

Análise de Sistemas

Prof. Romário Nzenguele da Silva



Licenciatura em Informática

Sobre o Documento



Material de apoio **gratuito**, desenvolvido exclusivamente para os estudantes do **3º Ano da Licenciatura em Informática**;



O conteúdo, sempre que necessário, será atualizado para refletir as mudanças e melhorias no decorrer do curso. Cada atualização será devidamente versionada, permitindo o fácil acompanhamento das revisões;



Este recurso é um **resumo das aulas**, essencial para apoiar no aprendizado da disciplina.

O **repositório da disciplina** (contendo exercícios, conteúdos detalhados, programa ...) será reencaminhado por email;



Sugestões de melhoria podem ser apresentadas pelo email que se encontra no rodapé.

Conteúdo

2. Modelo de objectos

2.1. Objecto e classe

2.2. Atributo

2.3. Associações

2.4. Operações

2.5. Abstracções particulares

Objectivos

- ✓ Modelar um sistema com base na Orientação a Objecto;
- ✓ Diferenciar objetos e classes;
- ✓ Definir atributos para representar propriedades relevantes dos objetos;
- ✓ Identificar e descrever associações;
- ✓ Definir operações que representem; corretamente o comportamento dos objetos;
- ✓ Criar abstrações para reutilizar comportamentos, proteger dados e facilitar a manutenção do modelo



2. Modelo de Objetos

Suporte: Notação UML

Projecto Referência:

Sistema de biblioteca digital com suas principais entidades e relacionamentos.

Nota: O modelo pode ser complementado com mais classes, atributos, operações, ...

Suporte:

- ✓ Notação UML: **Diagrama de Classes**
- ✓ [Ferramenta]: [Mermaid](#)

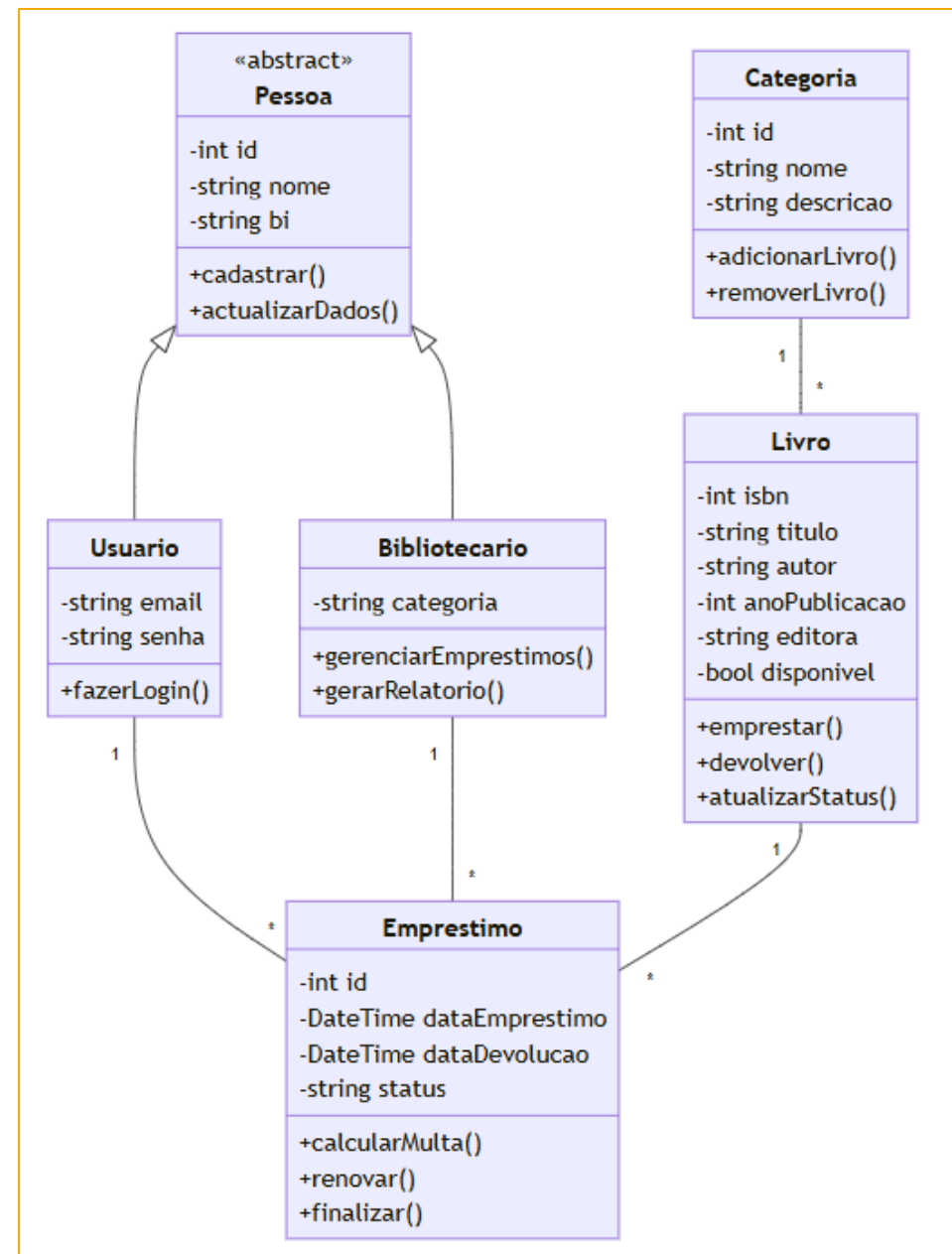


Diagrama de Classes

**O essencial sobre diagramas de classe.
Leitura recomendada:**

https://github.com/rns-yoda/recursos_comuns/blob/main/uml/uml_diagrama_classe.md

2.1 Objecto e Classe

- **Objecto:** Instância concreta com estado e comportamento próprios;
- **Classe:** define uma estrutura e comportamentos comuns para os objectos.

A distinção objeto-classe permite modelar o mundo real de forma mais precisa e flexível.

Analogia:

Classe



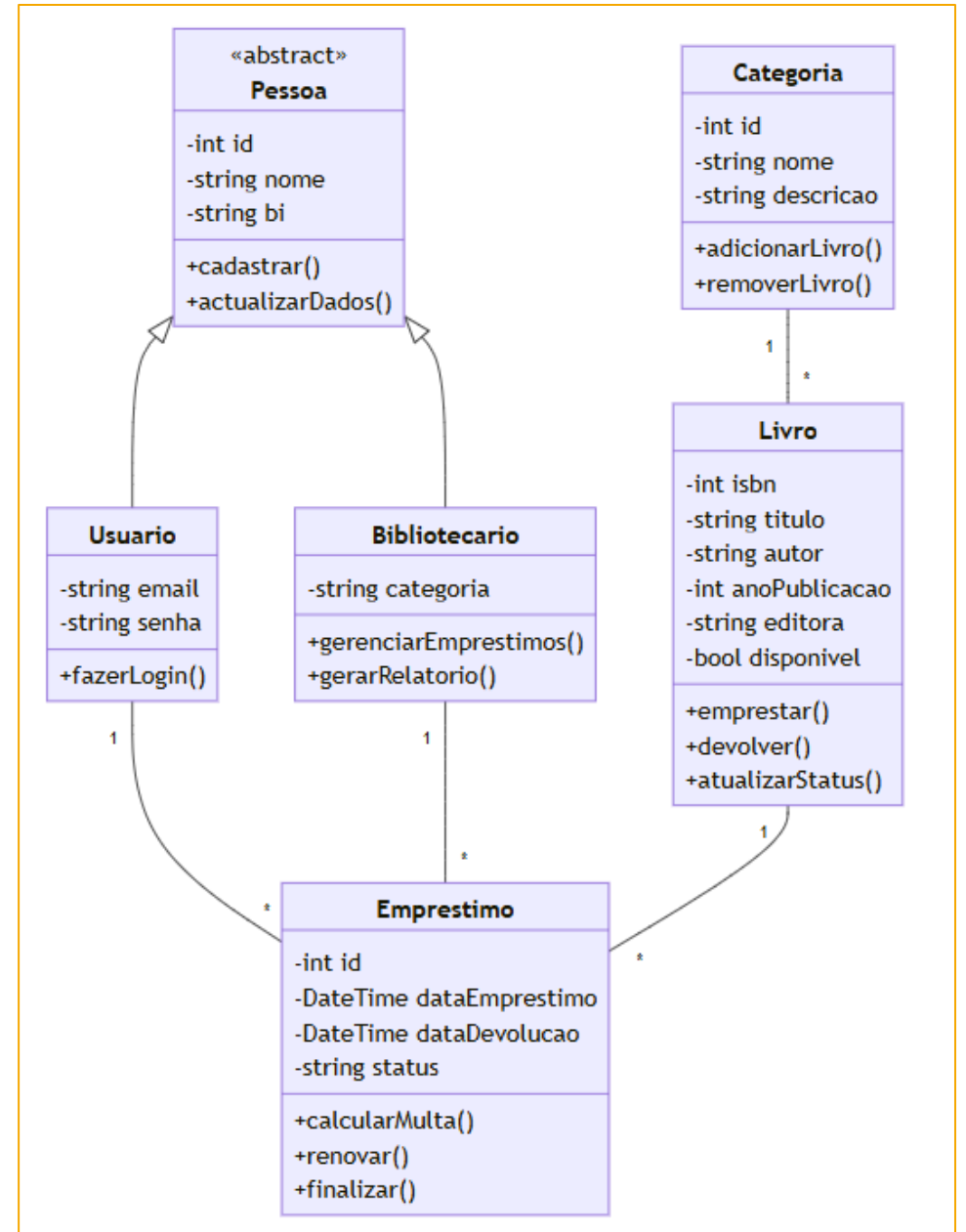
Objectos



Projecto Referência:

Sistema de biblioteca digital com suas principais entidades e relacionamentos.

- Cada **classe** representa um conjunto de objetos com características similares;
- Por exemplo, a classe Usuario pode ter várias instâncias (objetos) representando diferentes usuários do sistema;
- Foram modeladas 6 classes principais: Pessoa, Usuario, Livro, Emprestimo, Categoria e Bibliotecario.



2.2 Atributo

- Características que descrevem um objeto;
- Representam o estado do objeto.

Atributos permitem capturar as propriedades essenciais dos objetos, facilitando a representação de entidades complexas.

Ex: designação, peso, recheio, sabor, formato, etc...

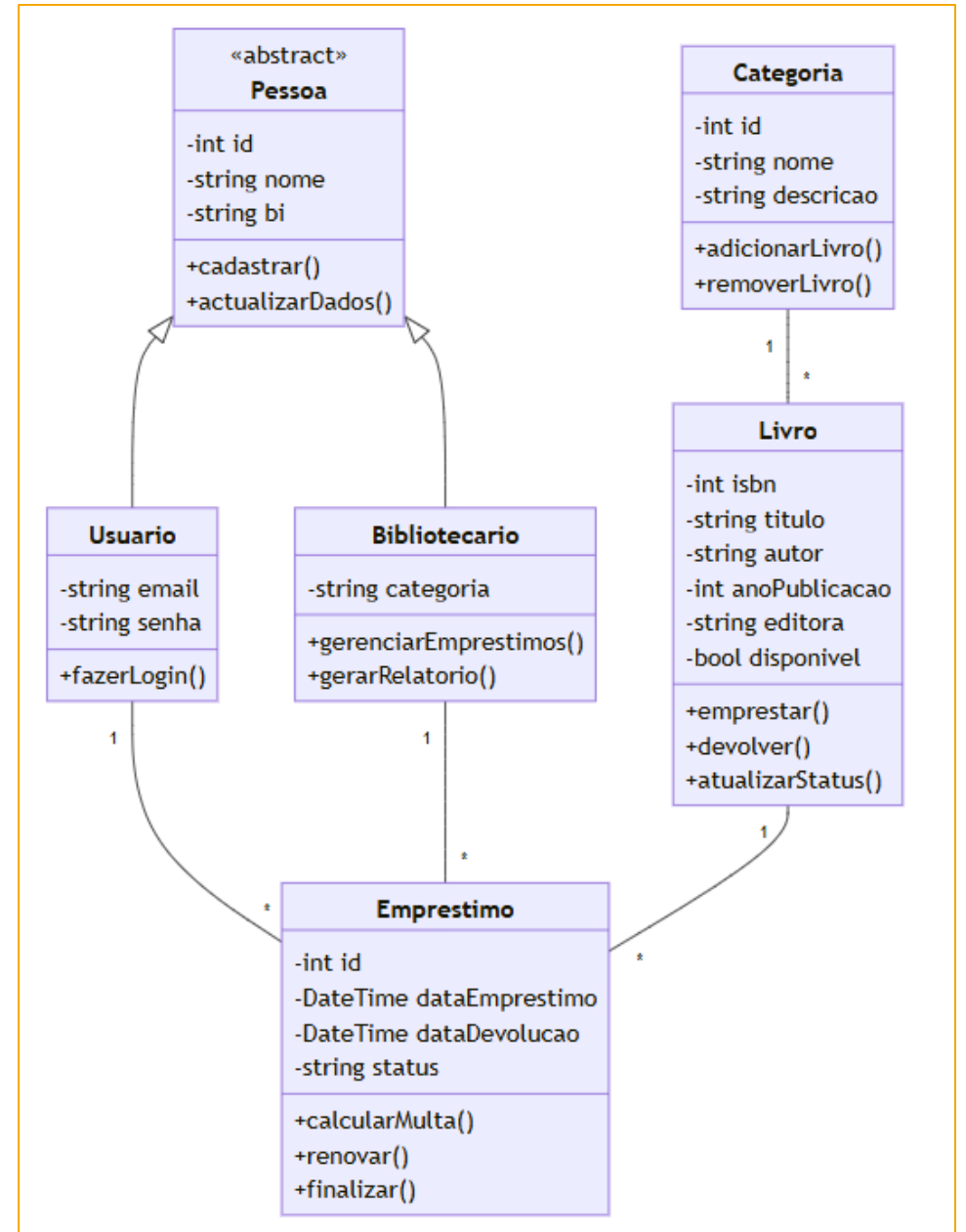


Imagem gerada por IA.

Projecto Referência:

Sistema de biblioteca digital com suas principais entidades e relacionamentos.

- Cada classe possui **atributos** específicos que descrevem suas características.
- Por exemplo, na classe **Livro**:
 - ✓ isbn: identificador único;
 - ✓ titulo: nome do livro;
 - ✓ autor: nome do autor;
 - ✓ anoPublicacao: ano em que foi publicado;
 - ✓ editora: nome da editora;
 - ✓ disponivel: status de disponibilidade



2.3 Associação

- Relação entre classes ou objetos;
- Tipos: um-para-um, um-para-muitos, muitos-para-muitos.

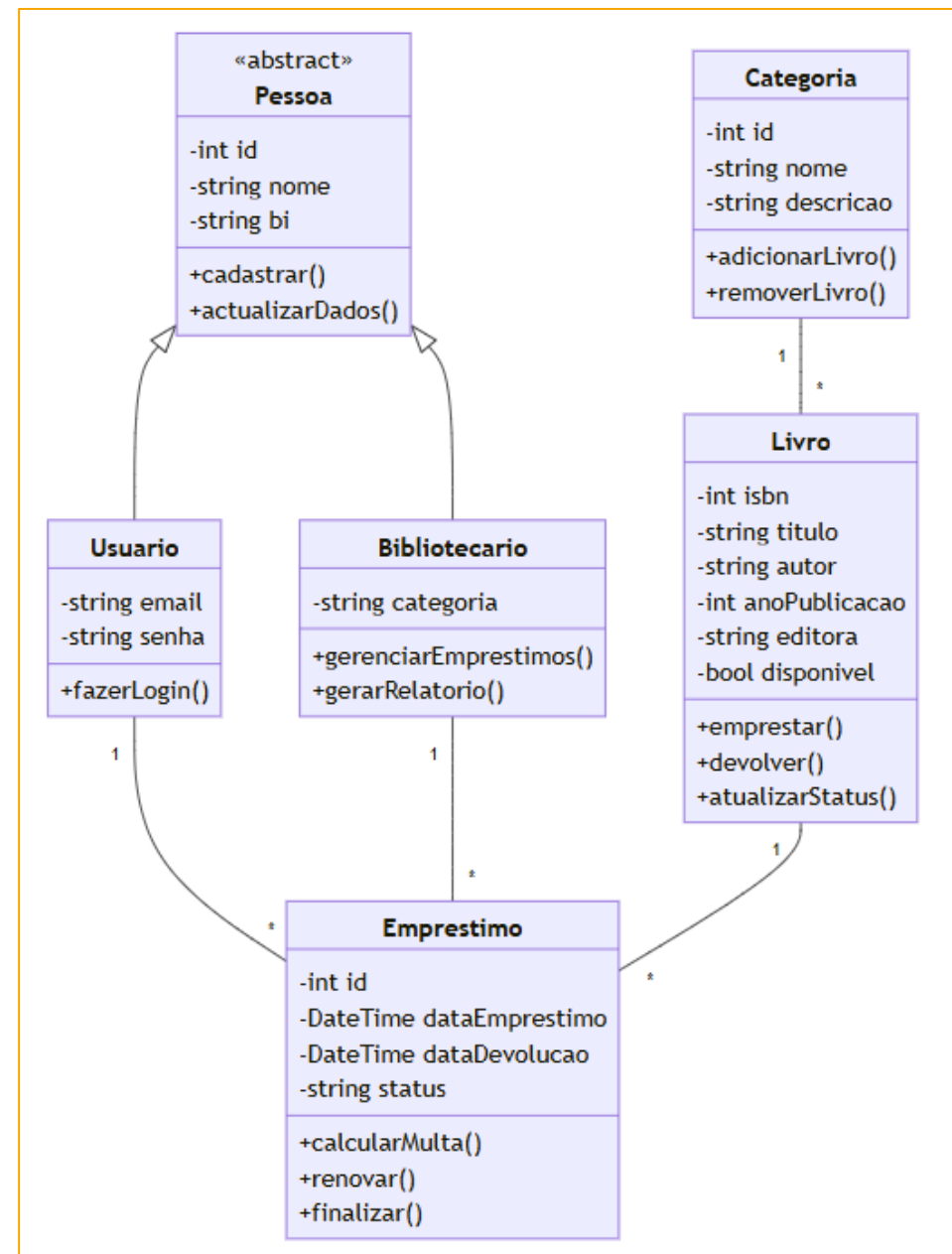
Associações modelam as interações entre objetos, refletindo a complexidade das relações no mundo real.

Projecto Referência:

Sistema de biblioteca digital com suas principais entidades e relacionamentos.

O diagrama mostra várias **associações** entre as classes:

- Usuario -- Empréstimo: um usuário pode ter múltiplos empréstimos;
- Livro -- Empréstimo: um livro pode estar em vários empréstimos (em momentos diferentes);
- Categoria -- Livro: uma categoria pode ter múltiplos livros;
- Bibliotecario -- Empréstimo: um bibliotecário pode gerenciar múltiplos empréstimos.



2.4 Operações

- Ações ou comportamentos que um objeto pode realizar;
- Também chamadas de métodos em POO.

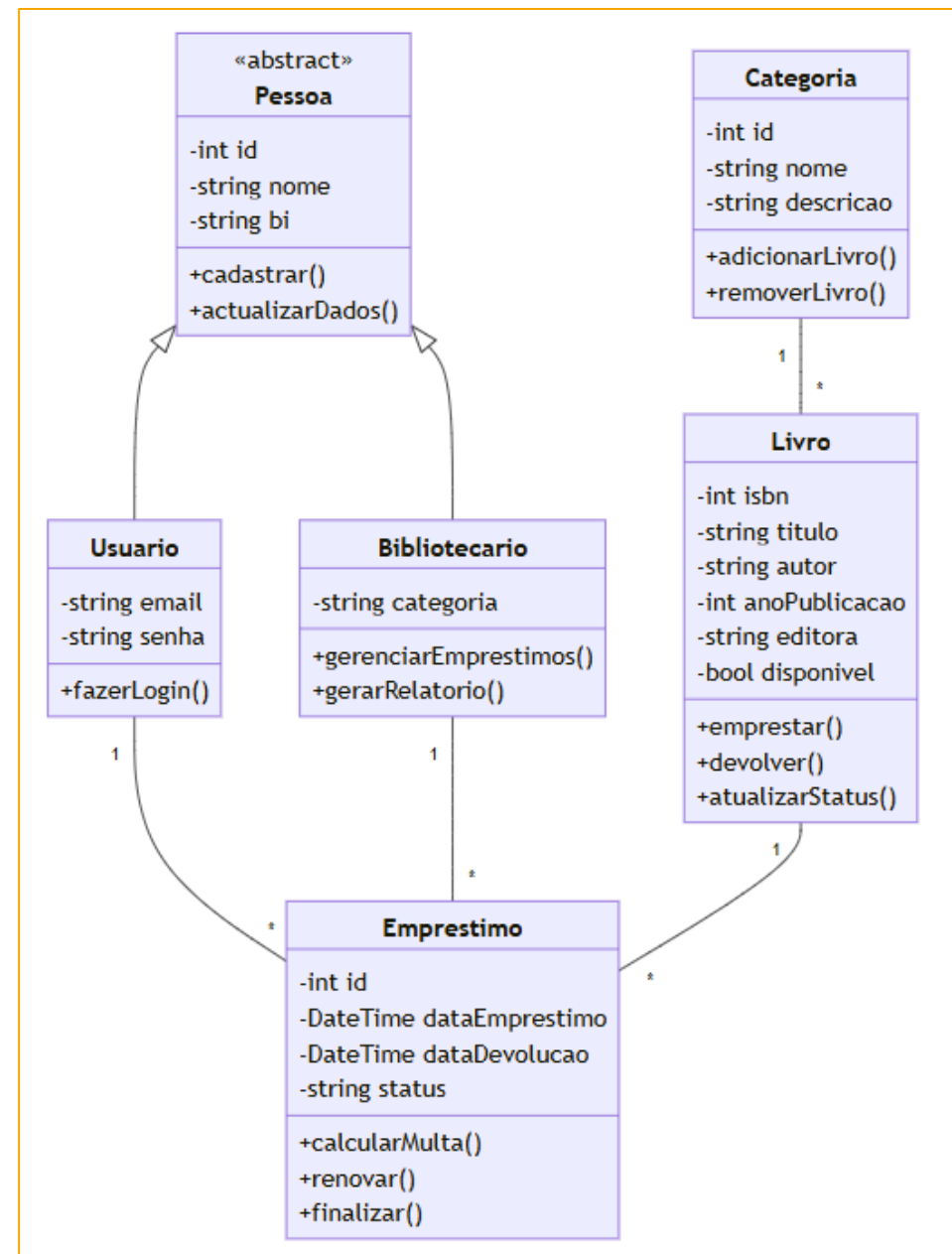
Operações encapsulam o comportamento, promovendo modularidade e facilitando a manutenção do sistema.

Projecto Referência:

Sistema de biblioteca digital com suas principais entidades e relacionamentos.

Cada classe possui **operações** (métodos) que representam seu comportamento:

- **Usuario:** fazerLogin();
- **Livro:** emprestar(), devolver(), atualizarStatus();
- **Emprestimo:** calcularMulta(), renovar(), finalizar().



2.5 Abstrações Particulares

- **Herança:** Criar novas classes baseadas em existentes;
- **Polimorfismo:** Objetos respondem à mesma mensagem de formas distintas;
- **Encapsulamento:** Ocultar detalhes internos.

Estas abstrações promovem reutilização de código, flexibilidade e robustez no design de sistemas.

Projecto Referência:

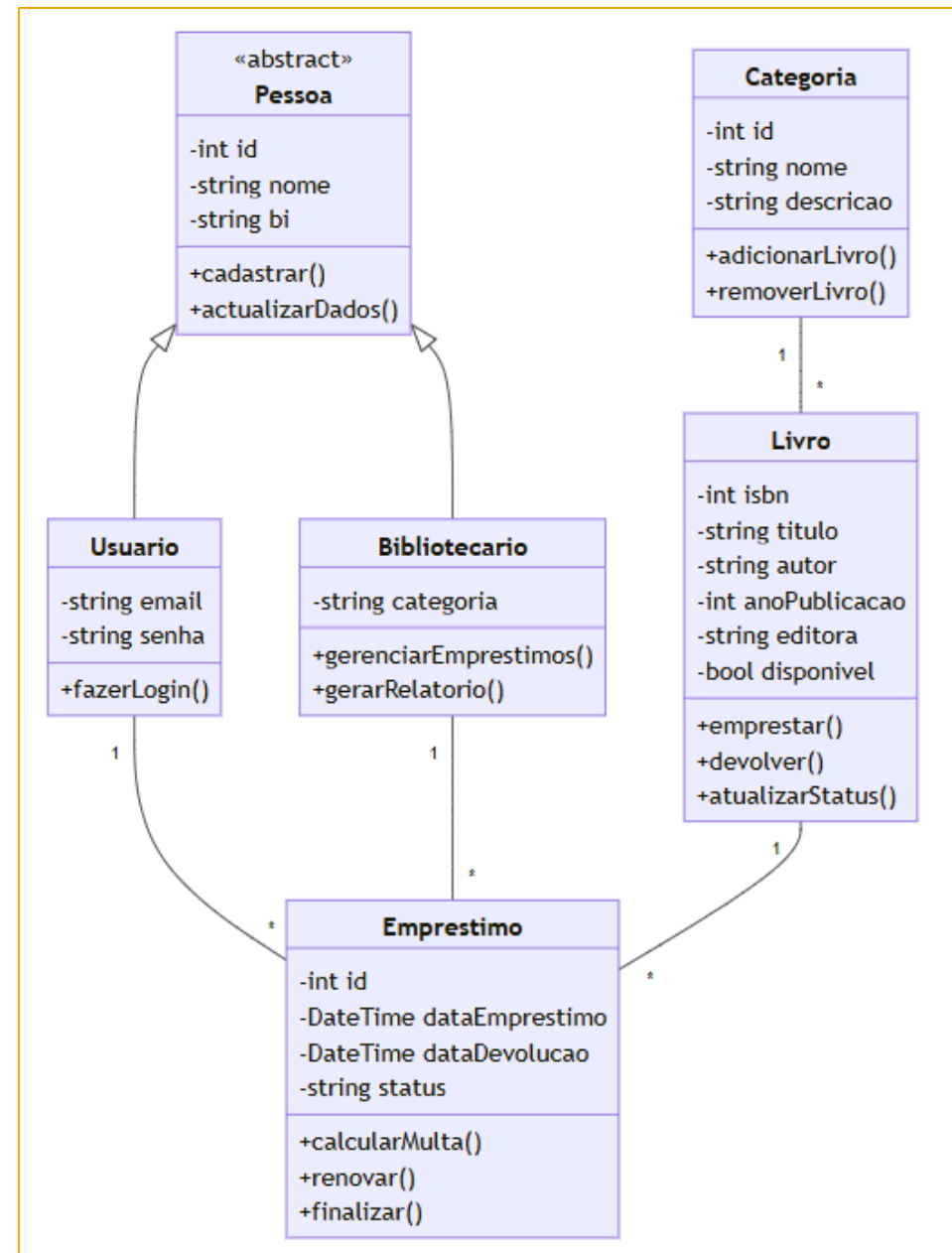
Sistema de biblioteca digital com suas principais entidades e relacionamentos.

❑ Herança

- Classe abstrata **Pessoa** é superclasse de:
 - **Usuario** e **Bibliotecario**;
 - Ambas herdam atributos comuns (id, nome, bi) e comportamentos base.

Nota:

Mais detalhes sobre **classes abstratas** em: https://github.com/rns-yoda/recursos_comuns/blob/main/POO/classes_metodos_abstratos.md



Projecto Referência:

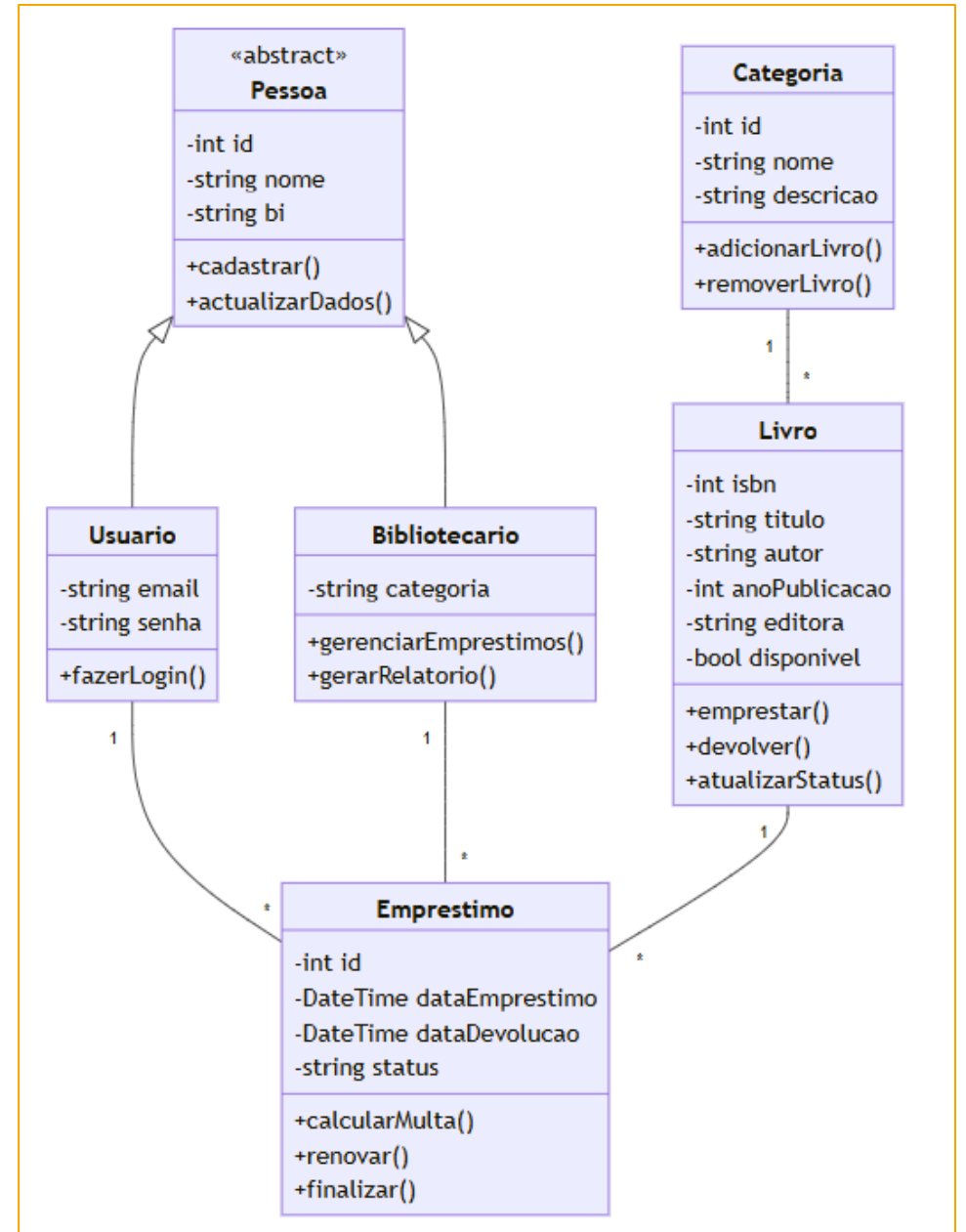
Sistema de biblioteca digital com suas principais entidades e relacionamentos.

❑ Polimorfismo:

- Métodos abstratos: em Pessoa:
 - ✓ cadastrar() e atualizarDados() serão implementados de forma diferente em **Usuario** e **Bibliotecario**

Nota:

Mais detalhes sobre **métodos abstratos** em: https://github.com/rns-yoda/recursos_comuns/blob/main/POO/classes_metodos_abstratos.md



Consolidação

O Modelo de Objetos fornece uma base sólida para análise e design de sistemas, permitindo representações mais fiéis e flexíveis do mundo real em software.

...

Sequência -> **M03**
