



*Técnico em Desenvolvimento de Sistemas Online*

# ANÁLISE DE SISTEMAS E PROJETOS

# GEEaD - Grupo de Estudo de Educação a Distância

## Centro de Educação Tecnológica Paula Souza

### Expediente

GEEaD – CETEC  
GOVERNO DO ESTADO DE SÃO PAULO  
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO  
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS  
FUNDAMENTOS DE INFORMÁTICA

*Autores:*

*Eliana Cristina Nogueira Barion  
Marcelo Fernando Iguchi*

*Revisão Técnica:*

*Lilian Aparecida Bertini*

*Revisão Gramatical:*

*Juçara Maria Montenegro Simonsen Santos*

*Editoração e Diagramação:*

*Flávio Biazim*

# APRESENTAÇÃO

Este material didático do Curso Técnico em Desenvolvimento de Sistemas modalidade EaD foi elaborado especialmente por professores do Centro Paula Souza para as Escolas Técnicas Estaduais – ETECs.

O material foi elaborado para servir de apoio aos estudos dos discentes para que estes atinjam as competências e as habilidades profissionais necessárias para a sua plena formação como Técnicos em Desenvolvimento de Sistemas.

Esperamos que este livro possa contribuir para uma melhor formação e aperfeiçoamento dos futuros Técnicos.

---

# AGENDA 12

---

## MODELAGEM DE DADOS ORIENTADA A OBJETOS





## Diagrama de Classes

O Diagrama de Classes é o mais utilizado da UML. Esse diagrama fornece uma visão estática do modelo a ser criado. Como as classes são um dos componentes mais importantes da orientação a objetos, esse diagrama deve constar de todo projeto orientado a objetos.

O Diagrama de classe não deve faltar em projetos orientados a objetos. Devemos prestar muita atenção ao criar um diagrama de classes, que será a base da nossa solução.

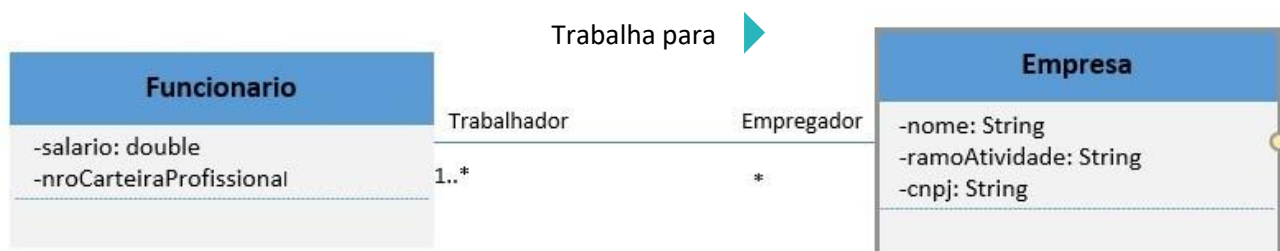
Principais componentes: classes, interfaces, relacionamentos.

Identificar uma classe não é tarefa das mais simples, mas o caminho é procurar itens que têm as mesmas informações e comportamentos. Nem sempre uma classe tem atributos e métodos. Pode ter apenas métodos ou apenas atributos. O caminho é analisar o cenário do projeto com base no diagrama de Caso de Uso, fazer uma lista do que você identificou como classes. Acrescente os atores, que geralmente são também classes de seu sistema. Analise as candidatas a classes e tente achar atributos para elas e, se possível, alguma funcionalidade. Coloque as classes em um diagrama e comece a analisar as relações entre elas, de acordo com os tipos de relacionamentos que esse diagrama propõe. Veja:

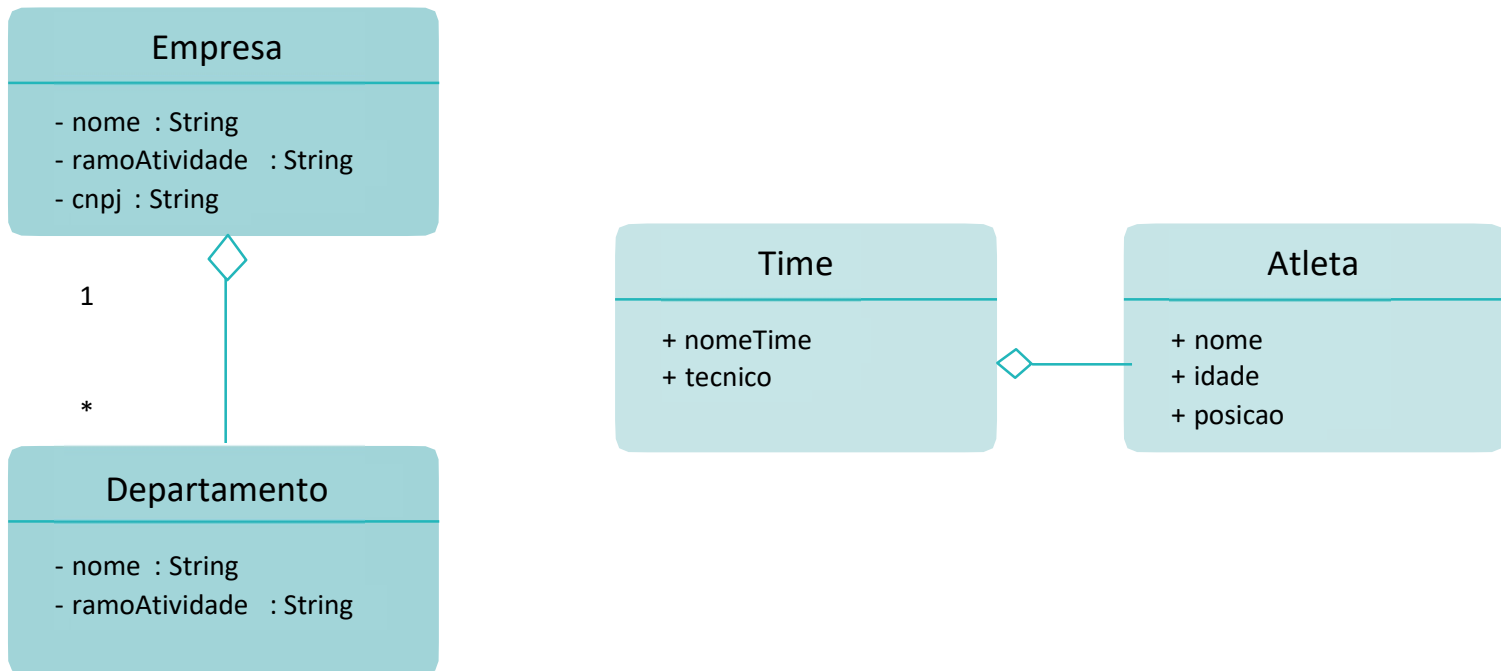
- Associação
- Agregação
- Composição
- Herança: Especialização/Generalização Dependência

**Associação:** é um relacionamento estrutural que especifica objetos de uma classe conectados a objetos de outra classe. A partir de uma associação, conectando duas classes, você é capaz de navegar do objeto de uma classe até o objeto de outra classe e vice-versa (BOOCH, RUMBAUGH e JACOBSON, 2005).

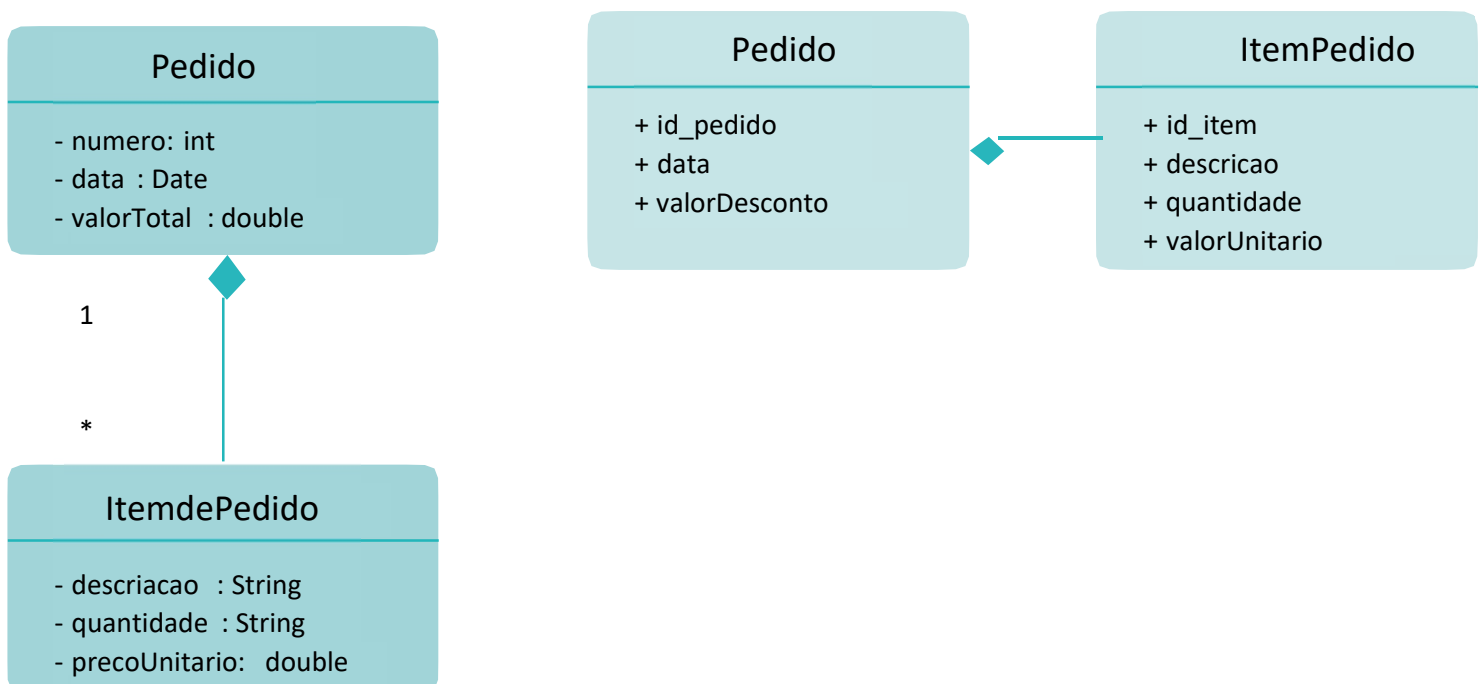
É representada por uma linha interligando as duas classes, uma associação pode definir papéis das classes relacionadas, assim como a multiplicidade de sua associação, além de ter um nome.



**Agregação:** é um tipo especial de associação que tenta demonstrar que as informações de um objeto-todo precisam ser complementadas pelas informações contidas em um (ou mais) objetos-parte. Exemplos de agregação:

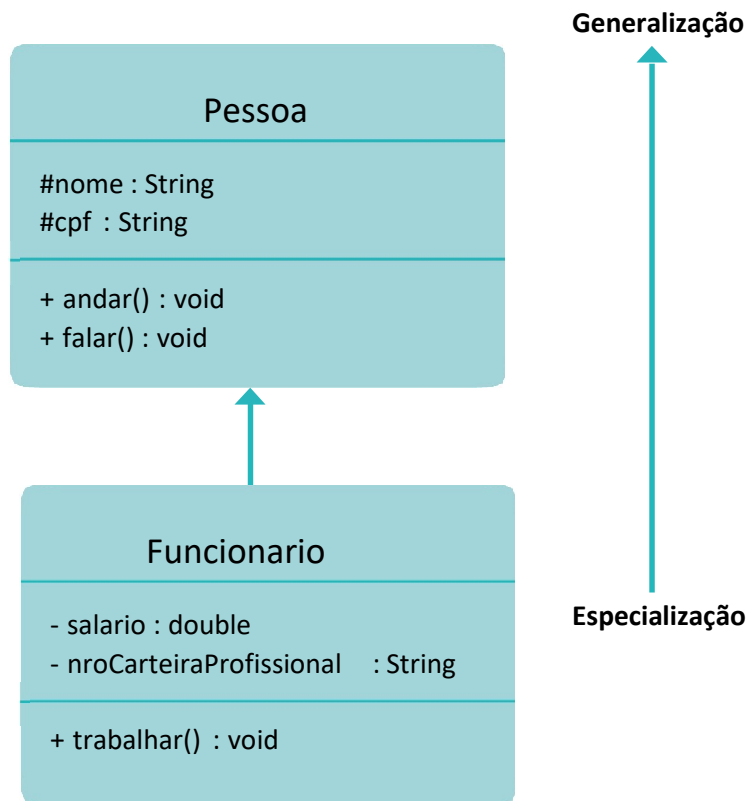


**Composição:** é uma variação da agregação e também representa uma relação todo - parte. No entanto, na composição o objeto-pai (todo) é responsável por criar e destruir suas partes. Em uma composição um mesmo objeto-parte não pode se associar a mais de um objeto-pai. Exemplos de Composição:



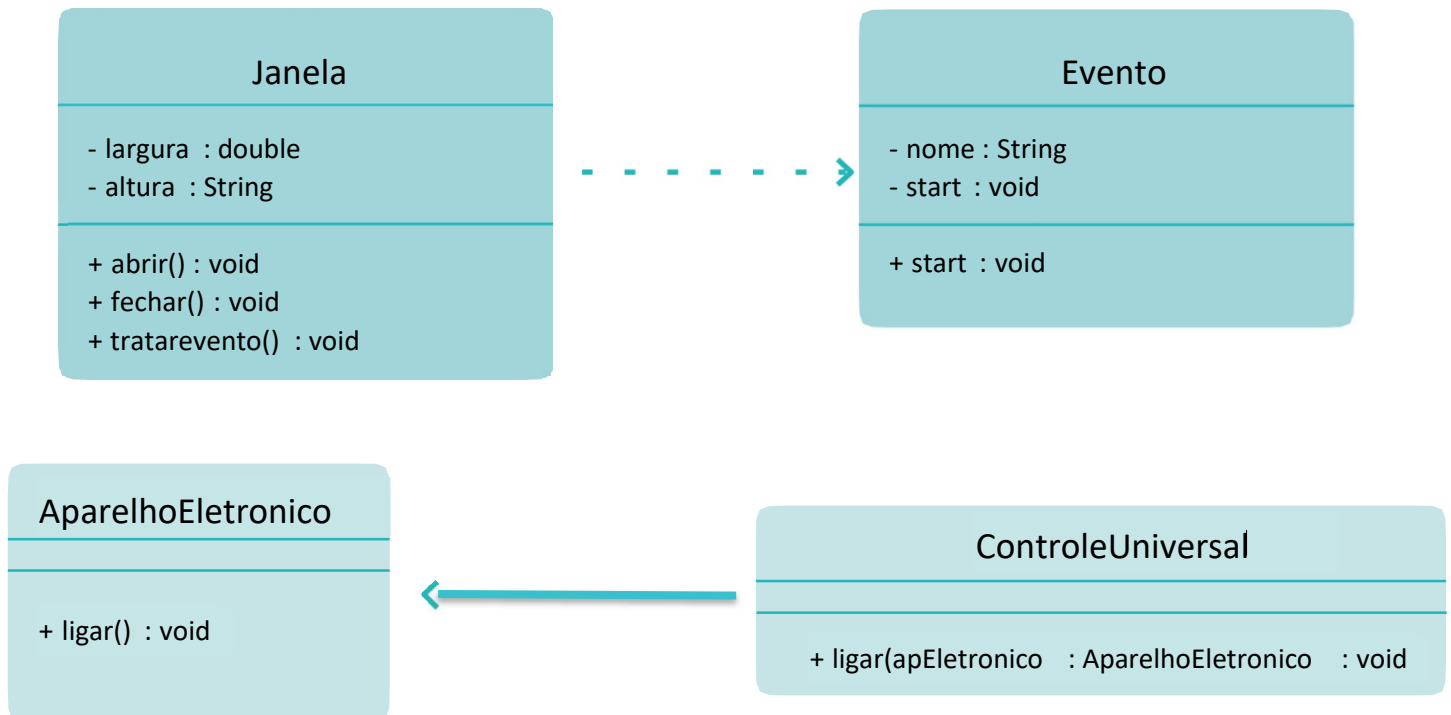
**Herança: Especialização e Generalização:** tem como objetivo identificar classes-mãe, denominadas de gerais, e classes-filha chamadas de especializadas. São chamados de relacionamentos “é um tipo de”.

Podemos verificar, na imagem a seguir, que a classe **Pessoa** possui os atributos `nome` e `CPF` e pode executar os métodos `andar` e `falar`. Já a classe **Funcionario**, por herdar os atributos e métodos da classe **Pessoa**, possui `nome`, `CPF`, `salário` e `nro Carteira Profissional`, podendo executar os métodos `andar`, `falar` e `trabalhar`. Dizemos que a classe **Pessoa** é a superclasse da classe **Funcionario**, que é sua subclasse, ou que **Pessoa** é a classe pai de **Funcionario** e que **Funcionario** é classe filha de **Pessoa**.



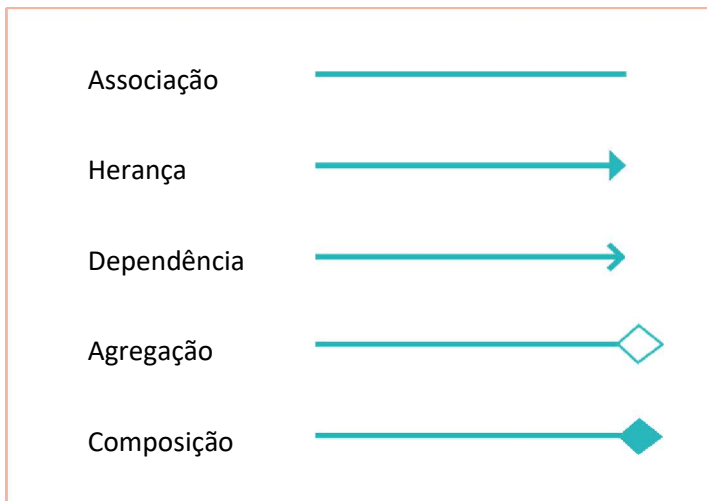
**Dependência:** como o nome sugere, indica um grau de dependência entre uma classe e outra. Uma dependência difere de uma associação porque a conexão entre as classes é temporária. É representada por uma seta tracejada entre duas classes.

Exemplos de Dependência:

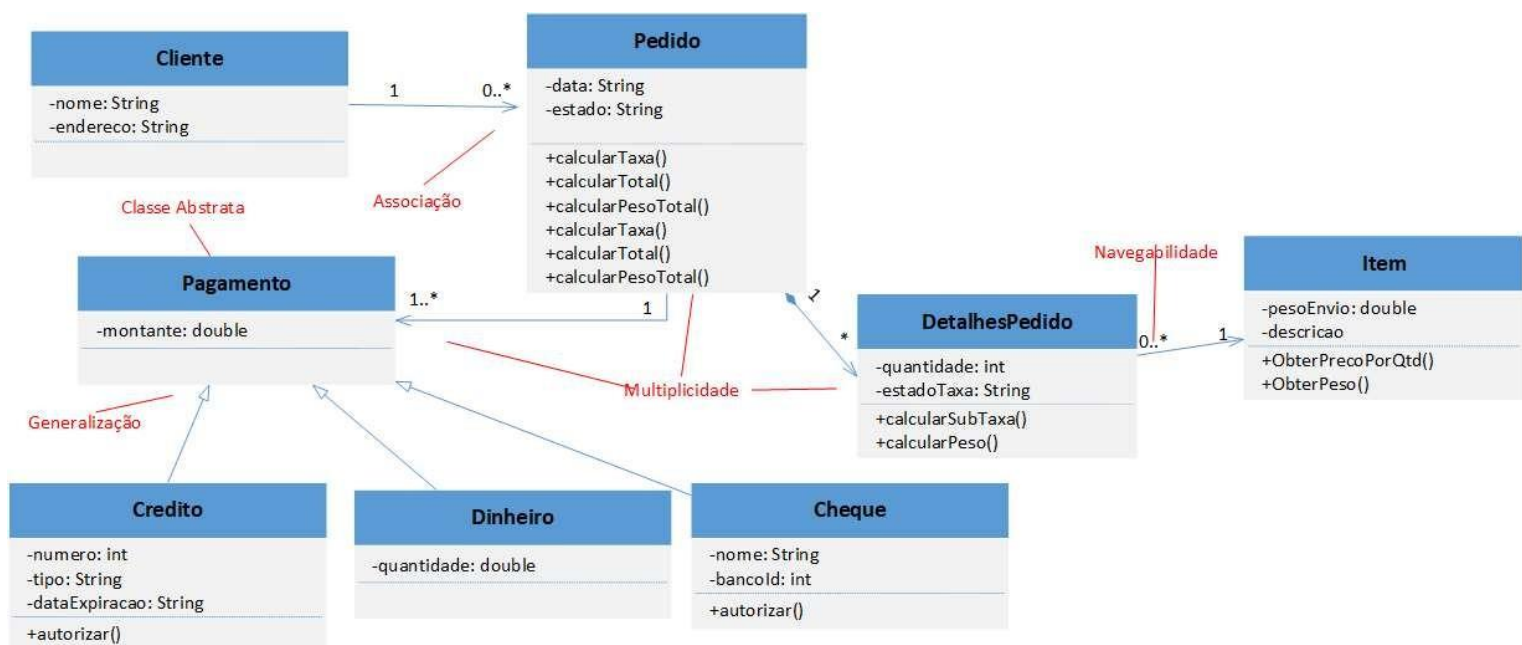




Representação dos tipos de relacionamentos da UML:



Exemplo de Diagrama de Classe e seus relacionamentos:



Exemplo de Diagrama de Classe (adaptado). Fonte: Tecnologia Local, 2009.

O diagrama de classe modela um pedido de um cliente para um catálogo de itens. A classe central é o Pedido. Associada a ela estão: o Cliente fazendo a compra e o Pagamento. Um Pagamento pode ser de três tipos: Dinheiro, Cheque ou Crédito. O pedido contém DetalhesPedido, cada um associado a um Item.

Assista ao Vídeo Programação Orientada a Objetos, disponível em <https://youtu.be/6kmDjHlK760?t=362> Acessado em 11/07/2022.