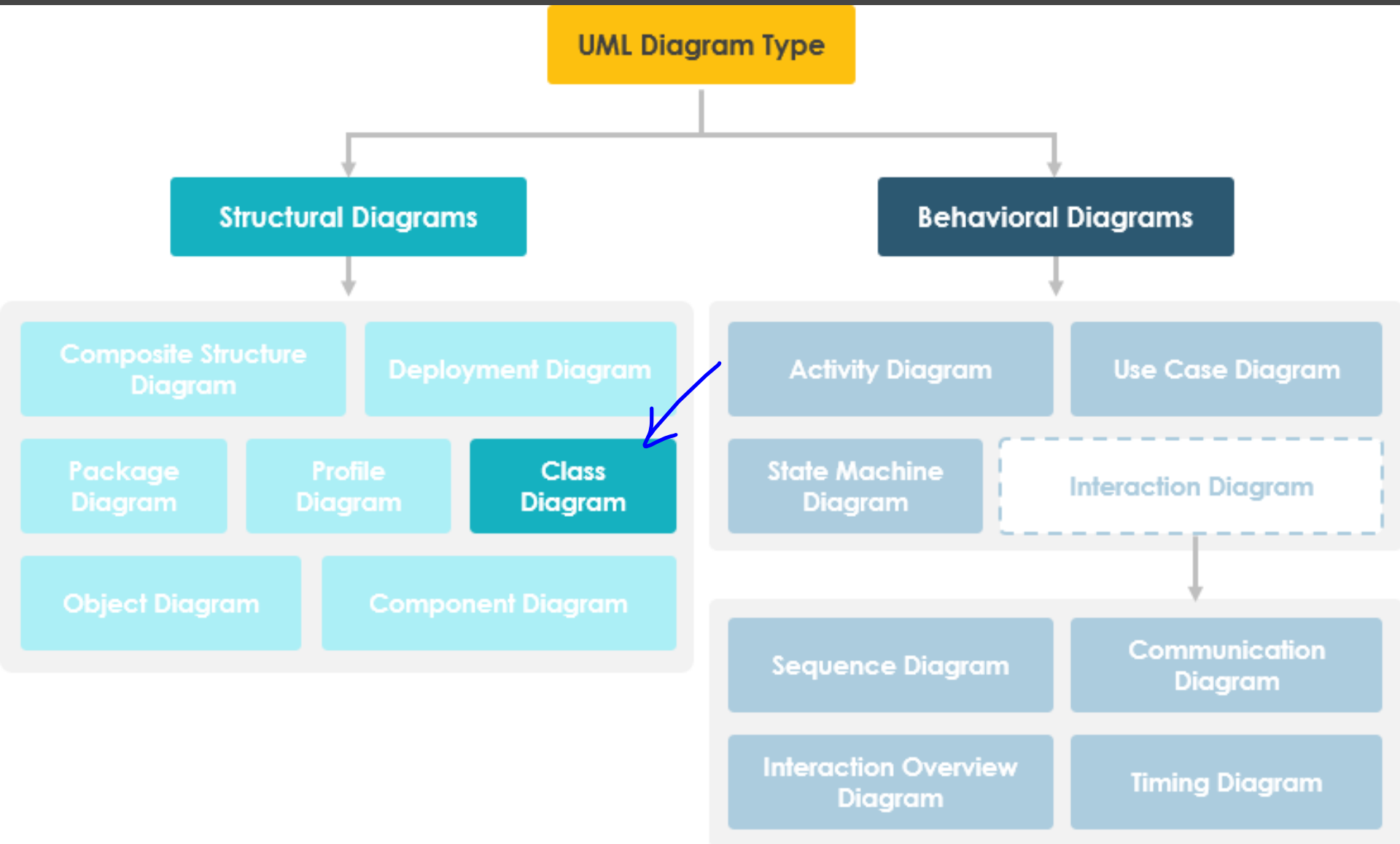


40431: Modelação e Análise de Sistemas

# Análise por objetos – situações de modelação mais avançadas (classes)

Ilídio Oliveira

v2022-10-21



Um tipo de dados auxiliar.  
Não é um conceito do domínio.

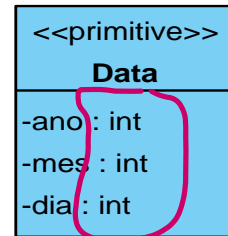
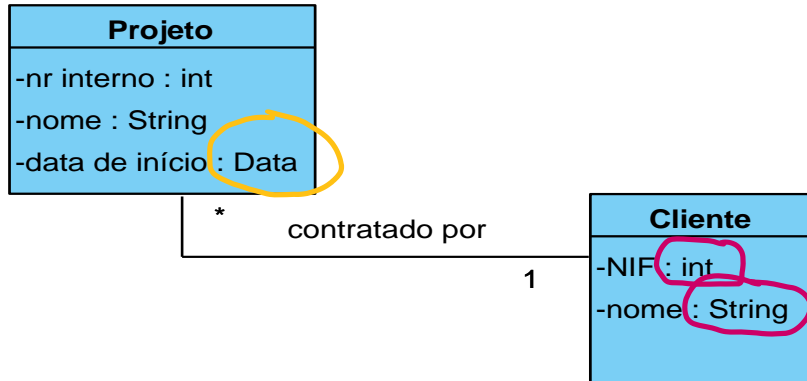
## Tipos de dados

Um modelo completo especifica os tipos de dados dos atributos.

Podemos usar os tipos "básicos" ou criar tipos adicionais

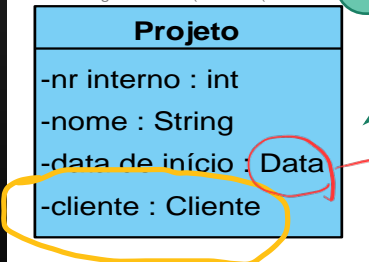
Na verdade, qualquer classe pode ser usada para indicar um tipo de dados de um atributo ( básicos vs objeto)

Visual Paradigm Standard (I Oliveira (Universidade de Aveiro))

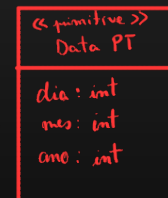


Um tipo de dados que é uma classe do domínio implica uma associação. Vamos preferir mostrar como associação.

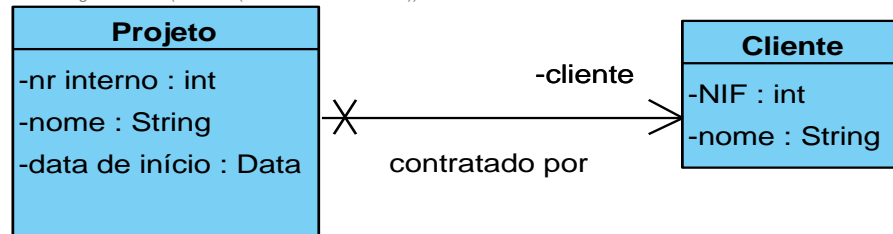
Visual Paradigm Standard (I Oliveira (Universidade de Aveiro))



*mão está definida, logo criamos*

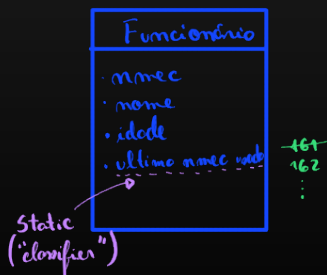
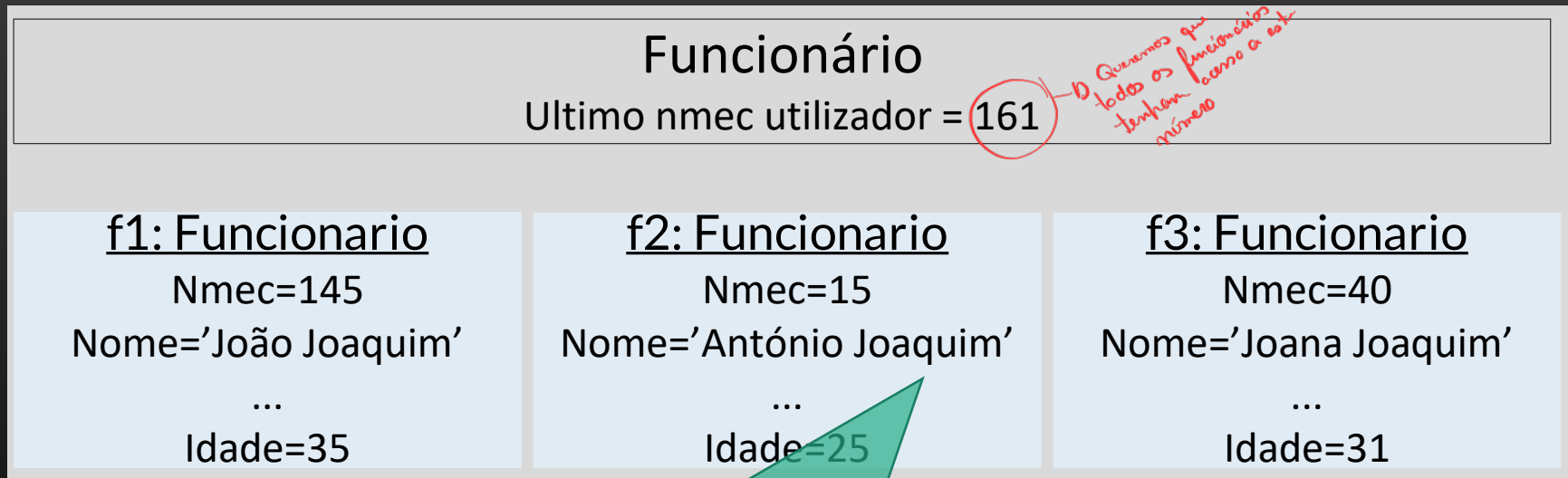


Visual Paradigm Standard (I Oliveira (Universidade de Aveiro))



# Atributos da classe (static)

Atributos de classe são partilhados por todas as instâncias



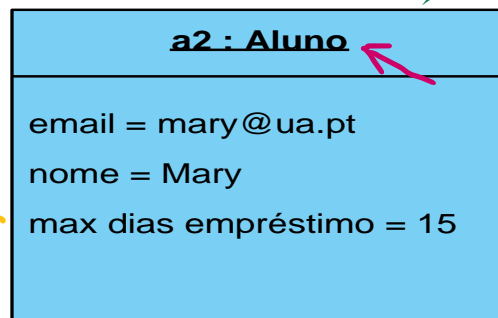
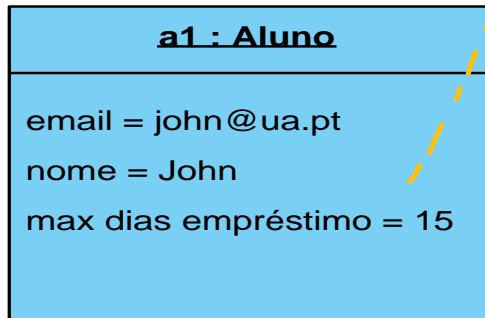
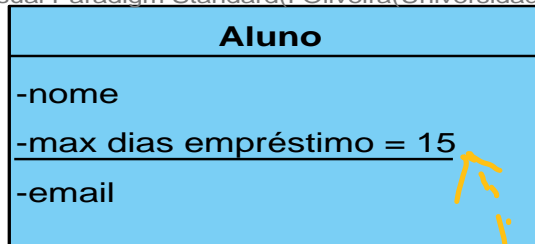
Os objetos f1, f2, f3 têm estado interno, independente entre si.  
Podem existir atributos que afetem o conjunto dos objetos, i.e., atributos cujo valor é partilhado por todos: "Ultimo nmec utilizado".  
Por isso, o âmbito deste atributo deve ser o classificador (e não a instância)

# Atributos da classe (*static*)

Classe com um atributo *static* (valor partilhado por todas as instâncias)

Alguns objetos que instanciam a classe. E.g.: o objeto chamado "a2" do tipo "Aluno"

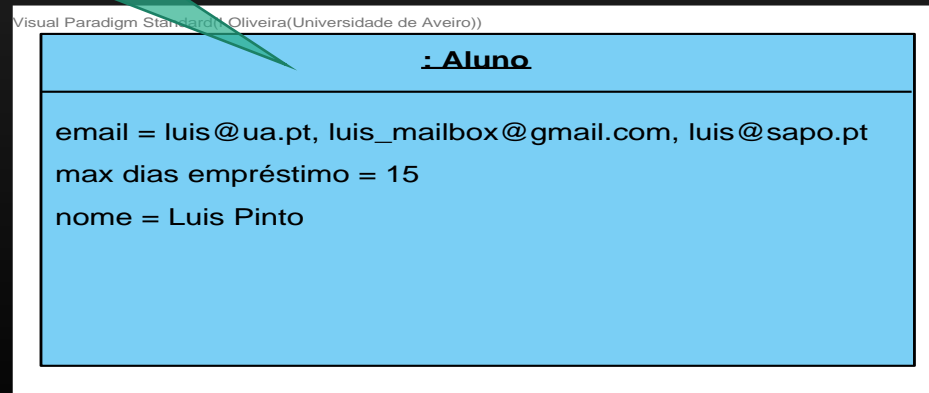
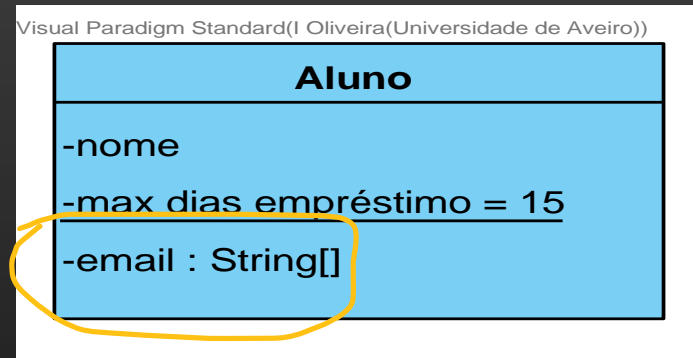
Visual Paradigm Standard (I Oliveira (Universidade de Aveiro))



# Atributos multi-valor

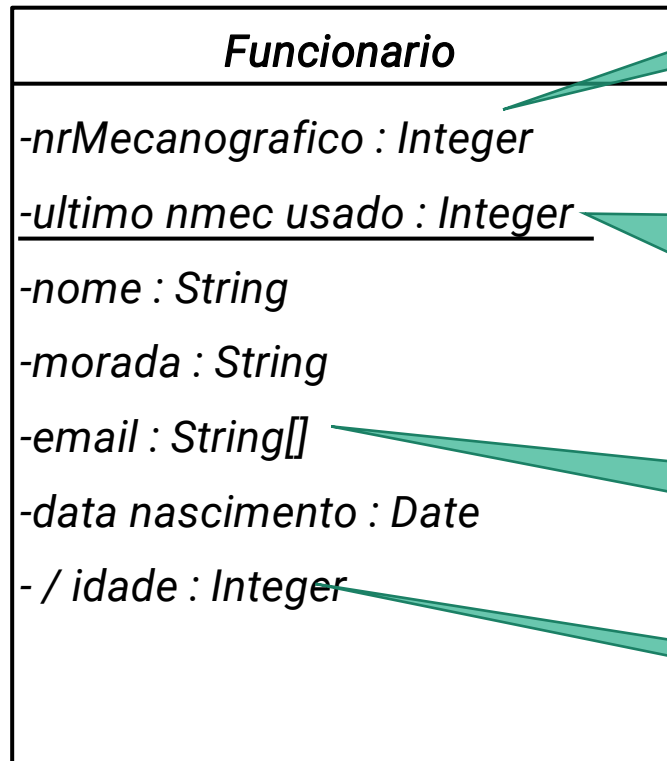
## Conceito de “vetor”

Exemplo: uma instância anônima, da classe Aluno, com os atributos (*slots*) definidos. Email é um atributo multi-valor, nesta instância, tem associada uma lista de 3 Strings



# Caraterização dos atributos

Visual Paradigm Standard(Universidade de Aveiro)



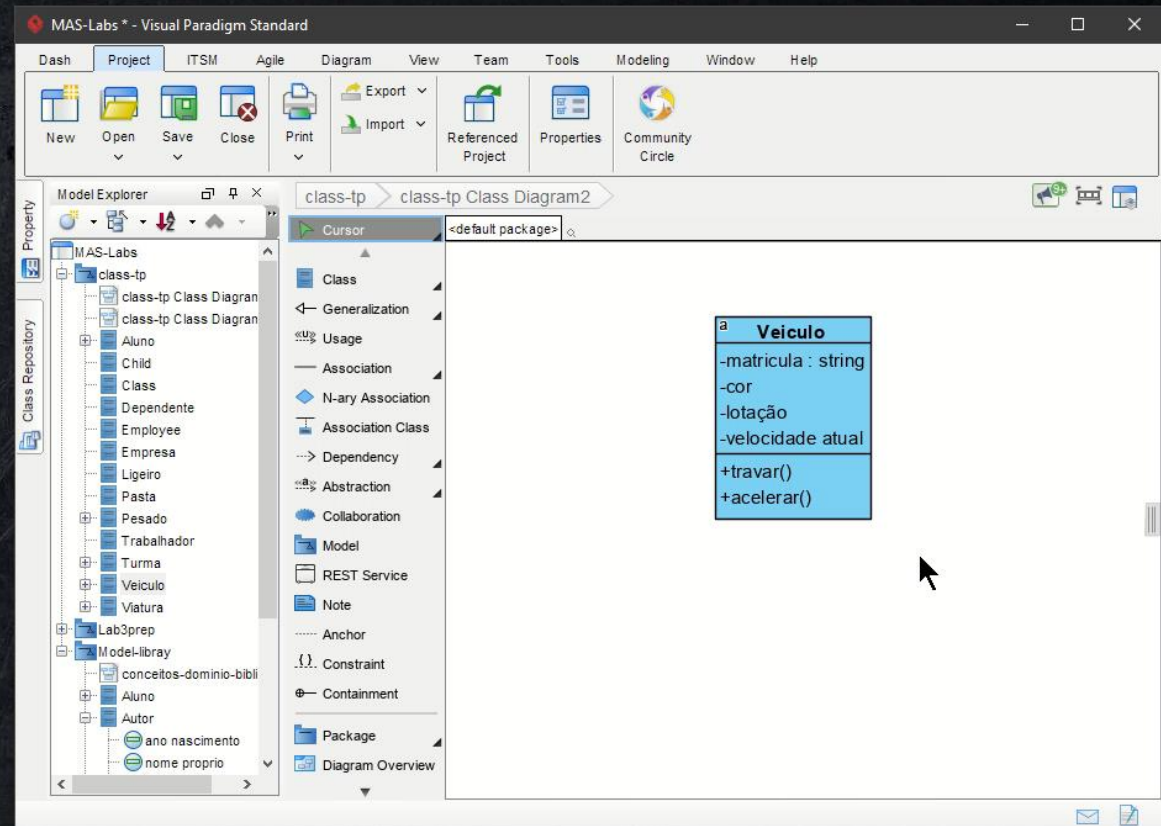
Tipo de dados (do atributo)

Atributo definido na classe,  
comum a todas as instâncias  
(*static*)

Multiplicidade (valores)

Atributo derivado (podia ser deduzido  
do restante estado)

# Caraterização dos atributos no Visual Paradigm

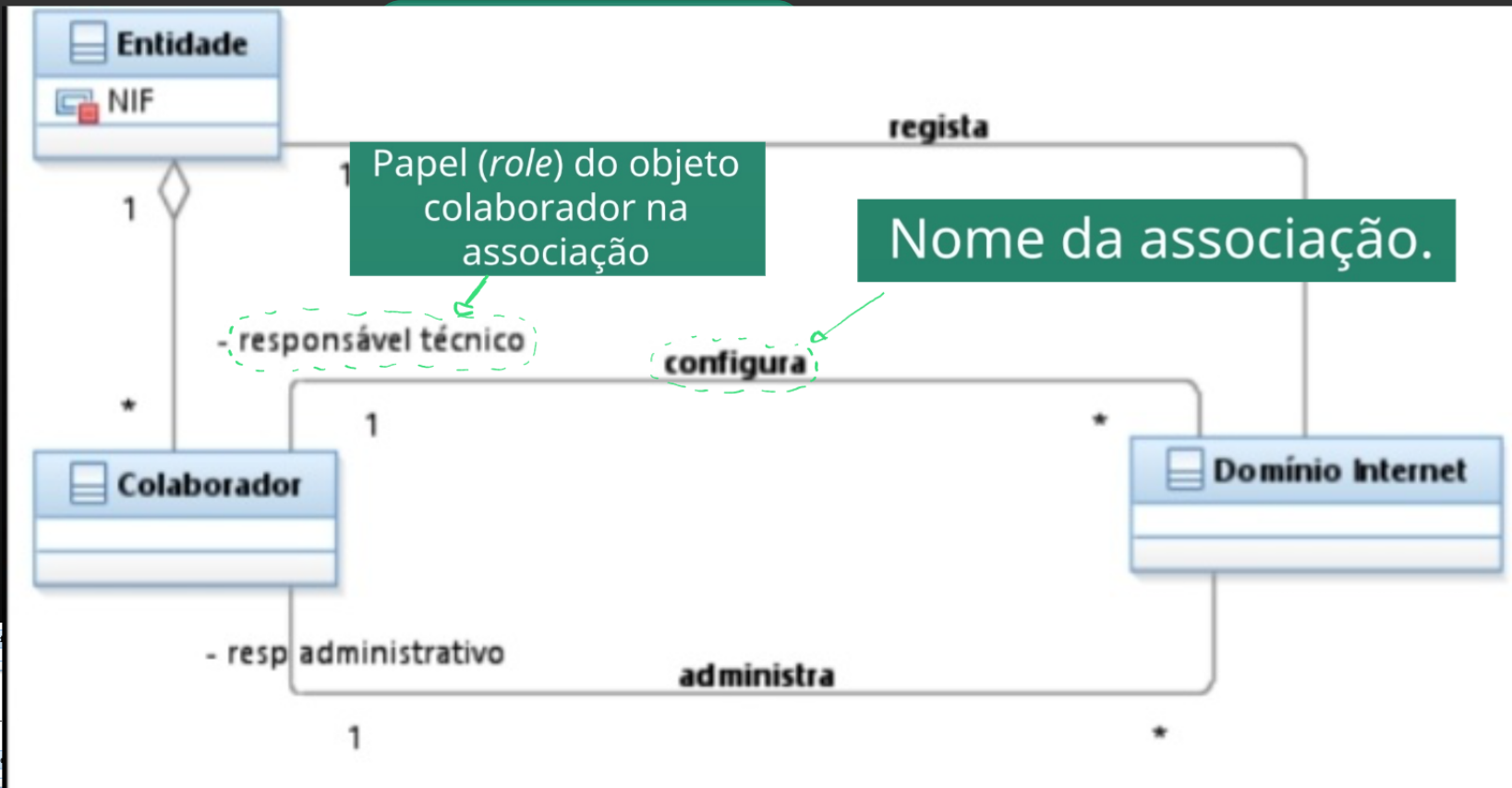




# Papéis ≠ nome da associação

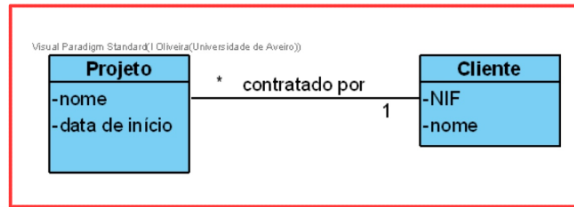
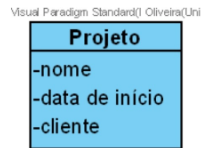
O **Nome** deve ajudar a ler a associação, como uma frase.

O **Papel** caracteriza a forma como instâncias de um tipo participam na associação.



## Caso 1

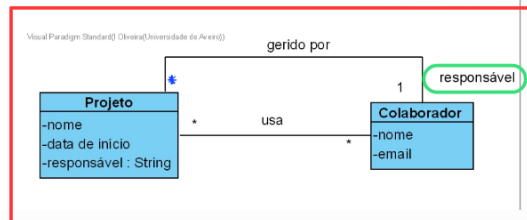
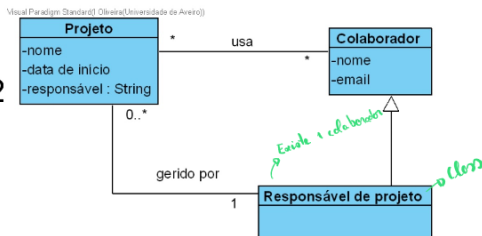
Atributo vs Conceito



→ Mais adequado!

## Caso 2

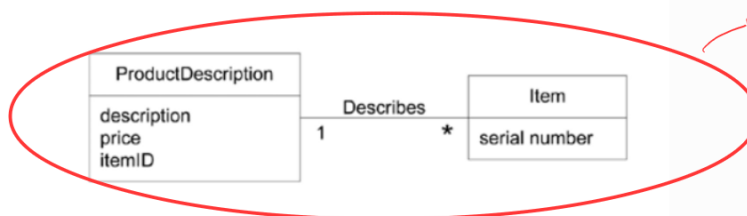
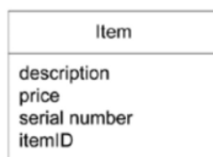
Papéis



→ Mais adequado!  
↓  
"Papéis" não se usa hierarquia NORMALMENTE

## Caso 3

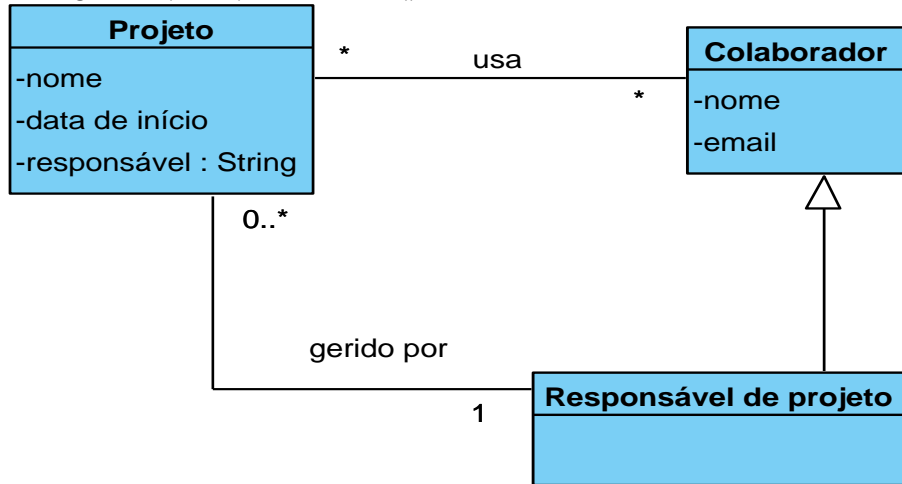
"Descrição do produto" vs "exemplares"



→ Mais adequado!  
↓  
Porque existem vários exemplares de 1 modelo

# Opções de modelação: papéis

Visual Paradigm Standard(I Oliveira(Universidade de Aveiro))



Visual Paradigm Standard(I Oliveira(Universidade de Aveiro))



## Agregação vs composição

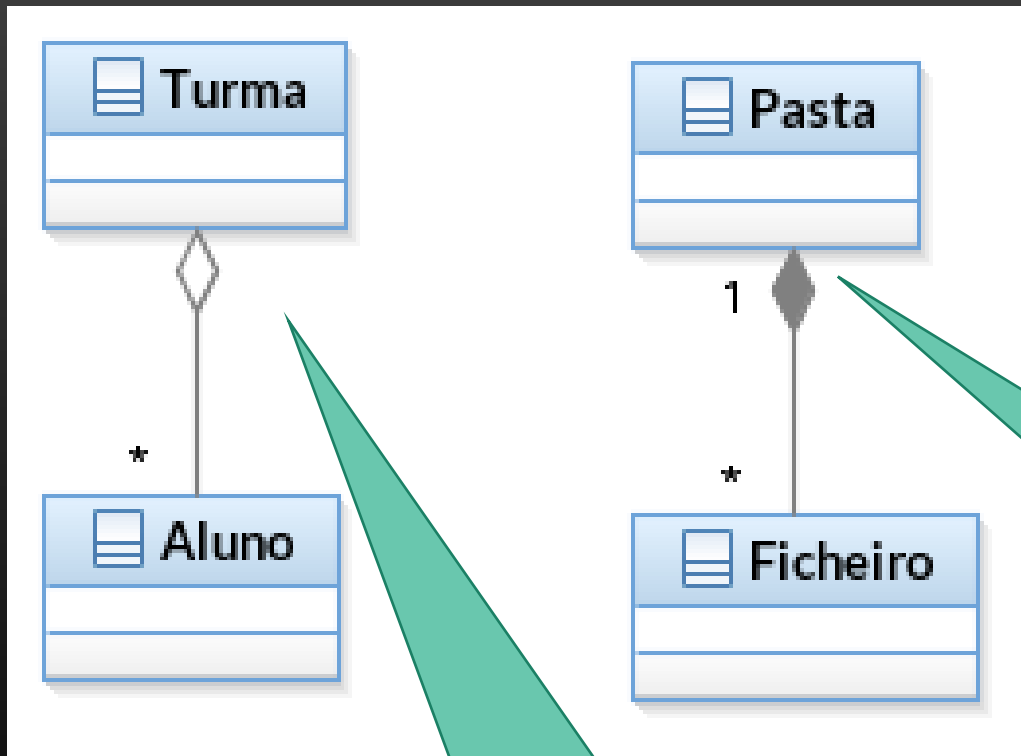
**Agregação:** A comporta partes B de forma não exclusiva

As instâncias de B são partilháveis

**Composição:** A comporta partes B de forma exclusiva

Os objetos B são partes constituintes de A

Instâncias de B não são partilháveis



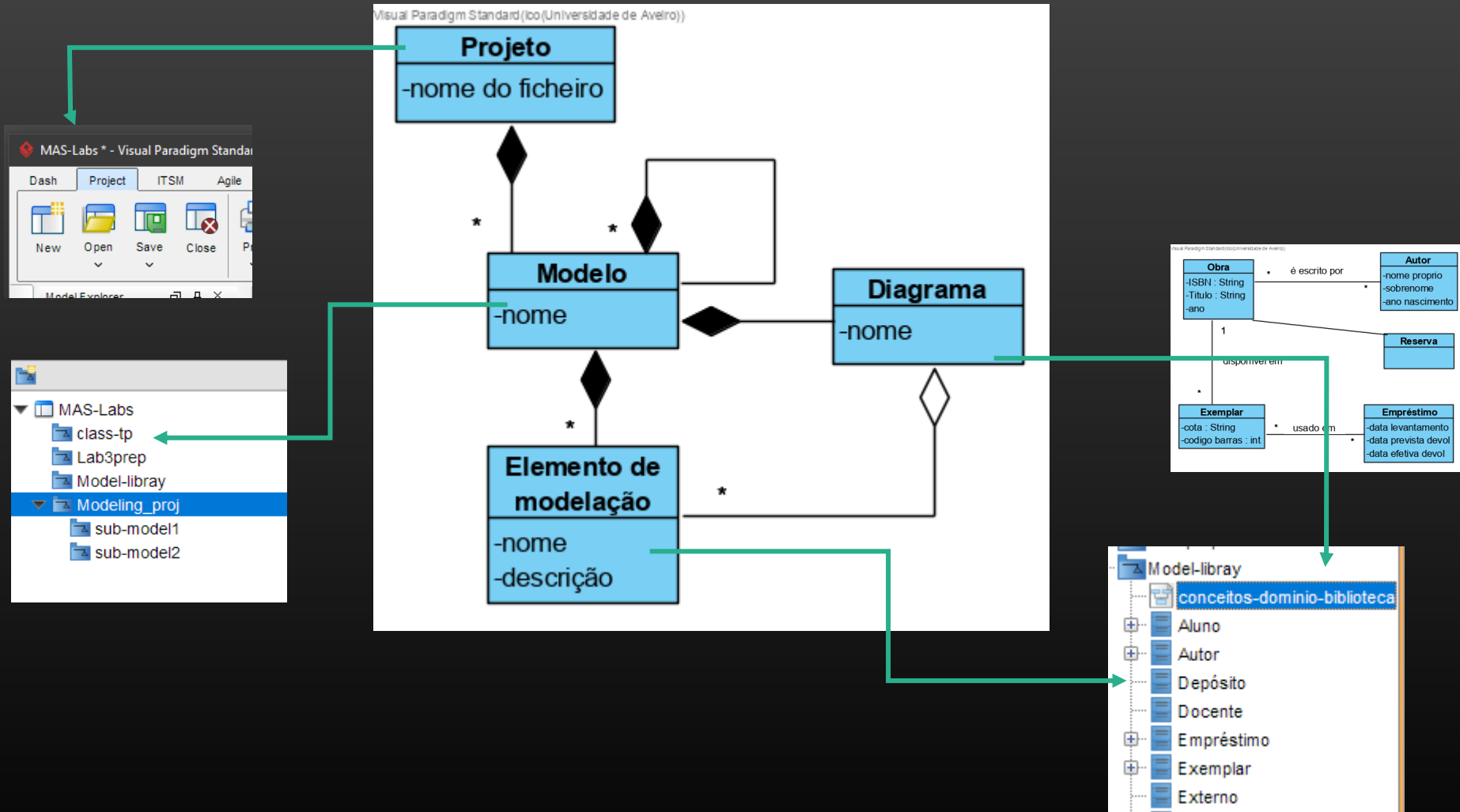
### Agregação

(a Turma agrupa objectos Aluno, mas o mesmo aluno pode ser parte de várias turmas → Aluno é partilhado)

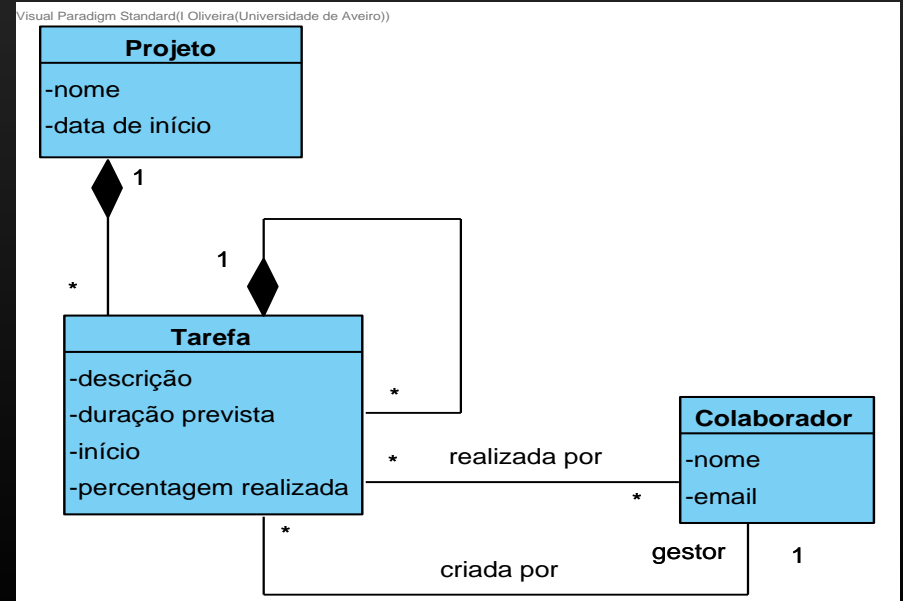
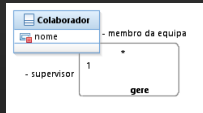
### Composição

(a Pasta comporta vários Ficheiros; um ficheiro nunca faz parte de várias pastas → exclusivo de uma Pasta)

# Exemplo: conceitos num projeto de modelação

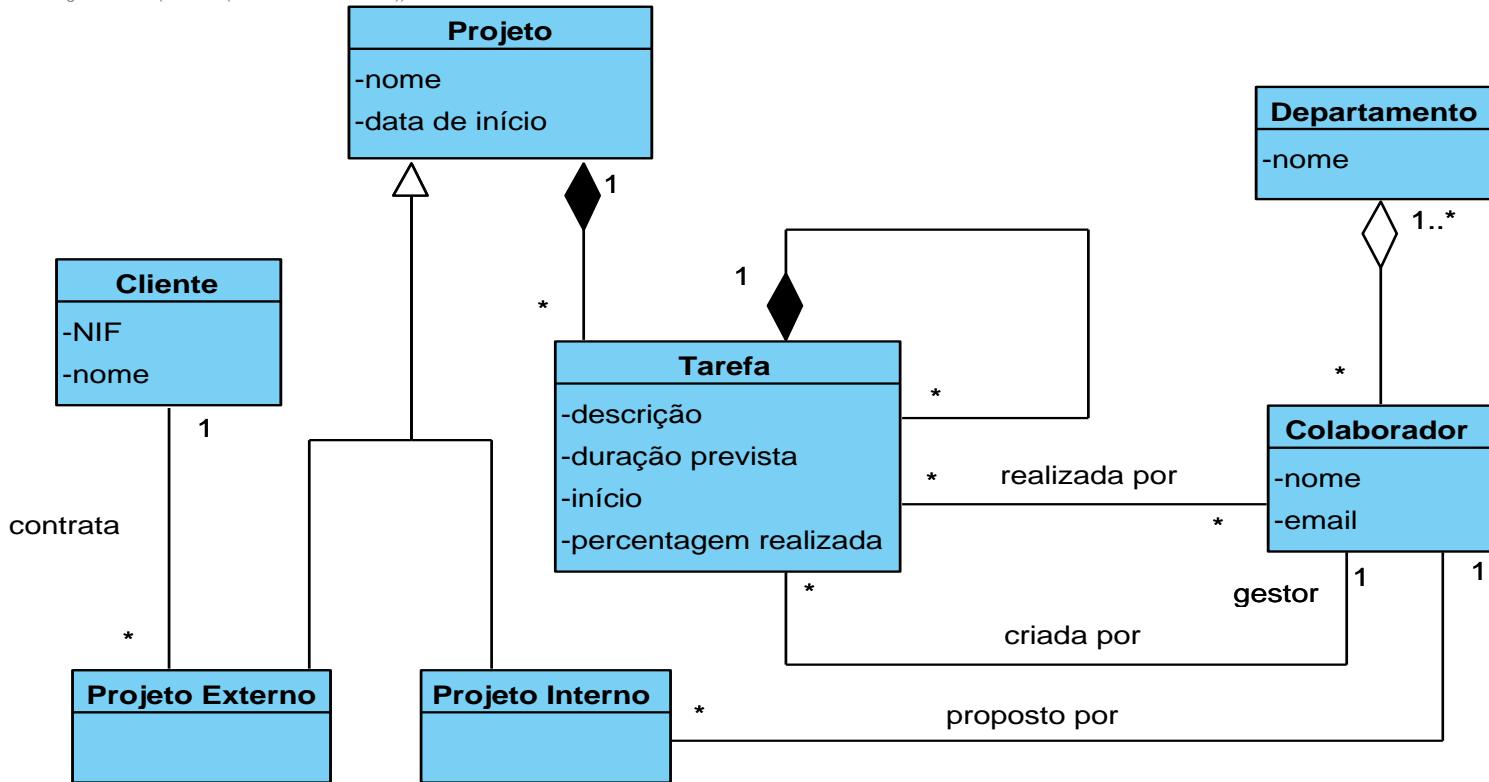


# Associações reflexivas relacionam instâncias da mesma classe



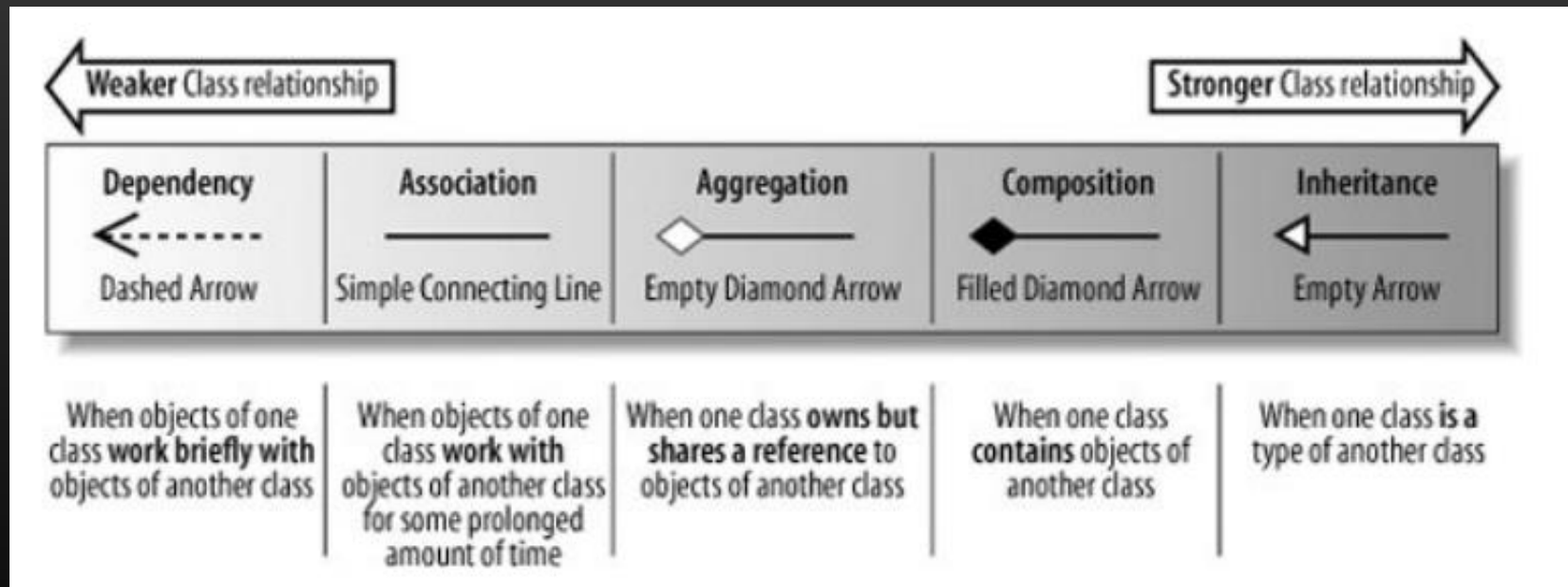
# “força” da associação entre os conceitos

Visual Paradigm Standard (I Oliveira (Universidade de Aveiro))



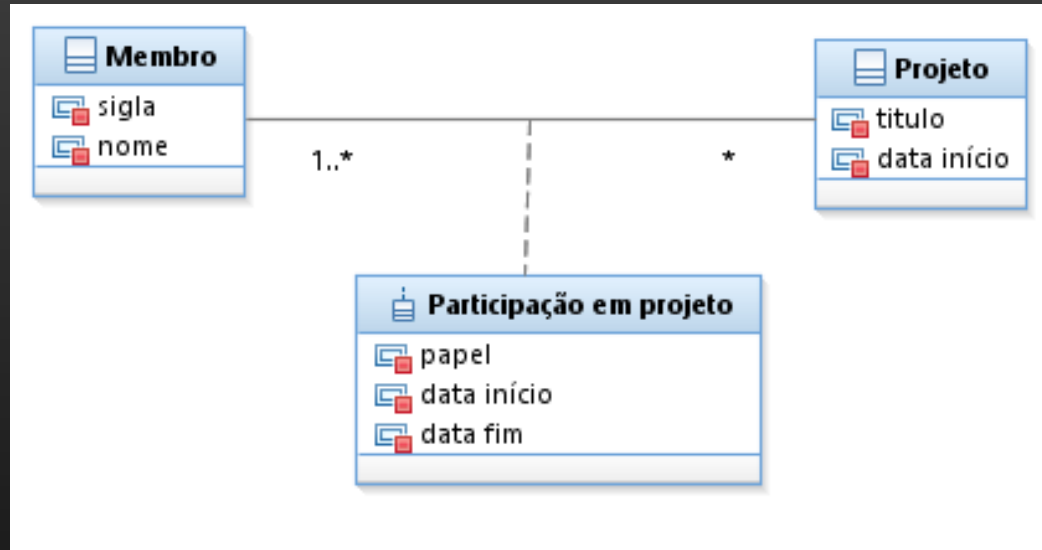
# “Força” da ligação estrutural entre classes

Cinco tipos diferentes de relacionamento entre classes, com diferentes níveis de ligação





## Classes-associação captam a informação que descreve o relacionamento

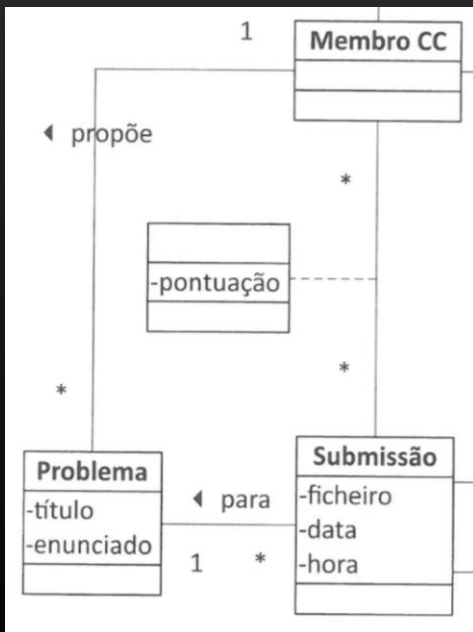
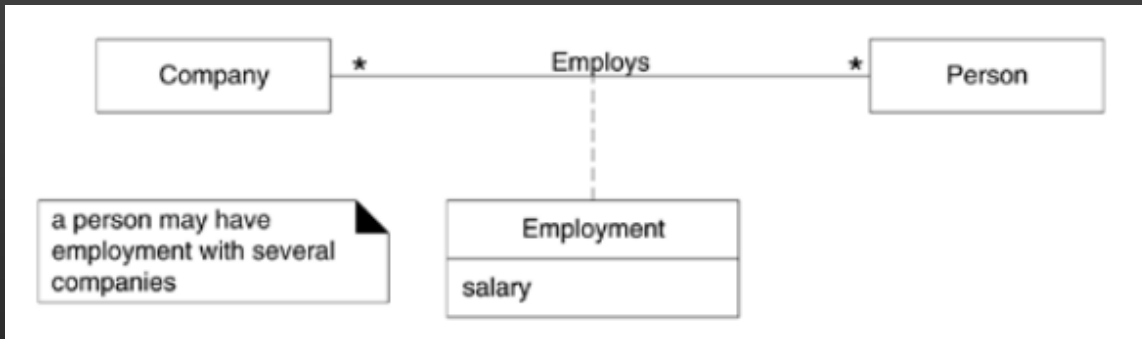


## Indicações para o uso de uma classe-associação no modelo do domínio

Um atributo está subordinado à ocorrência uma associação.

As instâncias da classe-associação têm um tempo de vida dependente da associação

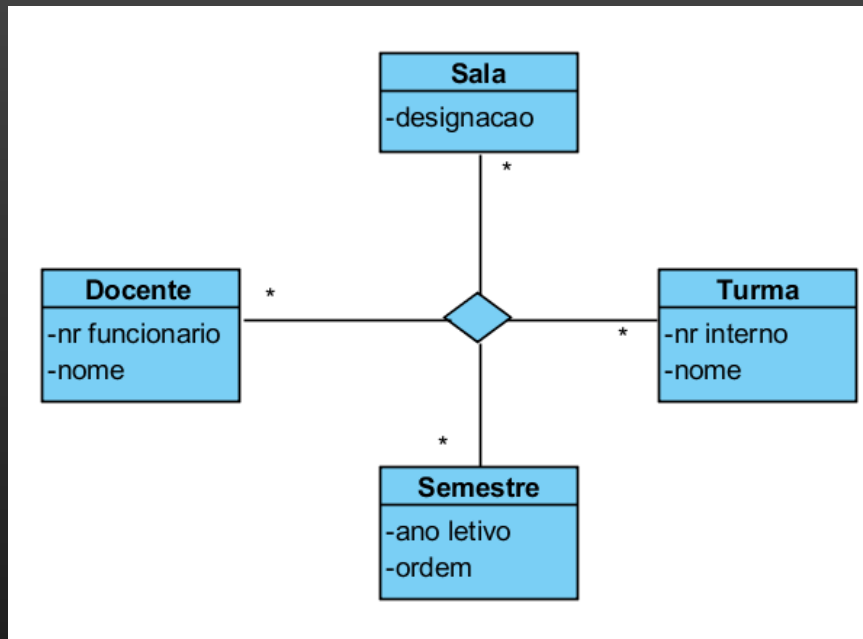
Há uma relação de N:M entre dois conceitos e é necessária informação para caraterizar a própria associação



## Associações N-árias

Vários objetos (de tipos diferentes) relacionados ao mesmo tempo, numa associação

Uma aula (presencial...) associa, ao mesmo tempo, um docente, uma sala, uma turma, e um semestre.



## Gestão de projetos (exemplo)

*Role*

*Visibility*

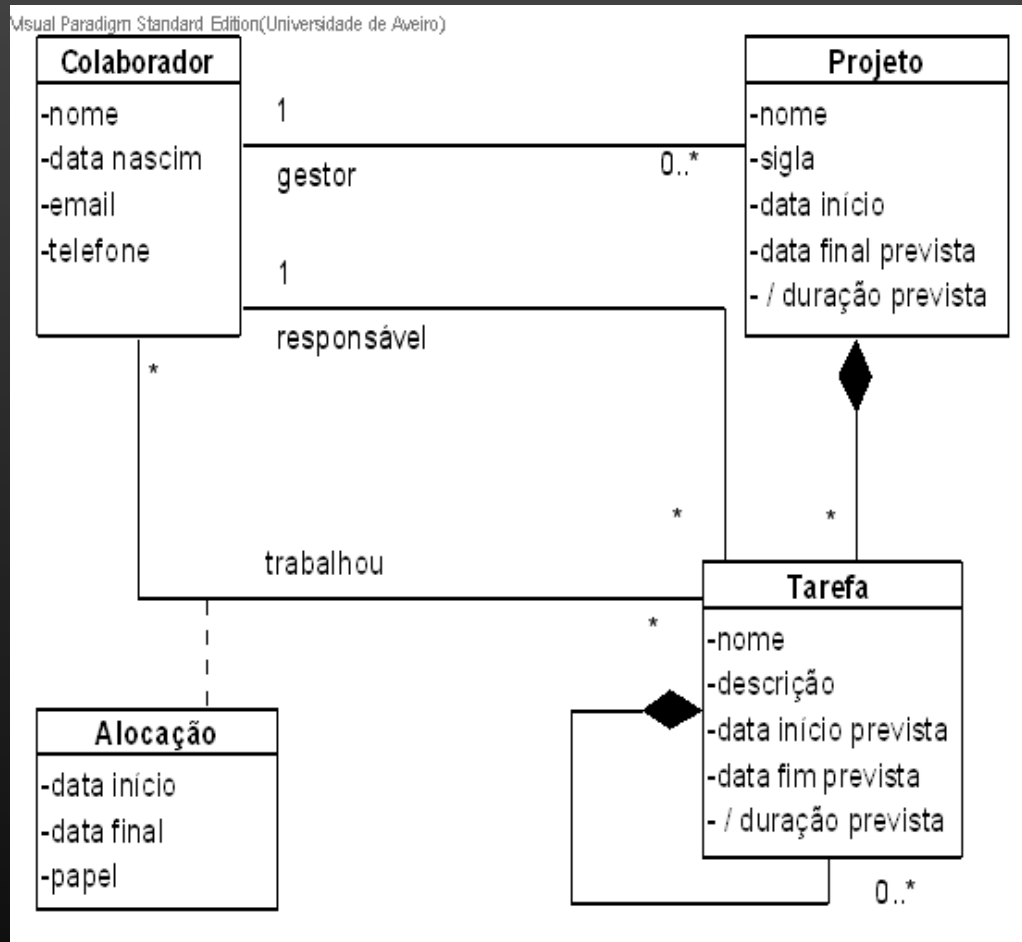
*Multiplicity*

*Derived attribute*

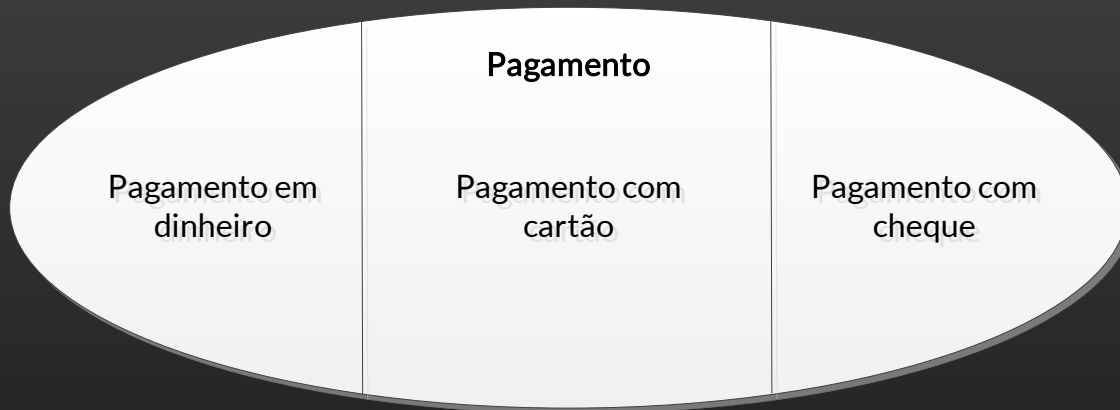
*Composition*

*Association class*

*Reflexive association*

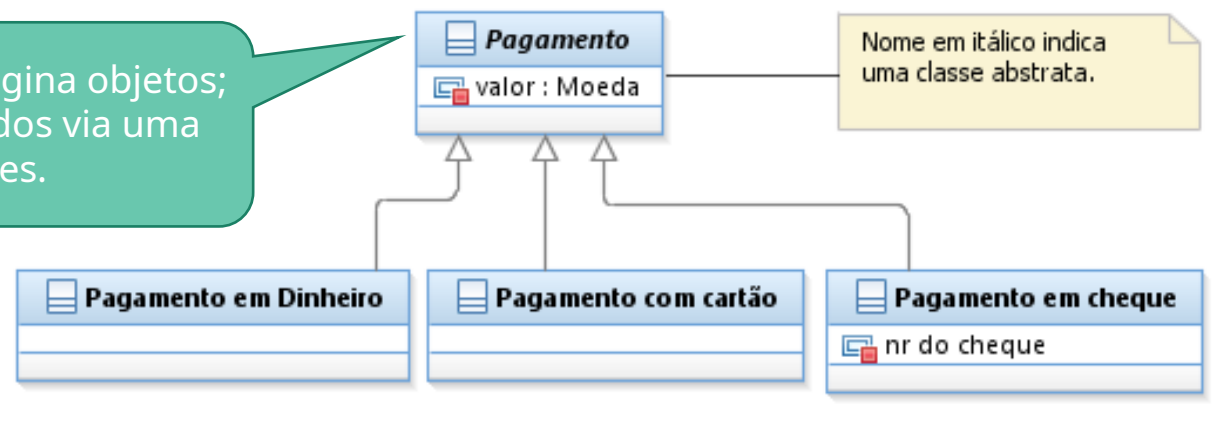


# Uma classe abstrata não é instanciada diretamente

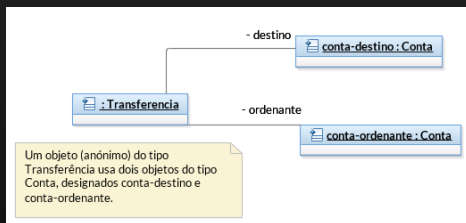


Um Pagamento tem de ser instanciado numa das formas indicadas pelas subclasses: em Dinheiro, Cartão ou Cheque. Pagamento deve ser uma classe abstrata.

A classe Pagamento não origina objetos; os objetos devem ser criados via uma das especializações.



# Diagramas de objetos



Cada “caixa” representa uma instância (no exemplo, 3 instâncias de 2 classes diferentes)