

# ENGENHARIA DE SOFTWARE

## *VIII Métodos e Técnicas de Análise e Projeto de Software*

### **Prof. André Luiz de Castro Leal**

Doutorando em Informática pela PUC-RIO  
Mestrado em Ciência da Computação pela UFV  
Especialista em Gestão de TI  
Especialista em Ciência da Computação  
[andrecaastro@ufrj.br](mailto:andrecaastro@ufrj.br)

### **Macro Estrutura do Conteúdo**

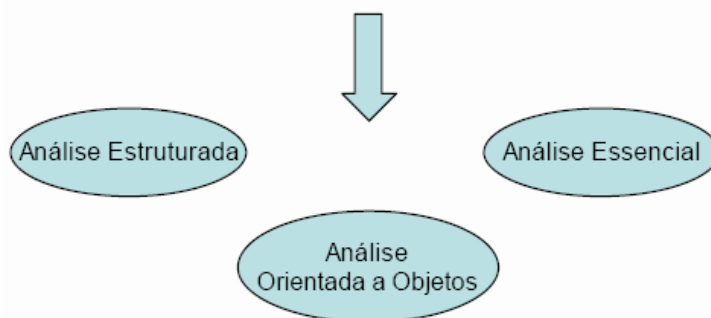
- 1) Linha do Tempo
- 2) Métodos de Desenvolvimento de Software
- 3) Análise Estruturada
- 4) Análise Essencial
- 5) Análise Orientada a Objetos

### Linha do Tempo

- 1957 – Criação da Linguagem de Programação Fortran
- 1958 – Criação da Linguagem de Programação Algol
- 1964 – Criação da Linguagem de Programação Basic
- 1967 - Ole-Johan Dhal e Kristen Nygaard desenvolvem a linguagem Simula-67, com vários conceitos que posteriormente seriam base para a Orientação a Objetos: classes, objetos, funções virtuais, garbage collection,...
- 1970 - Alan Key cunha os termos "object-oriented" e "objectoriented programming"
- ~1970 - Crise do Software
  - Elementos desencadeadores:
    - Desenvolvimento de Software de forma "artesanal" através do desenho de telas e arquivos
    - Constantes erros de execução
    - Pouco tempo para coletar dados
    - Não cumprimento de prazos
    - Problemas de custos inesperados relacionados a correção de erros e adaptação do código
    - Código/documentação ilegível ou inexistente
    - Comunicação durante o desenvolvimento muito fraca
    - Falta de testes complexos
    - Insatisfação de usuários

### Linha do Tempo

- ~ 1970 - Crise do Software
- A partir desta data foi iniciado o nascimento das Metodologias de Desenvolvimento de Softwares



### Linha do Tempo

1979 – O DoD (*Department of Defense*) lança a especificação preliminar de Ada.

1980 – Bjarne Stroustrup cria extensões orientadas a objeto para C e chama essa linguagem de "C com classes".

É lançada Smalltalk-80, com o conceito de metaclasses.

1981 – Grady Booch desenvolve um processo de desenvolvimento para Ada, com uma notação que viria a ser conhecida como "Diagramas de Booch".

1982 - A Agência Espacial Européia propõe uma técnica de projeto orientado a objetos, chamada HOOD (*Hierarchical Object-Oriented Design*), também direcionado para Ada.

1983 – A linguagem de Stroustrup é renomeada para C++

1985 – Bertrand Meyer propõe a linguagem Eiffel

1986 – Primeira conferência OOPSLA (*Object-oriented Programming, Systems, Languages, and Applications*)

1988 - Sally Shlaer e Stephen J. Mellor desenvolvem um método conhecido como *Object Oriented Structured Analysis* (OOSA), além de propor uma linguagem de modelagem chamada *Object Oriented Design Language* (OODLE).

### Linha do Tempo

1989 – É fundado o OMG (*Object Management Group*), que começa a trabalhar na especificação de CORBA (*Common Object Request Broker Architecture*)

1980 – Guido van Rossum desenvolve Python. Rebecca Wirfs-Brock propõe uma metodologia baseada em *Responsibility-Driven Design* (RDD) e cartões CRC (*Class-Responsibility-Collaboration*). Peter Coad e Edward Yourdon lançam uma metodologia conhecida como *Object-Oriented Analysis*.

1991 – James Rumbaugh propõe uma metodologia nomeada OMT (*Object Modeling Technique*). A notação do diagrama de classes de OMT viria a servir como base para UML.

1991 – A Sun começa a desenvolver uma nova linguagem orientada a objetos chamada Java, originalmente projetada para criar software para eletrodomésticos.

1992 – Ivar Jacobson lança o método OOSE (*Object Oriented Software Engineering*). Sua grande contribuição viria a ser a proposta dos Diagramas de Casos de Uso.

1994 – Grady Booch, James Rumbaugh e Ivar Jacobson começam a trabalhar na especificação de UML (*Unified Modeling Language*) Coleman e outros autores propõem a metodologia Fusion.

1995 - A Sun lança oficialmente Java. O JDK 1.0 viria somente em mais um ano.

### Linha do Tempo

1995 – A Borland lança Delphi 1.0

1997 – Lançada oficialmente a especificação 1.0 de UML. Revisão da UML pela OMG gerando a especificação 1.1.

1998 - O RUP 5.0, baseado em UML, é formado pela união das melhores metodologias de projeto de sis temas (*ObjectoryProcess*: versões 1.0 a 3.8 (1987 a 1995) que posteriormente integrou-se à abordagem da Rational (*Rational Objectory Process* – ROP).

2001 – Revisão da UML gerando a especificação 1.4 de UML.

2004 – Lançada a especificação 2.0 de UML.

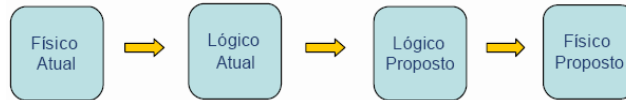
### Métodos de Desenvolvimento de Software

A análise auxilia na comunicação entre as pessoas envolvidas no processo de desenvolvimento do software, no gerenciamento da complexidade e na redução dos custos de desenvolvimento.

Técnica	Foco	Abordagem
Análise Estruturada	Processos e Dados	<i>Top-Down</i>
Análise Essencial	Controles, Processos e Dados	<i>Middle-Out</i>
Análise Orientada ao Objetos	Dados, Controles e Processos	Definição de Objetos

### Análise Estruturada

Descreve os modelos a partir de refinamentos e evoluções:



Enfatiza as funções, ênfase principal nos processos.

Utiliza as ferramentas:

- Diagrama de Fluxo de Dados
- Dicionário de Dados
- Especificação da Lógica de Processos

A análise estruturada clássica não modela o comportamento temporal, nem complexos relacionamentos de dados.

### Análise Estruturada

A seguir serão descritas ferramentas úteis no processo de análise estruturada:

- Diagramas de fluxo de dados – DFD
- Dicionário de dados – DD
- Especificação de Processo – EP
- Português estruturado – PE
- Tabelas de decisão – TD
- Árvores de decisão – AD

## Análise Estruturada

### DFD – Diagrama de Fluxos de Dados

- Utilizado para especificar o modelo dos processos do sistema, ou seja, como a informação é criada, armazenada e utilizada
- A ênfase está nos processos
- Devem ser simples para que possa ser compreendido facilmente pelo analista, pelo cliente e pela equipe de desenvolvimento
- Os componentes do DFD são: processo, fluxo de dados, entidade externa e depósito de dados

## Análise Estruturada

### DFD – Diagrama de Fluxos de Dados

- Deve ser utilizado quando for necessário conhecer:
  - a funcionalidade do sistema
  - a forma como tudo acontece
  - a forma como os elementos se inter-relacionam e se transformam
- A sua construção é realizada em níveis:
  - Nível 0: *Diagrama de Contexto*
  - Nível 1: Explosão do nível 0, contendo *Título e Numeração dos Processos*: 1; 2; 3;...
  - Nível 2: Explosão do nível 1, contendo *Título e Numeração dose Processos*: 1.1; 1.2; 1.3;... 2.1; 2.2;...
  - E assim por diante.

## Análise Estruturada

### DFD – Processo

- Representa o elemento que transforma *entradas* em *saídas*
- O processo sempre tem um nome que deve descrever o melhor possível a funcionalidade que será desempenhada pelo processo
- O nome do processo, geralmente, é formado por um verbo (Calcular) e um objeto (Média dos Alunos)
- Formas de representação:



## Análise Estruturada

### DFD – Entidade Externa

- Representa os elementos que não são do sistema, mas interagem com ele, inserindo ou recebendo dados
- Exemplo: pessoas, usuários, empresas, departamentos de uma empresa, outros sistemas
- Formas de representação:



## Análise Estruturada

### DFD – Depósito de Dados

- Utilizado para modelar um conjunto de dados em repouso e que precisam ser armazenados fisicamente
- Pode se tornar uma ou mais tabelas da base de dados ou um arquivo na versão final do sistema
- O nome de um depósito de dados geralmente está no plural e deve ser o mais representativo possível quanto aos dados que armazena
- Formas de representação:

Notas de Alunos

Notas de Alunos

D1 Notas de Alunos

## Análise Estruturada

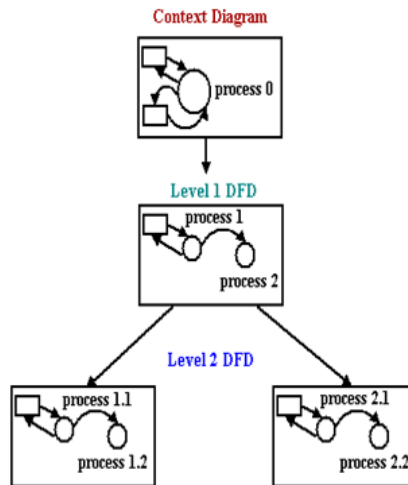
### DFD – Fluxo de Dados

- Representa o movimento dos dados (ou conjuntos de dados) dentro do sistema
- O fluxo de dados deve ter um sentido e pode ser:
  - de um processo para outro processo
  - da entidade externa para o processo
  - do processo para a entidade externa
  - do processo para o depósito de dados
  - do depósito de dados para o processo
- O fluxo de dados deve ter um nome
- Forma de representação:

Notas\_de\_Aluno →

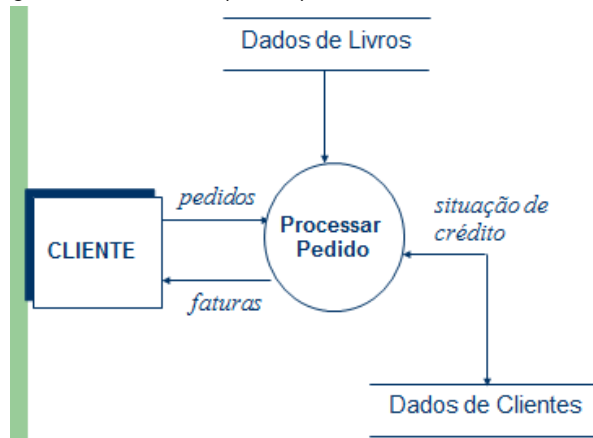


### Análise Estruturada – DFD (visão de explosões/detalhamentos)



### Análise Estruturada

#### DFD – Diagrama de Contexto (nível 0)



## Análise Estruturada

DFD – Nível 1 como explosão do diagrama de contexto



## Análise Estruturada

DFD – Nível 2 como explosão do Nível 1  
Processo Verificar Validade do Pedido



## Análise Estruturada

### *DD - Dicionário de dados*

É uma lista organizada dos elementos do sistema, com definições precisas e rigorosas, de forma que o usuário e o analista de sistemas tenham uma compreensão comum dos fluxos de entrada, dos fluxos de saída, dos componentes dos depósitos de dados e dos cálculos intermediários.

Na descrição do dicionário é utilizada uma gramática, quase formal, para descrever o conteúdo dos elementos definidos durante a análise estruturada.

Pode ser feito através de uma ferramenta CASE.

## Análise Estruturada

### *DD - Dicionário de dados*

O Dicionário de Dados:

- Descreve o significado dos fluxos e dos depósitos de dados
- Detalha a composição dos pacotes agregados de dados que se movimentam pelos fluxos
- Mostra a composição dos pacotes de dados nos depósitos
- Especifica os “valores” e “unidades” de partes elementares de informações dos fluxos de dados e dos depósitos de dados
- Detalha os relacionamentos entre os depósitos especificados em um DER

## Análise Estruturada

### *DD - Dicionário de dados*

#### Notação:

= : é composto de

+ : e

( ) : opcional

{ } : iteração (zero ou mais ocorrências)

[ ] : escolha uma das alternativas

\* \* : comentário

@ : identificador (campo chave) de um depósito

| : separa as alternativas na construção [ ]

## Análise Estruturada

### *DD - Dicionário de dados*

#### Exemplo de descrição para um elemento:

nome do aluno – faz parte do fluxo de dados “inf.alunos”:

nome do aluno = título + prim -nome + (nome-intermediário) + ult-nome

título = [ Sr. | Srta. | Sra. | vazio ]

prim -nome = { qualquer-caracter-válido }

nome-intermediário = { qualquer-caracter-válido }

ult-nome = { qualquer-caracter-válido }

qualquer-caracter-válido = [ A-Z | a-z | espaço em branco ]

## Análise Estruturada

### DD - Dicionário de dados - Exemplo

cliente-pedido = \* cliente que encaminhou o pedido \*  
nome-cliente + [endereço-cliente | telefone-cliente]

item-pedido = \* item do pedido de um cliente \* nome-item + quantidade-item

pedido-livro = \* pedido de livros dos clientes da livraria \*  
cliente-pedido + 1{item-pedido}

pedido-livro = \* pedido de livros dos clientes da livraria\* *(especificação ruim)*  
nome-cliente + endereço-cliente + (telefone-cliente) + 1{nome-item + quantidade-item}

pedido-livro = \*pedido de livros dos clientes da livrara\* *(especificação boa)*  
cliente-pedido + 1 {item-pedido}

## Análise Estruturada

### DD - Dicionário de dados - Exemplo

Quando dois ou mais dados possuem a mesma composição, suas definições podem ser feitas por meio de referências cruzadas.

Exemplo:

endereço = \*\*  
rua + complemento + bairro + cep + estado

endereço-cliente = \*\*  
endereço

endereço-fornecedor = \*\*  
endereço

## Análise Estruturada

### DD - Dicionário de dados - Exemplo

As mensagens transportadas pelos fluxos de dados, quando definidas no dicionário de dados, devem ser escritas entre aspas.

Exemplos:

pedido-inválido = \* pedido de cliente rejeitado pela livraria\*  
"livro inexistente"

situação-aluno = \* aproveitamento do aluno na disciplina\*  
[nota-aluno | "reprovado"]

livro-indisponível = \* livro indisponível para empréstimo\*  
nome-livro + ["emprestado" | "reservado"]

## Análise Estruturada

### EP - Especificação de Processo

É usada para descrever o que ocorre dentro de todos os processos que aparecem no nível mais baixo do DFD.

Também se utiliza o termo Mini-especificação.

Algumas ferramentas para especificação de processos são:

- Português-Estruturado
- Tabelas de Decisão
- Árvores de Decisão

A especificação de processos deve ser expressa de uma forma que possa ser verificada pelo usuário e pelo analista de sistemas.

#### PE - Português estruturado

É utilizado para descrever, passo a passo, a execução de alguma tarefa, utilizando uma linguagem natural, por exemplo, o português. Usa três estruturas de controle da programação estruturada:

- seqüência
- seleção: se... então... senão, escolha....
- repetição: enquanto... faça, para .... faça, repita... até.

## Análise Estruturada

### PE - Português estruturado

Exemplo: Verifica a situação do aluno na disciplina

lê nota1, nota2, nota3, nota4, notaSeminário

notaRelatório média =  $(\text{nota1} + \text{nota2} + \text{nota3} + \text{nota4} * 3) / 6$

se  $((m \text{ média} \geq 6) \text{ e } (\text{notaSeminário} \geq 5) \text{ e } (\text{notaRelatório} \geq 5))$  então  
imprima "aluno aprovado"

senão se  $((m \text{ média} \geq 6) \text{ e } (\text{notaSeminário} \geq 5))$  ou  $((m \text{ média} \geq 6) \text{ e } (\text{notaRelatório} \geq 5))$  ou  $((\text{notaSeminário} \geq 5) \text{ e } (\text{notaRelatório} \geq 5))$  então  
imprima "aluno em exame"

senão  
imprima "aluno reprovado"

## Análise Estruturada

### TD - Tabelas de decisão

São indicadas para especificar quais ações serão executadas a partir de um conjunto de valores condicionais.

É composta de:

- conjunto de regras
- linhas de condições
- linhas de ações

Formato Geral:

	TABELA-CASACO	R1	R2	R3
C1	chovendo	Y	Y	N
C2	frio	Y	N	Y
A1	usar capa forrada	X		
A2	usar capa sem forro		X	
A3	usar pullover de lã			X

## Análise Estruturada

### TD - Tabelas de decisão

Etapas para criação de uma Tabela de Decisão:

- Identifique todas as condições ou variáveis/valores.
- Calcule o número de combinações de condições. Se há N variáveis binárias o número de combinações é  $2^N$ .
- Identifique cada ação correspondente.
- Crie a tabela de decisão vazia, relacionando todas as condições e ações no lado esquerdo e numerando as regras no alto da tabela.
- Relacione todas as combinações de condições.
- Examine cada regra e identifique as ações adequadas a realizar.
- Identifique todas as omissões, contradições e ambigüidades.
- Discuta as omissões, contradições e ambigüidades com o usuário.

## Análise Estruturada: Exemplo

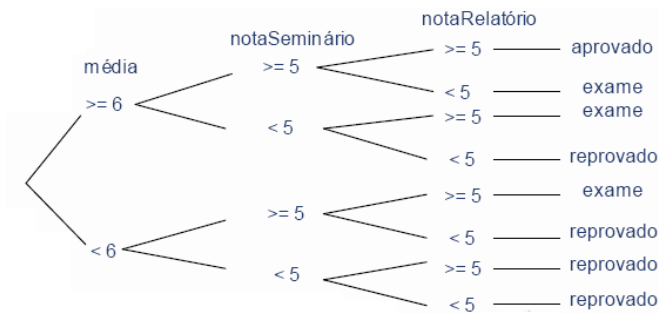
		1	2	3	4	5	6	7	8
C1:	media $\geq 6$	S	S	S	S	N	N	N	N
C2:	notaSeminário $\geq 3$	S	S	N	N	S	S	N	N
C3:	notaRelatório $\geq 3$	S	N	S	N	S	N	S	N
A1:	aluno aprovado	X							
A2:	aluno em exame		X	X		X			
A3:	aluno reprovado				X		X	X	X



## Análise Estruturada

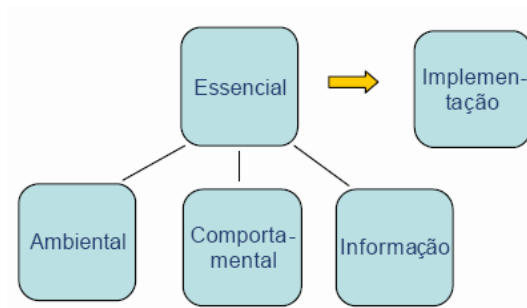
### AD - Árvores de decisão

É uma outra forma de representação para uma tabela de decisão.



## Análise Essencial

Descreve os modelos:



### Análise Essencial

É uma evolução da Análise Estruturada por adicionar a preocupação com o controle e faz uso de suas ferramentas para representar os requisitos do sistema.

A Análise Essencial é um composto de três elementos:

- descrição das características do sistema
- identificação das atividades essenciais do sistema
- classificação das restrições de implementação

Na análise essencial um requisito verdadeiro é uma característica ou capacidade que um sistema deve ter para cumprir sua finalidade, independente de como ele é implementado.

Os requisitos verdadeiros constituem a essência do sistema.

### Análise Essencial

*Modelo essencial* é construído sem considerar restrições de implementação (assume uma tecnologia perfeita) – essência do sistema.

O modelo essencial é formado pelo:

- *Modelo Ambiental* que define a fronteira entre o sistema e o ambiente
- *Modelo Comportamental* que descreve o comportamento interno do sistema.
- *Modelo de Informação* que modela os dados necessários às atividades essenciais do sistema
- *Modelo de Implementação* que é uma extensão do modelo essencial com restrições de implementação.

## Análise Essencial

### *Modelo Essencial - Modelo Ambiental*

#### **Diagrama de Contexto**

Define as interfaces entre o sistema e o ambiente. São identificadas informações externas e as produzidas como saída.

Visão geral das características do sistema:

- Entidades externas
- Dados recebidos e processados
- Dados produzidos e exteriorizados

#### **Lista de Eventos**

Identifica os eventos que ocorrem no ambiente e como o sistema deve reagir. Lista de eventos:

- Evento
- Estímulo
- Resposta

## Análise Essencial

### *Modelo Essencial - Modelo Comportamental*

Mostra o comportamento interno do sistema.

Aborda as atividades essenciais do sistema modeladas em um DFD.

Cada atividade corresponde a um evento.

Uma mini-especificação em português estruturado detalha o comportamento e valida a entradas e saídas das atividades essenciais (Especificação de processos – EP).

O Diagrama de Transição de Estado (DTE) mostra as transformações de controle do sistema no tempo.

Usa Dicionário de Dados.

## Análise Essencial

### *Modelo Essencial - Modelo de Informação*

Representa os dados necessários ao sistema.

Define os depósitos de dados que constituem a memória essencial do sistema e auxilia no levantamento da lista de eventos

As ferramentas utilizadas são:

- Diagrama de Entidade e Relacionamento
  - Deriva da lista de eventos
  - Representa a estrutura estática dos dados
- Dicionário de Dados

## Análise Essencial

### *Modelo de Implementação*

Considera restrições tais como processador, capacidade de armazenamento, Sistema Gerenciador de Banco de Dados (SGBD), programação, recursos humanos e financeiros, etc.

Aborda os seguintes pontos:

- Modelo Lógico de Dados
- Características de Processamento
- Interface Homem Máquina

## Análise Essencial

### *Modelo de Implementação – Modelo Lógico de Dados*

É voltado para as características de implementação do Sistema Gerenciador de Banco de Dados (SGBD).

Por exemplo:

Classe de entidades: Aluno

id_aluno	nome	curso	cpf
000001	José da Silva	Gastronomia	10010010011
000002	Mariana Santos	Telecomunicações	10020010012

## Análise Essencial

### *Modelo de Implementação – Características de Processamento*

Analisa, principalmente, os elementos:

- Tipo de processamento: Manual ou Automático.
- Modo de processamento: Lote, On-line, Tempo Real.
- A frequência de execução.
- As atividades adicionais: Introdução dos Dados e Correção de Erros.

### *Modelo de Implementação – Interface Homem Máquina*

Neste ponto são especificados:

- Os dispositivos de Entrada e Saída
- Formato de todas as entradas originadas pelas Entidades Externas
- Formato de todas as Saídas destinadas às Entidades Externas
- Seqüência e Temporização das Entradas e Saídas.

## Análise Essencial

### DER - Diagrama de Entidade-Relacionamento

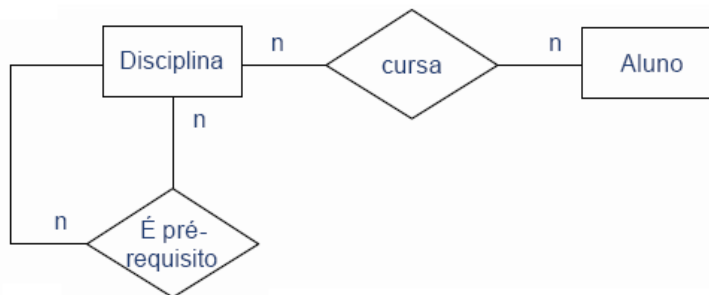
Foi originalmente proposto por Peter Chen (1976) e é o modelo mais utilizado no processo de projeto de base de dados relacional.

Os seus componentes são:

- Entidade: representa um objeto ou conceito existente no mundo real, como um aluno, uma disciplina ou as notas do aluno.
- Relacionamentos: associação entre conjuntos de dados ou entidades, podendo ser "1:1", "1:n" ou "n:n".
- Atributos: representa alguma característica de uma entidade ou de um relacionamento. Conceito de campo de dados.

Uma propriedade importante de um relacionamento é a *cardinalidade* que indica quantas ocorrências de uma entidade podem estar associadas a uma determinada ocorrência através do relacionamento.

### Análise Essencial: Exemplo DER

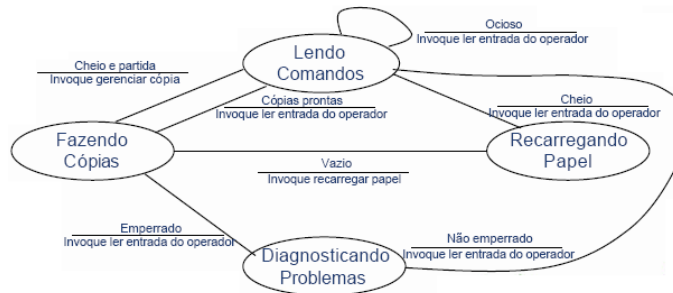


## Análise Essencial

### DTE - Diagrama de Transição de Estado

Representa o comportamento de um sistema, descrevendo seus estados e os eventos que fazem com que o sistema mude de estado.

Indica também quais ações são executadas como consequência de um dado evento. Diagrama de Transição de Estados para *software* de fotocopadora, extraído de "Engenharia de Software, Pressman R., 5aEd.,



## Análise Orientada a Objetos

O objetivo é encontrar objetos, organizá-los, descrever como interagem entre si através da troca de mensagens e definir as operações relativas aos seus comportamentos.

Concentra-se nos aspectos essenciais do objeto sem detalhamento, focando em suas características e no que ele faz.

Combina estrutura (dados) e comportamento (funções) em um único objeto.

Enfatiza a estrutura de objetos e não as estruturas de funções, ou seja, o que o objeto é e não como ele é utilizado.

Compartilha elementos estruturais e de comportamento com objetos de níveis inferiores.

### Análise Orientada a Objetos

Mudança do enfoque das funções para os dados.

Análise mais próxima da realidade. O mundo real é composto por objetos.

Objetos como entidades do mundo real.

Facilidade na comunicação com o usuário.

Objetos com estrutura e comportamento e que se comunicam.

*"Um sistema construído usando um método Orientado a Objetos é aquele cujos componentes são partes encapsuladas de dados e funções, que podem herdar atributos e comportamentos de outros componentes da mesma natureza, e cujos componentes comunicam -se entre si por meio de mensagens."*  
(Edward Yourdon)

### Análise Orientada a Objetos

Com o rápido crescimento da Orientação a Objetos surgiram várias metodologias.

Algumas metodologias, desenvolvidas entre 1898-1994, foram:

- Wirfs-Brock de Rebecca Wirfs-Brock, Wilkerson e Weiner
- BOOCH de Grady Booch
- COAD/YOURDON de Coad-Yourdon
- OOSE de Jacobson
- OMT de Rumbaugh



### Análise Orientada a Objetos

A partir do grande uso do método de Booch e do OMT, os autores juntaram forças para fazer um método unificado, com uma linguagem padrão.

Posteriormente Jacobson juntou-se a equipe.

Linguagem de modelagem proposta por Booch, Rumbaugh e Jacobson UML (*Unified Modeling Language*).

Padronizada pela OMG (*Object Management Group*).

Conta atualmente com o apoio de vários autores e várias empresas.

### Análise Orientada a Objetos

Os principais diagramas da UML são:

- Fase de Análise
  - Diagrama de casos de uso
- Fase de Projeto
  - Diagrama de classes
  - Diagramas de sequência
  - Diagramas de atividades e estado
  - Diagrama de componentes
  - Diagrama de implantação

O RUP (*Rational Unified Process*) foi proposto como uma metodologia para desenvolvimento de sistemas, orientada a objetos, utilizando UML.

### Análise Orientada a Objetos

Algumas possíveis vantagens do desenvolvimento orientado a objetos são:

- Melhor reutilização de código e de projeto
- Maior facilidade de manutenção
- Melhora a comunicação entre os usuários e os desenvolvedores do sistema
- Maior capacidade de abstração e encapsulamento

Algumas dificuldades são:

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

### Análise Orientada a Objetos

Nos métodos de análise estruturada e essencial, o comportamento do sistema e seus dados são considerados separadamente.

Na orientação a objetos, comportamento e dados são integrados, encapsulando detalhes internos de um objeto.

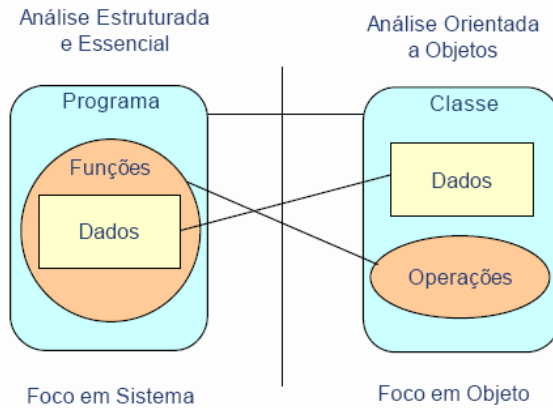
Análise Estruturada  
e Essencial

Conjunto de programas  
que executam processos  
sobre dados

Análise Orientada  
a Objetos

Conjunto de elementos  
que tem características  
e comportamentos  
próprios

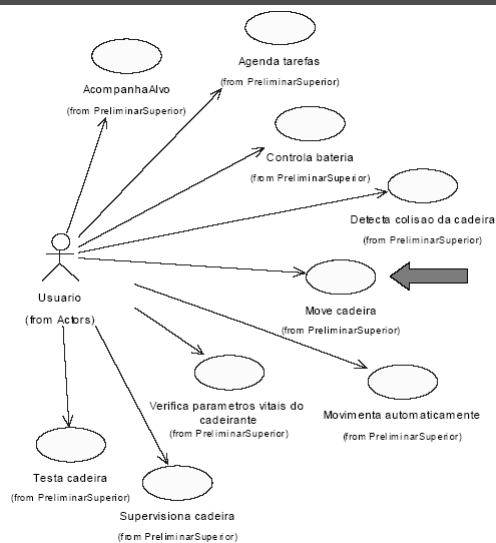
## Análise Orientada a Objetos



## Análise Orientada a Objetos

### Diagrama de Caso de Uso

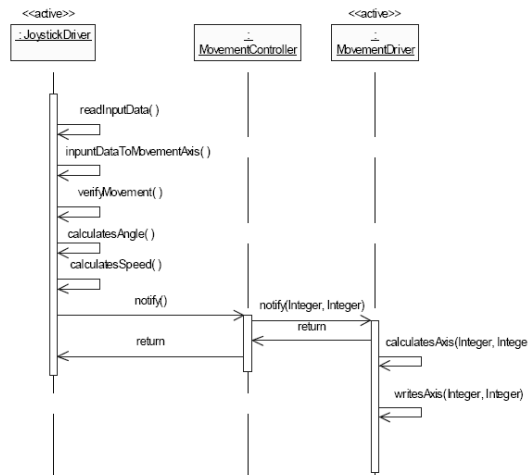
Fonte: <http://www.das.ufsc.br/~lbecker/DAS9002/uml.pdf>



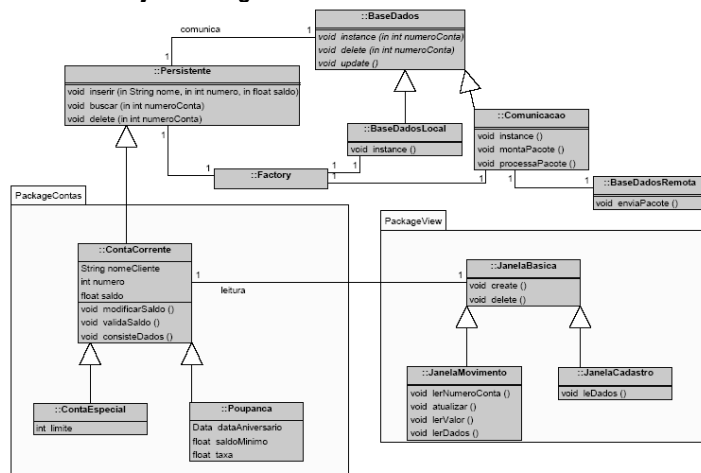
## Análise Orientada a Objetos

### Diagrama de Sequência

Fonte: <http://www.das.ufsc.br/~ibecker/DAS9002/uml.pdf>



## Análise Orientada a Objetos: Diagrama de Classes



Fonte: <http://www.das.ufsc.br/~ibecker/DAS9002/uml.pdf>

## ☛ Material de apoio:

### Bibliografia Básica

PRESSMAN, R. Engenharia de software. Rio de Janeiro: MacGraw-Hill, 2006.  
SOMMERVILLE, I. Engenharia de software. 8. ed. São Paulo: Addison Wesley, 2007.

### Bibliografia Complementar

PFLEEGER, S.L., et al, "Software Engineering", Prentice Hall, 2005, 3rd edition.  
IEEE Computer Society Real-World Software problems: A Self-Study Guide for Today's Software Professional, Wiley-IEEE Computer Society Press, 2006.  
Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, 2004. Disponível em <http://swebok.org>.

Booch, G., Rumbaugh, J. e Jacobson, I., *The Unified Modeling language User Guide*, Addison-Wesley, 1999.

Wazlawick, Raul Sidney, *Análise e Projeto de Sistemas de Informação Orientados a Objetos*, Ed. Campus, Série Campus/SBC, 2004.

FOWLER, Martin. *UML Essencial: um breve guia para a linguagem padrão de modelagem de objetos*. Bookman, 2005.

Rumbaugh, James, e outros, *Modelagem e Projetos Baseados em Objetos*, Editora Campus, 1994.

Coad, Peter e Yourdon, Edward, *Análise Baseada em Objetos*, Editora Campus, 1992.

Coad, Peter e Yourdon, Edward, *Projeto Baseado em Objetos*, Editora Campus, 1993.

Gane, Chris e Sarson, Trish, *Análise Estruturada de Sistemas*, Ed. LTC, 1983.

Davis, W. S., *Análise e Projetos de Sistemas: Uma abordagem Estruturada*, Ed. LTC, 1987.

DeMarco, T., *Análise Estruturada e Especificação de Sistema*, Ed. Campus, 1989.

Yourdon, E. *Análise Estruturada Moderna*, Ed. Campus, 1990.

McMenamim, S. M. e Palmer, J. F., *Análise Essencial de Sistemas*, Editora McGraw-Hill, Ltda., 1991.

Gorender, S., Santana, M., *Histórico das Metodologias de Desenvolvimento de Sistemas*, [http://www.lasid.ufba.br/easd/1\\_Apresenta\\_oHistoricoMetodologias\\_I.ppt](http://www.lasid.ufba.br/easd/1_Apresenta_oHistoricoMetodologias_I.ppt), junho, 2007.