



Instituto Federal de Minas Gerais - Campus Ouro Branco

Curso: Bac. Sistemas de Informação

Disciplina: Programação Orientada a Objetos I

Professor: Saulo Henrique Cabral Silva (saulo.cabral@ifmg.edu.br)

Trabalho Prático 3

POO & Processamento de Arquivos

Valor: 15 pontos

Entrega: 03/02/2025

Objetivos: O objetivo deste trabalho é criar um sistema de gestão de biblioteca funcional, interativo e que utilize os conceitos fundamentais de POO, como herança, encapsulamento e polimorfismo. O sistema deve gerenciar usuários, obras literárias e suas respectivas operações, como empréstimos e devoluções, com registro persistente em arquivos, para garantir que o estado do programa seja mantido mesmo após desligar a máquina.

Descrição:

Seu José, recentemente assumiu o cargo de chefe da Biblioteca Municipal de Ouro Branco, e na ocasião enfrenta desafios para gerenciar o crescente número de usuários e obras disponíveis. Ele percebeu que o método manual de controle não é mais eficiente, especialmente com a demanda crescente por empréstimos e devoluções. Para modernizar o sistema, ele solicita sua ajuda, para desenvolver um programa capaz de gerenciar as operações da biblioteca com base nos conceitos de Programação Orientada a Objetos (POO).

Requisitos:

Seu José precisa de um sistema que gerencie as seguintes entidades e funcionalidades:

- **Obras Literárias:** Cada obra terá informações como título, autor e quantidade disponível para empréstimo. O sistema deve atualizar a disponibilidade das obras conforme as operações de empréstimos e devoluções forem realizadas.
- **Usuários:** Existem três tipos de usuários no sistema:
 - Aluno:
 - Nome, e-mail e senha.
 - Matrícula e curso ao qual está vinculado.
 - Limite de até dois empréstimos simultâneos.
 - Professor:
 - Nome, e-mail e senha.
 - Departamento ao qual está vinculado.
 - Limite de até dez empréstimos simultâneos.

- Bibliotecário:
 - Nome, e-mail, senha e telefone.
 - Quantidade total de devoluções realizadas desde sua contratação.
 - Responsável por cadastrar novos usuários e registrar devoluções.
- Regras de Empréstimo:
 - Um empréstimo só pode ser realizado se houver exemplares disponíveis.
 - Usuários com atraso na devolução de qualquer obra ficam bloqueados para novos empréstimos até regularizarem a situação.
- Persistência de Dados:
 - Todos os dados, como usuários, obras e histórico de empréstimos, devem ser registrados em arquivos no formato **.txt**.
 - O sistema deve carregar os dados ao iniciar e salvá-los ao finalizar, garantindo a persistência do estado.

Funcionalidades do Sistema:

1. Cadastro de novos usuários (realizado pelo bibliotecário).
2. Login e Logout dos usuários (apenas um usuário ativo por vez)
3. Gerenciamento de Obras:
 - Consulta as informações de uma obra a partir do seu id, ou nome (qualquer usuário).
 - Atualização da quantidade disponível após empréstimos e devoluções (automático após cada ação).
4. Empréstimos e Devoluções:
 - Registro de empréstimos, respeitando os limites de cada tipo de usuário (disponível para professores e alunos).
 - Registro de devoluções e desbloqueio de usuários com atrasos regularizados (disponível para bibliotecário).
5. Relatórios:
 - Lista de obras emprestadas no momento (disponível para bibliotecário).
 - Usuários com atraso nos empréstimos (disponível para bibliotecário).
6. Menu:
 - Para utilizar qualquer funcionalidade é necessário que o usuário realize um login (informe email e senha cadastrados)
 - A depender do usuário logado, o menu apresentado deverá contemplar funcionalidades competentes ao seu perfil.
 - Garanta que no menu apresentado logo após o login, o mesmo possa realizar seu logout.

Base de dados: Considere o arquivo `acervo.csv` como as obras disponíveis para empréstimo na biblioteca. Atenção à quantidade de cada obra disponível para acesso, é esperado que este número modifique conforme as mesmas sejam emprestadas para os usuários.

O que deve ser entregue:

1. Código fonte do programa em Java (bem *identada* e comentada).
2. Documentação do trabalho. Entre outras coisas, a documentação deve conter:
 - 2.1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
 - 2.2. Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado.
 - 2.3. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
 - 2.4. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso
3. Formato: mandatoriamente em PDF (<http://www.pdf995.com/>).

Obs1: Apesar desse trabalho ser bem simples, a documentação pedida segue o formato da documentação que deverá ser entregue nos próximos trabalhos.

Obs2: Consulte as dicas do Prof. Nívio Ziviani de como deve ser feita uma boa implementação e documentação de um trabalho prático: <https://saulocabral.pagekite.me/roteirotp.pdf>

Como deve ser feita a entrega:

A entrega **DEVE** ser feita através da plataforma moodle na forma de um único arquivo *zipado*, contendo o código, os arquivos, o executável e a documentação.

Comentários Gerais:

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar;
- Clareza, *indentação* e comentários no programa também vão valer pontos;
- O trabalho pode ser feito em trios (**grupos de MÁXIMO 3 alunos**);
- Trabalhos copiados (e **FONTE**) terão nota ZERO;
- Trabalhos entregues em atraso serão aceitos, todavia a nota atribuída ao trabalho será zero
- Evite discussões inúteis com o professor em tentar postergar a data de entrega do referido trabalho.