

ENGENHARIA DE SOFTWARE

JOÃO GABRIEL SOUZA SARAIVA

RELATÓRIO DE AULA PRÁTICA:
Linguagem Orientada a Objetos

Itaboraí
2025

JOÃO GABRIEL SOUZA SARAIVA

RELATÓRIO DE AULA PRÁTICA:
Linguagem Orientada a Objetos - Portfólio

Trabalho de Portfólio apresentado como requisito parcial para a obtenção de média bimestral na
disciplina de Linguagem Orientada a Objetos

Orientador: Renan Cleverson Laureano Flor da Rosa

Itaborai
2025

SUMÁRIO

1 INTRODUÇÃO	1
2 DESENVOLVIMENTO	1
3 RESULTADO	1
4 CONCLUSÃO	1
REFERÊNCIAS	1

1 INTRODUÇÃO

Desafio proposto:

Utilizando os principais conceitos do paradigma de Orientação a Objetos, crie uma pequena aplicação de gerenciamento bancário que possibilite ao usuário informar seu nome, sobrenome e CPF. Além disso, a aplicação deverá possibilitar ao usuário consultar saldo, realizar depósitos e saques. Esses procedimentos devem se repetir até que o usuário escolha encerrar o uso da aplicação.

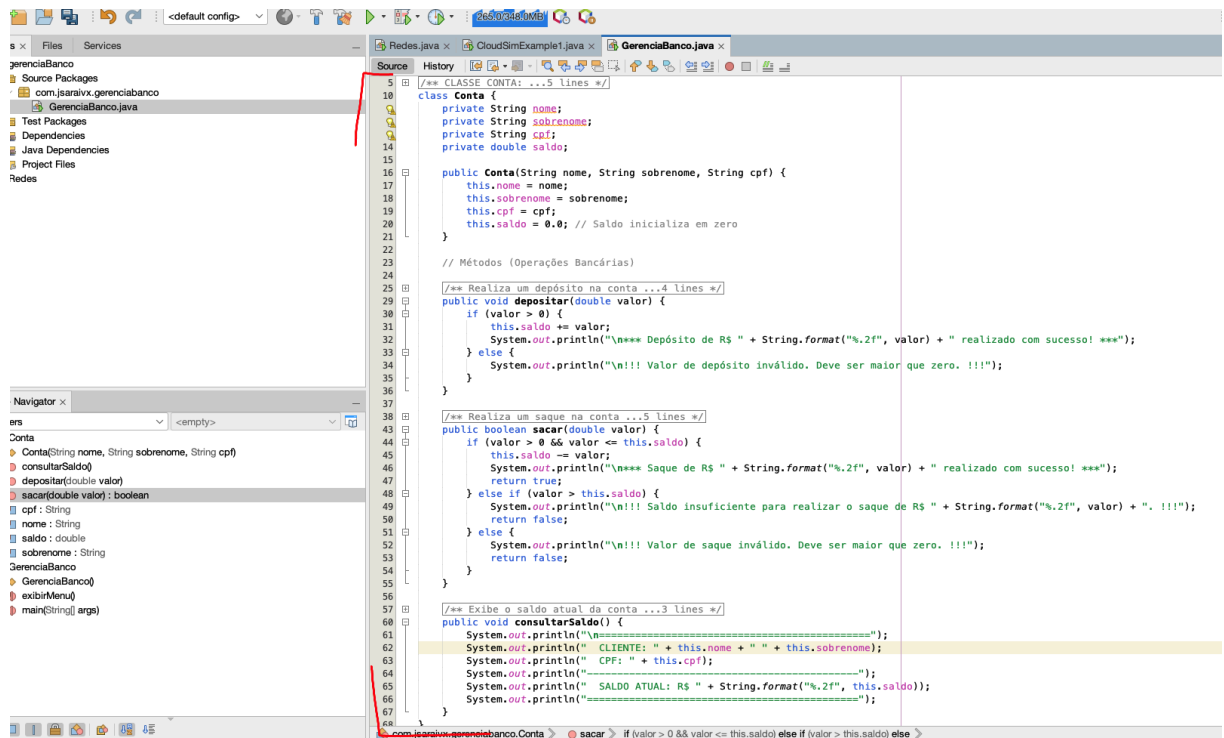
Usaremos o Netbeans, e o paradigma de Orientação de Objetos em Java para essa resolução.

Principais vantagens do uso de OO:

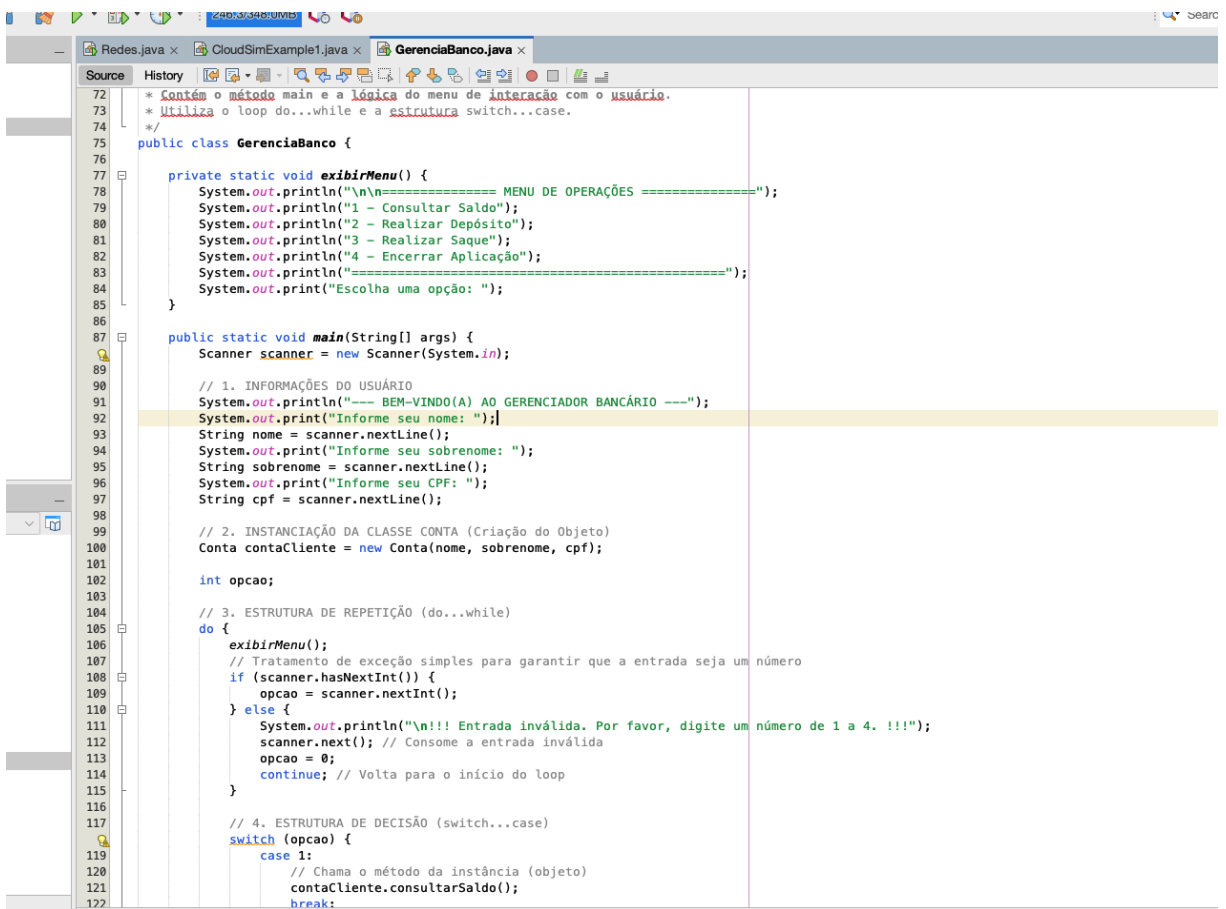
As vantagens da orientação a objetos (POO) incluem a reutilização de código (via herança), modularidade e manutenibilidade (através da organização em classes), escalabilidade (ao permitir a expansão de funcionalidades sem reescrever todo o código) e melhor organização e clareza (por modelar o sistema de forma mais próxima ao mundo real).

2 DESENVOLVIMENTO

Começaremos criando a classe Conta, que receberá os métodos depositar(), e sacar(), além de consultarSaldo().



Começaremos criando a classe GerenciaBanco, que contém o método Main() com a lógica do menu de interação com o usuário.



```

72  * Contém o método main e a lógica do menu de interação com o usuário.
73  * Utiliza o loop do...while e a estrutura switch...case.
74  */
75  public class GerenciaBanco {
76
77      private static void exibirMenu() {
78          System.out.println("\n\n===== MENU DE OPERAÇÕES =====");
79          System.out.println("1 - Consultar Saldo");
80          System.out.println("2 - Realizar Depósito");
81          System.out.println("3 - Realizar Saque");
82          System.out.println("4 - Encerrar Aplicação");
83          System.out.println("=====");
84          System.out.print("Escolha uma opção: ");
85      }
86
87      public static void main(String[] args) {
88          Scanner scanner = new Scanner(System.in);
89
90          // 1. INFORMAÇÕES DO USUÁRIO
91          System.out.println("--- BEM-VINDO(A) AO GERENCIADOR BANCÁRIO ---");
92          System.out.print("Informe seu nome: ");
93          String nome = scanner.nextLine();
94          System.out.print("Informe seu sobrenome: ");
95          String sobrenome = scanner.nextLine();
96          System.out.print("Informe seu CPF: ");
97          String cpf = scanner.nextLine();
98
99          // 2. INSTANCIÇÃO DA CLASSE CONTA (Criação do Objeto)
100         Conta contaCliente = new Conta(nome, sobrenome, cpf);
101
102         int opcao;
103
104         // 3. ESTRUTURA DE REPETIÇÃO (do...while)
105         do {
106             exibirMenu();
107             // Tratamento de exceção simples para garantir que a entrada seja um número
108             if (scanner.hasNextInt()) {
109                 opcao = scanner.nextInt();
110             } else {
111                 System.out.println("\n!!! Entrada inválida. Por favor, digite um número de 1 a 4. !!!");
112                 scanner.next(); // Consome a entrada inválida
113                 opcao = 0;
114                 continue; // Volta para o início do loop
115             }
116
117             // 4. ESTRUTURA DE DECISÃO (switch...case)
118             switch (opcao) {
119                 case 1:
120                     // Chama o método da instância (objeto)
121                     contaCliente.consultarSaldo();
122                     break;

```

Lógica de repetição de do/while e switch/case.

```

101 int opcao;
102
103 // 3. ESTRUTURA DE REPETIÇÃO (do...while)
104 do {
105     exibirMenu();
106     // Tratamento de exceção simples para garantir que a entrada seja um número
107     if (scanner.hasNextInt()) {
108         opcao = scanner.nextInt();
109     } else {
110         System.out.println("\n!!! Entrada inválida. Por favor, digite um número de 1 a 4. !!!");
111         scanner.next(); // Consome a entrada inválida
112         opcao = 0;
113         continue; // Volta para o início do loop
114     }
115
116 // 4. ESTRUTURA DE DECISÃO (switch...case)
117 switch (opcao) {
118     case 1:
119         // Chama o método da instância (objeto)
120         contaCliente.consultarSaldo();
121         break;
122     case 2:
123         System.out.print("\nInforme o valor para depósito: R$ ");
124         if (scanner.hasNextDouble()) {
125             double valorDeposito = scanner.nextDouble();
126             contaCliente.depositar(valorDeposito);
127         } else {
128             System.out.println("\n!!! Valor inválido. Digite um número decimal ou inteiro. !!!");
129             scanner.next();
130         }
131         break;
132     case 3:
133         System.out.print("\nInforme o valor para saque: R$ ");
134         if (scanner.hasNextDouble()) {
135             double valorSaque = scanner.nextDouble();
136             contaCliente.sacar(valorSaque);
137         } else {
138             System.out.println("\n!!! Valor inválido. Digite um número decimal ou inteiro. !!!");
139             scanner.next();
140         }
141         break;
142     case 4:
143         System.out.println("\n*** Obrigado por utilizar o Gerenciador Bancário. Aplicação encerrada. ***");
144         break;
145     default:
146         System.out.println("\n!!! Opção inválida. Tente novamente. !!!");
147         break;
148 }
149 } while (opcao != 4); // Condição de saída
150
151 com.jsaraivx.gerenciaBanco.GerenciaBanco > main >

```

3 RESULTADO

Então, depois desses executados de implementação, podemos finalmente dar "Run" em nosso projeto, para testarmos, e então verificar o funcionamento da resolução da atividade proposta.

Tenha acesso ao código desenvolvido no meu repositório:

<https://github.com/jsaraivx/software-engineering-bachelors/blob/main/period-4/oo-lan guage/gerenciaBanco.java>

Ao rodarmos o programa, nós informaremos nosso nome, sobrenome e então nosso CPF(usei um fictício para o exemplo abaixo).

```

--- compiler:3.13.0:compile (default-compile) @ gerenciaBanco ---
Nothing to compile - all classes are up to date.

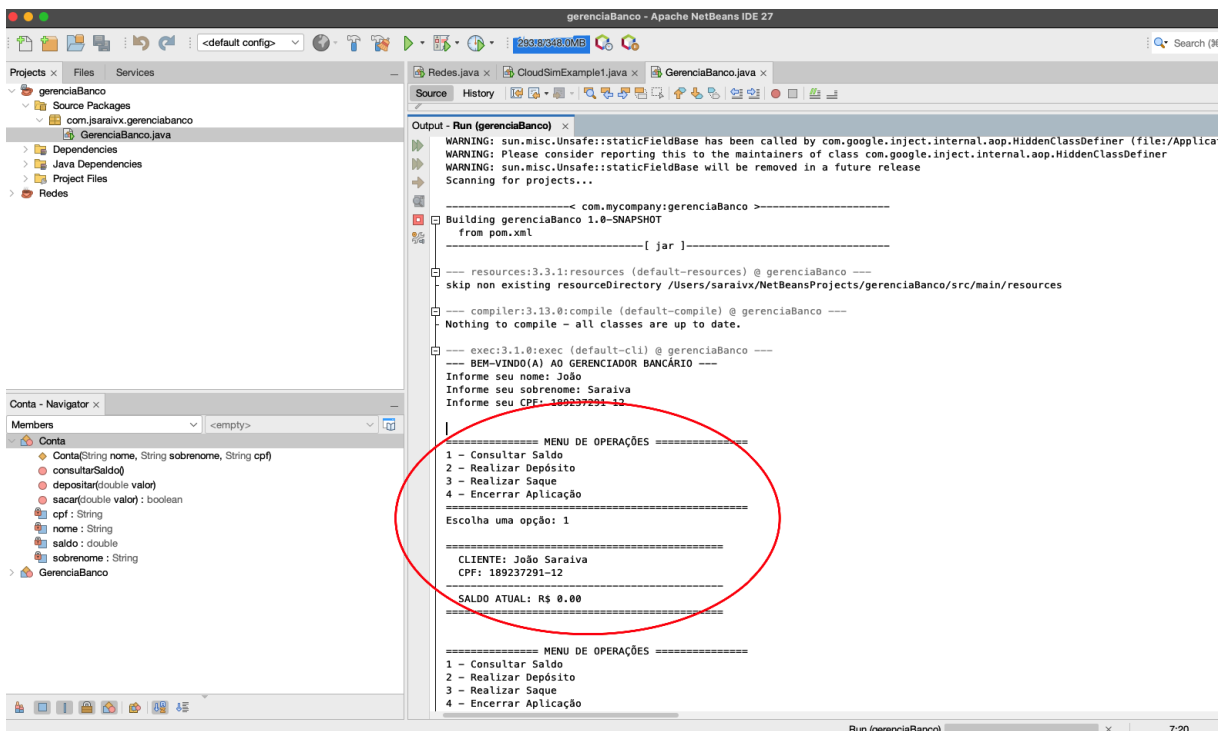
--- exec:3.1.0:exec (default-cli) @ gerenciaBanco ---
--- BEM-VINDO(A) AO GERENCIADOR BANCÁRIO ---
Informe seu nome: João
Informe seu sobrenome: Saraiva
Informe seu CPF: 189237291-12

===== MENU DE OPERAÇÕES =====

```

Então, após nos cadastrarmos, será exibido um menu de interação, onde temos acesso às principais operações disponíveis.

Abaixo selecionamos a operação de consulta de saldo, o que nos retornará 0, pois a conta acabou de ser criada.



Agora faremos um Depósito de teste, com um valor de aproximadamente 120 mil reais, e logo abaixo faremos uma nova consulta, que nos retornará o valor depositado anteriormente.

```

Output - Run (GerenciaBanco) x
=====
CLIENTE: João Saraiva
CPF: 189237291-12
=====
SALDO ATUAL: R$ 0,00
=====

===== MENU DE OPERAÇÕES =====
1 - Consultar Saldo
2 - Realizar Depósito
3 - Realizar Saque
4 - Encerrar Aplicação
=====
Escolha uma opção: 2

Informe o valor para depósito: R$ 122989

*** Depósito de R$ 122989,00 realizado com sucesso! ***

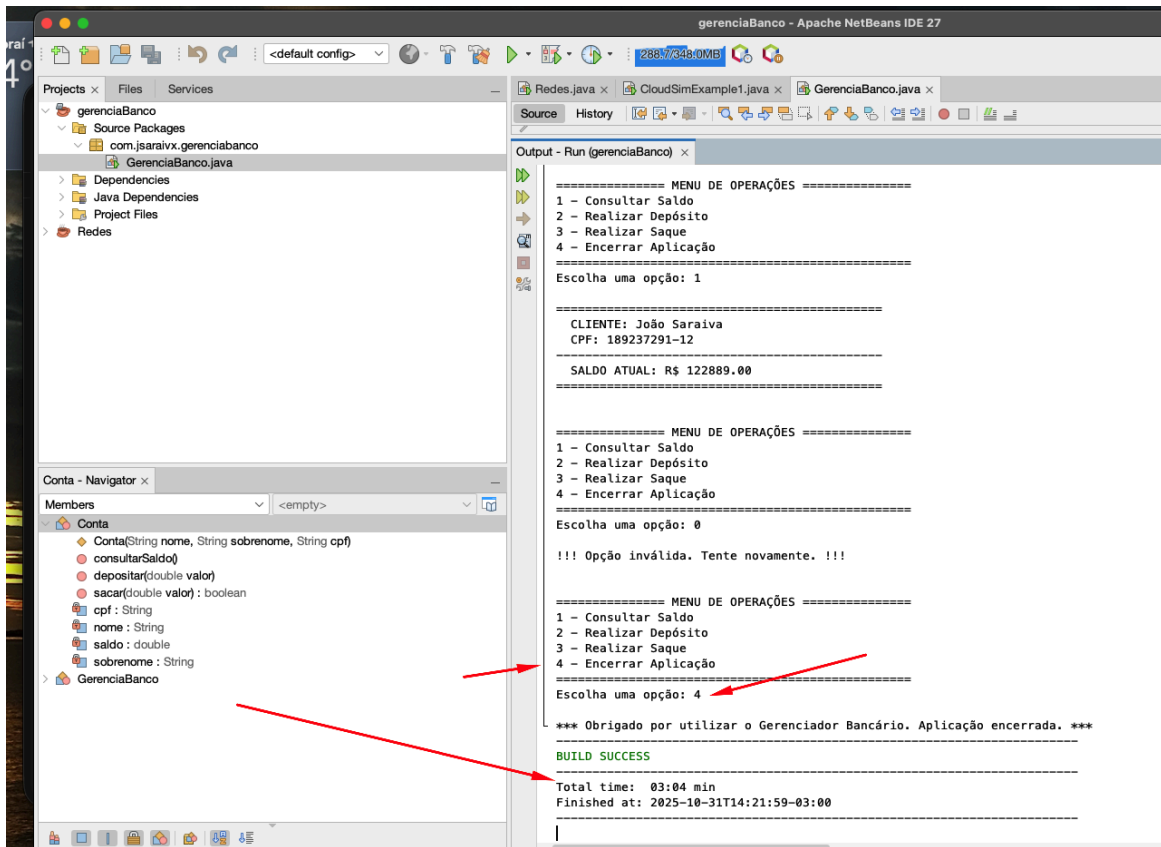
===== MENU DE OPERAÇÕES =====
1 - Consultar Saldo
2 - Realizar Depósito
3 - Realizar Saque
4 - Encerrar Aplicação
=====
Escolha uma opção: 1

=====
CLIENTE: João Saraiva
CPF: 189237291-12
=====
SALDO ATUAL: R$ 122989,00
=====

===== MENU DE OPERAÇÕES =====
1 - Consultar Saldo
2 - Realizar Depósito
3 - Realizar Saque
4 - Encerrar Aplicação
=====

```

Após esse teste, então para finalizarmos o programa, nós usamos a opção 4 para encerrar a execução.



4 CONCLUSÃO

Este trabalho teve como principal objetivo praticar o desenvolvimento de uma linguagem de programação orientada a objetos, nos trazendo a oportunidade de simular de forma simplificada o funcionamento de um banco(instituição financeira), aproveitando as vantagens da Orientação a Objetos e suas vantagens.

REFERÊNCIAS

MAINGUENEAU, Dominique. **Elementos de linguística para o texto literário**. São Paulo: Martins Fontes, 1996.

RAMPAZZO, Lino. **Metodologia científica**: para alunos dos cursos de graduação e pós-graduação. São Paulo: Stiliano, 1998.

REIS, José Luís. **O marketing personalizado e as tecnologias de Informação**. Lisboa: Centro Atlântico, 2000.

UNIVERSIDADE FEDERAL DO PARANÁ. Biblioteca Central. **Normas para apresentação de trabalhos**. 2. ed. Curitiba: UFPR, 1992. v. 2.