



# Estácio

**Campus:** Dorival Caymmi

**Nome:** Filippe Markouizos Duarte -  
202308599615

**Curso:** Desenvolvimento Full-Stack

**Disciplina:** Iniciando o caminho pelo Java

**Turma:** 2023.1

**3º Semestre**

## **Título:**

*Sistema de Cadastro de Pessoas com Persistência em Arquivos Binários*

## **Objetivo:**

*Implementar um sistema de cadastro que usa herança, polimorfismo, repositórios e persistência de dados.*

## **Códigos:**

### **Pessoa.Java:**

```
package model;

import java.io.Serializable;

public class Pessoa implements Serializable {

    private int id;

    private String nome;

    public Pessoa() {}

    public Pessoa(int id, String nome) {

        this.id = id;

        this.nome = nome;

    }

    public int getId() {

        return id;

    }

}
```

```

    public void setId(int id) {

        this.id = id;

    }

    public String getNome() {

        return nome;

    }

    public void setNome(String nome) {

        this.nome = nome;

    }

    public void exibir() {

        System.out.println("ID: " + id + " | Nome: " + nome);

    }

}

```

## PessoaFisica.Java:

```
package model;
```

```

public class PessoaFisica extends Pessoa {

    private String cpf;

    private int idade;

    public PessoaFisica() {

        super();

    }

    public PessoaFisica(int id, String nome, String cpf, int idade)

{

        super(id, nome);

        this.cpf = cpf;

        this.idade = idade;

    }
}

```

```

    public String getCpf() {
        return cpf;
    }

    public void setCpf(String cpf) {
        this.cpf = cpf;
    }

    public int getIdade() {
        return idade;
    }

    public void setIdade(int idade) {
        this.idade = idade;
    }

    @Override
    public void exibir() {
        super.exibir();

        System.out.println("CPF: " + cpf + " | Idade: " + idade);
    }
}

```

## PessoaJuridica.Java:

```

package model;

public class PessoaJuridica extends Pessoa {
    private String cnpj;

    public PessoaJuridica() {
        super();
    }
}

```

```

    }

    public PessoaJuridica(int id, String nome, String cnpj) {

        super(id, nome);

        this.cnpj = cnpj;

    }

    public String getCnpj() {

        return cnpj;

    }

    public void setCnpj(String cnpj) {

        this.cnpj = cnpj;

    }

    @Override

    public void exibir() {

        super.exibir();

        System.out.println("CNPJ: "+ cnpj);

    }

}

```

## PessoaFisicaRepo:

```

package model;

import java.io.*;

import java.util.ArrayList;

import java.util.List;

```

```
public class PessoaFisicaRepo    {

    private List<PessoaFisica> pessoasFisicas = new ArrayList<>();

    public void inserir(PessoaFisica pessoaFisica) {

        pessoasFisicas.add(pessoaFisica);

    }

    public void alterar(PessoaFisica pessoaFisica) {

        for (PessoaFisica pf : pessoasFisicas) {

            if (pf.getId() == pessoaFisica.getId()) {

                pf.setNome(pessoaFisica.getNome());

                pf.setCpf(pessoaFisica.getCpf());

                pf.setIdade(pessoaFisica.getIdade());

                break;

            }

        }

    }

    public void excluir(int id) {

        pessoasFisicas.removeIf(pf -> pf.getId() == id);

    }

    public PessoaFisica obter(int id) {

        for (PessoaFisica pf : pessoasFisicas) {

            if(pf.getId() == id) {

                return pf;

            }

        }

    }

}
```

```

        }

        return null;
    }

    public List<PessoaFisica> obterTodos() {

        return new ArrayList<>(pessoasFisicas);
    }


    public void persistir(String nomeArquivo) throws IOException {

        try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(nomeArquivo))) {

            oos.writeObject(pessoasFisicas);

        }

    }


    @SuppressWarnings("unchecked")

    public void recuperar(String nomeArquivo) throws IOException,
ClassNotFoundException {

        try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(nomeArquivo))) {

            pessoasFisicas = (ArrayList<PessoaFisica>)
ois.readObject();

        }

    }

}

```

## PessoaJuridicaRepo:

```
package model;

import java.io.*;

import java.util.ArrayList;

import java.util.List;

public class PessoaJuridicaRepo {

    private List<PessoaJuridica> pessoasJuridicas = new
ArrayList<>();

    public void inserir(PessoaJuridica pessoaJuridica) {

        pessoasJuridicas.add(pessoaJuridica);

    }

    public void alterar(PessoaJuridica pessoaJuridica) {

        for (PessoaJuridica pj : pessoasJuridicas) {

            if (pj.getId() == pessoaJuridica.getId()) {

                pj.setNome(pessoaJuridica.getNome());

                pj.setCnpj(pessoaJuridica.getCnpj());

                break;

            }

        }

    }

    public void excluir(int id) {

        pessoasJuridicas.removeIf(pj -> pj.getId() == id);

    }

}
```



```
public PessoaJuridica obter(int id) {  
    for (PessoaJuridica pj : pessoasJuridicas) {  
        if (pj.getId() == id) {  
            return pj;  
        }  
    }  
    return null;  
}
```

```
public List<PessoaJuridica> obterTodos() {  
    return new ArrayList<>(pessoasJuridicas);  
}
```

```
public void persistir(String nomeArquivo) throws IOException {  
    try (ObjectOutputStream oos = new ObjectOutputStream(new  
FileOutputStream(nomeArquivo))) {  
        oos.writeObject(pessoasJuridicas);  
    }  
}
```

```
@SuppressWarnings("unchecked")  
public void recuperar(String nomeArquivo) throws IOException,  
ClassNotFoundException {  
    try (ObjectInputStream ois = new ObjectInputStream(new  
FileInputStream(nomeArquivo))) {  
        pessoasJuridicas = (ArrayList<PessoaJuridica>)  
ois.readObject();  
    }  
}
```

```
    }  
    }  
}
```

## Main.Java:

```
package model;  
  
import java.io.IOException;  
import java.util.Scanner;  
  
public class Main {  
    public static void main(String [] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        PessoaFisicaRepo repol = new PessoaFisicaRepo();  
        repol.inserir(new PessoaFisica(1, "João Silva",  
"123.456.789-00", 30));  
        repol.inserir(new PessoaFisica(2, "Maria Oliveira",  
"987.654.321-00", 25));  
  
        try {  
            repol.persistir("pessoas_fisicas.bin");  
            System.out.println("Dados de pessoas físicas salvos  
com sucesso.");  
        } catch (IOException e) {  
            System.out.println("Erro ao salvar dados de pessoas  
físicas: " + e.getMessage());  
        }  
    }  
}
```

```
PessoaFisicaRepo repo2 = new PessoaFisicaRepo();

try {

    repo2.recuperar("pessoas_fisicas.bin");

    System.out.println("Dados de pessoas físicas
recuperados com sucesso.");

} catch (IOException | ClassNotFoundException e) {

    System.out.println("Erro ao recuperar dados de
pessoas físicas: " + e.getMessage());

}

System.out.println("Pessoas Físicas Recuperadas:");

for (PessoaFisica pf : repo2.obterTodos()) {

    pf.exibir();

}

PessoaJuridicaRepo repo3 = new PessoaJuridicaRepo();

repo3.inserir(new PessoaJuridica(1, "Empresa X",
"12.345.678/0001-99"));

repo3.inserir(new PessoaJuridica(2, "Empresa Y",
"98.765.432/0001-55"));

try {

    repo3.persistir("pessoas_juridicas.bin");

    System.out.println("Dados de pessoas jurídicas
salvos com sucesso.");

} catch (IOException e) {

    System.out.println("Erro ao salvar dados de pessoas
jurídicas: " + e.getMessage());

}
```

```
PessoaJuridicaRepo repo4 = new PessoaJuridicaRepo();

try {

    repo4.recuperar("pessoas_juridicas.bin");

    System.out.println("Dados de pessoas jurídicas
recuperados com sucesso.");

} catch (IOException | ClassNotFoundException e) {

    System.out.println("Erro ao recuperar dados de
pessoas jurídicas: " + e.getMessage());

}

System.out.println("Pessoas Jurídicas Recuperadas:");

for (PessoaJuridica pj : repo4.obterTodos()) {

    pj.exibir();

}

boolean running = true;

while(running) {

    System.out.println("\n --- Menu ---");

    System.out.println("1. Incluir");

    System.out.println("2. Alterar");

    System.out.println("3. Excluir");

    System.out.println("4. Exibir");

    System.out.println("5. Exibir Todos");

    System.out.println("6. Salvar Dados");

    System.out.println("7. Recuperar Dados");
```

```
System.out.println("0. Sair");

int opcao = scanner.nextInt();

scanner.nextLine();

switch (opcao) {

    case 1:

        System.out.println("1. Pessoa Jurídica");

        System.out.println("2. Pessoa Física");

        System.out.println("Escolha o tipo: ");

        int tipo = scanner.nextInt();

        scanner.nextLine();

        if (tipo == 1) {

            incluirPessoaJuridica(scanner, repo3);

        }

        else if (tipo == 2) {

            incluirPessoaFisica(scanner, repo1);

        }

        break;

    case 2:

        System.out.println("1. Pessoa Jurídica");

        System.out.println("2. Pessoa Física");

        System.out.println("Escolha o tipo: ");
```

```
        tipo = scanner.nextInt();

        scanner.nextLine();

        if (tipo == 1) {

            alterarPessoaJuridica(scanner, repo3);

        }

        else if (tipo == 2) {

            alterarPessoaFisica(scanner, repo1);

        }

        break;

case 3:

    System.out.println("1. Pessoa Jurídica");

    System.out.println("2. Pessoa Física");

    System.out.println("Escolha o tipo: ");

    tipo = scanner.nextInt();

    scanner.nextLine();

    if (tipo == 1) {

        excluirPessoaJuridica(scanner, repo3);

    }

    else if (tipo == 2) {

        excluirPessoaFisica(scanner, repo1);

    }

    break;
```

case 4:

```
System.out.println("1. Pessoa Jurídica");  
System.out.println("2. Pessoa Física");  
System.out.println("Escolha o tipo: ");
```

```
tipo = scanner.nextInt();  
scanner.nextLine();
```

```
if (tipo == 1) {  
    exibirPessoaJuridica(scanner, repo3);  
}
```

```
else if (tipo == 2) {  
    exibirPessoaFisica(scanner, repo1);  
}  
break;
```

case 5:

```
System.out.println("1. Pessoa Jurídica");  
System.out.println("2. Pessoa Física");  
System.out.println("Escolha o tipo: ");
```

```
tipo = scanner.nextInt();  
scanner.nextLine();
```

```
if (tipo == 1) {  
    exibirTodosJuridico(repo3);  
}
```

```

        else if(tipo == 2) {
            exibirTodosFisico(repo1);
        }
        break;
    case 6:
        System.out.println("Digite o prefixo do
arquivo: ");

        String prefixo = scanner.nextLine();
        salvarDados(prefixo, repo3, repo1);
        break;
    case 7:
        System.out.println("Digite o prefixo do
arquivo: ");

        prefixo = scanner.nextLine();
        recuperarDados(prefixo, repo4, repo2);
        break;
    case 0:
        running = false;
        break;
    default:
        System.out.println("Opção Inválida");
    }
}

scanner.close();
}

private static void incluirPessoaFisica(Scanner scanner,
PessoaFisicaRepo repoFisica) {

```



```

        System.out.println("Digite o ID: ");

        int id = scanner.nextInt();

        scanner.nextLine();

        System.out.println("Digite o nome: ");

        String nome = scanner.nextLine();

        System.out.println("Digite o CPF: ");

        String cpf = scanner.nextLine();

        System.out.println("Digite a idade: ");

        int idade = scanner.nextInt();


        PessoaFisica pessoaFisica = new PessoaFisica(id, nome,
cpf, idade);

        repoFisica.inserir(pessoaFisica);

        System.out.println("Pessoa física inserida com
sucesso!");

    }

    private static void incluirPessoaJuridica(Scanner scanner,
PessoaJuridicaRepo repoJuridica) {

        System.out.println("Digite o ID: ");

        int id = scanner.nextInt();

        scanner.nextLine();

        System.out.println("Digite o nome: ");

        String nome = scanner.nextLine();

        System.out.println("Digite o CNPJ: ");

        String cnpj = scanner.nextLine();


        PessoaJuridica pessoaJuridica = new PessoaJuridica(id,
nome, cnpj);

```

```
        repoJuridica.inserir(pessoaJuridica);

        System.out.println();

    }
}
```

```
    private static void alterarPessoaFisica(Scanner scanner,
PessoaFisicaRepo repoFisica) {

        System.out.println("Digite o ID da pessoa Fisica: ");

        int id = scanner.nextInt();

        scanner.nextLine();

        PessoaFisica pessoaFisica = repoFisica.obter(id);

        if (pessoaFisica != null) {

            System.out.println("Dados atuais: ");

            pessoaFisica.exibir();

            System.out.println("Digite o novo nome: ");

            String nome = scanner.nextLine();

            System.out.println("Digite o novo CPF: ");

            String cpf = scanner.nextLine();

            System.out.println("Digite a nova idade: ");

            int idade = scanner.nextInt();

            scanner.nextLine();

            pessoaFisica.setNome(nome);

            pessoaFisica.setCpf(cpf);

            pessoaFisica.setIdade(idade);

            repoFisica.alterar(pessoaFisica);

        }

    }
}
```

```
        System.out.println("Pessoa Física alterada com  
sucesso!");  
  
    } else {  
  
        System.out.println("Pessoa Física não encontrada.");  
  
    }  
  
}
```

```
    private static void alterarPessoaJuridica(Scanner scanner,  
PessoaJuridicaRepo repoJuridica) {  
  
        System.out.println("Digite o ID da pessoa Jurídica: ");  
  
        int id = scanner.nextInt();  
  
        scanner.nextLine();  
  
        PessoaJuridica pessoaJuridica = repoJuridica.obter(id);  
  
        if (pessoaJuridica != null) {  
  
            System.out.println("Dados atuais: ");  
  
            pessoaJuridica.exibir();  
  
  
  
            System.out.println("Digite o novo nome: ");  
  
            String nome = scanner.nextLine();  
  
            System.out.println("Digite o novo CNPJ: ");  
  
            String cnpj = scanner.nextLine();  
  
  
  
  
            pessoaJuridica.setNome(nome);  
  
            pessoaJuridica.setCnpj(cnpj);  
  
            repoJuridica.alterar(pessoaJuridica);  
  
            System.out.println("Pessoa Jurídica alterada com  
sucesso!");  
  
        } else {
```

```
                System.out.println("Pessoa Jurídica não  
encontrada.");  
            }  
        }  
    }
```

```
        private static void excluirPessoaFisica(Scanner scanner,  
PessoaFisicaRepo repoFisica) {  
  
            System.out.println("Digite o ID da pessoa física a ser  
excluída: ");  
  
            int id = scanner.nextInt();  
  
            scanner.nextLine();  
  
            repoFisica.excluir(id);  
  
            System.out.println("Pessoa Física excluída com  
sucesso.");  
        }  
    }
```

```
        private static void excluirPessoaJuridica(Scanner scanner,  
PessoaJuridicaRepo repoJuridica) {  
  
            System.out.println("Digite o ID da pessoa jurídica a ser  
excluída: ");  
  
            int id = scanner.nextInt();  
  
            scanner.nextLine();  
  
            repoJuridica.excluir(id);  
  
            System.out.println("Pessoa Jurídica excluída com  
sucesso.");  
        }  
    }
```

```
        private static void exibirPessoaFisica(Scanner scanner,  
PessoaFisicaRepo repoFisica) {  
  
            System.out.println("Digite o ID da pessoa física: ");  
        }  
    }
```

```

        int id = scanner.nextInt();

        scanner.nextLine();

        PessoaFisica pessoaFisica = repoFisica.obter(id);

        if (pessoaFisica != null){

            pessoaFisica.exibir();

        } else {

            System.out.println("Pessoa Física não encontrada.");

        }

    }

```

```

        private static void exibirPessoaJuridica(Scanner scanner,
        PessoaJuridicaRepo repoJuridica) {

            System.out.println("Digite o ID da pessoa jurídica: ");

            int id = scanner.nextInt();

            scanner.nextLine();

            PessoaJuridica pessoaJuridica = repoJuridica.obter(id);

            if (pessoaJuridica != null) {

                pessoaJuridica.exibir();

            } else {

                System.out.println("Pessoa Jurídica não
encontrada.");

            }

        }

```

```

        private static void exibirTodosFisico(PessoaFisicaRepo
repoFisica) {

            for (PessoaFisica pessoaFisica :
repoFisica.obterTodos()) {

```

```

        pessoaFisica.exibir();
    }
}

private static void exibirTodosJuridico(PessoaJuridicaRepo
repoJuridica) {
    for (PessoaJuridica pessoaJuridica :
repoJuridica.obterTodos()) {
        pessoaJuridica.exibir();
    }
}

private static void salvarDados(String prefixo, PessoaJuridicaRepo
repoJuridica, PessoaFisicaRepo repoFisica) {
    try {
        repoFisica.persistir(prefixo + ".fisica.bin");
        repoJuridica.persistir(prefixo + ".juridica.bin");
        System.out.println("Dados salvos com sucesso.");
    } catch (IOException e) {
        System.out.println("Erro ao salvar os dados: " +
e.getMessage());
    }
}

private static void recuperarDados(String prefixo,
PessoaJuridicaRepo repoJuridica, PessoaFisicaRepo repoFisica) {
    try {
        repoFisica.recuperar(prefixo + ".fisica.bin");
        repoJuridica.recuperar(prefixo + ".juridica.bin");
    }
}

```

```

        System.out.println("Dados recuperados com
sucesso.");

    } catch (IOException | ClassNotFoundException e) {

        System.out.println("Erro ao recuperar os dados: " +
e.getMessage());

    }

}

}

```

## Resultados:

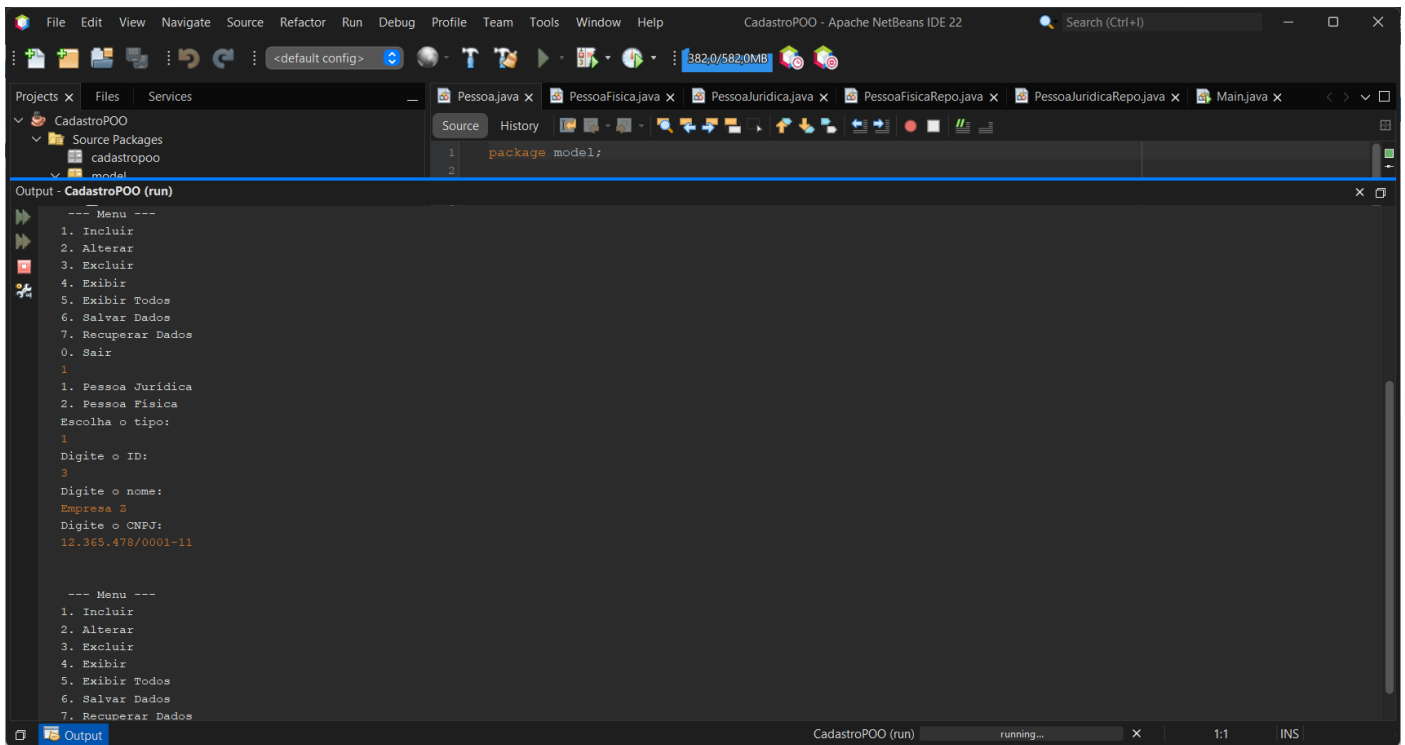
```

run:
Dados de pessoas fisicas salvos com sucesso.
Dados de pessoas fisicas recuperados com sucesso.
Pessoas Fisicas Recuperadas:
ID: 1 | Nome: João Silva
CPF: 123.456.789-00 | Idade: 30
ID: 2 | Nome: Maria Oliveira
CPF: 987.654.321-00 | Idade: 25
Dados de pessoas juridicas salvos com sucesso.
Dados de pessoas juridicas recuperados com sucesso.
Pessoas Juridicas Recuperadas:
ID: 1 | Nome: Empresa X
CNPJ: 12.345.678/0001-99
ID: 2 | Nome: Empresa Y
CNPJ: 98.765.432/0001-55

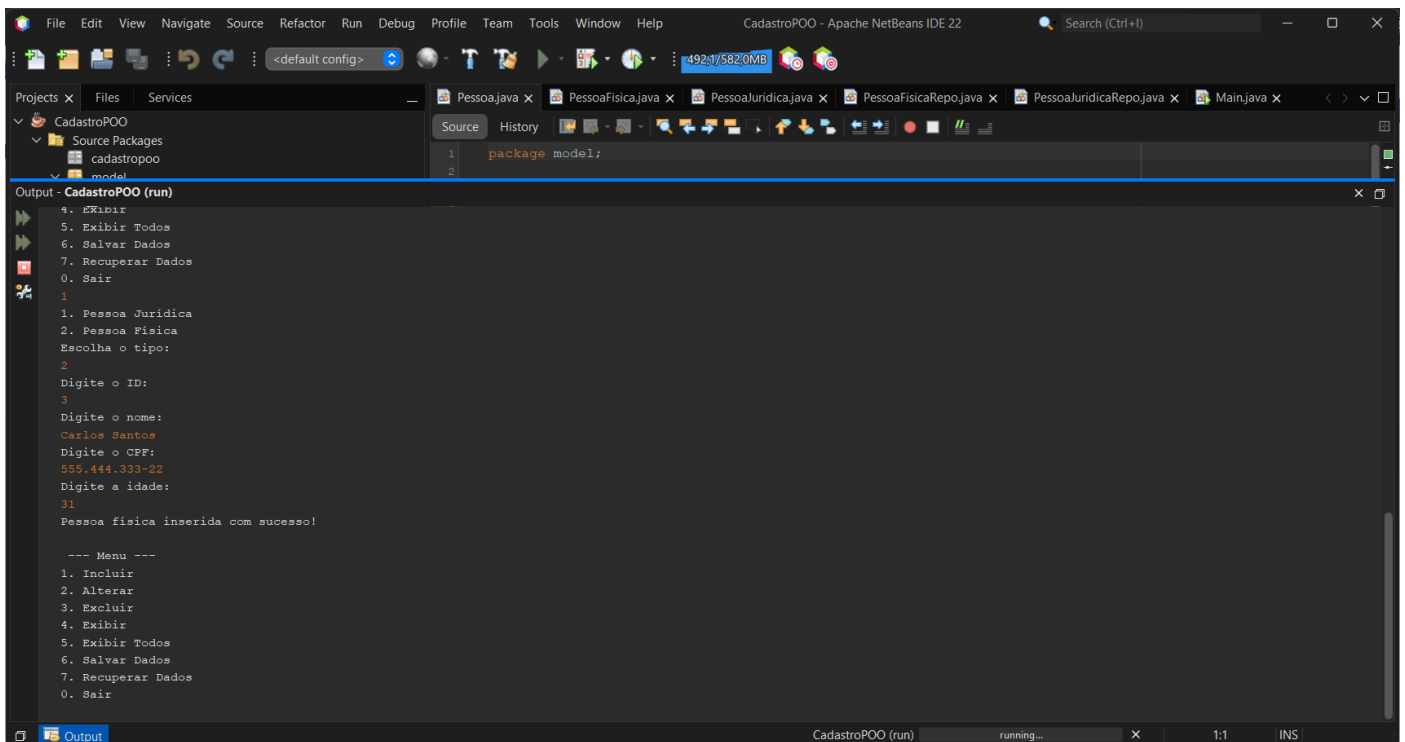
--- Menu ---
1. Incluir
2. Alterar
3. Excluir
4. Exibir
5. Exibir Todos
6. Salvar Dados
7. Recuperar Dados
0. Sair

```

**Primeira Imagem:** Aplicativo rodando no início

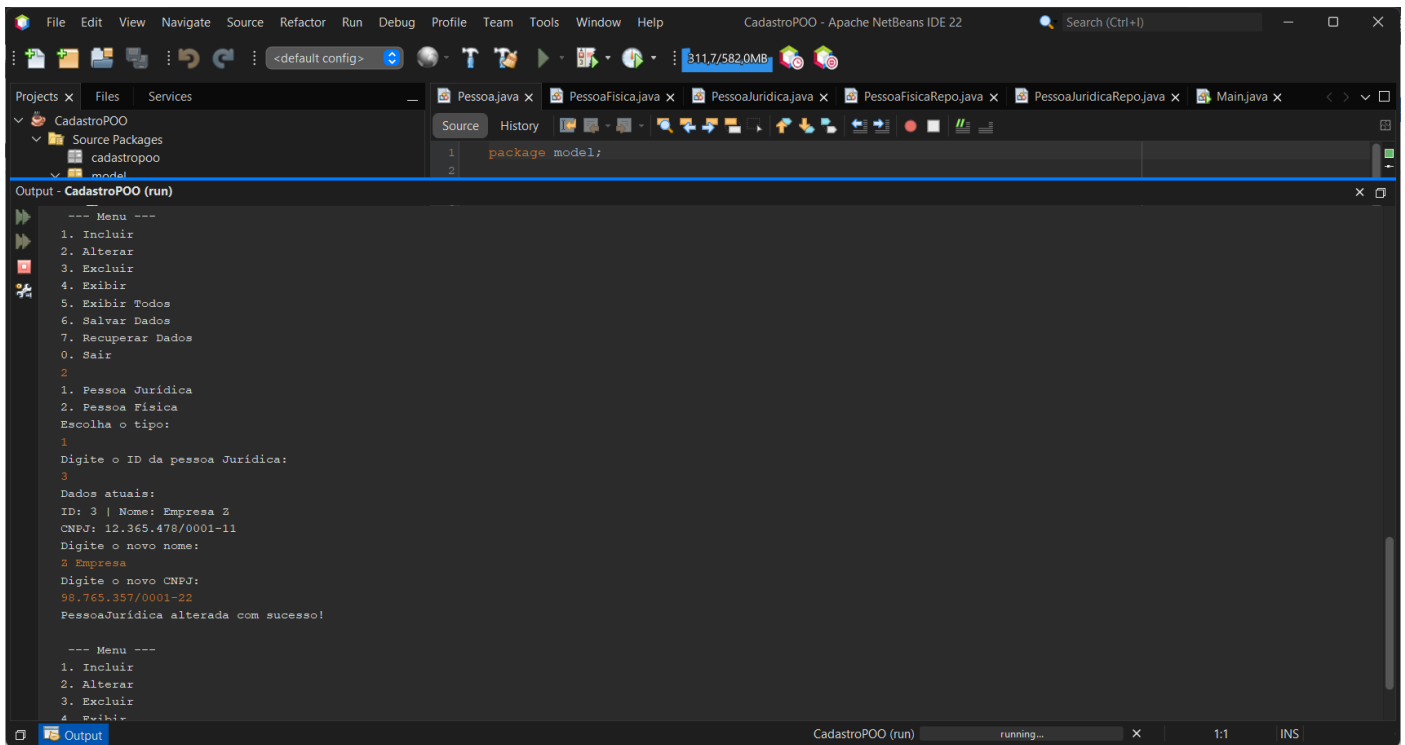


**Segunda Imagem:** Incluir pessoa jurídica funcionando.

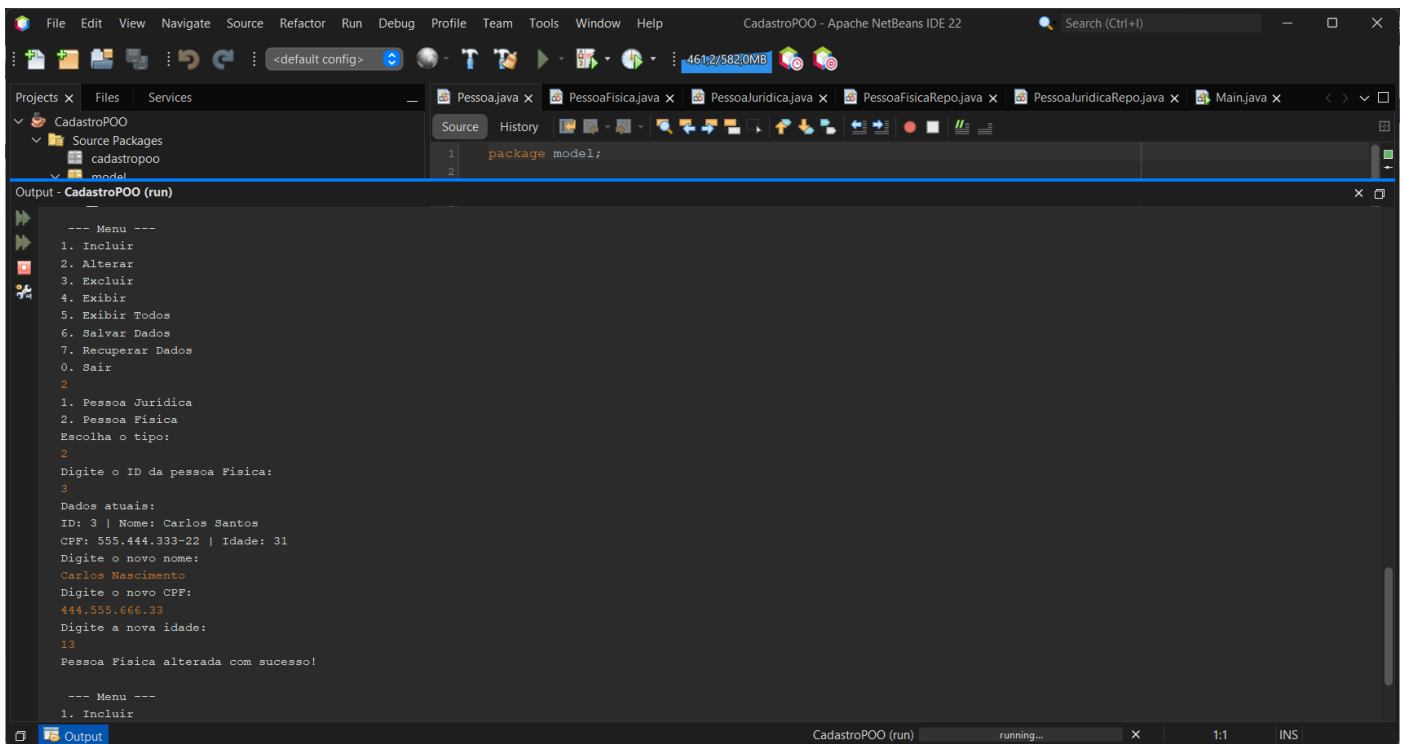


**Terceira Imagem:** Incluir pessoa física funcionando.

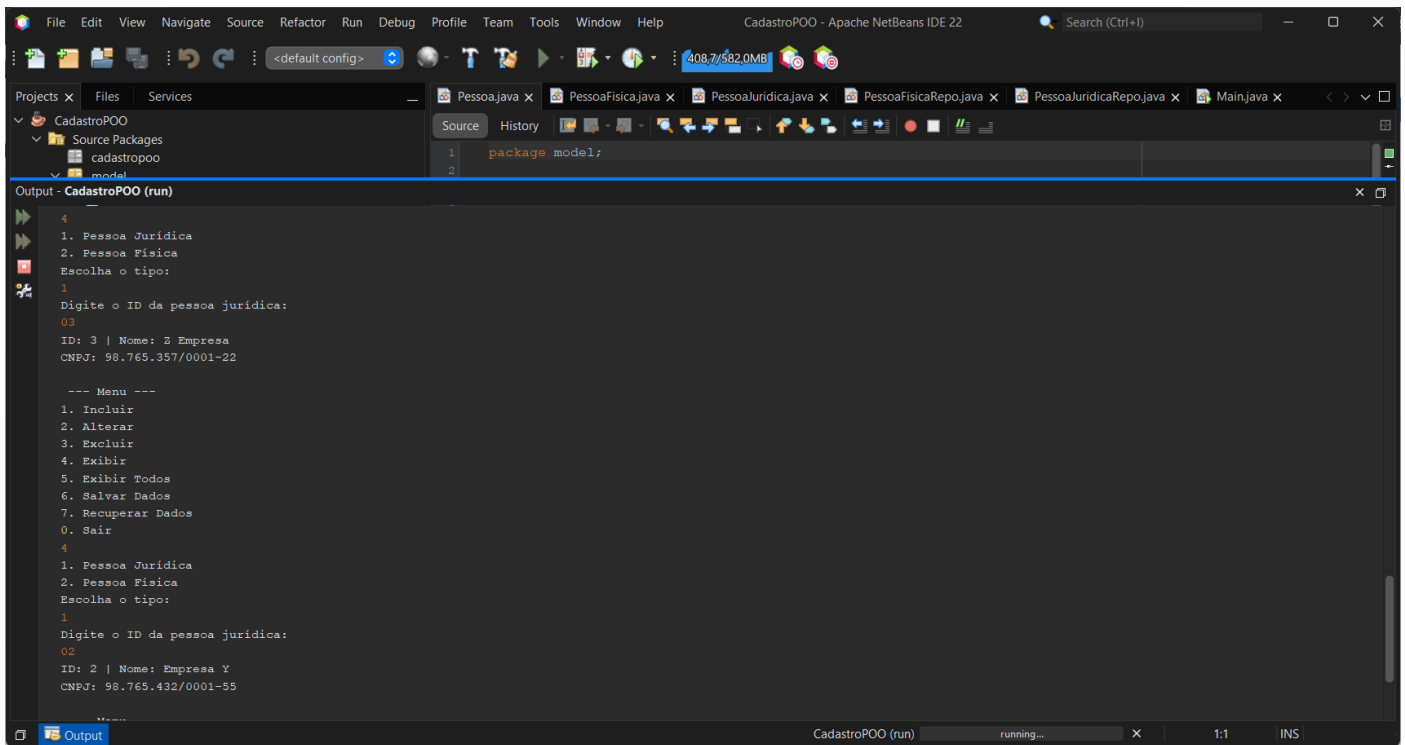




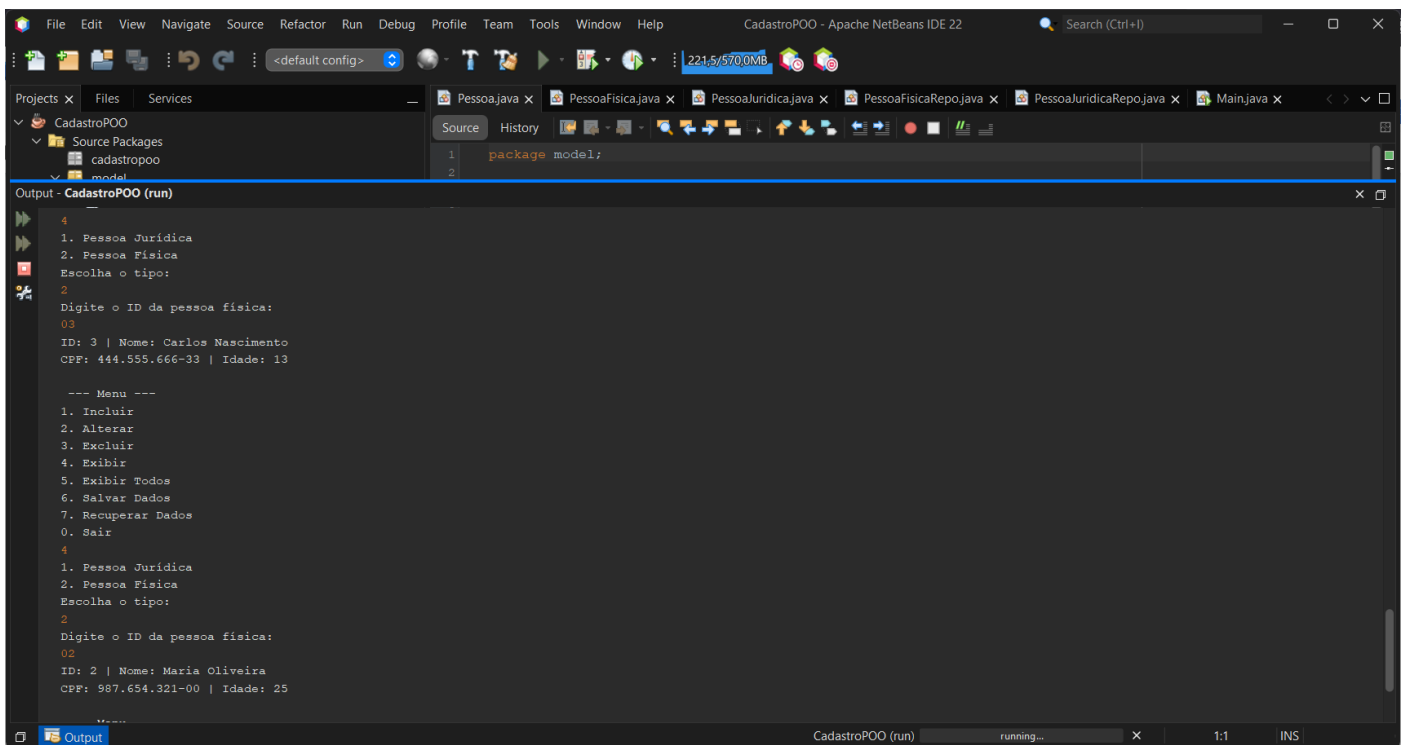
**Quarta Imagem:** Alterar pessoa jurídica funcionando.



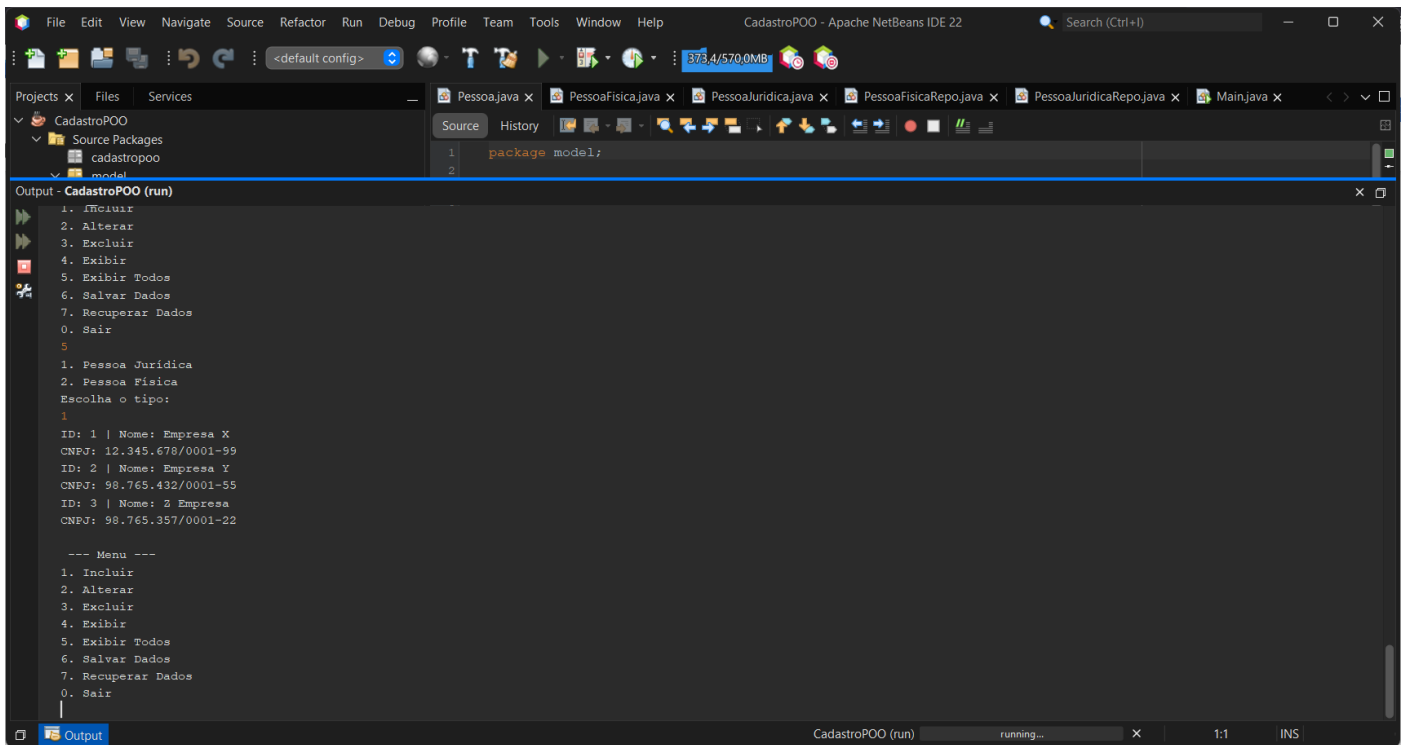
**Quinta Imagem:** Alterar pessoa física funcionando.



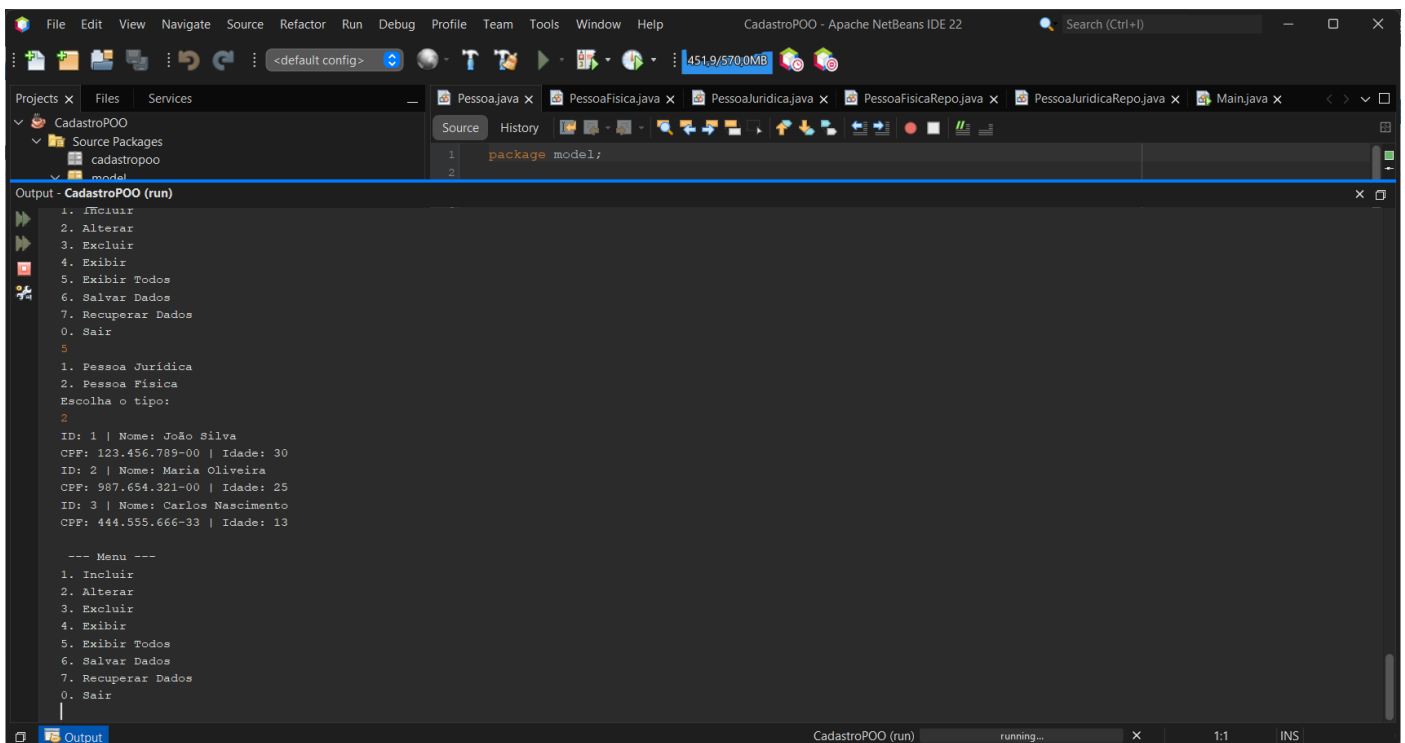
**Sexta Imagem:** Exibir pessoa jurídica por ID funcionando.



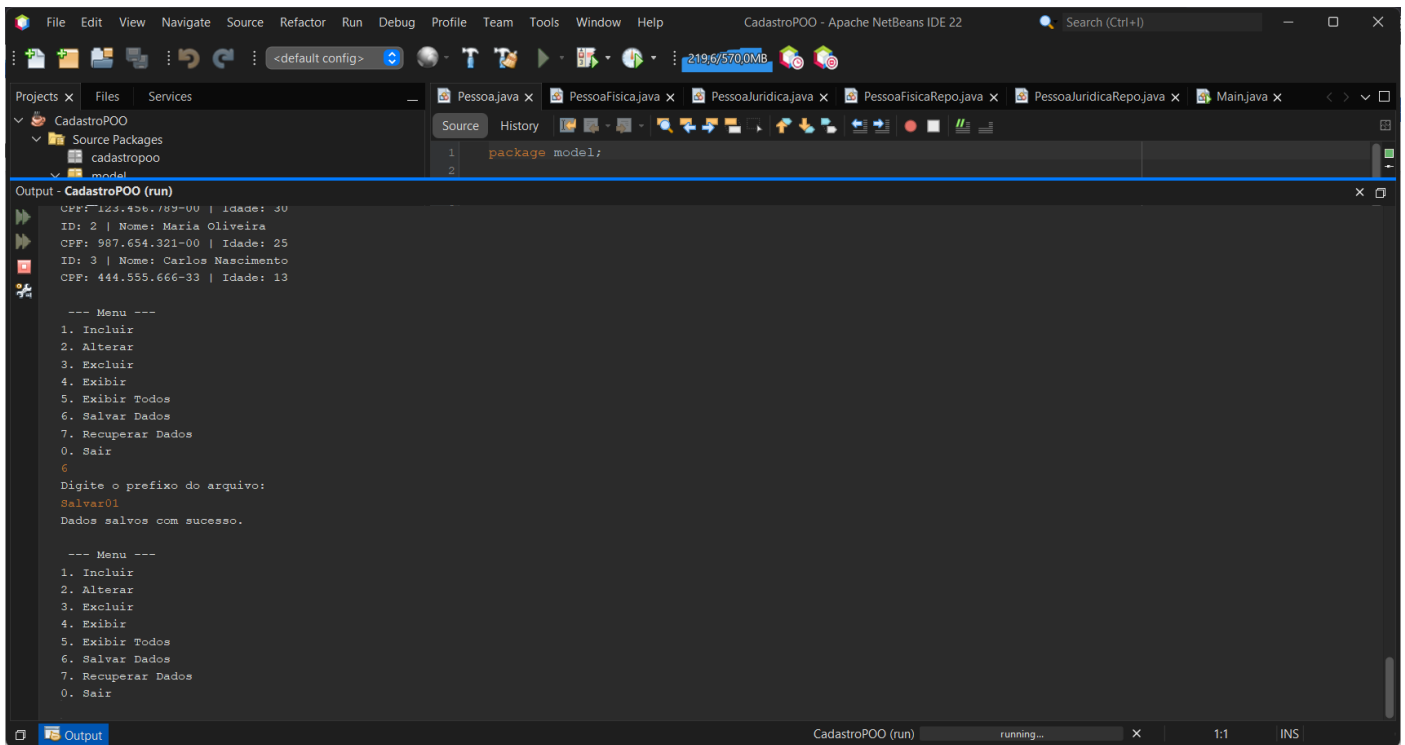
**Sétima Imagem:** Exibir pessoa física por ID funcionando.



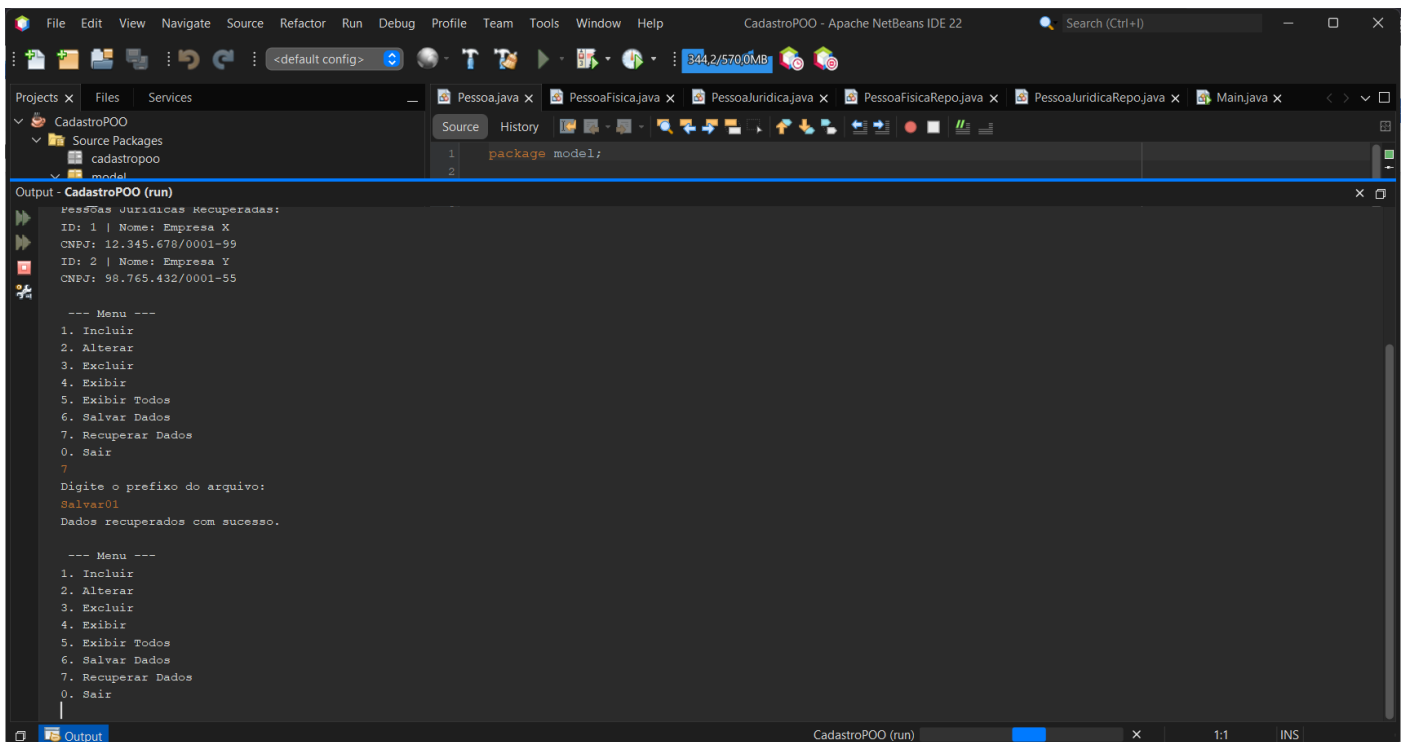
**Oitava Imagem: Exibir todos (Jurídico) funcionando.**



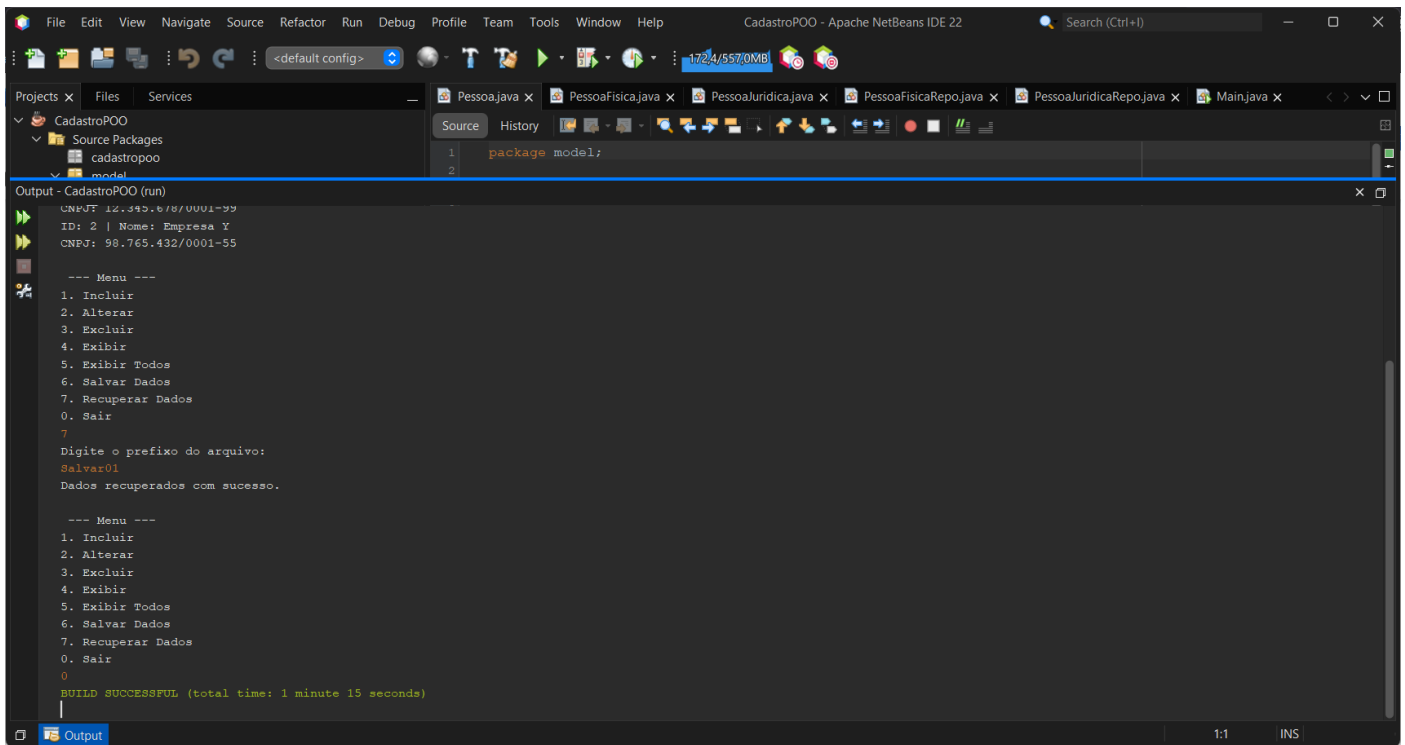
**Nona Imagem: Exibir todos (Físico) funcionando.**



**Decima Imagem: Salvar Dados funcionando.**



**Decima Primeira imagem: Recuperar dados funcionando.**



**Decima Segunda Imagem: Sair funcionando.**

## Análise e Conclusão

- **O que são elementos estáticos e qual o motivo para o método Main adotar esse modificador?**
  - Elementos estáticos (como métodos ou variáveis) pertencem à **classe** em si, e não a instâncias específicas da classe. Isso significa que podem ser acessados diretamente pela classe sem precisar criar um objeto.
- **Para que serve a classe Scanner?**
  - A classe **Scanner** em Java é usada para **ler a entrada de dados**. No nosso projeto, o Scanner foi utilizado para capturar a entrada do usuário no **menu interativo**, permitindo que o programa receba dados como IDs, nomes, CPFs, CNPJs, e outras informações diretamente do console.
- **Como o uso de classes de repositório impactou na organização do código?**

- O uso de **classes de repositório** melhorou significativamente a organização e a modularidade do código. Em vez de manipular diretamente os dados das entidades (PessoaFísica e PessoaJurídica) no método Main, os repositórios atuam como intermediários que **gerenciam as operações de CRUD** (inserir, alterar, excluir e recuperar) e a **persistência em arquivos**.