

RELATÓRIO – Sistema Bancário

Aluna: Maria Eduarda Dornelles Gobbi

Turma: 1F

1. Foi feito um sistema bancário simples com múltiplos clientes que funcione no terminal. O sistema deve permitir cadastrar contas, autenticar por número da conta e senha, e realizar depósito, saque e transferência nas contas, aplicando taxas para cada serviço. É necessário listar saldos e manter a lógica separada em funções, usando listas e matrizes.
2. Fontes buscadas:
 - Documentação oficial do Python (sintaxe de listas, funções, input/print).
 - Anotações de aula e exercícios anteriores sobre os códigos que precisava utilizar.
 - Pesquisei uns códigos no W3CSchools, também tive de revisar no IA porque tinha algumas partes que dava erros que eu não conseguia ajustar sozinha.
3. Lógica principal e estruturas
 - Lista de contas (lista de dicionários): cada conta tem número (único), nome, senha, saldo e status (ativa). Ex.:
`{'numero': 1001, 'nome': 'Juliana', 'senha': '123', 'saldo': 500.00, 'ativa': True}`.
 - Matriz de taxas (lista 2D): para cada serviço há [taxa_fixa, taxa_percentual]. Índice 0=depósito, 1=saque, 2=transferência. A taxa total = fixa + percentual*valor.
 - Matriz de transações (lista 2D): histórico com colunas [timestamp, tipo, origem, destino, valor, taxa, saldo_origem_pós, saldo_destino_pós].
 - Funções: cadastrar_conta, autenticar, ver_saldo, deposito, saque, transferencia, listar_saldos_gerente, mostrar_taxas, menu_conta e menu_principal.
 - Autenticação: login por número da conta + senha.
 - Várias operações: depois do login, o usuário pode executar quantas operações quiser antes de sair.
4. Regras das operações
 - DEPÓSITO: credita-se (valor - taxa). A taxa de depósito é fixa (R\$0,50).
 - SAQUE: debita-se (valor + taxa). A taxa é R\$1,00 + 1% do valor.
 - TRANSFERÊNCIA: debita-se na origem (valor + taxa) e credita-se na conta destino (valor). A taxa é R\$2,00 + 2% do valor.
 - Validações: não aceita valores ≤ 0 , impede transferir para a própria conta e garante saldo suficiente para cobrir valor+taxa.
 - Listagem de saldos: há um modo de gerente (senha “gerente123”) que apenas imprime nome, número e saldo de todas as contas (somente leitura).

5. Problemas encontrados e soluções • Unicidade do número da conta: criado verificador encontrar_conta_por_numero para bloquear cadastros duplicados.
 - Precisão monetária: os saldos e valores são arredondados para 2 casas decimais após cada operação.
 - Entradas inválidas: funções auxiliares
6. • Organização: menus separados para reduzir repetição de código e facilitar a leitura.
 - Matriz exigida no enunciado: além da lista de dicionários, foi utilizada a matriz de taxas e a matriz de transações para cumprir o requisito de “listas e matrizes”.
7. Como executar:
 - Salvar o arquivo sistema_bancario.py, abrir no Visual Studio Code e clicar no terminal para executá-lo.

Considerações finais:

O projeto atende às funcionalidades obrigatórias: cadastro, login com senha, saque, depósito, transferência com taxas, listagem de saldos e simulação de várias operações por sessão.