INSTITUTO DE CIÊNCIAS MATEMÁTICAS DE SÃO CARLOS

ICMSC-USP
PÓS-GRADUAÇÃO

UMA EXPERIÊNCIA DE APLICAÇÃO DE
MICROCOMPUTADORES A MICRO EMPRESAS

SANDRA C. PINTO FERRAZ FABBRI

UNIVERSIDADE DE SÃO PAULO

SÃO CARLOS - SÃO PAULO
BRASIL

UMA EXPERIÊNCIA DE APLICAÇÃO DE
MICROCOMPUTADORES A MICRO EMPRESAS

SANDRA C. PINTO FERRAZ FABBRI

Orientador:

Prof. Dr. Fernão Stella de Rodrigues Germano

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

SÃO CARLOS
1985

Dedico este trabalho

a meus pais e minha avô, por
não medirem esforços para mi
nha formação
ao Glauco, pela compreensão e
apoio de todos esses anos.

UMA EXPERIÊNCIA DE APLICAÇÃO DE MICROCOMPUTADORES A MICRO EMPRESAS

RESUMO

Um escritório de corretagem de seguros foi objeto de uma experiência destinada a apoiar por computador suas atividades. A Análise Estruturada foi aplicada na especificação do sistema manual existente. Foram desenhados diagramas de fluxo de dados conjugados com um dicionário de dados manual. As necessidades do usuário foram identificadas como relativas principalmente ao controle dos vencimentos de pagamentos e das datas de expiração de seguros bem como a recuperação de informações sobre os clientes e capitais segurados. As técnicas de projeto estruturado foram aplicadas em conjunto com programação estruturada para desenvolver uma implementação em dBaseII de um sistema para satisfazer essas necessidades. Um microcomputador tipo Apple com duas unidades de disco flexível e uma impressora foi usado na experiência. Os resultados indicam que a aplicação definitivamente não foi efetiva em termos de custo, principalmente devido ao tempo e recursos financeiros consumidos. Maneiras alternativas para resolver problemas de processamento de dados de micro empresas com microcomputadores são examinadas no trabalho. Quarenta e duas referências bibliográficas são discutidas nesse contexto. A experiência leva à conclusão de que para que microcomputadores possam ser aplicados efetivamente em termos de custo a micro empresas é necessário que esforços de pesquisas sejam concentrados em tornar mais expedito o esforço de análise.

AN EXPERIMENT IN THE APPLICATION OF MICROCOMPUTERS TO SMALL FIRMS

ABSTRACT

A small insurance brokerage firm has been the object of an experiment aiming to computer aid its activities. Structured analysis has been applied in the specification of the existing manual system. Data flow diagrams have been drafted together with a manual data dictionary. User requirements have been identified as referring mainly to payments due and insurance expiration date control as well as customer information and policy amount retrieval. Structured design techniques have been applied together with structured programming to develop an implementation in dBaseII of a system to satisfy these requirements. An Apple like microcomputer with two floppy disk drives and a printer has been used in the experiment. Results indicate that this application has definitely not been cost effective mainly due to analysis time and cost consumption. Alternative ways for solving small firms data processing problems with microcomputers are examined in the paper. Forty-two bibliographic references are discussed in this context. The experiment leads to the conclusion that research efforts should concentrate in expediting the analysis effort if microcomputers are to be cost effectively applied to small firms.

AGRADECIMENTOS

Ao Prof. Dr. *Fernão Stella de Rodrigues Germano* pela orientação e dedicação a mim dispensada, o que possibilitou vencer esta etapa de meu desenvolvimento científico.

Ao Prof. *Arne Solvberg*, pelos contatos quando de sua vinda ao ICMSC-USP e, posteriormente, pela atenção com que analisou a documentação da análise estruturada do sistema abordado neste trabalho, que lhe foi enviada.

Aos professores *Odelar Leite Linhares*, *Maximilian E. Hehl* e demais funcionários e amigos do ICMSC-USP pela atenção durante meu tempo de permanência naquela instituição.

A Profa. *Rosângela Ap. Delloso Penteado* por sua constante colaboração e principalmente por sua amizade.

A todos os colegas do Departamento de Computação da Universidade Federal de São Carlos, em especial aos professores *Tereza G. Kirner* e *Marco A. Zabotto*, do grupo de Sistemas de Informação, pelo constante apoio a mim dado.

Ao Sr. *Benvindo Augusto Dias* pela pronta colaboração em colocar ao meu dispor sua vasta experiência e conhecimentos no ramo de corretagem de seguros.

Ao Sr. *Erik Wilner*, pela oportunidade a mim concedida de trabalhar com um sistema de grande porte e nele aplicar meus conhecimentos das técnicas de projeto estruturado e de sistemas de apoio por computador à documentação de projetos.

Ao Departamento de Transportes da EESC pela autorização a mim concedida para utilização de seu microcomputador.

A *Maria Lúcia Clapis Facundo* pela dedicação e capricho com que datilografou este trabalho e principalmente pela amizade que surgiu de nosso conhecimento.

A todas as amigadas antigas que se firmaram e às no

vas que se formaram durante o período de pós-graduação. Que elas permaneçam apesar da distância que nos separa.

A CAPES pela bolsa de estudos concedida.

I N D I C E

Capítulo 1 - INTRODUÇÃO

1.1 - Considerações Gerais	01
1.2 - Assunto da Dissertação	03
1.3 - Apresentação	05

Capítulo 2 - ANÁLISE BIBLIOGRÁFICA

2.1 - Considerações Gerais	07
2.2 - Instrumental de Análise Estruturada	09
2.3 - Instrumental para desenvolvimento expedito de Sistemas	19
2.4 - Computadores e Micro Empresas	25
2.5 - Conclusões	33

Capítulo 3 - ANÁLISE ESTRUTURADA DO CASO PRÁTICO ABORDADO

3.1 - Considerações Gerais	35
3.2 - Comunicação com o usuário	37
3.3 - Elaboração do Diagrama de Fluxo de Dados...	39
3.4 - Elaboração do Dicionário de Dados	53
3.5 - Conclusões	63

Capítulo 4 - SOLUÇÃO DO CASO PRÁTICO ABORDADO

4.1 - Considerações Gerais	65
4.2 - Esquema da Base de Dados	67
4.3 - Estrutura Modular dos Programas	73
4.4 - Detalhamento dos Módulos	79
4.5 - Alguns Exemplos do Funcionamento do Sis tema	81
4.6 - Conclusões	93

Capítulo 5 - AVALIAÇÃO DO DESENVOLVIMENTO EFETUADO

5.1 - Considerações Gerais	95
5.2 - Avaliação da Análise Efetuada	97
5.3 - Avaliação da Implementação Efetuada	101
5.4 - Avaliação da Satisfação dos Requisitos do Usuário	105
5.5 - Conclusões	109

Capítulo 6 - CONCLUSÕES FINAIS E SUGESTÕES PARA NOVAS PESQUISAS

6.1 - Considerações Gerais	111
6.2 - Conclusões Finais	113
6.3 - Sugestões para Novas Pesquisas	115
6.4 - Contribuição deste Trabalho	117

BIBLIOGRAFIA	119
--------------------	-----

APÊNDICE	125
----------------	-----

C A P Í T U L O 1

INTRODUÇÃO

1.1 - CONSIDERAÇÕES GERAIS

No final do período de graduação, a autora estagiou durante um ano, como analista, numa firma de equipamento pesado. Durante esse estágio, desenvolveu um sistema para controle de aferição de instrumentos de medição, solicitado pelo departamento de controle de estoque. Desde essa época, a autora já possuía interesse no desenvolvimento de sistemas de informação e encontrou, nesse estágio, uma forma de satisfazê-lo. No entanto, durante esse trabalho, sentiu uma deficiência muito grande de uma ferramenta que apoiasse o desenvolvimento e documentação do sistema que estava desenvolvendo. A partir de então, surgiu seu interesse em direcionar sua pesquisa à área de apoio ao desenvolvimento e documentação de sistemas de informação.

Durante seus estudos de pós-graduação teve oportunidade de conhecer várias metodologias utilizadas para esta finalidade, como por exemplo, HIPO [Ib74], SADT [Sf76] e ISAC [Lu81]. Começou a se familiarizar efetivamente com tais metodologias quando participou, junto com uma equipe, do atendimento da solicitação de um grande banco internacional, o qual desejava tomar conhecimento de técnicas disponíveis para a documentação de sistemas. Durante esse serviço, a autora se deparou com o problema de um sistema já desenvolvido, mas não documentado, sendo o objetivo do grupo justamente propor uma técnica de documentação. Foi então utilizada a técnica de projeto estruturado proposta por Yourdon e Constantine, [Yo75], para extrair a estrutura do sistema e particioná-lo em módulos funcionais e, juntamente, foi utilizado o sistema PSL/PSA [Te77] para documentar todo esse processo e fornecer relatórios que apoiassem sua evolução.

Nesse mesmo período, teve contato com o caso prático abordado nesta dissertação, correspondente a um Escritório

de Corretagem de Seguros. O proprietário desse Escritório desejava comprar um equipamento para apoiar por computador algumas atividades desempenhadas pelo seu pessoal. Tal caso gerou muito interesse, por parte da autora, que agora tinha oportunidade de elaborar a análise de um sistema apoiada por ferramentas apropriadas, sendo que dentre elas, foi escolhida a metodologia de análise estruturada recomendada por Gane e Sarson [Ga77].

Além disso, esse caso era particularmente interessante, pois como será justificado no item seguinte, o Escritório em questão pode ser considerado uma micro empresa e, como tal, é um alvo muito fácil e frequentemente atingido por pessoas físicas e jurídicas cuja intenção primordial é vender seus produtos de software. Dessa forma, a micro empresa se torna insatisfeita, pois seus requisitos normalmente não são alcançados.

Portanto, acredita-se que a micro empresa, tanto quanto, ou mais que qualquer outra empresa, merece uma atenção muito especial, a qual pode ser traduzida numa dedicação particularmente dada à fase de desenvolvimento e documentação da análise do sistema, pois dessa forma pode-se atingir, com maior precisão, as reais necessidades da mesma.

Nesta dissertação, a abordagem de construção de protótipos para o desenvolvimento de sistemas não está sendo considerada, não porque não seja aplicável ao caso, mas porque é assunto de outra dissertação de mestrado desenvolvida no mesmo grupo de pesquisa.

1.2 - ASSUNTO DA DISSERTAÇÃO

A experiência relatada nesta dissertação, refere-se ao desenvolvimento e documentação da análise e à implementação de um sistema de informação para um Escritório de Corretagem de Seguros.

O Escritório de Corretagem de Seguros em questão possuía cerca de doze funcionários e mais o proprietário, o qual participava intensamente de suas atividades. Dentre os doze funcionários, um deles era um gerente geral que coordenava os demais e passava a eles todas as diretrizes para que desempenhassem suas funções. Quanto aos outros funcionários, eles consistiam de uma secretária, alguns vendedores e cobradores, alguns outros encarregados da realização de seguros no próprio Escritório e ainda outros, encarregados da preparação de documentos para a realização da contabilidade. A Corretora realizava seguros, agrupando-os em dois conjuntos distintos: um deles referente a seguros de Vida em Grupo e Acidentes Pessoais (VG/AP) e o outro, referente aos seguros de todos os outros tipos, isto é, seguros de automóveis, incêndio, roubo, etc.. Na época em que o trabalho foi realizado, a Corretora possuía por volta de cinco mil clientes, dentre os quais muitos estavam incluídos em seguros coletivos de Vida em Grupo e Acidentes Pessoais, que eram realizados normalmente para firmas, enquanto que os outros tipos eram normalmente realizados para clientes particulares.

No entanto, a cada ano, o Escritório de Corretagem de Seguros estava ultrapassando as expectativas de seu proprietário que, preocupado com este fato, sentia que a qualquer momento, seria indispensável um controle mais rígido e seguro, para que esse crescimento, além de suas expectativas, continuasse acontecendo. Daí o seu desejo de adquirir um equipamento para apoiar por computador principalmente aquelas atividades que precisavam ser rigorosamente controladas.

Possuindo as características descritas acima, pode-se considerar o Escritório em questão uma micro empresa. Na verdade, uma micro empresa possui os mesmos problemas de uma grande empresa, apenas em proporções menores, merecendo portanto, a

mesma dedicação dispensada a essas últimas. Portanto, assim como as grandes empresas, as micro empresas também possuem suas necessidades de processamento; o que elas podem não possuir como as outras, são as condições financeiras para apoiar desenvolvimento de sistemas de informação, de maneira adequada. Dessa forma, acredita-se que se trata claramente de um problema de aplicação de microcomputadores a micro empresas.

1.3 - APRESENTAÇÃO

Esta dissertação está organizada da seguinte maneira:

O Capítulo 2 refere-se à análise bibliográfica realizada, na qual estão fundamentados todos os outros itens desta dissertação. Essa bibliografia aborda o instrumental de análise estruturada disponível, o instrumental para desenvolvimento expedito de sistemas e a evolução do uso de computadores nas empresas.

O Capítulo 3 apresenta a análise estruturada do caso prático abordado, referente ao Escritório de Corretagem de Seguros.

O Capítulo 4 refere-se à implementação do caso prático, onde são apresentados o esquema da base de dados, a estrutura modular dos programas e alguns exemplos de funcionamento dos mesmos.

No Apêndice, são apresentados os detalhamentos dos módulos e as listagens dos respectivos programas.

O Capítulo 5 apresenta uma avaliação de todo o desenvolvimento efetuado: da análise, da implementação e da satisfação dos requisitos do usuário, frente à solução apresentada.

Finalmente, no Capítulo 6, faz-se algumas conclusões, sugestões para novas pesquisas e salienta-se as contribuições oferecidas por este trabalho.

C A P Í T U L O 2

ANÁLISE BIBLIOGRÁFICA

2.1 - CONSIDERAÇÕES GERAIS

Este capítulo tem por objetivo fornecer um apanhado geral da bibliografia sobre três aspectos fundamentais para esta dissertação. Assim, esses aspectos estão abordados nos itens a seguir, da seguinte maneira:

No item 2.2 faz-se uma revisão do instrumental de Análise Estruturada disponível, comentando-se várias publicações sobre o assunto, referentes aos principais autores que se dedicam a essa pesquisa e que propuseram metodologias de documentação, sistemas de apoio por computador à descrição de sistemas, metodologias específicas para descrever atividades de automação de escritórios e softwares de apoio por computador a todas as fases relacionadas ao desenvolvimento de um Sistema de Informação.

No item 2.3 fala-se sobre alguns softwares para desenvolvimento expedito de sistemas, correspondentes ao tipo de processamento dominante no caso prático abordado, que é o gerenciamento de dados.

No item 2.4 faz-se uma revisão da evolução do uso de computadores nas pequenas empresas, procurando-se posicionar principalmente o tipo de solução disponível ou proposta, de acordo com as exigências de processamento requeridas.

Finalmente, no item 2.5, faz-se uma síntese desses três aspectos abordados nos itens anteriores.

2.2 - INSTRUMENTAL DE ANÁLISE ESTRUTURADA

Em 1974, Stevens, Myers e Constantine, em [St74], propõem técnicas e considerações para reduzir a complexidade dos programas, dividindo-os em módulos funcionais. Tais técnicas, conhecidas como *Projeto Estruturado*, permitem transformar um sistema complexo em módulos simples, independentes e reutilizáveis e, além disso, são compatíveis com a técnica HIPO de documentação. São definidos os conceitos de acoplamento e coesão, que são a chave para se definir os módulos de forma a se obter os objetivos acima citados, bem como a simbologia básica para se construir os diagramas de estrutura. Apresentam os passos principais para a elaboração de um projeto estruturado e algumas das vantagens obtidas ao usá-lo.

Em 1975, Yourdon e Constantine publicam um livro, [Yo75], sobre projeto estruturado, que é definido por eles como "a arte de projetar as componentes de um sistema e o relacionamento entre elas, da melhor maneira possível". Enfatizam que não é suficiente encontrar um projeto para um sistema, mas sim o *melhor* e, pensando nisso, procuram desenvolver uma metodologia que ajude a elaborar bons projetos e a descartar os projetos ruins do modo mais simples possível. Propõem para isso um particionamento do sistema através da divisão do problema em subproblemas pequenos, de modo que cada um destes corresponda a um pedaço do sistema. Nesse contexto, consideram útil o conceito de "*caixa preta*", pois à medida em que as funções tornam-se necessárias durante o projeto, elas são definidas como tal, utilizadas sem preocupação com a sua estrutura e detalhadas posteriormente, no momento oportuno. A elaboração do projeto estruturado está toda baseada em diagramas de fluxo dos dados através do sistema. Com base neles, são definidos os graus de acoplamento e coesão entre os módulos, conceitos estes amplamente discutidos. Definem detalhadamente a simbologia utilizada na elaboração dos diagramas de estrutura, exemplificando situações em várias linguagens de programação. Salientam sua preocupação em obter projetos que tenham o menor custo nas fases de implementação, teste e manutenção e, além disso, associam a um bom projeto três qualidades muito importan

tes: *generalidade* que está associada a cada módulo separadamente e corresponde a ser amplamente usado, a resolver vários casos de uma classe de problemas, a ser prontamente adaptável a muitas variações e a funcionar em meios diferentes; *flexibilidade* que é uma medida do grau de um sistema, quanto a ele poder ser usado de vários modos, sem sofrer alterações e *confiabilidade* que corresponde ao comportamento do sistema quando surgem falhas nos dados que são processados. Além disso, falam também na facilidade obtida na fase de teste, pois como cada módulo corresponde a uma função distinta, eles podem ser testados separadamente e depois, inseridos no contexto geral.

Ainda em 1975, Yourdon, em [Yo75a], publica um livro sobre programação estruturada e projeto de programas. Utiliza como base as linguagens FORTRAN e COBOL e discute vários itens de programação como: flexibilidade, eficiência, minimização dos custos de desenvolvimento, etc.. Objetivando essas características, propõe um desenvolvimento de programas "top-down" durante todas as fases, desde a definição do problema, codificação e teste. Fala sobre a programação modular e técnicas para elaborá-la, como fluxogramas e tabelas de decisão e discute a programação estruturada fornecendo todos os conceitos e exemplificando várias situações.

Em 1976, Yourdon, em [Yo76], faz uma revisão de algumas técnicas de estruturação existentes, visando o gerente de processamento de dados, fornecendo-lhe as noções básicas para que ele tenha condições de introduzir e implementar essas técnicas dentro de sua organização. Geralmente, a resistência é bastante grande, quanto a mudanças no procedimento das pessoas. Com a intenção de criar uma base para que o gerente possa "negociar" essas técnicas dentro da empresa, o autor fala sobre *análise estruturada*, como uma coleção de regras e técnicas de comunicação gráfica que permitem ao analista descrever o sistema não de uma forma convencional, mas sim através de um novo tipo de especificação que pode ser lido e compreendido pelos usuários, onde são colocados os conceitos de diagrama de fluxo de dados, dicionário de dados e inglês estruturado; *implementação e projeto "top-down"*, onde coloca a estratégia de quebrar o sistema em funções gerais, es

tas em funções menores, e assim por diante, até que a implementação seja expressa em termos de instruções de programas; *projeto estruturado*, revisando o conjunto de regras e técnicas, tratadas com detalhes em sua publicação anterior [Yo75], que auxiliam a distinguir projetos bons e ruins, a um nível modular e *programação estruturada*, propondo que a lógica dos programas seja expressa através de combinações das estruturas básicas: sequência, iteração e seleção. Além disso, o autor fornece sugestões sobre qual técnica implementar primeiro, como escolher um projeto piloto, formação de equipes de revisão ("Walkthroughs"), etc..

Em 1977, Yourdon, em [Yo77], discute amplamente o conceito de "*walkthroughs*", mencionado em sua publicação anterior, [Yo76]. Define esse conceito como "um grupo de revisão de qualquer produto", isto é, um esquema de trabalho onde as pessoas, através de equipes, desenvolvem e avaliam todo tipo de atividade realizada, e assumem, como um todo, a responsabilidade de todos os produtos. Dessa forma, ele fala em "*walkthroughs*" de análise, de projeto, de programação, etc.; enfim, numa nova forma de produzir cada uma das etapas do desenvolvimento de um sistema, garantindo com isso um grande aumento de qualidade.

Ainda em 1977, Gane e Sarson, em [Ga77], propõem uma metodologia de *análise estruturada* baseada nos conceitos de diagramas de fluxo de dados já utilizados por Yourdon e Constantine em [Yo75]. Essa metodologia consiste em retratar logicamente o sistema, através de um processo de refinamento sucessivo de construção de diagramas de fluxo de dados, partindo de um diagrama geral, no qual são identificadas as funções principais do sistema, e elaborando diagramas detalhados que correspondem a explosão das funções do nível de detalhamento anterior. A preocupação, durante esse processo, é retratar da melhor maneira possível, o que o sistema faz e não como faz, conseguindo com isso, isolar completamente a fase de análise da fase de projeto. Propõem também a elaboração de um dicionário de dados para apoiar a descrição gráfica do sistema através dos diagramas de fluxo de dados. O dicionário de dados descreve todos os objetos utilizados nos diagramas: *entidades externas, processos, fluxos de dados e depósitos de dados*, e também, *estruturas de dados e ele*

mentos de dados que não aparecem explicitamente nos mesmos. São apresentadas várias técnicas de descrição das informações no dicionário, principalmente quanto a processos e depósitos de dados. Em seguida, os autores mostram como derivar um projeto estruturado a partir da metodologia de análise estruturada sendo que tal projeto corresponde à técnica desenvolvida em [Yo75]. Finalmente, de maneira bem rápida, falam como introduzir a análise estruturada na organização.

Nesse mesmo ano de 1977, Teichroew e Hershey, [Te77], publicam um artigo sobre o sistema PSL/PSA, que é uma técnica de apoio por computador à análise e documentação estruturadas de um sistema de informação. Essa técnica vinha sendo desenvolvida já há algum tempo. Durante o ciclo de vida de um sistema, ele existe em diferentes formas e sempre está recebendo especificações de uma atividade anterior e produzindo dados para uma atividade posterior. Como os indivíduos envolvidos com o sistema podem mudar, é necessária uma documentação para registrar as descrições do mesmo, sendo que a documentação final é muito importante por retratar o sistema na sua última forma, pois é esta que sofrerá manutenção. Dizem os autores que essa documentação intermediária é muito difícil, pois normalmente ela é falada ou colocada em documentos que acabam se perdendo e o resultado disso é que nunca se tem uma visão exata do sistema ou de como ele evoluiu. O sistema PSL/PSA ajuda justamente a melhorar o processo de desenvolvimento. O PSL - *Problem Statement Language*, é uma linguagem para descrever sistemas. Através dela podem ser expressos, de uma forma sintaticamente analisável, todos os objetos do sistema, tais como: fluxos de entrada /saída, estrutura do sistema, estrutura dos dados, derivação dos dados, volume e tamanho do sistema, dinâmica do sistema, propriedades do sistema e gerência de projeto. O PSL contém vários tipos de objetos e relacionamentos que permitem que todos esses aspectos sejam descritos. A capacidade de registrar tais descrições em uma base de dados, modificá-las quando necessário e, de acordo com a demanda, executar uma análise e produzir relatórios, é associada a um analisador, o PSA - *Problem Statement Analyzer*, o

qual é controlado por uma linguagem de comando. Através desse analisador, são extraídos vários tipos de relatórios sobre o sistema, inclusive relatórios gráficos, segundo vários tipos de cruzamento e enfoque, de forma a obter uma visão ampla e variada de todas as informações contidas na base de dados. Os autores colocam que o uso do sistema PSL/PSA não depende de qualquer estrutura particular para o processo de desenvolvimento do sistema, ou qualquer padronização no formato ou conteúdo da documentação.

Em 1978, Tom DeMarco, em [De78], reúne as idéias lançadas anteriormente por Constantine, quanto à utilização da análise e projeto estruturados no desenvolvimento de sistemas de informação. Sua proposta pode ser comparada àquela feita por Gane. [Ga77], pois as etapas a serem cumpridas são praticamente as mesmas, com a diferença de que DeMarco utiliza a mesma simbologia empregada por Constantine na elaboração dos diagramas de fluxo de dados. Ele também propõe um processo de elaboração desses diagramas, partindo de um nível geral e passando para níveis mais detalhados. Aborda a utilização de um dicionário de dados e também a derivação do projeto estruturado, segundo Yourdon, a partir da análise estruturada realizada.

Um apanhado geral de várias metodologias de análise estruturada foi realizado em 1978, por Weinberg, em [We78]. Ele diz que a análise estruturada fornece critérios para solucionar os problemas da análise de sistemas e técnicas gráficas para minimizar os problemas de comunicação, pois acredita ser esse o problema mais crítico na fase de análise, devido ao fato de que, frequentemente, nem usuários nem analistas conhecem as reais necessidades dos primeiros. Com a complexidade de tais necessidades, diz ter sido inevitável o surgimento de abordagens disciplinadas para tratá-las. Assim, ele fala sobre a primeira técnica amplamente usada que foi a programação estruturada, sobre tabelas de decisão, diagramas de fluxo de dados e algumas variações como os ativiogramas, dicionário de dados, diagramas de estrutura de dados, árvores de decisão, diagramas de estrutura modular, estruturas de dados em árvore verticais usadas por Jackson, estruturas de dados em árvore horizontais usadas por Warnier, diagramas hierárquico e detalhado utilizados na técnica Hipo, e outros. Além

disso, ele aborda vários outros aspectos fornecendo estratégias para desenvolver projetos, estratégias para implementar sistemas, como um plano de implementação "top-down", aspectos sobre a definição do problema, seus objetivos e estimativas de custo/benefício, algumas diretrizes para seleção de pacotes de software, etc..

Em 1981, Stevens, em [St81], faz uma revisão dos conceitos de projeto estruturado, fornecendo a simbologia básica utilizada, discutindo as noções de coesão e acoplamento dos módulos e fornecendo algumas técnicas para utilizar o projeto estruturado de modo mais rápido e fácil. Diz que as principais regras a serem seguidas são: *independência*, que reduz a complexidade e é conseguida através de uma modularização apropriada; *simplicidade*, que corresponde a facilitar as operações que devem ser realizadas por um módulo complexo, quebrando-o em funções menores e *observabilidade*, que mede a facilidade de se observar porque um módulo faz determinada tarefa. Mostra como retratar, através do projeto estruturado, várias situações especiais que acontecem na realidade e que não foram abordadas especificamente nas publicações anteriores, de onde vem a maior contribuição desse seu trabalho. Além disso, fala sobre o desempenho de sistemas desenvolvidos por projeto estruturado, afirmando serem mais rápidos e utilizarem menos memória do que quando elaborados por soluções tradicionais. E, quando uma solução estruturada não é totalmente satisfatória quanto ao desempenho, deve-se identificar os módulos que consomem mais tempo de CPU e reescrevê-los numa linguagem de mais baixo nível, mas ainda de maneira estruturada. Diz também que uma das maiores vantagens do projeto estruturado corresponde à independência e reutilização dos módulos, pois sendo independentes eles podem ser facilmente alterados sem afetar os outros módulos do sistema e, por corresponderem a funções específicas podem ser reutilizados por outros sistemas através de simples chamadas a módulos já existentes.

Em 1982, Solvberg, em [So82], baseado nos conceitos principais da metodologia recomendada por Gane e Sarson [Ga77], propõe várias complementações ao diagrama de fluxo de dados dessa metodologia, tais como: utilização de um símbolo de sinal responsável pela retratação da dinâmica do sistema, pois através

dele, processos podem ser disparados ou não; utilização do símbolo de decisão para representar situações mutuamente exclusivas, como por exemplo, fluxos que saem de um mesmo processo, mas não ocorrem ao mesmo tempo; diferenciação entre depósitos de dados que guardam documentos dos que guardam dados em geral; representação de vários tipos de fluxo de dados, mostrando o que eles carregam, como por exemplo, formulários em branco, mensagens em papel, documentos preenchidos, etc.; representação de entidades responsáveis por determinadas tarefas; representação simbólica para definir a composição das estruturas de dados e algumas outras, que vêm retratar, de maneira detalhada, vários aspectos que não foram diferenciados na metodologia recomendada por Gane.

Muitos softwares de apoio ao desenvolvimento de sistemas de informação têm surgido. Um deles é o *proMOD*, publicado por Hruschka [Hr83] e desenvolvido pelo grupo GEI. Esse software compreende um conjunto de métodos para apoiar o desenvolvimento de sistemas, desde o início da análise até a fase de teste, fornecendo técnicas interativas para se obter um modelo imediato de cada fase do desenvolvimento. Assim, durante a fase de análise, o método oferecido é a análise estruturada, definido por Constantine e tratado por DeMarco [De78], com o qual se desenvolve o diagrama de fluxo de dados, o dicionário de dados e as especificações de processos através de tabelas de decisão, pseudocódigo ou textos comuns. Essas informações são armazenadas numa biblioteca e, com elas, o sistema oferece uma série de relatórios e referências cruzadas dos dados. Durante a fase de projeto é utilizada a técnica de projeto estruturado, de Yourdon e Constantine [Yo75]. O sistema, de acordo com as definições realizadas durante a fase de análise, sugere uma estrutura modular hierárquica inicial, que pode ser examinada, modificada e completada pelo analista, de acordo com regras dessa técnica. Além disso, de acordo com o pseudocódigo definido anteriormente, *proMOD* cria uma estrutura de programa preliminar que pode ser completada pelo usuário. O autor diz que a tendência desse sistema é gerar codificação de programas numa linguagem de alto nível, como PASCAL e FORTRAN.

Outro software de apoio ao projeto e documentação de sistemas de informação é o *Excellerator*, [In84], desenvolvido pela Index Technology Corporation. As principais facilidades ofe

recidas por esse software são referentes à elaboração de: *gráficos* - que permitem substituir o desenho manual dos diagramas lógicos do sistema e dos diagramas de apresentação. Existem quatro tipos de gráficos e diagramas oferecidos: diagramas de fluxo de dados, segundo a representação de Gane [Ga77], incluindo a elaboração da explosão de um processo; diagramas de modelos de dados cuja finalidade é mostrar a estrutura dos dados requerida para apoiar o sistema; gráficos de estrutura, que correspondem aos diagramas de estrutura modular da técnica de projeto estruturado desenvolvida por Yourdon [Yo75] e gráficos de apresentação que são apresentações não técnicas dirigidas a usuários e gerentes. Pode-se adicionar, atualizar, delir, copiar, renomear e listar qualquer um desses tipos de diagramas. *Dicionário* - que permite registrar informação sobre qualquer tipo de objeto utilizado em qualquer um dos diagramas, relatórios e documentos que se pode elaborar. Com ele, o analista pode padronizar os nomes, fazer uma referência cruzada de tudo que foi definido durante o projeto e avaliar vários aspectos produzindo relatórios a partir do dicionário de dados. *Telas* - fornecendo facilidades para se desenvolver rápida e facilmente protótipos de telas e cardápios com os quais o usuário irá interagir. *Relatórios* - permitindo que rapidamente se faça projeto e geração de modelos de relatórios, pois para muitos usuários o valor de um sistema está nos relatórios que ele gera. *Documentação* - produzindo um documento completo da especificação ou uma descrição narrativa do sistema descrito. Todos os gráficos, relatórios obtidos a partir do dicionário, telas, modelos de relatórios e até um texto narrativo do projeto podem ser montados juntamente em um único documento final, pronto para ser apresentado a gerentes ou clientes.

Masiero, em [Ms84], propõe uma nova abordagem à metodologia de análise estruturada, recomendada por Gane [Ga77], extrapolando os limites do sistema descrito às entidades externas, onde muitas funções típicas de automação de escritório podem ser identificadas. Para documentá-las, ele apoia a metodologia de análise estruturada com o método *RCI - Redes de Controle da Informação*, [E179], podendo com este inserir aspectos de dinâmica e informações temporárias que não ficam bem representadas com a outra metodologia.

Em 1985, Stevens, em [St85], recomenda enfaticamente a utilização de diagramas de fluxo de dados na descrição de qualquer tipo de aplicação, pois eles fornecem um modo natural de descrevê-las. Existem vários tipos de diagramas que podem ser usados durante o processo de desenvolvimento de sistemas. Dentre eles, dedica uma atenção especial ao diagrama de fluxo de dados recomendado por Gane [Ga77], por ser ele muito simples, permitir explosão de processos em níveis de maior detalhamento e por retratar tanto atividades manuais como atividades apoiadas por computador, de modo que pessoas não treinadas possam facilmente compreendê-lo. Diz que o diagrama de fluxo de dados é a maneira natural de comunicação e então, quando as aplicações são construídas por funções conectadas por fluxo de dados, eles são fáceis de compreensão, desenvolvimento e manutenção.

2.3 - INSTRUMENTAL PARA DESENVOLVIMENTO EXPEDITO DE SISTEMAS

Existem atualmente vários softwares que podem ser utilizados no desenvolvimento expedito de sistemas, tais como: planilhas eletrônicas, processadores de textos, gerenciadores de base de dados, etc. Com essas ferramentas, surge a possibilidade de se construir sistemas para necessidades específicas, a um custo mais próximo ao de muitos aplicativos comercializados, projetados para cobrir variações de uma mesma aplicação.

Dentre esses softwares, os que interessam a este trabalho são aqueles referentes à gestão de base de dados, como o *dBaseII* [Ra81, Kr83, Co85, To85] e o *dBaseIII* [By85, As84] e linguagens de consulta e manipulação de dados, como o *Datatrieve* [Di83a, Di83b, Di83c, Di83d], tendo em vista que, na implementação do caso prático abordado, o tipo de processamento dominante é o de manipulação de arquivos, e esta foi feita utilizando-se o *dBaseII*.

O *dBaseII* é comercializado como um sistema de gestão de base de dados relacional. Assim, as informações referentes às entidades são armazenadas em forma de tabelas, e os relacionamentos entre elas devem ser criados logicamente, pois isso não é feito automaticamente pelo sistema. O *dBaseII* possui recursos interativos de armazenamento e recuperação de dados, com comandos poderosos que trabalham com arquivos inteiros e com vários arquivos, como por exemplo, comandos para fazer contagens nos arquivos - COUNT, SUM, TOTAL; comandos para fazer indexação como SORT e INDEX, sendo que o último permite inclusive indexação em mais de um campo simultaneamente; comandos para atualizar registros - EDIT, REPLACE, CHANGE; comando para juntar dois arquivos em um terceiro - JOIN; comando para atualizar um arquivo a partir de outro - UPDATE, e muitos outros. Possui também recursos para impressão de relatórios, com o comando REPORT, que, através de um diálogo com o usuário, permite estabelecer cabeçalhos, totais e subtotais, o que facilita bastante essa tarefa. Além de operar interativamente com os usuários, o *dBaseII* também possui recursos para desenvolvimento de sistemas, pois ele contém sua própria linguagem estruturada de programação. Estruturada pelo fato de possuir as principais construções da pro

gramação estruturada que são: DO WHILE, ENDDO; IF, ELSE, ENDF; e DO CASE, ENDCASE; e, em parte, pelo fato de não possuir o comando GOTO o qual provoca, muitas vezes, a realização de uma programação não estruturada, embora, não seja ele o responsável por esse acontecimento, pois pode-se perfeitamente elaborar um programa estruturado utilizando-se o comando GOTO. Outro aspecto importante do dBaseII é a facilidade de elaboração de diálogos que permite o desenvolvimento de sistemas visando os usuários finais. Assim, ao se combinar as construções de programação estruturada com os recursos de elaboração de diálogos, obtém-se programas reutilizáveis e fáceis de modificar. Com o dBaseII pode-se ter acesso simultâneo a dois arquivos de dados (de extensão .DBF) diferentes, um chamado PRIMÁRIO e o outro, SECUNDÁRIO. Dependendo da operação que se deseja realizar, é necessário selecionar um ou outro, como por exemplo, para adicionar registros num arquivo ou para se posicionar num registro diferente; mas, sempre é possível usar os campos dos dois registros correntes sem que seja necessária nova seleção. Ainda quanto a arquivos, o dBaseII permite transformar "para" ou "de" arquivos não dBase, possibilitando, dessa forma, a interação com outras linguagens de alto nível, já que estas não podem usar diretamente os arquivos dBase. Outra facilidade suprida por esse software é a manipulação de erros, pois comandos inválidos e erros de sintaxe podem ser tratados através de um diálogo com o usuário, na medida em que aparecem. Com isso, a execução de um programa nunca é interrompida por esses motivos, pois faz-se as alterações e a execução continua normalmente. Portanto, utilizando-se o dBaseII, os dados podem ser adicionados, editados, eliminados, classificados, indexados e extraídos em relatórios, com grande facilidade de programação e, acima de tudo, tem-se também um certo grau de independência entre dados e programas, pois as estruturas de dados podem ser alteradas sem que sejam necessárias muitas modificações nos programas.

Apesar de todas essas facilidades acima citadas, o dBaseII possui também algumas limitações. Cada registro pode possuir apenas 32 itens de dados e, no máximo, 1000 caracteres e cada arquivo pode possuir até 65.535 registros. Os programas são um pouco vagarosos, pois são interpretados e não compilados.

Além disso, o dBaseII ocupa muita memória, o que faz com que os programas não caibam na mesma, aumentando o número de acessos ao disco e agravando o problema de tempo de processamento. A linguagem de programação permite apenas 64 variáveis e não dispõe de variáveis indexadas. E, finalmente, talvez a pior limitação dentre todas essas: apenas dois arquivos de dados podem ser usados simultaneamente. Embora esse último problema possa ser solucionado, o tempo de processamento é um pouco afetado com essa solução.

Já, o dBaseIII, oferece maior versatilidade quanto ao número de arquivos que podem ser utilizados simultaneamente, visto que a sua limitação é bem menos significativa que a do dBaseII, pois ele possui a capacidade de manter até dez arquivos abertos ao mesmo tempo, o que facilita bastante a programação. Enquanto que, no dBaseII, para se ter acesso a campos que possuem o mesmo nome nos arquivos primário e secundário, usa-se respectivamente antes do nome "P." ou "S.", no dBaseIII, o prefixo para o nome do campo é o nome do arquivo. Outra facilidade que foi implementada no dBaseIII, refere-se à pesquisa de registros em um determinado arquivo, segundo um campo contido em outro. Enquanto que, no dBaseII, isso é feito iterativamente, selecionando-se um registro do arquivo primário e depois, procurando-se no arquivo secundário o registro que possui o mesmo conteúdo para aquele campo, no dBaseIII faz-se isso através do comando SET RELATION TO, o qual relaciona dois arquivos segundo um campo especificado. O dBaseIII possui cinco tipos de dados: caracter, numérico, lógico, data e memo. Dentre esses tipos, os três primeiros são também oferecidos pelo dBaseII. Quando um campo é declarado do tipo *data*, no momento de inserção dos dados, o formato apresentado corresponde àquele usado para datas, isto é, com as barras dividindo o dia, mês e ano (DD/MM/AA). Nesse momento, os dados são validados e, se for digitada uma data impossível, uma mensagem de erro é enviada. Além disso, pode-se fazer alguns tipos de operação com esse campo, como por exemplo, subtrair uma data de outra, adicionar ou subtrair dias de uma data, escrever o mês por extenso, solicitar o referido dia da semana, recuperar, separadamente, o dia, o mês ou o ano, e outras. O outro tipo especial, que também não existe em dBaseII, é o tipo *memo*. Campos desse tipo contêm textos,

como os campos tipo caracter, mas podem possuir até 4.000 caracteres. No arquivo de dados, eles ocupam apenas 10 caracteres, pois o texto propriamente dito é mantido em outro arquivo auxiliar. Os dados armazenados nos campos de tipo memo são, por exemplo, memorandos, resumos, comentários, etc.. Quanto às limitações, o dBaseIII permite até um bilhão de registros por arquivo; no máximo, 4.000 caracteres por registro e, até 128 campos por registro. Comparativamente, nota-se que esses números são bem maiores que os do dBaseII, o que significa uma maior potência com relação a ele. Embora existam várias outras inovações no dBaseIII, as mais marcantes são essas descritas acima.

Esses dois softwares de manipulação de arquivos compõem os recursos disponíveis para desenvolvimento expedito de sistemas quando se utiliza microcomputadores. Assim como eles, pode-se citar também, o Datatrieve, para computadores de médio porte, que é um dos softwares mais utilizados da Digital.

O Datatrieve é uma linguagem de 4ª geração que utiliza definições e comandos, no lugar de procedimentos complicados, para manipular os dados, isto é, diz-se o que se deseja, sem dizer como fazê-lo. Ela interage com outras linguagens de programação, de forma que seus comandos podem ser chamados a partir destas, ou então, dependendo da aplicação, esta pode ser resolvida, utilizando-se apenas o Datatrieve. Esse software oferece um acesso relacional aos dados, os quais são descritos através de estruturas específicas chamadas domínios (DOMAIN), que contêm todas as informações necessárias sobre os relacionamentos entre os dados e as descrições dos mesmos. Esses arquivos correspondem àqueles de extensão ".DBF" do dBaseII e dBaseIII. No entanto, diferentemente destes, quando se utiliza o Datatrieve, define-se tudo separadamente, isto é, a estrutura do registro, com os campos que o compõem, o arquivo de dados e um domínio que serve para associar os nomes dos dois, ou seja, a definição de um domínio determina que a definição de um registro específico é usada para interpretar os dados de um determinado arquivo. Os campos de um registro são definidos, normalmente, pela cláusula PICTURE e admitem o tipo OCCURS (como em COBOL). O Datatrieve permite a definição de visões específicas dos dados, que podem ser compostas por campos de um ou mais domínios, e que são

usadas para atender uma determinada aplicação. Ele possui vários comandos para se manipular os dados interativamente, os quais também podem ser agrupados, formando um procedimento, que é criado através da definição de uma PROCEDURE. Dentre esses comandos, pode-se citar aqueles para selecionar registros de um arquivo com determinada condição, modificar campos de um registro, combinar arquivos, e muitos outros, além daqueles que são comuns à maioria das linguagens de programação como IF THEN ELSE, REPEAT, BEGIN END, FOR, etc.. O Datatrieve também possui recursos para elaboração de relatórios, que permitem definição de cabeçalhos, sumarização de dados como contagem de registros, totais e médias dos valores de um campo, e outros. Além disso, oferece também facilidade para representação de dados graficamente, tais como gráfico de barras, gráfico de fatias, histogramas, etc., e alguns tipos de análise estatística com os mesmos. Na verdade, o Datatrieve interage com uma série de produtos da Digital. Ele pode ser usado para armazenar, atualizar e recuperar informação tanto de sistemas locais como de sistemas remotos. Ele trabalha igualmente com o sistema VAX DBMS, que é um gerenciador de base de dados tipo CODASYL, com o sistema VAX rdb/VMS, que é um gerenciador tipo relacional, e também, com arquivos comuns de usuários. Ele é um excelente recurso para construção de protótipos e desenvolvimento de aplicações, pois com seus recursos de linguagem de 4ª geração, permite que se desenvolva aplicações em muito menos tempo do que se usando linguagens tradicionais de 3ª geração. Todas essas facilidades são possíveis devido ao CDD (Common Data Dictionary), onde estão registradas todas as informações sobre os arquivos de dados.

Pode-se dizer que, independente do tipo de equipamento disponível, quer seja ele um microcomputador ou um computador de porte maior, tem-se, hoje em dia, uma grande facilidade para desenvolvimento expedito de sistemas, pelo fato da diversidade de softwares especializados existentes e que facilitam tarefas desse tipo.

Em 1983, Bowerman, em [Bo83], coloca esse problema dos softwares especializados, especialmente dos sistemas relacionais de base de dados para microcomputadores, como uma ameaça e também uma oportunidade, pois os usuários estão cientes des

ses produtos, e de que, com eles, desenvolve-se sistemas em muito menos tempo, comparando-se com a abordagem tradicional e, por outro lado, se bem aproveitados esses recursos, os sistemas relacionais de base de dados para microcomputadores são a solução de muitas funções que os usuários desejam realizar. Ele salienta que o ciclo de vida desses sistemas é bem mais rápido que o ciclo de vida tradicional, pois ele envolve menos passos e estes também são mais simples. Dentre os três tipos de sistemas de gestão de base de dados, em rede, hierárquico e relacional, Bowerman diz que o menos complexo deles é este último; daí os vários existentes, sendo que de todos eles um dos mais comercializados é o dBaseII.

2.4 - COMPUTADORES E MICRO EMPRESAS

Embora o problema do uso de computadores nas micro empresas tenha se evidenciado nos últimos anos com a evolução, difusão e barateamento dos microcomputadores, ele vem sendo estudado já há muito tempo, desde os anos 60. Nota-se que várias tentativas foram feitas desde então, mas muitas delas são hoje, inviáveis.

Em uma das primeiras publicações sobre o assunto, Hunt, em [Hu68] coloca que, no final da década de 60, iniciou-se um estudo dos possíveis usos de computadores em pequenas firmas. Tais firmas foram definidas como aquelas pequenas o bastante, para serem examinadas por um analista experiente, em dois ou três dias, no máximo. De acordo com o ponto de vista dessas firmas, quanto ao uso de computadores, o tamanho de suas equipes ou seus movimentos anuais eram bem menos importantes do que a complexidade da organização interna necessária para controlá-las. Mas, independente disso, a maior parte das firmas não haviam, por si mesmas, considerado o uso de computadores. Elas foram examinadas, fazendo-se uma referência especial aos seus procedimentos correntes de operação e então, foram classificadas de acordo com as necessidades de sistemas e de gerenciamento. Projetou-se então, um conjunto de *blocos funcionais de construção de software* que permitiam às firmas se ajustarem às categorias classificadas e reunirem seu software inicial, o qual dava condições à firma de utilizar o computador sem grandes mudanças na estrutura gerencial. Esse software não era tão eficiente quanto programas desenvolvidos individualmente, mas removia dois aspectos principais que impediam o uso de computador por pequenas firmas: a necessidade de uma reorganização radical com o uso do computador e o alto custo do software inicial. No processo de pesquisa dessas firmas, foram utilizadas várias alternativas e a que pareceu mais razoável, foi a elaboração de um diagrama de fluxo, mostrando a rotina das informações. Com esse procedimento, construiu-se modelos de diagramas que atendiam quase que inteiramente a todas as firmas pesquisadas, com pequenas modificações para algumas delas apenas. De acordo com essa pesquisa, notou-se que o uso desses modelos forneceu um esquema de classificação pa

ra as firmas examinadas, pois o número de modelos diferentes obtido foi pequeno. Além disso, esse esquema de classificação era apropriado para ajustar as firmas aos sistemas de informação baseados em computador. Na verdade, se o número de modelos permanecesse baixo, seria conveniente descartar o projeto inicial, do desenvolvimento de blocos de construção de software e, em vez disso, desenvolver *sistemas completamente integrados* que correspondessem a cada modelo separadamente.

Continuando a mesma pesquisa descrita anteriormente, da qual já havia participado, Buckle, em [Bu69], refere-se ao desenvolvimento de um outro método a ser adotado para agrupar as firmas, devido à similaridade dos sistemas usados em cada grupo, mesmo aquelas que lidavam com produtos diferentes. Várias outras firmas foram pesquisadas, apenas com a finalidade de se conhecer seus tamanhos e seus tipos de serviços. Os dados obtidos dessa pesquisa foram considerados de interesse, por duas razões: primeiro, por confirmarem os resultados da pesquisa anterior, isto é, de que uma grande quantidade de firmas podia ser descrita por um único modelo de procedimento; segundo, por permitirem a preparação de outro modelo de procedimento, considerado menos importante, para a maioria das firmas de prestação de serviço. No entanto, como grande porcentagem das firmas se ajustavam ao primeiro modelo, elas foram agrupadas e um sistema básico foi projetado para elas. Os objetivos desse projeto eram: atender as firmas o mais rapidamente possível, principalmente aquelas não tão pequenas, de modo que as vantagens pudessem ser vistas de imediato, e também, deixar o sistema maleável a crescimento futuro. Suas entradas eram todas feitas por cartão, pois segundo o documento, o uso de cartões dava aos usuários a sensação de algo possível de ser controlado, diminuindo a resistência das firmas com relação ao novo sistema de informação. Portanto, com esse projeto, confirmou-se os resultados da pesquisa anterior e o trabalho dos autores foi direcionado, definitivamente, para a preparação de um sistema integrado ao invés de blocos de construção.

Ainda em continuidade aos dois trabalhos anteriormente citados, Buckle, em [Bu70], apresenta uma versão do projeto para o outro modelo de procedimento, que atende as firmas de

prestação de serviço. E, para finalizar, faz uma pesquisa sobre os pacotes existentes para atender as pequenas firmas, obtendô o resultado de que pouca coisa já existia nessa área, amenos de programas de previsão. No entanto, estes envolviam taxas de estimativa que poderiam avaliar de maneira incorreta a demanda e, as pequenas firmas não tinham a experiência suficiente para levar isso em consideração. Além disso, elas eram muito vulneráveis e as tendências de mercado indicadas pelos pacotes poderiam trazer resultados desastrosos. Dessa forma, as pequenas firmas estavam sempre em desvantagem.

Em 1975, Marienthal, em [Ma75], destaca a crescente utilização dos recursos computacionais pelos pequenos usuários, que estava sendo diretamente proporcional à diminuição do preço dos equipamentos. Esses usuários começavam a instalar, principalmente minicomputadores e a utilizá-los de novas maneiras, quase sempre sem ter no seu quadro profissionais especializados. Diz que nessa época, apareceram os primeiros equipamentos, que não eram da IBM, sendo que, muitos deles eram comercializados por negociantes independentes ou intermediários. Os pequenos usuários ganharam um certo status nesse mercado, pois o custo de instalação e operação dos minis era bem menor que dos equipamentos necessários aos projetos convencionais. Essa redução de preço era decorrente das novas técnicas de mercado dos novos fabricantes, da utilização de terminais para a entrada de dados, ao invés de perfuradoras de cartão e da diminuição ou muitas vezes, eliminação de um departamento de processamento de dados. Intermediários ajudavam tanto os vendedores como os usuários e eram companhias que compravam hardware e vendiam hardware e software; eles simplificavam a instalação e implementação dos sistemas aos usuários, e os sistemas, por sua vez, simplificavam a entrada de dados, davam suporte à organização de processamento de dados e ao desenvolvimento de aplicações. O autor coloca a entrada de dados por terminais, como um dos fatores mais importantes na utilização de minis pelas pequenas empresas. Na verdade, algumas delas faziam a aquisição dos mesmos, pelo fato de poder contar com esse recurso. Utilizando-se terminais, a tarefa de entrada de dados pode ser realizada na própria fonte destes, de forma a combinar os passos para prepa-

ração dos documentos, edição, perfuração e verificação. Assim, a exatidão dos dados é muito maior, pois a pessoa que os insere está atenta ao seu significado. Além disso, os *softwares de aplicação* para os minis estavam sendo produzidos a um preço bem menor do que para os grandes equipamentos. Dessa forma, os pequenos sistemas possibilitavam às pequenas firmas possuir um equipamento próprio, a custo de processamento em bureau. O autor cita alguns exemplos de pequenos sistemas instalados em algumas companhias, todos alcançando sucesso e satisfação dos usuários. Dentre essas companhias, algumas delas já haviam experimentado por muito tempo serviço de bureau e depois então, foi que adquiriram um mini computador.

Em 1977, Burns, em [Br77], fala da utilização de computadores nas empresas, com *recursos de sistemas de informação de escritórios*, como por exemplo, processadores de palavra. Ele diz da cansativa tarefa de redigir memorandos, cartas e relatôrios sem esse recurso, pois com ele, grande parte do trabalho de reescrita desses documentos, durante a fase inicial de um rascunho preliminar, poderia ser eliminada. No entanto, muitas vezes não se podia instalar um equipamento em cada estação de serviço e então, fazia-se um centro de processamento de palavra localizado, como se fosse uma "secretária de correspondência". Devvido a este e outros fatores, a introdução de processadores de palavra foi uma experiência frustrante para algumas empresas pois estas não eram capazes, dessa forma, de agilizar o equipamento e o trabalho. Mas, muitos outros tipos de serviços passavam a ser solicitados, a nível de apoio gerencial, como necessidade de comunicação, melhores sistemas para tornar mais eficientes as operações do gerente e eliminação de várias funções da secretária. O autor diz que com a tendência de barateamento dos equipamentos, muitas aplicações ainda seriam desenvolvidas para acompanhar a evolução dos sistemas de informação de escritórios. Dentre essas, ele deu ênfase ao processamento de palavras com reursos de tela e comunicação com equipamentos remotos e também, ao armazenamento e recuperação de imagens, através de arquivos de informações relevantes, que poderiam ser consultadas também por terminais remotos.

O fato de uma verdadeira invasão de microcomputado

res, no lugar dos minis, que é o que acontece atualmente, já ha via sido mencionado por Canning e McNurlin em [Ca78]. Nesse ar tigo, os autores dizem que, da mesma forma como vinha acontecendo com a introdução dos minicomputadores nos serviços de processamento de dados, o mesmo já começava a acontecer com os micros, e logo esse fenômeno alcançaria grandes proporções. Dessa for ma, colocava-se as ferramentas de processamento de dados nas mãos dos usuários finais, fazendo com que estes ficassem responsáveis pelos seus processamentos, o que estava levando a uma uti lização de sistemas de aplicações previamente desenvolvidos ou de sistemas desenvolvidos através de serviços contratados. Os micros começavam a oferecer a capacidade dos grandes sistemas a preços bastante acessíveis. Devido ao baixo custo, o micro passava a ser acessível aos consultores e às pequenas companhias de software, ocasionando uma rápida proliferação dos *pacotes apli*cativos. Eles citam alguns problemas que estavam ocorrendo com certa frequência: compra precipitada, por causa do baixo custo que impressiona as pessoas, ocasionando, muitas vezes, a aquisição de equipamentos inadequados; perda de tempo imposta pelos pacotes de software pois estes sempre exigem alguns acertos pa ra atenderem problemas específicos e projeto inadequado dos sis temas, programas, pacotes bem como das aplicações encomendadas.

Marienthal, em [Ma78], diz que já naquela época, com a abertura de mercado para os pequenos sistemas, os vendedo res de mini e microcomputadores haviam encontrado novas maneiras de vender seus produtos aos usuários finais. Ele relaciona como principais vias de comercialização dos minis e micros, em especial daqueles a serem adquiridos pelas pequenas firmas, as seguintes: vendas diretas tradicionais, vendas através de dis tribuidores com regiões geográficas definidas, vendas através de firmas de O&M e vendas através de varejistas locais (lojas de micros). *Vendas diretas* referia-se ao modelo tradicional de vendas, isto é, através de uma filial de um fabricante de computador, formada por pessoal de venda, engenheiros de sistemas e de manutenção que davam apoio aos clientes. Esse ramo oferecia software de sistema operacional e pacotes aplicativos, sendo que esses últimos eram, principalmente, *pacotes horizontais*, isto é,

referentes ao desenvolvimento de funções específicas que serviam a vários tipos de negócios, funções essas relativas à Administração Geral, como por exemplo, recursos humanos, finanças, contabilidade, etc. e, em muitos poucos casos, eram pacotes verticais, compostos do desenvolvimento de várias funções que se aplicavam diretamente a um ramo empresarial específico que incluam sistemas terciários, isto é, aqueles que dizem respeito ao próprio objetivo da empresa. Esses softwares eram desenvolvidos pelos fabricantes de computadores ou pelas empresas de software, e entregues ao usuário com uma documentação elaborada. No entanto, tais fabricantes encontravam sérios problemas na instalação, para fazer as modificações referentes ao cliente específico, pois não contavam com pessoal especializado para tal. Esse serviço era então feito por programadores autônomos ou por empresas de software. Descontentes com essa solução, os fabricantes de computador tentavam desenvolver seus pacotes da maneira mais geral e flexível possível, de forma que tais modificações pudessem ser feitas na hora. O segundo tipo de via de comercialização, os distribuidores regionais, eram organizações de vendas de vários produtos, credenciadas pelos fabricantes. Estes forneciam aos distribuidores o hardware, material de venda e software de sistema operacional, sendo que raramente o faziam com relação a pacotes aplicativos, mas, em vez disso, tentavam patrocinar intercâmbio entre os vários distribuidores. Normalmente, estes tinham sua própria equipe de programadores, ou então usavam empresas de software locais e autônomos. Nesses termos, o distribuidor era uma pequena empresa, muito mais flexível que as filiais ditas anteriormente. Então, um arranjo entre distribuidor, empresa de software e cliente podia tomar várias formas: a programação podia ser feita sob contrato com o cliente, ou o distribuidor podia subcontratar a programação. Nesse caso, o distribuidor entregava um sistema fechado e o cliente só pagava depois de estar satisfeito com a programação. Para o distribuidor, o software era considerado um mal necessário, ou o fator que proporcionava a venda dos produtos. No entanto, isso não queria dizer que o software produzido pelo distribuidor era de pior qualidade, pois seu território era limitado e boas referências eram muito importantes para ele. As firmas de O&M, o terceiro tipo de via de comercialização, eram aquelas que compravam componentes

de vários fabricantes, adicionavam software e vendiam sistemas computacionais já funcionando. Várias delas tentavam se especializar no mercado vertical e o foco de seus negócios era o software. Firms de O&M bem sucedidas se especializaram nesse mercado; dessa forma, o seu software se tornou um bem ativo, pois mediante especialização, este podia ser reutilizado com mínimas modificações e com maiores chances de satisfazer o usuário. Finalmente, o quarto tipo, *os varejistas*, eram aqueles que vendiam produtos que eram geralmente classificados como sistemas baseados em microcomputadores. O autor coloca como tendência futura, o fato interessante de que talvez as funções dos computadores se tornem tão padronizadas, a ponto de que um pequeno empresário peça a um varejista um "computador para contas a receber", isto é, o hardware se tornará estreitamente associado ao software. Os pacotes não atuam tão bem nas aplicações relacionadas à essência do negócio. Os computadores são comprados pela sua flexibilidade e os usuários esperam que eles atendam as suas necessidades e não que eles, usuários, tenham que mudar sua maneira de executar o serviço para se adaptarem aos pacotes de software.

Em 1978, McCartney, em [Mc78], fala dos microcomputadores atuando em todos os tipos de negócios das pequenas empresas, desde a farmácia, o escritório de corretagem de seguros, e outros. Na verdade, os *pequenos sistemas* se haviam tornado fundamentais na indústria de processamento de dados, e estavam proporcionando um alastramento de microcomputadores. Os vendedores de grandes equipamentos, os de minicomputadores ou os de microcomputadores, estavam correndo ao encontro desse mercado. Pequenos sistemas para todas as funções estavam surgindo, a preços bastante acessíveis, de modo que uma clínica médica, um médico no seu consultório particular, uma firma de advocacia, todos eles podiam ter o seu próprio equipamento. Esses pequenos usuários estavam utilizando os pequenos sistemas nos seus negócios e, muitas vezes, eles próprios estavam instalando seu equipamento e escrevendo o seu software, sem o auxílio de profissionais especializados. Em geral, eles compravam seus equipamentos diretamente dos varejistas, que eram especializados na venda de microcomputadores.

Myers, em [My83], amplia as previsões de Marienthal, em [Ma78], sobre as tendências futuras, dizendo que muitas coi

sas que os minis não foram capazes de fazer, os micros o serão, através de redes de comunicação, onde estes trocarão informações com computadores de grande porte ou mesmo minicomputadores. Dessa forma, pequenas empresas poderão ter o seu próprio microcomputador funcionando como terminal inteligente que, quando necessário, interage com os outros computadores da rede, em busca de informação. Ele observa também que os próprios bureaux estão utilizando os microcomputadores como veículo de venda de seus produtos.

A Centurion, em [Ce84] fala sobre a utilização dos microcomputadores nas empresas, quer seja através de aplicativos quer de sistemas personalizados. No primeiro caso, a utilização dos micros possibilita o "escritório automatizado", melhoria da produtividade individual dentro da organização, apoio ao processo decisório dentro da empresa através da gestão das informações, interação com equipamentos de grande porte, etc.. No segundo caso, os micros permitem que os próprios usuários ou pessoas especializadas desenvolvam aplicações, com finalidade específica, embora essa solução seja discutível, quanto a relação custo/benefício. São também enumeradas várias cautelas para a utilização de microcomputadores, de modo que sua aquisição não traga resultados frustrantes. Fala também na necessidade de uma estratégia de implementação, como ponto fundamental para obter-se êxito na utilização dos micros em qualquer organização. Cita várias atividades que não podem deixar de ser realizadas para que esse êxito seja atingido, como por exemplo, planejamento dos objetivos, detalhamento de cada processamento requerido, escolha cuidadosa do hardware, escolha do modo em que os requisitos da organização serão atendidos, se através de aplicativos ou desenvolvimentos personalizados, elaboração de documentação adequada e treinamento de usuários. Diz também que um dos empregos mais eficientes do micro é quando ele atua como estação de trabalho, no apoio aos executivos e, além disso, fala da pequena empresa como sendo o mercado mais promissor para ele.

2.5 - CONCLUSÕES

Com relação ao instrumental de análise estruturada, percebe-se, nas publicações dos últimos dez anos, que esse assunto sofreu uma grande evolução nesse período e contribuiu bastante para o desenvolvimento de sistemas de informação. Os precursores das várias metodologias, atualmente utilizadas, foram principalmente Constantine e Yourdon. Através das idéias lançadas por eles, esse assunto desabrochou, dando origem a várias técnicas de documentação de análise e projeto, marcando definitivamente uma nova maneira de atacar o processo de desenvolvimento de sistemas. Dentre elas, encontra-se a metodologia recomendada por Gane e Sarson [Ga77], que foi utilizada na elaboração da análise do caso prático abordado, descrita no Capítulo 3 desta dissertação.

De acordo com essa análise estruturada realizada, observou-se que, no Escritório de Corretagem de Seguros, havia um tipo de processamento dominante, referente ao gerenciamento de informação. Levando-se esse fato em consideração e também por se tratar de um ambiente de pequena empresa, procurou-se levantar informações na bibliografia sobre os tipos de soluções possíveis para esse caso, através de uma análise, inicialmente da utilização de computadores nas empresas, até se chegar à época da utilização de microcomputadores nas micro empresas e identificar-se a abordagem de desenvolvimento vertical que trata, de maneira especializada, os problemas de um ramo de negócios específico.

Assim, acreditando-se ser essa a situação e, além disso, tendo em mãos um tipo de processamento específico, procurou-se abordar referências bibliográficas sobre os diferentes softwares para desenvolvimento expedito de sistemas que atendessem ao problema em questão, concentrando-se no dBaseII, que foi utilizado na implementação, como é mostrado no Capítulo 4 e no Apêndice.

Portanto, neste capítulo, forneceu-se uma visão geral dos três aspectos que estão interrelacionados nesta dissertação: metodologias de análise estruturada, recursos para desen

volvimento expedito de sistemas e utilização de microcomputadores em micro empresas. Reunindo essas informações bibliográficas, conseguiu-se fundamentos para se fazer uma avaliação, no Capítulo 5, da experiência realizada.

C A P Í T U L O 3

ANÁLISE ESTRUTURADA DO CASO PRÁTICO ABORDADO

3.1 - CONSIDERAÇÕES GERAIS

Neste capítulo é apresentada a Análise Estruturada do Escritório de Corretagem de Seguros, descrito no capítulo 1. Essa análise, tratada com detalhes em duas outras publicações da mesma autora, [Fa82] e [Fa84], foi feita através da metodologia recomendada por Gane [Ga77].

Primeiramente, no item 3.2, fala-se na facilidade de comunicação alcançada entre analista e usuário quando do uso dessa metodologia para descrição de sistemas.

Em seguida, no item 3.3, mostra-se os Diagramas de Fluxo de Dados, que são compostos por um diagrama geral do sistema e por diagramas detalhados, referentes a explosões de certos processos do diagrama geral.

No item 3.4, discute-se uma amostra do Dicionário Manual de Dados, que foi elaborado para o sistema de acordo com os diagramas construídos, e que possui a descrição de todos os objetos que os compõem. É no dicionário de dados, que se encontra completo em [Fa84], onde se obtém todos os detalhes que não ficam explícitos nos diagramas de fluxo de dados.

Finalmente, no item 3.5, faz-se algumas conclusões sobre a utilização dessa metodologia, as facilidades e dificuldades encontradas e as vantagens conseguidas.

3.2 - COMUNICAÇÃO COM O USUÁRIO

Fala-se muito na grande dificuldade de comunicação entre analista e usuário, principalmente durante a fase de análise de um sistema. Essa comunicação é fator de extrema importância, pois é nessa fase de análise que o usuário transmite suas necessidades. A dificuldade de comunicação é um obstáculo ao usuário que, não conseguindo se expressar com clareza, faz com que, muitas vezes, nem aquilo que é captado pelo analista esteja de acordo com o que o usuário pretende transmitir.

Esse é um aspecto bastante interessante na metodologia usada, [Ga77], pois a própria elaboração do diagrama de fluxo de dados torna-se a linguagem de comunicação entre analista e usuário. Isso não é constatado de imediato mas, logo se nota que essa é a solução. Na primeira reunião, houve uma tentativa por parte da analista, de anotar em forma narrativa, todas as informações que lhe estavam sendo transmitidas. No entanto, no momento de transformar essas informações em diagramas de fluxo de dados, notou-se que algumas informações relevantes passaram despercebidas e, conseqüentemente, não puderam ser registradas nos mesmos, como decorrência daquela primeira reunião. Com isso, pôde-se observar que a maneira mais adequada de se registrar as informações era elaborar o próprio diagrama de fluxo de dados à medida em que elas estivessem sendo transmitidas. Mas, para isso, era preciso que o usuário recebesse algumas noções de como tais diagramas eram elaborados, a fim de participar desse processo. Facilmente, essas noções foram transmitidas pela analista e captadas pelo usuário e, a partir daí, a própria elaboração do diagrama de fluxo de dados tornou-se um denominador comum entre as duas partes.

Dessa forma, a atividade de descrição do sistema tornou-se bem mais simples, pois o usuário pôde participar intensamente desse processo, garantindo assim, que o sistema descrito se aproximasse bastante da realidade.

3.3 - ELABORAÇÃO DO DIAGRAMA DE FLUXO DE DADOS

O diagrama de fluxo de dados (DFD) define o fluxo de dados lógico do sistema, isto é, "o que" o sistema faz e não "como" faz. Ele é composto de quatro tipos de objetos: *Entidade Externa*, *Processo*, *Depósito de Dados* e *Fluxo de Dados*, aos quais está associada a seguinte simbologia:

Entidade Externa:

- representa coisas ou pessoas que interagem com o sistema sendo fonte ou destino das informações.
- (*) qualquer letra minúscula, que tem por finalidade, identificar a entidade.

Processo:

- representa as atividades realizadas no sistema.
- (*) número atribuído ao processo para identificá-lo.
- (**) sentença imperativa, formada por um verbo que representa uma ação, seguido de um objeto.

Depósito de Dados:

- representa os "armazéns" de dados do sistema, que fisicamente, podem ser um fichário, uma pasta, um arquivo, etc..
- (*) um número que segue a letra D e que serve para identificar o depósito.
- (**) nome atribuído ao depósito que deve coincidir com o nome utilizado na realidade.

Fluxo de Dados:

- representa um sistema de "canalização" por onde fluem as informações do sistema.

- (*) nome atribuído ao fluxo de dados e que deve englobar todas as informações que passam por ele.

A elaboração do DFD é um processo iterativo que de manda a construção de várias versões, até que se chegue a uma versão definitiva. Primeiramente, constroi-se um DFD geral pa ra o sistema em questão, Figura 3.1, no qual são identificadas as funções principais do mesmo e depois, então, é que essas funções vão sofrendo explosões em DFD's mais detalhados, Figuras 3.2 a 3.5.

O diagrama geral de fluxo de dados fornece uma vi são geral do sistema e os seus processos representam, ou o agru pamento de um conjunto de atividades, como é o caso dos proces sos 2, 3, 8 e 9 que, em consequência disso, sofreram explosão, ou então, processos que representam atividades singulares e que portanto, não necessitaram detalhamento, como é o caso dos pro cessos 1, 4, 5, 6 e 7. Dentre esses últimos, exceto o processo 1, que não necessitou explosão por ser bastante simples e com preensível, os outros foram colocados no diagrama geral por tra tarem, na verdade, de atividades que são compartilhadas pelos demais processos do sistema e que portanto, não poderiam ser co locados como parte do detalhamento de um processo específico. Entende-se, dessa forma, que esses processos são utilizados pe los demais.

O diagrama detalhado é elaborado da mesma forma que o diagrama geral, com a diferença de que o primeiro refere-se ao refinamento de algum processo específico do segundo.

Para finalizar, pode-se dizer que o diagrama lógico de fluxo de dados identifica e denomina todas as atividades lô gicas, mostra as origens e destinos das informações, identifica e denomina os grupos de informação que ligam uma atividade a ou tra e também, identifica os depósitos de informações aos quais as atividades fazem acesso. A elaboração dos diagramas é um pro cesso de refinamento sucessivo, que se estende até que não seja mais útil subdividir as atividades lógicas que foram identificadas.

Maiores informações sobre a elaboração dos diagramas de fluxo de dados desse sistema, as convenções para exploração, etc., podem ser encontradas em [Fa84].

Figura 3.1 - Diagrama Geral de Fluxo de Dados do Sistema Manual de Acompanhamento de Atividades de Corretagem de Seguros.

Figura 3.4 - Explosão do Processo 8

Figura 3.5 - Explosão do Processo 9

3.4 - ELABORAÇÃO DO DICIONÁRIO DE DADOS

No item anterior, viu-se a elaboração do diagrama de fluxo de dados e, pelo que foi dito, nota-se que à medida que se faz o refinamento do mesmo, aumentam os detalhes dos fluxos de dados, depósitos de dados e dos processos e portanto, torna-se necessário um lugar para manter todos esses detalhes. Esse lugar é proporcionado pelo *Dicionário de Dados*. A elaboração deste consiste em:

- analisar cada fluxo de dados e descrevê-lo juntamente com as estruturas e elementos de dados que fluem por ele;
- descrever todos os depósitos de dados e todas as informações que neles estão armazenadas;
- descrever também os processos, especificando sua lógica, passo a passo, sendo que isto só é feito, quando não é mais útil subdividi-los em outras funções lógicas, originando portanto, outro diagrama detalhado de fluxo de dados.

É então, o dicionário de dados, um conjunto de "dados sobre os dados" e, grande parte de seu valor reside no fato dele ser um depósito central de dados para todos aqueles que estão envolvidos com o sistema.

No dicionário de dados pode-se descrever todos os objetos usados no diagrama de fluxo de dados (exceto as entidades externas, que normalmente não são descritas), além de estruturas e elementos de dados, que não aparecem explicitamente no mesmo. A maneira de descrevê-los encontra-se resumida a seguir e, com maiores detalhes, em [Fa84]:

- a) *Processos*: são as atividades realizadas no sistema. Por exemplo, pode-se citar o processo 2.1 - *Preencha Proposta e Ficha de Controle de Emissão*, Figura 3.6. As informações contidas no dicionário de dados sobre os processos são: nome; número de referência (consiste de um número atribuído a ele); descrição (consiste de uma explicação resumida sobre ele); entradas (consiste de to

dos os fluxos que chegam a ele); saídas (consiste de todos os fluxos que saem dele) e lógica (consiste de uma descrição, passo a passo, da lógica do processo e que, no caso em questão, foi feita utilizando-se português estruturado).

PREENCHA PROPOSTA E FICHA DE CONTROLE DE EMISSÃO

Processo-ref: 2.1

Descrição:

Através dos dados fornecidos pelo cliente, preenche-se a proposta do seguro e anota-se a data em que está foi enviada à seguradora.

Entradas	Lógica	Saídas
a-2.1 dados para preenchimento da proposta.	<div>Colete dados do cliente</div> <div>Preencha a proposta de seguro de acordo com os dados fornecidos.</div> <div>SE for seguro à vista, ENTÃO envie a proposta e o pagamento para a seguradora.</div> <div>SE for seguro à vista com prazo para pagamento, ENTÃO envie a proposta para a seguradora.</div> <div>SE for seguro financiado, ENTÃO execute processo 2.2</div> <div>Registre a data de emissão na ficha de controle de emissão e a anexe na cópia da proposta.</div> <div>Arquive a cópia da proposta em D2.1</div>	<div>2.1-D2/1 cópia de proposta e ficha de controle.</div> <div>2.1-2.2 financiado</div> <div>2.1-b proposta com ou sem pagamento antecipado</div> <div>2.1-a recibo de recebimentos</div>

Figura 3.6 - Processo: 2.1 - Preencha Proposta e Ficha de Controle de Emissão

b) *Fluxos de Dados*: são um sistema de "canalização" pelo qual as estruturas fluem. Por exemplo, po de-se citar o fluxo *Cópia de Proposta e Ficha de Controle*, Figura 3.7. As informações contidas no dicionário sobre os fluxos de dados são: nome; fonte e descrição (consiste da identificação da entidade externa, processo ou depósito de dados e seu respectivo nome); destino e descrição (consiste da identificação da entidade externa, processo ou depósito de dados e seu respectivo nome); descrição (consiste em dizer quando ou porque o fluxo ocorre); estruturas de dados perten

centes (consiste das estruturas que fluem por ele) e volume de informação (consiste, quando possível, da quantidade e frequência das informações que passam por ele).

<u>CÓPIA DE PROPOSTA E FICHA DE CONTROLE</u>	<u>Fluxo de Dado</u>
Fonte: <u>2.1</u> Descrição: <u>PREENCHA PROPOSTA E FICHA DE CONTROLE DE EMISSÃO</u>	
Destino: <u>D2/1</u> Descrição: <u>PASTA PARA CÓPIAS DE PROPOSTAS</u>	
Descrição Expandida: <u>A cópia da proposta ficará arquivada junto com a ficha de controle de emissão até que chegue a apólice correspondente a essa cópia de proposta.</u>	

Estruturas de Dados Pertencentes:	Volume de Informação
FICHA DE CONTROLE DE EMISSÃO	
PROPOSTA AUTOMÓVEIS (CÓPIA)	
PROPOSTA INCÊNDIO (CÓPIA)	
PROPOSTA LUCROS CESSANTES (CÓPIA)	
PROPOSTA RESPONSABILIDADE CIVIL (CÓPIA)	
PROPOSTA RISCOS AERONÁUTICOS (CÓPIA)	
PROPOSTA RISCOS DE ENGENHARIA (CÓPIA)	
PROPOSTA RISCOS DIVERSOS (CÓPIA)	
PROPOSTA ROUBO (CÓPIA)	
PROPOSTA TRANSPORTES (CÓPIA)	

<u>(Continuação)</u>	<u>Fluxo de Dado</u>
Fonte: _____ Descrição: _____	
Destino: _____ Descrição: _____	
Descrição Expandida: _____	

Estruturas de Dados Pertencentes:	Volume de Informação
PROPOSTA TUMULTOS (CÓPIA)	
ANEXO INCÊNDIO	
ANEXO ROUBO-RISCO COMERCIAL/INDUSTRIAL	
ANEXO ROUBO-RISCO RESIDENCIAL	
ANEXO ROUBO-VERANEIO	
RELATÓRIO DE VISTORIA PRÉVIA	

Figura 3.7 - Fluxo de Dado: *Cópia de Proposta e Ficha de Controle*

c) *Depósitos de Dados:* são os locais onde as estruturas de dados são armazenadas até serem solicitadas. Por exemplo, pode-se citar o depósito D2/1 - Pasta para Cópias de Proposta, Figuras 3.8a e 3.8b. As informações contidas no dicionário sobre os depósitos de dados são: nome; referência numérica (consiste de um número atribuído a ele); fluxos de dados de entrada (consiste de todos os fluxos que chegam a ele); fluxos de dados de saída (consiste de todos os fluxos que saem dele); conteúdo (consiste das estruturas que são armazenadas nele) e organização física (consiste do dispositivo físico, caso o sistema seja implementado).

PASTA PARA CÓPIAS DE PROPOSTAS Depósito de Dado-ref: D2/1
Descrição: Contém as cópias de propostas enquanto aguarda chegada da apólice.

Fluxos de Dados de Entrada

2.1-D2/1 CÓPIA DE PROPOSTA E FICHA DE
CONTROLE

2.3-D2/1 RECEBIMENTO DE APÓLICE

Fluxos de Dados de Saída (argumento)

D2/1-2.4 PROPOSTA DE APÓLICE RECEBIDA

Conteúdo:

FICHA DE CONTROLE DE EMISSÃO

PROPOSTA AUTOMÓVEIS (CÓPIA)

PROPOSTA INCÊNDIO (CÓPIA)

PROPOSTA LUCROS CESSANTES (CÓPIA)

PROPOSTA RESPONSABILIDADE CIVIL (CÓPIA)

Organização Física

Figura 3.8a - Depósito de Dado: D2/1 - Pasta para Cópias de Propostas

(Continuação) _____ Depósito de Dado-ref: _____

Descrição: _____

Conteúdo:

Organização Física

PROPOSTA RISCOS AERONÁUTICOS (CÓPIA)

PROPOSTA RISCOS DE ENGENHARIA (CÓPIA)

PROPOSTA RISCOS DIVERSOS (CÓPIA)

PROPOSTA ROUBO (CÓPIA)

PROPOSTA TRANSPORTES (CÓPIA)

PROPOSTA TUMULTOS (CÓPIA)

Continuação

d) *Estruturas de Dados*: são agrupamentos de dados, formados de elementos de dados ou outras estruturas. Por exemplo, pode-se citar as estruturas de dados *Ficha de Controle de Emissão e Proposta Automóveis* que, como se pode notar, Figuras 3.9 e 3.10a, 3.10b, 3.10c e 3.10d, podem ser formadas por outras estruturas de dados, por exemplo *Dados do Veículo*, que é uma informação que ainda pode ser subdividida, ou por elementos de dados, como exemplo, *Segurado*. As informações contidas no dicionário sobre as estruturas são: nome; descrição (consiste de uma explicação resumida sobre o que vem a ser tal estrutura ou então, para o que ela é usada); conteúdo (consiste dos elementos ou outras estruturas de dados que as compõem); estrutura/fluxo de dados relacionados (consiste das estruturas das quais aquela faz parte ou dos fluxos por onde ela passa) e volume de

informação (consiste de uma estimativa de ocorrência da mesma em alguma unidade de tempo).

FICHA DE CONTROLE DE EMISSÃO	
Descrição Controla as datas de emissão da proposta à Seguradora, data de recebimen- to da apólice, etc; é anexada a todas as propostas exceto VG/AP.	Estrutura de Dado
SEGURADO RAMO DATA EMISSÃO À CIS GRP-NRO DATA RECEB RECIBO PROVISÓRIO DATA AVISO AO CORRETOR DATA AVISO DE COBRANÇA DE PRÊMIO	Estrutura/Fluxo de Dado Relacionado 2.1-D2/1, D2/1-2.4, 2.4- D2/3
	Volume de Informação 100/mês

Figura 3.9 - Estrutura de Dado: *Ficha de Controle de Emissão*

PROPOSTA AUTOMÓVEIS	Estrutura de Dado
Descrição Proposta referente ao seguro de automóveis.	
PROPOSTA NRO	Estrutura/Fluxo de Dado
IMPORT SEGURADA CRS (*)	Relacionado
APÓLICE ANTERIOR NRO	2.1-02/1, 2.1-2.2, 2.1-b,
PRÊMIO (*)	02/1-2.4, 2.2-b
C-APÓLICE	
AD-FRAC	
IOF	
PRÊMIO TOTAL CRS	
PROPONENTE	
CPF/CGC	Volume de Informação
ENDEREÇO	30/mês
CEP	
INFORMAÇÕES	
HORA INICIAL	
DATA INICIAL	
HORA FINAL	

Figura 3.10a - Estrutura de Dado: *Proposta Automóveis*

(Continuação)	Estrutura de Dado
Descrição	
DATA FINAL	Estrutura/Fluxo de Dado Relacionado
LOCAL	
DATA	
ASS-PROPONENTE	
NOME CORRETOR	
ASS-CORRETOR	
REG SUSEP NRO	
INSPETORIA	
BANCO P/ COBRANÇA	
AGÊNCIA	Volume de Informação
END-SEGURADO P/ COBRANÇA	
CEP	
PROPONENTE	
ENDEREÇO	
NA QUALIDADE DE	
PROFISSÃO	

Figura 3.10b - Estrutura de Dado: *Continuação*

(Continuação)	Estrutura de Dado
Descrição	
DADOS DO VEÍCULO	Estrutura/Fluxo de Dado Relacionado
MARCA/ANO FABR	
TIPO-VEÍCULO	
CAPACIDADE	
NRO LICENÇA	
NRO MOTOR	
NRO CHASSI	
COR	
IMPORTANCIA SEGURADA	
CASCO	Volume de Informação
RESP CIVIL	
DP	
DM	
ACESSÓRIOS	
FRANQUIAS	
BÁSICA	

Figura 3.10c - Estrutura de Dado: *Continuação*

SEGURO

Elemento de Dado

Descrição

Nome do cliente.

Tipo A AN N

Sinônimos

Discreto		Contínuo
Valor	Significado	Intervalo _____ de valores _____
		Comprimento 50 caracteres
		Representação _____
		Interna _____
Mais de 6 valores, continue no verso		

Elementos/Estruturas de dados relacionados ANEXO INCÊNDIO, ANEXO ROUBO-RISCO COMERCIAL/INDUSTRIAL, ANEXO ROUBO-RISCO RESIDENCIAL, ANEXO ROUBO-VERANEIO, CERTIFICADO INDIVIDUAL, RELAÇÃO X, FICHA VG/AP, FICHA DE CONTROLE DE EMISSÃO, CARNE, PARTE CARNE BANCO, CARNE VG/AP, PARTES VG/AP CORRETORA-CLIENTE, PARTE VG/AP BANCO, FATURA, RELAÇÃO PESSOAS SEGURADAS, REGISTRO DE PRODUÇÃO, DA PROPOSTA, CONTRATO DE FINANCIAMENTO, COBRANÇA, RELAÇÃO PESSOAS COBRANÇA, LINHA2 COBRANÇA.

Figura 3.11 - Elemento de Dado: *Segurado*

PROPOSTA NRO

Elemento de Dado

Descrição

Número da proposta do seguro.

Tipo A AN N

Sinônimos

Discreto		Contínuo
Valor	Significado	Intervalo _____ de valores _____
		Comprimento 15 caracteres
		Representação _____
		Interna _____
Mais de 6 valores, continue no verso		

Elementos/Estruturas de dados relacionados PROPOSTA AUTOMÓVEIS, PROPOSTA INCÊNDIO, PROPOSTA RISCOS AERONÁUTICOS, PROPOSTA RISCOS DE ENGENHARIA, PROPOSTA RISCOS DIVERSOS, PROPOSTA ROUBO, PROPOSTA TRANSPORTES, PROPOSTA TUMULTOS, REGISTRO DE PRODUÇÃO, DA PROPOSTA.

Figura 3.12 - Elemento de Dado: *Proposta Nro*

3.5 - CONCLUSÕES

Pode-se dizer, que essa metodologia recomendada por Gane [Ga77], e que foi utilizada na descrição desse sistema de Corretagem de Seguros, traz grandes facilidades de comunicação entre analista e usuário, como já foi enfatizado no item 3.2, mas que é um ponto muito importante durante a fase de análise. Através da própria metodologia, encontrou-se a linguagem de comunicação entre ambas as partes, o que facilitou muito o trabalho e minimizou o problema de incompatibilidade, entre aquilo que o usuário quer dizer e aquilo que é entendido pelo analista, fazendo com que a descrição final do sistema se aproxime bastante da realidade.

As dificuldades sentidas na utilização dessa metodologia foram: primeiro, a percepção de que a própria elaboração do diagrama de fluxo de dados era o meio de comunicação entre analista e usuário. Isso aconteceu, talvez, devido ao fato de que o autor, em [Ga77], descreveu o seu sistema, primeiro de uma forma narrativa, para depois então, construir os diagramas, mas, naturalmente ele fez isso com a intenção de que o leitor conhecesse o sistema que seria abordado durante o texto. Assim, a analista sentiu-se induzida por essa idéia, rapidamente descartada, pois não foi difícil perceber que os diagramas deveriam ser elaborados à medida em que as informações fossem transmitidas. A segunda dificuldade encontrada refere-se à denominação dos fluxos de dados, quando vários deles saem do mesmo processo, mas não necessariamente, todos eles ocorrem, quando o processo é executado. Isto foi solucionado dando-se, a cada fluxo, um adjetivo que identificasse a ocorrência do mesmo. Finalmente, a outra dificuldade foi definir se o grau de detalhamento dos diagramas era suficiente ou não. Isso também foi solucionado da seguinte maneira: à medida em que se elaborava os diagramas de fluxo de dados, fazia-se um rascunho, bem a grosso modo, da lôgica dos processos; então, se esta fosse composta de muitos passos, significaria que o processo deveria sofrer explosão num diagrama detalhado de fluxo de dados.

A menos dessas dificuldades, a metodologia é bastante simples, traz a grande vantagem da facilidade de comunicação

entre analista e usuário, e também torna o sistema amplamente documentado através do dicionário de dados, que é construído com base nos diagramas. Devido a este último fato, pode-se dizer que, embora o trabalho tenha sido interrompido, para a Corretora, na fase de análise, ainda assim, ele trouxe suas contribuições pois, pelos diagramas de fluxo de dados mais o dicionário, as atividades da Corretora estão documentadas de tal forma que ela pode utilizar-se deste trabalho para a instrução de novos funcionários. Isto porque, pelos diagramas, pode-se conhecer todas as atividades desempenhadas pela Corretora e, pelo dicionário de dados, tem-se uma descrição detalhada de todas elas e também de todos os documentos utilizados, para que servem, como são preenchidos, o que significa cada informação dos mesmos, etc., o que faz com que a adaptação de um novo funcionário, com toda essa documentação, torne-se mais simples.

A documentação refere-se à situação existente e, com base nela e entendimentos com o usuário foi possível determinar suas necessidades mais prementes com relação ao sistema, concluindo-se que eram elas: controle de vencimento de carnês de financiamento e notas de cobrança bancária, de expiração de seguros, da produção da Corretora, do cadastramento de clientes e dos capitais segurados em VG/AP.

C A P Í T U L O 4

SOLUÇÃO DO CASO PRÁTICO ABORDADO

4.1 - CONSIDERAÇÕES GERAIS

Este capítulo refere-se à documentação sobre a implementação de algumas das atividades da Corretora de Seguros que foi tratada com detalhes em [Fa84] e, cuja análise estruturada está descrita no capítulo 3. Em [Fa84], colocou-se o problema de que, inicialmente, o proprietário dessa Corretora pretendia adquirir algum equipamento para apoiar por computador algumas das atividades desempenhadas pelo seu pessoal, otimizando assim, o serviço. No entanto, como a compra do equipamento não foi realizada, como foi dito no item 3.5, com relação à Corretora de Seguros, o trabalho foi interrompido na fase de análise. Portanto, este capítulo corresponde, justamente, à continuidade ao trabalho de análise, isto é, à implementação daquelas funções que o proprietário da Corretora desejava apoiar por computador.

Deve-se observar que grande parte da implementação corresponde, principalmente, à manipulação de depósitos de dados (arquivos). Dentre estes, muitos não puderam ser implementados por tratarem de documentos que têm validade por si mesmos, como é o caso do arquivo de apólices, pois a existência de qualquer tipo de seguro, só é comprovada mediante apresentação da mesma. E, assim como este, existem vários outros casos, que também não justificavam a colocação em arquivos eletrônicos.

Com relação ao equipamento, considerando-se que a Corretora em questão possuía as características descritas em [Fa84] e que por isso, podia ser considerada como um escritório de pequeno a médio porte, o equipamento, utilizado na implementação, deveria ser compatível com essa descrição, isto é, não muito caro, com facilidade de assistência técnica, mas que, por outro lado, atendesse ao processamento requerido, que consistia

basicamente de manipulação de arquivos, como foi dito anteriormente.

De acordo com essas considerações e também, pelo fato de que a autora tinha possibilidade de trabalhar com um microcomputador Exato Pro, da CCE, linha Apple, de 48k de memória e placas de expansão de 16k, 80 colunas e CP/M, e dois acionadores de disco de 5 1/4", fez-se a implementação nesse equipamento pois, além de ser acessível à Corretora (embora, na realidade a configuração devesse sofrer algumas alterações), possibilitava também o uso do software de gestão de base de dados, dBase II, mencionado no Capítulo 2. Esse software era uma boa opção para resolver os problemas da Corretora e também, facilitava o desenvolvimento do sistema voltado ao usuário, permitindo que um leigo pudesse operá-lo. Bowerman, em [Bo83], refere-se à utilização desse software, dizendo que, com ele pode-se resolver, com maior facilidade, as necessidades dos usuários.

Assim, considerando-se todos esses aspectos, é que o sistema foi desenvolvido e uma avaliação desse desenvolvimento é feita no Capítulo 5.

Este capítulo tem por finalidade, apresentar uma documentação da implementação realizada, da seguinte maneira:

No item 4.2, apresenta-se o esquema da base de dados, relativo às funções que puderam ser implementadas, e que eram de interesse do proprietário da Corretora.

No item 4.3, apresenta-se a estrutura modular dos programas, em nível mais geral de detalhamento.

No item 4.4, apresenta-se as diretrizes adotadas para detalhamento do diagrama de estrutura modular, apresentado no item anterior.

No Apêndice, apresenta-se os diagramas detalhados de estrutura modular dos módulos implementados, com as respectivas listagens dos programas.

No item 4.5, apresenta-se exemplos do funcionamento do sistema e, finalmente, no item 4.6, apresenta-se algumas conclusões referentes à implementação realizada, para o caso prático em questão.

4.2 - ESQUEMA DA BASE DE DADOS

Como foi dito no item 4.1, a implementação desse sistema refere-se à manipulação de arquivos, principalmente, daqueles que tratam de cadastramento e controle de vencimentos dos seguros e dos pagamentos. Muitos dos depósitos de dados, representados nos diagramas de fluxo de dados, Figuras 3.1 a 3.5, não permitem implementação, por tratar-se de arquivos de documentos que não podem ser transformados num arquivo eletrônico pois, apenas mediante a apresentação dos mesmos, é que se pode comprovar a realização do seguro. Como por exemplo, pode-se citar os depósitos de dados:

D2/1 - Pasta para Cópias de Propostas;

D2/3 - Arquivo de Cópias de Apólices;

D3/5 - Arquivo de Carnês de VG/AP;

D5 - Documentos à Contabilidade, e outros.

Pode-se representar o conjunto de informações, que foram implementadas em computador, através do seguinte esquema da base de dados no mundo real:

onde os relacionamentos têm o seguinte significado:

- 1- um determinado seguro pertence a um cliente.
- 2- um cliente pode possuir vários seguros.
- 3- um determinado carnê de financiamento pertence a um cliente.
- 4- um cliente pode possuir vários carnês de financiamento.
- 5- uma determinada nota de cobrança bancária pertence a um cliente.
- 6- um cliente pode possuir várias notas de cobrança bancária.
- 7- o livro de registro de produção relaciona todos os seguros.
- 8- cada seguro está relacionado no livro de registro de produção.
- 9- cada visita é feita a um cliente.
- 10- um cliente recebe várias visitas.

Observa-se que os relacionamentos 1-2, 3-4 e 5-6 possuem, associados a eles, atributos que são respectivamente:

- a) ramo, pois um cliente pode possuir vários seguros de ramos diferentes;
- b) dia de vencimento de carnê, pois para cada par carnê-cliente, associa-se um dia de vencimento para pagamento;
- c) data de vencimento de nota, pois para cada par nota de cobrança bancária-cliente, associa-se uma data de pagamento.

Dessa forma, os depósitos de dados que foram implementados em computador, foram os seguintes:

D2 - Fichas de Clientes: consistia de um arquivo de fichas de cadastramento, que eram preenchidas para todos os clientes, independente do tipo de seguro.

D2/5- Livro de Registro de Produção: consistia de um caderno onde eram marcados alguns dados sobre todas as propostas e apólices de séguros. Esses dados serviam para se fazer uma comparação com a estimativa de seguros efetuados, fei

ta para aquele ano, e para se fazer a estima
tiva para o ano seguinte.

D2/7- *Documentos em Cobrança Bancária*: consistia de um arquivo onde eram guardadas todas as notas de cobrança bancária referentes aos seguros, cujos pagamentos eram feitos em 3 parcelas. Como, para efetuar o pagamento, a própria nota tinha que ser apresentada, na implementação, esse arquivo contém apenas os dados sobre o vencimento delas, de forma que, as outras informações não foram armazenadas.

D2/8- *Agenda de Controle de Vencimentos*: consistia de uma agenda comum na qual eram marcados, na respectiva data, os carnês que venciam nessa data, para que a corretora efetuasse o pagamento. Na implementação, o arquivo correspondente contém o dia do vencimento da parcela, a data de vencimento total referente à última parcela de pagamento e o número de inscrição do cliente.

D3/1- *Pasta de Clientes*: consistia de um arquivo de cartões consignaçaõ que eram preenchidos para cada funcionário de uma firma, quando esta fizesse um seguro de vida em grupo e acidentes pessoais coletivos. Esse cartão demonstrava a aceitação do funcionário sobre esse seguro e autorizava ao estipulante o desconto em folha.

D3/3- *Arquivo de Relação X*: consistia de um caderno, cuja finalidade era marcar o mês referente ao pagamento de cada segurado de Vida em Grupo e Acidentes Pessoais.

D3/4- *Segurados VG/AP*: consistia de um arquivo de fichas de cadastramento que eram preenchidas apenas para os segurados de VG/AP e que con

tinham as especificações das quantias segura
das em cada uma das modalidades que esse tipo
de seguro oferecia.

Na implementação, cada um desses depósitos de dados
corresponde, praticamente, a um arquivo de dados, com exceção
do depósito de dados *D2- Fichas de Clientes*, que deu origem a
mais de um arquivo de dados pelo fato de possuir, além dos da
dos cadastrais, uma relação de todos os seguros que o cliente
possuía e uma relação de resultados de visitas ao cliente. Por
tanto, os arquivos de dados, usados na implementação, foram ob-
tidos dos depósitos de dados, da seguinte maneira:

De *D2- Fichas de Clientes* obteve-se:

- FICHACLI.DBF - que contém os dados cadastrais do cliente;
- SEGCLI.DBF - que contém a relação segurado/seguro e
- RESVISIT.DBF - que contém resultados de visitas a clientes.

De *D2/5- Livro de Registro de Produção* obteve-se:

- LVRGPROD.DBF - que contém as informações sobre todos os segu
ros efetuados.

De *D2/7- Documentos em Cobrança Bancária* obteve-se:

- DOCBANC.DBF - que contém informação sobre a data de pagamento
de nota.

De *D2/8- Agenda de Controle de Vencimentos* obteve-
-se:

- AGENDA.DBF - que contém informação sobre a data de pagamento
de parcelas do carnê de financiamento.

De *D3/1- Pasta de Clientes* obteve-se:

- PASTACLI.DBF - que contém os dados sobre um segurado de vida
em grupo e acidentes pessoais, referentes ao
cartão consignação.

De *D3/3- Arquivo de Relação X* obteve-se:

- ARQRELX.DBF - que contém informação sobre pagamento de clien
te VG/AP.

De D3/4 - Segurados VG/AP obteve-se:

- SEGVGAP.DBF - que contém informações sobre o seguro VG/AP de um cliente, especificando as quantias seguradas.

Além desses arquivos de dados, foram criados outros dois, para facilidade de processamento e também, para otimizar o espaço em disco. São eles: RAMOSEG.DBF - que contém os vários ramos de seguros e um código associado a cada um deles e TABRESUL.DBF - que contém os possíveis resultados de visitas a clientes e também, um código associado a cada um. Dessa forma, quando nos arquivos acima citados foi necessário fazer referência a um ramo de seguro ou resultado de visita, substitui-se essa informação pelo código associado.

De acordo com o esquema da base de dados no mundo real, pode-se dizer que cada instância da entidade *Cliente* sempre é representada no arquivo FICHACLI.DBF, pois qualquer cliente, independente do tipo de seguro que ele faça é representado, nesse arquivo, pelos seus dados cadastrais. Já a entidade *Seguro*, não é especificamente representada por um arquivo, pois isto é feito através do documento "apólice". No entanto, o relacionamento 1-2 entre essas duas entidades provoca uma instância no arquivo SEGCLI.DBF, que relaciona os seguros que o cliente possui e, se o seguro efetuado for de vida em grupo e acidentes pessoais, provoca também instâncias nos arquivos PASTACLI.DBF, SEGVGAP.DBF e ARQRELX.DBF.

A entidade *Carnê de Financiamento*, também não é representada especificamente por um arquivo, pois o documento "carnê" não pode deixar de existir fisicamente. Portanto, o que é representado na base de dados é o relacionamento 3-4, com a finalidade de controlar o dia de vencimento das parcelas de pagamento de um determinado seguro de um cliente, através do arquivo AGENDA.DBF. Pelo mesmo motivo, a entidade *Nota de Cobrança Bancária* não é representada especificamente, mas sim, o relacionamento 5-6, para controlar as datas de pagamento, através do arquivo DOCBANC.DBF.

A entidade *Produção* é representada pelo arquivo LVRGPRGD.DBF e o relacionamento 7-8, por uma instância nesse arquivo, referente ao registro de um seguro que foi efetuado,

que não seja do tipo VG/AP.

Portanto, a implementação realizada é composta de operações de inserção, emissão de relatórios e delição nos arquivos do dBase que contêm as informações correspondentes a esse esquema de base de dados no mundo real.

4.3 - ESTRUTURA MODULAR DOS PROGRAMAS

Como foi dito no item 4.1, o sistema desenvolvido utilizou o software de gestão de base de dados dBaseII. Foi elaborado numa forma de encadeamento de cardápios para facilitar a sua utilização, principalmente para as pessoas que não têm contato com computação. Assim, tudo o que uma pessoa precisa saber para operá-lo, é colocar o disquete, chamar o dBaseII e chamar o programa principal, que contém o cardápio das operações básicas e que, de acordo com a opção do usuário, se encarrega de ativar os outros programas.

Dessa forma, existe um programa que envia um cardápio principal, Figura 4.1, composto das operações de inserção, emissão de relatórios e deleção. Escolhida uma, dentre essas três operações, entra-se num cardápio mais detalhado, Figuras 4.2, 4.3 e 4.4, referente à operação escolhida anteriormente. Nesse ponto, escolhe-se então, uma operação específica que, após ser executada, dependendo do caso, ou volta ao cardápio anterior automaticamente, ou então, apresenta as possibilidades de se executar a mesma função novamente, voltar ao cardápio anterior, voltar ao cardápio principal, ou terminar o processamento.

MENU DAS OPERACOES BASICAS =====

- 1- INSERCAO
- 2- RELATORIOS PADROES
- 3- DELICAO
- 4- TERMINAR EXECUCAO

Figura 4.1 - Cardápio Principal

MENU DAS OPERACOES DE INSERCAO
=====

INSERCAO DE :

- 1- SEGURADO DE VG/AP
- 2- SEGURADOS DOS OUTROS TIPOS DE SEGUROS
- 3- RESULTADO DE VISITA A CLIENTE
- 4- VENCIMENTOS DE NOTAS DE COBRANCA BANCARIA
- 5- REGISTRO DE PRODUCAO
- 6- VENCIMENTO DE CARNE DE FINANCIAMENTO
- 7- RETORNAR AO MENU PRINCIPAL
- 8- TERMINAR EXECUCAO

Figura 4.2 - Cardápio Detalhado das Operações
de Inserção

MENU DOS RELATORIOS PADROES
=====

- 0- SEGURADOS POR TIPO DE SEGURO
- 1- SEGURADOS E RESPECTIVOS SEGUROS
- 2- NOTAS DE COBR. BANCARIA QUE VENCEM EM DETERMINADA DATA
- 3- SEGUROS QUE EXPIRAM EM DETERMINADO MES
- 4- PROPONENTES DE VG/AP POR ESTIPULANTE
- 5- PRODUCAO ATE O MOMENTO
- 6- CARNES A PAGAR NO DIA
- 7- CAPITAL SEGURADO DE UM CLIENTE VG/AP
- 8- RETORNAR AO MENU PRINCIPAL
- 9- TERMINAR EXECUCAO

Figura 4.3 - Cardápio Detalhado de Relatórios Padrão

MENU DAS OPERACOES DE DELICAO
=====

DELICAO DE :

- 1- NOTAS DE COBR. BANCARIA PAGAS
- 2- REGISTRO DE CARNE DA AGENDA
- 3- RETORNAR AO MENU PRINCIPAL
- 4- TERMINAR EXECUCAO

Figura 4.4 - Cardápio Detalhado das Operações
de Delicção

Portanto, ativando-se o programa inicial, consegue-se executar qualquer tipo de operação, sempre de modo interativo, através das opções oferecidas pelos cardápios, sem sair do controle do próprio sistema, até que isso seja realmente desejado pelo usuário.

A seguir, encontra-se o diagrama de estrutura modular dos programas, Figura 4.5, do nível mais geral, que compoem a implementação elaborada, isto é, dos programas que enviam os vários cardápios com todas as opções oferecidas pelo sistema. Tal diagrama corresponde ao projeto estruturado do sistema, segundo [Yo75].

Figura 4.5 - Estrutura Modular dos Programas - Nível Geral

4.4 - DETALHAMENTO DOS MÓDULOS

Como foi visto, o item 4.3 mostrou apenas o diagrama de estrutura modular dos programas mais gerais do sistema, Figura 4.5, referentes aos cardápios. Esse diagrama e também todos os outros correspondentes aos vários programas que compõem o sistema, encontram-se detalhados no Apêndice, onde se pode notar que os diagramas são compostos de dois tipos de "caixas": umas simples, e outras com as laterais duplas. As primeiras correspondem a instruções do corpo do respectivo programa e as segundas, a módulos chamados pelo programa em questão, isto é, a módulos distintos, que executam funções bem definidas no sistema, correspondendo cada um deles a um arquivo de comando (de extensão .CMD) do dBaseII.

Deve-se salientar que, utilizando-se o dBaseII para desenvolvimento de um sistema, sempre ocorrerá a situação acima citada, ou seja, a estrutura modular de um programa sempre será composta desses dois tipos de "caixas", pois esse software não permite que um módulo faça parte do corpo de um outro módulo como pode ocorrer, por exemplo, em COBOL, com os parágrafos que são chamados através do PERFORM, ou do PASCAL, através das PROCEDURES. Ao contrário, em dBaseII, para se fazer com que cada um dos módulos represente uma única função, estes são fisicamente distintos, mesmo que, um módulo B seja referenciado por apenas um outro módulo A do sistema.

Cada uma das caixas de laterais duplas dos diagramas de estrutura modular, apresentados no Apêndice, foi detalhada através de outro diagrama. Alguns desses diagramas detalhados, por sua vez, também possuem caixas de laterais duplas que também se encontram detalhadas em outro diagrama. Todos esses diagramas detalhados juntamente com as respectivas listagens dos programas encontram-se no Apêndice, dispostos da seguinte maneira: quando um módulo B está referenciado apenas no diagrama de detalhamento de um módulo A, o detalhamento do módulo B segue imediatamente após o do A; quando um módulo D está referenciado no diagrama de vários outros módulos, o detalhamento de D segue após os detalhamentos de todos os módulos que fazem referência a ele.

4.5 - ALGUNS EXEMPLOS DO FUNCIONAMENTO DO SISTEMA

Como foi dito no item anterior, o sistema foi elaborado através de um encadeamento de cardápios. Suas operações básicas são operações de inserção, de emissão de relatórios padrão e de delição, como mostrou a Figura 4.1, correspondente ao cardápio principal. Cada uma dessas operações possui um cardápio detalhado, como foi mostrado pelas Figuras 4.2, 4.3 e 4.4 respectivamente, que permite a seleção de uma operação específica dentro da operação básica escolhida no cardápio principal.

Com base nessas operações básicas, este item tem a finalidade de mostrar alguns exemplos de funcionamento das mesmas.

a) *Operações de Inserção*

Como pode-se observar pela Figura 4.2, existem várias operações de inserção que podem ser realizadas. No entanto, como tais operações correspondem à inserção de dados necessários para realizá-las, através de telas formatadas, não se consegue mostrá-las integralmente pois, na tela, são apresentados todos os campos a serem preenchidos, e o cursor volta a posicionar-se no primeiro campo, aguardando o fornecimento dos dados, o que não é possível ser feito com a impressora.

Dessa forma, embora todas as operações tenham sido implementadas, como pode-se constatar no Apêndice, aqui apresenta-se apenas uma delas.

a.1) *Inserção de: 2 - Segurados dos Outros Tipos de Seguros*

Primeiramente, é apresentada uma mensagem ao usuário e, logo em seguida, o último número de inscrição de cliente cadastrado (o qual foi criado para facilidade de processamento, e cujo número novo é igual ao que foi apresentado, acrescido de "1"), como mostra a Figura 4.6, a seguir.

NOTA :

A INSERCAO DE UM SEGURADO DE QUALQUER TIPO DE SEGURO, EXCETO VG/AP, CORRESPONDE A ATUALIZACAO DE ARQUIVOS DESSA BASE DE DADOS, REFERENTES A CADASTRAMENTO, A SABER : FICHA DE CLIENTE E RELACAO DE SEGUROS QUE O CLIENTE POSSUI.

O NUMERO DE INSCRICAO DO SEGURADO E FORMADO DA SEGUINTE

MANEIRA : . AANNNNN

ONDE : AA - CORRESPONDE AO ANO
 NNNNN - NUMERO SEQUENCIAL

O ULTIMO NUMERO DE INSCRICAO E : 8500022

Figura 4.6

Em seguida, é enviada uma tela solicitando dados cadastrais; operação esta, que não permite ser apresentada.

Logo após, é apresentada uma mensagem ao usuário, Figura 4.7, alertando-o que serão apresentados códigos associados a cada tipo de seguro, Figura 4.8, pois o usuário deverá fornecer essa informação ao sistema, para que este associe ao cliente, o tipo de seguro que ele efetuou.

ATENCAO!...

AGORA SERAO APRESENTADOS OS VARIOS TIPOS DE SEGUROS COM SEUS RESPECTIVOS CODIGOS. PARA EFEITO DE PROCESSAMENTO, VOCE DEVERA ANOTAR O(S) CODIGO(S) ASSOCIADO(S) AO(S) SEGURO(S) :

A- QUE O CLIENTE EFETUOU (NO CASO DE OPERACAO DE INSERCAO)

B- DOS QUAIS DESEJA-SE RELACAO DE SEGURADOS (CASO DE RELATORIOS)

PARA QUE VOCE POSSA PREENCHER CORRETAMENTE AS INFORMACOES QUE LHE SERAO SOLICITADAS.

Figura 4.7

CODIGO =====	TIPO DE SEGURO =====
1	AUTOMOVEIS
2	INCENDIO
3	LUCROS CESSANTES
4	RESPONSABILIDADE CIVIL
5	RISCOS AERONAUTICOS
6	RISCOS DE ENGENHARIA
7	RISCOS DIVERSOS
8	ROUBO
9	TRANSPORTES
10	TUMULTOS
11	VIDA EM GRUPO E ACIDENTES PESSOAIS

Figura 4.8

Para finalizar, é apresentado um conjunto de opções que o usuário pode fazer nesse momento, Figura 4.9, que são: repetir a mesma operação de inserção que ele acabou de efetuar; retornar ao cardápio anterior, que corresponde àquele das operações de inserção; retornar ao cardápio principal, o qual apresenta apenas as operações básicas, ou então, terminar a execução que corresponde a sair desse sistema e voltar ao dBaseII.

ESCOLHA SUA OPCAO :

- 1- REPETIR ESTA OPERACAO
- 2- RETORNAR AO MENU ANTERIOR
- 3- RETORNAR AO MENU PRINCIPAL
- 4- TERMINAR EXECUCAO

Figura 4.9

b) Relatórios Padrão

Como pode-se observar pela Figura 4.3, foram desenvolvidos vários tipos de relatórios que o usuário pode solicitar ao sistema. Para exemplificá-los, são apresentados em seguida.

b.1) Relatório de: $\emptyset$ - Segurados por Tipo de Seguro

Primeiramente, é apresentada ao usuário, a opção de solicitar o relatório para todos os tipos de seguros ou para um tipo específico (que é o caso), como mostra a Figura 4.10.

ESTE PROGRAMA EMITE UM RELATORIO DE SEGURADOS POR TIPO DE SEGURO, COM AS SEGUINTE ALTERNATIVAS :

- 1- SEGURADOS PARA CADA TIPO DE SEGURO EXISTENTE
- 2- SEGURADOS PARA UM TIPO ESPECIFICO DE SEGURO

Figura 4.10

Em seguida, se o usuário escolheu a opção "2", é enviada uma mensagem, Figura 4.7, alertando-o que serão enviados códigos associados a cada tipo de seguro, Figura 4.8, pois através do código, o usuário irá selecionar o tipo de seguro do qual ele deseja o relatório.

Em seguida, é apresentado o relatório solicitado, como mostra a Figura 4.11, a seguir.

RELATORIO DE SEGURADOS POR RAMO DE SEGURO

25/11/85

RAMO : AUTOMOVEIS

SEGURADO	INSCRICAO
AMADEU DOARTE	8500006
ANA MARIA DAS DORES	8500007
MARINA DIAS	8500008
EVANDRO EDUARDO	8500009
EUGENIO MARTINS	8500011
CONSTANCIO VERISSIMO	8500012
NESTOR FRAGAS DANTAS	8500013
SELMA CLEIA ANDRADE	8500015
ANTONIO BATISTA	8500016
PATRICIA MOREIRA LIMA	8500017
CLODOALDO RAMOS MARQUES	8500018
PLAUTO JUNQUEIRA	8500019
JULIO MESQUITA GOES	8500020
SEBASTIAO GENEROSO	8500021
FRANCISCO CARLOS GENEROSO	8500022

Figura 4.11

Finalmente, como o usuário pode desejar o mesmo relatório para um outro tipo de seguro, são apresentadas as opções da Figura 4.9.

b.2) Relatório de: 2 - *Notas de Cobr. Bancária que Vencem em Determinada Data*

Primeiramente, é enviada uma mensagem ao usuário, Figura 4.12, e logo em seguida, é solicitada a data a qual, da mesma forma que as operações de inserção ou as opções feitas nos relatórios anteriores, não é possível ser apresentada.

ESTE PROGRAMA EMITE UM RELATORIO DAS NOTAS EM COBRANCA BANCARIA
QUE VENCEM NA DATA QUE SERA SOLICITADA A SEGUIR.

Figura 4.12

Em seguida, é apresentado o relatório, sendo que ne
le consta a data que foi fornecida anteriormente, como mostra a
Figura 4.13.

RELATORIO DE NOTAS EM COBRANCA BANCARIA QUE VENCEM EM :

DATA :10/11/85

REFERENTES AOS SEGUINTE CLIENTES :

<u>SEGURADO</u>	<u>INSCRICAO</u>
EVANDRO EDUARDO	8500009
JOAO MARIO PEREIRA	8500014
CLODOALDO RAMOS MARQUES	8500018

Figura 4.13

b.3) Relatório de: 3 - Seguros que Expiram em Determinado
Mês

Primeiramente, é enviada uma mensagem explicando o
objetivo do relatório, Figura 4.14, e em seguida, embora não se
possa mostrar, é solicitado o mês do qual deseja-se o relatório.

ESTE PROGRAMA EMITE UM RELATORIO DOS SEGUROS QUE EXPIRAM NUM
DETERMINADO MES. ELE DEVE SER SOLICITADO NOS MEADOS DE UM MES,
COM REFERENCIA AO MES SEGUINTE, PARA QUE OS CLIENTES POSSAM
SER COMUNICADOS E TENHAM TEMPO PARA FAZER A RENOVACAO.

Figura 4.14

Em seguida, é apresentado o relatório, como mostra a Figura 4.15.

RELATORIO DOS SEGUROS QUE EXPIRAM NO MES : 12				25/11/85
-----				==
REFERENTES AOS SEGUINTE CLIENTES :				
INSCRICAO	SEGURADO	RAMO DO SEGURO	DIA VENCTO	
-----	-----	-----	-----	
8500006	AMADEU DOARTE	VIDA EM GRUPO E ACIDENTES PESSOAIS	12	
8500007	ANA MARIA DAS DORES	AUTOMOVEIS	24	
8500007	ANA MARIA DAS DORES	INCENDIO	12	
8500007	ANA MARIA DAS DORES	LUCROS CESSANTES	31	
8500012	CONSTANCIO VERISSIMO	LUCROS CESSANTES	10	
8500012	CONSTANCIO VERISSIMO	ROUBO	20	

Figura 4.15

b.4) Relatório de: 5 - *Produção até o Momento*

Primeiramente, é enviada uma mensagem explicando o objetivo do relatório, como mostra a Figura 4.16

ESTE PROGRAMA EMITE UM RELATORIO COM AS PORCENTAGENS DE PRODUCAO, POR RAMO DE SEGURO, ATE O MOMENTO

Figura 4.16

Em seguida, é apresentado o relatório, como mostra a Figura 4.17.

RELATORIO DOS PERCENTUAIS DE PRODUCAO ATE O MOMENTO - ANO REF : 85 25/11/85

TOTAL DE SEGUROS EFETUADOS =	34
RAMO DO SEGURO	PORCENTAGEM
AUTOMOVEIS	38,2 %
INCENDIO	17,6 %
LUCROS CESSANTES	11,7 %
RESPONSABILIDADE CIVIL	0 %
RISCOS AERONAUTICOS	0 %
RISCOS DE ENGENHARIA	2,9 %
RISCOS DIVERSOS	0 %
ROUBO	23,5 %
TRANSPORTES	5,8 %
TUMULTOS	0 %

Figura 4.17

b.5) Relatório de: 7 - *Capital Segurado de um Cliente VG/AP*

Primeiramente, é apresentada ao usuário, a opção de solicitar o relatório para todos os segurados ou para um seguro do específico (que é o caso), como mostra a Figura 4.18.

ESTE PROGRAMA EMITE UM RELATORIO DE SEGURADOS DE VIDA EM GRUPO E ACIDENTES PESSOAIS, COM OS CAPITAIS REFERENTES A CADA UMA DAS MODALIDADES DE SEGUROS PERMITIDAS POR ESSE RAMO. EXISTEM AS SEGUINTE ALTERNATIVAS :

- 1- RELACAO DE CAPITAIS DE TODOS OS SEGURADOS DE VG/AP
- 2- RELACAO DE CAPITAIS DE UM SEGURADO ESPECIFICO DE VG/AP

Figura 4.18

Em seguida, embora não se possa apresentar, é soli
citado o nome do cliente e então, é enviado o relatório, como
mostra a Figura 4.19.

```
RELATORIO DE CAPITAIS SEGURADOS EM VG/AP                                25/11/85
-----

PRINCIPAL : ANTONIO JOAO                                              INSCR : 8500002
-----

MORTE           : CR$  30,000,000
INV.PERMANENTE  : CR$  15,000,000
ASSIST.MED.DESP.SUPL. : CR$   7,500,000
DESP.HOSPITALARES : CR$   100,000

CONJUGE : MARTHA JOAO
-----

MORTE           : CR$  15,000,000
INV.PERMANENTE  : CR$   7,500,000
ASSIST.MED.DESP.SUPL. : CR$  3,750,000
DESP.HOSPITALARES : CR$   50,000
```

Figura 4.19

Como o usuário pode solicitar o mesmo relatório pa
ra outro cliente, é apresentado o conjunto de opções da Figura
4.9.

Supondo-se que a opção "1" fosse escolhida, seria so
licitado outro nome de cliente e um relatório correspondente se
ria enviado, como mostra a Figura 4.20.

```
RELATORIO DE CAPITAIS SEGURADOS EM VG/AP                                25/11/85
-----

PRINCIPAL : IRINEU DA SILVA                                          INSCR : 8500005
-----

MORTE           : CR$  100,000,000
INV.PERMANENTE  : CR$  100,000,000
ASSIST.MED.DESP.SUPL. : CR$   50,000,000
DESP.HOSPITALARES : CR$   1,000,000

NAO EXISTEM CAPITAIS SEGURADOS PARA CONJUGE
```

Figura 4.20

Em seguida, seria apresentado novamente o conjunto de opções da Figura 4.9, e assim como nos outros relatórios on de isso acontece, esse processo é iterativo até que o usuário escolha uma opção diferente de "1".

c) Operações de Delição

Foram desenvolvidas duas operações de delição, as quais fazem uma atualização nos arquivos que controlam as datas de vencimento de carnês ou de notas de cobrança bancária, para eliminar dos mesmos, os documentos quitados.

c.1) Delição de: 1 - Notas de Cobr. Bancária Pagas

Primeiramente, é enviada uma mensagem explicando o objetivo do progrma, como mostra a Figura 4.21.

ESTE PROGRAMA FAZ UMA DELICAO NO ARQUIVO QUE CONTROLA O
VENCIMENTO DE NOTAS EM COBRANCA BANCARIA,REFERENTE A NOTA
QUE JA FOI PAGA.

Figura 4.21

Em seguida, embora não se possa mostrar, são solici-
tados o nome do cliente e a data de vencimento da nota e, logo
após, são enviados os dados referentes ao registro que será eli-
minado do arquivo, como mostra a Figura 4.22.

SERA ELIMINADO DO ARQUIVO O REGISTRO REFERENTE AO

CLIENTE : JOAO MARIO PEREIRA

INSCR : 8500014

DATA DE EMISSAO DA NOTA : 261085

DATA DE VENCIMENTO DA NOTA : 101185

Figura 4.22

Em seguida, o sistema pede uma confirmação, que é dada pelo usuário e finalmente, é apresentado o conjunto de opções mostrado na Figura 4.9, pois o usuário pode desejar eliminar do arquivo dados referentes a outra nota quitada.

c.2) Delibação de: 2 - *Registro de Carnê da Agenda*

Primeiramente, é enviada uma mensagem explicando o objetivo do programa, como mostra a Figura 4.23 e, em seguida, embora não se possa mostrar, o sistema solicita o mês com o qual será feito o processamento.

ESTE PROGRAMA FAZ A DELICAO NO ARQUIVO QUE CONTROLA OS VENCIMENTOS DOS CARNES DE FINANCIAMENTO. COMO ESSES PAGAMENTOS NAO ATRASAM POIS ESTAO SOB RESPONSABILIDADE DA CORRETORA, TODO INICIO DE MES PODE-SE ELIMINAR DO ARQUIVO OS CARNES QUE ENCERRARAM NO MES ANTERIOR.

Figura 4.23

Em seguida, é apresentada uma relação dos registros que serão eliminados, Figura 4.24, pois correspondem a carnês que já foram quitados no mês que foi fornecido como entrada.

RELACAO DE CARNES QUITADOS NO MES : 11		25/11/85
SEGURADO	INSCRICAO	DATA VENCTO
ANTONIO BATISTA	8500016	9/11/85

Figura 4.24

4.6 - CONCLUSÕES

Através de [Yo75], sabe-se que a melhor solução para projetar um sistema é alcançada com a utilização do projeto estruturado. Dessa forma, como se pode observar pelos itens anteriores e pelo Apêndice, procurou-se elaborar o sistema através de módulos pequenos, o mais independente possível um do outro, cada qual executando uma tarefa específica, isto é, procurando fazer com que os módulos possuíssem coesão funcional, segundo [Ga77].

Um projeto altamente modular de um sistema facilita:

- a utilização, permitindo que o usuário execute tarefas específicas a cada momento;

- a implementação, permitindo que cada módulo do sistema seja testado separadamente, para depois, ser inserido no contexto geral e

- a manutenção, pois se uma função vier a sofrer alterações, como normalmente ela equivale a um módulo dentro do sistema, apenas esse módulo deverá ser alterado.

No caso em questão, a modularização do sistema facilitou o desenvolvimento dos cardápios que auxiliam bastante os usuários finais.

Quanto ao software utilizado, o dBaseII, pode-se dizer que a principal dificuldade sentida foi quando da necessidade de se utilizar três arquivos simultaneamente no mesmo programa. Nesses casos, deixava-se um arquivo de dados como primário (PRIMARY) e selecionava-se os outros dois como secundários (SECONDARY), um por vez, de acordo com a necessidade. A dificuldade que isso trouxe foi que, na maioria das vezes, precisava-se continuar o processamento do ponto onde se havia parado e isso não era possível, pois selecionar um segundo arquivo como secundário implica em fechar aquele que estava em uso e abrir o outro; sendo assim, no momento de se selecionar novamente o primeiro arquivo, o ponto de partida era o primeiro registro e não mais aquela posição referente ao momento em que a seleção do arquivo secundário fora alterada. Para solucionar esse problema,

a cada nova seleção de um mesmo arquivo, fez-se uma iteração até se encontrar o ponto de parada anterior para continuar o processamento. Embora o tempo de processamento fosse afetado com essa solução, foi a maneira encontrada de se superar essa situação, já que o dBaseII oferece facilidades para manipulação apenas de dois arquivos simultaneamente. Isso não teria acontecido se o software utilizado fosse o dBaseIII, pois este tem capacidade de manipular até dez arquivos ao mesmo tempo, tendo em vista suas características que foram discutidas no Capítulo 2.

Outra dificuldade encontrada, devido a particularidades do software, foi com relação ao uso do comando LOCATE. Esse comando possui o seguinte formato: LOCATE FOR <expressão> e serve para localizar registros de um arquivo que satisfaçam a <expressão>. Então, o comando LOCATE localiza o primeiro registro e, o comando CONTINUE, cada vez que é executado, serve para localizar o próximo registro, onde a <expressão> seja verdadeira. No entanto, quando se está usando dois arquivos simultaneamente, um primário e outro secundário e precisa-se localizar registros no primário, com uma determinada condição e no secundário, com outra condição, iterativamente, mesmo selecionando-se um ou outro arquivo, ao se passar pelo CONTINUE, a <expressão> que será pesquisada no arquivo, será aquela correspondente ao último comando LOCATE executado e, normalmente, não é isso que se deseja. Para resolver o problema, substituiu-se o LOCATE mais interno por um comando iterativo e dessa forma, o CONTINUE passou a corresponder ao LOCATE externo.

Com exceção dessas duas notificações, pode-se dizer que o software atendeu, com certa facilidade de programação, às funções que deviam ser desempenhadas pelo sistema e também auxiliou na obtenção de um sistema facilmente manipulado pelo usuário final.

C A P Í T U L O 5

AVALIAÇÃO DO DESENVOLVIMENTO EFETUADO

5.1 - CONSIDERAÇÕES GERAIS

Neste capítulo faz-se uma avaliação de todas as fases que compõem o desenvolvimento efetuado: a fase de análise, a implementação realizada para solucionar o problema e os resultados obtidos quanto à satisfação dos requisitos do usuário.

No item 5.2, avalia-se a realização da análise estruturada, descrita no Capítulo 3, que foi elaborada para retratar as atividades do Escritório de Corretagem de Seguros, mencionado no Capítulo 1, com base em outros recursos que poderiam ser nela utilizados, mencionados no Capítulo 2.

No item 5.3 avalia-se a implementação realizada, descrita no Capítulo 4, comparando-a com outras formas candidatas para se implementar as funções requeridas pelo usuário.

No item 5.4 avalia-se a satisfação dos requisitos do usuário, isto é, se a implementação realizada atende às necessidades de maior interesse do Escritório de Corretagem de Seguros, mencionando-se caminhos alternativos aos quais poderia recorrer para ter seus objetivos alcançados.

Finalmente, no item 5.5, procura-se fazer uma avaliação do desenvolvimento efetuado como um todo, consolidando as avaliações efetuadas nos itens precedentes.

5.2 - AVALIAÇÃO DA ANÁLISE EFETUADA

Como foi visto no Capítulo 3, para se fazer a análise das atividades do Escritório de Corretagem de Seguros, utilizou-se o instrumental de análise estruturada, mencionado no Capítulo 2, correspondente à metodologia recomendada por Gane e Sarson [Ga77]. Com essa metodologia, pode-se dizer que o sistema ficou bem documentado e tal documentação retrata-o de maneira bem próxima à realidade.

Um dos aspectos dessa metodologia, que criou uma ponte muito útil para a fase de implementação, foi quanto à definição dos depósitos de dados. Depois do sistema estar completamente documentado, foram identificadas as principais funções que o usuário pretendia apoiar por computador. Essas funções correspondiam basicamente à manipulação de dados, isto é, à criação de arquivos eletrônicos que permitissem armazenar todas as informações necessárias e, posteriormente, consultá-las de várias maneiras diferentes, de forma a atender a todos os requisitos do usuário. Portanto, identificado esse tipo de processamento e escolhida a maneira de implementá-lo, que foi através do software de gestão de base de dados, dBaseII, ficou muito fácil definir-se os arquivos da base de dados a partir dos depósitos de dados identificados durante a fase de análise, nos diagramas de fluxo de dados. Para tanto, bastou construir o esquema da base de dados no mundo real, identificar os depósitos de dados que descreviam as entidades e seus relacionamentos, e aplicar sobre eles os conceitos de normalização, obtendo-se de maneira simples, os arquivos que deviam ser criados na base de dados para que satisfizessem a todas as consultas requeridas.

No entanto, existem alguns aspectos que foram deixados a descoberto pela metodologia recomendada por Gane e que poderiam ter sido apoiados por outras técnicas do instrumental de análise estruturada disponível, mencionado no Capítulo 2. Tais aspectos correspondem principalmente à dinâmica do sistema.

Uma das maneiras de se dar cobertura a esses aspectos é buscar solução nas metodologias para análise de escritório, pois antes de tudo, a micro empresa analisada pode ser vis

ta como um escritório e como tal, pode ser descrita através de metodologias apropriadas para esse ambiente. Por exemplo, uma das técnicas de documentação utilizadas na automação de escritório e que poderia se articular com a metodologia recomendada por Gane, é o método RCI [E179], como propõe Masiero, em [Ms84]. Através dessa articulação, seria possível demonstrar, mesmo que parcialmente, a sequência em que os processos dos diagramas de fluxo de dados acontecem. Essa articulação não seria total, mas de qualquer forma, ao menos o paralelismo de processos ou as condições necessárias aos fluxos de dados de entrada, para que um processo aconteça ou não, poderiam ser mostradas explicitamente.

Outra maneira de se dar cobertura na metodologia recomendada por Gane, aos aspectos da dinâmica do sistema e muitos outros que poderiam ser mostrados na documentação da análise, seria através da utilização da metodologia recomendada por Solvberg [So82], que é basicamente a metodologia recomendada por Gane, acrescida de vários detalhes. Dessa forma, ficaria perfeitamente definida a dinâmica do sistema, pois como foi dito no Capítulo 2, Solvberg criou uma representação para sinais, que são encarregados de disparar os processos do sistema. Outro aspecto que seria melhor abordado com a metodologia recomendada por Solvberg é aquele de dar nomes aos fluxos de dados, mencionado em [Fa84] e também no Capítulo 3, quando vários deles são saídas de um mesmo processo, mas não ocorrem simultaneamente. Isso acontece, pois a proposta de Solvberg possui o símbolo de decisão, que pode ser usado junto com o símbolo de processo, de forma que os fluxos de saída partam do símbolo de decisão, mostrando que apenas um deles irá ocorrer. Os vários tipos de mensagens que ocorrem em um sistema, como cartas, comunicados, etc, também podem ser precisamente definidos, pois existe uma simbologia própria para isso, que é capaz de diferenciar formulários, mensagens em papel, documentos preenchidos, etc.. Esses são apenas alguns aspectos, outros poderiam se tornar úteis na descrição do caso prático, se fosse realizada uma análise mais profunda com relação aos aspectos descritos por Solvberg e não descritos por Gane.

Por outro lado, existem aspectos da metodologia recomendada por Gane, que têm possibilidade de serem apoiados por

computador, através de técnicas que também fazem parte do instrumental de análise estruturada descrito no Capítulo 2.

Uma delas refere-se ao apoio ao dicionário de dados. Como foi mencionado no Capítulo 2 e demonstrado no Capítulo 3, através da documentação da análise realizada, a metodologia recomendada por Gane é apoiada pela elaboração de um dicionário de dados, que no caso em questão foi feito manualmente, consuindo a maior parte do tempo de elaboração da análise. Esse dicionário gerou um volume tão grande de documentação, que não foi possível colocá-lo integralmente nesta dissertação, tornando-se um trabalho anexo à mesma, [Fa84]. Uma opção para a elaboração do dicionário de dados pode ser dada pelo sistema PSL/PSA, [Te77], mencionado no Capítulo 2. Esse sistema permite a descrição de todos os objetos descritos por Gane e muito mais. Através dele pode-se descrever também aspectos da dinâmica do sistema. Além disso, o forte desse sistema é o conjunto de relatórios que ele pode fornecer: qualquer tipo de referência cruzada, estrutura de processos, estruturas de dados, listagens variadas de todos os objetos descritos, retratação da dinâmica do sistema e muitos outros. Portanto, o dicionário de dados poderia ter sido apoiado por computador, através do sistema PSL/PSA, o que embora não eliminasse a tarefa do usuário de definir e fornecer todas as informações ao sistema, reduziria bastante esse trabalho, pois o próprio sistema se encarrega de montar todos os relacionamentos inversos, isto é, define-se, por exemplo, que uma estrutura de dados é composta de alguns elementos; ao definir-se tais elementos, o próprio sistema coloca a relação inversa de que o elemento é componente daquela estrutura. Além disso, é inviável gerar-se manualmente todos os relatórios oferecidos pelo sistema PSL/PSA, que no entretanto são de bastante utilidade no próprio processo de análise.

Outra técnica que poderia auxiliar a metodologia recomendada por Gane, refere-se ao apoio por computador ao traçado de diagramas. Um recurso desse tipo agilizaria e facilitaria muito a elaboração da análise, reduzindo o tempo consideravelmente. Como exemplo de softwares desse tipo, pode-se citar o proMOD, [Hr83], ou o Excelerator, [In84], mencionados no Capítulo 2. Esses softwares possuem recursos gráficos para de

envolvimento "on-line" de qualquer tipo de diagrama de análise e projeto estruturados, bem como recursos para elaboração do dicionário de dados, montagem rápida de formatos de relatórios ou telas com os quais o usuário irá interagir, e outros.

Existe um aspecto que não pode deixar de ser mencionado, quando se utiliza a metodologia recomendada por Gane: trata-se de uma atividade muito trabalhosa e demorada. Para se ter uma base a fim de se quantificar esses parâmetros, o desenvolvimento e documentação da análise realizada levaram cerca de cinco meses. Grande parte desse tempo foi dedicado à elaboração do dicionário de dados que é realmente a fase que demanda maior trabalho, principalmente quando a própria confecção das fichas fica por conta do analista. Esse aspecto torna-se ainda mais crítico quando se considera que a mão de obra para cuidar da fase de análise deve ser altamente especializada. O problema que isso gera está relacionado ao encarecimento do serviço, a um ponto tal que nem toda empresa, principalmente a micro empresa, teria condições de assumir um compromisso assim tão pesado. Talvez uma forma de minimizar esse custo fosse através da utilização de um software que apoiasse por computador o desenvolvimento e documentação da análise, poupando muito tempo de elaboração tanto dos diagramas como do dicionário de dados, reduzindo consideravelmente o tempo e quem sabe, proporcionando um custo mais acessível à pequena empresa.

Portanto, verifica-se que com o auxílio de outros instrumentos de análise estruturada, seria possível ainda complementar, alterar e apoiar a documentação realizada, visando um aperfeiçoamento da mesma e, em alguns casos, permitindo que com esse auxílio, o desenvolvimento e documentação da análise se tornasse mais acessível com relação ao fator custo.

5.3 - AVALIAÇÃO DA IMPLEMENTAÇÃO EFETUADA

Existem três maneiras de se apoiar pelo computador as atividades de uma empresa: pela aquisição de aplicativos, isto é, de pacotes desenvolvidos para executar tarefas específicas e que são comercializados no mercado, pelo desenvolvimento de sistemas em linguagens tradicionais, como COBOL, BASIC, etc e pelo desenvolvimento expedito de sistemas, que utiliza os softwares lançados recentemente como planilhas eletrônicas, gerenciadores de base de dados, processadores de textos, etc..

Dentre essas três, a mais usada hoje em dia para a micro empresa, é a *aquisição de aplicativos*; primeiro, devido ao baixo custo destes, comparado com o das outras duas e segundo, como acontece em muitos casos, devido ao procedimento dos vendedores de micros que, a fim de vender seus equipamentos, exibem os aplicativos como se estes fossem a solução de todos os problemas. Aliás, ao adquirir um equipamento por essas vias, a empresa comete um grande erro, pois normalmente, esse equipamento não vai corresponder às suas necessidades, visto que as especificações do mesmo não são definidas por uma pessoa que conhece os problemas da empresa, mas sim, pelo vendedor. Nesses casos, a própria ordem dos procedimentos, para ser mais correta, deve ser exatamente a contrária, isto é, a empresa define a sua necessidade de software, decide como vai obtê-lo (por desenvolvimento interno, por consultoria ou por aquisição de aplicativos) e, só então, ela tem condições de determinar o equipamento necessário.

Então, quando a opção feita é a aquisição de aplicativos, deve-se tomar muito cuidado, pois na verdade, eles podem trazer uma série de desvantagens à empresa, como por exemplo: em geral, os aplicativos não se adaptam perfeitamente aos problemas específicos da mesma e para que isso aconteça, devem sofrer uma série de modificações. No entanto, nem sempre tais modificações são possíveis pois, ao se adquirir um aplicativo, recebe-se apenas o código objeto e não o código fonte, o que inviabiliza tal solução. Se a empresa já possui um equipamento, o aplicativo deve ser com ele compatível e, se não o for, certamente não pode ser utilizado sem sofrer uma série de adaptações. Para se adquirir um aplicativo, deve-se conhecer o con

ceito da empresa que o comercializa e também, quais as empresas que o adquiriram, a fim de se indagar o nível de satisfação pela utilização do mesmo e o atendimento por parte do fornecedor. Deve-se também ter certeza de que o fornecedor garanta manutenção, implantação, treinamento na operação do produto, suporte em documentação e atendimento de emergência. Finalmente, deve-se verificar o grau de integração do aplicativo com outros sistemas que eventualmente devam ser desenvolvidos.

Em contra partida a essa opção de aquisição de aplicativos, tem-se outras duas, ditas anteriormente, que são o desenvolvimento em linguagens tradicionais e o desenvolvimento expedito de sistemas. As duas possuem vantagens sobre a primeira opção, pois ambas desenvolvem o sistema especificamente para atender aos problemas da empresa. Dessa forma, utilizando-se qualquer uma dessas duas opções pode-se dizer que se a empresa já possui um equipamento, não haverá problemas pois o sistema será desenvolvido de acordo com o equipamento existente; quanto ao atendimento, quer o sistema seja desenvolvido por pessoas internas ou por firmas de consultoria, sempre há interesse no bom atendimento e, quanto à integração, pelo fato do sistema ser desenvolvido para atender especificamente a empresa, naturalmente ele é mais maleável, pois seu projeto deve levar em consideração os problemas futuros e deixar o sistema apto a modificações e expansões.

Quanto ao *desenvolvimento em linguagens tradicionais* com COBOL, BASIC, etc, embora seja mais eficiente e mais vantajoso em relação aos aplicativos, pode-se dizer que seu projeto é muito trabalhoso, deve ser bem detalhado e, pelo fato das linguagens não possuírem recursos específicos para tipos de processamentos dominantes, como gerenciamento de base de dados, processamento de textos, etc, gera programas longos, complexos e difíceis de entender. Por todos esses motivos, o desenvolvimento do sistema pode tomar tal amplitude, de forma a envolver várias pessoas para se conseguir chegar a uma solução. Tudo isso faz com que, dentre as três opções de implementação de um sistema por computador, esta seja a mais onerosa delas.

Finalmente, através do *desenvolvimento expedito* de

sistemas, pode-se escolher uma linguagem que se adapte melhor ao problema em questão, pois isso facilita muito o desenvolvimento do sistema. Como exemplo, cita-se o desenvolvimento realizado, do caso prático em questão, descrito no Capítulo 4, onde se utiliza o software de gestão de base de dados, o dBaseII, para resolver os problemas da Corretora de Seguros, visto que o tipo de processamento dominante é o de manipulação de arquivos. Assim, por se usar uma linguagem adequada ao problema, obtém-se programas bem mais simples, fáceis de entender e pequenos, ao passo que utilizando-se linguagens tradicionais, os mesmos programas podem tornar-se extensos, complexos e de difícil alteração, caso isso seja necessário.

Segundo [Ba83], a tendência dos aplicativos que dominam o mercado hoje em dia, é desaparecer; as aplicações serão cada vez mais personalizadas e ajustadas para os propósitos específicos da empresa. Portanto, de um lado tem-se os aplicativos que possuem uma série de desvantagens inerentes, mas a grande vantagem do custo baixo; e, do outro lado, tem-se as outras duas opções para desenvolvimento de sistemas, bem mais eficientes e vantajosas que a primeira, mas que perdem para esta em termos de custos.

Assim, num consenso geral, pode-se dizer que, a menos que o aplicativo seja quase que cem por cento satisfatório para a empresa, deve-se optar por uma das outras duas alternativas para desenvolvimento de sistemas. E, além disso, pode-se colocar o desenvolvimento expedito de sistemas como um meio termo ou uma solução ponderada entre os dois extremos, pois o seu custo apesar de não ser baixo como o do aplicativo, também não é alto como o do desenvolvimento em linguagens tradicionais mas, sua eficiência e vantagens são tão boas quanto as desse último.

No entanto, deseja-se chamar a atenção para o seguinte fato: a implementação efetuada, como se pode observar no Capítulo 4, trata de funções específicas do ramo de negócio de Corretagem de Seguros e, como já dizia Marienthal em [Ma78], os pacotes, na verdade, não atacam o âmago de um negócio específico mas sim, são desenvolvimentos de funções que existem em vá

rios ramos de negócios distintos. Talvez tenha sido esse o motivo de não ter sido encontrado um pacote para esse ramo de neg^ocio. Portanto, não se pretende descartar por completo a aquisição de aplicativos mas, ao contrário, no caso deles existirem, é necessário um estudo minucioso antes de se optar por uma ou outra solução precipitadamente.

5.4 - AVALIAÇÃO DA SATISFAÇÃO DOS REQUISITOS DO USUÁRIO

Como se vem observando neste trabalho, principalmente nos Capítulos 3 e 4 que tratam respectivamente, da análise e implementação do caso prático abordado, referente ao Escritório de Corretagem de Seguros, as funções implementadas correspondem às necessidades prioritárias desse Escritório. E, como se nota, elas são funções bem específicas desse ramo de negócio.

Dessa forma, identifica-se nesse problema uma situação típica colocada por Marienthal, em [Ma78], mencionada no Capítulo 2 e também no item anterior, onde ele diz que para resolver funções específicas, raramente um pacote se ajusta perfeitamente à situação. Isso porque os pacotes correspondem a um desenvolvimento horizontal de software, ou seja, normalmente correspondem a funções específicas, mas que estão inseridas em vários ramos de negócios diferentes. Dificilmente uma empresa de software especializa-se no desenvolvimento vertical de pacotes, isto é, no desenvolvimento de funções distintas que fazem parte de um mesmo ramo de negócios. Em outras palavras, dificilmente uma empresa de software especializa-se num ramo de negócio específico, pois com o desenvolvimento horizontal, ela é capaz de cobrir uma gama muito maior de empresas que utilizam seus produtos.

Portanto, por se tratar de funções específicas de controle de um Escritório de Corretagem de Seguros, acredita-se que a abordagem dada à implementação realizada, atende satisfatoriamente aos requisitos do usuário. Acredita-se também ter sido correto utilizar-se um método de desenvolvimento expedito de sistemas, nesse caso particular, visto que o problema abordado, por corresponder a funções específicas, identifica-se com o desenvolvimento vertical. No entanto, se o proprietário da Corretora tivesse feito a aquisição de um microcomputador, como era sua intenção inicialmente, para atender as necessidades prioritárias, com certeza, depois destas, muitas outras iriam aparecer, como por exemplo, funções de contabilidade, de controle de comissões de colaboradores, de emissão de cartas, de emissão de etiquetas para correspondências, etc.. Então, o tipo de desen

volvimento adotado iria depender da função a ser desenvolvida, pois dentre elas, pode-se identificar antecipadamente que, por exemplo, a emissão de cartas, uma típica função de escritório, como destaca Burns, em [Br77], acompanharia o mesmo tipo de desenvolvimento, pois seria conveniente utilizar um processador de texto, que também é considerado um recurso para desenvolvimento expedito; na emissão de etiquetas, talvez um desenvolvimento de sistemas numa linguagem tradicional, tipo BASIC ou COBOL, seria melhor; no controle de comissões, o desenvolvimento expedito, através do próprio dBaseII, resolveria o problema e, quanto às funções de contabilidade, talvez um pacote seria a solução, ou seja, as necessidades posteriores poderiam se ajustar tanto à linha horizontal, como à linha vertical de desenvolvimento de sistemas e portanto, deveriam ser encaradas como casos distintos, merecendo um estudo particular para cada um deles. Mas, por outro lado, se a aquisição do micro tivesse sido realizada, e junto com ela, a aquisição do dBaseII para resolver as necessidades prioritárias, antes de ser adotada qualquer outra alternativa, talvez fosse aconselhável, em primeiro lugar, estudar a possibilidade de se resolver os novos problemas com os recursos já adquiridos para tentar minimizar os custos.

No entanto, sendo este trabalho referente à utilização de microcomputadores em micro empresas, não se sabe até que ponto uma micro empresa tem condições de custear uma documentação tão detalhada durante a fase de análise. Sendo assim, outros meios para uma micro empresa adquirir um sistema para apoiar por computador principalmente as suas funções específicas, são aqueles citados também por Marienthal, em [Ma78]: vendas diretas tradicionais, vendas através de distribuidores regionais, vendas através de firmas de O&M e vendas através de varejistas locais (lojas de micros). Dentre eles, os dois primeiros comercializam apenas equipamentos e o software de aplicação é, em geral, desenvolvido por pessoas externas. Os dois outros, isto é, as firmas de O&M e as lojas de micros, ambos comercializam software e o tratam como um recurso a mais para se firmarem no mercado. No entanto, os sistemas normalmente comercializados por elas são aqueles referentes a funções horizontais, isto é, pacotes, o que não impede à micro empresa encomendar um sistema para resolver funções específicas do seu ramo de negócio. Além

disso, McCartney, em [Mc78], fala da importância dos pequenos sistemas na indústria de processamento de dados, o que provoca uma corrida de todas as vias de comercialização a esse mercado, dando origem a sistemas mais específicos, de onde subentende-se que, talvez a tendência futura seja dar ênfase a sistemas verticais ao invés de sistemas horizontais.

Caso a pequena empresa seja filiada ou mantenha algum tipo de relação com uma grande empresa, como o caso em questão, pois de certa forma a Corretora pode ser encarada como uma subagência das Seguradoras com as quais ela trabalha, uma outra maneira da pequena empresa obter um sistema para suas funções essenciais é, segundo Myers [My83], possuir um microcompu tador que funcione como terminal inteligente, acoplado a um computador de grande porte da empresa com a qual ela trabalha. Além disso, o próprio Myers dá outra solução para a aquisição de pequenos sistemas que é através de bureaux, pois estes estão fazendo do microcomputador um novo veículo de venda de seus produ tos. Dessa forma, a pequena empresa pode obter o micro adequado ao sistema específico que ela encomendar.

Como se observou no Capítulo 2, esse problema vem sendo atacado já há muito tempo, desde antes de 1968. Nota-se que durante essas duas décadas, muita coisa evoluiu, começando-se com os blocos de construção de software, e chegando-se aos blocos de software mais hardware, que são considerados a tendência futura. Mas, o que se nota durante todo esse percurso, é uma constante preocupação com a análise, sempre na tentativa de eliminar o seu ônus. Na época dos blocos de construção de soft ware, a análise era feita a título de pesquisa, pelas universidades, justamente para minimizar os custos da aplicação de com putadores a micro empresas, o que dava origem a esquemas fixos, aos quais as empresas tinham que se ajustar. De certa forma, o mesmo acontece com as tendências futuras, de blocos de software mais hardware, quando se passará a pedir "um computador para contas a receber", segundo [Ma78], pois indiretamente, a empresa estará caracterizando o seu tipo de processamento como sendo mais tendente a certas funções do que a outras, não conseguindo atingir novamente a essência de seu negócio. Assim, nota-se que a

dificuldade toda está sempre incidindo no fato de se tentar desenvolver sistemas específicos a custos acessíveis para pequenas empresas e, independentemente de toda evolução que ocorreu, durante todo esse tempo, o trabalho de análise é uma constante.

O que se observa, como é dito em [Ca78], é que o preço do hardware vem diminuindo dia a dia, ao mesmo tempo em que suas capacidades se aproximam às dos equipamentos de grande porte; o software também vem acompanhando esse fato mais lentamente mas, quanto aos projetos de sistemas, ainda não se chegou a uma solução pois eles continuam não atendendo os objetivos essenciais da micro empresa.

Quem sabe não se chega um dia a um meio termo, ou seja, não se desenvolve um software específico, porém mais geral, bem documentado e fácil de sofrer alterações. Talvez isso ainda venha se concretizar como um modelo de aplicativo; mas, desde já, pode-se identificar pelo menos um problema que surgiria, que é aquele quanto à sua aquisição, conhecido como pirataria de software, pois sendo bem documentado e apto a modificações, qualquer pessoa poderia copiá-lo e ajustá-lo a seus propósitos particulares, sem pagar por isso.

Esse problema poderia ser contornado, através de bureau, pois este, dentre todas as outras fontes onde a pequena empresa pode buscar a solução para seus problemas de processamento de dados, parece ser a mais promissora. Isso não apenas pelo suporte técnico que o bureau possui para o desenvolvimento de sistemas mas também, pelo fato de poder permitir, como diz Myers em [My83], que as pequenas empresas com ele interajam através de terminais de teleprocessamento, usando aplicativos por ele desenvolvidos, bem documentados, fáceis de alterar para que atendam aos propósitos de uma empresa particular, e ao mesmo tempo protegidos, livres do problema de pirataria, pois ao usuário caberia apenas a utilização do pacote, ficando sua manutenção e alteração, a cargo do próprio bureau.

Além disso, a micro empresa teria acesso a um computador de grande porte que, para sistemas estruturados tem melhores condições de processamento do que os microcomputadores. Esses, além de poderem servir de terminais inteligentes para esse tipo de aplicação, poderiam fazer todo o processamento dos sistemas semi-estruturados e de apoio à decisão.

5.5 - CONCLUSÕES

Como foi mencionado no item 5.2, observa-se que, em bora o desenvolvimento e documentação da análise, realizados pa ra o caso prático em questão, tenha ficado bastante completo e tenha retratado o sistema de forma a ficar bem próximo do que acontece na realidade, ainda assim, tal desenvolvimento poderia ser aprimorado, fazendo-se uso de outras técnicas que compõem o instrumental de análise estruturada disponível. Salientou-se que a elaboração da análise é um ponto crítico no desenvolvimento rea lizado, pois ela foi muito trabalhosa, demorada e consequente mente acarretaria um alto custo, caso não tivesse sido desenvolvida num contexto acadêmico. Como não é o caso, verificou-se, juntamente com as outras técnicas, que o trabalho de análise po deria ser complementado. Sendo assim, dependendo da técnica utilizada, como por exemplo, os softwares de apoio por computa dor ao desenvolvimento da análise, o mesmo tipo de documentação poderia ser feito, talvez a um preço mais acessível à micro em presa, pois todo trabalho de elaboração dos diagramas de fluxo de dados, diagramas de estrutura modular, fichas do dicionário de dados, que foi todo feito manualmente, seria apoiado pelo pró prio computador.

De qualquer forma, foi apoiando-se na documentação da análise que identificou-se os requisitos básicos do usuário. Então, foram analisados os tipos de soluções que poderiam ser empregadas para apoiar por computador tais requisitos. Esses ti pos compreendem: a aquisição de aplicativos, o desenvolvimento em linguagens de programação tradicionais (COBOL, BASIC, etc.) e o desenvolvimento expedito, através de linguagens especializa das para o tipo de processamento requerido, que no caso em ques tão, foi identificado como o de manipulação de arquivos. Tal processamento visava soluções de problemas específicos do ramo de negócios de corretagem de seguros. Assim, podia ser encara do como um desenvolvimento vertical de sistemas, segundo [Ma78], mencionado no Capítulo 2, embora tenha-se classificado apenas os requisitos básicos do usuário e não todas as funções realiza das no Escritório.

Apesar dos aplicativos possuírem a grande vantagem

do baixo custo, possuem também a desvantagem de precisarem de uma série de adaptações para se ajustarem a problemas específicos, sem se falar no problema de que, normalmente, eles atendem a funções horizontais e não verticais. Quanto aos outros dois tipos de soluções, ambos podem ser empregados no desenvolvimento de funções específicas. No entanto, o desenvolvimento em linguagens tradicionais normalmente é caro e mais trabalhoso pelo fato da linguagem não possuir recursos para abordar tipos de processamento dominantes, como é o caso em questão. Dessa forma, nesse caso particular, optou-se por um desenvolvimento expedito de sistemas, principalmente por se ter disponível software de gestão de dados, com o qual podia-se resolver o problema de maneira mais fácil.

Assim, acredita-se que a implementação realizada atende aos requisitos do usuário. No entanto, considerando-se o desenvolvimento como um todo, isto é, análise e implementação, fica em suspenso a questão da análise, devido ao custo e, por esse motivo, mostrou-se outras formas através das quais uma micro empresa pode conseguir seu software, como por exemplo, em lojas de micros, bureaus, etc..

C A P Í T U L O 6

CONCLUSÕES FINAIS E SUGESTÕES PARA NOVAS PESQUISAS

6.1 - CONSIDERAÇÕES GERAIS

Como foi visto no Capítulo 1, há muito tempo a autora já possuía interesse na área de documentação de sistemas de informação e encontrou, no caso prático abordado, oportunidade de realizar o desenvolvimento de um sistema apoiado em instrumental especializado. Também nesse capítulo, caracterizou-se o ambiente do caso prático abordado, correspondente a um Escritório de Corretagem de Seguros.

No Capítulo 2, apresentou-se um referencial bibliográfico que deu apoio aos demais capítulos desta dissertação.

No Capítulo 3, apresentou-se a análise estruturada do caso prático descrito no Capítulo 1, utilizando-se instrumental mencionado no capítulo anterior.

O Capítulo 4 apresentou a implementação realizada para atender aos requisitos do usuário detectados nessa análise, utilizando-se um componente do instrumental para desenvolvimento expedito de sistemas, discutido no Capítulo 2.

No Capítulo 5, fez-se uma avaliação do trabalho realizado quanto à análise, implementação e resultados obtidos, em vista das necessidades do usuário, com base nas informações contidas em referências apresentadas no Capítulo 2.

Com base em todos esses capítulos precedentes, no item 6.2, apresenta-se as conclusões finais decorrentes de todo trabalho efetuado; no item 6.3 faz-se algumas sugestões para dar continuidade a esta pesquisa, concentrando-se em aspectos que poderiam ser complementados e finalmente, no item 6.4, fala-se sobre as contribuições conseguidas com a realização deste trabalho.

6.2 - CONCLUSÕES FINAIS

Ao analisar a bibliografia sobre o instrumental de análise estruturada, como mencionado no item 2.5, a metodologia recomendada por Gane e Sarson [Ga77] mostrou-se bastante apropriada, razão pela qual foi aplicada no desenvolvimento da análise do caso prático abordado. Tal aplicação, como foi mencionado no item 3.5, trouxe grande facilidade de comunicação entre analista e usuário, permitindo chegar-se a resultados muito satisfatórios, visto que a descrição final do sistema se aproximou bastante da realidade e, além disso, deixou o sistema amplamente documentado através dos diagramas de fluxo de dados e do dicionário de dados. Todavia, como foi mostrado no item 5.5, deixou a descoberto alguns aspectos, principalmente aqueles relacionados com a dinâmica do sistema, que poderiam ter sido melhor cuidados com uma articulação da metodologia utilizada com o método RCI [El79], ou então, com a utilização da metodologia recomendada por Solvberg [So82]. Além disso, como também foi mostrado no item 5.5, o desenvolvimento da análise poderia ter sido apoiado por computador através de softwares para traçado de diagramas e para apoio ao dicionário de dados, que ajudariam a reduzir o tempo desse desenvolvimento.

Ao analisar a bibliografia sobre o instrumental para desenvolvimento expedito de sistemas, como mencionado no item 2.5, o software de gestão de base de dados, dBaseII, mostrou-se apropriado para resolver o problema e por isso, foi utilizado na implementação. A utilização desse software, como mostrou o item 4.6, atendeu com certa facilidade ao processamento requerido, possibilitando um sistema facilmente utilizado pelo usuário final. Sua escolha deveu-se ao fato, como mostrou o item 2.5, de ter-se identificado, através da análise realizada, que nas funções a serem apoiadas por computador, predominava um tipo de processamento específico, correspondente à manipulação de arquivos e que a aplicação do dBaseII, como mostrou o item 5.5, correspondia ao desenvolvimento expedito de sistemas, pois este significa desenvolver-se um sistema através de linguagens especializadas para tipos de processamentos dominantes. No entanto, alguns módulos desse sistema necessitaram de informações conti

das em três arquivos diferentes e, nesse momento, o dBaseII dificultou um pouco a solução, pois ele permite que apenas dois arquivos estejam abertos simultaneamente; problema este que não teria surgido se o sistema fosse implementado com o dBase III.

Finalmente, ao analisar a bibliografia sobre a evolução da utilização de computadores nas empresas até chegar-se à época dos microcomputadores nas micro empresas, como mostrou o item 2.5, identificou-se que o caso prático seria melhor abordado através de um desenvolvimento vertical, que corresponde a desenvolver funções de um ramo de negócio específico. Outras abordagens que poderiam ter sido dadas à implementação, como mostrou o item 5.5, seriam através da aquisição de aplicativos ou do desenvolvimento em linguagens tradicionais, como COBOL, BASIC, etc. No entanto, essas abordagens foram descartadas, nesse caso específico, pois a primeira atende principalmente a funções específicas empregadas em vários ramos de negócios distintos, isto é, a um desenvolvimento horizontal e na segunda, normalmente, as linguagens não possuem recursos para atender a tipos de processamentos dominantes. No entanto, como também foi dito em 5.5, foram selecionados apenas os requisitos básicos do usuário e um desenvolvimento substancialmente vertical, na verdade, abrangeria todas as funções executadas no ramo de negócios, como por exemplo, folha de pagamento dos funcionários, contabilidade (na época, realizada externamente ao Escritório), controle de comissões de colaboradores, etc..

6.3 - SUGESTÕES PARA NOVAS PESQUISAS

Para dar continuidade a esta pesquisa, novos trabalhos poderiam abordar os seguintes aspectos:

Primeiro, quanto à análise realizada, pois como foi dito no item anterior, outras técnicas do instrumental de análise estruturada poderiam ter sido utilizadas para complementar a documentação realizada. Dessa forma, poder-se-ia aplicar, no mesmo problema abordado, a metodologia recomendada por Solvberg [So82], visto que possui recursos para explicitar muitos aspectos que não são cobertos pela metodologia utilizada neste trabalho. Com a documentação produzida, seria possível fazer-se uma comparação entre as duas metodologias e estabelecer-se situações onde uma é mais apropriada do que a outra; ou ainda, avaliá-las quanto à persistência das facilidades de comunicação com o usuário, após inserir-se uma série de outros detalhes nos diagramas de fluxo de dados.

Outra avaliação poderia ser feita, utilizando-se uma articulação do método RCI [El79], para análise de escritório, com a metodologia utilizada, verificando-se as vantagens e desvantagens conseguidas com essa articulação e contrapondo-a também com relação à metodologia recomendada por Solvberg, a fim de se fazer uma análise quanto à comunicação com o usuário, pois uma articulação do método RCI com a metodologia utilizada neste trabalho não gera um diagrama de fluxo de dados com tantos detalhes como a outra, não colocando em risco, dessa forma, o fator de comunicação.

Um outro aspecto a ser pesquisado, seria quanto ao tempo de desenvolvimento e documentação da análise, com as mesmas técnicas utilizadas mas, apoiando-se esse desenvolvimento por computador, através de softwares de apoio ao traçado de diagramas ou ao dicionário de dados, com os quais esse tempo talvez fosse bastante reduzido.

Segundo, quanto à implementação elaborada, no sentido de tentar-se acoplá-la com outras formas de desenvolvimento de sistemas, visto que, se todas as funções que são executadas no Escritório fossem apoiadas por computador, certamente algu

mas seriam melhor abordadas empregando-se um desenvolvimento em linguagens tradicionais, como talvez fosse o caso de impressão de etiquetas para correspondências; outras, pela aquisição de aplicativos, se eles atendessem satisfatoriamente ao caso específico do Escritório, como talvez fosse o caso da folha de pagamento dos funcionários e outros ainda, pelo próprio desenvolvimento expedito de sistemas, como por exemplo, a impressão de cartas, que com certeza seria facilmente resolvida com processadores de textos, os quais também fazem parte desse instrumental.

6.4 - CONTRIBUIÇÃO DESTE TRABALHO

Considera-se como principal contribuição deste trabalho, o estudo minucioso nele contido, sobre como resolver com microcomputadores os problemas de desenvolvimento de sistemas de informação para micro empresas. Diversas referências bibliográficas sobre esse assunto foram discutidas e os instrumentos mais promissores, nelas apresentados, foram aplicados a um caso prático real.

Nesse caso prático, as funções específicas do ramo de negócios foram identificadas e detalhadamente analisadas, com o que foi possível construir um sistema de informação para apoiá-las. Entretanto, com este trabalho, constatou-se que a aplicação de metodologias tradicionais para documentação e desenvolvimento de análise estruturada fica além das possibilidades de uma micro empresa. Portanto, qualquer esforço de pesquisa sobre a aplicação de microcomputadores a micro empresas deve estar concentrado principalmente no desenvolvimento da análise, criando-se uma maneira de transpor essa fase de forma mais acessível à micro empresa.

Verificou-se também que através de desenvolvimento expedito de sistemas, os problemas específicos de um ramo de negócios podem ser atacados diretamente, de forma que a solução obtida gere uma satisfação mais completa dos requisitos do usuário e sobretudo, que o custo desse tipo de implementação seja razoavelmente acessível à micro empresa.

Com relação ao Escritório de Corretagem de Seguros abordado nesta dissertação, o trabalho realizado contribuiu no seguinte aspecto: após o desenvolvimento da análise, as atividades do Escritório ficaram amplamente documentadas através dos diagramas de fluxo de dados e do dicionário de dados. Segundo o próprio proprietário, com essa documentação ele obteve um "guia" de seu Escritório, o qual pôde ser usado na instrução de novos funcionários, principalmente no que diz respeito ao preenchimento de formulários, pois todas as informações neles contidas, estão descritas detalhadamente no dicionário de dados.

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

- As84 ASHTON-TATE - dBaseIII User Manual. 1984.
- Ba83 BALZER, R.; CHEATHAM Jr., T.E.; GREEN, C. - Soft
ware Technology in the 1990's: Using a New Para
digm, nov. 1983.
- Bo83 BOEWERMAN, R. - Relational Database Systems for Mi
cros. Datamation, jul. 1983.
- Bu69 BUCKLE, D.G.R.; HUNT, C.H.; RYDER, M.M.; STOREY, S.
H. - Computers and the small firm: 2. The Com
puter Journal, 12(1), fev. 1969.
- Bu70 BUCKLE, D.G.R.; HUNT, C.H.; RYDER, M.M.; STOREY, S.
H. - Computers and the small firm: 3. The Com
puter Journal, 13(1), fev. 1970.
- Br77 BURNS, C. - The Evolution of Office Information Sys
tems. Datamation, abr. 77.
- By85 BYERS, R.A. - dBaseIII Banco de Dados para todas
as Aplicações. McGraw-Hill, 1985.
- Ca78 CANNING, R.G.; McNURLIN, B. - Micros Invade the Bu
siness World. Datamation, ago. 1978.
- Co85 COSENTINO, L.J.L. - dBaseII - Solução para Micro
computadores. Ed. Atlas, 1985.
- De78 DeMARCO, T. - Structured Analysis and System Speci
fication. Yourdon Inc., New York, 1978.
- Di83a DIGITAL Equipment Corporation - VAX-11 Datatrieve
Summary Description. 1983.
- Di83b DIGITAL Equipment Corporation - VAX-11 Datatrieve
User's Guide. 1983.

- Di83c DIGITAL Equipment Corporation - VAX-11 Datatrieve
Guide to Writing Reports. 1983.
- Di83d DIGITAL Equipment Corporation - VAX-11 Datatrieve
Guide to Using Graphics. 1983.
- E179 ELLIS, C.A. - Information Control Nets: A Mathemati
cal Model of Office Information Flow. In: Proc.
ACM Conf. Simulation, Modeling and Measurement
of Computer Systems, 1979, 225-240.
- Fa82 FABBRI, S. - Duas Experiências de Análise Estruturaa
da de Sistemas de Informação existentes. In:
Anais da SUCESU, Rio de Janeiro, 1982..
- Fa84 FABBRI, S. - Análise Estruturada de um Escritório
de Corretagem de Seguros, trabalho realizado dur
rante o período de Pós-graduação, no ICMSC-USP,
1984.
- Ga77 GANE, C.; SARSON, T. - Structured Systems Analysis:
Tools and Techniques. Improved System Technolog
gies Inc., 1977.
- Hr83 HRUSCHKA, P. - The Software Engineering Environment
proMOD. Proc. ESA/ESTEC Software Engineering
Seminar, Noordwijk, out. 1983.
- Hu68 HUNT, C.H.; RYDER, M.M.; STOREY, S.H. - Computers
and the small firm: 1. The Computer Journal,
11(1), mai. 1968.
- Ib74 IBM Corporation, Data Processing Division - HIPD: A
Design Aid and Documentation Technique. White
Plains, New York, 1974.
- In84 INDEX Technology Corporation - Excelerator: The Feat
tures of Excelerator. Cambridge, Massachusetts,
1984.

- Kr83 KRUGLINSKI, D. - Data Base Management Systems - A Guide to Microcomputer Software. Osborne/McGraw-Hill, California, 1983.
- Lu81 LUNDEBERG, M.; GOLDKUHL, G.; NILSON, A. - Information Systems Development: a Systematic Approach. Prentice-Hall, 1981.
- Ma75 MARIENTHAL, L.B. - Small Computers for Small Businesses. Datamation, jun. 1975.
- Ma78 MARIENTHAL, L.B. - Selling Small Business Systems. Datamation, out. 1978.
- Mc78 MCCARTNEY, L. - Small Business Systems: They're everywhere. Datamation, out. 1978.
- Ms84 MASIERO, P.C. - Um prognóstico sobre a Integração entre Automação de Escritório e Processamento de Dados. Tese de Doutorado apresentada à Faculdade de Economia e Administração da Universidade de São Paulo, 1984.
- My83 MYERS, E.D. - Bureaus Sell Micros. Datamation, abr. 1983.
- Ra81 RATLIFF, W. - Assembly Language Relational Database Management System. Ashton-Tate, California, 1981.
- Sf76 SOFTECH - The Software Technology Company - An Introduction to SADT - Structured Analysis and Design Technique. Softech, Inc., 1976.
- So82 SOLVBERG, A. - A Draft Proposal for Integrating System Specification Models. In: Information Systems Design Methodologies: A Comparative Review. North-Holland Publishing Company, 1982.

- St74 STEVENS, W.P.; MYERS, G.J.; CONSTANTINE, L.L.-Struc
tured Design. In: Tutorial on Structured Pro
gramming: Integrated Practices. IEEE, 1981.
- St81 STEVENS, W.P. - Using Structured Design. Wiley-In
terscience Publication, 1981.
- St85 STEVENS, W.P. - Using Data Flow for Application De
velopment. Byte, jun. 1985.
- Te77 TEICHROEW, D.; HERSHEY, E.A., III - PSL/PSA: A Com
puter-Aided Technique for Structured Documenta
tion and Analysis of Information Processing Sys
tems. IEEE Transactions on Software Engineering
Vol.SE-3, nº1, jan. 1977.
- To85 TOWNSEND, C. - dBaseII Guia do Usuário. McGraw-Hill,
1985.
- Yo75 YOURDON, E.; CONSTANTINE, L.L. - Structured Design:
Fundamentals of a Discipline of Computer Program
and Systems Design. Yourdon Press, New York,
1975.
- Yo75a YOURDON, E. - Techniques of Program Structure and
Design. Prentice-Hall, Englewood Cliffs, New
Jersey, 1975.
- Yo76 YOURDON, E. - Managing the Structured Techniques.
Prentice-Hall, Inc., Englewood Cliffs, New Jer
sey, 1976.
- Yo77 YOURDON, E. - Structured Walkthroughs. Prentice
-Hall, Inc., Englewood Cliffs, New Jersey, 1977.
- We78 WEINBERG, V. - Structured Analysis. Yourdon Press,
New York, 1978.

APENDICE

Este apêndice contém a estrutura modular de todos os programas desenvolvidos na implementação e ele está organizado da seguinte maneira:

Apresenta-se um diagrama de estrutura e, logo em seguida, a listagem do programa correspondente a ele. Esses diagramas são compostos de dois tipos de "caixas": uma simples e outra com as laterais duplicadas. O primeiro tipo corresponde a instruções do corpo do próprio programa, por exemplo, *Inicialize Sinaleira* corresponde à instrução *STORE S~TO XMENU*, se for considerado o primeiro diagrama e seu respectivo programa. O segundo tipo corresponde a um outro programa, cujos respectivos diagrama e listagem são apresentados mais adiante; por exemplo, *Obtenha Opção do Menu de Inserção* que corresponde à instrução *DO INSERCAO*, equivale a um outro programa que é apresentado com detalhes posteriormente e cujo nome aparece tanto na instrução *DO*, como também, em letras inclinadas, na primeira linha da referida caixa do diagrama de estrutura.

Existem programas que são chamados a partir de vários outros e portanto, aparecem representados em caixas de laterais duplicadas nos diagramas de todos eles. Nesses casos, o seu diagrama e respectivo programa são colocados no final, depois de todos os outros que fazem referência a ele.

No caso de um programa ser chamado somente por um outro, o seu diagrama e sua listagem seguem imediatamente àquela do programa que o chamou.

Além disso, a ordem em que eles estão dispostos é: primeiramente, o diagrama e a listagem do programa referente ao cardápio principal; em seguida, o cardápio de inserções com os respectivos programas de inserção; depois, o cardápio de relatórios padrão e os respectivos programas; depois, o cardápio de delições com os respectivos programas e, finalmente, os programas que são comuns a vários outros.

Cardápio Principal	128
Operações de Inserção	130
Emissão de Relatórios	152
Operações de Delição	178
Programas comuns a vários outros	184

Diagrama de Estrutura Modular e Listagem do Programa MENUPRIN.CMD

```

A)TYPE MENUPRIN.CMD
ERASE
SET TALK OFF
SET FORMAT TO PRINT
EJECT
STORE 'S' TO XMENU
DO WHILE XMENU = 'S'
    @ 3,25 SAY 'MENU DAS OPERACOES BASICAS'
    @ 4,25 SAY '=====
    @ 7,30 SAY '1- INSERCAO'
    @ 9,30 SAY '2- RELATORIOS PADROES'
    @ 11,30 SAY '3- DELICAO'
    @ 13,30 SAY '4- TERMINAR EXECUCAO'
    SET FORMAT TO SCREEN
    STORE ' ' TO XOPCAO
    @ 21,55 SAY 'DIGITE SUA OPCAO ';
        GET XOPCAO PICTURE 'X'
    READ
    CLEAR GETS
    DO CASE
        CASE XOPCAO = '1'
            DO INSERCAO
        CASE XOPCAO = '2'
            DO RELATPAD
        CASE XOPCAO = '3'
            DO DELICAO
        CASE XOPCAO = '4'
            STORE 'N' TO XMENU
    OTHERWISE
        @ 23,55 SAY 'NAO EXISTE ESSA OPCAO'
        WAIT
    ENDCASE
ERASE
ENDDO
RELEASE ALL
SET TALK ON
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa INSERCAO.CMD

```

A)TYPE INSERCAO.CMD
ERASE
SET TALK OFF
SET FORMAT TO PRINT
EJECT
STORE 'S' TO XMENUINS
DO WHILE XMENUINS = 'S'
    @ 1,25 SAY 'MENU DAS OPERACOES DE INSERCAO'
    @ 2,25 SAY '=====
    @ 4,6 SAY 'INSERCAO DE : '
    @ 6,20 SAY '1- SEGURADO DE VG/AP'
    @ 8,20 SAY '2- SEGURADOS DOS OUTROS TIPOS DE SEGUROS'
    @ 10,20 SAY '3- RESULTADO DE VISITA A CLIENTE'
    @ 12,20 SAY '4- VENCIMENTOS DE NOTAS DE COBRANCA BANCARIA'
    @ 14,20 SAY '5- REGISTRO DE PRODUCAO'
    @ 16,20 SAY '6- VENCIMENTO DE CARNE DE FINANCIAMENTO'
    @ 18,20 SAY '7- RETORNAR AO MENU PRINCIPAL'
    @ 20,20 SAY '8- TERMINAR EXECUCAO'
    SET FORMAT TO SCREEN
    STORE ' ' TO XOPIN
    @ 22,55 SAY 'DIGITE SUA OPCAO ' ;
        GET XOPIN PICTURE 'X'
    READ
    CLEAR GETS
    DO CASE
        CASE XOPIN = '1'
            DO ISEGVGAP
        CASE XOPIN = '2'
            DO ISEGGER
        CASE XOPIN = '3'
            DO IVISITA
        CASE XOPIN = '4'
            DO IVENCNOT
        CASE XOPIN = '5'
            DO IREGPROD
        CASE XOPIN = '6'
            DO IVCTOCRN
        CASE XOPIN = '7'
            STORE 'N' TO XMENUINS
        CASE XOPIN = '8'
            STORE 'N' TO XMENUINS
            STORE 'N' TO XMENU
        OTHERWISE
            @ 23,55 SAY 'NAO EXISTE ESSA OPCAO'
            WAIT
    ENDCASE
ENDDO
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa ISEGVGAP.CMD

```

A)TYPE ISEGVGAP.CMD
SET FORMAT TO PRINT
SET TALK OFF
ERASE
EJECT
@ 1,7 SAY 'NOTA : '
@ 2,14 SAY 'A INSERCAO DE UM SEGURADO VG/AP CORRESPONDE A ATUALIZA-'
@ 4,14 SAY 'CAO DE VARIOS ARQUIVOS DESSA BASE DE DADOS, REFERENTES '
@ 6,14 SAY 'A FICHAS DE CADASTRAMENTO E CONTROLE DE PAGAMENTO, A'
@ 8,14 SAY 'SABER : FICHA DE CLIENTE, FICHA VG/AP, CARTAO CONSIGNA-'
@ 10,14 SAY 'CAO E RELACAO X.'
SET FORMAT TO SCREEN
STORE 'S' TO XREPETIR
DO WHILE XREPETIR = 'S'
    DO ULTIMAIN
    WAIT
    DO FCHGERAL
    DO CONSIG
    DO FCHVGAP
    DO RELACAOX
    USE B:SEGCLI.DBF
    APPEND BLANK
    REPLACE INSCRICAO WITH XNROINS,RAMO WITH 11,SEGURADORA WITH 'CIS'
    ERASE
    RELEASE ALL EXCEPT X*
    USE
    DO TELAFIM
ENDDO
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa CONSIG.CMD

```

A)TYPE CONSIG.CMD
ERASE
@ 1,5 SAY 'AS INSERCOES A SEGUIR CORRESPONDEM AO CARTAO CONSIGNACAO'
USE B:PASTACLI.DBF
APPEND BLANK
@ 3,0 SAY 'DIGITE AS SEGUINTE INFORMACOES : '
STORE 0 TO NROCERTVG
@ 5,0 SAY 'NUMERO CERTIFICADO VG';
GET NROCERTVG PICTURE '999999'
STORE 0 TO NROAPOLVG
@ 7,0 SAY 'NUMERO APOLICE VG';
GET NROAPOLVG PICTURE '9999999999999999'
STORE 0 TO NROCERTAP
@ 9,0 SAY 'NUMERO CERTIFICADO AP';
GET NROCERTAP PICTURE '999999'
STORE 0 TO NROAPOLAP
@ 11,0 SAY 'NUMERO APOLICE AP';
GET NROAPOLAP PICTURE '9999999999999999'
STORE ' ' TO SECAOTRAB
@ 13,0 SAY 'SECAO DE TRABALHO';
GET SECAOTRAB PICTURE 'XXXXXXXXXXXXXXXXXXXXX'
STORE ' ' TO ESTIP
@ 15,0 SAY 'ESTIPULANTE';
GET ESTIP PICTURE;
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
STORE ' ' TO NOMECONJ
@ 17,0 SAY 'NOME DO CONJUGE';
GET NOMECONJ PICTURE;
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
STORE 0 TO CERTVGCONJ
@ 19,0 SAY 'NRO CERTIFICADO VG DO CONJUGE';
GET CERTVGCONJ PICTURE '999999'
STORE 0 TO CERTAPCONJ
@ 21,0 SAY 'NRO CERTIFICADO AP DO CONJUGE';
GET CERTAPCONJ PICTURE '999999'
READ
CLEAR GETS
ERASE
  
```

```

@ 1,1 SAY 'CONTINUACAO CARTAO CONSIGNACAO'
STORE 0 TO CAPMORTEP
@ 5,0 SAY 'CAPITAL SEGURADO POR MORTE - PRINCIPAL';
GET CAPMORTEP PICTURE '9999999999'
STORE 0 TO CAPINVPERP
@ 7,0 SAY 'CAPITAL SEGURADO POR INV.PERMANENTE - PRINCIPAL';
GET CAPINVPERP PICTURE '9999999999'
STORE 0 TO CAPAMDSP
@ 9,0 SAY 'CAPITAL SEGURADO POR ASSIST.MED.DESP.SUPL. - PRINCIPAL';
GET CAPAMDSP PICTURE '9999999999'
STORE 0 TO CAPDHP
@ 11,0 SAY 'CAPITAL SEGURADO POR DESP.HOSP. - PRINCIPAL';
GET CAPDHP PICTURE '99999999'
STORE 0 TO CAPMORTEC
@ 13,0 SAY 'CAPITAL SEGURADO POR MORTE - CONJUGE';
GET CAPMORTEC PICTURE '9999999999'
STORE 0 TO CAPINVPERC
@ 15,0 SAY 'CAPITAL SEGURADO POR INV.PERMANENTE - CONJUGE';
GET CAPINVPERC PICTURE '9999999999'
STORE 0 TO CAPAMDSC
@ 17,0 SAY 'CAPITAL SEGURADO POR ASSIST.MED.DESP.SUPL. - CONJUGE';
GET CAPAMDSC PICTURE '9999999999'
STORE 0 TO CAPDHC
@ 19,0 SAY 'CAPITAL SEGURO POR DESP.HOSP. - CONJUGE';
GET CAPDHC PICTURE '99999999'
READ
CLEAR GETS
ERASE
@ 1,1 SAY 'CONTINUACAO CARTAO CONSIGNACAO'
STORE 0 TO DESCINIC
@ 5,0 SAY 'DESCONTO INICIAL (CR%)';
GET DESCINIC PICTURE '99999999'
STORE 0 TO DATADESC
@ 7,0 SAY 'DATA DO DESCONTO INICIAL (AAMMDD)';
GET DATADESC PICTURE '99999999'
STORE 0 TO DATADIA
@ 9,0 SAY 'DATA DO SEGURO (AAMMDD)';
GET DATADIA PICTURE '99999999'
STORE 0 TO SALARIO
@ 11,0 SAY 'SALARIO DO CLIENTE';
GET SALARIO PICTURE '9999999999'
STORE 0 TO ALTSALAR
@ 13,0 SAY 'ALTERACAO DE SALARIO';
GET ALTSALAR PICTURE '9999999999'
STORE 0 TO DATALTSAL
@ 15,0 SAY 'DATA DE ALTERACAO DE SALARIO (AAMMDD)';
GET DATALTSAL PICTURE '99999999'
READ
CLEAR GETS
REPLACE INSCRICAO WITH XNROINS,CERTIFVGNR WITH NROCERTVG,APOLIVGNR WITH ;
NROAPOLVG,CERTIFAPNR WITH NROCERTAP,APOLIAPCNR WITH NROAPOLAP,SECAO ;
WITH SECAOTRAB,ESTIPULANT WITH ESTIP,CONJUGE WITH NOMECONJ
REPLACE CERTIFCJVG WITH CERTVGCONJ,CERTIFCJAP WITH CERTAPCONJ,MORTEPR WITH;
CAPMORTEP,INVPERMPR WITH CAPINVPERP,AMDSPR WITH CAPAMDSP,DHPR WITH;
CAPDHP,MORTECJ WITH CAPMORTEC,INVPERMCJ WITH CAPINVPERC,AMDSCJ WITH;
CAPAMDSC
REPLACE DHCJ WITH CAPDHC,DESCINCR WITH DESCINIC,DESCINDT WITH DATADESC,DATA;
WITH DATADIA,VENCSTOS WITH SALARIO,ALTVENCTOS WITH ALTSALAR,ALTDATA;
WITH DATALTSAL
ERASE
RELEASE ALL EXCEPT X*
RETURN

```


Diagrama de Estrutura Modular e Listagem do Programa FCHVGAP.CMD

```

A)TYPE FCHVGAP.CMD
ERASE
@ 1,5 SAY 'AS INSERCOES A SEGUIR CORRESPONDEM A FICHA VG/AP'
USE B:SEGVGAP.DBF
APPEND BLANK
@ 3,0 SAY 'DIGITE AS SEGUINTE INFORMACOES : '
STORE ' ' TO CPF
@ 5,0 SAY 'NUMERO DO CPF';
GET CPF PICTURE 'XXXXXXXXXX'
STORE ' ' TO SUBEST
@ 7,0 SAY 'NOME DO SUB ESTIPULANTE';
GET SUBEST PICTURE 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
STORE 0 TO NASCDAT
@ 9,0 SAY 'DATA DE NASCIMENTO';
GET NASCDAT PICTURE '999999'
STORE ' ' TO ESTADOCIV
@ 11,0 SAY 'ESTADO CIVIL';
GET ESTADOCIV PICTURE 'X'
STORE ' ' TO SEXOCLI
@ 13,0 SAY 'SEXO';
GET SEXOCLI PICTURE 'X'
STORE ' ' TO LOCTRAB
@ 15,0 SAY 'LOCAL DE TRABALHO';
GET LOCTRAB PICTURE 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
STORE ' ' TO BNFCIARIOS
@ 17,0 SAY 'BENEFICIARIOS';
GET BNFCIARIOS PICTURE 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX';
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
STORE 0 TO COBMNCIS
@ 20,0 SAY 'CAPITAL SEGURO CIS - MORTE NATURAL';
GET COBMNCIS PICTURE '9999999999'
STORE 0 TO COBMACIS
@ 22,0 SAY 'CAPITAL SEGURO CIS - MORTE ACIDENTAL';
GET COBMACIS PICTURE '9999999999'
READ
CLEAR GETS
ERASE
  
```

```

@ 1,1 SAY 'CONTINUACAO FICHA VG/AP'
STORE 0 TO COBIPCIS
@ 5,0 SAY 'CAPITAL SEGURADO CIS - INV.PERMANENTE';
GET COBIPCIS PICTURE '9999999999'
STORE 0 TO COBAMDCSIS
@ 7,0 SAY 'CAPITAL SEGURADO CIS - ASSIST.MED.DESP.SUPL.';
GET COBAMDCSIS PICTURE '9999999999'
STORE 0 TO COBDHCIS
@ 9,0 SAY 'CAPITAL SEGURADO CIS - DESP.HOSP.';
GET COBDHCIS PICTURE '9999999999'
STORE 0 TO COBMNCSCS
@ 11,0 SAY 'CAPITAL SEGURADO CSCS - MORTE NATURAL';
GET COBMNCSCS PICTURE '9999999999'
STORE 0 TO COBMACSCS
@ 13,0 SAY 'CAPITAL SEGURADO CSCS - MORTE ACIDENTAL';
GET COBMACSCS PICTURE '9999999999'
STORE 0 TO COBIPCSCS
@ 15,0 SAY 'CAPITAL SEGURADO CSCS - INV.PERMANENTE';
GET COBIPCSCS PICTURE '9999999999'
STORE 0 TO CBAMDCSCS
@ 17,0 SAY 'CAPITAL SEGURADO CSCS - ASSIST.MED.DESP.SUPL.';
GET CBAMDCSCS PICTURE '9999999999'
STORE 0 TO COBDHCSCS
@ 19,0 SAY 'CAPITAL SEGURADO CSCS - DESP.HOSP.';
GET COBDHCSCS PICTURE '9999999999'
READ
CLEAR GETS
ERASE
@ 1,1 SAY 'CONTINUACAO FICHA VG/AP'
STORE 0 TO TOTMN
@ 5,0 SAY 'TOTAL SEGURADO POR MORTE NATURAL';
GET TOTMN PICTURE '9999999999'
STORE 0 TO TOTMA
@ 7,0 SAY 'TOTAL SEGURADO POR MORTE ACIDENTAL';
GET TOTMA PICTURE '9999999999'
STORE 0 TO TOTIP
@ 9,0 SAY 'TOTAL SEGURADO POR INV.PERMANENTE';
GET TOTIP PICTURE '9999999999'
STORE 0 TO TOTAMDS
@ 11,0 SAY 'TOTAL SEGURADO POR ASSIST.MED.DESP.SUPL.';
GET TOTAMDS PICTURE '9999999999'
STORE 0 TO TOTDH
@ 13,0 SAY 'TOTAL SEGURADO POR DESP.HOSP.';
GET TOTDH PICTURE '9999999999'
STORE 0 TO DATINICOB
@ 15,0 SAY 'DATA DE INICIO DE COBRANCA';
GET DATINICOB PICTURE '999999'
STORE 0 TO DATADIT
@ 17,0 SAY 'DATA EM QUE SOFREU ADITIVO';
GET DATADIT PICTURE '999999'
STORE 0 TO NRCERTMATR
@ 19,0 SAY 'NUMERO CERTIFICADO DE MATRICULA';
GET NRCERTMATR PICTURE '999999'
READ
CLEAR GETS
REPLACE INSCRICAO WITH XNROINS,CPFNR0 WITH CPF,SUBESTIP WITH SUBEST,DATANASC;
WITH NASCDAT,ESTCIVIL WITH ESTADOCIV,SEXO WITH SEXOCLI,LOCALTRAB;
WITH LOCTRAB,BENEFICI WITH BNFCIARIOS,MNCIS WITH COBMNCIS,MACIS WITH;
COBMACIS
REPLACE IPCIS WITH COBIPCIS,AMDCSIS WITH COBAMDCSIS,DHCIS WITH;
COBDHCIS,MNCSCS WITH COBMNCSCS,MACSCS WITH COBMACSCS,IPCSCS WITH;
COBIPCSCS,AMDCSCS WITH CBAMDCSCS,DHCSCS WITH COBDHCSCS,MNTOTAL;
WITH TOTMN
REPLACE MATOTAL WITH TOTMA,IPTOTAL WITH TOTIP,AMDSTOTAL WITH TOTAMDS,DHTOTAL;
WITH TOTDH,INICIOCOB WITH DATINICOB,ADITIVOSDT WITH;
DATADIT,CERTMATNR0 WITH NRCERTMATR
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa RELACAOX.CMD

A)TYPE RELACAOX.CMD

ERASE

@ 1,5 SAY 'AS INSERCOES A SEGUIR CORRESPONDEM AO ARQUIVO RELACAO X'

USE B:ARQRELX.DBF

APPEND BLANK

@ 3,0 SAY 'DIGITE AS SEGUINTE INFORMACOES :'

STORE 0 TO MENSALIDAD

@ 5,0 SAY 'MENSALIDADE';

GET MENSALIDAD PICTURE '99999999'

STORE ' ' TO MESI

@ 7,0 SAY 'MES INICIO DE PAGAMENTO';

GET MESI PICTURE 'XXX'

STORE ' ' TO MESF

@ 9,0 SAY 'MES FINAL DE PAGAMENTO';

GET MESF PICTURE 'XXX'

STORE ' ' TO MESREF

@ 11,0 SAY 'MES REFERENCIA DE PAGAMENTO';

GET MESREF PICTURE 'XXX'

READ

CLEAR GETS

REPLACE INSCRICAO WITH XNROINS, VALORMENSA WITH MENSALIDAD, MESINICIO WITH;
MESI, MESFINAL WITH MESF, MESREFER WITH MESREF.

RETURN

Diagrama de Estrutura Modular e Listagem do Programa ISEGGER.CMD

```

A)TYPE ISEGGGER.CMD
ERASE
SET FORMAT TO PRINT
EJECT
@ 1,7 SAY 'NOTA : '
@ 2,14 SAY 'A INSERCAO DE UM SEGURADO DE QUALQUER TIPO DE SEGURO,EXCETO VG/AP,'
@ 4,14 SAY 'CORRESPONDE A ATUALIZACAO DE ARQUIVOS DESSA BASE DE DADOS, REFE-'
@ 6,14 SAY 'RENTES A CADASTRAMENTO, A SABER : FICHA DE CLIENTE E RELACAO DE'
@ 8,14 SAY 'SEGUROS QUE O CLIENTE POSSUI.'
SET FORMAT TO SCREEN
ERASE
STORE 'S' TO XREPETIR
DO WHILE XREPETIR = 'S'
  DO ULTIMAIN
  WAIT
  DO FCHGERAL
  DO IMPRCODS
  @ 1,4 SAY 'AS INSERCOES A SEGUIR CORRESPONDEM A RELACAO DE SEGUROS QUE O'
  @ 1,66 SAY 'CLIENTE POSSUI'
  USE B:SEGCLI.DBF
  STORE 'S' TO NROSEG
  DO WHILE NROSEG = 'S'
    APPEND BLANK
    @ 3,0 SAY 'DIGITE AS SEGUINTE INFORMACOES : '
    STORE 0 TO CODSEG
    @ 5,0 SAY 'CODIGO DO TIPO DE SEGURO EFETUADO';
    GET CODSEG PICTURE '99'
    STORE 0 TO DATAVENC
    @ 7,0 SAY 'DATA DE VENCIMENTO DO SEGURO';
    GET DATAVENC PICTURE '999999'
    STORE 0 TO PREMIOSEG
    @ 9,0 SAY 'PREMIO DO SEGURO';
    GET PREMIOSEG PICTURE '9999999999'
    STORE ' ' TO SEG DORA
    @ 11,0 SAY 'SIGLA DA SEGURADORA';
    GET SEG DORA PICTURE 'XXXX'
  READ
  CLEAR GETS
  REPLACE INSCRICAO WITH XNROINS, RAMO WITH CODSEG, DATAVENC TO WITH;
  DATAVENC, PREMIO WITH PREMIOSEG, SEGURADORA WITH SEG DORA
  @ 20,35 SAY 'O CLIENTE FEZ MAIS ALGUM SEGURO ? (S/N)';
  GET NROSEG PICTURE 'X'
  READ
  CLEAR GETS
  ERASE
ENDDO
USE
DO TELAFIM
ENDDO
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa IVISITA.CMD

```

A)TYPE IVISITA.CMD
ERASE
SET FORMAT TO PRINT
EJECT
@ 1,7 SAY 'NOTA : '
@ 2,14 SAY 'AS INSERCOES DOS RESULTADOS OBTIDOS DE VISITAS A CLIENTES CORRES-'
@ 4,14 SAY 'PONDEM A ATUALIZACAO DE UM ARQUIVO QUE MANTEM INFORMACOES SOBRE '
@ 6,14 SAY 'HA QUANTO TEMPO FEZ-SE UM CONTATO COM UM CLIENTE.'
@ 22,1 SAY ' '
SET FORMAT TO SCREEN
WAIT
STORE 'S' TO XREPETIR
DO WHILE XREPETIR = 'S'
  DO PROCURA
    IF XNROINS <> 0
      DO IMPRCODV
      USE B:RESVISIT.DBF
      APPEND BLANK
      @ 1,0 SAY 'AS INSERCOES A SEGUIR CORRESPONDEM AO ARQUIVO DE RESULTADOS '
      @ 1,60 SAY 'DE VISITA A CLIENTES '
      STORE 0 TO DATAVISITA
      @ 5,1 SAY 'DATA DA VISITA';
      GET DATAVISITA PICTURE '999999'
      STORE ' ' TO CTATOVISIT
      @ 7,1 SAY 'FEZ CONTATO COM';
      GET CTATOVISIT PICTURE 'XXXXXXXXXX'
      STORE 0 TO RESULT
      @ 9,1 SAY 'CODIGO DO RESULTADO DA VISITA';
      GET RESULT PICTURE '99'
      READ
      CLEAR GETS
      REPLACE INSCRICAO WITH XNROINS,DATA WITH DATAVISITA,CONTATO WITH;
        CTATOVISIT,RESULTADO WITH RESULT
    ENDIF
    RELEASE ALL EXCEPT X*
    USE
    DO TELAFIM
  ENDDO
RETURN

```


Diagrama de Estrutura Modular e Listagem do Programa IMPRCOV.CMD

```

A)TYPE IMPRCODV.CMD
SET FORMAT TO PRINT
ERASE
EJECT
@ 1,10 SAY 'ATENCAO !...'
@ 3,10 SAY '          AGORA SERAO APRESENTADOS ALGUNS DOS POSSIVEIS RESULTA-'
@ 5,10 SAY 'DOS DE UMA VISITA A CLIENTE. CADA RESULTADO E ASSOCIADO A UM CODI-'
@ 7,10 SAY 'GO PARA EFEITO DE PROCESSAMENTO. VOCE DEVERA ANOTAR O CODIGO REFE-'
@ 9,10 SAY 'RENTE AO RESULTADO, PARA QUE POSSA FORNECER CORRETAMENTE AS INFOR-'
@ 11,10 SAY 'MACOES QUE LHE SERAO SOLICITADAS.'
SET FORMAT TO SCREEN
@ 22,1 SAY ' '
WAIT
ERASE
USE B:TABRESUL.DBF
SET FORMAT TO PRINT
DO WHILE .NOT. EOF
    EJECT
    @ 1,18 SAY 'CODIGO'
    @ 1,35 SAY 'RESULTADO DA VISITA'
    @ 2,18 SAY '====='
    @ 2,35 SAY '===== '
    STORE 5 TO CONT
    DO WHILE CONT <= 19 .AND. .NOT. EOF
        @ CONT,20 SAY RESULTADO
        @ CONT,35 SAY TIPORESUL
        STORE CONT+2 TO CONT
        SKIP
    ENDDO
    @ 21,1 SAY ' '
    WAIT
    ERASE
ENDDO
USE
RELEASE ALL EXCEPT X*
SET FORMAT TO SCREEN
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa IVENCNOT.CMD

```

A>TYPE IVENCNOT.CMD
ERASE
SET FORMAT TO PRINT
EJECT
@ 1,7 SAY 'NOTA : '
@ 2,14 SAY 'AS INSERCOES DE VENCIMENTOS DE NOTAS DE COBRANCA BANCARIA COR-'
@ 4,14 SAY 'RESPONDEM A ATUALIZACAO DE UM ARQUIVO QUE ATUA COMO UMA AGENDA,'
@ 6,14 SAY 'E QUE DEVE SER CONSULTADO DIARIAMENTE PARA CONTROLAR O PAGAMEN-'
@ 8,14 SAY 'TO DESSAS NOTAS.'
SET FORMAT TO SCREEN
@ 22,1 SAY ' '
WAIT
STORE 'S' TO XREPETIR
DO WHILE XREPETIR = 'S'
  DO PROCURA
  IF XNROINS <> 0
    USE B:DOCBANC.DBF
    APPEND BLANK
    @ 1,1 SAY 'AS INSERCOES A SEGUIR CORRESPONDEM AO ARQUIVO DE NOTAS EM'
    @ 1,59 SAY 'COBRANCA BANCARIA'
    STORE 0 TO EMISSAO
    @ 5,1 SAY 'DATA DE EMISSAO DA NOTA';
    GET EMISSAO PICTURE '999999'
    STORE 0 TO COBRANCA
    @ 7,1 SAY 'DATA MAXIMA PARA COBRANCA';
    GET COBRANCA PICTURE '999999'
    READ
    CLEAR GETS
    REPLACE INSCRICAO WITH XNROINS,DATEMISSAO WITH EMISSAO,DATCOBRANC WITH;
    COBRANCA
  ENDIF
  USE
  RELEASE ALL EXCEPT X*
  DO TELAFIM
ENDDO
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa IREGPROD.CMD

```

A)TYPE IREGPROD.CMD
ERASE
SET FORMAT TO PRINT
EJECT
@ 1,7 SAY 'NOTA : '
@ 2,14 SAY 'AS INSERCOES NO LIVRO DE REGISTRO DE PRODUCAO TEM A FINALIDADE DE '
@ 4,14 SAY 'AUXILIAR A CORRETORA A FAZER UMA ESTIMATIVA DE CRESCIMENTO PARA O '
@ 6,14 SAY 'PROXIMO ANO E TAMBEM, UMA COMPARACAO COM O CRESCIMENTO ALCANCADO '
@ 8,14 SAY 'NO ANO CORRENTE.'
SET FORMAT TO SCREEN
@ 22,1 SAY ' '
WAIT
STORE 'S' TO XREPETIR
DO WHILE XREPETIR = 'S'
  DO PROCURA
    IF XNROINS <> 0
      DO IMPRCODS
        USE B:LVRGPROD.DBF
        @ 1,5 SAY 'AS INSERCOES A SEGUIR CORRESPONDEM AO LIVRO DE REGISTRO DE '
        @ 1,65 SAY 'PRODUCAO '
        STORE ' ' TO NROPROP
        @ 3,1 SAY 'NUMERO DA PROPOSTA';
        GET NROPROP PICTURE 'XXXXXXXXXXXXXXXXXX'
        STORE 0 TO ANOATUAL
        @ 5,1 SAY 'ANO CORRENTE (NN)';
        GET ANOATUAL PICTURE '99'
        STORE ' ' TO SEGDORA
        @ 7,1 SAY 'SIGLA DA SEGURADORA';
        GET SEGDORA PICTURE 'XXXX'
        STORE 0 TO CODSEG
        @ 9,1 SAY 'CODIGO DO TIPO DE SEGURO EFETUADO';
        GET CODSEG PICTURE '99'
        STORE 0 TO NROAPOL
        @ 11,1 SAY 'NUMERO DA APOLICE';
        GET NROAPOL PICTURE '99999999999999'
        STORE 0 TO DATAEMISS
        @ 13,1 SAY 'DATA DE EMISSAO';
        GET DATAEMISS PICTURE '999999'
        STORE 0 TO DATAVENC
        @ 15,1 SAY 'DATA DE VENCIMENTO DO SEGURO';
        GET DATAVENC PICTURE '999999'
        STORE 0 TO IMPORTSEG
        @ 17,1 SAY 'IMPORTANCIA SEGURADA (CR$)';
        GET IMPORTSEG PICTURE '99999999999999'
        STORE 0 TO PREMIOSEG
        @ 19,1 SAY 'PREMIO DO SEGURO (CR$)';
        GET PREMIOSEG PICTURE '99999999999999'
        STORE 0 TO COMISSAO
        @ 21,1 SAY 'VALOR DA COMISSAO (CR$)';
        GET COMISSAO PICTURE '99999999999999'
      READ
      CLEAR GETS
      REPLACE INSCRICAO WITH XNROINS,PROPOSTANR WITH NROPROP,ANO WITH;
        ANOATUAL,SEGURADORA WITH SEGDORA,RAMO WITH CODSEG,APOLICENR WITH;
        NROAPOL,DATHEMISSAO WITH DATAEMISS,DATVENC TO WITH DATAVENC
      REPLACE IMPTSEGCR WITH IMPORTSEG,PREMIO WITH PREMIOSEG,VALORCOMIS WITH;
        COMISSAO
    ENDIF
  RELEASE ALL EXCEPT X*
  USE
  DO TELAFIM
ENDDO
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa IVCTOCR.N.CMD

```

A)TYPE IVCTOCR.N.CMD
ERASE
SET FORMAT TO PRINT
EJECT
@ 1,7 SAY 'NOTA : '
@ 2,14 SAY 'AS INSERCOES DOS VENCIMENTOS DE CARNES DE FINANCIAMENTO CORRES-'
@ 4,14 SAY 'PONDEM A ATUALIZACAO DE UM ARQUIVO QUE ATUA COMO UMA AGENDA, E'
@ 6,14 SAY 'QUE DEVE SER CONSULTADO DIARIAMENTE PARA CONTROLAR O PAGAMENTO.'
SET FORMAT TO SCREEN
@ 22,1 SAY ' '
WAIT
STORE 'S' TO XREPETIR
DO WHILE XREPETIR = 'S'
  DO PROCURA
  IF XNROINS <> 0
    USE B:AGENDA.DBF
    APPEND BLANK
    @ 1,0 SAY 'AS INSERCOES A SEGUIR CORRESPONDEM A AGENDA DE CONTROLE DE'
    @ 1,60 SAY 'VENCIMENTO DE CARNES'
    STORE 0 TO DIA
    @ 3,0 SAY 'DIA DO MES EM QUE VENCE A PARCELA DO CARNE';
    GET DIA PICTURE '99'
    STORE 0 TO VENCFINAL
    @ 5,0 SAY 'DATA DE ENCERRAMENTO DE PAGAMENTO DO CARNE';
    GET VENCFINAL PICTURE '999999'
    READ
    CLEAR GETS
    REPLACE INSCRICAO WITH XNROINS,DIAPGTO WITH DIA,DATENCERR WITH VENCFINAL
  ENDIF
  RELEASE ALL EXCEPT X*
  USE
  DO TELAFIM
ENDDO
RETURN

```


Diagrama de Estrutura Modular e Listagem do Programa RELATPAD.CMD

```

A)TYPE RELATPAD.CMD
SET TALK OFF
SET FORMAT TO PRINT
EJECT
STORE 'S' TO XMENUREL
DO WHILE XMENUREL = 'S'
    ERASE
    @ 0,25 SAY 'MENU DOS RELATORIOS PADROES'
    @ 1,25 SAY '=====
    @ 3,15 SAY ' 0- SEGURADOS POR TIPO DE SEGURO'
    @ 5,15 SAY ' 1- SEGURADOS E RESPECTIVOS SEGUROS'
    @ 7,15 SAY ' 2- NOTAS DE COBR. BANCARIA QUE VENCEM EM DETERMINADA DATA'
    @ 9,15 SAY ' 3- SEGUROS QUE EXPIRAM EM DETERMINADO MES'
    @ 11,15 SAY ' 4- PROPONENTES DE VG/AP POR ESTIPULANTE'
    @ 13,15 SAY ' 5- PRODUCAO ATE O MOMENTO'
    @ 15,15 SAY ' 6- CARNES A PAGAR NO DIA'
    @ 17,15 SAY ' 7- CAPITAL SEGURO DE UM CLIENTE VG/AP'
    @ 19,15 SAY ' 8- RETORNAR AO MENU PRINCIPAL'
    @ 21,15 SAY ' 9- TERMINAR EXECUCAO'
    SET FORMAT TO SCREEN
    STORE ' ' TO XOPREL
    @ 22,55 SAY 'DIGITE SUA OPCAO ' ;
        GET XOPREL PICTURE 'X'

    READ
    CLEAR GETS
    DO CASE
        CASE XOPREL = '0'
            DO SDOTPSRO
        CASE XOPREL = '1'
            DO SDOSSROS
        CASE XOPREL = '2'
            DO VENCNOTA
        CASE XOPREL = '3'
            DO EXPIRAR
        CASE XOPREL = '4'
            DO PROPESTI
        CASE XOPREL = '5'
            DO PRODUCAO
        CASE XOPREL = '6'
            DO PGTCARNE
        CASE XOPREL = '7'
            DO CAPITAL
        CASE XOPREL = '8'
            STORE 'N' TO XMENUREL
        CASE XOPREL = '9'
            STORE 'N' TO XMENUREL
            STORE 'N' TO XMENU
    OTHERWISE
        @ 23,55 SAY 'NAO EXISTE ESSA OPCAO'
        WAIT
    ENDCASE
ENDDO
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa SDOTPSRO.CMD

```

A)TYPE SDOTPSRO.CMD
ERASE
SET FORMAT TO PRINT
STORE 'S' TO XREPETIR
DO WHILE XREPETIR = 'S'
  ERASE
  EJECT
  @ 2,10 SAY 'ESTE PROGRAMA EMITE UM RELATORIO DE SEGURADOS POR TIPO DE'
  @ 4,10 SAY 'SEGURO, COM AS SEGUINTE ALTERNATIVAS : '
  @ 7,15 SAY '1- SEGURADOS PARA CADA TIPO DE SEGURO EXISTENTE'
  @ 9,15 SAY '2- SEGURADOS PARA UM TIPO ESPECIFICO DE SEGURO'
  SET FORMAT TO SCREEN
  STORE ' ' TO ALTER
  @ 18,50 SAY 'DIGITE SUA OPCAO';
  GET ALTER PICTURE 'X'

  READ
  CLEAR GETS
  DO CASE
    CASE ALTER = '1'
      SELECT SECONDARY
      USE B:RAMOSEG.DBF
      INDEX ON TIPOSEG TO B:INDTIPOS
      USE B:RAMOSEG.DBF INDEX B:INDTIPOS
      COUNT TO NRREG
      GOTO TOP
      STORE 1 TO N
      DO WHILE N <= NRREG
        STORE 1 TO CONT
        DO WHILE CONT < N
          SKIP
          STORE CONT+1 TO CONT
        ENDDO
        STORE RAMO TO PONTEIRO
        STORE TIPOSEG TO TIPO
        DO IMPRSGDO
        STORE N+1 TO N
        SELECT SECONDARY
        USE B:RAMOSEG.DBF INDEX B:INDTIPOS
      ENDDO
    CASE ALTER = '2'
      DO IMPRCODS
      STORE 0 TO PONTEIRO
      @ 5,10 SAY 'CODIGO DO SEGURO';
      GET PONTEIRO PICTURE '99'

      READ
      CLEAR GETS
      SELECT SECONDARY
      USE B:RAMOSEG.DBF
      LOCATE FOR RAMO = PONTEIRO
      STORE TIPOSEG TO TIPO
      DO IMPRSGDO
    OTHERWISE
      @ 20,1 SAY 'NAO EXISTE ESSA OPCAO'
      WAIT
  ENDCASE
  ERASE
  USE
  RELEASE ALL EXCEPT X*
  DO TELAFIM
ENDDO
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa IMPRSGDQ.CMD

```

A>TYPE IMPRSGDO.CMD
SELECT SECONDARY
USE B:FICHACLI.DBF
SELECT PRIMARY
USE B:SEGCLI.DBF
LOCATE FOR RAMO = PONTEIRO
IF .NOT. EOF
    STORE 'S' TO NOVAPAG
    DO WHILE NOVAPAG = 'S'
        ERASE
        SET FORMAT TO PRINT
        EJECT
        @ 1,20 SAY 'RELATORIO DE SEGURADOS POR RAMO DE SEGURO'
        @ 1,70 SAY DATE()
        @ 2,20 SAY '-----'
        @ 5,1 SAY 'RAMO : '
        @ 5,8 SAY TIPO USING 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
        @ 6,1 SAY '----'
        @ 7,13 SAY 'SEGURADO'
        @ 7,57 SAY 'INSCRICAO'
        @ 8,13 SAY '-----'
        @ 8,57 SAY '-----'
        STORE 10 TO LIN
        DO WHILE .NOT. EOF .AND. LIN <= 78
            STORE INSCRICAO TO XNROINS
            SELECT SECONDARY
            GOTO TOP
            DO WHILE INSCRICAO (>) XNROINS
                SKIP
            ENDDO
            @ LIN,5 SAY NOMCLIENTE USING:
                'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
            @ LIN,58 SAY INSCRICAO USING '9999999'
            STORE LIN+2 TO LIN
            SELECT PRIMARY
            CONTINUE
        ENDDO
        IF EOF
            STORE 'N' TO NOVAPAG
        ENDIF
        SET FORMAT TO SCREEN
        @ 22,1 SAY ' '
        WAIT
    ENDDO
ELSE
    ERASE
    SET FORMAT TO PRINT
    EJECT
    @ 5,10 SAY 'PARA O TIPO DE SEGURO : '
    @ 5,34 SAY TIPO USING 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
    @ 7,10 SAY 'NAO EXISTE NENHUM SEGURADO'
    SET FORMAT TO SCREEN
    @ 22,1 SAY ' '
    WAIT
ENDIF
USE
SELECT SECONDARY
USE
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa SDOSSROS.CMD

```

A>TYPE SDOSSROS.CMD
ERASE
SET FORMAT TO PRINT
STORE 'S' TO XREPETIR
DO WHILE XREPETIR = 'S'
  ERASE
  EJECT
  @ 2,10 SAY 'ESTE PROGRAMA EMITE UM RELATORIO DE SEGUROS QUE UM SEGURADO'
  @ 4,10 SAY 'POSSUI, COM AS SEGUINTE ALTERNATIVAS : '
  @ 7,15 SAY '1- RELACAO DE SEGUROS PARA TODOS OS SEGURADOS DA CORRETORA'
  @ 9,15 SAY '2- RELACAO DE SEGUROS PARA UM SEGURADO ESPECIFICO'
  SET FORMAT TO SCREEN
  STORE ' ' TO ALTER
  @ 18,50 SAY 'DIGITE SUA OPCAO';
  GET ALTER PICTURE 'X'
  READ
  CLEAR GETS
  DO CASE
    CASE ALTER = '1'
      SELECT SECONDARY
      USE B:FICHACLI.DBF
      INDEX ON NOMCLIENTE TO B:INDNOME
      USE B:FICHACLI.DBF INDEX B:INDNOME
      COUNT TO NRORIG
      GOTO TOP
      STORE 1 TO N
      DO WHILE N <= NRORIG
        STORE 1 TO CONT
        DO WHILE CONT < N
          SKIP
          STORE CONT+1 TO CONT
        ENDDO
        STORE INSCRICAO TO XNROINS
        STORE NOMCLIENTE TO NOMECLI
        DO IMPRSGRO
        STORE N+1 TO N
        SELECT SECONDARY
        USE B:FICHACLI.DBF INDEX B:INDNOME
      ENDDO
    CASE ALTER = '2'
      SELECT SECONDARY
      DO PROCURA
      IF XNROINS <> 0
        DO IMPRSGRO
      ENDIF
    OTHERWISE
      @ 20,1 SAY 'NAO EXISTE ESSA OPCAO'
      WAIT
  ENDCASE
  ERASE
  USE
  RELEASE ALL EXCEPT X*
  DO TELAFIM
ENDDO
RETURN

```


Diagrama de Estrutura Modular e Listagem do Programa IMPRSGRO.CMD

```

A)TYPE IMPRSGRO.CMD
SELECT SECONDARY
USE B:RAMOSEG.DBF
SELECT PRIMARY
USE B:SEGCLI.DBF
LOCATE FOR INSCRICAO = XNROINS
IF .NOT. EOF
    STORE 'S' TO NOVAPAG
    DO WHILE NOVAPAG = 'S'
        ERASE
        SET FORMAT TO PRINT
        EJECT
        @ 1,20 SAY 'RELATORIO DE SEGUROS DE UM CLIENTE'
        @ 1,70 SAY DATE()
        @ 2,20 SAY '-----'
        @ 5,1 SAY 'NOME : '
        @ 5,8 SAY NOMECLI USING:
            'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
        @ 6,1 SAY '-----'
        @ 8,8 SAY 'SEGUROS : '
        STORE 10 TO LIN
        DO WHILE .NOT. EOF .AND. LIN <=78
            STORE RAMO TO CODSEG
            SELECT SECONDARY
            GOTO TOP
            DO WHILE RAMO <> CODSEG
                SKIP
            ENDDO
            @ LIN,18 SAY TIPOSEG USING 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
            STORE LIN+2 TO LIN
            SELECT PRIMARY
            CONTINUE
        ENDDO
        IF EOF
            STORE 'N' TO NOVAPAG
        ENDIF
        SET FORMAT TO SCREEN
        @ 22,1 SAY ' '
        WAIT
    ENDDO
ELSE
    ERASE
    SET FORMAT TO PRINT
    EJECT
    @ 5,5 SAY 'ATUALMENTE, O CLIENTE '
    @ 5,27 SAY NOMECLI USING 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
    @ 6,27 SAY '-----'
    @ 8,5 SAY 'NAO POSSUI NENHUM TIPO DE SEGURO'
    SET FORMAT TO SCREEN
    @ 22,1 SAY ' '
    WAIT
ENDIF
USE
SELECT SECONDARY
USE
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa VENCNOTA.CMD

```

A)TYPE VENCNOTA.CMD
SET FORMAT TO PRINT
SELECT SECONDARY
USE B:FICHACLI.DBF
SELECT PRIMARY
USE B:DOCBANC.DBF
ERASE
EJECT
@ 2,10 SAY 'ESTE PROGRAMA EMITE UM RELATORIO DAS NOTAS EM COBRANCA BANCARIA'
@ 4,10 SAY 'QUE VENCEM NA DATA QUE SERA SOLICITADA A SEGUIR.'
SET FORMAT TO SCREEN
STORE 0 TO DATA
@ 14,10 SAY 'DIGITE A DATA (DDMMAA)';
      GET DATA PICTURE '999999'

READ
CLEAR GETS
STORE DATA TO DATAUX
STORE $(STR(DATAUX,6),1,2) + '/' + $(STR(DATAUX,6),3,2) + '/' + ;
      $(STR(DATAUX,6),5,2) TO DATAREL
LOCATE FOR DATCOBRANC = DATA
IF .NOT. EOF
  STORE 'S' TO NOVAPAG
  DO WHILE NOVAPAG.= 'S'
    ERASE
    SET FORMAT TO PRINT
    EJECT
    @ 1,10 SAY 'RELATORIO DE NOTAS EM COBRANCA BANCARIA QUE VENCEM EM : '
    @ 2,10 SAY '-----'
    @ 4,28 SAY 'DATA : '
    @ 4,34 SAY DATAREL USING 'XXXXXXXX'
    @ 5,28 SAY '-----'
    @ 7,1 SAY 'REFERENTES AOS SEGUINTE CLIENTES : '
    @ 9,20 SAY 'SEGUADO '
    @ 9,55 SAY 'INSCRICAO '
    @ 10,20 SAY '-----'
    @ 10,55 SAY '-----'
    STORE 12 TO LIN
    DO WHILE .NOT. EOF .AND. LIN <= 78
      STORE INSCRICAO TO XNROINS
      SELECT SECONDARY
      GOTO TOP
      DO WHILE INSCRICAO <> XNROINS
        SKIP
      ENDDO
      @ LIN,5 SAY NOMCLIENTE USING;
        'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
      @ LIN,58 SAY INSCRICAO USING '9999999'
      STORE LIN+2 TO LIN
      SELECT PRIMARY
      CONTINUE
    ENDDO
    IF EOF
      STORE 'N' TO NOVAPAG
    ENDIF
    SET FORMAT TO SCREEN
    @ 22,1 SAY ' '
    WAIT
  ENDDO
ELSE
  ERASE
  SET FORMAT TO PRINT
  EJECT
  @ 5,32 SAY 'NA DATA : '
  @ 5,42 SAY DATAREL USING 'XXXXXXXX'
  @ 6,42 SAY '-----'
  @ 10,15 SAY 'NAO EXISTEM NOTAS EM COBRANCA BANCARIA EM VENCIMENTO'
  SET FORMAT TO SCREEN.
  @ 22,1 SAY ' '
  WAIT
ENDIF
USE
SELECT SECONDARY
USE
RELEASE ALL EXCEPT X*
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa EXPIRAR.CMD

```

A) TYPE EXPIRAR.CMD
SET FORMAT TO PRINT
EJECT
ERASE
@ 2,10 SAY 'ESTE PROGRAMA EMITE UM RELATORIO DOS SEGUROS QUE EXPIRAM NUM'
@ 4,10 SAY 'DETERMINADO MES. ELE DEVE SER SOLICITADO NOS MEADOS DE UM MES.'
@ 6,10 SAY 'COM REFERENCIA AO MES SEGUINTE, PARA QUE OS CLIENTES POSSAM'
@ 8,10 SAY 'SER COMUNICADOS E TENHAM TEMPO PARA FAZER A RENOVACAO.'
SET FORMAT TO SCREEN
STORE ' ' TO MESANO
@ 18,40 SAY 'DIGITE O MES E O ANO (MMAA)';
      GET MESANO PICTURE 'XXXX'

READ
CLEAR GETS
SELECT PRIMARY
USE B:SEGCLI.DBF
LOCATE FOR (%(STR(DATVENCTO,6),3,4)) = '&MESANO'
IF .NOT.EOF
  STORE 'S' TO NOVAPAG
  DO WHILE NOVAPAG = 'S'
    ERASE
    SET FORMAT TO PRINT
    EJECT
    @ 1,15 SAY 'RELATORIO DOS SEGUROS QUE EXPIRAM NO MES : '
    @ 1,58 SAY MESANO USING 'XX'
    @ 1,70 SAY DATE() USING 'XXXXXXXX'
    @ 2,15 SAY '-----'
    @ 4,1 SAY 'REFERENTES AOS SEGUINTE CLIENTES : '
    @ 6,1 SAY 'INSCRICAO'
    @ 6,25 SAY 'SEGURADO'
    @ 6,48 SAY 'RAMO DO SEGURO'
    @ 6,69 SAY 'DIA VENCTO'
    @ 7,1 SAY '-----'
    @ 7,25 SAY '-----'
    @ 7,48 SAY '-----'
    @ 7,69 SAY '-----'
    STORE 10 TO LIN
    DO WHILE .NOT. EOF .AND. LIN <= 78
      STORE INSCRICAO TO XNROINS
      STORE RAMO TO CODSEG
      STORE (%(STR(DATVENCTO,6),1,2)) TO DIA
      SELECT SECONDARY
      USE B:FICHACLI.DBF
      DO WHILE INSCRICAO <> XNROINS
        SKIP
      ENDDO
      STORE NOMCLIENTE TO NOMECLI
      SELECT SECONDARY
      USE B:RAMOSEG.DBF
      DO WHILE RAMO <> CODSEG
        SKIP
      ENDDO
      @ LIN,2 SAY INSCRICAO USING '9999999'
      @ LIN,15 SAY NOMECLI USING 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
      STORE LIN+1 TO LIN
      @ LIN,40 SAY TIPOSEG USING 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
      @ LIN,75 SAY DIA USING 'XX'
      STORE LIN+2 TO LIN
      SELECT PRIMARY
      CONTINUE
    ENDDO
    IF EOF
      STORE 'N' TO NOVAPAG
    ENDIF
    SET FORMAT TO SCREEN
    @ 22,1 SAY ' '
    WAIT
  ENDDO
ELSE
  ERASE
  SET FORMAT TO PRINT
  EJECT
  @ 5,72 SAY DATE() USING 'XXXXXXXX'
  @ 10,19 SAY 'NAO EXISTEM SEGUROS QUE EXPIRAM NO MES : '
  @ 10,60 SAY MESANO USING 'XX'
  @ 11,60 SAY '== '
  SET FORMAT TO SCREEN
  @ 22,1 SAY ' '
  WAIT
ENDIF
USE
SELECT SECONDARY
USE
RELEASE ALL EXCEPT X*
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa PRUPESTI.CMD

```

A>TYPE PROPESTI.CMD
SET FORMAT TO PRINT
ERASE
STORE 'S' TO XREPETIR
DO WHILE XREPETIR = 'S'
    ERASE
    EJECT
    @ 2,10 SAY 'ESTE PROGRAMA EMITE UM RELATORIO DE PROPONENTES DO SEGURO VG/AP,'
    @ 4,10 SAY 'POR ESTIPULANTE, COM AS SEGUINTE ALTERNATIVAS : '
    @ 7,15 SAY '1- RELACAO DE PROPONENTES PARA TODOS OS ESTIPULANTES'
    @ 9,15 SAY '2- RELACAO DE PROPONENTES PARA UM ESTIPULANTE ESPECIFICO'
    SET FORMAT TO SCREEN
    STORE ' ' TO ALTER
    @ 18,50 SAY 'DIGITE SUA OPCAO';
        GET ALTER PICTURE 'X'

    READ
    CLEAR GETS
    SELECT PRIMARY
    USE B:PASTACLI.DBF
    INDEX ON ESTIPULANT TO B:INDESTIP
    USE B:PASTACLI.DBF INDEX B:INDESTIP
    DO CASE
        CASE ALTER = '1'
            DO IMPRPROP
        CASE ALTER = '2'
            ERASE
            STORE ' ' TO PONTEIRO
            @ 10,5 SAY 'NOME DO ESTIPULANTE';
                GET PONTEIRO PICTURE;
                    'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'

            READ
            CLEAR GETS
            LOCATE FOR ESTIPULANT = '&PONTEIRO'
            DO IMPRPROP
        OTHERWISE
            @ 20,10 SAY 'NAO EXISTE ESSA OPCAO'
            WAIT
    ENDCASE
    ERASE
    USE
    RELEASE ALL EXCEPT X*
    DO TELAFIM
ENDDO
RETURN

```


Diagrama de Estrutura Modular e Listagem do Programa IMPRPROP.COMD

```

A>TYPE IMPRPROP.CMD
SET FORMAT TO PRINT
SELECT SECONDARY
USE B:FICHACLI.DBF
STORE 'S' TO NOVAPAG
DO WHILE NOVAPAG = 'S'
    ERASE
    EJECT
    @ 1,20 SAY 'RELATORIO DE PROPONENTES POR ESTIPULANTE'
    @ 1,70 SAY DATE() USING 'XXXXXXX'
    @ 2,20 SAY '-----'
    @ 4,1 SAY 'ESTIPULANTE : '
    @ 4,15 SAY ESTIPULANT USING;
        'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
    @ 6,13 SAY 'PROONENTE'
    @ 6,57 SAY 'INSCRICAO'
    @ 7,13 SAY '-----'
    @ 7,57 SAY '-----'
    STORE 10 TO LIN
    STORE ESTIPULANT TO PONTEIRO
    DO WHILE .NOT. EOF .AND. ESTIPULANT = '&PONTEIRO' .AND. LIN <= 78
        STORE INSCRICAO TO XNROINS
        SELECT SECONDARY
        GOTO TOP
        LOCATE FOR INSCRICAO = XNROINS
        @ LIN,5 SAY NOMCLIENTE USING;
            'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
        @ LIN,58 SAY INSCRICAO USING '9999999'
        STORE LIN+2 TO LIN
        SELECT PRIMARY
        SKIP
    ENDDO
    IF ESTIPULANT <> '&PONTEIRO' .OR. LIN > 78 .OR. EOF
        @ 22,1 SAY ' '
        WAIT
    ENDIF
    IF EOF .OR. (ESTIPULANT <> '&PONTEIRO' .AND. ALTER = '2')
        STORE 'N' TO NOVAPAG
    ENDIF
ENDDO
USE
SELECT SECONDARY
USE
SET FORMAT TO SCREEN
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa PRODUCAO.CMD

```

A>TYPE PRODUCAO.CMD
SET FORMAT TO PRINT
EJECT
ERASE
@ 10,10 SAY 'ESTE PROGRAMA EMITE UM RELATORIO COM AS PORCENTAGENS DE'
@ 12,10 SAY 'PRODUCAO, POR RAMO DE SEGURO, ATE O MOMENTO'
SET FORMAT TO SCREEN
STORE 0 TO ANOREF
@ 18,38 SAY 'DIGITE O ANO DO QUAL DESEJA PRODUCAO';
    GET ANOREF PICTURE '99'

READ
CLEAR GETS
@ 22,1 SAY ' '
WAIT
ERASE
SET FORMAT TO PRINT
EJECT
@ 1,1 SAY 'RELATORIO DOS PERCENTUAIS DE PRODUCAO ATE O MOMENTO - ANO REF : '
@ 1,65 SAY ANOREF USING '99'
@ 1,70 SAY DATE() USING 'XXXXXXXX'
@ 2,1 SAY '-----'
SELECT PRIMARY
USE B:LVRGPROD.DBF
COUNT FOR ANO = ANOREF TO TOTAL
IF TOTAL <> 0
    @ 4,5 SAY 'TOTAL DE SEGUROS EFETUADOS = '
    @ 4,35 SAY TOTAL PICTURE '99999'
    @ 5,5 SAY '-----'
    @ 7,15 SAY 'RAMO DO SEGURO'
    @ 7,42 SAY 'PORCENTAGEM'
    @ 8,15 SAY '-----'
    @ 8,42 SAY '-----'
    STORE 10 TO LIN
    SELECT SECONDARY
    USE B:RAMOSEG.DBF
    INDEX ON TIPOSEG TO B:INDTIPOS
    USE B:RAMOSEG.DBF INDEX B:INDTIPOS
    DO WHILE .NOT. EOF
        @ LIN,10 SAY TIPOSEG USING 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
        STORE RAMO TO PONTEIRO
        SELECT PRIMARY
        COUNT FOR (RAMO = PONTEIRO .AND. ANO = ANOREF) TO SUBTOT
        STORE INT((SUBTOT/TOTAL)*1000) TO PORCENT
        @ LIN,45 SAY PORCENT USING '99,9'
        @ LIN,50 SAY '%'
        STORE LIN+2 TO LIN
        SELECT SECONDARY
        SKIP
        IF RAMO = 11
            SKIP
        ENDIF
    ENDDO
    RELEASE ALL EXCEPT X*
    USE
    SELECT PRIMARY
    USE
ELSE
    @ 10,10 SAY 'NAO FOI REALIZADO NENHUM SEGURO NESSE ANO'
ENDIF
SET FORMAT TO SCREEN
@ 22,1 SAY ' '
WAIT
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa PGTCARNE.CMD

```

A>TYPE PGTCARNE.CMD
SET FORMAT TO PRINT
SELECT SECONDARY
USE B:FICHACLI.DBF
SELECT PRIMARY
USE B:AGENDA.DBF
ERASE
EJECT
@ 2,10 SAY 'ESTE PROGRAMA EMITE UM RELATORIO DOS CLIENTES CUJAS PARCELAS'
@ 4,10 SAY 'DOS CARNES VENCEM NO DIA QUE SERA SOLICITADO COMO ENTRADA,'
@ 6,10 SAY 'EM SEGUIDA.'
SET FORMAT TO SCREEN
STORE 0 TO DIA
@ 15,50 SAY 'DIGITE O DIA (DD)';
      GET DIA PICTURE '99'
READ
CLEAR GETS
LOCATE FOR DIAPGT0 = DIA
IF .NOT. EOF
  STORE 'S' TO NOVAPAG
  DO WHILE NOVAPAG = 'S'
    ERASE
    SET FORMAT TO PRINT
    EJECT
    @ 1,5 SAY 'RELATORIO DOS CARNES CUJAS PARCELAS VENCEM NO DIA : '
    @ 1,57 SAY DIA USING '99'
    @ 1,72 SAY DATE() USING 'XXXXXXXX'
    @ 2,5 SAY '-----'
    @ 4,1 SAY 'REFERENTES AOS SEGUINTE CLIENTES : '
    @ 6,20 SAY 'SEGURADO'
    @ 6,55 SAY 'INSCRICAO'
    @ 7,20 SAY '-----'
    @ 7,55 SAY '-----'
    STORE 9 TO LIN
    DO WHILE .NOT. EOF .AND. LIN <= 78
      STORE INSCRICAO TO XNROINS
      SELECT SECONDARY
      GOTO TOP
      DO WHILE INSCRICAO <> XNROINS
        SKIP
      ENDDO
      @ LIN,5 SAY NOMCLIENTE USING;
        'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
      @ LIN,58 SAY INSCRICAO USING '9999999'
      STORE LIN+2 TO LIN
      SELECT PRIMARY
      CONTINUE
    ENDDO
    IF EOF
      STORE 'N' TO NOVAPAG
    ENDIF
    SET FORMAT TO SCREEN
    @ 22,1 SAY ' '
    WAIT
  ENDDO
ELSE
  ERASE
  SET FORMAT TO PRINT
  EJECT
  @ 3,38 SAY 'NO DIA : '
  @ 3,47 SAY DIA USING '99'
  @ 4,38 SAY '-----'
  @ 8,20 SAY 'NAO EXISTEM PARCELAS DE CARNES EM VENCIMENTO'
  SET FORMAT TO SCREEN
  @ 22,1 SAY ' '
  WAIT
ENDIF
USE
SELECT SECONDARY
USE
RELEASE ALL EXCEPT X*
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa CAPITAL.CMD

```

A)TYPE CAPITAL.CMD
SET FORMAT TO PRINT
STORE 'S' TO XREPETIR
DO WHILE XREPETIR = 'S'
    ERASE
    EJECT
    @ 2,10 SAY 'ESTE PROGRAMA EMITE UM RELATORIO DE SEGURADOS DE VIDA EM GRUPO'
    @ 4,10 SAY 'E ACIDENTES PESSOAIS, COM OS CAPITAIS REFERENTES A CADA UMA DAS'
    @ 6,10 SAY 'MODALIDADES DE SEGUROS PERMITIDAS POR ESSE RAMO. EXISTEM AS SE-'
    @ 8,10 SAY 'GUINTE ALTERNATIVAS : '
    @ 11,15 SAY '1- RELACAO DE CAPITAIS DE TODOS OS SEGURADOS DE VG/AP'
    @ 13,15 SAY '2- RELACAO DE CAPITAIS DE UM SEGURADO ESPECIFICO DE VG/AP'
    SET FORMAT TO SCREEN
    STORE ' ' TO ALTER
    @ 20,50 SAY 'DIGITE SUA OPCAO';
    GET ALTER PICTURE 'X'

    READ
    CLEAR GETS
    SELECT PRIMARY
    USE B:PASTACLI.DBF
    DO CASE
        CASE ALTER = '1'
            DO WHILE .NOT. EOF
                DO IMPRCAP
                SKIP
            ENDDO
        CASE ALTER = '2'
            SELECT SECONDARY
            DO PROCURA
            IF XNROINS <> 0
                DO IMPRCAP
            ENDIF
        OTHERWISE
            @ 22,50 SAY 'NAO EXISTE ESSA OPCAO'
            WAIT
    ENDCASE
    ERASE
    USE
    RELEASE ALL EXCEPT X*
    DO TELAFIM
ENDDO
RETURN

```


Diagrama de Estrutura Modular e Listagem do Programa IMPRCAP.CMD

```

A)TYPE IMPRCAP.CMD
SET FORMAT TO PRINT
ERASE
EJECT
@ 1,15 SAY 'RELATORIO DE CAPITAIS SEGURADOS EM VG/AP'
@ 1,70 SAY DATE() USING 'XXXXXXXX'
@ 2,15 SAY '-----'
@ 4,0 SAY 'PRINCIPAL : '
IF ALTER = '1'
  STORE INSCRICAO TO XNROINS
  SELECT SECONDARY
  USE B:FICHACLI.DBF
  LOCATE FOR INSCRICAO = XNROINS
  STORE NOMCLIENTE TO NOMECLI
  USE
ENDIF
@ 4,12 SAY NOMECLI USING 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
@ 4,63 SAY 'INSCR : '
@ 4,71 SAY XNROINS USING '9999999'
@ 5,0 SAY '-----'
@ 5,63 SAY '-----'
SELECT PRIMARY
IF ALTER = '2'
  LOCATE FOR INSCRICAO = XNROINS
ENDIF
@ 7,5 SAY 'MORTE : CR$'
@ 7,32 SAY MORTEPR USING '9,999,999,999'
@ 8,5 SAY 'INV.PERMANENTE : CR$'
@ 8,32 SAY INVPERMPR USING '9,999,999,999'
@ 9,5 SAY 'ASSIST.MED.DESP.SUPL. : CR$'
@ 9,32 SAY AMDSPR USING '9,999,999,999'
@ 10,5 SAY 'DESP.HOSPITALARES : CR$'
@ 10,36 SAY DHPR USING '9,999,999'
IF MORTECJ <> 0
  @ 13,0 SAY 'CONJUGE : '
  @ 13,10 SAY CONJUGE USING:
    'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @ 14,0 SAY '-----'
  @ 16,5 SAY 'MORTE : CR$'
  @ 16,32 SAY MORTECJ USING '9,999,999,999'
  @ 17,5 SAY 'INV.PERMANENTE : CR$'
  @ 17,32 SAY INVPERMCJ USING '9,999,999,999'
  @ 18,5 SAY 'ASSIST.MED.DESP.SUPL. : CR$'
  @ 18,32 SAY AMDSCJ USING '9,999,999,999'
  @ 19,5 SAY 'DESP.HOSPITALARES : CR$'
  @ 19,36 SAY DHCJ USING '9,999,999'
ELSE
  @ 13,1 SAY 'NAO EXISTEM CAPITAIS SEGURADOS PARA CONJUGE'
ENDIF
SET FORMAT TO SCREEN
@ 22,1 SAY ' '
WAIT
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa DELICAO.CMD

```

A>TYPE DELICAO.CMD
ERASE
SET TALK OFF
SET FORMAT TO PRINT
EJECT
STORE 'S' TO XMENUDEL
DO WHILE XMENUDEL = 'S'
    @ 3,25 SAY 'MENU DAS OPERACOES DE DELICAO'
    @ 4,25 SAY '===== '
    @ 7,7 SAY 'DELICAO DE : '
    @ 9,21 SAY '1- NOTAS DE COBR. BANCARIA PAGAS'
    @ 11,21 SAY '2- REGISTRO DE CARNE DA AGENDA'
    @ 13,21 SAY '3- RETORNAR AO MENU PRINCIPAL'
    @ 15,21 SAY '4- TERMINAR EXECUCAO'
    SET FORMAT TO SCREEN
    STORE ' ' TO XOPDEL
    @ 18,55 SAY 'DIGITE SUA OPCAO ' ;
        GET XOPDEL PICTURE 'X'
    READ
    CLEAR GETS
    DO CASE
        CASE XOPDEL = '1'
            DO DNOTA
        CASE XOPDEL = '2'
            DO DCARNEAG
        CASE XOPDEL = '3'
            STORE 'N' TO XMENUDEL
        CASE XOPDEL = '4'
            STORE 'N' TO XMENUDEL
            STORE 'N' TO XMENU
        OTHERWISE
            @ 22,55 SAY 'NAO EXISTE ESSA OPCAO'
            WAIT
    ENDCASE
ENDDO
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa DNOTA.CMD

```

A)TYPE DNOTA.CMD
SET FORMAT TO PRINT
EJECT
ERASE
@ 2,10 SAY 'ESTE PROGRAMA FAZ UMA DELICAO NO ARQUIVO QUE CONTROLA O'
@ 4,10 SAY 'VENCIMENTO DE NOTAS EM COBRANCA BANCARIA,REFERENTE A NOTA'
@ 6,10 SAY 'QUE JA FOI PAGA.'
SET FORMAT TO SCREEN
@ 22,1 SAY ' '
WAIT
STORE 'S' TO XREPETIR
DO WHILE XREPETIR = 'S'
    DO PROCURA
    IF XNROINS <> 0
        STORE 0 TO DATAVENC
        @ 12,5 SAY 'DATA DE VENCIMENTO DA NOTA (DDMMAA)';
        GET DATAVENC PICTURE '999999'

        READ
        CLEAR GETS
        USE B:DOCBANC.DBF
        LOCATE FOR INSCRICAO = XNROINS .AND. DATCOBRANC = DATAVENC
        IF .NOT. EOF
            ERASE
            SET FORMAT TO PRINT
            EJECT
            @ 5,15 SAY 'SERA ELIMINADO DO ARQUIVO O REGISTRO REFERENTE AO'
            @ 6,15 SAY '-----'
            @ 10,1 SAY 'CLIENTE : '
            @ 10,11 SAY NOMECLI USING;
            'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
            @ 10,64 SAY 'INSCR : '
            @ 10,72 SAY XNROINS USING '9999999'
            @ 11,1 SAY '-----'
            @ 11,64 SAY '-----'
            @ 13,11 SAY 'DATA DE EMISSAO DA NOTA : '
            @ 13,37 SAY DATEMISSAO USING '999999'
            @ 15,11 SAY 'DATA DE VENCIMENTO DA NOTA : '
            @ 15,40 SAY DATCOBRANC USING '999999'
            SET FORMAT TO SCREEN
            STORE ' ' TO RESP
            @ 18,56 SAY 'ESTA CORRETO ? (S/N)';
            GET RESP PICTURE 'X'

            READ
            CLEAR GETS
            IF RESP = 'S'
                DELETE
            ELSE
                @ 20,1 SAY 'NAO SERA REALIZADA A DELICAO'
            ENDIF
        ELSE
            SET FORMAT TO PRINT
            @ 18,5 SAY 'NAO EXISTE NOTA DE COBRANCA BANCARIA COM ESSA DATA '
            @ 19,5 SAY 'DE VENCIMENTO'
            SET FORMAT TO SCREEN
        ENDIF
        @ 22,1 SAY ' '
        WAIT
    ENDIF
    USE
    RELEASE ALL EXCEPT X*
    DO TELAFIM
ENDDO
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa DCARNEAG.CMD

```

TYPE DCARNEAG.CMD
SET FORMAT TO PRINT
EJECT
ERASE
@ 2,10 SAY 'ESTE PROGRAMA FAZ A DELICAÇÃO NO ARQUIVO QUE CONTROLA OS VENCIMENTOS'
@ 4,10 SAY 'DOS CARNÊS DE FINANCIAMENTO. COMO ESSES PAGAMENTOS NÃO ATRASAM'
@ 6,10 SAY 'POIS ESTÃO SOB RESPONSABILIDADE DA CORRETORA, TODO INÍCIO DE MÊS'
@ 8,10 SAY 'PODE-SE ELIMINAR DO ARQUIVO OS CARNÊS QUE ENCERRARAM NO MÊS ANTE-'
@ 10,10 SAY 'RIOR.'
SET FORMAT TO SCREEN
STORE ' ' TO MES
@ 16,38 SAY 'DIGITE O MÊS ANTERIOR AO MÊS CORRENTE';
GET MES PICTURE 'XX'
READ
CLEAR GETS
SELECT PRIMARY
USE B:AGENDA.DBF
LOCATE FOR ((STR(DATENCERR,6),3,2) = 'MES' .AND. ;
              (STR(DATENCERR,6),5,2) <= (DATE(),7,2)))
  
```

```

IF .NOT. EOF
  STORE 'S' TO NOVAPAG
  DO WHILE NOVAPAG = 'S'
    ERASE
    SET FORMAT TO PRINT
    EJECT
    @ 1,15 SAY 'RELACAO DE CARNES QUITADOS NO MES : '
    @ 1,51 SAY MES USING 'XX'
    @ 1,70 SAY DATE() USING 'XXXXXXXX'
    @ 2,15 SAY '-----'
    @ 4,28 SAY 'SEGURADO'
    @ 4,56 SAY 'INSCRICAO'
    @ 4,67 SAY 'DATA VENCTO'
    @ 5,28 SAY '-----'
    @ 5,56 SAY '-----'
    @ 5,67 SAY '-----'
    STORE 7 TO LIN
    DO WHILE .NOT. EOF .AND. LIN <= 78
      STORE INSCRICAO TO XNROINS
      STORE $(STR(DATENCERR,6),1,2) + '/' + $(STR(DATENCERR,6),3,2) + '/' +
        $(STR(DATENCERR,6),5,2) TO DATAREL

      DELETE
      SELECT SECONDARY
      USE B:FICHACLI.DBF
      DO WHILE INSCRICAO <> XNROINS
        SKIP
      ENDDO
      @ LIN,5 SAY NOMCLIENTE USING;
        'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
      @ LIN,57 SAY XNROINS USING '9999999'
      @ LIN,68 SAY DATAREL USING 'XXXXXXXX'
      STORE LIN+2 TO LIN
      SELECT PRIMARY
      CONTINUE
    ENDDO
    IF EOF
      STORE 'N' TO NOVAPAG
    ENDIF
    SET FORMAT TO SCREEN
    @ 22,1 SAY ' '
    WAIT
  ENDDO
ELSE
  ERASE
  SET FORMAT TO PRINT
  EJECT
  @ 5,70 SAY DATE() USING 'XXXXXXXX'
  @ 10,30 SAY 'NO MES : '
  @ 10,39 SAY MES USING 'XX'
  @ 11,30 SAY '-----'
  @ 13,10 SAY 'NAO FOI QUITADO NENHUM CARNE DE FINANCIAMENTO DE SEGURO'
  SET FORMAT TO SCREEN
  @ 22,1 SAY ' '
  WAIT
ENDIF
ERASE
USE
SELECT SECONDARY
USE
RELEASE ALL EXCEPT X*
RETURN

```


Diagrama de Estrutura Modular e Listagem do Programa ULTIMAIN.CMD

```

A)TYPE ULTIMAIN.CMD
SET FORMAT TO PRINT
@ 14,7 SAY 'O NUMERO DE INSCRICAO DO SEGURADO E FORMADO DA SEGUINTE'
@ 15,7 SAY 'MANEIRA :   AANNNNN'
@ 16,7 SAY 'ONDE :           AA - CORRESPONDE AO ANO'
@ 17,7 SAY '           NNNNN - NUMERO SEQUENCIAL'
USE B=FICHACLI.DBF
GOTO BOTTOM
IF # = 0
    @ 19,7 SAY 'AINDA NAO EXISTEM CLIENTES CADASTRADOS PORTANTO, 0'
    @ 20,7 SAY 'NUMERO DE INSCRICAO SERA   AA00001'
ELSE
    @ 19,7 SAY 'O ULTIMO NUMERO DE INSCRICAO E : '
    @ 19,40 SAY INSCRICAO
ENDIF
@ 22,1 SAY ' '
USE
SET FORMAT TO SCREEN
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa FCHGERAL.CMD

```

A>TYPE FCHGERAL.CMD
ERASE
@ 1,5 SAY 'AS INSERCOES A SEGUIR CORRESPONDEM A FICHA DE CLIENTE'
USE B:FICHACLI.DBF
APPEND BLANK
@ 3,0 SAY 'DIGITE AS SEGUINTE INFORMACOES : '
STORE 0 TO XNROINS
@ 5,0 SAY 'NUMERO DE INSCRICAO';
GET XNROINS PICTURE '9999999'
STORE ' ' TO NOMECLI
@ 7,0 SAY 'NOME DO CLIENTE';
GET NOMECLI PICTURE;
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
STORE ' ' TO PROFISSAO
@ 9,0 SAY 'PROFISSAO';
GET PROFISSAO PICTURE 'XXXXXXXXXXXXXXXXXXXXX'
STORE ' ' TO END
@ 11,0 SAY 'ENDERECO';
GET END PICTURE;
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
STORE ' ' TO ENDSUC
@ 13,0 SAY 'ENDERECO SUCURSAIS';
GET ENDSUC PICTURE;
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
STORE ' ' TO NOMEREP
@ 15,0 SAY 'NOME DO REPRESENTANTE';
GET NOMEREP PICTURE;
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
STORE ' ' TO RESPON
@ 17,0 SAY 'RESPONSAVEL PELO SEGURO';
GET RESPON PICTURE;
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
STORE ' ' TO CARGOESP
@ 19,0 SAY 'CARGO DO RESPONSAVEL';
GET CARGOESP PICTURE 'XXXXXXXXXXXXXXXXXXXXX'
STORE ' ' TO GERDIR
@ 21,0 SAY 'NOME GERENTE/DIRETOR';
GET GERDIR PICTURE;
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
READ
CLEAR GETS
REPLACE INSCRICAO WITH XNROINS,NOMCLIENTE WITH NOMECLI,RAMOATIVID WITH ;
PROFISSAO,ENDERECO WITH END,ENDSUCURS WITH ENDSUC,NOMREPRES WITH ;
NOMEREP,RESPSEGURO WITH RESPON,CARGO WITH CARGOESP,GERENTEDIR WITH ;
GERDIR
ERASE
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa IMPRCODS.CMD

```

A)TYPE IMPRCODS.CMD
SET TALK OFF
ERASE
SET FORMAT TO PRINT
EJECT
@ 1,10 SAY 'ATENCAO!...'
@ 3,10 SAY '          AGORA SERAO APRESENTADOS OS VARIOS TIPOS DE SEGUROS COM'
@ 5,10 SAY 'SEUS RESPECTIVOS CODIGOS. PARA EFEITO DE PROCESSAMENTO, VOCE DEVE-'
@ 7,10 SAY 'RA ANOTAR O(S) CODIGO(S) ASSOCIADO(S) AO(S) SEGURO(S) : '
@ 9,10 SAY 'A- QUE O CLIENTE EFETUOU (NO CASO DE OPERACAO DE INSERCAO)'
@ 11,10 SAY 'B- DOS QUAIS DESEJA-SE RELACAO DE SEGURADOS (CASO DE RELATORIOS)'
@ 13,10 SAY 'PARA QUE VOCE POSSA PREENCHER CORRETAMENTE AS INFORMACOES QUE LHE'
@ 15,10 SAY 'SERAO SOLICITADAS.'
SET FORMAT TO SCREEN
@ 22,1 SAY ' '
WAIT
ERASE
USE B:RAMOSEG.DBF
SET FORMAT TO PRINT
DO WHILE .NOT. EOF
    EJECT
    @ 1,18 SAY 'CODIGO'
    @ 1,35 SAY 'TIPO DE SEGURO'
    @ 2,18 SAY '====='
    @ 2,35 SAY '===== '
    STORE 5 TO CONT
    DO WHILE CONT <= 21 .AND. .NOT. EOF
        @ CONT,20 SAY RAMO
        @ CONT,35 SAY TIPOSEG
        STORE CONT+2 TO CONT
        SKIP
    ENDDO
    @ 22,1 SAY ' '
    WAIT
    ERASE
ENDDO
USE
RELEASE ALL EXCEPT X*
SET FORMAT TO SCREEN
RETURN

```

Diagrama de Estrutura Modular e Listagem do Programa PROCURA.CMD

```

A)TYPE PROCURA.CMD
ERASE
USE B:FICHACLI.DBF
STORE 'S' TO XPEDENOME
DO WHILE XPEDENOME = 'S'
    GOTO TOP
    STORE ' ' TO NOMECLI
    @ 5,10 SAY 'NOME DO CLIENTE';
    GET NOMECLI PICTURE;
    'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
    READ
    CLEAR GETS
    LOCATE FOR NOMCLIENTE = '&NOMECLI'
    IF EOF
        STORE 0 TO XNROINS
        @ 10,30 SAY 'CLIENTE NAO CADASTRADO'
        @ 12,30 SAY 'DESEJA ENTRAR COM O NOME NOVAMENTE ? (S/N)';
        GET XPEDENOME PICTURE 'X'
        READ
        CLEAR GETS
    ELSE
        STORE INSCRICAO TO XNROINS
        STORE 'N' TO XPEDENOME
    ENDIF
    ERASE
ENDDO
USE
RETURN

```


Diagrama de Estrutura Modular e Listagem do Programa TELAFIM.CMD

```

A)TYPE TELAFIM.CMD
SET FORMAT TO PRINT
ERASE
  STORE 'S' TO XTELAFINAL
  DO WHILE XTELAFINAL = 'S'
    EJECT
    @ 6,5 SAY 'ESCOLHA SUA OPCAO : '
    @ 8,10 SAY '1- REPETIR ESTA OPERACAO '
    @ 10,10 SAY '2- RETORNAR AO MENU ANTERIOR '
    @ 12,10 SAY '3- RETORNAR AO MENU PRINCIPAL '
    @ 14,10 SAY '4- TERMINAR EXECUCAO '
    SET FORMAT TO SCREEN
    STORE ' ' TO XRESP
    @ 21,55 SAY 'DIGITE SUA OPCAO '
    GET XRESP PICTURE 'X'

    READ
    CLEAR GETS
    DO CASE
      CASE XRESP = '1'
        STORE 'N' TO XTELAFINAL
      CASE XRESP = '2'
        STORE 'N' TO XTELAFINAL
        STORE 'N' TO XREPETIR
      CASE XRESP = '3'
        STORE 'N' TO XTELAFINAL
        STORE 'N' TO XREPETIR
        STORE 'N' TO XMENUINS
        STORE 'N' TO XMENUREL
        STORE 'N' TO XMENUDEL
      CASE XRESP = '4'
        STORE 'N' TO XTELAFINAL
        STORE 'N' TO XREPETIR
        STORE 'N' TO XMENUINS
        STORE 'N' TO XMENUREL
        STORE 'N' TO XMENUDEL
        STORE 'N' TO XMENU
    OTHERWISE
      @ 23,55 SAY 'NAO EXISTE ESSA OPCAO '
      WAIT
    ENDCASE
  ERASE
ENDDO
RETURN

```