

Relatório Sistema Bancário Simples - Banco VL

O presente relatório tem como objetivo descrever e analisar o funcionamento de um sistema bancário simples implementado em Python. O programa foi construído a partir de funções e listas, de forma a organizar contas e permitir operações básicas como cadastro, depósito, saque, transferência, consulta de saldo e exibição de todas as contas registradas. Apesar da simplicidade da implementação, o sistema exemplifica bem como estruturas de dados básicas e funções podem ser utilizadas para simular situações reais em um ambiente de programação.

A função `criar_conta` é responsável por registrar um novo cliente no banco. Ela solicita ao usuário o número da conta e o nome do titular, armazenando essas informações em uma lista, junto com o saldo inicial de zero. Essa lista representando a conta é então adicionada à lista principal de contas. Assim, essa função garante que novos clientes possam ser cadastrados de forma organizada e identificável no sistema.

Para realizar depósitos, o sistema utiliza a função `depositar`. Ela recebe o número da conta e o valor a ser depositado, percorre a lista de contas em busca da correspondente e, ao encontrá-la, adiciona o valor informado ao saldo existente. Caso não seja encontrada nenhuma conta com o número fornecido, uma mensagem de erro é exibida ao usuário. Essa função é fundamental para simular a entrada de recursos financeiros em uma conta.

De forma semelhante, a função `sacar` possibilita a retirada de valores de uma conta bancária. Depois de identificar a conta pelo número fornecido, a função verifica se o saldo disponível é suficiente para o saque. Quando a condição é atendida, o valor é subtraído do saldo, refletindo a movimentação financeira. Se não houver fundos suficientes, o sistema informa ao usuário que a operação não pôde ser concluída. Essa função mostra como é possível trabalhar com condições lógicas simples para validar operações financeiras.

A função `transferir` realiza uma operação mais complexa, pois envolve duas contas. Inicialmente, o sistema localiza a conta de origem e verifica se o saldo é suficiente para cobrir o valor desejado. Em seguida, busca a conta de destino. Caso as duas contas existam e haja saldo disponível, o sistema desconta o valor da conta de origem e acrescenta o mesmo valor à conta

de destino. Se alguma das condições falhar, a transferência não é realizada, e o usuário é avisado. Essa função combina a lógica do saque e do depósito, representando de maneira simples como funcionam as transferências entre clientes.

Para consultar informações, o programa conta com a função `consultar_saldo`. Essa função recebe o número da conta como parâmetro, localiza o cliente correspondente e exibe o saldo atual junto ao nome do titular. Caso a conta não seja encontrada, o sistema emite uma mensagem de aviso. Essa operação permite ao usuário acompanhar suas movimentações e é essencial para qualquer sistema financeiro.

Além dessas operações, o programa dispõe da função `mostrar_contas`, que apresenta todas as contas registradas no sistema. Essa função percorre a lista de contas e exibe o número da conta, o nome do titular e o saldo de cada cliente. Dessa forma, o sistema fornece uma visão geral de todos os clientes cadastrados, o que pode ser útil tanto para fins de organização quanto de conferência.

A interação do usuário com o sistema ocorre por meio da função `menu`. Ela organiza todas as opções disponíveis em uma lista de escolhas, permitindo que o usuário decida a operação desejada, como criar contas, realizar depósitos, saques, transferências, consultar saldo ou visualizar todas as contas cadastradas. Essa função é o ponto central da aplicação, pois conecta o usuário às demais funções e garante o fluxo adequado de execução do programa.

O sistema bancário desenvolvido em Python, mesmo sendo simples, cumpre bem o papel de simular as principais operações de um banco. Cada função foi construída para desempenhar uma tarefa específica, e a organização do código em funções separadas facilita tanto a compreensão quanto a manutenção. A utilização de listas como estrutura para armazenar as informações das contas mostra como é possível trabalhar com dados de forma organizada e acessível. O menu, por sua vez, garante que todas essas operações fiquem integradas em uma interface textual direta e intuitiva. Assim, o projeto serve não apenas como exemplo de aplicação prática da linguagem Python, mas também como exercício didático para compreender conceitos fundamentais de programação estruturada.