

Professor: Walter Felipe

Período: 3º

Disciplina: Programação Orientada a Objetos

Enunciado do Projeto: Disciplina de Programação Orientada a Objetos (JAVA)

Objetivo:

Desenvolver um sistema em Java que utilize os conceitos de Programação Orientada a Objetos (POO), integrando coleções, tratamento de exceções, manipulação de arquivos e interfaces gráficas. O projeto deve resolver um problema prático, demonstrando originalidade, complexidade técnica e aplicabilidade no mundo real.

Requisitos Técnicos Obrigatórios

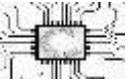
1. Coleções:
 - Uso de estruturas como ArrayList, HashMap, LinkedList, TreeSet, ou similares para armazenar dados.
2. Tratamento de Exceções:
 - Implementação de try-catch, validações personalizadas e mensagens de erro claras.
3. Manipulação de Arquivos:
 - Leitura/escrita de dados em formatos como CSV, JSON, binário ou texto.
4. Interface Gráfica (GUI):
 - Desenvolvimento de interface usando JavaFX ou Swing, com interatividade e usabilidade.
5. Organização do Código:
 - Aplicação de princípios de POO (herança, polimorfismo, encapsulamento, interfaces).

Critérios de Avaliação:

1. Uso Correto da Linguagem: Sintaxe adequada, organização em pacotes, classes bem definidas.
2. Complexidade: Uso de herança, polimorfismo, interfaces e estruturas de dados apropriadas.
3. Inovação/Originalidade: Features únicas (ex.: integração com APIs, algoritmos criativos).
4. Aplicabilidade: Utilidade prática e potencial de impacto real.

Roteiro de Apresentação (Pitch de 7 Minutos):

1. Introdução (1 min)
 - Nome do projeto e problema que resolve.
 - Exemplo: Nosso sistema evita perdas em estoques com alertas automatizados.
2. Demonstração Técnica (3 min)
 - Mostre a GUI em funcionamento.
 - Explique o uso de coleções, exceções e arquivos.
 - Exemplo: Usamos TreeMap para ordenar agendas médicas por data.
3. Diferenciais (1 min)



- Destaque inovações (ex.: integração com QR code, APIs externas).

4. Desafios Técnicos (1 min)

- Exemplo: Implementar serialização de dados exigiu ajustes na classe Transacao.

5. Aplicação no Mundo Real (1 min)

- Exemplo: Este sistema pode ser usado por academias para monitorar equipamentos.

6. Encerramento (30 seg)

- Convide para perguntas e agradeça.

Entrega:

- Código Fonte: Via GitHub e classrrom.
- Documentação: Diagrama de classes (UML) e relatório de 1 página, explicando funcionalidades e tecnologias usadas.
- Data: 04/06/2025

Dica: Testem o projeto em diferentes cenários e validem todas as exceções!

Sugestões de Projetos:

1. Sistema de Gerenciamento de Biblioteca

- Cadastro de livros, empréstimos e usuários.
- Recursos técnicos: HashMap para livros (ISBN como chave), salvamento em CSV.

2. Aplicativo de Gestão de Despesas Pessoais

- Registro de gastos e gráficos de despesas.
- Recursos técnicos: Gráficos com JFreeChart, serialização de dados.

3. Sistema de Reservas de Restaurante

- Reservas de mesas e cardápio digital.
- Recursos técnicos: PriorityQueue para filas, integração com QR code.

4. Controle de Estoque

- Monitoramento de produtos e alertas de baixo estoque.
- Recursos técnicos: Leitor de código de barras via webcam.

5. Agendamento de Consultas Médicas

- Agenda de médicos e lembretes automáticos.
- Recursos técnicos: Integração com API de SMS (Twilio).

(Mais ideias: Jogo de Quiz, Rede Social Simplificada, To-Do List com Sincronização, Aplicativo de Treino Fitness, Aluguel de Veículos.)