



I.C.M.S.C.

INSTITUTO DE CIÊNCIAS MATEMÁTICAS DE SÃO CARLOS

UMA EXPERIÊNCIA DE DESENVOLVIMENTO DE
SISTEMAS DE INFORMAÇÃO USANDO INSTRU
MENTAL DE 4^a GERAÇÃO

ELISA HATSUE MORIYA

UNIVERSIDADE DE SÃO PAULO

SÃO CARLOS - SÃO PAULO
BRASIL

UMA EXPERIÊNCIA DE DESENVOLVIMENTO DE
SISTEMAS DE INFORMAÇÃO USANDO INSTRU
MENTAL DE 4.^a GERAÇÃO

ELISA HATSUE MORIYA

Orientador:

Prof.Dr. Fernão Stella de Rodrigues Germano

Dissertação apresentada ao Instituto de
Ciências Matemáticas de São Carlos, da
Universidade de São Paulo, para obten
ção do título de Mestre em Ciências de
Computação.

- São Carlos -

1986

Dedico este trabalho

*A meus pais Yasuhide e Takako
que nunca pouparam esforços pa
ra que eu tivesse uma boa forma
ção.*

*Ao meu irmão Massatoyo que sem
pre colaborou no melhor de suas
possibilidades com todas as mi
nhas iniciativas.*

AGRADECIMENTOS

Ao Prof. Dr. Fernão S.R. Germano pela orientação e disponibilidade com que sempre me distinguiu, proporcionando o meu desenvolvimento científico além dos laços de amizade entre nós estabelecidos que acompanharão a jornada aqui iniciada.

Ao Prof. Dr. Paulo Cesar Masiero pelas valiosas sugestões dadas desde a definição e durante toda a elaboração deste trabalho. Agradeço também pelo esforço na obtenção de material bibliográfico recente e pela interação estabelecida apesar da distância e inúmeros afazeres de seu programa de Pós -Doutoramento que muito enriqueceram este trabalho.

Ao Prof. Dr. Sergio L. de Oliveira Assis, da Faculdade de Economia e Administração - USP, pela oportunidade de estágio a mim concedida e pelos contactos que me proporcionou com representantes do software utilizado neste trabalho.

Aos Profs. Rosangela Ap. Delloso Penteado e Marcos José Santana pelas valiosas sugestões dadas

Aos amigos, professores e funcionários do ICMSC pela acolhida, em particular aos Profs. Dr. Odelar Leite Linhares e Dr. Maximilian Emil Hehl.

À PROCONSULT pela oportunidade de utilização dos recursos disponíveis, com os quais pôde ser desenvolvida parte ponderável deste trabalho e em especial aos companheiros de trabalho: Sílvia Araújo de Oliveira Assis, Marta Borba, Mário Kuwaha, Armando Trivelato Filho e Fernando Ablas.

À TECOM SISTEMAS, nas pessoas de Wladyr Antonio Furegatti e Paulo Roberto Suzuque pelo apoio técnico, quando do contacto com o instrumental objeto deste trabalho.

Aos colegas do Departamento de Informática, da Fundação Universidade Estadual de Maringá, em especial aos Profs. Álvaro José Periotto, Divair Maria Terna Gomes, Jorge Yochiro Kakitani e Wilson José Bosso pela acolhida e incentivos dados.

Aos amigos Yoko Terada, Ivo Machado da Costa, Chiguero Okada e Maria Kimiyo Okada pelos constantes incentivos dados.

Às amigas Silvia Maria Levorato e Maria Cecília Ve
chiato Saenz Carneiro, companheiras de estudos.

Às amizades que nasceram nesta caminhada e que con
tinuarão apesar das distâncias que nos separam.

Ao CNPq, CAPES, FAPESP, FINEP que direta ou indire
tamente ofereceram incentivos financeiros.

AN EXPERIMENT IN INFORMATION SYSTEM DEVELOPMENT USING 4th
GENERATION TOOLS

ABSTRACT

IDMS (Integrated Data Management System) components have been used to develop a system for controlling capital quotas of a shopping mall. The system controls capital returns, ownership transferences, bank credits, monthly distributions, etc. ADS/O (Application Development System/on line) has been the component that served as the basic tool for the implementation. OLQ (On Line Query), OLE (On Line English), IDD (Integrated Data Dictionary) and CULPRIT are also mentioned in the dissertation. SADT (Structured Analysis and Design Technique) is used to describe the system in detail. Actigrams and Datagrams are combined with structure diagrams a la Constantine to exemplify the modules that are part of the system. The system has a total of 50 modules for the processing on line and about 8 COBOL programs for the batch part of the processing. The ADS/O code for about 10 of these modules is presented. In the appendix a summary of ADS/O features is presented together with information on OLQ resources. An evaluation of the development is included. It refers to how effective has been the use of the system development methodologies employed as well as of the data base management resources and of the query language facilities.

UMA EXPERIÊNCIA DE DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO
USANDO INSTRUMENTAL DE 4.^a GERAÇÃO

RESUMO

Componentes do IDMS (Integrated Data Management System) foram usados no desenvolvimento de um sistema de controle de cotas de um shopping center. O sistema controla rendimento das cotas, transferência de títulos, créditos bancários, renda mensal, etc. O ADS/O (Application Development System/on line) ; foi a componente que serviu de instrumento básico para a implementação. As componentes OLQ (On Line Query), OLE (One Line English), IDD (Integrated Data Dictionary), CULPRIT são também mencionados na dissertação. O SADT (Structured Analysis and Design Technique) é usado para descrever o sistema em detalhe. Os atvigramas e datagramas são combinados com os diagramas de estrutura de Constantine para exemplificar os módulos que constituem o sistema. O sistema tem um total de 50 módulos para o processamento imediato e cerca de 8 programas COBOL para o processamento por lotes. A codificação de cerca de 10 desses módulos em ADS/O é apresentada. No apêndice figura um resumo das características do ADS/O juntamente com informações sobre os recursos da OLQ. É incluída uma avaliação do desenvolvimento efetuado. Ela se refere à efetividade do uso das metodologias de desenvolvimento de sistemas empregadas bem como dos recursos de gestão de base de dados e das características da linguagem de consulta.

ÍNDICE

1. INTRODUÇÃO

1.1 - Considerações Gerais	001
1.2 - Assunto da Dissertação	003
1.3 - Apresentação do Trabalho	004

2. RESENHA BIBLIOGRÁFICA

2.1 - Considerações Gerais	005
2.2 - Referências sobre Instrumental de 4. ^a Geração e Conceito de Centro de Informação	007
2.3 - Referências sobre Exemplos de Metodologias de Análi se e Projeto que poderiam ser Acopladas a esse Instrumental	016
2.4 - Conclusões	021

3. UM EXEMPLO DE INSTRUMENTAL DE 4.^a GERAÇÃO

3.1 - Considerações Gerais	022
3.2 - O Dicionário de Dados - IDD	026
3.3 - A Ferramenta para Desenvolvimento - ADS/O	028
3.4 - A Linguagem de Consulta - OLQ	030
3.5 - Outros Recursos: OLE, CULPRIT	032
3.6 - Conclusões	035

4. UM EXEMPLO DE APLICAÇÃO DESSE INSTRUMENTAL

4.1 - Considerações Gerais	037
4.2 - O Fluxo de Informações do Sistema	038
4.3 - Componentes de Processamento por Lotes e Imediato	040
4.4 - Exemplos de Detalhamento de Componentes do Proces samento Imediato	046
4.5 - Conclusões	059

5. AVALIAÇÃO DO DESENVOLVIMENTO EFETUADO

5.1 - Considerações Gerais	060
5.2 - Sobre o Aproveitamento do Potencial dos Métodos de Análise e Projeto de Sistemas	061
5.3 - Sobre o Aproveitamento dos Recursos de Base de Dados	075
5.4 - Sobre o Aproveitamento dos Recursos de Linguagem de Consulta	076
5.5 - Conclusões	079

6. CONCLUSÕES

6.1 - Conclusões Finais	081
6.2 - Contribuição desse Trabalho	083
6.3 - Sugestões para Novas Pesquisas	084

BIBLIOGRAFIA	086
--------------------	-----

APÊNDICE I	091
------------------	-----

APÊNDICE II	134
-------------------	-----

CAPÍTULO I

INTRODUÇÃO

1.1 - Considerações Gerais

Os profissionais da área de Processamento de Dados, têm assistido a uma sensível mudança no comportamento do usuário frente ao desenvolvimento de sistemas. Essa mudança se reflete em uma participação cada vez mais ativa, quer nas fases iniciais de análise e projeto, quer naquelas posteriores à implementação.

Para atender a esse usuário cada vez mais ativo, tem sido tornadas disponíveis no mercado, ferramentas mais sofisticadas, que apoiam tal participação, denominadas de 4.^a geração.

Segundo Martin, [Ma81], uma linguagem de 4.^a geração tem como uma das principais características, permitir ao analista obter resultados mais rápidos do que ao especificar programas. O analista trabalha junto ao usuário interativamente, até que satisfaça as exigências dele. Ainda segundo Martin, existem linguagens de 4.^a geração voltadas ao usuário, que permitiriam a ele obter por si mesmo, os resultados desejados desde que os dados tenham sido devidamente estruturados.

A presença de tais ferramentas tem permitido além de uma interação maior do usuário com o analista, contribuindo assim para a obtenção em menor prazo de um sistema que realmente atenda às necessidades da empresa, a liberação da equipe de desenvolvimento para que se dedique a sistemas mais complexos. O usuário participando mais intensamente do desenvolvimento, pode por um lado passar a ter condições de melhor definir as especificações e por outro entender melhor os problemas que surgem durante o desenvolvimento e não exercer tanta pressão de tempo sobre a equipe. Em algumas empresas, dependendo dos "softwares" disponíveis e áreas que possam utilizá-los, pode-se eventualmente pensar na implantação de um Centro de Informação. Esta seria também uma forma de dividir as tarefas de desenvolvimento. Ter-se-ia

uma equipe de especialistas nos "softwares" disponíveis para que fosse dada a orientação mais adequada ao usuário.

1.2 - Assunto da Dissertação

Um problema interessante que surge na aplicação de tal instrumental ao desenvolvimento de sistemas de informação, é de como aumentar a eficiência e rapidez da análise, para poder tirar pleno proveito da maneira expedita como pode ser feita a construção do sistema.

Para resolver esse problema, foi feito um estudo de um instrumental integrado que inclui linguagem de 4.^a geração e participou-se de sua utilização no desenvolvimento de um sistema de controle de cotistas.

O ambiente integrado em que foi desenvolvido o sistema, permite a utilização de recursos de uma linguagem tradicional como o COBOL, bem como de uma linguagem de 4.^a geração pelo usuário final que eventualmente identifique novos requisitos após a implementação, pois os dados que a compõem, foram todos estruturados usando-se um dicionário de dados, que é parte integrante do "software" estudado.

1.3 - Apresentação do Trabalho

No capítulo 2, são apresentadas as principais referências nas quais se baseou o desenvolvimento deste trabalho, tais como: caracterização de linguagem de 4.^a geração; instrumental integrado para desenvolvimento de sistemas, particularmente o IDMS (Integrated Data Management System); conceito de Centro de Informação e Metodologias para análise e projeto de sistemas com enfoque sobre o SADT (Structured Analysis and Design Technique) e projeto estruturado.

No Capítulo 3, são descritos brevemente o ambiente integrado em que foi desenvolvido o sistema, com os componentes: dicionário de dados, ferramenta para desenvolvimento, linguagem de consulta, linguagem voltada para o usuário final e gerador de relatórios.

Nos Apêndices I e II, é encontrado detalhamento da ferramenta usada para o desenvolvimento e da linguagem de consulta, desse ambiente integrado.

No Capítulo 4, é apresentado o sistema que foi desenvolvido utilizando tais recursos, sendo ilustrados parte dos diálogos com os respectivos mapa, processos pré-mapa e de resposta ao mapa com algumas das sub-rotinas utilizadas.

No Capítulo 5, é feita uma avaliação da utilização dos recursos de métodos de análise e projeto de sistemas, de base de dados e de linguagens de consulta no desenvolvimento do sistema apresentado no Capítulo 4.

No Capítulo 6, são apresentadas as conclusões gerais obtidas, sendo feitas algumas sugestões para novas pesquisas.

CAPÍTULO II

RESENHA BIBLIOGRÁFICA

2.1 - Considerações Gerais

A dinâmica empresarial atual, solicita dos sistemas de informação, muitas vezes informações que não haviam sido previstas quando da especificação do sistema. Em ambientes tradicionais de desenvolvimento de sistemas, o usuário que necessita repentinamente de algum relatório ou mesmo alteração de um que já exista, acaba sujeitando-se à disponibilidade da equipe de desenvolvimento, que nem sempre pode atendê-lo de imediato.

É desejável portanto, ter disponível técnicas e ferramentas que permitam aos usuários tornarem-se cada vez mais independentes para o acesso às informações.

Assim, a cada dia, tornam-se disponíveis no mercado, técnicas e ferramentas conhecidas como sendo de 4.^a geração, que objetivam oferecer aos usuários e analistas facilidades de acesso e manipulação de dados, que podem estar armazenados em base de dados ou não. Com tais técnicas, a equipe de desenvolvimento pode entregar ao usuário sistemas em prazos muito menores, além de permitir que o usuário participe efetivamente do projeto. Decorrentes destas técnicas e buscando uma melhor utilização de las, nasceu também o conceito de Centro de Informação.

Não menos importante, são os métodos de Análise e Projeto de Sistemas que podem ser utilizados no desenvolvimento deles. Muitas das falhas que são detectadas nos sistemas deve-se em parte à negligência do analista em efetuar a análise funcional, que pode ser feita utilizando estes métodos. Um outro aspecto, importante também, é a análise dos dados, principalmente para ambientes com base de dados.

Este capítulo tem por objetivo mencionar e comentar trabalhos relevantes, que tratam de instrumental de 4.^a geração e métodos de análise e projeto, que podem ser utilizados no

desenvolvimento de sistemas usando esse instrumental.

Assim, na seção 2.2 são apresentadas referências sobre instrumental de 4.^a geração e conceito de Centro de Informação. Inicialmente são apresentadas aquelas referências que tratam das idéias gerais e críticas sobre tal instrumental, a seguir sobre o IDMS que é um exemplo de instrumental integrado para desenvolvimento de aplicações e finalmente sobre Centro de Informação. Na seção 2.3 são apresentadas referências sobre exemplos de metodologias de Análise e Projeto que poderiam ser acopladas a esse tipo de instrumental, como por exemplo SADT e projeto estruturado. As conclusões deste capítulo encontram-se na seção 2.4.

2.2 - Referências Sobre Instrumental de 4ª Geração e Conceito de Centro de Informação

Em 1981, Martin publicou seu livro [Ma81], onde é dedicado um capítulo para tratar da utilização de linguagem de 4ª Geração, que é iniciado com a apresentação do conceito de tal linguagem. São discutidos os três tipos de abordagens que podem ser consideradas: convencional, com gerador de aplicações, e sem programadores profissionais, que podem muitas vezes coexistir. O ADS, como pode ser visto nas referências que se seguem, é um gerador de aplicações, embora neste trabalho, tenha sido utilizado apenas como gerador de diálogos. A primeira abordagem gera uma grande barreira entre os profissionais de processamento de dados e os usuários e as duas últimas, já permitem uma maior participação do usuário durante o desenvolvimento, graças aos diferentes tipos de "software" disponíveis e que são citados nessa referência. São também discutidos conceitos de "Centro de Informação" e como estes podem contribuir para o desenvolvimento de sistemas de uma forma mais efetiva e satisfatória para o usuário.

Em [Ma82], são discutidas as tendências verificadas na área de processamento de dados, cuja preocupação maior é com o aumento de produtividade e com a melhor forma de atender rápida e diretamente aos usuários, utilizando ferramentas de 4ª geração. São discutidas ainda as características desejáveis em linguagens de programação dirigidas ao usuário final, bem como as facilidades que devem ser oferecidas para que ele desenvolva suas próprias aplicações. Alguns dos produtos disponíveis no mercado, bem como os tipos de ambientes de dados que podem ser definidos, são também abordados. Um outro aspecto que mereceu atenção foi aquele que se refere a Centro de Informação, sendo discutidos o conceito, a sua administração, os fatores que justificam a sua implantação, sendo também apresentados exemplos de Centro de Informação em funcionamento.

Em 1985, Tinnirello em [Ti85], discute a importância de um estudo criterioso para escolha da linguagem de 4ª Geração a ser utilizada. Sugere e discute alguns aspectos que podem ser considerados tais como: facilidade e eficiência na utilização, sua integrabilidade com os demais "software", a assistência

oferecida aos usuários pelo fornecedor, etc. É lembrado também que uma linguagem de 4.^a geração não faz desaparecer o custo de manutenção de sistemas, ela apenas diminui tal custo. Consta ainda nessa referência, uma tabela enumerando linguagens de 4.^a geração disponíveis no mercado, com os respectivos fornecedores, exigência de máquina e pacotes. Entre elas é citado o ADS que é objeto dos capítulos 3 e 4, bem como do Apêndice I desta dissertação.

Ainda em 1985, Rudolph discute em [Ru85], o aspecto de produtividade face à utilização de linguagem de 4.^a geração no desenvolvimento de aplicações. É dito que para escolher a linguagem de 4.^a geração que deve ser utilizada, deve-se estabelecer alguns critérios tais como: generalidades da linguagem (que esteja integrada ao "software" básico do fabricante), extensão do "hardware" apoiado (que toda a linha do fabricante esteja coberta para permitir o máximo de mobilidade) e o número de usuários, sendo discutidos cada um deles. O LINC (Logical and Information Network Computer) da Burroughs é apresentado, com uma breve descrição dos seus blocos básicos. É dito ainda que para medir a produtividade, o trabalho realizado deve ser avaliado juntamente com o valor ou tamanho do produto gerado. São sugeridos três métodos para definir tamanho de programa: linhas de código fonte, sistema métrico de Halstead e pontos por função, sendo discutidos cada um deles, com indicações de referências bibliográficas para os dois últimos métodos. No entanto, o método de pontos por função é o discutido em maiores detalhes. É concluído que as linguagens de 4.^a geração, entre elas o LINC, vem comprovar o ganho de produtividade citado em [Ma82], mesmo quando utilizado por pessoas com conhecimento limitado de processamento de dados e que os altos investimentos iniciais são rapidamente justificados, pelo aumento de produtividade. No entanto, é dito que em particular o LINC, consome muito tempo em qualquer modificação efetuada, por menor que seja, pois exige uma recompilação completa do programa. É lembrado também que a utilização das linguagens de 4.^a geração, pode esbarrar em dois fatores limitantes: a) a participação cada vez mais intensiva do usuário final permite que ele também faça exigências cada vez mais complexas e abrangentes, b) o hardware disponível pode apoiar tal complexidade somente até certo ponto. O LINC, como o ADS, pode ser utilizado como uma fer

ramenta no desenvolvimento de aplicações do tipo daquela apresentada no Capítulo 4.

Os conceitos do IDMS, que integra as ferramentas discutidas no Capítulo 3 desta dissertação, relativos a elementos, registros, sub-esquema, esquema, DMCL (Device Media Control Language), bem como as relações que podem ser estabelecidas entre os registros, são apresentados em [C83e]. Em relação ao aspecto físico, são apresentados conceitos de página, área, arquivo e diferentes modos de localização que podem ser utilizados para acesso aos registros. Os comandos de acesso à base de dados, os aspectos a serem considerados quando da definição de esquema, sub-esquema, DMCL, são brevemente descritos. Ainda são apresentados alguns comandos utilitários que podem ser utilizados pelo administrador da base de dados (DBA), para criação e/ou manutenção de uma base de dados.

Em [C83c], são apresentadas sugestões de diretrizes para elaboração do projeto de uma Base de Dados IDMS. Assim, são tratados inicialmente aspectos relativos à identificação, coleta e análise de dados, identificação das funções do sistema e estabelecimento de relações entre registros, bem como as convenções que devem ser estabelecidas para atribuição de nomes a elementos, registros e relações. Com as informações obtidas pode ser elaborado o diagrama de estrutura correspondente, com a preocupação de que todas as funções sejam satisfeitas. São também abordados aspectos relativos ao desempenho, privacidade e segurança, definição de área, página, etc... que devem ser considerados no projeto. Finalmente são tratados aspectos relativos à implementação -definição de esquema, sub-esquema e DMCL.

Em [C83f], são discutidos critérios para análise e projeto de base de dados, abordando a especificação conceitual, de projeto e de implementação. São apresentados por exemplo, sugestões para o cálculo do tamanho de páginas e determinação da posição dos ponteiros quando da especificação de SET LINKAGE (ligação entre registros mestre e membro) e opções de segurança que podem ser consideradas na definição do sub-esquema, para impedir o acesso de pessoas não autorizadas. São também apresentados e discutidos os comandos da DDL (Data Definition Language) utilizados para definição de esquema, sub-esquema e DMCL, com as respectivas sintaxes e cláusulas válidas.

Em [C82c], são apresentadas as características , benefícios e conceitos do IDD (Integrated Data Dictionary), que é o dicionário de dados integrado no IDMS. São discutidas as formas utilizadas para definir as ocorrências de entidades, bem como os diferentes tipos de relatórios que podem ser obtidos do dicionário de dados. Através desses relatórios, os usuários e analistas podem controlar e verificar o conteúdo do dicionário de dados, assegurando assim uma melhor utilização dos recursos disponíveis. São também discutidos os aspectos de segurança que devem ser considerados para que se obtenham informações exatas, oportunas e seguras.

Em [C83d], ainda sobre o IDD, são apresentadas e discutidas as entidades e operações que podem ser especificadas sobre os dados. As entidades são divididas em três tipos: básicas de teleprocessamento e especiais, sendo apresentadas as cláusulas e sintaxes válidas, acompanhadas de exemplos. São também apresentados os comandos que podem ser utilizados "on line" para executar operações sobre ocorrências das entidades.

Em [C84b], são apresentadas as características de um bom dicionário de dados. São sugeridas e discutidas etapas que poderiam ser seguidas quando do início da utilização de um dicionário de dados: planejamento, implementação e avaliação, de modo que possam ser melhor aproveitados os recursos disponíveis. Para proporcionar segurança e integridades dos dados, são sugeridos alguns critérios tais como: convenções de nomes, cronograma de processamento, utilização de rotinas padronizadas, etc... Os aspectos relativos à segurança (cópias ou restauração) da base de dados e da utilização "on line" do IDD para manutenção das ocorrências de entidades são também discutidos. O dicionário de dados pode ser utilizado por outros "softwares" que estejam integrados ao IDMS, assim os aspectos de integração e interação são também abordados.

Em [C82b], são apresentados os procedimentos para definir mapas "on line" e em "batch". Mapas devem ser definidos para cada diálogo ADS. Para definição de mapas "on line", são utilizadas as facilidades do "on line mapping", através de um conjunto de telas contendo campos que são preenchidos pelo analista e outros que o sistema utiliza para fornecer-lhe informações.

Cada um desses campos é explicado, sendo acompanhado de um exemplo completo para definição de mapa. Para definição de mapas em "batch", são apresentados os comandos que devem ser utilizados. São discutidos ainda aspectos relativos à manipulação de erros e edição automática, que devem estar definidos no dicionário de dados, bem como as considerações que devem ser feitas quando da definição, alteração ou utilização de elementos, registros ou tabelas utilizadas pelo mapa.

Em [Cl82a], são apresentadas as principais características do ADS/O, considerando o diálogo como uma componente da estrutura de uma aplicação. São também brevemente apresentados, os componentes de um diálogo, sendo citados os tipos de comandos que são válidos para definição de um processo, os comandos que definem o controle inter-diálogos e o conjunto de telas utilizado para gerar um diálogo.

Em [Cl83a], são apresentadas as finalidades de cada uma das telas que são utilizadas para gerar aplicações e diálogos, sendo explicado como o analista deve proceder para o preenchimento delas. Se houver necessidade de modificar ou delir componentes de aplicações e/ou diálogos, é explicado como proceder. São discutidos aspectos relativos à manipulação de erros e edição automática de campos que devem estar definidos no dicionário de dados, bem como do estabelecimento de níveis de segurança para impedir acesso às aplicações, por pessoas não autorizadas. É também explicado como proceder quando ocorrem erros durante a execução de um diálogo, com base nas mensagens de erro que são emitidas.

Em [Cl84a], são apresentados o gerador de aplicações, o gerador de diálogos e o sistema de execução para aplicações ADS/O, com as telas necessárias para a definição de aplicações e de diálogos, sendo explicadas cada uma das cláusulas que podem ser especificadas em cada uma das telas. São apresentados os comandos aritméticos e de atribuição, de definição de estrutura de iteração e seleção, de controle de sub-rotinas, de controle de processamento, de acesso e manipulação da base de dados, de modificação de mapas, de administração de fila e para restauração de processamento, bem como utilitários que são válidos para definição de processos associados aos diálogos. Para cada um des

ses comandos, é apresentada a sintaxe correspondente, sendo explicada cada uma das cláusulas que os compõem. As mensagens que são exibidas quando da execução de uma aplicação, são também apresentadas. Pode-se ainda encontrar nessa referência, os diferentes tipos de relatórios disponíveis ao projetista de aplicações ADS/O.

Em [Cl83j], são apresentados comandos e procedimentos válidos para utilização da linguagem de consulta (OLQ), que é objeto do Capítulo 3 e Apêndice II deste trabalho. São apresentados comandos para simples consulta, para definição de caminhos e q-arquivos, de controle e ajuste, bem como de formatação de relatórios, sendo explicadas a sintaxe e cláusulas válidas para cada um. Para comandos de ajuste e formatação de relatórios, são definidas algumas cláusulas quando da instalação do sistema, que permanecem em efeito até que sejam explicitamente alteradas.

Em [Cl83h], são apresentadas as características e vantagens da On Line English (OLE), que é uma linguagem voltada ao usuário final e que é também objeto do Capítulo 3 deste trabalho. O vocabulário que é utilizado é definido no "lexicon". São discutidas através de exemplos, as diferentes formas em que o usuário pode formular consultas e a forma como a OLE as interpreta. São ainda apresentados e ilustrados através de exemplos, processos pré-definidos no "lexicon", que podem ser referenciados por palavras-chaves, para executarem as funções definidas. É ainda lembrado o processo interativo que é estabelecido entre a OLE e o usuário final, pelas mensagens de erro emitidas pelo sistema e pelas definições complementares fornecidas pelo usuário, em resposta a essas mensagens.

Em [Cl83i], são apresentados os procedimentos para definição do "lexicon da aplicação" que é o seu vocabulário específico e considerações para o projeto das entidades. Para cada comando que pode ser utilizado para definir o "lexicon", é apresentada a sintaxe correspondente e as cláusulas válidas.

Em [Cl81], são apresentadas as características e benefícios do gerador de relatórios CULPRIT. São incluídos vários exemplos de relatórios, que procuram demonstrar através de discussões a flexibilidade para formatação de relatórios que os recursos disponíveis oferecem aos usuários que utilizam essa ferramen

ta, proporcionando assim um significativo aumento de produtividade. Dentre os exemplos apresentados, consta um em que o mesmo relatório foi gerado através de um programa codificado em COBOL e também utilizando CULPRIT, onde é evidente a redução do número de comandos utilizados.

Em [C083b], é apresentada a estrutura de programas CULPRIT, sendo discutidos os parâmetros de definição de perfil, entrada, saída, processos, área de trabalho e índices a serem de finidos. Para cada um desses parâmetros, é apresentada a sintaxe e as cláusulas que podem ser utilizadas, bem como as fases pelas quais passam os programas CULPRIT. São ainda discutidas as con siderações a serem feitas quando se define programas CULPRIT em am biente IDMS, os procedimentos a serem seguidos quando houver ne cessidade de combinar vários arquivos, quando houver necessidade de ter acesso a partes de programas CULPRIT armazenados ou ainda quando houver necessidade do usuário definir módulos para post erior utilização.

Em [Ibxx], é discutida a forma como os sistemas tem sido tradicionalmente desenvolvidos, a dificuldade de for ne cer informações ao usuário que solicita uma alteração ou um rela tório não rotineiro. É definido o que vem a ser um Cent ro de In formação (CI) e a sua estrutura organizacional, com os prof issio nais necessários para proporcionar um melhor atendimento aos usuá rios. São sugeridas também algumas áreas de empresas que pode riam beneficiar-se com um CI, sendo indicados alguns produtos que melhor atenderiam às atividades por elas executadas. Cada um des ses produtos é brevemente descrito.

Em [Sq81], é discutida a importância da docu menta ção de um CI estar disponível ao usuário. Assim, é apresentada su gestão do conteúdo que teria um guia do usuário, do "noticiário" que deve ser distribuído aos usuários e modelos de formulários que devem ser utilizados pelos usuários que solicitam ser viços, etc. ao CI. O guia do usuário, poderia conter: objetivos do CI, os benefícios, a seleção de produtos que atenderão as di ferentes áreas, os serviços de apoio disponíveis no CI para que os usuários possam tirar melhor proveito dele e os procedimentos para utilizar um produto e/ou o CI. São identificados alguns pro dutos que poderiam ser implantados em um CI, sendo apresentado

para cada um: descrição de o que é, quais as suas principais características, quando utilizá-los e quais os cursos que devem ser assistidos.

Em [Sq82], é discutido o que é e como proceder à implantação de um Centro de Informação (CI). São ainda apresentados: como identificar a necessidade de um CI, as responsabilidades atribuídas aos usuários e ao CI e os benefícios que o departamento de processamento de dados e o usuário final terão. Um dos compromissos do CI é fornecer apoio adequado aos usuários através de cursos, consultoria, assistência na resolução de problemas, planejamento de recursos de "hardware", etc. e não apenas tornar os produtos disponíveis. No que se refere à implantação do CI, é apresentada uma sugestão onde é elaborado um projeto piloto e identificadas fases que vão sendo executadas sucessivamente, até que o CI entre em operação. São sugeridos organogramas que comportam a implantação de um CI, sendo também discutidas as funções de cada elemento que compõe o seu corpo administrativo.

Em [Ha82], é discutido o elo de ligação que é estabelecido entre o usuário final e o sistema de informação através do CI, possibilitando assim que o próprio usuário execute algumas tarefas, antes executadas apenas pela equipe de desenvolvimento de sistemas. São citadas algumas razões que identificam a implantação de um CI, sendo sugeridos os componentes do documento: declaração da missão do CI, que é elaborado quando de sua implantação. São ainda sugeridas a organização da sua estrutura e os profissionais que compõe o seu corpo administrativo, sendo alertado no entanto que, inicialmente não há necessidade de muitos elementos devido a pequena quantidade de usuários. É lembrado que deve ser considerado o aspecto segurança sobre os dados. São discutidos também fatores tais como: produtividade, custo, benefícios através de exemplos, que justificam a implantação de um CI na empresa.

Em [So84], é discutido como surgiram as primeiras idéias de CI. São analisados os resultados obtidos, onde é evidente o aumento da disponibilidade de recursos para desenvolvimento de sistemas. Ainda é feito um estudo da estrutura organizacional e da equipe necessária para um CI, dependendo dos produtos que serão utilizados, recursos e áreas da empresa que devem

ser atendidas. São sugeridas algumas áreas e/ou atividades de uma empresa que poderiam utilizar um CI, sendo indicados alguns dos produtos disponíveis no mercado, que melhor se adequariam a essas áreas. São também discutidos planos de implantação e/ou operação de CI em quatro empresas brasileiras.

2.3 - Referências Sobre Exemplos de Metodologias de Análise e Projeto que poderiam ser Acopladas a esse Instrumental

Em [Sf76], é apresentado o método SADT (Structured Analysis and Design Technique), objeto do Capítulo 5 deste trabalho para documentar o sistema desenvolvido. São apresentados os conceitos de: modelo, a forma hierárquica e estruturada com que o problema é analisado, o enfoque dual sobre dados e atividades e a linguagem gráfica utilizada. É discutido ainda o procedimento a ser seguido quando da utilização do SADT durante a análise funcional, que compreende a elaboração de modelos SADT que devem ser usados para comunicar as idéias do grupo de trabalho. Essa comunicação consiste no ciclo leitor/autor onde são feitas todas as anotações que forem de interesse de modo que, as idéias que ficarem documentadas, serão aquelas que melhor expressam as exigências do sistema. É apresentado um exemplo de utilização do método SADT, inclusive com decomposição dos diagramas de atividades e dados-ativigramas e datagramas respectivamente, as relações estabelecidas pelas setas e as convenções gráficas utilizadas. São também apresentados os documentos adicionais que podem ser incluídos para um melhor entendimento quando da leitura dos diagramas SADT.

Em [Ro77a], é discutida a importância e necessidade da definição dos requisitos quando do desenvolvimento de sistemas de informação. É dito que muitas vezes o projeto e implementação começam antes que as reais necessidades e funções do sistema sejam totalmente conhecidas, o que provoca elevados custos e infundáveis remendos e reparos denominados "manutenção de sistema". É lembrado que a definição dos requisitos não deve se limitar à análise do problema que produz uma especificação funcional, mas deve ser muito mais abrangente: deve dizer porque o sistema é necessário (análise do contexto), o que será o sistema em termos das funções que deve executar (especificação funcional) e como isto deve ser construído (restrições do projeto). É dito ainda que a especificação funcional é modular e hierárquica, permitindo uma especificação que não é necessariamente como seria o sistema fisicamente e permite enxergar o sistema sob diferentes pontos de vista. É ressaltada também a importância da forma de documentação para comunicação de idéias e a interação que

deve existir entre as pessoas envolvidas no sistema. É apresentada brevemente o método SADT, como uma ferramenta que pode ser utilizada para a especificação funcional. É dito ainda que, a interação com o usuário até que se atinja um completo entendimento do sistema, com uma documentação adequada, é a chave do sucesso da especificação funcional e conseqüentemente do sistema como um todo.

Em [Ro77b] é discutida a necessidade de uma "linguagem" para comunicar idéias de uma forma clara e disciplinada no desenvolvimento de sistemas. A *Structured Analysis* (SA) é apontada como tal linguagem. Ela é derivada da forma como as situações dos problemas do mundo real são pensadas e entendidas, isto é, em termos de dados e atividades. A comunicação é baseada no princípio do "contador de estórias", que produz uma decomposição "top down". A saída, é uma estrutura hierarquicamente organizada de diagramas, em que cada um expõe apenas uma parte limitada do assunto, de modo que mesmo os assuntos mais complexos podem ser entendidos. A coleção de diagramas é denominada *modelo SA*. A SA sugere que dados e atividades sejam expressos em diagramas com não mais do que seis componentes, sendo que cada um desses componentes pode ser decomposto em outros diagramas. Esse processo recursivo pode ser aplicado até que não mais se justifique a decomposição. São apresentadas várias características da linguagem SA que são discutidas e ilustradas utilizando os recursos gráficos disponíveis no próprio método SADT. É dito ainda que, a forma como a SA é considerada, permite traduzir de uma forma organizada, rigorosa e clara, os requisitos de solução de projeto, implementação e manutenção de sistemas.

Em [Ch79], são apresentados exemplos onde são retratados os recursos gráficos disponíveis nos métodos: SADT com o ativograma e datagrama; análise estruturada com o método de bolhas e Diagrama de Fluxo de Dados (DFD). São discutidos os recursos desses métodos em relação à organização do diagrama, dado e fluxo de dados, processos e utilização dos diagramas. Dessa discussão é dito que o DFD e o método de bolhas, permitem expressar o diagrama sem muita pressão de espaço, enquanto que o ativograma/datagrama já exigem um controle mais rigoroso, quer na disposição como no detalhamento expresso a cada nível, além de não permitirem uma representação explícita das entidades receptoras ou

emissoras de informações. No que diz respeito à facilidade de utilização, é dito que o DFD e o diagrama de bolhas são mais fáceis de serem utilizadas. É lembrado também o conceito de modelo sobre o qual se baseia a elaboração de ativigramas e datagramas.

Em [La79], são apresentados conceitos para utilização de N^2 -charts. Na apresentação deles, utilizando a notação matricial, as funções são colocadas nas diagonais e as entradas e saídas são colocadas nas linhas e colunas, tudo isso organizado de uma forma um tanto quanto rigorosa. A representação gráfica aqui sugerida, talvez seja a idéia inspiradora para a representação gráfica do método SADT, inclusive no que se refere ao rigor da disposição gráfica. São também apresentados vários exemplos utilizando N^2 -charts.

Em [Ct81], são apresentadas as bases sobre as quais, possivelmente, se fundamentará a evolução dos métodos de análise e projeto de sistemas, através de uma série de trabalhos de diferentes autores. Entre eles em [En81] são discutidas as diferentes fases que podem ser identificadas no ciclo de vida de um sistema. Essas fases são identificadas para a abordagem tradicional e abordagem estruturada. Durante a discussão, são apresentadas técnicas (principalmente no que se refere à abordagem estruturada, como por exemplo: implementação "top-down", DFD, dicionário de dados, programação estruturada, etc...) que podem ser utilizadas em cada fase. As fases, atividades e documentos produzidos em cada fase para as duas abordagens, são também apresentadas.

No artigo de Chapin [Ch81a], é discutida a utilização de técnicas gráficas no desenvolvimento de sistemas. São sugeridas algumas considerações, como por exemplo: completividade dos diagramas produzidos, a utilidade dos diagramas, a não exigência de trabalho adicional para elaboração dos diagramas, o nível de detalhes, a habilidade exigida para a elaboração deles, etc... que devem ser consideradas para a escolha da técnica. É apresentada uma tabela contendo várias técnicas gráficas, dentre elas, o diagrama de estrutura, ativograma e datagrama. Essa tabela aborda dois aspectos: de um lado a utilização e de outro a natureza dessas técnicas. Assim, no primeiro aspecto ativogramas/datagramas tem como objetivo estabelecer interação de atividades/dados. Os diagramas são produzidos nas fases de análise e projeto das

interfaces entre as partes manual e automatizada do sistema. São de média facilidade para preparação e utilização e relativamente difíceis para manutenção. No segundo aspecto, na vertical dos diagramas pode-se ver os itens separados e o controle e na horizontal os itens separados e as entrada/saída. A direção de leitura é do canto superior à esquerda para o canto inferior à direita, sendo permitido ao leitor controlar o nível de detalhamento desejado. São ainda brevemente descritos alguns métodos, entre eles o SADT com detalhes para atvigramas e datagramas.

Em [Ch81b], são discutidos métodos de análise e projeto estruturado. Figuram entre os métodos de análise estruturada, o diagrama de fluxo de dados, atvigr^ama/datagrama, etc..., sendo dada uma breve descrição de cada um deles. É feita uma análise entre eles e dito que atvigr^ama/datagrama é aquele que mais envolve a participação do usuário, graças ao estabelecimento do ciclo autor/leitor, no entanto é aquele que dispensa a menor atenção no que se refere à combinação de condições. São discutidos também métodos de projeto estruturado, dentre eles o diagrama de estrutura, que é derivado do fluxo de dados.

Em [Cn81], são discutidos fatores tais como alto custo de software, identificação de erros na análise e projeto que influem no desenvolvimento de sistemas. É apresentado o método SADT, discutindo-se atvigramas, datagramas, modelos, referências cruzadas, e pontos de vista. É dito que o SADT é um bom método para ser utilizado na análise do sistema. É discutida a participação do usuário no ciclo leitor/autor que resulta na documentação de um sistema com menor probabilidade de erros. Um outro aspecto que é considerado, é a possibilidade de efetuar o rastreamento verificando se as exigências do sistema foram todas consideradas, o que é feito através da referência cruzada de diagramas resultantes de diferentes pontos de vista e modelos.

Em [Yo78], são apresentados os conceitos de Projeto Estruturado, sendo inicialmente apresentada a terminologia utilizada. São discutidos conceitos de acoplamento e coesão, que podem ser utilizados na avaliação da qualidade de uma modularização. São consideradas as abordagens: análise das transformações e análise das transações para construção de sistemas estruturados, sendo apresentadas sugestões para elaboração de projetos. São ain

da discutidos aspectos relativos à implementação pelas abordagens "top-down" e "bottom-up". No Capítulo 5 deste trabalho constam diagramas de estrutura correspondentes a uma parte do sistema desenvolvido.

Em [St85], são apresentados e discutidos os conceitos de projeto estruturado, incluindo acoplamento e coesão. São também discutidas sugestões como abordar problemas e as diretrizes a serem seguidas para elaboração do projeto. É apresentado um exemplo, que é gradativamente construído, sendo discutido cada passo. São discutidos como o projeto estruturado pode aumentar a produtividade: tempo, facilidade de correção e reutilização dos módulos.

Em [Ga84], é discutida a análise estruturada como uma ferramenta para especificação funcional de um sistema de informação. Assim, é explicado como analisar e definir "o que" o sistema faz, que depois é expresso no diagrama de fluxo de dados (DFD), utilizando blocos gráficos para representar cada componente: entidade externa, processo, fluxo de dados, depósito de dado. São discutidos conceitos de normalização, árvore e tábua de decisão, inglês estruturado, que podem ser utilizados para descrever alguns dos componentes gráficos. São também apresentados os conceitos de projeto estruturado, cuja ilustração é derivada do DFD do problema-exemplo apresentado nessa referência.

2.4 - Conclusões

A busca do aumento de produtividade no desenvolvimento de aplicações, tem incentivado a procura de técnicas que proporcionem tal melhora.

Dentre outros disponíveis no mercado, o IDMS, que acomoda um conjunto integrado de "softwares" pode ser utilizado no desenvolvimento de aplicações.

As ferramentas por ele integradas, apresentam facilidades que contribuem para agilizar o desenvolvimento de aplicações e também na fase que sucede à implementação, se efetivamente utilizadas.

No entanto, do material analisado, no que se refere ao IDMS, não é dedicada muita atenção quanto à representação gráfica, que para o pessoal administrativo seria de grande valia, sem contar que, pela própria finalidade, poderia ser utilizada diretamente pelo usuário final. No que se refere à análise e projeto de sistemas, observou-se que também não é dispensada a atenção merecida. Se por um lado são oferecidos os recursos de um gerador de aplicações com a definição da estrutura da aplicação com as funções e também o gerador de diálogos, além do tratamento tão criterioso dispensado ao projeto de base de dados, por outro lado não há preocupação em abordar de uma forma mais sistemática o projeto de diálogos.

Os métodos de análise e projeto de sistemas aqui apresentados, procuram oferecer por um lado recursos que permitam retratar o aspecto dado e por outro a estrutura modular de sistemas. Ambos os aspectos são abordados pelo instrumental integrado IDMS. Assim, estes métodos parecem oferecer recursos que poderiam ser utilizados no desenvolvimento de sistemas em tal ambiente.

CAPÍTULO III

UM EXEMPLO DE INSTRUMENTAL DE 4.^a GERAÇÃO

3.1 - Considerações Gerais

Este capítulo tem por objetivo apresentar um exemplo de instrumental de 4.^a geração disponível no mercado, que pode ser utilizado para desenvolvimento de sistemas.

Um bom instrumental de 4.^a geração deve ter, tanto quanto possível, recursos que permitam a máxima participação do usuário durante o desenvolvimento, bem como que o tornem independente ao ponto de conseguir obter respostas às questões do tipo "o que se" que são de grande valia no processo de tomada de decisão.

O instrumental aqui estudado é o IDMS - Integrated Data Management Systems, que é brevemente descrito nesta seção.

Os detalhes para um melhor entendimento dos recursos descritos nesta seção, podem ser encontrados nas referências bibliográficas que constam na seção 2.2.

Uma aplicação de parte dos recursos aqui descritos pode ser encontrada na implementação de um sistema utilizando o IDMS, que é apresentada no Capítulo 4. Uma avaliação da utilização dos recursos IDMS no que se refere à linguagem voltada para o usuário e base de dados, é encontrada no Capítulo 5.

A seguir é dada uma breve descrição do IDMS. Maiores detalhes são encontrados nas referências citadas no texto.

O IDMS é um "software" gerenciador de Base de Dados dirigido ao dicionário de dados.

Uma base de dados IDMS tem as seguintes componentes: *esquema*, *sub-esquema* e "*Device Media Control Language - DMCL*".

Um esquema é a descrição completa de uma base de dados contendo nomes e descrições de todos os arquivos, registros,

áreas e relações estabelecidas entre registros denominadas *sets*.

Um sub-esquema é uma visão da base de dados que satisfaz as necessidades de dados de um grupo de programas ou apliquações. Pode constituir-se de toda ou parte das descrições de elementos, registros, *sets* e áreas definidas no esquema.

DMCL é a componente utilizada para definir tamanho de "buffer", de área, de páginas e opcionalmente "journal file" que é utilizada para recuperação de informações.

Para descrever o esquema, sub-esquema e DMCL quando da definição de uma base de dados, tem se disponível a DDL (Data Description Language) cujos detalhes podem ser encontrados em [Cl83f].

A análise e projeto de uma base de dados, envolve atividades de identificação de elementos, registros, arquivos e *sets*. Nas relações que são estabelecidas entre registros, sempre é identificado o registro mestre e o registro membro, podendo ser um dos seguintes tipos: 1:1, 1:n e n:m, sendo que no caso n:m é dado um tratamento especial através da "junction record". Uma vez que os elementos, registros e *sets* foram acomodados satisfazendo às restrições IDMS, deve ser desenvolvido o diagrama de estrutura que contém informações relativas aos registros, como por exemplo: comprimento, modo de localização, modo de armazenamento; informações relativas aos *sets*, tais como: ordem de ocorrência dos registros membros, forma como um registro é conectado ou desconectado de um *set* e informações relativas à área que será atribuída ao registro.

O IDMS permite que as informações sejam mantidas em arquivos VSAM (Virtual Storage Access Memory) sem que sejam definidos como Base de Dados. Nesse caso, são feitas algumas considerações como pode ser visto em [Cl83f]. Detalhes complementares ao IDMS podem ser encontrados em [Cl83e].

Na Figura 3.01 estão ilustrados os recursos disponíveis em um ambiente IDMS. Alguns deles são descritos neste Capítulo. Assim, na seção 3.2 é apresentado o IDD (Integrated Data Dictionary) sistema para manutenção do dicionário de dados, na seção 3.3 o ADS/O (Application Development System/on Line) para desenvolvimento de aplicações, na seção 3.4 a linguagem de con

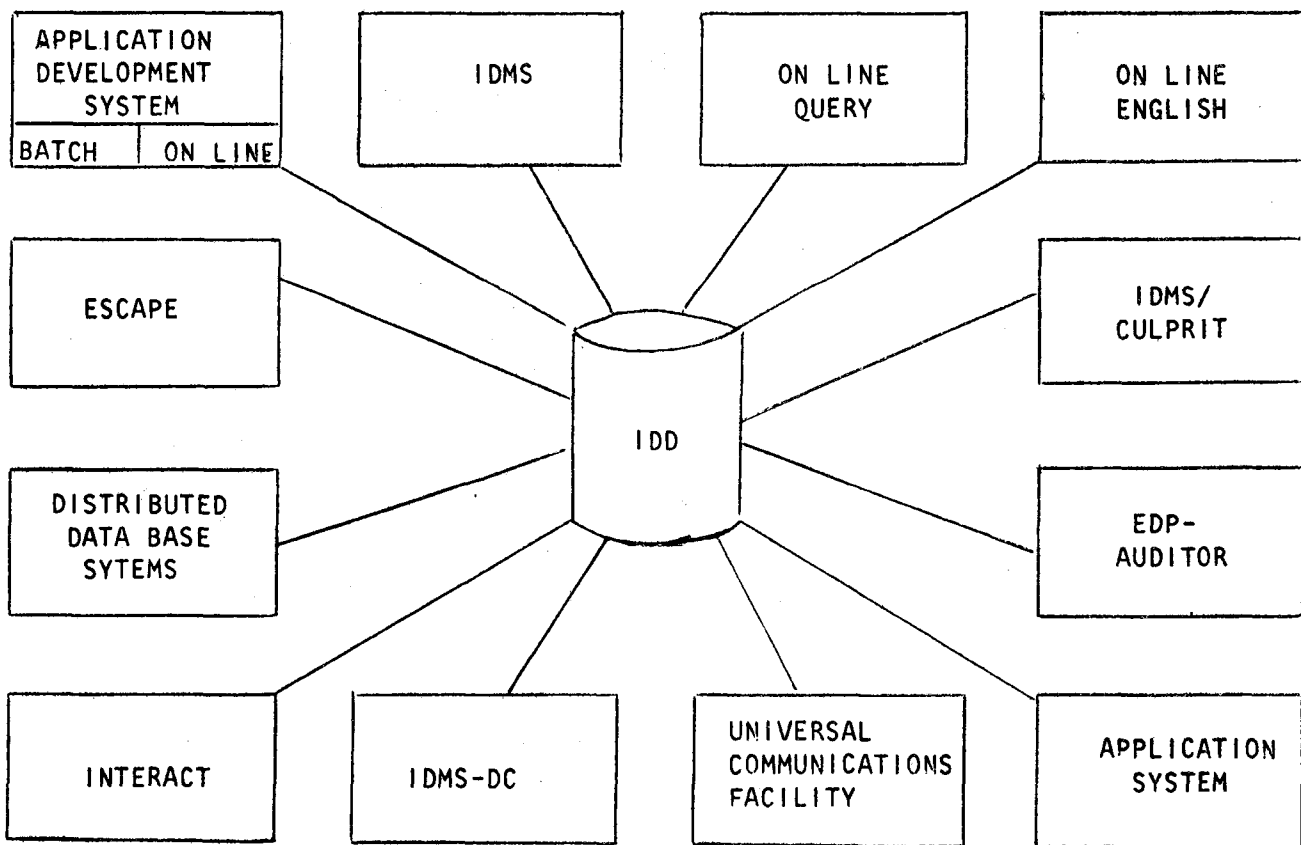


Figura 3.01 - Software Integrado de Administração de Dados.

sulta OLQ (On Line Query), na seção 3.5 o gerador de relatórios CULPRIT e a linguagem voltada ao usuário final OLE (On Line English). A seção 3.6 finaliza o capítulo com as conclusões.

Constam ainda no final do trabalho apêndices com descrição detalhada do ADS/O e do OLQ.

A seguir é dada uma descrição compacta de cada elemento da Figura 3.01, ainda não mencionado no texto anterior.

- IDMS-DC: é o monitor de teleprocessamento integrado ao IDMS.
- Distributed Data Base System: é o sistema que permite programas de aplicação sendo executados em múltiplas CPU, terem acesso e atualizar base de dados comuns.
- Application Development System/Batch: é a ferramenta que permite centralizar e simplificar as tarefas de validação de arquivos de transação e atualização de base de dados IDMS.
- Universal Communications Facility: é a ferramenta que possibilita o desenvolvimento de aplicações on line, sob qualquer monitor de teleprocessamento integrado com IDMS.
- Escape: é um conjunto de interfaces que permite utilizar DL/1 com IDMS.
- Interact: é o sistema que pode ser utilizado para edição de texto.
- EDP-Auditor: é um conjunto de rotinas que permite executar tarefas de auditoria como por exemplo, análise de excessão, modelagem.
- Application System: são aplicações desenvolvidas para finalidades específicas como por exemplo: ambiente bancário, de manufatura (controle estoque, produção, compra) e financeiro.

3.2 - O Dicionário de Dados - IDD

Os recursos dos dados que fluem em uma organização devem ser mantidos o mais atualizado possível, cuidando também dos aspectos integridade e flexibilidade. O IDMS oferece uma ferramenta - IDD - que é brevemente descrita a seguir.

Integrated Data Dictionary - IDD - é o sistema utilizado para armazenamento e manutenção dos recursos de dados num depósito central, denominado dicionário de dados.

No dicionário de dados são armazenadas ocorrências de componentes denominadas *entidades* e que podem ser divididas em: entidades básicas, de teleprocessamento e especiais.

As entidades básicas correspondem por exemplo a elemento, registro, arquivo, sistema, usuário, programa, etc...

As entidades de teleprocessamento correspondem às componentes do processamento imediato como por exemplo mapa, linha, terminal lógico, terminal físico, destino, etc...

As entidades especiais correspondem por exemplo a módulo de carga, entidades definidas pelo usuário, etc...

Uma ocorrência de entidade é uma instância específica de uma particular entidade. Assim, pode-se ter por exemplo a entidade ELEMENT e a ocorrência NOME-CLIENTE.

O dicionário de dados pode ser utilizado na fase de análise e projeto de sistemas, uma vez que para os dados obtidos através de entrevistas, usuários, fluxo de informações, podem ser definidas informações como fonte dos dados, características físicas, relações, frequência de acesso, etc... Para cada um ou conjunto desses dados pode ser identificada um entidade.

O IDD tem como componente a Data Dictionary Description Language (DDDL) que é utilizada para definir as entidades no dicionário de dados, cujos detalhes podem ser encontrados em [Cl83d], [Cl84b]. No entanto, a alimentação do dicionário de dados não deve ser feita indiscriminadamente. Devem ser estabelecidos padrões a serem seguidos como por exemplo, convenções para dar nomes a entidades, definição de rotinas padrão para que os sistemas que delas necessitem possam usá-las sem duplicar a defi

nição de rotinas, estabelecimento de cronogramas para o pessoal de produção, para que os dados necessários já estejam efetivamente atualizados, assegurar que somente pessoas autorizadas tenham acesso às informações, o que pode ser feito através de senhas e atribuição de níveis de autorização para ter acesso. O IDD permite ainda que sejam definidos sinônimos para nomes de entidade, oferecendo assim uma maior flexibilidade, pois uma única ocorrência da entidade no dicionário de dados, pode ser referenciada por diferentes nomes de acordo com as necessidades dos usuários.

Uma vez que os dados tenham sido armazenados no dicionário de dados, eles se tornam imediatamente disponíveis. Assim, se o usuário desejar, o IDD tem como componente o Data Dictionary Report (DDR) que permite ao usuário obter diretamente do dicionário de dados, diferentes tipos de relatórios. Alguns desses relatórios, como por exemplo relatório por programa, por usuário, por registro, relatório sumário, relatório com descrições do elemento, etc. são denominados relatórios-padrão e são fornecidos quando da instalação do sistema. No entanto, é dada ao usuário a flexibilidade de criar relatórios próprios quando a aplicação assim exigir.

O IDD oferece ainda recursos "on line", que podem ser utilizados para atualização de descrições de ocorrências de entidades no dicionário de dados. Estão disponíveis para utilização por exemplo os comandos FIND, END, SWAP, PRINT, DISPLAY.

3.3 - A Ferramenta para Desenvolvimento - ADS/O

É desejável uma ferramenta de fácil manipulação para o desenvolvimento de aplicações "on line". Para tanto, o IDMS tem disponível a ferramenta ADS/O que é brevemente descrita a seguir (no que se refere à aplicações compostas por diálogos). O Apêndice I no final deste trabalho contém uma descrição mais detalhada que pode ser complementada com as referências bibliográficas da seção 2.2.

Application Development System / On Line - ADS/O - é a ferramenta utilizada para desenvolvimento de aplicações "on line" em instalações que utilizam o IDMS.

Uma aplicação pode ser composta de um ou mais diálogos. Um diálogo pode ser entendido como uma unidade de trabalho ADS. A cada diálogo está associado um mapa, um ou mais processos, sub-esquema e registros que devem estar armazenados no dicionário de dados, para que o diálogo possa ser gerado e conseqüentemente a aplicação ser executada.

Para que se possa proceder à definição de um mapa é necessário que os registros de trabalho e tabelas a serem utilizados, estejam definidos no dicionário de dados. Uma vez que os registros e as tabelas tenham sido definidos, o projetista pode dar início à definição do mapa propriamente dita, utilizando os recursos de On Line Mapping (OLM), cujos detalhes podem ser encontrados em [C82b], que compreende o preenchimento de um conjunto de telas, onde são definidas características inerentes a cada campo que tenha sido "desenhado" no mapa.

Os processos que compõem um diálogo podem ser de dois tipos: pré-mapa e de resposta ao mapa, sendo eles quem definem o processamento a ser executado.

A cada diálogo pode estar associado um processo pré-mapa e um ou mais processos de resposta ao mapa. Um processo pré-mapa é caracterizado por efetuar todo o processamento necessário antes que o mapa seja exibido ao usuário. Já o processo de resposta ao mapa, executa o processamento de acordo com uma lógica definida, considerando os dados digitados pelo usuário.

A operação de exibir as informações das áreas dos

registros para a tela é denominada "mapout" e a operação de transferir os dados digitados na tela para as áreas dos registros, é denominada "mapin".

Para a definição do módulo fonte de um processo, pode ser utilizada uma combinação de comandos que definem, por exemplo, operações aritméticas, de alteração de mapa, de estruturas de seleção e iteração, para ter acesso à base de dados, etc... estabelecendo assim a lógica inerente a ele. Maiores detalhes sobre a sintaxe e utilização desses comandos podem ser encontrados em [C184a].

Após a definição do mapa e dos processos, deve-se gerar o diálogo. Para gerar um diálogo utiliza-se o AD SG. Deve ser preenchido um conjunto de telas contendo informações do nome do diálogo, do sub-esquema utilizado, do mapa associado, dos processos pré-mapa e de resposta ao mapa e a chave de controle associada ao processo. Maiores detalhes podem ser encontrados em [C183a, C184a].

Os erros que ocorrerem durante a compilação dos processos podem ser corrigidos utilizando os recursos disponíveis no IDD para processamento "on line". Esses mesmos recursos combinados ou não com o reprocessamento do sub-esquema, esquema ou mapa, podem ser utilizados quando houver necessidade de fazer alguma modificação em diálogos que já existam. Note-se no entanto, que neste último caso, não há necessidade de preencher o conjunto de telas acima referido, basta que sejam feitas as alterações necessárias e o diálogo regenerado, para que as alterações efetuadas tornem com efeito.

Para a execução das aplicações, o sistema de execução ADS/O carrega os módulos de carga exigidos, interpreta e processa os dados digitados no terminal, de acordo com as definições estabelecidas.

3.4 - A Linguagem de Consulta - OLQ

O acesso às informações armazenadas em estruturas IDMS, através de consultas pode ser feito utilizando a OLQ. Essas consultas muitas vezes apresentam tal nível de complexidade que exigem uma "navegação" pela estrutura, para que se possa obter as informações necessárias. A seguir, é brevemente apresentada a OLQ. Uma descrição mais detalhada pode ser encontrada no Apêndice II, bem como nas referências bibliográficas da seção 2.2.

O sistema conversacional On Line Query - OLQ - permite que se tenha acesso diretamente às informações de uma base de dados ou dicionário de dados, as quais podem ser formatadas em relatórios impressos ou exibidas diretamente na tela.

Para obter as informações necessárias, pode-se utilizar um dos seguintes tipos de comandos ou combinação deles: comandos de controle e ajuste, de recuperação de dados, de processamento de relatórios e de definição de caminhos e q-arquivos. Maiores detalhes sobre esses comandos podem ser encontrados em [C&83j]

Um caminho é constituído de vários comandos OLQ. A ordem em que eles são executados determina a "rota de busca" das informações em um sub-esquema de um registro mestre para um registro membro e vice-versa.

Um q-arquivo é constituído de vários comandos OLQ e de parâmetros e deve ser identificado por um nome pelo qual será referenciado posteriormente. Um q-arquivo pode ser catalogado no dicionário de dados, podendo assim ser referenciado e executado quantas vezes forem necessários, fornecendo os parâmetros convenientes para que melhor atendam as necessidades do usuário.

Caminhos e q-arquivos permitem ter acesso a diferentes tipos de registros, enquanto que os comandos de recuperação de dados permitem ter acesso a um único tipo de registro. As informações recuperadas, normalmente satisfazem algum critério especificado nos comandos de recuperação de dados.

Os comandos de processamento de relatórios, proporcionam uma maior flexibilidade para formatação de relatórios, pois têm disponíveis comandos que permitem efetuar cálculos so

bre determinados campos numéricos, ordenar a saída por um determinado campo, determinar o maior ou o menor valor, a média dos valores, o total, estabelecer pontos de quebra, etc...

A recuperação de informações pode ser feita uma a uma ou em conjunto. No primeiro caso, as informações são exibidas em linhas horizontais e no segundo caso, em colunas. Para ambos os casos, OLQ tem disponível comandos que permitem "avançar telas" para que o usuário tenha acesso às informações que não couberam numa única tela.

3.5 - Outros Recursos - OLE, CULPRIT

Uma característica de instrumental de quarta geração é oferecer recursos que permitam ao usuário final, obter as informações necessárias o mais independentemente possível. Assim, o IDMS tem disponível a OLE que é uma linguagem voltada para o usuário final. Tem também disponível o gerador de relatórios - CULPRIT, sendo que este exige o fornecimento de alguns parâmetros para a definição do relatório. Ambos são brevemente descritos a seguir.

On Line English - OLE é uma ferramenta IDMS que permite aos próprios usuários terem acesso diretamente aos dados, utilizando uma linguagem mais natural.

O vocabulário utilizado pelos usuários finais, está definido no "lexicon" sendo que uma parte desse vocabulário - "lexicon raiz" contém termos mais frequentemente utilizados, tais como: quanto, onde, quem, quais e é fornecida quando da instalação da OLE. A outra parte - "lexicon da aplicação", contém termos específicos da aplicação, tais como: nomes de arquivos, campos, agrupamentos de dados que serão utilizados pelo usuário final, permitindo assim um diálogo fluente entre o usuário e a OLE.

Na definição do "lexicon da aplicação", devem ser definidos os prováveis sinônimos para os nomes de campos que serão utilizados pelo usuário final. Assim, para uma aplicação que envolva o campo "empregado" por exemplo, seria interessante definir "funcionários", "pessoas" e "trabalhadores", como sinônimos. Um outro aspecto que poderia ser considerado é a definição de grupos. Assim, poderia por exemplo ser definido o grupo "lugar" como composto de "rua, cidade, estado" com o sinônimo "endereço". Essas considerações e outras que podem ser encontradas em maiores detalhes em [C831] dão maior flexibilidade para que o usuário construa as sentenças nas suas consultas.

O usuário final utiliza a OLE de forma interativa. Assim, quando é digitada no terminal uma consulta, a OLE tentará entendê-la e se todos os termos forem identificáveis, as informações serão recuperadas e exibidas ao usuário, no terminal. Por outro lado, se a consulta contiver termos não identificáveis, se

rão exibidas mensagens solicitando ao usuário uma complementação das informações até que todos os termos utilizados possam ser identificados e então possam ser recuperadas as informações desejadas.

Além das consultas tradicionais, o usuário tem disponível alguns processos identificados por palavras-chave definidos no "lexicon" que executam funções específicas. Através desses processos, pode ser obtida uma saída ordenada por um determinado campo ou executar processamento sobre campos numéricos. Tais processamentos permitem calcular percentagem, totais com quebras, cálculos estatísticos, etc.. Pode também ser especificado para que seja desenhado o gráfico de barras, tendo assim uma representação gráfica dos resultados obtidos.

CULPRIT é uma outra ferramenta IDMS que permite gerar relatórios que podem ser impressos ou não. A estrutura de um programa CULPRIT compreende de parâmetros para definição do perfil, da entrada, do processo, da saída, da área de trabalho e de campos com índice. Maiores detalhes sobre parâmetros e facilidades permitidas podem ser encontrados em [C83b].

Os parâmetros de definição do perfil, permitem especificar por exemplo, formato de impressão de "dump"; formato de impressão da hora de processamento; número de linhas/página; nome de usuário, etc...

Os parâmetros de definição de entrada, descrevem o arquivo de entrada, especificando por exemplo, tamanho do registro; tipo de arquivo; tamanho de bloco; os campos a serem utilizados por esse programa com os respectivos nomes, posição inicial, tamanho e posições decimais; critérios de seleção a serem considerados para cada registro de entrada, etc...

Os parâmetros de definição de saída, descrevem o relatório especificando por exemplo, tamanho do registro, tamanho de páginas; a indicação de que os campos serão ordenados ou não; caracter de edição para descrever linhas cabeçalho, detalhe e total.

Os parâmetros de definição de área de trabalho, definem as áreas de trabalho necessárias a um programa, podendo ser atribuídos valores iniciais a elas.

Os parâmetros de definição de índices, permitem descrever nomes de campos com índices que serão referenciados posteriormente durante o processamento.

Os dados aos quais um programa CULPRIT tem acesso, podem estar armazenados em arquivos convencionais ou em uma base de dados IDMS, podendo neste último caso, serem definidos caminhos e registros lógicos.

Um programa CULPRIT pode eventualmente fazer referências a arquivos de entrada que sejam constituídos de dados de mais de um arquivo, tendo sido assim definidos previamente. Uma outra facilidade permitida é a catalogação de módulos desenvolvidos pelos usuários, que quando desenvolvido em COBOL são tratados como sub-rotinas. São permitidas também catalogação de definição de saídas, processos frequentemente utilizados que podem ser copiados diretamente.

3.6 - Conclusões

Os recursos descritos neste capítulo, podem ser identificados em parte com as características apresentadas na Figura 3.02, adaptada de [Ma82].

	ADEQUADO PARA USUÁRIO FINAL	ADEQUADO PARA PROFISSIONAIS DE PD
Linguagens/Facilidade Consulta	OLQ/OLE	OLQ/ADS
Gerador de Relatórios	-	CULPRIT
Gerador de Aplicações	-	ADS
Linguagens Programação Altíssimo Nível	-	ADS

Figura 3.02 - Emprego mais Adequado das Ferramentas Discutidas

O dicionário de dados é sem dúvida uma fonte valiosa que o analista pode utilizar para armazenar informações a serem utilizadas na fase de análise e projeto de um sistema, como também posteriormente em consultas às informações necessárias.

Para o desenvolvimento de aplicações o ADS/O apresenta recursos que são de grande apoio. No entanto, o ADS/O é melhor aproveitado se utilizado por pessoas com algum conhecimento de processamento de dados uma vez que é exigida a especificação da lógica envolvida nos processos que compõem os diálogos.

A consulta às informações de uma base de dados ou dicionário de dados pode ser feita utilizando a OLQ. A definição de consultas mais complexas pode ser feita através de caminhos e q-arquivos e constitui-se numa boa ferramenta para projetistas e usuários.

O usuário final que queira obter informações por si, tem disponível a OLE que parece ser de fácil aprendizado. Nota-se no entanto uma restrição aos termos utilizados para consultas que são formuladas em inglês. Para um usuário totalmente leigo em inglês - ou qualquer outra língua que tivesse sido utiliza

da na definição do "lexicon" - exigiria que ele aprendesse, não só como utilizar o recurso, mas também a linguagem.

Já o gerador de relatórios CULPRIT, é uma ferramenta que pode aliviar o trabalho de programadores COBOL na especificação de espaçamento na impressão. Deve ser no entanto lembrada a exigência do fornecimento de parâmetros para um programa CULPRIT, solicitando assim do usuário algum conhecimento de processamento de dados.

Como pode ser visto na Figura 3.01, ainda existem outros "software" dessa família integrada que não foram aqui estudados, por fugirem do âmbito deste trabalho. No entanto é reconhecido que o estudo, bem como a sua utilização podem oferecer apoio aos analistas e usuários finais.

CAPÍTULO IV

UM EXEMPLO DE APLICAÇÃO DESSE INSTRUMENTAL

4.1 - Considerações Gerais

Este capítulo tem por objetivo apresentar a implementação de um sistema utilizando recursos do IDMS.

O Sistema Controle de Cotistas foi desenvolvido num bureau de prestação de serviço, utilizando a linguagem para desenvolvimento de aplicações "on line" - ADS/O, instalada no computador IBM-4341. Cabe no entanto lembrar que, como pode ser visto na Figura 4.01, o sistema possui também componentes de processamento por lotes que foram implementados utilizando a linguagem COBOL. Neste capítulo, o interesse maior é ilustrar apenas os módulos do processamento que utilizaram ADS/O. Assim, os módulos do processamento por lotes são apenas identificados e brevemente descritos.

No capítulo 3, foi apresentada uma idéia geral sobre o IDMS e os recursos disponíveis em tal ambiente.

Os detalhes relativos ao ADS/O e sua utilização são encontrados no Apêndice I ao final deste trabalho, bem como nas referências bibliográficas, na seção 2.2.

Uma avaliação da utilização dos recursos IDMS, bem como de métodos de análise e projeto de sistemas que poderiam ser utilizados, é apresentada no capítulo 5.

Assim, na seção 4.2 é apresentado o fluxo de informações do sistema, na 4.3 os componentes do processamento por lotes e imediato, na 4.4 exemplos de detalhamento de componentes de processamento imediato e a seção 4.5 finaliza o capítulo com as conclusões.

4.2 - O Fluxo de Informações do Sistema

Para um melhor entendimento do sistema controle de cotistas, como um todo, é mostrado na Figura 4.01 o diagrama que foi utilizado no desenvolvimento e que contém o fluxo do sistema

Esse diagrama contém componentes de processamento por lotes e imediato.

Alguns dos componentes de processamento imediato alimentam o sistema, com informações que são utilizadas quando da execução dos módulos de processamento por lotes. Outros desses componentes são utilizados apenas para atividades de consulta.

Muitas das informações necessárias aos usuários podem ser obtidas usando diretamente o diálogo correspondente, ficando a emissão de relatórios para atender aquelas atividades cujas informações são necessárias em épocas pré-determinadas, como por exemplo mensal, trimestral, semestral ou anualmente. Por outro lado, algum desses relatórios podem ser obtidos também quando é solicitado pelo usuário.

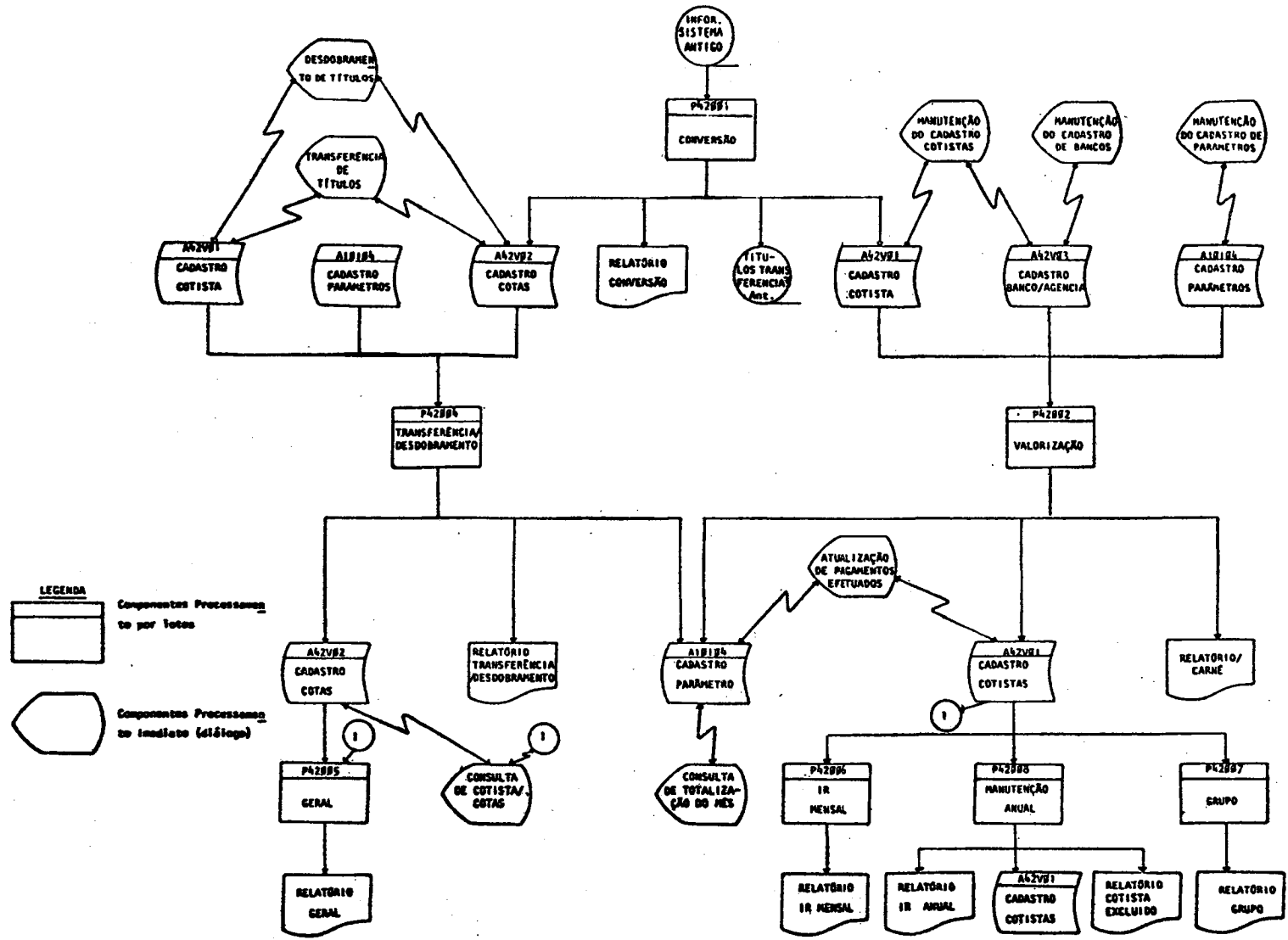


Figura 4.01 - Diagrama do Sistema Controle de Cotistas

4.3 - Componentes de Processamento por Lotes e Imediato

Como foi visto na seção 4.2, o sistema controle de cotistas acomoda componentes de processamento por lotes e imediato. Esse sistema, anteriormente já era automatizado, justificando assim a componente "conversão" na Figura 4.01. Porém agora, poderá atender muito mais rapidamente às necessidades dos usuários, haja visto a inclusão de diálogos que são descritos no decorrer desta seção.

Para a implementação dos componentes de processamento por lotes foi utilizada a linguagem COBOL e o dispositivo de acesso imediato VSAM.

Antes da breve descrição dos componentes de processamento, são apresentadas as convenções utilizadas para dar nomes aos programas, arquivos e diálogos, como se segue: Ass Vmm, Pss nnn, Dss fff. Assim, A, P e D identificam arquivo, programa e diálogo respectivamente; ss = sistema, V = VSAM, mm = número que identifica o arquivo no sistema, nnn = número que identifica o programa no sistema e fff = número que identifica a função que consta na Figura 4.02. O sistema aqui apresentado é identificado com o número 42, que é utilizado quer para os componentes de processamento por lotes como para os componentes de processamento imediato.

Os arquivos A42V01, A42V02, A42V03 e A10I04 são brevemente descritos a seguir:

- . A42V01 - Cadastro de Cotistas: contém informações cadastrais de cotistas: nome, endereço; valores acumulados: pagos, a pagar e Imposto de Renda; indicação de pessoa física/jurídica com o respectivo documento de identificação; banco; forma de pagamento; quantidade total de cotas e títulos; indicação se é cotista bloqueado com a data de bloqueio se for o caso; data e descrição de possíveis contactos que tenham sido efetuados se o cotista estiver envolvido em alguma transferência de título.

- . A42V02 - Cadastro de cotas: contém informações específicas de cotas, isto é, para cada cotista e título, a identificação do cotista, número do título, quantidade de cotas deste título, data do contrato, motivo do contrato e identificação do cotista anterior, número do título anterior e data do contrato anterior.

- . A42V03 - Cadastro de Banco: contém informações com o número de identificação de banco/agência e a descrição (nome) do banco/agência.
Na Figura 4.05 é ilustrada a estrutura desse arquivo.

- . A10I04 - Cadastro de Parâmetros: contém informações relativas ao valor da cota, limite de isenção do Imposto de Renda, valores resultantes da execução de P42002, P42004 e diálogos.

A seguir, é brevemente descrito cada componente de processamento por lotes:

- . P42001 - Programa executado para criação dos arquivos A42V01 - Cadastro de cotistas e A42V02 - cadastro de cotas na fase de conversão.

- . P42002 - Programa executado mensalmente, com base no valor da cota no mês corrente. É emitido relatório e/ou carnê com informações para efetuar pagamento ao cotista. São atualizados dos A42V01 e A10I04 com informações da valorização efetuada.

- . P42004 - Programa executado mensalmente. É emitido relatório onde são retratadas todas as transferências e desdobramentos de títulos que foram efetuados entre cotistas. É atualizado o A42V02 e A10I04.

- . P42005 - Programa utilizado para emissão do relatório geral, contendo todas as informações de cotas e cotistas.

- . P42006 - Programa executado mensalmente. É emitido relatório com informações de cotistas que tenham Imposto de Renda retido na fonte, com base na valorização efetuada.
- . P42007 - Programa executado para emitir relatório por grupo de cotistas.
- . P42008 - Programa executado anualmente para emitir relatório com informações sobre Imposto de Renda aos cotistas. Quando da execução deste programa serão excluídos de A42V01 aqueles cotistas que não mais possuírem cotas e nem saldo a receber. As informações desses cotistas constarão no relatório que será emitido.

Para cada um dos programas acima descritos, foram elaborados documentos de especificação de programas que foram então entregues aos programadores.

Os componentes de processamento imediato, aqui denominados funções constam na Figura 4.02, tendo a elas associados os respectivos diálogos.

A implementação dessas componentes foi feita utilizando os recursos de ADS/0, cujos detalhes podem ser encontrados na seção 3.3, no Apêndice I ao final deste trabalho e as referências bibliográficas a eles relevantes são citadas na seção 2.2.

Como dito anteriormente, quando da descrição dos componentes de processamento por lotes, os dados foram armazenados em arquivos VSAM. Embora não tenha havido preocupação em definir uma base de dados, foram feitas pelo administrador da base de dados - DBA, as considerações necessárias para a definição desses arquivos para o ambiente IDMS, para que pudesse ser utilizado o ADS/0.

Existem algumas funções que por decisão da fase de projeto, exibem um mapa para escolha da operação e um segundo mapa para que o usuário digite as informações necessárias. Outras funções não apresentam essa opção, por constituir-se de uma única operação a ser executada. Nesse sistema, exemplos do primeiro caso são as funções 01, 02 e 03, as demais são do segundo caso.

```

**                CONTROLE DE COTISTAS - SHOPPING                ***M42000**
----- SHOPPING      A      - SAO PAULO -----
***** FUNCOES *****
*
* 01 - MANUTENCAO DO CADASTRO COTISTAS      *
* 02 - MANUTENCAO DO CADASTRO DE BANCOS    *
* 03 - MANUTENCAO DO CADASTRO DE PARAMETROS *
* 04 - ATUALIZACAO DE PAGAMENTOS EFETUADOS *
* 05 - TRANSFERENCIA DE TITULOS           *
* 06 - DESDOBRAMENTO DE TITULOS           *
* 07 - CONSULTA DE COTISTA/COTAS         *
* 08 - CONSULTA DE TOTALIZACAO DO MES     *
FUNCAO? 00

```

Figura 4.02 - Funções Identificadas para o Sistema Controle de Cotistas.

A seguir, é brevemente descrito cada diálogo. Assim tem-se:

- . D42010 - Manutenção do cadastro de cotistas - compreende as operações de alteração, inclusão, consulta, bloqueio e desbloqueio que o usuário pode escolher conforme a conveniência, caracterizando assim o primeiro caso acima descrito.
- . D42020 - Manutenção do cadastro de Bancos - compreende as operações de inclusão, exclusão, alteração e consulta. É outro exemplo de função do primeiro caso acima descrito.
- . D42030 - Manutenção do cadastro de Parâmetros - compreende as operações de consulta, alteração e inclusão. Neste cadastro estão definidos diferentes tipos de registros, que são identificados por um número. São atualizados através desse diálogo: o valor da cota do mês, percentual e limite de isenção do Imposto de Renda. É através desse diálogo que o usuário tem acesso aos valores de cotas dos últimos onze meses.
- . D42041 - Atualização de Pagamentos efetuados - esse diálogo atualiza os pagamentos efetuados. Esses pagamentos referem-se àqueles efetuados aos cotistas de acordo com o valor da cota do mês e da quantidade de cotas que possuem. Esse valor foi determinado quando da execução de P42002 e pode eventualmente referir-se a um valor acumulado de vários meses. Esse diálogo não possui o mapa para seleção de operações, pois foi definido para uma única operação. A cada pagamento efetuado são atualizados A42V01 e A10I04.
- . D42051 - Transferência de títulos - Esse diálogo é utilizado para efetuar cada transferência realizada no mês e não possui mapa de operações. São digitados o código do cotista antigo e o número do título. O sistema exibe o nome do cotista, a quantidade de cotas e o código de bloqueio. A seguir, se puder prosseguir, são digitados o código do novo cotista, data do contrato e motivo do contrato. Estas informações são armazenadas em A42V02.

- . D42061 - Desdobramento de Títulos: Esse diálogo é utilizado quando ocorre desdobramento de um título. Deve ser digitado o código de cotista, o número do título que será desdobrado e os números dos títulos resultantes do desdobramento. Esse diálogo não possui mapa de operações. As informações são armazenadas em A42V02.

- . D42071 - Consulta de cotista/cotas: Esse diálogo é utilizado para consulta. Quando de sua utilização, podem ser obtidas informações cadastrais do cotista e informações dos títulos que possui. Se um determinado cotista possuir uma grande quantidade de títulos, pode-se "avançar" telas com as informações do próximo conjunto de títulos, que é possível exibir na tela. É permitido também obter, se necessário, informações a partir de um determinado título. O usuário deve digitar o código de cotista e o número do título desejado.

- . D42081 - Consulta de totalização mês: Esse diálogo é utilizado exclusivamente para consulta sendo de grande valia para o pessoal administrativo, uma vez que apresenta informações de valores e títulos o mais atualizado possível. Esses valores são recuperados do cadastro A10I04 que foram armazenados quando da execução de P42002 (para valores), D42041 (pagamento efetuados) e P42004 (transferência/desdobramentos do mês).

4.4 - Exemplos de Detalhamento de Componentes de Processamento Imediato

Os diálogos descritos na seção 4.3 são aqui parcialmente ilustrados.

Os diálogos são identificados por Dss fff, o processo pré-mapa por PM-Dss fff, os processos de resposta ao mapa por RP-Dss fff - pp e o mapa a eles associados por Mss fff. As convenções para ss, fff são como anteriormente descritos. O sufixo pp identifica o processo de resposta ao mapa no diálogo.

O usuário que for utilizar o sistema deve digitar ADS e selecionar o sistema ao qual sua senha permite que tenha acesso. Essa é uma medida de segurança necessária, para que pessoas não autorizadas não alterem indevidamente as informações.

Após selecionar o sistema, o usuário seleciona a função desejada no "menu" ilustrado na Figura 4.02.

Aos diálogos que possuírem mapa de operação, o mapa exibido ao usuário independe de operação escolhida, pois os campos a serem liberados para preenchimento para cada operação estão definidos nas sub-rotinas correspondentes.

O diálogo Manutenção do Cadastro de Bancos, é um exemplo em que é exibido mapa de operações. Assim, ao usuário que selecionar esta função, será exibido primeiramente o mapa ilustrado na Figura 4.03.

O usuário pode então escolher uma das operações, quando então será exibido o mapa ilustrado na Figura 4.04.

O usuário preencherá os campos correspondentes dependendo da operação escolhida. Assim, se a opção for inclusão, devem ser preenchidos todos os campos necessários, isto é, número de identificação banco/agência e o nome do banco/agência. Já no caso de uma consulta, basta preencher o número de identificação banco/agência e será exibido o nome correspondente.

Os bancos e agências cadastrados através deste diálogo, são aqueles que os cotistas utilizam para recebimento do pagamento da valorização efetuada. Uma agência não pode ser inserida enquanto o banco não tiver sido inserido. Analogamente, um banco não pode ser excluído enquanto houver pelo menos

```

**          **          MANUTENCAO BANCO/AGENCIA          ***M42020**
----- SHOPPING      A      - SAO PAULO -----
          **** OPERACAO ****
          *
          * 01 - INCLUSAO *
          *
          * 02 - ALTERACAO *
          *
          * 03 - EXCLUSAO *
          *
          * 04 - CONSULTA *
OPERACAO? 00

```

Figura 4.03 - Mapa com as Operações disponíveis para o Diálogo D42020

```

**          **          MANUTENCAO BANCO/AGENCIA          (CONSULTA)          ***M42021**
----- SHOPPING      A      - SAO PAULO -----
BANCO/AGENCIA: 0000 0000
NOME.....:

```

Figura 4.04 - Mapa do Diálogo - D42021

uma agência.

Como pode ser visto no Apêndice I, os registros - dos arquivos e de trabalho - a serem utilizados pelo mapa as sociado ao diálogo, devem estar definidos no dicionário de dados. A Figura 4.05, ilustra os registros necessários para este diálogo. Esta figura ilustra também os códigos de condição de estado que podem ser utilizados nos processos.

A Figura 4.06 ilustra o processo pré-mapa para es te diálogo.

A Figura 4.07 ilustra os processos de resposta ao mapa correspondentes ao gerenciamento de diálogos, comuns a to dos os diálogos desse tipo.

A Figura 4.08 ilustra o processo de resposta ao ma pa correspondente ao diálogo ilustrado. As operações válidas pa ra este diálogo conforme ilustrado na Figura 4.03, são definidas em forma de sub-rotinas. Assim, a Figura 4.08a ilustra a sub-ro tina correspondente à operação inclusão - R-INCLUS. Por sua vez, essa sub-rotina chama as sub-rotinas R-PREBAG e R-CONBAG que es tão ilustradas na Figura 4.08b e 4.08c respectivamente. Analoga mente, definem-se as rotinas correspondentes às demais operações que não são aqui ilustradas.

A Figura 4.09 ilustra o processo de resposta ao ma pa que define a restauração do processamento efetuado.

Uma segunda ilustração de detalhamento compreende o diálogo D42051. Esse diálogo como pode ser visto na seção 4.3, não apresenta mapa de operação.

Assim, ao usuário que selecionar esta função, será exibido de imediato o mapa ilustrado na Figura 4.10, onde então devem ser preenchidas as informações necessárias.

A Figura 4.11 ilustra o processo de resposta ao ma pa.

Esta função não possui mapa de operações sendo, por tanto, imediatamente chamada a sub-rotina que executa a função principal, que está ilustrado na Figura 4.11a.

AUTO-STATUS RECORD... ADSO-STAT-DEF-REC VERS. 0001 AUTO-STATUS

02	ERROR-STATUS	PIC 9(4).
88	DB-STATUS-DK	VALUE '0000'.
88	DB-END-OF-SET	VALUE '0307'.
88	DB-REC-NOT-FOUND	VALUE '0326'.
88	QUEUE-REC-NOT-FOUND	VALUE '4405'.
88	SCRATCH-REC-NOT-FOUND	VALUE '4305'.
88	SCRATCH-REC-REPLACED	VALUE '4317'.
88	DB-END-OF-INDEX	VALUE '1707'.
88	DB-INDEX-NOT-FOUND	VALUE '1726'.
88	QUEUE-ID-NOT-FOUND	VALUE '4404'.
88	SCRATCH-AREA-NOT-FOUND	VALUE '6303'.
88	DB-ANY-ERRDR	VALUE '0001' THRU '9999'.

RECORD... REG-WS-1002 VERS. 0001 MAP

03	RG-PARAMETROS.	
05	COO-EMPRESA	PIC X.
05	NOME-EMPRESA	PIC X(30).
05	INDICE-FUNCAO	PIC 99.
05	COO-OPERACAO	PIC 99.
05	COO-APLICACAO	PIC 99.
05	COO-FUNCAO	PIC 99.
03	NEXT-DIALOGOS-02.	
05	DIALOGO-GERENCIADOR	PIC X(08).
05	FUNCAO-DIALOGO	
	OCURS 20 TIMES.	
07	COO-FUNCAOX-D.	
09	COO-FUNCAO-D	PIC 99.
07	COO-DIALOGO	PIC X(8)
	OCURS 4 TIMES.	
03	SISTEMAS-EMPRESA	PIC X(20).
03	SISTEMAS-R REDEFINES SISTEMAS-EMPRESA	
	OCURS 10 TIMES	PIC 99.
03	INDICES-GERAIS.	
05	INDEX1	COMP-3 PIC 9999.
05	INDEX2	COMP-3 PIC 9999.
05	INDEX3	COMP-3 PIC 9999.
03	DECOO-OPERACOES-02.	
04	DECOO-ST-02	PIC 99.
04	DECOO-FN-02	PIC 99.
04	DECOO-OP-02	PIC 99.
03	FLAG-ATUALIZACAO	PIC X.
88	ATUALIZA	VALUE '1'.
88	NAOATUALIZA	VALUE '0'.
03	FLAG-COMMIT	PIC X.
88	CHECKPOINT	VALUE '1'.
88	NAOCHECKPOINT	VALUE '0'.
03	FLAG-PEDIDO	PIC X.
88	DIALOGO-SUPERIOR	VALUE '1'.
88	DIALOGO-INFERIOR	VALUE '0'.
03	INDEX4	COMP-3 PIC 9(5).
03	INDEX5	COMP-3 PIC 9(5).
03	INDEX6	COMP-3 PIC 9(5).

RECORD... REG-A42V03-2110 VERS. 0001 MAP/SS

02	CHAVE-42V03.	
03	EMPRESA-42V03	PIC 9(04).
03	GRUPO-42V03	PIC 9(02).
03	BCO-42V03	PIC 9(04).
03	AGE-42V03	PIC 9(04).
02	DESCR-42V03	PIC X(30).
02	FILLER	PIC X(10).

RECORD... REG-WS-4202 VERS. 0001 MAP

02	CHAVE-WS-4202.	
03	BCO	PIC 9(04).
03	AGE	PIC 9(04).

Figura 4.05 - Registros Necessários para a Execução de D42021

PREMAP PROCESS PH-D42021

VERS. 0001

MODULE DESCRIPTION

DATE CREATED 05/14/85 BY USER ZSILVIA

DATE LAST UPDATED 05/14/85 BY USER ZMARTA

00000100	INITIALIZE (REG-A42V03-2110).
00000200	MOVE ZEROS TO BCO.
00000300	MOVE ZEROS TO AGE.
00000400	DELETE SCRATCH ALL.
00000500	PROTECT ALL FIELDS PERMANENT.
00000600	UNPROTECT (AGE BCO) TEMPORARY.
00000700	MOVE COD-APLICACAO TO DECOD-ST-02.
00000800	MOVE COD-FUNCAO TO DECOD-FN-02.
00000900	MOVE COD-OPERACAO TO DECOD-OP-02.
00001000	MOVE '0' TO FLAG-ATUALIZACAO.
00001100	MOVE '0' TO FLAG-COMMIT.
00001200	DISPLAY.

Figura 4.06 - Processo Pré-mapa Associado ao Diálogo de D42021

RESPONSE PROCESS RP-D10000-01
EXECUTE ON EDIT ERRORS YES

VERS. 0001

KEY. PFI

MODULE DESCRIPTION RESPONSE DE RETORNO AO M A I N L I N E

DATE CREATED 03/13/84 BY USER

DATE LAST UPDATED 12/04/84 BY USER CULL OBA

00000100	DELETE SCRATCH ALL.
00000200	KEEP LONG ALL RELEASE.
00000300	LEAVE APPLICATION.

RESPONSE PROCESS RP-D10000-07
EXECUTE ON EDIT ERRORS YES

VERS. 0001

KEY. CLEAR

MODULE DESCRIPTION

DATE CREATED 03/13/84 BY USER

DATE LAST UPDATED 11/30/84 BY USER ZMARIO

00000100	DELETE SCRATCH ALL.
00000200	KEEP LONG ALL RELEASE.
00000300	RETURN TO COD-DIALOGO (INDICE-FUNCAO, 1) CONTINUE.

Figura 4.07 - Processos de Resposta ao Mapa para Gerenciamento de Diálogos.

EXECUTE ON EDIT ERRORS NO

MODULE DESCRIPTION

DATE CREATED 05/13/85 BY USER ZELISA

DATE LAST UPDATED 05/28/85 BY USER ZELISA

```

00000100      READY AREA-A42V03 USAGE-MODE IS UPDATE.
00000200      IF CHECKPOINT:
00000300          DO.
00000400          KEEP LONG ALL RELEASE.
00000500          DELETE SCRATCH ALL.
00000600          MOVE '0' TO FLAG-COMMIT.
00000700      END.
00000800      IF COD-OPERACAO EQ 1
00000900          CALL R-INCLUS.
00001000      IF COD-OPERACAO EQ 2
00001100          CALL R-ALTERA.
00001200      IF COD-OPERACAO EQ 3
00001300          CALL R-EXCLUS.
00001400      IF COD-OPERACAO EQ 4
00001500          CALL R-CONSUL.
00001600      DISPLAY MESSAGE TEXT IS
00001700          *** 00 - TIPO OPERACAO INDEFINIDO -TECLE CLEAR*.
00001800      *****

```

Figura 4.08 - Processo de Resposta ao Mapa com as Possíveis Operações

```

00001900      DEFINE SUBROUTINE R-INCLUS.
00002000      IF NAQATUALIZA
00002100          DO.
00002200          CALL R-PRERAG.
00002300          MOVE 1 TO FLAG-ATUALIZACAO.
00002400          UNPROTECT ALL EXCEPT ( BCD AGE
00002500              NOME-EMPRESA DECOD-OPERACOES-021
00002600              TEMPORARY.
00002700          MODIFY MAP PERMANENT FOR FIELD DESCR-42V03 REQUIRED.
00002800          MODIFY MAP TEMPORARY CURSOR AT FIELD DESCR-42V03.
00002900          DISPLAY.
00003000      END.
00003100      CALL R-CONBAG.
00003200      IF ANY FIELDS ARE IN ERROR
00003300          DISPLAY.
00003400      MOVE 0 TO FLAG-ATUALIZACAO.
00003500      MOVE AGE TO AGE-42V03.
00003600      STORE REG-A42V03-2110.
00003700      MOVE 1 TO FLAG-COMMIT.
00003800      UNPROTECT (BCD AGE ) TEMPORARY.
00003900      MODIFY MAP FOR FIELD DESCR-42V03 OPTIONAL.
00004000      DISPLAY MESSAGE CODE 900051.
00004100      GOBACK.
00004200      *****

```

Figura 4.08.a - Sub-rotina Correspondente à Inclusão R-INCLUS.

```

00008600 DEFINE SUBROUTINE R-PREBAG.
00008700 INITIALIZE (REG-A42V03-2110).
00008800 IF BCO EQ ZEROS
00008900 DO.
00009000     MODIFY MAP FOR (BCO AGE) EDIT IS ERROR.
00009100     DISPLAY MESSAGE TEXT IS
00009200     *** 05 - FALTA INFORMACAO
00009300     END.
00009400     MOVE 0222 TO EMPRESA-42V03.
00009500     MOVE BCO TO BCO-42V03.
00009600     MOVE AGE TO AGE-42V03.
00009700     MOVE ZEROS TO GRUPO-42V03.
00009800     FIND KEEP EXCLUSIVE CALC REG-A42V03-2110.
00009900     IF COD-OPERACAO EQ 4
00010000         OBTAIN CALC REG-A42V03-2110.
00010100     ELSE
00010200         OBTAIN KEEP EXCLUSIVE CALC REG-A42V03-2110.
00010300     IF DB-STATUS-OK
00010400         DO.
00010500             IF COD-OPERACAO EQ 1
00010600                 DO.
00010700                     MODIFY MAP FOR (BCO AGE) EDIT IS ERROR.
00010800                     DISPLAY MESSAGE TEXT IS
00010900                     *** 01 - CODIGO EXISTENTE
00011000                     END.
00011100                 IF COD-OPERACAO EQ 3 AND
00011200                     AGE EQ ZEROS
00011300                     DO.
00011400                         OBTAIN NEXT REG-A42V03-2110 WITHIN
00011500                         SET-A42V03 ALLOWING (DB-END-OF-SET).
00011600                         IF DB-END-OF-SET
00011700                             GOBACK.
00011800                         IF BCO EQ BCO-42V03
00011900                             DO.
00012000                                 MODIFY MAP FOR (BCO AGE) EDIT IS ERROR.
00012100                                 DISPLAY MESSAGE TEXT IS
00012200                                 *** 05 AINDA EXISTE AGENCIA
00012300                                 END.
00012400                             END.
00012500                         END.
00012600                     ELSE
00012700                         DO.
00012800
00012900                             IF COD-OPERACAO EQ 1 AND
00013000                                 AGE NE ZEROS
00013100                                 DO.
00013200                                     MOVE 0222 TO EMPRESA-42V03.
00013300                                     MOVE BCO TO BCO-42V03.
00013400                                     MOVE ZEROS TO AGE-42V03.
00013500                                     MOVE ZEROS TO GRUPO-42V03.
00013600                                     FIND CALC REG-A42V03-2110
00013700                                     ALLOWING (DB-REC-NOT-FOUND).
00013800                                     IF DB-REC-NOT-FOUND
00013900                                         DO.
00014000                                             MODIFY MAP FOR (BCO AGE) EDIT IS ERROR.
00014100                                             DISPLAY MESSAGE TEXT IS
00014200                                             *** 03 BANCO NAO CADASTRADO
00014300                                             END.
00014400                                         END.
00014500                                     IF COD-OPERACAO NE 1
00014600                                         DO.
00014700                                             MODIFY MAP FOR (BCO AGE) EDIT IS ERROR.
00014800                                             DISPLAY MESSAGE TEXT IS
00014900                                             *** 05 CODIGO INEXISTENTE
00015000                                             END.
00015100                                         END.
00015200                                     END.
00015300                                 GOBACK.
00015400                                 *****

```

Figura 4.08.b - Sub-rotina Chamada pela Rotina R-INCLUS.

```

00015400 *****
00015500 DEFINE SUBROUTINE R-CONBAG.
00015600 IF DESCR-42V03 EQ SPACE OR
00015700 DESCR-42V03 EQ ' '
00015800 MODIFY MAP FOR FIELD DESCR-42V03 EDIT IS ERROR.
00015900
00016000 GOBACK.
00016100 *****

```

Figura 4.08c - Esta é a outra Rotina Chamada por R-INCLUS.

```

RESPONSE PROCESS RP-042021-02 VERS. 0001 KEY. PF7
EXECUTE DN EDIT ERRORS NO

MODULE DESCRIPTION
DATE CREATED 05/14/85 BY USER ZSILVIA
DATE LAST UPDATED BY USER

00000100 READY AREA-A42V03 USAGE-MODE IS UPDATE.
00000200 UNPROTECT (BCD AGE) TEMPORARY.
00000300 IF NAOCHECKPOINT
00000400 DO.
00000500 INITIALIZE (REG-A42V03-2110).
00000600 MOVE '0' TO FLAG-ATUALIZACAO.
00000700 MODIFY MAP FOR FIELD DESCR-42V03
00000800 EDIT IS CORRECT OPTIONAL.
00000900 KEEP LONG ALL RELEASE.
00001000 DISPLAY MESSAGE CODE IS 900051.
00001100 END.
00001200 IF COD-OPERACAO EQ 1
00001300 DO.
00001400 ERASE REG-A42V03-2110.
00001500 CALL R-SAIDA.
00001600 END.
00001700 GET SCRATCH INTO REG-A42V03-2110 MAX LENGTH 60 FIRST.
00001800 IF COD-OPERACAO EQ 3
00001900 DO.
00002000 STORE REG-A42V03-2110.
00002100 CALL R-SAIDA.
00002200 END.
00002300
00002400 MODIFY REG-A42V03-2110.
00002500 CALL R-SAIDA.

```

Figura 4.09 - Processo de Resposta ao Mapa - RP - D42021-02

```

**                **                TRANSFERENCIAS                **M42051**
----- SHOPPING      A      - SAO PAULO -----
EMPR.....: 00
COTISTA ...: 000000
TITULO ....: 0-0-00000
***** DADOS INFORMATIVOS DO CEDENTE *****
NOME.....:
NUMERO COTAS .:      0      BLOQUEIO .....: 0
***** DADOS INFORMATIVOS DO CESSIONARIO/CONTRATO *****
CODIGO .....: 000000
DATA CONTRATO: 00/00/00      MOTIVO CONTRATO:

```

Figura 4.10 - Mapa Correspondente ao Diálogo - D42051

```

RESPONSE PROCESS RP-D42051-01 ..... VERS. 0001 ..... KEY. ENTER
EXECUTE ON EDIT ERRORS NO
MODULE DESCRIPTION
DATE CREATED 05/24/85 BY USER ZELISA
DATE LAST UPDATED 06/19/85 BY USER ZELISA
00000100 ..... READY AREA-A42V01 USAGE-MODE IS UPDATE.
00000200 ..... READY AREA-A42V02 USAGE-MODE IS UPDATE.
00000300 ..... IF CHECKPOINT
00000400 ..... DO.
00000500 ..... KEEP LONG ALL RELEASE.
00000600 ..... DELETE SCRATCH ALL.
00000700 ..... DELETE SCRATCH AREA ID '10100' ALL.
00000800 ..... DELETE SCRATCH AREA ID '10200' ALL.
00000900 ..... MOVE '0' TO FLAG-COMMIT.
00001000 ..... END.
00001100 ..... CALL R-TRANSE.
00001200 *****

```

Figura 4.11 - Processo de resposta ao mapa - RP - D42051 - 01


```

00001300 DEFINE SUBROUTINE R-TRANSF.
00001400 IF NA0ATUALIZA
00001500 DO.
00001600 CALL R-PRETRA.
00001700 MOVE '1' TO FLAG-ATUALIZACAO.
00001800 UNPROTECT ALL EXCEPT (GRUPO COD-COTI TITU-AUX
00001900 QT-COTA-42V02 BLD0-42V01
00002000 NOME-EMPRESA NOME-AUX (1)
00002100 NOME-AUX (2)) TEMPORARY.
00002200 MODIFY MAP PERMANENT FOR (DT-CONTR MOT-CONTR
00002300 COD-COTI-NOVO)
00002400 REQUIRED.
00002500 MODIFY MAP TEMPORARY CURSOR AT FIELD COD-COTI-NOVO.
00002600 DISPLAY.
00002700 END.
00002800 CALL R-CONTRA.
00002900 IF ANY FIELDS ARE IN ERROR
00003000 DISPLAY.
00003100 MOVE 0222 TO EMPRESA-42V02.
00003200 MOVE GRUPO TO GRUPO-42V02.
00003300 MOVE SER-AUX TO SER-42V02.
00003400 MOVE FASE-AUX TO FASE-42V02.
00003500 MOVE CONTR-AUX TO CONTR-42V02.
00003600 MOVE COD-COTI TO COD-COTI-42V02.
00003700 FIND KEEP EXCLUSIVE CALC REG-A42V02-2109.
00003800 OBTAIN CURRENT REG-A42V02-2109.
00003900 PUT SCRATCH AREA ID '10200' FROM REG-A42V02-2109 LENGTH 60.
00004000 MOVE QT-COTA-42V02 TO QT-COTA-AUX.
00004100 MOVE ZEROS TO QT-COTA-42V02.
00004200 MODIFY REG-A42V02-2109.
00004300 MOVE DIA-CONTR-42V02 TO DIA-AUX-TRANS.
00004400 MOVE MES-CONTR-42V02 TO MES-AUX-TRANS.
00004500 MOVE ANO-CONTR-42V02 TO ANO-AUX-TRANS.
00004600 MOVE 0222 TO EMPRESA-42V02.
00004700 MOVE GRUPO TO GRUPO-42V02.
00004800 MOVE COD-COTI-NOVO TO COD-COTI-42V02.
00004900 MOVE SER-AUX TO SER-42V02.
00005000 MOVE FASE-AUX TO FASE-42V02.
00005100 MOVE CONTR-AUX TO CONTR-42V02.
00005200 MOVE QT-COTA-AUX TO QT-COTA-42V02.
00005300 MOVE DIA-CONTR TO DIA-CONTR-42V02.
00005400 MOVE MES-CONTR TO MES-CONTR-42V02.
00005500 MOVE ANO-CONTR TO ANO-CONTR-42V02.
00005600 MOVE COD-COTI TO COD-COTI-ANT-42V02.
00005700 MOVE CONTR-AUX TO CONTR-ANT-42V02.
00005800 MOVE SER-AUX TO SER-ANT-42V02.
00005900 MOVE FASE-AUX TO FASE-ANT-42V02.
00006000 MOVE DIA-AUX-TRANS TO DIA-CONTR-ANT-42V02.
00006100 MOVE MES-AUX-TRANS TO MES-CONTR-ANT-42V02.
00006200 MOVE ANO-AUX-TRANS TO ANO-CONTR-ANT-42V02.
00006300 MOVE '0' TO FLAG-ATUALIZACAO.
00006400 MOVE MOT-CONTR TO MOT-CONTR-42V02.
00006500 STORE REG-A42V02-2109.
00006600 PUT SCRATCH AREA ID '10200' FROM REG-A42V02-2109 LENGTH 60.
00006700 MOVE 0222 TO EMPRESA-42V01.
00006800 MOVE GRUPO TO GRUPO-42V01.
00006900 MOVE COD-COTI TO COD-COTI-42V01.
00007000 FIND KEEP EXCLUSIVE CALC REG-A42V01-2108.
00007100 OBTAIN CALC REG-A42V01-2108.
00007200 PUT SCRATCH AREA ID '10100' FROM REG-A42V01-2108 LENGTH 450.
00007300 COMPUTE QT-TITU-42V01 = QT-TITU-42V01 - 1.
00007400 COMPUTE QT-COTA-42V01 = QT-COTA-42V01 - QT-COTA-AUX.
00007500 MODIFY REG-A42V01-2108.
00007600 MOVE 0222 TO EMPRESA-42V01.
00007700 MOVE GRUPO TO GRUPO-42V01.
00007800 MOVE COD-COTI-NOVO TO COD-COTI-42V01.
00007900 FIND KEEP EXCLUSIVE CALC REG-A42V01-2108.
00008000 OBTAIN CALC REG-A42V01-2108.
00008100 PUT SCRATCH AREA ID '10100' FROM REG-A42V01-2108 LENGTH 450.
00008200 COMPUTE QT-TITU-42V01 = QT-TITU-42V01 + 1.
00008300 COMPUTE QT-COTA-42V01 = QT-COTA-42V01 + QT-COTA-AUX.
00008400 MODIFY REG-A42V01-2108.
00008500 MOVE '1' TO FLAG-COMMIT.
00008600 UNPROTECT (TITU-AUX GRUPO COD-COTI) TEMPORARY.
00008700 MODIFY MAP PERMANENT FOR (DT-CONTR MOT-CONTR
00008800 COD-COTI-NOVO) OPTIONAL.
00008900 DISPLAY MESSAGE CODE IS 900051.
00009000 GOBACK.

```

Figura 4.11.a - Rotina R-TRANSF para Executar a Função Principal de D42051.

Note-se no entanto que nada impede que esta sub-rotina chame outras, no caso por exemplo a sub-rotina R-CONTRA i lustrada na Figura 4.11b. Essa sub-rotina por sua vez chama a sub-rotina R-CONDAT que está ilustrada na Figura 4.11c. Esta estrutura de módulos pode ser melhor vista no diagrama de estrutura mostrado no capítulo 5, na Figura 5.09.

Cabe no entanto lembrar que este diálogo tem ainda como componentes os processos pré-mapa e de resposta do mapa, que define a restauração do processamento efetuado. No entanto elas não foram aqui ilustradas por serem de definição análoga às aquelas ilustradas na Figura 4.06 e Figura 4.09, exceto no que diz respeito à lógica inerente a cada uma. Da mesma forma, não estão ilustrados também os registros associados ao mapa.

```

00016000 *****
00016100 DEFINE SUBROUTINE R-CONTRA.
00016200     MOVE Q222      TO EMPRESA-42V01.
00016300     MOVE GRUPO    TO GRUPO-42V01.
00016400     MOVE COD-COTI-NOVO TO COD-COTI-42V01.
00016500     OBTAIN CALC REG-A42V01-2108.
00016600     IF DB-REC-NOT-FOUND
00016700         DO.
00016800             MODIFY MAP FOR FIELD COD-COTI-NOVO EDIT IS ERROR.
00016900             DISPLAY MESSAGE TEXT IS
00017000             '** 05 - CESSIONARIO NAO CADASTRADO
00017100         END.
00017200     MOVE NOME-COTI-42V01 TO NOME-AUX (2).
00017300     MOVE DIA-CONTR TO DIA-AUX.
00017400     MOVE MES-CONTR TO MES-AUX.
00017500     MOVE ANO-CONTR TO ANO-AUX.
00017600     CALL R-CONDAT.
00017700     IF INDEX1 EQ 1
00017800         MODIFY MAP FOR FIELD DT-CONTR EDIT IS ERROR.
00017900     MOVE DATE TO DT-AUX.
00018000     IF ANO-CONTR LT ANO-AUX AND
00018100         MES-CONTR GT MES-AUX
00018200         MODIFY MAP FOR FIELD DT-CONTR EDIT IS ERROR.
00018300     IF MOT-CONTR NE 'C' AND
00018400         MOT-CONTR NE 'A'
00018500         MODIFY MAP FOR FIELD MOT-CONTR EDIT IS ERROR.
00018600     IF COD-COTI-NOVO EQ ZEROS
00018700         MODIFY MAP FOR FIELD COD-COTI-NOVO EDIT IS ERROR.
00018800     GOBACK.
00018900 *****

```

Figura 4.11.b - Rotina Chamada por R-TRANSF

```

00019000 DEFINE SUBROUTINE R-CONDAT.
00019100     MOVE ZEROS TO INDEX1.
00019200     IF MES-AUX EQ 2
00019300         DO.
00019400             COMPUTE INDEX2 = ANO-AUX / 4.
00019500             COMPUTE INDEX2 = INDEX2 * 4.
00019600             IF INDEX2 EQ ANO-AUX
00019700                 MOVE 29 TO ULTM-DIA (2).
00019800             ELSE
00019900                 MOVE 28 TO ULTM-DIA (2).
00020000         END.
00020100     IF MES-AUX EQ ZERO OR
00020200         MES-AUX GT 12 OR
00020300         DIA-AUX EQ ZERO
00020400         DO.
00020500             MOVE 1 TO INDEX1.
00020600             GOBACK.
00020700         END.
00020800     IF DIA-AUX GT ULTM-DIA (MES-AUX)
00020900         MOVE 1 TO INDEX1.
00021000     GOBACK.
00021100 *****

```

Figura 4.11.c - Rotina Chamada por R-CONTRA

4.5 - Conclusões

Face à implementação efetuada utilizando ADS/O, pode-se dizer que ele constitui um bom instrumento para desenvolvimento de aplicações "on line".

Este sistema como pode ser visto na seção 4.1, apresenta componentes de processamento por lotes e imediato. Apesar da presença dos componentes de processamento por lotes, observou-se que o sistema como um todo foi desenvolvido em um tempo relativamente pequeno. É claro que a variável tempo não deve ser considerada sozinha, deve-se considerar também a complexidade do sistema.

Um aspecto que pode ser abordado, é a facilidade e rapidez com que podem ser efetuadas alterações nos diálogos existentes, tanto após quanto durante o projeto, como pode ser visto no Apêndice I. O processo para gerar diálogos é bastante rápido, não necessitando recompilação de toda a estrutura envolvida, como foi visto em [Ru85] que exige a recompilação inclusive da estrutura de dados.

Para definição de mapas associados aos diálogos, tem-se bastante flexibilidade, uma vez que a tela é exibida "limpa" ao projetista que então "desenha" de acordo com as necessidades. O usuário poderia inclusive acompanhar esse processo de "desenho", pois ele poderia dar as sugestões que se fizerem necessário para que melhor sejam atendidas às exigências do sistema.

As chaves de controle (PF) que estão associadas aos processos são também de grande valia, pois o usuário não precisa aprender comandos complicados para executar um diálogo, basta que ele pressione a chave correspondente.

Como pode ser visto nos exemplos, os processos são compostos de sub-rotinas que definem uma estrutura modular, tornando-os mais claros.

CAPÍTULO V

AVALIAÇÃO DO DESENVOLVIMENTO EFETUADO

5.1 - Considerações Gerais

Este capítulo tem por objetivo apresentar uma avaliação da utilização de recursos de métodos de análise e projeto de sistemas, linguagem de consulta e recursos de base de dados, que podem ser utilizados no desenvolvimento de sistemas. A avaliação é feita em relação ao sistema cuja implementação foi parcialmente ilustrada no capítulo 4.

Uma idéia geral sobre linguagem de consulta foi apresentada no Capítulo 3, maiores detalhes podem ser encontrados nas referências bibliográficas citadas na seção 2.2 e também no Apêndice II, ao final deste trabalho. Ainda na seção 2.2 podem ser encontradas referências bibliográficas referentes aos recursos de base de dados - inclusive sugestão de diretrizes para o projeto de uma base de dados. No que se refere à métodos de análise e projeto, não foi dedicado no trabalho capítulo para descrevê-los. No entanto, é oportuna a avaliação aqui apresentada, uma vez que sendo dedicada a atenção merecida nas fases de análise e projeto, principalmente para sistemas complexos, os erros da especificações poderão ser detectados antecipadamente, contribuindo assim para a obtenção de um sistema que melhor atenda as necessidades do usuário. As referências bibliográficas dos métodos objetos de avaliação neste capítulo, são citadas na seção 2.3. Cabe, no entanto, lembrar que estes não são os únicos métodos existentes.

Assim, na seção 5.2 é apresentada uma avaliação dos métodos de análise e projeto de sistemas, na 5.3 da linguagem de consulta, na 5.4 dos recursos de base de dados e na seção 5.5 as conclusões que finalizam o capítulo.

5.2 - Sobre o Aproveitamento do Potencial dos Métodos de Análise e Projeto de Sistemas

Na implementação do Sistema Controle de Cotistas foram definidos como componentes do processamento imediato, os diálogos com as respectivas rotinas que constam na Tabela 5.01.

No entanto, na fase de projeto não foi utilizado nenhum dos métodos de Análise de Projeto de Sistemas, a não ser o diagrama, mostrado na Figura 4.01, tradicionalmente utilizado, que se preocupa de imediato com os aspectos físicos do armazenamento de dados. Poderia ter sido utilizado, por exemplo, o método do SADT (Structured Analysis and Design Technique) cujas referências bibliográficas estão citadas na seção 2.3. Esse método aborda o sistema sob dois aspectos: dados e atividades especificados em datagramas e ativigramas respectivamente.

O ambiente utilizando IDMS em que foi desenvolvido o sistema mostra-se bastante propício para a utilização de datagramas.

O método SADT permite uma abordagem rigorosa do sistema, cuidando para que nada seja esquecido. Concomitantemente podem ser elaborados os ativigramas, que cuidam das atividades que são executadas no sistema.

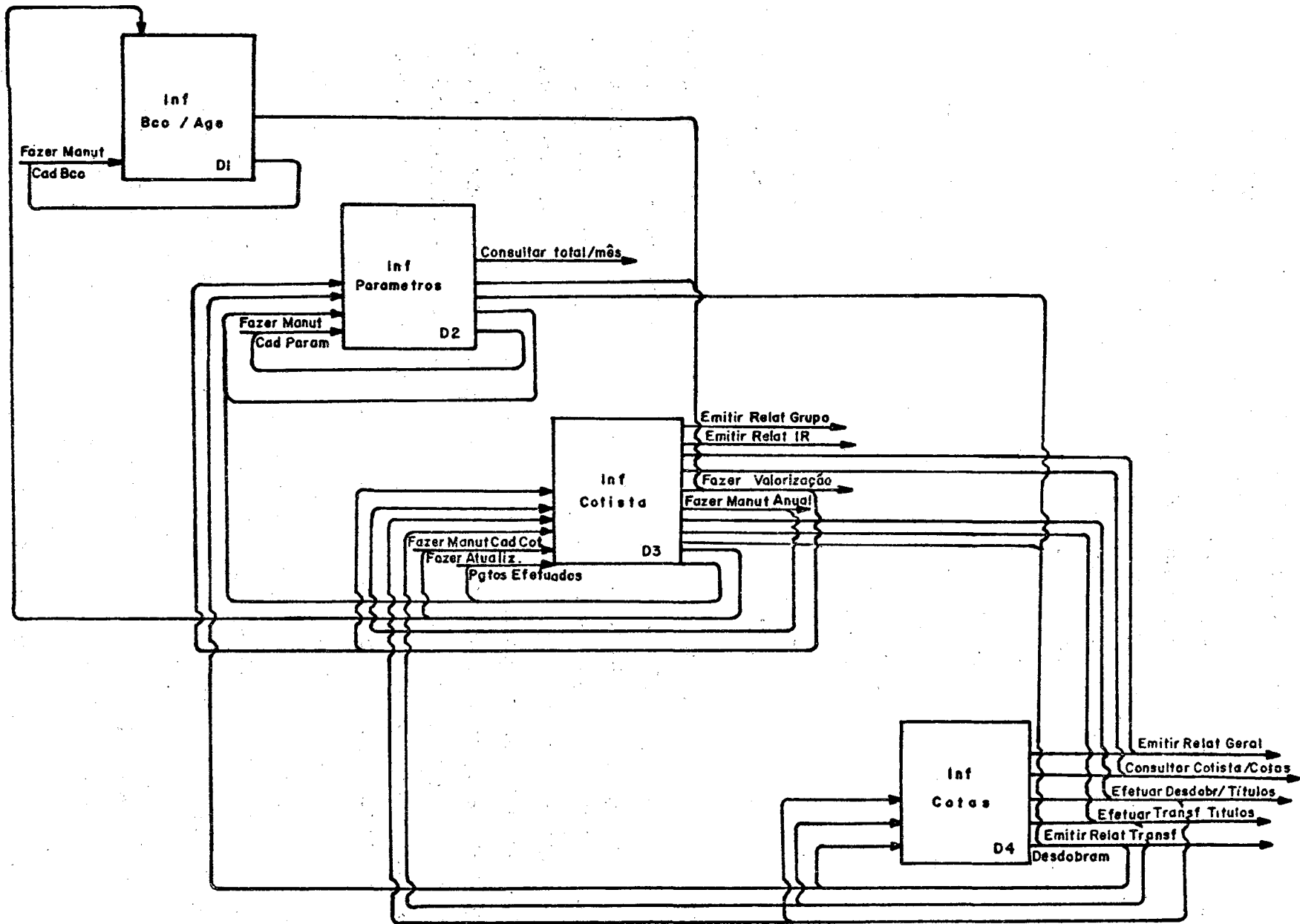
A Figura 5.01 ilustra o datagrama no nível mais abrangente, especificando os dados que fluem pelo sistema. As caixas representando os dados são exatamente os arquivos que foram identificados para o sistema.

As setas que inter-relacionam as caixas representando os dados, são atividades executadas no sistema. Além disso, as setas que "dão a volta" nas caixas representam a atividade de atualização dos dados (arquivos).

É interessante observar que neste diagrama constam algumas atividades rotuladas nas setas que não aquelas da Tabela 5.01. Essas são as atividades que se referem aos componentes do processamento por lotes que também fazem parte do sistema, como pode ser visto no Capítulo 4, tendo sido colocadas no diagrama para dar uma idéia mais completa do sistema como um todo.

TABELA 5.01 - Diálogos com as Respectivas Sub-rotinas

NOME DO DIÁLOGO	Nº DE SUB-RO TINAS	ROTINAS CHAMADAS
Manutenção do Cadastro Cotistas	10	R-INCLUS, R-ALTERA, R-CONSUL, R-BLOQUE, R-DESBLO, R-PRECOT, R-CONCOT, R-CLSOM, R-CENDAT, R-SAIDA
Manutenção do Cadastro Bancos	7	R-INCLUS, R-ALTERA, R-EXCLUS, R-CONSUL, R-PREBAG, R-CONBAG, R-SAIDA
Manutenção do Cadastro de Parâmetros	5	R-INCLUS, R-ALTERA, R-CONSUL, R-CONPAR, R-SAIDA
Atualização de Pagamentos Efetuados	7	R-PAGTOS, R-CONDAT, R-COMPL, R-TRIM, R-ALTCOT, R-ALTPGT, R-SAIDA
Transferência de Títulos	7	R-TRANSF, R-PRETRA, R-CONTRA, R-CONDAT, R-ALTCOT, R-ALTIT, R-SAIDA
Desdobramento de Títulos	7	R-DESDOB, R-PREDES, R-CONDES, R-CONDAT, R-ALTCOT, R-ALTIT, R-SAIDA
Consulta de Cotista/Cotas	7	R-PCOS, R-COMPL, R-MOVE, R-FIM, R-MODIF, R-CONDAT, R-SAIDA
Consulta de Totalização do Mês	1	R-CONSUL



DD - Datagrama

Figura 5.01 - Datagrama no Nível mais Geral

As caixas de dados que eventualmente exijam um maior esclarecimento podem ser detalhadas, constituindo-se assim um novo diagrama. A caixa que vai ser detalhada pertence ao diagrama pai e o diagrama resultante do detalhamento constitui o diagrama filho. Nesse diagrama, a caixa identificada por D2 foi detalhada como ilustrado na Figura 5.02. Pode-se observar que todas as entradas e saídas representadas em D2, também estão representadas no diagrama filho.

No que se refere à representação de atividades, é utilizado o ativograma como ilustrado na Figura 5.03, considerado num nível mais geral. Pode-se notar que esse ativograma permite ter uma idéia geral dos componentes do processamento imediato. Para tornar a representação de atividades mais completa, a Figura 5.04 contém as componentes do processamento por lotes.

Analisando a Figura 5.03, pode-se perceber que A1 - Faça Manutenção pode ser detalhado num outro diagrama, nas três componentes que podem ser vistas na Figura 5.05. Continuando a análise, nota-se que seria interessante um detalhamento de cada diálogo, pelo menos até o nível em que possam ser identificadas as rotinas que constam na Tabela 5.01. Cabe no entanto lembrar que como foi visto na seção 3.3 e detalhes no Apêndice I, um diálogo compõe-se de um mapa, um ou mais processos e sub-esquema. Dentre os processos, pode-se ter o processo pré-mapa e um ou mais processos de resposta ao mapa. Cada processo de resposta ao mapa, pode constituir-se de uma ou mais sub-rotinas. Na implementação, foram definidos processos de resposta ao mapa identificados como PF7, utilizados para restaurar o processamento.

O detalhamento do diálogo representado em A11 foi feito como ilustrado na Figura 5.06. Analisando a Figura 5.06, pode-se notar que o processo pré-mapa e os processos de resposta do mapa, estão muito bem representados principalmente no que se refere às setas indicando restrições dos dados, que são necessários para que se possa efetuar uma determinada atividade. No entanto, pela própria filosofia de utilização do PF7, esse processo não está muito bem representado na Figura 5.06, uma vez que o usuário pode eventualmente, apenas quando necessário, chamar essa rotina.

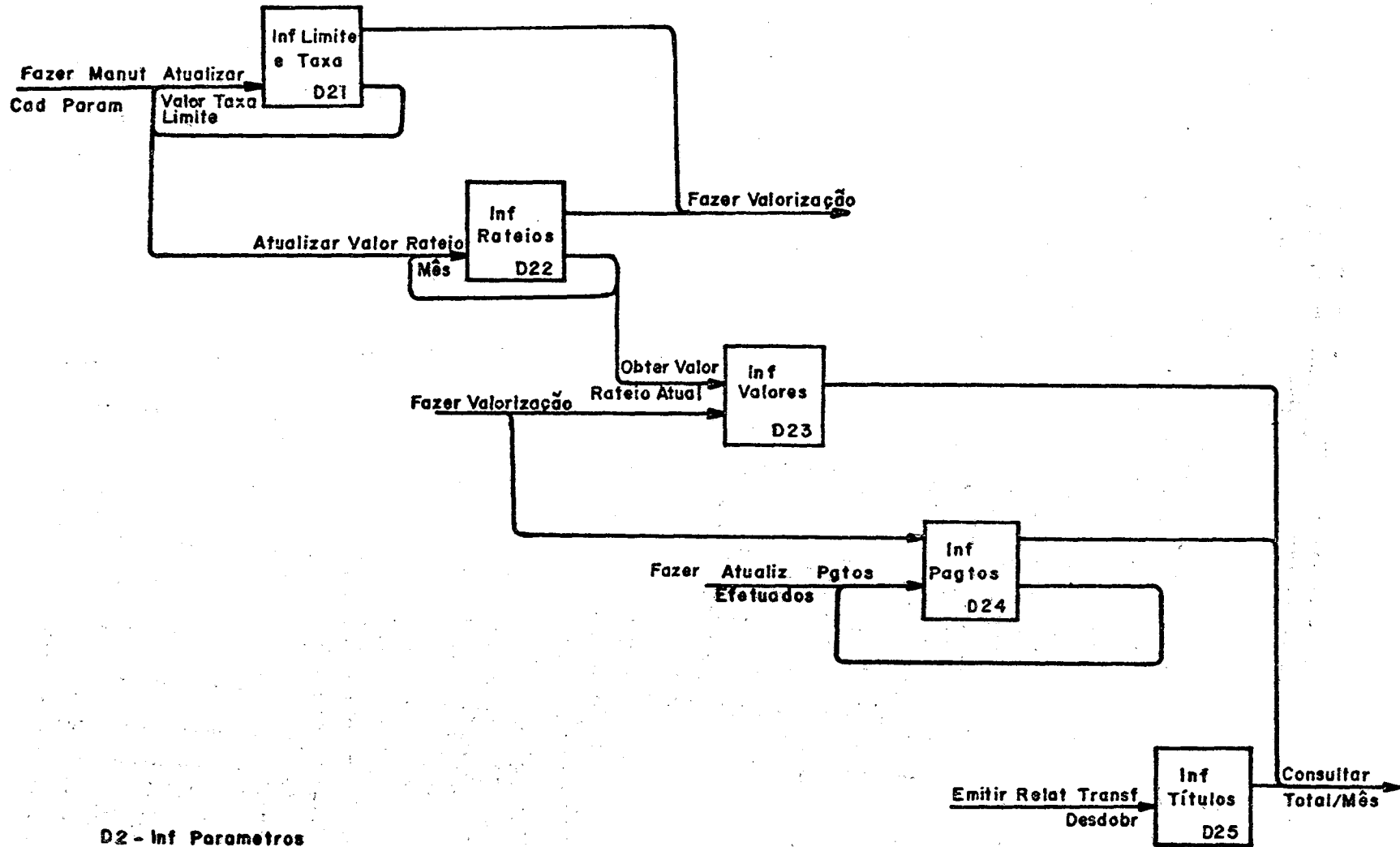
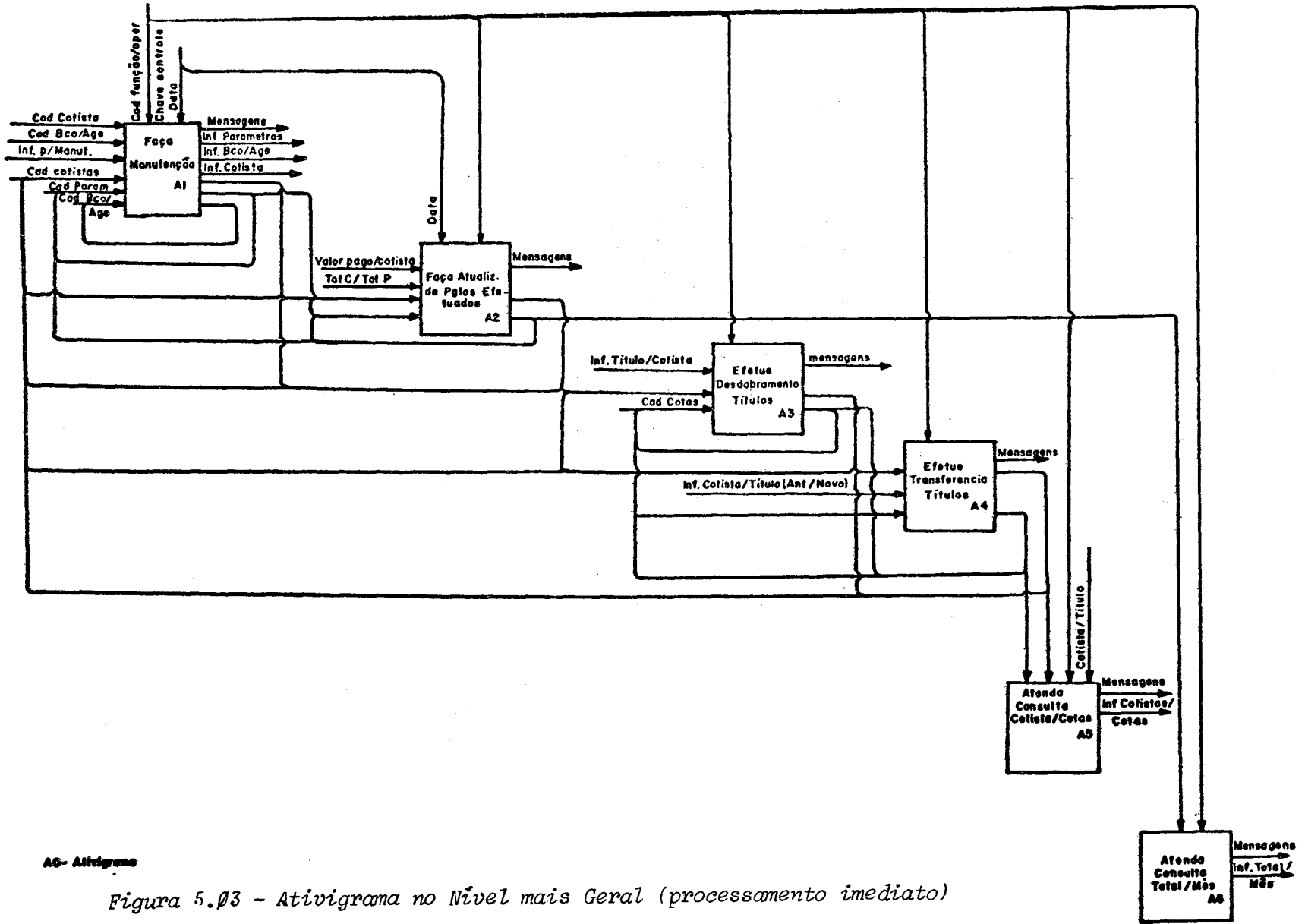
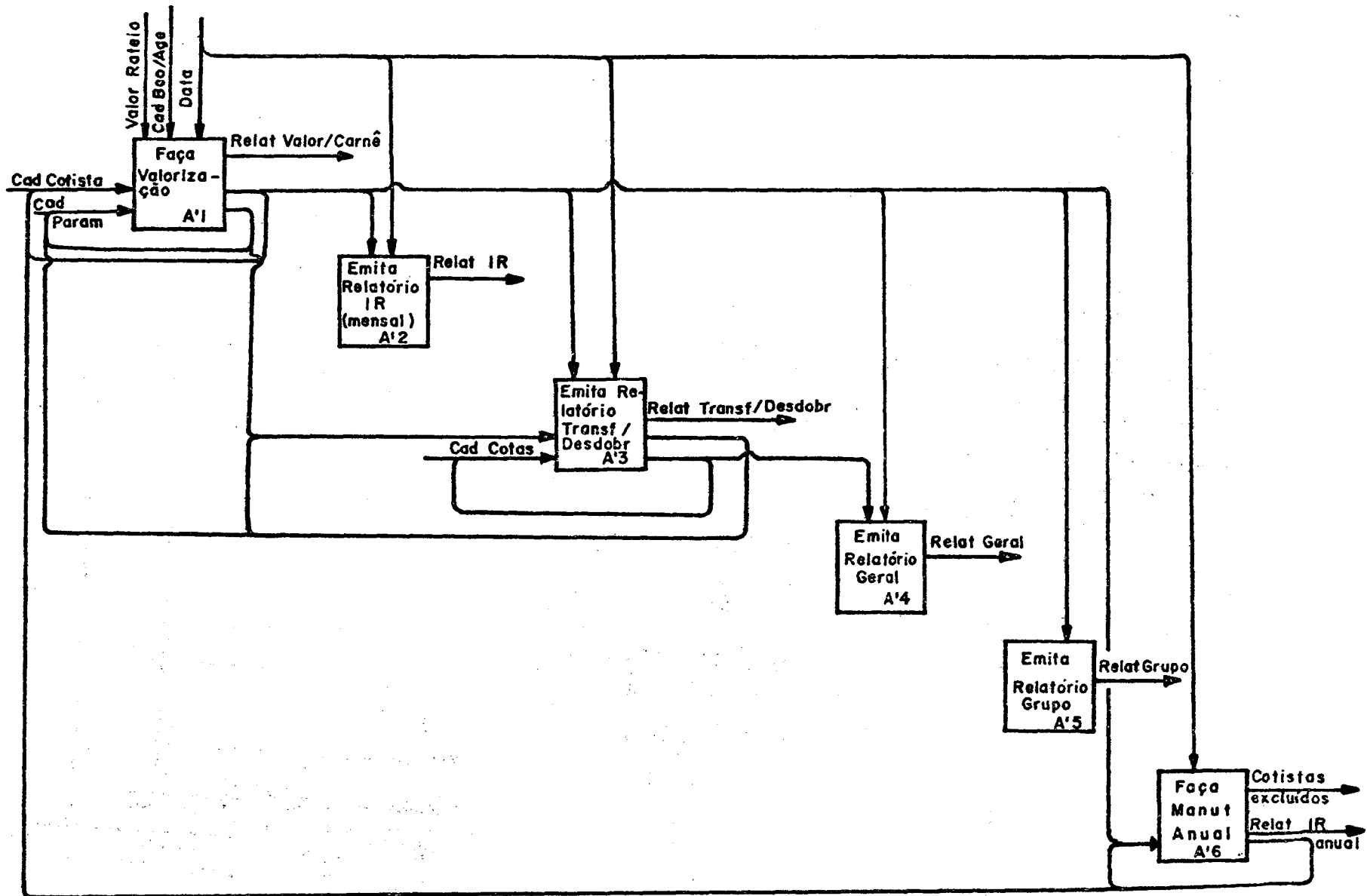


Figura 5.02 - Detalhamento da Componente D2



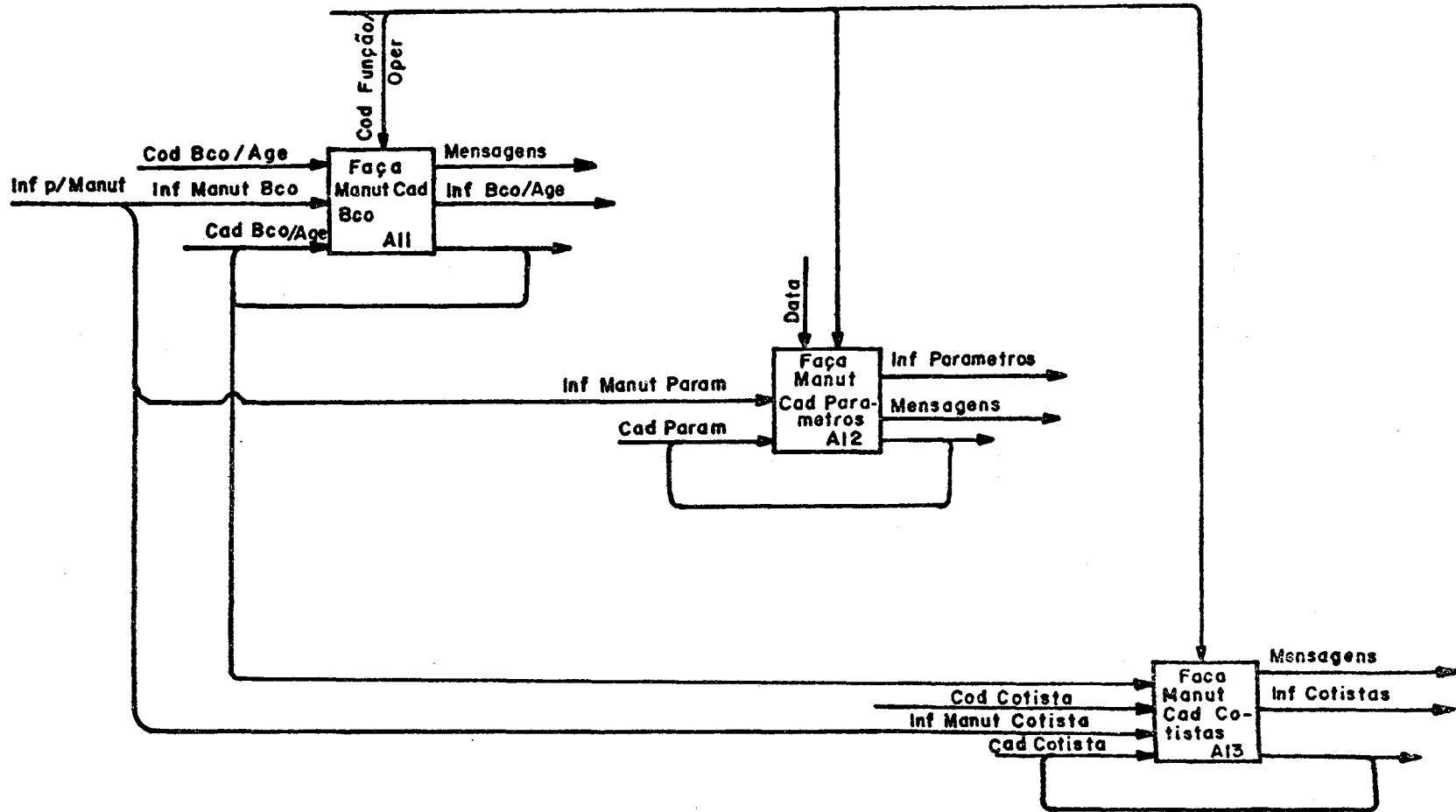
AG- Ativigrama

Figura 5.03 - Ativigrama no Nível mais Geral (processamento imediato)



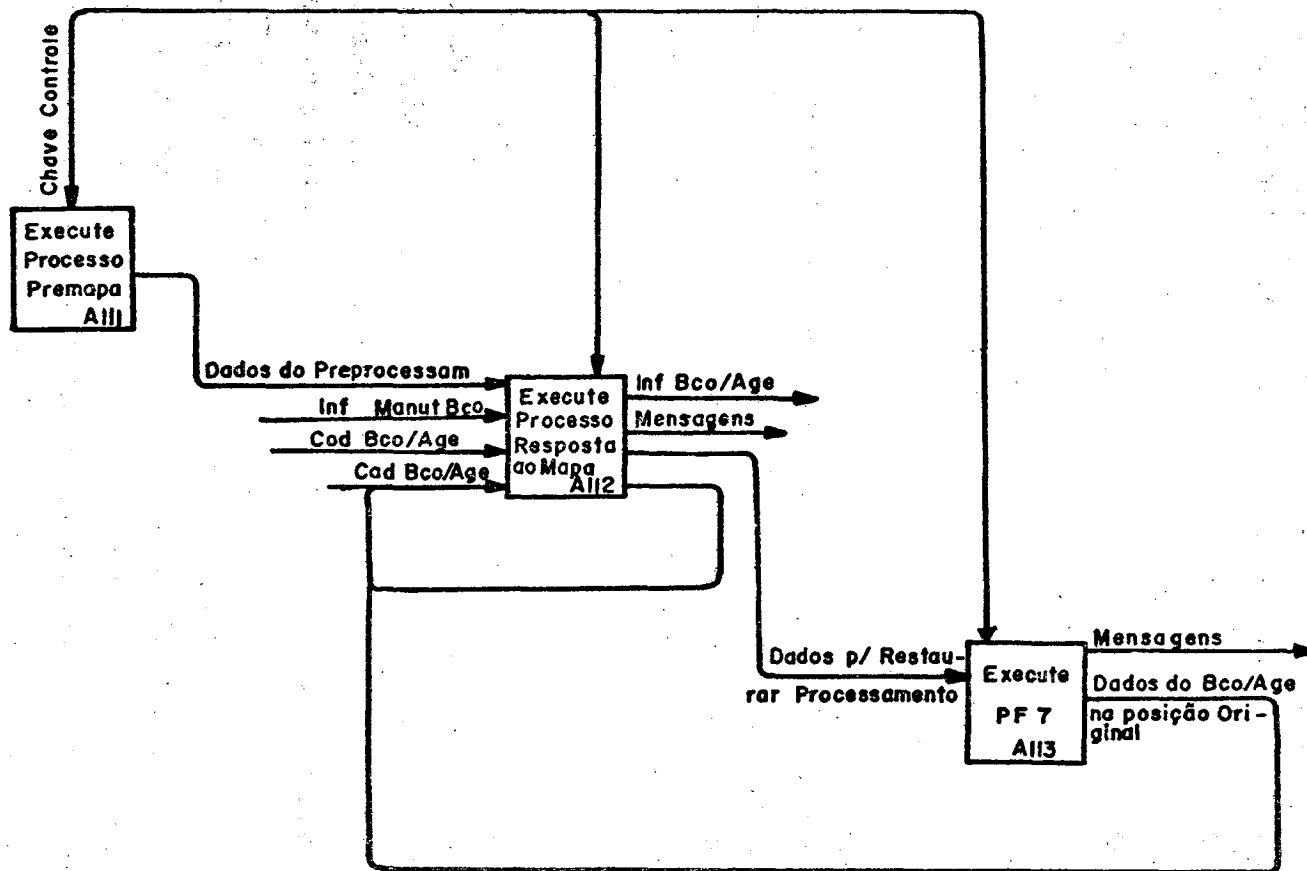
A0 - Ativograma-Batch

Figura 5.04 - Ativograma no Nível mais Geral (processamento por Lotes)



A1 - Faça Manutenção

Figura 5.05 - Detalhamento da Componente A1



A11 - Faça Manutenção Cad Bco

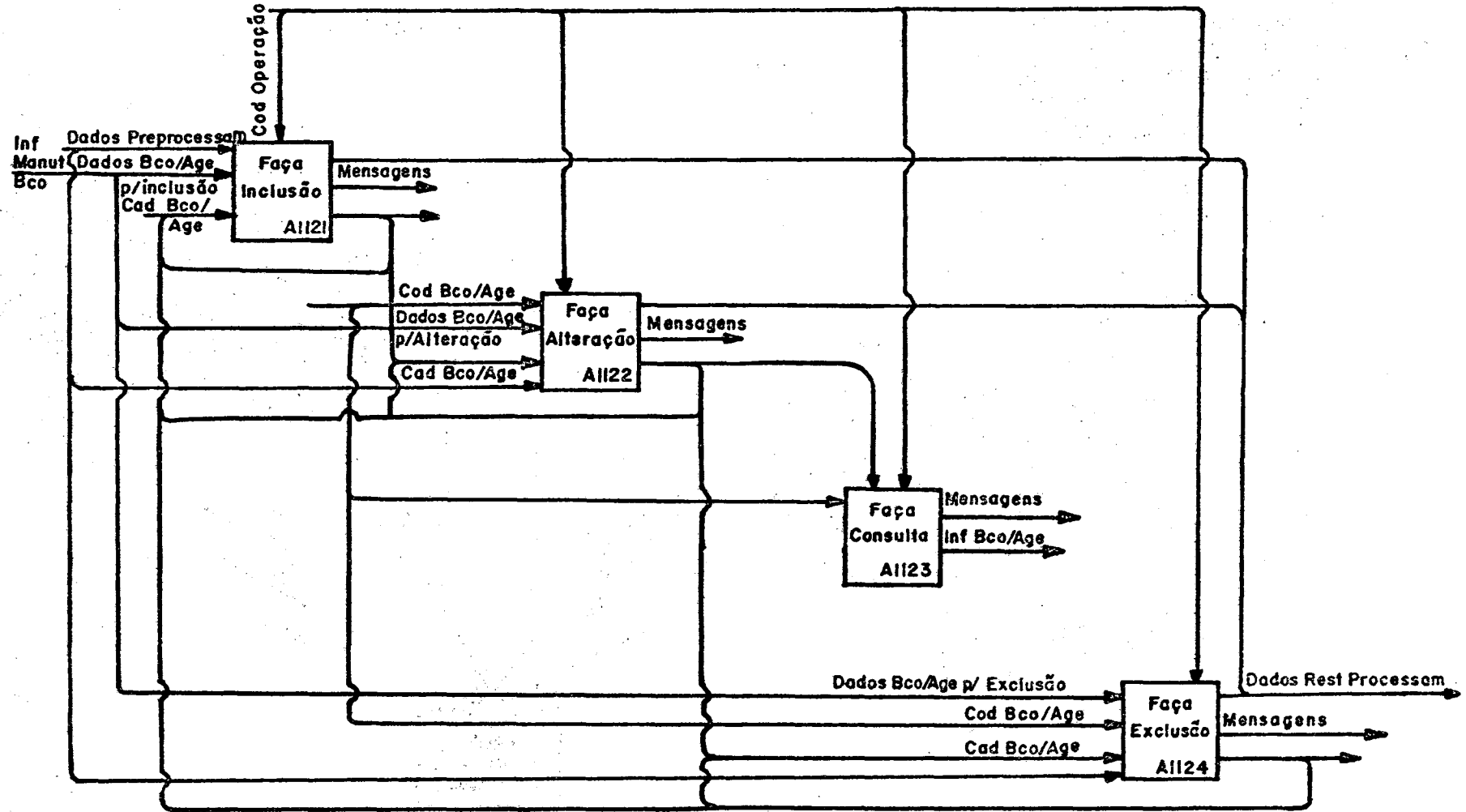
Figura 5.06 - Detalhamento da Componente A11

Cuidado semelhante deve ser tomado quando da interpretação da Figura 5.07, que é um detalhamento de A112. A cada código de operação é realizada uma atividade diferente, que nesse caso já estão ilustradas a nível de rotinas chamadas, que podem parcialmente ser identificadas na Tabela 5.01.

Na tentativa de encontrar uma melhor representação dos diálogos com os respectivos processos, foram elaborados diagramas de estrutura como constam em [Ga84] e [Yo78]. Pode-se perceber já na Figura 5.08, que é a representação correspondente às Figuras 5.06 e 5.07 em conjunto, uma maior flexibilidade na representação de situações alternativas, como aquelas que podem ser encontradas quando da representação das atividades a serem executadas dependendo do código de operação e/ou utilização do PF7. A Figura 5.09 ilustra o diagrama de estrutura correspondente à função Transferência de Títulos.

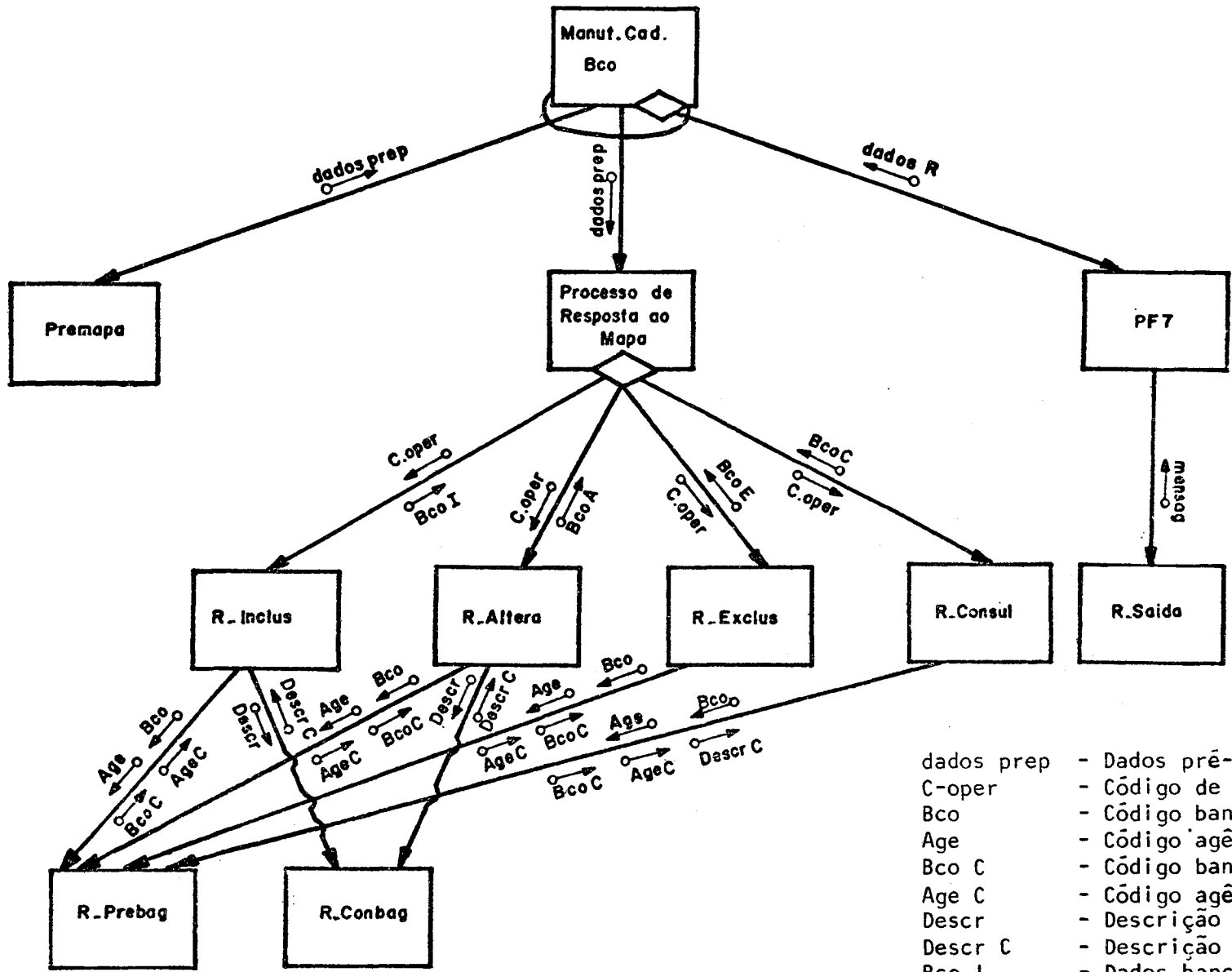
No diagrama de estrutura, podem ser identificadas todas as rotinas para cada diálogo, enquanto que no ativograma não foi possível esse nível de detalhamento, devido à restrição imposta pelo método SADT, de que cada diagrama deve ter não menos do que três e não mais do que seis caixas. Os dados que são transferidos de uma rotina para outra estão identificadas ao longo das setas.

Pode também ser utilizado o diagrama para representar o fluxo do sistema, como ilustrado na Figura 5.10. Como pode ser observado, este diagrama retrata a utilização do processo pré-mapa e processos de resposta do mapa com as respectivas chaves de controle a eles associadas. Identifica também os registros que são utilizados para cada um dos processos, no entanto não se consegue detalhes a nível de rotinas, que são chamadas como ilustrado no diagrama de estrutura e SADT. Por outro lado, dos três métodos aqui discutidos, é o único que consegue representar o fluxo do sistema em termos de o que acontece quando são pressionadas as chaves de controle associadas e/ou disponíveis.



A112 - Execute processo de resposta ao mapa(Banco)

Figura 5.07 - Detalhamento da Componente A112



- dados prep - Dados pré-processamento
- C-oper - Código de operação
- Bco - Código banco
- Age - Código agência
- Bco C - Código banco após consistência
- Age C - Código agência após consistência
- Descr - Descrição banco/agência
- Descr C - Descrição banco/agência após consistência
- Bco I - Dados banco/agência incluído

- Bco A - Dados banco/agência alterado
- Bco E - Dados banco/agência excluído
- Bco C - Dados banco/agência consulta
- Mensaq - Mensagem
- dado R - Dados restaurados

Figura 5.08 - Diagrama de Estrutura Correspondente ao Diálogo D42021

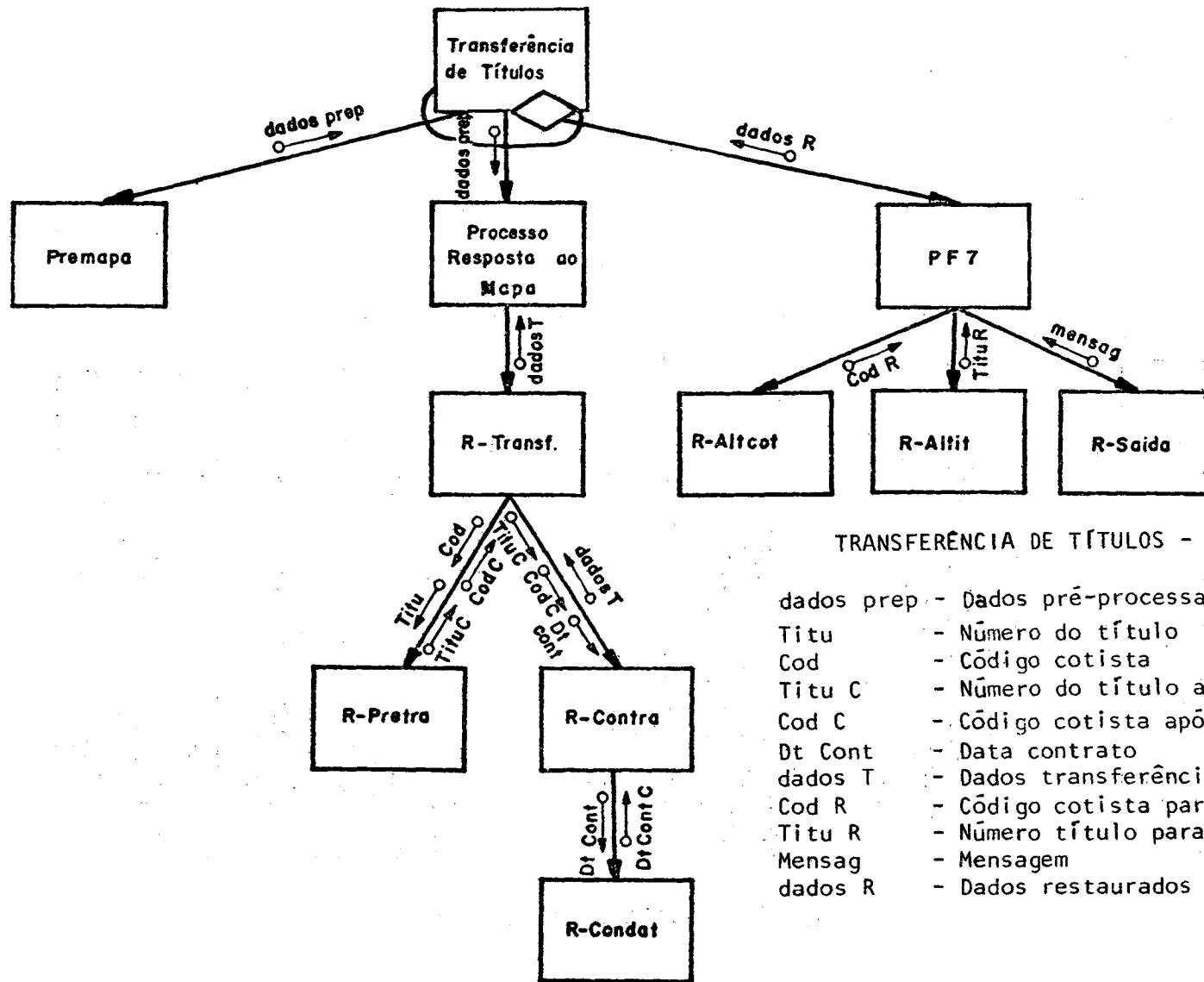
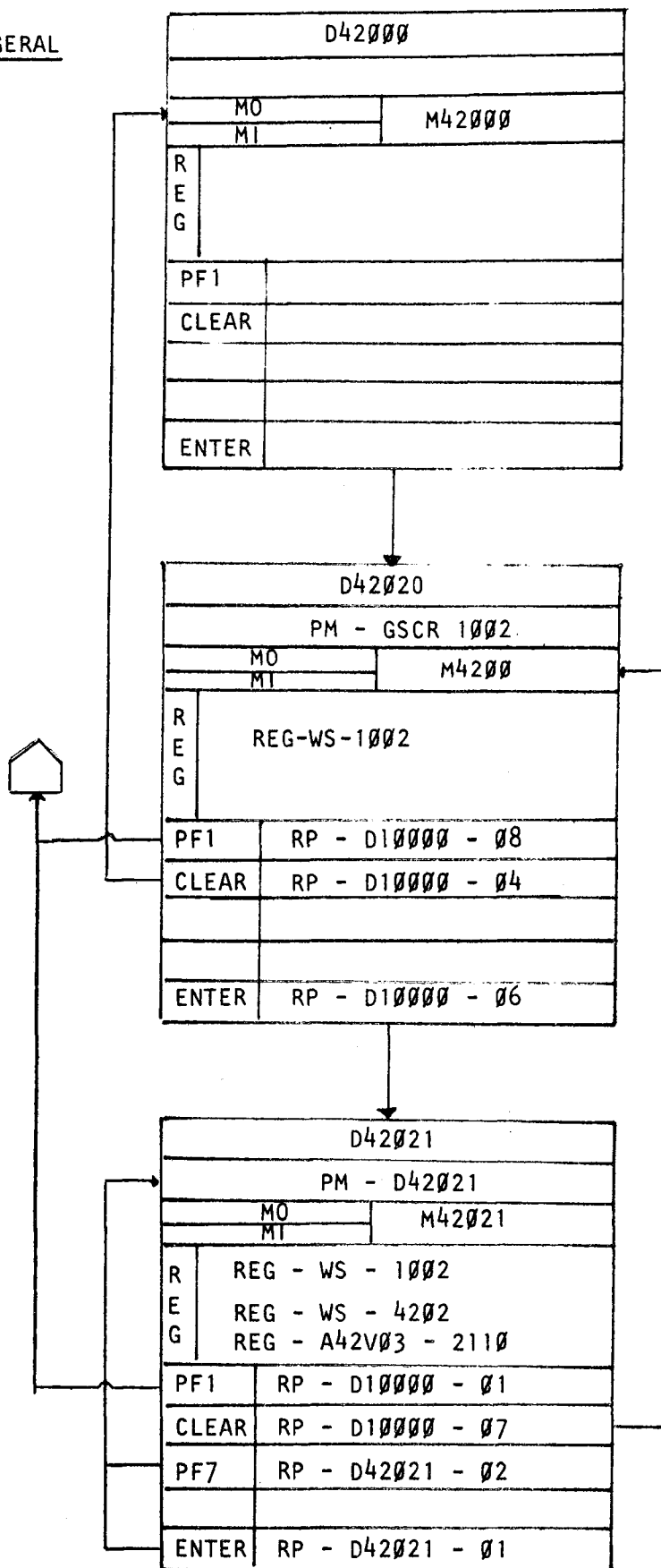


Figura 5.09 - Diagrama de Estrutura Correspondente ao Diálogo D42051

FLUXO GERAL



EMPRESA: Empresa-Cliente SISTEMA: Controle de Cotistas

FUNÇÃO: Manutenção do Cadastro de Bancos

Figura 5.10 - Diagrama do Fluxo do Sistema (chaves de controle)

5.3 - Sobre o Aproveitamento dos Recursos de Base de Dados

Como pôde ser visto no Capítulo 4, os dados do sistema desenvolvido, foram armazenados em arquivos VSAM sem a preocupação de definir uma base de dados.

O IDMS permite o acesso a ambos: tanto aos dados armazenados em arquivos VSAM como a uma base de dados. É claro que para utilizar os recursos para desenvolvimento ADS/O, em ambiente IDMS, sem uma base de dados houve necessidade de algumas adaptações que foram efetuadas pelo DBA, como foi dito na seção 4.3.

Na decisão da definição ou não de uma base de dados, devem ser considerados fatores tais como: porte do sistema, possibilidade de integração desse sistema com os demais da empresa. Assim, no caso desse sistema em específico, se houvesse projetos para sua integração futuramente com algum outro sistema, como por exemplo da área financeira, talvez pudesse ser pensado no projeto de uma base de dados. Sendo definida uma base de dados tornar-se-iam disponíveis comandos ADS característicos, para acesso à base de dados, como por exemplo CONNECT, DISCONNECT, FIND OWNER, etc...

Já no que se refere à linguagem de consulta OLQ, também tornariam disponíveis comandos para definição de caminhos e q-arquivos, que permitiriam "navegar" especificando explicitamente quem é o registro mestre e o registro membro, dando assim maior flexibilidade quer seja para atividade de consulta como de atualização de informações.

5.4 - Sobre o Aproveitamento dos Recursos de Linguagem de Consulta

Nos componentes de processamento imediato identificados para o sistema desenvolvido e que se encontram parcialmente ilustrado no Capítulo 4, a estrutura de alguns diálogos permite aos usuários escolherem a opção consulta, para terem acesso às informações, bem como foram definidos dois diálogos: consulta cotista/cotas e consulta totalização mês, totalmente voltados para consulta. Estes diálogos bem como as opções de consulta oferecidas por alguns, foram identificados já na fase de projeto e o acesso a elas é feito através do ADS.

No entanto, o IDMS oferece ferramentas como a OLQ e OLE entre outros, que talvez fosse interessante usá-las nas atividades de consulta. As referências bibliográficas relevantes a elas, encontram-se na seção 2.2.

O analista poderia utilizar a OLQ para ter acesso às informações de base de dados ou arquivos VSAM, como discutido na seção 3.4 e Apêndice II. Isto permitiria a ele o acesso ao conteúdo dos registros que poderiam ser utilizados, por exemplo, na fase de depuração dos diálogos.

O usuário pode, eventualmente, solicitar informações que envolvam um ou mais tipos de registros, que podem ser formatados em relatórios que não haviam sido identificados quando da especificação do sistema, ou mesmo consultas a registros individuais. No primeiro caso, se estes vierem a ser utilizados com maior frequência, podem ser definidos como q-arquivos e catalogados no dicionário de dados. Assim, o analista pode obter, para e juntamente com ele, informações que poderiam substituir em muitos casos, os tradicionais programas de emissão de relatório em COBOL.

Alguns dos recursos disponíveis na OLQ foram utilizados à parte da fase de desenvolvimento e se encontram ilustrados no Apêndice II. Pelos exemplos, pode-se perceber a facilidade e flexibilidade com que podem ser obtidas as informações.

Já a OLE, pela sua própria definição, está preocupada em oferecer recursos para que o usuário final possa por si,

formular as consultas, com vocabulário próprio para a aplicação e obter as informações necessárias. Pode-se também, utilizar a OLE, para obter o gráfico de barras que pode ser muito mais significativo. No entanto, não foi possível a utilização da OLE pela não disponibilidade da mesma, no momento em que foi feito o desenvolvimento.

Um assunto que também tem despertado interesse de muitos profissionais, tendo sido implantado em algumas empresas brasileiras, é o que se refere ao Centro de Informação (CI) que é brevemente descrito a seguir. As referências bibliográficas são citadas na seção 2.2.

Um CI serve de intermediário entre os recursos computacionais e os usuários finais que atuam nas diferentes áreas de uma empresa, tendo como objetivo proporcionar meios - técnicas e ferramentas - que permitam ao próprio usuário resolver os problemas das informações que necessita. Essas informações caracterizam as solicitações não rotineiras, como por exemplo, relatórios urgentes, consultas inesperadas, avaliação de alternativas, em que se basearão as decisões que podem decidir os rumos da empresa, devendo portanto serem rapidamente obtidas.

Quando da implantação, devem ser estabelecidas as responsabilidades a serem atribuídas ao CI, bem como aos usuários, além de ser definida a sua estrutura, uma vez que a sua implantação não implica no desaparecimento da equipe de desenvolvimento de sistemas.

Na estrutura que for definida, devem ser acomodados os profissionais responsáveis pela assistência que será proporcionada ao usuário. Devem ser designados profissionais especialistas em produtos (software) que se pretende tornar disponíveis, para que eles deem a orientação e treinamento mais adequado a cada problema. Esse treinamento pode ser feito através de cursos, exposições e até por computador, além de uma equipe em atendimento constante.

Em um ambiente com CI, talvez pudesse vir a ser utilizada a OLE principalmente para aquelas consultas não rotineiras e portanto não especificadas anteriormente, mas que são importantes para o usuário que está solicitando. Neste caso, um especialista em OLE estaria disponível para a orientação adequada.

Note-se no entanto que, mesmo com a implantação de um CI, as atividades de consulta e de emissão de relatórios, bem como os recursos de consulta oferecidos pela OLQ coexistiriam, procurando oferecer ao usuário as informações que ele necessita no menor tempo possível.

Além da OLE, existem outros produtos próprios para CI, disponíveis no mercado, dirigidos para diferentes áreas da empresa e portanto para diferentes aplicações.

5.5 - Conclusões

O IDMS tem disponível um conjunto de recursos que visam fornecer aos usuários e analistas, facilidades de acesso às informações.

Na avaliação efetuada, foi considerada apenas parte do conjunto de recursos.

No que se refere aos métodos de análise e projeto, como pode ser visto nas ilustrações, quando o sistema é documentado utilizando o SADT ou diagramas de estrutura, torna-se muito mais fácil a visualização do sistema como um todo, através da identificação dos sub-módulos que compõem cada módulo. Essa identificação pode ser feita a nível de aplicação ou mesmo de diálogos ADS. A utilização de datagramas permite determinar todos os dados necessários. A identificação de dados e a sua estruturação é de importância vital para desenvolvimento de sistemas consistentes, mas principalmente para ambientes que utilizam base de dados. É claro que as características inerentes a cada método, mostram-se mais propícias a cada situação. O analista deve ser flexível ao ponto de admitir e utilizar aqueles métodos que melhor se adequem.

O tratamento dispensado ao armazenamento de dados sem a utilização de base de dados atende às especificações do sistema atual. Houve, naturalmente uma adaptação nas definições para que pudessem ser utilizados os recursos ADS.

Já a linguagem de consulta OLQ, mostra-se versátil tanto para as recuperações de dados uma a uma como a um conjunto de dados que satisfaçam a determinados critérios. Esse último tipo, pode eventualmente, dependendo da complexidade da consulta e frequência de utilização, justificar uma catalogação no dicionário de dados. Assim, o usuário poderá utilizá-la quando necessário. Embora não tendo sido muito crítico, talvez a OLQ pudesse ter sido utilizada pelos analistas na fase de depuração dos diálogos, pois permitem um rápido acesso às informações.

A linguagem OLE, voltada ao usuário que queira efetuar consultas não rotineiras, mas que são muitas vezes fundamentais no processo de tomada de decisão, não pôde ser utilizada,

por ser um "software" ainda não liberado para utilização, na instalação em que foi feito o desenvolvimento. Muitas dessas consultas, poderiam talvez serem supridas utilizando recursos da OLQ, sendo que nesse caso, o vocabulário utilizado não seria tão familiar ao usuário, devendo portanto um analista acompanhá-lo durante a consulta.

Cabe ainda lembrar, que o sistema desenvolvido não estava inserido em um ambiente com CI. Se houver algum projeto de implantação de um CI, talvez ele fosse melhor aproveitado se implantado na "empresa-cliente" e não na empresa onde foi desenvolvido o sistema, que se responsabilizaria então de oferecer todo o treinamento necessário. Note-se no entanto que, a sua implantação, não implicaria na substituição total de recursos e procedimentos hoje adotados, muito pelo contrário, eles coexistiriam com os "softwares" como OLE e outros que existam no mercado, como suporte ao usuário.

CAPÍTULO VI

CONCLUSÕES

6.1 - Conclusões Finais

Ao final da implementação do sistema controle de cotistas, utilizando um software integrado de 4.^a geração - o IDMS, pode-se dizer que a ferramenta ADS/on line, proporcionou sem dúvida uma agilização no processo de desenvolvimento do sistema. A declaração para que se efetue a manipulação de erros e edição automática para os campos declarados no mapa, contribuem para que diminua a complexidade na codificação de programas (processos) . Embora os processos associados aos diálogos ADS, exijam a definição da lógica, isto não deve ser muito crítico, pois de qualquer forma, as pessoas envolvidas são da equipe de desenvolvimento de sistemas, que têm portanto a necessária experiência.

Para consultas a registros individualmente e mesmo aquelas mais complexas, é oferecida a ferramenta OLQ, que é bastante flexível para formatação das informações obtidas, em "relatórios on line".

No que se refere à documentação automática, o que é oferecido, como pode ser visto nas referências que constam no Capítulo 2, é um conjunto de relatórios. Para uma descrição mais detalhada de registros, processos, q-arquivos, etc... pode ser solicitada a descrição fonte que consta no dicionário de dados. Além disso, pelo que consta nas referências, sendo identificado um diálogo, poderão por exemplo ser obtidas informações de seus componentes: mapa, nomes dos processos com as respectivas chaves de controle associadas, data da última atualização efetuada, por quem foi feita a atualização, registros utilizados, esquema e sub-esquema associados. Nos processos identificados que eventualmente sejam compostos de várias sub-rotinas, estas não são explicitamente declaradas no relatório.

No que se refere a uma documentação manual dos diálogos componentes de uma aplicação e até mesmo das sub-rotinas

que compõe os processos associados a cada diálogo, também não puderam ser identificadas. Assim, como pode ser visto no Capítulo 5, foram elaborados diagramas de parte dos componentes do sistema desenvolvido, mas já pode ser sentida a facilidade de visualização e um melhor entendimento do sistema como um todo, inclusive já preocupando-se com o projeto de módulos dirigidos à implementação. Mesmo assim, ainda não está a nível de uma passagem imediata para a programação. Note-se no entanto, que de nada adiantarão esses diagramas se eles não forem mantidos atualizados.

Para a definição de uma base de dados, como pode ser visto nas referências, elementos e registros identificados durante a análise, devem ser criteriosamente agrupados em entidades que devem obedecer às regras de normalização para que se obtenha dados mais consistentes.

A definição destas entidades como também de processos no dicionário de dados e a integração de um conjunto de "software" do IDMS, que utilizam informações do dicionário de dados, permitem a obtenção de dados atualizados e consistentes. É claro que para que se possa desfrutar dessas facilidades, devem ser estabelecidos critérios de segurança para impedir que pessoas não autorizadas danifiquem as informações.

Como a definição das entidades: mapas, processos, pode ser feita gradualmente, embora não possua representação gráfica, talvez nas fases iniciais estas definições possam documentar cada etapa que estiver inserindo informações no dicionário de dados.

6.2 - Contribuição deste Trabalho

Em 1.2 é enunciado o problema que se propôs resolver nesta dissertação. Como lá foi dito, para sua solução estudou-se um instrumental integrado que inclui linguagem de 4.^a geração e participou-se de sua aplicação a um caso prático da vida real.

Nos estudos feitos do referido instrumental, notou-se que, apesar de haver abundância de detalhes no material bibliográfico reunido para permitir sua aplicação, havia pouca informação sobre os recursos de documentação oferecidos. Praticamente apenas em [C84a] encontram-se diretrizes específicas a respeito dos relatórios que podem ser obtidos sobre as informações contidas no dicionário de dados. Esses recursos mesmo aproveitados plenamente na fase de análise são insuficientes para apoiá-la adequadamente.

Assim, neste trabalho mostra-se como recursos adicionais extraídos de métodos de análise e projeto de sistemas de informação convencionais, podem ser aproveitados no contexto da aplicação desse instrumental integrado. Acredita-se que a combinação desses recursos adicionais, com aqueles oferecidos pelo dicionário de dados, possibilite aumentar a eficiência e rapidez da fase de análise para poder tirar pleno proveito da maneira expedita como pode ser feita a construção do sistema.

Considera-se assim uma contribuição deste trabalho a detecção e formulação de um problema que surge no desenvolvimento de sistemas de informação com instrumental de 4.^a geração e o equacionamento da sua solução.

Como sub-produto das atividades desenvolvidas para conseguir essa contribuição, obteve-se um aprofundamento substancial do conhecimento no meio em que desenvolveu-se essas atividades de alguns recursos do IDMS, principalmente daqueles efetivamente utilizados na aplicação prática realizada. Uma apresentação compacta em português desses recursos ilustrada com exemplos reais, que é incluído nesta dissertação pode ser considerada uma contribuição adicional deste trabalho.

6.3 - Sugestões para Futuras Pesquisas

No Capítulo 5, são apresentados os recursos de análise e projeto de sistemas que, combinados com aqueles oferecidos pelo IDMS, poderiam contribuir no desenvolvimento de sistemas de informação. Assim, seria interessante o desenvolvimento de um outro sistema real, procurando utilizar completamente tais recursos, desde a fase de análise e extendendo-se para a fase de projeto, para comprovar a conveniência de tal combinação. Procurar-se-ia na fase de análise, utilizar o SADT quer para a análise das atividades, quer para a análise de dados, obtendo-se assim uma representação gráfica do sistema, que poderia ser utilizada para discussão com o usuário. Já na fase de projeto, poderia ser utilizado o diagrama de estrutura para identificar as sub-rotinas que compõem cada processo e que poderiam ser armazenadas no dicionário de dados. Dependendo do nível de detalhamento, essa definição seria praticamente a definição do "programa". Ainda, o usuário poderia participar com o analista na elaboração dos mapas cuja necessidade tenha sido identificada. Todos os diagramas e relatórios do que tiver sido armazenado no dicionário de dados, poderiam constituir-se numa documentação derivada em paralelo às fases de análise e projeto de sistema.

Um outro assunto que merece ser explorado, é o que se refere a Centro de Informação. Assim, poderia ser estudado o tratamento adequado a ser dispensado a um sistema do tipo daquele apresentado neste trabalho, quando desenvolvido em ambientes que incorporem Centro de Informação. O referido estudo poderia ser feito cuidando do aspecto documentação: a necessidade e o método a ser utilizado.

Para que se possa efetivamente aplicar os conceitos para projeto de base de dados IDMS, seria interessante o desenvolvimento de um sistema real que fosse apoiado com estrutura de base de dados. O método SADT (ativigramas e datagramas) acima sugerido, poderia ser utilizado para identificação das atividades e dados do sistema, sobre os quais seria feita a referência cruzada para assegurar que as exigências do sistema ficam satisfeitas. Com base nesses dados e atividades que poderiam estar armazenados no dicionário de dados, seriam então identificadas as en

tidades que irão compor a estrutura da base de dados.

Ainda num contexto mais amplo, poderiam ser estudados outros instrumentos disponíveis no mercado, como por exemplo: LINC [Ru85] e DATATRIEVE [Di83a], [Di83b], [Di83c], [Di83d] e utilizá-los no desenvolvimento de um sistema real. Procurar-se-ia identificar recursos para definição e facilidade de manutenção dos componentes de processamento, facilidades de consulta, recursos que poderiam ser adotados em Centro de Informação e recursos para documentação, disponíveis em cada um deles. Nesse estudo poderiam ser determinados os aspectos positivos e negativos de cada um, com o que se poderia oferecer aos futuros usuários, subsídios para a escolha do instrumental mais adequado.

BIBLIOGRAFIA

- Ch79 - CHAPIN, N. - Some Structured Analysis Techniques.
Database: 16-23, 10(3) Winter, 1979.
- Ch81a - CHAPIN, N. - Graphic Tools In the Design of Information Systems. In: Cotterman, W.W et all. Systems Analysis and Design: A Foundation for the 1980's. North-Holland, 1981.
- Ch81b - CHAPPIN, N. - Structered Analysis and Structured Design an Overview. In: Cotterman, W.W. et all. Systems Analysis and Design: A Foundation for the 1980's. North-Holland, 1981.
- C281 - CULLINANE DATABASE SYSTEM INC. - CULPRIT Report Generator: A Management Summary, MACU-BR01-0582 . Cullinam Database System Inc, Westwood, Mass., 1981.
- C282a - CULLINANE DATABASE SYSTEM INC.- Application Development System/on line - Summary Description, SADO-1282-1001, release 1.0. Cullinane Database System Inc, Westwood, Mass., 1982.
- C282b - CULLINANE DATABASE SYSTEM INC. - IDMS - DC/UCF - Mapping Facility User's Guide, TDDC-0310-2000, revision 0.0, release 2.0. Cullinane Database System Inc, Westwood, Mass., 1982.
- C282c - CULLINANE DATABASE SYSTEM INC. - Integrated Data Dictionary Summary Description, SDID-0682-3001, release 5.0. Cullinane Database System Inc, Westwood, Mass., 1982.
- C283a - CULLINANE DATABASE SYSTEM INC. - ADS/O User's Guide, TDAO-0310-1100, revision P.0. Cullinane Database System Inc, Westwood, Mass., 1983.

- C283b - CULLINET SOFTWARE INC. - CULPRIT User's Guide, TACU-0310-6102, revision 0.2, release 6.1. Cullinet Software Inc, Westwood, Mass., 1983.
- C283c - CULLINET SOFTWARE INC. - Database Analysis and Design, EDDB-0040-5701, revision 0.0. Cullinet Software Inc, Westwood, Mass., 1983.
- C283d - CULLINET SOFTWARE INC. - IDD - DDDL Reference Guide, TDID-0321-3010, Revision 1.0, release 3.0. Cullinet Database System Inc, Westwood, Mass., 1983.
- C283e - CULLINANE DATABASE SYSTEM INC. - IDMS - System Overview, TDDB-0180-5502, Revision 0.2, release 5.5. Cullinane Database System Inc, Westwood, Mass., 1983.
- C283f - CULLINANE DATABASE SYSTEM INC. - IDMS - Database Design and Definition Guide, TDDB-0300-5521, revision 0.1, release 5.7. Cullinane Database System Inc, Westwood, Mass., 1983.
- C283g - CULLINET SOFTWARE INC. - Introduction to the Application Development System/on line, EDAO-0020-1002. Cullinet Software Inc, Westwood, Mass., 1983.
- C283h - CULLINANE DATABASE SYSTEM INC. - On line English-Summary Description, SDDE-0383-2002. Cullinane System Inc, Westwood, Mass., 1983.
- C283i - CULLINET SOFTWARE INC. - On line English - Design and Definition Guide, TDDE-0230-3000, revision p.0, release 3.0. Cullinet Software Inc, Westwood, Mass., 1983.
- C283j - CULLINET SOFTWARE INC. - On line Query - User's Guide, TDOQ-0320-3001. Cullinet Software Inc, Westwood, Mass., 1983.

- C284a - CULLINET SOFTWARE INC. - ADS/O - Reference,
TDAO-0330-100P0, revision P.0, release 10.0.
Cullinet Software Inc, Westwood Mass., 1984.
- C284b - CULLINET SOFTWARE INC. - Integrated Data Dictionary
User's Guide. Cullinet Software Inc, Westwood, Mass.,
1984.
- Cm78 - COMBELIC, D. - User Experience With New Software
Methods - Experience with SADT. National Computer
Conference: 631-633, 1978.
- Cn81 - CONNOR, M.F. - Structured Analysis and Design Technique
In: Cotterman, W.W. et all. Systems Analysis and
Design: A Foundation for the 1980's. North-Holland,
1981.
- Co85 - COBB, R.H. - In Praise of 4GLS. Datamation: 31(14)
90-96, jul. 1985.
- Ct81 - COTTERMAN, W.W. et all - Systems Analysis and Design:
A Foundation for the 1980's. North-Holland, 1981.
- Di83a - DIGITAL EQUIPMENT CORPORATION - VAX-11 DATATRIEVE
Guide to Using Graphics, AA-L631B-TE. Digital
Equipment Corporation, Maynard, Mass., 1983.
- Di83b - DIGITAL EQUIPMENT CORPORATION - VAX-11 DATATRIEVE
Guide to Writing Reports, AA-P862A-TE. Digital
Equipment Corporation Maynard, Mass., 1983.
- Di83c - DIGITAL EQUIPMENT CORPORATION - VAX-11 DATATRIEVE
Summary Description, AA-P860A-TE. Digital Equipment
Corporation, Maynard, Mass., 1983.
- Di83d - DIGITAL EQUIPMENT CORPORATION - VAX-11 DATATRIEVE
User's Guide, AA-K080B-TE. Digital Equipment
Corporation, Maynard, Mass., 1983.

- En81 - ENGER, N.L. - Classical and Structured Systems Life Cycle Phases and Documentation. In: Cotterman, W. W. et all. Systems Analysis and Design: A Foundation for the 1980's. North-Holland, 1981.
- Ga84 - GANE, C.& SARSON, T. - Análise Estruturada de Sistemas Livros Técnicos e Científicos Editora S.A., Rio de Janeiro, 1984.
- Gr85 - GRANT, F.J. - The Downside of 4GLS. Datamation:31(14) 99-104, jul., 1985.
- Ha82 - HAMMOND, L.W. - Management considerations for an Information Center. IBM System Journal, 21(2), 131-161, 1982.
- Ibxx - IBM do Brasil - Centro de Informações: veja de que maneira ele pode ajudar você. IBM; Rio de Janeiro, sem data.
- La79 - LANO, R.J. - A Technique for Software and System Design, North-Holland, 1979.
- Ma81 - MARTIN, J. & FINKELSTEIN, C. - Use of Fourth Generation Language In: Information Engineering. vol.2, Savant Institute, 1981.
- Ma82 - MARTIN, J. - Application Development Without Programmers. Prentice Hall, New Jersey, 1982.
- Ro77a - ROSS, D.T. - Structured Analysis (SA): A Language for Communicating Ideas. IEE Transaction Software Engineering, SE-2 (1), Jan. 1977.
- Ro77b - ROSS, D.T. & SCHOMAN, K.E. - Structured Analysis for Requirements Definition. IEE Transaction Software Engineering, SE-3(1), 1977.

- Ru85 - RUDOLPH, E.E. - A Produtividade no desenvolvimento de Aplicações por Computador. Data News, 20-27, ago. 1985.
- Rw85 - ROWE, L.A. - Tools for developing OLTP Applications Datamation: 31(15) 73-82, ago. 1985.
- Sf76 - SOFTECH INC. - An Introduction to SADT. (softech document # 9022-78). feb., 1976.
- So84 - SOUZA, J.M. de - CI: As Experiências de 4 Grandes Empresas. Data News, 22-26, mar., 84.
- Sq81 - SQUIRE, E. & YOSTRA, R. - Information Center Documentation Examples. Maryland, IBM Washington Systems Center, Technical Bulletin, 1981.
- Sq82 - SQUIRE, E. & YOSTRA, R. - Information Center Implementation Guide. Maryland, IBM Washington Systems Center, Technical Bulletin, 1982.
- St85 - STEVENS, W.P. - Projeto Estruturado de Sistemas, Editora Campus Ltda. Rio de Janeiro, 1985.
- Ti85 - TINNIRELLO, P. - Make a sensible 4GL Selection. Computer Decision, 40-45, jul., 1985.
- Yo78 - YOURDON, E. & CONSTANTINE, H. - Structured Design Yourdon Press, New York, 1978.

APÊNDICE I

A - CONSIDERAÇÕES GERAIS

A ferramenta para desenvolvimento de aplicações "on line" ADS/on line, apresenta como parte de seus componentes o gerador de aplicações - ADSA e o gerador de diálogos - ADSG.

Nesse Apêndice, são apresentados detalhes relativos aos procedimentos para gerar e executar aplicações compostas apenas de diálogos.

Uma visão geral sobre ela foi apresentada no Capítulo 3 e as referências bibliográficas citadas no capítulo 2.

Os detalhes aqui apresentados pretendem possibilitar ao leitor, um melhor entendimento dos exemplos apresentados na seção 4.4. Assim, são apresentados: conjunto de telas utilizadas para definir mapas, conjunto de comandos válidos para definir processos e o conjunto de telas utilizadas para gerar diálogos. De acordo com esse conteúdo e visando facilitar a leitura, este Apêndice está dividido em 3 partes a saber: B - Procedimentos para gerar mapas; C - Recursos para definir processos e D - Procedimentos para gerar diálogos. Para complementar a descrição textual, acompanham também ilustrações em cada uma destas partes.

Uma *aplicação* pode ser entendida como um conjunto de disparos e funções - definidas pelo gerador de aplicações, ou um conjunto de *diálogos* - definidos pelo gerador de diálogos, dispostos em uma determinada estrutura, para executar uma tarefa.

Um diálogo é a unidade de trabalho ADS. É composto de um *mapa*, um ou mais *processos*, sub-esquema e registros.

Um mapa pode ser entendido como uma tela formatada através da qual, o usuário recebe ou envia dados para processamento.

Um processo constitui-se de um ou mais comandos ADS que executam um processamento. Os comandos serão descritos posteriormente.

Para que uma aplicação possa ser executada, diálogos devem ser gerados e armazenados no dicionário de dados. Por sua vez, para que um diálogo possa ser gerado, seus componentes devem estar definidos no dicionário de dados. Portanto, para definir um mapa, devem ser definidos no dicionário de dados todos os registros de trabalho e tabelas utilizados no mapa, o que é feito através da DDDL (Data Dictionary Description Language) do IDD (Integrated Data Dictionary). Detalhes sobre a DDDL podem ser encontrados em [C83d]. A partir do momento em que as definições tenham sido concluídas, o projetista pode iniciar a definição do mapa. Para tanto, são utilizados os recursos do OLM (On Line Mapping) através de uma sequência de telas que serão discutidos a seguir. Detalhes sobre o OLM podem ser encontrados em [C82b].

B - PROCEDIMENTOS PARA GERAR MAPAS

A Figura I.01 ilustra a tela de definição inicial. É a primeira tela exibida quando da utilização do OLM. Nela podem ser definidos o nome do mapa, a opção para que soe o alarme quando da exibição do mapa, os nomes dos registros de trabalho ou do sub-esquema que serão utilizados pelo mapa. Pode-se definir também o nome do mapa que será copiado, se estiver sendo definido um mapa semelhante a um que já exista no dicionário de dados. A Figura I.01.a ilustra a tela preenchida.

A seguir, passa-se o controle para a tela de definição de formato do mapa. Esta tela é inicialmente exibida "limpa". Pode-se portanto "desenhar" o mapa com as informações que serão exibidas ao usuário, o que é feito movimentando o cursor na tela até a posição que o campo ocupará. Esses campos podem ser constantes literais ou campos variáveis e deve-se indicá-los com o caracter seletor de campo "%", como pode ser visto na Figura I.02. Os campos variáveis são os campos dos registros do sub-esquema ou de trabalho do mapa que foram previamente definidos.

Note-se que na Figura I.02, tem-se constantes literais como por exemplo COTISTA, TITULO, NOME, ETC... e outros campos que estão distinguidos pelo caracter "%", mas sem nenhum valor. Estes são campos variáveis que permitirão aos usuários, preencherem, ou então que através deles serão exibidos informações contidas nos registros. Nesta Figura por exemplo, "%" na frente de COTISTA permitirá ao usuário digitar o código do cotista e o empreendimento e o "%" na frente de NOME, o sistema usará para exibir o nome do cotista, cujo código é aquele que foi digitado pelo usuário.

A Figura I.03 ilustra a tela de seleção de campos que é utilizada para indicar o tratamento que será dado aos campos indicados pelo caracter seletor de campos. Assim, especifica-se E para editar cada campo selecionado, C para indicar que serão consideradas várias ocorrências do campo, D para delir todos os campos que foram selecionados, F para interromper a definição do mapa sem gerar o módulo de carga e G para gerar o módulo de carga no dicionário de dados. Pode-se especificar também o grupo de campos a serem selecionados (novos ou todos) e o tipo

IDMS-DC ON LINE MAPPING RELEASE 2.0

STEP

-- INITIAL DEFINITION --

```

ACTION:?      MAPNAME:?      VERSION:?      WRITE CONTROL CHARACTER
COPY FROM MAPNAME:?      VERSION:?      UNLOCK KYBD.:?Y RESET MDT...:?Y
COPY ACTION:? (ALL/FMT)    EDIT(Y/N)?Y   ALARM.....:?  START PRINT:?N
DICT:?        NODE:?        DEVICES:      NEWLINE...:?
                                     24X80:?Y 32X80:?Y
                                     43X80:?Y 27X132:?Y
    
```

```

                                USING RECORDS:
RECORD NAME                    VER      RECORD NAME                    VER
:?                               ?       :?                               ?
:?                               ?       :?                               ?
:?                               ?       :?                               ?
:?                               ?       :?                               ?
:?                               ?       :?                               ?
:?                               ?       :?                               ?
:?                               ?       :?                               ?
    
```

- 094 -

Figura I.01 - Tela de Definição Inicial do Mapa

%EMPR:%
%COTISTA:%
%TITULO:% % %
%***** DADOS INFORMATIVOS DO CEDENTE *****
%NOME:%
%NUMERO COTAS ...:% %BLOQUEIO:%
%**** DADOS INFORMATIVOS DO CESSIONARIO/CONTRATO****
%CODIGO:%
%DATA CONTRATO...:% %MOTIVO CONTRATO:%

Figura I.02 - "Desenho" do Mapa: Os Campos Seleccionados estão Distinguidos com o Caracter %

IDMS-DC ON LINE MAPPING RELEASE 2.0
-- FIELD SELECTION --

STEP

MAP:
VER:

ACTION :? (E) EDIT
(D) DELETE
(C) COPY?N (Y/N/INCR) OCCURS
?C (R/C) ROW/COL SEQUENCE
(F) FINISH
(G) GENERATE
GROUPING :? (N) NEW
(A) ALL
TYPE :? (A) ALL
(D) DATA
(L) LITERAL
FROM: ROW? 1 COL? 1
TO: ROW?24 COL? 80
? (Y/N) DEFAULT EXTENDED SCREEN

→ 353005 * 004 FIELDS DEFINED IN SELECTION LIST*

Figura I.03 - Tela de Seleção de Campos

de dados (todos, literais ou variáveis) que devem ser considerados na seleção.

As informações de nome e versão do mapa são fornecidas pelo sistema.

A mensagem que aparece na parte inferior da tela e que está indicada com +, informa ao projetista o número de campos que foram selecionados e devem sofrer a ação que for especificada.

A tela de edição de campos, Figura I.04, é utilizada para editar sequencialmente cada um dos campos selecionados na tela de definição de formato.

Ao ser exibida a tela, o nome do mapa, a linha e coluna em que se localiza o campo na tela e o modo de processamento, são informados ao projetista. O campo que está sendo editado aparece em tom brilhante na tela.

Os modos de processamento podem ser STEP e FAST. Se estiver definido no modo FAST, as especificações que foram indicadas serão efetuadas e a tela seguinte será exibida. No entanto, se o modo de processamento for STEP, as especificações indicadas serão efetuadas e o sistema reexibirá a tela para que o projetista "confira" o processamento efetuado, para então exibir a tela seguinte.

A modificação do modo de processamento pode ser feita pressionando-se a chave PF7, como consta na Tabela I.01.

Na tela de edição de campos podem ser definidos: atributos específicos para cada campo que está sendo editado, o nome do campo variável como ele for definido no registro do sub-esquema ou de trabalho do mapa, o índice se for o caso de um campo com índice.

Pode-se especificar ainda que a edição automática será executada e que a tela de edição de campo extendida, será selecionada. A Figura I.04a ilustra a tela preenchida.

A tela de edição de campo extendida, ilustrada na Figura I.05, é utilizada para definir atributos de edição, o nome da tabela ao qual, o campo que está sendo editado pertence, o tipo de identificação de erro associado a esse campo.

```

MAP:           DFLD      AT ROW:?   COL:?           DELETE:?(Y/N)   STEP
?A A/N ALPHA/NUM      DFLD      :?           SUB:?
?N S/N SKIP/NOSKIP   OF REC: ?           VER:?
?N Y/N MOD DATA TAG           FOR INPUT:           FOR OUTPUT:
?N D/N DETECT/NON-DETECT      ?Y Y/N DATA       ?Y Y/N/E DATA
?U U/P UNPROT/PROTECTED      ?L L/R JUSTIFY     ?N Y/N   BACKSCAN
?D B/D/K BRIGHT/DISPLAY/DARK  ? ?   C/X PAD CHARACTER
?s Y/N/S DELIMIT YES/NO/SKIP   ?           EDIT MODULE   ?           EDIT MODULE

                                     ? (Y/N) EDIT? (Y/N) EXT SCRN
:....5....10....15....20....25....30....35....40....45....50....55....60....65....70....75....80

```

Figura I.04 - Tela de Definição de Campo

```
MAP: M42051          DFLD      AT ROW: ?6  COL: ? 23      DELETE: ? (Y/N)      STEP
?N A/N ALPHA/NUM    DFLD      :?  COD-COTI          SUB: ?
?N S/N SKIP/NOSKIP  OF REC: ?  REG-WS-4205      VER: ?
?N Y/N MOD DATA TAG          FOR INPUT:          FOR OUTPUT:
?N D/N DETECT/NON-DETECT      ?Y Y/N DATA      ?Y Y/N/E DATA
?U U/P UNPROT/PROTECTED      ?L L/R JUSTIFY    ?N Y/N  BACKSCAN
?D B/D/K BRIGHT/DISPLAY/DARK  ? ?  C/X PAD CHARACTER
?s Y/N/S DELIMIT YES/NO/SKIP  ?          EDIT MODULE      ?          EDIT MODULE

                                     ?y (Y/N) EDIT? Y(Y/N) EXT SCRN
:....5....10....15....20....25....30....35....40....45....50....55....60....65....70....75....80
```

Figura I.04.a - Tela de Definição de Campo Preenchida

MAP:	DFLD	AT	ROW	COL:	REQUIRED:?	(Y/N)	STEP
?N Y/N BLINK		EXT	PIC:?				ZERO?Y Y/N
?N R/N REV-VIDEO/NORM-VIDEO		EDIT	TABLE ?		VER?	LINK?	VALID? Y/N/D
?N Y/N UNDERSCORE		CODE	TABLE ?		VER?	LINK?	
?N B/R/P/G/T/Y/W/N							
	BLUE/RED/PINK/GREEN/						
	TURQ/YELLOW/WHITE/NO	ERROR	MSG: ?	T/I/N	TEXT/ID/NULL	ID:?	TEXT:?

...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80

Figura I.05 - Tela de Definição de Campo Extendida

Ao ser exibida a tela, o nome do mapa, a linha e coluna em que se localiza o campo e o modo de processamento são fornecidos ao projetista. A Figura I.05.a ilustra a tela preenchida.

As telas de edição de campo e edição de campo extendida são "chamadas" tantas vezes quantos forem os campos selecionados na tela de definição de formato.

Os atributos a serem considerados para os campos corretos e incorretos, na hora da execução pela edição automática e manipulação de erros, podem ser definidos na tela de definição inicial extendida, ilustrada na Figura I.06. Essa tela está associada à chave de controle PF9, como consta na Tabela I.01. A Figura I.06.a ilustra a tela preenchida.

Ao concluir essas definições, deve-se selecionar a tela de seleção de campos, indicar a opção G para gerar o mapa, criando assim o módulo de carga correspondente.

O mapa exibido após ter sido gerado, contém o caracter ";" como caracter inicial de campo e "*" como caracter de limitador de campo, como ilustrado na Figura I.07. Entretanto, o mapa pode ser exibido ao usuário sem os delimitadores se for utilizado PF5, como consta na Tabela I.01.

O controle da sequência em que as telas relativas ao OLM acima descritas são "chamadas" é feito pelo projetista e existem chaves de controle, associadas a cada uma delas, como consta na Tabela I.01:

TABELA I.01 - CHAVE DE CONTROLE COM AS FUNÇÕES ASSOCIADAS PARA O OLM

CHAVE DE CONTROLE	FUNÇÃO EXECUTADA
PF1	Tela de definição inicial
PF2	Tela de definição de formato
PF3	Tela de seleção de campos
PF4	Tela de edição de campos
PF5	Tela com a imagem do mapa
PF6	Acerta o cursor
PF7	Modifica modo de processamento (STEP/FAST)
PF8	Propaga campos na tela de definição de formato
PF9	Tela de definição inicial extendida
PF10	Tela de edição de campo extendida
ENTER	Move para frente sequencialmente
CLEAR	Retrocede sequencialmente
PA1	Limpa processamento corrente
PA2	Limpa seção de mapeamento corrente


```

MAP: M42051          DFLDCOD-COTIAT  ROW 6  COL: 23          REQUIRED:?Y (Y/N)      STEP
?N Y/N BLINK          EXT PIC:? 9(6)          ZERO?Y Y/N
?N R/N REV-VIDEO/NORM-VIDEO EDIT TABLE ?      VER?      LINK?  VALID?  Y/N/D
?N Y/N UNDERSCORE    CODE TABLE ?      VER?      LINK?
?N B/R/P/G/T/Y/W/N
    BLUE/RED/PINK/GREEN/
    TURQ/YELLOW/WHITE/NO ERROR MSG: ? I T/I/N TEXT/ID/NULL  ID:? 900055  TEXT:?

```

```

...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80

```

Figura I.05.a - Tela de Definição de Campo Preenchida

IDMS-DC ON LINE MAPPING RELEASE 2.0
 ----EXTENDED INITIAL DEFINITION SCREEN----

STEP

ATTRIBUTES FOR INCORRECT FIELDS

ATTRIBUTES FOR CORRECT FIELDS

? A/N	ALPHANUMERIC/NUMERIC	? A/N	ALPHANUMERIC/NUMERIC
? P/U	PROTECTED/UNPROTECTED	? P/U	PROTECTED/UNPROTECTED
? D	DETECTABLE	? D	DETECTABLE
? D/K/B	DISPLAY/DARK/BRIGHT	? D/K/B	DISPLAY/DARK/BRIGHT
? S	SKIP	? S	SKIP
? Y/N	MDT/NOMDT	? Y/N	MDT/NOMDT
? Y/N	BLINK/NOBLINK	? Y/N	BLINK/NOBLINK
? N/R	NORMAL-VIDEO/REVERSE-VIDEO	? N/R	NORMAL-VIDEO/REVERSE-VIDEO
? Y/N	UNDERSCORE/NOUNDERSCORE	? Y/N	UNDERSCORE/NOUNDERSCORE
? B/R/P/G/T/Y/W/N	BLUE/RED/PINK/GREEN/ TURQUOISE/YELLOW/WHITE/ NOCOLOR	? B/R/P/G/T/Y/W/N	BLUE/RED/PINK/GREEN/ TURQUOISE/YELLOW/WHITE/ NOCOLOR

?N Y/N SOUND ALARM ON ERROR

Figura I.06 - Tela de Definição Inicial Estendida

;TRANSFERENCIAS

;**M42051**

```
;EMPR .....; *
;COTISTA .....; *
;TITULO .....; *; *; *
;***** DADOS INFORMATIVOS DO CEDENTE *****
;NOME .....; *
;NUMERO COTAS ...; * ;BLOQUEIO .....; *
;**** DADOS INFORMATIVOS DO CESSIONARIO/CONTRATO****
;CODIGO.....; *
;DATA CONTRATO...; * ;MOTIVO CONTRATO:; *
```

Figura I.07 - Mapa com os Caracteres Inicial e Delimitador Exibido após a Geração.

C - RECURSOS PARA DEFINIR PROCESSOS

Paralelamente ou após a definição dos mapas, pode se definir os processos que compõe o diálogo. Os processos podem ser de dois tipos: *processo pré-mapa* e *processo de resposta do mapa*.

O processo pré-mapa executa o processamento necessário antes que o mapa seja exibido ao usuário, como por exemplo a atribuição de valores iniciais aos campos e/ou registros.

O processo de resposta ao mapa executa o processamento, de acordo com uma lógica com base nos dados que foram digitados pelo usuário.

A operação de exibir as informações das áreas dos registros para a tela é denominada "mapout" e a operação de transferir os dados digitados na tela para as áreas dos registros é denominada "mapin".

Os módulos fonte dos processos que compõe o diálogo são definidos utilizando os recursos do IDD. Devem ser especificadas as cláusulas ADD MODULE, LANGUAGE IS PROCESS e MODULE SOURCE FOLLOWS.

Os comandos podem ser divididos nos seguintes tipos:

- a) Comandos aritméticos e de atribuição;
- b) Comandos que estabelecem componente estrutural de seleção e repetição;
- c) Comandos de controle de sub-rotinas;
- d) Comandos de acesso à base de dados;
- e) Comandos de modificação de mapas;
- f) Comandos de controle de filas e comandos para restaurar processamento;
- g) Comandos de fluxo de processamento;
- h) Comandos de registros lógicos;
- i) Comandos utilitários.

A seguir, será brevemente descrito cada um desses tipos, acompanhados de ilustração, quando possível. Maiores detalhes podem ser encontrados em |Cl84a|.

a) Comandos aritméticos e de atribuição permitem efetuar cálculos e atribuir valores. Assim, tem-se para a adição ADD, para subtração SUBTRACT, para multiplicação MULTIPLY, para divisão DIVIDE, para atribuição de valores MOVE e para cálculo de expressões COMPUTE. Alguns desses comandos estão ilustrados na Figura I.08. Se houver necessidade, pode-se também especificar a cláusula ROUNDED ou TRUNCATED.

b) Comandos que estabelecem componente estrutural de seleção e repetição.

Para estabelecer a componente estrutural de seleção, utiliza-se o comando IF. Esse componente permite efetuar testes, comparando: valores (=, >, <), como pode ser visto na linha 11.100 da Figura I.08; código de estado ou nome de condição, como pode ser visto na linha 10.300 da Figura I.08; nomes de campos variáveis que contenham algum valor, como pode ser visto na linha 14.000 da Figura I.08, ou expressões aritméticas.

O código de estado ou nome de condição que podem ser utilizados após a execução de comandos de acesso à base de dados, estão definidos no dicionário de dados, como mostra a Figura I.09. Comandos de acesso à base de dados são do tipo d descritos a seguir.

Quando houver uma combinação de testes, pode-se utilizar operadores booleanos (AND, OR, NOT), como ilustrado na Figura I.10. Note-se que os valores testados podem ser alfanuméricos, como por exemplo a linha 24.100 da Figura I.10 ou numéricos, como por exemplo a linha 25.000 da Figura I.10. Nessa mesma figura, tem-se um exemplo de teste de estado - se foi modificado ou não - de campos do mapa, como pode ser visto na linha 26.500.

```

00008900 DEFINE SUBROUTINE R-COMPL.
00009000 MOVE ZEROS TO ACUM-PAGTO.
00009100 MOVE ZEROS TO ACUM-COTI.
00009200 MOVE I TO INDEX1.
00009300 WHILE INDEX1 LE 24
00009400 REPEAT.
00009500 IF COD-COTI(INDEX1) NE ZEROS
00009600 DO.
00009700 COMPUTE ACUM-PAGTO = ACUM-PAGTO + VR-PAGTO (INDEX1).
00009800 COMPUTE ACUM-COTI = ACUM-COTI + 1.
00009900 MOVE D222 TO EMPRESA-42V01.
00010000 MOVE GRUPO TO GRUPO-42V01.
00010100 MOVE COD-COTI (INDEX1) TO COD-COTI-42V01.
00010200 FIND KEEP EXCLUSIVE CALC REG-A42V01-2108.
00010300 IF DB-REG-NOT-FOUND
00010400 DO.
00010500 MODIFY MAP FOR FIELD COD-COTI (INDEX1)
00010600 EDIT IS ERROR.
00010700 DISPLAY MESSAGE TEXT IS
00010800 *** 01 COTISTA NAO CADASTRADO .'.
00010900 END.
00011000 ELSE
00011100 IF INDEX1 GT 1
00011200 DO.
00011300 COMPUTE INDEX2 = INDEX1 - 1.
00011400 IF COD-COTI (INDEX1) EQ COD-COTI (INDEX2)
00011500 MODIFY MAP FOR FIELD COD-COTI (INDEX1)
00011600 EDIT IS ERROR.
00011700 END.
00011800 OBTAIN CALC REG-A42V01-2108.
00011900 KEEP LONG 'KEEPLN42V01' EXCLUSIVE REG-A42V01-2108.
00012000 IF SALDO-A-PAGAR-42V01 NE VR-PAGTO (INDEX1)
00012100 DO.
00012200 UNPROTECT COD-COTI (INDEX1).
00012300 MODIFY MAP FOR FIELD VR-PAGTO (INDEX1)
00012400 EDIT IS ERROR.
00012500 DISPLAY MESSAGE TEXT IS
00012600 *** 06 VALOR NAO CONFERE .'.
00012700 END.
00012800 ADD 1 TO INDEX1.
00012900 END.
00013000 ELSE
00013100 EXIT.
00013200 END.
00013300 IF TOT-PAGTO NE ACUM-PAGTO
00013400 DO.
00013500 MODIFY MAP FOR FIELD TOT-PAGTO EDIT IS ERROR.
00013600 DISPLAY MESSAGE TEXT IS
00013800 *** 05 TOTAL NAO CONFERE .'.
00013900 END.
00014000 IF TOT-COTI NE ACUM-COTI
00014100 DO.
00014200 MODIFY MAP FOR FIELD TOT-COTI EDIT IS ERROR.
00014300 DISPLAY MESSAGE TEXT IS
00014400 *** 06-TOTAL COTISTA NAO CONFERE .'.
00014500 END.
00014600 GOBACK.
*****

```

Figura I.08 - Comandos que Estabelecem Componentes Estrutural de Seleção, Iteração, Comandos Aritméticos e de Atribuição

VERS = 0001

CULLINET SOFTWARE
***** REQUEST

-- ADS DIALOG REPORTER

DIALOG...	D42030	MAP.....	M42030	SCHEMA...		SUBSCHEMA
VERS.	0001	VERS.	0001	VERS.	0000	AUTO-STATUS YES
DATE.	05/17/85	DATE.	05/17/85			
TIME.	174126	TIME.	174037			

AUTO-STATUS RECORD... ADSO-STAT-DEF-REC VERS. 0001 AUTO-STATUS

02	ERROR-STATUS	PIC 9(4).
88	DB-STATUS-OK	VALUE '0000'.
88	DB-END-DF-SET	VALUE '0307'.
88	DB-REC-NOT-FOUND	VALUE '0326'.
88	QUEUE-REC-NOT-FOUND	VALUE '4405'.
88	SCRATCH-REC-NOT-FOUND	VALUE '4305'.
88	SCRATCH-REC-REPLACED	VALUE '4317'.
88	DB-END-OF-INDEX	VALUE '1707'.
88	DB-INDEX-NOT-FOUND	VALUE '1726'.
88	QUEUE-ID-NOT-FOUND	VALUE '4404'.
88	SCRATCH-AREA-NOT-FOUND	VALUE '4303'.
88	DB-ANY-ERROR	VALUE '0001' THRU '9999'.

Figura I.09 - Registro com Código de Condição de Estado.

- III -


```

00022200      MOVE ZEROS          TO DGH-CGC (13).
00022300      MOVE 1           TO INDEX2.
00022400      MOVF 5           TO INDEX3.
00022500      CALL R-CLCSUM.
00022600      MOVE DV-CALC1     TO DGH-CGC (13).
00022700      MOVE ZEROS      TO SOMA.
00022800      MOVE 1           TO INDEX2.
00022900      MOVE 6           TO INDEX3.
00023000      CALL R-CLCSOM.
00023100      IF DV-CALC NE DGH-CGC (13) OR
00023200      DV-CALC1 NE DGH-CGC (14)
00023300      MODIFY MAP FOR
00023400      (CIC-CGC-42V01 INDIC-FIS-JUR1-42V01)
00023500      EDIT IS ERROR.
00023600      END.
00023700      IF ESTA-RG-42V01 EQ ' '
00023800      MODIFY MAP FOR (NUM-RG-42V01 ESTA-RG-42V01)
00023900      EDIT IS ERROR.
00024000      IF ESTA-CIVIL-42V01 EQ ' ' OR
00024100      ESTA-CIVIL-42V01 NE 'S' AND
00024200      ESTA-CIVIL-42V01 NE 'C' AND
00024300      ESTA-CIVIL-42V01 NE 'D' AND
00024400      ESTA-CIVIL-42V01 NE 'V' AND
00024500      ESTA-CIVIL-42V01 NE 'O' AND
00024600      ESTA-CIVIL-42V01 NE SPACES
00024700      MODIFY MAP FOR ESTA-CIVIL-42V01 EDIT IS ERROR.
00024800      IF NACION-42V01 EQ ' '
00024900      MODIFY MAP FOR NACION-42V01 EDIT IS ERROR.
00025000      IF TIPO-PAGTO-42V01 NE 1 AND
00025100      TIPO-PAGTO-42V01 NE 2 AND
00025200      TIPO-PAGTO-42V01 NE 3 AND
00025300      TIPO-PAGTO-42V01 NE 4 AND
00025400      TIPO-PAGTO-42V01 NE 5
00025500      MODIFY MAP FOR TIPO-PAGTO-42V01 EDIT IS ERROR.
00025600      IF DT-CADAS EQ ZEROS
00025700      MODIFY MAP FOR FIELD DT-CADAS EDIT IS ERROR.
00025800      ELSE
00025900      DO.
00026000      MOVE DT-CADAS TO DT-AUX.
00026100      CALL R-CONDAT.
00026200      IF INDEX1 EQ 1
00026300      MODIFY MAP FOR FIELD DT-CADAS EDIT IS ERROR.
00026400      END.
00026500      IF FIELD TIPO-PAGTO-42V01 IS CHANGED
00026600      DO.
00026700      IF TIPO-PAGTO-42V01 EQ 4
00026800      DO.
00026900      MOVE ZEROS TO BCO-42V01.
00027000      MOVE ZEROS TO AGE-42V01.
00027100      MOVE ZEROS TO CONTA-CORRE-42V01.
00027200      GOBACK.
00027300      END.
00027400      ELSE
00027500      NEXT.
00027600      END.

```

Figura I.10 - Combinação de Testes e Utilização de Operadores Booleanos

Para estabelecer uma componente estrutural de repetição utiliza-se o comando WHILE, como ilustrado na linha 9.300 da Figura I.08.

Após as componentes estruturais de seleção ou repetição podem seguir uma ou mais componentes de sequência, lembrando no entanto que se houver mais de uma componente de sequência, após a componente de seleção, estas devem ser precedidas pelo comando DO e seguidas pelo comando END. Pode também haver um aninhamento da componente estrutural de seleção, que devem ser codificados utilizando a tabulação, permitindo assim uma melhor compreensão e legibilidade. Quando houver necessidade de deixar a componente estrutural de repetição criada pelo WHILE, por ter satisfeito alguma condição, usa-se o comando EXIT. Na linha 13.100 da Figura I.08 está ilustrada a utilização do comando EXIT.

- c) Comandos de controle de sub-rotinas: são comandos que permitem chamar, definir ou retornar a uma sub-rotina.

Um processo de resposta ao mapa pode ter uma ou mais sub-rotinas.

Para passar o controle para uma sub-rotina tem-se o comando CALL. A rotina que é chamada é definida utilizando o comando DEFINE. Na chamada, o controle é passado para o começo da definição da sub-rotina e continua executando os comandos que a definem, inclusive uma chamada a outra sub-rotina, até que se encontre o comando GOBACK, que retorna o controle para a rotina que a chamou. O nome da sub-rotina que aparece no CALL ou DEFINE pode ter até 8 caracteres. A Figura I.11 ilustra um exemplo de definição de sub-rotina, inclusive com chamada a uma outra sub-rotina.

- d) Comandos de acesso à base de dados: são comandos utilizados para recuperar e/ou atualizar registros da base de dados ou de arquivos VSAM.

```

00005000 DEFINE SUBROUTINE R-ALTERA.
00005100     CALL R-CONPAR.
00005200     IF ANY FIELDS ARE IN ERROR
00005300         DISPLAY.
00005400     MOVE COD-EMPRESA TO EMPRESA-1014.
00005500     MOVE 25         TO REGTIPO-1014.
00005600     MOVE 010        TO REGCOMP-1014.
00005700     FIND CALC REG-A10104.
00005800     OBTAIN CURRENT REG-A10104.
00005900     PUT SCRATCH FROM REG-A10104 LENGTH 40.
00006000     MOVE TX-LIM TO CONTEUDO-1014.
00006100     MODIFY REG-A10104.
00006200     MOVE 26         TO REGTIPO-1014.
00006300     MOVE 014        TO REGCOMP-1014.
00006400     FIND CALC REG-A10104.
00006500     OBTAIN CURRENT REG-A10104.
00006600     PUT SCRATCH FROM REG-A10104 LENGTH 40.
00006700     MOVE TAB-AUX (4) TO CONTO-RATEIO (4).
00006800     MOVE TAB-RATEIO (4) TO CONTEUDO-1014.
00006900     MODIFY REG-A10104.
00007000     MOVE '1' TO FLAG-COMMIT.
00007100     DISPLAY MESSAGE TEXT
00007200     *** 02 - OPERACAO ENCERRADA. TECLE CLEAR     .'.
00007300 GOBACK.
00007400 *****
00007500 DEFINE SUBROUTINE R-CONSUL.
00007600     DISPLAY MESSAGE TEXT IS
00007700     *** 01- OPERACAO ENCERRADA. TECLE CLEAR     .'.
00007800 GOBACK.
00007900 *****
00008000 DEFINE SUBROUTINE R-CONPAR.
00008100     IF MES-R (12) EQ ZEROS OR
00008200         MES-R (12) GT 12 OR
00008300         ANO-R (12) LT 83
00008400         MODIFY MAP FOR (MES-R (12) ANO-R (12)) EDIT IS ERROR.
00008500     IF TX EQ ZEROS OR
00008600         LIM EQ ZEROS
00008700         MODIFY MAP FOR (TX LIM ) EDIT IS ERROR.
00008800     COMPUTE INDEX1 = ANO-R (11) + 1.
00008900     COMPUTE INDEX2 = MES-R (11) + 1.
00009000     IF MES-R (12) LT MES-R (11)
00009100         DO.
00009200         IF MES-R (11) NE 12 AND
00009300             ANO-R (12) NE INDEX1
00009400             MODIFY MAP FOR (MES-R (12) ANO-R (12)) EDIT IS ERROR.
00009500     END.
00009600     ELSE
00009700         IF ANO-R (12) NE ANO-R (11) AND
00009800             MES-R (12) NE INDEX2
00009900             MODIFY MAP FOR (MES-R (12) ANO-R (12)) EDIT IS ERROR.
00010000 GOBACK.
00010100 *****

```

Figura I.11 - Definição de Sub-rotina e Chamadas a outras Sub-rotinas.

A preparação da área da base de dados que será utilizada pela aplicação, pode ser feita através do comando READY, podendo ser especificados os modos UPDATE ou RETRIEVAL, como pode ser visto na linha 100 da Figura I.12.

Uma ocorrência de um registro pode ser localizada, utilizando o comando FIND. Se além de localizar, for tornar o conteúdo disponível, utiliza-se o comando OBTAIN. O acesso a uma ocorrência pode ser feito de várias formas e para cada uma existe um formato específico do comando.

Uma ocorrência de um registro pode ser localizada através da chave de armazenamento do registro, no formato FIND/OBTAIN CALC. Nesse caso, devem ser movidos para o campo chave, os valores ou nomes de campos que contenham os valores da chave do registro procurado, como ilustrado nas linhas 9.900, 10.000, 10.100 da Figura I.08. Essa é uma forma de obter aleatoriamente um registro.

Para se ter acesso à ocorrência do registro que se encontra na unidade de execução, o formato é FIND/OBTAIN CURRENT como pode ser visto na linha 2.700 da Figura I.12.

O acesso à ocorrência de um registro, em uma relação estabelecida entre registros mestres e registros membros, pode ser feito através de FIND/OBTAIN WITHIN SET *nome da relação*, podendo inclusive especificar se é permitida a ocorrência de um dos nomes de condição de erro, ilustrados na Figura I.09. Um exemplo de FIND/OBTAIN WITHIN SET está ilustrado na linha 3.600 da Figura I.12.

Esse comando apresenta formatos derivados: FIND/OBTAIN OWNER para obter diretamente o registro mestre e FIND/OBTAIN WITHIN SET USING *chave de ordenação*, para ter acesso a um registro membro em uma relação ordenada. Esses formatos derivados aplicam-se apenas aos registros definidos em base de dados.

O acesso a um registro em uma determinada área, pode ser feito através de FIND/OBTAIN WITHIN AREA.

```

00000100      READY AREA-A10104 USAGE-MODE IS UPDATE.
00000200      IF CHECKPOINT
00000300          DD.
00000400          KEEP LONG ALL RELEASE.
00000500          DELETE SCRATCH ALL.
00000600          MOVE '0' TO FLAG-COMMIT.
00000700          END.
00000800      IF COD-OPERACAO EQ 1
00000900          CALL R-INCLUS.
00001000      IF COD-OPERACAO EQ 2
00001100          CALL R-ALTERA.
00001200      IF COD-OPERACAO EQ 3
00001300          CALL R-CONSUL.
00001400      DISPLAY MESSAGE TEXT IS
00001500          *** 00 - TIPO OPERACAO INDEFINIDO -TECLE CLEAR*.
00001600      *****
00001700      DEFINE SUBROUTINE R-INCLUS.
00001800          CALL R-COMPAR.
00001900          IF ANY FIELDS ARE IN ERROR
00002000              DISPLAY.
00002100          MOVE 0          TO FLAG-ATUALIZACAO.
00002200          MOVE 0          TO FLAG (4).
00002300          MOVE COD-EMPRESA TO EMPRESA-1014.
00002400          MOVE 26         TO REGTIPO-1014.
00002500          MOVE 011        TO REGCOMP-1014.
00002600          FIND CALC REG-A10104.
00002700          OBTAIN CURRENT REG-A10104.
00002800          PUT SCRATCH FROM REG-A10104 LENGTH 40.
00002900          MOVE 1 TO INDEX1.
00003000          MOVE TAB-AUX (INDEX1) TO CONTD-RATEIO (INDEX1).
00003100          MOVE TAB-RATEIO (INDEX1) TO CONTEUDO-1014.
00003200          MODIFY REG-A10104.
00003300          ADD 1 TO INDEX1.
00003400          WHILE INDEX1 LE 4
00003500              REPEAT.
00003600                  OBTAIN NEXT REG-A10104 WITHIN SET-A10104 ALLOWING
00003700                      (DB-REC-NDT-FOUND).
00003800                  PUT SCRATCH FROM REG-A10104 LENGTH 40.
00003900                  MOVE TAB-AUX (INDEX1) TO CONTD-RATEIO (INDEX1).
00004000                  MOVE TAB-RATEIO (INDEX1) TO CONTEUDO-1014.
00004100                  MODIFY REG-A10104.
00004200                  ADD 1 TO INDEX1.
00004300              END.
00004400          MOVE 1 TO FLAG-COMMIT.
00004500          MODIFY MAP FOR (RATEIO-R (12) ANO-R (12) MES-R (12)) OPTIONAL.
00004600          DISPLAY MESSAGE TEXT
00004700          *** 01 - OPERACAO ENCERRADA.TECLE CLEAR          *.
00004800      GOBACK.
00004900      *****

```

Figura I.12 - Comandos de Acesso a Base de Dados, de Atribuição, de Modificação de Mapa.

Na utilização dos comandos FIND/OBTAIN, pode-se utilizar a cláusula KEEP que estabelece um estado de "do mínio" sobre a ocorrência à qual se teve acesso. Assim, para especificar que as outras unidades em execução podem ter acesso mas não podem atualizar o registro, utiliza-se KEEP. Para especificar que as demais unidades em execução não terão acesso e também não poderão atualizar o registro, utiliza-se KEEP EXCLUSIVE. A Figura I.13 contém, na linha 3.700, uma ilustração de KEEP EXCLUSIVE.

Para o armazenamento de uma ocorrência do registro deve-se mover todos os dados contidos na área do registro do diálogo, para a área da base de dados e emitir o comando STORE.

Para efetuar a modificação do conteúdo de um registro, deve-se mover os dados que estão na área do registro no diálogo, relativos àqueles campos que sofreram alteração, para a área da base de dados e emitir o comando MODIFY. Deve ser lembrado no entanto, que para utilizar o comando MODIFY, o modo de utilização especificado no comando READY deve ser UPDATE. Exemplos dos comandos STORE e MODIFY estão ilustrados nas linhas 6.500 e 4.200 da Figura I.13, respectivamente.

Para delir uma ocorrência do registro, tem-se o comando ERASE, ilustrado na linha 4.700 da Figura I.14. Além disso, tem-se o comando ACCEPT DBKEY para o acesso a uma chave de base de dados, CONNECT para conectar uma ocorrência de um registro da base de dados, DISCONNECT para desconectar uma ocorrência de um registro da base de dados, etc.

- e) Comandos de modificação de mapas: são comandos que permitem especificar atributos ao mapa que está associado ao diálogo. Esses atributos, que estão ilustrados na Figura I.15, se referem, por exemplo, à indicação da posição do cursor no mapa que está ilustrado na linha 6400, especificar os campos que serão liberados ou não para o usuário efetuar modificações na hora da execução como ilustrado na linha 6200. Po

```

00001300 DEFINE SUBROUTINE R-TRANSF.
00001400 IF NAQATUALIZA
00001500 DO.
00001600 CALL R-PRETRA.
00001700 MOVE '1' TO FLAG-ATUALIZACAO.
00001800 UNPROTECT ALL EXCEPT (GRUPO COD-COTI TITU-AUX
00001900 QT-COTA-42V02 BLDQ-42V01
00002000 NOME-EMPRESA NOME-AUX (1)
00002100 NOME-AUX (2)) TEMPORARY.
00002200 MODIFY MAP PERMANENT FOR (DT-CONTR MOT-CONTR
00002300 COD-COTI-NOVO)
00002400 REQUIRED.
00002500 MODIFY MAP TEMPORARY CURSOR AT FIELD COD-COTI-NOVO.
00002600 DISPLAY.
00002700 END.
00002800 CALL R-CONTRA.
00002900 IF ANY FIELDS ARE IN ERROR
00003000 DISPLAY.
00003100 MOVE 0222 TO EMPRESA-42V02.
00003200 MOVE GRUPO TO GRUPO-42V02.
00003300 MOVE SER-AUX TO SER-42V02.
00003400 MOVE FASE-AUX TO FASE-42V02.
00003500 MOVE CONTR-AUX TO CONTR-42V02.
00003600 MOVE COD-COTI TO COD-COTI-42V02.
00003700 FIND KEEP EXCLUSIVE CALC REG-A42V02-2109.
00003800 OBTAIN CURRENT REG-A42V02-2109.
00003900 PUT SCRATCH AREA ID '10200' FROM REG-A42V02-2109 LENGTH 60.
00004000 MOVE QT-COTA-42V02 TO QT-COTA-AUX.
00004100 MOVE ZEROS TO QT-COTA-42V02.
00004200 MODIFY REG-A42V02-2109.
00004300 MOVE DIA-CONTR-42V02 TO DIA-AUX-TRANS.
00004400 MOVE MES-CONTR-42V02 TO MES-AUX-TRANS.
00004500 MOVE ANO-CONTR-42V02 TO ANO-AUX-TRANS.
00004600 MOVE 0222 TO EMPRESA-42V02.
00004700 MOVE GRUPO TO GRUPO-42V02.
00004800 MOVE COD-COTI-NOVO TO COD-COTI-42V02.
00004900 MOVE SER-AUX TO SER-42V02.
00005000 MOVE FASE-AUX TO FASE-42V02.
00005100 MOVE CONTR-AUX TO CONTR-42V02.
00005200 MOVE QT-COTA-AUX TO QT-COTA-42V02.
00005300 MOVE DIA-CONTR TO DIA-CONTR-42V02.
00005400 MOVE MES-CONTR TO MES-CONTR-42V02.
00005500 MOVE ANO-CONTR TO ANO-CONTR-42V02.
00005600 MOVE COD-COTI TO COD-COTI-ANT-42V02.
00005700 MOVE CONTR-AUX TO CONTR-ANT-42V02.
00005800 MOVE SER-AUX TO SER-ANT-42V02.
00005900 MOVE FASE-AUX TO FASE-ANT-42V02.
00006000 MOVE DIA-AUX-TRANS TO DIA-CONTR-ANT-42V02.
00006100 MOVE MES-AUX-TRANS TO MES-CONTR-ANT-42V02.
00006200 MOVE ANO-AUX-TRANS TO ANO-CONTR-ANT-42V02.
00006300 MOVE '0' TO FLAG-ATUALIZACAO.
00006400 MOVE MOT-CONTR TO MOT-CONTR-42V02.
00006500 STORE REG-A42V02-2109.
00006600 PUT SCRATCH AREA ID '10200' FROM REG-A42V02-2109 LENGTH 60.
00006700 MOVE 0222 TO EMPRESA-42V01.
00006800 MOVE GRUPO TO GRUPO-42V01.
00006900 MOVE COD-COTI TO COD-COTI-42V01.
00007000 FIND KEEP EXCLUSIVE CALC REG-A42V01-2108.
00007100 OBTAIN CALC REG-A42V01-2108.
00007200 PUT SCRATCH AREA ID '10100' FROM REG-A42V01-2108 LENGTH 450.
00007300 COMPUTE QT-TITU-42V01 = QT-TITU-42V01 - 1.
00007400 COMPUTE QT-COTA-42V01 = QT-COTA-42V01 - QT-COTA-AUX.
00007500 MODIFY REG-A42V01-2108.
00007600 MOVE 0222 TO EMPRESA-42V01.
00007700 MOVE GRUPO TO GRUPO-42V01.
00007800 MOVE COD-COTI-NOVO TO COD-COTI-42V01.
00007900 FIND KEEP EXCLUSIVE CALC REG-A42V01-2108.
00008000 OBTAIN CALC REG-A42V01-2108.
00008100 PUT SCRATCH AREA ID '10100' FROM REG-A42V01-2108 LENGTH 450.
00008200 COMPUTE QT-TITU-42V01 = QT-TITU-42V01 + 1.
00008300 COMPUTE QT-COTA-42V01 = QT-COTA-42V01 + QT-COTA-AUX.
00008400 MODIFY REG-A42V01-2108.
00008500 MOVE '1' TO FLAG-COMMIT.
00008600 UNPROTECT (TITU-AUX GRUPO COD-COTI) TEMPORARY.
00008700 MODIFY MAP PERMANENT FOR (DT-CONTR MOT-CONTR
00008800 COD-COTI-NOVO) OPTIONAL.
00008900 DISPLAY MESSAGE CODE IS 900051.
00009000 GOBACK.

```

Figura I.13 - Comandos: FIND com Cláusula KEEP, STORE, MODIFY, Modificação de mapas.

```

00003600 *****
00003700 DEFINE SUBROUTINE R-ALTTIT.
00003800 GET SCRATCH AREA ID '10200' INTO REG-A42V02-2109
00003900 MAX LENGTH 60 FIRST.
00004000 FIND KEEP EXCLUSIVE CALC REG-A42V02-2109.
00004100 MODIFY REG-A42V02-2109.
00004200 GET SCRATCH AREA ID '10200' INTO REG-A42V02-2109
00004300 MAX LENGTH 60 NEXT ALLOWING (DB-ANY-ERROR).
00004400 FIND KEEP EXCLUSIVE CALC REG-A42V02-2109.
00004500 IF DB-ANY-ERROR
00004600 GOBACK.
00004700 ERASE REG-A42V02-2109.
00004800 GOBACK.
00004900 *****
00005000 DEFINE SUBROUTINE R-SAIDA.
00005100 MOVE '1' TO FLAG-COMMIT.
00005200 KEEP LONG ALL RELEASE.
00005300 DISPLAY MESSAGE TEXT IS '** 52 - OPERACAO ANTERIOR IGNORADA'.
00005400 GOBACK.
00005500 *****

```

Figura I.14 - Comandos Para Restaurar Processamento

```

00005600 *****
00005700 DEFINE SUBROUTINE R-ALTERA.
00005800 IF NAATUALIZA
00005900 DO.
00006000 CALL R-PREBAG.
00006100 MOVE '1' TO FLAG-ATUALIZACAO.
00006200 UNPROTECT ALL EXCEPT (BCD AGE NOME-EMPRESA
00006300 DECOD-OPERACOES-02) TEMPORARY.
00006400 MODIFY MAP TEMPORARY CURSOR AT FIELD DESCR-42V03.
00006500 PUT SCRATCH FROM REG-A42V03-2110 LENGTH 60.
00006600 KEEP LONG 'KEEPLN42V03' EXC REG-A42V03-2110.
00006700 DISPLAY.
00006800 END.
00006900 CALL R-CONBAG.
00007000 IF ANY FIELDS ARE IN ERROR
00007100 DISPLAY.
00007200 MOVE '0' TO FLAG-ATUALIZACAO.
00007300 FIND CURRENT REG-A42V03-2110.
00007400 MODIFY REG-A42V03-2110.
00007500 MOVE '1' TO FLAG-COMMIT.
00007600 UNPROTECT (BCD AGE) TEMPORARY.
00007700 DISPLAY MESSAGE CODE IS 900051.
00007800 GOBACK.

```

Figura I.15 - Comandos de Controle - DISPLAY

de-se especificar os campos que o usuário será obrigado a digitar na hora da execução como ilustrado na linha 2.200 da Figura I.13. Pode também ser utilizado para indicar campos que contenham erros, como ilustrado na linha 13.600 da Figura I.08.

A especificação dos atributos pode ser temporária ou permanente. Numa especificação temporária, os atributos têm efeito apenas para a próxima exibição do mapa, ao passo que em uma especificação permanente, os atributos têm efeito enquanto o diálogo estiver sendo executado.

- f) Comandos de controle de fila e comandos para restaurar processamento: são comandos que permitem alocar e ter acesso a registros que são armazenados no dicionário de dados. O conteúdo dos registros podem ser transferidos entre diferentes operações "mapin". Para delir a fila, ou um registro da fila, utiliza-se DELETE QUEUE, para armazenar um registro na fila PUT QUEUE e para transferir um registro da fila para a área de registro no diálogo GET QUEUE.

O conteúdo desses registros podem ser transferidos entre operações "mapin", concorrentes ou subsequentes, executadas num mesmo terminal ou em terminais diferentes.

Pode-se também efetuar a restauração de processamento. Para tanto, utiliza-se o comando PUT SCRATCH para armazenar sequencialmente os registros recuperados em uma área, com nomes de até 8 caracteres, para eventual utilização após uma operação "mapin". Para registros de tipos diferentes, identificam-se áreas distintas. O comando GET SCRATCH é utilizado, para transferir o registro da área onde estava armazenado para a área do registro correspondente, recuperando assim o estado anterior à modificação sofrida pelo registro. O conteúdo desses registros, pode ser transferido de uma operação "mapin" para outra, mas quando executadas sob o mesmo terminal.

Nos comandos PUT/GET SCRATCH, pode ser especificado o comprimento do registro que está sendo considerado. No comando GET SCRATCH, pode ser especificado a ordem lógica (FIRST, NEXT) em que se encontra o registro.

A Figura I.13 ilustra o comando PUT SCRATCH, nas linhas 6.600 e 7.200. Nesse exemplo, estão sendo considerados dois tipos de registros e portanto duas áreas distintas. Estão também sendo identificados os comprimentos dos respectivos registros.

A Figura I.14 ilustra o comando GET SCRATCH, na linha 3.800, com a especificação do comprimento e da ordem lógica.

- g) Comandos de controle de fluxo de processamento: são comandos utilizados para passar o controle para diferentes diálogos.

Assim, tem-se por exemplo, LINK para passar o controle para um diálogo do próximo nível da estrutura, TRANSFER para um diálogo do mesmo nível, LEAVE para deixar a estrutura, etc.

O comando DISPLAY pode ser utilizado para exibir o mapa como consta na linha 6700 da Figura I.15, ou para exibir mensagem de erro como consta na linha 7700 da Figura I.15.

- h) Comandos de registros lógico: Um registro lógico é constituído por campos selecionados de vários tipos de registros, que são concatenados para permitir o acesso a essas informações de uma única vez. É definido num sub-esquema pelo DBA. As restrições de acesso aos registros lógicos podem ser estabelecidas no momento de sua definição.

Assim, para delir ocorrências de registros lógicos, tem-se o comando ERASE, para modificar o conteúdo MODIFY, para armazenar STORE, para recuperar ocorrências OBTAIN e para execução de procedimentos adicionais após a execução de ERASE, MODIFY, STORE ou OBTAIN o comando ON.

i) Comandos utilitários: são comandos que permitem ter minar a execução de uma aplicação de forma anormal pelo ABORT, transmitir dados de um diálogo para im pressora, reiniciar áreas dos registros pelo INITIA LIZE, etc... O comando INITIALIZE está ilustrado na Figura I.15.a linha 400.

```

00000100      READY AREA-A10104 USAGE-MODE IS UPDATE.
00000200      IF NAOCHECKPOINT
00000300      DO.
00000400          INITIALIZE (REG-A10104).
00000500          MOVE ZEROS TO IX.
00000600          MOVE ZEROS TO LIM.
00000700          MOVE 1 TO INDEX1.
00000800          WHILE INDEX1 LE 12
00000900              REPEAT.
00001000                  MOVE ZEROS TO RATEID-R (INDEX1).
00001100                  MOVE ZEROS TO MES-R (INDEX1).
00001200                  MOVE ZEROS TO ANO-R (INDEX1).
00001300                  ADD 1 TO INDEX1.
00001400              END.
00001500          MOVE '0' TO FLAG-ATUALIZACAO.
00001600          KEEP LONG ALL RELEASE.
00001700          DISPLAY MESSAGE TEXT IS
00001800              *** 01 - OPERACAO ENCERRADA. TEGLE CLEAR
00001900          END.
00002000      IF COD-OPERACAO EQ 1
00002100      DO.
00002200          GET SCRATCH INTO REG-A10104 MAX LENGTH 40 FIRST.
00002300          WHILE DB-STATUS-DK
00002400              REPEAT.
00002500                  FIND KEEP EXCLUSIVE CALC REG-A10104.
00002600                  MODIFY REG-A10104.
00002700                  GET SCRATCH INTO REG-A10104 MAX LENGTH 40 NEXT.
00002800              END.
00002900          CALL R-SAIDA.
00003000      END.
00003100      IF COD-OPERACAO EQ 2
00003200      DO.
00003300          GET SCRATCH INTO REG-A10104 MAX LENGTH 40 FIRST.
00003400          FIND KEEP EXCLUSIVE CALC REG-A10104.
00003500          MODIFY REG-A10104.
00003600          GET SCRATCH INTO REG-A10104 MAX LENGTH 40 NEXT.
00003700          FIND KEEP EXCLUSIVE CALC REG-A10104.
00003800          MODIFY REG-A10104.
00003900          CALL R-SAIDA.

```

Figura I.15,a - Comando INITIALIZE

D) PROCEDIMENTOS PARA GERAR DIÁLOGOS

O diálogo é gerado a partir do momento em que estiverem concluídas as definições do mapa, registros e processos, utilizando-se de uma ferramenta disponível no ADS/O - O AD SG. O projetista digita o código AD SG e tem-se início a sessão que irá gerar um diálogo. Nessa sessão é utilizado um conjunto de telas que serão descritas a seguir.

A tela de definição de diálogo, Figura I.16, é utilizada para definir a ação, o nome do diálogo, o nome do sub-esquema, esquema e mapa associados a esse diálogo. A Figura I.16.a ilustra uma tela preenchida.

A tela de definição de pré-mapa, Figura I.17, é utilizada para definir o nome do processo pré-mapa associado ao diálogo, cujo nome é fornecido na tela junto com o nome do mapa e sub-esquema. Essa tela é selecionada ao assinalar a opção PREMAP PROCESS, na tela de definição do diálogo.

A Figura I.17.a ilustra uma tela preenchida com o nome do processo pré-mapa associado.

Um processo pré-mapa pode ser delido, bastando para isso especificar os campos correspondentes na tela de definição do pré-mapa, como ilustra a Figura I.17.b.

A tela de definição do processo de resposta ao mapa, Figura I.18, é utilizada para definir o nome do processo de resposta ao mapa, a chave de controle que está associada ao processo e a opção que especifica se o processo deve continuar ou não se for identificado algum erro, durante a entrada de dados. Esse processo estará associado ao diálogo cujo nome é fornecido junto com o nome do mapa e do sub-esquema. Essa tela é selecionada ao assinalar a opção NEW RESPONSE e pode ser "chamada" para definir cada processo de resposta ao mapa que estiver associado ao diálogo.

A Figura I.18.a ilustra uma tela preenchida com o nome do processo resposta ao mapa associado.

Um processo de resposta ao mapa pode ser delido, bastando para isso especificar os campos correspondentes na tela de definição do processo de resposta ao mapa.

CULLINET SOFTWARE
 ADS/ONLINE REL DIALOG GENERATOR ***DIALOG DEFINITION***

ACTION:?	(ADD/MOD/DEL)	DICTNAME:?	DICTNODE:?	MODE:?
DIALOG.....?		VER....:?	MAINLINE:?	(YES/NO)
SUBSCHEMA.:?		SCHEMA:?	VERSION:?	
MAPNAME....:?		VER....:?		
AUTOSTATUS:?	(YES/NO)	RECORD:?		VER:?
SELECT NEXT FUNCTION:				
?_ RESPONSE:?		?_ PREMAP PROCESS		?_ NEW RESPONSE
?_ DISPLAY		?_ DISPLAY MAP		?_ NEW COPY/WORK RECORD
?_ PRINT				?_ UPDATE MAP
?_ DISPLAY RECORD:?				VER:?
?_ GENERATE		?_ QUIT		?_ SUSPEND
?_ DIALOG OPTIONS		?_ SWITCH TASK:?		? (O-OLD/N-NEW)

Figura I.16 - Tela Definição de Diálogo

ADS/ONLINE REL

CULLINET SOFTWARE
DIALOG GENERATOR

DIALOG DEFINITION

ACTION:? (ADD/MOD/DEL) DICTNAME:? DICTNODE:? MODE:?
DIALOG.....? D42031 VER...:? MAINLINE:? (YES/NO)
SUBSCHEMA..? SUBS4200 SCHEMA:? SHOPSH42 VERSION:?
MAPNAME...? M42031 VER...:?
AUTOSTATUS:? (YES/NO) RECORD:? VER:?

SELECT NEXT FUNCTION: ?_ PREMAP PROCESS ?_ NEW RESPONSE
?_ RESPONSE:? ?_ NEW COPY/WORK RECORD
?_ DISPLAY ?_ DISPLAY MAP ?_ UPDATE MAP
?_ PRINT
?_ DISPLAY RECORD:? VER:?
?_ GENERATE ?_ QUIT ?_ SUSPEND
?_ DIALOG OPTIONS ?_ SWITCH TASK:? ? (O-OLD/N-NEW)

Figura I.16.a - Tela Definição de Diálogo Preenchida

ADS/ONLINE REL

DIALOG GENERATOR

PREMAP PROCESS DEFINITION

DIALOG: D42031 MAPNAME: M42031 SUBSCHEMA: SUBS4200 MODE:

DELETE:? Y (Y/YES)

PREMAP PROCESS NAME:? PM-D42031

VERSION:?

SELECT NEXT FUNCTION: ?_ PREMAP PROCESS ?_ NEW RESPONSE
?_ RESPONSE:? ?_ DISPLAY MAP ?_ NEW COPY/WORK RECORD
?_ DISPLAY ?_ DIALOG ?_ UPDATE PROCESS
?_ PRINT ?_ QUIT VER:?
?_ DISPLAY RECORD:? ?_ SWITCH TASK:? ?_ SUSPEND
?_ GENERATE ? (0-OLD/N-NEW)
?_ DIALOG OPTIONS

Figura I.17.b - Tela de Definição do Processo Pré-mapa com Informações para Deleção

```

DIALOG:          MAPNAME:          SUBSCHEMA:          MODE:
                                     ?
DELETE: ?        (Y/YES)

RESPONSE PROCESS NAME...: ?
CONTROL KEY.....: ?
RESPONSE FIELD VALUE...: ?
EXECUTE ON EDIT ERRORS.: ?        (YES/NO)

                                     VERSION: ?

SELECT NEXT FUNCTION:
?_ RESPONSE: ?
?_ DISPLAY
?_ PRINT
?_ DISPLAY RECORD: ?
?_ GENERATE
?_ DIALOG OPTIONS

?_ PREMAP PROCESS
?_ DISPLAY MAP
?_ DIALOG
?_ QUIT
?_ SWITCH TASK: ?

?_ NEW RESPONSE
?_ NEW COPY/WORK RECORD
?_ UPDATE PROCESS
VER: ?
?_ SUSPEND
? (O-OLD/N-NEW)

```

Figura I.18 - Tela de Definição do Processo de Resposta ao Mapa

ADS/ONLINE REL DIALOG GENERATOR ***RESPONSE PROCESS DEFINITION***

DIALOG: D42031 MAPNAME: M42031 SUBSCHEMA: SUBS4200 MODE: ?

DELETE: ? (Y/YES)

RESPONSE PROCESS NAME...? RP-D42031 - 01 VERSION: ?
 CONTROL KEY.....? ENTER
 RESPONSE FIELD VALUE...?
 EXECUTE ON EDIT ERRORS.:? NO (YES/NO)

SELECT NEXT FUNCTION: ?_ PREMAP PROCESS ?_ NEW RESPONSE
 ?_ RESPONSE: ? ?_ DISPLAY MAP ?_ NEW COPY/WORK RECORD
 ?_ DISPLAY ?_ DIALOG ?_ UPDATE PROCESS
 ?_ PRINT VER: ?
 ?_ DISPLAY RECORD: ? ?_ QUIT ?_ SUSPEND
 ?_ GENERATE ?_ SWITCH TASK: ? ? (O-OLD/N-NEW)
 ?_ DIALOG OPTIONS

Figura I.18.a - Tela de Definição de Processo de Resposta ao Mapa Preenchido

Na compilação dos processos, podem ocorrer erros. Nesse caso, seleciona-se a opção DISPLAY ou PRINT, na tela que está sendo exibida com o erro. Ao selecionar DISPLAY, o módulo do processo que contém o erro será exibido na tela para que o projetista identifique os erros. Ao final da exibição, o projetista seleciona a opção SUSPEND para suspender a compilação e então proceder às correções dos erros.

A correção dos erros é feita utilizando-se os recursos do IDD, especificando o comando DISPLAY MODULE *nome do processo* AS SYNTAX. Ao ser exibido o módulo do processo, deve-se alterar a cláusula ADD MODULE para MOD MODULE, delir o comando DISPLAY MODULE *nome do processo* AS SYNTAX e efetuar as correções necessárias. Após as correções retornar à compilação do processo. Esse ciclo de correção de erros pode ser efetuada tantas vezes quantas forem necessárias.

Para gerar o diálogo criando o módulo de carga correspondente seleciona-se a opção GENERATE.

Existem aplicações que utilizam duas telas adicionais e que não estão ilustradas, antes de selecionar a opção GENERATE. Uma dessas telas, a tela de definição de opções é utilizada para especificar opções quanto à atividade de documentação da utilização do potencial da base de dados, para cada diálogo, ou quanto às informações de mapas com inúmeras ocorrências de um determinado campo. A outra tela, a tela de definição de nova cópia/registro de trabalho para especificar os registros para os quais devem ser criadas novas áreas na hora da execução da aplicação ou se estes são registros de trabalho.

A aplicações ADS/O são executadas pelo sistema de execução ADS/O, que carrega os módulos de carga exigidos e prepara as áreas dos registros da aplicação. O sistema de execução interpreta e processa os dados e processos introduzidos no terminal, de acordo com as definições armazenadas no dicionário de dados. Permite várias aplicações serem executadas ao mesmo tempo.

Uma aplicação pode eventualmente exigir que sejam modificados, delidos ou incluídos alguns registros ou diálogos. Quando se tratar de uma alteração num diálogo, por exemplo, pode implicar em alterar o mapa, os registros de trabalho do mapa, a

inclusão ou modificação de algum processo. Isso pode exigir o re processamento do esquema, sub-esquema ou uma simples "chamada" à tela correspondente.

A modificação de um processo utiliza recursos do IDD. O procedimento é como aquele descrito quando da correção de erros que ocorrem durante a compilação dos processos. Para qualuer alteração que ocorrer, deve-se gerar novamente o diálogo. Nesse caso, não há necessidade de redefinir os nomes dos processos de resposta ao mapa e pré-mapa. Basta especificar o nome de diálogo e selecionar a opção GENERATE. Se ocorrer algum erro, procede-se às correções como anteriormente descrito.

Na criação de um processo, pode-se copiar um módulo fonte já existente, utilizando recursos do IDD. Neste caso, emite-se o comando DISPLAY MODULE *nome processo* AS SYNTAX. Quando for exibido o módulo na tela, altera-se o nome do processo para o nome do processo que será criado e mantém-se a cláusula ADD MODULE. Com base nessa copia, o projetista pode fazer as alterações necessárias e então gerar o diálogo.

APÊNDICE II

A ferramenta utilizada para efetuar consultas à Base de Dados e Dicionário de Dados é a OLQ, cujos detalhes são apresentados neste apêndice.

Uma visão geral sobre ela foi apresentada no capítulo 3, uma avaliação de sua utilização durante o desenvolvimento do sistema no capítulo 5 e as referências bibliográficas citadas no capítulo 2.

Os detalhes aqui apresentados pretendem oferecer alguns subsídios para o leitor que venha a utilizar a OLQ. São apresentados os diferentes tipos de comandos que estão disponíveis ao usuário OLQ. Acompanham ainda este Apêndice, ilustrações de comandos de recuperação de dados. Se observado o texto, será verificado que a OLQ possui ainda outros recursos além daqueles ilustrados, como por exemplo, de indicação de quebra, de "navegação" entre diferentes tipos de registros, etc...

As informações obtidas através de consultas podem ser formatadas em relatórios que contenham uma ou mais ocorrências de diferentes tipos de registros, registros lógicos ou ainda execução de caminhos e q-arquivos. Caminhos e q-arquivos serão descritos posteriormente no tipo d referente aos grupos de comandos OLQ.

Por questões de segurança, o acesso ao sub-esquema e aos comandos OLQ é permitido apenas de acordo com as definições estabelecidas no dicionário de dados para cada usuário.

Os comandos podem ser agrupados nos seguintes tipos:

- a) Comandos de controle e ajuste;
- b) Comandos de recuperação de dados;
- c) Comandos de processamento de relatórios;
- d) Comandos de definição de caminhos e q-arquivos.

Cada um desses grupos será brevemente descrito a seguir. Maiores detalhes podem ser encontrados em [C83j].

a) COMANDOS DE CONTROLE E AJUSTES

Esses comandos permitem iniciar ou finalizar uma sessão OLQ. Uma sessão desse tipo pode ser entendida como o período de utilização dos recursos disponíveis na OLQ. É iniciada pelo comando SIGNON, onde são especificados o nome e senha do usuário e o nome de sub-esquema. Esses nomes são então "verificados" no dicionário de dados e se forem válidos, tem início uma sessão OLQ. É finalizada pelo comando SIGNOFF.

Um usuário que iniciou a sessão pode obter informações sobre os dados que estão disponíveis a ele, através do comando HELP. Essas informações se referem, por exemplo, aos comandos OLQ, com uma breve descrição de cada um, ao conteúdo do relatório disponível, aos campos e descrições de campos do(s) registro(s) do sub-esquema, etc. Pode estabelecer ou liberar opções de exibição na tela ou de processamento através do comando OPTIONS. Essas opções se referem, por exemplo, à definição dos nomes que aparecerão no cabeçalho do relatório, à consideração ou não dos espaços em branco que constam da definição do registro ou ainda quanto a permitir ou não ao usuário ter acesso a blocos de registros, até que se conclua a busca de todos os registros desejados, com os critérios especificados, sem que haja necessidade de redigitar o comando, etc... O usuário pode ainda modificar a lista de campos de registros que serão recuperados e que irão compor o relatório através do comando FIELDS FOR. É interessante lembrar que essas alterações se aplicarão às recuperações subsequentes desse tipo de registro durante a sessão.

b) COMANDOS DE RECUPERAÇÃO DE DADOS

Esses comandos permitem o acesso a uma ou mais ocorrências de um determinado tipo de registro.

O nome do registro, o número de ocorrências, a lista de nomes de campos que aparecerão no relatório, bem como os critérios de seleção ou os valores chaves que devem ser satisfeitos pelos registros selecionados, podem ser especificados no próprio comando, como pode ser visto na Figura II.01. Nessa figura, bem como nas demais que seguem, será utilizada a seguinte conven


```
GET 3 SEQUENTIAL REG-A42V01-2108 RECORDS WHERE ANO-CADAS-42V01 GT 80
```

```
REG-A42V01-2108 (DBKEY 360159:99)
CHAVE-42V01      :
EMPRESA-42V01   :      222
GRUPO-42V01     :      1
COD-COTI-42V01  :      47531
NOME-COTI-42V01 : 'D Q G PARTICIPA
ENDE-42V01      : 'AV BRIG FARIA      119
CEP-42V01       :      451
CIDA-42V01      : 'CAPITAL
ESTA-42V01      : 'SP'
TEL-42V01       : '
INDIC-FIS-JURI-42V01 : 'J'
CIC-CGC-42V01   :      357540000109
RG-G-42V01      :
NUM-RG-42V01    :      0
ESTA-RG-42V01   : '
NACION-42V01    : '
ESTA-CIVIL-42V01 : '
DT-CADAS-42V01  :
ANO-CADAS-42V01 :      84
... CONTINUED
```

Figura II.1 - Exemplo de Comando de Recuperação de Dados, Especificando Quantidade de Registros, Critério de Seleção.

```
GET FIRST REG-A42V01-2108 WHERE NOME-COTI-42V01 MATCHES ANTAAA
```

```
REG-A42V01-2108 (DBKEY 360451:227)
COD-COTI-42V01   :      1030
NOME-COTI-42V01  : 'ANTOUN MICHEL MACOU
END OF RECORD
```

```
GET REG-A42V01-2108 WHERE ENDE-42V01 CONTAINS SA
```

```
REG-A42V01-2108 (DBKEY 360141:42)
COD-COTI-42V01   :      1360
NOME-COTI-42V01  : 'ABIL SIMO
END OF RECORD
```

Figura II.2 - Exemplo de Especificação de Critério com Cadeia de Caracteres

ção: a parte correspondente àquela que deve ser digitada pelo usuário está no interior do retângulo, sendo que a parte resultante da consulta que é exibida no terminal ou impressa, vem a seguir sem nenhum assinalamento.

A especificação dos critérios a serem satisfeitos pelos registros selecionados, pode ser feita através de expressões de comparação (=, >, <, MATCHES CONTAINS), expressão aritmética ou operadores booleanos (AND, OR, NOT), utilizando a cláusula WHERE, como pode ser visto na Figura II.02. Os valores que aparecem na especificação podem ser números inteiros, números decimais, cadeias de caracteres, etc..., como pode ser visto nas Figuras II.01 e II.02.

A recuperação dos registros pode ser especificada através dos comandos:

- FIND - recupera registro, constroe a lista de chaves da base de dados, mas não gera o relatório. A chave da base de dados é especificada em termos de página e linha em que se encontra o registro.
- GET - recupera registro, constroe a lista de chaves da base de dados e gera o relatório.
- REPEAT - repete o comando imediatamente precedente.

Os comandos de recuperação de dados apresentam variações no formato, dependendo do tipo de acesso pretendido.

A recuperação de um registro baseado na sua localização física na base de dados, pode ser especificada através de FIND/GET PHYSICAL; como ilustrado na Figura II.03.

A recuperação de um registro baseado na chave de armazenamento pode ser especificada através de FIND/GET (nome reg) WHERE CALCKEY = valor da chave, como ilustrado na Figura II.04.

A recuperação de um registro baseado na relação estabelecida entre registros mestres e registros membros, pode ser

GET 80 PHY REG-A42V01-2108 RECORDS WHERE QT-TITU-42V01 GT 02

REG-A42V01-2108 (DBKEY 360512:30)
CHAVE-42V01 :
EMPRESA-42V01 : 222
GRUPO-42V01 : 1
COD-COTI-42V01 : 4480
NOME-COTI-42V01 : 'AIRES A GASPAR'
ENDE-42V01 : 'RUA TABA N 153'
CEP-42V01 : 533
CIDA-42V01 : 'CAPITAL'
ESTA-42V01 : 'SP'
TEL-42V01 : '
INDIC-FIS-JURI-42V01 : 'F'
CIC-CGC-42V01 : 34413800
RG-G-42V01 :
NUM-RG-42V01 : 0
ESTA-RG-42V01 : ' '
NACION-42V01 : ' '
ESTA-CIVIL-42V01 : ' '
DT-CADAS-42V01 :
ANO-CADAS-42V01 : 65
... CONTINUED

Figura II.03 - Recuperação de Registro Baseada na Localização Física

GET REG-A42V01-2108 WHERE CALCKEY=022201047530

REG-A42V01-2108 (DBKEY 360199:181)
CHAVE-42V01 :
EMPRESA-42V01 : 222
GRUPO-42V01 : 1
COD-COTI-42V01 : 47530
NOME-COTI-42V01 : 'D. OG PARTICIPA'
ENDE-42V01 : 'AV. BRIG. FARIA 119'
CEP-42V01 : 451
CIDA-42V01 : 'CAPITAL'
ESTA-42V01 : 'SP'
TEL-42V01 : '
INDIC-FIS-JURI-42V01 : 'J'
CIC-CGC-42V01 : 35754000109
RG-G-42V01 :
NUM-RG-42V01 : 0
ESTA-RG-42V01 : ' '
NACION-42V01 : ' '
ESTA-CIVIL-42V01 : ' '
DT-CADAS-42V01 :
ANO-CADAS-42V01 : 82
... CONTINUED

Figura II.04 - Recuperação de Registro Baseada na Chave de Armazenamento.

especificada através de FIND/GET WITHIN SET como ilustrado na Figura II.05. Esse comando apresenta variações que permitem ter acesso diretamente ao registro mestre, utilizando o formato FIND/GET OWNER WITHIN SET ou o acesso ao registro membro em uma relação ordenada, baseada numa chave de ordenação no formato FIND/GET WITHIN SET (*chave ordenação*).

O acesso a um registro baseado na lista de chave da base de dados, pode ser especificada no formato FIND/GET WITHIN DBKEY LIST. Esse comando e as variações do FIND/GET WITHIN SET acima referidos, são válidos apenas para registros definidos em base de dados.

Pode-se também recuperar registros lógicos especificando o comando no formato GET LOGICAL RECORDS, bem como ter acesso ao registro ao qual se teve acesso mais recentemente, especificando o formato GET MOST RECENT. A Figura II.06 ilustra a utilização de GET MOST RECENT.

c) COMANDOS DE PROCESSAMENTO DE RELATÓRIOS

Esses comandos permitem definir o conteúdo e o formato dos relatórios que serão exibidos na tela ou impressos. Nessa formatação, pode-se referenciar: a posição em termos de coluna no registro recuperado, do campo que fará parte do relatório; o nome do campo obtido de cálculos através do comando COMPUTE; os pontos de quebra através do comando ON BREAK, podendo especificar se na quebra será indicado o total, o maior ou menor valor, a média ou o número de ocorrências, as opções de exibição especificadas no item a, bem como a opção quanto à ordenação ou não, de campos do relatório através dos comandos SORT e UNSORT. Quando o comando COMPUTE for utilizado, pode-se também especificar as características de edição, como por exemplo a inserção do símbolo "\$" ou da vírgula.

O relatório gerado pode ser impresso utilizando o comando PRINT ou exibido na tela, utilizando o comando DISPLAY, com a formatação acima descrita. Além disso, pode-se também especificar o número de cópias desejadas, a impressora utilizada, a página e a linha que se deseja do relatório.

GET FIRST REG-A42V01-2108 WITHIN SET-A42V01

REG-A42V01-2108 (DBKEY 360451:227)
CHAVE-42V01 :
EMPRESA-42V01 : 222
GRUPO-42V01 : 1
COD-COTI-42V01 : 1030
NOME-COTI-42V01 : 'ANTOIN MICHEL MACOU'
ENDE-42V01 : 'AV DUQUE 186 APTO 90'
CEP-42V01 : 214
CIDA-42V01 : 'CAPITAL'
ESTA-42V01 : 'SP'
TEL-42V01 : ''
INDIC-FIS-JURI-42V01 : 'F'
CIC-CGC-42V01 : 501860800
RG-G-42V01 :
NUM-RG-42V01 : 0
ESTA-RG-42V01 : ''
NACION-42V01 : ''
ESTA-CIVIL-42V01 : ''
DT-CADAS-42V01 :
ANO-CADAS-42V01 : 65
... CONTINUED

Figura II.05 - Recuperação de Registros Especificando o SET

GET MOST RECENT REG-A42V02-2109 RECORD

REG-A42V02-2109 (DBKEY 361480:128)
CHAVE-42V02 :
EMPRESA-42V02 : 222
GRUPO-42V02 : 1
COD-COTI-42V02 : 4050
TITU-42V02 :
CONTR-42V02 : 57912
SER-42V02 : 0
FASE-42V02 : 5
QT-COTA-42V02 : 10
DT-CONTR-42V02 :
ANO-CONTR-42V02 : 66
MES-CONTR-42V02 : 9
DIA-CONTR-42V02 : 28
MOT-CONTR-42V02 : ''
COD-COTI-ANT-42V02 : 0
TITU-ANT-42V02 :
CONTR-ANT-42V02 : 0
SER-ANT-42V02 : 0
FASE-ANT-42V02 : 0
... CONTINUED

Figura II.06 - Recuperação do Registro ao qual se teve Acesso mais Recentemente.

Aos comandos de controle, ajuste, recuperação de dados e processamento de relatórios, podem ser especificadas alguns atributos e/ou formatos quando da instalação do sistema e que são válidos até que sejam alterados explicitamente.

Observando os exemplos anteriormente citados, Figura, II.01, II.02, II.03, II.04, II.05, II.06, pode-se notar que quando a recuperação de dados é uma a uma, os nomes dos campos são exibidos em linhas horizontais, enquanto que quando se tratar de recuperação de um conjunto de dados, os nomes de campos são exibidos em colunas, como ilustrado na Figura II.07. Nesse último caso, pode eventualmente resultar em um conjunto com apenas uma ocorrência, sendo que esse número de ocorrências que irão compor o relatório, é exibido num resumo estatístico, como mostra a Figura II.07. Note que nesse exemplo, foram recuperados 3 registros e, como na tela não couberam todas as colunas que compõe o registro para uma única exibição, pode-se solicitar a exibição das demais colunas especificando-as no comando DISPLAY.

Por outro lado, quando a recuperação é uma a uma e também não couberem todas as linhas contendo os nomes dos campos em uma única tela, pode-se "avançar" telas, especificando DISPLAY NEXT como mostra a Figura II.08. Na Figura II.09 está ilustrada a utilização do comando REPEAT.

A Figura II.10 ilustra o caso de referenciar os nomes dos campos que aparecerão no relatório, bem como as diversas buscas que foram efetuadas em blocos de 100 registros - quantidade dessa definida na instalação do sistema - sem que o usuário tenha que intervir, redigitando o comando até obter o total de registros solicitados.

d) COMANDOS DE DEFINIÇÃO DE CAMINHOS E Q-ARQUIVOS

Os comandos de definição de caminhos permitem a recuperação de uma ou mais ocorrências de diferentes tipos de registros. Os registros recuperados compõem um único relatório.

A definição de um caminho tem início com o comando DEFINE PATH. Seguindo-o estão os comandos OLO do item b anteriormente descrito que compõe o caminho. A ordem em que esses coman

GET FIRST 3 REG-A42V01-2108 RECORDS WHERE ANO-CADAS-42V01 GT 80

3 PRIMARY RECORDS SELECTED, 3 RECORDS IN REPORT,
0 DATA ERRORS. PROCESSING ENDED.

DISPLAY

OLA 104009 04 SPECIFIED COLUMNS WILL NOT FIT ON DISPLAY - EXCESS IGNORED
REG-A42V01-2108 PAGE 1.1

EMPRESA-42V01 GRUPO-42V01 COD-COTI-42V01 NOME-COTI-42V01
222 1 2045 ADDA JULIETA BOSSA
222 1 3070 ADRIANA CRISTINA DA S OLI
222 1 3810 AGNALDO DE SO
END OF REPORT - PAGE 1

DISPLAY COLS = 5,6,7,8

REG-A42V01-2108 PAGE 1.1
LINE 1
ENDE-42V01 CEP-42V01 CIDA-42V01 ESTA-42V01
PRACA TRISTAO 18 189 CAPITAL SP
AV CIDADE JARDIM 42 APTO 6 453 CAPITAL SP
P COMENDADOR ESTEVES 88 37200 LAVRAS MG
END OF REPORT - PAGE 1

DISPLAY COLS = 9,10,11,12

REG-A42V01-2108 PAGE 1.1
LINE 1
TEL-42V01 INDIC-FIS-JURI-42V01 CIC-CGC-42V01 NUM-RG-42V01
F 0 0
F 701535819 0
F 453495615 0
END OF REPORT - PAGE 1

DISPLAY COLS = 13,14,15,16

REG-A42V01-2108 PAGE 1.1
LINE 1
ESTA-RG-42V01 NACION-42V01 ESTA-CIVIL-42V01 ANO-CADAS-42V01
83
85
85
END OF REPORT - PAGE 1

DISPLAY COLS = 17,18,19,20

REG-A42V01-2108 PAGE 1.1
LINE 1
MES-CADAS-42V01 DIA-CADAS-42V01 GRUPO-COTI-42V01 QT-TITU-42V01
2 28 0 1
5 31 0 3
2 28 0 2
END OF REPORT - PAGE 1

Figura II.07 - Recuperação de um Conjunto de Dados, Especificando Colunas a serem exibidas.

GET FIRST SEQUENTIAL REG-A42V01-2108

REG-A42V01-2108 (DBKEY 360451:227)
 CHAVE-42V01 :
 EMPRESA-42V01 : 222
 GRUPO-42V01 : 1
 COD-COTI-42V01 : 1030
 HOME-COTI-42V01 : 'ANTOUN MICHEL MACOU'
 ENDE-42V01 : 'AV DUQUE' 186 APTO 90
 CEP-42V01 : 214
 CIDA-42V01 : 'CAPITAL'
 ESTA-42V01 : 'SP'
 TEL-42V01 :
 INDIC-FIS-JURI-42V01 : 'F'
 CIC-CGC-42V01 : 501860800
 RG-G-42V01 :
 NUM-RG-42V01 : 0
 ESTA-RG-42V01 :
 NACION-42V01 :
 ESTA-CIVIL-42V01 :
 DT-CADAS-42V01 :
 ANO-CADAS-42V01 : 65
 ... CONTINUED

DISPLAY NEXT

REG-A42V01-2108 CONTINUED
 MES-CADAS-42V01 : 3
 DIA-CADAS-42V01 : 23
 GRUPO-COTI-42V01 : 0
 QT-TITU-42V01 : 1
 QT-COTA-42V01 : 1
 BLOQ-42V01 : 0
 DT-BLOQ-42V01 :
 ANO-BLOQ-42V01 : 0
 MES-BLOQ-42V01 : 0
 DIA-BLOQ-42V01 : 0
 TIPO-PAGTO-42V01 : 4
 BCO-AGE-42V01 :
 BCO-42V01 : 0
 AGE-42V01 : 0
 CONTA-CORRE-42V01 :
 NUM-CONTA-CORRE-42V01 : 0
 DIG-CONTA-CORRE-42V01 : '0'
 DT-ULT-PAGTO-42V01 :
 ANO-ULT-PAGTO-42V01 : 83
 ... CONTINUED

DISPLAY NEXT

REG-A42V01-2108 CONTINUED
 MES-ULT-PAGTO-42V01 : 4
 DIA-ULT-PAGTO-42V01 : 18
 SALDO-A-PAGAR-42V01 : 143801.00
 RATEIO-MES-42V01 : 13611.00
 IR-MES-42V01 : 0.00
 VR-PAGO-MES-42V01 : 0.00
 DADOS-TRIM-G-42V01 :
 DADOS-TRIM-42V01(1) :
 VR-CRED-TRIM-42V01(1) : 38780.00
 VR-PAGO-TRIM-42V01(1) : 0.00
 IR-RET-TRIM-42V01(1) : 0.00
 DADOS-TRIM-42V01(2) :
 VR-CRED-TRIM-42V01(2) : 22440.00
 VR-PAGO-TRIM-42V01(2) : 0.00
 IR-RET-TRIM-42V01(2) : 0.00
 DADOS-TRIM-42V01(3) :
 VR-CRED-TRIM-42V01(3) : 13611.00
 VR-PAGO-TRIM-42V01(3) : 0.00
 IR-RET-TRIM-42V01(3) : 0.00
 ... CONTINUED

DISPLAY NEXT

REG-A42V01-2108 CONTINUED
 DADOS-TRIM-42V01(4) :
 VR-CRED-TRIM-42V01(4) : 0.00
 VR-PAGO-TRIM-42V01(4) : 0.00
 IR-RET-TRIM-42V01(4) : 0.00
 DT-CONTAC-1-42V01 :
 ANO-CONTAC-1-42V01 : 0
 MES-CONTAC-1-42V01 : 0
 DIA-CONTAC-1-42V01 : 0
 DT-CONTAC-2-42V01 :
 ANO-CONTAC-2-42V01 : 0
 MES-CONTAC-2-42V01 : 0
 DIA-CONTAC-2-42V01 : 0
 VR-ORTN-42V01 : 0.00
 OBS-42V01 :

END OF RECORD

Figura II.08 - "Avanço" de Telas.

REPEAT

REG-A42V01-2108 (DBKEY 360451:227)
CHAVE-42V01 :
EMPRESA-42V01 : 222
GRUPO-42V01 : 1
COD-COTI-42V01 : 1030
NOME-COTI-42V01 : 'ANTOIN MICHEL MACO'
ENDE-42V01 : 'AV DUQUE' 186 APTO 90
CEP-42V01 : 214
CIDA-42V01 : 'CAPITAL'
ESTA-42V01 : 'SP'
TEL-42V01 : '
INDIC-FIS-JURI-42V01 : 'F'
CIC-CGC-42V01 : 501860800
RG-G-42V01 :
NUM-RG-42V01 : 0
ESTA-RG-42V01 : ' '
NACION-42V01 : '
ESTA-CIVIL-42V01 : ' '
DI-CADAS-42V01 :
ANO-CADAS-42V01 : 65
... CONTINUED

REPEAT WITH NEXT

REG-A42V01-2108 (DBKEY 360268:78)
CHAVE-42V01 :
EMPRESA-42V01 : 222
GRUPO-42V01 : 1
COD-COTI-42V01 : 1050
NOME-COTI-42V01 : 'ABAD O M MIZ'
ENDE-42V01 : 'R OSCAR' 8
CEP-42V01 : 105
CIDA-42V01 : 'CAPITAL'
ESTA-42V01 : 'SP'
TEL-42V01 : '
INDIC-FIS-JURI-42V01 : 'F'
CIC-CGC-42V01 : 1353800
RG-G-42V01 :
NUM-RG-42V01 : 0
ESTA-RG-42V01 : ' '
NACION-42V01 : '
ESTA-CIVIL-42V01 : ' '
DT-CADAS-42V01 :
ANO-CADAS-42V01 : 66
... CONTINUED

Figura II.09 - Comando de Repetição

GET FIRST 5 REG-A42V01-2108 (COD-COTI-42V01,NOME-COTI-42V01) WHERE QT-TITU-42V01
GT 20 AND ANO-CADAS-42V01 GT 80

100 PRIMARY RECORDS READ 0 RECORDS IN REPORT CONTINUE (YES/NO)?	0 PRIMARY RECORDS SELECTED 0 DATA ERRORS
YES	
200 PRIMARY RECORDS READ 1 RECORDS IN REPORT CONTINUE (YES/NO)?	1 PRIMARY RECORDS SELECTED 0 DATA ERRORS
YES	
300 PRIMARY RECORDS READ 1 RECORDS IN REPORT CONTINUE (YES/NO)?	1 PRIMARY RECORDS SELECTED 0 DATA ERRORS
YES	
400 PRIMARY RECORDS READ 2 RECORDS IN REPORT CONTINUE (YES/NO)?	2 PRIMARY RECORDS SELECTED 0 DATA ERRORS
YES	
500 PRIMARY RECORDS READ 2 RECORDS IN REPORT CONTINUE (YES/NO)?	2 PRIMARY RECORDS SELECTED 0 DATA ERRORS
YES	
600 PRIMARY RECORDS READ 2 RECORDS IN REPORT CONTINUE (YES/NO)?	2 PRIMARY RECORDS SELECTED 0 DATA ERRORS
YES	
700 PRIMARY RECORDS READ 2 RECORDS IN REPORT CONTINUE (YES/NO)?	2 PRIMARY RECORDS SELECTED 0 DATA ERRORS
YES	
800 PRIMARY RECORDS READ 2 RECORDS IN REPORT CONTINUE (YES/NO)?	2 PRIMARY RECORDS SELECTED 0 DATA ERRORS
YES	
900 PRIMARY RECORDS READ 3 RECORDS IN REPORT CONTINUE (YES/NO)?	3 PRIMARY RECORDS SELECTED 0 DATA ERRORS
YES	
1000 PRIMARY RECORDS READ 4 RECORDS IN REPORT CONTINUE (YES/NO)?	4 PRIMARY RECORDS SELECTED 0 DATA ERRORS

DISPLAY

```
REG-A42V01-2108 PAGE 1.1  
LINE 1  
COD-COTI-42V01 NOME-COTI-42V01  
7900 ALEXANDRE C C W V GERIC  
13655 ANNA XAVIER  
35260 CARLOS FRANCISCO RIBEIRO  
37950 CEJOTA PARTICIPA  
40060 CHRISTIAN GEORG  
END OF REPORT - PAGE 1
```

Figura II.10 - Especificação de Campo de Relatórios, com Pesquisa em Blocos de Tamanho Pré-definidos.

dos são executados, determina a rota de busca em um sub-esquema - denota o movimento que pode ser estabelecido entre registros mestres e registros membros e vice-versa, durante a recuperação dos dados. O comando END PATH finaliza a definição de caminhos e o comando EXECUTE PATH executa todos os comandos especificados na definição e ao final constroe o relatório. Um caminho pode ser extendido ou modificado, sendo que nesse caso, deve-se emitir o comando EXECUTE PATH para gerar um novo relatório, baseado nas alterações efetuadas.

Um q-arquivo consiste de um conjunto de comandos OLQ e parâmetros, se necessário. O q-arquivo deve ser identificado por um nome e estar definido no dicionário de dados. A definição de um q-arquivo é semelhante à definição de um caminho, com as seguintes vantagens: permite especificar parâmetros, o que dá maior flexibilidade na geração do relatório e permite ser executado tantas vezes quantos for requisitado, pois está definido no dicionário de dados.