



# Anhanguera

UNIVERSIDADE PITÁGORAS UNOPAR  
POLO DE APOIO JUNDIAÍ-SP  
CURSO SUPERIOR DE **SISTEMAS DE INFORMAÇÃO**

NOME DO AUTOR: ALISSON HENRIQUE CORREIA

LINGUAGEM ORIENTADA A OBJETOS

São Paulo—SP  
2023

# LINGUAGEM ORIENTADA A OBJETOS

Trabalho apresentado à Universidade Pitágoras Unopar  
como requisito parcial à aprovação no  
Segundo Semestre do curso de  
Sistemas de informação.

São Paulo—SP  
2023

# SUMÁRIO

INTRODUÇÃO.....	4
DESENVOLVIMENTO.....	5 à 12
CONSIDERAÇÕES FINAIS.....	13
REFERÊNCIAS.....	14

## Introdução

Neste trabalho foi proposto a elaboração de um código utilizando a ferramenta **NetBeans**, seguindo as regras propostas pelo tutor, solucionando assim o exercício solicitado. Com o fim de entender a forma organizada que um código deve ser desenvolvido para um sistema bancário, com saída e entrada de dados de um usuário e com opções de consulta de saldo, depósito e saque de valores na mesma.

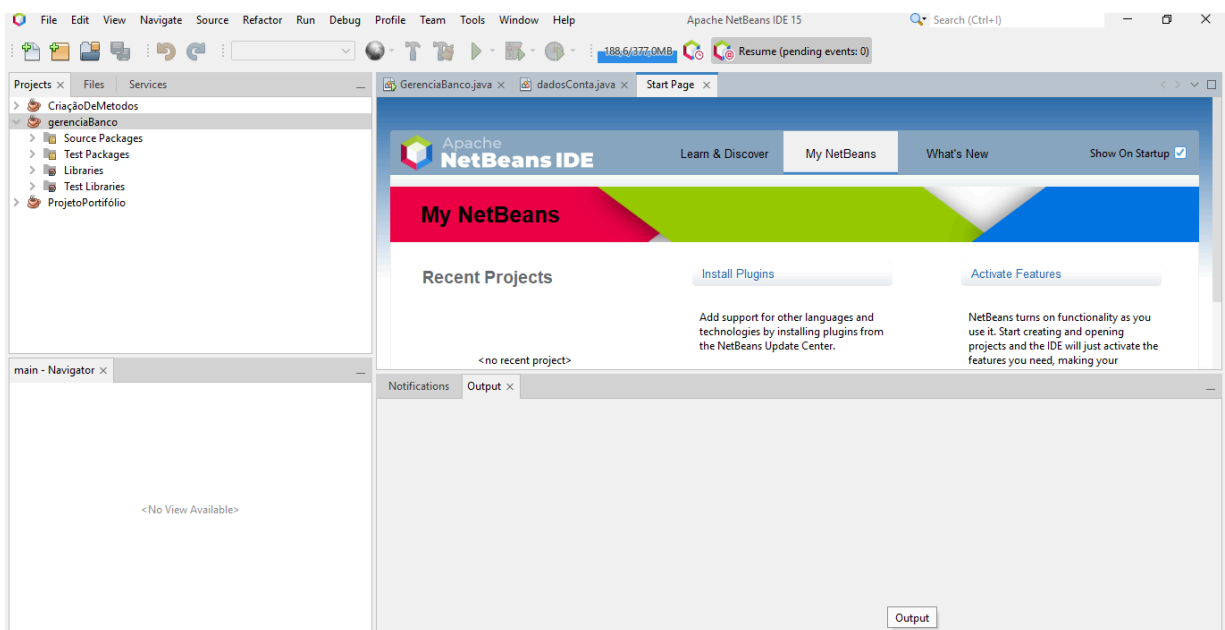
## Desenvolvimento

### Acesso a ferramenta e criação de um novo arquivo:

NetBeans:



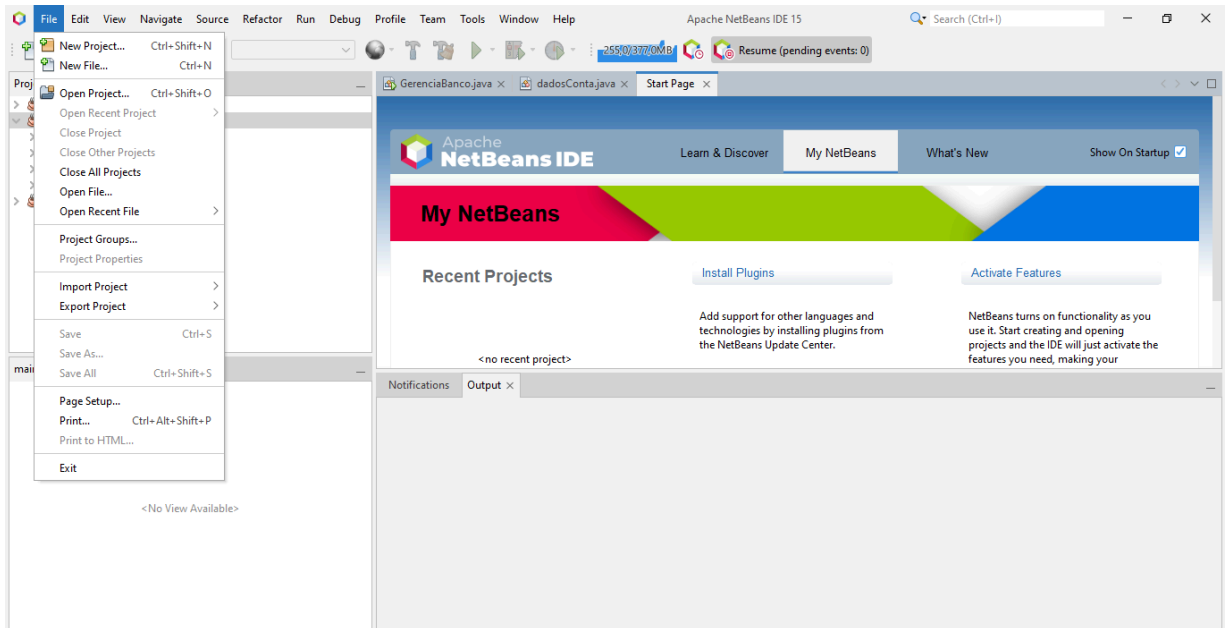
Após a instalação e configuração da ferramenta, essa é a apresentação de abertura ao acessar a IDE para iniciarmos o nosso primeiro projeto.



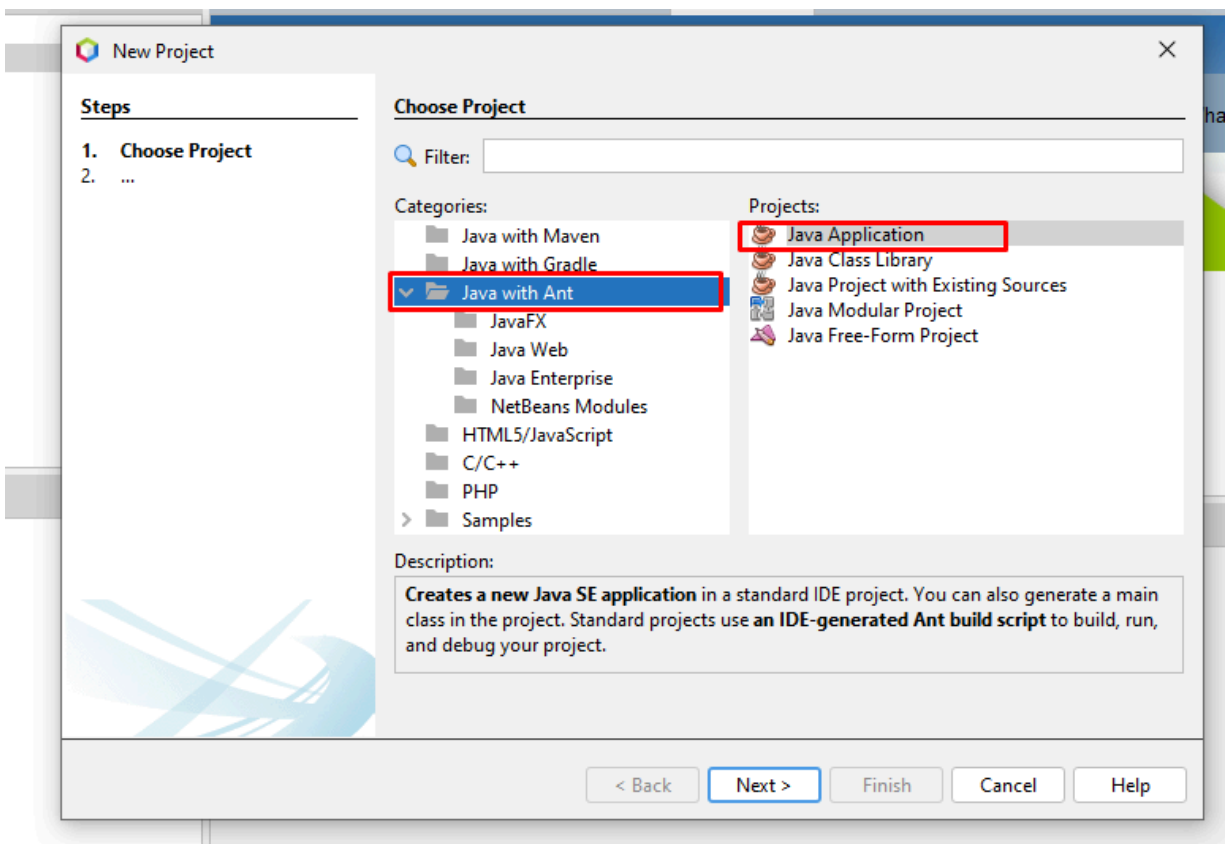
Essa é a interface da IDE aberta, sua página frontal e barra de ferramentas.

## Criando primeiro arquivo:

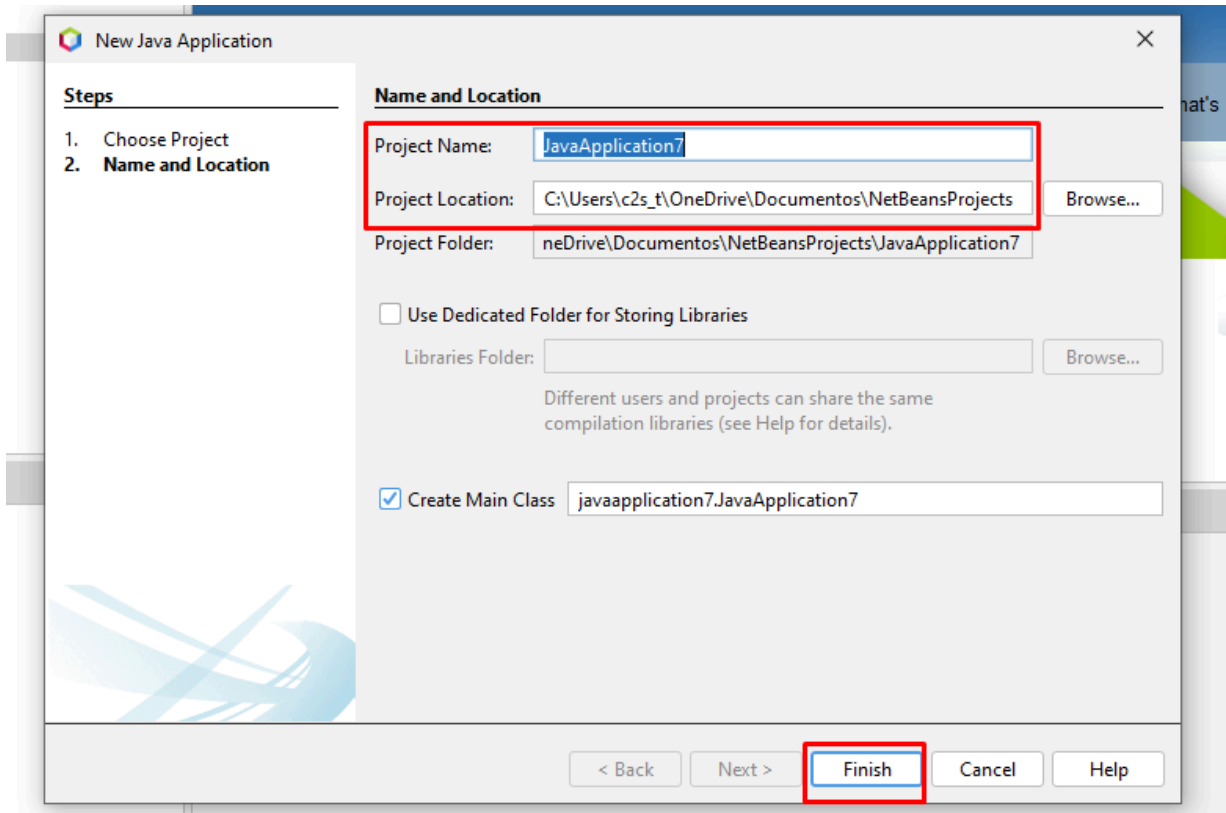
Clicando em File e depois em New Project.



## Selecionando a categoria e o tipo de projeto.

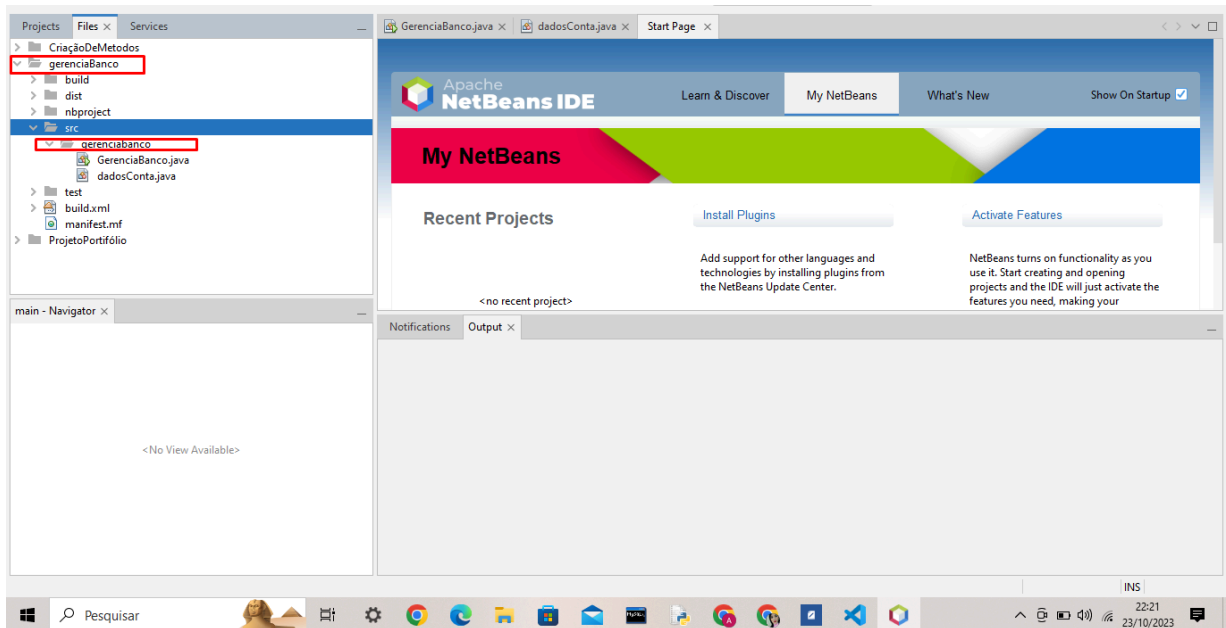


**Selecionando nome e local do arquivo do novo projeto.**



**Arquivos, paths e executáveis:**

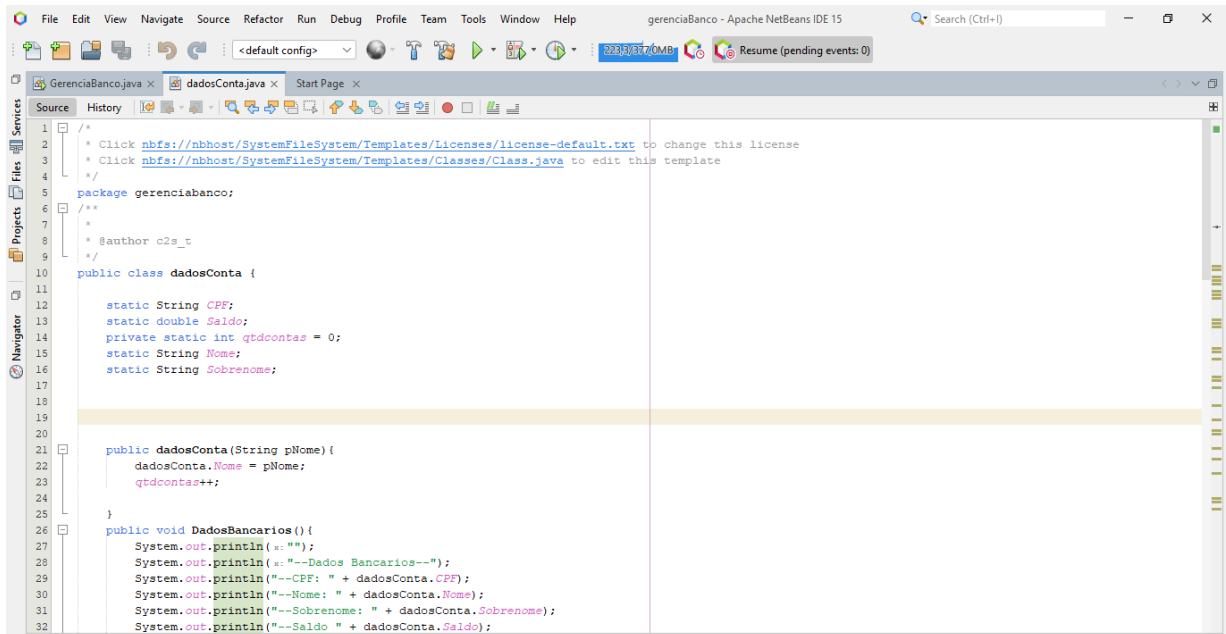
Todos os arquivos, classes e novas classes ficam disponíveis para visualização e criação na área destacada.



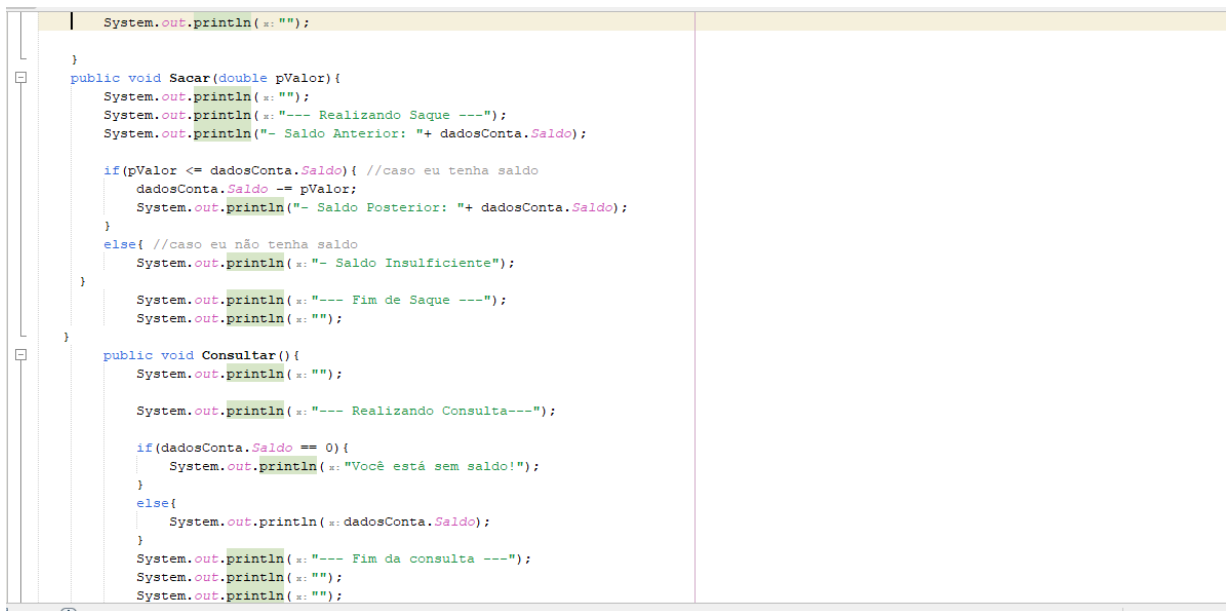
## Códigos:

Códigos: construtores, entradas e saídas de dados.

Construindo as variáveis, importando bibliotecas, criando packages e publicando classes que serão chamadas durante a entrada e saída de dados, controlando assim as atividades do usuário durante o uso da aplicação.



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package gerenciabanco;
6   /**
7    *
8    * @author c2e_t
9    */
10  public class dadosConta {
11
12      static String CPF;
13      static double Saldo;
14      private static int qtdcontas = 0;
15      static String Nome;
16      static String Sobrenome;
17
18
19
20
21      public dadosConta(String pNome) {
22          dadosConta.Nome = pNome;
23          qtdcontas++;
24      }
25
26      public void DadosBancarios() {
27          System.out.println(" ");
28          System.out.println(" ---Dados Bancarios---");
29          System.out.println(" ---CPF: " + dadosConta.CPF);
30          System.out.println(" ---Nome: " + dadosConta.Nome);
31          System.out.println(" ---Sobrenome: " + dadosConta.Sobrenome);
32          System.out.println(" ---Saldo " + dadosConta.Saldo);
```



```
          System.out.println(" ");
      }
      public void Sacar(double pValor) {
          System.out.println(" ");
          System.out.println(" --- Realizando Saque ---");
          System.out.println(" --- Saldo Anterior: " + dadosConta.Saldo);
          if (pValor <= dadosConta.Saldo) { //caso eu tenha saldo
              dadosConta.Saldo -= pValor;
              System.out.println(" --- Saldo Posterior: " + dadosConta.Saldo);
          }
          else { //caso eu não tenha saldo
              System.out.println(" --- Saldo Insuficiente");
          }
          System.out.println(" --- Fim de Saque ---");
          System.out.println(" ");
      }
      public void Consultar() {
          System.out.println(" ");
          System.out.println(" --- Realizando Consulta ---");
          if (dadosConta.Saldo == 0) {
              System.out.println(" --- Você está sem saldo!");
          }
          else {
              System.out.println(" --- " + dadosConta.Saldo);
          }
          System.out.println(" --- Fim da consulta ---");
          System.out.println(" ");
          System.out.println(" ");
      }
  }
```



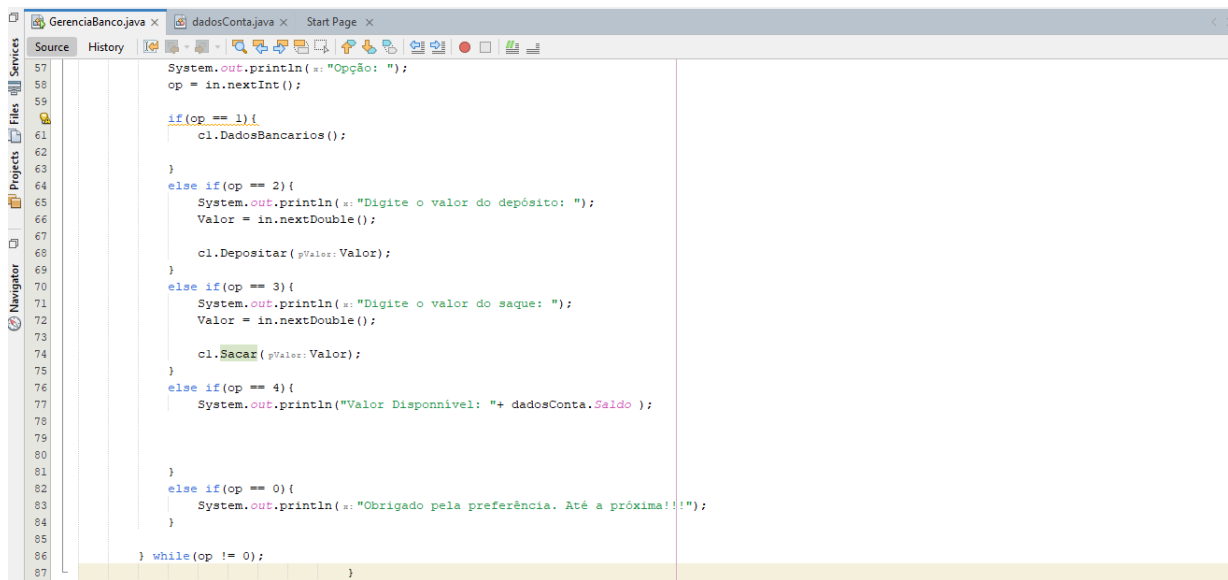
## Código de entrada e saída de dados.

Para o devido funcionamento dos códigos e comandos e uma melhor experiência do usuário, foram usadas estruturas condicionais e de repetição para controlar a sequência das etapas ou a permanência nela caso as respostas não fossem satisfatórias ou corretas.

```
1 package gerenciabanco;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author c2s_t
8  */
9 public class GerenciaBanco{
10
11     public static void main(String[] args){
12         Scanner in = new Scanner(System.in);
13         String aux;
14         int op;
15         double Valor;
16         int Senha = 0;
17
18         do{
19             System.out.println("Digite sua senha");
20             Senha = in.nextInt();
21
22             if(Senha != 0){
23                 System.out.println("Seja bem vindo!");
24                 break;
25             }
26             else{
27                 System.out.println("Senha Incorreta!");
28             }
29         }
30     }
31 }
```

```

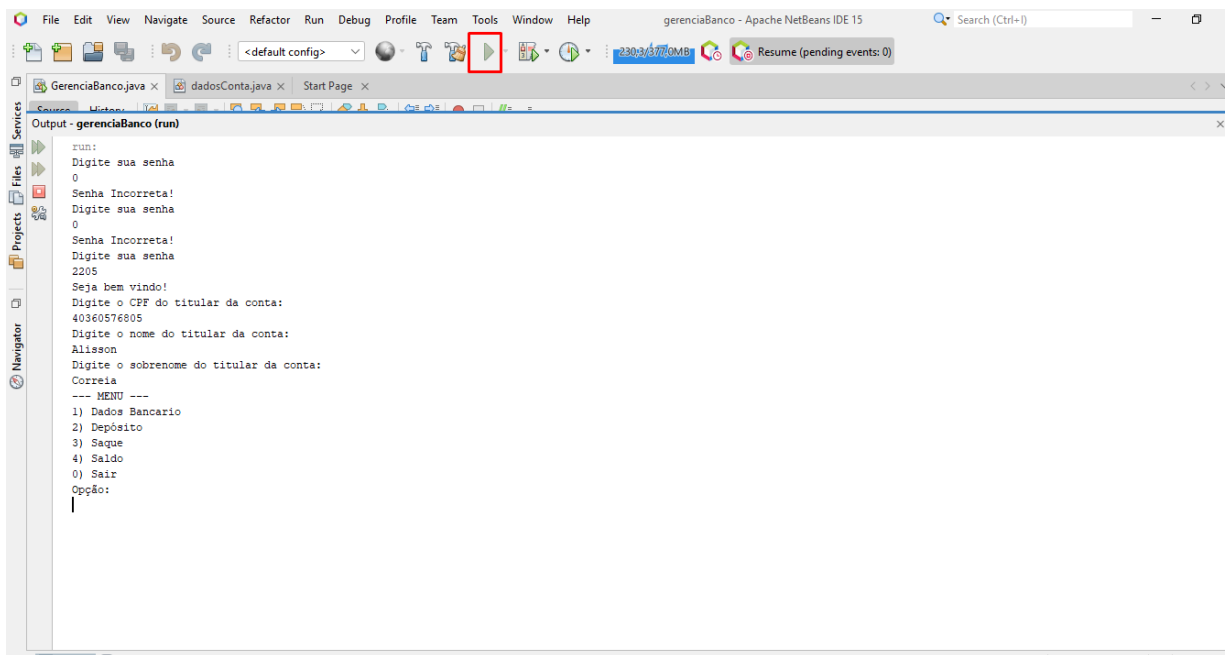
32 }while(Senha ==0);
33
34 System.out.println("Digite o CPF do titular da conta:");
35 dadosConta.CPF = in.next();
36 System.out.println("Digite o nome do titular da conta:");
37 dadosConta.Nome = in.next();
38 System.out.println("Digite o sobrenome do titular da conta:");
39 dadosConta.Sobrenome = in.next();
40
41 aux = in.nextLine();
42
43 dadosConta cl = new dadosConta(pNome: aux);
44
45 do{
46     System.out.println("--- MENU ---");
47     System.out.println("1) Dados Bancario");
48     System.out.println("2) Depósito");
49     System.out.println("3) Saque");
50     System.out.println("4) Saldo");
51     System.out.println("0) Sair");
52
53     System.out.println("Opção: ");
54     op = in.nextInt();
55
56     if(op == 1){
57         cl.DadosBancarios();
58     }
59 }
60 }
```



```
57 System.out.println("Opção: ");
58 op = in.nextInt();
59
60 if(op == 1){
61     cl.DadosBancarios();
62 }
63
64 else if(op == 2){
65     System.out.println("Digite o valor do depósito: ");
66     Valor = in.nextDouble();
67
68     cl.Depositar(pValor: Valor);
69 }
70
71 else if(op == 3){
72     System.out.println("Digite o valor do saque: ");
73     Valor = in.nextDouble();
74
75     cl.Sacar(pValor: Valor);
76 }
77
78 else if(op == 4){
79     System.out.println("Valor Disponivel: " + dadosConta.Saldo );
80 }
81
82 else if(op == 0){
83     System.out.println("Obrigado pela preferência. Até a próxima!!");
84 }
85
86 } while(op != 0);
87 }
```

## Resultado alcançado:

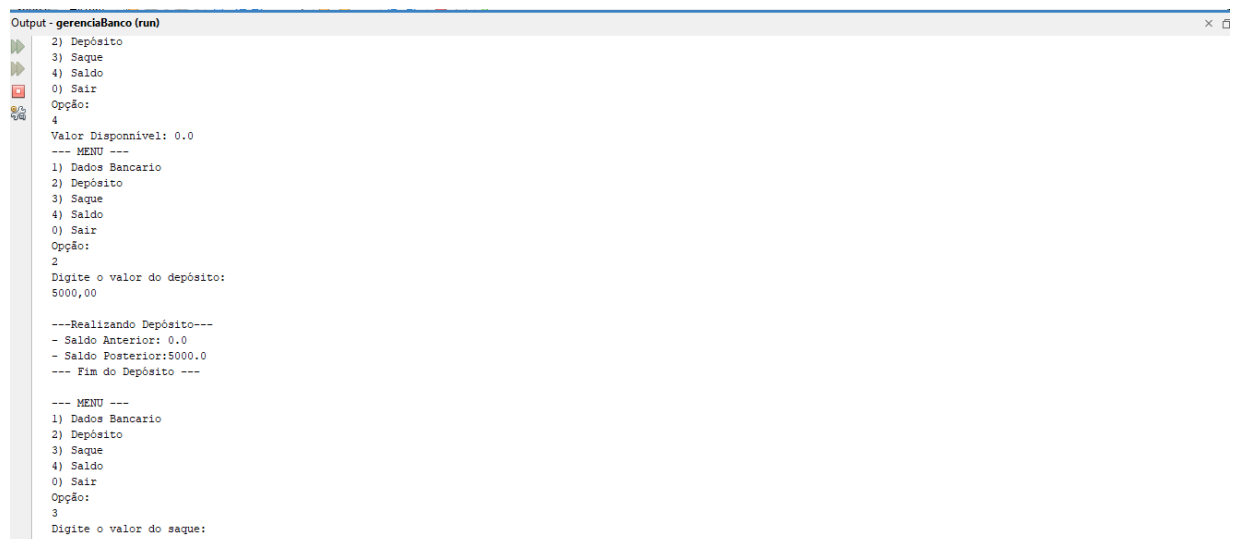
**seguir, temos o resultado alcançado, a conclusão do exercício requerido por meus tutores:** Ao clicarmos no botão executar, já vemos o código em seu pleno funcionamento. Uma mensagem solicitando a senha de acesso é apresentada e quando incorreta um aviso é apresentado, do contrário uma mensagem de boas-vindas indica que o acesso foi liberado e o próximo passo será liberado, permitindo a entrada dos dados pessoais do usuário.



```
run:
Digite sua senha
0
Senha Incorreta!
Digite sua senha
0
Senha Incorreta!
Digite sua senha
2205
Seja bem vindo!
Digite o CPF do titular da conta:
40360576805
Digite o nome do titular da conta:
Alisson
Digite o sobrenome do titular da conta:
Correia
--- MENU ---
1) Dados Bancario
2) Depósito
3) Saque
4) Saldo
0) Sair
Opção:
|
```

A seguir o Menu é apresentado, onde o usuário pode visualizar as opções, o que o permite escolher o tipo de atividade que ele deseja realizar no app bancário.

Veja a seguir a maneira que esse 'Menu de opções' reage ao ser interagido pelo usuário:



```
Output - gerenciaBanco (run)
2) Depósito
3) Saque
4) Saldo
0) Sair
Opção:
4
Valor Disponivel: 0.0
--- MENU ---
1) Dados Bancario
2) Depósito
3) Saque
4) Saldo
0) Sair
Opção:
2
Digite o valor do depósito:
5000,00
---Realizando Depósito---
- Saldo Anterior: 0.0
- Saldo Posterior:5000.0
--- Fim do Depósito ---
--- MENU ---
1) Dados Bancario
2) Depósito
3) Saque
4) Saldo
0) Sair
Opção:
3
Digite o valor do saque:
```

```
Output - gerenciaBanco (run)
1) Dados Bancario
2) Depósito
3) Saque
4) Saldo
0) Sair
Opção:
3
Digite o valor do saque:
2350,50

--- Realizando Saque ---
- Saldo Anterior: 5000.0
- Saldo Posterior: 2649.5
--- Fim de Saque ---

--- MENU ---
1) Dados Bancario
2) Depósito
3) Saque
4) Saldo
0) Sair
Opção:
1

--Dados Bancarios--
--CPF: 40360576805
--Nome:
--Sobrenome: Correia
--Saldo 2649.5
---Fim do Dados bancarios---
```

```
ierenciaBanco.java x dadosConta.java x Start Page x
ut - gerenciaBanco (run)

--- Realizando Saque ---
- Saldo Anterior: 5000.0
- Saldo Posterior: 2649.5
--- Fim de Saque ---

--- MENU ---
1) Dados Bancario
2) Depósito
3) Saque
4) Saldo
0) Sair
Opção:
1

--Dados Bancarios--
--CPF: 40360576805
--Nome:
--Sobrenome: Correia
--Saldo 2649.5
---Fim do Dados bancarios---

--- MENU ---
1) Dados Bancario
2) Depósito
3) Saque
4) Saldo
0) Sair
Opção:
0
Obrigado pela preferência. Até a próxima!!!
BUILD SUCCESSFUL (total time: 17 minutes 16 seconds)
```

Acima podemos notar a maneira que os dados são inseridos, registrados e apresentados conforme o usuário digita os valores, dados e realiza novas consultas em seu app de gerenciamento bancário.

## CONSIDERAÇÕES FINAIS

Nesse portfólio que foi solicitado pelo Tutor, consegui desenvolver um trabalho que me trouxe mais conhecimento sobre a matéria, durante o desenvolvimento do trabalho tive algumas dificuldades, mas com o conhecimento que é apresentado nas aulas e em outros cursos que venho estudando com professor Leonardo Santiago Sidon da Rocha no site YouTube, pude desenvolver e finalizar o portfólio.

## Referências

Aula de **Linguagem Orientada a Objeto**

Tutor: Leonardo Santiago Sidon da Rocha

Conteúdo fornecido por Faculdade Anhanguera

**Curso de Java básico**

Tutor: David Paolini Develly

Conteúdo: <https://www.youtube.com/watch?v=jUussoBnZfk>