UNIVERSIDADE FEDERAL DE ITAJUBÁ

UM SISTEMA PARA CONTROLE DE MANUTENÇÃO DE EQUIPAMENTOS INDÚSTRIAIS

JAIR ALVINO TAVARES JUNIOR

ITAJUBÁ, MARÇO DE 2003

Universidade Federal de Itajubá
Departamento de Engenharia Elétrica

Jair Alvino Tavares Junior

Um sistema para controle de manutenção de equipamentos industriais

Dissertação submetida ao Programa de pós-graduação
em Engenharia Elétrica como requisito
parcial à obtenção do título de Mestre
em Engenharia Elétrica

Orientador: Prof. Dr. Otávio Augusto Carpinteiro

Itajubá, Março de 2003

Júnior, Jair Alvino Tavares. Um sistema para controle de manutenção de equipamentos industriais .Itajubá: UNIFEI, 2003. 136 páginas (Dissertação de mestrado submetida ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Itajubá).

Palavras-chaves: prototipação, sistemas de informação, engenharia de software, ciclo de vida dos sistemas, ciclo de vida do software.

Universidade Federal de Itajubá
Departamento de Engenharia Elétrica

Jair Alvino Tavares Junior

Um sistema para controle de manutenção de equipamentos industriais

Dissertação submetida ao Programa de pós-graduação em Engenharia Elétrica como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica

Banca examinadora:
Prof. Dr. Otávio Augusto Carpinteiro (orientador)
Prof. Dr. Ricardo Rhomberg Martins
Prof. Dr. Antônio Carlos Zambroni de Souza

Itajubá, Março de 2003

DEDICATÓRIA

O amor é infinito e cabe num simples olhar.

À minha esposa Lúcia Helena, fonte inesgotável de carinho, incentivo e compreensão; e pelas horas roubadas dela e de meus filhos Andrei e Alison, durante a realização deste trabalho.

Agradecimentos

A caravana passou e os cães ladraram.

A Deus pela graça da vida.

Aos meus pais Jair (in memorian) e Vera, meus primeiros mestres.

Aos professores Ailema, Madisson e Otávio Carpinteiro, por acreditarem na seriedade de meu trabalho.

A Dawilmar e Hernani por fazerem do trabalho em equipe uma forma prazerosa de convivência.

Sumário

RESUMO

É fato que toda e qualquer empresa se utiliza de dados em sua gestão. É fato também que estes dados em sua forma bruta não contribuem em nada para uma mudança de comportamento empresarial. Quando, porém estes dados são processados, eles podem se tornar de grande valia para os executivos durante o processo de tomada de decisão.

Este trabalho é resultado do desenvolvimento de um software para controle de manutenção de equipamentos industriais, cujo objetivo principal é a redução de custos com a manutenção e ampliação do sistema elétrico de uma empresa.

ABSTRACT

It is a fact that all company uses data in their administration. It is also a fact that these data in its rough form don't contribute in anything for a change of managerial behavior, but when these data are processed, they can become a great worth for the executives during the process of getting of decision.

This work is resulted of the development of a software for control of maintenance of industrial equipments whose main objective is the reduction of costs with the maintenance and amplification of the electric system of a company.

Índice de figuras

Lista de siglas

CAD	Computer Aided Design
CST	Companhia Siderúrgica de Tubarão
DER	Diagrama de Entidade-Relacionamento
DFD	Diagrama de Fluxo de Dados
ERP	Enterprise Resoursing Planning
LDD	Linguagem de Descrição de Dados
LDEI	Linguagem de Definição das Estruturas de Informação
LEI	Linguagem de Especificação de Informações
LMD	Linguagem de Manipulação de Dados
LMI	Linguagem de Manipulação de Informações
MH	Modelo Hierárquico
MR	Modelo Relacional
MRd	Modelo de Redes
MS SQL	Microsoft Strutured Query Language
RFP	Requisição Formal de Proposta
SAD	Sistema de Apoio à Decisão
SIE	Sistema de Informação Executiva
SIG	Sistema de Informação Gerencial
SIO	Sistemas de Informação Operacional
SIPE	Sistema de Informações sobre a Planta Elétrica
SQL	Structured Query Language
UML	Unified Modeling Language

1. Introdução

Nos dias atuais, objetivando atingir um alto grau de competitividade, as empresas estão se tornando cada vez mais usuárias de sistemas integrados de gestão mais comumente conhecidos no mercado como ferramentas ERP (Enterprise Resouce Planning). Tal atitude por parte delas permite a disponibilização de informações de forma ágil e precisa, condição imprescindível para quem deseja se tornar ou manter-se competitivo.

Dentro desse contexto um aspecto que tem chamado a atenção de especialistas da área de gestão empresarial é o de que, apesar de existirem no mercado sistemas integrados de gestão bastante robustos, as empresas têm optado por uma solução global desenvolvida por equipes da própria corporação ou terceirizadas, enquanto que para as demais áreas a opção esta sendo feita pelas soluções dentre as mais reconhecidas em sua especialidade.

No caso da CST – Companhia Siderúrgica de Tubarão, toda a documentação de seu sistema elétrico se encontra armazenada em mídia impressa. Esta forma de armazenamento de informações sobre os equipamentos que compõem o seu parque fabril não é a mais ideal e segura devido aos seguintes aspectos:

- Exigem grande espaço físico para o seu armazenamento.
- Custo elevado de manutenção de armazenamento.
- Grande demanda de tempo para recuperação de uma informação sobre um dado equipamento ou conjunto de equipamentos.
- Atualização dos dados é feita na forma de documentos que recebem o nome de revisões.
- Facilita a geração de duplicidade de informações sobre um mesmo equipamento
- A ultima informação recuperada não é garantia de que seja a mais recente
- Não há como controlar de forma efetiva quem são os responsáveis por cada manutenção, atualização e/ou substituição de equipamentos.
- A recuperação do histórico de manutenções preventivas e corretivas de cada equipamento é inexistente.

Diante do exposto acima, o problema proposto e objeto deste trabalho consiste no desenvolvimento de uma ferramenta que permita a recuperação das informações de um dado equipamento ou conjunto de equipamentos de forma rápida e precisa. A esta ferramenta foi denominado o nome de SIPE (Sistema de Informação da Planta Elétrica da CST – Companhia Siderurgica de Tubarão).

O desenvolvimento dessa ferramenta está condicionado à utilização de um sistema gerenciador de banco de dados do tipo relacional e ainda, por determinação da própria empresa, tal sistema deve atender aos seguintes requisitos:

- Possuir uma interface amigável.
- Não utilizar nenhuma linguagem de programação a não ser a do próprio banco de dados.
- Permitir a importação e exportação parcial e total das informações contidas na base de dados.
- Interligar-se com outros sistemas a serem implantados no futuro.

Esse sistema deve ainda levar em consideração três aspectos:

- Rapidez em se obterem as informações desejadas.
- Precisão das informações.
- Segurança das informações.

O sistema desenvolvido utilizou exclusivamente o Access da Microsoft Corp. Assim sendo, a linguagem utilizada foi o SQL (Structured Query Language) que é suportada por ele. Tal sistema procurou explorar ao máximo os recursos oferecidos pela ferramenta indicada. Assim sendo, além dos recursos triviais em qualquer sistema (cadastro, consulta, exclusão e relatórios) foram implementadas rotinas de importação e exportação de dados e comunicação via rede de computadores. utilizando exclusivamente os recursos do banco de dados.

Para a rotina de comunicação foi usado o recurso de replicação de banco de dados e para as rotinas de exportação/importação de dados os recursos de LINGUAGEM SUPORTADA PELO BANCO DE DADOS.

A metodologia utilizada para implementação do sistema consistiu dos seguintes passos:

1. Identificação dos participantes do sistema "sistema atual".
2. Classificação dos participantes identificados por nível de função dentro da empresa.
3. Levantamento de todos os modelos de registros de informações sobre o "sistema atual" utilizado pelos participantes identificados no item um.
4. Classificação das informações levantadas por grupos de equipamentos.
5. Elaboração de um questionário único baseado nas informações obtidas nos itens três e quatro.
6. Aplicação deste questionário a todos os participantes do sistema.
7. Tratamento das respostas obtidas pelos questionários respondidos, classificando-as em níveis de importância.

8. Construção de um protótipo
9. Validação do protótipo
10. Implantação do sistema

Ao longo dos capítulos 2, 3, 4 e 5 é apresentada a fundamentação teórica, base para elaboração deste trabalho. Já no capitulo 6 são apresentados os modelos atual e proposto do sistema da CST. Enquanto que no capitulo 7 se discute a implementação do sistema proposto, isto é, a metodologia utilizada para solução do problema proposto, no capitulo 8, são apresentados os resultados obtidos quando da implementação sob a minha ótica e a da própria empresa. Por fim, no capitulo 8 são apresentadas as conclusões por mim obtidas ao longo de todo o processo de desenvolvimento, bem como as propostas de continuidade deste trabalho.

2. Conceitos gerais sobre sistemas de informação

O computador tem sido, nos últimos anos, o principal catalisador da revolução administrativa por que vem passando a empresa no mundo inteiro. A evolução histórica de seu uso pode ser dividida em três fases aqui denominadas de: aplicações isoladas, sistemas integrados e sistemas de informação.

Na primeira fase colocou-se o computador e conseqüentemente o próprio Centro de Processamento de Dados, como um elemento capaz de desenvolver cálculos, processando e imprimindo dados a alta velocidade. Nessa fase, as aplicações eram transpostas para o computador segundo processo similar aos manuais usados até então. Dessa forma, aplicações como folhas de pagamento, controle de estoques, contabilidade e orçamento eram atendidas isoladamente, não se dando importância às duplicidades de processos e dados.

Devido à evolução tecnológica e pela própria especialização da mão-de-obra existente, surgiu entre os homens de processamento de dados a tendência para o desenvolvimento de sistemas integrados. Nessa segunda fase, nasce a preocupação de se evitar redundância de dados nos diversos arquivos e a utilização da capacidade de transferir informações, via computador, de uma aplicação para outra.

Ao aliarmos essa possibilidade de integração à teoria de sistemas, surge a tendência atual em processamento de dados – o enfoque sistêmico. A utilização de sistemas de informação veio dar ao computador uma nova dimensão, transformando-o de mero processador de dados em elemento preponderante na racionalização e na dinamização do trabalho na empresa, modificando inclusive o conceito de centro de processamento dados.

Considerado como um fim e tendo sua filosofia de utilização ditada pelos técnicos de computação nas duas primeiras fases, o computador na fase atual passa a ser uma ferramenta a serviço da empresa.

2.1. Conceito de sistemas

A palavra sistema tem uma série de significados coloquiais, muitos deles ligados à utilização do termo em situações particulares. Vejamos algumas definições de sistema extraídas do dicionário Webster's [1]:

- Um grupo de corpos que interagem entre si ou que sejam interdependentes, formando um todo unificado.
- Um grupo de corpos que interagem entre si sob a influência de forças relacionadas (sistema gravitacional).

- Um grupo de órgãos do corpo que desempenham, em conjunto, uma ou mais funções vitais (sistema digestivo).
- Um grupo de objetos ou forças naturais relacionados entre si (sistema fluvial).

- Um conjunto organizado de doutrinas, idéias ou princípios, habitualmente previsto para explicar a organização ou funcionamento de um conjunto sistemático (sistema da mecânica newtoniana).
- Uma maneira de classificar, simbolizar ou esquematizar (sistema decimal).
- Organização harmoniosa ou modelo: ORDEM.
- Sociedade organizada ou situação social vista como desejável.

Em decorrência das definições acima, podemos concluir que, por existirem muitos tipos diferentes de sistemas quando lidamos com algo, este algo ou é um componente de algum sistema, ou é um sistema, às vezes ambas as coisas.

Ainda com relação às definições dadas acima, podemos perceber que elas apresentam uma certa classificação por categorias. Sendo o interesse principal deste trabalho os sistemas de processamento, os sistemas serão divididos em duas categorias: sistemas naturais e sistemas feitos pelo homem.

Os sistemas naturais, como o próprio nome já diz, são os sistemas encontrados na natureza; estes por sua vez, podem ser divididos em duas subcategorias básicas: os sistemas físicos (galáxias, geológicos, moleculares, etc.) e os vivos (animais, plantas, o próprio homem, etc.).

Já os sistemas feitos pelo homem, tais como: sistemas sociais, de transporte, comunicações, financeiros, também podem ser divididos em duas subcategorias: os automatizáveis e os não automatizáveis. Tal classificação pode ser feita levando-se em consideração diversos fatores como, por exemplo, custo, conforto, segurança, manutenibilidade e política, o que em outras palavras pode ser traduzido na determinação da essência do sistema.

Assim sendo, quando pensamos em um determinado sistema, devemos ter em mente cinco considerações básicas [2]:

- os objetivos totais do sistema
- o ambiente do sistema
- os recursos do sistema
- os componentes do sistema, suas finalidades, atividades e medidas de rendimento
- a administração do sistema.

Embora muitos tipos de sistemas pareçam ser totalmente diferentes, eles têm muitas semelhanças; existem princípios comuns, filosofias e teorias que se aplicam notavelmente bem a virtualmente todos os tipos de sistemas. Desta forma, com base na nossa experiência diária, bem como na experiência de cientistas e engenheiros das mais diversas áreas, podemos aplicar o que aprendemos sobre outros sistemas aos que elaboramos na área de computação [3].

2.2. Sistemas de informação

Toda empresa utiliza-se de dados. Por dados entende-se quantidade de produtos, custos de matéria-prima e de mão-de-obra, número de empregados, dentre outros mais. Porém, esses dados em sua forma bruta pouco contribuem para o executivo na busca de uma visão mais integrada de uma determinada situação. Tais executivos necessitam obter uma visão mais integrada da situação, e para isto utilizam-se de dados transformados, que podemos classificar de informação. A informação, ao ser utilizada pelo executivo, pode modificar ou afetar o comportamento existente na empresa. Oliveira [4] define informação como o dado trabalhado que permite ao executivo tomar decisões.

Segundo Wysk [5], sistema de informação é aquele sistema homem-máquina que atende às necessidades de informação de um indivíduo, grupo ou tarefa, definidas a partir de medidas que as quantifiquem, de maneira que uma organização é atendida por uma rede de sistemas de informação.

De acordo com Langefors [6], os sistemas de informação se desenvolvem em uma empresa segundo duas dimensões: os componentes da empresa (Figura 2.1), que correspondem aos diversos setores que executam as diferentes funções necessárias ao funcionamento da empresa e os níveis de decisão (Figura 2.2) que obedecem à hierarquia existente na empresa e são conhecidos como nível estratégico, nível tático e nível operacional.

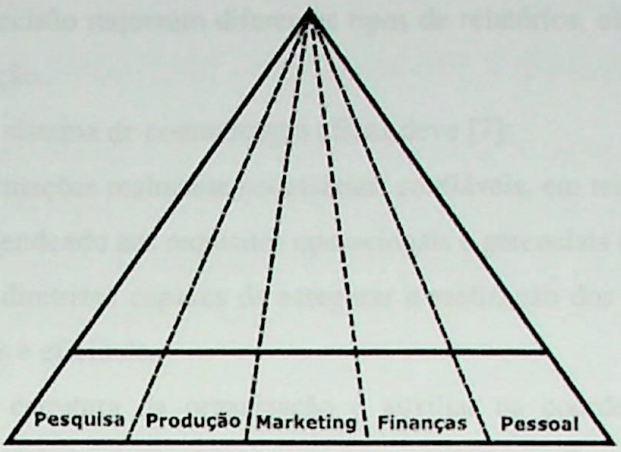


Figura 2.1 – Componentes da empresa

Figura 2.2 – Níveis de decisão

As decisões estratégicas se dão nos altos escalões da empresa e geram atos cujo efeito é duradouro e difícil de reverter. Essas emanam do planejamento a longo prazo da companhia, conhecido como planejamento estratégico.

As decisões táticas se dão nos escalões intermediários da empresa e geram atos de efeito a prazo mais curto, tendo porém menor impacto no funcionamento da companhia. As decisões táticas emanam do planejamento e do controle gerencial da empresa.

Por último as decisões operacionais estão ligadas ao controle operacional da empresa. Essas visam alcançar os padrões de funcionamento preestabelecidos.

É importante ressaltar que o tipo de decisão que é tomada em cada um dos níveis acima apresentados requer graus diferentes de agregação de informação; em outras palavras,

diferentes níveis de decisão requerem diferentes tipos de relatórios, com diferentes graus de agregação da informação.

Assim sendo, um sistema de comunicação eficaz deve [7]:

- Produzir informações realmente necessárias, confiáveis, em tempo hábil e com custo condizente, atendendo aos requisitos operacionais e gerenciais de tomada de decisão.
- Ter por base diretrizes capazes de assegurar a realização dos objetivos, de maneira direta, simples e eficiente.
- Integrar-se à estrutura da organização e auxiliar na coordenação das diferentes unidades organizacionais (departamentos, divisões, diretorias, etc.) por ele interligadas.
- Ter um fluxo de procedimentos (internos e externos ao processamento) racional, integrado, rápido e de menor custo possível.
- Contar com dispositivos de controle interno que garantam a confiabilidade das informações de saída e adequada proteção aos dados.
- Finalmente ser simples, seguro e rápido em sua operação.

3. Níveis de abstração de informações e dados

A evolução da humanidade pode ser encarada em parte como um trajeto no sentido da aquisição progressiva da capacidade individual da abstração. De um ser intimamente ligado ao Universo e em particular à natureza, o homem tornou-se ao longo do tempo um ente independente, isolado e com cada vez maior capacidade da introspecção objetiva, isto é, sem que esta dependa de fatores subjetivos, temporais e individuais. O aparecimento do computador deu-se numa época em que essa capacidade de abstração deixou de ser privilégio de alguns e passou a pertencer e ser exercida por todos aqueles cuja educação e ambiente fossem propícios ao desenvolvimento individual no sentido indicado. As informações informais deixaram de satisfazer aos anseios individuais de abstração e objetividade; cada vez mais é exigida informação mais objetiva e abstrata isto é, aquela que pode ser associada a conceitos universais e não-temporais. Dentre as informações formais, destacam-se as que podem ser expressas matematicamente. Estas são as que introduzimos no computador por meio de dados sempre tratados por espécies de fórmulas representadas pelos programas que a máquina executa direta ou indiretamente. Dados e programas são modelos formais matemáticos da realidade ou de abstrações.

Neste capítulo serão apresentados os vários níveis de abstração envolvidos no processo de se tratarem informações por meio da máquina abstrata (isto é, matemática) que é um computador, segundo a visão de Setzer [9].

Na figura 3.1 é apresentado um esquema que contém os vários níveis envolvidos em um possível processo de modelagem levando à criação de uma base de dados.

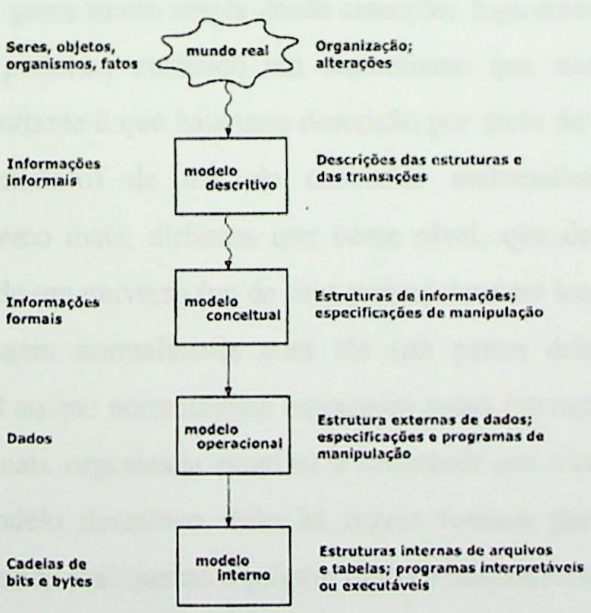


Figura 3.1 – Níveis envolvidos em processo de modelagem

O nível mais alto é o do mundo real, que do ponto de vista formal é ainda muito nebuloso. Vários cientistas e leigos têm a fé de que um dia o mundo real será todo formalizável, pois a sua visão do Homem e do universo é mecanicista. Mas mesmo os que têm essa moderna religiosidade [10] concordarão que o conhecimento científico atual do mundo real é ínfimo, e portanto ele permanece nebuloso em termos científicos clássicos, daí termo-lo representado por uma nuvem. Os "objetos" do mundo real são os seres, os fatos, as coisas, e os organismos sociais. Assim, estamos considerando um departamento de uma empresa como algo do mundo real, pois é um organismo social. Mesmo se fosse apenas uma especificação que existe num papel, ele poderia ser considerado como pertencente ao mundo real, como o são as plantas de um projeto de engenharia. Note-se que não desejamos dar precisão à palavra "mundo real", estamos cientes de que existem várias questões filosóficas e de percepção ligadas a esse conceito, como por exemplo se os pensamentos, sentimentos e a vontade são ou não reais ou, de outra maneira, a distinção entre o concreto e o abstrato. Materialistas diriam que apenas o que é fisicamente material é real; idealistas diriam que apenas o não-físico é real, e monistas considerariam ambos como realidades em planos diferentes (Goethe disse uma vez a Schiller que "observava" sua planta arquetípica com tanta realidade como se observa um objeto qualquer com nossos sentidos comuns). Com a nossa imprecisão, deixaremos para o projetista delimitar o que lhe interessa como mundo real, para fins de tratamento de informações.

O segundo nível é o das informações informais, e é caracterizado por relatórios escritos em uma linguagem natural (Português, Inglês, etc.). Já é um nível de abstrações, apesar dessas poderem abranger uma gama muito ampla desde asserções logicamente perfeitas, até frases ambíguas ou mesmo poéticas, contendo um simbolismo que transcende a experiência sensorial direta. O importante é que haja uma descrição por meio de frases, preferivelmente sem (ou com um mínimo) de uso de conceitos matematicamente formais. Para caracterizarmos um pouco mais, diríamos que nesse nível, que denominaremos de nível descritivo, a descrição de um universo (ou de suas partes) deve ser totalmente inteligível para as pessoas que interagem normalmente com ele (ou partes dele), sem se exigir um conhecimento adicional ao que normalmente empregam nessa interação. É evidente que essa descrição deve ser a mais organizada possível e constituir um modelo da realidade, que denominaremos de modelo descritivo. Não há regras formais para se desenvolver esse modelo, pois tanto o mundo real quanto o próprio modelo descritivo não são formais. Assim, pode-se no máximo dar certas diretrizes de como derivá-lo do mundo real e como organizá-lo.

O terceiro nível é o das informações formais, em que o modelo desenvolvido deve ser estritamente formal. Como o objetivo é chegar-se, em um nível posterior, a um modelo computacional, isto é, que pode ser fornecido a (e processado por) um computador, o formalismo a ser adotado é o da matemática. O computador só aceita linguagens estritamente formais, e é por si só uma máquina abstrata, uma máquina matemática. Isto é, todo o seu processamento pode ser representado por formalismos matemáticos. Assim, a adoção de modelos estritamente matemáticos no nível das informações formais é um passo dirigido para facilitar a posterior formulação no nível computacional. Vamos denominar esses modelos de modelos conceituais, para caracterizar que são baseados em símbolos para os quais deve haver uma conceituação rigorosa. Estamos cientes de que os conceitos transcendem o formalismo matemático, como por exemplo, o conceito de árvore, e que estamos, portanto deturpando a palavra "conceito". Mas como se tratará de "modelos conceituais de base de dados", cremos que o crime não é tão grande, e permite-nos usar essa expressão bastante comum na área, se bem que aplicada em casos um pouco diferentes do nosso. Assim sendo, o modelo descritivo é também conceitual, mas empregaremos esta denominação apenas para modelos do terceiro nível, isto é: sempre que nos referirmos a "modelo conceitual" queremos dizer "modelo conceitual formal". Denominaremos neste texto o nível das informações formais por nível conceitual.

Nos modelos conceituais aparecem dois aspectos distintos, em geral misturados nos modelos descritivos: trata-se das estruturas de informações e das manipulações das informações. Estes aspectos vão ficar claros quando descrevermos o modelo conceitual. Por ora, basta considerarmos que, de um lado, as informações podem ser organizadas estruturalmente, como por exemplo, o fato de as informações sobre fornecedores conterem partes referentes ao endereço - que por sua vez é estruturado em local, CEP e cidade, sendo o local subdividido em rua, número e complemento, referentes ao nome do fornecedor, seu CGC, etc. As informações sobre os materiais consumidos pela empresa contêm partes referentes à classe do material, à sua descrição, código interno, etc. Uma outra informação estrutural é a de que existe uma associação entre fornecedores e materiais, representando a informação de que material é fornecido por qual fornecedor (e vice-versa). Por outro lado, são manipulações de informações, por exemplo, a atualização do endereço de um fornecedor, a confecção de um relatório especificando para um dado material quais fornecedores de uma determinada cidade que o fornecem, etc. Assim, as estruturas das informações são meta-informações que descrevem as informações propriamente ditas. O fato de cada fornecedor ter um endereço é uma meta-informação; o endereço de um determinado fornecedor é uma

informação cuja estrutura segue as especificações estruturais da meta-informação. As manipulações das informações referem-se ao seu tratamento. Em geral esse tratamento pode ser subdividido em gravação, atualização, eliminação, leitura e processamento das informações. Por exemplo, se queremos indicar a mudança de endereço de um fornecedor, devemos especificar formalmente uma ação de atualização; se quisermos emitir um relatório sobre as quantidades de um certo material fornecidas pelos fornecedores de cada cidade, teremos que especificar ações de leitura e talvez de processamento. Pode-se estender a manipulação de informações com ações de definir, atualizar, eliminar e ler as meta-informações.

Neste terceiro nível é conveniente definir-se uma linguagem para se especificarem as estruturas das informações, que denominaremos de Linguagem de Definição das Estruturas de Informação (LDEI), e uma linguagem para se especificarem as manipulações, que denominaremos de Linguagem de Manipulação de Informações (LMI). Ambas podem ser fundidas em uma só linguagem, a Linguagem de Especificação de Informações (LEI). Na verdade cada uma pode ser uma família de linguagens; por exemplo, é conveniente ter-se tanto linguagens gráficas como linguagens escritas (isto é, com letras e símbolos tipógrafos). No caso da LMI, podem-se ter várias linguagens escritas abrangendo vários níveis de detalhamento das especificações. A nossa inclinação é por uma só linguagem de amplo espectro, cobrindo tudo o que se deseja expressar no nível conceitual.

Encontra-se na literatura a designação de operações para comandos ou especificações da LMI, como por exemplo, em [11] e [12].

O quarto nível é o nível dos dados, que são os símbolos a serem introduzidos no computador, tanto na descrição de estruturas (meta-dados, isto é, dados que descrevem os dados) como aqueles que constituem os dados a serem propriamente processados pela máquina. A máquina vai operar com os dados, daí termos denominado os modelos que descrevem os dados como modelos operacionais; dai denominarmos o nível em questão de nível operacional. Deve ficar bem clara, portanto, a distinção entre informações formais e dados. As primeiras podem seguir qualquer formalismo matemático, podem existir no papel ou mesmo na nossa mente. Os últimos devem ser expressos de tal forma que um computador os possa receber e tratar. Vamos dar um exemplo. Suponhamos que os dois últimos dígitos do CPF (número de inscrição de pessoas físicas na Receita Federal), que são os dígitos de controle, estejam sendo usados para verificação da consistência (integridade) desse número. Nesse caso deve existir uma fórmula matemática complexa com a qual se possam obter os dígitos de controle a partir dos outros dígitos. No estado atual das linguagens de programação,

não se pode fornecer essa fórmula diretamente ao computador; é necessário transformá-la em um algoritmo de cálculo que é então programado em alguma linguagem de programação. No modelo conceitual formal, devem-se expressar as informações o mais próximo possível do mundo real (ou do modelo descritivo), de modo que a fórmula matemática seria a maneira adequada. No entanto, no modelo operacional essa fórmula teria que ser expressa provavelmente em forma algorítmica, que certamente não será tão clara e compreensível como a original. Pior ainda, o algoritmo terá que ser expresso em "programês", isto é, em alguma linguagem de programação, o que torna ainda mais obscura a sua apresentação.

É interessante notar que os sistemas de computação estão sendo produzidos em nível cada vez mais alto do ponto de vista do usuário, isto é, este está precisando conhecer cada vez menos os detalhes da computação para poder especificar seus problemas e seus dados. Assim, os dados estão chegando cada vez mais próximos das informações formais. O usuário não está mais precisando aprender "programes" para usar o computador; através do uso de linguagens gerais de especificação de sistemas e de transações ele está precisando aprender o que poderíamos denominar de "analistês". Com o esforço de se desenvolverem sistemas especialistas, em que o usuário usa na comunicação com a máquina exclusivamente os termos que emprega profissionalmente passaremos, na área administrativa, a exigir do usuário apenas o aprendizado de um "administrês". Existem tentativas de se fornecer ao usuário sistemas em que ele emprega a linguagem do nível descritivo, que é a linguagem natural, isto é, apenas a sintaxe é a da linguagem natural, com grandes limitações no reconhecimento das especificações dadas pelo usuário. O grande problema de se usar diretamente nível conceitual ou mesmo o nível descritivo é que no nível operacional o usuário pode dar inúmeras especificações de eficiência, necessitando para isto de um bom conhecimento de "programês". Com o aumento da capacidade dos circuitos e com a programação de sistemas muito complexos, as máquinas estão começando a ter possibilidade de processar as especificações dadas pelos usuários otimizando as estruturas de dados e o código gerado. Com isso, a principal vantagem de se usar o nível operacional, que é a eficiência, está começando a desaparecer (antigamente havia um problema de possibilidade - os sistemas de computação só aceitavam dados e programas em "programês"), mas certamente ainda levará algum tempo para se conseguir eficiência em sistemas especificados no nível conceitual.

Da mesma maneira que no nível conceitual, o usuário deverá usar linguagens de especificação tanto das estruturas de dados (meta-dados), como do tratamento (gravação, atualização, eliminação, leitura e processamento) que ele deseja dar aos dados. Para usarmos uma nomenclatura já tradicional na área, chamaremos esses duas linguagens de Linguagem de

Descrição de Dados (DDL) e Linguagem de Manipulação de Dados (DML), provenientes de "Data Description Language" e de "Data Manipulation Language" respectivamente.

Os modelos de dados do nível operacional dividem-se tradicionalmente em Modelo Relacional (MR), Modelo de Redes (MRd) e Modelo Hierárquico (MH), que serão abordados com algum detalhe neste texto. Neste texto subdividiremos o MR em dois tipos: o Normalizado e o Não-Normalizado; praticamente todos os gerenciadores de bancos de dados comerciais relacionais são do primeiro tipo; o segundo está em pesquisa e nos parece indicar o futuro desses modelos. Há um sistema comercial que se aproxima parcialmente desse tipo.

Existem várias espécies de DML, que podem ser classificadas quanto ao nível de conhecimento que elas exigem do usuário. Isso é análogo aos níveis do "administrês", passando pelo "analistês" e terminando em "programês", com várias subdivisões em cada um. Por exemplo classificaríamos o que se costuma chamar de "query language", denominada por nós de "linguagem de consulta direta" (uma linguagem em que são dadas especificações não-algorítmicas das manipulações de dados) no nível de "analistês", exigindo pouco (ou nenhum) conhecimento de computação, mas exigindo ainda conhecimento tanto de uma sintaxe bastante limitada, quanto de alguma estrutura dos arquivos.

Os comandos e especificações da DML também são denominados, na literatura, de operações, como no nível conceitual. Sundgren [13] usa-o somente para o modelo operacional.

O quinto e último nível é o nível da máquina, não mais do ponto vista do usuário, mas dos aspectos internos, isto é, das representações internas dos dados e dos programas. Por exemplo, estes podem estar em linguagem de máquina (código objeto executável) ou em uma linguagem intermediária (código objeto interpretável). O usuário não toma conhecimento desses detalhes. Não lhe interessa qual a forma sob a qual seus dados estão descritos internamente, isto é, como as estruturas de dados que ele forneceu se encontram gravadas, nem como os seus dados propriamente ditos estão armazenados, e nem a forma sob a qual as suas especificações de manipulações de dados estão lá dentro da máquina. A esse nível máquina "por dentro" e não "por fora", denominaremos de nível interno; os modelos correspondentes serão os modelos internos, esse é o nível não mais dos dados, mas das cadeias de "bits" ou de "bytes". Note-se que essas cadeias não contêm nenhuma estrutura intrínseca; tanto podem ser dados do usuário, como estruturas puramente internas (contadores, apontadores, etc.) ou programas. As estruturas são definidas pelos programas que as utilizam.

Existe a possibilidade de se subdividir o nível interno separando-se, por exemplo, as cadeias com especificações estruturais das cadeias correspondentes aos dados propriamente

ditos, o que se poderia chamar de nível físico; o mais inferior possível. Mas não o faremos, pois para o usuário o computador deve ser sob certos aspectos análogo ao seu próprio organismo. Para ele não interessa a forma interna dos dados; não é mais seu problema. Na verdade, as coisas não foram bem assim no passado, e continuam sendo, em menor escala, no presente: para conseguir maior eficiência, em geral o usuário deve entrar em maior ou menor grau nos detalhes dos "bits" e dos "bytes". Mas isso está mudando, e temos a esperança de que num futuro próximo desapareça por completo, pelo menos na área de uso de gerenciadores de bases de dados, a necessidade de o usuário ter qualquer conhecimento de computação.

Antes de deixarmos este item, é interessante mencionar que níveis de abstração semelhantes aos expostos aqui foram usados em vários trabalhos. Por exemplo, no modelo Infológico desenvolvido por Langefors [6], o que denominamos de modelo descritivo é aproximadamente o que Sundgren [13] chama de "modelo do assunto" ("subject matter model"); nosso modelo conceitual é seu "modelo Infológico", se bem que ele emprega um particular modelo com essa denominação; nosso modelo operacional é seu "modelo datalógico". Um outro exemplo de esquema semelhante encontra-se em Teorey e Fry [14].

É importante distinguir nossa denominação de "conceitual" em contraposição a essa nomenclatura como introduzida pelo conhecido estudo ANSI/SPARC [15]; uma introdução ao mesmo pode ser encontrada no livro de Date [16]. No caso daquele estudo, a expressão "nível conceitual" foi empregada para denominar o conjunto de todos os dados, no nosso nível operacional. Ela é contraposta ao "nível externo", que indica a visão que cada usuário recebe dos dados, contendo apenas as estruturas e os dados que ele vai manipular. Por exemplo, um usuário poderia receber apenas as informações sobre os funcionários de uma empresa e os departamentos onde estão lotados, não tendo conhecimento da existência de dados sobre produtos, fornecedores de materiais, etc. A visão "conceitual" seria, segundo Date, "uma visão do conteúdo total do banco de dados, e o esquema conceitual é uma definição dessa visão". No nosso caso, fizemos uma distinção entre "informações" e "dados" que não ocorre no estudo ANSI/SPARC. Não abordamos as "visões dos usuários" pois, apesar de introduzirem uma noção muito importante, aplicam-se tanto para o nosso nível conceitual como para o operacional.

Verificamos a possibilidade de serem reconhecidos vários níveis de abstração, e o fato de o usuário enfrentar diretamente, em maior ou menor grau, quatro desses níveis, sendo que nos níveis conceitual e operacional devem existir modelos formais que descrevem tanto a estrutura como a manipulação de informações e dados. Partindo da premissa de que é possível

usar métodos para se derivar sistematicamente os modelos dos vários níveis, serão aqui abordados dois tipos de métodos, o descendente ("top-down") e o ascendente ("bottom-up").

Um método descendente é aquele que segue os níveis de abstração "de cima para baixo": inicialmente é feito um modelo descritivo a partir de observações e vivência do mundo real; a partir do modelo descritivo deriva-se um modelo conceitual, e deste um modelo operacional que é então introduzido no computador.

É evidente que não pode existir um método formal para se analisar o mundo real (pois este não é, por hipótese, formal), a fim de se derivar um modelo, descritivo, que por sua vez também não deve ser formal. Essa derivação não formal foi representada na figura 1 por um arco bem sinuoso que vai do mundo real para o modelo descritivo. Para se derivar um modelo conceitual formal a partir do modelo descritivo também não pode existir um método formal, pois o segundo modelo não é formal. Esse passo é certamente mais formal (ou menos informal) que o anterior, dai ter sido representado na figura 1 por um arco pouco sinuoso.

Já a derivação de um modelo operacional a partir de um modelo conceitual pode ser feita em quase sua totalidade de maneira puramente formal, o que foi mostrado na figura 1 por uma seta retilínea. Não pode haver um mapeamento total de um modelo no outro, pois, como já vimos, existem elementos que pertencem a cada um dos níveis e que não ocorrem no outro. Por exemplo: o fato de se desejar especificar no modelo operacional um item de busca, isto é, um dado que será muito empregado nas consultas à base de dados. Seria o caso da consulta ao conjunto de dados dos funcionários, que se deseja fazer com muita freqüência nos dados "salário" e "data de contratação", de modo que seria muito importante especificar ao sistema que esses dois dados devem ser estruturados internamente de maneira especial (por exemplo, usando-se listas invertidas) para permitir buscas eficientes a partir deles. Essa especificação de eficiência não deve existir no modelo conceitual, pois aí interessa apenas o fato de as especificações refletirem o melhor possível o mundo real, não importando o aspecto computacional. É evidente que aspectos computacionais poderiam ser introduzidos no nível conceitual, mas a nossa opinião é que eles devem ser em geral deixados exclusivamente para o nível operacional.

Os modelos operacionais não serão introduzidos como estruturas existentes de *per si*, mas como ferramentas na implementação de uma base de dados seguindo um método descendente do projeto. Assim, ao aspecto didático dessa abordagem soma-se o aspecto prático de técnica de projeto.

4. Participantes do sistema

Quando trabalhamos no desenvolvimento de um projeto interagimos com diversos tipos de pessoas. O elenco de personagens varia de projeto para projeto, as personalidades serão extremamente diferentes umas das outras e o número de pessoas com quem iremos interagir variará de apenas uma a diversas dúzias. Estes "personagens" além de serem altamente consistentes muitas vezes falam uma "linguagem" bastante diferente da utilizada pela equipe de desenvolvimento.

A capacidade de compreender tal linguagem por parte da equipe de desenvolvimento implica diretamente no grau de objetividade que será alcançado quando o projeto estiver concluído. A seguir serão apresentados os principais tipos de participantes que são encontrados em um típico projeto de desenvolvimento de sistemas.

Os participantes são classificados pelos seguintes grupos [14]:

- Usuários
- Gerentes
- Auditores, pessoal do controle de qualidade e mantenedores dos padrões
- Analistas de sistemas
- Projetistas de sistemas
- Programadores
- Pessoal operativo

Todos os participantes são pessoas, e elas têm diferentes metas, diferentes prioridades e diferentes perspectivas. Embora elas possam estar comprometidas com o sucesso do projeto, elas podem ter preocupações ocultas que se opõem a um ou mais aspectos desse projeto.

Todo o restante deste capítulo trata da descrição de cada tipo de participante do sistema, segundo a visão de Yordon [14].

4.1. Usuários

O primeiro, e de longe o mais importante, participante do sistema. É a pessoa ou grupo de pessoas que:

- Solicita, de modo formal ou informalmente, um sistema.
- Será entrevistada, muitas vezes detalhadamente, para saber que características o novo sistema deverá ter para ser bem sucedido.

- É proprietário no sentido de que ele recebe ou herda e às vezes possui o sistema depois de pronto.
- É o cliente, porque é ele quem paga pelo sistema.

Os usuários podem ainda ser classificados de dois modos: por nível de supervisão (ou tipo de função) e por nível de experiência com processamento de dados.

4.1.1. Classificação do usuário por tipo de função

Nesta categoria podemos destacar três tipos principais de usuários:

- Os operativos – que são funcionários burocratas, operativos e administrativos que com mais probabilidade terão contato diário com o novo sistema.
- Os supervisores – geralmente chefiam um grupo de usuários operativos e são responsáveis por seus desempenhos: são eles que normalmente definem os requisitos e a detalhada orientação comercial que o sistema deve seguir.
- Os executivos – normalmente não estão diretamente envolvidos nos projetos de desenvolvimento de sistemas, a menos que o projeto seja tão grande e tão importante que venha a ter grande impacto na organização. Estes são normalmente capazes de lidar com modelos abstratos de um sistema.

Em suma, um analista deve ter em mente que esses usuários têm diferentes perspectivas, interesses e prioridades e em muitos casos diferentes retrospectos. Ressalta-se aqui que qualquer um deles nem sempre estará satisfeito com o sistema. Para Marjorie Leeson [17], o analista que conhece a motivação básica, porque e como as pessoas resistem às modificações, pode ser capaz de superar algumas resistências. A maioria dos livros faz referência à hierarquia das necessidades, do psicólogo A. H. Maslow. A tabela a seguir apresenta as cinco categorias, da prioridade mais baixa à prioridade mais elevada:

CATEGORIA	PRIORIDADES
Fisiológica	Alimentação, vestuário e abrigo.
Segurança	Proteção contra perigos e perda de emprego
Social	Identificação com pessoas e grupos
Egoísta	Reconhecimento, status, e importância.
Realização pessoal	Realização de todo o potencial em criatividade e desenvolvimento pessoal

Figura 4.1 – Prioridades dos usuários

Desse modo, quando encontramos alguns usuários apresentando resistência à idéia de um novo sistema, devemos considerar a possibilidade de uma ou mais dessas necessidades não serem satisfeitas. É difícil, naturalmente, que um usuário se preocupe com o nível fisiológico de necessidades, mas não é absolutamente surpreendente descobrir que um usuário esteja preocupado com a perda de seu emprego. E também é comum que os usuários (principalmente os operativos) se preocupem com que o novo sistema faça com que eles não se identifiquem com seus respectivos grupos sociais. O usuário operativo que se tornou perito na execução de uma atividade manual pode achar que um novo sistema deixará suas necessidades "egoísticas" insatisfeitas; e o usuário que imaginar que o sistema poderá eliminar os aspectos de sua atividade atual também poderá opor-se.

4.1.2. Classificação do usuário por nível de experiência

Ao classificarmos os usuários pelo nível de experiência podemos distribuí-los em três grupos: os amadores, os novatos e os peritos em processamento. Nos últimos tempos o número de participantes do grupo de amadores tem aumentado rapidamente. O verdadeiro problema com o usuário amador é algo mais sutil. Ele pode sentir dificuldade em compreender a "linguagem" utilizada pelo analista de sistemas para descrever os recursos, funções e características do sistema a ser construído, ainda que a linguagem evite a terminologia relacionada a computadores. A criação de modelos por meio de representações gráficas facilita a compreensão do novo sistema por parte desse usuário, evitando desta forma que o mesmo fique insatisfeito como o novo sistema quando pronto.

Já o segundo usuário, o novato, é aquele que participou de um ou dois projetos de desenvolvimento de sistemas e que possui um certo domínio sobre algum tipo de aplicativo. Tal tipo de usuário freqüentemente alardeia que sabe exatamente o que deseja que o sistema faça e está disposto a apontar os erros cometidos pelo analista de sistemas no último projeto. Tudo isso é ótimo, exceto por um detalhe: o usuário muitas vezes se envolve demais em discussões sobre a especifica tecnologia que será utilizada na implementação do sistema, como por exemplo: definir tipo de hardware, quais ferramentas a serem utilizadas no desenvolvimento do sistema, qual ou quais tipos de bancos de dados. Tais opções eventualmente podem se revelar corretas, mas são prematuras as considerações desse tipo antes que os verdadeiros requisitos do sistema tenham sido documentados.

Por último, existem os usuários que realmente conhecem análise de sistemas, bem como a subjacente tecnologia dos computadores. É um prazer trabalhar com essas pessoas. Na realidade o único risco que pode haver em trabalhar com este tipo de usuário é que tanto ele

quanto o analista de sistemas sintam tanto prazer em conversar sobre as ferramentas e as técnicas da análise de sistemas que esqueçam que o objetivo real é a elaboração de um sistema que funcione.

4.2. Gerentes

Gerência é um termo bastante vago. O analista de sistemas provavelmente terá contato com vários tipos de gerentes, tais como:

- Gerentes usuários – são os gerentes encarregados de várias pessoas da área operativa em que o novo sistema será utilizado.
- Gerentes de projeto – são as pessoas encarregadas do projeto de desenvolvimento do sistema.
- Gerente geral – com um nível hierárquico mais elevado, não estão diretamente envolvidos nos projetos e nem na alocação de recursos. Normalmente estão mais interessados no planejamento estratégico e de apoio à decisão.

A principal interação entre o analista de sistemas e todos esses gerentes tem a ver com os recursos que serão destinados ao projeto. É tarefa do analista de sistemas identificar e documentar os requisitos do usuário e as restrições dentro das quais o sistema deverá ser construído.

4.3. Auditores, pessoal do controle de qualidade e mantenedores dos padrões

Dependendo do tamanho de seu projeto e da natureza da organização em que trabalha, você pode ou não ter auditores, pessoal de controle de qualidade e/ou membros do setor de padronização participando do projeto.

O objetivo geral desse grupo tão heterogêneo é garantir que o seu sistema será desenvolvido de acordo com vários padrões externos (ao seu projeto).

Existem três problemas que devem ser tratados antecipadamente ao se lidar com os membros desse grupo:

- Muitas vezes não se envolvem no projeto até que esteja terminado.
- Muitas vezes estão habituados com uma notação ou formato de documentação diferente dos que você esta utilizando no projeto
- Freqüentemente estão mais interessados na forma do que na substância: se seus documentos não estiverem exatamente corretos, poderão ser rejeitados.

4.4. Analistas de sistemas

Membro essencial de qualquer equipe de projeto de desenvolvimento de sistemas, o analista de sistemas desempenha vários papéis:

- Arqueólogo e escriba – uma de suas principais tarefas é detalhar e documentar a orientação comercial que possa existir no "folclore tribal", passado de geração em geração de usuários.
- Inovador – deve auxiliar o usuário a explorar as novas e úteis aplicações dos computadores bem como novas maneiras de o usuário conduzir seus negócios.
- Mediador – a principal atribuição do analista, aqui, é obter um consenso entre usuários, gerentes, programadores, auditores e vários outros participantes, o que requer a delicada arte da diplomacia e da negociação.
- Líder de projeto – não é um papel universal, mas ocorre com suficiente freqüência. O analista de sistemas costuma ser mais experiente que os programadores do projeto, além de ser designado para o projeto antes dos programadores, tornando-se uma tendência natural sua atribuição às responsabilidades da gerência do projeto.

Como mostrado acima, o analista de sistemas necessita de uma mente lógica e organizada; precisa ser capaz de visualizar um sistema de várias perspectivas diferentes, ser capaz de subdividi-lo em subsistemas de vários níveis e de pensar em um sistema em termos abstratos e físicos.

4.5. Projetistas de sistemas

Em muitos casos o analista de sistemas e o projetista são a mesma pessoa, ou membros do mesmo grupo unificado de pessoas. Mesmo que sejam pessoas diferentes, é importante para o analista e o projetista de sistemas permanecerem em estreito contato durante todo o projeto. O motivo disso é a realimentação que ocorre entre a análise e o projeto de sistemas. O analista de sistemas deve fornecer informações suficientemente detalhadas para que o projetista elabore um projeto tecnologicamente bom, e o projetista de sistemas deve fornecer informações suficientemente acuradas para que o analista de sistemas possa dizer se os requisitos do usuário que ele está documentando são tecnologicamente viáveis. Fundamentando-se nas informações recebidas do projetista de sistemas, o analista pode negociar com o usuário para modificar seus requisitos.

4.6. Programadores

Em grandes projetos de desenvolvimento de sistemas, os projetistas se assemelham a um "buffer" entre os analistas e os programadores, isto é, os analistas de sistemas entregam suas listas dos requisitos do sistema, independentes da tecnologia, aos projetistas de sistemas que, por sua vez, entregam aos programadores uma descrição arquitetural dos componentes de *hardware* e *software* que serão utilizados para implementar o sistema.

Uma outra razão para que o analista de sistemas e os programadores tenham pouco ou nenhum contato entre si consiste no fato de que muitas vezes o serviço é executado de maneira estritamente seqüencial na maioria dos projetos de desenvolvimento de sistemas, embora existam alguns projetos (e algumas organizações) onde a análise de sistemas, o projeto e a implementação são feitos pela mesma pessoa. Desse modo, a tarefa de análise de sistemas é iniciada antes e é *totalmente executada* antes que a tarefa de programação comece. Isso significa que o analista de sistemas termina seu trabalho e talvez já tenha sido designado para um novo projeto antes que o programador sequer tenha iniciado sua participação no projeto.

Entretanto, é provável que haja algum contato entre os programadores e os analistas de sistemas pelas seguintes razões:

- O analista às vezes serve como gerente de projeto.
- Muitas vezes é o programador que descobre ambigüidades na lista de requisitos produzida pelo analista de sistemas.
- O redesenvolvimento de projetos antigos (em que praticamente não há documentação que descreva como o sistema funciona) faz com que as atenções se voltem para o *programador de manutenção,* que é a pessoa que tem feito o sistema continuar funcionando nos últimos anos.
- Algumas organizações estão começando a modificar suas equipes de projeto, da estrutura vertical para uma horizontal. A adoção dessa abordagem permite que os analistas de sistemas e os programadores permaneçam em estreito contato durante todo o projeto. Na realidade cada um deles executa parte do trabalho originalmente feito pelo outro.

4.7. Técnicos operacionais ou grupo de suporte e serviços

Responsável pela segurança dos dados, do hardware, do sistema de telecomunicações, o analista deve conhecer as restrições impostas por eles, uma vez que tais restrições deverão fazer parte da especificação detalhada produzida pelo analista.

5. Gestão do projeto de desenvolvimento de sistemas

5.1. Engenharia de software

Durante o desenvolvimento de software surgem muitos problemas, de ordem técnica, política e financeira, entre outras. Com uma metodologia pré-definida torna-se mais fácil para a organização encontrar soluções. Sabe-se que os problemas não desaparecerão tão rapidamente, mas, com técnicas, ferramentas e estratégias adequadas aumentam-se as possibilidades de sucesso.

No entanto, nota-se que cada organização acaba por conceber uma metodologia própria, conforme suas necessidades e recursos disponíveis. Não existe na verdade uma única abordagem que seja a melhor para a solução dos problemas que abalam o desenvolvimento de software. No entanto, ao se combinarem metodologias abrangentes para todas as fases de desenvolvimento do software, melhores ferramentas para automatizar essas metodologias, blocos de construção mais poderosos para implementação, melhores técnicas para garantia da qualidade, e uma filosofia de coordenação predominante de controle e administração, pode-se conseguir uma disciplina para o desenvolvimento do software. Essa disciplina chama-se Engenharia de Software.

Diferentes meios para a abordagem do problema são propostos pela engenharia de software, cabendo aos desenvolvedores escolher aqueles que serão utilizados para o desenvolvimento do seu software, conforme sua situação particular. Essa escolha é até um tanto quanto difícil, pois envolve um conjunto de aspectos, tais como o perfil da equipe de desenvolvimento, o tipo de sistema a ser desenvolvido, o rigor do cumprimento do cronograma, o nível de qualidade desejado, as ferramentas disponíveis e as formas de utilização, entre outros.

A engenharia de software trata, portanto, da construção e manutenção dos sistemas de software, propondo técnicas, metodologias e ferramentas para abordar o processo de desenvolvimento, bem como suas utilizações.

Ela abrange também a organização do ambiente de trabalho e o controle e gerenciamento de todas as atividades relacionadas ao processo de desenvolvimento, visando sempre à produtividade das tarefas e à qualidade dos produtos [24].

5.2. Qualidade de software

Todos os ramos da engenharia preocupam-se em desenvolver, de forma econômica, produtos com qualidade preditível e controlável. Assim sendo, também em engenharia de software tem sido intensa a busca de melhores conhecimentos com relação à qualidade.

Qualidade é um atributo associado a alguma coisa. Assim sendo, qualidade não pode ser definida universalmente, mas deve ser definida para o item em questão (qualidade de especificações, de metodologias, de programas, etc.).

Qualidade é, também, um conceito multidimensional que se realiza através de um conjunto de atributos ou características. Industrias preocupadas com a produção de produtos de alta qualidade estabelecem um nível aceitável de qualidade e, através de mecanismos específicos, asseguram que esse nível de qualidade seja mantido. Organizações que desenvolvem software devem assumir a responsabilidade de determinar esse nível de qualidade e de estabelecer mecanismos para determinar se esse nível é atingido (controle de qualidade).

Existem diferentes classes de projetos, cada uma delas com exigências de qualidade concretas. Podemos, então, falar de níveis diferentes de qualidade, dependendo da aplicação (não é o mesmo considerar-se a qualidade de um sistema comercial e de um sistema de controle de tráfego aéreo ou de usinas nucleares). No entanto, qualquer que seja o sistema em questão, ele deve atingir um nível satisfatório de qualidade e, para que este nível seja atingido, ele precisa ser claramente definido pelos desenvolvedores do sistema.

Qualidade de software é, portanto, um conjunto de propriedades a serem satisfeitas em determinado grau, de modo que o software satisfaça às necessidades de seus usuários.

Não é suficiente, no entanto, somente desejar-se que o software a ser produzido atinja um nível adequado de qualidade. Para que esse nível seja alcançado, é necessário buscar seu alcance desde o início do desenvolvimento, uma vez que níveis satisfatórios de qualidade só são assegurados se o grupo de desenvolvedores atuar consciente e deliberadamente para atingi-los [25] [26].

5.3. Avaliação da qualidade de software

Uma vez evidenciada a importância de avaliar e controlar a qualidade de software, tornam-se necessários métodos que possibilitem essa avaliação e controle.

Um método para avaliação da qualidade deve ter as seguintes propriedades [25]:

- Confiabilidade – isto é, as medidas obtidas utilizando o método são indicativas da qualidade do produto. Um método é confiável quando:

- O atributo sob avaliação está explicitamente definido.
- A definição do atributo e do processo de avaliação não contém ambigüidades.
- A avaliação é repetitiva, isto é, se realizada em diferentes ocasiões produz a mesma opinião sobre a qualidade.
- A avaliação é independente do observador.
- A avaliação resulta numa opinião objetiva.
- Efetividade – isto é, a avaliação pode ser realizada e contribui para um melhor entendimento do produto. O método de avaliação é efetivo quando a avaliação é:
  - tecnicamente viável
  - economicamente viável
  - útil

Existem vários métodos para avaliação da qualidade de software [27]:

- avaliação por julgamento – um analista de controle da qualidade, usando sua experiência, emite uma opinião sobre a presença ou não de qualidade em um determinado produto. Avaliações, segundo este método, são bastante subjetivas e difíceis de serem aceitas pelos desenvolvedores. A credibilidade da avaliação depende diretamente da credibilidade do avaliador.
- Avaliação por critérios ausentes – um analista de controle da qualidade revê o produto procurando a presença ou ausência de critérios específicos. Com a identificação dos critérios para avaliação reduz-se bastante a subjetividade nas avaliações.

O controle da qualidade deve ser realizado durante todas as fases do desenvolvimento. Tal procedimento fornece uma indicação do progresso que se está obtendo rumo à qualidade desejada para um determinado produto. Quanto mais cedo, no ciclo de vida, forem realizadas avaliações de qualidade, mais indicativos serão obtidos.

Com relação à qualidade, existem dois tipos de medidas:

- A preditiva – que é orientada para o software disponível durante à fase de desenvolvimento, e que pode ser classificada de dois modos: a quantitativa e a binária, que determinam a presença ou ausência de um atributo;
- De aceitação – que pode ser vista como uma validação das medidas preditivas.

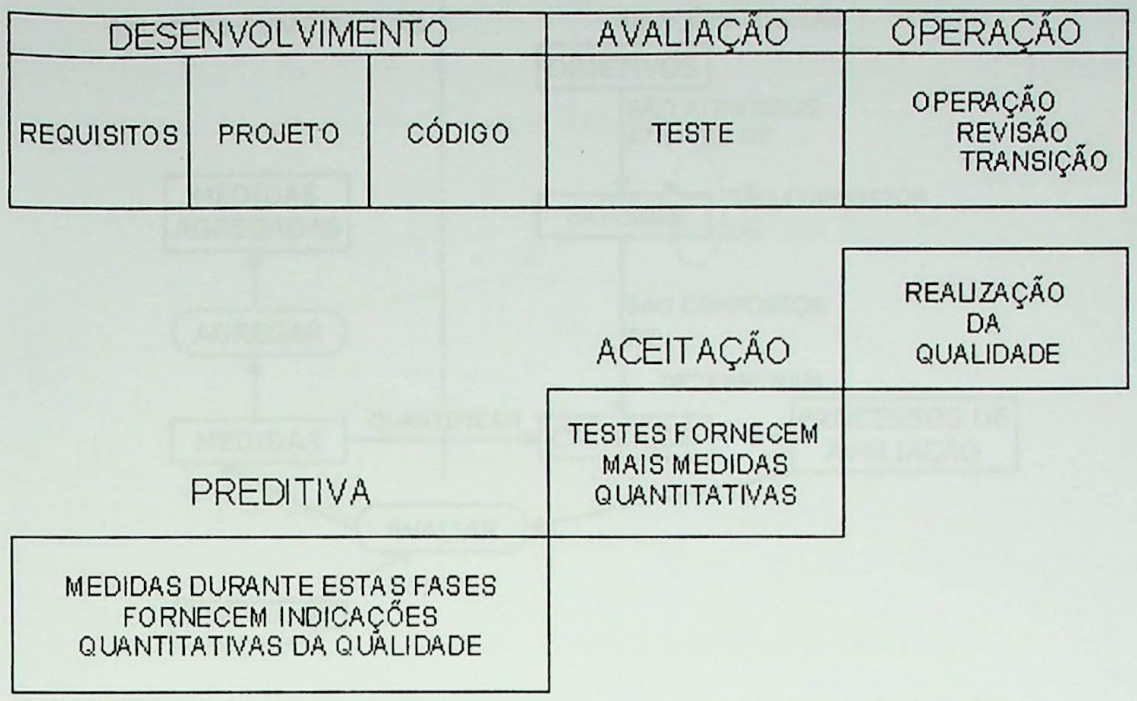


Figura 5.1 – Conceito de medidas de software.

A seguir é apresentado um método definido por Rocha [25] e que se baseia nos seguintes conceitos:

- Objetivos da qualidade – que são propriedades gerais que o produto deve possuir.
- Fatores de qualidade do produto – que determinam a qualidade do ponto de vista dos diferentes usuários do produto (usuário final, mantenedores, etc.).
- Critérios – que são atributos primitivos possíveis de serem avaliados.
- Processos de avaliação – que determinam o processo e os instrumentos a serem usados de forma a se medir o grau de presença, no produto, de um determinado atributo.
- Medidas – que são o resultado da avaliação do produto, segundo os critérios.
- Medidas agregadas – que são o resultado da agregação das medidas obtidas ao se avaliar segundo os critérios, e quantificam os fatores.

A Figura a seguir mostra a estrutura desse método.

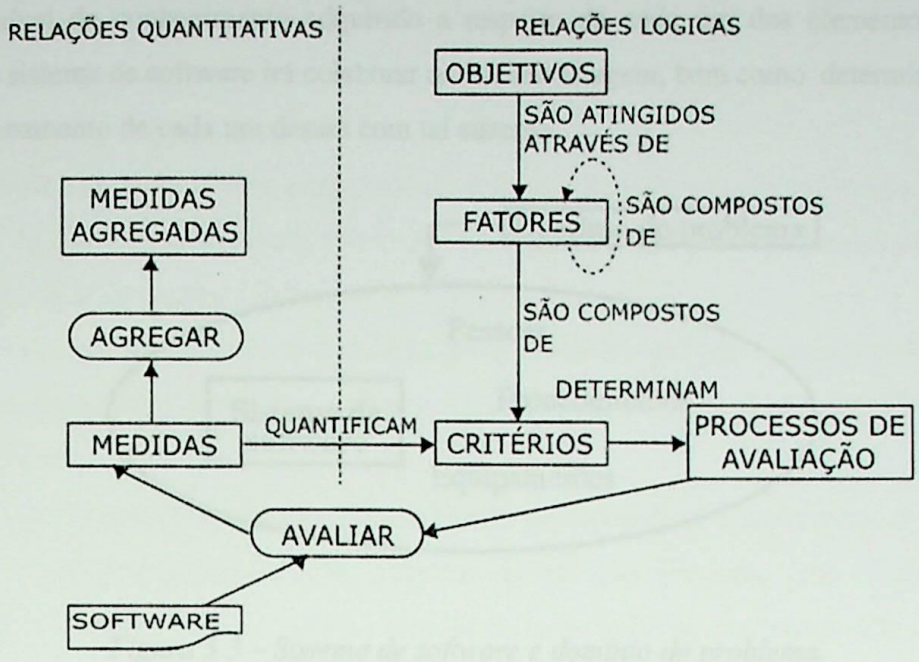


Figura 5.2 – Estrutura do método para avaliação da qualidade de software

5.4. Sistema de software e domínio do problema

O sistema de software se origina da combinação de três elementos básicos:

- pessoas – usuários, operadores, desenvolvedores, etc.
- hardware – computadores, equipamentos de conectividade e interconectividade, e equipamentos eletromecânicos.
- Procedimentos – etapas que definem o comportamento e/ou uso dos dois elementos anteriores.

Um quarto elemento poderá ainda ser adicionado à origem de um novo sistema de software. Tal elemento seriam outros sistemas de software.

O domínio do problema consiste em um ambiente onde o sistema de software está relacionado com os elementos citados acima e que de alguma maneira o afeta. Além disso, o domínio do problema poderá conter ou estar contido em outros domínios do problema (Figura 5.3).

No desenvolvimento de um software é necessária a investigação do domínio do problema desse software e suas responsabilidades dentro dele. Essa investigação para identificar as características do domínio do problema nem sempre é trivial, consistindo em um dos maiores problemas encontrados pelos analistas [27].

O nível de conhecimento adquirido a respeito de cada um dos elementos que irão compor o sistema de software irá colaborar na sua modelagem, bem como determinará o grau de relacionamento de cada um desses com tal sistema.

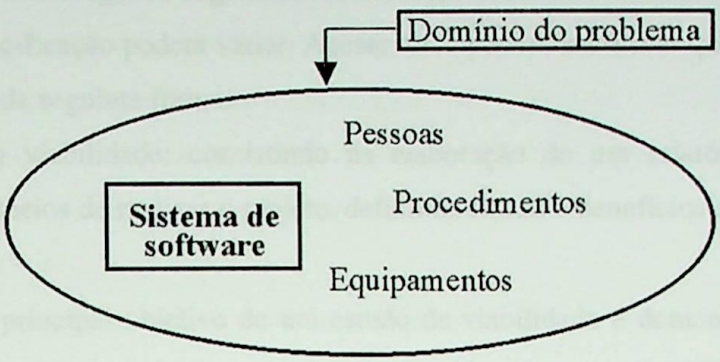


Figura 5.3 – Sistema de software e domínio do problema.

5.5. Ciclo de vida do software / Paradigmas da engenharia de software

O ciclo de vida do software consiste uma seqüência de fases ou etapas que se iniciam na sua concepção, e terminam na sua descontinuidade.

De uma maneira genérica, independentemente do paradigma, da área de aplicação, do tamanho e da complexidade do software, o seu ciclo de vida contém três grandes fases: análise, projeto e desenvolvimento, e uso e manutenção (figura 5.4) [28].

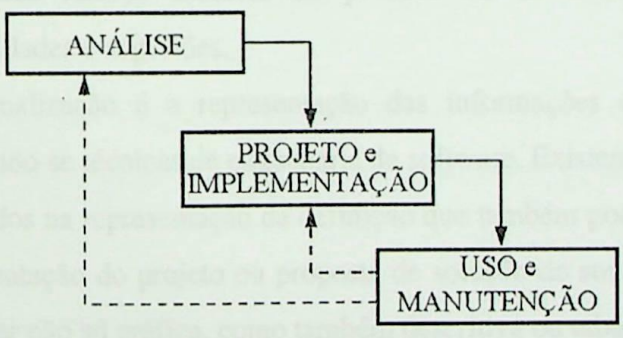


Figura 5.4 – Visão geral do ciclo de vida de um software

A fase de análise focaliza o "*o que*", e procura identificar quais são:

- os requisitos do sistema
- as informações que serão processadas
- as funções desejadas
- o desempenho esperado

- as interfaces que devem ser estabelecidas
- as restrições do projeto
- os critérios de validação

Conforme a metodologia de engenharia de software adotada, as técnicas utilizadas durante a fase de especificação podem variar. Apesar disto podemos afirmar que a fase de análise é subdividida da seguinte forma:

- Estudo da viabilidade: consistindo na elaboração de um relatório indicador dos possíveis meios de realizar o projeto, definindo custos e benefícios de cada alternativa [29].
    - O principal objetivo de um estudo de viabilidade é demonstrar o retorno do investimento de uma determinada solução proposta [30]. Consideram-se diferentes meios para a obtenção da solução de um problema, com seus custos e benefícios, observando-se o impacto técnico, político, legal, operacional e social. Certamente cada alternativa apresenta necessidades distintas e também oferece seus resultados específicos. Essa relação de dispêndios e resultados é comparada com as relações das outras alternativas. A melhor relação, não necessariamente a mais econômica, é a que deve ser adotada.
- Especificação: possui dois momentos distintos, que se podem denominar *definição* ou *especificação inicial* e *formalização da especificação.*
    - A definição consiste em uma especificação detalhada da função e dos requisitos do software. Isto significa que os desenvolvedores deverão obter junto aos futuros usuários do produto ou do software quais são suas necessidades e sugestões.
    - A formalização é a representação das informações obtidas na definição, utilizando-se técnicas de engenharia de software. Existem diferentes diagramas utilizados na representação da definição que também poderão ser utilizados na representação do projeto ou proposta de solução do software. A formalização pode ser não só gráfica, como também descritiva ou tabulada.

A fase de projeto e implementação focaliza o "*como*". Seus objetivos são:

- estabelecer a estrutura do software, através da definição das estruturas de dados e arquitetura do software;
- definir como os detalhes procedimentais deverão ser implementados
- a linguagem de programação a ser utilizada
- a forma de realização dos testes

Mais uma vez nesta fase, as técnicas aplicadas podem variar de acordo com a metodologia utilizada, mas em geral ela é composta das seguintes partes:

- Projeto de software: apresenta os requisitos do software, obtidos na definição e complementados com aspectos técnicos, num conjunto de representações que descrevem a estrutura de dados, a arquitetura, o procedimento algorítmico e as características de interface ou comunicação entre o sistema e os usuários.
- Codificação: consiste na conversão das representações do projeto em uma linguagem artificial que resulte em um conjunto de instruções que possam ser interpretadas e executadas pelo computador.
- Testes: após a codificação, o produto software deverá ser testado primeiramente pela equipe de desenvolvedores e depois pelos futuros usuários. Tal procedimento é importante e indispensável, uma vez que poderá haver erros tanto de codificação quanto de função.

Por fim, a fase de uso e manutenção refere-se às modificações que podem ocorrer no uso do software após ele ter entrado em funcionamento. Essas alterações podem ser para a correção de defeitos, acréscimos exigidos pelo cliente ou pela legislação e ainda adaptações exigidas à medida que o ambiente de software evolui, numa melhora contínua.

5.6. Técnicas utilizadas pela engenharia de software

Inúmeras são as técnicas utilizadas pela engenharia de software. Essas técnicas podem compor diversas metodologias de desenvolvimento de software, e a maneira como serão empregadas depende da metodologia escolhida, podendo ser classificadas em técnicas clássicas e técnicas de orientação a objeto.

Entretanto todas as técnicas partem da premissa "dividir para conquistar". Tal premissa nada mais é que do dividir problemas complexos em problemas menores para resolvê-los em partes.

5.6.1. Técnicas clássicas

As técnicas clássicas podem ser subdivididas em técnicas estruturadas e técnicas não estruturadas.

a) Técnicas estruturadas

Nas técnicas estruturadas pode-se aplicar o princípio de dividir para conquistar de forma recursiva, quebrando as partes em partes ainda menores (figura 5.5).

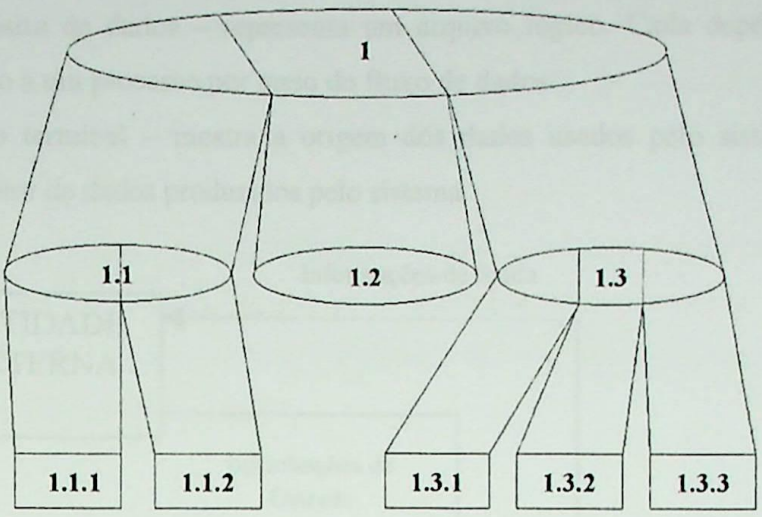


Figura 5.5 – Técnicas estruturadas e o princípio de dividir para conquistar

A equipe de desenvolvimento do software, em especial o analista de sistemas, deve observar se as partes juntas realmente compõem o problema inicial e se as interfaces entre as partes são simples, não aumentando a complexidade do problema ao invés de reduzi-la.

As técnicas estruturadas visam justamente garantir que as partes decompostas correspondam ao todo e que as interfaces entre essas partes sejam simples. Dentre as técnicas estruturadas pode-se destacar:

- Diagrama de fluxo de dados [31] – um diagrama de fluxo de dados apresenta os processos e o fluxo de dados entre eles. Em alto nível, é usado para mostrar eventos de negócios e as transações resultantes desses eventos, sejam elas feitas através de papéis ou por computador. Em nível mais baixo, é usado para mostrar programas ou módulos de programas e o fluxo de dados que passa entre eles.

Um diagrama de fluxo de dados é usado como o primeiro passo para um projeto estruturado. Apresenta o fluxo de dados global em um sistema ou programa. É principalmente uma ferramenta de análise de sistemas, para desenhar os componentes procedurais básicos e os dados que passam entre eles.

O diagrama de fluxo de dados é construído a partir de quatro elementos básicos:

- Fluxo de dados – condutor do fluxo de informações através dos processos de um sistema.
- Processo – é um componente procedural do sistema. Opera sobre (ou transforma) os dados. Nenhuma outra informação sobre o que faz o processo é mostrada no diagrama de fluxo de dados. Normalmente, os dados entram e saem de cada processo. Geralmente, existem múltiplos fluxos de dados entrando e saindo de um processo.

- Depósito de dados – representa um arquivo lógico. Cada depósito de dados é ligado a um processo por meio do fluxo de dados.
- Ponto terminal – mostra a origem dos dados usados pelo sistema e o último receptor de dados produzidos pelo sistema.

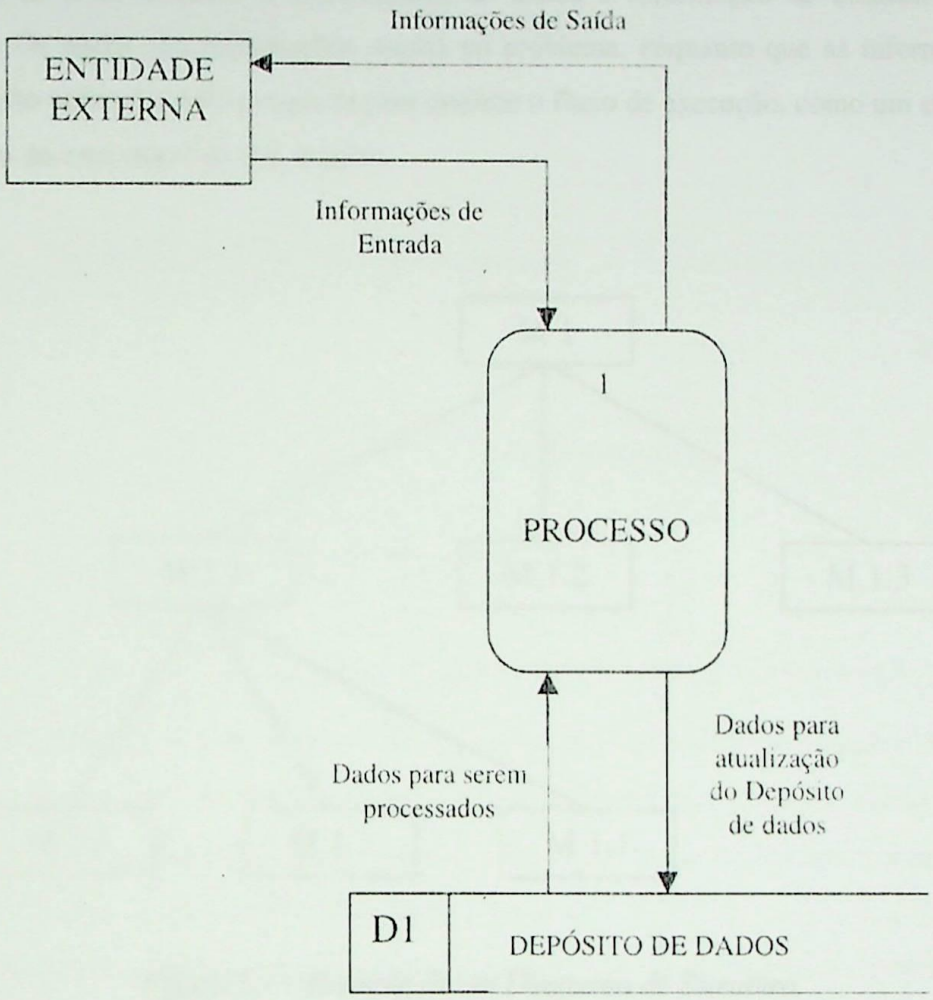


Figura 5.6 – Exemplo de um diagrama de fluxo de dados.

- Diagrama de estrutura dos módulos [31] – é uma forma de decomposição funcional. Juntamente com os diagramas de fluxo de dados, constitui uma metodologia de projeto estruturado usada comumente.

Os programas estruturados são organizados como uma hierarquia de módulos. O diagrama de estrutura é uma árvore ou diagrama hierárquico que define a estrutura global de um programa, mostrando os módulos de programa e suas inter-relações.

Os blocos básicos de construção dos diagramas de estrutura são retângulos e as setas que os interligam.

Um retângulo representa um módulo do programa. Cada módulo possui suas especificidades limitadas por um ponto de entrada e outro de saída. Estes módulos são identificados por um nome. Além deste nome, o diagrama de estrutura não fornece nenhuma informação sobre a natureza interna do módulo.

Já as setas indicam a transferência de dados e informação de controle entre os módulos. Os dados são informações usadas no problema, enquanto que as informações de controle são utilizadas pelo programa para orientar o fluxo de execução, como um sinalizador de erro ou de uma chave de fim arquivo.

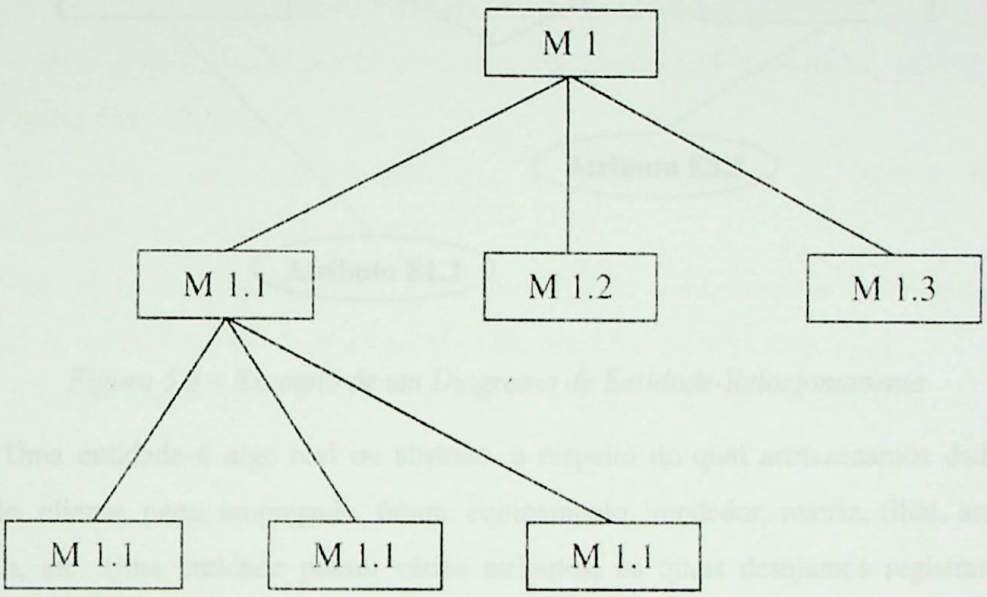


Figura 5.7 – Exemplo de um Diagrama de Estrutura.

b) Técnicas não estruturadas

Tais técnicas estão mais relacionadas com planejamento estratégico As técnicas não estruturadas não utilizam o principio de dividir para conquistar de maneira recursiva, e portanto não abordam o problema em diferentes níveis de abstração. Elas propõem apenas uma única quebra do problema em partes, o que significa que todos os detalhes são abordados de uma só vez. A seguir é apresentado um tipo de diagrama que aborda os fundamentos de tais técnicas:

- Diagrama de Entidade-Relacionamento [31] – consiste de uma técnica de modelagem de dados em que esses são considerados independentemente dos processos que os transformam.

Conseqüentemente o DER concentra-se apenas nos dados, representando uma rede de dados de um determinado sistema. O DER é uma ferramenta gráfica que representa o planejamento e a descrição desses dados. Em outras palavras, ele representa os relacionamentos entre os conjuntos de entidades do sistema [9] (figura 5.8).

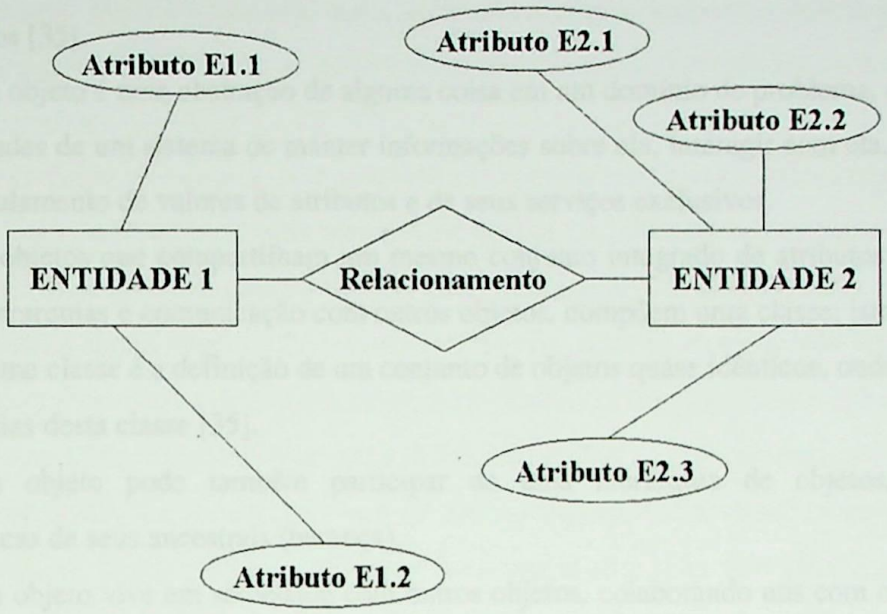


Figura 5.8 – Exemplo de um Diagrama de Entidade-Relacionamento

Uma entidade é algo real ou abstrato, a respeito do qual armazenamos dados. Por exemplo, cliente, peça, empregado, fatura, equipamento, vendedor, matriz, filial, armazém, depósito, etc. Uma entidade possui vários atributos, os quais desejamos registrar, como endereço, modelo, idade, número, tipo, nome, cidade, etc.

Um tipo de entidade é abreviado para entidade. Com esta abreviação, podemos então descrever os dados em termos de entidades e atributos. A diferença entre os dois é que as informações sobre uma entidade são armazenadas em múltiplos tipos de dados. Não se armazenam informações sobre um atributo em múltiplos tipos de itens de dados. Se um tipo de item de dados, que vem sendo chamado de atributo, requer informações armazenadas sobre ele, diferentes de seu valor, então esse item será na realidade uma entidade.

Uma entidade é graficamente representada por um retângulo, e a interligação entre duas ou mais entidades denomina-se relacionamento. Às vezes, o retângulo é usado também para mostrar um tipo de registro. Contudo, a finalidade deste modelo de dados é a representação dos dados, sem levar em consideração sua representação no computador. Podemos decidir representá-lo sem a estrutura convencional de registro.

5.6.2. Técnicas de orientação a objetos

No Paradigma Clássico o software é dividido em funções e dados enquanto que no paradigma da Orientação a Objetos o software é constituído de uma coleção de objetos que se inter-relacionam. Os objetos incorporam tanto o comportamento funcional como os dados manipulados [35].

Um objeto é uma abstração de alguma coisa em um domínio de problema, exprimindo as capacidades de um sistema de manter informações sobre ela, interagir com ela, ou ambos; um encapsulamento de valores de atributos e de seus serviços exclusivos.

Os objetos que compartilham um mesmo conjunto integrado de atributos e serviços, mesmas hierarquias e comunicação com outros objetos, compõem uma classe; isto é, pode-se dizer que uma classe é a definição de um conjunto de objetos quase idênticos, onde os objetos são instâncias desta classe [35].

Um objeto pode também participar de uma hierarquia de objetos, herdando características de seus ancestrais (herança).

Um objeto vive em sociedade com outros objetos, colaborando uns com os outros na execução de operações mais complexas, trocando mensagens (comunicação).

A comunicação entre os diversos objetos e classes de um sistema surge da própria necessidade de execução de uma tarefa quando um objeto sozinho não consegue alcançar o objetivo, ou seja, seus serviços são insuficientes para a realização da tarefa. Os objetos armazenam informações sobre si mesmos (seus atributos e serviços), e quando não podem executar um determinado serviço ou necessitam de uma outra informação, solicitam-na de outro objeto:

um objeto não pode ou não sabe fazer, mas sabe quem pode ou quem sabe.

A abordagem orientada para objetos está baseada numa visão do mundo como um sistema de agentes cooperadores. As tarefas em um sistema orientado para objeto são iniciadas por um objeto enviando uma requisição para outro objeto, pedindo a ele que realize uma de suas operações, ou que ele revele algumas de suas informações. A primeira requisição encadeia uma longa e complexa cadeia de requisições, fazendo com que os objetos sejam agentes de seus próprios sistemas [36].

Portanto, pode-se definir um software desenvolvido pelo paradigma *Orientado a Objeto* como um conjunto dos elementos classe, objeto, herança e comunicação [35].

Desta forma:

Orientado para Objetos = Classes e Objetos + Herança + Comunicação

Metodologias e técnicas orientadas a objetos

No final dos anos 80 e meados dos anos 90 surgiram conjuntos de metodologias de desenvolvimento de software orientado a objetos.

Cada metodologia propunha sua própria linguagem de representação e processo de desenvolvimento. Em 1995 existiam cerca de 50 metodologias propostas. Apesar de toda esta diversidade, todas representavam princípios e conceitos comuns que caracterizam a tecnologia orientada a objetos.

A tecnologia orientada a objetos está fundamentada em características como:

a) Encapsulamento: é uma maneira de ocultar as informações. O conhecimento ou inteligência encapsulada em um objeto possibilita a separação dos seus aspectos externos dos detalhes internos da implementação daquele objeto. Os aspectos externos são acessíveis por outros objetos, enquanto os detalhes internos ficam então ocultos dos demais objetos.

Pode-se afirmar que o objeto tem um lado particular, ou seja, somente ele é que sabe como realizar operações, sendo que *como* ele realiza tais operações não diz respeito a outras partes do sistema. Logo, os objetos são livres para mudar seus lados particulares sem afetar o resto do sistema [36]. Um objeto encapsula inteligência, que é representada no software por dados e processos [35].

O encapsulamento permite então que o modelo seja constituído de objetos que são vistos de fora e de maneira que cada um seja entendido em termos de seu procedimento intrínseco, representado pelos serviços que pode executar *(figura 5.9)[35]*.

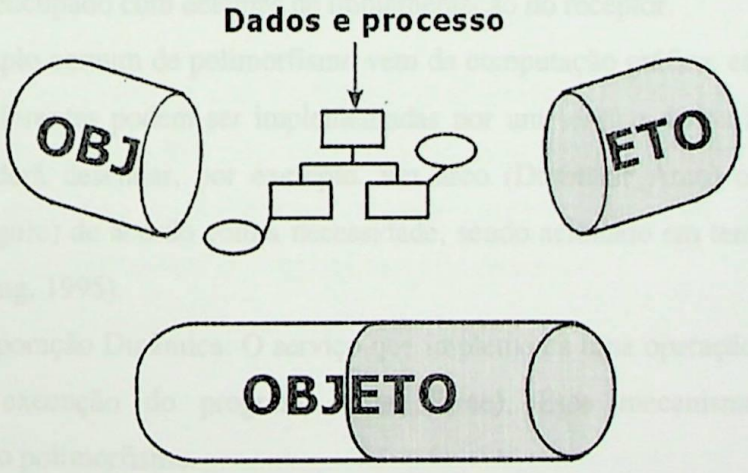


Figura 5.9 – Um objeto encapsula inteligência

b) Abstração: é o fato de se concentrar no que um objeto é e faz, antes de decidir como ele é implementado. O uso da abstração preserva a liberdade de se tomarem decisões, evitando tanto quanto possível comprometimentos prematuros com detalhes [37].

c) Compartilhamento: dirige a criação de estruturas de herança, onde dados e serviços que são compartilhados por um grupo de classes são descritos apenas uma vez, numa classe que é um antecessor comum para cada classe no grupo.

As técnicas orientadas a objetos promovem o compartilhamento em diversos níveis. A herança da estrutura de dados e do seu comportamento permite que a estrutura comum seja compartilhada por diversas subclasses semelhantes sem redundâncias. O compartilhamento de código com utilização de herança também é possível, constituindo-se em uma das principais vantagens das linguagens baseadas em objetos. Além disso, o desenvolvimento baseado em objetos oferece a possibilidade de reutilização de modelos e códigos em projetos futuros [37].

d) Ênfase na Estrutura de Objetos e não na Estrutura de Procedimentos. A tecnologia orientada a objetos preocupa-se em especificar o que um objeto é, e não como ele é utilizado. Os usos de um objeto são altamente dependentes dos detalhes de aplicação e freqüentemente mudam durante o desenvolvimento. À medida que os requisitos evoluem, as características de um objeto permanecem muito mais estáveis do que os modos como ele é utilizado, e por causa disso, os softwares construídos com base na estrutura de objetos são mais estáveis a longo prazo [37].

e) Polimorfismo: Consiste na habilidade de adquirir mais de uma *forma.* Um atributo de um objeto pode ter mais de um conjunto de valores, e uma operação pode ser implementada por mais de um serviço. Com o polimorfismo uma mensagem recebida por um objeto pode resultar em ações diferentes por parte desse objeto, ficando o objeto emissor da mensagem despreocupado com detalhes de implementação do receptor.

Um exemplo comum de polimorfismo vem da computação gráfica, em que operações com desenhos diferentes podem ser implementadas por um serviço denominado *Desenhar.* Esse serviço poderá desenhar, por exemplo, um arco (Desenhar_Arco) ou um retângulo (Desenhar Retângulo) de acordo com a necessidade, sendo acionado em tempo de execução do programa (Kung, 1995).

f) Incorporação Dinâmica: O serviço que implementa uma operação só é conhecido em tempo de execução do programa (Run Time). Esse mecanismo possibilita a implementação do polimorfismo.

Portanto, como o conceito de objeto engloba modularidade e encapsulamento, e as relações de herança facilitam as alterações e reutilizações, podem-se gerar softwares mais flexíveis para se alterar, valorizando a reutilização, a modularidade e o encapsulamento, e fazendo com que o degrau entre análise, projeto e implementação seja bem mais suave do que nas metodologias do Paradigma Clássico [35].

5.7. O ciclo de vida dos sistemas

Segundo Laudon [32] o desenvolvimento real de uma solução de sistema de informação pode seguir muitos rumos. A solução pode exigir um grande *mainframe* central interligando 20.000 pessoas ou um computador pessoal tipo *laptop*, programação e testes elaborados, ou um simples pacote de edição de textos e imagens. O problema a ser resolvido pode ser totalmente estruturado ou apenas semiestruturado. Um problema estruturado é aquele para o qual a solução é repetitiva, rotineira e envolve um procedimento definido que pode ser usado sempre que o mesmo problema é encontrado. No caso de um problema semi-estruturado, somente partes dele têm uma resposta imediata, proveniente de um procedimento definido. Dependendo do tamanho, escopo, complexidade e características da empresa, tipos diferentes de sistemas exigem diferentes abordagens para serem desenvolvidos.

Alguns métodos englobam uma abordagem mais formal de projeto de solução que outros. Alguns exigem funções claramente demarcadas entre os usuários finais e os especialistas técnicos; em outros, essa separação não é tão nítida. O que existe de comum em todas elas é a metodologia básica de solução de problemas.

5.7.1. O ciclo de vida de sistemas tradicionais

A metodologia mais antiga de construção de sistemas de informação é denominada ciclo de vida de sistemas tradicional. Esse método de desenvolver sistemas ainda é o método predominante na construção de grandes e médios sistemas baseados em *mainframes* atualmente.

A metáfora do "ciclo de vida" subdivide o desenvolvimento de um sistema em um conjunto formal de estágios, de modo semelhante ao ciclo de vida de um ser humano ou de outro organismo que possa ser dividido em estágios – com um início, um meio e um fim. O ciclo de vida do desenvolvimento de sistemas tem seis estágios:

Estagio 1 – Definição do projeto

Estagio 2 – Estudo do sistema

Estagio 3 – Projeto

Estagio 4 – Programação

Estagio 5 – Instalação

Estagio 6 – Pós-implementação

Cada estágio tem atividades vinculadas que devem ser completadas antes que o estágio seguinte comece. Assim, o sistema deve ser desenvolvido seqüencialmente, estágio por

estágio. Alguns acordos entre o pessoal técnico e os especialistas empresariais são necessários para marcar a conclusão de cada estágio.

Outra característica da metodologia do ciclo de vida é sua nítida e formal divisão do trabalho entre os especialistas empresariais e os especialistas técnicos. A maior parte do projeto de solução é entregue a técnicos, como analistas de sistemas e programadores. Os analistas de sistemas são responsáveis pela análise de problemas dos sistemas existentes e pelas especificações de projetos de soluções. Os programadores são responsáveis pela codificação e testes dos componentes de software dos sistemas. Tanto os analistas quanto os programadores utilizam informações e *feedback* proporcionado pelos especialistas empresariais para orientar seu trabalho, mas os especialistas empresariais desempenham um papel relativamente passivo.

5.7.2. O relacionamento do ciclo de vida com a solução de problemas

Os estágios do ciclo de vida correspondem até certo ponto às etapas de nossa metodologia de solução de problemas. A figura 5.10 mostra os estágios do ciclo de vida, as etapas correspondentes da solução de problemas e a divisão do trabalho apropriada entre os usuários e os especialistas técnicos ao longo do ciclo. Observe-se que nem sempre existe uma perfeita correlação um para um entre os estágios da solução de problemas e o ciclo de vida de sistemas, mas eles seguem um processo similar. O estágio de definição de projetos do ciclo de vida analisa se um problema existe de fato e se ele necessita de análise e pesquisa mais aprofundadas. Se isso ocorrer, será iniciado um projeto formal para construir um novo sistema de informação ou para modificar um sistema já existente. Assim, esse estágio incorpora alguns aspectos da primeira etapa da solução de problemas.

O estágio de estudo do problema incorpora parte da primeira etapa da análise do problema e também as duas etapas seguintes. As atividades durante esse estágio focalizam a descrição e a análise dos problemas de sistemas já existentes, especificando objetivos de soluções, descrevendo soluções possíveis e avaliando diversas alternativas de solução. Também são examinadas as restrições das soluções e a viabilidade de cada alternativa.

Todas as informações coletadas no estudo dos sistemas existentes e das entrevistas com os especialistas empresariais serão utilizadas para especificar os requisitos de informação. Uma solução de sistema deve identificar quem precisa de qual informação, onde, quando e como. Os requisitos devem ser especificados detalhadamente, até o último dado, e também devem levar em conta os procedimentos e as restrições organizacionais, bem como o hardware, o software e os dados.

É importante ressaltar que uma solução de sistema de informação não funcionará se não for erguida em torno do conjunto de requisitos correto. Caso contrário, o sistema terá de ser revisto ou descartado, muitas vezes com grande desperdício de tempo e dinheiro. A obtenção dos requisitos exatos talvez seja o aspecto mais difícil da elaboração de um sistema.

Depois que os requisitos tiverem sido obtidos, o estágio de projeto pode prosseguir. A esta altura, são geradas as especificações lógicas do projeto. As ferramentas de projeto e de documentação, como o diagrama de fluxo de dados, o dicionário de dados e o fluxograma de sistema, provavelmente serão empregados porque o ciclo de vida põe muita ênfase no detalhamento das especificações e na documentação escrita. Os especialistas empresariais e o pessoal técnico devem rever e aprovar esses documentos para que o projeto físico e a programação possam ter início.

Durante o estágio de programação, especificações detalhadas de projeto para arquivos, processos, relatórios e transações de entradas são traduzidas em software para o sistema de informação proposto. Os especialistas técnicos redigem os programas sob medida utilizando uma linguagem convencional de programação, como COBOL, FORTRAN, ou uma linguagem de quarta geração de alto nível.

Durante o estágio de instalação, o software é testado para se garantir que funciona adequadamente dos pontos de vista técnico e funcional, ou empresarial, e o sistema antigo é convertido para o novo. Os especialistas empresariais e técnicos são treinados na utilização do novo sistema. As atividades relacionadas a programação, testes, conversão e treinamento correspondem à quinta e última etapa de nossa metodologia de solução de problemas.

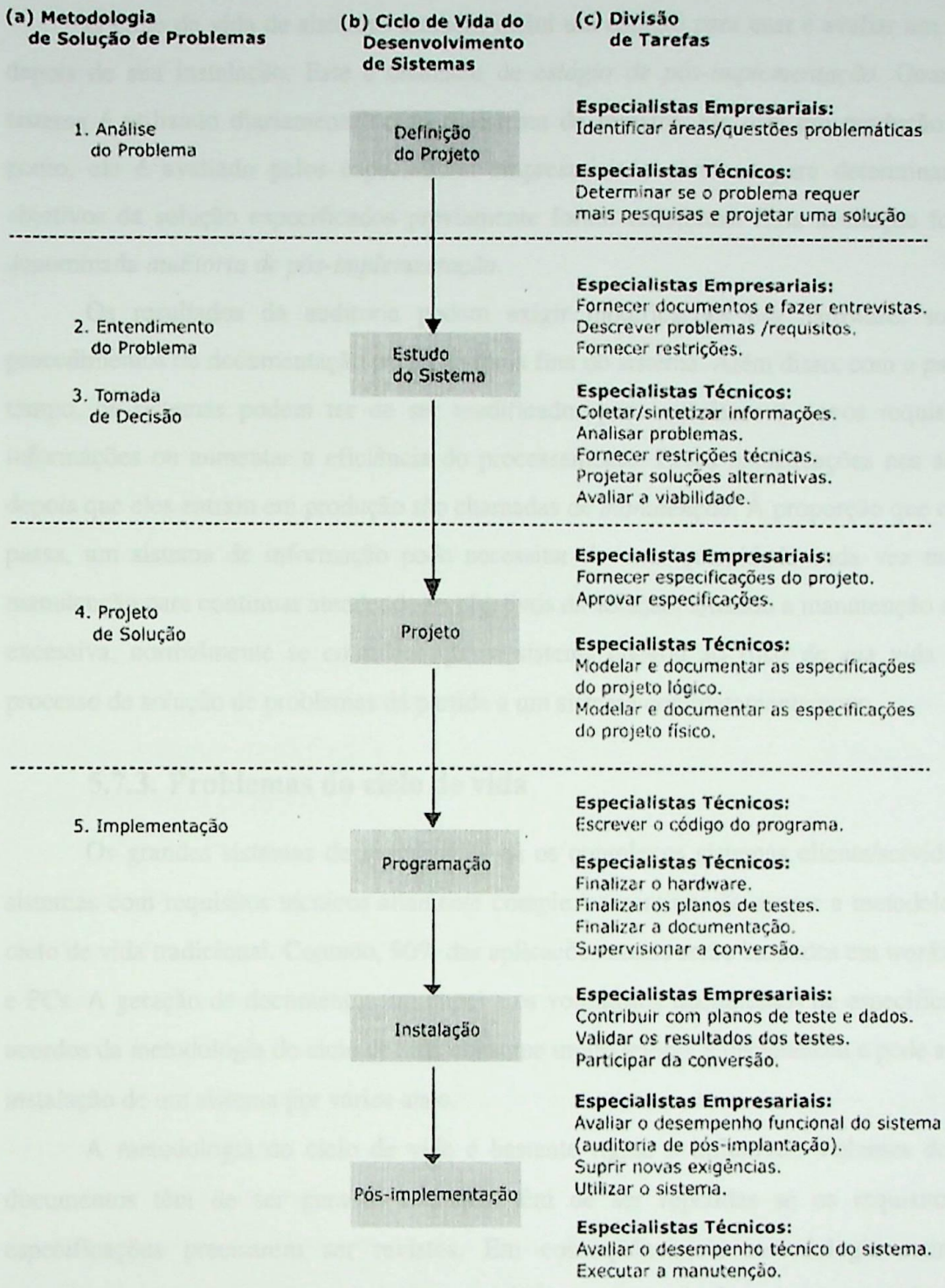


Figura 5.10 – O Ciclo de Vida do Desenvolvimento de Sistemas

0 ciclo de vida do desenvolvimento de sistemas divide o processo em uma série de estágios semelhantes ao ciclo de vida de um ser humano. Esses estágios correspondem às cinco etapas do processo de solução de problemas discutido nos capítulos precedentes. Como este diagrama mostra, o ciclo de vida de um sistema pode ser um processo muito formal, em que os usuários finais têm uma participação relativamente passiva.

O ciclo de vida de sistemas também inclui um estágio para usar e avaliar um sistema depois de sua instalação. Este é chamado de *estágio de pós-implementação*. Quando um sistema é utilizado diariamente como o sistema de registro, ele está em produção. Nesse ponto, ele é avaliado pelos especialistas empresariais e técnicos para determinar se os objetivos da solução especificados previamente foram satisfeitos. Essa avaliação formal é denominada *auditoria de pós-implementação*.

Os resultados da auditoria podem exigir modificações em hardware, software, procedimentos ou documentação para a sintonia fina do sistema. Além disso, com o passar do tempo, os sistemas podem ter de ser modificados para satisfazer a novos requisitos de informações ou aumentar a eficiência do processamento. Essas modificações nos sistemas depois que eles entram em produção são chamadas de *manutenção*. À proporção que o tempo passa, um sistema de informação pode necessitar de uma quantidade cada vez maior de manutenção para continuar atendendo os objetivos da solução. Quando a manutenção se torna excessiva, normalmente se considera que o sistema chegou ao final de sua vida útil. O processo de solução de problemas dá partida a um sistema completamente novo.

5.7.3. Problemas do ciclo de vida

Os grandes sistemas de *mainframes* ou os complexos sistemas cliente/servidor e os sistemas com requisitos técnicos altamente complexos continuarão a usar a metodologia do ciclo de vida tradicional. Contudo, 90% das aplicações atuais serão baseados em *workstations* e PCs. A geração de documentos em papel e os volumosos documentos de especificações e acordos da metodologia do ciclo de vida consome muito tempo, é dispendiosa e pode atrasar a instalação de um sistema por vários anos.

A metodologia do ciclo de vida é bastante rígida e inflexível. Volumes de novos documentos têm de ser gerados e etapas têm de ser repetidas se os requisitos e as especificações precisarem ser revistos. Em conseqüência, a metodologia estimula o congelamento das especificações logo no início do processo de desenvolvimento. Podem ser feitas modificações mais tarde, mas então elas serão muito dispendiosas. Se a solução de um sistema não puder ser visualizada imediatamente – como ocorre freqüentemente com as aplicações orientadas a decisões, essa metodologia não será de grande ajuda.

5.7.4. Alternativas do ciclo de vida

Outros meios de construir sistemas de informação podem superar algumas das limitações do ciclo de vida. Eles também se fundamentam na metodologia de solução de

problemas, que já definimos. Entretanto, os meios de estabelecer requisitos, de desenvolver software e finalizar a solução do sistema diferem dos meios do ciclo de vida tradicional, e os especialistas empresariais desempenham um papel muito mais importante no processo do projeto da solução. As alternativas mais importantes para o ciclo de vida tradicional são a prototipagem, o uso de pacotes de software, o desenvolvimento de quarta geração e a terceirização.

A prototipagem engloba a construção de um sistema experimental ou parte de um sistema de maneira rápida e pouco dispendiosa para que os usuários finais possam avaliá-lo. Na medida em que os usuários interagem com o protótipo, eles formam uma idéia melhor de quais são suas necessidades, e as características do sistema final podem ser adaptadas de acordo.

Os pacotes de software aplicativos representam uma alternativa à escrita de programas e ao desenvolvimento de sistemas customizados em uma empresa. Em vez disso, a empresa pode adquirir um pacote de software no qual todos os programas já foram escritos e testados. Os pacotes de software são mais apropriados quando a solução de sistema de informação é necessária a muitas organizações e quando os pacotes de software para atender a essas necessidades estão no mercado.

O desenvolvimento de quarta geração é voltado para o desenvolvimento de sistemas de informação com pouca ou nenhuma assistência de especialistas técnicos. Trata-se de uma abordagem útil para pequenos sistemas de informação e para aplicações de computadores pessoais, como gerenciamento de arquivos ou aplicações gráficas. Grande parte do processo de projeto de solução pode ser executada pelos próprios usuários finais. Quando os usuários conhecem os requisitos, eles mesmos podem projetar seus sistemas de informação utilizando ferramentas de software de quarta geração.

A terceirização envolve a utilização de uma empresa externa que executa o desenvolvimento (ou a operação) dos sistemas de informação de uma organização. A organização desenvolve aplicações utilizando os recursos da empresa externa em lugar de seu próprio pessoal de sistemas de informação. Essa abordagem é útil quando a organização não dispõe de recursos financeiros ou técnicos para desenvolver sistemas por si mesma.

5.7.4.1. A alternativa da prototipagem

Um protótipo é um modelo preliminar de uma solução de sistema (ou parte de um sistema, como telas para entrada de dados) para os usuários finais interagirem e analisarem. O protótipo é construído rapidamente e sem grandes despesas, em alguns dias ou semanas, com

a utilização de software de PCs ou de ferramentas de software de quarta geração. Os usuários finais põem à prova o modelo experimental para verificar se ele atende aos requisitos. No processo, eles podem descobrir novos requisitos que possam ter passado despercebidos ou sugerir novas áreas a serem melhoradas. O protótipo é modificado, devolvido aos usuários outra e outra vez, até que atenda exatamente às necessidades.

Na prototipagem, o projeto de solução é menos formal do que na metodologia do ciclo de vida. Em vez de analisar detalhadamente um problema, a prototipagem gera um projeto de solução, presumindo-se que o desejado seja uma solução de aplicação. Os requisitos são determinados dinamicamente à medida que o protótipo vai sendo construído. A análise do problema, o entendimento do problema, a tomada de decisões e o projeto de solução são agrupados em uma só atividade.

A abordagem de prototipagem é mais explicitamente iterativa do que a metodologia do ciclo de vida tradicional. As etapas do desenvolvimento da solução podem ser repetidas muitas vezes. Diversamente do ciclo de vida tradicional, que deve capturar a versão correta de um sistema desde o início, a prototipagem encoraja a experimentação e as repetidas modificações de projeto. A prototipagem também é altamente interativa. Nela, os usuários finais trabalham diretamente com projetos de solução desde os estágios iniciais do processo de desenvolvimento.

Em comparação com o ciclo de vida tradicional, a prototipagem exige um envolvimento maior dos especialistas empresariais no processo de solução de problemas. Os especialistas empresariais devem manter estreito contato com os especialistas técnicos que desenham o protótipo. Com ferramentas de software de quarta geração ou baseadas em PCs, os usuários finais podem na verdade desenvolver o protótipo por eles mesmos. Eles também muitas vezes terão de decidir sobre novos melhoramentos a cada vez que o protótipo for revisto.

5.7.4.2. Etapas da prototipagem

Como a figura 5.11 mostra, a prototipagem envolve quatro etapas que incorporam as etapas da metodologia de solução de problemas:

1. Identificação dos requisitos iniciais: um especialista técnico ou analista trabalha rapidamente com o especialista empresarial para encontrar uma solução básica e determinar as necessidades de informação, O processo é mais rápido e menos formal do que na metodologia do ciclo de vida. Algumas etapas do projeto de solução são consolidadas em apenas uma.

2. Desenvolvimento de um protótipo operacional: é criado rapidamente um protótipo que funcione. Ele pode consistir apenas em telas *online* ou em relatórios para um sistema proposto, ou pode ser um sistema completo com arquivos de dados muito pequenos.
3. Utilização do protótipo: o usuário final trabalha com o protótipo para verificar se ele atende a suas necessidades. O usuário é solicitado a fazer recomendações para melhorar o protótipo.
4. Revisão e aperfeiçoamento do protótipo: com base nas recomendações do usuário final, o especialista técnico ou analista efetua uma revisão do protótipo. O ciclo então retorna à etapa 3. As etapas 2, 3 e 4 são repetidas até que o usuário esteja inteiramente satisfeito. O protótipo aprovado fornece as especificações finais para a solução de sistema de informação. Às vezes o próprio protótipo torna-se a versão final do sistema.

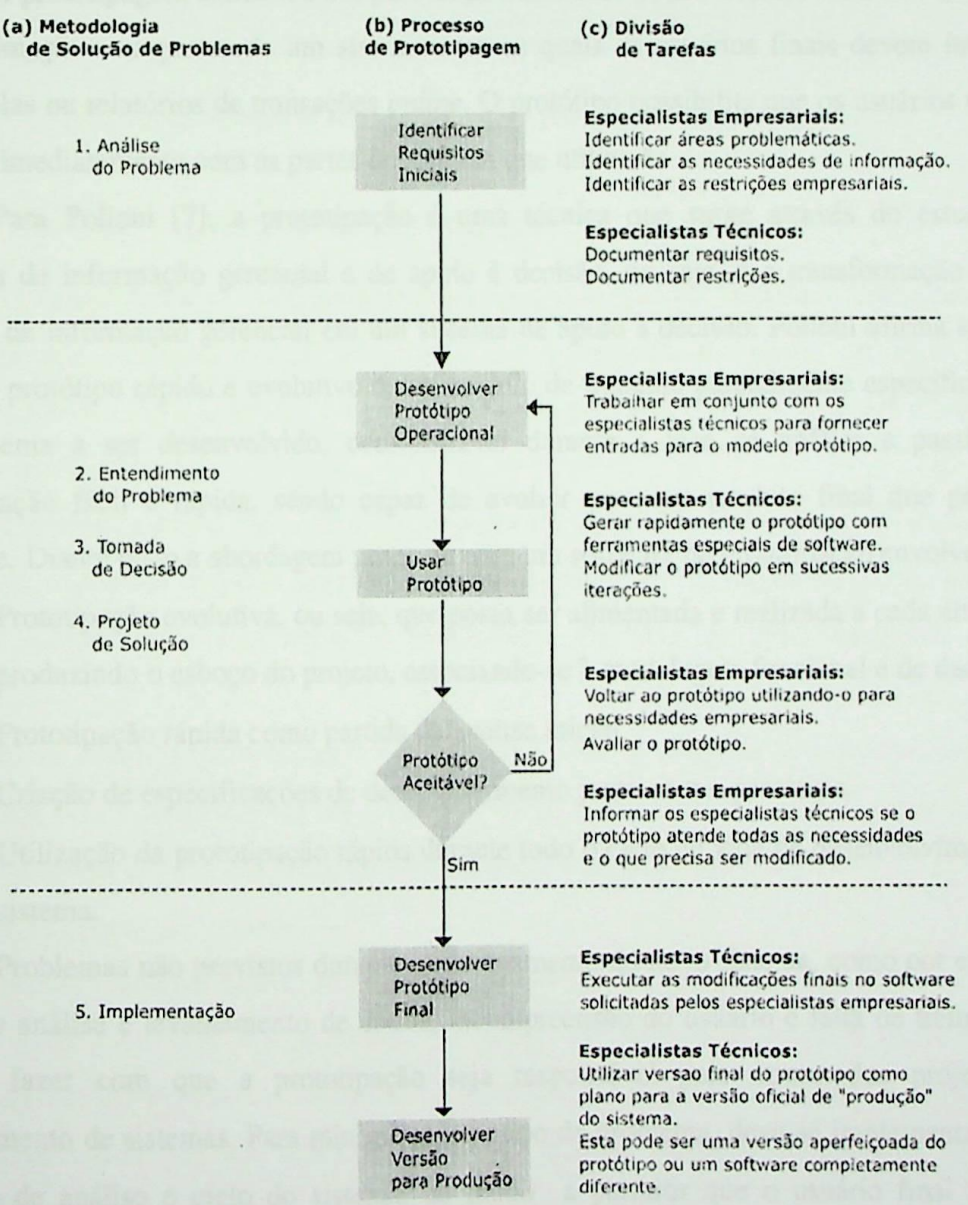


Figura 5.11 – Prototipagem: modo mais rápido de desenvolver um sistema

5.7.4.3. Quando usar a prototipagem

A prototipagem é mais eficaz quando os requisitos do usuário não são muito claros. Isso é uma característica de muitos sistemas orientados a decisões. Muitas vezes o sistema final não pode ser claramente visualizado porque o próprio processo decisório não chegou a ser completado. Por exemplo, uma empresa pode utilizar da prototipagem para agrupar os usuários que pensam "intuitivamente" e os que pensam "sistematicamente". Os pensadores intuitivos normalmente preferem imagens, gráficos, diagramas, linhas de tendências, enquanto os sistemáticos geralmente preferem informações exibidas quantitativamente, como datas, números e lugares. A vantagem de se trabalhar com um protótipo é que os especialistas empresariais podem utilizar um sistema que funciona como um mecanismo para facilitar o processo de solução de problemas, o que os ajuda a alcançar uma solução rapidamente.

A prototipagem também é útil para testar a interface com o usuário final de um sistema de informação - as partes de um sistema com as quais os usuários finais devem interagir, como telas ou relatórios de transações *online*. O protótipo possibilita que os usuários tenham contato imediatamente com as partes do sistema que utilizarão.

Para Polloni [7], a prototipação é uma técnica que surge através do estudo dos sistemas de informação gerencial e de apoio à decisão, ou seja, é a transformação de um sistema de informação gerencial em um sistema de apoio à decisão. Polloni afirma também que um protótipo rápido e evolutivo é um modelo de trabalho parcialmente especificado de um sistema a ser desenvolvido, demonstrável durante a fase de análise e passível de modificação fácil e rápida, sendo capaz de evoluir para um produto final que pode ser entregue. Diante disto a abordagem proposta para um software de prototipação envolve:

- Prototipação evolutiva, ou seja, que possa ser alimentada e realizada a cada alteração, produzindo o esboço do projeto, associando-se à modelagem funcional e de dados.
- Prototipação rápida como partida da analise inicial.
- Criação de especificações de desenvolvimento junto com o protótipo.
- Utilização da prototipação rápida durante todo o ciclo de vida de desenvolvimento do sistema.

Problemas não previstos durante o planejamento do novo sistema, como por exemplo erros de análise e levantamento de dados, incompreensão do usuário e falta de treinamento podem fazer com que a prototipação seja responsável pelo atraso dos projetos do planejamento de sistemas. Para minimizar esse tipo de problema, deve-se implementar ainda na fase de análise o ciclo do sistema, de modo a permitir que o usuário final tenha a

impressão de que o sistema está praticamente pronto, mesmo antes da sua implantação de fato [7]. A essa técnica damos o nome de *prototipação rápida de software* (figura 5.12).

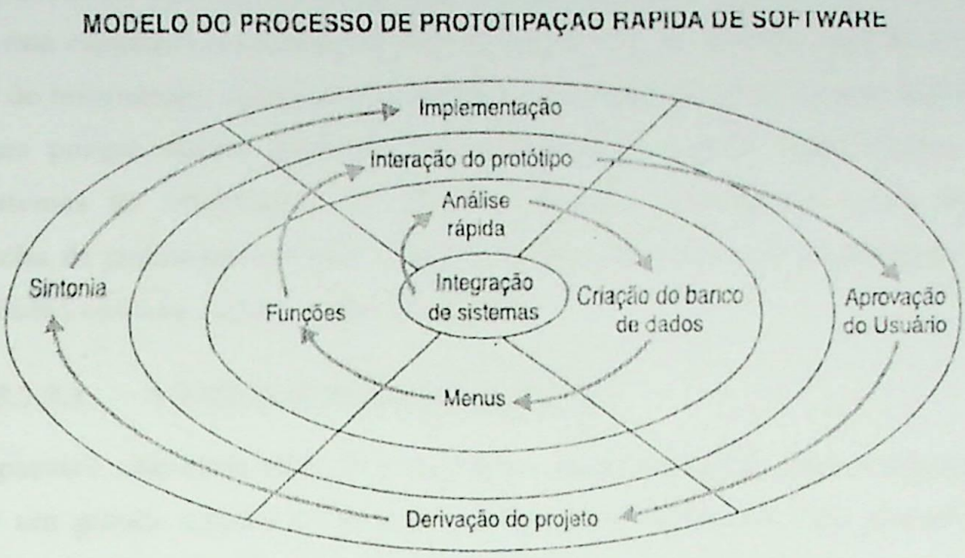


Figura 5.12 – Modelo do processo de prototipação rápida de software

5.7.4.4. Limitações da prototipagem

Alguns estudos já demonstraram que os protótipos que atendem inteiramente aos requisitos dos usuários podem ser criados em 10 a 20% do tempo estimado para o desenvolvimento convencional. Contudo, a prototipagem não é apropriada para certos tipos de sistemas de informação.

A prototipagem é mais eficaz para pequenas aplicações. Ela não pode ser aplicada facilmente a grandes sistemas baseados em mainframes com instruções de processamento e cálculos complexos; nesses casos, a metodologia do ciclo de vida tradicional é mais apropriada.

Além disso, a prototipagem não substitui toda a pesquisa e a análise detalhada necessárias para a construção de um sistema de informação. Grandes sistemas ainda precisam de uma completa análise e investigação do problema e especificações de requisitos para que a prototipagem possa ser iniciada.

Muitas vezes, atividades essenciais como testes e documentação são deixadas de lado por ser muito fácil criar um protótipo. Por vezes, um sistema protótipo é imediatamente convertido em um sistema de produção. Entretanto, em circunstâncias organizacionais reais, esse sistema pode não ser capaz de atender a grandes números de usuários, processar númerosas transações e manter grandes quantidades de registros.

5.7.5. Desenvolvendo soluções com pacotes de software

Os pacotes de software são programas pré-escritos, pré-codificados e comercialmente disponíveis que eliminam a necessidade de se escreverem programas quando se desenvolve um sistema de informação. Atualmente cada vez mais sistemas estão sendo construídos com esses pacotes porque alguns problemas encontrados por muitas organizações exigem os mesmos sistemas de informação ou sistemas bastante semelhantes como solução. Por exemplo, folha de pagamento, contas a pagar, contas a receber e processamento de pedidos são necessidades comuns a quase todas as empresas.

5.7.5.1. VANTAGENS DOS PACOTES

Os pacotes oferecem diversas vantagens, principalmente para empresas que não dispõem de um grande quadro de pessoal técnico em sistemas ou cujo pessoal carece dos conhecimentos técnicos necessários para desenvolver uma determinada aplicação. Os principais fornecedores de pacotes mantêm seu próprio quadro de pessoal de suporte técnico para proporcionar aos clientes orientação especializada depois que o sistema estiver instalado. Assim, uma empresa que adquire software em pacotes tem menos necessidade de manter seus próprios especialistas.

Uma vantagem relacionada é a economia de custos que as organizações podem alcançar com a aquisição de pacotes de software em vez de desenvolvê-los por si mesmas. Os pacotes também eliminam parte da necessidade de trabalhar e refazer as especificações de um sistema porque os usuários devem aceitar o pacote como ele é. Muitos recursos da solução do projeto já foram desenvolvidos, de modo que o comprador sabe precisamente quais são os recursos do sistema.

5.7.5.2. PACOTES E O PROCESSO DE PROJETO DE SOLUÇÕES

Como os pacotes se enquadram na metodologia de projeto de soluções? Como a figura 5.15 mostra, mesmo quando se considera um pacote, os construtores de sistemas têm de analisar o problema, especificar os objetivos da solução, considerar restrições e avaliar soluções alternativas. Durante esses processos, eles podem determinar se uma solução alternativa de pacote atenderá aos requisitos de informações. Em seguida, quando avaliarem as soluções alternativas, eles podem comparar a viabilidade de uma solução de pacote com outras opções de solução.

O processo de avaliação de pacotes muitas vezes se baseia nos resultados de uma Requisição Formal de Proposta (RFP). A RFP é uma lista detalhada de perguntas submetidas

aos fornecedores de pacotes de software. As perguntas são elaboradas visando a medir até que ponto cada pacote atende aos requisitos especificados durante o processo de projeto de solução.

Quando se decide por uma solução de pacote de software, a etapa de projeto da solução do problema é realizada em relação ao pacote. Os projetos lógico e físico não começam do nada, adaptados aos requisitos e especificações gerados antes. Em vez disso, o trabalho de projeto focaliza o ajuste dos requisitos do usuário e das especificações para atender às características do pacote. Em outras palavras, em vez de os usuários finais projetarem seu próprio controle de folha de pagamento, é utilizado o relatório de controle fornecido pelo pacote de folha de pagamento. Nesse sentido, a organização e os usuários finais têm menos controle sobre o resultado da forma e do projeto da solução.

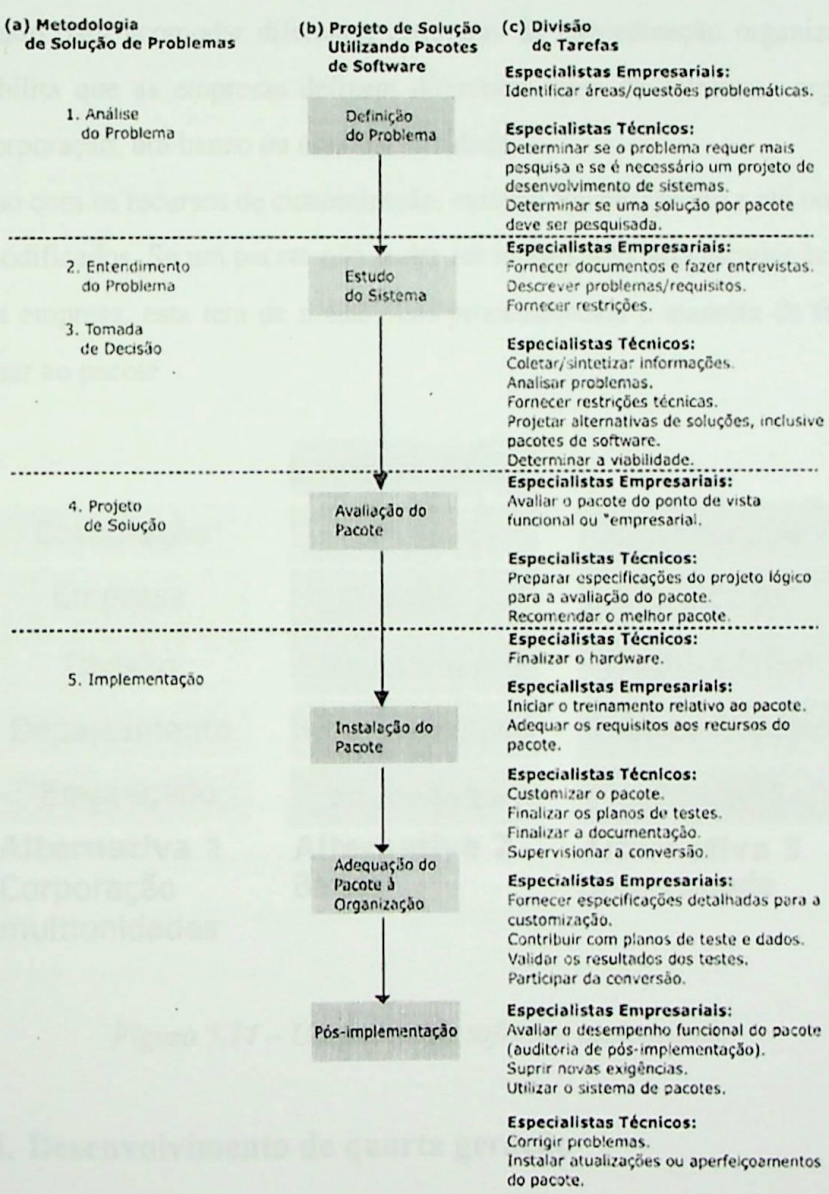


Figura 5.13 – Pacotes de softwares pré-escritos

5.7.5.3. DESVANTAGENS DOS PACOTES

A principal desvantagem dos pacotes de software é que muitas vezes eles não são capazes de satisfazer a todos os requisitos da organização. Cada organização trata até uma função padronizada como a folha de pagamento de forma um pouco diferente das outras organizações. Em conseqüência, podem ocorrer muitos casos em que um pacote não é capaz de satisfazer 100% dos requisitos de solução de uma empresa.

Os fornecedores de pacotes tentam resolver esse problema por meio da customização; isto é, o pacote contém recursos que permitem que ele seja modificado sem destruir a integridade do software. Por exemplo, o pacote pode conter áreas em seus arquivos ou bancos de dados, onde uma empresa pode acrescentar e manter seus próprios dados.

A figura 5.14 mostra como um grande pacote de software de folha de pagamento pode ser customizado para acomodar diferentes estruturas de subordinação organizacional. Esse pacote possibilita que as empresas definam diferentes tipos de estruturas organizacionais, como uma corporação, um banco ou uma universidade.

Mesmo com os recursos de customização, existem limites quanto a até onde os pacotes podem ser modificados. Se um pacote não puder ser modificado para atender às necessidades exclusivas da empresa, esta terá de mudar seus procedimentos e maneira de fazer negócios para se adequar ao pacote.

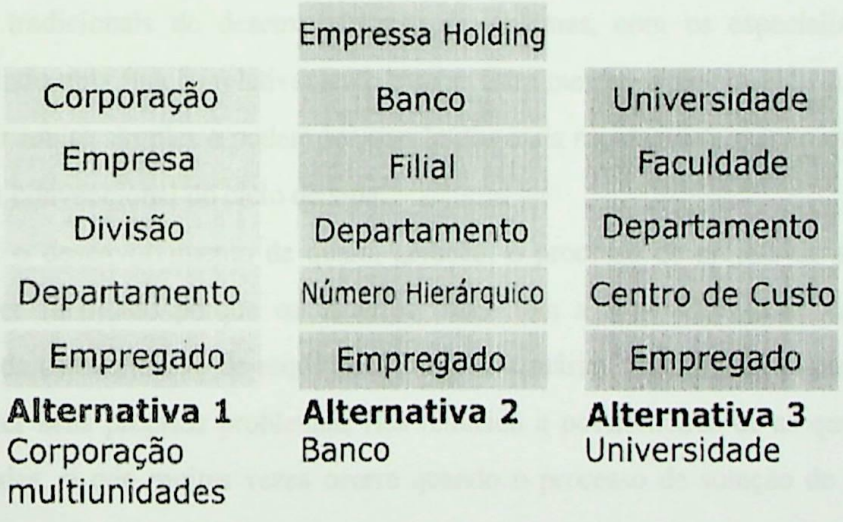


Figura 5.14 – Um pacote de software customizado

5.7.6. Desenvolvimento de quarta geração

Muitos sistemas de informação podem ser desenvolvidos pelos usuários finais com pouca ou nenhuma assistência de especialistas técnicos. Tal abordagem é denominada

desenvolvimento de quarta geração [32]. Ela incorpora algumas das ferramentas de software, como as linguagens de quarta geração, ferramentas de computadores pessoais (inclusive as ferramentas de criação de páginas da Web) e linguagens gráficas, que tornam possível que os usuários finais executem tarefas que antes exigiam especialistas treinados de sistemas de informação. (Como você deve lembrar, linguagens de quarta geração são linguagens de programação que não só são menos procedurais que as linguagens convencionais mas também possuem mais comandos semelhantes ao inglês, sendo, em vista disso, de utilização mais fácil para os especialistas não-técnicos.)

5.7.6.1. PROJETO DE SOLUÇÃO DE QUARTA GERAÇÃO

O desenvolvimento de quarta geração proporciona ao usuário final maior controle do processo de solução de problemas. Eles podem pesquisar e analisar um problema, especificar soluções alternativas, executar uma parte limitada do projeto lógico e físico e implementar a solução por si mesmos ou com a assistência de especialistas técnicos. Porém, para que os usuários finais possam desenvolver com sucesso aplicações utilizando dados corporativos com pouco ou nenhum envolvimento de especialistas técnicos, o banco de dados (de preferência um banco de dados relacional) deverá ter sido desenvolvido pelos especialistas.

A figura 5.15 mostra como o desenvolvimento de quarta geração afeta o processo de projeto de solução. Observe-se que o projeto de solução tende a ser menos formal do que nas abordagens tradicionais de desenvolvimento de sistemas, com os especialistas técnicos desempenhando uma função relativamente menor. Geralmente, os sistemas de quarta geração tendem a ser muito simples e podem ser concluídos mais rapidamente que os que utilizam a metodologia convencional do ciclo de vida.

Com o desenvolvimento de quarta geração, o processo de projeto de solução pode realmente ser facilitado porque os usuários finais têm a responsabilidade da análise do problema e da especificação de requisitos. Como os usuários finais estão na melhor posição para conhecer seus próprios problemas, fica reduzida a possibilidade de as questões serem mal entendidas, o que muitas vezes ocorre quando o processo de solução de problemas é dominado por especialistas técnicos.

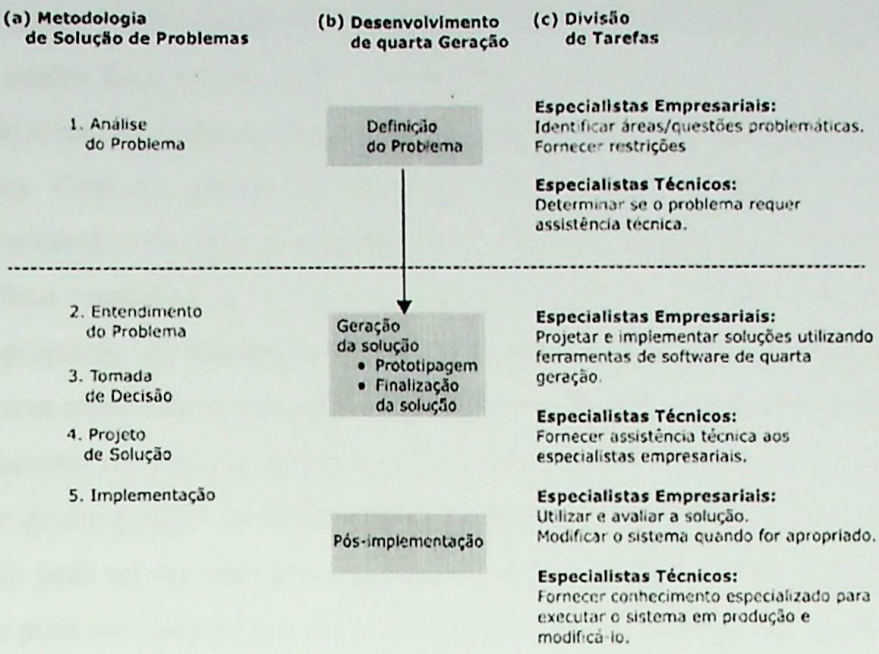


Figura 5.15 – Desenvolvimento de quarta geração

5.7.6.2. FERRAMENTAS DE SOFTWARE DE QUARTA GERAÇÃO

O desenvolvimento de quarta geração exige ferramentas de software de fácil utilização que podem ser empregadas somente pelos usuários finais ou por especialistas técnicos como auxílio à produtividade. A lista a seguir apresenta aqueles que são considerados os principais tipos de ferramentas de software para usuários finais:

1. Ferramentas de computadores pessoais
2. Linguagens de consulta e geradores de relatórios
3. Linguagens gráficas
4. Geradores de aplicações
5. Linguagens de programação de nível muito alto
6. Pacotes de software aplicativos

A figura 5.16 relaciona algumas ferramentas representativas e comercialmente disponíveis de cada categoria e indica as aplicações para as quais elas são mais apropriadas.

As ferramentas para computadores pessoais compreendem os produtos de software para PCs: planilhas eletrônicas, gerenciamento de bancos de dados, gráficos, editores de texto, navegadores da Web, ferramentas de criação de páginas da Web e software de comunicações. O software para PCs é especialmente adequado para o desenvolvimento pelos usuários finais por ter sido projetado para eles e não para especialistas técnicos, e porque os sistemas operacionais dos computadores pessoais são relativamente simples.

As linguagens de consulta (query languages) são linguagens de quarta geração de fácil utilização, usadas para acessar dados armazenados em bancos de dados ou arquivos. Os geradores de relatórios extraem dados de arquivos ou de bancos de dados para criar relatórios customizados. Com um gerador de aplicações, uma aplicação de sistema de informação completa (incluindo uma aplicação da Web) pode ser gerada sem programação customizada; o usuário final especifica as tarefas que devem ser feitas e o gerador cria o código de programa apropriado. As linguagens de programação de nível muito alto produzem código de programas com muito menos instruções que o requerido pelas linguagens convencionais.

Os pacotes de software aplicativos podem servir como ferramentas de processamento de dados de quarta geração, se forem simples e puderem ser instalados pelos usuários finais. Um exemplo pode ser um software de etiquetas de endereçamento ou um banco de dados de clientes que pode ser instalado por um usuário final em um computador pessoal sem qualquer programação especial ou customização.

Ultimamente, a funcionalidade das ferramentas de desenvolvimento de quarta geração expandiu-se tanto, que em breve será difícil classificar essas ferramentas dessa maneira tradicional.

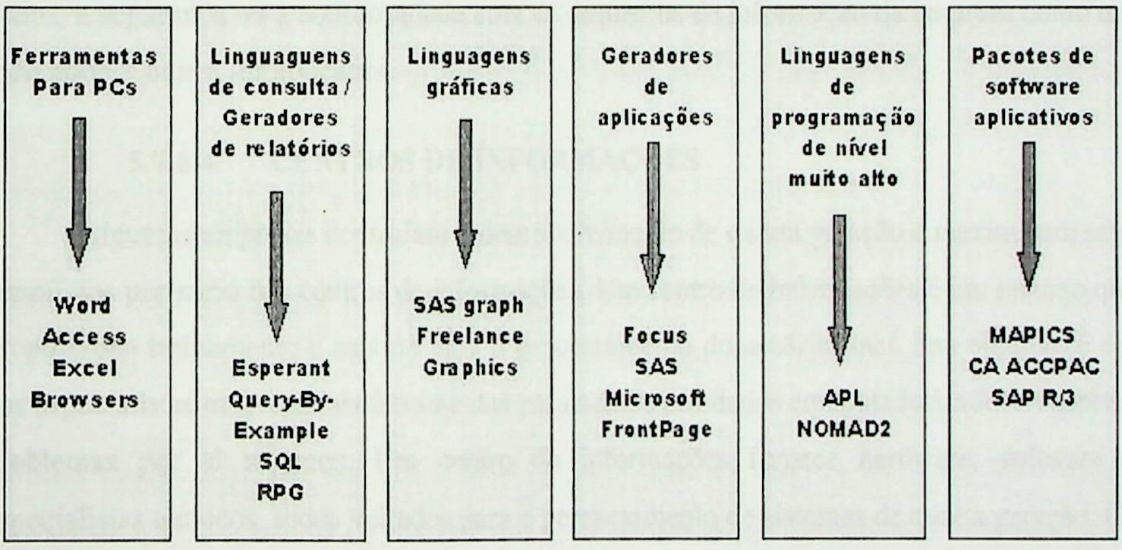


Figura 5.16 – *Ferramentas de projeto de soluções de quarta geração*

Muitos sistemas de gerenciamento de bancos de dados agora dispõem de ferramentas geradoras de aplicações para o usuário final, inclusive linguagens de consulta e geradores de relatórios. As linguagens de consulta e os geradores de relatórios muitas vezes são capazes de acessar outros bancos de dados além dos seus. Muitos pacotes aplicativos agora trazem geradores de relatórios controlados pelos usuários, linguagens gráficas e ferramentas da Internet.

5.7.6.3. DESVANTAGENS DO DESENVOLVIMENTO DE QUARTA GERAÇÃO

Embora o desenvolvimento de quarta geração ofereça muitas vantagens, ele só pode ser utilizado para resolver problemas quando a solução de sistema de informação é relativamente simples e facilmente entendida pelos usuários. As linguagens de quarta geração e outras ferramentas de processamento de dados tendem a funcionar melhor com arquivos pequenos e procedimentos de processamento simples. Como essas linguagens são não-procedurais, elas não podem manipular facilmente aplicações que requerem lógica procedural complexa. Sistemas de informação para suportar, por exemplo, a programação da produção ou o projeto de um reator nuclear ainda exigem a utilização de linguagens convencionais.

Outra desvantagem do desenvolvimento de quarta geração é a possível perda de controle, por parte da organização, sobre o processo de projeto de solução e seus recursos de informação. Os sistemas de quarta geração são desenvolvidos muito mais informalmente do que os que usam métodos tradicionais. Em conseqüência, os programadores profissionais ou analistas de sistemas não podem auxiliar na análise do problema, na avaliação de soluções alternativas e no projeto da solução. Além disso, os padrões para garantir a qualidade dos dados, a segurança ou a conformidade com os requisitos de informação da empresa como um todo podem nunca ser aplicados.

5.7.6.4. CENTROS DE INFORMAÇÕES

Algumas empresas controlam o desenvolvimento de quarta geração e maximizam seus benefícios por meio dos centros de informações. Um centro de informações é um recurso que proporciona treinamento e suporte para o processamento do usuário final. Seu objetivo é dar aos especialistas empresariais ferramentas para acesso aos dados computadorizados e resolver problemas por si mesmos. Um centro de informações fornece hardware, software e especialistas técnicos, todos voltados para o gerenciamento de sistemas de quarta geração. Os especialistas técnicos servem como instrutores e consultores; sua meta principal é treinar especialistas empresariais nas ferramentas de computação que vão usar, mas os especialistas também podem auxiliar na análise, no projeto e na programação de aplicações complexas.

Um centro de informações pode oferecer aos usuários acesso a *mainframes* e minicomputadores bem como a PCs, embora alguns desses centros disponham apenas de PCs. Os centros de informações suportam o processo de projeto de solução em muitos estágios. Seu pessoal é preparado para trabalhar intensamente para ajudar os usuários finais a conhecerem

seus problemas e a resolvê-los sozinhos tanto quanto possível. Os serviços oferecidos por um centro de informações podem ser resumidos desta forma:

- Sugerir aos especialistas empresariais aplicações de sistemas de informação existentes que possam ajudá-los a resolver seus problemas.
- Fornecer assistência técnica sugerindo hardware, software e metodologias apropriadas para resolver um determinado problema.
- Treinar especialistas empresariais nas ferramentas suportadas pelo centro de informações.
- Fornecer documentação e material de consulta para os recursos do centro de informações.
- Gerar protótipos para os especialistas empresariais avaliarem.
- Avaliar novos produtos de hardware e software.
- Proporcionar ao pessoal acesso a terminais, PCs, software associado e bancos de dados.

Outra vantagem dos centros de informações é que eles podem estabelecer padrões de hardware e software de modo que os usuários finais não introduzam muitos equipamentos ou dados incompatíveis na empresa. Normalmente, um centro de informações trabalha com o departamento de sistemas de informação da empresa para estabelecer padrões e diretrizes para aquisições de hardware e software; posteriormente, o centro de informações fornecerá assistência apenas para aquelas marcas de equipamentos e de software.

Por exemplo, um centro de informações pode proporcionar treinamento somente quanto ao Microsoft Word, e não para outros pacotes de edição de texto, porque o Word é o padrão da empresa. Caso contrário, seu pessoal teria de aprender diversos tipos de editores de texto, cada um dos quais utilizado por apenas um pequeno número de pessoas. Tal política também contribui para o uso eficiente das informações em uma organização. Os arquivos criados por um tipo de software de edição de texto nem sempre podem ser automaticamente usados por outro, o que restringe a transportabilidade de dados através da organização. Os centros de informações estão perdendo popularidade porque muitos usuários finais de hoje têm conhecimentos de processamento de dados, mas eles são úteis nas organizações interessadas na gerência do desenvolvimento pelos usuários finais.

5.7.7. Terceirização

Nos últimos anos, muitas organizações decidiram não mais manter os recursos internos necessários ao desenvolvimento e a operação de alguns ou de todos os seus sistemas

de informação. Elas preferiram a terceirização, a estratégia de transferir para empresas externas algumas ou todas as funções de sistemas de informação. As empresas terceirizadas fornecem diversos serviços, inclusive os seguintes:

- Preenchimento de vagas e gerenciamento do centro de processamento de dados da empresa.
- Operação de alguns ou de todos os sistemas de computador da empresa no centro de processamento de dados da empresa terceirizada.
- Desenvolvimento de aplicações para a organização.
- Operação da rede de telecomunicações da organização.

Algumas empresas muitas vezes optam pela terceirização quando não têm capacitação para desenvolver o sistema correto internamente. No entanto, a seleção da empresa a ser terceirizada exige conhecimento. A empresa externa precisa ter capacidade técnica e de aplicações. Quando se planeja terceirizar o desenvolvimento de um sistema de controle de processos de uma fábrica de produtos químicos, deve-se escolher uma empresa que não só tenha conhecimentos de desenvolvimento de aplicações mas que também conheça os controles de processos químicos. A figura 5.17 mostra como o processo de projeto de solução pode ser executado com a terceirização.

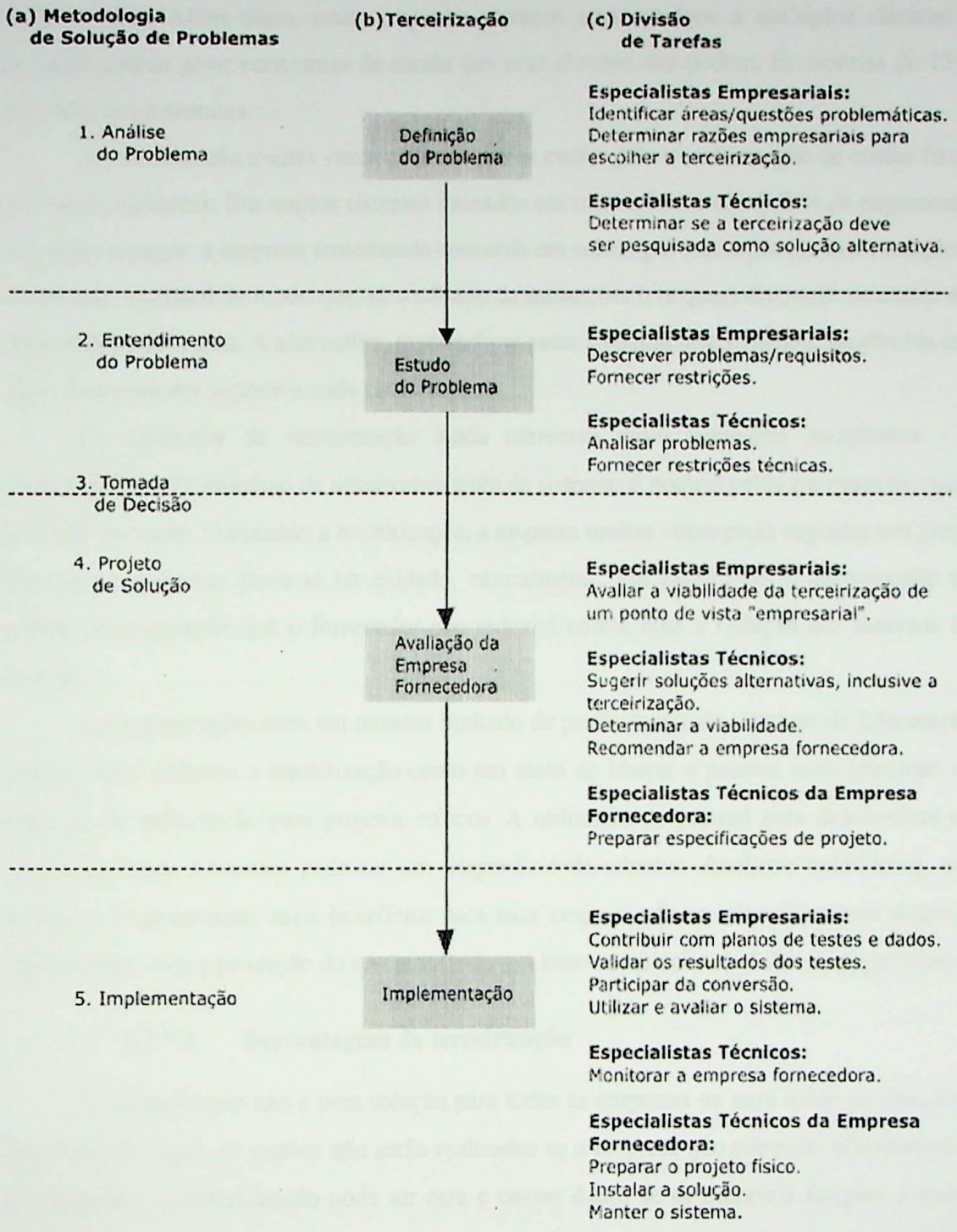


Figura 5.17 – Desenvolvendo um sistema através da terceirização

5.7.7.1. Vantagens da terceirização

As empresas optam pela terceirização por diversos motivos, dos quais o mais comum é o econômico. As empresas de terceirização são especialistas nos serviços que oferecem e por isso são capazes de fornecê-los a um preço mais em conta ou com melhor qualidade pelo

mesmo custo. Além disso, essas empresas vendem seus serviços a múltiplos clientes e portanto podem gerar economias de escala que seus clientes não podem. Economias de 15 a 30% não são incomuns.

A terceirização muitas vezes pode reduzir os custos pela transformação de custos fixos em custos variáveis. Em muitos sistemas baseados em transações, como folhas de pagamento ou contas a pagar, a empresa terceirizada concorda em cobrar por transação (nestes exemplos, um cheque emitido) de modo que, se o número de transações (cheques) diminuir, os custos do sistema também caem. A alternativa, mantendo-se esses sistemas internamente, resulta em um custo fixo para dar suporte a cada sistema.

Os contratos de terceirização ainda oferecem outra vantagem econômica - a previsibilidade. O processo de desenvolvimento de sistemas é notório pelos excessos de custo de 100% ou mais. Utilizando a terceirização, a empresa muitas vezes pode negociar um preço fixo para o sistema. Deve-se ter cuidado, naturalmente, em especificar o desempenho do sistema para garantir que o fornecedor não reduzirá custos com a redução dos recursos do sistema.

As organizações com um número limitado de profissionais de sistemas de informação muitas vezes utilizam a terceirização como um meio de liberar o pessoal mais talentoso de sistemas de informação para projetos críticos. A utilização de pessoal para desenvolver ou manter sistemas rotineiros pode ser um desperdício de talentos. Analistas habilidosos, por exemplo, proporcionam mais benefícios para suas empresas desenvolvendo novos sistemas relacionados com a produção do que mantendo um sistema rotineiro de folha de pagamento.

5.7.7.2. Desvantagens da terceirização

A terceirização não é uma solução para todas as empresas ou para todas as situações. Em primeiro lugar, os ganhos não serão realizados se a empresa não controlar eficazmente a terceirização. A terceirização pode ser cara e causar danos se as possíveis funções a serem terceirizadas não forem avaliadas cuidadosamente. É arriscado para uma empresa terceirizar aplicações das quais sua competitividade depende. Por exemplo, como a excelência de desempenho dos sistemas de reservas é crítica para o sucesso das redes de hotéis e empresas de transportes aéreos, esses sistemas provavelmente devem ser gerenciados internamente. Por outro lado, essas mesmas empresas provavelmente devem terceirizar seus sistemas de pedidos de reembolsos médicos porque perderão pouco se o processamento de pedidos de reembolsos for ocasionalmente interrompido. O mesmo não pode ser dito em relação aos grandes fornecedores de seguros de saúde, para os quais o sistema de reembolsos médicos é essencial.

Para estes, problemas ocasionais de processamento médico poderiam ser desastrosos, podendo até ameaçar sua sobrevivência. Assim, é recomendável que eles mantenham controle direto dos reembolsos médicos.

As empresas devem precaver-se contra a perda de controle que pode resultar da terceirização. As empresas por vezes descobrem que confiaram demais nos fornecedores; na verdade, o fornecedor pode ser a única alternativa para a terceirização de um serviço. Nessa situação inaceitável, o fornecedor fica livre para explorar o contrato pela elevação das taxas e redução do nível de serviço. O excesso de confiança também deixa a empresa vulnerável se o fornecedor vier a ter dificuldades financeiras ou fechar as portas.

5.8. UML – Unified Modeling Language

Cada vez mais as equipes responsáveis pelo desenvolvimento de sistemas são pressionadas a criarem softwares da mais alta qualidade e da forma mais rápida possível. Os ciclos de desenvolvimento estão-se tornando cada vez mais curtos. No passado, as companhias poderiam escolher entre sacrificar a qualidade de software para uma entrega mais rápida ou não implementar todas as funcionalidades do software a fim de entregá-lo no prazo estipulado. Na economia atual, nenhuma opção é possível. As companhias devem produzir um software da mais alta qualidade, em um prazo de entrega muito menor.

Habilitar os membros da equipe corporativa de desenvolvimento a se comunicar em um mesmo tom é essencial: diferentes colaboradores possuirão visões diferentes de um projeto ou de uma implementação de sistema e, a menos que todos falem um mesmo vocabulário comum e usem uma única linguagem de expressão, será impossível integrar perfeitamente todas as atividades desse time.

Este é o papel da UML, um padrão do Object Management Group. A UML é uma linguagem gráfica para visualização, especificação, construção e documentação dos artefatos de um sistema fundamentado em software. Com efeito, a UML é uma linguagem padrão para descrever imagens que será um software. A UML é resultado do esforço conjunto dos metodologistas Grady Booch, James Rumbaugh e Ivar Jacobson, e de um consórcio de empresas interessadas em modelagem orientada a objetos [33].

O UML é composto por nove tipos de diagramas, cada um com sua aplicação especifica:

- Diagrama de casos de uso
- Diagrama de classes
- Diagrama de objetos

- Diagrama de estado
- Diagrama de atividade
- Diagramas de interação
  - Diagrama de seqüência
  - Diagrama de colaboração
- Diagramas de implementação
  - Diagrama de componentes
  - Diagrama de execução

Cada um desses diagramas é descrito a seguir [30][34]. Por possuir sintaxe e semânticas bem definidas, a parte mais visível da sintaxe da UML é sua notação gráfica.

5.8.1. Diagrama de casos de uso

É um diagrama usado para se identificar como o sistema se comporta em várias situações que podem ocorrer durante sua operação. Ele descreve o sistema, seu ambiente e a relação entre os dois. Os componentes deste diagrama são os atores que representam tudo que é externo ao sistema, e os casos de uso que descrevem tudo que deve ser executado pelo sistema. O conjunto de casos de usos determina a completa funcionalidade do sistema, capturando assim o comportamento pretendido pelo sistema, do ponto de vista de quem interage com ele.

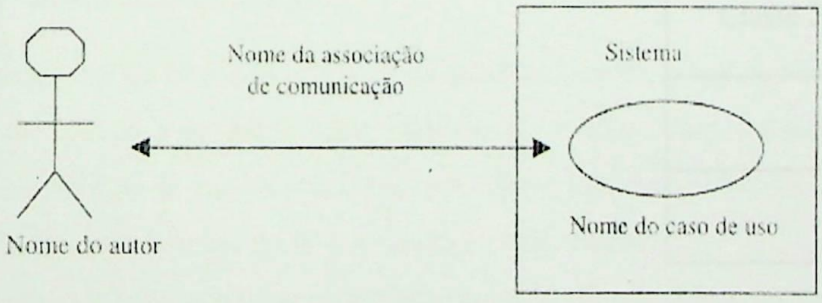


Figura 5.18 – Notação para o Diagrama de casos de uso

5.8.2. Diagrama de classes

O diagrama de classes representa a estrutura estática do sistema, exibindo as suas classes, suas estruturas internas, correspondentes aos atributos e serviços, seus

relacionamentos, correspondentes às conexões todo-parte e associações, suas estruturas de generalização e especialização.

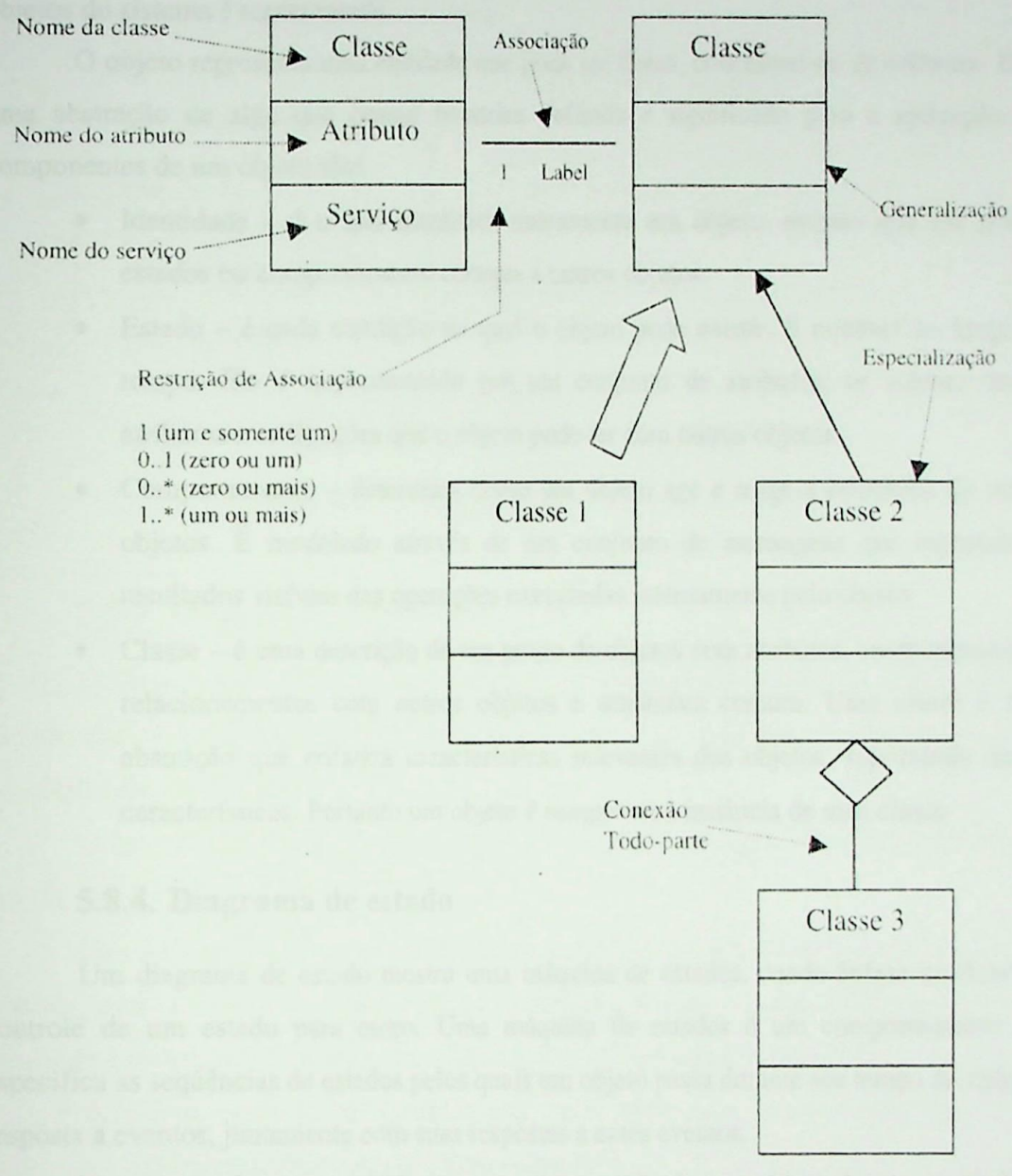


Figura 5.19 – Notação para o Diagrama de classes

5.8.3. Diagrama de objetos

O diagrama de objetos é uma instância dos diagramas de classes, incluindo objetos e valores de dados. Sua função é mostrar os objetos que foram instanciados das classes e exibir o perfil do sistema em um certo momento da execução. A mesma notação do diagrama de classes é utilizada, com duas exceções: os objetos são escritos com seus nomes sublinhados e todas as instâncias em um relacionamento são exibidas.

Os diagramas de objetos são úteis para exemplificar diagrama de classes complexas, principalmente para exibir exemplos de estruturas de dados. Esses diagramas também são utilizados como parte dos diagramas de colaboração, em que a colaboração dinâmica entre os objetos do sistema é representada.

O objeto representa uma entidade que pode ser física, conceitual ou de software. Ele é uma abstração de algo que possui fronteira definida e significado para a aplicação. Os componentes de um objeto são:

- Identidade – é o que identifica unicamente um objeto, mesmo que ele possua estados ou comportamento comuns a outros objetos.
- Estado – é cada condição na qual o objeto pode existir. É mutável ao longo do tempo. Ele é implementado por um conjunto de atributos, os valores desses atributos e as ligações que o objeto pode ter com outros objetos.
- Comportamento – determina como um objeto age e reage a estímulos de outros objetos. É modelado através de um conjunto de mensagens que representam resultados visíveis das operações executadas internamente pelo objeto.
- Classe – é uma descrição de um grupo de objetos com atributos, comportamentos, relacionamentos com outros objetos e semântica comum. Uma classe é uma abstração que enfatiza características relevantes dos objetos, suprimindo outras características. Portanto um objeto é sempre uma instância de uma classe.

5.8.4. Diagrama de estado

Um diagrama de estado mostra uma máquina de estados, dando ênfase ao fluxo de controle de um estado para outro. Uma máquina de estados é um comportamento que especifica as seqüências de estados pelos quais um objeto passa durante seu tempo de vida em resposta a eventos, juntamente com suas respostas a esses eventos.

Um estado é uma condição ou situação na vida de um objeto durante a qual ele satisfaz alguma condição, realiza uma atividade ou aguarda algum evento.

Um evento é uma especificação de uma ocorrência significativa que tem uma localização no tempo e no espaço. No contexto de uma máquina de estados, um evento é uma ocorrência de um estimulo capaz de ativar uma transição de estado.

Uma transição é um relacionamento entre dois estados, indicando que um objeto no primeiro estado realizará certas ações e entrará no segundo estado quando um evento especificado ocorrer e as condições especificadas estiverem satisfeitas.

Uma atividade é uma execução não-atômica em andamento em uma máquina de estados.

Uma ação é uma computação atômica executável que resulta em uma alteração do estado do modelo ou no retorno de um valor.

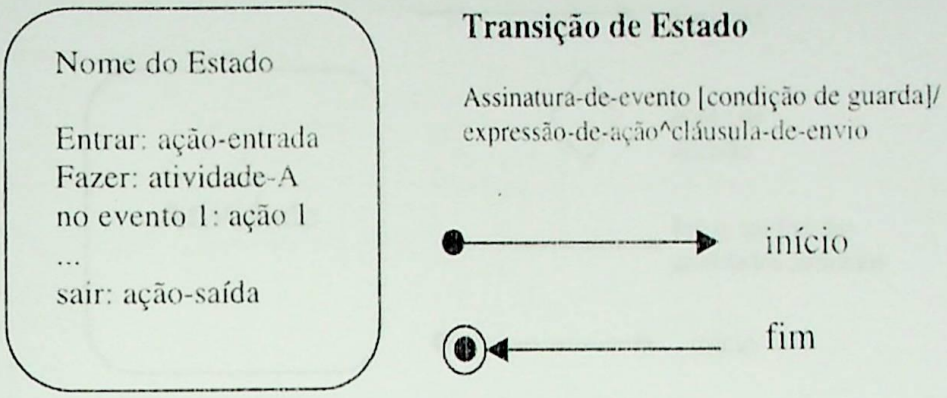


Figura 5.20 – Notação para o Diagrama de estado

5.8.5. Diagrama de atividade

O diagrama de atividade é uma variação do diagrama de estados, na qual os estados são atividades representando a execução de operações, e as transições são disparadas após a conclusão das operações.

O diagrama de atividades é utilizado para documentar a lógica de uma operação ou de um processo de negócio.

Eles não são importantes somente para a modelagem dos aspectos dinâmicos de um sistema, mas também para a construção de sistemas executáveis por meio de engenharia de produção e reversas. Eles podem ser utilizados com diferentes propósitos [34]:

a. Para capturar os trabalhos que serão executados quando uma operação é disparada. Este é seu uso mais comum.
b. Para capturar o trabalho interno de um objeto.
c. Para mostrar como um grupo de ações relacionadas podem ser executadas, e como elas vão afetar os objetos em torno delas.
d. Para mostrar como uma instância pode ser executada em termos de ações e objetos.
e. Para mostrar como um negócio funciona em termos de trabalhadores (atores), fluxos de trabalho, organização e objetos (fatores físicos e intelectuais usados no negócio).

O diagrama de atividades utiliza *swimlanes* para organizar responsabilidades para uma atividade pertencente a uma classe. *Swimlanes* geralmente correspondem a unidades organizacionais em um modelo de negócios, como por exemplo, seções ou departamentos.

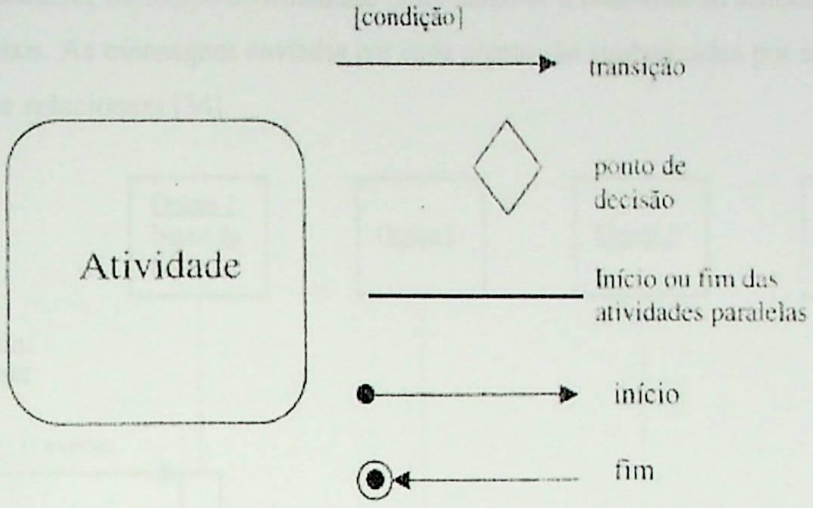


Figura 5.21 – Notação para o Diagrama de atividade

Este diagrama consiste em uma maneira alternativa de se mostrarem interações, com a possibilidade de expressar como as ações são executadas, o que elas fazem (mudanças dos estados dos objetos), quando elas são executadas (seqüência das ações), e onde elas acontecem (*swimlanes*).

5.8.6. Diagramas de interação

Os diagramas de interação exibem os padrões de interações entre objetos de um sistema. Uma interação consiste em uma especificação comportamental que inclui uma seqüência de objetos dentro de um contexto para realizar um propósito especifico, tal como a realização de um caso de uso [22].

A UML propõe dois tipos de diagramas de interação:

- Diagrama de seqüência.
- Diagrama de colaboração.

Estes diagramas representam as mesmas informações, porém cada um enfatiza um aspecto particular da informação.

- Diagrama de seqüência – exibe as interações entre os objetos organizados em seqüência cronológica.

Este diagrama é utilizado para documentar e verificar a lógica contida em casos de uso, mostrando a colaboração dinâmica entre os vários objetos envolvidos em um caso de uso,

as mensagens que eles enviam entre si, e qualquer valor de retorno associado à mensagem. Permite que se perceba a seqüência de mensagens enviadas entre os objetos.

O Diagrama de seqüência consiste em um número de objetos mostrados em linhas verticais. O decorrer do tempo é visualizado observando-se o diagrama no sentido vertical, de cima para baixo. As mensagens enviadas por cada objeto são simbolizadas por setas entre os objetos que se relacionam [34].

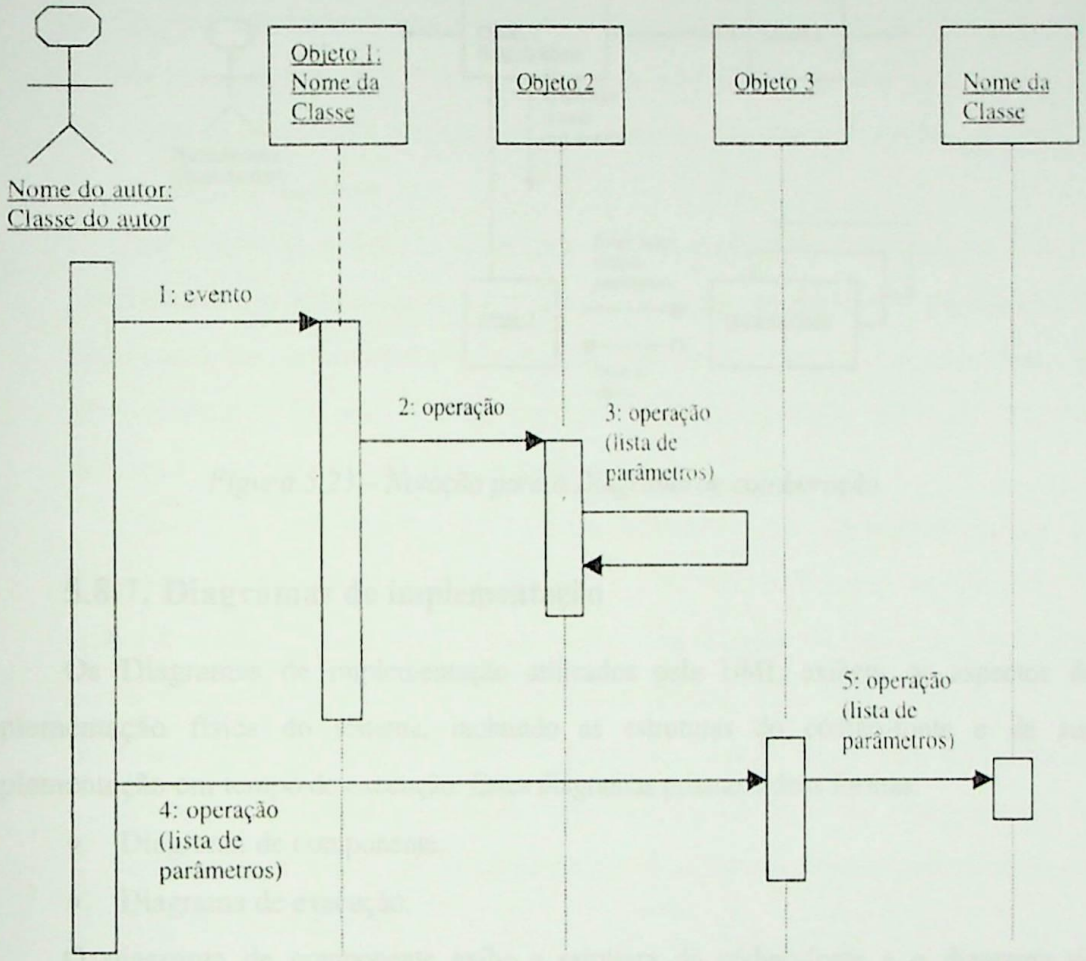


Figura 5.22 – Notação para o Diagrama de seqüência.

- Diagrama de colaboração – mostra, de maneira semelhante ao diagrama de seqüência, a colaboração dinâmica entre os objetos. Normalmente pode-se escolher entre utilizar o diagrama de colaboração ou o diagrama de seqüência.

Diferentemente do diagrama de seqüência, o diagrama de colaboração exibe as interações entre os objetos, mostrando a troca de mensagens entre eles, bem como seus relacionamentos. Por outro lado, o diagrama de colaboração não exibe o tempo como uma dimensão separada; a seqüência das mensagens e processos *(threads)* concorrentes deve ser determinada utilizando-se números em seqüência.

Portanto, se a ênfase da modelagem for o decorrer do tempo, é aconselhável utilizar o Diagrama de Seqüência; porém, se a ênfase for o contexto do Sistema, é melhor dar prioridade ao Diagrama de Colaboração.

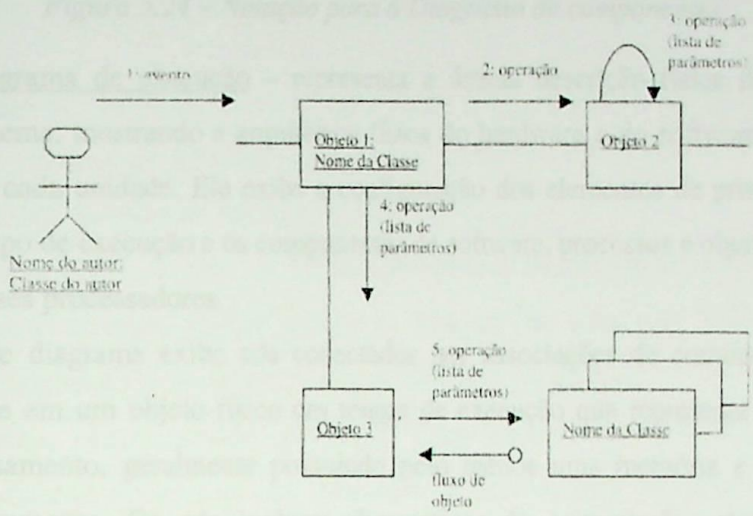


Figura 5.23 – Notação para o Diagrama de colaboração

5.8.7. Diagramas de implementação

Os Diagramas de implementação utilizados pela UML exibem os aspectos de implementação física do sistema, incluindo as estruturas do código-fonte e de sua implementação em tempo de execução. Estes diagramas possuem duas formas:

- Diagrama de componente.
- Diagrama de execução.

O diagrama de componente exibe a estrutura do código-fonte e o diagrama de execução mostra a estrutura do Sistema em tempo de execução.

- Diagrama de componentes – exibe os componentes de software do sistema e suas interdependências, representando a estrutura do código gerado. Os componentes são a implementação, na arquitetura física, dos conceitos e da funcionalidade definidos na arquitetura lógica, correspondentes às classes e objetos e seus relacionamentos.

Cada componente do diagrama pode ser documentado por um outro diagrama de componente mais detalhado, por um diagrama de casos de uso ou por um diagrama de classes.

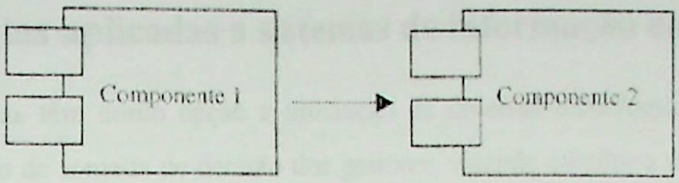


Figura 5.24 – Notação para o Diagrama de componentes

- Diagrama de execução – representa a última descrição física da topologia do Sistema, mostrando a arquitetura física do hardware e do software que executam em cada unidade. Ele exibe a configuração dos elementos de processamento em tempo de execução e os componentes de software, processos e objetos que residem nesses processadores.

Este diagrama exibe nós conectados por associações de comunicação. Um nó consiste em um objeto físico em tempo de execução que representa um recurso de processamento, geralmente possuindo pelo menos uma memória e capacidade de processamento. Os nós incluem dispositivos de computação, além de recursos humanos ou recursos mecânicos de processamento.

Observa-se claramente que, independentemente do paradigma usado no desenvolvimento, todos propõem modelar o sistema em diferentes visões antes de elaborar códigos, possibilitando uma melhor visualização do que será obtido bem como alterações com baixo custo. Desenvolver software desta forma requer:

a) maturidade das pessoas e das organizações. Sendo assim será discutido no próximo capítulo um modelo de capacidade e maturidade que visa o alcance de uma melhor disciplina no desenvolvimento de software e conseqüente melhoria da qualidade do software.

b) manutenção das visões obtidas e o gerenciamento da configuração de software.

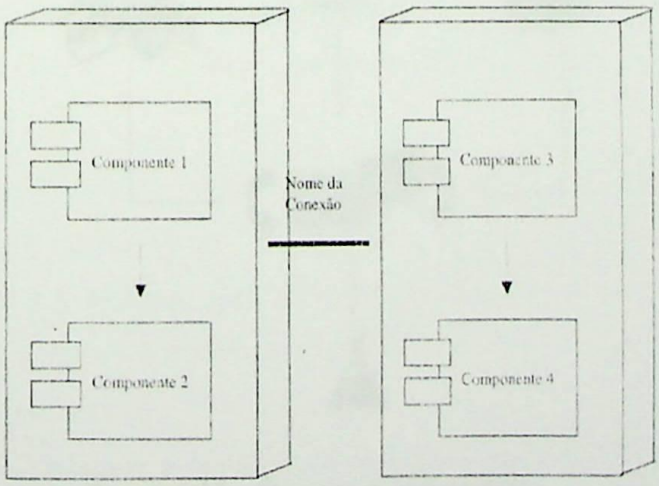


Figura 5.25 – Notação para o Diagrama de execução

5.9. Tecnologias aplicadas a sistemas de informação empresariais

As empresas têm como opção a utilização de diversas tecnologias modernas, para facilitar o processo de tomada de decisão dos gestores, visando atender a sua complexidade, seu crescimento, sua modernidade, sua perenidade, sua rentabilidade e sua competitividade [18].

Para Dalfovo [19], de acordo com Alter [20], sistemas de Informação são a combinação de práticas de trabalho, informações, pessoas e informações tecnológicas . Os componentes de um Sistema de Informação são:

a) informações: sistemas de informação podem incluir dados formatados, textos. imagens e sons. Dados são fatos, imagens ou sons que podem ou não ser pertinentes ou importantes para uma tarefa cm particular;
b) pessoas: exceto quando uma tarefa é totalmente automatizada, os Sistemas de Informação também podem necessitar de pessoas para dar entrada, processar ou usar o dados;
c) informações tecnológicas: inclui hardware e software para executar uma ou mais tarefas de processamento de dados tais como, captura, transmissão. armazenamento, recuperação, manipulação ou apresentação dos dados;
d) práticas de trabalho: são os métodos usados por pessoas e tecnologia para executar os trabalhos;
e) objetivos: são as metas a serem alcançadas, definidas pela empresa.

A relação existente entre os componentes de um sistema de informação é apresentada na Figura 5.26.

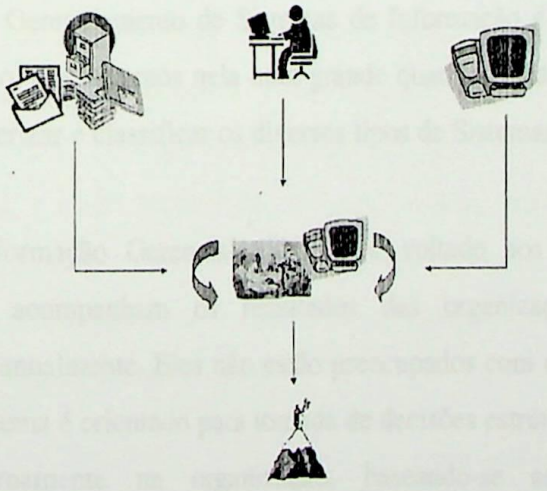


Figura 5.26 – Sistemas de informação

Gandara [21] frisa a importância de manter a vida dos Sistemas de Informação, e de recuperar e manter a integridade dos dados, reunindo os diversos tipos de bases de dados. Conforme observado na Figura 5.27, há três níveis de influência de um sistema de informação dentro de uma organização, sendo eles:

a) nível estratégico: interação entre as informações do ambiente empresarial (estão fora de empresa) e as informações internas da empresa;
b) nível tático: aglutinação de informações de uma área de resultado e não da empresa como um todo;
c) nível operacional: principalmente através de documentos escritos de várias informações estabelecidas.

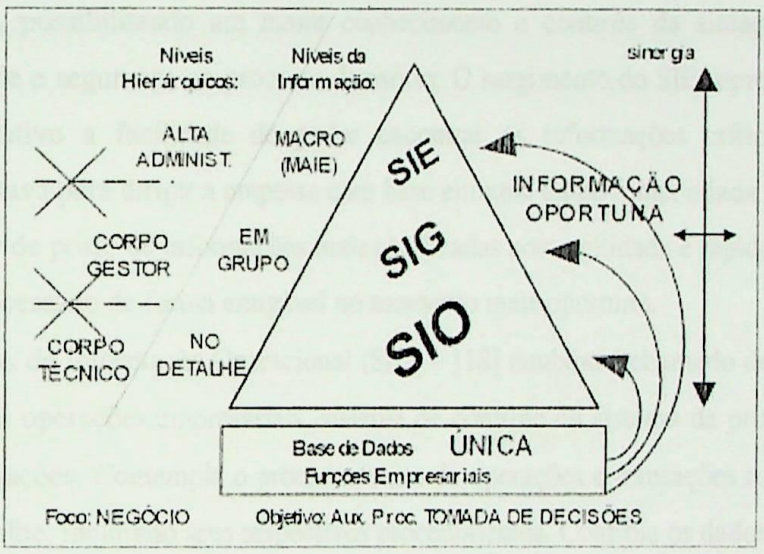


Figura 5.27 – Nível de influência do sistema de informação

A área de Gerenciamento de Sistemas de Informação é bastante abrangente [4][22][23]. Por isso, encontramos nela uma grande quantidade de termos, usados em tentativas de caracterizar e classificar os diversos tipos de Sistemas de Informação (SI), tais como:

- Sistema de Informação Gerencial (SIG) – é voltado aos administradores de empresas que acompanham os resultados das organizações semanalmente; mensalmente e anualmente. Eles não estão preocupados com os resultados diários. Esse tipo de sistema é orientado para tomada de decisões estruturadas. Os dados são coletados internamente na organização, baseando-se somente nos dados corporativos existentes e no fluxo de dados. A característica do SIG é utilizar

somente dados estruturados, que também são úteis para o planejamento de metas estratégicas.

- Sistema de Apoio à Decisão (SAD) – é um sistema mais complexo, que permite total acesso à base de dados corporativa, modelagem de problemas, simulações, e possui uma interface amigável. Além disso, auxilia o executivo em todas as fases de tomada de decisão, principalmente nas etapas de desenvolvimento, comparação e classificação dos riscos, além de fornecer subsídios para a escolha de uma boa alternativa.
- Sistema de Informação Executiva (SIE) – é uma tecnologia que integra num único sistema todas as informações necessárias para que o executivo possa verificá-las de forma rápida e amigável, desde o nível consolidado até o nível mais analítico que se desejar, possibilitando um maior conhecimento e controle da situação e maior agilidade e segurança no processo decisório. O surgimento do SIE representou para o executivo a facilidade de poder encontrar as informações críticas, de que necessitava para dirigir a empresa com base em uma única fonte, aliada à segurança de estar de posse de informações mais atualizadas com agilidade e rapidez, tudo isto sendo acessado de forma amigável no momento mais oportuno.
- Sistemas de Informação Operacional (SIO) – [18] também é chamado de sistema de apoio às operações empresariais, sistema de controle ou sistema de processamento de transações. Contempla o processamento de operações e transações rotineiras, no seu detalhe, incluindo seus respectivos procedimentos. Controla os dados detalhados das operações das funções empresariais imprescindíveis ao funcionamento harmônico da empresa, auxiliando a tomada decisão do corpo técnico das unidades de departamentais. As decisões operacionais estão ligadas ao controle e às atividades operacionais da empresa. Visam alcançar os padrões de funcionamento preestabelecidos, com controle do detalhe do planejamento operacional. No nível operacional está o corpo técnico da empresa, ou seja, engenheiros, assistentes, auxiliares, nas suas respectivas subunidades departamentais ou setores. Neste caso o nível da informação é detalhado (analítico), contemplando pormenores específicos de um dado, de uma tarefa ou atividade.

6. O Modelo atual e o modelo proposto

Baseado na analise do modelo atual para registro das informações sobre os equipamentos eletro-eletrônicos existentes na planta elétrica industrial da CST e em um conjunto de pré-condições para desenvolvimento determinadas pela equipe de projeto da CST, foi proposto um novo modelo eletrônico para arquivamento e gerenciamento das informações da planta em questão.

Desta forma, neste capitulo serão apresentadas as descrições do sistema atual e do modelo proposto para o novo sistema. Assim sendo, a especificação de pré-condições para desenvolvimento por parte da CST, e questões de ordem técnica tais como a escolha das ferramentas e as metodologias utilizadas para desenvolver o novo sistema, serão discutidas no capitulo seguinte, em que oportunamente as justificativas serão apresentadas.

6.1. O modelo atual

O modelo utilizado para armazenamento de informações sobre os equipamentos eletro-eletrônicos da planta industrial da CST é comprovadamente quase que ineficiente. Isto se deve basicamente ao fato de que grandes quantidades de informações estão armazenadas em mídia impressa e que parte das informações armazenadas em meio eletrônico são na maioria das vezes incompatíveis com os softwares utilizados e devidamente licenciados para a CST.

Com relação ao armazenamento em mídia eletrônica, o que ocorre é que a área de estudos da CST é praticamente toda terceirizada. Desta forma, ao se exigir que os arquivos eletrônicos sejam entregues, ocorre um aumento de custo nos serviços contratados. Este aumento de custo se deve ao fato de a empresa contratada cobrar pelo uso da ferramenta que permite a manipulação dos arquivos eletrônicos. Em outras palavras, as empresas contratadas possuem ferramentas de autoria própria ou licenciadas, e para disponibilizá-las para a CST ou para qualquer outro cliente é necessário repassar o custo de licenciamento.

Outras desvantagens foram detectadas no sistema atual da CST com relação a um equipamento especifico ou grupo de equipamentos, tais como:

- A maioria das informações se encontra registrada em mídia impressa – fornecidas pelos fabricantes dos equipamentos ou pelas empresas prestadoras de serviços que realizam os estudos da planta elétrica da CST.

- Dificuldade em recuperar qualquer tipo de informação pertinente aos equipamentos, isto é, a busca é feita manualmente nos relatórios impressos fornecidos pelas empresas prestadoras de serviços na área de estudos ou pelos fabricantes dos equipamentos.
- Não existe modo efetivo de inclusão de novos dados – e, quando feito, isso ocorre na forma de apontamentos nos espaços em branco das páginas dos relatórios fornecidos em mídia impressa feito pelas empresas prestadoras de serviços.
- A atualização das informações é feita de forma manuscrita. Esta ocorre da mesma forma quando do processo de inclusão.
- Diversas formas de apresentar a mesma informação ou conjunto de informações, ou seja, cada empresa que realiza um estudo da planta industrial da CST apresenta seus relatórios técnicos de uma forma.
- Informações não confiáveis – Quando as informações são encontradas, não existe uma garantia de que elas estejam atualizadas.
- Não existe o cruzamento de informações entre as ordens de graduação dos equipamentos e as folhas de coordenação de proteção.

É sabido por todos que os sistemas de informações que têm seus registros mantidos na forma manuscrita tendem a ser lentos nas respostas e na maioria das vezes imprecisos com relação às informações nèles contidas. A CST, ciente deste fato, adotou um sistema de atualização em que um pequeno grupo de funcionários era responsável pela manutenção das informações. Tal procedimento mostrou-se ineficiente ao longo do tempo, uma vez que sua planta industrial está dividida em várias áreas, e estas áreas interagem de forma quase que permanente com tal sistema. Em outras palavras, o volume de informações geradas e modificadas era maior do que a equipe responsável era capaz de manter. O aumento do número de responsáveis pela manutenção de tais informações estava implicando aumento de custos para esta forma de armazenamento e recuperação de informações.

6.2. O modelo proposto

Diante do exposto no item anterior, a solução encontrada foi criar uma base de dados única, de forma que toda a informação pertinente à planta elétrica industrial da CST ficasse centralizada e que todos os participantes do sistema tivessem acesso à mesma.

A segurança foi outro aspecto que também foi abordado, de modo a garantir que somente pessoas autorizadas tivessem acesso às informações contidas nessa base de dados.

Assim sendo foram criados vários níveis de acesso de acordo com o perfil de cada grupo de usuários do sistema. Para tanto, somente três grupos foram identificados:

- Os administradores: composto pelos engenheiros – chefe de equipe de manutenção. Estes com direito de acesso completo ao sistema, isto é, com permissão para incluir, excluir, alterar e emitir qualquer tipo de relatório.
- Administradores terceirizados: é composto pelas equipes técnicas de empresas contratadas pela CST para efetuar estudos de ampliação e reformas de sua planta elétrica industrial. Este grupo de usuários possui os mesmos direitos do grupo dos administradores. A diferença é que tal acesso é restrito a um número limitado de informações do banco de dados. Essa limitação é flexível, e está relacionada às áreas envolvidas no estudo.
- Operadores: composto pelos técnicos das equipes de manutenção. Para esse grupo de usuários só é permitido extrair relatórios contendo informações de ajustes e calibração de equipamentos. A eles nenhum tipo de exclusão, inclusão e alteração é permitido.

O novo sistema também permitirá tanto a exportação quanto a importação de dados. Este recurso tem por objetivo garantir o uso de um formato padrão para representar as informações sobre os equipamentos

O link entre as informações geradas por outros aplicativos, por exemplo, os arquivos gerados pelo programa de CAD (Computer Aided Design), é outro item contemplado pelo novo sistema. No sistema atual este era outro problema. Encontrar informações sobre determinado equipamento que constem no arquivo de CAD exige uma pesquisa nos relatórios gerados pelas equipes de estudo e análise. Já no novo sistema, tais arquivos de CAD são armazenados juntamente com as informações do equipamento. Note que aqui está sendo adotado um procedimento mais eficiente do que o do sistema atual, de modo a permitir recuperação da informação nos dois sentidos.

Nesse novo sistema os problemas de alteração, exclusão e inclusão de novas informações são praticamente extintos. Em outras palavras, qualquer manipulação da base de dados é disponibilizada para todos os usuários do sistema em tempo real.

Outro aspecto importante que o novo sistema possui é a eliminação de informações dúbias, isto é, toda e qualquer alteração dos dados de qualquer que seja o equipamento é feita diretamente na base de dados, de modo a não permitir que informações desatualizadas circulem entre os usuários participantes do sistema.

Outro problema que o sistema proposto resolve em definitivo é o da padronização de relatórios e termos. Cada uma das empresas prestadoras de serviços para a CST possui seus próprios padrões de relatórios e termos. Com o modelo proposto, qualquer que seja a empresa a realizar um estudo da planta elétrica industrial da CST, irá ao final entregar os relatórios dentro dos padrões determinados pela CST e, por conseguinte utilizará os termos também pré-definidos pela CST.

7. Metodologia de desenvolvimento

No capitulo anterior foram apresentados os modelos atual e proposto. Neste capítulo será apresentada a metodologia de desenvolvimento utilizada para implementação do software solicitado pela CST. Este software recebeu a denominação de SIPE (Sistema de Informações sobre a Planta Elétrica da CST).

Porém, para entrarmos no mérito da questão, faz-se mistér que se destaquem alguns pontos relevantes que influenciaram tanto na escolha do modelo de desenvolvimento quanto na seleção de técnicas e ferramentas utilizadas.

O desenvolvimento de software se baseou fundamentalmente em 4 condições pré-determinadas pela CST:

- que o software deveria estar operacionalmente pronto em 12 (doze) semanas, conforme cronograma apresentado na *figura 7.1*;
- que o custo de desenvolvimento obedecesse rigidamente à dotação de verba destinada para tal fim;
- que o software fosse desenvolvido baseado na ferramenta de banco de dados Microsoft Access 2000;
- que o software utilizasse exclusivamente a linguagem de programação suportada pela ferramenta de banco de dados, ou seja, o MSSQL (Microsoft Strutured Query Language).

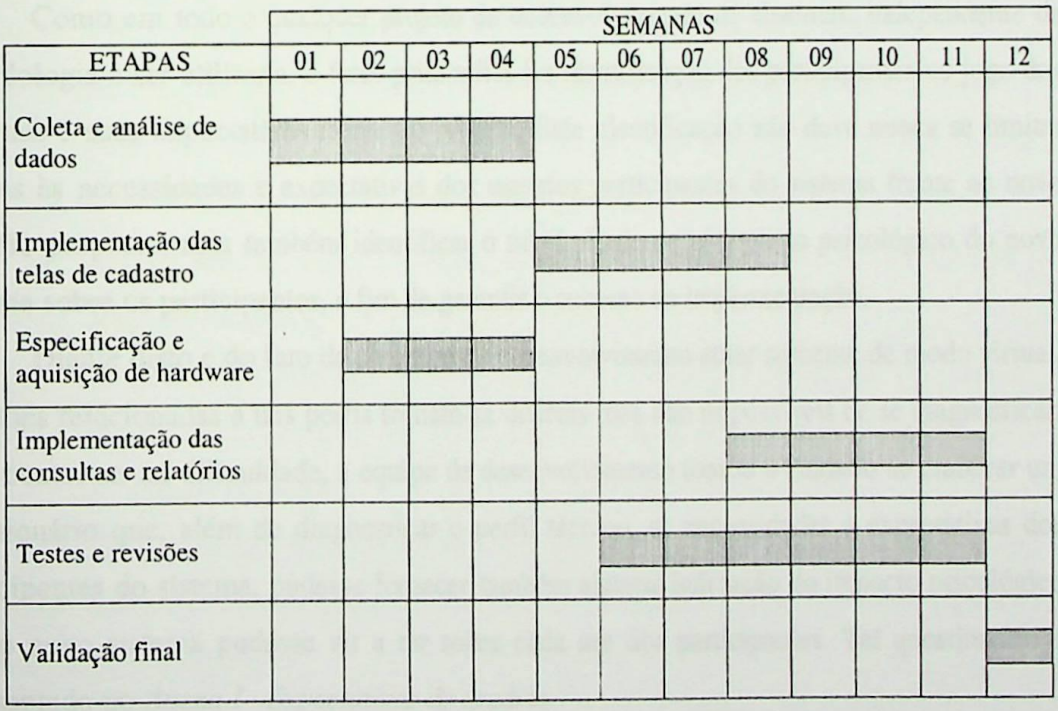


Figura 7.1 – Cronograma de desenvolvimento do SIPE

Diante do exposto acima, a metodologia de desenvolvimento escolhida para a implementação do software solicitado pela CST foi a da prototipação. Tal escolha levou em consideração os aspectos dos recursos temporais da prototipação, isto é, a necessidade premente de algo "palpável" e funcional, mesmo que de forma precária em um curto espaço de tempo.

Outro aspecto que influenciou substancialmente a escolha da técnica de prototipação foi a escolha, por parte da CST, do Microsoft Access como ferramenta de banco de dados, já que tal ferramenta, por meio de seus "assistentes", permite de modo fácil e simples a criação de diagramas de entidade e relacionamento, de consultas, construção de formulários e macros.

Por último, o fato de a dotação de verbas ser bastante limitada e a distância geográfica relativamente grande entre a equipe de desenvolvimento e a equipe da CST exigia uma estratégia de desenvolvimento um tanto inusitada, e por isso tornou-se tema deste trabalho. A suposta dificuldade estava em desenvolver um software cujos mecanismos de funcionamento em si, a equipe de desenvolvimento desconhecia, e as possibilidades de interação real entre os participantes do projeto de desenvolvimento eram praticamente inexistentes. Em outras palavras, o desafio aqui era desenvolver um sistema de software através do uso da rede mundial de computadores como único meio de interação entre as participantes do sistema e a equipe de desenvolvimento em tempo hábil.

7.1. Usuários e coleta de dados

Como em todo e qualquer projeto de desenvolvimento de sistemas, independente da metodologia a ser utilizada, o fator primordial é a identificação dos participantes no jogo dos sistemas e suas expectativas frente ao projeto. Esta identificação não deve nunca se limitar apenas às necessidades e expectativas dos usuários participantes do sistema frente ao novo modelo proposto, mas também identificar o nível técnico e o impacto psicológico do novo sistema sobre os participantes, a fim de garantir o sucesso da implementação.

Diante disto e do fato de a equipe de desenvolvimento atuar somente de modo virtual, questões relacionadas a tais perfis tornam-se difíceis mas não impossíveis de se diagnosticar. Para contornar tal dificuldade, a equipe de desenvolvimento tomou o cuidado de elaborar um questionário que, além de diagnosticar o perfil técnico, as necessidades e expectativas dos participantes do sistema, pudesse fornecer também alguma indicação do impacto psicológico que o novo sistema pudesse vir a ter sobre cada um dos participantes. Tal questionário é apresentado no *Anexo I –Expectativas do usuário.*

Analisando as respostas obtidas através do questionário citado acima, a equipe pode extrair as seguintes informações com relação aos usuários:

- Otimismo frente à possibilidade de se eliminarem problemas de informações redundantes e/ou desatualizadas.
- Grande expectativa para com a agilização da troca de informações entre as equipes de manutenção.
- Grande preocupação com a segurança das informações.
- Redução de custos dos serviços a serem executados na planta elétrica da CST, quer seja através da contratação de serviços terceirizados, quer seja pela própria equipe técnica da CST.
- Ainda existe o mito de que o computador pode substituir as pessoas.

Por outro lado, sobre as informações processadas pelo sistema em questão, foram os seguintes os indicadores extraídos do questionário *expectativas do usuário:*

- O número de entidades que interagem entre si é relativamente pequeno, mas o volume de registros de cada uma delas é consideravelmente grande. Em função disto a escolha do Microsoft Access foi satisfatória, apesar de suas limitações.
- Apesar de a CST possuir um padrão de relatório próprio, os usuários utilizam variantes desse padrão por julgá-lo incompleto.
- Um mesmo item de dados recebe diferentes formas de tratamento, isto é, uma mesma informação é referenciada através de nomes diversos.
- Utilização de unidades de medidas diferentes para um mesmo item de dados.
- Havia informações que eram exclusivamente utilizadas por apenas um usuário.
- Parte do universo de informações do sistema em questão já estava sendo armazenada em arquivos digitais, mas não existia nenhuma ligação entre eles.
- Alguns itens de dados se mostraram candidatos naturais para recuperação de um conjunto de dados.
- A grande maioria das informações sobre o sistema elétrico da CST ainda se encontra armazenada em mídia impressa, cuja denominação é *Relatórios de conclusão de estudos*.

Em função destas observações extraídas do questionário *expectativas do usuário,* a equipe de desenvolvimento criou um segundo questionário (*Anexo II – Especificação de campos*), cujo objetivo era:

- Eliminar por completo as redundâncias dos nomes dos itens de dados.
- Padronizar as unidades de medida utilizadas.

- Definir um novo padrão de registro de informações de modo que atendesse a todos os participantes do sistema.
- Verificar a importância do registro de informações particulares sobre o sistema frente aos demais participantes.
- Induzir os participantes do sistema a efetuarem uma classificação preliminar das informações a fim de diagnosticar a relevância da informação para o sistema como um todo.
- Sugerir algumas formas de os participantes do sistema poderem recuperar um determinado conjunto de informações sobre o ele (campo marcado de vermelho nas tabelas de equipamentos do questionário de *especificação de campos).*
- Detectar outras formas possíveis de como os participantes recuperam um determinado conjunto de informações sobre o sistema.

Ao se fazer o cruzamento das informações obtidas com as respostas dos dois questionários, conforme mostrado nos anexos I e II, foi possível concluir que:

- A planta elétrica da CST é composta basicamente por sete diferentes grupos de equipamentos eletro-eletrônicos.
- Alguns desses grupos poderiam ainda apresentar outras subdivisões.
- Apesar de os participantes utilizarem muitas formas para recuperação de informação sobre um dado equipamento ou conjunto de equipamentos pertencentes ao sistema elétrico, essas poderiam ser restringidas a um conjunto de oito campos, que passamos a denominar de *chaves de busca de informações* ou simplesmente *chaves.*
- Destas oito chaves, um máximo de três indicava ser mais que suficiente para cada tipo de equipamento ou conjunto de equipamentos, quando da recuperação de informações sobre o sistema elétrico.
- Informações complementares sobre o sistema elétrico da CST também estavam armazenadas em arquivos digitais gerados através do uso de um programa de CAD.
- Era importante que essas informações armazenadas ou geradas pelo programa de CAD fossem armazenadas conjuntamente com as informações de determinados tipos de equipamentos da planta elétrica da CST.

Como se pode observar, esta etapa teve também como objetivo obter informações suficientes a fim de se propor uma solução ótima para o problema, e ainda levantar o maior número possível de restrições de ordem técnica, operacional e temporal, entre outras.

Em função das observações apresentadas anteriormente, pode-se perceber a existência de uma quantidade considerável de entidades (objetos). Assim sendo, foi elaborado um DER utilizando-se os recursos do Microsoft Access. Poderíamos ainda listar uma série de outros fatores que nos levaram à implementação do DER (preferência do cliente, dados aparentemente triviais, etc.), mas nos limitaremos a dizer que a preocupação maior estava com o aumento do grau de complexidade do sistema em si, uma vez que o desafio aqui era a questão da virtualidade do desenvolvimento desse sistema.

A figura a seguir mostra como ficou esse Diagrama de Entidade-relacionamento:

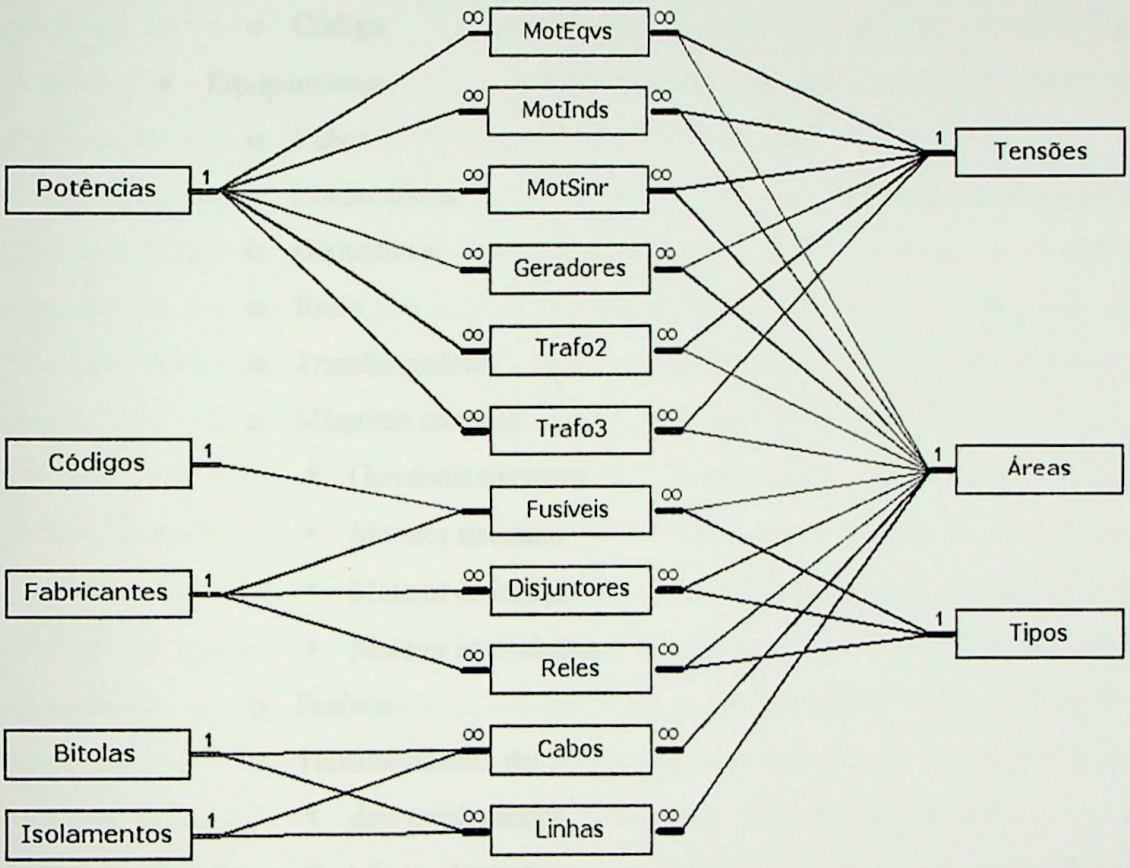


Figura 7.2 – Diagrama de entidade e relacionamento do SIPE

7.2. A implementação do sistema via prototipação

Uma vez modelado o sistema, o próximo passo foi a estruturação do software em si. Para uma primeira versão do versão do SIPE (Sistema de Informações sobre a Planta Elétrica da CST) foi então proposto um modelo com duas opções básicas: cadastramento e consultas/relatórios.

A opção cadastramento foi subdividida em duas categorias:

- Chaves
  - Área
  - Bitola
  - Isolamento
  - Fabricante
  - Potência
  - Tensão
  - Tipo
  - Código
- Equipamentos
  - Cabos
  - Linhas aéreas
  - Disjuntores
  - Relés
  - Transformadores
  - Máquinas rotativas
    - Geradores síncronos
    - Motores síncronos
    - Motores de indução
    - Motores equivalentes
  - Fusíveis
  - Transformadores de:
    - dois enrolamentos
    - três enrolamentos

Ainda com relação à opção de cadastramento, ela foi construída de forma a permitir que os usuários, além de cadastrarem as informações referentes a um dado equipamento, possam também consultá-las de forma seqüencial e até mesmo editá-las.

Já a opção consulta/relatórios, como o próprio nome indica, tem por objetivo único gerar relatórios que poderão ser convertidos em mídia impressa e/ou arquivos digitais. Tais consultas filtram as informações sobre os equipamentos baseadas em uma tríade de chaves que foram indicadas pelos participantes do sistema quando da coleta de dados. Os formatos dos relatórios foram elaborados baseados nos diversos modelos fornecidos pelos participantes do sistema. Estes formatos se encontram no *Anexo III – Padrões de relatórios.*

Até este ponto do desenvolvimento a ferramenta escolhida pela CST (Microsoft Access) para implementação do SIPE proposto se mostrou adequada para todas as solicitações feitas pelos usuários. Lembramos aqui que uma das condições determinadas pela CST, era de que o software deveria limitar-se ao uso exclusivo dos recursos suportados por tal banco de dados.

Em todo projeto de software, à medida que as etapas definidas no cronograma de desenvolvimento são concluídas, é normal que os participantes do sistema venham vislumbrar novas possibilidades de respostas. Em projetos que utilizam a técnica de prototipação essa possibilidade de geração de novas expectativas é mais intensa devido ao fato de que as primeiras versões do protótipo, após serem avaliadas, em muitos casos, são entregues aos usuários finais. O inconveniente disso, é que os *layouts* de telas e relatórios são relegados ao segundo plano, podendo vir a gerar um grande número de revisões.

Neste aspecto, o exercício da função de gerente de projeto é de suma importância para que os objetivos do projeto de software não se desvirtuem. Cabe a ele julgar se as novas expectativas dos usuários justificam ou não que os prazos definidos no cronograma de atividades sejam excedidos ou reduzidos. No caso do projeto SIPE, a interação do gerente de projeto com o analista e os demais participantes do sistema se mostrou eficiente e eficaz. Eficiente porque ele usou seu poder de decisão de forma precisa e convincente, e eficaz porque, ouvindo as opiniões dos usuários e discutindo-as com o analista de projeto, ele conseguiu superar as expectativas da proposta inicial do SIPE. Essa superação de expectativas consistiu na substituição de algumas rotinas que se mostraram desnecessárias ou de pouca importância sob a ótica dos usuários, por outras que de fato permitiram otimizar o trabalho deles, como por exemplo, redução do volume de dados digitados durante uma operação de cadastro, duplicação de dados, importação e exportação de dados de forma automática.

Um problema observado durante o desenvolvimento do SIPE, foi o de que algumas das expectativas geradas pelos seus usuários não eram suportadas pela ferramenta de bancos escolhida pela CST, como por exemplo, a importação e exportação parcial do banco de dados. Neste caso em especial, a solução encontrada foi combinar recursos da ferramenta escolhida com a utilização de recursos oferecidos por um pequeno aplicativo desenvolvido especificamente para contornar a limitação da ferramenta de banco de dados.

Um outro problema que foi detectado durante a fase de testes e validação foi a perda da formatação dos relatórios. O fato é que a CST possui um padrão que define os tamanhos de margens para qualquer tipo de relatório. Constatou-se que quando o software do SIPE era instalado em outro equipamento, ou ainda, a impressão era redirecionada para outra

impressora, esse formato era perdido. Esse problema não teve como ser solucionado. Após pesquisarmos em literatura especifica, grupos de discussão, fóruns e por ultimo efetuar consulta junto ao fabricante da ferramenta de banco de dados em questão, chegamos a conclusão de que o problema reside na ferramenta de banco de dados e não no software objeto deste trabalho.

8. Resultados

Neste capitulo apresentaremos os resultados obtidos ao final do desenvolvimento do modelo proposto, bem como teceremos considerações sobre os aspectos da metodologia por nós utilizada para coleta de dados e entrevista com os participantes do sistema, uma vez que a execução destas fases foram realizadas sem que houvesse a presença de qualquer membro da equipe de desenvolvimento. Em outras palavras, com o intuito de se reduzir custos de desenvolvimento, a equipe de desenvolvimento elaborou vários questionários que foram encaminhados ao gerente de projeto, e este por sua vez, foi quem realizou as entrevistas com os demais participantes do sistema.

Considerando-se que o problema apresentado possuía um número reduzido de entidades e os fluxos de dados eram bastante semelhantes, podemos afirmar que o sistema em si superou de forma considerável as expectativas de todos os participantes, visto que o protótipo sofreu apenas três revisões: uma referente aos layouts de tela, outra referente aos layouts dos relatórios e a terceira e última sobre questões referentes à formatação de textos e números, como por exemplo: tipo numérico inteiro ou real, número de casas decimais, quantidade de caracteres para campos alfanuméricos, dentre outras.

A versão final do sistema é composta de quatro telas básicas:

- Principal – que permite a navegação por três áreas especificas: cadastramento, consultas/relatórios e exportação/importação de dados.
- Tela de cadastramento – que permite, como o próprio nome já diz, a inclusão e/ou exclusão de chaves de buscas e dados sobre cada tipo de equipamento (figura 8.1).

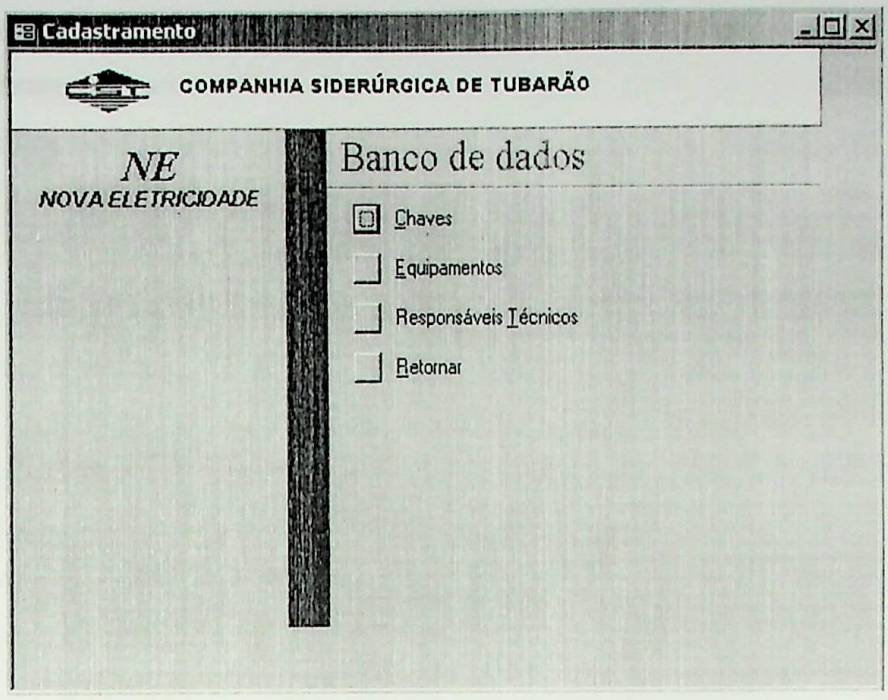


Figura 8.1 – Tela de cadastramento do SIPE

- Tela consultas/relatórios – que em função de um conjunto pré-definido de chaves permite recuperar informações sobre um dado equipamento e, se necessário for, imprimir os respectivos relatórios (figura 8.2).

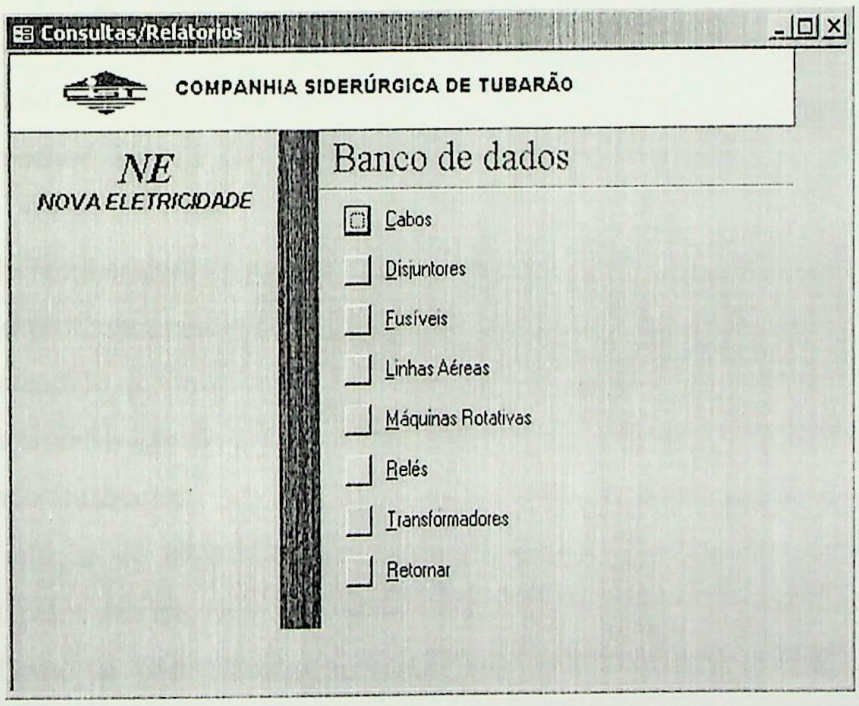


Figura 8.2 – Tela de consultas/relatórios do SIPE

- Tela exportação/importação de dados – permite que parte ou a totalidade das informações contidas na base de dados sejam repassadas a terceiros quando da realização de novos estudos a respeito da planta elétrica da CST (figura 8.3).

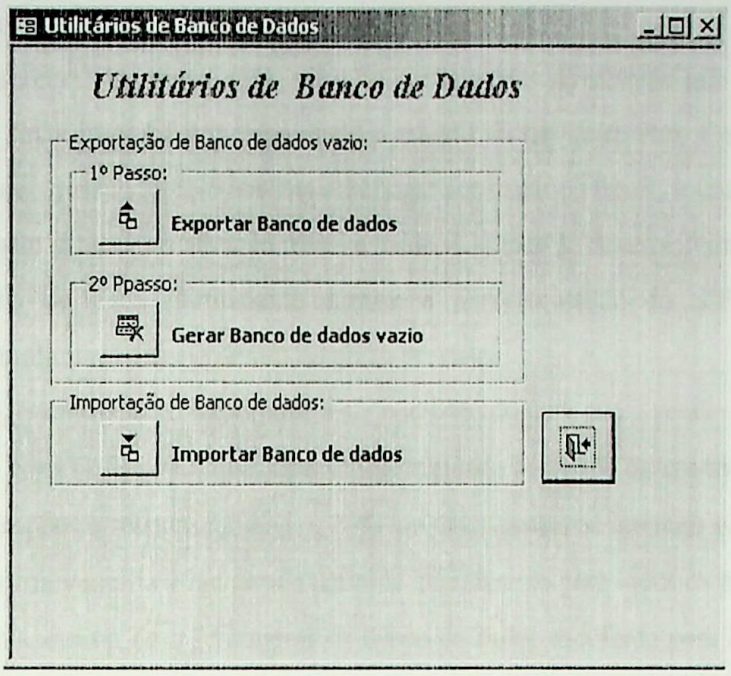


Figura 8.3 – Tela de exportação/importação do SIPE

As demais telas do SIPE, tais como formato de relatório e opções de consulta são apresentadas no anexo III – *Padrões de relatórios.*

É importante lembrar que um dos motivos da escolha da ferramenta de banco de dados – Microsoft Access deveu-se também ao fato de a equipe de usuários da CST possuir um certo grau de familiaridade com a mesma e que eles tinham como um dos objetivos, após a conclusão do protótipo, continuarem eles próprios a efetuar as manutenções no SIPE. Deve ficar bem entendido que manutenção, neste caso, não se limita somente ao sentido exato da palavra, mas também quando necessária for, a implementação de novas rotinas, o que de fato vem ocorrendo atualmente.

Com relação ao número de questionários elaborados pelo analista de sistemas e respondidos pelos futuros usuários e participantes, este também se mostrou suficiente e adequado. Como se pode observar, o primeiro questionário (*Anexo I –Expectativas do usuário*) procurou identificar o perfil de cada participante do sistema e as possíveis formas como eles registravam e recuperavam informações sobre o sistema elétrico-eletrônico da CST.

Por outro lado, o segundo questionário (*Anexo II – Especificação de campos*) procurou identificar o grau de importância que cada campo possuía para os participantes do sistema, de

modo a permitir que as chaves para armazenamento e recuperação de informações sobre cada tipo de equipamento da planta elétrica fosse identificada. Note que este segundo questionário foi montado em função das respostas obtidas pelo primeiro questionário e pelos modelos de registro dessas informações que cada usuário utilizava.

Também merecem destaque aqui, todos os participantes do sistema que mantiveram o tempo todo uma forte interação, mesmo tendo o mundo virtual entre eles. Começando pelo gerente de sistemas, passando pelo analista até chegar aos usuários finais, todos sem exceção, mantiveram um alto grau de motivação durante todas as etapas de desenvolvimento do SIPE. O forte empenho de cada participante durante a implementação do SIPE teve como conseqüência a implantação deste dentro do prazo previsto.

Além disto, outro aspecto importante a ser considerado para que a versão final do SIPE fosse implantada com sucesso foi que, uma vez identificado cada tipo de usuário e a eles atribuídas suas funções e responsabilidades, cada um deles procurou interagir com o sistema, respeitando de forma sumária a hierarquia definida inicialmente para todos os participantes.

Em resumo, apesar de a ferramenta de banco de dados escolhida para implementar o software em questão apresentar algumas limitações, foi perfeitamente possível atender todas as solicitações feitas inicialmente, quando da proposta de desenvolvimento do novo sistema, além de outras solicitações durante todas as etapas de desenvolvimento do SIPE (Sistema de Informações da Planta Elétrica).

9. Conclusão

Em um ambiente fabril de grande porte a maneira como as informações são armazenadas e recuperadas é de importância fundamental para a sobrevivência das empresas. Tal afirmação pode ser vista como uma visão de negócios, mas o que muitos empresários não percebem é a importância desta visão sobre suas ferramentas. Desta forma a compra de bens e serviços pode ser otimizada por meio da utilização de ferramentas que permitam uma melhor gerência de seus recursos de hardware.

No caso da CST, com a implantação do SIPE, houve uma mudança na forma de armazenamento e recuperação das informações, mais especificamente com relação às informações pertinentes ao seu sistema elétrico. Tal mudança permitiu um aumento substancial na confiabilidade das informações sobre o sistema elétrico, pelo simples fato de que a partir de então passou a existir um controle mais efetivo sobre quem acessava as informações e também sobre quem poderia modificá-las.

O aumento da confiabilidade trouxe por conseqüência redução dos custos decorrentes de novos projetos referentes à manutenção e/ou ampliação do sistema elétrico da CST. Na forma antiga de armazenamento de informações pertinentes ao seu sistema elétrico, os novos projetos de manutenções e/ou ampliações exigiam sempre uma certa disponibilidade "extra" de tempo e recursos financeiros. Qualquer que fosse a equipe responsável pela execução de um novo projeto, de posse das informações sobre os equipamentos envolvidos, havia sempre a necessidade de que essa equipe fizesse uma verificação *in loco* no sistema elétrico de modo a certificar a veracidade das informações sobre tais equipamentos.

Para desenvolver o SIPE, optamos pela metodologia da prototipação. A escolha de tal metodologia se baseou nos seguintes aspectos:

- Necessidade premente de algo "palpável" e funcional.
- Que o sistema fosse evolucionário.
- Por não conhecermos detalhes gerenciais e operacionais da CST.
- O nível de especificação e conhecimento de onde se quer chegar não era tão intenso quanto se exige nas metodologias tradicionais.
- Os requisitos não eram bastantes claros.
- A criação do banco de dados se da de uma forma mais livre e interativa, isto é, a medida que o domínio do problema em questão aumenta, pode-se facilmente modificar tal banco de dados.

Levando em consideração que o SIPE foi desenvolvido quase que exclusivamente por meio da interação virtual entre os participantes do sistema, durante a fase de coleta de dados tivemos o cuidado de procurar levantar o perfil técnico dos participantes, bem como o impacto psicológico que tal sistema causaria sobre cada um deles. Este último com o intuito de detectar possíveis focos de resistência ao novo sistema. Assim sendo, elaboramos dois questionários que foram distribuídos e respondidos por todos os usuários participantes. Em ambos os questionários tivemos a preocupação em ressaltar inicialmente a importância do usuário em questão frente ao desenvolvimento do novo sistema por meio de um texto, bem como explicar o objetivo de cada um dos questionários (vide anexos I e II).

Durante as etapas de desenvolvimento e implementação foi dada quase que de forma simultânea atenção aos seguintes quesitos:

- Dados: devido à precariedade assumida do protótipo é perfeitamente viável contarmos com entradas manuais para alimentar o banco de dados. Analogamente à condição das entradas manuais há a possibilidade de se estudar e detalhar com maior análise entre causas e efeitos as eventuais interfaces requeridas para operação em conjunto com outros sistemas e/ou banco da dados existentes na organização.
- Telas: representam uma das partes mais tangíveis ligadas aos usuários do sistema.
- Processamento: representa as variáveis e fórmulas de cálculos propriamente ditas, o que neste projeto praticamente não existiu.
- Informações: a fim de refinar o entendimento e os requisitos de informações a serem geradas pelo sistema em questão. Relatórios foram incluídos no protótipo de modo que criticas de consistência fossem feitas pelos usuários do mesmo.
- Segurança: em decorrência da identificação dos tipos de usuários com perfis e atribuições distintas foi possível determinar que tipos de níveis de acesso o protótipo teria.

Podemos ainda afirmar que essa "simultaneidade" permitida pela prototipação visou a:

- priorizar os problemas a serem considerados, em ordem do interesse da aplicação dos conceitos de gestão econômica do projeto;
- desenvolver rapidamente um sistema pequeno e precário, porém útil para os participantes do sistema;
- permitir a avaliação constante e permanente do protótipo pelos participantes;
- garantir subsídios para o sistema definitivo e para outros protótipos.

Analisando-se os resultados após todas as etapas do projeto proposto terem sido "concluídas", é perfeitamente possível afirmar que o projeto superou as expectativas iniciais sob todos os aspectos do desenvolvimento. É fato que havia certa apreensão com relação aos recursos das ferramentas escolhidas, mas as dificuldades encontradas devido a tais limitações puderam ser contornadas sem que se perdesse a referência de tais ferramentas.

Um exemplo disso foi a questão da importação e exportação de dados. Quando se faz necessário realizar o estudo de uma determinada área, na maioria das vezes somente os equipamentos pertencentes a essa área precisam ser analisados novamente.

O Microsoft Access possui o recurso chamado de *replicação de banco de dados*, este recurso permite que seja feita mais de uma copia do banco de dados original para que vários usuários trabalhem em rede de forma simultânea ou isoladamente. Aparentemente um recurso muito bom e de fácil uso, já que a execução de tal processo se dá por seleção de menus e opções. No caso da CST, esse recurso se mostrou inadequado, pois por questões de segurança não é aconselhável disponibilizar toda uma base de dados. A solução então, foi construir rotinas em SQL suportadas pelo Microsoft Access, que exportassem as chaves de acesso aos registros e somente as informações pertinentes aos equipamentos das áreas envolvidas no estudo em questão.

Portanto, este trabalho possui muitas opções de melhorias tanto em nível de ferramentas de software quanto em nível de novas rotinas. Com relação às ferramentas de software pode-se destacar:

- Utilização de um sistema de gerenciador de banco de dados mais robusto que suporte aplicações web, por exemplo, MySQL, MS-SQL, etc.
- Implementá-lo em uma linguagem que suporte interação via *browser*, por exemplo, Java, PHP, etc.

E com relação a implementação de novas rotinas destacamos:

- Otimização das rotinas de importação e exportação de dados.
- Controle de manutenção preventiva.
- Controle de manutenção corretiva.
- Interligar os diagramas unifilares com os registros dos equipamentos.
- Elaboração do manual *online* do usuário.

Por fim, para concluir este trabalho, ressalta-se aqui que toda a implementação foi feita seguindo rigidamente as determinações da CST, e que o produto final poderia apresentar resultados bem melhores se ferramentas de software mais robustas tivessem sido utilizadas.

10. Referências bibliográficas

1. Webster's New Collegiate Dictionary, Springfield, Mass.: G & C. Merriam Company, 1977.
2. CHURCHMAN, C. W. Introdução à teoria dos sistemas, Petrópolis, Editora Vozes, 1971.
3. YOURDON, E. Análise estruturada moderna. Rio de Janeiro, Editora Campus, 1990.
4. OLIVEIRA, Djalma de Pinho Rebouças. Sistemas de informações gerenciais. São Paulo, Atlas, 1992.
5. WYSK, R. B. Métodos de planejamento de sistemas de informação: um estudo de fronteiras. Rio de janeiro, COPPAD/UFRJ, 1980.
6. LANGEFORS, B. Theoretical analisys of information systems. Philadelphia, Anerbach, 1973.
7. POLLONI, E. G. F. Administrando sistemas de informação. São Paulo, Futura, 2000.
8. ALTER, Steven. Information systems: a management perspective. USA, Addison Publishing Company, 1992.
9. SETZER, VALDEMAR W. Banco de dados – conceitos, modelo, gerenciadores, projeto lógico e projeto físico. São Paulo, Editora Edgard Blücher Ltda., 3ª edição, 1989.
10. GROTHENDIEK, A. La nouvelle eglise universelle, in Jaulin, R. (Ed.) Pourquoi la mathématique?, Union Generales d'Editions, Paris, 1984.
11. TSICHRITZIS, D. C. e LOCHOWSKY, F. M. Data models, Englewood Cliffs, Prentice-Hall, 1984.
12. BORKIN S. A. Data models: a semantic approach for database systems. The MIT Press, Cambridge, 1980.
13. SUNDGREN, B. Theory of databases, Petroccelli, New York, 1975.
14. TEOREY, T. J. e FRY, J. P. Design of database structures, Englewood Cliffs, Prentice-Hall, 1982.
15. ANSI / X3 / SPARC. Study group on database management systems, ACM SIGMOD Bulletin, 7, 2,1975.
16. DATE, C. J. Introdução a sistemas de banco de dados, Rio de Janeiro, Editora Campus, 1991.

17. LEESON, M. Systems analysis and design. Chicago, Science Research Associates, 1981.
18. REZENDE, Denis Alcides; ABREU, Aline França de. Tecnologia da informação aplicada a sistemas de informação empresariais. São Paulo, Atlas, 2001.
19. Dalfovo O, Boni, A P e Maia L F J: Sistemas de informação baseado em datawarehouse aplicado na área de administração de materiais, in: M A S N Nunes (org): XI Seminário Regional de Informática (anais, pp. 119-127). Santo Ângelo, RS: URI, 2001.
20. ALTER, Steven. Information systems: a management perspective. USA, Addison Publishing Company, 1992.
21. BINDER, Fabio Vinícios. Sistemas de apoio à decisão. São Paulo, Érica, 1994.
22. FFURLAN, José Davi; IVO, Ivonildo da Motta; AMARAL, Francisco Piedade. Sistemas de informações executivas. São Paulo, Makron Books, 1994.
23. GANDARA, Fernando. EIS sistemas de informações empresariais. São Paulo, Érica, 1995.
24. ROOK, P. Controlling software projects, Califórnia, Editora Tutorial, 1988.
25. ROCHA, Ana Regina Cavalcanti da. Análise e projeto estruturado de sistemas, Rio de Janeiro, Editora Campus, 1986.
26. STAA, A. Engenharia de programas, Rio de Janeiro, LTC – Livros Técnicos e Científicos, 1987.
27. COAD, P.; YOURDON, E., Análise baseada em objetos, 2ª Ed., Rio de Janeiro, Editora Campus, 1992.
28. PRESMANN, R. S., Engenharia de software, São Paulo, Mc Graw-Hill, 1994.
29. BINGHAM, J. E., GARTH, W.P.D., Manual de análise de sistemas, Rio de Janeiro, Inerciência, 1996.
30. LOBO, José E M. Aplicação do desdobramento da função qualidade no gerenciamento da configuração de sistemas, (dissertação de mestrado, Luiz Gonzaga Mariano de Souza, orientador). Itajubá, MG, EFEI, 2000.
31. MARTIN, J., McCLURE, C., Técnicas estruturadas e case, São Paulo, Editora Makron Books, 1991.
32. LAUDON, Kenneth C.; LAUDON, Jane P. Sistemas de informação com internet, Rio de Janeiro, LTC – Livros Técnicos e Científicos S. A., 1999.

33. CASTRO, Alexandre Ramires de; COSTA, Elenio Pereira da. Um breve estudo sobre a ferramenta RAD Rational Rose, Axcel Books do Brasil Editora Ltda., Revista Developers' CIO Magazine, ano 5, nº 60, agosto de 2001.
34. BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML, Guia do usuário, 6ª edição, Rio de Janeiro, Editora Campus, 1999.
35. COAD, P.; YOURDON, E. Análise baseada em objetos, 2ª edição, Rio de Janeiro, Editora Campus, 1992.
36. WIFRS-BROCK, R.; WILKERSON, B. Desingning object-oriented software, New Jersey, Prentice Hall, 1990.
37. RUMBAUGH, J.; et al. Modelagem e projeto baseados em objetos, Rio de Janeiro, Editora Campus, 1994.

ANEXO I – Expectativas do usuário

Caro participante do sistema,

Você está recebendo este questionário para responder, porque você foi indicado por alguém que julga importante a sua participação na implantação de um novo sistema digital de armazenamento e recuperação de informações da planta elétrica da CST.

O questionário em questão tem por objetivo identificar as suas expectativas com relação ao novo sistema que será implantado. Para tanto se faz necessário que você responda ao máximo de questões.

Se durante o preenchimento de tal questionário surgir alguma dúvida, procure o senhor *fulano de tal, setor xxxx, ramal 999, email fulanodetal@cst.com.br,* pois ele é o gerente do projeto e responsável indicado pela CST.

Durante ou ao final do preenchimento deste questionário, sinta-se a vontade de tecer qualquer tipo de comentário que você desejar, pois afinal este sistema será seu quando estiver pronto.

Gostaríamos de ressaltar mais uma vez que o desenvolvimento do novo sistema tem por objetivo básico facilitar e democratizar a forma como as informações sobre o sistema elétrico da CST são armazenadas, recuperadas e distribuídas.

Assim sendo, nós da equipe de desenvolvimento esperamos entregar a você um produto que tenha ao menos seu toque pessoal.

Por fim lembramos a você que o senhor fulano de tal passará recolhendo este questionário no dia dd/mm/aaaa.

Abraços.

A equipe de desenvolvimento.

Expectativas do usuário

1. Nome:

2. Cargo/função:

3. Usa computador em seu trabalho?

   Sim ☐ Não ☐

4. Responda a este item somente se a resposta do item 3 tiver sido SIM. Descreva a configuração do computador que você utiliza no trabalho. Itens que você não queira responder deixe em branco.

   Marca do Processador Intel ☐ AMD ☐ Outro ☐
   Modelo Pentium II ☐
   Pentium III ☐
   Celeron ☐
   K6 ☐
   K6-II ☐
   Outro ☐
   Freqüência ☐
   Capacidade do HD ☐
   Memória RAM ☐

5. Responda a este item somente se a resposta do item 3 tiver sido SIM. Quais são os softwares instalados no computador com o qual você trabalha. Observação: Itens que você não queira responder deixe em branco.

   Sistema operacional e versão: ☐
   Editor de texto: ☐
   Planilha eletrônica: ☐
   Slide show: ☐
   Banco de dados: ☐
   CAD: ☐
   Antivírus: ☐
   Compilador: ☐
   Outros (descrever): ☐

6. Usa computador em casa?

   Sim ☐ Não ☐

7. Responda a este item somente se a resposta do item 6 tiver sido SIM. Descreva a configuração do computador que você utiliza no trabalho. Observação: Itens que você não queira responder deixe em branco.

Marca do Processador Intel ☐ AMD ☐ Outro ☐
Modelo Pentium II ☐
Pentium III ☐
Celeron ☐
K6 ☐
K6-II ☐
Outro ☐
Freqüência ☐
Capacidade do HD ☐
Memória RAM ☐

8. Responda a este item somente se a resposta do item 6 tiver sido SIM. Quais são os softwares instalados no computador com o qual você trabalha. Observação: Itens que você não queira responder deixe em branco.

Sistema operacional e versão	
Editor de texto:	
Planilha eletrônica:	
Slide show:	
Banco de dados:	
CAD:	
Antivírus:	
Compilador:	
Outros (descrever):	

9. Possui alguma forma de registrar as tarefas executadas por você?

Sim ☐ Não ☐

10. Responda a este item somente se a resposta do item 9 tiver sido SIM. De que forma você registra estas informações?

Segue um padrão da CST ☐ Padrão próprio ☐ Outro padrão ☐

*Anexe a este questionário uma cópia do(s) modelo(s) que você utiliza para registrar as informações, identificando-o(s) com o seu nome e com o número **10** seqüenciado pelas letras do alfabeto (por exemplo: 10a, 10b, 10c, ...).*

11. Responda a este item somente se você respondeu o item 10. Na forma utilizada por você para registrar a execução de tarefas, indique no verso de cada padrão que você anexou ao item 11 um mínimo de três campos e um máximo de cinco campos que você julga serem os mais importantes na recuperação das informações contidas em cada formulário.

12. Você já fez algum curso relacionado com a área de informática?

    Sim ☐ Não ☐

13. Responda a este item somente se a resposta do item 12 tiver sido SIM. Quais foram os cursos e a carga horária de cada um deles?

I	
ii	
Iii	
Iv	
...	

14. Qual o nível de envolvimento que você tem com relação ao uso das informações sobre sistema elétrico da CST?

    Alto ☐ Médio ☐ Baixo ☐

15. Existe um padrão para registro de informações sobre o sistema elétrico da CST?

    Sim ☐ Não ☐

16. Responda a te item somente se a resposta do item 15 tiver sido SIM. Por que você não o utiliza?

17. Com relação ao cargo que você ocupa, quais são as informações sobre o sistema elétrico da CST que você julga importantes para a realização de suas atividades? (Aqui você pode escrever um texto a respeito, listar na forma de tópicos ou ainda anexar tabelas e/ou formulários).

18. Que tipo de benefícios você espera obter com a centralização das informações sobre a planta elétrica da CST em uma base de dados única?

ANEXO II – Especificação de campos

Caro participante do sistema,

Neste documento encontra-se a relação dos tipos de equipamentos existentes na planta elétrica da CST e seus respectivos campos informativos. Tais informações foram extraídas do ***"Questionário de expectativas do usuário"*** preenchido por você anteriormente. Agora se torna necessário que você forneça informações mais especificas sobre cada tipo de equipamento baseado nos campos listados em cada uma das tabelas contidas neste documento.

Todos os questionários possuem um conjunto idêntico de quesitos que são explicados a seguir.

a) Nome do campo – de significado óbvio.

b) Representação da unidade (Hz, kV, A, ...) – informe o tipo de unidade que corresponde ao campo em questão.

c) Preenchimento Obrigatório: Sim ou Não? – *sim* para o caso da informação nunca poder ser deixada em branco e *não* quando a informação não for muito relevante.

d) Tipo do campo A – alfanumérico ou N – numérico – *alfanumérico* para campos que aceitem letras, números e outros caracteres e *numérico* para campos que aceitem exclusivamente números.

e) Tamanho do campo – comprimento maximo que o campo deve assumir, independentemente de o campo ser alfanumérico ou numérico.

f) Nº de casas decimais – em se tratando de campo numérico indique quantas casas decimais ele deve possuir. O valor fornecido aqui será descontado do valor fornecido no item *tamanho do campo*.

g) Descartar este campo? – indique com *sim* caso o campo em questão não possua nenhuma relevância para você.

h) Grau de importância (escala de 1 a 10) – variando de 1 para importância máxima até 10 para pouco relevante mas que deve constar na tabela.

Os campos marcados de vermelho são campos que estamos sugerindo como base para recuperação das informações sobre um dado equipamento ou conjunto de equipamentos.

Caso não exista na lista de cada equipamento algum campo que você julgue importante, sinta-se à vontade em acrescenta-lo ao final da lista em questão.

Para maiores esclarecimentos sobre o preenchimento deste questionário, favor contatar o senhor *fulano de tal, email fulanodetal@cst.com.br, ramal 999*, que é o gerente do projeto e responsável indicado pela CST.

Este questionário será recolhido pelo senhor *fulano de tal,* no dia dd/mm/aaaa.

Abraços.

A equipe de desenvolvimento.

Nome do equipamento: ***Cabos de força***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
1.	Área							
2.	Bitola							
3.	Circuito							
4.	Comprimento							
5.	Condutores por fase							
6.	Configuração							
7.	Folha de seletividade							
8.	Impedância R							
9.	Impedância X							
10.	Isolamento							
11.	Localização							
12.	Padrão							
13.	Reatância							
14.	Resistência							

Nome do equipamento: ***Linhas aéreas***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
1.	Área							
2.	Bitola							
3.	Circuito							
4.	Comprimento							
5.	Condutores por fase							
6.	Configuração							
7.	Distância Dab							
8.	Distância Dbc							
9.	Distância Dca							
10.	Folha de seletividade							
11.	Impedância R							
12.	Impedância X							
13.	Isolamento							
14.	Localização							
15.	Padrão							
16.	Reatância							
17.	Resistência							

Nome do equipamento: ***Disjuntores***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
1.	Área							
2.	Circuito							
3.	Discriminação							
4.	Fabricante							
5.	Faixa de ajuste							
6.	Folha de seletividade							
7.	Função							
8.	Graduação							
9.	Localização							
10.	Tensão							
11.	Tipo							
12.	Transformador de corrente							

Nome do equipamento: ***Fusíveis***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
1.	Área							
2.	Circuito							
3.	Classe de tensão							
4.	Código							
5.	Dimensão A							
6.	Dimensão B							
7.	Dimensão C							
8.	Dimensão D							
9.	Dimensão D1							
10.	Dimensão D2							
11.	Dimensão E							
12.	Dimensão F							
13.	Dimensão G							
14.	Dimensão H							
15.	Dimensão L							
16.	Dimensão M							
17.	Dimensão O							
18.	Discriminação							
19.	Fabricante							
20.	Folha de seletividade							
21.	Formato							
22.	Função							
23.	In / curva							
24.	Localização							
25.	Quantidade							
26.	Tipo							

Nome do equipamento: ***Relés***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
1.	Área							
2.	Circuito							
3.	Discriminação							
4.	Fabricante							
5.	Faixa de ajuste							
6.	Folha de seletividade							
7.	Função							
8.	Graduação							
9.	Localização							
10.	Tensão							
11.	Tipo							
12.	Transformador de corrente							
13.	Transformador de potencia							
14.	Tipo de rele							

Nome do equipamento: ***Transformadores de 2 enrolamentos***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
1.	Área							
2.	Circuito							
3.	Conexão primária							
4.	Conexão secundária							
5.	Denominação							
6.	Discriminação							
7.	Fabricante							
8.	Fabricante							
9.	Folha de seletividade							
10.	Impedância R							
11.	Impedância X							
12.	Impedância Z							
13.	Potência base							
14.	Potência Vent. Forçada							
15.	Potência Vent. Normal							
16.	Relação X/R							
17.	Tensão base							
18.	Tensão primária							
19.	Tensão secundária							

Nome do equipamento: ***Transformadores de 3 enrolamentos***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
1.	Área							
2.	Circuito							
3.	Conexão primária							
4.	Conexão secundária							
5.	Conexão terciária							
6.	Denominação							
7.	Discriminação							
8.	Fabricante							
9.	Folha de seletividade							
10.	Potência base							
11.	Potência primária							
12.	Potência secundária							
13.	Potência terciária							
14.	Relação X/R							
15.	Rps							
16.	Rpt							
17.	Rst							
18.	Tensão base							
19.	Tensão primária							
20.	Tensão secundária							
21.	Tensão terciária							
22.	Xps							
23.	Xpt							
24.	Xst							
25.	Zps							
26.	Zpt							
27.	Zst							

Nome do equipamento: ***Gerador síncrono***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
1.	Área							
2.	Constante de inércia em %							
3.	Constante de inércia em pu							
4.	Constante de inércia em s							
5.	Corrente nominal							
6.	Denominação							
7.	Discriminação							
8.	Fabricante							
9.	Fator de potencia							
10.	Folha de seletividade							
11.	Freqüência							
12.	Máquina acionada							
13.	Modelo							
14.	Momento de inércia em %							
15.	Momento de inércia em pu							
16.	Momento de inércia em s							
17.	Número de pólos							
18.	Número de série							
19.	Potência							
20.	R do estator em pu							
21.	R do estator em s							
22.	R do estator em%							
23.	Rotação							
24.	RPM							
25.	T´do em							
26.	T´do em							
27.	T´do em							

Nome do equipamento: ***Gerador síncrono***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
28.	T´qo em %							
29.	T´qo em pu							
30.	T´qo em s							
31.	T”do em							
32.	T”do em							
33.	T”do em							
34.	T”qo em %							
35.	T”qo em pu							
36.	T”qo em s							
37.	Tensão							
38.	Tipo							
39.	Tipo							
40.	Tipo da maquina							
41.	X dispersão do estator em %							
42.	X dispersão do estator em pu							
43.	X dispersão do estator em s							
44.	X´d em %							
45.	X´d em pu							
46.	X´d em s							
47.	X”d em %							
48.	X”d em pu							
49.	X”d em s							
50.	X”q em %							
51.	X”q em pu							
52.	X”q em s							
53.	Xd em %							
54.	Xd em pu							

Nome do equipamento: ***Gerador síncrono***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
55.	Xd em s							
56.	Xq em %							
57.	X'q em %							
58.	Xq em pu							
59.	X'q em pu							
60.	Xq em s							
61.	X'q em s							

Nome do equipamento: ***Motor síncrono***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
1.	Área							
2.	Constante de inércia em %							
3.	Constante de inércia em pu							
4.	Constante de inércia em s							
5.	Corrente de partida							
6.	Corrente nominal							
7.	Corrente nominal							
8.	Denominação							
9.	Discriminação							
10.	Discriminação							
11.	Fabricante							
12.	Fabricante							
13.	Fator de potencia							
14.	Folha de seletividade							
15.	Freqüência							
16.	Máquina acionada							
17.	Modelo							
18.	Momento de inércia em %							
19.	Momento de inércia em pu							
20.	Momento de inércia em s							
21.	Número de pólos							
22.	Número de série							
23.	Potência							
24.	R do estator em pu							
25.	R do estator em s							
26.	R do estator em%							
27.	Rotação							

Nome do equipamento: ***Motor síncrono***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
28.	RPM							
29.	RPM							
30.	T´do em							
31.	T´do em							
32.	T´do em							
33.	T´qo em %							
34.	T´qo em pu							
35.	T´qo em s							
36.	T”do em							
37.	T”do em							
38.	T”do em							
39.	T”qo em %							
40.	T”qo em pu							
41.	T”qo em s							
42.	Tempo de partida							
43.	Tempo de rotor bloqueado							
44.	Tensão							
45.	Tensão							
46.	Tipo							
47.	Tipo							
48.	Tipo da máquina							
49.	X dispersão do estator em %							
50.	X dispersão do estator em pu							
51.	X dispersão do estator em s							
52.	X”d							
53.	X´d em %							
54.	X´d em pu							

Nome do equipamento: ***Motor síncrono***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
55.	X´d em s							
56.	X"d em %							
57.	X"d em pu							
58.	X"d em s							
59.	X"q em %							
60.	X"q em pu							
61.	X"q em s							
62.	X'd							
63.	Xd em %							
64.	Xd em pu							
65.	Xd em s							
66.	Xq em %							
67.	X'q em %							
68.	Xq em pu							
69.	X'q em pu							
70.	Xq em s							
71.	X'q em s							

Nome do equipamento: ***Motor de indução***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
1.	Área							
2.	Denominação							
3.	Potência							
4.	Tensão							
5.	RPM							
6.	Tempo de Rotor bloqueado							
7.	Tempo de Partida							
8.	Folha de seletividade							
9.	Discriminação							
10.	Fabricante							
11.	Reatância %							
12.	Reatância pu							
13.	Corrente Nominal							
14.	Corrente de Partida							

Nome do equipamento: ***Motor equivalente***

	Nome do campo	Representação da unidade (Hz, kV, A, ...)	Preenchimento Obrigatório: Sim ou Não?	Tipo do campo A – alfanumérico N – numérico	Tamanho do campo	Nº de casas decimais	Descartar este campo?	Grau de importância (escala de 1 a 10)
1.	Área							
2.	Discriminação							
3.	Potência							
4.	Tensão							
5.	Reatância X"d %							
6.	Denominação							
7.	Reatância X"d pu							

ANEXO III – Padrões de relatórios

COMPANHIA SIDERÚRGICA DE TUBARÃO	Nome da empresa que executou o serviço	Feito por:	Data:	Folha:
		Aprovado por:	Revisão:	00 de nn
Área:		Equipamento: CABOS DE FORÇA		

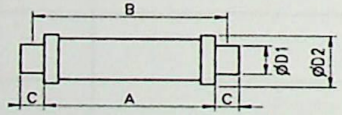
Item	Denominação	Tipo	Classe de Isol. (kV)	Bitola	Comprimento (km)	Impedância (Ohm/km/condutor)		Impedância/Fase (base 100 MVA)	
						R	X	Rpu	Xpu

COMPANHIA SIDERÚRGICA DE TUBARÃO	Nome da empresa que executou o serviço	Feito por:	Data:	Folha:
		Aprovado por:	Revisão:	00 de nn
Área:		Equipamento: DISJUNTORES		

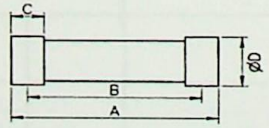
Item	Folha de seletividade	Localização/circuito	Fabricante	Tipo	Função	Faixa de Ajuste	Graduação

COMPANHIA SIDERÚRGICA DE TUBARÃO	Nome da empresa que executou o serviço	Feito por:	Data:	Folha:
		Aprovado por:	Revisão:	00 de nn
Código:		Equipamento:	FUSÍVEIS	

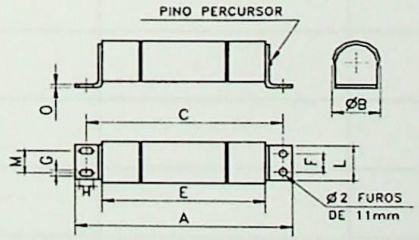
Item	Folha de seletividade	Área	Localização / circuito	Qtde.	Fabricante	Tipo	Classe de tensão (kV)	In/Curva	Formato	Dimensões (mm)						
										A	B	C	D	D1	D2	
										E	F	G	H	L	M	O



FORMATO–1



FORMATO–2



FORMATO–3

COMPANHIA SIDERÚRGICA DE TUBARÃO	Nome da empresa que executou o serviço	Feito por:	Data:	Folha:
		Aprovado por:	Revisão:	00 de nn
Área:		Equipamento: LINHAS AÉREAS		

Item	denominação	Tipo	Classe de Isol. (kV)	Bitola	Comprimento (km)	Impedância (Ohm/km/condutor)		Impedância/Fase (base 100 MVA)	
						R	X	Rpu	Xpu

COMPANHIA SIDERÚRGICA DE TUBARÃO	Nome da empresa que executou o serviço	Feito por:	Data:	Folha:
		Aprovado por:	Revisão:	00 de nn
Área:		Equipamento: RELÉS		

Item	Folha de seletividade	Localização/circuito	Fabricante	Tipo	Função	Faixa de Ajuste	Graduação	TC/TP

COMPANHIA SIDERÚRGICA DE TUBARÃO	Nome da empresa que executou o serviço	Feito por:	Data:	Folha:
		Aprovado por:	Revisão:	00 de nn
Área:		Equipamento: TRANSFORMADORES DE 2 ENROLAMENTOS		

Item	Denominação	Fabricante	Potência (MVA)	Tensão (kv)		Conexão		Impedância Z%	Impdedância (base 100 MVA)	
				Prim.	Sec.	Prim.	Sec.		Rpu	Xpu

COMPANHIA SIDERÚRGICA DE TUBARÃO	Nome da empresa que executou o serviço	Feito por:	Data:	Folha:
		Aprovado por:	Revisão:	00 de nn
Área:		Equipamento: TRANSFORMADORES DE 3 ENROLAMENTOS		

Item	Denominação	Fabricante	Potencias (MVA)	Tensões (kv)	Conexões	Impedâncias	X/R	Impedâncias pu (base 100 MVA)	
			Primária Secundária Terciária	Primária Secundária Terciária	Primária Secundária Terciária	Zps% Zpt% Zst%		Rps Rpt Rst	Xps Xpt Xst

COMPANHIA SIDERÚRGICA DE TUBARÃO	Nome da empresa que executou o serviço	Feito por:	Data:	Folha:
		Aprovado por:	Revisão:	00 de nn
Área:		Equipamento: MOTORES DE INDUÇÃO		

Item	Denominação	Potência (kW)	Tensão (kv)	RPM	X"d% Base própria	X"dpu Base 100 MVA	Corrente Nominal (A)	Corrente de Partida (A)	Tempo de Partida (s)	Tempo de Rotor bloqueado (s)

COMPANHIA SIDERÚRGICA DE TUBARÃO	Nome da empresa que executou o serviço	Feito por:	Data:	Folha:
		Aprovado por:	Revisão :	00 de nn
Área:		Equipamento: MOTORES EQUIVALENTES		

Item	Denominação	Potência (kW)	Tensão (kV)	X"d % Base própria	X"dpu Base 100 MVA

COMPANHIA SIDERÚRGICA DE TUBARÃO	Nome da empresa que executou o serviço

Área:

DADOS DE GERADORES SÍNCRONOS

Denominação:			
Fabricante:			
Número de série:		Tipo:	
Modelo:		Potência nominal:	(kVA)
Tensão:	(kV)	Corrente nominal:	(A)
Rotação:	(RPM)	Fator de potência:	
Freqüência:	(Hz)	Número de pólos:	
Máquina acionadora:		Tipo da máquina:	

	PARÂMETROS	Constante de Tempo (Segundo)	Base Própria %	Base 100 MVA PU
R	RESISTÊNCIA DO ESTATOR			
X	REATÂNCIA DE DISPERSÃO DO ESTATOR			
Xd	REATÂNCIA SÍNCRONA DE EIXO DIRETO			
X'd	REATÂNCIA TRANSITÓRIA DE EIXO DIRETO			
X"d	REATÂNCIA SUBTRANSITÓRIA DE EIXO DIRETO			
T'do	CONSTANTE DE TEMPO TRANS. DE EIXO DIRETO A VAZIO			
T"do	CONSTANTE DE TEMPO SUBTRANS. DE EIXO DIRETO A VAZIO			
Xq	REATÂNCIA SÍNCRONA DE EIXO EM QUADRATURA			
X'q	REATÂNCIA TRANSITÓRIA DE EIXO EM QUADRATURA			
X"q	REATÂNCIA SUBTRANSITÓRIA DE EIXO EM QUADRATURA			
T'qo	CONSTANTE DE TEMPO TRANS. DE EIXO EM QUAD. A VAZIO			
T"qo	CONSTANTE DE TEMPO SUBTRANS. DE EIXO EM QUAD. A VAZIO			
GD2	MOMENTO DE INÉRCIA (Kg x m2)			
H	CONSTANTE DE INÉRCIA (s)			

Observações:

Preparado por:	Data:	Folha:
Visto por:	Aprovado por:	00 de nn

COMPANHIA SIDERÚRGICA DE TUBARÃO	Nome da empresa que executou o serviço

Área:

DADOS DE GERADORES SÍNCRONOS

Denominação:			
Fabricante:			
Número de série:		Tipo:	
Modelo:		Potência nominal:	(kVA)
Tensão:	(kV)	Corrente nominal:	(A)
Rotação:	(RPM)	Fator de potência:	
Freqüência:	(Hz)	Número de pólos:	
Máquina acionadora:		Tipo da máquina:	

	PARÂMETROS	Constante de Tempo (Segundo)	Base Própria %	Base 100 MVA PU
R	RESISTÊNCIA DO ESTATOR			
X	REATÂNCIA DE DISPERSÃO DO ESTATOR			
Xd	REATÂNCIA SÍNCRONA DE EIXO DIRETO			
X'd	REATÂNCIA TRANSITÓRIA DE EIXO DIRETO			
X"d	REATÂNCIA SUBTRANSITÓRIA DE EIXO DIRETO			
T'do	CONSTANTE DE TEMPO TRANS. DE EIXO DIRETO A VAZIO			
T"do	CONSTANTE DE TEMPO SUBTRANS. DE EIXO DIRETO A VAZIO			
Xq	REATÂNCIA SÍNCRONA DE EIXO EM QUADRATURA			
X'q	REATÂNCIA TRANSITÓRIA DE EIXO EM QUADRATURA			
X"q	REATÂNCIA SUBTRANSITÓRIA DE EIXO EM QUADRATURA			
T'qo	CONSTANTE DE TEMPO TRANS. DE EIXO EM QUAD. A VAZIO			
T"qo	CONSTANTE DE TEMPO SUBTRANS. DE EIXO EM QUAD. A VAZIO			
GD2	MOMENTO DE INÉRCIA (Kg x m2)			
H	CONSTANTE DE INÉRCIA (s)			

Observações:

Preparado por:	Data:	Folha:
Visto por:	Aprovado por:	00 de nn