarda o término da operação, antes de devolver o controle para o programa.

OBS.: UM SEND SEGUIDO POR UM RECEIVE GERA UM WAIT IMPLÍCITO.

NOTAS:

COMANDOS DE TERMINAL CONTROL, (continuação)

EXEMPLO:

WORKING-STORAGE SECTION.

```
01  TRABALHO
      02  TAM              PIC  S9(4)  COMP.
      02  ERRO            PIC  S9(8)  COMP.
01  INAREA.
      02  FILLER          PIC  X(5) .
      02  ACAO           PIC  X(3) .
      02  RESTO          PIC  X(30) .
01  OUTAREA             PIC  X(30) .
01  ERRO01             PIC  X(30)  VALUE
      `TAMANHO MAIOR QUE O PERMITIDO.` .
01  MENS01             PIC  X(30)  VALUE
      `MENSAGEM RECEBIDA.` .
```

PROCEDURE DIVISION.

```
MOVE 9 TO TAM
EXEC CICS RECEIVE INTO(INAREA)
      LENGTH(TAM) RESP(ERRO) END-EXEC.
IF ERRO IS EQUAL TO DFHRESP(LENGERR)
THEN
      MOVE ERRO01 TO OUTAREA
ELSE
      MOVE MENS01 TO OUTAREA.
EXEC CICS SEND FROM (OUTAREA) ERASE
      LENGTH(21) END-EXEC.
EXEC CICS RETURN END-EXEC.
```


COMANDOS DE TERMINAL CONTROL, (continuação)

```
EXEC CICS ISSUE ERASEAUP  
END-EXEC
```

- limpa os campos desprotegidos da tela.
- libera o teclado.

APLICAÇÃO TÍPICA: ENTRADA DE DADOS

NOTAS:

COMANDOS DE INTERVAL CONTROL

EXEC CICS ASKTIME

[ABSTIME (data-area)]

END-EXEC

- atualiza o "TIME OF DAY" a partir do CLOCK do CICS
- atualiza os campos EIBDATE e EIBTIME
- recupera o horário corrente em número de milisegundos, a partir de 01/01/1900.

EXEC CICS FORMATTIME

ABSTIME (data-area)

DATESEP('c') TIMESEP('c')

DDMMYY (data-area)

TIME (data-area)

END-EXEC

- recupera o valor armazenado, adquirido via ASKTIME, no formato especificado pelo usuário, (Ver opções no comando).

COMANDOS DE INTERVAL CONTROL, (continuação)

EXEMPLO:

```
EXEC      CICS ASKTIME ABSTIME(ABSOLUTO)
END-EXEC
```

```
EXEC      CICS FORMATTIME ABSTIME(ABSOLUTO)
DATESEP('/') TIMESEP(':')
DDMMYY(DATAFOR)
TIME(TIMEFOR)
END-EXEC
```

OBS: Se ao invés de indicação explícita for especificado DATEFORM(DATAFOR), a data aparece no formato especificado na SIT.

ABSOLUTO área de 8 bytes em formato compactado.

PIC S9(15) COMP-3

DATAFOR campo de 8 bytes em caracter.

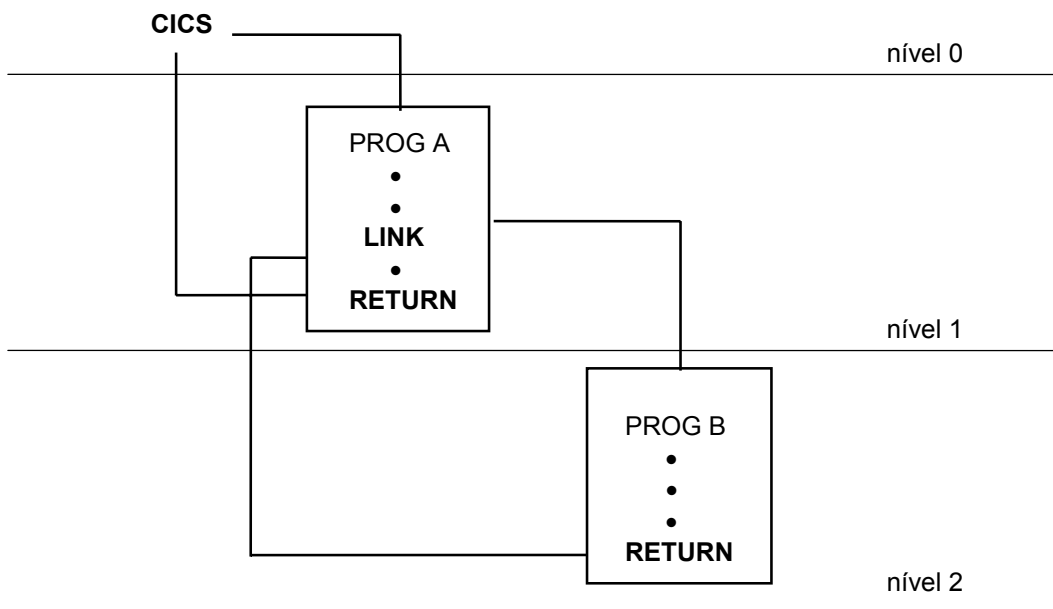
PIC X(8)

TIMEFOR idem.

NOTAS:

COMANDOS DE PROGRAM CONTROL

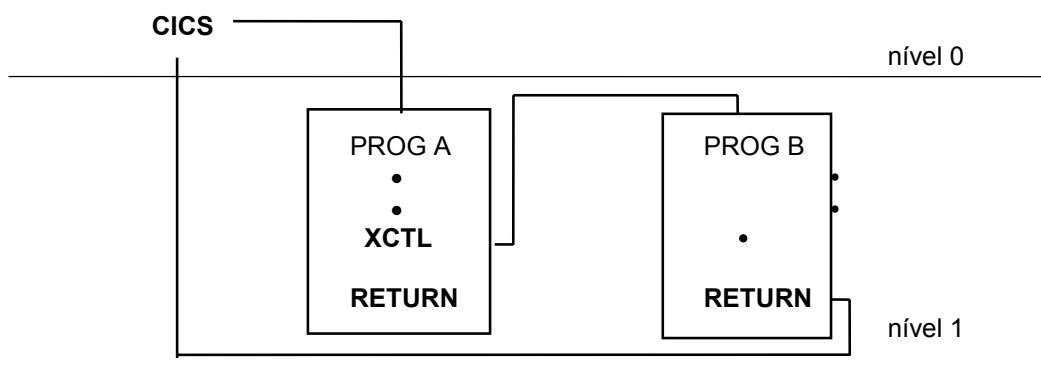
```
EXEC CICS LINK PROGRAM ('name')  
  [COMMAREA (data-area)  
  [LENGTH (data-value)]]  
END-EXEC
```



NOTAS:

COMANDOS DE PROGRAM CONTROL, (continuação)

```
EXEC CICS XCTL PROGRAM ('name')  
  [COMMAREA (data-area)  
  [LENGTH (data-value)]]  
END-EXEC
```



NOTAS:

COMANDOS DE PROGRAM CONTROL, (continuação)

```
EXEC CICS RETURN  
  [TRANSID ('name')  
  [COMMAREA (data-area)  
  [LENGTH (data-value)]]]  
END-EXEC
```

NOTAS:

COMANDOS DE PROGRAM CONTROL, (continuação)

USO DA COMMAREA

Podemos passar uma área com dados entre programas ou transações, através da COMMAREA.

- Quando um programa emite um **LINK** para outro programa.

Nesta caso, o programa chamado tem acesso **a mesma** COMMAREA do programa que emitiu o LINK.

- Quando um programa emite um **XCTL** para um outro programa ou um comando **EXEC CICS RETURN TRANSID()**.

Neste caso, a COMMAREA **é copiada** para o novo programa ou para a nova transação.

- O programa chamado **deve testar** o conteúdo do campo **EIBCALEN** para saber se existe ou não COMMAREA.

NOTAS:

COMANDOS DE PROGRAM CONTROL, (continuação)

EXEMPLOS DE USO DA COMMAREA

CASO 1 - COMMAREA passada via LINK

PGM0010

```
WORKING-STORAGE SECTION.  
01  PASSAR.  
    05  CHAVE  PIC  99.  
    05  NOME   PIC  X(18).  
  
PROCEDURE DIVISION.  
  
    EXEC  CICS LINK PROGRAM ('PGM0020')  
          COMMAREA (PASSAR)  
          LENGTH (20)  
          END-EXEC.  
  
    IF    CHAVE EQUAL 0  
        - - - -  
        - - - -  
        - - - -  
  
    EXEC  CICS RETURN  
          END-EXEC.
```

PGM0020

```
WORKING-STORAGE SECTION.  
  
LINKAGE SECTION.  
  
01  DFHCOMMAREA.  
    05  CHAVE  PIC  99.  
    05  NOME   PIC  X(18).  
  
PROCEDURE DIVISION.  
    - - - -  
    IF    EIBCALEN EQUAL 0  
        PERFORM 999-ERRO.  
    - - - -  
    - - - -  
    MOVE  0 TO CHAVE  
    - - - -  
    EXEC  CICS RETURN  
          END-EXEC.
```


COMANDOS DE PROGRAM CONTROL, (continuação)

CASO 2 - COMMAREA PASSADA VIA RETURN

PGM0010 - TRANSAÇÃO TA01

```
WORKING-STORAGE SECTION.  
01  COMUNICA  
    02 FLAG-W  PIC  X(01) VALUE 'N'.  
    02 RESTO-W PIC  X(50).  
  
    _____  
    _____  
    _____  
  
LINKAGE SECTION.  
01  DFHCOMMAREA.  
    02 FLAG-L  PIC  X(01).  
    02 RESTO-L PIC  X(50).  
  
    .  
    .  
    .  
  
PROCEDURE DIVISION.  
* A APLICACAO ESTA COMECANDO?  
  IF  EIBCALEN=0  
    MOVE 'S' TO FLAG-W  
  
    _____  
    _____  
    _____  
  
  EXEC  CICS RETURN  
  
    TRANSID ('TA01')  
  
    COMMAREA (COMUNICA)  
  
    LENGTH(51) END-EXEC.  
  
*A APLICACAO NAO ESTA COMECANDO!  
  
  EXEC  CICS LINK  
    PROGRAM ('PRCA02')  
    COMMAREA (DFHCOMMAREA)  
    LENGTH(EIBCALEN) END-EXEC.  
  
    _____  
  
  GOBACK.
```

BMS - BASIC MAPPING SUPPORT

O BMS-BASIC MAPPING SUPPORT é uma *interface* entre o CICS e o programa de aplicação. Existem 3 versões de **mapas** suportados pelo BMS:

- BMS básico, (minimum function BMS)
- BMS padrão, (standard function BMS)
- BMS total, (full function BMS)

Permite tornar o programa de aplicação independente das características do hardware do terminal, pois baseia-se em pré-definições existentes em tabelas do CICS e informações nos **mapas**.

MAPAS

Contém a formatação dos campos para input e output, tornando a comunicação com o usuário mais amigável e segura.

- **MAPAS FÍSICOS:**

Define o posicionamento, valor e atributo dos campos constantes e de campos variáveis.

- **MAPAS LÓGICOS:**

Define os "layouts" das áreas de saída (para mapas de saída) ou das áreas de entrada (para mapas de entrada) correspondentes aos respectivos mapas físicos.

- **MAPSET**

Um conjunto de mapas, utilizados conjuntamente em uma aplicação define um MAPSET. No entanto, um MAPSET pode ser constituído de um único mapa.

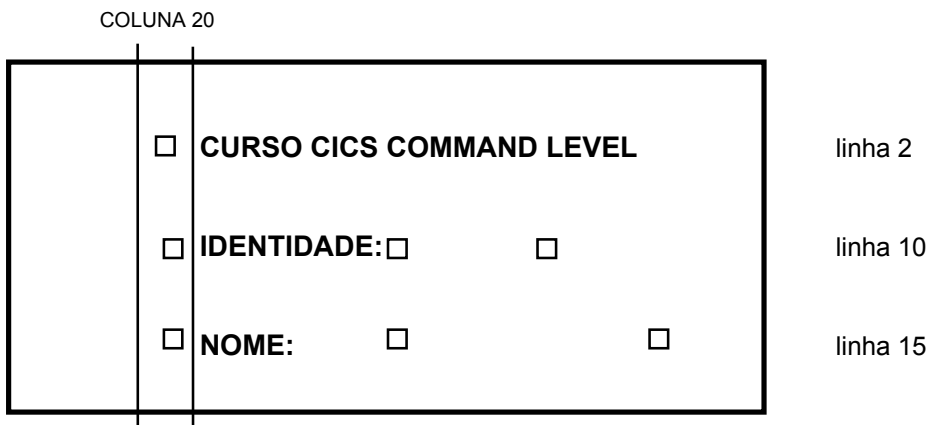
BMS - BASIC MAPPING SUPPORT, (continuação)

DFHMSD	especificação do MAPSET
DFHMDI	especificação do 1º mapa
DFHMDF	especificação do 1º campo do 1º mapa
DFHMDF	especificação do 2º campo do 1º mapa
DFHMDI	especificação do 2º mapa
DFHMDF	especificação do 1º campo do 2º mapa
DFHMDF	especificação do 2º campo do 2º mapa
DFHMSD	Fim do MAPSET

NOTAS:

BMS - BASIC MAPPING SUPPORT, (continuação)

EXEMPLO



DEFINIÇÃO DO MAPA FÍSICO

1	10	16	72
	PRINT NOGEN		
MSCURSO	DFHMSD	TYPE=MAP, TIOAPFX=YES, MODE=INOUT, CTRL= (FREEKB, FRSET)	x
MACURSO	DFHMDI	SIZE= (24, 80), LINE=1, COLUMN=1	
	DFHMDF	POS= (2, 20), LENGTH=24, ATTRB=PROT, INITIAL='CURSO CICS COMMAND LEVEL'	x
	DFHMDF	POS= (10, 20), LENGTH=11, ATTRB=PROT, INITIAL=' IDENTIDADE: '	x
IDENT	DFHMDF	POS= (10, 32), LENGTH=8, ATTRB= (UNPROT, IC)	x
	DFHMDF	POS= (10, 41), ATTRB=ASKIP	
	DFHMDF	POS= (15, 20), LENGTH=5, ATTRB=PROT, INITIAL=' NOME: '	x
NOME	DFHMDF	POS= (15, 26), LENGTH=30, ATTRB=UNPROT	
	DFHMDF	POS= (15, 57), ATTRB=ASKIP	
	DFHMSD	TYPE=FINAL	

Continuação: caractere diferente de **space** na coluna 72 e na próxima linha continuar a partir da coluna 16.

BMS - BASIC MAPPING SUPPORT, (continuação)

NOME DFHMSD TYPE= {DSECT | MAP}, TIOAPFX=YES,
[STORAGE=AUTO],
MODE = {IN | OUT | INOUT},
LANG = {ASM | COBOL},
CTRL = ([PRINT] [,FREEKB] [, ALARM] [,FRSET]),
[SUFFIX = suffix]

NOME DFHMDI SIZE= (linha,coluna),LINE=(número),
COLUMN = (número)

{NOME} DFHMDF POS= {(número) (linha,coluna)},LENGTH=(número),
INITIAL='constante',
JUSTIFY = ([LEFT | RIGHT] [,BLANK | ZERO])
ATTRB=([ASKIP | PROT | UNPROT] [,NUM]
[,BRT | NORM | DRK] [,IC] [,FSET])

DFHMSD TYPE=FINAL

DEFINIÇÃO DO MAPA LÓGICO

- Utiliza o mesmo conjunto de macros especificado no exemplo anterior, com **TYPE=DSECT** e **LANG=COBOL**

MSCURSO DFHMSD TYPE=DSECT, TIOAPFX=YES,
MODE=INOUT,
LANG=COBOL, CTRL={FREEKB, FRSET}

.
.

DFHMSD TYPE=FINAL

OBS:O nome do MAPSET deve constar da PPT.

- O mapa físico deve ser catalogado em uma **LOADLIB** conhecida do CICS e o mapa lógico em uma **SOURCE** ou **COPY LIB**.

BMS - BASIC MAPPING SUPPORT, (continuação)

SAÍDA GERADA PELO MAPA LÓGICO

```
01 MACURSOI .
    02 FILLER          PIC  X(12) .
    02 IDENTL         PIC  S9(4) COMP .
    02 IDENTA         PIC  X .
    02 FILLER         REDEFINES IDENTA .
        03 IDENTF     PIC  X .
    02 IDENTI         PIC  X(8) .
    02 NOMEI         PIC  S9(4) COMP .
    02 NOMEA         PIC  X .
    02 FILLER         REDEFINES NOMEA .
        03 NOMEF     PIC  X .
    02 NOMEI         PIC  X(30) .
01 MACCURSOO REDEFINES MACURSOI .
    02 FILLER          PIC  X(12) .
    02 FILLER          PIC  X(3) .
    02 IDENTO         PIC  X(8) .
    02 FILLER          PIC  X(3) .
    02 NOMEI         PIC  X(30) .
```

UTILIZANDO O MAPA LÓGICO

```
WORKING-STORAGE SECTION.
```

```
-----  
-----
```

```
COPY MSCURSO.
```

```
-----
```

BMS - BASIC MAPPING SUPPORT, (continuação)

```
EXEC CICS SEND MAP ('name')
          MAPSET ('name')
          FROM (data-area)
          [DATAONLY | MAPONLY] [ALARM]
          [CURSOR (data-value)]
          [ERASE] [FREEKB] [FRSET]
          END-EXEC
```

```
EXEC CICS RECEIVE MAP ('name')
          MAPSET ('name')
          INTO (data-area)
          END-EXEC
```

OBS: Toda a vez que for emitido um *RECEIVE MAP* para recebimento de dados não formatados ou toda a vez que o usuário apertar uma tecla funcional sem entrar dados o programa sai pela condição de **MAPFAIL**.

EXEMPLOS DE COMANDOS DE BMS

```
EXEC CICS SEND MAP ('MACURSO') MAPSET ('MSCURSO')
          MAPONLY END-EXEC.
```

```
EXEC CICS RECEIVE MAP ('MACURSO') MAPSET ('MSCURSO')
          INTO (MACURSOI) END-EXEC.
```

BMS - BASIC MAPPING SUPPORT, (continuação)

NOTAS:

BMS - BASIC MAPPING SUPPORT, (continuação)

ALTERANDO DINAMICAMENTE OS BYTES DE ATRIBUTO

WORKING-STORAGE SECTION.

COPY DFHBMSCA, (Ver Anexo 1)
DFHBMUNP - DESPROTEGIDO
DFHBMPRO - PROTEGIDO
DFHBMBRY - BRILHANTE
DFHBMASK - AUTOSKIP
DFHBMFSE - DESPROTEGIDO/MDT SET
DFHBMPRF - PROTEGIDO/MDT SET

PROCEDURE DIVISION.

MOVE DFHBMASK TO CODIGOA

MOVE LOW-VALUES TO CODIGOO

EXEC CICS SEND MAP ...

NOTAS:

BMS - BASIC MAPPING SUPPORT, (continuação)

TESTE DE TECLAS FUNCIONAIS

- Após a utilização de uma tecla funcional, o campo **EIBAID** é posicionado de acordo com a tecla pressionada.
- Procedimento de teste:
 1. **copiar o book DFHAID** na WORKING-STORAGE SECTION.
 2. testar o campo **EIBAID** com os **nomes gerados via DFHAID**.
- EXEMPLO:

WORKING-STORAGE SECTION

```
COPY DFHAID
    DFHENTER
    DFHCLEAR
    DFHPA1
    .
    DFHPF1
    .
```

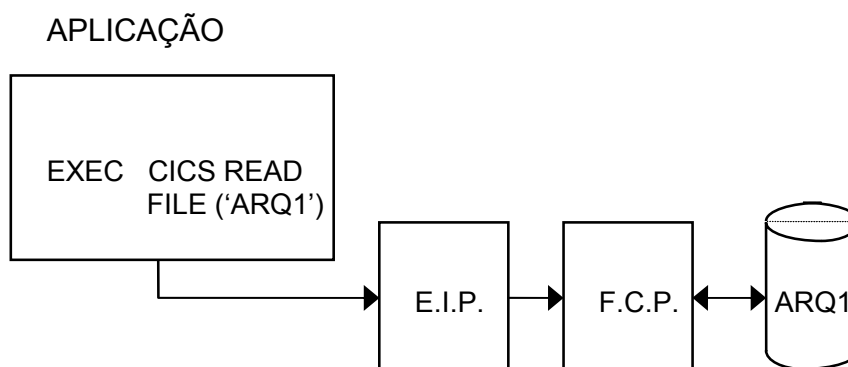
PROCEDURE DIVISION.

```
EXEC CICS RECEIVE MAP _____
IF EIBAID IS EQUAL DFHPF12 THEN
_____
_____
_____
```

NOTAS:

FILE CONTROL - Comandos, Opções e condições

ACESSO A ARQUIVOS



- O arquivo e suas características, bem como as possibilidades de acesso devem estar definidos na FCT (File Control Table)
 - Arquivos suportados pelo File Control:
 - VSAM
 - KSDS
 - ESDS
 - RRDS
- BDAM

NOTAS:

FILE CONTROL - Comandos, Opções e condições, (continuação)

PESQUISA A ARQUIVOS VSAM

ESDS

- todo registro é identificado pelo seu RBA.
- adição somente no final do arquivo.
- atualização **INPLACE** permitida sem alteração de tamanho.
- deleção não permitida.
- normalmente usado para pesquisa sequencial.

100	200	150	-----
bytes	bytes	bytes	
RBA=0	RBA=100	RBA=300	RBA=450

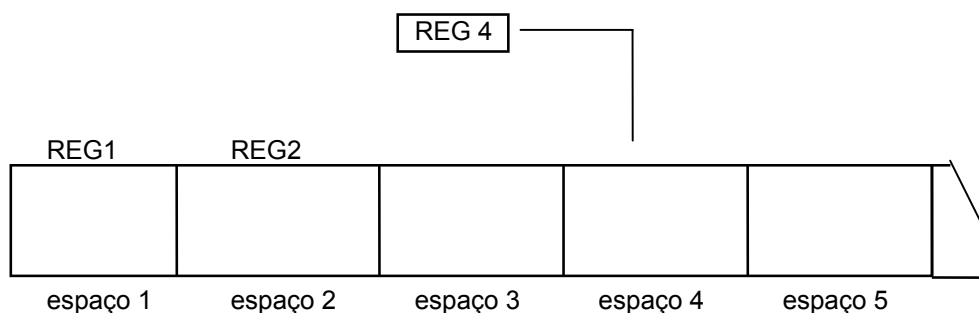
KSDS

- registros podem ser acessados sequencialmente ou diretamente através de campo chave.
- registros podem ser incluídos, atualizados e deletados.
- todo CLUSTER KSDS é constituído de uma área de índice e uma área de dados.
- espaços livres devem ser projetados dentro do CIs e dos CAs para minimizar a ocorrência de "splits".

FILE CONTROL - Comandos, Opções e condições, (continuação)

RRDS

- cada registro é acessado por sua posição específica dentro do arquivo.
- cada registro possui um espaço, de tamanho idêntico, reservado dentro do arquivo.
- os registros podem ser acessados sequencialmente ou diretamente pelo número relativo do registro no arquivo.



OBSERVAÇÃO:

- em todo acesso a arquivo VSAM é necessário informarmos um campo identificador:

KSDS — CHAVE COMPLETA OU PARCIAL

ESDS — RBA

RRDS — RRN

FILE CONTROL - Comandos, Opções e condições,
(continuação)

```
EXEC CICS READ FILE ('name')  
          LENGTH (data-area)  
          RIDFLD (data-area)  
          INTO (data-area)  
          [KEYLENGTH (data-value)]  
          [GENERIC]  
          [RBA | RRN]  
          [GTEQ | EQUAL]  
          END-EXEC
```

NOTAS:

FILE CONTROL - Comandos, Opções e condições, (continuação)

PESQUISA A ARQUIVOS VSAM

```
WORKING-STORAGE SECTION.  
01 REGISTRO.  
-----  
-----  
-----  
-----  
01 MAPAI.  
-----  
-----  
-----  
-----  
05 MATI          PIC X(6) .  
-----  
-----  
-----  
01 TAMREC          PIC S9(4)    COMP .  
-----  
-----  
-----  
-----  
PROCEDURE DIVISION.  
EXEC  CICS RECEIVE MAP('MAPA') - - MAPSET(--)  
      END-EXEC.  
MOVE 100 TO TAMREC.  
EXEC  CICS READ FILE('ARQ1') RIDFLD(MATI)  
      INTO(REGISTRO) LENGTH(TAMREC)  
      END-EXEC.
```

NOTAS:

FILE CONTROL - Comandos, Opções e condições, (continuação)

EXEMPLO DE PESQUISA A ARQUIVO

CHAVES

12500	13870	21325	23500	28120
-------	-------	-------	-------	-------

ARGUMENTO DE PESQUISA: 13100
PESQUISA COMPLETA MAIOR OU IGUAL
REGISTRO OBTIDO: 13870

PESQUISA GENÉRICA MAIOR OU IGUAL
TAMANHO DO ARGUMENTO DE PESQUISA: 3
ARGUMENTO DE PESQUISA: 200
REGISTRO OBTIDO: 21325

PESQUISA GENÉRICA IGUAL
TAMANHO DO ARGUMENTO DE PESQUISA: 2
ARGUMENTO DE PESQUISA: 22
REGISTRO NÃO ENCONTRADO

NOTAS:

FILE CONTROL - Comandos, Opções e condições, (continuação)

CONDIÇÕES EXCEÇÃO

PREVISTAS	[ENDFILE NOTFND]
ERROS	[INVREQ FILENOTFOUND ILLOGIC]
EXTERNAS	[IOERR DISABLED NOTOPEN]

- **ERROR** - QUALQUER CONDIÇÃO DIFERENTE DE **NORMAL**.
- **ILLOGIC** - ERRO INTERNO DO VSAM

PESQUISAR BYTES 1 E 2 DO EIBRCODE

BYTE1 - VSAM RETURN CODE

BYTE2 - VSAM ERROR CODE

FILE CONTROL - Comandos, Opções e condições, (continuação)

EXEMPLO DE TESTE DE CONDIÇÃO DE ERRO

WORKING-STORAGE SECTION

```
-----  
-----  
-----  
01      CODERRO      PIC      S9(8) COMP.  
-----  
-----
```

PROCEDURE DIVISION.

```
-----  
-----  
-----  
      EXEC CICS READ FILE('ARQ1') RESP(CODERRO)  
              END-EXEC.  
      IF CODERRO IS EQUAL TO DFHRESP(NORMAL) THEN  
-----  
-----  
-----  
      ELSE  
              IF CODERRO IS EQUAL TO DFHRESP(NOTFND) THEN  
-----  
-----  
-----
```

NOTAS:

FILE CONTROL - Comandos, Opções e condições, (continuação)

BROWSE - PRINCIPAIS COMANDOS

STARTBR/RESETBR

```
EXEC CICS STARTBR FILE('name') RIDFLD(data-area)
      [KEYLENGTH(data value) [GENERIC ]]
      [REQID(data-value)] [RBA | RRN]
      [GTEQ|EQUAL]
      END-EXEC
```

READNEXT/READPREV

```
EXEC CICS READNEXT FILE ('name') INTO (data-area)
      LENGTH (data-area) RIDFLD (data-area)
      [KEYLENGTH (data-value)] [RBA | RRN]
      REQID (data-value)
      END-EXEC
```

ENDBR

```
EXEC CICS ENDBR FILE('name') REQID(data-value)
```

NOTAS:

FILE CONTROL - Comandos, Opções e condições, (continuação)

EXEMPLO DE "BROWSE"

```

WORKING-STORAGE SECTION.
-----
-----
-----
    05 CHAVE          PIC X(8).
    05 CODERRO        PIC S9(8) COMP.
-----
-----
-----
PROCEDURE DIVISION.
-----
-----
MOVE  VALOR          TO CHAVE

EXEC  CICS STARBR FILE('ARQ1') RIDFLD(CHAVE)
      RESP(CODERRO) END-EXEC.

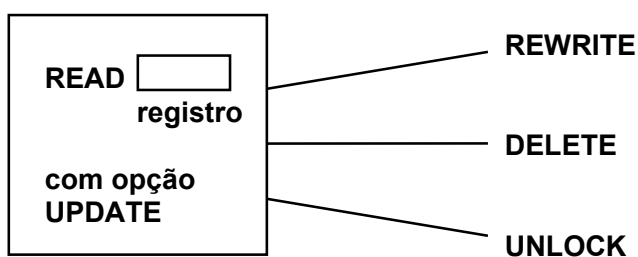
IF    CODERRO IS EQUAL DFHRESP(NORMAL)
      PERFORM ROT-LER UNTIL CODERRO
      EQUAL DFHRESP(ENDFILE)
      EXEC CICS ENDBR FILE('ARQ1') END-EXEC
ELSE
      GO TO ROTERRO.
-----
-----
-----
ROT-LER.

EXEC  CICS READNEXT FILE('ARQ1') INTO( )
      LENGTH ( ) RIDFLD(CHAVE) RESP(CODERRO)
      END-EXEC
-----
-----
-----
ROTERRO.
-----
-----

```

FILE CONTROL - Comandos, Opções e condições, (continuação)

ATUALIZAÇÃO DE REGISTROS VSAM



NOTAS:

FILE CONTROL - Comandos, Opções e condições, (continuação)

COMANDOS DE ATUALIZAÇÃO

REWRITE

```
EXEC CICS REWRITE FILE (name)
          FROM (data-area)
          LENGTH (data-value)
          END-EXEC
```

DELETE

```
EXEC CICS READ FILE ('name')
          RIDFLD (data-area)
          INTO (data-area) UPDATE
          END-EXEC
```

```
EXEC CICS DELETE FILE ('name') [GENERIC
          NUMREC (data-area)]
          [RBA/RRN]
          END-EXEC
```

```
EXEC CICS DELETE FILE ('name')
          RIDFLD (data-area)
```

UNLOCK

```
EXEC CICS UNLOCK FILE (name)
```

FILE CONTROL - Comandos, opções e condições, (continuação)

CONDIÇÕES DE EXCEÇÃO

- LENGERR
- NOSPACE
- INVREQ
- ILLOGIC

ILLOGIC - bytes 1 e 2 do campo **EIBRCODE**

VSAM RETURN CODE

VSAM ERROR CODE

NOTAS:

FILE CONTROL - Comandos, opções e condições, (continuação)

EXEMPLO DE ATUALIZAÇÃO DE ARQUIVO

```
MOVE valor TO CHAVE.  
EXEC CICS READ FILE ('ARQ1') RIDFLD (CHAVE)  
      INTO (ARQENT) LENGTH (RECL) UPDATE  
      RESP (CODERRO)  
      END-EXEC.  
IF CODERRO IS NOT EQUAL DFHRESP (NORMAL)  
  GO TO ROTERRO.  
* ATUALIZA OS CAMPOS DO REGISTRO  
EXEC CICS REWRITE FILE ('ARQ1') FROM (ARQENT)  
      LENGTH (RECL) RESP (CODERRO)  
      END-EXEC.  
IF CODERRO IS NOT EQUAL DFHRESP (NORMAL)  
  GO TO ROTERRO.
```

ATUALIZAÇÃO EM MODO CONVERSACIONAL

```
READ FILE ('ARQNC') UPDATE  
SEND MAP  
RECEIVE MAP  
REWRITE FILE ('ARQNC')
```

OBS: O Control Interval que contém o registro lido para atualização fica sob controle exclusivo até o REWRITE.

**FILE CONTROL - Comandos, Opções e condições,
(continuação)**

ATUALIZAÇÃO EM MODO PSEUDO CONVERSACIONAL

TRANSAÇÃO : TRU1

```
-----  
-----  
READ FILE ('ARQ01')  
SEND MAP  
RETURN TRANSID ('TRU2')
```

TRANSAÇÃO: TRU2

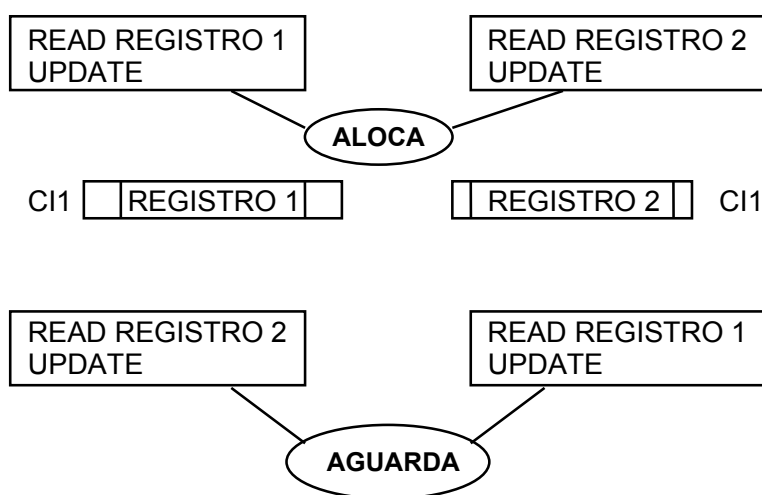
```
-----  
-----  
RECEIVE MAP  
READ FILE ('ARQ01') UPDATE  
REWRITE ('ARQ01')  
RETURN
```

- O EXEMPLO ACIMA FUNCIONA ?

NOTAS:

FILE CONTROL - Comandos, Opções e condições, (continuação)

CONDIÇÃO DE "DEADLOCK"



PARA EVITAR DEADLOCKS:

- todos os programas que acessam os mesmos conjuntos de arquivos devem emitir read com opção UPDATE na mesma ordem.
- procurar reter o C.I. durante o menor período de tempo possível.

OBS: CICS controla 'DEADLOCKS' via 'ABEND' da TASK por tempo.

NOTAS:

FILE CONTROL - Comandos, opções e condições, (continuação)

ADIÇÃO DE UM NOVO REGISTRO

```
EXEC CICS WRITE FILE ('name')  
      FROM (data-area)  
      LENGTH (data-value)  
      RIDFLD (data-area)  
      [RBA | RRN] [MASSINSERT]  
      END-EXEC
```

- a opção **MASSINSERT** deve ser utilizada quando se deseja incluir um conjunto de registros em ordem de chave crescente.
- após o término da operação de **MASSINSERT** deve ser emitido o comando **'UNLOCK'**.

NOTAS:

FILE CONTROL - Comandos, Opções e condições, (continuação)

EXEMPLO DE ADIÇÃO

```
-----  
-----  
-----  
EXEC CICS WRITE FILE('ARQNC') RIDFLD(CHAVE)  
          FROM(ENTRADA) LENGTH(RECL) RESP(CODERRO)  
          END-EXEC.  
  
IF CODERRO IS NOT EQUAL DFHRESP(NORMAL)  
  GO TO ROTERRO  
-----  
-----  
-----
```

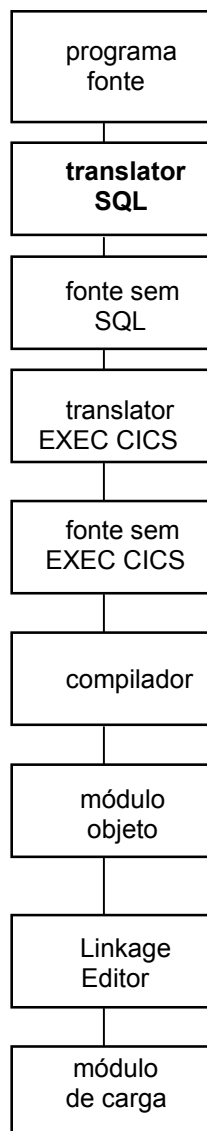
OBSERVAÇÃO:

- em arquivos ESDS o registro é sempre adicionado no fim do arquivo e o RBA do registro volta em RIDFLD.
- em arquivos KSDS se já existir registro com a mesma chave ocorre a condição DUPREC.

NOTAS:

ACESSANDO ARQUIVOS SQL/DS ou DB2

- Comando EXEC SQL requer uso de Tradutor especial, que transforma o EXEC SQL em CALL.



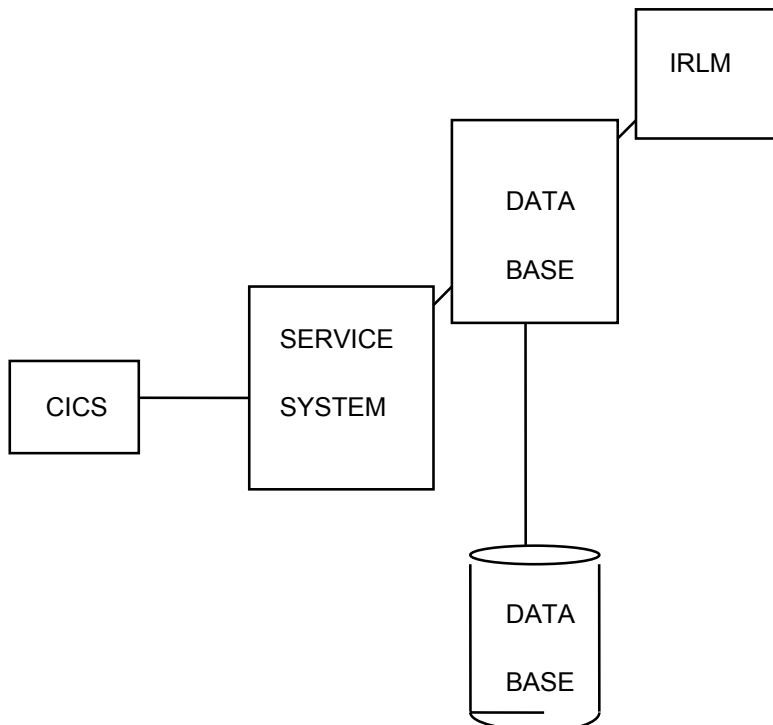
ACESSANDO ARQUIVOS SQL/DS ou DB2, (continuação)

OBSERVAÇÕES:

- DB2 executa em região fora CICS.
- Integridade dos dados é do próprio do DB2.
- Caminho do CICS para o DB2 são definidos através da Resource Control Table.
- Vantagens:

estrutura de dados maleável

independência da aplicação em relação a modificação dos dados.



ACESSANDO ARQUIVOS SQL/DS ou DB2, (continuação)

ACESSANDO TABELAS DB2/SQL

WORKING-STORAGE SECTION.

```
EXEC SQL INCLUDE SQLCA END-EXEC.  
** expande book SQL-CODE (mensagens de erro).
```

```
EXEC SQL INCLUDE .... END-EXEC.  
*** copy do layout de cada tabela
```

```
EXEC SQL DECLARE CURSORn  
          CURSOR FOR SELECT ....
```

PROCEDURE DIVISION.

```
EXEC SQL SELECT nome-col1, nome-col2, ...  
          INTO :nome-col1, :nome-col2, ...  
          FROM userid.nome da tabela  
          WHERE ....  
          ORDER BY
```

001-ROTINA-ERRO.

```
EXEC SQL ROLLBACK  
EXEC CICS SYNCPOINT ROLLBACK  
          NOHANDLE END-EXEC.
```

NOTAS:

ACESSO A INFORMAÇÕES ESPECIAIS

EXEMPLO:

NOME DO CICS	(APPLID)
TAMANHO DA CWA	(CWALENG)
TAMANHO DA TWA	(TWALENG)
TAMANHO DA TCTUA	(TCTUALENG)
NOME DO TERMINAL VTAM	(NETNAME)
'startcode'	(STARTCODE)
USERID	(USERID)

- como acessar ?

```
EXEC CICS ASSIGN option(data-area) ...  
END-EXEC
```

EXEMPLOS:

```
WORKING-STORAGE SECTION.  
77  TAMTWA      PIC S9(4) COMP.  
77  VNome      PIC X(8) .  
-----  
PROCEDURE DIVISION.  
-----  
EXEC CICS ASSIGN TWALENG(TAMTWA) END-EXEC  
EXEC CICS ASSIGN NETNAME(VNome) END-EXEC  
-----
```

NOTAS

ENDEREÇAMENTO DE ÁREAS EXTERNAS

```
LINKAGE SECTION.  
  
01  DFHCOMMAREA.  
    -----  
    -----  
  
01  LISTA.  
    |  
    | 02  PTRLST      PIC S9(8) COMP.  
    |  
    | 02  PTR-AREA1  PIC S9(8) COMP.  
    | 02  PTR-AREA2  PIC S9(8) COMP.  
    |  
    | 01  AREA1 .  
    |    -----  
    |    -----  
    |    -----  
    |  
    | 01  AREA2 .  
    |    -----  
    |    -----  
    |    -----
```

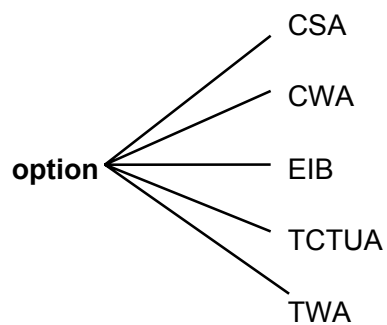
OBS:

- PTRLST endereça LISTA
- PTR-AREA1 endereça AREA1
- PTR-AREA2 endereça AREA2

NOTAS

ENDEREÇAMENTO DE ÁREAS EXTERNAS, (continuação)

```
EXEC CICS ADDRESS option(pointer-reference)
END-EXEC
```



- no CICS/ESA não é permitido o acesso a CSA portanto, independente do nível do CICS, evite:

EXEC CICS ADDRESS CSA.

NOTAS

ENDEREÇAMENTO DE ÁREAS EXTERNAS, (continuação)

EXEMPLOS DE ENDEREÇAMENTO DE ÁREAS EXTERNAS

```
WORKING-STORAGE SECTION.  
77    TAMCWA      PIC S9(4) COMP.  
      -----  
      -----  
      -----  
  
LINKAGE SECTION.  
01    DFHCOMMAREA.  
      -----  
      -----  
      -----  
  
01    LISTA.  
      02    PTR-LST PIC S9(8) COMP.  
      02    PTR-CWA PIC S9(8) COMP.  
  
01    CWAREG.  
      02    -----  
      02    -----  
  
PROCEDURE DIVISION.  
      EXEC CICS ASSIGN CWALENG(TAMCWA) END-EXEC  
      IF TAMCWA > 0  
          EXEC CICS ADDRESS CWA(PTR-CWA) END-EXEC  
      ELSE  
      -----  
      -----  
      -----
```

ENDEREÇAMENTO DE ÁREAS EXTERNAS, (continuação)

COBOL II

```
WORKING-STORAGE SECTION.  
77  LRECL-REC1  PIC S9(4) COMP.  
-----  
-----  
LINKAGE SECTION.  
01  DFHCOMMAREA.  
-----  
-----  
01  REC1.  
-----  
PROCEDURE DIVISION.  
-----  
-----  
      EXEC CICS READ SET (ADDRESS OF REC1) LENGTH (LRECL-REC1)  
      END-EXEC  
  
-----  
-----
```

COBOL OS/VS

```
WORKING-STORAGE SECTION.  
77  LRECL-REC1  PIC S9(4) COMP.  
-----  
-----  
LINKAGE SECTION.  
01  DFHCOMMAREA.  
-----  
-----  
01  LISTA.  
    05  FILLER          PIC  S9(8) COMP.  
    05  BLL-REC1       PIC  S9(8) COMP.  
01  REC1.  
-----  
-----  
PROCEDURE DIVISION.  
-----  
-----  
      EXEC  CICS READ SET (BLL-REC1) LENGTH (LRECL-REC1)  
      END-EXEC  
      SERVICE RELOAD REC1.  
-----  
-----  
-----
```

PESQUISA EM TABELA

```
EXEC CICS LOAD PROGRAM ('name')  
          [SET (ptr-ref) LENGTH (data-area)]  
          [HOLD]  
          END-EXEC
```

- OPÇÃO HOLD:

mantém a tabela residente na memória até a emissão de um RELEASE

se omitida, a tabela é automaticamente deletada da memória quando a TASK terminar.

OBS.: COMANDO DO P.C.P.

NOTAS

PESQUISA EM TABELA, (continuação)

CARGA DA TABELA

```
WORKING-STORAGE SECTION

77    TAM-TAB      PIC    S9(8) COMP
-----
-----
-----

LINKAGE SECTION.

01    DFHCOMMAREA.
-----
-----

01    LISTA.

      02    PTR-LST      PIC S9(8) COMP.

      02    PTR-TAB      PIC S9(8) COMP.
-----
-----
-----

01 TABELA.

      02 -----

      02 -----
-----
-----

PROCEDURE DIVISION.
-----
-----
EXEC CICS LOAD PROGRAM('TABNC')
      LENGTH(TAM-TAB) SET(PTR-TAB)
      END-EXEC
-----
-----
```

AQUISIÇÃO E LIBERAÇÃO DINÂMICA DE ÁREAS DE MEMÓRIA

```
EXEC CICS GETMAIN SET (ptr-ref)
          LENGTH (data-value)
          INITIMG (data-value) [NOSUSPEND]
END-EXEC
```

```
EXEC CICS FREEMAIN DATA(data-área)
END-EXEC
```

UTILIZAÇÃO:

quando se necessita a alocação de grandes áreas de memória durante curto período de tempo.

NOTAS:

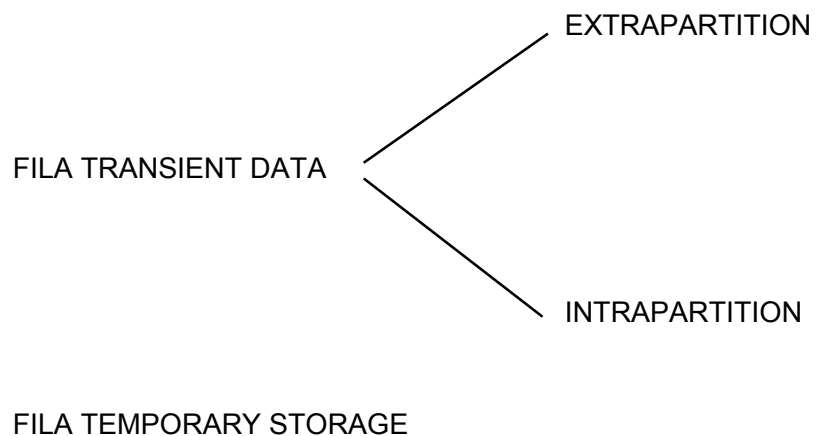
AQUISIÇÃO E LIBERAÇÃO DINÂMICA DE ÁREAS DE MEMÓRIA, (continuação)

EXEMPLO:

```
WORKING-STORAGE SECTION.
77  RECHEIO      PIC  X VALUE LOW-VALUE.
77  RETORNO      PIC  S9(8) COMP.
-----
-----
-----
LINKAGE SECTION.
-----
-----
01  LISTA.
     02  PTR-LST      PIC  S9(8) COMP.
     02  PTR-MEM      PIC  S9(8) COMP.
-----
-----
01  MEMORIA.
     02  -----
     02  -----
-----
PROCEDURE DIVISION.
-----
-----
EXEC  CICS GETMAIN SET(PTR-MEM) LENGTH(4096)
      INITIMG(RECHEIO) RESP(RETORNO) NOSUSPEND
      END-EXEC.
SET ADDRESS OF MEMORIA TO PTR-MEM
IF RETORNO IS NOT EQUAL DFHRESP(NORMAL)
-----
-----
-----
-----
EXEC  CICS FREEMAIN DATA(MEMORIA) END-EXEC.
```


MANIPULAÇÃO DE FILAS

- O CICS suporta 2 tipos de filas:



NOTAS

FILA TRANSIENT DATA

EXTRAPARTITION

- arquivo sequencial
- entrada ou saída
- destinos definidos na DCT

INTRAPARTITION

- filas sequenciais
- acesso destrutivo
- destinos definidos na **DCT**
- pode ser associado um '**TRIGGER**' a cada destino forçando o início automático de uma transação, se atingido.
- Cl's livres podem ou não ser reusado automaticamente.

NOTAS

FILA TRANSIENT DATA

GRAVAÇÃO DE MENSAGEM EM UMA FILA TD

```
EXEC CICS WRITEQ TD QUEUE ('name')  
FROM (data-area)  
LENGTH (data-value)  
END-EXEC
```

EXEMPLO:

```
WORKING-STORAGE SECTION.  
01  CONTROLES .  
    02  TDTAM      PIC      S9(4)  COMP .  
    02  RETORNO   PIC      S9(8)  COMP .  
    -----  
    -----  
    -----  
01  TDSAI .  
    02  -----  
    02  -----  
    -----  
    -----  
PROCEDURE DIVISION.  
    -----  
    -----  
    -----  
    MOVE 150 TO TDTAM  
    EXEC CICS  WRITEQ TD QUEUE ('NC01')  
              FROM(TDSAI) LENGTH(TDTAM)  
              RESP(RETORNO) END-EXEC.  
    IF RETORNO IS NOT EQUAL DFHRESP(NORMAL)  
      GO TO ROTERRO.  
    -----  
    -----  
    -----  
ROTERRO.  
    -----  
    -----  
    -----
```

FILA TRANSIENT DATA

LEITURA DE MENSAGENS DE UMA FILA TD

```
EXEC CICS READQ TD QUEUE ('name')  
          INTO (data-area)  
          [NOSUSPEND]  
          END-EXEC
```

EXEMPLO:

```
MOVE 150 TO TDTAM  
EXEC CICS  READQ TD QUEUE ('NCO05')  
          INTO (TDENT) LENGTH (TDTAM) RESP (RETORNO)  
          END-EXEC.  
IF RETORNO IS NOT EQUAL DFHRESP (NORMAL)  
  GO TO ROTERRO.  
-----  
-----  
ROTERRO.
```

FILA TRANSIENT DATA

DELEÇÃO DE UMA FILA TD

- necessário quando os CIs acessados para leitura não são liberados automaticamente: **(REUSE=NO)**

```
EXEC CICS DELETEQ TD QUEUE ('name')  
END-EXEC
```

EXEMPLO:

```
EXEC CICS DELETEQ TD QUEUE ('NC10')  
RESP (RETORNO) END-EXEC
```

CONDIÇÕES DE EXCEÇÃO EM ACESSO A FILA TD

QZERO fila vazia ou fim da fila

LENGERR tamanho omitido/estouro da área de entrada

NOSPACE

NOTOPEN

QBUSY

IOERR

QIDERR

FILAS DE TEMPORARY STORAGE

- memória ou disco
- DATAIDs não vinculados
- acesso não destrutivo
- usuário deve providenciar deleção
- acesso sequencial/randômico
- TST só para:
segurança de acesso
DATAIDS recuperáveis
DATAIDS remotos
- registros de tamanho variável

NOTAS

FILAS DE TEMPORARY STORAGE

GRAVAÇÃO DE MENSAGEM EM UMA FILA TS

```
EXEC CICS WRITEQ TS QUEUE ('name') FROM (data-area)
        LENGTH (data-value)
        [ITEM (data-area)] [MAIN | AUXILIARY]
        [REWRITE]
END-EXEC
```

OBS: Utilizado para a criação de uma fila TS. Após cada registro gravado, é retornado na área especificada em ITEM, o número do registro.

ITEM indica o número do registro a ser alterado.

NOTAS

FILAS DE TEMPORARY STORAGE

EXEMPLOS DE GRAVAÇÃO EM UMA FILA TS

CRIAÇÃO DA FILA:

```
EXEC CICS WRITEQ TS QUEUE ('NCTSFILA')  
      FROM (TS-SAI) LENGTH (100) AUXILIARY  
      END-EXEC
```

ATUALIZAÇÃO DA FILA:

```
WORKING-STORAGE SECTION.  
-----  
-----
```

```
77 NUMITEM PIC S9(4) COMP.  
-----  
-----  
-----
```

```
01 TSSAI
```

```
    02 -----  
    -----  
    -----
```

```
PROCEDURE DIVISION.  
-----  
-----  
-----
```

```
MOVE 3 TO NUMITEM
```

```
EXEC CICS WRITEQ TS QUEUE ('NCTSFILA')  
      FROM (TS-SAI) LENGTH (100) ITEM (NUMITEM)  
      REWRITE AUXILIARY END-EXEC.
```

```
-----  
-----
```


FILAS DE TEMPORARY STORAGE

LEITURA DE UMA FILA TS

- **Leitura sequencial por uma TASK:**

```
EXEC CICS READQ TS QUEUE ('name')
      INTO (data-area) LENGTH (data-area)
      [NUMITEMS (data-area)]
      END-EXEC
```

EXEMPLO:

```
EXEC CICS READQ TS QUEUE ('NCTSFILA')
      INTO (ENTTS)
      LENGTH (TSTAM)
      END-EXEC
```

- **Leitura randômica**

```
EXEC CICS READQ TS QUEUE ('name')
      INTO (data-area) LENGTH (data-area)
      [NUMITEMS (data-area)] [ITEM(data-area)]
      END-EXEC
```

EXEMPLO:

```
MOVE 5 TO NUMITEM

EXEC CICS READQ TS QUEUE ('NCTSFILA')
      INTO (ENTTS)
      LENGTH (TSTAM)
      ITEM (NUMITEM)
      END-EXEC
```

FILAS DE TEMPORARY STORAGE

LEITURA CONCORRENTE DE UMA FILA TS (EM SEQUÊNCIA)

- cada TASK utiliza a opção ITEM

EXEMPLO:

```

WORKING-STORAGE SECTION.

77  NUMITEM      PIC S9(4) COMP VALUE LOW-VALUES.

77  TSTAM        PIC  S9(4) COMP VALUE 200.
      -----
      -----
      -----

01  ENTTS.

      02 -----
      -----
      -----

PROCEDURE DIVISION
      -----
      -----
      -----

VOLTA.
      ADD 1 TO NUMITEM

      EXEC  CICS  READQ  TS  QUEUE('NCTSFILA')
              INTO(ENTTS)  LENGTH(TSTAM)  ITEM(NUMITEM)

              END-EXEC
      -----
      -----
      -----
      -----

      GO TO VOLTA
  
```

FILAS DE TEMPORARY STORAGE

DELEÇÃO DE UMA FILA TS

```
EXEC CICS DELETEQ TS QUEUE ('name')  
END-EXEC
```

EXEMPLO:

```
EXEC CICS DELETEQ TS QUEUE ('NCTSFILA')  
END-EXEC.
```

CONDIÇÕES DE EXCEÇÃO EM COMANDOS DE TS

ITEMERR o item especificado não existe

LENGERR erro de tamanho

NOSPACE

IOERR

QIDERR

NOTAS

COMO COMPARTILHAR DADOS ENTRE TRANSAÇÕES:

CWA

TCTUA

COMMAREA

VIA TELA

TEMPORARY STORAGE

TRANSIENT DATA

CLUSTER VSAM

NOTAS

PROCESSOS DISPONÍVEIS PARA TRATAMENTO DE CONDIÇÕES DE EXCEÇÃO/ERRO

- via RESP (---) dentro do próprio comando.
- deixando que assumido o tratamento 'DEFAULT' conforme o erro ocorrido.

ABEND TASK
SUSPEND TASK

- desviar para uma rotina comum ou específica de tratamento de erro.
- colocando a opção NOHANDLE no comando e testando o campo EIBRCODE.
- Após a execução do comando, testar o campo EIBRESP com o código específico da condição a ser validada.

00	OK
01	ERROR
.	
.	
.	
13	NOT FOUND
14	DUPREC
87	OPEN ERROR

NOTAS

PROCESSOS DISPONÍVEIS PARA TRATAMENTO DE CONDIÇÕES DE EXCEÇÃO/ERRO, (continuação)

TRATAMENTO DE CONDIÇÕES DE ERRO

- via codificação da instrução "HANDLE CONDITION" antes da codificação comandos aos quais ela se aplica.
- podemos codificar várias instruções "HANDLE CONDITION" a última ação especificada para tratamento de um certo erro é considerada.

EXEMPLO:

```

-----
-----
HANDLE CONDITION NOTFND(ROT1)
-----
-----
HANDLE CONDITION NOTFND(ROT2)
EXEC  CICS  READ  DATASET (-----)
                RIDFLD (CHAVE) -----  END-EXEC

```

- em qualquer instrução a opção NOHANDLE desativa o tratamento de erro para qualquer exceção ocorrida.

EXEMPLO:

```

HANDLE CONDITION NOTFND(ROT1) DSIDERR(ROT2)
EXEC  CICS  READ  DATASET (      ) RIDFLD(CHAVE)
                NOHANDLE -----  END-EXEC

```

PROCESSOS DISPONÍVEIS PARA TRATAMENTO DE CONDIÇÕES DE EXCEÇÃO/ERRO, (continuação)

TRATAMENTO DE CONDIÇÕES DE EXCEÇÃO

```
EXEC CICS HANDLE CONDITION condition (label)  
condition (label) condition (label) ----  
END-EXEC
```

EXEMPLOS:

```
1) EXEC CICS HANDLE CONDITION NOTFND(ROTNAO)  
ERROR(ROTERR) END-EXEC
```

```
2) EXEC CICS HANDLE CONDITION NOTFND(ROTNAO)  
DSIDERR(ROTDAT) END EXEC
```

- nesse caso qualquer condição de exceção provocará ação DEFAULT.

```
3) EXEC CICS HANDLE CONDITION NOTFND(ROTNAO)  
DSIDERR ERROR(ROTERR) END-EXEC
```

NOTAS:

PROCESSOS DISPONÍVEIS PARA TRATAMENTO DE CONDIÇÕES DE EXCEÇÃO/ERRO, (continuação)

COMO DESATIVAR AÇÕES INDICADAS VIA HANDLE CONDITION

- incluindo no comando a opção NOHANDLE
- via IGNORE CONDITION

formato:

```
EXEC CICS IGNORE CONDITION
condition condition
END-EXEC
```

EXEMPLO:

```
EXEC CICS IGNORE CONDITION NOTFND DSIDERR
END-EXEC
```

- a ocorrência das condições acima poderia ser testada pesquisando-se o EIBRCODE.

```
EXEC CICS READ DATASET (-----)
RIDFLD(CHAVE) ----- END-EXEC

IF EIBRCODE IS NOT EQUAL LOW-VALUES GO
TO ROTERRO.
-----
-----
-----
ROTERRO.
-----
-----
-----
```


PROCESSOS DISPONÍVEIS PARA TRATAMENTO DE CONDIÇÕES DE EXCEÇÃO/ERRO, (continuação)

COMO RESTAURAR AS CONDIÇÕES DEFAULTS DE TRATAMENTO DE EXCEÇÕES:

PUSH HANDLE

desativa, restaurando o tratamento DEFAULT, todos os procedimentos para tratamento de exceções.

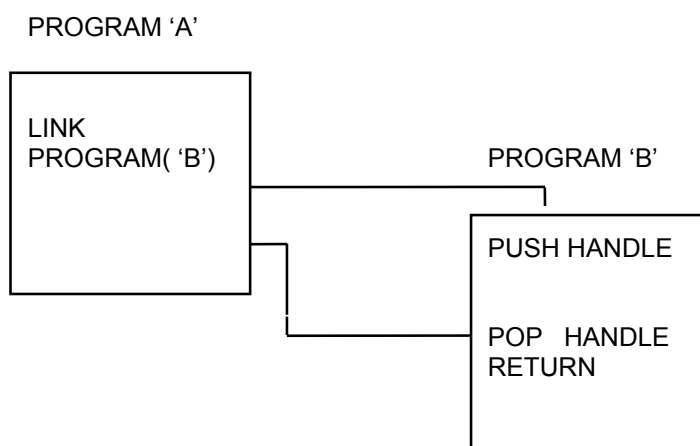
POP HANDLE

restaura os procedimentos de tratamento de exceções suspensos anteriormente.

- APLICAÇÃO:

quando um programa de aplicação é ativado via LINK e utiliza suas próprias rotinas de tratamento de condições de exceção.

EXEMPLO:



TRATAMENTO DE TECLAS FUNCIONAIS

- teclas PF (dados + byte AID)
- teclas PA ou CLEAR (byte AID)
- testando o byte de AID após o comando RECEIVE:

IF EIBAID IS EQUAL DFHENTER THEN _____

- via HANDLE AID

formato:

```
EXEC CICS HANDLE AID option (label)
              option (label option (label) _____)
END-EXEC
```

option:

ENTER

CLEAR

PF1 A PF24

PA1 A PA3

ANYKEY todas menos ENTER

TRATAMENTO DE TECLAS FUNCIONAIS

EXEMPLO DE HANDLE AID

```
EXEC CICS  HANDLE AID PA1 (LAB1)
           ANYKEY (LAB2) PF10 (LAB3)
           END-EXEC
```

OBS:

- se uma rotina não é especificada para uma determinada tecla o controle é passado para a instrução seguinte ao **RECEIVE**.
- podem ser colocados vários **HANDLE AIDs** no programa.
- **HANDLE AIDs** podem ser desativados via “**PUSH HANDLE**” e restaurados via “**POP HANDLE**”.

NOTAS:

TRATAMENTO DE TECLAS FUNCIONAIS

OBSERVAÇÕES GERAIS SOBRE HANDLE E CONDITION/HANDLE AID

- o comando **HANDLE AID** tem precedência sobre o comando **HANDLE CONDITION**.

EXEMPLO:

```
EXEC CICS HANDLE CONDITION MAPFAIL (FALHOU)
END-EXEC
```

```
EXEC CICS HANDLE AID PF (ROTPF1)
END-EXEC
```

```
EXEC CICS SEND MAP (-----) MAPSET (-----)
MAPONLY END-EXEC
```

```
EXEC CICS RECEIVE MAP (-----) MAPSET (-----)
INTO (--) END-EXEC
```

OBS:

se o operador apertar **PF1** sem digitar nada, o controle do programa é passado para **ROTPF1** e não para **FALHOU**.

NOTAS:

TRANSAÇÕES ÚTEIS A INSTALAÇÃO

- CEDF
- CMAC
- CEBR
- CECI / CECS

TRANSAÇÃO CEDF

CEDF xxxx

TRANSAÇÃO: **xxxx** PROGRAMA: **yyyyyy**
STATUS: **PROGRAM INITIATION**

EIBTIME = **105840**
EIBDATE = **95100**

ENTER = CONTINUA
PF7 = TELA ANTERIOR
PF8 = TELA ATUAL

ENTER

TRANSAÇÃO: **xxxx** PROGRAMA: **yyyyyy**
STATUS: **ABOUT TO EXECUTE COMMAND**

EXEC CICS SEND MAP
MAP('MAPA1')
MAPSET('MAPSET1')
ERASE

ENTER = CONTINUE PF5= WORKING STORAGE
PF12 = ABEND USER TASK

ENTER

TRANSAÇÕES ÚTEIS A INSTALAÇÃO, (continuação)

MAPA EXEMPLO

NOME:

ENDEREÇO:

ENTER

TRANSAÇÃO: **** PROGRAMA: yyyyyy
STATUS: **COMMAND EXECUTION COMPLETE**

RESPONSE: **NORMAL**
PF5= WORKING STORAGE
PF12 = ABEND USER TASK

ENTER

TRANSAÇÃO: xxxx PROGRAMA: yyyyyy
STATUS: **ABOUT TO EXECUTE COMMAND**

EXEC CICS RECEIVE MAP
 MAP('MAPA1')
 MAPSET('MAPSET1')
 INTO (AREA1)

ENTER - CONTINUE PF5 - WORKING STORAGE
PF12 - ABEND USER TASK PF3 - END EDF SESSION

ENTER

TRANSAÇÕES ÚTEIS A INSTALAÇÃO

MAPA EXEMPLO

NOME: João da Silva

ENDEREÇO: Rua São Paulo, 100

ENTER

TRANSAÇÃO: **** PROGRAMA: yyyyyy
STATUS: **COMMAND EXECUTION COMPLETE**

INTO (JOÃO DA SILVA RUA ITARARÉ, 100)

RESPONSE: **NORMAL**
PF5= WORKING STORAGE
PF12 = ABEND USER TASK PF3 - END EDF SESSION

PF3

CEDF PERMITE:

- localizar qualquer área de memória, digitando o endereço da mesma.
- alterar o conteúdo de qualquer área da WORKING-STORAGE.
- apertando ENTER, volta-se ao DISPLAY original.
- através do uso da função STOP CONDITIONS, (PF9), interromper a execução somente na ocorrência da condição especificada.
- depurar uma transação em qualquer terminal da rede, a partir de um único terminal.

TRANSAÇÕES ÚTEIS A INSTALAÇÃO, (continuação)

TRANSAÇÃO CEBR

- PERMITE:
 - listar o conteúdo de uma fila TS
 - deletar uma fila de TS
 - copiar uma fila de TS para TD
 - copiar uma fila de TD para TS.
- EXEMPLOS:

1. Listar a fila TS: NCTS01

CEBR

```
CEBR TS QUEUE
ENTER COMMAND          QUEUE NCTS01
```

ENTER

```
CEBR TS QUEUE .....
ENTER COMMAND=
***** TOP OF QUEUE *****
ITEM 1
ITEM 2
ITEM 3
*** BOTTOM OF QUEUE ***
```

OBS. a fila listada NCTS01 se torna a fila corrente.

TRANSAÇÕES ÚTEIS A INSTALAÇÃO, (continuação)

2. Comando PURGE

Elimina a fila de TS corrente

3. Comando PUT DESTID

Copia a fila de TS corrente na fila TD indicada por DESTID

4. Comando GET DESTID

Copia a fila de TD especificada para a fila de TS corrente.

TRANSAÇÕES CECS/CECI

- A transação CECS permite verificar a sintaxe de qualquer comand CICS.
- A transação CECI permite verificar a sintaxe de qualquer comando CICS e, em seguida, **executá-lo**.

OBSERVAÇÕES:

- A transação CECI deve ter o acesso controlado pela instalação
- Chave de segurança via SIGN ON
- PERMITs do RACF somente para usuários habilitados.

NOTAS:

ANEXO 1 ATRIBUTOS E CÓDIGOS DE RETORNO

COPY DFHBMSCA

NOME	N/P	PIC	ITEM	FSET	FORMATO BMS	LITERAL
DFHBMUNP	N	ALF	NORMAL	NAO	UNPROT	space
DFHBMFSE	N	ALF	NORMAL	SIM	UNPROT, FSET	A
DFHBMBVR	N	ALF	BRILHO	NAO	UNPROT, BRT	H
DFHUNIND	N	ALF	BRILHO	SIM	UNPROT, BRT, FSET	I
DFHBMNDAR	N	ALF	DARK	NAO	UNPROT, DRK	<
DFHUNNOD	N	ALF	DARK	SIM	UNPROT, DRK, FSET	(
DFHBMUNN	N	NUM	NORMAL	NAO	UNPROT, NUM	&
DFHUNNUM	N	NUM	NORMAL	SIM	UNPROT, NUM, FSET	J
DFHBMUNB	N	NUM	BRILHO	NAO	UNPROT, NUM, BRT	Q
DFHUNINT	N	NUM	BRILHO	SIM	UNPROT, NUM, BRT, FSET	R
	N	NUM	DARK	NAO	UNPROT, NUM, DRK	*
DFHUNNON	N	NUM	DARK	SIM	UNPROT, NUM, DRK, FSET)
DFHBMPRO	P	ALF	NORMAL	NAO	PROT	-
DFHBMPRF	P	ALF	NORMAL	SIM	PROT, FSET	/
DFHPROTI	P	ALF	BRILHO	NAO	PROT, BRT	Y
	P	ALF	BRILHO	SIM	PROT, BRT, FSET	Z
DFHPROTN	P	ALF	DARK	NAO	PROT, DRK	%
	P	ALF	DARK	SIM	PROT, DRK, FSET	_
DFHBMASK	P	ASK	NORMAL	NAO	ASKIP	0
DFHBMASF	P	ASK	NORMAL	SIM	ASKIP, FSET	1
DFHBMASB	P	ASK	BRILHO	NAO	ASKIP, BRT	8
	P	ASK	BRILHO	SIM	ASKIP, BRT, FSET	9
	P	ASK	DARK	NAO	ASKIP, DRK	@
	P	ASK	DARK	SIM	ASKIP, DRK, FSET	'

CÓDIGOS DE RETORNO

COD. RET.	CONDICAO
01	ERROR
11	TERMIDERR
12	DSIDERR
13	NOTFND
14	DUPREC
15	DUPKEY
16	INVREQ
17	IOERR
18	NOSPACE
19	NOTOPEN
20	ENDFILE
21	ILLOGIC
22	LENGERR
23	QZERO
26	ITEMERR
27	PGMIDERR
36	MAPFAIL
38	INVMPSZ

MANUAIS PARA REFERÊNCIA

CICS APPLICATION PROGRAMMING PRIMER
SC33-0674

CICS APPLICATION PROGRAMMING REFERENCE
SC33-1170

CICS MESSAGES AND CODES
SC33-1177

Veja também:


<http://www.raleigh.ibm.com/cgi-bin/bookmgr/>

CICS0020 - PROGRAMAÇÃO COMMAND LEVEL

EXERCÍCIO DE FIXAÇÃO 1

Escrever um programa NÃO CONVERSACIONAL que receba a MENSAGEM 1 abaixo e realize o processamento indicado.

MENSAGEM 1: **Txxx xx.....xx**




texto variável
cód. da transação

tamanho: 25

PROCESSAMENTO:

Receber a mensagem teclada pelo usuário e responder conforme MENSAGEM 2 abaixo e, encerrar a transação.

MENSAGEM 2: **MENSAGEM: xx.....xx**



texto variável
constante

Caso a MENSAGEM 1 recebida tenha mais que 25 caracteres, enviar a MENSAGEM 3 e, encerrar a transação.

MENSAGEM 3: **ERRO DE TAMANHO. REENTRE TODA A MENSAGEM.**

TRANSAÇÕES: TC1A à TC1J
 TA1A à TA1J

PROGRAMAS: PCOB01A à PCOB01J
 PASM01A à PASM01J

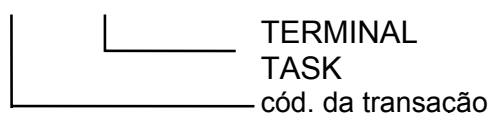
OBSERVAÇÃO: Escrever os programas em COBOL e ASSEMBLER
 Utilize a transação CEDF para depurar eventuais problemas.

CICS0020 - PROGRAMAÇÃO COMMAND LEVEL

EXERCÍCIO DE FIXAÇÃO 2

Escrever um programa CONVERSACIONAL que receba a MENSAGEM 1 abaixo e realize o processamento indicado.

MENSAGEM 1: **Txxx variável**



PROCESSAMENTO:

1. Se receber TERMINAL, enviar a MENSAGEM 2 e, encerrar a transação.

MENSAGEM 2: SEU TERMINAL É O xxxx.

2. Se receber TASK, enviar a MENSAGEM 3 e, encerrar a transação.

MENSAGEM 3: O NÚMERO DA TASK É xx.xxx

3. Qualquer variável diferente de TERMINAL/TASK enviar a MENSAGEM 4 e, aguardar o envio de variável válida.

MENSAGEM 4: VARIÁVEL DESCONHECIDA. REENTRE.

TRANSAÇÕES: TC2A à TC2J
 TA2A à TA2J

PROGRAMAS: PCOB02A à PCOB02J
 PASM02A à PASM02J

OBSERVAÇÃO: Utilize a transação CEDF para depurar eventuais problemas.

EXERCÍCIO DE FIXAÇÃO 3

Transformar o programa Pxxx02x em pseudo-conversacional.

TRANSAÇÕES: TC3A à TC3J
 TA3A à TA3J

PROGRAMAS: PCOB03A à PCOB03J
 PASM03A à PASM03J

OBSERVAÇÃO: Utilize a transação CEDF para depurar eventuais problemas.

CICS0020 - PROGRAMAÇÃO COMMAND LEVEL

EXERCÍCIO DE FIXAÇÃO 4

Escrever dois programas que através do uso da COMMAREA, realizem o processamento indicado.

PROGRAMA 1: Ao receber a MENSAGEM 1 chamar via LINK o PROGRAMA 2 transferindo uma COMMAREA de 16 posições para que o programa “linkado” devolva a data e a hora do dia no formato **dd/mm/aa e hh:mm:ss**. Ao receber as informações envia-las conforme a MENSAGEM 2 e, encerrar o processamento.

PROGRAMA 2: Ao receber o controle do PROGRAMA 1, formatar a data e hora, conforme descrito no programa e retorna-las através da COMMAREA recebida.

MENSAGEM 1: **Txxx** – cód. transação

MENSAGEM 2: **DATA E HORA DO PROCESSAMENTO: dd/mm/aa AS hh:mm:ss**

TRANSAÇÕES: TC4A à TC4J

PROGRAMAS: PCOB04A à PCOB04J
PCOBL4A à PCOBL4J

OBSERVAÇÃO: Utilize a transação CEDF para depurar eventuais problemas.

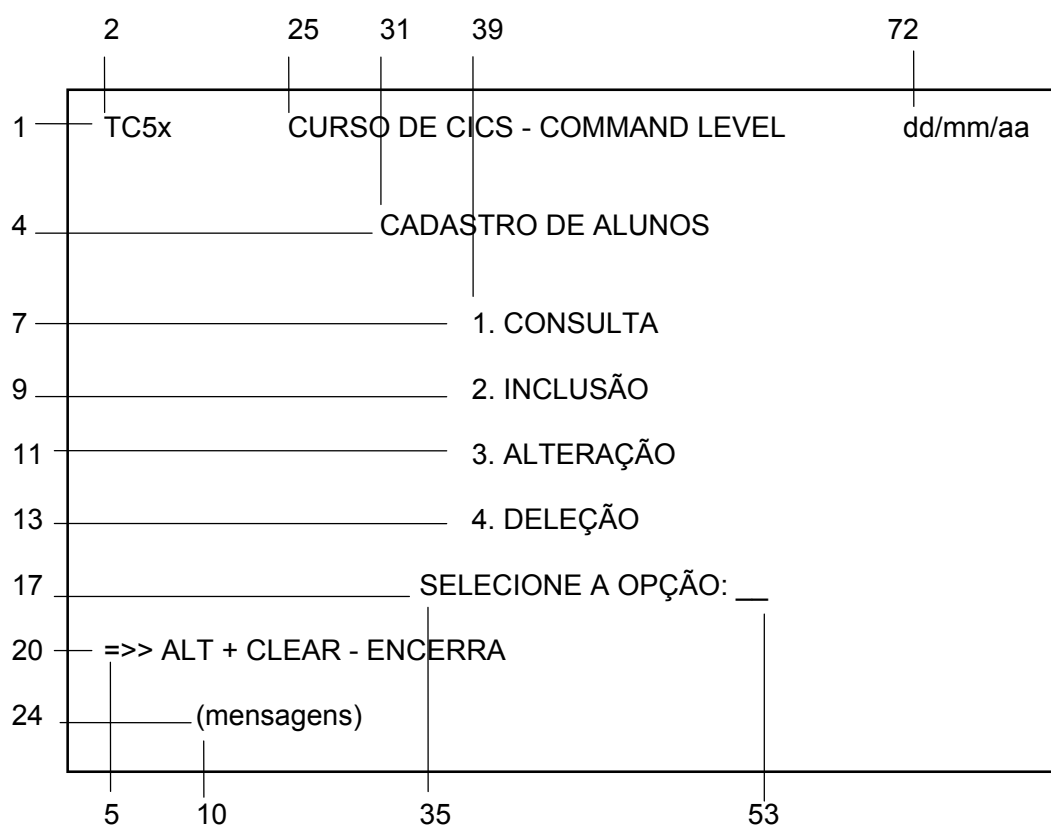
CICS0020 - PROGRAMAÇÃO COMMAND LEVEL

EXERCÍCIOS DE FIXAÇÃO 5

MAPSET: MPCUR1A à MPCUR1J

MAPAS: MAPA1 e MAPA2

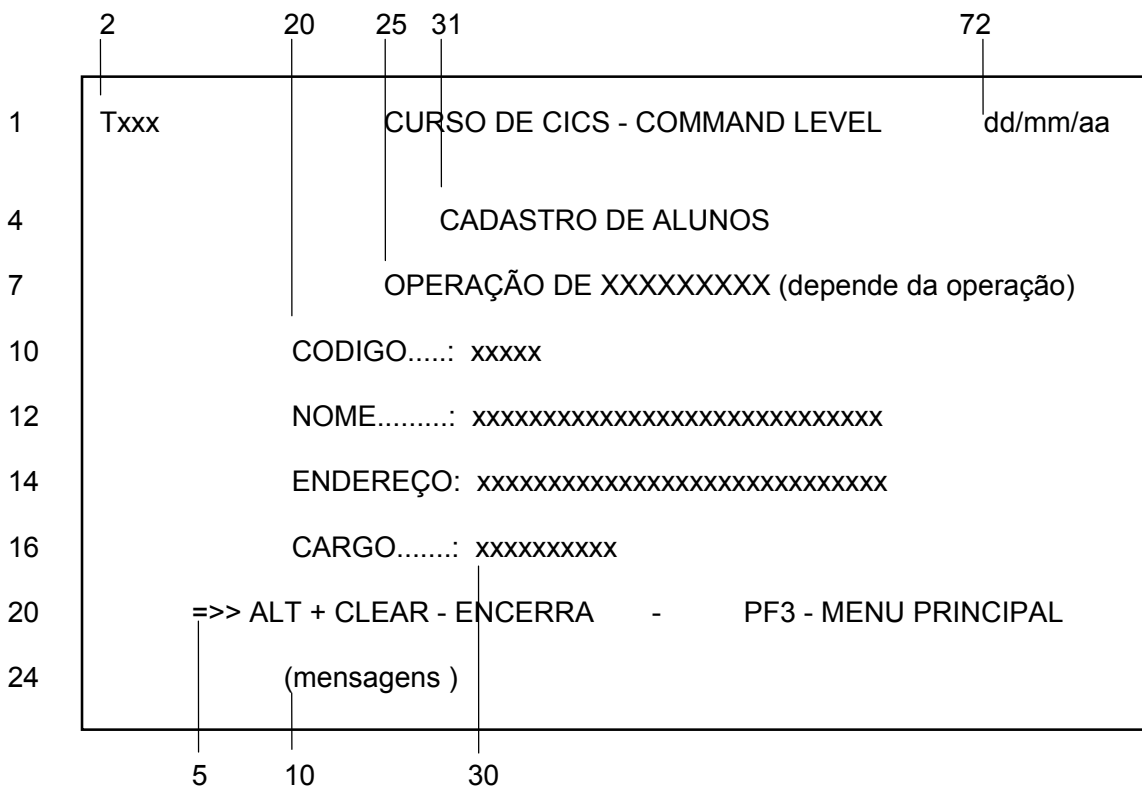
MAPA1



mensagem: **OPÇÃO INVÁLIDA REENTRE.**

CICS0020 - PROGRAMAÇÃO COMMAND LEVEL,

MAPA2



mensagens: **ALUNO NÃO CADASTRADO**
ALUNO JÁ CADASTRADO
PRESSIONE PF4 P/CONFIRMAR
CONFIRME A OPERAÇÃO
OPERAÇÃO REALIZADA

outras necessárias...

CICS0020 - PROGRAMAÇÃO COMMAND LEVEL

ARQUIVO: CADCURSA à CADCURSF **VSAM-KSDS**

LENGTH..: 80

KEYID.....: CODIGO

KEYLEN..: 5 **POSICÃO:** 0-4

LAYOUT

CODIGO	NOME	ENDEREÇO	CARGO	
05	30	30	10	05

CODIGO (<i>chave</i>)	PIC	9(05).
NOME	PIC	X(30).
ENDEREÇO	PIC	X(30).
CARGO	PIC	X(10).
FILLER	PIC	X(05).

TRANSAÇÕES/PROGRAMAS

Programas COBOL

TC5A à TC5J - PCOB05A à PCOB05J - MENU
 TC6A à TC6J - PCOB06A à PCOB06J - CONSULTA
 TC7A à TC7J - PCOB07A à PCOB07J - INCLUSÃO
 TC8A à TC8J - PCOB08A à PCOB08J - ALTERAÇÃO
 TC9A à TC9J - PCOB08A à PCOB08J - DELEÇÃO

Programas Assembler

TA5A à TA5J - PASM05A à PASM05J - MENU
 TA6A à TA6J - PASM06A à PASM06J - CONSULTA
 TA7A à TA7J - PASM07A à PASM07J - INCLUSÃO
 TA8A à TA8J - PASM08A à PASM08J - ALTERAÇÃO
 TA9A à TA9J - PASM08A à PASM08J - DELEÇÃO

CICS0020 - PROGRAMAÇÃO COMMAND LEVEL

TRANSAÇÃO TC5x - PCOB05x

- Ao ser ativada enviar o mapa MAPA1.
- Testar o código da opção ou transação fornecido pelo operador e realizar a operação desejada, passando o controle via XCTL ao programa responsável pelo processamento.

TRANSAÇÃO TC6x - PCOB06x

- Ao ser ativada enviar o mapa MAPA2, indicando a operação solicitada.
- Receber o Código do Aluno, acessar o Cadastro de Alunos e enviar as informações sobre o registro recuperado;
- Quando o usuário pressionar ALT+CLEAR, retornar ao MENU PRINCIPAL.

TRANSAÇÃO TC7x - PCOB07x

- Ao ser ativada enviar o mapa MAPA2, indicando a operação solicitada.
- Receber e consistir as informações recebidas.
- Testar se todos os campos foram fornecidos e, incluir o registro.
- Quando o usuário pressionar ALT+CLEAR, retornar ao MENU PRINCIPAL.

TRANSAÇÃO TC8x - PCOB08x

- Ao ser ativada enviar o mapa MAPA2, indicando a operação solicitada;
- Receber o Código do aluno fornecido e realizar a leitura do registro e enviá-lo para a alteração desejada;
- Receber e consistir as informações recebidas.
- Atualizar o registro.

TRANSAÇÃO TC9x - PCOB08x

- Ao ser ativada enviar o mapa MAPA2, indicando a operação solicitada;
- Receber o Código do aluno fornecido realizar a leitura do registro e enviá-lo para confirmação da exclusão;
- Excluir o registro.

OBSERVAÇÃO:

- Alterar os programas PCOB06x a PCOB08x para utilizar DB2
- Como as transações TC8x e TC9x utilizam o mesmo programa (PCOB08x), alterar o programa PCOB05x se necessário.

Uma solução para os exercícios propostos

EXERCÍCIO DE FIXAÇÃO 1

```

PRINT NOGEN                                00010005
MPCUR1I DFHMSD TYPE=MAP,TIOAPFX=YES,MODE=INOUT,CTRL=(FREEKB,FRSET), X00020018
LANG=COBOL                                00020118
*                                           00020215
* *****                                00021015
* * MENU PRINCIPAL - MAPA 1 *                00021115
* *****                                00021215
*                                           00022015
MAPA1 DFHMDI SIZE=(24,80),LINE=1,COLUMN=1  00030002
DFHMDF POS=(1,1),LENGTH=4,ATTRB=PROT,INITIAL='TC5I' 00040002
DFHMDF POS=(1,25),LENGTH=29,ATTRB=PROT, X00041012
INITIAL='CURSO DE CICS - COMMAND LEVEL' 00041111
DATAM1 DFHMDF POS=(1,71),LENGTH=8,ATTRB=ASKIP 00043019
DFHMDF POS=(4,31),LENGTH=18,ATTRB=PROT, X00044014
INITIAL='CADASTRO DE ALUNOS' 00045009
DFHMDF POS=(7,34),LENGTH=11,ATTRB=PROT, X00046014
INITIAL='1. CONSULTA' 00047009
DFHMDF POS=(9,34),LENGTH=11,ATTRB=PROT, X00048014
INITIAL='2. INCLUSAO' 00049009
DFHMDF POS=(11,34),LENGTH=12,ATTRB=PROT, X00049114
INITIAL='3. ALTERACAO' 00049209
DFHMDF POS=(13,34),LENGTH=12,ATTRB=PROT, X00049314
INITIAL='4. DELECAO' 00049409
DFHMDF POS=(17,30),LENGTH=18,ATTRB=PROT, X00049514
INITIAL='SELECIONE A OPCAO:' 00049609
OPCAM1 DFHMDF POS=(17,49),LENGTH=1,ATTRB=(UNPROT,IC) 00049714
DFHMDF POS=(17,51),ATTRB=ASKIP 00049814
DFHMDF POS=(20,05),LENGTH=25,ATTRB=(PROT,BRT), X00049907
INITIAL='=>> PAUSE BREAK - ENCERRA' 00050009
MENSM1 DFHMDF POS=(24,10),LENGTH=50 00051004
*                                           00051115
* *****                                00051215
* * DADOS DO ALUNO - MAPA 2 *                00051315
* *****                                00051415
*                                           00051515
*                                           00052015
MAPA2 DFHMDI SIZE=(24,80),LINE=1,COLUMN=1  00053022
TRANM2 DFHMDF POS=(1,1),LENGTH=4,ATTRB=(UNPROT,BRT),INITIAL='TC6I' 00053123
DFHMDF POS=(1,6),ATTRB=ASKIP 00053123
DFHMDF POS=(1,25),LENGTH=29,ATTRB=PROT, X00054015
INITIAL='CURSO DE CICS - COMMAND LEVEL' 00055015
DATAM2 DFHMDF POS=(1,71),LENGTH=8,ATTRB=PROT 00056015
DFHMDF POS=(4,31),LENGTH=18,ATTRB=PROT, X00057015
INITIAL='CADASTRO DE ALUNOS' 00058015
DFHMDF POS=(7,30),LENGTH=11,ATTRB=PROT, X00059017
INITIAL='OPERACAO DE' 00059115
OPERM2 DFHMDF POS=(7,42),LENGTH=09,ATTRB=PROT,INITIAL='CONSULTA' 00059217
LITCM2 DFHMDF POS=(10,20),LENGTH=12,ATTRB=PROT, X00059425
INITIAL='CODIGO.....' 00059515
CODAM2 DFHMDF POS=(10,33),LENGTH=05,ATTRB=(UNPROT,IC),JUSTIFY=ZERO 00059624
DFHMDF POS=(10,39),ATTRB=ASKIP 00059715
LITNM2 DFHMDF POS=(12,20),LENGTH=12,ATTRB=PROT, X00059825
INITIAL='NOME.....' 00059915
NOMEM2 DFHMDF POS=(12,33),LENGTH=30,ATTRB=(UNPROT) 00060015
DFHMDF POS=(12,64),ATTRB=ASKIP 00060115

```

```

LITEM2  DFHMD F POS=(14,20),LENGTH=12,ATTRB=PROT,          X00060225
        INITIAL='ENDERECO...!'                                00060315
ENDEM2  DFHMD F POS=(14,33),LENGTH=30,ATTRB=(UNPROT)        00060415
        DFHMD F POS=(14,64),ATTRB=ASKIP                      00060515
LITGM2  DFHMD F POS=(16,20),LENGTH=12,ATTRB=PROT,          X00060625
        INITIAL='CARGO.....!'                                00060715
CARGM2  DFHMD F POS=(16,33),LENGTH=10,ATTRB=(UNPROT)        00060816
        DFHMD F POS=(16,44),ATTRB=ASKIP                      00060915
        DFHMD F POS=(20,05),LENGTH=25,ATTRB=(PROT,BRT),     X00061715
        INITIAL='=>> PAUSE BREAK - ENCERRA'                  00061815
        DFHMD F POS=(20,31),LENGTH=25,ATTRB=(PROT,BRT),     X00061921
        INITIAL='- PF3 - MENU PRINCIPAL'                      00062020
MENSM2  DFHMD F POS=(24,10),LENGTH=50                        00062120
        DFHMSD TYPE=FINAL                                     00063020
        END                                                    00070000
    
```

```

//DEFINE EXEC PGM=IDCAMS,REGION=1M                                00067002
//SYSPRINT DD SYSOUT=*                                           00068000
//SYSIN DD *                                                      00069000
  DEL CLUSTER(NAME(TSO.CICS.INSTRU1.CADCUR)                       00070007
  DEF CLUSTER(NAME(TSO.CICS.INSTRU1.CADCUR) -                     00070107
    IXD -                                                         00071000
    REC(200)-                                                     00072002
    RECORDSIZE(80 80)-                                           00073004
    FREESPACE(5 5)-                                             00074000
    SHR(2))-                                                      00075002
  DATA(NAME(TSO.CICS.INSTRU1.CADCUR.DATA) -                    00077006
    UNIQUE -                                                      00078000
    KEYS(5 0))-                                                  00079002
  INDEX(NAME(TSO.CICS.INSTRU1.CADCUR.INDEX) -                   00080006
    UNIQUE)                                                       00081000
//INICIA EXEC PGM=IDCAMS                                         00084005
//SYSPRINT DD SYSOUT=*                                           00085005
//DD1 DD DSN=TSO.INSTRU1.CNTL(DADOVSAM),DISP=SHR                00086008
//SYSIN DD *                                                      00087005
  REPRO IFILE(DD1) ODS(TSO.CICS.INSTRU1.CADCUR) COUNT(4)       00088008
  LISTCAT ENT(TSO.CICS.INSTRU1.CADCUR) ALL                       00089006
//                                                                  00090005

```

```

IDENTIFICATION DIVISION.                                00010002
PROGRAM-ID. PCOB13A.                                    00020009
*                                                        00021009
*-----*                                              00021109
*                                                        00022009
* TESTE DOS COMANDOS SEND, RECEIVE E *                00023009
* DOS CAMPOS DO EIB. *                                00024009
*-----*                                              00024109
*                                                        00025009
*                                                        00026009
*                                                        00030002
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.                                00030107
SPECIAL-NAMES.                                       00031006
    DECIMAL-POINT IS COMMA.                            00032008
DATA DIVISION.                                        00040002
WORKING-STORAGE SECTION.                              00050002
01 PIC X(19) VALUE '**** INICIO-WORKING'.             00051009
01 TAMANHO PIC S9(4) COMP.                             00060002
01 ERRO PIC S9(8) COMP.                                00070002
01 MSG-ENTRADA.                                       00071004
    01 MSG-ENTRADA.                                    00080002
        05 PIC X(5).                                   00081002
        05 MSG-VAR PIC X(8) VALUE SPACES.             00082004
            88 TERMID VALUE 'TERMINAL'.               00083002
            88 TASK VALUE 'TASK '.                   00084003
            88 FIM VALUE 'FIM '.                      00084103
01 MSG-TERMINAL.                                       00085002
    05 PIC X(17) VALUE 'SEU TERMINAL E O '.           00086002
    05 MSG-TRM PIC X(8).                               00087002
01 MSG-TASK.                                           00088002
    05 PIC X(19) VALUE 'O NUMERO DA TASK E '.         00089002
    05 MSG-TSK PIC 99.999.                             00089102
01 MSG-FIM PIC X(20) VALUE 'TRANSACAO ENCERRADA'.     00089202
01 MSG-ERRO PIC X(31) VALUE 'VARIABEL DESCONHECIDA. REENTRE'. 00090002
01 PIC X(19) VALUE '**** FINAL-WORKING'.             00090109
                                                    00091009
PROCEDURE DIVISION.                                    00100002
                                                    00101009
INICIO.                                                00110002
    IF EIBCALEN = 0                                    00120002
        PERFORM 0010-RECEIVE-MSG-FULL                 00130002
    ELSE                                               00140002
        PERFORM 0020-RECEIVE-MSG-PART                 00150002
    END-IF.                                           00160002
    EVALUATE MSG-VAR                                  00170002
    WHEN 'TERMINAL'                                  00180005
        PERFORM 0030-MOVE-EIBTRMID                    00190002
    WHEN 'TASK '                                     00200005
        PERFORM 0040-MOVE-EIBTASKN                    00210002
    WHEN 'FIM '                                      00211005
        PERFORM 9999-ENCERRA                           00212002
    WHEN OTHER                                       00220002
        PERFORM 0050-MSG-ERRO                          00230002
    END-EVALUATE.                                     00240002
*                                                    00250002
0010-RECEIVE-MSG-FULL.                                00260002
    EXEC CICS RECEIVE INTO(MSG-ENTRADA) RESP(ERRO)    00270002
    END-EXEC.                                         00280002
    IF ERRO = DFHRESP(LENGERR) OR FIM                00290002
        PERFORM 9999-ENCERRA                           00300002
    END-IF.                                           00310002

```

0010-RECEIVE-EXIT.	00320002
EXIT.	00330002
*	00340002
0020-RECEIVE-MSG-PART.	00350002
EXEC CICS RECEIVE INTO(MSG-VAR) RESP(ERRO)	00360002
END-EXEC.	00370002
IF ERRO = DFHRESP(LENGERR) OR FIM	00380002
PERFORM 9999-ENCERRA	00390002
END-IF.	00400002
0020-RECEIVE-EXIT.	00410002
EXIT.	00420002
*	00421002
0030-MOVE-EIBTRMID.	00422002
MOVE EIBTRMID TO MSG-TRM.	00423002
EXEC CICS SEND FROM(MSG-TERMINAL) ERASE	00424002
END-EXEC.	00425002
EXEC CICS RETURN TRANSID('TC3A') COMMAREA(TAMANHO)	00425103
END-EXEC.	00425202
0030-MOVE-EXIT.	00428002
EXIT.	00429002
*	00429102
0040-MOVE-EIBTASKN.	00429202
MOVE EIBTASKN TO MSG-TSK.	00429302
EXEC CICS SEND FROM(MSG-TASK) ERASE	00429402
END-EXEC.	00429502
EXEC CICS RETURN TRANSID('TC3A') COMMAREA(TAMANHO)	00429603
END-EXEC.	00429702
0040-MOVE-EXIT.	00429802
EXIT.	00429902
*	00430002
0050-MSG-ERRO.	00440002
EXEC CICS SEND FROM(MSG-ERRO) ERASE	00450002
END-EXEC.	00460002
EXEC CICS RETURN TRANSID('TC3A') COMMAREA(TAMANHO)	00470003
END-EXEC.	00480002
0050-MSG-EXIT.	00500002
EXIT.	00510002
*	00520002
9999-ENCERRA.	00530002
EXEC CICS SEND FROM(MSG-FIM) ERASE	00540002
END-EXEC.	00550002
EXEC CICS RETURN END-EXEC.	00560002
9999-ENCERRA-EXIT.	00580002
EXIT.	00590002


```

IDENTIFICATION DIVISION.                                00010000
PROGRAM-ID. PCOBI4A.                                    00020000
*                                                         00021002
*-----*                                               00022002
* UTILIZANDO O COMANDO LINK *                           00023002
*-----*                                               00023102
*                                                         00025002
ENVIRONMENT DIVISION.                                   00030000
DATA DIVISION.                                         00040000
WORKING-STORAGE SECTION.                               00050000
                                                         00051002
01          PIC X(19) VALUE '**** INICIO-WORKING'.      00060000
                                                         00061002
01 TAMANHO   PIC S9(4) COMP.                             00070000
01 ERRO     PIC S9(8) COMP.                             00071000
01 MSG-LINK.                                         00072000
                                                         00072102
    05 DATA-DIA PIC X(8).                               00073000
    05 HORA-DIA  PIC X(8).                               00073100
                                                         00073202
01 MSG-ENTRADA PIC X(4).                               00084000
                                                         00084102
01 MSG-FIM.                                           00084200
    05          PIC X(30) VALUE 'DATA E HORA DO PROCESSAMENTO: '. 00084300
    05 DATA-FIM PIC X(8).                               00084400
    05          PIC X(4) VALUE ' AS '.                   00084500
    05 HORA-FIM  PIC X(8).                               00084600
    05          PIC X VALUE '.'.                          00084701
                                                         00084802
01          PIC X(19) VALUE '**** FIM DA WORKING'.      00084902
                                                         00085002
PROCEDURE DIVISION.                                    00086000
                                                         00086102
INICIO.                                                00087000
    PERFORM 0010-RECEIVE-MSG.                            00089000
    PERFORM 0020-LINK-PCOBI4L.                            00090000
    PERFORM 0030-ENVIA-MSG.                               00100000
    PERFORM 9999-ENCERRA.                                 00110000
*                                                         00220000
0010-RECEIVE-MSG.                                       00230000
    EXEC CICS RECEIVE INTO(MSG-ENTRADA) RESP(ERRO)        00240000
    END-EXEC.                                             00250000
    IF ERRO = DFHRESP(LENGERR)                            00260000
    PERFORM 9999-ENCERRA                                  00270000
    END-IF.                                               00280000
0010-RECEIVE-EXIT.                                       00290000
    EXIT.                                                 00300000
*                                                         00310000
0020-LINK-PCOBI4L.                                       00320000
    EXEC CICS LINK PROGRAM('PCOBI4L') COMMAREA(MSG-LINK) 00330000
    END-EXEC.                                             00340000
0020-LINK-EXIT.                                          00380000
    EXIT.                                                 00390000
*                                                         00400000
0030-ENVIA-MSG.                                          00410000
    MOVE DATA-DIA TO DATA-FIM.                          00420000
    MOVE HORA-DIA TO HORA-FIM.                            00420100
    EXEC CICS SEND FROM(MSG-FIM) ERASE                    00421000
    END-EXEC.                                             00422000
    EXEC CICS RETURN END-EXEC.                            00423000
0030-ENVIA-EXIT.                                        00425000

```

<i>CICS - Command Level</i>

EXIT.	00426000
*	00480000
9999-ENCERRA.	00490000
EXEC CICS SEND FROM(MSG-FIM) ERASE	00500000
END-EXEC.	00510000
EXEC CICS RETURN END-EXEC.	00520000
9999-ENCERRA-EXIT.	00530000
EXIT.	00540000

CICS - Command Level

IDENTIFICATION DIVISION.	00010000
PROGRAM-ID. PCOB14A.	00020000
ENVIRONMENT DIVISION.	00030000
DATA DIVISION.	00040000
WORKING-STORAGE SECTION.	00050000
01 PIC X(19) VALUE '**** INICIO-WORKING'.	00060000
01 TIME-CICS PIC 9(15) COMP-3.	00070001
01 MSG-LINK.	00072000
05 DATA-DIA PIC X(8).	00073000
05 HORA-DIA PIC X(8).	00074000
LINKAGE SECTION.	00080102
01 DFHCOMMAREA.	00080200
05 DATA-LINK PIC X(8).	00080305
05 HORA-LINK PIC X(8).	00080405
PROCEDURE DIVISION.	00081000
INICIO.	00082000
IF EIBCALEN NOT EQUAL 16	00082106
CONTINUE	00082200
ELSE	00082300
EXEC CICS ASKTIME ABSTIME(TIME-CICS) END-EXEC	00082405
EXEC CICS FORMATTIME ABSTIME(TIME-CICS)	00082503
DATESEP('/') TIMESEP(':')	00082603
DDMMYY(DATA-LINK) TIME(HORA-LINK)	00082704
END-EXEC	00082800
END-IF.	00082900
EXEC CICS RETURN END-EXEC.	00340000
9999-ENCERRA.	00350000
EXIT.	00360000

```

IDENTIFICATION DIVISION.                                00010000
PROGRAM-ID. PCOBI5A.                                    00020000
*                                                        00021025
*-----*                                              00022026
* TRANSACAO MENU *                                     00023026
*-----*                                              00024126
*                                                        00026025
*                                                        00030000
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.                                00031002
SPECIAL-NAMES.                                       00032002
    DECIMAL-POINT IS COMMA.                            00033002
DATA DIVISION.                                        00040000
WORKING-STORAGE SECTION.                              00050000
                                                    00051025
01          PIC X(19) VALUE '**** INICIO-WORKING'.     00060008
01 W-TAMANHO PIC S9(04) COMP.                          00070008
01 W-ERRO    PIC S9(08) COMP.                           00071008
01 W-DATA-ABS PIC S9(15) COMP-3.                        00071108
01 W-OPCAO-PCOB08I PIC X(01).                          00071208
01 MSG-FIM   PIC X(30) VALUE 'TRANSACAO ENCERRADA'.    00076005
01 MSG-ERRO  PIC X(30) VALUE 'OPCAO INVALIDA. REENTRE'. 00077006
*                                                    00078006
COPY DFHAID.                                         00079008
COPY DFHBMSCA.                                       00080013
COPY MPCUR1I.                                        00081009
                                                    00081225
01          PIC X(19) VALUE '**** FINAL-WORKING'.     00081325
                                                    00081425
PROCEDURE DIVISION.
INICIO.
    IF EIBRID = DFHCLEAR                                00083115
        PERFORM 9999-ENCERRA                            00083200
    END-IF.                                             00083300
    IF EIBCALEN > 0                                    00083400
        CONTINUE                                        00083500
    ELSE
        PERFORM 0010-SEND-MAPA                          00083600
        PERFORM 9910-RETURN-TRANSID                     00083700
        PERFORM 9910-RETURN-TRANSID                     00083800
    END-IF.                                             00083900
    PERFORM 0020-RECEIVE-MAPA.                          00084011
    PERFORM 0030-VALIDA-OPCAO.                          00085008
*                                                    00088000
0010-SEND-MAPA.                                       00089000
    MOVE LOW-VALUES TO MAPA10.                          00089123
    EXEC CICS ASKTIME ABSTIME(W-DATA-ABS) END-EXEC.     00089200
    EXEC CICS FORMATTIME ABSTIME(W-DATA-ABS) DDMMYY(DATAM10)
        DATESEP('/') END-EXEC.                          00089301
    MOVE DFHUNINT TO OPCAM1A.                           00089401
    EXEC CICS SEND MAP('MAPA1') MAPSET('MPCUR1I') FROM(MAPA10)
        RESP(W-ERRO) ERASE END-EXEC.                   00090011
    IF W-ERRO = DFHRESP(ERROR)                          00100010
        PERFORM 9999-ENCERRA                            00110012
    END-IF.                                             00120000
0010-SEND-EXIT.                                       00130000
EXIT.                                                 00140006
*                                                    00150000
0020-RECEIVE-MAPA.                                    00151006
    EXEC CICS RECEIVE MAP('MAPA1') MAPSET('MPCUR1I')
        INTO (MAPA1I)RESP(W-ERRO) END-EXEC.           00152006
    IF W-ERRO = DFHRESP(ERROR)                          00156011
                                                    00157011
                                                    00158012

```

MOVE MSG-ERRO TO MENSM10	00158106
PERFORM 0080-SEND-MAPA-DATAONLY	00158215
PERFORM 9910-RETURN-TRANSID	00158306
END-IF.	00158411
0020-RECEIVE-EXIT.	00159206
EXIT.	00159306
*	00160000
0030-VALIDA-OPCAO.	00170008
EVALUATE OPCAM1I	00180007
WHEN 1 PERFORM 0040-XCTL-PCOB06I-CONSULTA	00181008
WHEN 2 PERFORM 0050-XCTL-PCOB07I-INCLUSAO	00182008
WHEN 3 PERFORM 0060-XCTL-PCOB08I-ALTERACAO	00183008
WHEN 4 PERFORM 0070-XCTL-PCOB08I-EXCLUSAO	00184008
END-EVALUATE.	00190007
	00190125
MOVE MSG-ERRO TO MENSM10.	00191008
MOVE DFHBMASB TO MENSM1A.	00191117
PERFORM 0080-SEND-MAPA-DATAONLY.	00192015
PERFORM 9910-RETURN-TRANSID.	00193008
0030-VALIDA-EXIT.	00200008
EXIT.	00210000
*	00220000
0040-XCTL-PCOB06I-CONSULTA.	00230008
EXEC CICS XCTL PROGRAM('PCOB16A') END-EXEC.	00240024
0040-XCTL-EXIT.	00250008
EXIT.	00300000
*	00301008
0050-XCTL-PCOB07I-INCLUSAO.	00302008
EXEC CICS XCTL PROGRAM('PCOB17A') END-EXEC.	00303024
0050-XCTL-EXIT.	00304008
EXIT.	00305008
*	00310000
0060-XCTL-PCOB08I-ALTERACAO.	00311008
MOVE 'A' TO W-OPCAO-PCOB08I.	00311108
EXEC CICS XCTL PROGRAM('PCOB18A') COMMAREA(W-OPCAO-PCOB08I)	00312024
END-EXEC.	00312108
0060-XCTL-EXIT.	00313008
EXIT.	00314008
*	00314108
0070-XCTL-PCOB08I-EXCLUSAO.	00314208
MOVE 'E' TO W-OPCAO-PCOB08I.	00314308
EXEC CICS XCTL PROGRAM('PCOB18A') COMMAREA(W-OPCAO-PCOB08I)	00314424
END-EXEC.	00314508
0070-XCTL-EXIT.	00314615
EXIT.	00314708
*	00315008
0080-SEND-MAPA-DATAONLY.	00315116
MOVE DFHUNINT TO OPCAM1A.	00315221
EXEC CICS SEND MAP('MAPA1') MAPSET('MPCUR1I') FROM(MAPA10)	00316015
RESP(W-ERRO) DATAONLY END-EXEC.	00317015
IF W-ERRO = DFHRESP(ERROR)	00318015
PERFORM 9999-ENCERRA	00319015
END-IF.	00319115
0080-SEND-EXIT.	00319215
EXIT.	00319315
9910-RETURN-TRANSID.	00320004
EXEC CICS RETURN TRANSID('TC5I') COMMAREA(W-ERRO)	00350012
END-EXEC.	00351004
9910-RETURN-EXIT.	00360004
EXIT.	00370000
*	00380004

<i>CICS - Command Level</i>

9999-ENCERRA.	00390004
EXEC CICS SEND FROM(MSG-FIM) ERASE END-EXEC.	00391010
EXEC CICS RETURN END-EXEC.	00400005
9999-ENCERRA-EXIT.	00420005
EXIT.	00430004

```

IDENTIFICATION DIVISION.                                00010000
PROGRAM-ID. PCOB16A.                                   00020004
*                                                       00020114
*-----*                                              00021014
* TRANSACAO DE CONSULTA *                             00022014
*-----*                                              00023014
*                                                       00024014
ENVIRONMENT DIVISION.                                  00030000
CONFIGURATION SECTION.                                00031000
SPECIAL-NAMES.                                       00032000
    DECIMAL-POINT IS COMMA.                           00033000
DATA DIVISION.                                        00034000
WORKING-STORAGE SECTION.                              00035000
01          PIC X(19) VALUE '**** INICIO-WORKING'.     00036000
01 W-TAMANHO PIC S9(04) COMP.                          00037000
01 W-ERRO    PIC S9(08) COMP.                          00038000
01 W-DATA-ABS PIC S9(15) COMP-3.                       00039000
01 W-OPCAO-PCOB08I PIC X(01).                         00040000
01 MSG-OK    PIC X(30) VALUE 'OPERACAO REALIZADA'.     00050003
01 MSG-FIM   PIC X(30) VALUE 'TRANSACAO ENCERRADA'.   00051003
01 MSG-ERRO  PIC X(30) VALUE 'TECLA INVALIDA. REENTRE'. 00052006
01 MSG-ERRX  PIC X(30) VALUE 'ERRO DE ACESSO AO ARQUIVO'. 00052118
01 MSG-NOTF  PIC X(30) VALUE 'NAO CADASTRADO. REENTRE'. 00053016
*                                                       00070000
COPY DFHAID.                                         00071000
COPY DFHBMSCA.                                       00072000
COPY MPCUR11.                                        00073000
*                                                       00074000
01 AREA-ALUNO-ARQ.                                    00074103
    05 COD-ALUNO-ARQ PIC 9(05).                        00074203
    05 NOM-ALUNO-ARQ PIC X(30).                        00074303
    05 END-ALUNO-ARQ PIC X(30).                       00074403
    05 CAR-ALUNO-ARQ PIC X(10).                       00074503
    05          PIC X(05).                             00074603
*                                                       00074703
01          PIC X(19) VALUE '**** FINAL-WORKING'.     00074804
*                                                       00074904
PROCEDURE DIVISION.                                  00075003
                                                    00075121
                                                    00076000
INICIO.                                              00077000
    IF EIBAID = DFHCLEAR
        PERFORM 9999-ENCERRA                          00078000
    END-IF.
    IF EIBAID = DFHPPF3
        PERFORM 9900-XCTL-MENU-PRINCIPAL              00079000
    END-IF.
    IF EIBCALEN > 0
        CONTINUE                                       00079100
    ELSE
        PERFORM 0010-SEND-MAPA                         00079200
        PERFORM 9910-RETURN-TRANSID                   00079300
    END-IF.
                                                    00080000
    PERFORM 0020-RECEIVE-MAPA                          00081000
    PERFORM 0030-CONSULTA-CADALUNO.                   00082000
    PERFORM 0040-MONTA-TELA                           00083000
    PERFORM 0050-ENVIA-TELA-CONSULTA.                 00083100
    PERFORM 9910-RETURN-TRANSID                       00083200
*                                                       00083321
0010-SEND-MAPA.                                       00083400
    MOVE LOW-VALUES TO MAPA2O.

```

MOVE 'TC6' TO TRANM2O.	00084211
MOVE 'I' TO TRANM2A.	00084313
EXEC CICS ASKTIME ABSTIME(W-DATA-ABS) END-EXEC.	00084412
EXEC CICS FORMATTIME ABSTIME(W-DATA-ABS) DDMMYY(DATAM2O)	00084512
DATESEP('/') END-EXEC.	00084612
* MOVE ATRIBUTOS PARA OS CAMPOS DO MAPA	00084721
	00084821
	00085021
MOVE DFHUNINT TO CODAM2A.	00085121
MOVE DFHPROTI TO NOMEM2A.	00085221
MOVE DFHPROTI TO ENDEM2A.	00085321
MOVE DFHPROTI TO CARGM2A.	00085421
	00085521
MOVE 'CONSULTA' TO OPERM2O.	00085621
	00085721
EXEC CICS SEND MAP('MAPA2') MAPSET('MPCUR1I') FROM(MAPA2O)	00085821
RESP(W-ERRO) ERASE END-EXEC.	00085921
IF W-ERRO = DFHRESP(ERROR)	00086000
PERFORM 9999-ENCERRA	00087000
END-IF.	00088000
0010-SEND-EXIT.	00089000
EXIT.	00090000
*	00100000
0020-RECEIVE-MAPA.	00110000
EXEC CICS RECEIVE MAP('MAPA2') MAPSET('MPCUR1I')	00120000
INTO(MAPA2I) RESP(W-ERRO) END-EXEC.	00130009
IF W-ERRO = DFHRESP(ERROR)	00140000
MOVE MSG-ERRO TO MENSM2O	00150011
PERFORM 0050-ENVIA-TELA-CONSULTA	00151006
PERFORM 9910-RETURN-TRANSID	00152000
END-IF.	00153000
	00153121
* SE OPERADOR MUDAR A TRANSACAO, INICIA TRANSACAO SOLICITADA	00153221
	00153321
IF EIBTRNID = TRANM2I	00153410
CONTINUE	00153510
ELSE	00153610
EXEC CICS START TRANSID(TRANM2I) END-EXEC	00153710
EXEC CICS RETURN END-EXEC	00153812
END-IF.	00153910
0020-RECEIVE-EXIT.	00154000
EXIT.	00155000
*	00156000
0030-CONSULTA-CADALUNO.	00157002
MOVE CODAM2I TO COD-ALUNO-ARQ	00158102
EXEC CICS READ FILE('CADCUR02') INTO(AREA-ALUNO-ARQ)	00158219
RIDFLD(COD-ALUNO-ARQ) RESP(W-ERRO) END-EXEC	00158302
IF W-ERRO = DFHRESP(NOTFND)	00158402
MOVE MSG-NOTF TO MENSM2O	00191016
PERFORM 0050-ENVIA-TELA-CONSULTA	00192006
PERFORM 9910-RETURN-TRANSID	00193004
ELSE	00193116
IF W-ERRO = DFHRESP(NORMAL)	00193216
CONTINUE	00193317
ELSE	00193416
MOVE MSG-ERRX TO MENSM2O	00193517
PERFORM 0050-ENVIA-TELA-CONSULTA	00193617
PERFORM 9910-RETURN-TRANSID	00193717
END-IF.	00193804
0030-CONSULTA-EXIT.	00194003
EXIT.	00195000

*		00196000
0040-MONTA-TELA.		00197004
MOVE COD-ALUNO-ARQ TO CODAM2O.		00198003
MOVE NOM-ALUNO-ARQ TO NOMEM2O.		00198103
MOVE END-ALUNO-ARQ TO ENDEM2O.		00198203
MOVE CAR-ALUNO-ARQ TO CARGM2O.		00198303
MOVE MSG-OK TO MENS2O.		00198403
0040-MONTA-EXIT.		00198504
EXIT.		00198604
*		00198704
0050-ENVIA-TELA-CONSULTA.		00198804
MOVE 'I' TO TRANM2A.		00198920
EXEC CICS SEND MAP('MAPA2') MAPSET('MPCUR1I') FROM(MAPA2O)		00199003
RESP(W-ERRO) DATAONLY END-EXEC.		00199103
IF W-ERRO = DFHRESP(ERROR)		00199203
PERFORM 9999-ENCERRA		00199303
END-IF.		00199403
0050-ENVIA-EXIT.		00199504
EXIT.		00200000
*		00210004
9900-XCTL-MENU-PRINCIPAL.		00220004
EXEC CICS XCTL PROGRAM('PCOBI5A') END-EXEC.		00230004
9900-XCTL-EXIT.		00240004
EXIT.		00250004
*		00260004
9910-RETURN-TRANSID.		00319004
EXEC CICS RETURN TRANSID('TC6I') COMMAREA(W-ERRO)		00320004
END-EXEC.		00330000
9910-RETURN-EXIT.		00340000
EXIT.		00350000
*		00360000
9999-ENCERRA.		00370000
EXEC CICS SEND FROM(MSG-FIM) ERASE END-EXEC.		00380000
EXEC CICS RETURN END-EXEC.		00390000
9999-ENCERRA-EXIT.		00400000
EXIT.		00410000

```

IDENTIFICATION DIVISION.                                00010000
PROGRAM-ID. PCOB17A.                                   00020001
*                                                       00020100
*-----*                                              00020200
*  TRANSACAO DE INCLUSAO *                             00020301
*-----*                                              00020400
*                                                       00020500
ENVIRONMENT DIVISION.                                  00020600
CONFIGURATION SECTION.                                00020700
SPECIAL-NAMES.                                       00020800
    DECIMAL-POINT IS COMMA.                           00020900
DATA DIVISION.                                        00021000
WORKING-STORAGE SECTION.                              00022000
01          PIC X(19) VALUE '**** INICIO-WORKING'.     00023000
01 W-TAMANHO PIC S9(04) COMP.                          00024000
01 W-ERRO    PIC S9(08) COMP.                          00025000
01 W-DATA-ABS PIC S9(15) COMP-3.                       00026000
01 W-KEY     PIC X(03) VALUE SPACES.                   00027007
01 MSG-OK    PIC X(30) VALUE 'OPERACAO REALIZADA'.    00028000
01 MSG-FIM   PIC X(30) VALUE 'TRANSACAO ENCERRADA'.   00029000
01 MSG-ERRO  PIC X(30) VALUE 'TECLA INVALIDA. REENTRE'. 00030000
01 MSG-ERRO2 PIC X(30) VALUE 'REENTRE OS CAMPOS INDICADOS'. 00031001
01 MSG-ERRO3 PIC X(30) VALUE 'ALUNO JA CADASTRADO'.   00031102
01 MSG-ERRO4 PIC X(30) VALUE 'ERRO NO ACESSO AO ARQUIVO'. 00031202
01 MSG-ERRO5 PIC X(30) VALUE 'REENTRE CAMPOS DESEJADOS'. 00031312
01 MSG-PF4   PIC X(30) VALUE 'PRESSIONE PF4 P/CONFIRMAR'. 00032001
*                                                       00040000
COPY DFHAID.                                         00050000
COPY DFHBMSCA.                                       00060000
COPY MPCUR11.                                        00070000
*                                                       00071000
01 AREA-ALUNO-ARQ.                                   00072000
    05 COD-ALUNO-ARQ PIC 9(05).                        00073000
    05 NOM-ALUNO-ARQ PIC X(30).                        00074000
    05 END-ALUNO-ARQ PIC X(30).                       00074100
    05 CAR-ALUNO-ARQ PIC X(10).                       00074200
    05          PIC X(05).                             00074300
*                                                       00074400
01          PIC X(19) VALUE '**** FINAL-WORKING'.     00074500
*                                                       00074600
LINKAGE SECTION.                                      00074710
01 DFHCOMMAREA.                                       00074810
    05 L-KEY    PIC X(03).                             00074910
*                                                       00075008
PROCEDURE DIVISION.                                  00075108
INICIO.                                               00075208
    IF EIBAID = DFHCLEAR                               00075308
        PERFORM 9999-ENCERRA                          00075408
    END-IF.                                            00076000
    IF EIBAID = DFHPPF3                                00077000
        PERFORM 9900-XCTL-MENU-PRINCIPAL              00078000
    END-IF.                                            00079000
    IF EIBCALEN > 0                                    00080000
        CONTINUE                                       00081000
    ELSE                                               00082000
        PERFORM 0010-SEND-MAPA                        00083000
        PERFORM 9910-RETURN-TRANSID                  00083100
    END-IF.                                            00083200
    PERFORM 0020-RECEIVE-MAPA.                        00083317
    IF EIBAID = DFHPPF4 AND L-KEY = 'PF4'             00083408
        PERFORM 0050-INCLUI-REGISTRO                  00083501

```

ELSE	00083601
PERFORM 0030-CONSISTE-DADOS	00083902
END-IF.	00084002
PERFORM 0040-ENVIAR-MAPA.	00084102
PERFORM 9910-RETURN-TRANSID.	00084202
*	00084302
0010-SEND-MAPA.	00084402
MOVE LOW-VALUES TO MAPA2O.	00084502
MOVE 'TC7I' TO TRANM2O.	00084602
MOVE 'I' TO TRANM2A.	00084702
EXEC CICS ASKTIME ABSTIME(W-DATA-ABS) END-EXEC.	00084802
EXEC CICS FORMATTIME ABSTIME(W-DATA-ABS) DDMMYY(DATAM2O)	00084902
DATESEP('/') END-EXEC.	00085002
MOVE DFHUNNUM TO CODAM2A.	00085102
MOVE DFHBMFSE TO NOMEM2A.	00085202
MOVE DFHBMFSE TO ENDEM2A.	00085302
MOVE DFHBMFSE TO CARGM2A.	00085402
MOVE 'INCLUSAO' TO OPERM2O.	00085502
EXEC CICS SEND MAP('MAPA2') MAPSET('MPCUR1I') FROM(MAPA2O)	00085602
RESP(W-ERRO) ERASE END-EXEC.	00085702
IF W-ERRO = DFHRESP(ERROR)	00085802
PERFORM 9999-ENCERRA	00085902
END-IF.	00086002
0010-SEND-EXIT.	00086102
EXIT.	00086202
*	00086302
0020-RECEIVE-MAPA.	00086402
EXEC CICS RECEIVE MAP('MAPA2') MAPSET('MPCUR1I')	00086502
INTO(MAPA2I) RESP(W-ERRO) END-EXEC.	00087000
IF W-ERRO = DFHRESP(ERROR)	00088020
MOVE MSG-ERRO TO MENSM2O	00089000
PERFORM 0040-ENVIAR-MAPA	00090003
PERFORM 9910-RETURN-TRANSID	00100000
END-IF.	00110000
IF EIBTRNID = TRANM2I	00120000
CONTINUE	00130000
ELSE	00140000
EXEC CICS START TRANSID(TRANM2I) END-EXEC	00150000
EXEC CICS RETURN END-EXEC	00151000
END-IF.	00152000
0020-RECEIVE-EXIT.	00153000
EXIT.	00154000
*	00155000
0030-CONSISTE-DADOS.	00156003
MOVE LOW-VALUES TO MENSM2O.	00156101
MOVE -1 TO CODAM2L.	00156202
IF CODAM2I = 0 OR CODAM2I NOT NUMERIC	00157005
MOVE DFHPROTI TO LITCM2A	00157115
MOVE DFHUNINT TO CODAM2A	00157215
MOVE -1 TO CODAM2L	00157315
MOVE MSG-ERRO2 TO MENSM2O	00157415
ELSE	00157515
MOVE DFHBMPRO TO LITCM2A	00157616
MOVE DFHUNNUM TO CODAM2A	00157715
END-IF.	00157815
IF CARGM2I = SPACES OR CARGM2I = LOW-VALUES	00157915
MOVE DFHUNIMD TO CARGM2A	00158015
MOVE DFHPROTI TO LITGM2A	00158115
MOVE MSG-ERRO2 TO MENSM2O	00158215
MOVE -1 TO CARGM2L	00158315
MOVE 0 TO CODAM2L	00158415

ELSE	00158511
MOVE DFHBMPRO TO LITGM2A	00158616
MOVE DFHBMFSE TO CARGM2A	00158711
END-IF.	00158811
IF ENDEM2I = SPACES OR ENDEM2I = LOW-VALUES	00158911
MOVE DFHUNIMD TO ENDEM2A	00159011
MOVE DFHPROTI TO LITEM2A	00159115
MOVE MSG-ERRO2 TO MENSM2O	00159211
MOVE -1 TO ENDEM2L	00159311
MOVE 0 TO CODAM2L	00159411
MOVE 0 TO CARGM2L	00159511
ELSE	00159611
MOVE DFHBMPRO TO LITEM2A	00159716
MOVE DFHBMFSE TO ENDEM2A	00159811
END-IF.	00159911
IF NOMEM2I = SPACES OR NOMEM2I = LOW-VALUES	00160005
MOVE DFHUNIMD TO NOMEM2A	00160101
MOVE DFHPROTI TO LITNM2A	00160216
MOVE MSG-ERRO2 TO MENSM2O	00160303
MOVE -1 TO NOMEM2L	00160402
MOVE 0 TO CODAM2L	00160509
MOVE 0 TO ENDEM2L	00160611
MOVE 0 TO CARGM2L	00160711
ELSE	00160811
MOVE DFHBMPRO TO LITNM2A	00160916
MOVE DFHBMFSE TO NOMEM2A	00161011
END-IF.	00161111
IF MENSM2I = LOW-VALUES	00161211
MOVE MSG-PF4 TO MENSM2O	00161311
MOVE DFHBMPRO TO LITCM2A	00161415
MOVE DFHBMPRO TO LITNM2A	00161515
MOVE DFHBMPRO TO LITEM2A	00161615
MOVE DFHBMPRO TO LITGM2A	00161715
MOVE 'Z' TO CODAM2A	00161815
MOVE 'Z' TO NOMEM2A	00161915
MOVE 'Z' TO ENDEM2A	00162015
MOVE 'Z' TO CARGM2A	00162115
MOVE 'PF4' TO W-KEY	00162215
END-IF.	00162315
0030-CONSISTE-EXIT.	00163015
EXIT.	00170001
*	00194000
0040-ENVIA-MAPA.	00195001
MOVE 'I' TO TRANM2A.	00195104
EXEC CICS SEND MAP('MAPA2') MAPSET('MPCUR1') FROM(MAPA2O)	00196000
RESP(W-ERRO) DATAONLY CURSOR END-EXEC.	00197002
IF W-ERRO = DFHRESP(ERROR)	00198000
PERFORM 9999-ENCERRA	00199000
END-IF.	00199100
0040-ENVIA-EXIT.	00199201
EXIT.	00199300
*	00199400
0050-INCLUI-REGISTRO.	00199502
MOVE CODAM2I TO COD-ALUNO-ARQ.	00200102
MOVE NOMEM2I TO NOM-ALUNO-ARQ.	00200202
MOVE ENDEM2I TO END-ALUNO-ARQ.	00200302
MOVE CARGM2I TO CAR-ALUNO-ARQ.	00200402
EXEC CICS WRITE FILE('CADCUR02') FROM(AREA-ALUNO-ARQ)	00200514
RIDFLD(COD-ALUNO-ARQ) RESP(W-ERRO)	00200602
END-EXEC.	00200702
IF W-ERRO = DFHRESP(NORMAL)	00200802

OR W-ERRO = DFHRESP(INVREQ)	00200921
MOVE MSG-OK TO MENSM2O	00201002
MOVE -1 TO CODAM2L	00201102
MOVE DFHUNNUM TO CODAM2A	00201202
MOVE DFHBMFSE TO NOMEM2A	00201302
MOVE DFHBMFSE TO ENDEM2A	00201402
MOVE DFHBMFSE TO CARGM2A	00201502
ELSE	00201602
IF W-ERRO = DFHRESP(DUPREC)	00201702
MOVE MSG-ERRO3 TO MENSM2O	00201802
ELSE	00201902
MOVE MSG-ERRO4 TO MENSM2O	00202002
END-IF	00202102
END-IF.	00202202
0050-INCLUI-EXIT.	00202302
EXIT.	00202402
*	00202502
9900-XCTL-MENU-PRINCIPAL.	00202602
EXEC CICS XCTL PROGRAM('PCOB15A') END-EXEC.	00202702
9900-XCTL-EXIT.	00202802
EXIT.	00202902
*	00203002
9910-RETURN-TRANSID.	00204002
EXEC CICS RETURN TRANSID('TC7I') COMMAREA(W-KEY)	00210001
END-EXEC.	00220000
9910-RETURN-EXIT.	00230000
EXIT.	00240000
*	00250000
9999-ENCERRA.	00260000
EXEC CICS SEND FROM(MSG-FIM) ERASE END-EXEC.	00270000
EXEC CICS RETURN END-EXEC.	00280000
9999-ENCERRA-EXIT.	00290000
EXIT.	00300000

```

IDENTIFICATION DIVISION.                                00010000
PROGRAM-ID. PCOB18A.                                    00020000
*                                                        00030000
*-----*
* TRANSACAO DE ALTERACAO E EXCLUSAO *                  00040000
*-----*
*                                                        00050000
*                                                        00060000
*                                                        00070000
ENVIRONMENT DIVISION.                                   00080000
CONFIGURATION SECTION.                                  00090000
SPECIAL-NAMES.                                         00100000
    DECIMAL-POINT IS COMMA.                             00110000
DATA DIVISION.                                         00120000
WORKING-STORAGE SECTION.                               00130000
    01 PIC X(19) VALUE '**** INICIO-WORKING'.           00140000
    01 W-TAMANHO PIC S9(04) COMP.                        00141003
    01 W-ERRO PIC S9(08) COMP.                            00150000
    01 W-DATA-ABS PIC S9(15) COMP-3.                     00160000
    01 W-COMMAREA.                                       00170000
    05 W-OPERACAO PIC X(01) VALUE SPACE.                 00170103
    05 W-KEY PIC X(03) VALUE SPACES.                    00171105
    05 W-REGISTRO PIC X(80) VALUE SPACES.                00172003
    01 MSG-OK PIC X(30) VALUE 'OPERACAO REALIZADA'.      00173003
    01 MSG-FIM PIC X(30) VALUE 'OPERACAO REALIZADA'.    00174003
    01 MSG-ERRO PIC X(30) VALUE 'TRANSACAO ENCERRADA'.  00190000
    01 MSG-ERRO2 PIC X(30) VALUE 'TECLA INVALIDA. REENTRE'. 00200000
    01 MSG-ERRO3 PIC X(30) VALUE 'REENTRE OS CAMPOS INDICADOS'. 00210000
    01 MSG-ERRO4 PIC X(30) VALUE 'ALUNO JA CADASTRADO'.  00220000
    01 MSG-ERRO5 PIC X(30) VALUE 'ERRO NO ACESSO AO ARQUIVO'. 00230000
    01 MSG-ERRO6 PIC X(30) VALUE 'ALTERE OS CAMPOS DESEJADOS'. 00240000
    01 MSG-NOTF PIC X(30) VALUE 'OPERACAO NAO REALIZADA'. 00250002
    01 MSG-PF4 PIC X(30) VALUE 'ALUNO NAO CADASTRADO. REENTRE'. 00251001
    * PIC X(30) VALUE 'PRESSIONE PF4 P/CONFIRMAR'.       00260001
    *                                                        00270001
COPY DFHAID.                                           00280001
COPY DFHBMSCA.                                         00290001
COPY MPCUR11.                                          00300001
*                                                        00310001
*                                                        00320001
01 AREA-ALUNO-ARQ.                                     00330001
    05 COD-ALUNO-ARQ PIC 9(05).                          00330001
    05 NOM-ALUNO-ARQ PIC X(30).                          00340000
    05 END-ALUNO-ARQ PIC X(30).                          00350000
    05 CAR-ALUNO-ARQ PIC X(10).                          00360000
    05 PIC X(05).                                         00370000
*                                                        00380000
*                                                        00390000
01 PIC X(19) VALUE '**** FINAL-WORKING'.               00400000
*
LINKAGE SECTION.                                       00410000
01 DFHCOMMAREA.                                       00420000
    05 L-OPERACAO PIC X(01).                              00430000
    88 ALTERACAO VALUE 'A'.                              00440000
    88 EXCLUSAO VALUE 'E'.                              00450005
    05 L-KEY PIC X(03).                                  00460005
    05 L-REGISTRO PIC X(80).                             00470001
*                                                        00471006
*                                                        00480000
PROCEDURE DIVISION.                                    00490000
INICIO.                                                00500000
    IF EIBRID = DFHCLEAR OR EIBCALEN = 0                00510000
        PERFORM 9999-ENCERRA                             00520000

```

```

END-IF. 00530000
00531003
IF EIBAID = DFHPF3 00540000
  PERFORM 9900-XCTL-MENU-PRINCIPAL 00550000
END-IF. 00560000
00561003
MOVE DFHCOMMAREA TO W-COMMAREA. 00562007
00563007
00570000
IF EIBCALEN > 1
  CONTINUE 00580000
ELSE
  PERFORM 0010-SEND-MAPA 00590000
  PERFORM 9910-RETURN-TRANSID 00600000
  PERFORM 9910-RETURN-TRANSID 00610000
END-IF. 00620000
00620103
PERFORM 0020-RECEIVE-MAPA 00621003
00622003
IF L-KEY = 'ALT' 00630004
  PERFORM 0080-CONSISTE-ALTERACAO 00640004
ELSE 00650000
  IF EIBAID = DFHPF4 AND L-KEY = 'PF4' 00680204
    PERFORM 0050-ALTERA-INCLUI 00680304
  ELSE 00680404
    PERFORM 0030-CONSULTA-CADALUNO 00680504
  END-IF 00680604
END-IF. 00680704
00681003
PERFORM 0040-ENVIA-MAPA. 00690001
PERFORM 9910-RETURN-TRANSID. 00700001
* 00710001
0010-SEND-MAPA. 00720001
MOVE LOW-VALUES TO MAPA2O. 00730001
MOVE 'TC8I' TO TRANM2O. 00740001
MOVE 'I' TO TRANM2A. 00750001
EXEC CICS ASKTIME ABSTIME(W-DATA-ABS) END-EXEC. 00760001
EXEC CICS FORMATTIME ABSTIME(W-DATA-ABS) DDMMYY(DATAM2O) 00770001
  DATESEP('/') END-EXEC. 00780001
MOVE DFHUNNUM TO CODAM2A. 00790001
MOVE DFHPROTI TO NOMEM2A. 00800001
MOVE DFHPROTI TO ENDEM2A. 00810001
MOVE DFHPROTI TO CARGM2A. 00820001
IF ALTERACAO 00830001
  MOVE 'ALTERACAO' TO OPERM2O 00840000
ELSE 00850000
  MOVE 'EXCLUSAO' TO OPERM2O 00860000
END-IF. 00870000
EXEC CICS SEND MAP('MAPA2') MAPSET('MPCUR1I') FROM(MAPA2O) 00880000
  RESP(W-ERRO) ERASE END-EXEC. 00890000
IF W-ERRO = DFHRESP(ERROR) 00900000
  PERFORM 9999-ENCERRA 00910000
END-IF. 00920000
0010-SEND-EXIT. 00930000
EXIT. 00940000
00950000
* 00960000
0020-RECEIVE-MAPA. 00970000
EXEC CICS RECEIVE MAP('MAPA2') MAPSET('MPCUR1I') 00970000
  INTO(MAPA2I) RESP(W-ERRO) END-EXEC. 00980000
IF W-ERRO = DFHRESP(ERROR) 00990000
  MOVE MSG-ERRO TO MENSM2O 01000000
  PERFORM 0040-ENVIA-MAPA 01010000
  PERFORM 9910-RETURN-TRANSID 01020000

```

```

END-IF.                                01030000
IF EIBTRNID = TRANM2I                  01040000
  CONTINUE                              01050000
ELSE                                    01060000
  EXEC CICS START TRANSID(TRANM2I) END-EXEC 01070000
  EXEC CICS RETURN END-EXEC            01080000
END-IF.                                01090000
0020-RECEIVE-EXIT.                    01190000
EXIT.                                   01200000
*                                       01210000
0030-CONSULTA-CADALUNO.                01220001
MOVE CODAM2I TO COD-ALUNO-ARQ          01230003
EXEC CICS READ FILE('CADCUR02') INTO(AREA-ALUNO-ARQ) 01240003
  RIDFLD(COD-ALUNO-ARQ) RESP(W-ERRO) END-EXEC 01250003
IF W-ERRO = DFHRESP(NOTFND)            01260003
  MOVE -1 TO CODAM2L                    01270014
  MOVE MSG-NOTF TO MENSM2O              01271014
  PERFORM 0040-ENVIA-MAPA                01280003
  PERFORM 9910-RETURN-TRANSID           01290003
ELSE                                     01300003
  IF W-ERRO = DFHRESP(NORMAL)           01310003
    CONTINUE                              01320003
  ELSE                                    01330003
    MOVE MSG-ERRO4 TO MENSM2O           01340003
    PERFORM 0040-ENVIA-MAPA              01350003
    PERFORM 9910-RETURN-TRANSID         01360003
  END-IF.                                01370003
  01370103

MOVE COD-ALUNO-ARQ TO CODAM2O          01370203
MOVE NOM-ALUNO-ARQ TO NOMEM2O          01370303
MOVE END-ALUNO-ARQ TO ENDEM2O          01370403
MOVE CAR-ALUNO-ARQ TO CARGM2O         01370503
                                         01370603
IF EXCLUSAO                             01371003
  MOVE 'Z' TO CODAM2A                   01372003
  MOVE 'Z' TO NOMEM2A                   01373003
  MOVE 'Z' TO ENDEM2A                   01374003
  MOVE 'Z' TO CARGM2A                   01375003
  MOVE 'PF4' TO W-KEY                   01375103
  MOVE MSG-PF4 TO MENSM2O               01375203
ELSE                                     01376003
  MOVE DFHBMFSE TO NOMEM2A              01376108
  MOVE DFHBMFSE TO ENDEM2A              01376208
  MOVE DFHBMFSE TO CARGM2A              01376308
  MOVE 'Z' TO CODAM2A                   01378010
  MOVE -1 TO NOMEM2L                     01379010
  MOVE MSG-ERRO5 TO MENSM2O             01379303
  MOVE 'ALT' TO W-KEY                   01379403
  MOVE AREA-ALUNO-ARQ TO W-REGISTRO     01379505
  END-IF.                                01379603
                                         01420003
0030-CONSULTA-EXIT.                    01430001
EXIT.                                   01440001
*                                       01450000
0040-ENVIA-MAPA.                        01460000
MOVE 'I' TO TRANM2A.                   01470000
EXEC CICS SEND MAP('MAPA2') MAPSET('MPCUR1I') FROM(MAPA2O) 01480000
  RESP(W-ERRO) DATAONLY CURSOR END-EXEC. 01490000
IF W-ERRO = DFHRESP(ERROR)              01500000
  PERFORM 9999-ENCERRA                  01510000
  END-IF.                                01520000

```


0040-ENVIAR-EXIT.	01530000
EXIT.	01540000
*	01550000
0050-ALTERAR-INCLUI.	01560001
IF ALTERACAO	01570001
PERFORM 0060-ALTERAR	01580001
ELSE	01590001
PERFORM 0070-EXCLUI	01600001
END-IF.	01610001
0050-ALT-INC-EXIT.	01620001
EXIT.	01630001
*	01640001
0060-ALTERAR.	01650001
MOVE CODAM2I TO COD-ALUNO-ARQ	01660001
EXEC CICS READ FILE('CADCUR02') INTO(AREA-ALUNO-ARQ)	01670001
RIDFLD(COD-ALUNO-ARQ) RESP(W-ERRO) UPDATE END-EXEC	01680001
IF W-ERRO = DFHRESP(NOTFND)	01681001
MOVE MSG-NOTF TO MENSM2O	01682001
PERFORM 0040-ENVIAR-MAPA	01683001
PERFORM 9910-RETURN-TRANSID	01684001
ELSE	01685001
IF W-ERRO = DFHRESP(NORMAL)	01686001
OR W-ERRO = DFHRESP(INVREQ)	01686112
CONTINUE	01687001
ELSE	01688001
MOVE MSG-ERRO4 TO MENSM2O	01689001
PERFORM 0040-ENVIAR-MAPA	01689101
PERFORM 9910-RETURN-TRANSID	01689201
END-IF.	01689301
IF L-REGISTRO NOT = AREA-ALUNO-ARQ	01689401
MOVE MSG-ERRO6 TO MENSM2O	01689501
ELSE	01689601
	01689711
MOVE NOMEM2I TO NOM-ALUNO-ARQ	01690011
MOVE ENDEM2I TO END-ALUNO-ARQ	01690211
MOVE CARGM2O TO CAR-ALUNO-ARQ	01690411
	01690611
EXEC CICS REWRITE FILE('CADCUR02')	01690712
FROM(AREA-ALUNO-ARQ) RESP(W-ERRO) END-EXEC	01690812
MOVE MSG-OK TO MENSM2O	01690911
MOVE SPACE TO W-KEY W-REGISTRO	01691011
MOVE -1 TO CODAM2L	01691112
MOVE DFHUNNUM TO CODAM2A	01691212
IF W-ERRO = DFHRESP(NORMAL)	01691312
OR W-ERRO = DFHRESP(INVREQ)	01691412
CONTINUE	01691512
ELSE	01691612
MOVE MSG-ERRO4 TO MENSM2O	01691712
END-IF	01691812
END-IF.	01691912
PERFORM 0040-ENVIAR-MAPA.	01692012
PERFORM 9910-RETURN-TRANSID.	01692112
0060-ALTERAR-EXIT.	01692212
EXIT.	01692312
*	01692412
0070-EXCLUI.	01692512
MOVE CODAM2I TO COD-ALUNO-ARQ	01692612
EXEC CICS DELETE FILE('CADCUR02')	01692712
RIDFLD(COD-ALUNO-ARQ) RESP(W-ERRO) END-EXEC	01692812
IF W-ERRO = DFHRESP(NOTFND)	01692912
MOVE MSG-NOTF TO MENSM2O	01693012

PERFORM 0040-ENVIA-MAPA	01693112
PERFORM 9910-RETURN-TRANSID	01693212
ELSE	01693312
MOVE MSG-OK TO MENSM2O	01693413
MOVE SPACE TO W-KEY W-REGISTRO	01693513
MOVE -1 TO CODAM2L	01693613
MOVE DFHUNNUM TO CODAM2A	01693713
IF W-ERRO = DFHRESP(NORMAL)	01693812
OR W-ERRO = DFHRESP(INVREQ)	01693912
CONTINUE	01694012
ELSE	01694112
MOVE MSG-ERRO4 TO MENSM2O	01694212
PERFORM 0040-ENVIA-MAPA	01694312
PERFORM 9910-RETURN-TRANSID	01694412
END-IF.	01694501
PERFORM 0040-ENVIA-MAPA.	01694601
PERFORM 9910-RETURN-TRANSID.	01694701
0070-EXCLUI-EXIT.	01694804
EXIT.	01694901
*	01695001
0080-CONSISTE-ALTERACAO.	01695104
MOVE LOW-VALUES TO MENSM2O.	01696001
IF CARGM2I = SPACES OR CARGM2I = LOW-VALUES	01800001
MOVE DFHUNIMD TO CARGM2A	01810001
MOVE DFHPROTI TO LITGM2A	01820001
MOVE MSG-ERRO2 TO MENSM2O	01830001
MOVE -1 TO CARGM2L	01840001
MOVE 0 TO CODAM2L	01850001
ELSE	01860001
MOVE DFHBMPRO TO LITGM2A	01870001
MOVE DFHBMFSE TO CARGM2A	01880001
END-IF.	01890001
IF ENDEM2I = SPACES OR ENDEM2I = LOW-VALUES	01900001
MOVE DFHUNIMD TO ENDEM2A	01910001
MOVE DFHPROTI TO LITEM2A	01920001
MOVE MSG-ERRO2 TO MENSM2O	01930001
MOVE -1 TO ENDEM2L	01940001
MOVE 0 TO CODAM2L	01950001
MOVE 0 TO CARGM2L	01960001
ELSE	01970001
MOVE DFHBMPRO TO LITEM2A	01980001
MOVE DFHBMFSE TO ENDEM2A	01990001
END-IF.	02000001
IF NOMEM2I = SPACES OR NOMEM2I = LOW-VALUES	02001005
MOVE 0 TO CARGM2L	02010004
MOVE DFHUNIMD TO NOMEM2A	02020001
MOVE DFHPROTI TO LITNM2A	02030001
MOVE MSG-ERRO2 TO MENSM2O	02040001
MOVE -1 TO NOMEM2L	02050001
MOVE 0 TO CODAM2L	02060001
MOVE 0 TO ENDEM2L	02070001
MOVE 0 TO CARGM2L	02080001
ELSE	02090001
MOVE DFHBMPRO TO LITNM2A	02100001
MOVE DFHBMFSE TO NOMEM2A	02110001
END-IF.	02120001
IF MENSM2I = LOW-VALUES	02130001
MOVE MSG-PF4 TO MENSM2O	02140001
MOVE DFHBMPRO TO LITCM2A	02150001
MOVE DFHBMPRO TO LITNM2A	02160001
MOVE DFHBMPRO TO LITEM2A	02170001

MOVE DFHBMPRO TO LITGM2A	02180001
MOVE 'Z' TO CODAM2A	02190001
MOVE 'Z' TO NOMEM2A	02200001
MOVE 'Z' TO ENDEM2A	02210001
MOVE 'Z' TO CARGM2A	02220001
MOVE 'PF4' TO W-KEY	02230001
END-IF.	02240001
	02241004
0080-CONSISTE-EXIT.	02250004
EXIT.	02260001
	02261004
*	02750001
9900-XCTL-MENU-PRINCIPAL.	02760001
EXEC CICS XCTL PROGRAM('PCOB15A') END-EXEC.	02770001
9900-XCTL-EXIT.	02780001
EXIT.	02790001
*	02800001
9910-RETURN-TRANSID.	02810001
EXEC CICS RETURN TRANSID('TC8I') COMMAREA(W-COMMAREA)	02820007
END-EXEC.	02830001
9910-RETURN-EXIT.	02840001
EXIT.	02850000
*	02860000
9999-ENCERRA.	02870000
EXEC CICS SEND FROM(MSG-FIM) ERASE END-EXEC.	02880000
EXEC CICS RETURN END-EXEC.	02890000
9999-ENCERRA-EXIT.	02900000
EXIT.	02910000

```

IDENTIFICATION DIVISION.                                00010000
PROGRAM-ID. PCOB18A.                                    00020000
*                                                        00030000
*-----*                                              00040000
* TRANSACAO DE ALTERACAO E EXCLUSAO *                  00050000
* UTILIZANDO TEMPORARY STORAGE (TS) *                   00060000
*-----*                                              00070000
*                                                        00080000
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.                                00090000
SPECIAL-NAMES.
    DECIMAL-POINT IS COMMA.                            00110000
DATA DIVISION.
WORKING-STORAGE SECTION.
    01          PIC X(19) VALUE '**** INICIO-WORKING'.  00132000
                                                         00133000
    01 W-TAMANHO PIC S9(04) COMP.                        00134000
    01 W-ERRO    PIC S9(08) COMP.                        00135000
    01 W-DATA-ABS PIC S9(15) COMP-3.                     00136000
                                                         00137000
    01 W-COMMAREA.
        05 W-OPERACAO PIC X(01) VALUE SPACE.            00138000
        05 W-KEY      PIC X(03) VALUE SPACES.           00139000
                                                         00140000
    01 W-REGISTRO-TS PIC X(80) VALUE SPACES.            00141000
                                                         00150001
    01 TS-QUEUE.
        05 TS-TERMID PIC X(04) VALUE SPACES.           00151001
        05 TS-TRANSID PIC X(04) VALUE SPACES.          00152001
                                                         00153001
    01 MSG-OK      PIC X(30) VALUE 'OPERACAO REALIZADA'. 00154001
    01 MSG-FIM    PIC X(30) VALUE 'TRANSACAO ENCERRADA'. 00160000
    01 MSG-ERRO   PIC X(30) VALUE 'TECLA INVALIDA. REENTRE'. 00170000
    01 MSG-ERRO2  PIC X(30) VALUE 'REENTRE OS CAMPOS INDICADOS'. 00180000
    01 MSG-ERRO3  PIC X(30) VALUE 'ALUNO JA CADASTRADO'.  00190000
    01 MSG-ERRO4  PIC X(30) VALUE 'ERRO NO ACESSO AO ARQUIVO'. 00200000
    01 MSG-ERRO5  PIC X(30) VALUE 'ALTERE OS CAMPOS DESEJADOS'. 00210000
    01 MSG-ERRO6  PIC X(30) VALUE 'OPERACAO NAO REALIZADA'. 00220000
    01 MSG-NOTF   PIC X(30) VALUE 'ALUNO NAO CADASTRADO. REENTRE'. 00230000
    01 MSG-PF4    PIC X(30) VALUE 'PRESSIONE PF4 P/CONFIRMAR'. 00240000
    *                                                    00250000
    *                                                    00260000
    *                                                    00270000
    COPY DFHAID.                                       00280000
    COPY DFHBMSCA.                                    00290000
    COPY MPCUR11.                                     00300000
    *                                                  00310000
    01 AREA-ALUNO-ARQ.                                00320000
        05 COD-ALUNO-ARQ PIC 9(05).                    00330000
        05 NOM-ALUNO-ARQ PIC X(30).                     00340000
        05 END-ALUNO-ARQ PIC X(30).                     00350000
        05 CAR-ALUNO-ARQ PIC X(10).                      00360000
        05          PIC X(05).                           00370000
    *                                                  00380000
    01          PIC X(19) VALUE '**** FINAL-WORKING'.  00390000
    *
LINKAGE SECTION.
01 DFHCOMMAREA.
    05 L-OPERACAO PIC X(01).
    88 ALTERACAO VALUE 'A'.
    88 EXCLUSAO VALUE 'E'.
    05 L-KEY PIC X(03).

```

```

*
PROCEDURE DIVISION.
INICIO.
  IF EIBRID = DFHCLEAR OR EIBCALEN = 0
    PERFORM 9999-ENCERRA
  END-IF.

  IF EIBRID = DFHPF3
    PERFORM 9900-XCTL-MENU-PRINCIPAL
  END-IF.

  MOVE DFHCOMMAREA TO W-COMMAREA.

  IF EIBCALEN > 1
    CONTINUE
  ELSE
    PERFORM 9800-DELETA-TS
    PERFORM 0010-SEND-MAPA
    PERFORM 9910-RETURN-TRANSID
  END-IF.

  PERFORM 0020-RECEIVE-MAPA

  IF L-KEY = 'ALT'
    PERFORM 0080-CONSISTE-ALTERACAO
  ELSE
    IF EIBRID = DFHPF4 AND L-KEY = 'PF4'
      PERFORM 0050-ALTERA-INCLUI
    ELSE
      PERFORM 0030-CONSULTA-CADALUNO
    END-IF
  END-IF.

  PERFORM 0040-ENVIA-MAPA.
  PERFORM 9910-RETURN-TRANSID.
*
0010-SEND-MAPA.
  MOVE LOW-VALUES TO MAPA2O.
  MOVE 'TC8I' TO TRANM2O.
  MOVE 'I' TO TRANM2A.
  EXEC CICS ASKTIME ABSTIME(W-DATA-ABS) END-EXEC.
  EXEC CICS FORMATTIME ABSTIME(W-DATA-ABS) DDMMYY(DATAM2O)
    DATESEP('/') END-EXEC.
  MOVE DFHUNNUM TO CODAM2A.
  MOVE DFHPROTI TO NOMEM2A.
  MOVE DFHPROTI TO ENDEM2A.
  MOVE DFHPROTI TO CARGM2A.
  IF ALTERACAO
    MOVE 'ALTERACAO' TO OPERM2O
  ELSE
    MOVE 'EXCLUSAO' TO OPERM2O
  END-IF.
  EXEC CICS SEND MAP('MAPA2') MAPSET('MPCUR1I') FROM(MAPA2O)
    RESP(W-ERRO) ERASE END-EXEC.
  IF W-ERRO = DFHRESP(ERROR)
    PERFORM 9999-ENCERRA
  END-IF.
0010-SEND-EXIT.
  EXIT.
*
0020-RECEIVE-MAPA.

```

```

00480000
00490000
00500000
00510000
00520000
00530000
00531000
00532000
00533000
00534000
00535000
00536000
00537000
00538000
00539000
00540000
00550000
00551000
00560000
00570000
00580000
00590000
00600000
00610000
00620000
00630000
00640000
00650000
00660000
00670000
00680000
00680100
00680200
00680300
00680400
00680500
00680600
00680700
00680800
00680900
00681000
00682000
00683000
00684000
00685000
00686000
00687000
00688000
00689000
00690000
00700000
00710000
00720000
00730000
00740000
00750000
00760000
00770000
00780000
00790000
00800000

```

```

EXEC CICS RECEIVE MAP('MAPA2') MAPSET('MPCUR11')          00810000
  INTO(MAPA2I) RESP(W-ERRO) END-EXEC.                    00820000
IF W-ERRO = DFHRESP(ERROR)                                00830000
  MOVE MSG-ERRO TO MENSM2O                                00840000
  PERFORM 0040-ENVIA-MAPA                                  00850000
  PERFORM 9910-RETURN-TRANSID                              00860000
END-IF.                                                    00870000
IF EIBTRNID = TRANM2I                                     00880000
  CONTINUE                                                00890000
ELSE                                                       00900000
  EXEC CICS START TRANSID(TRANM2I) END-EXEC              00910000
  EXEC CICS RETURN END-EXEC                                00920000
END-IF.                                                    00930000
0020-RECEIVE-EXIT.                                       00940000
EXIT.                                                      00950000
*                                                         00960000
0030-CONSULTA-CADALUNO.                                   00970000
MOVE CODAM2I TO COD-ALUNO-ARQ                             00980000
EXEC CICS READ FILE('CADCUR02') INTO(AREA-ALUNO-ARQ)     00990000
  RIDFLD(COD-ALUNO-ARQ) RESP(W-ERRO) END-EXEC           01000000
IF W-ERRO = DFHRESP(NOTFND)                               01010000
  MOVE -1 TO CODAM2L                                      01020000
  MOVE MSG-NOTF TO MENSM2O                                01030000
  PERFORM 0040-ENVIA-MAPA                                  01040000
  PERFORM 9910-RETURN-TRANSID                              01050000
ELSE                                                       01060000
  IF W-ERRO = DFHRESP(NORMAL)                             01070000
    CONTINUE                                              01080000
  ELSE                                                    01090000
    MOVE MSG-ERRO4 TO MENSM2O                             01100000
    PERFORM 0040-ENVIA-MAPA                                  01110000
    PERFORM 9910-RETURN-TRANSID                              01120000
  END-IF.                                                 01130000
                                                         01140000
MOVE COD-ALUNO-ARQ TO CODAM2O                             01150000
MOVE NOM-ALUNO-ARQ TO NOMEM2O                             01160000
MOVE END-ALUNO-ARQ TO ENDEM2O                             01170000
MOVE CAR-ALUNO-ARQ TO CARGM2O                             01180000
                                                         01190000
IF EXCLUSAO                                               01200000
  MOVE 'Z' TO CODAM2A                                      01210000
  MOVE 'Z' TO NOMEM2A                                      01220000
  MOVE 'Z' TO ENDEM2A                                      01230000
  MOVE 'Z' TO CARGM2A                                      01240000
  MOVE 'PF4' TO W-KEY                                     01250000
  MOVE MSG-PF4 TO MENSM2O                                 01260000
ELSE                                                       01270000
  MOVE DFHBMFSE TO NOMEM2A                                01280000
  MOVE DFHBMFSE TO ENDEM2A                                01290000
  MOVE DFHBMFSE TO CARGM2A                                01300000
  MOVE 'Z' TO CODAM2A                                      01310000
  MOVE -1 TO NOMEM2L                                      01320000
  MOVE MSG-ERRO5 TO MENSM2O                               01330000
  MOVE 'ALT' TO W-KEY                                     01340000
  MOVE AREA-ALUNO-ARQ TO W-REGISTRO-TS                   01350000
  PERFORM 9810-GRAVA-TS                                   01351000
END-IF.                                                   01360000
                                                         01370000
0030-CONSULTA-EXIT.                                       01380000
EXIT.                                                      01390000
*                                                         01400000

```

```

0040-ENVIAMAPA.                                01410000
MOVE 'I' TO TRANM2A.                            01420000
EXEC CICS SEND MAP('MAPA2') MAPSET('MPCUR1I') FROM(MAPA2O) 01430000
RESP(W-ERRO) DATAONLY CURSOR END-EXEC.        01440000
IF W-ERRO = DFHRESP(ERROR)                      01450000
PERFORM 9999-ENCERRA                             01460000
END-IF.                                          01470000
0040-ENVIAMEXIT.                                01480000
EXIT.                                           01490000
*                                                01500000
0050-ALTERA-INCLUI.                             01510000
IF ALTERACAO                                    01520000
PERFORM 0060-ALTERA                             01530000
ELSE                                             01540000
PERFORM 0070-EXCLUI                             01550000
END-IF.                                          01560000
0050-ALT-INC-EXIT.                              01570000
EXIT.                                           01580000
*                                                01590000
0060-ALTERA.                                    01600000
MOVE CODAM2I TO COD-ALUNO-ARQ                   01610000
EXEC CICS READ FILE('CADCUR02') INTO(AREA-ALUNO-ARQ) 01620000
RIDFLD(COD-ALUNO-ARQ) RESP(W-ERRO) UPDATE END-EXEC 01630000
IF W-ERRO = DFHRESP(NOTFND)                     01640000
MOVE MSG-NOTF TO MENSM2O                        01650000
PERFORM 0040-ENVIAMAPA                          01660000
PERFORM 9910-RETURN-TRANSID                     01670000
ELSE                                             01680000
IF W-ERRO = DFHRESP(NORMAL)                     01681000
OR W-ERRO = DFHRESP(INVREQ)                     01682000
CONTINUE                                         01683000
ELSE                                             01684000
MOVE MSG-ERRO4 TO MENSM2O                       01685000
PERFORM 0040-ENVIAMAPA                          01686000
PERFORM 9910-RETURN-TRANSID                     01687000
END-IF.                                          01688000
PERFORM 9820-LE-TS.                              01688100
IF W-REGISTRO-TS NOT = AREA-ALUNO-ARQ          01689000
MOVE MSG-ERRO6 TO MENSM2O                       01689100
ELSE                                             01689200
01689300
MOVE NOMEM2I TO NOM-ALUNO-ARQ                   01689400
MOVE ENDEM2I TO END-ALUNO-ARQ                   01689500
MOVE CARGM2O TO CAR-ALUNO-ARQ                   01689600
01689700
EXEC CICS REWRITE FILE('CADCUR02')              01689800
FROM(AREA-ALUNO-ARQ) RESP(W-ERRO) END-EXEC     01689900
MOVE MSG-OK TO MENSM2O                          01690000
MOVE SPACE TO W-KEY                              01691001
MOVE -1 TO CODAM2L                              01691100
MOVE DFHUNNUM TO CODAM2A                        01691200
IF W-ERRO = DFHRESP(NORMAL)                     01691300
OR W-ERRO = DFHRESP(INVREQ)                     01691400
CONTINUE                                         01691500
ELSE                                             01691600
MOVE MSG-ERRO4 TO MENSM2O                       01691700
END-IF.                                          01691800
END-IF.                                          01691900
PERFORM 0040-ENVIAMAPA.                          01692000
PERFORM 9910-RETURN-TRANSID.                     01692100
0060-ALTERA-EXIT.                               01692200

```

```

EXIT. 01692300
* 01692400
0070-EXCLUI. 01692500
MOVE CODAM2I TO COD-ALUNO-ARQ 01692600
EXEC CICS DELETE FILE('CADCUR02') 01692700
RIDFLD(COD-ALUNO-ARQ) RESP(W-ERRO) END-EXEC 01692800
IF W-ERRO = DFHRESP(NOTFND) 01692900
MOVE MSG-NOTF TO MENSM2O 01693000
PERFORM 0040-ENVIA-MAPA 01693100
PERFORM 9910-RETURN-TRANSID 01693200
ELSE 01693300
MOVE MSG-OK TO MENSM2O 01693400
MOVE SPACE TO W-KEY 01693501
MOVE -1 TO CODAM2L 01693600
MOVE DFHUNNUM TO CODAM2A 01693700
IF W-ERRO = DFHRESP(NORMAL) 01693800
OR W-ERRO = DFHRESP(INVREQ) 01693900
CONTINUE 01694000
ELSE 01694100
MOVE MSG-ERRO4 TO MENSM2O 01694200
PERFORM 0040-ENVIA-MAPA 01694300
PERFORM 9910-RETURN-TRANSID 01694400
END-IF. 01694500
PERFORM 0040-ENVIA-MAPA. 01694600
PERFORM 9910-RETURN-TRANSID. 01694700
0070-EXCLUI-EXIT. 01694800
EXIT. 01694900
* 01695000
0080-CONSISTE-ALTERACAO. 01695100
MOVE LOW-VALUES TO MENSM2O. 01695200
IF CARGM2I = SPACES OR CARGM2I = LOW-VALUES 01695300
MOVE DFHUNIMD TO CARGM2A 01695400
MOVE DFHPROTI TO LITGM2A 01695500
MOVE MSG-ERRO2 TO MENSM2O 01695600
MOVE -1 TO CARGM2L 01695700
MOVE 0 TO CODAM2L 01695800
ELSE 01695900
MOVE DFHBMPRO TO LITGM2A 01696000
MOVE DFHBMFSE TO CARGM2A 01697000
END-IF. 01698000
IF ENDEM2I = SPACES OR ENDEM2I = LOW-VALUES 01699000
MOVE DFHUNIMD TO ENDEM2A 01700000
MOVE DFHPROTI TO LITEM2A 01710000
MOVE MSG-ERRO2 TO MENSM2O 01720000
MOVE -1 TO ENDEM2L 01730000
MOVE 0 TO CODAM2L 01740000
MOVE 0 TO CARGM2L 01750000
ELSE 01760000
MOVE DFHBMPRO TO LITEM2A 01770000
MOVE DFHBMFSE TO ENDEM2A 01780000
END-IF. 01790000
IF NOMEM2I = SPACES OR NOMEM2I = LOW-VALUES 01800000
MOVE 0 TO CARGM2L 01810000
MOVE DFHUNIMD TO NOMEM2A 01820000
MOVE DFHPROTI TO LITNM2A 01830000
MOVE MSG-ERRO2 TO MENSM2O 01840000
MOVE -1 TO NOMEM2L 01850000
MOVE 0 TO CODAM2L 01860000
MOVE 0 TO ENDEM2L 01870000
MOVE 0 TO CARGM2L 01880000
ELSE 01890000

```


MOVE DFHBMPRO TO LITNM2A	01900000
MOVE DFHBMFSE TO NOMEM2A	01910000
END-IF.	01920000
IF MENSM2I = LOW-VALUES	01930000
MOVE MSG-PF4 TO MENSM2O	01940000
MOVE DFHBMPRO TO LITCM2A	01950000
MOVE DFHBMPRO TO LITNM2A	01960000
MOVE DFHBMPRO TO LITEM2A	01970000
MOVE DFHBMPRO TO LITGM2A	01980000
MOVE 'Z' TO CODAM2A	01990000
MOVE 'Z' TO NOMEM2A	02000000
MOVE 'Z' TO ENDEM2A	02010000
MOVE 'Z' TO CARGM2A	02020000
MOVE 'PF4' TO W-KEY	02030000
END-IF.	02040000
	02050000
0080-CONSISTE-EXIT.	02060000
EXIT.	02070000
*	02071000
9800-DELETA-TS.	02072000
MOVE EIBTRNID TO TS-TRANSID.	02072100
MOVE EIBTRMID TO TS-TERMID.	02072200
EXEC CICS DELETEQ TS QUEUE(TS-QUEUE)	02073000
RESP(W-ERRO) END-EXEC.	02073101
9800-DELETA-EXIT.	02074000
EXIT.	02075000
*	02076000
9810-GRAVA-TS.	02077000
MOVE EIBTRNID TO TS-TRANSID.	02078000
MOVE EIBTRMID TO TS-TERMID.	02079000
EXEC CICS WRITEQ TS QUEUE(TS-QUEUE) FROM(W-REGISTRO-TS)	02080002
RESP(W-ERRO) END-EXEC	02081000
IF W-ERRO = DFHRESP(NORMAL)	02081100
CONTINUE	02081200
ELSE	02081300
MOVE MSG-ERRO4 TO MENSM2O	02081400
PERFORM 0040-ENVIA-MAPA	02081500
PERFORM 9910-RETURN-TRANSID.	02081601
9810-GRAVA-EXIT.	02082000
EXIT.	02083000
*	02084000
9820-LE-TS.	02085000
MOVE EIBTRNID TO TS-TRANSID.	02086000
MOVE EIBTRMID TO TS-TERMID.	02087000
EXEC CICS READQ TS QUEUE(TS-QUEUE) INTO(W-REGISTRO-TS)	02088002
RESP(W-ERRO) END-EXEC	02089000
IF W-ERRO = DFHRESP(NORMAL)	02089100
CONTINUE	02089200
ELSE	02089300
MOVE MSG-ERRO4 TO MENSM2O	02089400
PERFORM 0040-ENVIA-MAPA	02089500
PERFORM 9910-RETURN-TRANSID.	02089601
9820-LE-EXIT.	02089700
EXIT.	02089800
*	02090000
9900-XCTL-MENU-PRINCIPAL.	02100000
EXEC CICS XCTL PROGRAM('PCOBI5A') END-EXEC.	02110000
9900-XCTL-EXIT.	02120000
EXIT.	02130000
*	02140000
9910-RETURN-TRANSID.	02150000

EXEC CICS RETURN TRANSID('TC8I') COMMAREA(W-COMMAREA)	02160000
END-EXEC.	02170000
9910-RETURN-EXIT.	02180000
EXIT.	02190000
*	02200000
9999-ENCERRA.	02210000
EXEC CICS SEND FROM(MSG-FIM) ERASE END-EXEC.	02220000
EXEC CICS RETURN END-EXEC.	02230000
9999-ENCERRA-EXIT.	02240000
EXIT.	02250000