

47006- ANÁLISE E MODELAÇÃO DE SISTEMAS

Domain model and additional Class diagrams notation

Ilídio Oliveira

v2020/11/11, TP11

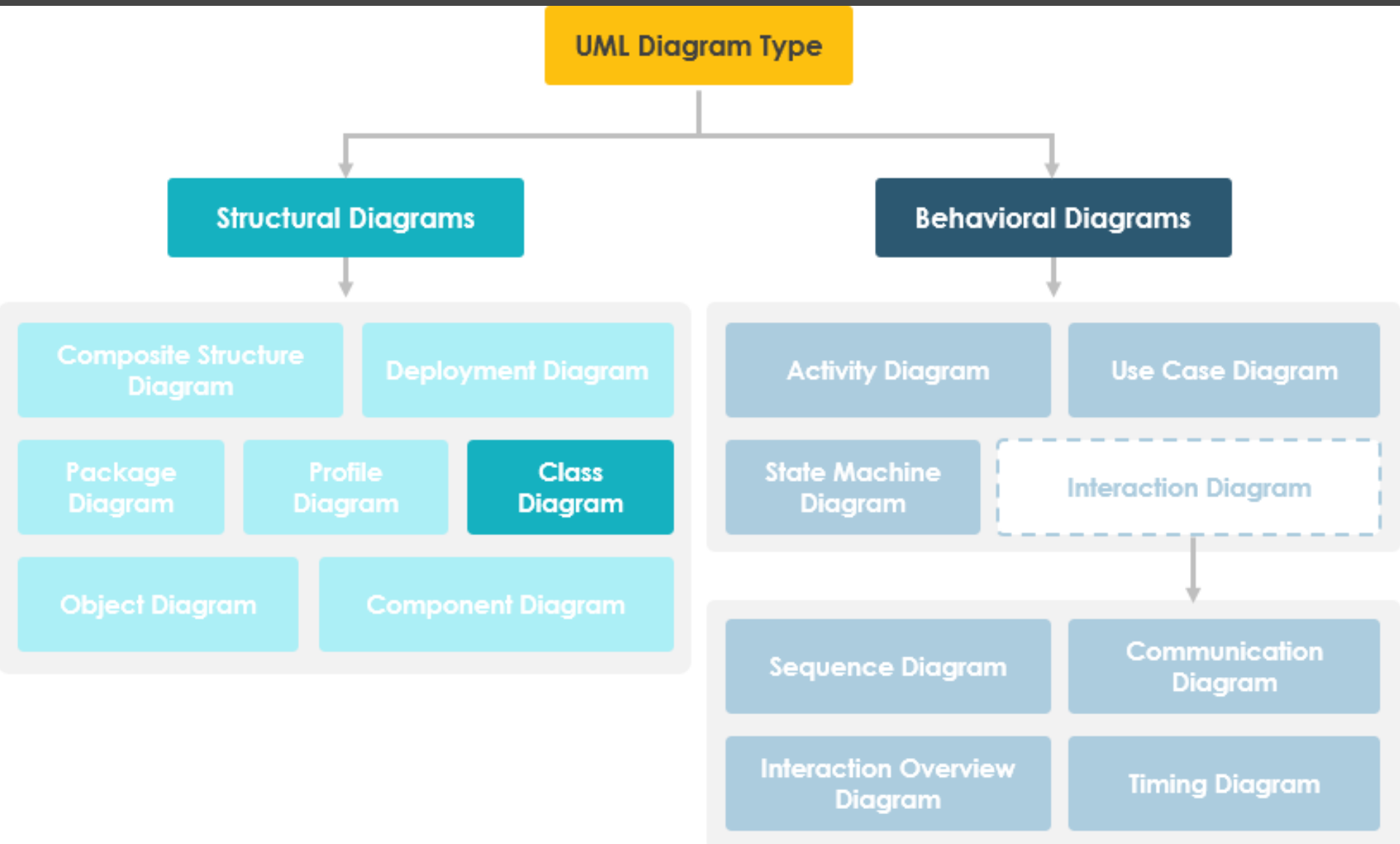
Learning objectives for this lecture

Draw a simple class diagram to capture the concepts of a problem domain.

Describe the types and roles of the different associations in the class diagram.

Critically review existing models to capture the domain concepts.

Spot implementation-specific constructs that may pollute the domain model.



Use of the UML classes diagram



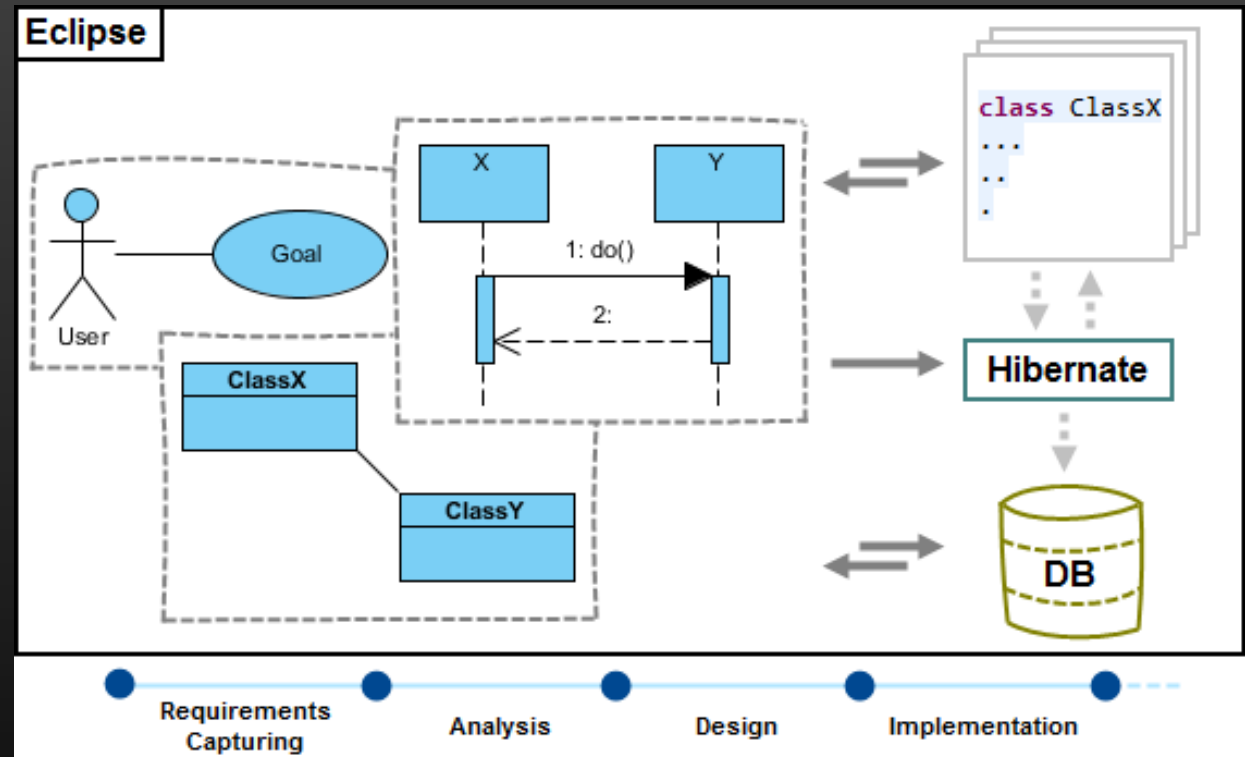
Analysis

→ Domain concepts

Design and implementation

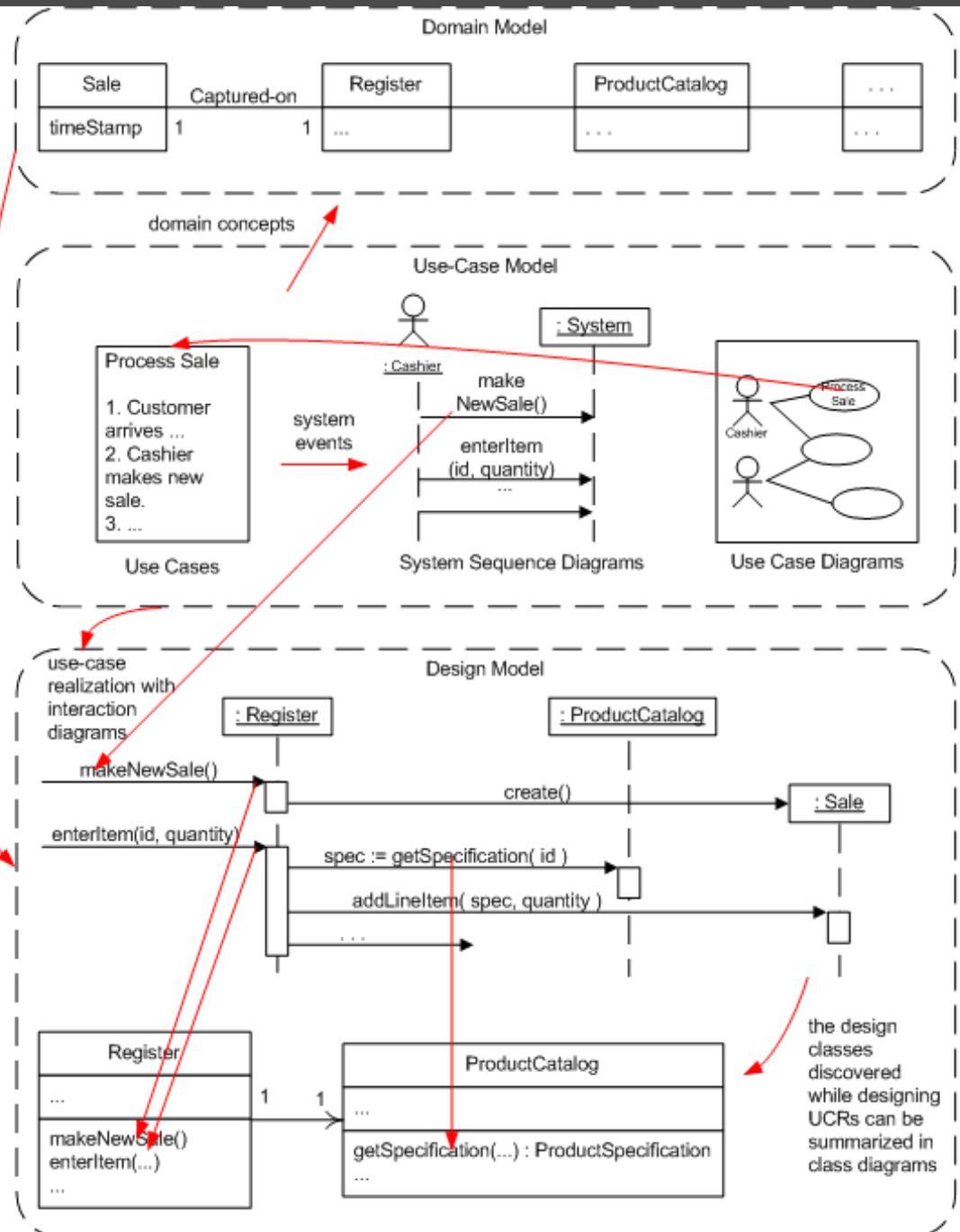
→ Programming entities (classes, enumerations,...)

Classes are used in the domain model and in the design model



Classes are used in the domain model and in the design model

conceptual classes in the domain inspire the names of some software classes in the design



O modelo do domínio não tem a implementação

Classes de análise: perspectiva concetual

Resultado da análise dos requisitos (**Analista**)

Neutro em relação à implementação

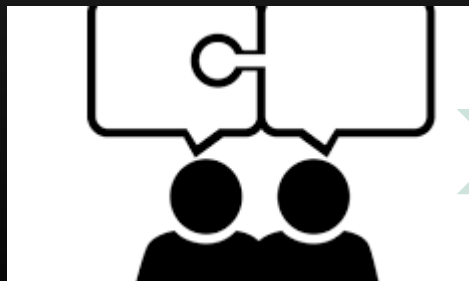
Não fornece diretamente o modelo de dados nem as classes de programação

Classes em Java: perspectiva de implementação

Resultado do desenho/implementação (**Programador**)

Escritas numa linguagem concreta (OO)

Assunto de P-3 (e, em parte, de MAS)



Análise

Implementação

Especificação



Domain model: focus on capturing the information and rules of the problem

O modelo do domínio é um *mapa dos conceitos* de uma área de aplicação

Mostra os conceitos de um problema

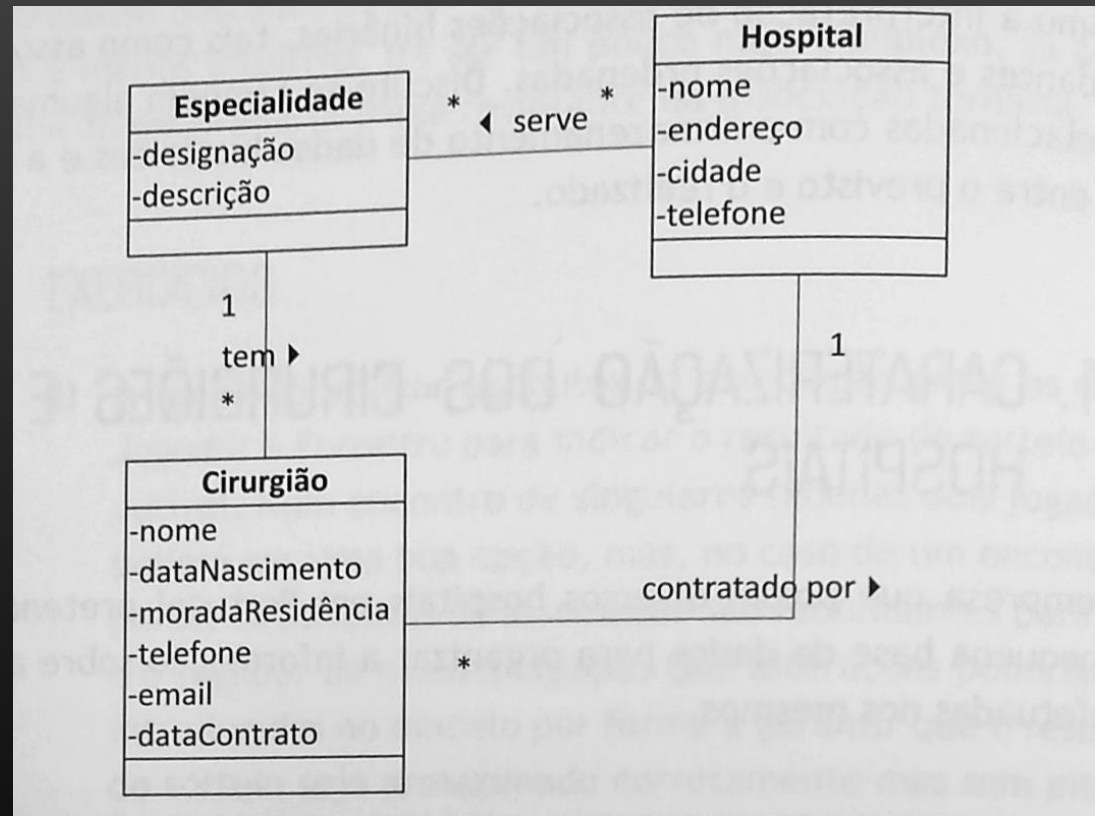
O vocabulário de uma área de aplicação

Não é software!

É um resultado da análise, na investigação do problema.

É visualizado num diagrama de classes

É uma **vista estrutural**.



Como encontrar as classes?

Procurar numa lista de situações comuns

→ categorias de classes

Análise documental

Explorar documentos/relatórios existentes na área do problema

Análise de nomes

→ explorar descrições do problema à procura dos substantivos

Situações comuns para encontrar classes candidatas

Conceptual Class Category	Examples
product or service related to a transaction or transaction line item <i>Guideline:</i> Transactions are <i>for</i> something (a product or service). Consider these next.	<i>Item</i> <i>Flight, Seat, Meal</i>
where is the transaction recorded? <i>Guideline:</i> Important.	<i>Register, Ledger</i> <i>FlightManifest</i>
roles of people or organizations related to the transaction; actors in the use case <i>Guideline:</i> We usually need to know about the parties involved in a transaction.	<i>Cashier, Customer, Store</i> <i>MonopolyPlayer Passenger, Airline</i>
place of transaction; place of service	<i>Store</i> <i>Airport, Plane, Seat</i>
noteworthy events, often with a time or place we need to remember	<i>Sale, Payment MonopolyGame Flight</i>
physical objects <i>Guideline:</i> This is especially relevant when creating device-control software, or simulations.	<i>Die</i>

[Larman 2004], Cap. 9.

Análise de nomes num texto

Nomes podem revelar conceitos (classes) ou atributos

Nem todos os nomes têm consequências diretas no modelo...

Os **projetos** têm uma **duração** variável, com **data de início** e de **fim** bem definidas.

Cada **projeto** é organizado em várias **tarefas** (ou **atividades**), com uma **duração** bem definida.

Um **projeto** vai ter uma **equipa** de funcionários atribuída.

Nem todos os funcionários participam em todas as tarefas, mas é importante saber que trabalhou numa tarefa.

O projeto tem um *Project Leader* atribuído (que é um funcionário)

Os **projetos** externos (ao contrário dos projetos internos), têm um **cliente**. O **NIF do contribuinte** é sempre necessário.

Domínio: clínica veterinária

- A clínica veterinária Faísca faz consultas a vários tipos de animais domésticos, e está em expansão na região de Aveiro.
- Em cada consulta só é visto um animal, por um veterinário da clínica.
- A clínica mantém a informação do dono do animal, incluindo nome, morada, telefone e NIF.
- Cada animal é tratado com elevados padrões de higiene, e tem uma ficha própria, em que se caracteriza o nome, género, espécie e data de nascimento.
- Os animais podem ser vacinados, de acordo com as vacinas indicadas para essa espécie.
- O Sr. Joaquim tem três cães, Tejo, Tamisa, Danúbio, que são pastores-alemães.

- A **clínica** veterinária Faísca faz **consultas** a vários tipos de **animais** domésticos, e está em expansão na **região** de Aveiro.
- Em cada **consulta** só é visto um **animal**, por um **veterinário** da **clínica**.
- A **clínica** mantém a informação do **dono** do **animal**, incluindo **nome**, **morada**, **telefone** e **NIF**.
- Cada **animal** é tratado com elevados padrões de higiene, e tem uma **ficha** própria, em que se caracteriza o **nome**, **género**, **espécie** e **data de nascimento**.
- Os **animais** podem **ser** **vacinados**, de acordo com as **vacinas** indicadas para essa **espécie**.
- O Sr. Joaquim tem três cães, Tejo, Tamisa, Danúbio, que são pastores-alemães.

Dá origem a: **Classe** **Atributo** **A ponderar**

Classe ou atributo?

É natural pensar num dado atributo como um “tipo primitivo” ou como um conceito (tipo de coisas)?

“tipos primitivos”: números, datas, uma String,...

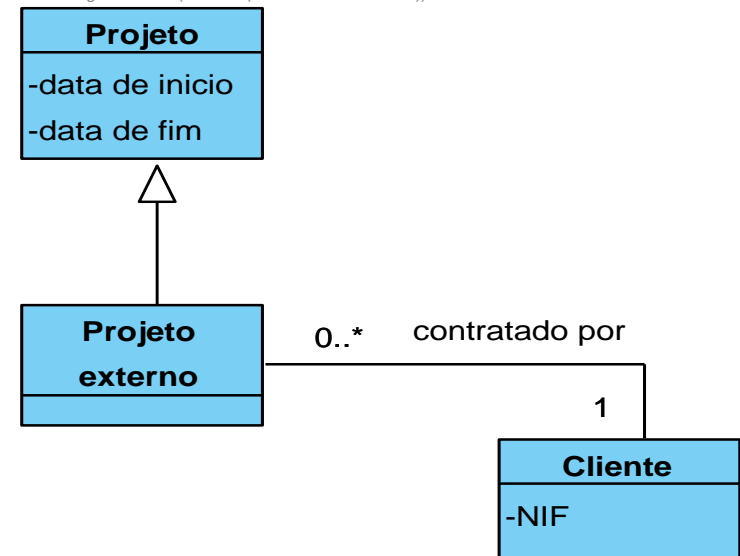
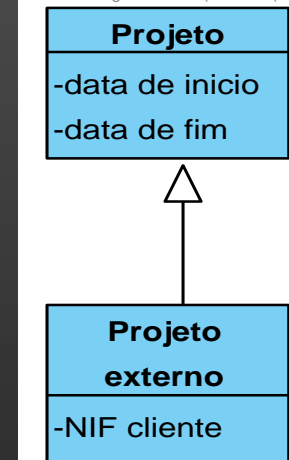
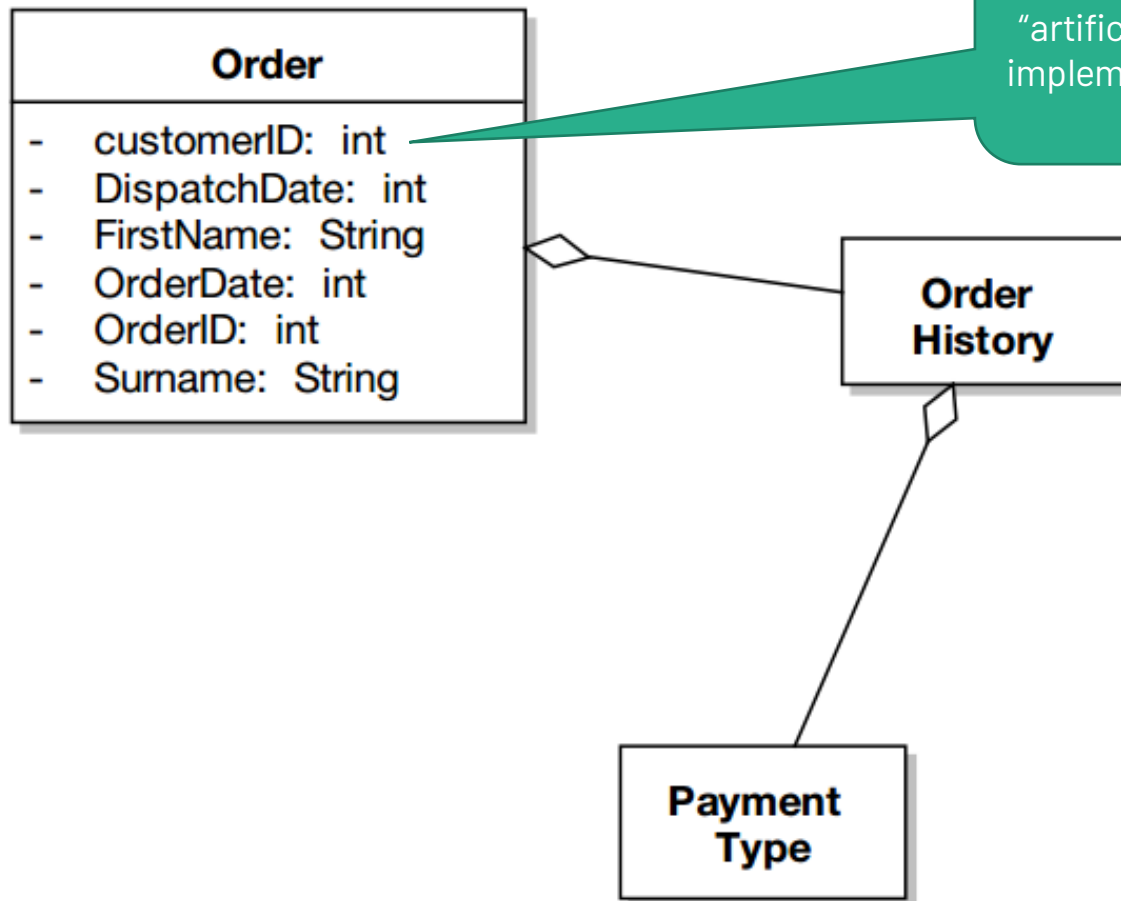


Figure 2-9 shows a domain model diagram with attributes on the Order class. What database-related problem does the diagram suggest?



Nos conceitos do modelo do domínio, não se devem inserir atributos "artificiais", ainda que necessários na implementação (e.g.: chave de ligação a outras "tabelas")

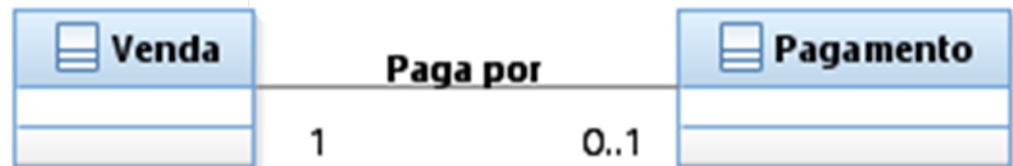
Só vamos incluir "chaves" se esses atributos forem naturais no domínio do problema. (e.g.: Nr Mecanográfico no aluno)

Nomear as associações com Classe → Expressão verbal → Classe

A sequência deve ser legível e revelar o significado da associação



O nome "tem" é pouco revelador da razão de ser da associação.



Boas práticas

Uma classe representa um tipo de coisas. O nome é no singular.

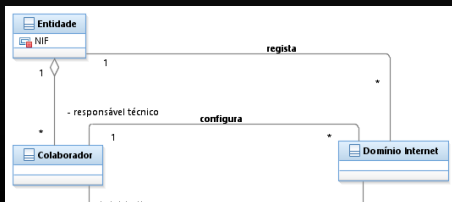
O nome da classe é um substantivo (representa um conceito, não uma ação).



Roles \neq name of the association

The name should clarify the association semantics

The role characterizes the way instances of a type participate in the association.



Clas diagrams: more syntax

Characterization of attributes

Visibility

Data type

Static (or class) attribute

Multi-value/array

Derived attribute

Visual Paradigm (Universidade de Aveiro)

Funcionario

-nrMecanografico : Integer

-ultimo nmec usado : Integer

-nome : String

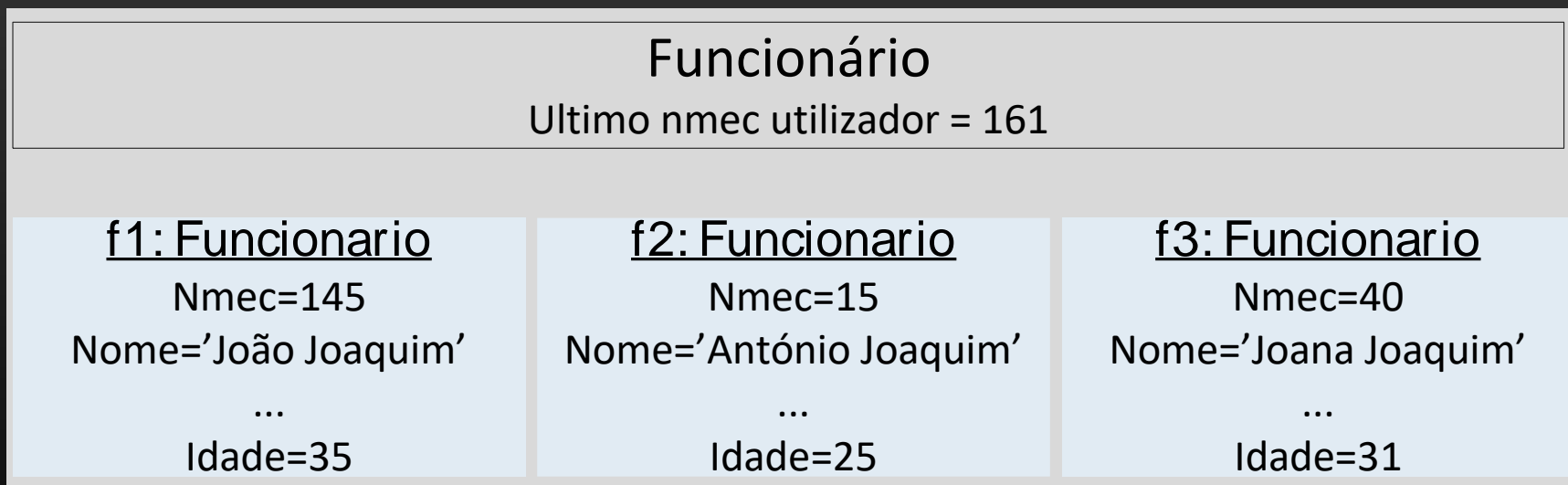
-morada : String

-email : String[]

-data nascimento : Date

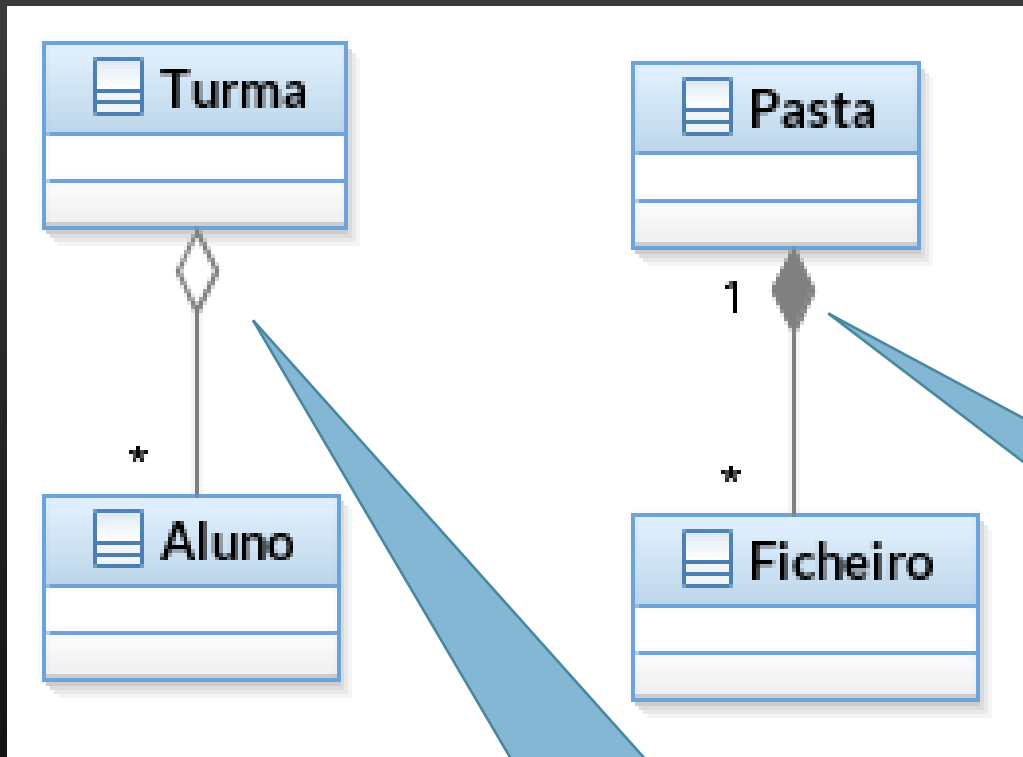
- / idade : Integer

Class attributes are shared by all instances



Aggregation vs. composition

Figure Shared or exclusive instances



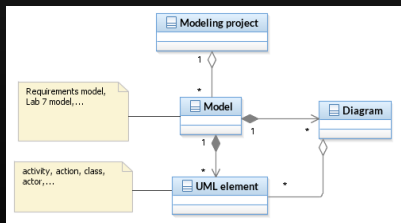
Aggregation: A owns parts B in a non-exclusive way

→ Shareable B Instances

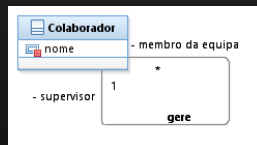
Composition: The B are constituent parts of A

→ Non-shareable B instances

Structure of a modeling project in a UML tool

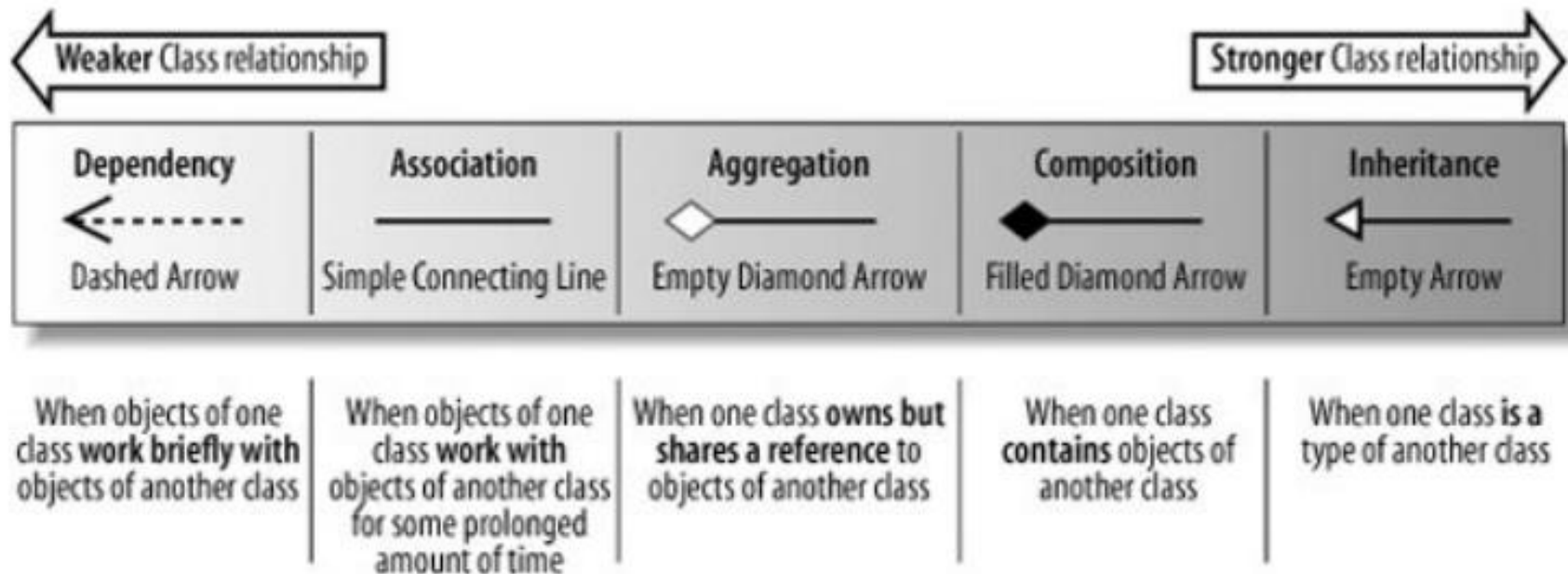


Reflexive associations relate instances of the same class



Structural bond strength between classes

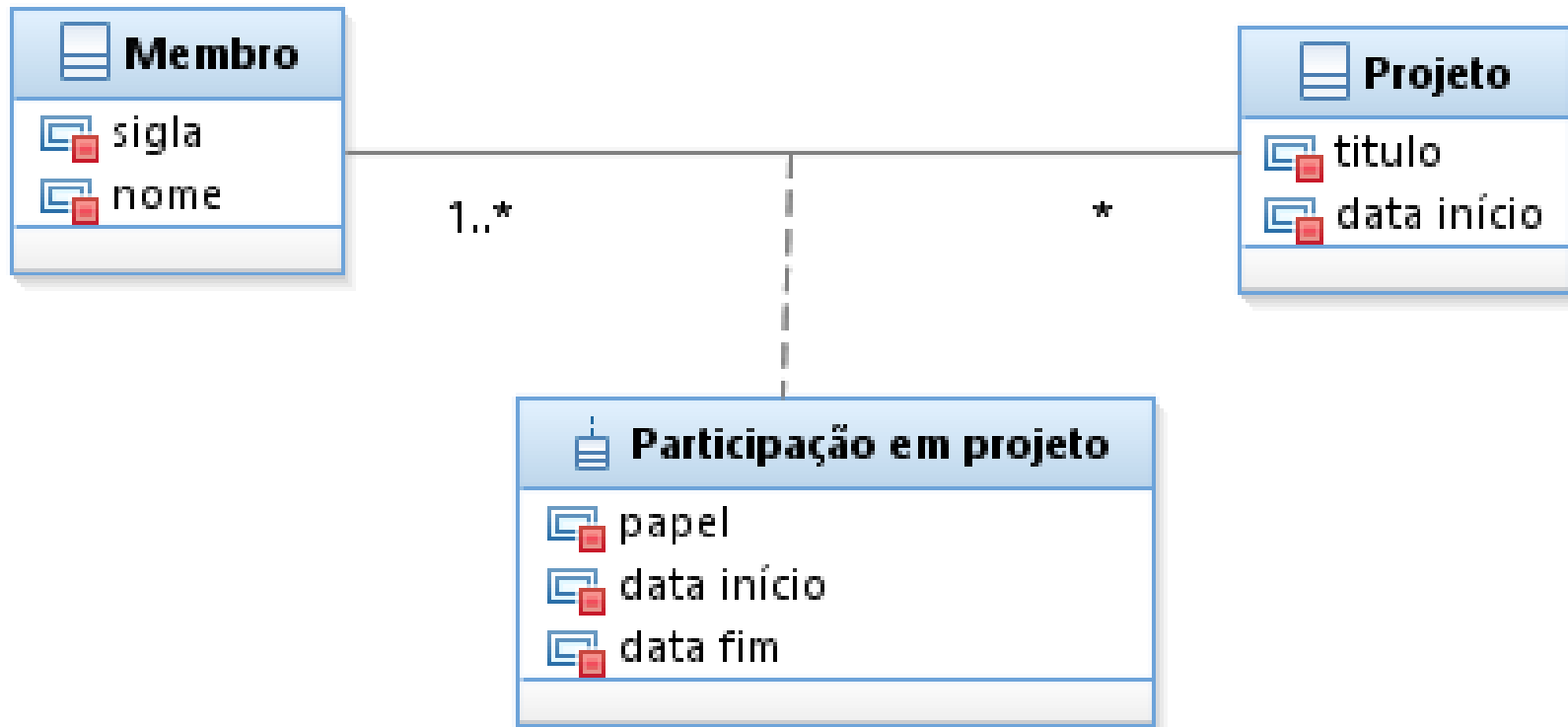
Five different types of relationship between classes, with different binding levels



Modeling problema....

A given Task (in a project) can be contributed by several employees, during different periods. Anna worked in the PayPal integration task , as a quality assurance engineer, for the first half of it.

Association-classes captures information describing the relationship



Indications for the use of a class-association in the domain model

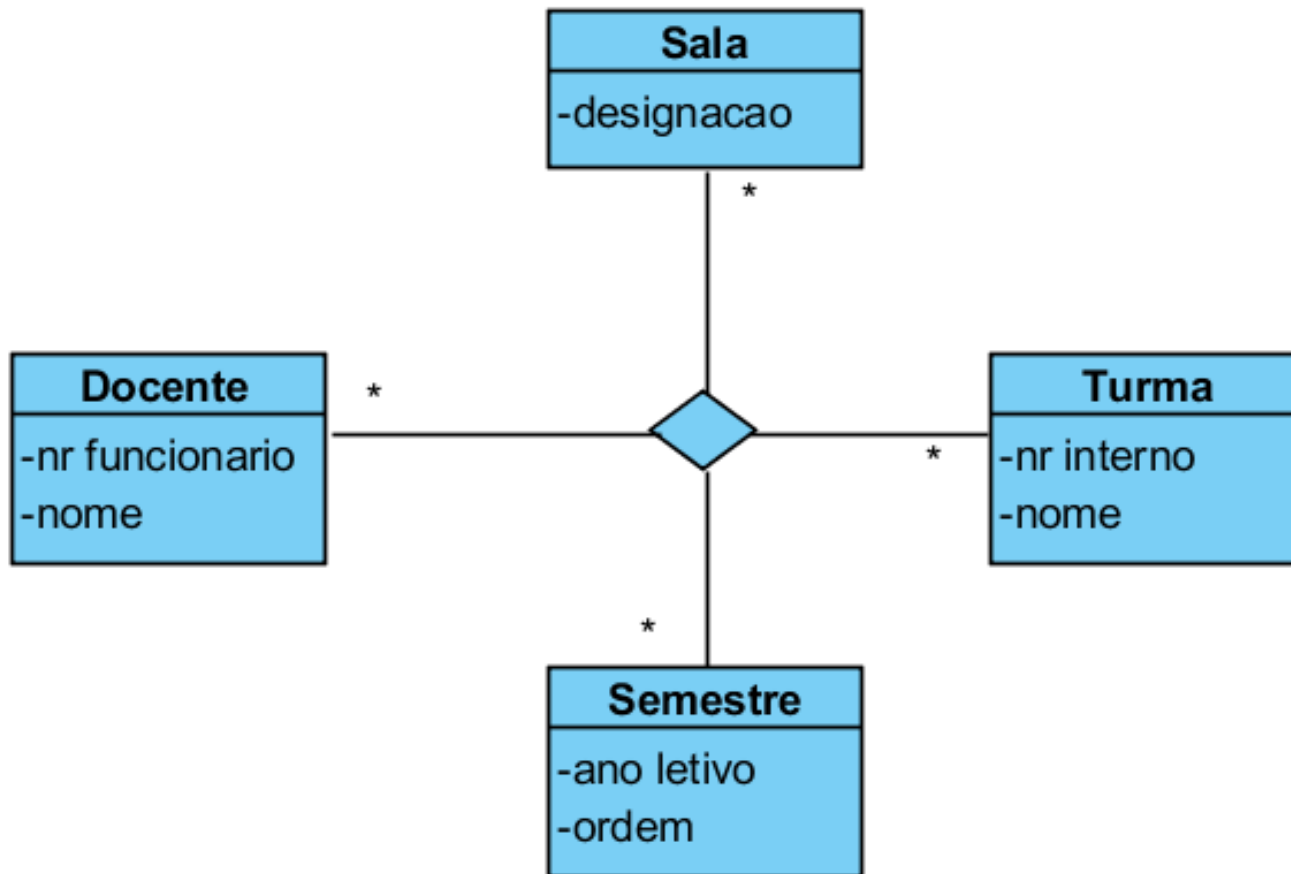
An attribute is related to (the occurrence of) an association.

Instances of the association-class have a lifetime dependent on the association

There is a N:M relationship between two concepts and information that characterizes the association itself



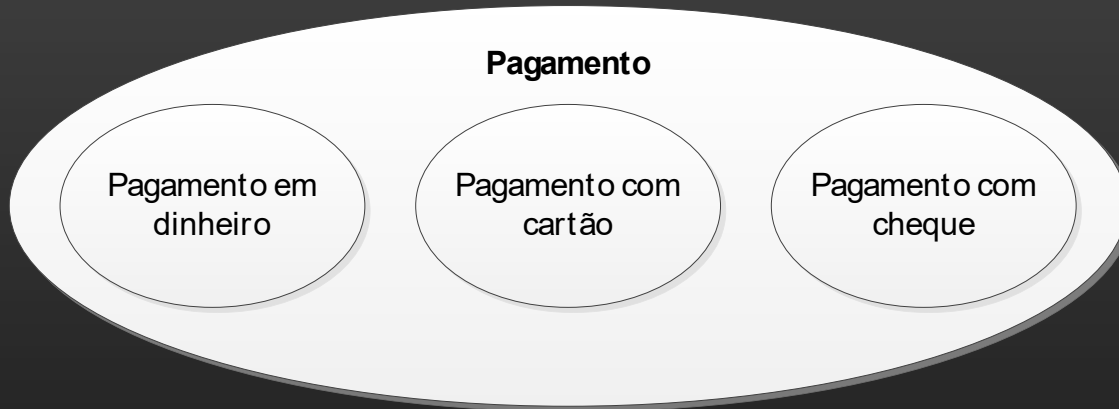
N-ary Associations



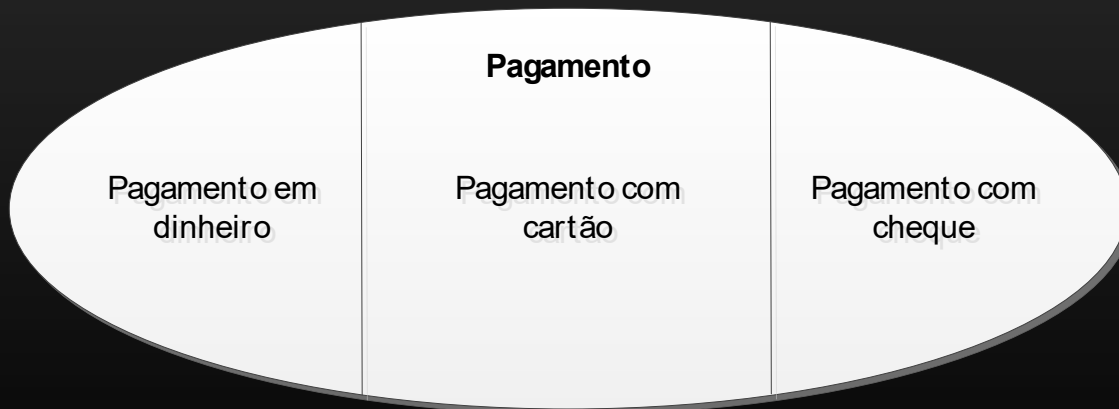
Modeling problema....

In our business, every payment (from sales) has to be one out of 3 possibilities: in cash, with card and in cheque. In the case of a cheque, it is important to keep the cheque number for follow-up.

Abstract Classes

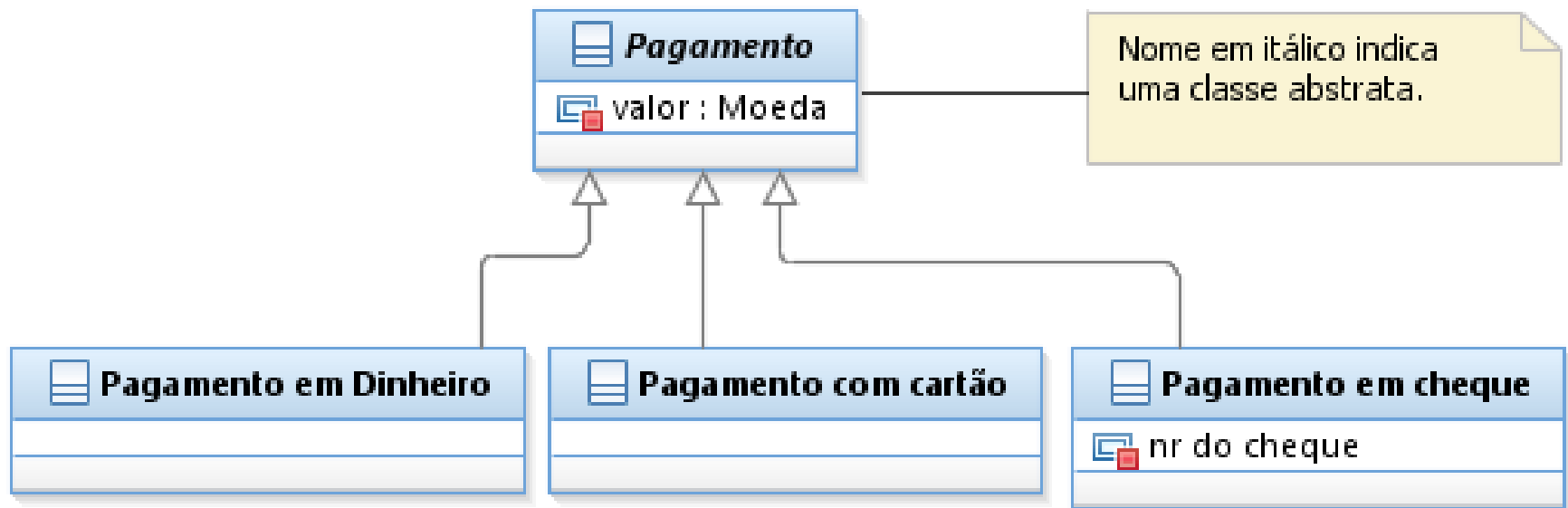


Pode haver instâncias de Pagamento que não sejam em Dinheiro, Cartão ou Cheque? Se sim, Pagamento não deve ser uma classe abstrata.

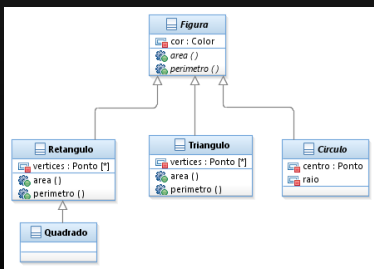


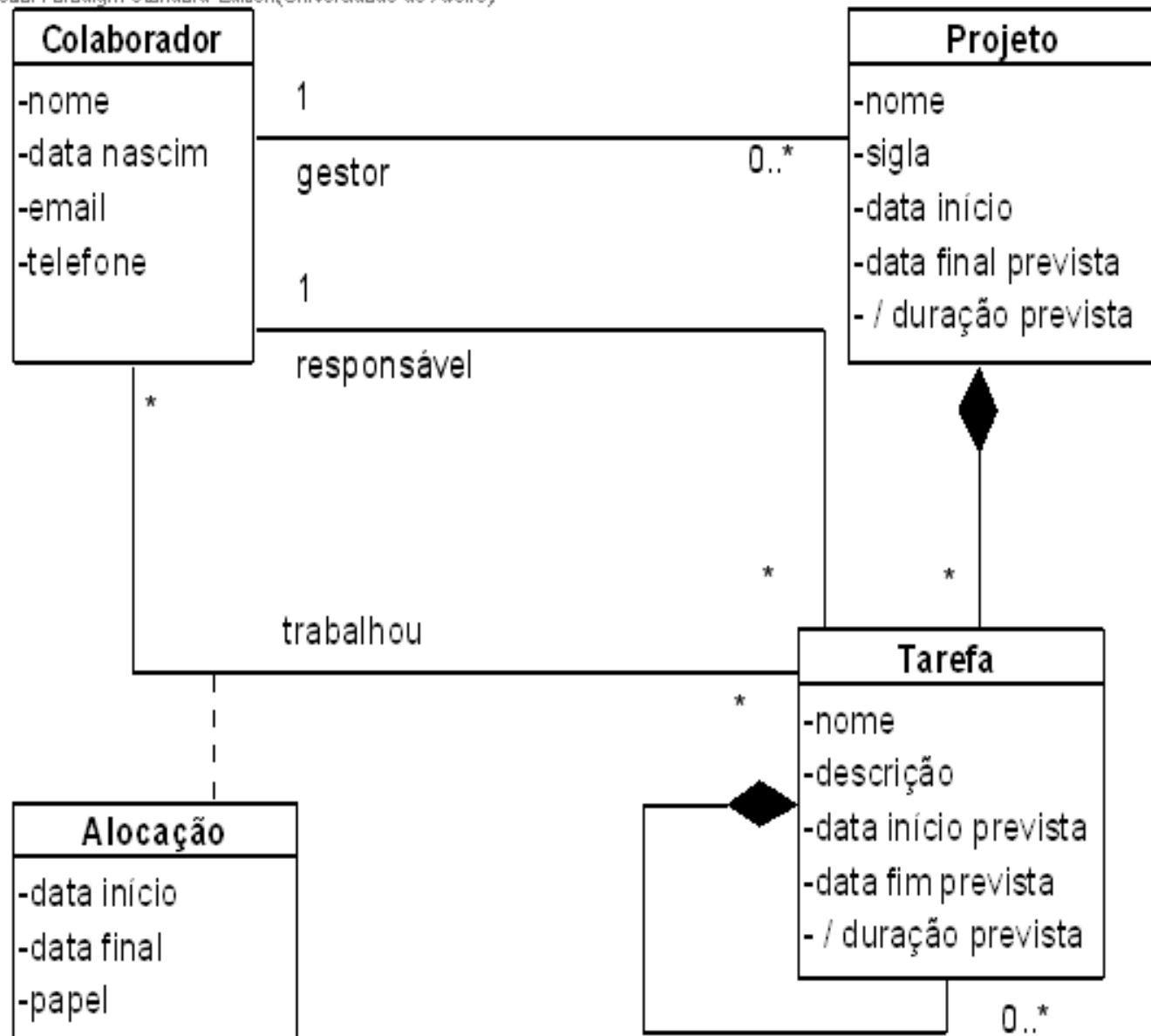
Um Pagamento tem de ser instanciado numa das formas indicadas pelas subclasses: em Dinheiro, Cartão ou Cheque. Pagamento deve ser uma classe abstrata.

An abstract class is not instantiated directly



Abstract Classes facilitate partial implementations (common parts in the superclass)





Perguntas de exame

16.

Considere o Diagrama 3:

- a) Um Projeto pode ou não ter um gestor.
- b) Um Projeto pode agregar sub-projetos.
- c) Uma Tarefa pode agregar sub-tarefas.
- d) Uma Tarefa pode ser realizada em diferentes projetos.
- e) Cada Colaborador é responsável por uma Tarefa.

17.

O Diagrama 3 utiliza uma classe de associação, para caracterizar a alocação a projetos.

- a) A classe de associação é uma classe cujas características descrevem uma associação, e não um objeto "normal".
- b) A classe de associação facilita a visualização da interdependência entre duas classes.
- c) A classe de associação pode ser suprimida, desde que se mova os respetivos atributos para uma das classes associadas.
- d) A classe de associação deve ser usada sempre que há uma multiplicidade de muitos para muitos (entre as classes base associadas).
- e) Um bom modelo deve evitar a utilização de classes de associação.

18.

Relativamente ao Diagrama 3, Tarefa e Projeto indicam durações previstas.

- a) Há um erro de notação no atributo associado à duração prevista.
- b) A duração prevista pode ser determinada à custa de outros atributos, não deve ser representada na classe.
- c) A duração prevista pode ser determinada à custa de outros atributos, é um atributo derivado.
- d) A duração prevista deveria ser definida apenas na Tarefa.
- e) A duração prevista é obtida por um método e é errado apresentar como um atributo.

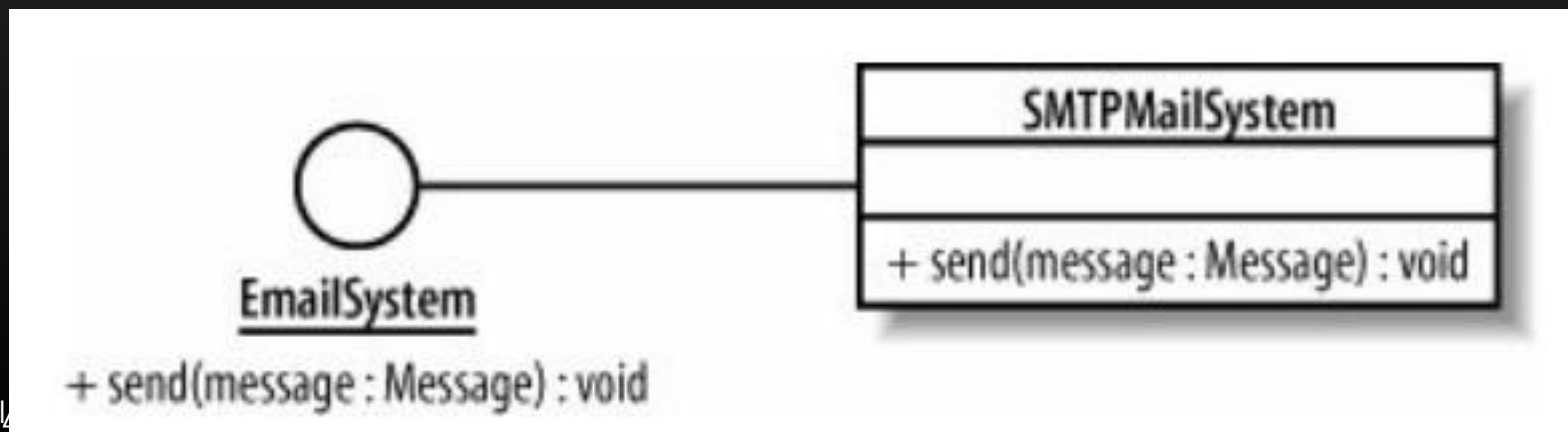
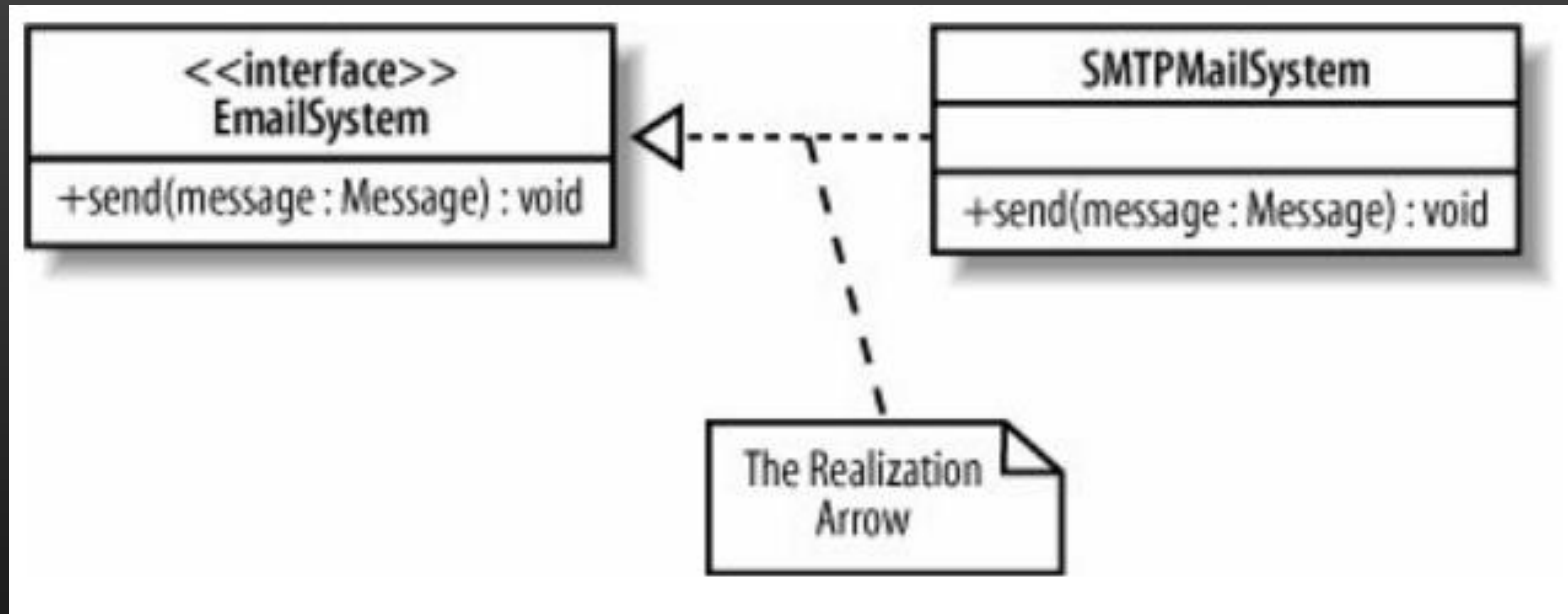
Object diagrams



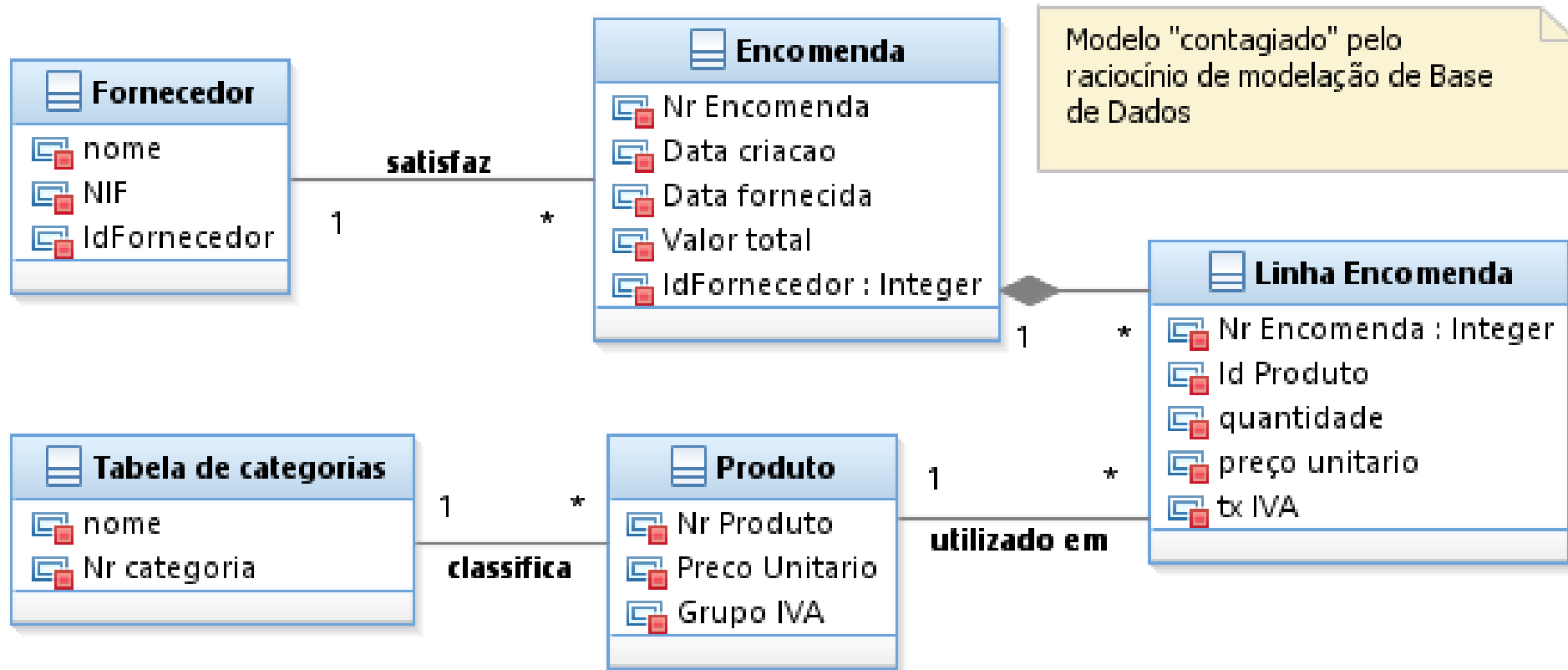
Example

O João Joaquim autorizou a requisição de material #75, que pedia tinteiros CLI-520.

Interfaces are unimplemented and stateless contracts (programming concept)



The domain model is not the model of a database



Readings & references

Core readings	Suggested readings
<ul style="list-style-type: none">• [Dennis15] – Chap. 5	<ul style="list-style-type: none">• [Larman12] –Chap. 9• VisualParadigm: what is the class diagram?