

## Sistema de Gerenciamento de Biblioteca

**Objetivo:** Criar um sistema de gerenciamento de biblioteca que permita gerenciar livros, usuários e empréstimos.

### Questões:

- **Diagrama de Classes UML:**
  - Você deve criar um diagrama de classes UML que inclua as seguintes classes:
    - Livro (Book)
    - Usuário (User)
    - Empréstimo (Loan)
    - Biblioteca (Library)
  - Cada classe deve incluir atributos e métodos relevantes.
  - O diagrama de classes deve ser feito em mermaid no arquivo *resolucao/diagramas.md*
- **Implementação em Python:**
  - Os alunos devem traduzir o diagrama de classes UML para código Python.
  - Implementar as classes com atributos, métodos e relacionamentos adequados.
  - Usar propriedades na declaração dos atributos e type hints na declaração dos métodos não é obrigatório, mas agrega nota.
  - Siga as regras e requisitos do sistema abaixo:

### Requisitos do Sistema:

#### Classe Livro (Book):

##### Atributos:

- título (str)
- autor (str)
- isbn (str)
- status (str) [Disponível, Emprestado]

##### Métodos:

- **disponivel()**: Retornar True se o livro estiver disponível, False caso contrário.
- **emprestar()**: Alterar o status do livro para 'Emprestado'.
- **devolver()**: Alterar o status do livro para 'Disponível'.

#### Classe Usuário (User):

#### Atributos:

- nome (str)
- id\_usuario (int)
- telefone (str)

#### Métodos:

- **exibir\_informacoes()**: Exibir informações do usuário.
- **emprestar\_livro(library, livro)**: Realizar um empréstimo de um livro.  
Deve verificar a disponibilidade do livro antes de realizar o empréstimo.

### Classe Empréstimo (Loan):

#### Atributos:

- usuario (User)
- livro (Book)
- data\_emprestimo (str) [Formato 'YYYY-MM-DD']
- data\_devolucao (str) [Formato 'YYYY-MM-DD']

#### Métodos:

- **confirmar\_emprestimo()**: Confirmar o empréstimo e alterar o status do livro para 'Emprestado'.
- **cancelar\_emprestimo()**: Cancelar o empréstimo e alterar o status do livro para 'Disponível'.

### Classe Biblioteca (Library):

#### Atributos:

- lista\_usuarios (list)
- lista\_livros (list)
- lista\_emprestimos (list)

#### Métodos:

- **registrar\_usuario(nome, telefone)**: Registrar um novo usuário no sistema.
- **adicionar\_livro(titulo, autor, isbn)**: Adicionar um novo livro no sistema.
- **emprestar\_livro(usuario, livro, data\_emprestimo, data\_devolucao)**: Realizar um novo empréstimo se o livro estiver disponível.
- **cancelar\_emprestimo(emprestimo)**: Cancelar um empréstimo existente.

### Regras para Implementação:

1. Verificação de Disponibilidade: Ao emprestar um livro, o método `emprestar_livro` da classe `Library` deve verificar se o livro está disponível.
2. Validação de Datas: As datas de empréstimo e devolução devem estar no formato 'YYYY-MM-DD' e a data de devolução deve ser posterior à data de empréstimo.
3. Manutenção de Estado: Ao confirmar um empréstimo, o status do livro deve ser alterado para 'Emprestado'. Ao cancelar um empréstimo, o status deve ser alterado para 'Disponível'.
4. Informações Consistentes: Métodos que exibem informações (como `exibir_informacoes` na classe `User`) devem retornar dados formatados de maneira clara e concisa.
5. Gestão de Listas: A classe `Library` deve manter listas consistentes de usuários, livros e empréstimos, garantindo que cada operação de adição ou remoção seja refletida corretamente nas listas correspondentes.