

 Estácio	<p align="center">UNIVERSIDADE ESTÁCIO DE SÁ</p> <p align="center">POLO INDAIATUBA – INDAIATUBA/SP</p> <p align="center">DESENVOLVIMENTO FULL STACK - 22.3</p> <p align="center">Relatório da Missão Prática Nível 1 Mundo 3</p>
Aluno:	SAULO HENRIQUE DOS SANTOS
Professor:	Rodrigo Dias
Repositório:	_____

Implementação de um cadastro de clientes em modo texto, com persistência em arquivos, baseado na tecnologia Java.

👉 1º Procedimento | Criação das Entidades e Sistema de Persistência

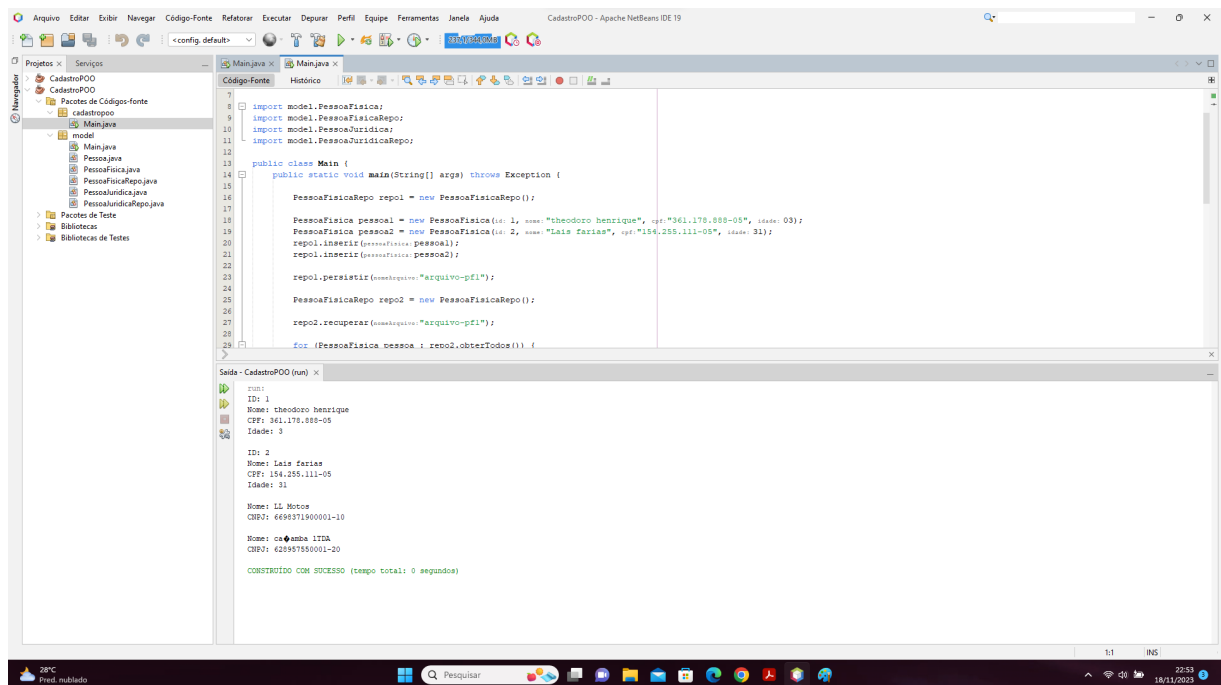
1. Objetivos da prática:

- ☐ Utilizar herança e polimorfismo na definição de entidades.
- ☐ Utilizar persistência de objetos em arquivos binários.
- ☐ Implementar uma interface cadastral em modo texto.
- ☐ Utilizar o controle de exceções da plataforma Java.
- ☐ No final do projeto, o aluno terá implementado um sistema cadastral em Java, utilizando os recursos da programação orientada a objetos e a persistência em arquivos binários.

2. Todos os códigos solicitados neste roteiro de aula:

<https://github.com/>

3. Resultados da execução dos códigos PT1:



Análise e Conclusão:

a) Quais as vantagens e desvantagens do uso de herança?

Vantagens:

Reutilização de código

Extensibilidade, ou seja, criação de novas classes baseadas em classes existentes.

Desvantagens:

Código menos flexível e mais difícil de manter, por conta do acoplamento.

Fragilidade da hierarquia

Complexidade

b) Por que a interface `Serializable` é necessária ao efetuar persistência em arquivos binários?

A interface `Serializable` é necessária ao efetuar persistência em arquivos binários em Java pois permite que os objetos de uma classe sejam convertidos em uma sequência de bytes, tornando-os serializáveis, o que facilita sua gravação em um arquivo binário.

c) Como o paradigma funcional é utilizado pela API stream no Java?

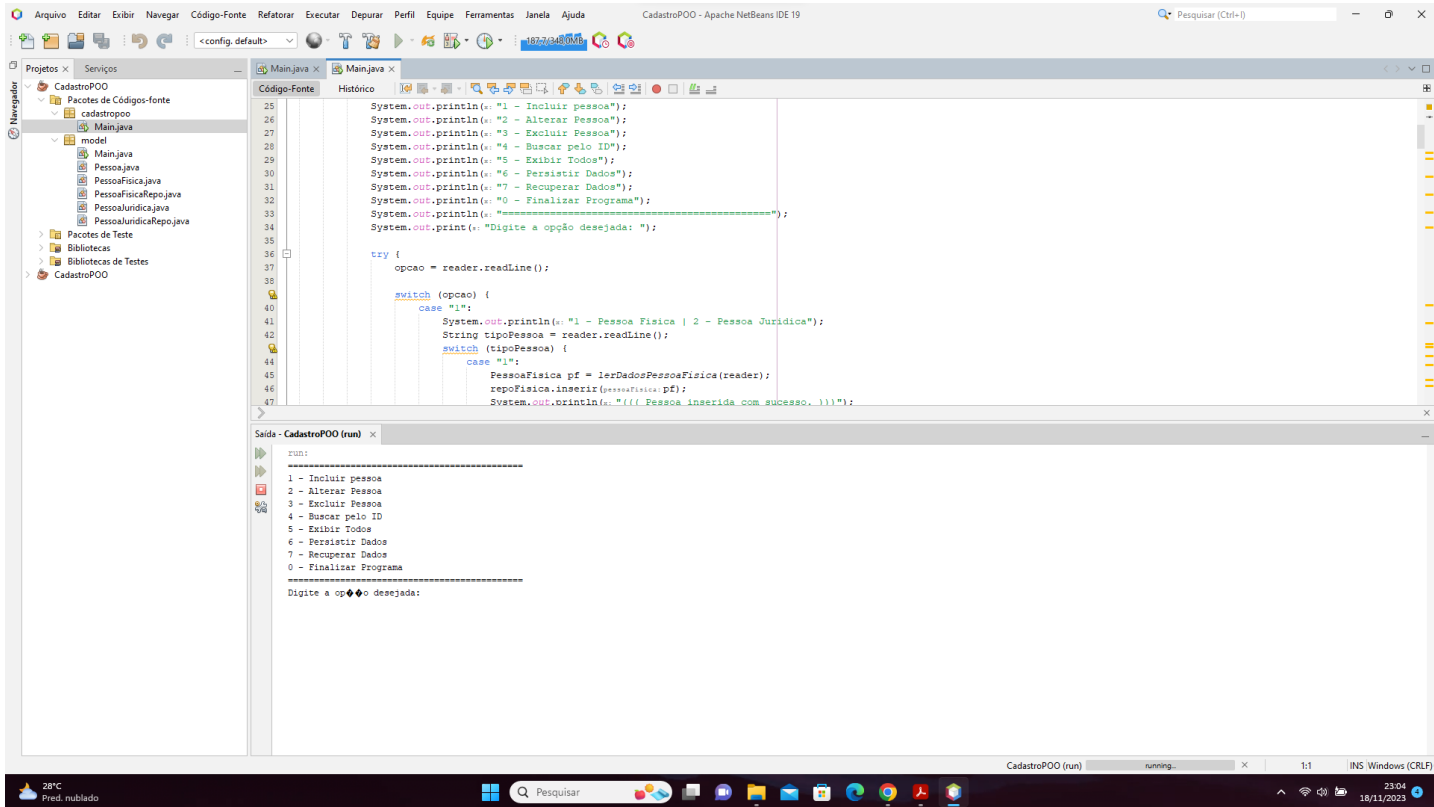
A API Stream no Java utiliza o paradigma funcional ao permitir operações de transformação e filtragem de dados em coleções de forma declarativa, usando funções lambda e expressões funcionais. Isso promove código mais conciso e legível, seguindo os princípios do paradigma funcional, como imutabilidade e composição de funções.

d) Quando trabalhamos com Java, qual padrão de desenvolvimento é adotado na persistência de dados em arquivos?

Em Java, um padrão comum de desenvolvimento para a persistência de dados em arquivos é o uso do padrão de projeto "Serialização". A serialização permite que objetos Java sejam convertidos em uma sequência de bytes e, em seguida, gravados em arquivos binários. Isso facilita a persistência e a recuperação de objetos e seus dados em arquivos, tornando-os portáteis e eficientes para armazenamento e transporte. Para implementar a serialização, é comum utilizar as interfaces **`Serializable`** e **`ObjectInputStream/ObjectOutputStream`**.

👉 2º Procedimento | Criação do Cadastro em Modo Texto

1. Resultados da execução dos códigos PT2:



```
System.out.println("1 - Incluir pessoa");
System.out.println("2 - Alterar Pessoa");
System.out.println("3 - Excluir Pessoa");
System.out.println("4 - Buscar pelo ID");
System.out.println("5 - Exibir Todos");
System.out.println("6 - Persistir Dados");
System.out.println("7 - Recuperar Dados");
System.out.println("0 - Finalizar Programa");
System.out.println("=====");
System.out.print("Digite a opção desejada: ");

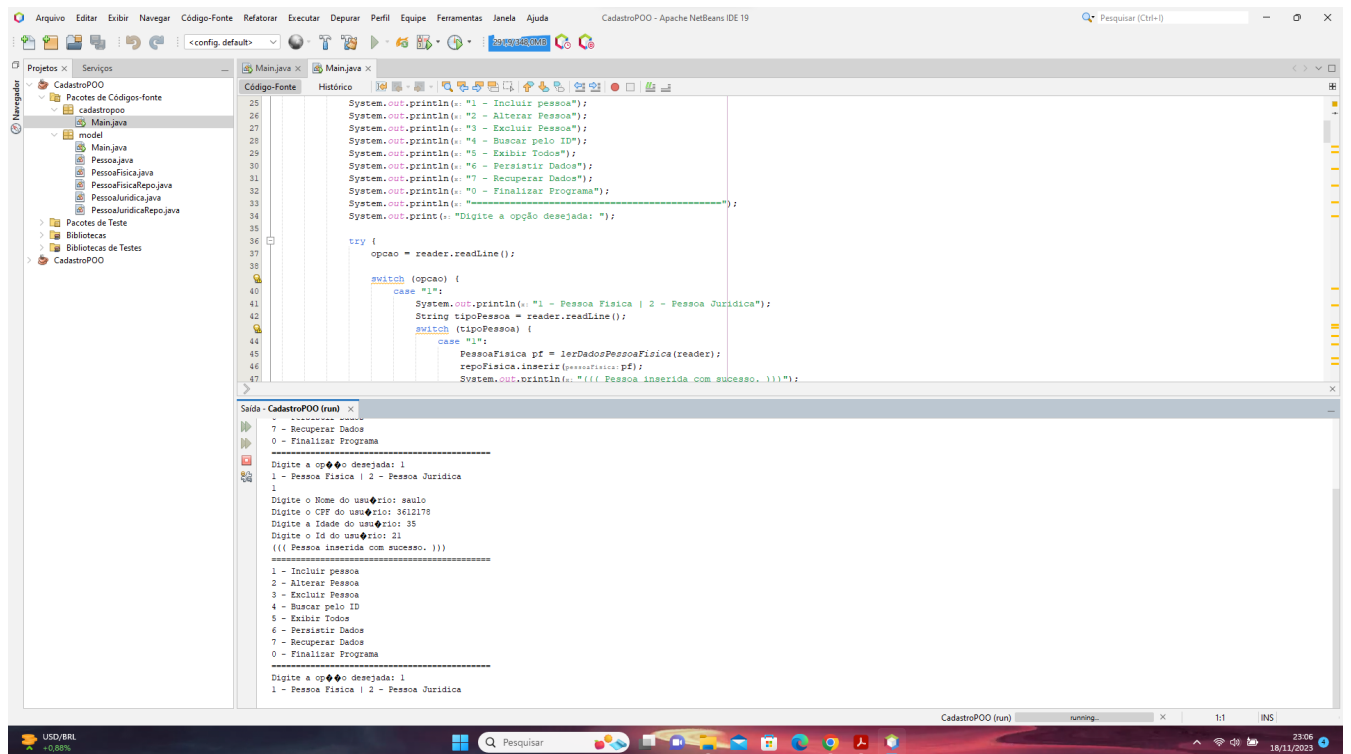
try {
    opcao = reader.readLine();

    switch (opcao) {
        case "1":
            System.out.println("1 - Pessoa Física | 2 - Pessoa Jurídica");
            String tipoPessoa = reader.readLine();
            switch (tipoPessoa) {
                case "1":
                    PessoaFisica pf = lerDadosPessoaFisica(reader);
                    repoFisica.inserir(pessoaFisica: pf);
                    System.out.println("((( Pessoa inserida com sucesso. )))");
            }
    }
}
```

run:

1 - Incluir pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====

Digite a opção desejada:



```
System.out.println("1 - Incluir pessoa");
System.out.println("2 - Alterar Pessoa");
System.out.println("3 - Excluir Pessoa");
System.out.println("4 - Buscar pelo ID");
System.out.println("5 - Exibir Todos");
System.out.println("6 - Persistir Dados");
System.out.println("7 - Recuperar Dados");
System.out.println("0 - Finalizar Programa");
System.out.println("=====");
System.out.print("Digite a opção desejada: ");

try {
    opcao = reader.readLine();

    switch (opcao) {
        case "1":
            System.out.println("1 - Pessoa Física | 2 - Pessoa Jurídica");
            String tipoPessoa = reader.readLine();
            switch (tipoPessoa) {
                case "1":
                    PessoaFisica pf = lerDadosPessoaFisica(reader);
                    repoFisica.inserir(pessoaFisica: pf);
                    System.out.println("((( Pessoa inserida com sucesso. )))");
            }
    }
}
```

run:

7 - Recuperar Dados
0 - Finalizar Programa
=====

Digite a opção desejada: 1

1 - Pessoa Física | 2 - Pessoa Jurídica

1

Digite o Nome do usuário: aulo

Digite o CPF do usuário: 3612178

Digite a Idade do usuário: 35

Digite o Id do usuário: 21

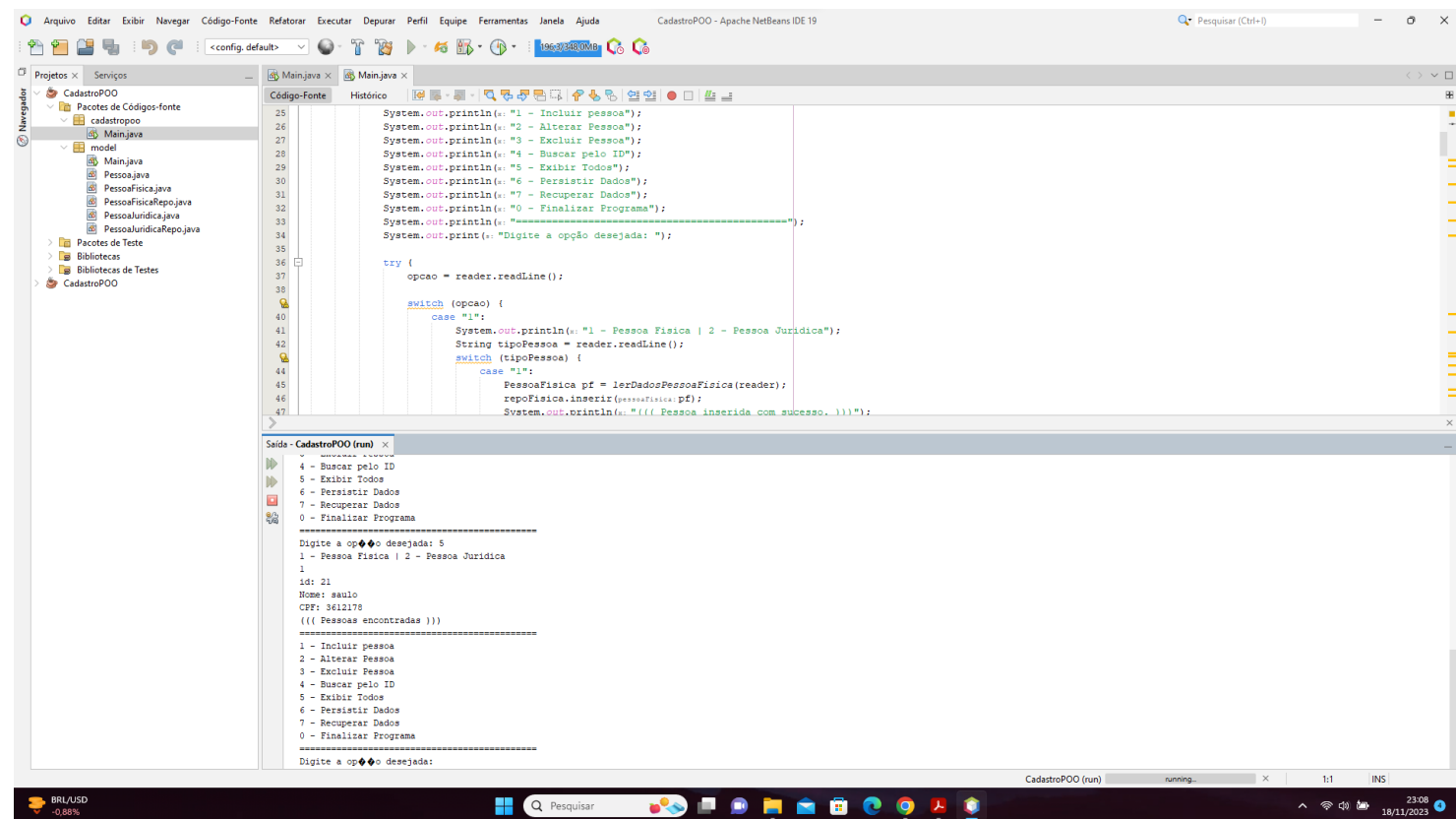
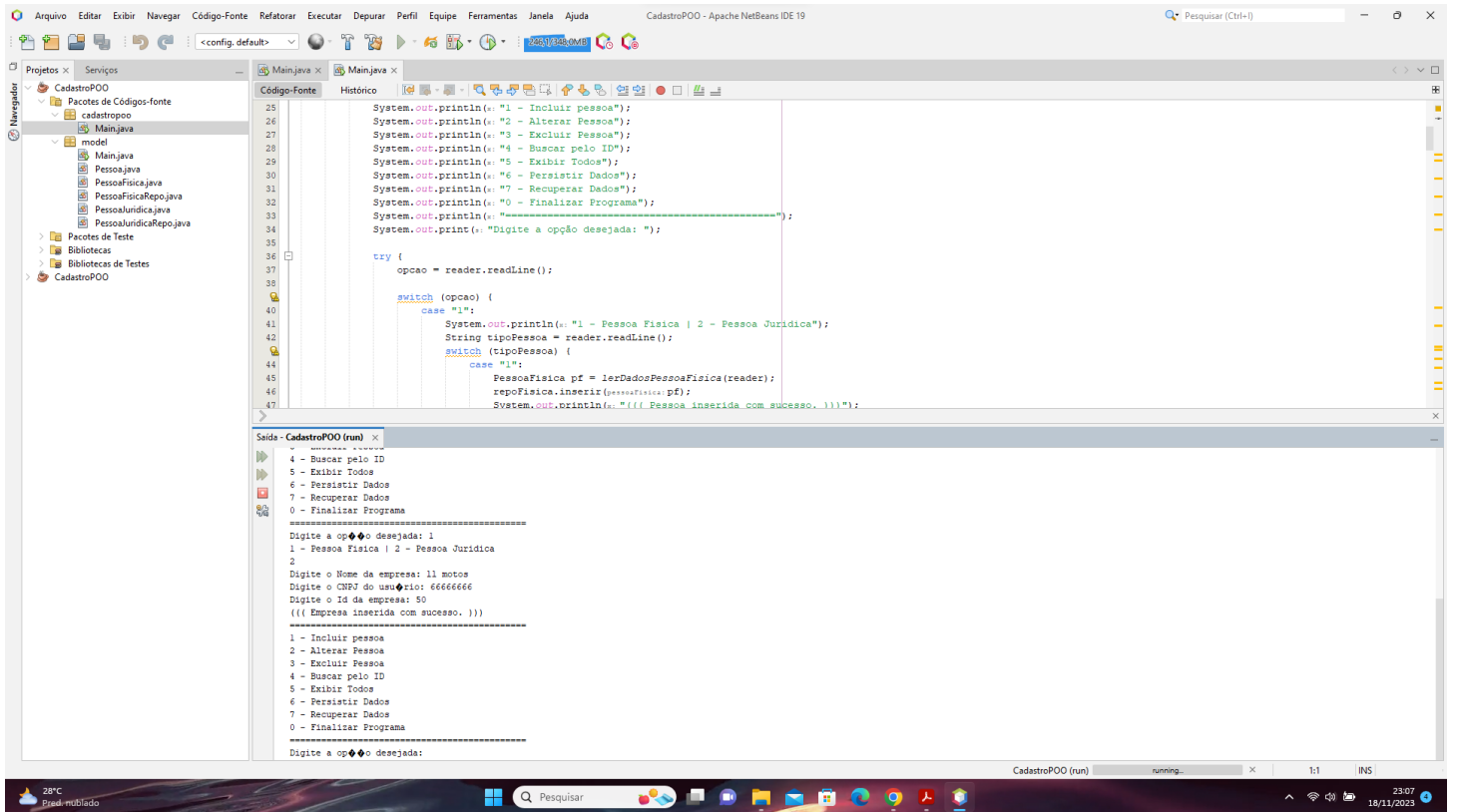
(((Pessoa inserida com sucesso.)))

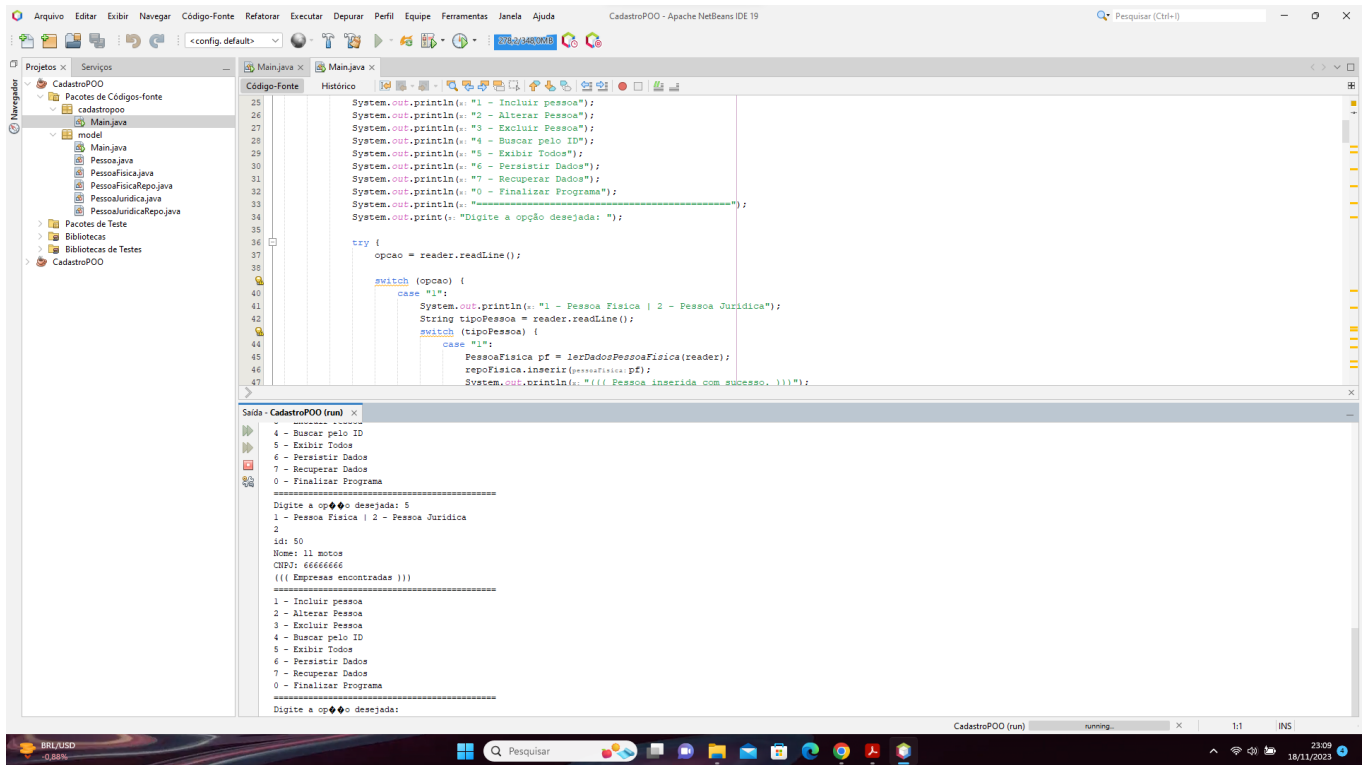
=====

1 - Incluir pessoa
2 - Alterar Pessoa
3 - Excluir Pessoa
4 - Buscar pelo ID
5 - Exibir Todos
6 - Persistir Dados
7 - Recuperar Dados
0 - Finalizar Programa
=====

Digite a opção desejada: 1

1 - Pessoa Física | 2 - Pessoa Jurídica





Análise e Conclusão:

a) O que são elementos estáticos e qual o motivo para o método main adotar esse modificador?

Elementos estáticos em Java são associados à classe em vez de instâncias individuais dessa classe. O método "main" é declarado como estático para que possa ser chamado sem a necessidade de criar um objeto da classe. Isso permite que seja o ponto de entrada do programa, sendo invocado diretamente pela JVM (Java Virtual Machine) durante a execução, sem a necessidade de criar uma instância da classe que contém o método.

b) Para que serve a classe Scanner?

A classe `Scanner` em Java é usada para ler entrada de dados do usuário ou de outros fluxos, como arquivos, de maneira simples e conveniente.

c) Como o uso de classes de repositório impactou na organização do código?

O uso de classes de repositório melhora a organização do código,

separando a lógica de acesso a dados da lógica de negócios,
promovendo maior reutilização, testabilidade e clareza no código.