UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

SISTEMA BANCÁRIO DIGITAL

IMPLEMENTAÇÃO BÁSICA DE UM SISTEMA DE GESTÃO FINANCEIRA

PRINCIPAIS FUNCIONALIDADES

- Cadastro e gerenciamento de contas PF/PJ
- Autenticação e transações
- Relatórios e aprovação de contas
- Bloqueio de cartões bancários

UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS

- Cadastro e gerenciamento de contas PF/PJ

```
34  class Banco {
35  private:
36      std::vector<std::shared_ptr<Conta>> contas;
37      std::vector<Transacao> transacoes;
38      std::vector<Cartao> cartoes;
39      std::vector<std::shared_ptr<Cliente>> clientes;
40      std::vector<Saque> saques;
41      std::vector<Deposito> depositos;
42
43  public:
44      int gerenciar_contas();
45      void validar_transacao(Transacao &transacao);
46      int validar_transacoes();
47      int criar_cartao();
48      int bloquear_cartao();
49      int gerar_relatorio();
50      int realizar_transacao();//conta origem = 1
51      int realizar_saque();
52      int realizar_deposito();
53      int posicao_id(int id);//retorna a posição do ID no vetor contas
54      bool verifica_id(int id);
55      bool autenticar_conta(int id_conta);
56      bool verificar_cartao(int id_conta, const std::string& numero_cartao);
57      void gerar_extrato(int id_conta);
58      // Getters para acesso controlado aos dados internos
59      std::vector<std::shared_ptr<Conta>>& getContas();
60      std::vector<Transacao>& getTransacoes();
61  };
```

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

POO APLICADO

- Encapsulamento, Herança, Polimorfismo
- Classes: Conta, ContaPf, ContaPj, Cliente, Banco, Gerente
- Métodos polimórficos e modularização clara

UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS

- Encapsulamento em *class* Cliente

```cpp
class Cliente {
private:
    std::string _nome;
    std::string _cpf_cnpj;
    std::string _rg;
    std::string _senha;
    std::string _endereco;
    std::string _email;
    std::string _telefone;

public:
    Cliente(std::string nome, std::string cpf_cnpj, std::string rg, std::string senha,
            std::string endereco, std::string email, std::string telefone);

    bool autenticar_usuario(const std::string& senha, const std::string& rg) const;
    std::string get_dados(int dado) const;
    void atualizar_metodos_contato(const std::string& endereco, const std::string& email, const std::string& telefone);

    // Getters
    std::string get_nome() const;
    std::string get_cpf_cnpj() const;
    std::string get_rg() const;
    std::string get_senha() const;
    std::string get_endereco() const;
    std::string get_email() const;
    std::string get_telefone() const;
};
```

- *class* Conta como Superclasse

```
public:
    virtual ~Conta() = default;
    virtual void bloquear() = 0;
    virtual void ativar() = 0;
    virtual std::string getNomeTitular() const = 0;
    bool depositar(double valor);
    bool sacar(double valor);
    void aprovar();
    void definirLimite(double novoLimite);
    int getId() const;
    double getSaldo() const;
    double getLimite() const;
    bool isAtivo() const;
    bool isAprovada() const;
    std::shared_ptr<Cliente> getTitular() const;
    void set_num_cartao(std::string num);
};
```

- Herança PF

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

```
 9  class ContaPf : public Conta {
10  private:
11      const float _taxaSelic = 0.0125;//ao mês
12      bool validarCpf();
13
14  public:
15      ContaPf(std::shared_ptr<Cliente> titular, int id, const std::string& senha, double saldoInicial = 0.0);
16
17      void bloquear() override;
18      void ativar() override;
19      std::string getNomeTitular() const override;
20      float calcualarTesouro(int tempo, float investimento);
21      std::string getCpf() const;
22  };
```

- Herança PJ

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

```
11  class ContaPj : public Conta {
12  private:
13      bool validarCnpj();
14      Calendario _dataCriacao;
15
16
17  public:
18      ContaPj(std::shared_ptr<Cliente> titular, int id, const std::string& senha, double saldoInicial);
19
20      void bloquear() override;
21      void ativar() override;
22      std::string getNomeTitular() const override;
23
24      std::string getRazaoSocial() const;
25      std::string getCnpj() const;
26  };
```

- Polimorfismo em *class* Conta

```
21  public:
22      virtual ~Conta() = default;
23
24      // Métodos virtuais puros
25      virtual void bloquear() = 0;
26      virtual void ativar() = 0;
27      virtual std::string getNomeTitular() const = 0;
28
29      // Métodos de operação
30      bool depositar(double valor);
31      bool sacar(double valor);
32      void aprovar();
33      void definirLimite(double novoLimite);
34
35      int getId() const;
36      double getSaldo() const;
37      double getLimite() const;
38      bool isAtivo() const;
39      bool isAprovada() const;
40      std::shared_ptr<Cliente> getTitular() const;
41      void set_num_cartao(std::string num);
42  };
```

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

- *Override* em PF

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

```
14  public:
15      ContaPf(std::shared_ptr<Cliente> titular, int id, const std::string& senha, double saldoInicial = 0.0);
16
17      void bloquear() override;
18      void ativar() override;
19      std::string getNomeTitular() const override;
20      float calcualarTesouro(int tempo, float investimento);
21      std::string getCpf() const;
22  };
```

- *Override* em PJ

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

```
17  public:
18      ContaPj(std::shared_ptr<Cliente> titular, int id, const std::string& senha, double saldoInicial);
19
20      void bloquear() override;
21      void ativar() override;
22      std::string getNomeTitular() const override;
23
24      std::string getRazaoSocial() const;
25      std::string getCnpj() const;
26  };
```

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

TRATAMENTO DE EXCEÇÕES

- EntradaInvalidaException
- ContaNaoEncontradaException
- SaldoInsuficienteException
- CartaoNaoEncontradoException

- Quatro ocorrências principais

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

```cpp
class ContaNaoEncontradaException : public std::runtime_error {
public:
    ContaNaoEncontradaException(int id) : std::runtime_error("Conta com ID " + std::to_string(id) + " nao encontrada.") {}
};

class CartaoNaoEncontradoException : public std::runtime_error {
public:
    CartaoNaoEncontradoException(const std::string& numero) : std::runtime_error("Cartao com numero " + numero + " nao encontrado.") {}
};

class EntradaInvalidaException : public std::runtime_error {
public:
    EntradaInvalidaException(const std::string& msg = "Entrada invalida.")
        : std::runtime_error(msg) {}
};

class SaldoInsuficienteException : public std::runtime_error {
public:
    SaldoInsuficienteException() : std::runtime_error("Saldo insuficiente para a operaçao.") {}
};
```

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

TESTES UNITÁRIOS

- Framework: doctest (~50% gcovr)
- Testes de cadastro, autenticação, transações, exceções.
- Cobertura estimada > 23 de 27 testes.

- Testes de cadastro, autenticação, transações

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

```
10  TEST_CASE("Validar transações com valor limite") {
11      Banco banco;
12      Transacao t1;
13      t1.conta_origem = 1;
14      t1.conta_destino = 2;
15      t1.valor = 3000;
16      t1.data = "01/01/2024";
17      t1.aprovada = false;
18
19      Transacao t2;
20      t2.conta_origem = 1;
21      t2.conta_destino = 2;
22      t2.valor = 7000;
23      t2.data = "01/01/2024";
24      t2.aprovada = false;
25
26      banco.validar_transacao(t1);
27      banco.validar_transacao(t2);
28
29      CHECK(t1.aprovada == true);
30      CHECK(t2.aprovada == false);
31  }
```

- Teste Tesouro

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

```cpp
TEST_CASE("Cálculo do Tesouro Direto com taxa Selic") {
    auto cliente = std::make_shared<Cliente>("Pedro", "88899900011", "PE123456", "senha", "Rua E", "[pedro@mail.com](mailto:pedro@mail.com)", "31955555555");
    ContaPf conta_pf(cliente, 1, "senha", 1000.0);

    float investimento = 1000.0f;
    int tempo = 6;
    float taxaMensal = 0.0125f;

    float esperado = investimento * std::pow(1 + taxaMensal, tempo);
    float montante = conta_pf.calcularTesouro(tempo, investimento);

CHECK(montante == doctest::Approx(esperado).epsilon(0.001)); // tolerância de ~0.1%
}
```

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

RELATÓRIOS E TEMPLATES

- Contas ativas/aprovadas
- Saldos e limites médios
- Uso de templates para exibição formatada

- Template principal

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

```cpp
template <typename T>
void exibirTotal(const std::string& rotulo, T valor) {
    std::cout << rotulo << ": " << valor << std::endl;
}

template <typename T>
void exibirPercentual(const std::string& rotulo, T parte, T total) {
    double percentual = 0.0;
    if (total != 0) {
        percentual = (static_cast<double>(parte) / total) * 100.0;
    }
    std::cout << rotulo << ": " << parte << " (" << std::fixed << std::setprecision(2) << percentual << "%)" << std::endl;
}

template <typename T1, typename T2>
void exibirMedia(const std::string& rotulo, T1 soma, T2 quantidade) {
    double media = 0.0;
    if (quantidade != 0) {
        media = static_cast<double>(soma) / quantidade;
    }
    std::cout << rotulo << ": R$ " << std::fixed << std::setprecision(2) << media << std::endl;
}
```

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

TESTE PRINCIPAL COM EXTRATO

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

OBRIGADO!

INTEGRANTES:

- Lucas Miller dos Santos Sousa
- Alan Pessoa Silva
- João Augusto Rosa Cunha
- Natan Inoue dos Reis