



Análise de Sistemas

Aula 2

Prof. Emerson Klisiewicz

Contextualização

Aula 2

- Análise Estruturada
- Análise Essencial
- Análise Orientada a Objetos

Introdução

- Grandes mudanças nos últimos anos, e o avanço da tecnologia, levaram a uma passagem da antiga sociedade industrial para a sociedade baseada em informação

- Nos dias de hoje, a empresa que dispõe de mais informações sobre seu processo está em vantagem em relação a suas competidoras

Instrumentalização



Um Pouco de História

Evolução

▪ Anos 60

- Textos e fluxogramas para especificar a lógica dos sistemas

▪ Meados de 70

- Análise estruturada: método que utiliza os componentes e conceitos da programação estruturada (Tom DeMarco, Gane & Sarson)

▪ Anos 80

- Surge a abordagem de estrutura de dados como forma de modelar sistemas

▪ Anos 90

- Surge a abordagem orientada a objetos

Métodos	Abordagens	Ferramentas
Análise Estruturada	<ul style="list-style-type: none"> ▪ Funcional ▪ Dados 	<ul style="list-style-type: none"> ▪ Diagrama de Fluxo de Dados (DFD) ▪ Diagrama de Estrutura de Dados ▪ Dicionário de Dados
Análise Essencial	<ul style="list-style-type: none"> ▪ Funcional ▪ Dados ▪ Controle 	<ul style="list-style-type: none"> ▪ Lista de Eventos ▪ DFD ▪ Diagrama Entidade-Relacionamento (DER) ▪ Diagrama de Transição de Estados (DTE) ▪ Normalização ▪ Miniespecificações ▪ Dicionário de Dados



Análise Estruturada

- A análise estruturada é uma abordagem sistemática para fazer a análise de um sistema de modo a produzir uma especificação funcional

Componentes do Modelo Estruturado

- Diagrama de Fluxo de Dados (DFD): representação gráfica da rede de processos interligados

- Dicionário de Dados: descrição das interfaces
- Especificação dos Processos: descrição do que cada processo faz

Diagrama de Fluxo de Dados (DFD)

- Forma gráfica de mostrar:
 - a interdependência dos processos que compõem um sistema

- os fluxos de dados entre elas
- arquivos lógicos de dados – depósito de dados
- entidades externas



DFDs desenvolvidos em níveis hierárquicos

- Um DFD não deve ter mais de 7+/-2 processos (A4)

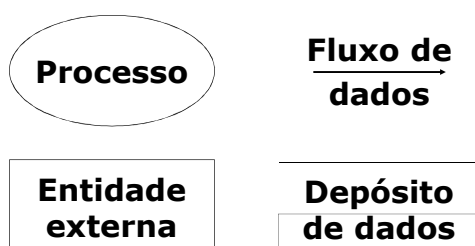
- O DFD de um Sistema de Informação é desenvolvido de acordo com uma decomposição hierárquica nos seguintes níveis (diagramas)

- Diagrama de contexto: diagrama de visão mais elevada. É um único processo representando o sistema inteiro e fluxos de dados que representam as interfaces com o exterior

- Diagrama 0 – visão global do sistema
 - ✓ Visão de alto nível dos principais processos do sistema e as ligações entre esses processos

- DFDs de nível n-1 – detalhe do sistema
- Sistema complexidade baixa → 2 a 3 níveis
- Sistema complexidade média → 3 a 6 níveis
- Sistema complexo → 5 a 8 níveis

Notações básicas do DFD



**Processo**

- Representa um transformador de informações que resida dentro dos limites do sistema a ser modelado

Entidade externa

- Representa um produtor ou consumidor de informações que resida fora dos limites do sistema a ser modelado

Fluxo de dados

- Representa o deslocamento de um item de dado ou coleção de itens de dados

Depósito de dados

- Representa um repositório de dados que são armazenados para serem usados em um ou mais processos

O dicionário de dados

- Proposto como gramática quase formal para descrever o **conteúdo de objetos** definidos durante a análise estruturada

- A maioria dos DD contém as seguintes informações
 - **Nome:** o nome principal do item de dados, do depósito de dados ou de uma entidade externa
 - **Alias:** outros nomes usados para a primeira entrada



- **Onde é usado/Como é usado:** listagem dos processos que usam o item de dados e como ele é usado
 - Ex.: entrada no processo, saída do processo

- **Descrição de conteúdo:** notação para representar o conteúdo
- **Informação complementar:** outras informações sobre tipos de dados, domínios, restrições ou limitações

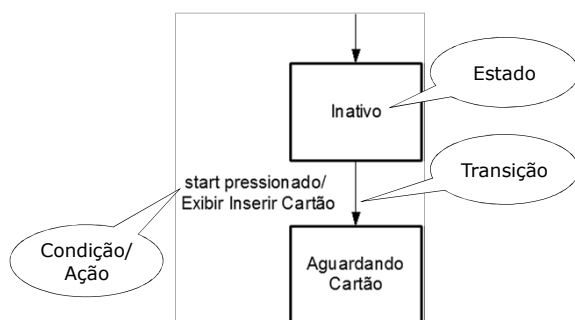
Diagrama de Transição de Estados (DTE)

- O Diagrama de Transição de Estados serve para especificar comportamentos do sistema em relação a eventos que ele recebe

DTE – Elementos

- São: estado, transição e ação
- As setas indicam como o sistema reage a eventos quando eles passam pelos estados do sistema

Elementos do DTE



Estados Típicos

- Aguardando o usuário introduzir sua senha
- Aguardando o próximo comando
- Aguardando dados para instrumento

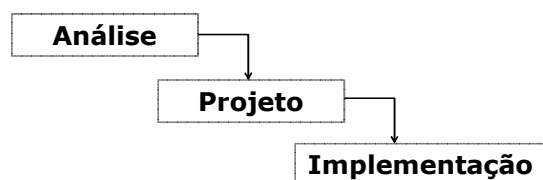


- Acelerando o motor
- Aquecendo uma mistura química
- Misturando ingredientes
- Enchendo o tanque
- Ocioso

Análise Essencial

O Desenvolvimento de um Sistema

Ciclo de vida do desenvolvimento de sistemas



- A análise essencial, propõe o particionamento do sistema por eventos
- Um **evento** pode ser definido informalmente como um acontecimento do mundo externo, que requer uma resposta do sistema

- Um **estímulo** é um ativador de uma função do sistema. É a forma como o evento age sobre o sistema
- Uma **resposta** é o resultado gerado pelo sistema devido à ocorrência de um evento

Definições

- **Processo**: conjunto de atividade que produz, modifica ou atribui qualidade às informações
- **Depósito de dados**: conjunto de informações armazenadas pelo processo que serão utilizadas por algum processo



- **Entidade externa:** é algo situado fora do escopo do sistema, que é fonte ou destino das suas informações
- **Fluxo de dados:** o nome deve expressar o significado do conjunto de informações que está fluindo

Miniespecificação

- A miniespecificação define a forma pela qual os fluxos de dados de entrada são transformados em fluxos de dados de saída, independente do fato da função ser executada manualmente ou por outra forma de implementação

- Em relação ao segundo aspecto, as principais técnicas de especificação são:
 - português estruturado
 - pseudocódigo
 - tabela de decisão
 - árvore de decisão

Diagrama Entidade

- O diagrama entidade relacionamento é um modelo diagramático que descreve o modelo de dados de um sistema com alto nível de abstração. Principal representação do Modelo de Entidades e Relacionamentos

DER – Componentes

- Composto de poucos símbolos gráficos que representam relacionamentos do banco
- **Retângulo:** é a entidade do BD, que muito provavelmente será uma tabela quando o banco for criado

- **Losango:** representa o relacionamento entre as entidades
- **Triângulo:** representa as especializações
- **Bolinhas:** representa os atributos

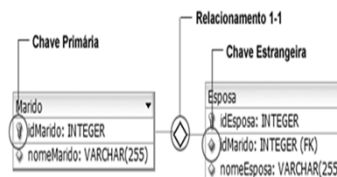


- **Chaves primárias:**
uma chave primária é um atributo usado como identificador do item na entidade. Deve ser único

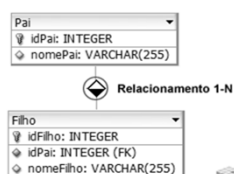
- **Entidades:** são qualquer elemento que possua atributos que serão utilizados na base de dados
 - Exemplo: CPF

- **Chaves estrangeiras:**
são responsáveis pelo relacionamento entre duas entidades
- **Relacionamento:** os relacionamentos ocorrem entre as entidades, e esse conjunto forma a base do banco de dados

- O relacionamento 1-1 (um para um) ocorre quando uma se relaciona com no máximo 1 para com a outra



- Os relacionamentos 1-N (um para n, um para muitos) ocorrem quando o relacionamento da entidade A para entidade B é de no máximo 1 e de B para A de no máximo N



- Os relacionamentos N-N (n para n, muitos para muitos) ocorrem quando tanto de A para B quanto de B para A o máximo de N



- Um exemplo desse tipo de relacionamento é o de compra, onde um cliente pode comprar vários produtos, e um produto pode ser comprado por vários clientes

- Exemplo N-N



Diagrama de Transição de Estado

- Em engenharia de *software*, diagrama de transição de estados representa estado ou situação que um objeto pode se encontrar durante a execução de processos

Análise Orientada a Objetos

- Tem como ênfase encontrar e descrever os objetos – ou conceitos – no domínio do problema que é a área de conhecimento específica na qual um sistema de *software* está sendo desenvolvido

- Nos métodos tradicionais de análise, o comportamento do sistema e seus dados eram considerados separadamente



- Com orientação a objetos, comportamento e dados são integrados, assim encapsulando detalhes internos de um objeto dos demais

Características

- Concentra-se nos aspectos essenciais do objeto sem detalhamento, focando em suas características e o que ele faz
- Impede que um sistema se torne tão interdependente

- Combina estrutura (dados) e comportamento (funções) em um único objeto
- Compartilha elementos estruturais e de comportamento com objetos de níveis inferiores
- Enfatiza estrutura de objetos ao invés da estrutura de procedimentos

Conceitos

- **Objeto:** cada conceito é uma ideia que temos do mundo. Essas coisas as quais nossos conceitos se aplicam são denominados objetos. Um objeto pode ser real ou abstrato
 - Exemplo: avião

- **Classe:** é a representação do objetos com seus atributos e métodos

- **Métodos ou serviços:** as ações que um objeto pode executar
 - Exemplos: latir, comer, sentar, dormir etc.



- **Atributos:** são características que descrevem o objeto
- Exemplos: objeto: cão – possui nome, idade, peso, cor etc.

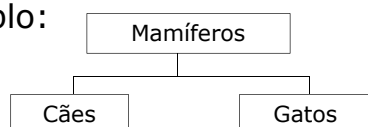
- **Abstração:** só deve ser representado aquilo que vai ser usado. Nos objetos são representadas somente as características que são relevantes para o problema em questão

- Exemplo: cor dos olhos (toda pessoa tem) – não é relevante para um sistema de folha de pagamento

- **Encapsulamento:** os dados e os processos que tratam esses dados estão “encapsulados” numa única entidade. Os objetos agem como uma “caixa preta”, se utiliza sem precisar saber como ele funciona

- **Hierarquia de classes:** classe que tem características comuns e que podem fazer parte de uma classe (categoria) maior

- Exemplo:



- **Classes ancestrais:** classes das quais as outras dependem
- **Classes descendentes:** as classes originadas a partir de outra



- **Herança:** significa que todos os atributos e métodos programados no ancestral já estarão automaticamente presentes em seus descendentes sem necessidade de reescrevê-los

- **Polimorfismo:** é o princípio relacionado com as diferentes formas de um objeto

- O polimorfismo pode ser visto no exemplo abaixo, onde se pode instanciar o objeto janela de várias formas:

Janela ()	<input type="text"/>
Janela (1 x 2, 2)	<input type="text"/> <input type="text"/>
Janela (1 x 2, 2, azul)	<input type="text"/> <input type="text"/>

- **Agregação:** é um mecanismo que permite a construção de uma classe agregada a partir de outras classes componentes
- Exemplo: casa

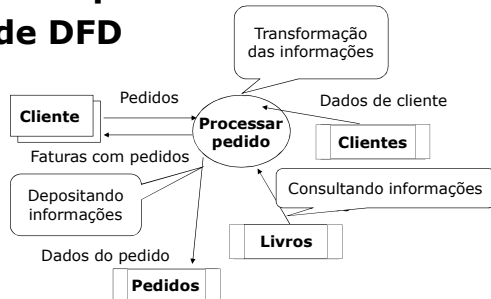
- **Associação:** é usada para agrupar objetos que ocorrem em algum ponto no tempo ou sob circunstâncias similares
- A associação é modelada através de uma conexão de ocorrências

Aplicação



Análise Estruturada

Exemplo de DFD



Especificação de Processo (Português Estruturado)

Analisar triângulo

Procedimento: analisar triângulo;

Ler dimensões dos lados;

Se qualquer dimensão for negativa

Então produza mensagem erro

Se a maior dimensão for menor que a soma das demais

Então

Se três lados forem iguais **então** o tipo é equilátero;

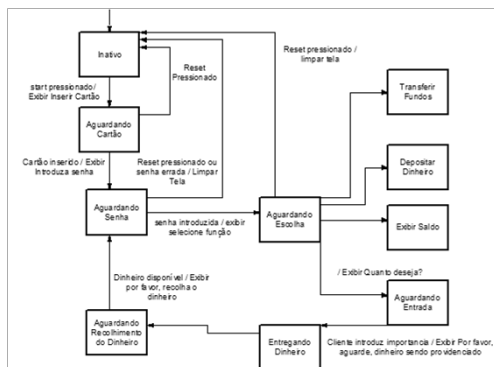
Se dois lados forem iguais **então** o tipo é isosceles;

Se nenhum dos lados forem iguais **então** o tipo é escaleno;

Imprima tipo de triângulo;

Senão imprima tipo=0, indicação de que não existe nenhum triângulo

DTE

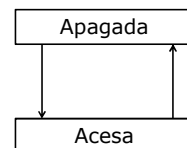


Análise Essencial

Diagrama de transição de estado

Interruptor
foi ligado

Acender
lâmpada



Interruptor
foi desligado

Apagar
lâmpada

Síntese

Análise Estruturada

- Dimensão exata das necessidades
- Expõe o que é feito por gráficos
- Dirigido para uma ferramenta
- Exige análise de cima para baixo através de refinamentos



Análise Essencial

- Os eventos são a pedra fundamental dos sistemas
- Especificação de um sistema deve começar pela identificação dos eventos

Análise Orientada a Objetos

- A essência é enfatizar, considerar um domínio de problema e uma solução lógica, segundo a perspectiva de objetos (coisas, conceitos e entidades)

Por que usar Orientação a Objetos?

- Atualmente temos ferramentas para sua utilização (integrando especificação e implementação)
- Praticamente todas as ferramentas novas de programação permitem suporte a sua utilização

- Produtividade em função do reuso
- Produção de códigos mais fáceis de serem entendidos
- Adequada a construção de sistemas distribuídos e para aplicações voltadas à Internet

Dificuldade

- Usuários não pensam em seus problemas de forma orientada a objetos
- Requisitos não são orientados a objetos