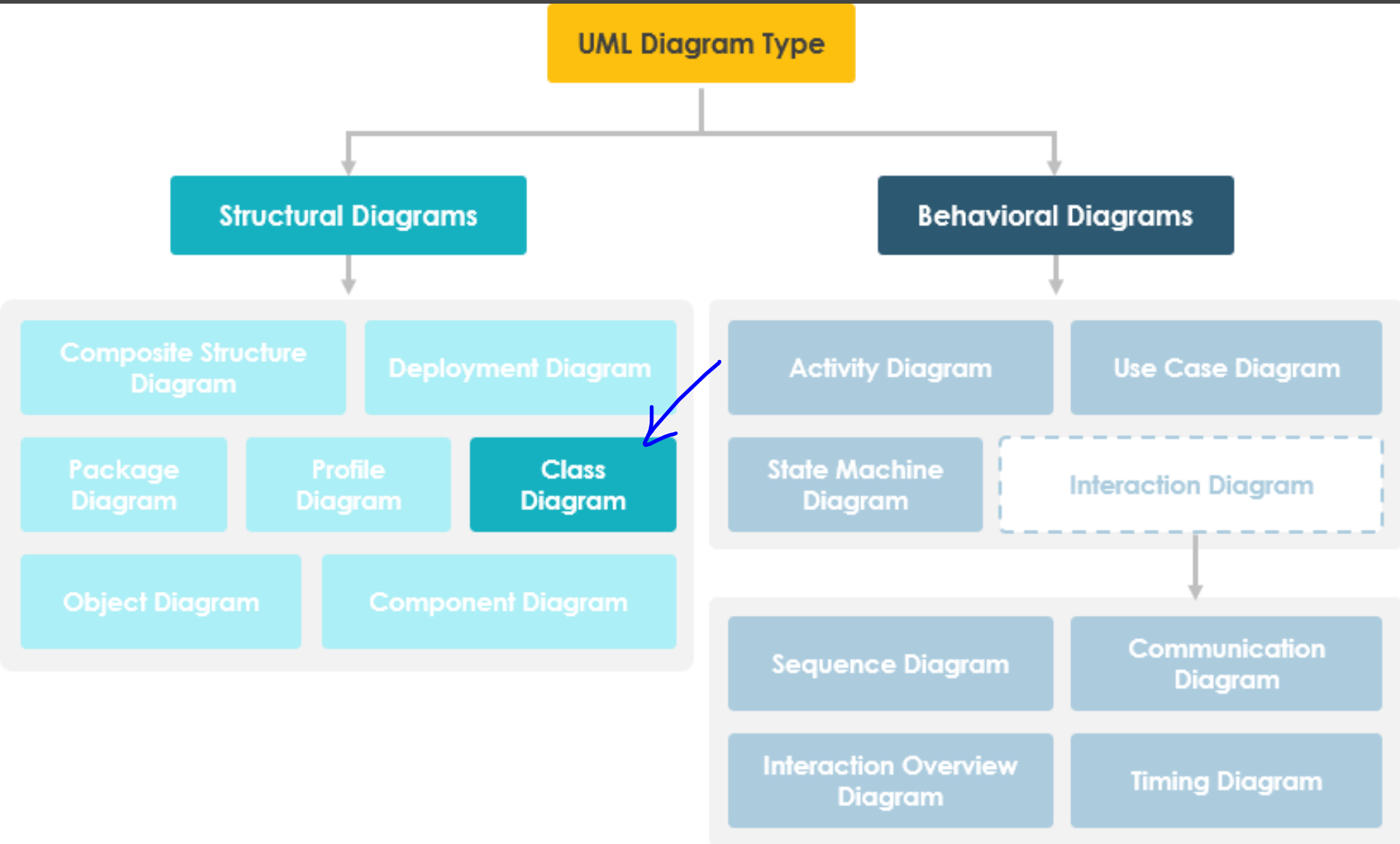


41951- ANÁLISE DE SISTEMAS

# Modelação com classes – cont. Diagrama de classes da UML

Ilídio Oliveira

v2024-02-27



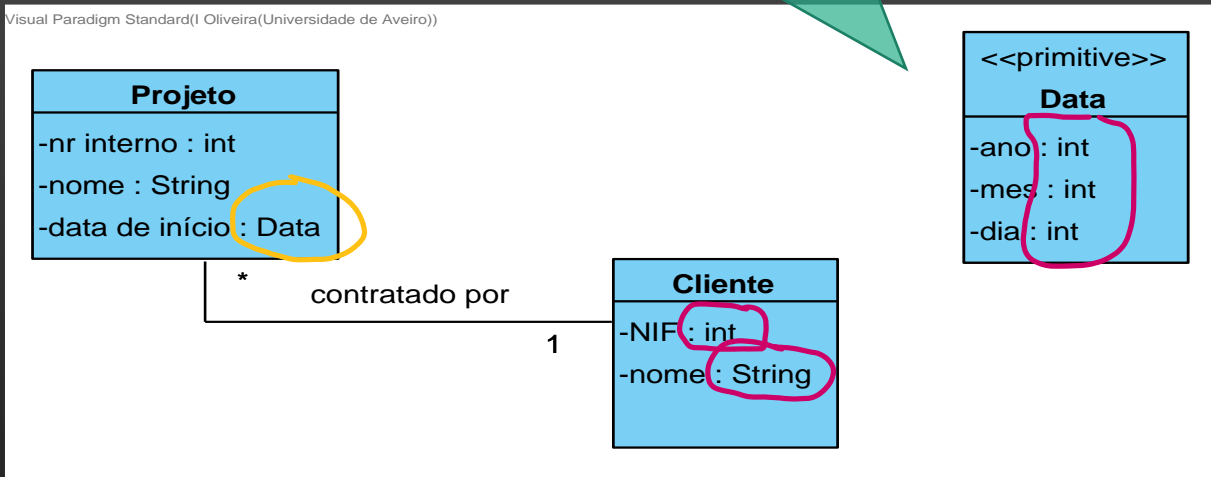
Um tipo de dados auxiliar. Não é um conceito do domínio.

## Tipos de dados

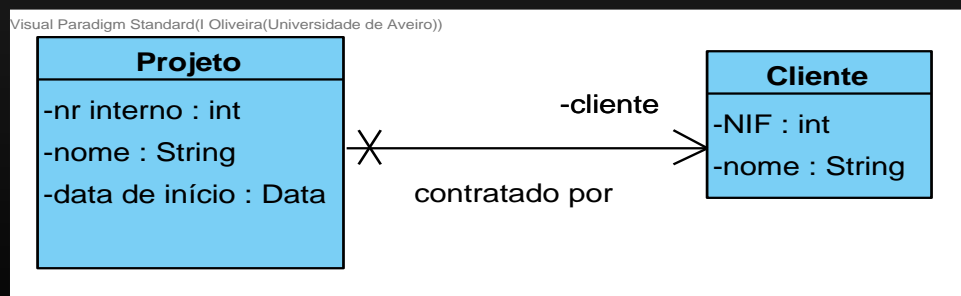
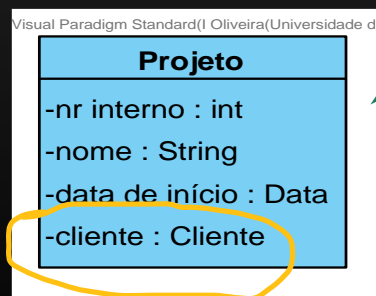
Um modelo completo especifica os tipos de dados dos atributos.

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)

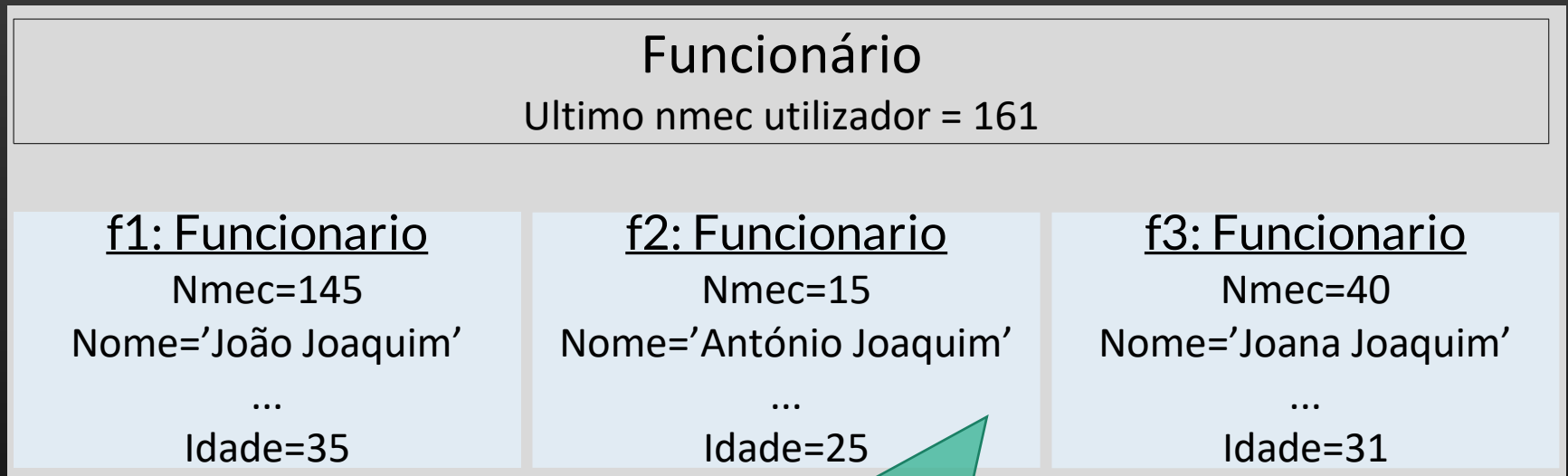


Um tipo de dados que é uma classe do modelo implica uma associação. Vamos preferir mostrar como associação.



# Atributos da classe (*static*)

Atributos de classe são partilhados por todas as instâncias



Os objetos f1, f2, f3 têm estado interno, independente entre si.

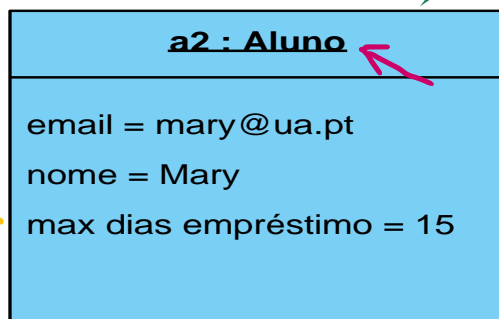
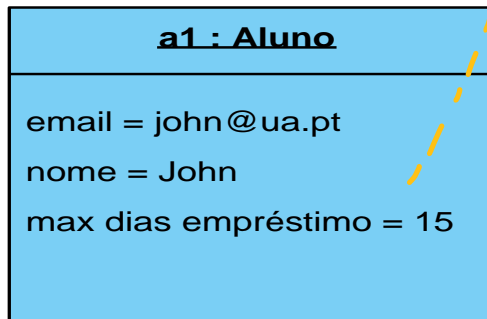
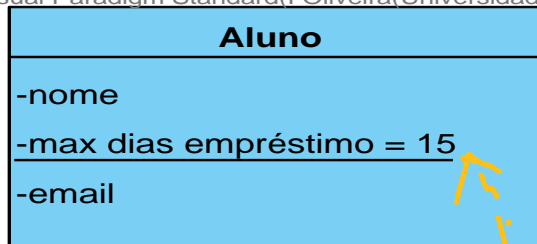
Podem existir atributos que afetem o conjunto dos objetos, i.e., atributos cujo valor é partilhado por todos: “Ultimo nmec utilizado”.

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)

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



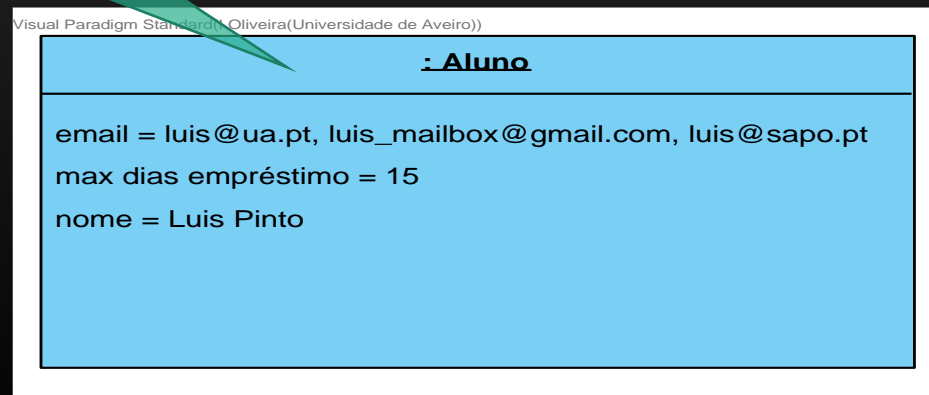
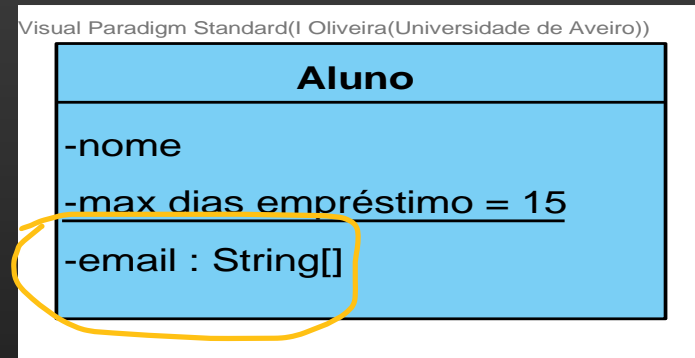
Alguns objetos que instanciam  
a classe. E.g.: o objeto  
chamado "a2" do tipo "Aluno"

instâncias

# Atributos multi-valor

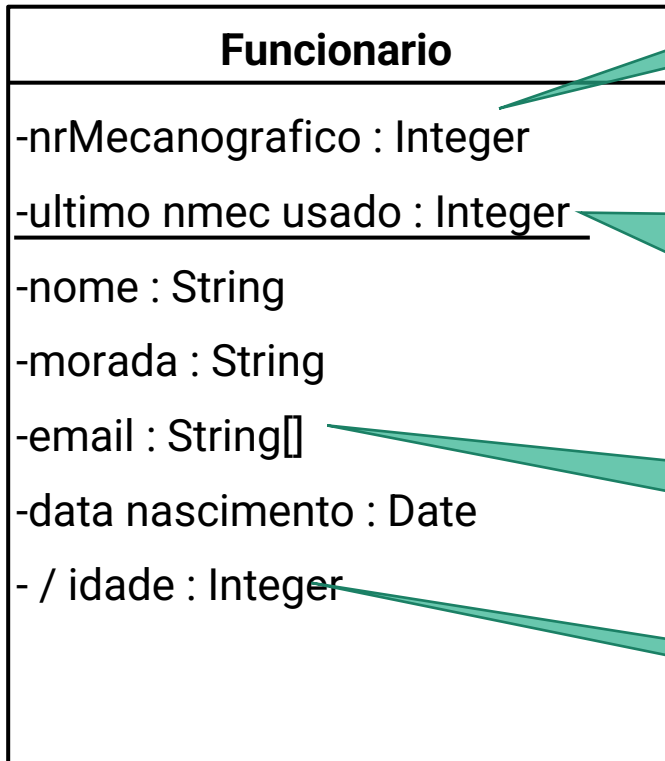
## Conceito de “vetor”

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

The screenshot displays the Visual Paradigm Standard interface. The main workspace shows a class diagram for a class named 'Veiculo'. The class is represented by a rectangular box with a blue header and a white body. The header contains the class name 'Veiculo' with a small 'a' icon to its left. The body is divided into two sections: the top section lists attributes, and the bottom section lists methods.

| Veiculo           |          |
|-------------------|----------|
| -matricula        | : string |
| -cor              |          |
| -lotação          |          |
| -velocidade atual |          |
| +travar()         |          |
| +acelerar()       |          |

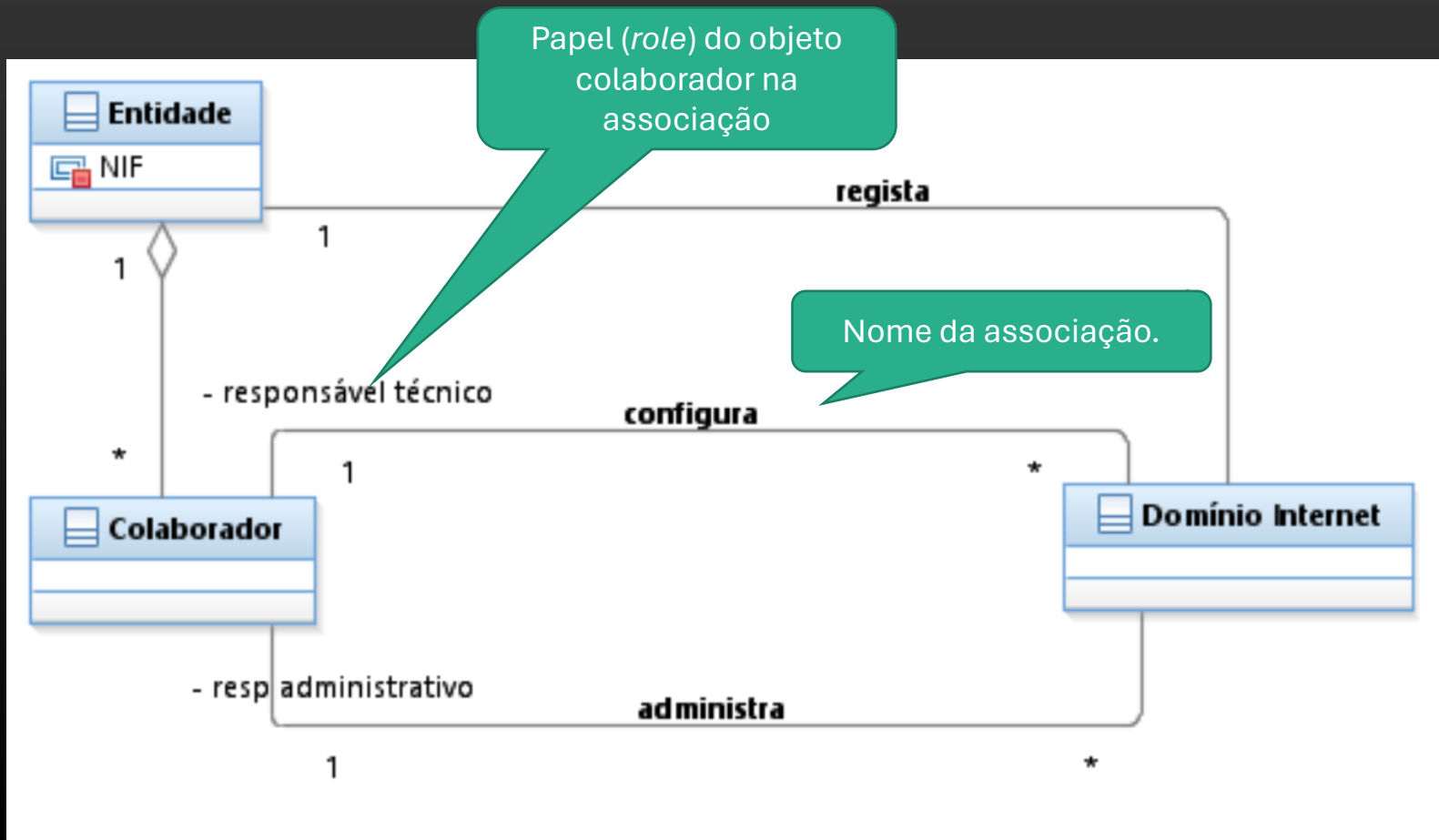
The interface also shows a 'Model Explorer' on the left side, listing various models and classes. The 'Class Repository' is visible, showing a tree structure of classes including 'Aluno', 'Child', 'Class', 'Dependente', 'Employee', 'Empresa', 'Ligeiro', 'Pasta', 'Pesado', 'Trabalhador', 'Turma', 'Veiculo', and 'Viatura'. The 'Veiculo' class is highlighted in the repository. The main workspace also shows a 'Cursor' and a '<default package>' label.



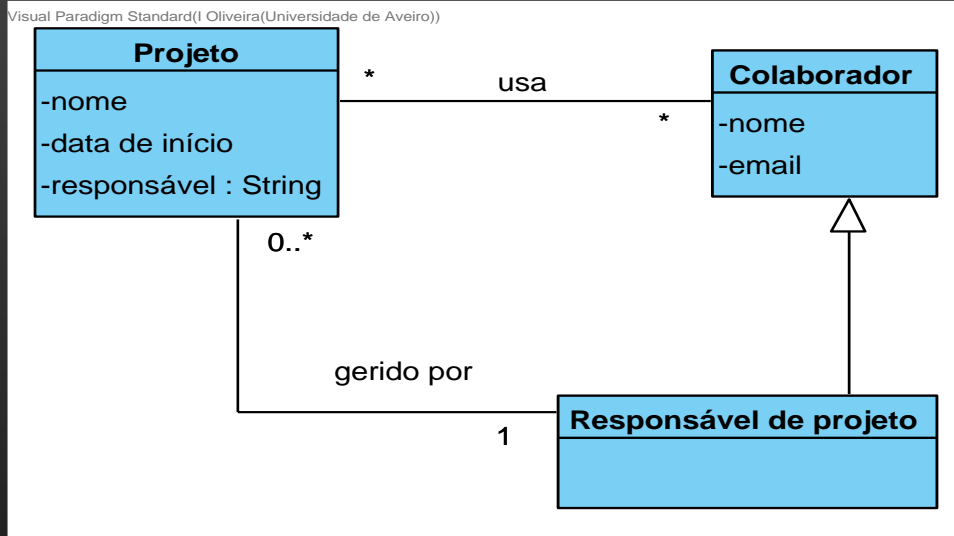
# Papéis ≠ nome da associação

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

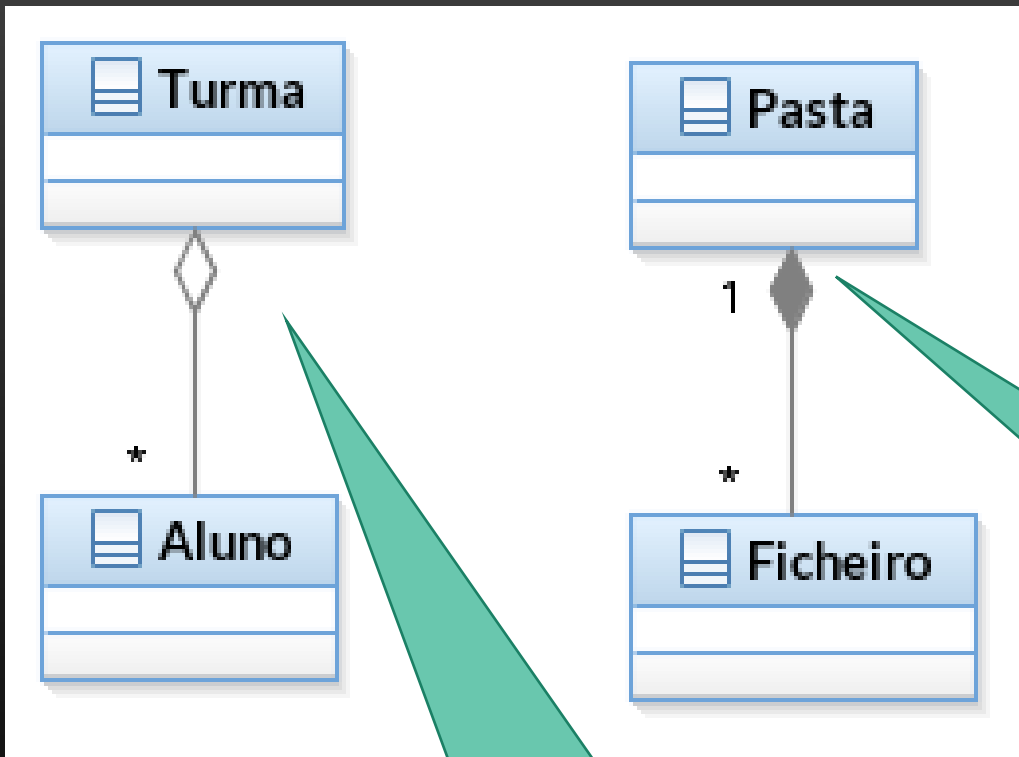
O **Papel** caracteriza a forma como instâncias de um tipo participam na associação.



# Opções de modelação: papéis



## Agregação vs composição



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

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

As instâncias de B são partilháveis

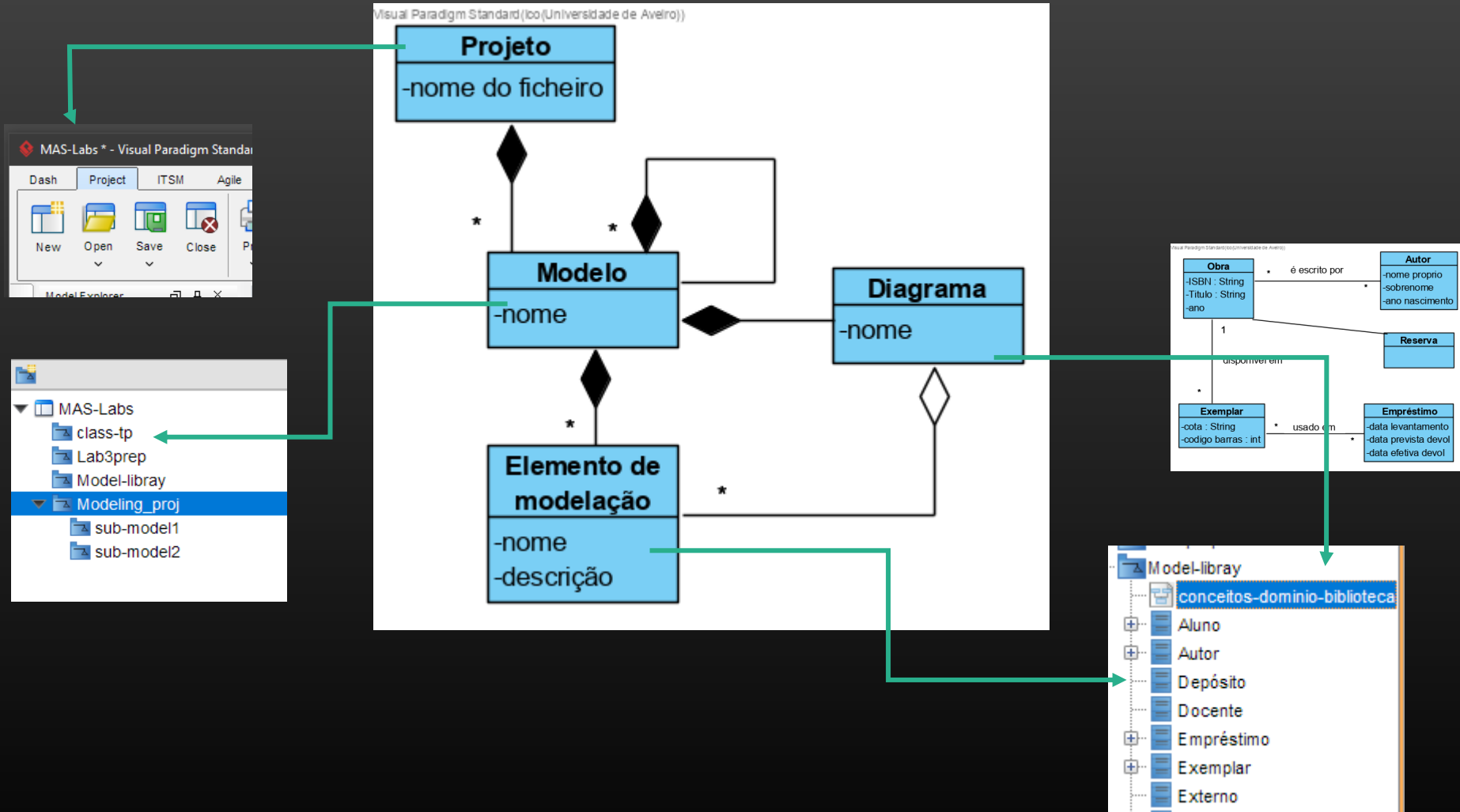
**Composição:** A comporta partes B de forma exclusiva

Os objetos B são partes constituintes de A

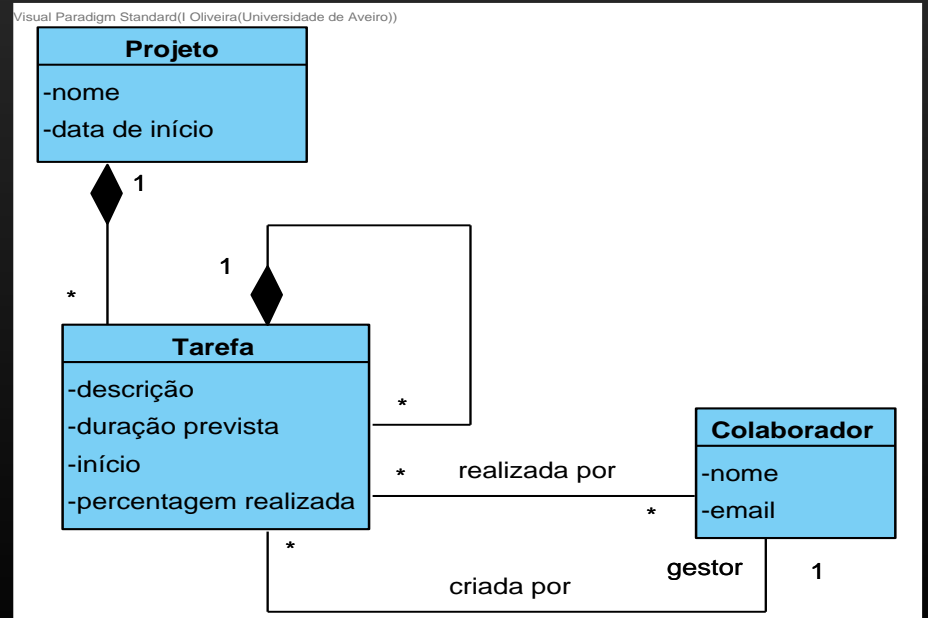
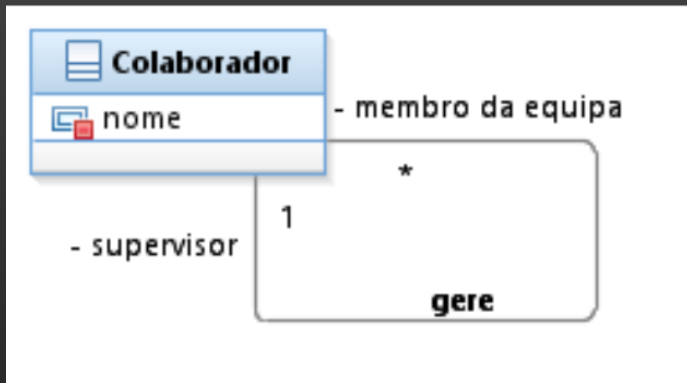
Instâncias de B não são partilháveis

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

# Exemplo: estrutura de um projeto de modelação no VP

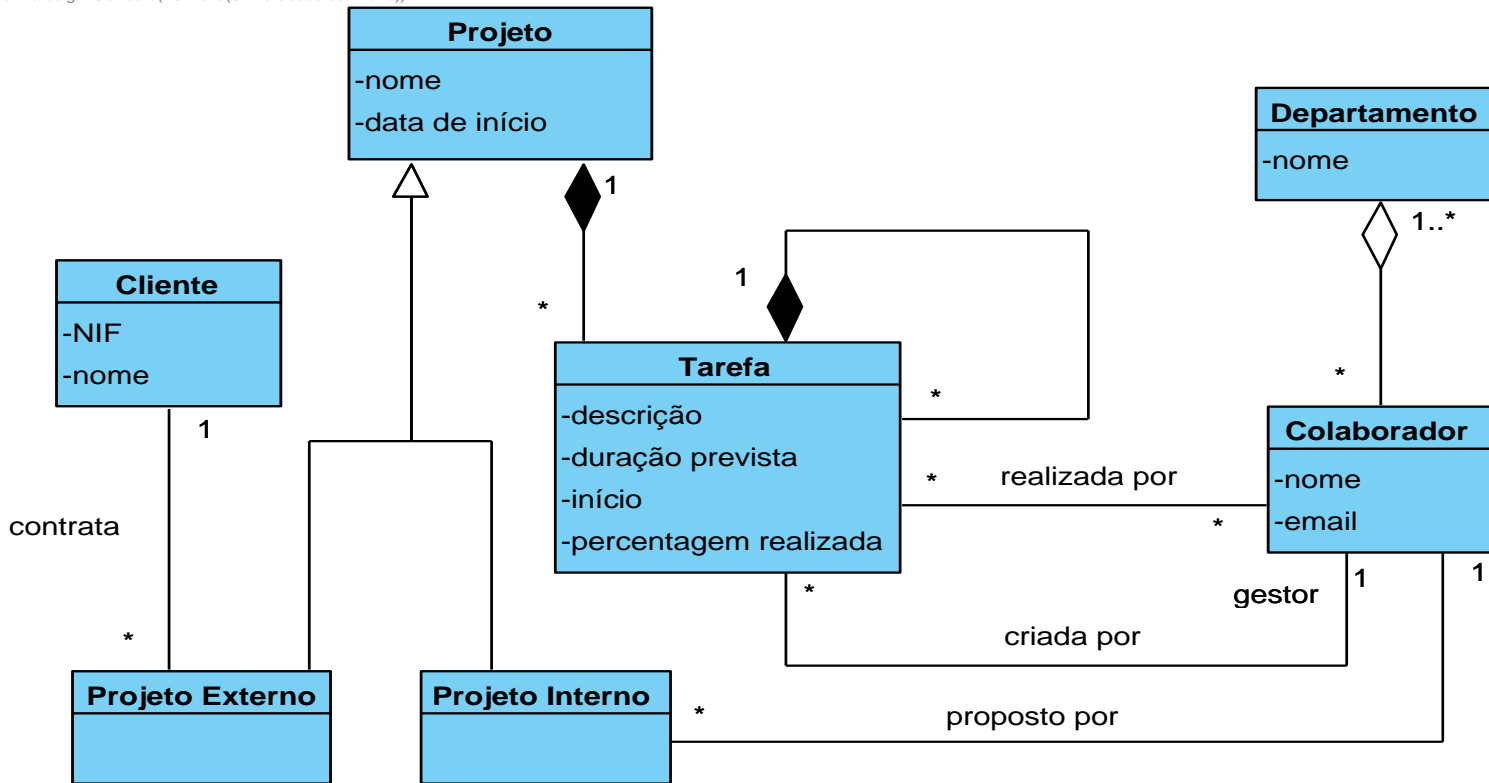


# Associações reflexivas relacionam instâncias da mesma classe



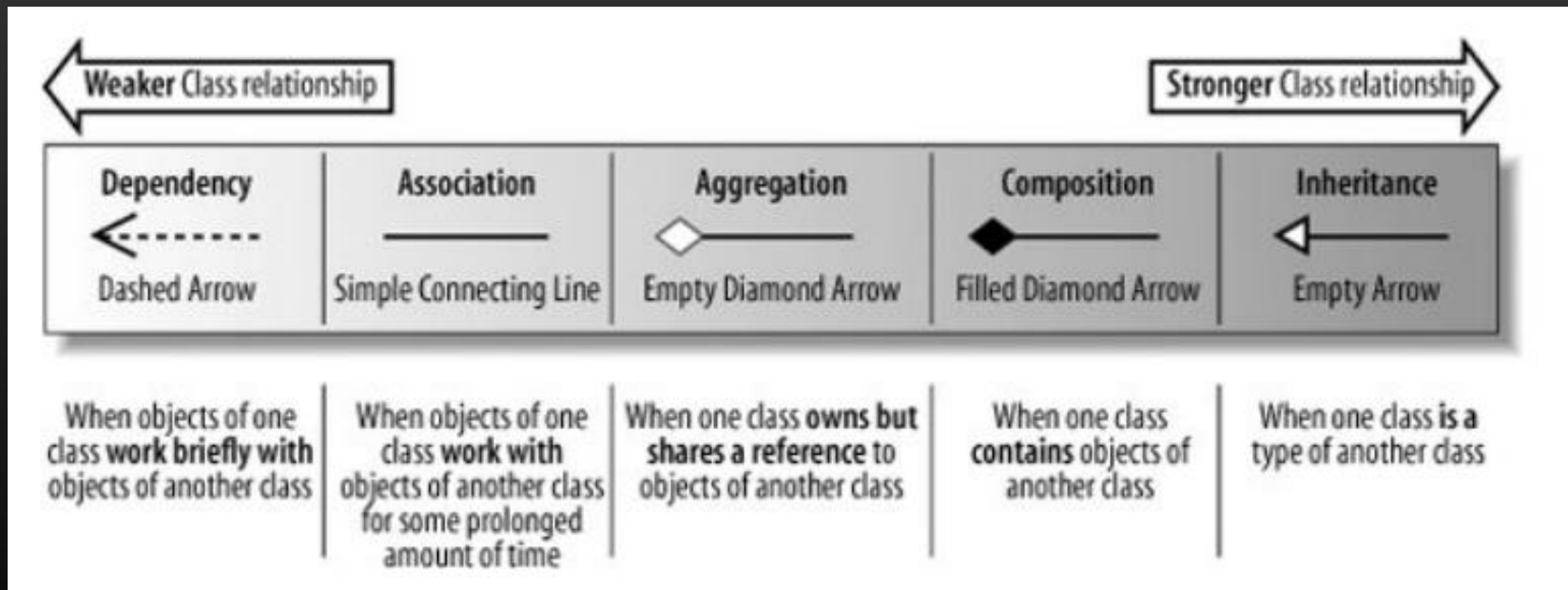
# “Força” da associação entre os conceitos (*coupling*)

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

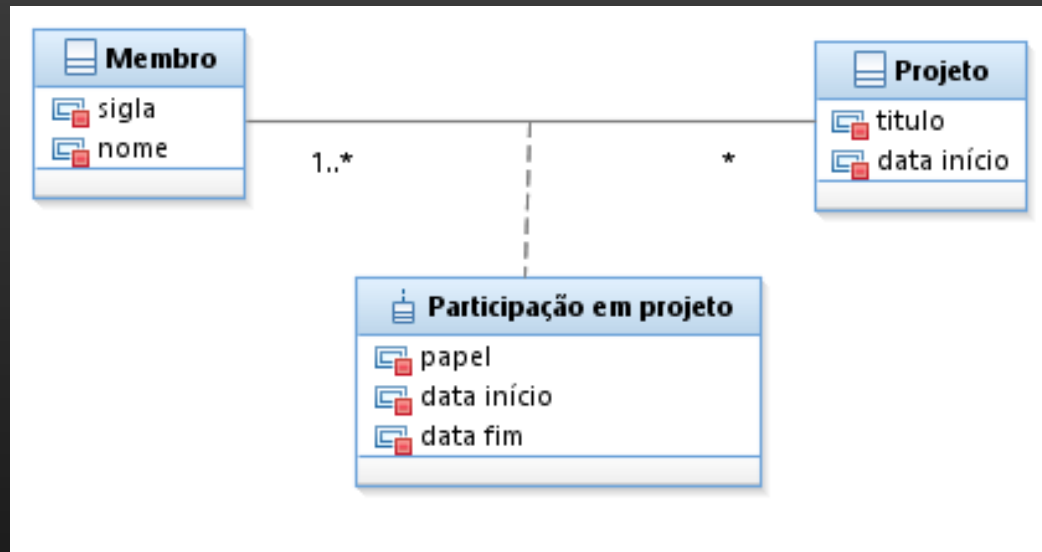


# “Força” da ligação estrutural entre classes

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



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



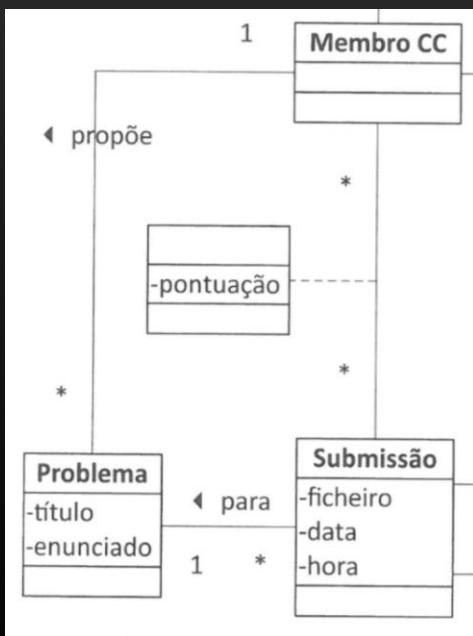
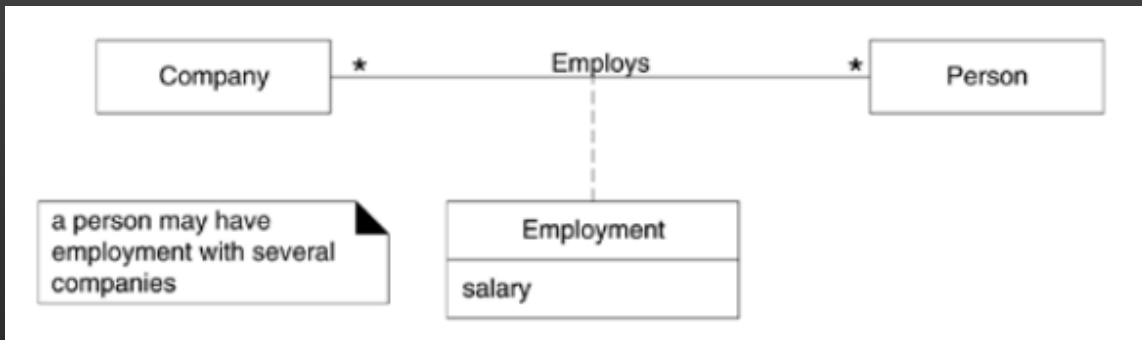


## Indicações para o uso de uma classe-associação no modelo do domínio

Um atributo está subordinado à ocorrência uma associação.

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

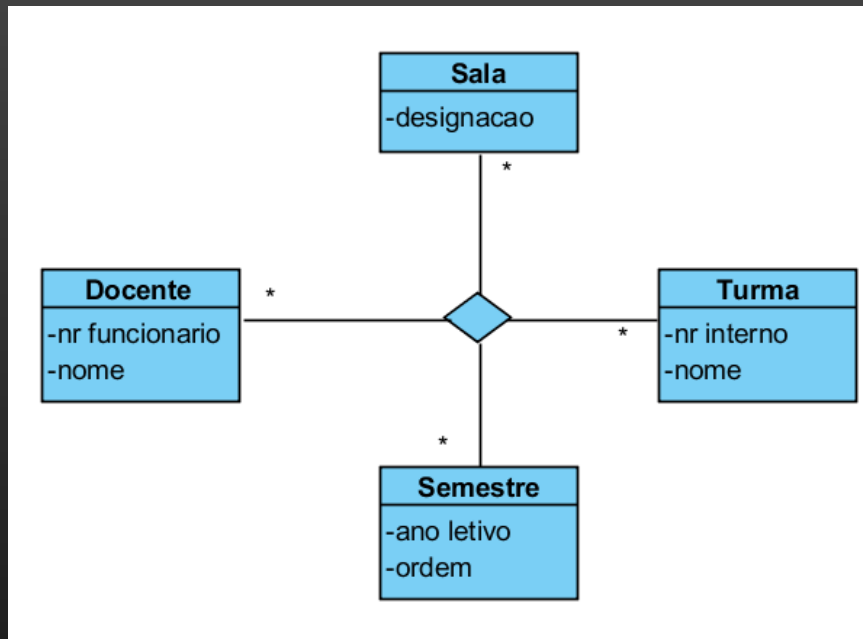
Há uma relação de N:M entre dois conceitos e é necessária informação para caracterizar a própria associação



## 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*

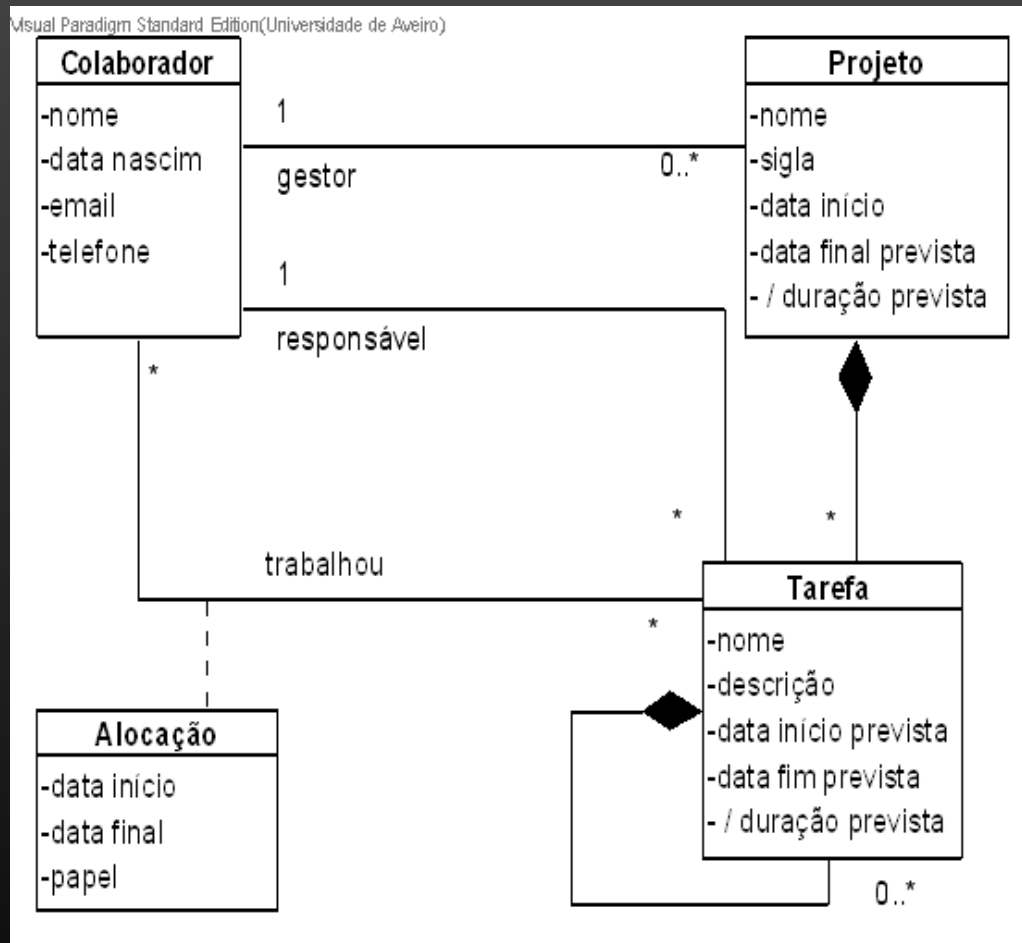
*Multiplicity*

*Derived attribute*

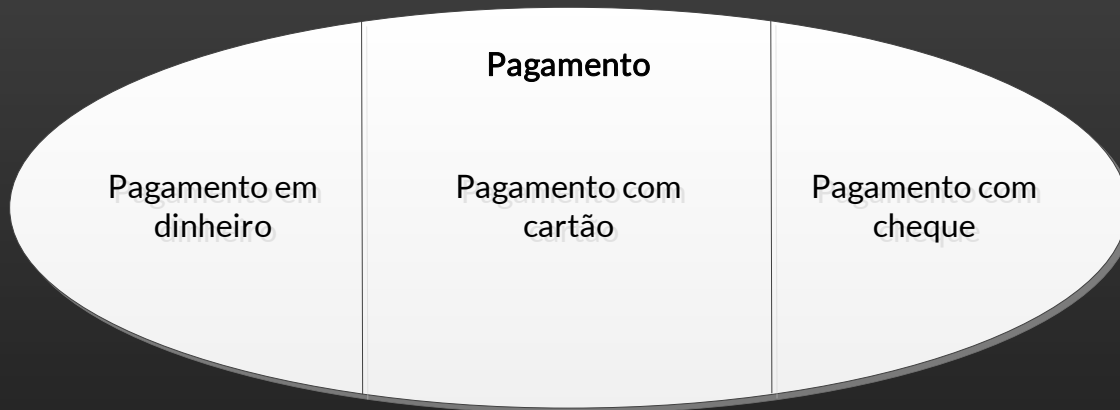
*Composition*

*Association class*

*Reflexive association*

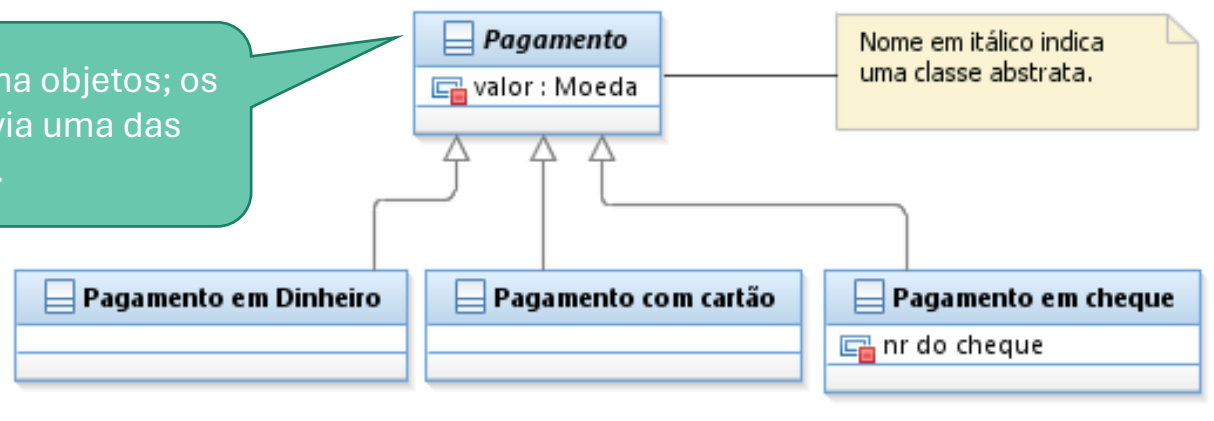


# Uma classe abstrata não é instanciada diretamente

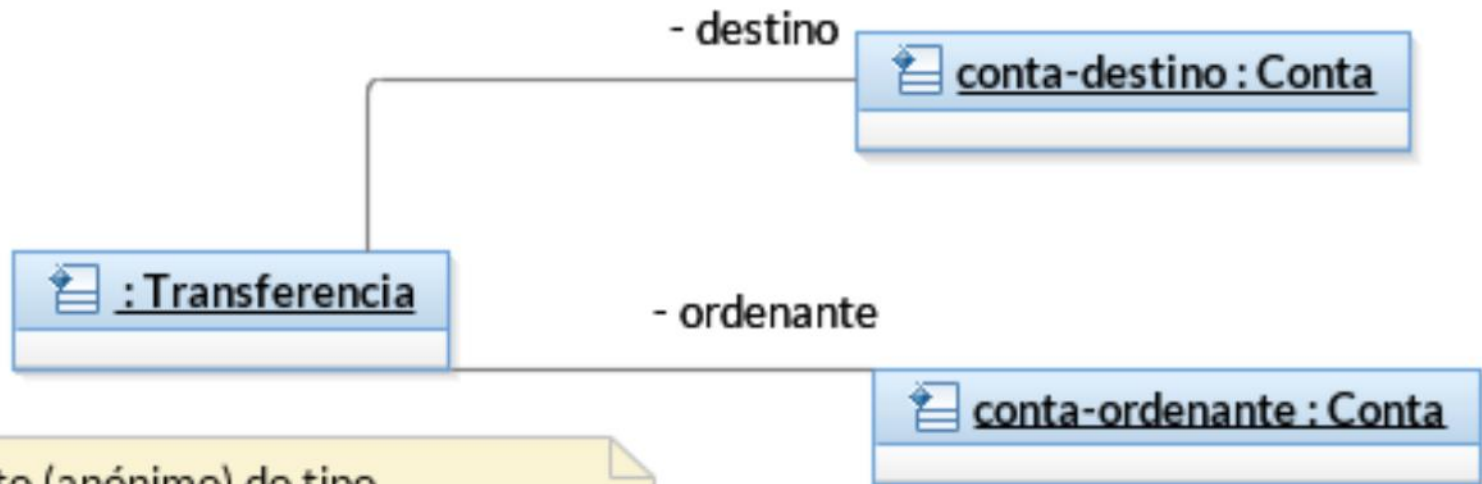


Um Pagamento tem de ser instanciado numa das formas indicadas pelas subclasses: em Dinheiro, Cartão ou Cheque. Pagamento deve ser uma classe abstrata.

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



# Diagramas de objetos



Um objeto (anônimo) do tipo Transferência usa dois objetos do tipo Conta, designados conta-destino e conta-ordenante.

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

## Exercício: domínio da gestão de projetos

- a) Os projetos têm um título descritivo, uma data de início e de fim.
- b) Um Projeto é implementado por um determinado Departamento.
- c) Cada Projeto define várias Tarefas que, por sua vez, podem estar organizadas em sub-tarefas.
- d) Um projeto tem uma equipa atribuída; as equipas têm vários colaboradores e um gestor.
- e) Os projetos externos são contratados por um cliente. Os internos, são pedidos por um Departamento.
- f) As tarefas têm data de início e duração estimada.