1) Reflexão sobre a aula e o entendimento de trechos de código

Este exercício sugere que reflitamos sobre a importância de **compreender** o código além de apenas copiá-lo. Isso destaca a necessidade de entender os **conceitos fundamentais** de programação, como algoritmos, estrutura de dados e paradigmas de programação. Como futuro desenvolvedor, você será responsável por fornecer respostas sobre **por que** uma solução foi implementada de determinada forma e se faz sentido no contexto do sistema.

2) Escolha do Paradigma de Programação

Resposta: Programação Orientada a Objetos (POO)

Justificativa:

- **Estrutura e Interação**: O sistema envolve **entidades** como livros, autores, usuários, empréstimos e devoluções. A POO é ideal para modelar essas entidades como objetos com seus atributos e métodos.
- **Reuso**: POO promove a **reutilização de código** por meio de herança, o que facilita a expansão do sistema, como adicionar novas funcionalidades de forma modular.
- **Facilidade de Evolução**: A POO facilita a evolução do sistema por meio do polimorfismo e encapsulamento. Novas funcionalidades podem ser adicionadas sem afetar outras partes do código.
- **Eficiência**: A POO organiza o código de maneira modular, permitindo que diferentes partes do sistema interajam de forma clara e escalável.

3) Diagrama de Classes (Primeira Versão)

Neste passo, você deve criar um **diagrama de classes** básico, identificando as principais **entidades** (ou classes) do sistema e seus atributos.

Classes sugeridas:

- **Livro**: `titulo (String)`, `autor (Autor)`, `genero (String)`, `status (boolean)`
- **Autor**: `nome (String)`, `nacionalidade (String)`, `obrasPublicadas (List<Livro>)`
- **Usuário**: `nome (String)`, `idade (int)`, `historicoEmprestimos (List<Emprestimo>)`

- **Empréstimo**: `dataRetirada (Date)`, `dataDevolucao (Date)`, `livro (Livro)`, `usuario (Usuario)`

Relacionamentos:

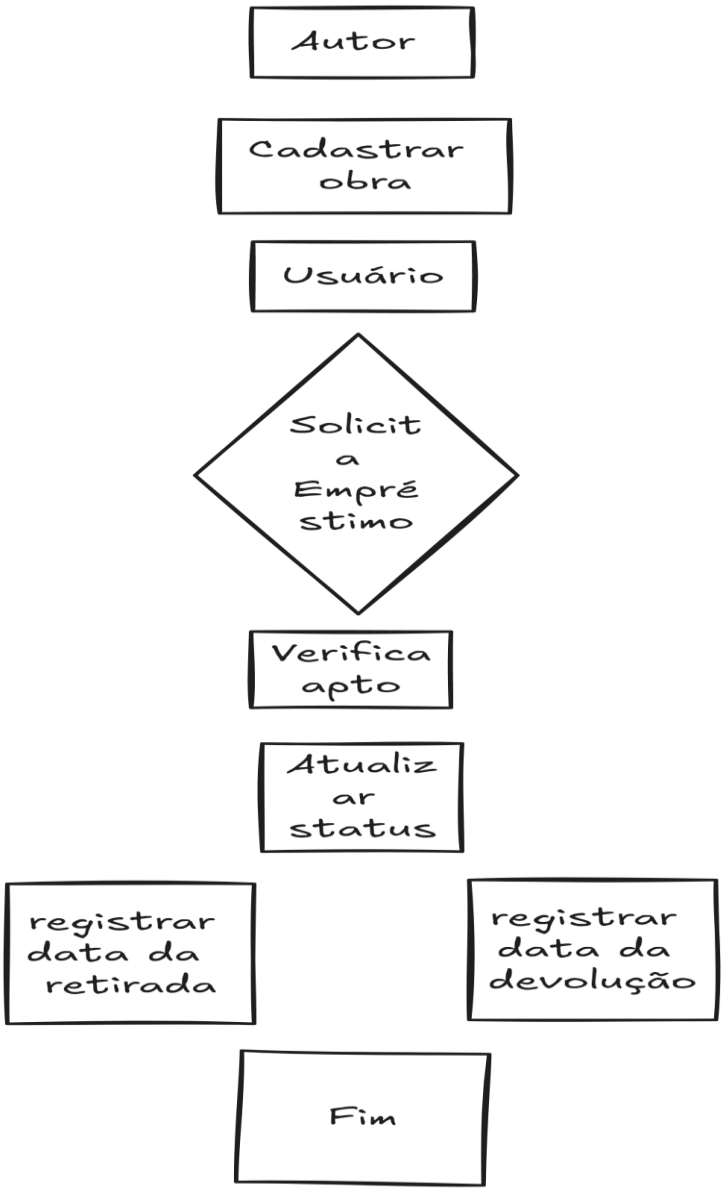
- **Autor** tem um relacionamento de **1 para muitos** com **Livro** (um autor pode ter várias obras).
- **Usuário** tem um relacionamento de **1 para muitos** com **Empréstimo** (um usuário pode ter vários empréstimos).
- **Livro** tem um relacionamento de **1 para 1** com **Autor** (um livro tem um único autor).

4) Diagrama de Classes com Métodos

Neste diagrama, devemos adicionar os métodos relacionados aos fluxos de empréstimo, devolução e cadastro.

Métodos sugeridos:

- **Usuário**:
  - `solicitarEmprestimo(Livro livro)`
  - `verificarApto()`: Verifica se o usuário pode realizar o empréstimo.
  - `devolverLivro(Livro livro)`
- **Livro**:
  - `atualizarStatus(boolean status)` (muda o status do livro para disponível ou não).
- **Empréstimo**:
  - `registrarDataRetirada()`
  - `registrarDataDevolucao()`
- **Autor**:
  - `cadastrarObra(Livro livro)`


Autor
Cadastrar obra
Usuário
Solicit a Empré stimo
Verifica apto
Atualiz ar status
registrar data da retirada
registrar data da devolução
Fim