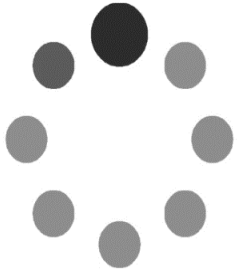
Curso: Desenvolvimento de Software Multiplataforma

Disciplina: Engenharia de Software II

Professor: André Olímpio

Engenharia de Software II



LOADING□

Diagrama de Classes

UNIFIED
MODELING
LANGUAGE

Fonte: UML.org

Diagrama de Classes

- É o principal diagrama da UML.
- É uma representação do sistema no ponto de vista dele mesmo, ou seja, de dentro para fora.
- Assim como o diagrama de casos de uso, é uma representação estática do sistema que será desenvolvido.
- É composto pelas classes e a associação entre as mesmas.

Diagrama de Classes

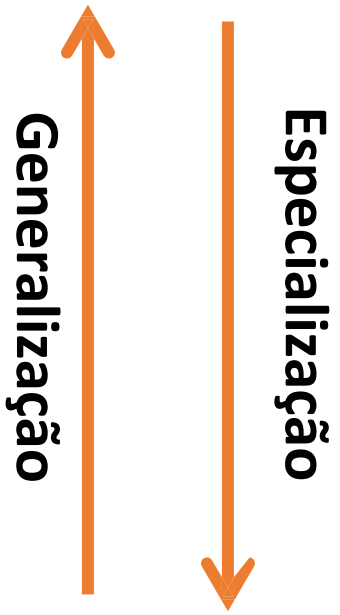
- Uma classe é formada por **métodos** e **atributos**.
- A classe pode ser **completa** ou **parcial**.
  - **Completa:** quando uma classe possui métodos e atributos definidos.
  - **Parcial:** quando uma classe possui somente métodos ou somente atributos.

Fonte: Pinterest.com

Diagrama de Classes

- O conceito de **generalização / especialização** neste diagrama segue o mesmo princípio da herança já estudado no diagrama anterior.

- Neste diagrama, assim como no de casos de uso, a herança também é representado por um triângulo. △



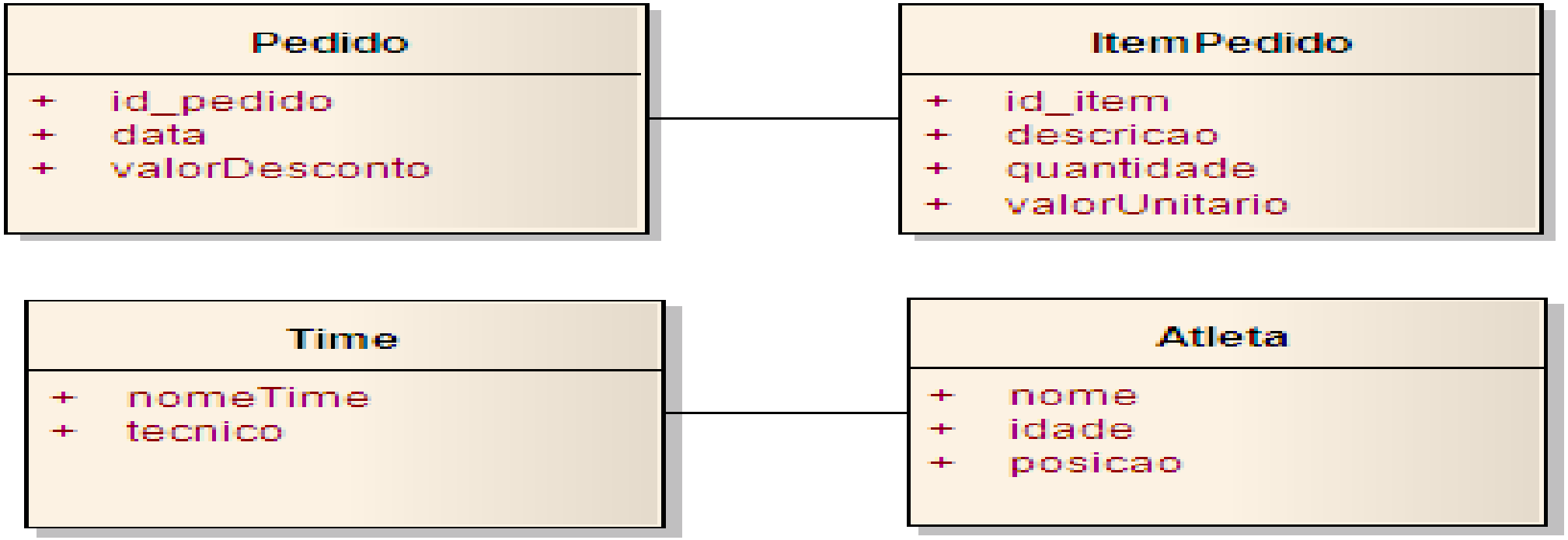
Fonte: Elaborado pelo autor

Diagrama de Classes

- Existem tipos específicos de associação entre classes.
- Sua utilização depende exclusivamente das regras de domínio do negócio a ser estudado.
- Estas associações são vitais para a definição clara do diagrama de classes.
- Estes tipos são: **agregação** e **composição**.

Composição / Agregação

- Entenda a seguinte situação:

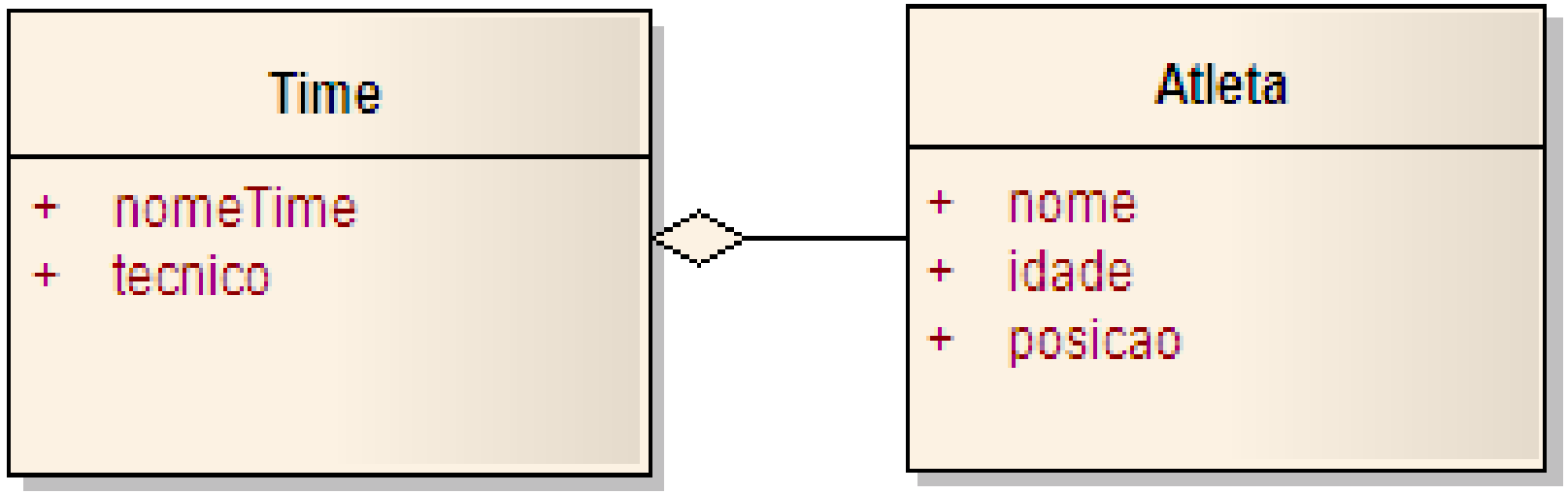


Pedido (Objeto-Todo) e **ItemPedido (Objeto-Parte)**
Time (Objeto-Todo) e **Atleta (Objeto-Parte)**

Fonte: Elaborado pelo autor

Agregação

- Na **Agregação**, a existência do Objeto-Parte faz sentido, mesmo não existindo o Objeto-Todo.
- Vejamos o exemplo Time-Atleta:

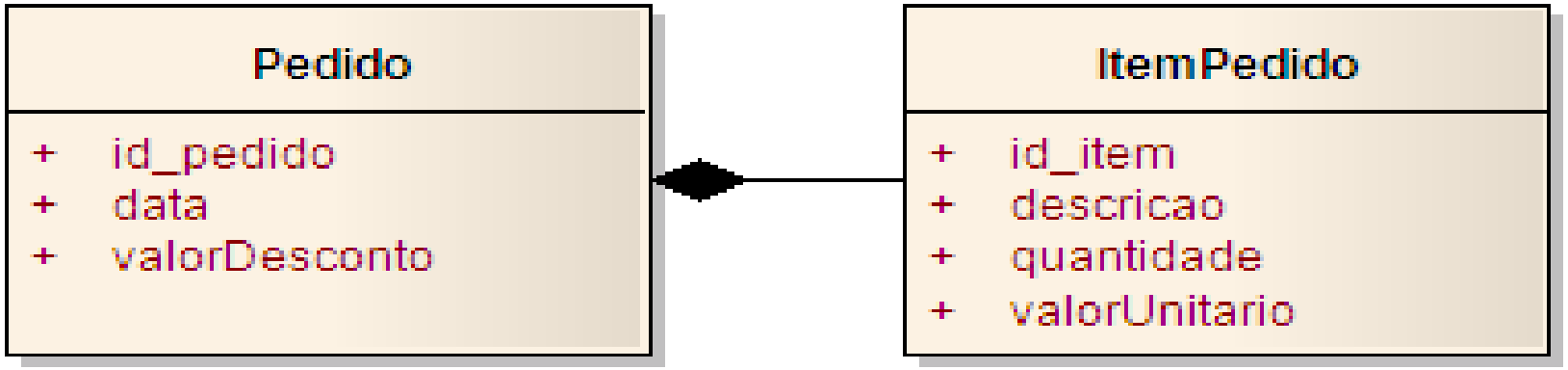


O símbolo de agregação difere do de associação por conter um **losango não preenchido** na extremidade da classe que contém os objetos-todo.

Fonte: Elaborado pelo autor

Composição

- É uma agregação mais forte.
- Nela a existência do Objeto-Parte NÃO faz sentido se o Objeto-Todo não existir.
- Vejamos o exemplo Pedido-ItemPedido:



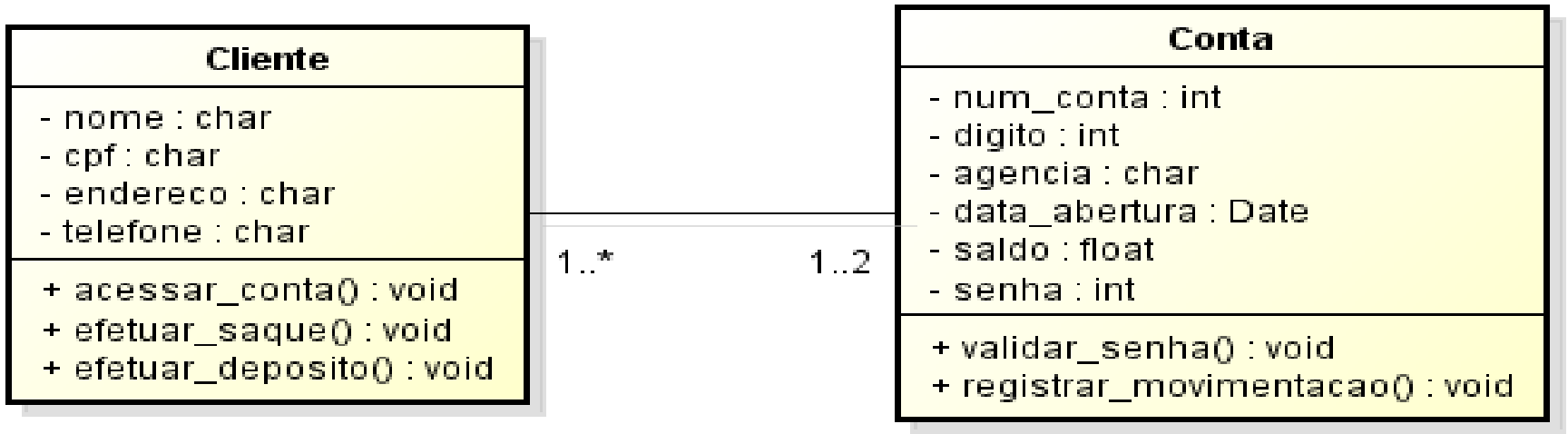
O símbolo de agregação difere do de associação por conter um **losango preenchido** na extremidade da classe que contém os objetos-todo.

Fonte: Elaborado pelo autor

Multiplicidade

- Indica o tipo de relacionamento entre as classes envolvidas.

Tipos	Significa
0..1	Zero ou uma instância. A notação n..m indica n para m instâncias.
0..* ou *	Não existe limite para o número de instâncias.
1	Exatamente uma instância.
1..*	Ao menos uma instância.



Fonte: Elaborado pelo autor

Atributos

- São as características da classe.
- São os valores tangíveis a serem armazenados na classe.
- Assim como uma variável em programação e um campo de uma tabela em banco de dados, todo atributo precisa ter um tipo de dado específico.
- Uma classe pode possui N atributos.

Métodos

- São as ações que a classe pode realizar.
- Um método obrigatoriamente retorna um valor tangível, através de parâmetros pré-definidos.
- **VOID** indica que o valor de retorno de um método pode ser nulo (não tangível).

Fonte: Pinterest.com

Estereótipos

Representa a toda a interface com o usuário do sistema a ser implementado. No diagrama é identificado pelo objeto **Boundary**.

Representa toda a implementação de código do sistema a ser desenvolvido. No diagrama é identificado pelo objeto **Control**.

Fonte: Elaborado pelo autor

Leitura Complementar

- **SOMOS TODOS T.I. – Diagrama da UML – Casos de Uso**

https://youtu.be/fk1JdFjKneE

SOMOS TODOS T.I. – Diagrama da UML – Classes

https://youtu.be/fH3mRDysjoA

- **SOMOS TODOS T.I. – Diagrama da UML – Sequência**

https://youtu.be/X-eTZgm4UQk

Fonte: FlatIcon.com