

# Apresentação Plaenge

O que vamos abordar?

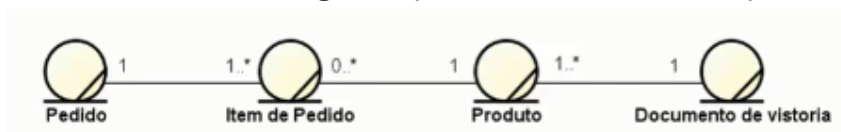
1. Modelagem de domínio e os níveis de abstração (nível conceitual X nível de design)
2. Conceitos e regras para os modelos
3. Como identificar conceitos e algumas orientações
4. Associações e multiplicidades
5. Desenho de instâncias
6. Exercício prático

## 1. Modelagem de domínio e os níveis de abstração

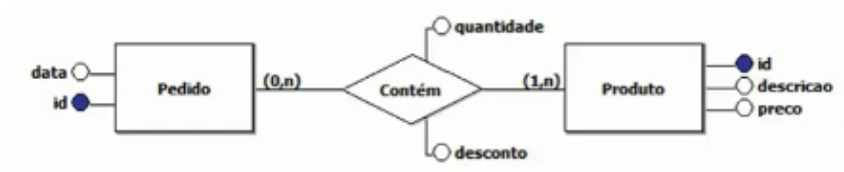
A Modelagem de domínio tem níveis de abstração e cada uma revela o quanto escondemos detalhes.

Nível	Objetivo
Conceitual ou de análise(de negócio)	Descrever as entidades do domínio (do negócio) e suas inter-relações. <b>Independente de sistema.</b>
Conceitual ou de análise (de sistema)	Descrever as entidades do domínio (do sistema) e suas inter-relações. <b>Independente de paradigma ou tecnologia</b>
Lógico ou de Design	Descreve as entidades do domínio (do sistema) e suas inter-relações. Preso a um <b>paradigma</b> (Ex: relacional, orientado a objetos) <b>porém independente de tecnologia.</b>
Físico ou de implementação	Descrever as entidades do domínio (do sistema) e suas inter-relações: Preso a um <b>paradigma</b> (Ex: relacional, orientado a objetos) e a uma <b>tecnologia</b> (Ex: Java, C#, Python)

- Conceitual de negócio (pode incluir entidades que não necessariamente farão parte do sistema):

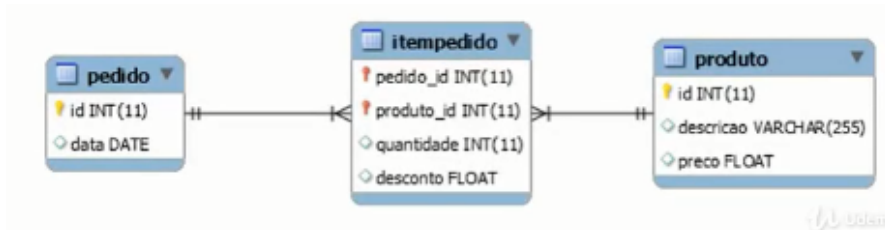


- Conceitual de sistema(Independente de paradigma):

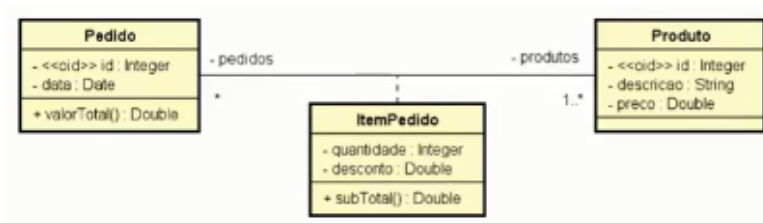


- Lógico ou design:

Relacional:



Orientado à objetos:



- Físico (implementação) - Código:

Preso ao paradigma: **relacional**

Preso à tecnologia: **MySQL (dialeto SQL)**

```
CREATE TABLE `produto` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `descricao` varchar(255) DEFAULT NULL,  
  `preco` float DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
(...)
```

Preso ao paradigma: **orientado a objetos**

Preso à tecnologia: **Java**

```
public class Produto {  
    private Integer id;  
    private String descricao;  
    private Double preco;  
  
    public Produto(Integer id, String descricao, Double preco) {  
        this.id = id;  
        this.descricao = descricao;  
        this.preco = preco;  
    }  
  
    (...)
```

## 2. Definições e regras para os modelos

**Conceitos** - Também compreendidos como **entidade**, devem possuir **significado**, **unidade coesa** e necessidade de **armazenamento**.

- Exemplos: cliente, pedido, produto, fornecedor, etc.

**Atributos** - Também chamados de **propriedades**.

- Formações alfanuméricas contidas nos conceitos
- Não devem ser multivalorados
  - Exemplo: Telefones "11999999999, 11888888888, 19555555555"
- Não deve ser composto
  - Exemplo: Endereço "Rua Florida, nº 250, Bairro da galega, CEP 99999-999"

**Regras para modelagem:**

Conceito ou entidade é representado por um retângulo dividido em três seções

- 1 - Nome conceito/entidade
- 2 - Atributos/propriedades
- 3 - Métodos (Não é usada no Modelo conceitual sem paradigma)

#### Opções

- Sempre ter um atributo identificador - estereótipo <<oid>>
- Atributos pode ter valor inicial
- Atributos podem ser de apenas leitura

## 3. Como identificar conceitos e algumas orientações

A modelagem de nível lógico pode depender da boa descrição dos requisitos e aqui seguem lugares que podem trazer essas informações:

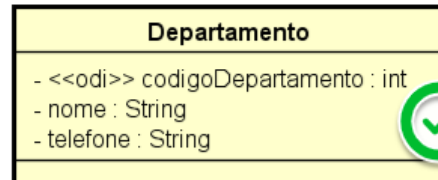
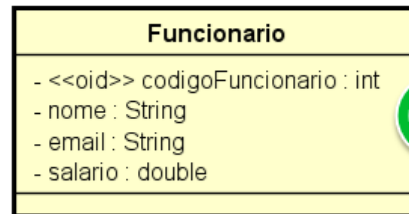
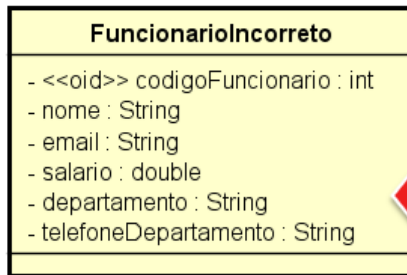
- Descrição do negócio
- Requisitos do sistema
- Casos de uso
- Documentação, regulamentos, normas e leis
- Processos de negócios

#### Orientações para identificar conceitos

- Documentos (Ordem de serviço, orçamentos)
- Pessoas (Cliente, fornecedor)
- Estrutura organizacional (Departamento)
- Eventos (Venda, reserva, atendimento)
- Procure por substantivos (Pessoa, compra, produto, pagamento)
- Expressões que denotem substantivos (autorização de pagamento)
- Verbos que indiquem possíveis conceitos (Comprar, pagar)

#### Exercício teste

"Deseja-se fazer um sistema para manter um **cadastro** dos **funcionários** de uma empresa. Deseja-se poder consultar o **email** e **salário** dos funcionários, bem como o **telefone** de seu **departamento**"



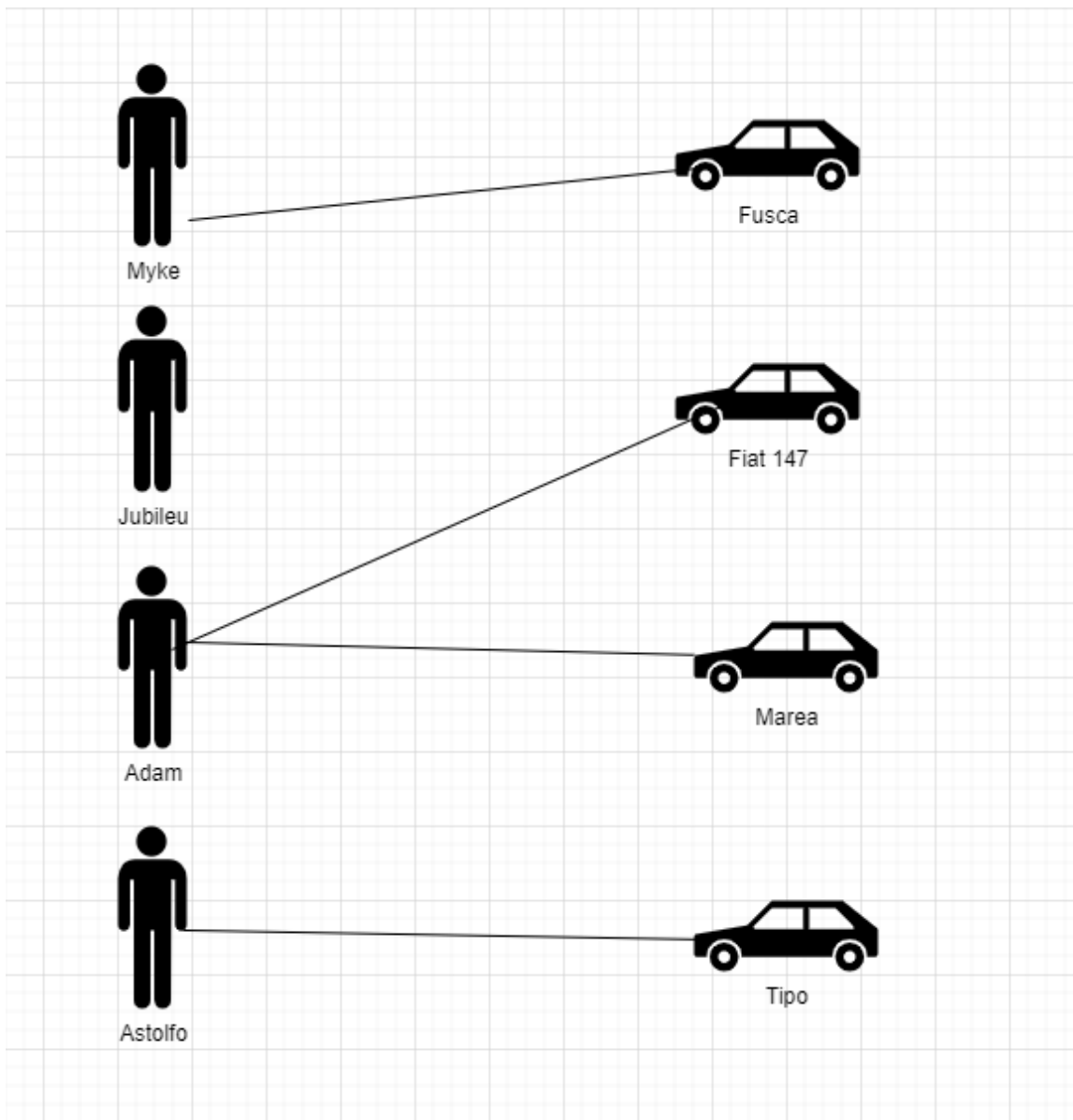
## 4. Associações e multiplicidades

A **associação** é um **relacionamento estático** entre dois conceitos

Exemplo:

"Desejo criar um sistema para armazenar informações de **pessoas** e **carros**."

Pergunta - Quem é o dono de cada carro?



A multiplicidade informa sobre o tipo de associação que ocorre entre os conceitos e para encontrar a multiplicidade, podemos recorrer ao seguinte método:

Pergunte para os dois lados:

1 <**conceito**> pode ter quantos <**papel**>?

1 carro pode ter quantos donos?

1 pessoa pode ter quantos carros?

(Sempre iniciar coma pergunta 1)

Existem alguns tipos de associação que podem ocorrer:

- Um para muitos
- Um para um
- Muitos para muitos

### Associação obrigatória

- Se o conceito associado tem multiplicidade maior que zero
  - Associação de uma pessoa com carros não é obrigatória
  - Associação de um carro com dono é obrigatória
  -



Embora pagamento seja "obrigatório" em um pedido, o pedido pode existir sem o pagamento por um tempo, logo não é uma associação obrigatória

### Conceito Dependente

- Se ele possuir pelo menos uma associação obrigatória
  - Pessoa é um conceito independente
  - Carro é um conceito que dependente
    - Só pode existir se existir uma pessoa dona dele
    - Se o dono deixa de existir, o carro também deixa de existir

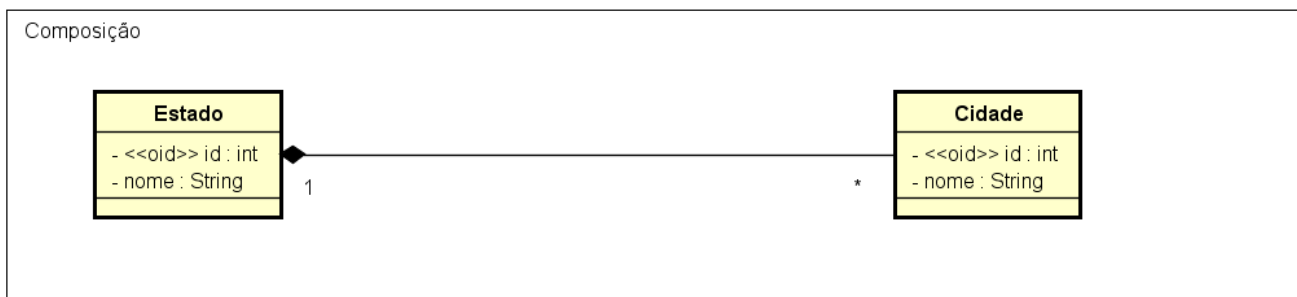
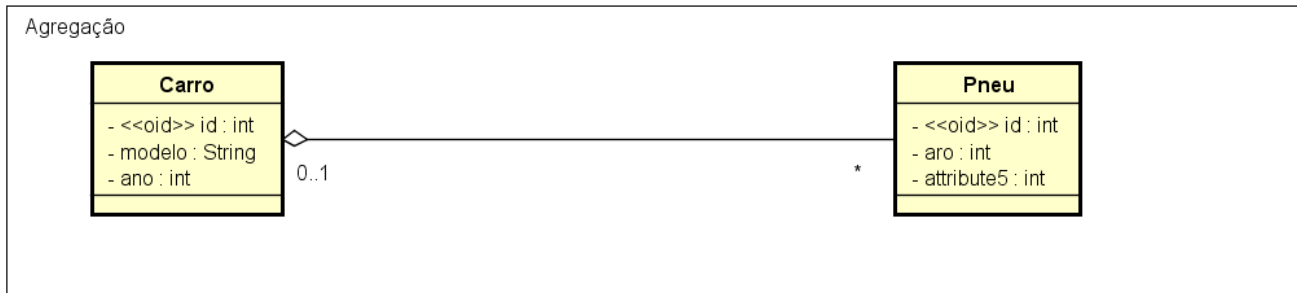
### Auto associação

- Quando um conceito é associado a ele mesmo

**Associação todo-parte** - Quando um conceito é parte de outro que representa um todo

- Agregação

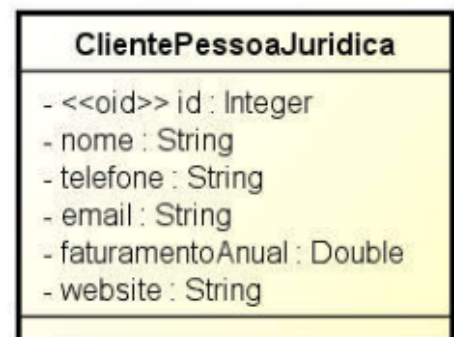
- Parte não exclusivo
- Composição
  - Parte exclusiva



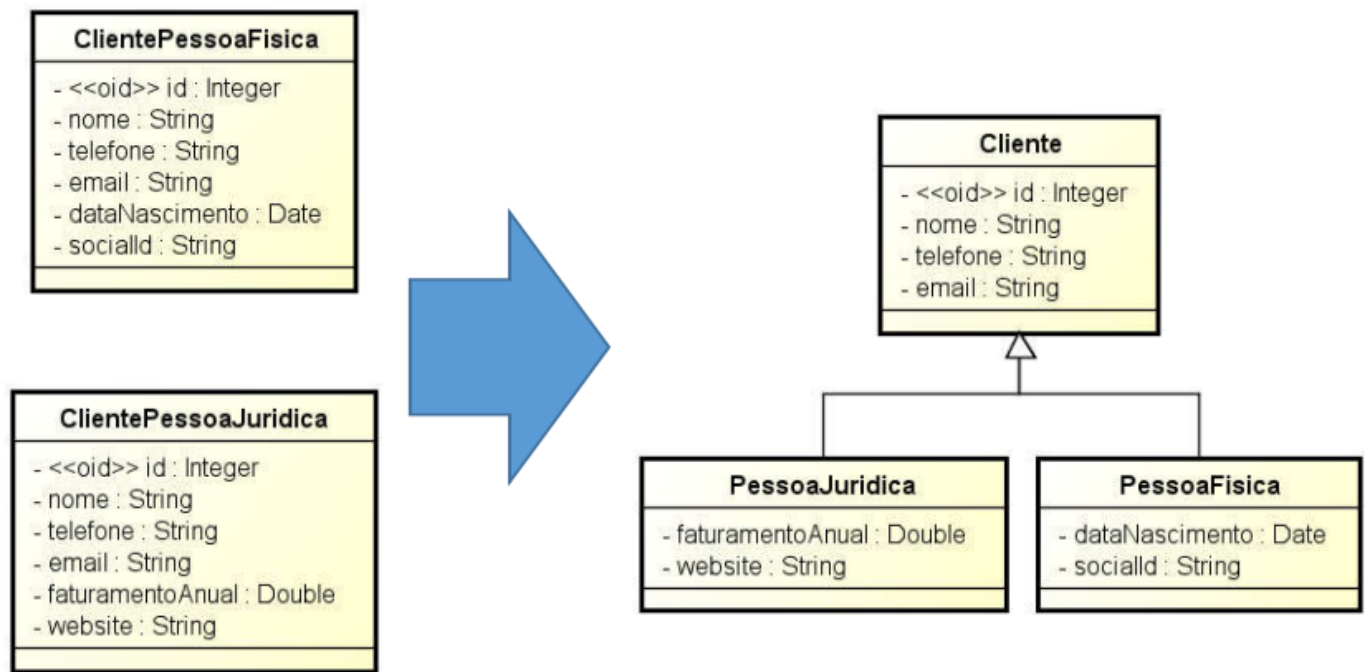
**Classe de associação** - Quando temos dados que pertencem a associação. Vamos entender melhor sobre essa associação no exercício ao final.

## Herança

Exemplo - Temos duas entidades cliente com atributos/propriedades em comum, logo pensamos em herança





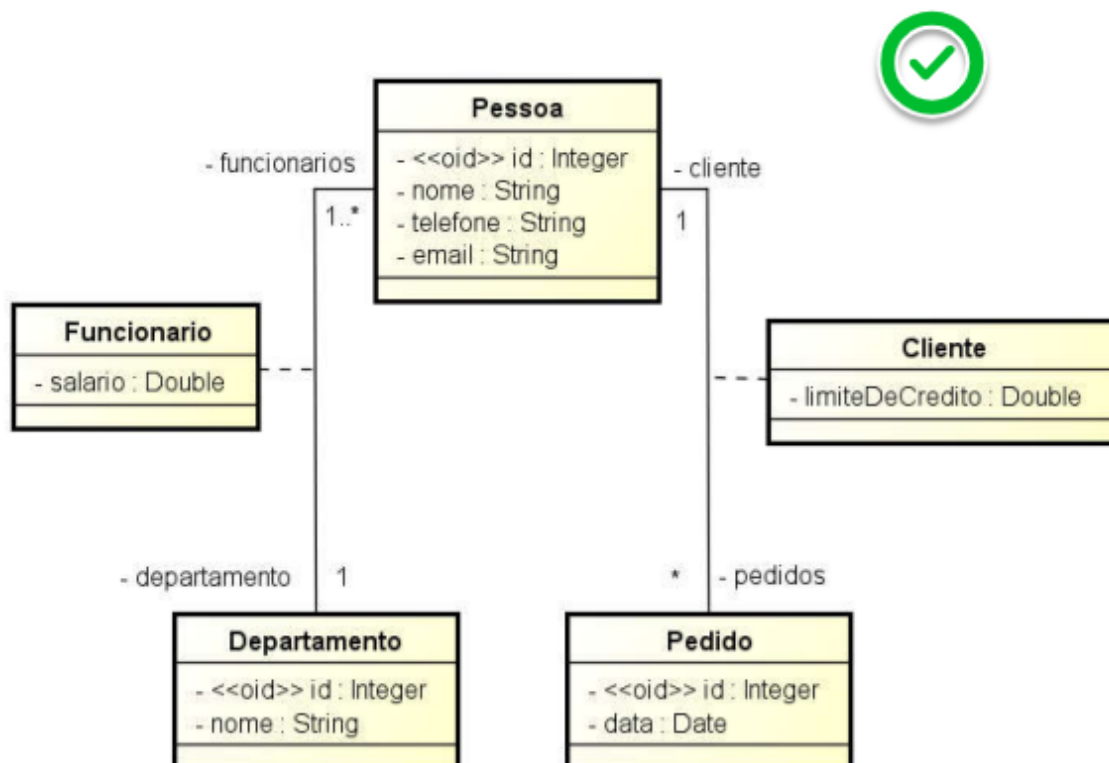
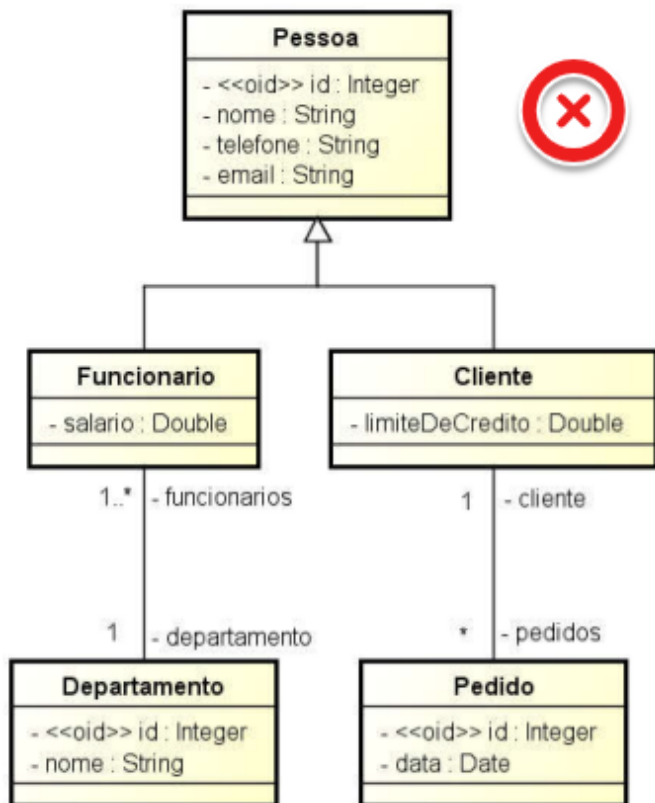


Considerações a realizar:

Existe estrutura em comum?

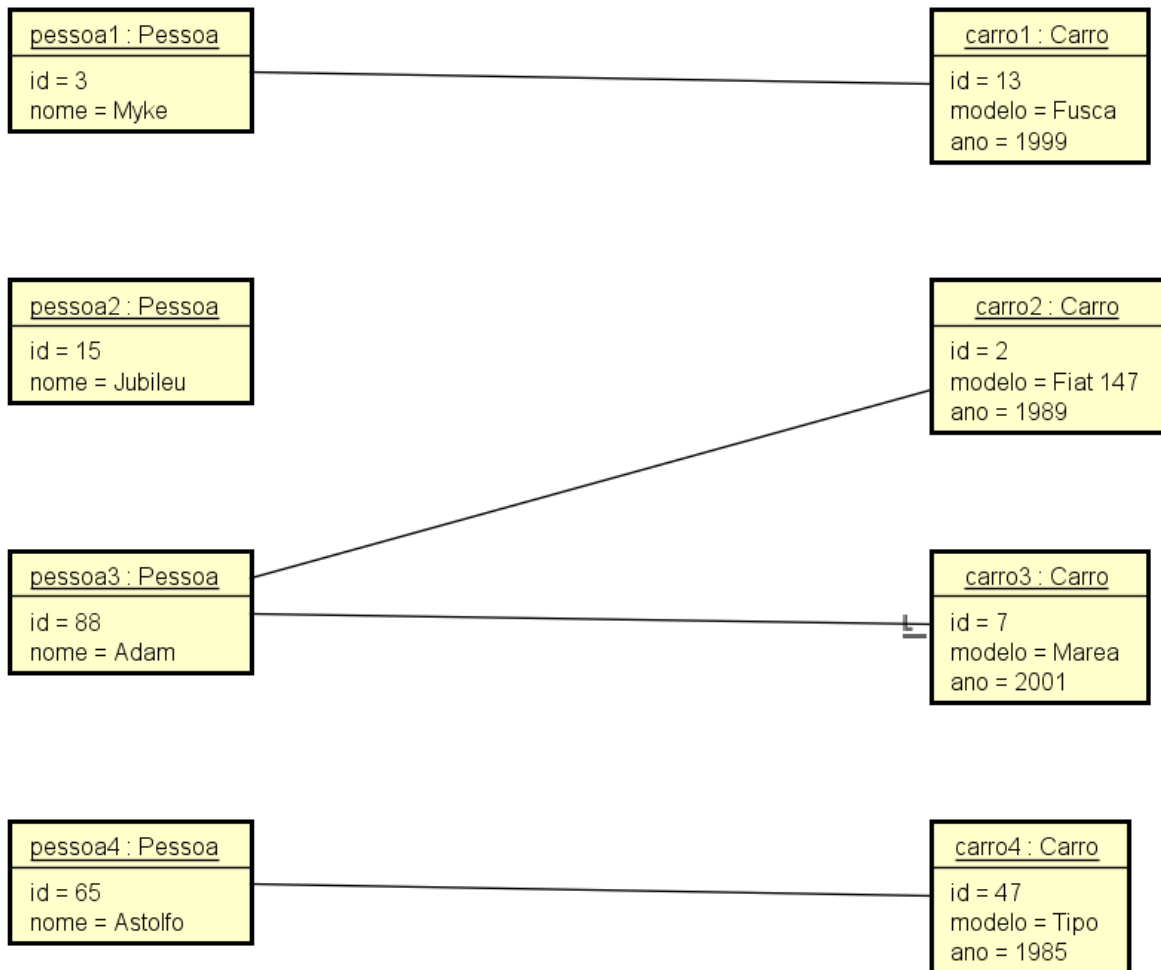
Há relação É-UM entre os conceitos e um conceito genérico que pode representar a estrutura em comum?

Uso impróprio de herança:

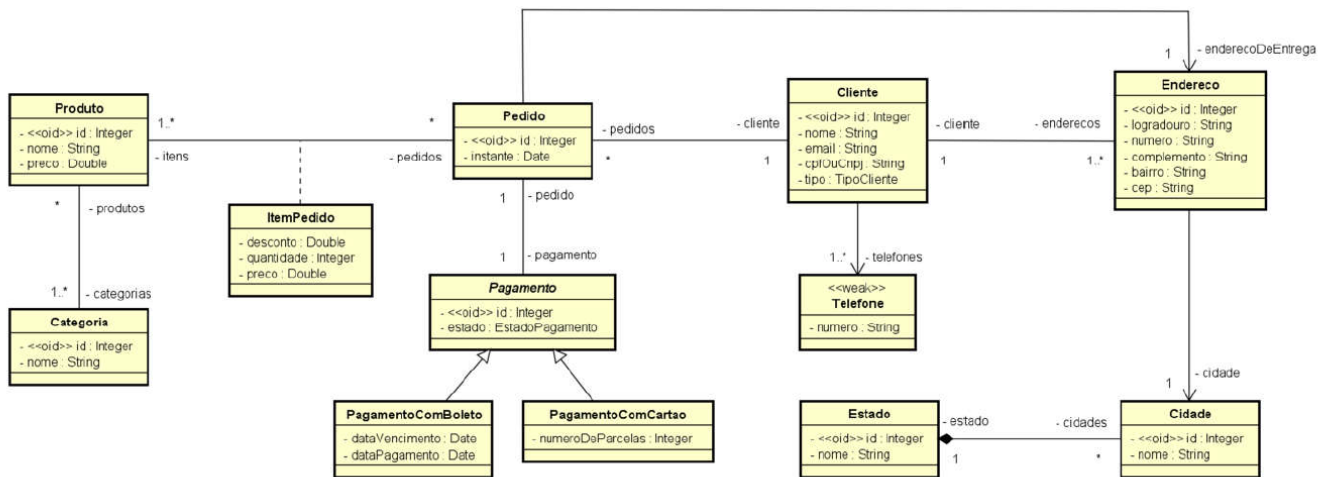


## 5. Desenho de instâncias

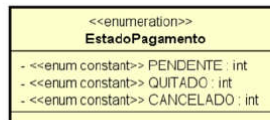
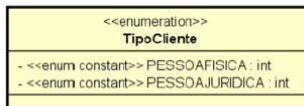
- Cada ocorrência de um conceito é chamado de instância ou objeto
- Ajuda a compreender e descobrir problemas
- Diagrama de objetos



Exemplo exercício



Enumerações:



Legendas:

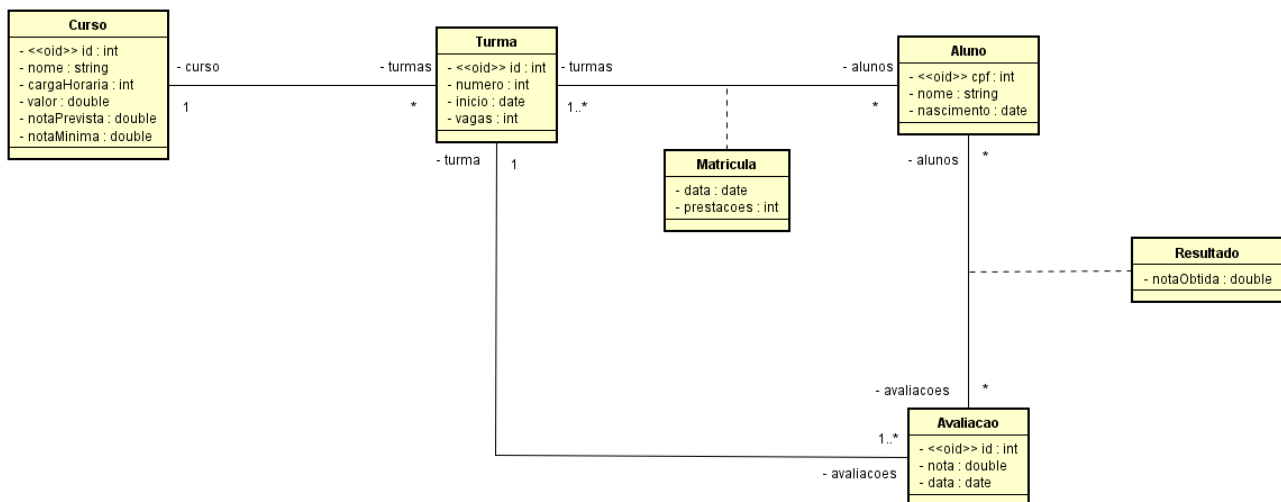


## 6. Exercício prático

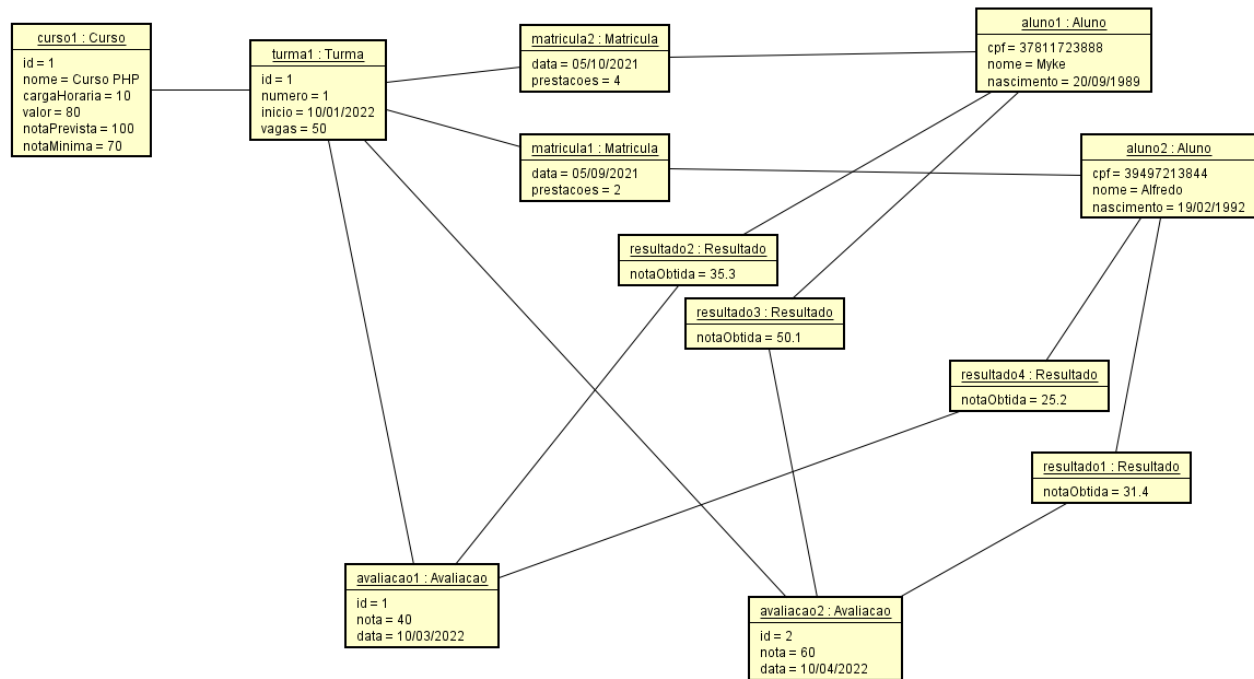
Deseja-se construir um sistema acadêmico. Para isso, são registrados os cursos disponíveis, onde cada um possui um nome, carga horária e valor. Quando um curso vai ser oferecido, é registrada uma turma, informando os seguintes dados: número da turma, data de início e número de vagas. Uma matrícula de um

aluno em uma turma consiste na data de matrícula e no número de prestações em que o aluno vai pagar o curso. Para cada aluno, é necessário cadastrar seu nome, CPF, e data de nascimento. Cada aluno passa por várias avaliações durante o desenrolar do curso que está cursando. Uma avaliação possui nota e data. Depois que a avaliação ocorre, é registrado resultado de cada aluno da turma (a nota que ele tirou). Um aluno é aprovado em um curso se sua nota total for maior ou igual à nota mínima de aprovação prevista para o curso.

Solução:



Instância mínima: 1 curso, 1 turma, 2 matrículas e 2 avaliações.



Fonte:

<https://www.udemy.com/course/uml-diagrama-de-classes/>