****

**FACULDADE ESTÁCIO**

**CÂMPUS VOLTA REDONDA – RJ**

**DESENVOLVIMENTO FULL STACK**

**DISCIPLINA – INICIANDO O CAMINHO PELO JAVA**

**TURMA – 2023.2**

**SEMESTRE – 3**

**VOLTA REDONDA, JUNHO 2024.**

**DESENVOLVIMENTO FULL STACK**

**DISCIPLINA – INICIANDO O CAMINHO PELO JAVA**

**TURMA – 2023.2**

**SEMESTRE – 3**

**ALUNO – BRUNO SAMPAIO BASTOS**

**TUTOR – MARIA MANSO**

**GITHUB - https://github.com/BrunoTI-Code?tab=repositories**

**VOLTA REDONDA, JUNHO 2024.**

**Resumo**

O código apresentado implementa um sistema de cadastro de pessoas físicas e jurídicas utilizando a linguagem Java. O sistema permite incluir, alterar, excluir, exibir pelo ID, exibir todos, salvar e recuperar dados de pessoas físicas e jurídicas. Para isso, são utilizadas classes de repositório (PessoaFisicaRepo e PessoaJuridicaRepo) que armazenam os dados e oferecem métodos para manipulá-los. A classe CadastroPOO contém o método main, que é responsável por exibir o menu de opções e gerenciar a interação com o usuário através da classe Scanner.

**SUMARIO**

[1 INTRODUÇÃO 2](#_Toc349561004)

[2 Cadastro poo 4](#_Toc1527934396)

[2.1 Código Cadastro POO 4](#_Toc1408944320)

[3 model 7](#_Toc995743184)

[3.1 Pessoa.java 7](#_Toc516837052)

[3.2 Pessoa fisica.java 8](#_Toc1415602328)

[3.3 Pessoa fisica repo.java 9](#_Toc2078429857)

[3.4 Pessoa juridica.java 10](#_Toc1883386379)

[3.5 Pessoa juridica repo.java 11](#_Toc1128005263)

[4 resultados da execução dos códigos 11](#_Toc1563016209)

[4.1 Incluir 12](#_Toc1735354482)

[4.2 Alterar 13](#_Toc38177486)

[4.3 Excluir 14](#_Toc730334653)

[4.4 Exibir pelo id 14](#_Toc35758163)

[4.5 Exibir todos 15](#_Toc978876506)

[4.6 Salvar dados 16](#_Toc982899018)

[4.7 Recuperar dados 17](#_Toc1311124940)

[5 ANALISE 19](#_Toc1656412097)

[5.1 PARA QUE SERVE A CLASSE SCANNER? 19](#_Toc360599610)

[5.2 PARA QUE SERVE A CLASSE SCANNER? 19](#_Toc1792568366)

[5.3 COMO O USO DE CLASSES DE REPOSITÓRIO IMPACTOU NA ORGANIZAÇÃO DO CÓDIGO? 20](#_Toc117735467)

[6 Conclusão 21](#_Toc634060347)

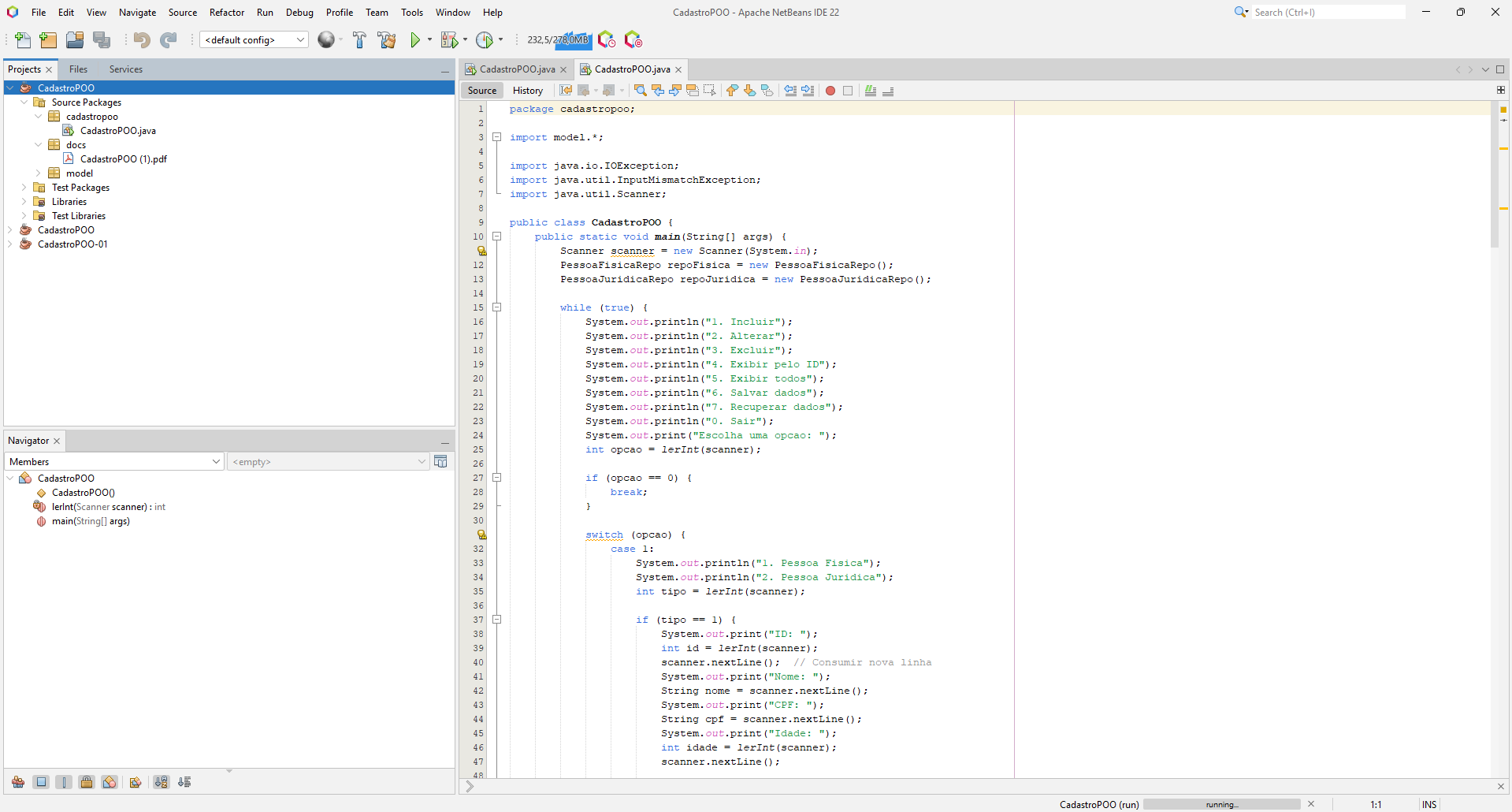
[7 REFERÊNCIAS 22](#_Toc827685843)

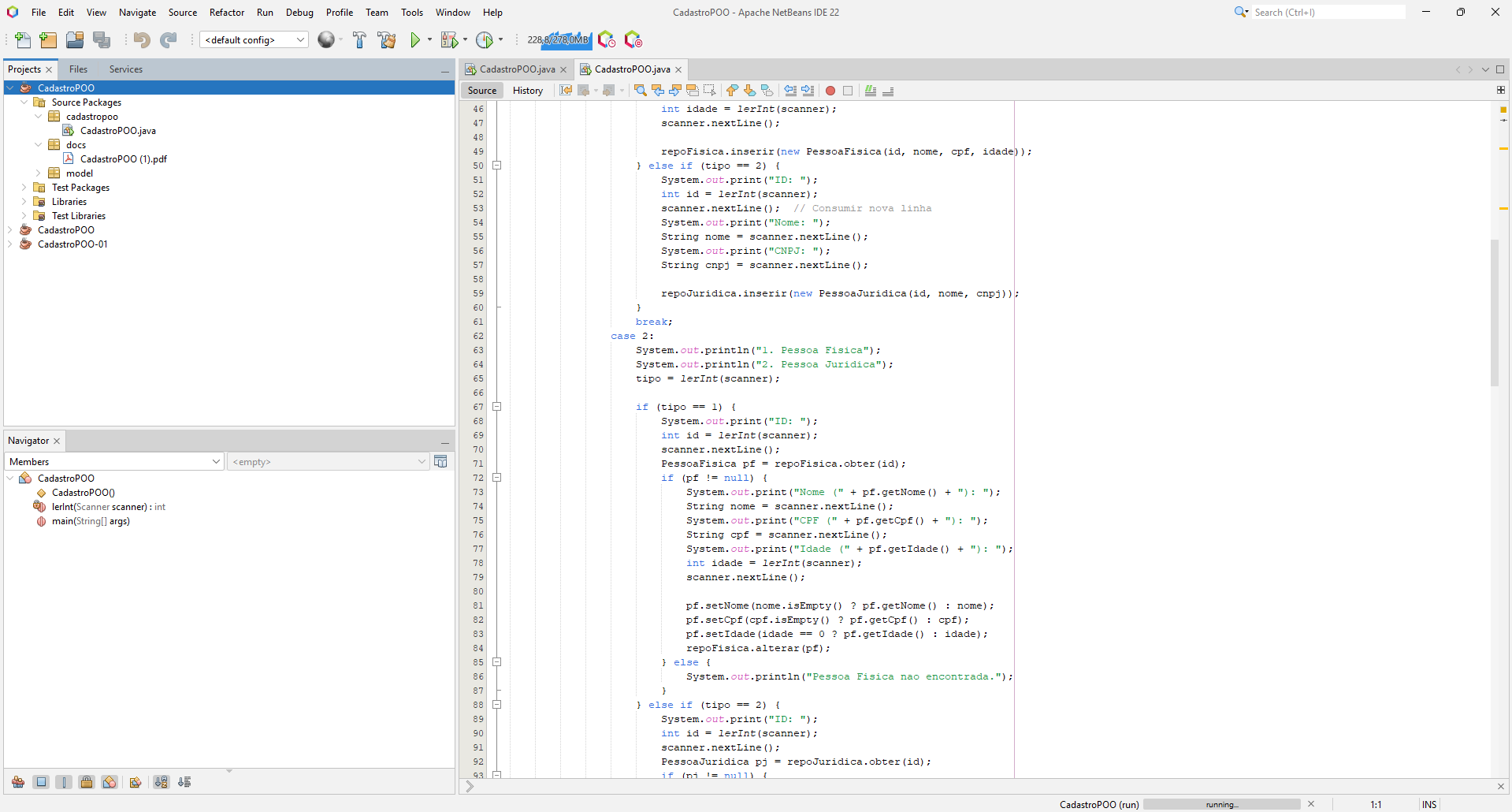
# INTRODUÇÃO

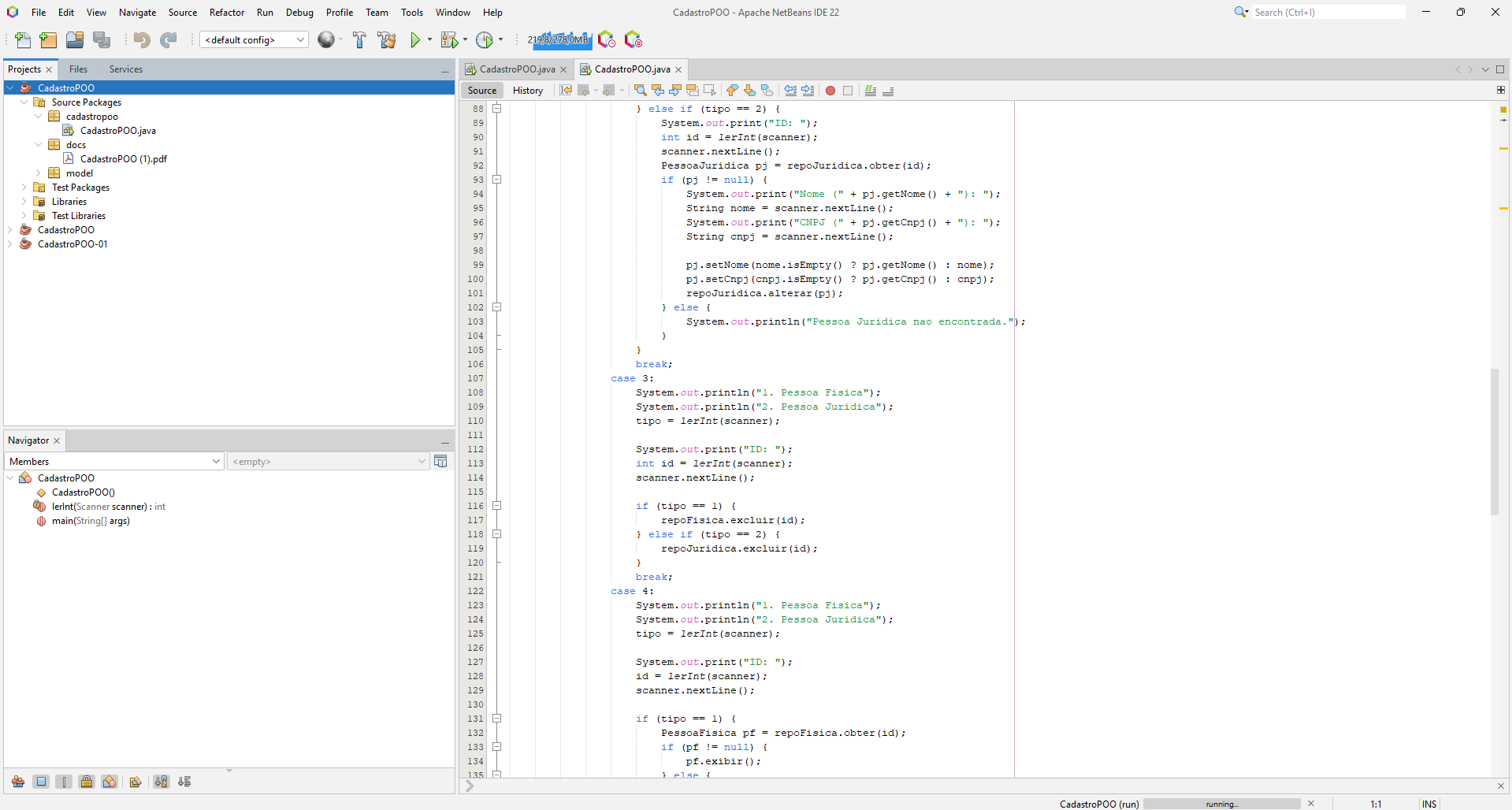
Este projeto descreve a criação de um sistema para cadastro de pessoas físicas e jurídicas usando a linguagem de programação Java. O sistema foi desenhado para ser eficiente e claro, permitindo operações como adicionar, alterar, excluir, exibir e manter dados. A implementação utiliza conceitos essenciais de programação orientada a objetos (POO), incluindo encapsulamento, modularização e reutilização de código, facilitando a manutenção e a expansão do sistema. O ponto de entrada da aplicação é o método main, marcado como estático, permitindo que a JVM (Java Virtual Machine) o chame diretamente sem necessidade de instanciar a classe CadastroPOO. A interação com o usuário é gerenciada pela classe Scanner, que captura as entradas do usuário e facilita a navegação pelas várias opções do menu. As operações de manipulação de dados são centralizadas em classes de repositório (PessoaFisicaRepo e PessoaJuridicaRepo), que encapsulam a lógica de acesso a dados e promovem uma separação clara entre a lógica de negócios e a interface do usuário. Esta estrutura não apenas melhora a legibilidade do código, mas também segue o princípio da responsabilidade única, tornando cada classe responsável por uma parte específica do sistema. Este documento também discutirá os elementos estáticos e seu papel no método main, a funcionalidade da classe Scanner e o impacto positivo do uso de classes de repositório na organização do código.

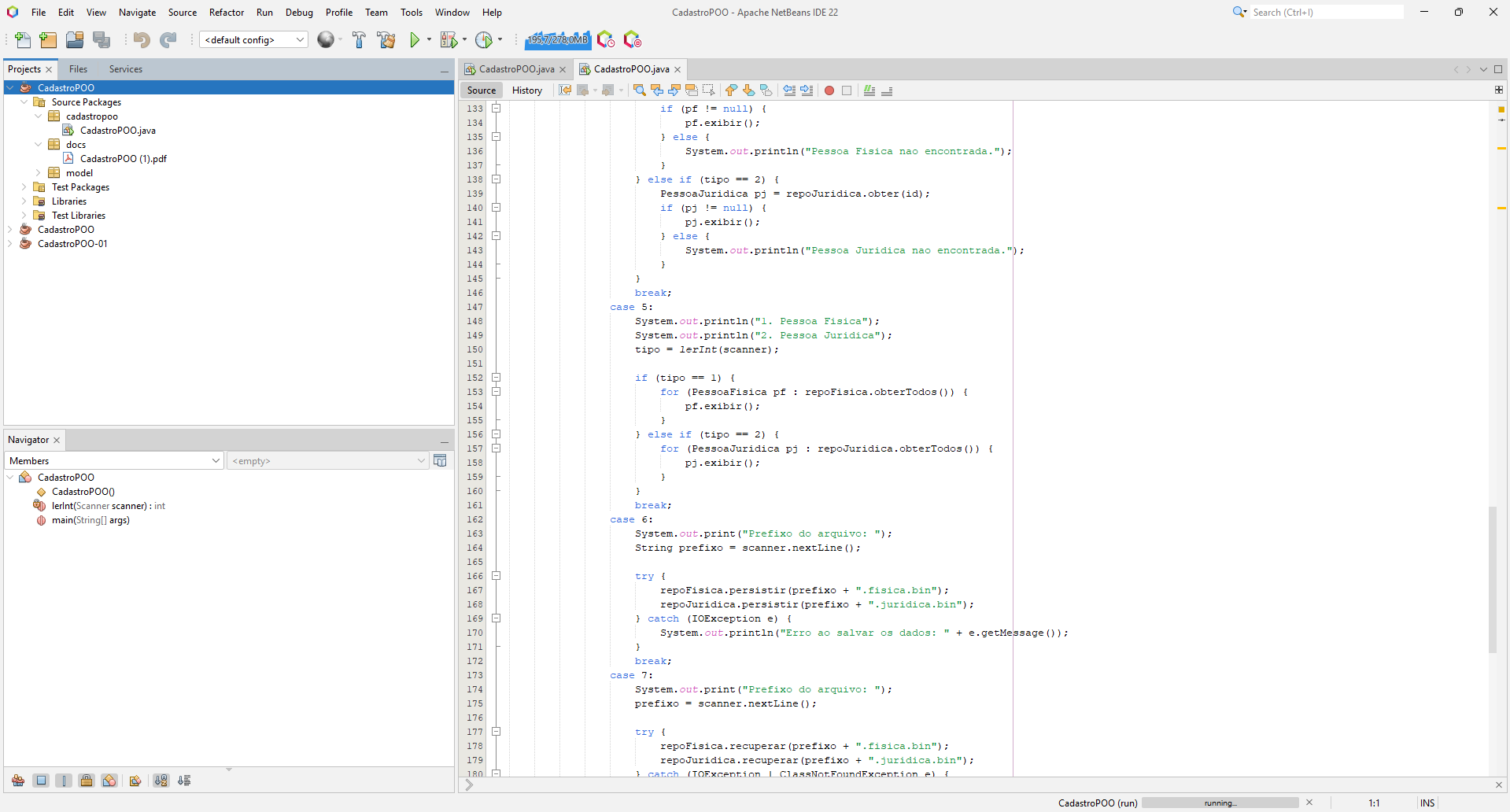
# Cadastro poo

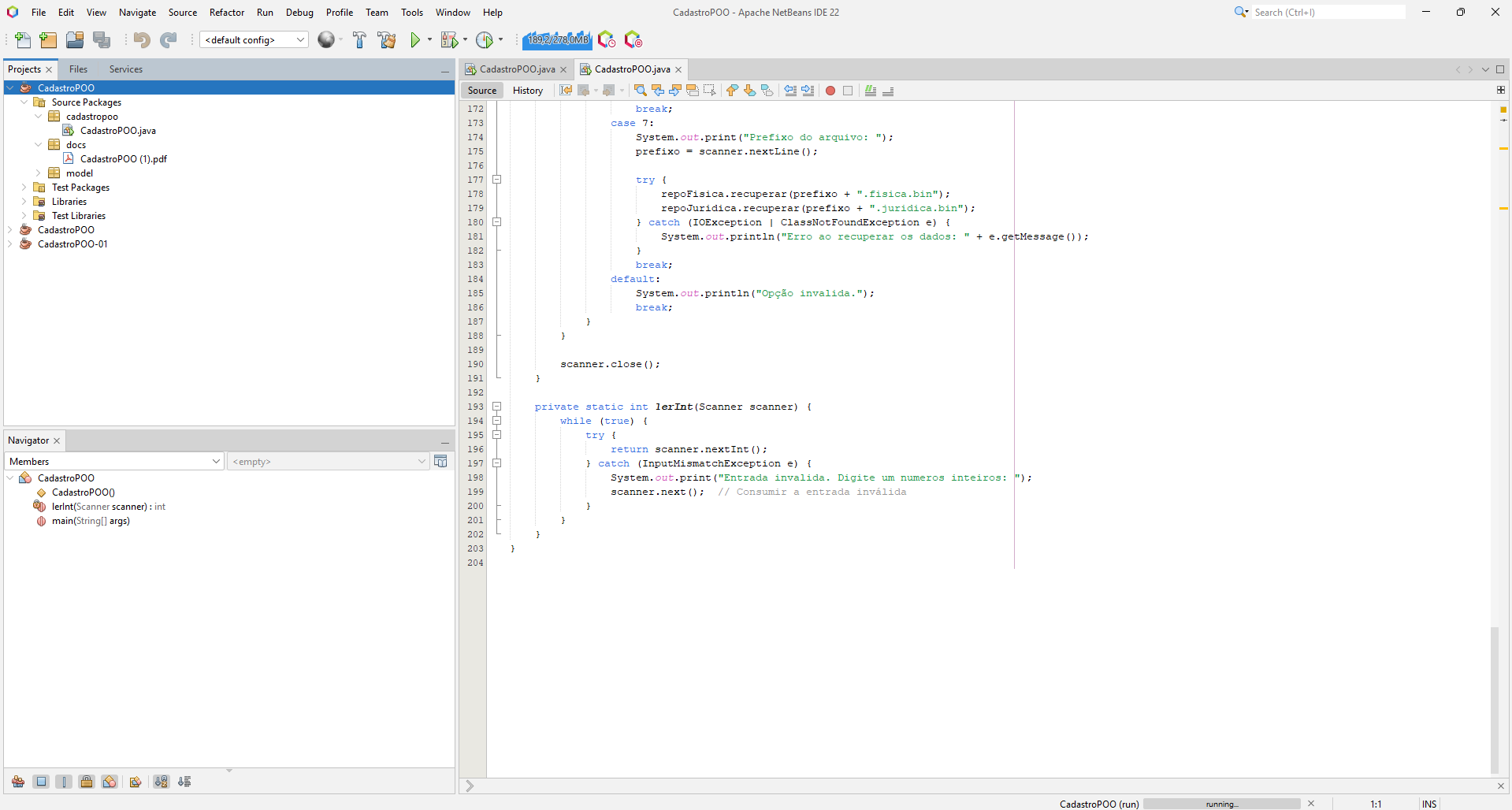
## Código Cadastro POO





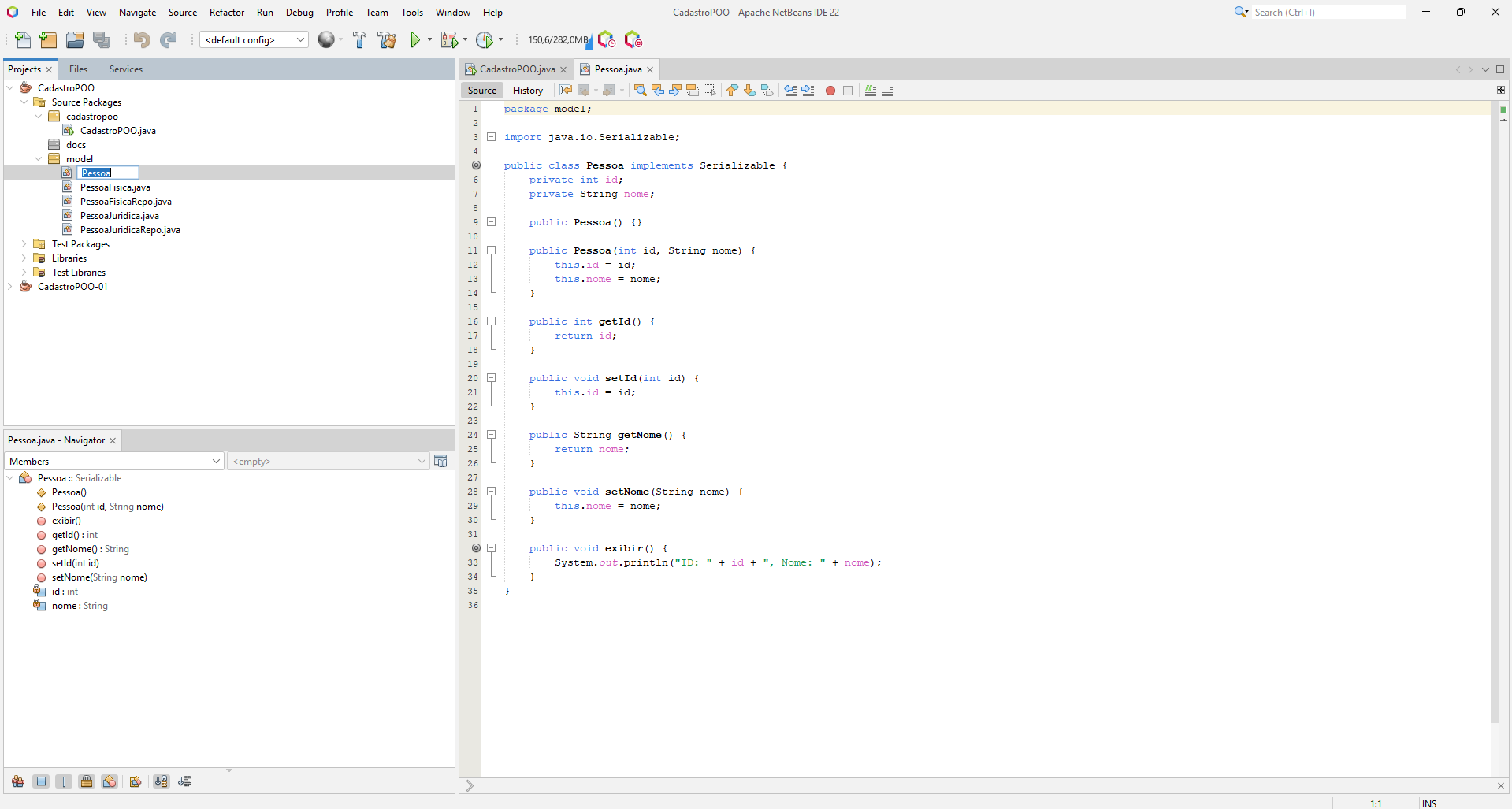




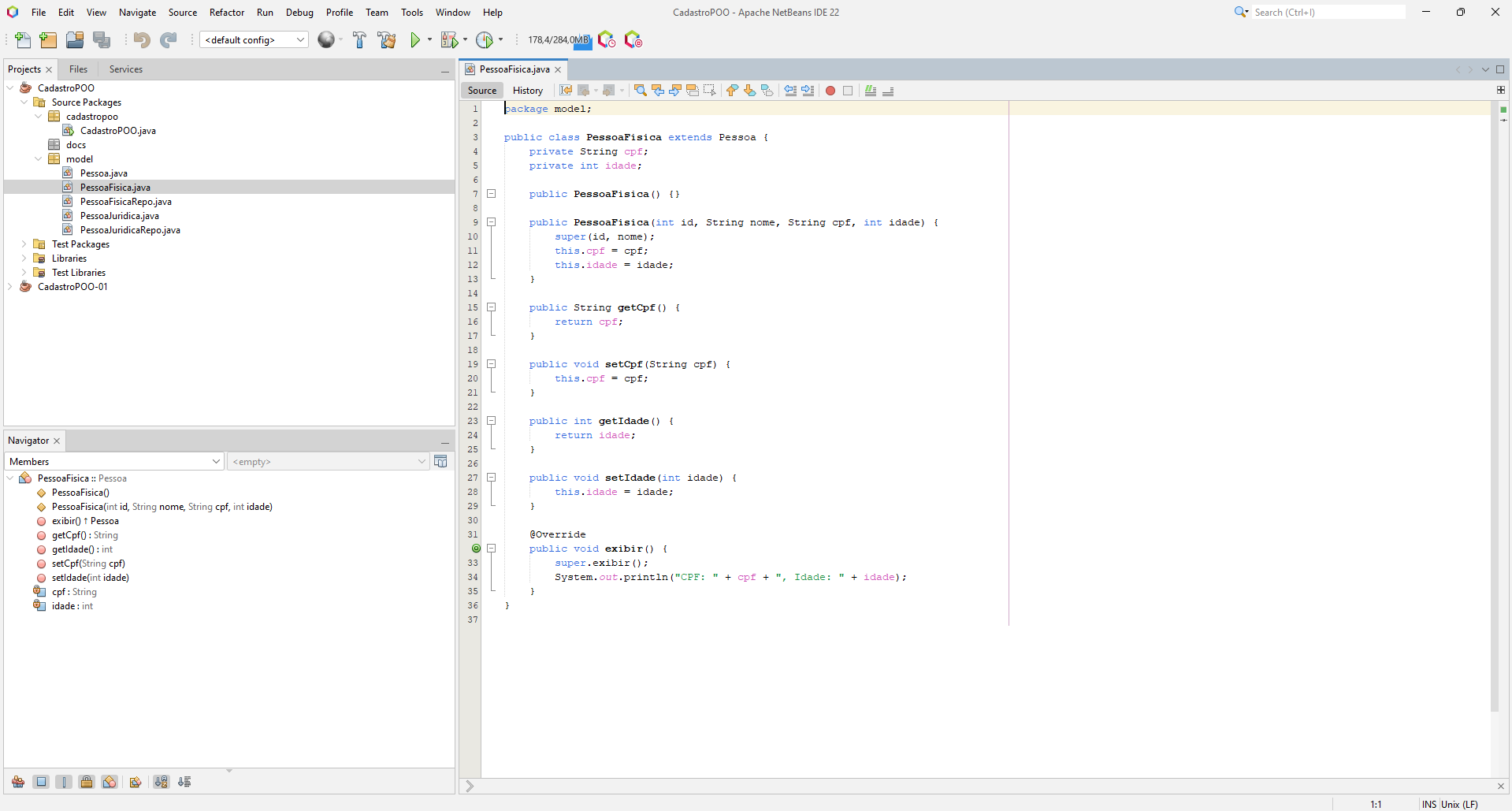


# model

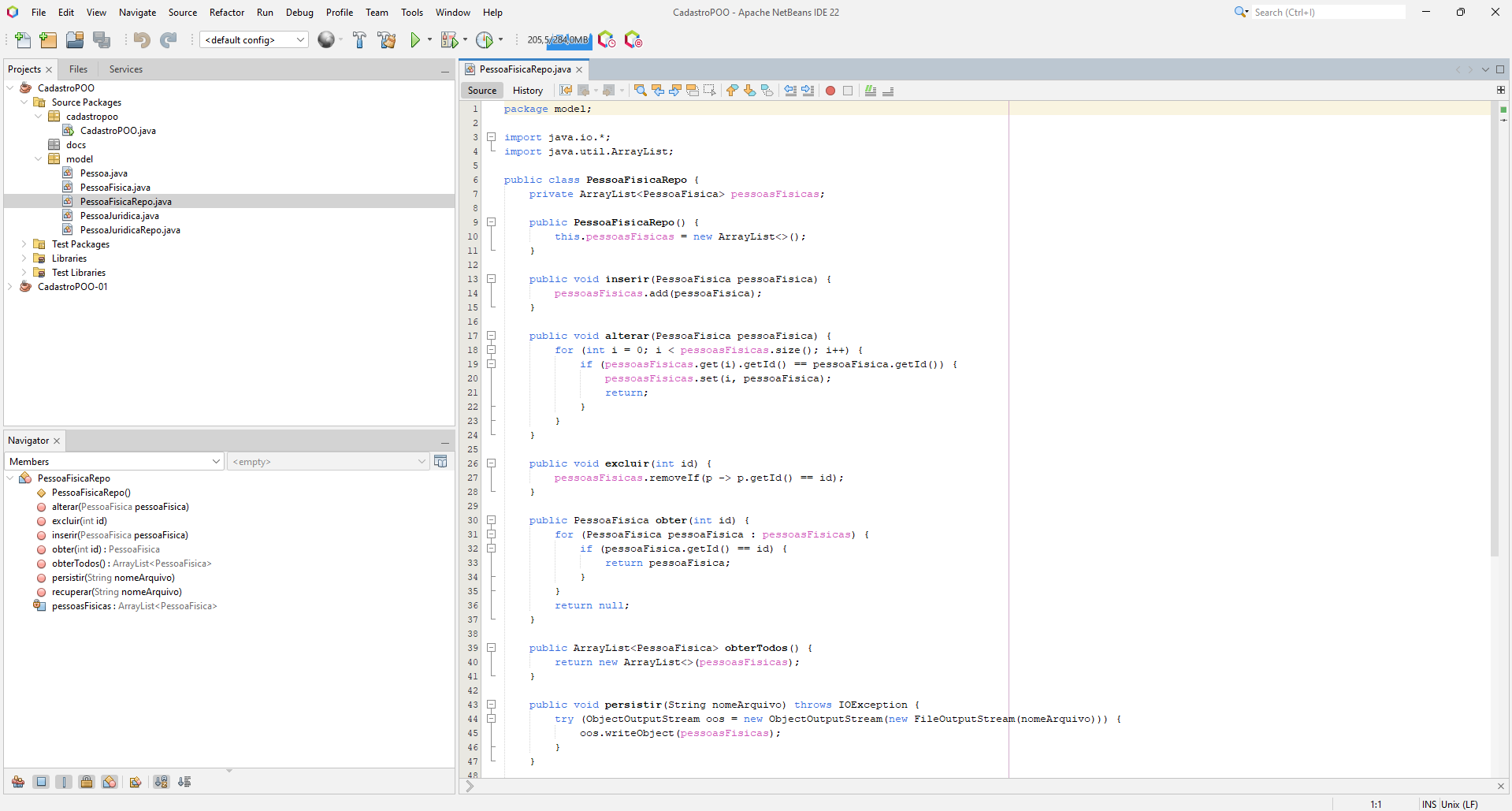
## Pessoa.java

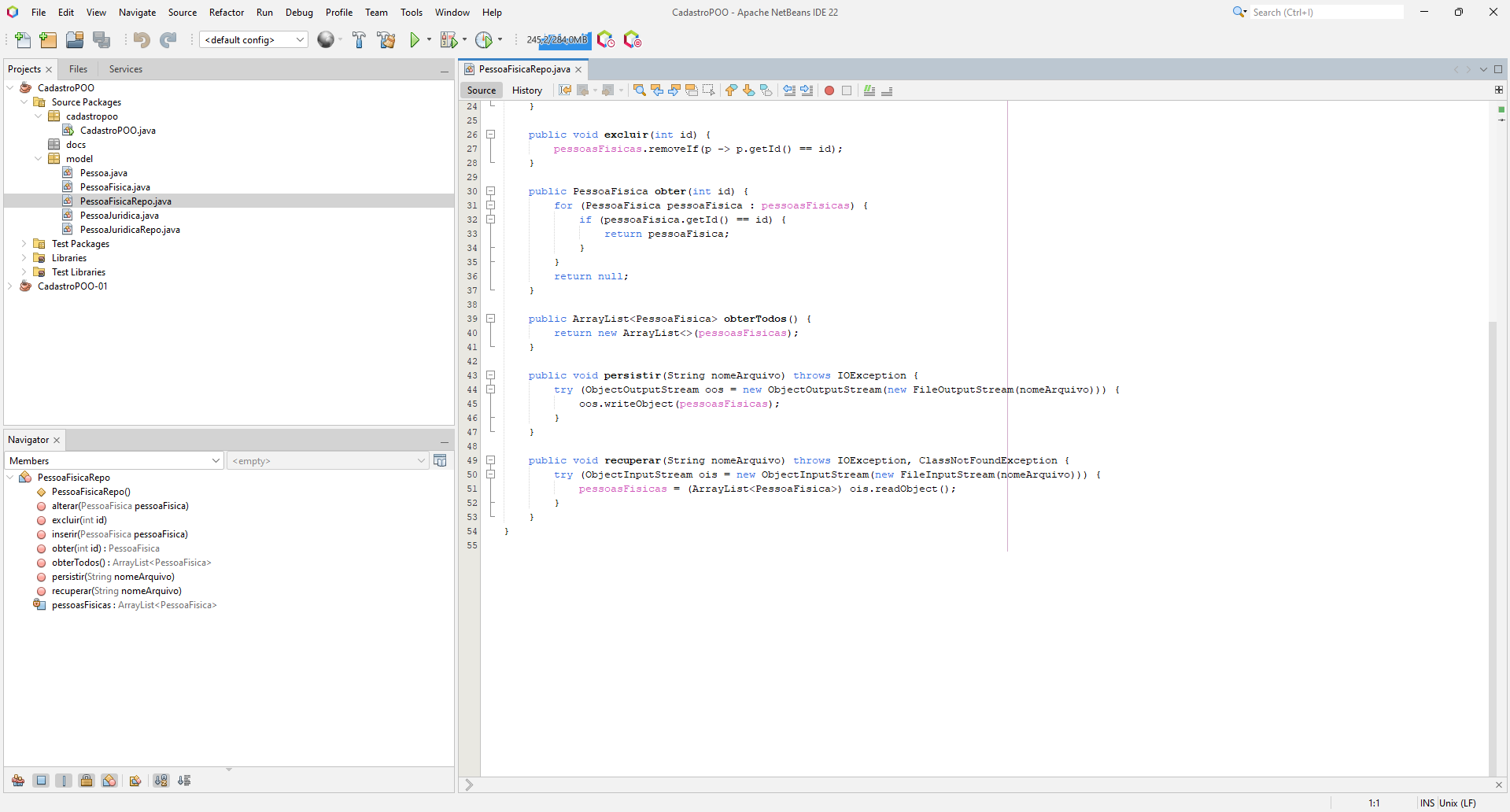


## Pessoa fisica.java

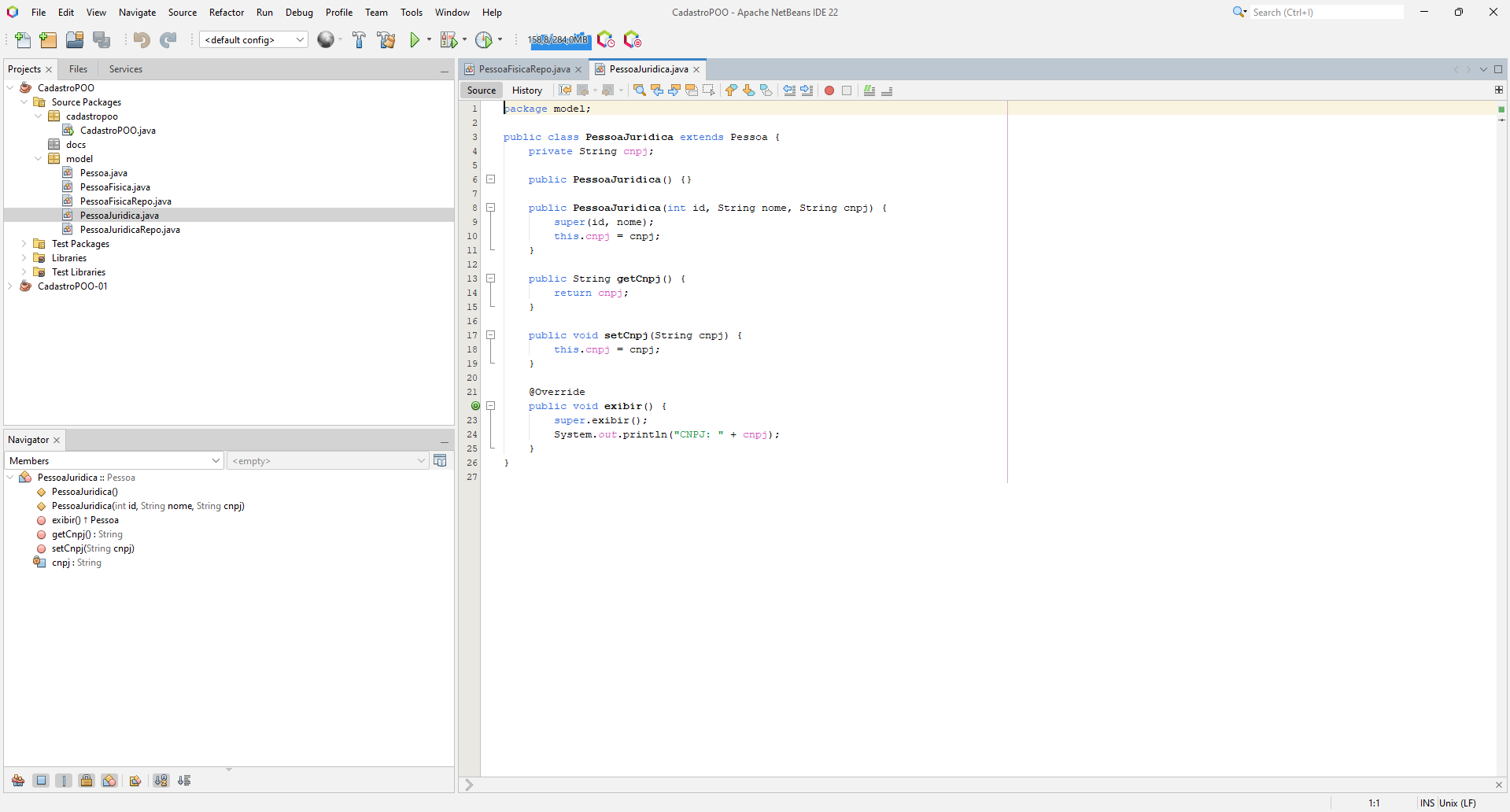


## Pessoa fisica repo.java

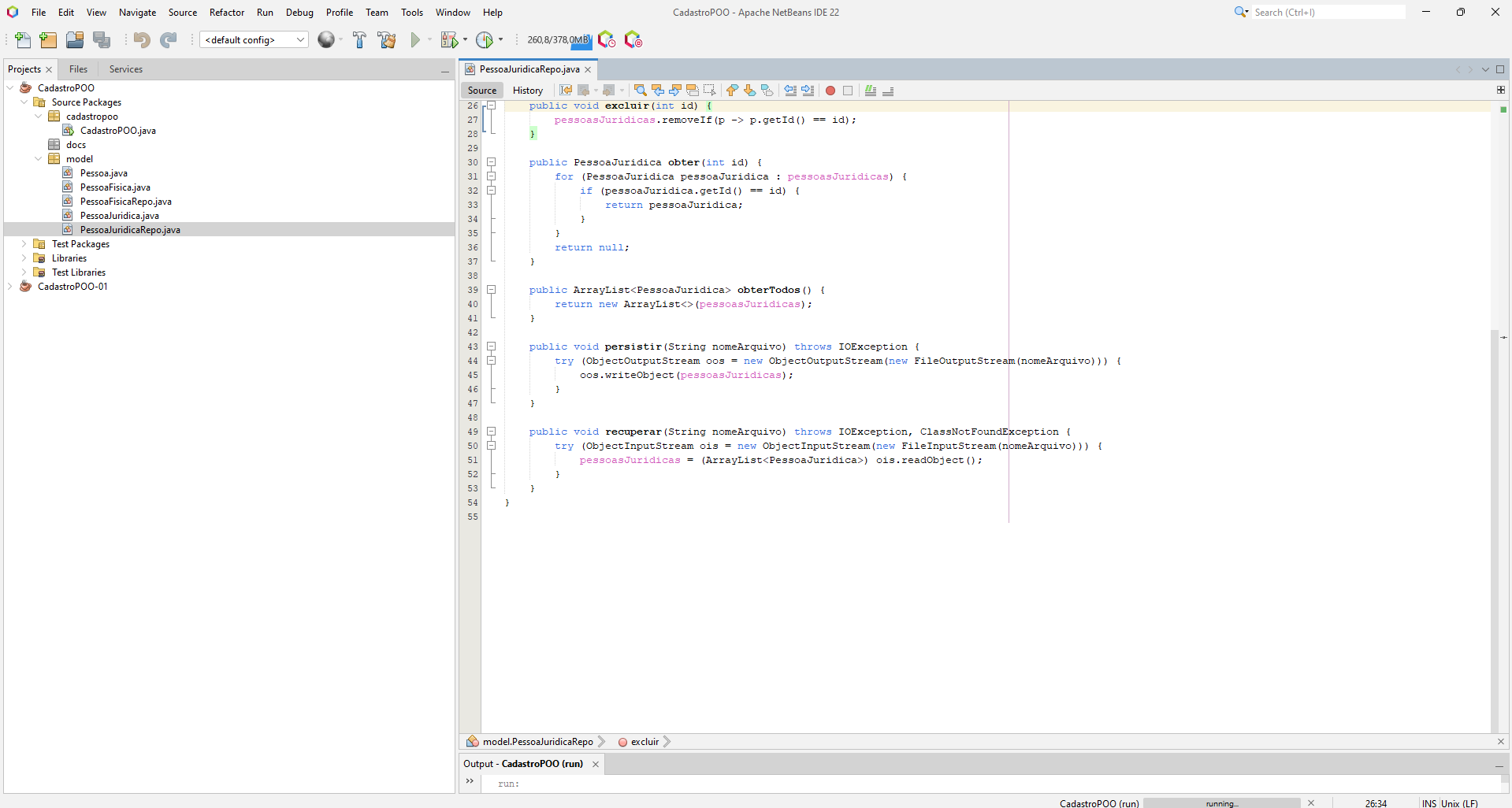
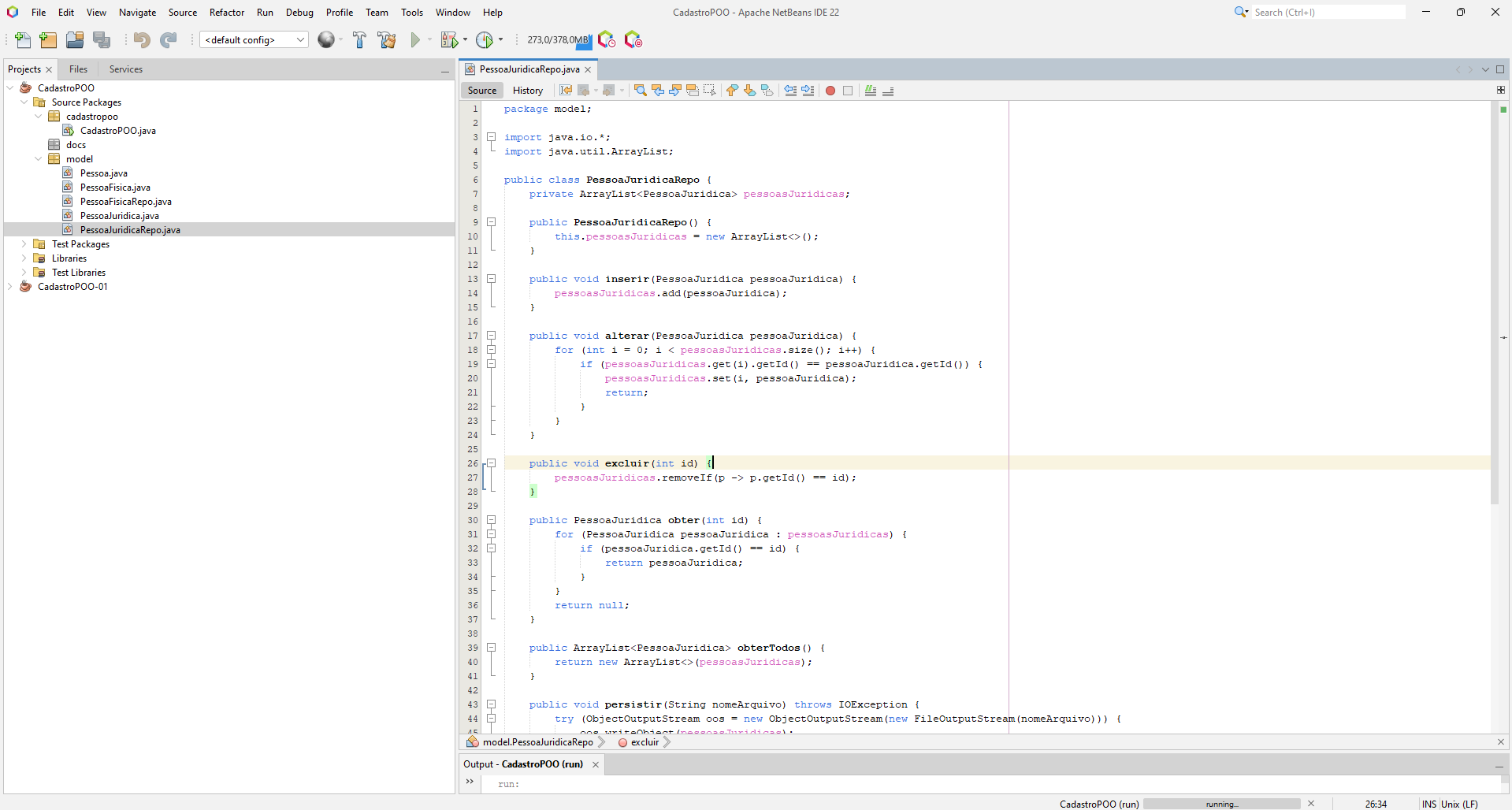




## Pessoa juridica.java

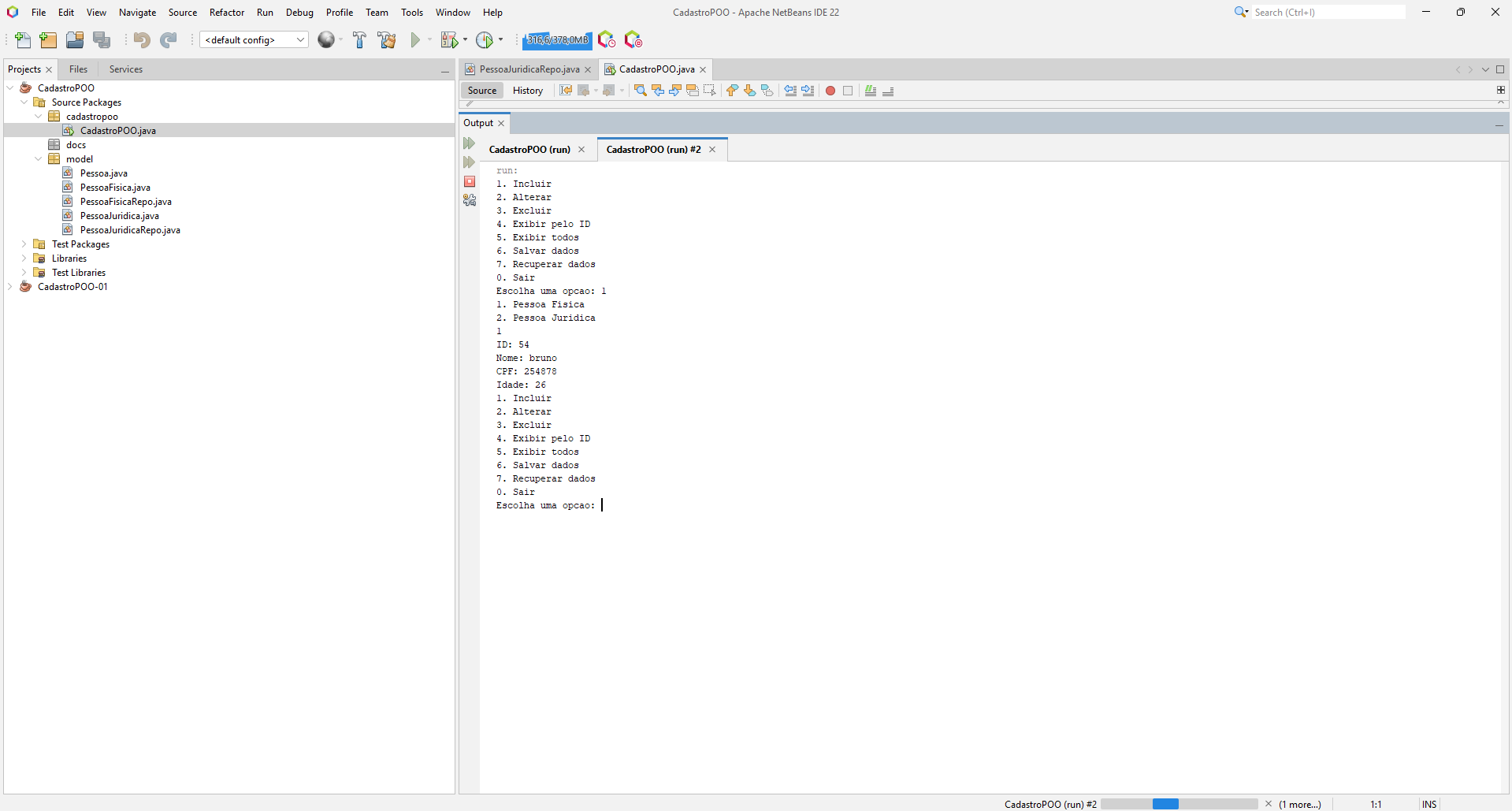


## Pessoa juridica repo.java

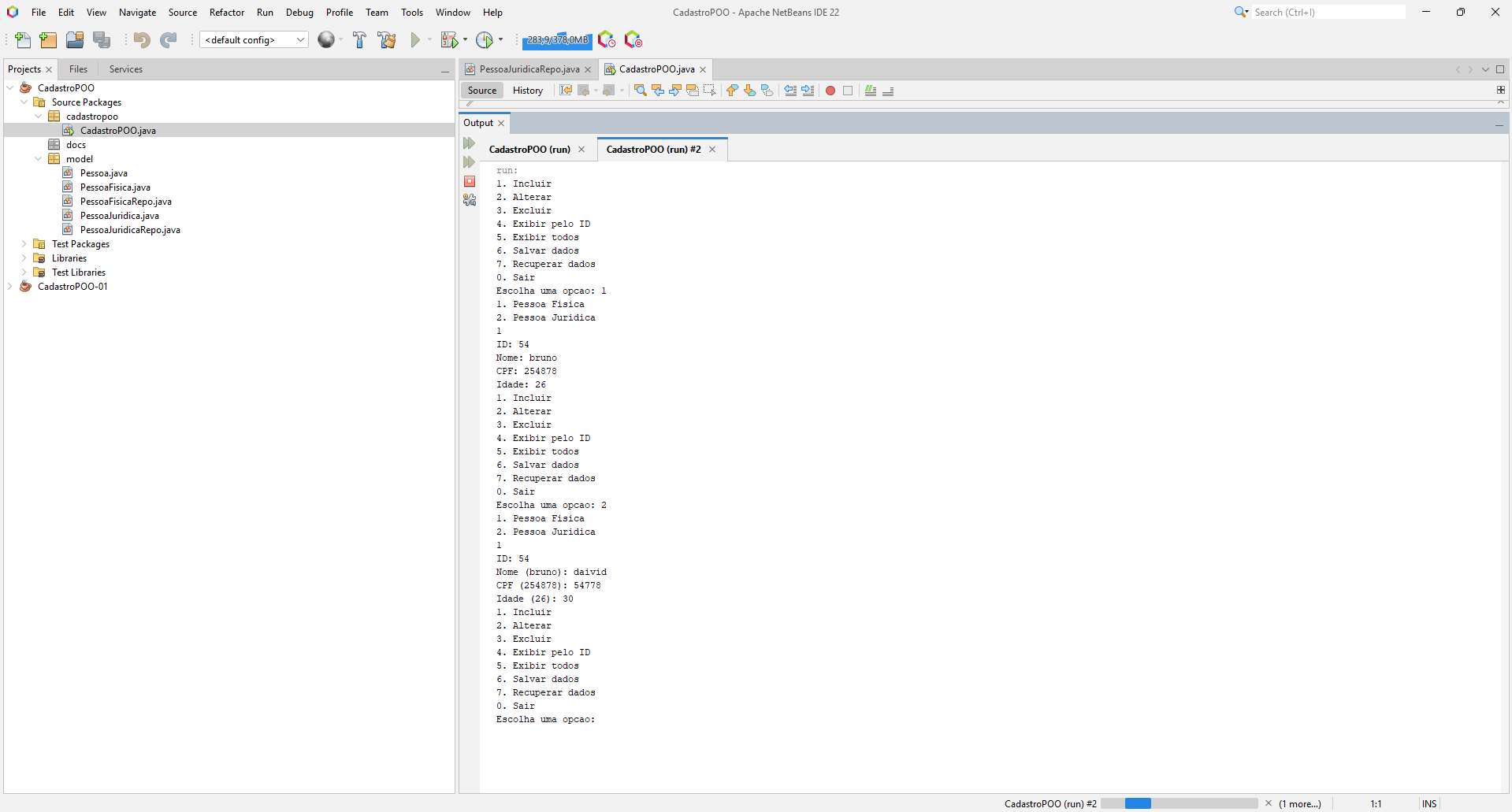


# resultados da execução dos códigos

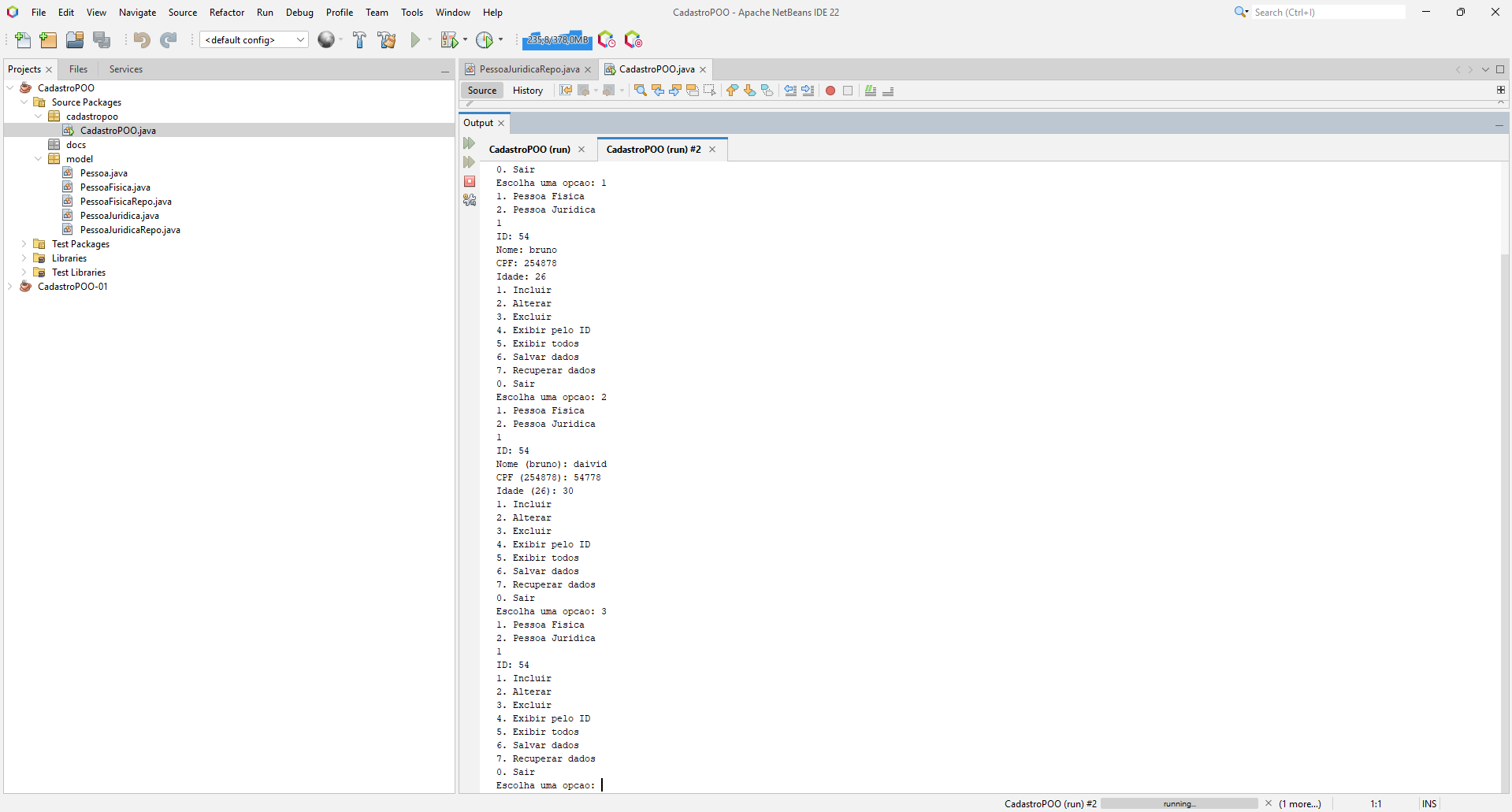
## Incluir



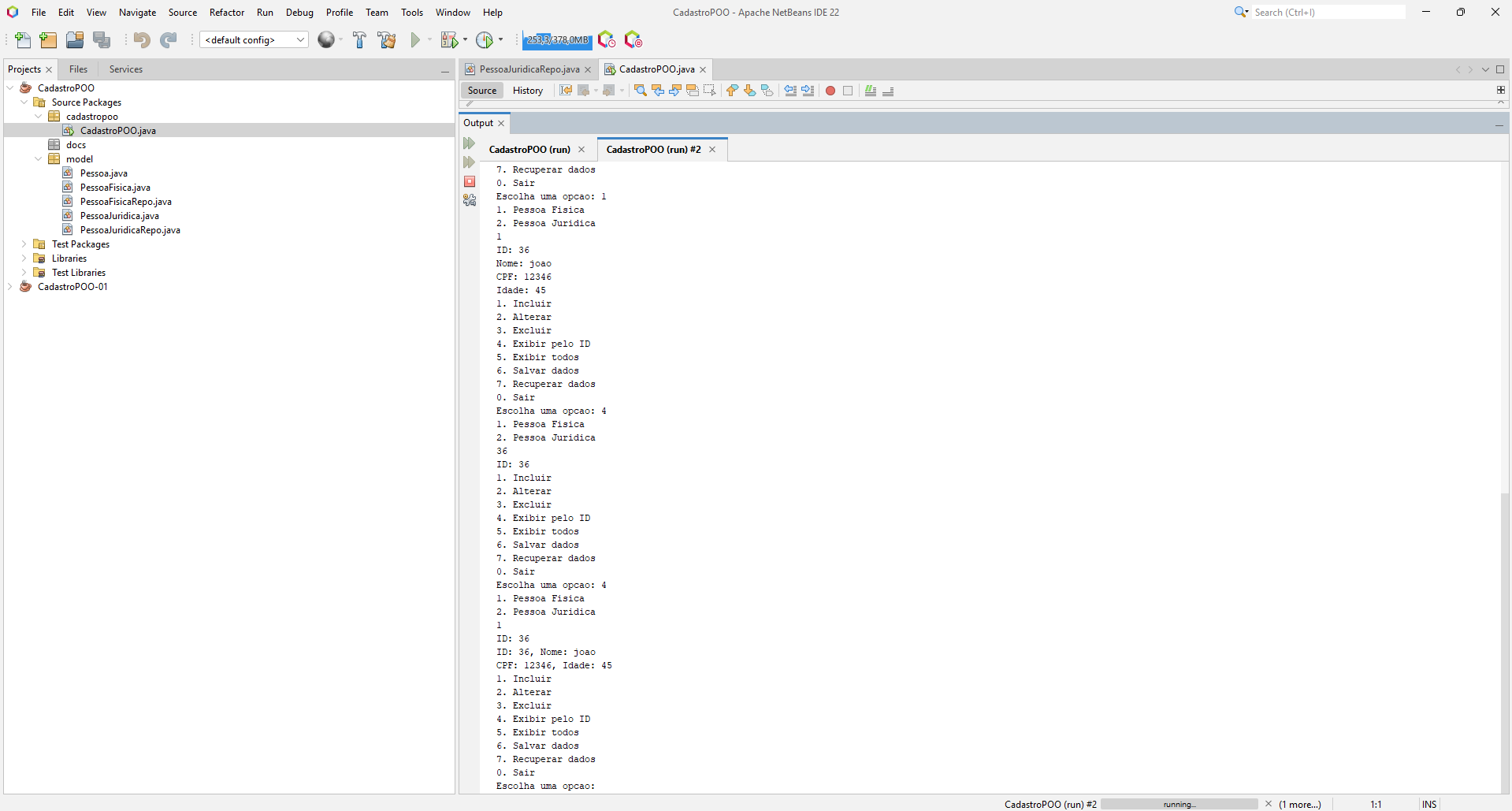
## Alterar



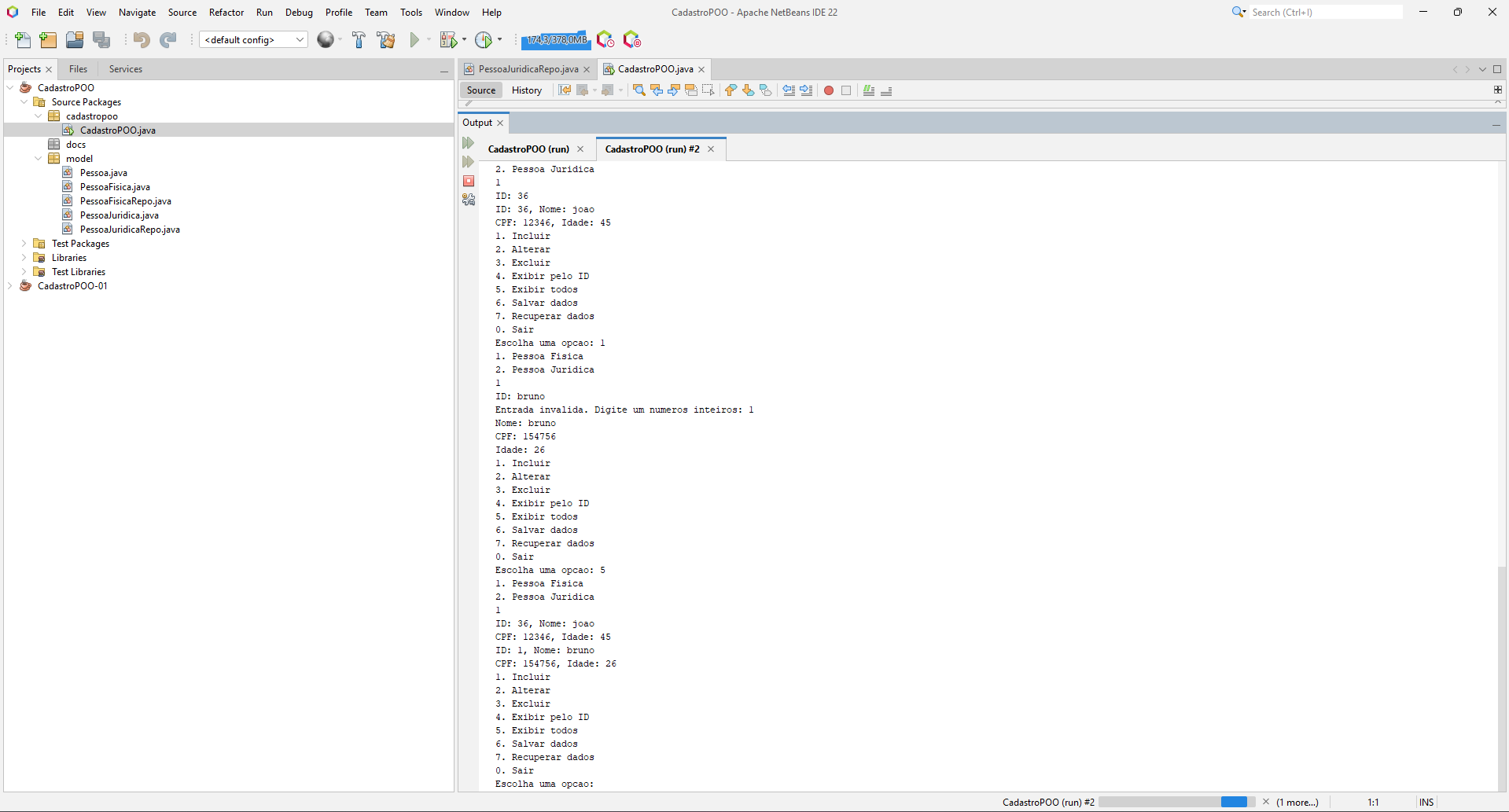
## Excluir



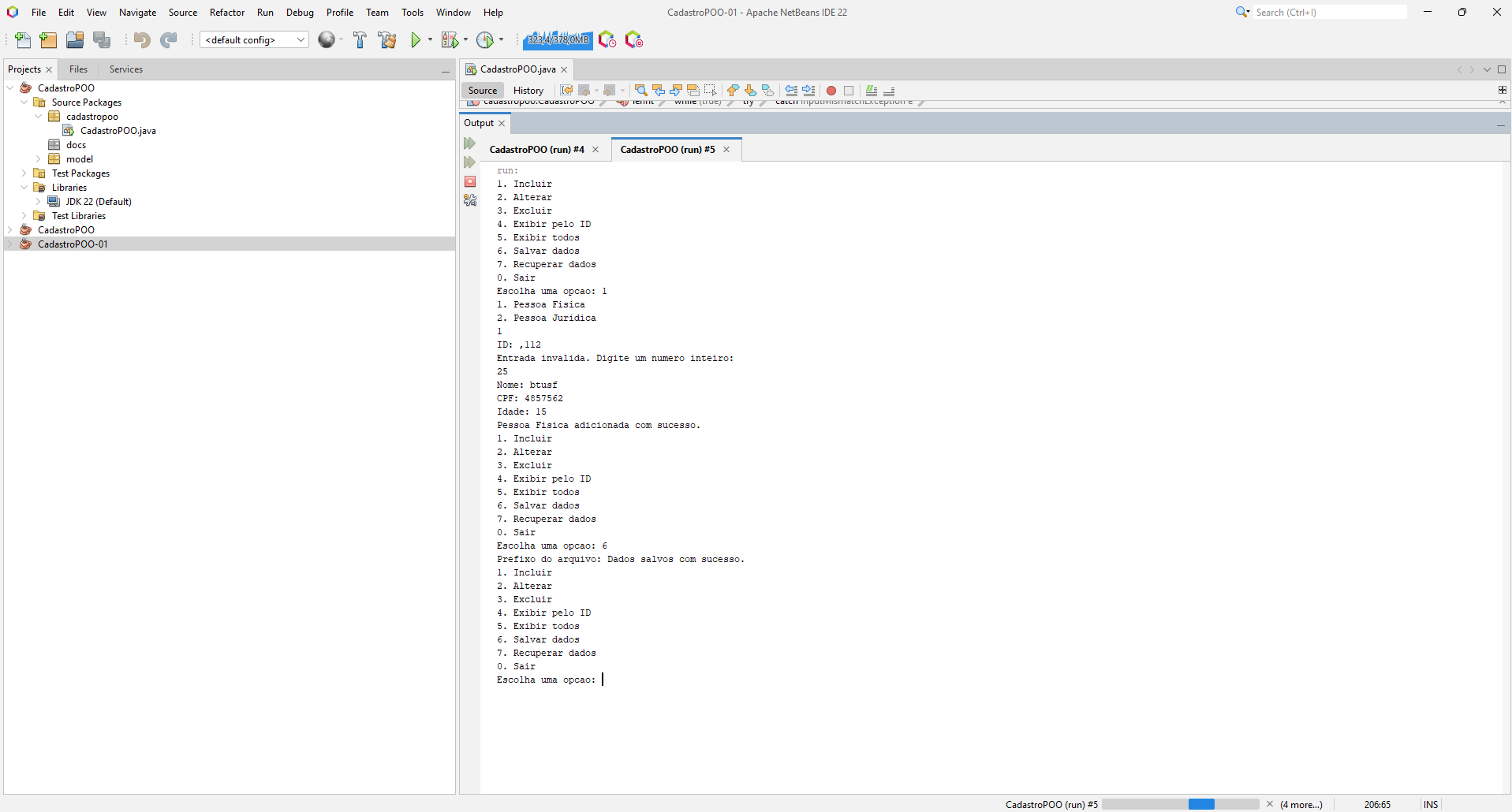
## Exibir pelo id



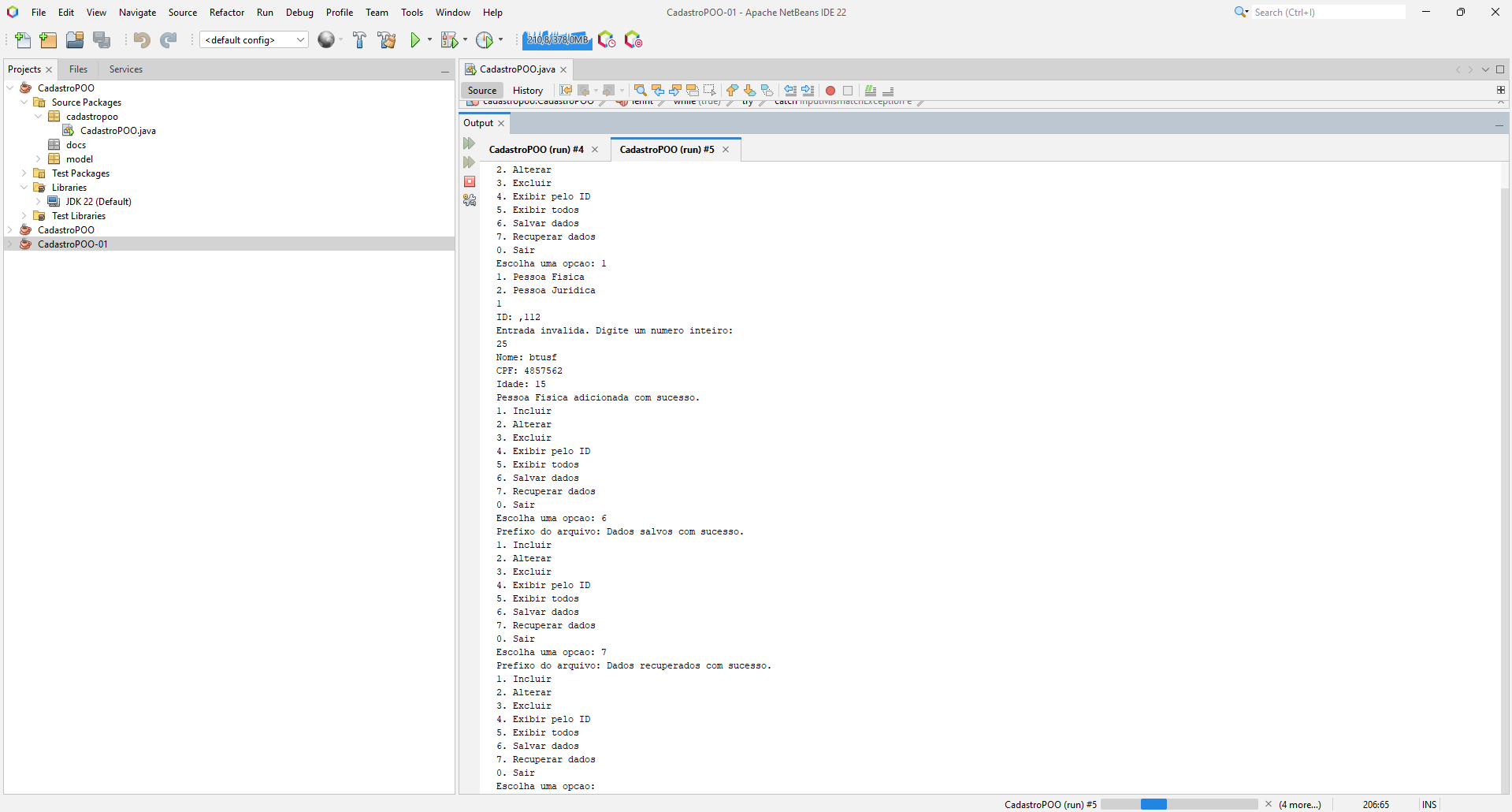
## Exibir todos



## Salvar dados



## Recuperar dados



# ANALISE

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

Elementos estáticos, como o método main, estão associados à classe em vez de suas instâncias, permitindo que a JVM (Java Virtual Machine) invoque o main diretamente para iniciar a execução do programa. O uso do modificador static é essencial para que o método main possa ser executado sem a necessidade de instanciar a classe CadastroPOO. Isso é crucial porque o método main serve como ponto de entrada da aplicação, e a JVM precisa ser capaz de chamá-lo diretamente sem criar um objeto da classe.

## PARA QUE SERVE A CLASSE SCANNER?

A classe Scanner tem um papel fundamental na interação com o usuário, possibilitando a leitura de diferentes tipos de dados (como inteiros, números decimais e strings) a partir da entrada padrão, geralmente o teclado. No contexto do sistema, o Scanner é utilizado para capturar as escolhas do usuário no menu e para coletar os dados necessários durante as operações de inclusão, alteração e exclusão de cadastros. Isso simplifica a entrada e manipulação de dados, contribuindo para uma interface do usuário mais amigável e eficiente.

## COMO O USO DE CLASSES DE REPOSITÓRIO IMPACTOU NA ORGANIZAÇÃO DO CÓDIGO?

O uso das classes de repositório, como PessoaFisicaRepo e PessoaJuridicaRepo, é uma abordagem essencial que melhora significativamente a organização do código. Essas classes são responsáveis pela administração dos dados, oferecendo métodos para adicionar, atualizar, remover e recuperar registros. Ao encapsular a lógica de manipulação de dados, os repositórios seguem o princípio da responsabilidade única, separando claramente a lógica de persistência da interação com o usuário. Isso resulta em um código mais modular, legível e fácil de manter.

Além disso, a divisão das operações de cadastro em métodos específicos na classe CadastroPOO (como adicionar, modificar, excluir, exibirPorId, exibirTodos, salvarDados, recuperarDados) contribui para uma estrutura mais organizada e compreensível do código. Cada método possui uma função bem definida, o que torna o fluxo do programa mais intuitivo e simplifica modificações ou expansões futuras do sistema.

A modularização do código, junto com o uso de repositórios, também facilita a implementação de funcionalidades adicionais, como validação de dados e tratamento de exceções. Por exemplo, ao persistir dados em arquivos binários, os métodos de persistência e recuperação nas classes de repositório garantem a integridade dos dados armazenados, possibilitando a recuperação e utilização desses dados em execuções posteriores do programa.

# Conclusão

O sistema desenvolvido em Java para cadastro de pessoas físicas e jurídicas exemplifica a aplicação eficaz de conceitos de programação orientada a objetos, como modularização e encapsulamento. O uso de elementos estáticos, como o método main, simplifica a inicialização do programa, enquanto a classe Scanner facilita a captura de entradas do usuário, tornando a interação mais intuitiva. As classes de repositório, responsáveis pela administração dos dados, estabelecem uma separação clara entre a lógica de negócios e a interface com o usuário, resultando em um código mais organizado e fácil de manter.

A modularização não apenas melhora a legibilidade, mas também facilita futuras expansões e adaptações do sistema. Essa abordagem permite uma manutenção eficiente e a implementação estruturada de novas funcionalidades. Em resumo, a estrutura do sistema é robusta, escalável e mantém uma clara divisão de responsabilidades, assegurando a integridade e eficiência da aplicação.

# REFERÊNCIAS

**SCIELO**, *As Criptomoedas e os novos desafios aos Sistema monetário: Uma abordagem pós-Keynesiana. https://www.scielo.br/j/ecos/a/twmcnj944hvrsbbsn88jnhd. 2020*.

***ORACLE***,*Java Downloads. Acessado em* [*https://www.oracle.com/java/technologies/downloads/*](https://www.oracle.com/java/technologies/downloads/) *2024.*

***GeeksforGeeks*** , *Java Programming Language. Acessado em.* [*https://www.geeksforgeeks.org/java/*](https://www.geeksforgeeks.org/java/)*. 2024*

**TutorialsPoint**, *Java Tutorial. Acessado em 2024. https://www.tutorialspoint.com/java/index.htm.*

**W3Schools***, Java Tutorial. Acessado em 2024. https://www.w3schools.com/java/.*