

## **SISTEMAS DE INFORMAÇÃO**

Modelação do Conhecimento e Bases de Dados

# **3.** *Modelação Conceptual de Classes*

---

# 3. Modelação Conceptual de Classes

---

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

# 3.1 O Processo de Modelação Estática de Classes

---

- 1 Introdução aos modelos e processos de modelação
- 2 Modelação Entidade-Associação (EA) e programação orientada por objectos (OO)
- 3 Modelação avançada OMT ('Object Modeling Technique') e UML ('Universal Modeling Language')
- 4 A função da decomposição na análise e modelação de Sistemas Complexos

# Os Modelos são obtidos com o Método ou Metodologia Científica

---

Os modelos são instrumentos para comunicar objectivamente, para testar hipóteses ou para prever o desempenho dos sistemas reais.

Exemplos de modelos em engenharia:

- Uma 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 particular deve ser à partida consistente e completo.

# Um Modelo usa uma Linguagem e resulta da aplicação de um processo

---

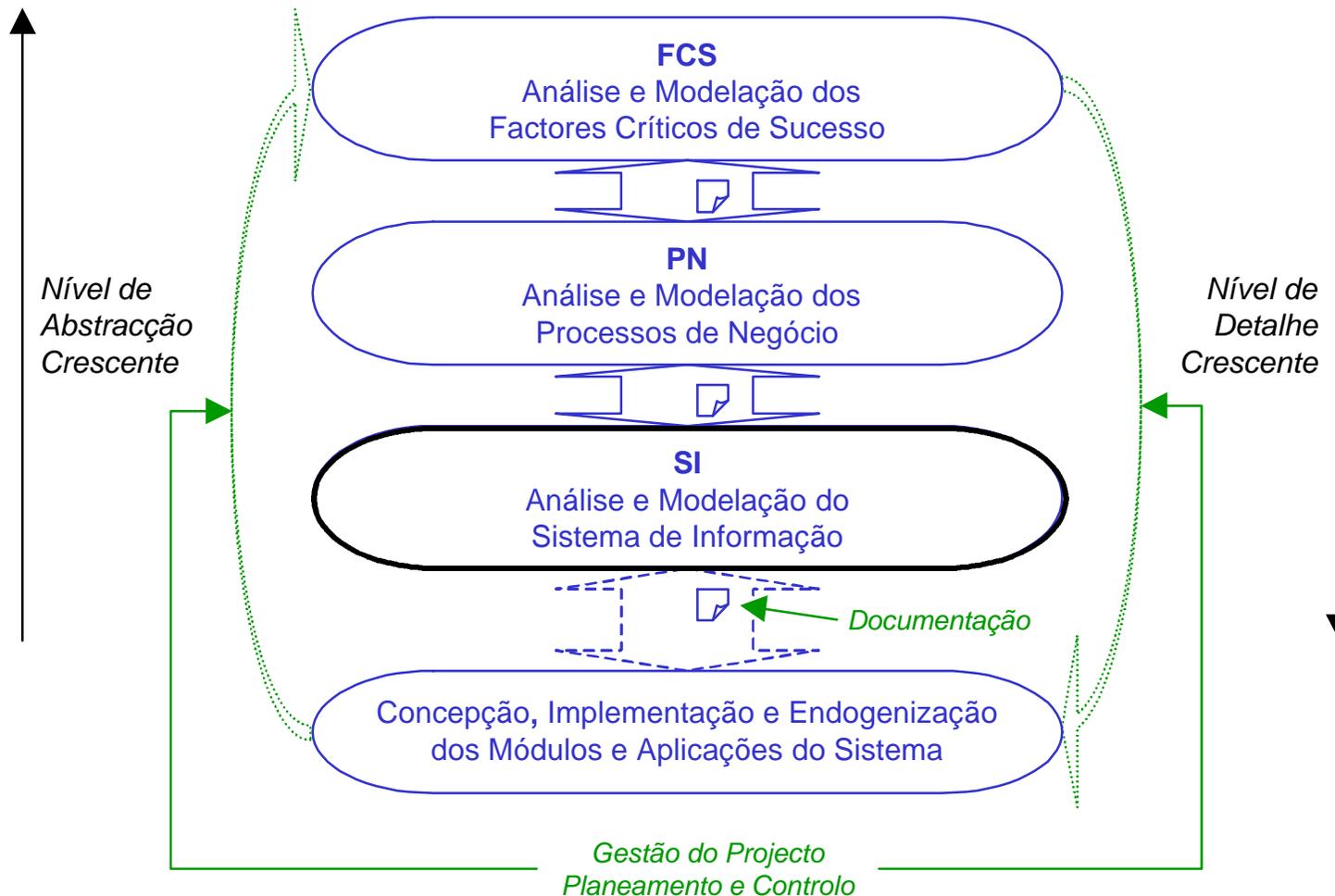
Uma linguagem tem regras de utilização informais e formais.

## **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.

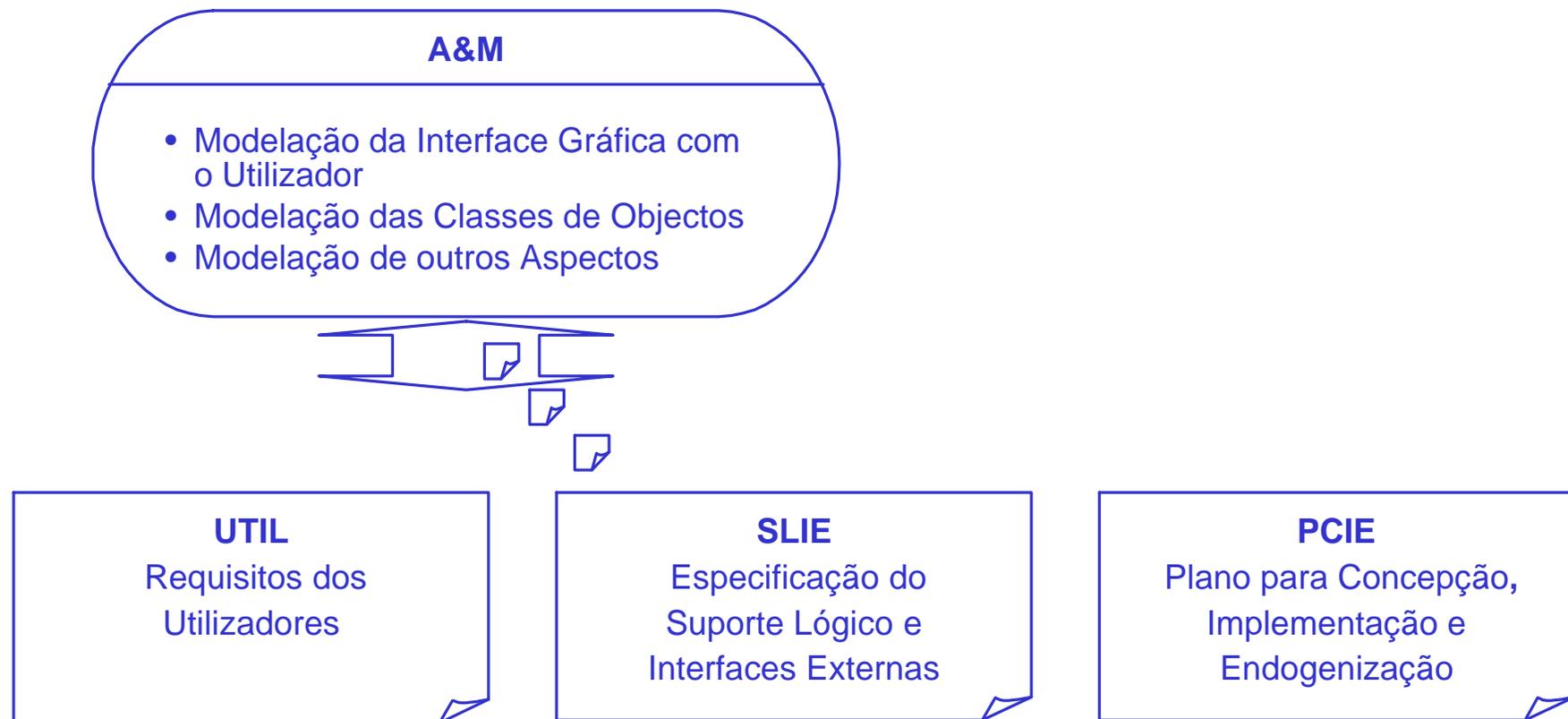
Os modelos obtidos devem ser **implementáveis informaticamente**, mas de forma independente do equipamento ou suporte lógico particular.

# Modelos para o estudo de processos e sistemas em organizações



# Análise e Modelação do Sistema de Informação

---



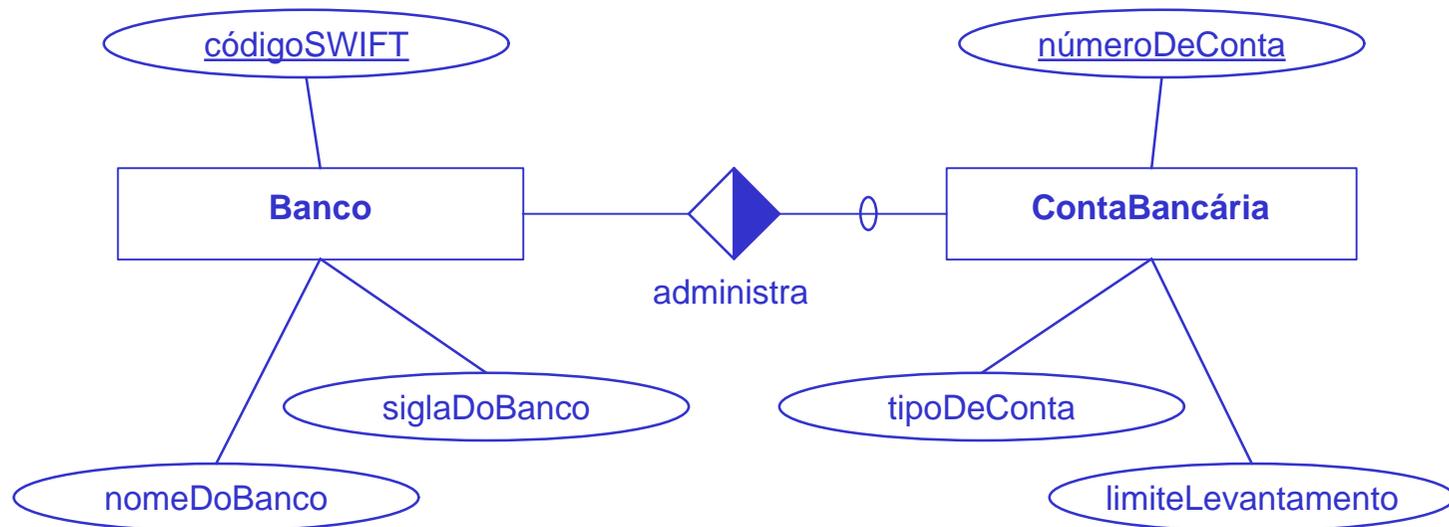
# Modelação Estática das Classes de Objectos

---

- **Estática ? ...**
  - Estática: parte da mecânica que estuda o equilíbrio das forças sobre corpos em repouso?
  - Dinâmica: parte da mecânica que estuda as relações entre as forças e os movimentos por elas produzidos?
  - Cinemática: parte da mecânica que estuda o movimento, independentemente das forças que o provocam?
    - **Nota:** ainda há a óptica, etc.
- **Classes**, Tipos, Conjuntos, Formas, ...
- **Objectos**, Entidades, Agentes, ...

# Modelação EA ou Entidade Associação (ER “Entity-Relationship”).

- Apenas Modelação estática, com apoio numa linguagem de comunicação gráfica, conceptualmente muito simples e clara (mas não concisa).
- Utiliza 4 conceitos primitivos: Entidade, Atributo, Associação, Generalização (EA generalizado ou EAG).



# Modelação OMT

## “Object Modeling Technique”.

---

- Modelação **estática**, **dinâmica** e **funcional** com apoio em 3 tipos de linguagem de comunicação gráfica, conceptualmente clara e concisa (mas não tão simples como a EA).
- Utiliza vários conceitos primitivos, para além dos de EAG.
- Exemplo de um modelo estático de Classes OMT, muito simples:



# Modelação UML

## “Universal Modeling Language”

---

- Utiliza os seguintes modelos (UML v1.1):
  - (1) Diagrama de casos de utilização ("use-cases")
  - (2) Diagrama de classes
  - Diagramas de comportamento:
    - (3) Diagrama de estados ("statecharts")
    - (4) Diagrama de actividades
    - Diagramas de interacção:
      - (5) Diagrama de sequências
      - (6) Diagrama de colaborações
  - Diagramas de Implementação:
    - (7) Diagrama de componentes
    - (8) Diagrama de implantação

# Modelação UML

- Utiliza dezenas de conceitos primitivos, para além dos de OMT
- Exemplo de um Diagrama de Classes UML muito simples:



**Nota:** Repare-se na notação \* de multiplicidade UML.

# Atributos dos Sistemas complexos

---

- A complexidade toma a forma de uma hierarquia; um sistema é composto por sub-sistemas interrelacionados.
- A escolha dos componentes primitivos (atômicos ?) é relativamente arbitrária e depende do observador do sistema.
- A ligação ou a comunicação interna a uma componente de um sistema é maior ou mais frequente do que a externa, entre componentes.
- Os sistemas hierárquicos são compostos por poucos tipos de sub-sistemas em muitas combinações ou arranjos.
- Um sistema complexo que funciona é sempre o resultado de um sistema mais simples que funcionava.

# Que Decomposição?

---

- Decomposição algorítmica ... vista de processo, ordenação de acções e eventos (menos estáveis), ...
- Decomposição orientada por objectos ... vista de agentes (mais estáveis) que executam acções, ...
  - Programação OO.
  - Modelação EA de Bases de Dados.
  - Filosofia Cartesiana (Descartes).

Qual das duas decomposições deve ser efectuada inicialmente?

# Como modelar e construir Sistemas complexos ?

---

- **Uma Linguagem ... da modelação à programação.**
- **Um Processo:** as regras para a obtenção ordenada das várias partes do modelo e para a construção do sistema (do tipo LUCID ou prototipificação evolutiva, mas aplicadas a cada um dos modelos particulares em causa).
- **Ferramentas:** utensílios para apoiar a produção dos modelos, de forma a facilitar a aplicação do processo, a automatizar a verificação de regras de consistência e a eliminar erros sintácticos.

# 3. Modelação Conceptual de Classes

---

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



## 3.2 Classes de Objectos, Atributos e Métodos

---

### ***CONTEÚDOS***

1. Objectos e Atributos
2. Classes de Objectos
3. Operações e Métodos de Classes
4. Domínios e Operações Complexas
5. Atributos Derivados
6. Atributos de Classe e Operações de Classe
7. Atributos candidatos a identificadores
8. Significados implementacional e denotacional

# Objectos e Atributos

---

- Dado um sistema ou problema em estudo, como identificar os objectos e respectivos atributos a incluir nos modelos de análise ou concepção?
  - “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. Este percurso foi formado pela sequência ligada dos seguintes 3 trajectos: Porto-Leixões/Porto-Serralves, Porto-Serralves/Braga-Centro e Braga-Centro/Porto-Leixões. O sistema de planeamento operacional GistRv sugeriu que fossem utilizados os trajectos indicados para uma viatura do depósito Porto-Leixões”.

# 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), OTN3456 (serviço), 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

## V3498DH:Viatura

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

## PLeia: PersonagemDeFilme

nome = Princesa Leia  
primeiroFilme=Star Wars

## FSCarneiro: Aeroporto

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

## Heathrow: Aeroporto

código = LHR  
nome = Heathrow Airport  
cidade = Londres

# 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. As classes são os nomes comuns das descrições.
- 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. O modelo de análise deve assim utilizar apenas atributos externos.

# Exemplos de Classes de Objectos

<b>Livro</b>
códigoISBN nExemplar data título

<b>Aeroporto</b>
código nome local

<b>CódigoPostal</b>
códigoPostal localidade

Classe *Pessoa*  
para  
*Análise:*

<b>Pessoa</b>
nome endereço códigoPostal país

Classe *Pessoa*  
para  
*Concepção:*

<b>Pessoa</b>
nomePróprio apelido outrosApelidos endereçoL1 endereçoL2 aldeiaVilaOuCidade códigoPostal país

# Operações e Métodos de Objectos

---

- As operações são funções ou procedimentos definidos no âmbito de um objecto com determinados argumentos e resultado, implementados como métodos da respectiva classe, por exemplo. Algumas operações são polimórficas, isto é o tipo dos seus argumentos pode variar.

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.



**Nota:** Não confundir com *Métodos sobre Classe*.

# Domínios e Operações Complexas

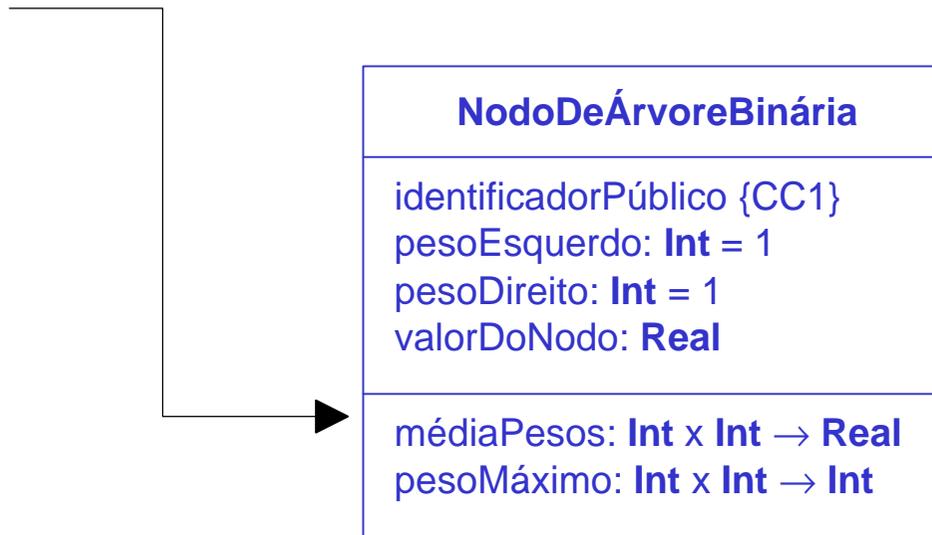
---

- 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 (simples, sobretudo em caso de implementação com SGBDr).
- Um domínio é um conjunto de valores; os valores são entidades abstractas que não têm identidade, ao contrário dos objectos, que correspondem a entidades com identidade na realidade do problema. A noção formal de domínio corresponde à noção tradicional de um conjunto a que se adiciona normalmente um elemento com valor indefinido.
- É possível ou desejável modelar Métodos ou Operações Complexas? (para SGBDr, a resposta é normalmente negativa).

# 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*.

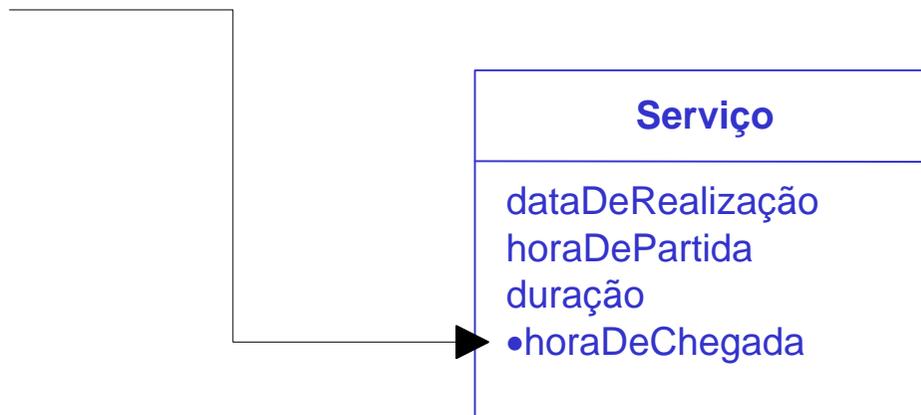


**Questão:** Como implementar num SGBDr?

# 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*.



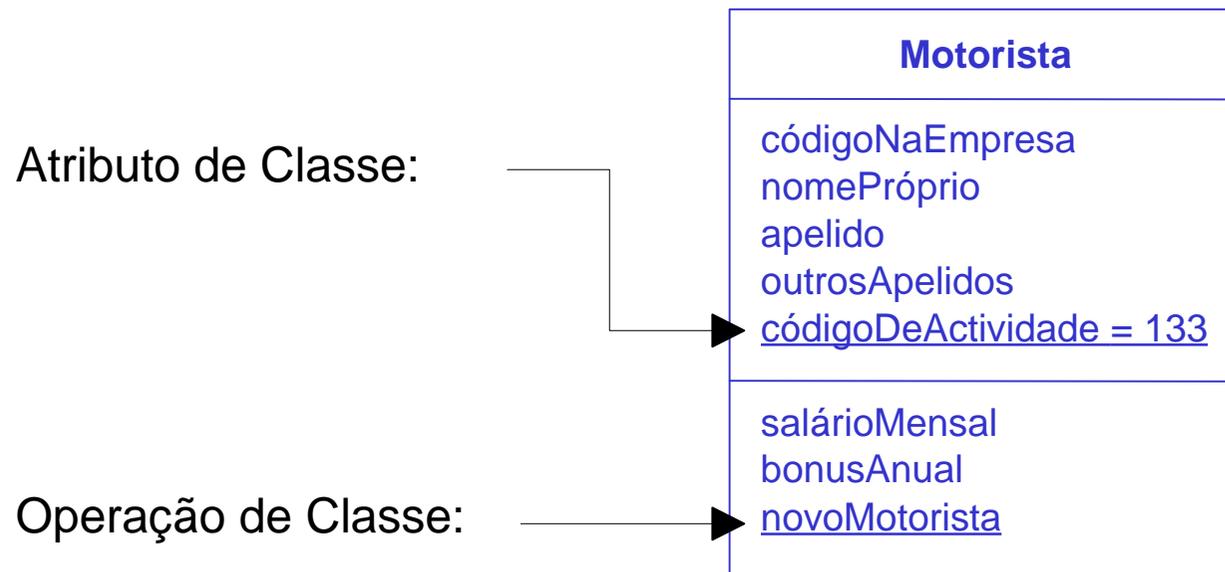
# Atributos sobre Classes e Operações sobre Classes

---

- 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.
- Operações de classe: úteis para criar um novo objecto; uma operação deste tipo tem assim como domínio a própria classe e não objectos ou seus atributos.

# Exemplos de Atributos de Classe e Operações 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. Na realidade 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 último 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. Obviamente nenhum identificador composto pode incluir um atributo identificador simples.

# 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.



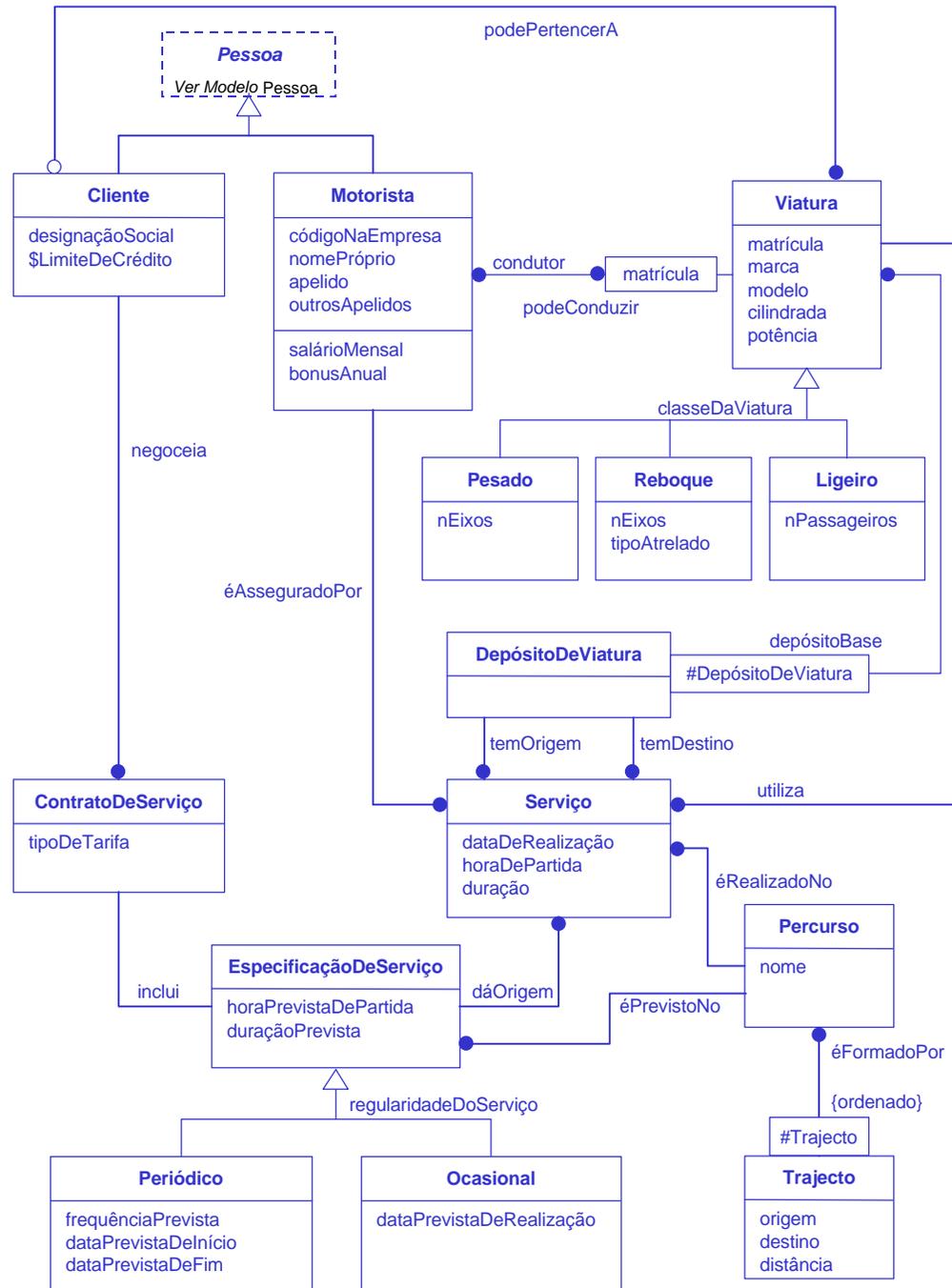
# Significados implementacional e denotacional

---

- **Significado Implementacional “*Como implemento?*”:** Por exemplo, envolvendo a tradução para o modelo relacional, baseado na álgebra relacional, seguindo-se a utilização de um dado SGBDr, SQL e linguagem de programação apropriada. Todos estes passos efectuados com base em normas existentes.
- **Significado Denotacional “*Que significa?*”:** Por exemplo, envolvendo modelos formais recorrendo a uma teoria de conjuntos ou domínios, e à lógica formal para fornecer um significado de referência, abstracto, independente de uma particular sequência de implementação.

# Exemplo de Modelo

Modelo estático de classes do subsistema de gestão de transportes da empresa OnTime:



# Conclusões

---

- A identificação das classes de objectos, e dos seus atributos e operações, é a primeira fase do processo de modelação estática de classes.
- A identificação das operações não é útil em termos de modelo relacional puro. Mas é importante quer para o modelo de interacção com o utilizador, quer para a implementação global das funcionalidades requeridas.
- A notação gráfica utilizada, baseada na norma UML do diagrama de classes, é simples, clara e concisa, tendo um significado denotacional e implementacional preciso.

# 3. Modelação Conceptual de Classes

---

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



## 3.3 Ligações, Associações e Agregações Simples

---

### ***CONTEÚDOS***

1. Ligações entre Objectos
2. Associações entre Classes
3. Atributos ou Classes em Ligações
4. Agregação entre Classes

# Ligações entre Objectos

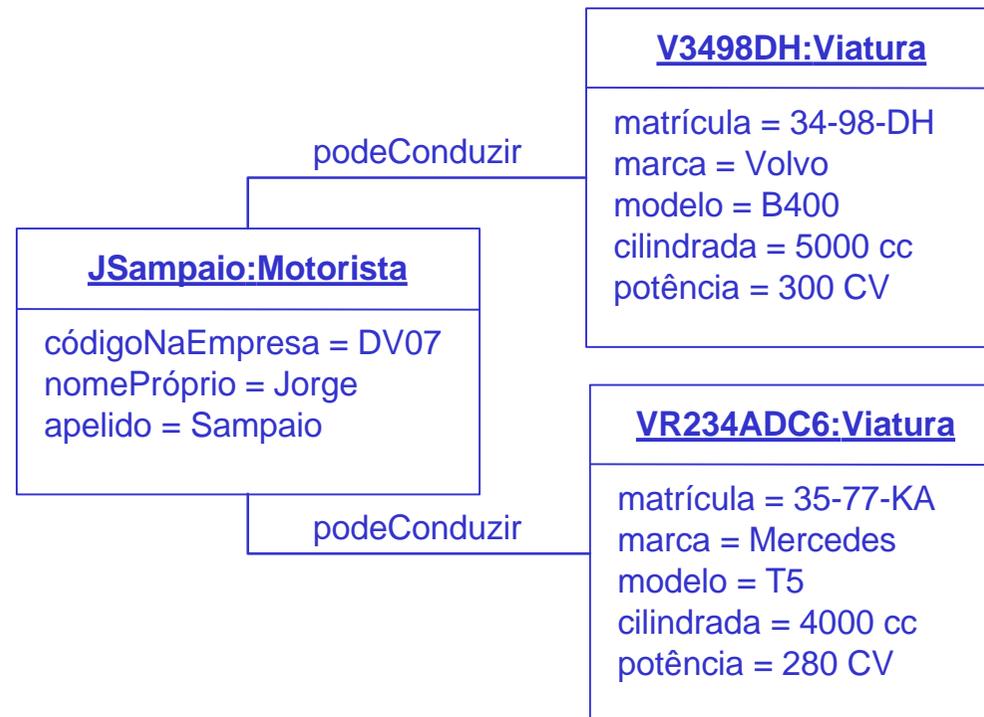
---

- 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: caso de uma autarquia, com os seus clientes, funcionários, departamentos, divisões e serviços. Os clientes são por exemplo os residentes que pagam a contribuição autárquica. Quais as relações existentes?

# 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.

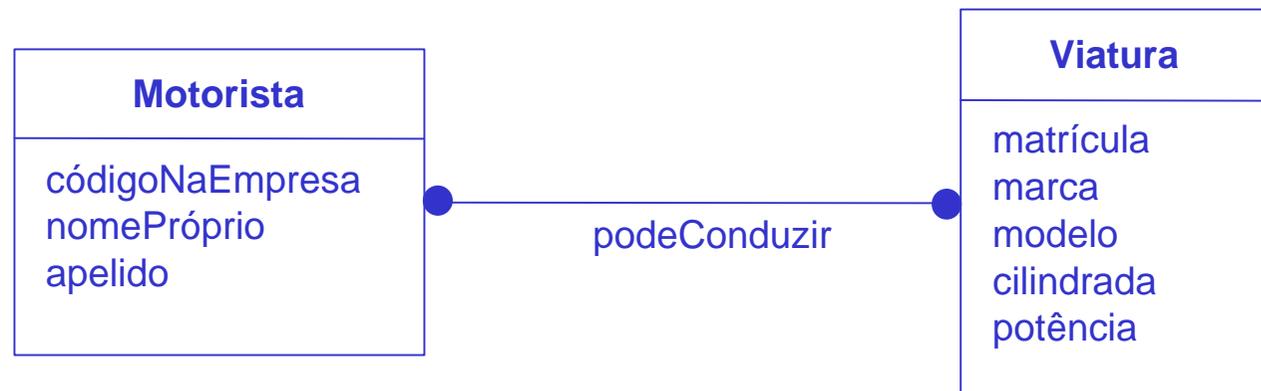
- Exemplo de ligações *podeConduzir* entre objectos das classes Motorista e Viatura:



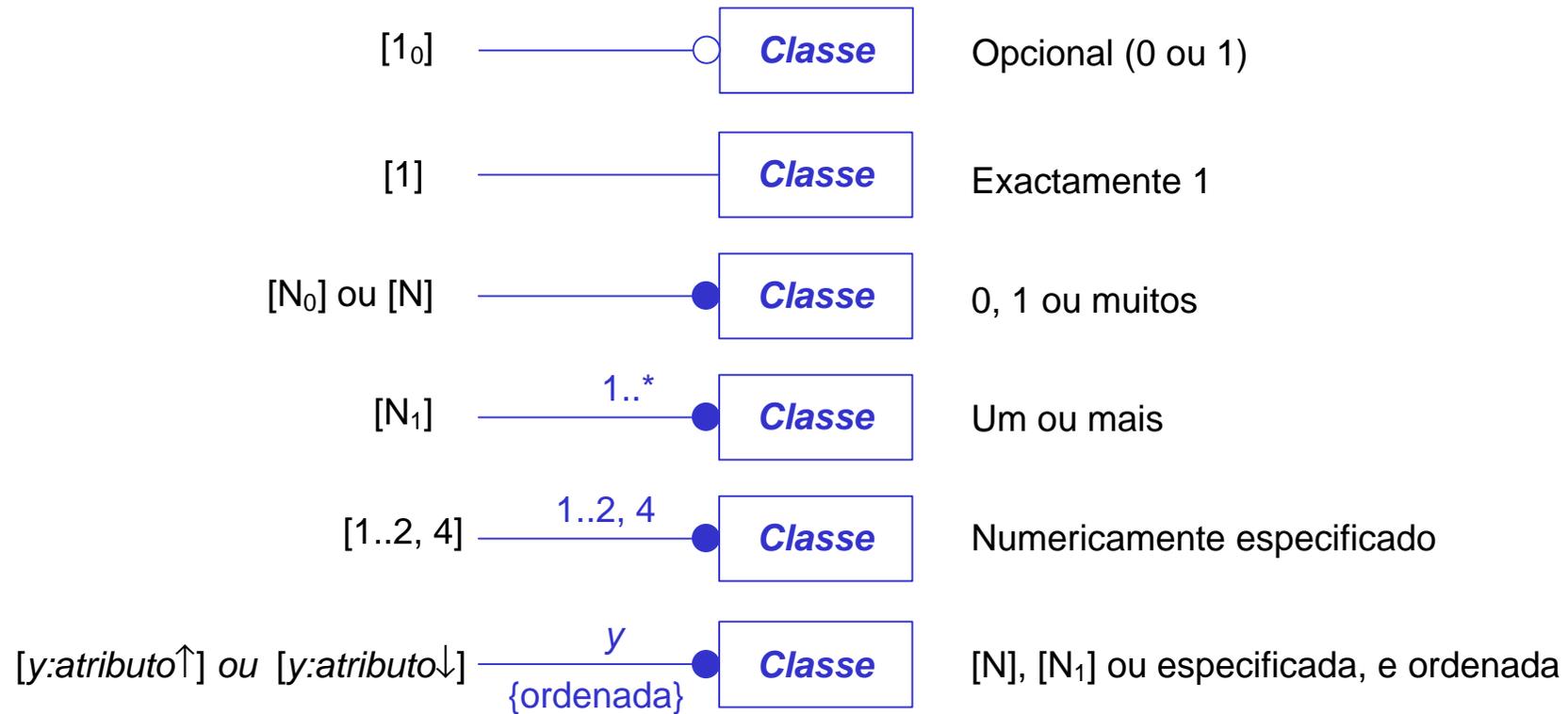
# Associações entre Classes

- 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:

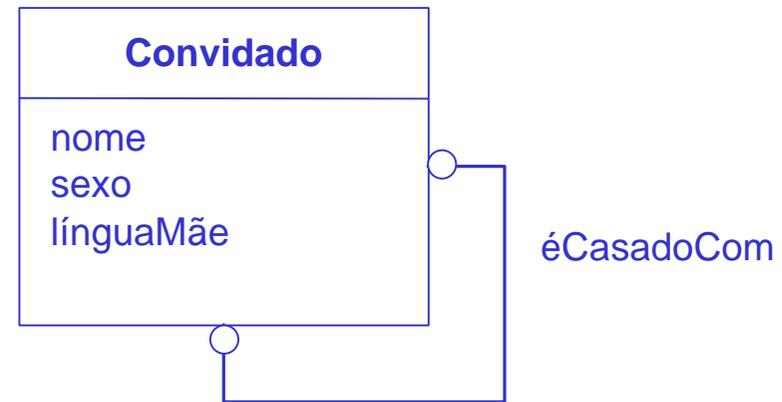


# Multiplicidade de Associações



# 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:



- O modelo com a associação unária é normalmente preferível (porquê?).

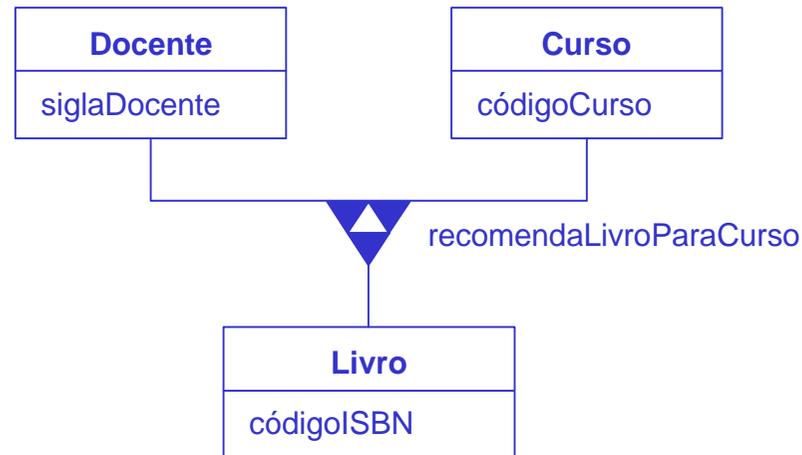


☹ *Modelo Incorrecto*



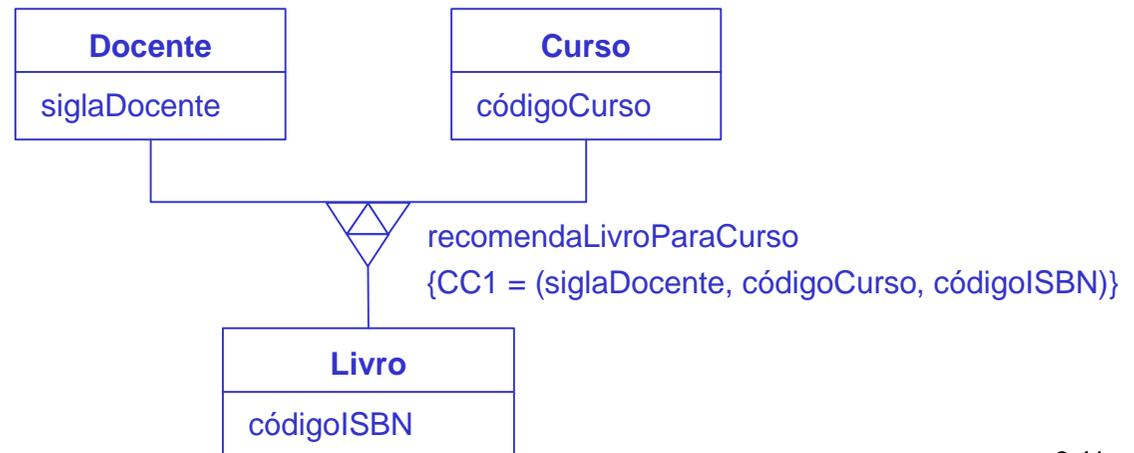
# Exemplo de Associação Ternária

- Multiplicidade representada graficamente:



**Questão:** qual é a situação que exige a associação aqui representada?

- Multiplicidade representada pelo identificador CC1 da associação:



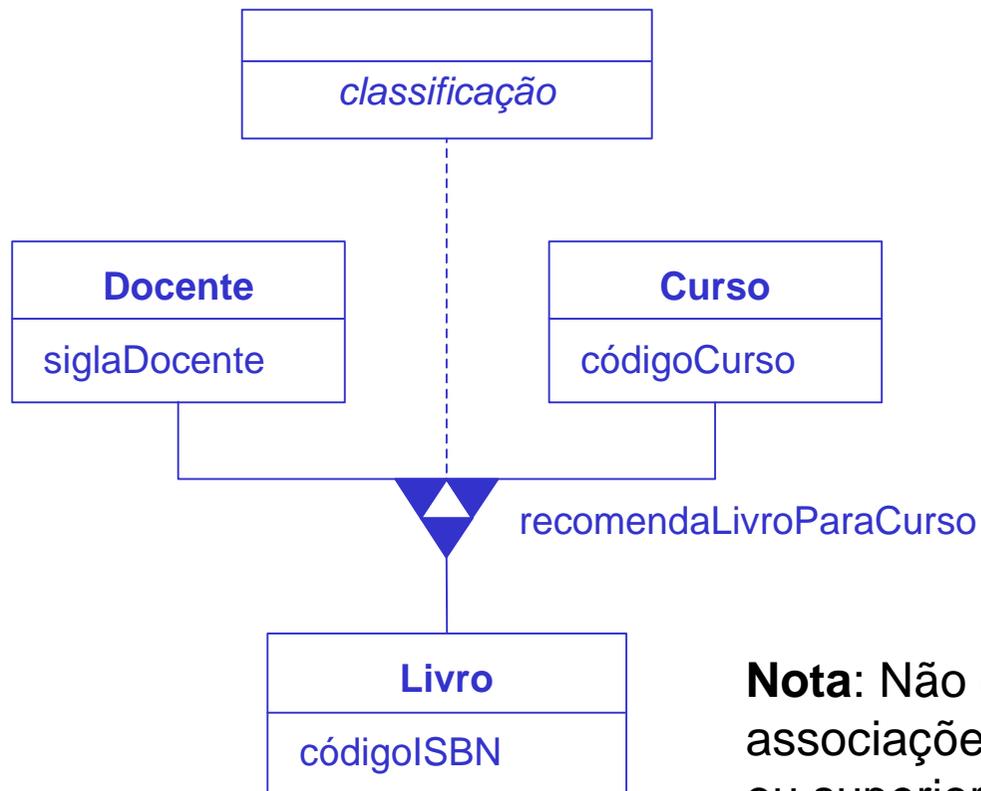
# Atributos ou Classes em Ligações

---

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

# Exemplo de Atributo em Ligação

---



**Nota:** Não é desejável a utilização de associações de multiplicidade ternária ou superior, devido à complexidade de leitura dos modelos resultantes.

# Associações Qualificadas

- Associação qualificada:



**Nota:** As linguagens OMT e UML utilizam uma notação diferente!

- Associação não qualificada:



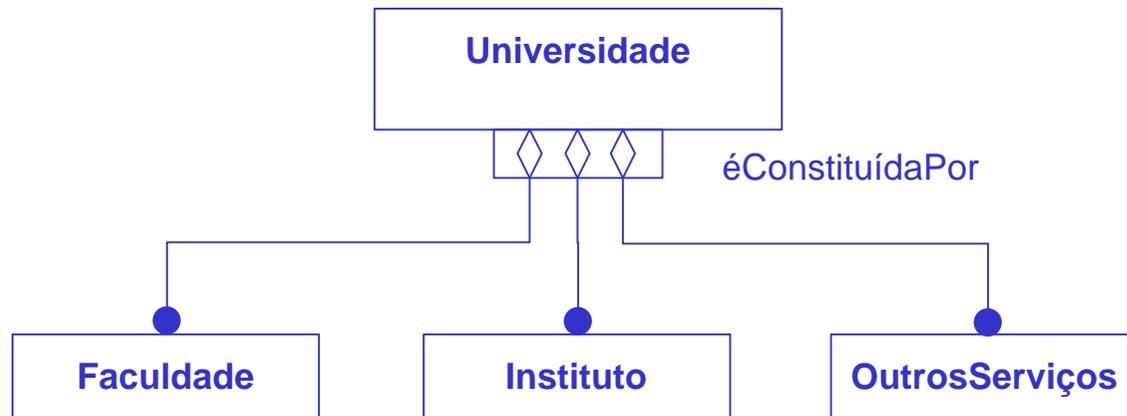
# Agregação entre Classes

---

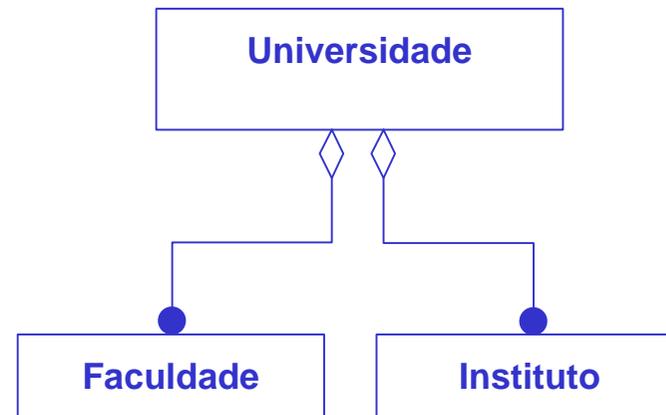
- A **agregação** é um tipo especial de associação, utilizada por exemplo para indicar a composição física de um tipo de produto, ou a estrutura de organização de uma entidade, conforme os exemplos seguintes.
- O conjunto de ligações entre os objectos que a agregação descreve, visto como uma relação matemática, caracteriza-se pelas propriedades de transitividade e anti-simetria.

# Exemplo de agregação entre classes

- Agregação fechada, com indicação de todas as classes constituintes:

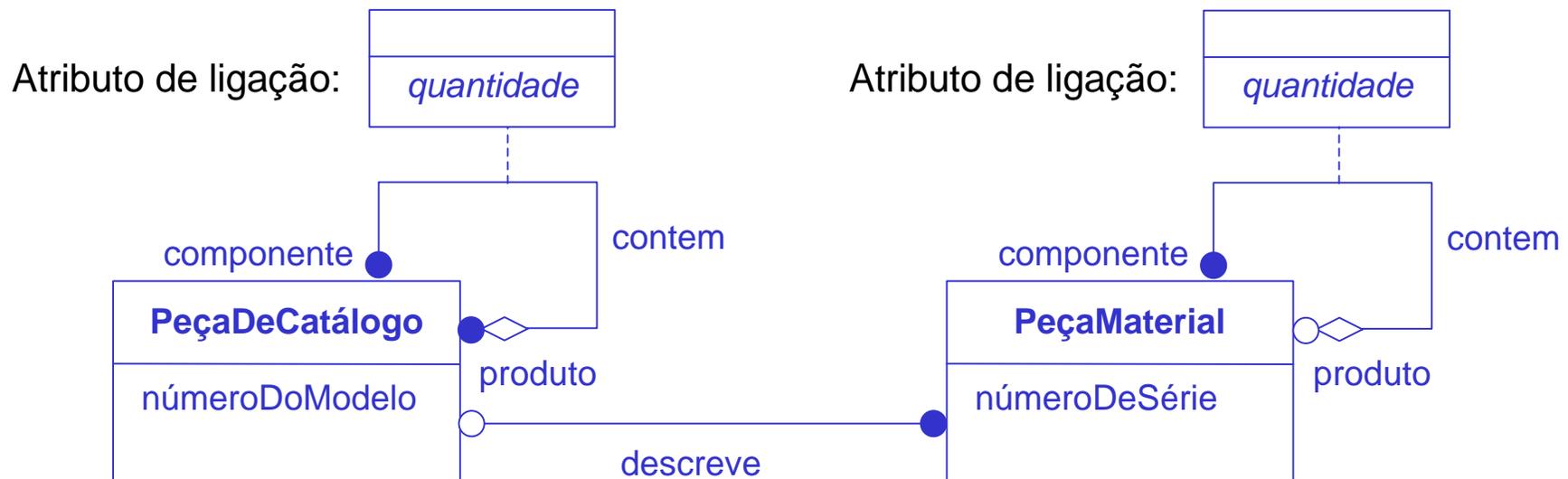


- Agregação aberta, indicando algumas classes constituintes:



# Agregação de catálogo e de produto

- Exemplo de agregação de catálogo e de produto: uma peça em catálogo pode pertencer a muitos produtos.



Exemplo adaptado de [Blaha & Premerlani 1998; 53].



# Conclusões

---

- Depois de identificar os objectos, classes e atributos pode-se observar como estes se relacionam, através de ligações, associações e agregações.
- As classes e associações a utilizar dependem do detalhe necessário ao modelo em construção. O mesmo conceito numa situação ou problema real pode originar atributos, objectos, classes, ligações, associações ou agregações diferentes em processos de Análise e Modelação orientados para níveis de detalhe ou objectivos diferentes.
- A análise global dos diagramas deve permitir validar a completude e verificar a consistência do modelo construído.

# 3. Modelação Conceptual de Classes

---

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

## 3.4 Generalização e Herança Simples

---

### ***CONTEÚDOS***

1. Generalização ou Particularização (exclusiva)
2. Classes Concretas e Abstractas
3. Particularização Inclusiva

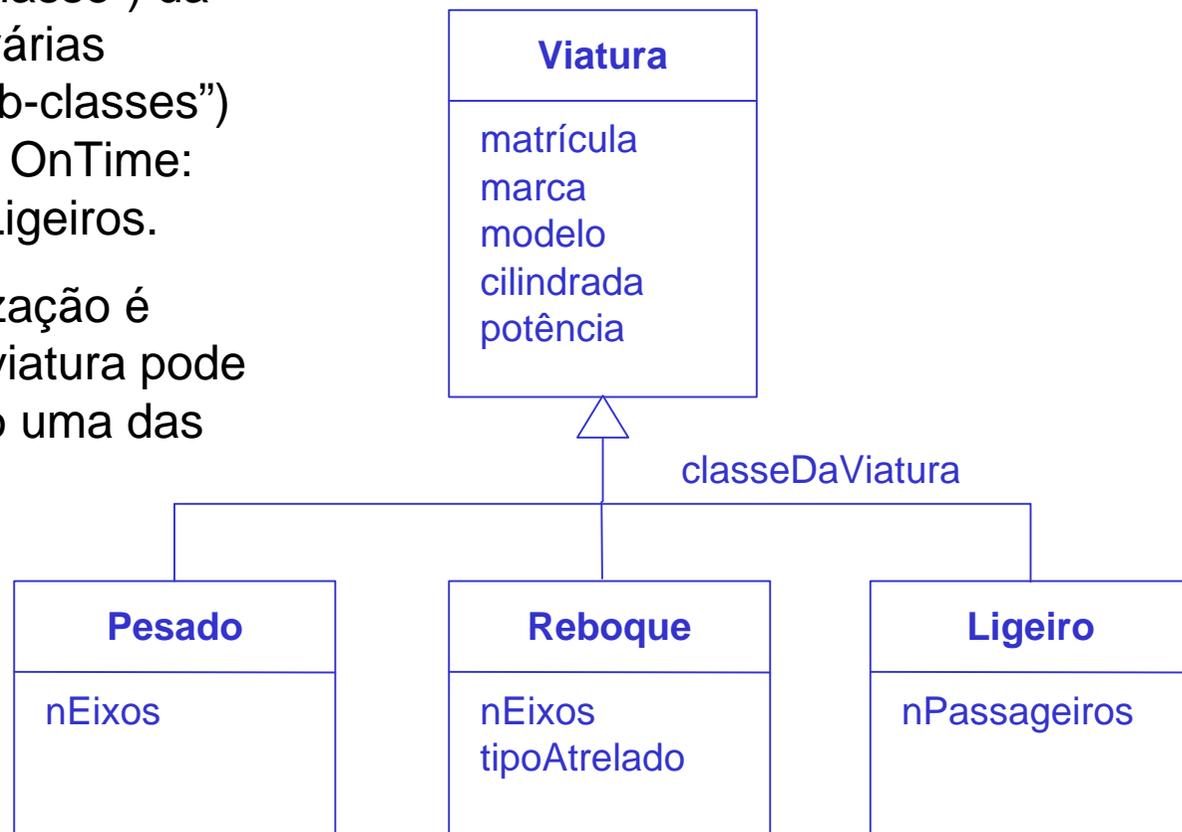
# 1. Generalização ou Particularização Exclusiva

---

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

# Exemplos de Generalização Exclusiva

- Exemplo de Generalização na classe Viatura (“super-classe”) da informação comum às várias classes de viaturas (“sub-classes”) utilizadas pela empresa OnTime: Pesados, Reboques e Ligeiros.
- Diz-se que a particularização é exclusiva quando uma viatura pode ser apenas de uma e só uma das classes particulares.



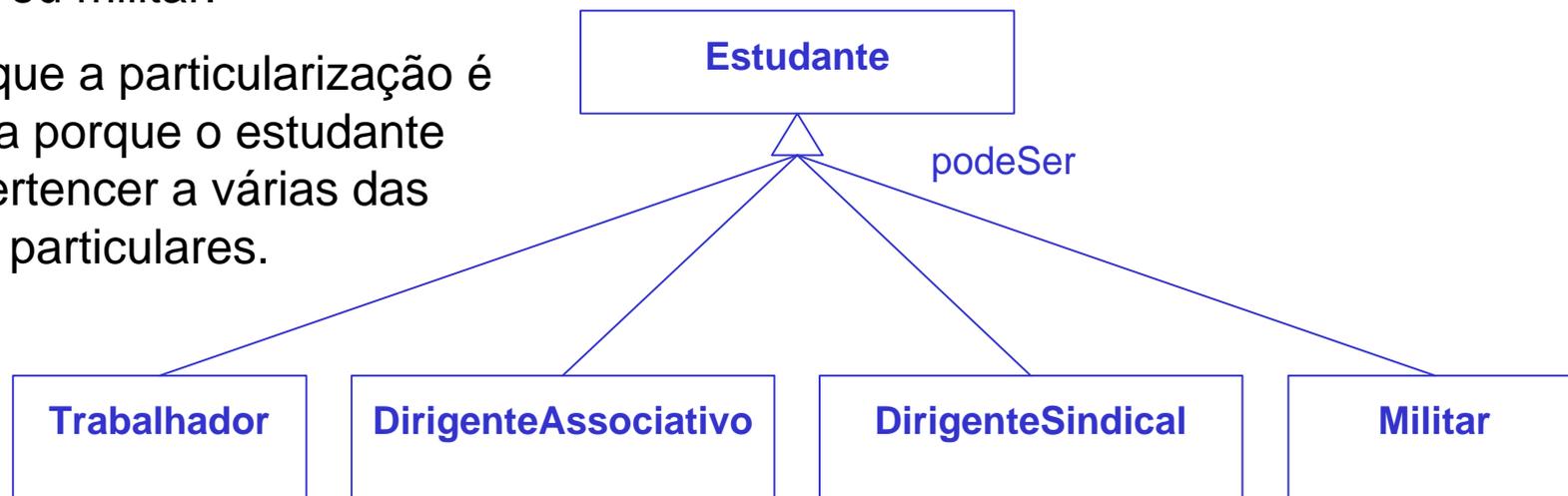
## 2. Classes Concretas e Abstractas

---

- Uma classe (super-classe) diz-se abstracta quando não tem instâncias. 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, nas suas formas de implementação.
- Por exemplo, vamos supor que identifico uma super-classe Pessoa e respectivas sub-classes Pessoa Singular e Pessoa Colectiva. Considerando que decido apenas implementar estas duas últimas, posso afirmar que a classe Pessoa é abstracta, visto não existirem instâncias ou objectos da classe Pessoa.

# Exemplo de Particularização Inclusiva

- Exemplo de Generalização Inclusiva na classe Estudante das classes trabalhador, dirigente associativo, dirigente sindical ou militar.
- Diz-se que a particularização é inclusiva porque o estudante pode pertencer a várias das classes particulares.





# Conclusões

---

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

# 3. Modelação Conceptual de Classes

---

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

## 3.5 Mapeamento entre o Modelo de Classes e o Modelo Relacional

---

### ***CONTEÚDOS***

1. Exemplo de Conversão de Classes
2. Exemplo de Conversão de Associações e Agregações
3. Exemplo de Conversão de Generalizações

# Exemplo de conversão de classes

---



Motorista			
<i>#motorista</i>	<i>códigoNaEmpresa</i>	<i>nomePróprio</i>	<i>apelido</i>
1	DV07	Jorge	Sampaio
3	J007	Maria	Branca
4	P235	José	Lami
5	Y666	Ana	Chaves

# Conversão de Associações e Agregações

---

- A implementação de uma associação entre duas ou mais classes num SGBDr pode ou não necessitar de uma tabela independente das tabelas correspondentes a essas classes. De seguida apresentam-se regras de conversão para os principais tipos de associações utilizadas, havendo normalmente várias opções de implementação, umas mais recomendadas do que outras.
- As regras de conversão para agregações são idênticas às das associações com a mesma multiplicidade, pelo que não se apresentam exemplos.

# Associações um-para-um



ContratoDeServiço		
#contratoDeServiço	tipoDeTarifa	#especificaçãoDeServiçoIncluída
	1 Alfa	1
	2 Beta	5
	4 Beta	4
	8 Gama	3
...		

EspecificaçãoDeServiço		
#especificaçãoDeServiço	horaPrevistaDePartida	duraçãoPrevista
1	10:00	2:44
3	15:00	3:23
4	10:00	5:44
5	23:23	3:30
...		

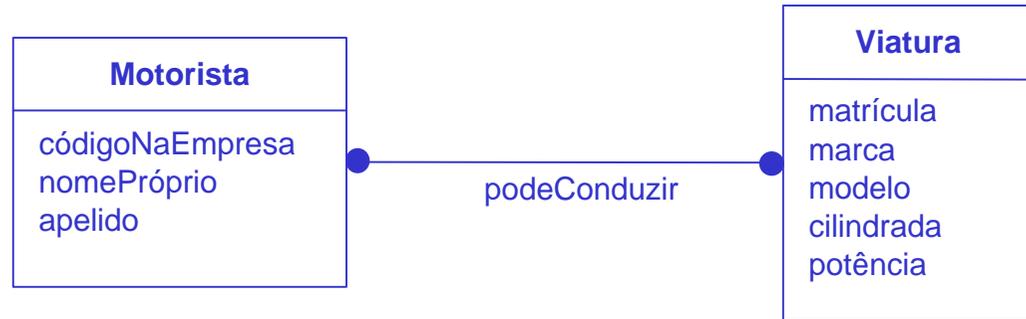
# Associações um-para-muitos



Banco			
#banco	nomeDoBanco	siglaDoBanco	códigoSWIFT
22	Banco Português do Atlântico	Atlântico	334455
33	Banco Comercial Português	BCP	778907
34	Caixa Geral de Depósitos	CGD	333333

ContaBancária				
#contaBancária	númeroDeConta	tipoDeConta	limiteLevantamento	#bancoQueAdministra
910988876	382-98989-210	O	200	34
910988878	12120009-32	O	100	34
910988880	AA23347800	O	50	22
910988881	334-00899-230	H		34

# Associações muitos-para-muitos

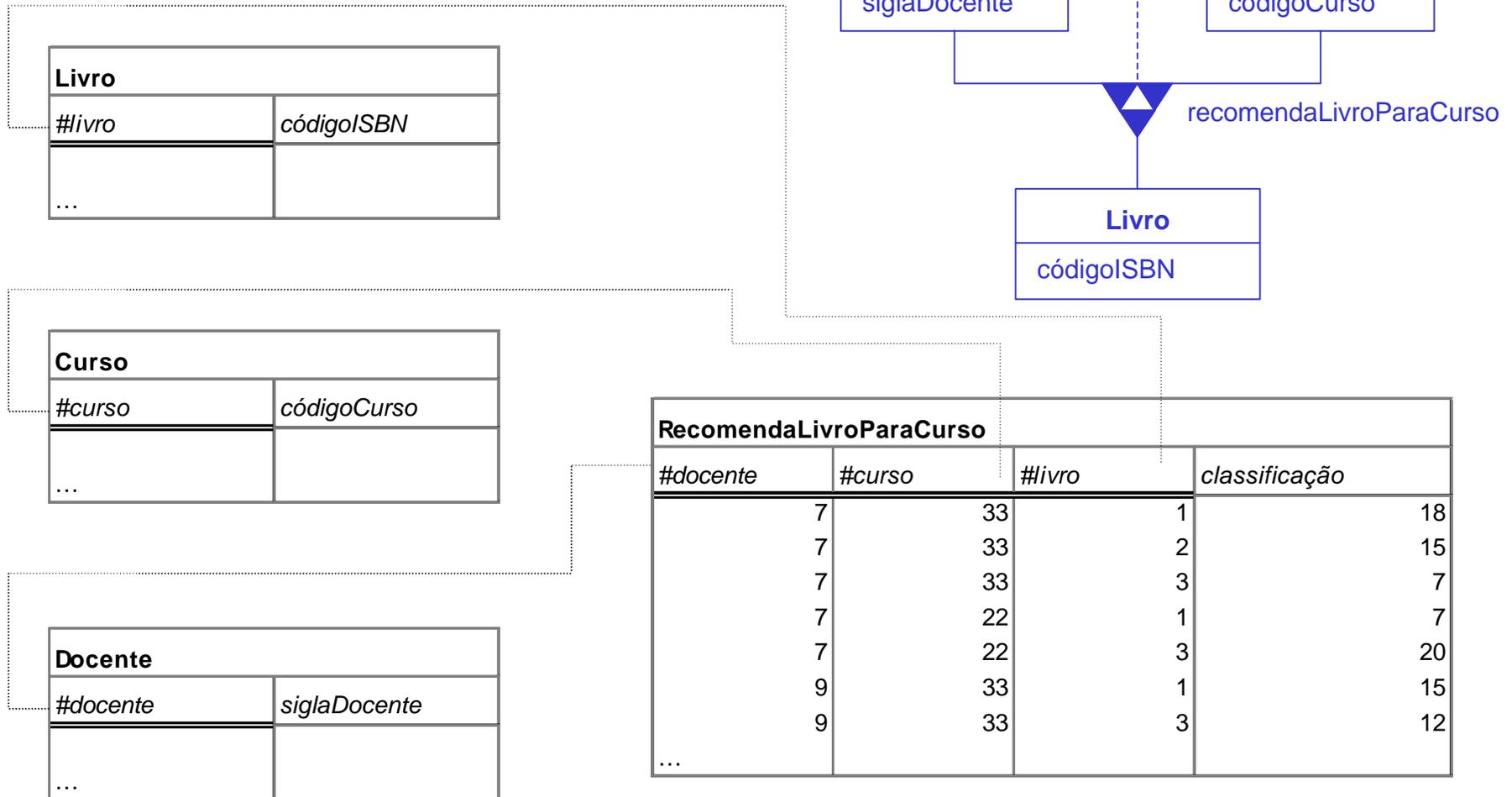


Motorista			
#motorista	<i>códigoNaEmpresa</i>	<i>nomePróprio</i>	<i>apelido</i>
...			

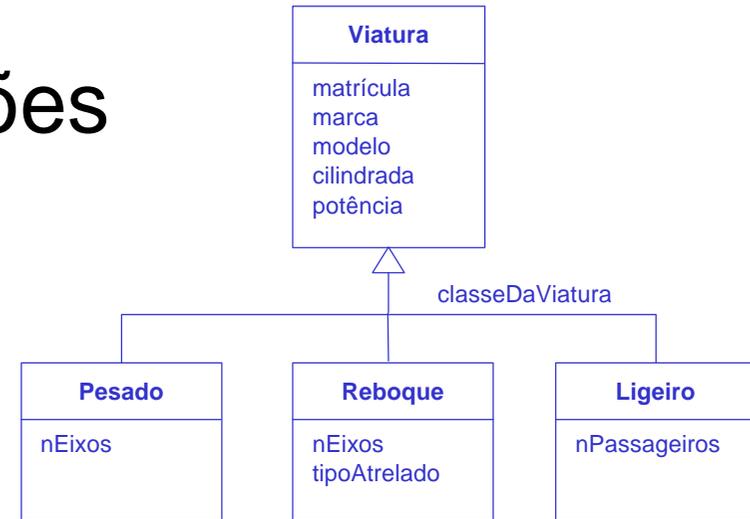
Motorista_podeConduzir_Viatura	
#motorista	#viatura
...	

Viatura					
#viatura	<i>matrícula</i>	<i>marca</i>	<i>modelo</i>	<i>cilindrada</i>	<i>potência</i>
...					

# Associações ternárias



# Conversão de generalizações exclusivas



Viatura						
#viatura	matrícula	marca	modelo	cilindrada	potência	classeDaViatura
12	12-12-TR	Volvo	Z707	2002	250	Ligeiro
13	13-44-TR	Volvo	Z704	2002	250	Ligeiro
77	22-98-ZR	Volvo	P4	8000	400	Pesado
78	22-99-ZR	Volvo	X700	2002	200	Ligeiro
79	22-12-ZZ	Volvo	P4	8000	500	Pesado
...						

Pesado	
#viaturaPesado	neixos
77	8
79	8
116	12
...	

Reboque		
#viaturaReboque	neixos	tipoAtrelado
101	8	Articulado
102	8	Articulado
117	12	Longo
...		

Ligeiro	
#viaturaLigeiro	nPassageiros
12	5
13	4
78	2
...	

# Conversão de generalizações inclusivas



Estudante	
#estudante	podeSer
	1 Trabalhador
	1 Militar
	3 Dirigente Associativo
	3 Militar
	5 Militar
...	

Trabalhador	
#estudanteTrabalhador	
	1
	7
	8
...	

DirigenteAssociativo	
#estudanteDirigenteAssociativo	
	3
	15
	19
...	

DirigenteSindical	
#estudanteDirigenteSindical	
	10
	11
	45
...	

Militar	
#estudanteMilitar	
	1
	3
	5
...	

# Conclusões

---

**Fase prévia:** Identificar e estudar todas as fontes relevantes de conhecimento sobre o sistema, com o objectivo de tipificar os utilizadores e descrever exhaustivamente os casos de utilização. Elaborar descrições do contexto e do problema, bem como maquetas de baixa resolução das principais interfaces.

**Fase de Modelação de Classes:** Classes, Associações, etc.

**Fase seguinte:** Traduzir o modelo de classes para o modelo do sistema informático mais apropriado a utilizar, por exemplo, obtendo o esquema relacional no caso de se perspectivar a implementação com apoio num SGBDr.

# Principais referências

---

**[Blaha & Premerlani 1998]:** *Object-Oriented Modeling and Design for Database Applications*, Prentice-Hall (Cap. 1, 2, 3, 7, 8, 9, 13, 14).

**[Rumbaugh *et al* 1991]** *Object-Oriented Modeling and Design*, Prentice-Hall (Cap. 1, 2, 3, 4, 5, 6).

**[Schneider & Winters 1998]** *Applying Uses Cases: a Practical Guide*, Addison-Wesley Longman (Cap. 1, 2, 3).

**[UML Summary 1997]** *UML Summary*, version 1.1, Rational Software Corporation ([www.rational.com/uml](http://www.rational.com/uml)).