

Análise e Projeto de Desenvolvimento de Software EAJ6009

Prof. Dr. Josenalde Barbosa de Oliveira

josenalde.oliveira@ufrn.br

Aulas: 35M45

<https://github.com/josenalde/apds>

Relembrando 2022.1

Programação Orientada a Objetos, Classes e Objetos, Métodos, Construtores
Encapsulamento, Composição e Agregação, Herança, Polimorfismo, Classes
Abstratas, Interfaces, Tratamento de Exceções

Ementa

1. Análise e Projeto Orientados a Objetos

1. Introdução ao Processo de Desenvolvimento de Software (pds)

- a. Principais atividades, participantes do processo, fases do PDS, Levantamento de Requisitos, Tipos de requisitos, Documento de requisitos, Estratégias para levantamento dos requisitos

2. Análise Orientada a Objetos

- a. Modelagem de casos de uso,
- b. Modelagem de classes conceituais (diagrama de classes e diagrama de objetos)

3. Projeto Orientado a Objetos

- a. Arquitetura do sistema
- b. Modelagem de classes de projeto

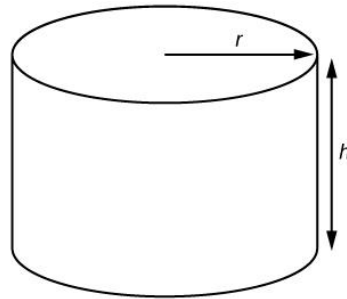
Datas planejadas avaliações

1. Introdução ao Processo de Desenvolvimento de Software

- No paradigma estruturado, tem-se a execução sequencial de instruções, com variáveis alocadas em determinadas posições de memória, com instruções de atribuição de valores a estas variáveis
- Pode-se ver um programa como uma hierarquia de “subprogramas”, tendo cada subprograma funções e variáveis que juntamente resolvem o problema global
- Descreve-se as ações para se chegar à resposta:

➤ Exemplo: cálculo do volume de um **cilindro** circular numa determinada **janela** de interação com o usuário

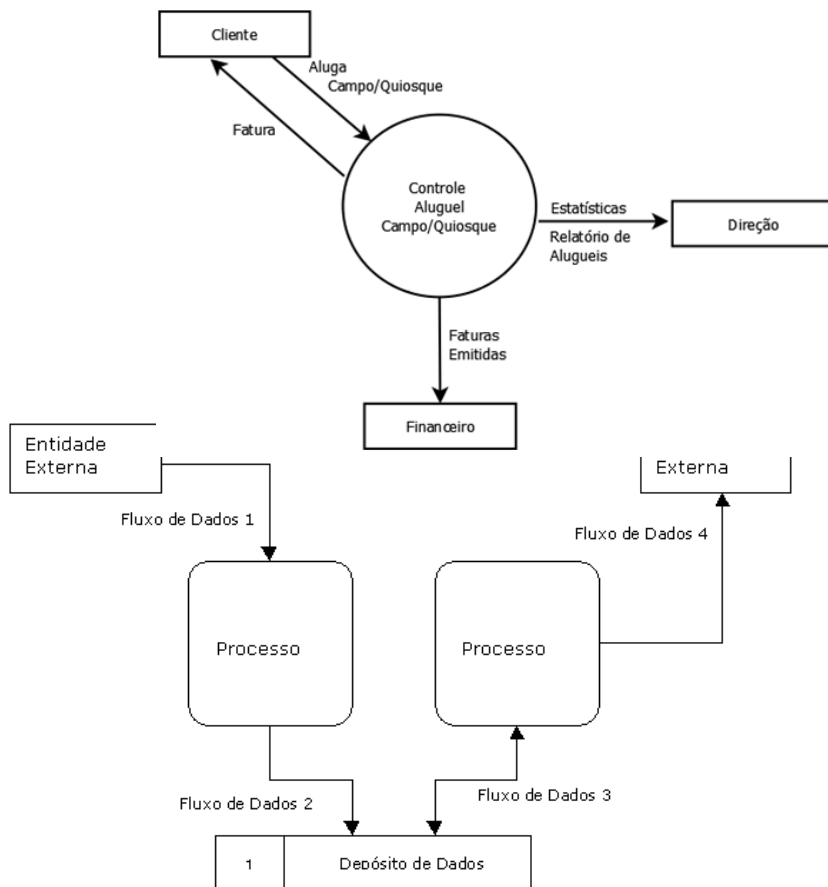
1. Ler altura (h)
2. Ler raio da base (r)
3. Calcular volume
4. Exibir volume



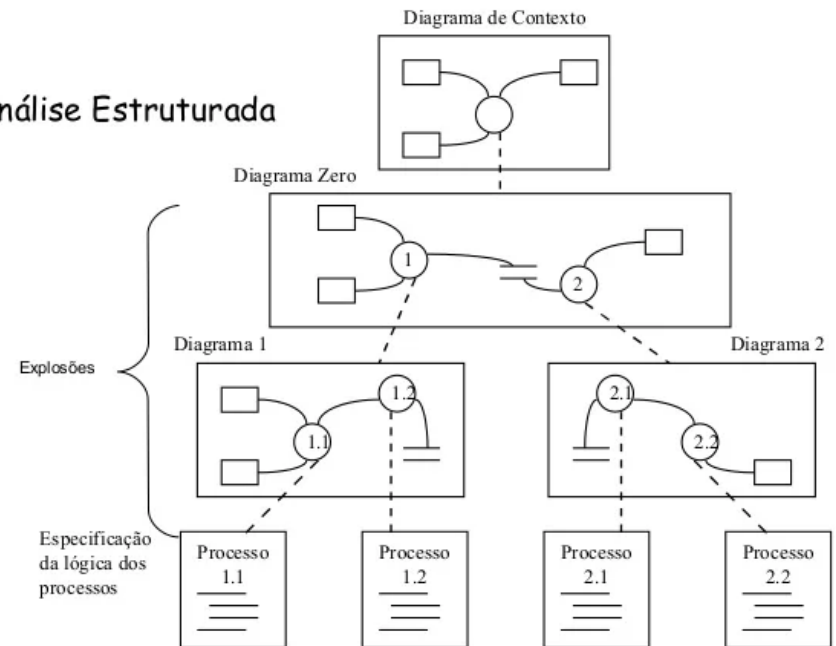
FOCO NA FUNCIONALIDADE DAS PARTES, LOGO ANÁLISE E PROJETO SEGUE ESTE PARADIGMA

1. Introdução ao Processo de Desenvolvimento de Software

► Instrumentos/Ferramentas comuns à análise estruturada de software: **Diagrama de Fluxo de Dados (DFD)**, Diagrama de Entidade Relacionamento (DER), Dicionário de Dados (DD)

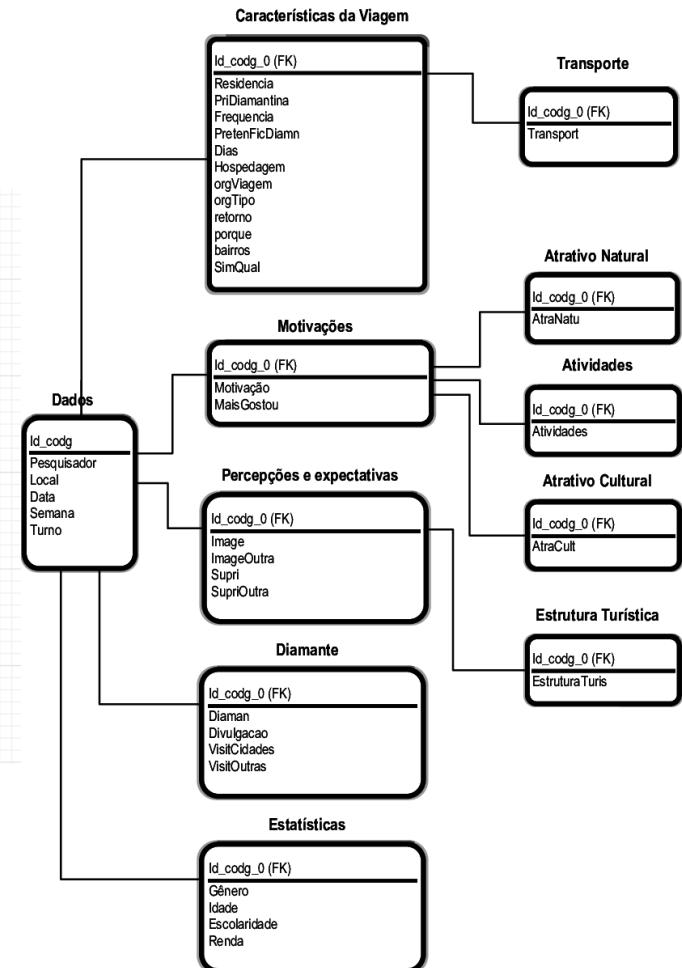
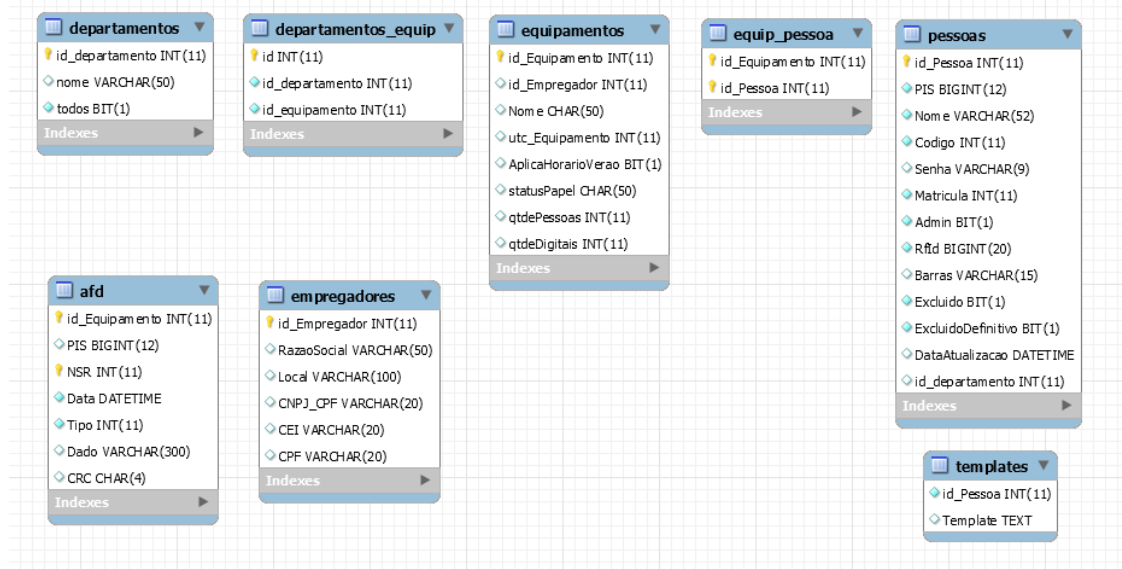


Análise Estruturada



1. Introdução ao Processo de Desenvolvimento de Software

► Instrumentos/Ferramentas comuns à análise estruturada de software: Diagrama de Fluxo de Dados (DFD), **Diagrama de Entidade Relacionamento (DER)**, Dicionário de Dados (DD)



1. Introdução ao Processo de Desenvolvimento de Software

► Instrumentos/Ferramentas comuns à análise estruturada de software: Diagrama de Fluxo de Dados (DFD), Diagrama de Entidade Relacionamento (DER), **Dicionário de Dados (DD)**

Feição: Árvore		
Representação: Ponto		
Atributo	Tipo de dado	Descrição
ID árvore	Alfanumérico	Chave de identificação da árvore
Nome_pop	Texto (80)	Nomes populares pela qual a espécie é conhecida
Nome_cient	Texto (30)	Nome científico atribuído a espécie
Família	Texto (20)	Família a qual a espécie pertence
Morfo	Texto (100)	Descrição morfológica da espécie (altura, espessura do tronco, folhas, ramos, época de floração e frutificação)
Ramificação	Boolean	Tipo? Simpodial/monopodial
CoordX	Real/Double	Valor da coordenada E (Sistema UTM)
CoordY	Real/Double	Valor da coordenada N (Sistema UTM)
Logradouro	Texto (lista)	Nome completo do Logradouro, com abreviação somente no tipo (Ex: R., Av., Trav., Rod.)
Calçada	Real/Double	Largura da calçada em metros
Uso_lote	Texto (lista)	Uso do lote em frente a amostra (desocupado, comercial, residencial, religioso, industrial, escolar, hospitalar, cultural, outros serviços)
DAP	Real/Double	Valor em centímetros
Fuste	Boolean	Boas condições? Sim/Não
Copa	Boolean	Boas condições? Sim/Não
Sujeira	Boolean	Produz muita sujeira? Sim/Não
FitoSanit	Texto (30)	Tipo de condições fitossanitárias e informações adicionais (se o ataque é de inseto qual o tipo ou doença)
Fios eletricos	Boolean	Interfere? Sim/Não
Poste iluminação	Boolean	Interfere? Sim/Não
Terço_inferior	Boolean	Interfere? Sim/Não
Poda_apar	Boolean	A árvore apresenta poda aparente? Sim/Não
Raiz	Texto (lista)	Calçada alterada com raiz aparente, calçada alterada sem raiz aparente, calçada intacta
Raiz_obs	Texto	Qualquer outra observação que queira incluir a respeito da raiz

Entidade: Cliente				
Atributo	Classe	Domínio	Tamanho	Descrição
Codigo_cliente	Determinante	Númerico		
Nome	Simples	Texto	50	
Telefone	Multivalorado	Texto	50	Valores sem as máscaras de entrada
Cidade	Simples	Texto	50	
data_nascimento	Simples	Data		Formato dd/mm/aaaa

1. Introdução ao Processo de Desenvolvimento de Software

➡ Retornando ao problema do volume do cilindro, no paradigma **Orientado a Objetos**, o foco está em a) identificar os objetos, b) suas propriedades/atributos e métodos (ações que podem ser executadas sobre e com os objetos), c) como se dá a interação entre os objetos

CILINDRO
JANELA

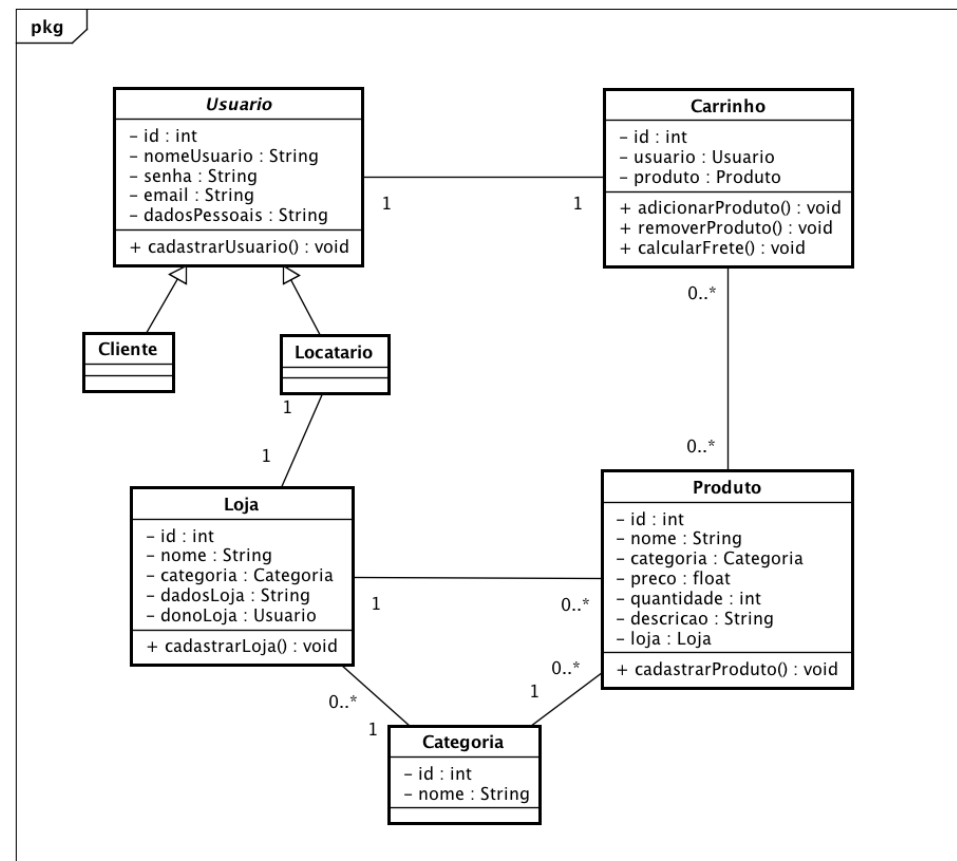
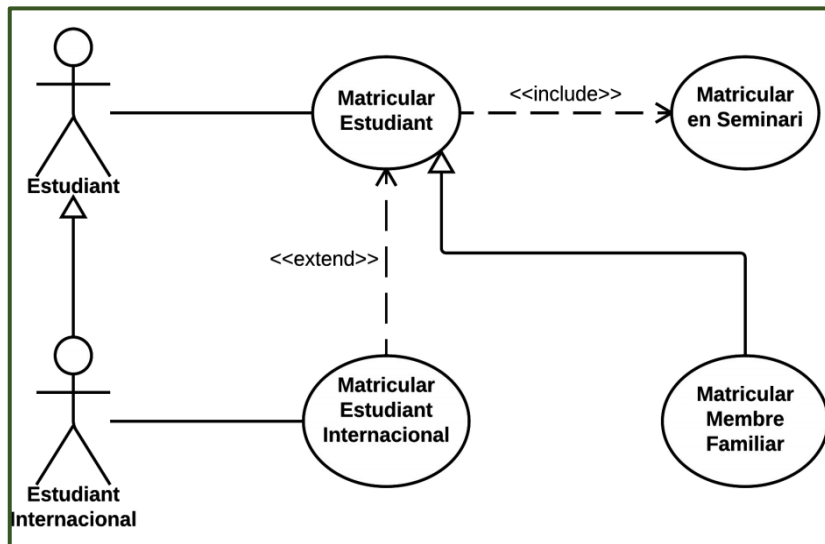
Objetos

CILINDRO: armazena e manipula altura e raio
calcula o volume

JANELA: recebe altura e raio e envia ao cilindro
solicita volume ao cilindro para exibir

1. Introdução ao Processo de Desenvolvimento de Software

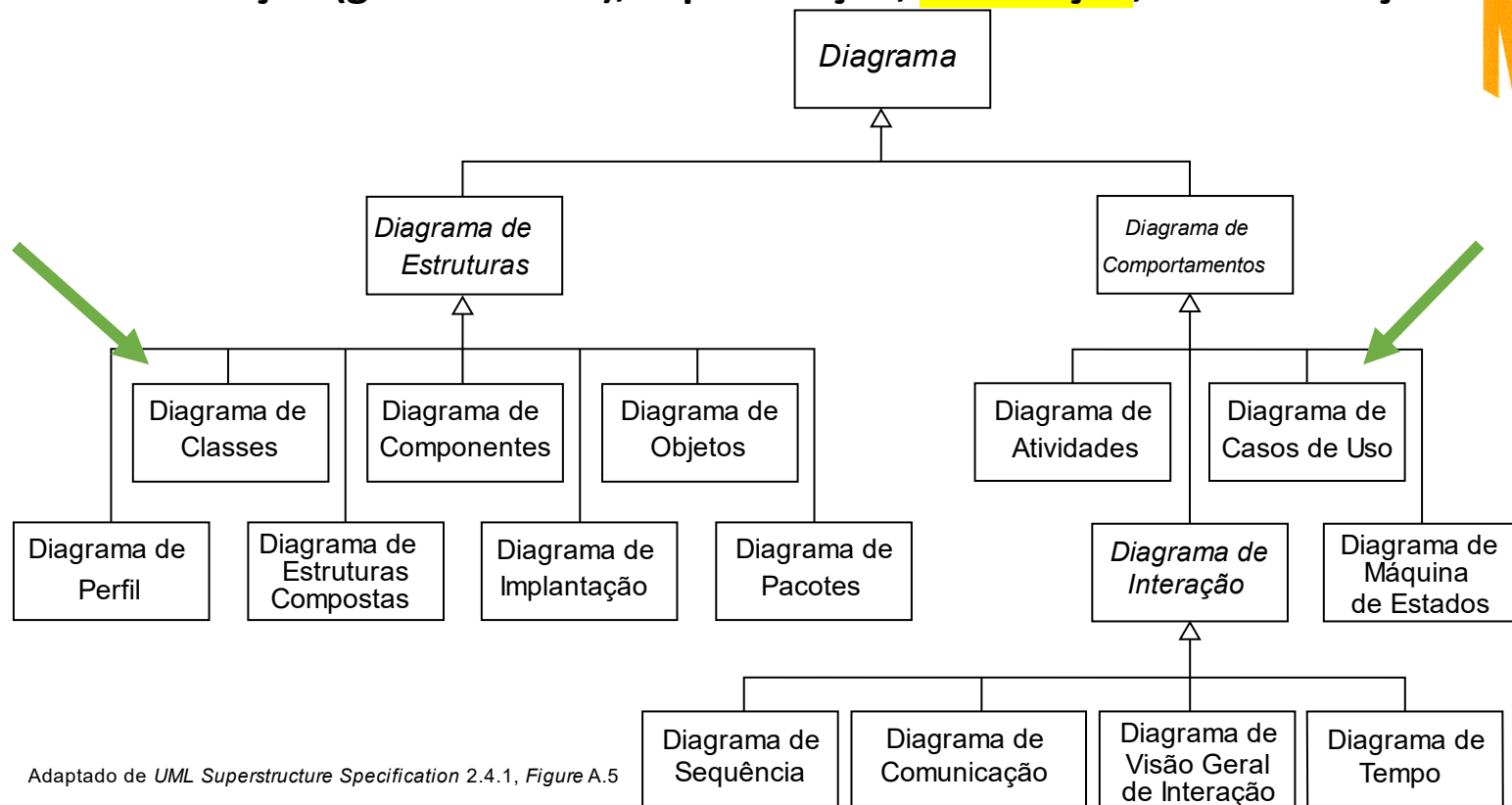
➡ No paradigma **Orientado a Objetos**, a análise e projeto basea-se em artefatos definidos em Linguagem Unificada de Modelagem (UML), sendo dois diagramas mais comuns o **Diagrama de Casos de Uso** e o **Diagrama de Classes**



1. Introdução ao Processo de Desenvolvimento de Software

► Só para ter uma ideia do conjunto de ferramentas da UML (1997..., OMG)

► **Visualização (graficamente), especificação, construção, documentação**

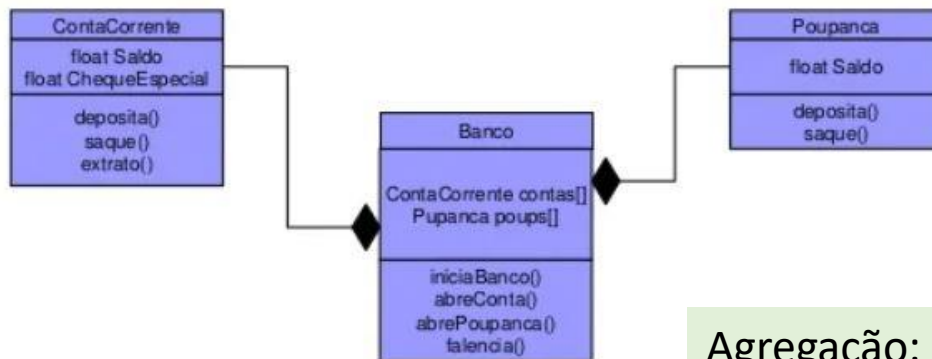


Adaptado de UML Superstructure Specification 2.4.1, Figure A.5

1. Introdução ao Processo de Desenvolvimento de Software

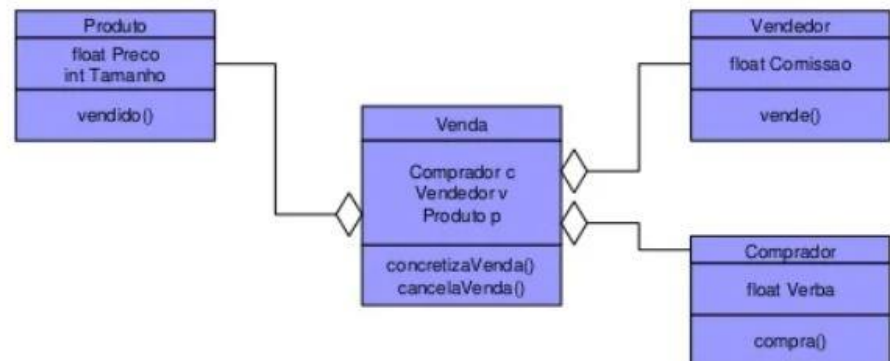
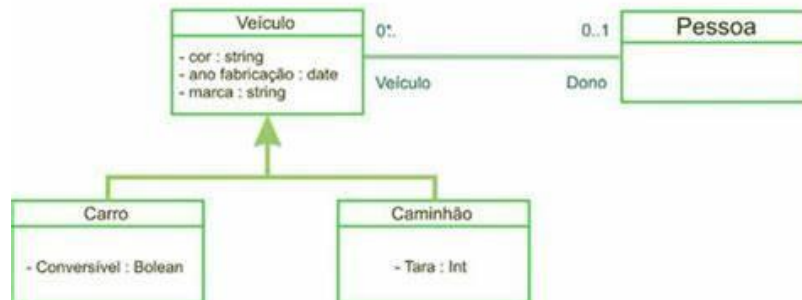
► Relembrando composições, associações e composições, que podem ser mapeadas em diagramas de classes

Composição: partes só existem se Todo (Banco) existir. Se banco for destruído, os outros são em cascata...No caso de bancos de dados relacionais, regras de integridade!



Agregação: Todo (Venda) existe se partes existem

Exemplo de Associação



1. Introdução ao Processo de Desenvolvimento de Software

- O desenvolvimento de um software é visto como um projeto de engenharia
 - **REQUISITOS, METODOLOGIA (ETAPAS, RESPONSÁVEIS), ENTREGAS, ACOMPANHAMENTO, FERRAMENTAS, DOCUMENTAÇÃO, TESTES**
- Perguntas comuns são:
 - *Por que o Sistema vai ser desenvolvido, o que pretende resolver (why?)*
 - *O que vai/deve ser feito (what)?*
 - *Quando vai ser feito (when)?*
 - *Quem é o responsável (who)?*
 - *Onde as responsabilidades estão localizadas (where)?*
 - *Como vai ser feito (how)?*
 - *Quanto vai custar (how much)?*

PDS: sequência de atividades, normalmente agrupadas em fases e tarefas, executadas de forma sistemática e uniformizada, realizadas por pessoas com responsabilidades bem definidas e que, a partir de um conjunto de entradas produzem um conjunto de saídas

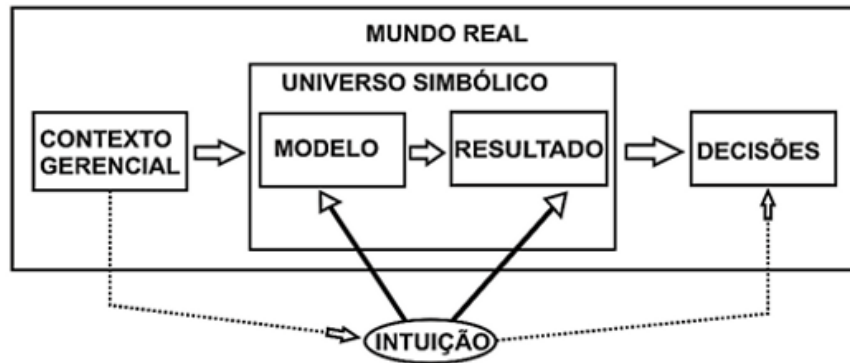
1. Introdução ao Processo de Desenvolvimento de Software

- ETAPAS

- *CONCEPÇÃO (análise): análise inclui a identificação detalhada das funcionalidades do sistema (levantamento de requisitos) e a respectiva descrição (especificação do sistema) de modo que os mesmos requisitos possam ser validados pelos usuários finais. A análise **MODELA o PROBLEMA** e **COMPREENDE** o problema. Responde: *o que deve ser feito?**
- *IMPLEMENTAÇÃO (projeto (**modelo da solução**)), desenvolvimento, testes de integração, instalação): na etapa de projeto tem-se a definição detalhada da arquitetura global da solução (módulos, tabelas, interface etc.) para que no desenvolvimento haja a codificação dos diversos componentes do sistema. Os testes são cruciais para verificação global dos objetivos e crítica pelos usuários/clientes. Na fase de instalação, denomina-se colocar o Sistema EM PRODUÇÃO. Responde: *como pode ser feito?**
- *MANUTENÇÃO: com a operação do Sistema faz-se necessário manter, seja pelo monitoramento e correção de bugs, seja pela inserção de novas funcionalidades*

1. Introdução ao Processo de Desenvolvimento de Software

- MODELO E MODELAGEM



- *Um modelo é um plano, desenho ou abstração que materializa a compreensão sobre um determinado contexto, problema ou situação; é representação (por vezes simplificada) da realidade; é o resultado do processo de modelagem*
- *Exemplo: construção de um prédio: profissionais (eng. Civis, arquitetos etc.) farão vistoria no terreno, análise do solo etc. Estes documentos servem para planejar a moradia; segue-se depois a planta, para os operários começarem o trabalho. Onde haverá uma parede? Qual o tamanho dos cômodos? A planta informará. Modelos refletem um planejamento.*

1. Introdução ao Processo de Desenvolvimento de Software



1. Introdução ao Processo de Desenvolvimento de Software

- MODELO E MODELAGEM
- *Em geral, a modelagem de sistemas tem alguns objetivos claros:*
 - *Diminuição dos custos de produção e manutenção*
 - *Aumento da escalabilidade e confiabilidade*
 - *Retorno (financeiro, pessoal etc.) obtido pelo uso do Sistema pelo cliente*
- *MODELOS devem:*
 - *Ajudar a visualizar o Sistema como ele é ou como nós gostaríamos que ele fosse*
 - *Permitir especificar a estrutura ou o comportamento*
 - *Fornecer um guia para a construção*
 - *Documentar as decisões que tomamos*

