

Sistema de Gerenciamento de Biblioteca (módulo Empréstimo)

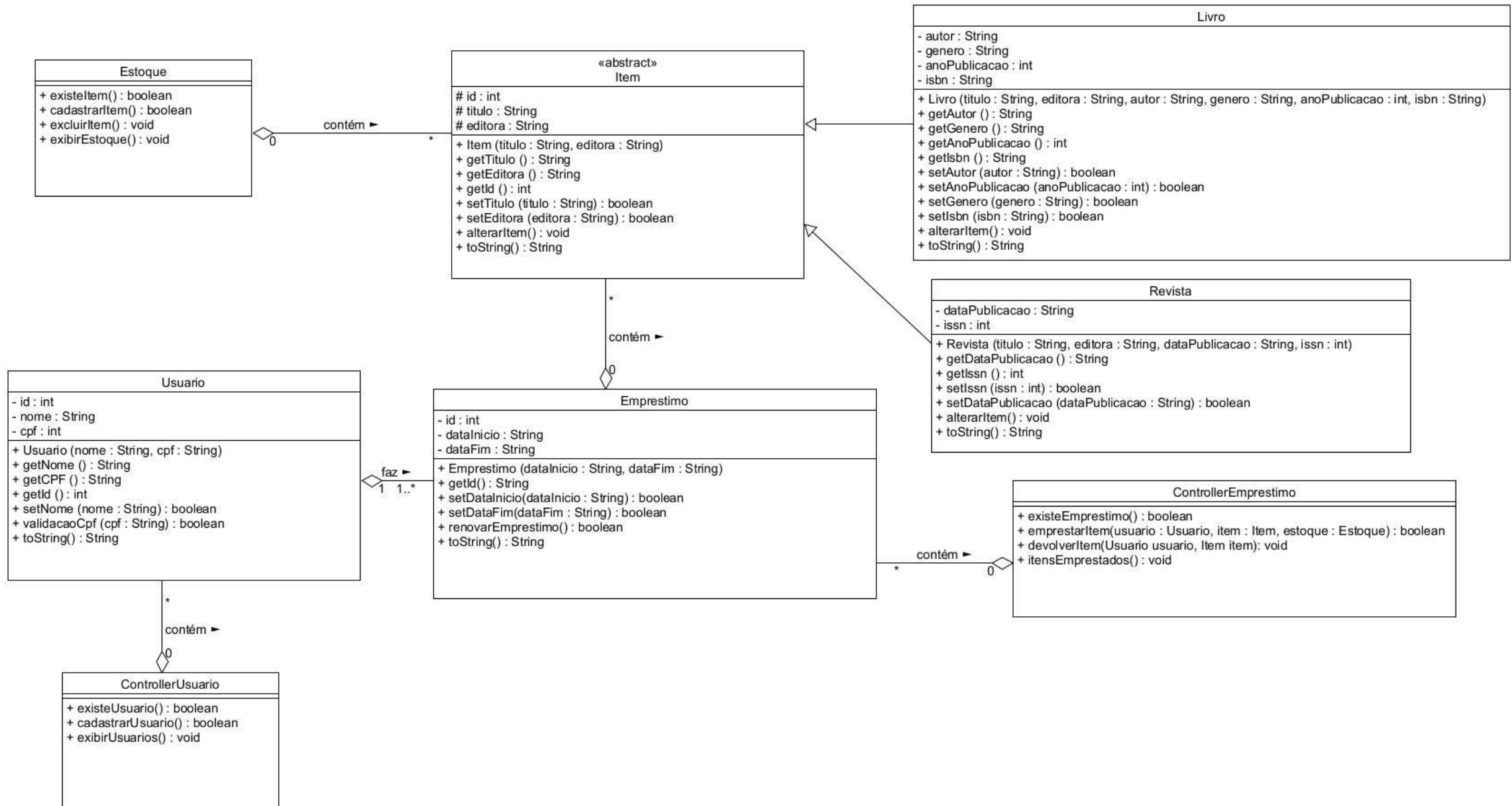
ALUNOS: ANA PAULA E HENRIQUE

Descrição do Projeto

Em um sistema de gerenciamento bibliotecário, a eficiência da rotina de empréstimo de livros desempenha um papel fundamental. Reconhecendo essa importância, nosso projeto concentra-se primordialmente nessa área, visando aprimorar a experiência dos usuários e otimizar os processos internos da biblioteca. Dentre as funcionalidades propostas para esta etapa, destacam-se:

- Empréstimo de itens disponíveis em estoque;
- Devolução de itens emprestados;
- Renovação de empréstimos.

Diagrama de Classes



Encapsulamento

- Para termos um melhor controle de acesso as classes e conseguirmos fazer qualquer manutenção necessária, utilizamos do encapsulamento para proteger os atributos internos da classe, restringindo seu acesso direto e garantindo que eles sejam manipulados apenas por meio de métodos específicos (getters e setters).

```
public abstract class Item {
    private static int lastId = 0; // variável de classe para rastrear o último ID usado
    private int id;
    private String titulo;
    private String editora;

    public Item(String titulo, String editora) {
        this.id = ++lastId; // incrementa automaticamente o ID
        setTitulo(titulo);
        setEditora(editora);
    }

    public int getId() {
        return this.id;
    }

    public String getTitulo() {
        return this.titulo;
    }

    public String getEditora() {
        return this.editora;
    }

    public boolean setTitulo(String titulo) {
        if (titulo != null && !titulo.isEmpty()) {
            this.titulo = titulo;
            return true;
        } else {
            return false;
        }
    }

    public boolean setEditora(String editora) {
        if (editora != null && !editora.isEmpty()) {
            this.editora = editora;
            return true;
        } else {
            return false;
        }
    }
}
```

Abstração

Com a necessidade de criarmos mais de um tipo de item a ser emprestado na biblioteca, criamos uma classe genérica/abstrata com o nome de Item, com os atributos id, nome e editora que nossas classes Livro e Revista podem herdar.

```
public abstract class Item {  
    private static int lastId = 0; // variável de classe para rastrear o último ID usado  
    private int id;  
    private String titulo;  
    private String editora;  
  
    public Item(String titulo, String editora) {  
        this.id = ++lastId; // incrementa automaticamente o ID  
        setTitulo(titulo);  
        setEditora(editora);  
    }  
}
```

Herança e Polimorfismo

- Aqui são as classes herdadas da classe abstrata Item, com os atributos da mesma e seus próprios.

```
class Livro extends Item {
    private String autor;
    private String genero;
    private int anoPublicacao;
    private String isbn; //International Standard Book Number/ Padrão Internacional de Numeração de Livro
    private static final List<String> GENEROS_PERMITIDOS = Arrays.asList("Romance", "Aventura", "Ficcao Cientifica", "Fantasia", "Policial", "Drama");

    public Livro(String titulo, String editora, String autor, String genero, int anoPublicacao, String isbn) {
        super(titulo, editora);
        setAutor(autor);
        setGenero(genero);
        setAnoPublicacao(anoPublicacao);
        setIsbn(isbn);
    }
}
```

```
class Revista extends Item {
    private String dataPublicacao;
    private int issn; //International Standard Serial Number / Número Internacional Normalizado das Publicações em Série

    public Revista(String titulo, String editora, String dataPublicacao, int issn) {
        super(titulo, editora);
        setDataPublicacao(dataPublicacao);
        setIssn(issn);
    }
}
```

Sobrescrita

- Método que possui todos os atributos das classes (toString), originalmente criado na classe genérica Item e sendo sobrescrito nas classes Livro e Revista.

Classe Abstrata Item

```
}  
  
public String toString() {  
    return "ID Item: "+getId()+" | "+  
           "Titulo: "+getTitulo()+" | "+  
           "Editora: "+getEditora();  
}
```

Classe Livro

```
@Override  
public String toString() {  
    return "ID Livro: "+getId()+" | "+  
           "Titulo: "+getTitulo()+" | "+  
           "Editora: "+getEditora()+" | "+  
           "Autor: "+getAutor()+" | "+  
           "Genero: "+getAnoPublicacao()+" | "+  
           "AnoPublicacao: "+getAnoPublicacao()+" | "+  
           "ISBN: "+getIsbn();  
}
```

Classe Revista

```
@Override  
public String toString() {  
    return "ID Revista: "+getId()+" | "+  
           "Titulo: "+getTitulo()+" | "+  
           "Editora: "+getEditora()+" | "+  
           "Data de Publicacao: "+getDataPublicacao()+" | "+  
           "ISSN: "+getIssn();  
}
```

Sobrecarga

- Aqui quisemos dar uma opção a alteração do item no estoque, onde o usuário pode querer alterar o item por completo ou então só uma característica.

Classe Abstrata Item: Método AlterarItem

```
public void alterarItem() {
    Scanner s = new Scanner(System.in);
    System.out.println("Digite a novo titulo do item:");
    setTitulo(s.nextLine());
    System.out.println("Digite a nova editora do item:");
    setEditora(s.nextLine());
    System.out.println("Item editado com sucesso!");
}

public void alterarItem(int opcao) {
    Scanner s = new Scanner(System.in);

    switch(opcao) {
        case 1:
            System.out.println("Digite o novo titulo do item:");
            setTitulo(s.nextLine());
            System.out.println("Titulo alterado com sucesso!");
            break;
        case 2:
            System.out.println("Digite a nova editora do item:");
            setEditora(s.nextLine());
            System.out.println("Editora alterada com sucesso!");
            break;
        default:
            System.out.println("Opcao invalida!");
    }
}
```