

Sistema de Biblioteca - Programação Orientada a Objetos

Gabriel Cristian Vivian Somariva¹, Gustavo Schwitzki Peretti¹, Jadson Butzk¹,
Natan Elias Patzlaff³

¹Bacharelado em Ciência da Computação
Instituto Federal Catarinense – Campus Concórdia

1. Introdução

Este relatório apresenta as decisões de modelagem empregadas na construção do sistema de biblioteca, utilizando UML e princípios da orientação a objetos. O foco está em abstração, polimorfismo e nas relações estruturais das classes principais. O diagrama de caso de uso e diagrama de classes podem ser encontrados em PDF junto a este documento.

2. Aplicação dos conceitos de Programação Orientada a Objetos

2.1. Abstração

A abstração está presente nas classes ‘Pessoa’ e ‘Item’, que concentram atributos e métodos comuns, permitindo a especialização de entidades concretas como “Membro”, “Bibliotecário”, “Administrador”, “Livro” e “Ebook”. Métodos abstratos, como `mostrar_dados()` e `mostrar_detalhes()`, garantem que cada subclasse implemente sua própria versão, promovendo flexibilidade e reutilização.

2.2. Polimorfismo

O polimorfismo ocorre via herança das classes abstratas. Os métodos abstratos podem ser sobreescritos, possibilitando que operações genéricas sejam realizadas conforme regras específicas de cada especialização. Por exemplo, operações genéricas de consulta e exibição de dados podem assumir comportamentos distintos dependendo se o objeto é um livro físico ou um *ebook*, ou se o usuário é membro, bibliotecário ou administrador.

2.3. Relações entre classes

A modelagem utilizou agregação e composição nas seguintes configurações:

- Na constituição da classe ‘Biblioteca’, que armazena as instâncias das classes principais por meio de composição (por exemplo, um ‘Usuario’ **pertence** à ‘Biblioteca’, e não existe sem ela).
- Na relação de composição entre ‘Multa’ e ‘Emprestimo’ (o empréstimo origina a multa, portanto, toda multa tem um empréstimo associado).
- Na relação de agregação entre ‘Reserva’ e ‘Emprestimo’. Isso se justifica pelo fato de que um empréstimo pode vir de uma reserva prévia. No entanto, uma reserva pode existir sem a necessidade de um empréstimo associado (por exemplo, uma reserva cancelada).
- Na participação do ‘Item’ em ‘Reserva’ e ‘Emprestimo’. Essa relação se classifica como agregação, visto que um ‘Item’ existe independentemente da reserva ou do empréstimo.

- Na participação do ‘Usuario’ em ‘Reserva’ e ‘Emprestimo’, semelhante à relação anterior. A relação é de agregação, pois um ‘Usuario’ pode existir sem uma reserva ou empréstimo associado.

Estes vínculos indicam que empréstimos dependem dessas entidades, sem comprometer a existência independente dos objetos vinculados. A composição pode ser adotada futuramente, caso cenários exijam dependência total entre entidades, como reservas vinculadas ao ciclo de vida de empréstimos.

2.4. Especialização e métodos

As classes especializadas do sistema são ‘Livro’ e ‘Ebook’, que possuem atributos em comum com a classe ‘Item’. Além disso, a herança foi utilizada na criação das classes que representam os atores, de modo que cada uma recebe os atributos de ‘Usuario’:

- Membro: operar empréstimos, devoluções, renovações, reservas e multas.
- Bibliotecário: gerenciar cadastro, atualização e exclusão de itens.
- Administrador: registrar, consultar e remover membros, além de visualizar movimentações.

A abordagem facilita a extensão do modelo, incluindo novos tipos de item ou funções para atores existentes.

2.5. Facilidade de Expansão

A separação clara das entidades e o uso correto de herança asseguraram flexibilidade e fácil manutenção. Novos tipos de itens ou atores podem ser incorporados sem impacto significativo no núcleo do sistema.

3. Conclusão

O sistema foi projetado seguindo boas práticas da orientação a objetos, com foco em modularidade, reaproveitamento e clareza nas responsabilidades de cada classe e ator. A estrutura favorece evoluções futuras sem perda de robustez ou legibilidade.