



DIAGRAMA DE CLASSES

ENGENHARIA DE SOFTWARE I

Análise e Desenvolvimento de Sistemas

Prof. Evandro Zatti, M. Eng.

MODELAGEM CONCEITUAL

- A modelagem de um sistema orientado a objetos consiste:
 - ✓ Na análise do domínio da aplicação;
 - ✓ Modelagem das entidades;
 - ✓ Modelagem dos fenômenos do domínio.
- Esta tarefa envolve basicamente dois mecanismos:
 - ✓ Abstração;
 - ✓ Representação.

MODELAGEM CONCEITUAL

ABSTRAÇÃO

Observar um domínio e capturar sua estrutura

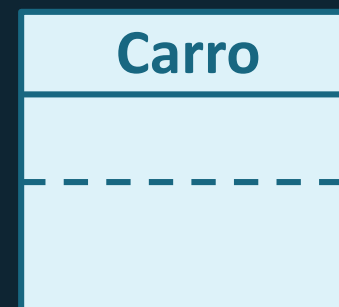


Entidade Observada



REPRESENTAÇÃO

Descrever o domínio de forma convencional (Ex.: UML)



Entidade Representada

FUNDAMENTOS

“Uma **classe** representa um conceito distinto dentro da aplicação que está sendo modelada, representando coisas de um **tipo específico** – um elemento físico (como um avião), um elemento comercial (como um pedido), [...], um elemento de aplicativo (como um botão) [...]. A **classe** é a descrição para um conjunto de objetos com **estrutura**, **comportamento** e **relacionamentos** similares.”

(BOOCH *et al*, 2004, p. 50)

CLASSE E OBJETO

- **Classe** é uma **definição** abstrata de um tipo composto;
 - ✓ permite a inserção de tipos heterogêneos de dados e as funcionalidades estão subordinadas a esse tipo.
- **Objeto** é a **instância** de uma classe.

ESTRUTURA DA CLASSE

- **Nome:** identificador único da classe
- **Atributos:** características da classe (variáveis de classe)
- **Métodos:** funcionalidades (operações) da classe

CLASSE (REPRESENTAÇÃO UML)

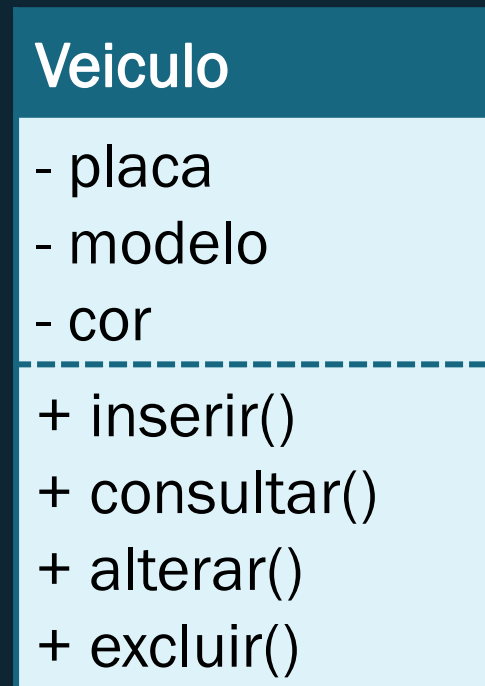
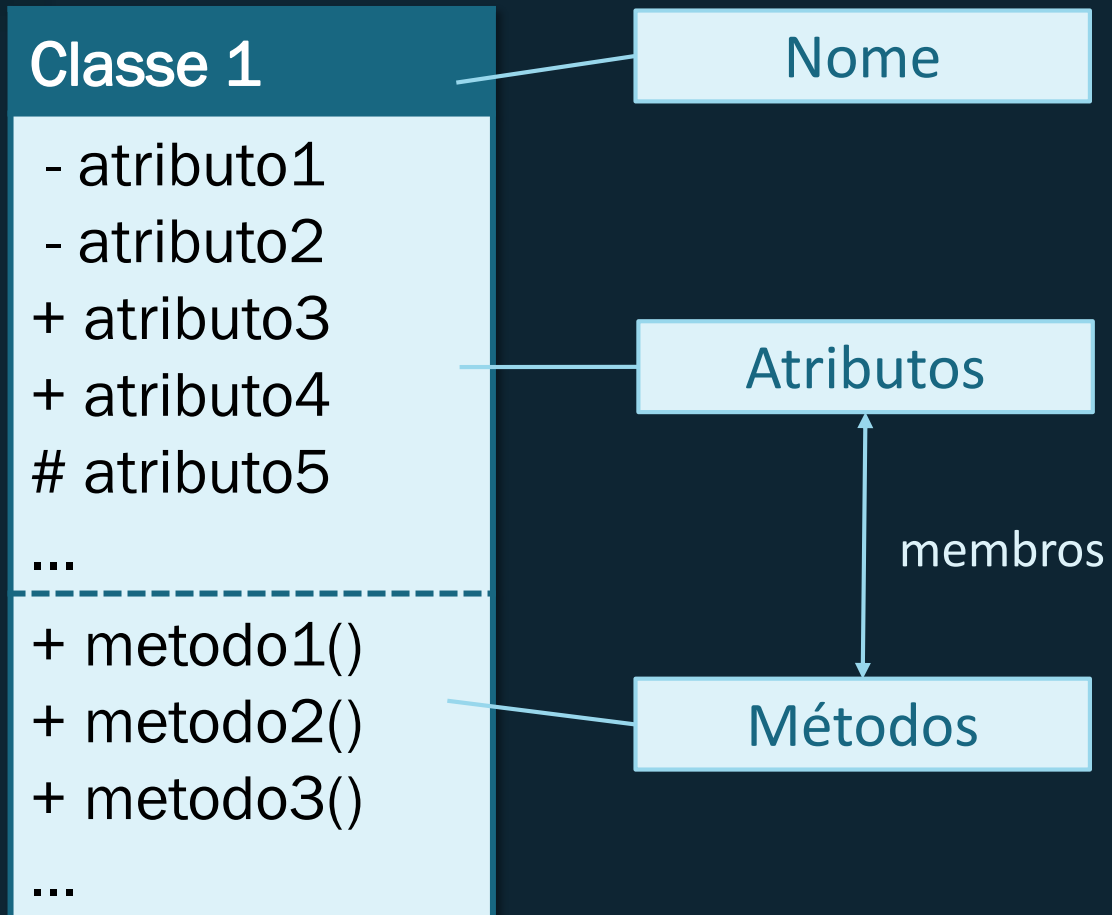


DIAGRAMA DE CLASSES

“A visão estática (**diagrama de classes**) é a base da UML. Os elementos da visão estática são conceitos que fazem sentido em uma aplicação, incluindo conceitos do mundo real, conceitos abstratos, conceitos de implementação, e conceitos computacionais. Os elementos chaves na visão estática são as **classes** e seus **relacionamentos**.”

(BOOCH *et al*, 2004, p. 47)

DIAGRAMA DE CLASSES

- É a representação gráfica das estruturas, comportamentos e relacionamentos entre as classes de um sistema:



RELACIONAMENTOS ENTRE CLASSES

- Os tipos de relacionamentos mais comuns são:

———— associação

← - - - - dependência

◁ ——— herança

◊ ——— composição

◊ ——— agregação

RELACIONAMENTOS ENTRE CLASSES

- **Associação:** indica que uma classe está relacionada a outra. Há uma relação **estrutural** entre elas: uma classe define o tipo de um atributo da outra classe.

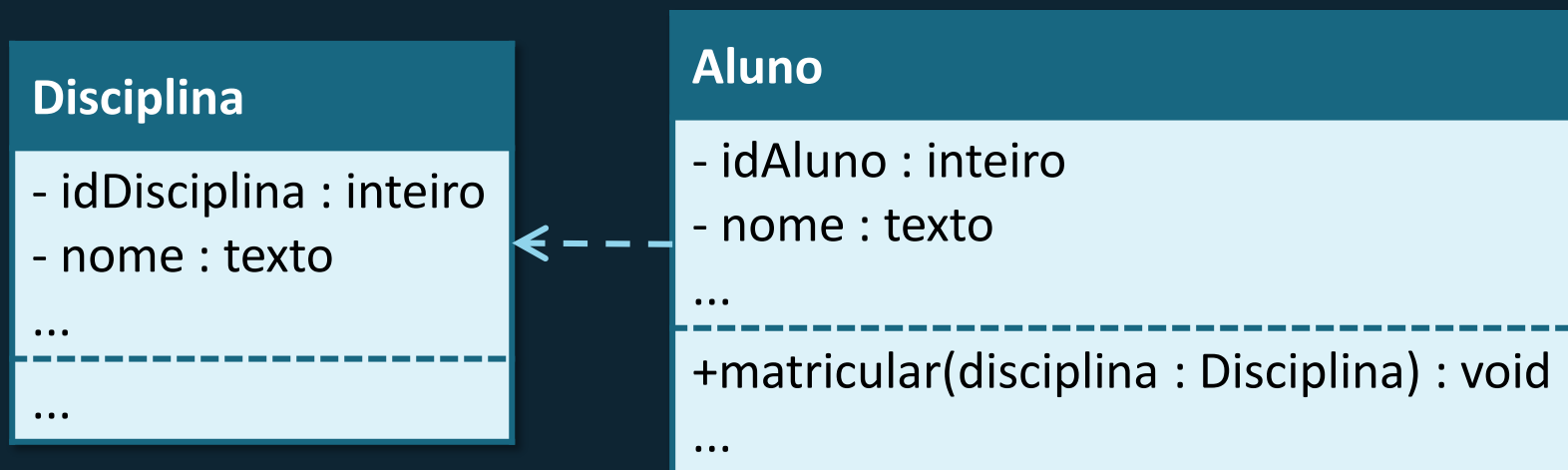
✓ Ex.: uma **cidade** pertence a um **estado**



RELACIONAMENTOS ENTRE CLASSES

- **Dependência:** é um relacionamento de **utilização**. Alterações na especificação de um tipo poderão afetar outro tipo que o utilize. Um exemplo comum é quando uma classe é usada como argumento de um método de outra.

✓ Ex.: matricular um **aluno** em uma **disciplina**



RELACIONAMENTOS ENTRE CLASSES

- **Herança**: relação de generalização/especialização.
 - ✓ Ex.: um **coordenador** é também um **professor**

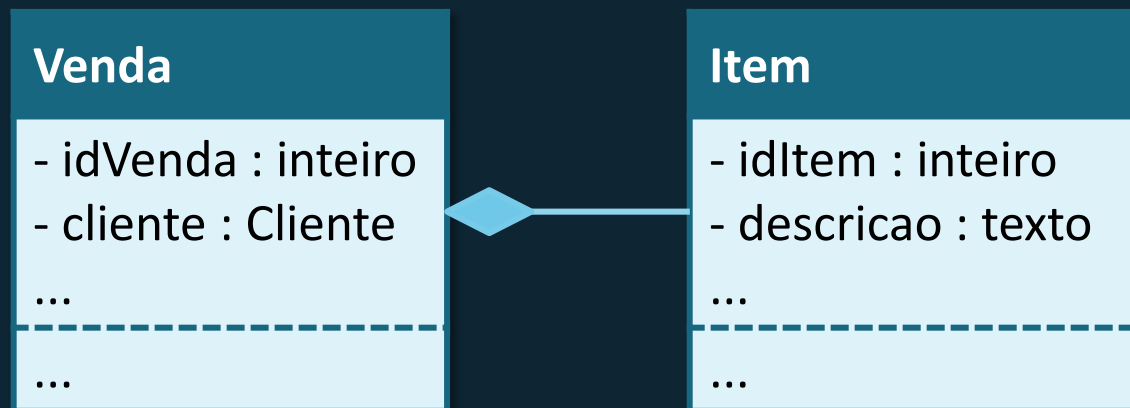


* O **coordenador**, que neste caso é a classe **especialista** (filha, derivada), irá herdar todos os membros do **professor**, que é a classe **generalista** (mãe, base).

RELACIONAMENTOS ENTRE CLASSES

- **Composição:** uma classe está contida na outra, (a parte não existe sem o todo).

✓ Ex.: uma **venda** é composta de **itens**.



* Um **produto** pode existir individualmente, independente da venda, mas ele só pode existir como um **item** de venda (com quantidade, valor comercializado, desconto, etc.), a partir do momento que existir uma **venda**. Se a **venda** deixar de existir, seus **itens** também deixarão de existir.

RELACIONAMENTOS ENTRE CLASSES

- **Agregação:** uma classe faz parte de um todo, porém sem contenção (a parte pode existir sem o todo).

✓ Ex.: um **computador** agrega vários **periféricos**.

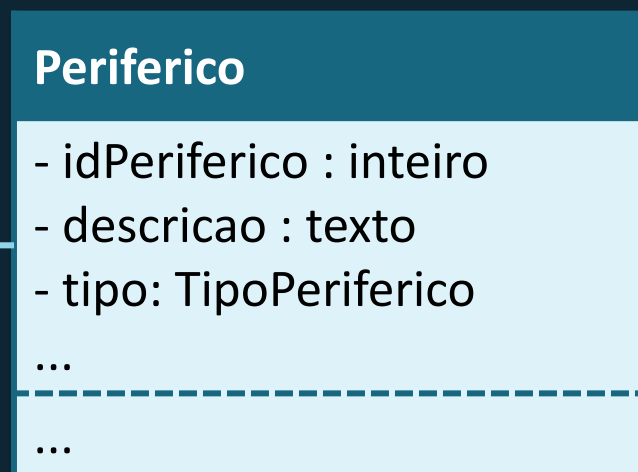
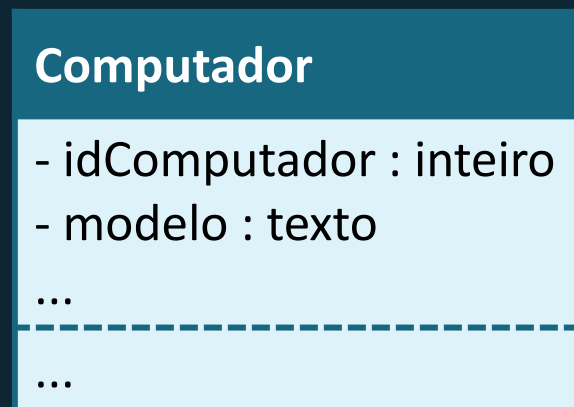
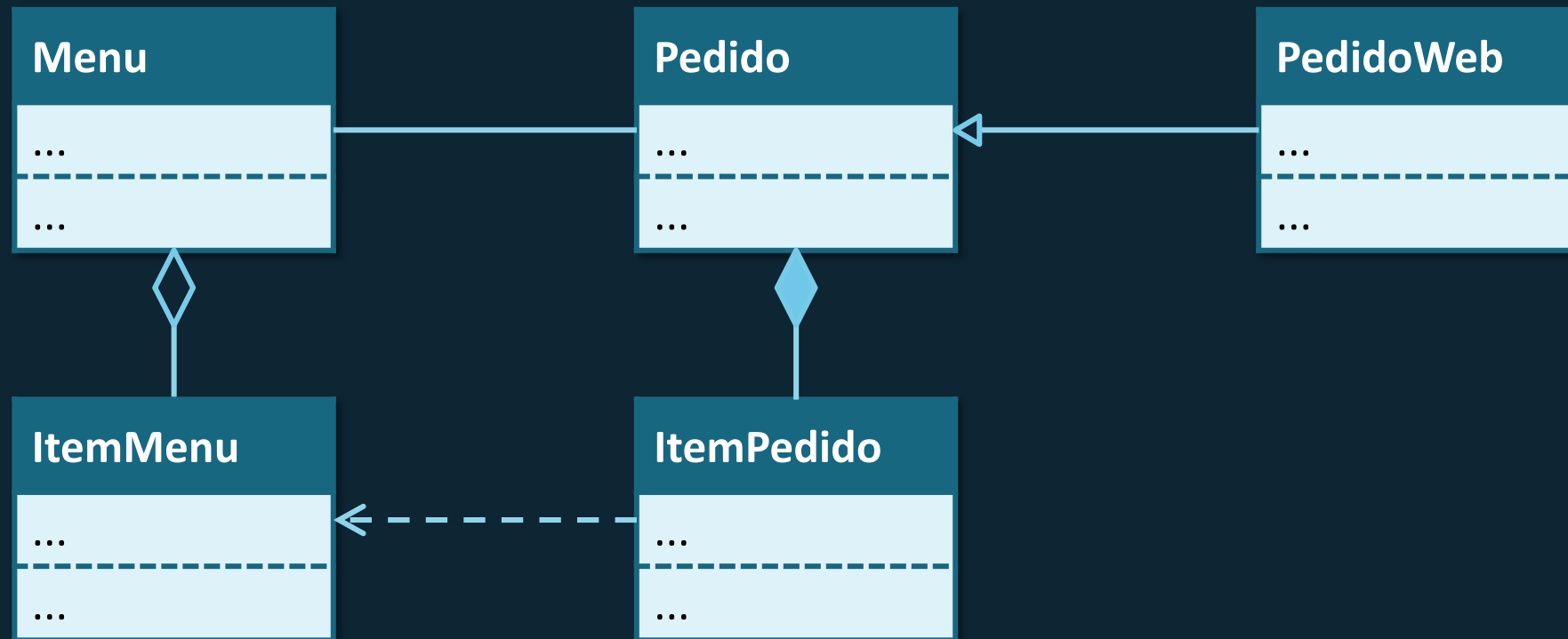


DIAGRAMA DE CLASSES (EXEMPLO)





ATIVIDADE PRÁTICA

REFERÊNCIAS

- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML 2.0 – *Reference Manual***. Boston: Addison Wesley, 2004.
- CARDOSO. A. Análise Orientada a Objetos.
 - ✓ Disponível em <http://www.alexandre.eletrica.ufu.br/esof/aula05.pdf>. Acesso em 16/03/2019.
- PRESSMAN, R. W, MAXIM B. R. ***Software Engineering - A Practitioner's Approach***. 8th Ed. New York: McGraw-Hill, 2015.