



Engenharia de Software I

Prof^a. Me. Cynara Leão Garcia

cynara.garcia@unicesumar.edu.br



Diagrama de Classes

Conceito Básico

- O caso de uso fornece uma perspectiva do sistema de um ponto de vista externo (do ator)
- Internamente os objetos colaboram para atender às funcionalidades do sistema
- Demonstra a estrutura estática dessa colaboração, mostra as classes de um sistema, seus atributos e operações, e como as classes se relacionam.
- O diagrama de objetos (que pode ser visto como uma instanciização do diagrama de classes) também representa a estrutura estática

Perspectiva de um Diagrama de Classes

- O modelo conceitual (análise) representa as classes no domínio do negócio em questão. Não leva em consideração restrições inerentes à tecnologia a ser utilizada na solução de um problema.
- O modelo de classes de especificação (projeto) é obtido através da adição de detalhes ao modelo anterior conforme a solução de software escolhida.
- O modelo de classes de implementação corresponde à implementação das classes em alguma linguagem de programação.

Definição de Classes

Uma classe de objetos, ou simplesmente classe, descreve :

- Um conjunto de objetos com propriedades semelhantes (atributos);
- O mesmo comportamento (operações);
- Os mesmos relacionamentos; e
- A mesma semântica.

São os blocos de construção mais importantes de qualquer sistema orientado a objetos;

Conceitos Básicos

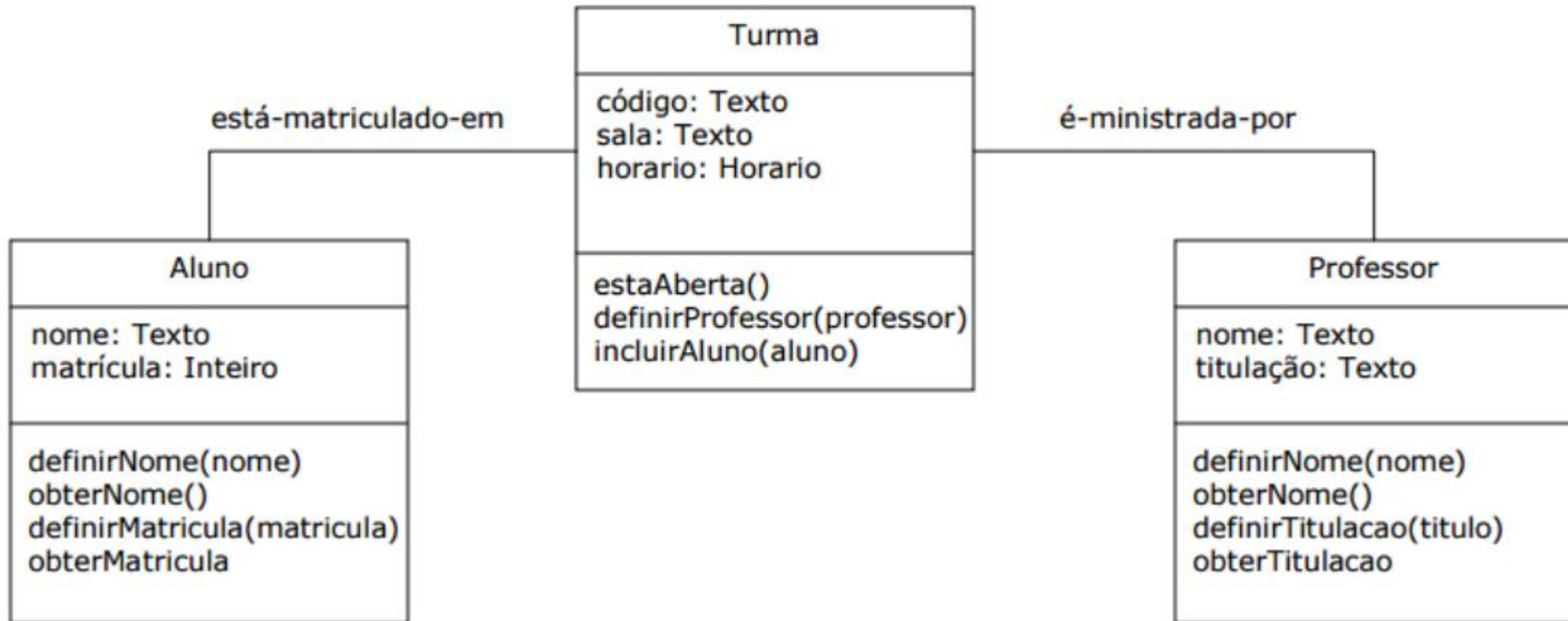
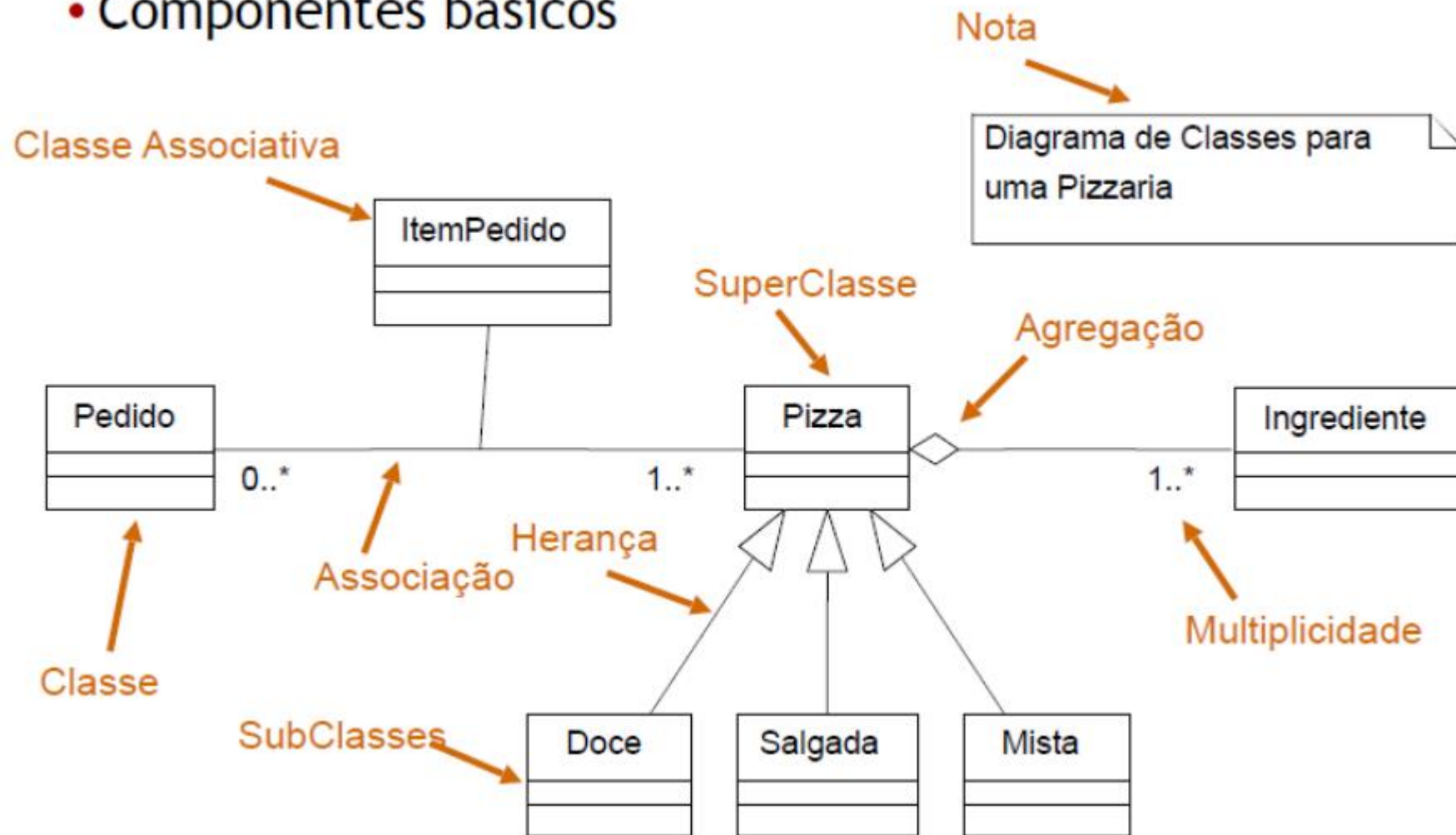


Diagrama de Classes

- Componentes básicos



Conceitos Básicos: Classes

- Representada através de uma “caixa” com no máximo três compartimentos exibidos (nome, atributos e métodos).
- Devem receber nomes de acordo com o vocabulário do domínio do problema.
- É comum adotar um padrão para nomeá-las

Ex: todos os nomes de classes serão substantivos singulares com a primeira letra maiúscula.

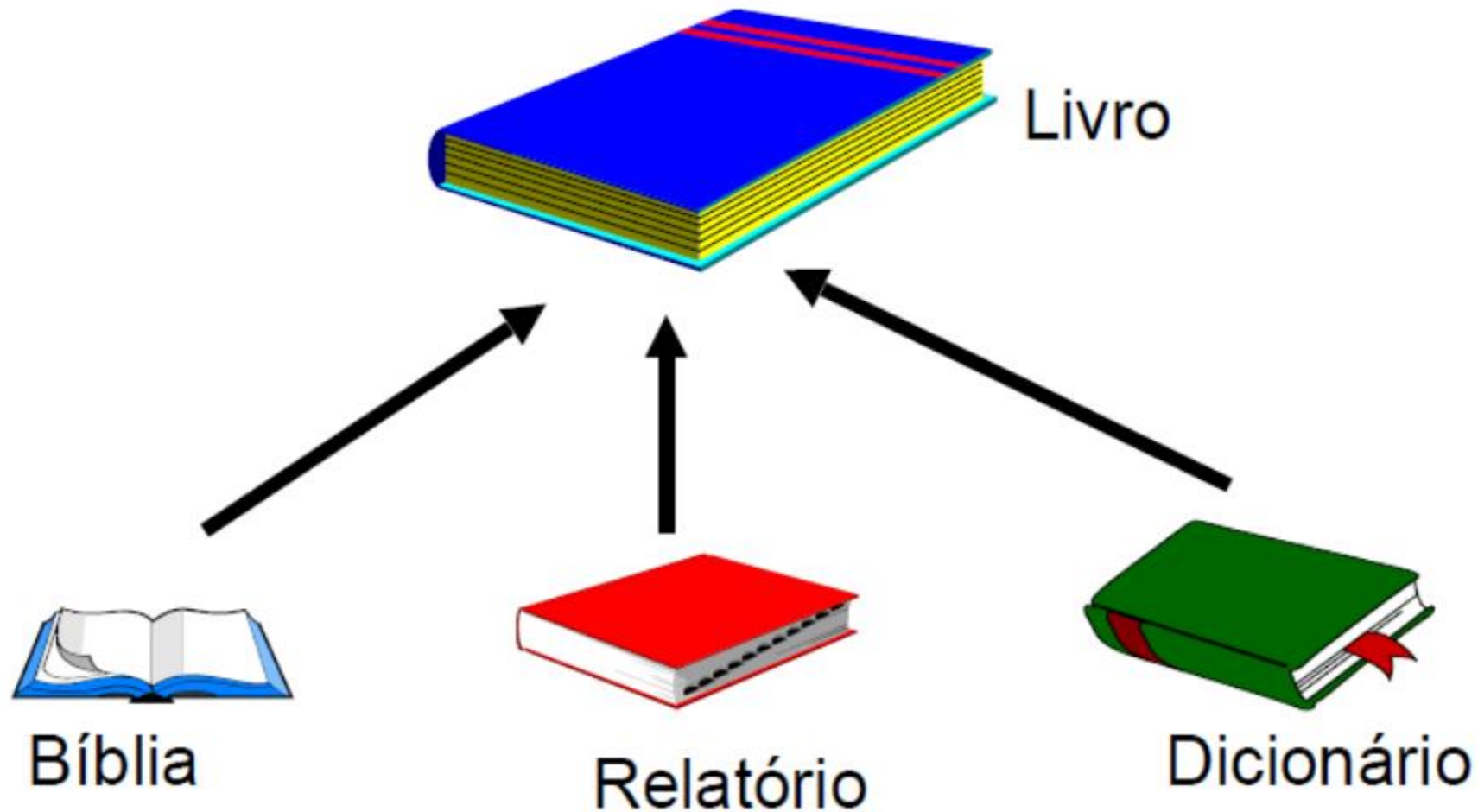
Nome da Classe

Nome da Classe
lista de atributos

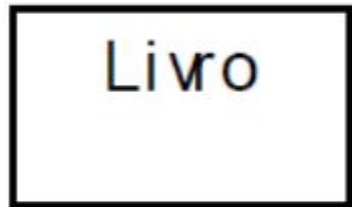
Nome da Classe
lista de operações

Nome da Classe
lista de atributos
lista de operações

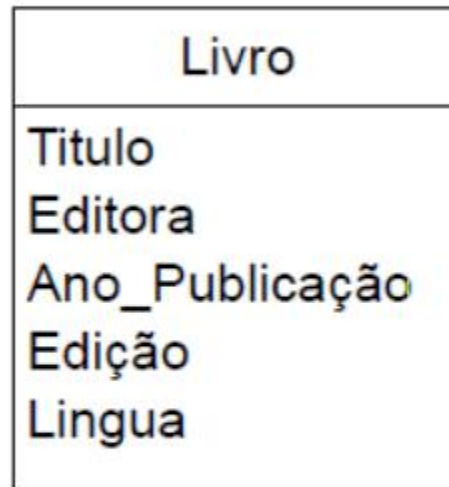
Conceitos Básicos: Classes



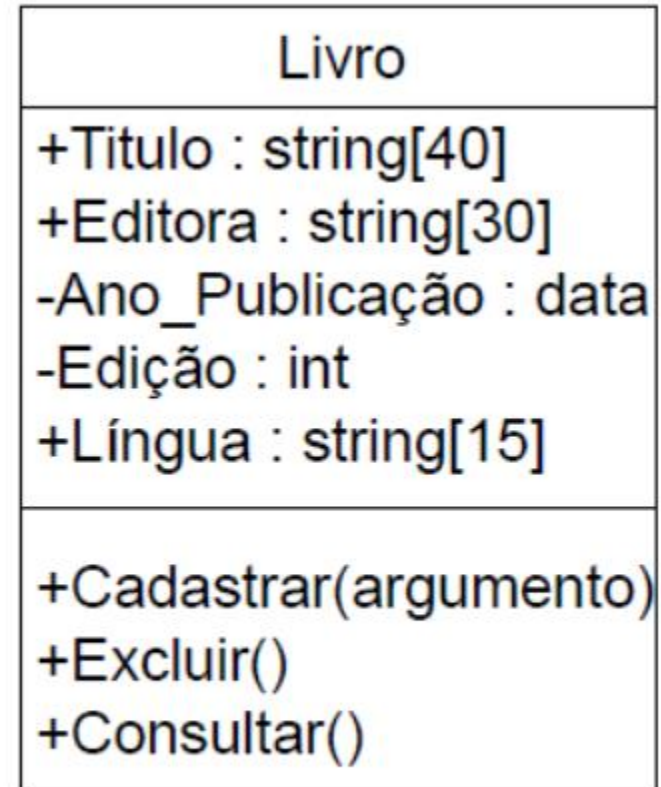
Conceitos Básicos: Classes



Classe com detalhes
suprimidos



Classe com detalhes
a nível de análise



Classe com detalhes
a nível de implementação

Conceitos Básicos: Atributos

- Representam uma propriedade do objeto que está sendo modelado;
- Cada atributo possui um valor para cada instância de um objeto;
- Cada nome de atributo deve ser único dentro de uma classe;
- Podem ser representados apenas por seu nome, ou especificando seu tipo e valor inicial;

Conceitos Básicos: Atributos

- Representam o conjunto de características (estado) dos objetos daquela classe
- Visibilidade:
 - + público: visível para qualquer classe
 - # protegido: visível somente para classes derivadas
 - privado: visível somente para classe

Exemplo:

+ nome : String

Conceitos Básicos: Atributos



Conceitos Básicos: Atributos

Livro
Titulo
Editora
Ano_Publicação
Edição
Lingua

Atributos Simples

Livro
+Titulo : string[40] +Editora : string[30] -Ano_Publicação : data -Edição : int +Língua : string[15]
+Cadastrar(argumento) +Excluir() +Consultar()

Atributos Completos

Conceitos Básicos: Métodos

- Representam o conjunto de operações (comportamento) que a classe fornece
- Visibilidade:
 - + público: visível para qualquer classe
 - # protegido: visível somente para classes derivadas
 - privado: visível somente para classe

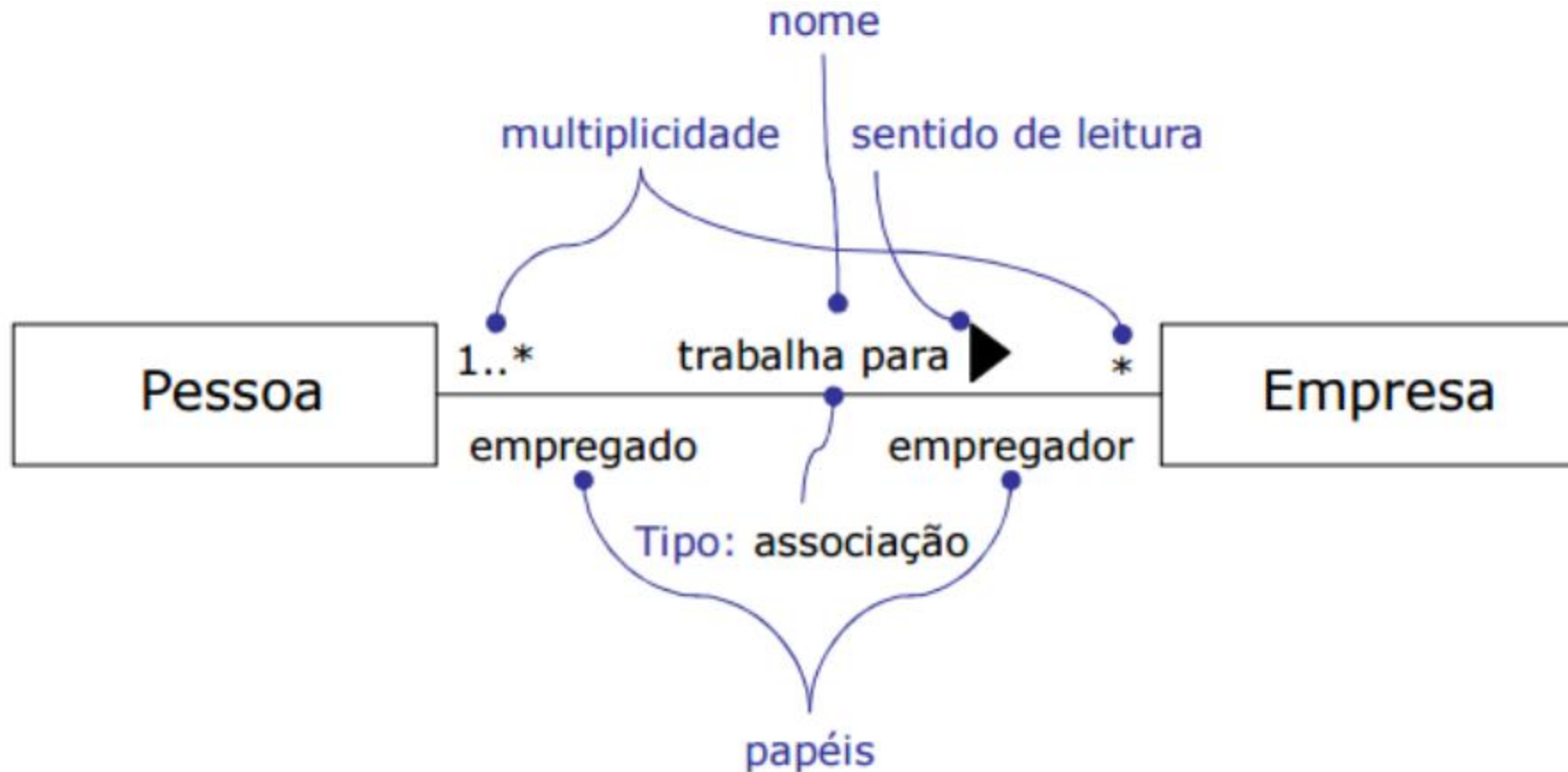
Exemplo:

- getNome() : String

Conceitos Básicos: Relacionamentos

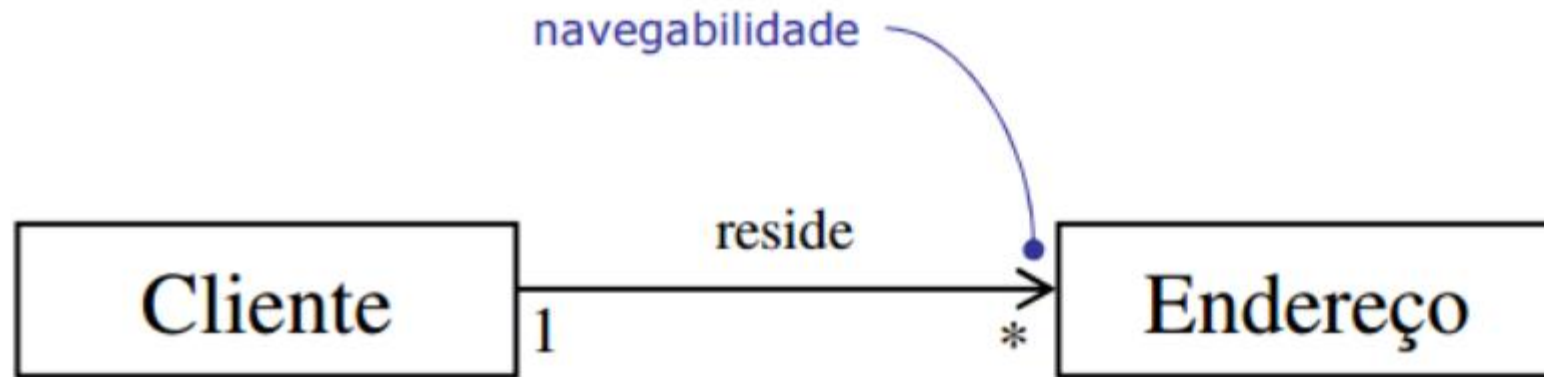
- Os relacionamentos possuem:
 - Nome: descrição dada ao relacionamento (faz, tem, possui,...)
 - Sentido de leitura
 - Navegabilidade: indicada por uma seta no fim do relacionamento
 - Multiplicidade: 0..1, 0..*, 1, 1..*, 2, 3..7
 - Tipo: associação (agregação, composição), generalização e dependência
 - Papéis: desempenhados por classes em um relacionamento

Conceitos Básicos: Relacionamentos



E a navegabilidade?

Conceitos Básicos: Relacionamentos



- O cliente sabe quais são seus endereços, mas o endereço não sabe a quais clientes pertence

Conceitos Básicos: Relacionamentos

Relacionamentos: **Associação**

- Uma associação é um relacionamento estrutural que indica que os objetos de uma classe estão vinculados a objetos de outra classe.
- Uma associação é representada por uma linha sólida conectando duas classes.



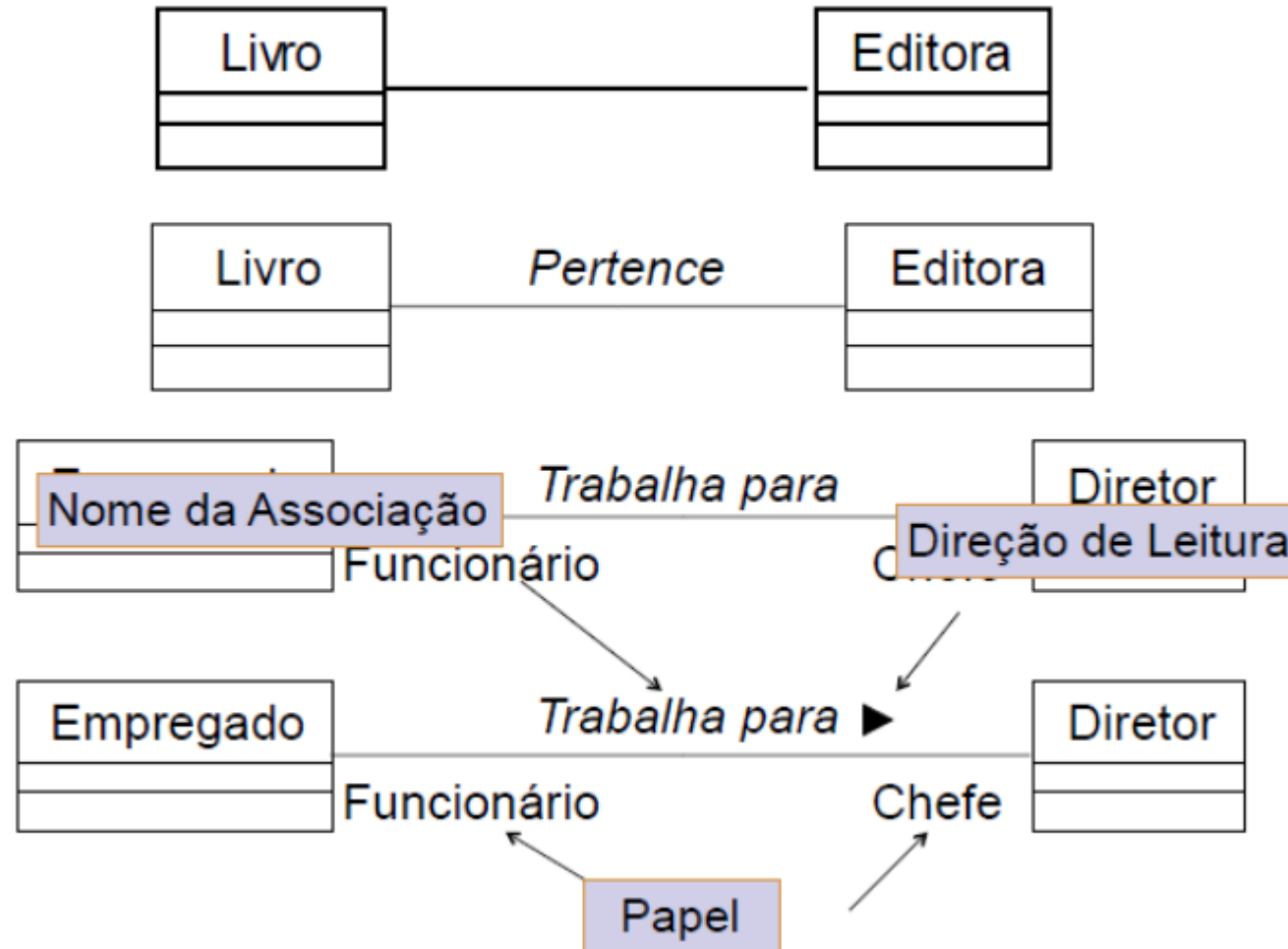
Conceitos Básicos: Relacionamentos

Relacionamentos: **Associação**

- Associações Normais (binárias)
- Tipo mais comum de associação, sendo uma conexão entre duas classes;
- Uma associação pode possuir um nome, que normalmente é um verbo (substantivos são aceitos);
- Cada terminação da associação pode possuir um papel, mostrando como uma classe é vista pela outra;

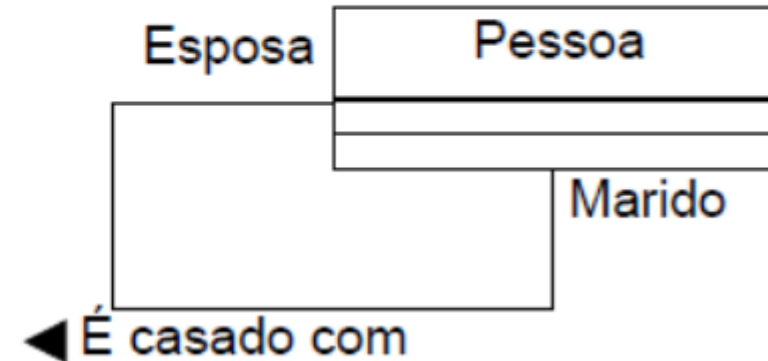
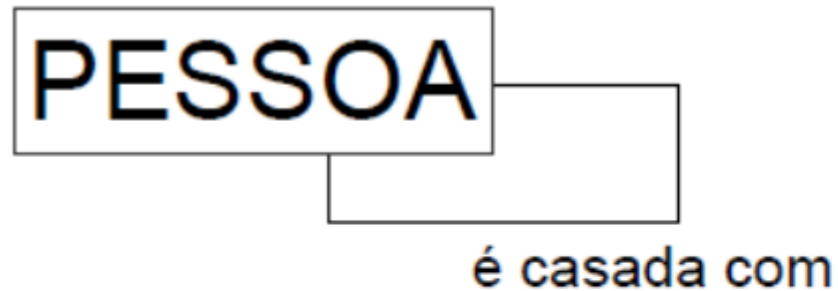
Conceitos Básicos: Relacionamentos

- Associações Normais



Conceitos Básicos: Relacionamentos

- Associações Binária Recursiva (reflexiva, autoassociação)
- Associação entre dois objetos, sendo que os mesmos pertencem a mesma classe;



Relacionamento: Associação

A multiplicidade especifica quantos objetos de uma classe relacionam-se a um único objeto de classe associada;

Indicadores de multiplicidade:

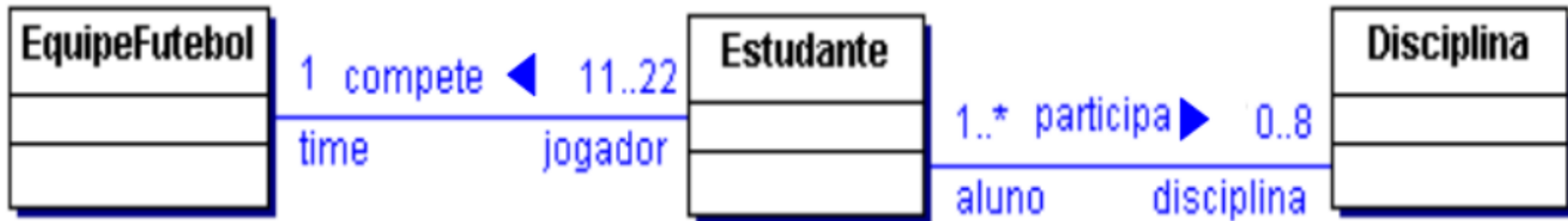
- 1 Exatamente um
- 1..* Um ou mais
- 0..* Zero ou mais (muitos)
- * Zero ou mais (muitos)
- 0..1 Zero ou um
- m..n Faixa de valores (por exemplo: 4..7)



Classes: Relacionamento

Relacionamentos: **Associação**

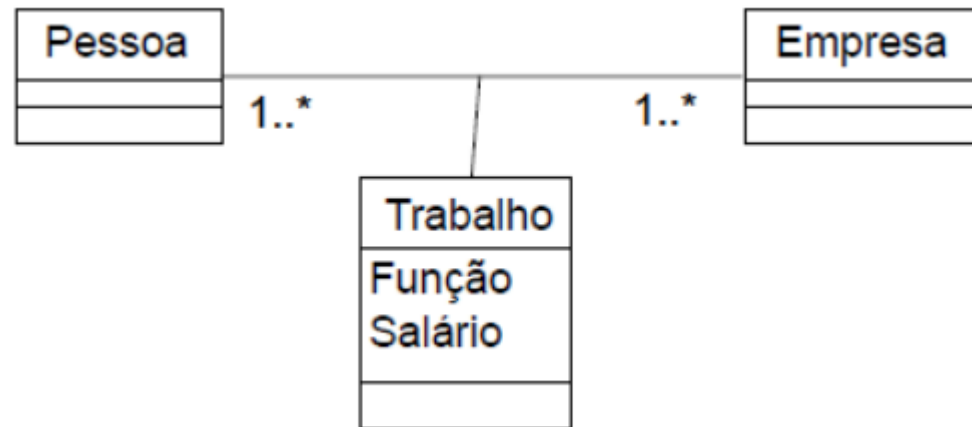
- Exemplo:
- Um Estudante pode ser um aluno de uma Disciplina e um jogador da Equipe de Futebol
- Cada Disciplina deve ser cursada por no mínimo 1 aluno
- Um aluno pode cursar de 0 até 8 disciplinas



Classes: Relacionamento

Classe Associativa

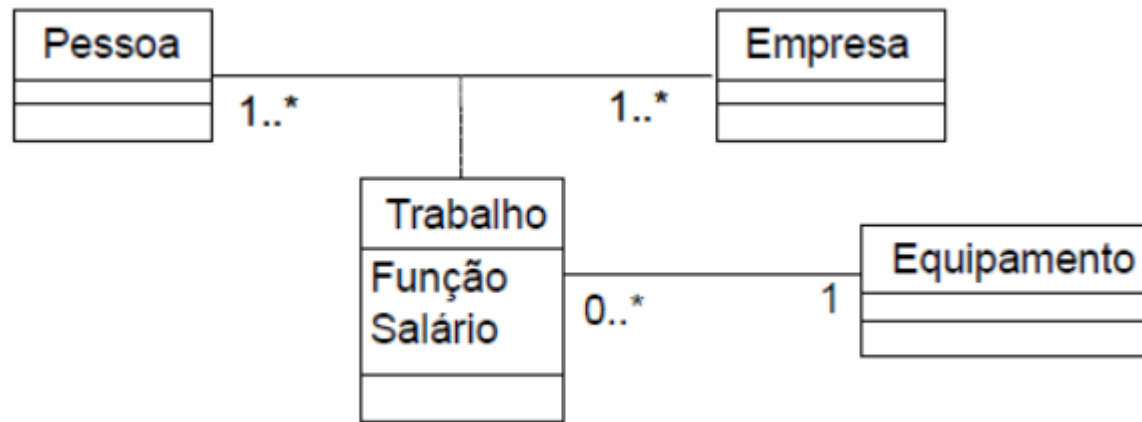
- Uma classe associativa é uma associação com propriedades de classe;
- Este tipo de classe surge quando duas ou mais classes estão associadas e é necessário manter informações sobre a associação;



Classes: Relacionamento

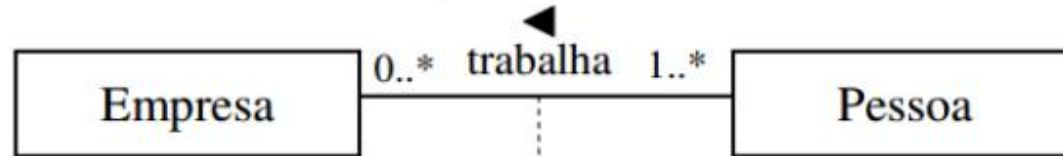
Classe Associativa

- Não se deve nomear a linha de associação de uma classe associativa;
- Somente o nome escolhido para a classe associativa deve ser suficiente para expressar o significado desejado;
- Uma classe associativa pode participar de outros relacionamentos;

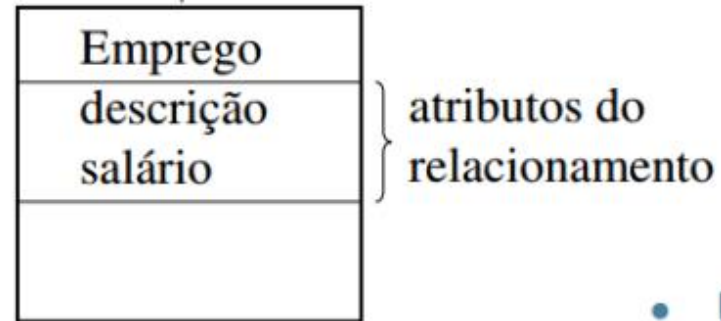


Classes: Relacionamento

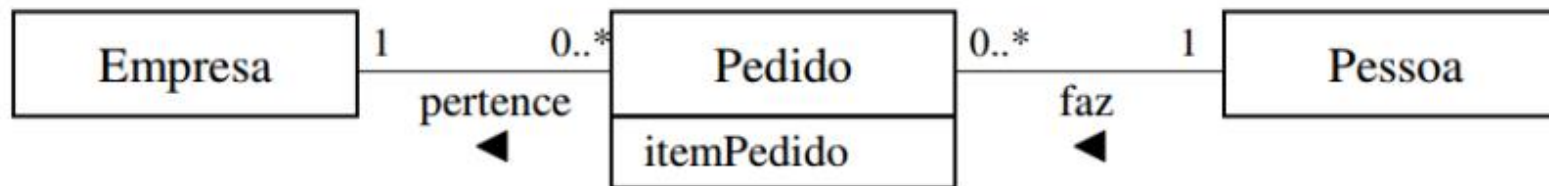
- Classe de associação



- Não existe uma pessoa com dois empregos na mesma empresa



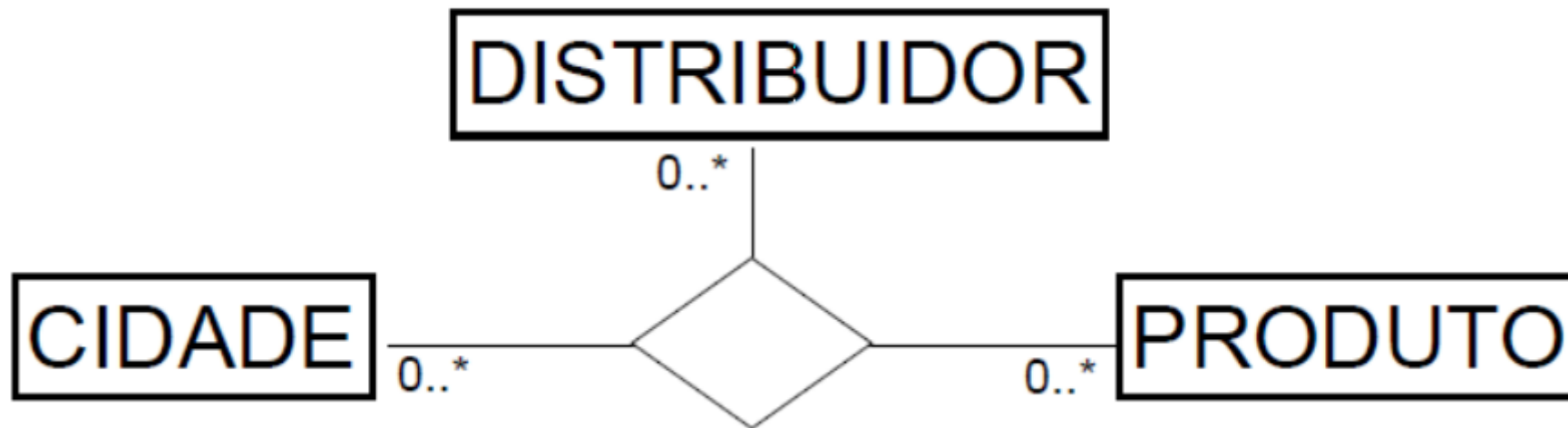
- Uma pessoa pode fazer mais de um pedido na mesma empresa



Classes: Relacionamento

Associações Ternárias

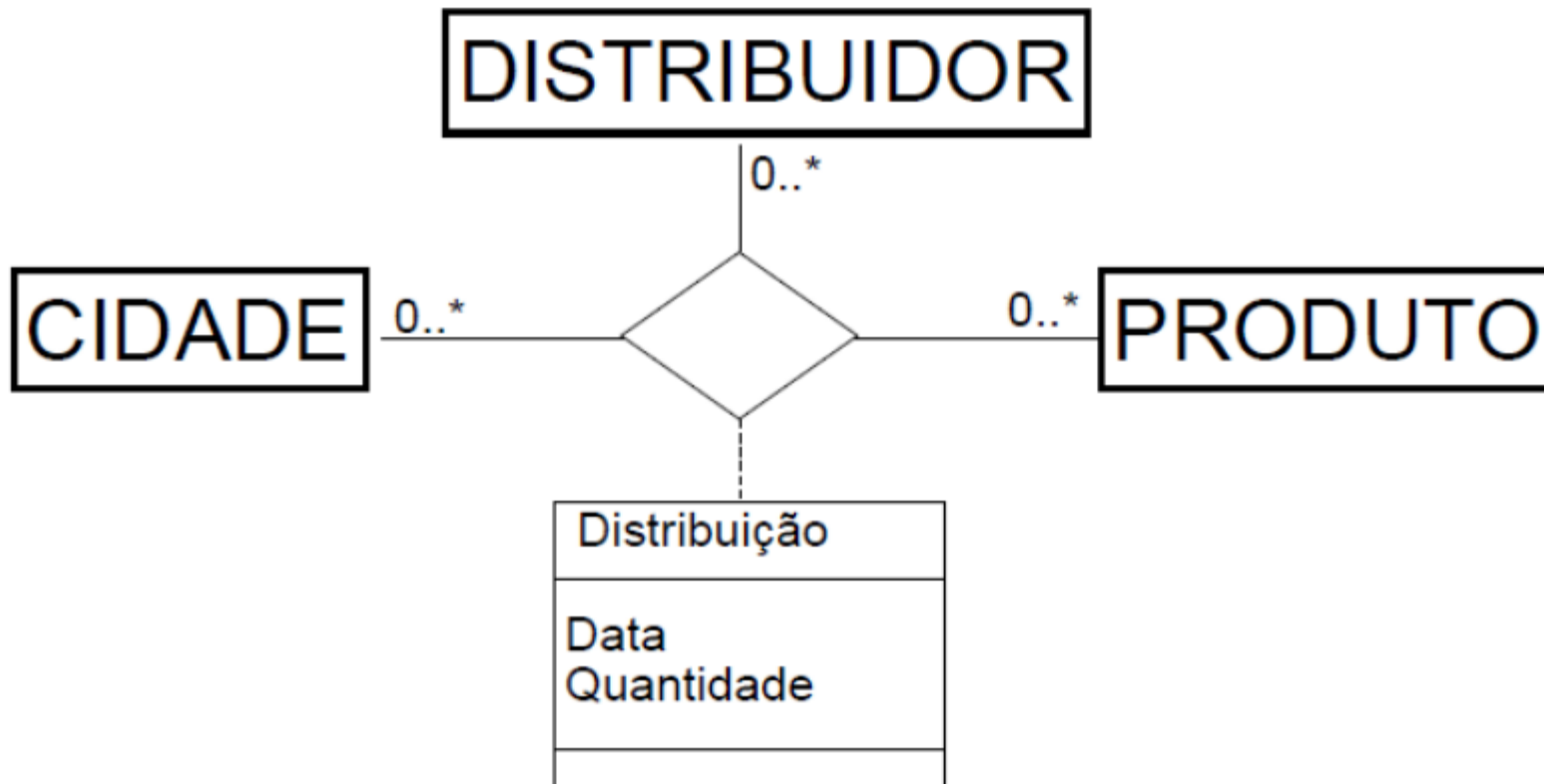
- Quando três classes estão associadas entre si;
- Representada por um losango, ligando cada classe do relacionamento;



Classes: Relacionamento

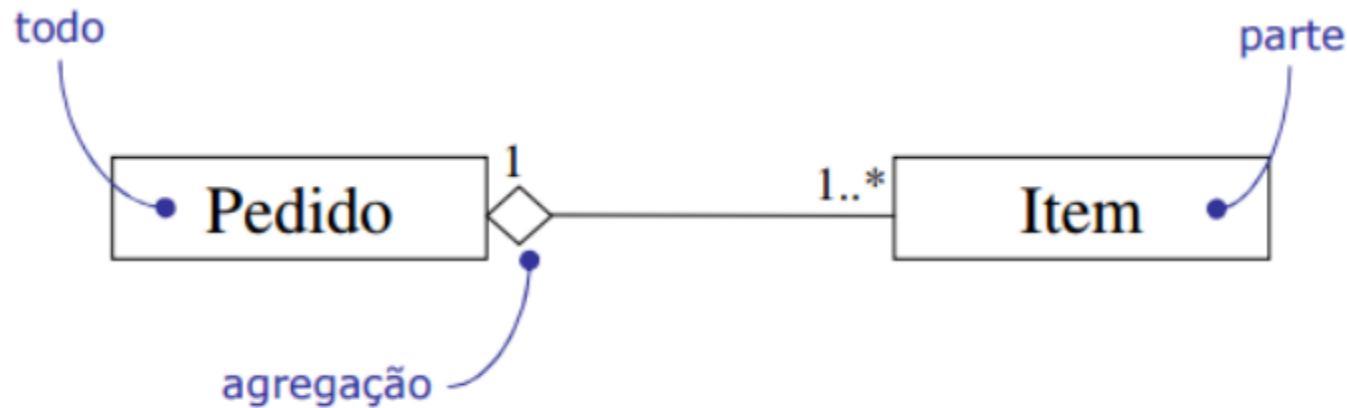
Associações Ternárias

- Pode ter uma classe associada na ternária



Relacionamento: Agregação

- É um tipo especial de associação
- Indica que uma das classes do relacionamento é uma parte de outra classe – relação todo-parte;



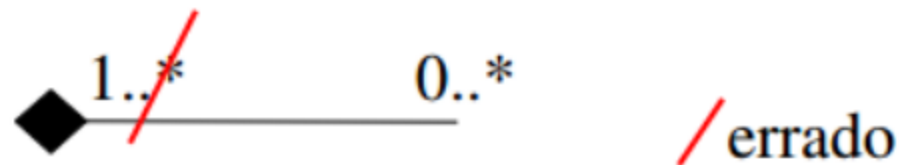
- Um objeto “parte” pode fazer parte de vários objetos “todo”

Relacionamento: Agregação

- A diferença entre associação e agregação é puramente semântica - em uma agregação um objeto está contido no outro;
- Onde se puder utilizar uma agregação, pode-se usar também
- uma associação;
- Palavras Chave: “consiste em”, “é parte de”;
- Graficamente representada por uma linha conectando as classes relacionadas, com um diamante (losango) branco perto da classe que representa o todo.

Relacionamento: Composição

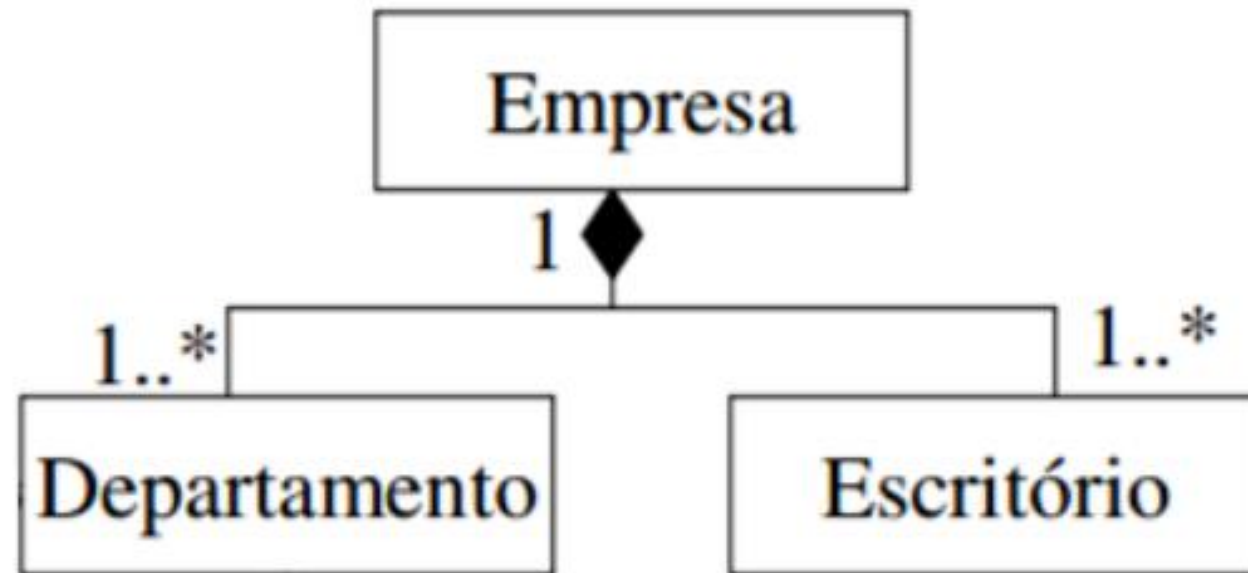
- É uma variante semanticamente mais “forte” da agregação.
- Os objetos “parte” só podem pertencer a um único objeto “todo” e têm o seu tempo de vida coincidente com o dele.



- Quando o “todo” morre todas as suas “partes” também morrem

Relacionamento: Composição

Ex:

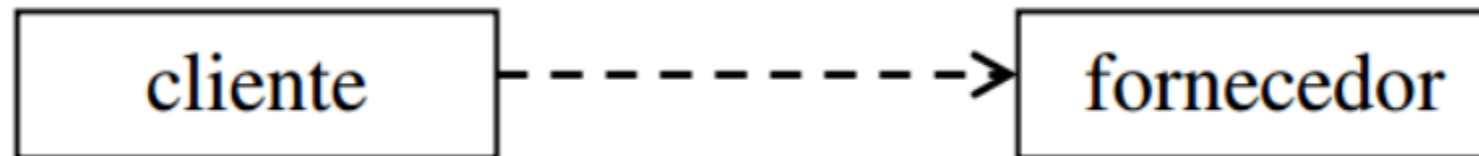


Agregação e Composição

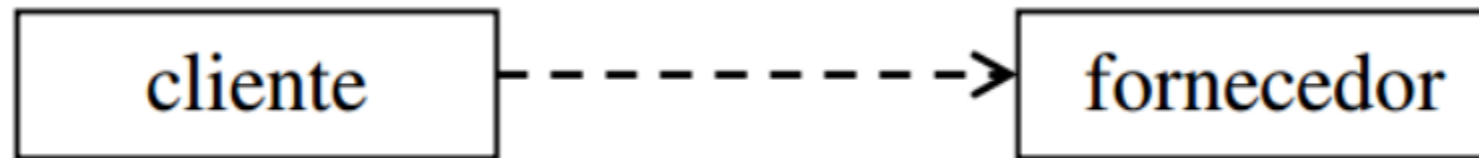


Relacionamento: Dependência

- Tipo menos comum de relacionamento
- Identifica uma ligação fraca entre objetos de duas classes
- Representado por uma reta tracejada entre duas classes
- Uma seta na extremidade indica o dependente
- Representa que a alteração de um objeto (o objeto independente) pode afetar outro objeto (o objeto dependente)



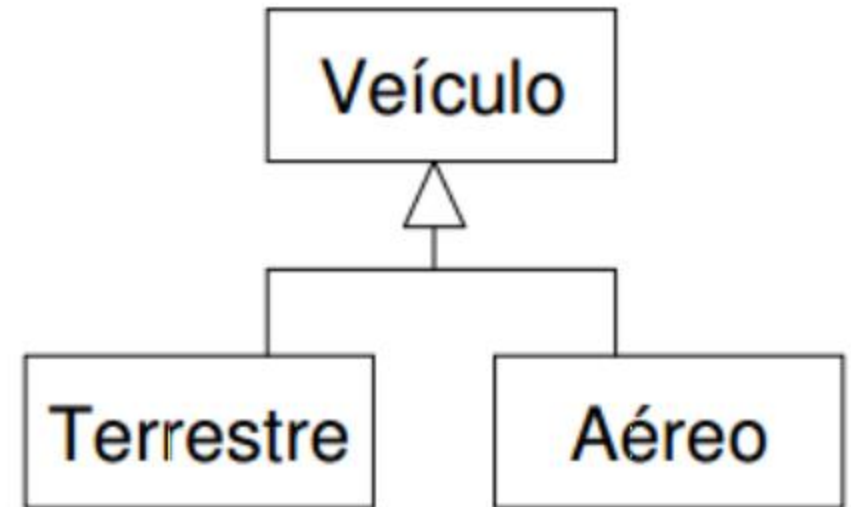
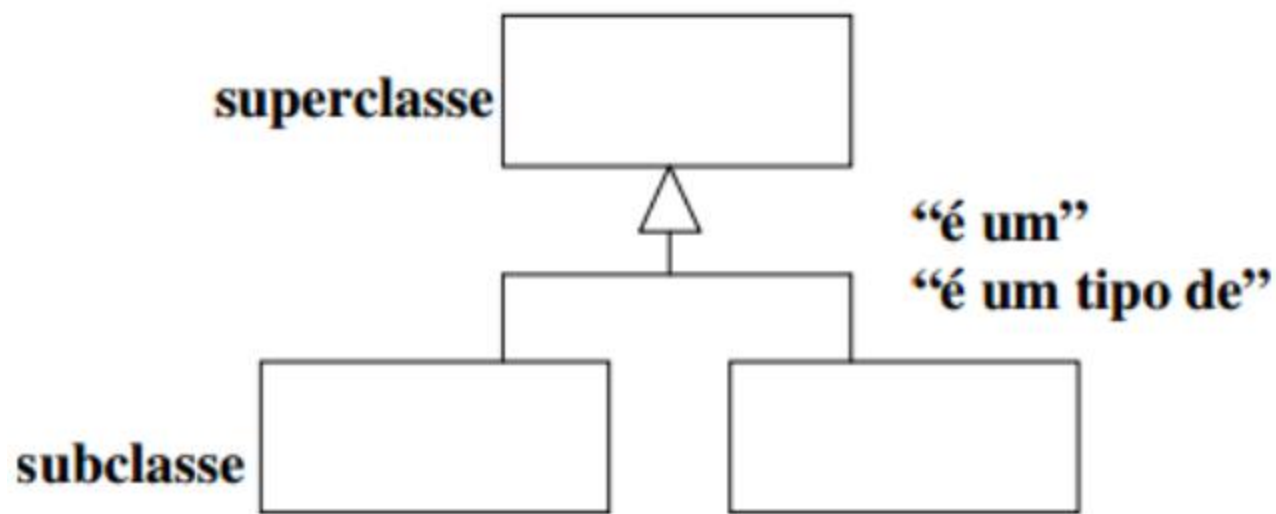
Relacionamento: Dependência



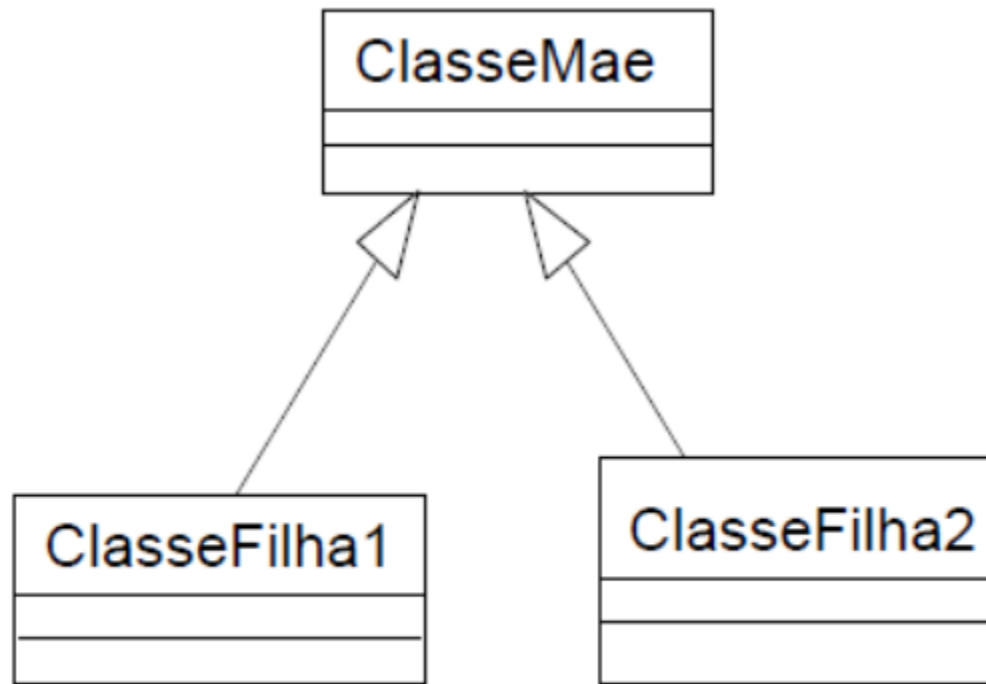
- A classe cliente depende de algum serviço da classe fornecedor
- A mudança de estado do fornecedor afeta o objeto cliente
- A classe cliente não declara nos seus atributos um objeto do tipo fornecedor
- Fornecedor é recebido por parâmetro de método

Relacionamento: Generalização

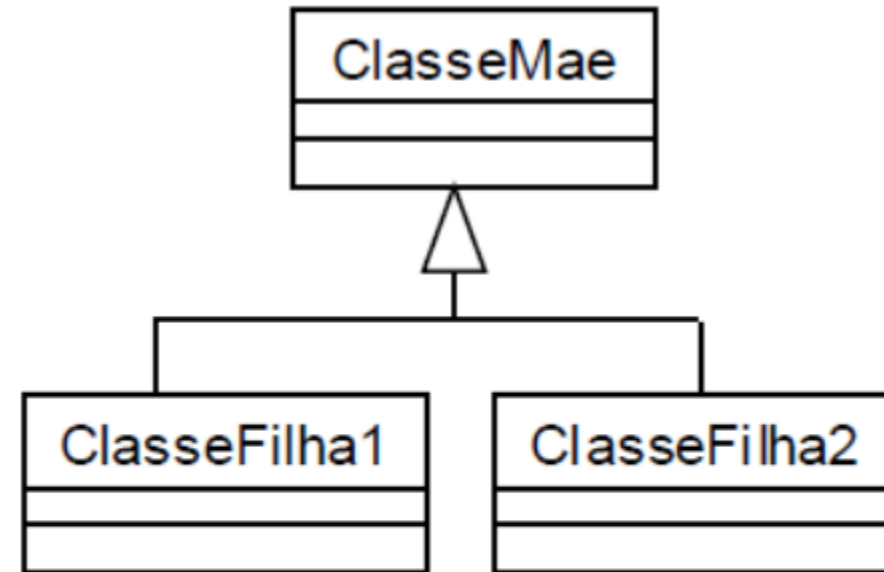
- É um relacionamento entre itens gerais (superclasses) e itens mais específicos (subclasses)



Relacionamento: Generalização



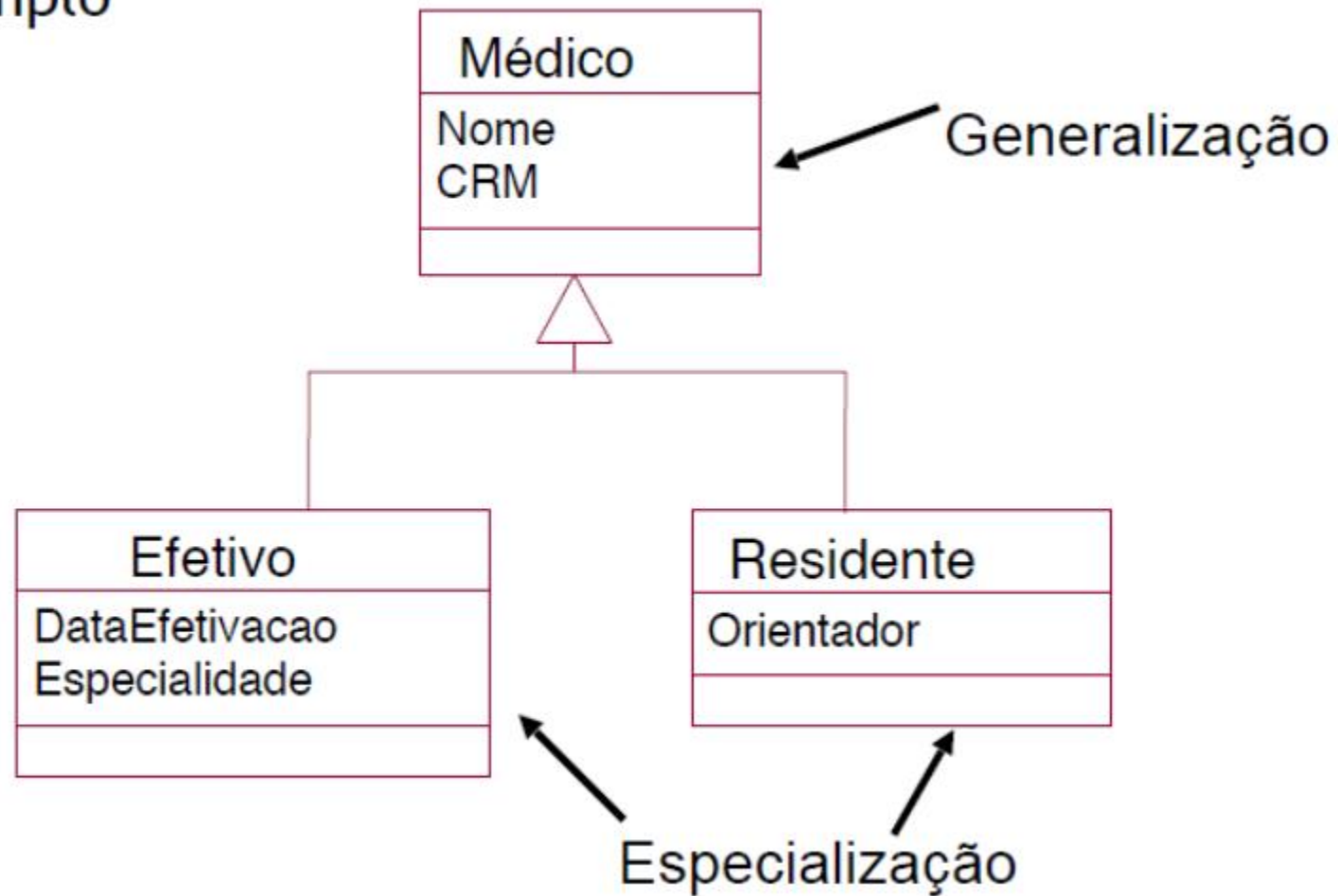
Estilo com setas separadas



Estilo com setas Agrupadas

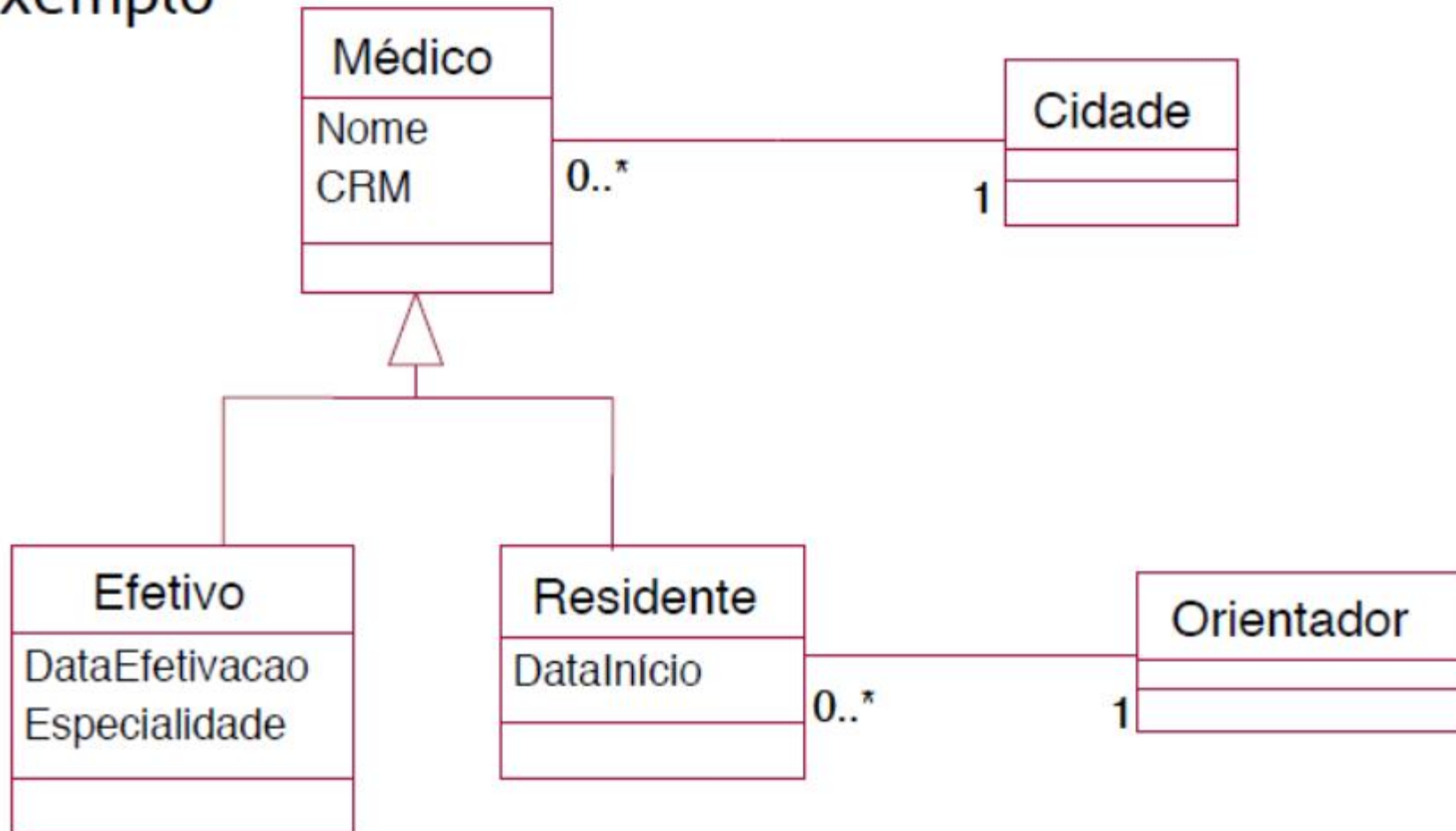
Relacionamento: Generalização

Exemplo

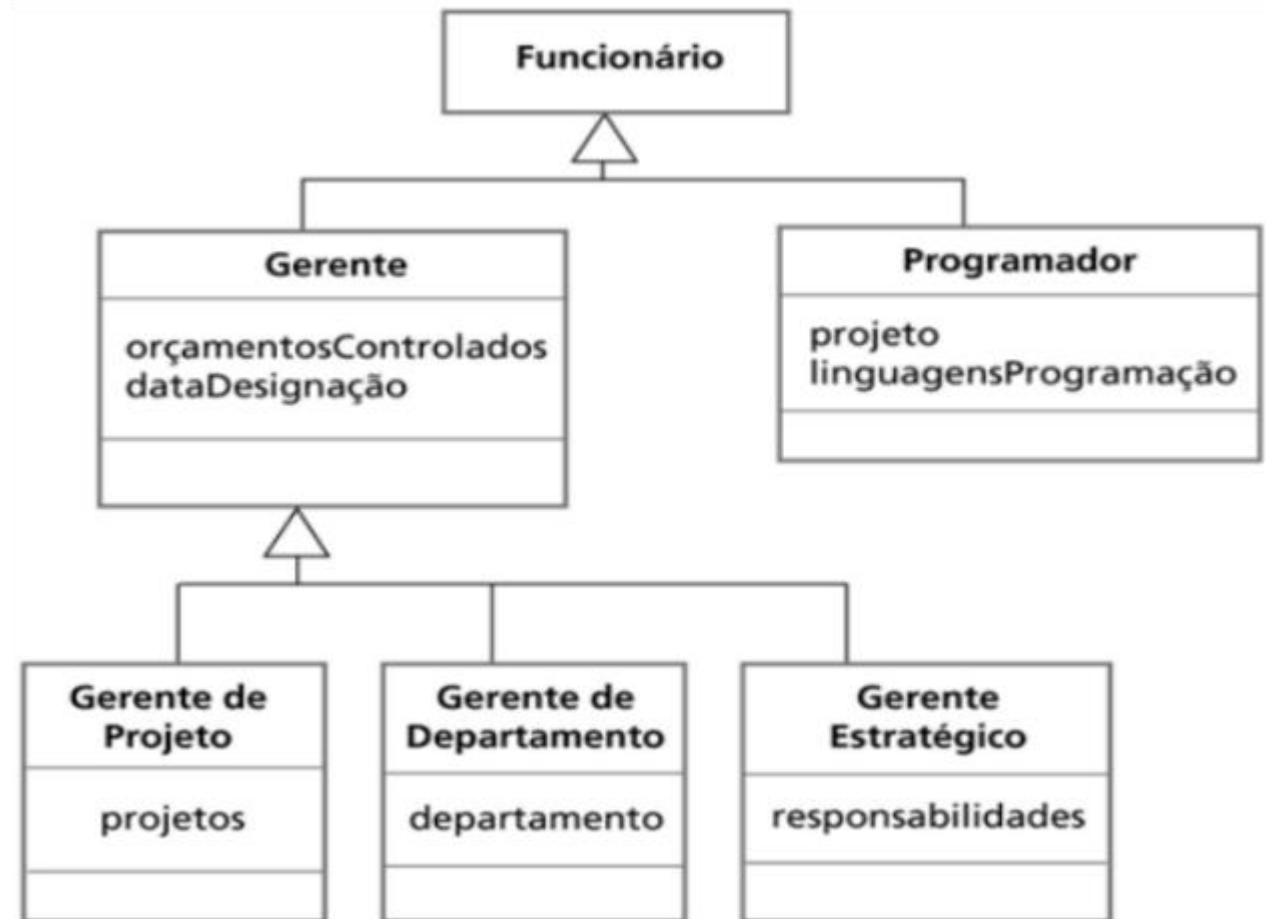
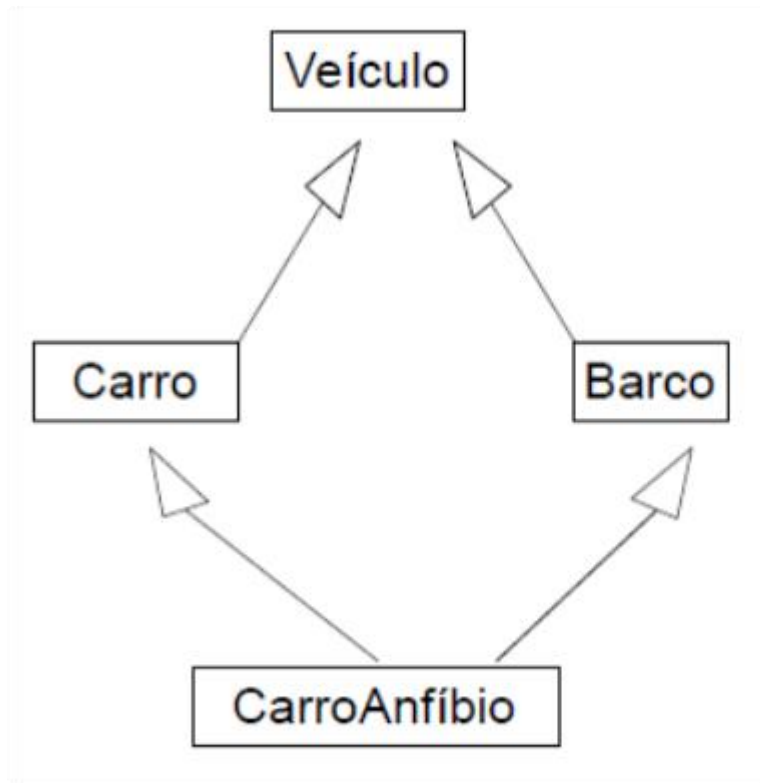


Relacionamento: Generalização

- Exemplo



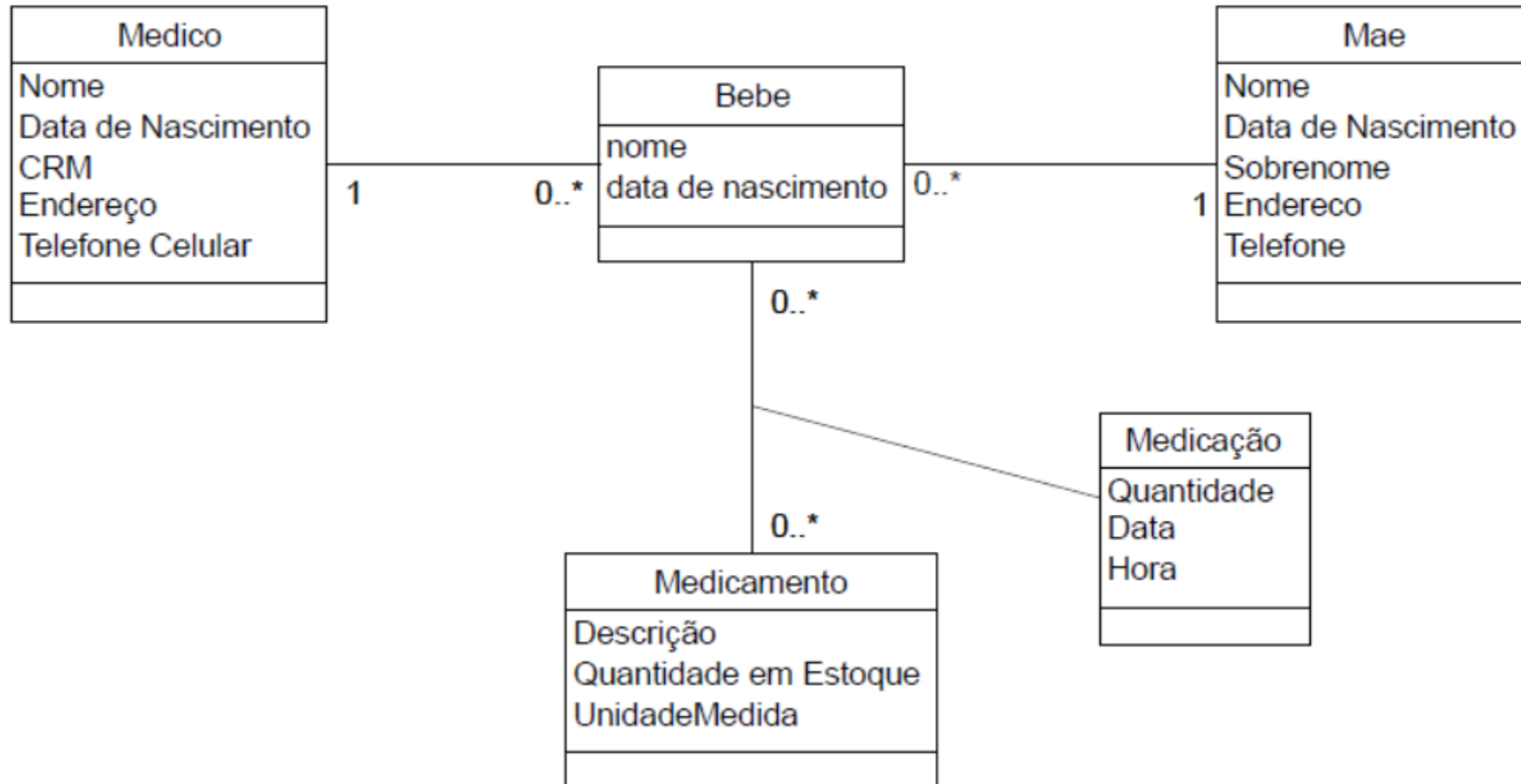
Relacionamento: Níveis de Generalização



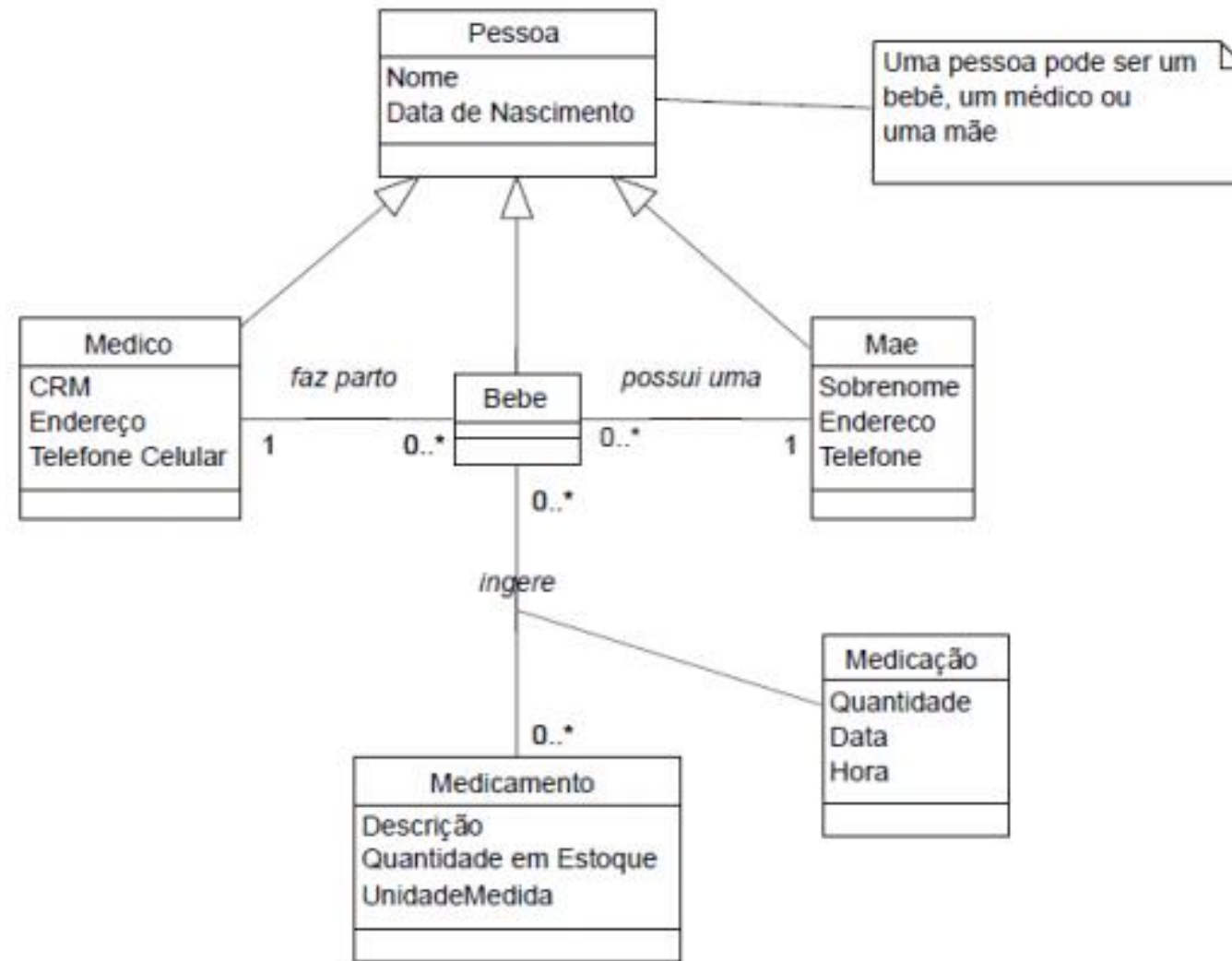
Exemplo

Um berçário deseja controlar suas tarefas. Para isso deseja manter um cadastro para os bebês, constando **nome, médico que fez seu parto, sua mãe e data de nascimento**. Para os médicos é necessário saber: **o nome, data de nascimento, CRM, endereço e telefone celular**. Para as mães dos bebês é necessário manter informações como: **nome, sobrenome, data de nascimento, endereço e telefone**. Ainda, é necessário manter um controle dos medicamentos ingeridos pelos bebês no berçário. Sobre os medicamentos é necessário manter: **descrição, quantidade em estoque e unidade de medida**. Um bebê pode tomar vários medicamentos, assim como um medicamento pode ser dado para vários bebês. Quando um bebê toma uma medicação ainda é importante saber a **quantidade, o dia e a hora do medicamento**.

Exemplo



Exemplo





Engenharia de Software I