

**Nome: Marco Antônio Dias Felipi**

**Professora: Vanessa Lago Machado**

## **1. Introdução**

O sistema de biblioteca desenvolvido tem como objetivo gerenciar o empréstimo de livros de forma organizada. O sistema, desenvolvido na linguagem Java, contém diversas classes relacionadas, sendo elas:

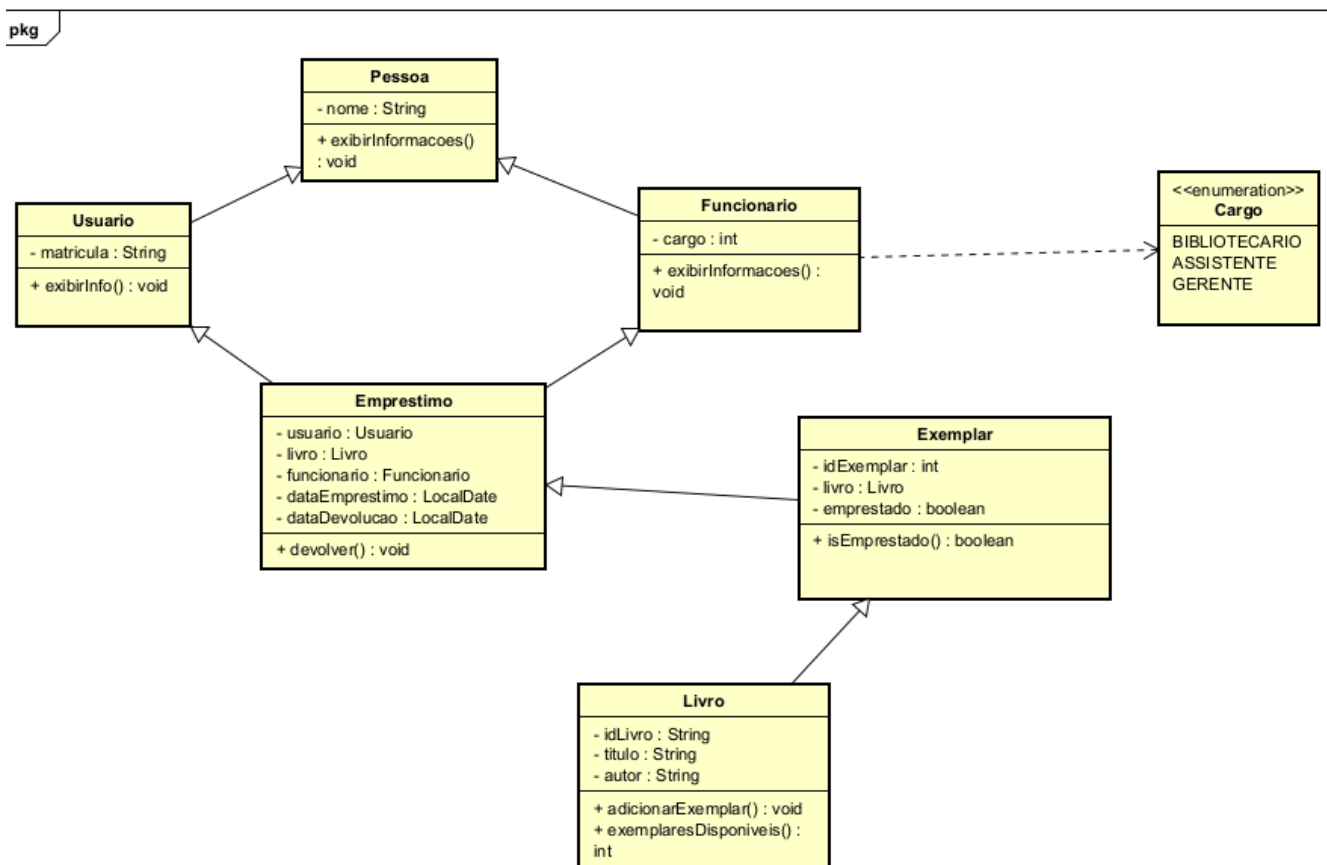
- a. Pessoa:** Classe abstrata representando uma pessoa genérica (usuário ou funcionário).
- b. Usuario:** Classe herdeira de Pessoa, representa um usuário da biblioteca que retira livros.
- c. Funcionario:** Classe que herda de Pessoa e representa os funcionários da biblioteca, com um cargo específico (definido por um Enum).
- d. Cargo:** Enum que define o cargo do funcionário.
- e. Livro:** Classe que representa um livro, contendo informações de título, autor e ID.
- f. Exemplar:** Classe que representa um exemplar específico de cada livro. Cada livro pode conter vários exemplares, e um exemplar pertence a somente um livro.
- g. Emprestimo:** Classe que representa o empréstimo de um exemplar, registrando

## **2. Pilares de TOO**

- a. Encapsulamento:** os atributos de todas as classes são privados, utilizando os métodos get e set para manipular cada um desses dados. Isso garante segurança e integridade aos dados que não podem ser acessados diretamente.

- b. **Abstração:** a classe Pessoa é uma classe abstrata que define um padrão de comportamento para suas classes filhas (Usuario e Funcionario). Assim, garante-se que as classes Usuario e Funcionario terão um nome e o método `exibirInformações` (também abstrato) mas com implementações específicas.
- c. **Herança:** a classe Usuario herda de Pessoa e adiciona o atributo `matricula`, que é exclusivo para cada usuário. A classe Funcionario também herda de Pessoa e adiciona o atributo `Cargo`. As duas podem sobrescrever os métodos da classe Pessoa (como fazem com o método `exibirInformacoes()`).
- d. **Polimorfismo:** O polimorfismo é aplicado principalmente no método `exibirInformacoes()`. A classe Pessoa define esse método, mas ele é sobrescrito nas classes Usuario e Funcionario para exibir informações específicas de cada tipo de pessoa. Isso permite que um objeto de tipo Pessoa (que pode ser um Usuario ou um Funcionario) tenha um comportamento diferente dependendo de sua classe real.

### 3. Diagrama UML



#### **4. Dificuldades encontradas**

- Flechas e uso do diagrama