



Vitor Manoel Vidal Braz

TRABALHO POO PRIMEIRA PARTE –

Trabalho em Aula: Sistema Bancário

Governador Valadares

2025

Documento Oficial da Aplicação Java – Sistema Bancário UniBankOne

1. Objetivo da Aplicação

O objetivo desta aplicação é desenvolver um sistema bancário completo em linguagem Java, utilizando os fundamentos da Programação Orientada a Objetos (POO). O sistema UniBankOne simula funcionalidades reais de um banco, como abertura de contas, movimentações financeiras e controle administrativo. Também contempla regras específicas como controle de menores de idade e autorização por responsáveis, seguindo práticas profissionais de desenvolvimento e organização de código.

2. Estrutura da Aplicação

Classes principais:

- **SistemaBancario.java** – Classe principal com o método `main` e controle de toda a aplicação.
- **Pessoa.java** – Classe base com atributos comuns como nome, CPF, endereço e data de nascimento.
- **Cliente.java** – Herda de Pessoa, com número de cliente e lista de contas bancárias.
- **Funcionario.java** – Herda de Pessoa, com atributos como cargo e senha para login administrativo.
- **Conta.java** – Classe abstrata com estrutura genérica de uma conta bancária.
- **ContaCorrente.java** – Conta com cheque especial.
- **ContaPoupanca.java** – Conta com taxa de juros.
- **Banco.java** – Gerencia os clientes e as contas do sistema.

Todas as classes utilizam encapsulamento, métodos `get/set`, validações e modularização.

3. Requisitos Atendidos

Requisito	Descrição	Implementação
Cadastro de Cliente	Solicita nome, CPF, endereço, data de nascimento e número do cliente.	Com validações de CPF único, nome e idade.
Abertura de Conta Corrente	Inclui limite de cheque especial. Verifica CPF e autorização se menor.	Validação implementada, com termo de autorização.
Abertura de Conta Poupança	Solicita taxa de juros e CPF. Verifica se o cliente pode abrir conta.	Autorização e validação incluídas.
Operações Bancárias	Depósito, Saque e Transferência entre contas.	Com verificação de saldo e contas válidas.
Proteção dos Atributos	Uso de modificadores <code>private</code> e métodos acessores.	Implementado em todas as classes.

Requisito	Descrição	Implementação
Controle de Menores	Menores só podem abrir conta com termo de autorização assinado.	Implementação completa no cadastro de conta.
Máscara de CPF	Formatação automática do CPF digitado.	Aplicado ao exibir e validar dados.
BackOffice (Área ADM)	Login com usuário e senha. Estatísticas, alterações, exclusões.	Com menu exclusivo, protegido e funcional.
Encerramento e Estilo Visual	Mensagens de entrada, crédito final e identidade visual.	Com base na VM ENGINE DEVELOPMENT v1.5.

4. Visão Geral das Classes

Pessoa.java

- Atributos: nome, cpf, endereco, dataNascimento
- Métodos: Getters e Setters

Cliente.java

- Herda de Pessoa
- Atributos: numeroCliente, List
- Métodos: abrirConta(), listarContas(), getIdade()

Funcionario.java

- Herda de Pessoa
- Atributos: cargo, senha
- Métodos: autenticar(String senha)

Conta.java (abstrata)

- Atributos: numeroConta, saldo, titular
- Métodos: sacar(), depositar(), transferir()

ContaCorrente.java

- Herda Conta
- Atributos: limiteChequeEspecial
- Métodos: sacar() com limite

ContaPoupanca.java

- Herda Conta
- Atributos: taxaJuros
- Métodos: aplicarJuros()

Banco.java

- Atributos: nome, listaClientes, listaContas

- Métodos: cadastrarCliente(), buscarContaPorNumero(), buscarClientePorCpf()
-

5. Funcionalidades Detalhadas do Sistema

Menu Principal

1. Cadastrar Cliente
2. Abrir Conta Corrente
3. Abrir Conta Poupança
4. Listar Contas do Cliente
5. Depositar
6. Sacar
7. Transferir
8. BackOffice UniBankOne (ADM)
9. Sobre o Sistema
10. Sair

Área Administrativa (BackOffice)

O acesso ao BackOffice é protegido por autenticação. Para acessar:

- **Usuário:** Admin
- **Senha:** 1234

Após login, são oferecidas as opções:

- Lista de clientes e contas
- Estatísticas (clientes, contas, menores)
- Buscar cliente por CPF
- Listar contas abertas
- Total de contas por tipo
- Listar responsáveis por menores
- Contas com saldo zerado
- Alterar dados do cliente
- Excluir conta bancária

Validações e Regras

- CPF com 11 dígitos, sem duplicação
 - Nome sem números
 - Termo de responsabilidade para menores
 - Autorização válida (nome, CPF e data)
 - Uma conta por número único
-

6. Máscara do CPF

O sistema exige que o CPF seja digitado com 11 dígitos e o formata para 000.000.000-00 para visualização padronizada.

7. Controle de Menores de Idade

Se o cliente for menor de 18 anos:

- É necessário preencher um termo de autorização com:
 - Nome e CPF do responsável legal
 - Data da autorização
 - A autorização é salva uma única vez e é vinculada ao menor.
 - Após o aceite, o menor pode abrir contas livremente.
-

8. Créditos e Informações Técnicas

Sistema desenvolvido por **Vitor Manoel Vidal Braz**

- Curso: Sistemas de Informação
- Período: 3º período
- Universidade: UNIVALE – Universidade Vale do Rio Doce
- Orientador: Prof. Vitor Silva Ribeiro
- Linguagem: Java (console)
- Estilo visual: **VM ENGINE DEVELOPMENT v1.5**

Repositório GitHub:

<https://github.com/vitormanoelvb/sistema-bancario-unibankone>

9. Conclusão

O UniBankOne apresenta uma estrutura sólida e funcional que simula com fidelidade a rotina de um sistema bancário. O projeto aplica os conceitos essenciais de POO e entrega uma experiência rica, mesmo em ambiente de terminal.

Com todas as validações implementadas, proteção de dados, controle de menores e painel administrativo completo, o sistema atende 100% das exigências acadêmicas do trabalho.

Sugestão futura: versão com interface gráfica (JavaFX), Implementação de Banco de Dados ou versão web com Spring Boot.