

# Modelagem de Software



---

Prof. Dr. Ronaldo C. de Oliveira

[ronaldo.co@ufu.br](mailto:ronaldo.co@ufu.br)

[www.facom.ufu.br/~ronaldooliveira](http://www.facom.ufu.br/~ronaldooliveira)

FACOM - 2019



---

# **Modelagem de Objetos**

## **Conceitos Básicos**

# Modelagem de Objetos

## Conceitos Básicos

---

- Objeto
  - É uma entidade real ou abstrata, com características específicas de interesse para o ambiente e que está associado a processos que modificam seu estado e a um certo conjunto de atividades do ambiente de negócios;

# Modelagem de Objetos

## Conceitos Básicos

---

- Exemplos de objetos
  - Cliente (ambiente qualquer);
  - Nota fiscal (ambiente de vendas);
  - Aula, Mapa de notas (escola, faculdade);
  - Produto (ambiente de vendas ou serviços);
  - Médico, medicamento, receita, prontuário, paciente, internação (hospital).

Obs.: O nível de detalhe que se deseja ter sobre um mesmo objeto varia de ambiente para ambiente.

# Modelagem de Objetos

## Conceitos Básicos

---

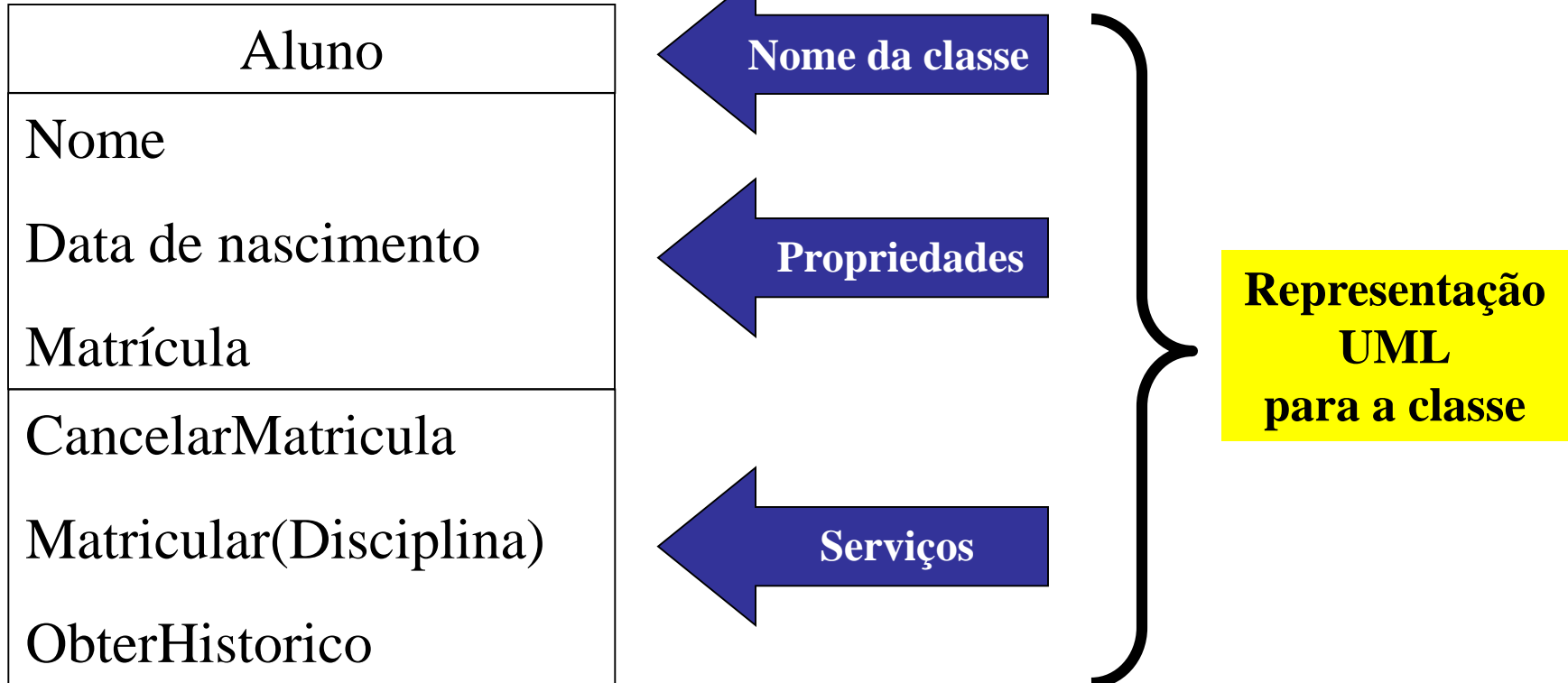
### ■ Classe

- É uma forma de categorizar um conjunto de objetos que compartilham características e comportamentos;
- Exemplo: você e seu colega são, obviamente, pessoas (objetos) diferentes, mas que no contexto deste instante ambos pertencem à uma mesma classe de pessoas: **Alunos**.
- Uma classe é representada por um nome, um conjunto de propriedades (atributos) que caracterizam esta classe e um conjunto de serviços (métodos) que manipulam os objetos desta classe.

# Modelagem de Objetos

## Conceitos Básicos

- Representação gráfica de uma Classe:

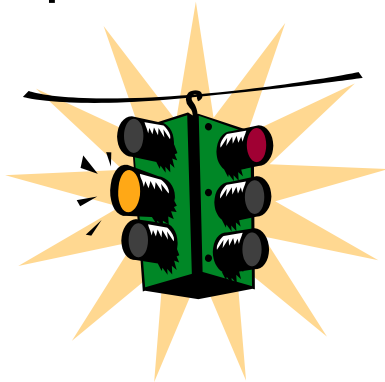


# Modelagem de Objetos

## Conceitos Básicos

### ■ Estado de um objeto

- É a situação em que se encontra um determinado objeto, sob a ótica daquilo que se deseja controlar.
- Dentro de um sistema um mesmo objeto pode possuir diversos estados.



**Estado = situação do objeto**

**Um semáforo pode conter o seguintes estados:**

**Verde**

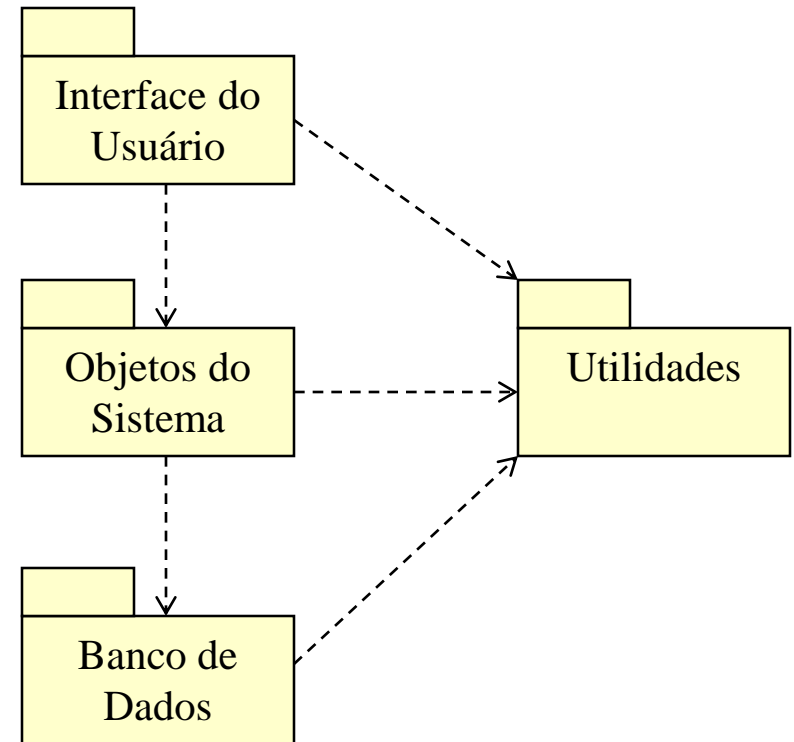
**Amarelo**

**Vermelho**

# Modelagem de Objetos

## Conceitos Básicos

- Pacotes:
  - É um mecanismo de propósito geral para organizar elementos semanticamente relacionados em grupos.



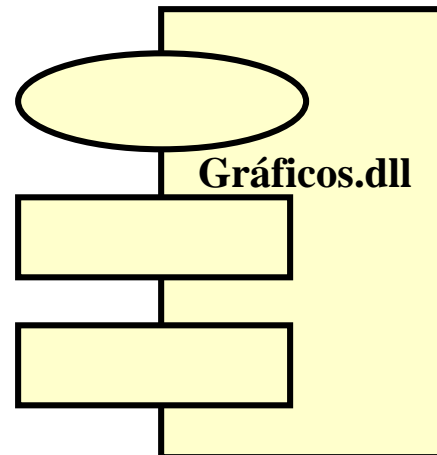
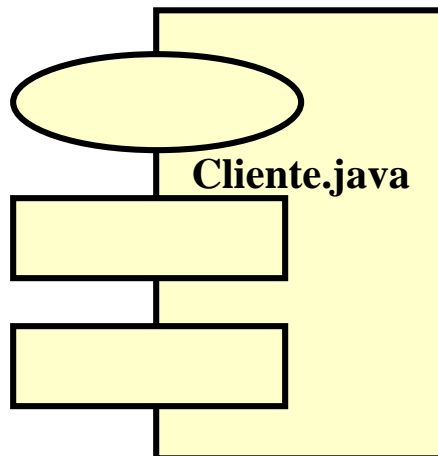


# Modelagem de Objetos

## Conceitos Básicos

- Componentes

- Um componente pode ser tanto um código em linguagem de programação como um código executável já compilado



# Modelagem de Objetos

## Conceitos Básicos

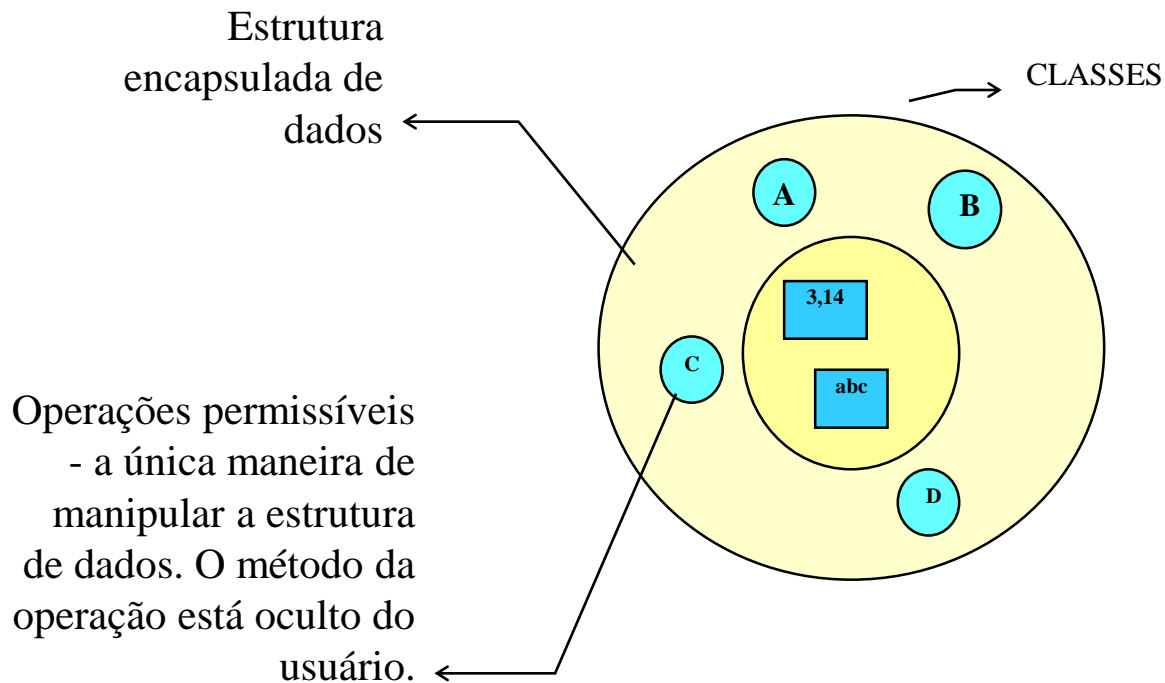
---

- Encapsulamento
  - É o resultado (ou ato) de ocultar do usuário os detalhes da implementação de um objeto, protegendo-o de adulteração. O usuário deve conhecer a classe juntamente com as suas propriedades e saber quais os métodos para manipular os objetos desta classe, mas não precisa saber como estes métodos foram implementados.

# Modelagem de Objetos

## Conceitos Básicos

### ■ Encapsulamento



# Modelagem de Objetos

## Conceitos Básicos

---

### ■ Mensagem

- É uma solicitação enviada para um, ou mais objetos, obtendo como resposta a execução de algum serviço.
- Ex.: Um objeto da classe A (emissor) envia uma mensagem para um objeto da classe B (receptor), este objeto recebe a mensagem, executa algum serviço, e retorna a resposta ao emissor da mensagem.

# Modelagem de Objetos

## Conceitos Básicos

---

### ■ Polimorfismo

- É uma das mais importantes características da orientação a objetos. O polimorfismo acontece quando uma mesma mensagem pode ser interpretada em diferentes caminhos por diferentes objetos.

Ex.: Objeto:	circulo	quadrado
Propriedades:	raio, centro	lado, altura
Serviços:	área, desenhar	área, desenhar

# Modelagem de Objetos

## Conceitos Básicos

---

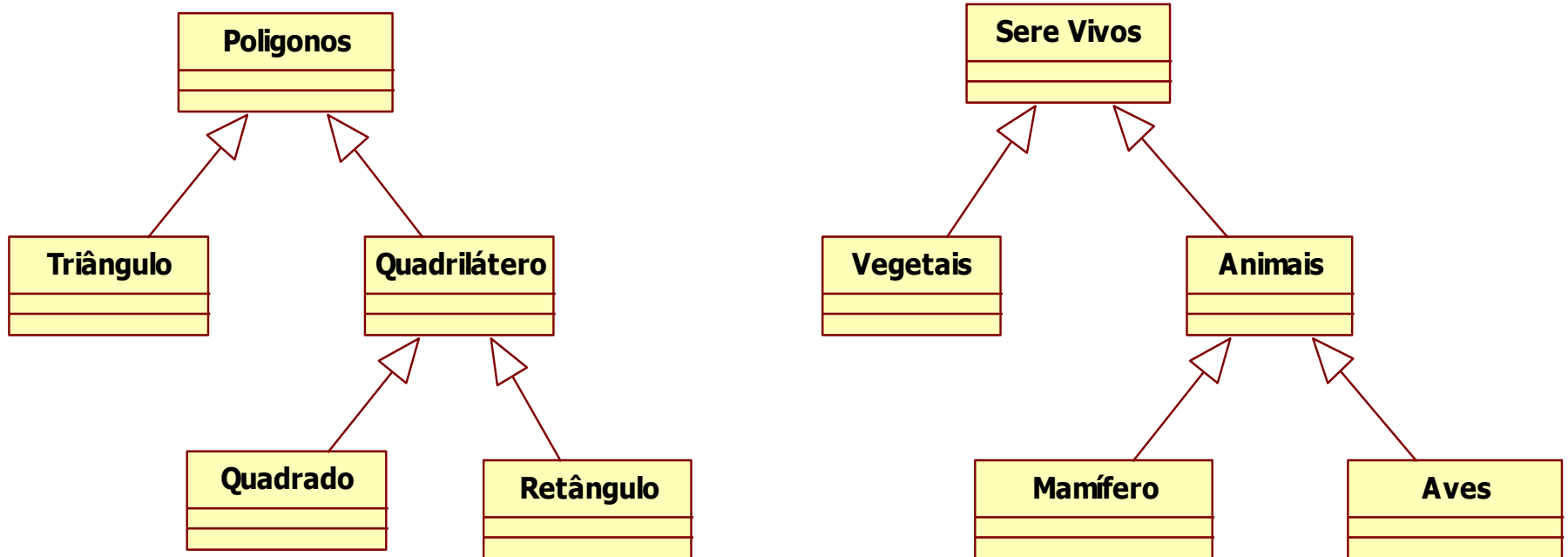
- Herança

- É um mecanismo específico pelo qual os objetos de uma sub-classe herdam toda a representação e comportamento das suas super-classes. Programação por herança é muito importante, devido ao reaproveitamento das definições semelhantes das super-classes.

# Modelagem de Objetos

## Conceitos Básicos

### ■ Herança



# Modelagem de Objetos

## Conceitos Básicos

---

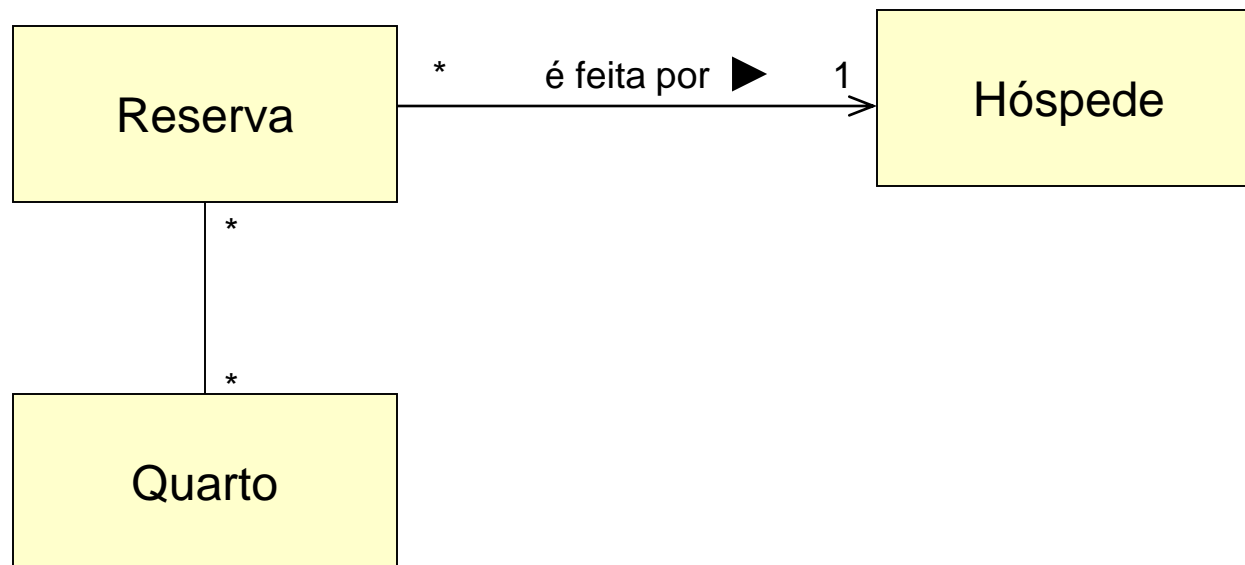
- Aspectos Estruturais: Associação simples
  - É um relacionamento estrutural que especifica objetos de um item conectado a objetos de outro item. A partir de uma associação conectamos duas classes, você é capaz de navegar de um objeto de uma classe até o objeto de outra classe e vice-versa.



# Modelagem de Objetos

## Conceitos Básicos

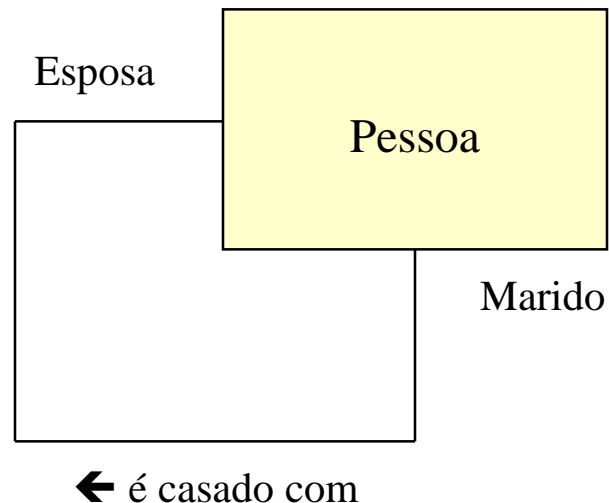
- Aspectos Estruturais: Associação simples



# Modelagem de Objetos

## Conceitos Básicos

- Aspectos Estruturais: Associação recursiva
  - Representação de de uma associação de objetos de uma mesma classe.



# Modelagem de Objetos

## Conceitos Básicos

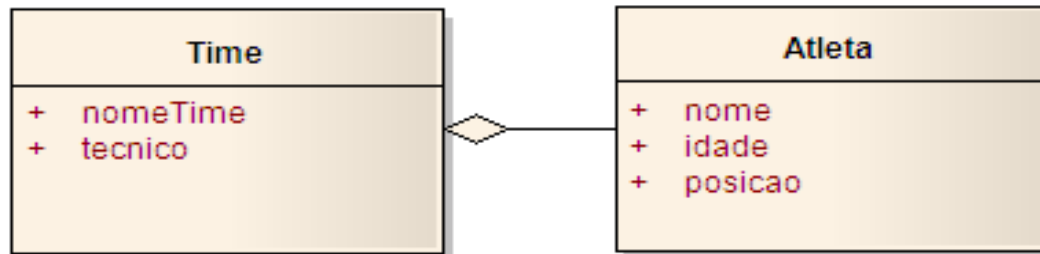
---

- Aspectos Estruturais: Relação todo-parte
  - Definição de um relacionamento onde uma classe representa um item maior (o “todo”), formado por itens menores (as “partes”), ou seja, um objeto do todo contém os objetos das partes. A relação todo-parte, na verdade, é apenas um tipo especial de associação, que é representada pela colocação de um diamante no objeto todo que compõe a relação. Temos dois tipos de relação todo-parte: Agregação e Composição.

# Modelagem de Objetos

## Conceitos Básicos

- Aspectos Estruturais: Relação todo-parte  
**AGREGAÇÃO** - a existência do Objeto-Parte faz sentido, mesmo não existindo o Objeto-Todo

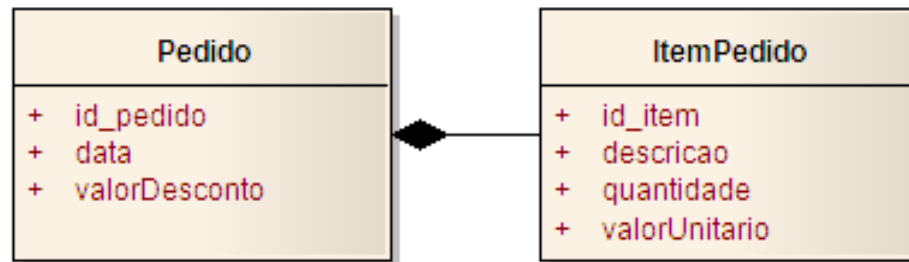


Um time é formado por atletas, ou seja, os atletas são parte integrante de um time, mas os atletas existem independentemente de um time existir. Nesse caso, chamamos esse relacionamento de **AGREGAÇÃO**.

# Modelagem de Objetos

## Conceitos Básicos

- Aspectos Estruturais: Relação todo-parte  
**COMPOSIÇÃO** - é uma agregação mais forte; nela, a existência do Objeto-Parte **NÃO** faz sentido se o Objeto-  
Todo não existir



Nesse caso, um pedido é composto por um ou vários itens, mas um produto **NÃO** é item de um pedido se não existe pedido. Assim, chamamos esse relacionamento de **COMPOSIÇÃO**.

# Modelagem de Objetos

## Conceitos Básicos

### ■ Multiplicidade

- Nos relacionamentos de associação simples e agregação, pode-se acrescentar a multiplicidade (similar a cardinalidade na modelagem estruturada), que especifica o número de instâncias de uma classe em relação a outra em um relacionamento:
  - 0..1 zero ou uma instância
  - 1 exatamente uma instância
  - 0..\* zero ou mais instâncias
  - \* ilimitado (valor padrão)
  - 1..\* uma ou mais instâncias
  - <literal>..\* número exato ou mais instâncias

# Modelagem de Objetos

## Conceitos Básicos

---

- Observação:
  - Todas as **associações simples**, e relações todo-parte (**agregação** e **composição**) devem ser nomeadas conforme o ambiente de negócios e também devem ser relacionadas as multiplicidades envolvidas. Estas características fornecem um melhor entendimento da relação entre os objetos.

# Modelagem de Objetos

## Conceitos Básicos

---

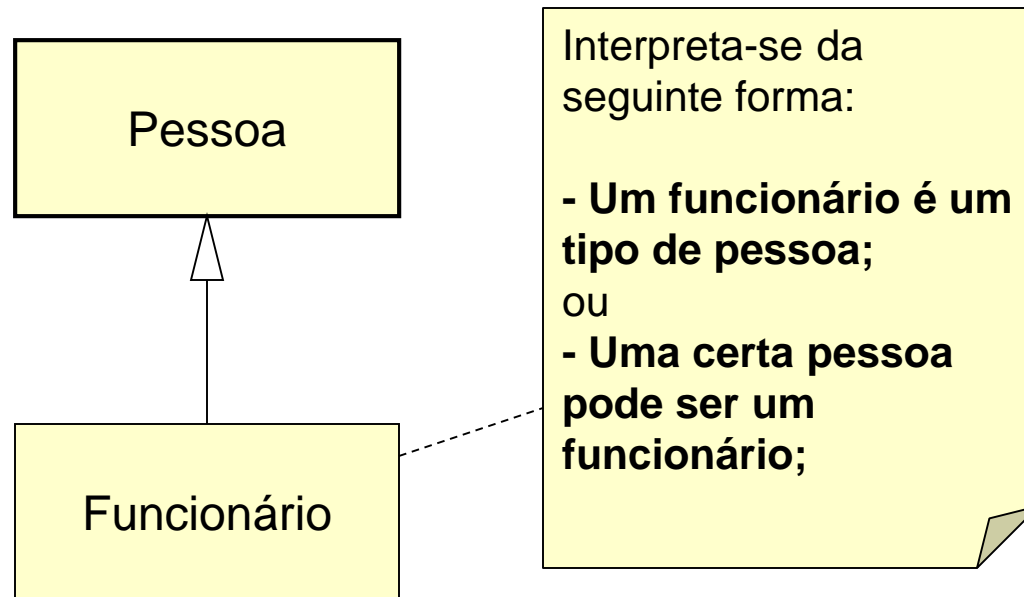
- Aspectos Estruturais: Relação de generalização /especialização (gen-espec ou herança)
  - É um relacionamento entre itens gerais (chamado superclasses ou classe-mãe) e tipos específicos desses itens (chamada de subclasse ou classe-filha). A generalização significa que os objetos da classe-filha herdam as propriedades (atributos e operações) da classe-mãe. Frequentemente, mas não sempre, as classes-filhas tem atributos e operações próprias, além daquelas encontradas nas classes-mãe.



# Modelagem de Objetos

## Conceitos Básicos

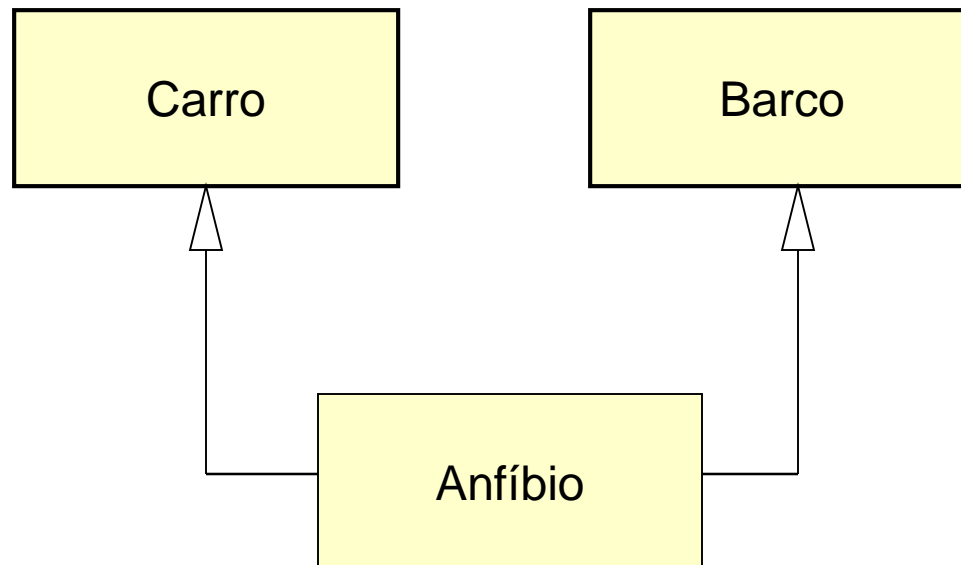
- Aspectos Estruturais: Relação de generalização /especialização (gen-espec ou herança)



# Modelagem de Objetos

## Conceitos Básicos

- Aspectos Estruturais: Relação de generalização (herança múltipla)





---

# Modelo Conceitual do Sistema (Diagrama de Classe Simplificado)



# UML - Modelo Conceitual

---

- Um *diagrama de classe* ilustra as especificações de software para as classes e interfaces do sistema
- Inclui:
  - Classes, associações e atributos;
  - Interfaces (com operações e constantes);
  - Métodos que manipulam os objetos;
  - Informação sobre o tipo dos atributos;
  - Visibilidade;
  - Navegabilidade;
  - Dependências;
- UML não diferencia modelo conceitual de diagrama de classe (o termo “classe de implementação” é usado para distinguir o segundo do primeiro)



# UML - Modelo Conceitual

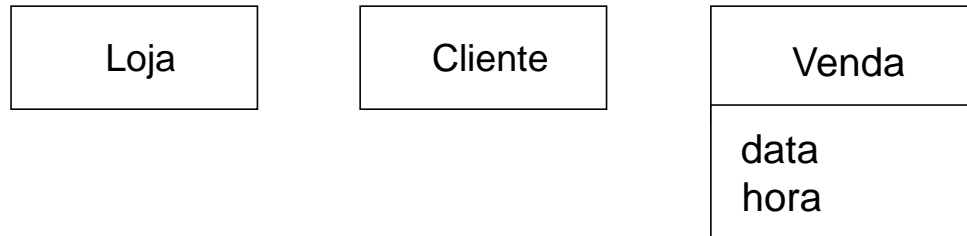
---

- Modelo Conceitual (diagrama de Classe Inicial)
  - Artefato mais importante da AOO
  - Representa conceitos relevantes (do ponto de vista do modelador) do domínio do problema
  - Na UML, ilustrado com diagramas de estruturas estáticas contendo:
    - Conceitos
    - Associações entre conceitos
    - Atributos de conceitos

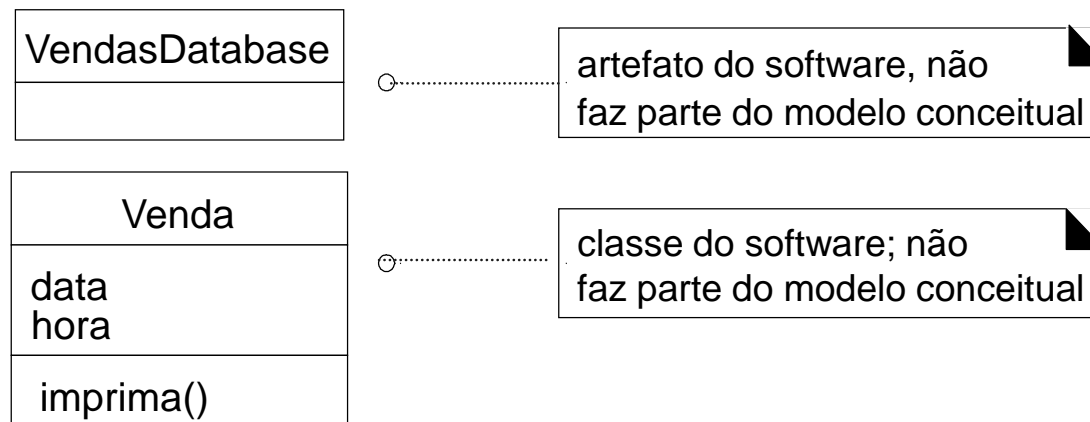
# UML - Modelo Conceitual

## ■ Conceitos

- Idéias, coisas, ou objetos do mundo real



- Não representam componentes de software





# UML - Modelo Conceitual

---

- Identificando Conceitos
  - Regras úteis:
    - É melhor especificar demais do que especificar de menos;
    - Não exclua conceitos simplesmente porque os requisitos não indicam a necessidade de guardar informações sobre eles
    - Comece fazendo uma lista de conceitos candidatos a partir de uma lista de conceitos comuns;
    - Considere os substantivos e frases nominais nas descrições dos cenários dos casos de uso como possíveis candidatos a conceitos ou atributos.

# UML - Modelo Conceitual

---

- Identificando os Conceitos
  - Analise a descrição dos casos de uso.
  - Identifique os substantivos ou cláusulas substantivas.
  - Classifique os substantivos para um melhor entendimento.



# UML - Modelo Conceitual

<b>Categoria</b>	<b>Exemplos</b>
Objeto físico ou tangível	Terminal de ponto-de-venda; Avião
Lugares	Loja; Aeroporto
Transações	Venda, Pagamento; Reserva
Itens de transação	Itens de venda; Parcelas de pagamento
Papéis de pessoas	Operador; Piloto
<i>Container</i> de coisas	Loja; Avião
Coisas em um <i>container</i>	Item; Passageiro
Sistemas externos	Serviço de crédito; Controle de tráfego aéreo
Nomes abstratos	Fome; Aracnofobia
Organizações	Departamento de vendas; Companhia aérea
Eventos	Política de devolução; Política de cancelamento
Catálogos	Catálogo de produtos; Catálogo de peças
Registros contratuais	Recibo, Contrato de trabalho; Registro de manutenção
Serviços financeiros	Linha de crédito; Ações
Manuais, livros	Manual do empregado; Manual de reparos



# UML - Modelo Conceitual

---

- Critérios para definição de conceitos
  - Possui atributos?
  - Guarda informação?
  - Há atributos comuns a outros conceitos?
  - O conceito é essencial ao sistema?



# UML - Modelo Conceitual

---

- Criando um Modelo Conceitual
  - Passos sugeridos:
    1. Liste os conceitos candidatos para os casos de usos em questão usando a lista de categorias comuns e identificação textual de nomes;
    2. Desenhe-os em um modelo conceitual;
    3. Adicione as associações necessárias para registrar os relacionamentos para os quais é preciso preservar alguma memória;
    4. Adicione os atributos necessários para cumprir os requisitos de informação.

# UML - Diagrama de Classe

## Modelo Conceitual

---

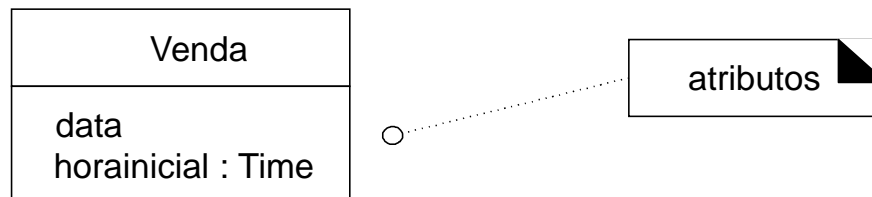
- Identificando Associações
  - Regras úteis:
    - Focar nas associações cujo conhecimento deve ser preservado;
    - Usar nomes baseados em expressões verbais que façam sentido quando lidas no contexto do modelo;
    - Evitar mostrar associações deriváveis ou redundantes;
    - É mais importante identificar conceitos do que associações;
    - Associações demais tendem a confundir um modelo conceitual ao invés de iluminá-lo .

# UML - Diagrama de Classe

## Modelo Conceitual

### ■ Atributo

- Um atributo é um dado lógico de um objeto do domínio
- Definidos para conceitos cujos requisitos (casos de uso) sugerem a necessidade de preservar algum tipo de informação
  - Ex.: atributos *data* e *hora* para o conceito Venda
- Notação na UML



# UML - Diagrama de Classe

## Modelo Conceitual

---

- Identificando Atributos
  - Atributos devem preferencialmente representar tipos primitivos de dados ou de valores simples
    - Ex.: *Data, Número, Texto, Hora, Endereço*, etc.
  - Atributos não devem ser usados para:
    - Representar um conceito complexo
    - Relacionar conceitos (atributo “chave-estrangeira”)

# UML - Diagrama de Classe

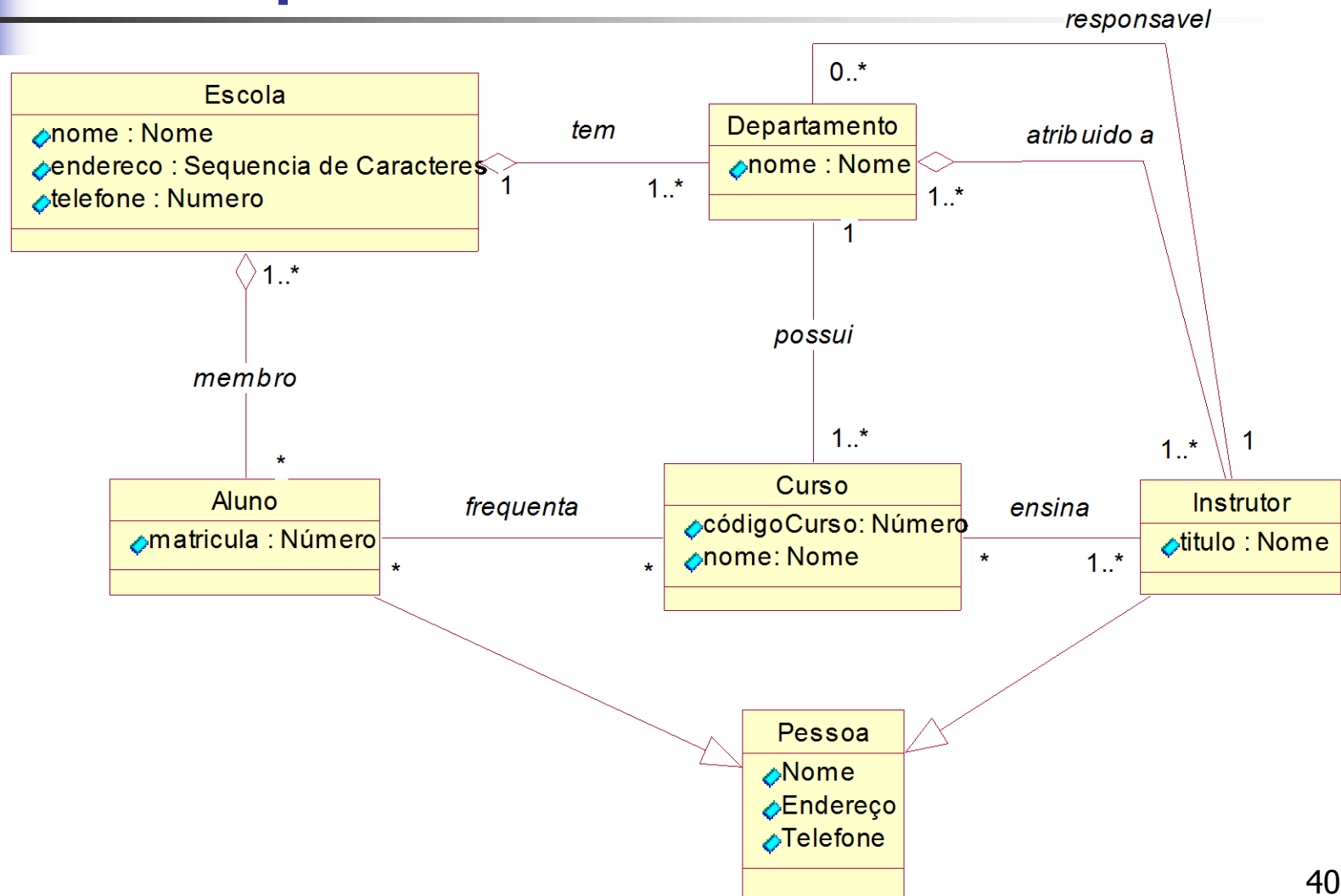
## Modelo Conceitual

---

- Um atributo deve ser de tipo não-primitivo quando:
  - É composto de seções separadas
    - Ex.: endereço, data
  - Precisa ser analisado ou validado
    - Ex.: CPF, número de matrícula
  - Possui outros atributos
    - Ex.: Um preço promocional com prazo de validade
  - É uma quantidade com uma unidade
    - Ex.: valores monetários, medidas

# UML - Diagrama de Classe

## Exemplo de Modelo Conceitual





# UML - Diagrama de Classe

## Exemplo de Modelo Conceitual

---

- Exercícios:
  - Construir o Modelo Conceitual do Sistema de Livraria

# Caso de Uso – Efetuar venda

Use Case: UC2 - Solicitar Venda

Atores: Cliente, Vendedor

Pré-condições: Livro cadastrado, o cliente cadastrado, e o Vendedor cadastrado

Cenário Principal:

- 1 - O cliente solicita um pedido de venda de livro
- 2 - O Vendedor inicia uma nova venda no sistema
- 3 - Identificar o cliente cadastrado na venda
- 4 - Identificar o vendedor responsável pela venda
- 6 - Repetir os seguintes passos enquanto existirem livros a serem vendidos
  - 6.1 - Inserir o livro desejado como item de venda
  - 6.2 - Inserir a quantidade de venda do item especificado
  - 6.2 - Definir o valor de venda a ser aplicado no livro inserido
- 7 - Definir a forma de pagamento da venda(a vista ou boleto)

8 - Definir a remessa do pedido venda(retirado na loja ou remessa)

9 - Finalizar pedido de venda de livros

10 - Baixar o estoque para cada item de livro vendido com a quantidade inserida

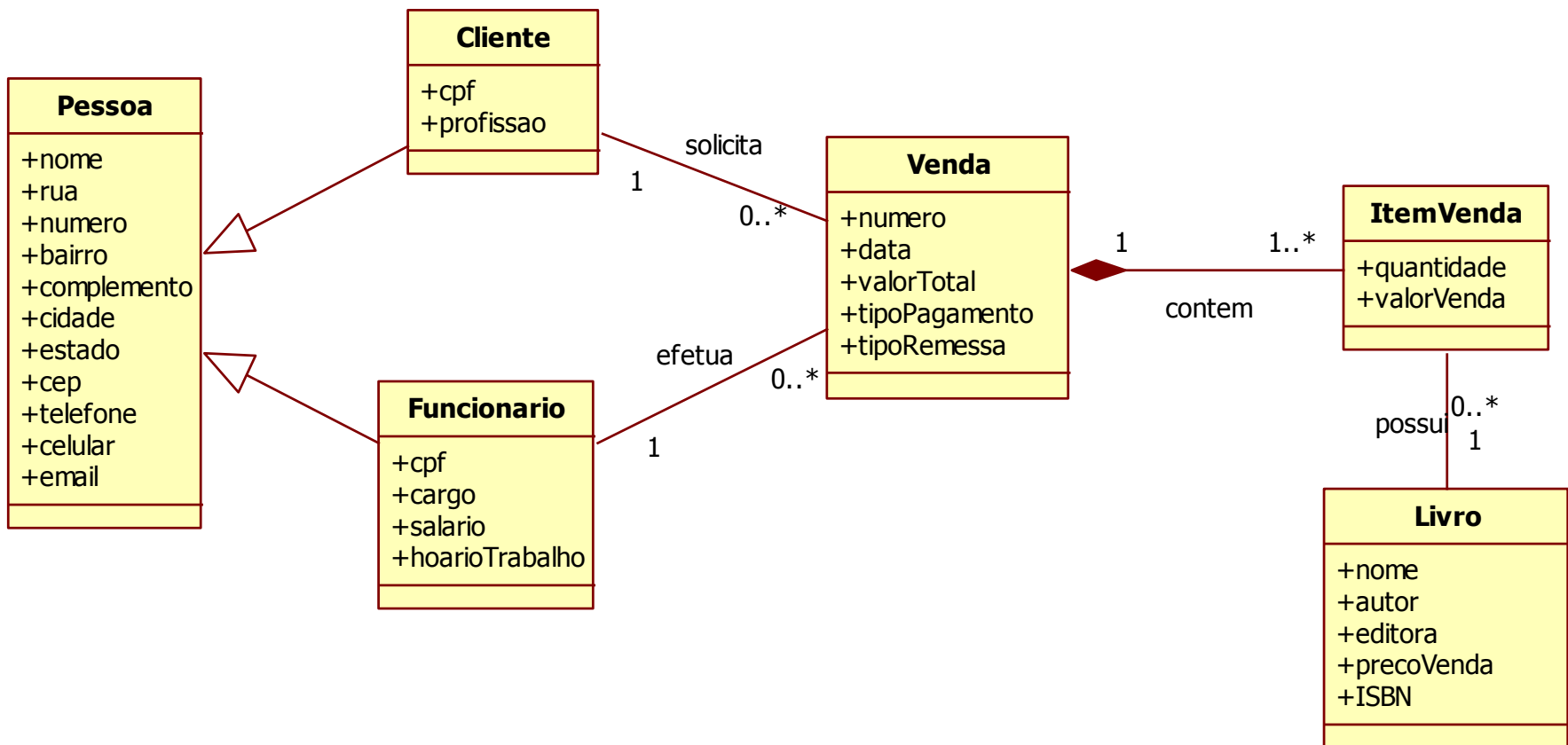
11 - Gerar cupom de venda de livro

Cenários Alternativos:

3.1 - Se cliente não cadastrado -- Cadastrar Cliente

6.1.1 - Se não existir estoque disponível para o livro que esta sendo vendido informar o vendedor e retirar o livro dos itens de venda

# Modelo Conceitual do Sistema de Livraria



# UML - Diagrama de Classe

## Exemplo de Modelo Conceitual

---

- Exercícios:
  - Construir o Modelo Conceitual do Sistema de Locadora de Games

# Sistema de Locadora de Games

- Você foi contratado para desenvolver um sistema para GG Locadora de Games. Nessa locadora, os títulos são classificados em vários gêneros de game. Os títulos estão gravados apenas em DVD e alguns títulos são constituídos por mais de um disco. Um título pode possuir várias cópias. O preço de aluguel de um título varia conforme sua classificação: lançamento, catálogo, clássico e museu. O sistema deve guardar um histórico de títulos por cliente e vice-versa. Um cliente possui um cadastro com seus dados (nome, endereço, ...) e o nome de seus dependentes. Os clientes podem fazer reserva de títulos. Um cliente não tem limite para empréstimo de filmes, mas ele só pode pegar novos filmes após a devolução de 50% dos títulos correspondente à locação anterior. Dependendo da classificação do título, o mesmo possui um prazo para locação (os lançamentos devem ser devolvidos em 24 horas. O sistema deve oferecer informação sobre o aluguel de títulos por gênero, quantas vezes um título foi alugado por cópia. O sistema deve informar também se um título já se pagou. A locação de um filme deve informar quem foi o atendente que a fez. Os títulos em uma locação podem ter diferentes datas de devolução. O sistema deve ser capaz de informar os títulos que estão com atraso de devolução. O sistema deve informar também qual a arrecadação diária da locadora. O sistema deve também controlar a venda de produtos da loja, tais como: balas, chicletes, ...

# Sistema de Vídeo Locadora

## UC01 Gerenciar Títulos (RF01)

### Descrição

Realiza a gerência do acervo da locadora quanto a inserção, remoção de títulos e modificação dados referentes aos mesmos.

**Pré-condições:** ---

**Atores:** Gerente

### **Cenário Principal**

(Inserção)

1. O gerente solicita a inserção de um novo título
2. O sistema lista todos os títulos cadastrados
3. Informar dados de um novo título
4. Validar dados do título
5. Confirmar inserção do título no sistema

(Modificação)

1. O gerente solicita a alteração dos dados de um título
2. O sistema lista todos os títulos cadastrados
3. Informar o código do título a ser modificado
4. Busca o título no sistema e apresenta na tela
5. Modificar os dados necessários para o título
6. Confirma a modificação do título

(Remoção)

1. O gerente solicita a exclusão de um título
2. O sistema lista todos os títulos cadastrados
3. Informar o código do título a ser excluído
4. Busca o título no sistema e apresenta na tela os dados
5. Solicita a confirmação de exclusão do título
6. Confirma a exclusão do título
7. Exclui o título e todas os registros de outras tabelas relacionados a ele

**Cenário Alternativo:** modificação de um título

- 2 Se o código do título não corresponder a um título cadastrado, dar oportunidade para o usuário apresentar novo título ou desistir do caso.

# Sistema de Vídeo Locadora

## **UC08 Realizar Locação (RF08)**

### **Descrição**

Realiza a locação de um ou mais títulos a um cliente.

### **Pré-condições**

Cliente deve estar cadastrado (UC05).

Título deve estar cadastrado (UC01).

**Atores:** Cliente, Atendente

### **Cenário Principal**

1. O cliente apresenta as cópias dos títulos a serem locados.
2. O atendente verifica as pendências do cliente.
3. O atendente cria uma locação para o cliente.
4. O atendente insere cada cópia de cada título como item de locação a essa locação.
5. O sistema estabelece a data de devolução de cada título, conforme classificação e o respectivo tempo de locação.
6. O sistema coloca as cópias locadas como indisponíveis para outras locações
7. O sistema emite comprovante de locação.

**Cenário Alternativo:** cliente com pendências

- 2.1 Se o cliente tiver multa por atraso, o atendente deve solicitar o pagamento da mesma, antes de realizar a locação. Se o cliente não pagá-la, cancelar a locação.
- 2.2 Se o cliente não tiver devolvido 50% dos títulos da última locação, cancelar a nova locação.

**Cenário Alternativo:** Cópia com reserva

- 4.1 Se a título que esta sendo inserida na locação tiver uma reserva associada a ele, para a data específica, o sistema informa o atendente e a cópia é retirada da locação.