
3. Modelação Conceptual de Classes

Teresa Galvão Dias



Faculdade de Engenharia da Universidade do Porto

Rua Dr. Roberto Frias, 4200-465 Porto

Tel.: 22-508 1639 • tgalvao@fe.up.pt

3. Modelação Conceptual de Classes

- 3.1 Introdução
- 3.2 Processo de Modelação Estática de Classes
- 3.3 Classes de Objectos, Atributos e Métodos
- 3.4 Ligações, Associações e Agregações Simples
- 3.5 Generalização e Herança Simples
- 3.6 Mapeamento entre o Modelo Conceptual e o Modelo Relacional
- 3.7 Conclusões e Principais Referências

Os modelos como representações abstractas da realidade

Os modelos são instrumentos para:

- comunicar objectivamente
- testar hipóteses
- prever o desempenho dos sistemas reais

Exemplos de modelos em engenharia:

- Carta topográfica
- Maqueta de um edifício
- Projecto estrutural de um edifício
- Desenho de um circuito integrado
- Conjunto de equações diferenciais
- Sequência de bases de uma proteína a sintetizar

Um Modelo usa uma Linguagem

Uma linguagem tem regras de utilização informais e formais.

Uma linguagem serve para comunicar efectivamente:

- deve ser conhecida pelos intervenientes, deve ser simples, clara e concisa.
- as suas expressões devem ter significados únicos, sem ambiguidade.
- as expressões devem ter uma Sintaxe e uma Semântica denotacional, ou matemática.

Os modelos obtidos devem ser **implementáveis informaticamente**, mas de forma independente do equipamento ou suporte lógico particular.

3. Modelação Conceptual de Classes

3.1 Introdução

3.2 Processo de Modelação Estática de Classes

3.3 Classes de Objectos, Atributos e Métodos

3.4 Ligações, Associações e Agregações Simples.

3.5 Generalização e Herança Simples

3.6 Mapeamento entre o Modelo Conceptual e o Modelo Relacional

3.7 Conclusões e Principais Referências

Metodologias de Modelação Orientadas por Objectos

OMT (Object Modeling Technique)

[Rumbaugh *et al* 1991] *Object-Oriented Modeling and Design*, Prentice-Hall

[Blaaha & Premerlani 1998]: *Object-Oriented Modeling and Design for Database Applications*, Prentice-Hall

UML (Unified Modeling Language)

[OMG-Object Management Group 2003]: *Unified Modeling Language Specification v1.5*.

<http://www.rational.com/uml/resources/documentation/index.jsp>

Modelação em UML – “*Unified Modeling Language*”

Utiliza os seguintes diagramas (UML v1.5)

Diagrama de **casos de utilização** ("use-cases") (1)

Diagrama de **classes** (baseado em OMT) (2)

Diagramas de comportamento:

Diagrama de estados ("statecharts") (3)

Diagrama de actividades (4)

Diagramas de interacção:

Diagrama de sequências (5)

Diagrama de colaborações (6)

Diagramas de Implementação:

Diagrama de componentes (7)

Diagrama de implantação (8)

Especificação dos requisitos do utilizador (Documento **UTIL**)

Requisitos: Serviços (funcionalidades) que o utilizador considera que o sistema deverá disponibilizar

Requisitos funcionais: descrevem o que o sistema deve fazer.

Requisitos não funcionais: descrevem a qualidade com que os requisitos funcionais são fornecidos (ex: medidas de performance, segurança, volume de dados).

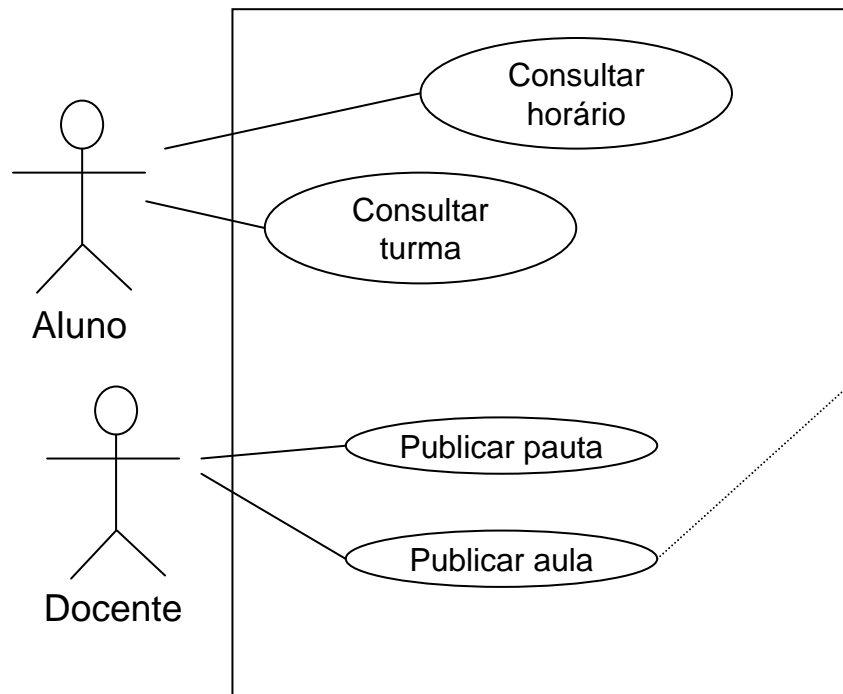
Requisitos de usabilidade: descrevem critérios e medidas que garantem uma boa interacção com o utilizador.

Técnicas de especificação de requisitos:

- reuniões (*workshops*)
- entrevistas
- questionários
- observação directa
- estudo de documentos e relatórios

UML – Diagramas de Casos de Utilização (use cases)

1. Identificação dos **actores**: entidades externas que interagem com o sistema
2. Identificação dos **casos de utilização**
3. Descrição dos casos de utilização: **cenários de utilização**



Publicar aula:

1. Depois de se validar no sistema, o professor poderá publicar as aulas
2. Cada aula é identificada pela disciplina associada, uma data e um número (opcional).
3. A cada aula poderão ser associados diversos documentos, ou url's
4. ...

Processo de Modelação Conceptual

Processo de abstracção através do qual se identificam os **objectos** (entidades e conceitos) relevantes para o sistema e se descrevem as suas características comuns: **propriedades** (atributos) e **comportamento** (operações)

Um modelo conceptual é um modelo **estático**, no sentido em que pretende capturar os requisitos de informação do sistema que são estáveis no tempo.

Exemplo: Descrição da empresa *OnTime*

“A empresa OnTime dispõe de uma frota própria de **viaturas** de carga, ligeiras e pesadas, incluindo reboques e atrelados. Dispõe ainda de viaturas que pertencem a **clientes**”.

(...)

“Cada **serviço** contratado é realizado num dado percurso, formado por uma sequência ligada de trajectos. Cada **percurso** tem origem e destino em **depósitos** de viaturas, podendo estes depósitos ser diferentes.”

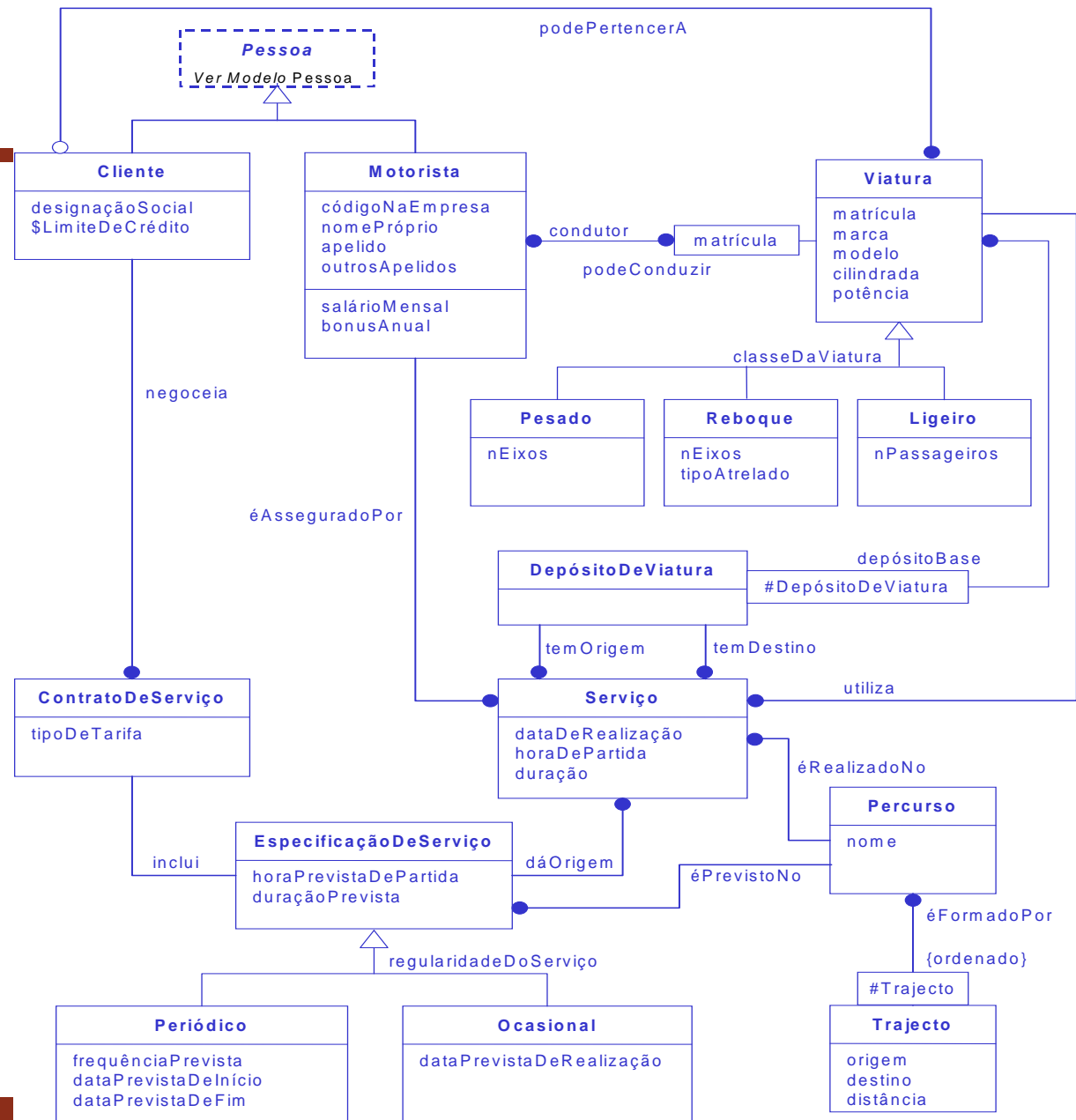
(...)

“Cada **trajecto**, com um nome único, é caracterizado por uma origem, um destino e uma distância em quilómetros. Por exemplo, o serviço OTN3456, previsto e realizado no percurso de Porto-Serralves para Braga-Centro, utilizou a viatura 34-98-DH e foi assegurado pelo Senhor Jorge Sampaio.”

(...)

Exemplo de Modelo

Modelo estático de classes do subsistema de gestão de transportes da empresa *On Time*



3. Modelação Conceptual de Classes

3.1 Introdução

3.2 Processo de Modelação Estática de Classes

3.3 Classes de Objectos, Atributos e Métodos

3.4 Ligações, Associações e Agregações Simples.

3.5 Generalização e Herança Simples

3.6 Mapeamento entre o Modelo Conceptual e o Modelo Relacional

3.7 Conclusões e Principais Referências

3.3 Classes de Objectos, Atributos e Métodos

- 3.3.1. Objectos e Atributos
- 3.3.2. Classes de Objectos
- 3.3.3. Operações e Métodos de Objectos
- 3.3.4 Domínios e Operações Complexas
- 3.3.5 Atributos derivados
- 3.3.6 Atributos de Classe e Operações de Classe
- 3.3.7 Atributos candidatos a identificadores

Modelação de Classes ([Blaha & Premerlani 1998; 127]):

Fases:

1. Identificar **Classes** (procurar nomes próprios e nomes comuns).
2. Identificar **Associações** (procurar verbos transitivos e preposições).
3. Acrescentar **Atributos**, detalhando as **Classes** e **Associações** (frases preposicionais possessivas).
4. Utilizar **Generalizações** para caracterizar semelhanças e diferenças entre objectos.
5. Testar caminhos de acesso e pesquisa de informação.
6. Iterar e refinar o modelo, acrescentando, eliminando ou alterando o detalhe ou nível de abstracção.
7. Organizar graficamente o modelo final.

Objectos e Atributos

Objecto: conceito com individualidade e identidade própria, normalmente identificado com um nome, sem ambiguidade num dado contexto ou situação. Podemos ter objectos concretos ou abstractos, mas um objecto é sempre único.

Os objectos estão normalmente associados a nomes próprios: Jorge Sampaio (motorista), Fernando Pessoa (poeta), Princesa Leia (Guerra das Estrelas), Teorema da Incompletude de Gödel.

Atributo: dimensão de observação que nos permite dar um valor a um objecto, valor esse que pertence a um conjunto bem definido. Os valores não têm identidade, ao contrário dos objectos, devido a serem abstracções matemáticas perfeitas.

Os atributos estão normalmente associados aos adjectivos, na descrição do problema.

Exemplos de Objectos e Atributos

FPessoa: Poeta

nome = Fernando Pessoa
dataDeNascimento=1888.06.13

PLeia: PersonagemDeFilme

nome = Princesa Leia
primeiroFilme=Star Wars

V3498DH:Viatura

matrícula = 34-98-DH
marca = Volvo
modelo = B400
cilindrada = 5000 cc
potência = 300 CV

Heathrow: Aeroporto

código = LHR
nome = Heathrow Airport
cidade = Londres

FSCarneiro: Aeroporto

código = OPO
nome = Aeroporto Francisco Sá Carneiro
cidade = Porto

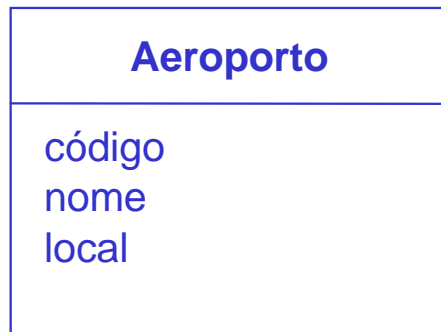
Classes de Objectos

Classe de objectos: caracterização formal de um grupo de objectos semelhantes, no que eles têm de comum.

Os objectos são instâncias de uma dada classe.

- Devido à necessidade de identificar cada objecto unicamente em todo o sistema, tem de existir um atributo externo, ou conjunto de atributos externos, que o permita fazer. Tais atributos são designados chaves candidatas a identificador.
- Contudo, por razões de implementação, deve-se sempre vir a ter para cada classe um único atributo identificador, interno ao sistema que a virá a implementar. Os códigos identificadores externos ou públicos, tais como o número do Bilhete de Identidade ou o código de um motorista na empresa, estão sujeitos a ser alterados com o tempo, não sendo adequados a servir de referências internas.

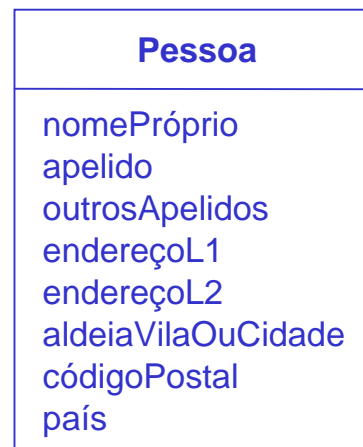
Exemplos de Classes de Objectos



Classe Pessoa
para análise



Classe Pessoa
para concepção



Operações e Métodos de Objectos

As **operações** são funções ou procedimentos definidos no âmbito de um objecto com determinados argumentos e resultado, podendo ser implementados como **métodos** da respectiva classe.

Método ***salárioMensal*** de objectos da Classe *Motorista*. Este método permite calcular o salário mensal de cada motorista, por exemplo, com base nas horas efectivamente trabalhadas.



Domínios e Operações Complexas

Dados primitivos:

tipo Natural (abreviado Nat), Inteiro (abreviado Int), Real, conjunto enumerado, ...

Um domínio corresponde a um tipo de dados definido a partir dos tipos primitivos através de construções simples.

Um domínio é um conjunto de valores

A noção formal de domínio corresponde à noção tradicional de um conjunto a que se adiciona normalmente um elemento com valor indefinido.

Exemplos de Domínios e Operações Complexas

- Indicação de Domínios de valores para os atributos e os métodos.
- Indicação de valores por defeito para os atributos *pesoEsquerdo* e *pesoDireito*.

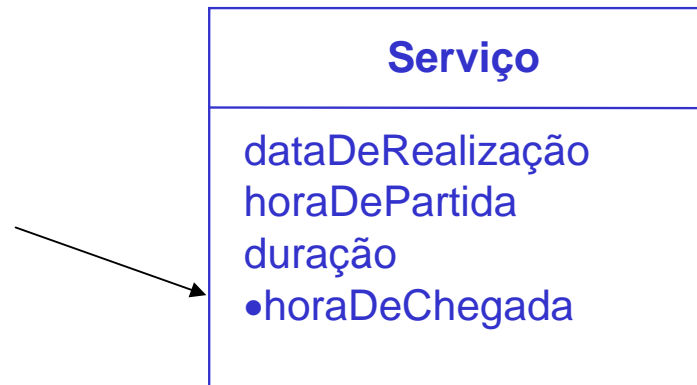
NodoDeÁrvoreBinária
identificadorPúblico {CC1} pesoEsquerdo: Int = 1 pesoDireito: Int = 1 valorDoNodo: Real
médiaPesos: Int x Int → Real pesoMáximo: Int x Int → Int

Atributos Derivados

Um **atributo derivado** contém informação que pode ser facilmente calculada a partir da informação presente em outros atributos.

Em termos de implementação, por exemplo recorrendo a um SGBDr, coloca-se a questão de tornar um atributo derivado persistente ou de calcular o seu valor no momento de consulta pelo utilizador.

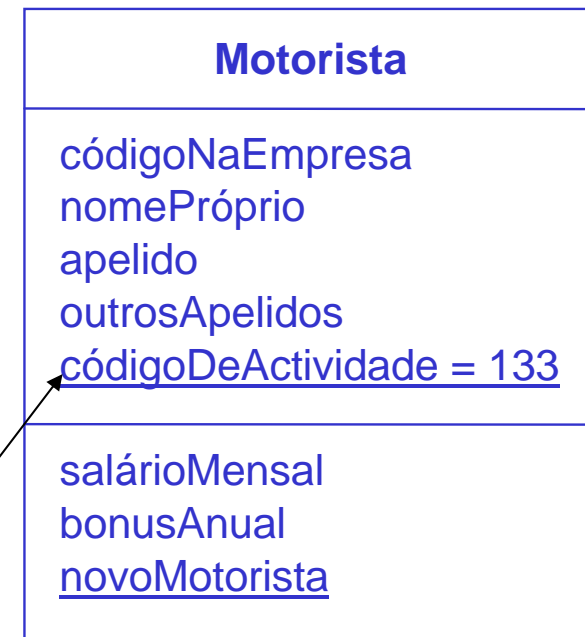
O atributo *horaDeChegada* pode ser calculado a partir dos atributos *horaDePartida* e *duração*.



Atributos de Classes e Operações de Classe

Atributos de classe: úteis para manter informação comum a todos os objectos da classe, como parâmetros ou constantes.

No entanto não é normalmente uma boa solução recorrer a estes atributos, pois existe sempre a possibilidade de definir outra classe, correspondente aos parâmetros ou constantes, e recorrer a associações para manter a caracterização desejada.

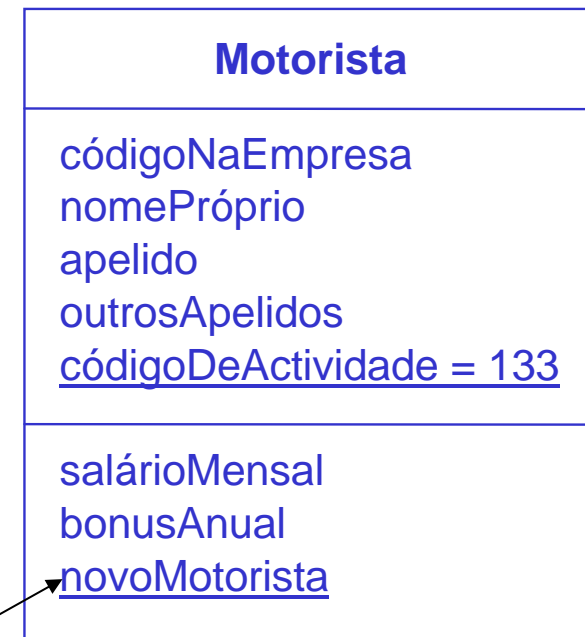


Atributo de classe

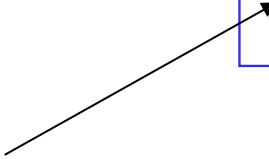
Atributos de Classes e Operações de Classe

Operações de classe: úteis para criar um novo objecto

Uma operação deste tipo tem como domínio a própria classe e não objectos ou seus atributos.



Operação de classe



Atributos candidatos a identificadores

Cada classe pode ter um atributo externo que permita **identificar** todas as suas instâncias, isto é nomear cada um dos possíveis objectos da classe, sem ambiguidade.

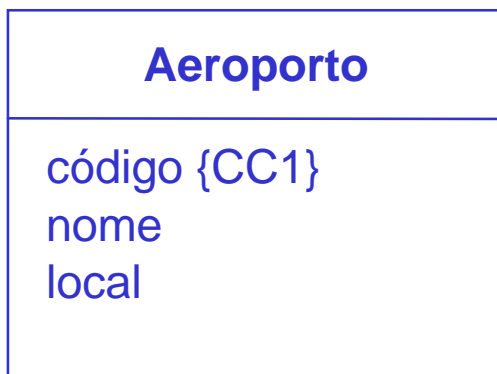
Pode haver mais do que um atributo que satisfaça esta condição, e assim teremos **vários** identificadores candidatos.

Podemos também ter necessidade de um conjunto de atributos para atingir este objectivo, pelo que então teremos um **identificador composto**. Neste caso os atributos têm que ser em número mínimo. Isto é, se retirando do conjunto um ou mais atributos se mantiver a possibilidade de identificação, então será este conjunto menor o identificador.

Exemplos de atributos candidatos a identificadores

Identificador Simples:

o *código* identifica unicamente um dado aeroporto.



Identificador Composto:

o código ISBN e o número de cópia do livro (os dois em conjunto) identificam unicamente um dado livro numa biblioteca.



3. Modelação Conceptual de Classes

1. Introdução
2. Processo de Modelação Estática de Classes
3. Classes de Objectos, Atributos e Métodos
4. Ligações, Associações e Agregações Simples.
5. Generalização e Herança Simples
6. Mapeamento entre o Modelo Conceptual e o Modelo Relacional
7. Conclusões e Principais Referências

3.4 Ligações, Associações e Agregações Simples

- 3.4.1. Ligações entre Objectos
- 3.4.2. Associações entre Classes
- 3.3.3. Atributos ou Classes em Associações
- 3.3.4 Associações qualificadas
- 3.4.5. Agregação entre Classes

Ligações entre objectos

Uma vez classificados os grupos de entidades, como objectos e classes, deve observar-se como é que estes se **relacionam** uns com os outros.

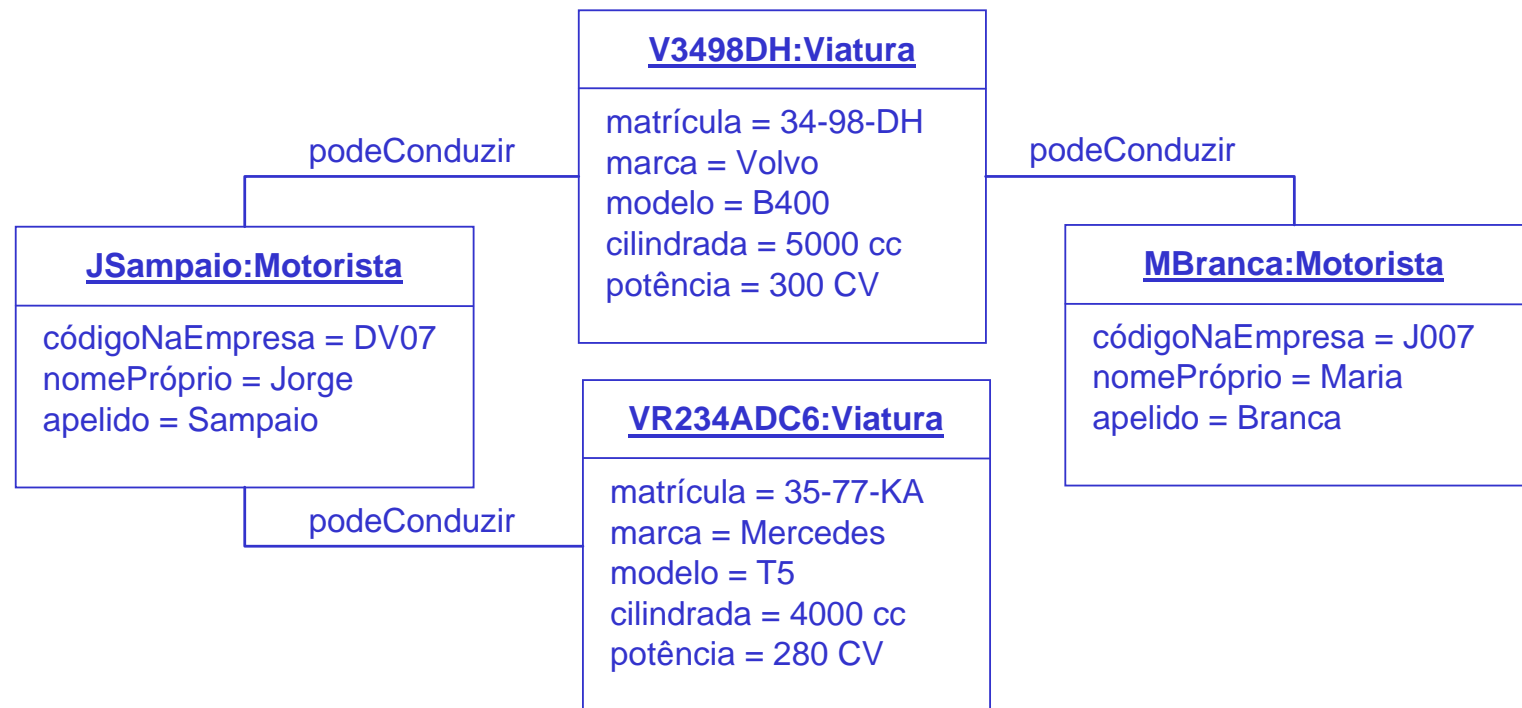
Há entidades que são autónomas entre si, mas que interagem ou comunicam, há outras que se agrupam para formar entidades mais complexas, e há ainda outras que nós próprios criamos, tendo normalmente uma existência abstracta dependente da de outras entidades.

Exemplo: numa faculdade, onde existem professores, alunos, funcionários, cursos disciplinas, horários, pautas, como se relacionam estas entidades entre si?

Exemplos de ligações entre objectos

Estabelece-se uma **ligação** entre 2 ou mais objectos quando é importante manter a informação que os relaciona de alguma forma relevante para a situação em estudo.

Exemplo de ligações *podeConduzir* entre objectos das classes Motorista e Viatura:

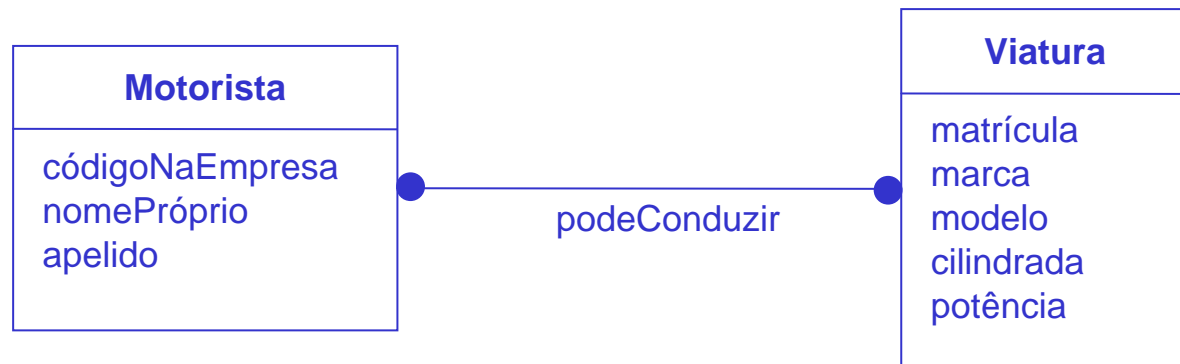


Associações entre classes

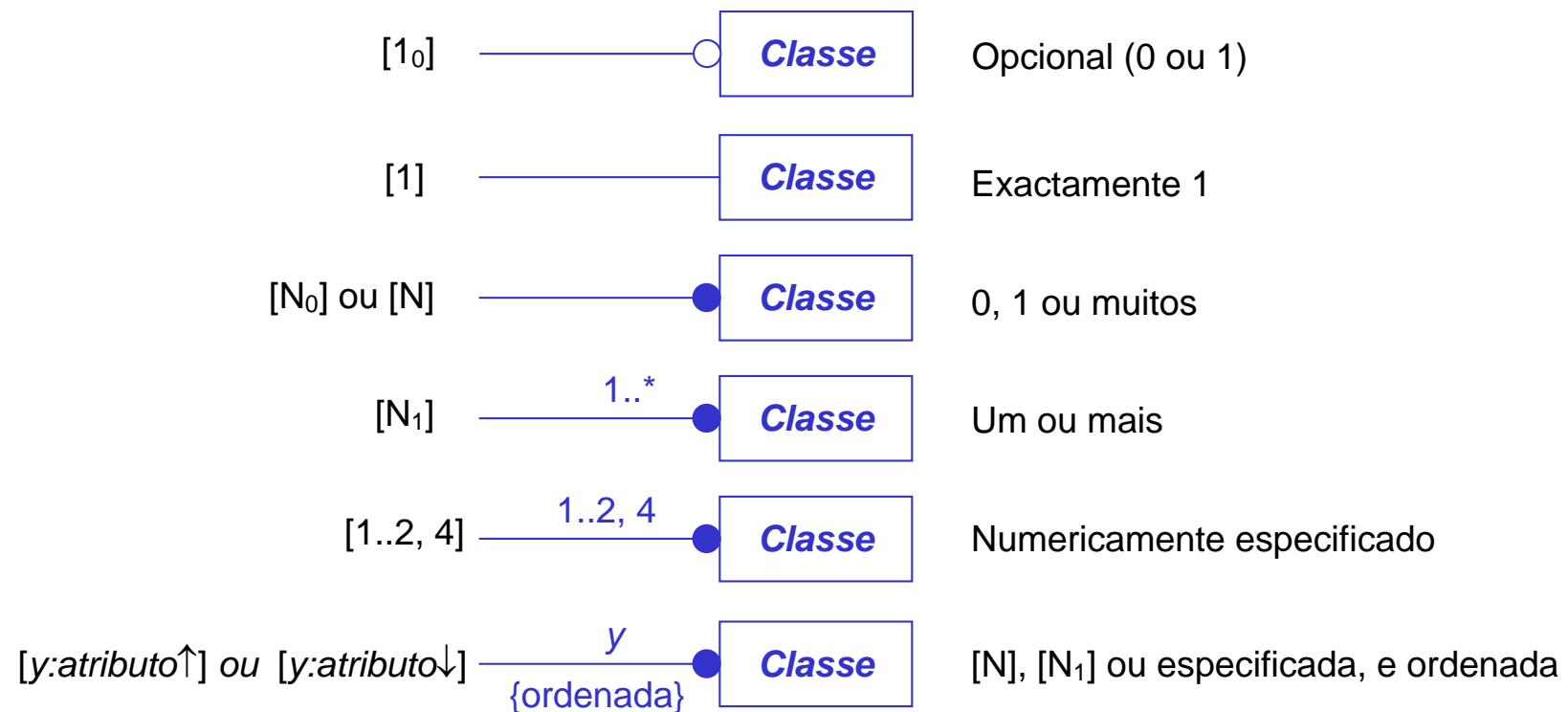
Uma **associação** entre duas ou mais classes corresponde à caracterização e tipificação de ligações semelhantes entre objectos dessas classes.

Assim como um objecto é uma instância de uma classe, uma ligação é uma instância de uma associação.

Exemplo de associação
podeConduzir entre
classes Motorista e
Viatura:

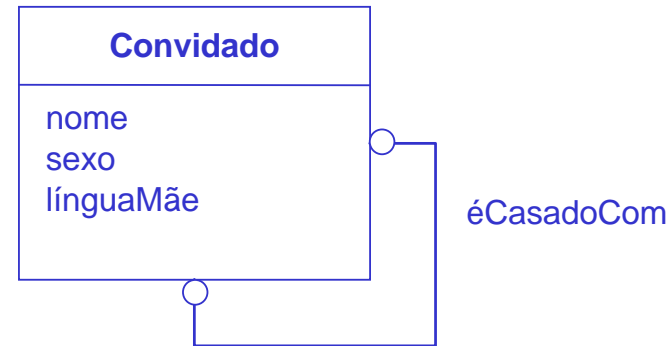


Multiplicidade de Associações



Exemplos de Associações Unárias

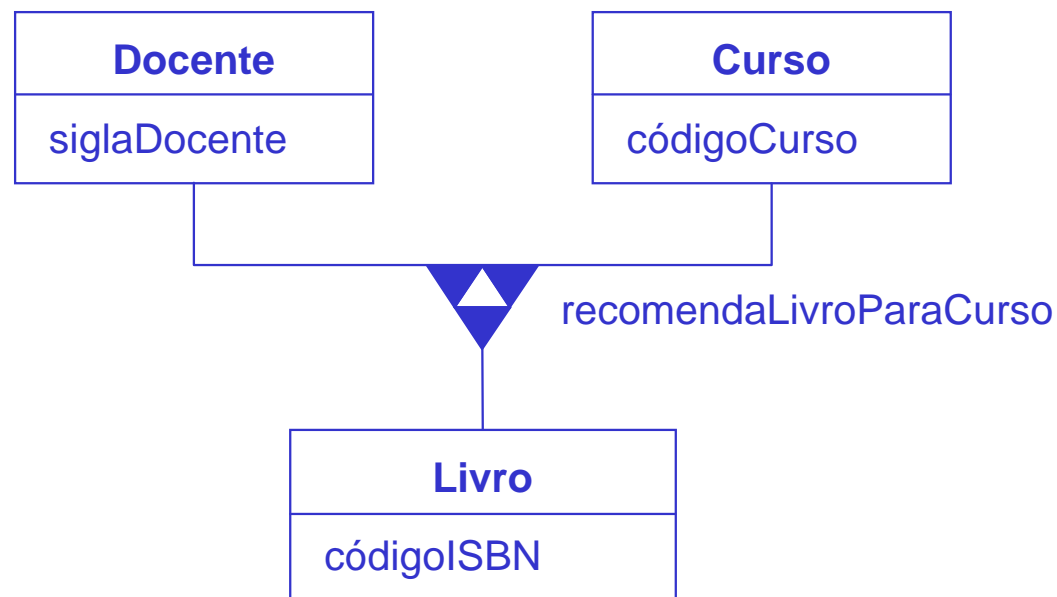
Exemplo de uma associação unária (ou reflexiva) relativa ao sistema para atribuição de lugares nas mesas aos convidados de uma empresa de gestão de recepções:



⊗ *Modelo Incorrecto*



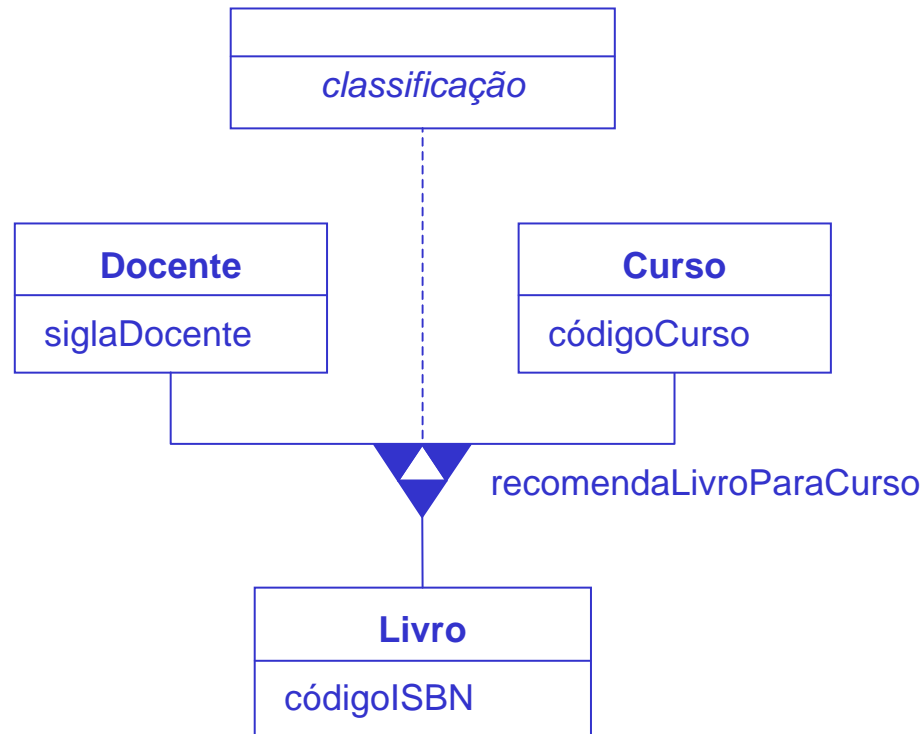
Exemplos de Associações Ternárias



Atributos ou classes em associações

- É possível haver atributos e classes em associações.
- Tal é normal e necessário, em particular, nas associações muitos-muitos, onde o atributo é claramente uma propriedade da associação.
- Numa associação de um-um ou um-muitos, o atributo pode normalmente ser considerado numa das classes. **Porquê?**
- Se uma dada associação puder participar em novas associações, deve ascender ao estatuto de classe da associação. **Porquê?**

Exemplo de um atributos numa associação



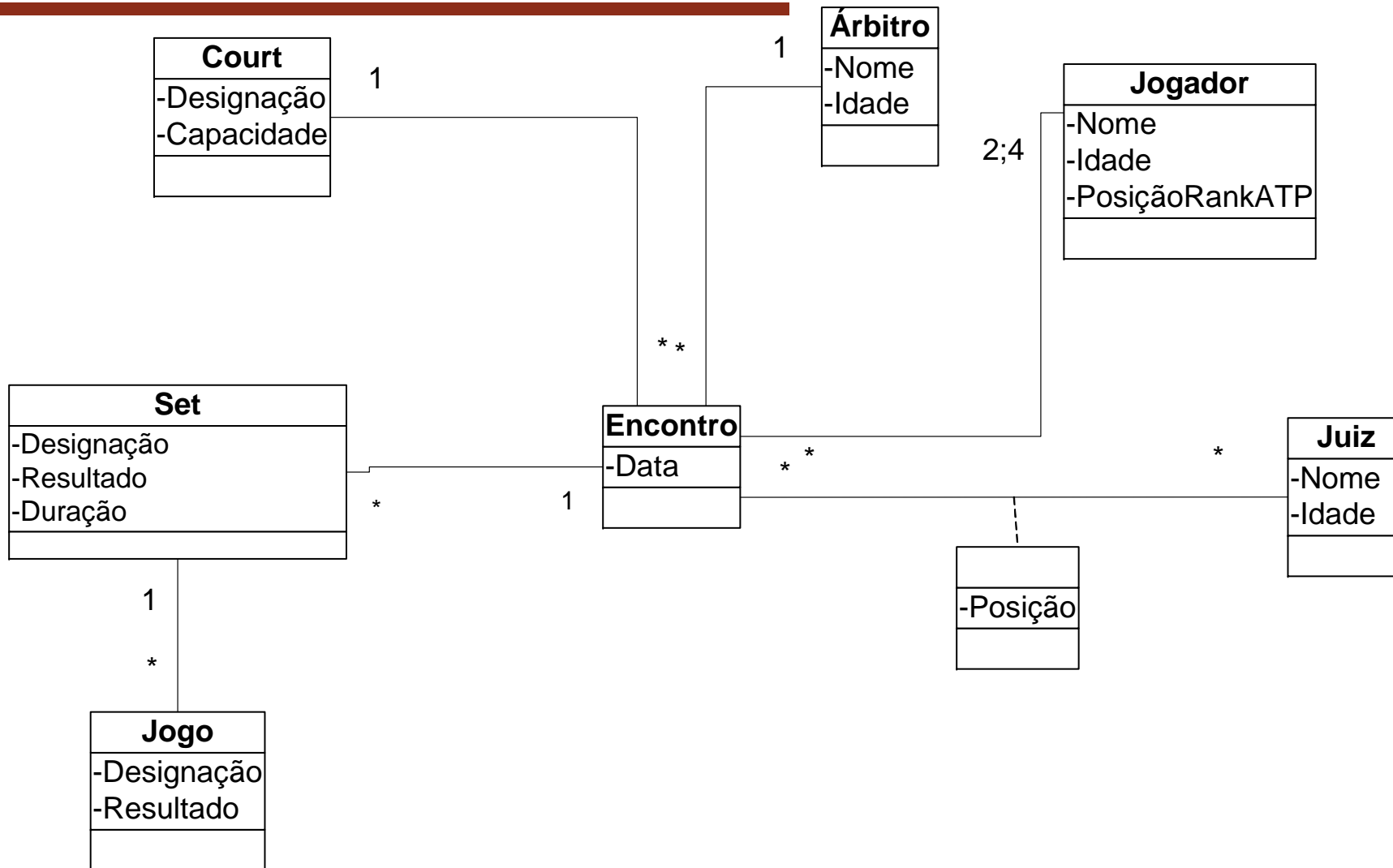
Exercício

Imagine que pertence à direcção de um clube de ténis como o de Wimbledon e que verificou a necessidade de implementar um sistema de informação com o objectivo de facilitar a gestão dos resultados dos torneios de ténis aí realizados.

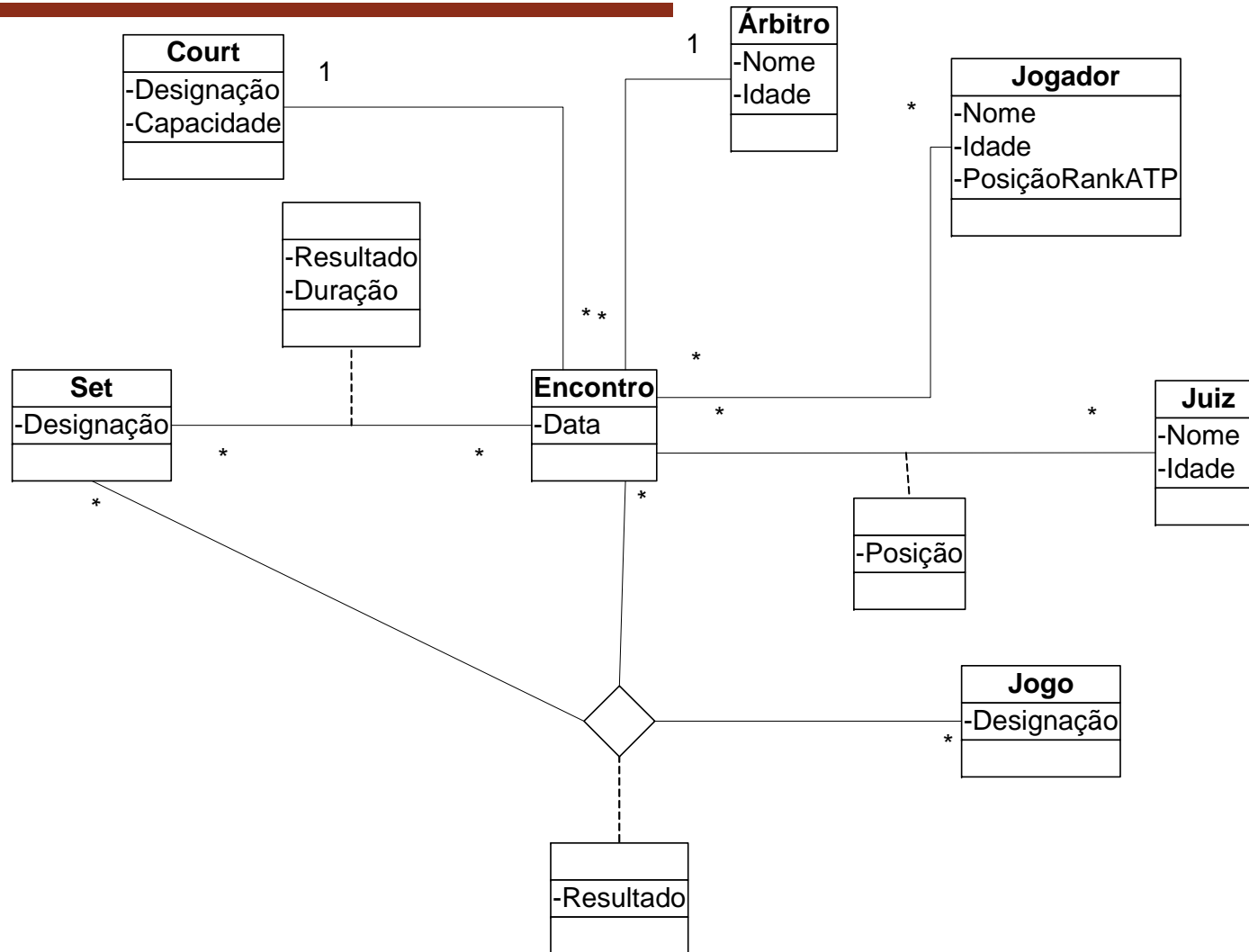
Sabendo que um encontro de ténis é composto por *Sets*, e que cada *Set* é formado por um conjunto de jogos, pretende-se saber o resultado e duração de cada *Set* referente a um dado encontro. Para cada *Set* de um dado encontro pretende-se guardar o resultado de cada um dos seus jogos.

Para um dado encontro interessa ainda saber qual a sua data de realização, em que *Court* se realizou, quem foi o árbitro, quem foram os juizes de linha e quais foram as suas posições respectivas (linha de fundo, linha de serviço, etc.). Pretende-se ainda saber qual o nome e idade de cada jogador, árbitro e juiz de linha, e também a posição no *Ranking ATP* de cada um dos jogadores. Relativamente aos *Courts* interessa saber a lotação máxima de cada um.

Resolução 1



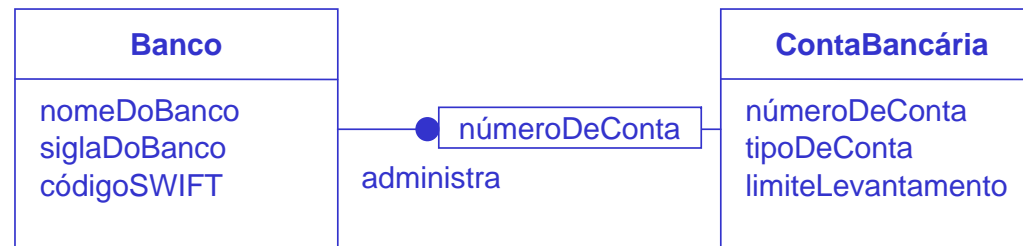
Resolução 2



Associações qualificadas

Uma **associação qualificada** é uma associação na qual os objectos da classe alvo (lado "muitos") são identificados totalmente ou parcialmente através de um atributo designado por qualificador.

Associação qualificada



Associação não qualificada

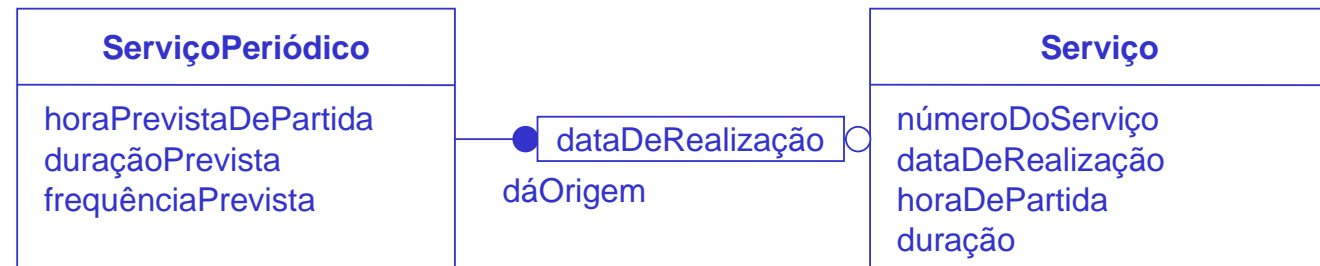


3.3.4 Associações qualificadas

As associações “um-para-muitos” ou “muitos-para-muitos” podem ser qualificadas. Um qualificador selecciona os objectos da classe alvo, reduzindo a multiplicidade da associação.

Na fase de implementação podem ser utilizados índices associados aos qualificadores, o que simplifica as pesquisas

Associação qualificada



Associação não qualificada



3.4.5 Agregação entre Classes

A **agregação** é um tipo especial de associação, utilizada por exemplo para indicar a composição física de um tipo de produto, ou a estrutura de organização de uma entidade.

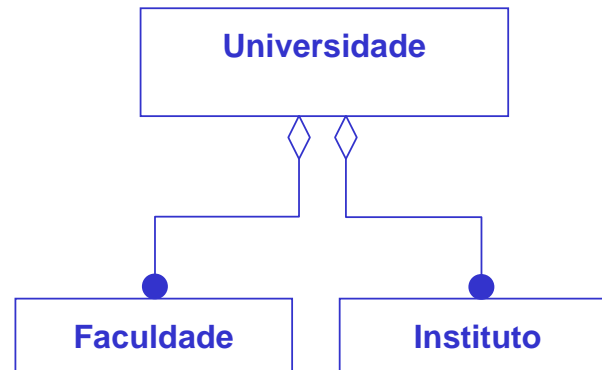
O conjunto de ligações entre os objectos que a agregação descreve, visto como uma relação matemática, caracteriza-se pelas propriedades de transitividade e anti-simetria.

Transitividade: Se A faz parte de B e B faz parte de C, então A faz parte de C

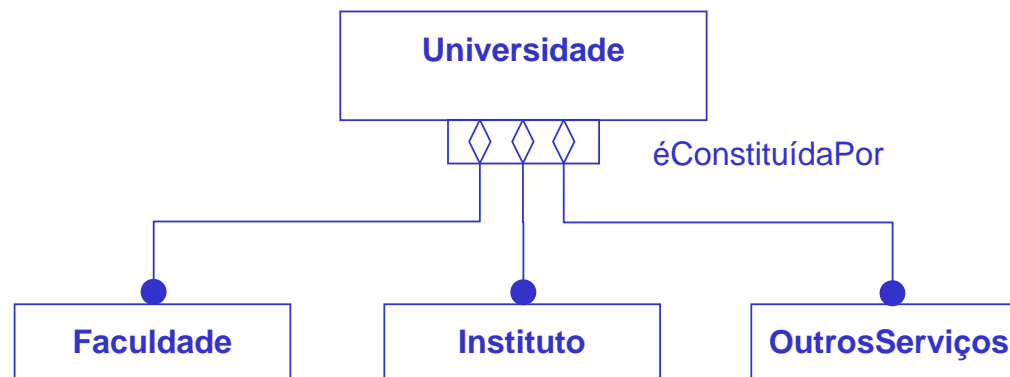
Anti-simetria: Se A faz parte de B, então B não faz parte de A

Agregações abertas e fechadas

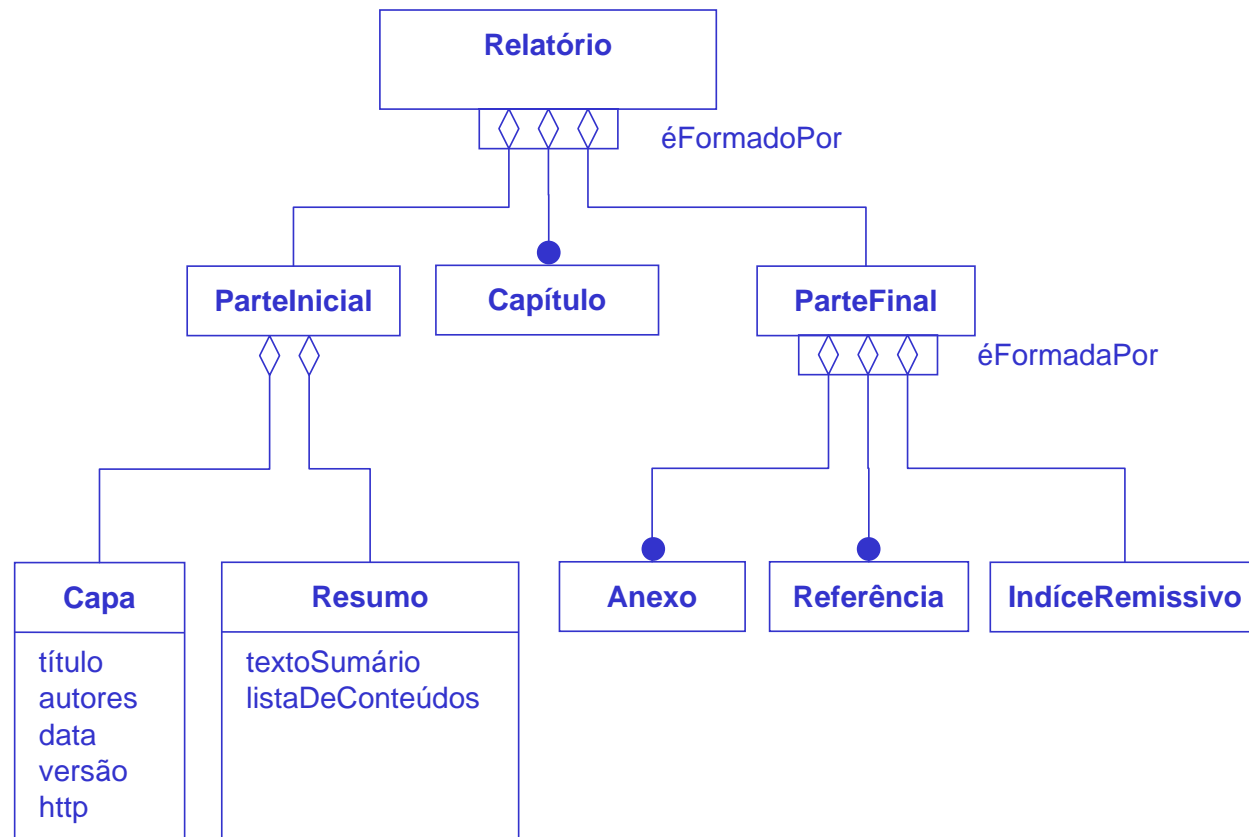
Agregação aberta, indicando algumas classes constituintes



Agregação fechada, indicando todas as classes constituintes



Exemplo de Agregação entre Classes

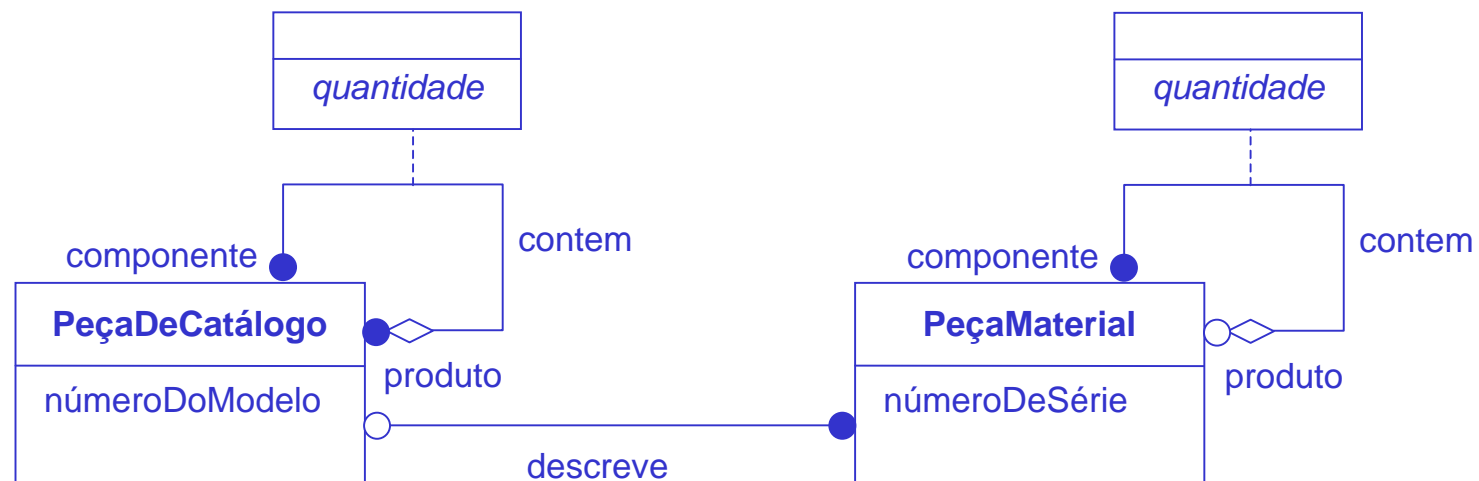


Agregação de catálogo e de produto

Agregação de catálogo: uma peça pode pertencer a vários produtos

Agregação de produto: uma peça apenas pode pertencer a um produto

Exemplo



Agregação entre classes

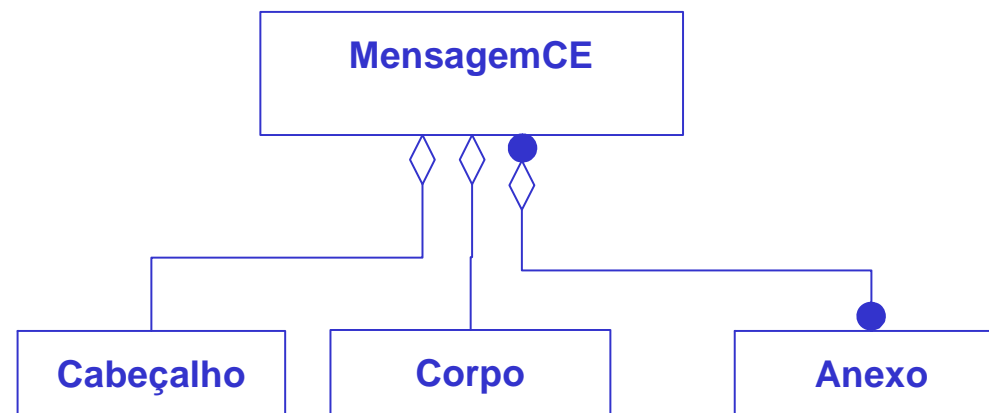
Exercício

Desenhe o modelo conceptual para uma mensagem de correio electrónico constituída por cabeçalho, corpo e anexo.

Agregação entre classes

Exercício

Desenhe o modelo conceptual para uma mensagem de correio electrónico constituída por cabeçalho, corpo e anexo.



3. Modelação Conceptual de Classes

1. Introdução
2. Processo de Modelação Estática de Classes
3. Classes de Objectos, Atributos e Métodos
4. Ligações, Associações e Agregações Simples.
5. Generalização e Herança Simples
6. Mapeamento entre o Modelo Conceptual e o Modelo Relacional
7. Conclusões e Principais Referências

3.5. Generalização e Herança Simples

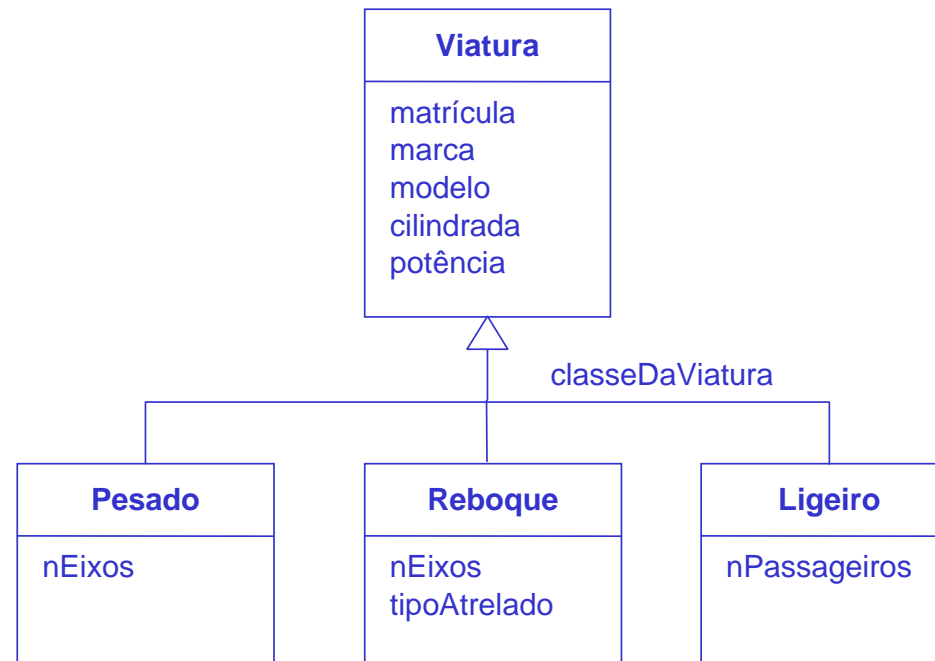
- 3.5.1. Generalização ou Particularização (exclusiva)
- 3.5.2. Classes Concretas e Abstractas
- 3.5.3. Particularização Inclusiva

3.5.1 Generalização ou Particularização (exclusiva)

- O mecanismo ou conceito de **generalização** permite-nos identificar numa classe geral tudo o que é comum a várias classes diferentes, mantendo em classes particulares as características distintas.
- As noções de associação e generalização capturam as duas formas mais tradicionais de análise e especificação de sistemas:
 - a **associação** indica-nos a organização do sistema através da sua decomposição em partes;
 - a **generalização** permite-nos identificar semelhanças entre sistemas ou seus componentes através da uma decomposição em tipos.

3.5.1 Generalização ou Particularização (exclusiva)

Podemos analisar as viaturas utilizadas por uma empresa nos seus diversos tipos, por exemplo pesados e ligeiros.



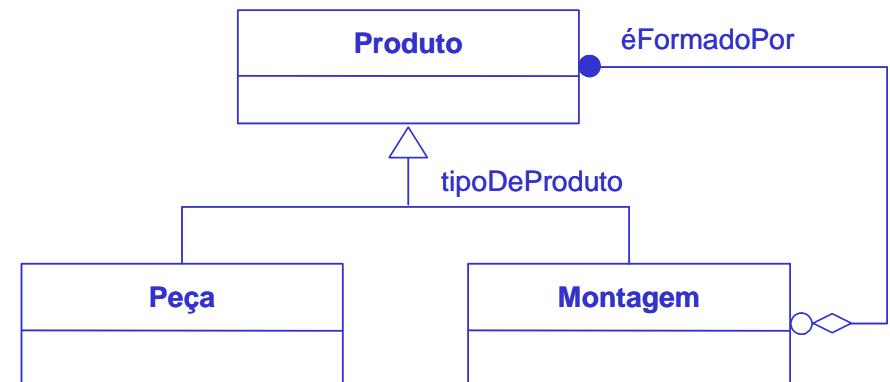
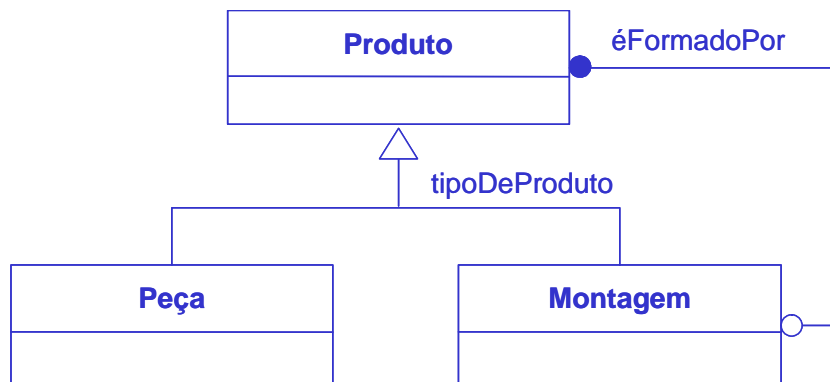
Diz-se que a particularização é **exclusiva** quando uma viatura pode ser apenas de uma e só uma das classes particulares.

3.5.1 Generalização ou Particularização (exclusiva)

- Podemos também alargar esta análise a níveis mais gerais: uma viatura é um bem móvel, e um bem móvel é um bem patrimonial. Temos assim uma hierarquia.
- A utilidade de qualquer hierarquia esgota-se na sua capacidade de resumir ou explicar conhecimento. Sendo assim aplicam-se as regras normais, pelo que não é natural permitir por exemplo menos do que 3 ou mais do que 7 particularizações distintas para cada classe de objectos, num dado nível.
- Há alguns problemas de implementação, dependentes do sistema a utilizar. Por exemplo na conversão para o modelo relacional, prévia à utilização de uma SGBDr, há que ter em consideração a quantidade de atributos e operações particulares de cada classe antes de decidir se se deve manter uma relação e correspondente tabela particular, ou se pelo contrário a relação ou tabela geral deve ser alargada para incluir os atributos particulares. A decisão depende obviamente dos volumes de informação a gerir, do tipo de operações a solicitar ao sistema e dos tempos de resposta que os utilizadores podem esperar.

Exercício

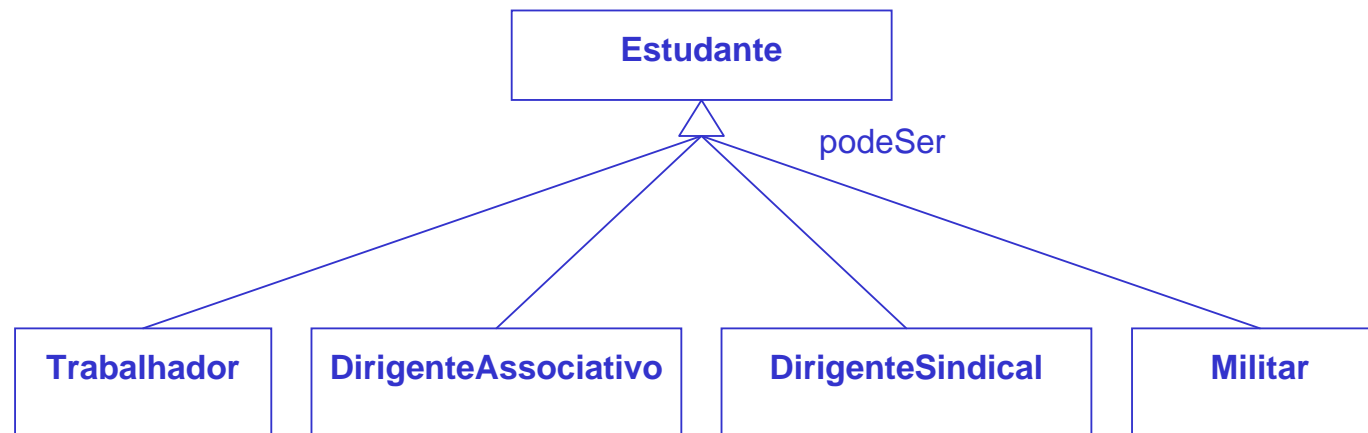
Num sistema de gestão de existências considere o problema de modelar a informação sobre um dado produto e os seus componentes. Suponha que um produto pode ser constituído por uma peça (simples) ou por uma montagem (conjunto de componentes). Proponha dois modelos para este problema recorrendo no primeiro caso apenas a uma associação e generalização e no segundo caso a uma agregação e generalização. Explique a diferença entre os dois modelos



3.5.2 Classes Concretas e Abstractas

- Uma classe (super-classe) diz-se **abstracta** quando não tem objectos (instâncias de classe). Por exemplo, quando no todo ou em parte não corresponde a nenhuma classe que irá ser implementada.
- No final de uma hierarquia de particularização haverá sempre classes concretas, isto é classes com instâncias a que correspondem directamente objectos.
- Por exemplo, suponhamos que foi definida uma super-classe Pessoa e respectivas sub-classes Pessoa Singular e Pessoa Colectiva. Considerando que é decidido apenas implementar estas duas últimas, a classe Pessoa é uma classe abstracta, visto não existirem instâncias ou objectos da classe Pessoa.

3.5.3 Particularização Inclusiva



Diz-se que a particularização é **inclusiva** porque o estudante pode pertencer a várias das classes particulares.

Conclusões

1. Identificar **Classes** (procurar nomes próprios e nomes comuns)
2. Identificar **Associações** (procurar verbos transitivos e preposições)
3. Acrescentar **Atributos**, detalhando as Classes e Associações (procurar frases preposicionais possessivas)
4. Utilizar **Generalizações** para caracterizar semelhanças e diferenças entre objectos
5. Testar **caminhos de acesso e pesquisa** de informação
6. **Iterar e refinar** o modelo, acrescentando, eliminando ou alterando o detalhe ou nível de abstracção
7. **Organizar graficamente** o modelo final