

Professores

Celso Rodrigo Giusti
Daniel Manoel Filho
Isadora Dias Afonso
Marlon Palata Fanger
Rodrigues

Diagrama de Classe



Objetivos

Ao final da aula, o aluno deverá ser capaz de:

- Entender o que é um **diagrama de classes**
- Identificar **classes, atributos e métodos**
- Aplicar **relacionamentos UML**

Introdução – Diagrama de Classe

Antes de escrever qualquer linha de código, um bom desenvolvedor precisa **entender o problema que está resolvendo**. É nesse ponto que entra o **diagrama de classes**.

O diagrama de classes é uma das ferramentas mais importantes da **UML (Unified Modeling Language)**. Ele permite representar graficamente a estrutura de um sistema orientado a objetos, mostrando como os elementos principais se organizam e se relacionam.

UML

A UML não é uma linguagem de programação. Ela é uma **linguagem de modelagem**, usada para:

- Planejar sistemas
- Facilitar a comunicação entre equipes
- Documentar projetos
- Ajudar na manutenção futura

UNIFIED
MODELING
LANGUAGE™



Papel do Diagrama de Classes

O que o diagrama de classes NÃO mostra:

- Fluxo de telas
- Ordem de execução de métodos
- Interface gráfica

O que ele MOSTRA:

- Classes
- Atributos
- Métodos
- Relacionamentos
- Dependências entre elementos

Conceito de Classe, Atributos e Métodos

Uma **classe** é um modelo, um molde, uma abstração de algo do mundo real ou do próprio sistema. Ela define **como os objetos serão criados**.

Os **atributos** representam as **características** de uma classe. Eles dizem o que o objeto **possui** ou **armazena**.

Os **métodos** representam os **comportamentos** da classe, ou seja, aquilo que o objeto **faz**.

Identificação de Classes em um Problema

Antes de desenhar qualquer diagrama, o primeiro passo é **ler o problema com atenção.**

Exemplo:

Uma biblioteca precisa controlar livros, usuários e empréstimos.

A partir desse texto, conseguimos identificar **substantivos importantes:**

Biblioteca, Livro, Usuário, Empréstimo.

Esses substantivos geralmente se transformam em **classes**.

Relacionamentos entre Classes

Classes raramente existem isoladas. Elas se **relacionam** para formar o sistema.

Associação – É o relacionamento mais simples. Indica que uma classe **conhece** ou **utiliza** outra.

Ex.: Usuário – Empréstimo

Agregação – A agregação representa uma relação “**tem um**”, mas onde os objetos podem existir separadamente.

Ex.: Biblioteca tem livros

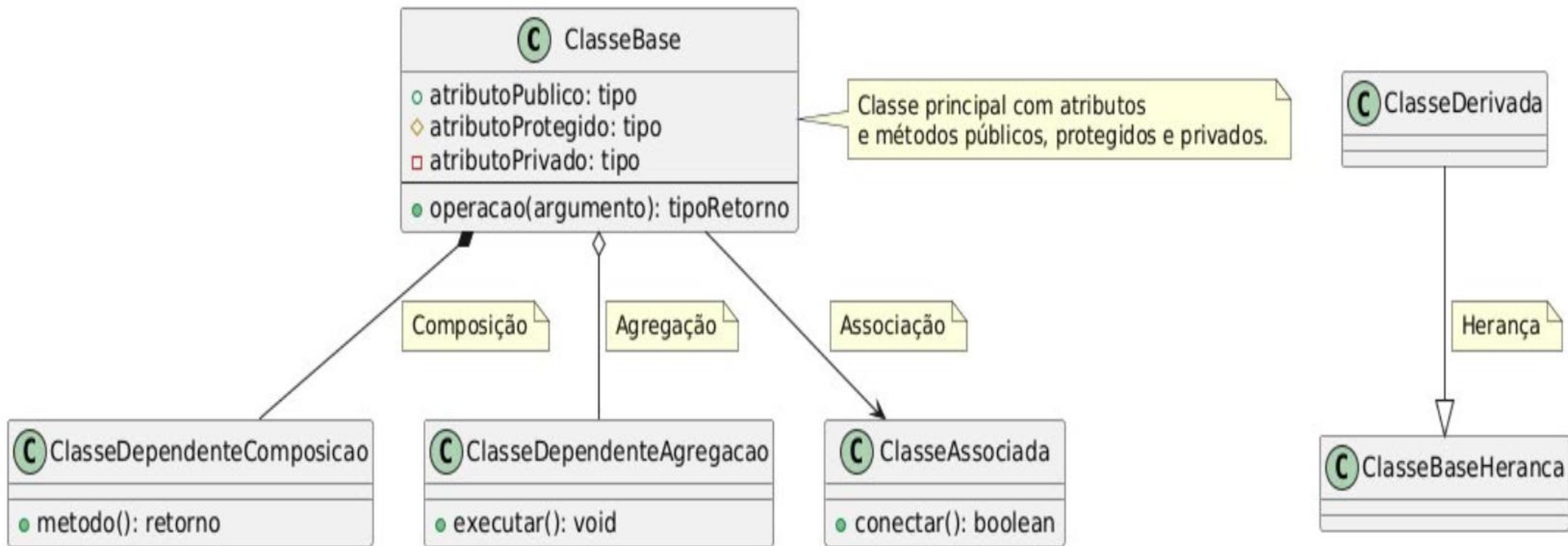
Relacionamentos entre Classes

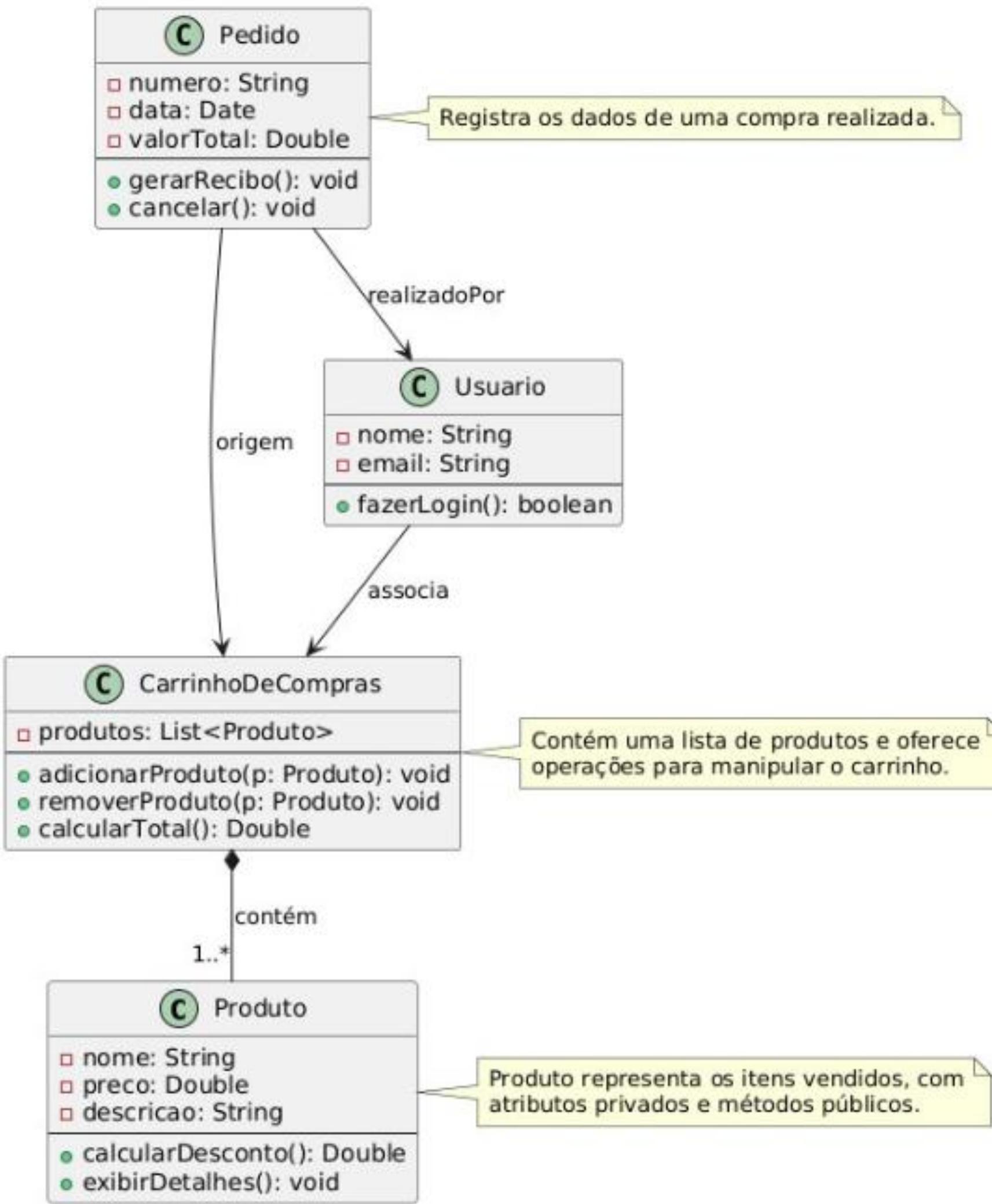
Classes raramente existem isoladas. Elas se **relacionam** para formar o sistema.

Composição – A composição é uma relação mais forte. Um objeto **não faz sentido sem o outro**.

Ex.: Empréstimo não existe sem Livro e Usuário

Relacionamentos entre Classes





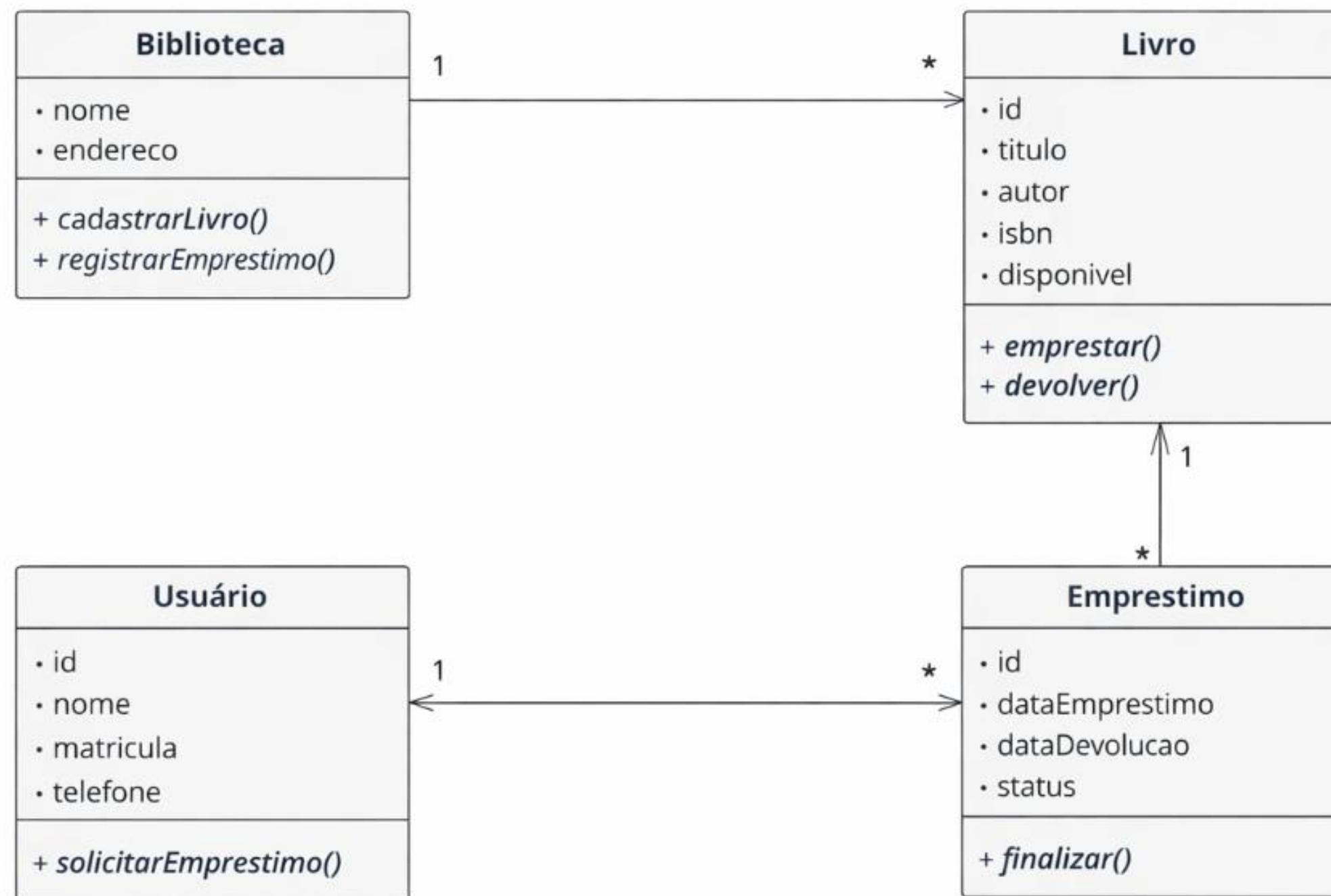
Boas Práticas

- Não criar classes desnecessárias
- Não exagerar na quantidade de métodos
- Manter nomes claros e significativos
- Pensar no **papel da classe**, não na implementação
- Lembrar: o diagrama **não é código**

Erros Comuns de Iniciantes

- Transformar tudo em classe
- Criar métodos que não pertencem à classe
- Confundir atributo com método
- Criar relacionamentos sem sentido lógico
- Pensar no código antes do modelo

Exemplo Biblioteca



Exercício Cinema

Uma rede de cinemas chamada **CineMax** deseja informatizar o controle de seus filmes, salas e sessões. Atualmente, a organização é feita de forma manual, o que gera erros na definição de horários e no controle de público. Antes do desenvolvimento do sistema, é necessário **modelar sua estrutura utilizando um diagrama de classes**.

O sistema deve permitir o cadastro de **filmes, salas, sessões** e **ingressos**. Cada filme pode possuir várias sessões, e cada sessão ocorre em uma sala específica. Uma sala pode receber várias sessões ao longo do dia. Para cada sessão, são gerados vários ingressos, respeitando a capacidade da sala.



Escola SENAI “Italo Bologna”

Av. Goiás, 139 – Itu/SP

Telefone

(11) 2396-1999

Instagram

@senai.itu

Facebook

/senai.itu

Site

<https://sp.senai.br/unidade/itu/>