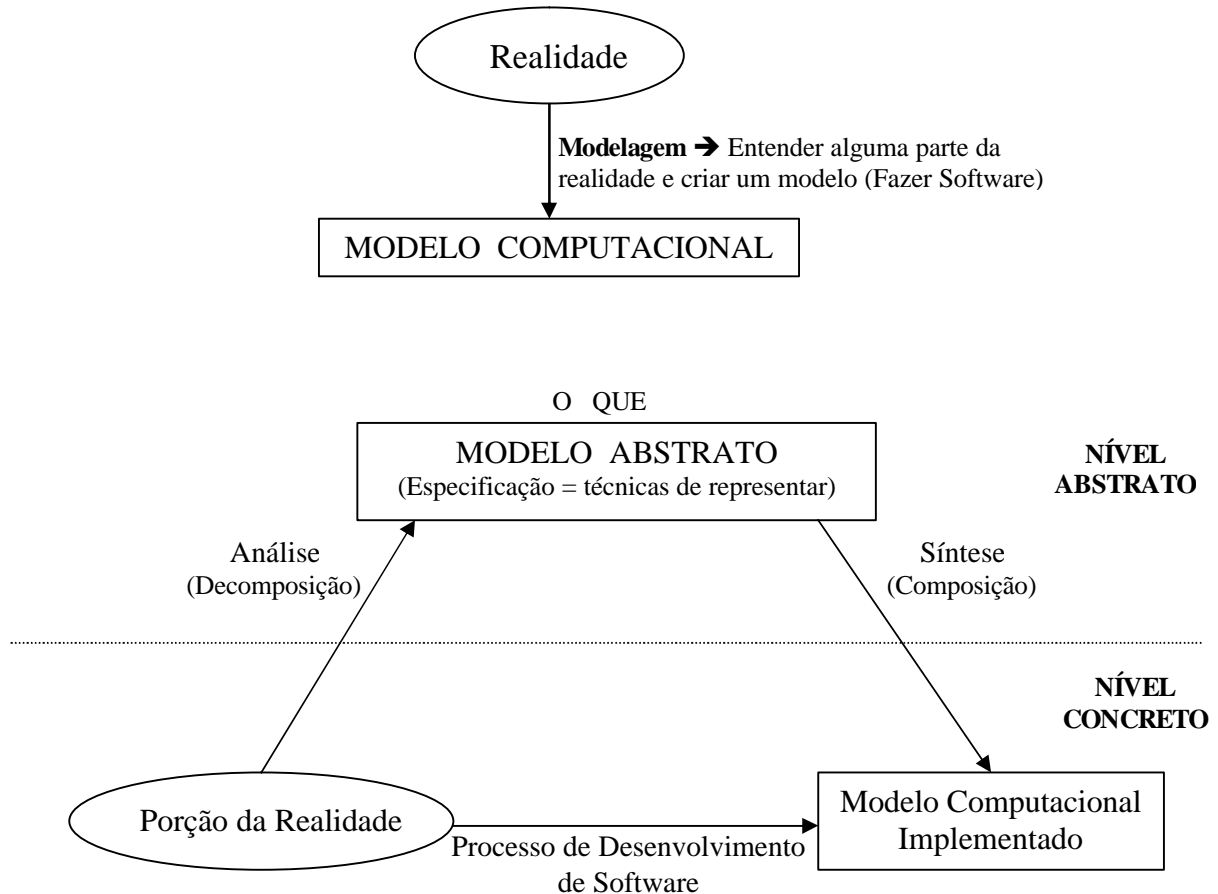
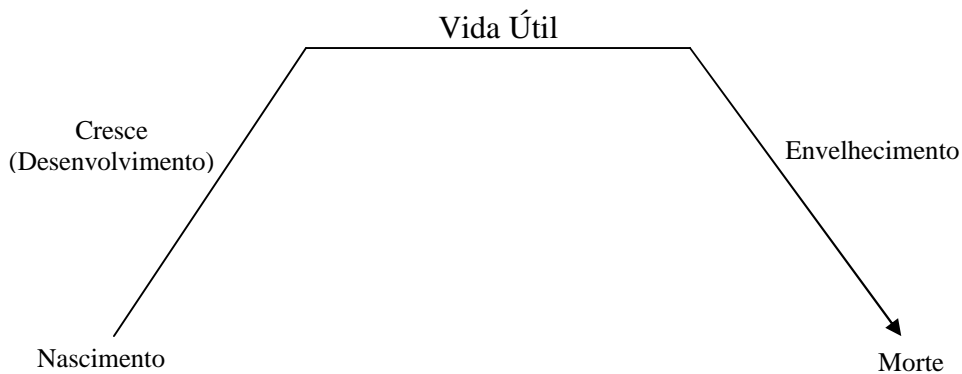


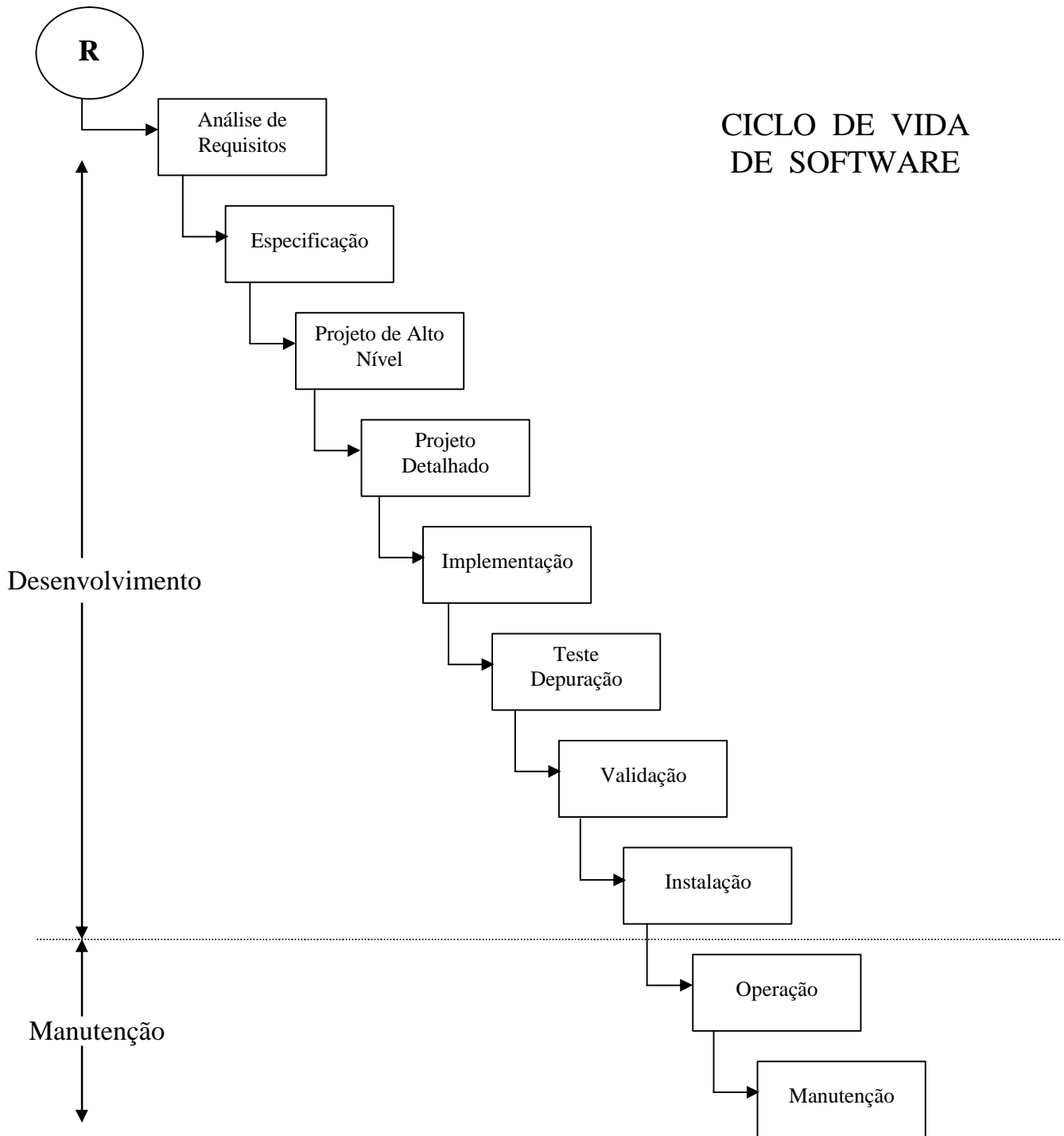
3. Metodologia E Documentação No Desenvolvimento De Sistemas

3.1 - Ciclo De Desenvolvimento De Software



Ciclo De Vida (Tradicional) De Software





3.1.1 - Característica:

Dividido em etapas sucessivas (os resultados de uma etapa são entradas da etapa seguinte)

3.1.2 - Princípios:

- 1º) Entender
- 2º) Definir
- 3º) Implementar

3.1.3 - Sistematização do processo de construção de software:

- Organizar Trabalho
- Gerenciar Equipe
- Documentar

3.1.4 - Fases do Ciclo de Vida Tradicional:

- 1. ANÁLISE DE REQUISITOS**
 - Estudo e compreensão do problema
 - Estudo do ambiente e usuários em que o problema se insere
 - Coleta de informações
 - Levantamento e identificação das necessidades (requisitos)
 - Estudo e Validação das informações:
 - O que o usuário espera
 - Limitações: Orçamento, Cronograma, Plataformas (Hardware, Software, ...)
 - Estudo da viabilidade
- 2. ESPECIFICAÇÃO → O Que Fazer**
 - Descrição conceitual (abstrata) do sistema
 - Representação precisa, completa e clara do QUE o sistema irá fazer, para evitar más interpretações.
 - Modelagem das FUNÇÕES que serão desempenhadas pelo sistema e das INFORMAÇÕES que serão manipuladas.
- 3. PROJETO DE ALTO NÍVEL**

Projeto: Definição de alternativas para satisfazer os requisitos e seleção de UMA (alternativa) que descreve COMO fazer o sistema conforme o ambiente, os recursos disponíveis e os critérios de qualidade adotados.

 - Construção de arquitetura do software – organização modular do software, ou seja, desenvolvimento de uma hierarquia apropriada de módulos de programa e interfaces entre esses módulos para implementar a especificação criada na fase 2 (especificação).
 - Projeto de Estrutura de Dados: transformação de modelos de dados de entidades-relacionamentos em um projeto de banco de dados.
- 4. PROJETO DETALHADO**
 - Refinamento da alternativa escolhida visando implementação

- Fluxogramas – algoritmos
- Descrição detalhada de cada um dos componentes

5. IMPLEMENTAÇÃO

- Transformação do projeto detalhado em um sistema concreto (real) – código executável
 - Codificação em uma linguagem de programação
 - Uso da máquina

6. TESTE E DEPURAÇÃO

- Testes: Execução (virtual ou real) do(s) componente(s) dos software visando encontrar erros.
 - “O sistema faz o que foi especificado?”
 - Depuração: Processo de localização e correção dos erros encontrados

7. VALIDAÇÃO

- Submissão do sistema ao usuário
- Verificação do atendimento dos requisitos
- “O sistema faz o que era esperado?”

8. INSTALAÇÃO / ENTREGA

- Integração do sistema ao ambiente do usuário
- Entrega e instalação do sistema

9. OPERAÇÃO

- Uso do sistema

10. MANUTENÇÃO

- Modificação do software após a entrega
- Correção de possíveis erros
- Alteração de requisitos (acrescentar funções, remover funções, melhorar funções, modificar funções)
- Adaptação a novas plataformas (hardware, software, linguagem de programação, etc.)

3.2 - Análise De Requisitos

É a 1ª Fase do Ciclo de vida do software.

3.2.1 - Idéia Fundamental

Entender precisamente os objetivos do (novo) sistema, ou seja, obter a Coleta de informações visando:

➔ Entender o ambiente:

- entender o comportamento do ambiente atual (manual ou informatizado)

- entender as necessidade (requisitos) de um novo sistema

➔ Verificar nossa compreensão do ambiente (validação)

O analista de sistema não é simplesmente uma pessoa com informação de computação:

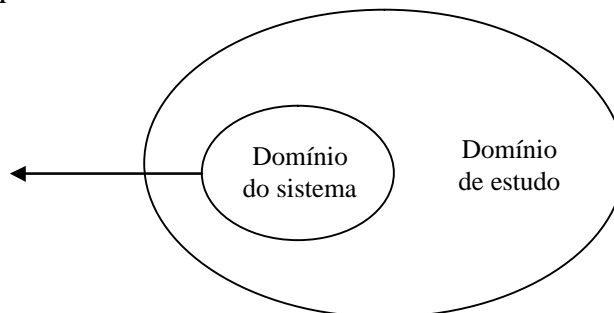
- Conhecimento tecnológico (noções de hardware, software)
- Habilidade para lidar com pessoas (comunicação): usuários, equipe
- Experiência com o domínio do problema – experiência prática (ambiente)

Não existe metodologia genérica para esta fase ➔ depende da aplicação

3.2.2 - Problemas na Análise

1. Escopo: O quanto do ambiente estudar?

- Diz respeito a alguns aspectos
- Não diz respeito a outros aspectos (não deve afetar)



Dica: Separar **SINTOMAS** de **CAUSAS** – evitar o aumento do escopo (domínio de estudo)

Reconhecimento do problema

A idéia de desenvolvimento de um novo sistema ocorre ao usuário quando ele reconhece ter problemas com os meios com os quais ele desenvolve seu negócio.

Tipicamente, o usuário está mais preocupado com a maneira pela qual ele está executando seu trabalho no que tange às deficiências físicas do sistema que utiliza.

2. Estudo da viabilidade: Verifica se todas as informações estão corretas (análise de custo-benefício: \$, tempo e esforço)

Talvez o usuário precise de um computador – mas talvez não. Esta é uma das decisões que devem ser tomadas após o término do próximo estágio: o estudo da viabilidade (ou avaliação). Este estudo serve para:

- Identificar o escopo do sistema corrente a ser estudado;
- Identificar problemas com o sistema corrente;
- Identificar os objetivos principais para o novo sistema;
- Identificar um número (possivelmente nenhum) de soluções que possam satisfazer as necessidades do usuário dentro de seu orçamento e planos;
- Desenvolver estimativas das vantagens e desvantagens de cada solução;
- Desenvolver um plano de realização do projeto com estimativas dos recursos necessários;
- Obter o parecer do usuários e da administração incluindo alterações nos tópicos anteriores;
- Obter decisão do usuários e da administração e o comprometimento deles até pelo menos a parte da análise do projeto.

OBS: Para a maioria dos projetos, o estudo de viabilidade consome em tempo menos de um mês.

3.2.3 - Fases da Análise de Requisitos

1. *Identificar usuários responsáveis e desenvolver um ESCOPO “inicial” do sistema (Não existe método para isso)*

➔ Universo de informação:

- Todas as FONTES de informação
- Todas as PESSOAS relacionadas ao sistema

➔ Quais usuários estão envolvidos? Quais não?

➔ Primeira estimativa do sistema / Primeira impressão

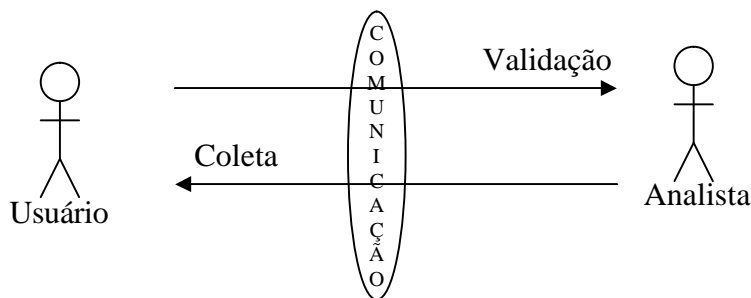
2. *Identificar necessidades atuais*

➔ Coletar aspectos (informações, tarefas, limitações) da organização: política do usuário

➔ Coletar os aspectos inadequados, que não estão de acordo, de preferência em ordem de prioridade:

- os que faltam
- os que atuam insatisfatoriamente (não confiáveis, lentos, etc.)
- os com custo excessivo
- etc.

- COLETA {
- 2a) Coleta de Informações : enumerar, listar os fatos ou informações que traduzem o mundo real, evitando ao máximo interpretações ("eu acho").
 - 2b) Comunicação : diálogo entre analista(s) e usuário(s).
 - 2c) Validação da Informação : confirmar acuidade do processo de coleta (evitar erros no início do processo).



2a) Técnicas de Coleta

- Evitar interpretações
 - Se há dúvidas, validar com o usuário e não assumir um default
- ➔ Entrevistas: Camuflada (não conversa social)
Formal (hora marcada, conjunto de questões pré-definidas, etc.)
- ➔ Questionários
- ➔ Observação do ambiente (participativa ou não)
- ➔ Análise de formulários (ver os documentos que a empresa tem)
- ➔ Análise do(s) sistema(s) informais existentes

2b) Comunicação

- Diálogo – interação entre analista-usuário
- Analista lidera o processo (dever de iniciativa, flexibilidade, clareza, ética)
- Engloba:
 - Apresentação: apresentar os fatos coletados
 - Negociação: resolução de conflitos, contradições
 - Linguagem: uso do vocabulário do usuário (domínio)
 - Participação do ambiente / entrosamento: ou usuário na equipe ou analista na empresa (grandes projetos)
- Treinamento

2c) Validação

- Tentar encontrar diferenças entre informações coletadas e o universo (real) da informação
- Confirmar que os fatos e requisitos (informações) do ambiente foram bem compreendidos
- Basicamente:
 - Leitura de descrições do ambiente, em linguagem natural (português)
 - Uso de “check-lists” para detectar problemas na expressão / formulação dos requisitos
 - check-lists → Gera lista de dúvidas / problemas / inconsistências: informações-faltantes, informações erradas, informações ambíguas

3. Estabelecer os OBJETIVOS para novo sistema (aplicação)

- Objetivos Funcionais
- Objetivos de performance / desempenho: objetivos da organização + objetivos da aplicação (requisitos)

OBS: Essas funções estão associados às necessidades detectadas

- Lista exaustiva (completa) de:
 - Funções requisitadas (novas funções e funções a serem modificadas / corrigidas)
 - Requisitos de performance / rendimento (por ex., tempo de resposta)
 - Requisitos de confiabilidade / precisão
 - Respeito às limitações (custo, prazos, recursos humanos e materiais, ...)

4. Determinar se é possível automatizar (automático) ou informatizar (organizar as informações) o sistema.

- Sugerir diversos sistemas possíveis e escolher uma alternativa junto com o usuário (verificar razões não só tecnológicas)
- Breve resumo dos novos sistemas propostos dignos de exame.
- Analisar limitações: tempo (prazo), orçamento, recursos financeiros, limites operacionais. Exemplos → solução mais barata, solução mais rápida, solução mais sofisticada

OBS: Podem não haver novos sistemas possíveis, ou seja, a forma atual (com todas as deficiências) pode ser mantida.

5. Preparar uma previsão do desenvolvimento (anteprojeto)

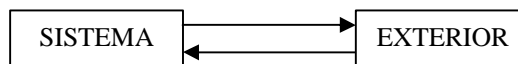
- Será usado para conduzir o restante do desenvolvimento
- Plano de Trabalho Detalhado:

- Definindo e descrevendo todas as atividades subsequentes, todas as subatividades, resultados intermediários e finais
- Cronograma

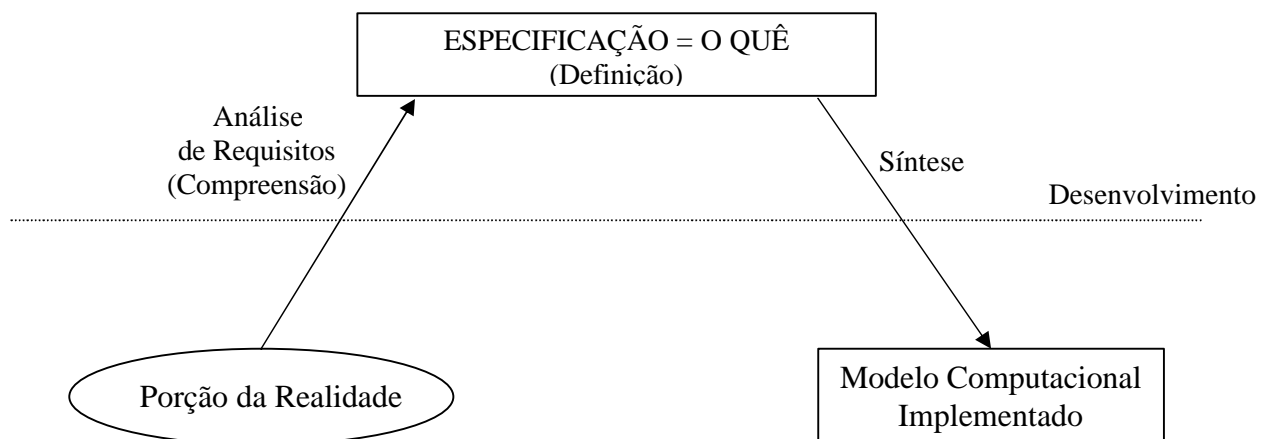
3.3 – Especificação De Sistemas

Definição do sistema de maneira completa, consistente, concisa e não ambígua.

- Descrição clara de **O QUÊ** o sistema irá fazer
- Componentes:
 - Componentes do sistema: define a estrutura
 - Interligações entre estes componentes: define a estrutura
 - Ligações do sistema com o ambiente externo: define o comportamento e a contextualização



- É o modo de representar o entendimento do problema: evitar erros de desenvolvimento
- É o modo de comunicação entre usuário(s) e a equipe: evitar confusões e expectativas errôneas



3.3.1 - Dificuldades para Especificação

- Requisitos/objetivos são imprecisos
 - Solução: melhor comunicação entre usuário e analista e equipe – comunicação gráfica (por diagramas)
- Requisitos/objetivos são dinâmicos

- Solução: Fácil modificação, estruturação em partes, alterar pequenas porções da especificação

Modelo → Serve para representar alguma coisa

Modelagem → É a criação de modelo (similar a especificação)

- Modelagem por Dados: concentra-se atenção na informação, nos dados, ou seja, o raciocínio é executado sobre os dados. Dados de entrada e saída, por exemplo. DER, entidades e atributos. É um processo mais demorado e estável. É uma Descrição Estática.

- Modelagem por Atividades: concentra-se atenção sobre as tarefas, as atividades, ou seja, o raciocínio é realizado sobre as funções do sistema. DFD, DD e especificação de processos. É um processo mais instável e imediato. É uma Descrição Dinâmica.

Daí, surge a **ANÁLISE ESTRUTURADA**, que é uma técnica de modelagem especificada em modelagem por atividades e dados.

3.3.2 - Análise Estruturada Moderna

Relembrando: Análise Clássica

- Textual, monolítica: era necessário ler toda a especificação funcional, do princípio ao fim, para poder compreendê-la.
- Redundante, difícil de manter/modificar: a mesma informação era muitas vezes repetida em diversas partes do documento. Isso conduzia a um grave problema: a inconsistência.
- Eram ambíguos: o detalhamento dos requisitos do usuário podia ser interpretado de modo diferente pelo usuário, pelo analista de sistemas, pelo programador, pelo projetista.
- Cheia de detalhes de implementação: a manutenção era impossível.

A Análise Estruturada

Princípio: “Dividir para conquistar”: 1 “problemão” em n “probleminhas”.

3.3.3 - Características da Análise Estruturada

- Gráfica e Concisa: contém símbolos do sistema, com adequado detalhamento textual de apoio, em vez de conter somente explicações por escrito. Usuários são mais propensos a olhar figuras, a ter de ler um documento mais longo.

- Não Redundante: uma informação é guardada uma e somente uma vez. Isso garante a consistência e fácil atualização.

- **Particionamento Top-Down:** segmentos pequenos e suaves, ou seja, para mostrar o exterior e visões gerais do sistema, seguidas de visões concentradas e detalhadas de uma área local. sistema subdividido

OBS: Top-Down: significa que alguma codificação e alguns testes ocorram paralelamente a outras fases do ciclo de vida. Isso pode significar uma realimentação entre a atividade de codificação e de teste e a de depuração de erros, ou seja, o processo top-down muitas vezes causa a realimentação entre o processo de implementação e o processo de análise. (Pg. 109 do Yourdon)

- **Lógica, e não Física:** concentra-se no que o sistema fará para o usuário (o lógico) e não como o sistema será implantado, ou seja, retrata as características essenciais sem se restringir a nenhuma implementação específica.

OBS.: Não confundir análise, projeto e programação estruturada.

Análise Estruturada: introduz uma ferramenta gráfica para facilitar a comunicação com o usuário através de DFDs. O objetivo é obter uma especificação precisa da realidade

Projeto Estruturado: procura documentar melhor o sistema e permitir uma implementação mais fácil e integrada. Está orientado para a solução implementação da solução do problema. Isto é, sua função é de transformar uma representação lógica “do que” um dado sistema é requerido a fazer, numa representação física de “como” o sistema irá fazer.

Programação Estruturada: Objetiva elaborar programas de claro entendimento, de modo a agilizar os testes e facilitar manutenções.

Modelo Essencial: indica o que o sistema deve fazer para satisfazer os requisitos do usuário, mencionando o mínimo possível sobre como o sistema será implementado. É composto por dois componentes:

1. **Modelo Ambiental:** Definir ligações (interfaces) entre o sistema e o resto do ambiente (contexto).

* *Tarefa difícil:* determinar o que faz parte dos sistemas e o que não faz parte.

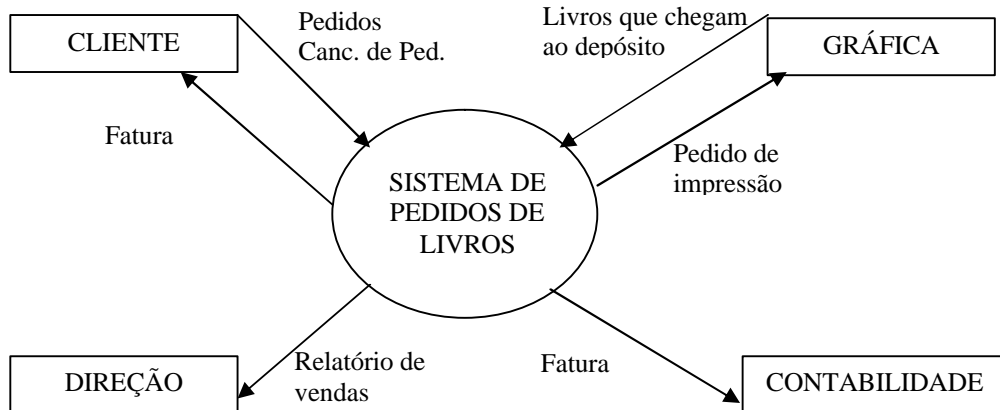
* Determina informações necessárias para realizar tarefas (entradas) e informações produzidas pelo sistema (saídas). Eventos que ocorrem no ambiente e aos quais o sistema deve reagir.

* entradas → estímulos; saídas → respostas, reações

- **Diagrama de Contexto:** descrição gráfica que representa as trocas de informações entre o sistema e o ambiente. Notação:

- Uma “bolha” representa o sistema inteiro.
- Fluxos (setas com identificação) representam as trocas de informação.
- Pessoas, organização ou outros sistemas (que compõe o ambiente) com os quais o sistema se comunica são representados por retângulos.

- Portanto, mostra apenas as Entidades Externas, Entradas e Saídas do Sistema.
- Veja um exemplo a seguir.



2. Modelo Comportamental: ligação entre os componentes do sistema. Mostra como são tratados os eventos¹. Descreve o comportamento do sistema de modo a interagir satisfatoriamente com o ambiente, ou seja, o que é feito para produzir respostas aos eventos.

3.3.4 - Ferramentas Componentes

- Diagramas de Fluxos de Dados (DFD's)
 - Dicionário de Dados (DD)
 - Especificação de Processos
 - Diagramas de Entidade-Relacionamento (DER)
- Modelagem por Atividades
- Modelagem por Dados

Leitura complementar recomendada:

YOURDON, Edward. "Análise Estruturada Moderna", Terceira Edição, Editora Campus, 1990, 836 p.

POMPILHO, S. "Análise Essencial – Guia Prático de Análise de Sistemas", IBPI Press, 1995, 298 p.

¹ Ocorrência significativa que deve ser reconhecida e reagida pelo sistema.