

Técnicas de Análise de Sistema

Osmar de Oliveira Braz Junior - 1997.

Tubarão / Santa Catarina / BRAZIL

E-Mail : osmarjr@unisul.rct-sc.br

<http://tec1.unisul.rct-sc.br/osmarjr/>

Sumário

- 1. Conceitos Básicos.....**
 - 1.1 Análise.....*
 - 1.2 Processo.....*
 - 1.3 Programa.....*
 - 1.4 Análise de Sistemas.....*
 - 1.5 Sistemas.....*
 - 1.5.1 Tipos de Sistemas.....*

- 2. Análise Estruturada.....**
 - 2.1 Análise Tradicional.....*
 - 2.1.1 Segunda Geração.....*
 - 2.1.2 Terceira Geração.....*
 - 2.2 Relacionamento Usuário e Analista.....*
 - 2.3 Problemas com Análise Clássica(Tradicional).....*
 - 2.3.1 Comunicação.....*
 - 2.3.2 Mudanças naturais exigidas pelo sistema.....*
 - 2.3.3 Falta de Ferramentas.....*
 - 2.3.4 Documentação.....*
 - 2.3.5 Formação do Profissional.....*
 - 2.3.6 Dificuldade de Fixação do Problema.....*
 - 2.4 Análise Tradicional X Análise Estruturada.....*
 - 2.4.1 Comparação.....*
 - 2.4.2 Objetivos da Análise Estruturada.....*
 - 2.4.3 Condução do Trabalho de Análise.....*
 - 2.5 Diálogo Usuário X Analista.....*
 - 2.6 As Ferramentas da Análise Estruturada.....*
 - 2.6.1 Diagrama de Fluxo de Dados.....*
 - 2.6.2 Dicionários de Dados.....*
 - 2.6.3 Descrição de Procedimentos.....*

- 3. Análise de Dados.....**
 - 3.1 Conceitos Básicos.....*
 - 3.2 Modelo de Dados.....*
 - 3.2.1 O que é Modelo de Dados?.....*
 - 3.2.2 Componentes do Modelo de Dados.....*
 - 3.2.3 Tipos de Entidade.....*
 - 3.2.4 Tipos de Relacionamento.....*
 - 3.2.5 Tipos de Chave.....*
 - 3.3 Consolidação de Modelos de Dados.....*
 - 3.3.1 O que é Consolidação?.....*
 - 3.3.2 Trabalhos Executados na Consolidação.....*
 - 3.4 Normalização.....*
 - 3.4.1 O que é Normalização.....*
 - 3.4.2 Anomalias de Atualização.....*
 - 3.4.3 Dependência Funcional.....*
 - 3.4.4 Dependência Funcional Composta ou Completa.....*
 - 3.4.5 Dependência Funcional Transitiva.....*
 - 3.4.6 Primeira Forma Normal (1FN).....*
 - 3.4.7 Segunda Forma Normal (2FN).....*
 - 3.4.8 Terceira Forma Normal (3FN).....*

3.4.9 Simplificação do Processo de Normalização.....

3.4.10 Regras Práticas.....

3.4.11 Exercícios:.....

4. Bibliografia.....

1. Conceitos Básicos

1.1 *Análise*

Derivado do grego *analýein* - desatar, soltar, significa dissolução de um conjunto em suas partes. Em sentido amplo, empregam-se os termos “análise” e “analisar” como sinônimos de exame e examinar, pesquisa e pesquisar, verificação e verificar.

1.2 *Processo*

Série de fenômenos sucessivos com relação de causa e efeito; por exemplo, uma empresa é uma série de causas (matérias primas, recursos humanos, tecnologia, etc.) que geram um efeito (produtos).

1.3 *Programa*

Escrito em que se dão os pormenores de um espetáculo, de uma cerimônia, das condições de um concurso, dos procedimentos para execução de uma tarefa.

1.4 *Análise de Sistemas*

Representa o estudo detalhado de uma área de trabalho (processo), que antecede uma ação que, quase sempre, implica no desenvolvimento de um conjunto de programas integrados (sistema) destinado à execução controle e acompanhamento do processo.

1.5 *Sistemas*

Como veremos, existe uma definição “oficial” do termo sistema no dicionário, que parecerá bastante abstrata. Existem, porém, muitos usos comuns do termo que lhe parecerão perfeitamente familiares, e existem muitos tipos comuns de sistemas com que temos contato todos os dias.

É importante estar familiarizado com diferentes espécies de sistemas por pelo menos dois motivos. Primeiro, mesmo que seu trabalho como analista se concentre em um tipo de sistema - um sistema automatizado de informações, computadorizado - ele normalmente fará parte de um sistema maior. Desse modo, você pode estar trabalhando em um sistema de pagamentos, que é parte de um sistema maior de “Recursos Humanos”, que, por sua vez, é parte da organização comercial geral (que constitui um sistema), que é, por sua vez, componente de um sistema econômico geral, e assim por diante. Ou você pode estar trabalhando em um sistema de controle de processos que é parte de uma refinaria química, ou em um sistema operacional que seja parte de um “pacote” de software de sistemas distribuídos por vendedores. Assim, para que o seu sistema tenha sucesso, é preciso conhecer os outros sistemas com os quais ele vai interagir.

Muitos dos sistemas de computadores que elaboramos são substituições ou novas implementações de sistemas não-computadorizados que já existem; além disso, a maioria dos sistemas computadorizados interage ou tem uma interface com vários sistemas existentes (alguns podem ser computadorizados ou não). Para que nosso sistema computadorizado seja bem-sucedido, precisamos conhecer, detalhadamente, como o sistema atual se comporta.

Em segundo lugar, embora muitos tipos de sistemas pareçam ser totalmente diferentes, eles têm muitas semelhanças; existem princípios comuns, filosóficas e teorias que se aplicam notavelmente bem a virtualmente todos os tipos de sistemas. Assim, podemos muitas vezes aplicar o que aprendemos sobre outros sistemas - com base em nossa experiência diária, bem como na experiência de cientistas e engenheiros em diversas áreas - aos sistemas que elaboramos na área da computação. Por exemplo, um dos importantes princípios de sistemas que primeiro foi observado no campo da biologia é conhecido como a lei da especialização; quanto mais adaptado for um organismo a um determinado ambiente, mais difícil será para esse organismo a adaptação a outro. Isso ajuda a explicar o desaparecimento dos dinossauros quando o clima da Terra modificou-se radicalmente; ajuda, também, aos analistas de sistemas a compreenderem que se otimizarem um sistema computadorizado de forma a tirar a máxima vantagem de uma determinada UCP, de uma linguagem

de programação e de um sistema de gerenciamento de banco de dados, poderão vir a ter sérios problemas em adaptar o sistema a ser processado em outra UCP ou com um diferente sistema de gerenciamento de banco de dados.

Dessa maneira, se conhecermos alguma coisa da teoria geral dos sistemas, ela pode nos ajudar a compreender melhor os sistemas computadorizados (automatizados) de informações. Isso é cada dia mais importante, pois queremos construir sistemas estáveis e confiáveis, que funcionarão bem em nossa complexa sociedade - e há, naturalmente, muitos sistemas não-computadorizados que vêm sobrevivendo por milhões de anos: a humilde barata provavelmente sobreviverá a todos os sistemas computadorizados já construídos ou a construir, e a toda a humanidade, também.

Assim vamos começar com uma definição do termo básico sistema.

1. um grupo de itens que interagem entre si ou que sejam interdependentes, formando um todo unificado <~numérico> : como
 - a. (1) um grupo de corpos que interagem entre si sob a influência de forças relacionadas <~gravitacional>
(2) uma mistura de substâncias em equilíbrio ou que tende para o equilíbrio <~termodinâmico>
 - b. (1) um grupo de órgãos do corpo que desempenham, em conjunto, uma ou mais funções vitais < o ~digestivo >
(2) o corpo, considerando como uma unidade funcional.
 - c. um grupo de objetos ou forças naturais relacionadas entre si < um ~fluvial >
 - d. um grupo de dispositivos ou uma organização em rede, principalmente para a distribuição de algum produto ou servindo a um propósito comum < um ~ telefônico > < um ~ de aquecimento> < um ~ rodoviário > < um ~ de processamento de dados >
2. um conjunto organizado de doutrinas, idéias ou princípios, habitualmente previsto para explicar a organização ou funcionamento de um conjunto sistemático < o ~ da mecânica newtoniana >
3. a. um procedimento organizado ou estabelecido < o ~ de toques da digitação >
b. uma maneira de classificar, simbolizar ou esquematizar < um ~ taxonômico > < o ~ decimal >
4. organização ou modelo: ORDEM
5. sociedade organizada ou situação social vista como indesejável: "ESTABLISHMENT".

1.5.1 Tipos de Sistemas

1.5.1.1 *Sistemas Naturais*

- Sistemas Estelares (galáxias, sistemas solares, etc.)
- Sistemas Geológicos (rios, cadeias de montanhas etc.)
- Sistemas Moleculares (organizações complexas de átomos)

1.5.1.2 *Sistemas feitos pelo Homem*

- Sistemas Sociais(organizações de leis, doutrinas, costumes, etc.)
- Sistemas de Transporte (redes rodoviárias, canais, linhas aéreas, petroleiros, e semelhantes).
- Sistemas de Comunicação (Telefone, telex, sinais de fumaça, sinais manuais, etc.)

- Sistemas de Manufatura (Fábricas, linhas de montagem, etc.)
- Sistemas Financeiros (contabilidade, inventários, livros-razão, controle de estoque, entre outros)

1.5.1.3 *Sistemas Automatizados*

- Hardware de computadores - UCP, terminais, impressoras, unidades de fita magnéticas, etc.
- Software de computadores - programas de sistemas, como sistemas operacionais, sistemas de bancos de dados e programas de controle de telecomunicações, além dos programas aplicativos que executam as funções desejadas pelo usuário.
- Pessoas - aquelas que operam o sistema, que fornecem as entradas e utilizam as saídas, e as que desempenham atividades de processamento manual em um sistema.
- Dados - as informações que o sistema conserva por um período de tempo.
- Procedimentos - determinações e instruções formais para a operação do sistema.

2. Análise Estruturada

2.1 *Análise Tradicional*

2.1.1 Segunda Geração

Até 1965 os computadores de grande porte instalados em nosso país eram classificados como de segunda geração, como por exemplo o 1401-IBM.

Máximo no desenvolvimento de sistemas, era um sistema de folha de pagamento, e um sistema de controle de estoque.

- Folha de pagamento (20 a 24 horas) para classificação de 10 mil funcionários.
- Não existia formação profissional.
- Sem documentação.

2.1.2 Terceira Geração

- 1965, chegada do COBOL (considerada auto documentável).
- Aumento considerável no número de usuários em informática.
- Documentação era compreendida somente pelo profissional que desenvolveu.
- A documentação representava somente a parte física da aplicação.
- As lógicas não existiam em lugar nenhum.

O Software e o Hardware tem se desenvolvido de forma acentuada mas a documentação continua em muitos CPD's sem metodologia alguma, visando apenas como feito a aplicação(software), ou seja, uma documentação física, dedutiva, difícil manutenção e difícil entendimento.

2.2 *Relacionamento Usuário e Analista*

- Analista união entre os usuários e os projetistas.
- Conclusão da etapa de requisitos funcionais do sistema.
- O Analista responde pelo usuário a qualquer dúvida que o projetista vem a ter.
- Esta ferramenta, diminui possíveis dúvidas a serem levantadas durante a fase de projeto.

- É preciso definir bem as responsabilidades de cada um, O analista é responsável por: estudos de viabilidade e alternativas, custo/benefícios, especificações, prazos e teste de aceitação, enquanto o usuário é o recebedor final do sistema. Este é o responsável pela decisão de integração do

sistema dentro das operações da empresa, ou não. Somente ele, o usuário pode aceitar o sistema.

2.3 Problemas com Análise Clássica(Tradicional)

2.3.1 Comunicação

Formas de interpretação diferentes, gerando interpretações erradas, e que levada adiante continuarão a serem distorcidas cada vez mais.

- Uso excessivo de termos técnicos(AnalistaXUsuário).

2.3.2 Mudanças naturais exigidas pelo sistema

- Maior nas aplicações comerciais.
- Número discreto e portarias aplicados pelos governos federal e estadual durante os últimos anos.

2.3.3 Falta de Ferramentas

- Ferramenta antiquadas de 20 anos atrás.
- Utilizando a narrativa proporcionando
- Perda de tempo.
- +50% das informações deduzidas pelo profissional de informática.

2.3.4 Documentação

- As empresas não adotam um padrão.
- Existe a figura do "Pai do Sistema".
- Dificuldade de manter a documentação (o trabalho manuscrito)

2.3.5 Formação do Profissional

- Precária formação profissional na área de análise de sistemas.
- Adeptos da forma estruturada são submetidos a velha forma tradicional.

2.3.6 Dificuldade de Fixação do Problema

- Com textos narrativos na fase de levantamento das necessidades do usuário +70% das informações da documentação.
- Localização dos pontos a sofrerem alteração levam muito tempo, sem a certeza de todos os pontos foram alterados.

2.4 Análise Tradicional X Análise Estruturada

2.4.1 Comparação

Enquanto na versão clássica qualquer produto final só pode ser analisado numa única dimensão, na versão estruturada um sistema pode ser analisado na dimensão exata das necessidades, tanto do analista quanto do usuário. Tudo vai depender da visão que o interessado deseja ter do sistema, se mais abrangente ou mais detalhada.

A versão clássica é totalmente prolixa(muito longa ou difusa), enquanto que a estruturada apresenta e expõe o que é feito e o que vai ser feito através do uso de gráficos, o que torna a visualização e entendimento muito mais claros e objetivos.

A versão clássica entra diretamente em detalhes, pelo simples fato que o usuário pensa no computador como a fórmula mágica para a solução de todos os seus problemas. O trabalho de análise é dirigido às vezes até inconscientemente dessa forma. O levantamento é feito principalmente

a partir dos problemas apresentados pelo usuário, um a um, sem a preocupação do todo. Por último levantando aquilo que na concepção do usuário está bem, sendo que às vezes, o que ia bem, ao ser informatizado passa a ir mal, simplesmente por falta de preocupação dos envolvidos com o todo. Na versão estruturada isso não acontece, pois o trabalho de análise deve ser dirigido para a ferramenta e esta exige que a análise deve ser feita de cima para baixo através de refinamentos sucessivos até atingir-se os detalhes. Durante a parte de levantamento, não deve existir por parte dos envolvidos analistas/usuários qualquer preocupação com problemas ou erros existentes. Primeiro deve-se construir o modelo lógico existente para em seguida, após uma análise conjunta bastante criteriosa, identificarmos os problemas e propormos as devidas soluções.

Por último, a versão clássica gera um produto monolítico enquanto que a versão estruturada um totalmente particionado, do maior ao menor nível de detalhe, possibilitando a identificação clara e simples de qualquer parte do sistema, bem como a agregação em pequenos blocos de funções afins.

2.4.2 Objetivos da Análise Estruturada

O documento a ser padronizado deve ser:

- Passível de manutenção
- Gráfico
- Lógico
- Rigoroso
- Conciso
- Legível

Tudo isso deve ser um sub-produto natural do trabalho. Ou seja, terminada a fase de análise, ninguém deve necessitar de mais tempo para preparar a documentação - ela já deve estar concluída.

2.4.3 Condução do Trabalho de Análise

A condução da análise deve ser:

- Dirigida para a Ferramenta
- Mensurável/Pré-Determinada
- Divisível

É de vital importância o cuidado de dirigir a análise para a ferramenta, pois caso contrário estaremos praticando a versão clássica para, numa segunda etapa, dispor as informações de forma gráfica. Ou seja a análise deve ser feita de cima para baixo. A preocupação de levantar o que é feito pelo usuário deve ser constante e não, à medida em que o usuário fala, pensar em como o analista vai mecanizar aquilo, quais vão ser as estruturas dos arquivos físicos, quais serão os métodos de acesso e outras preocupações mais. Estas deverão ser objeto de preocupação de quem vai desenvolver o projeto físico e não dele, mesmo que ele venha a acumular essas funções.

2.5 Diálogo Usuário X Analista

O diálogo usuário/analista deve ser:

- Iterativo
- Lógico
- Limitado

2.6 As Ferramentas da Análise Estruturada

2.6.1 Diagrama de Fluxo de Dados

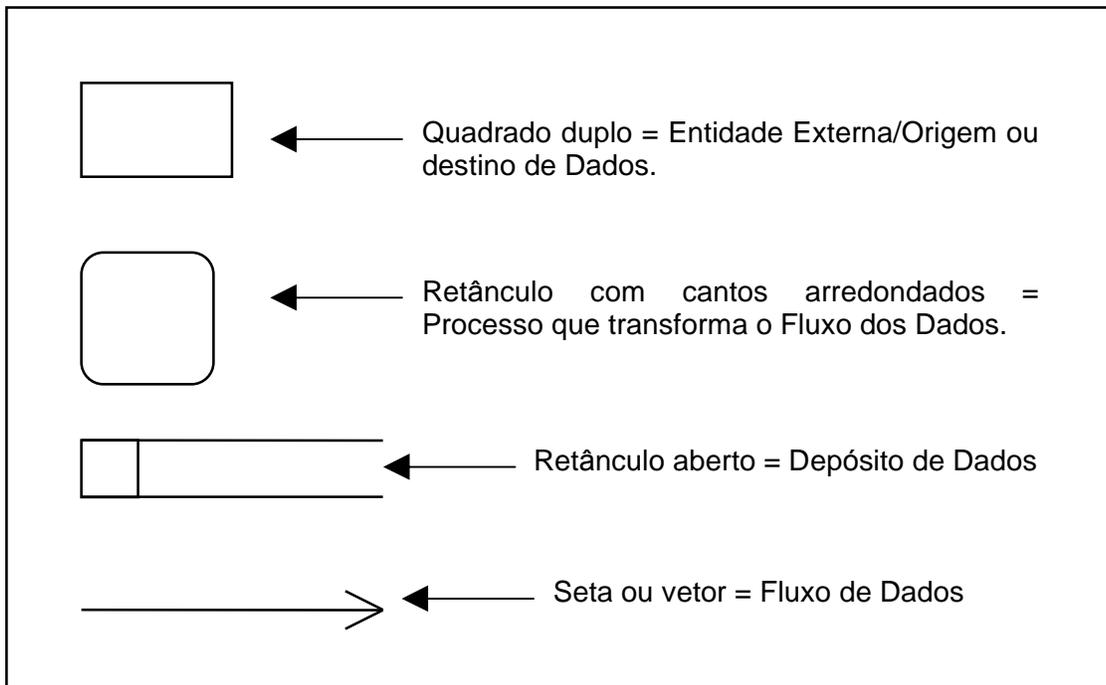
DFD é uma representação em rede dos processos (funções) do sistema e dos dados que ligam

esses processos. Ele mostra o que o sistema faz e não como é feito. É a ferramenta de demonstração central da análise estruturada.

Um DFD apresenta as partes componentes de um sistema e as interfaces entre elas. É um conjunto integrado de procedimentos, sendo que as partes do computados poderão estar inseridos ou não.

Na elaboração de um DFD, utilizaremos quatro símbolos que nos permitirão, debater e apresentar ao usuário todo o processo, sem assumir nenhum compromisso com implementações e demonstrar a sua fluência, sem a preocupação com a hierarquização e tomadas de decisão.

São os seguintes símbolos utilizados na elaboração de um DFD:



2.6.1.1 Elaborando um DFD

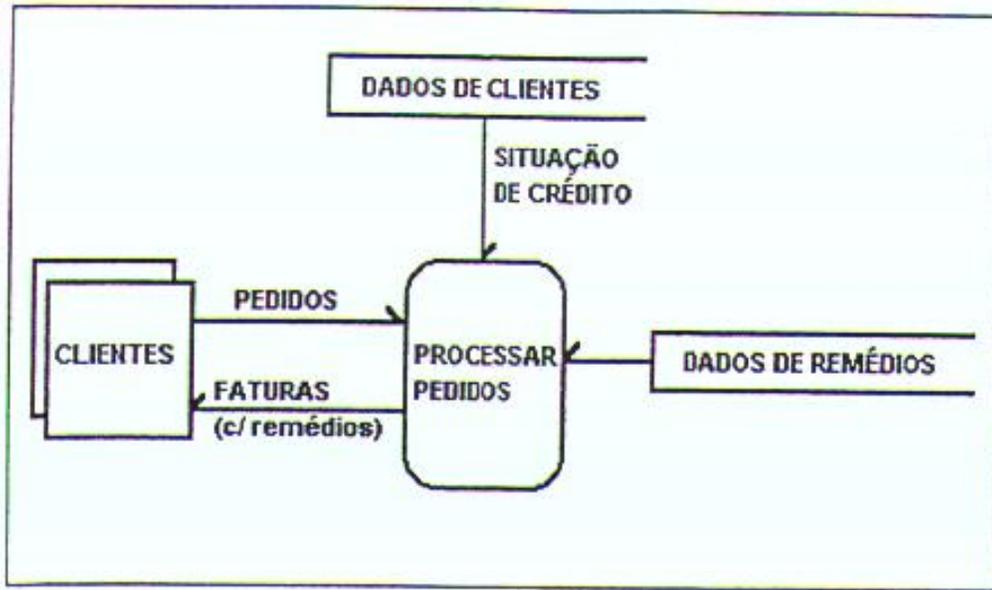
Suponhamos que uma distribuidora de produtos farmacêuticos nos contratou para analisar seu processo atual e verificar como expandir suas operações e melhorar seu nível de serviço.

A empresa em questão, RPC (Remédios Pelo Correio), fundada há cinco anos atua na distribuição de medicamentos, recebendo das farmácias os pedidos de medicamentos, fazendo encomenda aos laboratórios, com desconto, e atendendo ao pedido no ato do recebimento do dos remédios dos laboratórios. O processo é todo controlado manualmente através do preenchimento de formulários pré-impressos. Atualmente o volume de negócios atinge 150 pedidos por dia, cada um com um média de 5 medicamentos, e um valor de R\$ 500,00 em média. A administração pretende expandir as operações através da estocagem dos 100 medicamentos mais solicitados e atendendo solicitações de clínicas e médicos diretamente. As encomendas poderão ser feitas de qualquer ponto do Estado via telefone ou pelo correio.

O volume de negócios dependerá, logicamente, de fatores como divulgação do serviço, rapidez na entrega, confiabilidade, etc., mas a empresa espera aumentá-lo para 1000 negócios/dia, ou mais.

No plano geral, podemos afirmar que, da mesma forma que o atual, o novo processo de trabalho da empresa acatará pedidos de remédios, fará a verificação no arquivo de disponíveis, consultará, em outro arquivo, se o crédito do cliente é bom e fará com que o remédio solicitado seja encaminhado ao cliente com a respectiva fatura.

Demonstraremos isso de forma gráfica usando um diagrama de Fluxo de dados lógico.

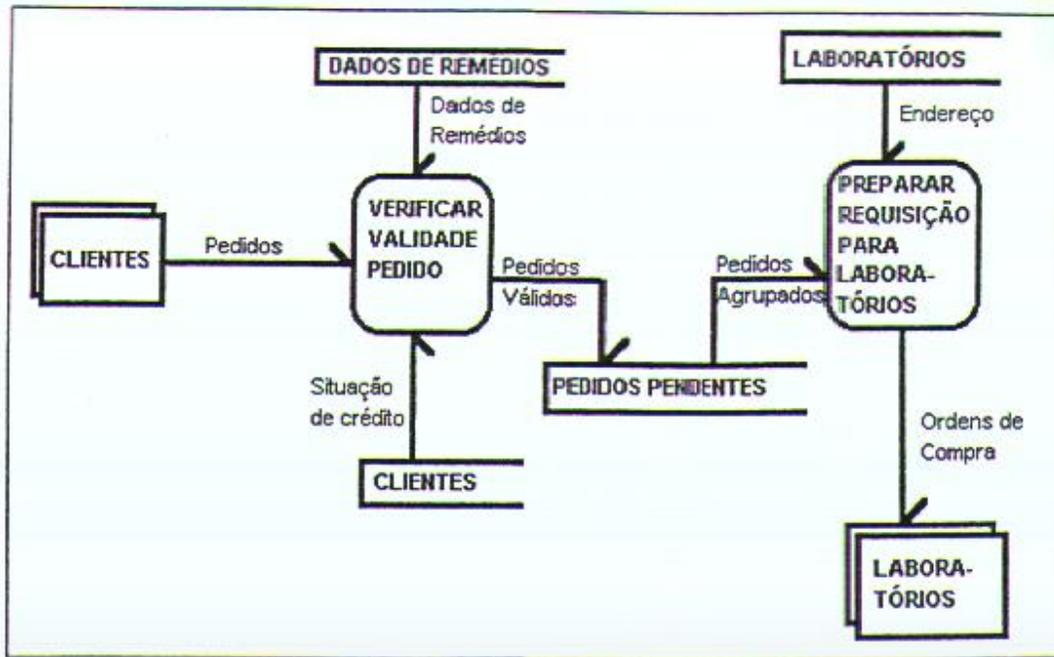


Analisando a figura, verificamos que, na verdade, ela nos diz muito pouco sobre o sistema.

Os símbolos constantes da figura e os conceitos que representam encontram-se no nível lógico; um fluxo de dados pode estar fisicamente numa carta, numa fatura, numa ligação telefônica, etc., ou seja, em qualquer lugar em que o dado passe de uma entidade ou processo para outro. Um processo pode ser fisicamente um escritório repleto de pessoas verificando e recebendo pedidos, calculando descontos, ou um programa, ou ainda uma combinação de atividades manuais e automatizadas. Um depósito de dados pode ser um armário de aço com gavetas, um fichário de cartões, uma fita magnética, um disquete. Utilizando os quatro símbolos, podemos desenhar um quadro do sistema sem nos comprometermos com sua implementação.

Vamos expandir “processar pedidos” para mostrar as funções lógicas que compõe o processo.

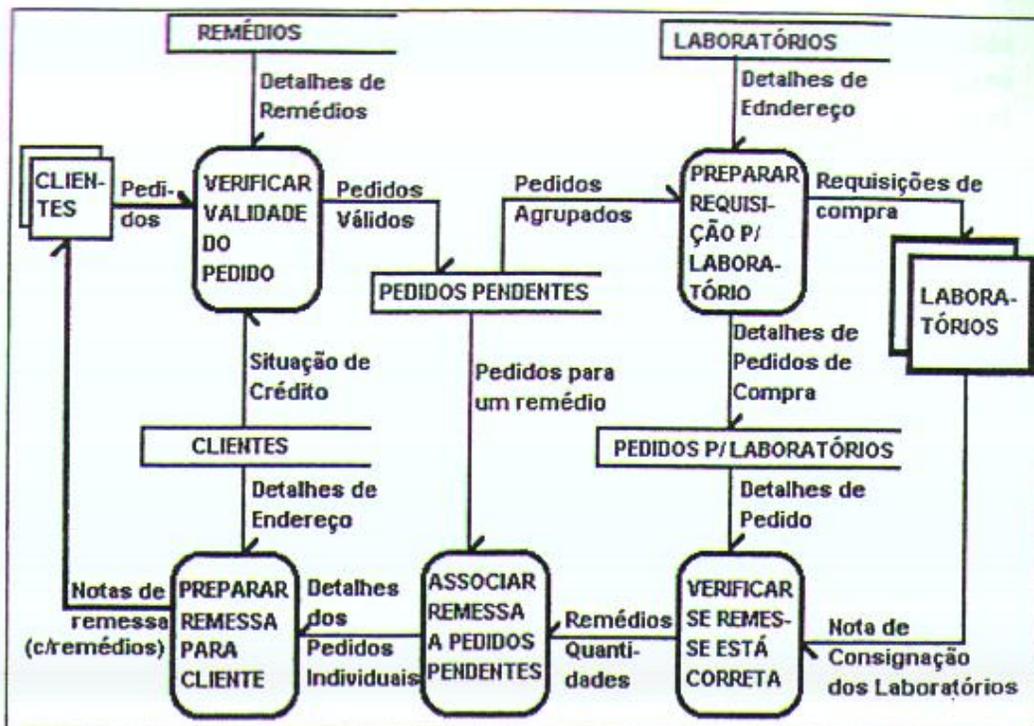
Observe o diagrama a seguir, onde representamos uma expansão do anterior, demonstrando os processos “Verificar validade dos pedidos” e “Preparar requisição par o laboratório”, além de depósitos de dados para armazenar dados de clientes, dados de laboratórios e dados dos pedidos pendentes, ou seja, aqueles que ficam aguardando a quantidade ótima para endereçarmos o pedido ao laboratório obtendo o maior desconto.



Até aqui, parece tudo bem. Mas será que vamos atender os pedidos e esperar pacientemente que o pagamento seja efetuado? E os laboratórios fornecedores não irão cobrar nunca os medicamentos remetidos? E se os medicamentos e quantidades remetidas pelos laboratórios não forem coerentes com as solicitações?

Vamos tentar incluir o aspecto “Comparar remessa a pedidos”.

Observe o próximo diagrama.



Não demonstraremos até aqui os movimentos dos remédios em si; para efeitos didáticos, os remédios são considerados dados e por isso não são representados no DFD. A relação entre um DFD e um diagrama de fluxo de materiais não será abordada por enquanto. Atualmente só nos interessam os itens que representam dados sobre remédios.

Até agora, ninguém recebeu nenhum pagamento. Devemos nos preocupar com a remessa de faturas para os clientes, tratamento a ser aplicado aos pagamentos efetuados pelos clientes, bem como cobranças efetuadas pelos laboratórios.

Acreditamos que, com o que já foi visto até aqui, você seria capaz, sozinho, de expandir nosso DFD, contemplando esses processos.

Não se esqueça que cada uma das caixas de processo pode ser expandida num diagrama de fluxo de dados de menor nível, assim sendo, procure, ao fazer o exercício proposto, não descer a detalhes muito minuciosos. Sua preocupação deve ser demonstrar em linhas gerais como seriam os processos de contas a receber e contas a pagar.

Outro aspecto importante, não abordado nos DFD's apresentados são as condições de erro.

Não especificamos ainda o que acontece com o pedido de um cliente cuja situação de crédito seja ruim, ou o que acontece quando o laboratório nos manda uma remessa e não localizamos nenhum pedido correspondente.

É evidente que tais situações precisam ser tratadas. Entretanto, se formos, desde logo, nos prender ao tratamento de erros e exceções, comprometeremos todo o nosso trabalho. O detalhamento dessas questões deve ser adiado para os diagramas de nível inferior, para que não interfiram no quadro geral do sistema.

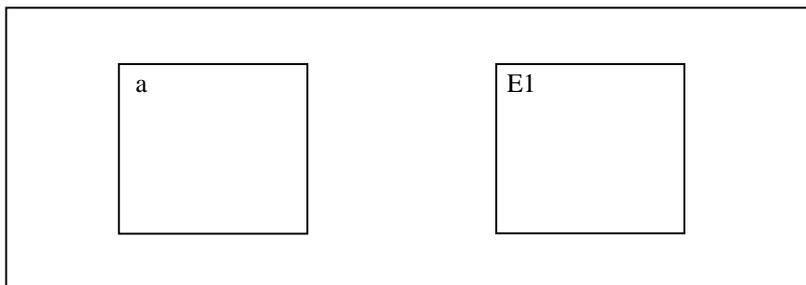
A conclusão dos DFD's do sistema proposto, com toda a abrangência, fica a cargo de vocês, basta aplicar os recursos até aqui apresentados, observando entretanto as seguintes convenções simbólicas:

2.6.1.2 Entidade

Identificamos como entidade, na maioria das vezes, categorias lógicas de coisas ou pessoas que representam uma origem ou destino de transações (Clientes, Fornecedores, Empregados, Etc.). Também podemos identificar como Entidades fontes ou destinos específicos tais como Departamentos da empresa, Receita Federal, Almoxarifado. É comum adotarmos a terminologia Entidade Externa. Quando um sistema recebe dados resultantes de outro, ou gera informações que servirão como dados de entrada para outro, esse outro sistema também é identificado como uma Entidade Externa.

O símbolo utilizado para representar já foi apresentado a você.

Por convenção, a fim de simplificar as referências e o processo de "dicionarização" dos dados, adicionamos como identificador de uma entidade uma letra minúscula no canto superior esquerdo do desenho ou a letra E maiúscula e um número, conforme abaixo:

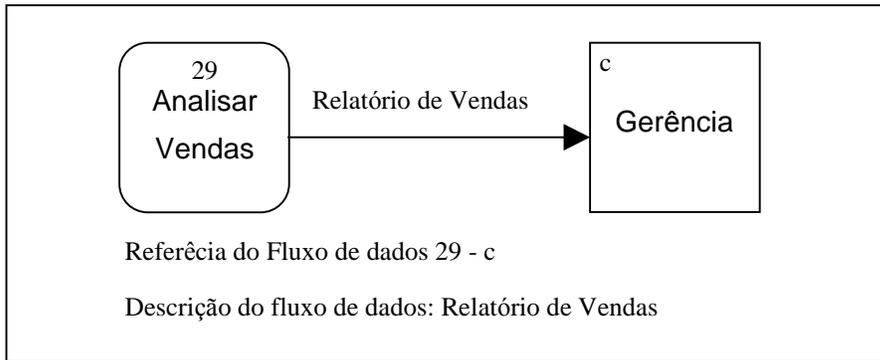


2.6.1.3 Fluxo de Dados

Podemos associar cada fluxo de dados com um tubo por onde passam pacotes de dados. Faremos referência ao Fluxo de Dados identificando os processos, entidades ou depósitos de dados das suas extremidades, anotando uma descrição do seu conteúdo ao longo de sua extensão. Lembre-

se que a descrição deve ser mais clara possível, de modo a simplificar o trabalho do usuário que irá realizar a revisão do DFD.

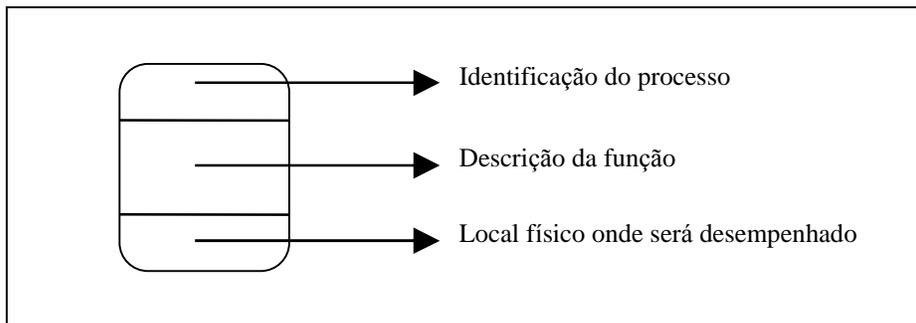
Observe um exemplo de referência e descrição de Fluxo de Dados:



2.6.1.4 Processo

Logicamente, é necessário descrever a função de cada processo, e, para facilitar atribuir uma identificação única para cada um, buscando, na medida do possível, associá-lo a um sistema físico.

A identificação pode ser um número, inicialmente posicionado na posição média superior da figura, não tendo nenhum outro significado além de identificar o processo. Não há porquê vincularmos a identificação com a descrição do processo, pois alguns deles serão subdivididos em dois ou mais nas fases de expansão - o que implicará no surgimento de novos números. Entretanto, a partir do instante que um processo recebe uma identificação, esta não deve mais ser modificada, sob a pena de comprometer o trabalho de “dicionarização” dos dados, exceto nos casos de desmembramentos e agrupamentos. Para simplificar o entendimento da figura, podemos adicionar linhas divisórias, marcando claramente o espaço destinado à identificação do processo, sua descrição e o local físico onde será desempenhado.

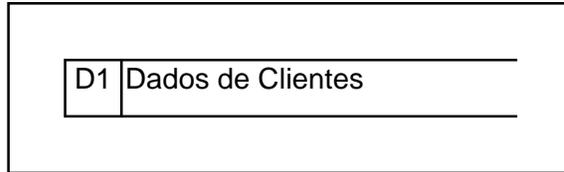


Vale ressaltar que a descrição da função deve ser sempre imperativa, composta por um verbo ativo (verificar, extrair, recuperar, comparar), seguida de uma cláusula, simples e objetiva.

A identificação do local físico onde a função será executada, opcional nos diagramas de nível mais abrangente, é extremamente útil a partir do instante em que a análise foi concluída e o projeto físico do sistema está sendo desenvolvido, pois denota o departamento ou programa que o desempenhará

2.6.1.5 Depósito de Dados

Convencionamos a identificação de um depósito de dados pela colocação de uma letra “D” maiúscula seguida de um número, na esquerda do desenho, separada da descrição por uma linha vertical.



2.6.2 Dicionários de Dados

O dicionário de dados é um repositório de dados sobre os dados do software. Ele deverá conter a definição dos elementos que tornam o Modelo de Dados e o Diagrama de Fluxo de Dados precisos, quais sejam:

- Fluxos de dados;
- Depósitos de Dados/Entidades;
- Atributos.

2.6.2.1 Regras para Formação de Nomes

- O nome deve ser formado por palavras separadas por sublinha até o máximo de 32 caracteres;
- Preferencialmente a nomeação deve ser feita de acordo com o usuário;
- Devem ser eliminados proposições e conjunções;
- Quando houver necessidade de abreviar uma palavra, observar que a abreviatura seja clara, ou inclui-la no dicionário.

2.6.2.2 Notação

Símbolo	Significado
=	É composto de
+	E
[]	Escolha uma das opções alternativas
{ }	interações de
()	Opcional (pode estar presente ou ausente)
	separa opções alternativas na construção []
**	Comentário
@	Identificador (campo chave) de um depósito.

2.6.2.2.1 Sendo Dado Elementar (Atributo):

- Com seleção de valores:

NOME_ITEM = ["valor 1" | "valor n"] ft,d onde:

f = formato (A - Alfabético, N - Numérico, X - Caracter Válido);

t = tamanho (em caracteres ou inteiros);

d = decimais.

Ex.: SEXO=["M"|"F"] A1 (Alfabético de tamanho 1, podendo assumir valores M ou F).

- Sem seleção de valores

NOME_ITEM = ft,d

Ex.: VALOR_REPRESENTACAO = N7,2 (Numérico com 7 inteiros e 2 decimais)

2.6.2.2.2 Sendo os limites de interação conhecidos:

NOME_ITEM_GRUPO = Y (nome_item_elementar_1 + nome_item_elementa_n) X

onde : X = limite superior, Y = limite inferior.

Ex.: REFERENCIAS = 2 (NOME + ENDERENCO) 10

2.6.2.2.3 Exemplos Gerais

- Fluxo de Dados

DETALHE_CUSTO = DESCRICAO_COMPLETA + MATRIZ_CUSTO

A definição demonstra que o fluxo de dados é composto de outros fluxos de dados e/ou dados elementares.

DESCRICAO_COMPLETA = NOME_ITEM + TIPO_ITEM + [PESO | VOLUME]

A definição demonstra que o fluxo é composto por dados elementares e um dos dados deve ser selecionado entre o peso ou o volume.

- Depósito de Dados/Entidades

ARQUIVO_DE_CUSTO = 1 (@NUMERO_ITEM + DETALHE_CUSTO) 1000

Os registros de arquivos vão de 1 a 1000 e o atributo NUMERO_ITEM é campo chave.

- Dados Elementares/Atributos

TIPO_ITEM = ["SOLIDO" | "LIQUIDO" | "GASOSO"] A7

2.6.3 Descrição de Procedimentos

O DFD declara a existência dos procedimentos e das interfaces entre elas, mas e o seu conteúdo? Como alternativas para a descrição de procedimentos podemos ter:

- TEXTO NARRATIVO;
- PORTUGUÊS ESTRUTURADO;
- TABELAS DE DECISÃO;
- ÁRVORES DE DECISÃO;
- FÓRMULAS MATEMÁTICAS;
- ALGUMAS COMBINAÇÕES DOS ACIMA.

2.6.3.1 *Texto Narrativo*

Exemplo:

O cálculo da depreciação deverá ser efetuado da seguinte forma:

- Quando o VALOR_CAPITAL_ATUAL for inferior a R\$ 1.000,00 e a CATEGORIA do bem sobre o qual vai ser calculada a depreciação for igual a "X", o valor da DEPRECIACAO deverá ser igual ao VALOR_CAPITAL_ATUAL e o VALOR_NOVO_CAPITAL deverá ser reduzido a zeros.

- Quando o VALOR_CAPITAL_ATUAL for inferior a R\$ 1.000,00 e a CATEGORIA do bem sobre o qual vai ser calculada a depreciação for igual a "Y", o valor da DEPRECIACAO deverá ser 20% do VALOR_CAPITAL_ATUAL e o VALOR_NOVO_CAPITAL deverá ser 80% do VALOR_CAPITAL_ATUAL.

- Quando o VALOR_CAPITAL_ATUAL for superior a R\$ 1.000,00 independentemente do valor da CATEGORIA, se "X" ou "Y", o valor da DEPRECIACAO deverá ser igual 35% do VALOR_CAPITAL_ATUAL e o VALOR_NOVO_CAPITAL deverá ser 65% do VALOR_CAPITAL_ATUAL.

- Quando o VALOR_CAPITAL_ATUAL for igual a R\$ 1.000,00 preceder de conformidade com o item anterior.

2.6.3.2 *Português Estruturado*

Exemplo:

Se VALOR_CAPITAL_ATUAL <= 1.000

Se CATEGORIA = X

Faça DEPRECIACAO = VALOR_CAPITAL_ATUAL

Faça VALOR_NOVO_CAPITAL = 0

Caso Contrario (Categoria = Y)

Faça $DEPRECIACAO = VALOR_CAPITAL_ATUAL * 0,20$

Faça $VALOR_NOVO_CAPITAL = VALOR_CAPITAL_ATUAL * 0,80$

Caso Contrário ($VALOR_CAPITAL_ATUAL \geq 1.000$)

Faça $DEPRECIACAO = VALOR_CAPITAL_ATUAL * 0,35$

Faça $VALOR_NOVO_CAPITAL = VALOR_CAPITAL_ATUAL * 0,65$

2.6.3.3 Tabelas de Decisão

Exemplo:

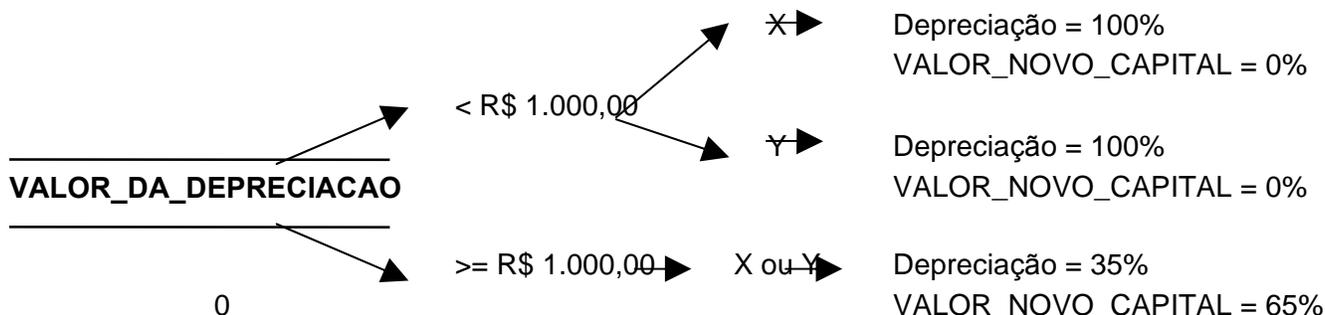
Condições	1	2	3	4
1. Categoria (X,Y)	X	Y	X	Y
2. VALOR_CAPITAL_ATUAL = R\$ 1.000,00 (<, >=)				
Ações				
1. Faça Depreciação igual a (% do VALOR_CAPITAL_ATUAL)	100	20	35	35
2. Faça VALOR_NOVO_CAPITAL igual a (% do VALOR_CAPITAL_ATUAL)	0	80	65	65

2.6.3.4 Árvores de Decisão

Exemplo:

VALOR_CAPITAL_ATUAL

CATEGORIA PROCEDIMENTO



3. Análise de Dados

Objetiva a definição e a estruturação de dados para servir às necessidades atuais e futuras de armazenamento e recuperação de informações.

3.1 Conceitos Básicos

- Entidades: Objeto existente na organização sobre o qual precisamos armazenar e/ou recuperar informações.

- Atributos: São as características que descrevem as entidades (informações que vão ser armazenadas sobre as entidades).

- Relacionamento: Associação entre duas ou mais entidades.

- Modelo de Dados: Representação gráfica das entidades e relacionamentos de interesse.

* Modelo Conceitual: Representa a visão dos dados sob o ponto de vista da organização.

* Modelo Lógico: Representa a visão dos dados sob o ponto de vista particular de um software.

* Modelo Físico: Representa a organização física da base de dados.

- Lista de Entidades: Lista das entidades e seus respectivos atributos.

- Identificador de Entidade: Atributo ou conjunto de atributos que identifica cada única ocorrência da entidade (chave primária).

- Normalização: Técnica de aplicação das Formas Normais às entidades de um modelo de dados para maior compreensão e simplificação da estrutura de dados.

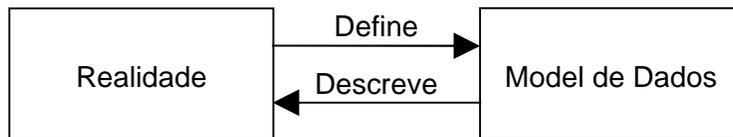
- Dependência Funcional: Quando em um relacionamento entre atributos de uma entidade, verifica-se que a cada valor de um atributo A corresponde um único valor de um ou mais atributos B, C, ..., dizemos que B, C, ... dependem funcionalmente de A.

- Consolidação de Modelos de Dados: Regras práticas para combinar diferentes modelos de dados em um único modelo.

3.2 Modelo de Dados

3.2.1 O que é Modelo de Dados?

Também conhecido como Diagrama E-R (Entidade -Relacionamento). É uma forma de representação gráfica do conhecimento que se tem sobre o ambiente (realidade) qualquer. Mostra uma visão estática das informações (entidades) de interesse e dos vínculos (relacionamentos) existentes entre elas.



O modelo de dados é uma nova forma de comunicação entre o técnico de processamento de dados e o usuário.

Essa nova forma de comunicação assegurará que :

- O modelo de dados conterá todos os dados necessários para suportar os processos de responsabilidade do usuário.

- O modelo de dados conterá os dados para suportar processos que serão modificados ou introduzidos em um futuro próximo.

3.2.2 Componentes do Modelo de Dados

3.2.2.1 Entidade

É algo, real ou abstrato, percebido no ambiente e sobre o qual nos interessa armazenar dados.

Exemplos:

Um objeto real (concreto)	- Um equipamento, Material
Uma pessoa	- Fornecedor Empregado
Um conceito abstrato	- Órgão, Cargo, Curso
Um evento	- Recebimento de Material
Um relacionamento	- Casamento

3.2.2.1.1 Representação Gráfica

Um entidade é representada num modelo de dados através de um retângulo.



3.2.2.2 Atributo

É um dos itens de dados que armazenamos sobre uma entidade. Caracteriza ou qualifica uma determinada propriedade de uma entidade.

Exemplo:

São atributos da entidade EMPREGADO:

- MATRICULA
- NOME
- ENDERECO
- CPF
- DATA NASCIMENTO

3.2.2.3 Chave de Identificação

A chave de identificação de uma entidade é definida por um atributo, ou conjunto de atributos, cujos valores individualizam uma única ocorrência dessa entidade.

Exemplo:

A chave de identificação da entidade EMPREGADO é o atributo MATRICULA.

3.2.2.4 Lista de Entidades

É uma relação de entidades com seus respectivos atributos, utilizada para documentar os trabalhos de análise de dados.

Formada pelo nome da entidade seguida da relação de atributos que compõem entre parênteses, e seguindo a convenção abaixo:

- Cada atributo é separado do outro pelo sinal de adição (+) ;

- O(s) atributo(s) que identificam a entidade devem estar no início da relação e sublinhados;
- O(s) atributo(s) que ocorrem mais de uma vez (repetitivos) são identificados por uma inclusão entre parênteses.

Exemplo:

FATURA(NUMERO_FATURA + CODIGO_FATURA +
(NUMERO_ITEM_FATURA + CODIGO_MATERIAL +
QUANTIDADE_MATERIAL + PRECO_UNITARIO +
PRECO_ITEM_FATURA) + PRECO_TOTAL_FATURA).

Obs.: Podem haver múltiplos níveis de repetição.

3.2.2.5 *Domínio*

São os possíveis valores que um atributo pode assumir.

Exemplo:

SEXO = [M | F]

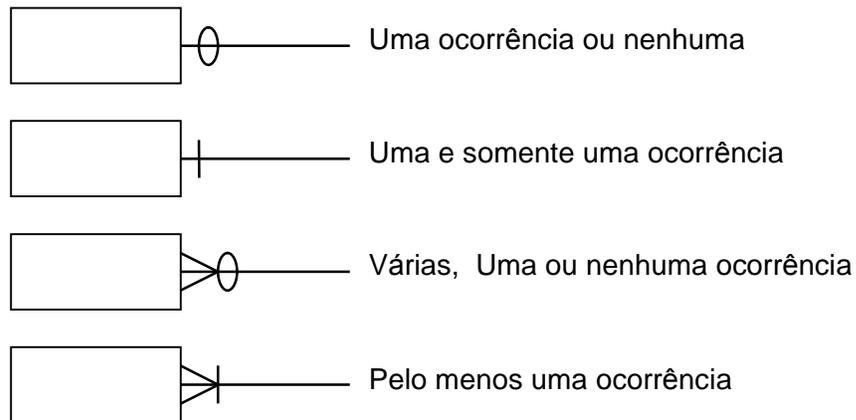
Sexo pode assumir dos valores M (Masculino) ou F (Feminino)

3.2.2.6 *Ocorrência*

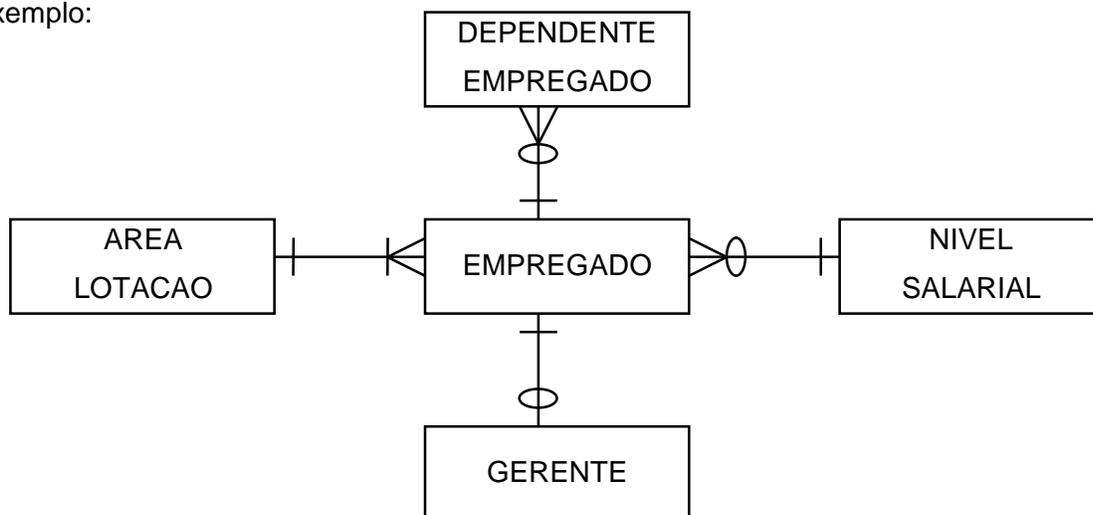
Representa o número vezes que determinado atributo aparece em outra entidade.

3.2.2.6.1 Representação Gráfica

Símbolos especiais colocados nas extremidades da linha que representa um relacionamento.



Exemplo:



- Uma ÁREA LOTAÇÃO tem obrigatoriamente pelo menos 1 empregado;
- Um EMPREGADO está vinculado obrigatoriamente a uma área de LOTAÇÃO;
- Um EMPREGADO pode ter vários, um ou nenhum DEPENDENTE;
- Um DEPENDENTE (se existir) está obrigatoriamente vinculado a um EMPREGADO.
- Um EMPREGADO pode ser GERENTE.
- Um GERENTE é um EMPREGADO
- Um EMPREGADO tem obrigatoriamente um NÍVEL SALARIAL;
- Em um mesmo NÍVEL SALARIAL podemos ter vários, um ou nenhum EMPREGADO.

3.2.3 Tipos de Entidade

3.2.3.1 Entidade Primária

É aquela cuja chave de identificação é feita exclusivamente através de seus atributos.

3.2.3.2 Entidade Dependente

É aquela cuja existência depende de outra, ou seja, parte da chave de identificação da entidade está condicionada a da entidade da qual ela depende.

3.2.3.3 Entidade Associativa

É aquela cuja chave de identificação é obtida através da concatenação das chaves de identificação das entidades que ela associa.

3.2.4 Tipos de Relacionamento

3.2.4.1 Relacionamento de Dependência

É aquele feito entre uma entidade e outra que dela seja dependente.

3.2.4.2 Relacionamento Associativo

É aquele que ocorre entre uma entidade associativa e a cada uma das entidades que participam da associação.

3.2.4.3 Categoria

Uma categoria é uma ligação entre uma entidade e suas espécies (tipos), sendo estas mutuamente excludentes.

3.2.4.4 Partição

É um caso particular de categoria, na qual as espécies (tipos) de uma entidade, não são mutuamente excludentes.

3.2.4.5 Relacionamento Normal

É aquele que não pode ser enquadrado em um dos tipos abaixo:

- Associativo
- Dependência
- Categoria
- Partição

3.2.4.6 Auto-Relacionamento

É aquele que ocorre entre uma mesma entidade.

3.2.4.7 Múltiplos Relacionamentos

Casos em que ocorre mais de um relacionamento entre duas mesmas entidades.

3.2.4.8 Relacionamento Mutuamente Exclusivo

Ocorre quando temos um relacionamento, por exemplo, entre as entidades "A" e "C" e também entre as entidades "B" e "C", porém nunca ao mesmo tempo.

3.2.5 Tipos de Chave

3.2.5.1 Chaves Candidatas

São as possíveis chaves de identificação de uma única ocorrência de uma entidade.

Exemplo

EMPREGADO (MATRICULA + NOME + CPF + ENDERENCO + SALARIO)

São chaves candidatas os atributos:

- MATRICULA
- CPF

3.2.5.2 Chave Primária

É uma das chaves candidatas, selecionada por melhor conveniência (facilidade de utilização, menor possibilidade de erros, etc. ...).

Exemplo

EMPREGADO (MATRICULA + NOME + CPF + ENDERENCO + SALARIO)

Chaves candidatas:

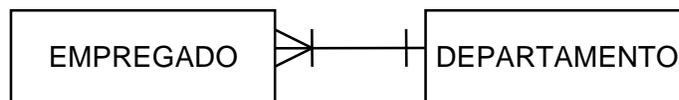
- MATRICULA
- CPF

Chave Primária escolhida - MATRICULA

3.2.5.3 Chave Estrangeira

Conjunto de um ou mais atributos de uma entidade que são chave primária em outra entidade.

Exemplo:



EMPREGADO (MATRICULA + NOME + CPF + COD_DEPTO)

DEPARTAMENTO (COD_DEPTO + NOME_DEPTO)

Na entidade EMPREGADO o atributo “COD_DEPTO” é chave estrangeira.

3.3 Consolidação de Modelos de Dados

3.3.1 O que é Consolidação?

Termo utilizado para representar os trabalhos de integração de um modelo de dados a outro ou, integração de modelos parciais a um modelo global de dados (empresa, assunto ou sistema).

3.3.2 Trabalhos Executados na Consolidação

Os Trabalhos da consolidação basicamente são os seguintes:

- Adição de entidades ainda inexistentes no modelo global de dados, relacionando-as às demais;
- Adição de novos atributos a entidades já existentes, desde que possuam chaves primárias idênticas;
- Identificação das entidades já implementadas;
- Eliminação de relacionamentos redundantes.

3.4 Normalização

3.4.1 O que é Normalização

Normalização é o processo formal que consiste em substituir um conjunto de entidades por outro conjunto capaz de comportar melhor as mudanças futuras.

Entidades normalizadas não possuem redundâncias (duplicação de dados) acidental. Cada atributo está relacionado com sua própria entidade e não se mistura com atributos relativos à entidades diferentes.

A normalização corresponde na realidade à formalização de regras baseadas no fato que as entidades possuem anomalias de atualização.

3.4.2 Anomalias de Atualização

Dada a entidade abaixo:

PEDIDO (NUMERO_PEDIDO + DATA_PEDIDO + NUMERO_CLIENTE +
NOME_CLIENTE + ENDERENCO_CLIENTE + (NUMERO_PRODUTO +
NOME_PRODUTO + QTDE_PEDIDA + PRECO_PRODUTO +
TOTAL_PRODUTO) + TOTAL_PEDIDO)

Quais as anomalias de atualização que acontecerão se:

- Um produto for descontinuado por seu fornecedor?
- O nome do produto for mudado?
- O cliente mudar de endereço?
- Os produtos ou as quantidades pedidas pelo cliente forem mudadas e o cliente esqueceu o número do pedido?

3.4.3 Dependência Funcional

Dados os atributos "A" e "B" de uma entidade, diz-se que "B" é funcionalmente dependente de "A" se e somente se, a cada valor de "A" está associado um único valor de "B".

Em outras palavras, se conhecermos o valor de "A" então podemos encontrar o valor de "B" associado a ele.

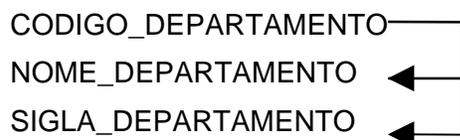
DIAGRAMA DE DEPENDÊNCIA FUNCIONAL



Nota - a seta parte de quem identifica.

Exemplo:

DEPARTAMENTO

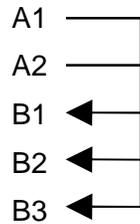


Nota - O exame das relações existentes entre os atributos de uma entidade deve ser feito a partir do conhecimento (conceitual) que se tem sobre o mundo real (ambiente modelado).

3.4.4 Dependência Funcional Composta ou Completa

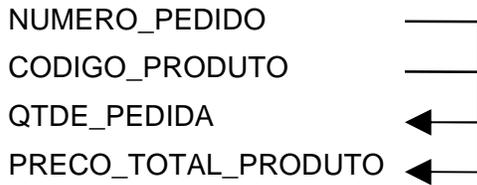
Dado um atributo ou um conjunto de atributos "B" de uma entidade, sendo a chave primária composta por um conjunto de atributos "A", diz-se que "B" é completamente dependente funcional da chave primária, se e somente se, a cada valor da chave (e não a parte dele), está associado um valor para cada atributo do conjunto "B".

DIAGRAMA DE DEPENDÊNCIA FUNCIONAL



Exemplo:

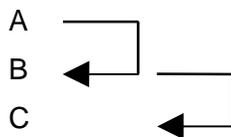
PRODUTO_FATURA



3.4.5 Dependência Funcional Transitiva

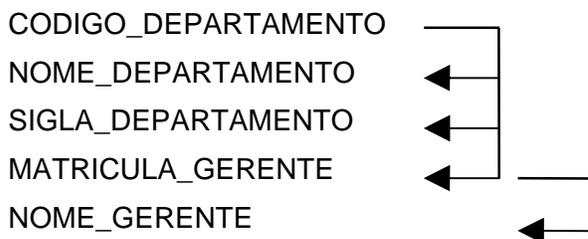
Dados os atributos “A”, “B” e “C” de uma entidade, sendo “A” a chave primária, diz-se que “B” e “C” são dependentes transitivos se e somente se, forem funcionalmente dependente de “A” além de existir uma dependência funcional entre eles.

DIAGRAMA DE DEPENDÊNCIA FUNCIONAL



Exemplo:

DEPARTAMENTO



3.4.6 Primeira Forma Normal (1FN)

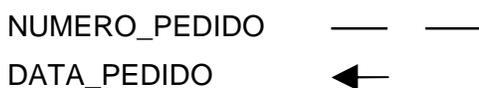
Uma entidade está na 1FN se ela não contém grupos de atributos repetitivos (multivalorados).

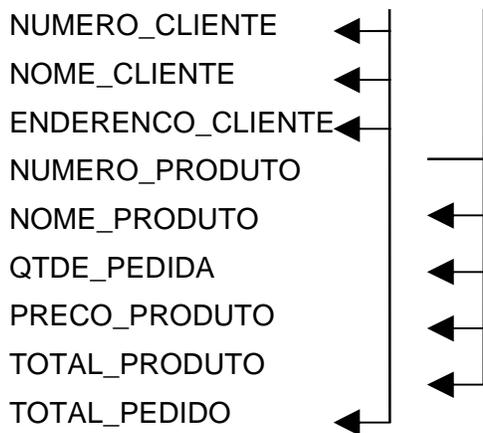
Exemplo:

- Entidade não normalizada:

PEDIDO (NUMERO_PEDIDO + DATA_PEDIDO + NUMERO_CLIENTE +
NOME_CLIENTE + ENDERENCO_CLIENTE + (NUMERO_PRODUTO +
NOME_PRODUTO + QTDE_PEDIDA + PRECO_PRODUTO +
TOTAL_PRODUTO) + TOTAL_PEDIDO)

- Remoção dos grupos de atributos repetitivos (1FN):



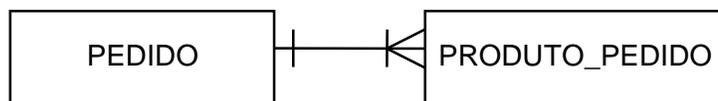


- Entidades da 1FN

PEDIDO (NUMERO_PEDIDO + DATA_PEDIDO + NUMERO_CLIENTE + NOME_CLIENTE + ENDERENCO_CLIENTE + TOTAL_PEDIDO)

PRODUTO_PEDIDO (NUMERO_PEDIDO + NUMERO_PRODUTO + NOME_PRODUTO + QTDE_PEDIDA + PRECO_PRODUTO + TOTAL_PRODUTO)

- Modelo de Dados



3.4.7 Segunda Forma Normal (2FN)

Uma entidade está na 2FN se ela está na 1FN e seus atributos são funcionalmente dependentes de sua chave (primária) completa.

Exemplo:

- Entidades da 1FN

PEDIDO (NUMERO_PEDIDO + DATA_PEDIDO + NUMERO_CLIENTE + NOME_CLIENTE + ENDERENCO_CLIENTE + TOTAL_PEDIDO)

PRODUTO_PEDIDO (NUMERO_PEDIDO + NUMERO_PRODUTO + NOME_PRODUTO + QTDE_PEDIDA + PRECO_PRODUTO + TOTAL_PRODUTO)

- Remoção dos atributos não funcionalmente dependentes de toda uma chave primária (2FN):

PEDIDO

NUMERO_PEDIDO —
 DATA_PEDIDO ←

PRODUTO_PEDIDO

NUMERO_PEDIDO —
 NUMERO_PRODUTO —

NUMERO_CLIENTE ←
 NOME_CLIENTE ←
 ENDERENCO_CLIENTE ←
 TOTAL_PEDIDO ←

NOME_PRODUTO ←
 QTDE_PEDIDA ←
 PRECO_PRODUTO ←
 TOTAL_PRODUTO ←

- Entidade na 2FN:

PEDIDO (NUMERO_PEDIDO + DATA_PEDIDO + NUMERO_CLIENTE +
 NOME_CLIENTE + ENDERENCO_CLIENTE + TOTAL_PEDIDO)

PRODUTO_PEDIDO (NUMERO_PEDIDO + NUMERO_PRODUTO + QTDE_PEDIDA +
 TOTAL_PRODUTO)

PRODUTO (NUMERO_PRODUTO + NOME_PRODUTO + PRECO_PRODUTO)

- Modelo de Dados:



3.4.8 Terceira Forma Normal (3FN)

Uma entidade está na 3FN se ela está na 2FN e não possui dependências transitivas.

Uma entidade que está na 2FN pode ter um atributo que não é uma chave mas que por si identifica outros atributos. Refere-se a isto como uma dependência transitiva.

Exemplo:

- Entidade na 2FN:

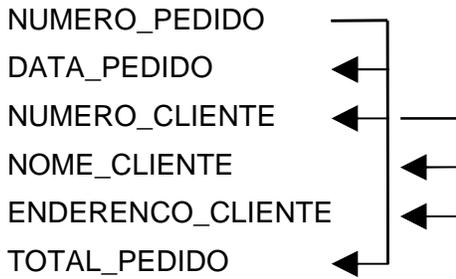
PEDIDO (NUMERO_PEDIDO + DATA_PEDIDO + NUMERO_CLIENTE +
 NOME_CLIENTE + ENDERENCO_CLIENTE + TOTAL_PEDIDO)

PRODUTO_PEDIDO (NUMERO_PEDIDO + NUMERO_PRODUTO + QTDE_PEDIDA +
 TOTAL_PRODUTO)

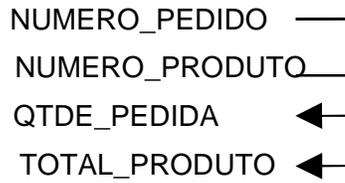
PRODUTO (NUMERO_PRODUTO + NOME_PRODUTO + PRECO_PRODUTO)

- Remoção das dependências transitivas

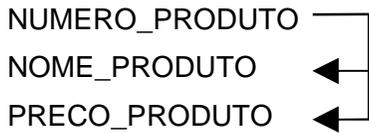
PEDIDO



PRODUTO_PEDIDO



PRODUTO



- Entidades na 3FN

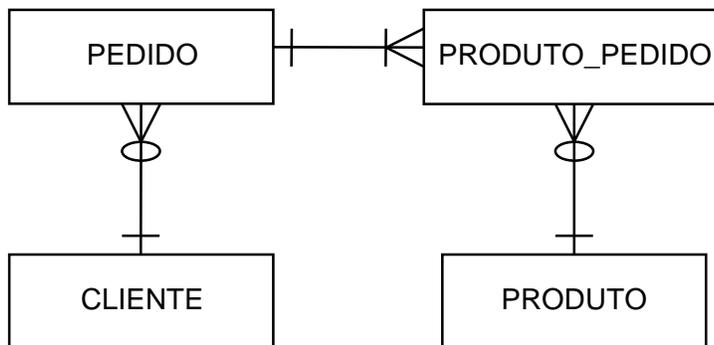
PEDIDO (NUMERO_PEDIDO + DATA_PEDIDO + TOTAL_PEDIDO)

CLIENTE (NUMERO_CLIENTE + NOME_CLIENTE + ENDERENCO_CLIENTE)

PRODUTO_PEDIDO (NUMERO_PEDIDO + NUMERO_PRODUTO + QTDE_PEDIDA + TOTAL_PRODUTO)

PRODUTO (NUMERO_PRODUTO + NOME_PRODUTO + PRECO_PRODUTO)

- Modelo de Dados.

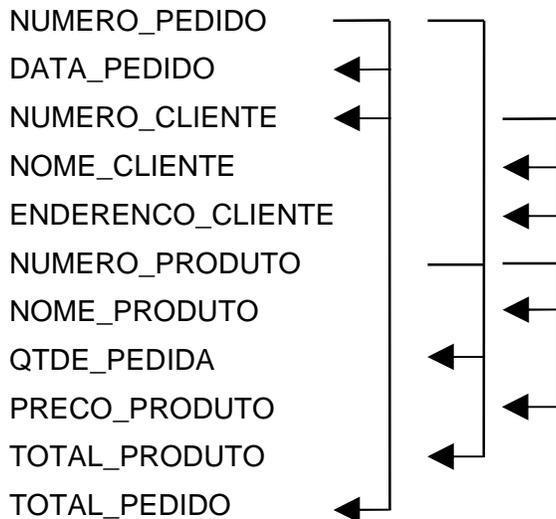


3.4.9 Simplificação do Processo de Normalização

A partir do diagrama de dependências funcionais podemos obter diretamente as entidades na

terceira forma normal. Para isso, devemos especificar uma entidade para cada conjunto de setas que o diagrama mostrar. A chave primária será formada pelos atributos dos quais partem as setas.

Exemplo:



3.4.10 Regras Práticas

- Se duas entidades possuírem a mesma chave de identificação:
 - Elas são a mesma entidade;
 - Seus atributos se complementam;
 - As suas ocorrências se complementam;
- Quando um atributo ou um conjunto de atributos identificadores de uma determinada entidade, for(em) também atributo(s) de uma outra entidade, deve haver um relacionamento do tipo 1:N entre elas.
- Atributos comuns a mais de uma entidade, devem ser, obrigatoriamente, chaves de identificação em uma das entidades; caso contrário será uma simples redundância.
- Nenhum atributo componente de uma chave primária deve poder assumir um valor nulo. Decorre do fato de que todos os objetos que se quer representar devam ser distinguíveis entre si.
- Um atributo que seja chave estrangeira só pode assumir:
 - Valor nulo;
 - Valor para o qual exista uma ocorrência da entidade da qual ela é chave primária.

3.4.11 Exercícios:

3.4.11.1 Estudo do Caso 1

Projetar uma base de dados - modelo relacional - nível conceitual - para atender a área de compras de uma empresa. Em levantamentos efetuados com o pessoal responsável pelo setor de compras, foram identificados os seguintes informações: E identificar as chaves de cada entidade.

ORDEM DE COMPRA

Código-Ordem-Compra

Data-Emissão

Código-Fornecedor

Nome-Fornecedor

Endereço-Fornecedor

%Materiais da ordem de Compra

codigo-item (n)

descrição-item(n)

valor-unitario-item (n)

quantidade-comprada-item (n)

valor-total-item (n)

valor-total-compra

Procure obter um modelo de dados sem redundâncias, desenhando o diagrama ER.

3.4.11.2 Estudo do Caso 2

Projetar uma base de dados - modelo relacional - nível conceitual - para atender a área de recursos humanos de uma empresa. Em levantamentos efetuados foram identificados os seguintes dados:

DADOS FUNCIONÁRIOS

Matricula - Funcionário

Nome - Funcionário

Endereço - Funcionário

Data - Funcionário

Código - Cargo

Valor - Salário

Numero - total - dependentes

Código - departamento

%Habilidades (grupo multivalorado)

 Código - Habilidade (n)

 Descrição - Habilidade (n)

 Data - Formação - Habilidade (n)

% dependentes (grupo multivalorado)

 Código - Dependente

 Nome - Dependente

 Data - nascimento - Dependente (n)

DADOS DEPARTAMENTO

Código - Departamento

Nome - Departamento

Localização - Departamento

Procure definir um modelo de dados sem redundâncias.

3.4.11.3 *Estudo do Caso 3*

Projetar uma base de dados - modelo relacional - nível conceitual - para atender necessidades de um candidato ao governo do estado.

Foram identificados os seguintes informações:

-
-

CIDADES

Código-Cidade(CEP)

Nome-Cidade

População-Cidade

Prefeito-Atual-Cidade

Partido-Prefeito-Atual

%Zonas (n)

Numero-Zona (n)

Local-Zona (n)

Número-Eleitores-Zona (n)

Cabo-Eleitoral-Principal(n)

VEREADORES

Código-Vereador

Nome-Vereador

Código-Cidade

Nome-Cidade

Partido-Vereador

Voto-Ultima-Eleição

Mandato-Vereador(período)

DEPUTADOS

Código-Deputado

Nome- Deputado

Código-Cidade

Nome-Cidade

Voto-Ultima-Eleição

Partido-Deputado

Mandato-Deputado(período)

Categoria-Deputado(Estadual-Federal)

PRINCIPAIS SOLICITAÇÕES CIDADES

Código-Cidade

Nome-Cidade

%Solicitações

Número-Solicitação (n)

Descrição-Solicitação(n)

Data-Solicitação (n)

Viabilidade-Atendimento(n)

Orgãos-Envolvidos(n)

- Obter E-R normalizado até a 3^a forma normal.
- Demonstrar as normalizações efetuadas via esquema de dependências funcionais
- Listar suposições que julgar necessárias sobre a semântica das dependências envolvidas
- Explicar campos de relacionamentos
- Se julgar necessário inclua novos campos e justifique a inclusão.

4. Bibliografia

Gane, Chris/Sarson, Trish - "Análise Estruturada de Sistemas.

King, David - "Criação de Software".

Guimarães, Ângelo de M. / Lages, Newton A.C. - "Algoritmos e Estruturas de Dados".

Coad, Peter/ Yourdon, Edward - "Análise Baseada em Objetos".

DeMarco, Tom - "Análise Estruturada e Especificação de Sistemas".

Yourdon, Edward - "Análise Estruturada Moderna.

Martins, James - Princípios de Análise e Projeto Baseado em Objetos.