

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

TRABALHO DE FORMATURA

A FÁBRICA DE SOFTWARE

MARIA ALICE BRAGA FRONTINI

ORIENTADOR : PROF. MARCELO PESSOA

1988

1988
F9284

AGRADECIMENTOS

- . Ao Prof. Marcelo Pessoa pela dedicação e orientação.
- . À Empresa pela oportunidade de estágio.
- . Ao Sr. Sérgio Leifert pelo apoio e orientação.
- . A Sra. Valéria Motta pelas experiências transmitidas.
- . Ao Engenheiro Jairo Avritchir pelas sugestões, conhecimentos transmitidos e dedicada orientação.
- . A Sra. Lia Pignatari Malmegrim pela revisão gramatical.
- . A Marcelo Frontini pelo companheirismo e amizade.
- . A todos os amigos pelo intenso apoio e ajuda.
- . A todos aqueles que, direta ou indiretamente, participaram na realização deste trabalho, e também aqueles que possam contribuir com críticas, após sua leitura.

SUMÁRIO

Este trabalho analisa a chamada Crise do Software, assunto de polêmicas discussões e grandes preocupações no ramo da informática como um todo.

Através da reunião de conceitos e técnicas já existentes tanto a nível de informática como a nível de organização do trabalho e da produção, é proposta uma solução para os problemas existentes no desenvolvimento de Software. Também é feita uma análise crítica à solução adotada.

O Trabalho aborda técnicas, metodologias e ferramentas para desenvolvimento de Sistemas, sendo algumas consagradas e outras extremamente atuais.

ÍNDICE

1. COMENTÁRIOS DA METODOLOGIA PARA RESOLUÇÃO DE PROBLEMAS.

1.1. Resumo.....	1
1.2. Introdução.....	2
1.3. Método de Resolução de Problemas.....	3
1.3.1. Formulação do Problema.....	3
1.3.2. Análise do Problema.....	4
1.3.3. Pesquisa das Alternativas.....	4
1.3.4. Avaliação das Alternativas.....	5
1.4. A Resolução de Problemas na Informática.....	6
1.5. A Abordagem Adotada.....	7

2. DESCRIÇÃO DO AMBIENTE.

2.1. Resumo.....	11
2.2. A Empresa.....	12
2.2.1. Introdução.....	12
2.2.2. Objetivos.....	16
2.2.3. Ambientes.....	17
2.2.4. Recursos.....	18
2.2.5. Componentes.....	20
2.2.6. Administração.....	22
2.3. O Subsistema da Produção.....	24
2.3.1. Objetivos.....	24
2.3.2. Ambientes.....	25
2.3.3. Recursos.....	25
2.3.4. Componentes.....	26

2.3.5. Administração.....	29
2.4. O Subsistema Câmbio.....	29
2.2.1. Introdução.....	29
2.2.2. Objetivos.....	30
2.2.3. Ambientes.....	30
2.2.4. Recursos.....	32
2.2.5. Componentes.....	33
2.2.6. Administração.....	38
2.5. O Estágio.....	38
 3. A CRISE DO SOFTWARE.	
3.1. Resumo.....	41
3.2. O Ciclo de Vida do Software.....	42
3.3. Definição da Crise do Software.....	52
3.4. A Crise do Software na Empresa.....	58
3.5. Uma Proposta de Solução.....	63
 4. A FÁBRICA DE SOFTWARE.	
4.1. Resumo.....	70
4.2. Justificando o Termo Fábrica.....	71
4.3. A Estrutura Organizacional de uma Fábrica.....	73
4.4. A Estrutura Organizacional da Fábrica de Software....	80
 5. A FABRICAÇÃO DE SOFTWARE.	
5.1. Resumo.....	88
5.2. Recursos para o Desenvolvimento de Software.....	89
5.3. Ambientes Centrados em Metodologia.....	94

5.3.1. Uma Breve Descrição das Metodologias.....	94
5.3.1.1. Conceituando Metodologia.....	94
5.3.1.2. Metodologias Baseadas em Processos.....	95
5.3.1.2. Metodologias Baseadas em Dados.....	108
5.3.2. Detalhando algumas metodologias.....	120
5.3.2.1. Apresentação.....	120
5.3.2.2. A Metodologia Remora.....	120
5.3.2.3. A Metodologia Method/1.....	130
5.3.3. Prototipação.....	134

6. CASE - ENGENHARIA DE SOFTWARE ASSISTIDA PELO COMPUTADOR.

6.1. Resumo.....	141
6.2. A Automação na Fábrica de Software.....	142
6.3. Definição e Considerações sobre CASE.....	145
6.4. Uma Análise Comparativa de Alguns Produtos do Mercado	156
6.5. O Que Está Sendo Feito a Nível de Pesquisa no Brasil.	161
6.5.1. Apresentação.....	161
6.5.2. MOSAICO.....	161
6.5.3. SIPS.....	163
6.5.4. MIND-AID.....	168
6.6. Tendências.....	171

7. UMA ANÁLISE CRÍTICA À FÁBRICA DE SOFTWARE.

7.1. Resumo.....	174
7.2. Consequências da Mudança de Processo Produtivo.....	175
7.3. Dificuldades para a Mudança no Processo de Produção de Software.....	179

7.4. A Evolução na Organização do Trabalho após Taylor....	183
7.4.1. Recapitulando Algumas Teorias da Organização do Trabalho.....	183
7.4.2. Questionando o Taylorismo na Fábrica de Software.....	189
 8. APLICANDO OS CASEs DESENVOLVIDOS NA EMPRESA.	
8.1. Resumo.....	193
8.2. O Projeto do Sistema de Câmbio.....	194
8.3. Gerenciador de Dicionário de Dados - o "Nosso CASE"..	196
8.4. Exemplificando o GDD e o MIND-AID.....	206
8.5. Comparando o Gerenciador de Dicionário de Dados com o MIND-AID.....	208
8.6. Facilitando a utilização da Remora.....	209
8.6.1. Dificuldades Encontradas na utilização da Remora.....	209
8.6.1. Aprimorando a Sequência de Eventos.....	211
8.6.3. Um Método para Converter a Sequência de Eventos em Modelo.....	213
8.7. Como Tratar Sistemas Complexos e Dinâmicos Como o Câmbio	215
 9. CONCLUSÃO.....	220
 10. BIBLIOGRAFIA.....	222

ANEXO 1.

- . Questionário para Avaliação de Metodologias.

ANEXO 2.

- . Um Exemplo de Aplicação da Metodologia Remora e
- . Um Esquema do Método da Metodologia Method/1

ANEXO 3.

- . Um Exemplo de Especificação de um Sistema por CASE.

ANEXO 4.

- . Listagens Resultantes da Aplicação do GDD.
- . Listagens Resultantes da Aplicação do MIND-AID.

ÍNDICE DE FIGURAS

FIG 1.5.1	- Esquema da abordagem Top-down.....	10
FIG 2.2.1.1	- Funcionamento de um banco comercial.....	13
FIG 2.2.1.2	- Estrutura Hierárquica do Sistema Financeiro..	15
FIG 2.2.5.1	- Estrutura do Sistema Financeiro.....	23
FIG 2.3.4.1	- Estrutura do Subsistema da Produção.....	28
FIG 2.4.5.1	- Estrutura básica do Câmbio.....	36
FIG 3.2.1	- Esquema da Fase de Planejamento do Ciclo de Vida do Software.....	46
FIG 3.2.2	- Esquema da Fase de Desenvolvimento do Ciclo de Vida do Software.....	49
FIG 3.2.3	- Esquema da Fase de Manutenção do Ciclo de Vida do Software.....	51
FIG 3.3.1	- Um Exemplo Clássico do Não Atendimento das Necessidades do Usuário.....	54
FIG 4.4.1	- Impacto no Custo do Sistema de Alterações Realizadas.....	87
FIG 5.3.1.2.1	- Exemplo de um Diagrama de Fluxo de Dados.....	97
FIG 5.3.1.2.2	- Exemplo de um Diagrama de Estrutura.....	99
FIG 5.3.1.2.3	- Exemplo de um diagrama utilizado pela SADT...	101
FIG 5.3.1.2.4	- Exemplo de um diagrama de Jackson.....	103
FIG 5.3.1.2.5	- Exemplo de um Diagrama de Warnier.....	106
FIG 5.3.1.3.1	- Exemplo de Diagrama Entidade-relacionamento com Atributos.....	111
FIG 5.3.1.3.2	- Exemplo de um Diagrama Voltado a Objetos.....	114
FIG 5.3.1.3.3	- Exemplo de um Modelo Rede.....	116

FIG 5.3.1.3.4	- Exemplo de um Modelo Hierárquico.....	118
FIG 5.3.2.1	- Dinâmica da Organização.....	124
FIG. 6.3.1	- Elementos de um CASE.....	147
GRÁF. 6.3.2	- Comparação entre os recursos necessários para o desenvolvimento de Sistemas sem automação e com automação (CASE).....	155
FIG. 6.5.1	- Uma Ilustração Esquemática da Estrutura do SIPS.....	167
FIG. 8.3.1	- Tela do Menu Principal do Gerenciador de Dicionário de Dados.....	201
FIG. 8.3.2	- Tela do Módulo de Consultas do Gerenciador de Dicionário de Dados.....	202
FIG. 8.3.3	- Tela do Módulo de Inclusão do Gerenciador de Dicionário de Dados.....	203
FIG. 8.3.4	- Tela do Módulo de Alteração do Gerenciador de Dicionário de Dados.....	204
FIG. 8.3.5	- Tela do Módulo de Exclusão do Gerenciador de Dicionário de Dados.....	205

OBSERVAÇÃO: as figuras, tabelas, desenhos, esquemas explicativos, etc., que fazem parte deste trabalho, quer seja no texto principal, quer seja nos anexos, foram elaborados pelo autor, a não ser os casos em que a fonte esteja explicitada no local.

RESUMO GERAL

Este trabalho procura fornecer uma solução à crise do desenvolvimento de Softwares, que basicamente consiste no baixo nível de qualidade e de produtividade na construção de Sistemas grandes e complexos.

Diversos conceitos existentes são utilizados e agrupados, de forma a obter uma solução do problema, partindo de soluções já propostas com um enfoque original - o da organização do trabalho.

Com a apresentação de técnicas, metodologias e ferramentas utilizadas na construção de sistemas, este trabalho também serve como base para a formação de estudantes do curso de Sistemas de Informação e dos indivíduos que estão se iniciando nesta área.

No primeiro capítulo, é tratada a importância de se ter em mente um método para a resolução de problemas. Também são discutidas as características deste processo na área de informática. As abordagens adotadas para a definição do objeto em estudo são especificadas.

No segundo capítulo, é feita uma descrição da empresa onde este trabalho foi realizado, de acordo com as abordagens adotadas no capítulo anterior. Além disso, é mostrado a origem do tema de nosso estudo.

No terceiro capítulo, são introduzidos conceitos do Ciclo de Vida do Software e da chamada Crise do Software, pois indícios desta foram encontrados na empresa do caso em estudo, como

explicitados neste mesmo capítulo. Uma solução para este problema é mencionada - a Fábrica de Software - , a qual vem aplicar os princípios da Administração Científica de Taylor.

No quarto capítulo, é tratada a Fábrica do Software. Para tal, o termo "fábrica" é justificado para a área de desenvolvimento de Sistemas. Todas as funções existentes no desenvolvimento de Sistemas são identificadas na estrutura organizacional clássica de uma fábrica, a fim de que um organograma para a Fábrica de Software seja aqui proposto.

No quinto capítulo, é enfocada a fase de fabricação do produto da Fábrica de Software. Dentre os tipos de recursos existentes nesta fase, concentramo-nos nos ambientes centrados em metodologias, através da descrição sucinta de diversas metodologias; na sequência foi elaborada uma descrição mais detalhada das metodologias Remora, Method/1 e do recurso de prototipação.

No sexto capítulo, são introduzidos os conceitos envolvidos na automação da Fábrica do Software. Também é apresentada uma nova tecnologia - O CASE (Computer Aided Software Engineering) - que vem propiciar esta automação. Para complementar, foi realizada uma análise comparativa de alguns produtos existentes no mercado, além de uma apresentação daquilo que vem sendo desenvolvido a nível de pesquisa no país.

No sétimo capítulo, são consideradas as dificuldades que podem surgir devido à implantação de CASE na Fábrica de Software. Também, é questionado o Taylorismo como solução adotada para a Crise do Software.

No oitavo capítulo, primeiramente, é descrito como se deu o projeto do novo Sistema de Câmbio da empresa e o CASE originado deste projeto - Gerenciador de Dicionário de Dados, elaborado por nós. Uma comparação com o CASE MIND-AID é realizada e a fim de ilustrar melhor fizemos uma aplicação dos dois CASEs para uma parte do Sistema de Câmbio. Durante esta aplicação algumas dificuldades foram sentidas e uma solução para estes problemas foi elaborada.

E no nono capítulo é feita uma conclusão a respeito deste trabalho.

1 COMENTÁRIOS DA METODOLOGIA PARA RESOLUÇÃO DE PROBLEMAS

1.1 Resumo

Este capítulo trata da importância de termos em mente um método para a resolução de problemas.

Além disso, discutem-se as características desse processo de resolução de problemas na área, em que desenvolvemos esse trabalho.

Também são definidas as abordagens adotadas para definição do objeto de estudo.

1.2 Introdução

Gostaríamos de, ao iniciar este trabalho, discutir um pouco a respeito de resolução de problemas.

Como sabemos o engenheiro de produção é um engenheiro de métodos, ou seja, através de métodos adequados ele procura solucionar o problema da melhor forma possível.

Quando iniciamos um trabalho, que envolve a resolução de um problema, as coisas parecem obscuras, sem saída, mas se utilizarmos um método de solução de problemas, conseguiremos chegar ao final com uma proposição. Assim, ao elaborarmos este trabalho de formatura, devemos ter em mente um processo de definição e resolução de problemas, por mais simples que ele seja.

Primeiramente, recapitulemos o método tradicional de resolução de problemas de métodos proposto por Krick. Este processo é subdividido nas fases:

- . Formulação do Problema,
- . Análise do problema, com coleta de informações,
- . Pesquisas das alternativas,
- . Avaliação das alternativas e escolha da melhor solução.

Logicamente, essas etapas não possuem uma delimitação muito rígida, dependendo do caso, algumas fases são executadas paralelamente e de forma iterativa. Por exemplo, muitas vezes a formulação do problema pode ser feita paralela à coleta de dados, e ao se coletar os dados, reformula-se o problema novamente.

1.3 Método de Resolução de Problemas

1.3.1 Formulação do problema

Nesta etapa o problema é identificado, ou seja, define-se o objeto de estudo de forma ampla.

Deve-se ter em mente, que é necessário determinar-se o verdadeiro problema a ser resolvido. Isto só é possível, se essa identificação for feita de forma global, sem se prender em detalhes.

Basicamente, a formulação do problema consiste em definir:

- . estado inicial - de onde se deve partir.
- . estado final - aonde se pretende chegar.
- . restrições fundamentais - são aquelas que de qualquer forma devem ser respeitadas. Deve-se considerar nessa fase o mínimo possível de restrições, a fim de que a identificação do problema fique a mais ampla possível, o que permitirá uma busca maior de alternativas. Quanto mais ampla for a abordagem adotada, mais profundas serão as mudanças.
- . criterio de preferência - é o critério que se define para se determinar a melhor alternativa.
- . prazo - é necessário que se tenha em mente um prazo disponível para a resolução do problema, pois de acordo com esse prazo varia a extensão e profundidade da análise a ser feita.

1.3.2 Análise do Problema

Nesta fase serão detalhadas todas as características do problema, relacionadas com o estado inicial, o estado final e com as restrições.

Todos os dados e informações pertinentes ao assunto devem ser coletados de forma crítica, a fim de que apenas os elementos significativos para a resolução do problema sejam considerados.

Deve-se, também, analisar todas as restrições, para se identificar as que realmente são válidas, as que podem ser modificadas e as que na realidade podem ser abandonadas, por terem sido adotadas de forma quase que inconsciente.

É muito importante que as restrições fictícias, ou seja, aquelas que são assumidas inconscientemente no decorrer do estudo, sejam abandonadas, para que se obtenha soluções realmente renovadoras, ao invés de soluções que geralmente se referem a características do método corrente.

1.3.3 Pesquisa das Alternativas

Esta é uma etapa de importância fundamental no processo de resolução de problemas, pois o sucesso desse trabalho depende da qualidade das alternativas pesquisadas.

O ideal seria que se conseguisse obter a alternativa ótima, mas isto é um processo difícil, já que apenas algumas alternativas são conhecidas.

O processo de busca de alternativas depende basicamente do conhecimento, experiência e criatividade do indivíduo, além de uma boa dose de sorte. Então, para se obter soluções renovadoras, deve-se tomar algumas medidas que contribuem para a melhora do processo de busca de alternativas, dentre elas:

- . formular o problema de forma ampla.
- . abordar o problema de forma sistemática.
- . evitar a familiaridade com o método existente.
- . ampliar os conhecimentos através de leituras, conversas, sugestões.
- . exercer um esforço em termos de trabalho mental.
- . adotar sempre uma atitude crítica.
- . acumular o maior número possível de alternativas.
- . evitar o conservadorismo.
- . evitar a rejeição ou satisfação prematura de idéias.

1.3.4 Avaliação das Alternativas

Aqui se faz uma análise das alternativas e se escolhe aquela que melhor satisfaz o critério de preferência, definido na etapa de definição do problema.

Para se continuar com a busca de mais alternativas, deve-se fazer uma análise de custo-benefício. Ou melhor, quando o benefício trazido pela pesquisa de outra alternativa não mais compensar o custo envolvido nessa busca, deve-se cessar com a pesquisa.

Com a escolha da melhor alternativa, termina-se o processo de resolução do problema. Quando a alternativa escolhida não mais satisfizer, devido a mudanças do ambiente, da realidade, o processo deve ser reiniciado.

1.4 A Resolução de Problemas na Informática

Apesar do processo de resolução de problemas ser aplicável em qualquer área, existem certos tipos de problema que requerem uma maior atenção em algumas fases desse processo do que em outras. A informática é uma dessas áreas.

Enfatizamos essa área, porque o problema por nós examinado aí se encontra.

Atualmente, na informática, a dificuldade maior ao se resolver um problema está na definição deste. A evolução tecnológica da informática tem sido muito grande, são tantas as ferramentas disponíveis no mercado, que muitas vezes a solução de um problema passa a ser a aplicação correta dessas ferramentas. No entanto, qual ferramenta deve ser utilizada ?

Para respondermos essa pergunta, a qual é o maior dilema atual dessa área, devemos ter uma definição clara e exata do problema.

Muitas pessoas possuem uma visão distorcida do que é essa definição do problema. A maioria, ao definir um problema, parte direto para a solução, dizendo "como" fará. Essa atitude é extremamente perigosa, pois soluções "viciadas" e muitas vezes

errôneas poderão ser tomadas. Ao passo se o "quê" está realmente ocorrendo for definido, teremos um retrato puro da situação, e soluções corretas serão adotadas.

Então, podemos perceber o quanto importante é, que os problemas da área de informática sejam corretamente definidos, para que ferramentas adequadas sejam aplicadas.

Para auxiliar essa fase de definição do problema, existem algumas abordagens. Nós procuraremos adotá-las, a fim de que distorções do verdadeiro problema não sejam cometidas.

1.5 A Abordagem Adotada

Primeiramente, procuraremos adotar uma abordagem sistêmica. Ou seja, procuraremos pensar no objeto de estudo como um Sistema. Mas o que é um Sistema ?

Segundo Churchman [12], um Sistema é um conjunto de partes coordenadas para realizar um conjunto de finalidades. Poderíamos, também, dar uma definição menos formal, dizendo que um Sistema é uma "caixa preta", que transforma dados, impulsos de entrada em dados ou impulsos de saída.

A abordagem sistêmica, de acordo com Churchman [12], consiste na identificação dos seguintes elementos do Sistema em estudo:

. objetivo - são as finalidades reais que o Sistema deseja alcançar acima de tudo. Para identificarmos os objetivos reais do Sistema, basta determinar se o Sistema sacrificaria outras

finalidades com o fim de atingir seus objetivos.

- . **ambiente** - é tudo aquilo que está situado "fora" do Sistema, ou seja, é tudo aquilo que não está sob controle do Sistema, mas que determina em parte o funcionamento deste, que influencia o Sistema, principalmente, em relação a seus objetivos.
- . **recursos** - é tudo o que o Sistema possui à sua disposição; são utilizados por ele, para que seus objetivos sejam atingidos. Como os recursos se encontram dentro do Sistema, eles estão sob o controle do mesmo, assim sendo, o Sistema pode decidir a respeito de seus recursos.
- . **componentes** - são aqueles que executam as tarefas, atividades do Sistema, para que este atinja seus objetivos.
- . **administração** - determina as finalidades de cada componente, aloca os recursos e controla o rendimento do Sistema..

As vantagens de se aplicar esse tipo de enfoque são:

- . obter uma descrição sistemática e estruturada do elemento em estudo.
- . garante-se que todos os elementos fundamentais do objeto em estudo sejam identificados.
- . evita-se que confusões sejam feitas, no sentido de achar que certos elementos fazem parte do Sistema, quando na realidade eles fazem parte de seu ambiente, no qual o

Sistema não tem poder de influência, de alteração.

Para complementar a abordagem sistêmica, utilizamos um enfoque "Top-down", ou seja, um enfoque por níveis crescente de detalhamento (fig. 1.5.1).

A vantagem de se utilizar esse enfoque, é que a profundidade e extensão das considerações vão aumentando de acordo com as necessidades. Evita-se que muitos detalhes sejam considerados prematuramente, o que poderia ocasionar um afastamento dos objetivos do estudo.

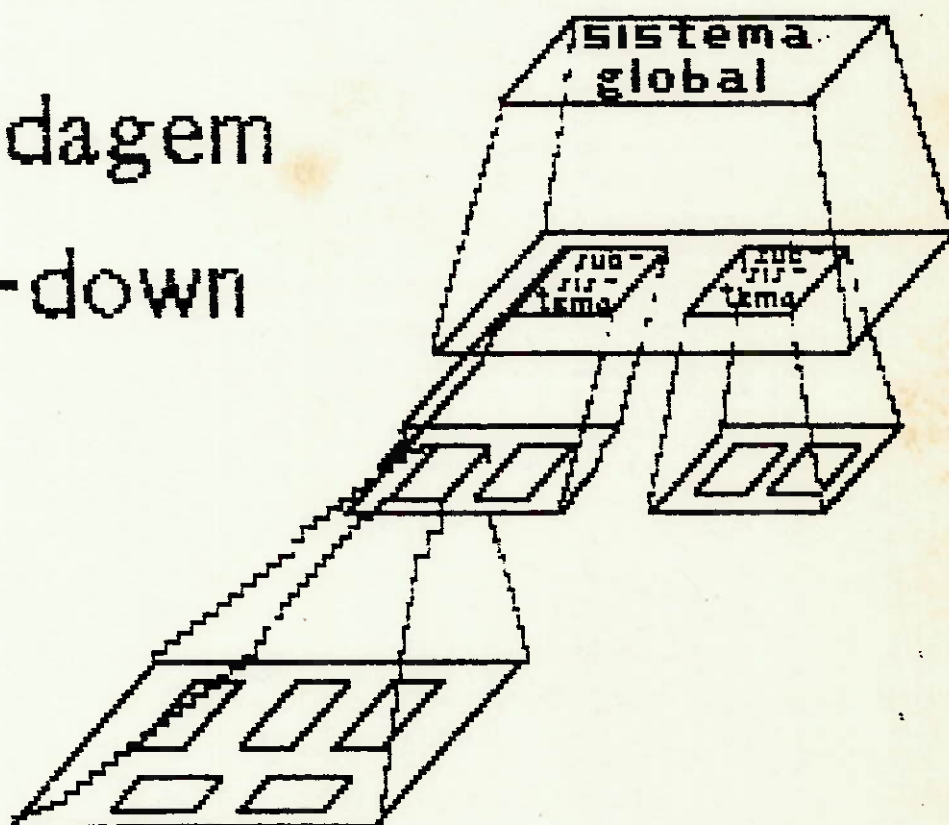
Um dos motivos pelo qual optamos por essas abordagens, além de suas vantagens já citadas, é o fato de acharmos interessante utilizar técnicas da própria área de informática, para desenvolver um trabalho dessa área. Isso demonstra o quanto essas técnicas são abrangentes.

Sallentamos, que utilizamos a abordagem sistêmica apenas para determinar o Sistema, onde se insere o elemento de nosso estudo. Já o enfoque top-down foi utilizado ao longo deste trabalho.

No próximo capítulo fazemos uma descrição do Ambiente de Trabalho. Precisamos definir o ambiente onde o objeto de estudo se encontra, para que o compreendamos devidamente.

FIG. 1.5.1 - Esquema da abordagem Top-down.

abordagem
top-down



2 DESCRIÇÃO DO AMBIENTE

2.1 Resumo

Neste capítulo está descrita a empresa, onde realizamos este trabalho. Essa descrição foi feita de acordo com a abordagem descrita no capítulo anterior.

Mostramos também neste capítulo, como surgiu o tema de nosso estudo.

2.2 A Empresa

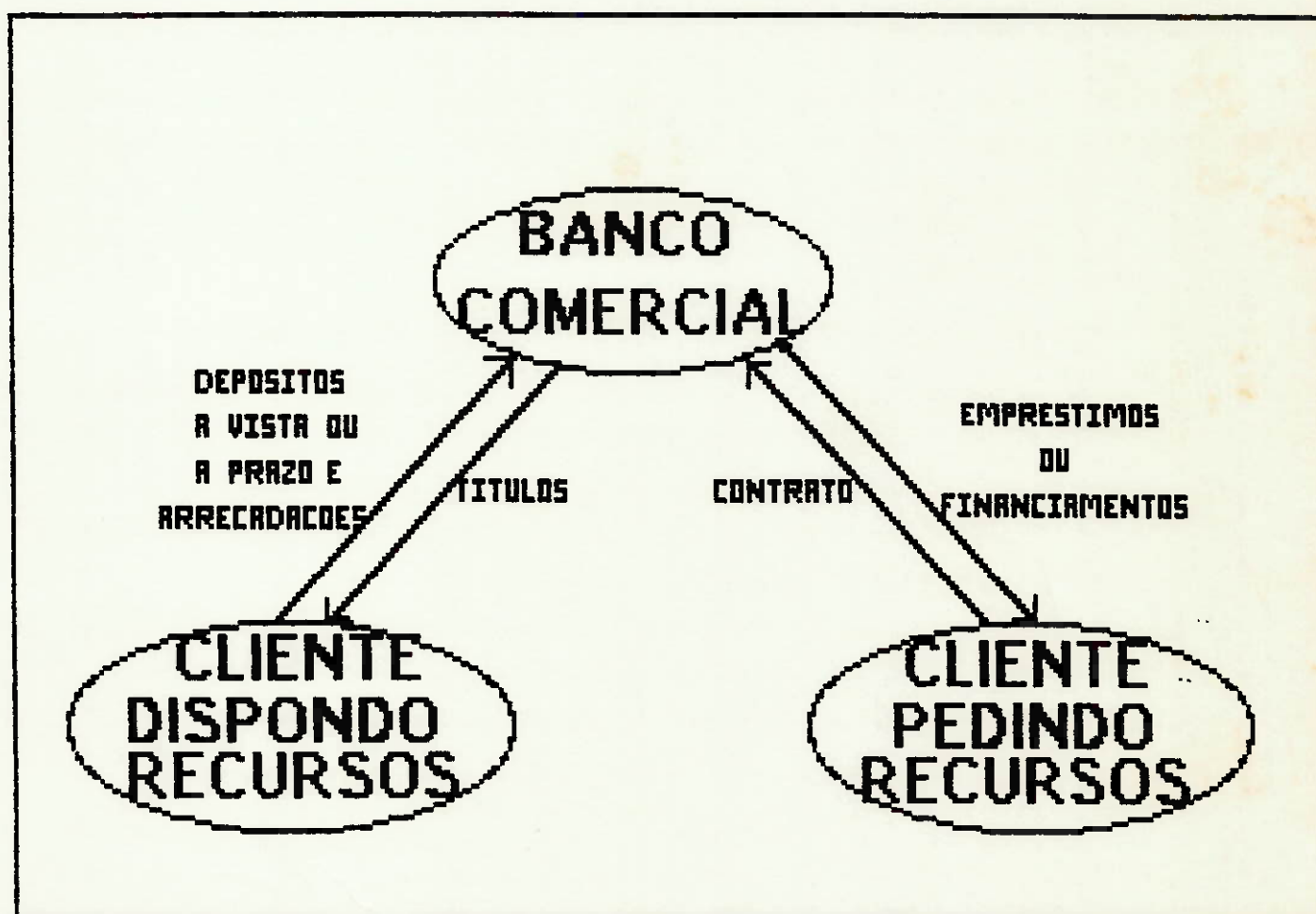
2.2.1 Introdução

A empresa, onde realizamos nosso estágio, consiste num Sistema Financeiro de capital majoritário estrangeiro

A principal empresa do sistema financeiro é o Banco Comercial, que tem por função primordial, promover e agilizar a circulação do dinheiro dentro da sociedade, captando-o das pessoas, empresas e governos que possuem esses recursos disponíveis e canalizando-o para as pessoas, empresas e governos que dele necessitam, para fazer investimentos ou satisfazer suas necessidades de consumo.

Basicamente, o Banco funciona como intermediário entre estes dois tipos de clientes. Ele capta recursos através dos depósitos à vista (contas correntes), depósitos à prazo (aplicações por prazos determinados - Certificado de Depósito Bancário, Recibo de Depósito Bancário, Letras de Câmbio), recursos oficiais, arrecadações de contas, impostos, carnês, etc., e aplica esses recursos através de empréstimos e financiamentos, seja para pessoas físicas ou jurídicas (fig. 2.2.1.1).

FIG. 2.2.1.1 - Funcionamento de um banco comercial.

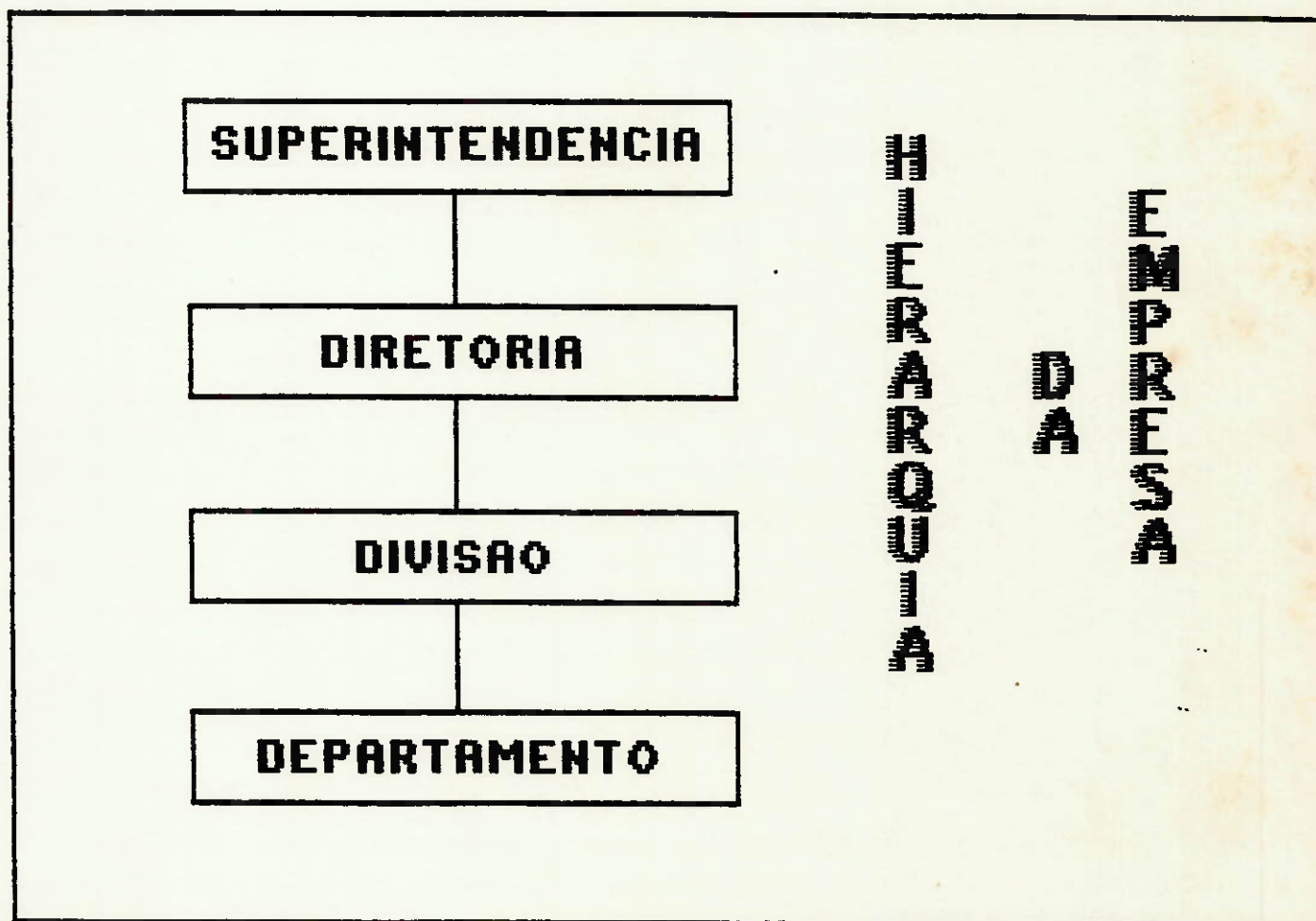


Além disso, o Banco presta uma série de serviços para seus clientes, por exemplo:

- . Arrecadação - o banco intermedia o pagamento de taxas, impostos, etc.
- . Carteira de Pagamentos - o banco administra a cobrança de duplicatas.
- . Custódia de Títulos - o banco guarda e administra os títulos financeiros que lhe são entregues.
- . Cofres - cofres de segurança existentes nas agências para guarda de valores.
- . Caixa Automático - permitem sacar dinheiro, depositar e realizar consultas a saldos e extratos a qualquer momento em qualquer caixa eletrônico da rede no país.

A estrutura hierárquica do Sistema Financeiro apresenta poucos níveis, sendo bem horizontal (fig. 2.2.1.2).

FIG. 2.2.1.2 - Estrutura Hierárquica do Sistema Financeiro.



2.2.2 Objetivos

Os objetivos do Sistema Financeiro como um todo são :

- . manter o bom nível de qualidade do relacionamento sistema-cliente,
- . aprimorar os níveis dos padrões de qualidade dos serviços,
- . desenvolvimento de novos produtos e serviços,
- . projetar no mercado nacional uma imagem de banco com sólidas relações internacionais,
- . desenvolver uma cultura de planejamento (objetivos, políticas e ações) dentro de Sistema a fim de se preparar para o futuro,
- . administrar seus recursos humanos através de uma política de vanguarda,
- . dar uma formação profissional e gerencial para seus funcionários,
- . melhorar os instrumentos de gestão do Sistema Financeiro,

. informatizar extensivamente o Sistema , a fim de que a Informática seja um instrumento de apoio à tomada de decisão.

Indiretamente, através destes objetivos a empresa estará atendendo seu objetivo capitalista de obter bons níveis de lucro.

2.2.3 Ambiente

Os elementos constituintes do ambiente do Sistema, segundo as definições do cap.1, são:

. Mercado Financeiro Internacional - este é o Sistema de nível superior onde se encontra a empresa, onde está sendo realizado este trabalho. Este Sistema é constituído por todas as instituições e autoridades financeiras internacionais. No caso em estudo, a maior parte do capital da empresa pertence a uma instituição estrangeira, que possui filiais em vários outros países.

. Mercado Financeiro Nacional - é formado por todas as empresas concorrentes nacionais ou multinacionais, além das autoridades financeiras governamentais que possuem forte influência, regulando o funcionamento do mercado e seus componentes.

. Mercado de Clientes - devido às características e à

política do Sistema Financeiro, ele possui uma atuação especializada, selecionando sua clientela, que é reduzida numericamente mas de grande expressão econômica e financeira, sendo constituída principalmente de pessoas jurídicas.

2.2.4 Recursos

Os recursos que o Sistema possui disponíveis são:

- . Capital Próprio - é o capital que provém dos investidores que possuem ações do Sistema Financeiro.
- . Capital de Terceiros - proveniente de recursos obtidos no país ou no exterior junto a clientes ou a instituições financeiras.
- . Pontos de comercialização - todas as 54 agências do Banco Comercial comercializam os produtos do Sistema Financeiro. As agências estão espalhadas por todo o país.
- . Empresas do Sistema Financeiro - essas empresas não possuem rede de agências própria para oferta de seus produtos ou serviços, pois se utilizam das agências do Banco Comercial. Embora cada uma delas tenha sua função específica, trabalham visando o mesmo objetivo : "Captar clientes para o Sistema e oferecer serviços especializados". O Sistema Financeiro é

constituído das seguintes empresas:

- . empresa de distribuição de títulos e valores mobiliários - é a distribuidora de títulos do Sistema. É responsável pela colocação dos títulos pela rede de agências e em locais onde não há agência do Banco Comercial.
- . empresa de turismo - é a companhia de turismo do Sistema. Faz reservas de passagens e hotéis, planeja e realiza excursões, convenções, etc.
- . empresa de crédito, financiamento e investimento - é a financeira do Sistema. Realiza financiamentos de bens de consumo e serviços para pessoas físicas ou jurídicas.
- . empresa de arrendamento mercantil - essa empresa realiza leasing mercantil de máquinas e equipamentos para pessoas jurídicas.
- . empresa de serviços e projetos técnicos à agropecuária - é a empresa que elabora projetos técnicos, que são obrigatórios para a obtenção de financiamentos dirigidos a agro-pecuária (crédito rural), dando assistência técnica e acompanhamento a execução desses projetos.
- . empresa de consultoria, estudos e planejamento - esta empresa elabora projetos técnicos, necessários para a obtenção

de financiamentos para áreas industrial e comercial. Além disso assessora e acompanha a execução desses projetos. Cuida também de todas as operações de repasse de recursos governamentais provenientes do B.N.D.E.S., CEF e BC.

. corretora de seguros - esta empresa coloca à disposição da clientela do Sistema os mais diversos tipos de seguros oferecidos pela seguradora com a qual trabalha.

. Recursos Humanos - constituído por todas as pessoas que trabalham no Sistema. Atualmente a empresa possui um efetivo de 3800 funcionários.

. Recursos Tecnológicos - são todos os equipamentos, ferramentas e Sistemas de informação que auxiliam os recursos humanos no bom desempenho de suas atividades, além das metodologias as quais fornecem procedimentos, que garantem um bom nível de qualidade de trabalho.

2.2.5 Componentes

Na fig. 2.2.5.1 podemos visualizar a estrutura básica do Sistema Financeiro. Ele pode ser dividido em sete subsistemas:

. Subsistema Administrativo - dá apoio aos órgãos do Sistema

quanto ao fornecimento e manutenção dos recursos humanos e materiais necessários ao seu funcionamento. Apropria os meios empregados para a obtenção dos resultados desejados, além de controlá-los.

- . Subsistema_Financeiro - administra e aplica os recursos financeiros em moeda nacional e estrangeira disponíveis em todo o Sistema, garantindo a sua liquidez.
- . Subsistema_de_Riscos - gerencia os riscos envolvidos em todas as operações existentes, estabelece e controla seus níveis aceitáveis de acordo com a política estabelecida pelo Sistema.
- . Subsistema_Comercial - administra e realiza todas as atividades comerciais do Sistema Financeiro. Este é o subsistema que mais tem contato com os clientes, por isso também é o responsável pela administração da carteira de clientes.
- . Subsistema_da_Produção - realiza o tratamento e processamento dos dados provenientes de todas as operações realizadas no Sistema, administra e controla o Sistema de telecomunicações e teleprocessamento, desenvolve e mantém os Sistemas de informação necessários em toda a organização, dá apoio informático a todos os demais subsistemas, desenvolve toda a sistemática e metodologia de trabalho, bem como sua execução, acompanhamento e avaliação.

- . Subsistema de planejamento - assessora os demais subsistemas no controle da gestão e nas atividades de marketing da empresa.
- . Subsistema Jurídico-Fiscal - auxilia os órgãos do Sistema nas funções jurídicas e fiscais.
- . Subsistema de Auditoria - fiscaliza, se os demais subsistemas estão cumprindo com seus objetivos dentro das normas internas e da legislação vigente.

2.2.6 Administração

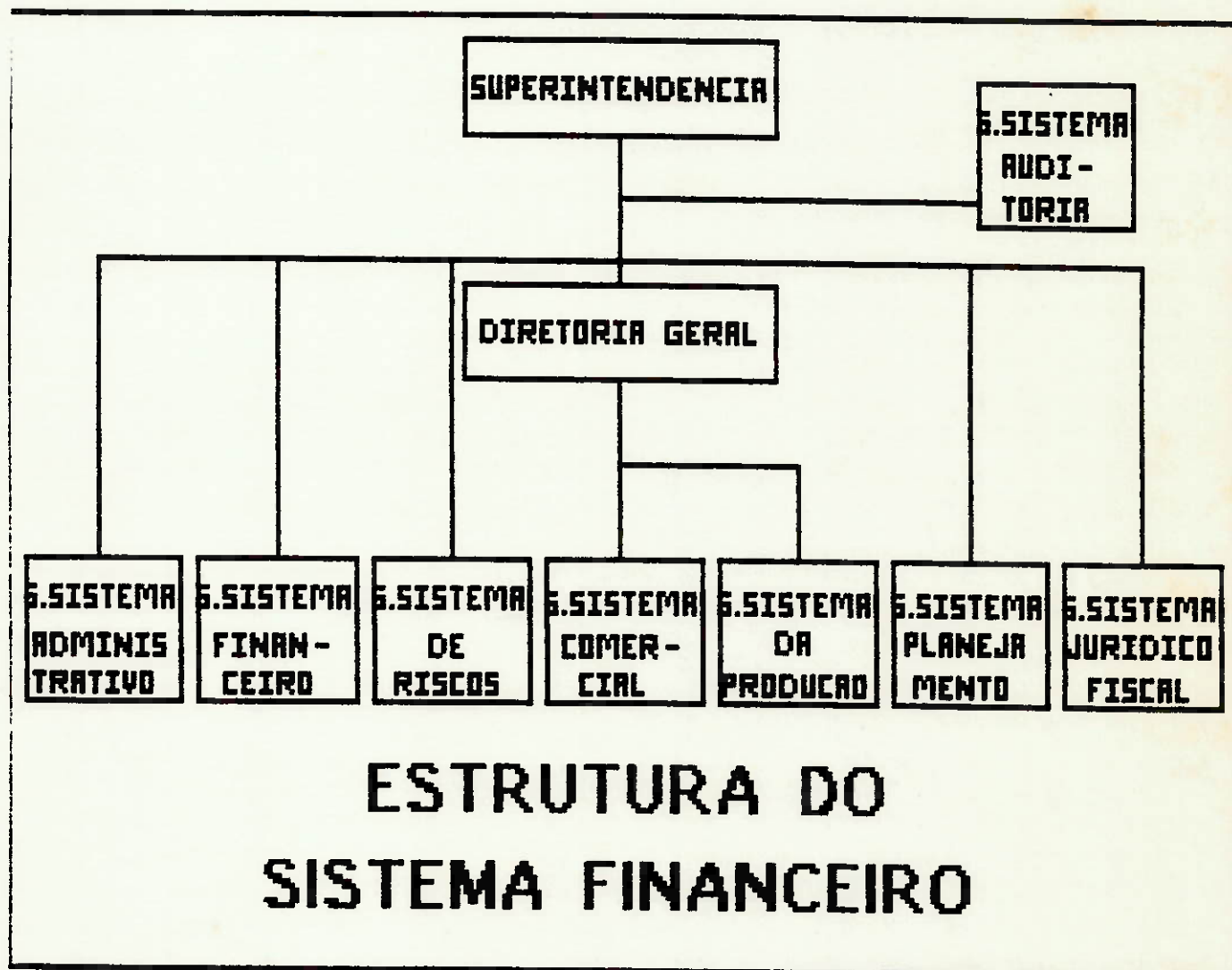
A administração da empresa é realizada por todos os gerentes dos Subsistemas.

Devido à política da empresa, sua administração possui um perfil tradicional em relação a execução de suas tarefas, mas relativamente participativo em relação ao seus funcionários.

Para facilitar a administração, cada gerente de um subsistema possui sob sua subordinação outros gerentes, que se reportam diretamente aos funcionários

O enfoque deste trabalho será dado no Subsistema de Produção, pois é neste subsistema que se encontra o departamento onde efetuamos nosso estágio.

FIG. 2.2.5.1 - Estrutura do Sistema Financeiro.



2.3 SUBSISTEMA DA PRODUÇÃO

2.3.1 OBJETIVOS

Os objetivos deste subsistema devem atender os do Sistema onde está inserido, além disso o subsistema possui objetivos próprios. Os principais objetivos do Subsistema da Produção são:

a. atendimento às necessidades de negócios do Sistema - A Empresa.

b. disponibilidade de dados:

- . ter o conhecimento das informações necessárias para os outros subsistema cumprirem suas missões.
- . dar apoio à ação comercial.
- . manipular as informações para analisar, reportar, apresentar, comparar e decidir.
- . utilizar as informações para simulações e para cálculos.

c. informatização extensiva do subsistema e do Sistema onde está inserido:

- . manter os dados normalizados, atualizados e confiáveis para não haver problemas quando os usuários os utilizarem.
- . garantir uma boa qualidade dos serviços proporcionados aos clientes.
- . procurar sempre reduzir os custos dos produtos e serviços

oferecidos pelo Sistema.

d. serviço a clientes:

- . padronizar ao máximo os tipos de serviços, evitando soluções particulares.
- . possibilitar o acesso do cliente aos dados de seu interesse.

e. racionalização da manutenção:

- . manter a manutenção no nível mais baixo possível.
- . garantir um bom grau de satisfação dos usuários.

2.3.2 AMBIENTE

O ambiente desse subsistema é o Sistema onde ele está inserido: a empresa

2.3.3 RECURSOS

Os recursos dividem-se em:

- . Recursos Humanos - a produção conta com um efetivo de 402 funcionários. As pessoas possuem uma formação em engenharia de produção, em administração, matemática, computação, processamento de dados ou 1º grau completo. Dentre as pessoas que possuem nível técnico 17% são operadores do computador, 30%

são programadores e 53% são analistas.

- . Recursos Tecnológicos - a Produção conta com um aparato informático para desenvolvimento e manutenção dos Sistemas de Informação do banco. Fazem parte desse aparato duas CPU.s (4381-24, 4381-14), duas CCU.s (COMTEN), um concentrador de terminais (3274) e trinta e cinco micro-computadores (PC-AT).
- . Recursos Financeiros - todos os recursos financeiros da Produção provém do Sistema de ordem superior - a empresa - já que esta é o único cliente deste subsistema. A Produção não coloca seus produtos à disposição no mercado.

2.3.4 COMPONENTES

O subsistema da Produção é composto por cinco subsistemas (fig.2.3.4.1):

- . Subsistema Administrativo - realiza todas as funções administrativas da Produção, tais como assistência a nível de custos (elaboração de orçamentos e faturamento mensal, apuração de custos), a nível de materiais (controle de móveis, máquinas e equipamentos) e a nível de pessoal (controle e acompanhamento de efetivo, de salários, de treinamento, de cursos). Este subsistema também dá apoio nos serviços de secretaria e biblioteca.

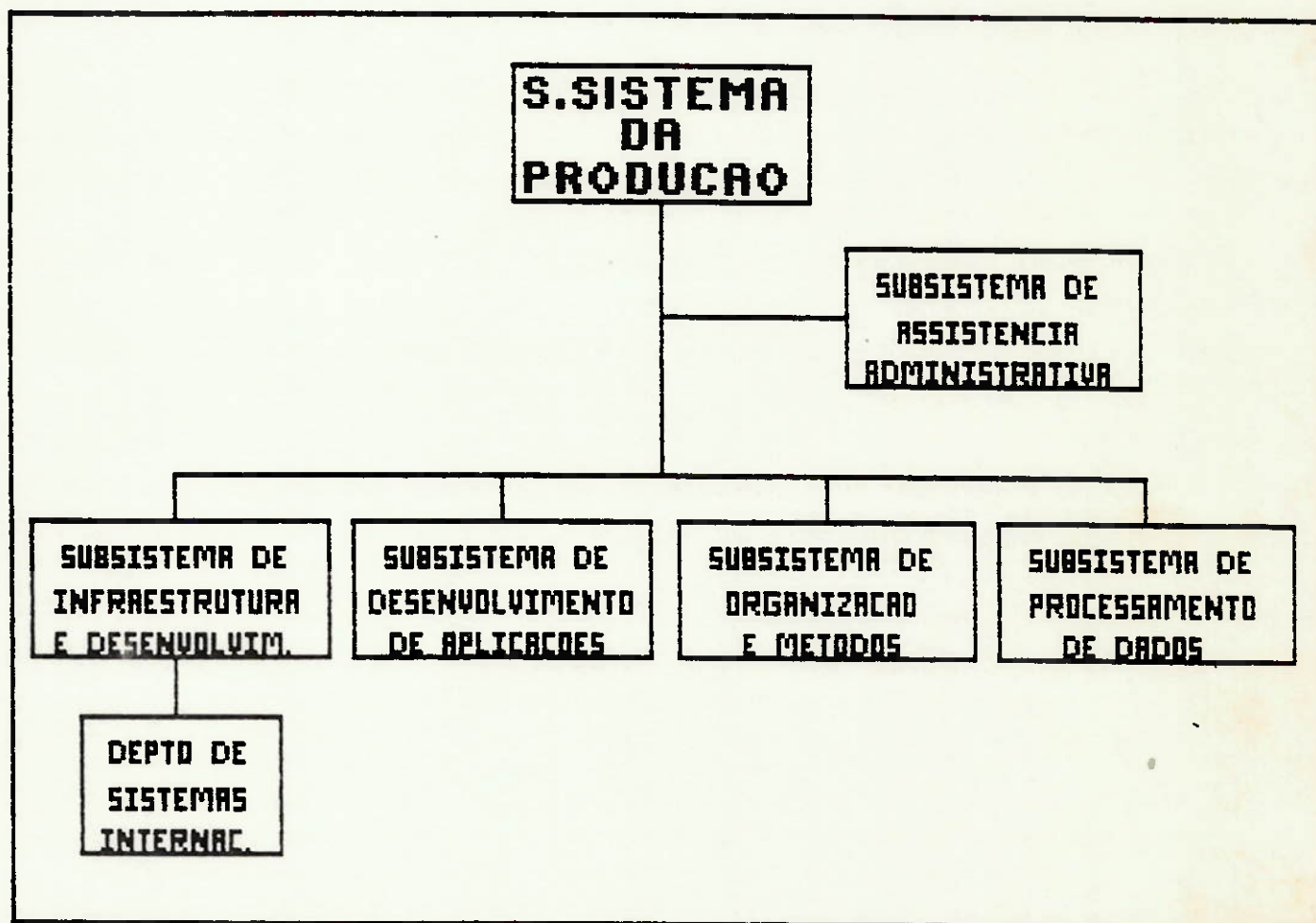
- . Subsistema de Infraestrutura de Desenvolvimento - define a estratégia de hardware, de Software, de formação e desenvolvimento de metodologia. Estabelece a política de prestação de serviços da Produção. Define, cria e mantém os Sistemas de informação bancários básicos. Gere os produtos do Subsistema Financeiro, os produtos envolvidos com moeda estrangeira e os dicionários de Sistemas. O Departamento de Sistemas Internacionais, onde realizamos nosso estágio, é o responsável pela definição, criação e manutenção dos Sistemas que tratam e contabilizam as operações de câmbio e as operações internacionais que envolvem moeda estrangeira.

- . Subsistema de Desenvolvimento de aplicações - define, cria e mantém os demais Sistemas básicos necessários a todos os subsistemas da empresa.

- . Subsistema de Organização e Métodos - trata da racionalização e da organização de toda a empresa, além de confeccionar, difundir e manter os manuais e circulares de normas e procedimentos.

- . Subsistema de Processamento de Dados - digita todos os dados de entrada, prepara os lotes para entrada em processamento, processa e controla os trabalhos em computador, controla e acompanha os periféricos, os arquivos back-up e a rede de comunicações, administra o Sistema de telecomunicações e assiste incidentes técnicos.

FIG. 2.3.4.1 - Estrutura do Subsistema da Produção.



2.3.5 ADMINISTRAÇÃO

A administração do subsistema da Produção possui as mesmas características da administração do Sistema - empresa.

Descreveremos a seguir o Subsistema Câmbio, pois os trabalhos realizados, no departamento a que pertencemos, estão relacionados com esta área. Além disso, nosso objetivo é explicar alguns conceitos desse subsistema, que serão citados ao longo deste trabalho.

2.4 SUBSISTEMA CAMBIO

2.4.1 INTRODUÇÃO

Nossa intenção é descrever o Subsistema Câmbio numa forma bem genérica e sintética, já que outros Trabalhos de Formatura o fizeram de forma bem completa, como é o caso daqueles elaborados por Haffenbaecher [25] ou Bartilotti [4].

O Câmbio é um Sistema muito complexo, pois é regido por muitas normas governamentais, que são alteradas constantemente.

Na maioria das vezes as pessoas confundem o que é Câmbio com sua parte operacional, ou seja, confundem "o quê é feito" com "o como é feito".

Nossa idéia é justamente fazer uma descrição "pura" do Câmbio, ou seja, dos seus conceitos básicos da forma mais simplificada possível.

2.4.2 OBJETIVOS

Os principais objetivos do Câmbio Independente de que empresa faça parte são:

- . atender às necessidades de seus clientes através da boa execução de suas atividades, de seus serviços, garantindo eficiência, bom atendimento e segurança.
- . contribuir para os objetivos globais da empresa a que pertence.

2.4.3 AMBIENTE

O ambiente desse subsistema é constituído pelos seguintes elementos:

- . Mercado Financeiro Internacional - como as transações cambiais no mercado internacional são conduzidas obrigatoriamente através de estabelecimentos bancários, este é o ambiente mais abrangente no qual o Câmbio se insere. Este Sistema é constituído por todas as instituições financeiras

internacionais (bancos comerciais, corretoras, seguradoras, agentes, etc.), pelas associações monetárias internacionais (FMI, GATT, etc.) e pelas autoridades financeiras governamentais de todos os países.

. Mercado_Financeiro_Nacional - este Sistema é constituído por todas as instituições financeiras do mercado nacional (bancos comerciais, corretoras, seguradoras), além das autoridades monetárias governamentais, que regulam o funcionamento do Câmbio a nível nacional. As principais autoridades governamentais relacionam-se abaixo:

. BACEN - o Banco Central expede os normativos da política cambial brasileira. Além de fiscalizar diariamente os bancos, o BACEN é participante ativo de forma a manter o mercado em situação estável e tranqüila.

. CACEX - a Carteira de Comércio Exterior do Banco do Brasil S.A. protege o interesse dos importadores e exportadores brasileiros, oferecendo-lhes todo o tipo de informação sobre o comércio exterior (preços, informações cadastrais, mercados, mercadorias) e autorizando a importação e exportação de mercadorias.

. Mercado_de_Clientes - é o mercado composto por aqueles que desejam realizar uma operação de câmbio. Ele é formado por:

- . pessoas jurídicas

- . pessoas físicas
- . instituições financeiras que operam no mercado nacional
- . empresas financeiras internacionais

O principal elemento do mercado são as pessoas jurídicas.

2.4.4 RECURSOS

- . Capital Próprio - componente do patrimônio líquido da empresa onde está inserido.
- . Recursos Financeiros de Terceiros -
 - . obtidos no país - empréstimos tomados junto a clientes ou a outras instituições financeiras.
 - . obtidos no exterior - proveniente de empréstimos junto a instituições financeiras no exterior.
- . Pontos de Comercialização - composto pelas filiais de câmbio.
- . Recursos Humanos - formado por todos os indivíduos que trabalham nessa área.
- . Recursos Tecnológicos - composto por equipamentos e Sistemas de informação que possibilitam um bom funcionamento do Câmbio.

2.4.5 COMPONENTES

Dividiremos o Câmbio em três subsistemas, geral, em moeda nacional e moeda estrangeira (fig. 2.4.5.1). Faremos isto, utilizando os conceitos da modelagem de dados, onde o problema é descrito de forma abstrata, através da descrição dos dados que o compõe. Esta é uma forma original de o fazer, pois se obtém uma descrição independente da parte operacional do problema.

. Subsistema_Geral -

. Operação Raiz - é o agrupamento de contratos de câmbio e documentos associados (saques, ordens de pagamentos, cartas de crédito) que tenham algo em comum, segundo critério de quem administra a operação.

. Operação Contrato - é o conjunto de dados que descreve o fechamento de um contrato de câmbio qualquer.

. Subsistema_em_Moeda_Estrangeira_(ME) -

. Operação ME tipo Compromisso - é o conjunto de documentos e procedimentos periféricos ao contrato de câmbio, que podem ou não resultar num fechamento de câmbio. Ex: cartas de crédito de exportação, ordens de pagamento recebidas.

. Operação ME tipo Financiamento - são as operações de

financiamento a clientes que resultam necessariamente no fechamento de câmbio. Ex: contrato de financiamento de importação FIRCE.

. **Operação Pontual** - é uma operação de pequeno valor e curta duração que implicam no fechamento de contratos de câmbio. Ex: boletos de compra e venda de Traveller's Checks.

. **Posição ME** - é o saldo em moeda estrangeira atualizado a cada fechamento de contrato de câmbio. Esta posição é limitada pelo BACEN.

. **Operações de Gestão de Contas no Exterior** - são operações que gerem os saldos das contas da empresa, a que pertence o câmbio, no exterior. Ex: operações de overnight em moeda estrangeira, contratação de Loans.

. **Contrato Liquídado ME** - é a liquidação em moeda estrangeira do contrato de câmbio previamente fechado.

. **Movimentação de Contas no Exterior** - são os débitos e créditos efetuados em contas correntes da empresa em bancos no exterior.

. **Fluxo de Caixa em ME (FCME)** - representa a movimentação financeira em moeda estrangeira prevista e efetiva.

. Contas em ME - são as contas de natureza contábil que são afetadas pelos eventos das operações de câmbio relacionados a moeda estrangeira.

. Subsistema em Moeda Nacional (MN) -

. Contas MN - são as contas de natureza contábil afetadas por eventos de operações de câmbio referentes a moeda nacional.

. Operações MN tipo Empréstimo - são operações originadas no fechamento de contrato de câmbio mas não são a ele obrigatoriamente vinculadas. Ex: adiantamento sobre contrato de câmbio de Exportação.

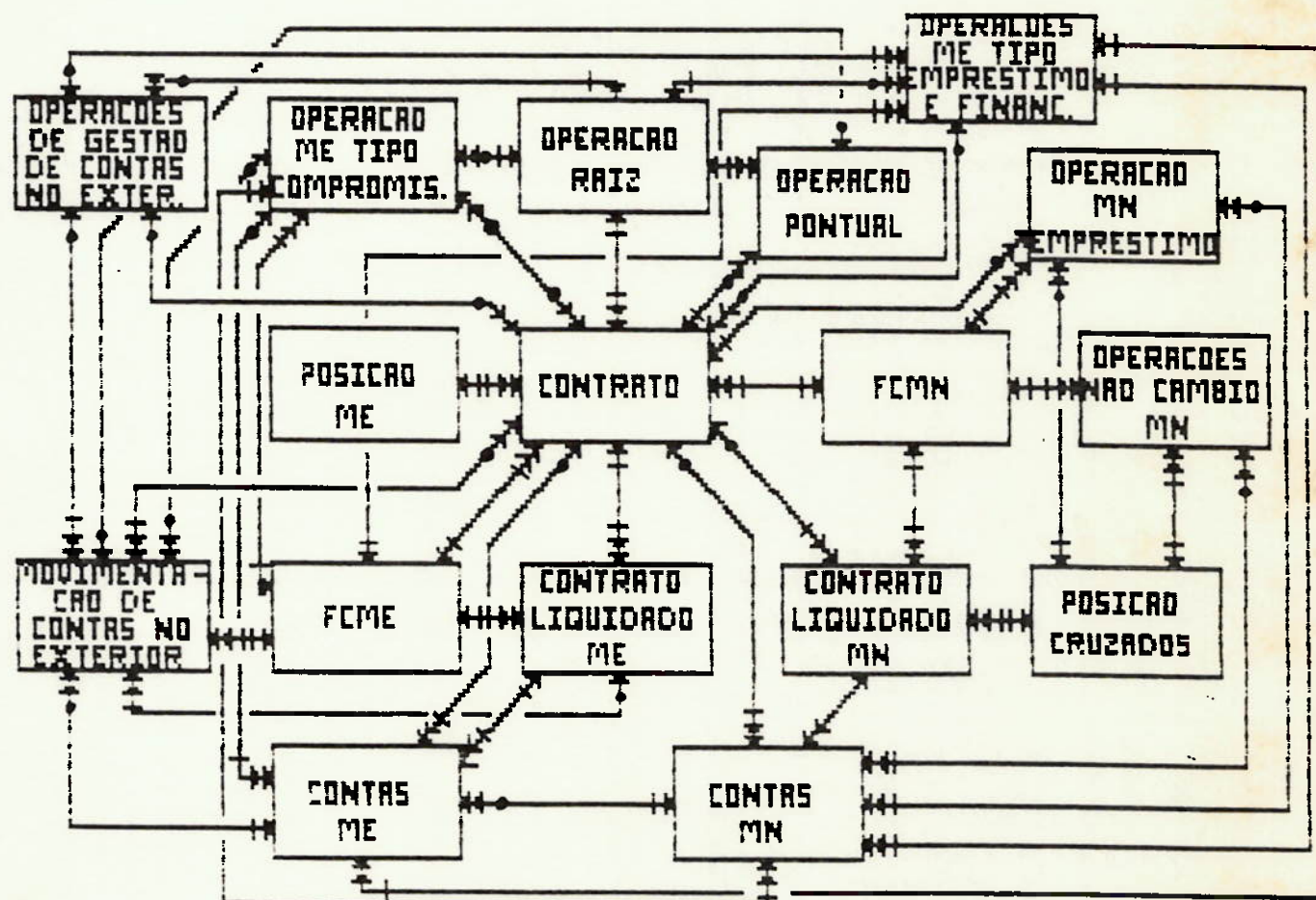
. Contrato Liquídado MN - é a liquidação em moeda nacional do contrato de câmbio previamente fechado.

. Operações Não Câmbio - são as operações de captação ou aplicação dos recursos em moeda nacional necessários as operações de câmbio.

. Fluxo de Caixa em MN (FCMN) - é a movimentação financeira em moeda nacional efetiva e prevista.

. Posição Cruzados - é o saldo econômico da movimentação de moeda nacional das filiais de câmbio.

FIG. 2.4.5.1 - Estrutura básica do Câmbio.



legenda:

.---. .---.

! A !---->>! B ! A está relacionada com uma ou mais
'---' '---' ocorrências de B

.---. .---.

! A !----->! B ! A está relacionada com uma única
'---' '---' ocorrência de B

.---. .---. o círculo sobre a seta indica que A pode
! A !--()->! B ! ou não estar relacionada com uma
'---' '---' ocorrência de B

.---. .---. o traço vertical sobre a seta indica que A
! A !--!-->! B ! necessariamente está relacionada com uma
'---' '---' ocorrência de B

A relação A ---->> B pode ser lida, conforme o caso, por:

- uma ocorrência do componente A necessariamente implica em uma ou várias ocorrências do componente B.
- o componente A é necessariamente afetado pela ocorrência do componente B uma ou várias vezes.

2.4.6 ADMINISTRAÇÃO

A administração do Câmbio é realizada pelos gerentes das filiais de câmbio e pela gerência da divisão de câmbio de cada banco comercial. Aquela varia de acordo com a política administrativa de cada banco comercial.

Esta é uma área fortemente controlada pelo governo, o que diminui muito a flexibilidade da administração

2.5 O ESTÁGIO

O estágio teve início junto ao banco comercial do sistema financeiro no final de julho/87. Primeiramente, fizemos um rodízio pelas áreas do sistema financeiro. Após o rodízio se iniciou efetivamente o estágio na área de informática da empresa, mais especificamente, no Departamento de Sistemas Internacionais. Durante o estágio tivemos uma formação geral, formação em Câmbio, em micro-informática e em main-frame, através de cursos, apostilas e atividades realizadas.

Ao longo do ano de 1988, desenvolvemos este trabalho. Achamos que uma revisão do processo de desenvolvimento de Sistemas seria importante, pois notamos algumas dificuldades nos trabalhos feitos nessa área.

Quando trabalhamos com os demais elementos da equipe sentimos que o trabalho era realizado com uma certa divisão das tarefas. Mas nenhum método específico é utilizado, não existe

nada que garanta que as etapas de trabalho estão sendo feitas corretamente, basta que o Sistema funcione. Mas o quê garante que esses Sistemas vão satisfazer as necessidades do usuário ? Nada o garante, apenas algumas reuniões com o usuário foram feitas para se poder definir o Sistema.

Conversando com alguns usuários ficou claro que:

- . nenhum dos Sistemas existentes os satisfazem completamente
- . apenas 60% das suas necessidades estão sendo satisfeitas
- . os Sistemas atuais da empresa são muito precários em relação aos existentes no mercado
- . o cliente não está satisfeito

Assim, a maioria dos produtos desenvolvidos na área de Informática dessa empresa satisfazem os usuários parcialmente e, na maioria das vezes precisam ser alterados, modificados devido a defeitos ou à própria insatisfação de quem o utiliza. Mais adiante, no item 3.4, detalhamos melhor os problemas encontrados.

Se os problemas de construir um Sistema que não satisfaça o usuário, ou o satisfaça parcialmente, que necessite de uma manutenção relativamente grande (devido a defeitos de projeto, devido a defeitos de fabricação, devido a evolução natural das necessidades, etc.), que possua baixa confiabilidade, não fossem grandes e de custos altíssimos, não nos preocuparíamos com isso. Mas algo deve ser feito para consertar essa situação. Situação essa encontrada também no ramo de Informática do mundo inteiro.

Como podemos ver estamos diante de indícios da "Crise do Software", que é a preocupação comum de vários estudiosos que enfocam a área de produção de Software.

Procuremos então entender melhor o que é essa crise e o que tem sido feito para solucioná-la. Para tal, precisamos primeiro conhecer o ciclo de vida do Software.

3 A CRISE DO SOFTWARE

3.1 Resumo

Neste capítulo é definida a Crise do Software, já que indícios dela foram encontrados na empresa onde este trabalho foi realizado, como mencionado no capítulo anterior.

Além disso, identificamos a Crise do Software no caso em estudo e uma solução para este problema foi mencionada.

Para complementar, foram introduzidos os conceitos do Ciclo de Vida do Software.

3.2 O Ciclo de Vida do Software

Para melhor entendermos a Crise do Software, descreveremos as etapas de desenvolvimento de Sistemas, ou seja, o ciclo de vida do Software. Muitos conceitos aqui presentes serão utilizados posteriormente.

Segundo Sommerville [80], o primeiro modelo provável do Ciclo de Vida do Software foi proposto por Royce em 1970. Podemos dizer que este é um modelo clássico, outras variações existem. Posteriormente, descreveremos outra versão do ciclo de vida proposto por Pressman [45].

De forma simplificada, o ciclo de vida do Software é dividido nas seguintes etapas de desenvolvimento:

- . **Definição e Análise dos Requisitos** - os serviços prestados pelo Sistema, suas restrições e seus objetivos são estabelecidos de acordo com as necessidades do usuário, através de reuniões com este. Uma vez definidos os elementos acima, eles precisam ser descritos de forma compreensível tanto aos usuários como à equipe de desenvolvimento.
- . **Projeto do Sistema e do Software** -
 - . **projeto do Sistema** - utilizando a definição dos elementos da etapa anterior como base, os requisitos são distribuídos pelos Sistemas de Hardware e de Software.
 - . **projeto do Software** - é o processo de representação

das funções de cada Sistema de Software, de tal forma, que essa representação esteja pronta para ser transformada em um ou mais programas. Essa etapa pode ser dividida em três fases:

1. associação dos componentes abstratos do Software com os serviços, determinados na etapa de definição dos requisitos e especificação precisa de cada componente abstrato.
 2. construção de um primeiro nível de projeto, mostrando como os componentes do Software se relacionam.
 3. formulação de um projeto detalhado para cada componente abstrato. Este é expresso através de linguagem simbólica, ou seja, de elementos que possibilitem a transformação imediata para o código.
- . **Implementação e Testes Unitários** - nesta etapa o projeto do Software é encarado como um conjunto de programas, que são escritos numa linguagem de programação executável. Os testes unitários envolvem a verificação do funcionamento de cada programa, e se ele está de acordo com a especificação definida.
- . **Testes do Sistema** - todos os programas do Sistema são integrados e testados como um todo, a fim de se certificar que os requisitos do Sistema foram atendidos. Após os testes o Sistema é colocado à disposição do usuário.
- . **Operação e Manutenção** - normalmente esta é a etapa mais longa da vida do Sistema. O Sistema é instalado e

colocado em uso. Ao longo da operação do Sistema surge a necessidade de manutenção, a qual pode envolver a correção de erros não detectados nas etapas anteriores, a complementação de unidades do Sistema ou o incremento de novos serviços pela existência de novas necessidades.

Gostaríamos de deixar claro, que essa divisão em etapas do Ciclo de Vida do Software é proveitosa, mas na prática essas etapas se superpõem e fornecem informações umas às outras.

Pressman fornece uma outra visão do Ciclo de Vida do Software. Procuraremos detalhar um pouco mais essa visão.

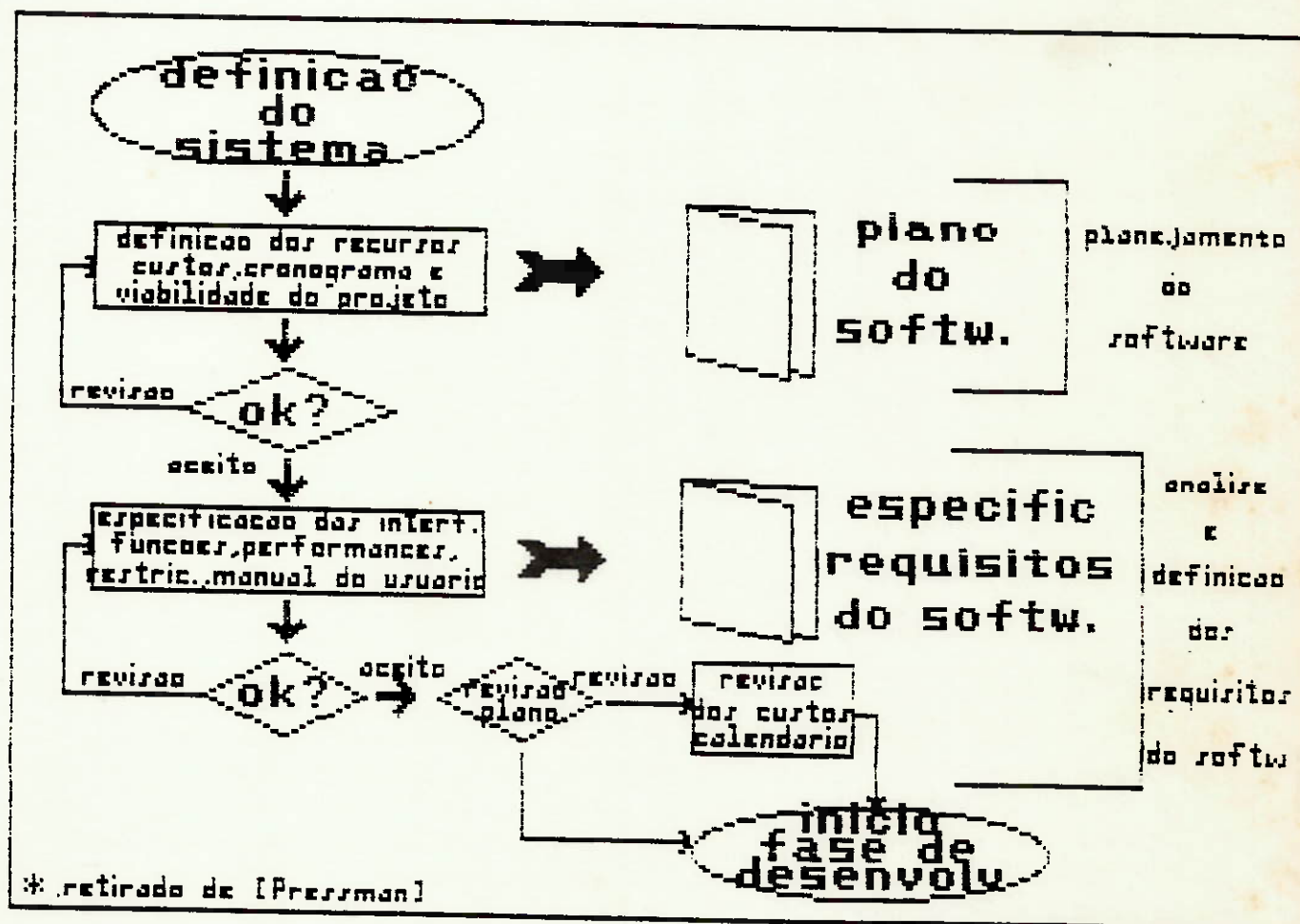
O Ciclo de Vida se divide em três fases principais, que são: planejamento, desenvolvimento e manutenção.

1. Fase de Planejamento - esta fase (fig.3.2.1) é dividida em outras três etapas:

- . **definição do Sistema** - nesta etapa o Sistema é examinado como um todo. São definidos o conceito básico do Sistema, suas interfaces, suas funções e performance, a alocação das funções no Hardware e no Software e a determinação dos custos e restrições de planejamento.
- . **planejamento do Software** - nesta etapa são determinados o esforço necessário para o desenvolvimento do Software, ou seja, o tipo de mão-de-obra e respectivas quantidades. Faz-se, também, uma previsão dos recursos necessários ao desenvolvimento do Sistema, estima-se os custos e um cronograma de trabalho. O principal objetivo desta etapa é indicar a viabilidade ou não do projeto em relação a seus custos e restrições de tempo.

. análise e definição dos requisitos do Software - nesta etapa o analista, juntamente com o usuário, detalham os elementos do Software. Através do fluxo e da estrutura dos dados do Sistema consegue-se especificar as interfaces e as características funcionais de cada elemento do Sistema.

FIG. 3.2.1 - Esquema da Fase de Planejamento do Ciclo de Vida do Software.

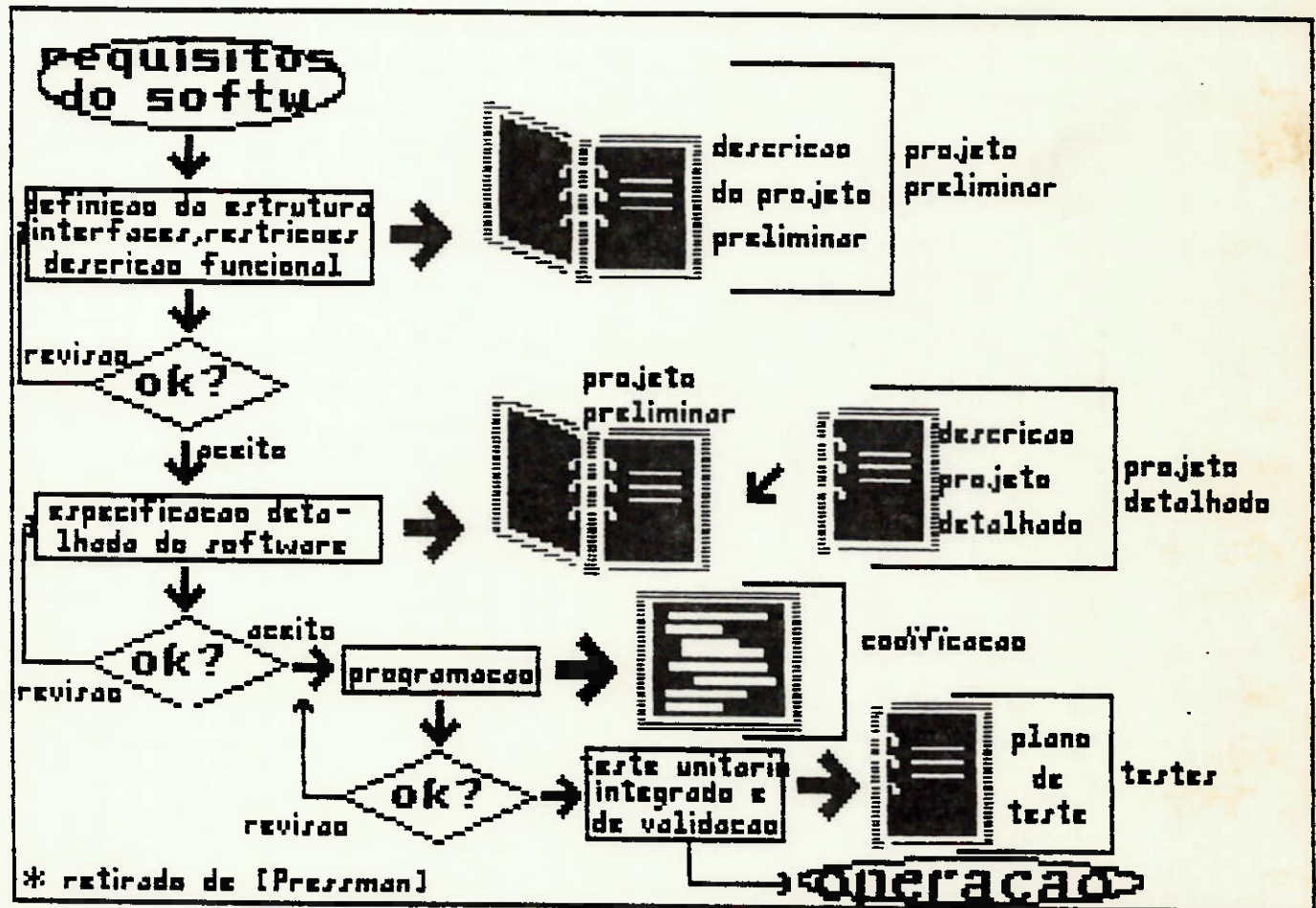


2. Fase de Desenvolvimento - basicamente o que se faz nesta fase é transformar um conjunto de requisitos, previamente especificados, em um elemento do Sistema, chamado de Software (fig.3.2.2). Esta fase divide-se em quatro etapas. São elas:

- . projeto preliminar - a partir dos requisitos definidos na etapa anterior e através da utilização de métodos de desenho, consegue-se obter uma representação do Software, chamada de estrutura ou arquitetura do Software. Além disso, essa etapa resulta nas interfaces entre os elementos internos do Software e externos do Sistema, numa descrição funcional de cada elemento e numa estrutura detalhada dos dados.
- . projeto detalhado - através de representações adequadas de desenho (gráficas ou textuais) cria-se uma especificação detalhada do Software. Essa descrição deve conter todas as informações necessárias para que uma pessoa, que não seja o projetista, tenha condições de codificar o Software.
- . codificação - após as duas etapas de projeto se realiza a codificação, que consiste na geração de um programa através de uma linguagem de programação apropriada. A codificação deve ser uma simples tradução do projeto, então é a qualidade do projeto que determina a qualidade do programa.
- . testes - existem três tipos de testes a serem realizados nessa etapa. O teste unitário - feito para verificar a performance funcional de um componente do Software. O

teste integrado - testa as funções e interfaces do Software como um todo, integrado. O teste de validação - verifica se todos os requisitos do Software estão sendo atendidos. Para cada um dos tipos de testes devem ser elaborados um plano e um procedimento de execução de testes. Após os testes e correções necessárias o Software é liberado para a operação.

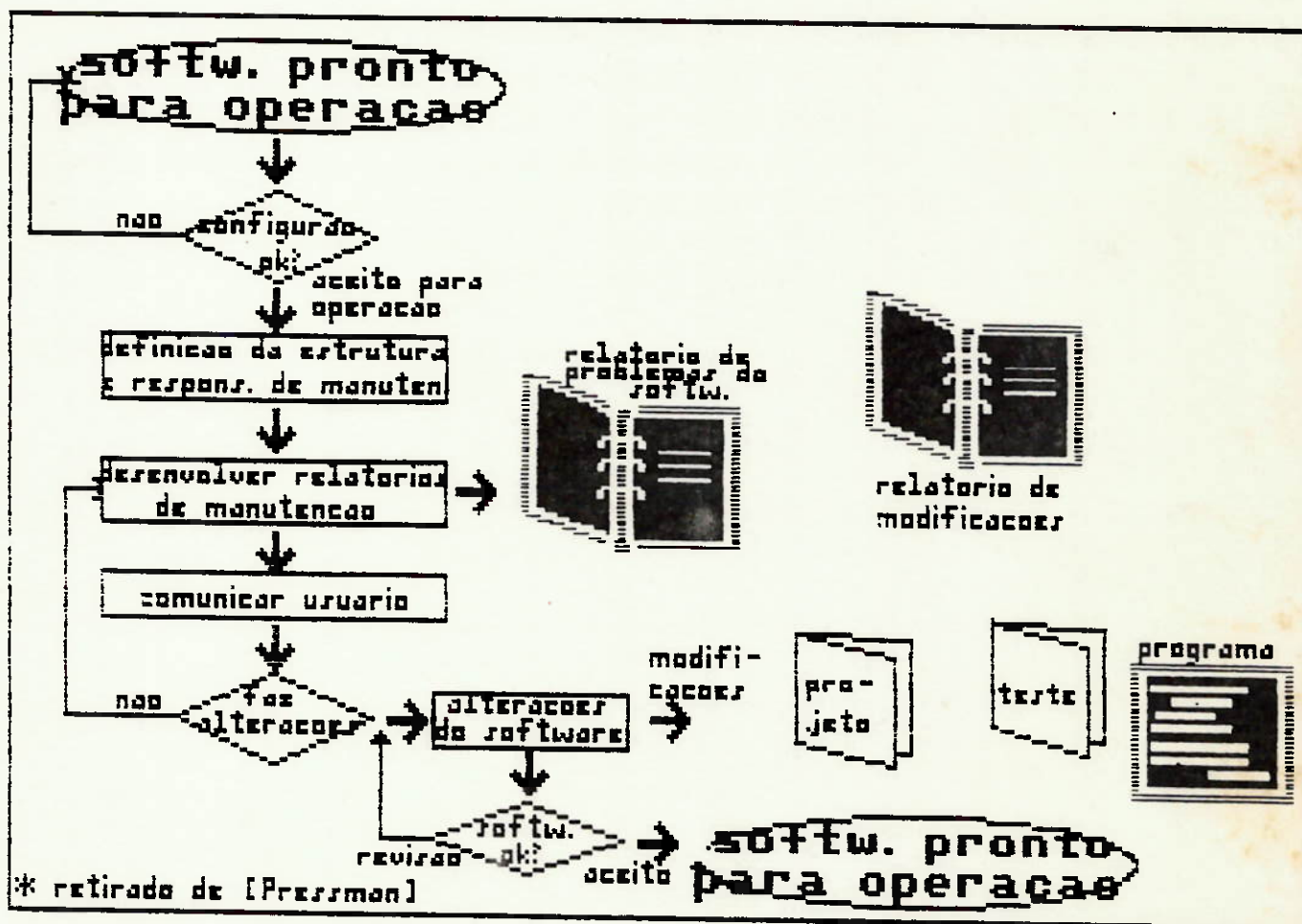
FIG. 3.2.2 - Esquema da Fase de Desenvolvimento do Ciclo de Vida do Software.



3. Fase de Manutenção - essa fase se inicia antes mesmo da liberação do Software para operação, com a revisão da configuração deste, a fim de se assegurar para as tarefas de manutenção que surgirão, que toda a documentação está compatível com o Software desenvolvido.

As tarefas de manutenção dependem do tipo de manutenção a ser realizada (corretiva, adaptativa ou de aperfeiçoamento). Mas em todos os casos a modificação do Software inclui a modificação de toda a configuração (documentos desenvolvidos na fase de planejamento e desenvolvimento) e, não apenas a modificação do programa (fig.3.2.3).

FIG. 3.2.3 - Esquema da Fase de Manutenção do Ciclo de Vida do Software.



3.3 Definição da Crise do Software

Como podemos perceber, a própria definição do ciclo de vida do Software é uma tentativa de solucionar os problemas de construção destes, na medida em que passa a haver a divisão do trabalho em etapas. Logicamente, essa preocupação existe desde o surgimento do computador. Mas a Crise do Software, preocupação que passou a existir desde o início dos anos 70 na construção de sistemas de grande porte e complexos (mais de 5000 linhas de código), é caracterizada por certos problemas, que não foram solucionados com a simples aplicação dos conceitos do Ciclo de Vida do Software.

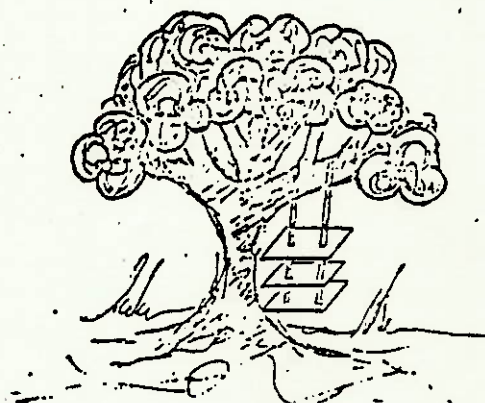
A Crise do Software é caracterizada pela detecção dos seguintes problemas:

- . os custos relativos dos componentes de Hardware vêm decrescendo drasticamente ao longo dos anos (de 100% na década de 50 para 10% na década de 90), ao passo que os custos relativos do Software vem crescendo na mesma proporção.
- . um fator que contribui significativamente para a crise do Software é o fato da complexidade dos Sistemas estar aumentando no decorrer do tempo.
- . os custos de manutenção de um Sistema, desenvolvido através do ciclo de vida, são em geral duas a quatro vezes maiores do que os previstos inicialmente. Onde dois-terços destes custos são devido a não identificação das necessidades reais do usuário ou devido a um projeto

conceitual Impróprio do Software.

- . 45% dos problemas do Software, que necessitarão de manutenção, são apenas detectados após o término dos testes de aceitação do Sistema.
- . o custo de problemas detectados durante a fase final de testes ou na fase de operação é de cinqüenta a cem vezes maior do que o custo dos problemas detectados durante a fase de especificação dos requisitos.
- . os Sistemas não são confiáveis, seu nível de qualidade é baixo.
- . é muito freqüente a insatisfação do usuário com o Sistema. Normalmente, os Sistemas contêm apenas pouquíssimos requisitos propostos pelo usuário, ou seja, os Sistemas de modo geral não estão atendendo às necessidades do usuário. A fig. 3.3.1 ilustra este fato, com o exemplo da árvore e gangorra.
- . na maioria das empresas a demanda por novas aplicações na área está crescendo muito e a capacidade de atendimento está aumentando numa proporção muito inferior. Esse aumento de demanda se dá muito pelo fato dos usuários estarem cada vez mais descobrindo as potencialidades do computador. Isso gera grandes filas de espera. Assim uma aplicação, que se faz necessário numa empresa atualmente, só ficará pronta após dois a quatro anos.

FIG. 3.3.1 - Um Exemplo Clássico do Não Atendimento das
Necessidades do Usuário.



AQUILO QUE O USUÁRIO QUER



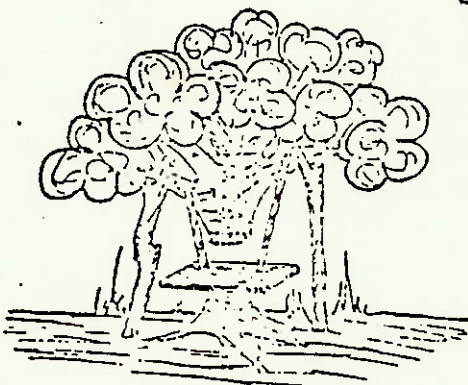
AQUILO QUE O CHÉFE DE PROJETO ENTENDEU



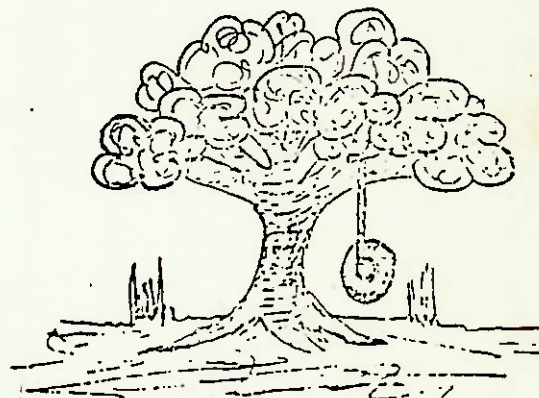
AQUILO QUE O ANALISTA PREVIO



AQUILO QUE O PROGRAMADOR FEZ



AQUILO QUE OS ULTIMOS ACERTOS FIZERAM



AQUILO QUE ERA PRECISO

- . o tempo utilizado para o desenvolvimento de novos Sistemas é muito grande, basicamente, devido a baixa produtividade. No caso de Sistemas complexos este tempo pode variar de dois a cinco anos. Assim, uma necessidade que o usuário possui atualmente só será atendida na melhor das hipóteses em dois anos; até isto acontecer pode ser que essa necessidade não exista mais ou tenha sofrido alterações.

São várias as causas desses problemas. Muitas delas dependem até da empresa, da forma de trabalho da equipe de desenvolvimento. Outras fazem parte dos conceitos e técnicas aplicados.

Poderíamos dizer que as principais causas da crise do Software são as seguintes:

- . a comunicação entre o usuário e a equipe de desenvolvimento é deficiente. Logicamente, cada um possui sua forma de se expressar e, normalmente, esta forma não é totalmente compreensível ao outro; isto é, a comunicação humana não é formal, estando sujeita a vários fatores subjetivos que conduzem a diferentes interpretações de uma mesma mensagem. Existem recursos que auxiliam essa comunicação, como o caso da prototipação que quando utilizada, se dá de forma precária.
- . a equipe de trabalho usualmente possui pouca formação nas técnicas mais atuais de desenvolvimento de Software, cometendo inclusive erros devido a vícios, que afetam a qualidade e a manutenção dos produtos resultantes.
- . é notório a falta de uma cultura nas empresas, que

estabeça um treinamento e condições a fim de que o funcionário aprenda e aplique as técnicas que seu superior acha conveniente.

- . talvez a principal causa da crise do Software seja a resistência às mudanças que todas as pessoas possuem. Enquanto o potencial do Hardware vem mudando num ritmo imenso, as pessoas da área de Software bloqueiam esse potencial através da oposição a mudanças introduzidas ou discutidas.
- . Sistema é produto de um trabalho intelectual, com isso ele está sujeito às condições pessoais de quem o desenvolveu (experiência, estado psicológico, etc.).
- . as pessoas que trabalham na área acreditam em alguns mitos, os quais são um dos fatores que em muito contribuem para os problemas da crise do Software. Eles propagam desinformação e confusão. Um exemplo é a idéia de que um conhecimento geral dos objetivos do Sistema já é suficiente para se iniciar a programação; outro mito é pensar que primeiro se programa e depois se faz o Software funcionar, então o trabalho está feito.
- . a maioria dos programas aplicativos são projetados individualmente, codificados manualmente, com métodos muito vagarosos. Chega a ser uma confecção artesanal; assim, a produtividade fica muito prejudicada.

Algumas soluções foram propostas para eliminar essas causas e, conseqüentemente, os problemas que originam a crise do Software.

- . para melhorar a comunicação entre o usuário e a equipe de desenvolvimento, foi proposto a utilização da prototipação (mais adiante no item 5.3.3, veremos em maiores detalhes no que isto consiste). Através desse recurso o usuário consegue visualizar o que seria o Sistema desenvolvido, consegue ver quais de suas necessidades o Sistema projetado está ou não atendendo. Dessa forma, os requisitos do Sistema são identificados corretamente de forma iterativa. Evita-se que a definição deste não seja correta e que seja necessário uma manutenção futura maior. Muitos problemas passam a ser detectados na fase de definição e não mais na fase de operação.
 - . direcionar a confecção de Sistemas para o usuário, ou seja, construí-los tendo como prioridade as necessidades e dificuldades do usuário, ao invés das vantagens e dificuldades de programação. Se possível, seria ideal que o próprio usuário desenvolvesse suas aplicações.
 - . atendendo essa última idéia surge o conceito do Centro de Informações (C.I.) que teria a função de treinar, encorajar, suportar e assistir os usuários finais na obtenção das aplicações que estes necessitam. Além disso o C.I. possuiria uma equipe de suporte técnico que faria a confecção e suporte de Sistemas, resolução de problemas técnicos e de Software, seleção de ferramentas, linguagens, grandes aplicativos (Sistemas de 4ª geração) e Hardware, etc.
- Existem soluções intermediárias, por exemplo, ensinar ao

usuário técnicas de especificação de Sistema (modelagem de dados, DFD, etc.) para criar uma linguagem formal comum entre usuário e analista. Isto possibilita ao usuário fazer a especificação do Sistema sozinho e passar ao analista, ou dá condições que o usuário desenvolva a especificação em estreita colaboração com o analista.

Enfim, o C.I. tem a função de dar suporte ao usuário em todos os níveis, a fim do próprio usuário desenvolver suas aplicações, seus Sistemas.

- . para se obter melhores índices de produtividade e maiores níveis de qualidade, propõe-se a utilização de princípios de engenharia, ou seja, a utilização de metodologias. Estas especificam os problemas a serem tratados, indicam como resolvê-los, garantem que as fases de desenvolvimento sejam feitas de forma ordenada e com níveis de qualidade especificados, através do fornecimento de métodos, ferramentas e procedimentos.

3.4 A Crise do Software na Empresa

Fizemos diversas entrevistas com os gerentes dos departamentos de desenvolvimento de Sistemas e com membros da equipe, para levantarmos os dados que caracterizam a crise do

Software dentro da empresa. Salientamos que muitos desses dados não existem formalmente, mas os gerentes, devido a seus conhecimentos, os quantificaram aproximadamente.

Dentre os aspectos que caracterizam a crise do Software já mencionados, foram encontrados no caso em estudo os seguintes pontos:

- . os Sistemas são pouco confiáveis, pois a quantidade de horas utilizadas para manutenção na empresa é superior às horas utilizadas para desenvolvimento, sendo essa relação de aproximadamente 60% de manutenção e 40% de desenvolvimento como mostram os dados abaixo.

SEMESTRE	HORAS TRABALHADAS	

	MANUTENÇÃO	DESENVOLVIM.

1º SEM / 87	30684,3	16263,0

2º SEM / 87	31855,9	22307,8

1º SEM / 88	34461,0	16838,5

- . a maior parte dos Sistemas são entregues com atraso. Em alguns departamentos todos os Sistemas são implantados atrasados. A média desse atraso chega a ser de quatro

meses.

- . em todos os departamentos existe uma defasagem de tempo grande entre a constatação da necessidade de um Sistema e sua implantação. No caso do departamento onde realizamos estágio, que cuida dos Sistemas para câmbio, os quais são muito complexos, a construção de um novo Sistema levaria cerca de cinco anos; só a fase de definição do Sistema, recém concluída demorou dois anos. Em Sistemas menos complexos essa defasagem de tempo é menor e em média de um ano e meio. Pequenas alterações dos Sistemas já existentes levam em média de um a dois meses até serem completadas, o que é um tempo razoável, comparativamente ao porte das modificações e aos transtornos ocasionados até seu término.
- . o usuário da Informática está de modo geral insatisfeito, principalmente, em relação ao volume de necessidades que estão sendo atendidas, que ainda é baixo. Alguns usuários da área de câmbio, os quais entrevistamos, fizeram alguns comentários. Eles estão insatisfeitos com os Sistemas, pois estes são obsoletos em relação aos existentes no mercado e sem nenhuma padronização, o que dificulta o aprendizado para utilização. Nenhum Sistema os satisfaz completamente, apenas 60% das suas necessidades estão sendo atendidas. As informações fornecidas pelos Sistemas são defasadas no tempo em pelo menos um dia, não refletindo a situação atual, o que muitas vezes dificulta as transações comerciais. Também

o cliente da empresa não está satisfeito com os serviços informáticos.

Dentre as causas já citadas, foram detectadas as seguintes:

. não existe um ciclo de vida do Software formalmente definido; cada departamento divide o trabalho em etapas, segundo sua própria interpretação do ciclo de vida . Esta interpretação, por vezes, consiste em uma simplificação das etapas definidas anteriormente (ítem 3.2). Isso vem dificultar a comunicação entre os membros da equipe de desenvolvimento, uma vez que suas tarefas não são bem definidas. A documentação e o próprio Sistema ficam falhos e requisitos podem ser esquecidos, pois não existe nenhum critério para se avaliar se o nível de qualidade desejado foi atingido para se permitir a evolução do desenvolvimento do Sistema.

. nenhuma metodologia específica é utilizada, cada gerente escolhe algumas ferramentas, que julgue ser útil e faz a equipe aplicá-las. Isso gera problemas, pois não basta aplicar ferramentas isoladas. Um procedimento precisa ser estabelecido, de forma a garantir os níveis de qualidade desejados. As metodologias consistem, justamente, de um conjunto de procedimentos para a utilização destas ferramentas. Assegura-se assim um nível mínimo de qualidade do Sistema assegurado. Nada garante, que as ferramentas seleccionadas pelos gerentes e a forma escolhida para aplicá-las, seja a melhor opção possível. Tal fato é comprovado pelo nível de manutenção que a

empresa realiza em seus Sistemas.

- . a equipe de desenvolvimento de modo geral não possui nenhuma formação específica em técnicas mais avançadas de desenvolvimento de Sistemas, tais como prototipação, metodologias para desenvolvimento de Sistemas, utilização de diagramas e de CASE (será posteriormente detalhado). Quando o indivíduo possui alguma formação nessas técnicas, ela é básica e quase sempre teórica. As pessoas também não acreditam muito nas vantagens que elas proporcionam. Para completar não existe nenhum programa de treinamento sistemático na empresa para desenvolver as pessoas nessas técnicas. A informática é um ramo cuja tecnologia avança assustadoramente rápido, novos recursos estão sempre surgindo, se as pessoas não procurarem se atualizar, estarão fazendo coisas de forma já ultrapassada e cometendo muitas vezes erros que poderiam ser evitados.
- . existe dificuldade de diálogo com o usuário, por isso, dentre outros motivos, nenhum Sistema o satisfaz totalmente. O nível de utilização de prototipação ainda é baixo, apenas telas são apresentadas ao usuário de forma não iterativa.
- . no que se refere a elaboração de Sistemas, não há padronização entre os diversos departamentos de desenvolvimento e nem entre os Sistemas de um mesmo departamento. Isto dificulta muito o processo de manutenção daqueles, pois não havendo uma padronização fica difícil que pessoas, que não desenvolveram o Sistema,

façam sua manutenção.

- . resistência às mudanças existe. Ao invés das pessoas aplicarem os recursos disponíveis, na maioria das vezes elas fazem da forma como estão acostumadas. Elas também só utilizam um novo recurso proposto, se elas próprias comprovarem suas vantagens; mas o critério de avaliação de um membro da equipe nem sempre é amplo o suficiente, para perceber as vantagens de um recurso.
- . não está sendo utilizada na empresa uma cultura, através da qual sejam dadas condições para que o funcionário aprenda as ferramentas, que seu superior deseja que sejam utilizadas. Desta forma, as pessoas acabam realizando o trabalho da forma como estão acostumadas, porque o tempo é limitado.
- . realmente os indivíduos acreditam em mitos. Seleccionamos alguns mitos principais e, perguntamos aos entrevistados se eles acreditavam ou não no que estávamos falando. Dentre os mitos apresentados 60% dos mitos em média foram aceitos.

3.5 Uma Proposta de Solução

Todos os problemas e causas detectados no caso em estudo estão presentes na área de informática de qualquer tipo de empresa, ou seja, não são problemas específicos de uma firma mas de toda uma área, onde segundo a revista Exame são investidos

anualmente no Brasil bilhões de dólares (em 1986 3 bilhões de dólares e em 1987 3,3 bilhões de dólares); na empresa em estudo foram dispendidos no ano de 1987 na informática 12,5 milhões de dólares, representando 26,3% do orçamento da empresa. Por este motivo salientamos a importância da solução destes problemas.

Ao analisarmos os problemas, suas causas e as soluções propostas, chegamos à conclusão de que é necessário reformular os conceitos existentes.

O desenvolvimento de Sistemas tem sido encarado como uma produção artesanal, onde o funcionário, artesão realiza todas as funções para produzir um produto. Além disso, predominam os mecanismos de "iniciativa e incentivo" do trabalho, ou seja, fica a critério do trabalhador a escolha do melhor e mais econômico método para realizar o trabalho, enquanto cabe ao administrador utilizar de meios para incentivar o funcionário a produzir o máximo.

Como podemos constatar, estamos em pleno século XX diante dos conceitos de trabalho das Corporações de Ofício. Está na hora de encararmos a produção de Sistemas como um processo industrial.

Em suma, a causa maior, subjacente à Crise do Software e nunca enunciada de forma clara, é a organização do trabalho de forma artesanal. Uma possível solução, a qual vem sendo enunciada e aplicada de uma ou outra forma desde a década de 70, é a aplicação dos princípios da Administração Científica de

Taylor (salientamos que são utilizados os princípios e não os meios de aplicá-los). Faremos uma análise crítica desta solução e de outras no Capítulo 7.

Dificuldades também existem, em se aplicar os princípios propostos por Taylor na área de desenvolvimento de Sistemas, já que este é um trabalho basicamente intelectual. As principais dificuldades existentes são:

- . **comunicação / especificação** - num trabalho intelectual a divisão do trabalho implica numa especificação das tarefas que depende muito da comunicação, pois essa especificação é feita através de palavras (oral ou escrita). Assim, nem sempre o que uma pessoa quis dizer é o que a outra entendeu. Dessa forma, existe uma dificuldade muito grande na especificação das tarefas, e muitas vezes os trabalhos resultantes não são o que se pretendia obter.
- . **diversas soluções** - em qualquer tipo de trabalho normalmente existe mais de uma solução, mas no caso do trabalho intelectual as soluções variam muito e dependem das experiências de cada pessoa. Assim, vivências diferentes implicam em soluções diferentes. Dessa forma fica muito difícil padronizar o trabalho.
- . **envolvimento psicológico** - o desempenho num trabalho intelectual depende muito do estado psicológico do funcionário, da sua concentração, tranqüilidade e satisfação pessoal.
- . **conhecimento especializado** - o nível de qualidade do trabalho depende dos conhecimentos prévios do funcionário,

das suas experiências anteriores.

Apesar dessas dificuldades acreditamos que através de recursos específicos, que posteriormente definiremos, seja possível aplicarmos os princípios da Administração Científica de Taylor.

Procuremos recapitular os princípios dessa teoria.

Primeiro Princípio - "à gerência é atribuída ... a função de reunir os conhecimentos tradicionais que no passado possuíram os trabalhadores e então classificá-los, tabulá-los, reduzi-los a normas, leis ou fórmulas, grandemente úteis ao operário para a execução do seu trabalho diário."

Além disso, Taylor utiliza a técnica da "análise científica" do trabalho, ou seja, o estudo dos tempos que possui a fase analítica e a fase construtiva. Basicamente, essas duas fases são constituídas da divisão do trabalho em movimentos elementares, do exame dos movimentos e da combinação deles na forma ótima. Ou seja, Taylor propõe um estudo do trabalho a ser executado de tal forma, que seja realizado uma divisão do trabalho em tarefas, que devem ser executadas numa forma ótima também estabelecida e padronizada.

Podemos dizer, que este princípio já vem sendo aplicado, uma vez que a gerência possui o conhecimento dos processos produtivos e existe a divisão do trabalho em tarefas, que devem ser executadas na forma ótima conforme determinado pela gerência, que é a abordagem clássica do Ciclo de Vida do Software.

O que falta é a padronização ou uma visão comum de como esta divisão do trabalho deve ser feita para todos os departamentos de desenvolvimento de Sistemas da empresa.

Segundo Princípio - "... a seleção, então, não consistiu em achar homens extraordinários, mas simplesmente em escolher entre homens comuns os poucos especialmente apropriados para o tipo de trabalho em vista."

Uma vez dominado o conhecimento do trabalho pela gerência, especificadas as tarefas, deve-se escolher as pessoas que possuem as qualidades necessárias para a boa execução das tarefas. Além disso, é necessário treinar o funcionário, para executar as tarefas conforme o determinado pela gerência.

Atualmente as empresas, ao selecionar seus funcionários, escolhem aqueles indivíduos que possuem as características necessárias à execução do trabalho. No entanto, normalmente, não há uma política de treinamento e formação do funcionário, onde sejam dadas condições para que este aprenda e aplique os novos conhecimentos.

Terceiro Princípio - "A idéia da tarefa é, quiçá, o mais importante elemento na administração científica. O trabalho de cada operário é completamente planejado pela direção, pelo menos com um dia de antecedência, e cada homem recebe, na maioria dos casos, instruções escritas completas que minudenciam a tarefa de que é encarregado e também os meios usados para realizá-la... A administração científica, em grande parte, consiste em preparar e fazer executar essas tarefas."

As funções de planejamento e controle da produção passam a fazer parte da gerência e não mais do funcionário. Esse princípio já vem sendo aplicado em parte. Uma vez que o funcionário, na maior parte das vezes, realiza o trabalho da forma como está acostumado, e dentro dos departamentos não existe uma padronização de como desenvolver os Sistemas. Assim, a gerência não possui o controle da produção, apenas o planejamento.

As vantagens de aplicarmos os princípios de Taylor são várias. Cada princípio traz uma vantagem específica, vejamos quais são.

Ao aplicarmos o Primeiro Princípio, a gerência passa a ter o controle da produção, a qual não é mais realizada de qualquer forma, mas sim na melhor forma existente. Com a divisão do trabalho deixamos de ter um processo artesanal, onde tudo depende de um só indivíduo e da forma como ele executa seu trabalho, para termos um processo produtivo industrial, com melhor produtividade e melhores níveis de qualidade.

A vantagem obtida através do Segundo Princípio é, basicamente, o fato de passarmos a possuir funcionários aptos e devidamente treinados para a boa execução das tarefas. Desta forma estaremos eliminando uma das causas da crise do Software: o não conhecimento das novas técnicas de informática por parte da equipe de desenvolvimento.

Com a aplicação do Terceiro Princípio a gerência passará a ter de fato o controle da produção dos Sistemas, podendo assim evitar os problemas de atraso e demora para adaptações do Sistema.

A fim de que as dificuldades de aplicação destes princípios sejam superadas, olharemos o processo de desenvolvimento de Sistemas de Software como uma Fábrica de Software. Faremos desse modo, porque numa fábrica os processos são bem definidos e divididos em tarefas na forma ótima, além de serem padronizados. No próximo capítulo introduziremos os conceitos da Fábrica do Software.

4 A FÁBRICA DE SOFTWARE

4.1 Resumo

Neste capítulo é tratada a Fábrica de Software. Primeiramente, justifica-se o termo "fábrica" para a área de desenvolvimento de Sistemas.

Uma correlação com a estrutura organizacional de uma fábrica é feita, de tal forma que sejam identificadas nesta estrutura todas as funções de desenvolvimento de Sistemas. Propomos também um organograma para a Fábrica de Software.

4.2 Justificando o Termo Fábrica

Para justificarmos o termo "Fábrica de Software", primeiramente recapitulemos os tipos básicos de produção. Eles são um total de cinco, que se dividem em:

- . **Produção contínua pura** - ela se caracteriza pela produção de um único produto em grandes quantidades. Como só se produz um produto, não há nenhuma variação na sequência das operações de produção. A produção é toda estocada, sendo assim grande a distância entre produtor e consumidor.

- . **Produção contínua com diferenciação** - caracteriza-se pela produção de poucos produtos, que possuem uma pequena diferenciação entre si, com grandes volumes de venda e com uma pequena variação na sequência das operações de produção. Também se produz para estoques, o que ocasiona uma distância grande entre consumidor e produtor.

- . **Produção intermitente repetitiva** - caracteriza-se pela produção de uma número razoável de produtos com um volume médio de vendas. A diferenciação entre os produtos já é maior e a variação entre a sequência das operações ou atividades de produção também é maior do que na produção contínua. Nesse tipo de produção ainda se produz para estoques.

- . **Produção intermitente sob encomenda** - ela se caracteriza por produzir muitos produtos bem diferenciados entre si e com um volume pequeno de vendas. A variação na sequência das operações de produção é alta e a produção se destina ao consumo final, ou seja, a distância entre o produtor e consumidor é pequena.

. **Produção de grandes projetos** - são projetos de longa duração, que se caracterizam por produtos ou atividades bastante diferenciadas. O volume de vendas é muito baixo, sendo muitas vezes unitário. Há uma grande variação na sequência dos processos ou atividades de produção e, normalmente, existem muitos processos num projeto. A produção é feita sob encomenda, o que faz ser muito pequena a distância entre consumidor e produtor.

Enquadramos a área de desenvolvimento de Softwares no tipo de produção de grandes projetos, já que este trabalho é desenvolvido sob encomenda, possui muitas atividades, que demoram um longo tempo (de meio ano a cinco anos). Além disso os produtos diferenciam-se muito entre si e o volume de vendas na maioria dos casos é baixo, praticamente unitário, com exceção dos aplicativos (compilador, planilha, gerador de banco de dados, etc.) que possuem uma grande demanda.

Já que o desenvolvimento de Sistemas se encaixa no tipo de produção de grandes projetos, pode parecer uma incoerência chamarmos essa atividade de fábrica, pois esse termo dá a idéia de uma produção contínua.

Mas o objetivo é justamente este, ou seja, através de novos recursos transformar o desenvolvimento de Sistemas numa produção do tipo contínuo diferenciado. Dessa forma estaremos resolvendo os problemas de filas de esperas, níveis de qualidade, insatisfação do usuário, aumento de produtividade, enfim todos os problemas da chamada Crise do Software.

Essa transformação é possível, pois, se analisarmos as atividades de desenvolvimento de Sistemas, veremos que elas são sempre as mesmas independente do tipo de Sistema, o que varia é o conteúdo. Ora, fazendo uma analogia com uma fábrica, os processos de fabricação não se alteram, a mudança está na matéria prima que varia conforme o produto.

O primeiro passo para estruturarmos a Fábrica de Software será a elaboração de sua estrutura organizacional, garantindo que todas as funções de desenvolvimento de Sistemas, estejam contidas nessa estrutura.

4.3 A Estrutura Organizacional de uma Fábrica

Partimos da estrutura organizacional, pois para atingir os objetivos, executar os planos e possibilitar às pessoas trabalharem eficientemente, as atividades precisam ser agrupadas de forma lógica e delegações de autoridades devem ser feitas de forma que não surjam conflitos.

Como pretendemos estruturar uma fábrica, faremos uma estrutura organizacional semelhante às utilizadas pelas organizações industriais. Vejamos então a relação das funções destas com as funções de desenvolvimento de Sistemas.

Existem diversos tipos de estruturas organizacionais, ou melhor, tipos de departamentalização: funcional, territorial, por produto, por freguês, etc.

Adotaremos a departamentalização funcional, pois é um método lógico, comprovado pelo tempo e pelo seu largo emprego.

Segundo Koontz e O'Donnell [34], qualquer empreendimento compreende a criação de uma utilidade e já que isso ocorre numa economia de troca, as funções fundamentais de qualquer empresa consistem em:

- . produção - criação de utilidade ou acréscimo de utilidade em um produto ou serviço.
- . venda - procura de fregueses, pacientes, clientes, estudantes ou membros que concordem aceitar o produto ou serviço a um determinado preço.
- . financiamento - levantamento e obtenção, proteção e desembolso de fundos da empresa.

Assim, a estrutura organizacional típica de uma organização industrial é departamentalizada da seguinte forma:

DEPTO FUNCIONAL PRINCIPAL	DEPTOS FUNCIONAIS DERIVADOS
Produção	Produção:
	Fabricação
	Montagem
	Manutenção
	Compras
	Controle de Produção:
	Programação
	Controle de Materiais
	Controle de Qualidades
	Custos
Vendas	Atividades de Vendas
	Propaganda
Finanças	Necessidades de Capital
	Recursos Financeiros
	Desembolsos/Crédito

Analisemos cada departamento. Dentro da Produção temos:

. **produção - fabricação** - aqui se produz todos os produtos da fábrica. Na fabricação estão presentes todos os processos de fabricação de todos os produtos, que transformam ou beneficiam os elementos constituintes. O

próprio desenvolvimento de Sistemas com todas as suas etapas são os processos de fabricação de nossa Fábrica de Software, pois transformam os requisitos, as necessidades dos usuários, em Sistemas de Informação.

. **produção - montagem** - os produtos semi-acabados produzidos pela fabricação são aqui montados. Na Fábrica de Software as rotinas de programas e programas, que são armazenados em bibliotecas para serem reutilizados, são aqui unidos ou montados para compor um Sistema, da mesma forma que os semi-acabados compõe um produto. Sistemas diferentes que devem ser enviados para um mesmo usuário, são também "montados" sob um mesmo menu.

. **produção - manutenção** - a manutenção preventiva, corretiva ou preditiva de todos os equipamentos e instalações da fábrica são feitas por esse departamento. São atividades muito importantes numa Indústria, pois se evita que a produção seja interrompida pela quebra dos equipamentos, além de se prolongar a vida útil destes. Numa fábrica de Software existem dois tipos de manutenção: a do Hardware e a do Software. A manutenção do Hardware é a manutenção de um equipamento eletrônico, enquanto na manutenção do Software tem-se a correção de erros do Software, a adaptação do Software a novas necessidades do usuário e o aperfeiçoamento deste a mudanças previstas.

. **compras** - todas as compras de matérias primas e materiais necessários à produção dos produtos e às atividades realizadas na fábrica são feitas por este departamento.

Na Fábrica de Software as compras se restringiriam aos materiais e ferramentas que auxiliam à produção, tais como: as ferramentas de desenvolvimento e depuração (Softwares básicos, aplicativos, linguagens de programação, metodologias, etc.), treinamento para a utilização dos recursos anteriores, materiais como papel, discos magnéticos flexíveis, materiais de secretaria. A matéria prima do desenvolvimento de Software é o conhecimento.

- . controle de produção - programação - toda a programação da produção, ou seja, que produtos serão produzidos, quanto, como e quando são determinados por este departamento. Na Fábrica de Software também precisam ser determinados quais Sistemas devem ser construídos, por quem, quando e de que modo (quais recursos serão utilizados).
- . controle da produção - controle de materiais - os níveis de estoques de matérias primas, ou de produtos semi-acabados ou de produtos acabados são controlados por este departamento. Os estoques na Fábrica de Software equivalem aos arquivos de dados, às bibliotecas com programas prontos, que serão reaproveitados ou reutilizados em outros Sistemas, e às memórias dos computadores, onde ficam armazenados os programas dos Sistemas e os arquivos. Os arquivos devem ser sempre administrados para que dados, que não são mais utilizados, sejam eliminados ou armazenados em outros arquivos; para que quando seja necessário armazenar novos dados, arquivos

sejam criados. Para cada arquivo deve ser feita uma previsão de volume de informações a ser ocupado. A memória e demais recursos das máquinas devem ser administrados, para não escassearem com o aumento dos volumes de processamento. Um Sistema de codificação dos programas deve ser elaborado para facilitar a consulta. Uma biblioteca contendo os manuais de aplicativos, de utilização e especificação dos Sistemas fabricados, jornais e revistas técnicas também é controlado por este departamento.

- . **controle da produção - controle de qualidade** - para assegurar os níveis de qualidade de uma fábrica, métodos devem ser aplicados para se controlar o nível de qualidade dos processos. Inspeções por amostragem também devem ser feitas para aprovar ou rejeitar um lote de produção. Na fábrica do Software esses métodos de controle de qualidade do processo também devem existir, além dos testes de qualidade do Sistema, a fim de se verificar se o Sistema está atendendo todas as suas especificações.
- . **custos** - todos os Sistemas de custeio da fábrica são elaborados e controlados por este departamento. Também na Fábrica de Software deve existir esse departamento com as mesmas funções. Tendo em mãos esses Sistemas, será possível avaliar de acordo com os custos, que produtos são prioritários para fabricação, além de ser possível elaborar uma análise de custo-benefício, pois muitas vezes vale mais a pena construir um novo Sistema do que

continuar fazendo manutenção num já existente, ou às vezes é melhor comprá-lo e fazer as adaptações necessárias do que desenvolvê-lo.

Dentro de Vendas temos:

. vendas - atividades de vendas - neste departamento são organizados as vendas dos produtos da fábrica, planos de acompanhamento dos clientes e pesquisas de mercado são elaboradas, além de toda a parte burocrática da operação. Na fábrica de Software, se esta for aberta ao mercado, esse departamento possuirá as mesmas funções acima apresentadas. Mas, se ela pertencer a outra empresa e apenas produzir para essa, suas atividades se voltarão para o usuário, tendo os mesmos objetivos: prospecção de necessidades dos usuários, análise de produtos concorrentes similares existentes no mercado, venda de seus produtos, ou melhor, esse departamento deverá tomar atitudes, que facilitem a aceitação do Sistema produzido pelo usuário, ou que mostrem ao usuário as vantagens da utilização do Sistema.

. vendas - propaganda - aqui é feita a propaganda dos produtos da empresa. Para facilitar a venda dos Sistemas ao usuário, deve esse departamento da Fábrica de Software fazer propaganda na forma que achar mais adequado.

Dentro de finanças existem:

. finanças - necessidades de capital - previsões da quantidade de capital necessário para a execução da produção, ou para investimentos em novos equipamentos, em

ampliações, etc. , são elaboradas. Na Fábrica de Software temos a mesma função.

. **finanças - recursos financeiros** - toda a administração dos recursos financeiros é feita por este departamento.

. **finanças - desembolsos/crédito** - todo o controle dos desembolsos e créditos efetuados nas contas das empresas deve ser controlado por este departamento.

Estes dois últimos departamentos possuem as mesmas funções na Fábrica de Software, desde que esta não faça parte de outra empresa. Se ela for dependente dessa empresa, na realidade estes departamentos já farão parte da última.

4.4 A Estrutura Organizacional da Fábrica de Software

A estrutura organizacional da Fábrica de Software será baseada no organograma acima mostrado, já que nossa intenção é formar uma organização industrial. Apenas pequenas alterações serão efetuadas, para melhor atender às necessidades do desenvolvimento de Sistemas.

A organização possuirá a seguinte estrutura:

DEPTO PRINCIPAL	DEPARTAMENTOS FUNCIONAIS DERIVADOS
Produção	Produção:
	Fabricação:
	Definição do Produto
	Fabricação do Produto
	Montagem
	Manutenção:
	Manutenção dos Equipamentos
	Manutenção do Produto
	Projeto de Produtos
Produção	Compras
	Controle de Produção:
	Programação
	Controle de Materiais
	Controle de Qualidade
	Custos
Vendas	Atividades de Vendas
	Propaganda
Finanças	Necessidades de Capital
	Recursos Financeiros
	Desembolsos/Crédito

Explicaremos apenas os departamentos, cujas funções diferenciam das mencionadas anteriormente.

- . **fabricação - definição do produto** - de acordo com as necessidades do usuário este departamento estabelece os Sistemas e os serviços prestados por estes, suas funções básicas e seus requisitos. Este departamento trabalhará juntamente com o usuário para que os produtos - os Sistemas - sejam corretamente definidos.
- . **fabricação - fabricação do produto** - a fabricação propriamente dita do Sistema é aqui elaborada, ou seja, as fases do ciclo de vida de projeto preliminar, projeto detalhado e codificação do Sistema são aqui realizadas. Também são executados os testes unitários do Sistema, os quais verificam a performance funcional de cada componente do Sistema. Percebemos o papel importante da documentação, toda a definição do Sistema deve ter sido documentada pelo departamento de definição do produto, a fim de que esse departamento dê continuidade ao trabalho. Da mesma forma este departamento deve documentar o projeto do Sistema, a fim de que um outro departamento faça os testes deste.
- . **manutenção - manutenção dos equipamentos** - esta manutenção pode ser preventiva, preditiva ou corretiva. Ela é realizada sobre os equipamentos para que reparos sejam feitos, se possível, sem interromper a produção.
- . **manutenção - manutenção dos Sistemas** - após o Sistema entrar em operação na maioria das vezes modificações

precisam ser feitas, estas podem ser para corrigir erros que não foram detectados ou para atender novas necessidades do usuário ou para ampliar certos serviços pela evolução natural das coisas. Esta fase exige não só a mudança do código do Sistema, mas também mudanças nas especificações, no projeto do Sistema e em toda documentação.

. **projeto de produtos** - em toda a empresa existe um departamento que projeta novos produtos, visando atender as necessidades do mercado. Na Fábrica de Software este departamento terá a função de especificar novos produtos, que ainda não tenham sido propostos pelo usuário. Cabe ressaltar, que este departamento deve estar à frente das necessidades do usuário, procurando projetar e especificar produtos que o usuário venha necessitar.

O organograma que propusemos para a fábrica de Software é do tipo funcional, mas a maior parte das empresas da área de informática possui um organograma estruturado por produto. Diríamos, que conforme o estágio de informatização em que a empresa se encontra, ela possui um tipo de estrutura organizacional.

Nolan [64], em meados da década de 70, delineou estágios de evolução da informática nas organizações, a partir do princípio que toda a evolução ocorre de forma gradativa.

Ele definiu seis estágios definidos da seguinte maneira:

1. **início** - onde é adquirido o computador. São desenvolvidas aplicações para reduzir os custos na área

de contabilidade.

2. **contágio** - proliferação de aplicações em todas as áreas funcionais da organização.
3. **controle** - surge a necessidade de controlar o uso dos computadores, com isso a gerência e os usuários passam a utilizar, o computador como instrumento para planejar e controlar.
4. **integração** - inicia-se a montagem de banco de dados como seus respectivos Sistemas para manipulação dos dados nas várias aplicações e consultas.
5. **administração de dados** - solidifica-se o conceito da administração de dados com Sistemas de dados partilhados e comuns, onde se consegue uma integração de várias aplicações das áreas usuárias.
6. **maturidade** - o dado como patrimônio da empresa é gerado e utilizado de uma forma distribuída. Além da administração dos dados há a administração dos recursos relacionados.

Quanto mais avançado for o estágio em que a empresa se encontra, mais características funcionais terá a estrutura organizacional de sua área de informática. Na realidade, essas características ainda são poucas, mesmo nas empresas que estão no estágio de maturidade.

A empresa em estudo está situada entre o quinto e sexto estágio de Nolan, apesar disto apenas alguns departamentos de sua organização foram estruturados pelo critério funcional, sendo que

as funções que eles exercem possuem menor importância se comparadas a outras. No restante da área de informática da empresa foi utilizada a divisão por produto.

Por exemplo, existem departamentos de controle de materiais, de manutenção dos equipamentos e de compras. No entanto, não existe um departamento separado que efetue o controle de qualidade dos processos ou dos produtos, departamento este considerado imprescindível nas indústrias atuais. Não há um departamento de pesquisa de novos produtos, o qual é necessário para que a empresa esteja na frente do mercado. Sistemas de custeio não são elaborados e muito menos existe um departamento que cuide desta função.

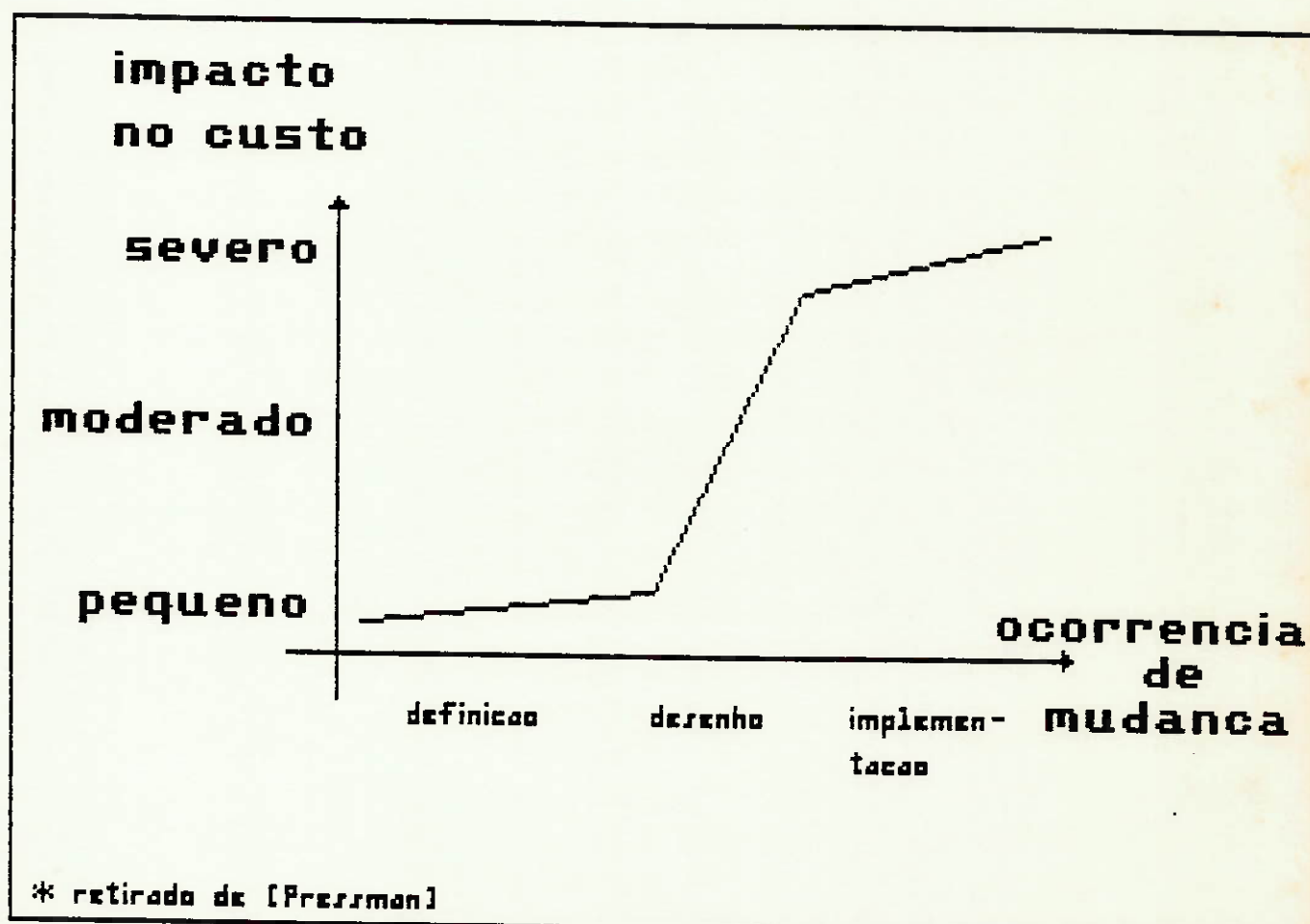
Como estas funções muitas outras estão sendo postas em segundo plano, e tal fato não poderia acontecer. Uma forma disto ser evitado é estruturar a organização da forma como propusemos.

Dentre todas as áreas da Fábrica de Software, para que os problemas da Crise do Software sejam realmente solucionados, devemos concentrar nossas atenções na área de fabricação: definição do produto e fabricação do produto. Pois só assim estaremos inovando a produção de Sistemas, onde se encontra a origem de todos os problemas anteriormente identificados.

Como cita Sommerville [60] a chave de um bom Sistema, fácil de entender, executar, implantar e manter é o correto projeto deste. Só que para se obter um projeto bem elaborado é necessário que a definição do Sistema seja a melhor possível.

Quanto melhor executadas forem essas duas etapas (definição e projeto do Sistema), menos problemas e modificações serão feitas na fase de manutenção. O efeito disto no custo do Sistema é relevante. A fig. 4.4.1 mostra o impacto no custo do Sistema de mudanças realizadas, conforme a fase do desenvolvimento em que elas se encontram.

FIG. 4.4.1 - Impacto no Custo do Sistema de Alterações Realizadas.



5 A FABRICAÇÃO DE SOFTWARE

5.1 Resumo

Neste capítulo é enfocada a fase de fabricação do produto da Fábrica de Software. Dentre os tipos de recursos existentes nesta fase, concentramo-nos nos ambientes centrados em metodologias.

Descrições sucintas de diversas metodologias são apresentadas; na sequência foi elaborada uma descrição mais detalhada das metodologias Remora e Method/1. O recurso de prototipação também está aqui presente, além da constatação de alguns problemas cuja solução é discutida no próximo capítulo.

5.2 Recursos para o Desenvolvimento de Software

Ao analisarmos a fabricação do Software, a fim de resolvermos os problemas da chamada Crise do Software, levaremos em consideração os recursos existentes que auxiliam este trabalho.

Gostaríamos de salientar que estes recursos procuram solucionar muitos dos problemas existentes na construção de sistemas grandes e complexos, como é o caso do Sistema de Câmbio da empresa onde realizamos estágio, o qual possui 150 arquivos com um total de 548 campos distintos, 287 operações diferentes e 107 produtos.

Carlos Lucena [35] divide esses recursos em três grupos, de acordo com as funções que eles proporcionam à equipe de desenvolvimento de Sistemas.

No desenvolvimento de Sistemas existem alguns espaços de trabalho ou ambientes, os quais utilizam linguagens para a descrição do Sistema, métodos, para a preparação e análise das descrições e ferramentas que são Sistemas de Software para manutenção, teste, depuração de textos de programas, etc.

Assim, de acordo com as funções acima, divide-se os recursos, ou ambientes em três grupos:

- . ambientes centrados em Sistemas operacionais,
- . ambientes centrados em linguagem,
- . ambientes centrados em metodologias.

No primeiro grupo estão presentes as ferramentas para a organização e manutenção de módulos, recursos para o ajuste do ambiente a seus usuários (definição e redefinição de comandos e sequências de comandos, interpretações padrão de operações, etc.), mecanismos de ajuda ("help"), suporte para equipes de programadores (correio eletrônico, planejamento de projetos), etc.

O segundo grupo é constituído pelas linguagens de programação, processadores de linguagens, editores de texto, recursos para transformação de texto (compiladores, interpretadores), recursos para a organização de textos, recursos para processamento simbólico, Sistemas de arquivo (estrutura hierárquica de nomes, recursos para categorização geral), recursos de proteção (auto-proteção, controle de acesso), etc.

Já do terceiro grupo fazem parte as metodologias de desenvolvimento de Softwares, os repositórios de informação sobre o projeto de Software, os mecanismos de catalogação e recuperação de informações, as técnicas de documentação em geral, os chamados ambientes integrados, que pretendem dar suporte automatizado para todo o processo de desenvolvimento de Sistemas, etc.

Fizemos um levantamento na empresa em estudo, para verificar em que ambientes é necessário investir, ou melhor, alterar ou incluir novos recursos para aprimorarmos a fase de fabricação do produto, a fim de solucionarmos os problemas identificados nos capítulos anteriores.

O levantamento mostra que a empresa possui os seguintes recursos:

! AMBIENTES !	RECURSOS EXISTENTES	! TIPO !	INÍCIO DE	!
! CENTRADOS !	NA EMPRESA	!	UTILIZAÇÃO	!
!-----!-----!-----!-----!				
!	! . a nível de main frame	!	!	!
! Em	! . MVS/XA	! S.OPER. !	1986	!
! Sistemas	! . MVS/ESA	! S.OPER. !	1989	!
! Operacio-	! . VM/MS	! S.OPER. !	1979/80	!
! nais	! . CICS	! M.TELE. !	1981	!
!	! . VTAM	! G.TELE. !	1981	!
!	! . a nível de micro-Informática!	!	!	!
!	! . MS-DOS + WINDOWS	! S.OPER. !	1986/87	!
!	! . OS/2 +	!	!	!
!	! + PRESENTATION MANAGER	! S.OPER. !	1989	!
!	! . REDE LOCAL CETTUS/NOVELL!	!	1988	!
!-----!-----!-----!-----!				

S.OPER. = Sistema Operacional.

M.TELE. = Monitor de Teleprocessamento.

G.TELE. = Gerador de Teleprocessamento.

AMBIENTES	RECURSOS EXISTENTES	TIPO	INÍCIO DE
CENTRADOS	NA EMPRESA		UTILIZAÇÃO
	. a nível de main frame		
	. META COBOL	L39G	1976
Em	. COMPILADOR COBOL	COMPIL.	1974
Linguagens	. APL	L39G	1983
	. AS	L49G	1988
	. NATURAL/ADABAS	L49G	1986
	. NATURAL2	L49G	1989
	. CSP/DB2	L49G	1988
	. a nível de micro-informática		
	. DBASE III	L49G	1986/87
	. PASCAL	L39G	1986/87
Em Meto-	Macrometodologia MIND	METOD.	1986
dologias	MIND - AID	FERR.	1988

. L39G = Linguagem de Terceira Geração.

. L49G = Linguagem de Quarta Geração.

. COMPIL. = Compilador.

. METOD. = Metodologia.

. FERR. = Ferramenta de apoio à Metodologia.

Como se pode perceber, a empresa vem investindo bastante nos ambientes centrados em Sistemas operacionais e em linguagens, embora ainda há muito no que evoluir, ela dispõe de alguns dos recursos mais avançados disponíveis no mercado. Nos ambientes centrados em metodologia, comparativamente aos demais, pouco investimento ou estudo foi realizado. Sallentamos o fato, de a empresa não estar satisfeita com o que possui na área de metodologia. Na realidade não existe nenhuma metodologia formal, apenas tentativas de aplicação de algumas ferramentas.

Desta forma, dentre os recursos existentes para auxiliar a fabricação de Sistemas, concentraremos nossa atenção nos ambientes centrados em metodologias.

Assim, com um investimento correto nos três tipos de recursos existentes para o desenvolvimento de Sistemas, os problemas da Crise do Software poderão ser solucionados, pois poderão ser obtidos aumento de produtividade e do nível de qualidade.

Nos propomos em nosso estudo, a descrever os diversos recursos existentes na área de metodologias. Mas não faremos uma análise comparativa extensiva destes, pois além de requerer uma vasta experiência de utilização dos recursos, isto por si só resultaria em uma monografia, o que não é o caso.

No entanto, escolhemos alguns recursos de uso mais corrente, para detalhá-los e explicá-los de forma mais completa. Expondo ao leitor, não familiarizado com a área, os recursos que tem sido desenvolvidos nos últimos anos, para ambientes centrados em metodologias; além de dar uma base para discussão e

aprofundamento posterior, de quais dentre estes recursos são mais apropriados à empresa. Muitos dos conceitos aqui introduzidos serão utilizados nos próximos capítulos.

5.3 Ambientes Centrados em Metodologia

5.3.1 Uma Breve Descrição das Metodologias

5.3.1.1 Conceituando Metodologia

Embora muitas vezes não saibamos exatamente o que é uma metodologia, sentimos que ela nos fornece um procedimento para executarmos as ações, de tal forma que níveis de qualidade do trabalho sejam atingidos.

Acreditamos que uma boa definição de metodologia é a proposta por Benci [5], na qual uma metodologia é caracterizada por quatro componentes distintos e integrados:

- . **modelo** - fornece os conceitos com os quais é possível compreendermos um dado tipo de problema.
- . **linguagem** - fornece a sintaxe e a semântica que permitem expressar o problema em termos de suas características e restrições, com base nos conceitos do modelo.
- . **ferramentas** - são os recursos técnicos que dispomos para a modelagem do problema e sua descrição através da linguagem.
- . **método** - consiste nas etapas que devemos seguir e nas

ações a serem executadas em cada uma delas, de modo que possamos resolver o problema usando o modelo, linguagem e ferramentas que dispomos.

Tão mais sólida será uma metodologia, quanto mais completos forem o modelo e a linguagem associados. E tão mais praticável será, quanto mais ricos forem o método e as ferramentas disponíveis.

Dentro do processo de fabricação do Software, mais especificamente, do processo de projeto do Software existem duas vertentes em que se baseiam as metodologias, as quais são:

- . metodologias baseadas em processos.
- . metodologias baseadas em dados.

5.3.1.2 Metodologias Baseadas em Processos

Nas metodologias baseadas em processo, o Sistema é construído através da análise e decomposição de seus processos ou de suas funções. Assim, parte-se da descrição dos processos existentes no problema em estudo. Cada vez que o usuário necessita de uma nova função, esta deve ser acrescida ao Sistema.

Existem diversas técnicas, em que se baseiam as metodologias, para a elaboração de Sistema na linha de processos, dentre elas:

- a. análise estruturada (SA) - foi proposta por Yordon e Constantine [72], com o objetivo de produzir uma especificação do Sistema, que define a estrutura do

problema com a visão do usuário. A especificação do Sistema é composta de diagrams de fluxo de dados (DFD), dicionário de dados e especificação de processos.

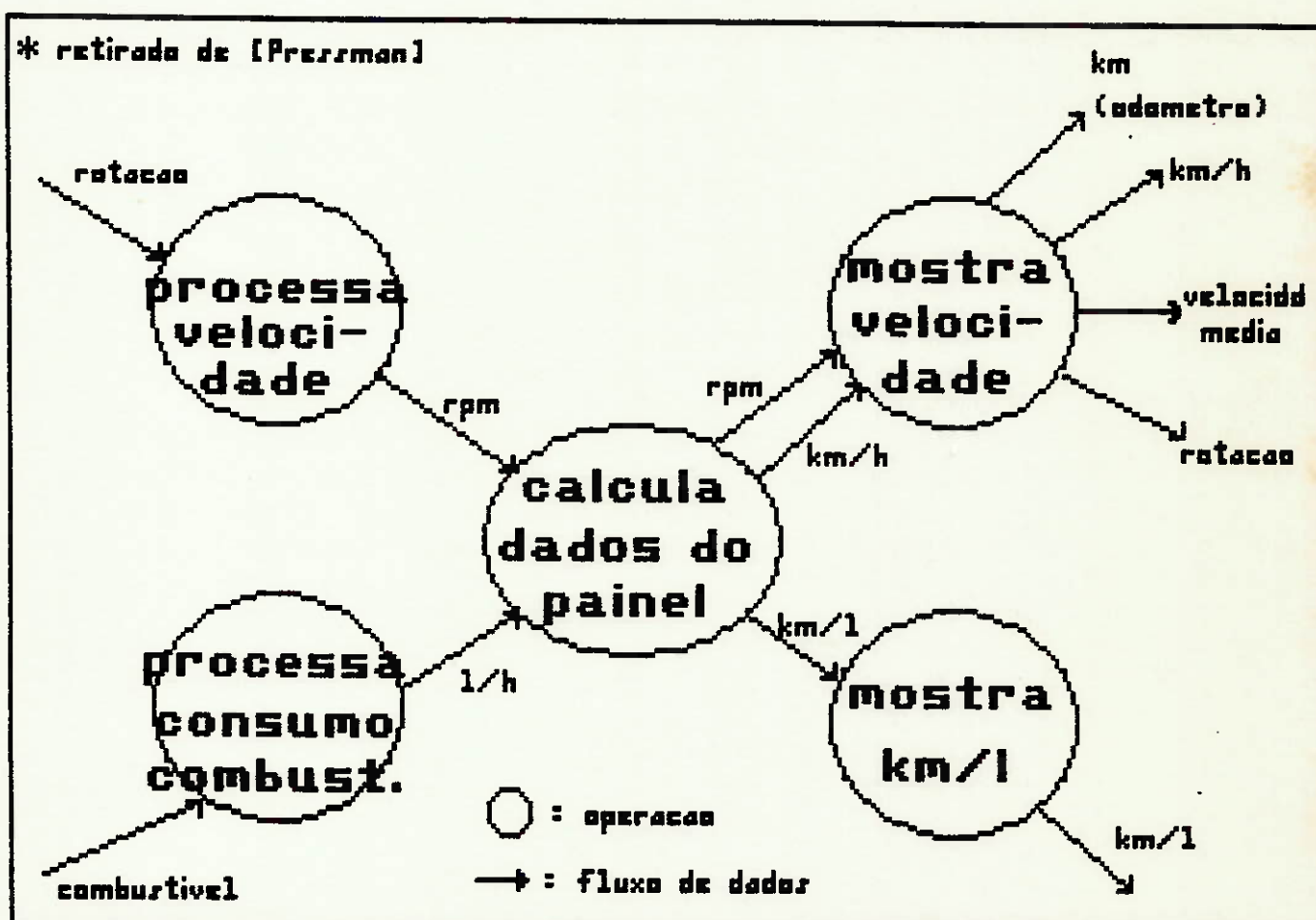
São definidos alguns conceitos como as entidades e os depósitos de dados. As entidades são categorias lógicas de coisas ou pessoas, que representam uma fonte ou destino dos dados e dos processos, enquanto que os depósitos de dados são o local, onde se armazenam os dados necessários à execução dos processos.

Os passos para a execução da SA consistem em:

1. identificar os processos, os fluxos de dados, os repositórios e as entidades, construindo o DFD.
2. refinar o diagrama construído no item 1 até o nível de detalhe desejado.
3. construir o dicionário de dados, o qual é um conjunto de definições dos dados presentes no DFD nos depósitos de dados ou sob a forma de fluxo de dados.
4. especificar os processos, descrevendo o que ocorre nos processos do DFD, ou seja, a transformação que ocorre dos dados de entrada em dados de saída. Esta descrição é feita em português estruturado.

Há diversos tipos diferentes de DFD, como é o caso do diagrama de Gane e Sarsson [22] (largamente utilizado no Brasil), ou o diagrama de DeMarco [17]. Mostraremos a seguir um exemplo de DFD com a notação de DeMarco elaborado para o Sistema: painel de um automóvel (fig.5.3.1.2.1).

FIG. 5.3.1.2.1 - Exemplo de um Diagrama de Fluxo de Dados.

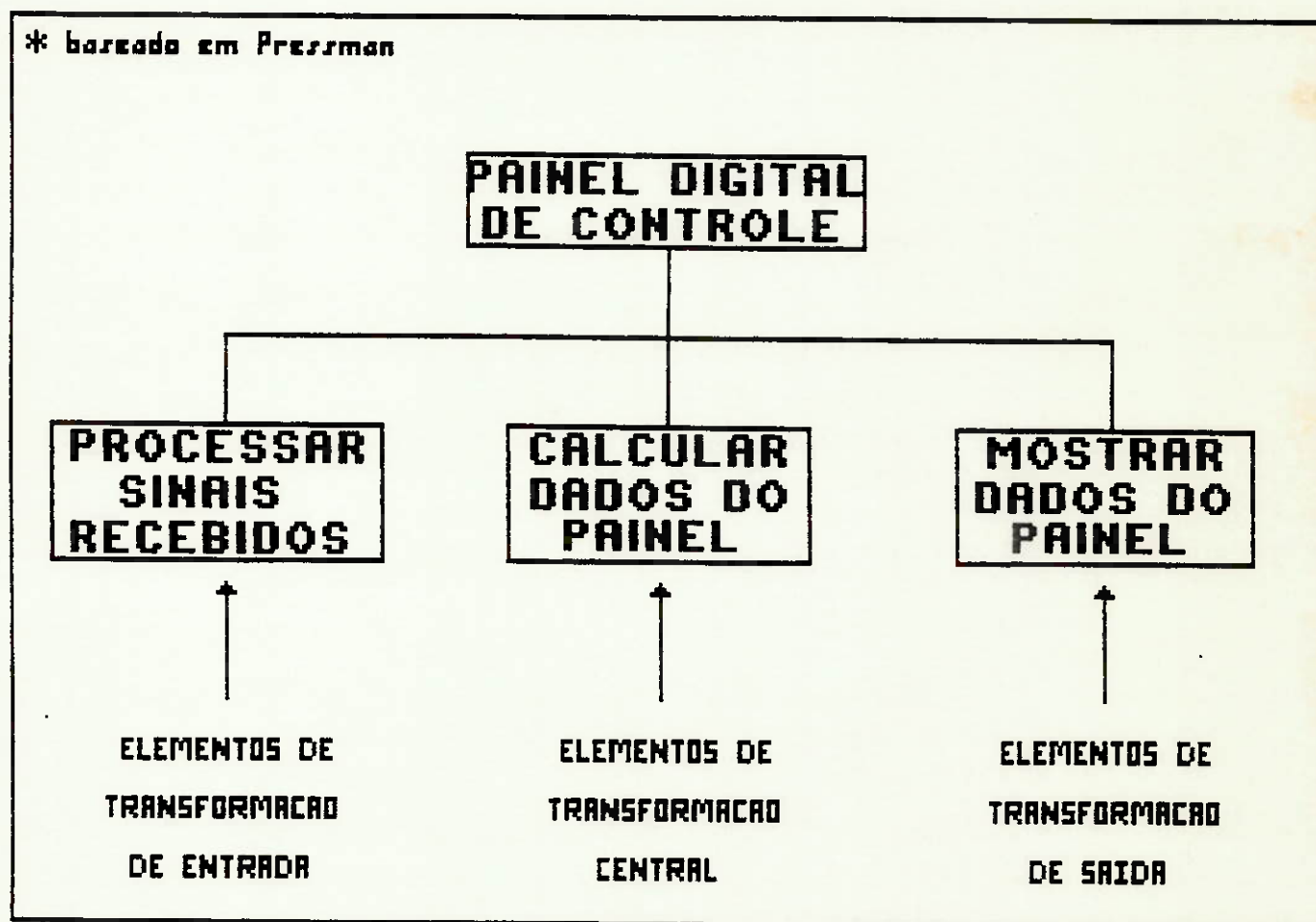


b. **projeto estruturado** - é baseado nos conceitos introduzidos por Yordon e Constantine [72]. Cronologicamente, o projeto estruturado surgiu antes da análise estruturada. Na realidade os dois se complementam. O projeto estruturado é um guia para a concepção do projeto do Sistema, através do mapeamento do fluxo de dados, processos do problema e refinamentos sucessivos destes. O procedimento de construção daquele consiste resumidamente em:

1. a partir do DFD elaborado na análise estruturada, identificar os elementos de transformação de entrada, centrais e de saída.
2. identificar os fatores de divisão dos elementos de transformação de entrada, centrais e de saída, a fim de formar uma estrutura hierárquica dos programas (diagrama de estrutura).
3. refinar e otimizar a estrutura do programa elaborada no item 2.

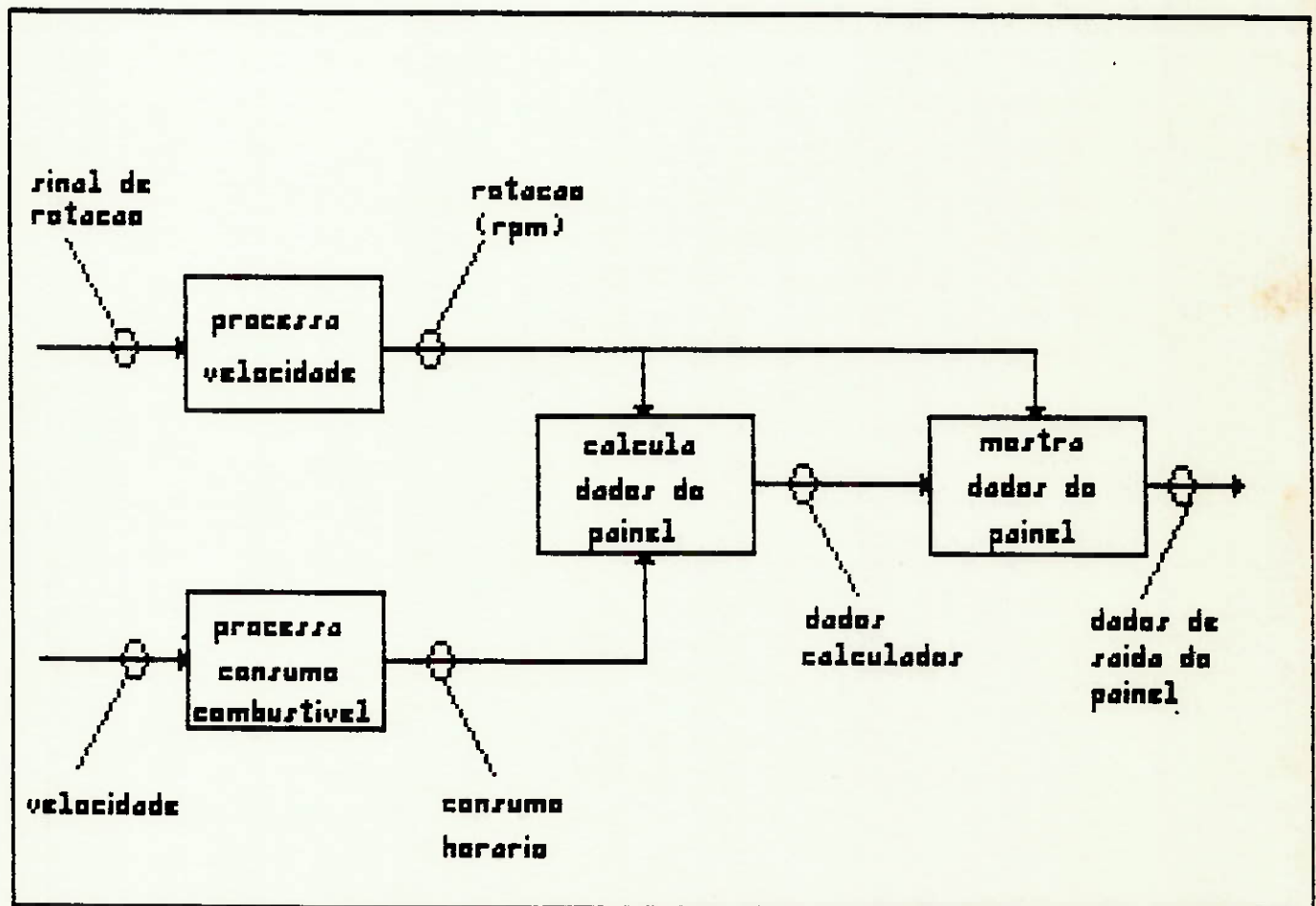
O projeto estruturado auxilia na rápida definição e no refinamento do fluxo de dados. Ele é permite que o desenhista do Sistema expresse sua percepção do problema em termos de fluxo de dados e diagramas de transformação e, também, ele provê meios de se evoluir o projeto. Esse método é mais conveniente para projetos de Sistemas que não possuam uma estrutura de dados previamente definida. A fig. 5.3.2.1.2 mostra o diagrama de estrutura elaborado para o Sistema: painel de um automóvel.

FIG. 5.3.1.2.2 - Exemplo de um Diagrama de Estrutura.



c. técnicas de projeto de análise estruturada (SADT) - são baseadas na análise estruturada (SA) proposta por Ross [53]. SA é uma linguagem gráfica utilizada para expressar de forma explícita as relações hierárquicas e funcionais entre os objetos e atividades do problema. A estrutura do Sistema representada graficamente, ressalta as interfaces entre os componentes do Sistema. A análise da estrutura do Sistema pode ser conduzida de forma "top-down", estruturada, modular e hierárquica. Além disso, a SADT inclui planejamento, controle da configuração dos procedimentos e condições de organizar o trabalho da equipe de desenvolvimento. SADT tem sido utilizada numa gama extensa de aplicações com bastante sucesso. Ela é particularmente eficiente nos primeiros e últimos estágios do ciclo de vida de desenvolvimento de Sistema, e menos eficiente no projeto detalhado do Sistema. No entanto, como ela permite que cada desenhista desenvolva de forma independente os diagramas, haverá dificuldades na integração dos diversos diagramas. Para ilustrar o diagrama utilizado em SADT, mostramos um exemplo na fig. 5.3.1.2.3.

FIG. 5.3.1.2.3 - Exemplo de um diagrama utilizado pela SADT.



d. metodologia para projeto de Sistemas de Jackson - a metodologia proposta por Jackson [28] dá ênfase a estrutura do problema, ou melhor, a estrutura dos dados é utilizada como a chave de um bom projeto do Sistema. A estrutura básica de um Sistema é determinada pela estrutura dos dados que ele processa e, o Sistema é encarado como um mecanismo que transforma dados de entrada em dados de saída. Utilizando estruturas de entrada e saída como um guia, será possível desenhar um Sistema bem estruturado.

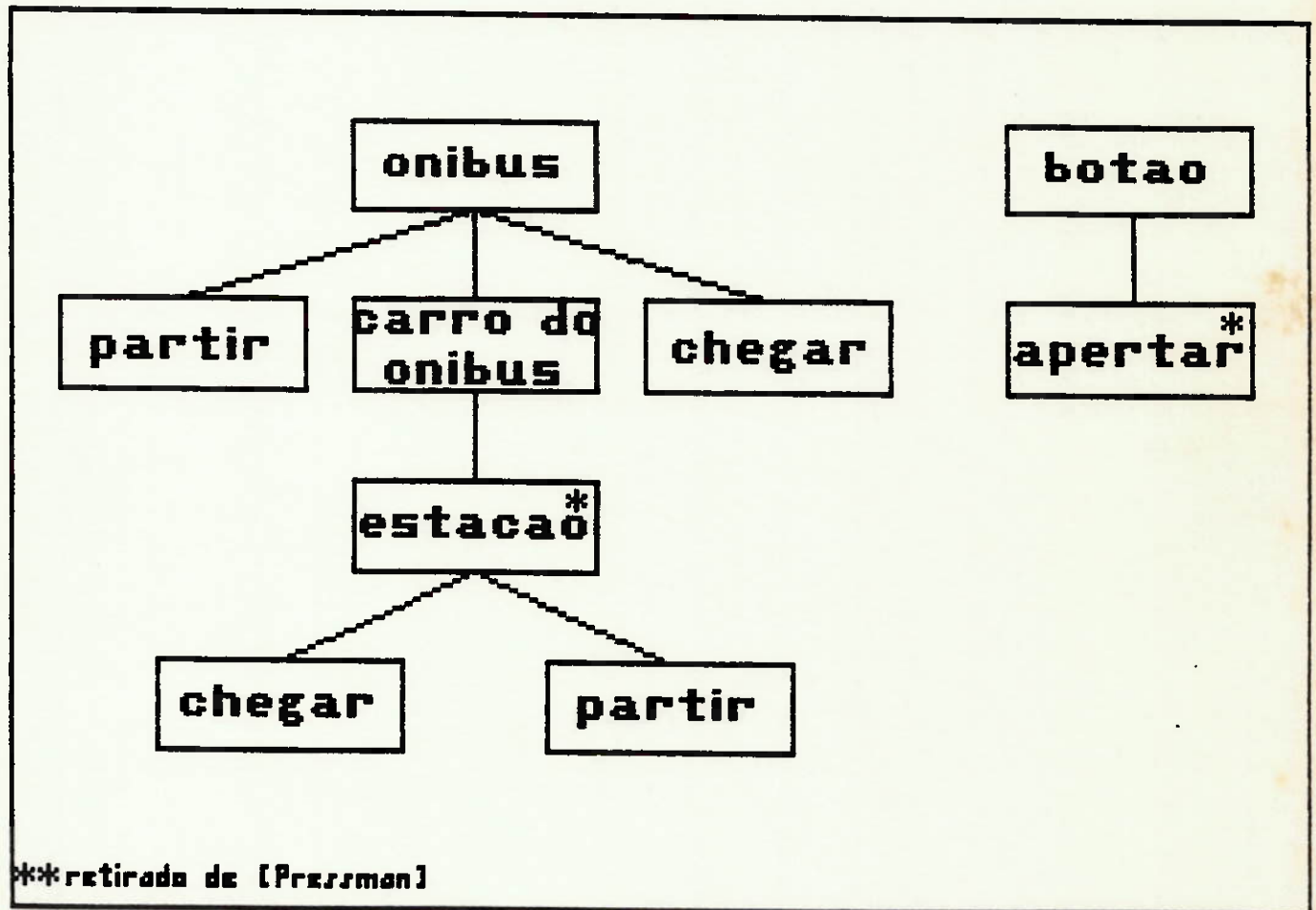
De forma resumida a metodologia consiste em:

1. identificar e desenhar a estrutura dos dados de entrada e de saída.
2. desenhar a estrutura do programa baseado nos diagramas de estrutura de dados.
3. detalhar e alocar as operações identificadas nos programas.
4. converter o texto de estrutura do programa em programa.

A fig.5.3.1.2.4 um exemplo do diagrama utilizado por Jackson para o Sistema: um ônibus que se movimenta conforme o botão de chamada é acionado.

As vantagens da metodologia de Jackson são o fato de cada passo no processo de projeto do Sistema poder ser verificado e, de diferentes desenhistas trabalhando independentemente num mesmo problema obterão o mesmo resultado. No entanto, essa metodologia não diz como se deve estruturar os dados.

FIG. 5.3.1.2.4 - Exemplo de um diagrama de Jackson.



e. metodologia de projeto de Sistema de Warnier - a metodologia proposta por Warnier [74] é similar a proposta por Jackson, pois ela também assume que a estrutura de dados é a força de um bom Sistema. Além disso as duas metodologias utilizam a mesma técnica de representação, apenas o tratamento das informações é diferente. A metodologia de Warnier fornece procedimentos mais detalhados para o projeto do Software do que a de Jackson. Ela utiliza quatro tipos de representação do projeto: diagrama de organização de dados, diagrama de sequência lógica, lista de instruções e pseudo-código. O diagrama de organização dos dados descreve os dados de entrada e saída, o diagrama de sequência lógica representa o fluxo lógico do Sistema, a lista de instruções contém as instruções utilizadas no projeto e o pseudo-código é a descrição do projeto. Resumidamente a metodologia é formada dos seguintes passos:

1. identificar os dados de entrada e de saída do Sistema.
2. organizar os dados de entrada na forma hierárquica.
3. definir de forma detalhada o formato de cada item dos arquivos e determinar o número de suas ocorrências.
4. fazer os itens 1 e 2 para os dados de saída.
5. especificar os detalhes dos programas através da identificação dos tipos de instruções contidas no projeto na seguinte ordem: instruções de leitura, instruções de desvio, cálculos, saídas e chamadas de subrotinas.

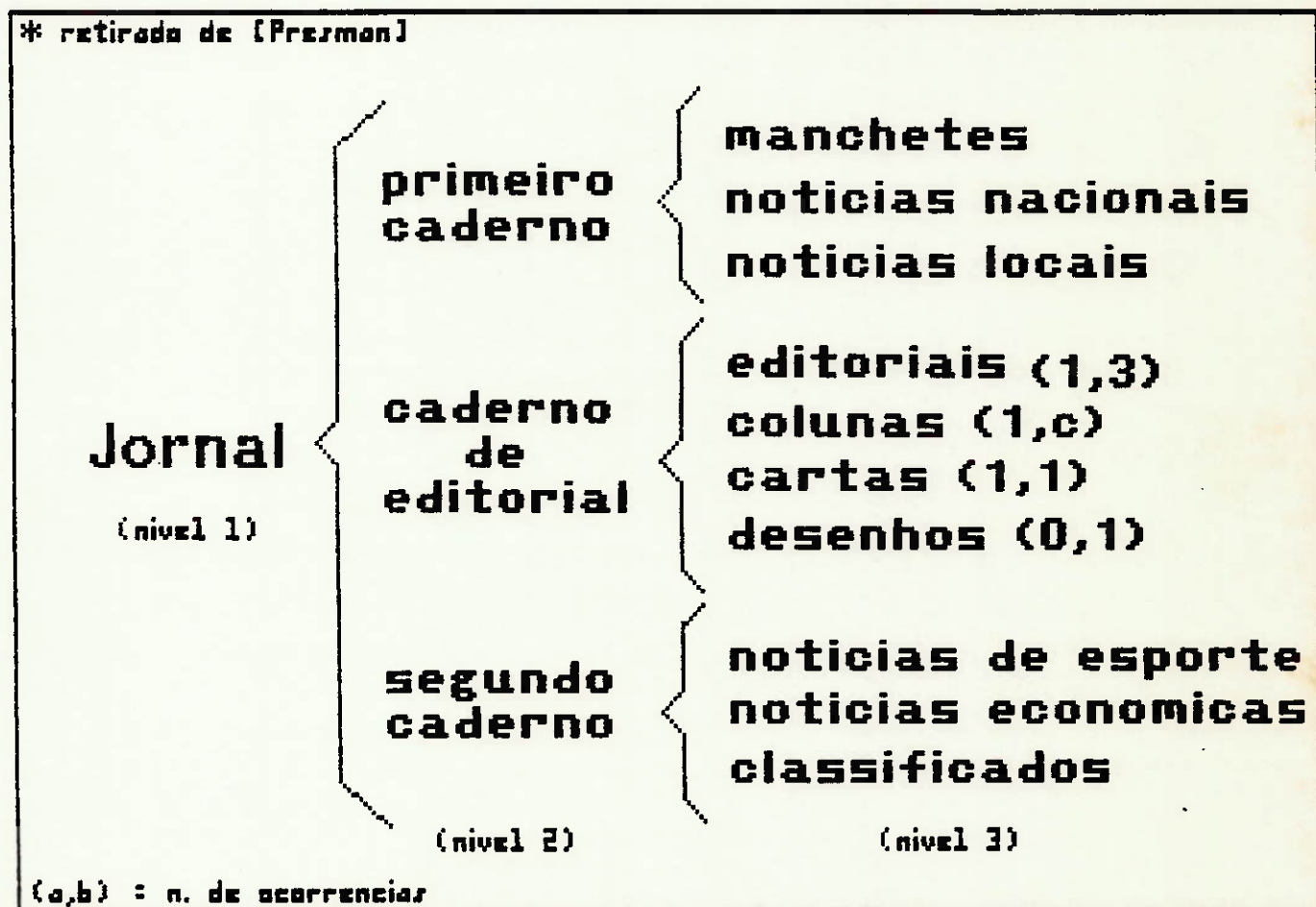
6. utilizando fluxograma, mostrar a sequência lógica das instruções utilizando símbolos especiais para representar o início de processo, fim de processo, subrotinas e a rotina principal.

7. numerar os elementos da sequência lógica e expandi-los através das instruções contidas no item 5.

A fig. 5.3.1.2.5 mostra um exemplo do diagrama de organização dos dados de Warnier para o caso de diagramação de um jornal, os números entre parênteses indicam o número de ocorrência.

Apesar das metodologias de Jackson e Warnier estruturarem os dados do Sistema, elas são baseadas em processos, pois, para se identificar os dados de entrada e de saída do Sistema, é necessário primeiramente identificar suas funções ou processos. Assim, está se partindo dos processos do Sistema para se elaborar seu projeto.

FIG. 5.3.1.2.5 - Exemplo de um Diagrama de Warnier.



f. hierarquia, entrada, processo e saída (HIPO) -desenvolvida pela IBM [63], esta técnica consiste num conjunto de diagramas e é utilizada como uma ferramenta de documentação. Há três tipos de diagramas para representar entrada, processo e saída. O processo é representado numa caixa central de processamento que está conectada com as caixas de entrada e de saída. O procedimento básico de projeto de Sistema utilizado com os diagramas HIPO consiste em:

1. Iniciar do ponto mais alto de abstração.
2. Identificar as entradas, saídas e os processos do Sistema.
3. conectar cada entrada e saída a seu respectivo processo.
4. documentar cada elemento do Sistema utilizando os diagramas HIPO.
5. refinar o diagrama do Sistema repetindo os itens 1-4.

As vantagens do HIPO são a facilidade de aprender e usar, a habilidade de representar as relações entre os dados de entrada, de saída e os processos do Sistema, a facilidade de decompor um Sistema na forma hierárquica sem envolver detalhes da sua lógica.

5.3.1.3 Metodologias Baseadas em Dados

Este grupo - metodologias baseadas em dados - dá ênfase ao desenho dos dados componentes do Sistema e as técnicas que derivam o projeto dos dados. Ou seja, primeiramente não se analisam as funções, mas sim os dados que constituem o Sistema, dessa forma se obtém uma visão abstrata e integrada dos dados do Sistema. Parte-se da descrição dos dados para se desenhar o Sistema. A vantagem dessa abordagem é que a estrutura de dados independe dos processos ou funções do Sistema, assim, ao se alterar algum processo, a base de dados permanece a mesma, sendo às vezes necessário apenas acrescentar ou eliminar alguns tipos de dados.

Primeiramente introduziremos alguns conceitos definidos por Kacuta [30], e que serão utilizados:

- . modelo - uma representação formal da realidade.
- . dados - coleção de símbolos organizados intencionalmente, para representar uma parte da realidade que estivermos tratando.
- . modelagem de dados - o modelo de dados não é um fluxograma de como o Sistema trabalha, nem o caminho através do qual os dados são processados desde a entrada até a saída. Ele deve refletir a estrutura do Sistema descrito. A modelagem de dados consiste em descobrir, identificar e definir: objetos, relacionamentos, ações que modificam objetos e relacionamentos, e os elementos de dados que descrevem os objetos e seus relacionamentos.

- . **entidade** - é algo concreto ou abstrato que possui características que o tornam distinguível, alguma coisa que desempenha um papel específico no Sistema que está sendo modelado. Pode ser algo tangível como um funcionário, um cliente, uma peça, etc.; pode ser intangível como um título de uma função, uma compra, uma estimativa, etc.
- . **relacionamento** - é uma associação com um significado entre duas ou mais entidades. Ex: casamento é um relacionamento entre duas entidades do tipo pessoa.

Existem três níveis de modelagem de dados, a saber:

- . **modelo conceitual de dados** - procura espelhar a realidade, independente das restrições de implementação, segundo uma visão global do Sistema.
- . **modelo lógico de dados** - representa a realidade, mas voltada para as características do Sistema gerenciador de banco de dados (hierárquico, rede, relacional, etc.).
- . **modelo físico de dados** - representa a organização física dos dados no equipamento de armazenamento. Contém detalhes de integridade, recuperação, e caminhos eficientes de obtenção e alteração dos dados.

Na **modelagem conceitual** de dados, segundo Sommerville [60], existem alguns modelos muito utilizados:

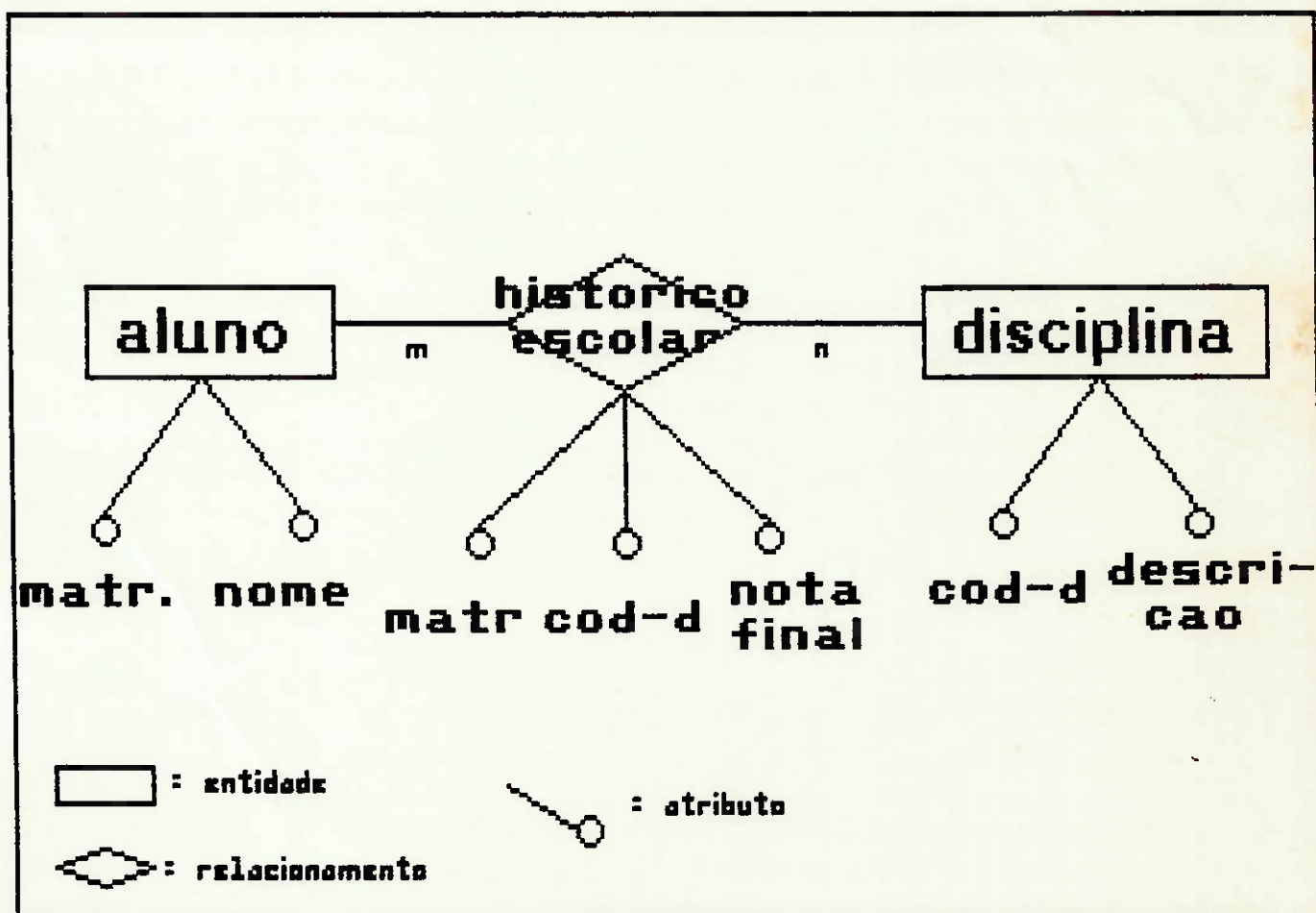
- a. **modelo entidade-relacionamento (MER)** - foi proposto por Peter Chen [11] e utiliza os conceitos acima apresentados. Além disso utiliza outros conceitos complementares:
 - . **atributo** - entidades e relacionamento possuem

propriedades que podem ser expressas em termos de pares de atributo-valor.

- . **identificador da entidade** - é um atributo ou conjunto de atributos que caracterizam, identificam a entidade.
Ex: nº-da-peça, código-do-projeto, nº-do-contrato, etc.
- . **identificador de relacionamento** - relacionamentos são identificados pela utilização dos identificadores das entidades envolvidas no relacionamento.

Mostramos a seguir um exemplo de um diagrama entidade-relacionamento com atributos (fig. 5.3.1.3.1). Conjuntos de entidades são representados por retângulos e conjuntos de relacionamentos são representados por losangos.

FIG. 5.3.1.3.1 - Exemplo de Diagrama Entidade-relacionamento
com Atributos.



b. **modelo orientado a objetos** - Sistemas orientados a objetos estão recebendo grande atenção atualmente nas áreas de automação de escritório, linguagens de programação, bancos de dados e inteligência artificial, porque é uma modelagem que possibilita maior facilidade de reuso de elementos (funções, rotinas, programas, etc.) no desenvolvimento de Sistemas. Embora o termo "objeto" tenha adquirido várias interpretações diferentes na literatura, um Sistema orientado a objeto tem as seguintes características:

- . abstração de dados,
- . herança de propriedade,
- . contém dados e operações.

Todas as entidades conceituais do Sistema são modeladas como objetos.

O comportamento de um objeto é expresso em termo de operações e, as operações são compostas pelo modo como é obtido o estado ou é manipulado um objeto. Os objetos podem se comunicar através de mensagens. As mensagens constituem a interface pública do objeto para qualquer mensagem compreendida por um objeto; existe uma operação correspondente que executa a mensagem retornando um objeto como resposta.

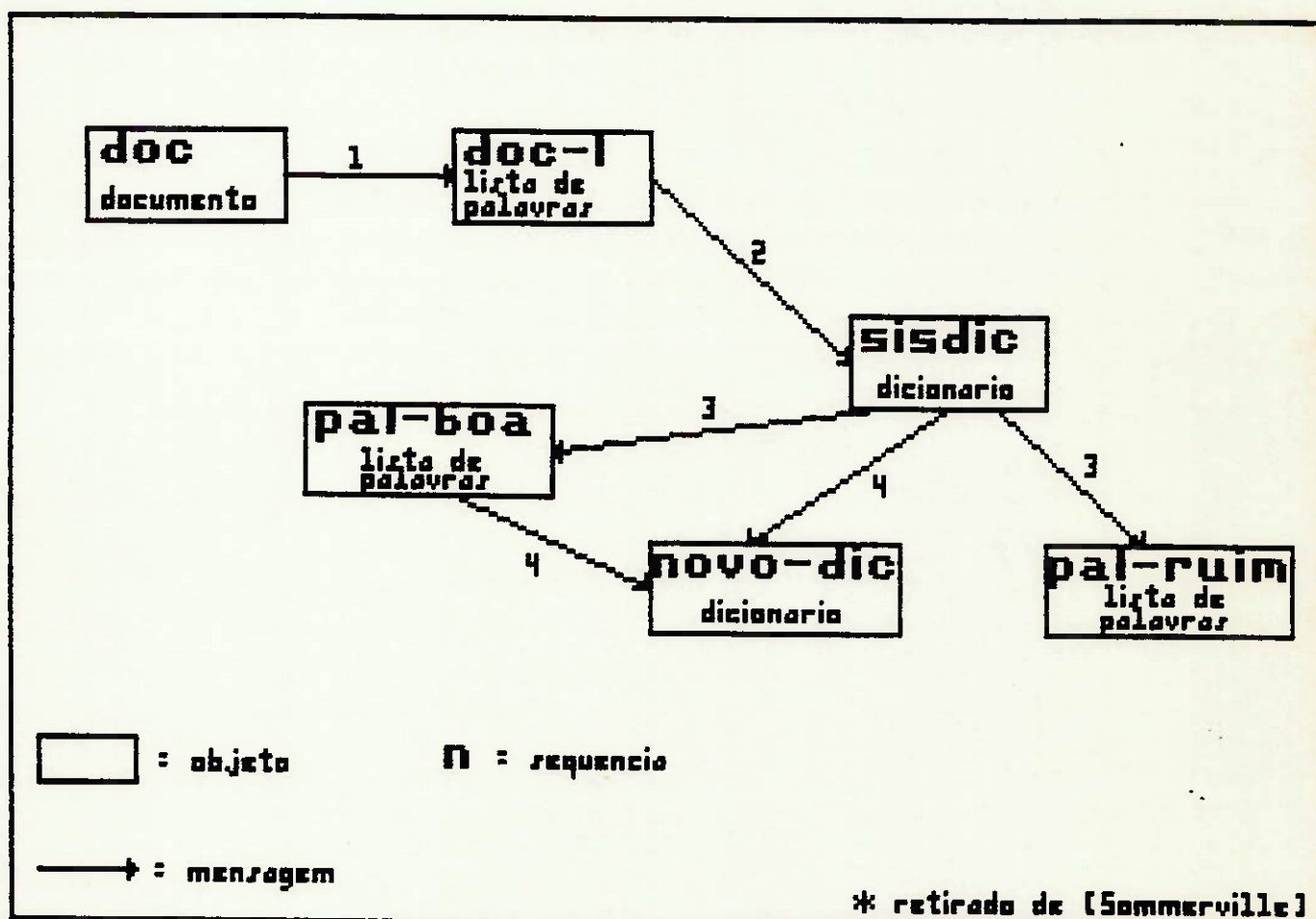
Por questão de simplicidade conceitual, objetos similares são agrupados em uma entidade chamada classe. Existe uma relação hierárquica, onde as propriedades (operações) e dados de uma classe são herdados por uma sub-classe (nível

inferior). Propriedades adicionais também podem ser especificadas para cada sub-classe. Nesse tipo de modelagem não existe a restrição de que uma classe tenha apenas uma super-classe (nível superior), como é o caso do modelo hierárquico (apresentado a seguir). Dessa forma a hierarquia pode ser extendida como uma rede ou um grafo acíclico (sem um sentido obrigatório).

O procedimento para a elaboração do Sistema por essa metodologia consta das seguintes fases:

1. Identificar os dados abstratos para cada subsistema.
 2. Identificar os atributos de cada dado abstrato identificado na fase anterior.
 3. Identificar as operações de cada dado abstrato.
 4. Identificar a comunicação (mensagens) entre os objetos.
 5. testar o projeto com exemplos
 6. verificar se não há herança, ou seja, se algum objeto não é um caso particular de outro.
 7. refinar o diagrama do Sistema repetindo os passos 1-6.
- Para melhor esclarecermos possíveis dúvidas, a fig. 5.3.1.3.2 contém um exemplo de diagrama voltado a objetos de um Sistema verificador de palavras, que identifica possíveis erros ortográficos. Notar que a sequência é mostrada pela enumeração das mensagens entre os objetos. As vantagens desta nova modelagem são muito importantes, como a reutilização de modelos, a possibilidade de modificar classes já definidas e o alto nível de coesão oferecido pelas classes.

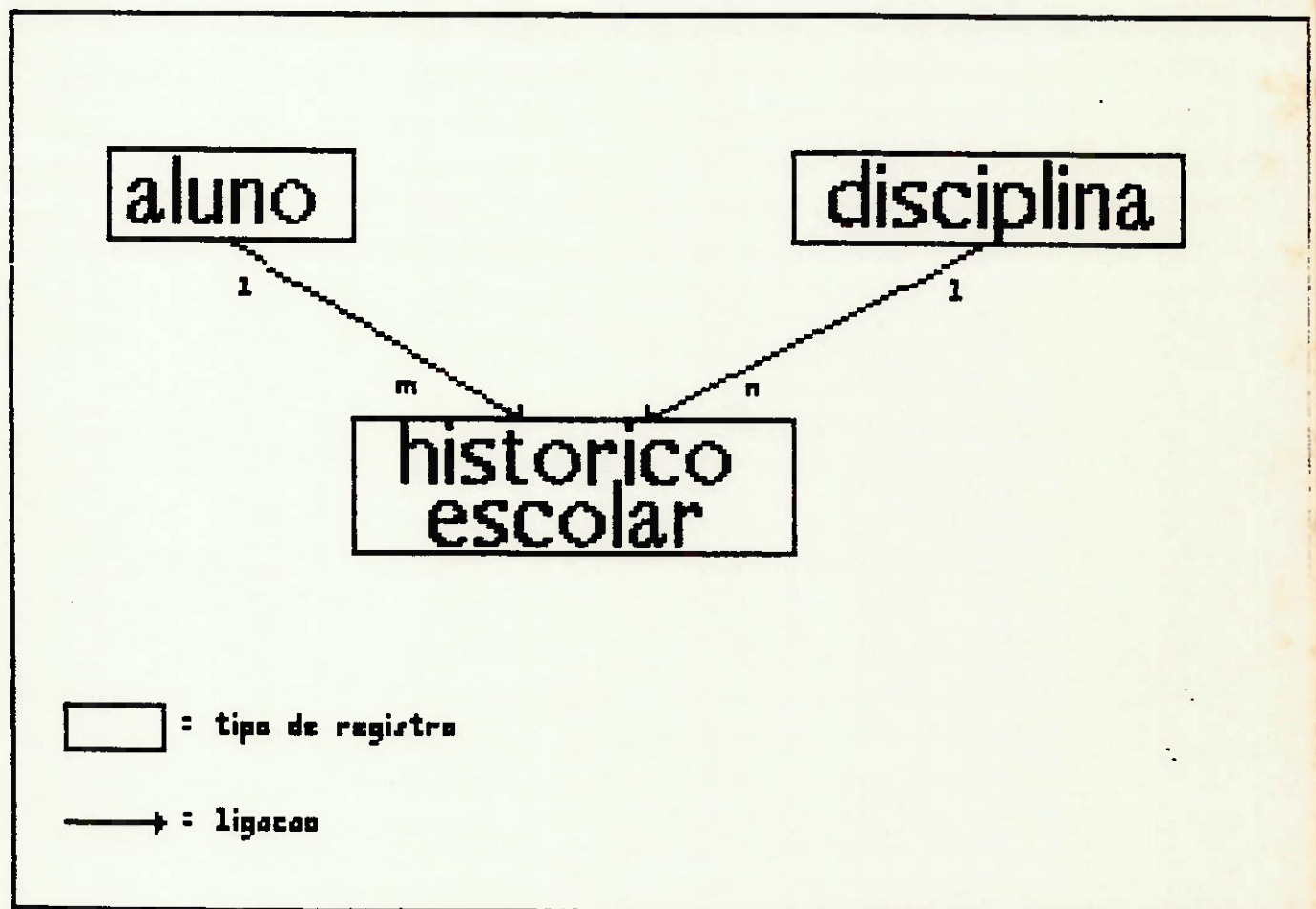
FIG. 5.3.1.3.2 - Exemplo de um Diagrama Voltado a Objetos.



A modelagem lógica compreende vários modelos, dentre eles:

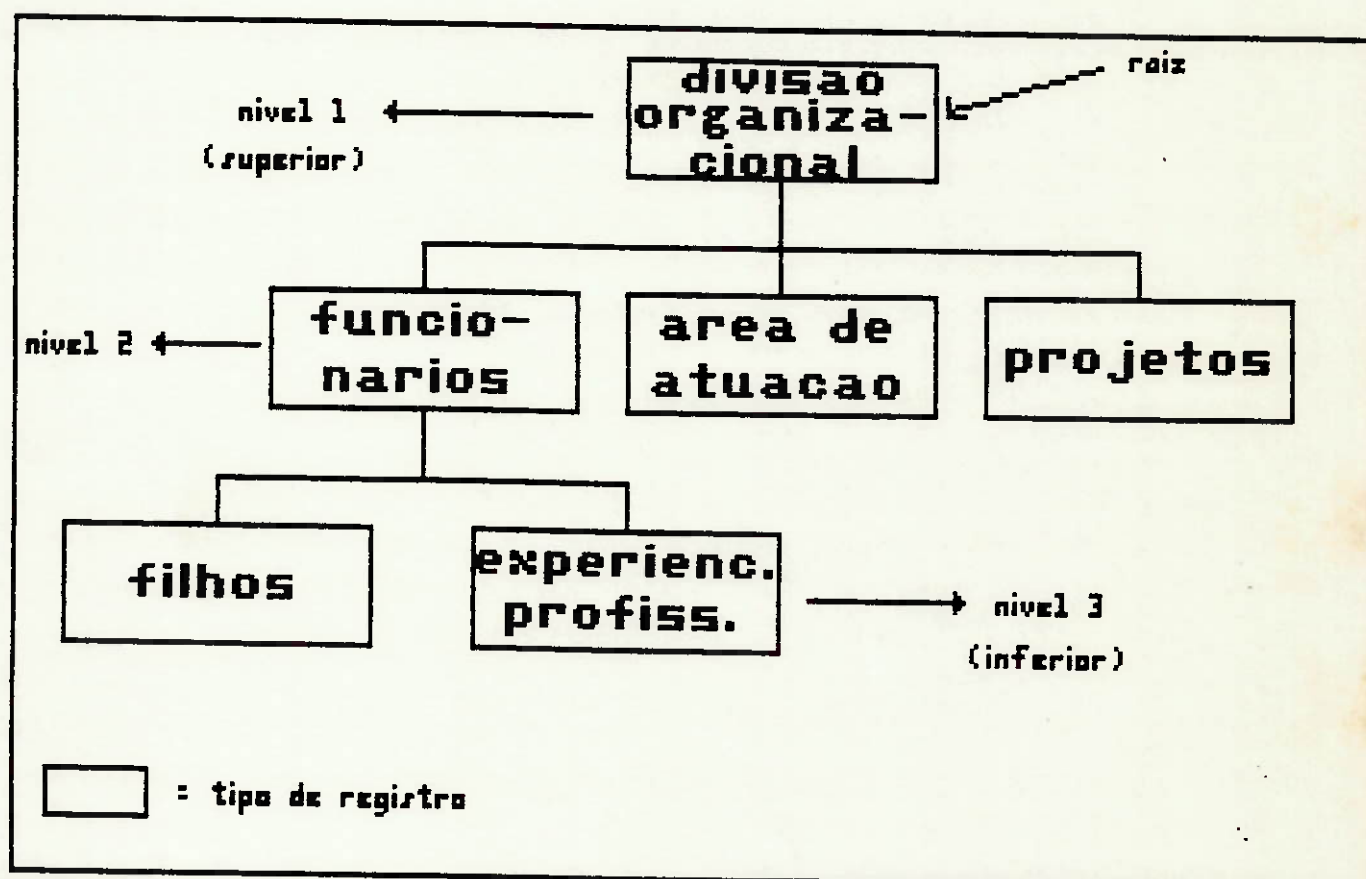
- a. **modelo rede** - o modelo rede utiliza como elemento básico de dados a ocorrência de registro (um conjunto de alguns tipos de dados relacionados entre si). Um conjunto de ocorrências de registros de um mesmo tipo determina um tipo de registro. Um tipo de registro é a representação de uma entidade. Um conjunto de tipos de registro relacionados entre si, através de referências especiais, forma uma estrutura de dados em rede. As referências especiais são conhecidas sob o nome de ligações (links). A representação gráfica do modelo foi formalizada por Bachman (1969) e chamada de diagrama de estrutura de dados. Um nodo do diagrama é chamado de tipo de registro e um arco, que representa uma ligação, é chamado de tipo "set". Esse diagrama mostra não somente os tipos de registros com as suas ligações, como também permite que seja feita a análise de rotas lógicas de acesso. O retângulo representa um tipo de entidade e a flecha é utilizada para relacionar dois tipos de registro ou entidade. O modelo hierárquico também pode ser representado por este diagrama. A fig 5.3.1.3.3 mostra um exemplo de um modelo rede.

FIG. 5.3.1.3.3 - Exemplo de um Modelo Rede.



b. **modelo hierárquico** - esse tipo de modelagem impõe certas restrições em relação a representação dos relacionamentos, e pode ser considerado como sendo um caso particular do **modelo rede**. As entidades são representadas como registros e os relacionamentos como ligações entre elas, de tal forma que uma estrutura de árvore ordenada é formada. Uma árvore ordenada é uma estrutura onde a ordem das sub-ordens é importante e as ligações são dirigidas da raiz (nível superior) para as folhas. A própria estrutura hierárquica já define as suas rotas de acesso, facilitando a manutenção do banco de dados, no entanto isto limita a flexibilidade deste último. O nível de um registro é equivalente ao comprimento do caminho a partir da raiz. Na fig. 5.3.1.3.4 é mostrado um exemplo de modelo hierárquico.

FIG. 5.3.1.3.4 - Exemplo de um Modelo Hierárquico.



c. **modelo relacional** - a modelagem relacional foi proposta por Codd [13] em 1970. Baseado em conceitos matemáticos, Codd propôs uma organização de dados utilizando-se de tabelas, às quais chamou de relações. Com uma abordagem marcada pela simplicidade de visualizar as informações, o modelo relacional permitiu o surgimento de Softwares de manipulação de dados simples e bastante flexíveis. Ao elaborarmos a relação, identificamos quais são as entidades e quais as propriedades envolvidas. A relação equivale à entidade, enquanto que os tipos de atributo são os próprios atributos da relação. Um exemplo de relação na sua forma tabular é mostrada a seguir para o caso da entidade denominada Funcionário.

atributos -->	! Nº FUNC !	NOME	! APELIDO !

	! 145367 !	José Aparício Santos	! Zé !
elementos	! 456589 !	João Carlos Silva	! João !
da relação -->	! 365120 !	M. Aparecida Lisboa	! Cida !
	! 984516 !	Marcelina Prado	! Lina !

Se for do interesse do leitor uma comparação mais extensiva dos recursos existentes, reproduzimos no Anexo 1 um questionário proposto em IFIP WG 8.1 [26] que auxilia esta análise.

5.3.2 Detalhando Algumas Metodologias

5.3.2.1 Apresentação

A fim de aprofundarmos mais o tema, descreveremos duas metodologias de forma mais completa, uma baseada em processos e a outra baseada em dados.

Não descreveremos outras metodologias por uma questão de tempo e por acreditarmos ser suficiente, para que o leitor deste trabalho tenha uma visão melhor, do que é uma metodologia para desenvolvimento de Sistemas.

Descreveremos as metodologias Remora e Method/1. A primeira é baseada em dados enquanto a segunda é baseada em processos. Descreveremos a Remora por propôr um novo modelo, que leva em consideração o tempo, o que torna o modelo mais completo que os demais. Já a Method/1 será descrita por ser muito abrangente e largamente utilizada pelas empresas.

5.3.2.2 A Metodologia Remora

Esta metodologia foi proposta por Colette Rolland [50] da Universidade de Paris.

O autor considera um Sistema de informação (SI) como sendo uma coleção de dados estruturados num banco de dados, um conjunto de programas e de atualizações realizadas sobre uma base de

programas, além de um conjunto de comandos sincronizados no tempo, que controlam a execução dos programas e a atualização dos dados.

Sabemos que o desenvolvimento de um SI completo e consistente é uma tarefa bem difícil. Principalmente quando o domínio de informação é grande e complexo, esta tarefa envolve por um longo período um número razoável de pessoas, executando atividades específicas.

Uma solução usual para a solução desse tipo de desenvolvimento complexo é a organização do processo em diversas etapas independentes. No caso dessa metodologia foram propostas duas etapas: a etapa conceitual e a etapa física.

Para cada etapa a solução é obtida por uma modelagem utilizando conceitos teóricos e ferramentas e pela expressão do modelo através de uma linguagem formal.

A primeira etapa é centrada na representação semântica do Sistema real. A solução desta etapa é denominada **esquema conceitual**, o qual é uma representação formal da estrutura natural dos fatos nas suas dimensões estática e dinâmica. A segunda etapa inclui os aspectos técnicos da solução, ignorados na fase anterior.

O esquema conceitual do SI permite uma representação completa, consistente, não redundante e econômica do Sistema real.

Acreditamos na hipótese, de que todos os aspectos reais do problema ou fenômeno devem ser representados no projeto do SI. Não somente os aspectos estáticos da organização (como num banco

de dados) devem ser representados, mas também a evolução no tempo de seus elementos. A transformação dos componentes da organização deve completar a estrutura estática, expressa pela estrutura de dados.

Ao que nos parece, o esquema conceitual proposto pela Remora é o único que representa a estrutura estática, as transformações e as inter-relações no tempo.

Para se definir o esquema conceitual, foi criado um modelo conceitual, o qual contém elementos para a construção do conjunto de dados, do conjunto de programas (processos) e do controle das relações no tempo entre dados e programas (sincronismo).

. Modelo:

O modelo foi definido através da análise do fenômeno real, o que leva a duas conclusões:

1. numa perspectiva dinâmica devemos representar três categorias de fenômenos, descritos de acordo com suas propriedades como objetos, eventos e operações.

2. a dimensão dinâmica é completamente representada por três categorias de associações entre as três categorias de fenômenos: modifica (operação, objeto), detecta (objeto, evento) e ativa (evento, operação).

. objeto - é um componente da organização durável, concreto ou abstrato, que pode ser particularizado. Como exemplos temos o cliente João ou o produto número 132.

. **operação** - é uma ação que pode ser executada num dado momento na organização e, que modifica o estado de um ou mais objetos. Um exemplo é a operação "analisar ordem" número 332 que cria o objeto "ordem aceita" número 202.

. **evento** - é qualquer coisa que pode acontecer num dado instante. É a constatação da mudança de estado de um ou mais objetos, e que significa a execução de uma operação. Por exemplo, o evento "chegada de ordem" é a constatação da criação do objeto ordem de número 44, que dispara a execução da operação "analisar ordem".

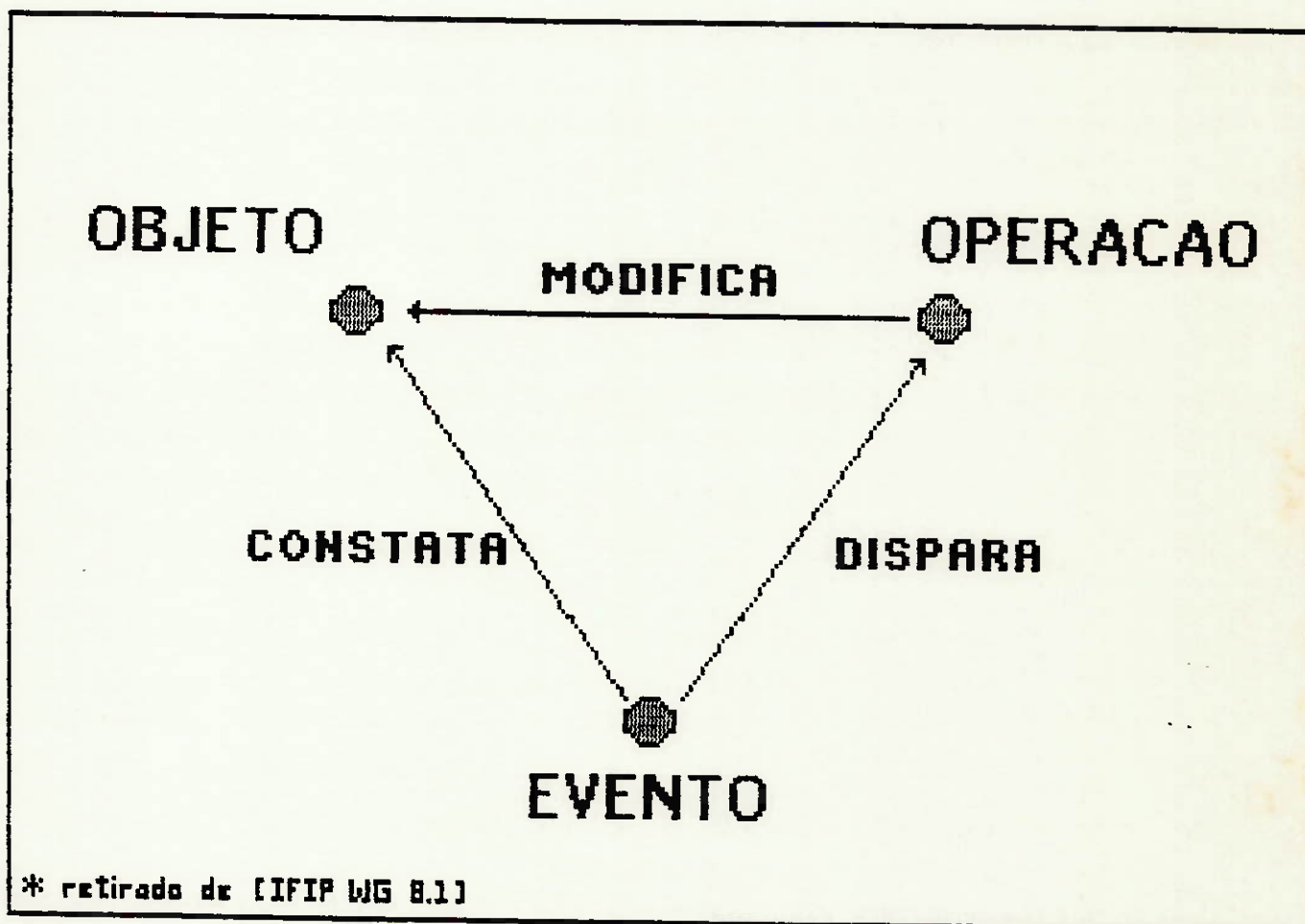
. **modifica (operação, objeto)** - é a associação que conecta uma operação a um ou mais objetos, onde uma operação modifica os objetos associados.

. **constata (objeto, evento)** - é a associação que conecta um evento a um ou mais objetos. Ela expressa que a mudança de estado do objeto é um evento.

. **dispara (evento, operação)** - é a associação que conecta um evento a uma ou mais operações. Ela expressa que um evento dispara uma ou mais operações.

A fig. 5.3.2.1 mostra a dinâmica da organização, onde os eventos causam a execução de operações, estas causam a mudança de estado dos objetos e estas mudanças podem ser eventos, os quais disparam operações.

FIG. 5.3.2.1 - Dinâmica da Organização



Salientamos a diferença nos seguintes conceitos:

- . estado - o estado de um objeto é indicado em um dado instante pelo valor de suas propriedades
- . mudança de estado - é sempre instantânea. Corresponde à passagem a novos valores de uma ou mais propriedades de um objeto.
- . evento - é uma mudança de estado particular que ativa uma relação dinâmica do Sistema. Nem toda mudança de estado de um objeto é um evento.

O modelo conceitual deve satisfazer duas condições: ser **formal** e representar de forma fácil e homogênea a realidade.

Para isso:

- . foi escolhido um **modelo relacional**.
- . foi introduzido o **tempo**.
- . foi introduzida a noção de **tipos de relação** (c-objeto, c-operação e c-evento) para diferenciar as categorias.

a. **c-objeto** - é uma relação do tipo permanente, onde cada atributo está em dependência permanente com a chave da relação. Uma dependência permanente entre dois atributos A e B ($A \rightarrow B$) é uma dependência elementar direta, onde qualquer que seja a de A e b de B, a e b têm a mesma duração de vida.

b. **c-operação** - é uma relação permanente, onde uma c-operação modifica um único objeto e de uma única maneira. Ela também representa uma regra de gestão da organização.

c. c-evento - refere-se a um c-objeto, correspondendo a uma única mudança de estado ou predicado, e gera uma ou várias operações.

Com esses conceitos é possível descrever um SI de forma integrada em seus dois aspectos:

- . estático (coleção de c-objetos) - corresponde à estrutura dos dados do SI, através da descrição dos c-objetos (entidades) e seus atributos. Esta descrição é feita pelo diagrama do Modelo Entidade Relacionamento (MER).
- . dinâmico (coleção de c-operações, c-eventos e c-objetos) - através das associações entre eventos, operações e objetos descreve-se as regras de evolução da organização, ou seja, a dinâmica da organização. Isto é representado através do diagrama dinâmico.

Um exemplo de aplicação desta metodologia, com o respectivo modelo dinâmico, é ilustrado no Anexo 2.

. Linguagem:

A linguagem utilizada para a descrição do modelo é a ISDEL (Information System Design Language). Suas características são:

- . permitir a descrição do modelo conceitual, através da descrição dos tipos de relação e dos controles de integridade associados.
- . ser utilizada pela MSI (ver ferramentas).

Para tal foi escolhida uma linguagem relacional completa, que permite fazer a descrição e manipulação do banco de dados.

A ISDEL contém construções derivadas de linguagens de terceira geração (PASCAL, PASCAL-R, etc.).

Através desta linguagem, podemos definir os textos das c-operações e das condições de aceitação dos eventos e de disparo das operações.

. Ferramentas:

A metodologia propõe a utilização de uma Máquina de Sistema de Informação (MSI) ainda abstrata.

Essa máquina possui as funções semelhantes de um Sistema gerenciador de banco de dados, só que mais complexas, já que o SI desenvolvido por essa metodologia é mais completo do que um banco de dados, por apresentar toda a dinâmica da organização.

A MSI deve desempenhar várias funções dentre elas:

- . geração da estrutura interna do SI,
- . criação do SI,
- . gerenciamento da evolução do SI,
- . utilização do SI.

De maneira simplificada, a MSI toma conhecimento das modificações ocorridas no SI e reage em função delas, ou seja:

- . identifica os eventos produzidos no SI.
- . determina as operações que deverão ser disparadas.
- . dispara essas operações e controla suas execuções.
- . certifica se as mudanças de estado ocorridas são eventos.
- . atualiza o SI.

Está sendo elaborada uma ferramenta na Universidade de Paris , que venha suportar o desenvolvimento de Sistemas na fase de análise e projeto.

. Método:

Existe um método completo, que fornece todo o procedimento para o projeto do SI. Esse método é utilizado para o projeto de SI, que necessitam do processamento automático dos dados e de resposta em tempo real.

Achamos que não cabe no escopo deste trabalho descrever esse método. Para os interessados recomendamos consultar Baraldi [2].

Esta metodologia introduz um aspecto muito importante no desenvolvimento de Sistemas: o tempo.

A Remora estabelece o sincronismo dos processos, ou seja, a ordem dos processos, ações no tempo. Em Sistemas muito complexos, onde existe um número grande de programas que devem ser executados, este é um fator muito importante para que as informações sejam processadas devidamente - na ordem correta - e para que assim seja garantida a veracidade das informações.

Além disso, através da abstração dos dados - definição destes de forma independente dos processos - é possível evitar a redundância dos dados e simplificar a manutenção, as

correções necessárias no projeto e na documentação do Sistema, pois alterações de suas funções não implicam em modificação da estrutura dos dados.

Além do mais, os dados de um Sistema são mais estáticos, alteram-se menos do que suas funções, as quais algumas vezes estão vinculadas ao aspecto operacional. Então, um modelo baseado em dados sofrerá menos alterações do que um que descreve as funções.

Ao elaborarmos o projeto de um Sistema baseado em dados, estaremos dando mais atenção a eles, assim será menos provável que modificações sejam feitas na estrutura dos dados. Mas, ao construirmos um modelo que descreve os processos com ou sem os fluxos de dados, provavelmente será mais difícil de se obter uma visão global dos dados, pois a atenção é maior para as funções do Sistema. Possivelmente, isto acarretará na redundância das informações e na necessidade de modificar a estrutura dos dados. Ora, é mais fácil alterar um programa do que uma base de dados, pois uma modificação da última acarreta na modificação de todos os programas que a utilizam.

Por todos estes motivos, acreditamos ser superior uma metodologia que utiliza a abstração dos dados.

5.3.2.3 A Metodologia Method/1

Esta metodologia foi elaborada pela empresa de consultoria Arthur Andersen [40] e é muito utilizada no mercado.

Ela trata de todas as fases do desenvolvimento de Software: planejamento de informações, projeto preliminar de Sistema, instalação de Sistema e suporte de Sistema em produção.

É muito abrangente e detalhada; por este motivo descreveremos apenas a etapa de projeto preliminar de Sistemas.

. Modelo:

A Method/1 não propõe nenhum modelo original para representar o problema. Utiliza diversos modelos já consagrados, alguns baseados em processos e outros baseados em dados.

Apesar de utilizar a análise dos dados essa metodologia se baseia em processos, pois ela parte da descrição das funções ou processos, oriundos da análise funcional, para estruturar os dados que estão envolvidos.

A análise funcional é um enfoque lógico e sistemático para identificar, descrever e documentar as funções que um novo Sistema deverá suportar a fim de satisfazer os objetivos do usuário. Não é um técnica altamente especializada.

Uma vez identificadas as funções do novo Sistema, através de entrevistas e observações, o próximo passo é documentá-las. O diagrama utilizado para esta documentação é o Diagrama de Warnier.

Detalhamentos sucessivos do diagrama são realizados até se obter o nível de detalhe desejado. Então, faz-se a descrição de cada função, através de uma descrição geral da função, dos dados (informações) necessários para a função e dos passos envolvidos para se completar a função.

A fim de se determinar os dados de entrada e saída do Sistema, utiliza-se o fluxograma, isto irá auxiliar a definição da estrutura de dados necessários para suportar as funções que foram identificadas.

As vantagens da análise funcional no projeto do Sistema é a garantia de que o novo Sistema suportará todas as funções necessárias para satisfazer os objetivos do usuário, e a facilidade de comunicação com o usuário a respeito do Sistema.

Segundo a metodologia, para se desenvolver a análise de dados é necessário possuir uma profunda compreensão da aplicação e de suas funções. E os relacionamentos entre dados só serão compreendidos de maneira efetiva, depois que as funções que se apóiam sobre esses dados forem entendidas. Deste modo, primeiro se determina as funções através da análise funcional e, então, para cada função determina-se os dados envolvidos.

A metodologia utiliza quatro níveis de modelos de dados de acordo com sua abrangência:

- . modelo de dados da organização (MDO) - mostra os dados e os relacionamentos dos dados envolvidos na organização. Ele fornece uma estrutura global para a arquitetura de dados de Sistemas da organização. Seu objetivo é orientar o trabalho subsequente, identificando os arquivos na

organização.

- . **modelo de dados de grupo de Sistemas (MDGS)** - é um subconjunto do MDO. Ele mostra os dados e os relacionamentos relevantes que serão usados por um conjunto de Sistemas de aplicação estreitamente relacionados.

- . **modelo de dados do projeto (MDP)** - cada projeto da organização requer um subconjunto do MDGS, este subconjunto chama-se MDP. Ele mostra os dados e os relacionamentos que deverão ser acessados por um Sistema único.

- . **modelo de dados de função (MDF)** - descreve as entidades que são acessadas e a ordem em que são acessadas para executar a função.

O diagrama utilizado, para representar os quatro níveis de modelo de dados, é o mesmo do Modelo Entidade Relacionamento (MER).

- . **Linguagem:**

A Metrod/1 não propõe nenhuma linguagem para a representação dos modelos utilizados.

. Ferramentas:

Foi construída uma ferramenta, denominada Design/1, que vem auxiliar, suportar a análise e a representação ou construção dos modelos.

Nesta metodologia também podem ser utilizadas quaisquer ferramentas existentes no mercado, desde que estas estejam voltadas para os diagramas utilizados (diagrama de Warnier, fluxograma, diagrama do MER).

. Método:

Basicamente o procedimento estabelecido por esta metodologia para a execução do projeto preliminar do Sistema pode ser resumido da seguinte maneira:

1. determinar as funções envolvidas no problema através do diagrama de Warnier.
2. descrever estas funções através de uma definição geral da função, dos dados envolvidos na função e dos passos necessários para completá-la.
3. para cada função identificar as entidades existentes.
4. definir o modelo de dados da função (MDF).
5. repetir os itens 3 e 4 para todas as funções.
6. agrupar todos os MDFs para obter um Modelo de Dados do Projeto.

O Anexo 2 contém um esquema do método definido para as quatro fases da metodologia.

Esta metodologia é interessante porque fornece um procedimento detalhado de tudo o que deve ser feito para o desenvolvimento de um Sistema, o que poucas fazem.

Representa a realidade a partir das funções existentes e para cada função é construído um modelo de dados, que no final são combinados.

Apesar de existir o modelo dos dados, acreditamos que uma modelagem que parte de cada função, não possuirá uma visão global da estrutura de dados do problema. Possivelmente, isto acarretará numa base de dados falha e redundante.

5.3.3 Prototipação

Um dos principais problemas da denominada Crise do Software é a dificuldade de comunicação entre usuários e analistas, o que origina uma série de erros na fase de especificação dos requisitos do Software, gerando uma alta manutenção deste.

Como mencionamos anteriormente, a chave de um bom Sistema é um projeto bem elaborado, o que só é possível mediante uma boa definição dos requisitos do Sistema.

A prototipação é um recurso centrado em metodologia, que vem auxiliar a comunicação entre usuários e analistas principalmente na fase de especificação dos requisitos do Software, diminuindo os erros originários desta fase.

A etimologia da palavra protótipo vem da forma francesa, que é derivada do grego ("protos" + "topos") significando "primeiro modelo" ou "primeiro tipo". A prototipação de um Sistema é o seu primeiro modelo.

Na indústria, antes de um produto entrar em fabricação, constrói-se um primeiro modelo do produto (protótipo), a fim de que testes experimentais sejam feitos, para ver se este é adequado ao uso e à produção. Este é um processo que dá a certeza de existir menos erros, quando o produto final for colocado no mercado. Como já dissemos, desenvolver Sistemas nada mais é do que uma indústria de geração de aplicativos - A Fábrica de Software - onde o mesmo procedimento pode ser aplicado.

O desenvolvimento de Sistema é um processo que requer um aprendizado concorrente entre o analista e o usuário. O analista primeiro precisa entender as necessidades do usuário, sua tarefa principal é ajudar o usuário a formalizar suas necessidades e processos de decisão. O usuário, por sua vez, precisa aprender as técnicas de modelagem de Sistemas utilizadas pelo analista e, entender o escopo de especificação do projeto. Enfim, este é um processo de mútuo aprendizado entre analistas e usuários.

A abstração ajuda o homem a entender um Sistema complexo através da sua simplificação. Ferramentas abstratas podem ser os fluxogramas, diagramas hierárquicos, diagrama de Warnier, etc. Através deste tipo de ferramenta o analista e o usuário podem se comunicar a respeito do projeto do Sistema e realizar aquele processo de mútuo aprendizado.

Normalmente, uma ferramenta apropriada ao analista pode não ser adequada ao usuário entender a estrutura do Sistema. Muitas vezes o usuário concorda com a especificação, quando na realidade ele não entendeu o projeto.

A ferramenta abstrata mais apropriada é aquela que encoraja o usuário a se envolver e a se sentir dono do projeto, além de suportar, ou pelo menos não impedir, o processo de aprendizado do analista e do usuário. A prototipação satisfaz este critério talvez melhor do que qualquer outra ferramenta abstrata, porque ela ajuda a converter uma construção mental ou uma idéia em um Sistema tangível e em funcionamento. O resultado deste recurso é algo que o usuário possa compreender, apesar dele não possuir conhecimentos específicos em desenvolvimento de Sistemas e em operação.

Um protótipo pode ser construído para modelar todo um Sistema ou somente parte dele e, pode ser criado em qualquer fase do ciclo de vida. Uma tela, um relatório, uma iteração entre arquivos ou um programa "batch" podem ser prototipados. Mas um protótipo para ser eficiente deve aparentar e agir como o Sistema alvo.

A forma de desenvolvimento do ciclo de vida depende do nível de incerteza dos requisitos do Sistema. Prototipação, a qual afeta o processo de desenvolvimento do Sistema, pode ser aplicada de acordo com o tipo do Sistema, o qual é um reflexo do nível de incertezas que possui.

Existem três tipos de Sistema:

- . Sistema de processamento de transações - possui um baixo

nível de incertezas, mas tem um alto grau de estrutura e um alto nível de atividade. Sua complexidade está na sofisticação de comunicação de dados entre estações de trabalho e base de dados. Este Sistema caracteriza-se por possuir como função básica uma adição, ou eliminação, ou modificação de registros. O protótipo é utilizado principalmente na fase de projeto preliminar do Sistema.

. **Sistema de controle e relatórios** - este tipo de Sistema possui um grau maior de incertezas dos requisitos. Suas aplicações são controlar e alocar os recursos da organização.

. **Sistema de apoio a decisão** - este Sistema possui o nível mais alto de incerteza de seus requisitos. Ele caracteriza-se por responder o seguinte tipo de pergunta ao usuário: quais as consequências existentes se isto acontecer? O que se deve fazer se este fato ocorrer? Nesta categoria de Sistemas, quando se realiza a prototipação, executa-se todo o ciclo de vida, apenas de forma mais rápida.

A maioria dos Sistemas exibem características operacionais pertencentes aos três tipos de Sistema apresentados. Poucas aplicações se enquadram apenas num tipo. É tarefa do analista e do usuário determinarem conjuntamente quais aspectos do Sistema se enquadram em que categoria. Uma vez isto estabelecido, o analista pode determinar o grau adequado de prototipação.

Deve ficar bem claro que a prototipação não substitui o ciclo de vida do Software, apenas lhe dá suporte.

Existem dois tipos de protótipos:

- . **protótipo rápido** - neste caso primeiro se especifica os requisitos básicos do Sistema, para então se construir um protótipo. Este é examinado pelo usuário que faz as modificações necessárias, assim se altera o protótipo até o momento em que o usuário estiver satisfeito com os resultados. O produto desta fase chama-se protótipo operacional. A partir deste protótipo operacional se faz as modificações na especificação de requisitos original. Esta última especificação forma a base a partir do qual o projeto do Sistema e sua implantação são elaborados. Como podemos verificar o protótipo é utilizado apenas para obter os requisitos do Sistema, após isto ser feito ele é abandonado, mas todo o processo de desenvolvimento é baseado nele.
- . **protótipo evolutivo** - este tipo de protótipo obedece a todos os passos de um protótipo normal, só que ao invés de se descartar os protótipos de versões anteriores, as sucessivas versões do protótipo vão evoluindo para compor o Sistema final do usuário e, são utilizadas à medida que se tornam aceitas como válidas, na abrangência a que se propunham. Neste caso para que o protótipo tenha êxito, depois de determinados os requisitos básicos do Sistema se faz a análise de dados e a funcional (projeto do Sistema), para então fazer a primeira versão do protótipo. Logicamente, estas análises não são tão detalhadas e abrangentes como no ciclo de vida clássico.

A prototipação realmente vem ao encontro do problema de comunicação entre analistas e usuários no desenvolvimento de Sistemas, pois faz com que o Sistema seja algo palpável para a percepção do usuário. Desta forma o usuário possui condições de prever grande parte de suas necessidades e, o analista tem condições de prever a possibilidade de atendimento.

Uma das vantagens da prototipação é que ela pode ser aplicada independentemente da metodologia utilizada para o desenvolvimento do Sistema. Isto vem facilitar muito o seu aprendizado, porque o analista pode usar a metodologia a qual já está acostumado.

Tendo em mãos este recurso, grande parte dos problemas que implicam na manutenção do Sistema tais como: Sistemas que não atendem a necessidade do usuário, Sistemas cujos requisitos não foram amplamente definidos, Sistemas que não se assemelham ao que o usuário imaginava, serão eliminados.

Construindo um protótipo, estaremos reduzindo bastante o nível de manutenção que sofrerá o Sistema, o que diminui consideravelmente seu custo.

Existe um problema que deve ser considerado na utilização de protótipos: a atualização da documentação.

Este tipo de dificuldade também existe num grau ainda maior na aplicação de metodologias sem protótipo, as quais formalizam o pensamento através de modelos, que são modificados constantemente, exigindo um esforço grande na sua atualização e nos documentos relacionados, tornando o processo aborrecido,

lento e em muitos casos ineficaz. Em virtude disto, as técnicas são mal ou parcialmente aplicadas, conseqüentemente as vantagens deixam de ser materializadas.

Na tentativa de solucionar estes e outros problemas, surgiu a idéia de automatizar o processo de especificação do Sistema e a respectiva documentação através do CASE (Computer Aided Software Engineering). No próximo capítulo este assunto será abordado.

6 CASE - ENGENHARIA DE SOFTWARE ASSISTIDA PELO COMPUTADOR

6.1 Resumo

Neste capítulo são introduzidos os conceitos envolvidos na automação do desenvolvimento de Sistemas, ou melhor, da Fábrica do Software.

Também é apresentada uma nova tecnologia - O CASE - que vem propiciar esta automação. Para complementar, foi realizada uma análise comparativa de alguns produtos existentes no mercado, além de uma apresentação daquilo que vem sendo desenvolvido a nível de pesquisa no país.

Para finalizar são apresentadas as tendências de evolução. As dificuldades, que vêm surgido para utilização desta tecnologia, são apresentadas no próximo capítulo.

6.2 A Automação na Fábrica de Software

"Em casa de ferreiro, espelo é de pau". Este ditado retrata bem a situação da área de informática: ela é a que menos utiliza os recursos informáticos para auxiliar seu trabalho.

Automatizar a Fábrica de Software provavelmente resolveria os problemas da Crise do Software - produtividade e qualidade - que outros recursos, aplicados isoladamente da automação, não conseguiram resolver, como é o caso do ciclo de vida, metodologias, prototipação, linguagens de quarta geração, banco de dados relacional e etc.

A automação também é um meio de se transformar o processo produtivo de "grandes projetos" em uma "produção contínua diferenciada", que é o que se propõe com a idéia da Fábrica de Software.

Para justificarmos esta transformação do processo produtivo de Software, faremos um paralelo com o mesmo processo na indústria automobilística.

Como Susan Ball [1] descreve, a indústria automobilística comemorou seu septuagésimo aniversário em 1986. De 1886 até o início da década de 1890, a indústria esteve no seu estágio de invenção, no qual as novas idéias eram experimentadas e poucos carros eram vendidos. Em 1891, as três principais marcas de automóveis da Europa produziam um total de 17 carros. Nesta época, a maioria das estradas não eram apropriadas aos automóveis.

No meio da década de 1890, a indústria passou a produzir sob encomenda. A produção média anual de uma indústria automobilística era de aproximadamente 50 veículos. Em 1900 a produção subiu de 100 para 200 carros por ano; no entanto, a oferta ainda não estava atendendo toda a demanda. Havia uma fila de espera de dois a vinte meses. Estes carros não eram de fácil utilização, nem confiáveis e necessitavam de uma manutenção constante.

Em 1908, a indústria automobilística passou a ter uma produção em massa (contínua) com a produção de Henry Ford do modelo T, usando a técnica de linha de montagem. Em 1909, sua firma era capaz de fabricar 12.000 carros, já em 1923 a produção era de 1,8 milhões de veículos por ano.

A produção contínua foi baseada em quatro princípios:

1. **divisão do trabalho** - a produção de automóveis foi dividida em uma série de etapas e uma pessoa realizava apenas uma etapa.
2. **componentes padronizados** - ferramentas foram criadas para produzir componentes para os carros padronizados.
3. **integração** - o resultado de uma etapa é a entrada da próxima.
4. **automação** - no caso de Ford, foi utilizada uma linha de montagem.

Em 1921 Ford possuía 55% do mercado automobilístico, em 1926 este número caiu para 30% e em 1930 a General Motors se tornou a líder do mercado. Por quê? Porque o mercado cresceu e o consumidor tornou-se mais sofisticado e exigente. Ele não queria

mais um único modelo de carro, eles queria opções. A General Motors percebeu essa necessidade e passou a produzir diferentes modelos de automóveis de diferentes linhas de produção (produção contínua diferenciada). Ford, por sua vez, centralizou a produção e não era tão flexível.

Da mesma forma que na indústria automobilística, podemos aplicar na Fábrica de Software os quatro princípios em que se baseou a produção contínua e o princípio de flexibilidade utilizado pela General Motors, para transformarmos a produção de Software do tipo de "grandes projetos" para uma "produção contínua diferenciada".

O primeiro princípio - divisão do trabalho - já vem sendo aplicado através do ciclo de vida do Software e da aplicação de metodologias de desenvolvimento de Sistemas.

Os demais princípios vão ser aplicados através da utilização do CASE (Computer Aided Software Engineering) - a Engenharia de Software Assistida pelo Computador. Ele deve permitir uma padronização dos elementos do Software, integrar as fases de desenvolvimento de Sistemas, ou seja, o resultado do trabalho executado numa fase deve ser a entrada da seguinte ou o elemento a partir do qual a próxima etapa será realizada e, além disso, deve automatizar processo de desenvolvimento do Software.

Vejamos melhor o que é CASE.

6.3 Definição e Considerações sobre CASE

CASE é a automação do processo de desenvolvimento e manutenção do Software ou, de forma ampla, é a automação de "qualquer coisa" que o homem faça para o Software. Seus objetivos são ambiciosos, por pretender suportar todo o ciclo de vida do Software.

É uma combinação de metodologias para desenvolvimento de Sistemas com as ferramentas de Software que dão suporte às diversas fases das metodologias. Essas ferramentas utilizam normalmente interfaces gráficas poderosas e automatizam a verificação do cumprimento das regras e padrões definidos pela metodologia.

Em geral um CASE contém os seguintes elementos (fig. 6.3.1):

a. **repósitório central de informações** - este é o coração, o elemento mais importante do CASE. Ele armazena e organiza todas as informações necessárias para criar, modificar e desenvolver o Sistema. Estas informações incluem, por exemplo, as estruturas de dados, as lógicas de processamento, regras, código fonte, definições de telas e relatórios, dados de teste e dados de administração do projeto. O ideal seria que este repositório de informações fosse ligado ao banco de dados corporativo.

b. **ferramentas "front-end"** - são utilizadas nas fases anteriores a programação do Sistema (análise e projeto do Sistema). Um requisito fundamental para estas ferramentas é a boa qualidade de desenho dos diagramas de estrutura dos programas, de dados, diagramas de fluxos de dados, layout das

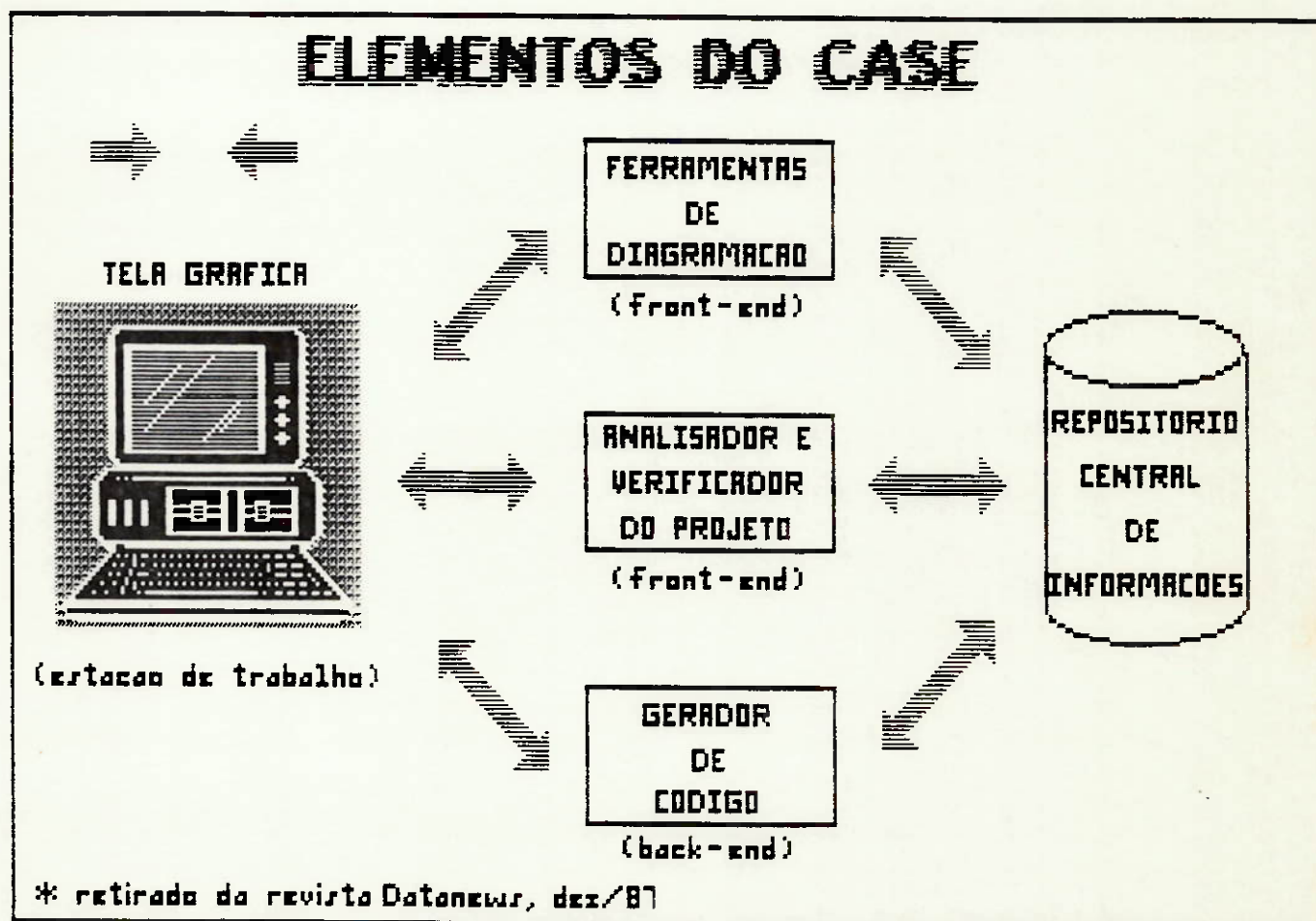
telas, etc. Mais do que armazenar as representações, estas ferramentas guardam o significado dos itens diagramados. Isto permite que uma mudança feita em um diagrama, seja refletida automaticamente nos diagramas relacionados. Uma outra propriedade destas ferramentas é a capacidade de fazer a consistência do projeto e dos diagramas, de acordo com as regras da metodologia utilizada.

c. ferramentas "back-end" - são os geradores de código, para geração automática do código dos programas. Acredita-se, que para este tipo de ferramenta ter sucesso, é necessário que ela possua interface para vários tipos de geradores de código.

d. estação de trabalho - é o último componente de um Sistema CASE. A maioria é baseada no computador pessoal (PC). Devido as manipulações gráficas é necessário a existência de uma estação com a sua própria unidade de processamento e memória.

Normalmente, não são incluídos como componente do CASE as metodologias de desenvolvimento de Sistemas e de administração de projetos, no entanto elas são imprescindíveis para que os Sistemas resultantes tenham bons níveis de qualidade.

FIG. 6.3.1 - Elementos de um CASE.



O CASE deve ser mais do que uma ferramenta, ele deve ser um ambiente de desenvolvimento de Sistemas para desenvolver Sistemas grandes e complexos, garantindo produtividade e qualidade.

. produtividade - nada do que foi utilizado até hoje (linguagens de quarta geração, geradores de código) aumentou significativamente a produtividade no desenvolvimento de grandes Sistemas. Apesar de numerosas ferramentas de automação ajudarem as diversas fases de desenvolvimento, estas ferramentas não são integradas, assim a saída de uma não pode ser utilizada como entrada de outra. Até que haja um CASE que integre todas as fases, um aumento significativo da produtividade não será alcançado.

. qualidade - é um dos objetivos que o CASE deve atingir e, dependendo do ponto de vista, possui vários significados. A partir do ponto de vista do usuário a qualidade de um produto significa ele ser confiável, durável, fácil de lidar e modificar, amigável, consistente e extensível; do ponto de vista do analista deve ser inteligível e fácil de ser modificado.

Para atingir estes dois objetivos (qualidade e produtividade), o ambiente de desenvolvimento proporcionado pelo CASE deve:

a. ajudar a rejuvenecer e migrar aplicações antigas - a maioria dos departamentos de desenvolvimento de Sistemas necessitam de ferramentas que os auxiliem na melhoria das aplicações já existentes, de tal forma que elas possam ser modificadas sem causar sérios problemas operacionais, ou

extendidas para executar novas funções, ou que sejam processadas de forma mais eficiente. Assim, a ferramenta de desenvolvimento deve auxiliar as equipes a lidarem com os seus Sistemas antigos.

b. transferir o trabalho das pessoas para o computador - como o custo da mão de obra está subindo e o dos computadores está diminuindo, esta transferência do trabalho deve ocorrer. O computador pode auxiliar na elaboração dos diversos tipos de diagramas e modelos existentes (modelo de dados e funções, diagrama corporativo, prioridades de desenvolvimento, etc.). Além do mais, o computador pode checar a consistência, validação e completude do projeto e, também, guiar o analista nas suas tarefas.

c. integrar as fases de desenvolvimento - a integração é necessária, para se minimizar os erros introduzidos quando as pessoas transferem as idéias sob uma forma (diagrama de projeto) para outra (respectivo código) e, para realmente aumentar a produtividade, pois o trabalho resultante de uma fase pode ser utilizado na próxima. Também a integração permite que diferentes ferramentas compartilhem as informações, assim uma mudança feita em algum lugar será refletida nos pontos relacionados. Para que sistemas sejam reutilizados se faz necessário que haja integração entre as ferramentas das diversas fases.

d. permitir o reuso - o reuso de Sistemas ou partes de Sistemas já desenvolvidos aparenta ser a chave para se obter produtividade e qualidade. A qualidade pode ser aumentada pela reutilização de Sistemas já testados. A produtividade pode ser melhorada pela diminuição do tempo de desenvolvimento. Inicia-se

Já com alguma coisa pronta, a qual será alterada onde necessário, o que é melhor do que começar do nada. Para se aplicar o conceito de reuso são necessárias três coisas: um mecanismo de identificação dos componentes reutilizáveis existentes; um suporte ao reuso, onde o programador seria informado dos impactos causados por uma mudança no elemento; e uma biblioteca de componentes para reuso com um gerenciador de biblioteca.

Existem dois requisitos básicos para se utilizar um CASE:

- . a empresa deve possuir um ciclo de vida padronizado para o desenvolvimento de suas aplicações, a fim de que estas possam ser padronizadas e reusadas.
- . a empresa deve medir efetivamente sua produtividade, para então ter condições de quantificar os benefícios obtidos com a aplicação do CASE.

Tipicamente as ferramentas de CASE oferecem as seguintes funções:

- a. **diagramação** - utilização do computador para desenvolver os diversos diagramas usados nas diversas fases do ciclo de vida do Software.
- b. **prototipação** - através da especificação dos relatórios e telas do Software, bem como da sua execução simulada, é possível a construção rápida de protótipos. Estes protótipos servem para a validação antecipada, por parte dos usuários, das funcionalidades do Software a ser desenvolvido.
- c. **manutenção do repositório central de informações do projeto** - centralização, num dicionário de dados único,

de todas as informações do projeto, permitindo a conversão automática de formatos entre as diversas ferramentas, a verificação de erros e a produção de relatórios com informações cruzadas sobre os elementos do projeto. Este repositório central é o instrumento principal de interligação funcional entre as ferramentas.

d. **verificação de erros** - automaticamente as ferramentas de CASE podem verificar erros ou omissões na especificação de um Software, bem como garantir os padrões e normas da metodologia usada.

e. **geração de código** - através da interação com o projetista a especificação pode ser convertida em código, total ou parcialmente, permitindo maior eficiência nas fases de implementação e de manutenção. Geralmente, é produzido um esqueleto de código, que precisa ser "recheado" pelo programador a posteriori.

f. **documentação automática** - como subproduto da automação do projeto, os Softwares de CASE podem gerar e auxiliar na manutenção da documentação do projeto.

Como já mencionamos, o CASE é um ambiente de desenvolvimento de Sistemas que suporta as atividades de especificação, projeto, implementação, teste e documentação do Software. Para isto, algumas características são essenciais a um Software de CASE, tais como:

- a. capacidade gráfica para desenhar diagramas estruturados.
- b. repositório central de informações do projeto.
- c. ferramentas integradas de Software de apoio.

- d. cobertura de todo o ciclo de vida do Software.
- e. suporte a prototipação.
- f. geração automática de código a partir da especificação.
- g. suporte a padronização do uso de metodologias, com verificações embutidas nas ferramentas.

A maior parte dos CASEs oferece três tipos de diagramas básicos: diagrama de fluxo de dados, diagrama de modelagem de dados e diagrama hierárquico de módulos. Além destes, outros tipos de diagramas são utilizados, dependendo do produto específico, tais como diagrama de navegação de dados, fluxograma, diagrama corporativo, etc.

O coração do CASE é o repositório central de informações. Suas informações servem de base para a emissão de relatórios e verificação do projeto.

Os relatórios emitidos, tipicamente, são os seguintes:

- a. referência cruzada - por exemplo, todos os DFDs onde os fluxos de dados aparecem.
- b. explosão de objetos - por exemplo, listagem de processos, com os processos que são seus descendentes.
- c. traçado de caminho - por exemplo, listagem de todos os objetos no caminho entre dois objetos num dado DFD.
- d. descrição de objetos selecionados do repositório central - por exemplo, listagem de todos os objetos que foram atualizados após certa data.

Na verificação do projeto são feitas, por exemplo, as seguintes análises:

- a. verificação de erros de sintaxe - por exemplo em um DFD,

- um processo P1 deve ter pelo menos um fluxo de entrada.
- b. definições incompletas de objetos - listagem dos objetos do repositório central que não foram ainda completamente definidos.
 - c. balanceamento entre os níveis de um DFD - análise da conservação dos dados, verificando que não haja perda nem inclusão de dados entre os níveis de uma família de DFDs.
 - d. análise de conectividade dos DFDs - listagem das conexões ilegais dos objetos em todos os DFDs ou em DFDs selecionados.
 - e. análise de fluxos de cada DFD - listagem dos dados que são entrada ou saída para cada processo ou depósito, indicando os elementos de dados derivados ou atualizados por cada processo.

Os Sistemas CASE são utilizados em conjunto com Metodologias de Desenvolvimento de Sistemas, em dois níveis:

- a. na automação da documentação requerida pela metodologia. Neste nível, as ferramentas do CASE são usadas para auxiliar no desenho de diagramas e na sua verificação. Dependendo da metodologia usada, ferramentas diferentes são necessárias.
- b. na automação das fases da metodologia. Neste caso, o CASE orienta o profissional para produzir os relatórios e documentos de cada fase e verifica o cumprimento dos padrões adotados. Por exemplo, o analista não pode iniciar o trabalho numa fase seguinte sem que todos os documentos da fase anterior tenham sido criados e

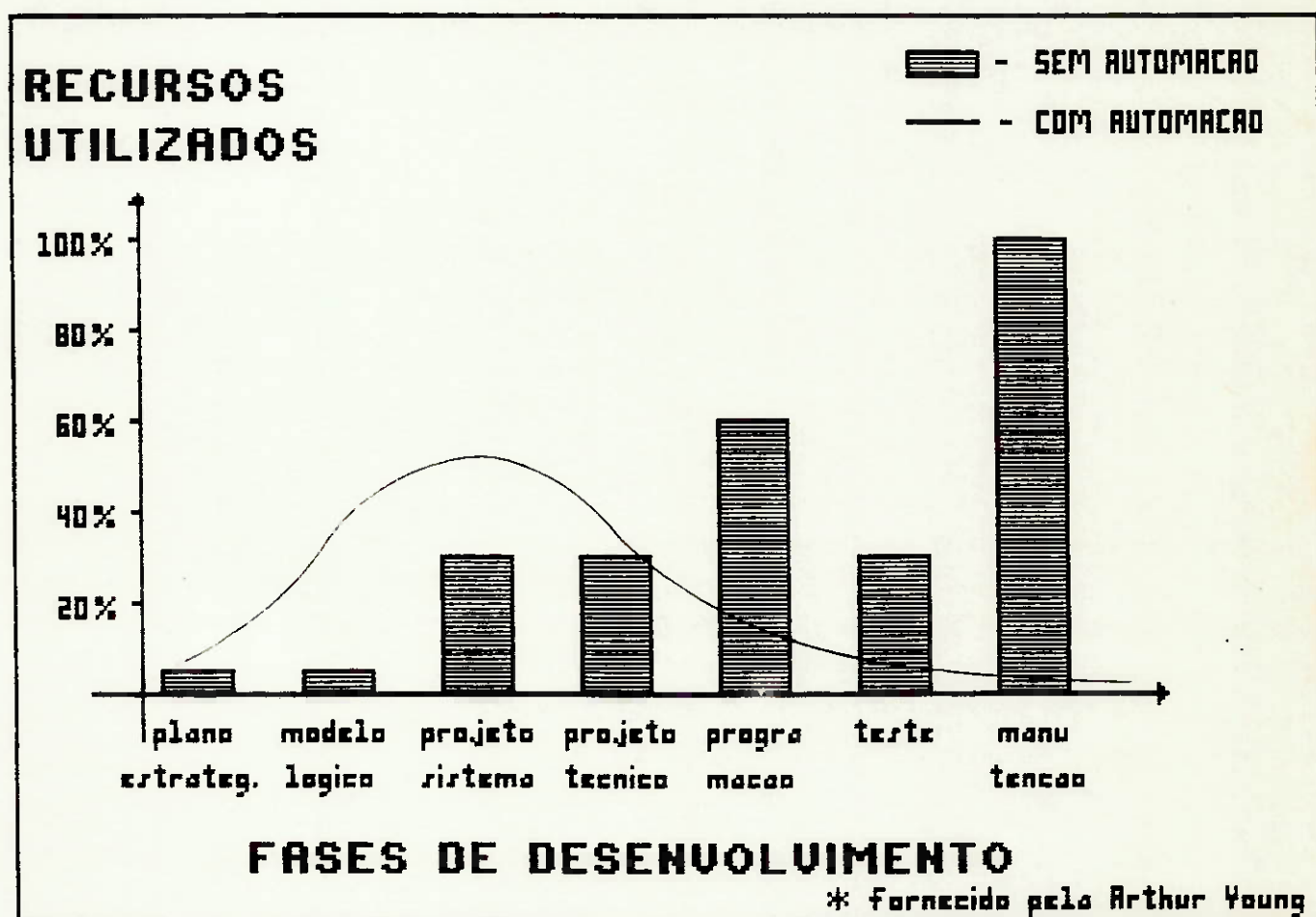
registrados no CASE.

Os benefícios que a utilização de CASE trazem são:

- a. tornar prático o uso das técnicas estruturadas.
- b. fazer cumprir as normas e procedimentos da metodologia adotada na empresa.
- c. melhorar a qualidade do Software desenvolvido, através da verificação automatizada de erros.
- d. tornar prático a adoção de prototipação.
- e. simplificar o processo de manutenção do Software.
- f. acelerar o processo de desenvolvimento do Software.
- g. liberar o analista para se concentrar no trabalho criativo do desenvolvimento.
- h. permitir o desenvolvimento incremental do Software.
- i. reusar os componentes do Software.

O gráfico 6.3.2 mostra uma comparação entre o nível de recursos gastos no desenvolvimento de Sistemas manualmente e com auxílio de um Software tipo CASE.

GRÁFICO 6.3.2 - Comparação entre os recursos necessários para o desenvolvimento de Sistemas sem automação e com automação (CASE).



O sucesso de um Sistema CASE depende de alguns aspectos que devem ser considerados:

- . CASE deve ser um produto que suporte todo o ciclo de vida do Software de forma integrada, ou seja, ele deve suportar desde o planejamento até a implementação e possível manutenção, de tal forma que a saída de uma fase de desenvolvimento seja a entrada da próxima.
- . deve haver uma interface perfeita entre projeto e geração de código, ou melhor, o código completo deve ser gerado a partir do projeto especificado.
- . ele deve ser um produto flexível, de tal forma que qualquer metodologia para desenvolvimento de Sistemas possa ser utilizada. Para isto, é necessário que exista um gerador de diagramas de especificação e um gerador de regras de consistência, sintaxe e validação, ou seja, o meta-sistema.

6.4 Uma Análise Comparativa de Alguns Produtos do Mercado

Neste item pretendemos fazer uma análise comparativa de alguns CASEs disponíveis no mercado nacional. Para tal, escolhemos alguns produtos e assistimos às suas demonstrações.

Nesta análise comparamos alguns pontos destes produtos, importantes para se seleccionar um CASE.

No mercado nacional existem disponíveis no momento os seguintes produtos:

Produtos Disponíveis no Mercado Nacional.

PRODUTO	FORNECEDOR	FUNÇÃO
Design/1	Arthur Andersen	análise e projeto
Excelerator	System Advisers do do Brasil	análise e projeto
IEW - Information Engineering Workbenche	Arthur Young Consultores	análise, projeto e implementação
Mosaico	IESA - TS	projeto
PGSA/PCDFD	Base Tecnologia	análise
Smart	Intertec Serviços	análise e projeto
Analyst/Designer Toolkit	Compucenter	análise e projeto

A seguir apresentamos a análise dos produtos selecionados. No Anexo 3 reproduzimos um exemplo elaborado pelo Prof. Décio Teixeira de especificação de um Sistema pelo CASE SMART.

Análise Comparativa de Alguns CASEs do Mercado Nacional.

ÍTEM	A/D Toolkit	SMART	IEW
Preço do produto básico.	700 OTNs.		1286 OTNs por módulo (total de 3 módulos)
Instalação requerida.	PC-XT,-AT e COMPATÍVEIS	PC-XT,-AT (PC-DOS) e Main Frame (CMS)	PC-AT e MAIN FRAME (MVS)
Metodologia suportada.	Análise estruturada (Yordon) e projeto estruturado.	Análise estruturada.	Análise, Projeto Estruturado e Modelagem de Dados.
Diagramas disponíveis.	Diagrama de contexto, entidade-relacionamento, transição de estado, hierárquico, fluxo de dados, fluxograma e diagramas comerciais	Diagrama de contexto, de fluxo de dados, entidade-relacionamento, de composição, transição e de estrutura (hierárquico).	Diagrama de decomposição, entidade-relacionamento, de fluxo de dados, de ação, de estrutura e matriz de dados x processos.
A ferramenta possui um gerenciador de dados ou de dicionário de dados.	Há um repositório central de dados, cujos dados podem ser manuseados.	Há um repositório central de dados, cujos dados podem ser manuseados.	Há uma enciclopédia de dados, que os contém e respectivos relacionamentos.
Especificação do sistema.	A especificação é feita diretamente no diagrama DFD.	A especificação é feita na forma de texto, através de português estruturado.	A especificação inicial é feita pelos diagramas, na ordem em que se desejar.
As tendências de evolução da ferramenta.	Será aprimorado o repositório central e sua interface com os diagramas.	Será aprimorado a interface homem-máquina e o repositório central.	Adaptação para outros sistemas operacionais e outras metodologias.
Facilidades de navegação entre os elementos da ferramenta.	A navegação existe, mas não é muito eficiente; um diagrama "pai" não chama o "filho".	Ela é feita através dos menus em janelas, mas a navegação dos diagramas não é boa.	A navegação entre os elementos é muito boa, nem se percebe o limite dos diagramas.
A ferramenta cria especificações para sistemas já desenvolvidos, ou seja, dá suporte a sistemas antigos.	Não há nenhum suporte a sistemas já construídos, ou seja, não há nenhum apoio à manutenção.	Este CASE não dá nenhum suporte à manutenção de sistemas previamente desenvolvidos.	Não há nenhum suporte à sistemas já desenvolvidos sem o CASE.
A ferramenta possui interface com outras ferramentas CASE, para que várias metodologias sejam usadas.	Não há nenhuma interface com outros CASEs.	Não há nenhuma interface com outros CASEs.	É possível acoplar outras ferramentas CASE ao IEW.
Número de níveis de detalhamento ("explosão") dos diagramas.	Não há limite de níveis na especificação dos diagramas.	Não há limite de níveis na especificação dos diagramas.	Não há limite de níveis na especificação dos diagramas.

ÍTEM	A/D Toolkit	SMART	IEW
Janelas que mostram simultaneamente diversos diagramas.	Até 7 janelas podem estar abertas simultaneamente, mas a navegação entre elas não é muito ágil	Apenas uma janela pode estar aberta, não é possível mostrar vários diagramas ao mesmo tempo.	Até 6 janelas podem estar abertas simultaneamente, com uma ótima navegação entre elas.
O diagrama corporativo fornece uma boa compreensão ou representação do planejamento da empresa.	Existe um diagrama de contexto que mostra a relação do sistema com o restante da corporação.	Existe um diagrama de contexto que mostra a relação do sistema com o restante da corporação.	Pode-se fazer todo o planejamento estratégico da corporação, mostrando as relações: dados x funções x deptos.
A ferramenta suporta a prototipação de sistemas.	Não existe nenhum suporte específico à prototipação.	Não existe nenhum suporte específico à prototipação.	A ferramenta gera telas e programas, podendo ser construído um protótipo ativo.
Futuramente a ferramenta transformará automaticamente o nível conceitual em nível físico.	Atualmente, não há nenhuma intenção e nem previsão desta automação.	Atualmente, não há nenhuma intenção e nem previsão desta automação.	A ferramenta já faz esta transformação, tanto a nível de dados como de funções.
Verificação de erros.	Há verificação de erros a nível de diagramas e de projeto.	Erros de validação e inconsistências são detectados. Nivelamento e balanceamento são feitos automaticamente.	Todas as verificações de erros são feitas automaticamente a nível de diagrama ou projeto.
Qualidade de emissão de relatórios.	O conteúdo dos relatórios ainda é bem simples, contendo algumas referências cruzadas	Há a emissão de inúmeras referências cruzadas além dos elementos do sistema.	Há a emissão de inúmeras referências cruzadas além dos elementos do sistema.
Geração de Código.	Não há nenhum suporte na geração de código.	Há geração de uma estrutura dos processos, que deve ser trabalhada para funcionar.	Há geração de código em COBOL, CSP, inclusive com comentários.
A ferramenta permite distribuição das responsabilidades de desenvolvimento	Não é possível desenvolver o sistema em partes ou módulos e depois reintegrá-los.	Não é possível desenvolver o sistema em partes ou módulos e depois reintegrá-los.	É possível distribuir o trabalho, desenvolvê-lo isoladamente e depois integrá-lo.
Capacidade de exportar partes de diagramas e de especificações do dicionário de dados.	É possível transferir ou copiar diagramas, especificações e trechos destes.	É possível transferir ou copiar diagramas, especificações e trechos destes.	É possível transferir ou copiar diagramas, especificações, trechos destes e de qualquer texto especificado
Interface com o banco de dados corporativo.	Há comunicação apenas com os dicionários de dados externos através do "Extract".	Há saídas para o dicionário de dados corporativo.	Importações e exportações do dicionário de dados corporativo podem ser feitas.

ÍTEMS	A/D Toolkit	SMART	IEW
Processamento de textos.	Não entende português estruturado (linguagem de especificação de programas), apenas identifica termos entre chaves.	Toda a especificação do sistema por parte do usuário é feita em português estruturado, que é depois processada.	A especificação dos elementos do sistema por parte do projetista é feita em português estruturado.
Administração de Projetos	Não há módulo que implemente funções de administração e planejamento de projetos.	O módulo "METRICS" fornece dados para prever esforço e duração do projeto	Há meios apenas de se estabelecer os módulos que são prioritários no projeto.
Possibilidade de adaptação da Ferramenta a outras metodologias.	Existe o módulo "Rule Tool" que possibilita a definição de outros diagramas e regras.	Existe o módulo "TIE" que permite a definição de apenas outras regras.	Não é possível definir outros diagramas e regras diferentes dos existentes.
Facilidade de manutenção do projeto e do sistema.	Qualquer alteração em uma especificação não é automaticamente refletida nas demais, há uma indicação do local onde deve ser modificado.	Qualquer alteração feita na especificação do sistema é refletida nos diagramas e no repositório central.	Qualquer alteração feita na especificação do sistema é refletida automaticamente nas especificações e nos diagramas.
Integração entre as ferramentas.	A integração entre as ferramentas é pequena, pois uma especificação não gera outras especificações.	Há uma boa integração, pois de uma especificação são gerados automaticamente diversos diagramas.	A integração entre as ferramentas é muito boa, o projetista nem percebe em qual das ferramentas do IEW ele está trabalhando.
Nível de Inteligência associada.	A Inteligência Artificial é utilizada para a verificação de erros.	Linguagem de IA é utilizada para definição de regras de produtividade e planejamento.	IA é utilizada para verificação de erros, definição de regras, construção de diagramas
Observações.	Este CASE é basicamente um diagramador, pois a interface diagramas-reposit. central é muito pequena e não há nenhuma implementação do código.	Este CASE é um pouco mais abrangente pois além de diagramar o sistema ele fornece uma estrutura dos processos, funções e dados do sistema.	Este é um CASE muito abrangente, pois suporta todo o ciclo de vida. Faltam a manutenção de sist. pré-CASE e abertura p/ outras metodologias.

6.5 O Que Está Sendo Feito a nível de Pesquisa no Brasil

6.5.1 Apresentação

Acreditamos ser interessante, darmos uma olhada naquilo que vem sendo desenvolvido em relação a CASE no Brasil, mais especificamente a nível de pesquisa. Existem alguns produtos bem interessantes e até arrojados.

Comentaremos três produtos, sendo um desenvolvido na PUC do Rio de Janeiro, outro no CTI (Centro Tecnológico para Informática) e o terceiro na empresa onde este trabalho foi desenvolvido.

6.5.2 MOSAICO

O Mosaico é um Software nacional, para micro computadores tipo PC, comercializado pela IESA - Tecnologia de Sistemas, que tem sua origem em trabalhos acadêmicos da PUC-RJ. Seu objetivo principal é o de apoiar o projeto, documentação e geração de programas em várias linguagens. Mas, atualmente, seu esforço é maior nas fases de codificação de programas e de testes do programa gerado.

Ele se baseia na metodologia de programação estruturada, empregando os diagramas de Jackson e Warnier na representação dos projetos e programas.

Dentro de suas limitações ele é um CASE integrado, pois executa suas tarefas de forma, que o resultado de uma etapa seja a entrada para a seguinte.

O produto é composto por nove blocos, mas os mais importantes são o editor de estrutura, o editor de texto, a base de Software e os linearizadores correspondentes às linguagens utilizadas.

. editor de estruturas - é onde se cria uma representação gráfica e top-down de um programa ou projeto. Utiliza a representação estruturada de Jackson e dispõe de diversos recursos para a edição e organização das estruturas, que compõe um projeto ou um programa. Apenas a estrutura funcional (descrição das operações ou funções) é representada; não há nenhuma facilidade para a estruturação e respectiva documentação dos dados. No entanto, este bloco permite que sejam elaborados programas estruturados e bem documentados.

. editor de texto - com este bloco são criadas especificações, condições lógicas associadas, definições de dados e outros tipos de textos agregados automaticamente pelo Mosaico a partes específicas da estrutura informada pelo usuário. Os textos devem ser escritos numa linguagem de programação específica, e há verificação de erros de sintaxe.

. base de Software - é o repositório central de informações, que registra as entidades de um projeto de programa, bem como os inter-relacionamentos existentes.

. linearizador - produz o código-fonte para o projeto de uma das linguagens apoiadas pelo Mosaico (Cobol, Fortran, C, Pascal, pseudocódigo Assembler e DbaseIII), a partir das informações passadas pelo editor de estruturas e o editor de textos. Mas esta geração de código é muito restrita, há apenas uma concatenação daquilo que foi especificado na estrutura do programa; por exemplo, no Pascal são gerados apenas begin, end e if, then, else.

Não há qualquer verificação de erros, nem erros de sintaxe e nem de validação, ou de consistência dos diagramas, ou de todo o projeto. Isto torna o processo de correção trabalhoso e, impede que uma estrutura lógica seja reaproveitada para a geração de programas em linguagens diferentes.

O produto gera uma lista dos caminhos de testes interessante, o que auxilia muito o procedimento de elaboração de testes dos programas, de tal forma que todos os pontos serão cobertos pelos testes.

6.5.3 SIPS

O Centro Tecnológico para Informática (CTI), situado em Campinas, iniciou o desenvolvimento do SIPS - Sistemas Integrados para Produção de Software - no Instituto de Automação para uso próprio, na resolução de problemas específicos do desenvolvimento de Sistemas em tempo real para as áreas de controle de processos e automação da manufatura.

Posteriormente, pensou-se em estender o núcleo genérico que foi construído, para atender a outros perfis de Software, de forma a tornar possível sua utilização por outros institutos e empresas.

O SIPS é um Sistema expansível para apoio à produção de Software, cujo objetivo é dar suporte a um desenvolvimento integrado, abrangendo todas as fase do ciclo de vida, através de ferramentas automatizadas que contribuam para um aumento de produtividade e melhor qualidade do Software.

As características principais do SIPS são:

- a. coexistência e integração de metodologias diversas, através da abordagem de meta-sistema. Isto torna o CASE flexível, pois se o usuário não utilizar a mesma metodologia que o SIPS usa, poderá especificar outros diagramas e regras da metodologia por ele utilizada.
- b. representação interna uniforme das informações, baseada no modelo entidade-relacionamento. Isto permite que o projeto do Sistema seja baseado em dados, originando uma estrutura de dados estável.
- c. interface de interação com o usuário padronizada. O que vem a permitir uma padronização dos Sistemas construídos com o SIPS, além de uma maior facilidade de aprendizado.
- d. possibilidade de incorporação gradativa de novas ferramentas.
- e. capacidade para suportar o desenvolvimento de Sistemas em tempo real.
- f. suporte ao gerenciamento de projeto e de configuração.

O SIPS possui como componentes

- a. um gerenciador de base de dados meta,
- b. um interpretador meta (estes dois últimos elementos possibilitam a definição de outros diagramas e regras de validação e consistência),
- c. um gerenciador de base de dados Sistema (dicionário de dados),
- d. rotinas de gerenciamento de memória e disco,
- e. interface homem/máquina,
- f. diversas ferramentas.

As ferramentas que o SIPS possui atualmente são:

- a. inicializador da base de dados.
- b. editor de descrições.
- c. editor gráfico de diagramas de fluxo de dados.
- d. editor de formulários.
- e. gerador de listagem de descrições.
- f. editor gráfico para diagramas entidade-relacionamento.
- g. editor gráfico para redes de Petri.

Como o produto ainda está sendo elaborado, muitas ferramentas ainda estão em fase de projeto ou construção, dentre elas:

- a. editor gráfico de diagramas de controle e fluxo de dados.
- b. simulador de dinâmica.
- c. gerenciador de projetos.
- d. gerenciador de configuração.
- e. gerador de relatórios.
- f. analisador genérico de consistência.

- g. editor gráfico de diagramas hierárquicos de funções.
- h. analisador de redes de Petri.
- i. editor gráfico para diagramas de uso de dados.

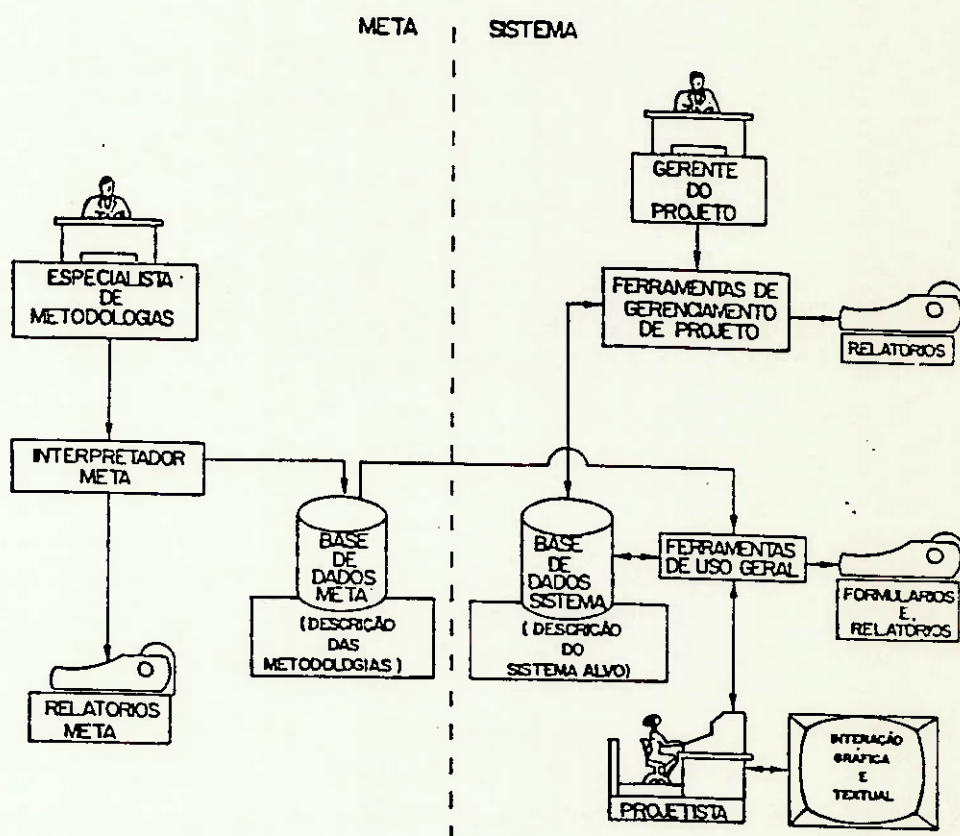
A fig. 6.5.1 ilustra a estrutura do SIPS.

O que o SIPS traz de mais interessante é a visão de META-SISTEMA, onde através do gerenciador de base de dados meta e interpretador meta o usuário pode especificar os diagramas e regras da metodologia por ele utilizada. Isto torna o CASE flexível, porque qualquer metodologia pode ser utilizada em conjunto com o SIPS para o desenvolvimento do Sistema. Poucos produtos possuem esta importante característica, da qual depende o sucesso de um CASE, como mencionamos anteriormente.

FIG. 6.5.1 - Uma Ilustração Esquemática da Estrutura do SIPS.

CENTRO TECNOLÓGICO PARA INFORMÁTICA

SISTEMA INTEGRADO PARA PRODUÇÃO DE SOFTWARE
SIPS



6.5.4 MIND-AID

Este é um CASE elaborado por Yaya Touré [69]. A MIND-AID foi desenvolvida dentro da empresa onde realizamos o estágio.

Esta ferramenta apoia o projetista, durante a concepção de um Sistema, através da metodologia MIND, derivada da Remora, descrita neste trabalho. Mais detalhadamente, este CASE suporta a metodologia Remora no seu nível conceitual. Para tal, ele dá suporte na elaboração do sub-esquema estático e sub-esquema dinâmico (item 5.3.2.2), os quais constituem o esquema conceitual.

Esta ferramenta apresenta as seguintes características principais:

- . Integração da especificação.
- . verificação.
- . documentação.

Existe um repositório central de informações, no caso denominado de metabase, que contém todas as informações do esquema conceitual.

A MIND-AID guia o projetista e faz certas deduções, mas não vai substituí-lo totalmente: o projetista permanece como tomador de decisões durante a análise.

As principais funções executadas pela MIND-AID são:

- . criação do sub-esquema estático - aqui é definida a estrutura dos dados do Sistema, através da definição das entidades ou objetos e respectivos atributos. A ferramenta identifica para o usuário os relacionamentos

existentes

- . documentação do sub-esquema estático - isto é feito através da geração de relatórios, que possuam dados necessários, principalmente, para a conferência do sub-esquema estático e para a elaboração do sub-esquema dinâmico.
- . criação do sub-esquema dinâmico - a MIND-AID auxilia a definição do modelo dinâmico, fazendo com que o usuário defina pelo menos um evento para cada objeto definido no esquema estático (o Sistema os apresenta um a um). Para cada evento definido, o usuário também deve definir as ações disparadas por ele.
- . documentação do esquema conceitual - ela fornece uma série de referências cruzadas, além de um diagrama da dinâmica do Sistema em elaboração.
- . modificação do sub-esquema estático - três tipos de modificações podem ser realizadas: criação ou supressão de um objeto e modificação de um atributo. O CASE efetua as alterações resultantes no sub-esquema estático e dinâmico, originadas pelas modificações realizadas.
- . modificação do sub-esquema dinâmico - possibilita a supressão ou criação de uma ação ou de todo o ciclo da dinâmica (evento, disparo e ação). Todas as alterações no sub-esquema dinâmico, causadas pela modificação realizada, são feitas pelo próprio CASE.
- . integridade da descrição - é possível fazer a verificação de erros de sintaxe (uso correto do modelo de acordo com

suas regras) e de semântica (qualidade da descrição), que o usuário possa ter cometido durante a definição do modelo. O analista é informado dos erros através de relatórios, mas cabe a ele conferir se o modelo elaborado corresponde ou não à realidade descrita.

. **edição de textos** - a MIND-AID utiliza uma linguagem de especificação própria da metodologia MIND, a fim de especificar as operações, suas condições de disparo e as condições de aceitação dos eventos. Esta linguagem propicia um ambiente de edição e de verificação da especificação dos textos citados.

. **implementação dos arquivos** - converte as bases de dados, originadas do sub-esquema estático, em arquivos em DBASE III, já que esta é uma linguagem muito utilizada pela empresa. Assim, o usuário passa a possuir a sua disposição todos os arquivos necessários ao tratamento do Sistema em desenvolvimento.

Este CASE apoia basicamente a fase de análise do Sistema. Em termos de implementação, apenas a parte estática do Sistema é transformada em arquivos, ainda não é possível gerar o seu código a partir do sub-esquema dinâmico. Futuramente, é possível que a MIND-AID seja complementada, a fim de suportar todo o ciclo de vida do Software.

6.6 Tendências

As principais tendências de evolução dos CASEs no mercado nacional e internacional, buscam garantir o sucesso desta tecnologia.

Como mencionamos anteriormente, um CASE para ser bem sucedido deve:

- . ser integrado.
- . automatizar todo o ciclo de vida do Software (especificação, projeto, codificação e manutenção).
- . ser flexível.

Atualmente, a maioria dos CASEs, apesar de serem integrados, suportam apenas a fase de análise do ciclo de vida e não são flexíveis.

As tendências do CASE são:

- . criação de um ambiente altamente iterativo e amigável, permitindo uma boa comunicação entre o produto e o usuário, através do uso de "mouse", janelas, interfaces gráficas, etc..
- . fazer com que possua um gerador automático de código a partir das especificações, de tal forma que seja criado um código quase que diretamente utilizável pela máquina, onde apenas pequenas complementações devam ser feitas pelo programador.
- . ser totalmente flexível, através da existência de um meta-sistema, o qual possibilite a definição por parte do usuário de novos diagramas, suas regras de validação e

sintaxe. Isto é muito importante, pois cada usuário deve utilizar a metodologia que melhor se adapta ao seu ambiente de desenvolvimento.

- . verificação contínua e automatizada da existência de erros de sintaxe e validação nos diagramas ou no projeto todo.
- . será dado cada vez mais ênfase na fase de especificação do projeto, pois quando os erros são identificados nesta fase, mais fácil é de os corrigir, além dos custos serem menores.
- . apoio de Sistemas especialistas, para auxiliarem o usuário na tomada de decisão durante a especificação e projeto do Sistema.
- . integração deve passar a existir entre todas as fases do ciclo de vida. Se um CASE não conseguir repercutir as mudanças ocorridas em todos os documentos e modelos devido a alguma modificação realizada, ele deve pelo menos indicar, quais as mudanças que o usuário deve efetuar.
- . reuso ou reutilização de componentes. Deve existir uma biblioteca contendo as aplicações de Software, funções, rotinas que serão reusadas e reunidas (montadas) na construção de um Sistema; além de um gerenciador de biblioteca, o qual informaria se existe nela um elemento, que execute, parcial ou totalmente, a função especificada no CASE. Assim, na elaboração de um Sistema, 90% de seus elementos já estarão fabricados e armazenados no biblioteca (estoque), bastando apenas construir os 10% restantes e montar todos os elementos. Isto é

equivalente ao que ocorre nas indústrias automobilísticas. Existem diferentes linhas de montagem, e a maioria das peças que estão armazenadas são usadas em todas as linhas, apenas alguns elementos variam entre os modelos de carros, por exemplo o radiador, o painel e outras peças dos automóveis atuais são os mesmos para todos os modelos de uma fábrica.

No CTI está sendo elaborado um projeto, denominado 'Projeto Fábrica de Software (PFS)', que construirá uma biblioteca, seu gerenciador e geradores de código, para possibilitar o reuso. Possivelmente este projeto será integrado ao SIPS, permitindo que este CASE especifique o projeto e o gerenciador de biblioteca do PFS identifique e indique quais dos elementos, que serão necessários à construção do Sistema, já existem armazenados na biblioteca.

O PFS é um projeto que está se iniciando, mas já é reconhecido internacionalmente. Em outros lugares do mundo existem projetos semelhantes, como é o caso dos EUA, Europa, Japão e China.

No próximo capítulo, questionaremos o CASE como a solução da Crise do Software, além de levantarmos outras possíveis soluções.

7 Uma Análise Crítica à Solução Adotada

7.1 Resumo

Este capítulo considera as dificuldades que podem vir a existir na transformação do processo produtivo de "grandes projetos" para "contínuo diferenciado", através da implantação de CASE na Fábrica de Software. Para tal recapitulamos como se deu a mesma mudança na Revolução Industrial.

Para finalizar é questionada a solução adotada para a Fábrica de Software - o Taylorismo

7.2 Consequências da Mudança de Processo Produtivo

Todo o processo de mudança gera problemas que devem ser gerenciados. Recapitulemos como se deu a mudança do processo produtivo artesanal para o processo de produção em massa na Revolução Industrial, pois esta é a mudança proposta na implantação da Fábrica de Software.

Segundo Reich [47], uma das dificuldades encontradas, nos Estados Unidos, na Revolução Industrial, foi a de dirigir pessoas dentro das novas empresas. Era necessário um maior contingente de trabalhadores na mesma localização, cujos esforços precisavam ser coordenados de forma mais exata, o que exigia uma administração cuidadosa.

O processo de mudança para a produção em massa foi ainda mais complicado pela resistência dos artesãos especializados na indústria de transformação, que dominavam o processo produtivo.

Até 1870 esses indivíduos não eram em geral assalariados. Contratavam diretamente seus serviços com os donos das empresas e eram pagos de acordo com o volume que produziam. Os artesãos, em seguida, subempreitavam parte de suas tarefas a outros trabalhadores. Graças a seus conhecimentos e ao controle das operações diárias de manufatura, esses artesãos exerciam enorme poder dentro da empresa. Este poder foi ampliado pela sua organização em sindicatos de ofícios, como o dos Trabalhadores em Ferro, Aço e Estanho.

Os artesãos especializados obstruíram a mudança para a produção em massa. Naturalmente resistiam à introdução de máquinas que tornariam redundantes suas especializações. E, uma vez que eram organizados rigorosamente por ofícios - e não por passos no processo produtivo - não estavam preparados para assumir as tarefas de dirigir, supervisionar e coordenar a fabricação em grande escala.

A primeira grande medida, para abolir o controle exercido pelos artesãos, foi tomada em 1892 na Siderúrgica Homestead. A companhia exigiu que os operários especializados, cujos empregos haviam sido afetados, aceitassem cortes nos salários. O sindicato recusou-se e ameaçou entrar em greve. Os proprietários fecharam a indústria aos artesãos especializados e declararam que a organização era "não-sindicalizada". A greve de Homestead assinalou o início da transição do controle pelos artesãos para o controle pela administração.

Outra medida foi a substituição dos operários especializados por capatazes. A eles foi atribuída responsabilidade total por passos específicos no processo de produção. Enquanto cada um deles coordenasse sua produção com a de outros colegas, tinha liberdade de dirigir os operários como bem lhe aprouvesse. Podia contratar e despedir, promover e rebaixar, ou de outra maneira recompensar ou disciplinar os trabalhadores em seus cargos. Dentro dessa esfera de responsabilidade não era menos autônomo do que haviam sido os artesãos especializados, mas com uma importante diferença: enquanto o controle do artesão sobre seus

subordinados tivera como justificação sua perícia superior, o poder do capataz dependia exclusivamente de sua posição oficial na hierarquia organizacional.

A liberdade absoluta abriu a porta para a supervisão arbitrária e idiossincrática. A impressão dos trabalhadores de que havia favoritismo, preconceito e injustiça na oficina inspirou uma campanha de organização sindical e o irrompimento de greves e paralisações do trabalho, que sacudiram a indústria na primeira década do século XX. Várias empresas tentaram reprimir a inquietação trabalhista, oferecendo uma ampla faixa de benefícios aos trabalhadores.

A substituição de artesãos por capatazes nas fábricas permitira a introdução da produção mecânica em grande escala, mas criara também novos problemas. Carecendo de mecanismos para fiscalizar e orientar as decisões dos capatazes, as recém-consolidadas empresas viram-se, de repente, a braços com resistência maciça à direção arbitrária. Expandindo-se as empresas endureceu-se a resistência. Tornou-se claro, à medida em que se adiantava o novo século, que as emergentes empresas precisavam de métodos mais eficientes e mais legítimos para dirigir suas operações.

Entre 1900 e 1920 a taxa de aumento do produto por trabalhador na economia declinou em aproximadamente 50% em comparação com a taxa que prevalecera na década anterior. A desaceleração assinalou o fim da primeira era da produtividade. O enorme surto de produtividade de fins da década de 1880 e década de 1890 fora devido à mobilização extraordinariamente

eficiente de recursos para a produção em grande escala: maquinaria que incorporava nova tecnologia; trabalhadores disciplinados, oriundos do campo ou do exterior; e progressos nos transportes e comunicações, permitindo que bens e materiais fluíssem ininterruptamente pelo processo manufatureiro e chegassem à distribuição. Os processos subjacentes à onda de mobilização, contudo não podiam sustentar a dinâmica que haviam posto em movimento.

Em suma, os problemas que obstruíram o estágio seguinte da industrialização tiveram origem em falhas de organização.

Outra falha organizacional perseguia a administração pública. Ao governo faltava capacidade para reagir eficiente e equitativamente a muitos dos males sociais da industrialização. Disto surgiu o início da radical clivagem ideológica entre os ideais de bem-estar social e crescimento industrial.

A economia em evolução exigia Sistemas de organização que lhe desse estabilidade e legitimidade. Empresas grandes, recentemente consolidadas, precisavam de estruturas institucionais que lhes confirmassem o novo papel e as ajudassem contra as devastações dos ciclos econômicos. Dentro de suas próprias empresas, os homens de negócio precisavam de um Sistema de regras e políticas a fim de controlar os atos arbitrários de capatazes, legitimar-lhes a autoridade aos olhos dos trabalhadores e ajudá-los a coordenar internamente a produção. E na sociedade em geral a aceitação pelo povo da industrialização dependia de um Sistema de administração pública que lhe pudesse minimizar o impacto social.

Na ausência desses Sistemas de organização, reduziu-se o crescimento contínuo da produção em grande escala. A inquietação trabalhista, as tensões sociais e a agitação política nas primeiras décadas do novo século limitaram a produtividade e colocaram questões fundamentais sobre o papel das grandes empresas.

Da mesma forma como a mudança do processo produtivo com a Revolução Industrial originou uma série de problemas com origens em fatores políticos e sociais, que dificultaram o crescimento da produção, a implantação da automação da produção de Software deve originar muitas dificuldades semelhantes que deverão ser transpostas, a fim de que se consiga um ganho real de produtividade.

Ao introduzirmos CASE na Fábrica de Software temos que nos atentar às possíveis dificuldades que surgirão, para que investimentos volumosos não sejam realizados em vão.

7.3 Dificuldades para a Mudança no Processo de Produção de Software

Apesar de aparentemente CASE resolver todos os problemas genericamente conhecidos como "A Crise do Software" - produtividade e qualidade - vários aspectos deverão ainda ser considerados. Conversamos com o consultor Fernando Moraes da Compucenter a respeito de alguns destes aspectos.

Primeiramente, verifica-se que são muito poucas as empresas que utilizam o produto no Brasil. Mais especificamente, são 34 empresas que adquiriram algum tipo de CASE, sendo que destas apenas duas ou três utilizam efetivamente a ferramenta, como é o caso da Alpargatas e da Ciba Geigy. O número é insignificante, se comparado à demanda existente.

Isto acontece, em parte, porque o CASE ainda é uma tecnologia recente, pouco divulgada. No Brasil, esta divulgação se iniciou em 1987, e apenas alguns poucos profissionais conhecem algo sobre estas ferramentas. Nos EUA a divulgação de ferramentas CASE é maior e começou em 1984, mais empresas utilizam-na, mas este número é ainda pequeno se comparado ao mercado potencial.

Além de ser uma tecnologia recente, algumas dificuldades devem ser superadas para utilizá-la. A ferramenta em si não possui nenhum problema e nem dificuldades de utilização, mas sua implantação gera mudanças no processo de trabalho que devem ser gerenciadas. Apontaremos a seguir algumas dificuldades encontradas na implantação de CASE.

- a. a utilização destes produtos requer que o indivíduo já esteja bem familiarizado com as técnicas estruturadas, ou com a metodologia na qual o CASE se apóia. Se o projetista não estiver habituado à metodologia utilizada, o ganho de produtividade será praticamente nulo ou até negativo. Apesar destas técnicas atualmente serem bastante divulgadas, são muito os profissionais que as utilizam efetivamente.

b. esta ferramenta, assim como qualquer implantação de método, procedimento ou técnica, introduz uma nova forma de trabalho, sendo esta transformação uma mudança significativa.

O ser humano é por natureza avesso às mudanças decididas fora de sua esfera e "inpingidas" a ele. Insegurança, medo ou desconfiança são reações naturais e, muitas vezes, inconscientes.

As mudanças a nível de conhecimentos (através de cursos, palestras, etc.), mais fáceis de instituir por serem mais superficiais, não implicam em mudanças de atitudes e menos ainda em mudanças de comportamento individual, como erroneamente se considera.

A participação em cursos ou a simples existência de normas, manuais ou ferramentas sofisticadas não significam que, necessariamente, as pessoas trabalhem melhor e mais produtivamente.

Para que novas abordagens e tecnologias venham, de fato, a beneficiar quem delas se utiliza (indivíduos e organizações), é fundamental que na sua instituição e operacionalização os aspectos humanos sejam considerados como determinantes essenciais. Sem esta preocupação constante, os riscos de insucesso são elevadíssimos e de alto custo.

Uma forma de lidar com este problema é fazer com que o analista participe do processo de escolha e adoção da ferramenta, ao invés desta lhe ser imposta. Imposição só

funciona quando o chefe está ao lado, depois disto a ferramenta é abandonada. Também deve se fazer uma implantação em pequenos passos, a fim de se desenvolver um especialista interno ou um núcleo de apoio à disseminação da ferramenta; estes profissionais devem ser pessoas abertas e com vontade de aprender. Existindo este núcleo de apoio qualquer dúvida, que os analistas possam ter na utilização da ferramenta, será esclarecida.

Tomando estas medidas, estaremos minimizando a possibilidade de resistência por parte dos analistas (artesãos) em utilizar a ferramenta.

- c. o CASE, juntamente com as metodologias, padroniza o trabalho intelectual, tirando aparentemente a liberdade do projetista de elaborar o Sistema da forma como prefere ou até como ele produz melhor. Esta padronização pode limitar a capacidade criativa do indivíduo, prejudicando os níveis de qualidade resultante do trabalho. No entanto, isto depende da forma como o produto foi implantado, pois isto é um caso particular de resistência à mudança.
- d. o CASE está sendo encarado como o solucionador de todos os problemas da Crise do Software. Realmente é o que aparenta ao analisarmos os problemas e o que ele se propõe. No entanto, não podemos esquecer, que mesmo com a utilização do CASE, é o projetista que faz a conversão do problema real para a sua representação nos modelos

utilizados (diagramas). Nesta conversão o CASE não auxilia o indivíduo e, apesar das técnicas e metodologias auxiliarem muito esta transformação, é ainda aqui onde se encontram as maiores dificuldades.

Apesar destes problemas existirem na implantação de CASE, a maioria deles pode ser solucionada. Esta tecnologia vem auxiliar o desenvolvimento de Sistemas, diminuindo muito o esforço dispendido por parte do analista em desenhar, corrigir, alterar, verificar, codificar e etc..

7.4 A Evolução na Organização do Trabalho após Taylor

7.4.1 Recapitulando Algumas Teorias da Organização do Trabalho

Retomando o paralelo entre a evolução da indústria automobilística e a Fábrica de Software iniciado no cap.6, vejamos as críticas que surgiram a partir da implantação do Taylorismo nas Indústrias. As premissas utilizadas para a transformação do processo produtivo de "grandes projetos" para "contínuo diferenciado" na Fábrica de Software são as mesmas propostas pela Administração Científica.

Segundo Fleury e Vargas [21], a preocupação básica da Administração Científica de Taylor é estruturar totalmente o trabalho, "racionalizando" a tarefa e formalizando as características do setor produtivo, desconsiderando qualquer aspecto do ser humano que não o fisiológico.

Em 1927 foi feita uma pesquisa na Western Electric, onde se detectou que os fatores psicológicos têm grande influência sobre a produtividade, talvez maior que os fatores fisiológicos, contrariando totalmente a Administração Científica, a qual considera os elementos técnicos e físicos como os únicos fatores de aumento de produtividade.

Após esta experiência, na década de 40, surgiram algumas teorias a respeito da motivação no trabalho:

- a. **Teoria da Hierarquia de Necessidades de Maslow** - o ser humano produz melhor quando está lutando por alguma necessidade. Existe uma "hierarquia de necessidades" (necessidades de caráter fisiológico, de segurança, social, de auto-estima e de auto-realização), onde um indivíduo não passa a perseguir as necessidades de nível superior, enquanto não tiver satisfeito as necessidades de nível mais baixo. Assim, o trabalho passa a ser um meio pelo qual as pessoas procuram satisfazer suas necessidades. A Administração Científica apenas considera as necessidades fisiológicas e de segurança. Desta forma o trabalho de Maslow foi um passo para que os psicólogos industriais passassem a se preocupar com as consequências do trabalho sobre a personalidade das pessoas.
- b. **Teoria de Organização e Personalidade** - Argyris caracterizou a personalidade do homem infantil e do homem adulto e concluiu que as organizações de trabalho se fundamentam no modelo do homem imaturo, exigindo

comportamento típicos de personalidade infantil. Conseqüentemente, as organizações não podem ser eficientes, pois os indivíduos experimentarão frustração, problemas psicológicos, perspectivas de curto prazo e conflito. As reações esperadas são que o trabalhador combata a empresa, abandone a organização permanente ou periodicamente, continue na organização mas abandone-a psicologicamente ou aumente a importância das recompensas recebidas, tornando-se orientado para o consumo.

- c. **Teoria de Herzberg** - Herzberg chegou a conclusão que os fatores determinantes da satisfação profissional são diferentes dos fatores que levam à insatisfação profissional. Assim, numa situação de trabalho existem fatores que, se presentes, levam o trabalhador a ficar satisfeito, mas se não estiverem presentes não o levam a ficar insatisfeito. Do mesmo modo, existem fatores que, se não estiverem presentes, levam o trabalhador a insatisfação, mas se estiverem presentes inibem a insatisfação, mas também não conduzem à satisfação.

A solução proposta para a utilização destas teorias é apresentada num conjunto de idéias, que se convencionou chamar de **Enriquecimento de Cargos**. Este é uma ampliação do trabalho de tal forma que isto traga maiores oportunidades, para que os trabalhadores desenvolvam um trabalho que os leve a atingir características de personalidade maduras.

Isto pode ser alcançado através dos seguintes métodos:

- . rotação de cargos - é o revezamento entre as pessoas envolvidas nas tarefas de um processo produtivo; embora cada pessoa tenha de desenvolver várias tarefas, ela só tem uma tarefa para desenvolver por um considerável espaço de tempo.
- . ampliação horizontal - agrupam-se diversas tarefas de mesma natureza num único cargo, com isto aumenta-se o número de habilidades requeridas do operário.
- . ampliação vertical - atribuem-se tarefas de diferentes naturezas para um cargo, com isto existe maior autonomia e controle do operador sobre o conteúdo do cargo.
- . enriquecimento de cargos - a ampliação horizontal e vertical são aplicadas a um único cargo, com isto se somariam os efeitos benéficos das duas.

Observa-se que o esquema de Enriquecimento de Cargos se baseia em duas hipóteses sobre o comportamento do homem no trabalho:

- . a produtividade de uma pessoa é tanto maior quanto mais ela estiver satisfeita.
- . a satisfação é decorrente de fatores intrínsecos ao trabalho.

Esta forma de organização do trabalho continua projetando cargos individuais, sequer cogitando a idéia de grupos, mas ressaltando sempre a preocupação de permitir ao trabalhador contato com outras pessoas e oportunidades para formar amizades. Assim, o Enriquecimento de Cargos deve ser visto como uma espécie de corretivo para as técnicas da Taylorização.

Como fruto da concepção sócio-técnica para a análise das organizações surgiu o esquema de Grupos Semi-Autônomos, o qual coloca que os dois enfoques acima são importantes e não exclusivos.

Um Grupo Semi-Autônomo (GSA) é uma equipe de trabalhadores que executa, cooperativamente, as tarefas que são designadas ao grupo, sem que haja uma predefinição de funções para os membros. O próprio grupo planeja o trabalho e o divide.

Sob o aspecto social admite-se que o ponto mais relevante é a cooperação requerida entre os elementos constituintes do grupo, ou seja, o suporte para o inter-relacionamento entre as pessoas são as relações de trabalho, e não relações espontâneas de amizade como colocam os defensores de Enriquecimento de Cargos. Sob o prisma individual, se requer o desenvolvimento de múltiplas habilidades.

Sob o aspecto técnico o conceito fundamental é o da auto-regulação. Isto decorre por se evitar a formalização de cargos e permite que o Sistema de produção seja flexível. Também, o GSA apresenta uma racionalidade produtiva bem clara, através do rompimento dos limites ao aumento de produtividade de processos de produção em linha, onde o tempo do ciclo é dado pela operação mais lenta; e através da redução da vulnerabilidade do processo produtivo, pois a falta de poucas pessoas pode inviabilizar uma linha, mas não um conjunto de grupos.

A idéia implícita é a de que um grupo formado pelo menor número de pessoas, capaz de desempenhar um trabalho completo e satisfazer as necessidades sociais e psicológicas de seus

membros, é o arranjo mais satisfatório e eficiente, tanto do ponto de vista do desempenho da tarefa quanto do ponto de vista daqueles que estão trabalhando.

Com a crise mundial, segundo Salerno [55], a briga pela manutenção ou ampliação de mercados ganha contornos distintos, onde a diversificação é uma arma para o atendimento dos consumidores. Portanto, as empresas de produção em massa devem estar atentas ao mercado, contando para isso com uma estrutura de produção flexível, com vistas a tender variações da quantidade e responder rapidamente a pedidos ou mudanças no comportamento do mercado.

A fim de se conseguir um Sistema produtivo flexível foi introduzido pelos japoneses o Sistema "Just in Time / Kanban". Na realidade ele pouco altera a estrutura organizacional, mas inova totalmente a forma de produção em série.

"Just in Time" seria produzir o que é necessário, na quantidade necessária e no momento necessário. Em termos de produção significa que, na montagem de um produto, as necessárias submontagens precedentes devem chegar na linha no momento necessário à montagem e na quantidade necessária.

Para tal, quem retira as peças de um posto de trabalho é o operador/departamento subsequente. Isto ocorre pelo simples fato de o subsequente sentir a necessidade das peças. Quem dá a ordem de produção a determinado posto é também o operador subsequente.

O Kanban propriamente dito é um Sistema de informações para administrar o just in time.

7.4.2 Questionando o Taylorismo na Fábrica de Software

Voltemos novamente à Fábrica de Software. As técnicas e metodologias utilizadas no desenvolvimento de Sistemas pressupõem indiretamente a adoção da Administração Científica como forma de organização do trabalho.

Tavares [65] analisa, sob este ponto de vista, as técnicas estruturadas.

Analogamente à noção de método em Taylor, a programação estruturada estabelece uma única forma de gerar programas ótimos, ou seja, guarda o mesmo caráter disciplinador da proposição taylorista.

O "melhor" método (o critério de eficiência da programação estruturada não é o tempo como na Gerência Científica, mas a redução do número de erros, o que em última análise vem a ser o mesmo) é também aquele que tende à uniformização através da limitação de soluções criativas.

O aspecto uniformizador é importante dentro de uma perspectiva de maior parcelamento de tarefas. Ele pretende garantir uma melhor composição de um todo que tenha sido fragmentado e, possivelmente, produzido por um grupo de trabalhadores que não mantenham contato direto durante a sua produção.

Nos primórdios da computação, o domínio da documentação pelo programador (símbolos e formas próprias de registro) gerava uma situação privilegiada deste diante da gerência. Durante o desenvolvimento de um Sistema e mesmo após sua implantação, a

gerência se encontrava atada a determinada equipe de trabalho que, não explicando uma lógica de estruturação e registro de seus programas, mantinha um bom nível de controle sobre o seu trabalho. Da mesma forma, como antes da Revolução Industrial, o artesão possuía o controle total do processo produtivo, pois somente ele dominava o conhecimento técnico, ficando a gerência sob sua dependência.

A esta situação a gerência reagiu estabelecendo normas para a geração de símbolos e impondo formas padronizadas de registro. A questão da legibilidade de programas torna-se mais importante quando se pretende uma maior hierarquização funcional e a implantação de esquemas de supervisão das tarefas.

Todas as demais metodologias e técnicas apresentadas neste trabalho possuem as mesmas características acima mencionadas, ou seja, as características básicas do Taylorismo.

O que questionamos é se esta é a melhor forma de organização do trabalho, a forma mais produtiva para o desenvolvimento de Sistemas. Aqui cabem também os mesmos questionamentos feitos pelas demais teorias apresentadas (item 7.4.1) a respeito da motivação no trabalho.

Será que a Fábrica de Software não passará pela mesma mudança organizacional do trabalho que a indústria de manufatura passou?

Acreditamos que sim, pois com a automação da produção de Software (CASE), o aspecto humano deverá ser levado em consideração para que a implantação daquela tenha sucesso.

O taylorismo considera apenas os aspectos técnicos, então será necessário uma organização do trabalho, que considere os fatores psicológicos como elementos influentes da produtividade. Afinal, foram exatamente estes fatores que originaram as dificuldades surgidas na transformação do processo produtivo durante a Revolução Industrial.

Talvez, a forma de trabalho que melhor se adapte à fabricação de Sistemas seja o Grupo Semi-Autônomo, pois este leva em consideração os aspectos humanos, motivando o indivíduo ao trabalho, permite a divisão do trabalho estabelecido pelas metodologias de desenvolvimento de Sistemas e oferece condições para que CASE seja utilizado como ferramenta de automação do trabalho. Em suma, o GSA permite a utilização dos conceitos que foram propostos para a Fábrica de Software.

Será possível implantar um Sistema flexível na produção de Sistemas?

O "just in time" tem como um de seus objetivos reduzir os estoques de matéria-prima. Estes, na Fábrica de Software, seriam o conhecimento humano, como definimos no capítulo 4. Os programas equivalem, por exemplo, às matrizes de fundição que são utilizadas quando se deseja uma "cópia" do produto anteriormente desenvolvido. Assim não é possível reduzi-los.

Reduzir o conhecimento é equivalente a reduzir os recursos humanos. Desta forma, a empresa utilizaria a sub-contratação quando novos Sistemas entrassem em desenvolvimento. Consideramos isto uma atitude arriscada, pois será difícil contratar profissionais que possuam o nível de formação requerido pelas

técnicas utilizadas na empresa e com isso os Sistemas podem continuar a não atender as necessidades do usuário. Além do mais a sub-contratação pode desmotivar os trabalhadores efetivos da fábrica.

Não podemos nos esquecer que estamos lidando com um trabalho intelectual, o qual possui certas particularidades que impedem a utilização de todos os conceitos envolvidos no trabalho de manufatura. De qualquer forma ficam em aberto estas questões organizacionais.

8 APLICANDO OS CASES DESENVOLVIDOS NA EMPRESA

8.1 Resumo

Neste capítulo, primeiramente, é descrito como se deu o projeto do novo Sistema de Câmbio da empresa, pois ele originou a construção de um CASE - Gerenciador de Dicionário de Dados, elaborado por nós.

As funções deste CASE são detalhadas e uma comparação com o CASE MIND-AID é realizada.

Para ilustrar melhor fizemos uma aplicação dos dois CASEs para uma parte do Sistema de Câmbio. As listagens resultantes desta aplicação são apresentadas no Anexo-4.

Durante esta aplicação algumas dificuldades foram sentidas na utilização da Metodologia Remora, para solucionar estes problemas foi elaborado um método para ser utilizado na construção do modelo dinâmico e foi proposto uma abordagem diferente para a programação dos Sistemas.

8.2 O Projeto do Sistema de Câmbio

Para aplicarmos os CASEs elaborados dentro da empresa, escolhemos o novo Sistema de Câmbio, por termos participado em parte do projeto.

O projeto de elaboração do novo Sistema de Câmbio constou de diversas etapas, algumas baseadas na metodologia Remora:

- a. **construção do modelo estático do Câmbio** - foi feita uma modelagem de dados baseando-se no modelo entidade-relacionamento, como propõe a Remora. Paralelamente foi desenvolvido um Sistema denominado "Gerenciador de Dicionário de Dados" (GDD) para auxiliar a documentação e definição das entidades, atributos e relacionamentos.
- b. **construção do modelo dinâmico do Câmbio** - foi construído uma espécie de modelo dinâmico da Remora, no qual não foram definidas as ações; apenas os objetos, eventos e a sequência em que estes eventos se dão para cada produto.
- c. **validação e alterações dos modelos junto ao usuário** - por ser o câmbio uma área muito complexa, foi necessário que o usuário verificasse os modelos do Sistema. Desta verificação surgiram diversas alterações.

Se o Sistema não fosse tão complexo, poderia se cogitar em fazê-las manualmente. Para se ter uma idéia do tamanho do Sistema, seus modelos contém 150 entidades ou objetos, 778 relacionamentos, 548 atributos, 107 produtos e 287

eventos.

Quando as alterações são feitas, algumas consistências devem ser realizadas. Por exemplo, se uma entidade é excluída devemos verificar se algum de seus atributos não pertence a mais nenhuma outra entidade, para também excluí-lo do dicionário de dados. Seria impraticável fazer as alterações necessárias sem uma ferramenta informática de auxílio. Coube a nós construir esta ferramenta.

Na realidade esta é uma ferramenta que guia o analista na realização das inclusões e atualizações e faz algumas das consistências necessárias. Por estes pontos acreditamos que esta ferramenta pode ser considerada como um CASE. No item 8.3 detalhamos melhor a ferramenta - Gerenciador de Dicionário de Dados.

- d. elaboração de um Plano Diretor de Informática - este plano consta da definição de todos os Sistemas, suas principais funções, sua classificação em tipos, definição dos usuários e o estabelecimento de prioridade de construção.
- e. elaboração do Levantamento de Tempos - este trabalho, por nós executado, teve por objetivo quantificar as variáveis organizacionais, ou seja, ao dividir o trabalho em tarefas, quantificamos e analisamos os tempos gastos para executá-las.

Desta forma, passam a ser conhecidas as ações que ocupam um tempo significativo. Tendo em mãos esses

dados, diversos trabalhos poderão ser realizados (trabalhos organizacionais, de tempos e métodos, de otimização de tarefas, etc.).

Quando iniciamos o estágio as duas primeiras etapas já haviam sido realizadas, por este motivo não acompanhamos a construção dos modelos do Sistema.

8.3 Gerenciador de Dicionário de Dados - o Nosso CASE

A primeira atividade que fizemos ao iniciarmos o estágio foi a construção do Módulo de Atualização do Gerenciador do Dicionário de Dados em Dbase III, que depois foi integrado ao Módulo de Consulta constituindo o "nosso CASE".

Por isso ele é um Sistema simples. Não utiliza inteligência artificial - uma maneira de programar onde a capacidade de raciocínio é transferida ao computador - como a maioria dos CASEs. Mas faz as consistências necessárias para que as atualizações não tornem o Dicionário de Dados - conjunto de arquivos que possuem os elementos do modelo estático - inconsistente e redundante.

Este CASE tem por objetivo permitir a consulta do Dicionário de Dados através de relatórios emitidos, guiar o usuário na inclusão e atualização dos elementos do Dicionário e também refletir a atualização destes nos demais elementos (entidade, atributo, relacionamento entre entidades e relação de atributo com entidade).

As principais funções da ferramenta são:

a. Consulta:

1. lista de todos os relacionamentos - lista todos os relacionamentos entidade-entidade existentes no dic. dados.
2. lista dos relacionamentos entidade-entidade de uma entidade escolhida - fornecendo o nome ou o número de uma entidade, este relatório fornece todas as entidades que estão relacionadas a ela, inclusive o tipo do relacionamento.
3. lista das entidades - este relatório fornece uma lista do nome e do número de todas as entidades existentes no dicionário.
4. lista da definição e dos atributos de uma entidade escolhida - fornecendo o nome ou número de uma entidade, é gerado um relatório contendo a definição da entidade e todos os atributos que ela contém (relacionamento entidade-atributo).
5. lista de todos os atributos e respectiva descrição - este relatório fornece uma relação de todos os atributos que existem no dicionário junto com sua definição.
6. lista dos atributos em ordem alfabética - é uma lista de todos os atributos existentes no dic. dados, com as respectivas descrições, domínio, regra de validação e formato.
7. a descrição de um atributo escolhido - fornece a descrição do atributo cujo nome foi especificado.
8. a que entidades pertence um atributo escolhido - ao

escolher um atributo, o Sistema fornece um relatório contendo todas as entidades a que ele pertence (relacionamento entidade-atributo).

b. Inclusão:

1. **de entidade** - a nova entidade é definida. O Sistema verifica se já existe alguma similar. Se a entidade ou alguma similar ainda não tiver sido cadastrada, então a inclusão é efetuada, senão uma mensagem é enviada. É solicitado pelo Sistema, o relacionamento da nova entidade com as outras entidades já existentes. Na sequência o Sistema pede a definição dos atributos que descrevem a entidade, através da inclusão de relacionamentos entidade-atributo e os atributos são incluídos, se ainda não existirem, no Dicionário de Dados.
2. **de relacionamento entre entidades** - é permitido a inclusão somente do relacionamento entre entidades já definidas.
3. **de relação de atributos com entidades** - pode ser estabelecido o relacionamento entidade-atributo de um atributo já definido, ou pode-se definir um novo atributo e então relacioná-lo com alguma entidade. A ferramenta não permite que um atributo ainda não definido seja relacionado a uma entidade e, também, verifica se o atributo que está sendo definido é semelhante ou igual a algum já existente.

c. Alteração:

1. **dos elementos de definição da entidade** - é possível alterar os elementos que fazem parte da definição de uma entidade.

Esta alteração só é efetuada se a entidade já estiver cadastrada.

2. do nome/número de entidade - é possível apenas alterar o nome ou o número de uma entidade já definida e, também, quando o novo nome e número é diferente dos já existentes. Quando a alteração é feita a ferramenta automaticamente altera o nome da entidade nos relacionamentos entidade-entidade e nos relacionamentos entidade-atributo.
3. dos elementos de definição do atributo - é possível alterar os elementos que fazem parte da definição de um atributo. Esta alteração só é efetuada se o atributo já estiver cadastrado.
4. do nome do Atributo - só é possível alterar o nome de um atributo já definido, quando o novo nome é diferente dos já existentes. A ferramenta efetua esta alteração automaticamente na definição do atributo e nos relacionamentos entidade-atributo.
5. dos elementos de definição do relacionamento entidade-entidade - pode-se alterar o tipo de relacionamento entre entidades já existentes.
6. dos elementos de definição do relacionamento entidade-atributo - é possível alterar os elementos que definem o relacionamento entidade-atributo.

c. exclusão:

1. de entidade - ao excluir uma entidade a ferramenta automaticamente exclui os respectivos relacionamentos

entidade-entidade e entidade-atributo. Se os atributos que foram excluídos do relacionamento entidade-atributo não pertencerem a mais nenhuma outra entidade, então a definição destes atributos também é eliminada pelo Sistema.

2. de relação entidade-atributo - é excluída a relação de determinado atributo com determinada entidade. Se o atributo que foi excluído do relacionamento entidade-atributo não pertencer a mais nenhuma outra entidade, então automaticamente a definição deste atributo também é eliminada; assim como, se a entidade não conter mais nenhum atributo relacionado a ela, então esta entidade e seus relacionamentos entidade-entidade serão automaticamente excluídos.

3. de atributo - são excluídos somente os elementos que constituem a definição do atributo.

Deve ser notado que este CASE pode ser utilizado na modelagem de dados de qualquer Sistema. Um dicionário de dados é um conjunto de arquivos que contêm as entidades e seus elementos de definição, os atributos e seus elementos de definição, o relacionamento entidade-entidade e seu tipo, e o relacionamento entidade-arquivo.

A seguir mostramos as telas de "menu" do Sistema Gerenciador de Dicionário de Dados.

FIG. 8.3.1 - Tela do Menu Principal do Gerenciador de
Dicionário de Dados.

GERENCIADOR DE DICCIONARIO DE DADOS

**** MENU PRINCIPAL ****

OPCOES:

- 1 - CONSULTA
- 2 - INCLUSAO
- 3 - ALTERACAO
- 4 - EXCLUSAO
- 5 - FIM

ESCOLHA SUA OPCAO:

FIG. 8.3.2 - Tela do Módulo de Consultas do Gerenciador de
Dicionário de Dados.

MODULO DE CONSULTAS

=====

1. LISTAR TODOS OS RELACIONAMENTOS
2. DADA UMA ENTIDADE, QUAIS ESTAO RELACIONADAS ?
3. LISTAR AS ENTIDADES EXISTENTES
4. DADA UMA ENTIDADE QUAL E A DESCRICAO E ATRIBUTOS ?
5. LISTAR AS DESCRICOES E ATRIBUTOS DE TODAS AS ENTIDADES
6. LISTAR TODOS OS ATRIBUTOS EM ORDEM ALFABETICA
7. DADO UM ATRIBUTO QUAL E SUA DESCRICAO ?
8. DADO UM ATRIBUTO A QUE ENTIDADES PERTENCE ?
9. ANALISE DA COBERTURA DAS ENTIDADES NO SIST.DO BAMERINDUS
- F. VOLTA AO MENU PRINCIPAL - DICIONARIO DE DADOS

DIGITE SUA OPCAO:

FIG. 8.3.3 - Tela do Módulo de Inclusão do Gerenciador de
Dicionário de Dados.

Caps

**** INCLUSAO ****

- 1 - ENTIDADE
- 2 - RELACIONAMENTO ENTRE ENTIDADES
- 3 - RELACAO DE ATRIBUTOS COM ENTIDADES
- 4 - VOLTAR AO MENU PRINCIPAL

ESCOLHA SUA OPCAO: 0

FIG. 8.3.4 - Tela do Módulo de Alteração do Gerenciador de
Dicionário de Dados.

Caps

ESCOLHA SUA OPCAO

- 1.ALTERACAO DOS CAMPOS DO ARQ. DE ENTIDADES
- 2.ALTERACAO DO NOME/NUMERO DA ENTIDADE
- 3.ALTERACAO DOS CAMPOS DO ARQ. ATRIBUTO
- 4.ALTERACAO DO NOME DO ATRIBUTO NOS ARQ. RELACAO/ATRIBUTO
- 5.ALTERACAO DO RELACIONAMENTO ENTRE ENTIDADES
- 6.ALTERACAO DE CHAVE/DESCRIPTOR NO ARQ. RELACAO
- 7.FIM DE ALTERACAO/VOLTA AO MENU PRINCIPAL

RESPOSTA:

0

FIG. 8.3.5 - Tela do Módulo de Exclusão do Gerenciador de
Dicionário de Dados.

Caps

****EXCLUSAO****

- 1- ENTIDADE
- 2- RELACAO DE ATRIBUTOS COM ENTIDADES
- 3- RELACIONAMENTO ENTRE ENTIDADES
- 4- ATRIBUTO SOMENTE NO ARG. ATRIBUTO
- 5- VOLTAR AO MENU PRINCIPAL

ESCOLHA SUA OPCAO:

8.4 Exemplificando o GDD e o MIND-AID

A fim de que o leitor fique mais familiarizado com as duas ferramentas descritas, tomaremos uma parte do novo Sistema de Câmbio, o módulo de exportação e faremos o seu projeto lógico através da aplicação dos dois CASEs - MIND-AID e GDD.

Não tomaremos todo o Sistema de Câmbio, para sermos mais didáticos, pois o Sistema é muito grande e complexo (como mencionado no item 8.2).

O material mostrado no Anexo-4 é o resultado do projeto, ou seja, são os relatórios que os CASEs emitem depois de construídos os modelos, a fim de que consultas sejam feitas.

Recomendamos ao leitor rever a descrição do Câmbio no item 2.4.5, a fim de facilitar sua compreensão dos relatórios emitidos.

Mostramos a seguir a equivalência entre as entidades escolhidas para a aplicação dos CASEs e os componentes do Câmbio descritos no capítulo 2.

ENTIDADES ESCOLHIDAS	COMPONENTES DO CÂMBIO
Contrato de Câmbio	Operação - Contrato
Carta de Crédito	Operação ME Tipo Compromisso
Carta de Crédito-Negociação	
Adiantamento	Operações MN Empréstimo
Encargos Regime Competência	Contas MN
Embarque ou Cambial	é o conjunto de documentos que comprova o embarque da mercadoria.
Aviso Débito/Crédito Recebido	Movimentação de Contas no Exterior e Contrato Liquidado ME
Liquidação do Contrato de Câmbio no País	Contrato Liquidado MN

8.5 Comparando o Gerenciador com a MIND-AID

Apesar dos dois CASEs basearem-se na metodologia Remora, existem algumas diferenças significativas entre eles.

Tanto o MIND-AID como o Gerenciador de Dicionário de Dados (GDD) trabalham com o modelo estático (MER). Mas o MIND-AID foi construído em linguagem de inteligência artificial, permitindo que sejam feitas consistências a nível das regras dos modelos, o que evita que o analista cometa erros no projeto lógico do Sistema. No GDD este tipo de verificação é superficial.

O MIND-AID e o GDD lidam com o modelo estático de formas diferentes. No GDD o usuário inicia especificando as entidades, seus relacionamentos entidade-entidade, entidade-atributo e os atributos. No MIND-AID supõe-se que o analista já possua em mãos um esboço do modelo entidade-relacionamento, para então especificar no CASE os tipos (definição mais global dos atributos), os atributos, os relacionamentos entidade-atributo, as entidades e os relacionamentos entidade-entidade. A partir dos relacionamentos entidade-atributo o próprio MIND-AID agrupa os atributos em entidades, e indica os prováveis relacionamentos entidade-entidade.

A maneira de especificar o esquema estático pelo MIND-AID não é a forma usual de trabalho dos analistas, que estão mais acostumados com a sequência utilizada no GDD, o que talvez possa gerar dificuldades na utilização do MIND-AID. Para resolver este problema propomos que o analista faça primeiro um esboço do

modelo entidade-relacionamento através do GDD, para então, com base nas listagens geradas pelo GDD, iniciar a especificação do esquema estático através do MIND-AID.

O GDD trabalha apenas com o modelo estático, enquanto que o MIND-AID também suporta a construção, consulta e atualização do modelo e do diagrama dinâmico, fazendo ainda as consistências das regras do modelo.

Elaboramos, além do GDD, um Sistema que permite apenas a consulta de um modelo dinâmico simplificado, pois este não possui as ações, apenas os objetos, eventos e a sequência em que os eventos ocorrem.

Diríamos que o MIND-AID é mais completo e abrangente do que o GDD, pois suporta melhor a construção dos dois modelos, apesar dele ainda possuir problemas de volume de memória exigido. Mas apesar do GDD ainda não tratar o modelo dinâmico, ele auxilia muito o tratamento do modelo estático.

8.6 Facilitando a utilização da Remora

8.6.1 Dificuldades Encontradas na utilização da Remora

A elaboração o Modelo Dinâmico da Remora para parte do Sistema de Câmbio foi facilitada por possuímos a Sequência de Eventos fornecida pelo Modelo Dinâmico Simplificado.

A Sequência de Eventos possui duas colunas: a da esquerda contém os eventos precedentes e a da direita os eventos seguintes. Na realidade um Sistema pode possuir várias Sequências de Eventos.

A Remora nos obriga a pensar em quais modificações de estado dos objetos originam que ações. No entanto acreditamos ser mais natural pensarmos ao contrário, ou seja, quais ações modificam que objetos. Esta última forma de pensar é fornecida na Sequência de Eventos, que é uma abordagem simplificada do modelo dinâmico da Remora. Além disso, a Sequência de Eventos permite que tenhamos em mente o sincronismo dos acontecimentos ou eventos, facilitando nossa compreensão da realidade em estudo.

Para facilitar a construção do Modelo Dinâmico da Remora para qualquer Sistema, elaboramos um método para a conversão da Sequência de Eventos (forma mais fácil de se pensar no problema) para o Modelo Dinâmico.

No entanto, para que esta conversão seja possível, impusemos um mínimo de disciplina na construção da Sequência de Eventos, sem que se perca a simplicidade e o sincronismo.

Este método procura explorar ao máximo as informações contidas na Sequência de Eventos, mas mesmo assim o Modelo Dinâmico oriundo do método deverá ser completado com outras informações exigidas pela Remora.

8.6.2 Aprimorando a Sequência de Eventos

A Sequência de Eventos deverá possuir as seguintes características, a fim de que seja possível sua transformação em Modelo Dinâmico.

1. passará a possuir três colunas. Na coluna da esquerda serão expressos os eventos precedentes (eventos da Remora), na coluna da direita os eventos seguintes (ações da Remora) e na coluna do meio as condições de disparo dos eventos seguintes (ações da Remora). Estas condições são consultas que devem ser realizadas a outras entidades que não às presentes no evento precedente, para que a ação seja disparada. A coluna do meio possui um preenchimento opcional, as demais são obrigatórias.
2. na coluna da esquerda (eventos precedentes ou eventos da Remora) devem ser utilizados substantivos no lugar dos verbos.
3. na coluna da direita (eventos seguintes ou ações da Remora) devem ser utilizados verbos na forma infinitiva. Se isto não for possível, então não é um evento seguinte (ação), mas sim um evento precedente (ex: evento nº 16 do Anexo-4).
4. na coluna da direita só pode aparecer o nome de uma entidade (objeto modificado).
5. utilizar o mesmo radical da palavra para expressar a ação e o evento decorrente, para que as palavras sugiram a

sequência dos eventos. Ex: ação - fechar Contrato de Câmbio, evento - fechamento Contrato de Câmbio.

6. ser fiel aos nomes das entidades e dos atributos do modelo entidade-relacionamento construído, ou seja, utilizar sempre os mesmos nomes definidos no MER.
7. não confundir evento com ação. Um evento é a constatação da mudança de estado de um objeto que implica numa ação.

Depois ou paralelamente à construção da Sequência de Eventos propomos que seja construída a seguinte tabela:

1. a tabela possuirá quatro colunas.
2. na primeira coluna colocar todas as ações descritas na Sequência de Eventos.
3. na segunda coluna colocar para cada ação qual é o objeto afetado.
4. na terceira coluna para cada objeto afetado descrever quais são os atributos afetados.
5. e na quarta coluna descrever para cada atributo se a ação corresponde a uma inclusão ou alteração do atributo.

Estas medidas propostas não tiram a simplicidade da Sequência de Eventos, apenas procuram evitar possíveis distorções dos conceitos envolvidos e facilitar a posterior identificação dos elementos do Modelo Dinâmico, fazendo com que as informações necessárias estejam presentes.

8.6.3 Um Método para Converter a Seq. de Eventos em Modelo Dinâmico

Para a construção do Modelo Dinâmico a partir da Sequência de Eventos estabelecemos as seguintes regras:

1. as entidades presentes na coluna da esquerda (eventos precedentes) são os objetos constatados do modelo dinâmico, ou seja, são os objetos nos quais a constatação de uma mudança de estado dispara uma ação.
2. as entidades presentes na coluna da direita (eventos seguintes) são os objetos afetados do modelo dinâmico, ou seja, são os objetos que sofrem uma mudança de estado devido ao disparo de uma ação.
3. os eventos precedentes, presentes na coluna da esquerda, são as constatações ou eventos no significado correto da Remora.
4. os eventos seguintes, presentes na coluna da direita, são as ações do modelo dinâmico.
5. se entidades forem mencionadas na condição de disparo, então elas são os objetos consultados da condição de disparo, ou seja, são os objetos que serão consultados para saber se a condição de disparo da ação foi aceita ou não.
6. se na coluna da esquerda aparecer o nome de mais de uma entidade, isto sugere a lista dos objetos a serem consultados na condição de aceitação do evento.
7. a tabela proposta no item anterior sugere quais são os atributos de cada objeto, auxiliando na melhora do esboço

do modelo de dados. Também permite o refinamento da Sequência de eventos, sugerindo quais são as condições de disparo das ações, como mostra a tabela abaixo:

```
,-----,
! ação1 ! objeto1 ! atributo1 ! alteração !
!-----!-----!-----!-----!
! ação2 ! objeto1 ! atributo2 ! alteração !
'-----'
```

sugere a seguinte condição de disparo da ação2: que o atributo1 do objeto1 possua um valor "x".

8. utilizar este conjunto de regras para cada sequência de eventos existente no Sistema.

Seguindo estas regras a construção do Modelo Dinâmico fica bem simples, faltando apenas a determinação de algumas outras informações para torná-lo completo de acordo com as exigências da Remora. Desta forma conseguimos a transformação de uma especificação informal (Sequência de Eventos) do Sistema em uma especificação formal (Modelo Dinâmico da Remora), ou seja, conseguimos dar continuidade ao processo de especificação do Sistema, que passará a ser realizado com o seguinte método proposto:

1. construção de um esboço do modelo de dados (MER) através do GDD.
2. estabelecimento das Sequências de Eventos, seguindo as

- características estabelecidas no item 8.6.2.
3. melhora do esboço da modelagem de dados tendo como base a tabela proposta no item 8.6.2 e utilizando o GDD.
 4. detalhamento da modelagem de dados (esquema estático) através do MIND-AID.
 5. transformação de cada Sequência de Eventos em um Modelo Dinâmico, através do método proposto neste item com a utilização do MIND-AID.
 6. definição completa de todos os Modelos Dinâmicos através do MIND-AID.

8.7 Como Tratar Sistemas Complexos e Dinâmicos Como o Câmbio

Ao aplicarmos parte do Sistema de Câmbio nos CASEs sentimos que este é um Sistema complexo, ou melhor, um Sistema que possui muitos eventos, condições de aceitação de eventos, condições de disparo de ações, enfim muitas regras e opções de tratamento.

De forma esquemática esta complexidade equivale a uma árvore de decisão com muitos níveis e com diversas alternativas em cada ramo.

Tratar problemas que possuem muitos níveis de controle na forma de programação tradicional ou procedural é possível, mas é complicado e trabalhoso. Para complementar, o Câmbio é um Sistema que se altera constantemente, pois sempre surgem novas

normas do BACEN, que implicam na necessidade de alteração do Sistema. Como o Câmbio existem muitos outros tipos de Sistemas que sofrem este mesmo tipo de problema.

Acreditamos que a forma de programação tradicional não se adapte bem a estes Sistemas dinâmicos. Vejamos então qual é a abordagem mais adequada.

Michael Treacy propõe a seguinte classificação para a adoção de uma abordagem de tratamento de Sistemas:

,-----,			
! P R O B L E M A !			
!-----!			
! SIMPLS !		! COMPLEXO !	
,-----!-----!			
! ESTRUTURADO !		! PROG. PROCEDURAL !	
		! S.E. !	
!-----!-----!			
! DESESTRUTURADO !		! S.A.D. !	
		! S.A.D.E. !	
,-----,			

S.E. = Sistema especialista.

S.A.D. = Sistema de Apoio à Decisão.

S.A.D.E. = Sistema de Apoio à Decisão Especialista.

O Câmbio se enquadra no caso de S.E. pois é um problema dinâmico, complexo, mas possível de ser estruturado.

Um Sistema Especialista são programas de computador destinados a solucionar problemas num campo específico de conhecimento. Tais programas possuem um desempenho comparável ao dos especialistas humanos do campo, pois ele não busca a solução ótima para um problema, mas fornece ao usuário a melhor resposta, como faria o melhor especialista da época. Um S.E. é composto dos seguintes elementos:

- . base de conhecimento - é a coleção de informações específicas sobre o assunto fornecidas pelo especialista humano. Estas informações são colocadas em qualquer ordem, o que facilita a alimentação e correção. Com as mudanças ocorridas na realidade, esta Base de Conhecimento deve ser revista.
- . base de fatos - são dados próprios à situação presente, por exemplo hipóteses, sintomas apresentados pelo usuário. Estas informações são utilizadas para acionar a base de conhecimento e chegar a um resultado.
- . máquina de inferência - é o programa que constrói o raciocínio a partir da base de conhecimentos, é um conjunto de mecanismos que vão explorar esta base. Ele é independente do assunto.

Vejamos as diferenças básicas entre a programação procedural e a programação lógica utilizada nos S.E.

De maneira geral os algoritmos são compostos de dois elementos: elementos de controle e elementos de regras. Na programação procedural estes dois elementos estão presentes juntos, de tal forma que ao modificar o Sistema, tanto as regras como os controles devem ser alterados. Em Sistemas complexos muitas vezes todo o programa deve ser modificado. Na programação lógica estes dois elementos estão presentes separadamente. A Máquina de Inferência contém os controles e na Base de Conhecimento estão armazenadas as regras. O controle independe das regras, assim qualquer mudança do Sistema equivale apenas à alteração das regras, o que é mais simples de se fazer para Sistemas complexos.

A Remora se adapta muito bem à abordagem de S.E., pois podemos estabelecer o seguinte paralelo:

Máquina de Sistema de Informação	<--->	Máquina de Inferência
Base de Dados	<--->	Base de Fatos
Base de Programa	<--->	Base de Conhecimento
(regras = eventos + ações)		

Qualquer modificação do Sistema é sempre igual a inclusão e/ou exclusão de alguns eventos com as respectivas ações no esquema dinâmico da Remora. A nível de programação lógica isto equivale a inclusão e/ou exclusão das regras da base de conhecimento, o que é fácil de se fazer.

Desta forma a utilização da abordagem de S.E. para a programação de Sistemas complexos ou dinâmicos, facilita muito os problemas de construção e manutenção, pois as regras independem dos elementos de controle, qualquer alteração do Sistema é facilmente refletida no seu programa e, se a metodologia Remora foi utilizada, também nos seus modelos.

Já na programação procedural estas alterações demandam um tempo razoável, e se por exemplo o DFD foi utilizado para a concepção do Sistema, na maioria das vezes ele deve ser quase todo modificado.

9 CONCLUSÃO

Esperamos ter atingido o objetivo a que nos propusemos: que este trabalho seja uma revisão superficial dos conceitos e técnicas empregados na área de Sistemas de Informação que, acompanhado de indicações bibliográficas, pudesse servir como um texto básico para a formação de pessoas que estão se iniciando na área. Abordamos os conceitos e polêmicas básicos relativos à construção de Sistemas, como o Ciclo de Vida do Software, a Crise do Software, as técnicas, ferramentas e metodologias que auxiliam a elaboração de Sistemas, etc.

Buscamos adotar uma abordagem inovadora de Engenharia de Produção. Encaramos o ambiente de desenvolvimento de Sistemas como uma fábrica e o processo de desenvolvimento de Software como um processo de fabricação. Acreditamos que a extensão crítica dos conceitos e técnicas já consagradas na área de manufatura à área de Sistemas de Informação possa contribuir para a solução da denominada Crise do Software, que tanto preocupa as empresas da área de Informática. Basicamente os elementos desta solução são:

- a. estabelecimento de um organograma para a Fábrica de Software, identificando, segregando e operacionalizando funções que hoje estão pouco definidas e difusas na Organização.
- b. transformação do processo produtivo de Software do tipo de "grandes projetos" em uma "produção contínua diferenciada" através:
 - b.1. da divisão do trabalho estabelecida pelos recursos

centrados em metodologias.

b.2. da integração entre as tarefas propiciada pelas metodologias e pelo CASE.

b.3. da padronização do processo de desenvolvimento de Sistema estabelecida pela implementação de CASE.

b.4. da automação do processo produtivo obtido pela utilização de CASE.

c. consideração dos aspectos humanos na implementação de qualquer nova técnica, ferramenta ou metodologia.

d. adoção do Grupo Semi-autônomo como forma de organização do trabalho, adequada ao tipo produtivo proposto.

e. adoção de abordagens distintas na fase de codificação para Sistemas estáticos e Sistemas que constantemente sofrem alterações.

Finalizando, esperamos que este trabalho contribua para a difusão de uma nova cultura na empresa, pois ele procura transmitir a necessidade e a importância da reflexão sobre a utilização de técnicas e ferramentas que suportem o desenvolvimento de Sistemas e sobre a organização deste processo produtivo.

10 BIBLIOGRAFIA

1. BALL, S. Successful implementation of computer-aided software engineering. Gurnee, Illinois, 12th Structured Methods Conference Proceedings, Strutured Techniques Association, agosto, 1987, p. 38-49.
2. BARALDI, Andrea B. A. Macrometodologia de desenvolvimento de sistemas. Proposta inicial de técnicas de construção. São Paulo, BFB, 1987, doc. RID0850F.
3. _____. Metodologia. São Paulo, BFB, 1987, doc. RID0550F.
4. BARTILOTTI, Antônio Sérgio. Operações de câmbio num sistema bancário: um modelo operacional. São Paulo, EPUSP - Departamento de Engenharia de Produção, 1982.
5. BENCI, G.; BODART, H.; CABANES, A. Concepts for the design of a conceptual schema. IFIP Working Conference, Freudenstadt, Alemanha, 1976.
6. BRANDÃO, Ronaldo. Prototipação: atingindo o alvo com o primeiro tiro. Datanews, s.l., janeiro, 1988 p. 16-18.
7. CARLYLE, Ralph E. High cost, lack of standards is slowing pace of CASE. Datamation, s.l., agosto, 1987, p. 23-24.
8. CAMERON, J. JSP and JSD: the Jackson approach to software development. Washington, DC: IEEE Computer Society Press, novembro, 1983.
9. CAUDILL, Ray. Understanding the development life cycle. National Computer Conference, s.l., 1977, p.269-275.
10. CERVENY, Robert P. Why software prototyping works. Datamation, s.l., agosto, 1987, p. 97-103.

11. CHEN, Peter P.-S. The entity-relationship model - Toward a unified view of data. ACM Transactions on Database Systems, s.l., março, 1976.
12. CHURCHMAN, C.W. Introdução à teoria dos sistemas. 2ª edição, São Paulo, Vozes, 1972.
13. CODD, E.F. A relational model of data for large shared data bank. Communications of the ACM, s.l., vol.13, nº 6, 1970.
14. CODD, E.F. Extending the database relational model to capture more meaning. Communications of the ACM, s.l., vol.4, nº 4, 1979.
15. COLANGELO, Lucio. Fatores críticos de sucesso e os novos desafios no planejamento e administração da informática. Anais do XVIII Congresso Nacional de Informática, São Paulo, SUCESU, 1985, p. 88-93.
16. DA SILVA, Aloisio M.; NISKIER, Celso. O uso de protótipos rápidos no desenvolvimento. Anais do XVIII Congresso Nacional de Informática, São Paulo, SUCESU, 1985, p. 766-771.
17. DeMARCO, T. Structured Analysis and System specification. Yourdon, Nova York, 1978.
18. DeREMÉR, F.; KRON, H.H. Programming-in-the-large versus programing-in-the-small. IEEE Transactions on Software Engineering, s.l., Junho, 1976, p. 80-86.
19. DIAZ, Rubén P.; FREEMAN, Peter. Classifying software for reusability. IEEE Software. Los Alamitos, Califórnia, Janeiro, 1987, p. 6-16.

20. FAULHABER, Henrique. Do artesanato à engenharia ... INFO, s.l., fevereiro, 1987, p. 42-43.
21. FLEURY, A.; VARGAS, N. Organização do trabalho. São Paulo, Atlas, 1983.
22. GANE, C.; SARSON, T. Análise Estruturada de sistemas. Rio de Janeiro, Livros Técnicos e Científicos, 1983.
23. GIBSON, Michael. A guide to selecting CASE tools. Datamation, s.l., julho, 1987, p. 65-66.
24. GUZZO, Henrique L.D. IAD - Interaction diagrams. Campinas, CTI/DDI/GP4, fevereiro, 1985.
25. HUFFENBAECHER, Lício Antonio. Sistema de auxílio à tomada de decisão na área cambial. São Paulo, EPUSP - Departamento de Engenharia de Produção, 1984.
26. IFIP WG 8.1. Information systems design methodologies: a comparative review. 3ª edição, Amsterdã, Elsevier Science, 1987.
27. INTERTEC. SMART para o projeto automatizado de sistemas de aplicação. São Paulo, 1987.
28. JACKSON, M.A. Principles of Program Design. Nova York, Academic, 1975.
29. JONES, Russel. Action diagrams for system analysis. Data Processing, s.l., maio, 1986.
30. KACUTA, Luiz Y. Modelagem de Dados. S.l., 1987.
31. KNOWLEDGEWARE INC. IEW, the planning workstation. Atlanta, GA.
32. KNOWLEDGEWARE INC. IEW, the analysis workstation. Atlanta, GA.

33. KNOWLEDGEWARE INC. IEW, the design workstation. Atlanta, GA.
34. KOONTZ, H.; O'DONNELL, G. Princípios de administração. Uma análise das funções administrativas. 9ª edição, São Paulo, Pioneira, 1974.
35. LUCENA, Carlos. Automação do processo de desenvolvimento de software. Inteligência Artificial e Engenharia de Software. Rio de Janeiro, PUC-RJ, 1987, cap. 3.
36. MARTIN, J. Manifesto de James Martin - presente e futuro da informática. São Paulo, Compucenter, 1983. Compucenter, 1983.
37. MAUTINHO, Fernando M. Pioneiro na área de CASE. PC Mundo, s.l., fevereiro, 1987, p. 3-4.
38. MCCLURE, C. Software reusability and CASE technology. The CASE Report, Southfield, Michigan, outubro, 1987.
39. MCNURLIN, Barbara Canning. Improving large application development. I/S Analyzer, Bethesda, Maryland, março, 1988, p. 1-12.
40. METHOD/1: metodologia de sistema de informações - material de treinamento. São Paulo, Arthur Andersen, 1985.
41. MORAIS, Fernando Luis. Ferramentas CASE para analistas e programadores. São Paulo, Plena Consultoria, 1987.
42. MYERS, Edith D. In search of CASE. Datamation, s.l., março, 1987.
43. PARNAS, D.L. On a buzzword hierarchical structure. IFIP Congresso, Estocolmo, 1974.

44. PARNAS, D.L. On the criteria to be used in decomposing systems into modules. Communications of the ACM, dezembro, 1972, p.1053-1058.
45. PRESSMAN, Roger S. Software engineering, a practitioner's approach. Nova York, McGraw-Hill, 1987.
46. PROIX, C.; ROLLAND, C. A knowledge base for information system design. Paris, Universidade de Paris.
47. REICH, Robert D. A próxima fronteira americana. Rio de Janeiro, Record, 1987.
48. RELVAS, Glaucio C. L.; ALVES, Carlos C. V. Modelagem de dados: evolução e tendências. MIS - O Relatório de Gerenciamento de Informação, s.l., nº8.
49. ROLLAND, C. A methodology for information system design. National Computer Conference, s.l., 1981, p. 583-588.
50. ROLLAND, C.; FOUCAUT, D.; BENCI, G. Conception des systemes d'information: la méthode REMORA. Paris, Eyrolles, 1988.
51. ROLLAND, Colette; RICHARD, Christian. The Remora methodology for information systems design and management. IFIP Working Conference, Holanda, 1982, p. 369-426.
52. ROLLAND, C.; RICHARD, C.; LEIFERT, S. A proposal for information systems design and management. Paris, Universidade de Paris, 1983.
53. ROSS, Douglas T. Structured Analysis (SA): a language for communicating ideas. IEEE Transactions on Software Engineering, s.l., janeiro, 1977.

54. ROSS, D.T.; SCHOMAN, K.E. Structured Analysis for requirements definition. IEEE Transactions on Software Engineering, s.l., Janeiro, 1977.
55. SALERNO, Mario S. Produção, trabalho e participação: CCO e KANBAN numa nova imigração japonesa. Rio de Janeiro, tese de mestrado junto a COPPE/UFRJ, 1984.
56. SHEMER, Itzhak. Systems analysis: a systemic analysis of a conceptual model. Communications of the ACM, s.l., junho, 1987, p. 506-512.
57. SILVA, Ysmar V. CASE: automação para desenvolver pacotes. Datanews, s.l., dezembro, 1987, p. 32-35.
58. SIPS - sistema integrado para produção de software. Campinas, Centro Tecnológico para Informática, setembro, 1987.
59. SMITH, Peter M. Um estudo de caso de prototipação. MIS - O Relatório de Gerenciamento de Informação, s.l., nº6.
60. SOMMERVILLE, Ian. Software engineering. 2ª edição, Workingkam, Inglaterra, Addison-Wesley, 1985.
61. SPITZEN, J.M.; LEVITT, K.; ROBINSON, L. An example of hierarchical design and example. Communications of the ACM, dezembro, 1978, p. 1064-1075.
62. STAMPS, David. Case: cranking out productivity. Datamation, s.l., julho, 1987, p. 55-58.
63. STAY, J.F. HIPO and integrated program design. IBM Syst. J., vol.15, nº 2, 1976.

64. TADROSS, Shafik Tadross. Tendências da informática e seu impacto no desenvolvimento de sistemas... Anais do XVIII Congresso Nacional de Informática, São Paulo, SUCESU, 1985, p.32-47.
65. TAVARES, Silvio R. S. Da crise do software ao projeto estruturado: a submissão do trabalho em programação. Rio de Janeiro, tese de mestrado junto a COPPE/UFRJ, 1979.
66. TEIXEIRA, Décio. SMART - um exemplo de modelagem. Campinas, FG&A, 1987.
67. THE emergence of CASE building a better mousetrap. IEEE Software. Los Alamitos, Califórnia, março, 1988.
68. TOURÉ, Yaya. Descrição da ferramenta de apoio à metodologia MIND-BFB. São Paulo, BFB, 1987, doc. YAY002R.
69. ----- Metodologia MIND-BFB: proposta de uma ferramenta. São Carlos, ICMSC-USP, dissertação para mestrado em ciências de computação, 1988.
70. VON STAA, Arndt. Critérios para avaliação de CASE. Dados e Ideias, s.l., abril, 1988, p. 22-26.
71. YAU, Stephen S.; TSAI, Jeffery J.P. A survey of Software design techniques. IEEE Transactions on Software Engineering, s.l. vol. SE-12, junho, 1986, p. 713-721.
72. YOURDON, E.; CONSTANTINE, L.L. Structure Design. Englewood Cliffs, NJ, Practice Hall, 1979.
73. YOURDON, INC. The Yourdon analyst/designer toolkit. Nova York, Nova York.
74. WARNIER, J.D. Logical construction of programs. Nova York, Van Nostrand Reinhold, 1976.

75. WEISMAN, Randy. Six Steps to AI-based functional prototyping.

Datamation, s.l., agosto, 1987, p. 71-72.

ANEXO 1

Questionário para Avaliação de Metodologias.

Neste anexo, apresentamos parte de um questionário elaborado pela IFIP, uma associação de reconhecimento internacional, para avaliar e comparar metodologias de desenvolvimento de sistemas.

REVISÃO DAS CARACTERÍSTICAS

Questões para determinar as características gerais da metodologia.

1. Análise de dados e processos.

1.1. Em quais das cinco categorias se enquadra a metodologia (assinale uma alternativa).

- a. pura análise de processos.
- b. análise de processos com um pouco de análise de dados.
- c. proporções iguais.
- d. análise de dados com um pouco de análise de processos.
- e. pura análise de dados.

1.2. Em caso de escolha de "b" ou "d", qual é sua estimativa da porcentagem de análise de processos existente na metodologia.

2. Liste os aspectos da metodologia apresentada que não podem ser classificados como análise de processos ou dados.

3. Detalhes da atividade de projeto.

3.1. A metodologia divide a atividade de projeto em fases, as quais são claramente definidas e utilizadas numa determinada sequência ? (responda sim ou não)

3.2. Se sim, liste as fases utilizando a sequência e a

terminologia empregada.

4. A metodologia como descrita é implementável em computador ?

5. Fatores únicos e não-únicos.

5.1. A metodologia é única se comparada com as conhecidas pelo avaliador (não somente as submetidas nesta análise) ? Se sim, explique porquê.

5.2. Se a metodologia for similar a uma ou mais conhecidas, liste estas dizendo no que elas se assemelham.

5.3. A metodologia é uma combinação de outras já conhecidas ? Se sim, quais ?

6. Aplicação.

6.1. A metodologia é aplicável a uma larga gama de sistemas de informação?

6.2. Se não, a que tipos de sistema acredita-se que ela não seja aplicável.

7. Comercialização ou recursos.

7.1. A metodologia aparenta ser comercialmente viável ?

ANEXO 2

Um Exemplo de Aplicação da Metodologia Remora e

Um Esquema do Método da Metodologia Method/1

Neste anexo é apresentado um exemplo de aplicação da metodologia Remora, o qual foi elaborado por uma equipe de desenvolvimento de sistemas na empresa em estudo.

Também é reproduzido o esquema do método proposto na metodologia Method/1 para as suas quatro fases do desenvolvimento de software: planejamento de informações, projeto preliminar de sistema, instalação de sistema e suporte de sistema em produção.

Um Exemplo de Aplicação da Metodologia Remora .

DESCRICAO DO SISTEMA

I	I NOME DO SISTEMA : Sistema de Gerencia de Fundos Indexados (FINGER)	I
I	I DESCRICAO :	I
I	I O sistema tem por objetivo efetuar a gestao de fundos (CLUBE) . Cada clube e' caracterizado por um con -	I
I	I Junto de cotistas (CLIENTE) que dispoe de um saldo (CONTA-CLUBE) de cotas (COTA). Cada clube e' gerido por uma	I
I	I EMPRESA.	I
I	I E' tratada apenas a parte de movimentacao das cotas, nao se preocupando com o controle dos ativos de cada	I
I	I clube.	I
I	I As cotas de um clube sao fiscais, movimentadas por operacoes de tipo contrato (OPERACAO CERTIFICADO).	I
I	I Diariamente cada clube tem sua cota atualizada. Os clientes podem se identificar ou nao (portador).	I
I	I	I

FIGURA 1 - Descricao do sistema

DIAGRAMA DE RELACIONAMENTOS

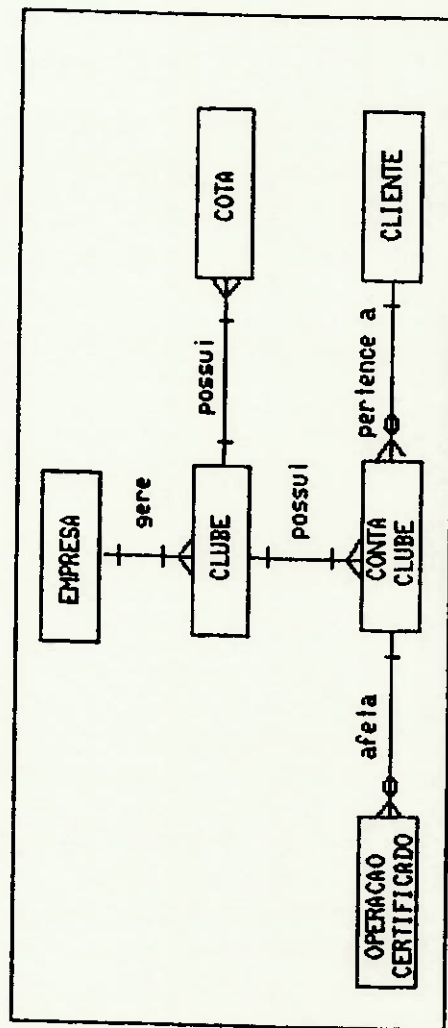


FIGURA 2 - DIAGRAMA DE RELACIONAMENTOS

NOME-TIPO	I	ESTRUTURA	I	ELEMENTOS	I	DOMINIO	I	COMENTARIOS	I
NUM-CONTA-CLUBE	I	registro	I	NUM-CLUBE	I	teste-validacao verifi-	I		I
	I		I		I	ca digito (NUM-CONTA-	I		I
	I		I		I	CLUBE,14,2,1,14,11) e	I		I
	I		I		I	I existe CLIENTE tg NUM-	I		I
	I		I		I	CLI = NUM-CLI e existe	I		I
	I		I		I	CLUBE tg NUM-CLUBE =	I		I
	I		I		I	NUM-CLUBE	I		I
	I		I	TIPO-CONTA	I		I		I
	I		I	NUM-CLI	I		I		I
	I		I	SUB-CONTA-CLUBE	I		I		I
	I		I	DIGITO	I		I		I
TIPO-CONTA	I	simples	I	I1	I	intervalo [1..2]	I	2 => cotas escriturais	I
SUB-CONTA-CLUBE	I	simples	I	I2	I		I	1 => cotas fiscais	I
DIGITO	I	simples	I	I1	I		I		I
NUM-CLI	I	registro	I	BF8-CLI	I		I		I
	I		I	COD-CLI	I		I		I
	I		I	I1	I	intervalo [1..2]	I	1 => usa no. SFFB	I
	I		I	I6	I		I	2 => usa no. qualquer	I
TOT-COTA-ACUM	I	simples	I	N10,4	I		I		I
NUM-CC	I	registro	I	AGENCIA	I		I		I
	I		I	NUM-CONTA	I		I		I
	I		I	DIGITO	I		I		I
	I		I	I3	I		I		I
AGENCIA	I	simples	I	I5	I	tabela AGENCIA	I		I
NUM-CONTA	I	simples	I	N12,2	I		I		I
TOT-VALOR	I	simples	I	I4,6	I		I		I
VALOR-COTA	I	simples	I		I		I		I

FIGURA 4 - Tipo de Dados

DIAGRAMA DA DINAMICA

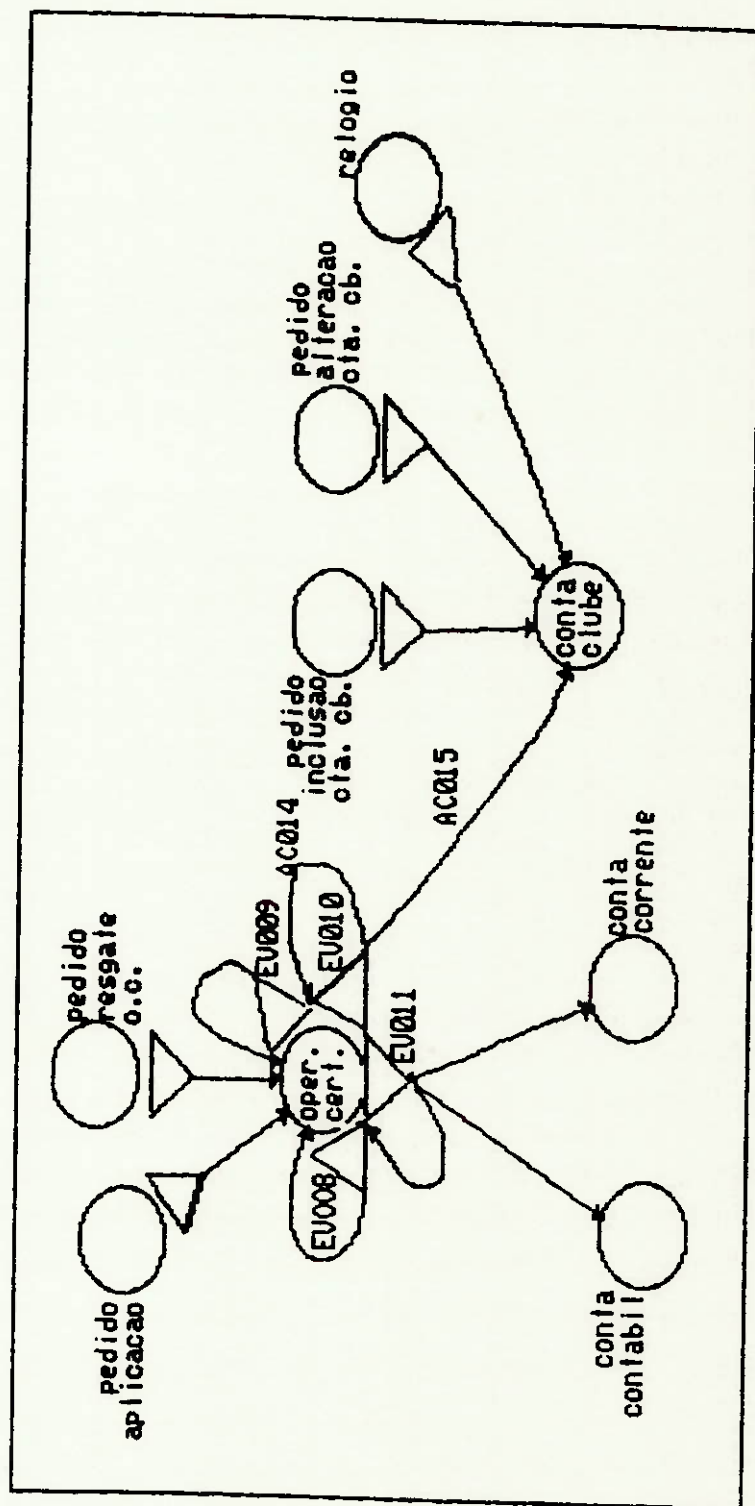


FIGURA 5 - DIAGRAMA DA DINAMICA

OBJETO-EVENTO				
NOME-OBETO	OPERACAO	CERTIFICADO		
COD-EVENTO	NOME-EVENTO	I	CONDICAO DE ACEPTACAO	COMENTARIOS
EV008	Coleta OC	I	est-oper = 'CO' ou 'RS'	I aplicacao ou resgate coletado
EV009	Conversao OC	I	(est-oper = 'IA' e data-conv-aplic (= data-do-dia) ou (est-oper = 'IR' e data-conv-resg (= data-do-dia)	I dia de conversao para o valor em cotas
EV010	Disponibilizacao OC	I	(est-oper = 'AV' ou 'AM' e data-dispon-aplic (= data-do-dia) ou (est-oper = 'RV' ou 'RM' e data-dispon-resg (= data-do-dia)	I dia de disponibilizacao, ou seja, de efetuar a repercussao na conta-clube
EV011	Efetivacao OC	I	(est-oper = 'RD' ou 'RC' e data-efetiv-resg (= data-do-dia) ou (sentido-oper = 'A' e data-efetiv-aplic = data-do-dia ou (data-efetiv-aplic (= data-do-dia e data-coleta = data-do-dia)	I dia de efetivacao, ou seja, de efetuar repercussao na conta corrente e na contabilidade

FIGURA 6 - Objeto - Evento

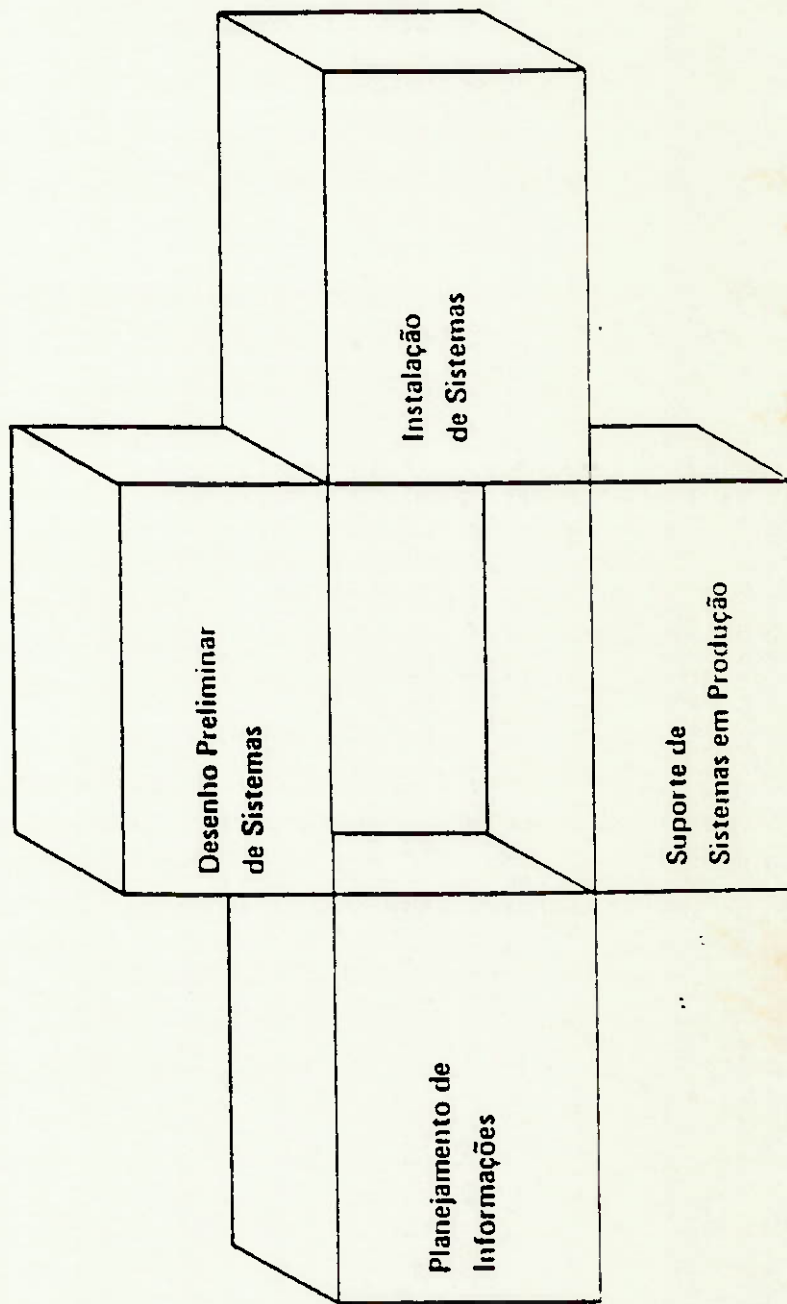


Um Esquema do Método da Metodologia Method/1.

Enfoque Global de Planejamento
Tópico 110
CP- 110
Anexo 4

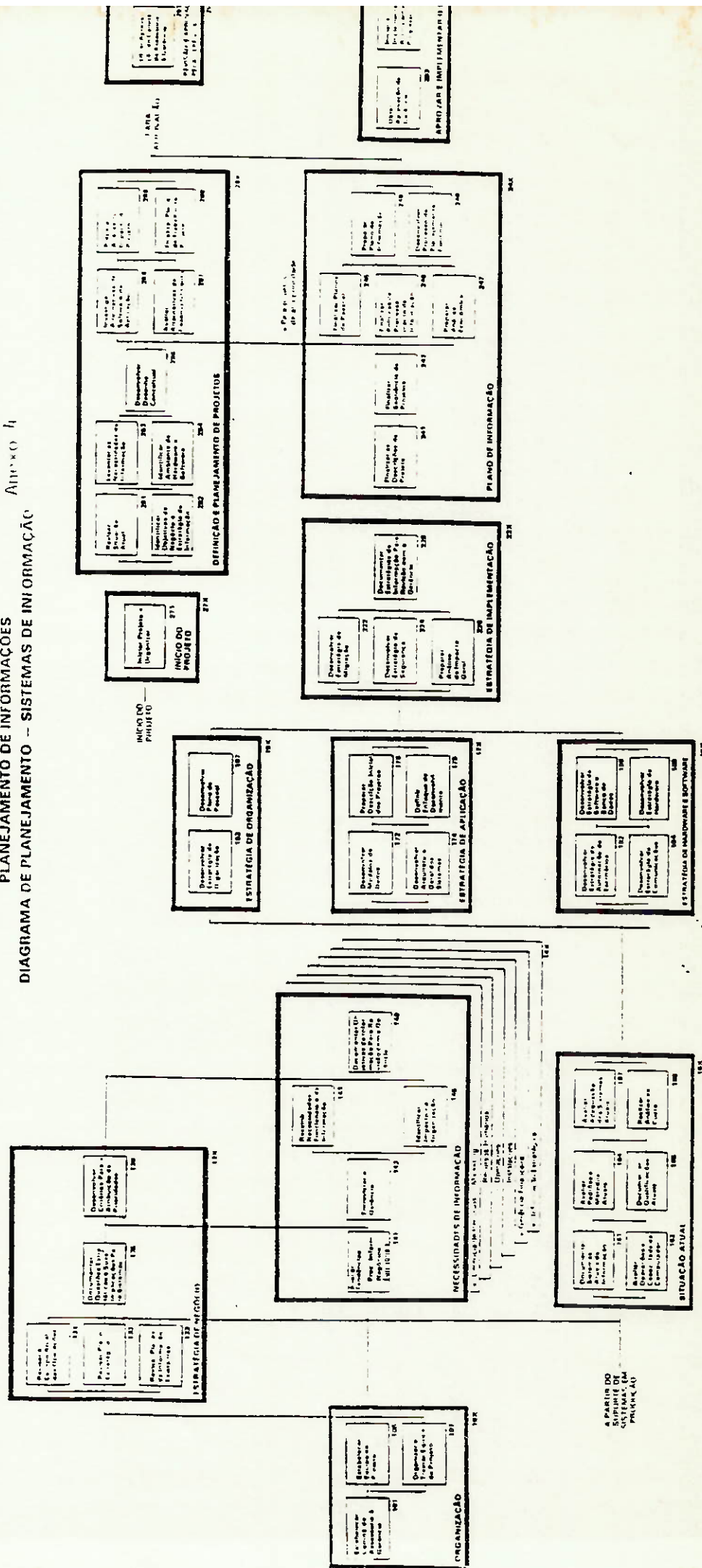
QUADRO OXX-1

ENFOQUE GLOBAL
DIAGRAMA DE PLANEJAMENTO – SISTEMAS DE INFORMAÇÃO



Tratamento do RH 110
Tópico 110
GP- 110
Anexo 4

PLANEJAMENTO DE INFORMAÇÕES DIAGRAMA DE PLANEJAMENTO - SISTEMAS DE INFORMAÇÃO



FORNO DE CONTROLE DE QUALIDADE

FORNO DE CONTROLE DE QUALIDADE

FORNO DE CONTROLE DE QUALIDADE

DESENHO PRELIMINAR DE SISTEMAS
DIAGRAMA DE PLANEJAMENTO — SISTEMAS DE INFORMAÇÃO

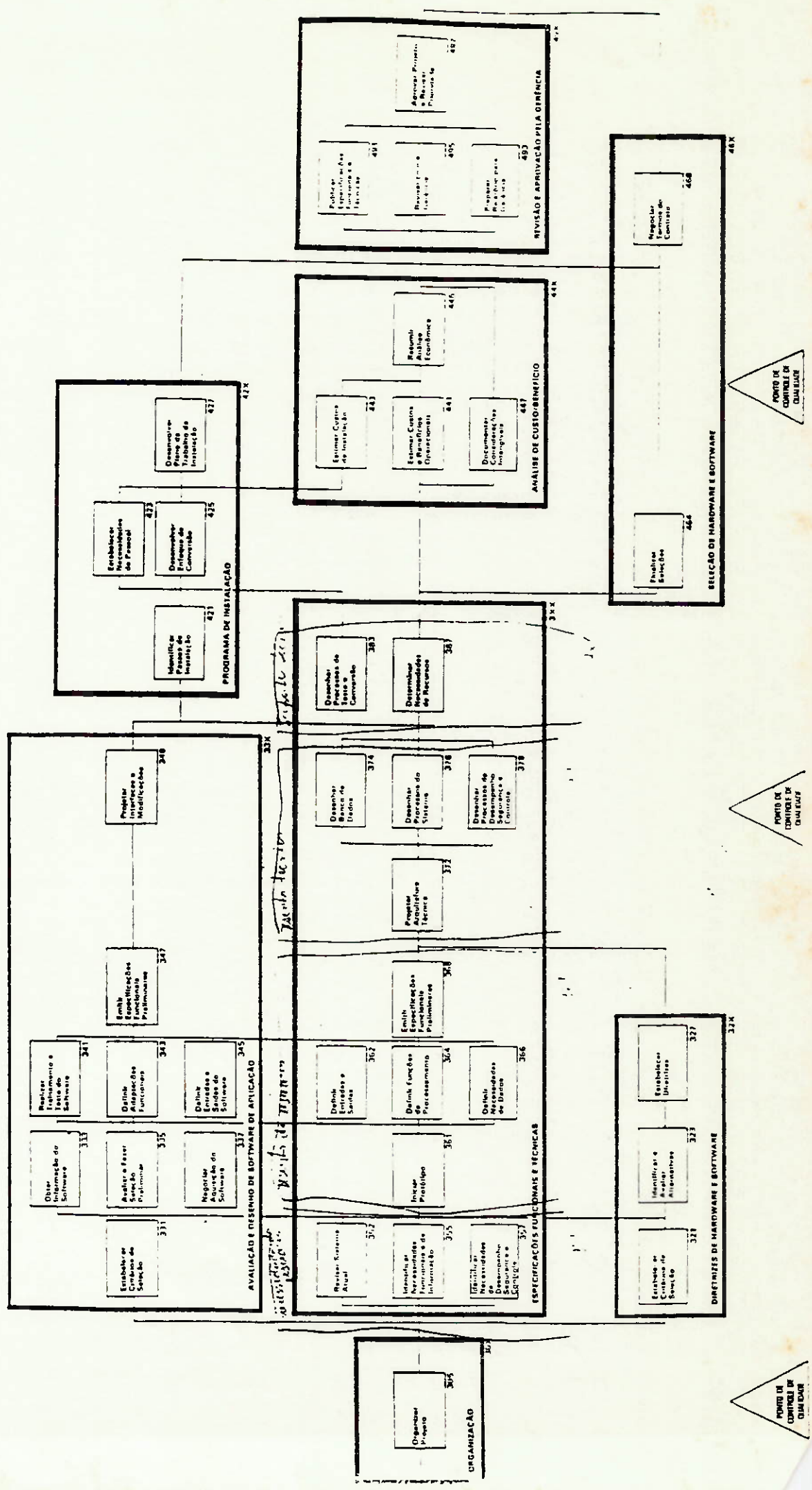
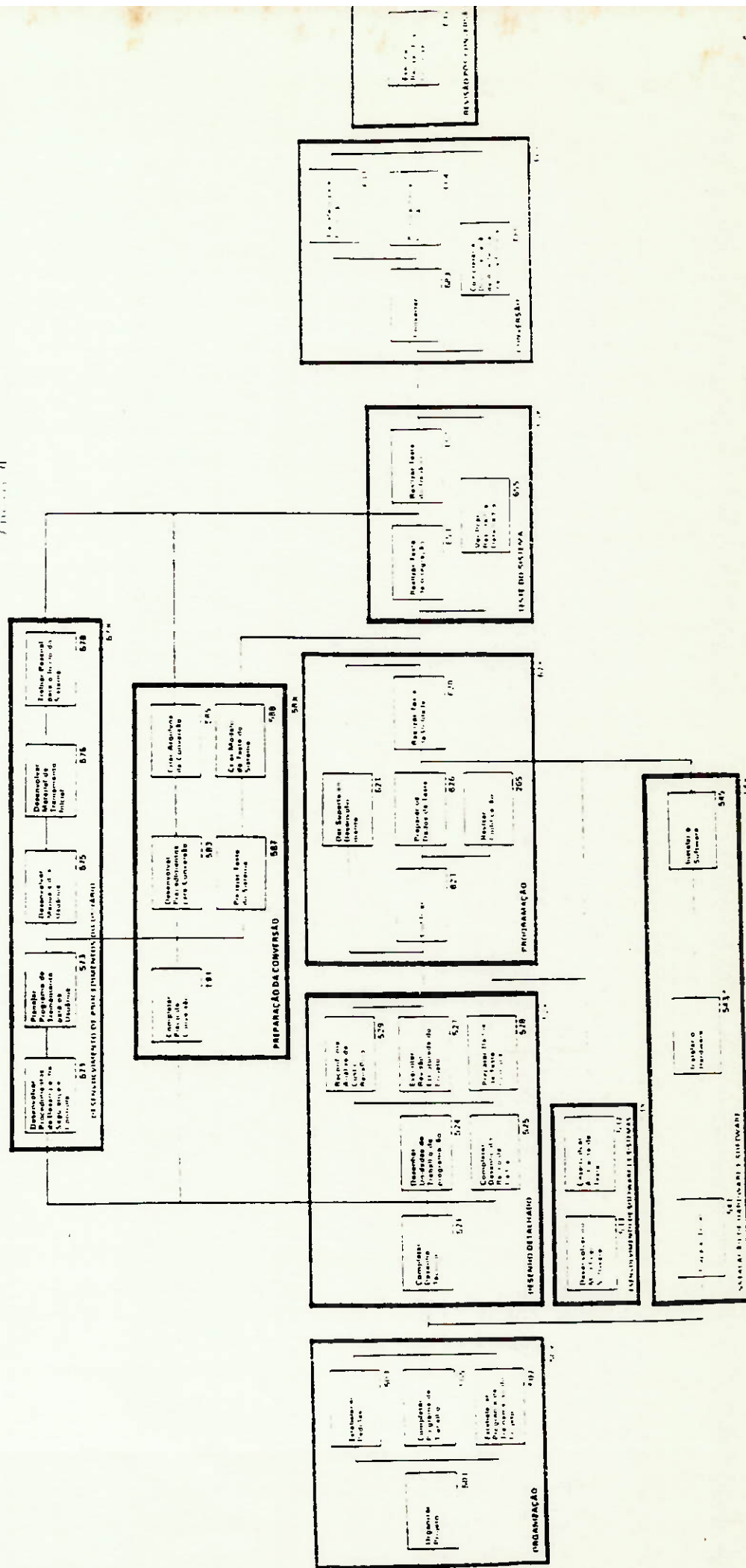
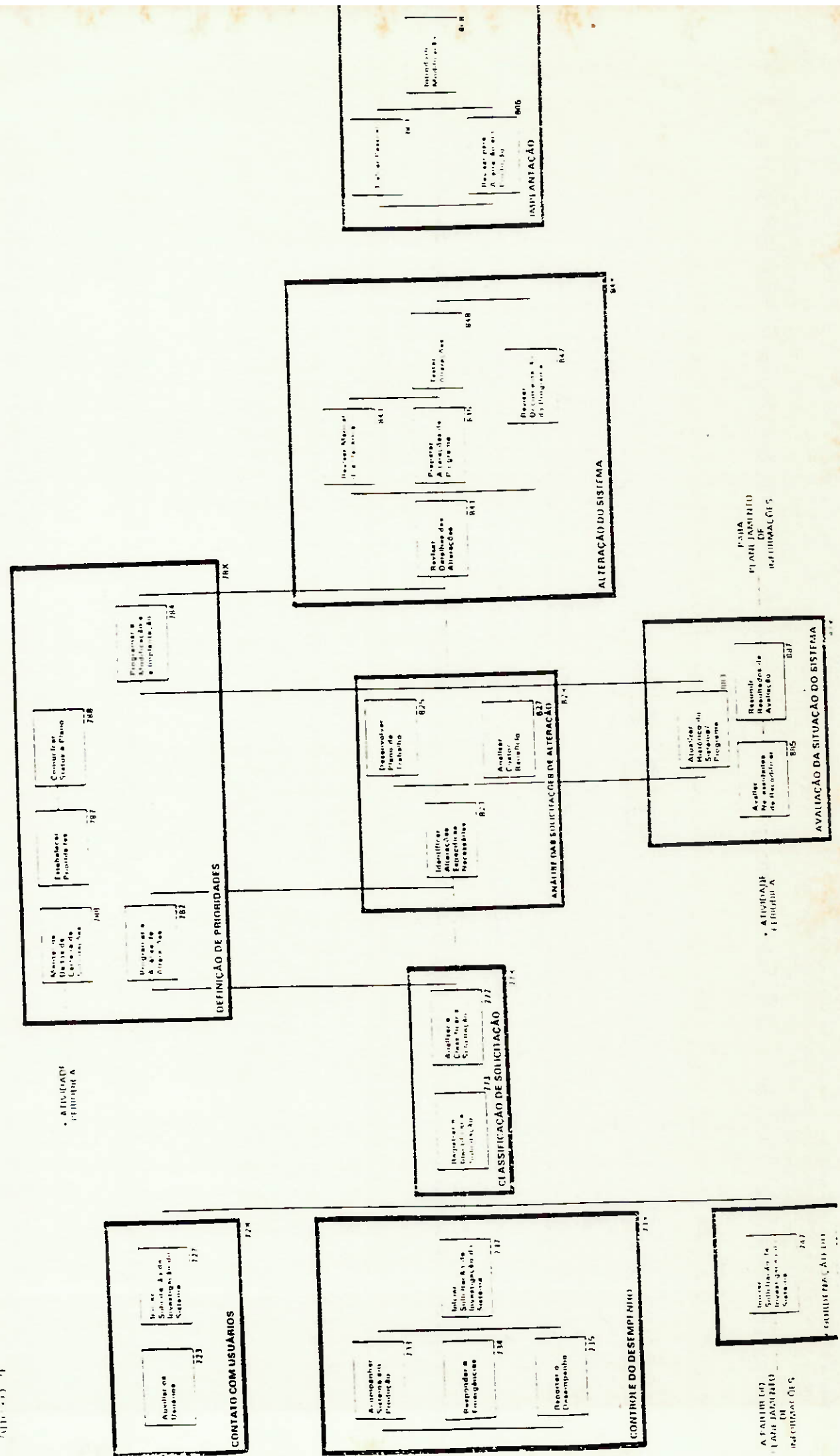


DIAGRAMA DE PLANEJAMENTO - SISTEMAS DE INFORMAÇÃO

1.1. Planejamento do SI (1100) 1
 1.1.1. 110
 1.1.2. 110
 1.1.3. 110

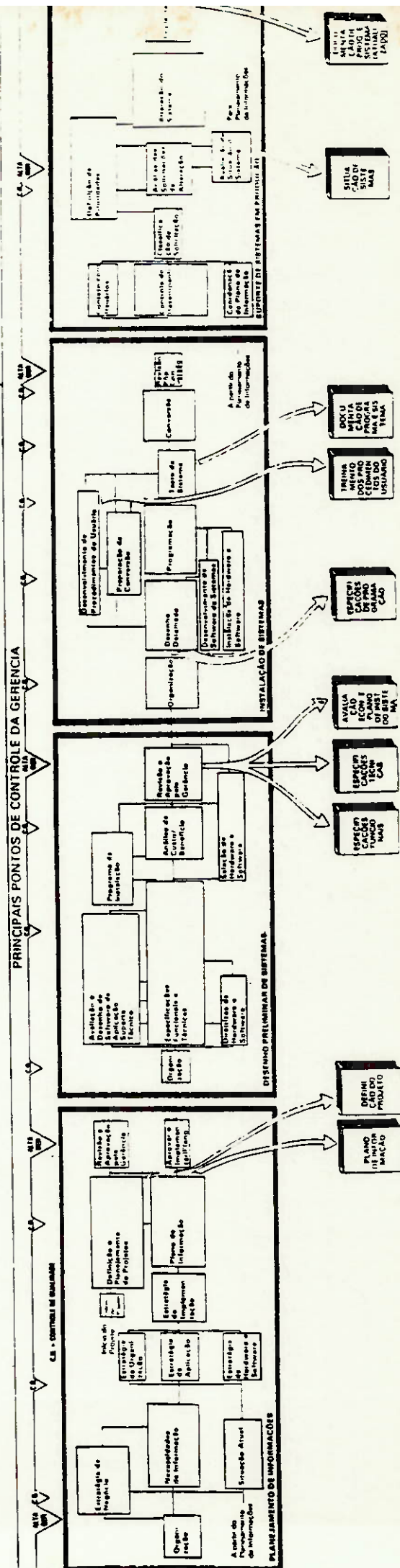


SUPORTE DE SISTEMAS EM PRODUÇÃO DIAGRAMA DE PLANEJAMENTO - SISTEMAS DE INFORMAÇÃO



CICLO DE VIDA DO DESENVOLVIMENTO DE SISTEMAS DIAGRAMA DE PLANEJAMENTO - SISTEMAS DE INFORMAÇÃO

Treinamento do M1110D/1
 Tópico 110
 GP- 110
 Anexo 4



ANEXO 3

Um Exemplo de Especificação de um Sistema por CASE.

Neste anexo reproduzimos um exemplo elaborado pelo Prof. Décio Teixeira para a especificação de um sistema numa locadora de vídeo - fitas através do CASE SMART. O exemplo foi fornecido pela empresa INTERTEC.

A LOCADORA DE VIDEO - FITAS

O Sr. Chaplin, proprietário da locadora de vídeo-fitas Bye Bye Brazil, devido a violenta concorrência existente no ramo, resolveu melhorar seus serviços comprando um computador. Como sói acontecer, o Sr. Chaplin entrou em contato com vendedores e comprou o melhor equipamento, aquele que após ligado resolveria todos os seus desejos (segundo o vendedor, é claro !) Logo ele percebeu o logro, mas não se deu por vencido: procurou "entendidos" em computador para fazer um programa que resolvesse o problema. Passado algum tempo, e muitos problemas de processamento, o Sr. Chaplin foi aconselhado por uma velha amiga, a Sra. Ada, a consultar uma empresa que realmente entendesse de sistemas informativos. Obviamente o Sr. Chaplin procurou nossa empresa. Explicamos a ele que era necessário um serviço de análise para estruturarmos a informação da Bye Bye. Ele concordou e marcamos a primeira reunião com o Sr. Allen, o gerente da Bye Bye.

PRIMEIRA REUNIAO COM O SR. ALLAN

Após as formalidades iniciais, mostramos ao Sr. Allan que precisávamos entender o mundo real da Bye Bye Brazil. Ele iniciou dizendo-nos que ela estava estruturada em 5 áreas: a gerência, o caixa, a recepção de fitas, a locação de fitas e a manutenção. Essas 5 áreas estão alojadas num sobrado antigo: a gerência e a manutenção no andar superior e as outras áreas no andar inferior. Neste existem duas grandes salas onde estão colocadas fichas descritivas das cópias de filmes disponíveis. A sala maior está reservada para cópias em VHS e a menor para cópias em BETAMAX.

Todo cliente, para levar filmes emprestados, precisa ser inscrito como sócio. Para tanto ele deve pagar uma taxa e informar seu nome, endereço, telefone, CPF e RG. Uma vez inscrito ele pode escolher os filmes desejados acessando as fichas descritivas e levando-as à área de locação onde está localizada a fitoteca. Lá existem dois funcionários, Sr. Redford e Sra. Fonda, que trocam as fichas pelas respectivas cópias e emitem um recibo de locação. Este contém os dados do sócio, o número e a data da locação e todos os códigos e descrições dos filmes emprestados.

Quando um cliente deseja devolver um filme, ele dirige-se à área de recepção e entrega a fita junto com o recibo a uma das duas funcionárias, a Sita. Braga e Srta. Fischer. Ela anota a data da devolução no recibo do sócio e na cópia da Bye Bye. Se todas as cópias do recibo já foram devolvidas, elas enviam os recibos para o caixa, Sr. Funaro, ao qual o sócio efetua o pagamento. Caso o sócio reclame da qualidade da fita, elas anotam no recibo o motivo da reclamação.

Quando um filme desejado pelo sócio não está disponível ele pode entrar numa fila espera informando seu número, o filme e as características do mesmo (tais como idioma, sistema de gravação, etc). Quando uma cópia do filme com tais características é devolvida, o sócio é avisado para vir retirá-la.

A área de manutenção é responsável pela reposição das fitas devolvidas na fitoteca, tirar e colocar fitas em circulação devido a manutenção e controlar a fila de espera.

Após recebermos todas essas informações, perguntamos ao Sr. Allan quais os principais problemas da Bye Bye. Ele nos enumerou os seguintes:

- 1) Em alguns dias da semana, existe um gargalo na área de recepção.
- 2) O Sr. Allan não possui meios de fechar o caixa com segurança e sentimos que ele desconfia do Sr. Funaro.

- 3) Está ficando frequente o extravio de fitas e não existem meios de descobrir as causas.
- 4) Algumas fichas desaparecem resultando numa locação deficiente da respectiva cópia.

SEGUNDA REUNIAO

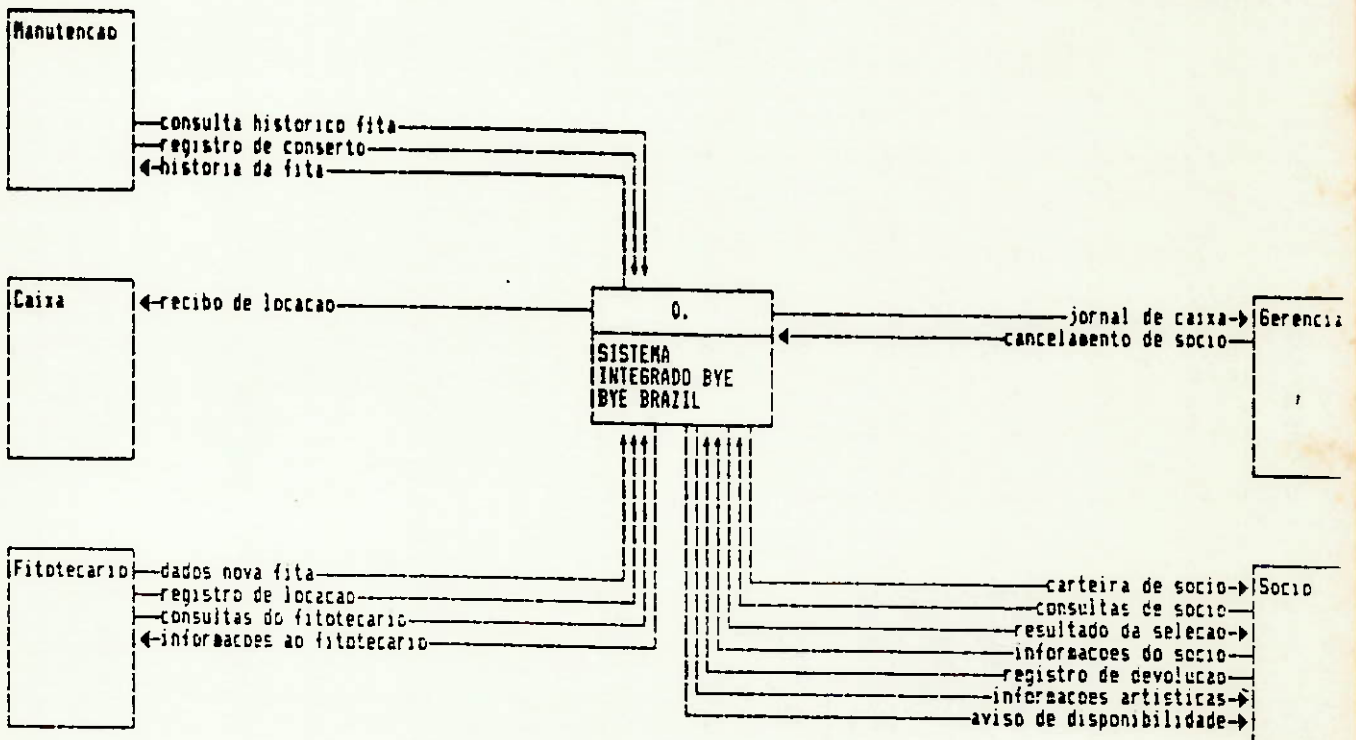
Após termos analisado as informações obtidas com o Sr. Allan, voltamos para a segunda reunião com o Sr. Chaplin e o Sr. Allan.

Procuramos então descobrir os objetivos principais desejados pelos dois diretores da empresa. Eles nos informaram que:

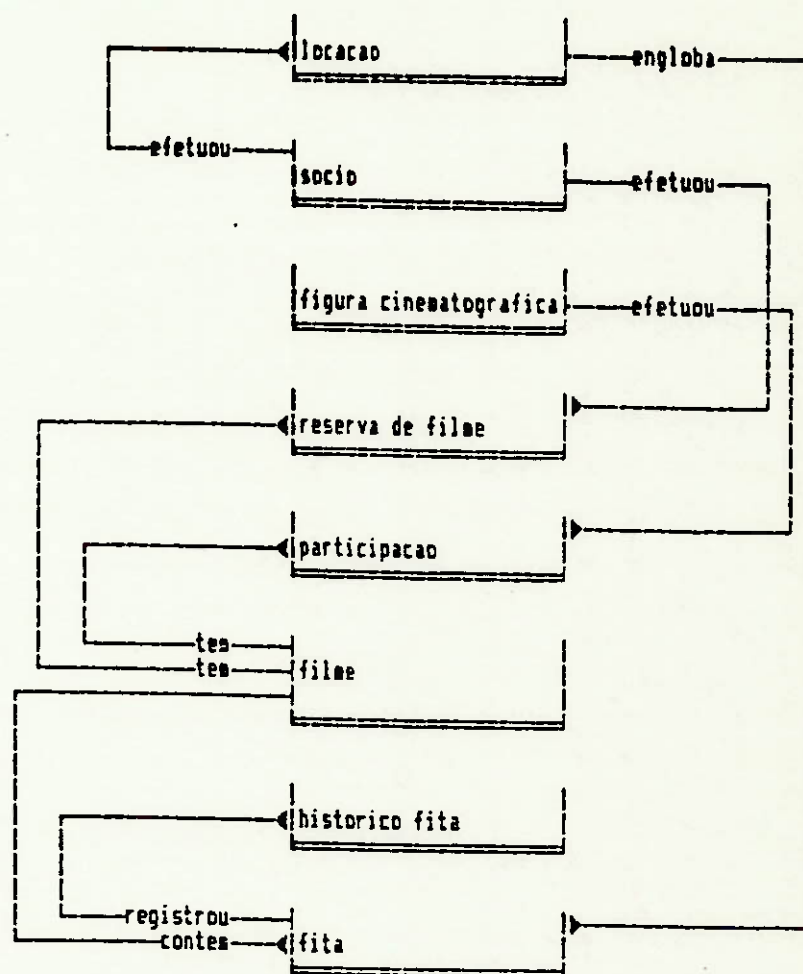
- 1) A principal prioridade deveria ser dada à solução do problema do gargalo e no extravio de fitas.
- 2) Em segundo lugar, eles gostariam de ter um bom controle da fila de espera.
- 3) Se possível eles gostariam de controlar a manutenção das fitas.
- 4) Gostariam que as fichas descritivas fossem emitidas por computador.
- 5) Queriam um relatório diário das locações quitadas no dia para possibilitar um controle do caixa.
- 6) Gostariam de oferecer algumas informações artísticas e técnicas aos sócios, tais como: nome, data de nascimento e morte, nacionalidade de figuras cinematográficas e seus relacionamentos com os filmes (diretor, produtor, artista principal, coadjuvante, roteirista, etc.), ano da filmagem, tempo de duração, país do filme, etc.

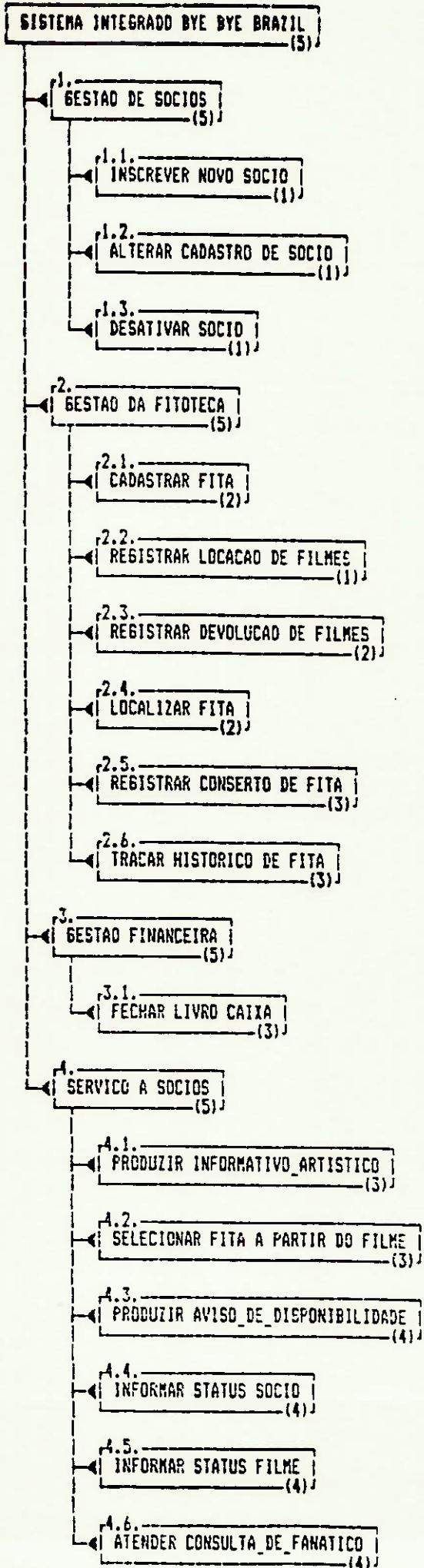
Após alguns papos sobre cinema, prometemos elaborar um modelo de dados completo e apresentá-lo numa terceira reunião.

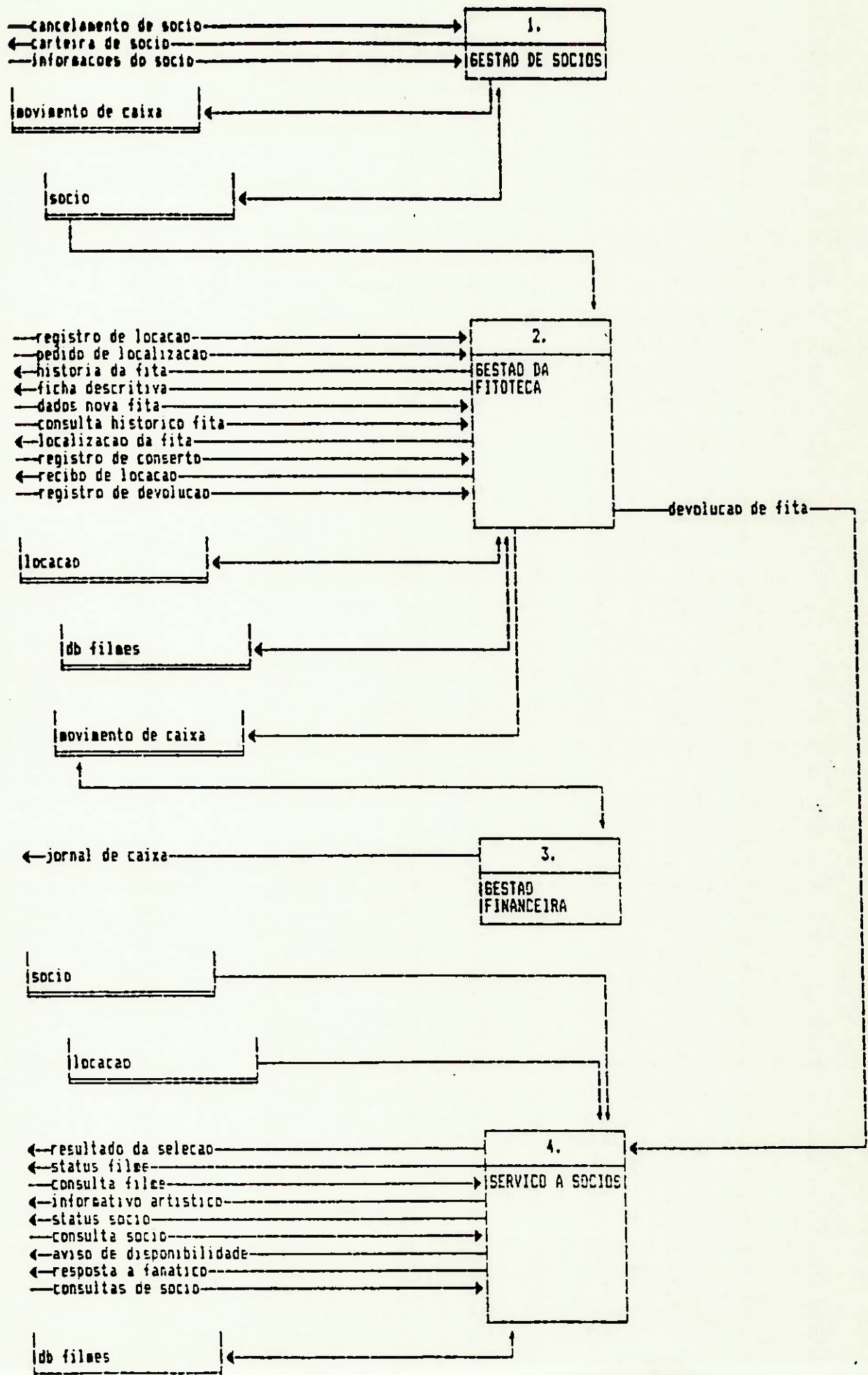
SISTEMA INTEGRADO BYE BYE BRAZIL



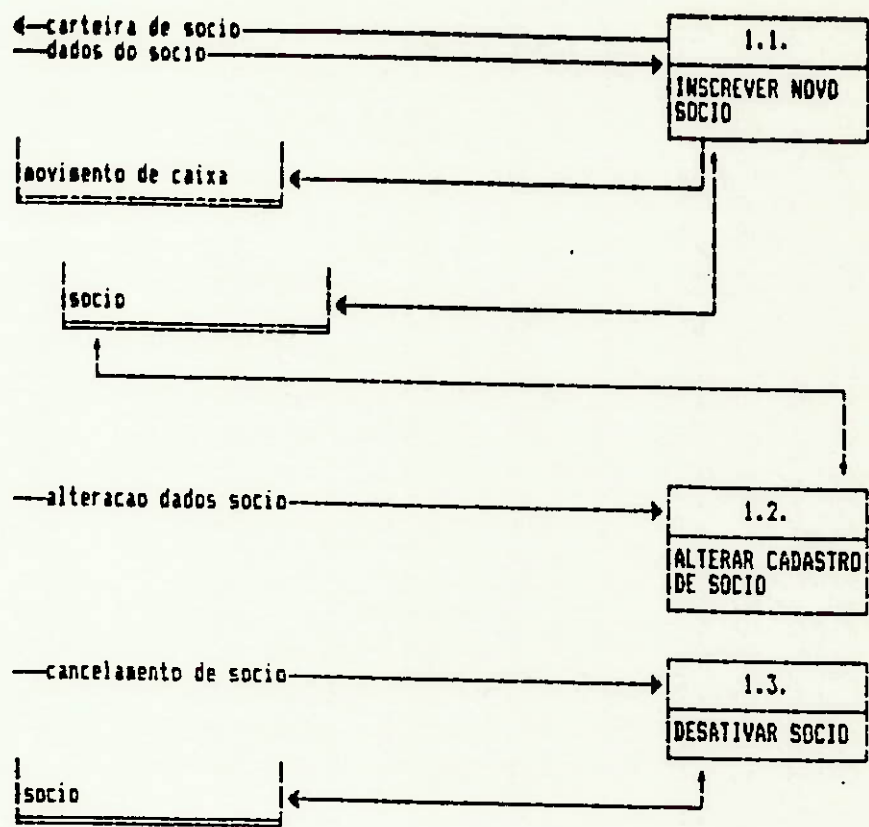
SISTEMA INTEGRADO BYE BYE BRAZIL



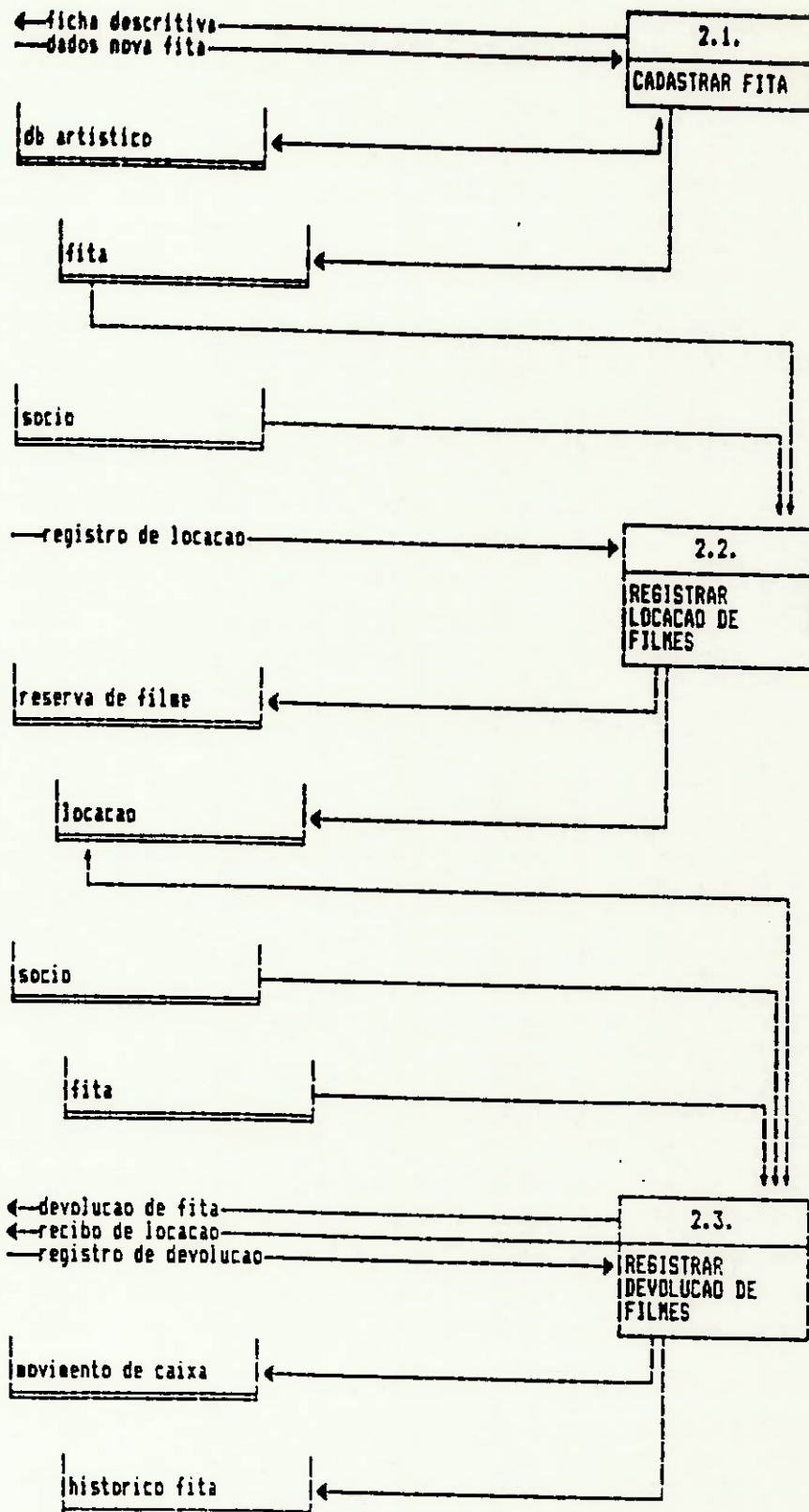




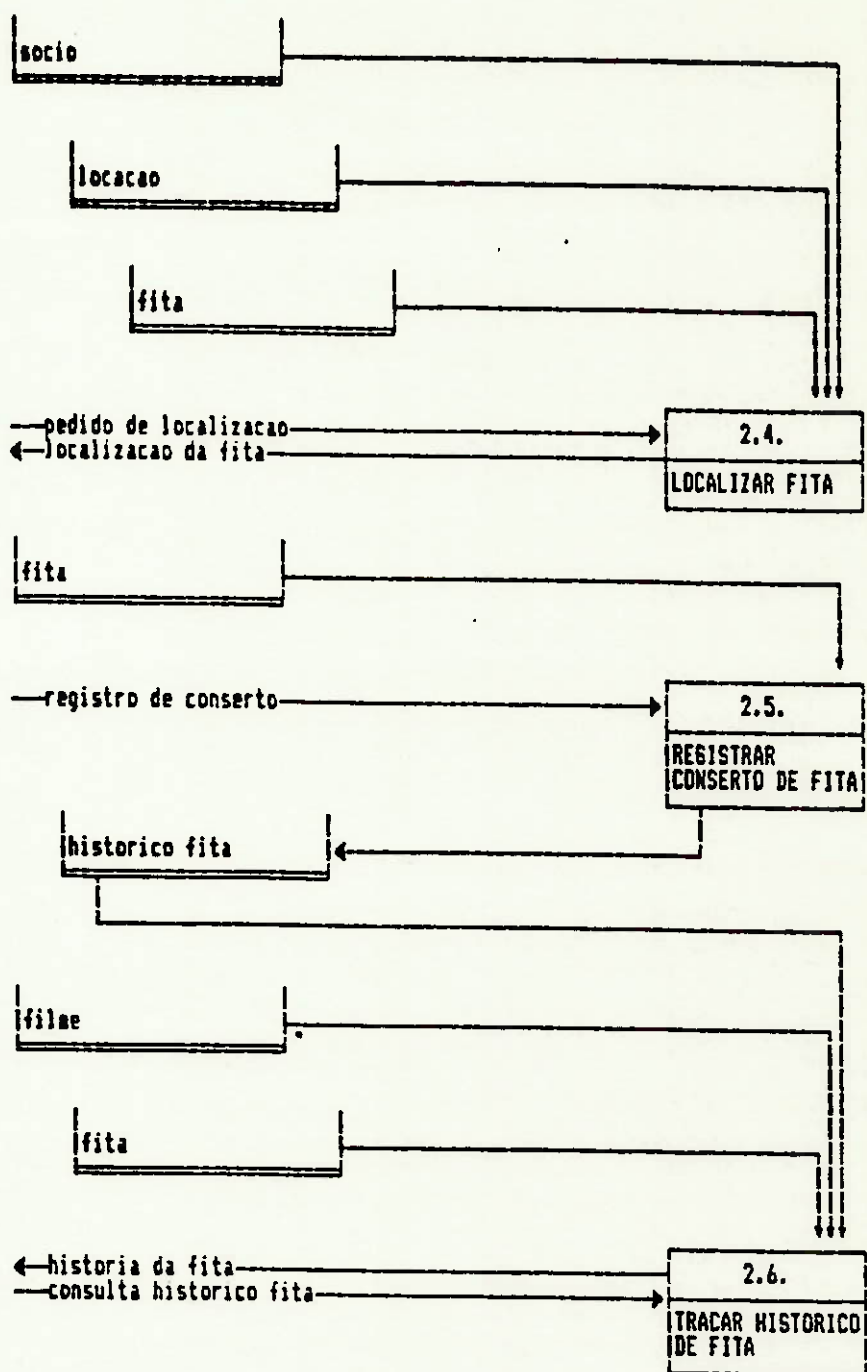
1. - GESTAO DE SOCIOS



2. - GESTAO DA BIBLIOTECA



2. - GESTAO DA FITOTECA



3. - GESTÃO FINANCEIRA

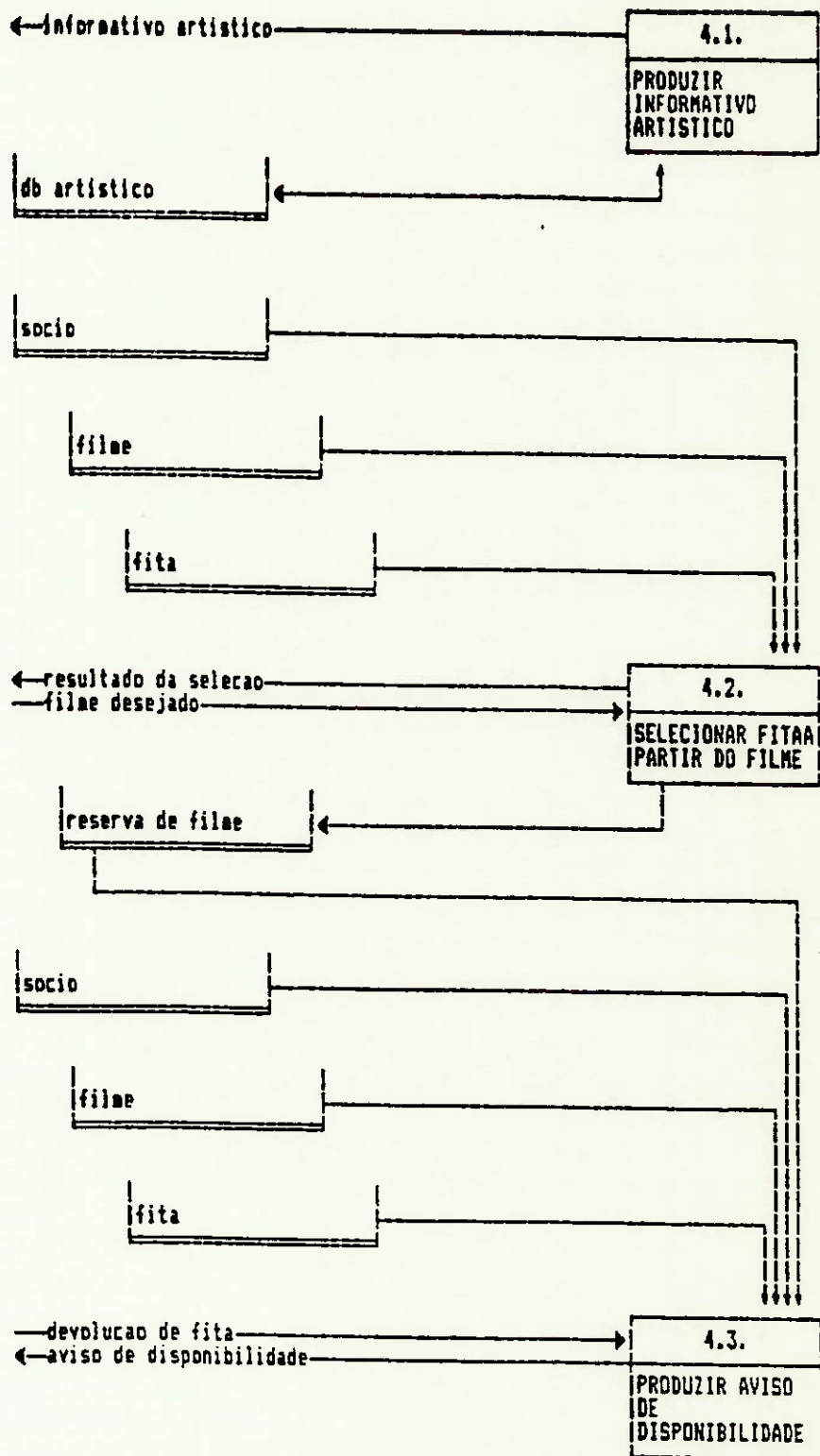
←jornal de caixa

3.1.

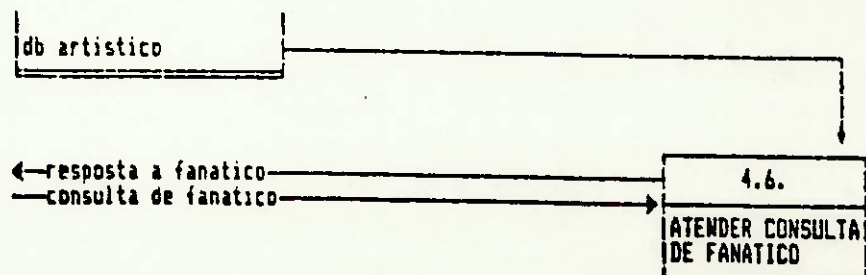
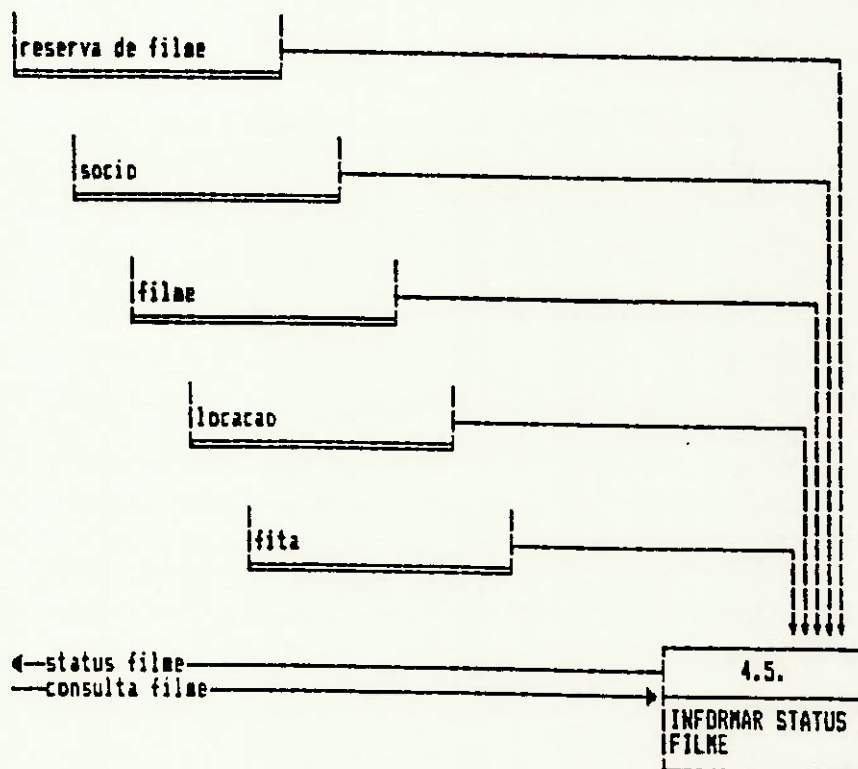
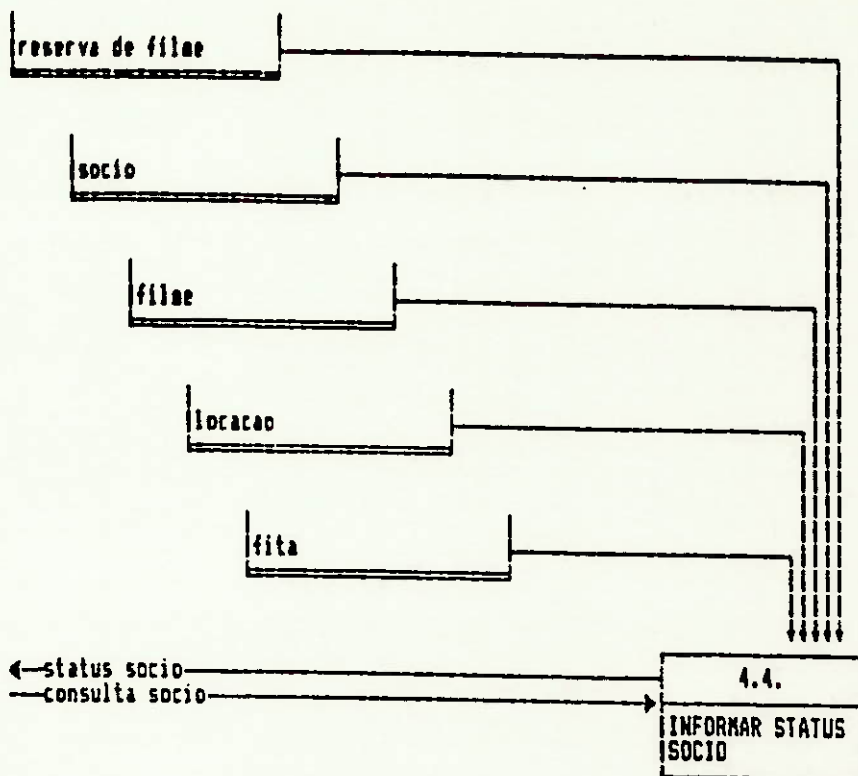
FECHAR LIVRO
CAIXA

movimento de caixa

4. - SERVICO A SOCIOS



4. - SERVICO A SOCIOS



PROCESSES & DATASTORE

RESERVA DE FILME
PARTICIPACAO
MOVIMENTO DE CAIXA
LOCALACAO
HISTORICO FITA
FITA
FILME
FIGURA CINEMATOGRAFIA

REGIST

SMART by FSLA
Rel May 31 1987 PC-DOS

Sep-11-87 6:28p

file: b

Data Store Usage

PROCESSES & DATA

----- **LEGEND** -----

- I - Imports
- E - Exports
- U - Updates
- C - Creates
- R - Removes
- B - Both

SOCIO
RESERVA DE FILME
PARTICIPACAO
MOVIMENTO DE CAIXA
LOCALACAO
HISTORICO FITA
FITA
FILME
FIBURA CINEMATOGRAFIA

ALTERAR CADASTRO DE SOCIO	U
ATENDER CONSULTA DE FANATICO	I I	I . .
CADASTRAR FITA	UC UC EC . . .	UC . .
DESATIVAR SOCIO	UR
FECHAR LIVRO CAIXA	UR . . .
INFORMAR STATUS FILME	. I I . I . .	I I
INFORMAR STATUS SOCIO	. I I . I . .	I I
INSCREVER NOVO SOCIO	EC . . UC
LOCALIZAR FITA	. . I . I . .	I
PRODUZIR AVISO DE DISPONIBILIDADE	. I I	I I
PRODUZIR INFORMATIVO ARTISTICO	I U	I . .
REGISTRAR CONCERTO DE FITA	. . I EC	
REGISTRAR DEVOLUCAO DE FILMES	. . I EC UR EC . .	I
REGISTRAR LOCACAO DE FILMES	. . I . EC . .	ER I
SELECIONAR FITA A PARTIR DO FILME	. I I	EC I
TRACAR HISTORICO DE FITA	. I I I	

S M A R T by FGLA
R01 May 31 1987 PC-DDS

Sep-11-87 6:28p

file: byob
page

External interface

EXTERNAL ENTITIES : FLC

----- LEGEND -----
R - Receives
S - Sends
B - Both

STATUS	SOCIETY
STATUS	FILME
RESULTADO	DASELECAO
REGISTRO	A FANATICO
REGISTRO	DE LOCACAO
REGISTRO	DEVOLUCA
RECIBIDO	DE CONSERVO
PEDIDO	DE LOCALIZACA
LOCALIZACA	DA FITA
JORNAL DE	CINEMA
INFORMATIVO	ARTISTIC
HISTORIA	DA FITA
FILME	DESEJADO
FILMA	DESEJATIVO
DEVOLUCA	DE FITA
DADOS NOV	FITA
DADOS DO	SOCIO
CONSULTA	HISTORICO
CONSULTA	FILME
CONSULTA	DE FANATICO
CANCELAMENTO	DE SOCIO
AVISO DE	DISPONIBILIT
ALTERACAO	DADOS SOCIO

Caixa	R	
Fitotecario	S	.	S	.	S	R	S	.	.	.	S	.	.	R	R
Gerencia	.	.	S	R
Manutencao	S	R	S
Socio	S	R	.	R	S	.	.	.	S	.	.	R	S	.	R	S	.	R	R	.

Full Group Explosion

GROUPS 1 17

----- LEGEND -----
C - Contains

[illegible][illegible]

*** DEMO COPY ***

recibo de locacao C
registro de conserto C
registro de devolucao C
registro de locacao
resposta a fanatico C
resultado da selecao	C
status filme	C	. . . C . . C C . .
status socio	C	. C C C C C C . . .

8 MAR 1 by F66A
Rel May 31 1987 PC-D05

8ep-11-87 6:28p

file:

Internal Interface

PROCESSES 1

----- LEGEND -----
R - Receives
S - Sends
B - Both

ALTERACAO DE DADOS SOCIO	AVISO DE DISPONIBILIDADE	CANCELAMENTO DE SOCIO	CARTEIRA DE FANATICO	CONSULTA DE FANATICO	CONSULTA DE HISTORICO	CONSULTA DE HISTORICO F	CONSULTA DE SOCIO	DADOS DO SOCIO	DADOS DO FITA	DEVOLUCAO DE FITA	FILME DESCRITIVO	FILME DESEJADO	HISTORIA DA FITA	INFORMATIVO ARTISTICO	JORNAL DE CAIXA	LOCALIZACAO DA FITA	PEDIDO DE LOCALIZACAO	RECEBIDO DE CONCERTO	REGISTRO DE DEVOLUCAO	REGISTRO DE LOCALIZACAO	RESULTADO DA SELECAO	STATUS FILME	STATUS SOCIO
--------------------------	--------------------------	-----------------------	----------------------	----------------------	-----------------------	-------------------------	-------------------	----------------	---------------	-------------------	------------------	----------------	------------------	-----------------------	-----------------	---------------------	-----------------------	----------------------	-----------------------	-------------------------	----------------------	--------------	--------------

ALTERAR CADASTRO DE SOCIO

R
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

ATENDER CONSULTA DE FANATICO

.	.	.	R	S
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

CADASTRAR FITA

.	R	S
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

DESATIVAR SOCIO

.	.	R
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

FECHAR LIVRO CAIXA

.	S
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

INFORMAR STATUS FILME

.	.	.	.	R	S	.	.	.
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

INFORMAR STATUS SOCIO

.	R	S	.	.
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

INSCREVER NOVO SOCIO

.	.	S	.	.	.	R
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

LOCALIZAR FITA

.	S	R
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PRODUZIR AVISO DE DISPONIBILIDADE

.	S	R
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

PRODUZIR INFORMATIVO ARTISTICO

.	S
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

REGISTRAR CONCERTO DE FITA

.	R
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

REGISTRAR DEVOLUCAO DE FILMES

.	S	S	R
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

REGISTRAR LOCALIZACAO DE FILMES

.	R
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SELECIONAR FITA A PARTIR DO FILME

.	R	S	.	.	.
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

TRACAR HISTORICO DE FITA

.	R	S
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

S M A R T by FGIA
Rel May 31 1987 PC-DOS

Sep-11-87 6:28p

file:

Memory Usage

PROCESSES : MEMORY

PROL551

----- LEGEND -----
I - Imports
E - Exports
U - Updates

figura_cinematografica	filme	fita	historico_fita	locacao	movimento_de_caixa								
DATA DE NASCIMENTO	ANDADA FILMAGEM	CODIGO	DURACAO	PAIS	TITULO	NUMERO	TIPO	DESCRICAO	DATADOR	VALOR	SOCIO	TIPO	VALOR
ALTERAR CADASTRO DE SOCIO
ATENDER CONSULTA DE FANATICO	.	.	I
CADASTRAR FITA	U	U	U	.	U	U	U	E	E
DESATIVAR SOCIO
FECHAR LIVRO CAIXA	I	I
INFORMAR STATUS FILME	I	.	.	.	I	I	.	.	.
INFORMAR STATUS SOCIO	I	.	.	I	I	.	.	.
INSCREVER NOVO SOCIO	E	E
LOCALIZAR FITA	I
PRODUZIR AVISO DE DISPONIBILIDADE	I	.	.	I	I	.	.	.
PRODUZIR INFORMATIVO ARTISTICO	I	I	I
REGISTRAR CONCERTO DE FITA	I	.	E	E	.
REGISTRAR DEVOLUCAO DE FILMES	I	.	E	E	.
REGISTRAR LOCACAO DE FILMES	I	.	E	E	.
SELECIONAR FITA A PARTIR DO FILME	I
TRACAR HISTORICO DE FITA	I	I	.	I	.

SMART by FGLA
Rel May 31 1987 PC-DOS

Sep-11-87 6:28p

file: b

Process Composition

PROCESSES : PROC

----- LEGEND -----
C - CONTAINS

TRACAR	HISTORICO	DE
SELECCIONAR	FITA	A PA
REGISTRAR	LOCALACAO	DE
REGISTRAR	DEVOLUCAO	
REBISTRAR	CONSERVADO	
PRODUIR	INFORMATIVO	
PRODUIR	AVISO	DE
LOCALIZAR	FITA	IIIIII
INSCREVER	NOVO	SOCIO
INFORMAR	STATUS	SOCIO
INFORMAR	STATUS	FILM
FECMAR	LIVRO	CAXIA
DESATIVAR	SOCIO	IIIIII
CADASTRAR	FITA	IIIIII
ATENDER	CONSULTA	DE
ALTERAR	CADASTRO	DE

[illegible]

Relationship Management

PROCESSES x RELATIONS

----- LEGEND -----	figura_cinematografica	filme	fita	locacao	socio
C - Creates	E	T	C	E	E
R - Removes	F	E	R	F	F
B - Both	U	M	O	U	U
I - Imports	P	A	I	L	R
	A	R	E	O	E
	R	S	S	C	S
	T	E	T	A	E
	I	R	R	C	E
	C	V	O	A	U
	I	A	H	D	O
	P	D	I	C	R
	A	E	S	A	E
	C	I	T	C	U
	A	L	R	A	O
	D	I	I	D	I
	I	L	C	E	E

ALTERAR CADASTRO DE SOCIO
ATENDER CONSULTA DE FANATICO	I	I	.	.	.
CADASTRAR FITA	C	C	.	.	.
DESATIVAR SOCIO
FECHAR LIVRO CAIXA
INFORMAR STATUS FILME	.	.	I	I	I
INFORMAR STATUS SOCIO	.	.	I	I	I
INSCREVER NOVO SOCIO
LOCALIZAR FITA	.	.	.	I	I
PRODUZIR AVISO DE DISPONIBILIDADE	.	.	I	I	.
PRODUZIR INFORMATIVO ARTISTICO	I	I	.	.	.
REGISTRAR CONSERVO DE FITA	.	.	.	C	.
REGISTRAR DEVOLUCAO DE FILMES	.	.	.	C	R
REGISTRAR LOCACAO DE FILMES	.	.	I	.	C
SELECIONAR FITA A PARTIR DO FILME	.	.	C	I	I
TRACAR HISTORICO DE FITA	.	.	.	I	I

```
figura_cinematografica =
    -- efetuou --> participacao
    figura_cinematografica.data_de_morte
    + .data_de_nascimento
    + .nome

filme =
    -- tem --> reserva_de_filme
    -- tem --> participacao
    filme.ano_da_filmagem
    + .codigo
    + .data_do_informativo
    + .duracao
    + .pais
    + .tipo
    + .titulo

fita =
    -- registrou --> historico_fita
    -- contem --> filme
    fita.numero
    + .tipo

historico_fita =
    historico_fita.data
    + .descricao

locacao =
    -- engloba --> fita
    locacao.data
    + .valor

movimento_de_caixa =
    movimento_de_caixa.socio
    + .tipo
    + .valor

participacao =
    participacao.descricao

reserva_de_filme =
    reserva_de_filme.data

socio =
    -- efetuou --> locacao
    -- efetuou --> reserva_de_filme
    socio.cpf
    + .endereco
    + .nome
    + .numero
    + .rg
    + .telefone
    + .tipo_de_video
```

PROCESSES

ALTERAR CADASTRO DE SOCIO
(Starting at:byebye.dlp:25, on list page:1)

ATENDER CONSULTA DE FANATICO
(Starting at:byebye.dlp:278, on list page:4)

CADASTRAR FITA
(Starting at:byebye.dlp:99, on list page:2)

DESATIVAR SOCIO
(Starting at:byebye.dlp:33, on list page:1)

FECHAR LIVRO CAIXA
(Starting at:byebye.dlp:164, on list page:3)

GESTAO DA FITOTECA
(Starting at:byebye.dlp:294, on list page:5)

GESTAO DE SOCIOS
(Starting at:byebye.dlp:288, on list page:5)

GESTAO FINANCEIRA
(Starting at:byebye.dlp:303, on list page:5)

INFORMAR STATUS FILME
(Starting at:byebye.dlp:257, on list page:4)

INFORMAR STATUS SOCIO
(Starting at:byebye.dlp:234, on list page:4)

INSCREVER NOVO SOCIO
(Starting at:byebye.dlp:10, on list page:1)

LOCALIZAR FITA
(Starting at:byebye.dlp:125, on list page:2)

PRODUZIR AVISO DE DISPONIBILIDADE
(Starting at:byebye.dlp:216, on list page:4)

PRODUZIR INFORMATIVO ARTISTICO
(Starting at:byebye.dlp:176, on list page:3)

REGISTRAR CONCERTO DE FITA
(Starting at:byebye.dlp:136, on list page:3)

REGISTRAR DEVOLUCAO DE FILMES
(Starting at:byebye.dlp:65, on list page:2)

REGISTRAR LOCACAO DE FILMES
(Starting at:byebye.dlp:41, on list page:1)

SELECIONAR FITA A PARTIR DO FILME
(Starting at:byebye.dlp:194, on list page:3)

SERVICO A SOCIOS
(Starting at:byebye.dlp:307, on list page:5)

SISTEMA INTEGRADO BYE BYE BRAZIL
(Starting at:byebye.dlp:316, on list page:5)

TRACAR HISTORICO DE FITA
(Starting at:byebye.dlp:150, on list page:3)

EXTERNAL ENTITIES

Caixa
Fitotecario
Gerencia
Manutencao
Socio

DATA FLOWS

alteracao_dados_socio
aviso de disponibilidade
cancelamento de socio
carteira de socio
consulta de fanatico
consulta filme
consulta historico_fita
consulta_socio
dados do socio
dados nova fita
devolucao de fita
ficha descritiva
filme desejado
historia da fita
informativo artistico
jornal de caixa
localizacao da fita
pedido de localizacao
recibo de locacao
registro de conserto
registro de devolucao
registro de locacao
resposta a fanatico
resultado da selecao
status filme
status socio

EXTERNAL EVENTS

inicia um mes
termina um dia

ELEMENTARY DATA ITEMS

codigo_do_filme_consultado
codigo_do_filme_contido
codigo_do_socio_consultado
codigo_filme_desejado
data
data de emissao
defeito
descricao do conserto
fita a localizar
fita locada
identificacao do locatario
nome do heroi
numero da fita consertada
numero da fita consultada
numero da fita devolvida
numero do socio alterado
numero do socio desativado
resposta
socio_solicitante
total do dia

DATA GROUPS OR ITEMS

cadastro de filme
consultas de socio
consultas do fitotecario
db artistico
db filmes
informacoes ao fitotecario
informacoes artisticas
informacoes do socio

*** STRUCTURED SPECIFICATION ***

DEFINA dados do socio = nome de socio + endereco + telefone + cpf + rg
+ tipo de video

DEFINA carteira de socio = numero de socio + nome -- cada socio precisa
-- uma carteira se nao como pode retirar fitas ?

PROCESSO inscrever novo socio
RECEBA dados do socio PROVENIENTE DE Socio
CRIE socio
numero fica proximo numero sequencial disponivel
PRODUZA carteira de socio DESTINADO A Socio
CRIE movimento de caixa
tipo fica "TAXA DE INSCRICAO"
valor fica valor da taxa de inscricao
socio fica numero de socio -- nao precisamos de um relacionamento

DEFINA alteracao dados socio = numero do socio alterado
+ < nome de socio ; endereco ; telefone ; cpf ; rg
; tipo de video >

PROCESSO alterar cadastro de socio
RECEBA alteracao dados socio PROVENIENTE DE Socio
considere o socio com numero = numero do socio alterado
atualize os dados informados

DEFINA cancelamento de socio = numero do socio desativado

PROCESSO desativar socio
RECEBA cancelamento de socio PROVENIENTE DE Gerencia
considere o socio com numero = numero do socio desativado
REMOVA socio

DEFINA registro de locacao = identificacao do locatario + { fita locac

PROCESSO registrar locacao de filmes
RECEBA registro de locacao PROVENIENTE DE Fitotecario
considere o socio com numero = identificacao do locatario
CRIE locacao
LEMBRE QUE socio efetuou locacao
PARA CADA fita locada
SE existe reserva de filme que socio efetuou tal que
filme tem reserva de filme
REMOVA reserva de filme
considere a fita com numero = fita locada
LEMBRE QUE locacao engloba fita
data de locacao fica data atual
valor de locacao fica 0

DEFINA registro de devolucao = { numero da fita devolvida
+ (defeito) }

DEFINA recibo de locacao = nome de socio + valor de locacao

DEFINA devolucao de fita = numero da fita devolvida

*** STRUCTURED SPECIFICATION ***

PROCESSO registrar devolucao de filmes

RECEBA registro de devolucao PROVENIENTE DE Socio

PARA CADA item do registro de devolucao

considere a fita com numero = numero da fita devolvida

considere a locacao que engloba a fita

ESQUECA QUE locacao engloba fita

valor de locacao aumenta de valor da locacao da fita considerada

SE e informado defeito

CRIE historico fita

data fica data atual

descricao fica defeito

LEMBRE QUE fita registrou historico fita

PRODUZA devolucao de fita

SE nao existe mais QUALQUER fita que a locacao engloba

PRODUZA recibo de locacao DESTINADO A Caixa

CRIE movimento de caixa

tipo fica "RECIBO DE LOCACAO"

socio fica numero de socio

valor de movimento de caixa fica valor de locacao

REMOVA locacao

DEFINA cadastro de filme = codigo de filme + tipo + titulo + ano da fita
+ pais + duracao + (nome de figura cinematografica
+ data de nascimento + (data de morte) + descricao de participacao

DEFINA ficha descritiva = cadastro de filme

DEFINA dados nova fita = tipo de fita + [cadastro de filme
| codigo do filme contido]

PROCESSO cadastrar fita

RECEBA dados nova fita PROVENIENTE DE Fitotecario

CRIE fita

numero fica proximo numero sequencial disponivel

SE e fornecido cadastro do filme

CRIE filme

PARA CADA figura cinematografica informada

SE ainda nao cadastrada

CRIE figura cinematografica

CRIE participacao

LEMBRE QUE figura cinematografica efetuou participacao

LEMBRE QUE filme tem participacao

PRODUZA ficha descritiva DESTINADO A Socio -- p/ sala de escuta

SE NAO -- o filme ja e cadastrado

considere filme com codigo = codigo do filme contido

LEMBRE QUE fita contem filme

DEFINA pedido de localizacao = fita a localizar

DEFINA localizacao da fita = ["DISPONIVEL" ; numero de socio + telefone
+ endereco + data de locacao]

PROCESSO localizar fita

RECEBA pedido de localizacao PROVENIENTE DE Fitotecario

considere a fita com numero = fita a localizar

considere a locacao que engloba a fita

considere o socio que efetuou a locacao

PRODUZA localizacao da fita DESTINADO A Fitotecario

DEFINA registro de conserto = numero da fita consertada
+ descricao do conserto

*** STRUCTURED SPECIFICATION ***

PROCESSO registrar conserto de fita
RECEBA registro de conserto PROVENIENTE DE Manutencao
considere a fita com numero = numero da fita consertada
CRIE historico fita
data fica data atual
descricao fica descricao do conserto
LEMBRE QUE fita registrou historico fita

DEFINA consulta historico fita = numero da fita consultada

DEFINA historia da fita = numero de fita + codigo de filme + titulo
+ (data de historico fita + descricao)

PROCESSO tracar historico de fita
RECEBA consulta historico fita PROVENIENTE DE Manutencao
considere a fita com numero = numero da fita consultada
considere o filme que a fita contem
PARA CADA historico fita que a fita registrou
em ordem de data de historico fita
adicione um item a historia da fita
PRODUZA historia da fita DESTINADO A Manutencao

DEFINA jornal de caixa = data + (tipo de movimento de caixa + socio
+ total do dia

PROCESSO fechar livro caixa SEMPRE QUE termina um dia
data fica data do dia que esta terminando
PARA CADA movimento de caixa
total do dia aumenta de valor
REMOVA movimento de caixa
PRODUZA jornal de caixa DESTINADO A Gerencia

DEFINA informativo artistico = data de emissao + (cadastro de filme
-- informativo mensal aos socios sobre novos filmes

PROCESSO produzir informativo artistico SEMPRE QUE inicia um mes
data de emissao fica data atual
PARA CADA filme onde data do informativo = "INDEFINIDO"
PARA CADA participacao o filme tem
considere a figura cinematografica que efetuou a particip
adicione um item ao cadastro de filme
adicione cadastro de filme a informativo artistico
data do informativo de filme fica data de emissao
PRODUZA informativo artistico DESTINADO A Socio

DEFINA filme desejado = codigo filme desejado
+ ("reservavel" + socio solicitante)

DEFINA resultado da selecao = resposta

PROCESSO selecionar fita a partir do filme
RECEBA filme desejado PROVENIENTE DE Socio
considere o filme com codigo = codigo filme desejado
SE existe QUALQUER fita que contem o filme e nao existe locacao -
que engloba a fita e tipo de video de socio = tipo de fit
resposta fica numero de fita
SE NAO SE e especificado "reservavel"
CRIE reserva de filme
data fica data e hora atual
LEMBRE QUE filme tem reserva de filme
considere o socio com numero = socio solicitante
LEMBRE QUE socio efetuou reserva de filme
resposta fica "RESERVADA!"
SE NAO
resposta fica "NAO DISPONIVEL!"
PRODUZA resultado da selecao DESTINADO A Socio

Sep-11-87 6:12p

file:

*** STRUCTURED SPECIFICATION ***

DEFINA aviso de disponibilidade = titulo de filme + codigo + nome de
+ telefone

PROCESSO produzir aviso de disponibilidade

RECEBA devolucao de fita

considere a fita com numero = numero da fita devolvida

considere o filme que a fita contem

SE existe QUALQUER reserva de filme que o filme tem

considere a primeira reserva de filme em ordem de data

onde o socio que efetuou a reserva de filme

tem tipo de video de socio = tipo de fita

PRODUZA aviso de disponibilidade DESTINADO A Socio

DEFINA consulta socio = codigo do socio consultado

DEFINA status socio = numero de socio + nome + endereco + telefone +
+ (data de locacao + numero de fita + codigo de filme + tit.
+ (data de reserva de filme + codigo de filme + titulo))

PROCESSO informar status socio

RECEBA consulta socio PROVENIENTE DE Fitotecario

considere o socio com numero = codigo do socio consultado

PARA CADA locacao que o socio efetuou

PARA CADA fita que a locacao engloba

considere o filme que a fita contem

adicione um item a status socio

PARA CADA reserva de filme que o socio efetuou

considere o filme que tem a reserva de filme

adicione um item a status socio

PRODUZA status socio DESTINADO A Fitotecario

DEFINA consulta filme = codigo do filme consultado

DEFINA status filme = codigo de filme + titulo
+ (numero de fita + tipo
+ (data de locacao + nome de socio + telefone)) -- se locada
+ (data de reserva de filme + tipo de video de socio + nome

PROCESSO informar status filme

RECEBA consulta filme PROVENIENTE DE Fitotecario

considere o filme com codigo = codigo do filme consultado

PARA CADA fita que contem o filme

SE existe locacao que engloba a fita

considere o socio que efetuou a locacao

complete item de status filme com os dados da locacao

PARA CADA reserva de filme que o filme tem

considere o socio que efetuou a reserva de filme

adicione um item a status filme

PRODUZA status filme DESTINADO A Fitotecario

DEFINA consulta de fanatico = nome do heroi

DEFINA resposta a fanatico = (codigo de filme + titulo + tipo
+ ano da filmagem + pais + duracao + descricao de participaca

PROCESSO atender consulta de fanatico

RECEBA consulta de fanatico PROVENIENTE DE Socio

considere a figura cinematografica com nome = nome do heroi

PARA CADA participacao que a figura cinematografica efetuou

considere o filme que tem a participacao

adicione um item a resposta a fanatico

PRODUZA resposta a fanatico DESTINADO A Socio

*** STRUCTURED SPECIFICATION ***

- PROCESSO gestao de socios
 - CONTEM inscrever novo socio (s)
 - CONTEM alterar cadastro de socio (s)
 - CONTEM desativar socio (s)
- PROCESSO gestao da fitoteca
 - CONTEM cadastrar fita (s)
 - CONTEM registrar locacao de filmes (s)
 - CONTEM registrar devolucao de filmes (s)
 - CONTEM localizar fita (s)
 - CONTEM registrar conserto de fita (s)
 - CONTEM tracar historico de fita (s)
- PROCESSO gestao financeira
 - CONTEM fechar livro caixa (s)
- PROCESSO servico a socios
 - CONTEM produzir informativo artistico (s)
 - CONTEM selecionar fita a partir do filme (s)
 - CONTEM produzir aviso de disponibilidade (s)
 - CONTEM informar status socio (s)
 - CONTEM informar status filme (s)
 - CONTEM atender consulta de fanatico (s)
- PROCESSO sistema integrado Bye Bye Brazil
 - CONTEM gestao de socios (s)
 - CONTEM gestao da fitoteca (s)
 - CONTEM gestao financeira (s)
 - CONTEM servico a socios (s)

DEFINA informacoes do socio = dados do socio + alteracao dados socio

DEFINA informacoes artisticas = ficha descritiva + informativo artist
+ resposta a fanatico

DEFINA consultas de socio = filme desejado + consulta de fanatico

DEFINA consultas do fitotecario = pedido de localizacao + consulta so
+ consulta filme

DEFINA informacoes ao fitotecario = status socio + status filme
+ localizacao da fita

DEFINA db artistico = filme + figura cinematografica + participacao

DEFINA db filmes = db artistico + fita + reserva de filme

ANEXO 4

Listagens resultantes da aplicação do Gerenciador de
Dicionário de Dados e do MIND-AID
para uma parte do Sistema de
Câmbio.

Listagens Resultantes da Aplicação do GDD.

ENTIDADES DO MODELO DE DADOS DE CAMBIO

=====

NUMERO ENTIDADE

175	ADIANTAMENTO CONCEDIDO
65	AVISO DEBITO CREDITO RECEBIDO
195	BANCO NO EXTERIOR
40	CAMBIAL OU CARTA REMESSA
685	CARTA DE CREDITO EXPORTACAO
680	CARTA DE CREDITO EXPORTACAO-NEGOCIACAO
315	CLIENTE
2	CONTRATO DE CAMBIO-EXPORTACAO
565	ENCARGOS REGIME DE COMPETENCIA
545	LIQUIDACAO DO CONTRATO DE CAMBIO NO PAIS

MODELO DE DADOS DE CAMBIO - RELACIONAMENTOS

=====

ENTIDADE 1	RELACAO	ENTIDADE 2
AVISO DEBITO CREDITO RECEBIDO	N 1	BANCO NO EXTERIOR
BANCO NO EXTERIOR	1 N	AVISO DEBITO CREDITO RECEBIDO
BANCO NO EXTERIOR	1 N	ENCARGOS REGIME DE COMPETENCIA
CAMBIAL OU CARTA REMESSA	N 1	CLIENTE
CLIENTE	1 N	CAMBIAL OU CARTA REMESSA
CLIENTE	1 N	ENCARGOS REGIME DE COMPETENCIA
ENCARGOS REGIME DE COMPETENCIA	N 1	BANCO NO EXTERIOR
ENCARGOS REGIME DE COMPETENCIA	N 1	CLIENTE

ENTIDADES DO MODELO DE DADOS DE CAMBIO

PAG.

1

=====

ADIANTAMENTO CONCEDIDO

SAO OS ADIANTAMENTOS PARCIAIS OU TOTAIS EM RELACAO AO VALOR DO CONTRATO DE CAMBIO EM M/N CONCEDIDOS AOS CLIENTES/BANCOS, QUANDO PAGUADO EM UMA OPERACAO DE EXPORTACAO OU COMPRA INTERBANCARIA

ATRIBUTOS

COD FORMA ENTREGA M/N AO BFB
 COD FORMA ENTREGA M/N PELO BFB
 DATA CONCESSAO ADIANTAMENTO
 DATA VENCTO
 INDICADOR FASE ADTO
 INDICATIVO CREDITO LIQUIDACAO (S/N)
 NR OPERACAO
 NR OPERACAO-REFERENCIA
 SALDO A LIQUIDAR M/E
 SALDO A LIQUIDAR M/N
 VALOR M/N

CHAVE

AVISO DEBITO CREDITO RECEBIDO

SAO OS AVISOS DE DEBITO OU CREDITO SOBRE NOSSAS CTAS CORRENTES NO EXTERIOR EM CONSEQUENCIA DAS OPERACOES TRANSACIONADAS COM O BFB

ATRIBUTOS

COD BANCO EXTERIOR
 COD MOEDA
 DATA AVISO
 DATA INSERCAO NO EXTRATO
 DATA RECEBTO DO AVISO
 DATA VALOR
 DEBITO/CREDITO
 NR CONTA CORRENTE EXTERIOR
 NR OPERACAO-REFERENCIA
 NR ORIGEM
 NR REFERENCIA BANCO NO EXTERIOR
 ORIGEM DO AVISO
 PENDENTE/NAO PENDENTE
 VALOR M/E

CHAVE

CHAVE

BANCO NO EXTERIOR

CADASTRO DE BANCOS NO EXTERIOR

ATRIBUTOS

AREA CONTATO
 CATEGORIA
 COD BANCO EXTERIOR
 COD INSTITUICAO
 COD SWIFT BCO EXT
 CONDICoes DE COBRANCA
 CONDICoes NEGOCIACAO DE TRAVELLER'S CHECKS
 CONTA CORRENTE (S/N)
 CONVENIADO (S/N)
 DATA ACORDO P/ VENDA TRAVELLER'S CHECKS
 ENDERECO BANCO NO EXTERIOR
 ENDERECO TELEGRAFICO
 NOME

CHAVE

=====

ATRIBUTOS (CONTINUACAO)

NOME CONTATO
NR TELEX I
NR TELEX II
NR TELEX III
SALDO MEDIO PACTUADO

CAMBIAL OU CARTA REMESSA

CONJUNTO DE DOCUMENTOS QUE REPRESENTAM UM EMBARQUE.

ATRIBUTOS

COD GRUPO VINCULACAO
COD PAIS DESTINO
COD SEGURADORA
COD TRANSPORTADORA
DATA AVISO RECEBTO CAMBIAL
DATA DESEMBOLSO
DATA EMBARQUE
DATA PREVISTA FORMAL P/ RECEPCAO DOS DOCTOS
DATA PREVISTA INFORMAL P/ RECEPCAO DOS DOCTOS
DATA RECEPCAO DOCTOS
DATA REMESSA DOCTOS
ENDERECO SACADO OU SACADOR
INDICADOR IMPORTACAO/EXPORTACAO
INDICATIVO DE RECEPCAO (RECEBIDO/ A RECEBER)
NOME SACADO OU SACADOR
NR CONTRATO DE GARANTIA DE IMPORTACAO
NR OPERACAO
NR REMESSA
TIPO TRANSPORTE
VALOR EMBARCADO M/E
VALOR EMBARCADO M/N
VALOR PREVISTO DO EMBARQUE

CHAVE

CARTA DE CREDITO EXPORTACAO

DOCUMENTO ONDE SAO DELINEADOS OS TERMOS E CONDICAOES EM QUE UMA OP
ERACAO DE EXPORTACAO AMPARADA SOB A MODALIDADE CREDITO, DEVA SER
CONCRETIZADA

ATRIBUTOS

A VISTA/A PRAZO
COD BANCO AVISADOR
COD BANCO CONFIRMADOR
COD BANCO INSTITUIDOR
COD BANCO NEGOCIADOR
COD CONVENIO
COD MOEDA
COD TIPO GARANTIA OBTIDA
CONFIRMADA (S/N)
DATA ABERTURA
DATA AVISO AO BENEFICIARIO
DATA CONFIRMACAO
DATA LIMITE P/ NEGOCIACAO
DATA LIMITE VALIDADE

=====

ATRIBUTOS(CONTINUACAO)

DATA RECEBTO DA CARTA DE CREDITO
EMENDADA (S/N)
NR OPERACAO
NR OPERACAO-REFERENCIA
NR REFERENCIA BANCO NO EXTERIOR
PERCENTUAL RED CLAUSE
RED CLAUSE (S/N)
REVOGAVEL (S/N)
TRANSFERIVEL (S/N)
VALOR FOB M/E
VALOR FRETE M/E
VALOR M/E
VALOR M/E JUROS L/C FINANCIADA
VALOR SEGURO M/E

CHAVE

CARTA DE CREDITO EXPORTACAO-NEGOCIACAO

E A ENTREGA E ANALISE DOS DOCTOS DE EMBARQUE EM CONJUNTO COM A CA
RTA DE CREDITO

ATRIBUTOS

DATA LIBERACAO DA RESERVA
DATA NEGOCIACAO
DATA SOLICITACAO CLIENTE P/ SOLUCAO DISCREPANCIA
DISCREPANCIA (S/N)
ENVIO DE DOCTOS SOB RESERVA (S/N)
LIBERACAO DA RESERVA (S/N)
NR OPERACAO
NR OPERACAO-REFERENCIA

CHAVE

CLIENTE

REFERE-SE AOS CLIENTES CADASTRADOS JUNTO AO BFB. OS ATRIBUTOS SAO
EXTENSIVAMENTE DESCRITOS NO MODELO DE DADOS DO PROJETO CLIENTES

ATRIBUTOS

COD CLIENTE
ENDERECO CLIENTE
NOME
NOME ABREVIADO
STATUS DO CLIENTE

CHAVE

CONTRATO DE CAMBIO-EXPORTACAO

E O CONTRATO DE CAMBIO REGULAMENTADO PELO BACEN ESTABELECIDO ENTR
E BFB E SEUS CLIENTES QUE DEFINE UMA OPERACAO DE EXPORTACAO EFETU
ADA COM O BFB

ATRIBUTOS

COD MODALIDADE TRANSACAO
NR LINHA CREDITO
PERCENTUAL DE ADIANTAMENTO NO FECHTO
SALDO SAQUES A VINCULAR
VALOR FOB M/E
VALOR FRETE M/E
VALOR SEGURO M/E

=====

ENCARGOS REGIME DE COMPETENCIA

SAO AS DESPESAS OU RECEITAS APROPRIADAS NO TEMPO :

ATRIBUTOS

AM/AA
COD BANCO EXTERIOR
COD FORMA ENTREGA M/N AO BFB
COD FORMA ENTREGA M/N PELO BFB
COD MOEDA
COD TIPO ENCARGO
DATA CONVERSÃO DO INDEXADOR P/ M/N
DATA FIM P/ CALCULO
DATA INICIO P/ CALCULO
DATA PAGTO EFETIVA
DATA PREVISTA P/ PAGTO
DATA PREVISTA P/ RECEBTO
DATA RECEBTO EFETIVA
DEBITO/CREDITO
INDICATIVO CREDITO LIQUIDACAO (S/N)
NR CONTA CORRENTE EXTERIOR
NR OPERACAO
NR OPERACAO-REFERENCIA
NR OPERACAO-REFERENCIA 1
TAXA BASE PARA CALCULO
VALOR
VALOR BASE P/ CALCULO
VALOR I.R. S/ ENCARGO

CHAVE

LIQUIDACAO DO CONTRATO DE CAMBIO NO PAIS

E A ENTREGA OU RECEBIMENTO DE MOEDA NACIONAL REFERENTE AS OPERACO
ES DE CAMBIO REALIZADAS COM O BFB

ATRIBUTOS

DATA EFETIVA DE LIQUIDACAO
NR OPERACAO
NR OPERACAO-REFERENCIA
NR OPERACAO-REFERENCIA 1
VALOR COMISSAO AGENTE M/E
VALOR FOB M/E LIQUIDADO
VALOR FRETE M/E LIQUIDADO
VALOR LIQUIDADO M/E
VALOR LIQUIDADO M/N
VALOR SEGURO M/E LIQUIDADO

CHAVE

A VISTA/A PRAZO

DESCRICAO: DEFINE SE O PAGTO OU RECEBTO E A VISTA OU A PRAZO

REGRA VALIDAC:U OU P

FORMATO: X(1)

DOMINIO: U= A VISTA/ P= A PRAZO

AM/AA

DESCRICAO: TAXA AO MES OU AO ANO

REGRA VALIDAC:AM OU AA

FORMATO: X(2)

DOMINIO: AM=AO MES/AA=AO ANO

AREA CONTATO

DESCRICAO: DEFINE A AREA/ORGAO DO BANCO NO EXTERIOR COM O QUAL TEMOS USUALME
NTE CONTATO

REGRA VALIDAC:

FORMATO: X(30)

DOMINIO:

CATEGORIA

DESCRICAO: INDICADOR QUE DETERMINA A CATEGORIA DO BANCO NO EXTERIOR

REGRA VALIDAC:IGUAL A 1 OU 2

FORMATO: 2(1)

DOMINIO: 1-PRIMEIRA LINHA/2-SEGUNDA LINHA

COD BANCO AVISADOR

DESCRICAO: CODIGO QUE IDENTIFICA O BANCO QUE EMITE AVISO AO CLIENTE S/CARTA
DE CREDITO OU PRESTACAO DE GARANTIA/DIGITO DE CONTROLE

REGRA VALIDAC:CONFORME CADASTROS: BANCO NO EXTERIOR E BANCOS NO PAIS/DIG.CONTE.

FORMATO: 9(4)-9

DOMINIO: 0001 A 9999

COD BANCO CONFIRMADOR

DESCRICAO: CODIGO QUE IDENTIFICA O BANCO CONFIRMADOR DA L/C

REGRA VALIDAC:CONFORME CADASTRO: BANCOS NO PAIS E BANCO NO EXTERIOR/DIG.CONTR.

FORMATO: 9(4)-9

DOMINIO: 0001 A 9999

COD BANCO EXTERIOR

DESCRICAO: CODIGO IDENTIFICADOR DO BANCO NO EXTERIOR

REGRA VALIDAC:CONFORME CADASTRO: BANQUEIRO NO EXTRIOR

FORMATO: 9(4)-9

DOMINIO: 0001 A 9999

COD BANCO INSTITUIDOR

DESCRICAO: CODIGO QUE IDENTIFICA O BANCO INSTITUIDOR DA L/C

REGRA VALIDAC:CONFORME CADASTRO:ARQUIVOS BANCOS NO PAIS/BANCO NO EXTERIOR

FORMATO: 9(4)-9

DOMINIO: 0001 A 9999

COD BANCO NEGOCIADOR

DESCRIÇÃO: BANCO NEGOCIADOR DE UMA CARTA DE CREDITO

REGRAS VALIDADE: CONFORME CADASTRO: BANCO NO EXTERIOR/DIGITO DE CONTROLE

FORMATO: 9(4)---

DOMINIO: 00001 A 9999

COD CLIENTE

DESCRIÇÃO: RADICAL ATRIBUÍDO AO CLIENTE DO SEU CADASTRAMENTO IDENTIFICANDO-LO DE FORMA ÚNICA NO BEEB

REGRAS VALIDADE: CONFORME CADASTRO: CLIENTES

FORMATO: 9(5)---

DOMINIO: 00001 A 99999

COD CONVENIO

DESCRIÇÃO: CÓDIGO QUE IDENTIFICA O TIPO DE CONVENIO

REGRAS VALIDADE: CONFORME ARQUIVO: CONVENIOS

FORMATO: 9(1)

DOMINIO:

COD GRUPO VINCULACAO

DESCRIÇÃO: CÓDIGO INDICADOR DO TIPO DE VINCULACAO ENTRE O CLIENTE, NO PAIS, E A PESSOA DO EXTERIOR, CFME O MANUAL ENOC, CFME TIPO DE VINCULACAO

REGRAS VALIDADE:

FORMATO: 9(2)---

DOMINIO:

COD INSTITUICAO

DESCRIÇÃO: INSTITUICAO FINANCEIRA NO EXTERIOR, QUE GRUPO BANCOS NO EXTERIOR, LOCALIZADOS EM DIVERSAS PRACAS/PAISES

REGRAS VALIDADE: CONFORME ARQUIVO: INSTITUICAO FINANCEIRA NO EXTERIOR

FORMATO:

DOMINIO:

COD MODALIDADE TRANSACAO

DESCRIÇÃO: CÓDIGO QUE IDENTIFICA A MODALIDADE DA TRANSACAO, QUANTO A SER CARTA DE CREDITO, COBRANCA, ETC

REGRAS VALIDADE: CONFORME CADASTRO: MODALIDADE DA TRANSACAO

FORMATO: 9(2)

DOMINIO:

COD MOEDA

DESCRIÇÃO: CÓDIGO QUE IDENTIFICA A MOEDA

REGRAS VALIDADE: CALCULAR DIGITO VERIFICADOR CONFORME CADASTRO: MOEDAS

FORMATO: 9(3)---

DOMINIO: TODOS OS CÓDIGOS EXISTENTES NO ARQUIVO MOEDAS

COD PAIS DESTINO

DESCRIÇÃO: CÓDIGO DO PAIS DESTINO DE UM EMBARQUE DE IMPORTACAO CONFORME CADASTRO: PAIS

REGRAS VALIDADE:

FORMATO:

DOMINIO:

COD SEGURADORA

DESCRICAO: CODIGO IDENTIFICADOR DA SEGURADORA DE UMA EXPORTACAO

REGRA VALIDAC: CONFORME CADASTRO: SEGURADORAS

FORMATO: 9(4)-9

DOMINIO: 0001 A 9999

COD SWIFT BCO EXT

DESCRICAO: IDENTIFICACAO DE BANCO NO EXTERIOR NO SISTEMA SWIFT

REGRA VALIDAC:

FORMATO: X(16)

DOMINIO:

COD TIPO ENCARGO

DESCRICAO: COD IDENTIFICADOR DO TIPO DE ENCARGO

REGRA VALIDAC: CONFORME TABELA TIPO DE ENCARGO

FORMATO: 9(2)

DOMINIO: DE "1" A "99"

COD TIPO GARANTIA OBTIDA

DESCRICAO: INDICA O TIPO DE GARANTIA OBTIDA (PROMISSORIA, ALIENACAO FIDUCIARIA, ETC)

REGRA VALIDAC: CONFORME CADASTRO: TIPO DE GARANTIA OBTIDA

FORMATO: 9(2)

DOMINIO:

COD TRANSPORTADORA

DESCRICAO: CODIGO IDENTIFICADOR DA TRANSPORTADORAS

REGRA VALIDAC: CONFORME CADASTRO: TRANSPORTADORAS

FORMATO: 9(4)-9

DOMINIO: 0001 A 9999

CONDICOES DE COBRANCA

DESCRICAO: DESCREVE A "QUALIDADE" DO PROCESSO DE COBRANCA DE UM BANCO NO EXTERIOR

REGRA VALIDAC:

FORMATO: X(120)

DOMINIO:

CONDICOES NEGOCIACAO DE TRAVELLER'S CHECKS

DESCRICAO: DESCREVE AS CONDICOES ESPECIFICAS PARA TRANSACIONAR TRAVELLER'S CHECKS DE UM BANCO NO EXTERIOR

REGRA VALIDAC:

FORMATO: X(100)

DOMINIO:

CONFIRMADA (S/N)

DESCRICAO: INDICADOR SE A CARTA DE CREDITO DE IMPORTACAO OU EXPORTACAO FOI OU NAO CONFIRMADA

REGRA VALIDAC: 0 OU 1

FORMATO: 9(1)

DOMINIO: 0="CONFIRMADA" 1="NAO CONFIRMADA"

CONVENIADO (S/N)

DESCRICAO: INDICA SE O BANCO NO EXTERIOR PARTICIPA DE CONVENIO BILATERAL COM
O BANCO CENTRAL DE SEU PAIS

REGRA VALIDAC: 0 OU 1

FORMATO: 9(1)

DOMINIO: 0="S" 1="N"

DATA CONCESSAO ADIANTAMENTO

DESCRICAO: ESPECIFICA A DATA DE CONCESSAO DO ADIANTAMENTO

REGRA VALIDAC: DATA VALIDA

FORMATO: 9(6)

DOMINIO: AAMMDD

DATA CONFIRMACAO

DESCRICAO: DATA DA CONFIRMACAO DE UMA CARTA DE CREDITO DE IMPORTACAO OU EXPO
RTACAO

REGRA VALIDAC: DATA DE CONFIRMACAO<=DATA DE ABERTURA DA CARTA DE CREDITO

FORMATO: 9(6)

DOMINIO: AAMMDD

DATA EMBARQUE

DESCRICAO: DATA DO EMBARQUE DAS MERCADORIAS

REGRA VALIDAC: DATAS VALIDAS

FORMATO: 9(6)

DOMINIO: AAMMDD

DATA INSERCAO NO EXTRATO

DESCRICAO: DATA DE INSERCAO NO EXTRATO, RECEBIDO DE UM BANCO NO EXTERIOR, DE
UM DEBITO OU CREDITO

REGRA VALIDAC: DATAS VALIDAS

FORMATO:

DOMINIO:

DATA LIMITE VALIDADE

DESCRICAO: DATA LIMITE DE VALIDADE DA CARTA DE CREDITO

REGRA VALIDAC: DATA VALIDA

FORMATO: 9(6)

DOMINIO: AAMMDD

DATA NEGOCIACAO

DESCRICAO: DATA DE NEGOCIACAO DA CARTA DE CREDITO

REGRA VALIDAC: DATAS VALIDAS

FORMATO: 9(6)

DOMINIO: AAMMDD

DATA NEGOCIACAO

DESCRICAO:

REGRA VALIDAC:

FORMATO:

DOMINIO:

DATA RECEPCAO DOCTOS

DESCRICAO: DATA DE RECEBIMENTO , PELO BFB, DE DOCUMENTOS QUALISQUER

REGRA VALIDAC: DATAS VALIDAS

FORMATO:

DOMINIO:

DATA REMESSA DOCTOS

DESCRICAO: DATA DE REMESSA DE DOCTOS AO EXTERIOR, VIA MALOTE

REGRA VALIDAC: DATAS VALIDAS

FORMATO:

DOMINIO:

DATA SOLICITACAO CLIENTE P/ SOLUCAO DISCREPANCIA

DESCRICAO: DATA EM QUE FOI SOLICITADO AO EXPORTADOR OU IMPORTADOR UMA SOLUCAO PARA DISCREPANCIA ENTRE OS DOCUMENTOS E A CARTA DE CREDITO

REGRA VALIDAC: DATAS VALIDAS

FORMATO:

DOMINIO:

DATA VALOR

DESCRICAO: DATA DE RECONHECIMENTO DE UM VALOR. PODE SER RETROATIVA

REGRA VALIDAC: DATAS VALIDAS

FORMATO:

DOMINIO:

DATA VENCIMENTO

DESCRICAO: DATA DE VENCIMENTO DE UM COMPROMISSO, CONTRATO OU DOCUMENTO QUALISQUER

REGRA VALIDAC: DATAS VALIDAS

FORMATO:

DOMINIO:

DEBITO/CREDITO

DESCRICAO: INDICA SE O LANCAMENTO E A DEBITO (CONTRA O BFB) OU A CREDITO (A FAVOR DO BFB)

REGRA VALIDAC: 0 OU 1

FORMATO: 9(1)

DOMINIO: 0=DEBITADO/1=CREDITADO

DISCREPANCIA (S/N)

DESCRICAO: INDICA SE A CARTA DE CREDITO APRESENTOU DISCREPANCIA COM OS DOCUMENTOS RELATIVOS A MERCADORIA EMBARCADA

REGRA VALIDAC: 0 OU 1

FORMATO: 9(1)

DOMINIO: 0=SEM DISCREPANCIA/1=COM DISCREPANCIA

EMENDADA (S/N)

DESCRICAO: INDICA SE A CARTA DE CREDITO FOI OU NAO EMENDADA

REGRA VALIDAC: 0 OU 1

FORMATO: 9(1)

DOMINIO: 0=SEM EMENDA/1=COM EMENDA

ENDERECO BANCO NO EXTERIOR

DESCRICAO: ENDERECO DO BANCO NO EXTERIOR

REGRA VALIDAC:
FORMATO: X(100)
DOMINIO:

ENDERECO CLIENTE

DESCRICAO: ENDERECO DO CLIENTE

REGRA VALIDAC:
FORMATO: X(100)
DOMINIO:

ENDERECO SACADO OU SACADOR

DESCRICAO: ENDERECO DO SACADO OU SACADOR DE UMA CAMBIAL

REGRA VALIDAC:
FORMATO: X(100)
DOMINIO:

ENDERECO TELEGRAFICO

DESCRICAO: ENDERECO TELEGRAFICO DO BANCO NO EXTERIOR

REGRA VALIDAC:
FORMATO: X(15)
DOMINIO:

ENVIO DE DOCTOS SOB RESERVA (S/N)

DESCRICAO:

REGRA VALIDAC: 0 OU 1
FORMATO: 9(1)
DOMINIO: 0=SEM RESERVA/1=COM RESERVA

INDICADOR FASE ADTO

DESCRICAO: INDICA SE O ADIANTAMENTO DE EXPORTACAO E SOBRE "LETRAS A ENTREGAR" (ACC) OU "LETRAS ENTREGUES" (ACE)

REGRA VALIDAC: 0 OU 1
FORMATO: 9(1)
DOMINIO: 0=ACC/1=ACE

INDICADOR IMPORTACAO/EXPORTACAO

DESCRICAO: INDICA SE O AGENTE E INTERMEDIARIO DE IMPORTACOES, EXPORTACOES OU AMBAS

REGRA VALIDAC: 0, 1 OU 2
FORMATO: 9(1)
DOMINIO: 0=AGENTE DE IMPORTACAO/1=AGENTE DE EXPORTACAO/3=AMBOS

INDICATIVO DE RECEPCAO (RECEBIDO/ A RECEBER)

DESCRICAO: INDICA SE A CAMBIAL JA FOI RECEBIDA PELO BFB OU AINDA DEVERA SE-L

REGRA VALIDAC: 0 OU 1
FORMATO: 9(1)
DOMINIO: 0=AINDA NAO RECEBIDA/1=JA RECEBIDA

LIBERACAO DA RESERVA (S/N)

DESCRICAO: INDICA SE UMA CARTA DE CREDITO NEGOCIADA SOB RESERVA FOI LIBERADA
OU NAO

REGRA VALIDAC: 0 OU 1

FORMATO: 9(1)

DOMINIO: 0=NAO LIBERADA/1=LIBERADA

NOME

DESCRICAO: USADO PARA DESIGNAR UMA PESSOA, EMPRESA OU OBJETO

REGRA VALIDAC:

FORMATO: X(35)

DOMINIO:

NOME ABREVIADO

DESCRICAO: ABREVIATURA DO NOME REFERENCIADO

REGRA VALIDAC:

FORMATO: X(15)

DOMINIO:

NOME CONTATO

DESCRICAO: NOME DO FUNCIONARIO DE BANCO NO EXTERIOR COM O QUAL TEMOS USUALME
NTE CONTATO

REGRA VALIDAC:

FORMATO: X(35)

DOMINIO:

NOME SACADO OU SACADOR

DESCRICAO: NOME DO SACADO OU SACADOR DE UMA CAMBIAL

REGRA VALIDAC:

FORMATO: X(35)

DOMINIO:

NR CONTA CORRENTE EXTERIOR

DESCRICAO: NR DA CONTA CORRENTE JUNTO A BANQUEIRO NO EXTERIOR

REGRA VALIDAC:

FORMATO: X(13)

DOMINIO:

NR CONTRATO DE GARANTIA DE IMPORTACAO

DESCRICAO: E O NUMERO DO CONTRATO DE GARANTIA DE PAGTO DE IMPORTACAO EMITIDO
PELO BFB A BENEFICIARIOS NO EXTERIOR

REGRA VALIDAC:

FORMATO:

DOMINIO:

NR LINHA CREDITO

DESCRICAO: NUMERO QUE IDENTIFICA A LINHA DE CREDITO CONCEDIDA AO BFB

REGRA VALIDAC: CONFORME ARQUIVO LINHA DE CREDITO

FORMATO: 9(6)

DOMINIO:

NR OPERACAO

DESCRICAO: NUMERO IDENTIFICADOR DO FATO, COMPOSTO, INTERNO DO SISTEMA

REGRA VALIDAC: COD AGENCIA CB/COD CLIENTE/NR OPER. RAIZ/NR DOCTO/COD TIPO DOCTO
 FORMATO: VER DOMIN
 DOMINIO: 9(3)/9(5)/9(3)/9(3)/9(2)

NR OPERACAO-REFERENCIA

DESCRICAO: NUMERO QUE INDICA A OPERACAO RELACIONADA COM A OPERACAO PRINCIPAL

REGRA VALIDAC:
 FORMATO: VER DOMIN
 DOMINIO: 9(3)/9(5)/9(3)/9(3)/9(2)

NR OPERACAO-REFERENCIA 1

DESCRICAO: NUMERO QUE IDENTIFICA A OPERACAO RELACIONADA COM A OPERACAO PRINCIPAL (EXISTE MAIS DE 1 OPERACAO REFERENCIA)

REGRA VALIDAC:
 FORMATO: VER DOMIN
 DOMINIO: 9(3)/9(5)/9(3)/9(3)/9(2)

NR REFERENCIA BANCO NO EXTERIOR

DESCRICAO: NUMERO ATRIBUIDO PELO BANCO NO EXTERIOR PARA IDENTIFICACAO DE UMA OCORRENCIA

REGRA VALIDAC:
 FORMATO: X(11)
 DOMINIO:

NR REMESSA

DESCRICAO: NUMERO EMITIDO PELO BFB IDENTIFICADOR DA REMESSA(LOTE), DE DOCUMENTOS OU VALORES AO EXTERIOR

REGRA VALIDAC:
 FORMATO: 9(6)
 DOMINIO:

NR TELEX I

DESCRICAO: NUMERO DO TELEX DO BANCO NO EXTERIOR

REGRA VALIDAC:
 FORMATO: X(15)
 DOMINIO:

NR TELEX II

DESCRICAO: NUMERO DO TELEX DO BANCO NO EXTERIOR (SE EXISTIR MAIS DE UM)

REGRA VALIDAC:
 FORMATO: X(15)
 DOMINIO:

NR TELEX III

DESCRICAO: NUMERO DO TELEX DO BANCO NO EXTERIOR (SE EXISTIR MAIS DE DOIS)

REGRA VALIDAC:
 FORMATO: X(15)
 DOMINIO:

ORIGEM DO AVISO

DESCRICAO: IDENTIFICA A ORIGEM DE RECEBIMENTO DE AVISO DE BANQUEIRO NO EXTERIOR (SWIFT,DHL ETC...)

REGRA VALIDAC:

FORMATO: X(10)

DOMINIO:

PENDENTE/NAO PENDENTE

DESCRICAO: DISCRIMINA SE O MONTANTE REFERENTE AO AVISO DE DEBITO OU CREDITO RECEBIDO DO BANCO NO EXTERIOR FOI OU NAO APLICADO

REGRA VALIDAC:P OU N

FORMATO: X(1)

DOMINIO: P=PENDENTE/N=NAO PENDENTE

PERCENTUAL DE ADIANTAMENTO NO FECHTO

DESCRICAO: IDENTIFICA O PERCENTUAL DE ADTO CONCEDIDO AO CLIENTE NO FECHTO DE UMA OPERACAO DE EXPORTACAO

REGRA VALIDAC:

FORMATO: 9(3)

DOMINIO: "0 A 999"

PERCENTUAL RED CLAUSE

DESCRICAO: IDENTIFICA O PERCENTUAL RED CLAUSE CONCEDIDO EM UMA CARTA DE CREDITO DE EXPORTACAO OU IMPORTACAO

REGRA VALIDAC:

FORMATO: 9(3)

DOMINIO: "0 A 999"

RED CLAUSE (S/N)

DESCRICAO: INDICA A EXISTENCIA OU NAO DE RED CLAUSE EM UMA CARTA DE CREDITO DE EXPORTACAO OU IMPORTACAO

REGRA VALIDAC:S OU N

FORMATO: X(1)

DOMINIO: S=SIM/N=NAO

REVOGAVEL (S/N)

DESCRICAO: INDICA SE A CARTA DE CREDITO E OU NAO REVOGAVEL

REGRA VALIDAC:0 OU 1

FORMATO: 9(1)

DOMINIO: 0=REVOGAVEL/1=IRREVOGAVEL

SALDO A LIQUIDAR M/E

DESCRICAO: INDICA O SALDO EM M/E NA RESPECTIVA OPERACAO QUE FALTA SER LIQUIDADO

REGRA VALIDAC:

FORMATO: 9(11)U99

DOMINIO:

SALDO A LIQUIDAR M/N

DESCRICAO: INDICA O SALDO EM M/N NA RESPECTIVA OPERACAO QUE FALTA SER LIQUIDADO

REGRA VALIDAC:IGUAL AO PRODUTO DE SALDO A LIQUIDAR M/E PELA TAXA CAMBIAL

FORMATO: 9(11)U99

DOMINIO:

SALDO MEDIO PACTUADO

DESCRICAO: SALDO MEDIO PACTUADO COM BANCO NO EXTERIOR COMO RECIPROCIDADE

REGRA VALIDAC:
FORMATO: 9(11)U99
DOMINIO:

SALDO SAQUES A VINCULAR

DESCRICAO: SALDO DOS SAQUES A VINCULAR COM OS CONTRATOS DE EXPORTACAO

REGRA VALIDAC:
FORMATO: 9(11)U99
DOMINIO:

STATUS DO CLIENTE

DESCRICAO: CODIGO QUE INDICA A SITUACAO ATUAL DO CLIENTE (CONCORDATARIO, FALIDO ETC...)

REGRA VALIDAC: CONFORME TABELA STATUS CLIENTE
FORMATO: 9(1)
DOMINIO:

TAXA BASE PARA CALCULO

DESCRICAO: PERCENTUAL ESTABELECIDO EM UM DETERMINADO PERIODO PARA CALCULOS DO VALOR DA RECEITA OU DESPESA ESTABELECIDADA PELO BFB

REGRA VALIDAC:
FORMATO: 9(3)U99
DOMINIO: AA/MM

TIPO TRANSPORTE

DESCRICAO: TIPO DE TRANSP. DA MERCADORIA ACOMPANHADA DE SUA RESPECTIVA CAMBIAL OU CARTA REMESSA (EX: AEREO, MARITIMO, TERRESTRE, FERROVIARIO)

REGRA VALIDAC:
FORMATO: X(12)
DOMINIO:

TRANSFERIVEL (S/N)

DESCRICAO: INDICA SE A CARTA DE CREDITO PODE OU NAO SER TRANSFERIDA PARA OUTRO BENEFICIARIO

REGRA VALIDAC: 0 OU 1
FORMATO: 9(1)
DOMINIO: 0=TRANSFERIVEL/1=INTRANSFERIVEL

VALOR

DESCRICAO: MONTANTE NUMERICO QUALQUER

REGRA VALIDAC:
FORMATO: 9(13)U99
DOMINIO:

VALOR BASE P/ CALCULO

DESCRICAO: VALOR UTILIZADO PARA BASE DE CALCULO DOS ENCARGOS REPASSADOS AOS CLIENTES

REGRA VALIDAC:
FORMATO: 9(9)U99
DOMINIO:

VALOR COMISSAO AGENTE M/E

DESCRICAO: REPRESENTA O VALOR A SER PAGO NO PAIS AO AGENTE INTERMEDIARIO DE
UMA OPERACAO DE IMPORTACAO OU EXPORTACAO

REGRA VALIDAC:

FORMATO: 9(11)U99

DOMINIO:

VALOR EMBARCADO M/E

DESCRICAO: VALOR EM MOEDA ESTRANGEIRA DA MERCADORIA EMBARCADA CONSTANTE NA C
AMBIAL OU CARTA REMESSA

REGRA VALIDAC:

FORMATO: 9(11)U99

DOMINIO:

VALOR EMBARCADO M/N

DESCRICAO: VALOR EM MOEDA NACIONAL DA MERCADORIA EMBARCADA CONSTANTE NA C
AMBIAL OU CARTA REMESSA

REGRA VALIDAC:

FORMATO: 9(11)U99

DOMINIO:

VALOR FOB M/E

DESCRICAO: VALOR EM M/E CORRESPONDENTE AO FOB DOS CONTRATOS DE CAMBIO E CART
AS DE CREDITO DE IMPORTACAO E EXPORTACAO

REGRA VALIDAC:

FORMATO: 9(11)U99

DOMINIO:

VALOR FOB M/E LIQUIDADO

DESCRICAO: E O VALOR DO FOB EM M/E LIQUIDADO NO PAIS PARA OS CONTRATOS DE CA
MBIO DE IMPORTACAO OU EXPORTACAO

REGRA VALIDAC:

FORMATO: 9(13)U99

DOMINIO:

VALOR FRETE M/E

DESCRICAO: VALOR EM M/E CORRESPONDENTE AO FRETE DAS MERCADORIAS DOS CONTRATO
S DE CAMBIO E CARTAS DE CREDITO DE IMPORTACAO E EXPORTACAO

REGRA VALIDAC:

FORMATO: 9(8)U99

DOMINIO:

VALOR FRETE M/E LIQUIDADO

DESCRICAO: E O VALOR DO FRETE EM M/E LIQUIDADO NO PAIS PARA OS CONTRATOS DE
CAMBIO DE IMPORTACAO OU EXPORTACAO

REGRA VALIDAC:

FORMATO: 9(8)U99

DOMINIO:

VALOR I.R. S/ ENCARGO

DESCRICAO: VALOR REFERENTE AO IMPOSTO DE RENDA DEVIDO S/ PAGTO DE ENCARGOS

REGRA VALIDAC:

FORMATO: 9(8)U99

DOMINIO:

VALOR LIQUIDADO M/E

DESCRICAO: VALOR TOTAL OU PARCIAL DA M/E DO CONTRATO DE CAMBIO LIQUIDADO NO PAIS

REGRA VALIDAC:

FORMATO: 9(11)U99

DOMINIO:

VALOR LIQUIDADO M/N

DESCRICAO: VALOR TOTAL OU PARCIAL DA M/N DO CONTRATO DE CAMBIO LIQUIDADO NO PAIS

REGRA VALIDAC:

FORMATO: 9(13)U99

DOMINIO:

VALOR M/E

DESCRICAO: VALOR REAL DA TRANSACAO EM MOEDA ESTRANGEIRA

REGRA VALIDAC:

FORMATO: 9(11)U99

DOMINIO:

VALOR M/E JUROS L/C FINANCIADA

DESCRICAO: E O VALOR DO JUROS MENCIONADO NA L/C DE IMPORTACAO/EXPORTACAO P/ OPERACOES C/ PRAZO P/ LIQUIDACAO SUPERIOR A 180 DIAS DO EMBARQUE

REGRA VALIDAC:

FORMATO: 9(8)U99

DOMINIO:

VALOR M/N

DESCRICAO: VALOR REAL DA TRANSACAO EM MOEDA NACIONAL

REGRA VALIDAC:

FORMATO: 9(13)U99

DOMINIO:

VALOR PREVISTO DO EMBARQUE

DESCRICAO: E O VALOR EM M/E DA MERCADORIA PARA CADA EMBARQUE, PREVISTO QUANDO DO FECHTO DE CONTRATO DE CAMBIO DE EXPORTACAO

REGRA VALIDAC:

FORMATO: 9(11)U99

DOMINIO:

VALOR SEGURO M/E

DESCRICAO: VALOR EM M/E CORRESPONDENTE AO SEGURO DAS MERCADORIAS DOS CONTRATOS DE CAMBIO DE IMPORTACAO OU EXPORTACAO E L/C DE EXPORTACAO

REGRA VALIDAC:

FORMATO: 9(8)U99

DOMINIO:

VALOR SEGURO M/E LIQUIDADO

DESCRICAO: E O VALOR DO SEGURO EM M/E LIQUIDADO NO PAIS, PARA OS CONTRATOS DE CAMBIO DE IMPORTACAO OU EXPORTACAO

REGRA VALIDAC:

FORMATO: 9(8)U99

DOMINIO:

MODELO DINAMICO: SEQUENCIA DOS EVENTOS POR OPERACAO

NUM EVENTO
ERO
EVE
NTO

NUM EVENTO SEGUINTE
ERO
EVE
NTO

** NUMERO DO PRODUTO: 2

1 RECEBTO DE CARTA DE CREDITO
12 AVISO AO CLIENTE S/RECEBTO DE CARTA DE CREDITO
12 AVISO AO CLIENTE S/RECEBTO DE CARTA DE CREDITO
12 AVISO AO CLIENTE S/RECEBTO DE CARTA DE CREDITO
13 CONFIRMACAO DE CARTA DE CREDITO
13 CONFIRMACAO DE CARTA DE CREDITO
162 FECHTO DE CONTRATO DE CAMBIO DE EXPORTACAO
14 CONCESSAO DE ADIANTAMENTO AO CLIENTE
162 FECHTO DE CONTRATO DE CAMBIO DE EXPORTACAO
162 FECHTO DE CONTRATO DE CAMBIO DE EXPORTACAO
16 RECEBTO DE DOCTOS/CAMBIAIS
17 CONFERENCIA DE DOCUMENTOS
24 DEVOLUCAO DE ADIANTAMENTO DO CLIENTE
17 CONFERENCIA DE DOCUMENTOS
10 REMESSA DE DOCTOS/CAMBIAIS AO BANQUEIRO
27 RECEBTO DE AVISO DE CREDITO DO BANQUEIRO

12 AVISO AO CLIENTE S/RECEBTO DE CARTA DE CREDITO
3 FIM
13 CONFIRMACAO DE CARTA DE CREDITO
162 FECHTO DE CONTRATO DE CAMBIO DE EXPORTACAO
3 FIM
162 FECHTO DE CONTRATO DE CAMBIO DE EXPORTACAO
14 CONCESSAO DE ADIANTAMENTO AO CLIENTE
15 RECEBTO DE DESAGIO DO CLIENTE
360 VISAR GUIA
16 RECEBTO DE DOCTOS/CAMBIAIS
17 CONFERENCIA DE DOCUMENTOS
24 DEVOLUCAO DE ADIANTAMENTO DO CLIENTE
20 LIQUIDACAO DE CONTRATO DE CAMBIO
10 REMESSA DE DOCTOS/CAMBIAIS AO BANQUEIRO
27 RECEBTO DE AVISO DE CREDITO DO BANQUEIRO
245 LIQUIDACAO DE CARTA DE CREDITO

Listagens Resultantes da Aplicação do MIND-AID.

T I P O S :

nome	estrutura	coment	validacao	valores(em baixo)
CODFORMA	simples	ComCodForm	tabela(FORMENTR)	
NROPERACAO DCLI	registro	ComNrOper	nil	
			NROPERRAIZ	NRDOCTO
			CODTIPODOCTO	
CODAGEN	simples	nil	tabela(AGENCIAS)	
CODCLI	simples	nil	nil	
NROPERRAIZ	simples	nil	nil	
NRDOCTO	simples	nil	nil	
CODTIPODOCTO	simples	nil	tabela(TIPODOCTO)	
VALOR	simples	nil	nil	
OPCAO	simples	nil	valores(S ou N)	
ENDERECO	simples	nil	nil	
NOME	simples	nil	nil	
CODBANCO R	registro	nil	nil	
NRBCO	simples	nil	nil	
DCONTR	simples	nil	nil	
PERCENTUAL	simples	nil	nil	
DATA	registro	nil	nil	
			ANO	
DIA	simples	nil	intervalo(1..31)	
MES	simples	nil	intervalo(1..12)	
ANO	simples	nil	nil	
NUMERICO	simples	nil	nil	
ALFA	simples	nil	nil	

A T R I B U T O S :

nome	tipo	validacao	coment
COD FORMA ENTR M/N AO BFBCODFORMA		nil	nil
COD FORMA ENTR M/N PELO BFBCODFORMA		nil	nil
DATA CONCES ADIANTO	DATA	nil	nil
DATA VENCTO	DATA	nil	nil
INDIC ADTO	NUMERICO	nil	nil
INDIC CRED LIQUID	OPCAO	nil	nil
NR OPERACAO	NROPERACAO	nil	nil
NR OPER REF	NROPERACAO	nil	nil
SALDO A LIQ M/E	VALOR	nil	nil
SALDO A LIQ M/N	VALOR	nil	nil
VALOR M/N	VALOR	nil	nil
COD CLIENTE	CODCLI	nil	nil
ENDEREÇO CLI	ENDEREÇO	nil	nil
NOME	NOME	nil	nil
NOME ABREVIADO	NOME	nil	nil
STATUS CLI	NUMERICO	tabela(STATUS CLIENTE)	nil
COD MODAL TRANSACAO	NUMERICO	tabela(COD TRANSACAO)	nil
NR LINHA CRED	NUMERICO	tabela(LINHA CRED)	nil
PERCENT ADTO FECHTO	PERCENTUAL	nil	nil
SALDO SAQUES A VINCULAR	VALOR	nil	nil

VALOR FOB M/E	VALOR	nil	nil
VALOR FRETE M/E	VALOR	nil	nil
VALOR SEGURO M/E	VALOR	nil	nil
DATA EFETIVA LIQUID	DATA	nil	nil
NR OPER REF1	NROPERACAO	nil	nil
VALOR COMIS AGENTE M/E	VALOR	nil	nil
VALOR FOB M/E LIQUIDD	VALOR	nil	nil
VALOR FRETE M/E LIQUIDD	VALOR	nil	nil
VALOR LIQUIDD M/E	VALOR	nil	nil
VALOR LIQUIDD M/N	VALOR	nil	nil
VALOR SEGURO M/E LIQUIDD	VALOR	nil	nil
DATA AVISO	DATA	nil	nil
COD BANCO EXT	CODBANCO	tabela(COD BCO EXTERIOR)	nil
COD MOEDA	NUMERICO	tabela(COD MOEDA)	nil
DATA INSCRC EXTRATO	DATA	nil	nil
DATA RECEBTO AVISO	DATA	nil	nil
DATA VALOR	DATA	nil	nil
DEBITO/CRED	ALFA	valores(D ou C)	nil
NR CONTA CORR EXT	NUMERICO	tabela(CONTA CORR EXT)	nil
NR ORIGEM	NUMERICO	nil	nil
NR REF BCO EXT	NUMERICO	tabela(NR REFER BCO EXT)	nil
ORIGEM AVISO	ALFA	nil	nil

PENDENTE/NAO PEND	ALFA	valores(N ou P)	nil
VALOR M/E	VALOR	nil	nil
COD PAIS DESTINO	NUMERICO	tabela(COD PAIS)	nil
COD SEGURADORA	NUMERICO	tabela(SEGURADORA)	nil
COD TRANSPORT	NUMERICO	tabela(TRANSPORTADORA)	nil
DATA RECEBTO CAMBIAL	DATA	nil	nil
DATA DESEMBOLSO	DATA	nil	nil
DATA EMBARQUE	DATA	nil	nil
AREA CONTATO	ALFA	nil	nil
COD INSTITUICAO	NUMERICO	tabela(INSTIT FINANC EXT)	nil
COD SWIFT BCO EXT	ALFA	nil	nil
CONDICOES COBRANCA	ALFA	nil	nil
CONTA CORRENTE	OPCAO	nil	nil
CONVENIADO	OPCAO	nil	nil
ENDEREÇO BCO EXT	ENDEREÇO	nil	nil
ENDEREÇO TELEGR	ENDEREÇO	nil	nil
NOME CONTATO	NOME	nil	nil
NR TELEX	NUMERICO	nil	nil
SALDO MEDIO PACTUADO	VALOR	nil	nil
COD PAIS DESTINO	NUMERICO	tabela(COD PAIS)	nil
COD SEGURADORA	NUMERICO	tabela(SEGURADORA)	nil
COD TRANSPORT	NUMERICO	tabela(TRANSPORTADORA)	nil

DATA RECEBTO CAMBIAL	DATA	nil	nil
DATA DESEMBOLSO	DATA	nil	nil
DATA EMBARQUE	DATA	nil	nil
DATA RECEBTO DOCTOS	DATA	nil	nil
DATA REMESSA DOCTOS	DATA	nil	nil
ENDERECO SACADO/R	ENDERECO	nil	nil
NOME SACADO/R	NOME	nil	nil
NR REMESSA	NUMERICO	nil	nil
VALOR EMBARCADO	VALOR	nil	nil
COD BCO AVISADOR	CODBANCO	nil	nil
COD BCO CONFIRM	CODBANCO	nil	nil
COD BCO INSTITUIDOR	CODBANCO	nil	nil
COD BCO NEGOCIADOR	CODBANCO	nil	nil
COD TIPO GARANTIA	NUMERICO	nil	nil
CONFIRMADA	OPCAO	nil	nil
DATA ABERTURA	DATA	nil	nil
DATA AVISO BENEF	DATA	nil	nil
DATA CONFIRMACAO	DATA	nil	nil
COD TIPO ENCARGO	NUMERICO	nil	nil
DATA PREV P/RECEBTO	DATA	nil	nil
INDIC CRED LIQUID	OPCAO	nil	nil
TAXA BASE P/CALCULO	PERCENTUAL	nil	nil

VALOR PGTO	VALOR	nil	nil
VALOR BASE P/CALCULO	VALOR	nil	nil
VALOR IR S/ENCARGO	VALOR	nil	nil
NR CONTA CORR EXT	NUMERICO	nil	nil
DATA LIBERACAO RESERV	DATA	nil	nil
DATA NEGOCIACAO	DATA	nil	nil
DATA SOLICITAC SOL.DISCREPDATA		nil	nil
DISCREPANCIA	OPCAO	nil	nil
ENVIO DOCTOS S/RESERVA	OPCAO	nil	nil
LIBERACAO RESERVA	OPCAO	nil	nil
NOME BENEFICIARIO	NOME	nil	nil
ENDEREÇO BENEFIC	ENDEREÇO	nil	nil
CONFIRMAC C.CRED	OPCAO	nil	nil
FECHTO CONTRATO	OPCAO	nil	nil

C L O B J E T O S

codigo nome criacao supressao atributos

0175 ADIANTO CONCEDIDO881018 0

NrOperacao

CodFormaEntrM

. N

. AoBFB

CodFormaEntrM .

N .

PeloBFB DataConcesAdianto

DataVencdo IndicAdto

IndicCredLiquid NrOperRef

SaldoALiqM .

E .

SaldoALiqM .

N .

ValorM .

N .

0315 CLIENTE 881018 0

CodCliente

EnderecoCli

NOME NomeAbreviado

StatusCli

02 CONTRATO CAMBIO881018 0

NrOperacao

CodModalTransacao

NrLinhaCred PercentAdtoFechto

SaldoSaquesAVincular ValorFobM

. E

. ValorFreteM

. E

. ValorSeguroM

0545 LIQUIDACAO CONTRATO881018 0

. E

.

NrOperacao

DataEfetivaLiquid

NrOperRef NrOperRef1

ValorComisAgenteM .

E .

ValorFobM .

E .

Liquidado ValorFreteM

. E

. Liquidado

ValorLiquiddM .

E .

ValorLiquiddM .

N .

ValorSeguroM .

ELiquidd

065 AVISO DEB/CRED RECEB881018 0

DataAviso

CodBancoExt

CodMoeda DataInsercExtrato

DataRecebtoAviso DataValor

DEBITO -

CRED NrContaCorrExt

NrOperRef NrOrigem

NrRefBcoExt OrigemAviso

PENDENTE -

NaoPEND ValorM

. E

.

040 CAMBIAL OU C.REMESSA881019 0

NrOperacao

CodSeguradora CodTransport
 DataRecebitoCambial DataDesembolso
 DataEmbarque DataRecebitoDoctos
 DataRemessaDoctos EnderecoSacado
 - r
 NomeSacado -
 r NrRemessa
 ValorEmbarcado

025 PEDIDO CLIENTE 881019 0

NrOperacao
 NomeBeneficiario
 ConfirmacO .
 Cred FechtoContrato

070 AVISO 881019 0

NrOperacao
 NomeBeneficiario
 EnderecoBenefic

0680 C.CREDITO-NEGOCIACA0881019 0

NrOperacao
 DataLiberacaoReserv
 DataNegociacao DataSolicitacSol
 . Discrep
 DISCREPANCIA EnvioDoctosS
 . Reserva
 LiberacaoReserva NrOperRef

0565 ENCARGOS REG CAIXA881019 0

NrOperacao
 CodMoeda
 CodTipoEncargo DataPrevP
 . Recebto
 IndicCredLiquid TaxaBaseP
 . Calculo
 ValorPgto ValorBaseP
 . Calculo

R .
S .
Encargo NrContaCorrExt
CodFormaEntrM .
N .
AoBFB

0685 CARTA DE CREDITO881019 0

NrOperacao

NomeBenefic
EnderecoBenef CodBcoAvisador
CodBcoConfirm CodBcoInstituidos
CodBcoNegociador CodMoeda
CodTipoGarantia CONFIRMADA
DataAbertura DataAvisoBenef
DataConfirmacao ValorM
/ E

0195 BANCO NO EXTERIOR881019 0

CodBancoExt

AreaContato
CodInstituicao CodSwiftBcoExt
CondicoesCobranca ContaCorrente
Conveniado EnderecoBcoExt
EnderecoTelegr NOME
NomeContato NrTelex
SaldoMedioPactuado

C H A V E S :

nome-objeto :

atributos-chaves :

PEDIDO CLIENTE

NR OPERACAO

AVISO

NR OPERACAO

C.CREDITO-NEGOCIACAO

NR OPERACAO

ENCARGOS REG CAIXA

NR OPERACAO

CARTA DE CREDITO

NR OPERACAO

CAMBIAL OU C.REMESSA

NR OPERACAO

BANCO NO EXTERIOR

COD BANCO EXT

ADIANTO CONCEDIDO

NR OPERACAO

CLIENTE

COD CLIENTE

CONTRATO CAMBIO

NR OPERACAO

LIQUIDACAO CONTRATO

NR OPERACAO

AVISO DEB/C RECEBIDO

DATA AVISO

C-EVENTOS

codigo :	nome :	coment :	criacao	supressao
E1751	CONCESS ADIANTO	nil	881020	0
E21	FECHTO CONTRATO	nil	881020	0
E651	RECEBTO CREDITO	nil	881020	0
E401	RECEBTO CAMBIAIS	nil	881020	0
E251	RECEBTO PEDIDO CLI	nil	881020	0
E252	REC PED FECHTO CONTR	nil	881020	0
E6801	DOCTOS S/DISCREPANC.	nil	881020	0
E6851	RECEBTO C.CREDITO	nil	881020	0

C-CONSTATAS

evento	origem	aceit.	consultados	criacao	supressao
E251	025	RE251	025	881020	0
E252	025	RE252	025	881020	0
E6801	0680	RE6801	0680	881020	0
E6851	0685	RE6851	0685	881020	0
E1751	0175	RE1751	0175	881020	0
E21	02	RE21	02	881020	0
E651	065	RE651	065	881020	0
E401	040	RE401	040	881020	0

C-ACOES

nome	tipo	alvo	consultados	texto	coment	criacao	supressao
RECEBTO DESAGIO CLI	inclusao	0565	0175	0565	A17511	ComA17511	881020 0
CONCES ADTO CLI	inclusao	0175	0175	02	A211	ComA211	881020 0
VINCULACAO	alteracao	0685	0685	02	A212	ComA212	881020 0
LIQUIDACAO C.CREDITO	alteracao	0680	0680	065	A6511	ComA6511	881020 0
CONFERENCIA DOCTOS	inclusao	0680	0680	040	A4011	ComA4011	881020 0
VINCULACAO	alteracao	02	02	040	A4012	ComA4012	881020 0
CONFIRMAC C.CREDITO	inclusao	0685	0685	025	A2511	ComA2511	881020 0
FECHTO CONTRATO	inclusao	02	02	025	A2521	ComA2521	881020 0
LIQUIDACAO CONTRATO	inclusao	0545	0545	0680	A6801	ComA68011	881020 0
ENVIO DOCTOS BCO EXT	alteracao	0195	0195	0680	A68012	ComA68012	881020 0
DEVOLUCAO ADTO CLI	alteracao	0175	0175	0680	A68013	ComA68013	881020 0
AVISO CLIENTE	inclusao	070	070	0685	A68511	ComA68511	881020 0

C-DISPARAS

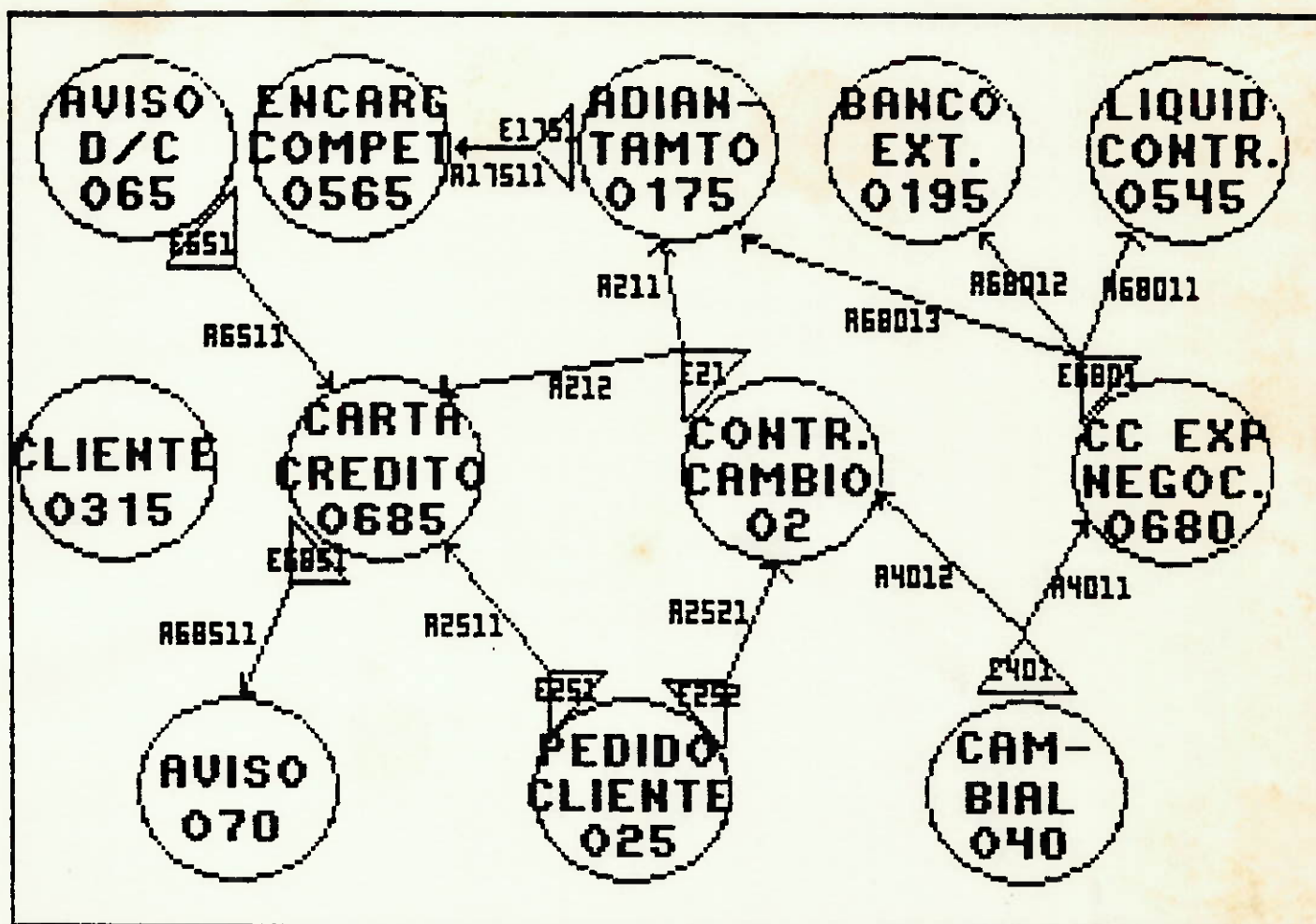
evento acao alvo aceit. consultados abrang consultados criacao supressao

E1751	17511	0565	nil	0175	nil	0565	881020	0
E21	211	0175	nil	02	nil	0175	881020	0
E21	212	0685	nil	02	nil	0685	881020	0
E651	6511	0680	nil	065	nil	0680	881020	0
E401	4011	0680	nil	040	nil	0680	881020	0
E401	4012	02	nil	040	nil	02	881020	0
E251	2511	0685	nil	0685	nil	025	881020	0
E252	2521	02	nil	02	nil	025	881020	0
E6801	68011	0545	nil	0545	nil	0680	881020	0
E6801	68012	0195	nil	0680	nil	0195	881020	0
E6801	68013	0175	nil	0680	nil	0175	881020	0
E6851	68511	070	nil	0685	nil	070	881020	0

C-MODIFICAS

acao alvo criacao supressao

17511	0565	881020	0
211	0175	881020	0
212	0685	881020	0
6511	0680	881020	0
4011	0680	881020	0
4012	02	881020	0
2511	0685	881020	0
2521	02	881020	0
68011	0545	881020	0
68012	0195	881020	0
68013	0175	881020	0
68511	070	881020	0



070 ** no terminal **

0680 < - - - E6801 - - - > 68011 68012 68013

1
1
1
1
1

0545 0195 0175

0565 ** no terminal **

0685 < - - - E6851 - - - > 68511

1
1
1
1

070

0195 ** no terminal **

***** FIM *****

codigo-objeto :

nome-objeto :

0175	ADIANTO CONCEDIDO
0315	CLIENTE
02	CONTRATO CAMBIO
0545	LIQUIDACAO CONTRATO
065	AVISO DEB/CRED RECEB
040	CAMBIAL OU C.REMESSA
025	PEDIDO CLIENTE
070	AVISO
0680	C.CREDITO-NEGOCIACAO
0565	ENCARGOS REG CAIXA
0685	CARTA DE CREDITO
0195	BANCO NO EXTERIOR

codigo-evento :

nome-evento :

E1751	CONCESS ADIANTO
E21	FECHTO CONTRATO
E651	RECEBTO CREDITO
E401	RECEBTO CAMBIAIS
E251	RECEBTO PEDIDO CLI
E252	REC PED FECHTO CONTR
E6801	DOCTOS S/DISCREPANC.
E6851	RECEBTO C.CREDITO

codigo-acao :

nome-acao :

17511	RECEBTO DESAGIO CLI
211	CONCES ADTO CLI
212	VINCULACAO
6511	LIQUIDACAO C.CREDITO
4011	CONFERENCIA DOCTOS
4012	VINCULACAO
2511	CONFIRMAC C.CREDITO
2521	FECHTO CONTRATO
68011	LIQUIDACAO CONTRATO
68012	ENVIO DOCTOS BCO EXT
68013	DEVOLUCAO ADTO CLI
68511	AVISO CLIENTE