

# RELATÓRIO - SISTEMA BANCÁRIO: Orientado a Objetos em Python

Nome: Evelyn e Isabella Fracaro

Turma: 1F

Este relatório apresenta a adaptação do projeto do segundo trimestre para o paradigma de Orientação a Objetos (POO). O projeto escolhido foi um Sistema Bancário que permite cadastrar clientes, realizar login e efetuar operações como saque, depósito, consulta de saldo e transferência entre contas.

O objetivo principal foi substituir a estrutura procedural anterior por um modelo baseado em classes, aplicando conceitos de abstração, encapsulamento, modularização e boa organização do código.

O sistema foi dividido em 4 módulos, cada um com responsabilidade própria:

- cliente.py

Contém a classe Cliente, que representa cada usuário do banco.

A classe armazena nome, número da conta, senha e saldo, além de métodos de depósito e saque.

Esse módulo aplica encapsulamento, mantendo os dados do cliente dentro da classe.

- banco.py

Contém a classe Banco, responsável por administrar todos os clientes.

Essa classe centraliza as regras de negócio do sistema, como:

Cadastro de novos clientes

Login

Transferência entre contas

Acesso aos objetos Cliente

Essa divisão ajuda a separar funções de gerenciamento dos dados individuais de cada cliente.

- sistema.py

Contém a classe Sistema Bancário, responsável pela interface textual com o usuário.

Essa classe cuida apenas dos menus e das interações, chamando métodos da classe Banco.

Isso implementa o conceito de abstração e evita misturar interface com lógica de negócios.

- main.py

Arquivo principal que apenas cria uma instância do sistema e inicia o programa.

## Abstração

Cada classe representa uma entidade do mundo real (cliente, banco, sistema), expondo somente o que é necessário para seu uso.

## Encapsulamento

Os atributos dos clientes não são manipulados diretamente fora da classe.

Todas as operações (saque, depósito, transferência) são feitas por métodos adequados.

Responsabilidade Única

Cada classe tem um papel bem definido:

**O sistema permite:**

- Cadastrar clientes
- Fazer login com número da conta e senha
- Exibir saldo
- Realizar saque
- Realizar depósito
- Realizar transferências entre contas
- Encerrar sessão

Todos os dados ficam armazenados em memória utilizando um dicionário dentro da classe Banco. A adaptação do projeto para orientação a objetos tornou o código mais organizado, modular e fácil de manter. Além disso, permitiu aplicar conceitos fundamentais da programação moderna, como encapsulamento e abstração. Foi usado IA para corrigir os erros do código, e também para deixar o código mais organizado e limpo, garantindo uma estrutura mais clara, legível e funcional.