

ANHANGUERA EDUCACIONAL
SCRN (ASA NORTE) 708/709 - BLOCO B, S/N, ASA NORTE - BRASÍLIA – DF

SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

AUTOR(ES):
LUCAS LOPES RIBEIRO

LINGUAGEM ORIENTADA A OBJETOS
SISTEMA DE GERENCIAMENTO BANCÁRIO EM JAVA

BRASILIA
2024

Lucas Lopes Ribeiro
LUCAS LOPES RIBEIRO

RA: 3669974002

LINGUAGEM ORIENTADA A OBJETOS

Elaboração de sistema de gerenciamento bancário utilizando o software Netbeans.

Portifólio de Linguagem orientada a objetos apresentado ao(s)
tutor(es): André Martins e Vinicius Mendes Gomes da Silva, do
curso de Tecnologia em Análise e Desenvolvimento de Sistemas
da instituição Anhanguera Educacional.

BRASILIA

2024

SUMARIO

1 INTRODUÇÃO.....	4
2 DESENVOLVIMENTO.....	5
3 CONCLUSÃO.....	10

INTRODUÇÃO

Neste trabalho, é apresentado o desenvolvimento de um sistema de gerenciamento bancário simples, implementado na linguagem Java. O sistema foi projetado para permitir aos usuários realizar operações bancárias básicas, como consultar saldo, efetuar depósitos e saques, através de uma interface de usuário fácil de usar.

O projeto foi desenvolvido seguindo uma abordagem de programação orientada a objetos. Isso permitiu a criação de um sistema com componentes que podem ser facilmente modificados ou estendido conforme necessário.

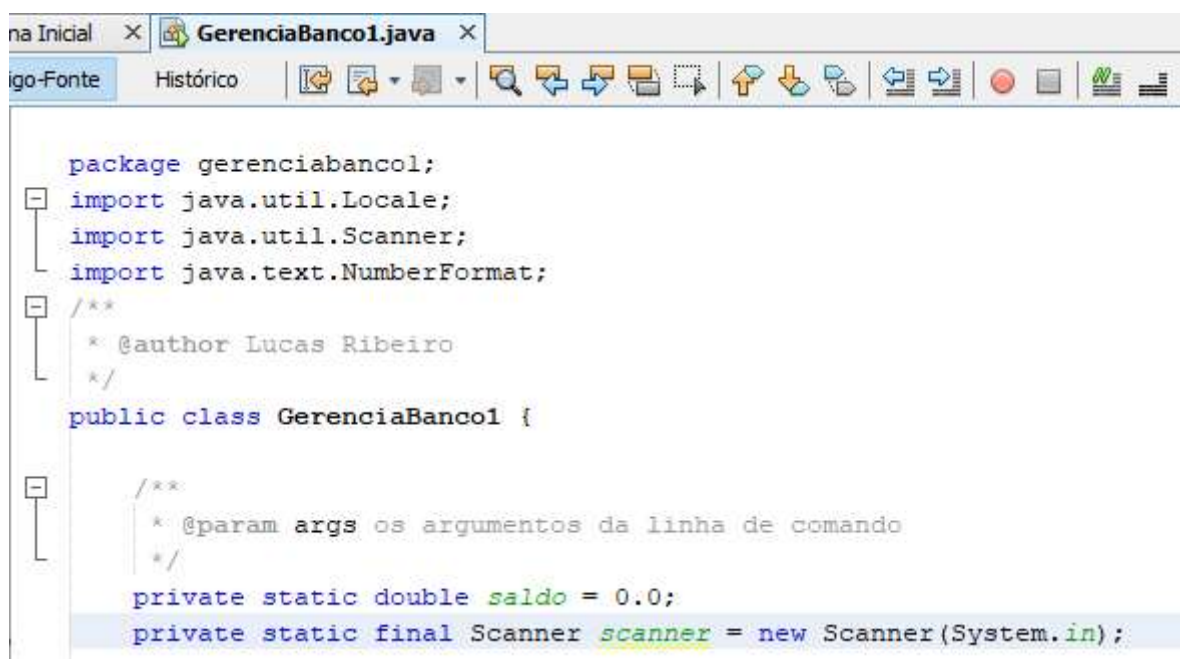
Durante o desenvolvimento, foi incluído casos de uso como consulta de saldo, depósito, saque e seleção de opções inválidas. Os resultados desses testes confirmaram a eficácia do sistema em lidar com diferentes situações.

Este projeto serve como um excelente exemplo de como os conceitos de programação em Java podem ser aplicados na prática para desenvolver um sistema de gerenciamento bancário funcional.

1 DESENVOLVIMENTO

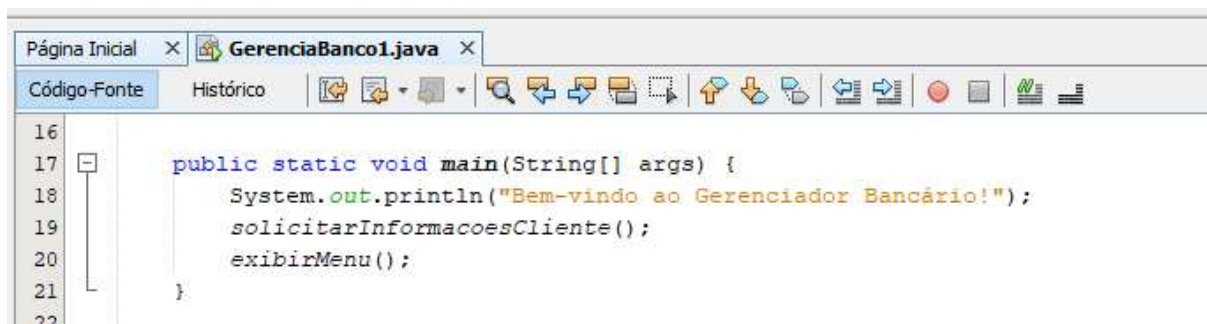
Para explicar como o programa funciona, vamos dividi-lo em 5 partes, começando por: 1 Importações e Declarações de Classe, 2 Método Principal, 3 Solicitar Informações do Cliente, 4 Exibir Menu, 5 Consultar Saldo, Realizar Depósito e Realizar Saque, 6 Formatar Saldo.

Começando com Importações e Declarações de Classe, a classe **GerenciaBanco1** tem um campo estático privado chamado `saldo` que é inicializado como 0.0. Também tem um objeto `Scanner` estático final chamado `scanner` que é usado para ler a entrada do usuário.



```
package gerenciabanco1;
import java.util.Locale;
import java.util.Scanner;
import java.text.NumberFormat;
/**
 * @author Lucas Ribeiro
 */
public class GerenciaBanco1 {
    /**
     * @param args os argumentos da linha de comando
     */
    private static double saldo = 0.0;
    private static final Scanner scanner = new Scanner(System.in);
```

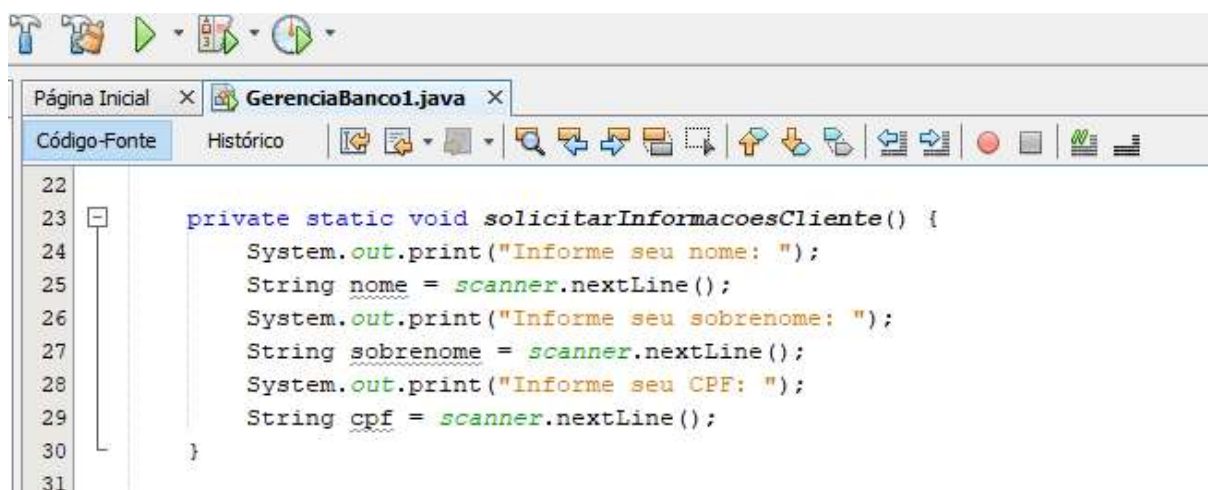
Seguindo para o Método Principal, o **main** é o ponto de entrada do programa. Ele imprime uma mensagem de boas-vindas e chama os métodos **solicitarInformacoesCliente** e **exibirMenu**.



The screenshot shows an IDE window titled 'GerenciaBanco1.java'. The 'Código-Fonte' tab is active. The code is as follows:

```
16  
17 public static void main(String[] args) {  
18     System.out.println("Bem-vindo ao Gerenciador Bancário!");  
19     solicitarInformacoesCliente();  
20     exibirMenu();  
21 }  
22
```

Já no método **solicitarInformacoesCliente** solicita ao usuário que insira seu nome, sobrenome e CPF. No entanto, essas informações não são usadas em nenhum outro lugar no programa.



The screenshot shows the same IDE window, but the code is now the **solicitarInformacoesCliente** method:

```
22  
23 private static void solicitarInformacoesCliente() {  
24     System.out.print("Informe seu nome: ");  
25     String nome = scanner.nextLine();  
26     System.out.print("Informe seu sobrenome: ");  
27     String sobrenome = scanner.nextLine();  
28     System.out.print("Informe seu CPF: ");  
29     String cpf = scanner.nextLine();  
30 }  
31
```

No método **exibirMenu** exibe um menu de opções para o usuário e lê a escolha do usuário. Dependendo da escolha do usuário, ele chama o método apropriado. Se o usuário escolher uma opção inválida, ele imprimirá uma mensagem de erro. O menu continuará sendo exibido até que o usuário escolha a opção para sair.

```

31
32 private static void exibirMenu() {
33     int opcao;
34     do {
35         System.out.println("\nMenu:");
36         System.out.println("1 - Consultar Saldo");
37         System.out.println("2 - Realizar Depósito");
38         System.out.println("3 - Realizar Saque");
39         System.out.println("4 - Sair");
40         System.out.print("Escolha uma opção: ");
41         opcao = scanner.nextInt();
42

```

```

43         switch (opcao) {
44             case 1:
45                 consultarSaldo();
46                 break;
47             case 2:
48                 realizarDeposito();
49                 break;
50             case 3:
51                 realizarSaque();
52                 break;
53             case 4:
54                 System.out.println("Obrigado por usar o Gerenciador Bancário. Até mais!");
55                 break;
56             default:
57                 System.out.println("Opção inválida. Tente novamente.");
58         }
59     } while (opcao != 4);
60
61     scanner.close();
62 }
63

```

Estes são os métodos que realizam as operações bancárias. **consultarSaldo** imprime o saldo atual do usuário. **realizarDeposito** solicita ao usuário que insira um valor de depósito, verifica se o valor é positivo e, em seguida, adiciona o valor ao saldo. **realizarSaque** solicita ao usuário que insira um valor de saque, verifica se o valor é positivo e se é menor ou igual ao saldo atual, e então subtrai o valor do saldo.

```

63
64 private static void consultarSaldo() {
65     System.out.println("Seu saldo atual é: " + formatarSaldo(saldo));
66 }
67
68 private static void realizarDeposito() {
69     System.out.print("Informe o valor do depósito: R$ ");
70     double deposito = scanner.nextDouble();
71     if (deposito > 0) {
72         saldo += deposito;
73         System.out.println("Depósito de " + formatarSaldo(deposito) + " realizado com sucesso!");
74         consultarSaldo();
75     } else {
76         System.out.println("O valor do depósito deve ser positivo.");
77     }
78 }
79
80 private static void realizarSaque() {
81     System.out.print("Informe o valor do saque: R$ ");
82     double saque = scanner.nextDouble();
83     if (saque > 0 && saque <= saldo) {
84         saldo -= saque;
85         System.out.println("Saque de " + formatarSaldo(saque) + " realizado com sucesso!");
86         consultarSaldo();
87     } else {
88         System.out.println("Valor de saque inválido ou saldo insuficiente.");
89     }
90 }

```

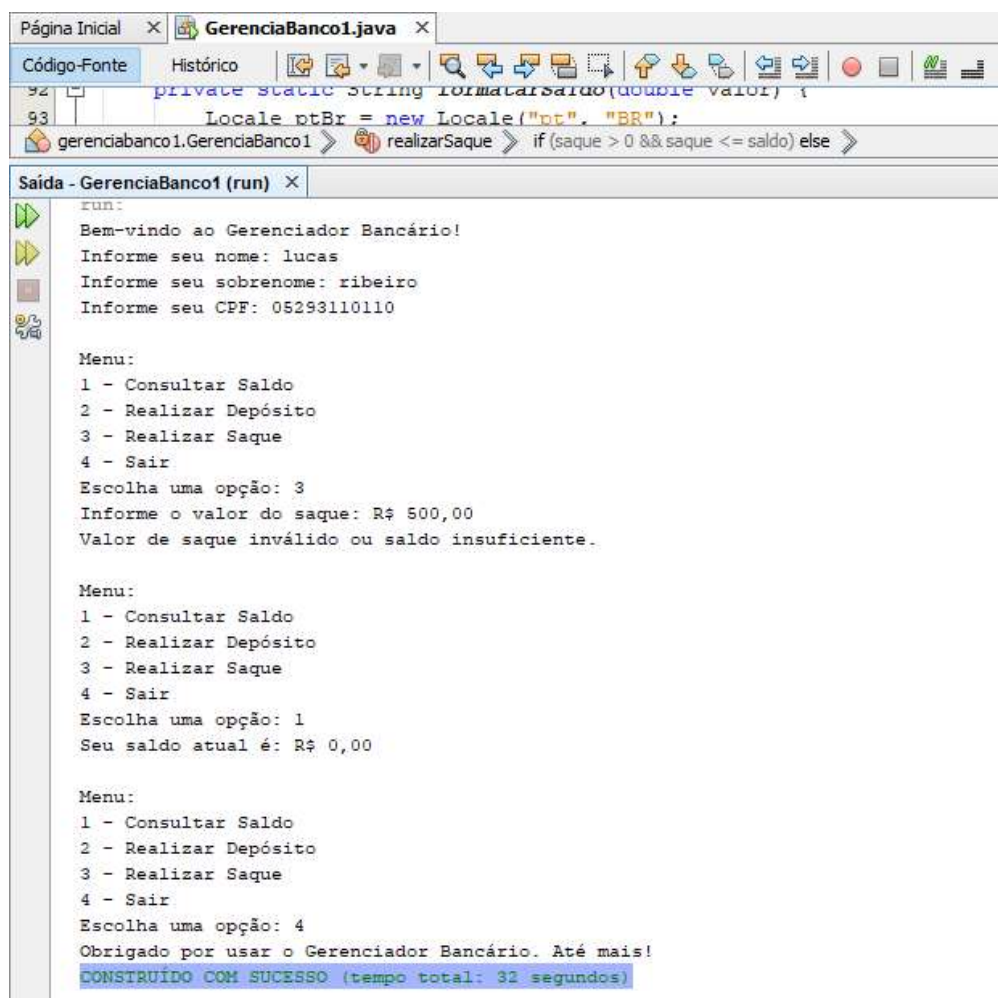
O método **formatarSaldo** recebe um valor **double** e retorna uma **string** formatada na moeda brasileira (Real).

```

90 }
91
92 private static String formatarSaldo(double valor) {
93     Locale ptBr = new Locale("pt", "BR");
94     NumberFormat formatador = NumberFormat.getCurrencyInstance(ptBr);
95     return formatador.format(valor);
96 }
97
98 }
99

```


Abaixo, demonstramos a tela do terminal onde é possível verificar o comportamento o código e seu resultado.



```
Página Inicial x GerenciaBanco1.java x
Código-Fonte Histórico
private static String formatarSaque(double valor) {
92 |
93 |     Locale ptBr = new Locale("pt", "BR");
gerenciabanco1.GerenciaBanco1 > realizarSaque > if (saque > 0 && saque <= saldo) else >

Saída - GerenciaBanco1 (run) x
run:
Bem-vindo ao Gerenciador Bancário!
Informe seu nome: lucas
Informe seu sobrenome: ribeiro
Informe seu CPF: 05293110110

Menu:
1 - Consultar Saldo
2 - Realizar Depósito
3 - Realizar Saque
4 - Sair
Escolha uma opção: 3
Informe o valor do saque: R$ 500,00
Valor de saque inválido ou saldo insuficiente.

Menu:
1 - Consultar Saldo
2 - Realizar Depósito
3 - Realizar Saque
4 - Sair
Escolha uma opção: 1
Seu saldo atual é: R$ 0,00

Menu:
1 - Consultar Saldo
2 - Realizar Depósito
3 - Realizar Saque
4 - Sair
Escolha uma opção: 4
Obrigado por usar o Gerenciador Bancário. Até mais!
CONSTRUÍDO COM SUCESSO (tempo total: 32 segundos)
```

1 CONCLUSÃO

O desenvolvimento do sistema de gerenciamento bancário em Java permitiu a aplicação prática dos conceitos teóricos aprendidos. O software desenvolvido, que permite operações bancárias básicas, demonstra como consultar saldo, realizar depósitos e saques. Isso foi alcançado através de uma interface de usuário simples e intuitiva que solicita informações ao usuário e exibe um menu de opções.

Este projeto serve como base para futuros desenvolvimentos na área de software bancário e sistemas de informação esta experiência foi uma jornada de aprendizado valiosa, permitindo a aplicação de conhecimentos teóricos em um cenário prático.