

UNIVERSIDADE FEDERAL DA PARAIBA
CENTRO DE CIÊNCIAS E TECNOLOGIA

CLÓVIS FORTUNATO DA MATA SOUZA

MÉTODO DE ANÁLISE COMPOSTA
APLICADO AO CONTROLE ACADÊMICO

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO
DOS CURSOS DE PÓS-GRADUAÇÃO DE ENGENHARIA DE SISTEMAS DO
CENTRO DE CIÊNCIAS TECNOLOGIA DA UNIVERSIDADE FEDERAL DA
PARAIBA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTEN-
ÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.)

COMISSÃO EXAMINADORA:

Prof. ORION DE OLIVEIRA SILVA-MSc.
Presidente

Prof. RAIMUNDO HAROLDO DO CARMO CATUNDA-M.Sc.
Examinador

Erato

Prof. ERATOSTENES EDSON RAMALHO DE ARAUJO-M.Sc.
Examinador

Campina Grande-Pb
Brasil
JULHO/77



S729m Souza, Clóvis Fortunato da Mata.
Método de análise composta aplicado ao controle acadêmico / Clóvis Fortunato da Mata Souza. - Campina Grande, 1977.
223 f.

Dissertação (Mestrado em Ciências) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 1977.
"Orientação : Prof. Orion de Oliveira Silva".
Referências.

1. Sistemas - Desenvolvimento. 2. Análise Composta - Método. 3. Sistema de Controle Acadêmico - Desenvolvimento. 4. Dissertação - Ciências. I. Silva, Orion de Oliveira. II. Universidade Federal da Paraíba - Campina Grande (PB). III. Título

CDU 004.65(043)

À minha esposa Maria Elisabete
e aos nossos filhos Luiz Cláu
dio e Karla Cristina pela par
ticipação e estímulo constan
te.

A G R A D E C I M E N T O S

Ao Professor Orion de Oliveira Silva, pela orientação, incentivo e acompanhamento atencioso em todo o desenvolvimento desta tese.

À Centrais Elétricas do Piauí S.A.- CEPISA, na pessoa dos seus diretores, pela oportunidade proporcionada e total apoio, sem o qual este trabalho não seria desenvolvido.

À Universidade Federal do Piauí e à Escola Técnica Federal do Piauí, pelo incentivo.

Ao meu pai, pela lembrança nunca esquecida, à minha mãe, que me orientou os primeiros passos e me fez atingir este objetivo.

Aos funcionários do CCT-UFPb, Leônia Leão da Nóbrega pela ajuda na correção gramatical deste trabalho, bem como a Antonio Costa Araújo e Francisco de Assis Coutinho Souto (do Controle Acadêmico) pelo apoio que prestaram em momento oportuno.

R E S U M O

Procede-se, nesta dissertação, um estudo de uma metodologia de desenvolvimento de software aplicado.

O objetivo desta obra é apresentar o método denominado Análise Composta, e, em seguida aplicá-lo no desenvolvimento do Sistema de Controle Acadêmico, melhorando a sua estruturação e apresentando sugestões.

A B S T R A C T

Following, in the dissertation, is a study of a methodology in development of applied software.

The object of this work is to present the method designated Composite Analysis and following to apply it in the development of an Academic System of Control thus improving its structure and furnishing hints.

I N D I C E

AGRADECIMENTOS	iii
RESUMO	iv
ABSTRACT	v
CAPÍTULO 1 - INTRODUÇÃO	01
CAPÍTULO 2 - DEFINIÇÕES	06
CAPÍTULO 3 - MODULARIDADE	15
3.1 - INTRODUÇÃO	15
3.2 - MÓDULO	19
3.2.1 - Considerações Sobre Módulos de Programação Modular	20
3.2.2 - Considerações Sobre Módulos de "Design" de Sistema	21
3.2.3 - Medidas de Módulos	28
3.2.3.1 - Eficácia de Módulos	29
3.2.3.2 - Acoplamento de Módulos	41
3.3 - "TOP-DOWN DESIGN"	53
3.4 - "BOTTOM-UP"	59
CAPÍTULO 4 - ANÁLISE COMPOSTA	61
4.1 - METODOLOGIA ANÁLISE COMPOSTA	66
4.1.1 - ESTRUTURA DO PROBLEMA	67
4.1.2 - FLUXO DE DADOS EXTERNOS	67
4.1.3 - PONTOS DE ALTA ABSTRAÇÃO	68
4.1.4 - DECOMPOSIÇÃO	71
4.1.5 - RECURSÃO	72
4.1.6 - TÉRMINO DA DECOMPOSIÇÃO	75
CAPÍTULO 5 - SISTEMA CONTROLE ACADÊMICO	77
5.1 - SUBSISTEMA CADASTRO DE ALUNOS	82
5.2 - SUBSISTEMA CURRÍCULOS DE CURSOS	90
5.3 - CADASTRO GERAL DE DISCIPLINAS	99
5.4 - SUBSISTEMA CADASTRO SEMESTRAL DE DISCIPLINAS	106

5.5 - SUBSISTEMA CADASTRO DE PROFESSORES	114
5.6 - SUBSISTEMA ESTUDO DE DEMANDA DE VAGAS	122
5.6.1 - EMISSÃO DE TOTAIS PARA UMA DISCIPLINA	125
5.6.2 - EMISSÃO DE TOTAIS PARA DUAS DISCIPLINAS	134
5.6.3 - EMISSÃO DE TOTAIS PARA TRES DISCIPLINAS	141
5.6.4 - EMISSÃO DE TOTAIS PARA QUATRO DISCIPLINAS	149
5.7 - SUBSISTEMA MATRÍCULA	155
5.7.1 - PRIMEIRA MATRÍCULA	158
5.7.2 - SEGUNDA MATRÍCULA	168
5.8 - SUBSISTEMA PROCESSAMENTO DE NOTAS	178
CAPÍTULO 6 - CONCLUSÕES	187
APÊNDICE	188

INDICE DE FIGURAS

FIGURA 1.1 - TENDÊNCIA DO CUSTO DE HARDWARE-SOFTWARE	02
2.1 - DESCRIÇÃO DE "INTERFACES"	09
2.2 - CICLO DE VIDA DO SOFTWARE	13
3.1 - MÓDULOS RECURSIVOS	22
3.2 - ESTRUTURA DE DECISÃO	23
3.3 - NOTAÇÃO BÁSICA	25
3.4 - DESCRIÇÃO DE "INTERFACES"	26
3.5 - ALOCAÇÃO DE PROGRAMAS	27
3.6 - MÓDULO "EDITE" TODOS OS DADOS	31
3.7 - MÓDULO DE EFICÁCIA CLÁSSICA	33
3.8 - MÓDULO DE EFICÁCIA INFORMACIONAL	37
3.9 - PROGRAMA A SER ESTUDADO	39
3.10- ACOPLAMENTO DE CONTROLE	46
3.11- ACOPLAMENTO DE DADOS VERSUS CONTROLE	50
3.12- ELEMENTOS DE UM MÓDULO	54
3.13(a)- DIAGRAMA HIERARQUICO	56
3.13(b)- SECÇÃO DE DESCRIÇÃO DO SISTEMA VENDA/ESTOQUE	57
3.14- DIAGRAMA DETALHADO 2.2, HIPO	58
3.15- ILUSTRAÇÃO UTILIZADA NO DESENVOLVIMENTO "BOTTOM-UP"	60
4.1 - PROJETO EXTERNO	63
4.2 - ARQUITETURA DO SISTEMA	64
4.3 - PROJETO DOS COMPONENTES E SUBCOMPONENTES DO SISTEMA	65
4.4 - ESTRUTURA DO PROBLEMA	67
4.5 - ABSTRAÇÃO DE DADOS NA ESTRUTURA DO PROBLEMA	70
4.6 - PONTOS DE ALTA ABSTRAÇÃO	70
4.7 - INICIANDO A ESTRUTURA DO PROBLEMA	72
4.8 - DECOMPOSIÇÃO DE MÓDULOS TIPO FONTE E SORVEDOURO	74
5.1 - SISTEMA CONTROLE ACADEMICO E O AMBIENTE	79
5.2 - ARQUITETURA DO SISTEMA CONTROLE ACADÊMICO	80

FIGURA 5.3 - PROJETO DOS COMPONENTES E SUBCOMPONENTES DO SISTEMA	81
5.4 - ESTRUTURA DO PROBLEMA	84
5.5 - "INTERFACE" E ESTRUTURA DE TOPO	84
5.6(a) - DIAGRAMA ESTRUTURAL DO CADASTRO DE ALUNOS	85
5.6(b) - "INTERFACES" DO CADASTRO DE ALUNOS	86
5.7 - ESTRUTURA DO PROGRAMA	92
5.8 - ESTRUTURA E "INTERFACES" DO TÔPO	92
5.9(a) - DIAGRAMA ESTRUTURAL DO SUBSISTEMA CURRÍCULOS DE CURSOS	93
5.9(b) - "INTERFACES" DO SUBSISTEMA CURRÍCULOS DE CURSOS	94
5.10- ESTRUTURA DO PROBLEMA	100
5.11- "INTERFACE" E ESTRUTURA DE TÔPO	100
5.12(a)- DIAGRAMA ESTRUTURAL DO CADASTRO GERAL DE DISCIPLINA	101
5.12(b)- "INTERFACES" DO CADASTRO GERAL DE DISCIPLINA	102
5.13- ESTRUTURA DO PROBLEMA	107
5.14- ESTRUTURA E "INTERFACES"	107
5.15(a)- DIAGRAMA ESTRUTURAL DO CADASTRO SEMESTRAL DE DISCIPLINAS	108
5.15(b)- "INTERFACES" DO CADASTRO GERAL SEMESTRAL DE DISCIPLINAS	109
5.16- ESTRUTURA DO PROBLEMA	115
5.17 ESTRUTURA E "INTERFACES" DE TOPO	115
5.18(a)- DIAGRAMA ESTRUTURAL DO SUBSISTEMA CADASTRO DE PROFESSORES	116
5.18(b)- "INTERFACE" DO SUBSISTEMA CADASTRO DE PROFESSORES	117
5.19- SUBDIVISÕES DO SUBSISTEMA ESTUDO DE DEMANDA DE VAGAS	125
5.20- ESTRUTURA DO PROBLEMA	126
5.21- ESTRUTURA E "INTERFACES" DE TÔPO	126
5.22(a)- DIAGRAMA ESTRUTURAL DA FASE 1	127
5.22(b)- "INTERFACES" DE FASE 1	128

FIGURA 5.23- ESTRUTURA DO PROBLEMA	134
5.24- ESTRUTURA E "INTERFACES" DO TOPO	134
5.25(a)- DIAGRAMA ESTRUTURAL DA FASE 2	135
5.25(b)- "INTERFACES" DE FASE 2	136
5.26- ESTRUTURA DO PROBLEMA	141
5.27- ESTRUTURA E "INTERFACES" DE TOPO	141
5.28(a)- DIAGRAMA ESTRUTURAL DA FASE 3	143
5.28(b)- "INTERFACES" DA PRIMEIRA MATRÍCULA	143
5.29- ESTRUTURA DO PROBLEMA	148
5.30- ESTRUTURA E "INTERFACES" DO TOPO	148
5.31(a)- DIAGRAMA ESTRUTURAL DA FASE 4	149
5.31(b)- "INTERFACES" DA FASE 4	150
5.32- SUBDIVISÕES DO SUBSISTEMA MATRÍCULA	158
5.33- ESTRUTURA DO PROBLEMA	158
5.34- ESTRUTURA E "INTERFACE" DE TOPO	159
5.35(a)- DIAGRAMA ESTRUTURAL DO SUBSISTEMA DA PRIMEIRA MATRÍCULA	160
5.35(b)- "INTERFACES" DA PRIMEIRA MATRÍCULA	161
5.36- ESTRUTURA DO PROBLEMA	168
5.37- ESTRUTURA E "INTERFACES" DE TOPO	168
5.38(a)- DIAGRAMA ESTRUTURAL DO SUBSISTEMA SEGUNDA MATRÍCULA	169
5.38(b)- INTERFACES DE SEGUNDA MATRICULA	171
5.39- ESTRUTURA DO PROBLEMA	180
5.40- ESTRUTURA E "INTERFACES" DO PROCESSAMENTO DE NOTAS	180
5.41(a)- DIAGRAMA ESTRUTURAL DO SUBSISTEMA PROCESSAMENTO DE NOTAS	181
5.41(b)- "INTERFACES" DO SUBSISTEMA PROCESSAMENTO DE NOTAS	182
 FIGURA A.1 - PARTE DO ARQUIVO CACAØ1	 189
A.2 - ESTRUTURA DO CÓDIGO DE CURSO	190
A.3 - ESTRUTURA DA NATUREZA DO INGRESSO	190
A.4 - ESTRUTURA DO CÓDIGO DE DISCIPLINA	191
A.5 - ESTRUTURA DO TIPO DE SANGUE	191

INDICE DE TABELAS

TABELA 1 - DETERMINAÇÃO DE EFICÁCIA	42
2 - ATRIBUTOS DE EFICÁCIA	43
3 - DETERMINAÇÃO DE ACOPLAMENTO	51
4 - ATRIBUTOS DE ACOPLAMENTOS	52

CAPÍTULO I

INTRODUÇÃO

A extrema dificuldade e o alto custo para se criar e manter grandes sistemas de computação vem se constituindo em um sério problema nos dias atuais, mormente, quando se observa que há alguns anos não se dispunha de grandes máquinas, e conseqüentemente, não se tinha grandes obstáculos com programação. Agora, porém, com o aparecimento dos sofisticados computadores, a tarefa de programar vem se tornando mais complicada e onerosa. O pesquisador Dijkstra, que vem acompanhando de perto o desenvolvimento de Software, resume a evolução computacional com as seguintes palavras: "Com o aparecimento dos computadores gigantes, a arte de programar tornou-se, também, um problema gigante". Tal fato evidencia-se, quando se observam os últimos estudos desenvolvidos nos Estados Unidos; onde se comprova que o custo anual de Software comparado com o custo de Hardware, tende a crescer continuamente, como mostra a Figura 1.1. Nos próximos dez anos ha

verã um crescimento na ordem de vinte e um a vinte e tres por cento, em face da crescente utilizaçãõ de Software no dia a - dia do mundo moderno, automatizando alguns processos que controlam o estilo de vida, proliferando a utilizaçãõ de Software, tais como: equipamentos m\u00e9dicos, controle de tr\u00e1fego, sistema de defesa, registros pessoais, conta banc\u00e1ria, controle de tr\u00e1fego a\u00e9reo, aplicações comerciais diversas etc.

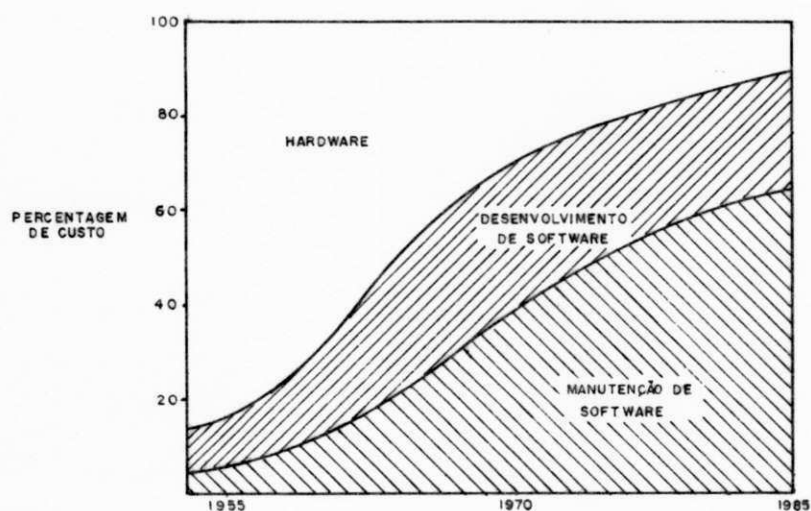


FIGURA 1.1 - TENDÊNCIA DO CUSTO DE HARDWARE-SOFTWARE

V\u00e1rios projetistas de Software tendem a decidir por determinadas t\u00e9cnicas fundamentadas em princ\u00edpios puramente emocionais, o que, de certa forma, \u00e9 justifi

cável, porque "programar" até certo ponto, é uma arte, constituindo-se, ainda, em parte de um processo criativo. Nos últimos anos, contudo, as técnicas de programação tem evoluído no sentido de estabelecer regras gerais que possam ser seguidas pelos projetistas no desenvolvimento de seus programas. Diversas abordagens tem sido sugeridas pelos pesquisadores Dijkstra, Wirth e Parnas, seja no campo da programação estruturada, seja na programação modular, todas, visando estabelecer uma metodologia para projeto de programas, que - por sua vez - muito dependem de como são apresentadas suas especificações e, estas dependendo de como o problema é abordado. As metodologias conhecidas pelos nomes de "top - down design", "Levels of abstractions", "step - wise refinement", sugerem que o desenvolvimento dos programas comecem com a especificação do problema dada por alto, ou seja, de forma geral, e não específica, sendo refinadas sucessivamente até atingir um nível de especificação desejado, capaz de ser traduzido para uma linguagem de programação. É a abstração - sem dúvida alguma - uma ferramenta poderosa que habilita um projetista a dividir e dominar a complexibilidade de um sistema, constituindo-se em descrever um sistema ou parte dele sem especificar completamente todos os detalhes, dando ao projetista uma visão macroscópica do sistema, como também, uma perspectiva das relações e propriedades globais dos principais elementos do sistema, que seriam mais difíceis de se obter se todos os detalhes se apresentassem ao mesmo tempo.

Nos dias atuais, com a utilização de siste

mas de computação em tarefas de grandes responsabilidades, tais como: no lançamento de U.S. Mariner e da história Apolo 11, tres fatores são importantíssimos em um sistema de programação: qualidade, custo e tempo. O fator qualidade associa-se à confiança (integridade) manutenção, modificabilidade, generalidade, uso e eficiência. A confiabilidade de um sistema é função do número de erros ou "bugs" que possam ser encontrados no mesmo. Porquanto, a manutenção refere-se ao esforço e ao tempo dispendidos na correção de tais erros. A modificabilidade por sua vez, implica em custo na mudança ou extensão do sistema, enquanto que, a generalidade refere-se às funções que os sistemas são capazes de suportar.

Experiências tem mostrado que o alto custo e a grande demanda de tempo de espera são fatores intimamente ligados à administração de projetos de sistemas, ocasionados na maioria das vezes por administração ineficiente e, em outros casos, devido à ausência total de administração caracterizada pelo desenvolvimento aleatório de atividades, onde a troca de pessoal é uma constante, as ferramentas de administração são carentes, resultando um projeto final diferente daquilo que se pretendia "a priori", carente de uma documentação completa e atualizada. Destarte, medidas devem ser tomadas quando do desenvolvimento de sistemas, atentando-se para os requisitos mínimos indispensáveis à administração, fornecendo-lhe pessoal qualificado, evitando-se o remanejamento do mesmo para outros projetos e, sobretudo, fornecendo-lhe recursos suficientes, que - por certo - reduzirão os

custos e o tempo de demanda.

Este trabalho se propõe a descrever a metodologia denominada Análise Composta, que se caracteriza pela decomposição de um sistema em um conjunto de partes altamente independentes, surtindo efeitos positivos na qualidade do sistema, permitindo um alto grau de confiabilidade, expansibilidade e redução de custos.

O Capítulo 2 é dedicado às definições de termos que se fazem necessários à compreensão do presente trabalho.

O Capítulo 3 é dedicado ao estudo genérico de modularidade de sistemas, dando ênfase à "eficácia de módulo", e ao "acoplamento entre módulos", apresentando medidas, (dentre outras) para maximização da eficácia e minimização entre módulos.

O Capítulo 4 se dedica ao estudo específico da metodologia "Análise Composta", onde se faz uma descrição desta técnica e apresentação de um exemplo ilustrativo.

O Capítulo 5 é dedicado ao desenvolvimento do Sistema de Controle Acadêmico da Universidade Federal da Paraíba, constituindo-se no ponto alto deste trabalho, onde a metodologia "Análise Composta" e os conceitos aqui apresentados, são utilizados no desenvolvimento do referido sistema.

CAPÍTULO 2

DEFINIÇÕES

A Ciência da Computação apresenta vários termos técnicos que - para muitos - se constitui em novos conhecimentos. Neste Capítulo, dá-se uma relação contendo alguns desses termos e seus significados, com a intenção de facilitar a compreensão deste trabalho.

Application Package - pacote de aplicação, numa rotina ou num conjunto de rotinas, destinadas a uma determinada aplicação.

Attribute - atributo, propriedade descritiva ou característica, associada a um nome.

Automatic Programming - programação automática. Qualquer técnica de utilização do computador, ou da assistência ao preparo da programação.

Estrutura - a palavra "estrutura" é usada para definir uma descrição parcial de um determinado sistema. A "estrutura" mostra o sistema dividido em um conjunto de módulos, dando algumas características de cada módulo, e especificando algumas conexões entre os mesmos.

Conexão - o termo conexão é usualmente aceitável ao pé da letra muitos asseguram que "conexão" são pontos de transferência de controle, parâmetros passados, particionamento de dados para Software, fios ou outras conexões físicas para hardware. A "conexão" entre módulos está na adoção que cada módulo faz sobre o outro.

Confiança - é uma medida de número de erros, ou "bugs" encontrado em um programa.

Manutenção - é uma medida de esforço e tempo requerido para "fix bugs" do programa.

Modificabilidade - é a medida do custo de mudança ou extensão do programa.

Generalidade - é a medida das funções que um programa desempenha.

Eficiência - é a medida da execução comportamental de um programa. Por exemplo: em termo de execução

ção são usadas velocidade e armazenamento.

Função - é uma descrição do que o programa faz (características externas do programa).

"Performance" - é uma descrição de como o programa desempenha bem suas funções, medida em termos de velocidade de execução, tamanho de armazenamento, recursos usados e tempo médio de execução.

Modularidade - a modularidade é um neologismo. Neste trabalho a palavra modular será usada para indicar um programa que tenha sido estruturado com alto grau de independência ou modularização.

Módulo - um módulo é um grupo de declarações de programa com as seguintes características:

- a) Função (o que o programa faz);
- b) "Performance" (Quão bem o programa trabalha);
- c) Estrutura (como o programa é construído).

Segmento - é um conjunto de declarações, tendo algumas das características de módulo. As declarações estão lexicamente juntas, limitadas e podem ou não ter um

nome coletivo (nome do segmento).

Fan-out - o "fan-out" de um módulo é o número de módulos por que são chamados por este módulo. Em um diagrama estruturado o mesmo representa o número de linhas originadas da parte de baixo de um módulo. Na Figura 2.1 o "fan-out" do módulo "A" é 2.

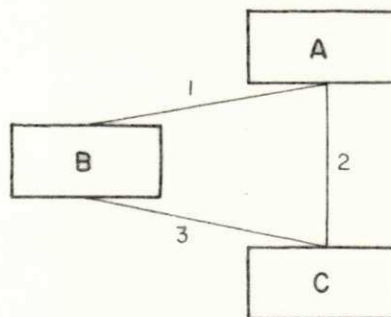


FIGURA 2.1 - DESCRIÇÃO DE "INTERFACES"

Fan-in - o "fan-in" de um módulo é o número de módulos que chama este módulo. Em um diagrama, estruturado o "fan-in" representa o número de linhas de entrada do lado de cima de um módulo. Na Fig. 2.1 o "fan-in" do módulo "C" é 2.

Contexto de um Módulo - é o uso particular do módulo para certa razão.

Abstração - é uma descrição de um sistema ou parte dele que não especifica todos os seus detalhes. Em cada sistema há muitas abstrações possíveis. Cada abstração dá ao observador uma perspectiva diferente sobre o que o sistema faz ou com que se assemelha.

Seja $A(1), A(2), \dots, A(n)$ uma série de abstrações que para cada i , $A(i+1)$ é um refinamento da prévia abstração $A(i)$. Cada abstração $A(i)$ é uma descrição completa, embora não necessariamente detalhada do sistema inteiro.

Nível de Abstração - é, portanto, um conceito relativo. Não faz sentido falar de alto nível de abstração, sem que a descrição envolvida, esteja sendo ocupada com alguma outra descrição que seja mais detalhada.

Máquina Virtual - é um conjunto de operações primitivas que descrevem a operação de um sistema, ou parte de um sistema. Denomina-se de máquina virtual porque não necessariamente deve ser implantada em hardware.

Programação Estruturada - a formulação exata da definição de programação estruturada não é fácil, pois o termo tem sido usado de diferentes maneiras. Examine-se

algumas dessas maneiras.

1 - "Um método de programar de acôrdo com um conjunto de regras com o objetivo de tornar fácil a leitura e a manutenção do programa" (Baker).

2 - "Um complexo de idéias de organização e disciplina que auxilia o processo de programação" (Mills).

Software - define-se software não sômente como programas de computadores, mas também como a documentação requerida para o desenvolvimento, operação e manutenção dos programas.

Engenharia de Software - é a aplicação prática do conhecimento científico no projeto e construção de programas e a documentação associada necessária para o desenvolvimento, operação e manutenção dos mesmos. Tres aspectos devem ser observados nesta definição:

a) A palavra "design" cobre uma gama de atividades de engenharia de Software.

b) A definição deve cobrir o ciclo de vida interno do Software, incluindo suas atividades de reprojeter e modificar frequentemente.

c) A manutenção do Software.

A Figura 2.2 indica o conjunto de atividades incluídas na definição.

Estrutura de Dados - é um grupo de um ou mais elementos de dados, os quais estão estruturalmente relacionados entre si. Os dois problemas fundamentais na manipulação de estruturas de dados são: a criação e a manutenção das mesmas. | 1 |

Elemento - é uma unidade lógica de informação.

Programa com Organização Funcional - significa que existe alta eficácia entre módulos.

Dados Externos - são dados que iniciam e/ou terminam fora da estrutura.

Dados Conceituais - são dados que são originários de um ou vários "devices" de entrada.

Ambiente - refere-se à estrutura de dados. Melhores esclarecimentos na referência bibliográfica 5.

Estrutura Global - significa que os dados

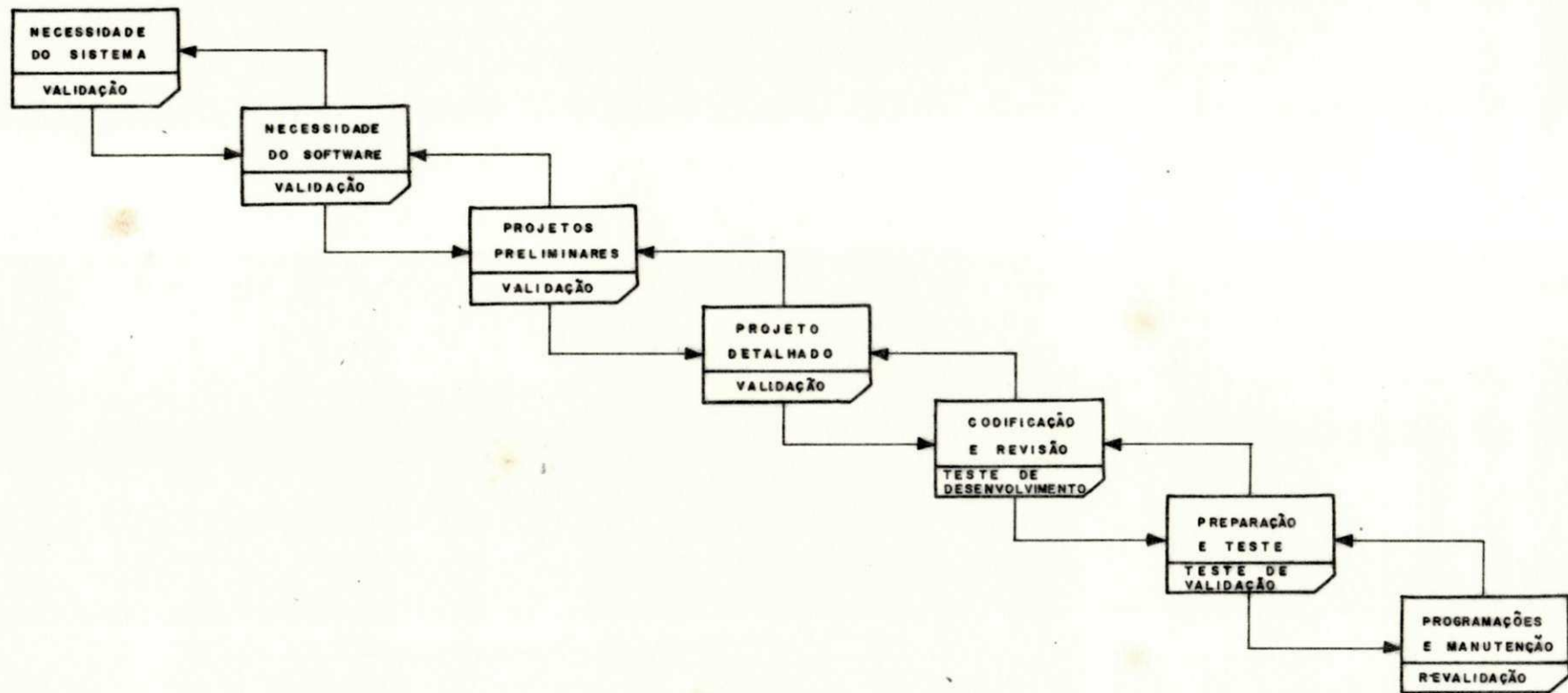


Fig. 2.2 CICLO DE VIDA DO SOFTWARE

não se encontram em um ambiente comum. Mai
ores detalhes na referência bibliográfica 3.

Caixa Preta - termo usado para indicar que
o módulo é independente, fechado entre si.

Processo - trecho codificado ou um progra
ma.

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Geral de Pós-Graduação
Rua Aprígio Veloso, 852 Tel. (662) 321-7222-R 355
68.100 - Campina Grande - Paraíba

CAPÍTULO 3

M O D U L A R I D A D E**3.1 - INTRODUÇÃO**

O termo modularidade vem se popularizando pelo crescente emprego por pessoas ligadas à área da Ciência da Computação, mormente, por aquelas imbuídas na tarefa de desenvolvimento de Software.

Programadores, analistas e gerentes de Instalações de Processamento de Dados, atualmente, já aceitam o fato de que a modularidade é uma ferramenta proveitosa no desenvolvimento de sistemas e programas. Contudo, uma boa modularidade é, ainda, tarefa difícil de se executar, embora, levando a resultados excelentes, principalmente quando aplicada a sistemas de grande porte. Tal técnica facilita bastante o manuseio de sistemas quando do seu desenvolvimento, permitindo o uso mais eficiente do pessoal disponível na fase de implantação e teste, de forma paralela de diferentes módulos,

evitando-se assim, a duplicação de esforços, permitindo a padronização de determinadas funções e seu uso em várias partes do Sistema.

Ao, se comparar dois programas que se propõem a executar o mesmo problema, sendo um projetado modularmente e o outro projetado monoliticamente (dividido em módulos arbitrários) certamente o programa modular apresentará resultados bem melhores do que o monolítico, em face da sua facilidade de ser mantido e ampliado.

Quando do desenvolvimento de um sistema ou programa, alguns aspectos devem ser levados em consideração, porque para muitos projetistas tais aspectos se constituem em favoráveis e desfavoráveis à modularidade, como:

1. Aspectos Favoráveis à Modularidade | 2 |

Quando da abordagem de um problema a ser projetado, necessário se torna que o projetista pense nos seguintes aspectos:

- a) Um programa modular é simples de ser escrito e revisado.
- b) A manutenção de um programa ou de um sistema modular é simples de ser efetuada.
- c) A administração e controle de um programa ou sistema modular, quando do seu

desenvolvimento, torna-se bem mais fá-
cil de se realizar, porquanto os módu-
los com maiores dificuldades de desen-
volvimento podem ser distribuídos com
profissionais mais experientes.

2. Aspectos Desfavoráveis à Modularida- de | 2 |

Depois de se haver mencionado algumas van-
tagens pertinentes à modularidade, pergunta-se: por que mui-
tos sistemas de programação não são escritos num estilo mais
modular? De fato, tal prática não ocorre com frequência. Gran-
des Organizações Comerciais dispendem mais de 50 % de seu or-
çamento destinado à Processamento de Dados em modificações de
sistemas existentes. Muitas modificações teriam sido executa-
das mais rapidamente e seriam menos onerosas se os sistemas
tivessem sido originalmente projetados, utilizando-se uma das
técnicas de modularidade existentes.

Algumas causas que levam à não aceitação
da modularidade por parte de muitos projetistas são:

i - Modularidade Requer Mais Trabalho que a Abordagem Convencional

No desenvolvimento de um sistema, aplican-
do-se uma técnica de modularidade, o projetista deve ser ma

is meticoloso na fase de "design", devendo projetar seus programas num estilo "Top-down" iniciando com o "top-level design" do programa e, a partir daí, trabalhar normalmente, detalhando os módulos individuais.

ii- Falta de Conhecimento do Método de Modularidade

Talvez a maior razão da não preferência por sistemas modulares seja a falta de conhecimento do método por parte de muitos projetistas.

iii- Abordagem Modular Ocasionalmente Requer Mais Tempo de C.P.U.

Este problema ocorre em face do uso de muitas subrotinas em linguagens de alto nível. O tempo adicional requerido não excede de 5 - 10 % do tempo de C.P.U., com relação à metodologia convencional. Isto é na verdade, um preço razoável, porque em contra partida tem-se um sistema, com mais facilidade de manutenção e extensão.

iv - Abordagem Modular Pode Requerer Logicamente Mais Espaço de Memória

Em muitos casos a modularidade não aumentará o programa por mais de 5 - 10 % em relação à metodologia

gia convencional. Isto não causará problemas, exceto em máquinas onde o tamanho do programa seja limitado, ou em mini computadores onde haja uma limitação física de memória.

v - Modularidade Pode Causar Problemas em Sistemas de Tempo Real ou On-Line

O problema torna-se mais sério quando se utiliza um computador SYSTEM/370 da IBM, com sistema de memória virtual. Muitos projetistas dividem os programas em subrotinas básicas, que se acomodam em uma página de máquina. Estas subrotinas ocasionalmente chamam outras que se colocam numa página ou num conjunto de páginas, as quais estão na memória, ao mesmo tempo, vindo a se constituir em uma medida que requer habilidade por parte dos projetistas.

3.2 - MÓDULO

O módulo em si, vem a se constituir em um poderoso instrumento no desenvolvimento da estrutura de programas modulares e "design" de sistemas que utilizam a técnica de modularidade. A seguir, emite-se algumas considerações importantes referentes a tipos e conteúdos de módulos, quer no seu emprego em programas, quer como instrumento de "design" de sistemas.

3.2.1 - CONSIDERAÇÕES SOBRE MÓDULOS DE PROGRAMAÇÃO MODULAR | 2 |

Conforme citamos acima, a programação modular produz pequenas secções denominadas módulos, os quais se constituem de um grupo de declarações de programas de forma que:

i - As declarações devem se apresentar lexicamente juntas, isto é: quando de posse de uma listagem de programa, o leitor terá todas as declarações fisicamente juntas, o que facilitará - sobremaneira - a leitura e compreensão do conteúdo do módulo.

ii - As declarações devem aparecer delimitadas por identificadores que estabeleçam o início e o fim de um módulo, tais como; "STAR" e "END".

iii - As declarações contidas em um módulo podem ser referenciadas pelo nome do módulo de alguma outra parte do programa ou - em outras palavras - um módulo de programa, tanto pode chamar outro módulo, como pode ser chamado de maneira recíproca.

iv - Há muitas controvérsias com relação ao tamanho de um módulo. Uns acham que deve conter 50 declarações de uma linguagem de alto nível, outros que deve conter até 200 declarações. Existem ainda os que argumentam que

um módulo deve conter de 40 a 120 declarações. De forma que não existe um tamanho rígido para um módulo.

v - Cada módulo deve ser como uma "caixa preta", isto é: os resultados de sua execução dependerão sõmente dos dados (parâmetros) de entrada.

Desta forma, os módulos correspondem a entidades estruturadas em muitas linguagens de programação, por exemplo: "SUB PROGRAMAS E FUNÇÕES" em Fortran, uma "PROCEDURE" em PL/I e "ALGOL" e um "SUB PROGRAMA" em "COBOL".

3.2.2 - CONSIDERAÇÕES SOBRE MÓDULOS DE "DESIGN" DE SISTEMA

Com a intenção de dotar o leitor de todas as informações possíveis, ou quase todas que sejam necessárias para o entendimento da metodologia "ANÁLISE COMPOSTA" que será vista no Capítulo seguinte, apresenta-se, a seguir, algumas características inerentes aos módulos de um sistema.

Simplicidade - Um módulo deve se destinar a solucionar um único problema, e, sempre que possível, deve ser generalizado a ponto de atender a outras partes do projeto. Tal generalização se relaciona intimamente com a estrutura dos dados. Por exemplo: se um módulo se destina a resolver equação do 2º grau, não deverá resolver equações do 3º

grau, mas, deve ser capaz de lidar com os mais variados tipos de dados, tais como: números de ponto flutuante, números inteiros etc.

Recursão - a recursão se apresenta quando um módulo chama outro que - por sua vez - chama outro que chama o módulo original, tal como mostra a Figura 3.1.

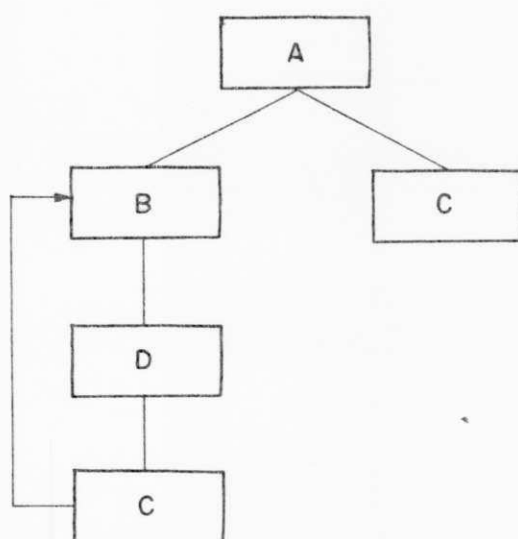


FIGURA 3.1 - MÓDULOS RECURSIVOS

Módulos Rígidos - são módulos que não são genéricos com relação à estrutura de dados. Por exemplo: são

aditem um tipo de dados como entrada. Tal prática é desaconselhável, porque contradiz um dos princípios básicos da modularização que é a generalização com relação ao ambiente.

Estrutura de Decisão - É preferível que um módulo afetado por uma decisão seja subordinado diretamente ao módulo que contém a decisão. Em tal estrutura o resultado de uma decisão fica oculto pelo módulo que toma a decisão, conforme mostra a Figura 3.2.

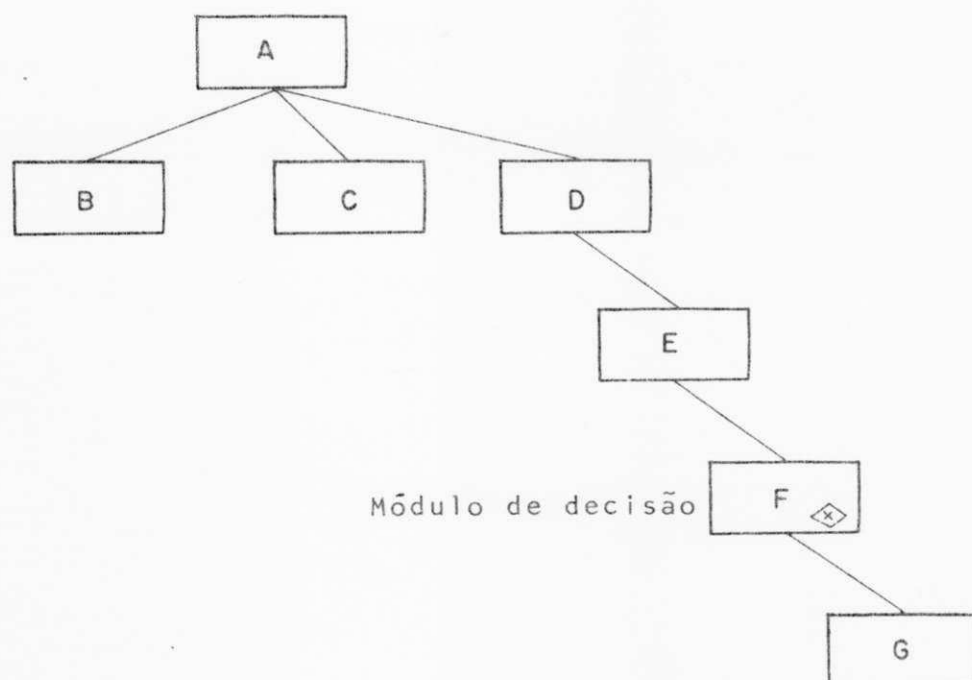


FIGURA 3.2 - ESTRUTURA DE DECISÃO

Acesso de Dados - Deve-se sempre que possível, minimizar a quantidade de dados a ser transmitida como parâmetro de um módulo. Ao invés de se transmitir uma estrutura de dados inteira, ou um registro, encaminham-se somente os dados, ou melhor, os campos que serão utilizados por aquele módulo.

Restrição de Módulo - Há casos em que um módulo restringe sua codificação e documentação a outro módulo. Contudo, para obtenção de um bom "design" tal prática deve ser evitada.

Isolamento de INPUT/OUTPUT - Isolar as operações de I/O dentro de um pequeno número de módulos, é uma meta recomendável. Tal estratégia engrandece a portabilidade e extensibilidade de um sistema permitindo que o mesmo seja convertido para outros sistemas operacionais sem muito trabalho.

Há módulos que contêm declarações executáveis e há outros que contêm declarações não executáveis. O propósito de um módulo executável, baseia-se em receber dados, transformá-los e retorná-los como dados de saída. Observando a Figura 3.3 pode-se obter as seguintes informações:

- a) Existem Quatro Módulos, A, B, C e D
- b) O módulo B recebe uma entrada D1 e faz retornar uma saída D2. O módulo C rece

be uma entrada D4 ou D5, e faz retornar uma saída D4 ou D6. As setas paralelas às "interfaces" dos módulos representam os parâmetros que passam de um módulo para outro (entradas e saídas).

- c) O módulo B é subordinado ao módulo A, e o módulo D, é subordinado aos módulos C e A.

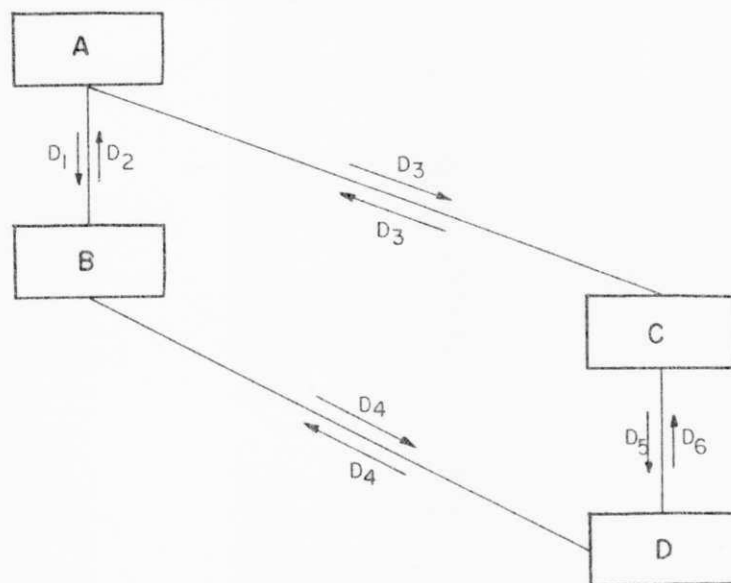


FIGURA 3.3 - NOTAÇÃO BÁSICA

A Figura 3.3 mostra um diagrama que enfoca somente o relacionamento estrutural, sem abordar qualquer procedimento ou relacionamento algorítmico. Observa-se, também, que o gráfico não diz se o módulo A chama o módulo B antes do módulo C ou vice-versa.

Outra maneira de ilustrar os parâmetros de entrada e saída de um módulo e suas "interfaces" é mostrada na Figura 3.4.

INTERFACES	PARÂMETROS	
	ENTRADA	SAÍDA
1	D1	D2
2	D3	D3
3	D4	D4
4	D5	D6

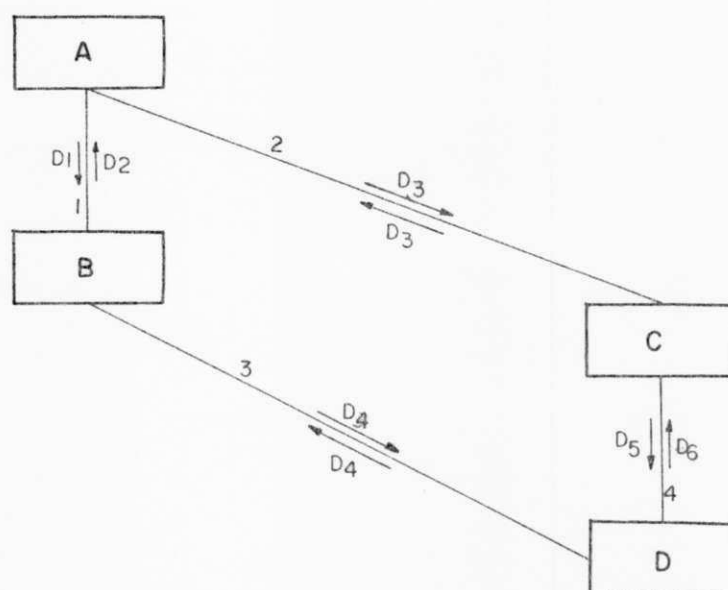


FIGURA 3.4 - DESCRIÇÃO DE "INTERFACES"

Os módulos de maneira geral possuem tres atributos básicos: Função, Lógica e "Interface".

Um módulo em si não se restringe a relatar unicamente operações de transformações por ele executadas, em determinados casos, referencia também funções de qualquer módulo chamado por ele. A Figura 3.5 mostra a função do "módulo topo" de um programa de alocação de equipamentos de entrada e saída.



FIGURA 3.5 - ALOCAÇÃO DE PROGRAMAS

Quando da descrição da função de um módulo, os seguintes critérios devem ser observados:

- a) A descrição de função pode conter um verbo, por exemplo: "Encontrar registro chave" , "Executar comandos de acesso" e "Executar comandos de alocação".
- b) Evitar o uso de verbos que são ambíguos tais como: fazer, processar, executar etc.
- c) Evitar palavras que denotam lógica, tais como "CONTROLE", "ROTA" e "INTERFACE".

3.2.3 - MEDIDAS DE MÓDULOS

Um módulo ótimo é aquele em que as relações entre elementos no mesmo módulo não possam ser mais minimizadas. Destarte, há duas maneiras de se chegar a um módulo ótimo.

i - Minimizando as relações entre módulos, o que significa - em linhas gerais - reduzir as "interfaces" entre os módulos de um sistema, denominadas acoplagem de módulos.

ii - A maximização das relações entre os

elementos constituintes de um módulo individual, denomina-se eficácia de módulos.

A seguir serão descritas algumas medidas de eficácia e acoplagem de módulos:

3.2.3.1 - Eficácia de Módulos

Não há dúvida de que a eficácia de um módulo se constitui numa das principais medidas de um "Design", que em escala descendente compõe-se de funcional, informacional, comunicacional, procedimental, clássica, lógica e coincidível. Tais medidas estão intimamente ligadas à qualidade e custo de um programa em termos de extensão, manutenção e integridade.

Eficácia Coincidível - Quando não há relação significativa entre os elementos de um módulo trata-se de eficácia coincidível. Normalmente, tal situação origina-se de tres circunstâncias:

i - Um programa monolítico foi modularizadado partindo-se em módulos separados.

ii - Foram criados módulos, que consolidada

vam uma codificação duplicada em vários módulos.

iii - Um programa existente foi partido em blocos em face de problemas encontrados com o tamanho de memória da máquina utilizada.

Por exemplo: um trecho de um programa -

A = B + C

GET CARD

PUT OUTPUT

IF B = 4, THEN E = 0

- considerando-se que tal trecho apareça várias vezes num módulo, ou em vários módulos, um programador bem intencionado poderia, após analisar a situação, decidir criar um único módulo capaz de conter tais instruções, que poderia ser chamado de diversos pontos do programa utilizando-se uma instrução "CALL". Tal procedimento se constituiria num exemplo típico de eficácia coincidível.

Eficácia Lógica - Denomina-se módulo de eficácia lógica aquele que, em cada chamada, executa uma série de funções nele contida. A Figura 3.6, representa um exemplo que evidencia a eficácia lógica de um módulo, utilizando-se a declaração: "Edite todos os dados", que seria implementado

como se segue:

Os dados do Arquivo Mestre se constituem de registros que seriam atualizados, retirados e adicionados. A "interface" transmitiria parâmetros acompanhados de um indicador da operação a ser realizada. A primeira instrução conteria quatro ramificações, gerando quatro secções a saber: Edite Registro Mestre, Edite Registro de Atualização, Edite Registro de Adição e Edite Registro de Retirada.

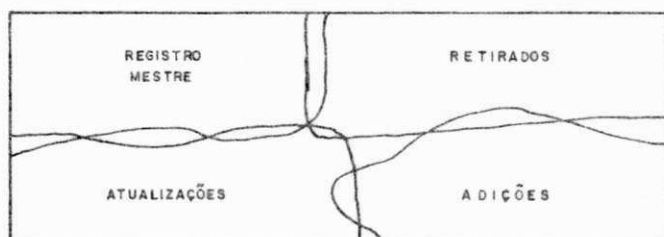


FIGURA 3.6 - MÓDULO "EDITE" TODOS OS DADOS. EXTRAÍDOS DE [3]

Como mostra a Figura 3.6, as quatro funções são entrelaçadas em algum lugar do módulo. O uso de módulos com eficácia lógica às vezes torna a codificação muito complicada, o que faz o problema difícil de ser modificado.

Eficácia Clássica - Um módulo com eficácia clássica guarda as mesmas características do módulo com eficácia lógica, excetuando-se o fato de que os elementos são relacionados com o tempo, tais como, tradicionais inicializações e terminações. Os elementos de um módulo de inicialização encontram-se logicamente, relacionados porque suas funções são executadas juntas e de forma sequencial.

Em face dos módulos, com eficácia clássica não necessitarem de lógica, nem de parâmetros que determinem quais os elementos que serão executados (em virtude dos elementos serem executados na sua ordem natural) tais módulos tornam-se simples de codificar. A Figura 3.7, mostra um exemplo de módulo de eficácia clássica.

Os módulos de eficácia clássica, tendem a ter um relacionamento muito íntimo com outros módulos, daí não serem altamente independentes. Um exemplo disto é: "INICIALIZA A TABELA DE SÍMBOLOS". Esta função tem uma relação com o módulo "ADICIONE ENTRADAS PARA TABELA DE SÍMBOLOS", porque ambas fazem parte da mesma estrutura de dados.

Eficácia Procedimental - Os módulos com eficácia procedimental - na maioria das vezes - representam o fluxo de um problema a ser resolvido, mostrando um ou mais blocos do programa. Apresentam-se muito similarmente aos módulos de eficácia clássica, excetuando-se no tocante ao fato de que as funções de um módulo de eficácia procedimental são



FIGURA 3.7 - MÓDULO DE EFICÁCIA CLÁSSICA. EXTRAÍDO DE [3]

apresentadas como "procedure" de um problema. Embora se apresente numa posição privilegiada na escala de eficácia, está muito aquém de ser o ideal, dada a sua relação muito íntima com a estrutura do problema.

Um método procedimental pode conter várias funções ou partes de uma função. Seja o módulo "SALTE O TOPO DA PÁGINA SEGUINTE E LEIA O REGISTRO SEGUINTE". Se em outra

situação somente a função "SALTE PARA O TOPO DA PÁGINA SEGUINTE E/OU LEIA O SEGUINTE REGISTRO" for necessária, a tendência do projetista será mudar a função original para "SALTE PARA O TOPO DA PÁGINA SEGUINTE E/OU LEIA O SEGUINTE REGISTRO". Tal decisão apresentar-se-ia indesejável porque teria que modificar o módulo e, em consequência, recairia num módulo de eficácia lógica.

Eficácia Comunicacional - O módulo com eficácia comunicacional além de ser um módulo de eficácia procedimental seus elementos se comunicam entre si. Em outras palavras, os elementos do módulo ou se relacionam entre si, ou transmitem dados entre si.

Sejam os exemplos envolvendo os Módulos A, B e C:

Módulo A : Atualize registro do "data base" e faça constar no "audit-trail".

Módulo B : Calcule nova trajetória e envie ao terminal.

Módulo C : Atualize registro no "data Base" e leia a transação seguinte.

O módulo "A" possui eficácia comunicacional, porque os elementos usam o mesmo conjunto de dados. O módulo "B" também possui eficácia comunicacional, porque o

primeiro elemento de saída (a trajetória) é a entrada do outro elemento (terminal). O módulo "C" possui eficácia procedimental, porque os elementos do processo não são do mesmo conjunto de dados.

Eficácia Funcional - Na escala de eficácia, encontra-se no topo a eficácia funcional. Neste tipo de môdulo, todos os elementos são relacionados, constituindo uma função. Em tal caso, uma questão que sempre surge: qual é a função?

Em matemática $y = f(x)$ lê-se, "y é função de x". "f" define uma transformação da variável independente "x", na variável y. Por isso uma função descreve uma transformação de algum dado de entrada para algum dado de saída. Em termos de programação esta definição é expandida para permitir funções sem dados de entrada ou sem dados de saída.

Os módulos abaixo representam exemplos de eficácia funcional.

Módulo A : Compute raiz quadrada

Módulo B : Obtenha números randômicos

Módulo C : Escreva registro no arquivo de Saída.

Módulo D : Retire registro do arquivo mestre.

O módulo "A" define uma função com uma entrada e uma saída. O módulo "B" define uma função com uma saída mas não possui entrada. Os módulos "C" e "D" são funções com parâmetros de entrada sem parâmetros de saída.

Eficácia Informacional - O módulo em apreço representa pacotes fisicamente juntos (dentro do módulo) de dois ou mais módulos de eficácia funcional. Considerando se dois módulos, ambos de eficácia funcional tais como: "INSERIR ENTRADAS NA TABELA DE SÍMBOLOS", e "PESQUISE PARA ENTRAR NA TABELA DE SÍMBOLOS" os quais possuem a mesma estrutura de dados, se juntarmos fisicamente em um módulo com dois pontos de entrada, onde cada ponto representa uma função, obtém-se um módulo de eficácia informacional, com a vantagem de se manuseiar a estrutura de dados (Tabela de Símbolos) num único módulo.

O módulo informacional deve observar os seguintes critérios: fazer múltiplas funções; cada função deve ser representada por um ponto de entrada no módulo; cada ponto de entrada deve ter as características de um módulo de eficácia funcional, e todas as funções devem fazer alguma transformação na mesma estrutura de dados. A Figura 3.8, mostra um módulo de eficácia informacional.

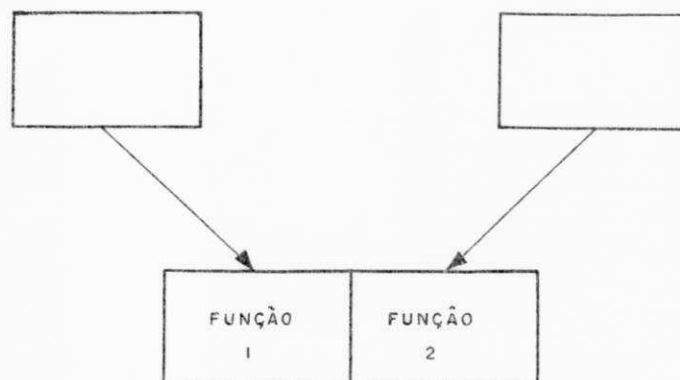


FIGURA 3.8 - MÓDULO DE EFICÁCIA INFORMACIONAL

**Determinação da Categoria de Módulos Com
Relação à Sua Eficácia** - Como se sabe, um

módulo se constitui de uma sentença que deverá ser analisada, a fim de se verificar de que tipo de eficácia se constitui. Para tal deve-se proceder com os testes abaixo discriminados:

- a) Se a sentença é composta, contém uma vírgula, ou mais de um verbo, o módulo provavelmente se constitui de mais de uma função. Não tem eficácia funcional. Se possui mais de um verbo, e são conectados por "ou", então, o módulo, provavelmente, tem eficácia lógica. Se os verbos forem conectados com "E" então, o módulo deve ter eficácia clássica, procedimental ou comunicacional.

- b) Se a sentença contém palavras relacionadas com ordem, tais como: primeiro, seguinte, então, depois, quando e início, então, o módulo tem eficácia procedimental.
- c) Se o predicado da sentença não contém um objeto seguido do verbo, então, o módulo provavelmente tem eficácia lógica.
- d) Palavras tais como: inicialize CLEAN-UP, implicam em eficácia funcional.
- e) Em determinados casos, pode-se ter dificuldade em verificar de que tipo de eficácia se constitui em módulo, em tal situação examinam-se as desvantagens que o módulo possui, e possivelmente, encontrar-se-á um tipo de eficácia que se enquadre ao módulo em apreço.

A Figura 3.9, representa um programa que se propõe a atualizar um "arquivo de clientes". A entrada para o primeiro módulo é o registro que contém os campos (número, nome, endereço, "status" e data da compra). O registro do "Arquivo Cliente" contém campos acima, seguidos do Nº de registro de vendedor, o qual - por sua vez - contém a data da venda, àquele cliente em particular.

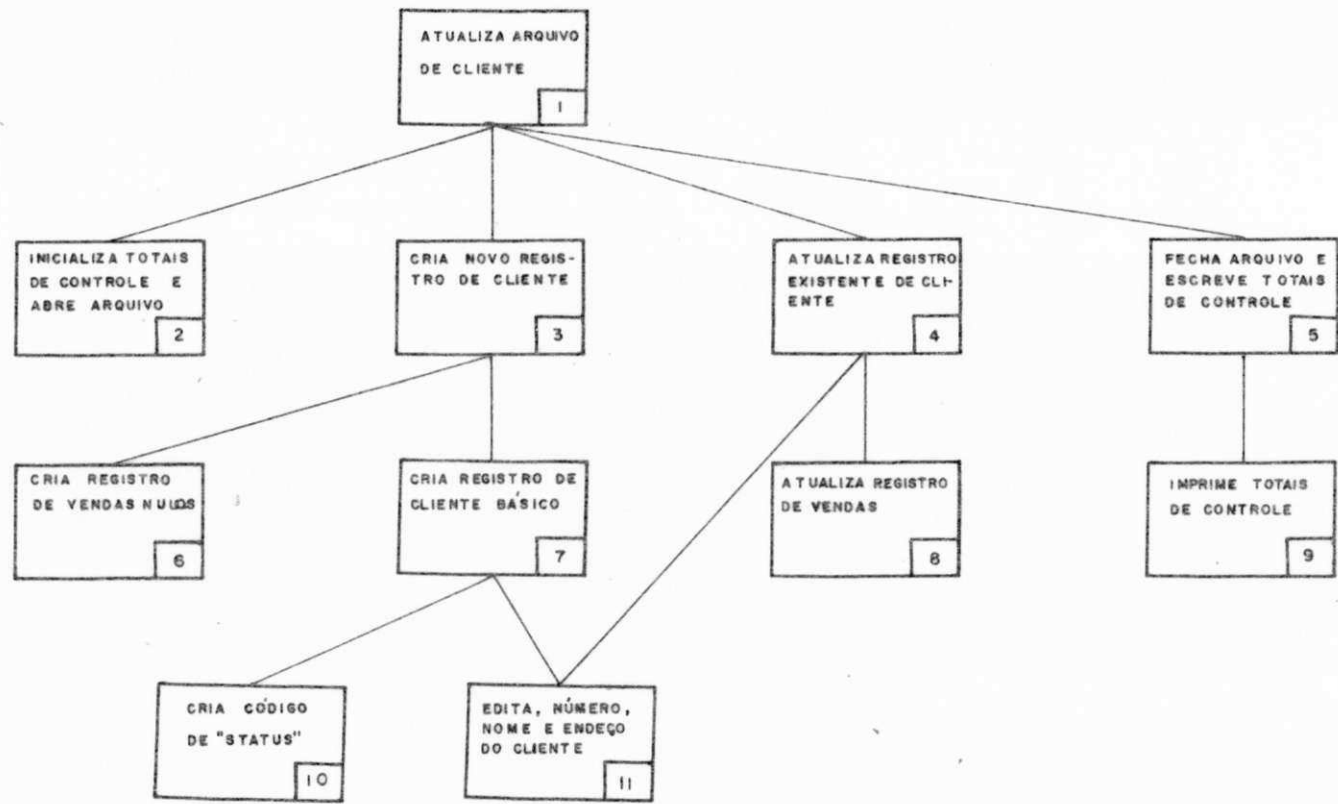


Fig 3.9 PROGRAMA A SER ESTUDADO (EXTRAIDO DE 3)

D I S C U S S Ã O

- o módulo - 2 tem eficácia clássica
- o módulo - 4 tem eficácia lógica, de uma vez que, executa uma classe de funções apresentadas logicamente. (atualização de registros).
- o módulo - 5 tem eficácia clássica, embora o elemento "IMPRIMA TOTAIS DE CONTROLE" seja procedimental, o outro elemento, "FECHA ARQUIVO" não o é.
- o módulo - 11, pode - a primeira vista - parecer possuir eficácia lógica, contudo, faz uma simples função em três conjuntos específicos de dados. Desta forma tem eficácia funcional.
- os demais módulos tem eficácia funcional.

Tabelas de decisão podem ser usadas como instrumentos capazes de ajudar na tarefa de determinar a eficácia de um módulo. A Tabela 1, é um tipo que pode ser usado.

A Tabela 2, sumariza os atributos de cada

tipo de eficácia, os quais são valiosíssimos como instrumentos utilizados na medição do grau de modularidade de um "design" de sistema.

3.2.3.2 - Acoplamento de Módulos

Num "design" de sistema os módulos se relacionam mutuamente através de suas "interfaces". A esta ligação dá-se o nome de ACOPLAMENTO DE MÓDULOS. Sabe-se que uma das características de um bom "design" é, sem dúvida, a independência entre módulos. Destarte conclui-se que os módulos de um sistema devem se ligar com os demais o mínimo possível, a fim de se obter um bom grau de modularidade quando do "design".

Um "design" altamente modularizado apresenta minimizada a relação entre módulos e, maximizada a relação entre elementos de um módulo.

A seguir serão apresentadas algumas medidas da ligação entre módulos que serão fundamentais na determinação do grau de modularidade de um sistema.

Acoplamento de Conteúdo - A medida Acoplamento de Conteúdo se fundamenta nas seguintes situações:

DIFÍCIL DE DESCREVER A FUNÇÃO DO MÓDULO	S	N	N	N	N	N	N	N
MÓDULO FAZ MAIS QUE UMA FUNÇÃO			S	S	S	S	S	N
FAZ SOMENTE UMA FUNÇÃO POR CHAMADA			S	N	N	N	S	
CADA FUNÇÃO TEM UM PONTO DE ENTRADA			N					
MÓDULO RELACIONA AS CLASSES DAS FUNÇÕES		N	S	S				
AS FUNÇÕES SÃO RELACIONADAS PELAS "PROCEDURE" DO PROBLEMA				N	S	S		
TODAS AS FUNÇÕES USAM OS MESMOS DADOS					N	S	S	
COINCIDÍVEL	X	X						
LÓGICA			X					
CLÁSSICA				X				
PROCEDIMENTAL					X			
COMUNICACIONAL						X		
INFORMACIONAL							X	
FUNCIONAL								X

TABELA 1 - DETERMINAÇÃO DE EFICÁCIA. EXTRAÍDA DE [3]

EFICÁCIA	INDEPENDÊNCIA DE OUTROS MÓDULOS	SUSCEPTIBILIDADE DE ERROS	USABILIDADE EM OUTROS PROGRAMAS	EXTENSIBILIDADE
Funcional	Alta	Baixa	Alta	Alta
Informacional	Alta	Média	Alta	Alta
Comunicacional	Média	Baixa	Média	Média
Procedimental	Média	Baixa	Baixa p /médio	Média
Clássica	Baixa	Média	Baixa	Média
Lógica	Média	Alta	Média	Baixa
Coincidência	Baixa	Muito Alta	Baixa	Baixa

TABELA 2 - ATRIBUTOS DE EFICÁCIA

i - Um módulo modifica um comando de programa em outro módulo.

ii - Dois módulos partilham do mesmo conteúdo, quando comandos de um módulo estão fisicamente dentro de outro módulo, ou dois módulos residem fisicamente numa entidade compatível.

Em consequência destas situações pode ocorrer que uma troca efetuada em um módulo implique em modificações no módulo ao mesmo acoplado.

Acoplamento Comum - Se vários módulos encontram-se partilhando da mesma "estrutura de dados global", existirá uma medida de acoplamento comum entre módulos. Tal situação ocorre, quando do emprego das declarações de linguagens de alto nível a saber:

- Comando COMMON em Fortran
- Atributo EXTERNAL

O acoplamento comum causa tres deficiências nos módulos que são acoplados comumente:

- a) Uma modificação em um dos módulos pode afetar todos os demais módulos acoplados comumente.

b) Se, se tiver um programa com 10 módulos e cada um deles contiver a declaração em PL/1, % INCLUDE P DATA, instruindo ao computador para copiar em cada módulo a estrutura:

```
DCL  1 P DATA EXTERNAL,  
    2 VENDEDOR FIXED,  
    2 CATEGORIA CHAR (4)  
    2 MATCH  FIXED  BINARY
```

Quando se desejar fazer a troca de CHAR (4) de CATEGORY por CHAR(6) em um módulo ter-se-a que recompilar todos os 10 módulos.

Uma forma de se evitar o acoplamento comum é limitar o número de módulos com "ambiente comum".

Acoplamento Externo - A medida de acoplamento externo é similar a de acoplamento comum, excetuando se o fato de que, no acoplamento externo, as referências externas são feitas entre "data itens" individuais e não entre estruturas de dados.

Acoplamento de Controle - Diz-se que dois

módulos tem medição de acoplamento de controle, se um módulo envia elementos de controle como argumento para outro módulo, e tal elemento influencia diretamente a execução do módulo chamado. Tal tipo de acoplamento é desejável pelo fato de diminuir a independência de módulos.

A Figura 3.10, mostra um exemplo de acoplamento de controle, porque o argumento ANALISE é uma chave que instrui ao módulo GET COMM a devolver COMANDO analisado ou não.

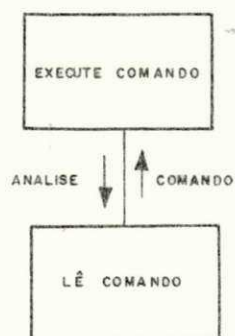


FIGURA 3.10 - ACOPLAMENTO DE CONTROLE

Acoplamento "Stamp" - Se dois módulos encontram-se referenciados na mesma estrutura de dados, e se esta estrutura não é global, tem-se aí uma medida de acoplamento "STAMP", tal acoplamento apresenta-se de forma similar ao acoplamento comum, excetuando o fato do ambiente comum.

O acoplamento comum, apresenta, dentre outras, as seguintes desvantagens:

- a) Uma troca aparentemente isolada numa es

estrutura, pode afetar os demais módulos.

- b) O acoplamento comum choca com a meta, que visa restringir o acesso a um módulo por um único "data ítem".
- c) Em face dos dados serem globalmente compreendidos, torna-se difícil usar um módulo de acoplamento comum em outros programas.

O acoplamento "STAMP" não resolve todos os problemas, mas pode reduzir os efeitos das duas últimas desvantagens.

A desvantagem "b" pode ser minorada se o acesso de dados for feito por uma estrutura isolada.

A desvantagem "c" seria reduzida em face de o programa poder manuseiar várias estruturas de dados.

Acoplamento de Dados - Se dois módulos não possuem acoplamento de controle, "Stamp", externo ou comum, forçosamente possuirão acoplamento de dados. Em outras palavras, todos argumentos são elementos de dados e não elementos de controle ou estrutura de dados.

Na escala de medidas de acoplamento encon

tra-se o acoplamento de dados com menor grau. Desta forma, módulos que possuem acoplamento de dados são altamente independentes. Neste caso os módulos podem ser vistos como uma "caixa preta".

O acoplamento de dados é, sem dúvida, uma situação desejada por um projetista de "design", porque tal acoplamento permite uma boa modularidade de "design".

Acoplamento de Dados Versus Controle - Em muitos casos é simples distinguir um acoplamento de controle de um acoplamento de dados, bastando - para isso - fazer um rápido exame nos parâmetros de passagem entre módulos, utilizando-se duas regras práticas:

- a) A classificação do parâmetro (controle e dados) depende de como o módulo que o envia percebe, e não de como o percebe o módulo recebedor.

A Figura 3.11, apresenta um exemplo ilustrativo, onde o módulo "A" passa o parâmetro "X" para o módulo "B" e "A" o percebe como um dado, implicando em existir um acoplamento de dados entre os módulos "A" e "B". Mas se o módulo "B", perceber o parâmetro "X" como um controle, haverá - neste caso - um acoplamento de controle entre módulos.

O mesmo raciocínio aplica-se a informações

que retornam de um módulo, tal como: um código de retorno ou "flag" de erro.

b) A informação de controle é usualmente criada de forma artificial.

Se o módulo "A" passa um comando para o módulo "B" (que o examina e determina como processá-lo) então, "A" e "B" possuem acoplamento de controle.

Se o módulo "A" passa um comando para o módulo "B" e, em adição, passa um código dizendo "PROCESSE O COMANDO "XYZ", então "A" e "B" possuem acoplamento de controle.

Apontador de Passagem ou Endereço - Em algumas linguagens de programação, é possível se passar apontadores ou endereços de dados como um parâmetro explícito. Por exemplo: em PL/I isto pode ocorrer.

Quando se analisar um acoplamento numa situação onde apontadores ou endereços, se fazem presentes, a análise não se baseia no apontador, e, sim, para onde o mesmo está apontando. Como exemplo, imagina-se "P" um apontador que aponta para uma estrutura "S". O módulo "A" chama o módulo "B", passando "P" como um parâmetro, logo, os módulos "A" e "B" possuem acoplamento "STAMP".

Tabelas de decisão podem ser usadas como

instrumentos capazes de ajudar na tarefa de determinar o grau de acoplamento entre módulos. A Tabela "3" é um exemplo de tais Tabelas.

A Tabela 4 sumariza os atributos de cada tipo de acoplamento, que são valiosíssimos como instrumento utilizado na medição do grau de acoplamento entre módulos de um "design" de sistema.

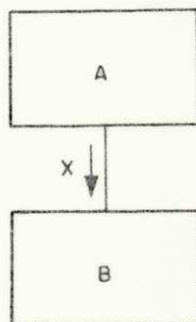


FIGURA 3.11 - ACOPLAMENTO DE DADOS VERSUS CONTROLE

REFERENCIA DIRETA ENTRE MODULOS	S		N	N	N	N	N
MÓDULOS SÃO CONTIDOS JUNTO		S	N	N	N	N	N
ALGUNS DADOS DE "INTERFACE" SÃO EXTERNOS OU GLOBAL			S	S	N	N	N
ALGUNS DADOS DE "INTERFACE" SÃO INFORMAÇÕES DE CONTROLE					S	N	N
ALGUNS DADOS DE "INTERFACE" ESTÃO NUMA ESTRUTURA DE DADOS			S	N		S	N
ACOPLAMENTO DE CONTEÚDO	X	X					
ACOPLAMENTO COMUM			X				
ACOPLAMENTO EXTERNO				X			
ACOPLAMENTO DE CONTROLE					X		
ACOPLAMENTO "STAMP"						X	
ACOPLAMENTO DE DADOS							X

TABELA 3 - DETERMINAÇÃO DE ACOPLAMENTO. EXTRAÍDA DE [3].

ACOPLAMENTO	INDEPENDÊNCIA COM OUTROS MÓDULOS	SUSCEPTIBILIDADE DE ERROS	USABILIDADE EM OUTROS PROGRAMAS	EXTENSIBILIDADE
Dados	Alto	Baixo	Alto	Alto
"Stamp"	Médio	Médio	Médio	Médio
Controle	Médio	Médio	Médio	Médio
Externo	Baixo p/médio	Alto	Baixo p/médio	Baixo
Comum	Baixo	Alto	Baixo	Baixo
Conteúdo	Baixo	Alto	Baixo	Baixo

TABELA 4 - ATRIBUTOS DE ACOPLAMENTOS.

3.3 - "TOP-DOWN DESIGN"

O "Top-down design" se apresenta como das mais naturais abordagens para o "design" de sistemas. Consiste, contudo, em se identificar as maiores funções de um sistema e, se ir decompondo-as em níveis de refinamento sucessivos, até que o nível de linguagem de programação seja encontrado. Nos refinamentos sucessivos, faz-se uso dos conceitos de níveis de abstração e máquina virtual. O "Top-down design", resulta num conjunto de componentes relacionados, e, a cada nível do design, tem-se uma abstração das especificações de um componente e refina-se aquela abstração em componentes menores e mais detalhados, começando-se por identificar as funções mais importantes, ou seja, aquelas que tenham operações comuns, e passa-se a refiná-las sucessivamente até se chegar ao nível das estruturas de dados necessários e das "procedures" requeridas neste nível. Cada estrutura de dados pode sofrer novo refinamento até se chegar à linguagem de programação necessária.

Algumas sugestões importantes para obtenção de um bom "Top-down design" serão aqui enumeradas para melhor orientação do projetista.

i - Considerando que, cada módulo, possui entrada, transformação e saída (conforme mostra a Figura 3.12)

o projetista deve se esforçar no sentido de bem determinar a entrada, transformação e a saída de cada módulo.

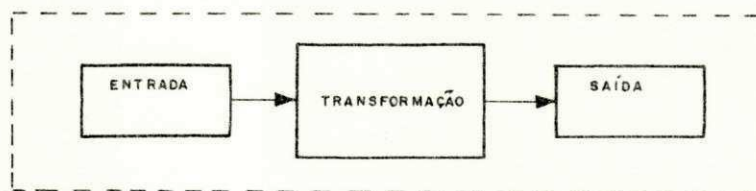


FIGURA 3.12 - ELEMENTOS DE UM MÓDULO

ii - Tendo o projetista se convencido de que uma função particular do problema pode ser contida num módulo, então passa a refiná-la sem se importar com o que acontece em torno do módulo.

iii - Cuidados devem ser tomados no sentido de que um aspecto trivial do problema não venha confundir o projetista.

iv - A cada nível do "design" o projetista deve estar atento para o fato de um módulo se constituir ou não de uma página de codificação ou uma página de fluxograma.

v - Muita atenção deve ser dispensada aos dados, bem como ao projeto de processos ou algoritmos. Em muitos casos, os dados são as "interfaces" dos módulos, e, o projetista dos módulos não se deve aprofundar sem que estas "interfaces" sejam cuidadosamente especificadas.

As "interfaces" ou especificações de entrada e saída e funções de um módulo são consideradas como elementos importantes, na abordagem de um "top-down design".

Uma abordagem de "Top-down design", feita sem muito critério pode dar ao projetista um falso senso de confiança. A fim de se formalizar os passos de um "top-down design" muitas organizações tem se esforçado no sentido de desenvolver métodos para tal. Uma das técnicas mais interessantes e um amplo exemplo de como se formalizar um "top-down design" foi desenvolvida pela IBM, denominada HIPO (Hierarchical - Input - Output - Process) permitindo que as várias estruturas do sistema sejam mostradas em um diagrama de visualização como mostrado na Figura 3.13. Cada módulo no diagrama de visualização é descrito adicionalmente, num diagrama como mostra a Figura 3.14. Se necessário, diagramas com níveis de detalhamento mais baixos podem ser mostrados. O diagrama de detalhe visto na Figura 3.14, tem como propósito dar uma simples e breve descrição de uma função particular, onde as especificações de entrada e saída são identificadas e associadas em bloco de processo que as utilizam.

O método HIPO é um instrumento a ser usado num "top-down design", embora várias outras abordagens de formalização tenham sido apresentadas por STEVEN, MAYERS. Como também a representação gráfica de CONSTANTINO. Melhores detalhes na referência bibliográfica 7.

A abordagem apresentada por MAYERS denominada Análise Composta é assunto do próximo Capítulo.

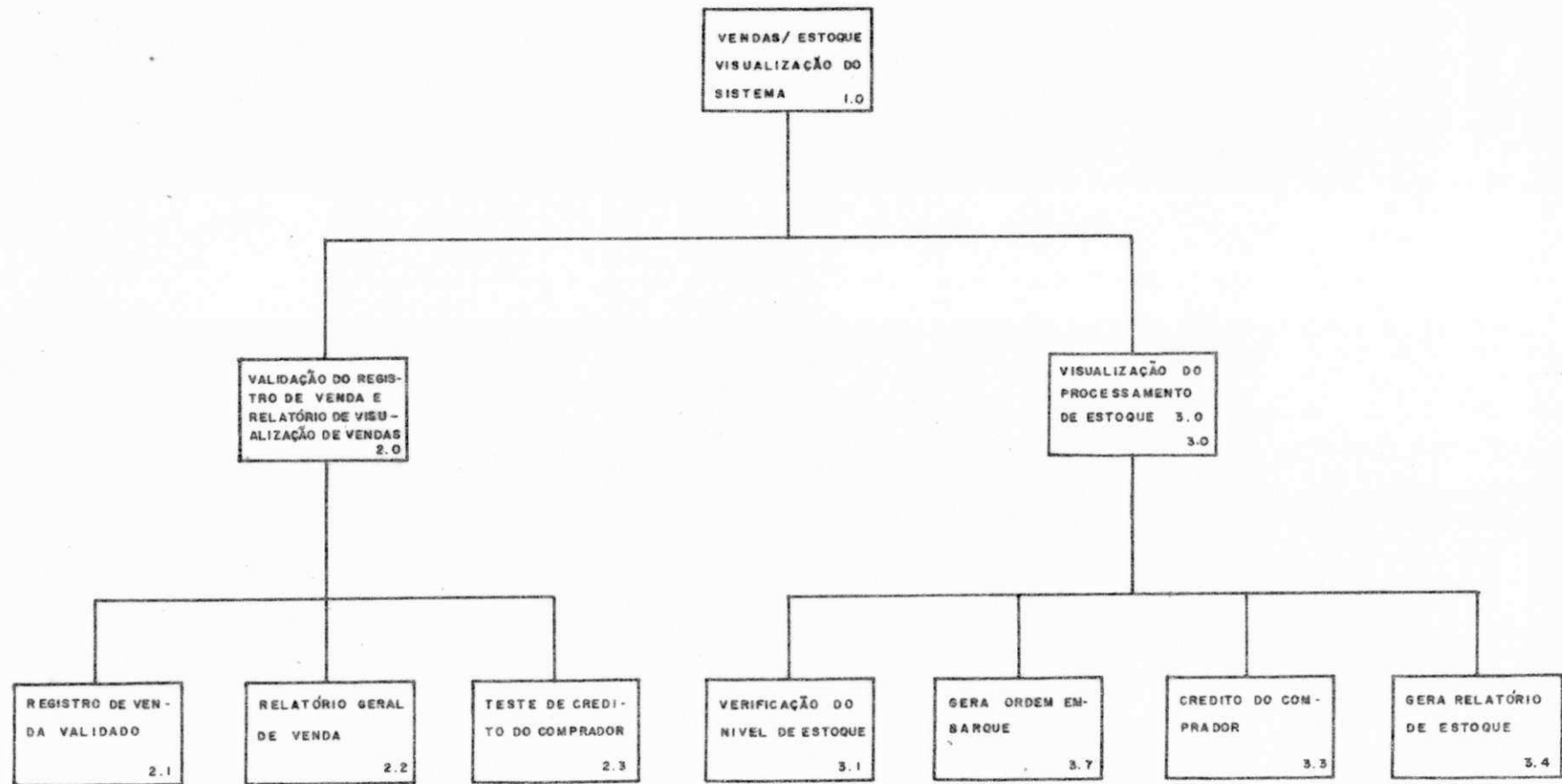


Fig. 3.13 (a) DIAGRAMA HIERARQUICO

- 1.0 - O Diagrama do Sistema de venda/estoque que processa ordem de venda e mantém o arquivo estoque, gera ordem de embarque, crédito aos compradores e produz relatórios de estoque e venda.
- 2.0 - O Diagrama de visualização de validação de ordem de venda, verificação de crédito de comprador e de geração de relatório de venda.
 - 2.1 - Diagrama detalhado de validação de ordem de venda e ordenação, por número de item de estoque, do arquivo transação.
 - 2.2 - Diagrama detalhado do relatório de venda, que sumariza arquivo transação por região de venda e por vendedor e computa comissões de vendas.
 - 2.3 - Diagrama detalhado de verificação de crédito do comprador, computação de desconto e determinação de modalidade de pagamento.
- 3.0 - Diagrama de Visualização do processamento de transações de estoque, relatórios e administração de estoque, créditos, descontos e embarques.
 - 3.1 - Diagrama detalhado de verificação de estoque, atende ao pedido de venda, ou solicita completação de estoque se o mesmo for insuficiente.
 - 3.2 - Diagrama detalhado de criação da ordem de embarque e geração de ordem de embalagens.
 - 3.3 - Diagrama detalhado de crédito de comprador, geração e operação do arquivo de contas a receber.
 - 3.4 - Diagrama detalhado da geração do relatório de estoque com entradas categorizadas por região de venda.

FIGURA 3.13(b) - SECÇÃO DE DESCRIÇÃO DO SISTEMA VENDA/ESTOQUE.

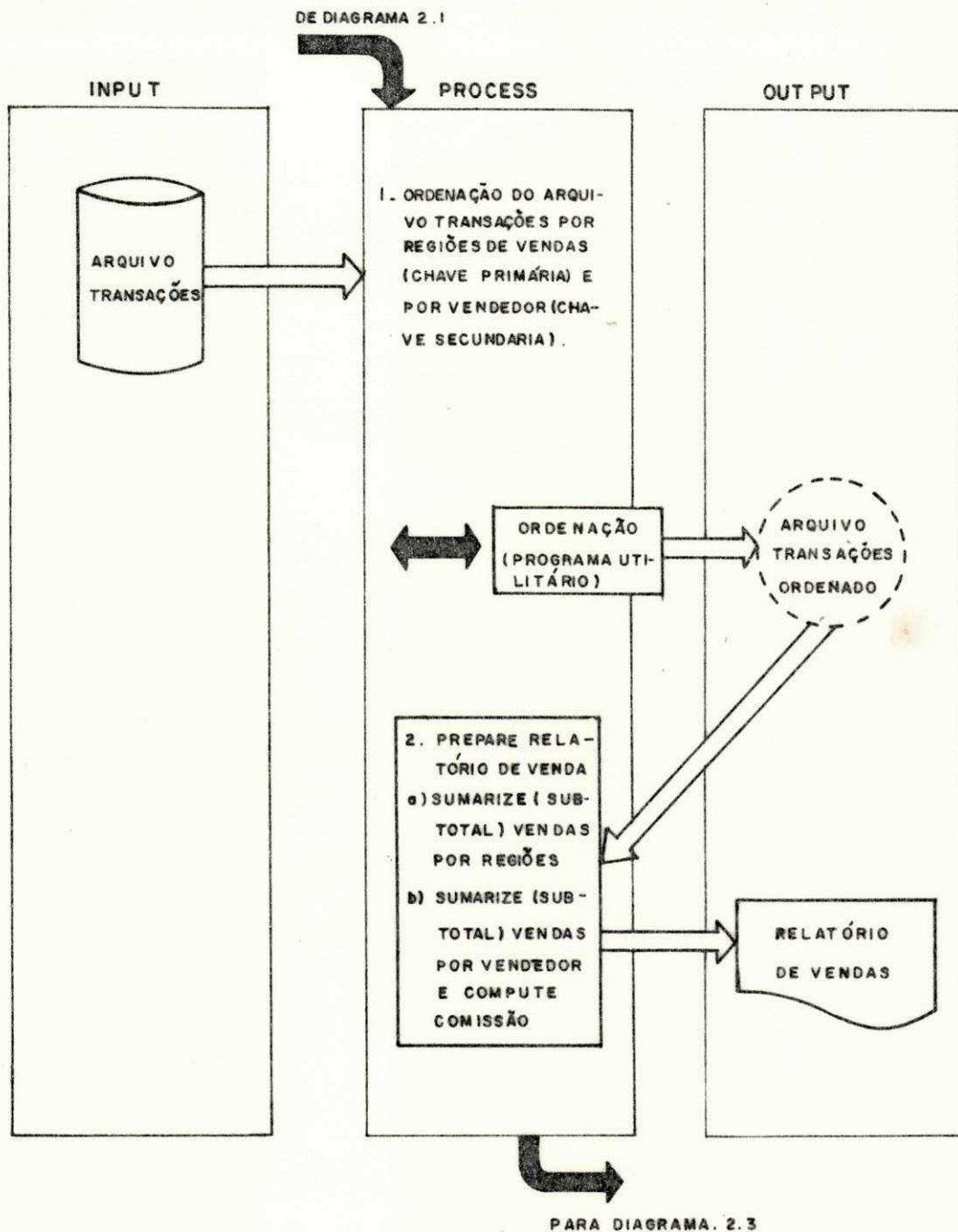


Fig. 3.14- DIAGRAMA DETALHADO 2.2, HIPO (ESTE DIAGRAMA CORRESPONDE AO "RELATÓRIO GERAL DE VENDAS")

3.4 - "BOTTOM-UP"

No desenvolvimento do "bottom-up" a ordem de integração, teste e codificação dos módulos início da base para o topo. A estrutura apresentada na Figura 3.15 - se desenvolvida por "bottom-up" - teria os módulos "E" e "F" codificados e testados em paralelo. Em seguida, os módulos "B" e "C" seriam, também, codificados e testados em paralelo. E, finalmente, o módulo "A" seria codificado e testado, usando-se os módulos "B", "C", "E" e "F" como subordinados.

O desenvolvimento "bottom-up" fundamenta-se no fato de que um módulo já testado, torna-se uma função primitiva que poderá ser antes construída.

Apresenta, assim, o desenvolvimento "bottom up" as seguintes vantagens:

- a) Pode-se encontrar mais paralelismos nas fases de codificação e teste.
- b) A integração de um ciclo torna-se simples. Contudo, o maior ramo de um programa não estaria completamente testado sem que um ciclo estivesse completo.
- c) Um ou mais teste pode ser inteiramente

completado. Contudo, no desenvolvimento "top-down", o módulo "E" para ser testado teria suas entradas fornecidas pelo módulo "A".

Sintetizando, o desenvolvimento, "bottom up" é uma técnica de implementação e não uma estratégia de "design". As vantagens e desvantagens do desenvolvimento "Bottom-up" e "top-down", não são bastantes conclusivas, para indicar que um é melhor que o outro. Diz-se-ia, contudo, que, no desenvolvimento "top-down", as "interfaces" se apresentam mais bem definidas. Contudo, ambas as metodologias de desenvolvimento podem ser empregadas com sucesso.

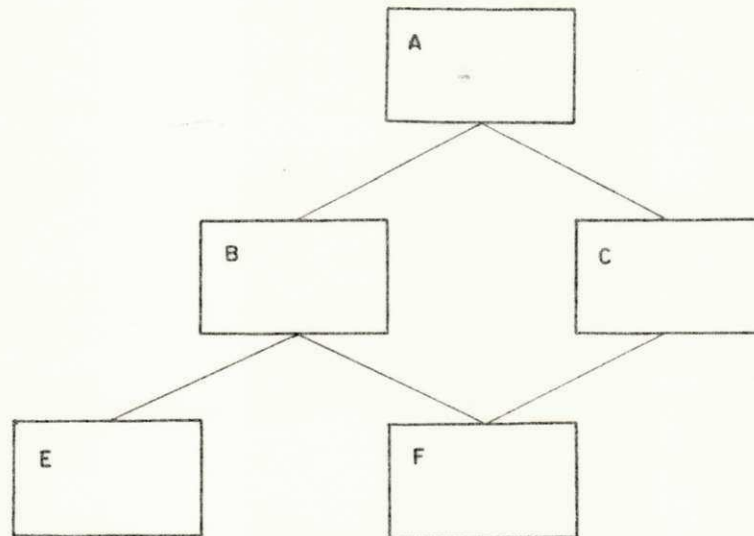


FIGURA 3.15 - ILUSTRAÇÃO UTILIZADA NO DESENVOLVIMENTO "BOTTOM-UP"

CAPÍTULO 4

ANÁLISE COMPOSTA

A metodologia Análise Composta tem a filosofia de "top-down", envolvendo interações entre secções da estrutura do problema, suprindo funções e, ao mesmo tempo, gerando novas funções para completar as secções da estrutura do problema.

Como é de notar, para aplicação da metodologia Análise Composta, necessário se faz a determinação da estrutura do problema. Tal determinação torna-se possível, utilizando-se a metodologia Projeto Composto que - por sua vez - se inicia com:

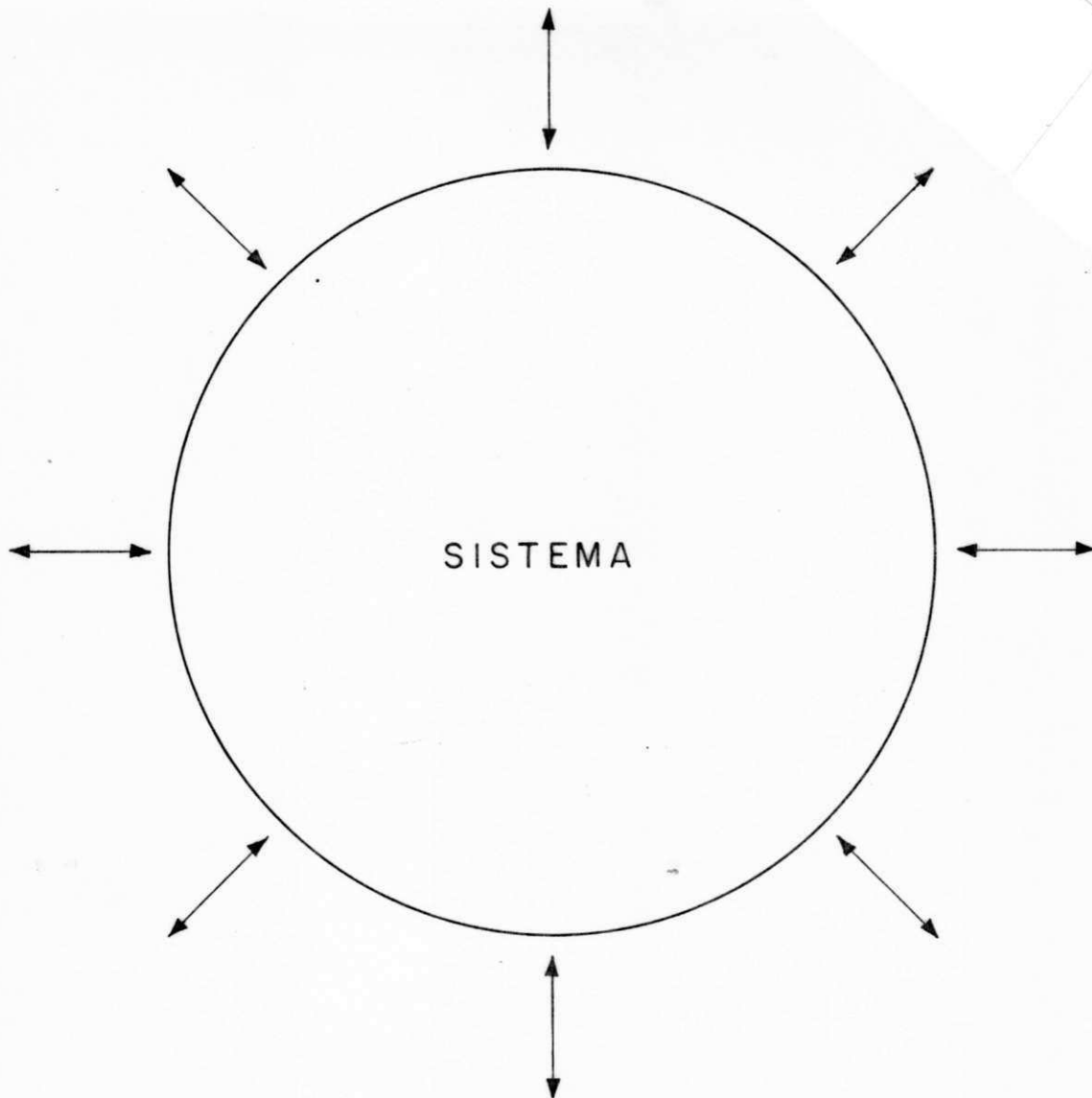
- 1) Projeto Externo do Processo se constituindo em especificar as características externas do sistema.
- 2) Projeto Estrutural do Processo que - por sua vez - se constitui do seguinte:

i) Inicia-se com a declaração do problema (ou especificação externa). Em seguida, projeta-se a estrutura do problema, utilizando-se uma forma convencional de análise, e, se necessário, decompõe-se o sistema em módulos.

ii) Revisa completamente a estrutura do sistema, com vista nas medidas apresentadas no Capítulo 3, a fim de que os referidos módulos - na medida do possível - tenham suas eficácias maximizadas, e seus acoplamentos minimizados.

iii) Procede-se com as atividades, observando-se as seguintes fases:

a) Definição Precisa das Características Externas do Sistema
(somente para grandes sistemas).



Fase: Projeto Externo

Metodologia: Não existe metodologia específica. Necessitando, contudo, de experiência em Análise de Sistemas.

Saída: Descrição das características externas (entradas, saídas, funções, fatores humanos) do sistema

Fig. 4.1 - Projeto Externo. |3|

b) Definição da Arquitetura Interna do Sistema.



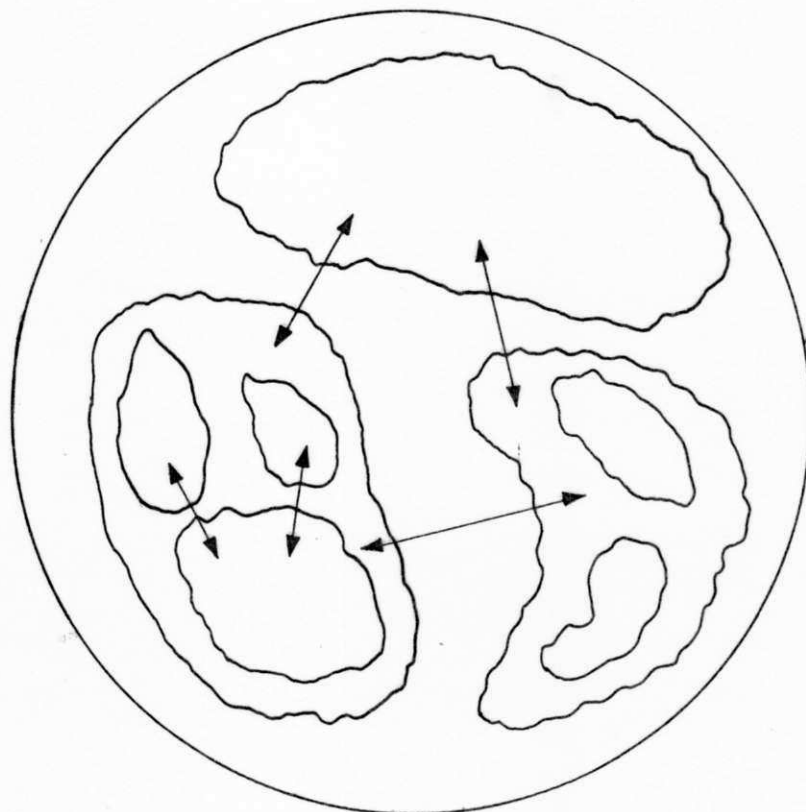
Fase: Projeto do Sistema

Metodologia: Não existe metodologia específica. O sistema é usualmente subdividido em componentes e subcomponentes.

Saída: Determinação dos componentes e subcomponentes.

Fig. 4.2 - Arquitetura do Sistema. |3|

c) Projeto das "Interfaces" dos Subcomponentes.



Fase: Projeto Lógico

Metodologia: Projeto Composto (uso de medidas para minimizar os acoplamentos)

Saída: Descrição precisa das "Interfaces" dos componentes e subcomponentes do sistema.

Fig. 4.3 - Projeto dos componentes e subcomponentes do Sistema. | 3 |

4.1 - METODOLOGIA ANÁLISE COMPOSTA

Após a obtenção do projeto estrutural do sistema como um todo, aplica-se a metodologia Análise Composta a cada subsistema, observando-se os seguintes passos:

- i) Identificar a estrutura do problema.
- ii) Identificar o principal fluxo de entrada e saída de dados no subsistema ou problema.
- iii) Identificar os pontos na estrutura do problema ou subsistema, onde os fluxos de dados se iniciam e terminam.
- iv) Decompor o problema ou subsistema em um conjunto de módulos subordinados.
- v) Repetir o processo, verificando cada módulo do passo anterior, como um novo subproblema.

Ressalte-se, contudo, que, a metodologia Análise Composta não é rígida - de vez que - dois projetistas podem desenvolver um mesmo problema e chegar a duas estruturas diferentes porém ótimas.

A seguir, será feito um estudo mais detalhado de cada passo acima descrito, como também, serão apresentadas algumas normas adicionais importantes ao entendimento da metodologia Análise Composta.

4.1.1 - ESTRUTURA DO PROBLEMA

Para definição da estrutura do problema, inicia-se esboçando-o num nível de abstração mais alto e em forma funcional, numa cadeia linear, composta em média de tres a dez processos. A título de exemplo apresenta-se um simples sistema de reserva de passagens aéreas, o qual tem a estrutura na Figura 4.4.

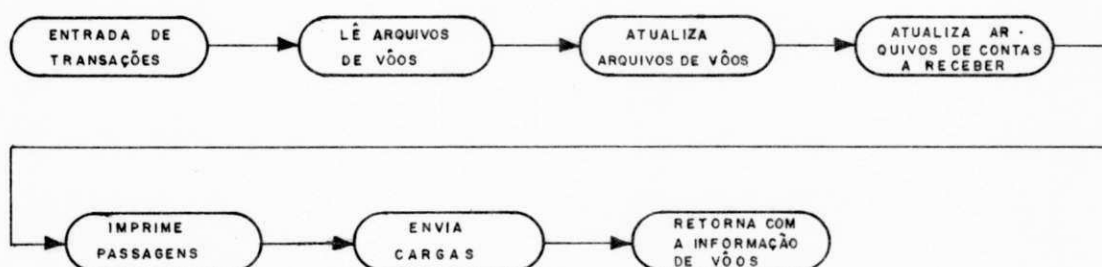


FIGURA 4.4 - ESTRUTURA DO PROBLEMA

4.1.2 - FLUXO DE DADOS EXTERNOS

Constitui-se numa importante tarefa, a identificação dos fluxos de dados denominados: externos e conceituais.

No Sistema de Reservas de Passagens Aé reas, as entradas de transações, passagens, informações de vôos, carga e ajuste de data, são exemplos de fluxos de da dos conceituais.

4.1.3 - PONTOS DE ALTA ABSTRAÇÃO

Na maioria dos problemas, há sempre um maior fluxo de dados de entrada (fonte primária de entrada) e um maior fluxo de dados de saída (resultados primários). A partir dos fluxos identificados no passo descrito anteriormente, identifica-se o maior fluxo de dados conceituais ex ternos (ambos, entrada e saída) no problema, e usando o dia grama de estrutura do problema, determinam-se os pontos de maior abstração desses fluxos. A Figura 4.5, representa um fluxo de dados, atravessando uma estrutura de problema, onde o maior fluxo de dados é visto passando através da estrutura do problema, trocando sua forma. Seguindo a trajetória do flu xo, observa-se que os dados vão se tornando mais abstratos até onde parecem desaparecer. Ao ponto onde isto ocorre deno mina-se "Ponto de Mais Alta Abstração da Cadeia de Entrada".

Fato idêntico acontece com a cadeia de da dos de saída. Cada problema abordado é visto como possuidor de uma maior cadeia de entrada de dados e uma maior cadeia de saída de dados, apresentando-se - às vezes - nas mais va riadas formas. No caso particular do Sistema de Reservas de Passagens Aéreas, as transações de entrada se apresentam co mo palavras faladas (do comprador para o balconista) introduzidas no terminal e recebidas pelo computador na forma digi tal. Em seguida é registrada e levada a exame para, poste riormente, passar através da estrutura, ou ser armazenada nu ma Tabela interna.

Voltando ao Sistema de Reservas de Passa gens Aéreas, observa-se que o ponto de maior abstração do fluxo de transações de entrada é o de validade das transaçõ es de entrada que se encontram armazenadas numa tabela inter na.

Não há dúvidas de que os pontos de maior abstração são determinados pelos maiores fluxos de entrada e saída. E, toda informação contida na estrutura do problema entre esses dois pontos é chamada de TRANSFORMAÇÃO CENTRAL DO PROBLEMA, como ilustra a Figura 4.6.

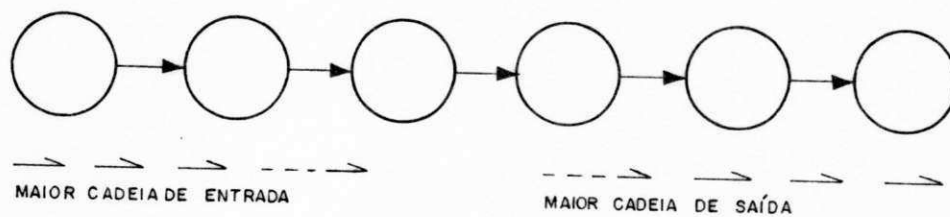


FIGURA 4.5 - ABSTRAÇÃO DE DADOS NA ESTRUTURA DO PROBLEMA

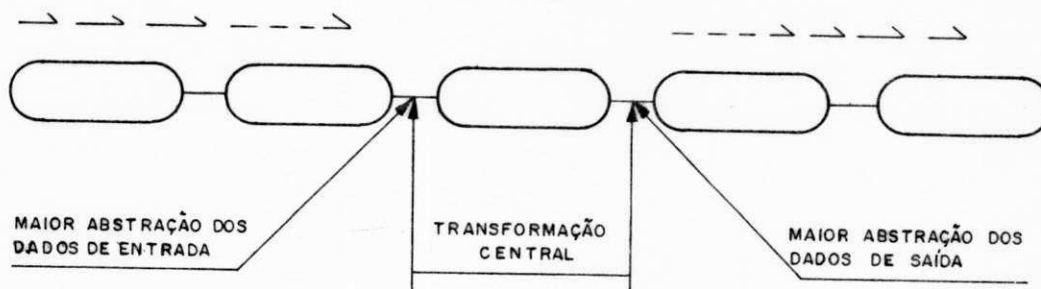


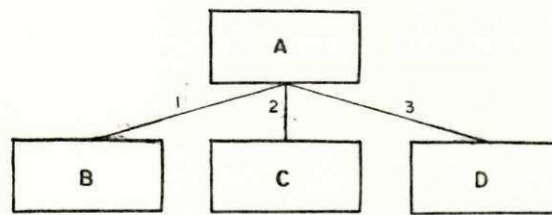
FIGURA 4.6 - PONTOS DE ALTA ABSTRAÇÃO

4.1.4 - DECOMPOSIÇÃO

A decomposição consiste em se dividir um problema em tres partes denominadas módulos subordinados e, escrever cada parte como uma função, conforme ilustra a Figura 4.8, onde as funções dos tres módulos subordinados, representam as tres partes do problema. A descrição das funções subordinadas, devem ser feitas de forma concisa, utilizando-se frases específicas que descrevam a TRANSFORMAÇÃO efetuada pelos módulos que contêm tais funções.

Quando o módulo "A" é chamado, o sistema entra em fase de execução. Por esta razão, a função do módulo "A" é equivalente à função de um programa. Como exemplo, admita-se que um problema seja: - "Escrever um Compilador de Fortran", a função do módulo seria: - "COMPILADOR DE PROGRAMA FORTRAN".

A Figura 4.7, mostra que a saída do módulo "B" é a mais abstrata entrada de dados; Um exemplo desse módulo seria: - "obtenha os seguintes comandos fontes na forma polonesa". O módulo "C", por sua vez, representa a TRANSFORMAÇÃO CENTRAL do programa, onde seu propósito reside em transformar "A MAIOR CADEIA DE ENTRADA" em "A MAIOR CADEIA DE SAÍDA"; um exemplo desse módulo seria: - "Display os resultados da Simulação".



INTERFACES	PARÂMETROS	
	ENTRADA	SAÍDA
1	Nenhuma entrada	Entrada de dados mais <u>abs</u> trata
2	Entrada de dados mais <u>abs</u> trata	Saída de dados mais <u>abs</u> trata
3	Saída de dados mais <u>abs</u> trata	Nenhuma Saída

FIGURA 4.7 - INICIANDO A ESTRUTURA DO PROBLEMA

4.1.5 - RECURSÃO

Na Figura 4.7, os módulos subordinados ao módulo "A" foram definidos como "B", "C" e "D". o resto do processo se prenderá à tarefa de se definir os módulos subordinados aos módulos "B", "C" e "D", e recursivamente o processo se repetirá para os subordinados desses, e, assim por diante.

A definição dos módulos subordinados será feita, partindo-se de um dos módulos, considerando-o sob a forma de um subproblema, e, reiniciando com a definição da "estrutura deste subproblema". Nesta análise pode ocorrer a identificação de mais de um fluxo de dados conceituais, surgindo - em decorrência - vários módulos tipo fonte e sorvedouro.

Quando da análise temporária de um módulo, deve-se olhá-lo como se fora um problema a ser resolvido, observando-se os seguintes critérios:

i - Não importa a ordem pela qual um módulo possa ser analisado.

ii - Não é necessário se analisar uma estrutura completa para, posteriormente, analisar-se outra.

iii - Antes de analisar o subordinado de um módulo, este deve ser completamente analisado. Assim, antes de um determinado módulo ter sido totalmente examinado, seus subordinados devem ser definidos.

iv - Quando se decompõe um módulo em vários subordinados, deve-se revisar toda a estrutura, a nível dos subordinados, o que possibilita a partilhação dos mesmos com os diferentes elementos da estrutura. Com tal medida, pode-se encontrar módulos idênticos (com funções idênticas, "in

terfaces" similares ou funções similares) que possam ser adaptados e utilizados, aumentando o seu "fan-in".

v - Quando se analisa um subproblema, a identificação da "fonte - transformação - sorvedouro" torna-se mais complexa, em face de que o maior fluxo de dados conceituais do subproblema pode ser absorvido, ou criado no módulo, que está sendo analisado. Na Figura 4.8, apresenta-se dois módulos de um subproblema em análise, onde se observa que, num módulo tipo fonte, sempre existe um fluxo conceitual no módulo quando retorna ao módulo que o chama. Em face disto, um módulo tipo fonte, pode - em parte - aparecer como sendo um módulo tipo sorvedouro com relação ao subproblema. O mesmo raciocínio aplica-se a um módulo tipo sorvedouro, em situação idêntica.

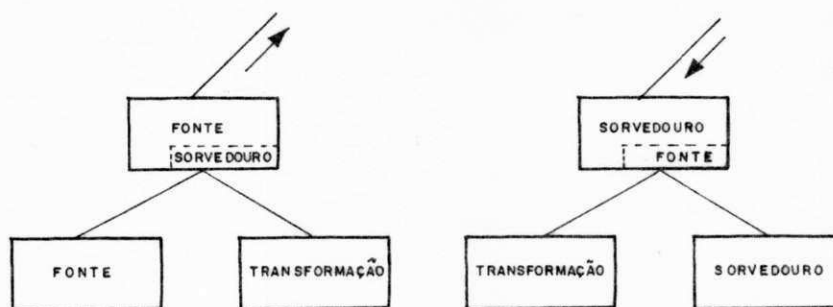


FIGURA 4.8 - DECOMPOSIÇÃO DE MÓDULOS TIPO FONTE E SORVEDOURO

vi - Os subordinados de um módulo tipo fonte são, geralmente, um ou mais módulos tipo fonte e um módulo tipo transformação. Pode ocorrer que um módulo tipo fonte te

nha um módulo tipo sorvedouro como subordinado (por exemplo: para escrever uma mensagem de erro).

vii - Os subordinados de um módulo tipo sorvedouro são geralmente, um ou mais módulos tipo sorvedouro e um módulo tipo transformação. Em casos raros, pode ocorrer que um módulo tipo sorvedouro tenha um módulo tipo fonte como subordinado.

viii - Os subordinados de um módulo tipo transformação, são, geralmente, módulos de mesmo tipo, podendo ocorrer - em alguns casos - subordinados tipo fonte e/ou sorvedouro.

4.1.6 - TÉRMINO DA DECOMPOSIÇÃO

Não há dúvida de que a metodologia Análise Composta é um processo iterativo, e, que obviamente, deve haver um término de tal processo. Para que isso aconteça, não existem critérios explícitos, contudo, algumas observações podem ser recomendadas:

i - Quando nenhum dos módulos puder ser decomposto em outros módulos subordinados, funcionalmente independentes, então, o processo de decomposição deve ser encerrado.

do.

ii) Se, se pode visualizar completamente na mente a lógica do módulo (o que geralmente indica que os módulos têm menos de 50 instruções) então, a decomposição do módulo não será mais necessária.

CAPÍTULO 5

SISTEMA CONTROLE ACADÊMICO**INTRODUÇÃO**

Na parte anterior deste trabalho foi descrita a metodologia de Análise Composta, algumas medidas que devem ser observadas, quando da aplicação da referida metodologia, bem como, algumas informações importantes com relação à modularidade de sistemas e programas.

Necessário se tornou, escolher uma aplicação em particular, a fim de elucidar de que forma, na prática, estas medidas e conceitos se aplicam, bem assim, a maneira pela qual a metodologia de Análise Composta deve ser empregada.

Naturalmente, há uma série de casos onde toda teoria apresentada poderia ser aplicada. Contudo, como este pretende ser um trabalho de caráter prático, ou seja, como o sistema apresentado deve ser um sistema real de implan

tação possível, a escolha recaiu no problema de Controle Acadêmico.

Os motivos que contribuíram para a escolha deste problema foram os abaixo mencionados.

i - Existência na UFPb, de um Sistema de Controle Acadêmico, já em funcionamento com as suas entradas e saídas bem definidas.

ii - O fato de o referido sistema funcionar dinamicamente, e mostrar-se propenso a se estruturar de maneira a se expandir mais rapidamente em forma modularizada, permitirá uma modificabilidade e extensibilidade mais eficiente e menos onerosa.

iii - A existência de um computador, cujas características permitem sua utilização para execução deste trabalho.

Assim sendo, todo o sistema descrito neste Capítulo, refere-se à modularidade do Sistema de Controle Acadêmico da Universidade Federal da Paraíba, que, com poucas alterações, poderá ser implantado em qualquer Universidade.

Dentro do ciclo de vida do sistema, o assunto abordado neste Capítulo, enquadra-se na fase de "design" lógico do sistema, de uma vez que, o referido sistema se

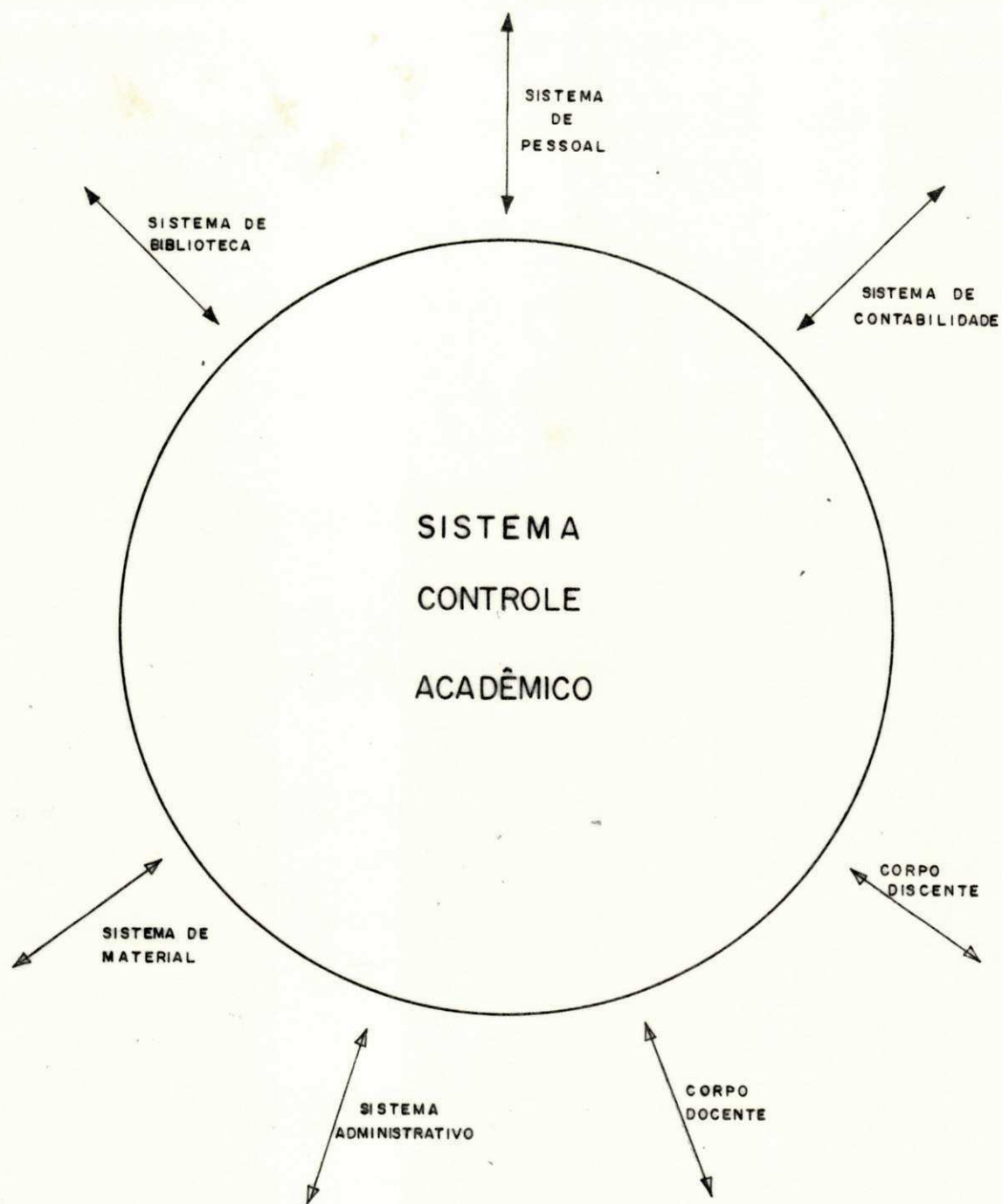
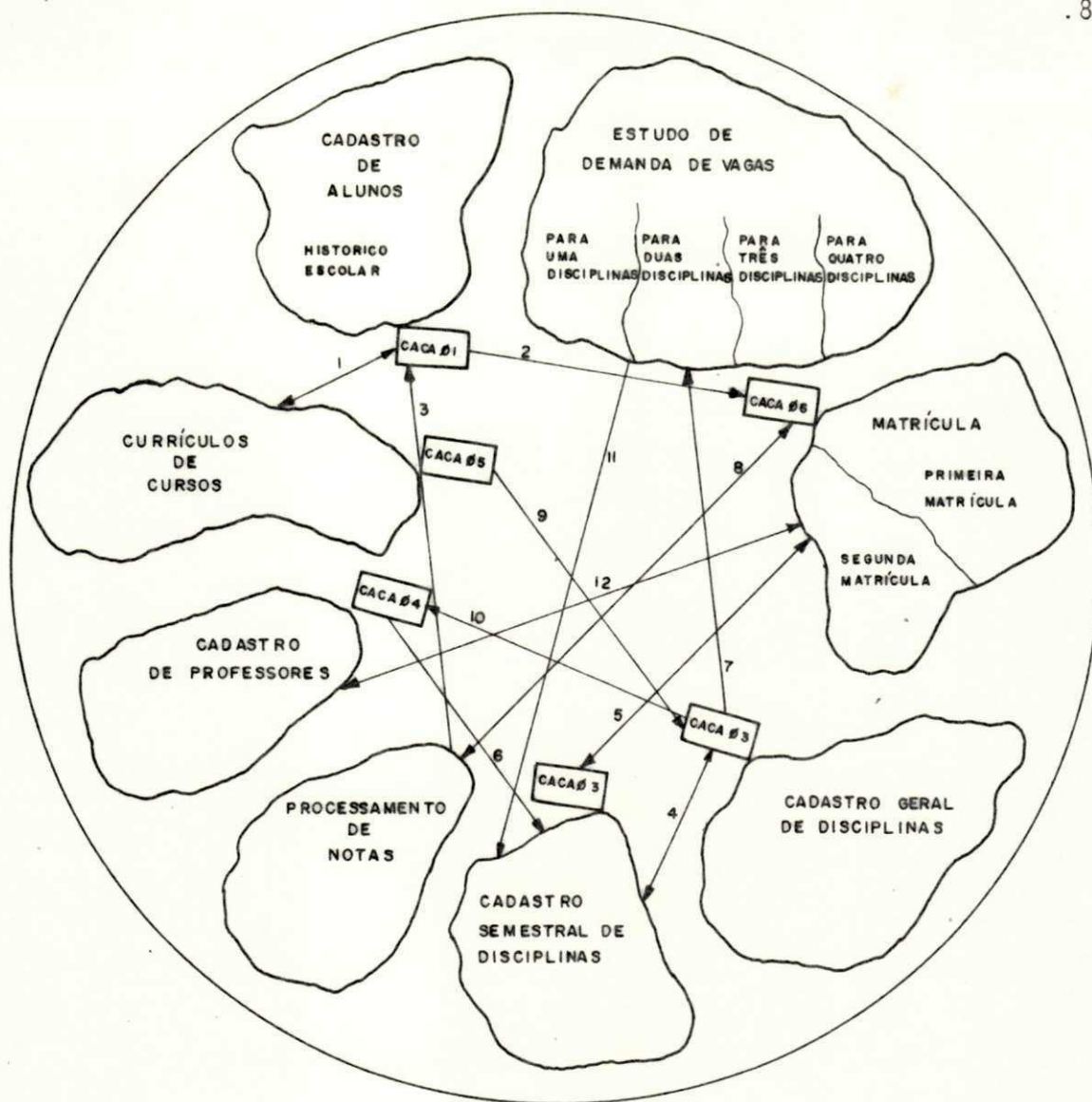


Fig. 5.1 - SISTEMA CONTROLE ACADÊMICO E O AMBIENTE (UNIVERSIDADE)



Fig. 5.2 - ARQUITETURA DO SISTEMA CONTROLE ACADÊMICO



INTER-FACES	DESCRIÇÃO
1	REGISTRO CONTENDO DISCIPLINAS CURSADAS POR ALUNO
2	NOME DO ALUNO
3	CÓDIGO ALUNO / CÓDIGO DISCIPLINA, CONCEITO (NOTA)
4	REGISTRO CONTENDO DADOS COMPLETOS SOBRE DETERMINADA DISCIPLINA E MARCA DA DISCIPLINA OFERECIDA
5	CÓDIGO DISCIPLINA, VAGA REMANESCENTE
6	CÓDIGO DO PROFESSOR
7	REGISTRO CONTENDO DADOS SOBRE DISCIPLINA
8	CÓDIGO DO ALUNO / CÓDIGO DE DISCIPLINA, CONCEITO (NOTA)
9	REGISTRO CONTENDO DADOS DE CADA DISCIPLINA CRIADA
10	APONTADORES PARA DISCIPLINA ENSINADA POR PROFESSORES
11	ANÁLISE DE RESULTADOS DE E.D.V, DISCIPLINAS OFERECIDAS
12	CÓDIGO DE DISCIPLINA

FIGURA 5.3 - PROJETO DOS COMPONENTES E SUBCOMPONENTES DO SISTEMA.

encontra em quase sua totalidade, já implantado, e, seus fluxos de dados de entrada e saída bem definidos.

O Sistema Controle Acadêmico destina-se a processar todas as informações, concernentes à vida Acadêmica do aluno, possibilitando o registro e manipulação de todas as informações que se relacionam com o aluno, direta ou indiretamente, constituindo-se, assim, dos subsistemas mostrados na Fig. 5.2, descritos e desenvolvidos utilizando-se a metodologia de Análise Composta.

5.1 - SUBSISTEMA CADASTRO DE ALUNOS

Este subsistema tem por finalidade o cadastramento dos dados pessoais de todos os alunos da Universidade, bem como de todas as disciplinas por eles cursadas. Tais informações serão constantes do Arquivo Cadastro de Alunos, CACAØ1, com estrutura de lista invertida a fim de facilitar a impressão de relatórios diversos (ver Apêndice "A").

Assim sendo, são suas finalidades:

i - Manter registrado em um arquivo com estrutura de lista invertida (ver Apêndice "A") todas as informações dos alunos.

ii- Atualizar todas as informações cadastrais;

iii - Emitir os mais variados relatórios estatísticos baseados em dados dos alunos;

iv - Fornecer subsídios ao Subsistema Matrricula, de uma vez que, somente os alunos devidamente cadastrados podem ser matriculados;

v - Manter arquivadas em disco, todas as informações dos alunos referentes às disciplinas por eles cursadas, como também, conceitos e números de créditos.

vi - Emitir documento denominado Histórico Escolar.

vii - Após a conclusão do curso, imprimir relatório contendo todas as informações contidas no cadastro e, posteriormente, retirá-las do arquivo.

FUNÇÃO DOS MÓDULOS

Módulo: 1.1

Função: Obtém Dados de Transação

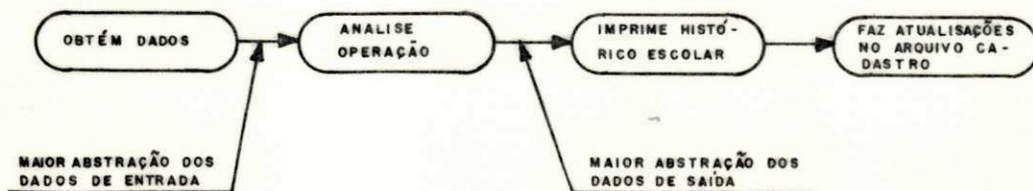
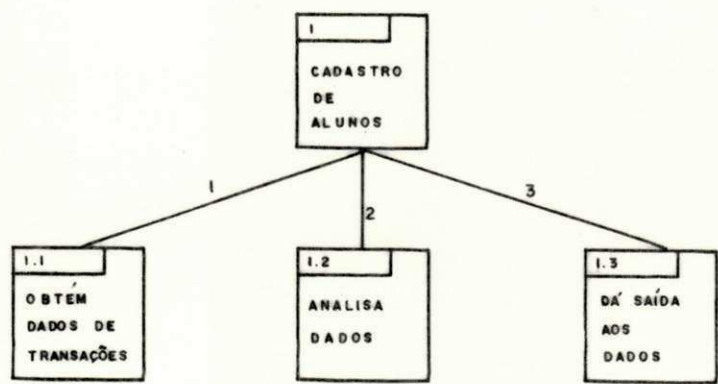


FIGURA 5.4 - ESTRUTURA DO PROBLEMA



INTERFACES	PARAMETROS	
	ENTRADA	SAÍDA
1		Registros do arquivo CAM001.
2	Registros do Arquivo CAM001	Tipo de operação
3	Tipo de operação	

FIGURA 5.5 - "INTERFACE" E ESTRUTURA DE TOPO

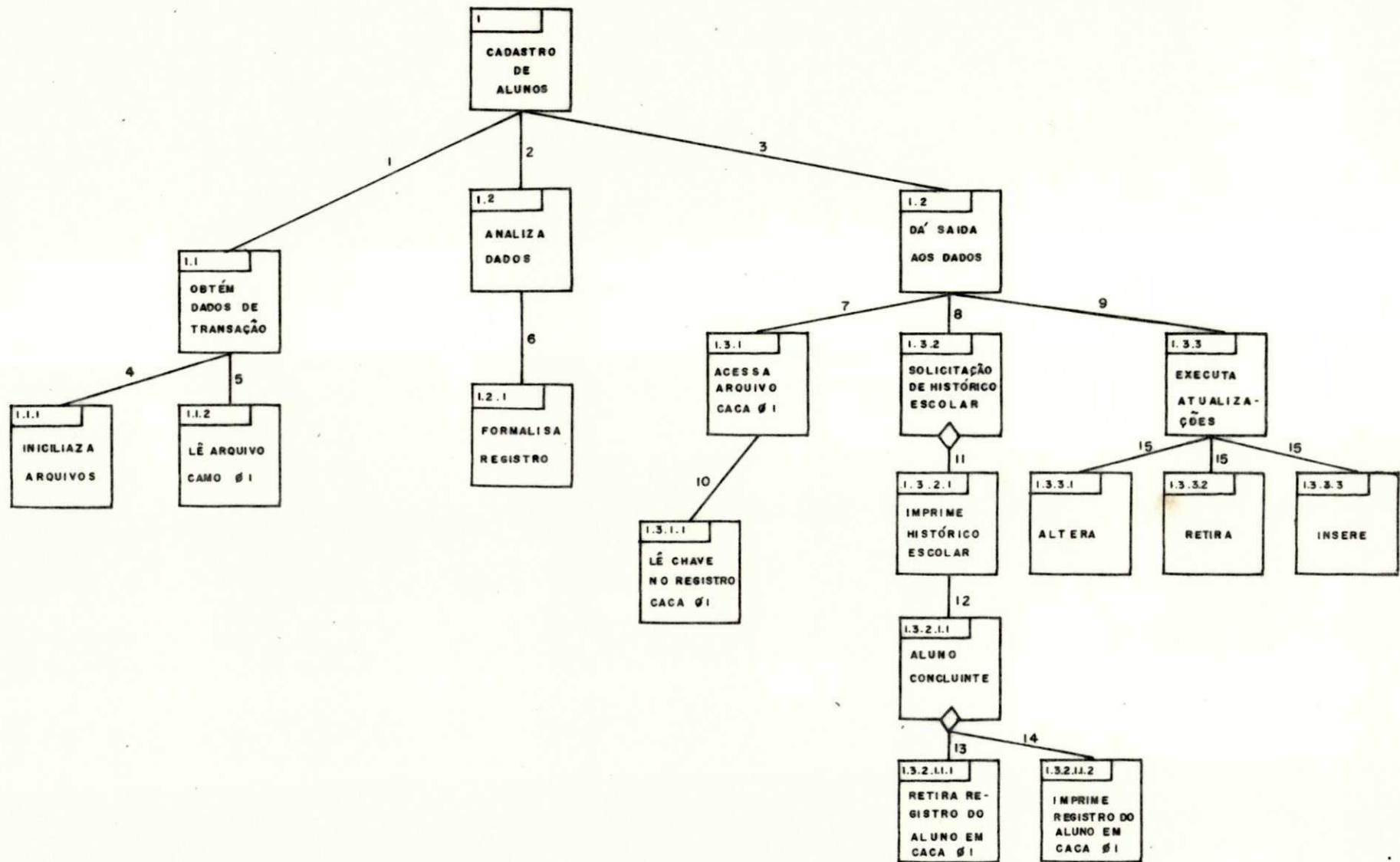


FIGURA 5.6(a) - DIAGRAMA ESTRUTURAL DO CADASTRO DE ALUNOS

INTER FACE.	P A R Â M E T R O S	
	E N T R A D A	S A I D A
1		Registros do arquivo CAM01
2	Registros do arquivo CAM01, código de operação	Código de Operação
3	Código de Operação	
4		Arquivos CAM01 e CACA01
5	Arquivo CAM01	Registros do arquivo CAM01
6	Código de operação, registro de CAM01	Registros formalizados
7	Registro de CACA01	Chave do Registro CACA01
8	Código de operação, registro de CACA01	
9	Registros formalizados, Chave do Registro CACA01, código operação	
10		Registro de CACA01
11	Registro de CACA01	
12	Registros do Arquivo CACA01	
13	Registros do Arquivo CACA01	
14	Registros do Arquivo CACA01	
15	Registros formalizados	

FIGURA 5.6(b) - "INTERFACES" DO CADASTRO DE ALUNOS

A partir do documento de entrada, são perfurados os cartões que constituem o arquivo CAM001, contendo informações cadastrais dos alunos.

Módulo: 1.2

Função: Analisa Dados

Os dados são analisados a partir do código de operação, o qual determina que tipo de operação será imposta aos dados.

Módulo: 1.3

Função: Dá Saída aos Dados

Neste módulo são executadas as operações de saída, bem como, uma atualização em arquivo ou mesmo uma impressão de documentos.

Módulo: 1.1.1

Função: Inicializa Arquivos

Neste módulo, os arquivos envolvidos no processamento são abertos ou inicializados, só sendo fechados no fim do processamento.

Módulo: 1.1.2

Função: Lê Arquivo CAM001

Após a abertura dos arquivos, o arquivo CAM001 deve ser lido, seus registros e campos ficam disponíveis para posterior operação sobre eles.

Módulo: 1.2.1

Função: Formaliza Registros

Os campos a serem alterados são devidamente formalizados num registro que será gravado no arquivo a ser modificado.

Módulo: 1.3.1

Função: Acessa Arquivo CACA01

Os registros do arquivo CACA01 devem ser acessados a fim de que as operações no referido arquivo possam ser levadas a efeito.

Módulo: 1.3.2

Função: Solicitação de Histórico Escolar?

Este é um módulo tipicamente de decisão que depende do código de operação. Se o código indicar a impressão de Histórico Escolar, tal documento será impresso.

Módulo: 1.3.1.1

Função: Lê Chave no Registro de CACA01

A partir do registro do arquivo CACAØ1 o campo da chave é lido para se verificar o endereço do aluno no arquivo CACAØ1.

Módulo: 1.3.2.1

Função: Imprime Histórico Escolar

O Histórico Escolar do aluno é impresso. Um formulário próprio deve ser pedido pela "console" a fim de se imprimir o Histórico Escolar.

Módulo: 1.3.3.1

Função: Altera

Um registro inteiro ou um campo do arquivo CACAØ1 deve ser alterado.

Módulo: 1.3.3.2

Função: Retira

Neste módulo um registro ou parte dele deve ser retirado.

Módulo: 1.3.3.3

Função: Insere

Um registro ou parte dele deve ser inse

rido.

Módulo: 1.3.2.1.1

Função: Aluno Concluinte

Módulo de decisão, onde é verificado se o aluno é concluinte ou não.

Módulo: 1.3.2.1.1.1

Função: Retira Registro do Aluno em CACAØ1

O aluno, pelo fato de haver concluído seu curso, terá todas as suas informações retiradas do arquivo CACAØ1.

Módulo: 1.3.2.1.1.2

Função: Imprime Registro do Aluno em CACAØ1

Tendo o aluno concluído o curso terá to dos os seus dados impressos e arquivados manualmente.

5.2 - SUBSISTEMA CURRÍCULOS DE CURSOS

Os Currículos de Cursos se constituem de:

- a) Currículo Mínimo
- b) Currículo Complementar Obrigatório
- c) Currículo Complementar Optativo.

Os objetivos do Subsistema Currículos de Cursos são assim apresentados:

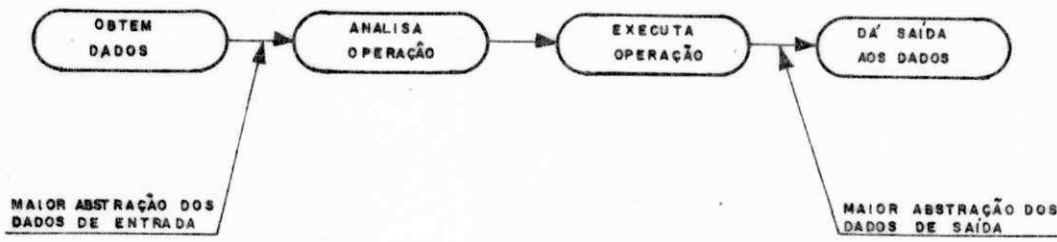
i - Manter cadastrados todos os currículos, tendo como chave, o código de curso/período.

ii - Cada curso poderá se constituir de mais de um currículo, dependendo das estruturas curriculares, que são função dos períodos de vigência e portarias de adaptação dos currículos.

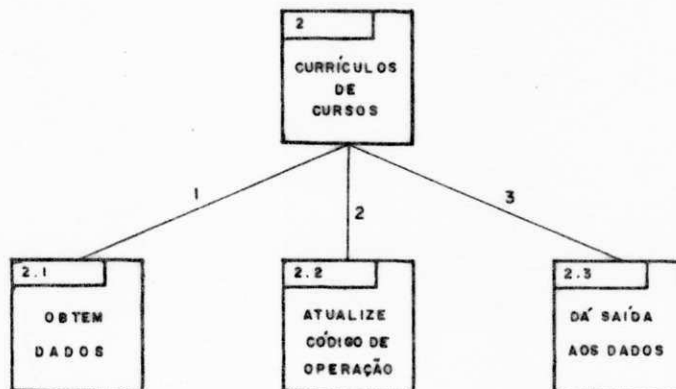
iii - Cada aluno se encaixará dentro de um currículo, dependendo do curso e do período do ingresso na Universidade.

iv - Em conjunto com o subsistema Cadastro de Alunos, fornecer relatórios posicionando a situação de cada aluno, indicando quais as disciplinas que foram cursadas, e quais as que devem ser oferecidas, o que poderá - no futuro - substituir o subsistema Estudo de Demanda de Vagas.

v - Ser capaz de suportar operações de



ENT Fig. 5.7 ESTRUTURA DO PROBLEMA



INTER-FACES	PARAMETROS	
	ENTRADA	SAÍDA
1	_____	REGISTRO DO ARQUIVO CAMO Ø 2
2	REGISTRO DO ARQUIVO CAMO Ø 2	CÓDIGO DE OPERAÇÃO
3	CÓDIGO DE OPERAÇÃO	_____

Fig. 5.8 - ESTRUTURA E "INTERFACES" DO TÔPO

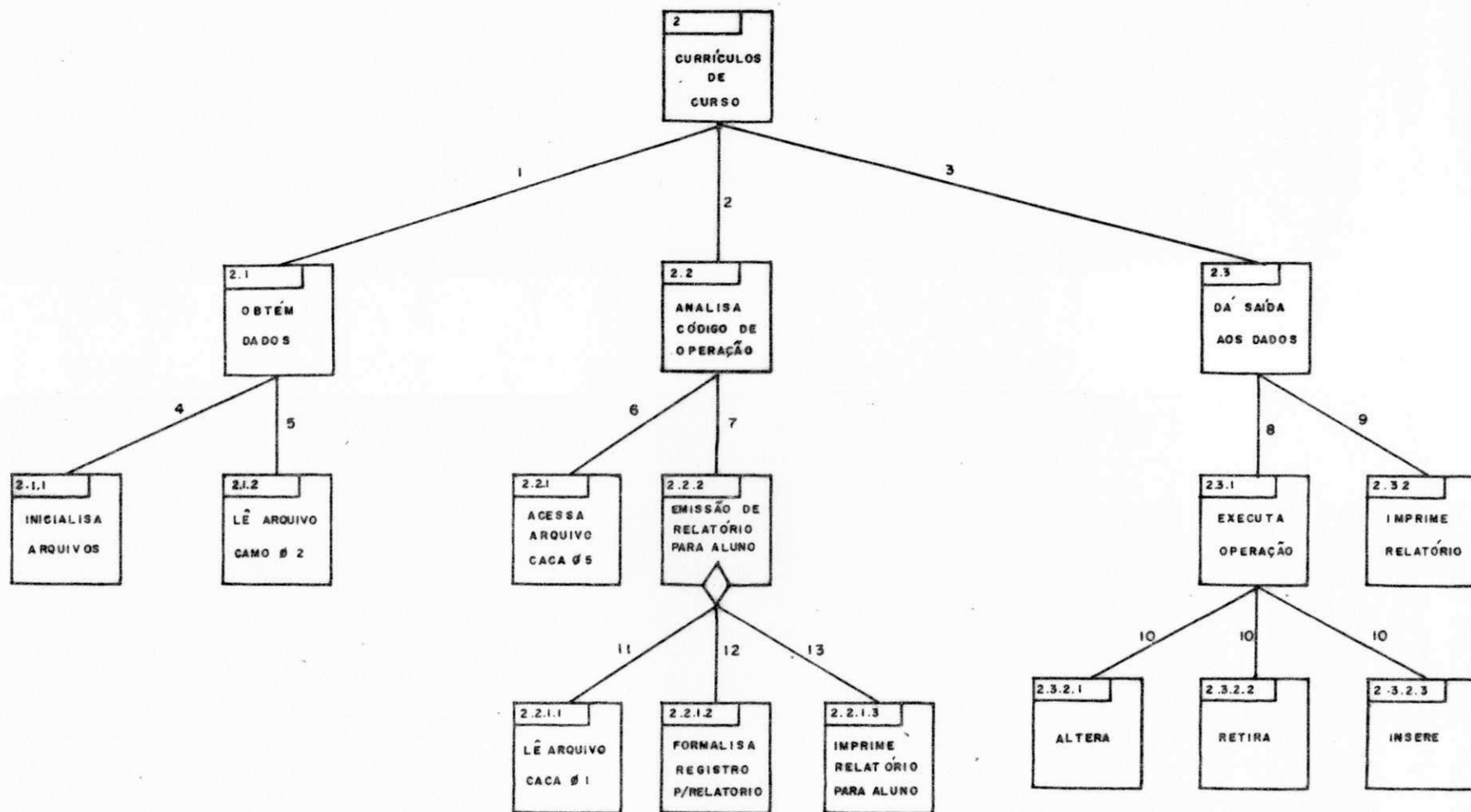


FIGURA 5.9(a) - DIAGRAMA ESTRUTURAL DO SUBSISTEMA CURRÍCULOS DE CURSOS

INTER FACE.	P A R Â M E T R O S	
	E N T R A D A	S A I D A
1		Registro do arquivo CAMP24
2	Registro do arquivo CAM003	Código de Operação, registro do arquivo CACA05
3	Código de operação e registro do arquivo CACA05	
4		Arquivo CAM002
5	Arquivo CAM002	Registros do arquivo CAM024
6	Código de operação, registro do arquivo CAM002	Registro do arquivo CACA05
7	Registro do arquivo CACA05, código de operação	Registros do arquivo CACA05
8	Código de operação, registros do arquivo CAMA02, registro do arquivo	Registro formalizado, para relatório
9	Registros formalizados, para relatório.	
10	Registros formalizado	
11	Arquivo CACA01	Registro do arquivo CACA01
12	Registros dos arquivos CACA05 e CACA01	Registro formalizado para relatório A
13	Registro formalizado para relatório.	

FIGURA 5.9(b) - "INTERFACES" DO SUBSISTEMA CURRÍCULOS DE CURSOS.

atualização em um registro, ou qualquer campo do arquivo ca
dastro.

No desenvolvimento deste subsistema a fa
se de consistência de dados foi excluída.

FUNÇÃO DOS MÓDULOS

Módulo: 2.1

Função: Obtém Dados

Os dados são colocados à disposição do pro
cessamento neste módulo.

Módulo: 2.1.1

Função: Inicializa Arquivos

Neste módulo os arquivos CAM002 e CACA05
são abertos, colocados à disposição do processamento, e fe
chados após o processamento.

Módulo: 2.1.2

Função: Lê. Arquivo CAM002

O arquivo CAM002 é lido e seus registros

são colocados à disposição do processamento.

Módulo: 2.2

Função: Analisa Código de Operação

De posse do código de operação os dados são analisados, a fim de que as operações sejam executadas.

Módulo: 2.2.1

Função: Acessa Arquivo CACA05

O arquivo Cadastro de Currículos é acessado e seus registros são colocados à disposição do processamento para posterior manipulação.

Módulo: 2.2.1.1

Função: Lê Arquivo CACA01

O arquivo Cadastro de Alunos é lido e seus registros colocados à disposição do processamento das operações de atualizações ou da impressão de relatório para alunos.

Módulo: 2.2.1.2

Função: Formaliza Registro para Relatório

Neste módulo formaliza-se um registro cons

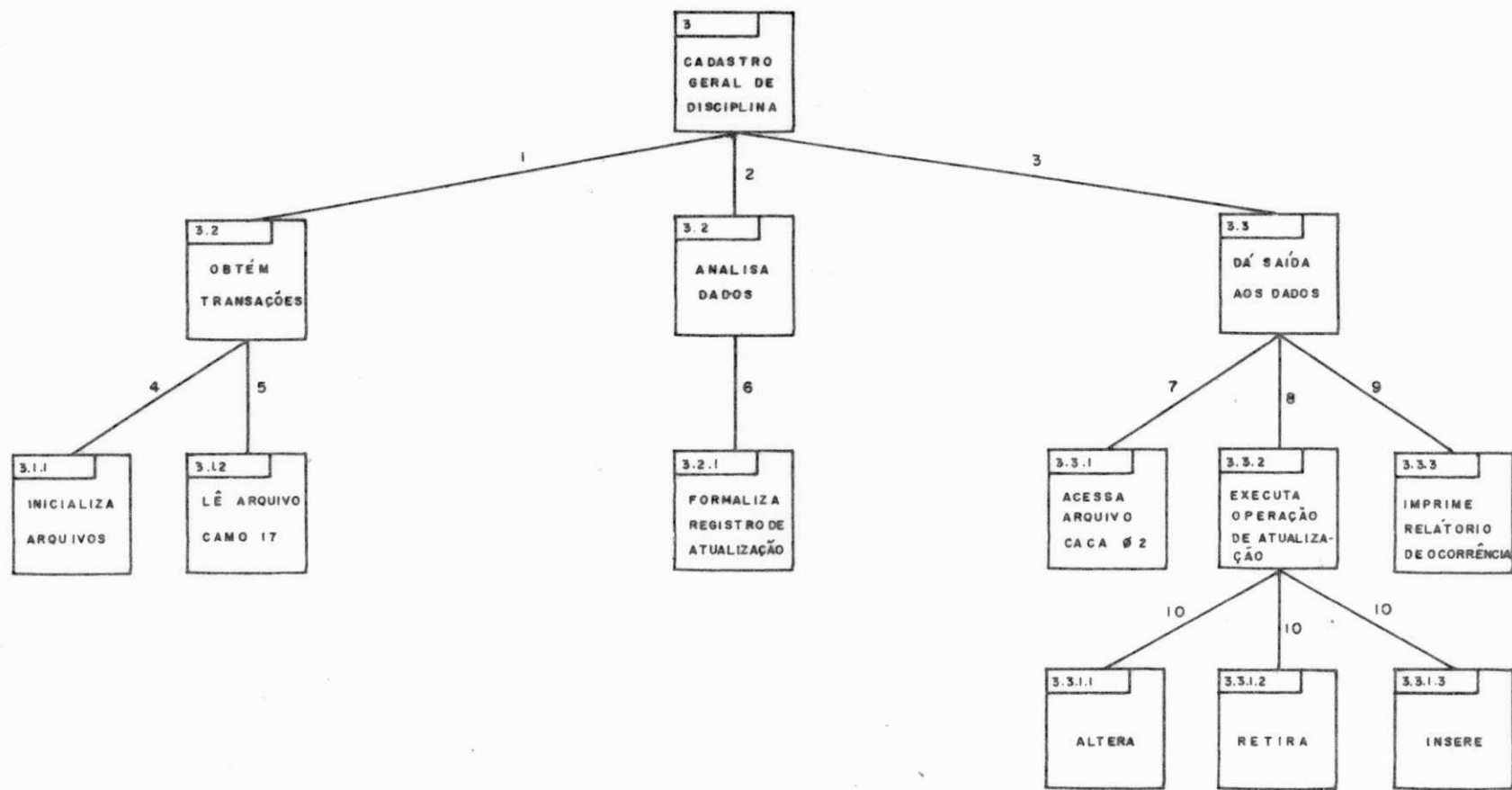


FIGURA 5.12(a) - DIAGRAMA ESTRUTURAL DO CADASTRO GERAL DE DISCIPLINA

INTER FACE.	P A R Â M E T R O S	
	E N T R A D A	S A I D A
1		Registro do arquivo CAM017
2	Registro do arquivo CAM017	Tipo de operação
3	Tipo de operação	
4		Arquivo CAM017
5	Arquivo CAM017	Registro do arquivo CAM020
6	Código de operação, registro de CAM017	Registro formalizado
7	Código de disciplina, CACA02	Registro do arquivo CACA02
8	Código de operação registro do arquivo CACA02, registro formalizado	
9	Registro formalizado	
10	Registro formalizado	

FIGURA 5.12(b) - "INTERFACES" DO CADASTRO GERAL DE DISCIPLINA.

Módulo: 3.1

Função: Obtém Transações

A partir do documento de entrada que contém os dados de cadastramento (código de operação, código de disciplina, nome de disciplina, pré-requisitos) ou informações de atualização (qualquer campo) são perfurados cartões que constituem o arquivo CAM017, contendo informações cadastrais das disciplinas.

Módulo: 3.2

Função: Analisa Dados

É feita uma análise do código de operação a fim de se verificar que tipo de operação será executada pelo referido módulo.

Módulo: 3.3

Função: Dá Saída aos Dados

As operações de saída são realizadas neste módulo, bem como: as atualizações.

Módulo: 3.1.1

Função: Inicializa Arquivos

Neste módulo todos os arquivos CAM017 e

CACAØ2, envolvidos no processamento, são abertos e, inicializados ficando à disposição de qualquer outro módulo.

Módulo: 3.1.2

Função: Lê Arquivo CAM017

Os dados do arquivo movimento CAM017 são lidos, ficando os registros à disposição do processamento.

Módulo: 3.2.1

Função: Formaliza Registro de Atualização

De posse dos registros dos dados de entrada, e, definido o código de operação, é feita a formalização do registro que irá atualizar o arquivo CACAØ2.

Módulo: 3.3.1

Função: Acessa Arquivo CACAØ2

O arquivo cadastro geral de disciplinas é acessado, colocando os registros a serem manipulados à disposição do processamento.

Módulo: 3.3.2

Função: Executa Operações de Atualização

Neste módulo as operações de atualização

são definidas e executadas.

Módulo: 3.3.3

Função: Imprime Relatório de Ocorrências

A medida em que as operações são executadas, relatórios, são impressos, descrevendo os tipos de operações e os campos onde elas se realizaram.

Módulo: 3.3.1.1

Função: Altera

Neste módulo, dá-se a operação de alteração de um registro ou campo do arquivo CACAØ2.

Módulo: 3.3.1.2

Função: Retira

Neste módulo, executa-se a operação de retirada de um registro ou Campo do arquivo CACAØ2.

Módulo: 3.3.1.3

Função: Insere

A operação de inserção de uma nova disciplina se dá neste módulo.

5.4 - SUBSISTEMA CADASTRO SEMESTRAL DE DISCIPLINAS

Em cada início de período letivo, são escolhidas várias disciplinas que serão oferecidas aos alunos com bases no resultado do estudo prévio, denominado Estudo de Demanda de Vagas, onde os alunos em quase sua totalidade, participam, indicando as disciplinas que pretendem cursar no período em enfoque. De posse dos resultados de tal estudo, a Coordenação de cada curso indica quais disciplinas pretende oferecer e seus respectivos horários. Constituindo-se, assim, em informações importantes para criação e manutenção do Cadastro Semestral de Disciplinas.

São assim, as finalidades deste subsistema:

i) Manter arquivadas todas as disciplinas que serão oferecidas num período semestral ou especial.

ii) Manter arquivados os códigos dos professores que ministrarão as disciplinas previamente escolhidas pelas Coordenações de Cursos.

iii) Ser capaz de suportar operações de atualização, permitindo: alterações, inserções e retiradas de dados do referido cadastro.

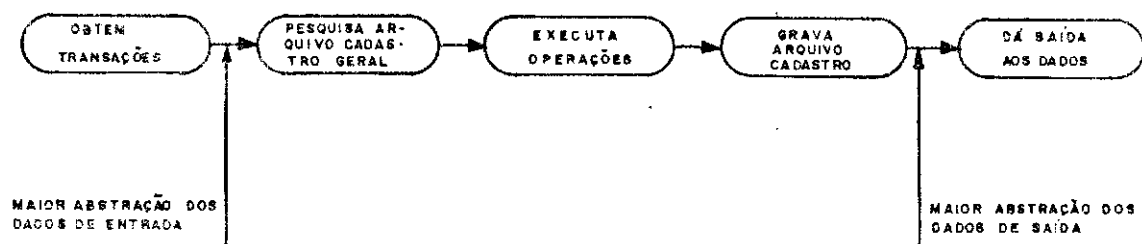
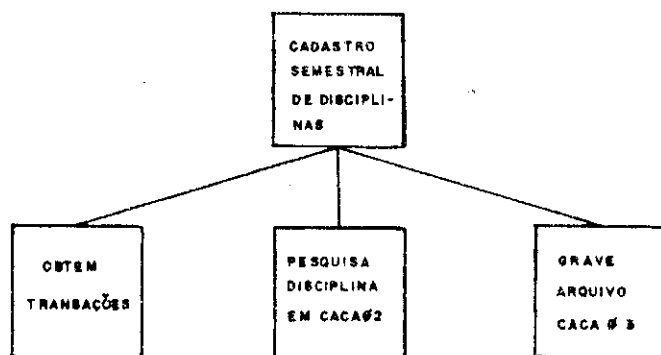
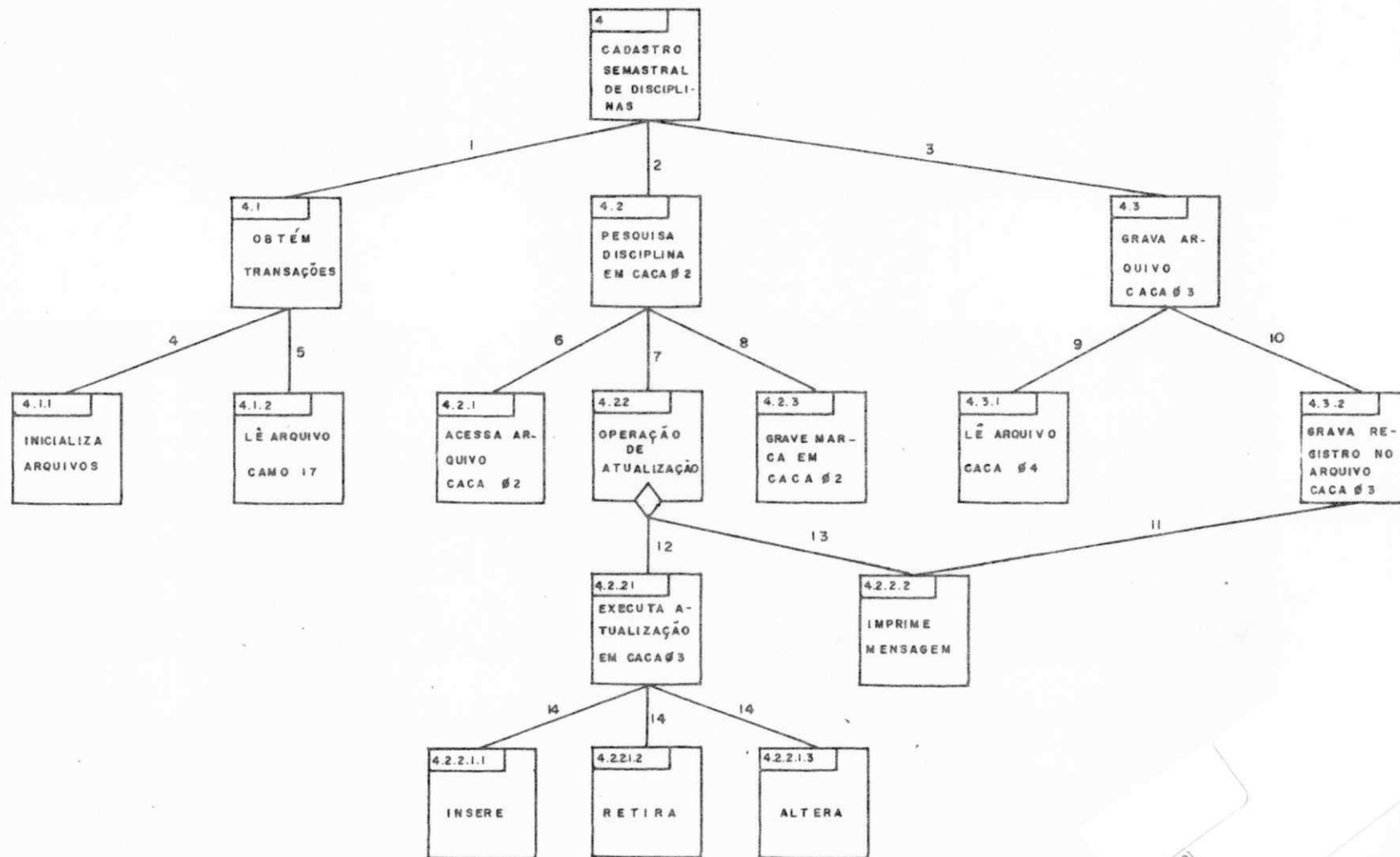


FIGURA 5.13 - ESTRUTURA DO PROBLEMA



INTERFACE	PARAMETROS	
	ENTRADA	SAÍDA
1		Arquivo CACA#4 Arquivo CAM017 Código de operação
2	Arquivo Camo21 Código de operação	Registros de CAM021
3	Registro de CAM021	

FIGURA 5.14 - Estrutura e "Interfaces"



5.15(a) - DIAGRAMA ESTRUTURAL DO CADASTRO SEMESTRAL DE DISCIPLINAS

INTERFACE.	P A R Â M E T R O S	
	E N T R A D A	S A I D A
1		Arquivo CAM017 código de operação
2	Arquivo CAM017, código de operação, Código de disciplina	Registro do arquivo CAM017, código de operação
3	Registro do arquivo CAM017	
4		Código de disciplina, arquivo CAM017
5	Arquivo CAMO 17	Registro do arquivo CAM017, código de operação
6	Código de disciplina, arquivo CACAø2	Registro do arquivo CACAø2
7	Registro do arquivo CAM017, código de operação	
8	Registro do arquivo CACAø2, código de disciplina	
9	Código do professor	Nome do professor
10	Registro do arquivo CAM017, Registro do arquivo CACAø2, Registro do arquivo CACAø3.	
11	Registro do arquivo CACAø3	
12	Registro do arquivo CAM017, Registro do arquivo CACAø2	
13	Registro do arquivo CACAø2	
14	Código de operação Registro do arquivo CACAø2	

FIGURA 5.15(b) - "INTERFACES" DO CADASTRO GERAL SEMESTRAL DE DISCIPLINAS.

iv) Emitir relatórios referentes às atualizações.

v) Colocar uma marca no registro do arquivo Cadastro Geral de Disciplinas CACA02, sempre que uma disciplina for oferecida.

A consistencia dos dados do arquivo movimento, CAM021 não é apresentada neste subsistema.

FUNÇÃO DOS MÓDULOS

Módulo: 4.1

Função: Obtém Dados

Neste módulo são obtidos os dados do arquivo CAM017 para posterior processamento, como também se verifica a operação de inicialização e abertura de arquivos.

Módulo: 4.1.1

Função: Inicializa Arquivos

São abertos ou inicializados todos os arquivos envolvidos no processamento, são os fechando após todas as operações executadas.

Módulo: 4.1.2

Função: Lê Arquivo CAM017

São lidos os registros do arquivo movimento, contendo dados que serão necessários para atualização do arquivo CACA02 ou a criação do arquivo CACA03.

Módulo: 4.2

Função: Pesquisa Disciplina em CACA02

Neste módulo, as disciplinas são pesquisadas pelo código, no arquivo CACA02.

Módulo: 4.2.1

Função: Pesquisa Disciplina em CACA02

De posse do registro do arquivo CAM002, e, do código da disciplina, o arquivo CACA02 que tem como chave o código da disciplina, é acessado e seus dados tornam-se disponíveis para a operação de atualização.

Módulo: 4.2.2

Função: Operação de Atualização

Módulo de decisão onde é verificado se o código de operação indica uma atualização ou outro tipo de operação.

Módulo: 4.2.3

Função: Grava Marca em CACAØ2

Neste módulo, é gravada uma marca no arquivo Cadastro Geral de Disciplina, CACAØ2, a qual indica que a disciplina constante daquele registro será oferecida naquele semestre ou período especial.

Módulo: 4.2.2.1

Função: Executa Atualização em CACAØ3

Uma operação de atualização no arquivo CACAØ2 se verifica neste módulo.

Módulo: 4.2.2.1.1

Função: Insere

Uma operação de inserção é verificada no arquivo CACAØ2.

Módulo: 4.2.2.1.2

Função: Retira

Neste módulo, dá-se uma operação de retirada de registros do arquivo CACAØ2.

Módulo: 4.2.2.1.3

Função: Altera

Uma operação de alteração é verificada num dos registros do arquivo CACAØ2.

Módulo: 4.2.2.2

Função: Imprime Mensagem

Após a operação no arquivo CACAØ2, uma mensagem de ocorrência deve ser impressa, constando do tipo e conteúdo da operação.

Módulo: 4.3

Função: Grava Arquivo CACAØ3

O Arquivo Cadastro Semestral de Disciplina é gravado a partir dos arquivos CAM021 e CAM0Ø2.

Módulo: 4.3.1

Função Lê Arquivo CACAØ4

Este módulo faz a leitura dos registros do arquivo CACAØ4, que tem como chave o código do professor, a fim de se obter o nome do professor que ministrará determinada disciplina.

Módulo: 4.3.2

Função: Grava Registro no Arquivo CACAØ3

Estando o arquivo CACAØ3 acessado, procede-se com a gravação dos registros que contêm as novas disciplinas.

5.5 - SUBSISTEMA CADASTRO DE PROFESSORES

É - na verdade - o Subsistema Cadastro de Professores, constituído de informações pessoais dos professores, mantidas arquivadas, capazes de auxiliar a Administração nas tomadas de decisões quando da designação de um professor para ministrar determinada disciplina, ou, mesmo, quando se fizer necessária a indicação de um professor para assumir função administrativa.

Sendo assim, são as suas finalidades:

i) Manter arquivados os dados pessoais dos professores.

ii) Auxiliar na tomada de decisões por meio de relatórios indicativos e estatísticos, tal como: relatório contendo as disciplinas que o professor já ensinou.

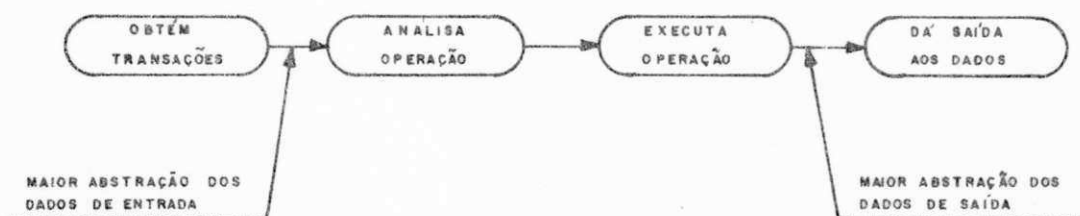
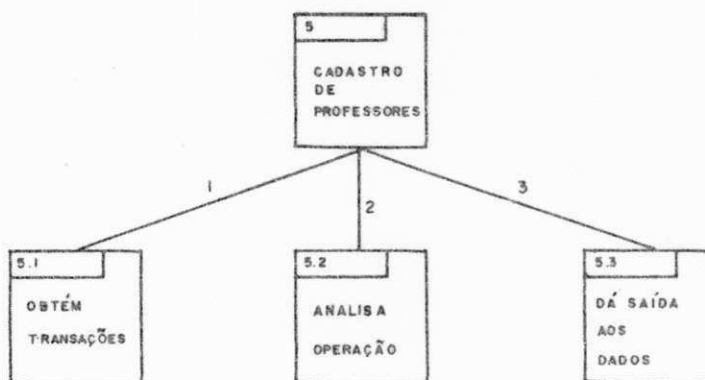


FIGURA 5.16 - ESTRUTURA DO PROBLEMA



INTERFACES	PARÂMETROS	
	ENTRADA	SAÍDA
1		Arquivo CAM022, código de operação.
2	Arquivo CAM022, código de operação.	Registros de CAM019
3	Registro de CAM019	

FIGURA 5.17 - ESTRUTURA E "INTERFACES" DE TOPO

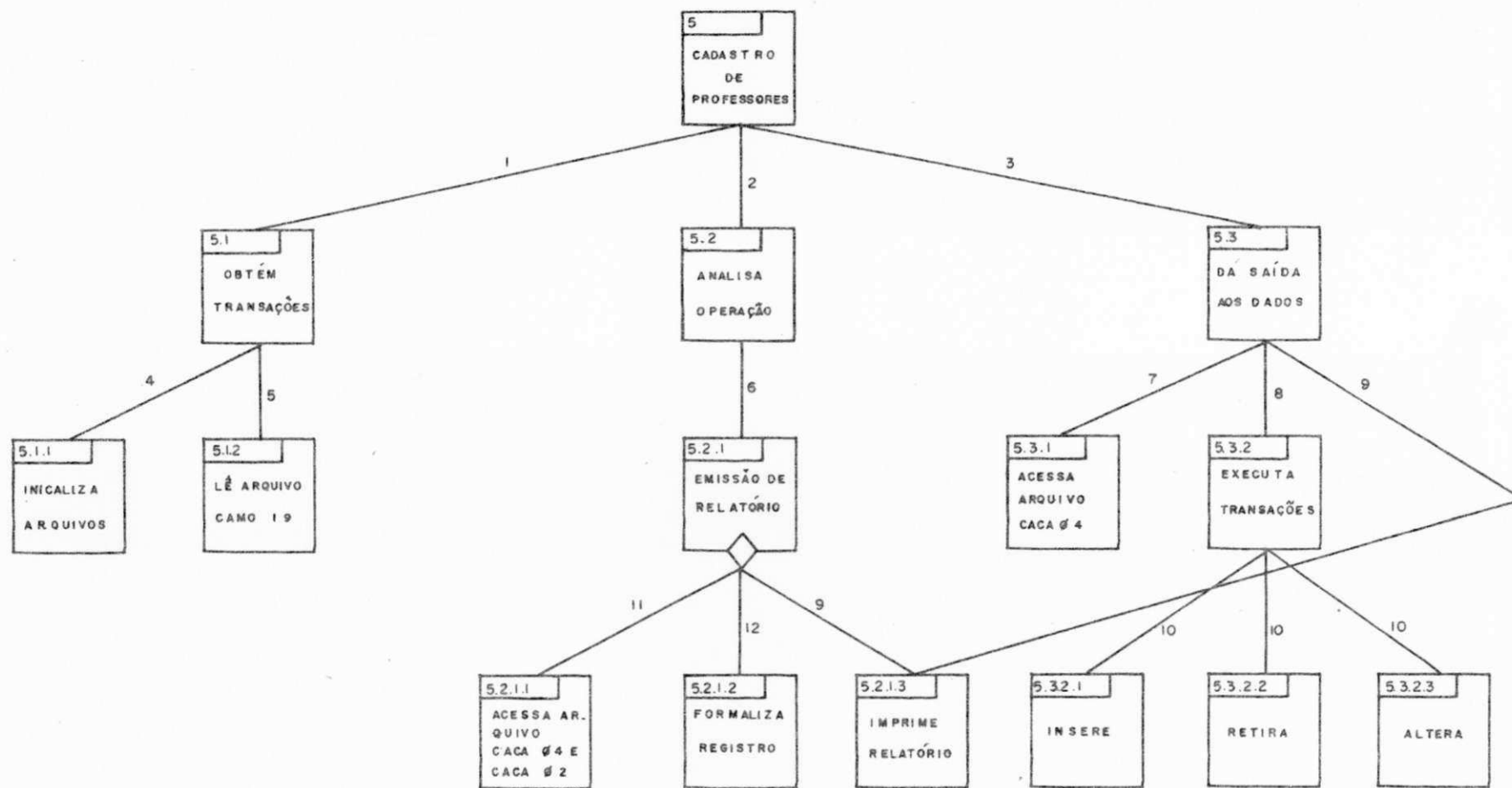


FIGURA 5.18(a) - DIAGRAMA ESTRUTURAL DO SUBSISTEMA CADASTRO DE PROFESSORES

INTER FACE.	P A R Â M E T R O S	
	E N T R A D A	S A I D A
1		Arquivo CAM019, código de operação
2	Arquivo CAM019, Código de operação	Registro do arquivo (CAM019
3	Registros do arquivo CAM019	
4		Arquivo CAM019
5	Registro do arquivo CAM019	Código de operação
6	Código de Operação, registro de arquivo CAM022	Registro formalizado
7	Código do professor, CACA04	Registro do arquivo CACA04
8	Código de operação, registro de CACA04, registro formalizado	Registro formalizado contendo operação de atualização
9	Registro formalizado contendo de atualização	
10	Registro formalizado	
11		Registro do arquivo CACA04
12	Registro do arquivo CACA04	Registro formalizado p/relatório

FIGURA 5.18(b) - "INTERFACE" DO SUBSISTEMA CADASTRO DE PROFESSORES.

de Departamento na escolha de professores para determinadas disciplinas.

iv) Ser capaz de suportar operações de a tualização em seu arquivo principal.

FUNÇÃO DOS MÓDULOS

Módulo: 5.1

Função: Obtém Transações

Os dados são obtidos, a partir dos documentos de entrada, que, após serem perfurados, são denominados arquivo CAM019. O referido arquivo contém, desta forma dados de cadastramento dos professores.

Módulo: 5.1.1

Função: Inicializa Arquivos

Os arquivos envolvidos no processamento são inicializados, ou abertos, sô os fechando após o término do processamento.

Módulo: 5.1.2

Função: Lê Arquivo CAM019

O arquivo CAM019 é lido e os dados tornam-se disponíveis para o processamento.

Módulo: 5.2

Função: Analisa Operação

Neste módulo, o campo código de operação é verificado a fim de se determinar que tipo de operação será executada pelos módulos de transformação.

Módulo: 5.2.1

Função: Emissão de Relatório?

De posse dos registros de dados de entrada, e, definido o código de operação, realiza-se a formalização do registro que será atualizado.

Módulo: 5.3

Função: Dá Saída aos Dados

As atualizações e operações de saída são executadas neste módulo.

Módulo: 5.3.1

Função: Acessa Arquivo CACA04

O arquivo com estrutura sequencial indexa

da Cadastro de Professores é acessado utilizando-se para tal o campo código de professor como chave. Após o que, os registros deste arquivo tornam-se disponíveis para operação.

Módulo: 5.3.2

Função: Executa Atualizações

Este módulo comanda os demais que executam as operações de atualização, fornecendo-lhe um registro formalizado para execução das referidas operações.

Módulo: 5.3.2.1

Função: Insere

A operação de inserção de registros no arquivo CACAØ4 se verifica neste módulo: utilizando-se o registro formalizado para execução da atualização.

Módulo: 5.3.2.2

Função: Retira

A operação de retirada de registros do arquivo CACAØ4, se verifica neste módulo, utilizando-se o registro formalizado para execução da operação de atualização.

Módulo: 5.3.2.3

Função: Altera

A operação de alteração de qualquer campo do registro do arquivo CACAØ4, é verificada neste módulo, utilizando-se o registro formalizado atualização.

Módulo: 5.2.1.1

Função: Acessa Arquivo CACAØ4 e CACAØ2

O Arquivo CACAØ4, com estrutura sequencial indexada, é acessado utilizando-se a chave de disciplina ensinada ou código do professor. Como também, o arquivo Cadastro Geral de Disciplinas, CACAØ2, e seus registros são colocados à disposição do processamento.

Módulo: 5.2.1.2

Função: Formaliza Registros

Utilizando-se os apontadores das disciplinas já ensinadas, fornecidas pelo arquivo CACAØ4, faz-se uma pesquisa no arquivo CACAØ2, afim de se obter os nomes das referidas disciplinas para posterior impressão de relatório.

Módulo: 5.2.1.3

Função: Imprime Relatórios

De posse dos registros formalizados, este módulo deve ser capaz de imprimir os relatórios a saber:

- a) relatório de operações de atualização

- b) relatório constando das disciplinas já ensinadas por determinado professor.

5.6 - SUBSISTEMA ESTUDO DE DEMANDA DE VAGAS

Com a implantação do sistema de crédito de disciplina, ou sistema parcelado, os cursos deixaram de a apresentar programações rígidas, onde a "priori", já se tinha uma relação das disciplinas que seriam oferecidas ao longo dos semestres, o que tornava a tarefa de oferecimento de disciplinas - em cada período - um problema de somenos importância.

Acontece, porém, que com a implantação do novo Sistema de créditos, a programação de disciplinas oferecidas em cada período ou semestre, constitui-se em tarefa que depende - antes de tudo - em se conhecer a situação de cada aluno, investigando-se quais as disciplinas que os mesmos estão aptos a cursar, em face do sistema de pré-requisitos. O Subsistema Estudo de Demanda de Vagas, propõe-se a fazer uma pesquisa junto aos alunos a fim de determinar: o que oferecer e para quem oferecer.

Desta forma, são suas finalidades:

- i) Manter arquivados os dados provenientes

da pesquisa feita junto aos alunos.

ii) Processar dados do levantamento e emitir relatórios como:

a) Relatório emitindo resultados de solicitações de uma disciplina, constando o referido relatório de:

Código e nome da disciplina

* { Nome do curso
Total de solicitações para: manhã, tarde, manhã extrapolada e tarde extrapolada.

b) Relatório emitindo resultados de solicitações de duas disciplinas concomitantes, constando o referido relatório de:

Código e nome das duas disciplinas

* { Nome do curso
Total de solicitações para; manhã, tarde, manhã extrapolada e tarde extrapolada.

c) Relatório emitindo resultados de solicitações concomitantes, de tres disciplinas, constando o referido relatório de:

Código e nome das tres disciplinas

* Repetindo-se 15 vezes

* { Nome do curso
Total de solicitações para; manhã, tarde,
manhã extrapolada e tarde extrapolada.

d) Relatório emitindo resultados de solici-
tações concomitantes de quatro discipli-
nas, constando de:

Código e nome das quatro disciplinas.

* { Nome do curso
Total de solicitações para; manhã, tarde,
manhã extrapolada e tarde extrapolada.

iii) Cada processamento para emissão de um
tipo de relatório apresenta-se em um bloco independente, que
se denomina fase. Portanto, o Subsistema Estudo de Demanda de
Vagas se apresenta dividido em quatro fases distintas a sa-
ber:

Fase 1 - Emissão de totais para uma disci-
plina

Fase 2 - Emissão de totais para duas disci-
plinas

Fase 3 - Emissão de totais para tres disci-

* Repetindo-se 15 vezes

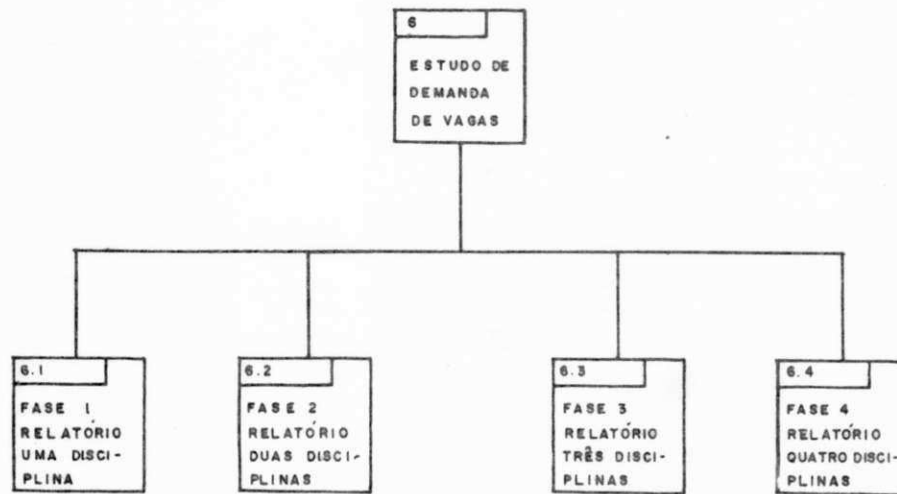


FIGURA 5.19 - SUBDIVISÕES DO SUBSISTEMA ESTUDO DE DEMANDA DE VAGAS.

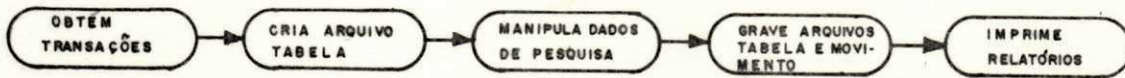
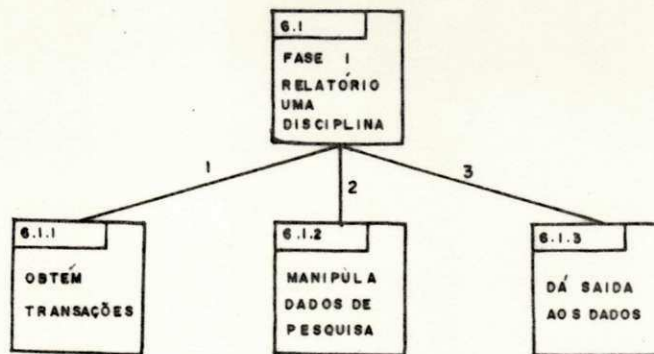


FIGURA 5.20 - ESTRUTURA DO PROBLEMA



INTERFACES	PARÂMETROS	
	ENTRADA	SAÍDA
1		Registros do Arquivo CATAØ1.
2	Registros do Arquivo CATAØ1	Manhã, Tarde, manhã extrapolada, tarde extrapolada, total código disciplina.
3	Manhã, tarde, manhã extrapolada, tarde extrapolada, total, código de disciplina.	

FIGURA 5.21 - ESTRUTURA E "INTERFACES" DE TOPO

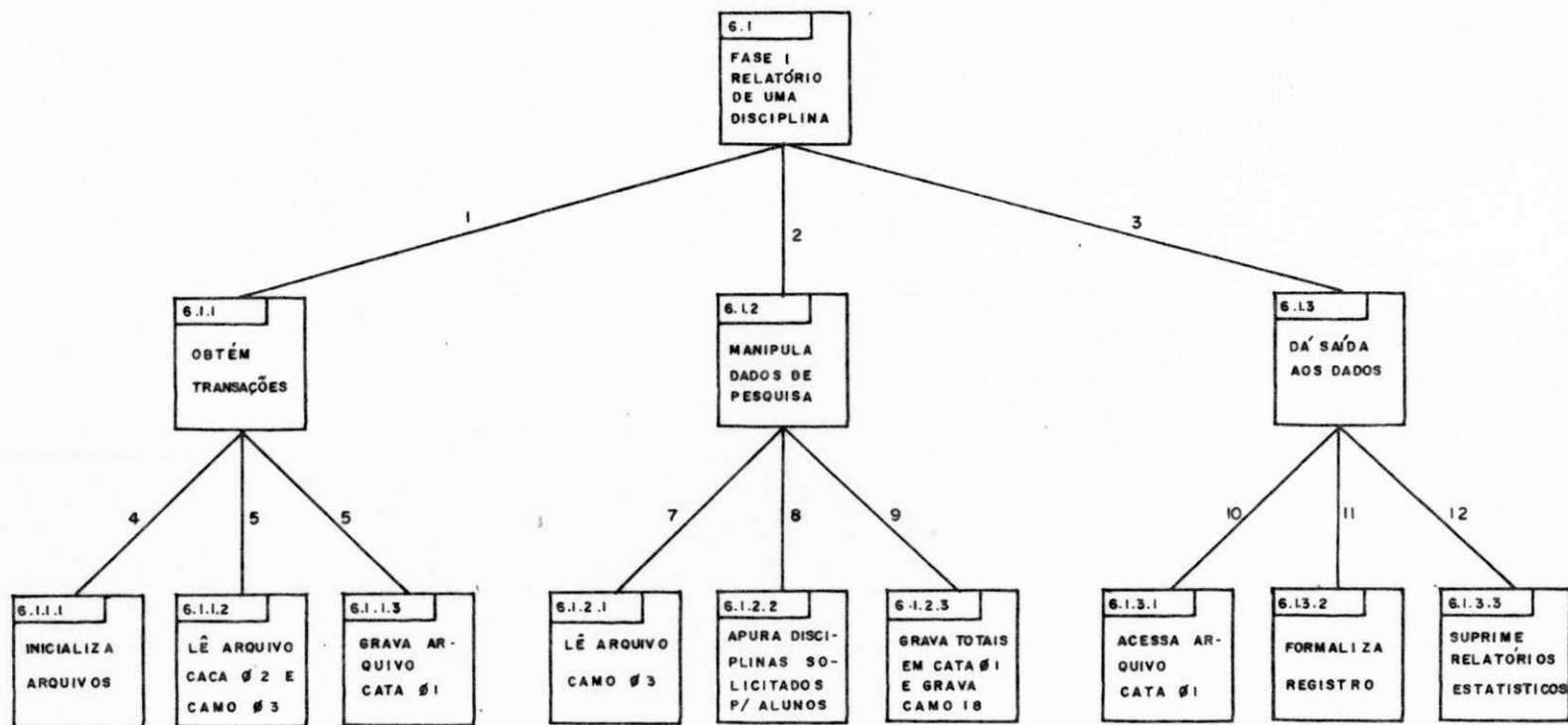


FIGURA 5.22(a) - DIAGRAMA ESTRUTURAL DA FASE I (RELATÓRIO DE UMA DISCIPLINA)

INTER FACES.	P A R Â M E T R O S	
	E N T R A D A	S A I D A
1		Registros do arquivo CATAØ1
2	Registros do arquivo CATAØ1	Manhã, tarde, manhã extrapolade, tarde extrapolada, total, código disciplina
3	Manhã, Tarde, manhã extrapolada, total, código de disciplina	
4		Arquivos CACAØ2 x CATAØ1
5	Arquivo CACAØ2	Código disciplinas nome disciplina, código de curso, manhã extrapolada, tarde extrapolada, tarde, manhã, total alunos participaram de E.D.V.
6	Código curso, código disciplina, manhã extrapolada, tarde extrapolada, tarde, manhã total alunos participaram	
7		Totais de alunos matriculados por cursos registros do arquivo CAM018
8	Totais de alunos matriculados por cursos, registros do arquivo CAM0Ø3	Código disciplina, código curso, manhã, tarde, manhã extrapolada, tarde extrapolada, total de alunos participaram do E.D.V.
9	Código disciplina, código curso, nome disciplina, tarde, manhã, tarde extrapolade, manhã extrapolada, total alunos precipitaram do E.D.V.	
10		Registro do arquivo CATAØ1
11	Registros do arquivo CATAØ1	Registros formalizados
12	Registros formalizados	

FIGURA 5.22(b) - "INTERFACES" DE FASE 1 (RELATÓRIO DE UMA DISCIPLINA).

plinas

Fase 4 - Emissão de totais para quatro disciplinas.

FUNÇÃO DOS MÓDULOS

Módulo: 6.1

Função: Fase 1 - Relatório de Uma Disciplina

Nesta fase são lidos os cartões, contendo os dados da pesquisa junto aos alunos, referentes às disciplinas que desejam cursar em determinado semestre. A partir daí, é criado um arquivo tabela que contém os valores computados, como também um arquivo movimento que contém a imagem dos dados levantados junto aos alunos. É ainda, nesta fase impresso o relatório estatístico.

Módulo: 6.1.1

Função: Obtém Transações

Os dados de transações são obtidos neste módulo quando os arquivos são colocados à disposição do processamento das informações colhidas junto aos alunos.

Módulo: 6.1.1.1

Função: Inicializa Arquivos

Todos os arquivos envolvidos no processamento são abertos ou inicializados, s^o os fechando ap^os a execu^ção do processamento.

Módulo: 6.1.1.2

Função: Lê Arquivos CACAØ2 e CAMOØ3

O arquivo Cadastro Geral de Disciplinas, CACAØ2 é lido a fim de que seja criado um arquivo CAMOØ3 que contém a imagem dos cartões solicita^ções, originários da pesquisa feita junto aos alunos.

Módulo: 6.1.1.3

Função: Grava Arquivo CATAØ1

O arquivo Tabela, CATAØ1, é criado a fim de armazenar valores apurados através dos dados levantados junto aos alunos, referenciando quais as disciplinas que de sejam cursar e os respectivos turnos.

Módulo: 6.1.2

Função: Manipula Dados de Pesquisa

Os dados obtidos junto aos alunos, são ma

nipulados, e os totais são apurados e armazenados nos campos existentes no arquivo CATA01.

Módulo: 6.1.2.1

Função: Lê Arquivo CAM003

O arquivo CAM003, contém os dados de pesquisa levantados junto aos alunos, determinando quais disciplinas desejam cursar e os respectivos turnos. Contém ainda o arquivo CAM003 um cartão inicial que indica o N° total de alunos matriculados em cada curso.

Módulo: 6.1.2.2

Função: Apura Disciplinas Solicitadas por Alunos

Os dados constantes do arquivo CAM003 são manipulados e os totais são estabelecidos mediante contagem feita para posterior gravação em seus campos, pré-estabelecidos. Também, são calculados os valores dos números de vagas para tarde e manhã extrapoladas acrescentando-se 30% aos números de vagas solicitadas pelos alunos, quando da pesquisa.

Módulo: 6.1.2.3

Função: Grava Totais Apurados no Arquivo CATA01 e Grava CAM018.

Os totais apurados são gravados no arquivo

tabela que será utilizado na emissão de relatórios. Como também, a imagem do arquivo CAM003, é gravada no arquivo CAM018 que será utilizado nas demais fases.

Módulo: 6.1.3

Função: Dá Saída aos Dados

Neste módulo, os resultados dos estudos são impressos em forma de relatório.

Módulo: 6.1.3.1

Função: Acessa CATA01

O arquivo CATA01 mantém armazenado os dados estatísticos que indicam os números de solicitações feitas pelos alunos para cada disciplina.

Módulo: 6.1.3.2

Função: Formaliza Registros

De posse das informações contidas em CATA01, os registros a serem impressos em forma de relatórios, são formalizados, apresentando os totais das solicitações feitas para cada disciplina, e por cursos isoladamente, constituindo-se assim, de um total de 15 cursos.

Módulo: 6.1.3.3

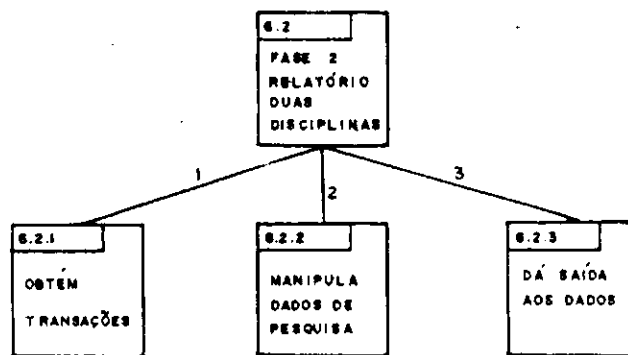
Função: Imprime Registro Formalizado

Neste módulo os registros constados dos resultados estatísticos são impressos em forma de relatórios, que, posteriormente, são enviados aos Departamentos.

5.6.2 - Fase 2 - EMISSÃO DE TOTAIS PARA DUAS DISCIPLINAS



FIGURA 5.23 - ESTRUTURA DO PROBLEMA



INTERFACES	PARÂMETROS	
	ENTRADA	SAÍDA
1		Registros do arquivo CAMOØ4.
2	Registro do arquivo CAMOØ4.	Código disciplinal, código disciplina2, código de curso, manhã tarde, total de alunos participantes do E.D.V.
3	Código disciplinal1, código disciplina2, código de curso, manhã tarde, total de alunos participantes do E.D.V.	

FIGURA 5.24 - ESTRUTURA E "INTERFACES" DO TOPO

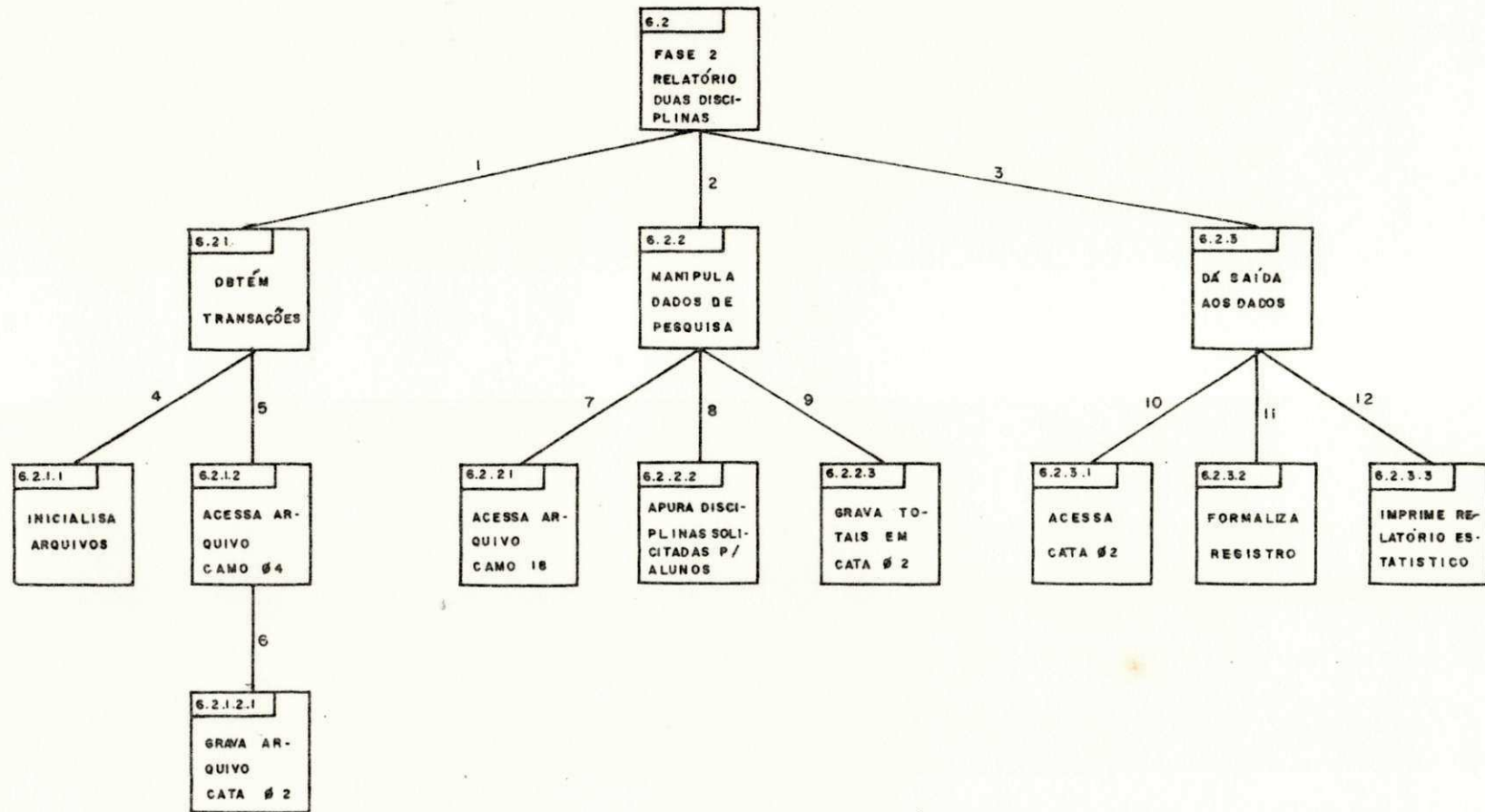


FIGURA 5.25(a) - DIAGRAMA ESTRUTURAL DA FASE 2 (RELATÓRIO DE DUAS DISCIPLINAS COMBINADAS).

INTER FACES.	P A R Â M E T R O S	
	E N T R A D A	S A I D A
1		Registros do arquivo CAM04
2	Registro de arquivo CAM04	Código disciplina 1, código de disciplina 2, código de curso , manhã, tarde, total alunos que participaram E.D.V.
3	Código de disciplinas 1, código de disciplina 2, código de curso, manhã, tarde total de alunos que participaram do E.D.V.	
4		Registro do arquivo CAM04
5	Registros do arquivo CAM04	
6	Registros do arquivo CAM04	
7		Registros de arquivo CAM018
8	Registros de CAM018, registros de CAM04	Código disciplina 1, código disciplina 2, código de curso, manhã, tarde, total alunos participaram E.D.V.
9	Código disciplina 1, código disciplina 2, código curso, manhã, tarde, total alunos participaram E.D.V.	
10		Registros do arquivo CATA2
11	Registro do arquivo CATA2	Registro formalizado
12	Registro formalizado	

FIGURA 5.25(b) - "INTERFACES" DE FASE 2 (RELATÓRIO DE DUAS DISCIPLINAS).

FUNÇÃO DOS MÓDULOS

Módulo: 6.2

Função: Fase 2 - Relatório de Duas Disciplinas

Nesta fase são lidos os cartões contendo dados portadores das combinações das disciplinas duas-a-duas, escolhidas pelas coordenações dos cursos. A partir daí, é criado um arquivo tabela que, conterà os valores computados, e, na fase final do processo é impresso um relatório estatístico.

Módulo: 6.2.1

Função: Obtém Transações

Os dados de transações contendo os códigos de disciplinas duas-a-duas, são obtidos neste módulo, quando os arquivos são colocados à disposição do processamento das informações colhidas junto aos alunos.

Módulo: 6.2.1.1

Função: Inicializa Arquivos

Os arquivos envolvidos no processamento são abertos ou inicializados, só os fechando após todo processamento.

Módulo: 6.2.1.2

Função: Acessa Arquivo CAM004

O arquivo CAM004 é acessado e lido para posteriormente gravar seus registros no arquivo tabela, CATA02, que conterà campos estatísticos para disciplinas duas-a-duas.

Módulo: 6.2.1.2.1

Função: Grava Arquivo CATA02

O arquivo tabela CATA02, é gravado com a imagem do arquivo movimento em cartão CAM004, contendo dentre outros, o campo código de disciplina1. código de disciplina2, que representa a chave do arquivo.

Módulo: 6.2.2

Função: Manipula Dados de Pesquisa

Os dados obtidos junto aos alunos, são manipulados, e os totais são apurados e armazenados nos campos existentes no arquivo CATA02.

Módulo: 6.2.2.1

Função: Acessa Arquivo CAM018

O arquivo movimento CAM018 é acessado, co

locando os registros à disposição do processamento, os quais contêm dados de levantamento junto aos alunos.

Módulo: 6.2.2.2

Função: Apura Disciplinas Solicitadas Pelos Alunos

A partir do arquivo CAM003, faz-se neste módulo, uma contagem dos alunos que solicitaram determinadas disciplinas duas-a-duas, e os resultados são acumulados nos campos; tarde, manhã, tarde extrapolada, manhã extrapolada e total do arquivo CATA02.

Módulo: 6.2.2.3

Função: Grava Totais em CATA02

A operação do número de alunos que escolheram as disciplinas duas-a-duas, com bases no arquivo CAM018, os referidos valores são gravados no arquivo CATA02.

Módulo: 6.2.3

Função: Dá Saída aos Dados

Após o processamento, os dados são preparados para serem impressos em forma de relatórios.

Módulo: 6.2.3.1

Função: Acessa Arquivo CATA02

O arquivo CATA02 é acessado e seus registros são colocados à disposição para emissão de relatórios.

Módulo: 6.2.3.2

Função: Formaliza Registros

A partir das informações contidas no arquivo tabela CATA02, os registros a serem impressos em forma de relatórios são formalizados, apresentando os totais das solicitações feitas para disciplinas duas-a-duas, e por cur sos isoladamente, constituindo-se assim, de um total de 15 cursos.

Módulo: 6.2.3.3

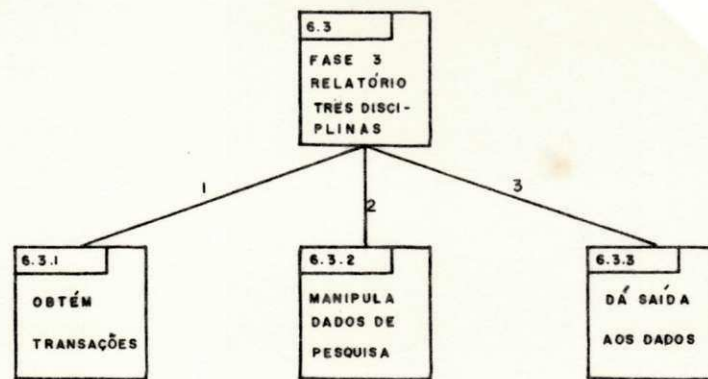
Função: Imprime Relatório Estatístico

Os registros após serem preparados são im pressos, gerando, assim, o relatório estatístico de solicitação de disciplinas duas-a-duas.

5.6.3 - Fase 3 - EMISSÃO DE TOTAIS PARA TRES DISCIPLINAS



FIGURA 5.26 - ESTRUTURA DO PROBLEMA



INTERFACES	PARÂMETROS	
	ENTRADA	SAÍDA
1		Registros do arquivo CAM005
2	Registros do arquivo CAM005	código disciplinal, código disciplina2, código disciplina3, código de curso, manhã, total de alunos que participaram do E.D.V.
3	código disciplinal, código disciplina2, código disciplina3, código curso, tarde, manhã, total de alunos que participaram do E.D.V.	

FIGURA 5.27 - ESTRUTURA E "INTERFACES" DE TOPO.

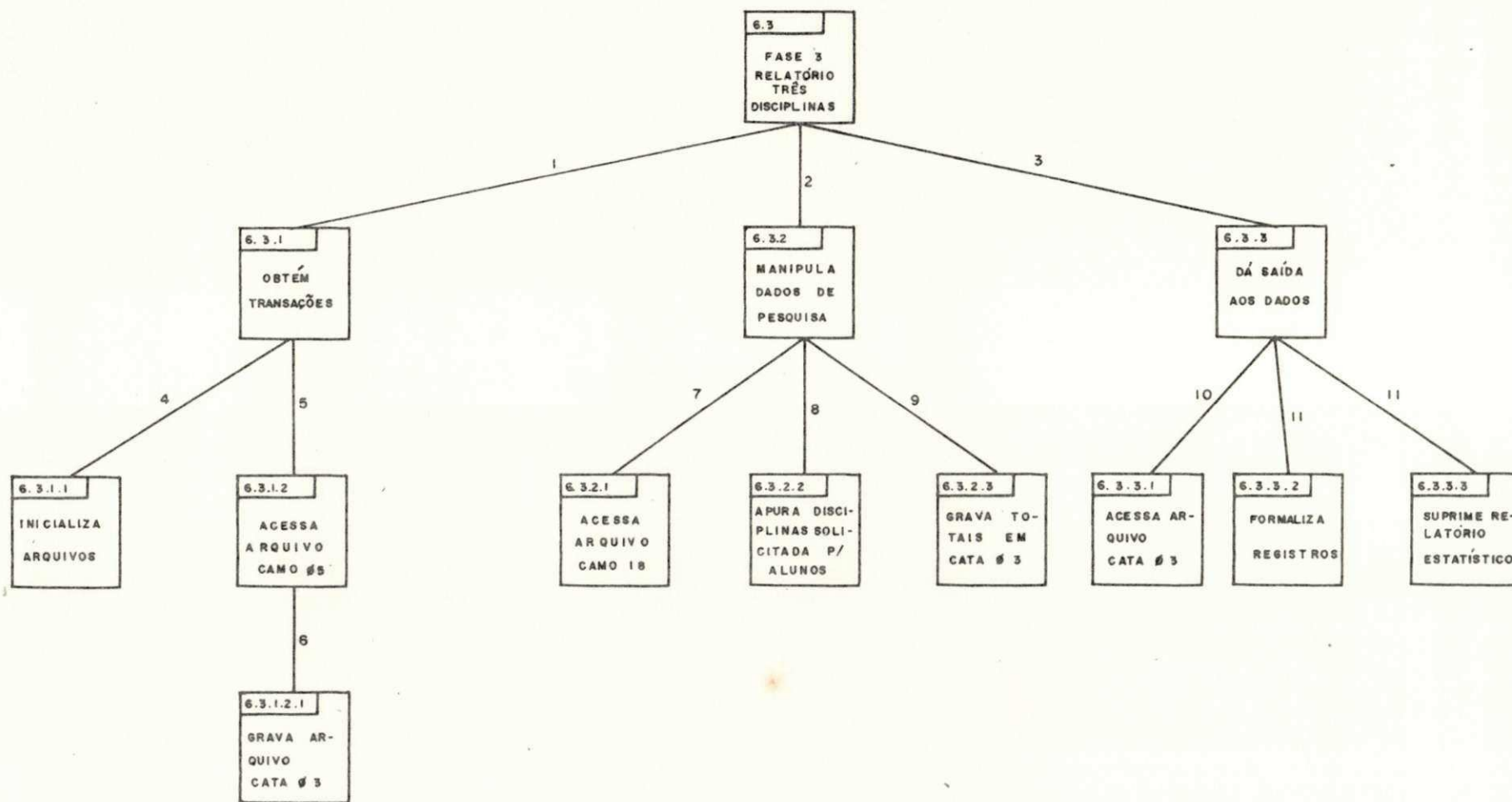


FIGURA 5.28 (a) - DIAGRAMA ESTRUTURAL DA FASE 3 (RELATÓRIO DE DUAS DISCIPLINAS COMBINADAS).

INTER FACES	PARÂMETROS	
	ENTRADA	SAIDA
1		Registro do arquivo CAM007
2	Registro do arquivo CAM007	Registro do arquivo CAM010
3	Registro do arquivo CAM010	
4		Arquivo CAM007
5	Arquivo CAM007	Registros do arquivo CAM007
6	Registro do arquivo CAM007	Registro do arquivo CAM009
7	Registros do arquivo CAM009	Registros do arquivo CAM009, e CACA03
8	Registro de CAM009	
9		Registro do arquivo CAM010
10	Registro	Registro formalizado
11	Registro formalizado	
12	Registros de CAM008 e CACA01	Registros de CAM008
13	Registro de CAM008	Registro de CAM009
14	Registros de CAM009	
15		Registro de CAM009 e CACA03
16	Registros de CAM009 e CACA03	Registro de CACA03
17	Registro de CAM009	
18		Registro de CAM010
19	Registro de CAM010	

FIGURA 5. 28(b) - "Interfaces" da Primeira Matrícula

FUNÇÃO DOS MÓDULOS

Módulo: 6.3

Função: Fase 3 - Relatório de Tres Disciplinas

Nesta fase são lidos os cartões contendo dados com as disciplinas combinadas tres-a-tres escolhidas pelas coordenações dos cursos. A partir destes é criado um arquivo tabela que, conterà os valores computados, e, na fase final do processo, imprime um relatório mostrando os valores computados.

Módulo: 6.3.1

Função: Obtém Transações

Os dados de transações são obtidos neste módulo quando os arquivos são colocados à disposição do processamento das informações colhidas junto aos alunos.

Módulo: 6.3.1.1

Função: Inicializa Arquivos

Os arquivos envolvidos no processamento são abertos ou inicializados, sendo fechados somente ao final do processamento.

Módulo: 6.3.1.2

Função: Acessa Arquivo CAM005

O arquivo movimento CAM005 é acessado e lido, a fim de que, as imagens dos registros sejam gravadas no arquivo tabela CATA03.

Módulo: 6.3.1.2.1

Função: Grava Arquivo CATA03

O arquivo tabela CATA03, é gravado com os registros que representam a imagem dos registros do arquivo em cartão CAM004, contendo - dentre outros - o campo: código disciplina1-código disciplina2-código disciplina3, que funciona como a chave do referido arquivo.

Módulo: 6.3.2

Função: Manipula Dados de Pesquisa

Os dados obtidos junto aos alunos, são manipulados, e os totais resultantes são armazenados nos campos existentes no arquivo CATA01.

Módulo: 6.3.2.1

Função: Acessa Arquivo CAM018

Neste módulo, o arquivo CAM018 é acessado,

colocando os registros (que contêm os totais de solicitações feitas por alunos para cada disciplina) à disposição do processamento, para posterior criação do arquivo CATAØ3.

Módulo: 6.3.2.2

Função: Apura Disciplinas Solicitadas Pelos Alunos

Os dados originários da pesquisa junto aos alunos são apurados, considerando-se as combinações de disciplinas tres-a-tres que são pesquisadas no arquivo CAM018 colocando os resultados acumulados nos campos: tarde, manhã e total, do arquivo CATAØ3.

Módulo: 6.3.2.3

Função: Grava Totais em CATAØ3

A apuração do número de alunos que escolheram as disciplinas apuradas tres-a-tres, com bases no arquivo CAM018, se realiza neste módulo, e, os referidos valores são gravados no arquivo CATAØ2.

Módulo: 6.3.3

Função: Dá Saída aos Dados

Os dados depois de processados, são preparados para serem impressos em forma de relatório.

Módulo: 6.3.3.1

Função: Acessa Arquivo CATA03

O arquivo CATA03 é acessado e seus registros são colocados à disposição para emissão de relatórios estatísticos.

Módulo: 6.3.3.2

Função: Formaliza Registros

De posse das informações contidas no arquivo CATA03, os registros a serem impressos em forma de relatórios são formalizados, apresentando os totais das solicitações feitas pelos alunos, com as disciplinas combinadas tres-a-tres.

Módulo: 6.3.3.3

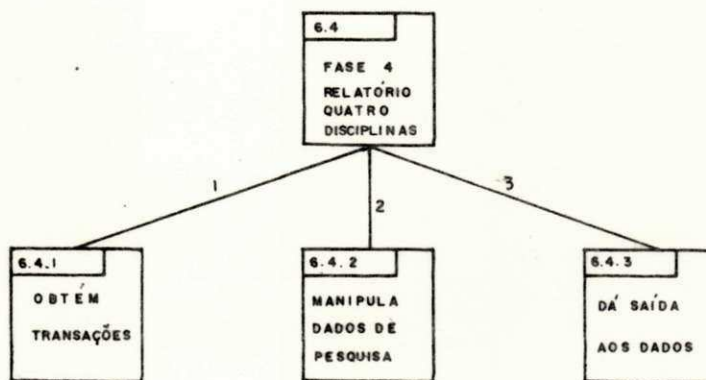
Função: Imprime Registro Formalizado

Os registros após formalização, são impressos, gerando assim, o relatório estatístico das solicitações das disciplinas combinadas tres-a-tres, que serão encaminhados às Coordenações de Cursos.

5.6.4 - Fase 4 - EMISSÃO DE TOTAIS PARA QUATRO DISCIPLINAS



FIGURA 5.29 - ESTRUTURA DO PROBLEMA



INTERFACES	PARAMETROS	
	ENTRADA	SAÍDA
1		Registros do arquivo CAM006
2	Registros do arquivo CAM006	cod.disciplinal, cod. disciplina2, cod. disciplina3, código disciplina4, cod. curso, manhã, tarde, total aluno que part. E.D.V.
3	cod. disciplinal, código discp.2 código discp.3 código discp.4, cod. curso, manhã, tarde, total de alunos partic. do E.D.V.	

FIGURA 5.30 - ESTRUTURA E "INTERFACES" DO TOPO

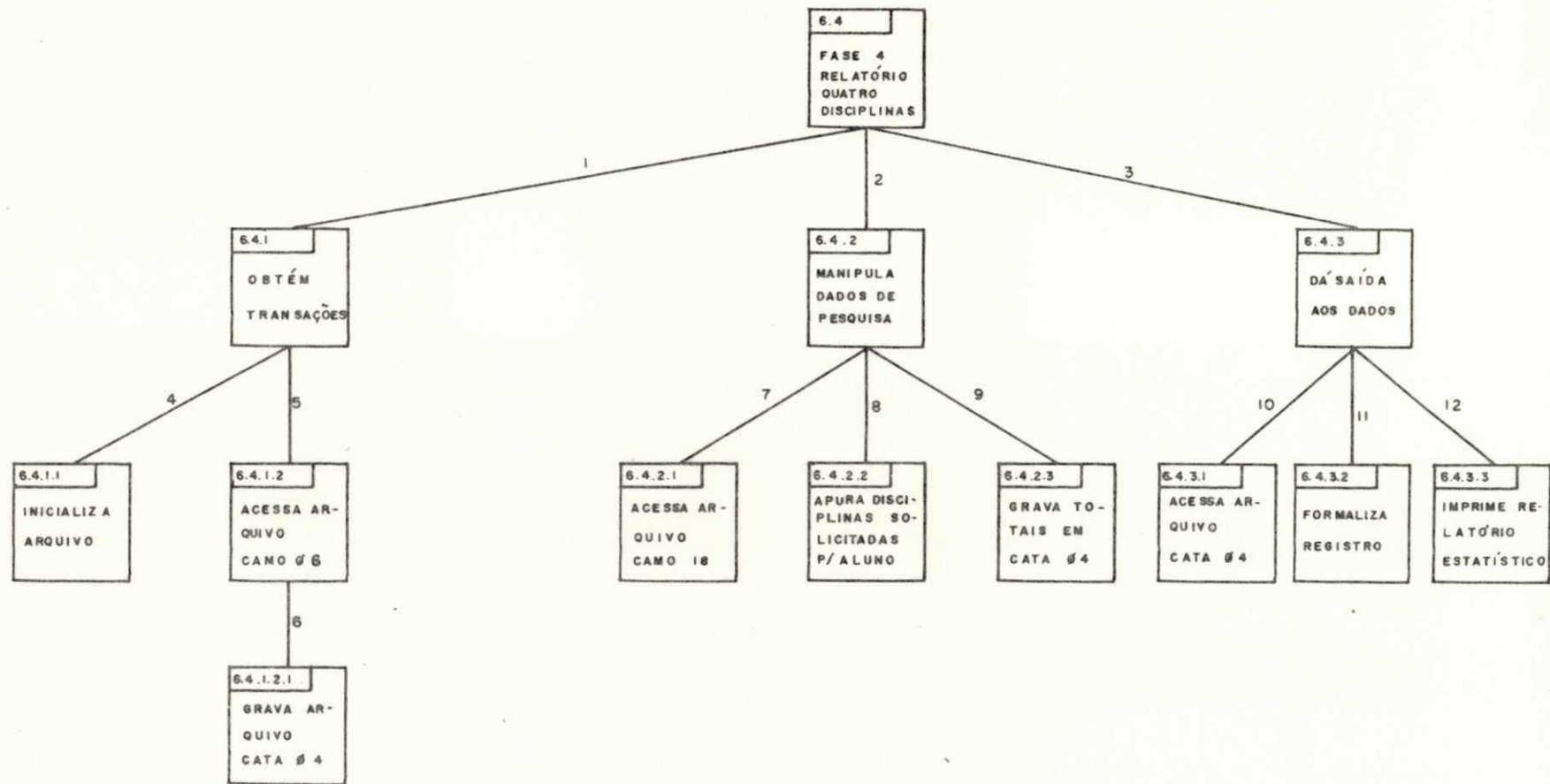


FIGURA 5.31(a) - DIAGRAMA ESTRUTURAL DA FASE 4 (RELATÓRIO DE QUATRO DISCIPLINAS COMBINADAS)

INTER FACES	P A R Â M E T R O S	
	E N T R A D A	S A I D A
1		Registro do arquivo CAM06
2	Registro do arquivo CAM06	Código disciplina 1, código disciplina 2, código disciplina 3, código disciplina 4, código curso, manhã, tarde, total alunos que participaram de E.D.V.
3	Código disciplina 1, código disciplina 3, código disciplina 4, código curso, manhã, tarde, total de alunos que participaram do E.D.V.	
4		Registros do arquivo CAM05
5	Registros do arquivo CAM05	
6	Registros do arquivo CAM018, Registros do arquivo CAM06	
7		Registros do arquivo CAM06
8	Registro do arquivo CAM06 Registro do arquivo CAM018	Código disciplina 1, código disciplina 2, código disciplina 3, código disciplina 4, código curso, manhã, tarde, total dos alunos que participaram do E.D.V.
9	Código disciplina 1, código disciplina 2, código disciplina 3, código disciplina 4, código curso, manhã, tarde, total dos alunos que participaram do E.D.V.	
10		Registros do arquivo CATA4
11	Registros do arquivo CATA4	Registros formalizados
12	Registros formalizados	

FIGURA 5.31(b) - "INTERFACES" DA FASE 4 (RELATÓRIO DE QUATRO DISCIPLINA

FUNÇÃO DOS MÓDULOS

Módulo: 6.4

Função: Fase 4 - Relatório de Quatro Disciplinas

Nesta fase, são lidos os cartões contendo dados com as disciplinas combinadas quatro-a-quatro, escolhidas pelas Coordenações dos Cursos. A partir destes, cria-se o arquivo tabela que armazena os valores computados, e, na fase final do processo imprime-se um relatório estatístico.

Módulo: 6.4.1

Função: Obtém Transações

Os dados são obtidos neste módulo, assim como, os arquivos são colocados à disposição do processamento.

Módulo: 6.4.1.1

Função: Inicializa Arquivos

Todos os arquivos envolvidos no processamento são aqui inicializados, ou abertos, só os fechando após o término da impressão do relatório estatístico.

Módulo: 6.4.1.2

Função: Acessa Arquivo CAM006

O arquivo CAM006, é acessado e lido para posterior gravação da imagem dos seus registros no arquivo CATA04.

Módulo: 6.4.1.2.1

Função: Grava Arquivo CATA04

O arquivo CATA04 é gravado para posteriormente receber nos seus campos: tarde, manhã e total de alunos que participaram do E.D.V., os valores resultantes do processamento das disciplinas solicitadas, e, que foram computadas quatro-a-quatro.

Módulo: 6.4.2

Função: Manipula Dados

Neste módulo, os dados obtidos junto aos alunos, são manipulados procedendo-se com as apurações, no número de solicitações, combinando-se disciplinas quatro-a-quatro.

Módulo: 6.4.2.1

Função: Acessa Arquivo CAM018

O arquivo CAM018, é acessado e lido colocando os registros à disposição do processamento, onde as operações das disciplinas solicitadas e combinadas quatro-a-quatro são efetuadas com bases nos dados do referido arquivo.

Módulo: 6.4.2.2

Função: Apura Disciplinas Solicitadas Pelos Alunos

Os dados originários da pesquisa junto aos alunos são apurados neste módulo.

Módulo: 6.4.2.3

Função: Grava Totais no Arquivo CATA04

Os valores resultantes das apurações das disciplinas solicitadas, e, combinadas quatro-a-quatro, são gravados no arquivo tabela CATA02, para posterior impressão.

Módulo: 6.4.3

Função: Dá Saída aos Dados

Os dados após processamento são preparados para serem impressos em forma de relatório.

Módulo: 6.4.3.1

Função: Acessa Arquivo CATA04

O arquivo CATAØ4 é acessado, lido, e, posteriormente seus registros são preparados para emissão de relatórios.

Módulo: 6.4.3.2

Função: Formaliza Registros

Os registros que serão impressos, constando das disciplinas combinadas quatro-a-quatro são formalizados neste módulo.

Módulo: 6.4.3.3

Função: Imprime Relatório Estatístico

Após a formalização dos registros, os mesmos são impressos em forma de relatórios, e, posteriormente, os referidos relatórios são encaminhados às Coordenações de Cursos.

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 832 - Tel (083) 321-7222-R 355
58.100 - Campina Grande - Paraíba

5.7 - SUBSISTEMA MATRÍCULA

Após o Estudo de Demanda de Vagas e da criação do Arquivo Cadastro Semestral de Disciplinas, antes do início do período, o aluno deve solicitar matrícula nas disciplinas de sua conveniência, passando assim, em alguns casos por duas fases do Subsistema denominadas:

a) Primeira Matrícula

b) Segunda Matrícula

As Coordenações de Cursos com bases no Estudo de Demanda de Vagas, elaboram horários, constando das salas e disciplinas oferecidas que serão posteriormente levados ao conhecimento dos alunos.

Quando da solicitação da matrícula, o aluno deve preencher o documento denominado Solicitação de Matrícula, constando de : número de inscrição do aluno, código do curso, o nome de no máximo onze disciplinas e suas prioridades, onde cuidados devem ser tomados para não haver choque de horários, e, a partir do qual inicia-se o processamento da matrícula do aluno nas disciplinas solicitadas.

i) Matricular os alunos da Universidade nas disciplinas por eles solicitadas, levando-se em considere

ração a prioridade dada pelos mesmos a cada disciplina, cons
tante do documento de solicitação de matrícula.

ii) O sistema deve ser capaz de verificar a nível de consistência, se as disciplinas solicitadas quan
do da matrícula, fazem parte do currículo de curso do aluno que as solicitou, como também, se o referido aluno já cursou os pré-requisitos das referidas disciplinas. Para tais veri
ficações, faz-se-á uma pesquisa nos Arquivos: Currículos e Cursos - CACA05 e Arquivo Cadastro de Alunos - CACA01.

iii) Processar a matrícula, dando priori
dade aos alunos mais antigos, com relação ao ano de ingresso na Universidade.

iv) Neste sistema a fase de consistência foi excetuada.

v) Após o processamento da Primeira Matrí
cula, manter gravado um arquivo constando dos nomes de todos os alunos matriculados, como também daqueles que embora te
nham solicitado matrículas em determinadas disciplinas não as tenham conseguido, por falta de vagas ou coincidência de horários com outra disciplina. Tais impasses, serão solucio
nados na Segunda Matrícula.

vi) Após a Primeira Matrícula, emitir re

latórios por cursos, para cada aluno, contendo a relação das disciplinas por eles solicitadas indicando, em cada, a obtenção de matrícula ou não. De posse destas informações, os alunos que não forem atendidos nas suas solicitações poderão refazê-las na Segunda Matrícula, onde o subsistema tentará solucionar o problema, com bases em aumento de número de vagas ou criação de novas turmas. Como também, o próprio aluno poderá substituir qualquer disciplina impedida, por outra.

vii) O aluno que se sentir prejudicado, deverá solicitar ao Controle Acadêmico, através das Coordenações de Curso, o seguinte:

- a) No máximo dois adicionamento / cancelamento de disciplina
- b) No máximo duas trocas de disciplinas

No caso adicionamento/cancelamento, se o campo prioridade for preenchido, a operação será executada independentemente da verificação de coincidências de horários ou de número de vagas.

viii) Emitir Relatórios de Frequência Escolar (Diário de Classe).

ix) Criar arquivo Cadastro Semestral de

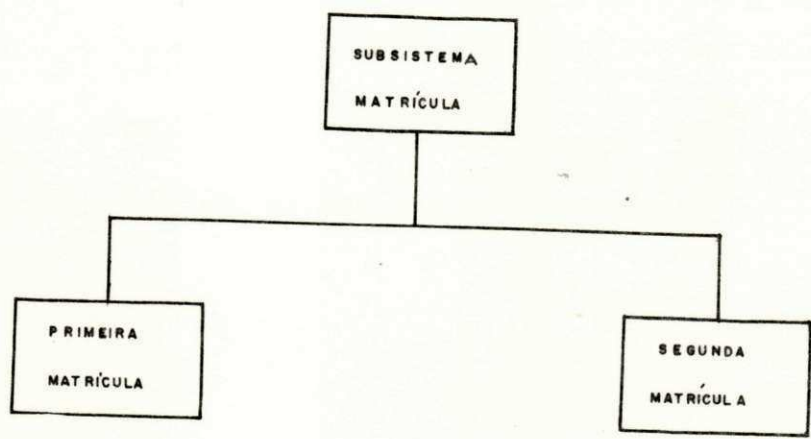
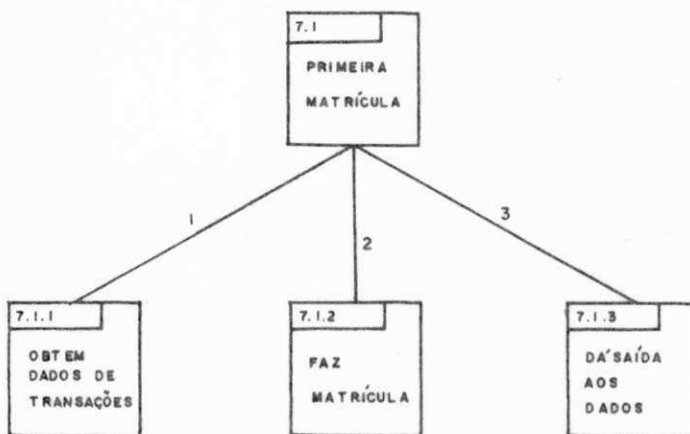


FIGURA 5.32 - SUBDIVISÕES DO SUBSISTEMA MATRÍCULA

5.7.1 - PRIMEIRA MATRÍCULA



FIGURA 5.33 - ESTRUTURA DO PROBLEMA



INTERFACES	PARÂMETROS	
	ENTRADA	SAÍDA
1		Registro do Arquivo CAM007.
2	Registro do Arquivo CAM007	Registro do Arquivo CAM010
3	Registro do Arquivo CAM010.	

FIGURA 5.34 - ESTRUTURA E "INTERFACE" DE TOPO.

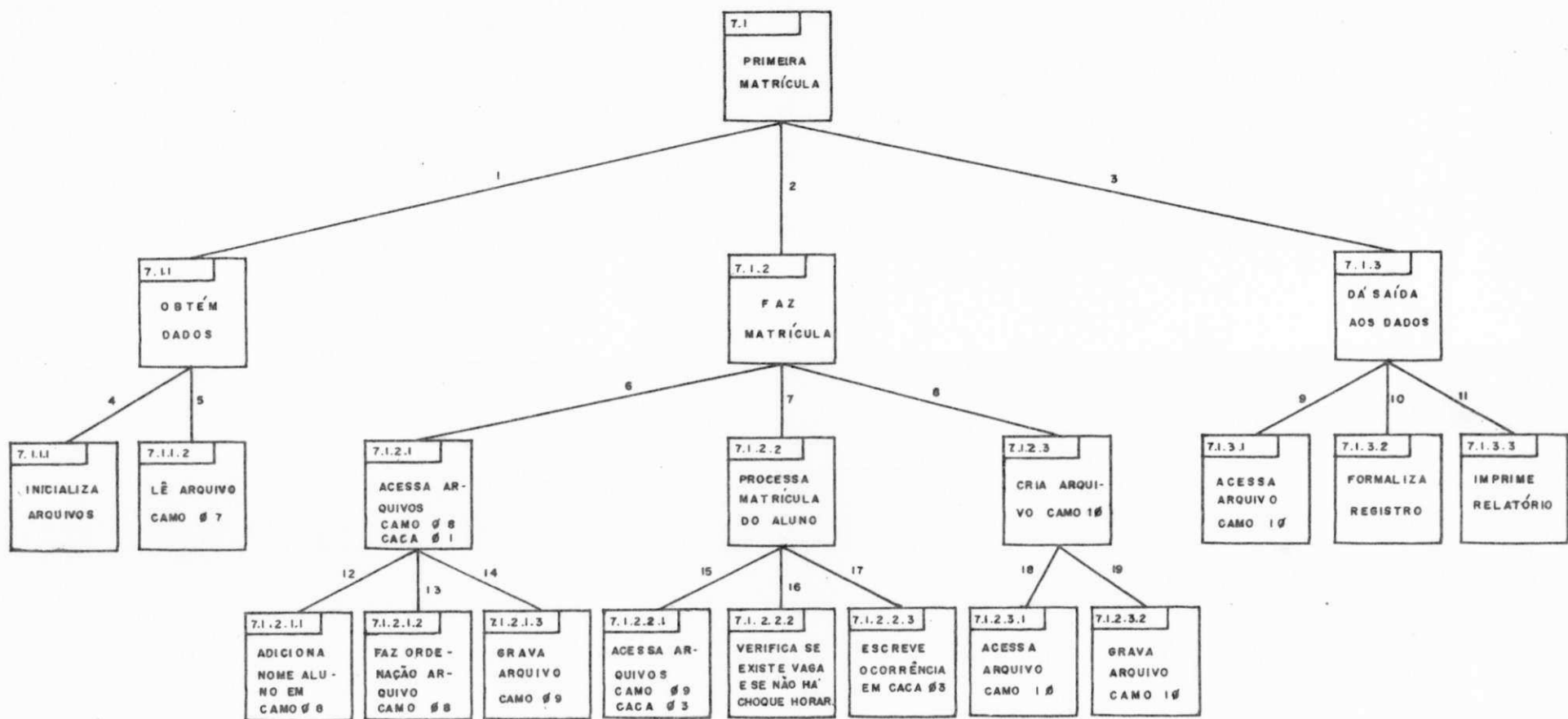


FIGURA 5.35(a) - DIAGRAMA ESTRUTURAL DO SUBSISTEMA DA PRIMEIRA MATRÍCULA

INTER FACES	P A R Â M E T R O S	
	E N T R A D A	S A I D A
1	Registro do arquivo CAM07	Registro do arquivo CAM07
2	Registro do arquivo CAM10	Registro do arquivo CAM10
3	Registro do arquivo CAM10	
4		Arquivo CAM07
5	Arquivo CAM07	Registros do arquivo CAM07
6	Registro do arquivo CAM07	Registro do arquivo CAM09
7	Registros do arquivo CAM09	Registro do arquivo CAM09, e CACA3
8	Registro de CAM09	
9		Registro do arquivo CAM10
10	Registro do arquivo CAM10	Registro formalizado
11	Registro formalizado	
12	Registro de CAM08 e CACA1	Registros de CAM08
13	Registro de CAM08	Registro de CAM09
14	Registros de CAM09	
15		Registro de CAM09 e CACA3
16	Registros de CAM09 e CACA3	Registro de CACA 3
17	Registro de CAM09	
18		Registro de CAM10
19	Registro de CAM10	

FIGURA 5.35(b) - "INTERFACES" DA PRIMEIRA MATRÍCULA

Disciplinas/Alunos a fim de ser utilizado posteriormente, no Subsistema Processamento de Notas.

FUNÇÃO DOS MÓDULOS:

Módulo: 7.1.1

Função: Obtém Dados de Transações

A partir do documento solicitação de matrícula, são perfurados os cartões que constituem o arquivo CAM007.

Módulo: 7.1.1.1

Função: Inicializa Arquivos

São abertos ou inicializados todos arquivos envolvidos no processamento, só os fechando após executadas todas as operações inerentes ao processo da primeira matrícula.

Módulo: 7.1.1.2

Função: Lê Arquivo CAM007

Neste módulo, são lidos os registros do arquivo movimento CAM007, em cartão, contendo as solicitações

ções de matrículas feitas pelos alunos.

Módulo: 7.1.2

Função: Faz Matrícula

A matrícula dos alunos é executada neste módulo, gerando um arquivo que contém ocorrências de matrículas, tais como: aluno matriculado em determinada disciplina, e, aluno não matriculado por falta de vaga.

Módulo: 7.1.2.1

Função: Acessa Arquivos CAM008 e CACA01

Os arquivos CAM008 e CACA01 são acessados, e os registros são colocados à disposição do processamento de matrícula.

Módulo: 7.1.2.1.1

Função: Adiciona Nome do Aluno em CAM008

O arquivo CAM008 é gravado a partir do arquivo em cartão CAM007. Posteriormente, grava os nomes dos alunos em CAM008, a partir do arquivo CACA01.

Módulo: 7.1.2.1.2

Função: Faz Ordenação do Arquivo CAM008.

O arquivo CAM008 é ordenado sobre a chave inscrição do aluno, a fim de que os alunos mais antigos tenham prioridades na matrícula.

Módulo: 7.1.2.1.3

Função: Grava Arquivo CAM009

Após ordenação do arquivo CAM008, é gravado o arquivo CAM009, ordenado sobre a chave: inscrição do aluno.

Módulo: 7.1.2.2

Função: Processa Matrícula de Aluno

Neste módulo, dá-se o processamento de matrícula de alunos, constando em se marcar no campo reservado do arquivo CAM010, tal ocorrência atualizando o campo: situação.

Módulo: 7.1.2.2.1

Função: Acessa Arquivos CAM009 e CACA03

Os arquivos CAM009 e CACA03 são acessados, e, seus registros são colocados à disposição do processamento da matrícula, neste módulo.

Módulo: 7.1.2.2.2

Função: Verifica se Existe Vagas e se Não Há Choques de Horários

De posse do registro de um aluno, constando das disciplinas e horário por ele solicitadas, faz-se uma pesquisa no arquivo Cadastro Semestral de Disciplinas CACAØ3, e verifica se existe vaga, e, se o horário coincide com o desejado pelo aluno. Caso positivo, verifica se o aluno não solicitou disciplinas com horários coincidentes.

Módulo: 7.1.2.2.3

Função: Escreve Ocorrência em CACAØ3

Após verificar que existe vaga, e que o aluno pode ser matriculado na disciplina solicitada, faz-se a subtração de uma unidade no campo vagas remanescentes do arquivo CACAØ3, e, o aluno é considerado matriculado. Se houver empecilho, é colocada uma marca no registro do aluno em CAMØØ9.

Módulo: 7.1.2.3

Função: Cria Arquivo CAMØ1Ø

Neste módulo, o arquivo movimento CAMØ1Ø é criado constando das ocorrências de matrícula, ou seja: alunos matriculados e não matriculados.

Módulo: 7.1.2.3.1

Função: Acessa Arquivo CAM010

O arquivo CAM010 é acessado e seus registros são colocados à disposição do processamento de matrícula.

Módulo: 7.1.2.3.2

Função: Grava Arquivo CAM010

O arquivo CAM010 é gravado a partir dos arquivos CAM009 e CACA03, seus registros conterão as ocorrências de matrículas.

Módulo: 7.1.3

Função: Dar Saída aos Dados

Este módulo, dá saída aos dados em forma de relatório após o processamento da matrícula.

Módulo: 7.1.3.1

Função: Acessa Arquivo CAM010

Os registros do arquivo CAM010 são acessados para posteriormente serem impressos.

Módulo: 7.1.3.2

Função: Formaliza Registros

Os dados a serem impressos, são préviam
te formalizados neste módulo.

Módulo: 7.1.3.3

Função: Imprime Relatórios para Alunos

São impressos relatórios para cada aluno participante da matrícula, constando a situação, no que se refere às ocorrências de matrícula, tais como: aluno matri
culado ou não matriculado em determinada disciplina, e, a causa da não matrícula.

5.7.2 - SEGUNDA MATRÍCULA

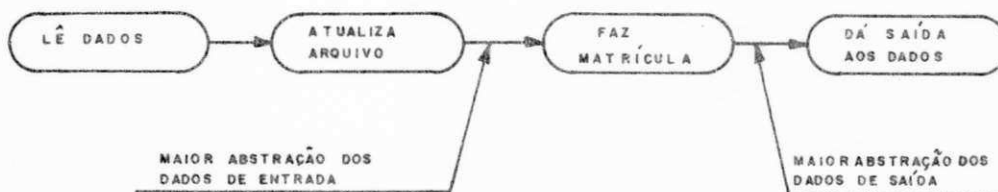
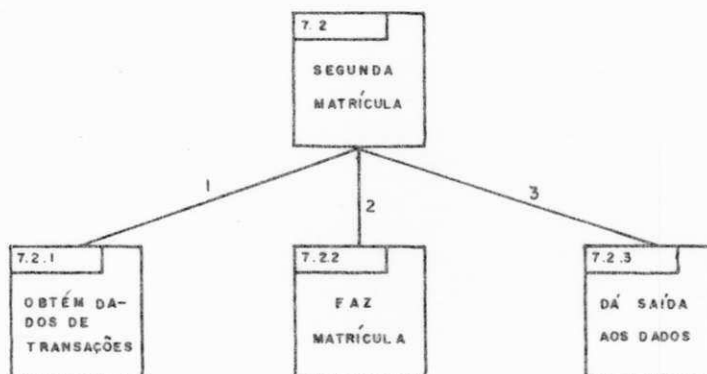


FIGURA 5.36 - ESTRUTURA DO PORBLEMA



INTERFACES	PARÂMETROS	
	ENTRADA	SAÍDA
1		Arquivos CAM012, CAM013 e CACA03.
2	Arquivos CAM012, CAM013, e CACA03.	Arquivo CAM010 registro do arquivo CAM014
3	Arquivos CAM010 e registro do arquivo CAM014.	

FIGURA 5.37 - ESTRUTURA E "INTERFACES" DE TOPO

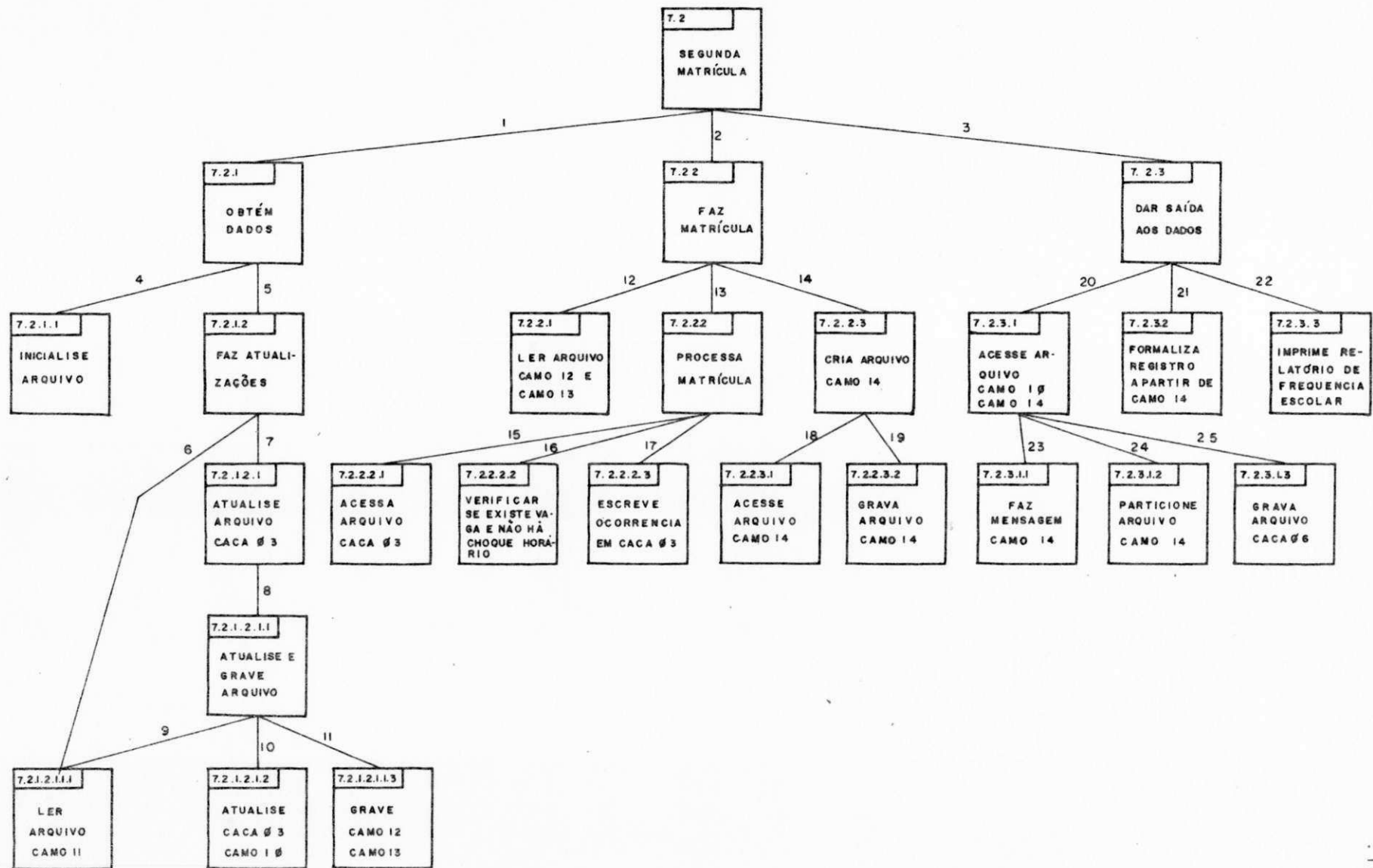


FIGURA 5.30(a) - DIAGRAMA ESTRUTURAL DO SUBSISTEMA SEGUNDA MATRÍCULA

INTER FACES	P A R Â M E T R O S	
	E N T R A D A	S A I D A
1		Arquivos CAMO 12, CAMO13, CACAø3
2	Arquivos CAMO12, CAMO13 e CACAø3	Arquivo CAMO2ø e registro do arquivo CAMO 14
3	Arquivo CAMO1ø e registro do arquivo CAMO14	
4		Arquivos CACAø3, CAMO12 e CAMO13
5	Arquivos CAMO1ø, CAMO11, CAMO12, CAMO13 e CACAø3	Arquivos CAMO1ø, CAMO11, CAMO12, CAMO13 e CACAø3
6	Arquivo CAMO 11	Registros do arquivo CAMO11
7	Arquivos CAMO1ø, CAMO11, CAMO12, CAMO13 e CACAø3	Arquivos CAMO1ø, CAMO11, CAMO12, CAMO13 e CACAø3
8	Arquivos CAMO1ø, CAMO11,, CAMO12, CAMO13 e CACAø3	Arquivos CAMO1ø, CAMO22, CAMO12, CAMO13 e CACAø3
9	Arquivo CAMO 11	Registros do arquivo CAMO11
10	Arquivo CACAø3 e CAMO1ø. Re gistros de arquivo CAMO22	Registros dos arquivos CAMO12 e CAMO13
11	Registros dos arquivos CAMO 12 e CAMO13	
12	Arquivos CAMO12, CAMO13 e CACAø3	Registros dos arquivos CAMO12, CAMO13, e CACAø3
13	Registros do arquivo CAMO12, CAMO13 e CACAø3	Arquivo CACAø3 atualizado
14	Arquivos CAMO 12 e CAMO13	
15	Arquivo CACAø3	Registros do arquivo CACAø3
16	Registros de CACAø3, CAMO13 e CAMO 12	Registro de CACAø3
17	Registro de CACAø3	
18	Arquivo CAMO 14	Registro do arquivo CACAø3
19	Registro do arquivo CAMO14	
20	Arquivos CAMO1ø e CAMO14	Registros do arquivo CAMO14
21	Registros do arquivo CAMO14	Registros formalizados
22	Segue...	

Cont...

INTER FACES	P A R Â M E T R O S	
	E N T R A D A	S A I D A
22	Registros formalizados	
23	Registros do arquivo CAM01Ø e CAM014	
24	Registros do arquivo CAM014	Registros de Arquivo CAM014
25	Registro do arquivo CAMO 14	

FIGURA 5.38(b) - INTERFACES DE SEGUNDA MATRICULA

FUNÇÃO DOS MÓDULOS

Módulo: 7.2.1

Função: Obtém Dados de Transações

Neste módulo, todos os dados envolvidos no processamento são colocados à disposição do mesmo.

Módulo: 7.2.1.1

Função: Inicializa Arquivos

Todos os arquivos envolvidos no processamento são abertos ou inicializados neste módulo, só os fechando, após o término do mesmo.

Módulo: 7.2.1.2

Função: Faz Atualizações

Neste módulo os arquivos CACAØ3 e CAM011 são atualizados.

Módulo: 7.2.1.2.1

Função: Atualiza o Arquivo CACAØ3

A partir do arquivo movimento, em cartões, CAM011, o arquivo Cadastro Semestral de Disciplinas CACAØ3

sofre atualização nos campos: número de vagas oferecidas, o qual possibilita novas matrículas de alunos não atendidos na Primeira Matrícula.

Módulo: 7.2.1.2.1.1

Função: Atualiza e Grava Arquivo

O arquivo CACA03 é novamente atualizado, como também, o arquivo movimento, em disco CAM010 que contém as ocorrências da Primeira Matrícula. Na atualização de CAM010 é dada prioridade aos alunos em fase de conclusão de curso, no que se refere ao cancelamento de matrículas ou substituição de disciplinas.

Módulo: 7.2.1.2.1.1.1

Função: Lê Arquivo CAM011

O arquivo CAM011 contém as solicitações de cancelamento ou substituição de disciplinas, feitas pelos alunos, nesta fase da Segunda Matrícula.

Módulo: 7.2.1.2.1.1.2

Função: Atualiza CACA03 e CAM010

O arquivo CACA03, tem pela segunda vez os campos de totais de vagas oferecidas, atualizados. O arquivo CAM010, que contém as ocorrências da Primeira Matrícula

é mais uma vez atualizado. Mais precisamente, substituindo algumas disciplinas ou cancelando-as, a partir do arquivo CA M011.

Módulo: 7.2.1.2.1.1.3

Função: Grava CAM012 e CAM013

É criado o arquivo movimento em disco CA M012, contendo os alunos que não conseguiram matrícula, quando da atualização anterior. Como também, é criado o arquivo movimento CAM013, contendo os alunos que conseguiram matrículas.

Módulo: 7.2.2

Função: Faz Matrícula

Neste módulo, executa-se algumas matrículas.

Módulo: 7.2.2.1

Função: Lê Arquivo CAM012 e CAM013

Os arquivos CAM012 e CAM013 são lidos e seus registros colocados à disposição do processamento de matrículas.

Módulo: 7.2.2.2

Função: Processa Matrícula

Neste módulo, dar-se-ão as operações inerentes à matrícula.

Módulo: 7.2.2.3

Função: Cria Arquivo CAM014

Após a matrícula, o arquivo CAM014 é criado contendo a relação dos alunos que conseguiram matrícula.

Módulo: 7.2.2.2.1

Função: Acessa Arquivo CACA03

O arquivo CACA03 é acessado para atualização dos registros, indicando alunos matriculados ou não matriculados.

Módulo: 7.2.2.2.2

Função: Verifica Se Existe Vaga e Se Não Há Choque de Horários.

A verificação da existência de vaga é feita lendo-se cada campo: número de vaga do arquivo CACA03, a partir do arquivo CAM012. Em seguida, faz-se a verificação da existência de choque de horários, tomando-se, também, o arquivo CACA03. Se não houver empecilho, a matrícula se rea

lizará.

Módulo: 7.2.2.2.3

Função: Escreve Ocorrência em CACAØ3

Após efetivação da matrícula, o arquivo CACAØ3 é atualizado, a partir do arquivo CAM012, que contém os alunos que conseguiram matrículas.

Módulo: 7.2.2.3.1

Função: Acessa Arquivo CAM014

Após a matrícula, o arquivo movimento, CAM014, em disco, é acessado para posterior gravação de to dos alunos que conseguiram matrículas, com bases no arqui vo CAM012.

Módulo: 7.2.2.3.2

Função: Grava Arquivo CAM014

O arquivo CAM014 contendo todos alunos ma triculados é gravado neste módulo.

Módulo: 7.2.3

Função: Dá Saída aos Dados

Neste módulo, verifica-se a saída do Rela

tório de Frequencia Escolar.

Módulo: 7.2.3.1

Função: Acessa Arquivos CAM01Ø e CAM014

Os arquivos movimento, CAM01Ø e CAM014 são acessados para posteriormente serem submetidos a uma operação de "Merge".

Módulo: 7.2.3.2

Função: Formaliza Registro a Partir de CAM014

São aqui formalizados os registros para posteriormente serem impressos em forma de relatórios.

Módulo: 7.2.3.3

Função: Imprime Relatórios de Frequência Escolar

Os relatórios de Frequência Escolar, contém a relação dos alunos matriculados por disciplina/curso.

Módulo: 7.2.3.1.1

Função: Faz "Merge" em CAM01Ø e CAM014

Os arquivos CAM01Ø e CAM014 são submetidos a uma operação de "Merge", juntando-se os alunos que foram matriculados em CAM014.

Módulo: 7.2.3.1.2

Função: Particiona Arquivo CAM014

O arquivo CAM014 é particionado, gerando os registros com chaves disciplina/aluno, devidamente ordenados.

Módulo: 7.2.3.1.3

Função: Grava Arquivo CACA06

A partir dos registros particionados de CAM014 é gravado o arquivo cadastro CACA06, tendo como chave o campo disciplina/aluno.

5.8 - SUBSISTEMA PROCESSAMENTO DE NOTAS

No fim de cada semestre, quando as atividades inerentes ao desenvolvimento de cada curso se concluiu, inicia-se a ativação do Subsistema Processamento de Notas, onde, as notas de cada aluno é processada, obtendo-se em seguida as respectivas médias que indicarão o aproveitamento dos mesmos.

Desta forma, o referido subsistema tem como finalidades, o que se segue:



i) Manter arquivadas, temporariamente, todas as notas obtidas pelos alunos, constantes no Boletim Escolar, como também, suas situações nas respectivas disciplinas com relação às suas frequências.

ii) A partir das notas de cada aluno, calcular as médias obtidas pelos mesmos nas respectivas disciplinas.

iii) Atualizar o Cadastro Semestral de Alunos Matriculados, isto é: preenchendo os campos reservados para médias.

iv) Gravar no arquivo cadastro de alunos, os códigos das disciplinas por eles cursadas e seus respectivos conceitos.

v) Imprimir Boletim de Aproveitamento por aluno.

vi) Imprimir relação de disciplinas com as respectivas notas (conceitos).

FUNÇÃO DOS MÓDULOS

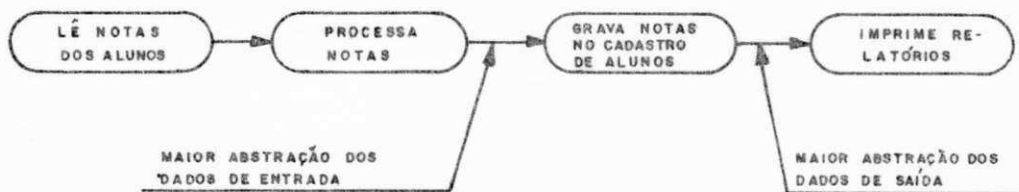
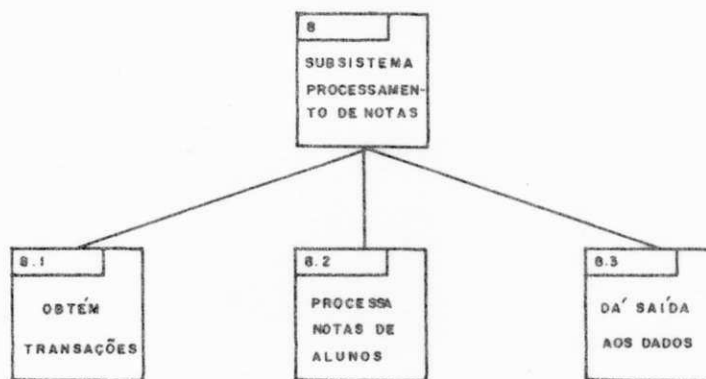


FIGURA 5.39 - ESTRUTURA DO PROBLEMA



INTERFACES	PARÂMETROS	
	ENTRADA	SAÍDA
1		Registros do arquivo CAM015.
2	Registros do arquivo CAM015.	Arquivo CAM014
3	Arquivo CAM014	

FIGURA 5.40 - ESTRUTURA E "INTERFACES" DO PROCESSAMENTO DE NOTAS.

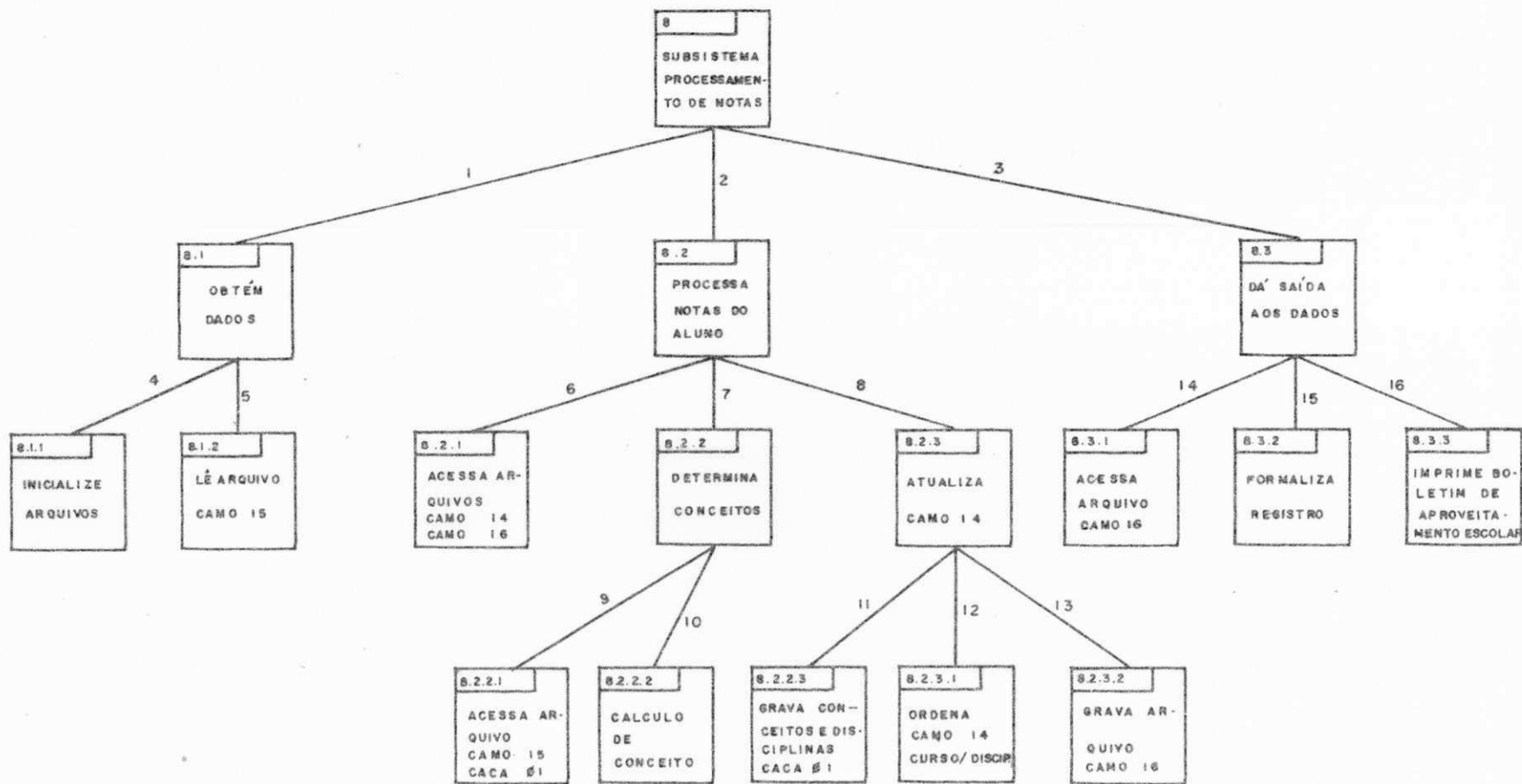


FIGURA 5.41(a) - DIAGRAMA ESTRUTURAL DO SUBSISTEMA PROCESSAMENTO DE NOTAS

INTER FACES	P A R Â M E T R O S	
	E N T R A D A	S A Í D A
1		Registros do arquivo CAM015
2	Arquivo CAM015	Arquivo CAMO 14
3	Arquivo CAM014	
4		Arquivo CAMO 15
5	Arquivo CAMO 15	Registros do arquivo CAM015
6		Registros dos arquivos CAM014 e CAM016
7	Arquivo CAM015	Código de disciplina, inscrição do aluno, conceito
8	Registros arquivo CAM015	Registros de arquivo CAM016
9		Registros dos arquivos CAM015 e CACAØ1
10	Registros do arquivo CAM015	Código de disciplina, inscrição do aluno, conceito
11	Código disciplina, inscri- ção do aluno, conceito	
12	Registro do arquivo CAM014	Registro do arquivo CAM014, or- denado p/chave curso/disciplina
13	Registro do arquivo CAM014 ordenado p/chave Curso/dis- ciplina	Arquivo CAM016
14	Arquivo CAM016	Registro do arquivo CAM016
15	Registros do arquivo CAM016	Registros formalizados
16	Registros formalizados	

FIGURA 5.41(b) - "INTERFACES" DO SUBSISTEMA PROCESSAMENTO DE NOTAS.

Módulo: 8.1

Função: Obtém Dados

Os dados necessários ao processamento são obtidos neste módulo.

Módulo: 8.1.1

Função: Inicializa Arquivos

Todos arquivos envolvidos no processamento são abertos ou inicializados neste módulo, são os fechando após o processamento.

Módulo: 8.1.2

Função: Lê Arquivo CAM015

O arquivo CAM015 contém os dados obtidos a partir do Diário de Classe, constando das notas obtidas pelos alunos no desenvolvimento das disciplinas cursadas. Como também, dados referentes às frequências dos referidos alunos.

Módulo: 8.2

Função: Processa Notas dos Alunos

Os dados são aqui processados, tais como: processamento de notas e frequência dos alunos.

Módulo: 8.2.1

Função: Acessa Arquivos CACA06 e CAM016

Os arquivos CACA06 e CAM016 são acessa
dos, e, seus registros colocados à disposição do processa
mento.

Módulo: 8.2.2

Função: Determina Conceitos

Neste módulo, são determinados os concei
tos dos alunos nas diversas disciplinas cursadas.

Módulo: 8.2.2.1

Função: Acessa Arquivo CAM015 e CACA01

Os arquivos CAM015 e CACA01 são acessa
dos e seus registros são colocados à disposição do processa
mento de notas.

Módulo: 8.2.2.2

Função: Cálculo de Conceito

De posse das notas obtidas pelos alunos,
que se encontram gravadas no arquivo CAM015, o cálculo dos
conceitos dar-se-á neste módulo.

Módulo: 8.2.2.3

Função: Grava Conceitos e Disciplinas em CACA01

O arquivo Cadastro de Alunos CACA01, tem seus campos, que se encontravam previamente reservados, e, gravados os códigos das disciplinas por eles cursadas e seus respectivos conceitos.

Módulo: 8.2.3

Função: Atualiza CACA06

O arquivo CACA06 contém as ocorrências de matrículas, ou melhor, todos os registros disciplina /aluno. Neste módulo, o campo reservado para os conceitos, são gravados.

Módulo: 8.2.3.1

Função: Ordena CAM014 Curso/Disciplina

O arquivo movimento CAM014 é ordenado pela chave: curso/disciplina.

Módulo: 8.2.3.2

Função: Grava Arquivo CAM016

Neste módulo, grava-se o arquivo CAM016 em decorrência da ordenação do arquivo CAM014.

Módulo: 8.3

Função: Dá Saída aos Dados

Este módulo, destina-se a dar saída aos dados sob a forma de relatório.

Módulo: 8.3.1

Função: Acessa Arquivo CAM016

O arquivo CAM016 é acessado, e, os registros colocados à disposição do processamento, para posterior impressão dos mesmos.

Módulo: 8.3.2

Função: Formaliza Registro

Os registros do arquivo CAM016 são formalizados para posterior impressão.

Módulo: 8.3.3

Função: Imprime Boletim Escolar

Os conceitos dos alunos, são impressos sobre a forma de Boletim Escolar, e, encaminhados a cada aluno.

CAPÍTULO 6

CONCLUSÕES

É estimado para um futuro próximo que as Universidades brasileiras tenham implantado em Computador Sistemas de: Contrôles Acadêmicos, Controle de Biblioteca, Controle de Material, Controle Contábil e Controle de Pessoal.

Certamente, o problema não será solucionado com este trabalho, mas, provavelmente a metodologia aqui desenvolvida irá ajudar aos projetistas de Software no desenvolvimento dos sistemas vigentes.

Ao que se refere à implantação do Sistema de Controle Acadêmico desenvolvido neste trabalho é aconselhável a escolha de uma linguagem para geração e acesso aos arquivos, e, a escolha de uma linguagem de alto nível que suporte esta linguagem.

A P E N D I C E

1. ARQUIVO CADASTRO DE ALUNOS - CACAØ1

A título de exemplo de um arquivo com estrutura de Lista Invertida, apresenta-se o arquivo CACAØ1 (Arquivo Cadastro de Alunos) que se constitui de várias informações redundantes entre si. Com a intenção de se reduzir tais redundâncias, utilizou-se um arquivo com estrutura de Lista Invertida, onde, os campos redundantes são as chaves: código de curso, código de aluno, natureza do ingresso, código de disciplina e tipo de sangue. E, os apontadores, indicam os endereços dos registros contendo as demais informações não redundantes, constantes do arquivo CACAØ1 mostrado na Figura A.1

As estruturas apresentadas nas Figura A.2, A.3, A.4 e A.5, representam as estruturas de listas invertidas, onde, as redundâncias foram evitadas.

Em tal estrutura, pode-se observar que:

o aluno JOSEBIAS, faz o curso de Zootecnia, ingressou na Universidade mediante exame de vestibular, cursou as disciplinas Iniciação a Ciência da Computação e Cálculo Numérico. Tem sangue tipo B. Para melhor compreensão da estrutura, observa-se que: o referido aluno tem endereço 8 na Figura A.1 e nas Figuras A.2, A.3, A.4 e A.5, existem apontadores para o endereço 8.

ENDEREÇO	INSCRIÇÃO	NOME	DATA NASCIMENTO	C.P.F.	ENDEREÇO	TELEFONE
1	7011 0289	EDUARDO J.	20 06 40	011305103	OTACILIO ALBUQUERQUE	8144
2	7011 030X	EMERSON P.	15 04 38	011407104	JOÃO PESSOA	4044
3	7011 0319	EMERSON L.	05 03 48	0115 07103	JOÃO SUASSUNA	2333
4	7011 036X	FRANCISCO A.	02 07 50	0114 13189	FREI CANECA	3543
5	7011 0572	JOSÉ A.	10 08 52	011334431	AV. GETÚLIO VARGAS	—
6	7011 0918	JOSE A. N.	30 07 53	011441545	JOÃO ANTONIO	—
7	7011 0742	LUCEVAL L.	11 01 51	111144345	PADRE VALDEVINO	8081
8	7011 0718	JOSEBIAS N.	11 09 49	131321450	PADRE ALEIXO	—

FIGURA A.1 - PARTE DO ARQUIVO CACAØ1

Chave: Código de Curso

- 04 Economia
- 22 Engenharia Civil
- 23 Engenharia Elétrica
- 47 Cooperativismo
- 49 Zootecnia
- 60 Bacharelado em Computação.

CHAVES	APONTADORES						
0 4	5						
2 2	6	4					
2 3	1						
4 7	2						
4 9	3	8					
6 0	8						

FIGURA A.2 - ESTRUTURA DO CÓDIGO DE CURSO

Chave: Natureza do Ingresso

- 1 Vestibular
- 2 Transferido
- 3 Reinício do Curso
- 4 Graduado

CHAVES	APONTADORES						
1	1	8	2	6			
2	3	5					
3	4						
4	7						

FIGURA A.3 - ESTRUTURA DA NATUREZA DO INGRESSO

Chave: Código de Disciplina

- P080011 - 0 Seminário de Engenharia
- P080024 - 2 Máquinas Hidráulicas
- P080133 - 8 Iniciação à Ciência da Computação
- P080153 - 2 Cálculo Numérico
- P080184 - 2 Processamento de Dados

CHAVE	APONTADORES					
PO80011-0	1	7	2	6		
PO80024-2	8	4	6	1		
PO80133-8	3	2	7	8		
PO80153-2	8	5				
PO80184-2	5	6				

A.4 - ESTRUTURA DE CÓDIGO DE DISCIPLINA

Chave: Tipo de Sangue

- 1 A
- 2 B
- 3 O
- 4 AB

CHAVE	APONTADORES					
1	8	3	4			
2	1	6				
3	5	2				
4	7					

A.5 - ESTRUTURA DO TIPO DE SANGUE

2. CÓDIGOS DE DISCIPLINAS

Para efeito de orientação, as disciplinas são identificadas por códigos assim estruturados:

1º Caracter:

Indica o Centro em que a disciplina está ligada.

2º e 3º Caracteres

Identifica o Departamento dentro do Centro.

4º, 5º e 6º Caracteres

Representa o Nº de ordem da disciplina dentro do Departamento.

7º Caracter

Indica o número de créditos da disciplina

8º Caracter

Dígito verificador

A título de exemplo apresenta-se o código H030025-0 que significa:

H - Centro de Ciências Humanas, Letras, Artes.

03 - Departamento de Letras Clássicas e Vernáculas, terceiro do Centro de Ciências Humanas, Letras e Artes.

002 - Segunda disciplina do Departamento

5 - Número de créditos

0 - Dígito verificador

3. INSCRIÇÃO DO ALUNO

A inscrição do aluno está assim estruturada:

1º e 2º Caracteres

Indica o ano em que o aluno ingressou na Universidade.

3º Caracter

Indica o período dentro do ano em que o aluno ingressou na Universidade.

4º Caracter

Indica a área a que o aluno pertence.

5º, 6º e 7º Caracteres

Indica o número de ordem do aluno dentro da área e do período.

8º Caracter

Dígito verificador

Exemplo: 7011028 9

70 - Ano de ingresso do aluno

1 - Indica que o aluno ingressou no pri
meiro período de 1970.

1 - Indica que o aluno pertence à área 1.

028 - O aluno é o vigésimo oitavo daquela
área e daquele período.

9 - Dígito verificador

4. NOMENCLATURA UTILIZADA NO SISTEMA

A nomenclatura utilizada para definir os diversos tipos de arquivos, envolve uma codificação alfanu
mérica de 6 caracteres, significando o seguinte:

1º e 2º Caracteres

Identifica o sistema. No caso do Sistema Controle Acadêmico, todos os códigos possuirão o prefixo CA.

3º e 4º Caracteres

Identifica o tipo de arquivo a saber:

- CA - Cadastro
- MO - Movimento
- TA - Tabela

5º e 6º Caracteres

Indica o número de ordem do arquivo dentro de cada tipo.

Como exemplo, apresenta-se o arquivo denominado: CAM002, onde,

- CA - Indica que pertence ao Sistema Controle Acadêmico
- MO - Indica que é um arquivo movimento.
- 02 - Indica que é o segundo arquivo movimento.

5. ARQUIVOS ENVOLVIDOS NO SISTEMA CONTROLE ACADÊMICO

Apresenta-se nesta parte do trabalho uma visão de todos arquivos envolvidos no Sistema Controle Acadêmico, constando de: nomenclaturas, descrição e os subsistemas que os utilizam, como mostra a Figura A.6.

NOMENCLATURA	D E S C R I Ç Ã O	SUBSSISTEMAS QUE O UTILIZAM
CACAO1	Cadastro de aluno	Cadastro de alunos, Matrículas, Currículos de Cursos, Processamento de Notas
CACAO2	Cadastro Geral de Disciplinas	Cadastro Geral de Disciplinas, Estudo de Demanda de Vagas e Cadastro Semestral Disciplinas.
CACAO3	Cadastro Semestral de Disciplinas	Cadastro Semestral de Disciplinas e Matrícula
CACAO4	Cadastro de Professores	Cadastro de Professores, Matrícula, Cadastro Semestral Disciplinas.
CACAO5	Cadastro de Currículos de Cursos	Currículo de Cursos e Cadastro de alunos.
CAMO01	Arquivo movimento contendo dados cadastrais dos alunos	Cadastro de alunos
CAMO02	Contém dados cadastrais dos currículos de cursos.	Currículo de Cursos
CAMO03	Contém dados de levantamento junto aos alunos.	Estudo de Demanda de Vagas
CAMO04	Contém códigos de duas disciplinas	Estudo de Demanda de Vagas.
CAMO05	Contém códigos de tres disciplinas	Estudo de Demanda de Vagas
CAMO06	Contém códigos de 4 disciplinas	Estudo de Demanda de Vagas
CAMO07	Contém dados de solicitação de matrícula.	Matrícula
CAMO08	Contém dados de solicitação de matrícula.	Matrícula
CAMO09	Contém disciplinas solicitadas los alunos.	Matrícula
CAMO10	Contém disciplinas solicitadas los alunos, ordenado.	Matrícula
CAMO11	Contém solicitações de cancelamento e substituição.	Matrícula
CAMO12	Contém disciplinas solicitadas los alunos	Matrícula

NOMENCLATURA	DESCRITÃO	SUBSISTEMAS QUE O UTILIZAM
CAM013	Contém dados, solicitação de cancelamento e substituição.	Matrícula
CAM014	Contém disciplinas solicitadas pelos alunos ordenado	Matrícula e Processamento de Notas.
CAM015	Contém notas dos alunos	Processamento de notas
CAM016	Contém resultados dos alunos nas disciplinas cursadas.	Processamento de notas.
CAM017	Contém informações cadastrais das disciplinas	Cadastro Geral de Disciplinas, Cadastro Semestral de Disciplinas
CAM018	Contém dados de levantamento junto aos alunos.	Estudo de Demanda de Vagas
CAM019	Contém dados Cadastrais dos Professores.	Cadastro de Professores.
CATA01	Contém valores acumulados do levantamento junto aos alunos.	Estudo de Demanda de Vagas.
CATA02	Contém valores da operação de solicitações p/combinções de 2 disciplinas	Estudo de Demanda de Vagas
CATA03	Contém valores da operação de solicitações p/combinções de 3 disciplinas.	Estudo de Demanda de Vagas.
CATA04	Contém valores de operação de solicitações p/combinções de 4 disciplinas	Estudo de Demanda de Vagas.

FIGURA A.6 - ARQUIVOS DO SISTEMA CONTROLE ACADÊMICO

6. DESCRIÇÃO DOS CAMPOS DOS ARQUIVOS

CACAØ1 - Arquivo Cadastro de Alunos, em disco, com estrutura de lista invertida, tendo como chaves: código de curso, natureza do ingresso, código de disciplina, tipo de sangue e inscrição do aluno.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Inscrição do aluno	8
Nível	2
Natureza do ingresso	1
Sexo	1
Estado civil	1
Tipo de sangue	1
Fator RH	1
C.E.P.	5
Data de nascimento	6
Nome	40
Código de operação	1
Nome do pai	32
Nome da mãe	32
Documento militar	9
Orgão expedidor	5
C.P.F.	11
Nº da identidade	8
Orgão expedidor	4
U.F.	2
Data expedição	6
Cert. alis. militar	1

<u>Campos</u>	<u>Posições</u> (bytes)
Cert. dispensa corporação	1
Cert. reservista 1. ^a categoria	1
Cert. reservista 2. ^a categoria	1
Atestado desobrigação militar	1
Atestado situação militar	1
Cert. de isenção militar	1
Certidão de situação militar	1
Carta patente	1
Provisão de reforma	1
Militar da ativa	1
Endereço local	30
Número	5
Telefone	8
C.E.P.	5
Endereço permanente	30
Número	5
Telefone	8
Município	21
U.F.	2
Código disciplina	8
N vezes	
Conceito	3

NOTA:

N, significa o número de disciplinas previsto para que um aluno conclua um determinado curso.

CACAD2 - Arquivo Cadastro Geral de Disciplinas, em disco, com organização sequencial indexada, tendo como

chave: código de disciplina

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Código disciplina	8
Nome disciplina	30
Carga horária teórica	3
Carga horária prática	3
Pré-requisito 1	8
Pré-requisito 2	8
Pré-requisito 3	8
Pré-requisito 4	8
Posição p/marca	1

CACA03 - Arquivo Cadastro Semestral de Disciplinas, em disco, com organização sequencial indexada, tendo como chave: código de disciplina

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Código disciplina	8
turma	2
Nome disciplina	30
Número sala	18
Vagas oferecidas	2

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Vagas remanescentes	2
Carga horária teórica	3
Carga horária prática	3
Nome do professor	11
Pré-requisito 1	8
Pré-requisito 2	8
Pré-requisito 3	8
Pré-requisito 4	8
Horário	24

CACAØ4 - Arquivo Cadastro de Professores, em disco, com organização sequencial indexada, tendo como chave: código do professor

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Código do professor	8
Nome do professor	40
Nome popular	11
Matrícula na universidade	6
Carga horária de contrato	2
Horas destinadas as aulas	2
Horas destinadas à administração	2
Área que ensina	2
Código disciplina que ensina (4)	32
Disciplinas ensinadas (10 apontadores)	20

CACA05 - Arquivo Cadastro Currículo de Cursos, em disco, organização sequencial indexada, tendo como chave: código de curso/período.

<u>Campos</u>	<u>Posições</u> (bytes)
Curso/período	11
Código disciplina	8
Pré-requisitos	32*
"Flag"	1
Importância curricular da disciplina	1

NOTA:

1. "Flag" - para indicar fim dos pré-requisitos.
2. Importância curricular da disciplina, constituindo-se de:

<u>Código</u>	<u>Descrição</u>
1	Currículo mínimo
2	Currículo complementar obrigatório
3	Currículo complementar optativo

CACA06 - Arquivo Cadastro Disciplina/Aluno em disco, organização sequencial indexada, tendo como chave: Disciplina/Aluno.

NOTA:

Campos idênticos ao arquivo CAM014.

* Constando no máximo de quatro disciplinas.

CAM001 - Arquivo Movimento que contém dados cadastrais do aluno, em disco, organização sequencial indexada tendo como chave: inscrição de aluno.

<u>Campos</u>	<u>Posições (bytes)</u>
Inscrição de aluno	8
Nível	2
Natureza do ingresso	1
Sexo	1
Estado civil	1
Tipo de sangue	1
Fator RH	1
C.E.P.	5
Data nascimento	6
Nome	40
Código operação	1
Nome do pai	32
Nome da mãe	32
Doc. militar	9
Orgão expedidor	5
C.P.F.	11
Nº da identidade	4
Orgão expedidor	2
U.F.	2
Data expedição	6
Cert. alistamento militar	1
Cert. dispensa corporação	1
Cert. reservista 1ª categoria	1
Cert. reservista 2ª categoria	1
Atestado desobrigação militar	1
Atestado situação militar	1

<u>Campos</u>	<u>Posições</u> (bytes)
Certificado de isenção militar	1
Certidão de situação militar	1
Carta patente	1
Provisão de reforma	1
Militar da ativa	1
Endereço local	30
Número	5
Telefone	8
C.E.P.	5
Endereço permanente	30
Número	5
Telefone	8
Município	21
U.F.	2

CAM002 - Arquivo Movimento que contém os dados cadastrais dos currículos de cursos, em disco, organização sequencial indexada tendo como chave: curso/período.

<u>Campos</u>	<u>Posições</u> (bytes)
Curso/período	11
Código disciplina	8
Pré-requisitos	32*
Importancia curricular	1
Código operação	1

* Constando no máximo de quatro disciplinas

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
---------------	-----------------------------------

Código do aluno	8
-----------------	---

CAM003 - Arquivo Movimento que contém dados de levantamento de solicitação de disciplinas junto aos alunos, em disco, organização sequencial indexada, tendo como chave: inscrição do aluno.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
---------------	-----------------------------------

Inscrição do aluno	8
Código curso	8
Prioridade	1
12 vezes Código disciplina	8
turno	1

CAM004 - Arquivo Movimento que contém códigos de duas disciplinas, em disco, organização sequencial indexada, tendo como chave código disciplina1 / código disciplina2.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
---------------	-----------------------------------

Código disciplina 1	8
Código disciplina 2	8

CAM005 - Arquivo Movimento que contém códigos de tres disciplinas, em disco, organização sequencial indexada, tendo como chave: código disciplina1 / código disciplina2/código disciplina3.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Código disciplina 1	8
Código disciplina 2	8
Código disciplina 3	8

CAM006 - Arquivo Movimento que contém códigos de quatro disciplinas, em disco, organização sequencial indexada, tendo como chave: código disciplina1 / código disciplina2/código disciplina3/código disciplina4.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Código disciplina 1	8
Código disciplina 2	8
Código disciplina 3	8
Código disciplina 4	8

CAM007 - Arquivo Movimento que contém dados de Solicitação de Matrícula em disco, organização sequencial indexada, tendo como chave: inscrição do aluno.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Inscrição do aluno	8
Curso	2
Prioridade I	8
Turma I	1
Alternativa I	2
Prioridade II	8
Turma II	1
Alternativa II	2
Prioridade III	8
Turma III	1
Alternativa III	2
Prioridade IV	8
Turma IV	1
Alternativa IV	2
Prioridade V	8
Turma V	1
Alternativa V	2
Prioridade VI	8
Turma VI	1
Alternativa VI	2
Período	3
Prioridade VII	8
Turma VII	1
Alternativa VII	2
Prioridade VIII	8
Turma VIII	1
Alternativa VIII	2
Prioridade IX	8
Turma IX	1
Alternativa IX	2
Cod-alternativa I	8
Turma I	1
Cod-alternativa II	8

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Turma II	1
Cod-alternativa III	8
Turma III	1
Período	3

CAM008 - Arquivo Movimento que contém dados de solicitação de matrícula, em disco, organização sequencial indexada, tendo como chave: inscrição do aluno.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Inscrição do aluno	8
Curso	2
Prioridade I	8
Turma I	1
Alternativa I	2
Prioridade II	8
Turma II	1
Alternativa II	2
Prioridade III	8
Turma III	1
Alternativa III	2
Prioridade IV	8
Turma IV	1
Alternativa IV	2
Prioridade V	8
Turma V	1
Alternativa V	2
Prioridade VI	8

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Turma VI	1
Alternativa VI	2
Prioridade VII	8
Turma VII	1
Alternativa VII	2
Prioridade VIII	8
Turma VIII	1
Alternativa VIII	2
Prioridade IX	8
Turma IX	1
Alternativa IX	2
Cod-alternativa I	8
Turma I	1
Cod-alternativa II	8
Turma II	1
Cod-alternativa	8
Turma III	1
Período	3

CAM009 - Arquivo Movimento que contém disciplinas solicitadas pelos alunos, em disco, organização sequencial indexada, tendo como chave: inscrição do aluno.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Inscrição do aluno	8
Curso	2
Prioridade I	8
Turma I	1

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Alternativa I	2
Prioridade II	8
Turma II	1
Alternativa II	2
Prioridade III	8
Turma III	1
Alternativa III	2
Prioridade IV	8
Turma IV	1
Alternativa IV	2
Prioridade V	8
Turma V	1
Alternativa V	2
Prioridade VI	8
Turma VI	1
Alternativa VI	2
Prioridade VII	8
Turma VII	1
Alternativa VII	2
Prioridade VIII	8
Turma VIII	1
Alternativa VIII	2
Prioridade IX	8
Turma IX	1
Alternativa	2
Cod-alternativa I	8
Turma I	1
Cod-alternativa II	8
Turma II	1
Cod-alternativa III	8
Turma III	1
Período	3
Situação	1

CAM010 - Arquivo Movimento que contém ocorrências de matrícula, em disco, organização sequencial indexada, tendo como chave: inscrição do aluno.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Inscrição do aluno	8
Curso	2
Prioridade I	8
Turma I	2
Alternativa I	1
Situação I	1
Prioridade II	8
Turma II	1
Alternativa II	2
Situação II	1
Prioridade III	8
Turma III	1
Alternativa III	2
Situação III	1
Prioridade IV	8
Turma IV	1
Alternativa IV	2
Situação IV	1
Prioridade V	8
Turma V	1
Alternativa V	2
Situação V	1
Prioridade VI	8
Turma VI	1
Alternativa VI	2
Situação VI	1
Prioridade VII	8

<u>Campos</u>	<u>Posições</u> (bytes)
Turma VII	1
Alternativa VII	2
Situação VII	1
Prioridade VIII	8
Turma VIII	1
Alternativa VIII	2
Situação VIII	1
Prioridade IX	8
Turma IX	1
Alternativa IX	2
Situação IX	1

CAM011 - Arquivo Movimento que contém solicitações de cancelamentos e substituições de disciplinas, em disco, organização sequencial indexada, tendo como chave: inscrição do aluno.

<u>Campos</u>	<u>Posições</u> (bytes)
Inscrição do aluno	8
Código curso	2
Adicionamento/cancelamento I	8
Turma I	2
Prioridade I	1
Adicionamento/cancelamento II	8
Turma II	2
Prioridade II	1
Troca I	8
Turma III	2
Desejada I	8

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Turma IV	2
Prioridade III	1
Troca II	8
Turma V	2
Desejada II	8
Turma VI	2
Prioridade IV	1

CAM012 - Arquivo Movimento que contém disciplinas solicita das pelos alunos, em disco, organização sequencial indexada, tendo como chave: insc. do aluno.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Inscrição do aluno	8
Curso	2
Prioridade I	8
Turma I	2
Alternativa I	1
Situação I	1
Prioridade II	8
Turma II	2
Alternativa II	1
Situação II	1
Prioridade III	8
Turma III	2
Alternativa III	1
Situação III	1
Prioridade IV	8
Turma IV	2

<u>Campos</u>	<u>Posições</u> (bytes)
Turma IV	2
Prioridade III	1
Troca II	8
Turma V	2
Desejada II	8
Turma VI	2
Prioridade IV	1

CAM012 - Arquivo Movimento que contém disciplinas solicita das pelos alunos, em disco, organização sequencial indexada, tendo como chave: insc. do aluno.

<u>Campos</u>	<u>Posições</u> (bytes)
Inscrição do aluno	8
Curso	2
Prioridade I	8
Turma I	2
Alternativa I	1
Situação I	1
Prioridade II	8
Turma II	2
Alternativa II	1
Situação II	1
Prioridade III	8
Turma III	2
Alternativa III	1
Situação III	1
Prioridade IV	8
Turma IV	2

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Alternativa IV	1
Situação IV	1
Prioridade V	8
Turma V	2
Alternativa V	1
Situação V	1
Prioridade VI	8
Turma IV	2
Alternativa VI	1
Situação VI	1
Prioridade VII	8
Turma VII	2
Alternativa VII	1
Situação VII	1
Prioridade VIII	8
Turma VIII	2
Alternativa VIII	1
Situação VIII	1
Prioridade IX	8
Turma IX	2
Alternativa IX	1
Situação IX	1

CAM013 - Arquivo Movimento, que contém dados de substituição e cancelamento de disciplinas, em disco, organização sequencial indexada, tendo como chave: inscrição do aluno.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Inscrição do aluno	8
Código curso	2
Adicionamento/cancelamento I	8
Turma I	2
Prioridade I	1
Adicionamento/cancelamento II	8
Turma II	2
Prioridade II	1
Troca I	8
Turma III	2
Desejada I	8
Turma IV	2
Prioridade III	1
Troca II	8
Turma V	2
Desejada II	8
Turma IV	2
Prioridade IV	1

CAMO14 - Arquivo Movimento que contém disciplinas solicita das pelos alunos, ordenado por inscrição de alunos, em disco, com organização sequencial indexada, tendo como chave: código de disciplina / inscrição do aluno.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Códigos disciplina/insc. do aluno	16
Turma	2

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Código de curso	2
Conceito	3

CAM015 - Arquivo Movimento que contém as notas dos alunos, originárias do "Diário de Classe", em disco, organização sequencial indexada, tendo como chave: código disciplina/insc. do aluno.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Código disciplina/código do aluno	16
Total de Faltas	2
Nota exame final	3
Nota exercício de casa	3
Nota laboratório	3
Nota projeto	3

CAM016 - Arquivo Movimento que contém os conceitos dos alunos após processamento das notas, em disco, organização sequencial indexada, tendo como chave: código de curso/código de disciplina.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Código de curso/código de disciplina	10

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Inscrição de alunos	8
Turma	2
Conceito	3

CAM017 - Arquivo Movimento que contém informações cadastrais das disciplinas, em disco, com organização sequencial indexada, tendo como chave: o código de disciplina.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Código de disciplina	8
Pré-requisito 1	8
Pré-requisito 2	8
Pré-requisito 3	8
Pré-requisito 4	8
Nome disciplina	30
Carga horária teórica	3
Carga horária prática	3
Código operação	1
Código do professor	8

CAM018 - Arquivo Movimento que contém dados de levantamento feito junto aos alunos para o Estudo de Demanda de Vagas, em disco, com organização sequencial indexada, tendo como chave: insc. do aluno.

<u>Campos</u>	<u>Posições</u> (<u>byter</u>)
Inscrição do aluno	8
Código curso	8
Prioridade	1
12 vezes Código de disciplinas	8
Turno	1

CAM019 - Arquivo Movimento que contém dados cadastrais dos professores, em disco, organização sequencial in dexada, tendo como chave: código do professor.

<u>Campos</u>	<u>Posições</u> (<u>byter</u>)
Código professor	8
Nome professor	30
Nome popular	11
Matricula universidade	6
Carga horária contrato	2
Horas destinadas às aulas	2
Horas destinadas à administração	2
Área que ensina	2
Disciplina 1 que ensina	8
Disciplina 2 que ensina	8
Disciplina 3 que ensina	8
Disciplina 4 que ensina	8

CATA01 - Arquivo tabela que contém valores acumulados do levantamento feito junto aos alunos para o Estudo de Demanda de Vagas, em disco tendo como chave: códi go de disciplina.

<u>Campos</u>	<u>Posições</u> (bytes)
Código disciplinas	8
Nome disciplinas	30
Código curso	2
Manhã	3
15 vezes Tarde	3
Manhã extrapolada	3
Tarde extrapolada	3
Total alunos participaram E.D.V.	3

CATA02 - Arquivo tabela que contém os valores da apuração de solicitações para combinações de duas disciplinas, em disco tendo como chave: código disciplina1código disciplina2.

<u>Campos</u>	<u>Posições</u> (bytes)
Código Disciplina 1	8
Código Disciplina 2	8
Código do curso	2
15 vezes Manhã	3
Tarde	3

CATA03 - Arquivo tabela que contém os valores da apuração de solicitações de disciplinas para combinações de tres disciplinas, em disco, tendo como chave: código disciplina1códigodisciplina2códigodisciplina2códigodisciplina3.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Código disciplina 1	8
Código disciplina 2	8
Código disciplina 3	8
código de curso	2
15 vezes manhã	3
tarde	3

CATAØ4 - Arquivo tabela que contém valores da apuração de solicitações para combinações de quatro disciplinas, em disco, tendo como chave: código disciplina1código disciplina2código disciplina3código disciplina4.

<u>Campos</u>	<u>Posições</u> <u>(bytes)</u>
Código disciplina 1	8
Código disciplina 2	8
Código disciplina 3	8
Código disciplina 4	8
Código curso	2
15 vezes Manhã	3
Tarde	3

B I B L I O G R A F I A

01. TSICHRITZIS, Dyonysios and BERNSTEIN, Philip A. - Computer Science and Applied Mathematics. New Jersey. 1974
02. YOURDON, Edward - Techniques of Program Structure and Design. New Jersey, 1975.
03. MYERS, Glenford J. - Reliable Software Through Composite Design. New York, 1975.
04. BAUER, F. L. and DENNIS, J. B. GOOS, G. TSICHRITZIS, D - Software Engineering. New York, 1975.
05. POLLACK, S. V. and STERLING, T. D. - A Guide To PL/I. New York. 1969.
06. KATZAN, Harry Jr. - Systems Design and Documentation. New York, 1976.
07. STEYENS, W. P. MYERS, G. J. CONSTANTINE, L. L - Structured Design. IBM Systems Journal, May 1974.
08. PARNAS, D. L. - A Technique for Software Modulo Specification With Examples. Carnegie - Mellon University. ACM, May, 1972.

09. TSICHRITZIS, Dionysios C., BERNSTEIN, Philip A. - Operating Systems. New Jersey, 1975.
10. BOEHM, Barry W. - Software Engineering IEEE Transactions on Computers, Vol C - 25 N° 12, 1976.
11. TSICHRITZIS, Dionysios C. - An Advanced Course. New Jersey, 1975.
12. HAMILTON, Margaret and ZELDIN, Saydean - Higher Order Software. IEEE Transactions on Software Engineering, Vol. SE - 2, N° 1, March 1976.
13. THIERAUF, Robert J - Systems Analysis and Design of Real-Time Management Information Systems. New Jersey, 1975.
14. HARTMAN, W, MATTHES, H and PROEME, A - Management Information Systems Handbook. New York, 1972.
15. FITZGERALD, John M., FITZGERALD, Ardra F. - Fundamentals of Systems Analysis, Toronto, 1973.
16. DIJKSTRA, Edsger W. - The Structure of the "THE" - Multi programming System. Communications of the ACM. Volume 11/Number 5/May, 1968.

17. PARNAS, D. L. - On the Criteria To Be Used in Decomposing Systems into Modules. Communications of the ACM, Volume 15/Number 12/December 1972.

18. BOEHM, Barry W. - Software Engineering. IEEE Transactions on Computers, Vol. C - 25, Nº 12, December 1976.



UNIVERSIDADE FEDERAL DA PARAIBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprigio Veloso, 882 - Tel. (083) 321-7222-R 355
58.100 - Campina Grande - Paraíba