

Capacitação Asimov Jr

Davi Ferreira Tase

1. Descrição Geral do Sistema

O Sistema de Gestão Bibliotecária tem como objetivo organizar o acervo de uma biblioteca comunitária, controlando livros e revistas, seus empréstimos e devoluções.

A proposta segue princípios de Programação Orientada a Objetos, utilizando herança, polimorfismo e boas práticas de modelagem.

O sistema permite:

- Registrar itens do acervo (livros e revistas)
- Verificar disponibilidade de cópias
- Realizar empréstimos e devoluções
- Calcular multa por atraso
- Exibir relatórios básicos de estoque

O foco é construir regras de negócio sólidas antes do desenvolvimento de interface.

2. Requisitos Funcionais do Sistema

RF01 – Listar acervo

O sistema deve permitir listar todos os itens cadastrados (livros e revistas), mostrando seus dados completos.

RF02 – Verificar disponibilidade

Ao tentar emprestar um item, o sistema deve verificar se há cópias disponíveis.

RF03 – Empréstimo

O sistema deve permitir emprestar itens, reduzindo a quantidade de cópias.

RF04 – Devolução

A devolução deve aumentar a quantidade de cópias e calcular o valor a ser pago baseado em:

- custo base do item
- multa por dia de atraso

RF05 – Registro de datas

O sistema deve registrar datas de empréstimo e devolução para cálculo de atraso.

RF06 – Suporte a dois tipos de itens

- **Livro** (autor, ISBN)
- **Revista** (edição, mês de publicação)

RF07 – Sistema polimórfico

Ambos devem herdar de uma classe genérica **ItemBiblioteca**.

RF08 – Simulação

No `main.dart`, deve haver:

- uma lista mista com livros e revistas
- empréstimo de 3 livros e 3 revistas
- devolução com cálculo de pagamento

3. Modelagem Orientada a Objetos

Atributos gerais:

- título
- ano de publicação
- cópias disponíveis
- preço base de empréstimo

- multa por dia de atraso
- dataEmpréstimo
- dataDevolução

Métodos:

- emprestar()
- devolver()
- calcularPagamento()
- detalhes() (polimórfico)

3.2 Classe Livro (herda de ItemBiblioteca)

Atributos adicionais:

- autor
- ISBN

Regras financeiras:

- **Preço base:** R\$ 15,00
- **Multa diária:** R\$ 2,50 por dia de atraso

3.3 Classe Revista (herda de ItemBiblioteca)

Atributos adicionais:

- número da edição
- mês de publicação

Regras financeiras:

- **Preço base:** R\$ 5,00
- **Multa diária:** R\$ 1,00 por dia de atraso

4. Comportamento Geral do Sistema

Empréstimo

- verificar disponibilidade (cópias > 0)
- se disponível, reduzir 1 unidade
- registrar a data do empréstimo

Devolução

- repor cópia
- calcular atraso usando:

`diasAtraso = dataDevolucao - dataEmprestimo`

- calcular pagamento:

`valor = precoBase + (diasAtraso * multaDiaria)`

- exibir valor total
- registrar data de devolução

5. Arquivo main.dart

O `main.dart` deve:

✓ Criar lista com livros e revistas

- no mínimo 6 itens (3 livros + 3 revistas)

✓ Listar todos os itens

Usar o método `detalhes()` para exibir informações de todos os objetos.

✓ Realizar empréstimos

Para todos os 6 itens, testar `emprestar()`.

✓ Realizar devoluções

Para todos os 6 itens, testar `devolver()` e exibir:

- nome do item
- preço base
- dias de atraso
- valor final calculado

✓ Exibir saída organizada

Com:

- estoque inicial
- operações de empréstimo
- devoluções com pagamento
- estoque final

```

1  import 'dart:math';
2
3  class ItemBiblioteca {
4    String titulo;
5    int anoPublicacao;
6    int copias;
7    double precoBase;
8    double multaDia;
9    DateTime? dataEmprestimo;
10
11    ItemBiblioteca(
12      this.titulo, this.anoPublicacao, this.copias, this.precoBase, this.multaDia);
13
14    bool emprestar() {
15      if (copias > 0) {
16        copias--;
17        dataEmprestimo = DateTime.now();
18        return true;
19      }
20      return false;
21    }
22
23    double devolver() {
24      if (dataEmprestimo == null) {
25        print("Este item não foi emprestado!");
26        return 0.0;
27      }
28
29      copias++;
30      DateTime dataDevolucao = DateTime.now();
31      int diasAtraso = dataDevolucao.difference(dataEmprestimo!).inDays;
32

```

```

32
33      double pagamento = precoBase + max(0, diasAtraso) * multaDia;
34
35      return pagamento;
36    }
37
38    String detalhes() {
39      return ""
40      Título: $titulo
41      Ano: $anoPublicacao
42      Cópias: $copias
43      Preço Base: R\$ ${precoBase.toStringAsFixed(2)}
44      Multa/dia: R\$ ${multaDia.toStringAsFixed(2)}
45      """;
46    }
47  }
48
49  class Livro extends ItemBiblioteca {
50    String autor;
51    String isbn;
52
53    Livro(String titulo, int anoPublicacao, int copias, this.autor, this.isbn)
54      : super(titulo, anoPublicacao, copias, 15.0, 2.5);
55
56    @override
57    String detalhes() {
58      return ""
59      [LIVRO]
60      ${super.detalhes()}Autor: $autor
61      ISBN: $isbn
62      """;

```

```

63     }
64 }
65
66 class Revista extends ItemBiblioteca {
67     int edicao;
68     String mes;
69
70     Revista(String titulo, int anoPublicacao, int copias, this.edicao, this.mes)
71         : super(titulo, anoPublicacao, copias, 5.0, 1.0);
72
73     @Override
74     String detalhes() {
75         return ""
76 [REVISTA]
77 ${super.detalhes()}Edição: $edicao
78 Mês: $mes
79 """;
80     }
81 }
82
83 // -----
84 // MAIN - SIMULAÇÃO SOLICITADA
85 // -----
86
87 void main() {
88     List<ItemBiblioteca> acervo = [
89         Livro("O Pequeno Príncipe", 1943, 3, "Saint-Exupéry", "978123456789"),
90         Livro("1984", 1949, 2, "George Orwell", "978987654321"),
91         Livro("Dom Casmurro", 1899, 1, "Machado de Assis", "978112233445"),
92         Revista("Superinteressante", 2023, 4, 320, "Outubro"),

```

```
93     Revista("Veja", 2024, 2, 410, "Janeiro"),
94     Revista("National Geographic", 2024, 1, 550, "Dezembro"),
95 ];
96
97 print("=== ESTOQUE INICIAL ===");
98 for (var item in acervo) {
99     print(item.detalhes());
100 }
101
102 print("\n=== EMPRÉSTIMO DE 3 LIVROS E 3 REVISTAS ===");
103 for (int i = 0; i < 6; i++) {
104     bool ok = acervo[i].emprestar();
105     print($"{acervo[i].titulo} -> ${ok ? "Emprestado" : "Indisponível"}");
106 }
107
108 print("\n=== DEVOLUÇÃO E CÁLCULO DE PAGAMENTO ===");
109 for (int i = 0; i < 6; i++) {
110     double valor = acervo[i].devolver();
111     print($"{acervo[i].titulo} -> Pagamento: R\${valor.ToStringAsFixed(2)}");
112 }
113
114 print("\n=== ESTOQUE FINAL ===");
115 for (var item in acervo) {
116     print(item.detalhes());
117 }
118 }
```