

Questão 1. Com base no diagrama UML de um sistema de biblioteca responda:

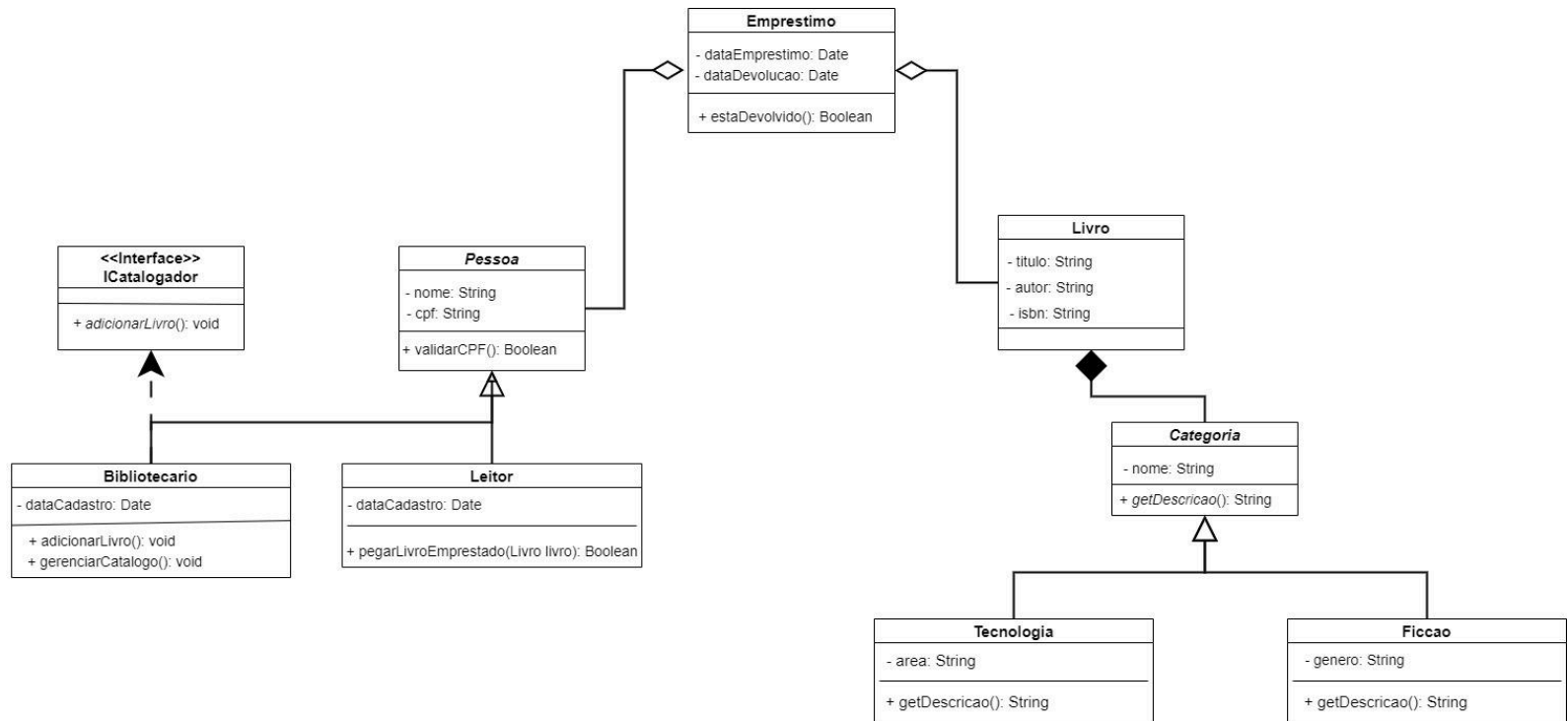


Figura 1. Diagrama de um Sistema de Biblioteca.

- Quais os relacionamentos entre classes? Cite todos os relacionamentos presentes.
- Especifique as multiplicidades das associações entre as classes (ex: 1..*, 0..1).
- Quais classes estão utilizando herança?
- Quais conceitos de Programação Orientada a Objetos estão presentes nessa modelagem?
- Considere o relacionamento entre a classe Pessoa e Empréstimo. Está apropriado?
- Considere o relacionamento entre a classe Livro e Categoria. Está apropriado?
- O sistema precisa ser modificado para representar também livros virtuais (eBooks). Como você alteraria o diagrama para atender ao novo requisito? Indique se deve haver herança, composição ou outro tipo de relacionamento. Justifique suas escolhas.

Questão 2. Você está desenvolvendo um sistema de gestão de veículos para uma empresa de locação. O sistema precisa lidar com diferentes tipos de veículos, como:

- **Carro:** Pode ser um veículo de passeio ou uma **Picape**. A Picape, por sua vez, pode desempenhar tanto o papel de veículo de passeio quanto de veículo de carga.
- **Moto:** Representa um veículo de duas rodas.
- **Caminhão:** Veículo de grande porte destinado ao transporte de cargas.

Todos os veículos compartilham atributos básicos como **marca**, **modelo** e **ano**, além de comportamentos comuns como **ligar()**, **desligar()**, **acelerar()** e **frear()**.

Alguns veículos possuem comportamentos específicos:

- O **Carro** possui o método adicional **abrirPortas()**.
- A **Moto** é capaz de realizar a ação **empinar()**.
- O **Caminhão** pode **carregar()** e **descarregar()** cargas.

Dado essas informações, faça:

- Usando os conceitos de herança, polimorfismo e interfaces estudados até o momento, modele uma solução para esse cenário em UML.
- Implemente a solução em Java, respeitando a modelagem proposta.

Questão 3. Observe o código abaixo.

```
class ContaBancaria {  
    public double saldo;  
  
    public void sacar(double valor) {  
        saldo = saldo - valor;  
    }  
  
    public void depositar(double valor) {  
        saldo = saldo + valor;  
    }  
}
```

Figura 2. Classe ContaBancaria.

Com base no código da Figura 2, responda:

- A. Quais princípios da programação orientada a objetos não estão sendo seguidos?
- B. O que poderia ser alterado na classe ContaBancaria?

Questão 4.

- A. Explique o conceito de **polimorfismo** em Programação Orientada a Objetos.
- B. Quando e por que um método da classe pai pode ser sobrescrito em uma classe filha? Forneça um exemplo prático.

Questão 5. Diferenças entre classe abstrata e interface.

- A. Qual é a principal diferença na herança de uma classe abstrata em comparação à implementação de uma interface?
- B. Em que situações você utilizaria uma **classe abstrata** em vez de uma **interface**? E vice-versa? Dê exemplos de cenários para cada caso.