

Curso: ADS	Disciplina: FPOO
Professor(a): Daves Martins	
Aluno(a):	Turma:
Data:	Valor Total: 5 Nota Obtida:

1. Monte uma aplicação para o problema abaixo:

Tecnologias obrigatórias: Java Desktop com Swing + banco de dados relacional (MySQL) + conceitos de POO (herança, composição, encapsulamento, pacotes), tratamento de exceções e validações. Uso de Git com commits frequentes e coerentes.

Sistema de Bolsa de Valores (SIBOV)

Contexto

Investidores pessoa física operam **ações** e **FII's** e recebem **dividendos/proventos**. Muitos controlam a carteira em planilhas e perdem o histórico de operações, o saldo disponível e o registro correto de proventos.

O objetivo deste trabalho é construir um **sistema desktop** que permita **comprar/vender ativos, controlar o saldo em caixa, registrar dividendos, e gerar relatórios e histórico** da carteira.

Objetivos do Sistema

- **Cadastro** de ativos (ações e FII's) e do investidor.
- **Lançamento de operações de compra e venda** com data, quantidade, preço e taxas.
- **Controle de saldo** do investidor (aportes/retiradas, débitos e créditos das operações e proventos).
- **Registro de dividendos/proventos** (dividendos/JCP/etc.) com crédito automático em caixa.
- **Relatórios** de posição, preço médio, P/L não realizado, e histórico de proventos.

Papéis (Usuários)

- **Investidor:** realiza operações de compra/venda, lança aportes/retiradas, consulta posição e histórico, registra/visualiza proventos.
- **Administrador:** gerencia cadastros de ativos, corrige lançamentos (via **estorno**), gera relatórios consolidados e configura parâmetros (taxas, por exemplo).

Requisitos Funcionais (mínimos)

1. Cadastro de Ativos

- Campos: **ticker** (ex.: BBDC4, KNRI11), **tipo** (Ação, FII), **segmento/setor** (opcional), observações.
- Herança recomendada: Ativo → Acao, FII.

2. Carteira e Saldo

- **Saldo inicial** cadastrado pelo usuário (com possibilidade de aportes e retiradas).
- Bloquear **compra** quando não houver **saldo** suficiente.
- **Venda** credita automaticamente o saldo.

3. Operações (Ordem/Negócio)

- **Compra e Venda:** data, quantidade, preço unitário, taxa de corretagem (opcional) e observações.
- **Validações:**
 - Não permitir **venda descoberta** (quantidade vendida \leq quantidade em carteira do ativo).
 - Quantidade e preço **positivos**.
- **Histórico** imutável (não apagar; utilizar **estorno** para correções).

4. Preço Médio (PM)

- Atualizar **PM** do ativo por **média ponderada** a cada **compra**.

- Ao **vender**, reduzir apenas a **quantidade** (não altera o PM; calcular **resultado** da venda para relatório de P/L não realizado/realizado conforme desejar).

5. Dividendos/Proventos

- Cadastrar eventos de **proventos** por ativo: tipo (Dividendo/JCP/Outro), **valor por cota/ação**, **data de pagamento**.
- No processamento do evento, **creditar** automaticamente: $\text{valor_por_cota} \times \text{quantidade_detida}$ **na data de pagamento**.
- Registrar no **histórico de proventos** e no **extrato do caixa**.

6. Relatórios e Consultas

- **Posição atual da carteira**: para cada ativo → quantidade, PM, preço total investido, valor de mercado (campo opcional de referência manual), **P/L não realizado** (opcional).
- **Extrato do caixa**: aportes, retiradas, compras, vendas, proventos (com saldo após cada lançamento).
- **Histórico de operações** por período (filtros por ativo e tipo).
- **Proventos recebidos** por período e por ativo, soma total e **DY sobre PM** (yield on cost) **opcional**.

7. Autenticação

- Tela de login com perfis **Investidor** e **Administrador**.

8. Funcionalidade Extra

- Cada dupla deve propor um funcionalidade nova, que envolva cadastro e relatório desse cadastro, não pode haver funcionalidades repetidas.

Procedimentos e Avaliação

1. **Duplas**: registrem a formação no fórum da disciplina **até** a data estipulada em sala.
2. **Proposta de extra**: descrevam a funcionalidade extra **inédita** no fórum até a mesma data. Funcionalidades repetidas entre duplas não serão aceitas.
3. **Apresentação obrigatória**: na data final, cada dupla demonstrará o sistema e responderá a perguntas.

4. **Repositório Git:** o projeto deve estar na **pasta da disciplina no GitLab**. Commits serão avaliados por **quantidade, coerência e distribuição** entre os membros.

Critérios de avaliação POO

- **Funcionalidade e regras de negócio corretas : 1,5**
- **Persistência e integridade de dados 1,0**
- **UI/Usabilidade : 1,0**
- **Arquitetura/OOP/Tratamento de exceções: 1,5**
- **Git/Organização/Apresentação : 0; desconto de pontos**

Critérios de avaliação BDA

- **Script de criação do banco de dados e engenharia reversa – 0,5 ponto**
- **Script com todas as operações realizadas no banco – 0,5 ponto**
- **Script das consultas realizadas – 1,0 ponto**
- **Escreva pelo menos uma das consultas em álgebra relacional – 0,5 ponto**

Observações importantes

- **Trabalhos muito semelhantes** entre si receberão **nota ZERO**.
- Utilize **ao máximo** os conceitos vistos em sala.
- Os **dados devem permanecer salvos** no banco de dados.
- Na apresentação, prepare **casos de teste** que demonstrem: compra sem saldo (deve falhar), venda sem posição (deve falhar), compra que altera o PM corretamente, lançamento de provento e crédito no caixa, estorno de lançamento e sua trilha no extrato.
- Os scripts de criação, operações e consultas ao banco deverão ser postados no ambiente da disciplina de BDA, com a indicação do nome dos membros do grupo.