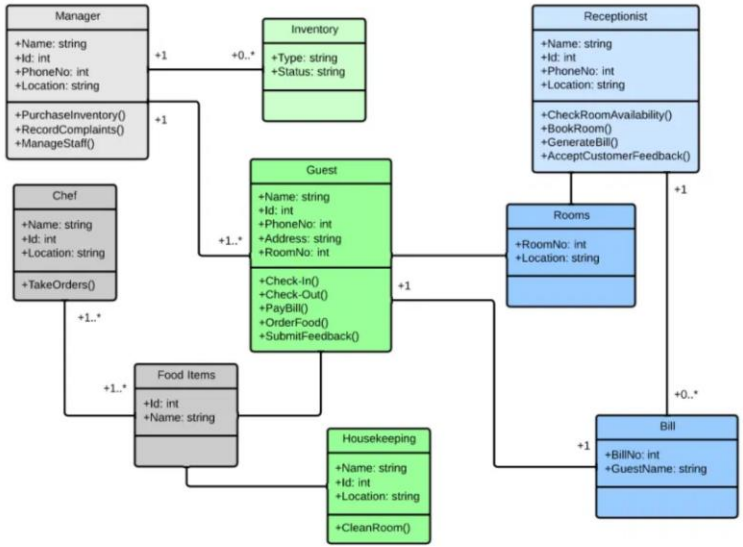
Diagramas de Classe UML

Os **diagramas de classe UML** são ferramentas essenciais na modelagem de software orientado a objetos.

- Mapeiam a **estrutura** de um sistema
- Modelam **classes**, **atributos** e **relações**
- Visualizam a arquitetura antes da implementação



Classe

Uma **classe** é como um **molde** ou **modelo** para criar objetos.
Ela define as características e comportamentos que os objetos daquele tipo terão.

Exemplo:
Imagine uma classe chamada Carro.
Ela pode ter **atributos** como **cor, modelo e ano**, e métodos como **ligar() e acelerar().**

Atributo

Um **atributo** é uma **variável** que pertence a uma classe ou a um objeto.
Ele representa uma **característica** do objeto.

Tipos:
Atributos de instância: específicos de cada objeto.
Atributos de classe: compartilhados por todos os objetos da classe.

Métodos

Um **método** é uma **função** definida dentro de uma classe.
Ele representa uma **ação** que os objetos daquela classe podem realizar.

Componentes Básicos de um Diagrama de Classe

Um diagrama de classe é composto por três partes principais:

Nome da Classe

Primeira seção - sempre obrigatória

Atributos

Segunda seção - descreve as propriedades

Métodos/Operações

Terceira seção - descreve os comportamentos

Modificadores de Acesso e Visibilidade

Os modificadores definem a **visibilidade** dos atributos e métodos:

Símbolo	Modificador
+	Público - visível para todas as classes
-	Privado - apenas para a própria classe
#	Protegido - classe e suas subclasses
~	Pacote - classes no mesmo pacote

Relacionamentos entre Classes - Parte 1

Generalização (Herança)

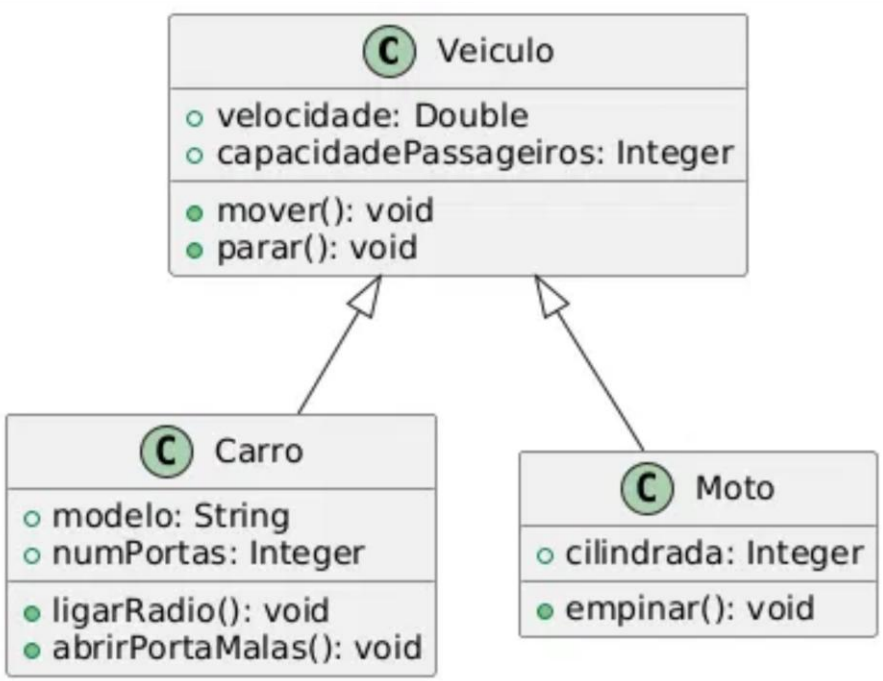
A subclasse herda todos os comportamentos e atributos da classe pai.

Notação: Linha sólida com triângulo vazio ▷

Implementação

Uma classe implementa os métodos definidos em uma interface.

Notação: Linha pontilhada com triângulo vazio ▷



Relacionamentos entre Classes - Parte 2

Associação

Relação entre classes independentes.

Notação: Linha sólida →

Agregação

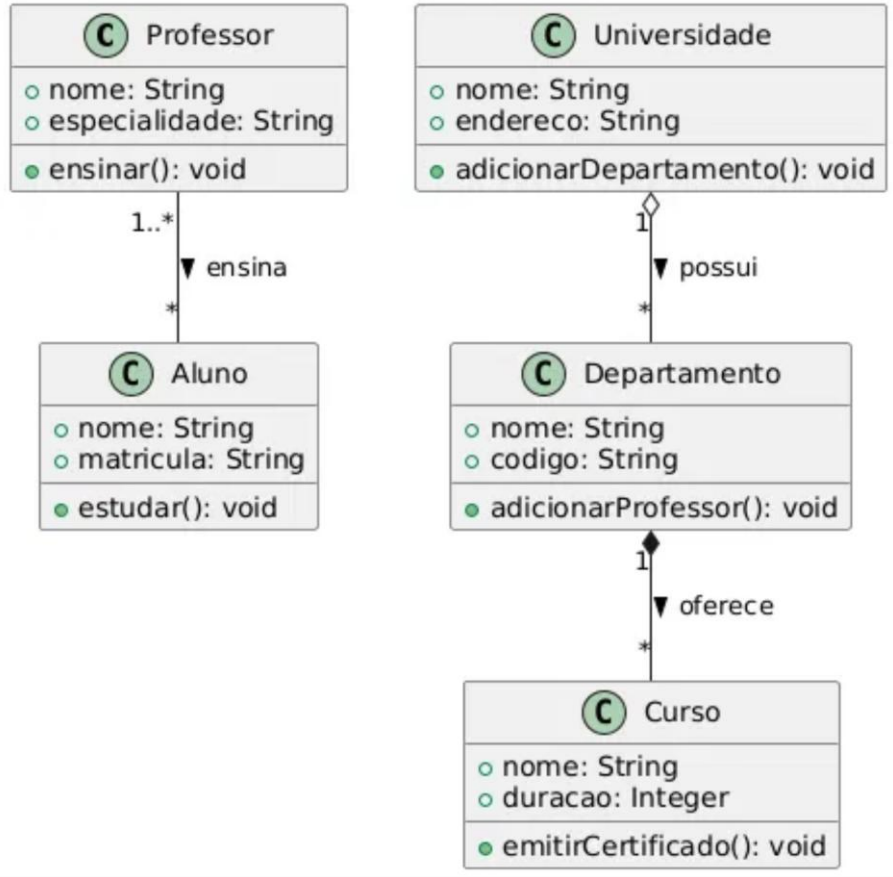
Relação "todo-parte" fraca.

Notação: Diamante vazio ◇

Composição

Relação "todo-parte" forte.

Notação: Diamante preenchido ◆



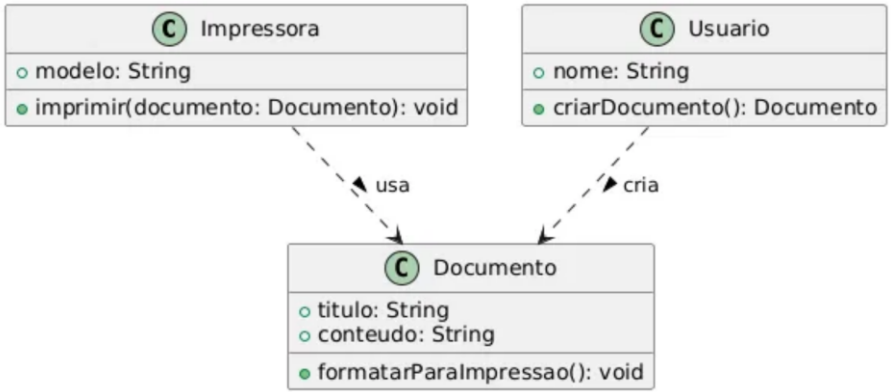
Relacionamento de Dependência

Dependência é um relacionamento onde uma classe **usa** outra classe temporariamente.

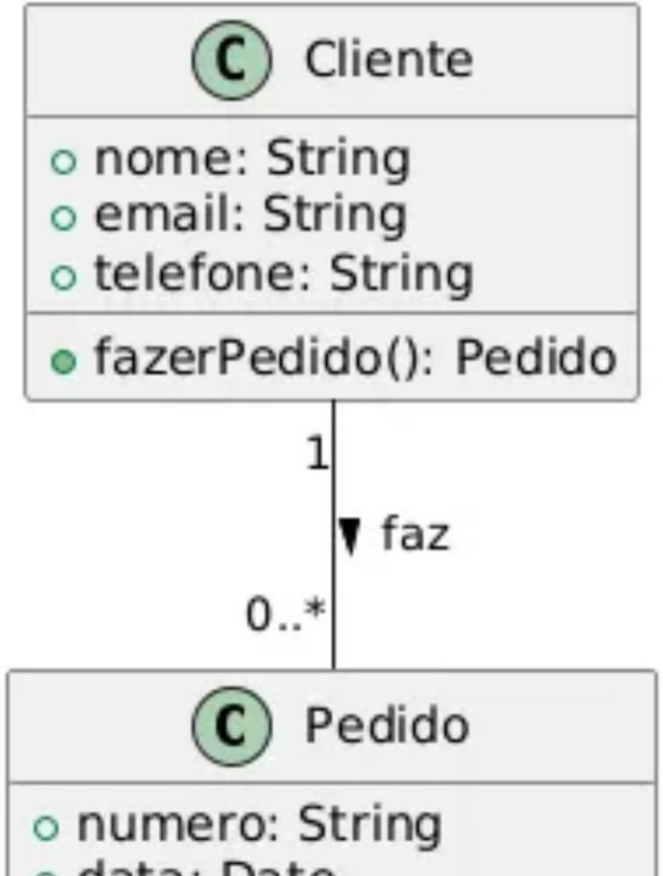
Características principais:

- Relação mais **fraca** entre classes
- Geralmente **unidirecional**
- Alterações na classe alvo afetam a classe dependente

Notação: **Linha pontilhada com seta** -->



Multiplicidade nos Relacionamentos



Multiplicidade define quantos

Exemplo Completo: Sistema de Biblioteca

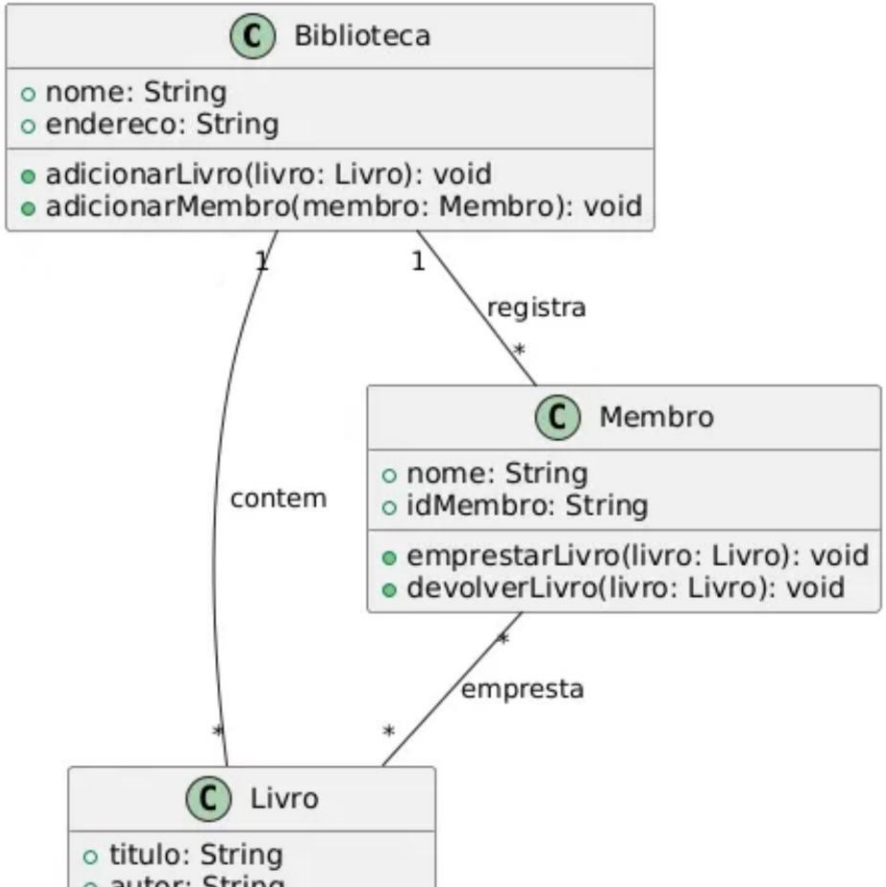
Este exemplo ilustra um **sistema de biblioteca** com classes e relacionamentos.

Elementos do diagrama:

- Classes: **Biblioteca**, **Livro**, **Membro**
- Atributos e métodos com visibilidade
- Relacionamentos com multiplicidade

Relacionamentos presentes:

- **Associação** entre classes



Boas Práticas na Criação de Diagramas de Classe

Recomendações:

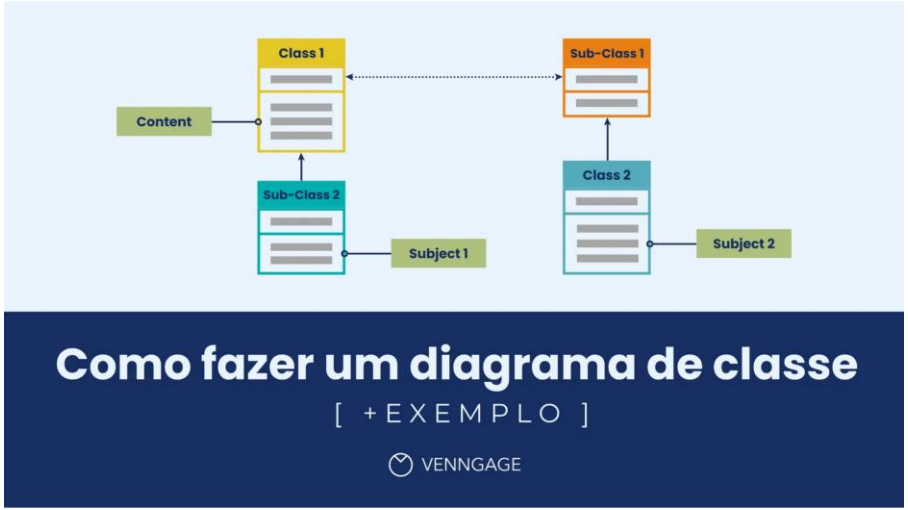
Mantenha o diagrama **simples e claro**

Use **nomes significativos** para classes e atributos

Inclua apenas **atributos e métodos relevantes**

Organize as classes de forma **lógica** no diagrama

Evite **cruzamento de linhas** nos relacionamentos



Conclusão e Recursos Adicionais

Recapitulação

Os diagramas de classe UML são ferramentas essenciais para:

- Modelar a **estrutura** de sistemas
- Visualizar **relacionamentos** entre classes
- Facilitar a **comunicação** entre equipes

Ferramentas: **Lucidchart**, **Draw.io**, **Visual Paradigm**