

Fundamentos de Base de Dados

Diagrama de Classes

Pedro Nogueira Ramos

(Pedro.Ramos@iscte.pt)

ISTA / DCTI

UML

(Booch, Jacobson, Rumbaugh, 1999)

UML - Unified Modeling Language, i.e., é uma linguagem formal (essencialmente diagramática) de modelação.

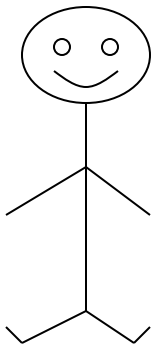
É uma linguagem **genérica**, i.e., permite a modelação de qualquer sistema (componentes **estática e dinâmica**) em diferentes níveis de análise (desde modelos genéricos a modelos para implementação). **Não é** uma linguagem específica para sistemas de informação.

Baseia-se no **paradigma dos objectos**, nomeadamente suporta a definição de objectos e classes.

Objectos

Objecto: qualquer coisa relevante do domínio que pretendemos modelar e que têm:

- . **Identidade** (forma de o identificar)
- . **Estado** (conjunto de atributos)
- . **Comportamento** (operações que sobre ele podem ser efectuadas)

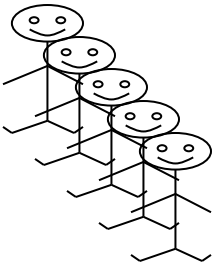


- É **distinto** de outros clientes da empresa
- Atributos: **nome, morada, n° contribuinte, ...**
- Operações: **emitir facturas, alterar morada, ...**

Cliente **‘João Silva’**

Classes

Classe: conjunto de objectos que partilham o mesmo Mecanismo de Identificação, Estado, Comportamento, Relações e Semântica.

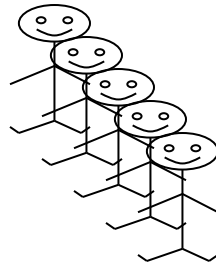


Classe dos
Clientes

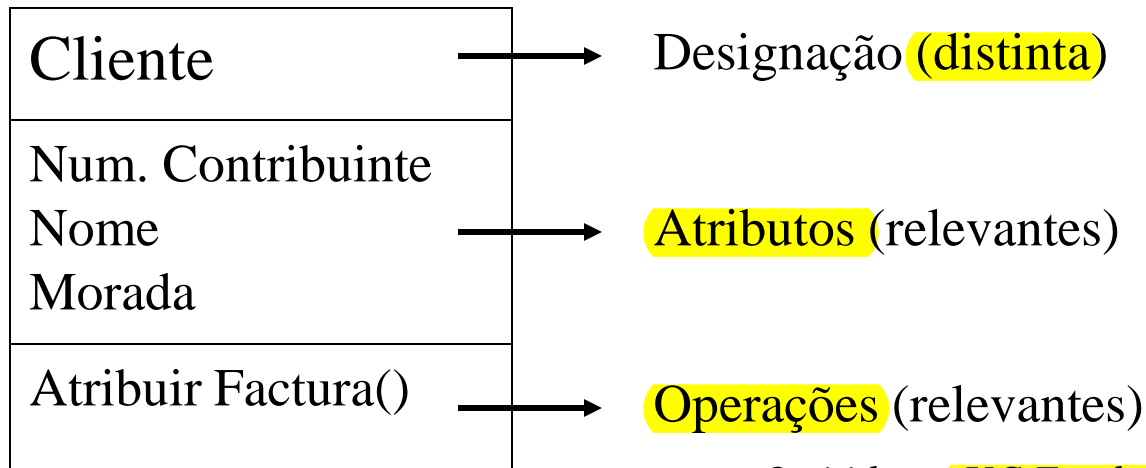
- Todos **distintos** uns dos outros
- **Partilham** atributos e operações
- Relacionam-se com **as mesmas classes** (e.g., produtos que adquirem)
- Representam **a mesma realidade** (semântica)

Os objectos não têm necessariamente que corresponder a entidades humanas ou, mais genericamente, a entidades com representação física (e.g., uma factura). Um **conceito abstracto**, por exemplo um departamento, pode ser um objecto (caso seja relevante para o domínio em causa).

Classe: Representação Gráfica



Classe dos
Clientes



*Omitido na UC Fundamentos de Bases de
Dados*

Relações

Em qualquer sistema existem objectos que **se relacionam** entre si. Por exemplo, se considerarmos a classe das facturas da empresa, o cliente ‘João Silva’ relaciona-se com as facturas a ele emitidas.

A relação entre objectos é representada através das relações entre as classes, que podem ser de dois tipos:

1) **Associações** —————> Casos especiais:

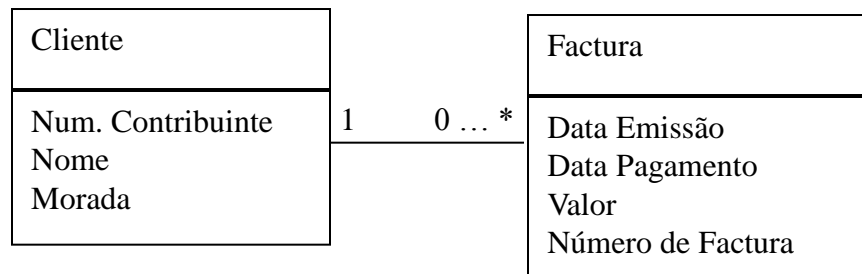
Composição

Agregação

2) **Generalizações**

Associações

Uma associação é uma relação que permite especificar que objectos de uma dada classe se relacionam com objectos de outra classe.



Um cliente pode estar associado a n facturas, ou a nenhuma ($[0, *]$), e

Uma factura está obrigatoriamente associada a um e apenas a um cliente ($[1, 1]$).

Nota: 1 é equivalente a 1 ... 1

Multiplicidade das Associações

0 ... 1, zero ou um

0 ... *, de zero a n

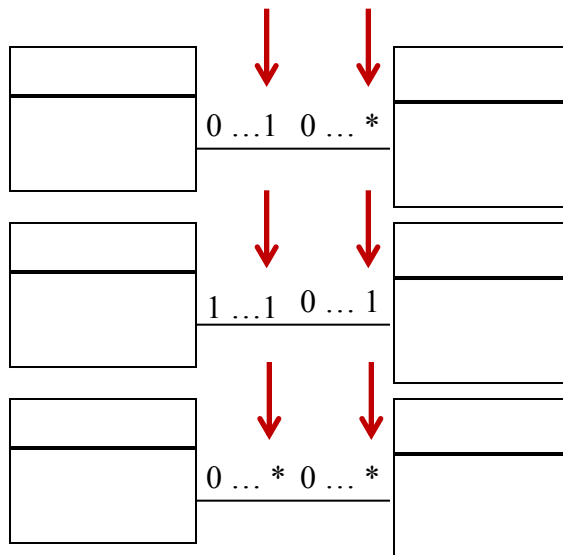
0 ... 3, de zero a 3

1 ... 1, um e apenas um

1 ... *, de um a n

1 ... 3, de um a 3

... infinitas combinações que é vulgar agruparem-se em:

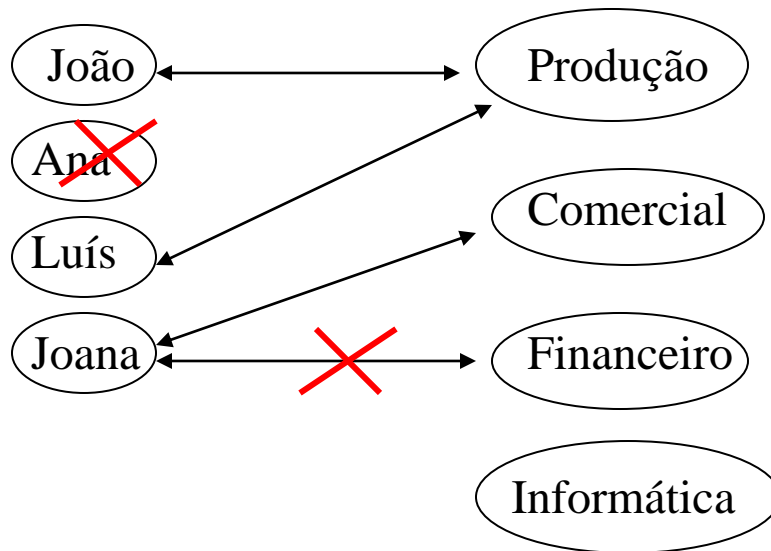
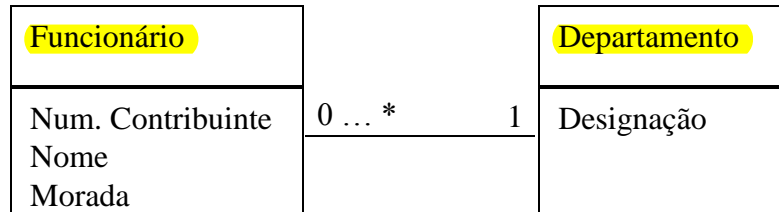


“um para muitos”

“um para um”

“muitos para muitos”

Associação “um para muitos”



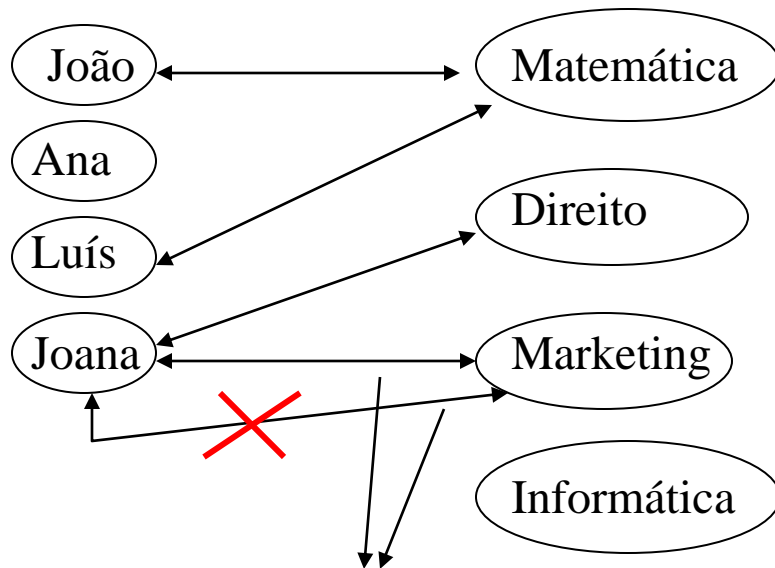
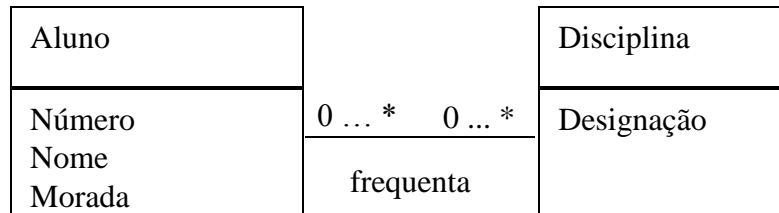
Semântica

Um funcionário tem **necessariamente** que estar **associado a um departamento**, e, a um departamento podem-se associar vários ou nenhum funcionários.

Funcional

Dado **um funcionário** é possível determinar em que **departamento ele trabalha**, e, dado um departamento é possível identificar os seus funcionários.

Associação “muitos para muitos”



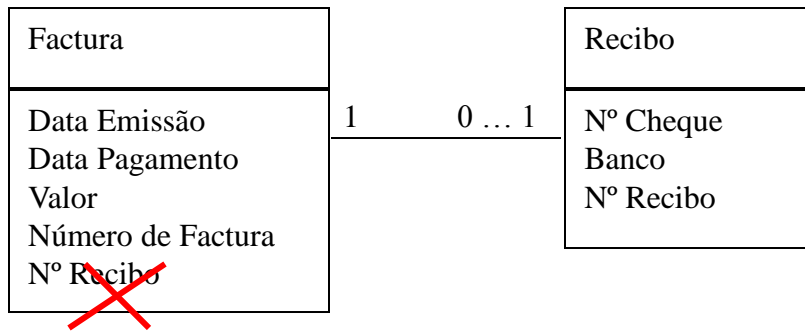
A mesma associação (domínio e co-domínio idênticos)

Um objecto não pode
estar **duplamente**
associado a outro objecto

(Joana / Marketing). À
semelhança das classes
(em que os objectos são
distintos), as **associações**
também têm que ser
distintas.

As associações podem ter
nomes, nomes esses que
terão que ser distintos

Associação “um para um”

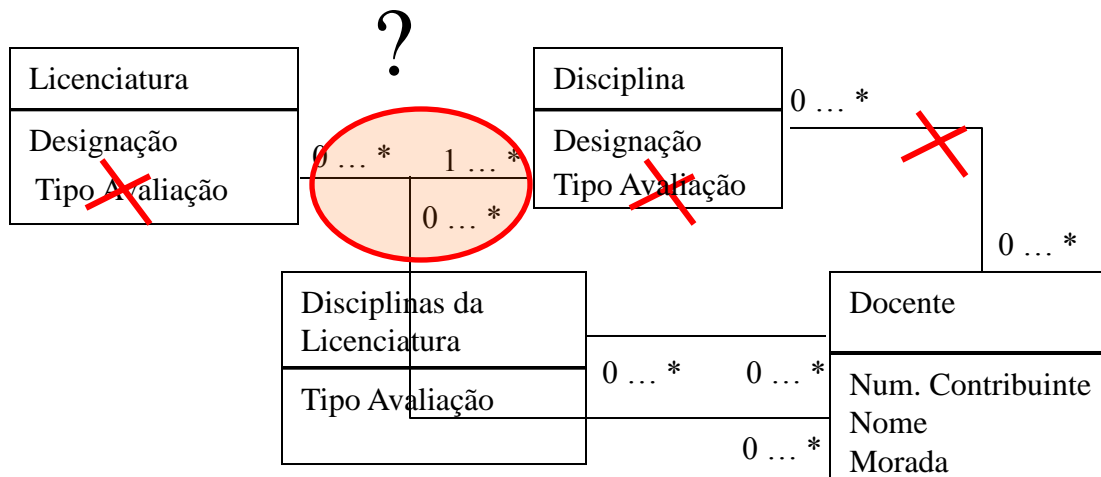


É a associação que *atribui* um número de recibo à factura. Caso contrário, uma factura estaria associada a dois recibos (o recibo indicado no atributo da factura e o recibo resultante da associação).

Classes Associativas

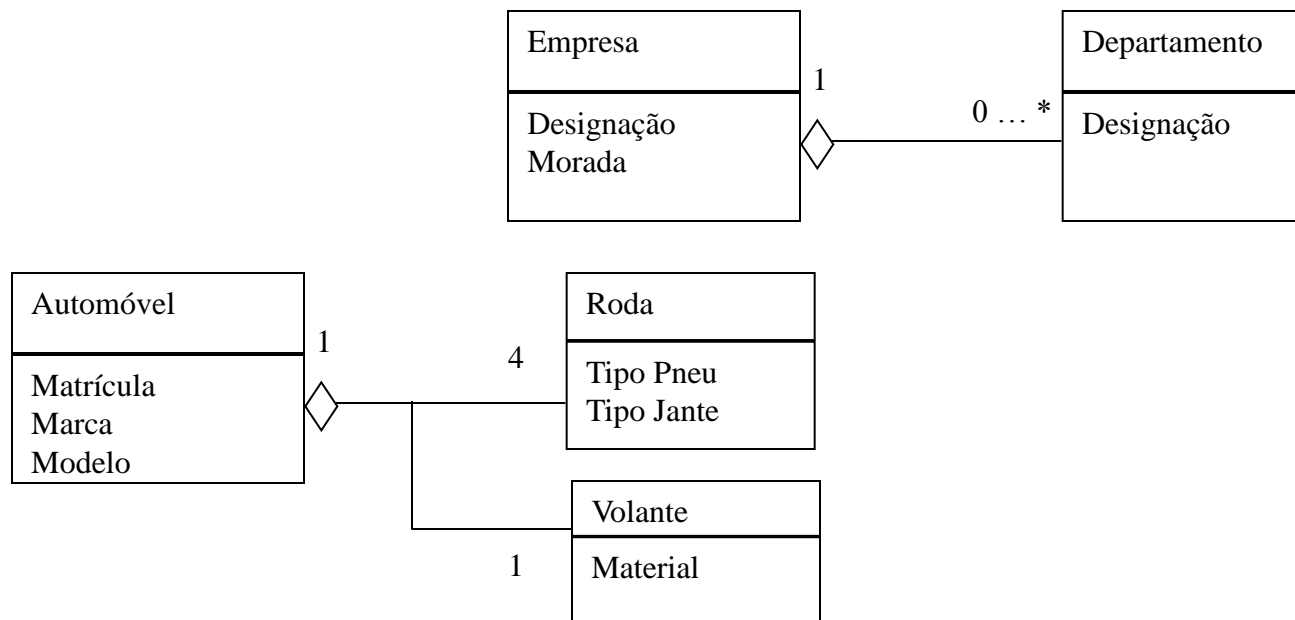
As Classes Associativas são associações que se *transformam* em classes quando é necessário:

- a) Colocar atributos na associação ou/e;
- b) Associar uma classe a uma associação.



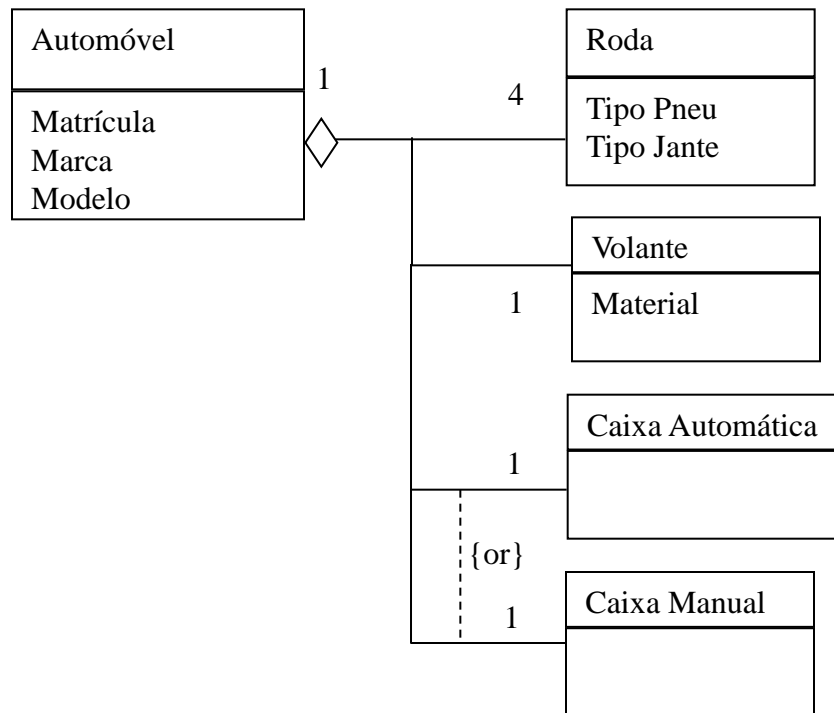
Agregações (I)

As Agregações são associações que se utilizam quando se pretende representar a **noção de Todo/Parte** (um todo constituído por partes). A associação “lê-se” de forma semelhante às restantes associações.



Agregações (II)

Indicação explícita de **disjunções (Or)** nas agregações. Logo, em qualquer associação.



Composições (I)

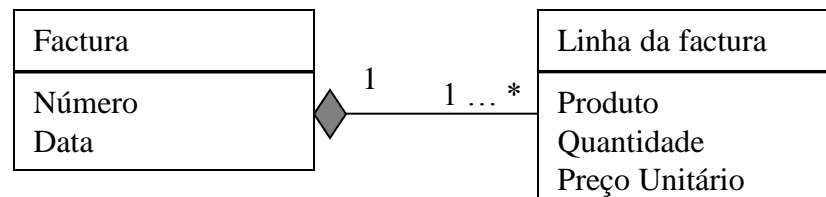
Factura nº 123		Data: 12/12/1999	
Cliente João Silva		Nº Cont. 1234567	
Produto	Quant.	P. Unit	Total
Produto A	2	5000	10000
Produto B	1	3000	3000
Produto X	3	2000	6000
Total			19000

Só existe no contexto da
factura 123

Factura nº 100		Data: 12/10/1999	
Cliente Ana Silva		Nº Cont. 1234568	
Produto	Quant.	P. Unit	Total
Produto X	2	5000	10000
Produto B	1	3000	3000
Produto Y	3	2000	6000
Total			19000

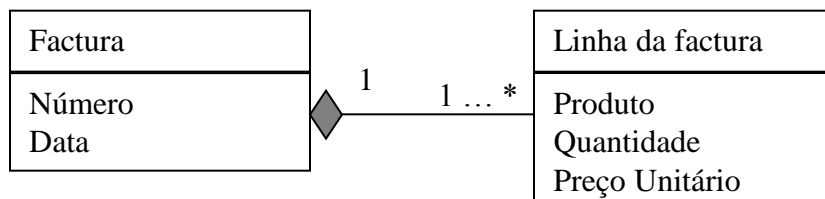
linhas diferentes,

daí que uma linha apenas
esteja associada a uma
factura.



Composições (II)

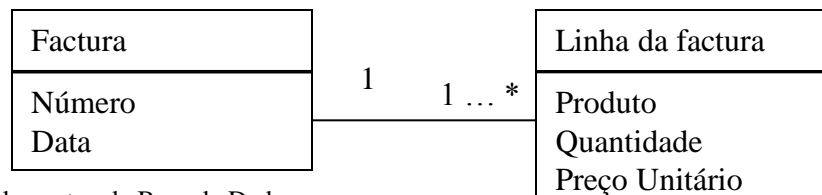
As composições são um caso especial de Agregações (normalmente associações “um para muitos”) que representam situações em que um objecto de uma classe (composição) resulta da composição de um conjunto de objectos de outra classe (componente), e em que os componentes apenas existem no contexto da classe composição. São **Agregações em que o sentido da pertença é mais forte.**



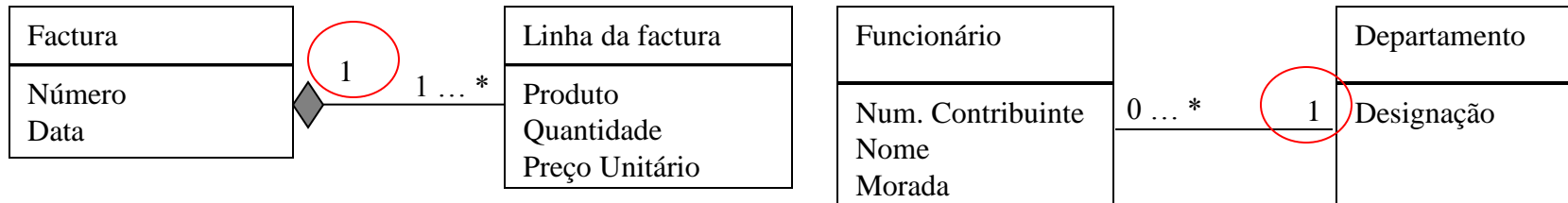
Uma factura **é uma composição** de *linhas*.

Caso optasse pela representação abaixo apresentada *escondia* uma realidade que pode ser considerada relevante: as linhas apenas podem ser identificadas se previamente for identificada a factura da qual fazem parte.

Note-se que em ambas as representações fica explicitamente descrito que, se removermos uma factura, as linhas dessa factura também deverão ser removidas.



Composições (III)

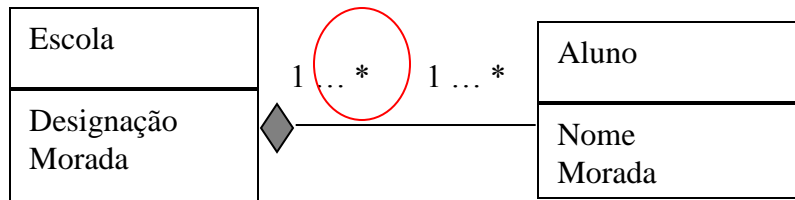


Apesar da obrigatoriedade existir em ambas as associações (1), são situações com semânticas distintas.

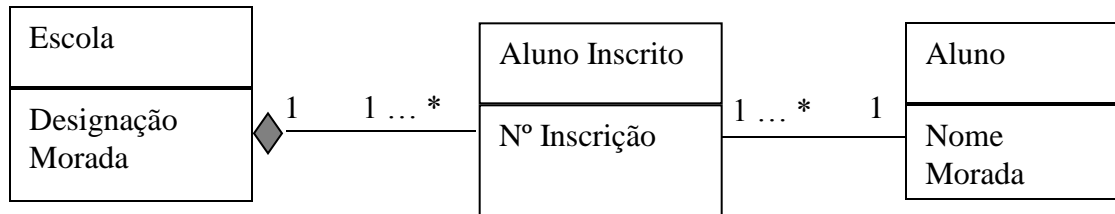
O funcionário **existe *per si***, não necessita de estar associado a um departamento para ser referido (ou seja, para ser editado, pesquisado, removido). No entanto, um funcionário que não trabalhe num departamento não é relevante para o domínio em causa (**se o seu departamento for removido ele terá que ser excluído**).

A Linha da factura só pode ser distinguida das restantes (logo, editada, pesquisada, removida) se for indicada a factura correspondente.

Composições (IV)

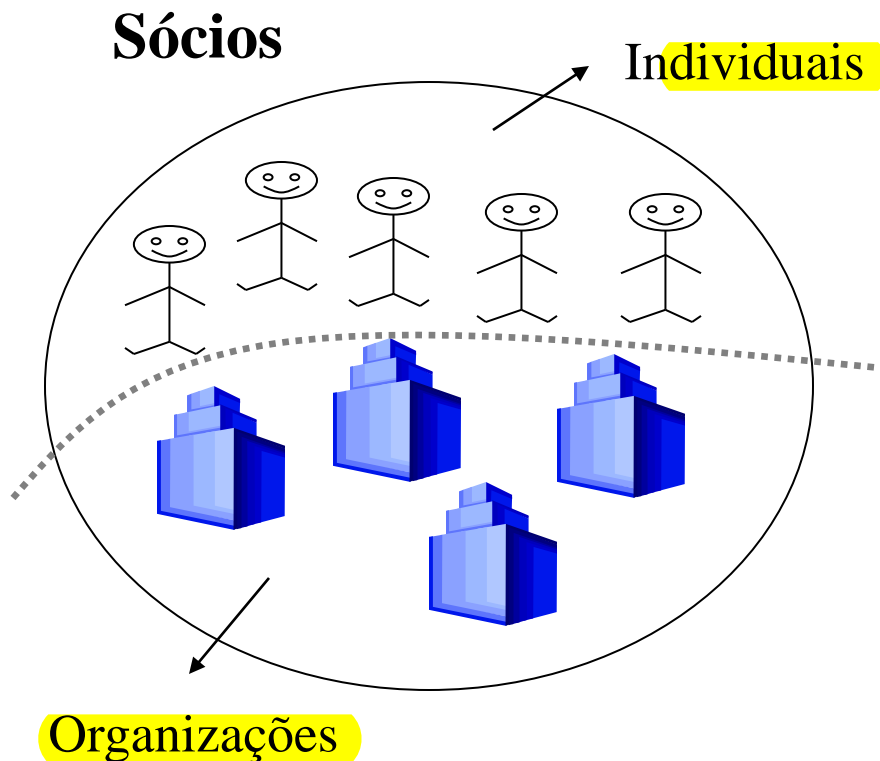


abreviatura de ...



Generalização (I)

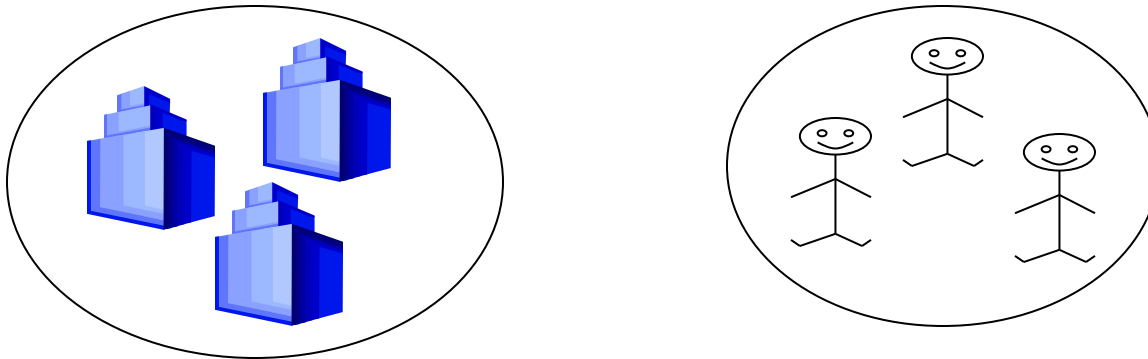
A generalização é uma relação (“um para um”) que permite representar a noção de pertença ou especificidade.



Todos os sócios partilham
informação comum:

Nome, morada, telefone,
valor quotização, etc. ...

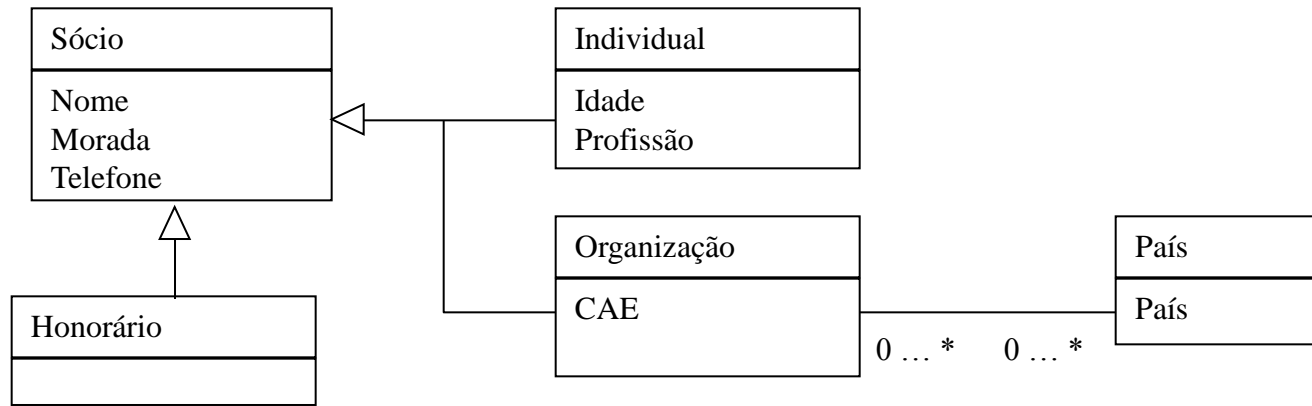
Generalização (II)



Porquê distingui-los?

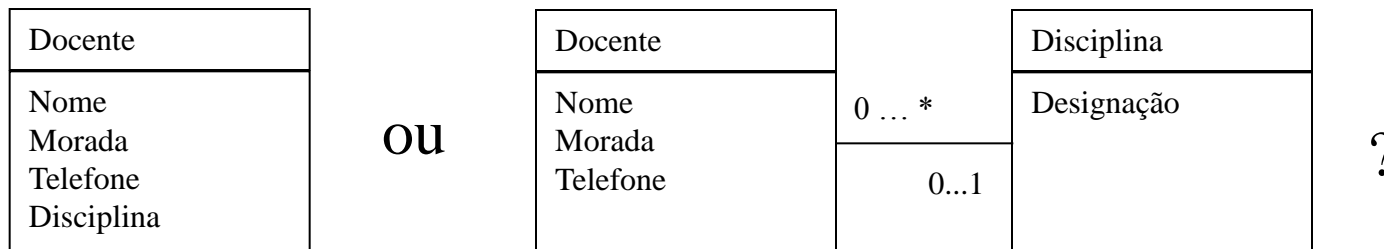
- 1) **Atributos específicos**, ex. CAE em Organizações, Idade em Individuais;
- 2) **Associações específicas**: ex. Mercados onde as Organizações actuam;
- 3) Dar relevo a uma **especificidade do domínio**.

Generalização (III)



- São associações “um para um”: um sócio apenas pode corresponder a uma organização e uma organização corresponde (obrigatoriamente) a um sócio;
- Um sócio não pode ser simultaneamente Individual e Organização;
- Um sócio pode não ser nem Individual nem Organização;
- Um sócio pode ser Honorário e Individual, ou Honorário e Organização;
- As subclasses herdam todos os atributos da *supraclasse*.

Atributos *versus* Classes (I)



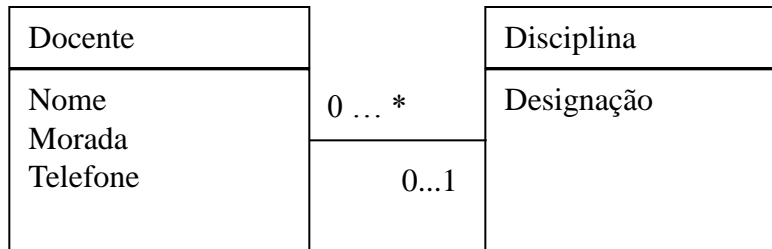
É relevante realçar a noção de disciplina ?

É importante referir que é **obrigatório** um docente estar associado a uma disciplina ?

É importante *validar* as disciplinas (apenas permitir associar a um docente uma disciplina pré-definida)?

A designação das disciplinas pode ser alterada?

Atributos *versus* Classes (II)



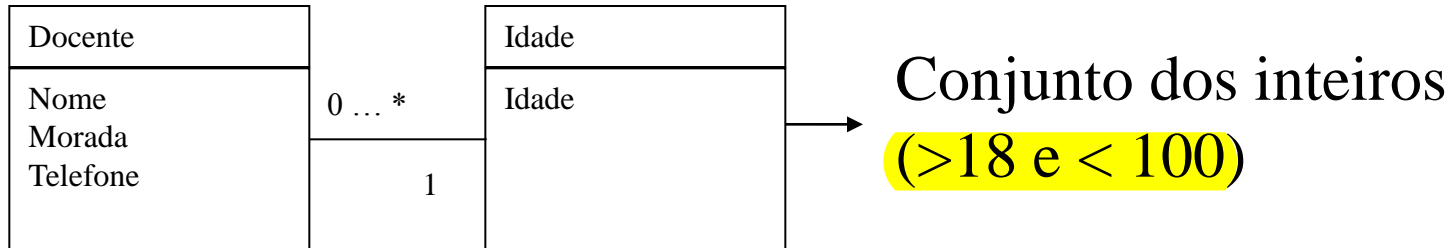
no computador

The screenshot shows a web-based form with the following fields and values:

- Numero**: 123
- Nome**: João Silva
- Disciplina**: A dropdown menu is open, showing a list of disciplines: Direito, História, Marketing, Sistemas de Informação, Sociologia, and Web Design.

Em rigor, **não é necessária** a Classe Disciplina para poder obter a funcionalidade indicada. No entanto, caso a classe não exista, as disciplinas válidas são constantes (não é permitida a sua manutenção).

Atributos *versus* Classes (III)



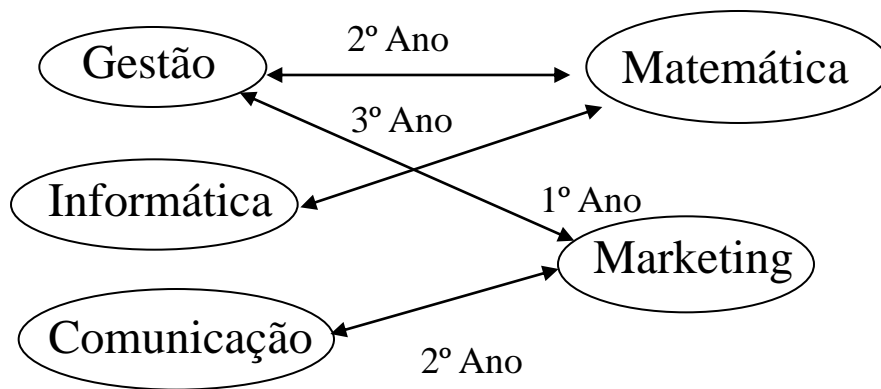
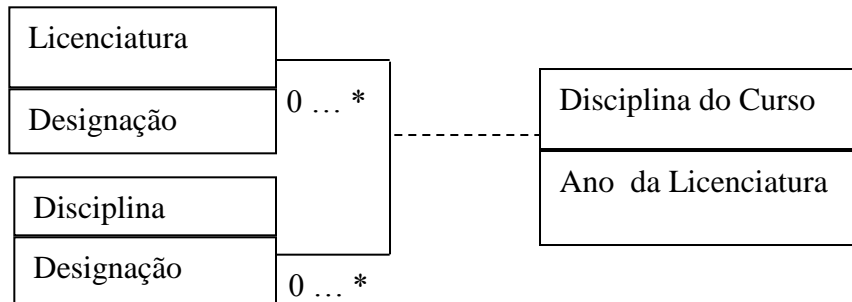
no computador

Numero: 123
Nome: João Silva
Idade: [18, 19, 20, 21, 22, 23, 24, 25]

Mesmo que pretenda a funcionalidade indicada, os limites de idade **são imutáveis**, logo a validação não necessita de ser efectuada através de uma classe Idade.

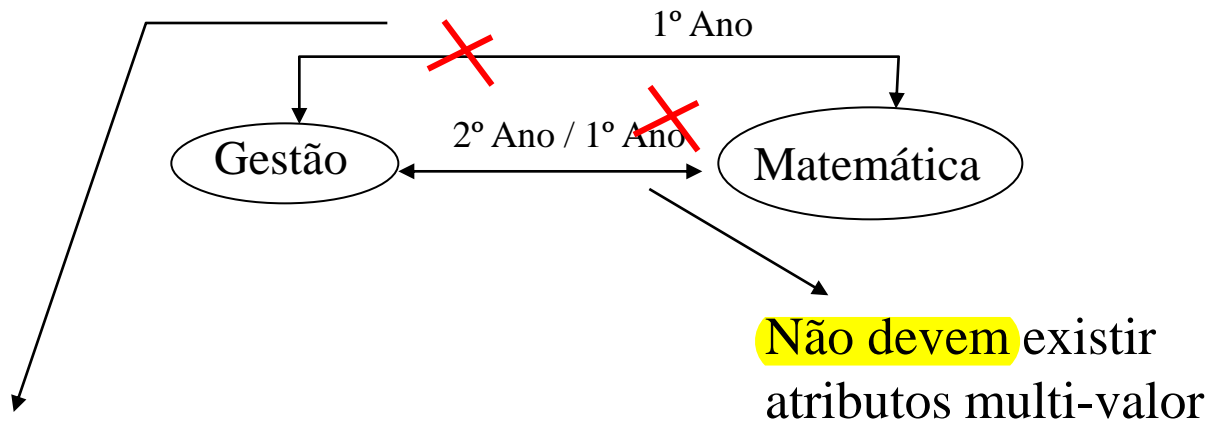
Não parece ser relevante destacar a Idade.

Associações n-árias (I)



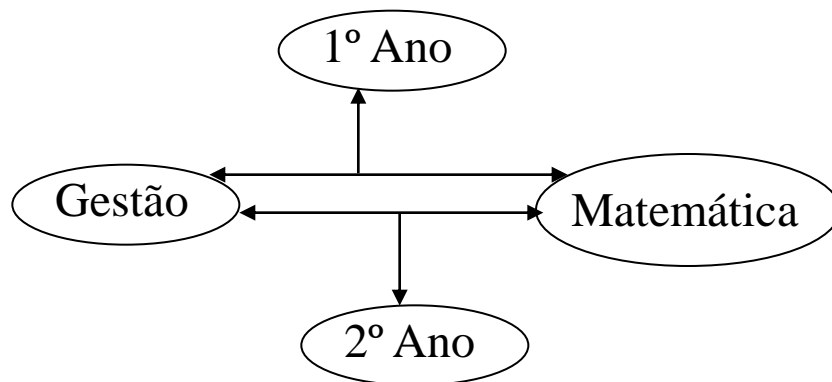
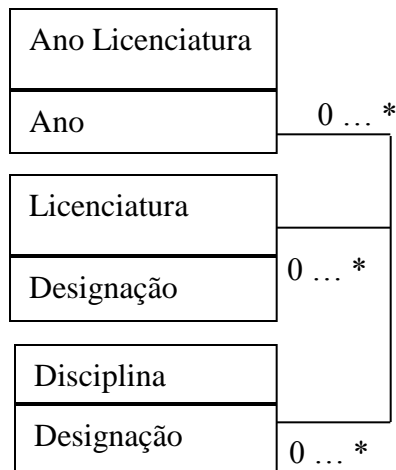
Associações n-árias (II)

Cenário de **Reestruturação de Licenciatura** (ano de transição):
Matemática ocorre no 1º e 2º ano da Licenciatura em Gestão.

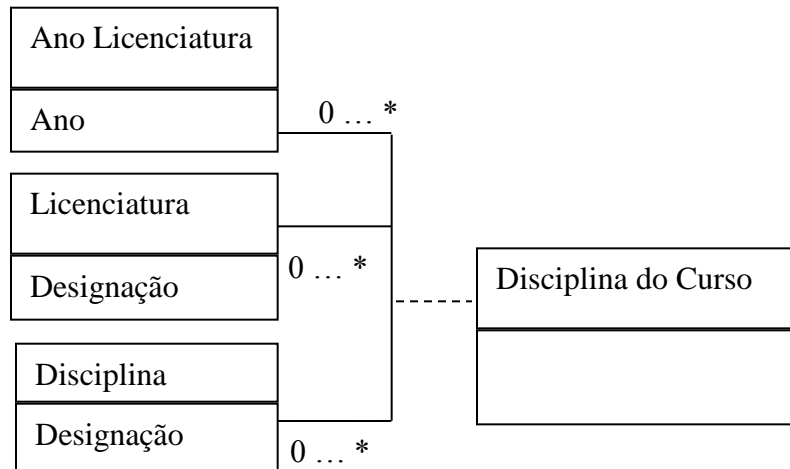


Apesar de o ano ser diferente nas duas associações, as associações são as mesmas: **uma associação é identificada** (distingue-se) através dos seus **argumentos** (e não pelos atributos).

Associações n-árias (III)

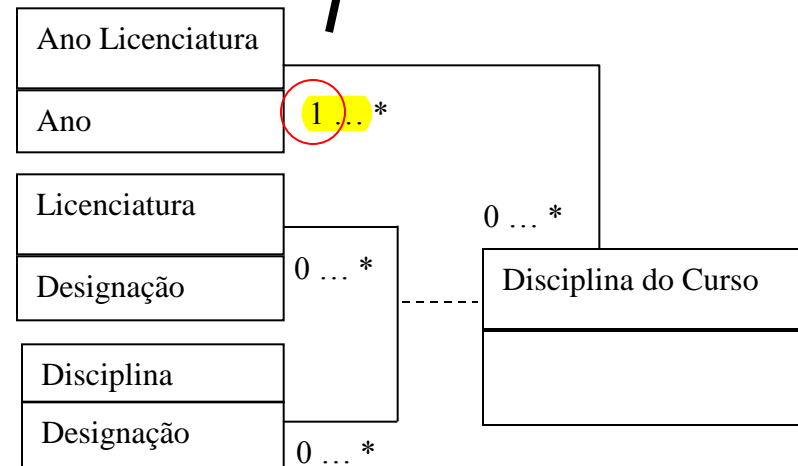


Associações n-árias (IV)



Uma instância em **Disciplina** do
Curso têm que estar obrigatoriamente
associada a um ano

alternativa →



Associações com apenas uma Classe

